

## IDENTIFICATION

PRODUCT CODE: AC-F602B MC  
PRODUCT NAME: C.MDFB0 11/24 OPTIONS DIAGNOSTIC  
PRODUCT DATE: DEC 84  
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): 1981,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAP <sup>c</sup>	DEC/X11

HISTORY

REVISION 1  
REVISION 2

FIRST RELEASE OF DIAGNOSTIC  
THIS REVISION WAS MADE SINCE THE  
WRONG BINARY WAS IN THE FIELD

## TABLE OF CONTENTS

1.0	General Program Information
1.1	Abstract
1.2	System Requirements
	A. Hardware Requirements
	B. Software Environments
1.3	Related Documents and Standards
1.4	Prerequisite Diagnostics
1.5	Assumptions
2.0	Operating Instructions
2.1	Loading and Starting Procedures
2.1.1	Starting Procedure
2.2	Execution times
3.0	Error Information
3.1	Error Reporting Procedures
3.2	Error Halts
4.0	Performance and Progress Reports
5.0	DEVICE INFORMATION TABLES
6.0	PROGRAM DESCRIPTION
6.1	PROGRAM EXECUTION CHARACTERISTICS
6.2	SUBTEST SUMMARIES
6.3	SPECIAL SUBROUTINE DESCRIPTIONS
6.3.1	ECHO TEST
6.3.2	TERMINAL OUTPUT TEST
7.0	LISTING

## 1.0 GENERAL PROGRAM INFORMATION

### 1.1 ABSTRACT

This program tests both serial line units (SLU's) and the line time clock (LTC) on the 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

## 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 SLUI is operating 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 Location = 176)

- A START AT 200.  
 After checking the transmitter, the program will print its identification and report the number of devices under test (number is octal). "END PASS" is printed after a full pass has been made on all devices under test.
- B START AT 204.       \*\*\*\*NOTE\*\*\*\*  
 The "ECHO" test will be executed. An "\*" is printed at the beginning of the test. The echo test reads a character from the terminal, writes that character to the terminal, and reports any error flags set in the receiver buffer. A CONTROL-C (?C) halts the test and prints "STOP" at the terminal continuing restarts the ECHO test.
- C. START AT 210.       \*\*\*\*NOTE\*\*\*\*

The terminal output test will be executed. Depressing any character at the terminal halts the test. Continuing restarts the test. The test outputs 32 characters on a line and repeats the pattern every three lines. The pattern is as follows (OCTAL code 040 -> 377):

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

This bottom line could be the following if the terminal does not have lower case:

```
@ABCDEFGHIJKLMNPQRSTUVWXYZ[] (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 =
```

where XXXXXX is equal to the present switch register contents in octal. The operator can then type;

1. <CR> if no changes are to be made.
2. 6 DIGITS <CR> to represent in octal the new switch register setting.
3. CONTROL-U if the operator makes an error while inputting the new switch register setting.

### 2.3 EXECUTION TIMES

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

### 3.0 ERROR INFORMATION

#### 3.1 ERROR REPORTING PROCEDURES

If a routine fails and the inhibit error timeout (BIT13) of the SWR is 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 timeout. 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.)

for software switch operation, the switch register can be changed by typing a CONTROL G at the console during error printouts. 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 timeout.

### 3.2 ERROR HALTS

There are 5 errors in this program that cause a halt. These errors are on 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 unaffected 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.

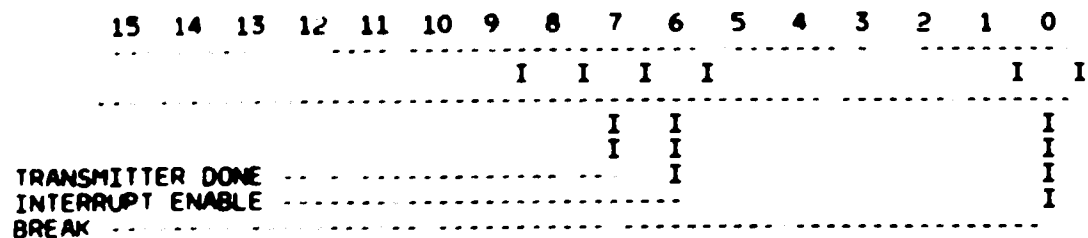




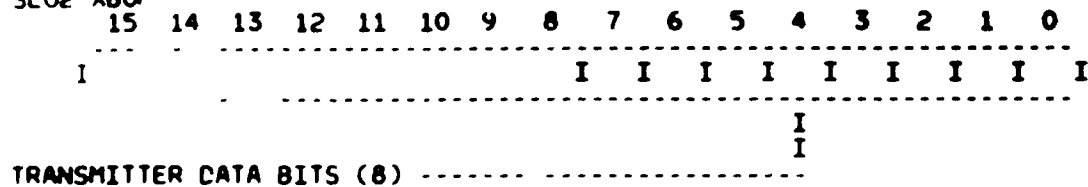


L1

SLU2 XCSR



SLU2 XBUF



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

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

- TEST4 TEST THAT SLU1 TRANSMITTER BIT7 (DONE) SETS WITH RESET. THE TRANSMITTER BUFFER IS LOADED WITH A CHARACTER, AS SOON AS "DONE" SETS A SECOND CHARACTER IS LOADED INTO THE BUFFER. BECAUSE THE FIRST CHARACTER IS STILL BEING SHIFTED OUT OF THE UART "DONE" WILL NOT NORMALLY SET FOR AT LEAST 1 MS (DEPENDING ON BAUD RATE). THE PROGRAM ISSUES A RESET BEFORE THE TIME IS UP AND IMMEDIATELY CHECKS FOR "DONE", IF IT IS SET THE PROGRAM ASSUMES IT WAS SET BY THE RESET INSTRUCTION. THIS ERROR DOES NOT CAUSE A HALT.
- TEST5 TEST ABILITY TO ACCESS SLU1 RECEIVER CONTROL AND STATUS REGISTER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.
- TEST6 TEST ABILITY TO ACCESS SLU1 RECEIVER BUFFER. AN ERROR IS REPORTED IF ACCESS CAUSES A TIMEOUT TRAP.
- 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.

- 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

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.

#### 6.3.2 TERMINAL OUTPUT TEST

C.  
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.

SEQ 0015

7.0 LISTING

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

```

.TITLE CJKDFBO 11/24 OPTIONS DIAGNOSTIC
;*COPYRIGHT (C) DEC-84
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP 11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC 11-DZQAC-C5)DEC-84.
;*
$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
DOISP= 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
PRO= 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
  
```

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



57 000340  
58  
59  
60 100000  
61 040000  
62 020000  
63 010000  
64 004000  
65 002000  
66 001000  
67 000400  
68 000200  
69 000100  
70 000040  
71 000020  
72 000010  
73 000004  
74 000002  
75 000001  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88 100000  
89 040000  
90 020000  
91 010000  
92 004000  
93 002000  
94 001000  
95 000400  
96 000200  
97 000100  
98 000040  
99 000020  
100 000010  
101 000004  
102 000002  
103 000001  
104  
105  
106  
107  
108  
109  
110  
111  
112

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

```

113
114
115
116      000004
117      000010
118      000014
119      000014
120      000014
121      000020
122      000024
123      000030
124      000034
125      000060
126      000064
127      000240
128      176500
129      000300
130      000400
131      000001
132      161000
133      000003
134
135
136      000000
137
138
139
140
141
142
143      000014
144      000014 015070
145      000016 000340
146
147      000042
148      000042 000000
149
150
151
152
153      000174
154      000174 000000
155      000176 000000
156
157      000200
158      000200 000137 003036
159      000204 000137 020106
160      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
AUSMR= 400
$TN= 1
$SMR= 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 IMPPOPERLY LOADED VECTORS

.=14              ; THE BPT TRAP VECTOR POINTS TO THE
.WORD CATCH      ; ILLEGAL TRAP HANDLER "CATCH"
.WORD 340

.=42
.WORD 0

.=174
DISPREG: .WORD 0
SMREG: .WORD 0

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

```

```

161
162      000500
163
164      .SBTTL      ACT11 HOOKS
165
166      ;*****
167      ;HOOKS REQUIRED BY ACT11
168      $SVC=      ;SAVE PC
169      .-46
170      $ENDAD      ;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .IEOP
171      .-52
172      .WORD      0      ;2)SET LOC.52 TO ZERO
173      .-$SVC      ; RESTORE PC
174      .SBTTL      APT PARAMETER BLOCK
175
176      ;*****
177      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
178      ;*****
179      . $X=      ;;SAVE CURRENT LOCATION
180      .-24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
181      200      ;;FOR APT START UP
182      .-44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
183      $APTHDR    ;;POINT TO APT HEADER BLOCK
184      .-$X      ;;RESET LOCATION COUNTER
185      ;*****
186      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
187      ;INTERFACE SPEC.
188
189      $APTHD:
190      $HIBTS: .WORD      0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
191      $MBADR: .WORD      $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0 15)
192      $TSTM:  .WORD      50      ;;RUN TIM OF LONGEST TEST
193      $PASTM: .WORD      60      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
194      $UNITM: .WORD      55      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
195      .WORD      $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

196  
 197  
 198  
 199  
 200  
 201  
 202 001000  
 203 001000  
 204 001000 000000  
 205 001002 000  
 206 001003 000  
 207 001004 000000  
 208 001006 000000  
 209 001010 000000  
 210 001012 000000  
 211 001014 000  
 212 001015 001  
 213 001016 000000  
 214 001020 000000  
 215 001022 000000  
 216 001024 000000  
 217 001026 000000  
 218 001030 000000  
 219 001032 000000  
 220 001034 000  
 221 001035 000  
 222 001036 000000  
 223 001040 177570  
 224 001042 177570  
 225 001044 177560  
 226 001046 177562  
 227 001050 177564  
 228 001052 177566  
 229 001054 000  
 230 001055 002  
 231 001056 012  
 232 001057 000  
 233 001060 000000  
 234 001062 000000  
 235 001064 000000  
 236 001066 000000  
 237 001070 000000  
 238 001072 000000  
 239 001074 077  
 240 001075 015  
 241 001076 000012  
 242  
 243  
 244  
 245  
 246  
 247 001100  
 248 001100 000000  
 249 001102 000000  
 250 001104 000000  
 251 001106 000000

.SBTTL COMMON TAGS

```

;*****
; *THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
; *USED IN THE PROGRAM.
;*****
      .-1000
$CMTAG:      .-1000      ;; START OF COMMON TAGS
      .WORD      0
$TSTM:      .BYTE      0      ;; CONTAINS THE TEST NUMBER
$ERFLG:     .BYTE      0      ;; CONTAINS ERROR FLAG
$ICNT:      .WORD      0      ;; CONTAINS SUBTEST ITERATION COUNT
$LPADR:     .WORD      0      ;; CONTAINS SCOPE LOOP ADDRESS
$LPERR:     .WORD      0      ;; CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL:     .WORD      0      ;; CONTAINS TOTAL ERRORS DETECTED
$ITEMB:     .BYTE      0      ;; CONTAINS ITEM CONTROL BYTE
$ERMAX:     .BYTE      1      ;; CONTAINS MAX. ERRORS PER TEST
$ERRPC:     .WORD      0      ;; CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR:     .WORD      0      ;; CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR:     .WORD      0      ;; CONTAINS ADDRESS OF 'BAD' DATA
$GDAT:      .WORD      0      ;; CONTAINS 'GOOD' DATA
$BDAT:      .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
SMR:        .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:        .ASCII    <12>     ;; LINE FEED
;*****

```

.SBTTL APT MAILBOX-ETABLE

```

;*****
.EVEN
$MAIL:      .WORD      0          ;; APT MAILBOX
$MSGTY:     .WORD      AMSGTY    ;; MESSAGE TYPE CODE
$FATAL:     .WORD      AFATAL    ;; FATAL ERROR NUMBER
$TESTN:     .WORD      ATESTN    ;; TEST NUMBER
$PASS:      .WORD      APASS     ;; PASS COUNT
;*****

```

252	001110	000000	\$DEVCT: .WORD	ADEVCT	::DEVICE COUNT
253	001112	000000	\$UNIT: .WORD	AUNIT	::I/O UNIT NUMBER
254	001114	000000	\$MSGAD: .WORD	AMSGAD	::MESSAGE ADDRESS
255	001116	000000	\$MSGLG: .WORD	AMSLG	::MESSAGE LENGTH
256	001120		\$ETABLE:		::APT ENVIRONMENT TABLE
257	001120	000	\$ENV: .BYTE	AENV	::ENVIRONMENT BYTE
258	001121	000	\$ENVM: .BYTE	AENVM	::ENVIRONMENT MODE BITS
259	001122	000000	\$SWREG: .WORD	ASWREG	::APT SWITCH REGISTER
260	001124	000400	\$USWR: .WORD	AUSWR	::USER SWITCHES
261	001126	000000	\$CPUOP: .WORD	ACPUOP	::CPU TYPE,OPTIONS
262			;*:		BITS 15-11-CPU TYPE
263			;*:		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
264			;*:		11/70=06,PDQ=07,Q=10
265			;*:		BIT 10-REAL TIME CLOCK
266			;*:		BIT 9-FLOATING POINT PROCESSOR
267			;*:		BIT 8-MEMORY MANAGEMENT
268	001130	000	\$MAMS1: .BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
269	001131	000	\$MTYP1: .BYTE	AMTYP1	::MEM. TYPE,BLK#1
270			;*:		MEM.TYPE BYTE -- (HIGH BYTE)
271			;*:		900 NSEC CORE=001
272			;*:		400 NSEC BIPOLAR=002
273			;*:		500 NSEC MOS=003
274	001132	000000	\$MADR1: .WORD	AMADR1	::HIGH ADDRESS,BLK#1
275			;*:		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
276	001134	000	\$MAMS2: .BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
277	001135	000	\$MTYP2: .BYTE	AMTYP2	::MEM.TYPE,BLK#2
278	001136	000000	\$MADR2: .WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
279	001140	000	\$MAMS3: .BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
280	001141	000	\$MTYP3: .BYTE	AMTYP3	::MEM.TYPE,BLK#3
281	001142	000000	\$MADR3: .WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
282	001144	000	\$MAMS4: .BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
283	001145	000	\$MTYP4: .BYTE	AMTYP4	::MEM.TYPE,BLK#4
284	001146	000000	\$MADR4: .WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
285	001150	000300	\$VECT1: .WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
286	001152	000000	\$VECT2: .WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
287	001154	176500	\$BASE: .WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
288	001156	000000	\$DEVH: .WORD	ADEVH	::DEVICE MAP
289	001160		\$ETEND:		
290			.MEXIT		

.SBTTL ERROR POINTER TABLE

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

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

#ERRTB:

.#ERRTB:

.BLKW 20 ;THIS BLOCK OF 16 LOCATIONS IS HERE TO PACIFY SYSMAC

291  
 292  
 293  
 294  
 295  
 296  
 297  
 298  
 299  
 300  
 301  
 302  
 303  
 304  
 305 001160  
 306  
 307 001160  
 308 001160 000020  
 309  
 310  
 311  
 312 001220 020510  
 313 001222 027431  
 314 001224 030212  
 315 001226 000000  
 316  
 317 001230 020552  
 318 001232 027456  
 319 001234 030222  
 320 001236 000000  
 321  
 322 001240 020611  
 323 001242 027503  
 324 001244 030232  
 325 001246 000000  
 326  
 327 001250 000004  
 328  
 329  
 330 001260 020650  
 331 001262 027431  
 332 001264 030212  
 333 001266 000000  
 334  
 335 001270 000004  
 336  
 337  
 338 001300 020706  
 339 001302 027431  
 340 001304 030212  
 341 001306 000000  
 342  
 343 001310 020754  
 344 001312 027431  
 345 001314 030212  
 346 001316 000000

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,CRCR  
 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

347					
348	001320	021024	EM15		;"SLUI XMIT INTERRUPTED WITH PRIORITY 7"
349	001322	027431	DM5		;"TEST# ERR PC TCSR"
350	001324	030212	DT5		;"#TESTN,#ERRPC,CTCSR"
351	001326	000000	0		
352					
353	001330	021072	EM16		;"CAN NOT SET BIT6 OF SLUI TCSR"
354	001332	027431	DM5		;"TEST# ERR PC TCSR"
355	001334	030212	DT5		;"#TESTN,#ERRPC,CTCSR"
356	001336	000000	0		
357					
358	001340	021130	EM17		;"CAN NOT CLEAR BIT6 OF SLUI TCSR"
359	001342	027431	DM5		;"TEST# ERR PC TCSR"
360	001344	030212	DT5		;"#TESTN,#ERRPC,CTCSR"
361	001346	000000	0		
362					
363	001350	021170	EM20		;"RESET DID NOT CLEAR BIT6 OF SLUI TCSR"
364	001352	027431	DM5		;"TEST# ERR PC TCSR"
365	001354	030212	DT5		;"#TESTN,#ERRPC,CTCSR"
366	001356	000000	0		
367					
368	001360	021236	EM21		;"BIT6 OF SLUI RCSR NOT CLEAR AFTER RESET"
369	001362	027456	DM6		;"TEST# ERR PC RCSR"
370	001364	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
371	001366	000000	0		
372					
373	001370	021306	EM22		;"SLUI RCVR INTERRUPT WITH PRIORITY 7"
374	001372	027456	DM6		;"TEST# ERR PC RCSR"
375	001374	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
376	001376	000000	0		
377					
378	001400	021352	EM23		;"CAN NOT SET BIT6 OF SLUI RCSR"
379	001402	027456	DM6		;"TEST# ERR PC RCSR"
380	001404	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
381	001406	000000	0		
382					
383	001410	021410	EM24		;"CAN NOT CLEAR BIT6 OF SLUI RCSR"
384	001412	027456	DM6		;"TEST# ERR PC RCSR"
385	001414	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
386	001416	000000	0		
387					
388	001420	021450	EM25		;"CAN NOT CLEAR BIT6 OF SLUI RCSR WITH RESET2"
389	001422	027456	DM6		;"TEST# ERR PC RCSR"
390	001424	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
391	001426	000000	0		
392					
393	001430	021523	EM26		;"SLUI RECEIVER DONE NEVER SET"
394	001432	027456	DM6		;"TEST# ERR PC RCSR"
395	001434	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
396	001436	000000	0		
397					
398	001440	021560	EM27		;"RESET DID NOT CLEAR SLUI RCVR DONE"
399	001442	027456	DM6		;"TEST# ERR PC RCSR"
400	001444	030222	DT6		;"#TESTN,#ERRPC,CRCSR"
401	001446	000000	0		
402					

403	001450	021623	EM30	;"READING SLU1 RBUF DID NOT CLEAR RCVR DONE"
404	001452	027456	DH6	;"TEST# ERR PC RCSR"
405	001454	030222	DT6	;"#TESTN,#ERRPC,RCSR"
406	001456	000000	0	
407				
408	001460	021675	EM31	;"SLU2 TCSR DID NOT RETURN SSYNC"
409	001462	027431	DH5	;"TEST# ERR PC TCSR"
410	001464	030242	DT31	;"#TESTN,#ERRPC,TCSR"
411	001466	000000	0	
412				
413	001470	021734	EM32	;"SLU2 TBUF DID NOT RETURN SSYNC"
414	001472	027530	DH32	;"TEST# ERR PC TBUF"
415	001474	030252	DT32	;"#TESTN,#ERRPC,TBUF"
416	001476	000000	0	
417				
418	001500	021773	EM33	;"SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL"
419	001502	027431	DH5	;"TEST# ERR PC TCSR"
420	001504	030242	DT31	;"#TESTN,#ERRPC,TCSR"
421	001506	000000	0	
422				
423	001510	022045	EM34	;"SLU2 TCSR DONE NOT SET AFTER TRANSMIT"
424	001512	027431	DH5	;"TEST# ERR PC TCSR"
425	001514	030242	DT31	;"#TESTN,#ERRPC,TCSR"
426	001516	000000	0	
427				
428	001520	022113	EM35	;"SLU2 TCSR DONE NOT SET WITH RESET"
429	001522	027431	DH5	;"TEST# ERR PC TCSR"
430	001524	030242	DT31	;"#TESTN,ERRPC,TCSR"
431	001526	000000	0	
432				
433	001530	022155	EM36	;"SLU2 RCSR DID NOT RETURN SSYNC"
434	001532	027456	DH6	;"TEST# ERR PC RCSR"
435	001534	030262	DT36	;"#TESTN,ERRPC,RCSR"
436	001536	000000	0	
437				
438	001540	022214	EM37	;"SLU2 RBUF DID NOT RETURN SSYNC"
439	001542	027503	DH7	;"TEST# ERR PC RBUF"
440	001544	030272	DT37	;"#TESTN,ERRPC,RBUF"
441	001546	000000	0	
442				
443	001550	022253	EM40	;"BIT0 OF SLU2 TCSR NOT CLEAR AFTER RESET"
444	001552	027431	DH5	;"TEST# ERR PC TCSR"
445	001554	030242	DT31	;"#TESTN,#ERRPC,TCSR"
446	001556	000000	0	
447				
448	001560	022323	EM41	;"CAN NOT SET BIT0 OF SLU2 TCSR"
449	001562	027431	DH5	;"TEST# ERR PC TCSR"
450	001564	030242	DT31	;"#TESTN,#ERRPC,TCSR"
451	001566	000000	0	
452				
453	001570	022361	EM42	;"CAN NOT CLEAR BIT0 OF SLU2 TCSR"
454	001572	027431	DH5	;"TEST# ERR PC TCSR"
455	001574	030242	DT31	;"#TESTN,#ERRPC,TCSR"
456	001576	000000	0	
457				
458	001600	022421	EM43	;"RESET DID NOT CLEAR BIT0 OF SLU2 TCSR"



459	001602	027431	DH5	;"TEST# ERR PC TCSR
460	001604	030242	DT31	;"TESTN,\$ERRPC,TCSR
461	001606	000000	0	
462				
463	001610	022467	EM44	;"BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET2
464	001612	027431	DH5	;"TEST# ERR PC TCSR
465	001614	030242	DT31	;"TESTN,\$ERRPC,TCSR
466	001616	000000	0	
467				
468	001620	022537	EM45	;"SLU2 XMIT INTERRUPTED WITH PRIORITY 7"
469	001622	027431	DH5	;"TEST# ERR PC TCSR
470	001624	030242	DT31	;"TESTN,\$ERRPC,TCSR
471	001626	000000	0	
472				
473	001630	022605	EM46	;"CAN NOT SET BIT6 OF SLU2 TCSR"
474	001632	027431	DH5	;"TEST# ERR PC TCSR
475	001634	030242	DT31	;"TESTN,\$ERRPC,TCSR
476	001636	000000	0	
477				
478	001640	022643	EM47	;"CAN NOT CLEAR BIT6 OF SLU2 TCSR"
479	001642	027431	DH5	;"TEST# ERR PC TCSR
480	001644	030242	DT31	;"TESTN,\$ERRPC,TCSR
481	001646	000000	0	
482				
483	001650	022703	EM50	;"RESET DID NOT CLEAR BIT6 OF SLU2 TCSR"
484	001652	027431	DH5	;"TEST# ERR PC TCSR
485	001654	030242	DT31	;"TESTN,\$ERRPC,TCSR
486	001656	000000	0	
487				
488	001660	022751	EM51	;"BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET"
489	001662	027456	DH6	;"TEST# ERR PC RCSR"
490	001664	030262	DT36	;"TESTN,ERRPC,RCSR
491	001666	000000	0	
492				
493	001670	023021	EM52	;"SLU2 RCVR INTERRUPT WITH PRIORITY 7"
494	001672	027456	DH6	;"TEST# ERR PC RCSR"
495	001674	030262	DT36	;"TESTN,ERRPC,RCSR
496	001676	000000	0	
497				
498	001700	023065	EM53	;"CAN NOT SET BIT6 OF SLU2 RCSR"
499	001702	027456	DH6	;"TEST# ERR PC RCSR"
500	001704	030262	DT36	;"TESTN,ERRPC,RCSR
501	001706	000000	0	
502				
503	001710	023123	EM54	;"CAN NOT CLEAR BIT6 OF SLU2 RCSR"
504	001712	027456	DH6	;"TEST# ERR PC RCSR"
505	001714	030262	DT36	;"TESTN,ERRPC,RCSR
506	001716	000000	0	
507				
508	001720	023163	EM55	;"CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET2"
509	001722	027456	DH6	;"TEST# ERR PC RCSR"
510	001724	030262	DT36	;"TESTN,ERRPC,RCSR
511	001726	000000	0	
512				
513	001730	023236	EM56	;"SLU2 RECEIVER DONE NEVER SET"
514	001732	027456	DH6	;"TEST# ERR PC RCSR"

515	001734	030262	DT36	;	TESTN,ERRPC,RCSR
516	001736	000000	0		
517					
518	001740	023273	EM57	;	"RESET DID NOT CLEAR SLU2 RCVR DONE"
519	001742	027456	DM6	;	"TEST# ERR PC RCSR"
520	001744	030262	DT36	;	TESTN,ERRPC,RCSR
521	001746	000000	0		
522					
523	001750	023336	EM60	;	"READING SLU2 RBUF DID NOT CLEAR RCVR DONE"
524	001752	027456	DM6	;	"TEST# ERR PC RCSR"
525	001754	030262	DT36	;	TESTN,ERRPC,RCSR
526	001756	000000	0		
527					
528	001760	023410	EM61	;	LKS DID NOT RETURN SSYNC
529	001762	027555	DM61	;	"TEST# ERR PC LKS"
530	001764	030302	DT61	;	TESTN,ERRPC,LKS
531	001766	000000	0		
532					
533	001770	023441	EM62	;	"BIT6 OF LKS NOT CLEAR AFTER RESET"
534	001772	027555	DM61	;	"TEST# ERR PC LKS"
535	001774	030302	DT61	;	TESTN,ERRPC,LKS
536	001776	000000	0		
537					
538	002000	023503	EM63	;	LKS INTERRUPT WITH PRIORITY 7
539	002002	027555	DM61	;	"TEST# ERR PC LKS"
540	002004	030302	DT61	;	TESTN,ERRPC,LKS
541	002006	000000	0		
542					
543	002010	023541	EM64	;	"CAN NOT SET BIT6 OF LKS"
544	002012	027555	DM61	;	"TEST# ERR PC LKS"
545	002014	030302	DT61	;	TESTN,ERRPC,LKS
546	002016	000000	0		
547					
548	002020	023571	EM65	;	"CAN NOT CLEAR BIT6 OF LKS"
549	002022	027555	DM61	;	"TEST# ERR PC LKS"
550	002024	030302	DT61	;	TESTN,ERRPC,LKS
551	002026	000000	0		
552					
553	002030	023623	EM66	;	"RESET DID NOT CLEAR BIT6 OF LKS"
554	002032	027555	DM61	;	"TEST# ERR PC LKS"
555	002034	030302	DT61	;	TESTN,ERRPC,LKS
556	002036	000000	0		
557					
558	002040	023663	EM67	;	"BIT7 OF LKS NOT SET AFTER RESET2"
559	002042	027555	DM61	;	"TEST# ERR PC LKS"
560	002044	030302	DT61	;	TESTN,ERRPC,LKS
561	002046	000000	0		
562					
563	002050	023723	EM70	;	"CAN NOT CLEAR BIT7 OF LKS"
564	002052	027555	DM61	;	"TEST# ERR PC LKS"
565	002054	030302	DT61	;	TESTN,ERRPC,LKS
566	002056	000000	0		
567					
568	002060	023755	EM71	;	"BIT7 OF LKS DOES NOT SET"
569	002062	027555	DM61	;	"TEST# ERR PC LKS"
570	002064	030302	DT61	;	TESTN,ERRPC,LKS

571	002066	000000	0	
572				
573	002070	024006	EM72	; WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER
574	002072	027601	DM72	; "TEST# ERR PC GOOD BAD GDATA BDATA
575	002074	030312	DT72	; #TESTN, #ERRPC, #GDADR, #BDADR, #GDDAT, #BDDAT
576	002076	000000	0	
577				
578	002100	000004	.BLKW 4	; THE LAST IN A LONG LINE OF PACIFICATION
579				
580				
581	002110	024067	EM74	; "SLU1 XMIT INTERRUPTS WHEN DISABLED
582	002112	027431	DM5	; "TEST# ERR PC TCSR"
583	002114	030212	DT5	; #TESTN, #ERRPC, CTCSR
584	002116	000000	0	
585				
586	002120	024132	EM75	; "SLU1 XMIT DID NOT INTERRUPT
587	002122	027431	DM5	; "TEST# ERR PC TCSR"
588	002124	030212	DT5	; #TESTN, #ERRPC, CTCSR
589	002126	000000	0	
590				
591	002130	024166	EM76	; "SLU1 XMIT INTERRUPT AT PRIORITY 7"
592	002132	027431	DM5	; "TEST# ERR PC TCSR"
593	002134	030212	DT5	; #TESTN, #ERRPC, CTCSR
594	002136	000000	0	
595				
596	002140	024230	EM77	; "SLU1 XMIT INTERRUPTS WITH ENABLE CLEAR
597	002142	027431	DM5	; "TEST# ERR PC TCSR"
598	002144	030212	DT5	; #TESTN, #ERRPC, CTCSR
599	002146	000000	0	
600				
601	002150	024277	EM100	; "SLU1 XMIT DID NOT INTERRUPT
602	002152	027431	DM5	; "TEST# ERR PC TCSR"
603	002154	030212	DT5	; #TESTN, #ERRPC, CTCSR
604	002156	000000	0	
605				
606	002160	024333	EM101	; "SLU1 XMIT RE-INTERRUPTED
607	002162	027431	DM5	; "TEST# ERR PC TCSR"
608	002164	030212	DT5	; #TESTN, #ERRPC, CTCSR
609	002166	000000	0	
610				
611	002170	024364	EM102	; "LOADING SLU1 TBUF DID NOT CLEAR INTERRUPT
612	002172	027431	DM5	; "TEST# ERR PC TCSR"
613	002174	030212	DT5	; #TESTN, #ERRPC, CTCSR
614	002176	000000	0	
615				
616	002200	024436	EM103	; "SLU1 RCVR INTERRUPTS WITH ENABLE CLEAR"
617	002202	027456	DM6	; "TEST# ERR PC RCSR"
618	002204	030222	DT6	; #TESTN, #ERRPC, CRCSR
619	002206	000000	0	
620				
621	002210	024505	EM104	; "SLU1 RCVR DID NOT INTERRUPT
622	002212	027456	DM6	; "TEST# ERR PC RCSR"
623	002214	030222	DT6	; #TESTN, #ERRPC, CRCSR
624	002216	000000	0	
625				
626	002220	024541	EM105	; "SLU1 RCVR INTERRUPTS AT PRIORITY 7"

627	002222	027456	DM6	;"TEST# ERR PC RCSR
628	002224	030222	DT6	;"TESTN,#ERRPC,CRCR
629	002226	000000	0	
630				
631	002230	024604	EM106	;"SLU1 RCVR INTERRUPTS WITH INTERRUPT' ENABLE CLEAR
632	002232	027456	DM6	;"TEST# ERR PC RCSR
633	002234	030222	DT6	;"TESTN,#ERRPC,CRCR
634	002236	000000	0	
635				
636	002240	024665	EM107	;"SLU1 RCVR DID NOT INTERRUPT
637	002242	027456	DM6	;"TEST# ERR PC RCSR"
638	002244	030222	DT6	;"TESTN,#ERRPC,CRCR
639	002246	000000	0	
640				
641	002250	024721	EM110	;"SLU1 RECEIVER RE-INTERRUPTED"
642	002252	027456	DM6	;"TEST# ERR PC RCSR"
643	002254	030222	DT6	;"TESTN,#ERRPC,CRCR
644	002256	000000	0	
645				
646	002260	024756	EM111	;"SLU1 READING RBUF DID NOT CLEAR INTERRUPT
647	002262	027456	DM6	;"TEST# ERR PC RCSR"
648	002264	030222	DT6	;"TESTN,#ERRPC,CRCR
649	002266	000000	0	
650				
651	002270	025030	EM112	;"RESET DID NOT CLEAR SLU1 RCVR INTERRUPT
652	002272	027456	DM6	;"TEST# ERR PC RCSR"
653	002274	030222	DT6	;"TESTN,#ERRPC,CRCR
654	002276	000000	0	
655				
656	002300	025100	EM113	;"SLU1 'OR' FLAG DID NOT SET"
657	002302	027456	DM6	;"TEST# ERR PC RCSR"
658	002304	030222	DT6	;"TESTN,#ERRPC,CRCR
659	002306	000000	0	
660				
661	002310	025133	EM114	;"SLU1 'ERROR' NOT SET WITH OR FLAG
662	002312	027456	DM6	;"TEST# ERR PC RCSR"
663	002314	030222	DT6	;"TESTN,#ERRPC,CRCR
664	002316	000000	0	
665				
666	002320	025177	EM115	;"DATA COMPARE ERROR"
667	002322	027657	DM115	;"TEST# ERR PC CRCR GOOD BAD"
668	002324	030330	DT115	;"TESTN,#ERRPC,CRCR,GD,BD"
669	002326	000000	0	
670				
671	002330	025222	EM116	;"SLU2 XMIT INTERRUPTS WHEN DISABLED
672	002332	027431	DM5	;"TEST# ERR PC TCSR"
673	002334	030242	DT31	;"TESTN,#ERRPC,TCSR"
674	002336	000000	0	
675				
676	002340	025265	EM117	;"SLU2 XMIT DID NOT INTERRUPT"
677	002342	027431	DM5	;"TEST# ERR PC TCSR"
678	002344	030242	DT31	;"TESTN,#ERRPC,TCSR"
679	002346	000000	0	
680				
681	002350	025321	EM120	;"SLU2 XMIT INTERRUPT AT PRIORITY "
682	002352	027431	DM5	;"TEST# ERR PC TCSR"

683	002354	030242	DT31	;#TESTN,#ERRPC,TCSR
584	002356	000000	0	
685				
686	002360	025363	EM121	;"SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR
687	002362	027431	DM5	;"TEST# ERR PC TCSR"
688	002364	030242	DT31	;#TESTN,#ERRPC,TCSR
689	002366	000000	0	
690				
691	002370	025432	EM122	;"SLU2 XMIT DID NOT INTERRUPT
692	002372	027431	DM5	;"TEST# ERR PC TCSR"
693	002374	030242	DT31	;#TESTN,#ERRPC,TCSR
694	002376	000000	0	
695				
696	002400	025466	EM123	;"SLU2 XMIT RE-INTERRUPTED"
697	002402	027431	DM5	;"TEST# ERR PC TCSR"
698	002404	030242	DT31	;#TESTN,#ERRPC,TCSR
699	002406	000000	0	
700				
701	002410	025517	EM124	;"LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT"
702	002412	027431	DM5	;"TEST# ERR PC TCSR"
703	002414	030242	DT31	;#TESTN,#ERRPC,TCSR
704	002416	000000	0	
705				
706	002420	025571	EM125	;"SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR"
707	002422	027456	DM6	;"TEST# ERR PC RCSR"
708	002424	030262	DT36	;#TESTN,#ERRPC,RCSR
709	002426	000000	0	
710				
711	002430	025640	EM126	;"SLU2 RCVR DID NOT INTERRUPT"
712	002432	027456	DM6	;"TEST# ERR PC RCSR"
713	002434	030262	DT36	;#TESTN,#ERRPC,RCSR
714	002436	000000	0	
715				
716	002440	025674	EM127	;"SLU2 RCVR INTERRUPTS AT PRIORITY 7"
717	002442	027456	DM6	;"TEST# ERR PC RCSR"
718	002444	030262	DT36	;#TESTN,#ERRPC,RCSR
719	002446	000000	C	
720				
721	002450	025737	EM130	;"SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR"
722	002452	027456	DM6	;"TEST# ERR PC RCSR"
723	002454	030262	DT36	;#TESTN,#ERRPC,RCSR
724	002456	000000	0	
725				
726	002460	026020	EM131	;"SLU2 RCVR DID NOT INTERRUPT"
727	002462	027456	DM6	;"TEST# ERR PC RCSR"
728	002464	030262	DT36	;#TESTN,#ERRPC,RCSR
729	002466	000000	0	
730				
731	002470	026054	EM132	;"SLU2 RECEIVER RE-INTERRUPTED"
732	002472	027456	DM6	;"TEST# ERR PC RCSR"
733	002474	030262	DT36	;#TESTN,#ERRPC,RCSR
734	002476	000000	0	
735				
736	002500	026111	EM133	;"SLU2 READING RBUF DID NOT CLEAR INTERRUPT"
737	002502	027456	DM6	;"TEST# ERR PC RCSR"
738	002504	030262	DT36	;#TESTN,#ERRPC,RCSR

739	002506	000000	0	
740				
741	002510	026163	EM134	;"RESET DID NOT CLEAR SLU2 RCVR INTERRUPT"
742	002512	027456	DM6	;"TEST# ERR PC RCSR"
743	002514	030262	DT36	;"TESTN,#ERRPC,RCSR"
744	002516	000000	0	
745				
746	002520	026233	EM135	;"SLU2 OR' FLAG DID NOT SET"
747	002522	027456	DM6	;"TEST# ERR PC RCSR"
748	002524	030262	DT36	;"TESTN,#ERRPC,RCSR"
749	002526	000000	0	
750				
751	002530	026266	EM136	;"SLU2 'ERROR' NOT SET WITH 'OR' FLAG"
752	002532	027456	DM6	;"TEST# ERR PC RCSR"
753	002534	030262	DT36	;"TESTN,#ERRPC,RCSR"
754	002536	000000	0	
755				
756	002540	026332	EM137	;"SLU2 BREAK DID NOT TRANSMIT ALL ZEROES"
757	002542	027723	DM137	;"TEST# ERR PC RCSR DATA"
758	002544	030344	DT137	;"TESTN,#ERRPC,RCSR,#BDDAT"
759	002546	000000	0	
760				
761	002550	026401	EM140	;"BREAK DID NOT SET FRAMING ERROR"
762	002552	027456	DM6	;"TEST# ERR PC RCSR"
763	002554	030222	DT6	;"TESTN,#ERRPC,RCSR"
764	002556	000000	0	
765				
766	002560	026441	EM141	;"SLU2 'ERROR' NOT SET WITH 'FR' FLAG"
767	002562	027456	DM6	;"TEST# ERR PC RCSR"
768	002564	030262	DT36	;"TESTN,#ERRPC,RCSR"
769	002566	000000	0	
770				
771	002570	026505	EM142	;"DATA COMPARE ERROR WITH CABLE"
772	002572	027760	DM142	;"TEST# ERR PC RCSR GOOD BAD"
773	002574	030356	DT142	;"TESTN,#ERRPC,RCSR,GD,BD"
774	002576	000000	0	
775				
776	002600	026543	EM143	;"RTC INTERRUPT AT PRIORITY 7"
777	002602	027555	DM61	;"TEST# ERR PC LKS"
778	002604	030302	DT61	;"TESTN,#ERRPC,LKS"
779	002606	000000	0	
780				
781	002610	026577	EM144	;"RTC INTERRUPTS WHEN DISABLED"
782	002612	027555	DM61	;"TEST# ERR PC LKS"
783	002614	030302	DT61	;"TESTN,#ERRPC,LKS"
784	002616	000000	0	
785				
786	002620	026634	EM145	;"RTC INTERRUPT DID NOT OCCUR"
787	002622	027555	DM61	;"TEST# ERR PC LKS"
788	002624	030302	DT61	;"TESTN,#ERRPC,LKS"
789	002626	000000	0	
790				
791	002630	026670	EM146	;"RTC INTERRUPT DID NOT OCCUR"
792	002632	027555	DM61	;"TEST# ERR PC LKS"
793	002634	030302	DT61	;"TESTN,#ERRPC,LKS"
794	002636	000000	0	

795				
796	002640	026724	EM147	;"RTC DOUBLE INTERRUPT"
797	002642	027555	DW61	;"TEST# ERR PC LKS"
798	002644	030302	DT61	;"TESTN, \$ERRPC, LKS"
799	002646	000000	0	
800				
801	002650	026751	EM150	;"RESET DID NOT CLEAR RTC INTERRUPT"
802	002652	027555	DW61	;"TEST# ERR PC LKS"
803	002654	030302	DT61	;"TESTN, \$ERRPC, LKS"
804	002656	000000	0	
805				
806	002660	027013	EM151	;"RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS"
807	002662	027555	DW61	;"TEST# ERR PC LKS"
808	002664	030302	DT61	;"TESTN, \$ERRPC, LKS"
809	002666	000000	0	
810				
811	002670	027070	EM152	;"CLOCK REPEATABILITY ERROR"
812	002672	030014	DW152	;"TEST# ERR PC LKS CNT1 CNT2"
813	002674	030370	DT152	;"\$TESTN, \$ERRPC, LKS, FIRST, SECD"
814	002676	000000	0	
815				
816	002700	027122	EM153	;"SLU1 RECEIVER STATUS ERROR"
817	002702	027657	DW115	;"TEST# ERR PC RCSR GOOD BAD"
818	002704	030330	DT115	;"TESTN, \$ERRPC, RCSR, \$GDDTA, \$BDDAT"
819	002706	000000	0	
820				
821	002710	027155	EM154	;"SLU2 RECEIVER STATUS ERROR"
822	002712	027760	DW142	;"TESTN ERR PC RCSR GOOD BAD"
823	002714	030356	DT142	;"TESTN, \$ERRPC, RCSR, \$GDDAT, \$BDDAT"
824	002716	000000	0	
825				
826	002720	027210	EM155	;"INCORRECT RECEIVE COUNT SLU1"
827	002722	030061	DW155	;"TEST# ERR PC RCSR TRANS RCV"
828	002724	030404	DT155	;"TESTN, \$ERRPC, RCSR, XMTCT1, RCVCT1"
829	002726	000000	0	
830				
831	002730	027245	EM156	;"SLU1 DATA COMPARE ERROR IN EXERCISER"
832	002732	027657	DW115	;"TEST# ERR PC RCSR GOOD BAD"
833	002734	030330	DT115	;"TESTN, \$ERRPC, RCSR, GD, BD"
834	002736	000000	0	
835				
836	002740	027312	EM157	;"INCORRECT RECEIVE COUNT SLU2"
837	002742	030061	DW155	;"TEST# ERR PC RCSR TRANS RCV"
838	002744	030420	DT157	;"TESTN, \$ERRPC, RCSR, XMTCT2, RCVCT2"
839	002746	000000	0	
840				
841	002750	027347	EM160	;"SLU2 DATA COMPARE ERROR IN EXERCISER"
842	002752	027760	DW142	;"TEST# ERR PC RCSR GOOD BAD"
843	002754	030356	DT142	;"TESTN, \$ERRPC, RCSR, GD, BD"
844	002756	000000	0	
845				
846	002760	027414	EM161	;"TRAP CATCHER"
847	002762	030125	DW161	;"TEST# ERR PC OLDPC TRAP ADR"
848	002764	030434	DT161	;"TESTN, \$ERRPC, OLDPC, BDVECT"
849	002766	000000	0	
850				

851  
 852  
 853 002770 176500  
 854 002772 176502  
 855 002774 176504  
 856 002776 176506  
 857 003000 177560  
 858 003002 177562  
 859 003004 177564  
 860 003006 177566  
 861 003010 177546  
 862  
 863  
 864  
 865  
 866 003012 000300  
 867 003014 000302  
 868 003016 000304  
 869 003020 000306  
 870 003022 000060  
 871 003024 000062  
 872 003026 000064  
 873 003030 000066  
 874 003032 000100  
 875 003034 000102  
 876  
 877 003036 005037 001102  
 878 003042 005037 001100  
 879 003046 005037 001104  
 880 003052 005037 001156  
 881 003056  
 882  
 883  
 884 003056 012706 001000  
 885 003062 005026  
 886 003064 022706 001040  
 887 003070 001374  
 888 003072 012706 001000  
 889  
 890 003076 012737 015614 000020  
 891 003104 012737 000340 000022  
 892 003112 012737 015120 000030  
 893 003120 012737 000340 000032  
 894 003126 012737 020026 000034  
 895 003134 012737 000340 000036  
 896 003142 012737 015436 000024  
 897 003150 012737 000340 000026  
 898 003156 013737 014770 014762  
 899 003164 005037 001072  
 900 003170 112737 000001 001015  
 901 003176 012737 003176 001006  
 902 003204 012737 003204 001010  
 903  
 904  
 905 003212 013746 000004  
 906 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  
 CTCR: 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 \$SMR,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 \$PMRDN,\$PMRVEC ;;POWER FAILURE VECTOR  
 MOV \$340,\$PMRVEC+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



```

907 003224 012737 177570 001040      MOV    #DSWR,SWR      ;;SETUP FOR A HARDWARE SWICH REGISTER
908 003232 012737 177570 001042      MOV    #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
909 003240 022777 177777 175572      CMP    0-1,0SWR      ;;TRY TO REFERENCE HARDWARE SWR
910 003246 001012                    BNE    66$           ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
911                                ;;AND THE HARDWARE SWR IS NOT = 1
912 003250 000403                    BR     65$           ;;BRANCH IF NO TIMEOUT
913 003252 012716 003260 64$:      MOV    #65$,(SP)     ;;SET UP FOR TRAP RETURN
914 003256 000002                    RTI
915 003260 012737 000176 001040 65$:  MOV    #SWREG,SWR    ;;POINT TO SOFTWARE SWR
916 003266 012737 000174 001042      MOV    #DISPREG,DISPLAY
917 003274 012637 000004 66$:      MOV    (SP)+,#ERRVEC ;;RESTORE ERROR VECTOR
918
919 003300 005037 001106                    CLR    #PASS        ;;CLEAR PASS COUNT
920 003304 132737 000200 001121      BITB  #APTSIZE,#ENVM ;;TEST USER SIZE UNDER APT
921 003312 001403                    BEQ    67$           ;;YES,USE NON-APT SWITCH
922 003314 012737 001122 001040      MOV    #ISWREG,SWR  ;;NO,USE APT SWITCH REGISTER
923 003322
924                                67$:
.SBTL GET .ALUE FOR SOFTWARE SWITCH REGISTER
925 003322 005737 000042                    TST    #042         ;;ARE WE RUNNING UNDER XXDP/ACT?
926 003326 001012                    BNE    68$           ;;BRANCH IF YES
927 003330 123727 001120 000001      CMPB  #ENV,#1       ;;ARE WE RUNNING UNDER APT?
928 003336 001406                    BEQ    68$           ;;BRANCH IF YES
929 003340 023727 001040 000176      CMP    SWR,#SWREG   ;;SOFTWARE SWITCH REG SELECTED?
930 003346 001005                    BNE    69$           ;;BRANCH IF NO
931 003350 104406                    GTSWR
932 003352 000403                    BR     69$           ;;GET SOFT-SWR SETTINGS
933 003354 112737 000001 001034 68$:  MOVB  #1,#AUTOB     ;;SET AUTO-MODE INDICATOR
934 003362
935 003362 032777 000060 175450      BIT   #BIT4!BITS,0SWR ;;IS SLU1 TO BE TESTED
936 003370 001003                    BNE    2$           ;;IF EITHER BIT IS SET THEN DON'T TEST IT
937 003372 052737 000001 001156      BIS   #BIT0,#DEVM   ;;SET DEVICE FLAG TO TEST SLU1
938 003400 032777 000050 175432 2$:  BIT   #BIT3!BITS,0SWR ;;IS SLU2 TO BE TESTED
939 003406 001003                    BNE    3$           ;;IF EITHER BIT IS SET THEN DON'T TEST IT
940 003410 052737 000002 001156      BIS   #BIT1,#DEVM   ;;SET DEVICE FLAG TO TEST SLU2
941 003416 032777 000100 175414 3$:  BIT   #BIT6,0SWR    ;;IS LTC TO BE TESTED
942 003424 001003                    BNE    4$           ;;IF BIT IS SET THEN DON'T TEST IT.
943 003426 052737 000004 001156      BIS   #BIT2,#DEVM   ;;SET DEVICE FLAG TO TEST LTC
944 003434 032737 000001 001156 4$:  BIT   #BIT0,#DEVM   ;;IS SLU1 UNDER TEST
945 003442 001002                    BNE    TST1        ;;IF YES TEST XMIT REG. BEFORE TYPING TITLE
946 003444 000137 003700                    JMP    ID           ;;IF NO SKIP TESTS AND TYPE IT NOW

```

```

947
948
949
950
951
952 003450 000004
953 003452 012737 000001 001104
954 003460 013703 000004
955 003464 012737 003500 000004
956 003472 005777 177306
957 003476 000405
958 003500 022626
959 003502 004737 016004
960 003506 000001
961 003510 000000
962 003512 010337 000004
963
964
965
966
967
968
969 003516 000004
970 003520 012737 000002 001104
971 003526 013703 000004
972 003532 012737 003546 000004
973 003540 005777 177242
974 003544 000405
975 003546 022626
976 003550 004737 016004
977 003554 000002
978 003556 000000
979 003560 010337 000004
980
981
982
983
984
985 003564 000004
986 003566 012737 000003 001104
987 003574 032737 000001 001120
988 003602 001405
989 003604 005737 001106
990 003610 001402
991 003612 000137 004726
992 003616 005077 177164
993 003622 105777 177156
994 003626 100011
995
996
997 003630 005077 177152
998 003634 105777 177144
999 003640 100004
1000 003642 004737 016004
1001 003646 000003
1002 003650 000000

;*****
;TEST 1 TEST ABILITY TO ACCESS SLU1 TCSR
;*****
TST1: SCOPE
      MOV #1,TESTN ;SET TEST NUMBER IN APT MAIL BOX
      MOV R04,R3 ;SAVE TIMEOUT VECTOR
      MOV #1,R04 ;SET UP TIMEOUT VECTOR
      TST BCTCSR ;REFERENCE THE XMIT COMMAND/STATUS REG.
      BR 2 ; GO TO END OF TEST
10: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
      JSR PC,ATY4 ;ONLY REPORT A FATAL ERROR
      ;THE ERROR NUMBER (FROM APT LIST)
      HALT
20: MOV R3,R04 ;RESTORE TIMEOUT VECTOR

;*****
;TEST 2 TEST ABILITY TO ACCESS SLU1 TBUF
;*****
TST2: SCOPE
      MOV #2,TESTN ;SET TEST NUMBER IN APT MAIL BOX
      MOV R04,R3 ;SAVE TIMEOUT VECTOR
      MOV #1,R04 ;SET UP TIMEOUT VECTOR
      TST BCTBUF ;REFERENCE THE XMIT BUFFER
      BR 2 ;GO TO END OF TEST
10: CMP (SP)+,(SP)+ ;RESTORE SP AFTER TIMEOUT
      JSR PC,ATY4 ;ONLY REPORT A FATAL ERROR
      ;THE ERROR NUMBER (FROM APT LIST)
      HALT
20: MOV R3,R04 ;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,R04ENV ;ARE WE RUNNING UNDER APT
      BEQ 70 ;IF NO THEN DO TEST
      TST B04PASS ;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 BCTBUF ;LOAD XBUF
      TSTB BCTCSR ;CHECK DONE
      BPL 10 ;BR IF CLEAR
      ;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
      ; FIRST TEST TO FAIL
      CLR BCTBUF ;FILL DOUBLE BUFFER
      TSTB BCTCSR ;CHECK DONE
      BPL 10 ;BR IF CLEAR
      JSR PC,ATY4 ;ONLY REPORT A FATAL ERROR
      ;THE ERROR NUMBER (FROM APT LIST)
      HALT ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
  
```

1003	003652	005000			1#:	CLR	R0	;CLEAR TIMER
1004	003654	105777	177124		2#:	TSTB	@CTCSR	;CHECK FOR XMIT DONE
1005	003660	100407				BMI	ID	;IF DONE SETS, BR TO END OF TEST
1006	003662	005200				INC	R0	;INCREMENT TIMER
1007	003664	001373				BNE	2#	;BR IF TIMER NOT DONE
1008	003666	004737	016004			JSR	PC,#ATY4	;;ONLY REPORT A FATAL ERROR
1009	003672	000004				4		;;THE ERROR NUMBER (FROM APT LIST)
1010	003674	000000				HALT		
1011	003676	000416				BR	TST4	;BR TO NEXT TEST, AND SKIP THE TYPEOUT THAT FOLLOWS
1012								; BECAUSE OF THIS FAILURE
1013								
1014	003700	023737	000042	000046	ID:	CMP	@#42,@#46	;UNDER ACT11?
1015	003706	001412				BEQ	6#	;IF YES, SKIP IDENT. TYPEOUT
1016	003710	005737	001106			TST	#PASS	;IS THIS THE FIRST PASS?
1017	003714	001007				BNE	6#	;IF NOT BR TO NEXT TEST & SKIP THE IDENTIFICATION TYPEOU
1018	003716	005737	001110			TST	#DEVCT	;IS THIS THE FIRST SUBPASS?
1019	003722	001004				BNE	6#	;IF NOT, BR TO NEXT TEST
1020	003724	104401				TYPE		;TYPE PROGRAM IDENTIFICATION
1021	003726	030176				M1		
1022	003730	104401				TYPE		;TYPE NUMBER OF DEVICES UNDER TEST
1023	003732	030210				M2		
1024	003734				6#:			
1025								
1026								
1027								
1028								
1029	003734	000004						
1030	003736	012737	000004	001104	TST4:	SCOPE		
1031	003744	032737	000001	001156		MOV	#4,#TESTN	;;SET TEST NUMBER IN APT MAIL BOX
1032	003752	001002				BIT	@BIT0,@#DEVN	;DO THESE TESTS FOR THIS DEVICE?
1033	003754	000137	004726			BNE	99#	;IF YES CONTINUE WITH TESTS
1034	003760					JMP	SLU2RT	;IF NO GO TO START OF NEXT SET OF TESTS.
1035	003760	005077	177022		99#:			
1036	003764	105777	177014		1#:	CLR	@CTBUF	;LOAD TRANSMIT BUFFER
1037	003770	100375				TSTB	@CTCSR	;WAIT FOR DONE
1038	003772	005077	177010			BPL	1#	
1039	003776	000240				CLR	@CTBUF	;LOAD SECOND BUFFER
1040	004000	000005				NOP		
1041	004002	105777	176776			RESET		;CLEAR DONE WITH RESET
1042	004006	100401				TSTB	@CTCSR	;CHECK FOR DONE SET
1043						BMI	TST5	;BR TO NEXT TEST IF DONE SET
1044	004010	104005				ERROR	5	;TCSR "DONE" DOES NOT SET WITH RESET
1045								
1046								
1047								
1048								
1049								
1050								
1051	004012	000004						
1052	004014	012737	000005	001104	TST5:	SCOPE		
1053	004022	013703	000004			MOV	#5,#TESTN	;;SET TEST NUMBER IN APT MAIL BOX
1054	004026	012737	004042	000004		MOV	@#4,R3	;SAVE TIMEOUT VECTOR
1055	004034	005777	176740			MOV	@1#,@#4	;SET UP TIMEOUT VECTOR
1056	004040	000402				TST	@CRCSR	;ACCESS RCSR
1057						BR	2#	;BR TO END OF TEST
1058	004042	022626			1#:	CMP	(SP)+,(SP)+	;RESTORE SP AFTER TIMEOUT

```

1059 004044 104006
1060 004046 010337 000004
1061
1062
1063
1064
1065
1066 004052 000004
1067 004054 012737 000006 001104
1068 004062 013703 000004
1069 004066 012737 004102 000004
1070 004074 005777 176702
1071 004100 000402
1072
1073 004102 022626
1074 004104 104007
1075 004106 010337 000004
1076
1077
1078
1079
1080
1081
1082 004112 000004
1083 004114 012737 000007 001104
1084 004122 042777 000004 176654
1085 004130 032777 000004 176646
1086 004136 001404
1087 004140 004737 016004
1088 004144 000010
1089 004146 000000
1090 004150 052777 000004 176626
1091 004156 032777 000004 176620
1092 004164 001001
1093 004166 104011
1094 004170 042777 000004 176606
1095 004176 032777 000004 176600
1096 004204 001404
1097 004206 004737 016004
1098 004212 000012
1099 004214 000000
1100 004216 052777 000004 176560
1101 004224 000005
1102 004226 032777 000004 176550
1103 004234 001404
1104 004236 042777 000004 176540
1105 004244 104013
1106
1107
1108
1109
1110
1111
1112 004246 000004
1113 004250 012737 000010 001104
1114 004256 042777 000100 176520

```

```

ERROR 6 ;CAN NOT ACCESS RCSR
2: MOV R3,004 ;RESTORE TIMEOUT VECTOR

;*****
;*TEST 6 TEST ABILITY TO ACCESS SLU1 RBUF
;*****
TST6: SCOPE
MOV #6,TESTN ;SET TEST NUMBER IN APT MAIL BOX
MOV #04,R3 ;SAVE TIMEOUT VECTOR
MOV #1,004 ;SET UP TIMEOUT VECTOR
TST BCRBUF ;ACCESS RBUF
BR 2 ;BR TO END OF TEST

1: CMP (SP),(SP) ;RESTORE SP AFTER TIMEOUT
ERROR 7 ;CAN NOT ACCESS RBUF
2: MOV R3,004 ;RESTORE TIMEOUT VECTOR

;*****
;*TEST 7 TEST THAT SLU1 BIT2(MAINT. BIT) CAN BE SET & RESET
;*****
TST7: SCOPE
MOV #7,TESTN ;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT2,BC TCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
BIT #BIT2,BC TCSR ;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)
HALT
1: BIS #BIT2,BC TCSR ;SET BIT2 OF TCSR
BIT #BIT2,BC TCSR ;TEST FOR BIT2 SET
BNE 2 ;BR IF SET
ERROR 11 ;BIT2 OF TCSR WILL NOT SET
2: BIC #BIT2,BC TCSR ;CLEAR BIT2 OF TCSR
BIT #BIT2,BC TCSR ;TEST BIT2 CLEAR
BEQ 3 ;BR IF CLEAR
JSR PC,#ATY4 ;ONLY REPORT A FATAL ERROR
12 ;THE ERROR NUMBER (FROM APT LIST)
HALT ;BIT0 OF TCSR WILL NOT CLEAR
3: BIS #BIT2,BC TCSR ;SET BIT2 OF TCSR
RESET ;CLEAR BIT2 WITH RESET
BIT #BIT2,BC TCSR ;TEST FOR BIT2 CLEAR
BEQ TST10 ;IF CLEAR, GO TO NEXT TEST
BIC #BIT2,BC TCSR ;CLEAR BIT2, TO PRINT ERROR
ERROR 13 ;RESET DID NOT CLEAR BIT2 OF TCSR

;*****
;*TEST 10 TEST THAT SLU1 BIT6(XMIT INT EN) CAN BE SET & RESET
;*****
TST10: SCOPE
MOV #10,TESTN ;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,BC TCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED

```

```
1115 004264 017703 176536      MOV      @CTVECT,R3      ;SAVE XMIT VECTOR
1116 004270 012777 004320 176530  MOV      @1,@CTVECT     ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1117 004276 004737 015056      JSR      PC,WRPSW       ;SET PSW TO PRIORITY-7
1118 004302 000340      .WORD    340
1119 004304 032777 000100 176472  BIT      @BIT6,@CTCSR   ;TEST BIT6 OF TCSR
1120 004312 001404      BEQ     2#              ;BR IF ZERO
1121 004314 104014      ERROR   14
1122
1123 004316 000402      BR      2#              ;BIT6 IN TCSR NOT CLEAR AFTER RESET
1124
1125 004320 022626      1# :    CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1126 004322 104015      ERROR   15
1127
1128
1129 004324 052777 000100 176452  2# :    BIS      @BIT6,@CTCSR ;SET BIT6 OF TCSR
1130 004332 032777 000100 176444  BIT      @BIT6,@CTCSR   ;TEST BIT6 OF TCSR
1131 004340 001001      BNE     3#              ;BR, IF SET
1132
1133 004342 104016      ERROR   16
1134
1135
1136 004344 042777 000100 176432  3# :    BIC      @BIT6,@CTCSR ;CLEAR BIT6 OF TCSR
1137 004352 032777 000100 176424  BIT      @BIT6,@CTCSR   ;TEST BIT6 OF TCSR
1138 004360 001401      BEQ     4#              ;BR IF CLEAR
1139 004362 104017      ERROR   17
1140
1141
1142 004364      4# :
1143 004364 052777 000100 176412  BIS      @BIT6,@CTCSR   ;SET BIT6 OF TCSR
1144 004372 000005      RESET   ;CLEAR BIT6 WITH RESET
1145 004374 032777 000100 176402  BIT      @BIT6,@CTCSR   ;TEST BIT6 OF TCSR
1146 004402 001401      BEQ     5#              ;BR IF CLEAR
1147
1148 004404 104020      ERROR   20
1149
1150 004406 010377 176414      5# :    MOV      R3,@CTVECT  ;RESTORE XMIT VECTOR
1151
1152
1153
1154
1155
1156 004412 000004      TST11: SCOPE
1157 004414 012737 000011 001104  MOV      @11,@TESTN    ;;SET TEST NUMBER IN APT MAIL BOX
1158 004422 042777 000100 176350  BIC      @BIT6,@RCRCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1159 004430 017703 176366      MOV      @RCRVECT,R3   ;SAVE RECEIVE VECTOR
1160 004434 012777 004464 176360  MOV      @1,@RCRVECT   ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1161 004442 004737 015056      JSR      PC,WRPSW       ;SET PSW TO PRIORITY-7
1162 004446 000340      .WORD    340
1163 004450 032777 000100 176322  BIT      @BIT6,@RCRCSR ;TEST BIT6 OF RCSR
1164 004456 001404      BEQ     2#              ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1165 004460 104021      ERROR   21
1166
1167 004462 000402      BR      2#
1168
1169 004464 022626      1# :    CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1170 004466 104022      ERROR   22
```

```

1171                                     ;RCVR INTERRUPT WITH PRIORITY=7
1172
1173 004470 052777 000100 176302 2$: BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
1174 004476 032777 000100 176274 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1175 004504 001001 BNE 3$ ;BR, IF SET
1176
1177 004506 104023 ERROR 23 ;CANNOT SET BIT6 OF RCSR
1178
1179
1180 004510 042777 000100 176262 3$: BIC #BIT6,@RCSR ;CLEAR BIT6 OF RCSR
1181 004516 032777 000100 176254 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1182 004524 001401 BEQ 4$ ;BR, IF CLEAR
1183
1184 004526 104024 ERROR 24 ;CANNOT CLEAR BIT6 OF RCSR
1185
1186
1187 004530 4$: BIS #BIT6,@RCSR ;SET BIT6 OF RCSR
1188 004530 052777 000100 176242 RESET ;CLEAR BIT6 OF RCSR WITH RESET
1189 004536 000005 BIT #BIT6,@RCSR ;TEST BIT6 OF RCSR
1190 004540 032777 000100 176232 BEQ 5$ ;BR, IF CLEAR
1191 004546 001401
1192
1193 004550 104025 ERROR 25 ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1194
1195 004552 010377 176244 5$: MOV R3,@RVECT ;RESTORE RECEIVE VECTOR
1196
1197
1198
1199
1200 ;:*****
1201 ;*TEST 12 TEST THAT SLU1 RCVR DONE (7) SET & CLEAR PROPERLY
1202 ;:*****
1203 004556 000004 TST12: SCOPE
1204 004560 012737 000012 001104 MOV #12,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1205 004566 052777 000004 176210 BIS #BIT2,@CTCSR ;TURN ON INTERNAL MAINTENANCE WRAP AROUND
1206 004574 005000 CLR R0 ;CLEAR TIMER
1207 004576 005077 176204 CLR @CTBUF ;LOAD TRANSMIT BUFFER
1208 004602 105777 176172 CWDONE: TSTB @RCSR ;CHECK FOR RECEIVER DONE
1209 004606 100406 BMI 1$ ;BR, IF DONE
1210 004610 005200 INC R0 ;INCREMENT TIMER, IF NOT DONE
1211 004612 001373 BNE CWDONE ;CONTINUE WAIT IF TIME REMAINS
1212 004614 042777 000004 176162 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1213 004622 104026 ERROR 26 ;RECEIVER DONE NEVER SET
1214
1215 004624 000005 1$: RESET ;CLEAR DONE WITH RESET
1216 004626 105777 176146 TSTB @RCSR ;CHECK FOR DONE CLEAR
1217 004632 001404 BEQ 2$
1218
1219 004634 042777 000004 176142 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1220 004642 104027 ERROR 27 ;RESET DID NOT CLEAR RCVR DONE
1221
1222
1223 004644 042777 000004 176126 2$: BIC #BIT2,@RCSR ;CLEAR MAINTENANCE BIT
1224
1225
1226 ;:*****

```

```

1227
1228
1229 004652 000004
1230 004654 012737 000013 001104
1231 004662 000005
1232 004664 052777 000004 176112
1233 004672 005077 176110
1234 004676 105777 176076
1235 004702 100375
1236 004704 017700 176072
1237 004710 042777 000004 176066
1238 004716 105777 176056
1239 004722 001401
1240 004724 104030
1241

```

```

:*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 @CTBUF ;LOAD TRANSMITTER
1$: TSTB @CRCSR ;WAIT FOR RECEIVER DONE
BPL 1$
MOV @RBUF,R0 ;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

```

```

1242 004726
1243
1244
1245
1246
1247 004726 000004
1248 004730 012737 000014 001104
1249 004736 032737 000002 001156
1250 004744 001002
1251 004746 000137 006152
1252 004752
1253 004752 013703 000004
1254 004756 012737 004772 000004
1255 004764 005777 176004
1256 004770 000402
1257 004772 022626
1258 004774 104031
1259 004776 010337 000004
1260
1261
1262
1263
1264
1265
1266 005002 000004
1267 005004 012737 000015 001104
1268 005012 013703 000004
1269 005016 012737 005032 000004
1270 005024 005777 175746
1271 005030 000402
1272 005032 022626
1273 005034 104032
1274 005036 010337 000004
1275
1276
1277
1278
1279
1280 005042 000004
1281 005044 012737 000016 001104
1282 005052 032737 000001 001120
1283 005060 001403
1284 005062 005737 001106
1285 005066 001022
1286 005070
1287 005070 005077 175702
1288 005074 105777 175674
1289 005100 100006
1290
1291
1292 005102 005077 175670
1293 005106 105777 175662
1294 005112 100001
1295 005114 104033
1296 005116 005000
1297 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,#DEVN ;DO THESE TESTS FOR THIS DEVICE?
BNE 991 ;IF YES CONTINUE WITH TESTS
JMP LTRT ;IF NO GO TO START OF NEXT SET OF TESTS.

991:
MOV #04,R3 ;SAVE TIMEOUT VECTOR
MOV #18,#04 ;SET UP TIMEOUT VECTOR
TST #TCSR ;REFERENCE THE XMIT COMMAND/STATUS REG.
BR 21 ;GO TO END OF TEST
11: CMP (SP),.(SP) ;RESTORE SP AFTER TIMEOUT
ERROR 31 ;XMIT CSR ADDRESS DOES NOT RETURN SSYNC
21: MOV R3,#04 ;RESTORE TIMEOUT VECTOR

;*****
;TEST 15 TEST ABILITY TO ACCESS SLU2 TBUF
;*****
TST15: SCOPE
MOV #15,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
MOV #04,R3 ;SAVE TIMEOUT VECTOR
MOV #18,#04 ;SET UP TIMEOUT VECTOR
TST #TBUF ;REFERENCE THE XMIT BUFFER
BR 21 ;GO TO END OF TEST
11: CMP (SP),.(SP) ;RESTORE SP AFTER TIMEOUT
ERROR 32 ;XMIT BUFFER ADDRESS DOES NOT RETURN SSYNC
21: MOV R3,#04 ;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 PUNNING UNDER APT
BEQ 701 ;IF NO THEN DO TEST
TST #PASS ;IS THIS FIRST PASS
BNE TST17 ;IF NO THEN SKIP TO NEXT TEST

701:
CLR #TBUF ;LOAD XBUF
TSTB #TCSR ;CHECK DONE
BPL 11 ;BR IF CLEAR
;FILL SECOND BUFFER, BECAUSE REFRESH COULD CAUSE
; FIRST TEST TO FAIL
CLR #TBUF ;FILL DOUBLE BUFFER
TSTB #TCSR ;CHECK DONE
BPL 11 ;BR IF CLEAR
ERROR 33 ;TCSR "DONE" NOT CLEARED WITH TBUF FULL
11: CLR R0 ;CLEAR TIMER
21: TSTB #TCSR ;CHECK FOR XMIT DONE

```



1298	005124	100403			BMI	TST17		;IF DONE SETS, BR TO NEXT TEST
1299	005126	005200			INC	R0		;INCREMENT TIMER
1300	005130	001373			BNE	Z1		;BR IF TIMER NOT DONE
1301	005132	104034			ERROR	34		;XMIT DONE BIT DOES NOT RESET AFTER TRANSMIT

```

1302
1303
1304
1305 ;*****
; *TEST 17 TEST THAT SLU2 TCSR "DONE" SETS WITH RESET
1306 ;*****

```

1307	005134	000004			TST17:	SCOPE		
1308	005136	012737	000017	001104	MOV	#17,#TESTN		;SET TEST NUMBER IN APT MAIL BOX
1309	005144	032737	000001	001120	BIT	#1,#ENV		;ARE WE RUNNING UNDER APT
1310	005152	001403			BEQ	Z01		;IF NO THEN DO TEST
1311	005154	005737	001106		TST	#0,PASS		;IS THIS FIRST PASS
1312	005160	001015			BNE	TST20		;IF NO THEN SKIP TO NEXT TEST
1313	005162				701:			
1314	005162	005077	175610		CLR	#TBUF		;LOAD TRANSMIT BUFFER
1315	005166	105777	175602		11:	TSTB	#TCSR	;WAIT FOR DONE
1316	005172	100375			BPL	Z1		
1317	005174	005077	175576		CLR	#TBUF		;LOAD SECOND BUFFER
1318	005200	000240			NOP			
1319	005202	000005			RESET			;CLEAR DONE WITH RESET
1320	005204	105777	175564		TSTB	#TCSR		;CHECK FOR DONE SET
1321	005210	100401			BMI	TST20		;BR TO NEXT TEST IF DONE SET
1322								
1323	005212	104035			ERROR	35		;TCSR "DONE" DOES NOT SET WITH RESET

```

1324
1325
1326
1327 ;*****
; *TEST 20 TEST ABILITY TO ACCESS SLU2 RCSR
1328 ;*****
1329

```

1330	005214	000004			TST20:	SCOPE		
1331	005216	012737	000020	001104	MOV	#20,#TESTN		;SET TEST NUMBER IN APT MAIL BOX
1332	005224	013703	000004		MOV	#04,R3		;SAVE TIMEOUT VECTOR
1333	005230	012737	005244	000004	MOV	#11,#04		;SET UP TIMEOUT VECTOR
1334	005236	005777	175526		TST	#RCSR		;ACCESS RCSR
1335	005242	000402			BR	Z1		;BR TO END OF TEST
1336								
1337	005244	022626			11:	CMP	(SP)+,(SP)+	;RESTORE SP AFTER TIMEOUT
1338	005246	104036			ERROR	36		;CAN NOT ACCESS RCSR
1339	005250	010337	000004		21:	MOV	R3,#04	;RESTORE TIMEOUT VECTOR

```

1340
1341
1342
1343 ;*****
; *TEST 21 TEST ABILITY TO ACCESS SLU2 RBUF
1344 ;*****
1345

```

1346	005254	000004			TST21:	SCOPE		
1347	005256	012737	000021	001104	MOV	#21,#TESTN		;SET TEST NUMBER IN APT MAIL BOX
1348	005264	013703	000004		MOV	#04,R3		;SAVE TIMEOUT VECTOR
1349	005270	012737	005304	000004	MOV	#11,#04		;SET UP TIMEOUT VECTOR
1350	005276	005777	175470		TST	#RBUF		;ACCESS RBUF
1351	005302	000402			BR	Z1		;BR TO END OF TEST
1352								
1353	005304	022626			11:	CMP	(SP)+,(SP)+	;RESTORE SP AFTER TIMEOUT

1354 005306 104037  
 1355 005310 010337 000004  
 1356  
 1357  
 1358  
 1359  
 1360  
 1361  
 1362 005314 000004  
 1363 005316 012737 000022 001104  
 1364 005324 042777 000001 175442  
 1365 005332 032777 000001 175434  
 1366 005340 001401  
 1367 005342 104040  
 1368 005344 052777 000001 175422  
 1369 005352 032777 000001 175414  
 1370 005360 001001  
 1371 005362 104041  
 1372 005364 042777 000001 175402  
 1373 005372 032777 000001 175374  
 1374 005400 001401  
 1375 005402 104042  
 1376 005404  
 1377 005404 032737 000001 001120  
 1378 005412 001403  
 1379 005414 005737 001106  
 1380 005420 001014  
 1381 005422  
 1382 005422 052777 000001 175344  
 1383 005430 000005  
 1384 005432 032777 000001 175334  
 1385 005440 001404  
 1386 005442 042777 000001 175324  
 1387 005450 104043  
 1388  
 1389  
 1390  
 1391  
 1392  
 1393 005452 000004  
 1394 005454 012737 000023 001104  
 1395 005462 042777 000100 175304  
 1396 005470 017703 175322  
 1397 005474 012777 005524 175314  
 1398 005502 004737 015056  
 1399 005506 000340  
 1400 005510 032777 000100 175256  
 1401 005516 001404  
 1402 005520 104044  
 1403  
 1404 005522 000402  
 1405  
 1406 005524 022626  
 1407 005526 104045  
 1408  
 1409

```

ERROR 37 ;CAN NOT ACCESS RBUF
MOV R3,004 ;RESTORE TIMEOUT VECTOR

;*****
;TEST 22 TEST SLU2 BIT0(BREAK BIT) CAN BE SET & CLEARED & RESET
;*****
TST22: SCOPE
MOV #22,TESTN ;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT0,BTCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
BIT #BIT0,BTCSR ;CHECK BIT0 OF TCSR CLEAR
BEQ 10 ;BR IF CLEAR
ERROR 40 ;BIT0 WAS NOT CLEAR AFTER RESET
10: BIS #BIT0,BTCSR ;SET BIT0 IN TCSR
BIT #BIT0,BTCSR ;TEST BIT0 OF TCSR
BNE 20 ;BR IF SET
ERROR 41 ;BIT0 OF TCSR WILL NOT SET
20: BIC #BIT0,BTCSR ;CLEAR BIT0 OF TCSR
BIT #BIT0,BTCSR ;TEST BIT0 OF TCSR
BEQ 30
ERROR 42 ;BIT0 OF TCSR WILL NOT CLEAR
30: 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
70: BIS #BIT0,BTCSR ;SET BIT0 IN TCSR
RESET ;CLEAR BIT0 WITH RESET
BIT #BIT0,BTCSR ;TEST BIT0 CLEAR
BEQ TST23 ;BR IF CLEAR
BIC #BIT0,BTCSR ;CLEAR BIT0, TO PRINT ERROR
ERROR 43 ;RESET DID NOT CLEAR BIT0 OF TCSR

;*****
;TEST 23 TEST THAT SLU2 BIT6(XMIT INT EN) CAN BE SET & RESET
;*****
TST23: SCOPE
MOV #23,TESTN ;SET TEST NUMBER IN APT MAIL BOX
BIC #BIT6,BTCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
MOV #TVECT,R3 ;SAVE XMIT VECTOR
MOV #10,TVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
JSR PC,WRPSW ;SET PSW TO PRIORITY-7
.WORD 340
BIT #BIT6,BTCSR ;TEST BIT6 OF TCSR
BEQ 20 ;BR IF ZERO
ERROR 44 ;BIT6 IN TCSR NOT CLEAR AFTER RESET
BR 20
10: CMP (SP),.(SP) ;RESTORE SP AFTER INTERRUPT
ERROR 45 ;XMIT INTERRUPT OCCURRED PRIORITY-7

```

```

1410 005530 052777 000100 175236 2#: BIS #BIT6,BTCR ;SET BIT6 OF TCSR
1411 005536 032777 000100 175230 BIT #BIT6,BTCR ;TEST BIT6 OF TCSR
1412 005544 C01001 BNE 3# ;BR, IF SET
1413
1414 005546 104046 ERROR 46
1415 ;CANNOT SET BIT6 OF TCSR
1416
1417 005550 042777 000100 175216 3#: BIC #BIT6,BTCR ;CLEAR BIT6 OF TCSR
1418 005556 032777 000100 175210 BIT #BIT6,BTCR ;TEST BIT6 OF TCSR
1419 005564 001401 BEQ 4# ;BR IF CLEAR
1420 005566 104047 ERROR 47
1421 ;CANNOT CLEAR BIT6 OF TCSR
1422
1423 005570 032737 000001 001120 4#: BIT #1,B#ENV ;ARE WE RUNNING UNDER APT
1424 005576 001403 BEQ 70# ;IF NO THEN DO TEST
1425 005600 005737 001106 TST B#PASS ;IS THIS FIRST PASS
1426 005604 001011 BNE 5# ;IF NO THEN SKIP TO END OF TEST
1427 005606
1428 005606 052777 000100 175160 70#: BIS #BIT6,BTCR ;SET BIT6 OF TCSR
1429 005614 000005 RESET ;CLEAR BIT6 WITH RESET
1430 005616 032777 000100 175150 BIT #BIT6,BTCR ;TEST BIT6 OF TCSR
1431 005624 001401 BEQ 5# ;BR IF CLEAR
1432
1433 005626 104050 ERROR 50
1434 ;CANNOT CLEAR BIT6 OF TCSR WITH RESET
1435 005630 010377 175162 5#: MOV R3,BTVECT ;RESTORE XMIT VECTOR
1436
1437
1438 ;*****
1439 ;*TEST 24 TEST THAT SLU2 BIT6 OF RCSR CAN BE SET & RESET
1440 ;*****
1441 005634 000004 TST24: SCOPE
1442 005636 012737 MOV #24,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1443 005644 042777 000100 175116 BIC #BIT6,BRCSR ;MAKE SURE BIT UNDER TEST IS INITIALIZED
1444 005652 017703 175134 MOV BRVECT,R3 ;SAVE RECEIVE VECTOR
1445 005656 012777 005706 175126 MOV #1#,BRVECT ;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1446 005664 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY-7
1447 005670 000340 .WORD 340
1448 005672 032777 000100 175070 BIT #BIT6,BRCSR ;TEST BIT6 OF RCSR
1449 005700 001404 BEQ 2#
1450 005702 104051 ERROR 51 ;BIT6 OF RCSR NOT CLEAR AFTER RESET
1451
1452 005704 000402 BR 2#
1453
1454 005706 022626 1#: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1455 005710 104052 ERROR 52 ;RCVR INTERRUPT WITH PRIORITY-7
1456
1457
1458 005712 052777 000100 175050 2#: BIS #BIT6,BRCSR ;SET BIT6 OF RCSR
1459 005720 032777 000100 175042 BIT #BIT6,BRCSR ;TEST BIT6 OF RCSR
1460 005726 001001 BNE 3# ;BR, IF SET
1461
1462 005730 104053 ERROR 53 ;CANNOT SET BIT6 OF RCSR
1463
1464
1465 005732 042777 000100 175030 3#: BIC #BIT6,BRCSR ;CLEAR BIT6 OF RCSR

```

```

1466 005740 032777 000100 175022      BIT    #BIT6,RCSR      ;TEST BIT6 OF RCSR
1467 005746 001401                      BEQ    4#              ;BR, IF CLEAR
1468
1469 005750 104054                      ERROR  54
1470
1471
1472 005752 032737 000001 001120 4# :   BIT    #1,#ENV        ;ARE WE RUNNING UNDER APT
1473 005760 001403                      BEQ    70#            ;IF NO THEN DO TEST
1474 005762 005737 001106                      TST    #PASS         ;IS THIS FIRST PASS
1475 005766 001011                      BNE    5#            ;IF NO THEN SKIP TO END OF TEST
1476 005770
1477 005770 052777 000100 174772 70# :   BIS    #BIT6,RCSR      ;SET BIT6 OF RCSR
1478 005776 000005                      RESET  ;CLEAR BIT6 OF RCSR WITH RESET
1479 006000 032777 000100 174762      BIT    #BIT6,RCSR      ;TEST BIT6 OF RCSR
1480 006006 001401                      BEQ    5#            ;BR, IF CLEAR
1481
1482 006010 104055                      ERROR  55
1483
1484 006012 010377 174774 5# :   MOV    R3,BRVECT      ;CANNOT CLEAR BIT6 OF RCSR WITH RESET
1485
1486
1487
1488
1489
1490
1491 006016 000004
1492 006020 012737 000025 001104
1493 006026 005077 174740
1494 006032 005000
1495 006034 005077 174736
1496 006040 105777 174724      WDONE: TSTB   RCSR      ;CHECK FOR RECEIVER DONE
1497 006044 100403                      BMI    1#            ;BR, IF DONE
1498 006046 005200                      INC    R0             ;INCREMENT TIMER, IF NOT DONE
1499 006050 001373                      BNE    WDONE         ;CONTINUE WAIT IF TIME REMAINS
1500 006052 104056                      ERROR  56
1501
1502
1503 006054 032737 000001 001120 1# :   BIT    #1,#ENV        ;ARE WE RUNNING UNDER APT
1504 006062 001403                      BEQ    70#            ;IF NO THEN DO TEST
1505 006064 005737 001106                      TST    #PASS         ;IS THIS FIRST PASS
1506 006070 001005                      BNE    2#            ;IF NO THEN SKIP TO END OF TEST
1507 006072
1508 006072 000005                      RESET  ;CLEAR DONE WITH RESET
1509 006074 105777 174670      WDONE: TSTB   RCSR      ;CHECK FOR DONE CLEAR
1510 006100 001401                      BEQ    2#
1511
1512 006102 104057                      ERROR  57
1513
1514 006104 005777 174662 2# :   TST    RBUF          ;RESET DID NOT CLEAR RCVR DONE
1515
1516
1517
1518
1519
1520
1521 006110 000004

```

```

;*****
;*TEST 26      TEST SLU2 THAT READING RBUF CLEARS RECEIVER DONE
;*****
TST26: SCOPE

```

```

1522 006112 012737 000026 001104      MOV    #26,#TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1523 006120 005077 174646              CLR    RBUF           ;;INITIALIZE REGISTER BEFORE TESTING,
1524 006124 005077 174646              CLR    RBUF           ;;LOAD TRANSMITTER
1525 006130 105777 174634      1#:   TSTB  BRCSR       ;;WAIT FOR RECEIVER DONE
1526 006134 100375              BPL    1#
1527 006136 017700 174630      MOV    RBUF,R0        ;;READ RECEIVE BUFFER
1528 006142 105777 174622      TSTB  BRCSR         ;;CHECK FOR RECEIVE DONE CLEAR
1529 006146 001401              BEQ    TST27          ;;BR, IF CLEAR TO NEXT TEST
1530 006150 104060      ERROR  60
1531                                     ;;READING RBUF DID NOT CLEAR RCVR DONE
1532 006152      LTCRT:
1533
1534      ;;*****
1535      ;;*TEST 27      TEST ABILITY TO ACCESS LKS
1536      ;;*****
1537 006152 000004      TST27: SCOPE
1538 006154 012737 000027 001104      MOV    #27,#TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1539 006162 032737 000004 001156      BIT    #BIT2,#DEVH     ;;DO THESE TESTS FOR THIS DEVICE?
1540 006170 001002              BNE    99#            ;;IF YES CONTINUE WITH TESTS
1541 006172 000137 006524              JMP    UNIQUE         ;;IF NO GO TO START OF NEXT SET OF TESTS.
1542 006176
1543 006176 013703 000004      99#:   MOV    R4,R3        ;;SAVE TIMEOUT VECTOR
1544 006202 012737 006216 000004      MOV    #1#,#R4        ;;SET UP TIMEOUT VECTOR
1545 006210 005777 174574      TST    BLKS           ;;ACCESS LKS
1546 006214 000402              BR     2#            ;;NO TIMEOUT - BR TO END OF TEST
1547
1548 006216 022626      1#:   CMP    (SP)+,(SP)+   ;;RESTORE SP AFTER TIMEOUT
1549 006220 104061      ERROR  61            ;;CAN NOT ACCESS LKS
1550
1551 006222 010337 000004      2#:   MOV    R3,#R4        ;;RESTORE TIMEOUT VECTOR
1552
1553      ;;*****
1554      ;;*TEST 30      TEST THAT BIT6 OF LKS CAN BE SET & RESET
1555      ;;*****
1556 006226 000004      TST30: SCOPE
1557 006230 012737 000030 001104      MOV    #30,#TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1558 006236 042777 000100 174544      BIC    #BIT6,BLKS     ;;MAKE SURE BIT UNDER TEST IS INITIALIZED BEFORE TESTING
1559 006244 017703 174562              MOV    RTRCVT,R3      ;;SAVE LINE CLOCK VECTOR
1560 006250 012777 006300 174554      MOV    #1#,RTRCVT     ;;SET UP INTERRUPT VECTOR FOR ERROR REPORT
1561 006256 004737 015056              JSR    PC,WIPSW       ;;SET PSW TO PRIORITY 7
1562 006262 000340              .WORD 340
1563 006264 032777 000100 174516      BIT    #BIT6,BLKS     ;;TEST BIT6 OF LKS
1564 006272 001404              BEQ    2#            ;;BIT6 OF LKS NOT CLEAR AFTER RESET
1565 006274 104062      ERROR  62
1566
1567 006276 000402              BR     2#
1568
1569 006300 022626      1#:   CMP    (SP)+,(SP)+   ;;RESTORE SP AFTER INTERRUPT
1570 006302 104063      ERROR  63            ;;LKS INTERRUPT WITH PRIORITY-7
1571
1572
1573 006304 052777 000100 174476      2#:   BIS    #BIT6,BLKS   ;;SET BIT6 OF LKS
1574 006312 032777 000100 174470      BIT    #BIT6,BLKS     ;;TEST BIT6 OF LKS
1575 006320 001001              BNE    3#            ;;BR IF SET
1576
1577 006322 104064      ERROR  64

```

```
1578 ;CANNOT SET BIT6 OF LK
1579
1580 006324 042777 000100 174456 3#: BIC #BIT6,BLKS ;CLEAR BIT6 OF LKS
1581 006332 032777 000100 174450 BIT #BIT6,BLKS ;TEST BIT6 OF LK
1582 006340 001401 BEQ 4#
1583 006342 104065 ERROR 65
1584 ;CANNOT CLEAR BIT6 OF LKS
1585 006344 032737 000001 001120 4#: BIT #1,#ENV ;ARE WE RUNNING UNDER APT
1586 006352 001403 BEQ 70# ;IF NO THEN DO TEST
1587 006354 005737 001106 TST #PASS ;IS THIS FIRST PASS
1588 006360 001011 BNE 5# ;IF NO THEN SKIP TO END OF TEST
1589 006362
1590 006362 052777 000100 174420 70#: BIS #BIT6,BLKS ;SET BIT6 OF LKS
1591 006370 000005 RESET ;CLEAR BIT6 OF LKS WITH RESET
1592 006372 032777 000100 174410 BIT #BIT6,BLKS ;TEST BIT6 OF LKS
1593 006400 001401 BEQ 5# ;BR IF CLEAR
1594
1595 006402 104066 ERROR 66
1596 ;CANNOT CLEAR BIT6 OF LKS WITH RESET
1597 006404 010377 174422 5#: MOV R3,BRTCVT ;RESTORE LINE CLOCK VECTOR
1598
1599
1600
1601
1602 ;:*****
1603 ;*TEST 31 TEST THAT BIT7 OF LKS SETS & CAN BE CLEARED
1604 ;:*****
1605 006410 000004 TST31: SCOPE
1606 006412 012737 000031 001104 MOV #31,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
1607 006420 032737 000001 001120 BIT #1,#ENV ;ARE WE RUNNING UNDER APT
1608 006426 001403 BEQ 70# ;IF NO THEN DO TEST
1609 006430 005737 001106 TST #PASS ;IS THIS FIRST PASS
1610 006434 001033 BNE TST32 ;IF NO THEN SKIP TO NEXT TEST
1611 006436
1612 006436 000005 70#: RESET ;CLEAR EVERYTHING & SET BIT7 OF LKS
1613 006440 105777 174344 1#: TSTB BLKS ;TEST FOR BIT7 OF LKS
1614 006444 100401 BMI 2# ;BR IF SET
1615
1616 006446 104067 ERROR 67 ;BIT7 OF LKS DID NOT SET WITH RESET
1617
1618 006450 042777 000200 174332 2#: BIC #BIT7,BLKS ;CLEAR BIT7 OF LKS
1619 006456 032777 000200 174324 BIT #BIT7,BLKS ;TEST BIT7 OF LKS
1620 006464 001410 BEQ 3#
1621 006466 042777 000200 174314 BIC #BIT7,BLKS ;TRY ONE MORE TIME BECAUSE THE CLOCK
1622 006474 032777 000200 174306 BIT #BIT7,BLKS ;MAY HAVE SET IMMEDIATELY AFTER THE FIRST CLEAR
1623 006502 001401 BEQ 3#
1624
1625 006504 104070 ERROR 70 ;CAN NOT CLEAR BIT7 OF LKS
1626
1627 006506 005000 3#: CLR R0 ;CLEAR TIMER
1628 006510 105777 174274 CONT: TSTB BLKS ;TEST FOR BIT7 OF LKS
1629 006514 100403 BMI TST32 ;BR, IF SET
1630 006516 005200 INC R0 ;INCREMENT TIMER
1631 006520 001373 BNE CONT ;CONTINUE UNTIL TIME EXPIRES
1632
1633 006522 104071 ERROR 71 ;BIT7 OF LKS DOES NOT SET
```

```

1634
1635 006524
1636
1637
1638
1639
1640 006524 000004
1641 006526 012737 000032 001104
1642 006534 032737 000001 001120
1643 006542 001403
1644 006544 005737 001106
1645 006550 001053
1646 006552
1647 006552 012737 000340 177776
1648
1649 006560 012700 002770
1650 006564 012703 002770
1651
1652 006570 012701 000011
1653 006574 005033
1654 006576 077102
1655 006600 012770 000100 000000
1656 006606 012701 002770
1657 006612 012702 000011
1658 006616 032731 000100
1659 006622 001006
1660 006624 077204
1661 006626 020027 003010
1662 006632 001422
1663 006634 005030
1664
1665 006636 00: 752
1666 006640 021041
1667 006642 001413
1668 006644 011037 001020
1669 006650 011137 001022
1670 006654 017037 000000 001024
1671 006662 017137 000000 001026
1672 006670 104072
1673
1674 006672 062701 000002
1675 006676 000752
1676 006700
1677
  
```

```

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
      MOV      #RCSR,R0        ;GET LOCATION OF FIRST REGISTER ADDRESS
1$:  MOV      #RCSR,R3        ;MAKE R3 POINT TO LOCATION OF FIRST
      ;REGISTER ADDRESS
      MOV      #11,R1          ;SET LOOP COUNTER TO CLEAR ALL REG.
2$:  CLR      @R3)             ;CLEAR A REGISTER
      SOB     R1,2$           ;LOOP UNTIL ALL REGISTERS CLEARED
      MOV      @BIT6,@R0)     ;SET TEST BIT IN DEVICE REGISTERS
      MOV      #RCSR,R1        ;GET LOCATION OF FIRST REGISTER ADDRESS
      MOV      #11,R2          ;SET UP TEST LOOP COUNTER
3$:  BIT      @BIT6,@R1)     ;IS TEST BIT SET IN THIS REGISTER
      BNE     5$              ;IF YES GO SEE IF THERE IS AN ERROR
4$:  SOB     R2,3$           ;LOOP UNTIL ALL REGISTER CHECKED
      CMP     R0,#LKS          ;ARE WE DONE TESTING
      BEQ     7$              ;IF YES GO TO NEXT TEST
      CLR     @R0)             ;CLEAR REGISTER JUST TESTED AND POINT
      ;TO NEXT ONE
      BR     1$               ;CONTINUE TESTING
5$:  CMP     (R0),-(R1)        ;DID WE COMPARE THE REGISTER TO ITSELF?
      BEQ     6$              ;IF YES GET OVER ERROR CALL
      MOV     (R0),#GDADR      ;IF NO SET UP ERROR INFORMATION
      MOV     (R1),#BDADR
      MOV     @R0),#GDDAT
      MOV     @R1),#BDDAT
      ERROR  72
      ;WRITE TO 1 INTERNAL ADDRESS MODIFIED
      ;ANOTHER SO ADDRESS NOT UNIQUE
6$:  ADD     #2,R1            ;RESTORE POINTER
7$:  BR     4$                ;GET BACK IN TEST LOOP
  
```

```

1678 006700
1679
1680
1681
1682
1683 006700 000004
1684 006702 012737 000033 001104
1685 006710 032737 000001 001120
1686 006716 001405
1687 006720 005737 001106
1688 006724 001402
1689 006726 000137 010600
1690 006732
1691 006732 032737 000001 001156
1692 006740 001002
1693 006742 000137 010600
1694 006746
1695 006746 000005
1696 006750 042777 000100 174026
1697 006756 017703 174044
1698 006762 012777 007006 174036
1699 006770 105777 174010
1700 006774 100375
1701 006776 004737 015056
1702 007002 000140
1703 007004 000402
1704
1705 007006 022626
1706 007010 104074
1707
1708 007012 012777 007032 174006
1709 007020 052777 000100 173756
1710 007026 000240
1711
1712 007030 104075
1713
1714 007032 042777 000100 173744
1715 007040 022626
1716 007042 010377 173760
1717
1718
1719
1720
1721
1722 007046 000004
1723 007050 012737 000034 001104
1724 007056 042777 000100 173720
1725 007064 004737 015056
1726 007070 000340
1727 007072 017703 173730
1728 007076 012777 007124 173722
1729 007104 105777 173674
1730 007110 100375
1731 007112 052777 000100 173664
1732 007120 000240
1733 007122 000402
  
```

SLU1IT:

```

;*****
;*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,#DEVH ;DO THESE TESTS FOR THIS DEVICE?
BNE 99# ;IF 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,#RPSW ;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,#RPSW ;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
  
```



```

1734
1735 007124 022626
1736 007126 104076
1737
1738 007130 042777 000100 173646
1739 007136 012777 007156 173662
1740 007144 004737 015056
1741 007150 000140
1742 007152 000240
1743 007154 000402
1744
1745 007156 022626
1746 007160 104077
1747
1748 007162 010377 173640
1749
1750
1751
1752
1753
1754 007166 000004
1755 007170 012737 000035 001104
1756 007176 042777 000100 173600
1757 007204 017703 173616
1758 007210 017704 173614
1759 007214 012777 007256 173604
1760 007222 012777 000340 173600
1761 007230 004737 015056
1762 007234 000140
1763 007236 105777 173542
1764 007242 100375
1765 007244 052777 000100 173532
1766 007252 000240
1767
1768 007254 104100
1769
1770 007256 022626
1771 007260 012777 007306 173540
1772 007266 004737 015056
1773 007272 000140
1774 007274 000240
1775 007276 042777 000100 173500
1776 007304 000402
1777
1778 007306 022626
1779 007310 104101
1780
1781 007312 010377 173510
1782 007316 010477 173506
1783
1784
1785
1786
1787 007322 000004
1788 007324 012737 000036 001104
1789 007332 042777 000100 173444

2#: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
  ERROR 76
;XMIT INTERRUPTS AT PRIORITY=7
5#: BIC #BIT6,BCTCSR ;CLEAR INTERRUPT ENABLE
  MOV #4,BCTVECT ;POINT XMIT VECTOR TO ERROR REPORT
  JSR PC,WRPSW ;SET PSW TO PRIORITY 3
  .WORD 140
BR 5# ;BR TO END OF TEST-NO INTERRUPT

4#: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
  ERROR 77
;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
5#: MOV R3,BCTVECT ;RESTORE XMIT VECTOR

;*****
;*TEST 35 TEST SLUI TRANSMITTER FOR DOUBLE INTERRUPTS
;*****
TST35: SCOPE
  MOV #35,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
  BIC #BIT6,BCTCSR ;CLEAR INTERRUPT ENABLE
  MOV BCTVECT,R3 ;SAVE XMIT VECTOR
  MOV BCTPSW,R4 ;SAVE XMIT PSW VECTOR
  MOV #2,BCTVECT ;SET UP XMIT VECTOR
  MOV #340,BCTPSW ;SET PIO 7 AFTER INTERRUPT
  JSR PC,WRPSW ;SET PSW TO PRIORITY 3
  .WORD 140
1#: TSTB BCTCSR ;WAIT FOR DONE
  BPL 1#
BIS #BIT6,BCTCSR ;ENABLE INTERRUPTS
  NOP
  ERROR 100
;XMIT INTERRUPT DID NOT OCCUR
2#: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
  MOV #4,BCTVECT ;POINT XMIT VECTOR TO ERROR
  JSR PC,WRPSW ;SET PSW TO PRIORITY 3
  .WORD 140
  NOP ;GIVE TIME FOR ANY INTERRUPTS
  BIC #BIT6,BCTCSR ;DISABLE INTERRUPTS
  BR 5# ;BR TO END OF TEST

4#: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
  ERROR 101
;XMIT RE-INTERRUPTED
5#: MOV R3,BCTVECT ;RESTORE XMIT VECTOR
  MOV R4,BCTPSW ;RESTORE XMIT PSW VECTOR

;*****
;*TEST 36 TEST THAT SLUI XMIT INTERRUPT CLEARS WITH LOADING TBUF
;*****
TST36: SCOPE
  MOV #36,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
  BIC #BIT6,BCTCSR ;DISABLE INTERRUPTS

```

```

1790 007340 004737 015056          JSR    PC,WRPSW          ;SET PSW TO PRIORITY 7
1791 007344 000340                   .WORD 340
1792 007346 017703 173454          MOV    @CTVECT,R3       ;SAVE XMIT VECTOR
1793 007352 012777 007424 173446  MOV    #2,@CTVECT      ;POINT XMIT VECTOR TO ERROR
1794 007360 052777 000100 173416  BIS    @BIT6,@CTCSR    ;ENABLE INTERRUPTS
1795 007366 005077 173414          CLR    @CTBUF          ;LOAD TBUF
1796 007372 105777 173406          TSTB  @CTCSR          ;WAIT FOR DONE (INTERRUPT)
1797 007376 100375                   BPL    1#
1798 007400 005077 173402          CLR    @CTBUF          ;FILL SECOND BUFFER TO RESET INT
1799 007404 004737 015056          JSR    PC,WRPSW          ;ALLOW INTERRUPTS
1800 007410 000140                   .WORD 140
1801 007412 000240                   NOP
1802 007414 042777 000100 173362  BIC    @BIT6,@CTCSR    ;GIVE TIME FOR ANY INTERRUPTS
1803 007422 000402                   BR     3#              ;DISABLE INTERRUPTS
1804                                     ;BR TO END OF TEST
1805 007424 022626          2# :  CMP    (SP)+,(SP)+    ;RESTORE SP AFTER INTERRUPT
1806 007426 104102          ERROR 102
1807                                     ;LOADING TBUF DID NOT CLEAR INTERRUPT.
1808 007430 010377 173372          3# :  MOV    R3,@CTVECT    ;RESTORE XMIT VECTOR
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818                                     ;*****
1819                                     ;*TEST 37 TEST THAT SLU1 RCVR INTERRUPTS ONLY WHEN ENABLED
1820 007434 000004          TST37: SCOPE
1821 007436 012737 000037 001104  MOV    #37,@TESTN     ;SET TEST NUMBER IN APT MAIL BOX
1822 007444 000005          RESET                 ;MAKE SURE NO INTERRUPTS ARE PENDING FROM
1823                                     ;ANOTHER TEST
1824 007446 042777 000100 173330  BIC    @BIT6,@CTCSR    ;DISABLE TRANSMIT INTERRUPTS
1825 007454 042777 000100 173316  BIC    @BIT6,@RCRCSR   ;DISABLE RECEIVER INTERRUPTS
1826 007462 052777 000004 173314  BIS    @BIT2,@CTCSR    ;SET MAINTENANCE WRAP
1827 007470 017703 173326          MOV    @RCVVECT,R3     ;SAVE RECEIVE VECTOR
1828 007474 012777 007532 173320  MOV    #2,@RCVVECT     ;POINT RCV VECTOR TO ERROR REPORT
1829 007502 004737 015056          JSR    PC,WRPSW          ;SET PSW TO PRIORITY 3
1830 007506 000140                   .WORD 140
1831 007510 005077 173272          CLR    @CTBUF          ;SEND A CHARACTER
1832 007514 105777 173260          1# :  TSTB  @RCRCSR          ;WAIT FOR RECEIVER DONE
1833 007520 100375                   BPL    1#
1834 007522 042777 000004 173254  BIC    @BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
1835 007530 000405                   BR     3#              ;CONTINUE TEST
1836
1837 007532 042777 000004 173244  2# :  BIC    @BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT
1838 007540 022626          CMP    (SP)+,(SP)+    ;RESTORE SP AFTER INTERRUPT
1839 007542 104103          ERROR 103
1840                                     ;RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR
1841
1842 007544 012777 007572 173250  3# :  MOV    #4,@RCVVECT    ;POINT RCV VECTOR TO END OF TEST
1843 007552 052777 000100 173220  BIS    @BIT6,@RCRCSR   ;ENABLE RCV INTERRUPTS
1844 007560 000240                   NOP
1845 007562 042777 000004 173214  BIC    @BIT2,@CTCSR    ;CLEAR MAINTENANCE BIT

```

```

1846 007570 104104          ERROR 104
1847                                     ;RCVR DID NOT INTERRUPT
1848
1849 007572 042777 000100 173200 4: BIC #BIT6,@CRCSR ;DISABLE INTERRUPTS
1850 007600 042777 000004 173176 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1851 007606 022626 CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1852 007610 010377 173206 MOV R3,@CRVECT ;RESTORE RECEIVE VECTOR
1853
1854
1855 ;*****
1856 ;*TEST 40 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
1857 ;*****
1858 007614 000004 TST40: SCOPE
1859 007616 012737 000040 001104 MOV #40,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1860 007624 000005 RESET ;CLEAR EVERYTHING
1861 007626 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY 7
1862 007632 000340 .WORD 340
1863 007634 017703 173162 MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
1864 007640 012777 007700 173154 MOV #2,@CRVECT ;POINT RCVR VECTOR TO ERROR REPORT
1865 007646 052777 000004 173130 BIS #BIT2,@CTCSR ;SET MAINTENANCE WRAP
1866 007654 005077 173126 CLR @CTBUF ;SEND A CHARACTER
1867 007660 105777 173114 1: TSTB @CRCSR ;WAIT FOR RECEIVER DONE
1868 007664 100375 BPL 1:
1869 007666 052777 000100 173104 BIS #BIT6,@CRCSR ;ENABLE INTERRUPTS
1870 007674 000240 NOP ;GIVE TIME FOR INTERRUPT
1871 007676 000405 BR 3: ;CONTINUE TEST
1872 007700 042777 000004 173076 2: BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1873 007706 022626 CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1874 007710 104105 ERROR 105
1875                                     ;RCVR INTERRUPTS AT PRIORITY 7
1876
1877 007712 042777 000100 173060 3: BIC #BIT6,@CRCSR ;CLEAR INTERRUPT ENABLE
1878 007720 012777 007746 173074 MOV #4,@CRVECT ;POINT RCVR VECTOR TO ERROR REPORT
1879 007726 004737 015056 JSR PC,WRPSW ;SET PSW TO PRIORITY 3
1880 007732 000140 .WORD 140
1881 007734 000240 NOP ;GIVE TIME FOR ANY INTERRUPT
1882 007736 042777 000004 173040 BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1883 007744 000405 BR 5: ;BR TO END OF TEST, IF NO INTERRUPT
1884
1885 007746 042777 000004 173030 4: BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
1886 007754 022626 CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1887 007756 104106 ERROR 106
1888                                     ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
1889 007760 010377 173036 5: MOV R3,@CRVECT ;RESTORE RECEIVE VECTOR
1890
1891
1892 ;*****
1893 ;*TEST 41 TEST SLUI RECEIVER FOR DOUBLE INTERRUPTS
1894 ;*****
1895 007764 000004 TST41: SCOPE
1896 007766 012737 000041 001104 MOV #41,#TESTN ;;SET TEST NUMBER IN APT MAIL BOX
1897 007774 000005 RESET ;CLEAR EVERYTHING
1898 007776 017703 173020 MOV @CRVECT,R3 ;SAVE RECEIVE VECTOR
1899 010002 017704 173016 MOV @CRPSW,R4 ;SAVE RECEIVE PSW VECTOR
1900 010006 012777 010070 173006 MOV #2,@CRVECT ;POINT RCVR VECTOR TO CONTINUE TEST
1901 010014 012777 000340 173002 MOV #340,@CRPSW ;SET PRIORITY TO 7 AFTER INTERRUPT

```

```

1902 010022 004737 015056      JSR      PC,WRPSW      ;SET PSW TO PRIORITY 3
1903 010026 000140                .WORD    140
1904 010030 052777 000004 172746  BIS      @BIT2,@CTCSR  ;SET MAINTENANCE WRAP
1905 010036 005077 172744                CLR      @CTBUF        ;SEND A CHARACTER
1906 010042 105777 172732      1$:     TSTB     @CRCSR    ;WAIT FOR RCVR DONE
1907 010046 100375                BPL      1$
1908 010050 042777 000004 172726  BIC      @BIT2,@CTCSR  ;CLEAR MAINTENANCE BIT
1909 010056 052777 000100 172714  BIS      @BIT6,@CRCSR  ;ENABLE RCV INTERRUPTS
1910 010064 000240                NOP
1911
1912 010066 104107                ERROR    107
1913
1914
1915 010070 022626                2$:     CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1916 010072 012777 010124 172722  MOV      @3,@CRVECT   ;POINT RCV VECTOR TO ERROR REPORT
1917 010100 004737 015056      JSR      PC,WRPSW      ;RESET PSW TO PRIORITY 3
1918 010104 000140                .WORD    140
1919 010106 000240                NOP
1920 010110 042777 000100 172662  BIC      @BIT6,@CRCSR  ;CLEAR INTERRUPT ENABLE
1921 010116 010477 172702      MOV      R4,@CRPSW    ;RESTORE RECEIVE PSW VECTOR
1922 010122 000402                BR       4$
1923
1924 010124 022626                3$:     CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
1925 010126 104110                ERROR    110
1926
1927 010130 010377 172666      4$:     MOV      R3,@CRVECT ;RESTORE RECEIVE VECTOR
1928
1929
1930
1931
1932
1933 010134 000004                ;*****
1934 010136 012737 000042 001104  ;*TEST 42      TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
1935 010144 000005                ;*****
1936 010146 004737 015056      TST42:  SCOPE
1937 010152 000340                MOV      @42,@TESTN   ;;SET TEST NUMBER IN APT MAIL BOX
1938 010154 017703 172642                RESET
1939 010160 012777 010246 172634  JSR
1940 010166 052777 000100 172604  MOV      @2,@CRVECT   ;SAVE RECEIVE VECTOR
1941 010174 052777 000004 172602  BIS      @BIT6,@CRCSR  ;POINT RCV VECTOR TO ERROR REPORT
1942 010202 005077 172600                BIS      @BIT2,@CTCSR  ;SET RCVR INTERRUPT ENABLE
1943 010206 105777 172566                CLR      @CTBUF        ;SET MAINTENANCE WRAP
1944 010212 100375                1$:     TSTB     @CRCSR    ;SEND A CHARACTER
1945 010214 042777 000004 172562  BPL      1$           ;WAIT FOR DONE (INTERRUPT)
1946 010222 005777 172554                BIC      @BIT2,@CTCSR  ;CLEAR MAINTENANCE BIT
1947 010226 004737 015056      TST      @CRBUF        ;READ RBUF TO CLEAR PENDING INTERRUPT
1948 010232 000140                JSR      PC,WRPSW      ;SET PSW TO PRIORITY 3
1949 010234 000240                .WORD    140
1950 010236 042777 000100 172534  NOP
1951 010244 000402                BIC      @BIT6,@CRCSR  ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
1952
1953 010246 022626                2$:     BR       3$           ;NO INTERRUPT CLEAR INT. ENABLE
1954 010250 104111                CMP      (SP)+,(SP)+  ;RESTORE SP AFTER INTERRUPT
1955
1956 010252 010377 172544      3$:     ERROR    111      ;READING RBUF DID NOT CLEAR INTERRUPT
1957

```

85

1958  
 1959  
 1960  
 1961  
 1962  
 1963 010256 000004  
 1964 010260 012737 000043 001104  
 1965 010266 000005  
 1966 010270 004737 015056  
 1967 010274 000340  
 1968 010276 017703 172520  
 1969 010302 012777 010362 172512  
 1970 010310 052777 000100 172462  
 1971 010316 052777 000004 172460  
 1972 010324 012777 000377 172454  
 1973 010332 105777 172442  
 1974 010336 100375  
 1975 010340 000005  
 1976 010342 004737 015056  
 1977 010346 000140  
 1978 010350 000240  
 1979 010352 042777 000100 172420  
 1980 010360 000402  
 1981  
 1982  
 1983 010362 022626  
 1984 010364 104112  
 1985  
 1986 010366 010377 172430  
 1987  
 1988  
 1989  
 1990  
 1991  
 1992 010372 000004  
 1993 010374 012737 000044 001104  
 1994 010402 032777 002000 170430  
 1995 010410 001032  
 1996 010412 000005  
 1997 010414 052777 000004 172362  
 1998 010422 012700 000003  
 1999 010426 005077 172354  
 2000 010432 105777 172346  
 2001 010436 100375  
 2002 010440 005300  
 2003 010442 001371  
 2004 010444 042777 000004 172332  
 2005 010452 032777 040000 172322  
 2006 010460 001001  
 2007 010462 104113  
 2008  
 2009  
 2010 010464 032777 100000 172310  
 2011 010472 001001  
 2012 010474 104114  
 2013

```

;*****
;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, WPSM          ;SET PSM 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
        TSTB   @CRCSR          ;WAIT FOR RCV DONE
        BPL    1#
        RESET   ;CLEAR RCV INTERRUPT & RBUF
        JSR     PC, WPSM          ;SET PSM TO PRIORITY 3
        .WORD   140
        NOP
        BIC    @BIT6, @CRCSR    ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
        BR     3#              ;NO INTERRUPT-CLEAR INT. ENABLE
                                ;CONTINUE TEST

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

        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.
        CLR    @CTBUF          ;LOAD TRANSMIT BUFFER
        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

        BIT     @BIT15, @CRBUF  ;TEST "ERROR" FLAG
        BNE    4#             ;BR, IF SET
        ERROR  114
                                ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
  
```

```

2014 010476
2015
2016
2017
2018
2019
2020 010476 000004
2021 010500 012737 000045 001104
2022 010506 000005
2023 010510 005001
2024 010512 052777 000004 172264
2025 010520 105201
2026 010522 010177 172260
2027 010526 105777 172246
2028 010532 100375
2029 010534 017702 172242
2030 010540 020102
2031 010542 001003
2032 010544 105701
2033 010546 001411
2034 010550 000763
2035 010552 010137 001024
2036 010556 010237 001026
2037 010562 042777 000004 172214
2038 010570 104115
2039
2040 010572 042777 000004 172204
2041
2042
  
```

```

48:
;*****
;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
18: INCB R1 ;INCREMENT THE TEST DATA
MOV R1,@CTBUF ;XMIT A CHARACTER
28: TSTB @CRCSR ;WAIT FOR RECETVER DONE
BPL 28
MOV @CRBUF,R2 ;GET RECEIVED CHARACTER
CMP R1,R2 ;COMPARE DATA
BNE 38 ;BR, IF NON-COMPARE
TSTB R1 ;TEST XMIT DATA FOR ZERO
BEQ 48 ;BR, IF FINISHED
18 ;CONTINUE IF NOT
38: MOV R1,#GDDAT ;STORE THE EXPECTED DATA
MOV R2,#BDDAT ;STORE RECEIVED DATA
BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
ERROR 115 ;DATA COMPARE DATA
48: BIC #BIT2,@CTCSR ;CLEAR MAINTENANCE BIT
  
```

2043 010600  
 2044  
 2045  
 2046  
 2047  
 2048 010600 000004  
 2049 010602 012737 000046 001104  
 2050 010610 032737 000002 001156  
 2051 010616 001002  
 2052 010620 000137 012544  
 2053 010624  
 2054 010624 042777 000100 172142  
 2055 010632 017703 172160  
 2056 010636 012777 010662 172152  
 2057 010644 105777 172124  
 2058 010650 100375  
 2059 010652 004737 015056  
 2060 010656 000140  
 2061 010660 000402  
 2062  
 2063 010662 022626  
 2064 010664 104116  
 2065  
 2066 010666 012777 010706 172122  
 2067 010674 052777 000100 172072  
 2068 010702 000240  
 2069  
 2070 010704 104117  
 2071  
 2072 010706 042777 000100 172060  
 2073 010714 022626  
 2074 010716 010377 172074  
 2075  
 2076  
 2077  
 2078  
 2079  
 2080 010722 000004  
 2081 010724 012737 000047 001104  
 2082 010732 042777 000100 172034  
 2083 010740 004737 015056  
 2084 010744 000340  
 2085 010746 017703 172044  
 2086 010752 012777 011000 172036  
 2087 010760 105777 172010  
 2088 010764 100375  
 2089 010766 052777 000100 172000  
 2090 010774 000240  
 2091 010776 000402  
 2092  
 2093 011000 022626  
 2094 011002 104120  
 2095  
 2096 011004 042777 000100 171762  
 2097 011012 012777 011032 171776  
 2098 011020 004737 015056

SLU2IT:  
 ;\*\*\*\*\*  
 ;TEST 46 TEST THAT SLU2 XMIT INTERRUPTS ONLY WHEN ENABLED  
 ;\*\*\*\*\*  
 TST46: SCOPE  
 MOV #46,#TESTN ;SET TEST NUMBER IN APT MAIL BOX  
 BIT #BIT1,#DEVH ;DO THESE TESTS FOR THIS DEVICE?  
 BNE 991 ;IF YES CONTINUE WITH TESTS  
 JMP LTCIT ;IF NO GO TO START OF NEXT SET OF TESTS.  
 991:  
 BIC #BIT6,#TCSR ;CLEAR TRANSMIT INTERRUPT ENABLE  
 MOV #TVECT,R3 ;SAVE XMIT VECTOR  
 MOV #21,#TVECT ;POINT XMIT VECTOR TO ERROR REPORT  
 11: TSTB #TCSR ;WAIT FOR DONE  
 BPL 11  
 JSR PC,WRPSW ;SET PSW TO PRIORITY 3  
 .WORD 140  
 BR 31  
 21: CMP (SP),.(SP) ;RESTORE SP AFTER INTERRUPT  
 ERROR 116  
 31: MOV #41,#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  
 41: 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 #21,#TVECT ;POINT XMIT VECTOR TO ERROR REPORT  
 11: TSTB #TCSR ;WAIT FOR DONE  
 BPL 11  
 BIS #BIT6,#TCSR ;ENABLE INTERRUPT  
 NOP  
 BR 31 ;CONTINUE TEST  
 21: CMP (SP),.(SP) ;RESTORE SP AFTER INTERRUPT  
 ERROR 120  
 31: BIC #BIT6,#TCSR ;XMIT INTERRUPTS AT PRIORITY 7  
 MOV #41,#TVECT ;CLEAR INTERRUPT ENABLE  
 JSR PC,WRPSW ;POINT XMIT VECTOR TO ERROR REPORT  
 ;SET PSW TO PRIORITY 3

E 5

```
2099 011024 000140 .WORD 140
2100 011026 000240 NOP
2101 011030 000402 BR 58 ;BR TO END OF TEST-NO INTERRUPT
2102
2103 011032 022626 44: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2104 011034 104121 ERROR 121
2105
2106 011036 010377 171754 58: MOV R3,@TVECT ;XMIT INTERRUPT OCCURES WITH BIT6 CLEAR
2107 ;RESTORE XMIT VECTOR
2108
2109
2110
2111
2112
2113 ;*****
;*TEST 50 TEST SLU2 TRANSMITTER FOR DOUBLE INTERRUPTS
;*****
2114 011042 000004 TST50: SCOPE
2115 011044 012737 000050 001104 MOV #50,@TESTN ;SET TEST NUMBER IN APT MAIL BOX
2116 011052 042777 000100 171714 BIC #BIT6,@TCSR ;CLEAR INTERRUPT ENABLE
2117 011060 017703 171732 MOV @TVECT,R3 ;SAVE XMIT VECTOR
2118 011064 017704 171730 MOV @TPSW,R4 ;SAVE XMIT PSW VECTOR
2119 011070 012777 011132 171720 MOV #2,@TVECT ;SET UP XMIT VECTOR
2120 011076 012777 000340 171714 MOV #340,@TPSW ;SET PIO 7 AFTER INTERRUPT
2121 011104 004737 015056 JSR PC,@RPSW ;SET PSW TO PRIORITY 3
2122 011110 000140 .WORD 14C
2123 011112 105777 171656 18: TSTB @TCSR ;WAIT FOR DONE
2124 011116 100375 BPL 18
2125 011120 052777 000100 171646 BIS #BIT6,@TCSR ;ENABLE INTERRUPTS
2126 011126 000240 NOP
2127
2128 011130 104122 ERROR 122
2129 ;XMIT INTERRUPT DID NOT OCCUR
2130 011132 022626 28: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2131 011134 012777 011162 171654 MOV #4,@TVECT ;POINT XMIT VECTOR TO ERROR
2132 011142 004737 015056 JSR PC,@RPSW ;SET PSW TO PRIORITY 3
2133 011146 000140 .WORD 140
2134 011150 000240 NOP
2135 011152 042777 000100 171614 BIC #BIT6,@TCSR ;GIVE TIME FOR ANY INTERRUPTS
2136 011160 000402 BR 58 ;DISABLE INTERRUPTS
2137 ;BR TO END OF TEST
2138 011162 022626 44: CMP (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2139 011164 104123 ERROR 123
2140 ;XMIT RE-INTERRUPTED
2141 011166 010377 171624 58: MOV R3,@TVECT ;RESTORE XMIT VECTOR
2142 011172 010477 171622 MOV R4,@TPSW ;RESTORE XMIT PSW VECTOR
2143
2144 ;*****
;*TEST 51 TEST THAT SLU2 XMIT INTERRUPT CLEARS WITH LOADING TBUF
;*****
2145
2146
2147 011176 000004 TST51: SCOPE
2148 011200 012737 000051 001104 MOV #51,@TESTN ;SET TEST NUMBER IN APT MAIL BOX
2149 011206 032737 000001 001120 BIT #1,@ENV ;ARE WE RUNNING UNDER APT
2150 011214 001403 BEQ 708 ;IF NO THEN DO TEST
2151 011216 005737 001106 TST @PASS ;IS THIS FIRST PASS
2152 011222 001046 BNE TST52 ;IF NO THEN SKIP TO NEXT TEST
2153 011224
2154 011224 042777 000100 171542 708: BIC #BIT6,@TCSR ;DISABLE INTERRUPTS
```



2155	011232	004737	015056		JSR	PC,WRPSW	;SET PSW TO PRIORITY
2156	011236	000340				.WORD 340	
2157	011240	017703	171552		MOV	@TVECT,R3	;SAVE XMIT VECTOR
2158	011244	012777	011316	171544	MOV	@2,@TVECT	;POINT XMIT VECTOR TO ERROR
2159	011252	052777	000100	171514	BIS	@BIT6,@TCSR	;ENABLE INTERRUPTS
2160	011260	005077	171512		CLR	@TBUF	;LOAD TBUF
2161	011264	105777	171504		1@:	TSTB @TCSR	;WAIT FOR DONE (INTERRUPT)
2162	011270	100375				BPL 1@	
2163	011272	005077	171500		CLR	@TBUF	;FILL SECOND BUFFER TO RESET INT.
2164	011276	004737	015056		JSR	PC,WRPSW	;ALLOW INTERRUPTS
2165	011312	000140				.WORD 140	
2166	011304	000240			NOP		;GIVE TIME FOR ANY INTERRUPTS
2167	011306	042777	000100	171460	BIC	@BIT6,@TCSR	;DISABLE INTERRUPTS
2168	011314	000402			BR	3@	;BR TO END OF TEST
2169							
2170	011316	022626			2@:	CMP (SP)+,(SP)+	;RESTORE SP AFTER INTERRUPT
2171	011320	104124				ERROR 124	
2172							;LOADING TBUF DID NOT CLEAR INTERRUPT.
2173	011322	005001			3@:	CLR R1	;INITIALIZE LOOP COUNTER
2174	011324	005201			4@:	INC R1	;INCREMENT LOOP COUNTER
2175	011326	001376				BNE 4@	;UNTIL LOOP COUNTER EQUALS 0
2176	011330	005777	171436			TST @RBUF	;CLEAR RECEIVER BUFFER
2177	011334	010377	171456			MOV R3,@TVECT	;RESTORE XMIT VECTOR
2178							
2179							
2180							
2181							
2182							
2183							
2184							
2185	011340	000004					
2186	011342	012737	000052	001104	TST52:	SCOPE	
2187	011350	042777	000100	171416		MOV #52,@TESTN	;SET TEST NUMBER IN APT MAIL BOX
2188	011356	042777	000100	171404		BIC @BIT6,@TCSR	;DISABLE TRANSMIT INTERRUPTS
2189	011364	017703	171422			BIC @BIT6,@RCRCSR	;DISABLE RECEIVER INTERRUPTS
2190	011370	012777	011420	171414		MOV @RVECT,R3	;SAVE RECEIVE VECTOR
2191	011376	004737	015056			MOV @2,@RVECT	;POINT RCV VECTOR TO ERROR REPORT
2192	011402	000140				JSR PC,WRPSW	;SET PSW TO PRIORITY 3
2193	011404	005077	171366			.WORD 140	
2194	011410	105777	171354			CLR @TBUF	;SEND A CHARACTER
2195	011414	100375			1@:	TSTB @RCRCSR	;WAIT FOR RECEIVER DONE
2196	011416	000402				BPL 1@	
2197						BR 3@	;CONTINUE TEST
2198	011420	022626			2@:	CMP (SP)+,(SP)+	;RESTORE SP AFTER INTERRUPT
2199	011422	104125				ERROR 125	
2200							;RECEIVER INTERRUPTS WITH INT. ENABLE CLEAR
2201							
2202	011424	012777	011444	171360	3@:	MOV @4,@RVECT	;POINT RCV VECTOR TO END OF TEST
2203	011432	052777	000100	171330		BIS @BIT6,@RCRCSR	;ENABLE RCV INTERRUPTS
2204	011440	000240				NOP	;GIVE ANY INTERRUPTS TIME
2205	011442	104126				ERROR 126	
2206							;RCVR DID NOT INTERRUPT
2207							
2208	011444	042777	000100	171316	4@:	BIC @BIT6,@RCRCSR	;DISABLE INTERRUPTS
2209	011452	022626				CMP (SP)+,(SP)+	;RESTORE SP AFTER INTERRUPT
2210	011454	010377	171332			MOV R3,@RVECT	;RESTORE RECEIVE VECTOR

```

2211
2212
2213
2214
2215
2216 011460 000004
2217 011462 012737 000053 001104
2218 011470 004737 015056
2219 011474 000340
2220 011476 017703 171310
2221 011502 012777 011534 171302
2222 011510 005077 171262
2223 011514 105777 171250
2224 011520 100375
2225 011522 052777 000100 171240
2226 011530 000240
2227 011532 000402
2228 011534 022626
2229 011536 104127
2230
2231
2232 011540 042777 000100 171222
2233 011546 012777 011566 171236
2234 011554 004737 015056
2235 011560 000140
2236 011562 000240
2237 011564 000402
2238
2239 011566 022626
2240 011570 104130
2241
2242 011572 010377 171214
2243
2244
2245
2246
2247
2248 011576 000004
2249 011600 012737 000054 001104
2250 011606 017703 171200
2251 011612 017704 171176
2252 011616 012777 011664 171166
2253 011624 012777 000340 171162
2254 011632 004737 015056
2255 011636 000140
2256 011640 005077 171132
2257 011644 105777 171120
2258 011650 100375
2259 011652 052777 000100 171110
2260 011660 000240
2261
2262 011662 104131
2263
2264
2265 011664 022626
2266 011666 012777 011720 171116

```

```

;*****
;TEST 53 TEST THAT RCVR INTERRUPTS DO NOT OCCUR WHEN DISABLED
;*****
TST53: SCOPE
MOV #53, ITESTN ;SET TEST NUMBER IN APT MAIL BOX
JSR PC, WRPSW ;SET PSW TO PRIORITY 7
.WORD 340
MOV BRVECT, R3 ;SAVE RECEIVE VECTOR
MOV #21, BRVECT ;POINT RCVR VECTOR TO ERROR REPORT
CLR BTBUF ;SEND A CHARACTER
TSTB BRCSR ;WAIT FOR RECEIVER DONE
BPL 11
BIS #BIT6, BRCSR ;ENABLE INTERRUPTS
NOP ;GIVE TIME FOR INTERRUPT
BR 31 ;CONTINUE TEST
CMP (SP)+, (SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 127 ;RCVR INTERRUPTS AT PRIORITY 7
31: BIC #BIT6, BRCSR ;CLEAR INTERRUPT ENABLE
MOV #41, BRVECT ;POINT RCVR VECTOR TO ERROR REPORT
JSR PC, WRPSW ;SET PSW TO PRIORITY 3
.WORD 140
NOP ;GIVE TIME FOR ANY INTERRUPT
BR 51 ;BR TO END OF TEST, IF NO INTERRUPT
41: CMP (SP)+, (SP)+ ;RESTORE SP AFTER INTERRUPT
ERROR 130 ;RCVR INTERRUPT REQUEST PASSED WITH BIT6 CLEAR
51: MOV R3, BRVECT ;RESTORE RECEIVE VECTOR
;*****
;TEST 54 TEST SLU2 RECEIVER FOR DOUBLE INTERRUPTS
;*****
TST54: SCOPE
MOV #54, ITESTN ;SET TEST NUMBER IN APT MAIL BOX
MOV BRVECT, R3 ;SAVE RECEIVE VECTOR
MOV BRPSW, R4 ;SAVE RECEIVE PSW VECTOR
MOV #21, BRVECT ;POINT RCVR VECTOR TO CONTINUE TEST
MOV #340, BRPSW ;SET PRIORITY TO 7 AFTER INTERRUPT
JSR PC, WRPSW ;SET PSW TO PRIORITY 3
.WORD 140
CLR BTBUF ;SEND A CHARACTER
TSTB BRCSR ;WAIT FOR RCVR DONE
BPL 11
BIS #BIT6, BRCSR ;ENABLE RCVR INTERRUPTS
NOP ;GIVE SOME TIME
ERROR 131 ;RCVR INTERRUPT DID NOT OCCUR
21: CMP (SP)+, (SP)+ ;RESTORE SP AFTER INTERRUPT
MOV #31, BRVECT ;POINT RCVR VECTOR TO ERROR REPORT

```

```

2267 011674 004737 015056      JSR    PC,WRPSW      ;RESET PSW TO PRIORITY 3
2268 011700 000140              .WORD    140
2269 011702 000240              NOP
2270 011704 042777 000100 171056  BIC    #BIT6,BRCSR   ;GIVE SOME TIME
2271 011712 010477 171076      MOV    R4,BRPSW     ;CLEAR INTERRUPT ENABLE
2272 011716 000402              BR      41          ;RESTORE RECEIVE PSW VECTOR
2273                                ;BR TO END OF TEST
2274 011720 022626      31:    CMP    (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2275 011722 104132      ERROR  132
2276                                ;RECEIVER RE INTERRUPTED
2277 011724 010377 171062      41:    MOV    R3,BRVECT ;RESTORE RECEIVE VECTOR
2278
2279
2280                                ;*****
2281                                ;*TEST 55      TEST THAT RCVR INTERRUPT CLEARS BY READING RBUF
2282                                ;*****
2283                                TST55:  SCOPE
2284 011730 000004              MOV    #55,#TESTN    ;SET TEST NUMBER IN APT MAIL BOX
2285 011732 012737 000055 001104  JSR    PC,WRPSW     ;SET PSW PRIORITY TO 7
2286 011740 004737 015056              .WORD    340
2287 011744 000340              MOV    BRVECT,R3    ;SAVE RECEIVE VECTOR
2288 011746 017703 171040      MOV    #21,BRVECT   ;POINT RCV VECTOR TO ERROR REPORT
2289 011752 012777 012024 171032  BIS    #BIT6,BRCSR   ;SET RCVR INTERRUPT ENABLE
2290 011760 052777 000100 171002  CLR    @TBUF        ;SEND A CHARACTER
2291 011766 005077 171004      11:    TSTB  BRCSR        ;WAIT FOR DONE (INTERRUPT)
2292 011772 105777 170772      BPL    11
2293 011776 100375              TST    BRBUF        ;READ RBUF TO CLEAR PENDING INTERRUPT
2294 012000 005777 170766      JSR    PC,WRPSW     ;SET PSW TO PRIORITY 3
2295 012004 004737 015056              .WORD    140
2296 012010 000140              NOP
2297 012012 000240              ;ALLOW TIME FOR ANY ERRONEOUS INTERRUPT
2298 012014 042777 000100 170746  BIC    #BIT6,BRCSR   ;NO INTERRUPT-CLEAR INT. ENABLE
2299 012022 000402              BR      31
2300                                21:    CMP    (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2301                                ERROR  133
2302                                ;READING RBUF DID NOT CLEAR INTERRUPT
2303 012024 022626      31:    MOV    R3,BRVECT   ;RESTORE RECEIVE VECTOR
2304
2305
2306
2307                                ;*****
2308                                ;*TEST 56      TEST SLU2 THAT RESET CLEARS RECEIVE INTERRUPT
2309                                ;*****
2310                                TST56:  SCOPE
2311 012034 000004              MOV    #56,#TESTN    ;SET TEST NUMBER IN APT MAIL BOX
2312 012036 012737 000056 001104  BIT    #1,@#ENV     ;ARE WE RUNNING UNDER APT
2313 012044 032737 000001 001120  BEQ    701          ;IF NO THEN DO TEST
2314 012052 001403              BEQ    701
2315 012054 005737 001106      TST    @#PASS       ;IS THIS FIRST PASS
2316 012060 001037              BNE    TST57        ;IF NO THEN SKIP TO NEXT TEST
2317 012062 000005      701:    RESET
2318 012064 004737 015056      JSR    PC,WRPSW     ;CLEAR EVERYTHING
2319 012070 000340              .WORD    340       ;SET PSW TO PRIORITY 7
2320 012072 017703 170714      MOV    BRVECT,R3    ;SAVE RECEIVE VECTOR
2321 012076 012777 012150 170706  MOV    #21,BRVECT   ;POINT RCV VECTOR TO ERROR REPORT
2322 012104 052777 000100 170656  BIS    #BIT6,BRCSR   ;SET RCV INTERRUPT ENABLE

```

```

2323 012112 012777 000377 170656      MOV    #377,@TBUF    ;SEND AN ALL 1'S CHARACTER
2324 012120 105777 170644      1#:   TSTB   @RCSR    ;WAIT FOR RCV DONE
2325 012124 100375                    BPL    1#
2326 012126 000005                    RESET  ;CLEAR RCV INTERRUPT & RBUF
2327 012130 004737 015056      JSR    PC,WRPSW     ;SET PSW TO PRIORITY 3
2328 012134 000140                    .WORD 140
2329 012136 000240                    NOP
2330 012140 042777 000100 170622      BIC    @BIT6,@RCSR  ;ALLOW TIME FOR AN ERRONEOUS INTERRUPT
2331 012146 000402                    BR     3#           ;NO INTERRUPT-CLEAR INT. ENABLE
2332
2333
2334 012150 022626      2#:   CMP    (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2335 012152 104134      ERROR 134          ;RESET DID NOT CLEAR RCVR INTERRUPT
2336
2337 012154 010377 170632      3#:   MOV    R3,@RVECT ;RESTORE RECEIVE VECTOR
2338
2339
2340
2341
2342
2343 012160 000004      ;*****
2344 012162 012737 000057 001104      ;*TEST 57 TEST SLU2 THAT "OVERRUN & ERROR" BITS CAN BE SET
2345 012170 032777 002000 166642      ;*****
2346 012176 001023      TST57: SCOPE
2347 012200 012700 000003      MOV    #57,@TESTN   ;SET TEST NUMBER IN APT MAIL BOX
2348 012204 005077 170566      1#:   BIT    @BIT10,@SWR ;IS THIS TEST DISABLED
2349 012210 105777 170560      2#:   BNE   T560      ;IF NOT ENABLED, BR TO NEXT TEST
2350 012214 100375      MOV    #3,R0        ;SET CHARACTER COUNT TO SEND 3 CHAR.
2351 012216 005300      CLR    @TBUF        ;LOAD TRANSMIT BUFFER
2352 012220 001371      2#:   TSTB  @TCSR    ;WAIT FOR TRANSMIT DONE
2353 012222 032777 040000 170542      BPL    2#
2354 012230 001001      DEC    R0           ;DECREMENT CHARACTER COUNT
2355 012232 104135      BNE   1#           ;BR IF ALL CHARACTERS NOT TRANSMITTED
2356
2357
2358 012234 032777 100000 170530      BIT    @BIT14,@RBUF ;TEST FOR "OR" ERROR FLAG
2359 012242 001001      BNE   3#           ;BR, IF SET
2360 012244 104136      ERROR 135
2361
2362 012246      ;"OR" ERROR FLAG DID NOT SET
2363
2364
2365
2366
2367
2368 012246 000004      3#:   BIT    @BIT15,@RBUF ;TEST "ERROR" FLAG
2369 012250 012737 000060 001104      BNE   4#           ;BR, IF SET
2370 012256 012777 177777 170512      4#:   ERROR 136
2371 012264 105777 170500
2372 012270 100375
2373 012272 005777 170474
2374 012276 052777 000001 170470
2375 012304 005000
2376 012306 105777 170456
2377 012312 100406
2378 012314 005200
;*****
;*TEST 60 TEST THAT BREAK TRANSMITS ALL ZEROES
;*****
TST60: SCOPE
MOV    #60,@TESTN   ;SET TEST NUMBER IN APT MAIL BOX
MOV    #-1,@TBUF    ;TRANSMIT ALL ONES TO RCVR
1#:   TSTB  @RCSR    ;WAIT FOR RCVR DONE
BPL    1#
TST   @RBUF        ;CLEAR DONE (LEAVING ALL ONES IN RBUF)
BIS   @BIT0,@TCSR  ;TRANSMIT BREAK
CLR   R0           ;CLEAR A TIMER
2#:   TSTB  @RCSR    ;WAIT FOR RCVR DONE
BMI   CONT41      ;BR IF DONE
INC   R0           ;IF NOT, INCREMENT TIMER

```

```
2379 012316 001373 BNE 2# ;BR IF TIME REMAINS
2380
2381 012320 042777 000001 170446 BIC #BIT0,@TCSR ;CLEAR BREAK BITS
2382 012326 104137 ERROR 137 ;BREAK DID NOT TRANSMIT ANYTHING
2383
2384 012330 105777 170436 CONT41: TSTB #RBUF ;CHECK RECEIVE BUFFER FOR ZERO
2385 012334 001404 BEQ 3# ;BR, IF ZERO
2386 012336 042777 000001 170430 BIC #BIT0,@TCSR ;CLEAR BREAK BITS
2387
2388 012344 104137 ERROR 137 ;BREAK DID NOT TRANSMIT ALL ZEROES
2389
2390 012346 042777 000001 170420 3#: BIC #BIT0,@TCSR ;CLEAR BREAK BITS
2391
2392
2393 ;*****
2394 ;*TEST 61 TEST THAT "FR" ERROR CAN BE SET DURING BREAK
2395 ;*****
2396 012354 000004 TST61: SCOPE
2397 012356 012737 000061 001104 MOV #61,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
2398 012364 032777 002000 166446 BIT #BIT10,@SWR ;IS THIS TEST DISABLED
2399 012372 001025 BNE TST62 ;BR TO NEXT TEST, IF DISABLED
2400 012374 052777 000001 170372 BIS #BIT0,@TCSR ;SEND BREAK
2401 012402 005077 170370 CLR @RBUF ;TRANSMIT A CHARACTER TO TIME BREAK
2402 012406 105777 170356 1#: TSTB #RCSR ;WAIT FOR RCVR DONE
2403 012412 100375 RPL 1#
2404 012414 042777 000001 170352 BIC #BIT0,@TCSR ;CLEAR BREAK BITS
2405 012422 032777 020000 170342 BIT #BIT13,@RBUF ;CHECK FOR FRAMING ERROR FLAG
2406 012430 001001 BNE 2# ;BR, IF SET
2407
2408 012432 104140 ERROR 140
2409
2410 012434 032777 100000 170330 2#: BIT #BIT15,@RBUF ;BREAK DID NOT SET FRAMING ERROR
2411 012442 001001 BNE 3# ;TEST "ERROR" FLAG
2412 ;BR, IF SET
2413 012444 104141 ERROR 141
2414 ;"ERROR" FLAG DID NOT SET WITH "OR" FLAG
2415 012446 3#:
2416 ;*****
2417 ;*TEST 62 TEST DATA PATHS USING WRAP CABLE
2418 ;*****
2419 012446 000004 TST62: SCOPE
2420 012450 012737 000062 001104 MOV #62,#TESTN ;SET TEST NUMBER IN APT MAIL BOX
2421 012456 032777 000200 166354 BIT #BIT7,@SWR ;IS THIS TEST ENABLED
2422 012464 001027 BNE TST63 ;BR, IF NOT
2423 012466 005001 CLR R1 ;CLEAR REGISTER FOR TEST DATA
2424 ;TRANSMIT A BINARY COUNT PATTERN UP
2425 ;TO THE BIT POSITION INDICATED BY THE
2426 ;CONTENTS OF LOCATION "#USWR"
2427 012470 105201 1#: INCB R1 ;INCREMENT THE TEST DATA
2428 012472 010177 170300 MOV R1,@RBUF ;XMIT A CHARACTER
2429 012476 005000 CLR R0 ;CLEAR A TIMER
2430 012500 105777 170264 2#: TSTB #RCSR ;WAIT FOR RECEIVER DONE
2431 012504 100403 BMI 3# ;BR IF DONE
2432 012506 005200 INC R0 ;INCREMENT TIMER IF NOT
2433 012510 001373 BNE 2# ;BR IF TIME REMAINS
2434
```

2435	012512	104056		ERROR	56	;RECEIVER DONE NOT SET	
2436							
2437	012514	017702	170252	3#:	MOV	BRBUF,R2	;GET RECEIVED CHARACTER
2438	012520	020102			CMF	R1,R2	;COMPARE DATA
2439	012522	001003			BNE	4#	;BR, IF NON-COMPARE
2440	012524	105701			TSTB	R1	;TEST XMIT DATA FOR ZERO
2441	012526	001406			BEQ	TST63	;BR, IF FINISHED
2442	012530	000757			BR	1#	;CONTINUE IF NOT
2443	012532	010137	001024	4#:	MOV	R1,#GDDAT	;STORE EXPECTED DATA
2444	012536	010237	001026		MOV	R2,#BDDAT	;STORE RECEIVED DATA
2445							
2446	012542	104142		ERROR	142	;DATA COMPARE ERROR WITH WRAP CABLE	
2447							
2448							

2449 012544  
 2450  
 2451  
 2452  
 2453  
 2454 012544 000004  
 2455 012546 012737 000063 001104  
 2456 012554 032737 000004 001156  
 2457 012562 001002  
 2458 012564 000137 013600  
 2459 012570  
 2460 012570 004737 015056  
 2461 012574 000340  
 2462 012576 017703 170230  
 2463 012602 017704 170226  
 2464 012606 012777 012650 170216  
 2465 012614 012777 000340 170212  
 2466 012622 042777 000200 170160  
 2467 012630 052777 000100 170152  
 2468 012636 105777 170146  
 2469 012642 100375  
 2470 012644 000240  
 2471 012646 000402  
 2472  
 2473 012650 022626  
 2474 012652 104143  
 2475  
 2476 012654 005077 170130  
 2477 012660 012777 012706 170144  
 2478 012666 004737 015056  
 2479 012672 000240  
 2480 012674 105777 170110  
 2481 012700 100375  
 2482 012702 000240  
 2483 012704 000402  
 2484  
 2485 012706 022626  
 2486 012710 104144  
 2487  
 2488 012712 012777 012746 170112  
 2489 012720 042777 000200 170062  
 2490 012726 052777 000100 170054  
 2491 012734 105777 170050  
 2492 012740 100375  
 2493 012742 000240  
 2494  
 2495 012744 104145  
 2496  
 2497 012746 022626  
 2498 012750 042777 000100 170032  
 2499 012756 010377 170050  
 2500 012762 010477 170046  
 2501  
 2502  
 2503  
 2504

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,#DEVN ;DO THESE TESTS FOR THIS DEVICE?
BNE 99# ;IF YES CONTINUE WITH TESTS
JMP BLAST ;IF NO GO TO START OF NEXT SET OF TESTS.

99#: 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,@BLKS ;CLEAR CLOCK DONE FLAG
BIS #BIT6,@BLKS ;SET INTERRUPT ENABLE
1#: TSTB @BLKS ;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 @BLKS ;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 @BLKS ;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,@BLKS ;CLEAR CLOCK DONE FLAG
BIS #BIT6,@BLKS ;ALLOW INTERRUPTS
6#: TSTB @BLKS ;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,@BLKS ;DISABLE INTERRUPTS
MOV R3,@RTCVT ;RESTORE LINE CLOCK VECTOR
MOV R4,@RTCP SW ;RESTORE LINE CLOCK PSW VECTOR

```

;:\*\*\*\*\*

```
2505 ;*TEST 64 TEST RTC FOR DOUBLE INTERRUPTS
2506 ;*****
2507 012766 000004 TST64: SCOPE
2508 012770 012737 000064 001104 MOV #64, #TESTN ;SET TEST NUMBER IN APT MAIL BOX
2509 012776 032737 000001 001120 BIT #1, #ENV ;ARE WE RUNNING UNDER APT
2510 013004 001403 BEQ 70$ ;IF NO THEN DO TEST
2511 013006 005737 001106 TST #PASS ;IS THIS FIRST PASS
2512 013012 001052 BNE TST65 ;IF NO THEN SKIP TO NEXT TEST
2513 013014 70$:
2514 013014 017703 170012 MOV #RTCVT, R3 ;SAVE LINE CLOCK VECTOR
2515 013020 017704 170010 MOV #RTCPSW, R4 ;SAVE LINE CLOCK PSW VECTOR
2516 013024 012777 013774 170000 MOV #2$, #RTCVT ;SET UP RTC INTERRUPT VECTOR
2517 013032 012777 000340 167774 MOV #340, #RTCPSW ;DISALLOW INTERRUPTS AFTER THE INTERRUPT
2518 013040 004737 015056 JSR PC, WRPSW ;SET PRIORITY TO 5
2519 013044 000240 .WORD 240
2520 013046 042777 000200 167734 BIC #BIT7, #BLKS ;CLEAR CLOCK DONE FLAG
2521 013054 052777 000100 167726 BIS #BIT6, #BLKS ;ENABLE CLOCK INTERRUPTS
2522 013062 105777 167722 1$: TSTB #BLKS ;WAIT FOR DONE
2523 013066 100375 BPL 1$
2524 013070 000240 NOP ;GIVE TIME FOR ANY INTERRUPT
2525
2526 013072 104146 ERROR 146 ;RTC INTERRUPT DID NOT OCCUR
2527
2528 013074 022626 2$: CMP (SP)+, (SP)+ ;RESTORE SP AFTER INTERRUPT
2529 013076 012777 013116 167726 MOV #3$, #RTCVT ;POINT RTC VECTOR TO ERROR REPORT
2530 013104 004737 015056 JSR PC, WRPSW ;SET PSW TO PRIORITY 5
2531 013110 000240 .WORD 240
2532 013112 000240 NOP ;GIVE SOME TIME FOR AN INTERRUPT
2533 013114 000402 BR 4$ ;NO INTERRUPT - BR TO END OF TEST
2534
2535 013116 022626 3$: CMP (SP)+, (SP)+ ;RESTORE SP AFTER INTERRUPT
2536 013120 104147 ERROR 147 ;INTERRUPT SEQUENCE DID NOT CLEAR
2537 ;INTERRUPT REQUEST
2538
2539 013122 042777 000100 167660 4$: BIC #BIT6, #BLKS ;DISABLE CLOCK INTERRUPTS
2540 013130 010377 167676 MOV R3, #RTCVT ;RESTORE LINE CLOCK VECTOR
2541 013134 010477 167674 MOV R4, #RTCPSW ;RESTORE LINE CLOCK PSW VECTOR
2542
2543
2544 ;*****
2545 ;*TEST 65 TEST THAT RTC INTERRUPT CLEARS WITH RESET
2546 ;*****
2547 013140 000004 TST65: SCOPE
2548 013142 012737 000065 001104 MOV #65, #TESTN ;SET TEST NUMBER IN APT MAIL BOX
2549 013150 032737 000001 001120 BIT #1, #ENV ;ARE WE RUNNING UNDER APT
2550 013156 001403 BEQ 70$ ;IF NO THEN DO TEST
2551 013160 005737 001106 TST #PASS ;IS THIS FIRST PASS
2552 013164 001036 BNE TST66 ;IF NO THEN SKIP TO NEXT TEST
2553 013166 70$:
2554 013166 004737 015056 JSR PC, WRPSW ;SET PRIORITY TO 7
2555 013172 000340 .WORD 340
2556 013174 017703 167632 MOV #RTCVT, R3 ;SAVE LINE CLOCK VECTOR
2557 013200 012777 013252 167624 MOV #2$, #RTCVT ;POINT RTC VECTOR TO ERROR REPORT
2558 013206 042777 000200 167574 BIC #BIT7, #BLKS ;CLEAR CLOCK DONE FLAG
2559 013214 052777 000100 167566 BIS #BIT6, #BLKS ;ENABLE CLOCK INTERRUPTS
2560 013222 105777 167562 1$: TSTB #BLKS ;WAIT FOR DONE (INTERRUPT REQUEST)
```



```

2561 013226 100375          BPL      1#
2562 013230 000005          RESET          ;CLEAR PENDING INTERRUPT WITH RESET
2563 013232 004737 015056  JSR      PC,WRPSW      ;SET PRIORITY TO 5
2564 013236 000240          .WORD    240
2565 013240 000240          NOP
2566 013242 042777 000100 167540 BIC      #BIT6,BLKS      ;GIVE TIME FOR ANY INTERRUPT
2567 013250 000402          BR       3#           ;DISALLOW INTERRUPTS
2568
2569 013252 022626          2# :    CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2570 013254 104150          ERROR    150         ;RESET DID NOT CLEAR INTERRUPT
2571
2572 013256 010377 167550          3# :    MOV      R3,BRTCVT   ;RESTORE LINE CLOCK VECTOR
2573
2574
2575
2576
2577
2578 013262 000004          ;:*****
2579 013264 012737 000066 001104 ;*TEST 66      TEST THAT RTC INTERRUPT CLEARS BY CLEARING BIT7 OF LKS
2580 013272 032737 000001 001120 ;:*****
2581 013300 001403          TST66:  SCOPE
2582 013302 005737 001106          MOV      #66,#TESTN    ;:SET TEST NUMBER IN APT MAIL BOX
2583 013306 001043          BIT      #1,#ENV       ;:ARE WE RUNNING UNDER APT
2584 013310 013310          BEQ      70#          ;:IF NO THEN DO TEST
2585 013310 004737 015056          TST      #0#PASS      ;:IS THIS FIRST PASS
2586 013314 000340          BNE      TST67        ;:IF NO THEN SKIP TO NEXT TEST
2587 013316 017703 167510          70# :    JSR      PC,WRPSW      ;SET PRIORITY TO 7
2588 013322 012777 013400 167502  .WORD    340
2589 013330 042777 000200 167452  MOV      BRTCVT,R3    ;SAVE LINE CLOCK VECTOR
2590 013336 052777 000100 167444  MOV      #2#,BRTCVT   ;POINT RTC VECTOR TO ERROR REPORT
2591 013344 105777 167440          BIC      #BIT7,BLKS    ;CLEAR CLOCK DONE FLAG
2592 013350 100375          BIS      #BIT6,BLKS    ;ENABLE CLOCK INTERRUPTS
2593 013352 042777 000200 167430 1# :    TSTB     BLKS       ;WAIT FOR DONE (INTERRUPT REQUEST)
2594 013360 004737 015056          BPL      1#
2595 013364 000240          BIC      #BIT7,BLKS    ;CLEAR DONE & INTERRUPT
2596 013366 000240          JSR      PC,WRPSW      ;ALLOW INTERRUPTS
2597 013370 042777 000100 167412  .WORD    240
2598 013376 000402          NOP
2599
2600
2601 013400 022626          2# :    CMP      (SP)+,(SP)+ ;RESTORE SP AFTER INTERRUPT
2602 013402 104151          ERROR    151         ;CLEARING BIT7 OF LKS DID NOT CLEAR INTERRUPT
2603
2604 013404 010377 167422          3# :    MOV      R3,BRTCVT   ;RESTORE LINE CLOCK VECTOR
2605 013410 004737 015056          JSR      PC,WRPSW      ;RESTORE PRIORITY TO 7
2606 013414 000340          .WORD    340
2607
2608
2609
2610
2611
2612 013416 000004          ;:*****
2613 013420 012737 000067 001104 ;*TEST 67      TEST CLOCK REPEATABILITY
2614 013426 032737 000001 001120 ;:*****
2615 013434 001403          TST67:  SCOPE
2616 013436 005737 001106          MOV      #67,#TESTN    ;:SET TEST NUMBER IN APT MAIL BOX
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700

```

2617	013442	001056				BNE	TST70		; IF NO THEN SKIP TO NEXT TEST
2618	013444				70:				
2619	013444	042777	000100	167336		BIC	#BIT6,BLKS		; DISALLOW INTERRUPTS
2620	013452	005000				CLR	R0		; CLEAR A TIMER
2621	013454	012701	177777			MOV	#-1,R1		; SET A FLAG INDICATING FIRST PASS THRU THIS LOOP
2622	013460	005002			18:	CLR	R2		; CLEAR CLOCK COUNTER
2623	013462	005077	167322			CLR	BLKS		; CLEAR DONE
2624	013466	105777	167316		28:	TSTB	BLKS		; SYNC ON DONE
2625	013472	100375				BPL	28		
2626	013474	005077	167310			CLR	BLKS		; CLEAR DONE
2627	013500	105777	167304		38:	TSTB	BLKS		; IS CLOCK DONE?
2628	013504	100003				BPL	48		; BR IF NOT, TO INCREMENT TIMER
2629	013506	005202				INC	R2		; IF DONE, INCREMENT CLOCK COUNT
2630	013510	005077	167274			CLR	BLKS		; CLEAR DONE
2631	013514	005200			48:	INC	R0		; INCREMENT TIMER
2632	013516	001370				BNE	38		; BR IF TIME REMAINS
2633	013520	005201				INC	R1		; INCREMENT LOOP PASS FLAG
2634	013522	001003				BNE	CMPCARE		; BR IF TWO PASSES HAVE BEEN MADE
2635	013524	010237	013574			MOV	R2,FIRST		; IF NOT, STORE FIRST CLOCK COUNT
2636	013530	000753				BR	18		; DO LOOP AGAIN
2637	013532	013701	013574		CMPCARE	MOV	FIRST,R1		; RECALL FIRST CLOCK COUNT
2638	013536	160201				SUB	R2,R1		; CALCULATE DIFFERENCE OF TWO COUNTS
2639	013540	100001				BPL	TOLER		; IF POSITIVE, SKIP NEGATION OF DIFFERENCE
2640	013542	005401				NEG	R1		; MAKE DIFFERENCE A POSITIVE NUMBER
2641	013544	020127	000002		TOLER:	CMPCARE	R1,#2		; COMPARE DIFFERENCE WITH DESIRED TOLERANCE
2642	013550	003403				BLE	58		; BR, IF LOWER/EQUAL TO TOLERANCE
2643									
2644	013552	010237	013576			MOV	R2,SECND		; STORE SECOND COUNT
2645	013556	104152				ERROR	152		; CLOCK REPEATABILITY ERROR
2646									
2647	013560	032777	000040	165252	58:	BIT	#BIT5,BSWR		; CLOCK TESTS ONLY?
2648	013566	001404				BEQ	TST70		; BR IF NOT
2649	013570	000137	014740			JMP	#EOP		; ELSE, JUMP TO END OF PASS ROUTINE
2650									
2651	013574	000000			FIRST:	0			
2652	013576	000000			SECND:	0			

```

2653 013600
2654
2655
2656
2657
2658 013600 000004
2659 013602 012737 000070 001104
2660 013610 032737 000001 001120
2661 013616 001405
2662 013620 005737 001106
2663 013624 001402
2664 013626 000137 014740
2665 013632 000005
2666 013634 012737 060340 177776
2667 013642 017737 167150 001060
2668 013650 017737 167136 001062
2669 013656 017737 167144 001064
2670 013664 017737 167132 001066
2671 013672 017737 167134 001070
2672 013700 005037 014726
2673 013704 005037 014730
2674 013710 005037 014732
2675 013714 005037 014734
2676 013720 005037 014736
2677 013724 012777 014154 167074
2678 013732 012777 000340 167070
2679 013740 012777 014210 167054
2680 013746 012777 000340 167050
2681 013754 012777 014270 167034
2682 013762 012777 000340 167030
2683 013770 012777 014324 167014
2684 013776 012777 000340 167010
2685 014004 012777 014144 167020
2686 014012 012777 000340 167014
2687 014020 032737 000001 001156
2688 014026 001413
2689 014030 052777 000004 166746
2690 014036 052777 000100 166740
2691 014044 052777 000100 166726
2692 014052 012702 030450
2693 014056 032737 000002 001156
2694 014064 001410
2695 014066 052777 000100 166700
2696 014074 052777 000100 166666
2697 014102 012703 031050
2698 014106 032737 000004 001156
2699 014114 001403
2700 014116 052777 000100 166664
2701 014124 012700 177777
2702 014130 012701 177777
2703 014134 005037 177776
2704 014140 000001
2705 014142 000776
2706
2707 014144 005237 014736
2708 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 @BC*VECT, #TMP2
MOV @BCRVECT, #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, #DEVH ;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, #DEVH ;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, #DEVH ;IS LTC UNDER TEST
BEQ 3# ;IF NO DON'T SET IT UP
BIS #BIT6, @LCS ;ENABLE RTC INTERRUPTS
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

```

2709	014154	005237	014726			XMIT1:	INC	XMTCT1	;	UPDATE XMIT INTERRUPT COUNT
2710	014160	005200					INC	RO	;	UPDATE XMIT DATA
2711	014162	010077	166620				MOV	RO, @CTBUF	;	SEND NEXT CHARACTER
2712	014166	023727	014726	00040C			CMP	XMTCT1, #400	;	IF 256 CHARACTERS HAVE NOT BEEN
2713	014174	002403					BLT	#	;	TRANSFERRED CONTINUE
2714	014176	042777	000100	166600			BIC	@BIT6, @CTCSR	;	IF YES TURN OFF TRANSMITTER INTERRUPTS
2715	014204	000137	014400			1:	JMP	IOHAND	;	GO TO INTERRUPT HANDLER
2716										
2717	014210	005237	014732			REC1:	INC	RECCT1	;	UPDATE RECEIVER INTERRUPT COUNT
2718	014214	005777	166562				TST	BCRBUF	;	BIT 15 SET IF ANY ERRORS OCCURRED
2719	014220	100017					BPL	#	;	IF BIT IS CLEAR NO ERROR
2720	014222	017737	166554	001026			MOV	BCRBUF, @BDDAT	;	GET ERROR INFORMATION
2721	014230	000005					RESET		;	CLEAR THE WORLD STOP ALL
2722									;	INTERRUPTS
2723	014232	020227	030450				CMP	R2, @BUF1	;	WAS MORE THAN 1 WORD TRANSFERRED
2724	014236	003004					BGT	#	;	IF YES GET LAST GOOD DATA
2725	014240	012737	177777	001024			MOV	#-1, @GDDAT	;	MAKE GOOD DATA = -1
2726	014246	000403					BR	#	;	GO TO ERROR REPORT
2727	014250	116237	177777	001024		1:	MOVB	-1(R2), @GDDAT	;	GET LAST GOOD DATA
2728	014256	104153				2:	ERROR	153	;	RECEIVER STATUS ERROR
2729	014260	117722	166516			3:	MOVB	BCRBUF, (R2)	;	GET DATA AND STORE IT
2730	014264	000137	014400				JMP	IOHAND	;	GO TO INTERRUPT HANDLER
2731										
2732	014270	005237	014730			XMIT2:	INC	XMTCT2	;	UPDATE XMIT INTERRUPT COUNT
2733	014274	005201					INC	R1	;	UPDATE XMIT DATA
2734	014276	010177	166474				MOV	R1, @TBUF	;	SEND NEXT CHARACTER
2735	014302	023727	014730	000400			CMP	XMTCT2, #400	;	IF 256 CHARACTERS HAVE NOT
2736	014310	002403					BLT	#	;	BEEN TRANSFERRED CONTINUE
2737	014312	042777	000100	166454			BIC	@BIT6, @TCSR	;	ELSE NO MORE XMIT INTERRUPTS
2738	014320	000137	014400			1:	JMP	IOHAND	;	GO TO INTERRUPT HANDLER
2739										
2740	014324	005237	014734			REC2:	INC	RECCT2	;	UPDATE RECEIVER INTERRUPT COUNT
2741	014330	005777	166436				TST	BRBUF	;	BIT 15 SETS IF ANY ERRORS OCCURRED
2742	014334	100017					BPL	#	;	IF BIT IS CLEAR NO ERRORS
2743	014336	017737	166430	001026			MOV	BRBUF, @BDDAT	;	GET ERROR INFORMATION
2744	014344	000005					RESET		;	CLEAR THE WORLD - STOP ALL
2745									;	INTERRUPTS
2746	014346	020327	031050				CMP	R3, @BUF2	;	WAS MORE THAN 1 WORD TRANSFERRED
2747	014352	003004					BGT	#	;	IF YES GET LAST GOOD DATA
2748	014354	012737	177777	001024			MOV	#-1, @GDDAT	;	IF NO MAKE GOOD DATA -1
2749	014362	000403					BR	#	;	AND GET TO ERROR REPORT
2750	014364	116337	177777	001024		1:	MOVB	-1(R3), @GDDAT	;	GET LAST GOOD DATA RECEIVED
2751	014372	104154				2:	ERROR	154	;	RECEIVER STATUS ERROR
2752	014374	117723	166372			3:	MOVB	BRBUF, (R3)	;	GET DATA AND STORE IT
2753										
2754	014400	032737	000004	001156		IOHAND:	BIT	@BIT2, @DEVH	;	IS RTC UNDER TEST
2755	014406	001416					BEQ	#	;	IF NO CHECK OTHER DEVICES
2756	014410	023727	014736	000074			CMP	COUNT, #74	;	HAS 1 SEC ELAPSED
2757	014416	002427					BLT	#	;	IF NO CONTINUE TEST
2758	014420	042777	000100	166356			BIC	@BIT6, @CTCSR	;	IF YES STOP TRANSMISSIONS
2759	014426	042777	000100	166340			BIC	@BIT6, @TCSR	;	
2760	014434	042777	000100	166346			BIC	@BIT6, @LKS	;	TURN OFF LINE CLOCK
2761	014442	000416					BR	WAITER	;	
2762	014444	032737	000001	001156		1:	BIT	@BIT0, @DEVH	;	IS SLU1 UNDER TEST
2763	014452	001405					BEQ	#	;	IF NO CHECK FOR SLU2
2764	014454	032777	000100	166322			BIT	@BIT6, @CTCSR	;	IS TRANSMITTER SHUTDOWN

```
2765 014462 001405      BEQ      3#      ;IF NO CONTINUE TEST
2766 014464 000405      BR       WAITER ;IF YES GO WAIT FOR POSSIBLE
2767
2768 014466 032777 000100 166300 2# : BIT      #BIT6, #TCSR ;IS TRANSMITTER SHUT DOWN
2769 014474 001001      BNE      WAITER ;IF YES GO WAIT FOR POSSIBLE
2770
2771 014476 000002      3# : RTI      ;RETURN FROM INTERRUPT TO AWAIT NEXT
2772
2773 014500 005037 177776      WAITER: CLR     PS      ;MAKE PROCESSOR PRIORITY 0
2774 014504 012705 140000      MOV     #-40000, R5 ;SET UP LOOP COUNTER
2775 014510 062705 000001      1# : ADD     #1, R5     ;DO LOOP UNTIL R5 = 0
2776 014514 001375      BNE      1#
2777 014516 000005      RESET
2778 014520 012706 001000      MOV     #1000, SP ;STOP EVERYONE SHOULD BE DONE
;RESET STACK AFTER LAST INTERRUPT
2779
2780 014524 032737 000001 001156 CHECK1: BIT     #BIT0, #DEVM ;SLU1 UNDER TEST
2781 014532 001424      BEQ     CHECK2 ;IF NO GO CHECK SLU2 DATA
2782 014534 023737 014726 014732      CMP     XMTCT1, RECCT1 ;# OF XMIT INTERRUPTS = REC INTERRUPTS
2783 014542 001401      BEQ     1#      ;IF YES GET OVER ERROR AND CHECK DATA
2784 014544 104155      ERROR  155      ;INTERRUPT COMPARISON ERROR
2785 014546 012702 030450      1# : MOV     #BUF1, R2 ;POINT TO FIRST DATA
2786 014552 005000      CLR     R0      ;INITIALIZE TO FIRST DATA XMIT
2787 014554 013704 014726      MOV     XMTCT1, R4 ;GET # OF BYTES TRANSFERRED
2788 014560 122200      2# : CMPB   (R2)+, R0 ;IS RECEIVED DATA = EXPECTED
2789 014562 001406      BEQ     3#      ;IF YES CONTINUE
2790 014564 114237 001026      MOVB   -(R2), #BDDAT ;IF NO GET ERROR INFORMATION
2791 014570 010037 001024      MOV     R0, #GDDAT
2792 014574 104156      ERROR  156      ;SLU1 DATA COMPARISON ERROR
2793 014576 005202      INC     R2      ;IF CONTINUE ON ERROR RESET POINTER
2794 014600 005200      3# : INC     R0      ;UPDATE TO NEXT GOOD DATA
2795 014602 077412      SOB    R4, 2# ;LOOP UNTIL ALL DATA CHECKED
2796 014604 032737 000002 001156 CHECK2: BIT     #BIT1, #DEVM ;SLU2 UNDER TEST
2797 014612 001424      BEQ     FINIE   ;IF NO WE'RE DONE
2798 014614 023737 014730 014734      CMP     XMTCT2, RECCT2 ;#OF XMIT INTERRUPTS = REC INTERRUPTS
2799 014622 001401      BEQ     1#      ;IF YES CHECK DATA
2800 014624 104157      ERROR  157      ;INTERRUPT COMPARISON ERROR
2801 014626 012703 031050      1# : MOV     #BUF2, R3 ;INITIALIZE TO FIRST RECEIVED DATA
2802 014632 005001      CLR     R1      ;INITIALIZE TO FIRST XMIT DATA
2803 014634 013704 014730      MOV     XMTCT2, R4 ;GET # OF BYTES TRANSFERRED
2804 014640 122301      2# : CMPB   (R3)+, R1 ;IS RECEIVED DATA = EXPECTED DATA
2805 014642 001406      BEQ     3#      ;IF YES CONTINUE TESTING
2806 014644 114337 001026      MOVB   -(R3), #BDDAT ;IF NO GET ERROR INFORMATION
2807 014650 010137 001024      MOV     R1, #GDDAT
2808 014654 104160      ERROR  160      ;SLU2 DATA COMPARISON ERROR
2809 014656 005203      INC     R3      ;IF COONTINUE ON ERROR RESET POINTER
2810 014660 005201      3# : INC     R0      ;UPDATE TO NEXT GOOD DATA
2811 014662 077412      SOB    R4, 2# ;LOOP UNTIL ALL DATA CHECKED
2812 014664 013777 001060 166124 FINIE: MOV     #TMP0, #TVECT ;RESTORE VECTORS
2813 014672 013777 001062 166112      MOV     #TMP1, #RVECT
2814 014700 013777 001064 166120      MOV     #TMP2, #CTVECT
2815 014706 013777 001066 166106      MOV     #TMP3, #CRVECT
2816 014714 013777 001070 166110      MOV     #TMP4, #RCTVCT
2817 014722 000137 014740      JMP     #EOP    ;FINISHED TESTING GO TO END OF PASS
2818
2819 014726 000000      XMTCT1: .WORD  0
2820 014730 000000      XMTCT2: .WORD  0
```

ft.

CJKDFB 11/24 OPTIONS DIAGNOSTIC  
CJKDFB P11 04 JAN 85 11:44

MACY11 30(1046) 04-JAN-85 11:58 PAGE 56  
170 TEST ALL INTERNAL OPTIONS SIMULTANEOUSLY

SEQ 0070

2821 014732 000000  
2822 014734 000000  
2823 014736 000000  
2824

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

```

2825
2826
2827
2828
2829
2830
2831
2832
2833
2834 014740
2835 014740 000004
2836 014742 005037 001002
2837 014746 005237 001106
2838 014752 042737 100000 001106
2839 014760 005327
2840 014762 000001
2841 014764 003022
2842 014766 012737
2843 014770 000001
2844 014772 014762
2845 014774 104401 015041
2846 015000 013746 001106
2847 015004 104405
2848 015006 104401 015036
2849 015012 013700 000042
2850 015016 001405
2851 015020 000005
2852 015022 004710
2853 015024 000240
2854 015026 000240
2855 015030 000240
2856 015032
2857 015032 000137
2858 015034 003450
2859 015036 377 377 000
2860 015041 015 042412 042116
2861 015046 050040 051501 020123
2862 015054 000043
2863
2864 015056 011646
2865 015060 013616
2866 015062 062746 000002
2867 015066 000002
2868
2869
2870
2871 015070 012600
2872 015072 162700 000004
2873 015076 010037 015116
2874 015102 016637 000002 015114
2875 015110 104161
2876
2877 015112 000000
2878 015114 000000
2879 015116 000000
2880

.SBTL 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 $TSTNM ; 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: .ASCIZ <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 EHRNEOUS 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),OLDCPC ; 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

```

2881  
 2882  
 2883  
 2884  
 2885  
 2886  
 2887  
 2888  
 2889  
 2890  
 2891  
 2892  
 2893  
 2894  
 2895  
 2896  
 2897  
 2898  
 2899  
 2900  
 2901  
 2902  
 2903  
 2904  
 2905  
 2906  
 2907  
 2908  
 2909  
 2910  
 2911  
 2912  
 2913  
 2914  
 2915  
 2916  
 2917  
 2918  
 2919  
 2920  
 2921  
 2922  
 2923  
 2924  
 2925  
 2926  
 2927  
 2928  
 2929  
 2930

015120  
 015120 105237 001003  
 015124 001775  
 015126 013777 001002 163706  
 015134 005237 001012  
 015140 011637 001016  
 015144 162737 000002 001016  
 015152 117737 163640 001014  
 015160 032777 020000 163652  
 015166 001004  
 015170 004737 015302  
 015174 104401 001075  
 015200  
 015200 122737 000001 001120  
 015206 001007  
 015210 113737 001014 015222  
 015216 004737 016004  
 015222 000  
 015223 000  
 015224 000777  
 015226 005777 163606  
 015232 100001  
 015234 000000  
 015236 104407  
 015240 032777 001000 163572  
 015246 001402  
 015250 013716 001010  
 015254 005737 001072  
 015260 001402  
 015262 013716 001072  
 015266  
 015266 022737 015022 000042  
 015274 001001  
 015276 000000  
 015300  
 015300 000002

```

;*****
; 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,$ERRPC
        MOV      @($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$:
        CMP      @APTENV,$ENV   ;RUNNING IN APT MODE
        BNE      21$          ;NO, SKIP APT ERROR REPORT
        MOV      $ITEMB,21$    ;SET ITEM NUMBER AS ERROR NUMBER
        JSR      PC,$ATY4      ;REPORT FATAL ERROR TO APT

21$:      .BYTE      0
        .BYTE      0

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

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

6$:      RTI                  ;RETURN
  
```



2931  
 2932  
 2933  
 2934  
 2935  
 2936  
 2937  
 2938  
 2939 015302  
 2940 015302 104401 001075  
 2941 015306 010046  
 2942 015310 005000  
 2943 015312 153700 001014  
 2944 015316 001004  
 2945  
 2946 015320 013746 001016  
 2947  
 2948 015324 104402  
 2949 015326 000426  
 2950 015330 005300  
 2951 015332 006300  
 2952 015334 006300  
 2953 015336 006300  
 2954 015340 062700 001160  
 2955 015344 012037 015354  
 2956 015350 001404  
 2957 015352 104401  
 2958 015354 000000  
 2959 015356 104401 001075  
 2960 015362 012037 015372  
 2961 015366 001404  
 2962 015370 104401  
 2963 015372 000000  
 2964 015374 104401 001075  
 2965 015400 011000  
 2966 015402 001004  
 2967 015404 012600  
 2968 015406 104401 001075  
 2969 015412 000207  
 2970 015414  
 2971 015414 013046  
 2972 015416 104402  
 2973 015420 005710  
 2974 015422 001770  
 2975 015424 104401 015432  
 2976 015430 000771  
 2977 015432 020040 000  
 2978 015436  
 2979

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

```

2980
2981 .SBTTL POWER DOWN AND UP ROUTINES
2982 ;:*****
2983 ;*POWER DOWN ROUTINE
2984 ;:*****
2985 015436 012737 015576 000024 $PWRDN: MOV    ##ILLUP, @PWRVEC ;SET FOR FAST UP
2986 015444 012737 000340 000026      MOV    @340, @PWRVEC+2 ;PRIO:7
2987 015452 010046      MOV    R0, -(SP)      ;PUSH R0 ON STACK
2988 015454 010146      MOV    R1, -(SP)      ;PUSH R1 ON STACK
2989 015456 010246      MOV    R2, -(SP)      ;PUSH R2 ON STACK
2990 015460 010346      MOV    R3, -(SP)      ;PUSH R3 ON STACK
2991 015462 010446      MOV    R4, -(SP)      ;PUSH R4 ON STACK
2992 015464 010546      MOV    R5, -(SP)      ;PUSH R5 ON STACK
2993 015466 017746 163346      MOV    @SWR, -(SP)    ;PUSH @SWR ON STACK
2994 015472 010637 015602      MOV    SP, $SAVR6     ;SAVE SP
2995 015476 012737 015510 000024      MOV    ##PWRUP, @PWRVEC ;SET UP VECTOR
2996 015504 000000      HALT
2997 015506 000776      BR     .-2           ;HANG UP
2998
2999
3000 ;:*****
3001 ;*POWER UP ROUTINE
3002 ;:*****
3003 015510 012737 015576 000024 $PWRUP: MOV    ##ILLUP, @PWRVEC ;SET FOR FAST DOWN
3004 015516 013706 015602      MOV    $SAVR6, SP     ;GET SP
3005 015522 012677 163312      MOV    (SP)+, @SWR    ;POP STACK INTO @SWR
3006 015526 012605      MOV    (SP)+, R5
3007 015530 012604      MOV    (SP)+, R4     ;POP STACK INTO R4
3008 015532 012603      MOV    (SP)+, R3     ;POP STACK INTO R3
3009 015534 012602      MOV    (SP)+, R2     ;POP STACK INTO R2
3010 015536 012601      MOV    (SP)+, R1     ;POP STACK INTO R1
3011 015540 012600      MOV    (SP)+, R0     ;POP STACK INTO R0
3012 015542 012737 015436 000024      MOV    ##PWRDN, @PWRVEC ;SET UP THE POWER DOWN VECTOR
3013 015550 012737 000340 000026      MOV    @340, @PWRVEC+2 ;PRIO:7
3014 015556 005037 015602      CLR    $SAVR6        ;WAIT LOOP FOR THE TTY
3015 015562 005237 015602      1$: INC    $SAVR6     ;WAIT FOR THE INC
3016 015566 001375      BNE    1$            ;OF WORD
3017 015570 104401      TYPE                                ;REPORT THE POWER FAILURE
3018 015572 015604      $PWRMG: .WORD    $POWER ;POWER FAIL MESSAGE POINTER
3019 015574 000002      RTI
3020 015576 000000      $ILLUP: HALT        ;THE POWER UP SEQUENCE WAS STARTED
3021 015600 000776      BR     .-2           ; BEFORE THE POWER DOWN WAS COMPLETE
3022 015602 000000      $SAVR6: 0           ;PUT THE SP HERE
3023 015604 005015 047520 042527 $POWER: .ASCIZ  <15><12>"POWER"
3024 015612 000122
3025

```

3026  
 3027  
 3028  
 3029  
 3030  
 3031  
 3032  
 3033  
 3034  
 3035  
 3036  
 3037  
 3038  
 3039 015614  
 3040 015614 104407  
 3041 015616 032777 040000 163214  
 3042 015624 001052  
 3043  
 3044 015626 000416  
 3045  
 3046 015630 013746 000004  
 3047 015634 012737 015654 000004  
 3048 015642 005737 177060  
 3049 015646 012637 000004  
 3050 015652 000421  
 3051 015654 022626  
 3052 015656 012637 000004  
 3053 015662 000407  
 3054 015664  
 3055 015664 105737 001003  
 3056 015670 001412  
 3057 015672 032777 001000 163140  
 3058 015700 001404  
 3059 015702 013737 001010 001006  
 3060 015710 000420  
 3061 015712 105037 001003  
 3062 015716 105237 001002  
 3063 015722 113737 001002 001104  
 3064 015730 011637 001006  
 3065 015734 011637 001010  
 3066 015740 005037 001072  
 3067 015744 112737 000001 001015  
 3068 015752 013777 001002 163062  
 3069 015760 013716 001006  
 3070 015764 000002  
 3071

.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)
                        ;;SAVE THE CONTENTS OF THE ERROR VECTOR
    MOV    @ERRVEC,-(SP)
    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*****
    TSTB   $ERFLG     ;;HAS AN ERROR OCCURRED?
    BEQ    $SVLAD     ;;BR IF NO
    BIT    #BIT09,$SWR ;;LOOP ON ERROR?
    BEQ    4$        ;;BR IF NO
7$:  MOV    $LPER', $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

```

```
3072  
3073  
3074  
3075  
3076  
3077 015766 112737 000001 01232 $ATY1: MOVB #1,$FFLG ;TO REPORT FATAL ERROR  
3078 015774 112737 000001 016230 $ATY3: MOVB #1,$MFLG ;TO TYPE A MESSAGE  
3079 016002 000403  
3080 016004 112737 000001 016232 $ATY4: MOVB #1,$FFLG ;TO ONLY REPORT FATAL ERROR  
3081 016012 $ATYC:  
3082 016012 010046 MOV RO,-(SP) ;PUSH RO ON STACK  
3083 016014 010146 MOV R1,-(SP) ;PUSH R1 ON STACK  
3084 016016 105737 016230 TSTB $MFLG ;SHOULD TYPE A MESSAGE?  
3085 016022 001450 BEQ 5# ;IF NOT: BR  
3086 016024 122737 000001 001120 CMPB #APTENV,$ENV ;OPERATING UNDER APT?  
3087 016032 001031 BNE 3# ;IF NOT: BR  
3088 016034 132737 000100 001121 BITB #APTSPOOL,$ENVM ;SHOULD SPOOL MESSAGE?  
3089 016042 001425 BEQ 3# ;IF NOT: BR  
3090 016044 017600 000004 MOV #4(SP),RO ;GET MESSAGE ADDRESS  
3091 016050 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS  
3092 016056 005737 001100 1# TST $MSGTYPE ;SEE IF DONE W/ LAST XMISSION?  
3093 016062 001375 BNE 1# ;IF NOT: WAIT  
3094 016064 010037 001114 MOV RO,$MSGAD ;PUT ADDRESS IN MAILBOX  
3095 016070 105720 2# TSTB (RO)+ ;FIND END OF MESSAGE  
3096 016072 001376 BNE 2#  
3097 016074 163700 001114 SUB $MSGAD,RO ;SUB START OF MESSAGE  
3098 016100 006200 ASR RO ;GET MESSAGE LENGTH IN WORDS  
3099 016102 010037 001116 MOV RO,$MSGLGT ;PUT LENGTH IN MAILBOX  
3100 016106 012737 000004 001100 MOV #4,$MSGTYPE ;TELL APT TO TAKE MESSAGE  
3101 016114 000413 BR 5#  
3102 016116 017637 000004 016142 3# MOV #4(SP),4# ;PUT MSG ADDR IN JSR LINKAGE  
3103 016124 062766 000002 000004 ADD #2,4(SP) ;BUMP RETURN ADDRESS  
3104 016132 013746 177776 MOV 177776,-(SP) ;PUSH 177776 ON STACK  
3105 016136 004737 016234 JSR PC,$TYPE ;CALL TYPE MACRO  
3106 016142 000000 4# .WORD 0  
3107 016144 5#  
3108 016144 105737 016232 10# TSTB $FFLG ;SHOULD REPORT FATAL ERROR?  
3109 016150 001413 BEQ 12# ;IF NOT: BR  
3110 016152 005737 001120 TST $ENV ;RUNNING UNDER APT?  
3111 016156 001410 BEQ 12# ;IF NOT: BR  
3112 016160 005737 001100 11# TST $MSGTYPE ;FINISHED LAST MESSAGE?  
3113 016164 001375 BNE 11# ;IF NOT: WAIT  
3114 016166 017637 000004 001102 MOV #4(SP),$FATAL ;GET ERROR #  
3115 016174 005237 001100 INC $MSGTYPE ;TELL APT TO TAKE ERROR  
3116 016200 062766 000002 000004 12# ADD #2,4(SP) ;BUMP RETURN ADDRESS  
3117 016206 105037 016232 CLRB $FFLG ;CLEAR FATAL FLAG  
3118 016212 105037 016231 CLRB $LFLG ;CLEAR LOG FLAG  
3119 016216 105037 016230 CLRB $MFLG ;CLEAR MESSAGE FLAG  
3120 016222 012601 MOV (SP)+,R1 ;POP STACK INTO R1  
3121 016224 012600 MOV (SP)+,RO ;POP STACK INTO R1  
3122 016226 000207 RTS PC ;RETURN  
3123 016230 000 $MFLG: .BYTE 0  
3124 016231 000 $LFLG: .BYTE 0 ;LOG FLAG  
3125 016232 000 $FFLG: .BYTE 0 ;FATAL FLAG  
3126  
3127 016234 .EVEN
```

CJKDFB0 11/24 OPTIONS DIAGNOSTIC  
CJKDFB.P11 04-JAN-85 11:44

MACY11 30(1046) 04-JAN 85 11:58 PAGE 63  
APT COMMUNICATIONS ROUTINE

SEQ 0077

3128 000200  
3129 000001  
3130 000100  
3131 000040  
3132

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

```

3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3:50
3151 016234 105737 001057
3152 016240 100002
3153 016242 000000
3154 016244 000430
3155 016246 010046
3156 016250 017600 000002
3157 016254 122737 000001 001120
3158 016262 001011
3159 016264 132737 000100 001121
3160 016272 001405
3161 016274 010037 016304
3162 016300 004737 015774
3163 016304 000000
3164 016306 132737 000040 001121
3165 016314 001003
3166 016316 112046
3167 016320 001005
3168 016322 005726
3169 016324 012600
3170 016326 062716 000002
3171 016332 000002
3172 016334 122716 000011
3173 016340 001430
3174 016342 122716 000200
3175 016346 001006
3176 016350 005726
3177 016352 104401
3178 016354 001075
3179 016356 105037 016564
3180 016362 000755
3181 016364 004737 016446
3182 016370 123726 001056
3183 016374 001350
3184 016376 013746 001054
3185
3186 016402 105366 000001
3187 016406 002770
3188 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
RMC 60$ ;YES,SKIP TYPE OUT
2$: (RO)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
L 4$ ;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

```

```
3189 016414 105337 016564          DECB          $CHARCNT          ;;DO NOT COUNT AS A COUNT
3190 016420 000770                    BR          78          ;;LOOP
3191
3192          ;HORIZONTAL TAB PROCESSOR
3193
3194 016422 112716 000040          88:  MOVB      0' ,(SP)          ;;REPLACE TAB WITH SPACE
3195 016426 004737 016446          98:  JSR      PC,$TYPEC          ;;TYPE A SPACE
3196 016432 132737 000007 016564          BITB      07,$CHARCNT          ;;BRANCH IF NOT AT
3197 016440 001372                    BNE      98          ;;TAB STOP
3198 016442 005726                    TST      (SP)          ;;POP SPACE OFF STACK
3199 016444 000724                    BR       28          ;;GET NEXT CHARACTER
3200 016446          $TYPEC:
3201 016446 105777 162372          TSTB     88TKS          ;;CHAR IN KYBD BUFFER?          ;MJD001
3202 016452 100022                    BPL      108          ;;BR IF NOT          ;MJD001
3203 016454 017746 162366          MOV      88TKB,-(SP)          ;;GET CHAR          ;MJD001
3204 016460 042716 177600          BIC      0177600,(SP)          ;;STRIP EXTRANEIOUS BITS          ;MJD001
3205 016464 122716 000023          CMPB     08XOFF,(SP)          ;;WAS CHAR XOFF          ;MJD001
3206 016470 001012                    BNE      1028          ;;BR IF NOT          ;MJD001
3207 016472
3208 016472 105777 162346          1018:  TSTB     88TKS          ;;WAIT FOR CHAR          ;MJD001
3209 016476 100375                    BPL      1018          ;MJD001
3210 016500 117716 162342          MOVB     88TKB,(SP)          ;;GET CHAR          ;MJD001
3211 016504 042716 177600          BIC      0177600,(SP)          ;;STRIP IT          ;MJD001
3212 016510 122716 000021          CMPB     08XON,(SP)          ;;WAS IT XON?          ;MJD001
3213 016514 001366                    BNE      1018          ;;BR IF NOT          ;MJD001
3214 016516
3215 016516 005726          1028:  TST      (SP)          ;;FIX STACK          ;MJD001
3216 016520
3217 016520 105777 162324          108:  TSTB     88TPS          ;;WAIT UNTIL PRINTER IS READY          ;MJD001
3218 016524 100375                    BPL      108          ;MJD001
3219 016526 116677 000002 162316          MOVB     2(SP),88TPB          ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3220 016534 122766 000015 000002          CMPB     0CR,2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
3221 016542 001003                    BNE      18          ;;BRANCH IF NO
3222 016544 105037 016564          CLRB     $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
3223 016550 000406                    BR       $TYPEX          ;;EXIT
3224 016552 122766 000012 000002 18:  CMPB     0LF,2(SP)          ;;IS CHARACTER A LINE FEED?
3225 016560 001402                    BEQ      $TYPEX          ;;BRANCH IF YES
3226 016562 105227                    INCB     (PC)          ;;COUNT THE CHARACTER
3227 016564 000000          $CHARCNT: .WORD          0          ;;CHARACTER COUNT STORAGE
3228 016566 000207          $TYPEX: RTS          PC
3229
3230          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3231
3232          ;*****
3233          ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3234          ;OCTAL (ASCII) NUMBER AND TYPE IT.
3235          ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3236          ;CALL:
3237          ;     MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
3238          ;     TYPOS          ;;CALL FOR TYPEOUT
3239          ;     .BYTE   N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3240          ;     .BYTE   M          ;;M=1 OR 0
3241          ;
3242          ;           ;1=TYPE LEADING ZEROS
3243          ;           ;0=SUPPRESS LEADING ZEROS
3244          ;
3244          ;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
```

3245										
3246										
3247										
3248										
3249										
3250										
3251										
3252										
3253										
3254										
3255	016570	017646	000000							
3256	016574	116637	000001	017013						
3257	016602	112637	017015							
3258	016606	062716	000002							
3259	016612	000406								
3260	016614	112737	000001	017013						
3261	016622	112737	000006	017015						
3262	016630	112737	000005	017012						
3263	016636	010346								
3264	016640	010446								
3265	016642	010546								
3266	016644	113704	017015							
3267	016650	005404								
3268	016652	062704	000006							
3269	016656	110437	017014							
3270	016662	113704	017013							
3271	016666	016605	000012							
3272	016672	005003								
3273	016674	006105								
3274	016676	000404								
3275	016700	006105								
3276	016702	006105								
3277	016704	006105								
3278	016706	010503								
3279	016710	006103								
3280	016712	105337	017014							
3281	016716	100016								
3282	016720	042703	177770							
3283	016724	001002								
3284	016726	005704								
3285	016730	001403								
3286	016732	005204								
3287	016734	052703	000060							
3288	016740	052703	000040							
3289	016744	110337	017010							
3290	016750	104401	017010							
3291	016754	105337	017012							
3292	016760	003347								
3293	016762	002402								
3294	016764	005204								
3295	016766	000744								
3296	016770	012605								
3297	016772	012604								
3298	016774	012603								
3299	016776	016666	000002	000004						
3300	017004	012616								

```

;: $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
;:          MOVB   1 (SP), @FILL   ;: LOAD ZERO FILL SWITCH
;:          MOVB   (SP)+, @MODE+1 ;: NUMBER OF DIGITS TO TYPE
;:          ADD    @2, (SP)        ;: ADJUST RETURN ADDRESS
;:          BR     $TYPON
;: $TYPOC: MOVB   @1, @FILL       ;: SET THE ZERO FILL SWITCH
;:          MOVB   @6, @MODE+1    ;: SET FOR SIX(6) DIGIT
;: $TYPON: MOVB   @5, @CNT        ;: SET THE ITERATION COUNT
;:          MOV    R3, -(SP)       ;: SAVE R3
;:          MOV    R4, -(SP)       ;: SAVE R4
;:          MOV    R5, -(SP)       ;: SAVE R5
;:          MOVB   @MODE+1, R4     ;: GET THE NUMBER OF DIGITS TO TYPE
;:          NEG    R4
;:          ADD    @6, R4          ;: SUBTRACT IT FOR MAX. ALLOWED
;:          MOVB   R4, @MODE      ;: SAVE IT FOR USE
;:          MOVB   @FILL, 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
;:          ROL   R5
;:          ROL   R5
;:          MOV   R5, R3
;:          ROL   R3              ;: GET LSB OF THIS DIGIT
;:          DECB  @MODE          ;: 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
;:          MOVB R3, @4          ;: SAVE FOR TYPING
;:          TYPE , @4           ;: GO TYPE THIS DIGIT
;:          DECB @CNT           ;: 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)

```



3301 017005 000002  
 3302 017010 000  
 3303 017011 000  
 3304 017012 000  
 3305 017013 000  
 3306 017014 000000  
 3307  
 3308  
 3309  
 3310  
 3311  
 3312  
 3313  
 3314  
 3315  
 3316  
 3317  
 3318  
 3319 017016  
 3320 017016 010046  
 3321 017020 010146  
 3322 017022 010246  
 3323 017024 010346  
 3324 017026 010546  
 3325 017030 012746 020200  
 3326 017034 016605 000020  
 3327 017040 100004  
 3328 017042 005405  
 3329 017044 112766 000055 000001  
 3330 017052 005000  
 3331 017054 012703 017232  
 3332 017060 112723 000040  
 3333 017064 005002  
 3334 017066 016001 017222  
 3335 017072 160105  
 3336 017074 002402  
 3337 017076 005202  
 3338 017100 000774  
 3339 017102 060105  
 3340 017104 005702  
 3341 017106 001002  
 3342 017110 105716  
 3343 017112 100407  
 3344 017114 106316  
 3345 017116 103003  
 3346 017120 116663 000001 177777  
 3347 017126 052702 000060  
 3348 017132 052702 000040  
 3349 017136 110223  
 3350 017140 005720  
 3351 017142 020027 000010  
 3352 017146 002746  
 3353 017150 003002  
 3354 017152 010502  
 3355 017154 000764  
 3356 017156 105726

```

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

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

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

3357	017160	100003			BPL	98	::BR IF NO
3358	017162	116663	177777	177776	MOV	-1(SP),-2(R3)	::YES--SET THE SIGN FOR TYPING
3359	017170	105013			CLRB	(R3)	::SET THE TERMINATOR
3360	017172	012605			MOV	(SP),R5	::POP STACK INTO R5
3361	017174	012603			MOV	(SP),R3	::POP STACK INTO R3
3362	017176	012602			MOV	(SP),R2	::POP STACK INTO R2
3363	017200	012601			MOV	(SP),R1	::POP STACK INTO R1
3364	017202	012600			MOV	(SP),R0	::POP STACK INTO R0
3365	017204	104401	017232		TYPE	,#DBLK	::NOW TYPE THE NUMBER
3366	017210	016666	000002	000004	MOV	2(SP),4(SP)	::ADJUST THE STACK
3367	017216	012616			MOV	(SP),(SP)	
3368	017220	000002			RTI		::RETURN TO USER
3369	017222	023420			#DTBL:	10000.	
3370	017224	001750				1000.	
3371	017226	000144				100.	
3372	017230	000012				10.	
3373	017232	000004			#DBLK:	.BLKW 4	
3374							

```

3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386 017242 022737 000176 001040
3387 017250 001074
3388 017252 105777 161566
3389 017256 100071
3390 017260 117746 161562
3391 017264 042716 177600
3392 017270 022726 000007
3393 017274 001062
3394 017276 123727 001034 000001
3395 017304 001456
3396
3397 017306 104401 017777
3398 017312 104401 020004
3399 017316 013746 000176
3400 017322 104402
3401 017324 104401 020015
3402 017330 005046
3403 017332 005046
3404 017334 105777 161504
3405 017340 100375
3406
3407 017342 117746 161500
3408 017346 042716 177600
3409
3410
3411
3412 017352 021627 000025
3413 017356 001005
3414 017360 104401 017772
3415 017364 062706 000006
3416 017370 000757
3417
3418
3419 017372 021627 000015
3420 017376 001022
3421 017400 005766 000004
3422 017404 001403
3423 017406 016677 000002 161424
3424 017414 062706 000006
3425 017420 104401 001075
3426 017424 123727 001035 000001
3427 017432 001003
3428 017434 012777 000100 161402
3429 017442 000002
3430 017444 004737 016446

```

```

.SBTTL TTY INPUT ROUTINE
;*****
.ENABL LSB
;*****
;SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
;ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
;SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
;WHEN OPERATING IN TTY FLAG MODE.
%CKSWR: CMP    #SWREG,SWR    ;; IS THE SOFT-SWR SELECTED?
        BNE    15;         ;; BRANCH IF NO
        TSTB   #TKS        ;; CHAR THERE?
        BPL    15;         ;; IF NO, DON'T WAIT AROUND
        MOVB   #TKB,-(SP)   ;; SAVE THE CHAR
        BIC    #C177,(SP)  ;; STRIP-OFF THE ASCII
        CMP    #7,(SP);    ;; IS IT A CONTROL G?
        BNE    15;         ;; NO, RETURN TO USER
        CMPS   #AUTOB,#1   ;; ARE WE RUNNING IN AUTO-MODE?
        BEQ    15;         ;; BRANCH IF YES
;*****
%GTSWR: TYPE   ,#CNTLG     ;; ECHO THE CONTROL-G (*G)
        TYPE   ,#MSWR     ;; TYPE CURRENT CONTENTS
        MOV    SWREG,-(SP) ;; SAVE SWREG FOR TYPEOUT
        TYPOC  ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
        TYPE   ,#MNEW     ;; PROMPT FOR NEW SWR
        CLR    -(SP)      ;; CLEAR COUNTER
        CLR    -(SP)      ;; THE NEW SWR
        TSTB   #TKS      ;; CHAR THERE?
        BPL    7;        ;; IF NOT TRY AGAIN
;*****
        MOVB   #TKB,-(SP); ;; PICK UP CHAR
        BIC    #C177,(SP) ;; MAKE IT 7-BIT ASCII
;*****
        CMP    (SP),#25   ;; IS IT A CONTROL-U?
        BNE    10;       ;; BRANCH IF NOT
        TYPE   ,#CNTLU   ;; YES, ECHO CONTROL-U (*U)
        ADD    #6,SP     ;; IGNORE PREVIOUS INPUT
        BR     19;       ;; LET'S TRY IT AGAIN
;*****
        CMP    (SP),#15  ;; IS IT A <CR>?
        BNE    16;       ;; BRANCH IF NO
        TST    4(SP)     ;; YES, IS IT THE FIRST CHAR?
        BEQ    11;       ;; BRANCH IF YES
        MOV    2(SP),#SWR ;; SAVE NEW SWR
        ADD    #6,SP     ;; CLEAR UP STACK
        TYPE   ,#CRLF    ;; ECHO <CR> AND <LF>
        CMPS   #INTAG,#1 ;; RE-ENABLE TTY KBD INTERRUPTS?
        BNE    15;       ;; BRANCH IF NOT
        MOV    #100,#TKS ;; RE-ENABLE TTY KBD INTERRUPTS
        RTI
        JSR    PC,#TYPEC ;; ECHO CHAR

```

```

3431 017450 021627 000060          CMP      (SP),#60          ;;CHAR < 0?
3432 017454 002420          BLT      18;              ;;BRANCH IF YES
3433 017456 021627 000067          CMP      (SP),#67          ;;CHAR > 7?
3434 017462 003015          BGT      18;              ;;BRANCH IF YES
3435 017464 042726 000060          BIC      #60,(SP)         ;;STRIP-OFF ASCII
3436 017470 005766 000002          TST      2(SP)            ;;IS THIS THE FIRST CHAR
3437 017474 001403          BEQ      17;              ;;BRANCH IF YES
3438 017476 006316          ASL      (SP)             ;;NO, SHIFT PRESENT
3439 017500 006*16          ASL      (SP)             ;; CHAR OVER TO MAKE
3440 017502 006J16          ASL      (SP)             ;; ROOM FOR NEW ONE.
3441 017504 005266 000002          17;: INC      2(SP)            ;;KEEP COUNT OF CHAR
3442 017510 056616 177776          BIS      -2(SP),(SP)     ;;SET IN NEW CHAR
3443 017514 000707          BR       7;              ;;GET THE NEXT ONE
3444 017516 104401 001074          18;: TYPE     ,#QUES      ;;TYPE ?<CR><LF>
3445 017522 000720          BR       20;              ;;SIMULATE CONTROL U
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457 017524 011646          $RDCHR: MOV      (SP),-(SP)  ;;PUSH DOWN THE PC
3458 017526 016666 000004 000002  MOV      4(SP),2(SP)     ;;SAVE THE PS
3459 017534 105777 161304 18;: TSTB     @1TKS        ;;WAIT FOR
3460 017540 100375          BPL      1;              ;;A CHARACTER
3461 017542 117766 161300 000004  MOVB     @1TKB,4(SP)     ;;READ THE TTY
3462 017550 042766 177600 000004  BIC      @+C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
3463 017556 026627 000004 000023  CMP      4(SP),#23      ;;IS IT A CONTROL S?
3464 017564 001013          BNE      3;              ;;BRANCH IF NO
3465 017566 105777 161252 24;: TSTB     @1TKS        ;;WAIT FOR A CHARACTER
3466 017572 100375          BPL      2;              ;;LOOP UNTIL ITS THERE
3467 017574 117746 161246  MOVB     @1TKB,-(SP)     ;;GET CHARACTER
3468 017600 042716 177600  BIC      @+C177,(SP)    ;;MAKE IT 7-BIT ASCII
3469 017604 022627 000021  CMP      (SP),#21      ;;IS IT A CONTROL-Q?
3470 017610 001366          BNE      2;              ;;IF NOT DISCARD IT
3471 017612 000750          BR       1;              ;;YES, RESUME
3472 017614 026627 000004 000021 34;: CMP      4(SP),#1XON   ;;IS IT A RANDOM XON?
3473 017622 001744          BEQ      1;              ;;BRANCH IF YES ;RAN001
3474 017624 026627 000004 000140  CMP      4(SP),#140     ;;IS IT UPPER CASE? ;RAN001
3475 017632 002407          BLT      4;              ;;BRANCH IF YES
3476 017634 026627 000004 000175  CMP      4(SP),#175     ;;IS IT A SPECIAL CHAR?
3477 017642 003003          BGT      4;              ;;BRANCH IF YES
3478 017644 042766 000040 000004  BIC      #40,4(SP)      ;;MAKE IT UPPER CASE
3479 017652 000002          44;: RTI                    ;;GO BACK TO USER
3480
3481
3482
3483
3484
3485
3486

```

;;\*\*\*\*\*  
 ;\*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  
 ;

34;: ;;IS IT A RANDOM XON? ;RAN001  
 ;;BRANCH IF YES ;RAN001

;;\*\*\*\*\*  
 ;\*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



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

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

```

3563          .SBTTL ECHO TEST
3564          ;*****
3565          ;*THIS ROUTINE WILL ECHO ANY CHARACTER TYPED
3566          ;*ON EITHER SLU1 OR SLU2. DEFAULT IS TO THE CONSOLE DEVICE SLU1.
3567          ;*THE TEST IS HALTED BY TYPING A CONTROL-C
3568          ;*TEST CAN BE RESTARTED AFTER HALTING BY JUST CONTINUING
3569          ;*****
3570 020106 012737 000176 001040 ECHO: MOV    #SWREG,SWR          ;SET UP FOR SOFTWARE SWITCH REGISTER
3571 020114 012737 000174 001042      MOV    #DISPREG,DISPLAY      ;AND DISPLAY REGISTER
3572 020122 032777 00002C 160710      BIT    #BIT4, #SWR          ;CHECK IF BIT4 SET IN SWITCH REG
3573 020130 001403                BEQ    14                    ;IF NOT THEN SELECT SLU1
3574 020132 012703 002770      MOV    #RCSR, R3           ;IF BIT4 SET THEN SELECT SLU2
3575 020136 000402                BR     24
3576 020140 012703 003000      14:   MOV    #RCSR, R3           ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
3577 020144 000005                24:   RESET                   ;CLEAR EVERYTHING
3578 020146 112773 000052 000006      MOVB  #' ',B6(R3)          ;TRANSMIT PROMPT " "
3579 020154 105773 000000      34:   TSTB  B(R3)              ;WAIT FOR INPUT
3580 020160 100375                BPL    34
3581 020162 117373 000002 000006      MOVB  B2(R3),B6(R3)        ;ECHO INPUT
3582 020170 017300 000002      MOV    B2(R3),R0           ;STORE INPUT
3583 020174 100023                BPL    64
3584 020176 052701 010000      BIS    #BIT12,R1           ;SET PARITY ERROR TEST MASK
3585 020202 030100                BIT    R1,R0               ;CHECK FOR PARITY ERROR FLAG
3586 020204 001403                BEQ    44
3587 020206 004737 020270      JSR    PC,MSG              ;REPORT PARITY ERROR
3588 020212 020316                MPAR
3589 020214 006301                44:   ASL    R1                  ;SHIFT MASK TO TEST "FR" FLAG
3590 020216 030100                BIT    R1,R0               ;TEST FOR FRAMING ERROR FLAG
3591 020220 001403                BEQ    54
3592 020222 004737 020270      JSR    PC,MSG              ;REPSORT FRAMING ERROR
3593 020226 020327                MFR
3594 020230 006301                54:   ASL    R1                  ;SHIFT MASK TO TEST "OR" FLAG
3595 020232 030100                BIT    R1,R0               ;TEST FOR OVERFLOW ERROR
3596 020234 001403                BEQ    64
3597 020236 004737 020270      JSR    PC,MSG              ;REPORT OVERFLOW ERROR
3598 020242 020341                MOR
3599 020244 042700 000200      64:   BIC    #BIT7,R0           ;CLEAR ANY PARITY BIT
3600 020250 022700 000003      CMP    #3,R0               ;WAS INPUT CONTROL-C
3601 020254 001337                BNE    34
3602 020256 004737 020270      JSR    PC,MSG              ;REPORT PROGRAM STOP
3603 020262 020354                MSTOP
3604 020264 000000                HALT
3605 020266 000707                BR     ECHO                ;END OF TEST HALT
3606                                     ;AFTER END OF TEST HALT
3607                                     ;PRESS CONTINUE TO RESTART ECHO TEST
3608 020270 013600                MSG:  MOV    B(SP),R0         ;PICK UP MESSAGE POINTER
3609 020272 062746 000002      ADD    #2,-(SP)            ;ADJUST RETURN PC
3610 020276 105773 000004      WAIT: TSTB  B4(R3)         ;WAIT FOR XMIT DONE
3611 020302 100375                BPL    WAIT
3612 020304 112073 000006      MOVB  (R0),B6(R3)         ;SEND CHARACTER
3613 020310 105710                TSTB  (R0)                 ;IS THIS END OF MESSAGE?
3614 020312 001371                BNE    WAIT                ;BR IF NOT
3615 020314 000207                RTS    PC                  ;RETURN
3616
3617 020316 005015 040520 044522 MPAR: .ASCIZ <CR><LF>/PARITY/
3618 020324 054524 000

```

3619	020327	015	043012	040522	MFR:	.ASCIZ	<CR><LF>/FRAMING/
3620	020334	044515	043516	000			
3621	020341	015	047412	042526	MOR:	.ASCIZ	<CR><LF>/OVERFLOW/
3622	020346	043122	047514	000127			
3623	020354	005015	052123	050117	MSTOP:	.ASCIZ	<CR><LF>/STOP/
3624	020362	000					



```

3625
3626
3627          020364          .EVEN
3628
3629          .SBTTL  TERMINAL OUTPUT TEST
3630
3631          ;*****
3632          ;*THIS ROUTINE WILL OUTPUT ALL WRITABLE CHARACTERS FOR THE
3633          ;*THE OCTAL CODE 040 --> 377
3634          ;*32 CHARACTERS ARE PRINTED ON EACH LINE
3635          ;*THE PATTERN IS REPEATED EVERY THREE LINES
3636          ;*
3637          ;*****
3638
3639          020364  012737  000176  001040  OUTTST:  MOV    #SWREG,SWR          ;SET UP FOR SOFTWARE SWITCH REGISTER
3640          020372  012737  000174  001042          MOV    #DISPREG,DISPLAY      ;AND DISPLAY REGISTER
3641          020400  032777  000020  160432          BIT    #BIT4, #SWR          ;CHECK IF BIT4 SET IN SWITCH REG
3642          020406  001403                      BEQ    1$                    ;IF NOT THEN SELECT SLU1
3643          020410  012703  002770          MOV    #RCSR, R3            ;IF BIT4 SET THEN SELECT SLU2
3644          020414  000402                      BR     2$
3645          020416  012703  003000          1$:  MOV    #RCSR, R3          ;SELECT SLU1 (THIS IS DEFAULT DEVICE)
3646          020422  000005                      2$:  RESET                   ;CLEAR THE WORLD
3647          020424  012701  000040          3$:  MOV    #40,R1            ;LOAD FIRST WRITABLE CHARACTER
3648          020430  012700  000040          4$:  MOV    #40,R0            ;LOAD CHAR COUNT PER LINE
3649          020434  105773  000004          5$:  TSTB   #4(R3)            ;WAIT FOR DONE
3650          020440  100375                      BPL   5$
3651          020442  010173  000006          MOV    R1,#6(R3)           ;TRANSMIT A CHARACTER
3652          020446  105201                      INCB  R1                    ;INCREMENT CHARACTER CODE
3653          020450  005300                      DEC   R0                    ;DECREMENT CHAR COUNT
3654          020452  001370                      BNE   5$                    ;BR IF LINE NOT COMPLETE
3655          020454  004737  020270          JSR   PC,MSG               ;SSUE CR,LINE FEED
3656
3657          020460  001075                      $CRLF
3658          020462  105773  000000          TSTB   #0(R3)              ;ANY CHARACTER RECEIVED?
3659          020466  100404                      BMI   6$                    ;BR IF YES
3660          020470  032701  000200          BIT    #BIT7,R1            ;FINISHED ONE PASS OF WRITABLE CHARACTERS?
3661          020474  001353                      BNE   3$                    ;BR IF YES
3662          020476  000754                      BR    4$                    ;IF NOT WRITE NEXT LINE
3663
3664          020500  005073  000002          6$:  CLR    #2(R3)            ;CLEAR RECEIVER
3665          020504  000000                      HALT
3666          020506  000726                      BR    OUTTST               ;RESTART TEST IF CONTINUED

```

3667						
3668						
3669	020510	046123	030525	052040	EM5:	.ASCIZ /SLU1 TCSR DONE NOT SET WITH RESET
3670	020516	051503	020122	047504		
3671	020524	042516	047040	052117		
3672	020532	051440	052105	053440		
3673	020540	052111	020110	042522		
3674	020546	042523	000124			
3675	020552	046123	030525	051040	EM6:	.ASCIZ /SLU1 RCSR DID NOT RETURN SSYNC/
3676	020560	051503	020122	044504		
3677	020566	020104	047516	020124		
3678	020574	042522	052524	047122		
3679	020602	051440	054523	041516		
3680	020610	000				
3681	020611	123	052514	020061	EM7:	.ASCIZ /SLU1 RBUF DID NOT RETURN SSYNC/
3682	020616	041122	043125	042040		
3683	020624	042111	047040	052117		
3684	020632	051040	052105	051125		
3685	020640	020116	051523	047131		
3686	020646	000103				
3687	020650	040503	020116	047516	EM11:	.ASCIZ /CAN NOT SET BIT2 OF SLU1 TCSR/
3688	020656	020124	042523	020124		
3689	020664	044502	031124	047440		
3690	020672	020106	046123	030525		
3691	020700	052040	051503	000122		
3692	020706	042522	042523	020124	EM13:	.ASCIZ /RESET DID NOT CLEAR BIT2 OF SLU1 TCSR/
3693	020714	044504	020104	047516		
3694	020722	020124	046103	040505		
3695	020730	020122	044502	031124		
3696	020736	047440	020106	046123		
3697	020744	030525	052040	051503		
3698	020752	000122				
3699	020754	044502	033124	047440	EM14:	.ASCIZ /BIT6 OF SLU1 TCSR NOT CLEAR AFTER RESET/
3700	020762	020106	046123	030525		
3701	020770	052040	051503	020122		
3702	020776	047516	020124	046103		
3703	021004	040505	020122	043101		
3704	021012	042524	020122	042522		
3705	021020	042523	000124			
3706	021024	046123	030525	054040	EM15:	.ASCIZ /SLU1 XMIT INTERRUPTED WITH PRIORITY 7/
3707	021032	044515	020124	047111		
3708	021040	042524	051122	050125		
3709	021046	042524	020104	044527		
3710	021054	044124	050040	044522		
3711	021062	051117	052111	020131		
3712	021070	000067				
3713	021072	040503	020116	047516	EM16:	.ASCIZ /CAN NOT SET BIT6 OF SLU1 TCSR/
3714	021100	020124	042523	020124		
3715	021106	044502	033124	047440		
3716	021114	020106	046123	030525		
3717	021122	052040	051503	000122		
3718	021130	040503	020116	047516	EM17:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU1 TCSR/
3719	021136	020124	046103	040505		
3720	021144	020122	044502	033124		
3721	021152	047440	020106	046123		
3722	021160	030525	052040	051503		

3723	021166	000122									
3724	021170	042522	042523	020124	EM20:	.ASCIZ	/RESET DID NOT CLEAR BIT6 OF SLU1 TCSR/				
3725	021176	044504	020104	047516							
3726	021204	020124	046103	040505							
3727	021212	020122	044502	033124							
3728	021220	047440	020106	046123							
3729	021226	030525	052040	051503							
3730	021234	000122									
3731	021236	044502	033124	047440	EM21:	.ASCIZ	/BIT6 OF SLU1 RCSR NOT CLEAR AFTER RESET/				
3732	021244	020106	046123	030525							
3733	021252	051040	051503	020122							
3734	021260	047516	020124	046103							
3735	021266	040505	020122	043101							
3736	021274	042524	020122	042522							
3737	021302	042523	000124								
3738	021306	046123	030525	051040	EM22:	.ASCIZ	/SLU1 RCVR INTERRUPT WITH PRIORITY 7/				
3739	021314	053103	020122	047111							
3740	021322	042524	051122	050125							
3741	021330	020124	044527	044124							
3742	021336	050040	044522	051117							
3743	021344	052111	020131	000067							
3744	021352	040503	020116	047516	EM23:	.ASCIZ	/CAN NOT SET BIT6 OF SLU1 RCSR/				
3745	021360	020124	042523	020124							
3746	021366	044502	033124	047440							
3747	021374	020106	046123	030525							
3748	021402	051040	051503	000122							
3749	021410	040503	020116	047516	EM24:	.ASCIZ	/CAN NOT CLEAR BIT6 OF SLU1 RCSR/				
3750	021416	020124	046103	040505							
3751	021424	020122	044502	033124							
3752	021432	047440	020106	046123							
3753	021440	030525	051040	051503							
3754	021446	000122									
3755	021450	040503	020116	047516	EM25:	.ASCIZ	/CAN NOT CLEAR BIT6 OF SLU1 RCSR WITH RESET/				
3756	021456	020124	046103	040505							
3757	021464	020122	044502	033124							
3758	021472	047440	020106	046123							
3759	021500	030525	051040	051503							
3760	021506	020122	044527	044124							
3761	021514	051040	051505	052105							
3762	021522	000									
3763	021523	123	052514	020061	EM26:	.ASCIZ	/SLU1 RECEIVER DONE NEVER SET/				
3764	021530	042522	042503	053111							
3765	021536	051105	042040	047117							
3766	021544	020105	042516	042526							
3767	021552	020122	042523	000124							
3768	021560	042522	042523	020124	EM27:	.ASCIZ	/RESET DID NOT CLEAR SLU1 RCVR DONE/				
3769	021566	044504	020104	047516							
3770	021574	020124	046103	040505							
3771	021602	020122	046123	030525							
3772	021610	051040	053103	020122							
3773	021616	047504	042516	000							
3774	021623	122	040505	044504	EM30:	.ASCIZ	/READING SLU1 RBUF DID NOT CLEAR RCVR DON' /				
3775	021630	043516	051440	052514							
3776	021636	020061	041122	043125							
3777	021644	042040	042111	047040							
3778	021652	052117	041440	042514							

3779	021660	051101	051040	053103		
3780	021666	020122	047504	042516		
3781	021674	000				
3782	021675	123	052514	020062	EM31:	.ASCIZ /SLU2 TCSR DID NOT RETURN SSYNC/
3783	021702	041524	051123	042040		
3784	021710	042111	047040	052117		
3785	021716	051040	052105	051125		
3786	021724	020116	051523	047131		
3787	021732	000103				
3788	021734	046123	031125	052040	EM32:	.ASCIZ /SLU2 TBUF DID NOT RETURN SSYNC/
3789	021742	052502	020106	044504		
3790	021750	020104	047516	020124		
3791	021756	042522	052524	047122		
3792	021764	051440	054523	041516		
3793	021772	000				
3794	021773	123	052514	020062	EM33:	.ASCIZ /SLU2 TCSR DONE NOT CLEARED WITH TBUF FULL
3795	022000	041524	051123	042040		
3796	022006	047117	020105	047516		
3797	022014	020124	046103	040505		
3798	022022	042522	020104	044527		
3799	022030	044124	052040	052502		
3800	022036	020106	052506	046114		
3801	022044	000				
3802	022045	123	052514	020062	EM34:	.ASCIZ /SLU2 TCSR DONE NOT SET AFTER TRANSMIT/
3803	022052	041524	051123	042040		
3804	022060	047117	020105	047516		
3805	022066	020124	042523	020124		
3806	022074	043101	042524	020122		
3807	022102	051124	047101	046523		
3808	022110	052111	000			
3809	022113	123	052514	020062	EM35:	.ASCIZ /SLU2 TCSR DONE NOT SET WITH RESET/
3810	022120	041524	051123	042040		
3811	022126	047117	020105	047516		
3812	022134	020124	042523	020124		
3813	022142	044527	044124	051040		
3814	022150	051505	052105	000		
3815	022155	123	052514	020062	EM36:	.ASCIZ /SLU2 RCSR DID NOT RETURN SSYNC/
3816	022162	041522	051123	042040		
3817	022170	042111	047040	052117		
3818	022176	051040	052105	051125		
3819	022204	020116	051523	047131		
3820	022212	000103				
3821	022214	046123	031125	051040	EM37:	.ASCIZ /SLU2 RBUF DID NOT RETURN SSYNC/
3822	022222	052502	020106	044504		
3823	022230	020104	047516	020124		
3824	022236	042522	052524	047122		
3825	022244	051440	054523	041516		
3826	022252	000				
3827	022253	102	052111	020060	EM40:	.ASCIZ /BITO OF SLU2 TCSR NOT CLEAR AFTER RESET/
3828	022260	043117	051440	052514		
3829	022266	020062	041524	051123		
3830	022274	047040	052117	041440		
3831	022302	042514	051101	040440		
3832	022310	052106	051105	051040		
3833	022316	051505	052105	000		
3834	022323	103	047101	047040	EM41:	.ASCIZ /CAN NOT SET BITO OF SLU2 TCSR/

3835	022330	052117	051440	052105		
3836	022336	041040	052111	020060		
3837	022344	043117	051440	052514		
3838	022352	020062	041524	051123		
3839	022360	000				
3840	022361	103	047101	047040	EM42:	.ASCIZ /CAN NOT CLEAR BIT0 OF SLU2 TCSR/
3841	022366	052117	041440	042514		
3842	022374	051101	041040	052111		
3843	022402	020060	043117	051440		
3844	022410	052514	020062	041524		
3845	022416	051123	000			
3846	022421	122	051505	052105	EM43:	.ASCIZ /RESET DID NOT CLEAR BIT0 OF SLU2 TCSR/
3847	022426	042040	042111	047040		
3848	022434	052117	041440	042514		
3849	022442	051101	041040	052111		
3850	022450	020060	043117	051440		
3851	022456	052514	020062	041524		
3852	022464	051123	000			
3853	022467	102	052111	020066	EM44:	.ASCIZ /BIT6 OF SLU2 TCSR NOT CLEAR AFTER RESET/
3854	022474	043117	051440	052514		
3855	022502	020062	041524	051123		
3856	022510	047040	052117	041440		
3857	022516	042514	051101	040440		
3858	022524	052106	051105	051040		
3859	022532	051505	052105	000		
3860	022537	123	052514	020062	EM45:	.ASCIZ /SLU2 XMIT INTERRUPTED WITH PRIORITY 7/
3861	022544	046530	052111	044440		
3862	022552	052116	051105	052522		
3863	022560	052120	042105	053440		
3864	022566	052111	020110	051120		
3865	022574	047511	044522	054524		
3866	022602	033440	000			
3867	022605	103	047101	047040	EM46:	.ASCIZ /CAN NOT SET BIT6 OF SLU2 TCSR/
3868	022612	052117	051440	052105		
3869	022620	041040	052111	020066		
3870	022626	043117	051440	052514		
3871	022634	020062	041524	051123		
3872	022642	000				
3873	022643	103	047101	047040	EM47:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 TCSR/
3874	022650	052117	041440	042514		
3875	022656	051101	041040	052111		
3876	022664	020066	043117	051440		
3877	022672	052514	020062	041524		
3878	022700	051123	000			
3879	022703	122	051505	052105	EM50:	.ASCIZ /RESET DID NOT CLEAR BIT6 OF SLU2 TCSR/
3880	022710	042040	042111	047040		
3881	022716	052117	041440	042514		
3882	022724	051101	041040	052111		
3883	022732	020066	043117	051440		
3884	022740	052514	020062	041524		
3885	022746	051123	000			
3886	022751	102	052111	020066	EM51:	.ASCIZ /BIT6 OF SLU2 RCSR NOT CLEAR AFTER RESET/
3887	022756	043117	051440	052514		
3888	022764	020062	041522	051123		
3889	022772	047040	052117	041440		
3890	023000	042514	051101	040440		

3891	023006	052106	051105	051040		
3892	023014	051505	052105	000		
3893	023021	123	052514	020062	EM52:	.ASCIZ /SLU2 RCVR INTERRUPT WITH PRIORITY /
3894	023026	041522	051126	044440		
3895	023034	052116	051105	052522		
3896	023042	052120	053440	052111		
3897	023050	020110	051120	047511		
3898	023056	044522	054524	033440		
3899	023064	000				
3900	023065	103	047101	047040	EM53:	.ASCIZ /CAN NOT SET BIT6 OF SLU2 RCSR/
3901	023072	052117	051440	052105		
3902	023100	041040	052111	020066		
3903	023106	043117	051440	052514		
3904	023114	020062	041522	051123		
3905	023122	000				
3906	023123	103	047101	047040	EM54:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR/
3907	023130	052117	041440	042514		
3908	023136	051101	041040	052111		
3909	023144	020066	043117	051440		
3910	023152	052514	020062	041522		
3911	023160	051123	000			
3912	023163	103	047101	047040	EM55:	.ASCIZ /CAN NOT CLEAR BIT6 OF SLU2 RCSR WITH RESET/
3913	023170	052117	041440	042514		
3914	023176	051101	041040	052111		
3915	023204	020066	043117	051440		
3916	023212	052514	020062	041522		
3917	023220	051123	053440	052111		
3918	023226	020110	042522	042523		
3919	023234	000124				
3920	023236	046123	031125	051040	EM56:	.ASCIZ /SLU2 RECEIVER DONE NEVER SET/
3921	023244	041505	044505	042526		
3922	023252	020122	047504	042516		
3923	023260	047040	053105	051105		
3924	023266	051440	052105	000		
3925	023273	122	051505	052105	EM57:	.ASCIZ /RESET DID NOT CLEAR SLU2 RCVR DONE/
3926	023300	042040	042111	047040		
3927	023306	052117	041440	042514		
3928	023314	051101	051440	052514		
3929	023322	020062	041522	051126		
3930	023330	042040	047117	000105		
3931	023336	042522	042101	047111	EM60:	.ASCIZ /READING SLU2 RBUF DID NOT CLEAR RCVR DONE/
3932	023344	020107	046123	031125		
3933	023352	051040	052502	020106		
3934	023360	044504	020104	047516		
3935	023366	020124	046103	040505		
3936	023374	020122	041522	051126		
3937	023402	042040	047117	000105		
3938	023410	045514	020123	044504	EM61:	.ASCIZ /LKS DID NOT RETURN SSYNC/
3939	023416	020104	047516	020124		
3940	023424	042522	052524	047122		
3941	023432	051440	054523	041516		
3942	023440	000				
3943	023441	102	052111	020066	EM62:	.ASCIZ /BIT6 OF LKS NOT CLEAR AFTER RESET/
3944	023446	043117	046040	051513		
3945	023454	047040	052117	041440		
3946	023462	042514	051101	040440		

EX

3947	023470	052106	051105	051040	
3948	023476	051505	052105	000	
3949	023503	114	051513	044440	EM63: .ASCIZ /LKS INTERRUPT WITH PRIORITY 7/
3950	023510	052116	051105	052522	
3951	023516	052170	053440	052111	
3952	023524	020110	051120	047511	
3953	023532	044522	054524	033440	
3954	023540	000			
3955	023541	103	047101	047040	EM64: .ASCIZ /CAN NOT SET BIT6 OF LKS/
3956	023546	052117	051440	052105	
3957	023554	041040	052111	020066	
3958	023562	043117	046040	051513	
3959	023570	000			
3960	023571	103	047101	047040	EM65: .ASCIZ /CAN NOT CLEAR BIT6 OF LKS/
3961	023576	052117	041440	042514	
3962	023604	051101	041040	052111	
3963	023612	020066	043117	046040	
3964	023620	051513	000		
3965	023623	122	051505	052105	EM66: .ASCIZ /RESET DID NOT CLEAR BIT6 OF LKS/
3966	023630	042040	042111	047040	
3967	023636	052117	041440	042514	
3968	023644	051101	041040	052111	
3969	023652	020066	043117	046040	
3970	023660	051513	000		
3971	023663	102	052111	020067	EM67: .ASCIZ /BIT7 OF LKS NOT SET AFTER RESET/
3972	023670	043117	046040	051513	
3973	023676	047040	052117	051440	
3974	023704	052105	040440	052106	
3975	023712	051105	051040	051505	
3976	023720	052105	000		
3977	023723	103	047101	047040	EM70: .ASCIZ /CAN NOT CLEAR BIT7 OF LKS/
3978	023730	052117	041440	042514	
3979	023736	051101	041040	052111	
3980	023744	020067	043117	046040	
3981	023752	051513	000		
3982	023755	102	052111	020067	EM71: .ASCIZ /BIT7 OF LKS DOES NOT SET/
3983	023762	043117	046040	051513	
3984	023770	042040	042517	020123	
3985	023776	047516	020124	042523	
3986	024004	000124			
3987	024006	051127	052111	047111	EM72: .ASCIZ /WRITING TO ONE INTERNAL ADDRESS MODIFIED ANOTHER/
3988	024014	020107	047524	047440	
3989	024022	042516	044440	052116	
3990	024030	051105	040516	020114	
3991	024036	042101	051104	051505	
3992	024044	020123	047515	044504	
3993	024052	044506	042105	040440	
3994	024060	047516	044124	051105	
3995	024066	000			
3996	024067	123	052514	020061	EM74: .ASCIZ /SLU1 XMIT INTERRUPTS WHEN DISABLED/
3997	024074	046530	052111	044440	
3998	024102	052116	051105	052522	
3999	024110	052120	020123	044127	
4000	024116	047105	042040	051511	
4001	024124	041101	042514	000104	
4002	024132	046123	030525	054040	EM75: .ASCIZ /SLU1 XMIT DID NOT INTERRUPT/

4003	024140	044515	020124	044504	
4004	024146	020104	047516	020124	
4005	024154	047111	042524	051122	
4006	024162	050125	000124		
4007	024166	046123	030525	054040	EM76: .ASCIZ /SLUI XMIT INTERRUPT AT PRIORITY 7,
4008	024174	044515	020124	047111	
4009	024202	042524	051122	050125	
4010	024210	020124	052101	050040	
4011	024216	044522	051117	052111	
4012	024224	020131	000067		
4013	024230	046123	030525	054040	EM77: .ASCIZ /SLUI XMIT INTERRUPTS WITH ENABLE CLEAR/
4014	024236	044515	020124	047111	
4015	024244	042524	051122	050125	
4016	024252	051524	053440	052111	
4017	024260	020110	047105	041101	
4018	024266	042514	041440	042514	
4019	024274	051101	000		
4020	024277	123	052514	020061	EM100: .ASCIZ /SLUI XMIT DID NOT INTERRUPT/
4021	024304	046530	052111	042040	
4022	024312	042111	047040	052117	
4023	024320	044440	052116	051105	
4024	024326	052522	052120	000	
4025	024333	123	052514	020061	EM101: .ASCIZ /SLUI XMIT RE-INTERRUPTED/
4026	024340	046530	052111	051040	
4027	024346	026505	047111	042524	
4028	024354	051122	050125	042524	
4029	024362	000104			
4030	024364	047514	042101	047111	EM102: .ASCIZ /LOADING SLUI TBUF DID NOT CLEAR INTERRUPT/
4031	024372	020107	046123	030525	
4032	024400	052040	052502	020106	
4033	024406	044504	020104	047516	
4034	024414	020124	046103	040505	
4035	024422	020122	047111	042524	
4036	024430	051122	050125	000124	
4037	024436	046123	030525	051040	EM103: .ASCIZ /SLUI RCVR INTERRUPTS WITH ENABLE CLEAR/
4038	024444	053103	020122	047111	
4039	024452	042524	051122	050125	
4040	024460	051524	053440	052111	
4041	024466	020110	047105	041101	
4042	024474	042514	041440	042514	
4043	024502	051101	000		
4044	024505	123	052514	020061	EM104: .ASCIZ /SLUI RCVR DID NOT INTERRUPT/
4045	024512	041522	051126	042040	
4046	024520	042111	047040	052117	
4047	024526	044440	052116	051105	
4048	024534	052522	052120	000	
4049	024541	123	052514	020061	EM105: .ASCIZ /SLUI RCVR INTERRUPTS AT PRIORITY 7,
4050	024546	041522	051126	044440	
4051	024554	052116	051105	052522	
4052	024562	052120	020123	052101	
4053	024570	050040	044522	051117	
4054	024576	052111	020131	000067	
4055	024604	046123	030525	051040	EM106: .ASCIZ /SLUI RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4056	024612	053103	020122	047111	
4057	024620	042524	051122	050125	
4058	024626	051524	053440	052111	



4059	024634	020110	047111	042524	
4060	024642	051122	050125	020124	
4061	024650	047105	041101	042514	
4062	024656	041440	042514	051101	
4063	024664	000			
4064	024665	123	052514	020061	EM107: .ASCIZ /SLU1 RCVR DID NOT INTERRUPT/
4065	024672	041522	051126	042040	
4066	024700	042111	047040	052117	
4067	024706	044440	052116	051105	
4068	024714	052522	052120	000	
4069	024721	123	052514	020061	EM110: .ASCIZ /SLU1 RECEIVER RE-INTERRUPTED/
4070	024726	042522	042503	053111	
4071	024734	051105	051040	026505	
4072	024742	047111	042524	051122	
4073	024750	050125	042524	000104	
4074	024756	046123	030525	051040	EM111: .ASCIZ /SLU1 READING RBUF DID NOT CLEAR INTERRUPT/
4075	024764	040505	044504	043516	
4076	024772	051040	052502	020106	
4077	025000	044504	020104	047516	
4078	025006	020124	046103	040505	
4079	025014	020122	047111	042524	
4080	025022	051122	050125	000124	
4081	025030	042522	042523	020124	EM112: .ASCIZ /RESET DID NOT CLEAR SLU1 RCVR INTERRUPT/
4082	025036	044504	020104	047516	
4083	025044	020124	046103	040505	
4084	025052	020122	046123	030525	
4085	025060	051040	053103	020122	
4086	025066	047111	042524	051122	
4087	025074	050125	000124		
4088	025100	046123	030525	023440	EM113: .ASCIZ /SLU1 'OR' FLAG DID NOT SET/
4089	025106	051117	020047	046106	
4090	025114	043501	042040	042111	
4091	025122	047040	052117	051440	
4092	025130	052105	000		
4093	025133	123	052514	020061	EM114: .ASCIZ /SLU1 'ERROR' NOT SET WITH 'OR' FLAG/
4094	025140	042447	051122	051117	
4095	025146	020047	047516	020124	
4096	025154	042523	020124	044527	
4097	025162	044124	023440	051117	
4098	025170	020047	046106	043501	
4099	025176	000			
4100	025177	104	052101	020101	EM115: .ASCIZ /DATA COMPARE ERROR/
4101	025204	047503	050115	051101	
4102	025212	020105	051105	047522	
4103	025220	000122			
4104	025222	046123	031125	054040	EM116: .ASCIZ /SLU2 XMIT INTERRUPTS WHEN DISABLED/
4105	025230	044515	020124	047111	
4106	025236	042524	051122	050125	
4107	025244	051524	053440	042510	
4108	025252	020116	044504	040523	
4109	025260	046102	042105	000	
4110	025265	123	052514	020062	EM117: .ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
4111	025272	046530	052111	042040	
4112	025300	042111	047040	052117	
4113	025306	044440	052116	051105	
4114	025314	052522	052120	000	

4115	025321	123	052514	020062	EM120:	.ASCIZ /SLU2 XMIT INTERRUPT AT PRIORITY
4116	025326	046530	052111	044440		
4117	025334	052116	051105	052522		
4118	025342	052120	040440	020124		
4119	025350	051120	047511	044522		
4120	025356	054524	033440	000		
4121	025363	123	052514	020062	EM121:	.ASCIZ /SLU2 XMIT INTERRUPTS WITH ENABLE CLEAR
4122	025370	046530	052111	044440		
4123	025376	052116	051105	052522		
4124	025404	052120	020123	044527		
4125	025412	044124	042440	040516		
4126	025420	046102	020105	046103		
4127	025426	040505	000122			
4128	025432	046123	031125	054040	EM122:	.ASCIZ /SLU2 XMIT DID NOT INTERRUPT/
4129	025440	044515	020124	044504		
4130	025446	020104	047516	020124		
4131	025454	047111	042524	051122		
4132	025462	050125	000124			
4133	025466	046123	031125	054040	EM123:	.ASCIZ /SLU2 XMIT RE-INTERRUPTED/
4134	025474	044515	020124	042522		
4135	025502	044455	052116	051105		
4136	025510	052522	052120	042105		
4137	025516	000				
4138	025517	114	040517	044504	EM124:	.ASCIZ /LOADING SLU2 TBUF DID NOT CLEAR INTERRUPT/
4139	025524	043516	051440	052514		
4140	025532	020062	041124	043125		
4141	025540	042040	042111	047040		
4142	025546	052117	041440	042514		
4143	025554	051101	044440	052116		
4144	025562	051105	052522	052120		
4145	025570	000				
4146	025571	123	052514	020062	EM125:	.ASCIZ /SLU2 RCVR INTERRUPTS WITH ENABLE CLEAR/
4147	025576	041522	051126	044440		
4148	025604	052116	051105	052522		
4149	025612	052120	020123	044527		
4150	025620	044124	042440	040516		
4151	025626	046102	020105	046103		
4152	025634	040505	000122			
4153	025640	046123	031125	051040	EM126:	.ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
4154	025646	053103	020122	044504		
4155	025654	020104	047516	020124		
4156	025662	047111	042524	051122		
4157	025670	050125	000124			
4158	025674	046123	031125	051040	EM127:	.ASCIZ /SLU2 RCVR INTERRUPTS AT PRIORITY 7/
4159	025702	053103	020122	047111		
4160	025710	042524	051122	050125		
4161	025716	051524	040440	020124		
4162	025724	051120	047511	044522		
4163	025732	054524	033440	000		
4164	025737	123	052514	020062	EM130:	.ASCIZ /SLU2 RCVR INTERRUPTS WITH INTERRUPT ENABLE CLEAR/
4165	025744	041522	051126	044440		
4166	025752	052116	051105	052522		
4167	025760	052120	020123	044527		
4168	025766	044124	044440	052116		
4169	025774	051105	052522	052120		
4170	026002	042440	040516	046102		

4171	026010	020105	046103	040505		
4172	026016	000122				
4173	026020	046123	031125	051040	EM131:	.ASCIZ /SLU2 RCVR DID NOT INTERRUPT/
4174	026026	053103	020122	044504		
4175	026034	020104	047516	020124		
4176	026042	047111	042524	051122		
4177	026050	050125	000124			
4178	026054	046123	031125	051040	EM132:	.ASCIZ /SLU2 RECEIVER RE-INTERRUPTED/
4179	026062	041505	044505	042526		
4180	026070	020122	042522	044455		
4181	026076	052116	051105	052522		
4182	026104	052120	042105	000		
4183	026111	123	052514	020062	EM133:	.ASCIZ /SLU2 READING RBUF DID NOT CLEAR INTERRUPT/
4184	026116	042522	042101	047111		
4185	026124	020107	041122	043125		
4186	026132	042040	042111	047040		
4187	026140	052117	041440	042514		
4188	026146	051101	044440	052116		
4189	026154	051105	052522	052120		
4190	026162	000				
4191	026163	122	051505	052105	EM134:	.ASCIZ /RESET DID NOT CLEAR SLU2 RCVR INTERRUPT/
4192	026170	042040	042111	047040		
4193	026176	052117	041440	042514		
4194	026204	051101	051440	052514		
4195	026212	020062	041522	051126		
4196	026220	044440	052116	051105		
4197	026226	052522	052120	000		
4198	026233	123	052514	020062	EM135:	.ASCIZ /SLU2 'OR' FLAG DID NOT SET/
4199	026240	047447	023522	043040		
4200	026246	040514	020107	044504		
4201	026254	020104	047516	020124		
4202	026262	042523	000124			
4203	026266	046123	031125	023440	EM136:	.ASCIZ /SLU2 'ERROR' NOT SET WITH 'OR' FLAG/
4204	026274	051105	047522	023522		
4205	026302	047040	052117	051440		
4206	026310	052105	053440	052111		
4207	026316	020110	047447	023522		
4208	026324	043040	040514	000107		
4209	026332	046123	031125	041040	EM137:	.ASCIZ /SLU2 BREAK DID NOT TRANSMIT ALL ZEROES/
4210	026340	042522	045501	042040		
4211	026346	042111	047040	052117		
4212	026354	052040	040522	051516		
4213	026362	044515	020124	046101		
4214	026370	020114	042532	047522		
4215	026376	051505	000			
4216	026401	102	042522	045501	EM140:	.ASCIZ /BREAK DID NOT SET FRAMING ERROR/
4217	026406	042040	042111	047040		
4218	026414	052117	051440	052105		
4219	026422	043040	040522	044515		
4220	026430	043516	042440	051122		
4221	026436	051117	000			
4222	026441	123	052514	020062	EM141:	.ASCIZ /SLU2 'ERROR' NOT SET WITH 'FR' FLAG/
4223	026446	042447	051122	051117		
4224	026454	020047	047516	020124		
4225	026462	042523	020124	044527		
4226	026470	044124	023440	051106		

4227	026476	020047	046106	043501			
4228	026504	000					
4229	026505	104	052101	020101	EM142:	.ASCIZ	/DATA COMPARE ERROR WITH CABLE/
4230	026512	047503	050115	051101			
4231	026520	020105	051105	047522			
4232	026526	020122	044527	044124			
4233	026534	041440	041101	042514			
4234	026542	000					
4235	026543	122	041524	044440	EM143:	.ASCIZ	/RTC INTERRUPT AT PRIORITY 7/
4236	026550	052116	051105	052522			
4237	026556	052120	040440	020124			
4238	026564	051120	047511	044522			
4239	026572	054524	033440	000			
4240	026577	122	041524	044440	EM144:	.ASCIZ	/RTC INTERRUPTS WHEN DISABLED/
4241	026604	052116	051105	052522			
4242	026612	052120	020123	044127			
4243	026620	047105	042040	051511			
4244	026626	041101	042514	000104			
4245	026634	052122	020103	047111	EM145:	.ASCIZ	/RTC INTERRUPT DID NOT OCCUR/
4246	026642	042524	051122	050125			
4247	026650	020124	044504	020104			
4248	026656	047516	020124	041517			
4249	026664	052503	000122				
4250	026670	052122	020103	047111	EM146:	.ASCIZ	/RTC INTERRUPT DID NOT OCCUR/
4251	026676	042524	051122	050125			
4252	026704	020124	044504	020104			
4253	026712	047516	020124	041517			
4254	026720	052503	000122				
4255	026724	052122	020103	047504	EM147:	.ASCIZ	/RTC DOUBLE INTERRUPT/
4256	026732	041125	042514	044440			
4257	026740	052116	051105	052522			
4258	026746	052120	000				
4259	026751	122	051505	052105	EM150:	.ASCIZ	/RESET DID NOT CLEAR RTC INTERRUPT/
4260	026756	042040	042111	047040			
4261	026764	052117	041440	042514			
4262	026772	051101	051040	041524			
4263	027000	044440	052116	051105			
4264	027006	052522	052120	000			
4265	027013	122	041524	044440	EM151:	.ASCIZ	/RTC INTERRUPT DID NOT CLEAR WITH BIT7 OF LKS/
4266	027020	052116	051105	052522			
4267	027026	052120	042040	042111			
4268	027034	047040	052117	041440			
4269	027042	042514	051101	053440			
4270	027050	052111	020110	044502			
4271	027056	033524	047440	020106			
4272	027064	045514	000123				
4273	027070	046103	041517	020113	EM152:	.ASCIZ	/CLOCK REPEATABILITY ERROR/
4274	027076	042522	042520	052101			
4275	027104	041101	046111	052111			
4276	027112	020131	051105	047522			
4277	027120	000122					
4278	027122	046123	030525	051040	EM153:	.ASCIZ	/SLU1 RECEIVER STATUS ERROR/
4279	027130	041505	044505	042526			
4280	027136	020122	052123	052101			
4281	027144	051525	042440	051122			
4282	027152	051117	000				

4283	027155	123	052514	020062	EM154:	.ASCIZ	/SLU2 RECEIVER STATUS ERROR/
4284	027162	042522	042503	053111			
4285	027170	051105	051440	040524			
4286	027176	052524	020123	051105			
4287	027204	047522	000122				
4288	027210	047111	047503	051122	EM155:	.ASCIZ	/INCORRECT RECEIVE COUNT SLU1/
4289	027216	041505	020124	042522			
4290	027224	042503	053111	020105			
4291	027232	047503	047125	020124			
4292	027240	046123	030525	000			
4293	027245	123	052514	020061	EM156:	.ASCIZ	/SLU1 DATA COMPARE ERROR IN EXERCISER/
4294	027252	040504	040524	041440			
4295	027260	046517	040520	042522			
4296	027266	042440	051122	051117			
4297	027274	044440	020116	054105			
4298	027302	051105	044503	042523			
4299	027310	000122					
4300	027312	047111	047503	051122	EM157:	.ASCIZ	/INCORRECT RECEIVE COUNT SLU2/
4301	027320	041505	020124	042522			
4302	027326	042503	053111	020105			
4303	027334	047503	047125	020124			
4304	027342	046123	031125	000			
4305	027347	123	052514	020062	EM160:	.ASCIZ	/SLU2 DATA COMPARE ERROR IN EXERCISER/
4306	027354	040504	040524	041440			
4307	027362	046517	040520	042522			
4308	027370	042440	051122	051117			
4309	027376	044440	020116	054105			
4310	027404	051105	044503	042523			
4311	027412	000122					
4312	027414	051124	050101	041440	EM161:	.ASCIZ	/TRAP CATCHER/
4313	027422	052101	044103	051105			
4314	027430	G00					
4315							
4316	027431	124	051505	021524	DH5:	.ASCIZ	/TEST# ERR PC TCSR/
4317	027436	020040	042440	051122			
4318	027444	050040	020103	052040			
4319	027452	051503	000122				
4320	027456	042524	052123	020043	DH6:	.ASCIZ	/TEST# ERR PC RCSR/
4321	027464	020040	051105	030122			
4322	027472	041520	020040	041522			
4323	027500	051123	000				
4324	027503	124	051505	021524	DH7:	.ASCIZ	/TEST# ERR PC RBUF/
4325	027510	020040	042440	051122			
4326	027516	050040	020103	051040			
4327	027524	052502	000106				
4328	027530	042524	052123	020043	DH32:	.ASCIZ	/TEST# ERR PC TBUF/
4329	027536	020040	051105	020122			
4330	027544	041520	020040	041124			
4331	027552	043125	000				
4332	027555	124	051505	021524	DH61:	.ASCIZ	/TEST# ERR PC LKS/
4333	027562	020040	042440	051122			
4334	027570	050040	020103	046040			
4335	027576	051513	000				
4336	027601	124	051505	021524	DH72:	.ASCIZ	/TEST# ERR PC GADR BOADR GDDAT BDDAT/
4337	027606	020040	042440	051122			
4338	027614	050040	020103	043440			



```
4395 030252 001104 001016 002776 DT32: .WORD $TESTN,$ERRPC,TBUF,0
4396 030260 000000
4397 030262 001104 001016 002770 DT36: .WORD $TESTN,$ERRPC,RCSR,0
4398 030270 000000
4399 030272 001104 001016 002772 DT37: .WORD $TESTN,$ERRPC,RBUF,0
4400 030300 000000
4401 030302 001104 001016 003010 DT61: .WORD $TESTN,$ERRPC,LKS,0
4402 030310 000000
4403 030312 001104 001016 001020 DT72: .WORD $TESTN,$ERRPC,$GDADR,$BDADR,$GDDAT,$BDDAT,0
4404 030320 001022 001024 001026
4405 030326 000000
4406 030330 001104 001016 003000 DT115: .WORD $TESTN,$ERRPC,CRCR,$GDDAT,$BDDAT,0
4407 030336 001024 001026 000000
4408 030344 001104 001016 002770 DT137: .WORD $TESTN,$ERRPC,RCSR,$BDDAT,0
4409 030352 001026 000000
4410 030356 001104 001016 001024 DT142: .WORD $TESTN,$ERRPC,$GDDAT,$BDDAT,0
4411 030364 001026 000000
4412 030370 001104 001016 003010 DT152: .WORD $TESTN,$ERRPC,LKS,FIRST,SECND,0
4413 030376 013574 013576 000000
4414 030404 001104 001016 003000 DT155: .WORD $TESTN,$ERRPC,CRCR,XMTCT1,RECCT1,0
4415 030412 014726 014732 000000
4416 030420 001104 001016 002770 DT157: .WORD $TESTN,$ERRPC,RCSR,XMTCT2,RECCT2,0
4417 030426 014730 014734 000000
4418 030434 001104 001016 002770 DT161: .WORD $TESTN,$ERRPC,RCSR,OLDPC,BDVECT,0
4419 030442 015114 015116 000000
4420 030450 000200 BUF1: .BLKW 200 ;SLU1 INPUT BUFFER FOR BLAST TEST
4421 031050 000200 BUF2: .BLKW 200 ;SLU2 INPIT BUFFER FOR BLAST TEST
4422 000001 .END
```

ABASE = 176500	BIT0 = 000001	DM142 = 027760	EM125 = 025571	EM43 = 022421
ACDW1 = 000000	BIT00 = 000001	DM152 = 030014	EM126 = 025640	EM44 = 022467
ACDW2 = 000000	BIT01 = 000002	DM155 = 030061	EM127 = 025674	EM45 = 022537
ACPUOP = 000000	BIT02 = 000004	DM161 = 030125	EM13 = 020706	EM46 = 022605
ADDW0 = 000000	BIT03 = 000010	DM32 = 027530	EM130 = 025737	EM47 = 022643
ADDW1 = 000000	BIT04 = 000020	DM5 = 027431	EM131 = 026020	EM5 = 020510
ADDW10 = 000000	BIT05 = 000040	DM6 = 027456	EM132 = 026054	EM50 = 022703
ADDW11 = 000000	BIT06 = 000100	DM61 = 027555	EM133 = 026111	EM51 = 022751
ADDW12 = 000000	BIT07 = 000200	DM7 = 027503	EM134 = 026163	EM52 = 023021
ADDW13 = 000000	BIT08 = 000400	DM72 = 027601	EM135 = 026233	EM53 = 023065
ADDW14 = 000000	BIT09 = 001000	DISPLA = 001042	EM136 = 026266	EM54 = 023123
ADDW15 = 000000	BIT1 = 000002	DISPRE = 000174	EM137 = 026332	EM55 = 023163
ADDW2 = 000000	BIT10 = 002000	DSWR = 177570	EM14 = 020754	EM56 = 023236
ADDW3 = 000000	BIT11 = 004000	DT115 = 030330	EM140 = 026401	EM57 = 023273
ADDW4 = 000000	BIT12 = 010000	DT137 = 030344	EM141 = 026441	EM6 = 020552
ADDW5 = 000000	BIT13 = 020000	DT142 = 030356	EM142 = 026505	EM60 = 023336
ADDW6 = 000000	BIT14 = 040000	DT152 = 030370	EM143 = 026543	EM61 = 023410
ADDW7 = 000000	BIT15 = 100000	DT155 = 030404	EM144 = 026577	EM62 = 023441
ADDW8 = 000000	BIT2 = 000004	DT157 = 030420	EM145 = 026634	EM63 = 023503
ADDW9 = 000000	BIT3 = 000010	DT161 = 030434	EM146 = 026670	EM64 = 023541
ADEVCT = 000000	BIT4 = 000020	DT31 = 030242	EM147 = 026724	EM65 = 023571
ADEVH = 000000	BIT5 = 000040	DT32 = 030252	EM15 = 021024	EM66 = 023623
AENV = 000000	BIT6 = 000100	DT36 = 030262	EM150 = 026751	EM67 = 023663
AENVH = 000000	BIT7 = 000200	DT37 = 030272	EM151 = 027013	EM7 = 020611
AFATAL = 000000	BIT8 = 000400	DT5 = 030212	EM152 = 027070	EM70 = 023723
AMADR1 = 000000	BIT9 = 001000	DT6 = 030222	EM153 = 027122	EM71 = 023755
AMADR2 = 000000	BLAST = 013600	DT61 = 030302	EM154 = 027155	EM72 = 024006
AMADR3 = 000000	BPT = 000003	DT7 = 030232	EM155 = 027210	EM74 = 024067
AMADR4 = 000000	BPTVEC = 000014	DT72 = 030312	EM156 = 027245	EM75 = 024132
AMAMS1 = 000000	BUF1 = 030450	ECHO = 020106	EM157 = 027312	EM76 = 024166
AMAMS2 = 000000	BUF2 = 031050	EMTVEC = 000030	EM16 = 021072	EM77 = 024230
AMAMS3 = 000000	CATCH = 015070	EM100 = 024277	EM160 = 027347	ERRVEC = 000004
AMAMS4 = 000000	CHECK1 = 014524	EM101 = 024333	EM161 = 027414	FINIE = 014664
AMSGAD = 000000	CHECK2 = 014604	EM102 = 024364	EM17 = 021130	FIRST = 013574
AMSGLG = 000000	CKSWR = 104407	EM103 = 024436	EM20 = 021170	GTSWR = 104406
AMSGTY = 000000	CMPARE = 013532	EM104 = 024505	EM21 = 021236	HT = 000011
AMTYP1 = 000000	CONT = 006510	EM105 = 024541	EM22 = 021306	ID = 003700
AMTYP2 = 000000	CONT41 = 012330	EM106 = 024604	EM23 = 021352	IOHAND = 014400
AMTYP3 = 000000	COUNT = 014736	EM107 = 024665	EM24 = 021410	IOTVEC = 000020
AMTYP4 = 000000	CR = 000015	EM11 = 020650	EM25 = 021450	LF = 000012
APASS = 000000	CRBUF = 003002	EM110 = 024721	EM26 = 021523	LKS = 003010
APRIOR = 000000	CRCSR = 003000	EM111 = 024756	EM27 = 021560	LTCIT = 012544
APTCSU = 000040	CRLF = 000200	EM112 = 025030	EM30 = 021623	LTCRT = 006152
APTENV = 000001	CRPSW = 003024	EM113 = 025100	EM31 = 021675	MFR = 020327
APTSIZ = 000200	CRVECT = 003022	EM114 = 025133	EM32 = 021734	MOR = 020341
APTSP0 = 000100	CTBUF = 003006	EM115 = 025177	EM33 = 021773	MPAR = 020316
ASWREG = 000000	CTCSR = 003004	EM116 = 025222	EM34 = 022045	MSG = 020270
ATESTN = 000000	CTPSW = 003030	EM117 = 025265	EM35 = 022113	MSTOP = 020354
AUNIT = 000000	CTVECT = 003026	EM120 = 025321	EM36 = 022155	M1 = 030176
AUSWR = 000400	CWDONE = 004602	EM121 = 025363	EM37 = 022214	M2 = 030210
AVECT1 = 000300	DDISP = 177570	EM122 = 025432	EM40 = 022253	OLDPC = 015114
AVECT2 = 000000	DM115 = 027657	EM123 = 025466	EM41 = 022323	OUTTST = 020364
BDVECT = 015116	DM137 = 027723	EM124 = 025517	EM42 = 022361	PIRQ = 177772



PIRQVE - 000240	SW15 - 100000	TST46 G10600	%CNTLU 017772	%MBADR 000502
PRO - 000000	SW2 - 000004	TST47 010722	%CPUOP 001126	%MFLG 016230
PR1 - 000040	SW3 - 000010	TST5 004012	%CRLF 001075	%MNEW 020015
PR2 - 000100	SW4 - 000020	TST50 011042	%DBLK 017232	%MSGAD 001114
PH3 - 000140	SW5 - 000040	TST51 011176	%DEVCT 001110	%MSGLG 001116
PR4 - 000200	SW6 - 000100	TST52 011340	%DEVN 001156	%MSGTY 001100
PR5 - 000240	SW7 - 000200	TST53 011460	%DOAGN 015032	%MSMR 020004
PR6 - 000300	SW8 - 000400	TST54 011576	%DTBL 017222	%MTYP1 001131
PR7 - 000340	SW9 - 001000	TST55 011730	%ENDAD 015022	%MTYP2 001135
PS - 177776	TBITVE - 000014	TST56 012034	%ENDCT 014770	%MTYP3 001141
PSM - 177776	TBUF 002776	TST57 012160	%ENDMG 015041	%MTYP4 001145
PLRVEC - 000024	TCSR 002774	TST6 004052	%ENULL 015036	%NULL 001054
RBUF 002772	TICKER 014144	TST60 012246	%ENV 001120	%NMTST - 000001
RCSR 002770	TKVEC - 000060	TST61 012354	%ENVN 001121	%OCNT 017012
RDCMR - 104410	TOLER 013544	TST62 012446	%ECP 014740	%OMODE 017014
RDLIN - 104411	TPSM 003020	TST63 012544	%EOPCT 014762	%OVER 015732
RECCT1 014732	TPVEC - 000064	TST64 012766	%ERFLG 001003	%PASS 001106
RECCT2 014734	TRAPVE - 000034	TST65 013140	%ERMAX 001015	%PASTM 000506
REC1 014210	TRTVEC - 000014	TST66 013262	%ERROR 015120	%POWER 015604
REC2 014324	TST1 003450	TST67 013416	%ERRPC 001016	%PRDN 015436
RESVEC - 000010	TST10 004246	TST7 004112	%ERRTB 001160	%PRMG 015572
RPSM 003014	TST11 004412	TST70 013600	%ERTY 015302	%PRUP 015510
RTCPSM 003034	TST12 004556	TVECT 003016	%ERTTL 001012	%QUES 001074
RTCVT 003032	TST13 004652	TYPDS - 104405	%ESCAP 001072	%RDCMR 017524
RVECT 003012	TST14 004726	TYPE - 104401	%ETABL 001120	%RDLIN 017654
R6 - 0000006	TST15 005002	TYPC - 104402	%ETEND 001160	%RDSZ - 000010
R7 - 0000007	TST16 005042	TYPON - 104404	%FATAL 001102	%RTNAD 015034
SECND 013576	TST17 005134	TYPOS - 104403	%FFLG 016232	%SAVR6 015602
SLUIT 006700	TST2 003516	UNIQUE 006524	%FILLC 001056	%SCOPE 015614
SLUZIT 010600	TST20 005214	WAIT 020276	%FILLS 001055	%SETUP - 000137
SLUZRT 004726	TST21 005254	WAITER 014500	%GDADR 001020	%STUP - 177777
STACK - 001100	TST22 005314	WAITIO 014140	%GDDAT 001024	%SVLAD 015716
START 003036	TST23 005452	WDONE 006040	%GET42 015012	%SVPC - 000500
STKLMT - 177774	TST24 005634	WRPSW 015056	%GTSWR 017312	%SMR - 161000
SWR 001040	TST25 006016	XMIT1 014154	%HD - 000003	%SMREG 001122
SWREG 000176	TST26 006110	XMIT2 014270	%HIBTS 000500	%SMRK - 000000
SW0 - 000001	TST27 006152	XMTCT1 014726	%ICNT 001004	%TESTN 001104
SW00 - 000001	TST3 003564	XMTCT2 014730	%ILLUP 015576	%TKB 001046
SW01 - 000002	TST30 006226	%APTHD 000500	%INTAG 001035	%TKS 001044
SW02 - 000004	TST31 006410	%ATYC 016012	%ITEMB 001014	%TMP0 001060
SW03 - 000010	TST32 006524	%ATY1 015766	%LF 001076	%TMP1 001062
SW04 - 000020	TST33 006700	%ATY3 015774	%LFLG 016231	%TMP2 001064
SW05 - 000040	TST34 007046	%ATY4 016004	%LPADR 001006	%TMP3 001066
SW06 - 000100	TST35 007166	%AUTOB 001034	%LPERR 001010	%TMP4 001070
SW07 - 000200	TST36 007322	%BASE 001154	%MADR1 001132	%TN - 000071
SW08 - 000400	TST37 007434	%BDADR 001022	%MADR2 001136	%TPB 001052
SW09 - 001000	TST4 003734	%BDDAT 001026	%MADR3 001142	%TPFLG 001057
SW1 - 000002	TST40 007614	%CHARC 016564	%MADR4 001146	%TPS 001050
SW10 - 002000	TST41 007764	%CKSMR 017242	%MAIL 001100	%TRAP 020026
SW11 - 004000	TST42 010134	%CHTAG 001000	%MAMS1 001130	%TRAP2 020050
SW12 - 010000	TST43 010256	%CM3 - 000000	%MAMS2 001134	%TRP - 000012
SW13 - 020000	TST44 010372	%CM4 - 000005	%MAMS3 001140	%TRPAD 020062
SW14 - 040000	TST45 010476	%CNTLG 017777	%MAMS4 001144	%TSTM 000504

CJKDFB 11-24 OPTIONS DIAGNOSTIC  
CJKDFB P11 04 JAN 85 11:44

MACY11 30(1046) 04 JAN 85 11:58 PAGE 93  
SYMBOL TABLE

\$STNM 001002	\$TYPEX 016566	\$UNITM 000510	\$XON * 000021	.SERV* 001160
\$TTYIN 017762	\$TYPOC 016614	\$USMR 001124	\$XTSTR 015626	.ST * 000500
\$TYPDS 017016	\$TYPON 016630	\$VECT1 001150	\$GET4= 000000	
\$TYPE 016234	\$TYPOS 016570	\$VECT2 001152	\$OFILL 017013	
\$YPEC 016446	\$UNIT 001112	\$XOFF = 000023	. * 031450	

ABS. 031450 000

ERRORS DETECTED: 0

CJKDFB.CJKDFB/SOL/ML:TOC=SYSMAC SML.CJKDFB.P11  
RUN TIME: 18 20 .6 SECONDS  
RUN-TIME RATIO: 92/40=2.2  
CORE USED: 43K (86 PAGES)