

.REM Z

IDENTIFICATION

PRODUCT CODE: AC-T052A-MC

PRODUCT NAME: CJKDIAO 11/238 SLU LTC REPAIR

PRODUCT DATE: SEPT-81

MAINTAINER: DIAGNOSTIC ENGINEERING

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C): 1982 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	DEC/X11

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

670	BASIC DEFINITIONS
715	ACT11 HOOKS
716	APT PARAMETER BLOCK
719	COMMON TAGS
(2)	APT MAILBOX-ETABLE
(1)	ERROR POINTER TABLE
1287	INITIALIZE THE COMMON TAGS
1288	GET VALUE FOR SOFTWARE SWITCH REGISTER
1304	T1 TEST ABILITY TO ACCESS SLU1 TCSR
1316	T2 TEST ABILITY TO ACCESS SLU1 TBUF
1327	T3 TEST SLU1 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1365	T4 TEST THAT SLU1 TCSR 'DONE' SETS WITH RESET
1380	T5 TEST ABILITY TO ACCESS SLU1 RCSR
1391	T6 TEST ABILITY TO ACCESS SLU1 RBUF
1405	T7 TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET
1445	T10 TEST THAT SLU1 BIT6 OF RCSR CAN BE SET & RESET
1489	T11 TEST ABILITY TO ACCESS SLU2 TCSR
1501	T12 TEST ABILITY TO ACCESS SLU2 TBUF
1511	T13 TEST SLU2 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
1530	T14 TEST THAT SLU2 TCSR 'DONE' SETS WITH RESET
1545	T15 TEST ABILITY TO ACCESS SLU2 RCSR
1557	T16 TEST ABILITY TO ACCESS SLU2 RBUF
1569	T17 TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
1591	T20 TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET
1635	T21 TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
1681	T22 TEST THAT SLU2 RCVR DONE (?) SET & CLEAR PROPERLY
1707	T23 TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
1719	T24 TEST ABILITY TO ACCESS LKS
1731	T25 TEST THAT BIT6 OF LKS CAN BE SET & RESET
1784	T26 UNIQUE INTERNAL ADDRESS TEST
1822	T27 TEST THAT SLU1 XMIT INTERRUPTS ONLY WHEN ENABLED
1853	T30 TEST SLU1 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
1881	T31 TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS
1910	T32 TEST THAT SLU1 XMIT INTERRUPT CLEARS WITH LOADING TBUF
1935	T33 TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED
1960	T34 TEST SLU2 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
1990	T35 TEST SLU2 TRANSMITTER FOR DOUBLE INTERRUPTS
2019	T36 TEST THAT SLU2 XMIT INTERRUPT CLEARS WITH LOADING TBUF
2049	T37 TEST THAT RCVR INTERRUPTS ONLY WHEN ENABLED
2076	T40 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
2104	T41 TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
2135	T42 TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2158	T43 TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
2183	T44 TEST SLU2 THAT 'OVERRUN & ERROR' BITS CAN BE SET
2204	T45 TEST THAT BREAK TRANSMITS ALL ZEROES
2228	T46 TEST THAT 'FR' ERROR CAN BE SET DURING BREAK
2247	T47 TEST DATA PATHS USING WRAP CABLE
2278	T50 TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
2331	T51 TEST RTC FOR DOUBLE INTERRUPTS
2365	T52 TEST THAT RTC INTERRUPT CLEARS WITH RESET
2391	T53 TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY
2494	END OF PASS ROUTINE
2565	ERROR MESSAGE TYPEOUT ROUTINE
2569	POWER DOWN AND UP ROUTINES
2615	SCOPE HANDLER ROUTINE
2620	APT COMMUNICATIONS ROUTINE

CJKDIAO 11/23B SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82^D 16.42
TABLE OF CONTENTS

SFO 0003

2681	TYPE ROUTINE
2682	BINARY TO OCTAL (ASCII) AND TYPE
2683	CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
2687	TTY INPUT ROUTINE
2692	TRAP DECODER
(3)	TRAP TABLE
2695	ECHO TEST
2758	TERMINAL OUTPUT TEST

CJKDIAO 11/23B SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82^{E 1} 16:42 PAGE 2

SEQ 0004

36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

HISTORY

REVISION A FIRST RELEASE OF DIAGNOSTIC

TABLE OF CONTENTS

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

1.0 GENERAL PROGRAM INFORMATION
1.1 ABSTRACT
1.2 SYSTEM REQUIREMENTS
 A. HARDWARE REQUIREMENTS
 B. SOFTWARE ENVIRONMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 PREREQUISITE DIAGNOSTICS
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 LOADING AND STARTING PROCEDURES
2.1.1 STARTING PROCEDURE
2.2 EXECUTION TIMES

3.0 ERROR INFORMATION
3.1 ERROR REPORTING PROCEDURES
3.2 ERROR HALTS

4.0 PERFORMANCE AND PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 PROGRAM DESCRIPTION
6.1 PROGRAM EXECUTION CHARACTERISTICS
6.2 SUBTEST SUMMARIES
6.3 SPECIAL SUBROUTINE DESCRIPTIONS
6.3.1 ECHO TEST
6.3.2 TERMINAL OUTPUT TEST

7.0 LISTING

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

THIS PROGRAM TESTS BOTH SERIAL LINE UNITS (SLU'S) AND THE LINE TIME CLOCK (LTC) ON THE M8189 MODULE. ITS MAIN PURPOSE IS TO PROVIDE SCOPE LOOPING FOR REPAIR PERSONNEL.
ERROR TYPE-OUTS IDENTIFY A FUNCTION BEING DONE OR A FUNCTION THAT FAILED AND TO WHAT LOGICAL PORTION OF THE BOARD IT FAILED ON (I.E., TRIED TO SET BIT 6 ON CSR OF SLU). THIS PROGRAM IS BASICALLY A REWRITE OF THE DL11-W DIAGNOSTIC AND THEREFORE IS WRITTEN IN MACRO-11 USING THE SYSMAC MACRO PACKAGE. IT

IS COMPATIBLE WITH ALL EXISTING MANUFACTURING AND FIELD SERVICE AUTO-
MATED TEST SYSTEMS.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

- 11/23B CPU MODULE
- MINIMUM 8K OF MEMORY
- TURN AROUND JUMPER INSTALLED ON SLU2

B. SOFTWARE ENVIRONMENTS

- XXDP STAND-ALONE
- XXDP CHAIN MODE
- APT
- ACT
- SLIDE

1.3 RELATED DOCUMENTS AND STANDARDS

- DIAGNOSTIC ENGINEERING FUNCTIONAL SPECIFICATION FOR 11/24 ON
BOARD OPTIONS TEST
BGI-79-004-00-U
- STANDARD APT SYSTEM TO PDP-11 DIAGNOSTIC INTERFACE REV. 15
16-FEBRUARY-76 APT SYSTEM GROUP
- DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS
175-003-009-02

1.4 PREREQUISITE DIAGNOSTICS

THIS PROGRAM ASSUMES THE CORRECT OPERATION OF THE CPU INSTRUCTION SET.
THIS IS TO BE VERIFIED BY EITHER.

CJKDBXX DCF11-AA CPU DIAGNOSTIC
OR
CJKDEXX F11 QUICK TEST

1.5 ASSUMPTIONS

IT IS ASSUMED THAT THE CONSOLE DEVICE THAT IS CONNECTED TO SLU1 IS OP-
ERATING CORRECTLY. IF THE TERMINAL IS NOT OPERATING CORRECTLY FALSE
INDICATIONS CAN BE EXPECTED.

2.0 OPERATING INSTRUCTIONS

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

2.1 LOADING AND STARTING PROCEDURES

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES.

2.1.1 STARTING PROCEDURE -

LOAD THE SWITCH REGISTER WITH SETTING (SOFTWARE SWITCH REGISTER LOCATION = 176)

A. START AT 200.
AFTER CHECKING THE TRANSMITTER, THE PROGRAM WILL PRINT ITS IDENTIFICATION AND REPORT THE NUMBER OF DEVICES UNDER TEST (NUMBER IS OCTAL). 'END PASS' IS PRINTED AFTER A FULL PASS HAS BEEN MADE ON ALL DEVICES UNDER TEST.

B. START AT 204. *****NOTE*****
THE 'ECHO' TEST WILL BE EXECUTED. AN '*' IS PRINTED AT THE BEGINNING OF THE TEST. THE ECHO TEST READS A CHARACTER FROM THE TERMINAL, WRITES THAT CHARACTER TO THE TERMINAL, AND REPORTS ANY ERROR FLAGS SET IN THE RECEIVER BUFFER. A CONTROL-C (^C) HALTS THE TEST AND PRINTS 'STOP' AT THE TERMINAL CONTINUING RESTARTS THE ECHO TEST.

C. START AT 210. *****NOTE*****

THE TERMINAL OUTPUT TEST WILL BE EXECUTED. DEPRESSING ANY CHARACTER AT THE TERMINAL HALTS THE TEST. CONTINUING RESTARTS THE TEST. THE TEST OUTPUTS 32 CHARACTERS ON A LINE AND REPEATS THE PATTERN EVERY THREE LINES. THE PATTERN IS AS FOLLOWS (OCTAL CODE 040 --> 377):

THIS BOTTOM LINE COULD BE THE FOLLOWING IF THE TERMINAL DOES NOT HAVE LOWER CASE:

@ABCDEFGHIJKLMN0PQRSTUVWXYZ[[UPPER CASE ALPHA]

2.2 PROGRAM OPTIONS

BIT15 - HALT ON ERROR
BIT14 - SCOPE LOOP
BIT13 - INHIBIT ERROR TYPEOUT
BIT12 - UNUSED
BIT11 - UNUSED
BIT10 - INHIBIT ERROR FLAGS TEST

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

BIT09 - LOOP ON ERROR
BIT08 - UNUSED
BIT07 - DISABLE SLU2 DATA TEST
BIT06 - INHIBIT LTC TESTS
BIT05 - INHIBIT ALL SLU TESTS (BOTH SLUS)
BIT04 - INHIBIT SLU1 TESTING
BIT03 - INHIBIT SLU2 TESTING
BIT02 - UNUSED
BIT01 - UNUSED
BIT00 - UNUSED

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF THE SOFTWARE SWITCH REGISTER DURING EXECUTION. TO DO THIS THE OPERATOR MUST TYPE A "CNTL G" (TYPING A "G" WITH THE "CTRL" KEY HELD DOWN AT THE SAME TIME). THIS IS PROCESSED AT KEY TIMES DURING THE PROGRAM (I.E. ON ERRORS, IN BETWEEN EACH TEST). A PROBLEM CAN OCCUR SINCE THE PROGRAM MAY BE TESTING THE SLU CONNECTED TO THE CONSOLE DEVICE. IF THIS HAPPENS IT SHOULD NOT CAUSE AN ERROR BUT THE "CNTL G" MAY BE LOST, SO IF THE PROGRAM DOES NOT RESPOND TO THE FIRST "CNTL G" TYPE A FEW MORE UNTIL THE RESPONSE IS RECEIVED. WHEN THE "CNTL G" IS RECEIVED THE PROGRAM WILL RESPOND WITH:

SWR = XXXXXX NEW =

WHERE XXXXXX IS EQUAL TO THE PRESENT SWITCH REGISTER CONTENTS IN OCTAL. THE OPERATOR CAN THEN TYPE;

1. <CR> IF NO CHANGES ARE TO BE MADE.
2. 6 DIGITS <CR> TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER SETTING.
3. CONTROL-U IF THE OPERATOR MAKES AN ERROR WHILE INPUTTING THE NEW SWITCH REGISTER SETTING.

2.3 EXECUTION TIMES

1ST PASS RUNTIME(WORST CASE).....15 SECONDS
LONGEST TEST TIME.....12.5 SECONDS
ADDITIONAL RUN TIME(EXTRA UNITS).....NONE
LONGEST PASS TIME.....15 SECONDS

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

IF A ROUTINE FAILS AND THE INHIBIT ERROR TIMEOUT (BIT13) OF THE SWR IS

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

NOT SET, A PRINTOUT RESULTS IN THE FORM:

```
''(SOME ASCII MESSAGE)''  
TEST   ERR PC   RCSR   [ANY APPLICABLE DATA HEADINGS]  
XXXXXX XXXXXX XXXXXX [ANY APPLICABLE DATA]
```

NOTE: 'RCSR' IS DEPENDENT ON THE FAILURE THEREFORE
COULD BE TCSR, RBUF, TBUR, OR I.KS

WHERE 'XXXXXX' IS AN OCTAL NUMBER.

THIS ERROR PRINTOUT OCCURS PROVIDED THE ERROR THAT EXISTS WOULD NOT
HINDER THE TYPEOUT. IN CASES WHERE IT IS NOT POSSIBLE TO PRINT AN
ERROR MESSAGE (I.E. FATAL CONSOLE TRANSMITTER FAILURES), A HALT
OCCURS. (SEE SECTION 3.2 FOR ERROR HALT INFORMATION.)

NOTE

FOR SOFTWARE SWITCH OPERATION, THE
SWITCH REGISTER CAN BE HANGED BY TYPING
A CONTROL-G AT THE CONSOLE DURING ERROR
PRINTOUTS. AFTER CONTINUING FROM THE
ERROR HALT THE OLD SWR CONTENTS IS DIS-
PLAYED AND THE NEW CONTENTS CAN BE EN-
TERED. IF ERROR HALTS ARE DISABLED, THE
CONTROL-G RESPONSE OCCURS IMMEDIATELY
FOLLOWING THE TYPEOUT.

3.2 ERROR HALTS

THERE ARE 4 ERRORS IN THIS PROGRAM THAT CAUSE A HALT. THESE ERRORS
ARE IN TESTS 1, 2, AND 3.

TEST 1 ERROR - ACCESS TO SLU1 TRANSMITTER CSR CAUSE TIME-OUT TRAP.
THIS PROBLEM WILL PROBABLY CAUSE AN INABILITY OF THE
MICRO-ODT TO RUN.

TEST 2 ERROR - ACCESS TO SLU1 TRANSMITTER DATA BUFFER CAUSED TIME-OUT
TRAP. THIS PROBLEM WILL PROBABLY CAUSE AN INABILITY OF
THE MICRO-ODT TO RUN.

TEST 3 ERROR - THE FIRST ERROR IS THAT DONE DID NOT CLEAR WHEN THE
TRANSMITTER BUFFER WAS FILLED AS IT SHOULD.

THE SECOND ERROR INDICATES THAT DONE DID NOT RESET IN A
REASONABLE TIME AFTER THE DATA BUFFER WAS FILLED, INDI-
CATING THAT THE CHARACTER WAS NEVER TRANSMITTED. THIS
COULD CAUSE MICRO-ODT TO NOT RUN OR GARBLED OUTPUT FROM
THE MICRO-ODT.

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

4.0 PERFORMANCE AND PROGRESS REPORTS

THE ONLY REPORT FROM THIS PROGRAM OTHER THAN ERROR REPORTS IS THE END PASS MESSAGE. IT IS IN THE FORM:

END PASS #XXXXX

WHERE XXXXX IS THE DECIMAL NUMBER OF END OF PASSES COMPLETED.

5.0 DEVICE INFORMATION TABLES

SLU1 RCSR

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I									I	I	I					I
									I	I						
RECEIVER DONE									I	I						
RECEIVER INTERRUPT ENABLE																

SLU1 RBUF

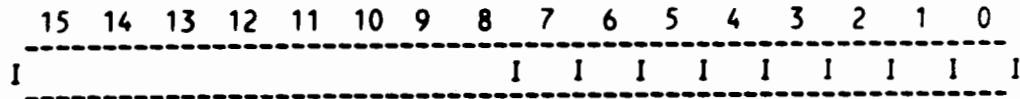
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	I	I	I	I					I	I	I	I	I	I	I	I
	I	I	I	I												I
ERROR-		I	I	I												I
OVERRUN			I	I												I
FRAME ERROR				I												I
RECEIVE PARITY																I
ERROR																I
RECEIVED DATA BITS (8)																

SLU1 XCSR

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I									I	I	I					I
									I	I						
TRANSMITTER READY																
TRANSMITTER INTERRUPT ENABLE																I

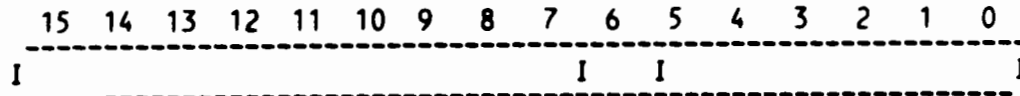
SLU1 XBUF

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



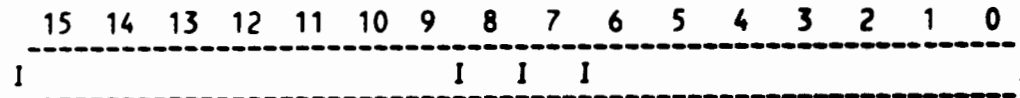
TRANSMITTER DATA BITS (8) -----

LTC CSR



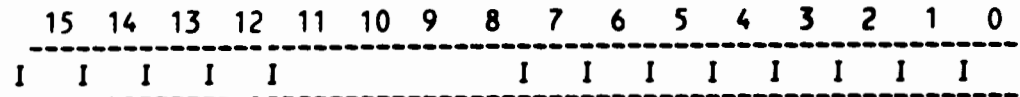
INTERRUPT ENABLE -----

SLU2 RCSR



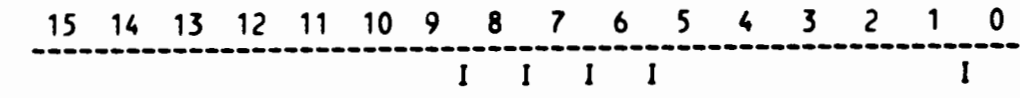
RECEIVER DONE -----
INTERRUPT ENABLE -----

SLU2 RBUF



ERROR - I I I I I
OVERRUN --- I I I I I
FRAME ERROR --- I I I I I
RECEIVER PARITY --- I I I I I
ERROR I I I I I
RECEIVER DATA BITS (8) -----

SLU2 XCSR



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

```

-----
                                I  I
                                I  I
TRANSMITTER DONE -----
INTERRUPT ENABLE -----
BREAK -----

```

```

SLU2 XBUF
 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
-----
I           I  I  I  I  I  I  I  I  I
-----
                                I
                                I
TRANSMITTER DATA BITS (8) -----

```

6.0 PROGRAM DESCRIPTION

6.1 PROGRAM EXECUTION CHARACTERISTICS

THIS PROGRAM TESTS ALL THE SELECTED DEVICES AS A SINGLE UNIT. IT FIRST VERIFIES THAT ALL REGISTERS CAN BE ACCESSED, THE WRITEABLE BITS ARE ABLE TO BE WRITTEN TO AND ARE UNIQUE ON ALL THREE DEVICES. IT THEN CHECKS EACH DEVICE FOR INTERRUPTS AND FOR DATA RELIABILITY. FINALLY IT SETS ALL SELECTED DEVICES UP TO GO AT THE SAME TIME ENABLES INTERRUPTS AND STARTS THEM OFF. THIS TEST CHECKS SYSTEM INTERACTION. WHEN ALL TESTS HAVE BEEN COMPLETED THE END OF PASS MESSAGE IS TYPED.

6.2 SUBTEST SUMMARIES

- TEST1 TEST ABILITY TO ACCESS SLU1 TRANSMITTER CONTROL AND STATUS REGISTER. HALTS IF ACCESS CAUSES TIMEOUT TRAP.
- TEST2 TEST ABILITY TO ACCESS SLU1 TRANSMITTER BUFFER. HALTS IF ACCESS CAUSES TIMEOUT TRAP.
- TEST3 TEST SLU1 TRANSMITTER BIT7 (DONE) CLEARS WHEN TRANSMITTER BUFFER IS LOADED. THE BUFFER IS LOADED WHICH SHOULD CLEAR

496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536

THE DONE BIT. AFTER IT IS VERIFIED THAT 'DONE' CLEARS THE PROGRAM WAITS TO RECEIVE DONE BACK AFTER THE DATA IS TRANSFERRED OUT OF THE BUFFER. IF DONE DOES NOT INITIALLY CLEAR OR FAILS TO RESET THE PROGRAM HALTS.

- TEST4 TEST THAT SLU1 TRANSMITTER BIT7 (DONE) SETS WITH RESET. THE TRANSMITTER BUFFER IS LOADED WITH A CHARACTER, AS SOON AS 'DONE' SETS A SECOND CHARACTER IS LOADED INTO THE BUFFER. BECAUSE THE FIRST CHARACTER IS STILL BEING SHIFTED OUT OF THE UART 'DONE' WILL NOT NORMALLY SET FOR AT LEAST 1 MS (DEPENDING ON BAUD RATE). THE PROGRAM ISSUES A RESET BEFORE THE TIME IS UP AND IMMEDIATELY CHECKS FOR 'DONE', IF IT IS SET THE PROGRAM ASSUMES IT WAS SET BY THE RESET INSTRUCTION. THIS ERROR DOES NOT CAUSE A HALT.
- TEST5 TEST ABILITY TO ACCESS SLU1 RECEIVER CONTROL AND STATUS REGISTER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.
- TEST6 TEST ABILITY TO ACCESS SLU1 RECEIVER BUFFER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.
- TEST10 TEST THAT SLU1 TRANSMITTER INTERRUPT ENABLE (BIT6) CAN BE SET AND RESET. THIS TEST CHECKS THAT THE BIT CAN BE WRITTEN INTO AND READ, CLEARED BY WRITING A ZERO TO IT AND CLEARED BY A 'RESET'.
- TEST11 TEST THAT SLU1 RECEIVER INTERRUPT ENABLE (BIT6) CAN BE SET AND RESET. SAME TEST AS FOR TRANSMITTER INTERRUPT ENABLE.
- TEST14 SAME AS TEST1 BUT DONE ONSLU2 AND ERROR DOES NOT CAUSE HALT.
- TEST15 SAME AS TEST2 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.
- TEST16 SAME AS TEST3 BUT FOR SLU2 AND ERROR DOES NOT CAUSE HALT.
- TEST17-21 SAME AS TEST4-6 BUT FOR SLU2.

538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593

TEST22 TEST SLU2 BREAK BIT (BIT0) CAN BE SET, CLEARED, AND RESET.

TEST23-26 SAME AS TEST 10-13 BUT FOR SLU2.

TEST27 TEST ABILITY TO ACCESS LINE CLOCK STATUS REGISTER. ERROR REPORTED IF ACCESS CAUSES TIMEOUT TRAP.

TEST30 TEST THAT LINE CLOCK INTERRUPT ENABLE (BIT6) CAN BE SET, CLEARED, AND "RESET".

TEST32 UNIQUE INTERNAL ADDRESS TEST. THIS TEST WRITES A BIT INTO ONE OF THE DEVICE REGISTERS AND THEN VERIFIES THAT BIT IS NOT SET IN OTHER REGISTERS. THE TEST IS REPEATED FOR OTHER REGISTERS.

TEST33 TEST THAT SLU1 TRANSMITTER INTERRUPTS ONLY WHEN ENABLED. THIS TEST CHECKS THAT THE TRANSMITTER ONLY INTERRUPTS WHEN ITS INTERRUPT ENABLE BIT IS SET.

TEST34 TEST SLU1 TRANSMITTER INTERRUPTS DO NOT OCCUR WHEN DISABLED. THIS TEST CHECKS THAT THE TRANSMITTER DOES NOT INTERRUPT WHEN THE PROCESSOR PRIORITY IS 7 OR THE INTERRUPT ENABLE BIT IS CLEARED.

TEST35 TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS. THIS TEST FIRST CHECKS THAT THE TRANSMITTER CAN INTERRUPT THEN MAKES SURE THAT ONLY ONE INTERRUPT IS REQUESTED FOR EACH SETTING OF "DONE".

TEST36 TEST THAT SLU1 TRANSMITTER INTERRUPT CLEARS WITH LOADING OF TRANSMITTER BUFFER. THIS TEST PUTS THE PROCESSOR AT 7, SETS TRANSMITTER INTERRUPT ENABLE AND FILLS THE TRANSMITTER BUFFER. WHEN "DONE" SETS THE SLU SHOULD HAVE AN INTERRUPT PENDING. THE PROGRAM THEN FILLS THE BUFFER AGAIN WHICH SHOULD CLEAR THE INTERRUPT.

TEST46-57 SAME AS TESTS 33-44 EXCEPT FOR SLU2 INSTEAD OF SLU1

TEST60 TEST THAT BREAK TRANSMITS ALL ZEROES. PUT ALL ONES INTO RECEIVER BUFFER THEN ISSUE A BREAK THE RECEIVER BUFFER SHOULD CONTAIN ZEROES.

TEST61 TEST THAT FRAMING ERROR (BIT13) CAN BE SET DURING BREAK. TRANSMIT A BREAK AND A CHARACTER JUST TO LET US KNOW WHEN TO LOOK FOR THE ERROR BIT. WHEN RECEIVER "DONE" SETS BOTH BREAK AND CHARACTER SHOULD BE THERE, CHECK FOR BIT 13 IN RECEIVER STATUS REGISTER.

TEST62 TEST SLU2 DATA PATH USING WRAP CABLE CONNECTOR. SAME AS TEST 45 ON SLU1.

594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
643
651
655
662
663
664
669
(1)
(1)
(1)
(1)
(1)
(1)

TEST63 TEST LINE TIME CLOCK INTERRUPTS PROPERLY.
TEST64 TEST LINE TIME CLOCK FOR DOUBLE INTERRUPTS.
TEST65 TEST THAT LINE TIME CLOCK INTERRUPT CLEARS WITH RESET.

TEST70 BLAST TEST. THIS TEST RUNS ALL SELECTED DEVICES SIMULTANEOUSLY IN INTERRUPT MODE. AFTER 60 INTERRUPTS FROM THE LINE CLOCK OR 256 (10) BYTES HAVE BEEN TRANSFERRED BY THE SLU'S EVERYTHING IS SHUT DOWN. THE PROGRAM THEN VERIFIES THAT NO INTERRUPTS WERE LOST ON EITHER SLU AND THAT THE DATA TRANSFERRED WAS CORRECT.

NOTE: IF RUNNING UNDER THE APT ENVIRONMENT MANY OF THE ABOVE TESTS ARE ONLY EXECUTED DURING FIRST PASS.

6.3 SPECIAL SUBROUTINE DESCRIPTIONS

6.3.1 ECHO TEST -

THIS ROUTINE WILL ECHO ANY CHARACTER TYPED ON EITHER SLU1 OR SLU2. DEFAULT IS TO THE CONSOLE DEVICE SLU1. THE TEST IS HALTED BY TYPING A CONTROL-C. TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING.

6.3.2 TERMINAL OUTPUT TEST -

THIS ROUTINE WILL OUTPUT ALL WRITEABLE CHARACTERS FOR THE OCTAL CODE 040 --> 377. 32 CHARACTERS ARE PRINTED ON EACH LINE. THE PATTERN IS REPEATED EVERY THREE LINES.

7.0 LISTING

%

.TITLE CJKDIAO 11/23B SLU LTC REPR DIAG
;*COPYRIGHT (C) SEPT-81
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*

```
(*)
(1)
(1)
(1) 000001
(1) 160000
670
(1)
(1) 001100
(1)
(1)
(1)
(1) 000011
(1) 000012
(1) 000015
(1) 000200
(1) 177776
(1)
(1) 177774
(1) 177772
(1) 177570
(1) 177570
(1)
(1)
(1) 000000
(1) 000001
(1) 000002
(1) 000003
(1) 000004
(1) 000005
(1) 000006
(1) 000007
(1) 000006
(1) 000007
(1)
(1) 000000
(1) 000040
(1) 000100
(1) 000140
(1) 000200
(1) 000240
(1) 000300
(1) 000340
(1)
(1) 100000
(1) 040000
(1) 020000
(1) 010000
(1) 004000
(1) 002000
(1) 001000
(*) 000400
(1) 000200
```

```

;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
;*
$TN=1
$SWR=160000      ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS
HT= 11      ;;CODE FOR HORIZONTAL TAB
LF= 12      ;;CODE FOR LINE FEED
CR= 15      ;;CODE FOR CARRIAGE RETURN
CRLF= 200   ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776  ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774      ;;STACK LIMIT REGISTER
PIRQ= 177772      ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570      ;;HARDWARE SWITCH REGISTER
DDISP= 177570     ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= X0      ;;GENERAL REGISTER
R1= X1      ;;GENERAL REGISTER
R2= X2      ;;GENERAL REGISTER
R3= X3      ;;GENERAL REGISTER
R4= X4      ;;GENERAL REGISTER
R5= X5      ;;GENERAL REGISTER
R6= X6      ;;GENERAL REGISTER
R7= X7      ;;GENERAL REGISTER
SP= X6      ;;STACK POINTER
PC= X7      ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
PR0= 0      ;;PRIORITY LEVEL 0
PR1= 40     ;;PRIORITY LEVEL 1
PR2= 100    ;;PRIORITY LEVEL 2
PR3= 140    ;;PRIORITY LEVEL 3
PR4= 200    ;;PRIORITY LEVEL 4
PR5= 240    ;;PRIORITY LEVEL 5
PR6= 300    ;;PRIORITY LEVEL 6
PR7= 340    ;;PRIORITY LEVEL 7

;*"SWITCH REGISTER" SWITCH DEFINITIONS
SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
```


(1) 000060
 (1) 000064
 (1) 000240
 671 176500
 672 000300
 673 000400
 674 000001
 675 161000
 676 000003
 677
 678
 679 000000
 680
 681
 682
 683
 684
 692
 693 000014 000014
 694 000014 012370
 695 000016 000340
 696
 697 000042 000042
 698 000042 000000
 699
 700
 702
 703
 704 000174 000174
 705 000174 000000
 706 000176 000000
 707
 708 000200 000200
 709 000200 000137 003016
 710 000204 000137 015406
 711 000210 000137 015664

TKVEC= 60 ;:TTY KEYBOARD VECTOR
 TPVEC= 64 ;:TTY PRINTER VECTOR
 PIRQVEC=240 ;:PROGRAM INTERRUPT REQUEST VECTOR
 ABASE= 176500
 AVECT1= 300
 AUSWR= 400
 \$TN= 1
 \$\$SWR= 161000
 BPT= 000003 ;:THIS IS THE COMMAND FOR A TRAP
 ; THROUGH 14 (BPT TRAP)
 .=0
 ;:*****
 ;:ALL UNUSED LOCATIONS FROM 4-776 CONTAIN A '+2,BPT'
 ;:SEQUENCE TO CATCH ILLEGAL TRAPS & INTERRUPTS
 ;:LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
 .=14 ;:THE BPT TRAP VECTOR POINTS TO THE
 .WORD CATCH ; ILLEGAL TRAP HANDLER 'CATCH'
 .WORD 340
 .= 42
 .WORD 0
 .= 174
 DISPREG: .WORD 0
 SWREG: .WORD 0
 .-200
 JMP START ;DO INTERFACE TEST
 JMP ECHO ;DO ECHO TEST
 JMP OUTTST ;DO OUTPUT TEST TO TERMINAL

713
714 000500
715
(1)
(2)
(1)
(1) 000500
(1) 000046
(1) 000046 012322
(1) 000052
(1) 000052 000000
(1) 000500
716
(1)
(2)
(1)
(2)
(1) 000500
(1) 000024 000024
(1) 000024 000200
(1) 000044 000044
(1) 000044 000500
(1) 000500
(2)
(1)
(1)
(1)
(1) 000500
(1) 000500 000000
(1) 000502 001100
(1) 000504 000050
(1) 000506 000060
(1) 000510 000055
(1) 000512 000030
717

```
. = 500
.SBTTL ACT11 HOOKS
:*****
:HOOKS REQUIRED BY ACT11
  $SVPC=          ;SAVE PC
  =46             ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
  $ENDAD          ;
  =52             ;;2)SET LOC.52 TO ZERO
  .WORD 0         ;; RESTORE PC
  = $SVPC
.SBTTL APT PARAMETER BLOCK
:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
  . $X=          ;;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
  =. $X        ;;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.
$APTHD:
$HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 50 ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 60 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 55 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE (WORDS)
```

719
(1)
(2)
(1)
(1)
(1)
(1) 001000
(1) 001000 000000
(1) 001002 000
(1) 001003 000
(1) 001004 000000
(1) 001006 000000
(1) 001010 000000
(1) 001012 000000
(1) 001014 000
(1) 001015 001
(1) 001016 000000
(1) 001020 000000
(1) 001022 000000
(1) 001024 000000
(1) 001026 000000
(1) 001030 000000
(1) 001032 000000
(1) 001034 000
(1) 001035 000
(1) 001036 000000
(1) 001040 177570
(1) 001042 177570
(1) 001044 177560
(1) 001046 177562
(1) 001050 177564
(1) 001052 177566
(1) 001054 000
(1) 001055 002
(1) 001056 012
(1) 001057 000
(3) 001060 000000
(3) 001062 000000
(3) 001064 000000
(3) 001066 000000
(3) 001070 000000
(1) 001072 000000
(1) 001074 077
(1) 001075 015
(1) 001076 000012
(2)
(2)
(2)
(3)
(2)
(2) 001100
(2) 001100 000000
(2) 001102 000000
(2) 001104 000000
(2) 001106 000000

.SBTTL COMMON TAGS

*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.
.=1000
\$CMTAG: .; START OF COMMON TAGS
\$TSTNM: .WORD 0 .; CONTAINS THE TEST NUMBER
\$ERFLG: .BYTE 0 .; CONTAINS ERROR FLAG
\$ICNT: .WORD 0 .; CONTAINS SUBTEST ITERATION COUNT
\$LPADR: .WORD 0 .; CONTAINS SCOPE LOOP ADDRESS
\$LPERR: .WORD 0 .; CONTAINS SCOPE RETURN FOR ERRORS
\$ERTTL: .WORD 0 .; CONTAINS TOTAL ERRORS DETECTED
\$ITEMB: .BYTE 0 .; CONTAINS ITEM CONTROL BYTE
\$ERMAX: .BYTE 1 .; CONTAINS MAX. ERRORS PER TEST
\$ERRPC: .WORD 0 .; CONTAINS PC OF LAST ERROR INSTRUCTION
\$GDADR: .WORD 0 .; CONTAINS ADDRESS OF 'GOOD' DATA
\$BDADR: .WORD 0 .; CONTAINS ADDRESS OF 'BAD' DATA
\$GDDAT: .WORD 0 .; CONTAINS 'GOOD' DATA
\$BDDAT: .WORD 0 .; CONTAINS 'BAD' DATA
 .WORD 0 .; RESERVED--NOT TO BE USED
\$AUTOB: .BYTE 0 .; AUTOMATIC MODE INDICATOR
\$INTAG: .BYTE 0 .; INTERRUPT MODE INDICATOR
 .WORD 0
\$SWR: .WORD DSWR .; ADDRESS OF SWITCH REGISTER
\$DISPLAY: .WORD DDISP .; ADDRESS OF DISPLAY REGISTER
\$TKS: 177560 .; TTY KBD STATUS
\$TKB: 177562 .; TTY KBD BUFFER
\$TPS: 177564 .; TTY PRINTER STATUS REG. ADDRESS
\$TPB: 177566 .; TTY PRINTER BUFFER REG. ADDRESS
\$NULL: .BYTE 0 .; CONTAINS NULL CHARACTER FOR FILLS
\$FILLS: .BYTE 2 .; CONTAINS # OF FILLER CHARACTERS REQUIRED
\$FILLC: .BYTE 12 .; INSERT FILL CHARS. AFTER A 'LINE FEED'
\$TPFLG: .BYTE 0 .; 'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
\$TMP0: .WORD 0 .; USER DEFINED
\$TMP1: .WORD 0 .; USER DEFINED
\$TMP2: .WORD 0 .; USER DEFINED
\$TMP3: .WORD 0 .; USER DEFINED
\$TMP4: .WORD 0 .; USER DEFINED
\$ESCAPE: 0 .; ESCAPE ON ERROR ADDRESS
\$QUES: .ASCII /?/ .; QUESTION MARK
\$CRLF: .ASCII <15> .; CARRIAGE RETURN
\$LF: .ASCIIZ <12> .; LINE FEED

.SBTTL APT MAILBOX-ETABLE

\$MAIL: .; APT MAILBOX
\$MSGTY: .WORD AMSGTY .; MESSAGE TYPE CODE
\$FATAL: .WORD AFATAL .; FATAL ERROR NUMBER
\$TESTN: .WORD ATESTN .; TEST NUMBER
\$PASS: .WORD APASS .; PASS COUNT

(2) 001110 000000
(2) 001112 000000
(2) 001114 000000
(2) 001116 000000
(2) 001120
(2) 001120 000
(2) 001121 000
(2) 001122 000000
(2) 001124 000400
(2) 001126 000000
(2)
(2)
(2)
(2)
(2)
(2)
(2) 001130 000
(2) 001131 000
(2)
(2)
(2)
(2) 001132 000000
(2)
(2) 001134 000
(2) 001135 000
(2) 001136 000000
(2) 001140 000
(2) 001141 000
(2) 001142 000000
(2) 001144 000
(2) 001145 000
(2) 001146 000000
(2) 001150 000300
(2) 001152 000000
(2) 001154 176500
(2) 001156 000000
(2) 001160

\$DEVCT: .WORD ADEVCT ;;DEVICE COUNT
\$UNIT: .WORD AUNIT ;;I/O UNIT NUMBER
\$MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS
\$MSGLG: .WORD AMSGLG ;;MESSAGE LENGTH
\$ETABLE: ;;APT ENVIRONMENT TABLE
\$ENV: .BYTE AENV ;;ENVIRONMENT BYTE
\$ENVM: .BYTE AENVM ;;ENVIRONMENT MODE BITS
\$\$SWREG: .WORD ASWREG ;;APT SWITCH REGISTER
\$USWR: .WORD AUSWR ;;USER SWITCHES
\$CPUOP: .WORD ACPUOP ;;CPU TYPE,OPTIONS
:*
:*
:*
:*
:*
:*
:*
:*
:*
:*
BIT 15-11=CPU TYPE
11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
11/70=06,PDQ=07,Q=10
BIT 10=REAL TIME CLOCK
BIT 9=FLOATING POINT PROCESSOR
BIT 8=MEMORY MANAGEMENT
\$MAMS1: .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE
\$MTYP1: .BYTE AMTYP1 ;;MEM. TYPE,BLK#1
:*
:*
:*
:*
:*
MEM. TYPE BYTE -- (HIGH BYTE)
900 NSEC CORE=001
300 NSEC BIPOLAR=002
500 NSEC MOS=003
\$MADR1: .WORD AMADR1 ;;HIGH ADDRESS,BLK#1
:*
MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
\$MAMS2: .BYTE AMAMS2 ;;HIGH ADDRESS,M.S. BYTE
\$MTYP2: .BYTE AMTYP2 ;;MEM. TYPE,BLK#2
\$MADR2: .WORD AMADR2 ;;MEM.LAST ADDRESS,BLK#2
\$MAMS3: .BYTE AMAMS3 ;;HIGH ADDRESS,M.S.BYTE
\$MTYP3: .BYTE AMTYP3 ;;MEM. TYPE,BLK#3
\$MADR3: .WORD AMADR3 ;;MEM.LAST ADDRESS,BLK#3
\$MAMS4: .BYTE AMAMS4 ;;HIGH ADDRESS,M.S.BYTE
\$MTYP4: .BYTE AMTYP4 ;;MEM. TYPE,BLK#4
\$MADR4: .WORD AMADR4 ;;MEM.LAST ADDRESS,BLK#4
\$VECT1: .WORD AVECT1 ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
\$VECT2: .WORD AVECT2 ;;INTERRUPT VECTOR#2BUS PRIORITY#2
\$BASE: .WORD ABASE ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
\$DEVVM: .WORD ADEVVM ;;DEVICE MAP
\$ETEND:
.MEXIT

.SBTTL ERROR POINTER TABLE

.*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
.*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
.*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
.*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
.*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

.* EM ;;POINTS TO THE ERROR MESSAGE
.* DH ;;POINTS TO THE DATA HEADER
.* DT ;;POINTS TO THE DATA
.* DF ;;POINTS TO THE DATA FORMAT

\$ERRTB:

.\$ERRTB: .BLKW 20 ;THIS BLOCK OF 16 LOCATIONS IS HERE TO PACIFY SYSMAC

(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
001160
720
721 001160
722 001160 000020
723
724
725
726 001220 016010
727 001222 024731
728 001224 025512
729 001226 000000
730
731 001230 016052
732 001232 024756
733 001234 025522
734 001236 000000
735
736 001240 016111
737 001242 025003
738 001244 025532
739 001246 000000
740
741 001250 000004
742
743
744 001260 016150
745 001262 024731
746 001264 025512
747 001266 000000
748
749 001270 000004
750
751
752 001300 016206
753 001302 024731
754 001304 025512
755 001306 000000
756
757 001310 016254
758 001312 024731
759 001314 025512
760 001316 000000

EM5
DH5
DT5
0

;SLU1 TCSR DONE NOT SET WITH RESET
; 'TEST# ERR PC TCSR'
;\$TESTN,\$ERRPC,CTCSR

EM6
DH6
DT6
0

;SLU1 RCSR DID NOT RETURN SSYNC
; 'TEST# ERR PC RCSR'
;\$TESTN,\$ERRPC,CRCSR

EM7
DH7
DT7
0

;SLU1 RBUF DID NOT RETURN SSYNC
; 'TEST# ERR PC RBUF'
;\$TESTN,\$ERRPC,CRBUF

.BLKW 4

;MORE PACIFICATION

EM11
DH5
DT5
0

; 'CAN NOT SET BIT2 OF SLU1 TCSR'
; 'TEST# ERR PC TCSR'
;\$TESTN,\$ERRPC,CTCSR

.BLKW 4

;STILL MORE PACIFICATION

EM13
DH5
DT5
0

; 'RESET DID NOT CLEAR BIT2 OF SLU1 TCSR'
; 'TEST# ERR PC TCSR'
;\$TESTN,\$ERRPC,CTCSR

EM14
DH5
DT5
0

; 'BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET2
; 'TEST# ERR PC TCSR'
;\$TESTN,\$ERRPC,CTCSR

761				
762	001320	016324	EM15	:'SLU1 XMIT INTERRUPTED WITH PRIORITY 7''
763	001322	024731	DH5	:'TEST# ERR PC TCSR''
764	001324	025512	DT5	:\$TESTN,\$ERRPC,CTCSR
765	001326	000000	0	
766				
767	001330	016372	EM16	:'CAN NOT SET BIT6 OF SLU1 TCSR''
768	001332	024731	DH5	:'TEST# ERR PC TCSR''
769	001334	025512	DT5	:\$TESTN,\$ERRPC,CTCSR
770	001336	000000	0	
771				
772	001340	016430	EM17	:'CAN NOT CLEAR BIT6 OF SLU1 TCSR''
773	001342	024731	DH5	:'TEST# ERR PC TCSR''
774	001344	025512	DT5	:\$TESTN,\$ERRPC,CTCSR
775	001346	000000	0	
776				
777	001350	016470	EM20	:'RESET DID NOT CLEAR BIT6 OF SLU1 TCSR''
778	001352	024731	DH5	:'TEST# ERR PC TCSR''
779	001354	025512	DT5	:\$TESTN,\$ERRPC,CTCSR
780	001356	000000	0	
781				
782	001360	016536	EM21	:'BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET''
783	001362	024756	DH6	:'TEST# ERR PC RCSR''
784	001364	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
785	001366	000000	0	
786				
787	001370	016606	EM22	:'SLU1 RCVR INTERRUPT WITH PRIORITY 7''
788	001372	024756	DH6	:'TEST# ERR PC RCSR''
789	001374	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
790	001376	000000	0	
791				
792	001400	016652	EM23	:'CAN NOT SET BIT6 OF SLU1 RCSR''
793	001402	024756	DH6	:'TEST# ERR PC RCSR''
794	001404	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
795	001406	000000	0	
796				
797	001410	016710	EM24	:'CAN NOT CLEAR BIT6 OF SLU1 RCSR''
798	001412	024756	DH6	:'TEST# ERR PC RCSR''
799	001414	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
800	001416	000000	0	
801				
802	001420	016750	EM25	:'CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET2
803	001422	024756	DH6	:'TEST# ERR PC RCSR''
804	001424	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
805	001426	000000	0	
806				
807	001430	017023	EM26	:'SLU1 RECEIVER DONE NEVER SET''
808	001432	024756	DH6	:'TEST# ERR PC RCSR''
809	001434	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
810	001436	000000	0	
811				
812	001440	017060	EM27	:'RESET DID NOT CLEAR SLU1 RCVR DONE''
813	001442	024756	DH6	:'TEST# ERR PC RCSR''
814	001444	025522	DT6	:\$TESTN,\$ERRPC,CRCSR
815	001446	000000	0	
816				

817	001450	017123	EM30	;'READING SLU1 RBUF DID NOT CLEAR RCVR DONE''
818	001452	024756	DH6	;'TEST# ERR PC RCSR''
819	001454	025522	DT6	;\$TESTN,\$ERRPC,RCSR
820	001456	000000	0	
821				
822	001460	017175	EM31	;'SLU2 TCSR DID NOT RETURN SSYNC
823	001462	024731	DH5	;'TEST# ERR PC TCSR''
824	001464	025542	DT31	;\$TESTN,\$ERRPC,TCSR
825	001466	000000	0	
826				
827	001470	017234	EM32	;'SLU2 TBUF DID NOT RETURN SSYNC
828	001472	025030	DH32	;'TEST# ERR PC TBUF''
829	001474	025552	DT32	;\$TESTN,\$ERRPC,TBUF
830	001476	000000	0	
831				
832	001500	017273	EM33	;'SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL''
833	001502	024731	DH5	;'TEST# ERR PC TCSR''
834	001504	025542	DT31	;\$TESTN,\$ERRPC,TCSR
835	001506	000000	0	
836				
837	001510	017345	EM34	;'SLU2 TCSR DONE NOT SET AFTER TRANSMIT''
838	001512	024731	DH5	;'TEST# ERR PC TCSR''
839	001514	025542	DT31	;\$TESTN,\$ERRPC,TCSR
840	001516	000000	0	
841				
842	001520	017413	EM35	;'SLU2 TCSR DONE NOT SET WITH RESET''
843	001522	024731	DH5	;'TEST# ERR PC TCSR''
844	001524	025542	DT31	;\$TESTN,\$ERRPC,TCSR
845	001526	000000	0	
846				
847	001530	017455	EM36	;'SLU2 RCSR DID NOT RETURN SSYNC''
848	001532	024756	DH6	;'TEST# ERR PC RCSR''
849	001534	025562	DT36	;\$TESTN,\$ERRPC,RCSR
850	001536	000000	0	
851				
852	001540	017514	EM37	;'SLU2 RBUF DID NOT RETURN SSYNC''
853	001542	025003	DH7	;'TEST# ERR PC RBUF''
854	001544	025572	DT37	;\$TESTN,\$ERRPC,RBUF
855	001546	000000	0	
856				
857	001550	017553	EM40	;'BITO OF SLU2 TCSR NOT CLEAR AFTER RESET''
858	001552	024731	DH5	;'TEST# ERR PC TCSR
859	001554	025542	DT31	;\$TESTN,\$ERRPC,TCSR
860	001556	000000	0	
861				
862	001560	017623	EM41	;'CAN NOT SET BITO OF SLU2 TCSR''
863	001562	024731	DH5	;'TEST# ERR PC TCSR
864	001564	025542	DT31	;\$TESTN,\$ERRPC,TCSR
865	001566	000000	0	
866				
867	001570	017661	EM42	;'CAN NOT CLEAR BITO OF SLU2 TCSR''
868	001572	024731	DH5	;'TEST# ERR PC TCSR
869	001574	025542	DT31	;\$TESTN,\$ERRPC,TCSR
870	001576	000000	0	
871				
872	001600	017721	EM43	;'RESET DID NOT CLEAR BITO OF SLU2 TCSR''

873	001602	024731	DH5	:'TEST# ERR PC TCSR
874	001604	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
875	001606	000000	0	
876				
877	001610	017767	EM44	:'BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET2
878	001612	024731	DH5	:'TEST# ERR PC TCSR
879	001614	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
880	001616	000000	0	
881				
882	001620	020037	EM45	:'SLU2 XMIT INTERRUPTED WITH PRIORITY 7'
883	001622	024731	DH5	:'TEST# ERR PC TCSR
884	001624	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
885	001626	000000	0	
886				
887	001630	020105	EM46	:'CAN NOT SET BIT6 OF SLU2 TCSR'
888	001632	024731	DH5	:'TEST# ERR PC TCSR
889	001634	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
890	001636	000000	0	
891				
892	001640	020143	EM47	:'CAN NOT CLEAR BIT6 OF SLU2 TCSR'
893	001642	024731	DH5	:'TEST# ERR PC TCSR
894	001644	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
895	001646	000000	0	
896				
897	001650	020203	EM50	:'RESET DID NOT CLEAR BIT6 OF SLU2 TCSR'
898	001652	024731	DH5	:'TEST# ERR PC TCSR
899	001654	025542	DT31	:'\$TESTN,\$ERRPC,TCSR
900	001656	000000	0	
901				
902	001660	020251	EM51	:'BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET'
903	001662	024756	DH6	:'TEST# ERR PC RCSR'
904	001664	025562	DT36	:'\$TESTN,ERRPC,RCSR
905	001666	000000	0	
906				
907	001670	020321	EM52	:'SLU2 RCVR INTERRUPT WITH PRIORITY 7'
908	001672	024756	DH6	:'TEST# ERR PC RCSR'
909	001674	025562	DT36	:'\$TESTN,ERRPC,RCSR
910	001676	000000	0	
911				
912	001700	020365	EM53	:'CAN NOT SET BIT6 OF SLU2 RCSR'
913	001702	024756	DH6	:'TEST# ERR PC RCSR'
914	001704	025562	DT36	:'\$TESTN,ERRPC,RCSR
915	001706	000000	0	
916				
917	001710	020423	EM54	:'CAN NOT CLEAR BIT6 OF SLU2 RCSR'
918	001712	024756	DH6	:'TEST# ERR PC RCSR'
919	001714	025562	DT36	:'\$TESTN,ERRPC,RCSR
920	001716	000000	0	
921				
922	001720	020463	EM55	:'CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET2
923	001722	024756	DH6	:'TEST# ERR PC RCSR'
924	001724	025562	DT36	:'\$TESTN,ERRPC,RCSR
925	001726	000000	0	
926				
927	001730	020536	EM56	:'SLU2 RECEIVER DONE NEVER SET'
928	001732	024756	DH5	:'TEST# ERR PC RCSR'

929	001734	025562	DT36	;\$TESTN,ERRPC,RCSR
930	001736	000000	0	
931				
932	001740	020573	EM57	;'RESET DID NOT CLEAR SLU2 RCVR DONE''
933	001742	024756	DH6	;'TEST# ERR PC RCSR''
934	001744	025562	DT36	;\$TESTN,ERRPC,RCSR
935	001746	000000	0	
936				
937	001750	020636	EM60	;'READING SLU2 RBUF DID NOT CLEAR RCVR DONE''
938	001752	024756	DH6	;'TEST# ERR PC RCSR''
939	001754	025562	DT36	;\$TESTN,ERRPC,RCSR
940	001756	000000	0	
941				
942	001760	020710	EM61	;'LKS DID NOT RETURN SSYNC
943	001762	025055	DH61	;'TEST# ERR PC LKS''
944	001764	025602	DT61	;\$TESTN,ERRPC,LKS
945	001766	000000	0	
946				
947	001770	020741	EM62	;'BIT6 OF LKS NOT CLEAR AFTER RESET''
948	001772	025055	DH61	;'TEST# ERR PC LKS''
949	001774	025602	DT61	;\$TESTN,\$ERRPC,LKS
950	001776	000000	0	
951				
952	002000	021003	EM63	;'LKS INTERRUPT WITH PRIORITY 7''
953	002002	025055	DH61	;'TEST# ERR PC LKS''
954	002004	025602	DT61	;\$TESTN,\$ERRPC,LKS
955	002006	000000	0	
956				
957	002010	021041	EM64	;'CAN NOT SET BIT6 OF LKS''
958	002012	025055	DH61	;'TEST# ERR PC LKS''
959	002014	025602	DT61	;\$TESTN,\$ERRPC,LKS
960	002016	000000	0	
961				
962	002020	021071	EM65	;'CAN NOT CLEAR BIT6 OF LKS''
963	002022	025055	DH61	;'TEST# ERR PC LKS''
964	002024	025602	DT61	;\$TESTN,\$ERRPC,LKS
965	002026	000000	0	
966				
967	002030	021123	EM66	;'RESET DID NOT CLEAR BIT6 OF LKS''
968	002032	025055	DH61	;'TEST# ERR PC LKS''
969	002034	025602	DT61	;\$TESTN,\$ERRPC,LKS
970	002036	000000	0	
971				
972	002040	021163	EM67	;'BIT7 OF LKS NOT SET AFTER RESET2
973	002042	025055	DH61	;'TEST# ERR PC LKS''
974	002044	025602	DT61	;\$TESTN,\$ERRPC,LKS
975	002046	000000	0	
976				
977	002050	021223	EM70	;'CAN NOT CLEAR BIT7 OF LKS''
978	002052	025055	DH61	;'TEST# ERR PC LKS''
979	002054	025602	DT61	;\$TESTN,\$ERRPC,LKS
980	002056	000000	0	
981				
982	002060	021255	EM71	;'BIT7 OF LKS DOES NOT SET''
983	002062	025055	DH61	;'TEST# ERR PC LKS''
984	002064	025602	DT61	;\$TESTN,\$ERRPC,LKS

Line No.	Address	Code	Description
985	002066	000000	0
986			
987	002070	021306	EM72 ;WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER
988	002072	025101	DH72 ;'TEST# ERR PC GOOD BAD GDDATA BDDATA
989	002074	025612	DT72 ;\$TESTN,\$ERRPC,\$GDADR,\$BDADR,\$GDDAT,\$BDDAT
990	002076	000000	0
991			
992	002100	000004	.BLKW 4 ;THE LAST IN A LONG LINE OF PACIFICATION
993			
994			
995	002110	021367	EM74 ;'SLU1 XMIT INTERRUPTS WHEN DISABLED''
996	002112	024731	DH5 ;'TEST# ERR PC TCSR''
997	002114	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
998	002116	000000	0
999			
1000	002120	021432	EM75 ;'SLU1 XMIT DID NOT INTERRUPT''
1001	002122	024731	DH5 ;'TEST# ERR PC TCSR''
1002	002124	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1003	002126	000000	0
1004			
1005	002130	021466	EM76 ;'SLU1 XMIT INTERRUPT AT PRIORITY 7''
1006	002132	024731	DH5 ;'TEST# ERR PC TCSR''
1007	002134	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1008	002136	000000	0
1009			
1010	002140	021530	EM77 ;'SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR''
1011	002142	024731	DH5 ;'TEST# ERR PC TCSR''
1012	002144	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1013	002146	000000	0
1014			
1015	002150	021577	EM100 ;'SLU1 XMIT DID NOT INTERRUPT''
1016	002152	024731	DH5 ;'TEST# ERR PC TCSR''
1017	002154	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1018	002156	000000	0
1019			
1020	002160	021633	EM101 ;'SLU1 XMIT RE-INTERRUPTED''
1021	002162	024731	DH5 ;'TEST# ERR PC TCSR''
1022	002164	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1023	002166	000000	0
1024			
1025	002170	021664	EM102 ;'LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT''
1026	002172	024731	DH5 ;'TEST# ERR PC TCSR''
1027	002174	025512	DT5 ;\$TESTN,\$ERRPC,CTCSR
1028	002176	000000	0
1029			
1030	002200	021736	EM103 ;'SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR''
1031	002202	024756	DH6 ;'TEST# ERR PC RCSR''
1032	002204	025522	DT6 ;\$TESTN,\$ERRPC,CRCSR
1033	002206	000000	0
1034			
1035	002210	022005	EM104 ;'SLU1 RCVR DID NOT INTERRUPT''
1036	002212	024756	DH6 ;'TEST# ERR PC RCSR''
1037	002214	025522	DT6 ;\$TESTN,\$ERRPC,CRCSR
1038	002216	000000	0
1039			
1040	002220	022041	EM105 ;'SLU1 RCVR INTERRUPTS AT PRIORITY 7''

1041	002222	024756	DH6	:'TEST# ERR PC RCSR''
1042	002224	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1043	002226	000000	0	
1044				
1045	002230	022104	EM106	:'SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR''
1046	002232	024756	DH6	:'TEST# ERR PC RCSR''
1047	002234	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1048	002236	000000	0	
1049				
1050	002240	022165	EM107	:'SLU1 RCVR DID NOT INTERRUPT''
1051	002242	024756	DH6	:'TEST# ERR PC RCSR''
1052	002244	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1053	002246	000000	0	
1054				
1055	002250	022221	EM110	:'SLU1 RECEIVER RE-INTERRUPTED''
1056	002252	024756	DH6	:'TEST# ERR PC RCSR''
1057	002254	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1058	002256	000000	0	
1059				
1060	002260	022256	EM111	:'SLU1 READING RBUF DID NOT CLEAR INTERRUPT''
1061	002262	024756	DH6	:'TEST# ERR PC RCSR''
1062	002264	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1063	002266	000000	0	
1064				
1065	002270	022330	EM112	:'RESET DID NOT CLEAR SLU1 RCVR INTERRUPT''
1066	002272	024756	DH6	:'TEST# ERR PC RCSR''
1067	002274	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1068	002276	000000	0	
1069				
1070	002300	022400	EM113	:'SLU1 'OR' FLAG DID NOT SET''
1071	002302	024756	DH6	:'TEST# ERR PC RCSR''
1072	002304	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1073	002306	000000	0	
1074				
1075	002310	022433	EM114	:'SLU1 'ERROR' NOT SET WITH 'OR' FLAG''
1076	002312	024756	DH6	:'TEST# ERR PC RCSR''
1077	002314	025522	DT6	;\$TESTN,\$ERRPC,CRCR
1078	002316	000000	0	
1079				
1080	002320	022477	EM115	:'DATA COMPARE ERROR''
1081	002322	025157	DH115	:'TEST# ERR PC CRCR GOOD BAD''
1082	002324	025630	DT115	;\$TESTN,\$ERRPC,CRCR,GD,BD
1083	002326	000000	0	
1084				
1085	002330	022522	EM116	:'SLU2 XMIT INTERRUPTS WHEN DISABLED''
1086	002332	024731	DH5	:'TEST# ERR PC TCSR''
1087	002334	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1088	002336	000000	0	
1089				
1090	002340	022565	EM117	:'SLU2 XMIT DID NOT INTERRUPT''
1091	002342	024731	DH5	:'TEST# ERR PC TCSR''
1092	002344	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1093	002346	000000	0	
1094				
1095	002350	022621	EM120	:'SLU2 XMIT INTERRUPT A' PRIORITY 7''
1096	002352	024731	DH5	:'TEST# ERR PC TCSR''

Line No	Code	Address	Label	Description
1097	002354	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1098	002356	000000	0	
1099				
1100	002360	022663	EM121	;'SLU2 XMIT INTERRUPTS WITH ENABLE CLFAR''
1101	002362	024731	DH5	;'TEST# ERR PC TCSR''
1102	002364	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1103	002366	000000	0	
1104				
1105	002370	022732	EM122	;'SLU2 XMIT DID NOT INTERRUPT''
1106	002372	024731	DH5	;'TEST# ERR PC TCSR''
1107	002374	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1108	002376	006000	0	
1109				
1110	002400	022766	EM123	;'SLU2 XMIT RE-INTERRUPTED''
1111	002402	024731	DH5	;'TEST# ERR PC TCSR''
1112	002404	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1113	002406	000000	0	
1114				
1115	002410	023017	EM124	;'LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT''
1116	002412	024731	DH5	;'TEST# ERR PC TCSR''
1117	002414	025542	DT31	;\$TESTN,\$ERRPC,TCSR
1118	002416	000000	0	
1119				
1120	002420	023071	EM125	;'SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR''
1121	002422	024756	DH6	;'TEST# ERR PC RCSR''
1122	002424	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1123	002426	000000	0	
1124				
1125	002430	023140	EM126	;'SLU2 RCVR DID NOT INTERRUPT''
1126	002432	024756	DH6	;'TEST# ERR PC RCSR''
1127	002434	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1128	002436	000000	0	
1129				
1130	002440	023174	EM127	;'SLU2 RCVR INTERRUPTS AT PRIORITY 7''
1131	002442	024756	DH6	;'TEST# ERR PC RCSR''
1132	002444	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1133	002446	000000	0	
1134				
1135	002450	023237	EM130	;'SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR''
1136	002452	024756	DH6	;'TEST# ERR PC RCSR''
1137	002454	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1138	002456	000000	0	
1139				
1140	002460	023320	EM131	;'SLU2 RCVR DID NOT INTERRUPT''
1141	002462	024756	DH6	;'TEST# ERR PC RCSR''
1142	002464	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1143	002466	000000	0	
1144				
1145	002470	023354	EM132	;'SLU2 RECEIVER RE-INTERRUPTED''
1146	002472	024756	DH6	;'TEST# ERR PC RCSR''
1147	002474	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1148	002476	000000	0	
1149				
1150	002500	023411	EM133	;'SLU2 READING RBUF DID NOT CLEAR INTERRUPT''
1151	002502	024756	DH6	;'TEST# ERR PC RCSR''
1152	002504	025562	DT36	;\$TESTN,\$ERRPC,RCSR

1153	002506	000000	0	
1154				
1155	002510	023463	EM134	;'RESET DID NOT CLEAR SLU2 RCVR INTERRUPT''
1156	002512	024756	DH6	;'TEST# ERR PC RCSR''
1157	002514	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1158	002516	000000	0	
1159				
1160	002520	023533	EM135	;'SLU2 'OR' FLAG DID NOT SET''
1161	002522	024756	DH6	;'TEST# ERR PC RCSR''
1162	002524	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1163	002526	000000	0	
1164				
1165	002530	023566	EM136	;'SLU2 'ERROR' NOT SET WITH 'OR' FLAG''
1166	002532	024756	DH6	;'TEST# ERR PC RCSR''
1167	002534	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1168	002536	000000	0	
1169				
1170	002540	023632	EM137	;'SLU2 BREAK DID NOT TRANSMIT ALL ZEROES''
1171	002542	025223	DH137	;'TEST# ERR PC RCSR DATA''
1172	002544	025644	DT137	;\$TESTN,\$ERRPC,RCSR,\$BDDAT
1173	002546	000000	0	
1174				
1175	002550	023701	EM140	;'BREAK DID NOT SET FRAMING ERROR''
1176	002552	024756	DH6	;'TEST# ERR PC RCSR''
1177	002554	025522	DT6	;\$TESTN,\$ERRPC,RCSR
1178	002556	000000	0	
1179				
1180	002560	023741	EM141	;'SLU2 'ERROR' NOT SET WITH 'FR' FLAG''
1181	002562	024756	DH6	;'TEST# ERR PC RCSR''
1182	002564	025562	DT36	;\$TESTN,\$ERRPC,RCSR
1183	002566	000000	0	
1184				
1185	002570	024005	EM142	;'DATA COMPARE ERROR WITH CABLE''
1186	002572	025260	DH142	;'TEST# ERR PC RCSR GOOD BAD''
1187	002574	025656	DT142	;\$TESTN,\$ERRPC,RCSR,GD,BD
1188	002576	000000	0	
1189				
1190	002600	024043	EM143	;'RTC INTERRUPT AT PRIORITY 7''
1191	002602	025055	DH61	;'TEST# ERR PC LKS''
1192	002604	025602	DT61	;\$TESTN,\$ERRPC,LKS
1193	002606	000000	0	
1194				
1195	002610	024077	EM144	;'RTC INTERRUPTS WHEN DISABLED''
1196	002612	025055	DH61	;'TEST# ERR PC LKS''
1197	002614	025602	DT61	;\$TESTN,\$ERRPC,LKS
1198	002616	000000	0	
1199				
1200	002620	024134	EM145	;'RTC INTERRUPT DID NOT OCCUR''
1201	002622	025055	DH61	;'TEST# ERR PC LKS''
1202	002624	025602	DT61	;\$TESTN,\$ERRPC,LKS
1203	002626	000000	0	
1204				
1205	002630	024170	EM146	;'RTC INTERRUPT DID NOT OCCUR''
1206	002632	025055	DH61	;'TEST# ERR PC LKS''
1207	002634	025602	DT61	;\$TESTN,\$ERRPC,LKS
1208	002636	000000	0	

1209				
1210	002640	024224	EM147	:'RTC DOUBLE INTERRUPT''
1211	002642	025055	DH61	:'TEST# ERR PC LKS''
1212	002644	025602	DT61	:\$TESTN,\$ERRPC,LKS
1213	002646	000000	0	
1214				
1215	002650	024251	EM150	:'RESET DID NOT CLEAR RTC INTERRUPT''
1216	002652	025055	DH61	:'TEST# ERR PC LKS''
1217	002654	025602	DT61	:\$TESTN,\$ERRPC,LKS
1218	002656	000000	0	
1219				
1220	002660	024313	EM151	:'RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS''
1221	002662	025055	DH61	:'TEST# ERR PC LKS''
1222	002664	025602	DT61	:\$TESTN,\$ERRPC,LKS
1223	002666	000000	0	
1224				
1225				
1226	002670	024422	EM153	:'SLU1 RECEIVER STATUS ERROR''
1227	002672	025157	DH115	:'TEST# ERR PC CRCSR GOOD BAD''
1228	002674	025630	DT115	:\$TESTN,\$ERRPC,CRCSR,\$GDDTA,\$BDDAT
1229	002676	000000	0	
1230				
1231	002700	024455	EM154	:'SLU2 RECEIVER STATUS ERROR
1232	002702	025260	DH142	:'TESTN ERR PC RCSR GOOD BAD''
1233	002704	025656	DT142	:\$TESTN,\$ERRPC,RCSR,\$GDDAT,\$BDDAT
1234	002706	000000	0	
1235				
1236				
1237	002710	024545	EM156	:'SLU1 DATA COMPARE ERROR IN EXERCISER''
1238	002712	025157	DH115	:'TEST# ERR PC CRCSR GOOD BAD''
1239	002714	025630	DT115	:\$TESTN,\$ERRPC,CRCSR,GD,BD
1240	002716	000000	0	
1241				
1242	002720	024612	EM157	:'INCORRECT RECEIVE COUNT SLU2
1243	002722	025361	DH155	:'TEST# ERR PC RCSR TRANS RCV''
1244	002724	025670	DT157	:\$TESTN,\$ERRPC,RCSR,XMTCT2,RCVCT2
1245	002726	000000	0	
1246				
1247	002730	024647	EM160	:'SLU2 DATA COMPARE ERROR IN EXERCISER''
1248	002732	025260	DH142	:'TEST# ERR PC RCSR GOOD BAD''
1249	002734	025656	DT142	:\$TESTN,\$ERRPC,RCSR,GD,BD
1250	002736	000000	0	
1251				
1252	002740	024714	EM161	:'TRAP CATCHER''
1253	002742	025425	DH161	:'TEST# ERR PC OLDPC TRAP ADR''
1254	002744	025704	DT161	:\$TESTN,\$ERRPC,OLDPC,BDVECT
1255	002746	000000	0	

;REGISTER ADDRESSES OF INTERNAL ON BOARD OPTIONS

1259	002750	176500	RCSR: .WORD 176500	;'SLU2 COMMAND/STATUS REGISTER
1260	002752	176502	RBUF: .WORD 176502	;'SLU2 RECEIVER BUFFER
1261	002754	176504	TCSR: .WORD 176504	;'SLU2 TRANSMITTER COMMAND/STATUS REGISTER
1262	002756	176506	TBUF: .WORD 176506	;'SLU2 TRANSMITTER BUFFER
1263	002760	177560	CRCSR: 177560	;'SLU1 RECEIVER COMMAND/STATUS REGISTER
1264	002762	177562	CRBUF: 177562	;'SLU1 RECEIVER BUFFER

1265 002764 177564
 1266 002766 177566
 1267 002770 177546
 1268
 1269
 1270
 1271
 1272 002772 000300
 1273 002774 000302
 1274 002776 000304
 1275 003000 000306
 1276 003002 000060
 1277 003004 000062
 1278 003006 000064
 1279 003010 000066
 1280 003012 000100
 1281 003014 000102
 1282
 1283 003016 005037 001102
 1284 003022 005037 001100
 1285 003026 005037 001104
 1286 003032 005037 001156
 1287 003036
 (1)
 (1)
 (1) 003036 012706 001000
 (1) 003042 005026
 (1) 003044 022706 001040
 (1) 003050 001374
 (1) 003052 012706 001000
 (1)
 (1) 003056 012737 013114 000020
 (1) 003064 012737 000340 000022
 (1) 003072 012737 012420 000030
 (1) 003100 012737 000340 000032
 (1) 003106 012737 015326 000034
 (1) 003114 012737 000340 000036
 (1) 003122 012737 012736 000024
 (1) 003130 012737 000340 000026
 (1) 003136 013737 012270 012262
 (1) 003144 005037 001072
 (1) 003150 112737 000001 001015
 (1) 003156 012737 003156 001006
 (1) 003164 012737 003164 001010
 (2)
 (2)
 (2) 003172 013746 000004
 (2) 003176 012737 003232 000004
 (2) 003204 012737 177570 001040
 (2) 003212 012737 177570 001042
 (2) 003220 022777 177777 175612
 (2) 003226 001012
 (2)
 (2) 003230 000403
 (2) 003232 012716 003240
 (2) 003236 000002

CTCSR: 177564 ;SLU1 TRANSMITTER COMMAND/STATUS REGISTER
 CTBUF: 177566 ;SLU1 TRANSMITTER BUFFER
 LKS: .WORD 177546 ;LTC COMMAND/STATUS REGISTER

;VECTOR ADDRESSES FOR ON BOARD OPTIONS

RVECT: .WORD 300
 RPSW: .WORD 302
 TVECT: .WORD 304
 TPSW: .WORD 306
 CRVECT: 60 ;RECEIVER INTERRUPT VECTOR
 CRPSW: 62
 CTVECT: 64 ;TRANSMITTER INTERRUPT VECTOR
 CTPSW: 66
 RTCVT: .WORD 100
 RTCPSW: .WORD 102
 START: CLR \$FATAL ;CLEAR ERROR NO.
 CLR \$MSGTYP ;CLEAR MESSAGE TYPE
 CLR \$TESTN ;CLEAR TEST NO.
 CLR \$DEVM ;CLEAR FLAGS INDICATING DEVICES UNDER TEST

1\$:
 .SBTTL INITIALIZE THE COMMON TAGS
 ;;CLEAR THE COMMON TAGS (\$CMTAG) AREA
 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
 CLR (R6)+ ;;CLEAR MEMORY LOCATION
 CMP #SWR,R6 ;;DONE?
 BNE -6 ;;LOOP BACK IF NO
 MOV #1000,SP ;;SETUP THE STACK POINTER
 ;;INITIALIZE A FEW VECTORS
 MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
 MOV #340,@IOTVEC+2 ;;LEVEL 7
 MOV #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
 MOV #340,@EMTVEC+2 ;;LEVEL 7
 MOV #TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
 MOV #340,@TRAPVEC+2;LEVEL 7
 MOV #SPWRDN,@PWRVEC ;;POWER FAILURE VECTOR
 MOV #340,@PWRVEC+2 ;;LEVEL 7
 MOV \$ENDCT,\$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
 CLR \$ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
 MOVB #1,\$ERMAX ;;ALLOW ONE ERROR PER TEST
 MOV #,\$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
 MOV #,\$LPERR ;;SETUP THE ERROR LOOP ADDRESS
 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
 MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
 MOV #64,\$@ERRVEC ;;SET UP ERROR VECTOR
 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
 BNE 66\$;;BRANCH IF NO TIMEOUT TRAP OCCURRED
 ;;AND THE HARDWARE SWR IS NOT = -1
 BR 65\$;;BRANCH IF NO TIMEOUT
 64\$: MOV #65\$,(SP) ;;SET UP FOR TRAP RETURN
 RTI

(2)	003240	012737	000176	001040	65\$:	MOV	#SWREG,SWR	::POINT TO SOFTWARE SWR
(2)	003246	012737	000174	001042		MOV	#DISPREG,DISPLAY	
(2)	003254	012637	000004		66\$:	MOV	(SP)+,@#ERRVEC	::RESTORE ERROR VECTOR
(1)								
(2)	003260	005037	001106			CLR	\$PASS	::CLEAR PASS COUNT
(2)	003264	132737	000200	001121		BITB	#APTSIZE,\$ENVM	::TEST USER SIZE UNDER APT
(2)	003272	001403				BEQ	67\$::YES,USE NON-APT SWITCH
(2)	003274	012737	001122	001040		MOV	#\$\$SWREG,SWR	::NO,USE APT SWITCH REGISTER
(2)	0C3302				67\$:			
1288					.SBTTL	GET VALUE FOR SOFTWARE SWITCH REGISTER		
(1)	003302	005737	000042			TST	@#42	::ARE WE RUNNING UNDER XXDP/ACT?
(1)	003306	001012				3NE	68\$::BRANCH IF YES
(1)	003310	123727	001120	000001		CMPB	\$ENV,#1	::ARE WE RUNNING UNDER APT?
(1)	003316	001406				BEQ	68\$::BRANCH IF YES
(1)	003320	023727	001040	000176		CMP	SWR,#SWREG	::SOFTWARE SWITCH REG SELECTED?
(1)	003326	001005				BNE	69\$::BRANCH IF NO
(1)	003330	104406				GTSWR		::GET SOFT-SWR SETTINGS
(1)	003332	000403				BR	69\$	
(1)	003334	112737	000001	001034	68\$:	MOVB	#1,\$AUTOB	::SET AUTO-MODE INDICATOR
(1)	0C3342				69\$:			
1289	003342	032777	000060	175470		BIT	#BIT4!BIT5,@SWR	:IS SLU1 TO BE TESTED
1290	003350	001003				BNE	2\$:IF EITHER BIT IS SET THEN DON'T TEST IT
1291	003352	052737	000001	001156		BIS	#BIT0,\$DEVN	:SET DEVICE FLAG TO TEST SLU1
1292	003360	032777	000050	175452	2\$:	BIT	#BIT3!BIT5,@SWR	:IS SLU2 TO BE TESTED
1293	003366	001003				BNE	3\$:IF EITHER BIT IS SET THEN DON'T TEST IT
1294	003370	052737	000002	001156		BIS	#BIT1,\$DEVN	:SET DEVICE FLAG TO TEST SLU2
1295	003376	032777	000100	175434	3\$:	BIT	#BIT6,@SWR	:IS LTC TO BE TESTED
1296	003404	001003				BNE	4\$:IF BIT IS SET THEN DON'T TEST IT.
1297	003406	052737	000004	001156		BIS	#BIT2,\$DEVN	:SET DEVICE FLAG TO TEST LTC
1298	003414	032737	000001	001156	4\$:	BIT	#BIT0,\$DEVN	:IS SLU1 UNDER TEST
1299	003422	001002				BNE	TST1	:IF YES TEST XMIT REG. BEFORE TYPING TITLE
1300	003424	000137	003660			JMP	ID	:IF NO SKIP TESTS AND TYPE IT NOW

```
1302
1303
1304
(3)
(3)
(2) 003430 000004
(2) 003432 012737 000001 001104
1305 003440 013703 000004
1306 003444 012737 003460 000004
1307 003452 005777 177306
1308 003456 000405
1309 003460 022626
1310 003462 004737 013304
(1) 003466 000001
1311 003470 000000
1312 003472 010337 000004
1313
1314
1315
1316
(3)
(3)
(2) 003476 000004
(2) 003500 012737 000002 001104
1317 003506 013703 000004
1318 003512 012737 003526 000004
1319 003520 005777 177242
1320 003524 000405
1321 003526 022626
1322 003530 004737 013304
(1) 003534 000002
1323 003536 000000
1324 003540 010337 000004
1325
1326
1327
(3)
(3)
(2) 003544 000004
(2) 003546 012737 000003 001104
1328 003554 032737 000001 001120
1329 003562 001405
1330 003564 005737 001106
1331 003570 001402
1332 003572 000137 004402
1333 003576 005077 177164
1334 003602 105777 177156
1335 003606 100011
1336
1337
1338 003610 005077 177152
1339 003614 105777 177144
1340 003620 100004
1341 003622 004737 013304
(1) 003626 000003
1342 003630 000000

*****
*TEST 1 TEST ABILITY TO ACCESS SLU1 TCSR
*****
TST1: SCOPE
MOV #1,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;;SAVE TIMEOUT VECTOR
MOV #1$,@#4 ;;SET UP TIMEOUT VECTOR
TST @CTCSR ;;REFERENCE THE XMIT COMMAND/STATUS REG.
BR 2$ ;;GO TO END OF TEST
1$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER TIMEOUT
JSR PC,$ATY4 ;;ONLY REPORT A FATAL ERROR
1 ;;THE ERROR NUMBER (FROM APT LIST)
HALT
2$: MOV R3,@#4 ;;RESTORE TIMEOUT VECTOR

*****
*TEST 2 TEST ABILITY TO ACCESS SLU1 TBUF
*****
TST2: SCOPE
MOV #2,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;;SAVE TIMEOUT VECTOR
MOV #1$,@#4 ;;SET UP TIMEOUT VECTOR
TST @CTBUF ;;REFERENCE THE XMIT BUFFER
BR 2$ ;;GO TO END OF TEST
1$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER TIMEOUT
JSR PC,$ATY4 ;;ONLY REPORT A FATAL ERROR
2 ;;THE ERROR NUMBER (FROM APT LIST)
HALT
2$: MOV R3,@#4 ;;RESTORE TIMEOUT VECTOR

*****
*TEST 3 TEST SLU1 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
*****
TST3: SCOPE
MOV #3,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#5ENV ;;ARE WE RUNNING UNDER APT
BEQ 70$ ;;IF NO THEN DO TEST
TST @#5PASS ;;IS THIS FIRST PASS
BEQ 70$ ;;IF YES THEN DO THIS SERIES OF TESTS
JMP SLU2RT ;;IF NO THEN SKIP THIS SERIES OF TESTS
70$: CLR @CTBUF LOAD XBUF
TSTB @CTCSR ;;CHECK DONE
BPL 1$ ;;BR IF CLEAR
;;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
;; FIRST TEST TO FAIL
CLR @CTBUF ;;FILL DOUBLE BUFFER
TSTB @CTCSR ;;CHECK DONE
BPL 1$ ;;BR IF CLEAR
JSR PC,$ATY4 ;;ONLY REPORT A FATAL ERROR
3 ;;THE ERROR NUMBER (FROM APT LIST)
HALT ;;TCSR 'DONE' NOT CLEARED WITH TBUF FULL
```

1343 003632 005000
1344 003634 105777 177124
1345 003640 100407
1346 003642 005200
1347 003644 001373
1348 003646 004737 013304
(1) 003652 000004
1349 003654 000000
1350 003656 000416
1351
1352
1353 003660 023737 000042 000046
1354 003666 001412
1355 003670 005737 001106
1356 003674 001007
1357 003676 005737 001110
1358 003702 001004
1359 003704 104401
1360 003706 025476
1361 003710 104401
1362 003712 025510
1363 003714
1364
1365
(3)
(3)
(2) 003714 000004
(2) 003716 012737 000004 001104
1366 003724 032737 000001 001156
(1) 003732 001002
(1) 003734 000137 004402
(1) 003740
1367 003740 005077 177022
1368 003744 105777 177014
1369 003750 100375
1370 003752 005077 177010
1371 003756 000240
1372 003760 000005
1373 003762 105777 176776
1374 003766 100401
1375
1376 003770 104005
1377
1378
1379
1380
(3)
(3)
(2) 003772 000004
(2) 003774 012737 000005 001104
1381 004002 013703 000004
1382 004006 012737 004022 000004
1383 004014 005777 176740
1384 004020 000402
1385
1386 004022 022626

1\$: CLR R0 ;CLEAR TIMER
2\$: TSTB @CTCSR ;CHECK FOR XMIT DONE
BMI ID ;IF DONE SETS, BR TO END OF TEST
INC R0 ;INCREMENT TIMER
BNE 2\$;BR IF TIMER NOT DONE
JSR PC,\$ATY4 ;:ONLY REPORT A FATAL ERROR
4 ;:THE ERROR NUMBER (FROM APT LIST)
HALT
BR TST4 ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
; BECAUSE OF THIS FAILURE
ID: CMP @#42,@#46 ;UNDER ACT11?
BEQ 6\$;IF YES, SKIP IDENT. TYPEOUT
TST \$PASS ;IS THIS THE FIRST PASS?
BNE 6\$;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
TST \$DEVCT ;IS THIS THE FIRST SUBPASS?
BNE 6\$;IF NOT, BR TO NEXT TEST
TYPE ;TYPE PROGRAM IDENTIFICATION
M1 ;TYPE NUMBER OF DEVICES UNDER TEST
TYPE M2
6\$:
:*****
:*TEST 4 TEST THAT SLU1 TCSR 'DONE' SETS WITH RESET
:*****
TST4: SCOPE
MOV #4,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT0,@#\$DEVM ;DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;IF YES CONTINUE WITH TESTS
JMP SLU2RT ;IF NO GO TO START OF NEXT SET OF TESTS.
99\$:
1\$: CLR @CTBUF ;LOAD TRANSMIT BUFFER
TSTB @CTCSR ;WAIT FOR DONE
BPL 1\$
CLR @CTBUF ;LOAD SECOND BUFFER
NOP
RESET ;CLEAR DONE WITH RESET
TSTB @CTCSR ;CHECK FOR DONE SET
BMI TST5 ;BR TO NEXT TEST IF DONE SET
ERROR 5 ;TCSR 'DONE' DOES NOT SET WITH RESET
:*****
:*TEST 5 TEST ABILITY TO ACCESS SLU1 RCSR
:*****
TST5: SCOPE
MOV #5,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;SET UP TIMEOUT VECTOR
TST @CRCSR ;ACCESS RCSR
BR 2\$;BR TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT

1387 004024 104006
 1388 004026 010337 000004
 1389
 1390
 1391
 (3)
 (3)
 (2) 004032 000004
 (2) 004034 012737 000006 001104
 1392 004042 013703 000004
 1393 004046 012737 004062 000004
 1394 004054 005777 176702
 1395 004060 000402
 1396
 1397 004062 022626
 1398 004064 104007
 1399 004066 010337 000004
 1400
 1401
 1402
 1403
 1404
 1405
 (3)
 (3)
 (2) 004072 000004
 (2) 004074 012737 000007 001104
 1406 004102 042777 000100 176654
 1407 004110 017703 176672
 1408 004114 012777 004144 176664
 1409 004122 004737 012356
 1410 004126 000340
 1411 004130 032777 000100 176626
 1412 004136 001404
 1413 004140 104014
 1414
 1415 004142 000402
 1416
 1417 004144 022626
 1418 004146 104015
 1419
 1420
 1421 004150 052777 000100 176606
 1422 004156 032777 000100 176600
 1423 004164 001001
 1424
 1425 004166 104016
 1426
 1427
 1428 004170 042777 000100 176566
 1429 004176 032777 000100 176560
 1430 004204 001401
 1431 004206 104017
 1432
 1433
 1434 004210

```

ERROR 6 ;CAN NOT ACCESS RCSR
2$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR

:*****
:*TEST 6 TEST ABILITY TO ACCESS SLU1 RBUF
:*****
TST6: SCOPE
MOV #6,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1$,@#4 ;SET UP TIMEOUT VECTOR
TST @CRBUF ;ACCESS RBUF
BR 2$ ;BR TO END OF TEST

1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ERROR 7 ;CAN NOT ACCESS RBUF
2$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR

:*****
:*TEST 7 TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET
:*****
TST7: SCOPE
MOV #7,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@CTCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV @CTVECT,R3 ;SAVE XMIT VECTOR
MOV #1$,@CTVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY=7
;WORD 340
BIT #BIT6,@CTCSR ;TEST BIT6 OF TCSR
BEQ 2$ ;BR IF ZERO
ERROR 14 ;BIT6 IN TCSR NOT CLEAR AFTER RESET
BR 2$

1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 15 ;XMIT INTERRUPT OCCURRED PRIO=7

2$: BIS #BIT6,@CTCSR ;SET BIT6 OF TCSR
BIT #BIT6,@CTCSR ;TEST BIT6 OF TCSR
BNE 3$ ;BR, IF SET

ERROR 16 ;CANNOT SET BIT6 OF TCSR

3$: BIC #BIT6,@CTCSR ;CLEAR BIT6 OF TCSR
BIT #BIT6,@CTCSR ;TEST BIT6 OF TCSR
BEQ 4$ ;BR IF CLEAR
ERROR 17 ;CANNOT CLEAR BIT6 OF TCSR

4$:

```

```
1435 004210 052777 000100 176546 BIS #BIT6,@CTCSR ;SET BIT6 OF TCSR
1436 004216 000005 RESET ;CLEAR BIT6 WITH RESET
1437 004220 032777 000100 176536 BIT #BIT6,@CTCSR ;TEST BIT6 OF TCSR
1438 004226 001401 BEQ 5$ ;BR IF CLEAR
1439
1440 004230 104020 ERROR 20
1441
1442 004232 010377 176550 5$: MOV R3,@CTVECT ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1443 ;RESTORE XMIT VECTOR
1444
1445 ::*****
(3) ;*TEST 10 TEST THAT SLU1 BIT6 OF RCSR CAN BE SET & RESET
(3) ;*****
(2) 004236 000004 TST10: SCOPF
(2) 004240 012737 000010 001104 MOV #10,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1446 004246 042777 000100 176504 BIC #BIT6,@CRCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1447 004254 017703 176522 MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
1448 004260 012777 004310 176514 MOV #1$,@CRVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1449 004266 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY=7
1450 004272 000340 .WORD 340
1451 004274 032777 000100 176456 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1452 004302 001404 BEQ 2$
1453 004304 104021 ERROR 21 ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1454
1455 004306 000402 BR 2$
1456
1457 004310 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1458 004312 104022 ERROR 22 ;RCVR INTERRUPT WITH PRIORITY=7
1459
1460
1461 004314 052777 000100 176436 2$: BIS #BIT6,@CRCSR ;SET BIT6 OF RCSR
1462 004322 032777 000100 176430 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1463 004330 001001 BNE 3$ ;BR, IF SET
1464
1465 004332 104023 ERROR 23 ;CANNOT SET BIT6 OF RCSR
1466
1467
1468 004334 042777 000100 176416 3$: BIC #BIT6,@CRCSR ;CLEAR BIT6 OF RCSR
1469 004342 032777 000100 176410 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1470 004350 001401 BEQ 4$ ;BR, IF CLEAR
1471
1472 004352 104024 ERROR 24 ;CANNOT CLEAR BIT6 OF RCSR
1473
1474
1475 004354 4$:
1476 004354 052777 000100 176376 BIS #BIT6,@CRCSR ;SET BIT6 OF RCSR
1477 004362 000005 RESET ;CLEAR BIT6 OF RCSR WITH RESET
1478 004364 032777 000100 176366 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1479 004372 001401 BEQ 5$ ;BR, IF CLEAR
1480
1481 004374 104025 ERROR 25 ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1482 5$: MOV R3,@CRVECT ;RESTORE RECEIVE VECTOR
1483
1484
1485
```

1487 004402
1488
1489
(3)
(3)
(2) 004402 000004
(2) 004404 012737 000011 001104
1490 004412 032737 000002 001156
(1) 004420 001002
(1) 004422 000137 005626
(1) 004426
1491 004426 013703 000004
1492 004432 012737 004446 000004
1493 004440 005777 176310
1494 004444 000402
1495 004446 022626
1496 004450 104031
1497 004452 010337 000004
1498
1499
1500
1501
(3)
(3)
(2) 004456 000004
(2) 004460 012737 000012 001104
1502 004466 013703 000004
1503 004472 012737 004506 000004
1504 004500 005777 176252
1505 004504 000402
1506 004506 022626
1507 004510 104032
1508 004512 010337 000004
1509
1510
1511
(3)
(3)
(2) 004516 000004
(2) 004520 012737 000013 001104
1512 004526 032737 000001 001120
(1) 004534 001403
(1) 004536 005737 001106
(2) 004542 001022
(1) 004544
1513 004544 005077 176206
1514 004550 105777 176200
1515 004554 100006
1516
1517
1518 004556 005077 176174
1519 004562 105777 176166
1520 004566 100001
1521 004570 104033
1522 004572 005000
1523 004574 105777 176154

SLU2RT:
:*****
:*TEST 11 TEST ABILITY TO ACCESS SLU2 TCSR
:*****
TST11: SCOPE
MOV #11,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #BIT1,@#\$DEVN ;;DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;F YES CONTINUE WITH TESTS
JMP LTCRT ;IF NO GO TO START OF NEXT SET OF TESTS.
99\$:
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;SET UP TIMEOUT VECTOR
TST @TCSR ;REFERENCE THE XMIT COMMAND/STATUS REG.
BR 2\$;GO TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ERROR 31 ;XMIT CSR ADDRESS DOES NOT RETURN SSYNC
2\$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR
:*****
:*TEST 12 TEST ABILITY TO ACCESS SLU2 TBUF
:*****
TST12: SCOPE
MOV #12,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;SET UP TIMEOUT VECTOR
TST @TBUF ;REFERENCE THE XMIT BUFFER
BR 2\$;GO TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ERROR 32 ;XMIT BUFFER ADDRESS DOES NOT RETURN SSYNC
2\$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR
:*****
:*TEST 13 TEST SLU2 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
:*****
TST13: SCOPE
MOV #13,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#\$ENV ;ARE WE RUNNING UNDER APT
BEQ 70\$;IF NO THEN DO TEST
TST @#\$PASS ;IS THIS FIRST PASS
BNE TST14 ;IF NO THEN SKIP TO NEXT TEST
70\$:
CLR @TBUF ;LOAD XBUF
TSTB @TCSR ;CHECK DONE
BPL 1\$;BR IF CLEAR
;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
; FIRST TEST TO FAIL
CLR @TBUF ;FILL DOUBLE BUFFER
TSTB @TCSR ;CHECK DONE
BPL 1\$;BR IF CLEAR
ERROR 33 ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
1\$: CLR R0 ;CLEAR TIMER
2\$: TSTB @TCSR ;CHECK FOR XMIT DONE

1524 004600 100473 BMI TST17 ;IF DONE SETS, BR TO NEXT TEST
1525 004602 005200 INC R0 ;INCREMENT TIMER
1526 004604 001373 BNE 2\$;BR IF TIMER NOT DONE
1527 004606 104034 ERROR 34 ;XMIT DONE BIT DOES NOT RESET AFTER TRANSMIT

1528
1529

1530 ;:*****
(3) ;*TEST 14 TEST THAT SLU2 TCSR 'DONE' SETS WITH RESET
(3) ;:*****

(2) 004610 000004 TST14: SCOPE
(2) 004612 012737 000014 001104 MOV #14,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1531 004620 032737 000001 001120 BIT #1,@\$ENV ;:ARE WE RUNNING UNDER APT
(1) 004626 001403 BEQ 70\$;:IF NO THEN DO TEST
(1) 004630 005737 001106 TST @\$PASS ;:IS THIS FIRST PASS
(2) 004634 001015 BNE TST15 ;:IF NO THEN SKIP TO NEXT TEST

(1) 004636 70\$: CLR @TBUF ;:LOAD TRANSMIT BUFFER
1532 004636 005077 176114 TSTB @TCSR ;:WAIT FOR DONE
1533 004642 105777 176106 BPL 1\$
1534 004646 100375 CLR @TBUF ;:LOAD SECOND BUFFER
1535 004650 005077 176102 NOP
1536 004654 000240 RESET ;:CLEAR DONE WITH RESET
1537 004656 000005 TSTB @TCSR ;:CHECK FOR DONE SET
1538 004660 105777 176070 BMI TST20 ;:BR TO NEXT TEST IF DONE SET
1539 004664 100520
1540
1541 004666 104035 ERROR 35 ;:TCSR 'DONE' DOES NOT SET WITH RESET

1542
1543
1544
1545

(3) ;:*****
(3) ;*TEST 15 TEST ABILITY TO ACCESS SLU2 RCSR
(3) ;:*****

(2) 004670 000004 TST15: SCOPE
(2) 004672 012737 000015 001104 MOV #15,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1546 004700 013703 000004 MOV @#4,R3 ;:SAVE TIMEOUT VECTOR
1547 004704 012737 004720 000004 MOV #1\$,@#4 ;:SET UP TIMEOUT VECTOR
1548 004712 005777 176032 TST @RCSR ;:ACCESS RCSR
1549 004716 000402 BR 2\$;:BR TO END OF TEST

1550
1551 004720 022626 1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER TIMEOUT
1552 004722 104036 ERROR 36 ;:CAN NOT ACCESS RCSR
1553 004724 010337 000004 2\$: MOV R3,@#4 ;:RESTORE TIMEOUT VECTOR

1554
1555
1556
1557

(3) ;:*****
(3) ;*TEST 16 TEST ABILITY TO ACCESS SLU2 RBUF
(3) ;:*****

(2) 004730 000004 TST16: SCOPE
(2) 004732 012737 000016 001104 MOV #16,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1558 004740 013703 000004 MOV @#4,R3 ;:SAVE TIMEOUT VECTOR
1559 004744 012737 004760 000004 MOV #1\$,@#4 ;:SET UP TIMEOUT VECTOR
1560 004752 005777 175774 TST @RBUF ;:ACCESS RBUF
1561 004756 000402 BR 2\$;:BR TO END OF TEST

1562
1563 004760 022626 1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER TIMEOUT

1564 004762 104037
1565 004764 010337 000004
1566
1567
1568
1569
(3)
(3)
(2) 004770 000004
(2) 004772 012737 000017 001104
1570 005000 042777 000001 175746
1571 005006 032777 000001 175740
1572 005014 001401
1573 005016 104040
1574 005020 052777 000001 175726 1\$:
1575 005026 032777 000001 175720
1576 005034 001001
1577 005036 104041
1578 005040 042777 000001 175706 2\$:
1579 005046 032777 000001 175700
1580 005054 001401
1581 005056 104042
1582 005060 3\$:
(1) 005060 032737 000001 001120
(1) 005066 001403
(1) 005070 005737 001106
(2) 005074 001014
(1) 005076 70\$:
1583 005076 052777 000001 175650
1584 005104 000005
1585 005106 032777 000001 175640
1586 005114 001404
1587 005116 042777 000001 175630
1588 005124 104043
1589
1590
1591
(3)
(3)
(2) 005126 000004
(2) 005130 012737 000020 001104
1592 005136 042777 000100 175610
1593 005144 017703 175626
1594 005150 012777 005200 175620
1595 005156 004737 012356
1596 005162 000340
1597 005164 032777 000100 175562
1598 005172 001404
1599 005174 104044
1600
1601 005176 000402
1602
1603 005200 022626 1\$:
1604 005202 104045
1605
1606

2\$: ERROR 37 :CAN NOT ACCESS RBUF
MOV R3,#4 :RESTORE TIMEOUT VECTOR

*TEST 17 TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET

TST17: SCOPE
MOV #17,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT0,@TCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
BIT #BIT0,@TCSR ;CHECK BIT0 OF TCSR CLEAR
BEQ 1\$;BR IF CLEAR
ERROR 40 ;BIT0 WAS NOT CLEAR AFTER RESET
1\$: BIS #BIT0,@TCSR ;SET BIT0 IN TCSR
BIT #BIT0,@TCSR ;TEST BIT0 OF TCSR
BNE 2\$;BR IF SET
ERROR 41 ;BIT0 OF TCSR WILL NOT SET
2\$: BIC #BIT0,@TCSR ;CLEAR BIT0 OF TCSR
BIT #BIT0,@TCSR ;TEST BIT0 OF TCSR
BEQ 3\$
ERROR 42 ;BIT0 OF TCSR WILL NOT CLEAR
3\$: BIT #1,@#SENV ;ARE WE RUNNING UNDER APT
BEQ 70\$;IF NO THEN DO TEST
TST @#SPASS ;IS THIS FIRST PASS
BNE TST20 ;IF NO THEN SKIP TO NEXT TEST
70\$: BIS #BIT0,@TCSR ;SET BIT0 IN TCSR
RESET ;CLEAR BIT0 WITH RESET
BIT #BIT0,@TCSR ;TEST BIT0 CLEAR
BEQ TST20 ;BR IF CLEAR
BIC #BIT0,@TCSR ;CLEAR BIT0, TO PRINT ERROR
ERROR 43 ;RESET DID NOT CLEAR BIT0 OF TCSR

*TEST 20 TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET

TST20: SCOPE
MOV #20,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@TCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV @TVECT,R3 ;SAVE XMIT VECTOR
MOV #1\$,@TVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY=7
;WORD 340
BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
BEQ 2\$;BR IF ZERO
ERROR 44 ;BIT6 IN TCSR NOT CLEAR AFTER RESET
BR 2\$
1\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 45 ;XMIT INTERRUPT OCCURRED PRIO=7


```

1607 005204 052777 000100 175542 2$: BIS #BIT6,@TCSR ;SET BIT6 OF TCSR
1608 005212 032777 000100 175534 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
1609 005220 001001 BNE 3$ ;BR, IF SET
1610
1611 005222 104046 ERROR 46 ;CANNOT SET BIT6 OF TCSR
1612
1613
1614 005224 042777 000100 175522 3$: BIC #BIT6,@TCSR ;CLEAR BIT6 OF TCSR
1615 005232 032777 000100 175514 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
1616 005240 001401 BEQ 4$ ;BR IF CLEAR
1617 005242 104047 ERROR 47 ;CANNOT CLEAR BIT6 OF TCSR
1618
1619
1620 005244 032737 000001 001120 4$: BIT #1,@$ENV ;ARE WE RUNNING UNDER APT
1621 005252 001403 BEQ 70$ ;IF NO THEN DO TEST
1622 005254 005737 001106 TST @$PASS ;IS THIS FIRST PASS
1623 005260 001011 BNE 5$ ;IF NO THEN SKIP TO END OF TEST
1624
1625 005262 052777 000100 175464 70$: BIS #BIT6,@TCSR ;SET BIT6 OF TCSR
1626 005270 000005 RESET ;CLEAR BIT6 WITH RESET
1627 005272 032777 000100 175454 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
1628 005300 001401 BEQ 5$ ;BR IF CLEAR
1629
1630 005302 104050 ERROR 50 ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1631
1632 005304 010377 175466 5$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
1633
1634
1635
(3)
(3)
(2) 005310 000004
(2) 005312 012737 000021 001104
1636 005320 042777 000100 175422
1637 005326 017703 175440
1638 005332 012777 005362 175432
1639 005340 004737 012356
1640 005344 000340
1641 005346 032777 000100 175374
1642 005354 001404
1643 005356 104051
1644
1645 005360 000402
1646
1647 005362 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1648 005364 104052 ERROR 52 ;RCVR INTERRUPT WITH PRIORITY=7
1649
1650
1651 005366 052777 000100 175354 2$: BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
1652 005374 032777 000100 175346 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1653 005402 001001 BNE 3$ ;BR, IF SET
1654
1655 005404 104053 ERROR 53 ;CANNOT SET BIT6 OF RCSR
1656
1657
1658 005406 042777 000100 175334 3$: BIC #BIT6,@RCSR ;CLEAR BIT6 OF RCSR
    
```

```

;*****
;*TEST 21 TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
;*****
    
```

```

TST21: SCOPE
MOV #21,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@RCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV @RVECT,R3 ;SAVE RECEIVE VECTOR
MOV #1$,@RVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY=7
    
```

```

.WORD 340
BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
BEQ 2$
ERROR 51 ;BIT6 OF RCSR NOT CLEAR AFTER RESET
BR 2$
    
```

```

    
```

```

    
```

```
1659 005414 032777 000100 175326 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1660 005422 001401 BEQ 4$ ;BR, IF CLEAR
1661
1662 005424 104054 ERROR 54
1663 ;CANNOT CLEAR BIT6 OF RCSR
1664
1665 005426 032737 000001 001120 4$: BIT #1,@$ENV ;ARE WE RUNNING UNDER APT
1666 005434 001403 BEQ 70$ ;IF NO THEN DO TEST
1667 005436 005737 001106 TST @$SPASS ;IS THIS FIRST PASS
1668 005442 001011 BNE 5$ ;IF NO THEN SKIP TO END OF TEST
1669 005444 70$:
1670 005444 052777 000100 175276 BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
1671 005452 000005 RESET ;CLEAR BIT6 OF RCSR WITH RESET
1672 005454 032777 000100 175266 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1673 005462 001401 BEQ 5$ ;BR, IF CLEAR
1674
1675 005464 104055 ERROR 55
1676 ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1677 005466 010377 175300 5$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
1678
1679
1680
1681 ;:*****
(3) ;*TEST 22 TEST THAT SLU2 RCVR DONE (7) SET & CLEAR PROPERLY
(3) ;:*****
(2) 005472 000004 TST22: SCOPE
(2) 005474 012737 000022 001104 MOV #22,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1682 005502 005077 175244 CLR @RBUF ;:INITIALIZE REGISTER BEFORE TESTING
1683 005506 005000 CLR R0 ;:CLEAR TIMER
1684 005510 005077 175242 CLR @TBUF ;:LOAD TRANSMIT BUFFER
1685 005514 105777 175230 WDONE: TSTB @RCSR ;:CHECK FOR RECEIVER DONE
1686 005520 100403 BMI 1$ ;:BR, IF DONE
1687 005522 005200 INC R0 ;:INCREMENT TIMER, IF NOT DONE
1688 005524 001373 BNE WDONE ;:CONTINUE WAIT IF TIME REMAINS
1689 005526 104056 ERROR 56
1690 ;:RECEIVER DONE NEVER SET
1691
1692 005530 032737 000001 001120 1$: BIT #1,@$ENV ;:ARE WE RUNNING UNDER APT
1693 005536 001403 BEQ 70$ ;:IF NO THEN DO TEST
1694 005540 005737 001106 TST @$SPASS ;:IS THIS FIRST PASS
1695 005544 001005 BNE 2$ ;:IF NO THEN SKIP TO END OF TEST
1696 005546 70$:
1697 005546 000005 RESET ;:CLEAR DONE WITH RESET
1698 005550 105777 175174 TSTB @RCSR ;:CHECK FOR DONE CLEAR
1699 005554 001401 BEQ 2$
1700
1701 005556 104057 ERROR 57
1702 ;:RESET DID NOT CLEAR RCVR DONE
1703 005560 005777 175166 2$: TST @RBUF ;:CLEAR RECEIVER BUFFER
1704
1705
1706
1707 ;:*****
(3) ;*TEST 23 TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
(3) ;:*****
(2) 005564 000004 TST23: SCOPE
```

(2) 005566 012737 000023 001104
1708 005574 005077 175152
1709 005600 005077 175152
1710 005604 105777 175140
1711 005610 100375
1712 005612 017700 175134
1713 005616 105777 175126
1714 005622 001401
1715 005624 104060
1716
1717 005626
1718
1719

MOV #23,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
CLR @RBUF ;:INITIALIZE REGISTER BEFORE TESTING
CLR @TBUF ;:LOAD TRANSMITTER
1\$: TSTB @RCR ;:WAIT FOR RECEIVER DONE
BPL 1\$
MOV @RBUF,R0 ;:READ RECEIVE BUFFER
TSTB @RCR ;:CHECK FOR RECEIVE DONE CLEAR
BEQ TST24 ;:BR, IF CLEAR TO NEXT TEST
ERROR 60 ;:READING RBUF DID NOT CLEAR RCVR DONE

LTCRT:

*:TEST 24 TEST ABILITY TO ACCESS LKS

(3)
(3)
(2) 005626 000004
(2) 005630 012737 000024 001104
1720 005636 032737 000004 001156
(1) 005644 001002
(1) 005646 000137 006100
(1) 005652
1721 005652 013703 000004
1722 005656 012737 005672 000004
1723 005664 005777 175100
1724 005670 000402
1725
1726 005672 022626
1727 005674 104061
1728
1729 005676 010337 000004
1730
1731

TST24: SCOPE
MOV #24,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT2,@#\$DEVM ;:DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;:F YES CONTINUE WITH TESTS
JMP UNIQUE ;:IF NO GO TO START OF NEXT SET OF TESTS.
99\$: MOV @#4,R3 ;:SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;:SET UP TIMEOUT VECTOR
TST @LKS ;:ACCESS LKS
BR 2\$;:NO TIMEOUT - BR TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER TIMEOUT
ERROR 61 ;:CAN NOT ACCESS LKS
2\$: MOV R3,@#4 ;:RESTORE TIMEOUT VECTOR

*:TEST 25 TEST THAT BIT6 OF LKS CAN BE SET & RESET

(3)
(3)
(2) 005702 000004
(2) 005704 012737 000025 001104
1732 005712 042777 000100 175050
1733 005720 017703 175066
1734 005724 012777 005754 175060
1735 005732 004737 012356
1736 005736 000340
1737 005740 032777 000100 175022
1738 005746 001404
1739 005750 104062
1740
1741 005752 000402
1742
1743 005754 022626
1744 005756 104063
1745
1746
1747 005760 052777 000100 175002
1748 005766 032777 000100 174774
1749 005774 001001
1750
1751 005776 104064

TST25: SCOPE
MOV #25,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@LKS ;:MAKE SURE BIT UNDER TEST IS INITIALIZED BEFORE TESTING
MOV @RTCVT,R3 ;:SAVE LINE CLOCK VECTOR
MOV #1\$,@RTCVT ;:SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR PC,WRPSW ;:SET PSW TO PRIORITY 7
WORD 340
BIT #BIT6,@LKS ;:TEST BIT6 OF LKS
BEQ 2\$
ERROR 62 ;:BIT6 OF LKS NOT CLEAR AFTER RESET
BR 2\$
1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
ERROR 63 ;:LKS INTERRUPT WITH PRIORITY=7
2\$: BIS #BIT6,@LKS ;:SET BIT6 OF LKS
BIT #BIT6,@LKS ;:TEST BIT6 OF LKS
BNE 3\$;:BR IF SET
ERROR 64

```
1752                                     :CANNOT SET BIT6 OF LKS
1753
1754 006000 042777 000100 174762 3$: BIC #BIT6,@LKS ;CLEAR BIT6 OF LKS
1755 006006 032777 000100 174754 BIT #BIT6,@LKS ;TEST BIT6 OF LK
1756 006014 001401 BEQ 4$
1757 006016 104065 ERROR 65
1758
1759 006020 032737 000001 001120 4$: BIT #1,@$ENV ;CANNOT CLEAR BIT6 OF LKS
1760 006026 001403 BEQ 70$ ;ARE WE RUNNING UNDER APT
1761 006030 005737 001106 TST @$SPASS ;IF NO THEN DO TEST
1762 006034 001011 BNE 5$ ;IS THIS FIRST PASS
1763 006036
1764 006036 052777 000100 174724 70$: BIS #BIT6,@LKS ;SET BIT6 OF LKS
1765 006044 000005 RESET ;CLEAR BIT6 OF LKS WITH RESET
1766 006046 032777 000100 174714 BIT #BIT6,@LKS ;TEST BIT6 OF LKS
1767 006054 001401 BEQ 5$ ;BR IF CLEAR
1768
1769 006056 104066 ERROR 66
1770
1771 006060 010377 174726 5$: MOV R3,@RTCVT ;CANNOT CLEAR BIT6 OF LKS WITH RESET
1772 006064 000405 BR UNIQUE ;RESTORE LINE CLOCK VECTOR
1773 ;GO TO NEXT TEST
1774
1775 006066 177560 DADTBL: .WORD 177560
1776 006070 177564 .WORD 177564
1777 006072 176500 .WORD 176500
1778 006074 176504 .WORD 176504
1779 006076 177564 TBLEND: .WORD 177564
1780
1781
1782 006100 UNIQUE:
1783
1784 ;:*****
(3) ;*TEST 26 UNIQUE INTERNAL ADDRESS TEST
(3) ;:*****
(2) 006100 000004 TST26: SCOPE
(2) 006102 012737 000026 001104 MOV #26,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1785 006110 032737 000001 001120 BIT #1,@$ENV ;ARE WE RUNNING UNDER APT
(1) 006116 001403 BEQ 70$ ;IF NO THEN DO TEST
(1) 006120 005737 001106 TST @$SPASS ;IS THIS FIRST PASS
(2) 006124 001056 BNE TST27 ;IF NO THEN SKIP TO NEXT TEST
(1) 006126
1786 006126 012737 000340 177776 70$: MOV #340,PS ;WE WILL BE PLAYING WITH BIT6
1787 ;SO LOCK OUT EXTRAINEOUS INTERRUPTS
1788 006134 012700 006066 MOV #DADTBL,R0 ;GET LOCATION OF FIRST REGISTER ADDRESS
1789 006140 012703 006066 1$: MOV #DADTBL,R3 ;MAKE R3 POINT TO LOCATION OF FIRST
1790 ;REGISTER ADDRESS
1791 006144 012701 000005 MOV #5,R1 ;SET LOOP COUNTER TO CLEAR ALL REG.
1792 006150 005033 2$: CLR @(R3)+ ;CLEAR A REGISTER
1793 006152 077102 SOB R1,2$ ;LOOP UNTIL ALL REGISTERS CLEARED
1794 006154 012770 000100 000000 MOV #BIT6,@(R0) ;SET TEST BIT IN DEVICE REGISTERS
1795 006162 012701 006066 MOV #DADTBL,R1 ;GET LOCATION OF FIRST REGISTER ADDRESS
1796 006166 012702 000005 MOV #5,R2 ;SET UP TEST LOOP COUNTER
1797 006172 032731 000100 3$: BIT #BIT6,@(R1)+ ;IS TEST BIT SET IN THIS REGISTER
1798 006176 001006 BNE 5$ ;IF YES GO SEE IF THERE IS AN ERROR
1799 006200 077204 4$: SOB R2,3$ ;LOUP UNTIL ALL REGISTER CHECKED
```

CJKDIAO 11/238 SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82 16:42 PAGE 9-7
T26 UNIQUE INTERNAL ADDRESS TEST

SEQ 0045

1800	006202	005030			CLR	@(R0)+		:CLEAR REGISTER JUST TESTED AND POINT :TO NEXT ONE
1801								
1802	006204	020027	006076		CMP	R0,#TBLEND		
1803	006210	001421			BEQ	7\$		
1804	006212	000752			BR	1\$:CONTINUE TESTING
1805	006214	021041		5\$:	CMP	(R0),-(R1)		:DID WE COMPARE THE REGISTER TO ITSELF?
1806	006216	001413			BEQ	6\$:IF YES GET OVER ERROR CALL
1807	006220	011037	001020		MOV	(R0),\$GDADR		:IF NO SET UP ERROR INFORMATION
1808	006224	011137	001022		MOV	(R1),\$BDADR		
1809	006230	017037	000000	001024	MOV	@(R0),\$GDDAT		
1810	006236	017137	000000	001026	MOV	@(R1),\$BDDAT		
1811	006244	104072			ERROR	72		:WRITE TO 1 INTERNAL ADDRESS MODIFIED
1812								:ANOTHER SO ADDRESS NOT UNIQUE
1813	006246	062701	000002	6\$:	ADD	#2,R1		:RESTORE POINTER
1814	006252	000752			BR	4\$:GET BACK IN TEST LOOP
1815	006254	005000		7\$:	CLR	R0		
1816	006256	005200		8\$:	INC	R0		
1817	006260	001376			BNE	8\$		
1818								

1820 006262
1821
1822
(3)
(3)
(2) 006262 000004
(2) 006264 012737 000027 001104
1823 006272 032737 000001 001120
1824 006300 001405
1825 006302 005737 001106
1826 006306 001402
1827 006310 000137 007016
1828 006314
(1) 006314 032737 000001 001156
(1) 006322 001002
(1) 006324 000137 007016
(1) 006330
1829 006330 000005
1830 006332 042777 000100 174424
1831 006340 017703 174442
1832 006344 012777 006370 174434
1833 006352 105777 174406
1834 006356 100375
1835 006360 004737 012356
1836 006364 000140
1837 006366 000402
1838
1839 006370 022626
1840 006372 104074
1841
1842 006374 012777 006414 174404
1843 006402 052777 000100 174354
1844 006410 000240
1845
1846 006412 104075
1847
1848 006414 042777 000100 174342
1849 006422 022626
1850 006424 010377 174356
1851
1852
1853
(3)
(3)
(2) 006430 000004
(2) 006432 012737 000030 001104
1854 006440 042777 000100 174316
1855 006446 004737 012356
1856 006452 000340
1857 006454 017703 174326
1858 006460 012777 006506 174320
1859 006466 105777 174272
1860 006472 100375
1861 006474 052777 000100 174262
1862 006502 000240
1863 006504 000402

SLU1IT:
:*****
:*TEST 27 TEST THAT SLU1 XMIT INTERRUPTS ONLY WHEN ENABLED
:*****
TST27: SCOPE
MOV #27,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#\$ENV ;IF NOT UNDER APT
BEQ 70\$; THEN RUN THIS SERIES OF TESTS
TST @#\$PASS ; ELSE IF FIRST PASS
BEQ 70\$; THEN RUN THESE TESTS
JMP SLU2IT ; ELSE DO NOT RUN THESE TESTS
70\$:
BIT #BIT0,@#\$DEV ;DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;F YES CONTINUE WITH TESTS
JMP SLU2IT ;IF NO GO TO START OF NEXT SET OF TESTS.
99\$:
RESET ;CLEAR THE WORLD
BIC #BIT6,@CTCSR ;CLEAR TRANSMIT INTERRUPT ENABLE
MOV @CTVECT,R3 ;SAVE XMIT VECTOR
MOV #2\$,@CTVECT ;POINT XMIT VECTOR TO ERROR REPORT
1\$: TSTB @CTCSR ;WAIT FOR DONE
BPL 1\$
JSR PC,WRPSW ;SET PSW TO PRIORITY 3
WORD 140
BR 3\$
2\$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 74
3\$: MOV #4\$,@CTVECT ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
BIS #BIT6,@CTCSR ;SET XMIT VECTOR TO END OF TEST
NOP ;ENABLE INTERRUPTS
ERROR 75 ;XMIT DID NOT INTERRUPT
4\$: BIC #BIT6,@CTCSR ;DISABLE INTERRUPTS
CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
MOV R3,@CTVECT ;RESTORE XMIT VECTOR
:*****
:*TEST 30 TEST SLU1 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
:*****
TST30: SCOPE
MOV #30,\$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@CTCSR ;DISABLE INTERRUPTS
JSR PC,WRPSW ;SET PSW TO PRIORITY 7
WORD 340
MOV @CTVECT,R3 ;SAVE XMIT VECTOR
MOV #2\$,@CTVECT ;POINT XMIT VECTOR TO ERROR REPORT
1\$: TSTB @CTCSR ;WAIT FOR DONE
BPL 1\$
BIS #BIT6,@CTCSR ;ENABLE INTERRUPT
NOP
BR 3\$;CONTINUE TEST

```
1864
1865 006506 022626      2$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
1866 006510 104076      ERROR      76
1867
1868 006512 042777 000100 174244 3$:  BIC      #BIT6,@CTCSR      ;XMIT INTERRUPTS AT PRIORITY=7
1869 006520 012777 006540 174260      MOV      #4$,@CTVECT      ;CLEAR INTERRUPT ENABLE
1870 006526 004737 012356      JSR      PC,WRPSW      ;POINT XMIT VECTOR TO ERROR REPORT
1871 006532 000140      .WORD    140      ;SET PSW TO PRIORITY 3
1872 006534 000240      NOP
1873 006536 000402      BR       5$      ;BR TO END OF TEST-NO INTERRUPT
1874
1875 006540 022626      4$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
1876 006542 104077      ERROR      77
1877
1878 006544 010377 174236      5$:  MOV      R3,@CTVECT      ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
1879
1880
1881 ;:*****
1882 (3) ;*TEST 31 TEST SLU1 TRANSMITTER FOR DOUBLE INTERRUPTS
1883 (3) ;:*****
1884 (2) 006550 000004      TST31: SCOPE
1885 (2) 006552 012737 000031 001104      MOV      #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1886 006560 042777 000100 174176      BIC      #BIT6,@CTCSR      ;CLEAR INTERRUPT ENABLE
1887 006566 017703 174214      MOV      @CTVECT,R3      ;SAVE XMIT VECTOR
1888 006572 017704 174212      MOV      @CTPSW,R4      ;SAVE XMIT PSW VECTOR
1889 006576 012777 006640 174202      MOV      #2$,@CTVECT      ;SET UP XMIT VECTOR
1890 006604 012777 000340 174176      MOV      #340,@CTPSW      ;SET PIO 7 AFTER INTERRUPT
1891 006612 004737 012356      JSR      PC,WRPSW      ;SET PSW TO PRIORITY 3
1892 006616 000140      .WORD    140
1893 006620 105777 174140      1$:  TSTB     @CTCSR      ;WAIT FOR DONE
1894 006624 100375      BPL      1$
1895 006626 052777 000100 174130      BIS      #BIT6,@CTCSR      ;ENABLE INTERRUPTS
1896 006634 000240      NOP
1897
1898 006636 104100      ERROR    100
1899
1900 006640 022626      2$:  CMP      (SP)+,(SP)+      ;XMIT INTERRUPT DID NOT OCCUR
1901 006642 012777 006670 174136      MOV      #4$,@CTVECT      ;RESTORE SP AFTER INTERRUPT
1902 006650 004737 012356      JSR      PC,WRPSW      ;POINT XMIT VECTOR TO ERROR
1903 006654 000140      .WORD    140      ;SET PSW TO PRIORITY 3
1904 006656 000240      NOP      ;GIVE TIME FOR ANY INTERRUPTS
1905 006660 042777 000100 174076      BIC      #BIT6,@CTCSR      ;DISABLE INTERRUPTS
1906 006666 000402      BR       5$      ;BR TO END OF TEST
1907
1908 006670 022626      4$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
1909 006672 104101      ERROR    101
1910
1911 006674 010377 174106      5$:  MOV      R3,@CTVECT      ;XMIT RE-INTERRUPTED
1912 006700 010477 174104      MOV      R4,@CTPSW      ;RESTORE XMIT VECTOR
1913
1914 ;:*****
1915 (3) ;*TEST 32 TEST THAT SLU1 XMIT INTERRUPT CLEARS WITH LOADING TBUF
1916 (3) ;:*****
1917 (2) 006704 000004      TST32: SCOPE
1918 (2) 006706 012737 000032 001104      MOV      #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1919 006714 042777 000100 174042      BIC      #BIT6,@CTCSR      ;DISABLE INTERRUPTS
```

```
1912 006722 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 7
1913 006726 000340 .WORD 340
1914 006730 017703 174052 MOV @TVECT,R3 ;SAVE XMIT VECTOR
1915 006734 012777 007006 174044 MOV #2,@TVECT ;POINT XMIT VECTOR TO ERROR
1916 006742 052777 000100 174014 BIS #BIT6,@TCSR ;ENABLE INTERRUPTS
1917 006750 005077 174012 CLR @TBUF ;LOAD TBUF
1918 006754 105777 174004 1$: TSTB @TCSR ;WAIT FOR DONE (INTERRUPT)
1919 006760 100375 BPL 1$
1920 006762 005077 174000 CLR @TBUF ;FILL SECOND BUFFER TO RESET INT.
1921 006766 004737 012356 JSR PC,WRPSW ;ALLOW INTERRUPTS
1922 006772 000140 .WORD 140
1923 006774 000240 NOP ;GIVE TIME FOR ANY INTERRUPTS
1924 006776 042777 000100 173760 BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
1925 007004 000402 BR 3$ ;BR TO END OF TEST
1926
1927 007006 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1928 007010 104102 ERROR 102
1929
1930 007012 010377 173770 3$: MOV R3,@TVECT ;LOADING TBUF DID NOT CLEAR INTERRUPT.
1931 ;RESTORE XMIT VECTOR
1932
1933 007016 SLU2IT:
1934
1935 ;:*****
(3) ;*TEST 33 TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED
(3) ;:*****
(2) 007016 000004 TST33: SCOPE
(2) 007020 012737 000033 001104 MOV #33,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1936 007026 032737 000002 001156 BIT #BIT1,@#$DEVM ;DO THESE TESTS FOR THIS DEVICE?
(1) 007034 001002 BNE 99$ ;F YES CONTINUE WITH TESTS
(1) 007036 000137 010762 JMP LTCIT ;IF NO GO TO START OF NEXT SET OF TESTS.
(1) 007042
1937 007042 042777 000100 173704 99$: BIC #BIT6,@TCSR ;CLEAR TRANSMIT INTERRUPT ENABLE
1938 007050 017703 173722 MOV @TVECT,R3 ;SAVE XMIT VECTOR
1939 007054 012777 007100 173714 MOV #2,@TVECT ;POINT XMIT VECTOR TO ERROR REPORT
1940 007062 105777 173666 1$: TSTB @TCSR ;WAIT FOR DONE
1941 007066 100375 BPL 1$
1942 007070 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
1943 007074 000140 .WORD 140
1944 007076 000402 BR 3$
1945
1946 007100 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1947 007102 104116 ERROR 116
1948
1949 007104 012777 007124 173664 3$: MOV #4,@TVECT ;XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
1950 007112 052777 000100 173634 BIS #BIT6,@TCSR ;SET XMIT VECTOR TO END OF TEST
1951 007120 000240 NOP ;ENABLE INTERRUPTS
1952
1953 007122 104117 ERROR 117 ;XMIT DID NOT INTERRUPT
1954
1955 007124 042777 000100 173622 4$: BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
1956 007132 022626 CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1957 007134 010377 173636 MOV R3,@TVECT ;RESTORE XMIT VECTOR
1958
1959
1960 ;:*****
```



```
(3) ;*TEST 34 TEST SLU2 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
(3) ;*****
(2) 007140 000004 TST34: SCOPE
(2) 007142 012737 000034 001104 MOV #34,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1961 007150 042777 000100 173576 BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
1962 007156 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 7
1963 007162 000340 .WORD 340
1964 007164 017703 173606 MOV @TVECT,R3 ;SAVE XMIT VECTOR
1965 007170 012777 007216 173600 MOV #2$,@TVECT ;POINT XMIT VECTOR TO ERROR REPORT
1966 007176 105777 173552 1$: TSTB @TCSR ;WAIT FOR DONE
1967 007202 100375 BPL 1$
1968 007204 052777 000100 173542 BIS #BIT6,@TCSR ;ENABLE INTERRUPT
1969 007212 000240 NOP
1970 007214 000402 BR 3$ ;CONTINUE TEST
1971
1972 007216 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1973 007220 104120 ERROR 120
1974 ;XMIT INTERRUPTS AT PRIORITY=7
1975 007222 042777 000100 173524 3$: BIC #BIT6,@TCSR ;CLEAR INTERRUPT ENABLE
1976 007230 012777 007250 173540 MOV #4$,@TVECT ;POINT XMIT VECTOR TO ERROR REPORT
1977 007236 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
1978 007242 000140 .WORD 140
1979 007244 000240 NOP
1980 007246 000402 BR 5$ ;BR TO END OF TEST-NO INTERRUPT
1981
1982 007250 022626 4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1983 007252 104121 ERROR 121
1984 ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
1985 007254 010377 173516 5$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
1986
1987
1988
1989
1990
```

```
;;*****
(3) ;*TEST 35 TEST SLU2 TRANSMITTER FOR DOUBLE INTERRUPTS
(3) ;*****
(2) 007260 000004 TST35: SCOPE
(2) 007262 012737 000035 001104 MOV #35,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1991 007270 042777 000100 173456 BIC #BIT6,@TCSR ;CLEAR INTERRUPT ENABLE
1992 007276 017703 173474 MOV @TVECT,R3 ;SAVE XMIT VECTOR
1993 007302 017704 173472 MOV @TPSW,R4 ;SAVE XMIT PSW VECTOR
1994 007306 012777 007350 173462 MOV #2$,@TVECT ;SET UP XMIT VECTOR
1995 007314 012777 000340 173456 MOV #340,@TPSW ;SET PIO 7 AFTER INTERRUPT
1996 007322 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
1997 007326 000140 .WORD 140
1998 007330 105777 173420 1$: TSTB @TCSR ;WAIT FOR DONE
1999 007334 100375 BPL 1$
2000 007336 052777 000100 173410 BIS #BIT6,@TCSR ;ENABLE INTERRUPTS
2001 007344 000240 NOP
2002
2003 007346 104122 ERROR 122
2004 ;XMIT INTERRUPT DID NOT OCCUR
2005 007350 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2006 007352 012777 007400 173416 MOV #4$,@TVECT ;POINT XMIT VECTOR TO ERROR
2007 007360 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
2008 007364 000140 .WORD 140
```

```
2009 007366 000240      NOP      :GIVE TIME FOR ANY INTERRUPTS
2010 007370 042777 000100 173356  BIC      #BIT6,@TCSR  :DISABLE INTERRUPTS
2011 007376 000402      BR       5$          :BR TO END OF TEST
2012
2013 007400 022626 4$:      CMP      (SP)+,(SP)+  :RESTORE SP AFTER INTERRUPT
2014 007402 104123      ERROR   123
2015
2016 007404 010377 173366 5$:      MOV      R3,@TVECT   :XMIT RE-INTERRUPTED
2017 007410 010477 173364      MOV      R4,@TPSW    :RESTORE XMIT VECTOR
                          :RESTORE XMIT PSW VECTOR
2018
2019
(3)
(3)
(2) 007414 000004      TST36:  SCOPE
(2) 007416 012737 000036 001104  MOV      #36,$TESTN   ;;SET TEST NUMBER IN APT MAIL BOX
2020 007424 032737 000001 001120  BIT      #1,@#SENV    :ARE WE RUNNING UNDER APT
(1) 007432 001403      BEQ      70$         :IF NO THEN DO TEST
(1) 007434 005737 001106      TST      @#$PASS     :IS THIS FIRST PASS
(2) 007440 001046      BNE      TST37       :IF NO THEN SKIP TO NEXT TEST
(1) 007442
2021 007442 042777 000100 173304  BIC      #BIT6,@TCSR  :DISABLE INTERRUPTS
2022 007450 004737 012356      JSR      PC,WRPSW    :SET PSW TO PRIORITY 7
2023 007454 000340      .WORD   340
2024 007456 017703 173314      MOV      @TVECT,R3   :SAVE XMIT VECTOR
2025 007462 012777 007534 173306  MOV      #2$,@TVECT  :POINT XMIT VECTOR TO ERROR
2026 007470 052777 000100 173256  BIS      #BIT6,@TCSR :ENABLE INTERRUPTS
2027 007476 005077 173254      CLR      @TBUF       :LOAD TBUF
2028 007502 105777 173246 1$:      TSTB    @TCSR       :WAIT FOR DONE (INTERRUPT)
2029 007506 100375      BPL      1$
2030 007510 005077 173242      CLR      @TBUF       :FILL SECOND BUFFER TO RESET INT.
2031 007514 004737 012356      JSR      PC,WRPSW    :ALLOW INTERRUPTS
2032 007520 000140      .WORD   140
2033 007522 000240      NOP
2034 007524 042777 000100 173222  BIC      #BIT6,@TCSR :GIVE TIME FOR ANY INTERRUPTS
2035 007532 000402      BR       3$          :DISABLE INTERRUPTS
                          :BR TO END OF TEST
2036
2037 007534 022626 2$:      CMP      (SP)+,(SP)+  :RESTORE SP AFTER INTERRUPT
2038 007536 104124      ERROR   124
2039
2040 007540 005001      3$:      CLR      R1          :LOADING TBUF DID NOT CLEAR INTERRUPT.
2041 007542 005201      4$:      INC      R1          :INITIALIZE LOOP COUNTER
2042 007544 001376      BNE      4$          :INCREMENT LOOP COUNTER
2043 007546 005777 173200      TST      @RBUF       :UNTIL LOOP COUNTER EQUALS 0
2044 007552 010377 173220      MOV      R3,@TVECT  :CLEAR RECEIVER BUFFER
                          :RESTORE XMIT VECTOR
2045
2046
2047
2048
2049
(3)
(3)
(2) 007556 000004      TST37:  SCOPE
(2) 007560 012737 000037 001104  MOV      #37,$TESTN   ;;SET TEST NUMBER IN APT MAIL BOX
2050 007566 042777 000100 173160  BIC      #BIT6,@TCSR  :DISABLE TRANSMIT INTERRUPTS
2051 007574 042777 000100 173146  BIC      #BIT6,@RCSR  :DISABLE RECEIVER INTERRUPTS
2052 007602 017703 173164      MOV      @RVECT,R3   :SAVE RECEIVE VECTOR
```

```
2053 007606 012777 007636 173156      MOV    #2$,@RVECT      ;POINT RCV VECTOR TO ERROR REPORT
2054 007614 004737 012356              JSR    PC,WRPSW        ;SET PSW TO PRIORITY 3
2055 007620 000140                      .WORD 140
2056 007622 005077 173130              CLR    @TBUF           ;SEND A CHARACTER
2057 007626 105777 173116      1$:   TSTB  @RCSR         ;WAIT FOR RECEIVER DONE
2058 007632 100375                      BPL    1$
2059 007634 000402                      BR     3$              ;CONTINUE TEST
2060
2061 007636 022626      2$:   CMP    (SP)+,(SP)+   ;RESTORE SP AFTER INTERRUPT
2062 007640 104125                      ERROR  125
2063
2064
2065 007642 012777 007662 173122      3$:   MOV    #4$,@RVECT   ;POINT RCV VECTOR TO END OF TEST
2066 007650 052777 000100 173072      BIS    #BIT6,@RCSR    ;ENABLE RCV INTERRUPTS
2067 007656 000240                      NOP
2068 007660 104126                      ERROR  126            ;GIVE ANY INTERRUPTS TIME
2069
2070
2071 007662 042777 000100 173060      4$:   BIC    #BIT6,@RCSR  ;DISABLE INTERRUPTS
2072 007670 022626                      CMP    (SP)+,(SP)+   ;RESTORE SP AFTER INTERRUPT
2073 007672 010377 173074                      MOV    R3,@RVECT     ;RESTORE RECEIVE VECTOR
2074
2075
2076
2076 (3)
2076 (3)
2076 (2) 007676 000004
2076 (2) 007700 012737 000040 001104      TST40: SCOPE
2077 007706 004737 012356              MOV    #40,$TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
2078 007712 000340                      JSR    PC,WRPSW      ;SET PSW TO PRIORITY 7
2079 007714 017703 173052                      .WORD 340
2080 007720 012777 007752 173044      MOV    @RVECT,R3     ;SAVE RECEIVE VECTOR
2081 007726 005077 173024      MOV    #2$,@RVECT   ;POINT RCVR VECTOR TO ERROR REPORT
2082 007732 105777 173012      1$:   CLR    @TBUF       ;SEND A CHARACTER
2083 007736 100375                      TSTB  @RCSR         ;WAIT FOR RECEIVER DONE
2084 007740 052777 000100 173002      BPL    1$
2085 007746 000240                      BIS    #BIT6,@RCSR  ;ENABLE INTERRUPTS
2086 007750 000402                      NOP                ;GIVE TIME FOR INTERRUPT
2087 007752 022626      2$:   BR     3$          ;CONTINUE TEST
2088 007754 104127                      CMP    (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
2089
2090
2091 007756 042777 000100 172764      3$:   ERROR  127        ;RCVR INTERRUPTS AT PRIORITY 7
2092 007764 012777 010004 173000      BIC    #BIT6,@RCSR  ;CLEAR INTERRUPT ENABLE
2093 007772 004737 012356      MOV    #4$,@RVECT   ;POINT RCVR VECTOR TO ERROR REPORT
2094 007776 000140                      JSR    PC,WRPSW     ;SET PSW TO PRIORITY 3
2095 010000 000240                      .WORD 140
2096 010002 000402                      NOP                ;GIVE TIME FOR ANY INTERRUPT
2097
2098 010004 022626      4$:   BR     5$          ;BR TO END OF TEST, IF NO INTERRUPT
2099 010006 104130                      CMP    (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
2100
2101 010010 010377 172756      5$:   ERROR  130
2102
2103
2104
```

```
(3) ;*TEST 41 TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
(3) ;:*****
(2) 010014 000004 TST41: SCOPE
(2) 010016 012737 000041 001104 MOV #41,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2105 010024 017703 172742 MOV @RVECT,R3 ;;SAVE RECEIVE VECTOR
2106 010030 017704 172740 MOV @RPSW,R4 ;;SAVE RECEIVE PSW VECTOR
2107 010034 012777 010102 172730 MOV #2$,@RVECT ;;POINT RCV VECTOR TO CONTINUE TEST
2108 010042 012777 000340 172724 MOV #340,@RPSW ;;SET PRIORITY TO 7 AFTER INTERRUPT
2109 010050 004737 012356 JSR PC,WRPSW ;;SET PSW TO PRIORITY 3
2110 010054 000140 .WORD 140
2111 010056 005077 172674 CLR @TBUF ;;SEND A CHARACTER
2112 010062 105777 172662 1$: TSTB @RCSR ;;WAIT FOR RCVR DONE
2113 010066 100375 BPL 1$
2114 010070 052777 000100 172652 BIS #BIT6,@RCSR ;;ENABLE RCV INTERRUPTS
2115 010076 000240 NOP ;;GIVE SOME TIME
2116
2117 010100 104131 ERROR 131
2118
2119 ;;RCVR INTERRUPT DID NOT OCCUR
2120 010102 022626 2$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER INTERRUPT
2121 010104 012777 010136 172660 MOV #3$,@RVECT ;;POINT RCV VECTOR TO ERROR REPORT
2122 010112 004737 012356 JSR PC,WRPSW ;;RESET PSW TO PRIORITY 3
2123 010116 000140 .WORD 140
2124 010120 000240 NOP ;;GIVE SOME TIME
2125 010122 042777 000100 172620 BIC #BIT6,@RCSR ;;CLEAR INTERRUPT ENABLE
2126 010130 010477 172640 MOV R4,@RPSW ;;RESTORE RECEIVE PSW VECTOR
2127 010134 000402 BR 4$ ;;BR TO END OF TEST
2128
2129 010136 022626 3$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER INTERRUPT
2130 010140 104132 ERROR 132
2131
2132 010142 010377 172624 4$: MOV R3,@RVECT ;;RECEIVER RE-INTERRUPTED
2133 ;;RESTORE RECEIVE VECTOR
2134
2135 ;:*****
(3) ;*TEST 42 TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
(3) ;:*****
(2) 010146 000004 TST42: SCOPE
(2) 010150 012737 000042 001104 MOV #42,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
2136 010156 004737 012356 JSR PC,WRPSW ;;SET PSW PRIORITY TO 7
2137 010162 000340 .WORD 340
2138 010164 017703 172602 MOV @RVECT,R3 ;;SAVE RECEIVE VECTOR
2139 010170 012777 010242 172574 MOV #2$,@RVECT ;;POINT RCV VECTOR TO ERROR REPORT
2140 010176 052777 000100 172544 BIS #BIT6,@RCSR ;;SET RCVR INTERRUPT ENABLE
2141 010204 005077 172546 CLR @TBUF ;;SEND A CHARACTER
2142 010210 105777 172534 1$: TSTB @RCSR ;;WAIT FOR DONE (INTERRUPT)
2143 010214 100375 BPL 1$
2144 010216 005777 172530 TST @RBUF ;;READ RBUF TO CLEAR PENDING INTERRUPT
2145 010222 004737 012356 JSR PC,WRPSW ;;SET PSW TO PRIORITY 3
2146 010226 000140 .WORD 140
2147 010230 000240 NOP ;;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2148 010232 042777 000100 172510 BIC #BIT6,@RCSR ;;NO INTERRUPT-CLEAR INT. ENABLE
2149 010240 000402 BR 3$
2150
2151 010242 022626 2$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER INTERRUPT
2152 010244 104133 ERROR 133 ;;READING RBUF DID NOT CLEAR INTERRUPT
```

```

2153
2154 010246 010377 172520 3$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
2155
2156
2157
2158
(3)
(3)
(2) 010252 000004
(2) 010254 012737 000043 001104
2159 010262 032737 000001 001120
(1) 010270 001403
(1) 010272 005737 001106
(2) 010276 001037
(1) 010300
2160 010300 000005
2161 010302 004737 012356
2162 010306 000340
2163 010310 017703 172456
2164 010314 012777 010366 172450
2165 010322 052777 000100 172420
2166 010330 012777 000377 172420
2167 010336 105777 172406
2168 010342 100375
2169 010344 000005
2170 010346 004737 012356
2171 010352 000140
2172 010354 000240
2173 010356 042777 000100 172364
2174 010364 000402
2175
2176
2177 010366 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2178 010370 104134 ERROR 134 ;RESET DID NOT CLEAR RCVR INTERRUPT
2179
2180 010372 010377 172374 3$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
2181
2182
2183
(3)
(3)
(2) 010376 000004
(2) 010400 012737 000044 001104
2184 010406 032777 002000 170424
2185 010414 001023
2186 010416 012700 000003
2187 010422 005077 172330
2188 010426 105777 172322
2189 010432 100375
2190 010434 005300
2191 010436 001371
2192 010440 032777 040000 172304
2193 010446 001001
2194 010450 104135
2195
2196

::*****
:*TEST 43 TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
::*****
TST43: SCOPE
MOV #43,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #1,@$SENV ;:ARE WE RUNNING UNDER APT
BEQ 70$ ;:IF NO THEN DO TEST
TST @$$PASS ;:IS THIS FIRST PASS
BNE TST44 ;:IF NO THEN SKIP TO NEXT TEST

70$:
RESET ;:CLEAR EVERYTHING
JSR PC,WRPSW ;:SET PSW TO PRIORITY 7
.WORD 340
MOV @RVECT,R3 ;:SAVE RECEIVE VECTOR
MOV #2$,@RVECT ;:POINT RCV VECTOR TO ERROR REPORT
BIS #BIT6,@RCSR ;:SET RCV INTERRUPT ENABLE
MOV #377,@TBUF ;:SEND AN ALL 1'S CHARACTER
1$: TSTB @RCSR ;:WAIT FOR RCV DONE
BPL 1$
RESET ;:CLEAR RCV INTERRUPT & RBUF
JSR PC,WRPSW ;:SET PSW TO PRIORITY 3
.WORD 140
NOP
BIC #BIT6,@RCSR ;:ALLOW TIME FOR AN ERRONEOUS INTERRUPT
BR 3$ ;:NO INTERRUPT-CLEAR INT. ENABLE
;:CONTINUE TEST

2$:
CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
ERROR 134 ;:RESET DID NOT CLEAR RCVR INTERRUPT

3$:
MOV R3,@RVECT ;:RESTORE RECEIVE VECTOR

::*****
:*TEST 44 TEST SLU2 THAT 'OVERRUN & ERROR' BITS CAN BE SET
::*****
TST44: SCOPE
MOV #44,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT10,@$SWR ;:IS THIS TEST DISABLED
BNE TST45 ;:IF NOT ENABLED, BR TO NEXT TEST
MOV #3,R0 ;:SET CHARACTER COUNT TO SEND 3 CHAR.
1$: CLR @TBUF ;:LOAD TRANSMIT BUFFER
2$: TSTB @TCSR ;:WAIT FOR TRANSMIT DONE
BPL 2$
DEC R0 ;:DECREMENT CHARACTER COUNT
BNE 1$ ;:BR IF ALL CHARACTERS NOT TRANSMITTED
BIT #BIT14,@RBUF ;:TEST FOR 'OR' ERROR FLAG
BNE 3$ ;:BR, IF SET
ERROR 135 ;:'OR' ERROR FLAG DID NOT SET

```

2197 010452 032777 100000 172272
2198 010460 001001
2199 010462 104136
2200
2201 010464
2202
2203
2204
(3)
(3)
(2) 010464 000004
(2) 010466 012737 000045 001104
2205 010474 012777 177777 172254
2206 010502 105777 172242
2207 010506 100375
2208 010510 005777 172236
2209 010514 052777 000001 172232
2210 010522 005000
2211 010524 105777 172220
2212 010530 100406
2213 010532 005200
2214 010534 001373
2215
2216 010536 042777 000001 172210
2217 010544 104137
2218
2219 010546 105777 172200
2220 010552 001404
2221 010554 042777 000001 172172
2222
2223 010562 104137
2224
2225 010564 042777 000001 172162
2226
2227
2228
(3)
(3)
(2) 010572 000004
(2) 010574 012737 000046 001104
2229 010602 032777 002000 170230
2230 010610 001025
2231 010612 052777 000001 172134
2232 010620 005077 172132
2233 010624 105777 172120
2234 010630 100375
2235 010632 042777 000001 172114
2236 010640 032777 020000 172104
2237 010646 001001
2238
2239 010650 104140
2240
2241 010652 032777 100000 172072
2242 010660 001001
2243
2244 010662 104141

3\$: BIT #BIT15,@RBUF ;TEST 'ERROR' FLAG
BNE 4\$;BR, IF SET
ERROR 136
; 'ERROR' FLAG DID NOT SET WITH 'DR' FLAG
4\$:
:*****
:*TEST 45 TEST THAT BREAK TRANSMITS ALL ZEROES
:*****
TST45: SCOPE
MOV #45,\$TESTN ;SET TEST NUMBER IN APT MAIL BOX
MOV #-1,@TBUF ;TRANSMIT ALL ONES TO RCVR
1\$: TSTB @RCSR ;WAIT FOR RCVR DONE
BPL 1\$
TST @RBUF ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
BIS #BIT0,@TCSR ;TRANSMIT BREAK
CLR R0 ;CLEAR A TIMER
2\$: TSTB @RCSR ;WAIT FOR RCVR DONE
BMI CONT41 ;BR IF DONE
INC R0 ;IF NOT, INCREMENT TIMER
BNE 2\$;BR IF TIME REMAINS
BIC #BIT0,@TCSR ;CLEAR BREAK BITS
ERROR 137 ;BREAK DID NOT TRANSMIT ANYTHING
CONT41: TSTB @RBUF ;CHECK RECEIVE BUFFER FOR ZERO
BEQ 3\$;BR, IF ZERO
BIC #BIT0,@TCSR ;CLEAR BREAK BITS
ERROR 137 ;BREAK DID NOT TRANSMIT ALL ZEROES
3\$: BIC #BIT0,@TCSR ;CLEAR BREAK BITS
:*****
:*TEST 46 TEST THAT 'FR' ERROR CAN BE SET DURING BREAK
:*****
TST46: SCOPE
MOV #46,\$TESTN ;SET TEST NUMBER IN APT MAIL BOX
BIT #BIT10,@SWR ;IS THIS TEST DISABLED
BNE TST47 ;BR TO NEXT TEST, IF DISABLED
BIS #BIT0,@TCSR ;SEND BREAK
CLR @TBUF ;TRANSMIT A CHARACTER TO TIME BREAK
1\$: TSTB @RCSR ;WAIT FOR RCVR DONE
BPL 1\$
BIC #BIT0,@TCSR ;CLEAR BREAK BITS
BIT #BIT13,@RBUF ;CHECK FOR FRAMING ERROR FLAG
BNE 2\$;BR, IF SET
ERROR 140
2\$: BIT #BIT15,@RBUF ;BREAK DID NOT SET FRAMING ERROR
BNE 3\$;TEST 'ERROR' FLAG
;BR, IF SET
ERROR 141

;'ERROR' FLAG DID NOT SET WITH 'OR' FLAG

2245
2246 010664
2247
(3)
(3)
(2) 010664 0G0004
(2) 010666 012737 000047 001104
2248 010674 032777 000200 170136
2249 010702 001027
2250 010704 005001

3\$:
:*****
:*TEST 47 TEST DATA PATHS USING WRAP CABLE
:*****
TST47: SCOPE
MOV #47,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT7,@\$SWR ;:IS THIS TEST ENABLED
BNE TST50 ;:BR, IF NOT
CLR R1 ;:CLEAR REGISTER FOR TEST DATA

2251
2252
2253
2254 010706 105201
2255 010710 010177 172042
2256 010714 005000
2257 010716 105777 172026
2258 010722 100403
2259 010724 005200
2260 010726 001373

:TRANSMIT A BINARY COUNT PATTERN - UP
:TO THE BIT POSITION INDICATED BY THE
:CONTENTS OF LOCATION '\$USWR'
1\$: INCB R1 ;:INCREMENT THE TEST DATA
MOV R1,@TBUF ;:XMIT A CHARACTER
CLR R0 ;:CLEAR A TIMER
2\$: TSTB @RCSR ;:WAIT FOR RECEIVER DONE
BMI 3\$;:BR IF DONE
INC R0 ;:INCREMENT TIMER IF NOT
BNE 2\$;:BR IF TIME REMAINS

2261
2262 010730 104056
2263
2264 010732 017702 172014
2265 010736 020102
2266 010740 001003
2267 010742 105701
2268 010744 001406
2269 010746 000757
2270 010750 010137 001024
2271 010754 010237 001026

ERROR 56 ;:RECEIVER DONE NOT SET
3\$: MOV @RBUF,R2 ;:GET RECEIVED CHARACTER
CMP R1,R2 ;:COMPARE DATA
BNE 4\$;:BR, IF NON-COMPARE
TSTB R1 ;:TEST XMIT DATA FOR ZERO
BEQ TST50 ;:BR, IF FINISHED
BR 1\$;:CONTINUE IF NOT
4\$: MOV R1,\$GDDAT ;:STORE EXPECTED DATA
MOV R2,\$BDDAT ;:STORE RECEIVED DATA

2272
2273 010760 104142
2274
2275

ERROR 142 ;:DATA COMPARE ERROR WITH WRAP CABLE

2276 010762
2277
2278

LTCIT:
:*****
:*TEST 50 TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY
:*****

(3)
(3)
(2) 010762 000004
(2) 010764 012737 000050 001104
2279 010772 032737 000004 001156
(1) 011000 001002
(1) 011002 000137 011474
(1) 011006
2280 011006 005000
2281 011010 004737 012356
2282 011014 000340
2283 011016 017703 171770
2284 011022 017704 171766
2285 011026 012777 011060 171756
2286 011034 012777 000340 171752

TST50: SCOPE
MOV #50,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT2,@#\$DEVN ;:DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;:IF YES CONTINUE WITH TESTS
JMP BLAST ;:IF NO GO TO START OF NEXT SET OF TESTS.
99\$: CLR R0 ;:SET PSW TO PRIORITY 7
JSR PC,WRPSW
WORD 340
MOV @RTCVT,R3 ;:SAVE LINE CLOCK VECTOR
MOV @RTCP\$W,R4 ;:SAVE LINE CLOCK PSW VECTOR
MOV #2\$,@RTCVT ;:SET RTC INTERRUPT VECTOR TO ERROR REPORT
MOV #340,@RTCP\$W ;:KEEP PRIORITY AT 7

2287
2288 011042 052777 000100 171720
2289 011050 005200

1\$: BIS #BIT6,@LKS ;:SET INTERRUPT ENABLE
INC R0


```
2290 011052 001376 BNE 1$
2291 011054 000240 NOP
2292 011056 000402 BR 3$ ;GIVE TIME FOR ANY INTERRUPTS
;BR, IF NO INTERRUPT OCCURS
2293
2294 011060 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2295 011062 104143 ERROR 143 ;RTC INTERRUPTS AT PRIORITY 7
2296
2297 011064 005077 171700 3$: CLR @LKS ;DISABLE RTC INTERRUPTS
2298 011070 005000 CLR R0
2299 011072 012777 011110 171712 MOV #XXX,@RTCVT ;SET RTC INTERRUPT VECTOR FOR ERROR
2300 011100 004737 012356 JSR PC,WRPSW ;CHANGE PSW TO PRIORITY 5
2301 011104 000240 .WORD 240
2302 011106 104146 ERROR 146
2303 011110 012777 011134 171674 XXX: MOV #4$,@RTCVT
2304 011116 106427 000240 MTPS #240
2305 011122 005200 20$: INC R0
2306 011124 005700 TST R0
2307 011126 100375 BPL 20$
2308 011130 000240 NOP ;GIVE TIME FOR ANY INTERRUPT
2309 011132 000402 BR 5$ ;IF NO INTERRUPT - BR TO CONTINUE TEST
2310
2311 011134 022626 4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2312 011136 104144 ERROR 144 ;RTC INTERRUPTS WITH INTERRUPTS DISABLED
2313
2314 011140 012777 011170 171644 5$: MOV #7$,@RTCVT ;POINT RTC VECTOR TO END OF TEST
2315 011146 005000 CLR R0
2316 011150 052777 000100 171612 BIS #BIT6,@LKS ;ALLOW INTERRUPTS
2317 011156 005200 6$: INC R0
2318 011160 005700 TST R0
2319 011162 100375 BPL 6$
2320 011164 000240 NOP ;GIVE TIME FOR INTERRUPT
2321
2322 011166 104145 ERROR 145 ;RTC INTERRUPT DID NOT OCCUR
2323
2324 011170 022626 7$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2325 011172 042777 000100 171570 BIC #BIT6,@LKS ;DISABLE INTERRUPTS
2326 011200 010377 171606 MOV R3,@RTCVT ;RESTORE LINE CLOCK VECTOR
2327 011204 010477 171604 MOV R4,@RTCP SW ;RESTORE LINE CLOCK PSW VECTOR
2328
2329
2330
2331
(3)
(3)
(2) 011210 000004
(2) 011212 012737 000051 001104
2332 011220 032737 000001 001120
(1) 011226 001403
(1) 011230 005737 001106
(2) 011234 001050
(1) 011236
2333 011236 017703 171550
2334 011242 017704 171546
2335 011246 012777 011312 171536
2336 011254 012777 000340 171532
2337 011262 004737 012356 JSR PC,WRPSW ;SET PRIORITY TO 5
```

```
::*****
:*TEST 51 TEST RTC FOR DOUBLE INTERRUPTS
:*****
```

```
TST51: SCOPE
MOV #51,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #1,@$SENV ;:ARE WE RUNNING UNDER APT
BEQ 70$ ;:IF NO THEN DO TEST
TST @$SPASS ;:IS THIS FIRST PASS
BNE TST52 ;:IF NO THEN SKIP TO NEXT TEST
70$: MOV @RTCVT,R3 ;SAVE LINE CLOCK VECTOR
MOV @RTCP SW,R4 ;SAVE LINE CLOCK PSW VECTOR
MOV #2$,@RTCVT ;SET UP RTC INTERRUPT VECTOR
MOV #340,@RTCP SW ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
JSR PC,WRPSW ;SET PRIORITY TO 5
```



```
2338 011266 000240 .WORD 240
2339
2340 011270 005000 CLR R0
2341 011272 052777 000100 171470 BIS #BIT6,@LKS ;ENABLE CLOCK INTERRUPTS
2342 011300 005200 1$: INC R0
2343 011302 005700 TST R0
2344 011304 100375 BPL 1$
2345 011306 000240 NOP ;GIVE TIME FOR ANY INTERRUPT
2346
2347 011310 104146 ERROR 146 ;RTC INTERRUPT DID NOT OCCUR
2348
2349 011312 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2350 011314 012777 011334 171470 MOV #3$,@RTCVT ;POINT RTC VECTOR TO ERROR REPORT
2351 011322 004737 012356 JSR PC,WRPSW ;SET PSW TO PRIORITY 5
2352 011326 000240 .WORD 240
2353 011330 000240 NOP ;GIVE SOME TIME FOR AN INTERRUPT
2354 011332 000402 BR 4$ ;NO INTERRUPT - BR TO END OF TEST
2355
2356 011334 022626 3$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2357 011336 104147 ERROR 147 ;INTERRUPT SEQUENCE DID NOT CLEAR
2358 ;INTERRUPT REQUEST
2359
2360 011340 042777 000100 171422 4$: BIC #BIT6,@LKS ;DISABLE CLOCK INTERRUPTS
2361 011346 010377 171440 MOV R3,@RTCVT ;RESTORE LINE CLOCK VECTOR
2362 011352 010477 171436 MOV R4,@RTCP SW ;RESTORE LINE CLOCK PSW VECTOR
2363
2364
2365 ;:*****
(3) ;:*TEST 52 TEST THAT RTC INTERRUPT CLEARS WITH RESET
(3) ;:*****
(2) 011356 000004 TST52: SCOPE
(2) 011360 012737 000052 001104 MOV #52,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
2366 011366 032737 000001 001120 BIT #1,@$ENV ;:ARE WE RUNNING UNDER APT
(1) 011374 001403 BEQ 70$ ;:IF NO THEN DO TEST
(1) 011376 005737 001106 TST @$SPASS ;:IS THIS FIRST PASS
(2) 011402 001034 BNE 70$ ;:IF NO THEN SKIP TO NEXT TEST
(1) 011404 70$: JSR PC,WRPSW ;SET PRIORITY TO 7
2367 011404 004737 012356 .WORD 340
2368 011410 000340 MOV @RTCVT,R3 ;SAVE LINE CLOCK VECTOR
2369 011412 017703 171374 MOV #2$,@RTCVT ;POINT RTC VECTOR TO ERROR REPORT
2370 011416 012777 011464 171366
2371
2372 011424 005000 CLR R0
2373 011426 052777 000100 171334 BIS #BIT6,@LKS ;ENABLE CLOCK INTERRUPTS
2374 011434 005200 1$: INC R0
2375 011436 005700 TST R0
2376 011440 100375 BPL 1$
2377 011442 000005 RESET ;CLEAR PENDING INTERRUPT WITH RESET
2378 011444 004737 012356 JSR PC,WRPSW ;SET PRIORITY TO 5
2379 011450 000240 .WORD 240
2380 011452 000240 NOP ;GIVE TIME FOR ANY INTERRUPT
2381 011454 042777 000100 171306 BIC #BIT6,@LKS ;DISALLOW INTERRUPTS
2382 011462 000402 BR 3$ ;BR TO END OF TEST
2383
2384 011464 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2385 011466 104150 ERROR 150 ;RESFT DID NOT CLEAR INTERRUPT
```

```
2386  
2387 011470 010377 171316 3$: MOV R3,@RTCVT ;RESTORE LINE CLOCK VECTOR  
2388  
2389 011474 BLAST:  
2390  
2391  
(3) ;*****  
(3) ;*TEST 53 TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY  
(2) ;*****  
(2) 011474 000004 TST53: SCOPE ;  
2392 011476 012737 000053 001104 MOV #53,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX  
2393 011512 001405 BIT #BIT0,@$ENV ;:ARE WE RUNNING UNDER APT  
2394 011514 005737 001106 BEQ 70$ ;:IF NO DO TEST  
2395 011520 001402 TST @$SPASS ;:IS THIS FIRST PASS  
2396 011522 000137 012240 BEQ 70$ ;:IF YES DO TEST  
2397 011526 000005 JMP $EOP ;:IF NO DO NOT DO TEST  
2398 011530 012737 000340 177776 70$: RESET ;:CLEAR EVERY BODY  
2399 011536 017737 171234 001060 MOV #340,PS ;:SET PROCESSOR PRIORITY TO 7  
2400 011544 017737 171222 001062 MOV @TVECT,$TMP0  
2401 011552 017737 171230 001064 MOV @RVECT,$TMP1  
2402 011560 017737 171216 001066 MOV @CTVECT,$TMP2  
2403 011566 017737 171220 001070 MOV @CRVECT,$TMP3  
2404 011574 005037 012232 CLR XMTCT2  
2405 011600 005037 012234 CLR RECCT2  
2406 011604 005037 012236 CLR COUNT  
2407  
2408 011610 012777 011726 171160 MOV #XMIT2,@TVECT ;:SET UP SLU2 TRANSMIT VECTOR  
2409 011616 012777 000340 171154 MOV #340,@TPSW ;:AND PSW  
2410 011624 012777 011762 171140 MOV #REC2,@RVECT ;:SET UP SLU2 RECEIVER VECTOR  
2411 011632 012777 000340 171134 MOV #340,@RPSW ;:AND PSW  
2412 011640 012777 011716 171144 MOV #TICKER,@RTCVT ;:SET UP RTC VECTOR  
2413 011646 012777 000340 171140 MOV #340,@RTCPSW ;:AND PSW  
2414  
2415 011654 052777 000100 171072 BIS #BIT6,@TCSR ;:ENABLE SLU2 XMIT INTERRUPT  
2416 011662 052777 000100 171060 BIS #BIT6,@RCR ;:ENABLE SLU2 RECEIVER INTERRUPT  
2417 011670 012703 026320 MOV #BUF2,R3 ;:SET UP RECEIVER BUFFER  
2418  
2419 011674 052777 000100 171066 3$: BIS #BIT6,@LKS ;:ENABLE RTC INTERRUPTS  
2420 011702 012701 177777 MOV #-1,R1 ;:INITIALIZE DATA FOR SLU2(1ST CHR. 0)  
2421 011706 005037 177776 CLR PS ;:DROP PROCESSOR PRIORITY TO ZERO  
2422 011712 000001 WAITIO: WAIT ;:WAIT FOR INTERRUPT  
2423 011714 000776 BR WAITIO  
2424  
2425 011716 005237 012236 TICKER: INC COUNT ;:UPDATE COUNT  
2426 011722 000137 012036 JMP IOHAND ;:GO TO INTERRUPT HANDLER  
2427  
2428  
2429 011726 005237 012232 XMIT2: INC XMTCT2 ;:UPDATE XMIT INTERRUPT COUNT  
2430 011732 005201 INC R1 ;:UPDATE XMIT DATA  
2431 011734 010177 171016 MOV R1,@TBUF ;:SEND NEXT CHARACTER  
2432 011740 023727 012232 000400 CMP XMTCT2,#400 ;:IF 256 CHARACTERS HAVE NOT  
2433 011746 002403 BLT 1$ ;:BEEN TRANSFERRED CONTINUE  
2434 011750 042777 000100 170776 BIC #BIT6,@TCSR ;:ELSE NO MORE XMIT INTERRUPTS  
2435 011756 000137 012036 1$: JMP IOHAND ;:GO TO INTERRUPT HANDLER  
2436  
2437 011762 005237 012234 REC2: INC RECCT2 ;:UPDATE RECEIVER INTERRUPT COUNT
```

```

2438 011766 005777 170760          TST  @RBUF          ;BIT 15 SETS IF ANY ERRORS OCCURRED
2439 011772 100017                    BPL  3$             ;IF BIT IS CLEAR NO ERRORS
2440 011774 017737 170752 001026    MOV  @RBUF, $BDDAT ;GET ERROR INFORMATION
2441 012002 000005                    RESET ;CLEAR THE WORLD - STOP ALL
2442                                     ;INTERRUPTS
2443 012004 020327 026320          CMP  R3, #BUF2     ;WAS MORE THAN 1 WORD TRANSFERRED
2444 012010 003004                    BGT  1$             ;IF YES GET LAST GOOD DATA
2445 012012 012737 177777 001027    MOV  #-1, $GDDAT  ;IF NO MAKE GOOD DATA -1
2446 012020 000403                    BR   2$             ;AND GET TO ERROR REPORT
2447 012022 116337 177777 001024    1$: MOVB -1(R3), $GDDAT ;GET LAST GOOD DATA RECEIVED
2448 012030 104154                    2$: ERROR 154      ;RECEIVER STATUS ERROR
2449 012032 117723 170714          3$: MOVB  @RBUF, (R3)+ ;GET DATA AND STORE IT
2450
2451 012036 023727 012236 000074    IOHAND: CMP  COUNT, #74 ;HAS 1 SEC ELAPSED
2452 012044 002412                    BLT  3$             ;IF NO CONTINUE TEST
2453 012046 042777 000100 170710    BIC  #BIT6, @CTCSR ;IF YES STOP TRANSMISSIONS
2454 012054 042777 000100 170672    BIC  #BIT6, @TCSR  ;
2455 012062 042777 000100 170700    BIC  #BIT6, @LKS   ;TURN OFF LINE CLOCK
2456 012070 000401                    BR   WAITER        ;
2457 012072 000002                    3$: RTI             ;RETURN FROM INTERRUPT TO AWAIT NEXT
2458
2459 012074 005037 177776          WAITER: CLR  PS     ;MAKE PROCESSOR PRIORITY 0
2460 012100 012705 140000          MOV  #-40000, R5  ;SET UP LOOP COUNTER
2461 012104 062705 000001          1$: ADD  #1, R5   ;DO LOOP UNTIL R5 = 0
2462 012110 001375                    BNE  1$
2463 012112 000005                    RESET ;STOP EVERYONE SHOULD BE DONE
2464 012114 012706 001000          MOV  #1000, SP   ;RESET STACK AFTER LAST INTERRUPT
2465
2466
2467 012120 023737 012232 012234    CHECK2: CMP  XMTCT2, RECCT2 ;#OF XMIT INTERRUPTS = REC INTERRUPTS
2468 012126 001401                    BEQ  1$             ;IF YES CHECK DATA
2469 012130 104157                    ERROR 157          ;INTERRUPT COMPARISON ERROR
2470 012132 012703 026320          1$: MOV  #BUF2, R3 ;INITIALIZE TO FIRST RECEIVED DATA
2471 012136 005001                    CLR  R1             ;INITIALIZE TO FIRST XMIT DATA
2472 012140 013704 012232          MOV  XMTCT2, R4   ;GET # OF BYTES TRANSFERRED
2473 012144 122301                    2$: CMPB (R3)+, R1 ;IS RECEIVED DATA = EXPECTED DATA
2474 012146 001406                    BEQ  3$             ;IF YES CONTINUE TESTING
2475 012150 114337 001026          MOVB -(R3), $BDDAT ;IF NO GET ERROR INFORMATION
2476 012154 010137 001024          MOV  R1, $GDDAT
2477 012160 104160                    ERROR 160          ;SLU2 DATA COMPARISON ERROR
2478 012162 005203                    INC  R3             ;IF COONTINUE ON ERROR RESET POINTER
2479 012164 005201                    3$: INC  R1         ;UPDATE TO NEXT GOOD DATA
2480 012166 077412                    SOB  R4, 2$        ;LOOP UNTIL ALL DATA CHECKED
2481 012170 013777 001060 170600    FINIE: MOV  $TMP0, @TVECT ;RESTORE VECTORS
2482 012176 013777 001062 170566    MOV  $TMP1, @RVECT
2483 012204 013777 001064 170574    MOV  $TMP2, @CTVECT
2484 012212 013777 001066 170562    MOV  $TMP3, @CRVECT
2485 012220 013777 001070 170564    MOV  $TMP4, @RTCVT
2486 012226 000137 012240          JMP  $EOP         ;FINISHED TESTING GO TO END OF PASS
2487
2488 012232 000000          XMTCT2: .WORD 0
2489 012234 000000          RECCT2: .WORD 0
2490 012236 000000          COUNT:  .WORD 0
2491

```

2493
2494

(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1) 012240
(1) 012240 000004
(1) 012242 005037 001002
(1) 012246 005237 001106
(1) 012252 042737 100000 001106
(1) 012260 005327
(1) 012262 000001
(1) 012264 003022
(1) 012266 012737
(1) 012270 000001
(1) 012272 012262
(1) 012274 104401 012341
(2) 012300 013746 001106
(2) 012304 104405
(1) 012306 104401 012336
(1) 012312 013700 000042
(1) 012316 001405
(1) 012320 000005
(1) 012322 004710
(1) 012324 000240
(1) 012326 000240
(1) 012330 000240
(1) 012332
(1) 012332 000137
(1) 012334 003430
(1) 012336 377 377 000
(1) 012341 015 042412 042116 020123
(1) 012346 050040 051501
(1) 012354 000043

.SBTTL END OF PASS ROUTINE

*INCREMENT THE PASS NUMBER (\$PASS)
*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
*IF THERES A MONITOR GO TO IT
*IF THERE ISN'T JUMP TO TST1

\$EOP:

SCOPE
CLR \$STSTM ;:ZERO THE TEST NUMBER
INC \$PASS ;:INCREMENT THE PASS NUMBER
BIC #100000,\$PASS ;:DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;:LOOP?

\$EOPCT: .WORD 1

BGT \$DOAGN ;:YES
MOV (PC)+,@(PC)+ ;:RESTORE COUNTER

\$ENDCT: .WORD 1

\$EOPCT
TYPE \$SENDMG ;:TYPE 'END PASS #'
MOV \$PASS,-(SP) ;:SAVE \$PASS FOR TYPEOUT
TYPDS ;:GO TYPE--DECIMAL ASCII WITH SIGN

TYPE \$ENULL ;:TYPE A NULL CHARACTER

\$GET42: MOV @#42,R0 ;:GET MONITOR ADDRESS

BEQ \$DOAGN ;:BRANCH IF NO MONITOR

RESET ;:CLEAR THE WORLD

\$ENDAD: JSR PC,(R0) ;:GO TO MONITOR

NOP ;:SAVE ROOM

NOP ;:FOR

NOP ;:ACT11

\$DOAGN: JMP @(PC)+ ;:RETURN

\$RTNAD: .WORD TST1

\$ENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING

\$SENDMG: .ASCIZ <15><12>/END PASS #/

2495

2496 012356 011646
2497 012360 013616
2498 012362 062746 000002
2499 012366 000002

WRPSW: MOV(SP),-(SP) ;:COPY RETURN PC
MOV @(SP)+,(SP) ;:MOVE NEW PSW TO STACK
ADD #2,-(SP) ;:ADJUST JSR RETURN
RTI ;:POP RETURN PC & NEW PSW

2500

2501

2502

2503 012370 012600
2504 012372 162700 000004
2505 012376 010037 012416
2506 012402 016637 000002 012414
2507 012410 104161

;SUBROUTINE TO REPORT UNEXPECTED OR ERRONEOUS TRAPS OR INTERRUPTS

CATCH: MOV (SP)+,R0 ;:GET ADDRESS OF TRAP VECTOR + 4
SUB #4,R0 ;:ADJUST TO POINT TO TRAP ADDRESS
MOV R0,BDVECT ;:STORE TRAP OR INTERRUPT ADDRESS
MOV 2(SP),OLDPC ;:GET PC WHERE TRAP OR INTERRUPT OCCURRED
ERROR 161 ;:REPORT ERROR

2508

2509 012412 000000

HALT ;:PROGRAM MUST BE RESTARTED AT THIS POINT

2510 012414 000000

OLDPC: .WORD 0

2511 012416 000000

BDVECT: .WORD 0

2512

2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526 012420
2527 012420 105237 001003
2528 012424 001775
2529 012426 013777 001002 166406
2530 012434 005237 001012
2531 012440 011637 001016
2532 012444 162737 000002 001016
2533 012452 117737 166340 001014
2534 012460 032777 020000 166352
2535 012466 001004
2536 012470 004737 012602
2537 012474 104401 001075
2538 012500
2539 012500 122737 000001 001120
2540 012506 001007
2541 012510 113737 001014 012522
2542 012516 004737 013304
2543 012522 000
2544 012523 000
2545 012524 000777
2546 012526 005777 166306
2547 012532 100001
2548 012534 000000
2549 012536 104407
2550 012540 032777 001000 166272
2551 012546 001402
2552 012550 013716 001010
2553 012554 005737 001072
2554 012560 001402
2555 012562 013716 001072
2556 012566
2557 012566 022737 012322 000042
2558 012574 001001
2559 012576 000000
2560 012600
2561 012600 000002
2562

```
*****
;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
;SAVE THE ERROR ITEM NUMBER AND ADDRESS OF THE ERROR CALL
;AND GO TO $ERRTYP ON ERROR
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW15=1      HALT ON ERROR
;SW13=1      INHIBIT ERROR TYPEOUTS
;SW09=1      LOOP IN ERROR
;CALL
;*          ERROR      N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER
*****

$ERROR:
7$:      INCB      $ERFLG          ;SET THE ERROR FLAG
        BEQ      7$              ;DON'T LET FLAG GO TO ZERO
        MOV      $STNM,@DISPLAY  ;DISPLAY TEST NUMBER AND ERROR FLAG
        INC      $ERTL          ;INCREMENT ERROR COUNT
        MOV      (SP),$ERRPC     ;GET ADDRESS OF ERROR INSTRUCTION
        SUB      #2,$ERRPC
        MOVB     @$ERRPC,$ITEMB  ;STRIP AND SAVE THE ERROR ITEM CODE
        BIT      #BIT13,@SWR     ;SKIP TYPEOUT IF SET
        BNE     20$             ;SKIP TYPEOUTS
        JSR      PC,$ERRTYP      ;GO TO USER ERROR ROUTINE
        TYPE     ,SCLF

20$:     CMPB     #APTENV,$ENV    ;RUNNING IN APT MODE
        BNE     2$              ;NO, SKIP APT ERROR REPORT
        MOVB     $ITEMB,21$      ;SET ITEM NUMBER AS ERROR NUMBER
        JSR      PC,$ATY4        ;REPORT FATAL ERROR TO APT

21$:     .BYTE   0
        .BYTE   0

22$:     BR      22$            ;APT ERROR LOOP
2$:      TST     @SWR           ;HALT ON ERROR
        BPL     3$             ;SKIP IF CONTINUE
        HALT    ;HALT ON ERROR!
3$:      CKSWR  ;TEST FOR CHANGE IN SOFT-SWR
        BIT     #BIT09,@SWR     ;LOOP ON ERROR SWITCH SET?
        BEQ     4$             ;BR IF NO
        MOV     $LPERR,(SP)     ;FUDGE RETURN FOR LOOPING
        TST     $ESCAPE        ;CHECK FOR AN ESCAPE ADDRESS
        BEQ     5$             ;BR IF NONE
        MOV     $ESCAPE,(SP)   ;FUDGE RETURN ADDRESS FOR ESCAPE

5$:      CMP     #$ENDAD,@#42   ;ACT-11 AUTO-ACCEPT?
        BNE     6$             ;BR IF NO
        HALT    ;YES

6$:      RTI                    ;RETURN
```

2564
 2565

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

(1)
 (2)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1) 012602
 (1) 012602 104401 001075
 (1) 012606 010046
 (1) 012610 005000
 (1) 012612 153700 001014
 (1) 012616 001004
 (1)
 (2) 012620 013746 001016
 (2)
 (2) 012624 104402
 (1) 012626 000426
 (1) 012630 005300
 (1) 012632 006300
 (1) 012634 006300
 (1) 012636 006300
 (1) 012640 062700 001160
 (1) 012644 012037 012654
 (1) 012650 001404
 (1) 012652 104401
 (1) 012654 000000
 (1) 012656 104401 001075
 (1) 012662 012037 012672
 (1) 012666 001404
 (1) 012670 104401
 (1) 012672 000000
 (1) 012674 104401 001075
 (1) 012700 011000
 (1) 012702 001004
 (1) 012704 012600
 (1) 012706 104401 001075
 (1) 012712 000207
 (1)
 (2) 012714 013046
 (2) 012716 104402
 (1) 012720 005710
 (1) 012722 001770
 (1) 012724 104401 012732
 (1) 012730 000771
 (1) 012732 020040 000
 (1)
 2566

```

*****
*THIS ROUTINE USES THE 'ITEM CONTROL BYTE' ($ITEMB) TO DETERMINE WHICH
*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE 'ERROR TABLE' ($ERRTB),
*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
*****
$ERRTYP:
      TYPE  , $CRLF      ;; 'CARRIAGE RETURN' & 'LINE FEED'
      MOV   RO, -(SP)    ;; SAVE RO
      CLR   RO           ;; PICKUP THE ITEM INDEX
      BISB  @#$ITEMB, RO
      BNE   1$          ;; IF ITEM NUMBER IS ZERO, JUST
                          ;; TYPE THE PC OF THE ERROR
                          ;; SAVE $ERRPC FOR TYPEOUT
                          ;; ERROR ADDRESS
                          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
                          ;; GET OUT
      MOV   $ERRPC, -(SP)
                          ;; ADJUST THE INDEX SO THAT IT WILL
                          ;; WORK FOR THE ERROR TABLE
      TYPOC
      BR    6$
1$:   DEC   RO
      ASL  RO
      ASL  RO
      ASL  RO
      ADD  #$ERRTB, RO   ;; FORM TABLE POINTER
      MOV  (RO)+, 2$    ;; PICKUP 'ERROR MESSAGE' POINTER
      BEQ  3$          ;; SKIP TYPEOUT IF NO POINTER
      TYPE 'ERROR MESSAGE'
                          ;; 'ERROR MESSAGE' POINTER GOES HERE
2$:   .WORD 0
      TYPE , $CRLF      ;; 'CARRIAGE RETURN' & 'LINE FEED'
3$:   MOV  (RO)+, 4$    ;; PICKUP 'DATA HEADER' POINTER
      BEQ  5$          ;; SKIP TYPEOUT IF 0
      TYPE 'DATA HEADER'
                          ;; 'DATA HEADER' POINTER GOES HERE
4$:   .WORD 0
      TYPE , $CRLF      ;; 'CARRIAGE RETURN' & 'LINE FEED'
5$:   MOV  (RO), RO     ;; PICKUP 'DATA TABLE' POINTER
      BNE  7$          ;; GO TYPE THE DATA
6$:   MOV  (SP)+, RO    ;; RESTORE RO
      TYPE , $CRLF      ;; 'CARRIAGE RETURN' & 'LINE FEED'
7$:   RTS   PC         ;; RETURN
      MOV  @ (RO)+, -(SP) ;; SAVE @ (RO)+ FOR TYPEOUT
      TYPOC
                          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
                          ;; IS THERE ANOTHER NUMBER?
      TST  (RO)
      BEQ  8$          ;; BR IF NO
      TYPE '      '    ;; TYPE TWO(2) SPACES
      BR   7$         ;; LOOP
8$:   .ASCIIZ / /      ;; TWO(2) SPACES
      .EVEN
  
```

2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612

```
.SBTTL POWER DOWN AND UP ROUTINES  
:;*****  
:;*POWER DOWN ROUTINE  
:;*****  
$PWRDN: MOV    $IILLUP,@#PWRVEC ;SET FOR FAST UP  
          MOV    #340,@#PWRVEC+2 ;PRIO:7  
          MOV    R0,-(SP) ;PUSH R0 ON STACK  
          MOV    R1,-(SP) ;PUSH R1 ON STACK  
          MOV    R2,-(SP) ;PUSH R2 ON STACK  
          MOV    R3,-(SP) ;PUSH R3 ON STACK  
          MOV    R4,-(SP) ;PUSH R4 ON STACK  
          MOV    R5,-(SP) ;PUSH R5 ON STACK  
          MOV    @SWR,-(SP) ;PUSH @SWR ON STACK  
          MOV    SP,$SAVR6 ;SAVE SP  
          MOV    #PWRUP,@#PWRVEC ;SET UP VECTOR  
          HALT  
          BR     .-2 ;HANG UP  
  
:;*****  
:;*POWER UP ROUTINE  
:;*****  
$PWRUP: MOV    $IILLUP,@#PWRVEC ;SET FOR FAST DOWN  
          MOV    $SAVR6,SP ;GET SP  
          MOV    (SP)+,@SWR ;POP STACK INTO @SWR  
          MOV    (SP)+,R5  
          MOV    (SP)+,R4 ;POP STACK INTO R4  
          MOV    (SP)+,R3 ;POP STACK INTO R3  
          MOV    (SP)+,R2 ;POP STACK INTO R2  
          MOV    (SP)+,R1 ;POP STACK INTO R1  
          MOV    (SP)+,R0 ;POP STACK INTO R0  
          MOV    #PWRDN,@#PWRVEC ;SET UP THE POWER DOWN VECTOR  
          MOV    #340,@#PWRVEC+2 ;PRIO:7  
          CLR    $SAVR6 ;WAIT LOOP FOR THE TTY  
1$:      INC    $SAVR6 ;WAIT FOR THE INC  
          BNE   1$ ;OF WORD  
          TYPE  ;REPORT THE POWER FAILURE  
$PWRMG: .WORD  $POWER ;POWER FAIL MESSAGE POINTER  
          RTI  
$IILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED  
          BR     .-2 ;BEFORE THE POWER DOWN WAS COMPLETE  
          $SAVR6: 0 ;PUT THE SP HERE  
$POWER: .ASCIIZ <15><12>'POWER''
```



```
2618  
2619  
2620  
2621  
2622  
2623 013266 112737 000001 013532 SATY1: MOV #1,$FFLG ;TO REPORT FATAL ERROR  
2624 013274 112737 000001 013530 SATY3: MOV #1,$MFLG ;TO TYPE A MESSAGE  
2625 013302 000403 BR SATYC  
2626 013304 112737 000001 013532 SATY4: MOV #1,$FFLG ;TO ONLY REPORT FATAL ERROR  
2627 013312 SATYC:  
2628 013312 010046 MOV R0,-(SP) ;PUSH R0 ON STACK  
2629 013314 010146 MOV R1,-(SP) ;PUSH R1 ON STACK  
2630 013316 105737 013530 TSTB $MFLG ;SHOULD TYPE A MESSAGE?  
2631 013322 001450 BEQ 5$ ;IF NOT: BR  
2632 013324 122737 000001 001120 CMPB #APTENV,$ENV ;OPERATING UNDER APT?  
2633 013332 001031 BNE 3$ ;IF NOT: BR  
2634 013334 132737 000100 001121 BITB #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?  
2635 013342 001425 BEQ 3$ ;IF NOT: BR  
2636 013344 017600 000004 MOV @4(SP),R0 ;GET MESSAGE ADDRESS  
2637 013350 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS  
2638 013356 005737 001100 1$: TST $MSGTYPE ;SEE IF DONE W/ LAST XMISSION?  
2639 013362 001375 BNE 1$ ;IF NOT: WAIT  
2640 013364 010037 001114 MOV R0,$MSGAD ;PUT ADDRESS IN MAILBOX  
2641 013370 105720 2$: TSTB (R0)+ ;FIND END OF MESSAGE  
2642 013377 001376 BNE 2$  
2643 013374 163700 001114 SUB $MSGAD,R0 ;SUB START OF MESSAGE  
2644 013400 006200 ASR R0 ;GET MESSAGE LENGTH IN WORDS  
2645 013402 010037 MOV R0,$MSGLGT ;PUT LENGTH IN MAILBOX  
2646 013406 012737 000004 001100 MOV #4,$MSGTYPE ;TELL APT TO TAKE MESSAGE  
2647 013414 000413 BR 5$  
2648 013416 017637 000004 013442 3$: MOV @4(SP),4$ ;PUT MSG ADDR IN JSR LINKAGE  
2649 013424 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS  
2650 013432 013746 177776 MOV 177776,-(SP) ;PUSH 177776 ON STACK  
2651 013436 004737 013534 JSR PC,$TYPE ;CALL TYPE MACRO  
2652 013442 000000 4$: .WORD 0  
2653 013444 5$:  
2654 013444 105737 013532 10$: TSTB $FFLG ;SHOULD REPORT FATAL ERROR?  
2655 013450 001413 BEQ 12$ ;IF NOT: BR  
2656 013452 005737 001120 TST $ENV ;RUNNING UNDER APT?  
2657 013456 001410 BEQ 12$ ;IF NOT: BR  
2658 013460 005737 001100 11$: TST $MSGTYPE ;FINISHED LAST MESSAGE?  
2659 013464 001375 BNE 11$ ;IF NOT: WAIT  
2660 013466 017637 000004 001102 MOV @4(SP),$FATAL ;GET ERROR #  
2661 013474 005237 001100 INC $MSGTYPE ;TELL APT TO TAKE ERROR  
2662 013500 062766 000002 000004 12$: ADD #2,4(SP) ;BUMP RETURN ADDRESS  
2663 013506 105037 013532 CLRFB $FFLG ;CLEAR FATAL FLAG  
2664 013512 105037 013531 CLRFB $LFLG ;CLEAR LOG FLAG  
2665 013516 105037 013530 CLRFB $MFLG ;CLEAR MESSAGE FLAG  
2666 013522 012601 MOV (SP)+,R1 ;POP STACK INTO R1  
2667 013524 012600 MOV (SP)+,R0 ;POP STACK INTO R1  
2668 013526 000207 RTS PC ;RETURN  
2669 013530 000 $MFLG: .BYTE 0  
2670 013531 000 $LFLG: .BYTE 0 ;LOG FLAG  
2671 013532 000 $FFLG: .BYTE 0 ;FATAL FLAG  
2672  
2673 013534 .EVEN
```

CJKDIAO 11/238 SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82^{B 6} 16:42 PAGE 15-1
APT COMMUNICATIONS ROUTINE

SEQ 0066

2674 000200
2675 000001
2676 000100
2677 000040
2678

APTSIZE=200
APTENV=001
APTSPOOL=100
APTCSUP=040


```
(1) 013714 105337 014064          DECB  $CHARCNT      ;;DO NOT COUNT AS A COUNT
(1) 013720 000770                   BR    7$           ;;LOOP
(1)
(1)
(1)          ;HORIZONTAL TAB PROCESSOR
(1) 013722 112716 000040          8$:  MOVB  #' , (SP)      ;;REPLACE TAB WITH SPACE
(1) 013726 004737 013746          9$:  JSR   PC,$TYPEC      ;;TYPE A SPACE
(1) 013732 132737 000007 014064  BITB  #7,$CHARCNT      ;;BRANCH IF NOT AT
(1) 013740 001372                   BNE   9$           ;;TAB STOP
(1) 013742 005726                   TST  (SP)+         ;;POP SPACE OFF STACK
(1) 013744 000724                   BR   2$           ;;GET NEXT CHARACTER
(1) 013746                   $TYPEC:
(1) 013746 105777 165072          TSTB  @STKS         ;;CHAR IN KYBD BUFFER?      ;:MJD001
(1) 013752 100022                   BPL  10$          ;;BR IF NOT                  ;:MJD001
(1) 013754 017746 165066          MOV   @STKB,-(SP)    ;;GET CHAR                    ;:MJD001
(1) 013760 042716 177600          BIC  #177600,(SP)   ;;STRIP EXTRANEOUS BITS     ;:MJD001
(1) 013764 122716 000023          CMPB  #$XOFF,(SP)   ;;WAS CHAR XOFF              ;:MJD001
(1) 013770 001012                   BNE  102$         ;;BR IF NOT                  ;:MJD001
(1) 013772                   101$:
(1) 013772 105777 165046          TSTB  @STKS         ;;WAIT FOR CHAR              ;:MJD001
(1) 013776 100375                   BPL  101$         ;;BR IF NOT                  ;:MJD001
(1) 014000 117716 165042          MOVB  @STKB,(SP)    ;;GET CHAR                    ;:MJD001
(1) 014004 042716 177600          BIC  #177600,(SP)   ;;STRIP IT                    ;:MJD001
(1) 014010 122716 000021          CMPB  #$XON,(SP)   ;;WAS IT XON?                ;:MJD001
(1) 014014 001366                   BNE  101$         ;;BR IF NOT                  ;:MJD001
(1) 014016                   102$:
(1) 014016 005726                   TST  (SP)+         ;;FIX STACK                  ;:MJD001
(1) 014020                   10$:
(1) 014020 105777 165024          TSTB  @STPS         ;;WAIT UNTIL PRINTER IS READY ;:MJD001
(1) 014024 100375                   BPL  10$           ;;BR IF NOT                  ;:MJD001
(1) 014026 116677 000002 165016  MOVB  2(SP),@STPB   ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 014034 122766 000015 000002  CMPB  #CR,2(SP)     ;;IS CHARACTER A CARRIAGE RETURN?
(1) 014042 001003                   BNE  1$           ;;BRANCH IF NO
(1) 014044 105037 014064          CLRB  $CHARCNT     ;;YES--CLEAR CHARACTER COUNT
(1) 014050 000406                   BR   $TYPEX       ;;EXIT
(1) 014052 122766 000012 000002  1$:  CMPB  #LF,2(SP)   ;;IS CHARACTER A LINE FEED?
(1) 014060 001402                   BEQ  $TYPEX       ;;BRANCH IF YES
(1) 014062 105227                   INCB (PC)+        ;;COUNT THE CHARACTER
(1) 014064 000000          $CHARCNT: .WORD  0      ;;CHARACTER COUNT STORAGE
(1) 014066 000207          $TYPEX:  RTS   PC
(1)
(1)          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
(1)
(1)          ;*****
(1)          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
(1)          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
(1)          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)          ;*CALL:
(1)          ;*      MOV    NUM,-(SP)      ;;NUMBER TO BE TYPED
(1)          ;*      TYPOS      ;;CALL FOR TYPEOUT
(1)          ;*      .BYTE  N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1)          ;*      .BYTE  M          ;;M=1 OR 0
(1)          ;*
(1)          ;*          ;;1=TYPE LEADING ZEROS
(1)          ;*          ;;0=SUPPRESS LEADING ZEROS
(1)          ;*
(1)          ;*$STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1)
```

2682

(1)					;*STYPOS OR \$TYPOC	
(1)					;*CALL:	
(1)					;* MOV NUM,-(SP)	::NUMBER TO BE TYPED
(1)					;* TYPON	::CALL FOR TYPEOUT
(1)					;* \$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER	
(1)					;*CALL:	
(1)					;* MOV NUM,-(SP)	::NUMBER TO BE TYPED
(1)					;* TYPOC	::CALL FOR TYPEOUT
(1)					;	
(1)	014070	017646	000000		\$TYPOS: MOV @ (SP),-(SP)	::PICKUP THE MODE
(1)	014074	116637	000001	014313	MOVB 1(SP), \$OFILL	::LOAD ZERO FILL SWITCH
(1)	014102	112637	014315		MOVB (SP)+, \$OMODE+1	::NUMBER OF DIGITS TO TYPE
(1)	014106	062716	000002		ADD #2,(SP)	::ADJUST RETURN ADDRESS
(1)	014112	000406			BR \$TYPON	
(1)	014114	112737	000001	014313	\$TYPOC: MOVB #1,\$OFILL	::SET THE ZERO FILL SWITCH
(1)	014122	112737	000006	014315	MOVB #6,\$OMODE+1	::SET FOR SIX(6) DIGITS
(1)	014130	112737	000005	014312	\$TYPON: MOVB #5,\$OCNT	::SET THE ITERATION COUNT
(1)	014136	010346			MOV R3,-(SP)	::SAVE R3
(1)	014140	010446			MOV R4,-(SP)	::SAVE R4
(1)	014142	010546			MOV R5,-(SP)	::SAVE R5
(1)	014144	113704	014315		MOVB \$OMODE+1,R4	::GET THE NUMBER OF DIGITS TO TYPE
(1)	014150	005404			NEG R4	
(1)	014152	062704	000006		ADD #6,R4	::SUBTRACT IT FOR MAX. ALLOWED
(1)	014156	110437	014314		MOVB R4,\$OMODE	::SAVE IT FOR USE
(1)	014162	113704	014313		MOVB \$OFILL,R4	::GET THE ZERO FILL SWITCH
(1)	014166	016605	000012		MOV 12(SP),R5	::PICKUP THE INPUT NUMBER
(1)	014172	005003			CLR R3	::CLEAR THE OUTPUT WORD
(1)	014174	006105		1\$:	ROL R5	::ROTATE MSB INTO 'C'
(1)	014176	000404			BR 3\$::GO DO MSB
(1)	014200	006105		2\$:	ROL R5	::FORM THIS DIGIT
(1)	014202	006105			ROL R5	
(1)	014204	006105			ROL R5	
(1)	014206	010503			MOV R5,R3	
(1)	014210	006103		3\$:	ROL R3	::GET LSB OF THIS DIGIT
(1)	014212	105337	014314		DECB \$OMODE	::TYPE THIS DIGIT?
(1)	014216	100016			BPL 7\$::BR IF NO
(1)	014220	042703	177770		BIC #177770,R3	::GET RID OF JUNK
(1)	014224	001002			BNE 4\$::TEST FOR 0
(1)	014226	005704			TST R4	::SUPPRESS THIS 0?
(1)	014230	001403			BEQ 5\$::BR IF YES
(1)	014232	005204		4\$:	INC R4	::DON'T SUPPRESS ANYMORE 0'S
(1)	014234	052703	000060		BIS #'0,R3	::MAKE THIS DIGIT ASCII
(1)	014240	052703	000040	5\$:	BIS #' ,R3	::MAKE ASCII IF NOT ALREADY
(1)	014244	110337	014310		MOVB R3,8\$::SAVE FOR TYPING
(1)	014250	104401	014310		TYPE 8\$::GO TYPE THIS DIGIT
(1)	014254	105337	014312	7\$:	DECB \$OCNT	::COUNT BY 1
(1)	014260	003347			BGT 2\$::BR IF MORE TO DO
(1)	014262	002402			BLT 6\$::BR IF DONE
(1)	014264	005204			INC R4	::INSURE LAST DIGIT ISN'T A BLANK
(1)	014266	000744			BR 2\$::GO DO THE LAST DIGIT
(1)	014270	012605		6\$:	MOV (SP)+,R5	::RESTORE R5
(1)	014272	012604			MOV (SP)+,R4	::RESTORE R4
(1)	014274	012603			MOV (SP)+,R3	::RESTORE R3
(1)	014276	016666	000002 000004		MOV 2(SP),4(SP)	::SET THE STACK FOR RETURNING
(1)	014304	012616			MOV (SP)+,(SP)	

(1) 014306 000002
 (1) 014310 000
 (1) 014311 000
 (1) 014312 000
 (1) 014313 000
 (1) 014314 000000

2683

```

RTI                ::RETURN
8$: .BYTE 0        ::STORAGE FOR ASCII DIGIT
   .BYTE 0        ::TERMINATOR FOR TYPE ROUTINE
$OCNT: .BYTE 0    ::OCTAL DIGIT COUNTER
$OFILL: .BYTE 0   ::ZERO FILL SWITCH
$OMODE: .WORD 0   ::NUMBER OF DIGITS TO TYPE
.SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
  
```

(1)
 (2)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.
*CALL:
  
```

```

*      MOV    NUM,-(SP)  ::PUT THE BINARY NUMBER ON THE STACK
*      TYPDS  ::GO TO THE ROUTINE
  
```

(1) 014316
 (3) 014316 010046
 (3) 014320 010146
 (3) 014322 010246
 (3) 014324 010346
 (3) 014326 010546
 (1) 014330 012746 020200
 (1) 014334 016605 000020
 (1) 014340 100004
 (1) 014342 005405
 (1) 014344 112766 000055 000001
 (1) 014352 005000 1\$:
 (1) 014354 012703 014532
 (1) 014360 112723 000040 2\$:
 (1) 014364 005002
 (1) 014366 016001 014522 3\$:
 (1) 014372 160105
 (1) 014374 002402
 (1) 014376 005202
 (1) 014400 000774
 (1) 014402 060105 4\$:
 (1) 014404 005702
 (1) 014406 001002
 (1) 014410 105716
 (1) 014412 100407
 (1) 014414 106316 5\$:
 (1) 014416 103003
 (1) 014420 116663 000001 177777
 (1) 014426 052702 000060 6\$:
 (1) 014432 052702 000040 7\$:
 (1) 014436 110223
 (1) 014440 005720
 (1) 014442 020027 000010
 (1) 014446 002746
 (1) 014450 003002
 (1) 014452 010502
 (1) 014454 000764
 (1) 014456 105726

```

$TYPDS:
MOV    R0,-(SP)      ::PUSH R0 ON STACK
MOV    R1,-(SP)      ::PUSH R1 ON STACK
MOV    R2,-(SP)      ::PUSH R2 ON STACK
MOV    R3,-(SP)      ::PUSH R3 ON STACK
MOV    R5,-(SP)      ::PUSH R5 ON STACK
MOV    #20200,-(SP)  ::SET BLANK SWITCH AND SIGN
MOV    20(SP),R5     ::GET THE INPUT NUMBER
BPL    1$            ::BR IF INPUT IS POS.
NEG    R5            ::MAKE THE BINARY NUMBER POS.
MOVB   #'-,1(SP)     ::MAKE THE ASCII NUMBER NEG.
1$:   CLR    R0      ::ZERO THE CONSTANTS INDEX
      MOV   #SDBLK,R3  ::SETUP THE OUTPUT POINTER
      MOVB #'',(R3)+  ::SET THE FIRST CHARACTER TO A BLANK
2$:   CLR    R2      ::CLEAR THE BCD NUMBER
      MOV   $DTBL(R0),R1  ::GET THE CONSTANT
3$:   SUB    R1,R5   ::FORM THIS BCD DIGIT
      BLT   4$      ::BR IF DONE
      INC  R2      ::INCREASE THE BCD DIGIT BY 1
4$:   ADD    R1,R5   ::ADD BACK THE CONSTANT
      TST   R2      ::CHECK IF BCD DIGIT=0
      BNE   5$      ::FALL THROUGH IF 0
      TSTB (SP)     ::STILL DOING LEADING 0'S?
      BMI   7$      ::BR IF YES
5$:   ASLB  (SP)     ::MSD?
      BCC  6$      ::BR IF NO
      MOVB 1(SP),-1(R3)  ::YES--SET THE SIGN
6$:   BIS   #'0,R2  ::MAKE THE BCD DIGIT ASCII
7$:   BIS   #' ,R2  ::MAKE IT A SPACE IF NOT ALREADY A DIGIT
      MOVB R2,(R3)+  ::PUT THIS CHARACTER IN THE OUTPUT BUFFER
      TST  (R0)+    ::JUST INCREMENTING
      CMP  R0,#10   ::CHECK THE TABLE INDEX
      BLT  2$      ::GO DO THE NEXT DIGIT
      BGT  8$      ::GO TO EXIT
      MOV  R5,R2    ::GET THE LSD
      BR   6$      ::GO CHANGE TO ASCII
8$:   TSTB (SP)+    ::WAS THE LSD THE FIRST NON-ZERO?
  
```

```
(1) 014460 100003
(1) 014462 116663 177777 177776
(1) 014470 105013
(3) 014472 012605
(3) 014474 012603
(3) 014476 012602
(3) 014500 012601
(3) 014502 012600
(1) 014504 104401 014532
(1) 014510 016666 000002 000004
(1) 014516 012616
(1) 014520 000002
(1) 014522 023420
(1) 014524 001750
(1) 014526 000144
(1) 014530 000012
(1) 014532 000004

          BPL      9$
          MOVB    -1(SP),-2(R3)
          CLRB    (R3)
          MOV     (SP)+,R5
          MOV     (SP)+,R3
          MOV     (SP)+,R2
          MOV     (SP)+,R1
          MOV     (SP)+,R0
          TYPE    $DBLK
          MOV     2(SP),4(SP)
          MOV     (SP)+,(SP)
          RTI
          $DTBL: 10000.
                1000.
                100.
                10.
          $DBLK: .BLKW 4

          ;;BR IF NO
          ;;YES--SET THE SIGN FOR TYPING
          ;;SET THE TERMINATOR
          ;;POP STACK INTO R5
          ;;POP STACK INTO R3
          ;;POP STACK INTO R2
          ;;POP STACK INTO R1
          ;;POP STACK INTO R0
          ;;NOW TYPE THE NUMBER
          ;;ADJUST THE STACK
          ;;RETURN TO USER
```

2684

2686
 2687

.SBTTL TTY INPUT ROUTINE

```

(1)
(2)
(1)
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 014542 022737 000176 001040 $CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
(1) 014550 001074 BNE 15$ ;;BRANCH IF NO
(1) 014552 105777 164266 TSTB @STKS ;;CHAR THERE?
(1) 014556 100071 BPL 15$ ;;IF NO, DON'T WAIT AROUND
(1) 014560 117746 164262 MOVB @STKB,-(SP) ;;SAVE THE CHAR
(1) 014564 042716 177600 BIC #^C177,(SP) ;;STRIP-OFF THE ASCII
(1) 014570 022726 000007 CMP #7,(SP)+ ;;IS IT A CONTROL G?
(1) 014574 001062 BNE 15$ ;;NO, RETURN TO USER
(1) 014576 123727 001034 000001 CMPB $AUTOB,#1 ;;ARE WE RUNNING IN AUTO-MODE?
(1) 014604 001456 BEQ 15$ ;;BRANCH IF YES
(1)
(1) 014606 104401 015277 TYPE , $CNTLG ;;ECHO THE CONTROL-G (^G)
(1) 014612 104401 015304 $GTSWR: TYPE , $MSWR ;;TYPE CURRENT CONTENTS
(2) 014616 013746 000176 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT
(2) 014622 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 014624 104401 015315 TYPE , $MNEW ;;PROMPT FOR NEW SWR
(1) 014630 005046 19$: CLR -(SP) ;;CLEAR COUNTER
(1) 014632 005046 CLR -(SP) ;;THE NEW SWR
(1) 014634 105777 164204 7$: TSTB @STKS ;;CHAR THERE?
(1) 014640 100375 BPL 7$ ;;IF NOT TRY AGAIN
(1)
(1) 014642 117746 164200 MOVB @STKB,-(SP) ;;PICK UP CHAR
(1) 014646 042716 177600 BIC #^C177,(SP) ;;MAKE IT 7-BIT ASCII
(1)
(1)
(1)
(1) 014652 021627 000025 9$: CMP (SP),#25 ;;IS IT A CONTROL-U?
(1) 014656 001005 BNE 10$ ;;BRANCH IF NOT
(1) 014660 104401 015272 TYPE , $CNTLU ;;YES, ECHO CONTROL-U (^U)
(1) 014664 062706 000006 20$: ADD #6,SP ;;IGNORE PREVIOUS INPUT
(1) 014670 C00757 BR 19$ ;;LET'S TRY IT AGAIN
(1)
(1)
(1) 014672 021627 000015 10$: CMP (SP),#15 ;;IS IT A <CR>?
(1) 014676 001022 BNE 16$ ;;BRANCH IF NO
(1) 014700 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
(1) 014704 001403 BEQ 11$ ;;BRANCH IF YES
(1) 014706 016677 000002 164124 MOV 2(SP),@SWR ;;SAVE NEW SWR
(1) 014714 062706 000006 11$: ADD #6,SP ;;CLEAR UP STACK
(1) 014720 104401 001075 14$: TYPE , $CRLF ;;ECHO <CR> AND <LF>
(1) 014724 123727 001035 000001 CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?
(1) 014732 001003 BNE 15$ ;;BRANCH IF NOT
(1) 014734 012777 000100 164102 MOV #100,@STKS ;;RE-ENABLE TTY KBD INTERRUPTS
(1) 014742 000002 15$: RTI ;;RETURN
(1) 014744 004737 013746 16$: JSR PC,$TYPEC ;;ECHO CHAR

```


(1) 014750 021627 000060
(1) 014754 002420
(1) 014756 021627 000067
(1) 014762 003015
(1) 014764 042726 000060
(1) 014770 005766 000002
(1) 014774 001403
(1) 014776 006316
(1) 015000 006316
(1) 015002 006316
(1) 015004 005266 000002
(1) 015010 056616 177776
(1) 015014 000707
(1) 015016 104401 001074
(1) 015022 000720

CMP (SP),#60 ::CHAR < 0?
BLT 18\$::BRANCH IF YES
CMP (SP),#67 ::CHAR > 7?
BGT 18\$::BRANCH IF YES
BIC #60,(SP)+ ::STRIP-OFF ASCII
TST 2(SP) ::IS THIS THE FIRST CHAR
BEQ 17\$::BRANCH IF YES
ASL (SP) ::NO, SHIFT PRESENT
ASL (SP) :: CHAR OVER TO MAKE
ASL (SP) :: ROOM FOR NEW ONE.
17\$: INC 2(SP) ::KEEP COUNT OF CHAR
BIS -2(SP),(SP) ::SET IN NEW CHAR
BR 7\$::GET THE NEXT ONE
18\$: TYPE \$QUES ::TYPE ?<CR><LF>
BR 20\$::SIMULATE CONTROL-U
.DSABL _SB

(1)
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)

*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
* RDCHR ::INPUT A SINGLE CHARACTER FROM THE TTY
* RETURN HERE ::CHARACTER IS ON THE STACK
* ::WITH PARITY BIT STRIPPED OFF
*

(1) 015024 011646
(1) 015026 016666 000004 000002
(1) 015034 105777 164004
(1) 015040 100375
(1) 015042 117766 164000 000004
(1) 015050 042766 177600 000004
(1) 015056 026627 000004 000023
(1) 015064 001013
(1) 015066 105777 163752
(1) 015072 100375
(1) 015074 117746 163746
(1) 015100 042716 177600
(1) 015104 022627 000021
(1) 015110 001366
(1) 015112 000750
(1) 015114 026627 000004 000021
(1) 015122 001744
(1) 015124 026627 000004 000140
(1) 015132 002407
(1) 015134 026627 000004 000175
(1) 015142 003003
(1) 015144 042766 000040 000004
(1) 015152 000002

\$RDCHR: MOV (SP),-(SP) ::PUSH DOWN THE PC
MOV 4(SP),2(SP) ::SAVE THE PS
1\$: TST @STKS ::WAIT FOR
BPL 1\$::A CHARACTER
MOVB @STKB,4(SP) ::READ THE TTY
BIC #^C<177>,4(SP) ::GET RID OF JUNK IF ANY
CMP 4(SP),#23 ::IS IT A CONTROL-S?
BNE 3\$::BRANCH IF NO
2\$: TST @STKS ::WAIT FOR A CHARACTER
BPL 2\$::LOOP UNTIL ITS THERE
MOVB @STKB,-(SP) ::GET CHARACTER
BIC #^C177,(SP) ::MAKE IT 7-BIT ASCII
CMP (SP)+,#21 ::IS IT A CONTROL-Q?
BNE 2\$::IF NOT DISCARD IT
BR 1\$::YES, RESUME
3\$: CMP 4(SP),#\$XON ::IS IT A RANDOM XON? ;RAN001
BEQ 1\$::BRANCH IF YES ;RAN001
CMP 4(SP),#140 ::IS IT UPPER CASE?
BLT 4\$::BRANCH IF YES
CMP 4(SP),#175 ::IS IT A SPECIAL CHAR?
BGT 4\$::BRANCH IF YES
BIC #40,4(SP) ::MAKE IT UPPER CASE
4\$: RTI ::GO BACK TO USER

(2)
(1)
(1)
(1)
(1)
(1)
(1)

*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
*CALL:
* RDLIN ::INPUT A STRING FROM THE TTY
* RETURN HERE ::ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
* ::TERMINATOR WILL BE A BYTE OF ALL 0'S
*

(1)	015154	010346			\$RDLIN: MOV	R3,-(SP)	::SAVE R3
(1)	015156	012703	015262		1\$: MOV	#\$TTYIN,R3	::GET ADDRESS
(1)	015162	022703	015272		2\$: CMP	#\$TTYIN+8.,R3	::BUFFER FULL?
(1)	015166	101405				4\$::BR IF YES
(1)	015170	104410			BLOS		::GO READ ONE CHARACTER FROM THE TTY
(1)	015172	112613			RDCHR		::GET CHARACTER
(1)	015174	122713	000177		10\$: MOV	(SP)+,(R3)	::IS IT A RUBOUT
(1)	015200	001003			CMPB	#177,(R3)	::SKIP IF NOT
(1)	015202	104401	001074		BNE	3\$::TYPE A '?'
(1)	015206	000763			4\$: TYPE	\$QUES	::CLEAR THE BUFFER AND LOOP
(1)	015210	111337	015260		BR	1\$::ECHO THE CHARACTER
(1)	015214	104401	015260		3\$: MOV	(R3),9\$	
(1)	015220	122723	000015		TYPE	.9\$	
(1)	015224	001356			CMPB	#15,(R3)+	::CHECK FOR RETURN
(1)	015226	105063	177777		BNE	2\$::LOOP IF NOT RETURN
(1)	015232	104401	001076		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
(1)	015236	012603			TYPE	.5LF	::TYPE A LINE FEED
(1)	015240	011646			MOV	(SP)+,R3	::RESTORE R3
(1)	015242	016666	000004	000002	MOV	(SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
(1)	015250	012766	015262	000004	MOV	4(SP),2(SP)	:: FIRST ASCII CHARACTER ON IT
(1)	015256	000002			MOV	#\$TTYIN,4(SP)	
(1)	015260	000			RTI		::RETURN
(1)	015261	000			9\$: .BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
(1)	015262	000010			.BYTE	0	::TERMINATOR
(1)	015272	052536	005015	000	\$TTYIN: .BLKB	8.	::RESERVE 8 BYTES FOR TTY INPUT
(1)	015277	136	006507	000012	\$CNTLU: .ASCIZ	/^U/<15><12>	::CONTROL 'U'
(1)	015304	005015	053523	020122	\$CNTLG: .ASCIZ	/^G/<15><12>	::CONTROL 'G'
(1)	015312	020075	000		\$MSWR: .ASCIZ	<15><12>/SWR = /	
(1)	015315	040	047040	053505	\$MNEW: .ASCIZ	/ NEW = /	
(1)	015322	036440	000040				


```

2695          .SBTTL ECHO TEST
2696          ;;*****
2697          :*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
2698          :*ON EITHER SLU1 OR SLU2. DEFAULT IS TO THE CONSOLE DEVICE SLU1.
2699          :*THE TEST IS HALTED BY TYPING A CONTROL-C
2700          :*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
2701          ;;*****
2702          ECHO:  MOV   #SWREG,SWR           ;SET UP FOR SOFTWARE SWITCH REGISTER
2703                MOV   #DISPREG,DISPLAY      ;AND DISPLAY REGISTER
2704                BIT   #BIT4,@SWR           ;CHECK IF BIT4 SET IN SWITCH REG
2705                BEQ   1$                   ;IF NOT THEN SELECT SLU1
2706                MOV   #RCSR,R3             ;IF BIT4 SET THEN SELECT SLU2
2707                BR    2$
2708          1$:  MOV   #CRCSR,R3             ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
2709          2$:  RESET                          ;CLEAR EVERYTHING
2710                MOV  #'*,@6(R3)           ;TRANSMIT PROMPT '*'
2711          3$:  TSTB  @4(R3)                ;WAIT FOR INPUT
2712                BPL   3$
2713                MOV  @2(R3),@6(R3)        ;ECHO INPUT
2714                MOV  @2(R3),R0           ;STORE INPUT
2715                BPL   6$                   ;BR IF 'ERROR' NOT SET
2716                BIS  #BIT12,R1          ;SET PARITY ERROR TEST MASK
2717                BIT  R1,R0               ;CHECK FOR PARITY ERROR FLAG
2718                BEQ  4$                   ;BR IF NOT SET
2719                JSR  PC,MSG              ;REPORT PARITY ERROR
2720                MPAR
2721          4$:  ASL   R1                       ;SHIFT MASK TO TEST 'FR' FLAG
2722                BIT  R1,R0               ;TEST FOR FRAMING ERROR FLAG
2723                BEQ  5$                   ;BR IF NOT SET
2724                JSR  PC,MSG              ;REPSORT FRAMING ERROR
2725                MFR
2726          5$:  ASL   R1                       ;SHIFT MASK TO TEST 'OR' FLAG
2727                BIT  R1,R0               ;TEST FOR OVERFLOW ERROR
2728                BEQ  6$                   ;BR IF NOT SET
2729                JSR  PC,MSG              ;REPORT OVERFLOW ERROR
2730                MOR
2731          6$:  BIC  #BIT7,R0                ;CLEAR ANY PARITY BIT
2732                CMP  #3,R0                ;WAS INPUT CONTROL-C
2733                BNE  3$                   ;BR IF IS NOT
2734                JSR  PC,MSG              ;REPORT PROGRAM STOP
2735                MSTOP
2736                HALT
2737                BR   ECHO                 ;END OF TEST HALT
2738                ; AFTER END OF TEST HALT
2739                ; PRESS CONTINUE TO RESTART ECHO TEST
2740          MSG:  MOV   @4(R3),R0            ;PICK UP MESSAGE POINTER
2741                ADD  #2,-(SP)             ;ADJUST RETURN PC
2742          WAIT: TSTB  @4(R3)                ;WAIT FOR XMIT DONE
2743                BPL   WAIT
2744                MOV  (R0)+,@6(R3)        ;SEND CHARACTER
2745                TSTB (R0)                 ;IS THIS END OF MESSAGE?
2746                BNE  WAIT                 ;BR IF NOT
2747                RTS   PC                  ;RETURN
2748
2749          015616 005015 040520 044522 MPAR: .ASCIZ <CR><LF>/PARITY/
          015624 054524 000
  
```

CJKDIAO 11/23B SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82^{M 6} 16:42 PAGE 19-1
ECHO TEST

SEQ 0077

2750	015627	015	043012	040522	MFR:	.ASCIZ	<CR><LF>/FRAMING/
	015634	044515	043516	000			
2751	015641	015	047412	042526	MOR:	.ASCIZ	<CR><LF>/OVERFLOW/
	015646	043122	047514	000127			
2752	015654	005015	052123	050117	MSTOP:	.ASCIZ	<CR><LF>/STOP/
	015662	000					

2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795

015664

.EVEN

.SBTTL TERMINAL OUTPUT TEST

```
::*****  
: *THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE  
: *THE OCTAL CODE 040 --> 377  
: *32 CHARACTERS ARE PRINTED ON EACH LINE  
: *THE PATTERN IS REPEATED EVERY THREE LINES  
: *  
:*****
```

```
OUTTST: MOV #SWREG,SWR ;SET UP FOR SOFTWARE SWITCH REGISTER  
MOV #DISPREG,DISPLAY ;AND DISPLAY REGISTER  
BIT #BIT4,@SWR ;CHECK IF BIT4 SET IN SWITCH REG  
BEQ 1$ ;IF NOT THEN SELECT SLU1  
MOV #RCSR,R3 ;IF BIT4 SET THEN SELECT SLU2  
BR 2$  
1$: MOV #CRCR,R3 ;SELECT SLU1 (THIS IS DEFAULT DEVICE)  
2$: RESET ;CLEAR THE WORLD  
3$: MOV #40,R1 ;LOAD FIRST WRITABLE CHARACTER  
4$: MOV #40,R0 ;LOAD CHAR COUNT PER LINE  
5$: TSTB @4(R3) ;WAIT FOR DONE  
BPL 5$  
MOV R1,@6(R3) ;TRANSMIT A CHARACTER  
INCB R1 ;INCREMENT CHARACTER CODE  
DEC R0 ;DECREMENT CHAR COUNT  
BNE 5$ ;BR IF LINE NOT COMPLETE  
JSR PC,MSG ;SSUE CR,LINE FEED  
  
$CRLF  
TSTB @4(R3) ;ANY CHARACTER RECEIVED?  
BMI 6$ ;BR IF YES  
BIT #BIT7,R1 ;FINISHED ONE PASS OF WRITABLE CHARACTERS?  
BNE 3$ ;BR IF YES  
BR 4$ ;IF NOT WRITE NEXT LINE  
  
6$: CLR @2(R3) ;CLEAR RECEIVER  
HALT ;STOP TEST  
BR OUTTST ;RESTART TEST IF CONTINUED
```

2797
2798
2799 016010 046123 030525 052040 EM5: .ASCIZ /SLU1 TCSR DONE NOT SET WITH RESET/
016016 051503 020122 047504
016024 042516 047040 052117
016032 051440 052105 053440
016040 052111 020110 042522
2800 016046 042523 000124
016052 046123 030525 051040 EM6: .ASCIZ /SLU1 RCSR DID NOT RETURN SSYNC/
016060 051503 020122 044504
016066 020104 047516 020124
016074 042522 052524 047122
016102 051440 054523 041516
016110 000
2801 016111 123 052514 020061 EM7: .ASCIZ /SLU1 RBUF DID NOT RETURN SSYNC/
016116 041122 043125 042040
016124 042111 047040 052117
016132 051040 052105 051125
016140 020116 051523 047131
016146 000103
2802 016150 040503 020116 047516 EM11: .ASCIZ /CAN NOT SET BIT2 OF SLU1 TCSR/
016156 020124 042523 020124
016164 044502 031124 047440
016172 020106 046123 030525
016200 052040 051503 000122
2803 016206 042522 042523 020124 EM13: .ASCIZ /RESET DID NOT CLEAR BIT2 OF SLU1 TCSR/
016214 044504 020104 047516
016222 020124 046103 040505
016230 020122 044502 031124
016236 047440 020106 046123
016244 030525 052040 051503
016252 000122
2804 016254 044502 033124 047440 EM14: .ASCIZ /BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET/
016262 020106 046123 030525
016270 052040 051503 020122
016276 047516 020124 046103
016304 040505 020122 043101
016312 042524 020122 042522
016320 042523 000124
2805 016324 046123 030525 054040 EM15: .ASCIZ /SLU1 XMIT INTERRUPTED WITH PRIORITY 7/
016332 044515 020124 047111
016340 042524 051122 050125
016346 042524 020104 044527
016354 044124 050040 044522
016362 051117 052111 020131
016370 000067
2806 016372 040503 020116 047516 EM16: .ASCIZ /CAN NOT SET BIT6 OF SLU1 TCSR/
016400 020124 042523 020124
016406 044502 033124 047440
016414 020106 046123 030525
016422 052040 051503 000122
2807 016430 040503 020116 047516 EM17: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU1 TCSR/
016436 020124 046103 040505
016444 020122 044502 033124
016452 047440 020106 046123
016460 030525 052040 051503

2808	016466	000122									
	016470	042522	042523	020124	EM20:	.ASCIZ	/RESET DID NOT CLEAR BIT6 OF SLU1 TCSR/				
	016476	044504	020104	047516							
	016504	020124	046103	040505							
	016512	020122	044502	033124							
	016520	047440	020106	046123							
	016526	030525	052040	051503							
	016534	000122									
2809	016536	044502	033124	047440	EM21:	.ASCIZ	/BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET/				
	016544	020106	046123	030525							
	016552	051040	051503	020122							
	016560	047516	020124	046103							
	016566	040505	020122	043101							
	016574	042524	020122	042522							
	016602	042523	000124								
2810	016606	046123	030525	051040	EM22:	.ASCIZ	/SLU1 RCVR INTERRUPT WITH PRIORITY 7/				
	016614	053103	020122	047111							
	016622	042524	051122	050125							
	016630	020124	044527	044124							
	016636	050040	044522	051117							
	016644	052111	020131	000067							
2811	016652	040503	020116	047516	EM23:	.ASCIZ	/CAN NOT SET BIT6 OF SLU1 RCSR/				
	016660	020124	042523	020124							
	016666	044502	033124	047440							
	016674	020106	046123	030525							
	016702	051040	051503	000122							
2812	016710	040503	020116	047516	EM24:	.ASCIZ	/CAN NOT CLEAR BIT6 OF SLU1 RCSR/				
	016716	020124	046103	040505							
	016724	020122	044502	033124							
	016732	047440	020106	046123							
	016740	030525	051040	051503							
	016746	000122									
2813	016750	040503	020116	047516	EM25:	.ASCIZ	/CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET/				
	016756	020124	046103	040505							
	016764	020122	044502	033124							
	016772	047440	020106	046123							
	017000	030525	051040	051503							
	017006	020122	044527	044124							
	017014	051040	051505	052105							
	017022	000									
2814	017023	123	052514	020061	EM26:	.ASCIZ	/SLU1 RECEIVER DONE NEVER SET/				
	017030	042522	042503	053111							
	017036	051105	042040	047117							
	017044	020105	042516	042526							
	017052	020122	042523	000124							
2815	017060	042522	042523	020124	EM27:	.ASCIZ	/RESET DID NOT CLEAR SLU1 RCVR DONE/				
	017066	044504	020104	047516							
	017074	020124	046103	040505							
	017102	020122	046123	030525							
	017110	051040	053103	020122							
	017116	047504	042516	000							
2816	017123	122	040505	044504	EM30:	.ASCIZ	/READING SLU1 RBUF DID NOT CLEAR RCVR DONE/				
	017130	043516	051440	052514							
	017136	020061	041122	043125							
	017144	042040	042111	047040							
	017152	052117	041440	042514							

	017160	051101	051040	053103		
	017166	020122	047504	042516		
	017174	000				
2817	017175	123	052514	020062	EM31:	.ASCIZ /SLU2 TCSR DID NOT RETURN SSYNC/
	017202	041524	051123	042040		
	017210	042111	047040	052117		
	017216	051040	052105	051125		
	017224	020116	051523	047131		
	017232	000103				
2818	017234	046123	031125	052040	EM32:	.ASCIZ /SLU2 TBUF DID NOT RETURN SSYNC/
	017242	052502	020106	044504		
	017250	020104	047516	020124		
	017256	042522	052524	047122		
	017264	051440	054523	041516		
	017272	000				
2819	017273	123	052514	020062	EM33:	.ASCIZ /SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL/
	017300	041524	051123	042040		
	017306	047117	020105	047516		
	017314	020124	046103	040505		
	017322	042522	020104	044527		
	017330	044124	052040	052502		
	017336	020106	052504	046114		
	017344	000				
2820	017345	123	052514	020062	EM34:	.ASCIZ /SLU2 TCSR DONE NOT SET AFTER TRANSMIT/
	017352	041524	051123	042040		
	017360	047117	020105	047516		
	017366	020124	042523	020124		
	017374	043101	042524	020122		
	017402	051124	047101	046523		
	017410	052111	000			
2821	017413	123	052514	020062	EM35:	.ASCIZ /SLU2 TCSR DONE NOT SET WITH RESET/
	017420	041524	051123	042040		
	017426	047117	020105	047516		
	017434	020124	042523	020124		
	017442	044527	044124	051040		
	017450	051505	052105	000		
2822	017455	123	052514	020062	EM36:	.ASCIZ /SLU2 RCSR DID NOT RETURN SSYNC/
	017462	041522	051123	042040		
	017470	042111	047040	052117		
	017476	051040	052105	051125		
	017504	020116	051523	047131		
	017512	000103				
2823	017514	046123	031125	051040	EM37:	.ASCIZ /SLU2 RBUF DID NOT RETURN SSYNC/
	017522	052502	020106	044504		
	017530	020104	047516	020124		
	017536	042522	052524	047122		
	017544	051440	054523	041516		
	017552	000				
2824	017553	102	052111	020060	EM40:	.ASCIZ /BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET/
	017560	043117	051440	052514		
	017566	020062	041524	051123		
	017574	047040	052117	041440		
	017602	042514	051101	040440		
	017610	052106	051105	051040		
	017616	051505	052105	000		
2825	017623	103	047101	047040	EM41:	.ASCIZ /CAN NOT SET BIT0 OF SLU2 TCSR/

	017630	052117	051440	052105		
	017636	041040	052111	020060		
	017644	043117	051440	052514		
	017652	020062	041524	051123		
	017660	000				
2826	017661	103	047101	047040	EM42:	.ASCIZ /CAN NOT CLEAR BIT0 OF SLU2 TCSR/
	017666	052117	041440	042514		
	017674	051101	041040	052111		
	017702	020060	043117	051440		
	017710	052514	020062	041524		
	017716	051123	000			
2827	017721	122	051505	052105	EM43:	.ASCIZ /RESET DID NOT CLEAR BIT0 OF SLU2 TCSR/
	017726	042040	042111	047040		
	017734	052117	041440	042514		
	017742	051101	041040	052111		
	017750	020060	043117	051440		
	017756	052514	020062	041524		
	017764	051123	000			
2828	017767	102	052111	020066	EM44:	.ASCIZ /BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET/
	017774	043117	051440	052514		
	020002	020062	041524	051123		
	020010	047040	052117	041440		
	020016	042514	051101	040440		
	020024	052106	051105	051040		
	020032	051505	052105	000		
2829	020037	123	052514	020062	EM45:	.ASCIZ /SLU2 XMIT INTERRUPTED WITH PRIORITY 7/
	020044	046530	052111	044440		
	020052	052116	051105	052522		
	020060	052120	042105	053440		
	020066	052111	020110	051120		
	020074	047511	044522	054524		
	020102	033440	000			
2830	020105	103	047101	047040	EM46:	.ASCIZ /CAN NOT SET BIT6 OF SLU2 TCSR/
	020112	052117	051440	052105		
	020120	041040	052111	020066		
	020126	043117	051440	052514		
	020134	020062	041524	051123		
	020142	000				
2831	020143	103	047101	047040	EM47:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 TCSR/
	020150	052117	041440	042514		
	020156	051101	041040	052111		
	020164	020066	043117	051440		
	020172	052514	020062	041524		
	020200	051123	000			
2832	020203	122	051505	052105	EM50:	.ASCIZ /RESET DID NOT CLEAR BIT6 OF SLU2 TCSR/
	020210	042040	042111	047040		
	020216	052117	041440	042514		
	020224	051101	041040	052111		
	020232	020066	043117	051440		
	020240	052514	020062	041524		
	020246	051123	000			
2833	020251	102	052111	020066	EM51:	.ASCIZ /BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET/
	020256	043117	051440	052514		
	020264	020062	041522	051123		
	020272	047040	052117	041440		
	020300	042514	051101	040440		

	020306	052106	051105	051040		
	020314	051505	052105	000		
2834	020321	123	052514	020062	EM52:	.ASCIZ /SLU2 RCVR INTERRUPT WITH PRIORITY 7/
	020326	041522	051126	044440		
	020334	052116	051105	052522		
	020342	052120	053440	052111		
	020350	020110	051120	047511		
	020356	044522	054524	033440		
	020364	000				
2835	020365	103	047101	047040	EM53:	.ASCIZ /CAN NOT SET BIT6 OF SLU2 RCSR/
	020372	052117	051440	052105		
	020400	041040	052111	020066		
	020406	043117	051440	052514		
	020414	020062	041522	051123		
	020422	000				
2836	020423	103	047101	047040	EM54:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR/
	020430	052117	041440	042514		
	020436	051101	041040	052111		
	020444	020066	043117	051440		
	020452	052514	020062	041522		
	020460	051123	000			
2837	020463	103	047101	047040	EM55:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET/
	020470	052117	041440	042514		
	020476	051101	041040	052111		
	020504	020066	043117	051440		
	020512	052514	020062	041522		
	020520	051123	053440	052111		
	020526	020110	042522	042523		
	020534	000124				
2838	020536	046123	031125	051040	EM56:	.ASCIZ /SLU2 RECEIVER DONE NEVER SET/
	020544	041505	044505	042526		
	020552	020122	047504	042516		
	020560	047040	053105	051105		
	020566	051440	052105	000		
2839	020573	122	051505	052105	EM57:	.ASCIZ /RESET DID NOT CLEAR SLU2 RCVR DONE/
	020600	042040	042111	047040		
	020606	052117	041440	042514		
	020614	051101	051440	052514		
	020622	020062	041522	051126		
	020630	042040	047117	000105		
2840	020636	042522	042101	047111	EM60:	.ASCIZ /READING SLU2 RBUF DID NOT CLEAR RCVR DONE/
	020644	020107	046123	031125		
	020652	051040	052502	020106		
	020660	044504	020104	047516		
	020666	020124	046103	040505		
	020674	020122	041522	051126		
	020702	042040	047117	000105		
2841	020710	045514	020123	044504	EM61:	.ASCIZ /LKS DID NOT RETURN SSYNC/
	020716	020104	047516	020124		
	020724	042522	052524	047122		
	020732	051440	054523	041516		
	020740	000				
2842	020741	102	052111	020066	EM62:	.ASCIZ /BIT6 OF LKS NOT CLEAR AFTER RESET/
	020746	043117	046040	051513		
	020754	047040	052117	041440		
	020762	042514	051101	040440		

	020770	052106	051105	051040		
	020776	051505	052105	000		
2843	021003	114	051513	044440	EM63:	.ASCIZ /LKS INTERRUPT WITH PRIORITY 7/
	021010	052116	051105	052522		
	021016	052120	053440	052111		
	021024	020110	051120	047511		
	021032	044522	054524	033440		
	021040	000				
2844	021041	103	047101	047040	EM64:	.ASCIZ /CAN NOT SET BIT6 OF LKS/
	021046	052117	051440	052105		
	021054	041040	052111	020066		
	021062	043117	046040	051513		
	021070	000				
2845	021071	103	047101	047040	EM65:	.ASCIZ /CAN NOT CLEAR BIT6 OF LKS/
	021076	052117	041440	042514		
	021104	051101	041040	052111		
	021112	020066	043117	046040		
	021120	051513	000			
2846	021123	122	051505	052105	EM66:	.ASCIZ /RESET DID NOT CLEAR BIT6 OF LKS/
	021130	042040	042111	047040		
	021136	052117	041440	042514		
	021144	051101	041040	052111		
	021152	020066	043117	046040		
	021160	051513	000			
2847	021163	102	052111	020067	EM67:	.ASCIZ /BIT7 OF LKS NOT SET AFTER RESET/
	021170	043117	046040	051513		
	021176	047040	052117	051440		
	021204	052105	040440	052106		
	021212	051105	051040	051505		
	021220	052105	000			
2848	021223	103	047101	047040	EM70:	.ASCIZ /CAN NOT CLEAR BIT7 OF LKS/
	021230	052117	041440	042514		
	021236	051101	041040	052111		
	021244	020067	043117	046040		
	021252	051513	000			
2849	021255	102	052111	020067	EM71:	.ASCIZ /BIT7 OF LKS DOES NOT SET/
	021262	043117	046040	051513		
	021270	042040	042517	020123		
	021276	047516	020124	042523		
	021304	000124				
2850	021306	051127	052111	047111	EM72:	.ASCIZ /WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER/
	021314	020107	047524	047440		
	021322	042516	044440	052116		
	021330	051105	040516	020114		
	021336	042101	051104	051505		
	021344	020123	047515	044504		
	021352	044506	042105	040440		
	021360	047516	044124	051105		
	021366	000				
2851	021367	123	052514	020061	EM74:	.ASCIZ /SLU1 XMIT INTERRUPTS WHEN DISABLED/
	021374	046530	052111	044440		
	021402	052116	051105	052522		
	021410	052120	020123	044127		
	021416	047105	042040	051511		
	021424	041101	042514	000104		
2852	021432	046123	030525	054040	EM75:	.ASCIZ /SLU1 XMIT DID NOT INTERRUPT/

	021440	044515	020124	044504		
	021446	020104	047516	020124		
	021454	047111	042524	051122		
	021462	050125	000124			
2853	021466	046123	030525	054040	EM76:	.ASCIZ /SLU1 XMIT INTERRUPT AT PRIORITY 7/
	021474	044515	020124	047111		
	021502	042524	051122	050125		
	021510	020124	052101	050040		
	021516	044522	051117	052111		
	021524	020131	000067			
2854	021530	046123	030525	054040	EM77:	.ASCIZ /SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR/
	021536	044515	020124	047111		
	021544	042524	051122	050125		
	021552	051524	053440	052111		
	021560	020110	047105	041101		
	021566	042514	041440	042514		
	021574	051101	000			
2855	021577	123	052514	020061	EM100:	.ASCIZ /SLU1 XMIT DID NOT INTERRUPT/
	021604	046530	052111	042040		
	021612	042111	047040	052117		
	021620	044440	052116	051105		
	021626	052522	052120	000		
2856	021633	123	052514	020061	EM101:	.ASCIZ /SLU1 XMIT RE-INTERRUPTED/
	021640	046530	052111	051040		
	021646	026505	047111	042524		
	021654	051122	050125	042524		
	021662	000104				
2857	021664	047514	042101	047111	EM102:	.ASCIZ /LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT/
	021672	020107	046123	030525		
	021700	052040	052502	020106		
	021706	044504	020104	047516		
	021714	020124	046103	040505		
	021722	020122	047111	042524		
	021730	051122	050125	000124		
2858	021736	046123	030525	051040	EM103:	.ASCIZ /SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR/
	021744	053103	020122	047111		
	021752	042524	051122	050125		
	021760	051524	053440	052111		
	021766	020110	047105	041101		
	021774	042514	041440	042514		
	022002	051101	000			
2859	022005	123	052514	020061	EM104:	.ASCIZ /SLU1 RCVR DID NOT INTERRUPT/
	022012	041522	051126	042040		
	022020	042111	047040	052117		
	022026	044440	052116	051105		
	022034	052522	052120	000		
2860	022041	123	052514	020061	EM105:	.ASCIZ /SLU1 RCVR INTERRUPTS AT PRIORITY 7/
	022046	041522	051126	044440		
	022054	052116	051105	052522		
	022062	052120	020123	052101		
	022070	050040	044522	051117		
	022076	052111	020131	000067		
2861	022104	046123	030525	051040	EM106:	.ASCIZ /SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
	022112	053103	020122	047111		
	022120	042524	051122	050125		
	022126	051524	053440	052111		

	022134	020110	047111	042524	
	022142	051122	050125	020124	
	022150	047105	041101	042514	
	022156	041440	042514	051101	
	022164	000			
2862	022165	123	052514	020061	EM107: .ASCIZ /SLU1 RCVR DID NOT INTERRUPT/
	022172	041522	051126	042040	
	022200	042111	047040	052117	
	022206	044440	052116	051105	
	022214	052522	052120	000	
2863	022221	123	052514	020061	EM110: .ASCIZ /SLU1 RECEIVER RE-INTERRUPTED/
	022226	042522	042503	053111	
	022234	051105	051040	026505	
	022242	047111	042524	051122	
	022250	050125	042524	000104	
2864	022256	046123	030525	051040	EM111: .ASCIZ /SLU1 READING RBUF DID NOT CLEAR INTERRUPT/
	022264	040505	044504	043516	
	022272	051040	052502	020106	
	022300	044504	020104	047516	
	022306	020124	046103	040505	
	022314	020122	047111	042524	
	022322	051122	050125	000124	
2865	022330	042522	042523	020124	EM112: .ASCIZ /RESET DID NOT CLEAR SLU1 RCVR INTERRUPT/
	022336	044504	020104	047516	
	022344	020124	046103	040505	
	022352	020122	046123	030525	
	022360	051040	053103	020122	
	022366	047111	042524	051122	
	022374	050125	000124		
2866	022400	046123	030525	023440	EM113: .ASCIZ /SLU1 'OR' FLAG DID NOT SET/
	022406	051117	020047	046106	
	022414	043501	042040	042111	
	022422	047040	052117	051440	
	022430	052105	000		
2867	022433	123	052514	020061	EM114: .ASCIZ /SLU1 'ERROR' NOT SET WITH 'OR' FLAG/
	022440	042447	051122	051117	
	022446	020047	047516	020124	
	022454	042523	020124	044527	
	022462	044124	023440	051117	
	022470	020047	046106	043501	
	022476	000			
2868	022477	104	052101	020101	EM115: .ASCIZ /DATA COMPARE ERROR/
	022504	047503	050115	051101	
	022512	020105	051105	047522	
	022520	000122			
2869	022522	046123	031125	054040	EM116: .ASCIZ /SLU2 XMIT INTERRUPTS WHEN DISABLED/
	022530	044515	020124	047111	
	022536	042524	051122	050125	
	022544	051524	053440	042510	
	022552	020116	044504	040523	
	022560	046102	042105	000	
2870	022565	123	052514	020062	EM117: .ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
	022572	046530	052111	042040	
	022600	042111	047040	052117	
	022606	044440	052116	051105	
	022614	052522	052120	000	

2871	022621	123	052514	020062	EM120: .ASCIZ /SLU2 XMIT INTERRUPT AT PRIORITY 7/
	022626	046530	052111	044440	
	022634	052116	051105	052522	
	022642	052120	040440	020124	
	022650	051120	047511	044522	
	022656	054524	033440	000	
2872	022663	123	052514	020062	EM121: .ASCIZ /SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR/
	022670	046530	052111	044440	
	022676	052116	051105	052522	
	022704	052120	020123	044527	
	022712	044124	042440	040516	
	022720	046102	020105	046103	
	022726	040505	000122		
2873	022732	046123	031125	054040	EM122: .ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
	022740	044515	020124	044504	
	022746	020104	047516	020124	
	022754	047111	042524	051122	
	022762	050125	000124		
2874	022766	046123	031125	054040	EM123: .ASCIZ /SLU2 XMIT RE-INTERRUPTED/
	022774	044515	020124	042522	
	023002	044455	052116	051105	
	023010	052522	052120	042105	
	023016	000			
2875	023017	114	040517	044504	EM124: .ASCIZ /LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT/
	023024	043516	051440	052514	
	023032	020062	041124	043125	
	023040	042040	042111	047040	
	023046	052117	041440	042514	
	023054	051101	044440	052116	
	023062	051105	052522	052120	
	023070	000			
2876	023071	123	052514	020062	EM125: .ASCIZ /SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR/
	023076	041522	051126	044440	
	023104	052116	051105	052522	
	023112	052120	020123	044527	
	023120	044124	042440	040516	
	023126	046102	020105	046103	
	023134	040505	000122		
2877	023140	046123	031125	051040	EM126: .ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
	023146	053103	020122	044504	
	023154	020104	047516	020124	
	023162	047111	042524	051122	
	023170	050125	000124		
2878	023174	046123	031125	051040	EM127: .ASCIZ /SLU2 RCVR INTERRUPTS AT PRIORITY 7/
	023202	053103	020122	047111	
	023210	042524	051122	050125	
	023216	051524	040440	020124	
	023224	051120	047511	044522	
	023232	054524	033440	000	
2879	023237	123	052514	020062	EM130: .ASCIZ /SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
	023244	041522	051126	044440	
	023252	052116	051105	052522	
	023260	052120	020123	044527	
	023266	044124	044440	052116	
	023274	051105	052522	052120	
	023302	042440	040516	046102	

	023310	020105	046103	040505		
	023316	000122				
2880	023320	046123	031125	051040	EM131:	.ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
	023326	053103	020122	044504		
	023334	020104	047516	020124		
	023342	047111	042524	051122		
	023350	050125	000124			
2881	023354	046123	031125	051040	EM132:	.ASCIZ /SLU2 RECEIVER RE-INTERRUPTED/
	023362	041505	044505	042526		
	023370	020122	042522	044455		
	023376	052116	051105	052522		
	023404	052120	042105	000		
2882	023411	123	052514	020062	EM133:	.ASCIZ /SLU2 READING RBUF DID NOT CLEAR INTERRUPT/
	023416	042522	042101	047111		
	023424	020107	041122	043125		
	023432	042040	042111	047040		
	023440	052117	041440	042514		
	023446	051101	044440	052116		
	023454	051105	052522	052120		
	023462	000				
2883	023463	122	051505	052105	EM134:	.ASCIZ /RESET DID NOT CLEAR SLU2 RCVR INTERRUPT/
	023470	042040	042111	047040		
	023476	052117	041440	042514		
	023504	051101	051440	052514		
	023512	020062	041522	051126		
	023520	044440	052116	051105		
	023526	052522	052120	000		
2884	023533	123	052514	020062	EM135:	.ASCIZ /SLU2 'OR' FLAG DID NOT SET/
	023540	047447	023522	043040		
	023546	040514	020107	044504		
	023554	020104	047516	020124		
	023562	042523	000124			
2885	023566	046123	031125	023440	EM136:	.ASCIZ /SLU2 'ERROR' NOT SET WITH 'OR' FLAG/
	023574	051105	047522	023522		
	023602	047040	052117	051440		
	023610	052105	053440	052111		
	023616	020110	047447	023522		
	023624	043040	040514	000107		
2886	023632	046123	031125	041040	EM137:	.ASCIZ /SLU2 BREAK DID NOT TRANSMIT ALL ZEROES/
	023640	042522	045501	042040		
	023646	042111	047040	052117		
	023654	052040	040522	051516		
	023662	044515	020124	046101		
	023670	020114	042532	047522		
	023676	051505	000			
2887	023701	102	042522	045501	EM140:	.ASCIZ /BREAK DID NOT SET FRAMING ERROR/
	023706	042040	042111	047040		
	023714	052117	051440	052105		
	023722	043040	040522	044515		
	023730	043516	042440	051122		
	023736	051117	000			
2888	023741	123	052514	020062	EM141:	.ASCIZ /SLU2 'ERROR' NOT SET WITH 'FR' FLAG/
	023746	042447	051122	051117		
	023754	020047	047516	020124		
	023762	042523	020124	044527		
	023770	044124	023440	051106		

	023776	020047	046106	043501		
	024004	000				
2889	024005	104	052101	020101	EM142:	.ASCIZ /DATA COMPARE ERROR WITH CABLE/
	024012	047503	050115	051101		
	024020	020105	051105	047522		
	024026	020122	044527	044124		
	024034	041440	041101	042514		
	024042	000				
2890	024043	122	041524	044440	EM143:	.ASCIZ /RTC INTERRUPT AT PRIORITY 7/
	024050	052116	051105	052522		
	024056	052120	040440	020124		
	024064	051120	047511	044522		
	024072	054524	033440	000		
2891	024077	122	041524	044440	EM144:	.ASCIZ /RTC INTERRUPTS WHEN DISABLED/
	024104	052116	051105	052522		
	024112	052120	020123	044127		
	024120	047105	042040	051511		
	024126	041101	042514	000104		
2892	024134	052122	020103	047111	EM145:	.ASCIZ /RTC INTERRUPT DID NOT OCCUR/
	024142	042524	051122	050125		
	024150	020124	044504	020104		
	024156	047516	020124	041517		
	024164	052503	000122			
2893	024170	052122	020103	047111	EM146:	.ASCIZ /RTC INTERRUPT DID NOT OCCUR/
	024176	042524	051122	050125		
	024204	020124	044504	020104		
	024212	047516	020124	041517		
	024220	052503	000122			
2894	024224	052122	020103	047504	EM147:	.ASCIZ /RTC DOUBLE INTERRUPT/
	024232	041125	042514	044440		
	024240	052116	051105	052522		
	024246	052120	000			
2895	024251	122	051505	052105	EM150:	.ASCIZ /RESET DID NOT CLEAR RTC INTERRUPT/
	024256	042040	042111	047040		
	024264	052117	041440	042514		
	024272	051101	051040	041524		
	024300	044440	052116	051105		
	024306	052522	052120	000		
2896	024313	122	041524	044440	EM151:	.ASCIZ /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
	024320	052116	051105	052522		
	024326	052120	042040	042111		
	024334	047040	052117	041440		
	024342	042514	051101	053440		
	024350	052111	020110	044502		
	024356	033524	047440	020106		
	024364	045514	000123			
2897	024370	046103	041517	020113	EM152:	.ASCIZ /CLOCK REPEATABILITY ERROR/
	024376	042522	042520	052101		
	024404	041101	046111	052111		
	024412	020131	051105	047522		
	024420	000122				
2898	024422	046123	030525	051040	EM153:	.ASCIZ /SLU1 RECEIVER STATUS ERROR/
	024430	041505	044505	042526		
	024436	020122	052123	052101		
	024444	051525	042440	051122		
	024452	051117	000			

2926	025552	001104	001016	002756	DT32:	.WORD	\$TESTN,\$ERRPC,TBUF,0		
	025560	000000							
2927	025562	001104	001016	002750	DT36:	.WORD	\$TESTN,\$ERRPC,RCSR,0		
	025570	000000							
2928	025572	001104	001016	002752	DT37:	.WORD	\$TESTN,\$ERRPC,RBUF,0		
	025600	000000							
2929	025602	001104	001016	002770	DT61:	.WORD	\$TESTN,\$ERRPC,LKS,0		
	025610	000000							
2930	025612	001104	001016	001020	DT72:	.WORD	\$TESTN,\$ERRPC,\$GDADR,\$BDADR,\$GDDAT,\$BDDAT,0		
	025620	001022	001024	001026					
	025626	000000							
2931	025630	001104	001016	002760	DT115:	.WORD	\$TESTN,\$ERRPC,CRCR,\$GDDAT,\$BDDAT,0		
	025636	001024	001026	000000					
2932	025644	001104	001016	002750	DT137:	.WORD	\$TESTN,\$ERRPC,RCSR,\$BDDAT,0		
	025652	001026	000000						
2933	025656	001104	001016	001024	DT142:	.WORD	\$TESTN,\$ERRPC,\$GDDAT,\$BDDAT,0		
	025664	001026	000000						
2934	025670	001104	001016	002750	DT157:	.WORD	\$TESTN,\$ERRPC,RCSR,XMTCT2,RECCT2,0		
	025676	012232	012234	000000					
2935	025704	001104	001016	002750	DT161:	.WORD	\$TESTN,\$ERRPC,RCSR,OLDPC,BDVECT,0		
	025712	012414	012416	000000					
2936	025720	000200			BUF1:	.BLKW	200		:SLU1 INPUT BUFFER FOR BLAST TEST
2937	026320	000200			BUF2:	.BLKW	200		:SLU2 INPUT BUFFER FOR BLAST TEST
2938		000001				.END			

ABASE = 176500	BIT0 = 000001	DH161 025425	EM131 023320	EM5 016010
ACDW1 = 000000	BIT00 = 000001	DH32 025030	EM132 023354	EM50 020203
ACDW2 = 000000	BIT01 = 000002	DH5 024731	EM133 023411	EM51 020251
ACPUOP = 000000	BIT02 = 000004	DH6 024756	EM134 023463	EM52 020321
ADDW0 = 000000	BIT03 = 000010	DH61 025055	EM135 023533	EM53 020365
ADDW1 = 000000	BIT04 = 000020	DH7 025003	EM136 023566	EM54 020423
ADDW10 = 000000	BIT05 = 000040	DH72 025101	EM137 023632	EM55 020463
ADDW11 = 000000	BIT06 = 000100	DISPLA 001042	EM14 016254	EM56 020536
ADDW12 = 000000	BIT07 = 000200	DISPRE 000174	EM140 023701	EM57 020573
ADDW13 = 000000	BIT08 = 000400	DSWR = 177570	EM141 023741	EM6 016052
ADDW14 = 000000	BIT09 = 001000	DT115 025630	EM142 024005	EM60 020636
ADDW15 = 000000	BIT1 = 000002	DT137 025644	EM143 024043	EM61 020710
ADDW2 = 000000	BIT10 = 002000	DT142 025656	EM144 024077	EM62 020741
ADDW3 = 000000	BIT11 = 004000	DT157 025670	EM145 024134	EM63 021003
ADDW4 = 000000	BIT12 = 010000	DT161 025704	EM146 024170	EM64 021041
ADDW5 = 000000	BIT13 = 020000	DT31 025542	EM147 024224	EM65 021071
ADDW6 = 000000	BIT14 = 040000	DT32 025552	EM15 016324	EM66 021123
ADDW7 = 000000	BIT15 = 100000	DT36 025562	EM150 024251	EM67 021163
ADDW8 = 000000	BIT2 = 000004	DT37 025572	EM151 024313	EM7 016111
ADDW9 = 000000	BIT3 = 000010	DT5 025512	EM152 024370	EM70 021223
ADEVCT = 000000	BIT4 = 000020	DT6 025522	EM153 024422	EM71 021255
ADEVN = 000000	BIT5 = 000040	DT61 025602	EM154 024455	EM72 021306
AENV = 000000	BIT6 = 000100	DT7 025532	EM155 024510	EM74 021367
AENVN = 000000	BIT7 = 000200	DT72 025612	EM156 024545	EM75 021432
AFATAL = 000000	BIT8 = 000400	ECHO 015406	EM157 024612	EM76 021466
AMADR1 = 000000	BIT9 = 001000	EMTVEC = 000030	EM16 016372	EM77 021530
AMADR2 = 000000	BLAST 011474	EM100 021577	EM160 024647	ERRVEC = 000004
AMADR3 = 000000	BPT = 000003	EM101 021633	EM161 024714	FINIE 012170
AMADR4 = 000000	BPTVEC = 000014	EM102 021664	EM17 016430	GTSWR = 104406
AMAMS1 = 000000	BUF1 025720	EM103 021736	EM20 016470	HT = 000011
AMAMS2 = 000000	BUF2 026320	EM104 022005	EM21 016536	ID 003660
AMAMS3 = 000000	CATCH 012370	EM105 022041	EM22 016606	IOHAND 012036
AMAMS4 = 000000	CHECK2 012120	EM106 022104	EM23 016652	IOTVEC = 000020
AMSGAD = 000000	CKSWR = 104407	EM107 022165	EM24 016710	LF = 000012
AMSGLG = 000000	CONT41 010546	EM11 016150	EM25 016750	LKS 002770
AMSGTY = 000000	COUNT 012236	EM110 022221	EM26 017023	LTCIT 010762
AMTYP1 = 000000	CR = 000015	EM111 022256	EM27 017060	LTCRT 005626
AMTYP2 = 000000	CRBUF 002762	EM112 022330	EM30 017123	MFR 015627
AMTYP3 = 000000	CRCSR 002760	EM113 022400	EM31 017175	MOR 015641
AMTYP4 = 000000	CRLF = 000200	EM114 022433	EM32 017234	MPAR 015616
APASS = 000000	CRPSW 003004	EM115 022477	EM33 017273	MSG 015570
APRIOR = 000000	CRVECT 003002	EM116 022522	EM34 017345	MSTOP 015654
APTCSU = 000040	CTBUF 002766	EM117 022565	EM35 017413	M1 025476
APTENV = 000001	CTCSR 002764	EM120 022621	EM36 017455	M2 025510
APTSIZ = 000200	CTPSW 003010	EM121 022663	EM37 017514	OLDPC 012414
APTSPO = 000100	CTVECT 003006	EM122 022732	EM40 017553	OUTTST 015664
ASWREG = 000000	DADTBL 006066	EM123 022766	EM41 017623	PIRQ = 177772
ATESTN = 000000	DDISP = 177570	EM124 023017	EM42 017661	PIRQVE = 000240
AUNIT = 000000	DH115 025157	EM125 023071	EM43 017721	PRO = 000000
AUSWR = 000400	DH137 025223	EM126 023140	EM44 017767	PR1 = 000040
AVECT1 = 000300	DH142 025260	EM127 023174	EM45 020037	PR2 = 000100
AVECT2 = 000000	DH152 025314	EM13 016206	EM46 020105	PR3 = 000140
BDVECT 012416	DH155 025361	EM130 023237	EM47 020143	PR4 = 000200

PR5 = 000240	TBITVE= 000014	TVECT 002776	SESCAP 001072	SRDCHR 015024
PR6 = 000300	TBLEND 006076	TYPDS = 104405	SETABL 001120	SRDLIN 015154
FR7 = 000340	TBUF 002756	TYPE = 104401	SETEND 001160	SRDSZ = 000010
PS = 177776	TCSR 002754	TYPOC = 104402	SFATAL 001102	SRTNAD 012334
PSW = 177776	TICKER 011716	TYPON = 104404	SFFLG 013532	SSAVR6 013102
PWRVEC= 000024	TKVEC = 000060	TYPOS = 104403	SFILLC 001056	SSCOPE 013114
RBUF 002752	TPSW 003000	UNIQUE 006100	SFILLS 001055	\$SETUP= 000137
RCSR 002750	TPVEC = 000064	WAIT 015576	\$GDADR 001020	\$STUP = 177777
RDCMR = 104410	TRAPVE= 000034	WAITER 012074	\$GDDAT 001024	SSVLAD 013216
RDLIN = 104411	TRTVEC= 000014	WAITIO 011712	\$GET42 012312	SSVPC = 000500
RECC2 012234	TST1 003430	WDONE 005514	\$GTSWR 014612	SSWR = 161000
REC2 011762	TST10 004236	WRPSW 012356	\$HD = 000003	SSWREG 001122
RESVEC= 000010	TST11 004402	XMIT2 011726	\$HIBTS 000500	SSWRMK= 000000
RPSW 002774	TST12 004456	XMTCT2 012232	\$ICNT 001004	\$TESTN 001104
RTCPSW 003014	TST13 004516	XXX 011110	\$ILLUP 013076	\$TKB 001046
RTCVT 003012	TST14 004610	\$APTHD 000500	\$INTAG 001035	\$TKS 001044
RVECT 002772	TST15 004670	\$ATYC 013312	\$ITEMB 001014	\$TMP0 001060
R6 =X000006	TST16 004730	\$ATY1 013266	\$LF 001076	\$TMP1 001062
R7 =X000007	TST17 004770	\$ATY3 013274	\$LFLG 013531	\$TMP2 001064
SLU1IT 006262	TST2 003476	\$ATY4 013304	\$LPADR 001006	\$TMP3 001066
SLU2IT 007016	TST20 005126	\$AUTOB 001034	\$LPERR 001010	\$TMP4 001070
SLU2RT 004402	TST21 005310	\$BASE 001154	\$MADR1 001132	\$TN = 000054
STACK = 001100	TST22 005472	\$BDADR 001022	\$MADR2 001136	\$TPB 001052
START 003016	TST23 005564	\$BDDAT 001026	\$MADR3 001142	\$TPFLG 001057
STKLMT= 177774	TST24 005626	\$CHARC 014064	\$MADR4 001146	\$TPS 001050
SWR 001040	TST25 005702	\$CKSWR 014542	\$MAIL 001100	\$TRAP 015326
SWREG 000176	TST26 006100	\$CMTAG 001000	\$MAMS1 001130	\$TRAP2 015350
SW0 = 000001	TST27 006262	\$CM3 = 000000	\$MAMS2 001134	\$TRP = 000012
SW00 = 000001	TST3 003544	\$CM4 = 000005	\$MAMS3 001140	\$TRPAD 015362
SW01 = 000002	TST30 006430	\$CNTLG 015277	\$MAMS4 001144	\$STM 000504
SW02 = 000004	TST31 006550	\$CNTLU 015272	\$MBADR 000502	\$STSTM 001002
SW03 = 000010	TST32 006704	\$CPUOP 001126	\$MFLG 013530	\$TTYIN 015262
SW04 = 000020	TST33 007016	\$CRLF 001075	\$MNEW 015315	\$TYPDS 014316
SW05 = 000040	TST34 007140	\$DBLK 014532	\$MSGAD 001114	\$TYPE 013534
SW06 = 000100	TST35 007260	\$DEVCT 001110	\$MSGLG 001116	\$TYPEC 013746
SW07 = 000200	TST36 007414	\$DEVM 001156	\$MSGTY 001100	\$TYPEX 014066
SW08 = 000400	TST37 007556	\$DOAGN 012332	\$MSWR 015304	\$TYPOC 014114
SW09 = 001000	TST4 003714	\$DTBL 014522	\$MTYP1 001131	\$TYPON 014130
SW1 = 000002	TST40 007676	\$ENDAD 012322	\$MTYP2 001135	\$TYPOS 014070
SW10 = 020000	TST41 010014	\$ENDCT 012270	\$MTYP3 001141	\$UNIT 001112
SW11 = 004000	TST42 010146	\$ENDMG 012347	\$MTYP4 001145	\$UNITM 000510
SW12 = 010000	TST43 010252	\$ENULL 012336	\$NULL 001054	\$USWR 001124
SW13 = 020000	TST44 010376	\$ENV 001120	\$NWTST= 000001	\$VECT1 001150
SW14 = 040000	TST45 010464	\$ENVM 001121	\$OCNT 014312	\$VECT2 001152
SW15 = 100000	TST46 010572	\$EOP 012240	\$OMODE 014314	\$XOFF = 000023
SW2 = 000004	TST47 010664	\$EOPCT 012262	\$OVER 013252	\$XON = 000021
SW3 = 000010	TST5 003772	\$ERFLG 001003	\$PASS 001106	\$XTSTR 013126
SW4 = 000020	TST50 010762	\$ERMAX 001015	\$PASTM 000506	\$\$GET4= 000000
SW5 = 000040	TST51 011210	\$ERROR 012420	\$POWER 013104	\$OFILL 014313
SW6 = 000100	TST52 011356	\$ERRPC 001016	\$PWRDN 012736	. = 026720
SW7 = 000200	TST53 011474	\$ERRTB 001160	\$PWRMG 013072	.\$ERRT 001160
SW8 = 000400	TST6 004032	\$ERRTY 012602	\$PWRUP 013010	.\$X = 000500
SW9 = 001000	TST7 004072	\$ERTTL 001012	\$QUES 001074	

. ABS. 026720 000 CON RO ABS GBL D

CJKDIAO 11/23B SLU LTC REPR DIAG
CJKDIA.P11 20-OCT-81 16:08

DNMAC X24.07-563 02-FEB-82^{E 8} 16:42 PAGE 22-2
SYMBOL TABLE

SEQ 0095

ERRORS DETECTED: 0

CJKDIA,CJKDIA=CJKDIA.P11
RUN-TIME: 20 12 .5 SECONDS
RUN-TIME RATIO: 66/33=1.9
CORE USED: 22K (43 PAGES)