

11/24

11/24 OPTIONS DIAG
CJKDFAO

AH-F604A-MC
FICHE 1 OF 1

AUG 1981
COPYRIGHT © 1981
MADE IN USA



000000

.REPT 0

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

IDENTIFICATION

PRODUCT CODE: AC-F602A-MC
PRODUCT NAME: CJKDFAO 11/24 OPTIONS DIAGNOSTIC
PRODUCT DATE: APR-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 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): 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	DEC/X11

CARD 40 11/24 OPTIONS DIAGNOSTIC
CARD A.P11 19-MAR-81 14:27

MACY11 30A(1052) 19-MAR-81^{C 1} 14:27 PAGE 2

SEQ 0002

46
47
48
49
50

REVISION A

HISTORY
FIRST RELEASE OF DIAGNOSTIC

TABLE OF CONTENTS

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

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

88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141

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 M7133 MODULE. ITS MAIN PURPOSE IS TO PROVIDE SCOPE LOOPING FOR REPAIR PERSONNEL. THE PROGRAM DETECTS FROM 85-95% OF ALL STUCK-AT-0, STUCK-AT-1 FAULTS. 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 SLU1). 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 AUTOMATED TEST SYSTEMS.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

- 11/24 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
- KDF11-UA (M7133) MODULE SPECIFICATION REV. 0.
25-AUGUST-78 BILL BERNSTEIN
- 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

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

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

IF AN END PASS MESSAGE IS RECEIVED FROM CJKDEXX THERE IS NO NEED TO
RUN THIS DIAGNOSTIC BECAUSE ALL DEVICES HAVE BEEN FULLY TESTED.

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

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 LOCA-
TION = 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 RE-
PORTS ANY ERROR FLAGS SET IN THE RECEIVER BUFFER. A
CONTROL-C (^C) HALTS THE TEST AND PRINTS 'STOP' AT THE TERMI-
NAL CONTINUING RESTARTS THE ECHO TEST.
- C. START AT 210. *****NOTE*****

197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

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):

(OCTAL CODE)
!' \$% '()*+,-./0123456789:;<=>? (040 --> 077
@ABCDEFGHIJKLMN0PQRSTUVWXYZ[] (100 --> 137)
'ABCDEFGHIJKLMN0PQRSTUVWXYZ (140 --> 177) [LOWER CASE
ALPHA]

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
- 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 AND HAVE THE SLU IN MAINTENANCE MODE. 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 =

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300

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 TYPEOUT (BIT13) OF THE SWR IS 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 LKS

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.)

301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355

NOTE

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

3.2 ERROR HALTS

THERE ARE 5 ERRORS IN THIS PROGRAM THAT CAUSE A HALT. THESE ERRORS ARE IN TESTS 1, 2, 3, AND 7.

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, INDICATING THAT THE CHARACTER WAS NEVER TRANSMITTED. THIS COULD CAUSE MICRO-ODT TO NOT RUN OR GARBLED OUTPUT FROM THE MICRO-ODT.

TEST 7 ERROR - THIS ERROR INDICATES THAT THE SLU CANNOT BE TAKEN OUT OF MAINTENANCE MODE. MICRO-ODT SHOULD BE UNEFFECTED BY THIS ERROR.

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.

356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400

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	-----															
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
	I	I	I	I												I
ERROR-OVERRUN	-----															
FRAME ERROR	-----															
RECEIVE PARITY ERROR	-----															
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

									I	I			I			
TRANSMITTER READY	-----															
TRANSMITTER INTERRUPT ENABLE	-----															
MAINTENANCE	-----															

401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454

SLU1 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

TRANSMITTER DATA BITS (8) -----

LTC CSR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-----								I	I	I						I

LINE CLOCK MONITOR -----

INTERRUPT ENABLE -----

SLU2 RCSR

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
-----								I	I	I						I

RECEIVER DONE -----

INTERRUPT ENABLE -----

SLU2 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

ERROR -

OVERRUN ---

FRAME ERROR ---

RECEIVER PARITY ---

ERROR

RECEIVER DATA BITS (8) -----

455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509

SLU2 XCSR

	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 DONE	-----															I
INTERRUPT ENABLE	-----															I
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

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

510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
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

- 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 (DEPENDENT 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.
- TEST7 TEST THAT SLU1 BIT2 (MAINTENANCE) CAN BE SET AND RESET. BECAUSE AN ERROR ON THIS TEST MAY LEAVE THE SLU IN AN UNKNOWN STATE, ERRORS CAUSE THE PROGRAM TO HALT.
- 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.
- TEST12 TEST THAT SLU1 RECEIVER TEST 7 (DONE) SETS AND CLEARS PROPERLY. THIS TEST PUTS THE SLU INTO MAINTENANCE MODE AND TRANSMITS A CHARACTER. RECEIVER 'DONE' SHOULD SET WHEN CHARACTER IS RECEIVED. AFTER 'DONE' SETS THE PROGRAM ATTEMPTS TO CLEAR IT WITH A 'RESET'.
- TEST13 TEST SLU1 THAT READING RECEIVER BUFFER CLEAR RECEIVER 'DONE'. A CHARACTER IS TRANSMITTED IN MAINTENANCE MODE, WHEN RECEIVER 'DONE' SETS THE RECEIVER BUFFER IS READ WHICH SHOULD CLEAR DONE.
- 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.

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
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

- 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".
- TEST31 TEST THAT BIT7 (DONE) OF LINE CLOCK STATUS REGISTER SETS AND CAN BE CLEARED.
- 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 ANY OTHER REGISTER. THE TEST IS REPEATED FOR ALL THE 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.
- TEST37-42 SAME AS TESTS FOR SLU1 RECEIVER INTERRUPTS AS TRANSMITTER INTERRUPTS.
- TEST43 TEST SLU1 THAT RESET CLEARS RECEIVER INTERRUPTS. SET UP FOR RECEIVER INTERRUPT PENDING WITH PROCESSOR AT PRIORITY 7, ISSUE A "RESET" AND DROP PROCESSOR PRIORITY. THE RECEIVER SHOULD NOT INTERRUPT.
- TEST44 TEST SLU1 THAT OVERRUN AND ERROR (BITS 14 AND 15) CAN BE SET. THIS TEST PUTS THE SLU1 IN MAINTENANCE MODE AND TRANSMITS 3 CHARACTERS WITHOUT READING THE RECEIVER BUFFER. THIS SHOULD CAUSE BITS 14 AND 15 TO SET.
- TEST45 TEST SLU1 DATA PATH USING MAINTENANCE WRAP AROUND. THIS

620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675

TEST TRANSMITS AN INCREMENTING DATA PATTERN AND VERIFIES THE CORRECT DATA IS RECEIVED.

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.

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.

TEST66 TEST THAT LINE TIME CLOCK INTERRUPT CLEARS BY CLEARING BIT7 OF LINE CLOCK STATUS REGISTER.

TEST67 TEST LINE TIME CLOCK REPEATABILITY THIS TEST VERIFIES THAT THE PROCESSOR DOES THE SAME AMOUNT OF INSTRUCTIONS FOR TWO INTERRUPTS OF THE LINE CLOCK, THUS INDICATING EQUAL TIMES FOR EACH INTERRUPT. BECAUSE OF PROBLEMS OF SYNCHRONIZING THE PROCESSOR AND CLOCK A SMALL DEVIATION IS ALLOWED.

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.

676
677
678
679
680
681
682
683
684
685
686
687
688
689

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

.ENDR

690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745

000001
160000

001100

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

000000
000040
000100
000140
000200
000240
000300

```

.TITLE CJKDFAO 11/24 OPTIONS DIAGNOSTIC
:*COPYRIGHT (C) JAN-81
:*DIGITAL EQUIPMENT CORP.
:*MAYNARD, MASS. 01754
:*
:*
:*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= %0      ;;GENERAL REGISTER
R1= %1      ;;GENERAL REGISTER
R2= %2      ;;GENERAL REGISTER
R3= %3      ;;GENERAL REGISTER
R4= %4      ;;GENERAL REGISTER
R5= %5      ;;GENERAL REGISTER
R6= %6      ;;GENERAL REGISTER
R7= %7      ;;GENERAL REGISTER
SP= %6      ;;STACK POINTER
PC= %7      ;;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

```


746 000340
747
748
749 100000
750 040000
751 020000
752 010000
753 004000
754 002000
755 001000
756 000400
757 000200
758 000100
759 000040
760 000020
761 000010
762 000004
763 000002
764 000001
765
766
767
768
769
770
771
772
773
774
775
776
777 100000
778 040000
779 020000
780 010000
781 004000
782 002000
783 001000
784 000400
785 000200
786 000100
787 000040
788 000020
789 000010
790 000004
791 000002
792 000001
793
794
795
796
797
798
799
800
801

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
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

;'DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1


```

802
803
804
805      000004
806      000010
807      000014
808      000014
809      000014
810      000020
811      000024
812      000030
813      000034
814      000060
815      000064
816      000240
817      176500
818      000300
819      000400
820      000001
821      161000
822      000003
823
824
825      000000
826
827
828
829
830
831
832      000014
833 000014 015070
834 000016 000340
835
836      000042
837 000042 000000
838
839
840
841
842      000174
843 000174 000000
844 000176 000000
845
846      000200
847 000200 000137 003036
848 000204 000137 020106
849 000210 000137 020364
  
```

```

.EQUIV BIT00,BIT0

;*BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4           ;;TIME OUT AND OTHER ERRORS
RESVEC= 10          ;;RESERVED AND ILLEGAL INSTRUCTIONS
TBITVEC=14          ;;"T" BIT
TRTVEC= 14          ;;TRACE TRAP
BPTVEC= 14          ;;BREAKPOINT TRAP (BPT)
IOTVEC= 20          ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
PWRVEC= 24          ;;POWER FAIL
EMTVEC= 30          ;;EMULATOR TRAP (EMT) **ERROR**
TRAPVEC=34          ;;"TRAP" TRAP
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
;WORD CATCH
;WORD 340
;THE BPT TRAP VECTOR POINTS TO THE
; ILLEGAL TRAP HANDLER "CATCH"

.= 42
;WORD 0

DISPREG: .WORD 174
SWREG: .WORD 0

.=200
JMP START ;DO INTERFACE TEST
JMP ECHO ;DO ECHO TEST
JMP OUTST ;DO OUTPUT TEST TO TERMINAL
  
```


850
 851 000500
 852
 853
 854
 855
 856 000500
 857 000046
 858 000046 015022
 859 000052 000052
 860 000052 000000
 861 000500
 862
 863
 864
 865
 866
 867 000500
 868 000024
 869 000024 000200
 870 000044 000044
 871 000044 000500
 872 000500
 873
 874
 875
 876
 877 000500
 878 000500 000000
 879 000502 001100
 880 000504 000050
 881 000506 000060
 882 000510 000055
 883 000512 000030
 884

```

      . = 500
.SBTTL ACT11 HOOKS

:*****
:HOOKS REQUIRED BY ACT11
      $SVPC= .           ;SAVE PC
      . = 46           ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
      $ENDAD           ;;2)SET LOC.52 TO ZERO
      . = 52           ;; RESTORE PC
      .WORD 0
      .=$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)
$TSTM:  .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)
  
```


885
886
887
888
889
890
891 001000 001000
892 001000 000000
893 001000 000000
894 001002 000
895 001003 000
896 001004 000000
897 001006 000000
898 001010 000000
899 001012 000000
900 001014 000
901 001015 001
902 001016 000000
903 001020 000000
904 001022 000000
905 001024 000000
906 001026 000000
907 001030 000000
908 001032 000000
909 001034 000
910 001035 000
911 001036 000000
912 001040 177570
913 001042 177570
914 001044 177560
915 001046 177562
916 001050 177564
917 001052 177566
918 001054 000
919 001055 002
920 001056 012
921 001057 000
922 001060 000000
923 001062 000000
924 001064 000000
925 001066 000000
926 001070 000000
927 001072 000000
928 001074 077
929 001075 015
930 001076 000012
931
932
933
934
935
936 001100
937 001100 000000
938 001102 000000
939 001104 000000
940 001106 000000

```
.SBTTL COMMON TAGS

:*****
:*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
:*USED IN THE PROGRAM.

      . =1000
$CMTAG:      ::START OF COMMON TAGS
      .WORD 0
$STNM: .BYTE 0      ::CONTAINS THE TEST NUMBER
$ERFLG: .BYTE 0     ::CONTAINS ERROR FLAG
$IICNT: .WORD 0     ::CONTAINS SUBTEST ITERATION COUNT
$LPADR: .WORD 0     ::CONTAINS SCOPE LOOP ADDRESS
$LPERR: .WORD 0     ::CONTAINS SCOPE RETURN FOR ERRORS
$ERTTI: .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
      .WORD 0
$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

:*****
.EVEN
$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
```


941	001110	000000	\$DEVCT: .WORD	ADEVCT	::DEVICE COUNT
942	001112	000000	\$UNIT: .WORD	AUNIT	::I/O UNIT NUMBER
943	001114	000000	\$MSGAD: .WORD	AMSGAD	::MESSAGE ADDRESS
944	001116	000000	\$MSGLG: .WORD	AMSGLG	::MESSAGE LENGTH
945	001120		\$ETABLE:		::APT ENVIRONMENT TABLE
946	001120	000	\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
947	001121	000	\$ENVM: .BYTE	AENVM	::ENVIRONMENT MODE BITS
948	001122	000000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
949	001124	000400	\$USWR: .WORD	AUSWR	::USER SWITCHES
950	001126	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE,OPTIONS
951			.*		BITS 15-11=CPU TYPE
952			.*		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
953			.*		11/70=06,PDQ=07,Q=10
954			.*		BIT 10=REAL TIME CLOCK
955			.*		BIT 9=FLOATING POINT PROCESSOR
956			.*		BIT 8=MEMORY MANAGEMENT
957	001130	000	\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
958	001131	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE,BLK#1
959			.*		MEM.TYPE BYTE -- (HIGH BYTE)
960			.*		900 NSEC CORE=001
961			.*		300 NSEC BIPOLAR=002
962			.*		500 NSEC MOS=003
963	001132	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS,BLK#1
964			.*		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
965	001134	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
966	001135	000	\$MTYP2: .BYTE	AMTYP2	::MEM. TYPE,BLK#2
967	001136	000000	\$MADR2: .WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
968	001140	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
969	001141	000	\$MTYP3: .BYTE	AMTYP3	::MEM. TYPE,BLK#3
970	001142	000000	\$MADR3: .WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
971	001144	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
972	001145	000	\$MTYP4: .BYTE	AMTYP4	::MEM. TYPE,BLK#4
973	001146	000000	\$MADR4: .WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
974	001150	000300	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRICRITY#1
975	001152	000000	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
976	001154	176500	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
977	001156	000000	\$DEVCT: .WORD	ADEVCT	::DEVICE MAP
978	001160		\$ETEND:		
979			.MEXIT		

980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 990
 991
 992
 993
 994 001160
 995
 996 001160
 997 001160 000020
 998
 999
 1000
 1001 001220 020510
 1002 001222 027431
 1003 001224 030212
 1004 001226 000000
 1005
 1006 001230 020552
 1007 001232 027456
 1008 001234 030222
 1009 001236 000000
 1010
 1011 001240 020611
 1012 001242 027503
 1013 001244 030232
 1014 001246 000000
 1015
 1016 001250 000004
 1017
 1018
 1019 001260 020650
 1020 001262 027431
 1021 001264 030212
 1022 001266 000000
 1023
 1024 001270 000004
 1025
 1026
 1027 001300 020706
 1028 001302 027431
 1029 001304 030212
 1030 001306 000000
 1031
 1032 001310 020754
 1033 001312 027431
 1034 001314 030212
 1035 001316 000000

.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

EM5 ;SLU1 TCSR DONE NOT SET WITH RESET
 DH5 ;'TEST# ERR PC TCSR'
 DT5 ;\$TESTN,\$ERRPC,CTCSR
 0

EM6 ;SLU1 RCSR DID NOT RETURN SSYNC
 DH6 ;'TEST# ERR PC RCSR'
 DT6 ;\$TESTN,\$ERRPC,CRCSR
 0

EM7 ;SLU1 RBUF DID NOT RETURN SSYNC
 DH7 ;'TEST# ERR PC RBUF'
 DT7 ;\$TESTN,\$ERRPC,CRBUF
 0

.BLKW 4 ;MORE PACIFICATION

EM11 ;'CAN NOT SET BIT2 OF SLU1 TCSR'
 DH5 ;'TEST# ERR PC TCSR'
 DT5 ;\$TESTN,\$ERRPC,CTCSR
 0

.BLKW 4 ;STILL MORE PACIFICATION

EM13 ;'RESET DID NOT CLEAR BIT2 OF SLU1 TCSR'
 DH5 ;'TEST# ERR PC TCSR'
 DT5 ;\$TESTN,\$ERRPC,CTCSR
 0

EM14 ;'BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET2
 DH5 ;'TEST# ERR PC TCSR'
 DT5 ;\$TESTN,\$ERRPC,CTCSR
 0

1036					
1037	001320	021024	EM15	:	'SLU1 XMIT INTERRUPTED WITH PRIORITY 7''
1038	001322	027431	DH5	:	'TEST# ERR PC TCSR''
1039	001324	030212	DT5	:	\$TESTN,\$ERRPC,CTCSR
1040	001326	000000	0		
1041					
1042	001330	021072	EM16	:	'CAN NOT SET BIT6 OF SLU1 TCSR''
1043	001332	027431	DH5	:	'TEST# ERR PC TCSR''
1044	001334	030212	DT5	:	\$TESTN,\$ERRPC,CTCSR
1045	001336	000000	0		
1046					
1047	001340	021130	EM17	:	'CAN NOT CLEAR BIT6 OF SLU1 TCSR''
1048	001342	027431	DH5	:	'TEST# ERR PC TCSR''
1049	001344	030212	DT5	:	\$TESTN,\$ERRPC,CTCSR
1050	001346	000000	0		
1051					
1052	001350	021170	EM20	:	'RESET DID NOT CLEAR BIT6 OF SLU1 TCSR''
1053	001352	027431	DH5	:	'TEST# ERR PC TCSR''
1054	001354	030212	DT5	:	\$TESTN,\$ERRPC,CTCSR
1055	001356	000000	0		
1056					
1057	001360	021236	EM21	:	'BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET''
1058	001362	027456	DH6	:	'TEST# ERR PC RCSR''
1059	001364	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1060	001366	000000	0		
1061					
1062	001370	021306	EM22	:	'SLU1 RCVR INTERRUPT WITH PRIORITY 7''
1063	001372	027456	DH6	:	'TEST# ERR PC RCSR''
1064	001374	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1065	001376	000000	0		
1066					
1067	001400	021352	EM23	:	'CAN NOT SET BIT6 OF SLU1 RCSR''
1068	001402	027456	DH6	:	'TEST# ERR PC RCSR''
1069	001404	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1070	001406	000000	0		
1071					
1072	001410	021410	EM24	:	'CAN NOT CLEAR BIT6 OF SLU1 RCSR''
1073	001412	027456	DH6	:	'TEST# ERR PC RCSR''
1074	001414	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1075	001416	000000	0		
1076					
1077	001420	021450	EM25	:	'CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET2
1078	001422	027456	DH6	:	'TEST# ERR PC RCSR''
1079	001424	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1080	001426	000000	0		
1081					
1082	001430	021523	EM26	:	'SLU1 RECEIVER DONE NEVER SET''
1083	001432	027456	DH6	:	'TEST# ERR PC RCSR''
1084	001434	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1085	001436	000000	0		
1086					
1087	001440	021560	EM27	:	'RESET DID NOT CLEAR SLU1 RCVR DONE''
1088	001442	027456	DH6	:	'TEST# ERR PC RCSR''
1089	001444	030222	DT6	:	\$TESTN,\$ERRPC,CRCSR
1090	001446	000000	0		
1091					

1092	001450	021623	EM30	;'READING SLU1 RBUF DID NOT CLEAR RCVR DONE''
1093	001452	027456	DH6	;'TEST# ERR PC RCSR''
1094	001454	030222	DT6	;\$TESTN,\$ERRPC,CRCR
1095	001456	000000	0	
1096				
1097	001460	021675	EM31	;'SLU2 TCSR DID NOT RETURN SSYNC
1098	001462	027431	DH5	;'TEST# ERR PC TCSR''
1099	001464	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1100	001466	000000	0	
1101				
1102	001470	021734	EM32	;'SLU2 TBUF DID NOT RETURN SSYNC
1103	001472	027530	DH32	;'TEST# ERR PC TBUF''
1104	001474	030252	DT32	;\$TESTN,\$ERRPC,TBUF
1105	001476	000000	0	
1106				
1107	001500	021773	EM33	;'SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL''
1108	001502	027431	DH5	;'TEST# ERR PC TCSR''
1109	001504	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1110	001506	000000	0	
1111				
1112	001510	022045	EM34	;'SLU2 TCSR DONE NOT SET AFTER TRANSMIT''
1113	001512	027431	DH5	;'TEST# ERR PC TCSR''
1114	001514	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1115	001516	000000	0	
1116				
1117	001520	022113	EM35	;'SLU2 TCSR DONE NOT SET WITH RESET''
1118	001522	027431	DH5	;'TEST# ERR PC TCSR''
1119	001524	030242	DT31	;\$TESTN,ERRPC,TCSR
1120	001526	000000	0	
1121				
1122	001530	022155	EM36	;'SLU2 RCSR DID NOT RETURN SSYNC''
1123	001532	027456	DH6	;'TEST# ERR PC RCSR''
1124	001534	030262	DT36	;\$TESTN,ERRPC,RCSR
1125	001536	000000	0	
1126				
1127	001540	022214	EM37	;'SLU2 RBUF DID NOT RETURN SSYNC''
1128	001542	027503	DH7	;'TEST# ERR PC RBUF''
1129	001544	030272	DT37	;\$TESTN,ERRPC,RBUF
1130	001546	000000	0	
1131				
1132	001550	022253	EM40	;'BITO OF SLU2 TCSR NOT CLEAR AFTER RESET''
1133	001552	027431	DH5	;'TEST# ERR PC TCSR
1134	001554	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1135	001556	000000	0	
1136				
1137	001560	022323	EM41	;'CAN NOT SET BITO OF SLU2 TCSR''
1138	001562	027431	DH5	;'TEST# ERR PC TCSR
1139	001564	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1140	001566	000000	0	
1141				
1142	001570	022361	EM42	;'CAN NOT CLEAR BITO OF SLU2 TCSR''
1143	001572	027431	DH5	;'TEST# ERR PC TCSR
1144	001574	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1145	001576	000000	0	
1146				
1147	001600	022421	EM43	;'RESET DID NOT CLEAR BITO OF SLU2 TCSR''

1148	001602	027431	DH5	;'TEST# ERR PC TCSR
1149	001604	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1150	001606	000000	0	
1151				
1152	001610	022467	EM44	;'BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET2
1153	001612	027431	DH5	;'TEST# ERR PC TCSR
1154	001614	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1155	001616	000000	0	
1156				
1157	001620	022537	EM45	;'SLU2 XMIT INTERRUPTED WITH PRIORITY 7'
1158	001622	027431	DH5	;'TEST# ERR PC TCSR
1159	001624	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1160	001626	000000	0	
1161				
1162	001630	022605	EM46	;'CAN NOT SET BIT6 OF SLU2 TCSR'
1163	001632	027431	DH5	;'TEST# ERR PC TCSR
1164	001634	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1165	001636	000000	0	
1166				
1167	001640	022643	EM47	;'CAN NOT CLEAR BIT6 OF SLU2 TCSR'
1168	001642	027431	DH5	;'TEST# ERR PC TCSR
1169	001644	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1170	001646	000000	0	
1171				
1172	001650	022703	EM50	;'RESET DID NOT CLEAR BIT6 OF SLU2 TCSR'
1173	001652	027431	DH5	;'TEST# ERR PC TCSR
1174	001654	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1175	001656	000000	0	
1176				
1177	001660	022751	EM51	;'BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET'
1178	001662	027456	DH6	;'TEST# ERR PC RCSR'
1179	001664	030262	DT36	;\$TESTN,ERRPC,RCSR
1180	001666	000000	0	
1181				
1182	001670	023021	EM52	;'SLU2 RCVR INTERRUPT WITH PRIORITY 7'
1183	001672	027456	DH6	;'TEST# ERR PC RCSR'
1184	001674	030262	DT36	;\$TESTN,ERRPC,RCSR
1185	001676	000000	0	
1186				
1187	001700	023065	EM53	;'CAN NOT SET BIT6 OF SLU2 RCSR'
1188	001702	027456	DH6	;'TEST# ERR PC RCSR'
1189	001704	030262	DT36	;\$TESTN,ERRPC,RCSR
1190	001706	000000	0	
1191				
1192	001710	023123	EM54	;'CAN NOT CLEAR BIT6 OF SLU2 RCSR'
1193	001712	027456	DH6	;'TEST# ERR PC RCSR'
1194	001714	030262	DT36	;\$TESTN,ERRPC,RCSR
1195	001716	000000	0	
1196				
1197	001720	023163	EM55	;'CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET2
1198	001722	027456	DH6	;'TEST# ERR PC RCSR'
1199	001724	030262	DT36	;\$TESTN,ERRPC,RCSR
1200	001726	000000	0	
1201				
1202	001730	023236	EM56	;'SLU2 RECEIVER DONE NEVER SET'
1203	001732	027456	DH6	;'TEST# ERR PC RCSR'

1204	001734	030262	DT36	;\$TESTN,ERRPC,RCSR
1205	001736	000000	0	
1206				
1207	001740	023273	EM57	;'RESET DID NOT CLEAR SLU2 RCVR DONE''
1208	001742	027456	DH6	;'TEST# ERR PC RCSR''
1209	001744	030262	DT36	;\$TESTN,ERRPC,RCSR
1210	001746	000000	0	
1211				
1212	001750	023336	EM60	;'READING SLU2 RBUF DID NOT CLEAR RCVR DONE''
1213	001752	027456	DH6	;'TEST# ERR PC RCSR''
1214	001754	030262	DT36	;\$TESTN,ERRPC,RCSR
1215	001756	000000	0	
1216				
1217	001760	023410	EM61	;'LKS DID NOT RETURN SSYNC
1218	001762	027555	DH61	;'TEST# ERR PC LKS''
1219	001764	030302	DT61	;\$TESTN,ERRPC,LKS
1220	001766	000000	0	
1221				
1222	001770	023441	EM62	;'BIT6 OF LKS NOT CLEAR AFTER RESET''
1223	001772	027555	DH61	;'TEST# ERR PC LKS''
1224	001774	030302	DT61	;\$TESTN,\$ERRPC,LKS
1225	001776	000000	0	
1226				
1227	002000	023503	EM63	;'LKS INTERRUPT WITH PRIORITY 7''
1228	002002	027555	DH61	;'TEST# ERR PC LKS''
1229	002004	030302	DT61	;\$TESTN,\$ERRPC,LKS
1230	002006	000000	0	
1231				
1232	002010	023541	EM64	;'CAN NOT SET BIT6 OF LKS''
1233	002012	027555	DH61	;'TEST# ERR PC LKS''
1234	002014	030302	DT61	;\$TESTN,\$ERRPC,LKS
1235	002016	000000	0	
1236				
1237	002020	023571	EM65	;'CAN NOT CLEAR BIT6 OF LKS''
1238	002022	027555	DH61	;'TEST# ERR PC LKS''
1239	002024	030302	DT61	;\$TESTN,\$ERRPC,LKS
1240	002026	000000	0	
1241				
1242	002030	023623	EM66	;'RESET DID NOT CLEAR BIT6 OF LKS''
1243	002032	027555	DH61	;'TEST# ERR PC LKS''
1244	002034	030302	DT61	;\$TESTN,\$ERRPC,LKS
1245	002036	000000	0	
1246				
1247	002040	023663	EM67	;'BIT7 OF LKS NOT SET AFTER RESET2
1248	002042	027555	DH61	;'TEST# ERR PC LKS''
1249	002044	030302	DT61	;\$TESTN,\$ERRPC,LKS
1250	002046	000000	0	
1251				
1252	002050	023723	EM70	;'CAN NOT CLEAR BIT7 OF LKS''
1253	002052	027555	DH61	;'TEST# ERR PC LKS''
1254	002054	030302	DT61	;\$TESTN,\$ERRPC,LKS
1255	002056	000000	0	
1256				
1257	002060	023755	EM71	;'BIT7 OF LKS DOES NOT SET''
1258	002062	027555	DH61	;'TEST# ERR PC LKS''
1259	002064	030302	DT61	;\$TESTN,\$ERRPC,LKS

1260	002066	000000	0	
1261				
1262	002070	024006	EM72	:WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER
1263	002072	027601	DH72	: 'TEST# ERR PC GOOD BAD GDDATA BDDATA
1264	002074	030312	DT72	: \$TESTN,\$ERRPC,\$GDADR,\$BDADR,\$GDDAT,\$BDDAT
1265	002076	000000	0	
1266				
1267	002100	000004	.BLKW 4	:THE LAST IN A LONG LINE OF PACIFICATION
1268				
1269				
1270	002110	024067	EM74	: 'SLU1 XMIT INTERRUPTS WHEN DISABLED'
1271	002112	027431	DH5	: 'TEST# ERR PC TCSR'
1272	002114	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1273	002116	000000	0	
1274				
1275	002120	024132	EM75	: 'SLU1 XMIT DID NOT INTERRUPT'
1276	002122	027431	DH5	: 'TEST# ERR PC TCSR'
1277	002124	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1278	002126	000000	0	
1279				
1280	002130	024166	EM76	: 'SLU1 XMIT INTERRUPT AT PRIORITY 7'
1281	002132	027431	DH5	: 'TEST# ERR PC TCSR'
1282	002134	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1283	002136	000000	0	
1284				
1285	002140	024230	EM77	: 'SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR'
1286	002142	027431	DH5	: 'TEST# ERR PC TCSR'
1287	002144	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1288	002146	000000	0	
1289				
1290	002150	024277	EM100	: 'SLU1 XMIT DID NOT INTERRUPT'
1291	002152	027431	DH5	: 'TEST# ERR PC TCSR'
1292	002154	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1293	002156	000000	0	
1294				
1295	002160	024333	EM101	: 'SLU1 XMIT RE-INTERRUPTED'
1296	002162	027431	DH5	: 'TEST# ERR PC TCSR'
1297	002164	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1298	002166	000000	0	
1299				
1300	002170	024364	EM102	: 'LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT'
1301	002172	027431	DH5	: 'TEST# ERR PC TCSR'
1302	002174	030212	DT5	: \$TESTN,\$ERRPC,CTCSR
1303	002176	000000	0	
1304				
1305	002200	024436	EM103	: 'SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR'
1306	002202	027456	DH6	: 'TEST# ERR PC RCSR'
1307	002204	030222	DT6	: \$TESTN,\$ERRPC,CRCSR
1308	002206	000000	0	
1309				
1310	002210	024505	EM104	: 'SLU1 RCVR DID NOT INTERRUPT'
1311	002212	027456	DH6	: 'TEST# ERR PC RCSR'
1312	002214	030222	DT6	: \$TESTN,\$ERRPC,CRCSR
1313	002216	000000	0	
1314				
1315	002220	024541	EM105	: 'SLU1 RCVR INTERRUPTS AT PRIORITY 7'

1316	002222	027456	DH6	:'TEST# ERR PC RCSR'
1317	002224	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1318	002226	000000	0	
1319				
1320	002230	024604	EM106	:'SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR'
1321	002232	027456	DH6	:'TEST# ERR PC RCSR'
1322	002234	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1323	002236	000000	0	
1324				
1325	002240	024665	EM107	:'SLU1 RCVR DID NOT INTERRUPT'
1326	002242	027456	DH6	:'TEST# ERR PC RCSR'
1327	002244	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1328	002246	000000	0	
1329				
1330	002250	024721	EM110	:'SLU1 RECEIVER RE-INTERRUPTED'
1331	002252	027456	DH6	:'TEST# ERR PC RCSR'
1332	002254	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1333	002256	000000	0	
1334				
1335	002260	024756	EM111	:'SLU1 READING RBUF DID NOT CLEAR INTERRUPT'
1336	002262	027456	DH6	:'TEST# ERR PC RCSR'
1337	002264	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1338	002266	000000	0	
1339				
1340	002270	025030	EM112	:'RESET DID NOT CLEAR SLU1 RCVR INTERRUPT'
1341	002272	027456	DH6	:'TEST# ERR PC RCSR'
1342	002274	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1343	002276	000000	0	
1344				
1345	002300	025100	EM113	:'SLU1 'OR' FLAG DID NOT SET'
1346	002302	027456	DH6	:'TEST# ERR PC RCSR'
1347	002304	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1348	002306	000000	0	
1349				
1350	002310	025133	EM114	:'SLU1 'ERROR' NOT SET WITH 'OR' FLAG'
1351	002312	027456	DH6	:'TEST# ERR PC RCSR'
1352	002314	030222	DT6	:\$TESTN,\$ERRPC,CRCSR
1353	002316	000000	0	
1354				
1355	002320	025177	EM115	:'DATA COMPARE ERROR'
1356	002322	027657	DH115	:'TEST# ERR PC CRCSR GOOD BAD'
1357	002324	030330	DT115	:\$TESTN,\$ERRPC,CRCSR,GD,BD
1358	002326	000000	0	
1359				
1360	002330	025222	EM116	:'SLU2 XMIT INTERRUPTS WHEN DISABLED'
1361	002332	027431	DH5	:'TEST# ERR PC TCSR'
1362	002334	030242	DT31	:\$TESTN,\$ERRPC,TCSR
1363	002336	000000	0	
1364				
1365	002340	025265	EM117	:'SLU2 XMIT DID NOT INTERRUPT'
1366	002342	027431	DH5	:'TEST# ERR PC TCSR'
1367	002344	030242	DT31	:\$TESTN,\$ERRPC,TCSR
1368	002346	000000	0	
1369				
1370	002350	025321	EM120	:'SLU2 XMIT INTERRUPT AT PRIORITY 7'
1371	002352	027431	DH5	:'TEST# ERR PC TCSR'

1372	002354	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1373	002356	000000	0	
1374				
1375	002360	025363	EM121	;'SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR''
1376	002362	027431	DH5	;'TEST# ERR PC TCSR''
1377	002364	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1378	002366	000000	0	
1379				
1380	002370	025432	EM122	;'SLU2 XMIT DID NOT INTERRUPT''
1381	002372	027431	DH5	;'TEST# ERR PC TCSR''
1382	002374	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1383	002376	000000	0	
1384				
1385	002400	025466	EM123	;'SLU2 XMIT RE-INTERRUPTED''
1386	002402	027431	DH5	;'TEST# ERR PC TCSR''
1387	002404	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1388	002406	000000	0	
1389				
1390	002410	025517	EM124	;'LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT''
1391	002412	027431	DH5	;'TEST# ERR PC TCSR''
1392	002414	030242	DT31	;\$TESTN,\$ERRPC,TCSR
1393	002416	000000	0	
1394				
1395	002420	025571	EM125	;'SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR''
1396	002422	027456	DH6	;'TEST# ERR PC RCSR''
1397	002424	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1398	002426	000000	0	
1399				
1400	002430	025640	EM126	;'SLU2 RCVR DID NOT INTERRUPT''
1401	002432	027456	DH6	;'TEST# ERR PC RCSR''
1402	002434	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1403	002436	000000	0	
1404				
1405	002440	025674	EM127	;'SLU2 RCVR INTERRUPTS AT PRIORITY 7''
1406	002442	027456	DH6	;'TEST# ERR PC RCSR''
1407	002444	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1408	002446	000000	0	
1409				
1410	002450	025737	EM130	;'SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR''
1411	002452	027456	DH6	;'TEST# ERR PC RCSR''
1412	002454	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1413	002456	000000	0	
1414				
1415	002460	026020	EM131	;'SLU2 RCVR DID NOT INTERRUPT''
1416	002462	027456	DH6	;'TEST# ERR PC RCSR''
1417	002464	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1418	002466	000000	0	
1419				
1420	002470	026054	EM132	;'SLU2 RECEIVER RE-INTERRUPTED''
1421	002472	027456	DH6	;'TEST# ERR PC RCSR''
1422	002474	030262	DT36	;\$TESTN,\$ERRPC,RCSR
1423	002476	000000	0	
1424				
1425	002500	026111	EM133	;'SLU2 READING RBUF DID NOT CLEAR INTERRUPT''
1426	002502	027456	DH6	;'TEST# ERR PC RCSR''
1427	002504	030262	DT36	;\$TESTN,\$ERRPC,RCSR

1428	002506	000000	0	
1429				
1430	002510	026163	EM134	:'RESET DID NOT CLEAR SLU2 RCVR INTERRUPT''
1431	002512	027456	DH6	:'TEST# ERR PC RCSR''
1432	002514	030262	DT36	:\$TESTN,\$ERRPC,RCSR
1433	002516	000000	0	
1434				
1435	002520	026233	EM135	:'SLU2 'OR' FLAG DID NOT SET''
1436	002522	027456	DH6	:'TEST# ERR PC RCSR''
1437	002524	030262	DT36	:\$TESTN,\$ERRPC,RCSR
1438	002526	000000	0	
1439				
1440	002530	026266	EM136	:'SLU2 'ERROR' NOT SET WITH 'OR' FLAG''
1441	002532	027456	DH6	:'TEST# ERR PC RCSR''
1442	002534	030262	DT36	:\$TESTN,\$ERRPC,RCSR
1443	002536	000000	0	
1444				
1445	002540	026332	EM137	:'SLU2 BREAK DID NOT TRANSMIT ALL ZEROES''
1446	002542	027723	DH137	:'TEST# ERR PC RCSR DATA''
1447	002544	030344	DT137	:\$TESTN,\$ERRPC,RCSR,\$BDDAT
1448	002546	000000	0	
1449				
1450	002550	026401	EM140	:'BREAK DID NOT SET FRAMING ERROR''
1451	002552	027456	DH6	:'TEST# ERR PC RCSR''
1452	002554	030222	DT6	:\$TESTN,\$ERRPC,RCSR
1453	002556	000000	0	
1454				
1455	002560	026441	EM141	:'SLU2 'ERROR' NOT SET WITH 'FR' FLAG''
1456	002562	027456	DH6	:'TEST# ERR PC RCSR''
1457	002564	030262	DT36	:\$TESTN,\$ERRPC,RCSR
1458	002566	000000	0	
1459				
1460	002570	026505	EM142	:'DATA COMPARE ERROR WITH CABLE''
1461	002572	027760	DH142	:'TEST# ERR PC RCSR GOOD BAD''
1462	002574	030356	DT142	:\$TESTN,\$ERRPC,RCSR,GD,BD
1463	002576	000000	0	
1464				
1465	002600	026543	EM143	:'RTC INTERRUPT AT PRIORITY 7''
1466	002602	027555	DH61	:'TEST# ERR PC LKS''
1467	002604	030302	DT61	:\$TESTN,\$ERRPC,LKS
1468	002606	000000	0	
1469				
1470	002610	026577	EM144	:'RTC INTERRUPTS WHEN DISABLED''
1471	002612	027555	DH61	:'TEST# ERR PC LKS''
1472	002614	030302	DT61	:\$TESTN,\$ERRPC,LKS
1473	002616	000000	0	
1474				
1475	002620	026634	EM145	:'RTC INTERRUPT DID NOT OCCUR''
1476	002622	027555	DH61	:'TEST# ERR PC LKS''
1477	002624	030302	DT61	:\$TESTN,\$ERRPC,LKS
1478	002626	000000	0	
1479				
1480	002630	026670	EM146	:'RTC INTERRUPT DID NOT OCCUR''
1481	002632	027555	DH61	:'TEST# ERR PC LKS''
1482	002634	030302	DT61	:\$TESTN,\$ERRPC,LKS
1483	002636	000000	0	

1484				
1485	002640	026724	EM147	:'RTC DOUBLE INTERRUPT''
1486	002642	027555	DH61	:'TEST# ERR PC LKS''
1487	002644	030302	DT61	:\$TESTN,\$ERRPC,LKS
1488	002646	000000	0	
1489				
1490	002650	026751	EM150	:'RESET DID NOT CLEAR RTC INTERRUPT''
1491	002652	027555	DH61	:'TEST# ERR PC LKS''
1492	002654	030302	DT61	:\$TESTN,\$ERRPC,LKS
1493	002656	000000	0	
1494				
1495	002660	027013	EM151	:'RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS''
1496	002662	027555	DH61	:'TEST# ERR PC LKS''
1497	002664	030302	DT61	:\$TESTN,\$ERRPC,LKS
1498	002666	000000	0	
1499				
1500	002670	027070	EM152	:'CLOCK REPEATABILITY ERROR''
1501	002672	030014	DH152	:'TEST# ERR PC LKS CNT1 CNT2''
1502	002674	030370	DT152	:\$TESTN,\$ERRPC,LKS,FIRST,SECND
1503	002676	000000	0	
1504				
1505	002700	027122	EM153	:'SLU1 RECEIVER STATUS ERROR''
1506	002702	027657	DH115	:'TEST# ERR PC CRCSR GOOD BAD''
1507	002704	030330	DT115	:\$TESTN,\$ERRPC,CRCSR,\$GDDTA,\$BDDAT
1508	002706	000000	0	
1509				
1510	002710	027155	EM154	:SLU2 RECEIVER STATUS ERROR
1511	002712	027760	DH142	:'TESTN ERR PC RCSR GOOD BAD''
1512	002714	030356	DT142	:\$TESTN,\$ERRPC,RCSR,\$GDDAT,\$BDDAT
1513	002716	000000	0	
1514				
1515	002720	027210	EM155	:'INCORRECT RECEIVE COUNT SLU1
1516	002722	030061	DH155	:'TEST# ERR PC RCSR TRANS RCV''
1517	002724	030404	DT155	:\$TESTN,\$ERRPC,CRCSR,XMTCT1,RCVCT1
1518	002726	000000	0	
1519				
1520	002730	027245	EM156	:'SLU1 DATA COMPARE ERROR IN EXERCISER''
1521	002732	027657	DH115	:'TEST# ERR PC CRCSR GOOD BAD''
1522	002734	030330	DT115	:\$TESTN,\$ERRPC,CRCSR,GD,BD
1523	002736	000000	0	
1524				
1525	002740	027312	EM157	:'INCORRECT RECEIVE COUNT SLU2
1526	002742	030061	DH155	:'TEST# ERR PC RCSR TRANS RCV''
1527	002744	030420	DT157	:\$TESTN,\$ERRPC,RCSR,XMTCT2,RCVCT2
1528	002746	000000	0	
1529				
1530	002750	027347	EM160	:'SLU2 DATA COMPARE ERROR IN EXERCISER''
1531	002752	027760	DH142	:'TEST# ERR PC RCSR GOOD BAD''
1532	002754	030356	DT142	:\$TESTN,\$ERRPC,RCSR,GD,BD
1533	002756	000000	0	
1534				
1535	002760	027414	EM161	:'TRAP CATCHER''
1536	002762	030125	DH161	:'TEST# ERR PC OLDPC TRAP ADR''
1537	002764	030434	DT161	:\$TESTN,\$ERRPC,OLDPC,BDVECT
1538	002766	000000	0	
1539				


```

1540
1541
1542 002770 176500
1543 002772 176502
1544 002774 176504
1545 002776 176506
1546 003000 177560
1547 003002 177562
1548 003004 177564
1549 003006 177566
1550 003010 177546
1551
1552
1553
1554
1555 003012 000300
1556 003014 000302
1557 003016 000304
1558 003020 000306
1559 003022 000060
1560 003024 000062
1561 003026 000064
1562 003030 000066
1563 003032 000100
1564 003034 000102
1565
1566 003036 005037 001102
1567 003042 005037 001100
1568 003046 005037 001104
1569 003052 005037 001156
1570 003056
1571
1572
1573 003056 012706 001000
1574 003062 005026
1575 003064 022706 001040
1576 003070 001374
1577 003072 012706 001000
1578
1579 003076 012737 015614 000020
1580 003104 012737 000340 000022
1581 003112 012737 015120 000030
1582 003120 012737 000340 000032
1583 003126 012737 020026 000034
1584 003134 012737 000340 000036
1585 003142 012737 015436 000024
1586 003150 012737 000340 000026
1587 003156 013737 014770 014762
1588 003164 005037 001072
1589 003170 112737 000001 001015
1590 003176 012737 003176 001006
1591 003204 012737 003204 001010
1592
1593
1594 003212 013746 000004
1595 003216 012737 003252 000004

```

;REGISTER ADDRESSES OF INTERNAL ON BOARD OPTIONS

```

RCSR: .WORD 176500 ;SLU2 COMMAND/STATUS REGISTER
RBUF: .WORD 176502 ;SLU2 RECEIVER BUFFER
TCSR: .WORD 176504 ;SLU2 TRANSMITTER COMMAND/STATUS REGISTER
TBUF: .WORD 176506 ;SLU2 TRANSMITTER BUFFER
CRCSR: 177560 ;SLU1 RECEIVER COMMAND/STATUS REGISTER
CRBUF: 177562 ;SLU1 RECEIVER BUFFER
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 # $PWRDN,@#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

```



```

1596 003224 012737 177570 001040      MOV    #DSWR,SWR      ;;SETUP FOR A HARDWARE SWICH REGISTER
1597 003232 012737 177570 001042      MOV    #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1598 003240 022777 177777 175572      CMP    #-1,@SWR      ;;TRY TO REFERENCE HARDWARE SWR
1599 003246 001012                    BNE    66$           ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1600                                ;;AND THE HARDWARE SWR IS NOT = -1
1601 003250 000403                    BR     65$           ;;BRANCH IF NO TIMEOUT
1602 003252 012716 003260      54$:  MOV    #65$, (SP)   ;;SET UP FOR TRAP RETURN
1603 003256 000002                    RTI
1604 003260 012737 000176 001040      65$:  MOV    #SWREG,SWR   ;;POINT TO SOFTWARE SWR
1605 003266 012737 000174 001042      MOV    #DISPREG,DISPLAY
1606 003274 012637 000004      66$:  MOV    (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR
1607
1608 003300 005037 001106                    CLR    $PASS        ;;CLEAR PASS COUNT
1609 003304 132737 000200 001121      BITB   #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
1610 003312 001403                    BEQ    67$           ;;YES,USE NON-APT SWITCH
1611 003314 012737 001122 001040      MOV    #SSWREG,SWR   ;;NO,USE APT SWITCH REGISTER
1612 003322
1613      67$:  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1614 003322 005737 000042      TST    @#42          ;;ARE WE RUNNING UNDER XXDP/ACT?
1615 003326 001012                    BNE    68$           ;;BRANCH IF YES
1616 003330 123727 001120 000001      CMPB   $ENV,#1       ;;ARE WE RUNNING UNDER APT?
1617 003336 001406                    BEQ    68$           ;;BRANCH IF YES
1618 003340 023727 001040 000176      CMP    SWR,#SWREG    ;;SOFTWARE SWITCH REG SELECTED?
1619 003346 001005                    BNE    69$           ;;BRANCH IF NO
1620 003350 104406                    GTSWR
1621 003352 000403                    BR     69$           ;;GET SOFT-SWR SETTINGS
1622 003354 112737 000001 001034      68$:  MOVB   #1,$AUTOB    ;;SET AUTO-MODE INDICATOR
1623 003362
1624 003362 032777 000060 175450      BIT    #BIT4!BIT5,@SWR ;;IS SLU1 TO BE TESTED
1625 003370 001003                    BNE    2$           ;;IF EITHER BIT IS SET THEN DON'T TEST IT
1626 003372 052737 000001 001156      BIS    #BIT0,$DEVM   ;;SET DEVICE FLAG TO TEST SLU1
1627 003400 032777 000050 175432      2$:  BIT    #BIT3!BIT5,@SWR ;;IS SLU2 TO BE TESTED
1628 003406 001003                    BNE    3$           ;;IF EITHER BIT IS SET THEN DON'T TEST IT
1629 003410 052737 000002 001156      BIS    #BIT1,$DEVM   ;;SET DEVICE FLAG TO TEST SLU2
1630 003416 032777 000100 175414      3$:  BIT    #BIT6,@SWR    ;;IS LTC TO BE TESTED
1631 003424 001003                    BNE    4$           ;;IF BIT IS SET THEN DON'T TEST IT.
1632 003426 052737 000004 001156      BIS    #BIT2,$DEVM   ;;SET DEVICE FLAG TO TEST LTC
1633 003434 032737 000001 001156      4$:  BIT    #BIT0,$DEVM   ;;IS SLU1 UNDER TEST
1634 003442 001002                    BNE    TST1         ;;IF YES TEST XMIT REG. BEFORE TYPING TITLE
1635 003444 000137 003700                    JMP    ID            ;;IF NO SKIP TESTS AND TYPE IT NOW
  
```



```

1636
1637
1638
1639
1640
1641 003450 000004
1642 003452 012737 000001 001104
1643 003460 013703 000004
1644 003464 012737 003500 000004
1645 003472 005777 177306
1646 003476 000405
1647 003500 022626
1648 003502 004737 016004
1649 003506 000001
1650 003510 000000
1651 003512 010337 000004
1652
1653
1654
1655
1656
1657
1658 003516 000004
1659 003520 012737 000002 001104
1660 003526 013703 000004
1661 003532 012737 003546 000004
1662 003540 005777 177242
1663 003544 000405
1664 003546 022626
1665 003550 004737 016004
1666 003554 000002
1667 003556 000000
1668 003560 010337 000004
1669
1670
1671
1672
1673
1674 003564 000004
1675 003566 012737 000003 001104
1676 003574 032737 000001 001120
1677 003602 001405
1678 003604 005737 001106
1679 003610 001402
1680 003612 000137 004726
1681 003616 005077 177164
1682 003622 105777 177156
1683 003626 100011
1684
1685
1686 003630 005077 177152
1687 003634 105777 177144
1688 003640 100004
1689 003642 004737 016004
1690 003646 000003
1691 003650 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
    
```



```
1692 003652 005000 177124 1$: CLR R0 ;CLEAR TIMER
1693 003654 105777 177124 2$: TSTB @CTCSR ;CHECK FOR XMIT DONE
1694 003660 100407 BMI ID ;IF DONE SETS, BR TO END OF TEST
1695 003662 005200 INC R0 ;INCREMENT TIMER
1696 003664 001373 BNE 2$ ;BR IF TIMER NOT DONE
1697 003666 004737 016004 JSR PC,$ATY4 ;:ONLY REPORT A FATAL ERROR
1698 003672 000004 4 ;:THE ERROR NUMBER (FROM APT LIST)
1699 003674 000000 HALT
1700 003676 000416 BR TST4 ;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
1701 ; BECAUSE OF THIS FAILURE
1702
1703 003700 023737 000042 000046 ID: CMP @#42,@#46 ;UNDER ACT11?
1704 003706 001412 BEQ 6$ ;IF YES, SKIP IDENT. TYPEOUT
1705 003710 005737 001106 TST $PASS ;IS THIS THE FIRST PASS?
1706 003714 001007 BNE 6$ ;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
1707 003716 005737 001110 TST $DEVCT ;IS THIS THE FIRST SUBPASS?
1708 003722 001004 BNE 6$ ;IF NOT, BR TO NEXT TEST
1709 003724 104401 TYPE ;TYPE PROGRAM IDENTIFICATION
1710 003726 030176 M1
1711 003730 104401 TYPE ;TYPE NUMBER OF DEVICES UNDER TEST
1712 003732 030210 M2
1713 003734 6$:
1714
1715 ;:*****
1716 ;*TEST 4 TEST THAT SLU1 TCSR 'DONE' SETS WITH RESET
1717 ;:*****
1718 003734 000004 TST4: SCOPE
1719 003736 012737 000004 001104 MOV #4,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1720 003744 032737 000001 001156 BIT #BIT0,@#$DEVM ;DO THESE TESTS FOR THIS DEVICE?
1721 003752 001002 BNE 99$ ;IF YES CONTINUE WITH TESTS
1722 003754 000137 004726 JMP SLU2RT ;IF NO GO TO START OF NEXT SET OF TESTS.
1723 003760 99$:
1724 003760 005077 177022 CLR @CTBUF ;LOAD TRANSMIT BUFFER
1725 003764 105777 177014 TSTB @CTCSR ;WAIT FOR DONE
1726 003770 100375 BPL 1$
1727 003772 005077 177010 CLR @CTBUF ;LOAD SECOND BUFFER
1728 003776 000240 NOP
1729 004000 000005 RESET ;CLEAR DONE WITH RESET
1730 004002 105777 176776 TSTB @CTCSR ;CHECK FOR DONE SET
1731 004006 100401 BMI TST5 ;BR TO NEXT TEST IF DONE SET
1732
1733 004010 104005 ERROR 5 ;TCSR 'DONE' DOES NOT SET WITH RESET
1734
1735
1736
1737 ;:*****
1738 ;*TEST 5 TEST ABILITY TO ACCESS SLU1 RCSR
1739 ;:*****
1740 004012 000004 TST5: SCOPE
1741 004014 012737 000005 001104 MOV #5,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
1742 004022 013703 000004 MOV @#4,R3 ;SAVE TIMEOUT VECTOR
1743 004026 012737 004042 000004 MOV #1$,@#4 ;SET UP TIMEOUT VECTOR
1744 004034 005777 176740 TST @RCSR ;ACCESS RCSR
1745 004040 000402 BR 2$ ;BR TO END OF TEST
1746
1747 004042 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
```


1748 004044 104006
1749 004046 010337 000004

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

1750
1751

1752
1753

*TEST 6 TEST ABILITY TO ACCESS SLU1 RBUF

1754
1755 004052 000004
1756 004054 012737 000006 001104
1757 004062 013703 000004
1758 004066 012737 004102 000004
1759 004074 005777 176702
1760 004100 000402

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

1761
1762 004102 022626
1763 004104 104007
1764 004106 010337 000004

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

1765
1766

1767
1768

1769
1770

*TEST 7 TEST THAT SLU1 BIT2(MAINT. BIT) CAN BE SET & RESET

1771 004112 000004
1772 004114 012737 000007 001104
1773 004122 042777 000004 176654
1774 004130 032777 000004 176646
1775 004136 001404
1776 004140 004737 016004
1777 004144 000010

TST7: SCOPE
MOV #7,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT2,@CTCSR ;:MAKE SURE BIT UNDER TEST IS INITIALIZED
BIT #BIT2,@CTCSR ;:TEST FOR BIT2 OF TCSR CLEAR
BEQ 1\$;:BR IF CLEAR
JSR PC,\$ATY4 ;:ONLY REPORT A FATAL ERROR
10 ;:THE ERROR NUMBER (FROM APT LIST)

1778 004146 000000
1779 004150 052777 000004 176626
1780 004156 032777 000004 176620
1781 004164 001001
1782 004166 104011
1783 004170 042777 000004 176606
1784 004176 032777 000004 176600

1\$: BIS #BIT2,@CTCSR ;:SET BIT2 OF TCSR
BIT #BIT2,@CTCSR ;:TEST FOR BIT2 SET
BNE 2\$;:BR IF SET
ERROR 11 ;:BIT2 OF TCSR WILL NOT SET
2\$: BIC #BIT2,@CTCSR ;:CLEAR BIT2 OF TCSR
BIT #BIT2,@CTCSR ;:TEST BIT2 CLEAR
BEQ 3\$;:BR IF CLEAR

1785 004204 001404
1786 004206 004737 016004
1787 004212 000012
1788 004214 000000
1789 004216 052777 000004 176560
1790 004224 000005
1791 004226 032777 000004 176550

3\$: BIS #BIT2,@CTCSR ;:SET BIT2 OF TCSR
RESET ;:CLEAR BIT2 WITH RESET
BIT #BIT2,@CTCSR ;:TEST FOR BIT2 CLEAR
BEQ TST10 ;:IF CLEAR, GO TO NEXT TEST
BIC #BIT2,@CTCSR ;:CLEAR BIT2, TO PRINT ERROR

1792 004234 001404
1793 004236 042777 000004 176540
1794 004244 104013
1795
1796
1797

;RESET DID NOT CLEAR BIT2 OF TCSR

1798
1799

*TEST 10 TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET

1800
1801 004246 000004
1802 004250 012737 000010 001104
1803 004256 042777 000100 176520

TST10: SCOPE
MOV #10,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@CTCSR ;:MAKE SURE BIT UNDER TEST IS INITIALIZED


```

1804 004264 017703 176536      MOV    @CTVECT,R3      ;SAVE XMIT VECTOR
1805 004270 012777 004320 176530  MOV    #1$,@CTVECT    ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1806 004276 004737 015056      JSR    PC,WRPSW        ;SET PSW TO PRIORITY=7
1807 004302 000340      .WORD 340
1808 004304 032777 000100 176472  BIT    #BIT6,@CTCSR    ;TEST BIT6 OF TCSR
1809 004312 001404      BEQ    2$              ;BR IF ZERO
1810 004314 104014      ERROR 14
1811                                ;BIT6 IN TCSR NOT CLEAR AFTER RESET
1812 004316 000402      BR     2$
1813
1814 004320 022626      1$:   CMP    (SP)+,(SP)+    ;RESTORE SP AFTER INTERRUPT
1815 004322 104015      ERROR 15
1816                                ;XMIT INTERRUPT OCCURRED PRIO=7
1817
1818 004324 052777 000100 176452  2$:   BIS    #BIT6,@CTCSR  ;SET BIT6 OF TCSR
1819 004332 032777 000100 176444  BIT    #BIT6,@CTCSR  ;TEST BIT6 OF TCSR
1820 004340 001001      BNE    3$              ;BR, IF SET
1821
1822 004342 104016      ERROR 16
1823                                ;CANNOT SET BIT6 OF TCSR
1824
1825 004344 042777 000100 176432  3$:   BIC    #BIT6,@CTCSR  ;CLEAR BIT6 OF TCSR
1826 004352 032777 000100 176424  BIT    #BIT6,@CTCSR  ;TEST BIT6 OF TCSR
1827 004360 001401      BEQ    4$              ;BR IF CLEAR
1828 004362 104017      ERROR 17
1829                                ;CANNOT CLEAR BIT6 OF TCSR
1830
1831 004364      4$:
1832 004364 052777 000100 176412  BIS    #BIT6,@CTCSR  ;SET BIT6 OF TCSR
1833 004372 000005      RESET ;CLEAR BIT6 WITH RESET
1834 004374 032777 000100 176402  BIT    #BIT6,@CTCSR  ;TEST BIT6 OF TCSR
1835 004402 001401      BEQ    5$              ;BR IF CLEAR
1836
1837 004404 104020      ERROR 20
1838                                ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1839 004406 010377 176414      5$:   MOV    R3,@CTVECT    ;RESTORE XMIT VECTOR
1840
1841
1842
1843
1844
1845 004412 000004      TST11: SCOPE
1846 004414 012737 000011 001104  MOV    #11,$TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
1847 004422 042777 000100 176350  BIC    #BIT6,@CRCSR  ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1848 004430 017703 176366      MOV    @CRVECT,R3    ;SAVE RECEIVE VECTOR
1849 004434 012777 004464 176360  MOV    #1$,@CRVECT    ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1850 004442 004737 015056      JSR    PC,WRPSW        ;SET PSW TO PRIORITY=7
1851 004446 000340      .WORD 340
1852 004450 032777 000100 176322  BIT    #BIT6,@CRCSR  ;TEST BIT6 OF RCSR
1853 004456 001404      BEQ    2$
1854 004460 104021      ERROR 21
1855                                ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1856 004462 000402      BR     2$
1857
1858 004464 022626      1$:   CMP    (SP)+,(SP)+    ;RESTORE SP AFTER INTERRUPT
1859 004466 104022      ERROR 22

```

```

*****
*TEST 11 TEST THAT SLU1 BIT6 OF RCSR CAN BE SET & RESET
*****

```

```

TST11: SCOPE
MOV    #11,$TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
BIC    #BIT6,@CRCSR  ;MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV    @CRVECT,R3    ;SAVE RECEIVE VECTOR
MOV    #1$,@CRVECT    ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR    PC,WRPSW        ;SET PSW TO PRIORITY=7
WORD 340
BIT    #BIT6,@CRCSR  ;TEST BIT6 OF RCSR
BEQ    2$
ERROR 21
;BIT6 OF RCSR NOT CLEAR AFTER RESET
BR     2$
1$:   CMP    (SP)+,(SP)+    ;RESTORE SP AFTER INTERRUPT
ERROR 22

```



```
1860 ;RCVR INTERRUPT WITH PRIORITY=7
1861
1862 004470 052777 000100 176302 2$: BIS #BIT6,@CRCSR ;SET BIT6 OF RCSR
1863 004476 032777 000100 176274 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1864 004504 001001 BNE 3$ ;BR, IF SET
1865
1866 004506 104023 ERROR 23 ;CANNOT SET BIT6 OF RCSR
1867
1868
1869 004510 042777 000100 176262 3$: BIC #BIT6,@CRCSR ;CLEAR BIT6 OF RCSR
1870 004516 032777 000100 176254 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1871 004524 001401 BEQ 4$ ;BR, IF CLEAR
1872
1873 004526 104024 ERROR 24 ;CANNOT CLEAR BIT6 OF RCSR
1874
1875
1876 004530 4$:
1877 004530 052777 000100 176242 BIS #BIT6,@CRCSR ;SET BIT6 OF RCSR
1878 004536 000005 RESET ;CLEAR BIT6 OF RCSR WITH RESET
1879 004540 032777 000100 176232 BIT #BIT6,@CRCSR ;TEST BIT6 OF RCSR
1880 004546 001401 BEQ 5$ ;BR, IF CLEAR
1881
1882 004550 104025 ERROR 25 ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1883
1884 004552 010377 176244 5$: MOV R3,@CRVECT ;RESTORE RECEIVE VECTOR
1885
1886
1887
1888
1889
1890
1891 004556 000004
1892 004560 012737 000012 001104
1893 004566 052777 000004 176210
1894 004574 005000
1895 004576 005077 176204
1896 004602 105777 176172 CWDONE:
1897 004606 100406 BMI 1$
1898 004610 005200 INC R0 ;INCREMENT TIMER, IF NOT DONE
1899 004612 001373 BNE CWDONE ;CONTINUE WAIT IF TIME REMAINS
1900 004614 042777 000004 176162 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1901 004622 104026 ERROR 26 ;RECEIVER DONE NEVER SET
1902
1903
1904 004624 000005 1$: RESET ;CLEAR DONE WITH RESET
1905 004626 105777 176146 TSTB @CRCSR ;CHECK FOR DONE CLEAR
1906 004632 001404 BEQ 2$
1907
1908 004634 042777 000004 176142 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1909 004642 104027 ERROR 27 ;RESET DID NOT CLEAR RCVR DONE
1910
1911
1912 004644 042777 000004 176126 2$: BIC #BIT2,@CRCSR ;CLEAR MAINTENANCE BIT
1913
1914
1915
```

```
::*****
:*TEST 12 TEST THAT SLU1 RCVR DONE (7) SET & CLEAR PROPERLY
:*****
```

```
TST12: SCOPE
MOV #12,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIS #BIT2,@CTCSR ;:TURN ON INTERNAL MAINTENENCE WRAP AROUND
CLR R0 ;:CLEAR TIMER
CLR @CTBUF ;:LOAD TRANSMIT BUFFER
TSTB @CRCSR ;:CHECK FOR RECEIVER DONE
BMI 1$ ;:BR, IF DONE
INC R0 ;:INCREMENT TIMER, IF NOT DONE
BNE CWDONE ;:CONTINUE WAIT IF TIME REMAINS
BIC #BIT2,@CTCSR ;:CLEAR MAINTENANCE BIT
ERROR 26 ;:RECEIVER DONE NEVER SET

1$: RESET ;:CLEAR DONE WITH RESET
TSTB @CRCSR ;:CHECK FOR DONE CLEAR
BEQ 2$

BIC #BIT2,@CTCSR ;:CLEAR MAINTENANCE BIT
ERROR 27 ;:RESET DID NOT CLEAR RCVR DONE

2$: BIC #BIT2,@CRCSR ;:CLEAR MAINTENANCE BIT
```

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


1916
1917
1918 004652 000004
1919 004654 012737 000013 001104
1920 004662 000005
1921 004664 052777 000004 176112
1922 004672 005077 176110
1923 004676 105777 176076
1924 004702 100375
1925 004704 017700 176072
1926 004710 042777 000004 176066
1927 004716 105777 176056
1928 004722 001401
1929 004724 104030
1930

```
;*TEST 13 TEST SLU1 THAT READING RBUF CLEARS RECEIVER DONE
:*****
TST13: SCOPE
MOV #13,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
RESET ;;CLEAR EVERYTHING
BIS #BIT2,@CTCSR ;;SET MAINTENANCE WRAP
CLR @CTB,IF ;;LOAD TRANSMITTER
1$: TSTB @CRCSR ;;WAIT FOR RECEIVER DONE
BPL 1$
MOV @CRBUF,RO ;;READ RECEIVE BUFFER
BIC #BIT2,@CTCSR ;;CLEAR MAINTENANCE BIT
TSTB @CRCSR ;;CHECK FOR RECEIVE DONE CLEAR
BEQ TST14 ;;BR, IF CLEAR TO NEXT TEST
ERROR 30
;;READING RBUF DID NOT CLEAR RCVR DONE
```



```

1931 004726
1932
1933
1934
1935
1936 004726 000004
1937 004730 012737 000014 001104
1938 004736 032737 000002 001156
1939 004744 001002
1940 004746 000137 006152
1941 004752
1942 004752 013703 000004
1943 004756 012737 004772 000004
1944 004764 005777 176004
1945 004770 000402
1946 004772 022626
1947 004774 104031
1948 004776 010337 000004
1949
1950
1951
1952
1953
1954
1955 005002 000004
1956 005004 012737 000015 001104
1957 005012 013703 000004
1958 005016 012737 005032 000004
1959 005024 005777 175746
1960 005030 000402
1961 005032 022626
1962 005034 104032
1963 005036 010337 000004
1964
1965
1966
1967
1968
1969 005042 000004
1970 005044 012737 000016 001104
1971 005052 032737 000001 001120
1972 005060 001403
1973 005062 005737 001106
1974 005066 001022
1975 005070
1976 005070 005077 175702
1977 005074 105777 175674
1978 005100 100006
1979
1980
1981 005102 005077 175670
1982 005106 105777 175662
1983 005112 100001
1984 005114 104033
1985 005116 005000
1986 005120 105777 175650
    
```

SLU2RT:

```

*****
:*TEST 14 TEST ABILITY TO ACCESS SLU2 TCSR
*****
TST14: SCOPE
MOV #14,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #BIT1,@#$DEV M ;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 15 TEST ABILITY TO ACCESS SLU2 TBUF
*****
TST15: SCOPE
MOV #15,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;SAVE TIMEOUT VECTOR
MOV #1$,@#4 ;SET UP TIMEOUT VECTOR
TST @TBUF ;REFERENCE THE XMIT BUFFER
BR 2$ ;GO TO END OF TEST
1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
ERROR 32 ;XMIT BUFFER ADDRESS DOES NOT REUTRN SSYNC
2$: MOV R3,@#4 ;RESTORE TIMEOUT VECTOR
    
```

```

*****
:*TEST 16 TEST SLU2 TCSR BIT7(DONE) CLEARS WHEN XBUF IS LOADED
*****
TST16: SCOPE
MOV #16,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#$ENV ;ARE WE RUNNING UNDER APT
BEQ 70$ ;IF NO THEN DO TEST
TST @#$PASS ;IS THIS FIRST PASS
BNE TST17 ;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
    
```


1987 005124 100403
1988 005126 005200
1989 005130 001373
1990 005132 104034
1991
1992
1993
1994
1995
1996 005134 000004
1997 005136 012737 000017 001104
1998 005144 032737 000001 001120
1999 005152 001403
2000 005154 005737 001106
2001 005160 001015
2002 005162
2003 005162 005077 175610
2004 005166 105777 175602
2005 005172 100375
2006 005174 005077 175576
2007 005200 000240
2008 005202 000005
2009 005204 105777 175564
2010 005210 100401
2011
2012 005212 104035
2013
2014
2015
2016
2017
2018
2019 005214 000004
2020 005216 012737 000020 001104
2021 005224 013703 000004
2022 005230 012737 005244 000004
2023 005236 005777 175526
2024 005242 000402
2025
2026 005244 022626
2027 005246 104036
2028 005250 010337 000004
2029
2030
2031
2032
2033
2034
2035 005254 000004
2036 005256 012737 000021 001104
2037 005264 013703 000004
2038 005270 012737 005304 000004
2039 005276 005777 175470
2040 005302 000402
2041
2042 005304 022626

BMI TST17 ;IF DONE SETS, BR TO NEXT TEST
INC RO ;INCREMENT TIMER
BNE 2\$;BR IF TIMER NOT DONE
ERROR 34 ;XMIT DONE BIT DOES NOT RESET AFTER TRANSMIT

*TEST 17 TEST THAT SLU2 TCSR 'DONE' SETS WITH RESET

TST17: SCOPE
MOV #17,\$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 TST20 ;:IF NO THEN SKIP TO NEXT TEST
70\$:
CLR @TBUF ;:LOAD TRANSMIT BUFFER
1\$: TSTB @TCSR ;:WAIT FOR DONE
BPL 1\$
CLR @TBUF ;:LOAD SECOND BUFFER
NOP
RESET ;:CLEAR DONE WITH RESET
TSTB @TCSR ;:CHECK FOR DONE SET
BMI TST20 ;:BR TO NEXT TEST IF DONE SET
ERROR 35 ;:TCSR 'DONE' DOES NOT SET WITH RESET

*TEST 20 TEST ABILITY TO ACCESS SLU2 RCSR

TST20: SCOPE
MOV #20,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;:SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;:SET UP TIMEOUT VECTOR
TST @RCSR ;:ACCESS RCSR
BR 2\$;:BR TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER TIMEOUT
ERROR 36 ;:CAN NOT ACCESS RCSR
2\$: MOV R3,@#4 ;:RESTORE TIMEOUT VECTOR

*TEST 21 TEST ABILITY TO ACCESS SLU2 RBUF

TST21: SCOPE
MOV #21,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
MOV @#4,R3 ;:SAVE TIMEOUT VECTOR
MOV #1\$,@#4 ;:SET UP TIMEOUT VECTOR
TST @RBUF ;:ACCESS RBUF
BR 2\$;:BR TO END OF TEST
1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER TIMEOUT

2043 005306 104037
2044 005310 010337 000004

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

2045
2046
2047

2048
2049
2050

: *TEST 22 TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET

2051 005314 000004
2052 005316 012737 000022 001104
2053 005324 042777 000001 175442
2054 005332 032777 000001 175434

TST22: SCOPE
MOV #22,\$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

2055 005340 001401
2056 005342 104040
2057 005344 052777 000001 175422
2058 005352 032777 000001 175414

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

2059 005360 001001
2060 005362 104041
2061 005364 042777 000001 175402
2062 005372 032777 000001 175374

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\$

2063 005400 001401
2064 005402 104042
2065 005404
2066 005404 032737 000001 001120
2067 005412 001403

ERROR 42 ;:BIT0 OF TCSR WILL NOT CLEAR
3\$: BIT #1,@#\$ENV ;:ARE WE RUNNING UNDER APT
BEQ 70\$;:IF NO THEN DO TEST
TST @#\$PASS ;:IS THIS FIRST PASS
BNE TST23 ;:IF NO THEN SKIP TO NEXT TEST

2068 005414 005737 001106
2069 005420 001014
2070 005422
2071 005422 052777 000001 175344
2072 005430 000005

70\$: BIS #BIT0,@TCSR ;:SET BIT0 IN TCSR
RESET ;:CLEAR BIT0 WITH RESET
BIT #BIT0,@TCSR ;:TEST BIT0 CLEAR
BEQ TST23 ;:BR IF CLEAR

2073 005432 032777 000001 175334
2074 005440 001404
2075 005442 042777 000001 175324
2076 005450 104043

BIC #BIT0,@TCSR ;:CLEAR BIT0, TO PRINT ERROR
ERROR 43 ;:RESET DID NOT CLEAR BIT0 OF TCSR

2077
2078
2079

: *TEST 23 TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET

2080
2081
2082 005452 000004
2083 005454 012737 000023 001104
2084 005462 042777 000100 175304

TST23: SCOPE
MOV #23,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@TCSR ;:MAKE SURE BIT UNDER TEST IS INITIALIZED

2085 005470 017703 175322
2086 005474 012777 005524 175314
2087 005502 004737 015056
2088 005506 000340

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

2089 005510 032777 000100 175256
2090 005516 001404
2091 005520 104044
2092

BIT #BIT6,@TCSR ;:TEST BIT6 OF TCSR
BEQ 2\$;:BR IF ZERO
ERROR 44 ;:BIT6 IN TCSR NOT CLEAR AFTER RESET

2093 005522 000402
2094
2095 005524 022626
2096 005526 104045
2097
2098

BR 2\$
1\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
ERROR 45 ;:XMIT INTERRUPT OCCURRED PRIO=7


```
2099 005530 052777 000100 175236 2$: BIS #BIT6,@TCSR ;SET BIT6 OF TCSR
2100 005536 032777 000100 175230 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
2101 005544 001001 BNE 3$ ;BR, IF SET
2102
2103 005546 104046 ERROR 46 ;CANNOT SET BIT6 OF TCSR
2104
2105
2106 005550 042777 000100 175216 3$: BIC #BIT6,@TCSR ;CLEAR BIT6 OF TCSR
2107 005556 032777 000100 175210 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
2108 005564 001401 BEQ 4$ ;BR IF CLEAR
2109 005566 104047 ERROR 47 ;CANNOT CLEAR BIT6 OF TCSR
2110
2111
2112 005570 032737 000001 001120 4$: BIT #1,@#SENV ;ARE WE RUNNING UNDER APT
2113 005576 001403 BEQ 70$ ;IF NO THEN DO TEST
2114 005600 005737 001106 TST @#SPASS ;IS THIS FIRST PASS
2115 005604 001011 BNE 5$ ;IF NO THEN SKIP TO END OF TEST
2116 005606
2117 005606 052777 000100 175160 70$: BIS #BIT6,@TCSR ;SET BIT6 OF TCSR
2118 005614 000005 RESET ;CLEAR BIT6 WITH RESET
2119 005616 032777 000100 175150 BIT #BIT6,@TCSR ;TEST BIT6 OF TCSR
2120 005624 001401 BEQ 5$ ;BR IF CLEAR
2121
2122 005626 104050 ERROR 50 ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
2123
2124 005630 010377 175162 5$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
2125
2126
2127
2128
2129
2130 005634 000004
2131 005636 012737 000024 001104
2132 005644 042777 000100 175116
2133 005652 017703 175134
2134 005656 012777 005706 175126
2135 005664 004737 015056
2136 005670 000340
2137 005672 032777 000100 175070
2138 005700 001404
2139 005702 104051
2140
2141 005704 000402
2142
2143 005706 022626 1$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2144 005710 104052 ERROR 52 ;RCVR INTERRUPT WITH PRIORITY=7
2145
2146
2147 005712 052777 000100 175050 2$: BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
2148 005720 032777 000100 175042 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
2149 005726 001001 BNE 3$ ;BR, IF SET
2150
2151 005730 104053 ERROR 53 ;CANNOT SET BIT6 OF RCSR
2152
2153
2154 005732 042777 000100 175030 3$: BIC #BIT6,@RCSR ;CLEAR BIT6 OF RCSR
```

```
::*****
:*TEST 24 TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
:*****
TST24: SCOPE
```

```
MOV #24,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@RCSR ;:MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV @RVECT,R3 ;:SAVE RECEIVE VECTOR
MOV #1$,@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$
CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
ERROR 52 ;:RCVR INTERRUPT WITH PRIORITY=7
```

```
BIS #BIT6,@RCSR ;:SET BIT6 OF RCSR
BIT #BIT6,@RCSR ;:TEST BIT6 OF RCSR
BNE 3$ ;:BR, IF SET
```

```
ERROR 53 ;:CANNOT SET BIT6 OF RCSR
```

```
BIC #BIT6,@RCSR ;:CLEAR BIT6 OF RCSR
```



```
2155 005740 032777 000100 175022 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
2156 005746 001401 BEQ 4$ ;BR, IF CLEAR
2157
2158 005750 104054 ERROR 54 ;CANNOT CLEAR BIT6 OF RCSR
2159
2160
2161 005752 032737 000001 001120 4$: BIT #1,@#SENV ;ARE WE RUNNING UNDER APT
2162 005760 001403 BEQ 70$ ;IF NO THEN DO TEST
2163 005762 005737 001106 TST @#$PASS ;IS THIS FIRST PASS
2164 005766 001011 BNE 5$ ;IF NO THEN SKIP TO END OF TEST
2165 005770 70$:
2166 005770 052777 000100 174772 BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
2167 005776 000005 RESET ;CLEAR BIT6 OF RCSR WITH RESET
2168 006000 032777 000100 174762 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
2169 006006 001401 BEQ 5$ ;BR, IF CLEAR
2170
2171 006010 104055 ERROR 55 ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
2172
2173 006012 010377 174774 5$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
2174
2175
2176
2177
```

```
::*****
:*TEST 25 TEST THAT SLU2 RCVR DONE (7) SET & CLEAR PROPERLY
:*****
```

```
2178
2179
2180 006016 000004 TST25: SCOPE
2181 006020 012737 000025 001104 MOV #25,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
2182 006026 005077 174740 CLR @RBUF ;:INITIALIZE REGISTER BEFORE TESTING
2183 006032 005000 CLR R0 ;:CLEAR TIMER
2184 006034 005077 174736 CLR @TBUF ;:LOAD TRANSMIT BUFFER
2185 006040 105777 174724 WDONE: TSTB @RCSR ;:CHECK FOR RECEIVER DONE
2186 006044 100403 BMI 1$ ;:BR, IF DONE
2187 006046 005200 INC R0 ;:INCREMENT TIMER, IF NOT DONE
2188 006050 001373 BNE WDONE ;:CONTINUE WAIT IF TIME REMAINS
2189 006052 104056 ERROR 56 ;:RECEIVER DONE NEVER SET
2190
```

```
2191
2192 006054 032737 000001 001120 1$: BIT #1,@#SENV ;ARE WE RUNNING UNDER APT
2193 006062 001403 BEQ 70$ ;IF NO THEN DO TEST
2194 006064 005737 001106 TST @#$PASS ;IS THIS FIRST PASS
2195 006070 001005 BNE 2$ ;IF NO THEN SKIP TO END OF TEST
2196 006072 70$:
2197 006072 000005 RESET ;:CLEAR DONE WITH RESET
2198 006074 105777 174670 TSTB @RCSR ;:CHECK FOR DONE CLEAR
2199 006100 001401 BEQ 2$
2200
2201 006102 104057 ERROR 57 ;:RESET DID NOT CLEAR RCVR DONE
2202
2203 006104 005777 174662 2$: TST @RBUF ;:CLEAR RECEIVER BUFFER
2204
2205
2206
2207
```

```
::*****
:*TEST 26 TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
:*****
```

```
2208
2209
2210 006110 000004 TST26: SCOPE
```


2211 006112 012737 000026 001104
2212 006120 005077 174646
2213 006124 005077 174646
2214 006130 105777 174634
2215 006134 100375
2216 006136 017700 174630
2217 006142 105777 174622
2218 006146 001401
2219 006150 104060
2220
2221 006152
2222
2223
2224
2225
2226 006152 000004
2227 006154 012737 000027 001104
2228 006162 032737 000004 001156
2229 006170 001002
2230 006172 000137 006524
2231 006176
2232 006176 013703 000004
2233 006202 012737 006216 000004
2234 006210 005777 174574
2235 006214 000402
2236
2237 006216 022626
2238 006220 104061
2239
2240 006222 010337 000004
2241
2242
2243
2244
2245 006226 000004
2246 006230 012737 000030 001104
2247 006236 042777 000100 174544
2248 006244 017703 174562
2249 006250 012777 006300 174554
2250 006256 004737 015056
2251 006262 000340
2252 006264 032777 000100 174516
2253 006272 001404
2254 006274 104062
2255
2256 006276 000402
2257
2258 006300 022626
2259 006302 104063
2260
2261
2262 006304 052777 000100 174476
2263 006312 032777 000100 174470
2264 006320 001001
2265
2266 006322 104064

```
MOV #26,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
CLR @RBUF ;;INITIALIZE REGISTER BEFORE TESTING
CLR @TBUF ;;LOAD TRANSMITTER
1$: TSTB @RCSR ;;WAIT FOR RECEIVER DONE
BPL 1$
MOV @RBUF,R0 ;;READ RECEIVE BUFFER
TSTB @RCSR ;;CHECK FOR RECEIVE DONE CLEAR
BEQ TST27 ;;BR, IF CLEAR TO NEXT TEST
ERROR 60 ;;READING RBUF DID NOT CLEAR RCVR DONE

LTCRT:
*****
:*TEST 27 TEST ABILITY TO ACCESS LKS
*****
TST27: SCOPE
MOV #27,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #BIT2,@#$DEVN ;;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 30 TEST THAT BIT6 OF LKS CAN BE SET & RESET
*****
TST30: SCOPE
MOV #30,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@LKS ;;MAKE SURE BIT UNDER TEST IS INITIALIZED BEFORE TESTING
MOV @RTCVT,R3 ;;SAVE LINE CLOCK VECTOR
MOV #1$,@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
```



```

2267                                     ;CANNOT SET BIT6 OF LKS
2268
2269 006324 042777 000100 174456 3$:   BIC   #BIT6,@LKS   ;CLEAR BIT6 OF LKS
2270 006332 032777 000100 174450       BIT   #BIT6,@LKS   ;TEST BIT6 OF LK
2271 006340 001401       BEQ   4$
2272 006342 104065       ERROR  65
2273                                     ;CANNOT CLEAR BIT6 OF LKS
2274 006344 032737 000001 001120 4$:   BIT   #1,@#$ENV   ;ARE WE RUNNING UNDER APT
2275 006352 001403       BEQ   70$           ;IF NO THEN DO TEST
2276 006354 005737 001106       TST  @#$PASS      ;IS THIS FIRST PASS
2277 006360 001011       BNE   5$           ;IF NO THEN SKIP TO END OF TEST
2278 006362
2279 006362 052777 000100 174420 70$:  BIS   #BIT6,@LKS   ;SET BIT6 OF LKS
2280 006370 000005       RESET
2281 006372 032777 000100 174410       BIT   #BIT6,@LKS   ;CLEAR BIT6 OF LKS WITH RESET
2282 006400 001401       BEQ   5$           ;TEST BIT6 OF LKS
2283                                     ;BR IF CLEAR
2284 006402 104066       ERROR  66
2285                                     ;CANNOT CLEAR BIT6 OF LKS WITH RESET
2286 006404 010377 174422 5$:   MOV   R3,@RTCVT   ;RESTORE LINE CLOCK VECTOR
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322

```

 *TEST 31 TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED

 TST31: SCOPE
 MOV #31,\$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 TST32 ;:IF NO THEN SKIP TO NEXT TEST
 70\$:
 RESET ;:CLEAR EVERYTHING & SET BIT7 OF LKS
 1\$: TSTB @LKS ;:TEST FOR BIT7 OF LKS
 BMI 2\$;:BR IF SET
 ERROR 67 ;:BIT7 OF LKS DID NOT SET WITH RESET
 2\$: BIC #BIT7,@LKS ;:CLEAR BIT7 OF LKS
 BIT #BIT7,@LKS ;:TEST BIT7 OF LKS
 BEQ 3\$
 BIC #BIT7,@LKS ;:TRY ONE MORE TIME BECAUSE THE CLOCK
 BIT #BIT7,@LKS ;:MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
 BEQ 3\$
 ERROR 70 ;:CAN NOT CLEAR BIT7 OF LKS
 3\$: CLR R0 ;:CLEAR TIMER
 CONT: TSTB @LKS ;:TEST FOR BIT7 OF LKS
 BMI TST32 ;:BR, IF SET
 INC R0 ;:INCREMENT TIMER
 BNE CONT ;:CONTINUE UNTIL TIME EXPIRES
 ERROR 71 ;:BIT7 OF LKS DOES NOT SET

2323
2324 006524
2325
2326
2327
2328
2329 006524 000004
2330 006526 012737 000032 001104
2331 006534 032737 000001 001120
2332 006542 001403
2333 006544 005737 001106
2334 006550 001053
2335 006552
2336 006552 012737 000340 177776
2337
2338 006560 012700 002770
2339 006564 012703 002770
2340
2341 006570 012701 000011
2342 006574 005033
2343 006576 077102
2344 006600 012770 000100 000000
2345 006606 012701 002770
2346 006612 012702 000011
2347 006616 032731 000100
2348 006622 001006
2349 006624 077204
2350 006626 020027 003010
2351 006632 001422
2352 006634 005030
2353
2354 006636 000752
2355 006640 021041
2356 006642 001413
2357 006644 011037 001020
2358 006650 011137 001022
2359 006654 017037 000000 001024
2360 006662 017137 000000 001026
2361 006670 104072
2362
2363 006672 062701 000002
2364 006676 000752
2365 006700
2366

UNIQUE:

```
*****  
: *TEST 32          UNIQUE INTERNAL ADDRESS TEST  
*****  
TST32: SCOPE  
MOV      #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX  
BIT      #1,@#$ENV      ;ARE WE RUNNING UNDER APT  
BEQ      70$            ;IF NO THEN DO TEST  
TST      @#$PASS        ;IS THIS FIRST PASS  
BNE      TST33          ;IF NO THEN SKIP TO NEXT TEST  
  
70$:     MOV      #340,PS      ;WE WILL BE PLAYING WITH BIT6  
                ;SO LOCK OUT EXTRANEIOUS INTERRUPTS  
                ;GET LOCATION OF FIRST REGISTER ADDRESS  
1$:     MOV      #RCSR,R0      ;MAKE R3 POINT TO LOCATION OF FIRST  
                ;REGISTER ADDRESS  
                ;SET LOOP COUNTER TO CLEAR ALL REG.  
2$:     CLR      @ (R3)+      ;CLEAR A REGISTER  
                ;LOOP UNTIL ALL REGISTERS CLEARED  
                ;SET TEST BIT IN DEVICE REGISTERS  
                ;GET LOCATION OF FIRST REGISTER ADDRESS  
                ;SET UP TEST LOOP COUNTER  
3$:     MOV      #BIT6,@ (R0)  ;IS TEST BIT SET IN THIS REGISTER  
                ;IF YES GO SEE IF THERE IS AN ERROR  
                ;LOOP UNTIL ALL REGISTER CHECKED  
4$:     MOV      #RCSR,R1      ;ARE WE DONE TESTING  
                ;IF YES GO TO NEXT TEST  
                ;CLEAR REGISTER JUST TESTED AND POINT  
                ;TO NEXT ONE  
                ;CONTINUE TESTING  
5$:     BR       1$           ;DID WE COMPARE THE REGISTER TO ITSELF?  
                ;IF YES GET OVER ERROR CALL  
                ;IF NO SET UP ERROR INFORMATION  
                ;WRITE TO 1 INTERNAL ADDRESS MODIFIED  
                ;ANOTHER SO ADDRESS NOT UNIQUE  
                ;RESTORE POINTER  
                ;GET BACK IN TEST LOOP  
        MOV      (R0),$GDADR  
        MOV      (R1),$BDADR  
        MOV      @ (R0),$GDDAT  
        MOV      @ (R1),$BDDAT  
        ERROR   72  
  
6$:     ADD      #2,R1  
7$:     BR       4$
```



```

2367 006700
2368
2369
2370
2371
2372 006700 000004
2373 006702 012737 000033 001104
2374 006710 032737 000001 001120
2375 006716 001405
2376 006720 005737 001106
2377 006724 001402
2378 006726 000137 010600
2379 006732
2380 006732 032737 000001 001156
2381 006740 001002
2382 006742 000137 010600
2383 006746
2384 006746 000005
2385 006750 042777 000100 174026
2386 006756 017703 174044
2387 006762 012777 007006 174036
2388 006770 105777 174010
2389 006774 100375
2390 006776 004737 015056
2391 007002 000140
2392 007004 000402
2393
2394 007006 022626
2395 007010 104074
2396
2397 007012 012777 007032 174006
2398 007020 052777 000100 173756
2399 007026 000240
2400
2401 007030 104075
2402
2403 007032 042777 000100 173744
2404 007040 022626
2405 007042 010377 173760
2406
2407
2408
2409
2410
2411 007046 000004
2412 007050 012737 000034 001104
2413 007056 042777 000100 173720
2414 007064 004737 015056
2415 007070 000340
2416 007072 017703 173730
2417 007076 012777 007124 173722
2418 007104 105777 173674
2419 007110 100375
2420 007112 052777 000100 173664
2421 007120 000240
2422 007122 000402
  
```

SLUIT:

```

*****
:*TEST 33 TEST THAT SLU1 XMIT INTERRUPTS ONLY WHEN ENABLED
*****
TST33: SCOPE
MOV #33,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#$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$: BIT #BIT0,@#$DEVN ;;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 34 TEST SLU1 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED
*****
TST34: SCOPE
MOV #34,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@CTCSR ;;DISABLE INTERRUPTS
JSR PC,WRPSW ;;SET PSW TO PRIORITY 7
WORD 340
MOV @CTVECT,R3 ;;SAVE XMIT VECTOR
MOV #2$,@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
  
```



```
2423
2424 007124 022626      2$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
2425 007126 104076      ERROR      76
2426
2427 007130 042777 000100 173646 3$:  BIC      #BIT6,@CTCSR      ;XMIT INTERRUPTS AT PRIORITY=7
2428 007136 012777 007156 173662      MOV      #4$,@CTVECT      ;CLEAR INTERRUPT ENABLE
2429 007144 004737 015056      JSR      PC,WRPSW          ;POINT XMIT VECTOR TO ERROR REPORT
2430 007150 000140      .WORD    140              ;SET PSW TO PRIORITY 3
2431 007152 000240      NOP
2432 007154 000402      BR       5$              ;BR TO END OF TEST-NO INTERRUPT
2433
2434 007156 022626      4$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
2435 007160 104077      ERROR      77
2436
2437 007162 010377 173640 5$:  MOV      R3,@CTVECT      ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2438
2439
2440
2441
2442
2443 007166 000004      TST35: SCOPE
2444 007170 012737 000035 001104      MOV      #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2445 007176 042777 000100 173600      BIC      #BIT6,@CTCSR      ;CLEAR INTERRUPT ENABLE
2446 007204 017703 173616      MOV      @CTVECT,R3      ;SAVE XMIT VECTOR
2447 007210 017704 173614      MOV      @CTPSW,R4      ;SAVE XMIT PSW VECTOR
2448 007214 012777 007256 173604      MOV      #2$,@CTVECT      ;SET UP XMIT VECTOR
2449 007222 012777 000340 173600      MOV      #340,@CTPSW      ;SET PIO 7 AFTER INTERRUPT
2450 007230 004737 015056      JSR      PC,WRPSW          ;SET PSW TO PRIORITY 3
2451 007234 000140      .WORD    140
2452 007236 105777 173542      1$:  TSTB     @CTCSR          ;WAIT FOR DONE
2453 007242 100375      BPL      1$
2454 007244 052777 000100 173532      BIS      #BIT6,@CTCSR      ;ENABLE INTERRUPTS
2455 007252 000240      NOP
2456
2457 007254 104100      ERROR      100
2458
2459 007256 022626      2$:  CMP      (SP)+,(SP)+      ;XMIT INTERRUPT DID NOT OCCUR
2460 007260 012777 007306 173540      MOV      #4$,@CTVECT      ;RESTORE SP AFTER INTERRUPT
2461 007266 004737 015056      JSR      PC,WRPSW          ;POINT XMIT VECTOR TO ERROR
2462 007272 000140      .WORD    140              ;SET PSW TO PRIORITY 3
2463 007274 000240      NOP
2464 007276 042777 000100 173500      BIC      #BIT6,@CTCSR      ;GIVE TIME FOR ANY INTERRUPTS
2465 007304 000402      BR       5$              ;DISABLE INTERRUPTS
2466
2467 007306 022626      4$:  CMP      (SP)+,(SP)+      ;RESTORE SP AFTER INTERRUPT
2468 007310 104101      ERROR      101
2469
2470 007312 010377 173510 5$:  MOV      R3,@CTVECT      ;XMIT RE-INTERRUPTED
2471 007316 010477 173506      MOV      R4,@CTPSW        ;RESTORE XMIT VECTOR
2472
2473
2474
2475
2476 007322 000004      TST36: SCOPE
2477 007324 012737 000036 001104      MOV      #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2478 007332 042777 000100 173444      BIC      #BIT6,@CTCSR      ;DISABLE INTERRUPTS
```



```
2479 007340 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY 7
2480 007344 000340 .WORD 340
2481 007346 017703 173454 MOV @CTVECT,R3 ;SAVE XMIT VECTOR
2482 007352 012777 007424 173446 MOV #2$,@CTVECT ;POINT XMIT VECTOR TO ERROR
2483 007360 052777 000100 173416 BIS #BIT6,@CTCSR ;ENABLE INTERRUPTS
2484 007366 005077 173414 CLR @CTBUF ;LOAD TBUF
2485 007372 105777 173406 1$: TSTB @CTCSR ;WAIT FOR DONE (INTERRUPT)
2486 007376 100375 BPL 1$
2487 007400 005077 173402 CLR @CTBUF ;FILL SECOND BUFFER TO RESET INT.
2488 007404 004737 015056 JSR PC,WRPSW ;ALLOW INTERRUPTS
2489 007410 000140 .WORD 140
2490 007412 000240 NOP ;GIVE TIME FOR ANY INTERRUPTS
2491 007414 042777 000100 173362 BIC #BIT6,@CTCSR ;DISABLE INTERRUPTS
2492 007422 000402 BR 3$ ;BR TO END OF TEST
2493
2494 007424 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2495 007426 104102 ERROR 102
2496
2497 007430 010377 173372 3$: MOV R3,@CTVECT ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2498 ;RESTORE XMIT VECTOR
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509 007434 000004
2510 007436 012737 000037 001104
2511 007444 000005
2512
2513 007446 042777 000100 173330 BIC #BIT6,@CTCSR ;DISABLE TRANSMIT INTERRUPTS
2514 007454 042777 000100 173316 BIC #BIT6,@CRCSR ;DISABLE RECEIVER INTERRUPTS
2515 007462 052777 000004 173314 BIS #BIT2,@CTCSR ;SET MAINTENANCE WRAP
2516 007470 017703 173326 MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
2517 007474 012777 007532 173320 MOV #2$,@CRVECT ;POINT RCV VECTOR TO ERROR REPORT
2518 007502 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
2519 007506 000140 .WORD 140
2520 007510 005077 173272 CLR @CTBUF ;SEND A CHARACTER
2521 007514 105777 173260 1$: TSTB @CRCSR ;WAIT FOR RECEIVER DONE
2522 007520 100375 BPL 1$
2523 007522 042777 000004 173254 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
2524 007530 000405 BR 3$ ;CONTINUE TEST
2525
2526 007532 042777 000004 173244 2$: BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
2527 007540 022626 CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2528 007542 104103 ERROR 103
2529 ;RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR
2530
2531 007544 012777 007572 173250 3$: MOV #4$,@CRVECT ;POINT RCV VECTOR TO END OF TEST
2532 007552 052777 000100 173220 BIS #BIT6,@CRCSR ;ENABLE RCV INTERRUPTS
2533 007560 000240 NOP ;GIVE ANY INTERRUPTS TIME
2534 007562 042777 000004 173214 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
```

```
::*****
:*TEST 37 TEST THAT SLU1 RCVR INTERRUPTS ONLY WHEN ENABLED
:*****
```

```
TST37: SCOPE
MOV #37,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
RESET ;MAKE SURE NO INTERRUPTS ARE PENDING FROM
;ANOTHER TEST
BIC #BIT6,@CTCSR ;DISABLE TRANSMIT INTERRUPTS
BIC #BIT6,@CRCSR ;DISABLE RECEIVER INTERRUPTS
BIS #BIT2,@CTCSR ;SET MAINTENANCE WRAP
MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
MOV #2$,@CRVECT ;POINT RCV VECTOR TO ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY 3
.WORD 140
CLR @CTBUF ;SEND A CHARACTER
1$: TSTB @CRCSR ;WAIT FOR RECEIVER DONE
BPL 1$
BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
BR 3$ ;CONTINUE TEST
2$: BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 103 ;RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR
3$: MOV #4$,@CRVECT ;POINT RCV VECTOR TO END OF TEST
BIS #BIT6,@CRCSR ;ENABLE RCV INTERRUPTS
NOP ;GIVE ANY INTERRUPTS TIME
BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
```



```
2535 007570 104104          ERROR 104          ;RCVR DID NOT INTERRUPT
2536
2537
2538 007572 042777 000100 173200 4$:  BIC  #BIT6,@CRCSR  ;DISABLE INTERRUPTS
2539 007600 042777 000004 173176  BIC  #BIT2,@CTCSR  ;CLEAR MAINTENANCE BIT
2540 007606 022626          CMP  (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
2541 007610 010377 173206          MOV  R3,@CRVECT   ;RESTORE RECEIVE VECTOR
2542
2543
2544
2545      ;*****
2546      ;*TEST 40      TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
2547      ;*****
2547 007614 000004          TST40: SCOPE
2548 007616 012737 000040 001104  MOV  #40,$TESTN  ;;SET TEST NUMBER IN APT MAIL BOX
2549 007624 000005          RESET          ;CLEAR EVERYTHING
2550 007626 004737 015056          JSR  PC,WRPSW    ;SET PSW TO PRIORITY 7
2551 007632 000340          .WORD 340
2552 007634 017703 173162          MOV  @CRVECT,R3  ;SAVE RECEIVE VECTOR
2553 007640 012777 007700 173154  MOV  #2,@CRVECT  ;POINT RCVR VECTOR TO ERROR REPORT
2554 007646 052777 000004 173130  BIS  #BIT2,@CTCSR ;SET MAINTENANCE WRAP
2555 007654 005077 173126          CLR  @CTBUF      ;SEND A CHARACTER
2556 007660 105777 173114          1$:  TSTB @CRCSR     ;WAIT FOR RECEIVER DONE
2557 007664 100375          BPL  1$
2558 007666 052777 000100 173104  BIS  #BIT6,@CRCSR ;ENABLE INTERRUPTS
2559 007674 000240          NOP            ;GIVE TIME FOR INTERRUPT
2560 007676 000405          BR   3$        ;CONTINUE TEST
2561 007700 042777 000004 173076 2$:  BIC  #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
2562 007706 022626          CMP  (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2563 007710 104105          ERROR 105
2564
2565
2566 007712 042777 000100 173060 3$:  BIC  #BIT6,@CRCSR ;CLEAR INTERRUPT ENABLE
2567 007720 012777 007746 173074  MOV  #4,@CRVECT  ;POINT RCVR VECTOR TO ERROR REPORT
2568 007726 004737 015056          JSR  PC,WRPSW    ;SET PSW TO PRIORITY 3
2569 007732 000140          .WORD 140
2570 007734 000240          NOP            ;GIVE TIME FOR ANY INTERRUPT
2571 007736 042777 000004 173040  BIC  #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
2572 007744 000405          BR   5$        ;BR TO END OF TEST, IF NO INTERRUPT
2573
2574 007746 042777 000004 173030 4$:  BIC  #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
2575 007754 022626          CMP  (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2576 007756 104106          ERROR 106
2577
2578 007760 010377 173036          5$:  MOV  R3,@CRVECT  ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
2579
2580
2581
2582      ;*****
2583      ;*TEST 41      TEST SLU1 RECEIVER FOR DOUBLE INTERRUPTS
2584      ;*****
2584 007764 000004          TST41: SCOPE
2585 007766 012737 000041 001104  MOV  #41,$TESTN  ;;SET TEST NUMBER IN APT MAIL BOX
2586 007774 000005          RESET          ;CLEAR EVERYTHING
2587 007776 017703 173020          MOV  @CRVECT,R3  ;SAVE RECEIVE VECTOR
2588 010002 017704 173016          MOV  @CRPSW,R4   ;SAVE RECEIVE PSW VECTOR
2589 010006 012777 010070 173006  MOV  #2,@CRVECT  ;POINT RCVR VECTOR TO CONTINUE TEST
2590 010014 012777 000340 173002  MOV  #340,@CRPSW ;SET PRIORITY TO 7 AFTER INTERRUPT
```



```

2591 010022 004737 015056      JSR    PC,WRPSW      ;SET PSW TO PRIORITY 3
2592 010026 000140                .WORD    140
2593 010030 052777 000004 172746  BIS    #BIT2,@CTCSR  ;SET MAINTENANCE WRAP
2594 010036 005077 172744                CLR    @CTBUF        ;SEND A CHARACTER
2595 010042 105777 172732      1$:   TSTB   @CRCSR      ;WAIT FOR RCVR DONE
2596 010046 100375                BPL    1$
2597 010050 042777 000004 172726  BIC    #BIT2,@CTCSR  ;CLEAR MAINTENANCE BIT
2598 010056 052777 000100 172714  BIS    #BIT6,@CRCSR  ;ENABLE RCV INTERRUPTS
2599 010064 000240                NOP
2600
2601 010066 104107                ERROR   107
2602
2603
2604 010070 022626                2$:   CMP    (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
2605 010072 012777 010124 172722  MOV    #3$,@CRVECT  ;POINT RCV VECTOR TO ERROR REPORT
2606 010100 004737 015056      JSR    PC,WRPSW      ;RESET PSW TO PRIORITY 3
2607 010104 000140                .WORD    140
2608 010106 000240                NOP
2609 010110 042777 000100 172662  BIC    #BIT6,@CRCSR  ;CLEAR INTERRUPT ENABLE
2610 010116 010477 172702      MOV    R4,@CRPSW    ;RESTORE RECEIVE PSW VECTOR
2611 010122 000402                BR     4$            ;BR TO END OF TEST
2612
2613 010124 022626                3$:   CMP    (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
2614 010126 104110                ERROR   110
2615
2616 010130 010377 172666      4$:   MOV    R3,@CRVECT ;RESTORE RECEIVE VECTOR
2617
2618
2619
2620
2621
2622 010134 000004                ::*****
2623 010136 012737 000042 001104  ;*TEST 42      TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2624 010144 000005                ;*****
2625 010146 004737 015056      TST42:  SCOPE
2626 010152 000340                MOV    #42,$TESTN   ;:SET TEST NUMBER IN APT MAIL BOX
2627 010154 017703 172642                RESET                ;:CLEAR EVERYTHING
2628 010160 012777 010246 172634  JSR    PC,WRPSW      ;:SET PSW PRIORITY TO 7
2629 010166 052777 000100 172604  .WORD    340
2630 010174 052777 000004 172602  MOV    @CRVECT,R3   ;:SAVE RECEIVE VECTOR
2631 010202 005077 172600                MOV    #2$,@CRVECT  ;:POINT RCV VECTOR TO ERROR REPORT
2632 010206 105777 172566      1$:   BIS    #BIT6,@CRCSR  ;:SET RCVR INTERRUPT ENABLE
2633 010212 100375                BIS    #BIT2,@CTCSR  ;:SET MAINTENANCE WRAP
2634 010214 042777 000004 172562  CLR    @CTBUF        ;:SEND A CHARACTER
2635 010222 005777 172554                TSTB   @CRCSR      ;:WAIT FOR DONE (INTERRUPT)
2636 010226 004737 015056      BPL    1$
2637 010232 000140                BIC    #BIT2,@CTCSR  ;:CLEAR MAINTENANCE BIT
2638 010234 000240                TST    @CRBUF        ;:READ RBUF TO CLEAR PENDING INTERRUPT
2639 010236 042777 000100 172534  JSR    PC,WRPSW      ;:SET PSW TO PRIORITY 3
2640 010244 000402                .WORD    140
2641
2642 010246 022626                2$:   NOP
2643 010250 104111                BIC    #BIT6,@CRCSR  ;:ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2644
2645 010252 010377 172544      3$:   BR     3$            ;:NO INTERRUPT-CLEAR INT. ENABLE
2646

```



```

2647
2648
2649
2650
2651
2652 010256 000004
2653 010260 012737 000043 001104
2654 010266 000005
2655 010270 004737 015056
2656 010274 000340
2657 010276 017703 172520
2658 010302 012777 010362 172512
2659 010310 052777 000100 172462
2660 010316 052777 000004 172460
2661 010324 012777 000377 172454
2662 010332 105777 172442 1$:
2663 010336 100375
2664 010340 000005
2665 010342 004737 015056
2666 010346 000140
2667 010350 000240
2668 010352 042777 000100 172420
2669 010360 000402
2670
2671
2672 010362 022626 2$:
2673 010364 104112 ERROR 112
2674
2675 010366 010377 172430 3$:
2676
2677
2678
2679
2680
2681 010372 000004
2682 010374 012737 000044 001104
2683 010402 032777 002000 170430
2684 010410 001032
2685 010412 000005
2686 010414 052777 000004 172362
2687 010422 012700 000003
2688 010426 005077 172354 1$:
2689 010432 105777 172346 2$:
2690 010436 100375
2691 010440 005300
2692 010442 001371
2693 010444 042777 000004 172332
2694 010452 032777 040000 172322
2695 010460 001001
2696 010462 104113
2697
2698
2699 010464 032777 100000 172310 3$:
2700 010472 001001
2701 010474 104114
2702

```

```

*****
*TEST 43 TEST SLU1 THAT RESET CLEARS RECEIVE INTERRUPT
*****
TST43: SCOPE
MOV #43,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
RESET ;;CLEAR EVERYTHING
JSR PC,WRPSW ;;SET PSW TO PRIORITY 7
.WORD 340
MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
MOV #2$,@CRVECT ;POINT RCV VECTOR TO ERROR REPORT
BIS #BIT6,@CRCSR ;SET RCV INTERRUPT ENABLE
BIS #BIT2,@CTCSR ;SET MAINTENANCE WRAP
MOV #377,@CTBUF ;SEND AN ALL 1'S CHARACTER
1$: TSTB @CRCSR ;WAIT FOR RCV DONE
BPL 1$
RESET ;CLEAR RCV INTERRUPT & RBUF
JSR PC,WRPSW ;SET PSW TO PRIORITY 3
.WORD 140
NOP ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
BIC #BIT6,@CRCSR ;NO INTERRUPT-CLEAR INT. ENABLE
BR 3$ ;CONTINUE TEST

2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 112 ;RESET DID NOT CLEAR RCVR INTERRUPT

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

*****
*TEST 44 TEST SLU1 THAT 'OVERRUN & ERROR' BITS CAN BE SET
*****
TST44: SCOPE
MOV #44,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #BIT10,@SWR ;IS THIS TEST ENABLED
BNE TST45 ;IF DISABLED, BR TO NEXT TEST
RESET ;CLEAR EVERYTHING
BIS #BIT2,@CTCSR ;SET MAINTENANCE WRAP
MOV #3,R0 ;SET CHARACTER COUNT TO SEND 3 CHAR.
1$: CLR @CTBUF ;LOAD TRANSMIT BUFFER
2$: TSTB @CTCSR ;WAIT FOR TRANSMIT DONE
BPL 2$
DEC R0 ;DECREMENT CHARACTER COUNT
BNE 1$ ;BR IF ALL CHARACTERS NOT TRANSMITTED
BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
BIT #BIT14,@CRBUF ;TEST FOR 'OR' ERROR FLAG
BNE 3$ ;BR, IF SET
ERROR 113 ;'OR' ERROR FLAG DID NOT SET

3$: BIT #BIT15,@CRBUF ;TEST 'ERROR' FLAG
BNE 4$ ;BR, IF SET
ERROR 114 ;'ERROR' FLAG DID NOT SET WITH 'OR' FLAG

```


2703 010476
2704
2705
2706
2707
2708
2709 010476 000004
2710 010500 012737 000045 001104
2711 010506 000005
2712 010510 005001
2713 010512 052777 000004 172264
2714 010520 105201
2715 010522 010177 172260
2716 010526 105777 172246
2717 010532 100375
2718 010534 017702 172242
2719 010540 020102
2720 010542 001003
2721 010544 105701
2722 010546 001411
2723 010550 000763
2724 010552 010137 001024
2725 010556 010237 001026
2726 010562 042777 000004 172214
2727 010570 104115
2728
2729 010572 042777 000004 172204
2730
2731

4\$:

```
*****  
: *TEST 45 TEST SLU1 DATA PATH USING MAINTENANCE WRAP  
*****  
TST45: SCOPE  
MOV #45,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX  
RESET ;:CLEAR EVERYTHING  
CLR R1 ;:CLEAR REGISTER FOR TEST DATA  
BIS #BIT2,@CTCSR ;:SET MAINTENANCE WRAP  
1$: INCB R1 ;:INCREMENT THE TEST DATA  
MOV R1,@CTBUF ;:XMIT A CHARACTER  
2$: TSTB @CRCSR ;:WAIT FOR RECEIVER DONE  
BPL 2$  
MOV @CRBUF,R2 ;:GET RECEIVED CHARACTER  
CMP R1,R2 ;:COMPARE DATA  
BNE 3$ ;:BR, IF NON-COMPARE  
TSTB R1 ;:TEST XMIT DATA FOR ZERO  
BEQ 4$ ;:BR, IF FINISHED  
BR 1$ ;:CONTINUE IF NOT  
3$: MOV R1,$GDDAT ;:STORE THE EXPECTED DATA  
MOV R2,$BDDAT ;:STORE RECEIVED DATA  
BIC #BIT2,@CTCSR ;:CLEAR MAINTENANCE BIT  
ERROR 115 ;:DATA COMPARE DATA  
4$: BIC #BIT2,@CTCSR ;:CLEAR MAINTENANCE BIT
```


2732 010600

SLU2IT:

2733

2734

2735

2736

2737

2738

2739

2740

2741

2742

2743

2744

2745

2746

2747

2748

2749

2750

2751

2752

2753

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

010600 000004 000046 001104
010602 012737 000046 001156
010610 032737 000002 001156
010616 001002
010620 000137 012544
010624
010624 042777 000100 172142
010632 017703 172160
010636 012777 010662 172152
010644 105777 172124
010650 100375
010652 004737 015056
010656 000140
010660 000402
010662 022626
010664 104116
010666 012777 010706 172122
010674 052777 000100 172072
010702 000240
010704 104117
010706 042777 000100 172060
010714 022626
010716 010377 172074
010722 000004
010724 012737 000047 001104
010732 042777 000100 172034
010740 004737 015056
010744 000340
010746 017703 172044
010752 012777 011000 172036
010760 105777 172010
010764 100375
010766 052777 000100 172000
010774 000240
010776 000402
011000 022626
011002 104120
011004 042777 000100 171762
011012 012777 011032 171776
011020 004737 015056

*TEST 46 TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED

TST46: SCOPE
MOV #46,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIT #BIT1,@#\$DEVM ;:DO THESE TESTS FOR THIS DEVICE?
BNE 99\$;:F YES CONTINUE WITH TESTS
JMP LTCIT ;:IF NO GO TO START OF NEXT SET OF TESTS.
99\$:
BIC #BIT6,@TCSR ;:CLEAR TRANSMIT INTERRUPT ENABLE
MOV @TVECT,R3 ;:SAVE XMIT VECTOR
MOV #2\$,@TVECT ;:POINT XMIT VECTOR TO ERROR REPORT
1\$: TSTB @TCSR ;: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 116
3\$: MOV #4\$,@TVECT ;:XMIT INTERRUPTS WITH INTERRUPT ENABLE CLEAR
BIS #BIT6,@TCSR ;:SET XMIT VECTOR TO END OF TEST
NOP ;:ENABLE INTERRUPTS
ERROR 117 ;:XMIT DID NOT INTERRUPT
4\$: BIC #BIT6,@TCSR ;:DISABLE INTERRUPTS
CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
MOV R3,@TVECT ;:RESTORE XMIT VECTOR

*TEST 47 TEST SLU2 XMIT INTERRUPTS DO NOT OCCUR WHEN DISABLED

TST47: SCOPE
MOV #47,\$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,@TCSR ;:DISABLE INTERRUPTS
JSR PC,WRPSW ;:SET PSW TO PRIORITY 7
.WORD 340
MOV @TVECT,R3 ;:SAVE XMIT VECTOR
MOV #2\$,@TVECT ;:POINT XMIT VECTOR TO ERROR REPORT
1\$: TSTB @TCSR ;:WAIT FOR DONE
BPL 1\$
BIS #BIT6,@TCSR ;:ENABLE INTERRUPT
NOP
BR 3\$;:CONTINUE TEST
2\$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
ERROR 120
3\$: BIC #BIT6,@TCSR ;:XMIT INTERRUPTS AT PRIORITY=7
MOV #4\$,@TVECT ;:CLEAR INTERRUPT ENABLE
JSR PC,WRPSW ;:POINT XMIT VECTOR TO ERROR REPORT
;:SET PSW TO PRIORITY 3


```
2788 011024 000140 .WORD 140
2789 011026 000240 NOP
2790 011030 000402 BR 5$ ;BR TO END OF TEST-NO INTERRUPT
2791
2792 011032 022626 4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2793 011034 104121 ERROR 121
2794
2795 011036 010377 171754 5$: MOV R3,@TVECT ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2796 ;RESTORE XMIT VECTOR
2797
2798
2799
```

```
2800
2801
2802
2803 011042 000004
2804 011044 012737 000050 001104
2805 011052 042777 000100 171714
2806 011060 017703 171732
2807 011064 017704 171730
2808 011070 012777 011132 171720
2809 011076 012777 000340 171714
2810 011104 004737 015056
2811 011110 000140
2812 011112 105777 171656 1$: TSTB @TCSR ;WAIT FOR DONE
2813 011116 100375 BPL 1$
2814 011120 052777 000100 171646 BIS #BIT6,@TCSR ;ENABLE INTERRUPTS
2815 011126 000240 NOP
2816
2817 011130 104122 ERROR 122 ;XMIT INTERRUPT DID NOT OCCUR
2818 ;RESTORE SP AFTER INTERRUPT
2819 011132 022626 2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2820 011134 012777 011162 171654 MOV #4$,@TVECT ;POINT XMIT VECTOR TO ERROR
2821 011142 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
2822 011146 000140 .WORD 140
2823 011150 000240 NOP ;GIVE TIME FOR ANY INTERRUPTS
2824 011152 042777 000100 171614 BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
2825 011160 000402 BR 5$ ;BR TO END OF TEST
2826
2827 011162 022626 4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2828 011164 104123 ERROR 123
2829 ;XMIT RE-INTERRUPTED
2830 011166 010377 171624 5$: MOV R3,@TVECT ;RESTORE XMIT VECTOR
2831 011172 010477 171622 MOV R4,@TPSW ;RESTORE XMIT PSW VECTOR
2832
2833
```

```
2834
2835
2836 011176 000004
2837 011200 012737 000051 001104
2838 011206 032737 000001 001120
2839 011214 001403
2840 011216 005737 001106
2841 011222 001046
2842 011224
2843 011224 042777 000100 171542 70$: BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
```



```
2844 011232 004737 015056      JSR      PC,WRPSW      ;SET PSW TO PRIORITY 7
2845 011236 000340              .WORD      340
2846 011240 017703 171552      MOV      @TVECT,R3    ;SAVE XMIT VECTOR
2847 011244 012777 011316 171544  MOV      #2$,@TVECT   ;POINT XMIT VECTOR TO ERROR
2848 011252 052777 000100 171514  BIS      #BIT6,@TCSR  ;ENABLE INTERRUPTS
2849 011260 005077 171512      CLR      @TBUF        ;LOAD TBUF
2850 011264 105777 171504      1$:     TSTB      @TCSR   ;WAIT FOR DONE (INTERRUPT)
2851 011270 100375              BPL      1$
2852 011272 005077 171500      CLR      @TBUF        ;FILL SECOND BUFFER TO RESET INT.
2853 011276 004737 015056      JSR      PC,WRPSW     ;ALLOW INTERRUPTS
2854 011302 000140              .WORD      140
2855 011304 000240              NOP
2856 011306 042777 000100 171460  BIC      #BIT6,@TCSR  ;GIVE TIME FOR ANY INTERRUPTS
2857 011314 000402              BR       3$           ;DISABLE INTERRUPTS
2858                                     ;BR TO END OF TEST
2859 011316 022626      2$:     CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2860 011320 104124      ERROR   124
2861                                     ;LOADING TBUF DID NOT CLEAR INTERRUPT.
2862 011322 005001      3$:     CLR      R1      ;INITIALIZE LOOP COUNTER
2863 011324 005201      4$:     INC      R1      ;INCREMENT LOOP COUNTER
2864 011326 001376      BNE     4$           ;UNTIL LOOP COUNTER EQUALS 0
2865 011330 005777 171436      TST     @RBUF        ;CLEAR RECEIVER BUFFER
2866 011334 010377 171456      MOV     R3,@TVECT   ;RESTORE XMIT VECTOR
2867
2868
2869
2870
2871
2872
2873
2874 011340 000004      TST52:  SCOPE
2875 011342 012737 000052 001104  MOV     #52,$TESTN   ;:SET TEST NUMBER IN APT MAIL BOX
2876 011350 042777 000100 171416  BIC     #BIT6,@TCSR  ;:DISABLE TRANSMIT INTERRUPTS
2877 011356 042777 000100 171404  BIC     #BIT6,@RCSR  ;:DISABLE RECEIVER INTERRUPTS
2878 011364 017703 171422      MOV     @RVECT,R3   ;:SAVE RECEIVE VECTOR
2879 011370 012777 011420 171414  MOV     #2$,@RVECT  ;:POINT RCV VECTOR TO ERROR REPORT
2880 011376 004737 015056      JSR     PC,WRPSW     ;:SET PSW TO PRIORITY 3
2881 011402 000140              .WORD     140
2882 011404 005077 171366      CLR     @TBUF        ;:SEND A CHARACTER
2883 011410 105777 171354      1$:     TSTB     @RCSR   ;:WAIT FOR RECEIVER DONE
2884 011414 100375              BPL     1$
2885 011416 000402              BR      3$           ;:CONTINUE TEST
2886
2887 011420 022626      2$:     CMP     (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
2888 011422 104125      ERROR   125
2889                                     ;:RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR
2890
2891 011424 012777 011444 171360  3$:     MOV     #4$,@RVECT ;:POINT RCV VECTOR TO END OF TEST
2892 011432 052777 000100 171330  BIS     #BIT6,@RCSR  ;:ENABLE RCV INTERRUPTS
2893 011440 000240              NOP
2894 011442 104126      ERROR   126
2895                                     ;:RCVR DID NOT INTERRUPT
2896
2897 011444 042777 000100 171316  4$:     BIC     #BIT6,@RCSR  ;:DISABLE INTERRUPTS
2898 011452 022626      CMP     (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT
2899 011454 010377 171332      MOV     R3,@RVECT   ;:RESTORE RECEIVE VECTOR
```



```
2900
2901
2902
2903
2904
2905 011460 000004
2906 011462 012737 000053 001104
2907 011470 004737 015056
2908 011474 000340
2909 011476 017703 171310
2910 011502 012777 011534 171302
2911 011510 005077 171262
2912 011514 105777 171250 1$:
2913 011520 100375
2914 011522 052777 000100 171240
2915 011530 000240
2916 011532 000402
2917 011534 022626 2$:
2918 011536 104127
2919
2920
2921 011540 042777 000100 171222 3$:
2922 011546 012777 011566 171236
2923 011554 004737 015056
2924 011560 000140
2925 011562 000240
2926 011564 000402
2927
2928 011566 022626 4$:
2929 011570 104130
2930
2931 011572 010377 171214 5$:
2932
2933
2934
2935
2936
2937 011576 000004
2938 011600 012737 000054 001104
2939 011606 017703 171200
2940 011612 017704 171176
2941 011616 012777 011664 171166
2942 011624 012777 000340 171162
2943 011632 004737 015056
2944 011636 000140
2945 011640 005077 171132
2946 011644 105777 171120 1$:
2947 011650 100375
2948 011652 052777 000100 171110
2949 011660 000240
2950
2951 011662 104131
2952
2953
2954 011664 022626 2$:
2955 011666 012777 011720 171116
```

```
*****
*TEST 53 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
*****
TST53: SCOPE
MOV #53,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
JSR PC,WRPSW ;;SET PSW TO PRIORITY 7
        .WORD 340
MOV @RVECT,R3 ;SAVE RECEIVE VECTOR
MOV #2$,@RVECT ;POINT RCVR VECTOR TO ERROR REPORT
CLR @TBUF ;SEND A CHARACTER
1$: TSTB @RCSR ;WAIT FOR RECEIVER DONE
    BPL 1$
    BIS #BIT6,@RCSR ;ENABLE INTERRUPTS
    NOP ;GIVE TIME FOR INTERRUPT
    BR 3$ ;CONTINUE TEST
2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
    ERROR 127 ;RCVR INTERRUPTS AT PRIORITY 7.
3$: BIC #BIT6,@RCSR ;CLEAR INTERRUPT ENABLE
    MOV #4$,@RVECT ;POINT RCVR VECTOR TO ERROR REPORT
    JSR PC,WRPSW ;SET PSW TO PRIORITY 3
        .WORD 140
    NOP ;GIVE TIME FOR ANY INTERRUPT
    BR 5$ ;BR TO END OF TEST, IF NO INTERRUPT
4$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
    ERROR 130 ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
5$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
*****
*TEST 54 TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
*****
TST54: SCOPE
MOV #54,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
MOV @RVECT,R3 ;SAVE RECEIVE VECTOR
MOV @RPSW,R4 ;SAVE RECEIVE PSW VECTOR
MOV #2$,@RVECT ;POINT RCVR VECTOR TO CONTINUE TEST
MOV #340,@RPSW ;SET PRIORITY TO 7 AFTER INTERRUPT
JSR PC,WRPSW ;SET PSW TO PRIORITY 3
        .WORD 140
CLR @TBUF ;SEND A CHARACTER
1$: TSTB @RCSR ;WAIT FOR RCVR DONE
    BPL 1$
    BIS #BIT6,@RCSR ;ENABLE RCV INTERRUPTS
    NOP ;GIVE SOME TIME
    ERROR 131 ;RCVR INTERRUPT DID NOT OCCUR
2$: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
    MOV #3$,@RVECT ;POINT RCV VECTOR TO ERROR REPORT
```



```

2956 011674 004737 015056      JSR      PC,WRPSW      ;RESET PSW TO PRIORITY 3
2957 011700 000140              .WORD      140
2958 011702 000240              NOP
2959 011704 042777 000100 171056  BIC      #BIT6,@RCSR   ;GIVE SOME TIME
2960 011712 010477 171076      MOV      R4,@RPSW     ;CLEAR INTERRUPT ENABLE
2961 011716 000402              BR        4$          ;RESTORE RECEIVE PSW VECTOR
2962                                     ;BR TO END OF TEST
2963 011720 022626      3$:      CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2964 011722 104132              ERROR      132
2965                                     ;RECEIVER RE-INTERRUPTED
2966 011724 010377 171062      4$:      MOV      R3,@RVECT ;RESTORE RECEIVE VECTOR
2967
2968
2969
2970
2971
2972 011730 000004              ::*****
2973 011732 012737 000055 001104  :*TEST 55      TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2974 011740 004737 015056      :*****
2975 011744 000340      TST55:  SCOPE
2976 011746 017703 171040      MOV      #55,$TESTN   ;;SET TEST NUMBER IN APT MAIL BOX
2977 011752 012777 012024 171032  JSR      PC,WRPSW     ;SET PSW PRIORITY TO 7
2978 011760 052777 000100 171002  .WORD      340
2979 011766 005077 171004      MOV      @RVECT,R3   ;SAVE RECEIVE VECTOR
2980 011772 105777 170772      MOV      #2$,@RVECT  ;POINT RCV VECTOR TO ERROR REPORT
2981 011776 100375              BIS      #BIT6,@RCSR  ;SET RCVR INTERRUPT ENABLE
2982 012000 005777 170766      CLR      @TBUF       ;SEND A CHARACTER
2983 012004 004737 015056      1$:      TSTB     @RCSR     ;WAIT FOR DONE (INTERRUPT)
2984 012010 000140              BPL      1$
2985 012012 000240              TST     @RBUF       ;READ RBUF TO CLEAR PENDING INTERRUPT
2986 012014 042777 000100 170746  JSR      PC,WRPSW     ;SET PSW TO PRIORITY 3
2987 012022 000402              .WORD      140
2988                                     ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2989 012024 022626      2$:      CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2990 012026 104133              ERROR      133      ;READING RBUF DID NOT CLEAR INTERRUPT
2991
2992 012030 010377 170756      3$:      MOV      R3,@RVECT ;RESTORE RECEIVE VECTOR
2993
2994
2995
2996
2997
2998
2999 012034 000004              ::*****
3000 012036 012737 000056 001104  :*TEST 56      TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
3001 012044 032737 000001 001120  :*****
3002 012052 001403      TST56:  SCOPE
3003 012054 005737 001106      MOV      #56,$TESTN   ;;SET TEST NUMBER IN APT MAIL BOX
3004 012060 001037              BIT      #1,@#$ENV    ;ARE WE RUNNING UNDER APT
3005 012062              BEQ      70$          ;IF NO THEN DO TEST
3006 012062 000005              TST     @#$PASS      ;IS THIS FIRST PASS
3007 012064 004737 015056      BNE     TST57        ;IF NO THEN SKIP TO NEXT TEST
3008 012070 000340      70$:      RESET
3009 012072 017703 170714      JSR      PC,WRPSW     ;CLEAR EVERYTHING
3010 012076 012777 012150 170706  .WORD      340      ;SET PSW TO PRIORITY 7
3011 012104 052777 000100 170656  MOV      @RVECT,R3   ;SAVE RECEIVE VECTOR
              MOV      #2$,@RVECT  ;POINT RCV VECTOR TO ERROR REPORT
              BIS      #BIT6,@RCSR  ;SET RCV INTERRUPT ENABLE

```



```
3012 012112 012777 000377 170656      MOV    #377,@TBUF    ;SEND AN ALL 1'S CHARACTER
3013 012120 105777 170644      1$:   TSTB   @RCSR      ;WAIT FOR RCV DONE
3014 012124 100375                BPL    1$
3015 012126 000005                RESET   ;CLEAR RCV INTERRUPT & RBUF
3016 012130 004737 015056      JSR    PC,WRPSW     ;SET PSW TO PRIORITY 3
3017 012134 000140                .WORD  140
3018 012136 000240                NOP
3019 012140 042777 000100 170622      BIC   #BIT6,@RCSR   ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
3020 012146 000402                BR     3$           ;NO INTERRUPT-CLEAR INT. ENABLE
                                ;CONTINUE TEST
```

```
3021
3022
3023 012150 022626      2$:   CMP    (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
3024 012152 104134      ERROR  134         ;RESET DID NOT CLEAR RCVR INTERRUPT
3025
3026 012154 010377 170632      3$:   MOV    R3,@RVECT  ;RESTORE RECEIVE VECTOR
3027
3028
```

```
::*****
:*TEST 57 TEST SLU2 THAT 'OVERRUN & ERROR' BITS CAN BE SET
::*****
```

```
3029
3030
3031 TST57: SCOPE
3032 012160 000004      MOV    #57,$TESTN  ;;SET TEST NUMBER IN APT MAIL BOX
3033 012162 012737 000057 001104      BIT    #BIT10,@SWR ;IS THIS TEST DISABLED
3034 012170 032777 002000 166642      BNE   TST60        ;IF NOT ENABLED, BR TO NEXT TEST
3035 012176 001023                MOV    #3,R0       ;SET CHARACTER COUNT TO SEND 3 CHAR.
3036 012200 012700 000003                CLR   @TBUF        ;LOAD TRANSMIT BUFFER
3037 012204 005077 170566      1$:   TSTB   @TCSR      ;WAIT FOR TRANSMIT DONE
3038 012210 105777 170560      2$:   BPL    2$
3039 012214 100375                DEC   R0           ;DECREMENT CHARACTER COUNT
3040 012216 005300                BNE   1$          ;BR IF ALL CHARACTERS NOT TRANSMITTED
3041 012220 001371                BIT    #BIT14,@RBUF ;TEST FOR 'OR' ERROR FLAG
3042 012222 032777 040000 170542      BNE   3$          ;BR, IF SET
3043 012230 001001                ERROR  135
3044 012232 104135                ;'OR' ERROR FLAG DID NOT SET
3045
3046
3047 012234 032777 100000 170530      3$:   BIT    #BIT15,@RBUF ;TEST 'ERROR' FLAG
3048 012242 001001                BNE   4$          ;BR, IF SET
3049 012244 104136                ERROR  136
3050                ;'ERROR' FLAG DID NOT SET WITH 'OR' FLAG
3051 012246      4$:
3052
3053
```

```
::*****
:*TEST 60 TEST THAT BREAK TRANSMITS ALL ZEROES
::*****
```

```
3054
3055
3056 TST60: SCOPE
3057 012246 000004      MOV    #60,$TESTN  ;;SET TEST NUMBER IN APT MAIL BOX
3058 012250 012737 000060 001104      MOV    #-1,@TBUF   ;TRANSMIT ALL ONES TO RCVR
3059 012256 012777 177777 170512      1$:   TSTB   @RCSR      ;WAIT FOR RCVR DONE
3060 012264 105777 170500                BPL    1$
3061 012270 100375                TST   @RBUF        ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
3062 012272 005777 170474                BIS   #BIT0,@TCSR  ;TRANSMIT BREAK
3063 012276 052777 000001 170470      CLR   R0           ;CLEAR A TIMER
3064 012304 005000                TSTB   @RCSR      ;WAIT FOR RCVR DONE
3065 012306 105777 170456      2$:   BMI   CONT41     ;BR IF DONE
3066 012312 100406                INC   R0           ;IF NOT, INCREMENT TIMER
3067 012314 005200
```



```
3068 012316 001373          BNE      2$          ;BR IF TIME REMAINS
3069
3070 012320 042777 000001 170446      BIC      #BIT0,@TCSR ;CLEAR BREAK BITS
3071 012326 104137          ERROR    137        ;BREAK DID NOT TRANSMIT ANYTHING
3072
3073 012330 105777 170436      CONT41: TSTB     @RBUF ;CHECK RECEIVE BUFFER FOR ZERO
3074 012334 001404          BEQ      3$          ;BR, IF ZERO
3075 012336 042777 000001 170430      BIC      #BIT0,@TCSR ;CLEAR BREAK BITS
3076
3077 012344 104137          ERROR    137        ;BREAK DID NOT TRANSMIT ALL ZEROES
3078
3079 012346 042777 000001 170420 3$:      BIC      #BIT0,@TCSR ;CLEAR BREAK BITS
3080
3081
3082
3083
3084
3085 012354 000004          *****
3086 012356 012737 000061 001104      :*TEST 61      TEST THAT 'FR' ERROR CAN BE SET DURING BREAK
3087 012364 032777 002000 166446      *****
3088 012372 001025          TST61: SCOPE
3089 012374 052777 000001 170372      MOV      #61,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
3090 012402 005077 170370          BIT      #BIT10,@SWR ;IS THIS TEST DISABLED
3091 012406 105777 170356      1$:      BNE      TST62      ;BR TO NEXT TEST, IF DISABLED
3092 012412 100375          BIS      #BIT0,@TCSR ;SEND BREAK
3093 012414 042777 000001 170352      CLR      @TBUF      ;TRANSMIT A CHARACTER TO TIME BREAK
3094 012422 032777 020000 170342      TSTB     @RCSR      ;WAIT FOR RCVR DONE
3095 012430 001001          BPL      1$          ;CLEAR BREAK BITS
3096
3097 012432 104140          BIC      #BIT0,@TCSR ;CHECK FOR FRAMING ERROR FLAG
3098
3099 012434 032777 100000 170330 2$:      BIT      #BIT13,@RBUF ;BR, IF SET
3100 012442 001001          BNE      2$          ;BREAK DID NOT SET FRAMING ERROR
3101
3102 012444 104141          ERROR    140        ;TEST 'ERROR' FLAG
3103
3104 012446          3$:      ERROR    141        ;BR, IF SET
3105
3106
3107
3108 012446 000004          *****
3109 012450 012737 000062 001104      :*TEST 62      TEST DATA PATHS USING WRAP CABLE
3110 012456 032777 000200 166354      *****
3111 012464 001027          TST62: SCOPE
3112 012466 005001          MOV      #62,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX
3113
3114
3115
3116 012470 105201          BIT      #BIT7,@SWR ;IS THIS TEST ENABLED
3117 012472 010177 170300          BNE      TST63      ;BR, IF NOT
3118 012476 005000          CLR      R1         ;CLEAR REGISTER FOR TEST DATA
3119 012500 105777 170264      1$:      CLR      R1         ;TRANSMIT A BINARY COUNT PATTERN - UP
3120 012504 100403          INC     R1         ;TO THE BIT POSITION INDICATED BY THE
3121 012506 005200          INC     R1         ;CONTENTS OF LOCATION '$USWR'
3122 012510 001373          INCB    R1         ;INCREMENT THE TEST DATA
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
```



```
3124 012512 104056          ERROR 56          ;RECEIVER DONE NOT SET
3125
3126 012514 017702 170252  3$:  MOV  @RBUF,R2      ;GET RECEIVED CHARACTER
3127 012520 020102          CMP  R1,R2         ;COMPARE DATA
3128 012522 001003          BNE  4$           ;BR, IF NON-COMPARE
3129 012524 105701          TSTB R1           ;TEST XMIT DATA FOR ZERO
3130 012526 001406          BEQ  TST63        ;BR, IF FINISHED
3131 012530 000757          BR   1$          ;CONTINUE IF NOT
3132 012532 010137 001024  4$:  MOV  R1,$GDDAT   ;STORE EXPECTED DATA
3133 012536 010237 001026  MOV  R2,$BDDAT   ;STORE RECEIVED DATA
3134
3135 012542 104142          ERROR 142        ;DATA COMPARE ERROR WITH WRAP CABLE
3136
3137
```


3138 012544
3139
3140
3141
3142
3143 012544 000004
3144 012546 012737 000063 001104
3145 012554 032737 000004 001156
3146 012562 001002
3147 012564 000137 013600
3148 012570
3149 012570 004737 015056
3150 012574 000340
3151 012576 017703 170230
3152 012602 017704 170226
3153 012606 012777 012650 170216
3154 012614 012777 000340 170212
3155 012622 042777 000200 170160
3156 012630 052777 000100 170152
3157 012636 105777 170146
3158 012642 100375
3159 012644 000240
3160 012646 000402
3161
3162 012650 022626
3163 012652 104143
3164
3165 012654 005077 170130
3166 012660 012777 012706 170144
3167 012666 004737 015056
3168 012672 000240
3169 012674 105777 170110
3170 012700 100375
3171 012702 000240
3172 012704 000402
3173
3174 012706 022626
3175 012710 104144
3176
3177 012712 012777 012746 170112
3178 012720 042777 000200 170062
3179 012726 052777 000100 170054
3180 012734 105777 170050
3181 012740 100375
3182 012742 000240
3183
3184 012744 104145
3185
3186 012746 022626
3187 012750 042777 000100 170032
3188 012756 010377 170050
3189 012762 010477 170046
3190
3191
3192
3193

LTCIT:

```
*****  
: *TEST 63 TEST THAT THE REAL TIME CLOCK INTERRUPTS PROPERLY  
: *****  
TST63: SCOPE  
MOV #63,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX  
BIT #BIT2,@#$DEVM ;:DO THESE TESTS FOR THIS DEVICE?  
BNE 99$ ;:F YES CONTINUE WITH TESTS  
JMP BLAST ;:IF NO GO TO START OF NEXT SET OF TESTS.  
  
99$: JSR PC,WRPSW ;:SET PSW TO PRIORITY 7  
 .WORD 340  
MOV @RTCVT,R3 ;:SAVE LINE CLOCK VECTOR  
MOV @RTCP SW,R4 ;:SAVE LINE CLOCK PSW VECTOR  
MOV #2$,@RTCVT ;:SET RTC INTERRUPT VECTOR TO ERROR REPORT  
MOV #340,@RTCP SW ;:KEEP PRIORITY AT 7  
BIC #BIT7,@LKS ;:CLEAR CLOCK DONE FLAG  
BIS #BIT6,@LKS ;:SET INTERRUPT ENABLE  
1$: TSTB @LKS ;:WAIT FOR RTC DONE(INTERRUPT REQUEST)  
 BPL 1$  
NOP ;:GIVE TIME FOR ANY INTERRUPTS  
BR 3$ ;:BR, IF NO INTERRUPT OCCURS  
  
2$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT  
 ERROR 143 ;:RTC INTERRUPTS AT PRIORITY 7  
  
3$: CLR @LKS ;:DISABLE RTC INTERRUPTS & CLEAR DONE  
MOV #4$,@RTCVT ;:SET RTC INTERRUPT VECTOR FOR ERROR  
JSR PC,WRPSW ;:CHANGE PSW TO PRIORITY 5  
 .WORD 240  
20$: TSTB @LKS ;:WAIT FOR DONE (INTERRUPT REQUEST)  
 BPL 20$  
NOP ;:GIVE TIME FOR ANY INTERRUPT  
BR 5$ ;:IF NO INTERRUPT - BR TO CONTINUE TEST  
  
4$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT  
 ERROR 144 ;:RTC INTERRUPTS WITH INTERRUPTS DISABLED  
  
5$: MOV #7$,@RTCVT ;:POINT RTC VECTOR TO END OF TEST  
BIC #BIT7,@LKS ;:CLEAR CLOCK DONE FLAG  
BIS #BIT6,@LKS ;:ALLOW INTERRUPTS  
6$: TSTB @LKS ;:WAIT FOR RTC DONE  
 BPL 6$  
NOP ;:GIVE TIME FOR INTERRUPT  
  
ERROR 145 ;:RTC INTERRUPT DID NOT OCCUR  
  
7$: CMP (SP)+,(SP)+ ;:RESTORE SP AFTER INTERRUPT  
BIC #BIT6,@LKS ;:DISABLE INTERRUPTS  
MOV R3,@RTCVT ;:RESTORE LINE CLOCK VECTOR  
MOV R4,@RTCP SW ;:RESTORE LINE CLOCK PSW VECTOR
```

```

3194
3195
3196 012766 000004
3197 012770 012737 000064 001104
3198 012776 032737 000001 001120
3199 013004 001403
3200 013006 005737 001106
3201 013012 001052
3202 013014
3203 013014 017703 170012
3204 013020 017704 170010
3205 013024 012777 013074 170000
3206 013032 012777 000340 167774
3207 013040 004737 015056
3208 013044 000240
3209 013046 042777 000200 167734
3210 013054 052777 000100 167726
3211 013062 105777 167722
3212 013066 100375
3213 013070 000240
3214
3215 013072 104146
3216
3217 013074 022626
3218 013076 012777 013116 167726
3219 013104 004737 015056
3220 013110 000240
3221 013112 000240
3222 013114 000402
3223
3224 013116 022626
3225 013120 104147
3226
3227
3228 013122 042777 000100 167660
3229 013130 010377 167676
3230 013134 010477 167674
3231
3232
3233
3234
3235
3236 013140 000004
3237 013142 012737 000065 001104
3238 013150 032737 000001 001120
3239 013156 001403
3240 013160 005737 001106
3241 013164 001036
3242 013166
3243 013166 004737 015056
3244 013172 000340
3245 013174 017703 167632
3246 013200 012777 013252 167624
3247 013206 042777 000200 167574
3248 013214 052777 000100 167566
3249 013222 105777 167562
  
```

```

;*TEST 64 TEST RTC FOR DOUBLE INTERRUPTS
*****
TST64: SCOPE
MOV #64,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#$ENV ;;ARE WE RUNNING UNDER APT
BEQ 70$ ;;IF NO THEN DO TEST
TST @#$PASS ;;IS THIS FIRST PASS
BNE TST65 ;;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
.WORD 240
BIC #BIT7,@LKS ;;CLEAR CLOCK DONE FLAG
BIS #BIT6,@LKS ;;ENABLE CLOCK INTERRUPTS
1$: TSTB @LKS ;;WAIT FOR DONE
BPL 1$
NOP ;;GIVE TIME FOR ANY INTERRUPT

ERROR 146 ;;RTC INTERRUPT DID NOT OCCUR

2$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER INTERRUPT
MOV #3$,@RTCVT ;;POINT RTC VECTOR TO ERROR REPORT
JSR PC,WRPSW ;;SET PSW TO PRIORITY 5
.WORD 240
NOP ;;GIVE SOME TIME FOR AN INTERRUPT
BR 4$ ;;NO INTERRUPT - BR TO END OF TEST

3$: CMP (SP)+,(SP)+ ;;RESTORE SP AFTER INTERRUPT
ERROR 147 ;;INTERRUPT SEQUENCE DID NOT CLEAR
;;INTERRUPT REQUEST

4$: BIC #BIT6,@LKS ;;DISABLE CLOCK INTERRUPTS
MOV R3,@RTCVT ;;RESTORE LINE CLOCK VECTOR
MOV R4,@RTCP SW ;;RESTORE LINE CLOCK PSW VECTOR
  
```

```

*****
;*TEST 65 TEST THAT RTC INTERRUPT CLEARS WITH RESET
*****
TST65: SCOPE
MOV #65,$TESTN ;;SET TEST NUMBER IN APT MAIL BOX
BIT #1,@#$ENV ;;ARE WE RUNNING UNDER APT
BEQ 70$ ;;IF NO THEN DO TEST
TST @#$PASS ;;IS THIS FIRST PASS
BNE TST66 ;;IF NO THEN SKIP TO NEXT TEST

70$: JSR PC,WRPSW ;;SET PRIORITY TO 7
.WORD 340
MOV @RTCVT,R3 ;;SAVE LINE CLOCK VECTOR
MOV #2$,@RTCVT ;;POINT RTC VECTOR TO ERROR REPORT
BIC #BIT7,@LKS ;;CLEAR CLOCK DONE FLAG
BIS #BIT6,@LKS ;;ENABLE CLOCK INTERRUPTS
1$: TSTB @LKS ;;WAIT FOR DONE (INTERRUPT REQUEST)
  
```



```

3250 013226 100375          BPL      1$
3251 013230 000005          RESET
3252 013232 004737 015056  JSR      PC,WRPSW      ;CLEAR PENDING INTERRUPT WITH RESET
3253 013236 000240          .WORD   240           ;SET PRIORITY TO 5
3254 013240 000240          NOP
3255 013242 042777 000100 167540 BIC     #BIT6,@LKS     ;GIVE TIME FOR ANY INTERRUPT
3256 013250 000402          BR      3$           ;DISALLOW INTERRUPTS
3257
3258 013252 022626          2$:    CMP     (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
3259 013254 104150          ERROR   150         ;RESET DID NOT CLEAR INTERRUPT
3260
3261 013256 010377 167550  3$:    MOV     R3,@RTC.VT ;RESTORE LINE CLOCK VECTOR
3262
3263
3264
3265
3266
3267 013262 000004          ::*****
3268 013264 012737 000066 001104  *TEST 66 TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
3269 013272 032737 000001 001120  ::*****
3270 013300 001403          TST66: SCOPE
3271 013302 005737 001106  MOV     #66,$TESTN    ;:SET TEST NUMBER IN APT MAIL BOX
3272 013306 001043          BIT     #1,@#$ENV     ;:ARE WE RUNNING UNDER APT
3273 013310          BEQ     70$           ;:IF NO THEN DO TEST
3274 013310 004737 015056  70$:   JSR     PC,WRPSW    ;:SET PRIORITY TO 7
3275 013314 000340          .WORD   340
3276 013316 017703 167510  MOV     @RTC.VT,R3    ;:SAVE LINE CLOCK VECTOR
3277 013322 012777 013400 167502  MOV     #2,@RTC.VT   ;:POINT RTC VECTOR TO ERROR REPORT
3278 013330 042777 000200 167452  BIC     #BIT7,@LKS   ;:CLEAR CLOCK DONE FLAG
3279 013336 052777 000100 167444  BIS     #BIT6,@LKS   ;:ENABLE CLOCK INTERRUPTS
3280 013344 105777 167440  1$:   TSTB   @LKS        ;:WAIT FOR DONE (INTERRUPT REQUEST)
3281 013350 100375          BPL     1$
3282 013352 042777 000200 167430  BIC     #BIT7,@LKS   ;:CLEAR DONE & INTERRUPT
3283 013360 004737 015056  JSR     PC,WRPSW    ;:ALLOW INTERRUPTS
3284 013364 000240          .WORD   240
3285 013366 000240          NOP
3286 013370 042777 000100 167412  BIC     #BIT6,@LKS   ;:GIVE TIME FOR ANY INTERRUPT
3287 013376 000402          BR      3$           ;:DISALLOW INTERRUPTS
3288
3289
3290 013400 022626          2$:   CMP     (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
3291 013402 104151          ERROR   151         ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT
3292
3293 013404 010377 167422  3$:   MOV     R3,@RTC.VT ;RESTORE LINE CLOCK VECTOR
3294 013410 004737 015056  JSR     PC,WRPSW    ;RESTORE PRIORITY TO 7
3295 013414 000340          .WORD   340
3296
3297
3298
3299
3300
3301 013416 000004          ::*****
3302 013420 012737 000067 001104  *TEST 67 TEST CLOCK REPEATABILITY
3303 013426 032737 000001 001120  ::*****
3304 013434 001403          TST67: SCOPE
3305 013436 005737 001106  MOV     #67,$TESTN    ;:SET TEST NUMBER IN APT MAIL BOX
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```



```

3306 013442 001056          BNE     TST70          ;IF NO THEN SKIP TO NEXT TEST
3307 013444                70$:    BIC     #BIT6,@LKS      ;DISALLOW INTERRUPTS
3308 013444 042777 000100 167336 CLR     R0              ;CLEAR A TIMER
3309 013452 005000                MOV     #-1,R1         ;SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
3310 013454 012701 177777                CLR     R2              ;CLEAR CLOCK COUNTER
3311 013460 005002                CLR     @LKS           ;CLEAR DONE
3312 013462 005077 167322                TSTB   @LKS           ;SYNC ON DONE
3313 013466 105777 167316                BPL     2$              ;CLEAR DONE
3314 013472 100375                CLR     @LKS           ;IS CLOCK DONE?
3315 013474 005077 167310                TSTB   @LKS           ;BR IF NOT , TO INCREMENT TIMER
3316 013500 105777 167304                BPL     4$              ;IF DONE, INCREMENT CLOCK COUNT
3317 013504 100003                INC     R2              ;CLEAR DONE
3318 013506 005202                CLR     @LKS           ;INCREMENT TIMER
3319 013510 005077 167274                INC     R0              ;BR IF TIME REMAINS
3320 013514 005200                BNE     3$              ;INCREMENT LOOP PASS FLAG
3321 013516 001370                INC     R1              ;BR IF TWO PASSES HAVE BEEN MADE
3322 013520 005201                BNE     CMPARE          ;IF NOT, STORE FIRST CLOCK COUNT
3323 013522 001003                MOV     R2,FIRST        ;DO LOOP AGAIN
3324 013524 010237 013574                BR      1$              ;RECALL FIRST CLOCK COUNT
3325 013530 000753                CMPARE: MOV    FIRST,R1  ;CALCULATE DIFFERENCE OF TWO COUNTS
3326 013532 013701 013574                SUB     R2,R1           ;IF POSITIVE,SKIP NEGATION OF DIFFERENCE
3327 013536 160201                BPL     TOLER           ;MAKE DIFFERENCE A POSITIVE NUMBER
3328 013540 100001                NEG     R1              ;COMPARE DIFFERENCE WITH DESIRED TOLERANCE
3329 013542 005401                TOLER: CMP    R1,#2     ;BR, IF LOWER/EQUAL TO TOLERANCE
3330 013544 020127 000002                BLE     5$              ;STORE SECOND COUNT
3331 013550 003403                MOV     R2,SECND        ;CLOCK REPEATABILITY ERROR
3332 013552 010237 013576                ERROR  152
3333 013556 104152                5$:    BIT     #BIT5,@SWR  ;CLOCK TESTS ONLY?
3334 013560 032777 000040 165252                BEQ     TST70          ;BR IF NOT
3335 013566 001404                JMP     $EOP           ;ELSE, JUMP TO END OF PASS ROUTINE
3336 013570 000137 014740
3337 013574 000000
3338 013576 000000
3339
3340 FIRST: 0
3341 SECND: 0
  
```


3342 013600
3343
3344
3345
3346
3347 013600 000004
3348 013602 012737 000070 001104
3349 013610 032737 000001 001120
3350 013616 001405
3351 013620 005737 001106
3352 013624 001402
3353 013626 000137 014740
3354 013632 000005
3355 013634 012737 000340 177776
3356 013642 017737 167150 001060
3357 013650 017737 167136 001062
3358 013656 017737 167144 001064
3359 013664 017737 167132 001066
3360 013672 017737 167134 001070
3361 013700 005037 014726
3362 013704 005037 014730
3363 013710 005037 014732
3364 013714 005037 014734
3365 013720 005037 014736
3366 013724 012777 014154 167074
3367 013732 012777 000340 167070
3368 013740 012777 014210 167054
3369 013746 012777 000340 167050
3370 013754 012777 014270 167034
3371 013762 012777 000340 167030
3372 013770 012777 014324 167014
3373 013776 012777 000340 167010
3374 014004 012777 014144 167020
3375 014012 012777 000340 167014
3376 014020 032737 000001 001156
3377 014026 001413
3378 014030 052777 000004 166746
3379 014036 052777 000100 166740
3380 014044 052777 000100 166726
3381 014052 012702 030450
3382 014056 032737 000002 001156
3383 014064 001410
3384 014066 052777 000100 166700
3385 014074 052777 000100 166666
3386 014102 012703 031050
3387 014106 032737 000004 001156
3388 014114 001403
3389 014116 052777 000100 166664
3390 014124 012700 177777
3391 014130 012701 177777
3392 014134 005037 177776
3393 014140 000001
3394 014142 000776
3395
3396 014144 005237 014736
3397 014150 000137 014400

BLAST:

```
::*****  
:*TEST 70 TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY  
:*****  
TST70: SCOPE  
MOV #70,$TESTN ;:SET TEST NUMBER IN APT MAIL BOX  
BIT #BIT0,@#$ENV ;:ARE WE RUNNING UNDER APT  
BEQ 70$ ;:IF NO DO TEST  
TST @#$PASS ;:IS THIS FIRST PASS  
BEQ 70$ ;:IF YES DO TEST  
JMP $EOP ;:IF NO DO NOT DO TEST  
70$: RESET ;:CLEAR EVERY BODY  
MOV #340,PS ;:SET PROCESSOR PRIORITY TO 7  
MOV @TVECT,$TMP0  
MOV @RVECT,$TMP1  
MOV @CTVECT,$TMP2  
MOV @CRVECT,$TMP3  
MOV @RTCVT,$TMP4  
CLR XMTCT1 ;:INITIALIZE COUNTERS  
CLR XMTCT2  
CLR RECCT1  
CLR RECCT2  
CLR COUNT  
MOV #XMIT1,@CTVECT ;:SET UP SLU1 TRANSMIT VECTOR  
MOV #340,@CTPSW ;:AND PSW  
MOV #REC1,@CRVECT ;:SET UP SLU1 RECEIVER VECTOR  
MOV #340,@CRPSW ;:AND PSW  
MOV #XMIT2,@TVECT ;:SET UP SLU2 TRANSMIT VECTOR  
MOV #340,@TPSW ;:AND PSW  
MOV #REC2,@RVECT ;:SET UP SLU2 RECEIVER VECTOR  
MOV #340,@RPSW ;:AND PSW  
MOV #TICKER,@RTCVT ;:SET UP RTC VECTOR  
MOV #340,@RTCPSW ;:AND PSW  
BIT #BIT0,$DEVM ;:IS SLU1 UNDER TEST  
BEQ 1$ ;:IF NO DON'T TURN IT ON  
BIS #BIT2,@CTCSR ;:ENABLE SLU1 MAINTENANCE WRAP  
BIS #BIT6,@CTCSR ;:ENABLE SLU1 XMIT INTERRUPT  
BIS #BIT6,@CRCSR ;:ENABLE SLU1 RECEIVER INTERRUPT  
MOV #BUF1,R2 ;:SET UP RECEIVER BUFFER  
BIT #BIT1,$DEVM ;:IS SLU2 UNDER TEST  
BEQ 2$ ;:IF NO DON'T SET IT UP  
BIS #BIT6,@TCSR ;:ENABLE SLU2 XMIT INTERRUPT  
BIS #BIT6,@RCRCSR ;:ENABLE SLU2 RECEIVER INTERRUPT  
MOV #BUF2,R3 ;:SET UP RECEIVER BUFFER  
BIT #BIT2,$DEVM ;:IS LTC UNDER TEST  
BEQ 3$ ;:IF NO DON'T SET IT UP  
BIS #BIT6,@LKS ;:ENABLE RTC INTERRUPTS  
3$: MOV #-1,R0 ;:INITIALIZE DATA FOR SLU1(1ST CHR. 0)  
MOV #-1,R1 ;:INITIALIZE DATA FOR SLU2(1ST CHR. 0)  
CLR PS ;:DROP PROCESSOR PRIORITY TO ZERO  
WAITIO: WAIT ;:WAIT FOR INTERRUPT  
BR WAITIO  
TICKER: INC COUNT ;:UPDATE COUNT  
JMP IOHAND ;:GO TO INTERRUPT HANDLER
```


3398	014154	005237	014726		XMIT1:	INC	XMTCT1	:UPDATE XMIT INTERRUPT COUNT
3399	014160	005200				INC	R0	:UPDATE XMIT DATA
3400	014162	010077	166620			MOV	R0, @CTBUF	:SEND NEXT CHARACTER
3401	014166	023727	014726	000400		CMP	XMTCT1, #400	:IF 256 CHARACTERS HAVE NOT BEEN
3402	014174	002403				BLT	1\$:TRANSFERRED CONTINUE
3403	014176	042777	000100	166600		BIC	#BIT6, @CTCSR	:IF YES TURN OFF TRANSMITTER INTERRUPTS FIRST
3404	014204	000137	014400		1\$:	JMP	IOHAND	:GO TO INTERRUPT HANDLER
3405								
3406	014210	005237	014732		REC1:	INC	RECCT1	:UPDATE RECEIVER INTERRUPT COUNT
3407	014214	005777	166562			TST	@CRBUF	:BIT 15 SET IF ANY ERRORS OCCURRED
3408	014220	100017				BPL	3\$:IF BIT IS CLEAR NO ERROR
3409	014222	017737	166554	001026		MOV	@CRBUF, \$BDDAT	:GET ERROR INFORMATION
3410	014230	000005				RESET		:CLEAR THE WORLD STOP ALL
3411								:INTERRUPTS
3412	014232	020227	030450			CMP	R2, #BUF1	:WAS MORE THAN 1 WORD TRANSFERRED
3413	014236	003004				BGT	1\$:IF YES GET LAST GOOD DATA
3414	014240	012737	177777	001024		MOV	#-1, \$GDDAT	:MAKE GOOD DATA = -1
3415	014246	000403				BR	2\$:GO TO ERROR REPORT
3416	014250	116237	177777	001024	1\$:	MOVB	-1(R2), \$GDDAT	:GET LAST GOOD DATA
3417	014256	104153			2\$:	ERROR	153	:RECEIVER STATUS ERROR
3418	014260	117722	166516		3\$:	MOVB	@CRBUF, (R2)+	:GET DATA AND STORE IT
3419	014264	000137	014400			JMP	IOHAND	:GO TO INTERRUPT HANDLER
3420								
3421	014270	005237	014730		XMIT2:	INC	XMTCT2	:UPDATE XMIT INTERRUPT COUNT
3422	014274	005201				INC	R1	:UPDATE XMIT DATA
3423	014276	010177	166474			MOV	R1, @TBUF	:SEND NEXT CHARACTER
3424	014302	023727	014730	000400		CMP	XMTCT2, #400	:IF 256 CHARACTERS HAVE NOT
3425	014310	002403				BLT	1\$:BEEN TRANSFERRED CONTINUE
3426	014312	042777	000100	166454		BIC	#BIT6, @TCSR	:ELSE NO MORE XMIT INTERRUPTS
3427	014320	000137	014400		1\$:	JMP	IOHAND	:GO TO INTERRUPT HANDLER
3428								
3429	014324	005237	014734		REC2:	INC	RECCT2	:UPDATE RECEIVER INTERRUPT COUNT
3430	014330	005777	166436			TST	@RBUF	:BIT 15 SETS IF ANY ERRORS OCCURRED
3431	014334	100017				BPL	3\$:IF BIT IS CLEAR NO ERRORS
3432	014336	017737	166430	001026		MOV	@RBUF, \$BDDAT	:GET ERROR INFORMATION
3433	014344	000005				RESET		:CLEAR THE WORLD - STOP ALL
3434								:INTERRUPTS
3435	014346	020327	031050			CMP	R3, #BUF2	:WAS MORE THAN 1 WORD TRANSFERRED
3436	014352	003004				BGT	1\$:IF YES GET LAST GOOD DATA
3437	014354	012737	177777	001024		MOV	#-1, \$GDDAT	:IF NO MAKE GOOD DATA -1
3438	014362	000403				BR	2\$:AND GET TO ERROR REPORT
3439	014364	116337	177777	001024	1\$:	MOVB	-1(R3), \$GDDAT	:GET LAST GOOD DATA RECEIVED
3440	014372	104154			2\$:	ERROR	154	:RECEIVER STATUS ERROR
3441	014374	117723	166372		3\$:	MOVB	@RBUF, (R3)+	:GET DATA AND STORE IT
3442								
3443	014400	032737	000004	001156	IOHAND:	BIT	#BIT2, \$DEVN	:IS RTC UNDER TEST
3444	014406	001416				BEQ	1\$:IF NO CHECK OTHER DEVICES
3445	014410	023727	014736	000074		CMP	COUNT, #74	:HAS 1 SEC ELAPSED
3446	014416	002427				BLT	3\$:IF NO CONTINUE TEST
3447	014420	042777	000100	166356		BIC	#BIT6, @CTCSR	:IF YES STOP TRANSMISSIONS
3448	014426	042777	000100	166340		BIC	#BIT6, @TCSR	:
3449	014434	042777	000100	166346		BIC	#BIT6, @LKS	:TURN OFF LINE CLOCK
3450	014442	000416				BR	WAITER	
3451	014444	032737	000001	001156	1\$:	BIT	#BIT0, \$DEVN	:IS SLU1 UNDER TEST
3452	014452	001405				BEQ	2\$:IF NO CHECK FOR SLU2
3453	014454	032777	000100	166322		BIT	#BIT6, @CTCSR	:IS TRANSMITTER SHUTDOWN


```

3454 014462 001405      BEQ      3$      ;IF NO CONTINUE TEST
3455 014464 000405      BR       WAITER ;IF YES GO WAIT FOR POSSIBLE
3456
3457 014466 032777 000100 166300 2$:  BIT     #BIT6, @TCSR ;IS TRANSMITTER SHUT DOWN
3458 014474 001001      BNE     WAITER  ;IF YES GO WAIT FOR POSSIBLE
3459
3460 014476 000002      3$:  RTI      ;RETURN FROM INTERRUPT TO AWAIT NEXT
3461
3462 014500 005037 177776      WAITER: CLR    PS      ;MAKE PROCESSOR PRIORITY 0
3463 014504 012705 140000      MOV     #-40000, R5 ;SET UP LOOP COUNTER
3464 014510 062705 000001      1$:  ADD     #1, R5    ;DO LOOP UNTIL R5 = 0
3465 014514 001375      BNE     1$
3466 014516 000005      RESET
3467 014520 012706 001000      MOV     #1000, SP ;STOP EVERYONE SHOULD BE DONE
3468
3469 014524 032737 000001 001156 CHECK1: BIT     #BIT0, $DEVM ;SLU1 UNDER TEST
3470 014532 001424      BEQ     CHECK2  ;IF NO GO CHECK SLU2 DATA
3471 014534 023737 014726 014732      CMP     XMTCT1, RECCT1 ;# OF XMIT INTERRUPTS = REC INTERRUPTS
3472 014542 001401      BEQ     1$      ;IF YES GET OVER ERROR AND CHECK DATA
3473 014544 104155      ERROR  155     ;INTERRUPT COMPARISON ERROR
3474 014546 012702 030450      1$:  MOV     #BUF1, R2 ;POINT TO FIRST DATA
3475 014552 005000      CLR     R0      ;INITIALIZE TO FIRST DATA XMIT
3476 014554 013704 014726      MOV     XMTCT1, R4 ;GET # OF BYTES TRANSFERRED
3477 014560 122200      2$:  CMPB   (R2)+, R0 ;IS RECEIVED DATA = EXPECTED
3478 014562 001406      BEQ     3$      ;IF YES CONTINUE
3479 014564 114237 001026      MOVB   -(R2), $BDDAT ;IF NO GET ERROR INFORMATION
3480 014570 010037 001024      MOV     R0, $GDDAT
3481 014574 104156      ERROR  156     ;SLU1 DATA COMPARISON ERROR
3482 014576 005202      INC     R2      ;IF CONTINUE ON ERROR RESET POINTER
3483 014600 005200      3$:  INC     R0      ;UPDATE TO NEXT GOOD DATA
3484 014602 077412      SOB    R4, 2$  ;LOOP UNTIL ALL DATA CHECKED
3485 014604 032737 000002 001156 CHECK2: BIT     #BIT1, $DEVM ;SLU2 UNDER TEST
3486 014612 001424      BEQ     FINIE   ;IF NO WE'RE DONE
3487 014614 023737 014730 014734      CMP     XMTCT2, RECCT2 ;#OF XMIT INTERRUPTS = REC INTERRUPTS
3488 014622 001401      BEQ     1$      ;IF YES CHECK DATA
3489 014624 104157      ERROR  157     ;INTERRUPT COMPARISON ERROR
3490 014626 012703 031050      1$:  MOV     #BUF2, R3 ;INITIALIZE TO FIRST RECEIVED DATA
3491 014632 005001      CLR     R1      ;INITIALIZE TO FIRST XMIT DATA
3492 014634 013704 014730      MOV     XMTCT2, R4 ;GET # OF BYTES TRANSFERRED
3493 014640 122301      2$:  CMPB   (R3)+, R1 ;IS RECEIVED DATA = EXPECTED DATA
3494 014642 001406      BEQ     3$      ;IF YES CONTINUE TESTING
3495 014644 114337 001026      MOVB   -(R3), $BDDAT ;IF NO GET ERROR INFORMATION
3496 014650 010137 001024      MOV     R1, $GDDAT
3497 014654 104160      ERROR  160     ;SLU2 DATA COMPARISON ERROR
3498 014656 005203      INC     R3      ;IF COONTINUE ON ERROR RESET POINTER
3499 014660 005201      3$:  INC     R1      ;UPDATE TO NEXT GOOD DATA
3500 014662 077412      SOB    R4, 2$  ;LOOP UNTIL ALL DATA CHECKED
3501 014664 013777 001060 166124 FINIE: MOV     $TMP0, @TVECT ;RESTORE VECTORS
3502 014672 013777 001062 166112      MOV     $TMP1, @RVECT
3503 014700 013777 001064 166120      MOV     $TMP2, @CTVECT
3504 014706 013777 001066 166106      MOV     $TMP3, @CRVECT
3505 014714 013777 001070 166110      MOV     $TMP4, @RTCVT
3506 014722 000137 014740      JMP     $EOP
3507
3508 014726 000000      XMTCT1: .WORD 0
3509 014730 000000      XMTCT2: .WORD 0
;FINISHED TESTING GO TO END OF PASS

```


CJKDFAO 11/24 OPTIONS DIAGNOSTIC
CJKDFA.P11 19-MAR-81 14:27

MACY11 30A(1052) 19-MAR-81 14:27 PAGE 70
170 TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY

SEQ 0070

3510 014732 000000
3511 014734 000000
3512 014736 000000
3513

RECCT1: .WORD 0
RECCT2: .WORD 0
COUNT: .WORD 0

3514
 3515
 3516
 3517
 3518
 3519
 3520
 3521
 3522
 3523 014740
 3524 014740 000004
 3525 014742 005037 001002
 3526 014746 005237 001106
 3527 014752 042737 100000 001106
 3528 014760 005327
 3529 014762 000001
 3530 014764 003022
 3531 014766 012737
 3532 014770 000001
 3533 014772 014762
 3534 014774 104401 015041
 3535 015000 013746 001106
 3536 015004 104405
 3537 015006 104401 015036
 3538 015012 013700 000042
 3539 015016 001405
 3540 015020 000005
 3541 015022 004710
 3542 015024 000240
 3543 015026 000240
 3544 015030 000240
 3545 015032
 3546 015032 000137
 3547 015034 003450
 3548 015036 377 377 000
 3549 015041 015 042412 042116
 3550 015046 050040 051501 020123
 3551 015054 000043
 3552
 3553 015056 011646
 3554 015060 013616
 3555 015062 062746 000002
 3556 015066 000002
 3557
 3558
 3559
 3560 015070 012600
 3561 015072 162700 000004
 3562 015076 010037 015116
 3563 015102 016637 000002 015114
 3564 015110 104161
 3565
 3566 015112 000000
 3567 015114 000000
 3568 015116 000000
 3569

.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 $STNM          ;; 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 $ENDMG       ;; 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
  
```

\$ENDMG: .ASCIIZ

```

<15><12>/END PASS #/
  
```

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
  
```

; 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
  
```

HALT

```

;; PROGRAM MUST BE RESTARTED AT THIS POINT
OLDPC: .WORD 0
BDVECT: .WORD 0
  
```


3570
 3571
 3572
 3573
 3574
 3575
 3576
 3577
 3578
 3579
 3580
 3581
 3582
 3583 015120
 3584 015120 105237 001003
 3585 015124 001775
 3586 015126 013777 001002 163706
 3587 015134 005237 001012
 3588 015140 011637 001016
 3589 015144 162737 000002 001016
 3590 015152 117737 163640 001014
 3591 015160 032777 020000 163652
 3592 015166 001004
 3593 015170 004737 015302
 3594 015174 104401 001075
 3595 015200
 3596 015200 122737 000001 001120
 3597 015206 001007
 3598 015210 113737 001014 015222
 3599 015216 004737 016004
 3600 015222 000
 3601 015223 000
 3602 015224 000777
 3603 015226 005777 163606
 3604 015232 100001
 3605 015234 000000
 3606 015236 104407
 3607 015240 032777 001000 163572
 3608 015246 001402
 3609 015250 013716 001010
 3610 015254 005737 001072
 3611 015260 001402
 3612 015262 013716 001072
 3613 015266
 3614 015266 022737 015022 000042
 3615 015274 001001
 3616 015276 000000
 3617 015300
 3618 015300 000002
 3619

```

*****
*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      $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
        INC      $ERTTL      ;INCREMENT ERROR COUNT
        MOV      (SP), $ERRPC   ;GET ADDRESS OF ERROR INSTRUCTION
        SUB      #2, $ERRPC
        MOVVB   @$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    , $CRLF

20$:
        CMPB    #APTENV, $ENV   ;RUNNING IN APT MODE
        BNE     2$             ;NO, SKIP APT ERROR REPORT
        MOVVB   $ITEMB, 21$     ;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$:      TST      @SWR
        BPL     3$             ;HALT ON ERROR
        HALT    ;SKIP IF CONTINUE
        ;HALT ON ERROR!

3$:      CKSWR
        BIT     #BIT09, @SWR   ;TEST FOR CHANGE IN SOFT-SWR
        BEQ     4$             ;LOOP ON ERROR SWITCH SET?
        BR     IF NO

4$:      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
  
```



```

3620
3621
3622
3623
3624
3625
3626
3627
3628 015302
3629 015302 104401 001075
3630 015306 010046
3631 015310 005000
3632 015312 153700 001014
3633 015316 001004
3634
3635 015320 013746 001016
3636
3637 015324 104402
3638 015326 000426
3639 015330 005300
3640 015332 006300
3641 015334 006300
3642 015336 006300
3643 015340 062700 001160
3644 015344 012037 015354
3645 015350 001404
3646 015352 104401
3647 015354 000000
3648 015356 104401 001075
3649 015362 012037 015372
3650 015366 001404
3651 015370 104401
3652 015372 000000
3653 015374 104401 001075
3654 015400 011000
3655 015402 001004
3656 015404 012600
3657 015406 104401 001075
3658 015412 000207
3659 015414
3660 015414 013046
3661 015416 104402
3662 015420 005710
3663 015422 001770
3664 015424 104401 015432
3665 015430 000771
3666 015432 020040 000
3667 015436
3668
  
```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

*****
*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      , $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
      MOV      R0, -(SP)    :: SAVE R0
      CLR      R0          :: PICKUP THE ITEM INDEX
      BISB     @#$ITEMB, R0
      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)
      MOV      $ERRPC, -(SP)
                          :: GET OUT
                          :: ADJUST THE INDEX SO THAT IT WILL
                          :: WORK FOR THE ERROR TABLE
      TYPOC
      BR      6$
1$:   DEC      R0
      ASL     R0
      ASL     R0
      ASL     R0
      ADD     #$ERRTB, R0  :: FORM TABLE POINTER
      MOV     (R0)+, 2$    :: PICKUP "ERROR MESSAGE" POINTER
      BEQ     3$          :: SKIP TYPEOUT IF NO POINTER
      TYPE   0            :: TYPE THE "ERROR MESSAGE"
                          :: "ERROR MESSAGE" POINTER GOES HERE
      TYPE   , $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
      MOV     (R0)+, 4$    :: PICKUP "DATA HEADER" POINTER
      BEQ     5$          :: SKIP TYPEOUT IF 0
      TYPE   0            :: TYPE THE "DATA HEADER"
                          :: "DATA HEADER" POINTER GOES HERE
      TYPE   , $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
      MOV     (R0), R0     :: PICKUP "DATA TABLE" POINTER
      BNE     7$          :: GO TYPE THE DATA
      MOV     (SP)+, R0    :: RESTORE R0
      TYPE   , $CRLF      :: "CARRIAGE RETURN" & "LINE FEED"
      RTS     PC          :: RETURN
      MOV     @ (R0)+, -(SP) :: SAVE @ (R0)+ FOR TYPEOUT
      TYPOC
      TST     (R0)        :: GO TYPE--OCTAL ASCII(ALL DIGITS)
      BEQ     6$          :: IS THERE ANOTHER NUMBER?
      TYPE   , 8$         :: BR IF NO
                          :: TYPE TWO(2) SPACES
      BR      7$          :: LOOP
8$:   .ASCIZ  / /        :: TWO(2) SPACES
      .EVEN
  
```


3669
 3670
 3671
 3672
 3673
 3674 015436 012737 015576 000024
 3675 015444 012737 000340 000026
 3676 015452 010046
 3677 015454 010146
 3678 015456 010246
 3679 015460 010346
 3680 015462 010446
 3681 015464 010546
 3682 015466 017746 163346
 3683 015472 010637 015602
 3684 015476 012737 015510 000024
 3685 015504 000000
 3686 015506 000776
 3687
 3688
 3689
 3690
 3691
 3692 015510 012737 015576 000024
 3693 015516 013706 015602
 3694 015522 012677 163312
 3695 015526 012605
 3696 015530 012604
 3697 015532 012603
 3698 015534 012602
 3699 015536 012601
 3700 015540 012600
 3701 015542 012737 015436 000024
 3702 015550 012737 000340 000026
 3703 015556 005037 015602
 3704 015562 005237 015602
 3705 015566 001375
 3706 015570 104401
 3707 015572 015604
 3708 015574 000002
 3709 015576 000000
 3710 015600 000776
 3711 015602 000000
 3712 015604 005015 047520 042527
 3713 015612 000122
 3714

```

.SBTTL POWER DOWN AND UP ROUTINES
:*****
;*POWER DOWN ROUTINE
:*****
$PWRDN: MOV    #$ILLUP,@#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
:*****
$PWRUP: MOV    #$ILLUP,@#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   $SAVR6        ;REPORT THE POWER FAILURE
        $PWRMG: .WORD $POWER ;POWER FAIL MESSAGE POINTER
        RTI
        $ILLUP: HALT        ;THE POWER UP SEQUENCE WAS STARTED
        BR     .-2            ; BEFORE THE POWER DOWN WAS COMPLETE
        $SAVR6: 0            ;PUT THE SP HERE
        $POWER: .ASCIZ <15><12>'POWER'
  
```


3715
 3716
 3717
 3718
 3719
 3720
 3721
 3722
 3723
 3724
 3725
 3726
 3727
 3728 015614
 3729 015614 104407
 3730 015616 032777 040000 163214
 3731 015624 001052
 3732
 3733 015626 000416
 3734
 3735 015630 013746 000004
 3736 015634 012737 015654 000004
 3737 015642 005737 177060
 3738 015646 012637 000004
 3739 015652 000421
 3740 015654 022626
 3741 015656 012637 000004
 3742 015662 000407
 3743 015664
 3744 015664 105737 001003
 3745 015670 001412
 3746 015672 032777 001000 163140
 3747 015700 001404
 3748 015702 013737 001010 001006
 3749 015710 000420
 3750 015712 105037 001003
 3751 015716 105237 001002
 3752 015722 113737 001002 001104
 3753 015730 011637 001006
 3754 015734 011637 001010
 3755 015740 005037 001072
 3756 015744 112737 000001 001015
 3757 015752 013777 001002 163062
 3758 015760 013716 001006
 3759 015764 000002
 3760

.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW09=1      LOOP ON ERROR
*CALL
*          SCOPE          ;;SCOPE=IOT

$SCOPE:
1$:      CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
        BIT          #BIT14,@SWR      ;;LOOP ON PRESENT TEST?
        BNE          $OVER          ;;YES IF SW14=1
        ;#####START OF CODE FOR THE XOR TESTER#####
$XTSTR:  BR          6$          ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
        ;THIS INSTRUCTION TO A 'NOP' (NOP=240)
        MOV          @#ERRVEC,-(SP)   ;;SAVE THE CONTENTS OF THE ERROR VECTOR
        MOV          #5$,@#ERRVEC    ;;SET FOR TIMEOUT
        TST          @#177060        ;;TIME OUT ON XOR?
        MOV          (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
        BR          $SVLAD          ;;GO TO THE NEXT TEST
5$:      CMP          (SP)+,(SP)+    ;;CLEAR THE STACK AFTER A TIME OUT
        MOV          (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
        BR          7$          ;;LOOP ON THE PRESENT TEST
6$:      ;#####END OF CODE FOR THE XOR TESTER#####
2$:      TSTB         $ERFLG        ;;HAS AN ERROR OCCURRED?
        BEQ          $SVLAD        ;;BR IF NO
        BIT          #BIT09,@SWR    ;;LOOP ON ERROR?
        BEQ          4$          ;;BR IF NO
7$:      MOV          $LPERR,$LPADR  ;;SET LOOP ADDRESS TO LAST SCOPE
        BR          $OVER
4$:      CLRB         $ERFLG        ;;ZERO THE ERROR FLAG
$SVLAD:  INCB         $TSTNM        ;;COUNT TEST NUMBERS
        MOVB        $TSTNM,$TESTN  ;;SET TEST NUMBER IN APT MAILBOX
        MOV          (SP),$LPADR    ;;SAVE SCOPE LOOP ADDRESS
        MOV          (SP),$LPERR    ;;SAVE ERROR LOOP ADDRESS
        CLR         $ESCAPE        ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
        MOVB        #1,$ERMAX      ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
$OVER:  MOV          $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
        MOV          $LPADR,(SP)    ;;FUDGE RETURN ADDRESS
        RTI          ;;FIXES PS
  
```



```

3761
3762
3763
3764
3765
3766 015766 112737 000001 016232 $ATY1: MOVB #1,$FFLG ;TO REPORT FATAL ERROR
3767 015774 112737 000001 016230 $ATY3: MOVB #1,$MFLG ;TO TYPE A MESSAGE
3768 016002 000403
3769 016004 112737 000001 016232 $ATY4: MOVB #1,$FFLG ;TO ONLY REPORT FATAL ERROR
3770 016012 $ATYC:
3771 016012 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
3772 016014 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
3773 016016 105737 016230 TSTB $MFLG ;SHOULD TYPE A MESSAGE?
3774 016022 001450 BEQ 5$ ;IF NOT: BR
3775 016024 122737 000001 001120 CMPB #APTENV,$ENV ;OPERATING UNDER APT?
3776 016032 001031 BNE 3$ ;IF NOT: BR
3777 016034 132737 000100 001121 BITB #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?
3778 016042 001425 BEQ 3$ ;IF NOT: BR
3779 016044 017600 000004 MOV @4(SP),R0 ;GET MESSAGE ADDRESS
3780 016050 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS
3781 016056 005737 001100 1$: TST $MSGTYPE ;SEE IF DONE W/ LAST XMISSION?
3782 016062 001375 BNE 1$ ;IF NOT: WAIT
3783 016064 010037 001114 MOV R0,$MSGAD ;PUT ADDRESS IN MAILBOX
3784 016070 105720 2$: TSTB (R0)+ ;FIND END OF MESSAGE
3785 016072 001376 BNE 2$
3786 016074 163700 001114 SUB $MSGAD,R0 ;SUB START OF MESSAGE
3787 016100 006200 ASR R0 ;GET MESSAGE LENGTH IN WORDS
3788 016102 010037 001116 MOV R0,$MSGGLT ;PUT LENGTH IN MAILBOX
3789 016106 012737 000004 001100 MOV #4,$MSGTYPE ;TELL APT TO TAKE MESSAGE
3790 016114 000413 BR 5$
3791 016116 017637 000004 016142 3$: MOV @4(SP),4$ ;PUT MSG ADDR IN JSR LINKAGE
3792 016124 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS
3793 016132 013746 177776 MOV 177776,-(SP) ;PUSH 177776 ON STACK
3794 016136 004737 016234 JSR PC,$TYPE ;CALL TYPE MACRO
3795 016142 000000 4$: .WORD 0
3796 016144 5$:
3797 016144 105737 016232 10$: TSTB $FFLG ;SHOULD REPORT FATAL ERROR?
3798 016150 001413 BEQ 12$ ;IF NOT: BR
3799 016152 005737 001120 TST $ENV ;RUNNING UNDER APT?
3800 016156 001410 BEQ 12$ ;IF NOT: BR
3801 016160 005737 001100 11$: TST $MSGTYPE ;FINISHED LAST MESSAGE?
3802 016164 001375 BNE 11$ ;IF NOT: WAIT
3803 016166 017637 000004 001102 MOV @4(SP),$FATAL ;GET ERROR #
3804 016174 005237 001100 INC $MSGTYPE ;TELL APT TO TAKE ERROR
3805 016200 062766 000002 000004 12$: ADD #2,4(SP) ;BUMP RETURN ADDRESS
3806 016206 105037 016232 CLRB $FFLG ;CLEAR FATAL FLAG
3807 016212 105037 016231 CLRB $LFLG ;CLEAR LOG FLAG
3808 016216 105037 016230 CLRB $MFLG ;CLEAR MESSAGE FLAG
3809 016222 012601 MOV (SP)+,R1 ;POP STACK INTO R1
3810 016224 012600 MOV (SP)+,R0 ;POP STACK INTO R1
3811 016226 000207 RTS PC ;RETURN
3812 016230 000 $MFLG: .BYTE 0
3813 016231 000 $LFLG: .BYTE 0 ;LOG FLAG
3814 016232 000 $FFLG: .BYTE 0 ;FATAL FLAG
3815
3816 016234 .EVEN
  
```


CJKDFAO 11/24 OPTIONS DIAGNOSTIC
CJKDFA.P11 19-MAR-81 14:27

MACY11 30A(1052) 19-MAR-81^{M 6} 14:27 PAGE 77
APT COMMUNICATIONS ROUTINE

SEQ 0077

3817 000200
3818 000001
3819 000100
3820 000040
3821

APTSIZE=200
APTENV=001
APTSPOOL=100
APTC SUP=040

3822
 3823
 3824
 3825
 3826
 3827
 3828
 3829
 3830
 3831
 3832
 3833
 3834
 3835
 3836
 3837
 3838
 3839
 3840
 3841
 3842
 3843
 3844
 3845
 3846
 3847
 3848
 3849
 3850
 3851
 3852
 3853
 3854
 3855
 3856
 3857
 3858
 3859
 3860
 3861
 3862
 3863
 3864
 3865
 3866
 3867
 3868
 3869
 3870
 3871
 3872
 3873
 3874
 3875
 3876
 3877

016234 105737 001057
 016240 100002
 016242 000000
 016244 000430
 016246 010046
 016250 017600 000002
 016254 122737 000001 001120
 016262 001011
 016264 132737 000100 001121
 016272 001405
 016274 010037 016304
 016300 004737 015774
 016304 000000
 016306 132737 000040 001121
 016314 001003
 016316 112046
 016320 001005
 016322 005726
 016324 012600
 016326 062716 000002
 016332 000002
 016334 122716 000011
 016340 001430
 016342 122716 000200
 016346 001006
 016350 005726
 016352 104401
 016354 001075
 016356 105037 016564
 016362 000755
 016364 004737 016446
 016370 123726 001056
 016374 001350
 016376 013746 001054
 016402 105366 000001
 016406 002770
 016410 004737 016446

```

.SBTTL TYPE ROUTINE
*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
*
*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*
$TYPE: TSTB $TPFLG ;;IS THERE A TERMINAL?
BPL 1$ ;;BR IF YES
HALT ;;HALT HERE IF NO TERMINAL
BR 3$ ;;LEAVE
1$: MOV RO,-(SP) ;;SAVE RO
MOV @2(SP),RO ;;GET ADDRESS OF ASCIZ STRING
CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
BNE 62$ ;;NO,GO CHECK FOR APT CONSOLE
BITB #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
BEQ 62$ ;;NO,GO CHECK FOR CONSOLE
MOV RO,61$ ;;SETUP MESSAGE ADDRESS FOR APT
JSR PC,$ATY3 ;;SPOOL MESSAGE TO APT
61$: .WORD 0 ;;MESSAGE ADDRESS
62$: BITB #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
BNE 60$ ;;YES,SKIP TYPE OUT
2$: MOVB (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+,RO ;;RESTORE RO
3$: ADD #2,(SP) ;;ADJUST RETURN PC
RTI ;;RETURN
4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
BEQ 8$
CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
BNE 5$
TST (SP)+ ;;POP <CR><LF> EQUIV
TYPE ;;TYPE A CR AND LF
$CRLF
CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
BR 2$ ;;GET NEXT CHARACTER
5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
BNE 2$ ;;IF NO GO GET NEXT CHAR.
MOV $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
;;AND THE NULL CHAR.
7$: DECB 1(SP) ;;DOES A NULL NEED TO BE TYPED?
BLT 6$ ;;BR IF NO--GO POP THE NULL OFF OF STACK
JSR PC,$TYPEC ;;GO TYPE A NULL

```



```

3878 016414 105337 016564          DECB  $CHARCNT      ;;DO NOT COUNT AS A COUNT
3879 016420 000770                    BR    7$           ;;LOOP
3880
3881 ;HORIZONTAL TAB PROCESSOR
3882
3883 016422 112716 000040          8$:  MOVB  #' (SP)      ;;REPLACE TAB WITH SPACE
3884 016426 004737 016446          9$:  JSR   PC,$TYPEC    ;;TYPE A SPACE
3885 016432 132737 000007 016564  BITB  #7,$CHARCNT    ;;BRANCH IF NOT AT
3886 016440 001372                    BNE   9$           ;;TAB STOP
3887 016442 005726                    TST  (SP)+        ;;POP SPACE OFF STACK
3888 016444 000724                    BR    2$           ;;GET NEXT CHARACTER
3889 016446
3890 016446 105777 162372          $TYPEC: TSTB  @$TKS      ;;CHAR IN KYBD BUFFER?
3891 016452 100022                    BPL  10$          ;;BR IF NOT
3892 016454 017746 162366          MOV   @$TKB,-(SP)    ;;GET CHAR
3893 016460 042716 177600          BIC   #177600,(SP)  ;;STRIP EXTRANEIOUS BITS
3894 016464 122716 000023          CMPB  #$XOFF,(SP)   ;;WAS CHAR XOFF
3895 016470 001012                    BNE  102$         ;;BR IF NOT
3896 016472
3897 016472 105777 162346          101$: TSTB  @$TKS      ;;WAIT FOR CHAR
3898 016476 100375                    BPL  101$
3899 016500 117716 162342          MOVB  @$TKB,(SP)    ;;GET CHAR
3900 016504 042716 177600          BIC   #177600,(SP)  ;;STRIP IT
3901 016510 122716 000021          CMPB  #$XON,(SP)   ;;WAS IT XON?
3902 016514 001366                    BNE  101$         ;;BR IF NOT
3903 016516
3904 016516 005726                    102$: TST  (SP)+    ;;FIX STACK
3905 016520
3906 016520 105777 162324          10$:  TSTB  @$TPS      ;;WAIT UNTIL PRINTER IS READY
3907 016524 100375                    BPL  10$
3908 016526 116677 000002 162316  MOVB  2(SP),@$TPB   ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3909 016534 122766 000015 000002  CMPB  #CR,2(SP)     ;;IS CHARACTER A CARRIAGE RETURN?
3910 016542 001003                    BNE  1$           ;;BRANCH IF NO
3911 016544 105037 016564          CLRB  $CHARCNT     ;;YES--CLEAR CHARACTER COUNT
3912 016550 000406                    BR   $TYPEX       ;;EXIT
3913 016552 122766 000012 000002  1$:  CMPB  #LF,2(SP)  ;;IS CHARACTER A LINE FEED?
3914 016560 001402                    BEQ  $TYPEX       ;;BRANCH IF YES
3915 016562 105227                    INCB (PC)+        ;;COUNT THE CHARACTER
3916 016564 000000          $CHARCNT: .WORD  0 ;;CHARACTER COUNT STORAGE
3917 016566 000207          $TYPEX: RTS   PC
3918
3919 ;SBTTL BINARY TO OCTAL (ASCII) AND TYPE
3920
3921 ;*****
3922 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3923 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3924 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3925 ;*CALL:
3926 ;*   MOV   NUM,-(SP)      ;;NUMBER TO BE TYPED
3927 ;*   TYPOS      ;;CALL FOR TYPEOUT
3928 ;*   .BYTE  N           ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3929 ;*   .BYTE  M           ;;M=1 OR 0
3930 ;*                               ;;1=TYPE LEADING ZEROS
3931 ;*                               ;;0=SUPPRESS LEADING ZEROS
3932 ;*
3933 ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```



```

3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944 016570 017646 000000
3945 016574 116637 000001 017013
3946 016602 112637 017015
3947 016606 062716 000002
3948 016612 000406
3949 016614 112737 000001 017013
3950 016622 112737 000006 017015
3951 016630 112737 000005 017012
3952 016636 010346
3953 016640 010446
3954 016642 010546
3955 016644 113704 017015
3956 016650 005404
3957 016652 062704 000006
3958 016656 110437 017014
3959 016662 113704 017013
3960 016666 016605 000012
3961 016672 005003
3962 016674 006105 1$:
3963 016676 000404 BR
3964 016700 006105 2$:
3965 016702 006105 ROL
3966 016704 006105 ROL
3967 016706 010503 MOV
3968 016710 006103 3$:
3969 016712 105337 017014 DECB
3970 016716 100016 BPL
3971 016720 042703 177770 BIC
3972 016724 001002 BNE
3973 016726 005704 TST
3974 016730 001403 BEQ
3975 016732 005204 4$:
3976 016734 052703 000060 BIS
3977 016740 052703 000040 5$:
3978 016744 110337 017010 MOV
3979 016750 104401 017010 TYPE
3980 016754 105337 017012 7$:
3981 016760 003347 BGT
3982 016762 002402 BLT
3983 016764 005204 INC
3984 016766 000744 BR
3985 016770 012605 6$:
3986 016772 012604 MOV
3987 016774 012603 MOV
3988 016776 016666 000002- 000004 MOV
3989 017004 012616 MOV

;*$TYPOS OR $TYPOC
;*$CALL:
;* MOV NUM,-(SP) ;:NUMBER TO BE TYPED
;* TYPON ;:CALL FOR TYPEOUT
;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;*$CALL:
;* MOV NUM,-(SP) ;:NUMBER TO BE TYPED
;* TYPOC ;:CALL FOR TYPEOUT
$TYPOS: MOV @(SP),-(SP) ;:PICKUP THE MODE
MOV 1(SP),$OFILL ;:LOAD ZERO FILL SWITCH
MOV (SP)+,$SOMODE+1 ;:NUMBER OF DIGITS TO TYPE
ADD #2,(SP) ;:ADJUST RETURN ADDRESS
BR $TYPON
$TYPOC: MOV #1,$OFILL ;:SET THE ZERO FILL SWITCH
MOV #6,$SOMODE+1 ;:SET FOR SIX(6) DIGITS
$TYPON: MOV #5,$OCNT ;:SET THE ITERATION COUNT
MOV R3,-(SP) ;:SAVE R3
MOV R4,-(SP) ;:SAVE R4
MOV R5,-(SP) ;:SAVE R5
MOV $SOMODE+1,R4 ;:GET THE NUMBER OF DIGITS TO TYPE
NEG R4
ADD #6,R4 ;:SUBTRACT IT FOR MAX. ALLOWED
MOV R4,$SOMODE ;:SAVE IT FOR USE
MOV $OFILL,R4 ;:GET THE ZERO FILL SWITCH
MOV 12(SP),R5 ;:PICKUP THE INPUT NUMBER
CLR R3 ;:CLEAR THE OUTPUT WORD
ROL R5 ;:ROTATE MSB INTO 'C'
BR 3$ ;:GO DO MSB
ROL R5 ;:FORM THIS DIGIT
MOV R5,R3
ROL R3 ;:GET LSB OF THIS DIGIT
DECB $SOMODE ;:TYPE THIS DIGIT?
BPL 7$ ;:BR IF NO
BIC #177770,R3 ;:GET RID OF JUNK
BNE 4$ ;:TEST FOR 0
TST R4 ;:SUPPRESS THIS 0?
BEQ 5$ ;:BR IF YES
INC R4 ;:DON'T SUPPRESS ANYMORE 0'S
BIS #'0,R3 ;:MAKE THIS DIGIT ASCII
BIS #' ,R3 ;:MAKE ASCII IF NOT ALREADY
MOV R3,8$ ;:SAVE FOR TYPING
TYPE 8$ ;:GO TYPE THIS DIGIT
DECB $OCNT ;:COUNT BY 1
BGT 2$ ;:BR IF MORE TO DO
BLT 6$ ;:BR IF DONE
INC R4 ;:INSURE LAST DIGIT ISN'T A BLANK
BR 2$ ;:GO DO THE LAST DIGIT
MOV (SP)+,R5 ;:RESTORE R5
MOV (SP)+,R4 ;:RESTORE R4
MOV (SP)+,R3 ;:RESTORE R3
MOV 2(SP),4(SP) ;:SET THE STACK FOR RETURNING
MOV (SP)+,(SP)

```


4046 017160 100003
4047 017162 116663 177777 177776
4048 017170 105013
4049 017172 012605
4050 017174 012603
4051 017176 012602
4052 017200 012601
4053 017202 012600
4054 017204 104401 017232
4055 017210 016666 000002 000004
4056 017216 012616
4057 017220 000002
4058 017222 023420
4059 017224 001750
4060 017226 000144
4061 017230 000012
4062 017232 000004
4063

```
9$: BPL 9$          ;;BR IF NO
    MOVB -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
    CLRB (R3)        ;;SET THE TERMINATOR
    MOV (SP)+,R5     ;;POP STACK INTO R5
    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
    TYPE $DBLK      ;;NOW TYPE THE NUMBER
    MOV 2(SP),4(SP) ;;ADJUST THE STACK
    MOV (SP)+,(SP)
    RTI            ;;RETURN TO USER

$DTBL: 10000.
        1000.
        100.
        10.

$DBLK: .BLKW 4
```



```
4064  
4065  
4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075 017242 022737 000176 001040 $CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?  
4076 017250 001074 BNE 15$ ;;BRANCH IF NO  
4077 017252 105777 161566 TSTB @$TKS ;;CHAR THERE?  
4078 017256 100071 BPL 15$ ;;IF NO, DON'T WAIT AROUND  
4079 017260 117746 161562 MOVB @$TKB,-(SP) ;;SAVE THE CHAR  
4080 017264 042716 177600 BIC #^C177,(SP) ;;STRIP-OFF THE ASCII  
4081 017270 022726 000007 CMP #7,(SP)+ ;;IS IT A CONTROL G?  
4082 017274 001062 BNE 15$ ;;NO, RETURN TO USER  
4083 017276 123727 001034 000001 CMPB $AUTOB,#1 ;;ARE WE RUNNING IN AUTO-MODE?  
4084 017304 001456 BEQ 15$ ;;BRANCH IF YES  
4085  
4086 017306 104401 017777 $GTSWR: TYPE , $CNTLG ;;ECHO THE CONTROL-G (^G)  
4087 017312 104401 020004 TYPE , $MSWR ;;TYPE CURRENT CONTENTS  
4088 017316 013746 000176 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT  
4089 017322 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)  
4090 017324 104401 020015 TYPE , $MNEW ;;PROMPT FOR NEW SWR  
4091 017330 005046 19$: CLR -(SP) ;;CLEAR COUNTER  
4092 017332 005046 CLR -(SP) ;;THE NEW SWR  
4093 017334 105777 161504 7$: TSTB @$TKS ;;CHAR THERE?  
4094 017340 100375 BPL 7$ ;;IF NOT TRY AGAIN  
4095  
4096 017342 117746 161500 MOVB @$TKB,-(SP) ;;PICK UP CHAR  
4097 017346 042716 177600 BIC #^C177,(SP) ;;MAKE IT 7-BIT ASCII  
4098  
4099  
4100  
4101 017352 021627 000025 9$: CMP (SP),#25 ;;IS IT A CONTROL-U?  
4102 017356 001005 BNE 10$ ;;BRANCH IF NOT  
4103 017360 104401 017772 TYPE , $CNTLU ;;YES, ECHO CONTROL-U (^U)  
4104 017364 062706 000006 20$: ADD #6,SP ;;IGNORE PREVIOUS INPUT  
4105 017370 000757 BR 19$ ;;LET'S TRY IT AGAIN  
4106  
4107  
4108 017372 021627 000015 10$: CMP (SP),#15 ;;IS IT A <CR>?  
4109 017376 001022 BNE 16$ ;;BRANCH IF NO  
4110 017400 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?  
4111 017404 001403 BEQ 11$ ;;BRANCH IF YES  
4112 017406 016677 000002 161424 MOV 2(SP),@SWR ;;SAVE NEW SWR  
4113 017414 062706 000006 11$: ADD #6,SP ;;CLEAR UP STACK  
4114 017420 104401 001075 14$: TYPE , $CRLF ;;ECHO <CR> AND <LF>  
4115 017424 123727 001035 000001 CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?  
4116 017432 001003 BNE 15$ ;;BRANCH IF NOT  
4117 017434 012777 000100 161402 MOV #100,@$TKS ;;RE-ENABLE TTY KBD INTERRUPTS  
4118 017442 000002 15$: RTI ;;RETURN  
4119 017444 004737 016446 16$: JSR PC,$TYPEC ;;ECHO CHAR
```


4120 017450 021627 000060
4121 017454 002420
4122 017456 021627 000067
4123 017462 003015
4124 017464 042726 000060
4125 017470 005766 000002
4126 017474 001403
4127 017476 006316
4128 017500 006316
4129 017502 006316
4130 017504 005266 000002
4131 017510 056616 177776
4132 017514 000707
4133 017516 104401 001074
4134 017522 000720
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146 017524 011646
4147 017526 016666 000004 000002
4148 017534 105777 161304
4149 017540 100375
4150 017542 117766 161300 000004
4151 017550 042766 177600 000004
4152 017556 026627 000004 000023
4153 017564 001013
4154 017566 105777 161252
4155 017572 100375
4156 017574 117746 161246
4157 017600 042716 177600
4158 017604 022627 000021
4159 017610 001366
4160 017612 000750
4161 017614 026627 000004 000021
4162 017622 001744
4163 017624 026627 000004 000140
4164 017632 002407
4165 017634 026627 000004 000175
4166 017642 003003
4167 017644 042766 000040 000004
4168 017652 000002
4169
4170
4171
4172
4173
4174
4175

```
      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  LSB
```

```
*****
*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
*
```

```
$RDCHR: MOV      (SP),-(SP)   ::PUSH DOWN THE PC
      MOV      4(SP),2(SP)   ::SAVE THE PS
1$:   TSTB     @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$:   TSTB     @STKS         ::WAIT FOR A CHARACTER
      BPL      2$           ::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?
      BEQ      1$           ::BRANCH IF YES
      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
```

```
*****
*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
*
```


4176	017654	010346			\$RDLIN: MOV	R3,-(SP)	::SAVE R3
4177	017656	012703	017762		1\$: MOV	#\$TTYIN,R3	::GET ADDRESS
4178	017662	022703	017772		2\$: CMP	#\$TTYIN+8.,R3	::BUFFER FULL?
4179	017666	101405			BLOS	4\$::BR IF YES
4180	017670	104410			RDCHR		::GO READ ONE CHARACTER FROM THE TTY
4181	017672	112613			MOVB	(SP)+,(R3)	::GET CHARACTER
4182	017674	122713	000177		10\$: CMPB	#177,(R3)	::IS IT A RUBOUT
4183	017700	001003			BNE	3\$::SKIP IF NOT
4184	017702	104401	001074		4\$: TYPE	,\$QUES	::TYPE A '?'
4185	017706	000763			BR	1\$::CLEAR THE BUFFER AND LOOP
4186	017710	111337	017760		3\$: MOVB	(R3),9\$::ECHO THE CHARACTER
4187	017714	104401	017760		TYPE	,\$	
4188	017720	122723	000015		CMPB	#15,(R3)+	::CHECK FOR RETURN
4189	017724	001356			BNE	2\$::LOOP IF NOT RETURN
4190	017726	105063	177777		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
4191	017732	104401	001076		TYPE	,\$LF	::TYPE A LINE FEED
4192	017736	012603			MOV	(SP)+,R3	::RESTORE R3
4193	017740	011646			MOV	(SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
4194	017742	016666	000004	000002	MOV	4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
4195	017750	012766	017762	000004	MOV	#\$TTYIN,4(SP)	
4196	017756	000002			RTI		::RETURN
4197	017760	000			9\$: .BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
4198	017761	000			.BYTE	0	::TERMINATOR
4199	017762	000010			\$TTYIN: .BLKB	8.	::RESERVE 8 BYTES FOR TTY INPUT
4200	017772	052536	005015	000	\$CNTLU: .ASCIZ	/'^U/<15><12>	::CONTROL 'U'
4201	017777	136	006507	000012	\$CNTLG: .ASCIZ	/'^G/<15><12>	::CONTROL 'G'
4202	020004	005015	053523	020122	\$MSWR: .ASCIZ	<15><12>/SWR = /	
4203	020012	020075	000				
4204	020015	040	047040	053505	\$MNEW: .ASCIZ	/ NEW = /	
4205	020022	036440	000040				
4206							

4207
 4208
 4209
 4210
 4211
 4212
 4213
 4214
 4215
 4216
 4217 020026 010046
 4218 020030 016600 000002
 4219 020034 005740
 4220 020036 111000
 4221 020040 006300
 4222 020042 016000 020062
 4223 020046 000200
 4224
 4225
 4226
 4227
 4228 020050 011646
 4229 020052 016666 000004 000002
 4230 020060 000002
 4231
 4232
 4233
 4234
 4235
 4236
 4237
 4238
 4239 020062 020050
 4240 020064 016234
 4241 020066 016614
 4242 020070 016570
 4243 020072 016630
 4244 020074 017016
 4245
 4246 020076 017312
 4247
 4248 020100 017242
 4249 020102 017524
 4250 020104 017654
 4251

.SBTTL TRAP DECODER

 *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
 *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 *GO TO THAT ROUTINE.

```
$TRAP:  MOV    R0,-(SP)      ;;SAVE R0
        MOV    2(SP),R0    ;;GET TRAP ADDRESS
        TST   -(R0)       ;;BACKUP BY 2
        MOVB  (R0),R0      ;;GET RIGHT BYTE OF TRAP
        ASL   R0           ;;POSITION FOR INDEXING
        MOV   $TRPAD(R0),R0 ;;INDEX TO TABLE
        RTS   R0           ;;GO TO ROUTINE
```

;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO

```
$TRAP2: MOV   (SP),-(SP)   ;;MOVE THE PC DOWN
        MOV   4(SP),2(SP)  ;;MOVE THE PSW DOWN
        RTI                    ;;RESTORE THE PSW
```

.SBTTL TRAP TABLE

*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 *BY THE 'TRAP' INSTRUCTION.

```

:      ROUTINE
:      -----
$TRPAD: .WORD  $TRAP2
        $TYPE  ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
        $TYPOC ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

        $GTSWR ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

        $CKSWR ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
        $RDCHR ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
        $RDLIN ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
```


4252
4253
4254
4255
4256
4257
4258
4259 020106 012737 000176 001040
4260 020114 012737 000174 001042
4261 020122 032777 000020 160710
4262 020130 001403
4263 020132 012703 002770
4264 020136 000402
4265 020140 012703 003000
4266 020144 000005
4267 020146 112773 000052 000006
4268 020154 105773 000000
4269 020160 100375
4270 020162 117373 000002 000006
4271 020170 017300 000002
4272 020174 100023
4273 020176 052701 010000
4274 020202 030100
4275 020204 001403
4276 020206 004737 020270
4277 020212 020316
4278 020214 006301
4279 020216 030100
4280 020220 001403
4281 020222 004737 020270
4282 020226 020327
4283 020230 006301
4284 020232 030100
4285 020234 001403
4286 020236 004737 020270
4287 020242 020341
4288 020244 042700 000200
4289 020250 022700 000003
4290 020254 001337
4291 020256 004737 020270
4292 020262 020354
4293 020264 000000
4294 020266 000707
4295
4296
4297 020270 013600
4298 020272 062746 000002
4299 020276 105773 000004
4300 020302 100375
4301 020304 112073 000006
4302 020310 105710
4303 020312 001371
4304 020314 000207
4305
4306 020316 005015 040520 044522
4307 020324 054524 000

```

.SBTTL 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
*****
ECHO:  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$:   MOV    #RCRCSR,R3            ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
2$:   RESET                          ;CLEAR EVERYTHING
      MOV    #'*,@6(R3)           ;TRANSMIT PROMPT '*'
3$:   TSTB   @R3                   ;WAIT FOR INPUT
      BPL    3$
      MOV    @2(R3),@6(R3)        ;ECHO INPUT
      MOV    @2(R3),R0            ;STORE INPUT
      BPL    6$                   ;BR IF 'ERROR' NOT SET
      BIS    #BIT12,R1           ;SET PARITY ERROR TEST MASK
      BIT    R1,R0                ;CHECK FOR PARITY ERROR FLAG
      BEQ    4$                   ;BR IF NOT SET
      JSR    PC,MSG              ;REPORT PARITY ERROR
4$:   ASL    R1                     ;SHIFT MASK TO TEST 'FR' FLAG
      BIT    R1,R0                ;TEST FOR FRAMING ERROR FLAG
      BEQ    5$                   ;BR IF NOT SET
      JSR    PC,MSG              ;REPSORT FRAMING ERROR
5$:   ASL    R1                     ;SHIFT MASK TO TEST 'OR' FLAG
      BIT    R1,R0                ;TEST FOR OVERFLOW ERROR
      BEQ    6$                   ;BR IF NOT SET
      JSR    PC,MSG              ;REPORT OVERFLOW ERROR
6$:   BIC    #BIT7,R0              ;CLEAR ANY PARITY BIT
      CMP    #3,R0                ;WAS INPUT CONTROL-C
      BNE    3$                   ;BR IF IS NOT
      JSR    PC,MSG              ;REPORT PROGRAM STOP
      MSTOP
      HALT
      BR     ECHO
      ; END OF TEST HALT
      ; AFTER END OF TEST HALT
      ; PRESS CONTINUE TO RESTART ECHO TEST

MSG:  MOV    @(SP)+,R0            ;PICK UP MESSAGE POINTER
      ADD    #2,-(SP)             ;ADJUST RETURN PC
WAIT: TSTB   @4(R3)              ;WAIT FOR XMIT DONE
      BPL    WAIT
      MOV    (R0)+,@6(R3)        ;SEND CHARACTER
      TSTB   (R0)                ;IS THIS END OF MESSAGE?
      BNE    WAIT                ;BR IF NOT
      RTS    PC                  ;RETURN

MPAR: .ASCIZ <CR><LF>/PARITY/

```


CJKDFAO 11/24 OPTIONS DIAGNOSTIC
CJKDFA.P11 19-MAR-81 14:27

MACY11 30A(1052) 19-MAR-81^{K 7} 14:27 PAGE 88
ECHO TEST

SEQ 0088

4308	020327	015	043012	040522	MFR:	.ASCIZ	<CR><LF>/FRAMING/
4309	020334	044515	043516	000			
4310	020341	015	047412	042526	MOR:	.ASCIZ	<CR><LF>/OVERFLOW/
4311	020346	043122	047514	000127			
4312	020354	005015	052123	050117	MSTOP:	.ASCIZ	<CR><LF>/STOP/
4313	020362	000					

4356						
4357						
4358	020510	046123	030525	052040	EM5:	.ASCIZ /SLU1 TCSR DONE NOT SET WITH RESET/
4359	020516	051503	020122	047504		
4360	020524	042516	047040	052117		
4361	020532	051440	052105	053440		
4362	020540	052111	020110	042522		
4363	020546	042523	000124			
4364	020552	046123	030525	051040	EM6:	.ASCIZ /SLU1 RCSR DID NOT RETURN SSYNC/
4365	020560	051503	020122	044504		
4366	020566	020104	047516	020124		
4367	020574	042522	052524	047122		
4368	020602	051440	054523	041516		
4369	020610	000				
4370	020611	123	052514	020061	EM7:	.ASCIZ /SLU1 RBUF DID NOT RETURN SSYNC/
4371	020616	041122	043125	042040		
4372	020624	042111	047040	052117		
4373	020632	051040	052105	051125		
4374	020640	020116	051523	047131		
4375	020646	000103				
4376	020650	040503	020116	047516	EM11:	.ASCIZ /CAN NOT SET BIT2 OF SLU1 TCSR/
4377	020656	020124	042523	020124		
4378	020664	044502	031124	047440		
4379	020672	020106	046123	030525		
4380	020700	052040	051503	000122		
4381	020706	042522	042523	020124	EM13:	.ASCIZ /RESET DID NOT CLEAR BIT2 OF SLU1 TCSR/
4382	020714	044504	020104	047516		
4383	020722	020124	046103	040505		
4384	020730	020122	044502	031124		
4385	020736	047440	020106	046123		
4386	020744	030525	052040	051503		
4387	020752	000122				
4388	020754	044502	033124	047440	EM14:	.ASCIZ /BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET/
4389	020762	020106	046123	030525		
4390	020770	052040	051503	020122		
4391	020776	047516	020124	046103		
4392	021004	040505	020122	043101		
4393	021012	042524	020122	042522		
4394	021020	042523	000124			
4395	021024	046123	030525	054040	EM15:	.ASCIZ /SLU1 XMIT INTERRUPTED WITH PRIORITY 7/
4396	021032	044515	020124	047111		
4397	021040	042524	051122	050125		
4398	021046	042524	020104	044527		
4399	021054	044124	050040	044522		
4400	021062	051117	052111	020131		
4401	021070	000067				
4402	021072	040503	020116	047516	EM16:	.ASCIZ /CAN NOT SET BIT6 OF SLU1 TCSR/
4403	021100	020124	042523	020124		
4404	021106	044502	033124	047440		
4405	021114	020106	046123	030525		
4406	021122	052040	051503	000122		
4407	021130	040503	020116	047516	EM17:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU1 TCSR/
4408	021136	020124	046103	040505		
4409	021144	020122	044502	033124		
4410	021152	047440	020106	046123		
4411	021160	030525	052040	051503		

4412	021166	000122			
4413	021170	042522	042523	020124	EM20: .ASCIZ /RESET DID NOT CLEAR BIT6 OF SLU1 TCSR/
4414	021176	044504	020104	047516	
4415	021204	020124	046103	040505	
4416	021212	020122	044502	033124	
4417	021220	047440	020106	046123	
4418	021226	030525	052040	051503	
4419	021234	000122			
4420	021236	044502	033124	047440	EM21: .ASCIZ /BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET/
4421	021244	020106	046123	030525	
4422	021252	051040	051503	020122	
4423	021260	047516	020124	046103	
4424	021266	040505	020122	043101	
4425	021274	042524	020122	042522	
4426	021302	042523	000124		
4427	021306	046123	030525	051040	EM22: .ASCIZ /SLU1 RCVR INTERRUPT WITH PRIORITY 7/
4428	021314	053103	020122	047111	
4429	021322	042524	051122	050125	
4430	021330	020124	044527	044124	
4431	021336	050040	044522	051117	
4432	021344	052111	020131	000067	
4433	021352	040503	020116	047516	EM23: .ASCIZ /CAN NOT SET BIT6 OF SLU1 RCSR/
4434	021360	020124	042523	020124	
4435	021366	044502	033124	047440	
4436	021374	020106	046123	030525	
4437	021402	051040	051503	000122	
4438	021410	040503	020116	047516	EM24: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU1 RCSR/
4439	021416	020124	046103	040505	
4440	021424	020122	044502	033124	
4441	021432	047440	020106	046123	
4442	021440	030525	051040	051503	
4443	021446	000122			
4444	021450	040503	020116	047516	EM25: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET/
4445	021456	020124	046103	040505	
4446	021464	020122	044502	033124	
4447	021472	047440	020106	046123	
4448	021500	030525	051040	051503	
4449	021506	020122	044527	044124	
4450	021514	051040	051505	052105	
4451	021522	000			
4452	021523	123	052514	020061	EM26: .ASCIZ /SLU1 RECEIVER DONE NEVER SET/
4453	021530	042522	042503	053111	
4454	021536	051105	042040	047117	
4455	021544	020105	042516	042526	
4456	021552	020122	042523	000124	
4457	021560	042522	042523	020124	EM27: .ASCIZ /RESET DID NOT CLEAR SLU1 RCVR DONE/
4458	021566	044504	020104	047516	
4459	021574	020124	046103	040505	
4460	021602	020122	046123	030525	
4461	021610	051040	053103	020122	
4462	021616	047504	042516	000	
4463	021623	122	040505	044504	EM30: .ASCIZ /READING SLU1 RBUF DID NOT CLEAR RCVR DONE/
4464	021630	043516	051440	052514	
4465	021636	020061	041122	043125	
4466	021644	042040	042111	047040	
4467	021652	052117	041440	042514	

4468	021660	051101	051040	053103	
4469	021666	020122	047504	042516	
4470	021674	000			
4471	021675	123	052514	020062	EM31: .ASCIZ /SLU2 TCSR DID NOT RETURN SSYNC/
4472	021702	041524	051123	042040	
4473	021710	042111	047040	052117	
4474	021716	051040	052105	051125	
4475	021724	020116	051523	047131	
4476	021732	000103			
4477	021734	046123	031125	052040	EM32: .ASCIZ /SLU2 TBUF DID NOT RETURN SSYNC/
4478	021742	052502	020106	044504	
4479	021750	020104	047516	020124	
4480	021756	042522	052524	047122	
4481	021764	051440	054523	041516	
4482	021772	000			
4483	021773	123	052514	020062	EM33: .ASCIZ /SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL/
4484	022000	041524	051123	042040	
4485	022006	047117	020105	047516	
4486	022014	020124	046103	040505	
4487	022022	042522	020104	044527	
4488	022030	044124	052040	052502	
4489	022036	020106	052506	046114	
4490	022044	000			
4491	022045	123	052514	020062	EM34: .ASCIZ /SLU2 TCSR DONE NOT SET AFTER TRANSMIT/
4492	022052	041524	051123	042040	
4493	022060	047117	020105	047516	
4494	022066	020124	042523	020124	
4495	022074	043101	042524	020122	
4496	022102	051124	047101	046523	
4497	022110	052111	000		
4498	022113	123	052514	020062	EM35: .ASCIZ /SLU2 TCSR DONE NOT SET WITH RESET/
4499	022120	041524	051123	042040	
4500	022126	047117	020105	047516	
4501	022134	020124	042523	020124	
4502	022142	044527	044124	051040	
4503	022150	051505	052105	000	
4504	022155	123	052514	020062	EM36: .ASCIZ /SLU2 RCSR DID NOT RETURN SSYNC/
4505	022162	041522	051123	042040	
4506	022170	042111	047040	052117	
4507	022176	051040	052105	051125	
4508	022204	020116	051523	047131	
4509	022212	000103			
4510	022214	046123	031125	051040	EM37: .ASCIZ /SLU2 RBUF DID NOT RETURN SSYNC/
4511	022222	052502	020106	044504	
4512	022230	020104	047516	020124	
4513	022236	042522	052524	047122	
4514	022244	051440	054523	041516	
4515	022252	000			
4516	022253	102	052111	020060	EM40: .ASCIZ /BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET/
4517	022260	043117	051440	052514	
4518	022266	020062	041524	051123	
4519	022274	047040	052117	041440	
4520	022302	042514	051101	040440	
4521	022310	052106	051105	051040	
4522	022316	051505	052105	000	
4523	022323	103	047101	047040	EM41: .ASCIZ /CAN NOT SET BIT0 OF SLU2 TCSR/

4524	022330	052117	051440	052105	
4525	022336	041040	052111	020060	
4526	022344	043117	051440	052514	
4527	022352	020062	041524	051123	
4528	022360	000			
4529	022361	103	047101	047040	EM42: .ASCIZ /CAN NOT CLEAR BIT0 OF SLU2 TCSR/
4530	022366	052117	041440	042514	
4531	022374	051101	041040	052111	
4532	022402	020060	043117	051440	
4533	022410	052514	020062	041524	
4534	022416	051123	000		
4535	022421	122	051505	052105	EM43: .ASCIZ /RESET DID NOT CLEAR BIT0 OF SLU2 TCSR/
4536	022426	042040	042111	047040	
4537	022434	052117	041440	042514	
4538	022442	051101	041040	052111	
4539	022450	020060	043117	051440	
4540	022456	052514	020062	041524	
4541	022464	051123	000		
4542	022467	102	052111	020066	EM44: .ASCIZ /BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET/
4543	022474	043117	051440	052514	
4544	022502	020062	041524	051123	
4545	022510	047040	052117	041440	
4546	022516	042514	051101	040440	
4547	022524	052106	051105	051040	
4548	022532	051505	052105	000	
4549	022537	123	052514	020062	EM45: .ASCIZ /SLU2 XMIT INTERRUPTED WITH PRIORITY 7/
4550	022544	046530	052111	044440	
4551	022552	052116	051105	052522	
4552	022560	052120	042105	053440	
4553	022566	052111	020110	051120	
4554	022574	047511	044522	054524	
4555	022602	033440	000		
4556	022605	103	047101	047040	EM46: .ASCIZ /CAN NOT SET BIT6 OF SLU2 TCSR/
4557	022612	052117	051440	052105	
4558	022620	041040	052111	020066	
4559	022626	043117	051440	052514	
4560	022634	020062	041524	051123	
4561	022642	000			
4562	022643	103	047101	047040	EM47: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 TCSR/
4563	022650	052117	041440	042514	
4564	022656	051101	041040	052111	
4565	022664	020066	043117	051440	
4566	022672	052514	020062	041524	
4567	022700	051123	000		
4568	022703	122	051505	052105	EM50: .ASCIZ /RESET DID NOT CLEAR BIT6 OF SLU2 TCSR/
4569	022710	042040	042111	047040	
4570	022716	052117	041440	042514	
4571	022724	051101	041040	052111	
4572	022732	020066	043117	051440	
4573	022740	052514	020062	041524	
4574	022746	051123	000		
4575	022751	102	052111	020066	EM51: .ASCIZ /BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET/
4576	022756	043117	051440	052514	
4577	022764	020062	041522	051123	
4578	022772	047040	052117	041440	
4579	023000	042514	051101	040440	

4580	023006	052106	051105	051040	
4581	023014	051505	052105	000	
4582	023021	123	052514	020062	EM52: .ASCIZ /SLU2 RCVR INTERRUPT WITH PRIORITY 7/
4583	023026	041522	051126	044440	
4584	023034	052116	051105	052522	
4585	023042	052120	053440	052111	
4586	023050	020110	051120	047511	
4587	023056	044522	054524	033440	
4588	023064	000			
4589	023065	103	047101	047040	EM53: .ASCIZ /CAN NOT SET BIT6 OF SLU2 RCSR/
4590	023072	052117	051440	052105	
4591	023100	041040	052111	020066	
4592	023106	043117	051440	052514	
4593	023114	020062	041522	051123	
4594	023122	000			
4595	023123	103	047101	047040	EM54: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR/
4596	023130	052117	041440	042514	
4597	023136	051101	041040	052111	
4598	023144	020066	043117	051440	
4599	023152	052514	020062	041522	
4600	023160	051123	000		
4601	023163	103	047101	047040	EM55: .ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET/
4602	023170	052117	041440	042514	
4603	023176	051101	041040	052111	
4604	023204	020066	043117	051440	
4605	023212	052514	020062	041522	
4606	023220	051123	053440	052111	
4607	023226	020110	042522	042523	
4608	023234	000124			
4609	023236	046123	031125	051040	EM56: .ASCIZ /SLU2 RECEIVER DONE NEVER SET/
4610	023244	041505	044505	042526	
4611	023252	020122	047504	042516	
4612	023260	047040	053105	051105	
4613	023266	051440	052105	000	
4614	023273	122	051505	052105	EM57: .ASCIZ /RESET DID NOT CLEAR SLU2 RCVR DONE/
4615	023300	042040	042111	047040	
4616	023306	052117	041440	042514	
4617	023314	051101	051440	052514	
4618	023322	020062	041522	051126	
4619	023330	042040	047117	000105	
4620	023336	042522	042101	047111	EM60: .ASCIZ /READING SLU2 RBUF DID NOT CLEAR RCVR DONE/
4621	023344	020107	046123	031125	
4622	023352	051040	052502	020106	
4623	023360	044504	020104	047516	
4624	023366	020124	046103	040505	
4625	023374	020122	041522	051126	
4626	023402	042040	047117	000105	
4627	023410	045514	020123	044504	EM61: .ASCIZ /LKS DID NOT RETURN SSYNC/
4628	023416	020104	047516	020124	
4629	023424	042522	052524	047122	
4630	023432	051440	054523	041516	
4631	023440	000			
4632	023441	102	052111	020066	EM62: .ASCIZ /BIT6 OF LKS NOT CLEAR AFTER RESET/
4633	023446	043117	046040	051513	
4634	023454	047040	052117	041440	
4635	023462	042514	051101	040440	

4636	023470	052106	051105	051040	
4637	023476	051505	052105	000	
4638	023503	114	051513	044440	EM63: .ASCIZ /LKS INTERRUPT WITH PRIORITY 7/
4639	023510	052116	051105	052522	
4640	023516	052120	053440	052111	
4641	023524	020110	051120	047511	
4642	023532	044522	054524	033440	
4643	023540	000			
4644	023541	103	047101	047040	EM64: .ASCIZ /CAN NOT SET BIT6 OF LKS/
4645	023546	052117	051440	052105	
4646	023554	041040	052111	020066	
4647	023562	043117	046040	051513	
4648	023570	000			
4649	023571	103	047101	047040	EM65: .ASCIZ /CAN NOT CLEAR BIT6 OF LKS/
4650	023576	052117	041440	042514	
4651	023604	051101	041040	052111	
4652	023612	020066	043117	046040	
4653	023620	051513	000		
4654	023623	122	051505	052105	EM66: .ASCIZ /RESET DID NOT CLEAR BIT6 OF LKS/
4655	023630	042040	042111	047040	
4656	023636	052117	041440	042514	
4657	023644	051101	041040	052111	
4658	023652	020066	043117	046040	
4659	023660	051513	000		
4660	023663	102	052111	020067	EM67: .ASCIZ /BIT7 OF LKS NOT SET AFTER RESET/
4661	023670	043117	046040	051513	
4662	023676	047040	052117	051440	
4663	023704	052105	040440	052106	
4664	023712	051105	051040	051505	
4665	023720	052105	000		
4666	023723	103	047101	047040	EM70: .ASCIZ /CAN NOT CLEAR BIT7 OF LKS/
4667	023730	052117	041440	042514	
4668	023736	051101	041040	052111	
4669	023744	020067	043117	046040	
4670	023752	051513	000		
4671	023755	102	052111	020067	EM71: .ASCIZ /BIT7 OF LKS DOES NOT SET/
4672	023762	043117	046040	051513	
4673	023770	042040	042517	020123	
4674	023776	047516	020124	042523	
4675	024004	000124			
4676	024006	051127	052111	047111	EM72: .ASCIZ /WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER/
4677	024014	020107	047524	047440	
4678	024022	042516	044440	052116	
4679	024030	051105	040516	020114	
4680	024036	042101	051104	051505	
4681	024044	020123	047515	044504	
4682	024052	044506	042105	040440	
4683	024060	047516	044124	051105	
4684	024066	000			
4685	024067	123	052514	020061	EM74: .ASCIZ /SLU1 XMIT INTERRUPTS WHEN DISABLED/
4686	024074	046530	052111	044440	
4687	024102	052116	051105	052522	
4688	024110	052120	020123	044127	
4689	024116	047105	042040	051511	
4690	024124	041101	042514	000104	
4691	024132	046123	030525	054040	EM75: .ASCIZ /SLU1 XMIT DID NOT INTERRUPT/

4692	024140	044515	020124	044504	
4693	024146	020104	047516	020124	
4694	024154	047111	042524	051122	
4695	024162	050125	000124		
4696	024166	046123	030525	054040	EM76: .ASCIZ /SLU1 XMIT INTERRUPT AT PRIORITY 7/
4697	024174	044515	020124	047111	
4698	024202	042524	051122	050125	
4699	024210	020124	052101	050040	
4700	024216	044522	051117	052111	
4701	024224	020131	000067		
4702	024230	046123	030525	054040	EM77: .ASCIZ /SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR/
4703	024236	044515	020124	047111	
4704	024244	042524	051122	050125	
4705	024252	051524	053440	052111	
4706	024260	020110	047105	041101	
4707	024266	042514	041440	042514	
4708	024274	051101	000		
4709	024277	123	052514	020061	EM100: .ASCIZ /SLU1 XMIT DID NOT INTERRUPT/
4710	024304	046530	052111	042040	
4711	024312	042111	047040	052117	
4712	024320	044440	052116	051105	
4713	024326	052522	052120	000	
4714	024333	123	052514	020061	EM101: .ASCIZ /SLU1 XMIT RE-INTERRUPTED/
4715	024340	046530	052111	051040	
4716	024346	026505	047111	042524	
4717	024354	051122	050125	042524	
4718	024362	000104			
4719	024364	047514	042101	047111	EM102: .ASCIZ /LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT/
4720	024372	020107	046123	030525	
4721	024400	052040	052502	020106	
4722	024406	044504	020104	047516	
4723	024414	020124	046103	040505	
4724	024422	020122	047111	042524	
4725	024430	051122	050125	000124	
4726	024436	046123	030525	051040	EM103: .ASCIZ /SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR/
4727	024444	053103	020122	047111	
4728	024452	042524	051122	050125	
4729	024460	051524	053440	052111	
4730	024466	020110	047105	041101	
4731	024474	042514	041440	042514	
4732	024502	051101	000		
4733	024505	123	052514	020061	EM104: .ASCIZ /SLU1 RCVR DID NOT INTERRUPT/
4734	024512	041522	051126	042040	
4735	024520	042111	047040	052117	
4736	024526	044440	052116	051105	
4737	024534	052522	052120	000	
4738	024541	123	052514	020061	EM105: .ASCIZ /SLU1 RCVR INTERRUPTS AT PRIORITY 7/
4739	024546	041522	051126	044440	
4740	024554	052116	051105	052522	
4741	024562	052120	020123	052101	
4742	024570	050040	044522	051117	
4743	024576	052111	020131	000067	
4744	024604	046123	030525	051040	EM106: .ASCIZ /SLU1 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4745	024612	053103	020122	047111	
4746	024620	042524	051122	050125	
4747	024626	051524	053440	052111	

4748	024634	020110	047111	042524	
4749	024642	051122	050125	020124	
4750	024650	047105	041101	042514	
4751	024656	041440	042514	051101	
4752	024664	000			
4753	024665	123	052514	020061	EM107: .ASCIZ /SLU1 RCVR DID NOT INTERRUPT/
4754	024672	041522	051126	042040	
4755	024700	042111	047040	052117	
4756	024706	044440	052116	051105	
4757	024714	052522	052120	000	
4758	024721	123	052514	020061	EM110: .ASCIZ /SLU1 RECEIVER RE-INTERRUPTED/
4759	024726	042522	042503	053111	
4760	024734	051105	051040	026505	
4761	024742	047111	042524	051122	
4762	024750	050125	042524	000104	
4763	024756	046123	030525	051040	EM111: .ASCIZ /SLU1 READING RBUF DID NOT CLEAR INTERRUPT/
4764	024764	040505	044504	043516	
4765	024772	051040	052502	020106	
4766	025000	044504	020104	047516	
4767	025006	020124	046103	040505	
4768	025014	020122	047111	042524	
4769	025022	051122	050125	000124	
4770	025030	042522	042523	020124	EM112: .ASCIZ /RESET DID NOT CLEAR SLU1 RCVR INTERRUPT/
4771	025036	044504	020104	047516	
4772	025044	020124	046103	040505	
4773	025052	020122	046123	030525	
4774	025060	051040	053103	020122	
4775	025066	047111	042524	051122	
4776	025074	050125	000124		
4777	025100	046123	030525	023440	EM113: .ASCIZ /SLU1 'OR' FLAG DID NOT SET/
4778	025106	051117	020047	046106	
4779	025114	043501	042040	042111	
4780	025122	047040	052117	051440	
4781	025130	052105	000		
4782	025133	123	052514	020061	EM114: .ASCIZ /SLU1 'ERROR' NOT SET WITH 'OR' FLAG/
4783	025140	042447	051122	051117	
4784	025146	020047	047516	020124	
4785	025154	042523	020124	044527	
4786	025162	044124	023440	051117	
4787	025170	020047	046106	043501	
4788	025176	000			
4789	025177	104	052101	020101	EM115: .ASCIZ /DATA COMPARE ERROR/
4790	025204	047503	050115	051101	
4791	025212	020105	051105	047522	
4792	025220	000122			
4793	025222	046123	031125	054040	EM116: .ASCIZ /SLU2 XMIT INTERRUPTS WHEN DISABLED/
4794	025230	044515	020124	047111	
4795	025236	042524	051122	050125	
4796	025244	051524	053440	042510	
4797	025252	020116	044504	040523	
4798	025260	046102	042105	000	
4799	025265	123	052514	020062	EM117: .ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
4800	025272	046530	052111	042040	
4801	025300	042111	047040	052117	
4802	025306	044440	052116	051105	
4803	025314	052522	052120	000	

4804	025321	123	052514	020062	EM120: .ASCIZ /SLU2 XMIT INTERRUPT AT PRIORITY 7/
4805	025326	046530	052111	044440	
4806	025334	052116	051105	052522	
4807	025342	052120	040440	020124	
4808	025350	051120	047511	044522	
4809	025356	054524	033440	000	
4810	025363	123	052514	020062	EM121: .ASCIZ /SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR/
4811	025370	046530	052111	044440	
4812	025376	052116	051105	052522	
4813	025404	052120	020123	044527	
4814	025412	044124	042440	040516	
4815	025420	046102	020105	046103	
4816	025426	040505	000122		
4817	025432	046123	031125	054040	EM122: .ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
4818	025440	044515	020124	044504	
4819	025446	020104	047516	020124	
4820	025454	047111	042524	051122	
4821	025462	050125	000124		
4822	025466	046123	031125	054040	EM123: .ASCIZ /SLU2 XMIT RE-INTERRUPTED/
4823	025474	044515	020124	042522	
4824	025502	044455	052116	051105	
4825	025510	052522	052120	042105	
4826	025516	000			
4827	025517	114	040517	044504	EM124: .ASCIZ /LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT/
4828	025524	043516	051440	052514	
4829	025532	020062	041124	043125	
4830	025540	042040	042111	047040	
4831	025546	052117	041440	042514	
4832	025554	051101	044440	052116	
4833	025562	051105	052522	052120	
4834	025570	000			
4835	025571	123	052514	020062	EM125: .ASCIZ /SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR/
4836	025576	041522	051126	044440	
4837	025604	052116	051105	052522	
4838	025612	052120	020123	044527	
4839	025620	044124	042440	040516	
4840	025626	046102	020105	046103	
4841	025634	040505	000122		
4842	025640	046123	031125	051040	EM126: .ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
4843	025646	053103	020122	044504	
4844	025654	020104	047516	020124	
4845	025662	047111	042524	051122	
4846	025670	050125	000124		
4847	025674	046123	031125	051040	EM127: .ASCIZ /SLU2 RCVR INTERRUPTS AT PRIORITY 7/
4848	025702	053103	020122	047111	
4849	025710	042524	051122	050125	
4850	025716	051524	040440	020124	
4851	025724	051120	047511	044522	
4852	025732	054524	033440	000	
4853	025737	123	052514	020062	EM130: .ASCIZ /SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4854	025744	041522	051126	044440	
4855	025752	052116	051105	052522	
4856	025760	052120	020123	044527	
4857	025766	044124	044440	052116	
4858	025774	051105	052522	052120	
4859	026002	042440	040516	046102	

4860	026010	020105	046103	040505	
4861	026016	000122			
4862	026020	046123	031125	051040	EM131: .ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
4863	026026	053103	020122	044504	
4864	026034	020104	047516	020124	
4865	026042	047111	042524	051122	
4866	026050	050125	000124		
4867	026054	046123	031125	051040	EM132: .ASCIZ /SLU2 RECEIVER RE-INTERRUPTED/
4868	026062	041505	044505	042526	
4869	026070	020122	042522	044455	
4870	026076	052116	051105	052522	
4871	026104	052120	042105	000	
4872	026111	123	052514	020062	EM133: .ASCIZ /SLU2 READING RBUF DID NOT CLEAR INTERRUPT/
4873	026116	042522	042101	047111	
4874	026124	020107	041122	043125	
4875	026132	042040	042111	047040	
4876	026140	052117	041440	042514	
4877	026146	051101	044440	052116	
4878	026154	051105	052522	052120	
4879	026162	000			
4880	026163	122	051505	052105	EM134: .ASCIZ /RESET DID NOT CLEAR SLU2 RCVR INTERRUPT/
4881	026170	042040	042111	047040	
4882	026176	052117	041440	042514	
4883	026204	051101	051440	052514	
4884	026212	020062	041522	051126	
4885	026220	044440	052116	051105	
4886	026226	052522	052120	000	
4887	026233	123	052514	020062	EM135: .ASCIZ /SLU2 'OR' FLAG DID NOT SET/
4888	026240	047447	023522	043040	
4889	026246	040514	020107	044504	
4890	026254	020104	047516	020124	
4891	026262	042523	000124		
4892	026266	046123	031125	023440	EM136: .ASCIZ /SLU2 'ERROR' NOT SET WITH 'OR' FLAG/
4893	026274	051105	047522	023522	
4894	026302	047040	052117	051440	
4895	026310	052105	053440	052111	
4896	026316	020110	047447	023522	
4897	026324	043040	040514	000107	
4898	026332	046123	031125	041040	EM137: .ASCIZ /SLU2 BREAK DID NOT TRANSMIT ALL ZEROES/
4899	026340	042522	045501	042040	
4900	026346	042111	047040	052117	
4901	026354	052040	040522	051516	
4902	026362	044515	020124	046101	
4903	026370	020114	042532	047522	
4904	026376	051505	000		
4905	026401	102	042522	045501	EM140: .ASCIZ /BREAK DID NOT SET FRAMING ERROR/
4906	026406	042040	042111	047040	
4907	026414	052117	051440	052105	
4908	026422	043040	040522	044515	
4909	026430	043516	042440	051122	
4910	026436	051117	000		
4911	026441	123	052514	020062	EM141: .ASCIZ /SLU2 'ERROR' NOT SET WITH 'FR' FLAG/
4912	026446	042447	051122	051117	
4913	026454	020047	047516	020124	
4914	026462	042523	020124	044527	
4915	026470	044124	023440	051106	

4916	026476	020047	046106	043501	
4917	026504	000			
4918	026505	104	052101	020101	EM142: .ASCIZ /DATA COMPARE ERROR WITH CABLE/
4919	026512	047503	050115	051101	
4920	026520	020105	051105	047522	
4921	026526	020122	044527	044124	
4922	026534	041440	041101	042514	
4923	026542	000			
4924	026543	122	041524	044440	EM143: .ASCIZ /RTC INTERRUPT AT PRIORITY 7/
4925	026550	052116	051105	052522	
4926	026556	052120	040440	020124	
4927	026564	051120	047511	044522	
4928	026572	054524	033440	000	
4929	026577	122	041524	044440	EM144: .ASCIZ /RTC INTERRUPTS WHEN DISABLED/
4930	026604	052116	051105	052522	
4931	026612	052120	020123	044127	
4932	026620	047105	042040	051511	
4933	026626	041101	042514	000104	
4934	026634	052122	020103	047111	EM145: .ASCIZ /RTC INTERRUPT DID NOT OCCUR/
4935	026642	042524	051122	050125	
4936	026650	020124	044504	020104	
4937	026656	047516	020124	041517	
4938	026664	052503	000122		
4939	026670	052122	020103	047111	EM146: .ASCIZ /RTC INTERRUPT DID NOT OCCUR/
4940	026676	042524	051122	050125	
4941	026704	020124	044504	020104	
4942	026712	047516	020124	041517	
4943	026720	052503	000122		
4944	026724	052122	020103	047504	EM147: .ASCIZ /RTC DOUBLE INTERRUPT/
4945	026732	041125	042514	044440	
4946	026740	052116	051105	052522	
4947	026746	052120	000		
4948	026751	122	051505	052105	EM150: .ASCIZ /RESET DID NOT CLEAR RTC INTERRUPT/
4949	026756	042040	042111	047040	
4950	026764	052117	041440	042514	
4951	026772	051101	051040	041524	
4952	027000	044440	052116	051105	
4953	027006	052522	052120	000	
4954	027013	122	041524	044440	EM151: .ASCIZ /RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
4955	027020	052116	051105	052522	
4956	027026	052120	042040	042111	
4957	027034	047040	052117	041440	
4958	027042	042514	051101	053440	
4959	027050	052111	020110	044502	
4960	027056	033524	047440	020106	
4961	027064	045514	000123		
4962	027070	046103	041517	020113	EM152: .ASCIZ /CLOCK REPEATABILITY ERROR/
4963	027076	042522	042520	052101	
4964	027104	041101	046111	052111	
4965	027112	020131	051105	047522	
4966	027120	000122			
4967	027122	046123	030525	051040	EM153: .ASCIZ /SLU1 RECEIVER STATUS ERROR/
4968	027130	041505	044505	042526	
4969	027136	020122	052123	052101	
4970	027144	051525	042440	051122	
4971	027152	051117	000		

Line	Code	Address	Value	Code	Value	Label	Text
4972	027155	123	052514	020062		EM154:	.ASCIZ /SLU2 RECEIVER STATUS ERROR/
4973	027162	042522	042503	053111			
4974	027170	051105	051440	040524			
4975	027176	052524	020123	051105			
4976	027204	047522	000122				
4977	027210	047111	047503	051122		EM155:	.ASCIZ /INCORRECT RECEIVE COUNT SLU1/
4978	027216	041505	020124	042522			
4979	027224	042503	053111	020105			
4980	027232	047503	047125	020124			
4981	027240	046123	030525	000			
4982	027245	123	052514	020061		EM156:	.ASCIZ /SLU1 DATA COMPARE ERROR IN EXERCISER/
4983	027252	040504	040524	041440			
4984	027260	046517	040520	042522			
4985	027266	042440	051122	051117			
4986	027274	044440	020116	054105			
4987	027302	051105	044503	042523			
4988	027310	000122					
4989	027312	047111	047503	051122		EM157:	.ASCIZ /INCORRECT RECEIVE COUNT SLU2/
4990	027320	041505	020124	042522			
4991	027326	042503	053111	020105			
4992	027334	047503	047125	020124			
4993	027342	046123	031125	000			
4994	027347	123	052514	020062		EM160:	.ASCIZ /SLU2 DATA COMPARE ERROR IN EXERCISER/
4995	027354	040504	040524	041440			
4996	027362	046517	040520	042522			
4997	027370	042440	051122	051117			
4998	027376	044440	020116	054105			
4999	027404	051105	044503	042523			
5000	027412	000122					
5001	027414	051124	050101	041440		EM161:	.ASCIZ /TRAP CATCHER/
5002	027422	052101	044103	051105			
5003	027430	000					
5004							
5005	027431	124	051505	021524		DH5:	.ASCIZ /TEST# ERR PC TCSR/
5006	027436	020040	042440	051122			
5007	027444	050040	020103	052040			
5008	027452	051503	000122				
5009	027456	042524	052123	020043		DH6:	.ASCIZ /TEST# ERR PC RCSR/
5010	027464	020040	051105	020122			
5011	027472	041520	020040	041522			
5012	027500	051123	000				
5013	027503	124	051505	021524		DH7:	.ASCIZ /TEST# ERR PC RBUF/
5014	027510	020040	042440	051122			
5015	027516	050040	020103	051040			
5016	027524	052502	000106				
5017	027530	042524	052123	020043		DH32:	.ASCIZ /TEST# ERR PC TBUF/
5018	027536	020040	051105	020122			
5019	027544	041520	020040	041124			
5020	027552	043125	000				
5021	027555	124	051505	021524		DH61:	.ASCIZ /TEST# ERR PC LKS/
5022	027562	020040	042440	051122			
5023	027570	050040	020103	046040			
5024	027576	051513	000				
5025	027601	124	051505	021524		DH72:	.ASCIZ /TEST# ERR PC GDADR BDADR GDDAT BDDAT/
5026	027606	020040	042440	051122			
5027	027614	050040	020103	043440			

5084	030252	001104	001016	002776	DT32:	.WORD	\$TESTN,\$ERRPC,TBUF,0		
5085	030260	000000							
5086	030262	001104	001016	002770	DT36:	.WORD	\$TESTN,\$ERRPC,RCSR,0		
5087	030270	000000							
5088	030272	001104	001016	002772	DT37:	.WORD	\$TESTN,\$ERRPC,RBUF,0		
5089	030300	000000							
5090	030302	001104	001016	003010	DT61:	.WORD	\$TESTN,\$ERRPC,LKS,0		
5091	030310	000000							
5092	030312	001104	001016	001020	DT72:	.WORD	\$TESTN,\$ERRPC,\$GDADR,\$BDADR,\$GDDAT,\$BDDAT,0		
5093	030320	001022	001024	001026					
5094	030326	000000							
5095	030330	001104	001016	003000	DT115:	.WORD	\$TESTN,\$ERRPC,CRCSR,\$GDDAT,\$BDDAT,0		
5096	030336	001024	001026	000000					
5097	030344	001104	001016	002770	DT137:	.WORD	\$TESTN,\$ERRPC,RCSR,\$BDDAT,0		
5098	030352	001026	000000						
5099	030356	001104	001016	001024	DT142:	.WORD	\$TESTN,\$ERRPC,\$GDDAT,\$BDDAT,0		
5100	030364	001026	000000						
5101	030370	001104	001016	003010	DT152:	.WORD	\$TESTN,\$ERRPC,LKS,FIRST,SECND,0		
5102	030376	013574	013576	000000					
5103	030404	001104	001016	003000	DT155:	.WORD	\$TESTN,\$ERRPC,CRCSR,XMTCT1,RECCT1,0		
5104	030412	014726	014732	000000					
5105	030420	001104	001016	002770	DT157:	.WORD	\$TESTN,\$ERRPC,RCSR,XMTCT2,RECCT2,0		
5106	030426	014730	014734	000000					
5107	030434	001104	001016	002770	DT161:	.WORD	\$TESTN,\$ERRPC,RCSR,OLDPC,BDVECT,0		
5108	030442	015114	015116	000000					
5109	030450	000200			BUF1:	.BLKW	200		:SLU1 INPUT BUFFER FOR BLAST TEST
5110	031050	000200			BUF2:	.BLKW	200		:SLU2 INPIT BUFFER FOR BLAST TEST
5111		000001				.END			

DT72	030312	1264	5092#	
ECHO	020106	848	4259#	4294
EMTVEC=	000030	812#	1581*	1582*
EM100	024277	1290	4709#	
EM101	024333	1295	4714#	
EM102	024364	1300	4719#	
EM103	024436	1305	4726#	
EM104	024505	1310	4733#	
EM105	024541	1315	4738#	
EM106	024604	1320	4744#	
EM107	024665	1325	4753#	
EM11	020650	1019	4376#	
EM110	024721	1330	4758#	
EM111	024756	1335	4763#	
EM112	025030	1340	4770#	
EM113	025100	1345	4777#	
EM114	025133	1350	4782#	
EM115	025177	1355	4789#	
EM116	025222	1360	4793#	
EM117	025265	1365	4799#	
EM120	025321	1370	4804#	
EM121	025363	1375	4810#	
EM122	025432	1380	4817#	
EM123	025466	1385	4822#	
EM124	025517	1390	4827#	
EM125	025571	1395	4835#	
EM126	025640	1400	4842#	
EM127	025674	1405	4847#	
EM13	020706	1027	4381#	
EM130	025737	1410	4853#	
EM131	026020	1415	4862#	
EM132	026054	1420	4867#	
EM133	026111	1425	4872#	
EM134	026163	1430	4880#	
EM135	026233	1435	4887#	
EM136	026266	1440	4892#	
EM137	026332	1445	4898#	
EM14	020754	1032	4388#	
EM140	026401	1450	4905#	
EM141	026441	1455	4911#	
EM142	026505	1460	4918#	
EM143	026543	1465	4924#	
EM144	026577	1470	4929#	
EM145	026634	1475	4934#	
EM146	026670	1480	4939#	
EM147	026724	1485	4944#	
EM15	021024	1037	4395#	
EM150	026751	1490	4948#	
EM151	027013	1495	4954#	
EM152	027070	1500	4962#	
EM153	027122	1505	4967#	
EM154	027155	1510	4972#	
EM155	027210	1515	4977#	
EM156	027245	1520	4982#	
EM157	027312	1525	4989#	
EM16	021072	1042	4402#	

EM160	027347	1530	4994#						
EM161	027414	1535	5001#						
EM17	021130	1047	4407#						
EM20	021170	1052	4413#						
EM21	021236	1057	4420#						
EM22	021306	1062	4427#						
EM23	021352	1067	4433#						
EM24	021410	1072	4438#						
EM25	021450	1077	4444#						
EM26	021523	1082	4452#						
EM27	021560	1087	4457#						
EM30	021623	1092	4463#						
EM31	021675	1097	4471#						
EM32	021734	1102	4477#						
EM33	021773	1107	4483#						
EM34	022045	1112	4491#						
EM35	022113	1117	4498#						
EM36	022155	1122	4504#						
EM37	022214	1127	4510#						
EM40	022253	1132	4516#						
EM41	022323	1137	4523#						
EM42	022361	1142	4529#						
EM43	022421	1147	4535#						
EM44	022467	1152	4542#						
EM45	022537	1157	4549#						
EM46	022605	1162	4556#						
EM47	022643	1167	4562#						
EM5	020510	1001	4358#						
EM50	022703	1172	4568#						
EM51	022751	1177	4575#						
EM52	023021	1182	4582#						
EM53	023065	1187	4589#						
EM54	023123	1192	4595#						
EM55	023163	1197	4601#						
EM56	023236	1202	4609#						
EM57	023273	1207	4614#						
EM6	020552	1006	4364#						
EM60	023336	1212	4620#						
EM61	023410	1217	4627#						
EM62	023441	1222	4632#						
EM63	023503	1227	4638#						
EM64	023541	1232	4644#						
EM65	023571	1237	4649#						
EM66	023623	1242	4654#						
EM67	023663	1247	4660#						
EM7	020611	1011	4370#						
EM70	023723	1252	4666#						
EM71	023755	1257	4671#						
EM72	024006	1262	4676#						
EM74	024067	1270	4685#						
EM75	024132	1275	4691#						
EM76	024166	1280	4696#						
EM77	024230	1285	4702#						
ERRVEC=	000004	805#	1594	1595*	1606*	3735	3736*	3738*	3741*
FINIE	014664	3486	3501#						
FIRST	013574	3324*	3326	3340#	5101				

\$\$NEWT	817#	1638	1655	1671	1715	1737	1752	1768	1798	1842	1888	1915	1933	1952	1966
	1993	2016	2032	2048	2079	2127	2177	2207	2223	2242	2291	2326	2369	2408	2440
	2473	2506	2544	2581	2619	2649	2678	2706	2734	2766	2800	2833	2871	2902	2934
	2969	2996	3029	3051	3082	3105	3140	3193	3233	3264	3298	3344			
\$\$SET	4232#	4241	4242	4243	4244	4246	4248	4249	4250						
\$\$SETM	1608#														
\$\$SKIP	817#														
.EQUAT	696#	707													
.HEADE	696#														
.SETUP	696#	840													
.\$ACT1	696#	852													
.\$APT8	696#	932#													
.\$APTH	696#	862													
.\$CMTA	696#	885													
.\$EOP	696#	3515													
.\$ERRT	696#	3621													
.\$READ	696#	4065													
.\$SCOP	696#	3716													
.\$TRAP	696#	4209													
.\$TYPD	696#	3996													
.\$TYPE	696#	3823													
.\$TYPO	696#	3919													

. ABS. 031450 000

ERRORS DETECTED: 0

DSKM:CJKDFA,DSKZ:CJKDFA.LST/SOL/NL:TOC/CRF=CJKDFA.P11
RUN-TIME: 75 47 4 SECONDS
RUN-TIME RATIO: 184/127=1.4
CORE USED: 29K (57 PAGES)