

.REM *

I D E N T I F I C A T I O N

PRODUCT NAME: CNDUSAO DUV11 OFLNE RCVR TMG

PRODUCT CODE: AC-T453A-MC

PRODUCT DATE: DEC, 1982

MAINTAINER : DIAGNOSTICS SERVICES/ISS

AUTHOR: K.LIND

*
.REM *

COPYRIGHT (C) 1982, 1983
DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS.

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

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

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

*

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUV11'S. THIS MEANS THAT
16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC
MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE,
AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL
HALTED.

.REM *

1. THE DUV11 OFFLINE RECEIVER TIMING TESTS VERIFY THAT THE
RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT
AT THE PROPPER TIME

* .REM *

2. REQUIREMENTS

PDP-11/21 COMPUTER (LSI)

DUV11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

- 2.2 STORAGE
THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

* .REM *

3. LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED.

STARTING ADDRESS
FOR ABSOLUTE LOADER

4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

- 4.1 CONTROL SWITCH SETTINGS

NOTE: ALL SWITCHES RESIDE INTERNAL TO THE CPU AT ADDRESS
176. THESE MAY BE SET VIA THE CONSOLE TTY BY DIRECTLY
MODIFYING LOC. 176.

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER
CALLED '\$USWR'. IN ORDER TO BE FLEXIBLE ON THE AVAILABILITY OF THE
H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11.
BIT 0 IN \$USWR REFLECTS THIS STATUS, A 0 = CONNECTOR
PRESENT, A 1 = CONNECTOR NOT AVAILABLE.

THE USER CHANGES THE CONTENTS OF THIS LOCATION
WHEN BUILDING THE E TABLE, BY ANSWERING THE
PROMPT "SWITCH 2".

- 4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
ALL CONSOLE SWITCHES DOWN
- 4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES
SW00=1
- 4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW01=1
- 4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW14=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1
- 4.2 STARTING ADDRESS
THE STARTING ADDRESS FOR ALL TESTS IS 000200
THE RETARTING ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000700
THE STARTING ADDRESS TO LOCK ON TEST IS 000200
- 4.3 PROGRAM AND/OR OPERATOR ACTION
 - 4.3.1 INITIAL PROGRAM START
 - 4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER
 - 4.3.1.2 SET SWITCH REGISTER (LOC. 176) TO ZERO.
 - 4.3.1.3 TYPE 200G.
 - 4.3.1.4 PROGRAM WILL START.
 - 4.3.1.5 THE PROGRAM WILL TYPE 'DUV11 CNDUS-A TAPE C' (ONCE ONLY)
 - 4.3.1.6 THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN
 - 4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN
 - 4.3.2.1 THE PROGRAM WILL TYPE 'R' AND WILL COMMENCE TESTING
 - 4.3.3 PROGRAM RESTART WITH SW00=1

* .REM *

* .REM *

- 4.3.3.1 SET SWITCH REGISTER (LOC. 176) TO A 000001.
- 4.3.3.2 TYPE 200G.
- 4.3.3.3 PROGRAM WILL START.
- 4.3.3.4 THE PROGRAM WILL TYPE " 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD
- 4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

- 4.3.3.6 THE PROGRAM WILL TYPE 'VECTOR ADDRESS-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD
- 4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS FOR THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

- 4.3.3.8 THE PROGRAM WILL TYPE 'ARE YOU RUNNING MULTIPLE DEVICES ?'
(Y OR N)-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD
- 4.3.3.9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?"
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A 'NO' ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12
IF A 'YES' ANSWER IS GIVEN: THE NEXT QUESTION IS ASKED

- 4.3.3.10 THE PROGRAM WILL TYPE 'LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD
- 4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10
NOTE:ALL ADDRESSES SHALL BE CONTIGUOUS

- 4.3.3.11.1 IF AN 'OUT OF RANGE' ADDRESS IS TYPED
IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS).....THE
PROGRAM WILL TYPE 'OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-'

AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

- 4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.11.1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED.....
.....SCHOOLS OUT.....THERE IS NO PROTECTION FOR THIS.
THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM 1ST DEVICE ADDRESS).THE SAME APPLIES TO IDENTICAL ADDRESSES TYPED FOR FIRST AND LAST DEVICE.
OBSERVE LOCATION @ ACTREG: SEE SECTION 7.2

- 4.3.3.12 THE PROGRAM WILL TYPE '# OF SYNC CHARS SELECTED (1 OR 2)-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD. REFER TO MANUAL FOR PROPER SWITCH SETTINGS OF SWITCH E55-4.

- 4.3.3.13 TYPE IN THE APPROPRIATE ANSWER '1' OR '2' FOLLOWED BY A <CARRIAGE RETURN>.(NOTE:ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.12

- 4.3.3.14 THE PROGRAM WILL TYPE 'IS SEC XMIT SWITCH E55-2 ON? (Y OR N)-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

- 4.3.3.15 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>.(NOTE THAT ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.14

- 4.3.3.16 THE PROGRAM WILL TYPE 'IS SEC REC SWITCH E55-3 ON? (Y OR N)-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

- 4.3.3.17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4.3.3.16

- 4.3.3.18 THE PROGRAM WILL TYPE 'IS OPT CLR ENABLE SWITCH E55-1 ON? (Y OR N)-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

- 4.3.3.19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 57-5
 CNDUS1.M11 10-DEC-82 13:49

SEQ 0006

BY A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE '?'
 AND WILL REPEAT THE MESSAGE OF 4.3.3.18

4.3.3.20 THE PROGRAM WILL TYPE 'ARE YOU RUNNING IN MAINT.
 MODE EXTERNAL ? ANDDO YOU HAVE THE EXTERNAL MODEM
 BYPASS JUMPER CONNECTOR ON ? (Y OR N)-' AND WAIT FOR AN
 INPUT FROM THE TELETYPE KEYBOARD

4.3.3.21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY
 A <CARRIAGE RETURN>. (NOTE: ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE '?'
 AND WILL REPEAT THE MESSAGE OF 4.3.3.20

4.3.3.22 THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT
 HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4.3.4 PROGRAM RESTART WITH SW01=1
 NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
 ,,,IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED
 IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200,
 AND SELECT SW00=1 AND ANSWER 'NO' TO THE MULTIPLE DEVICE QUESTION
 SEE 4.3.3

4.3.4.1 SET SW01=1 IN SWITCH REG (LOC. 176)

4.3.4.2 TYPE 200G.

4.3.4.3 PROGRAM WILL START.

4.3.4.4 THE PROGRAM WILL TYPE 'TEST PC-' AND WAIT FOR AN INPUT FROM
 THE TELETYPE KEYBOARD

4.3.4.5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO
 BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4.3.4.6 THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT HAS STARTED
 TESTING AT THE SELECTED TEST

NOTE: CARE MUST BE TAKEN WHEN THIS FEATURE IS USED
 ,SINCE THERE IS NO PROTECTION AGAINST SELECTING AN ADDRESS
 THAT IS IN THE MIDDLE OF A TEST

4.3.5 PROGRAM RESTART WITH SW14 =1
 NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
 SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW14=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

4.3.5.4 THE PROGRAM WILL TYPE 'LOCK ON SELECTED TEST ? (Y OR N)-'
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
<CARRIAGE RETURN>

IF A NO ANSWER IS GIVEN: THIS LOCK ON TEST WILL BE IGNORED
AND THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1

4.3.5.6 IF A YES ANSWER WAS GIVEN: THE PROGRAM WILL ACT AS FOLLOWS...
THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED
OR IF ANY KEY IS STRUCK ON THE TELETYPE, THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 =1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC. 176).

SW15 =1 HALT ON ERROR
SW14 =1 LOOP ON CURRENT TEST
SW13 =1 INHIBIT ERROR TYPEOUT
SW11 =1 INHIBIT ITERATIONS
SW10 =1 ESCAPE TO NEXT TEST ON ERROR
SW09 =1 LOOP ON ERROR
SW01 =1 RESTART PROGRAM AT SELECTED TEST
SW00 =1 RESELECT VECTOR AND CONTROL REGISTER ADDRESSES
&PARAMETERS AFTER A PROGRAM RESTART
TO INHIBIT 'END OF PASS' TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O.D.T.) THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6.1.1 PC+2 = ERROR PC WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2 REFER TO THE ABOVE 'HLT' IN DIAGNOSTIC FOR ERROR DESCRIPTION

CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER
TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC REGISTER EXPECTED ACTUAL 16XXXX YYYYYY ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

6.2.1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6.2.2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE 'HLT' WILL BE DISPLAYED IN THE DATA LIGHTS

6.2.3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR.

6.2.4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT: & PASCNT:
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY.
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.

6.3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM:

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

7.1 MULTIPLE DEVICES

UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY
MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

NOTE: IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR
YOU CAN CHANGE "ZERO: ADD #10,BASEIV ;NEXT BLOCK
(VECTORS)" TO "ZERO: ADD #0,BASEIV";
THEREBY THE VECTOR ADDRESSES WILL NOT BE
UPDATED AFTER EACH PASS.

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

WHEN RUNNING MULTIPLE DEVICES AN ACTIVE BIT IS SET
FOR EACH DEVICE RUNNING UNDER TEST IE. BIT 0 FOR
DEVICE 0 ,BIT 15 FOR DEVICE 15
TO DISQUALIFY DEVICES:

7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED ,SIMPLY RESTART PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.

7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG: OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0) AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

7.2.2.1 TO RESTART...TYPE 200G...
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

7.2.2.2ORSET SW00=1 IN SWITCH REG (LOC. 176) AND TYPE 200G....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
.....THE PROGRAM WILL CONTINUE WITH DEVICE 0

7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM
WILL TYPEOUT AN ERROR MESSAGE.....TYPE 200G.

7.3 CABLE DELAYS

NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE, LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES. PRESENTLY "HOLD:" =20 IS SUFFICIENT TIME ON AN 11/21 MACHINE.

BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS

7.4 TO USE THE "XOR" TESTER ,THE BRANCH AROUND THE "XOR"

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 57-9
 CNDUS1.M11 10-DEC-82 13:49

SEQ 0010

CODE MUST BE PATCHED TO A 'NOP'. (SEE LISTINGS FOR DETAILS)

8. DEFAULT PARAMETERS:
 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 174300
 VECTOR ADDRESS- DURIV: 330
 ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0
 LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0
 # OF SYNC CHARS SELECTED - 2 SYNCNO: 377
 IS SEC XMIT SWITCH E55-2 ON?- YES SEXMIT: 377
 IS SEC REC SWITCH E55-3 ON?- YES SEREC: 377
 IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR: 377
 DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
 CONNECTOR ON (H315)- YES JMRBY: 377

9. PROGRAM DESCRIPTION

- 9.1 THIS PROGRAM PERFORMS THE OFFLINE RECEIVER TIMING TESTING
 OF THE DEVICE
 SEE LISTING FOR DETAILS

10. FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

11. REVISION HISTORY
 DZDUSB1 DIAGNOSTIC WAS MODIFIED TO WORK IN SBC 11/21 PROCESSOR
 PRIORITY 340 WAS CHANGED TO 300 WHEREVER ENCOUNTERED
 DEFAULT CSR AND VECTOR ADDRESSES WERE CHANGED
 .INIT CALL ADDED AT THE END OF PROGRAM
 DIAGNOSTIC WAS RENAMED TO CNDUSAO.
 12. LISTINGS

* .REM *

* .REM *

*

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 58
 CNDUS2.M11 30-OCT-82 12:20

SEQ 0011

6488
 6489
 6490
 6491
 6585
 6586
 6587
 6598
 6612
 6613
 6614
 6675
 6676
 6881
 7009
 7010

;;IN ORDER TO BE SPECIFIC TO SBC 11/21 PROCESSOR
 ;;ODT VECTOR ADDRESS AND BEVNT WERE INITIALIZED AND
 ;;ALL PRIORITY WORD 340 WERE MODIFIED TO 300.

.SBTTL APT COMMUNICATIONS ROUTINE

```

(1)
(2)
(1) 000000 112767 000001 000236 $ATY1: MOVB #1,$FFLG ;;TO REPORT FATAL ERROR
(1) 000006 112767 000001 000226 $ATY3: MOVB #1,$MFLG ;;TO TYPE A MESSAGE
(1) 000014 000403 BR $ATYC
(1) 000016 112767 000001 000220 $ATY4: MOVB #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
(1) 000024 $ATYC:
(3) 000024 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
(3) 000026 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
(1) 000030 105767 000206 TSTB $MFLG ;;SHOULD TYPE A MESSAGE?
(1) 000034 001450 BEQ 5$ ;;IF NOT: BR
(1) 000036 122767 000001 001502 CMPB #APTENV,$ENV ;;OPERATING UNDER APT?
(1) 000044 001031 BNE 3$ ;;IF NOT: BR
(1) 000046 132767 000100 001473 BITB #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(1) 000054 001425 BEQ 3$ ;;IF NOT: BR
(1) 000056 017600 000004 MOV @4(SP),R0 ;;GET MESSAGE ADDR.
(1) 000062 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDR.
(1) 000070 005767 001432 1$: TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
(1) 000074 001375 BNE 1$ ;;IF NOT: WAIT
(1) 000076 010067 001440 MOV R0,$MSGAD
(1) ;;PUT ADDR IN MAILBOX
(1) 2$: TSTB (R0)+ ;;FIND END OF MESSAGE
(1) 000102 105720 BNE 2$
(1) 000104 001376 BNE 2$
(1) 000106 166700 001430 SUB $MSGAD,R0 ;;SUB START OF MESSAGE
(1) 000112 006200 ASR R0 ;;GET MESSAGE LGTH IN WORDS
(1) 000114 010067 001424 MOV R0,$MSGGLT ;;PUT LENGTH IN MAILBOX
(1) 000120 012767 000004 001400 MOV #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
(1) 000126 000413 BR 5$
(1) 000130 017667 000004 000016 3$: MOV @4(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
(1) 000136 062766 000002 000004 ADD #2,4(SP) ;;BUMP RETURN ADDRESS
(3) 000144 016746 177626 MOV 177776,-(SP) ;;PUSH 177776 ON STACK
(1) 000150 004767 012702 JSR PC,$TYPE ;;CALL TYPE MACRO
(1) 000154 000000 4$: .WORD 0
(1) 000156 5$:
(1) 000156 105767 000062 10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
(1) 000162 001416 BEQ 12$ ;;IF NOT: BR
(1) 000164 005767 001356 TST $ENV ;;RUNNING UNDER APT?
(1) 000170 001413 BEQ 12$ ;;IF NOT: BR
(1) 000172 005767 001330 11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
(1) 000176 001375 BNE 11$ ;;IF NOT: WAIT

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 58-1
 CNDUS2.M11 30-OCT-82 12:20 APT COMMUNICATIONS ROUTINE

SEQ 0012

```

(1) 000200 017667 000004 001322      MOV      @4(SP), $FATAL      ;;GET ERROR #
(1) 000206 062766 000002 000004      ADD      #2, 4(SP)      ;;BUMP RETURN ADDR.
(1) 000214 005267 001306              INC      $MSGTYPE      ;;TELL APT TO TAKE ERROR
(1) 000220 105067 000020      12$:    CLRB     $FFLG      ;;CLEAR FATAL FLAG
(1) 000224 105067 000013      CLRB     $LFLG      ;;CLEAR LOG FLAG
(1) 000230 105067 000006      CLRB     $MFLG      ;;CLEAR MESSAGE FLAG
(3) 000234 012601              MOV      (SP)+, R1      ;;POP STACK INTO R1
(3) 000236 012600              MOV      (SP)+, R0      ;;POP STACK INTO R0
(1) 000240 000207              RTS      PC      ;;RETURN
(1) 000242      000      $MFLG: .BYTE 0      ;;MESSG. FLAG
(1) 000243      000      $LFLG: .BYTE 0
(1)              ;;LOG FLAG
(1) 000244      000      $FFLG: .BYTE 0      ;;FATAL FLAG
(1)              .EVEN
(1)              APTSIZE=200
(1)              APTENV=001
(1)              APTSPool=100
(1)              APTCSUP=040
7249      000001      $TN=1
7375
7379
7414
7431
7444
7456
7469

```

N 1

CNDUS-A MACY1: 30(1046) 14-DEC-82 09:59 PAGE 60
CNDUS2.M11 30-OCT-82 12:20 APT COMMUNICATIONS ROUTINE

SEQ 0013

7492
7558
7564
7700
7728
7761
7802
7833
7884
7932
7985
8000
8012
8114

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61
 CNDUSA.M11 30-OCT-82 10:55 APT COMMUNICATIONS ROUTINE

SEQ 0014

```

8143      ;IN ORDER TO BE SPECIFIC TO SBC 11/21 PROCESSOR ALL PRIORITY
8144      ;WORD 340 WERE MODIFIED TO 300 AND PROGRAM RENAMED.
8149      .ENABLE ABS
8150
(2)      ;DUV11 CNDUS-A TAPE C
(2)      ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
(2)
(2)      ;STARTING PROCEDURE
(2)      ;TYPE 200G
(2)      ;PROGRAM WILL TYPE 'DUV11 CNDUS-A TAPE C '
(2)      ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
(2)      ;AT THE END OF A PASS, PROGRAM WILL TYPE 'END OF PASS TAPE C'
(2)      ;AND THEN RESUME TESTING
(2)
(2)      .SBTTL BASIC DEFINITIONS
(2)
(2)      ;*INITIAL ADDRESS OF THE STACK POINTER *** 11 J ***
(2)      001100 STACK= 1100
(2)      .EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
(2)      .EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
(2)
(2)      ;*MISCELLANEOUS DEFINITIONS
(2)      000011 HT= 11          ;;CODE FOR HORIZONTAL TAB
(2)      000012 LF= 12          ;;CODE FOR LINE FEED
(2)      000015 CR= 15          ;;CODE FOR CARRIAGE RETURN
(2)      000200 CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
(2)      177776 PS= 177776     ;;PROCESSOR STATUS WORD
(2)      .EQUIV PS,PSW
(2)      177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
(2)      177772 PIRQ= 177772   ;;PROGRAM INTERRUPT REQUEST REGISTER
(2)      177570 DSWR= 177570   ;;HARDWARE SWITCH REGISTER
(2)      177570 DDISP= 177570  ;;HARDWARE DISPLAY REGISTER
(2)      ;***** THE FOLLOWING ODT START ADDRESS FOR SBC 11/21 IS ADDED
(2)      170000 ODTST= 170000
(2)      ;*GENERAL PURPOSE REGISTER DEFINITIONS
(2)      000000 R0= %0          ;;GENERAL REGISTER
(2)      000001 R1= %1          ;;GENERAL REGISTER
(2)      000002 R2= %2          ;;GENERAL REGISTER
(2)      000003 R3= %3          ;;GENERAL REGISTER
(2)      000004 R4= %4          ;;GENERAL REGISTER
(2)      000005 R5= %5          ;;GENERAL REGISTER
(2)      000006 R6= %6          ;;GENERAL REGISTER
(2)      000007 R7= %7          ;;GENERAL REGISTER
(2)      000006 SP= %6          ;;STACK POINTER
(2)      000007 PC= %7          ;;PROGRAM COUNTER
(2)
(2)      ;*PRIORITY LEVEL DEFINITIONS
(2)      000000 PR0= 0          ;;PRIORITY LEVEL 0
(2)      000040 PR1= 40         ;;PRIORITY LEVEL 1
(2)      000100 PR2= 100        ;;PRIORITY LEVEL 2
(2)      000140 PR3= 140        ;;PRIORITY LEVEL 3
(2)      000200 PR4= 200        ;;PRIORITY LEVEL 4
(2)      000240 PR5= 240        ;;PRIORITY LEVEL 5
(2)      000300 PR6= 300        ;;PRIORITY LEVEL 6
(2)      000340 PR7= 340        ;;PRIORITY LEVEL 7

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-1
 CNDUSA.M11 30-OCT-82 10:55 BASIC DEFINITIONS

SEQ 0015

```

(2)          : *SWITCH REGISTER SWITCH DEFINITIONS
(2)          100000 SW15= 100000
(2)          040000 SW14= 40000
(2)          020000 SW13= 20000
(2)          010000 SW12= 10000
(2)          004000 SW11= 4000
(2)          002000 SW10= 2000
(2)          001000 SW09= 1000
(2)          000400 SW08= 400
(2)          000200 SW07= 200
(2)          000100 SW06= 100
(2)          000040 SW05= 40
(2)          000020 SW04= 20
(2)          000010 SW03= 10
(2)          000004 SW02= 4
(2)          000002 SW01= 2
(2)          000001 SW00= 1
(2)          .EQUIV SW09,SW9
(2)          .EQUIV SW08,SW8
(2)          .EQUIV SW07,SW7
(2)          .EQUIV SW06,SW6
(2)          .EQUIV SW05,SW5
(2)          .EQUIV SW04,SW4
(2)          .EQUIV SW03,SW3
(2)          .EQUIV SW02,SW2
(2)          .EQUIV SW01,SW1
(2)          .EQUIV SW00,SW0
(2)          : *DATA BIT DEFINITIONS (BIT00 TO BIT15)
(2)          100000 BIT15= 100000
(2)          040000 BIT14= 40000
(2)          020000 BIT13= 20000
(2)          010000 BIT12= 10000
(2)          004000 BIT11= 4000
(2)          002000 BIT10= 2000
(2)          001000 BIT09= 1000
(2)          000400 BIT08= 400
(2)          000200 BIT07= 200
(2)          000100 BIT06= 100
(2)          000040 BIT05= 40
(2)          000020 BIT04= 20
(2)          000010 BIT03= 10
(2)          000004 BIT02= 4
(2)          000002 BIT01= 2
(2)          000001 BIT00= 1
(2)          .EQUIV BIT09,BIT9
(2)          .EQUIV BIT08,BIT8
(2)          .EQUIV BIT07,BIT7
(2)          .EQUIV BIT06,BIT6
(2)          .EQUIV BIT05,BIT5
(2)          .EQUIV BIT04,BIT4
(2)          .EQUIV BIT03,BIT3
(2)          .EQUIV BIT02,BIT2
(2)          .EQUIV BIT01,BIT1
(2)          .EQUIV BIT00,BIT0

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-2
 CNDUSA.M11 30-OCT-82 10:55 BASIC DEFINITIONS

SEQ 0016

(2)		;*BASIC "CPU" TRAP VECTOR ADDRESSES	
(2)	000004	ERRVEC= 4	:::TIME OUT AND OTHER ERRORS
(2)	000010	RESVEC= 10	:::RESERVED AND ILLEGAL INSTRUCTIONS
(2)	000014	TBITVEC=14	:::"T" BIT
(2)	000014	TRTVEC= 14	:::TRACE TRAP
(2)	000014	BPTVEC= 14	:::BREAKPOINT TRAP (BPT)
(2)	000020	IOTVEC= 20	:::INPUT/OUTPUT TRAP (IOT) **SCOPE**
(2)	000024	PWRVEC= 24	:::POWER FAIL
(2)	000030	EMTVEC= 30	:::EMULATOR TRAP (EMT) **ERROR**
(2)	000034	TRAPVEC=34	:::"TRAP" TRAP
(2)	000060	TKVEC= 60	:::TTY KEYBOARD VECTOR
(2)	000064	TPVEC= 64	:::TTY PRINTER VECTOR
(2)		***** THE FOLLOWING BREAK VECTOR AND LINE CLOCK VECTOR ARE INCLUDED	
(2)	000100	LKVEC= 100	:::LINE CLOCK VECTOR
(2)	000140	BRKVEC= 140	:::BREAK VECTOR
(2)	000240	PIRQVEC=240	:::PROGRAM INTERRUPT REQUEST VECTOR

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-3
 CNDUSA.M11 30-OCT-82 10:55 BASIC DEFINITIONS

SEQ 0017

```

(2)                                ;STANDARD INTERRUPT VECTORS
(2)
(2)
(2) 000174 000174                .=174
(2) 000174 000000                DISPREG:0
(2) 000176 000000                SWREG:0
(2) 000200 000200                .=200
(2) 000200 000167 001746        JMP      .START          ;GO TO START OF PROGRAM
(2)
(2)
(2) 001100 001100                .=1100
(2) 001100 000000                .WORD 0
(2) 001102 177570                LIGHTS:177570
(2)
(2)
(2)                                ;PROGRAM CONTROL PARAMETERS
(2)
(2) 001104 000000                RETURN: 0
(2) 001106 000000                NEXT: 0                    ;ADDRESS OF NEXT TEST TO BE EXECUTED
(2) 001110 000000                LOCK: 0                    ;ADDRESS FOR LOCK ON CURRENT DATA
(2) 001112 000000                PASCNT: 0                  ;ADDRESS CONTAINING PASS COUNT
(2) 001114 000000                ERRCNT: 0                  ;ERROR COUNT
(2) 001116 000000                SAVSP: 0                   ;STACK POINTER STORAGE
(2)
(2)                                ;PROGRAM VARIABLES
(2)
(2) 001120 000020                HOLD: 20                   ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
(2) 001122 000000                SHIFT: 0                  ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
(2) 001124 000000                COUNT: 0                   ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
(2) 001126 000000                SAVPC: 0                   ;PROGRAM COUNTER STORAGE
(2) 001130 000000                HLD0: 0
(2) 001132 000000                HLD1: 0
(2) 001134 000000                HLD2: 0
(2) 001136 000000                HLD3: 0
(2) 001140 000000                HLD4: 0
(2) 001142 000000                HLD5: 0
(2) 001144 000000                HLD6: 0

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-4
 CNDUSA.M11 30-OCT-82 10:55 BASIC DEFINITIONS

SEQ 0018

```

(2)                                     ;PROGRAM CONVERSATIONAL PARAMETERS
(2) 001146      377      SYNCNO: .BYTE 377      ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
(2) 001147      377      SEXMIT: .BYTE 377      ;SEC XMIT JUMPER 'IN'
(2) 001150      377      SEREC:  .BYTE 377      ;SEC REC JUMPER 'IN'
(2) 001151      377      OPTCLR: .BYTE 377      ;OPTIONAL JUMPER CLR 'IN'
(2) 001152      000      MULTD:  .BYTE 0        ;NO MULTIPLE DEVICE FLAG
(2) 001153      377      JMRBY:  .BYTE 377      ;EXTERNAL MODEM BYPASS JUMPER 'IN'
(2)                                     .EVEN
(2)
(2)                                     ;PROGRAM MULTIPLE DEVICE PARAMETERS
(2) 001154      000000    BASEADD:      0        ;PROG CONTROLLED 1ST DEVICE ADDR
(2) 001156      000000    KEEPADD:     0        ;SAVED 1ST DEVICE ADDR
(2) 001160      000000    LASTADD:     0        ;LAST DEVICE RXCSR ADDR
(2) 001162      000000    BASEIV:      0        ;PROG CONTROLLED IV
(2) 001164      000000    KEEPIV:      0        ;SAVED INTR VECTOR
(2) 001166      000000    ACTREG:      0        ;ACTIVE REGISTER,,,MODIFY THIS
(2)                                     ;LOCATION TO DISQUALIFY OR QUALIFY
(2)                                     ;DEVICES (1= RUN,,,0= DON'T RUN)
(2) 001170      000000    ROTADD:      0        ;ROTATING POINTER FOR ACTREG..POINTS
(2)                                     ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES
(2)
(2)                                     ;PROGRAM CONTROL FLAGS
(2)
(2) 001172      000      INIFLG: .BYTE 0        ;PROGRAM INITIALIZATION FLAG
(2) 001173      000      STFLG:  .BYTE 0        ;TEST START FLAG
(2) 001174      000      LOKFLG: .BYTE 0        ;LOCK ON CURRENT TEST FLAG
(2)                                     .EVEN
(1)                                     .=1400
(2)
(2)

```


CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-6
CNDUSA.M11 30-OCT-82 10:55 BASIC DEFINITIONS

SEQ 0020

(2)	000040	DNAINT=BIT5	;DNA INTR ENAB
(2)	000020	SEND=BIT4	;SEND
(2)	000010	HDXEN=BIT3	;HDX/FDX
(2)	000001	BREAK=BIT0	;BREAK
(2)		;TXCSR WRD DEFINITIONS	
(2)	000000	USER=0	;USER MODE
(2)	004000	MINT=4000	;MAINT INT MODE
(2)	010000	MEXT=10000	;MAINT EXT MODE
(2)	014000	SYSTST=14000	;SYSTEM TEST MODE

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-7
 CNDUSA.M11 30-OCT-82 10:55 COMMON TAGS

SEQ 0021

```

(2) .SBTTL COMMON TAGS
(2)
(3) *****
(2) *THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(2) *USED IN THE PROGRAM.
(2)
(2) 001400
(2) 001400 000000 SCMTAG: .=. ::START OF COMMON TAGS
(2) 001402 000 $TSTNM: .BYTE 0 ::CONTAINS THE TEST NUMBER
(2) 001403 000 $SERFLG: .BYTE 0 ::CONTAINS ERROR FLAG
(2) 001404 000000 $ICNT: .WORD 0 ::CONTAINS SUBTEST ITERATION COUNT
(2) 001406 000000 $LPADR: .WORD 0 ::CONTAINS SCOPE LOOP ADDRESS
(2) 001410 000000 $LPERR: .WORD 0 ::CONTAINS SCOPE RETURN FOR ERRORS
(2) 001412 000000 $ERTYL: .WORD 0 ::CONTAINS TOTAL ERRORS DETECTED
(2) 001414 000 $ITEMB: .BYTE 0 ::CONTAINS ITEM CONTROL BYTE
(2) 001415 001 $ERMAX: .BYTE 1 ::CONTAINS MAX. ERRORS PER TEST
(2) 001416 000000 $ERRPC: .WORD 0 ::CONTAINS PC OF LAST ERROR INSTRUCTION
(2) 001420 000000 $GDADR: .WORD 0 ::CONTAINS ADDRESS OF 'GOOD' DATA
(2) 001422 000000 $BDADR: .WORD 0 ::CONTAINS ADDRESS OF 'BAD' DATA
(2) 001424 000000 $GDDAT: .WORD 0 ::CONTAINS 'GOOD' DATA
(2) 001426 000000 $BDDAT: .WORD 0 ::CONTAINS 'BAD' DATA
(2) 001430 000000 .WORD 0 ::RESERVED--NOT TO BE USED
(2) 001432 000000 .WORD 0
(2) 001434 000 $AUTOB: .BYTE 0 ::AUTOMATIC MODE INDICATOR
(2) 001435 000 $INTAG: .BYTE 0 ::INTERRUPT MODE INDICATOR
(2) 001436 000000 .WORD 0
(2) 001440 177570 SWR: .WORD DSWR ::ADDRESS OF SWITCH REGISTER
(2) 001442 177570 DISPLAY: .WORD DDISP ::ADDRESS OF DISPLAY REGISTER
(2) 001444 177560 $TKS: 177560 ::TTY KBD STATUS
(2) 001446 177562 $TKB: 177562 ::TTY KBD BUFFER
(2) 001450 177564 $TPS: 177564 ::TTY PRINTER STATUS REG. ADDRESS
(2) 001452 177566 $TPB: 177566 ::TTY PRINTER BUFFER REG. ADDRESS
(2) 001454 000 $NULL: .BYTE 0 ::CONTAINS NULL CHARACTER FOR FILLS
(2) 001455 002 $FILLS: .BYTE 2
(2) ::CONTAINS # OF FILLER CHARACTERS REQUIRED
(2) 001456 012 $FILLC: .BYTE 12 ::INSERT FILL CHARS. AFTER A 'LINE FEED'
(2) 001457 000 $TPFLG: .BYTE 0 ::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
(2) 001460 000000 $REGAD: .WORD 0 ::CONTAINS THE ADDRESS FROM
(2) WHICH ($REG0) WAS OBTAINED
(4) 001462 000000 $REG0: .WORD 0 ::CONTAINS (($REGAD)+0)
(4) 001464 000000 $REG1: .WORD 0 ::CONTAINS (($REGAD)+2)
(4) 001466 000000 $REG2: .WORD 0 ::CONTAINS (($REGAD)+4)
(4) 001470 000000 $REG3: .WORD 0 ::CONTAINS (($REGAD)+6)
(4) 001472 000000 $REG4: .WORD 0 ::CONTAINS (($REGAD)+10)
(4) 001474 000000 $REG5: .WORD 0 ::CONTAINS (($REGAD)+12)
(4) 001476 000000 $TMP0: .WORD 0 ::USER DEFINED
(4) 001500 000000 $TMP1: .WORD 0 ::USER DEFINED
(4) 001502 000000 $TMP2: .WORD 0 ::USER DEFINED
(4) 001504 000000 $TMP3: .WORD 0 ::USER DEFINED
(4) 001506 000000 $TMP4: .WORD 0 ::USER DEFINED
(4) 001510 000000 $TMP5: .WORD 0 ::USER DEFINED
(2) 001512 000000 $TIMES: 0 ::MAX. NUMBER OF ITERATIONS
(2) 001514 000000 $ESCAPE: 0 ::ESCAPE ON ERROR ADDRESS
(2) 001516 177607 $BELL: .ASCII <207><377><377> ::CODE FOR BELL
(2) 001522 077 $QUES: .ASCII /?/ ::QUESTION MARK
  
```

000377

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-8
 CNDUSA.M11 30-OCT-82 10:55 COMMON TAGS

SEQ 0022

```

(2) 001523 015 $CRLF: .ASCII <15> ;;CARRIAGE RETURN
(2) 001524 000012 $LF: .ASCIIZ <12> ;;LINE FEED
(3) *****
(3) $BTTL APT MAILBOX-ETABLE
(4) *****
(3) .EVEN
(3) $MAIL: ;;APT MAILBOX
(3) 001526 000000 $MSGTY: .WORD AMSTY ;;MESSAGE TYPE CODE
(3) 001530 000000 $FATAL: .WORD AFATAL ;;FATAL ERROR NUMBER
(3) 001532 000000 $TESTN: .WORD ATESTN ;;TEST NUMBER
(3) 001534 000000 $PASS: .WORD APASS ;;PASS COUNT
(3) 001536 000000 $DEVCT: .WORD ADEVCT ;;DEVICE COUNT
(3) 001540 000000 $UNIT: .WORD AUNIT ;;I/O UNIT NUMBER
(3) 001542 000000 $MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS
(3) 001544 000000 $MSGLG: .WORD AMSLG ;;MESSAGE LENGTH
(3) 001546 $ETABLE: ;;APT ENVIRONMENT TABLE
(3) 001546 000 $ENV: .BYTE AENV ;;ENVIRONMENT BYTE
(3) 001547 000 $ENVM: .BYTE AENVM
(3) ;;ENVIRONMENT MODE BITS
(3) 001550 000000 $SWREG: .WORD ASWREG ;;APT SWITCH REGISTER
(3) 001552 000000 $USWR: .WORD AUSWR ;;USER SWITCHES
(3) 001554 000000 $CPUOP: .WORD ACPUOP ;;CPU TYPE,OPTIONS
(3) * BITS 15-11=CPU TYPE
(3) * 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(3) * 11/70=06,PDQ=07,Q=10
(3) * BIT 10=REAL TIME CLOCK
(3) * BIT 9=FLOATING POINT PROCESSOR
(3) * BIT 8=MEMORY MANAGEMENT
(3) 001556 000 $MAMS1: .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE
(3) 001557 000 $MTYP1: .BYTE AMTYP1 ;;MEM. TYPE,BLK#1
(3) * MEM.TYPE BYTE -- (HIGH BYTE)
(3) * 900 NSEC CORE=001
(3) * 300 NSEC BIPOLAR=002
(3) * 500 NSEC MOS=003
(3) 001560 000000 $MADR1: .WORD AMADR1 ;;HIGH ADDRESS,BLK#1
(3) * MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
(3) 001562 000 $MAMS2: .BYTE AMAMS2 ;;HIGH ADDRESS,M.S. BYTE
(3) 001563 000 $MTYP2: .BYTE AMTYP2 ;;MEM.TYPE,BLK#2
(3) 001564 000000 $MADR2: .WORD AMADR2 ;;MEM.LAST ADDRESS,BLK#2
(3) 001566 000 $MAMS3: .BYTE AMAMS3 ;;HIGH ADDRESS,M.S.BYTE
(3) 001567 000 $MTYP3: .BYTE AMTYP3 ;;MEM.TYPE,BLK#3
(3) 001570 000000 $MADR3: .WORD AMADR3 ;;MEM.LAST ADDRESS,BLK#3
(3) 001572 000 $MAMS4: .BYTE AMAMS4 ;;HIGH ADDRESS,M.S.BYTE
(3) 001573 000 $MTYP4: .BYTE AMTYP4 ;;MEM.TYPE,BLK#4
(3) 001574 000000 $MADR4: .WORD AMADR4 ;;MEM.LAST ADDRESS,BLK#4
(3) 001576 000000 $VECT1: .WORD AVECT1 ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(3) 001600 000000 $VECT2: .WORD AVECT2 ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(3) 001602 000000 $BASE: .WORD ABASE
(3) *;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(3) 001604 000000 $DEVM: .WORD ADEVM ;;DEVICE MAP
(3) 001606 000000 $CDW1: .WORD ACDW1 ;;CONTROLLER DESCRIPTION WORD#1
(3) 001610 000000 $CDW2: .WORD ACDW2 ;;CONTROLLER DESCRIPTION WORD#2
(3) 001612 000000 $DDW0: .WORD ADDW0 ;;DEVICE DESCRIPTOR WORD#0
(3) 001614 000000 $DDW1: .WORD ADDW1 ;;DEVICE DESCRIPTOR WORD#1
(3) 001616 000000 $DDW2: .WORD ADDW2 ;;DEVICE DESCRIPTOR WORD#2

```


CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-11
CNDUSA.M11 30-OCT-82 10:55 APT MAILBOX-ETABLE

SEQ 0025

(4)	000040	DNAINTE=BIT5	;DNA INTR ENAB
(4)	000020	SEND=BIT4	;SEND
(4)	000010	HDXEN=BIT3	;HDX/FDX
(4)	000001	BREAK=BIT0	;BREAK
(4)		;TXCSK WRD DEFINITIONS	
(4)	000000	USER=0	;USER MODE
(4)	004000	MINT=4000	;MAINT INT MODE
(4)	010000	MEXT=10000	;MAINT EXT MODE
(4)	014000	SYSTST=14000	;SYSTEM TEST MODE

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-12
 CNDUSA.M11 30-OCT-82 10:55 ERROR POINTER TABLE

SEQ 0026

```

(2)      .SBTTL  ERROR POINTER TABLE
(2)
(2)      ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(2)      ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(2)      ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(2)      ;*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(2)      ;*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(2)
(2)      ;*      EM      ;;POINTS TO THE ERRJR MESSAGE
(2)      ;*      DH      ;;POINTS TO THE DATA HEADER
(2)      ;*      DT      ;;POINTS TO THE DATA
(2)      ;*      DF      ;;POINTS TO THE DATA FORMAT
(2)
(2)      $ERRTB:
(1)      ;ERROR TABLE
(1)      001652      001762      EM1      ;ERROR 1      REGISTER ERROR
(1)      001654      002067      DH1
(1)      001656      002116      DT1
(1)      001660      002132      DF1
(1)      001662      002022      EM2      ;ERROR 2      RECEIVER ERROR
(1)      001664      002067      DH1
(1)      001666      002116      DT1
(1)      001670      002132      DF1
(1)      001672      002043      EM3      ;ERROR 3      TRANSMITTER ERROR
(1)      001674      002067      DH1
(1)      001676      002116      DT1
(1)      001700      002132      DF1
(1)      001702      001746      EM4      ;ERROR 4      BIT ERROR (GENERAL)
(1)      001704      000600      0
(1)      001706      002126      DT4
(1)      001710      002132      DF1
(1)
(1)      ;DEFAULT DU ADDRESSES
(1)      001712      174300      RXCSR: 174300
(1)      001714      174301      HRXCSR: 174301
(1)      001716      174302      RXDBUF: 174302
(1)      001720      174303      HRXDBUF: 174303
(1)      001722      174302      PARCSR: 174302
(1)      001724      174303      HPARCSR: 174303
(1)      001726      174304      TXCSR: 174304
(1)      001730      174305      HTXCSR: 174305
(1)      001732      174306      TXDBUF: 174306
(1)      001734      174307      HTXDBUF: 174307
(1)
(1)      ;DEFAULT DU VECTORS
(1)      001736      000330      DURIV: 330      ;REC INTR VECTOR
(1)      001740      000332      DURIS: 332      ;REC INTR STATUS
(1)      001742      000334      DUTIV: 334      ;XMIT INTR VECTOR
(1)      001744      000336      DUTIS: 336      ;XMIT INTR STATUS
(1)
(1)      ;ERROR MESSAGES
(1)      001746      020040      051105      047522      EM4:      .ASCIZ / ERROR PC /
(1)      001754      020122      041520      000040
(1)      001762      020040      047503      050115      EM1:      .ASCIZ / COMPARISON ERROR ON REGISTERS/
(1)      001770      051101      051511      047117
(1)      001776      042440      051122      051117
(1)      002004      047440      020116      042522

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-13
 CNDUSA.M11 30-OCT-82 10:55 ERROR POINTER TABLE

SEQ 0027

```

(1) 002012 044507 052123 051105
(1) 002020 000123
(1) 002022 020040 042522 042503 EM2: .ASCIZ / RECEIVER ERROR/
(1) 002030 053111 051105 042440
(1) 002036 051122 051117 000
(1) 002043 040 052040 040522 EM3: .ASCIZ / TRANSMITTER ERROR/
(1) 002050 051516 044515 052124
(1) 002056 051105 042440 051122
(1) 002064 051117 000
(1)
(1) 002067 105 051122 041520 DH1: ;DATA HEADERS FOR ERROR MESSAGES
(1) 002074 020040 040527 052116 ;ASCIZ /ERRPC WANTED ACTUAL/
(1) 002102 042105 020040 041501
(1) 002110 052524 046101 000
(1) 002116 001416 001130 001132 DT1: .EVEN ;DATA TABLES FOR ERROR MESSAGES
(1) 002124 000000 .WORD $ERRPC,HLD0,HLD1,0
(1) 002126 001416 000000 DT4: .WORD $ERRPC,0
(1) 002132 000 000 000 DF1: .BYTE 0,0,0,0
(1) 002135 000
(1) .EVEN
(2) .SBTTL ACT11 HOOKS
(2)
(3) ;*****
(2) ;HOOKS REQUIRED BY ACT11
(2) $SVPC=. ;SAVE PC
(2) .=46
(2) 000046 012660 $ENDAD ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
(2) 000052 000052 .=52
(2) 000000 .WORD 0 ;:2)SET LOC.52 TO ZERO
(2) 002136 .=$SVPC ;: RESTORE PC
(2) .SBTTL APT PARAMETER BLOCK
(2)
(3) ;*****
(2) ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(3) ;*****
(2) .SX=. ;:SAVE CURRENT LOCATION
(2) 000024 .=24 ;:SET POWER FAIL TO POINT TO START OF PROGRAM
(2) 000020 200 ;:FOR APT START UP
(2) 000044 .=44 ;:POINT TO APT INDIRECT ADDRESS PNTR.
(2) 000044 $APTHDR ;:POINT TO APT HEADER BLOCK
(2) 002136 .=$X ;:RESET LOCATION COUNTER
(2) 002136
(3) ;*****
(2) ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(2) ;INTERFACE SPEC.
(2)
(2) $APTHD:
(2) 002136 000000 $HIBTS: .WORD 0 ;:TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(2) 002140 001526 $MBADR: .WORD $MAIL ;:ADDRESS OF APT MAILBOX (BITS 0-15)
(2) 002142 000010 $TSTM: .WORD 10 ;:RUN TIM OF LONGEST TEST
(2) 002144 000010 $PASTM: .WORD 10 ;:RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(2) 002146 000000 $UNITM: .WORD ;:ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(2) 002150 000052 .WORD $ETEND-$MAIL/2 ;:LENGTH MAILBOX-ETABLE(WORDS)

```


CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-17
 CNDUSA.M11 30-OCT-82 10:55 INITIALIZE THE COMMON TAGS

SEQ 0031

```

(1) 003166 012767 000300 013602 OUTMUL: MOV #300,DUPRT
(1) 003174 004767 013526 JSR PC,DULEV
(2) :COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
(2) :BUFFER TO THE CHARACTERS '1' AND '2'.
(2) :IF THE CHARACTER IS '1' CLEAR THE FLAG
(2) :IF THE CHARACTER IS '2' SET THE FLAG
(2) 003200 AAA:
(2) 003200 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(2) 003202 015575 MSYNC :MESSAGE
(2) 003204 122767 000061 012724 3$: CMPB #'1,INBUF :IS IT '1' ?
(2) 003212 001003 BNE 1$
(2) 003214 105067 175726 CLRB SYNCNO ;000
(2) 003220 000412 BR 4$
(2) 003222 122767 000062 012706 1$: CMPB #'2,INBUF :IS IT '2' ?
(2) 003230 001004 BNE 2$
(2) 003232 112767 177777 175706 MOVB #-1,SYNCNO ;377
(2) 003240 000402 BR 4$
(2) 003242 104407 2$: INSTER :RETRY
(2) 003244 000757 BR 3$
(2) 003246 000240 4$: NOP
(2) 003250 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(2) 003252 015643 MWIRE6 :MESSAGE
(2) 003254 104414 SETFLG :SET FLAG BASED UPON INPUT STRING
(2) 003256 001147 SEXMIT :THIS FLAG
(2) 003260 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(2) 003262 015714 MWIRE5 :MESSAGE
(2) 003264 104414 SETFLG :SET FLAG BASED UPON INPUT STRING
(2) 003266 001150 SEREC :THIS FLAG
(2) 003270 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(2) 003272 015764 MWIRE4 :MESSAGE
(2) 003274 104414 SETFLG :SET FLAG BASED UPON INPUT STRING
(2) 003276 001151 OPTCLR :THIS FLAG
(2) 003300 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(2) 003302 016043 MEXTJ :MESSAGE
(2) 003304 104414 SETFLG :SET FLAG BASED UPON INPUT STRING
(2) 003306 001153 JMRBY :THIS FLAG
(2)
(2) :TEST START AND RESTART
(2)
(2) 003310 012706 001100 .BEGIN: MOV #STACK,SP :SET UP STACK
(2) 003314 106427 000300 MTPS #300 :LOCK OUT INTERRUPTS
(2) 003320 032777 000002 176112 BIT #SW01,@SWR :IF SW01=1, GET STARTING PC
(2) 003326 001413 BEQ 3$
(3) 003330 104406 INSTR :OUTPUT MESSAGE & GET INPUT STRING
(3) 003332 015527 MTSTPC :MESSAGE
(3) 003334 104410 PARAM :CONVERT STRING
(3) 003336 003374 TST1 :LOW LIMIT
(3) 003340 017500 17500 :HIGH LIMIT
(3) 003342 001402 $TSTNM :STORE AT THIS LOCATION
(3) 003344 001 .BYTE 1 :MASK
(3) 003345 001 .BYTE 1 :HOW MANY TIMES + 2
(2) 003346 016767 176030 175530 MOV $TSTNM,RETURN
(2) 003354 000403 BR 4$
(2) 003356 012767 003374 175520 3$: MOV #TST1,RETURN :START AT TEST 1
(2) 003364 104401 015523 4$: TYPE ,MR :TYPE R
(2) 003370 000177 175510 JMP @RETURN :START TESTING

```

CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-18
 CNDUSA.M11 30-OCT-82 10:55 INITIALIZE THE COMMON TAGS

SEQ 0032

```

8151
8152      ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
(1)      ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
(1)      ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
(1)      ;; (OVRUN, RXERR)
(1)      ;; MODE: SYNEXT
(1)      ;; LENGTH: SEVEN
(1)      ;; CHAR: 177
(1)      ;;
(5)      *****
(4) 003374 000004      TST1: SCOPE
(4)
(3) 003376 052777 000400 176322      BIS      #MRESET, @TXCSR      ; MASTER RESET
(2) 003404 012777 020000 176310      MOV      #SYNEXT, @PARCSR      ; SET THE MODE
(3) 003412 052777 000400 176306      BIS      #MRESET, @TXCSR      ; MASTER RESET
(2)
(2)      ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
(2) 003420 012777 064001 176300      MOV      #MTDATA!CLK!MINT!BREAK, @TXCSR
(2)
(2)      ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
(2) 003426 012777 024000 176266      MOV      #SYNEXT!SEVEN!NOPAR!0, @PARCSR
(1) 003434 052777 000020 176250      BIS      #SYNSCH, @RXCSR      ; SET SEARCH SYNC
(2)      ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
(2) 003442 042777 020000 176256      BIC      #CLK, @TXCSR      ; POKE CLK DOWN
(2) 003450 052777 020000 176250      BIS      #CLK, @TXCSR      ; POKE CLK UP
(1) 003456 016703 176234      MOV      RXDBUF, R3      ; SET UP FOR ERROR MESSAGE
(1) 003462 012700 000177      MOV      #177, R0      ; EXPECTED
(1) 003466 012767 000007 175426      MOV      #7, SHIFT      ; # OF SHIFTS
(1) 003474 012767 000177 175776      MOV      #177, $TMP1      ; DATA CHAR
(1) 003502 004767 013430      JSR      PC, RPOKE      ; SHIFT IN THIS CHAR
(1) 003506 105777 176200      TSTB      @RXCSR      ; RXDONE ?
(1) 003512 100401      BMI      +4
(1) 003514 104004      ERROR      4      ; RXDONE SHOULD BE SET
(1) 003516 017701 176174      MOV      @RXDBUF, R1      ; ACTUAL
(1) 003522 020001      CMP      R0, R1      ; COMPARE EXPECTED VS. ACTUAL
(1) 003524 001401      BEQ      +4
(1) 003526 104002      ERROR      2      ; RECEIVED DATA DID NOT MATCH
(1)      ; EXPECTED DATA - CHECK MAINT DATA
(1)      ; OR RECEIVER LOGIC
(1) 003530 012767 000007 175364      MOV      #7, SHIFT      ; # OF SHIFTS
(1) 003536 012767 000177 175734      MOV      #177, $TMP1      ; DATA CHAR
(1) 003544 004767 013366      JSR      PC, RPOKE      ; SHIFT IN THIS CHAR
(1)      ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
(1) 003550 012767 000007 175344      MOV      #7, SHIFT      ; # OF SHIFTS
(1) 003556 012767 000177 175714      MOV      #177, $TMP1      ; DATA CHAR
(1) 003564 004767 013346      JSR      PC, RPOKE      ; SHIFT IN THIS CHAR
(1) 003570 012700 140177      MOV      #140000!177, R0      ; EXPECTED DATA PLUS
(1)      ; RXERR & OVRUN
(1) 003574 017701 176116      MOV      @RXDBUF, R1      ; ACTUAL
(1) 003600 020001      CMP      R0, R1      ; COMPARE EXP VS. ACT
(1) 003602 001401      BEQ      +4
(1) 003604 104002      ERROR      2      ; SPECIFICALLY LOOK AT RXERR &
(1)      ; OVRUN BITS...THEY BOTH SHOULD BE SET
8153
(1)      ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
      ;; RECEIVER SECTION, IT USES THE ERROR FLAGS

```


CNDUS-A MACY11 30(1046) 14-DEC-82 09:59 PAGE 61-20
 CNDUSA.M11 30-OCT-82 10:55 INITIALIZE THE COMMON TAGS

SEQ 0034

```

(1)          ;;LENGTH:EIGHT
(1)          ;;CHAR:125
(1)          ;;
(5)          *****
(4) 004020 000004 TST3: SCOPE
(4)
(3) 004022 052777 000400 175676 BIS #MRESET,@TXCSR ;MASTER RESET
(2) 004030 012777 020000 175664 MOV #SYNEXT,@PARCSR ;SET THE MODE
(3) 004036 052777 000400 175662 BIS #MRESET,@TXCSR ;MASTER RESET
(2)
(2) 004044 012777 064001 175654 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
(2) MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
(2)
(2) 004052 012777 026000 175642 ;SET MODE ,# OF BITS,PARITY SEN E,&LOAD SYNC REG
(1) 004060 052777 000020 175624 MOV #SYNEXT!EIGHT!NOPAR!0,@PARCSR
(2) BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
(2) ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
(2) 004066 042777 020000 175632 BIC #CLK,@TXCSR ;POKE CLK DOWN
(2) 004074 052777 020000 175624 BIS #CLK,@TXCSR ;POKE CLK UP
(1) 004102 016703 175610 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
(1) 004106 012700 000125 MOV #125,R0 ;EXPECTED
(1) 004112 012767 000010 175002 MOV #8,SHIFT ;# OF SHIFTS
(1) 004120 012767 000125 175352 MOV #125,$TMP1 ;DATA CHAR
(1) 004126 004767 013004 JSR PC,RPOKE ;SHIFT IN THIS CHAR
(1) 004132 105777 175554 TSTB @RXCSR ;RXDONE ?
(1) 004136 100401 BMI +4
(1) 004140 104004 ERROR 4 ;RXDONE SHOULD BE SET
(1) 004142 017701 175550 MOV @RXDBUF,R1 ;ACTUAL
(1) 004146 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
(1) 004150 001401 BEQ +4
(1) 004152 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
(1) ;EXPECTED DATA - CHECK MAINT DATA
(1) ;OR RECEIVER LOGIC
(1) 004154 012767 000010 174740 MOV #8,SHIFT ;# OF SHIFTS
(1) 004162 012767 000125 175310 MOV #125,$TMP1 ;DATA CHAR
(1) 004170 004767 012742 JSR PC,RPOKE ;SHIFT IN THIS CHAR
(1) ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
(1) 004174 012767 000010 174720 MOV #8,SHIFT ;# OF SHIFTS
(1) 004202 012767 000125 175270 MOV #125,$TMP1 ;DATA CHAR
(1) 004210 004767 012722 JSR PC,RPOKE ;SHIFT IN THIS CHAR
(1) 004214 012700 140125 MOV #140000!125,R0 ;EXPECTED DATA PLUS
(1) ;RXERR & OVRRUN
(1) 004220 017701 175472 MOV @RXDBUF,R1 ;ACTUAL
(1) 004224 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
(1) 004226 001401 BEQ +4
(1) 004230 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
(1) ;OVRRUN BITS...THEY BOTH SHOULD BE SET
(1)
8155 ;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
(1) ;RECEIVER SECTION,IT USES THE ERROR FLAGS
(1) ;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
(1) ;(OVRRUN,RXERR)
(1) ;MODE:SYNEXT
(1) ;LENGTH:EIGHT
(1) ;CHAR:252
(1) ;

```



```

(2) 012732 017767 166510 000422      MOV    @STKB,.MSG      ;GET CHARACTER
(2) 012740 042767 177600 000414      BIC    #177600,.MSG    ;STRIP JUNK
(2) 012746 122767 000007 000406      CMPB   #7,.MSG        ;IS IT <^G> ?
(2) 012754 001017                BNE     OUT            ;NO
(2) 012756 104401 016103                TYPE   ,MCNTG
(2) 012762 005137 013022      CNTLU:  COM    @#RDSW
(2) 012766 104401 016113                TYPE   ,MMSWR
(2) 012772 104413                CONVRT
(2) 012774 013024                SWREGL
(2) 012776 104406 016124                INSTR,MMNEW
(2) 013002 104410                PARAM
(2) 013004 000000                0
(2) 013006 177777                177777
(2) 013010 000176                SWREG
(2) 013012 000000 0001       .BYTE   0,1
(2) 013014 005037 013022      OUT:    CLR    @#RDSW
(2) 013020 000207                RTS     PC
(2) 013022 000000      RDSW:    .WORD   0
(2) 013024 000001      SWREGL: 1
(2) 013026 000006 002       .BYTE   6,2
(2) 013030 000176                SWREG
(2) 013032 000005                5
(2)
(2) ;CHECK FOR FREEZE ON CURRENT DATA
(2)
(2) 013034 004767 177646      .SCOP1: JSR    PC,CKSWR
(2) 013040 032777 001000 166372      BIT    #SW09,@SWR
(2) 013046 001402                BEQ     1$
(2) 013050 016716 166034      MOV    LOCK,(SP)
(2) 013054 000002      1$:      RTI
(2) .SBTTL TYPE ROUTINE
(2)
(2) ;*****
(2) ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(2) ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(2) ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(2) ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(2) ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(2) ;*
(2) ;*CALL:
(2) ;*1) USING A TRAP INSTRUCTION
(2) ;*      TYPE    ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(2) ;*OR
(2) ;*      TYPE
(2) ;*      MESADR
(2) ;*
(2)
(2) 013056 105767 166375      $TYPE:  TSTB   $TPFLG      ;;IS THERE A TERMINAL?
(2) 013062 100002                BPL     1$          ;;BR IF YES
(2) 013064 000000                HALT                ;;HALT HERE IF NO TERMINAL
(2) 013066 000430                BR      3$          ;;LEAVE
(2) 013070 010046      1$:      MOV    R0,-(SP)        ;;SAVE R0
(2) 013072 017600                MOV    @2(SP),R0     ;;GET ADDRESS OF ASCIZ STRING
(2) 013076 122767 000001 166442      CMPB   #APTENV,$ENV    ;;RUNNING IN APT MODE
(2) 013104 001011                BNE     62$          ;;NO,GO CHECK FOR APT CONSOLE

```



```

(2) 013566 001003
(2) 013570 005737 013022
(2) 013574 001023
(2) 013576 104407
(2) 013600 000742
(2)
(2)
(2)
(2) 013602 020567 000042
(2) 013606 101365
(2) 013610 020567 000032
(2) 013614 103762
(2) 013616 136705 000032
(2) 013622 001357
(2)
(2)
(2)
(2) 013624 016704 000022
(2) 013630 010524
(2) 013632 062705 000002
(2) 013636 105367 000013
(2) 013642 001372
(2) 013644 000002
(2) 013646 000000
(2) 013650 000000
(2) 013652 000000
(2) 013654 000000
(2) 013655
(2)
(2)
(2)
(2) 013656 016667 000004 165242 .SAV05: MOV 4(SP),SAVPC
(2)
(2)
(2)
(2)
(2) 013664 010567 165604
(2) 013670 010467 165576
(2) 013674 010367 165570
(2) 013700 010267 165562
(2) 013704 010167 165554
(2) 013710 010067 165546
(2) 013714 000002
(2)
(2)
(2)
(2)
(2) 013716 016700 165540
(2) 013722 016701 165536
(2) 013726 016702 165534
(2) 013732 016703 165532
(2) 013736 016704 165530
(2) 013742 016705 165526
(2) 013746 000002
(2)
(2)
(2)
(2) 013750 104401
  
```

```

BNE 120$
TST @#RDSW ;CK SWR USED
BNE PARTI
120$: INSTER ;RETRY
BR PARAM1

;TEST TO SEE IF NUMBER IS WITHIN LIMITS
LIMITS: CMP R5,HILIM
BHI PARERR ;THE # IS TOO HIGH
CMP R5,LOLIM
BLO PARERR ;THE # IS TOO LOW
BITB LOBITS,R5 ;TEST BY MASKINGTHE #
BNE PARERR

;STORE NUMBER AT SPECIFIED ADDRESS
1$: MOV DEVADR,R4 ;GET STARTING ADDR OF
MOV R5,(R4)+ ;STORE AT THIS ADDR
ADD #2,R5
DECB ADRCNT ;HOW MANY TIMES + 2 ?
BNE 1$
PARTI: RTI
LOLIM: 0
HILIM: 0
DEVADR: 0
LOBITS: 0
ADRCNT=LOBITS+1

;SAVE PC OF TEST THAT FAILED AND R0-R5
.SAV05: MOV 4(SP),SAVPC

;SAVE R0-R5
SV05: MOV R5,$REG5
MOV R4,$REG4
MOV R3,$REG3
MOV R2,$REG2
MOV R1,$REG1
MOV R0,$REG0
RTI

;RESTORE R0-R5
.RES05: MOV $REG0,R0
MOV $REG1,R1
MOV $REG2,R2
MOV $REG3,R3
MOV $REG4,R4
MOV $REG5,R5
RTI

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
.CONVR: TYPE
  
```


