

DU 11

OFF-LINE LOGIC TEST
CZDUAE0

AH-8677E-MC

COPYRIGHT 1973-78
FICHE 1 OF 1

JAN 1979
digital
MADE IN USA

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

PRODUCT CODE: AC-8676E-MC

PRODUCT NAME: CZDUAEO DU11 OFFLINE LOGIC TESTS

RELEASE DATE: JUN 1978

MAINTAINER : DIAGNOSTICS

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.

COPYRIGHT (C) 1973, 1978 BY DIGITAL EQUIPMENT CORPORATION

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DU11'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.

1. THE DU11 OFFLINE LOGIC TESTS VERIFY THAT ALL REGISTERS EXIST ,AND ALL RESPECTIVE BITS CAN BE MASTER CLEARED, READ, WRITTEN AND/OR READ/WRITTEN

2. REQUIREMENTS

PDP-11 FAMILY STANDARD COMPUTER WITH OR WITHOUT HARDWARE SWITCH REGISTER (LOC. 177570)

DU11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE

THE PROGRAM LOADS AND RUNS IN 8K OF MEMORY.

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

NOTE:

BEFORE PROCEEDING IT IS IMPORTANT TO
TO REALIZE IF ONE DOESNOT
HAVE THE DU11 SET UP TO THE
DEFAULT PARAMETERS (SEE SECTION

8 OF THIS DOCUMENT) , THEN ONE MUST
SET SW00 = 1, AND ANSWER THE PARAMETER
QUESTION ROUTINE.

4.1 CONTROL SWITCH SETTINGS

NOTE: SOFTWARE SWITCH REGISTER IS DEFINED AS LOC. 176, WHILE
THE SOFTWARE DISPLAY REGISTER IS DEFINED AS LOC. 174.

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)

SW02=1

NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=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 000200
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 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE 'DU11 CZDUA-E TAPE A' (ONCE ONLY)

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING

WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

4.3.1.7 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

4.3.3.1 LOAD ADDRESS 000200

4.3.3.2 SET SW00-1

4.3.3.3 PRESS START

NOTE:IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW- (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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 DU11 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 DU11 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 DU11 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 DU11 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 'DU PRIORITY LEVEL-' AND
WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.13 TYPE IN THE APPROPRIATE DEVICE PRIORITY LEVEL OF THE
DU11 OR DU11'S TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>
(NOTE THAT ALL MULTIPLE DEVICES MUST BE AT THE SAME PRIORITY
LEVEL). IE '5'

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

4.3.3.14 THE PROGRAM WILL TYPE '# OF SYNC CHARS
SELECTED (1 OR 2)-' AND WAIT FOR AN INPUT FROM THE TELETYPE
KEYBOARD

4.3.3.15 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 4.3.3.14

4.3.3.16 THE PROGRAM WILL TYPE '' IS SEC XMIT JUMPER #6 IN ? (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 THAT ALL MULTIPLE DEVICES
MUST BE THE SAME)

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

4.3.3.18 THE PROGRAM WILL TYPE ''IS SEC REC JUMPER # 5 IN ?
(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
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 ''IS OPT CLR ENABLE JUMPER
4 IN ? (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 ''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

4.3.3.23 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 4.3.3.22

4.3.3.24 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 LOAD 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX NEW= (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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 SW02 -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 LOAD ADDRESS 000200

4.3.5.2 SET SW02 -1
NOTE: IT MAY BE ADVANTAGEOUS TO SET SW01 1 (OPTIONAL)

4.3.5.3 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR-XXXXXX NEW- (REFER TO SECTION 5. FOR OPERATOR'S OPTION)

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

4.4 STATUS MAP

THE STATUS MAP IS AN AREA OF THE DU11 DIAGNOSTICS, WHICH WILL ALLOW THE TRANSFER OF PARAMETERS BETWEEN DIAGNOSTICS. IF YOU WISH TO TEST A DU11, WHICH IS NOT AT THE DEFAULT VALUES, YOU NEED ONLY GO THROUGH THE TEDIOUS QUESTIONING AND ANSWERING ROUTINE ONCE.

THE FOLLOWING COMBINATIONS OF SWITCH REGISTER SETTINGS WILL ALLOW YOU ACCESS TO THE STATUS MAP.

- 1) SW07=1
- 2) START AT 200
- 3) THE DIAGNOSTIC WILL GO TO THE STATUS MAP AND BYPASS ALL OF THE QUESTIONING ROUTINE.

NOTE: IT IS EXTREMELY IMPORTANT THAT EITHER YOU HAVE JUST ANSWERED THESE QUESTIONS DURING A PRIOR DIAGNOSTIC OR THAT YOU HAVE MANUALLY ENTERED THE CORRECT VALUES FOR VECTOR ADDRESSES ETC., IN THE AREA DESIGNATED FOR THE STATUS MAP. IT IS IMPORTANT THAT THIS BE PERFORMED BEFORE STARTING AT 200.

THE DIAGNOSTIC HAS NO METHOD TO DETERMINE THAT THE STATUS MAP HAS INDEED BEEN LOADED CORRECTLY. THE DIAGNOSTIC ASSUMES THAT WHEN SW07=1 THE VALUES IN THE STATUS MAP ARE THE VALUES TO BE USED. THESE VALUES CAN BE THE WRONG VALUES, BUT THE DIAGNOSTIC WILL NOT REALIZE THAT A MISTAKE HAS BEEN MADE.

IF BOTH SW07 AND SW00 (SWITCH REGISTER SWITCHES) ARE SET (EQUAL TO 1), THE PROGRAM WILL IGNORE SW00 AND SEEING SW07 SET, THE VALUES FROM THE STATUS MAP WILL BE USED. TO USE THE DEFAULT VALUES FOR THE DU11'S THE OPERATOR MUST SET SW00=0 AND SW07=0. THE USE OF SW00 IS EXPLAINED IN GREATER DETAIL IN SECTION 4.3 OF THIS DOCUMENT.

THE FIRST TIME A PROGRAM IS LOADED OR THE FIRST TIME A PROGRAM IS ALTERED VIA THE PARAMETER RESELECTION QUESTION AND ANSWER ROUTINE, A PARTIAL STATUS MAP WILL BE PRINTED. THIS MAP WILL BE PRINTED ONCE FOR ANY COMBINATION OF SWITCHES EXCEPT SW01. RESTARTING THE PROGRAM WILL NOT PRINT OUT A MAP UNLESS THE PROGRAM PARAMETERS ARE BEING RESELECTED BY PUTTING SW00-1.(ON)

THE MAP WILL LOOK LIKE:

STATUS MAP

1300/ 177777

1302/ 000000

1304/ 177777

THE BYTES ARE DEFINED AS FOLLOWS:

1300 THE NUMBER OF SYNCHRONOUS CHARACTERS REQUIRED FOR
SYNCHRONIZATION.
1301 SEC TRANSMIT JUMPER
1302 SEC RECEIVER JUMPER
1303 OPTIONAL JUMPER
1304 MULTIPLE DEVICES (NO=0 , YES= 1)
1305 EXTERNAL MODEM BYPASS? (NO=0 ,YES= 1)

IF THE BYTE IS 0 , THE JUMPER IS NOT CONNECTED
AND IF THE BYTE IS 377 ETC. THE JUMPER SHOULD BE CONNECTED.

5. OPERATING PROCEDURE

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
DOING THE FOLLOWING:

- 1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO
LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR-XXXXXXNEW- (XXXXXX IS THE OCTAL CONTENTS
OF THE SOFTWARE SWITCH REGISTER.)
- 3) AFTER THE ''NEW-'' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
OF THE FOLLOWING AT THE TTY:
 - A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>.
(ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.
 - B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
BACK TO STEP 2.

5.1

OPERATIONAL SWITCH SETTINGS

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
SW08 =1 LOOP ON ERROR
SW07 =1 USE STATUS MAP PARAMETERS
SW02 =1 LOCK ON TEST
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
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

NOTE: IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT
THE THE OPERATOR IS REQUIRED TO TYPE A <^G> BEFORE DEPRESSING CONTINUE.
THE FOLLOWING WILL BE TYPED:
SWR-XXXXXX NEW- (REFER TO SECTION 5. FOR OPERATOR OPTION)

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...LOAD 000200 IN SWR AND DEPRESS START....
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.
 - 7.2.2.2ORLOAD 000200 WITH SW00 =1 AND DEPRESS START....
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.....LOAD & START AT 000200
- 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 11/20 MACHINE.
IF RUNNING ON AN 11/40 OR AN 11/45 'HOLD:' MUST BE PATCHED TO 40

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'
CODE MUST BE PATCHED TO A 'NOP'. (SEE LISTINGS FOR DETAILS)
- 8. DEFAULT PARAMETERS:
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160040

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

DU PRIORITY LEVEL- LEVEL 5 DUPRT: LEVEL 5

OF SYNC CHARS SELECTED - 2 SYNCNO: 377

IS SEC XMIT JUMPER # 6 IN ?- YES SEXMIT: 377

IS SEC REC JUMPER # 5 IN ?- YES SREC: 377

IS OPT CLR ENABLE JUMPER # 4 IN ?- 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 LOGIC BIT BANGING
OF THE DEVICE
SEE LISTING FOR DETAILS

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

11. LISTINGS

LZDUA-E MACY11 30A(1052) 21-NOV-78 15:18 PAGE 15
EZDU22.PAR 21-NOV-78 15:17

B 2

SEQ 0014

674
675 000000' 000000G

D

```
676 .ENABLE ABS
677
678 ;DU11 CZDUA-E TAPE A
679 ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
680
681 ;STARTING PROCEDURE
682 ;LOAD PROGRAM
683 ;PRESS START
684 ;PROGRAM WILL TYPE 'DU11 CZDUA-E TAPE A '
685 ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
686 ;AT THE END OF A PASS, PROGRAM WILL TYPE 'END OF PASS TAPE A'
687 ;AND THEN RESUME TESTING
688
689
690 ;SWITCH REGISTER OPTIONS
691
692 100000 SW15=100000 ;=1,HALT ON ERROR
693 040000 SW14=40000 ;=1,LOOP ON CURRENT TEST
694 020000 SW13=20000 ;=1,INHIBIT ERROR TYPEOUT
695 010000 SW12=10000
696 004000 SW11=4000 ;=1,INHIBIT ITERATIONS
697 002000 SW10=2000 ;=1,ESCAPE TO NEXT TEST ON ERROR
698 001000 SW09=1000 ;=1,LOOP WITH CURRENT DATA
699 000400 SW08=400 ;=1,LOOP ON ERROR
700 000200 SW07=200 ;+ =1, USE STATUS MAP
701 000100 SW06=100
702 000040 SW05=40
703 000020 SW04=20
704 000010 SW03=10
705 000004 SW02=4
706 000002 SW01=2
707 000001 SW00=1
708
709 ;LOCK ON TEST SELECT
;RESTART PROGRAM AT SELECTED TEST
;RESELECT VECTOR AND CONTROL REGISTER
;ADDRESS AFTER PROGRAM RESTART
```

710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765

000000
000001
000002
000003
000004
000005
000006
000007

177570
177570
177776
001100

005746
005726
010046
012600
024646
022626

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

000340
000300
000240
000200
000140
000100
000040
000000

;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
 SP=%6 ;PROCESSOR STACK POINTER
 PC=%7 ;PROGRAM COUNTER

;LOCATION EQUIVALENCIES

DSWR=177570 ;HARDWARE SWITCH REGISTER LOC.
 DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
 PS=177776 ;PROCESSOR STATUS WORD
 STACK=1100 ;START OF PROCESSOR STACK

;INSTRUCTION DEFINITIONS

PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
 POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD -TST (SP)+
 PUSHRO=10046 ;SAVE R0 ON STACK =MOV R0,-(SP)
 POPRO=12600 ;RESTORE R0 FROM STACK =MOV (SP)+,R0
 PUSH2SP=24646 ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
 POP2SP=22626 ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
 .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL

BIT15=100000
 BIT14=40000
 BIT13=20000
 BIT12=10000
 BIT11=4000
 BIT10=2000
 BIT9=1000
 BIT8=400
 BIT7=200
 BIT6=100
 BIT5=40
 BIT4=20
 BIT3=10
 BIT2=4
 BIT1=2
 BIT0=1

;PROCESSOR LEVELS

LEVEL7=340
 LEVEL6=300
 LEVEL5=240
 LEVEL4=200
 LEVEL3=140
 LEVEL2=100
 LEVEL1=040
 LEVEL0=000

```

766      ;REGISTER DEFINITIONS
767      ;RXCSR BIT DEFINITIONS
768      100000      DSC=BIT15      ;DATA SET CHANGE
769      040000      RING=BIT14     ;RING
770      020000      CTS=BIT13      ;CLR TO SEND
771      010000      CARDET=BIT12   ;CARRIER DETECT
772      004000      RECACT=BIT11   ;REC ACTIVE
773      002000      SRD=BIT10      ;SEC REC DATA
774      001000      DSR=BIT9       ;DATA SET RDY
775      000400      STPSYN=BIT8    ;STRIP SYNC
776      000200      RXDONE=BIT7    ;REC DONE
777      000100      RINTEN=BIT6    ;REC INTR ENABLE
778      000040      DSINTE=BIT5    ;DSC INTR ENABLE
779      000020      SYN SCH=BIT4    ;SYNC SEARCH
780      000010      STD=BIT3       ;SEC XMIT DATA
781      000004      RTS=BIT2       ;REQ TO SEND
782      000002      DTR=BIT1       ;DATA TERM RDY
783      000001      VOID=BIT0
784      ;RXDBUF BIT DEFINITIONS
785      100000      RXERR=BIT15     ;REC ERROR
786      040000      OVR RUN=BIT14   ;OVERRUN
787      020000      FRMERR=BIT13    ;FRAME ERROR
788      010000      PARER=BIT12     ;PARITY ERROR
789      ;PARCSR BIT DEFINITIONS
790      001000      PAREN=BIT9       ;PARITY ENABLE
791      000400      EVPAR=BIT8      ;EVEN PARITY SENSE
792      ;PARCSR WRD DEFINITIONS
793      030000      SYNINT=30000     ;SYNC EXTERNAL MODE
794      020000      SYNEXT=20000    ;SYNC INTERNAL MODE
795      000000      ISYMOD=0         ;ISOC MODE
796      000000      FIVE=0          ;WORD LENGTH 5 BITS
797      002000      SIX=2000        ;WORD LENGTH 6 BITS
798      004000      SEVEN=4000      ;WORD LENGTH 7 BITS
799      006000      EIGHT=6000     ;WORD LENGTH 8 BITS
800      000000      NOPAR=0         ;NO PARITY
801      001000      ODDPAR=1000     ;ODD PARITY
802      001400      EVEPAR=1400     ;EVEN PARITY
803      ;TXCSR BIT DEFINITIONS
804      100000      DNA=BIT15        ;DATA NOT AVAILABLE
805      040000      MTDATA=BIT14    ;MAINT DATA
806      020000      CLK=BIT13       ;CLK
807      002000      BITW=BIT10      ;BIT WINDOW
808      000400      MRESET=BIT8     ;MASTER RESET
809      000200      TXDONE=BIT7     ;XMIT DONE
810      000100      TXINTE=BIT6     ;XMIT INTR ENABLE
811      000040      DNAINTE=BIT5    ;DNA INTR ENAB
812      000020      SEND=BIT4       ;SEND
813      000010      HDXEN=BIT3      ;HDX/FDX
814      000001      BREAK=BIT0      ;BREAK
815      ;TXCSR WRD DEFINITIONS
816      000000      USER=0           ;USER MODE
817      004000      MINT=4000        ;MAINT INT MODE
818      010000      MEXT=10000       ;MAINT EXT MODE
819      014000      SYSTST=14000     ;SYSTEM TEST MODE
820      ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
  
```

```

821                                     ;STANDARD INTERRUPT VECTORS
822
823
824      000024      000024      .-24
825      000024      016250      .PFail      ;POWER FAIL HANDLER
826      000026      000340      340      ;SERVICE AT LEVEL 7
827      000030      016000      .HLT      ;ERROR HANDLER
828      000032      000340      340      ;SERVICE AT LEVEL 7
829      000034      015746      .TRPSRV      ;GENERAL HANDLER DISPATCH SERVICE
830      000036      000340      340      ;SERVICE AT LEVEL 7
831
832      ;SOFTWARE SWITCH REGISTER
833
834      000174      000174      .=174
835      000174      000000      DISPREG: .WORD 0      ;SOFTWARE DISPLAY REG.
836      000176      000000      SWREG: .WORD 0      ;SOFTWARE SWITCH REGISTER
837      000200      000167      001214      JMP .START      ;GO TO START OF PROGRAM
838
839
840
841      001100      .=1100
842
843      ;INDIRECT POINTERS
844
845      001100      177570      SWR: 177570      ;SWITCH REGISTER POINTER
846      001102      177570      LIGHTS: 177570      ;DISPLAY REGISTER POINTER
847      001104      177560      TKCSR: 177560      ;TELETYPE KEYBOARD CONTROL REGISTER
848      001106      177562      TKDBR: 177562      ;TELETYPE KEYBOARD DATA BUFFER
849      001110      177564      TPCSR: 177564      ;TELEPRINTER CONTROL REGISTER
850      001112      177566      TPDBR: 177566      ;TELEPRINTER DATA BUFFER
851
852      ;PROGRAM CONTROL PARAMETERS
853
854      001114      000000      RTRN: 0      ;SCOPE ADDRESS FOR LOOP ON TEST
855      001116      000000      NEXT: 0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
856      001120      000000      LOCK: 0      ;ADDRESS FOR LOCK ON CURRENT DATA
857      001122      000000      ICOUNT: 0      ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
858      001124      000000      LPCNT: 0      ;NUMBER OF ITERATIONS COMPLETED
859      001126      000000      TSTNO: 0      ;NUMBER OF TEST IN PROGRESS
860      001130      000000      PASCNT: 0      ;NUMBER OF PASSES COMPLETED
861      001132      000000      ERRCNT: 0      ;TOTAL NUMBER OF ERRORS
862      001134      000000      LSTERR: 0      ;PC OF LAST ERROR CALL
863
864      ;PROGRAM VARIABLES
865
866      001136      000020      HOLD: 20      ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
867      001140      000000      SHIFT: 0      ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
868      001142      000000      COUNT: 0      ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
869      001144      000000      TEMP1: 0      ;TEMPORARY STORAGE
870      001146      000000      TEMP2: 0      ;TEMPORARY STORAGE
871      001150      000000      TEMP3: 0      ;TEMPORARY STORAGE
872      001152      000000      TEMP4: 0      ;TEMPORARY STORAGE
873      001154      000000      TEMP5: 0      ;TEMPORARY STORAGE
874      001156      000000      SAVR0: 0      ;R0 STORAGE
875      001160      000000      SAVR1: 0      ;R1 STORAGE
876      001162      000000      SAVR2: 0      ;R2 STORAGE

```

877 001164 000000
878 001166 000000
879 001170 000000
880 001172 000000
881 001174 000000

SAVR3: 0
SAVR4: 0
SAVR5: 0
SAVSP: 0
SAVPC: 0

;R3 STORAGE
;R4 STORAGE
;R5 STORAGE
;STACK POINTER STORAGE
;PROGRAM COUNTER STORAGE

```

882      ;PROGRAM CONVERSATIONAL PARAMETERS
883 001176 377 SYNCNO: .BYTE 377 ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
884 001177 377 SEXMIT: .BYTE 377 ;SEC XMIT JUMPER 'IN'
885 001200 377 SEREC: .BYTE 377 ;SEC REC JUMPER 'IN'
886 001201 377 OPTCLR: .BYTE 377 ;OPTIONAL JUMPER CLR 'IN'
887 001202 000 MULTD: .BYTE 0 ;NO MULTIPLE DEVICE FLAG
888 001203 377 JMRBY: .BYTE 377 ;EXTERNAL MODEM BYPASS JUMPER 'IN'
889 .EVEN
890
891      ;PROGRAM MULTIPLE DEVICE PARAMETERS
892 001204 000000 BASEADD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
893 001206 000000 KEEPADD: 0 ;SAVED 1ST DEVICE ADDR
894 001210 000000 LASTADD: 0 ;LAST DEVICE RXCSR ADDR
895 001212 000000 BASEIV: 0 ;PROG CONTROLLED IV
896 001214 000000 KEEPIV: 0 ;SAVED INTR VECTOR
897 001216 000000 ACTREG: 0 ;ACTIVE REGISTER , , ,MODIFY THIS
898 ;LOCATION TO DISQUALIFY OR QUALIFY
899 ;DEVICES (1= RUN , , 0= DON'T RUN)
900 001220 000000 ROTADD: 0 ;ROTATING POINTER FOR ACTREG..POINTS
901 ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE
902 ;*****
903
904 ; THESE ARE STORAGE FOR THE STATUS MAP PRINT OUT
905 001222 000000 FLAG:0 ; FLAGS FOR STATUS MAP PRINT OUT (SSP)
906 001224 000000 HOLD0:0 ; HOLDS R0 IN STATUS MAP PRINT
907 001226 000000 HOLD1:0 ; R1 ETC.
908 001230 000000 COUNT1:0 ; FOR COUNTING 3 WORDS
909 001232 000002 TABLE : 2 ; FOR CONVRT ROUTINE
910 001234 003006 0 3006
911 001236 000000 0 3006
912 001240 003006 0
913 001242 000000 0
914 ;*****
915 ;PROGRAM CONTROL FLAGS
916
917
918 001244 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
919 001245 000 STFLG: .BYTE 0 ;TEST START FLAG
920 001246 000 ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
921 001247 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
922 .EVEN
923
924
925 ;***** STATUS MAP *****
926
927 .-1300
928 001300 000001 STATUS: NOSYNC: .BLKB 1 ;SYNC CHARS
929 001301 000001 MITSEX: .BLKB 1 ;XMIT JUMPER
930 001302 000001 RESEC: .BLKB 1 ;REC SEC JUMPER
931 001303 000001 CLROPT: .BLKB 1 ;OPTIONAL JUMPER
932 001304 000001 DMULT: .BLKB 1 ;MULTIPLE DEVICE FLAG
933 001305 000001 BYJMR: .BLKB 1 ;EXTERNAL MODEM
934 .
935 ; MULTIPLE DEVICE PARAMETERS
936
937 001306 000001 ADDRBASE: .BLKW 1 ;PROG CONTROLLED 1ST DEVICE ADDR

```

938	001310	000001	ADDKEEP: .BLKW 1	; SAVED 1ST DEVICE ADDR
939	001312	000001	ADDLAST: .BLKW 1	; LAST DEVICE RXCSR ADDR
940	001314	000001	IVBASE: .BLKW 1	; PROG CONTROLLED IV
941	001316	000001	IVKEEP: .BLKW 1	; SAVED INTR VECTOR
942	001320	000001	REGACT: .BLKW 1	; ACTIVE REGISTER
943	001322	000001	ADDROT: .BLKW 1	; ROTATING POINTER
944	001324	000001	PRTDU: .BLKW 1	; DU11 PRIORITY
945	001326	000001	RIVDU: .BLKW 1	; DU11 REC INTR VECTOR
946	001330	000001	TIVDU: .BLKW 1	; DU11 XMIT INTR VECTOR
947	001332	000001	TISDU: .BLKW 1	; DU11 XMIT INTR STATUS
948	001334	000001	RISDU: .BLKW 1	; DU11 REC INTR STATUS
949	001336	000001	L1ESS: .BLKW 1	; PRIORITY TO ALLOW INTR
950	001340	000001	CSRRX: .BLKW 1	; DEFAULT OR ALTERED PARAMETERS
951	001342	000001	CSRHRX: .BLKW 1	;
952	001344	000001	BUFRXD: .BLKW 1	
953	001346	000001	BUFHRXD: .BLKW 1	
954	001350	000001	CSRPAR: .BLKW 1	
955	001352	000001	CSRHPAR: .BLKW 1	
956	001354	000001	CSRTX: .BLKW 1	
957	001356	000001	CSRHTX: .BLKW 1	
958	001360	000001	BUFTXD: .BLKW 1	
959	001362	000001	BUFHTXD: .BLKW 1	
960	001364	000001	BASEDU: .BLKW 1	; DU11 RXCSR BASE ADDR
961			.EVEN	

;DEFINITIONS FOR TRAP SUBROUTINE CALLS
 ;POINTERS TO SUBROUTINES CAN BE FOUND
 ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS

967	001366		.TRPTAB:	
968			*****	
969			*****	
970		104400	SCOPE TRAP+0	; CALL TO SCOPE LOOP AND ITERATION HANDLER
971	001366	014532	.SCOPE	
972		104401	SCOPI TRAP+1	; CALL TO LOOP ON CURRENT DATA HANDLER
973	001370	014716	.SCOP1	
974		104402	TYPE-TRAP+2	; CALL TO TELETYPE OUTPUT ROUTINE
975	001372	014736	.TYPE	
976		104403	INSTR-TRAP+3	; CALL TO ASCII STRING INPUT ROUTINE
977	001374	014776	.INSTR	
978		104404	INSTER=TRAP+4	; CALL TO INPUT ERROR HANDLER
979	001376	015114	.INSTER	
980		104405	PARAM=TRAP+5	; CALL TO NUMERICAL DATA INPUT ROUTINE
981	001400	015146	.PARAM	
982		104406	SAV05-TRAP+6	; CALL TO REGISTER SAVE ROUTINE
983	001402	015362	.SAV05	
984		104407	RES05-TRAP+7	; CALL TO REGISTER RESTORE ROUTINE
985	001404	015422	.RES05	
986		104410	CONVRT-TRAP+10	; CALL TO DATA OUTPUT ROUTINE
987	001406	015454	.CONVRT	
988		104411	CNVRT=TRAP+11	; CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
989	001410	015460	.CNVRT	
990		104412	SETFLG TRAP+12	; CALL TO FLAG SET ROUTINE
991	001412	015700	.SETFLG	
992		104413	CKSWR-TRAP+13	; CALL TO ALLOW SWREG TO BE LOADED FROM TTY
993	001414	016414	.CKSWR	

```

994      104414      CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
995 001416 016470 .CNTLU
996      ;*****
997      ;*****
998
999      ;PROGRAM INITIALIZATION
1000     ;LOCK OUT INTERRUPTS
1001     ;SET UP PROCESSOR STACK
1002     ;SET UP POWER FAIL VECTOR
1003     ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1004     ;TYPE TITLE MESSAGE
1005
1006 001420 012767 000340 176350 .START: MOV #340,PS ;LOCK OUT INTERRUPTS'
1007 001426 012706 001100 MOV #STACK,SP ;SET UP STACK
1008 001432 012737 016250 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
1009 001440 005067 177460 CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
1010 001444 105067 177575 CLRB STFLG ;CLEAR START FLAG
1011 001450 005067 177454 CLR PASCNT ;CLEAR PASS COUNT
1012 001454 105067 177566 CLRB ERRFLG ;CLEAR ERROR FLAG
1013 001460 005067 177446 CLR ERRCNT ;CLEAR ERROR COUNT
1014 001464 005067 177444 CLR LSTERR ;CLEAR LAST ERROR POINTER
1015 001470 012767 000001 177430 MOV #1,TSTNO ;SET UP FOR TEST 1
1016 001476 012767 001420 177410 MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
1017                                     ;TESTING STARTS
1018 001504 105767 177534 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1019 001510 001004 BNE ONCE
1020 001512 104402 016570 TYPE ,MTITLE ;TYPE TITLE MESSAGE
1021 001516 105167 177522 COMB INIFLG ;IF NOT SET FLAG AND DO
1022 001522 012767 177570 177350 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1023 001530 012767 177570 177344 MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1024 001536 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
1025 001542 013746 000004 MOV @#4,-(SP)
1026 001546 012737 001566 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
1027 001554 022777 177777 177316 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1028 001562 001402 BEQ 65$
1029 001564 000407 BR 66$
1030 001566 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
1031 001570 012767 000176 177302 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1032 001576 012767 000174 177276 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
1033 001604 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
1034 001610 012637 000006 MOV (SP)+,@#6
1035 001614 005737 000042 TST @#42 ;UNDER MONITOR
1036 001620 001005 BNE MAP
1037 001622 022767 000176 177250 CMP #SWREG,SWR ;IS SWREG USED
1038 001630 001001 BNE MAP ; BRANCH TO CHECK FOR STATUS MAP
1039 001632 104414 CNTLU
1040      ;*****
1041      ; CODE FOR STATUS MAP
1042      ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1043      ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1044      ; SETUP, OR REENTER QUESTIONING ROUTINE
1045
1046 001634 032777 000200 177236 MAP: BIT #SW07, @SWR ; IS SW07=1?
1047 001642 001537 BEQ $67 ; IF NOT, GO TO TEST FOR SW00=1
1048      ; NOW SET JP MAP VALUES FOR PROGRAM
1049      ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE

```

```

1050      ; OPERATION OF THIS PROGRAM.
1051 001644 116767 177430 177324  MOVB NOSYNC ,SYNCRNO      ; SYNC CHAR
1052 001652 116767 177423 177317  MOVB MITSEX  ,SEXMIT      ; XMIT JUMPER
1053 001660 116767 177416 177312  MOVB RESEC   ,SEREC      ; SEC REC JUMPER
1054 001666 116767 177411 177305  MOVB CLROPT ,OPTCLR     ; OPTIONAL JUMPER
1055 001674 116767 177404 177300  MOVB DMULT  ,MULTD     ; MULTIPLE DEVICE
1056 001702 116767 177377 177273  MOVB BYJMR  ,JMRBY     ; EXTERNAL MODEM
1057 001710 016767 177372 177266  MOV ADDBASE ,BASEADD    ; PROG 1ST DEVICE ADDR
1058 001716 016767 177366 177262  MOV ADDKEEP ,KEEPADD    ; SAVED 1ST DEVICE ADDR
1059 001724 016767 177362 177256  MOV ADDLAST ,LASTADD    ; LAST DEVICE RXCSR ADDR
1060 001732 016767 177360 177254  MOV IVKEEP  ,KEEPIV     ; SAVED INTR VECTOR
1061 001740 016767 177354 177250  MOV REGACT ,ACTREG    ; ACTIVE REGISTER
1062 001746 016767 177350 177244  MOV ADDROT ,ROTADD    ; ROTATING POINTER
1063 001754 016767 177334 177230  MOV IVBASE ,BASEIV     ; BASE INTR VECTOR
1064 001762 016767 177220 177214  MOV KEEPADD ,BASEADD    ; RELOAD BASEADD
1065 001770 016767 177332 016736  MOV RIVDU  ,DURIV     ; REC INTR VECTOR
1066 001776 016767 177332 016732  MOV RISDJ ,DURIS     ; REC INTR STATUS
1067 002004 016767 177320 016726  MOV TIVDU  ,DUTIV     ; XMIT INTR VECTOR
1068 002012 016767 177314 016722  MOV TISDU  ,DUTIS     ; XMIT INTR STATUS
1069 002020 016767 177312 016230  MOV LESS  ,LESS1     ; PRIORITY TO ALLOW INTR
1070 002026 013737 001324 020254  MOV @#PRTDU, @#DUPRT  ; PRIORITY RELOADED
1071 002034 016767 177324 016350  MOV BASEDU ,DUBASE
1072 002042 016767 177272 016640  MOV CSRRX  ,RXCSR
1073 002050 016767 177266 016634  MOV CSRHRX ,HRXCSR
1074 002056 016767 177262 016630  MOV BUFRXD ,RXDBUF
1075 002064 016767 177256 016624  MOV BUFRXD ,RXDBUF
1076 002072 016767 177252 016620  MOV CSRRXD ,HRXDBUF
1077 002100 016767 177246 016614  MOV CSRRPAR,PARCSR
1078 002106 016767 177242 016610  MOV CSRRPAR,PARCSR
1079 002114 016767 177236 016604  MOV CSRTX  ,TXCSR
1080 002122 016767 177232 016600  MOV CSRTX  ,TXCSR
1081 002130 016767 177226 016574  MOV BUFTXD ,TXDBUF
1082 002136 000167 000466  MOV BUFTXD ,HTXDBUF
1083      JMP .BEGIN ;BRANCH TO BEGIN TESTING
1084 002142 032777 000001 176730 $67: BIT #SW00 ,@SWR ; RESELECT VECTOR $ CONTROL REG?
1085 002150 001002      BNE 1$ ; BRANCH TO QUESTIONING
1086 002152 000167 000452      JMP .BEGIN ; GO TO LOAD STATUS MAP ETC.
1087 002156 005037 001222 1$: CLR @#FLAG ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1088 002162 012700 000300      MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1089 002166 012701 000302      MOV #302,R1 ;START AT LOCATION 300
1090 002172 012702 000004      MOV #4,R2
1091 002176 010110 2$: MOV R1,(R0)
1092 002200 005011      CLR (R1)
1093 002202 060200      ADD R2,R0
1094 002204 060201      ADD R2,R1
1095 002206 022701 001000      CMP #1000,R1
1096 002212 002771      BLT 2$ ;END AT LOCATION 776
1097 002214 104403      INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1098 002216 016644      MREGAD ;MESSAGE
1099 002220 104405      PARAM ;CONVERT STRING
1100 002222 160000      160000 ;LOW LIMIT
1101 002224 167776      167776 ;HIGH LIMIT
1102 002226 020412      DUBASE ;STORE AT THIS LOCATION
1103 002230 001 .BYTE 1 ;MASK
1104 002231 001 .BYTE 1 ;HOW MANY TIMES + 2
1105 002232 016767 016154 176746 MOV DUBASE,KEEPADD ;SAVE

```

1106	002240	004767	016014		JSR	PC,DUADDR	
1107	002244	016767	176736	176732	MOV	KEEPADD,BASEADD	;RESTORE FOR ROTATION
1108	002252	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1109	002254	016622			MVECTO		;MESSAGE
1110	002256	104405			PARAM		;CONVERT STRING
1111	002260	000300			300		;LOW LIMIT
1112	002262	000776			776		;HIGH LIMIT
1113	002264	020734			DURIV		;STORE AT THIS LOCATION
1114	002266	001			.BYTE 1		;MASK
1115	002267	004			.BYTE 4		;HOW MANY TIMES + 2
1116	002270	016767	016440	176716	MOV	DURIV,KEEPIV	;SAVE
1117	002276	016767	016432	176706	MOV	DURIV,BASEIV	;SET UP FOR ROTATION
1118	002304	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1119	002306	016725			MMULT		;MESSAGE
1120	002310	104412			SETFLG		;SET FLAG BASED UPON INPUT STRING
1121	002312	001202			MULTD		;THIS FLAG
1122	002314	105767	176662		TSTB	MULTD	;ARE THERE MULTIPLE DEVICES
1123							;ON THE SYSTEM ?
1124	002320	100406			BMI	BBB	;YES,ASK NEXT QUESTION
1125	002322	005067	176670		CLR	ACTREG	
1126	002326	005067	176666		CLR	ROTADD	
1127	002332	000167	000140		JMP	OUTMUL	;JUMP AROUND NEXT QUESTION
1128	002336				BBB:		
1129	002336	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1130	002340	017004			MLASTD		;MESSAGE
1131	002342	104405			PARAM		;CONVERT STRING
1132	002344	160000			160000		;LOW LIMIT
1133	002346	167776			167776		;HIGH LIMIT
1134	002350	001210			LASTADD		;STORE AT THIS LOCATION
1135	002352	001			.BYTE 1		;MASK
1136	002353	001			.BYTE 1		;HOW MANY TIMES + 2
1137							;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1138	002354	012767	000001	176636	1\$: MOV	#1,ROTADD	;SET UP POINTER
1139	002362	005067	176630		CLR	ACTREG	;CLR ACTIVE REGISTER
1140	002366	056767	176626	176622	2\$: BIS	ROTADD,ACTREG	;MAKE THIS DEVICE ACTIVE
1141	002374	000241			CLC		
1142	002376	006167	176616		ROL	ROTADD	;SET UP POINTER
1143	002402	103421			BCS	3\$	;ARE YOU OUT OF RANGE ?
1144	002404	062767	000010	176572	ADD	#10,BASEADD	;SET UP BASE ADDRESS
1145	002412	026767	176572	176564	CMP	LASTADD,BASEADD	;IS THIS THE LAST DEVICE ?
1146	002420	101362			BHI	2\$	;NO DO IT AGAIN
1147	002422	056767	176572	176566	BIS	ROTADD,ACTREG	;THIS ASSUMES THAT THERE ARE AT
1148							;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1149							;MULTIPLE DEVICE QUESTION
1150	002430	012767	000001	176562	4\$: MOV	#1,ROTADD	;SET UP FOR LATER USE IN END OF PASS ROUTINE
1151	002436	016767	176544	176540	MOV	KEEPADD,BASEADD	;DITTO
1152	002444	000414			BR	OUTMUL	;CONTINUE QUESTIONS
1153	002446	016767	176534	176530	3\$: MOV	KEEPADD,BASEADD	;RESTORE
1154	002454	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1155	002456	017167			MRANGE		;MESSAGE
1156	002460	104405			PARAM		;CONVERT STRING
1157	002462	160000			160000		;LOW LIMIT
1158	002464	167776			167776		;HIGH LIMIT
1159	002466	001210			LASTADD		;STORE AT THIS LOCATION
1160	002470	001			.BYTE 1		;MASK
1161	002471	001			.BYTE 1		;HOW MANY TIMES + 2

```

1162 002472 000167 177656          JMP      1$      ;DO IT AGAIN
1163 002476                                OUTMUL:
1164 002476 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1165 002500 017453                    MLEVEL        ;MESSAGE
1166 002502 104405                    PARAM          ;CONVERT STRING
1167 002504 000004                    4             ;LOW LIMIT
1168 002506 000007                    7             ;HIGH LIMIT
1169 002510 020254                    DUPRT          ;STORE AT THIS LOCATION
1170 002512 000          .BYTE        0             ;MASK
1171 002513 001          .BYTE        1             ;HOW MANY TIMES + 2
1172 002514 004767 015464          JSR      PC,DULEV
1173                                ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1174                                ;BUFFER TO THE CHARACTERS '1' AND '2'
1175                                ;IF THE CHARACTER IS '1' CLEAR THE FLAG
1176                                ;IF THE CHARACTER IS '2' SET THE FLAG
1177 002520                                AAA:
1178 002520 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1179 002522 017500                    MSYNC          ;MESSAGE
1180 002524 122767 000061 015312 3$: CMPB      #'1',INBUF ;IS IT '1' ?
1181 002532 001003                    BNE          1$
1182 002534 105067 176436                    CLRB      SYNCNO ;000
1183 002540 000412                    BR          4$
1184 002542 122767 000062 015274 1$: CMPB      #'2',INBUF ;IS IT '2' ?
1185 002550 001004                    BNE          2$
1186 002552 112767 177777 176416          MOVB      #-1,SYNCNO ;377
1187 002560 000402                    BR          4$
1188 002562 104404                    2$: INSTR          ;RETRY
1189 002564 000757                    BR          3$
1190 002566 000240                    4$: NOP
1191 002570 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1192 002572 017546                    MWIRE6        ;MESSAGE
1193 002574 104412                    SETFLG        ;SET FLAG BASED UPON INPUT STRING
1194 002576 001177                    SEXMIT        ;THIS FLAG
1195 002600 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1196 002602 017614                    MWIRE5        ;MESSAGE
1197 002604 104412                    SETFLG        ;SET FLAG BASED UPON INPUT STRING
1198 002606 001200                    SEREC        ;THIS FLAG
1199 002610 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1200 002612 017661                    MWIRE4        ;MESSAGE
1201 002614 104412                    SETFLG        ;SET FLAG BASED UPON INPUT STRING
1202 002616 001201                    OPTCLR        ;THIS FLAG
1203 002620 104403                    INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1204 002622 017735                    MEXTJ        ;MESSAGE
1205 002624 104412                    SETFLG        ;SET FLAG BASED UPON INPUT STRING
1206 002626 001203                    JMRBY        ;THIS FLAG
1207
1208
1209                                ;TEST START AND RESTART
1210
1211 002630 012767 000340 175140 .BEGIN: MOV      #340,PS ;LOCK OUT INTERRUPTS
1212                                ; ***** LOAD STATUS MAP *****
1213                                ; THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1214                                ; USED IN THIS PROGRAM AND WILL BE PASSED TO ANY
1215                                ; OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1216 002636 032777 000200 176234          BIT      #SW07 ;@SWR ; SW07 SET , IF YES BRANCH
1217 002644 001132                    BNE      HEREU

```

1218	002646	116767	176324	176424	MOVB SYNCNO	,NOSYNC	: SYNC CHARS
1219	002654	116767	176317	176417	MOVB SEXMIT	,MITSEX	: XMIT JUMPER
1220	002662	116767	176312	176412	MOVB SEREC	,RESEC	: SEC REC JUMPER
1221	002670	116767	176305	176405	MOVB OPTCLR	,CLROPT	: OPTIONAL JUMPER
1222	002676	116767	176300	176400	MOVB MULT	,DMULT	: MULTIPLE DEVICES
1223	002704	116767	176273	176373	MOVB JMRBY	,BYJMR	: EXTERNAL MODEM
1224	002712	016767	176266	176366	MOV BASEADD	,ADDBASE	: PROG CONTROLLED 1ST ADDR
1225	002720	016767	176262	176362	MOV KEEPADD	,ADDKEEP	: SAVED 1ST DEVICE ADDR
1226	002726	016767	176256	176356	MOV LASTADD	,ADDLAST	: LAST DEVICE RXCSR ADDR
1227	002734	016767	176254	176354	MOV KEEPIV	,IVKEEP	: SAVED INTR VECTOR
1228	002742	016767	176244	176344	MOV BASEIV	,IVBASE	: RELOAD BASE INTR VECTOR
1229	002750	016767	176242	176342	MOV ACTREG	,REGACT	: ACTIVE REGISTER
1230	002756	016767	176236	176336	MOV ROTADD	,ADDROT	: ROTATING POINTER
1231	002764	013737	020254	001324	MOV @WDUPRT	,@WPRTDU	: DU11 PRIORITY
1232	002772	016767	015736	176326	MOV DURIV	,RIVDU	: REC INTR VECTOR
1233	003000	016767	015732	176326	MOV DURIS	,RISDU	: REC INTR STATUS
1234	003006	016767	015726	176314	MOV DUTIV	,TIVDU	: XMIT INTR VECTOR
1235	003014	016767	015722	176310	MOV DUTIS	,TISDU	: XMIT INTR STATUS
1236	003022	016767	015230	176306	MOV LESS1	,L1ESS	: PRIORITY TO ALLOW INTR
1237	003030	016767	015356	176326	MOV DUBASE	,BASEDU	: RXCSR BASE ADDRESS
1238	003036	016767	015646	176274	MOV RXCSR,	CSRRX	
1239	003044	016767	015642	176270	MOV HRXCSR,	CSRHRX	
1240	003052	016767	015636	176264	MOV RXDBUF,	BUFRXD	
1241	003060	016767	015632	176260	MOV HRXDBUF,	BUFHRXD	
1242	003066	016767	015626	176254	MOV PARCSR,	CSRPAR	
1243	003074	016767	015622	176250	MOV HPARCSR,	CSRHPAR	
1244	003102	016767	015616	176244	MOV TXCSR,	CSRTX	
1245	003110	016767	015612	176240	MOV HTXCSR,	CSRHTX	
1246	003116	016767	015606	176234	MOV TXDBUF,	BUFTXD	
1247	003124	016767	015602	176230	MOV HTXDBUF,	BUFHTXD	
1248					:*****		
1249					: THE FOLLOWING CODE WILL PRINT		
1250					: THE CONVERSATIONALLY SET JUMPER		
1251					: SETTINGS FROM THE STATUS MAP		
1252					: ON THE FIRST PASS OF		
1253					: THIS DIAGNOSTIC OR		
1254					: JUST AFTER THE QUESTIONING		
1255					: AND ANSWERING .		
1256					:*****		
1257							
1258	003132	005737	001222		HEREU:	TST @WFLAG	: TEST IF 1ST PASS
1259	003136	001402				BEQ SETFG	: IF FIRST PASS SET FLAG/PRINT
1260	003140	000167	000116			JMP THRU	: AROUND IF PASS > 1
1261	003144	104402	020012		SETFG:	TYPE MSTATUS	: PRINT 'STATUS MAP'
1262	003150	062737	000001	001222		ADD #1, @WFLAG	: SET FLAG ON 1ST PASS
1263	003156	010067	176042			MOV R0, HOLD0	: SAVE R0
1264	003162	010167	176040			MOV R1, HOLD1	: SAVE R1
1265	003166	012767	000003	176034		MOV #3, COUNT1	: COUNTER FOR WORDS PRINTED
1266	003174	012700	000002			MOV #BUFF1, R0	
1267	003200	012701	001300			MOV #STATUS, R1	: (BUFF1)=STATUS ETC.
1268	003204	010120			FILBUF:	MOV R1, (R0)+	: LOAD BUFF AS ABOVE
1269	003206	062701	000002			ADD #2, R1	: PREPARE STATUS ADDRESS
1270	003212	020127	001306			CMP R1, #STATUS+6	: CHECK IF 3 WORDS LOADED
1271	003216	001372				BNE FILBUF	: BACK TO LOAD NEXT ADDRESS
1272	003220	012700	000002			MOV #BUFF1, R0	: LOAD FOR PRINT OUT
1273	003224	010067	176006		UP:	MOV R0, TABLE+4	: LOAD ADDRESS TO PRINT

1274	003230	012067	176006			MOV (R0)+, TABLE+10	: LOAD CONTENTS
1275	003234	104410	001232			CONVRT, TABLE	: PRINT ADDRESS/CONTENTS PAIR
1276	003240	104402	017254			TYPE, MCRLF	: CR AND LF
1277	003244	005367	175760			DEC COUNT1	: COUNT WORDS PRINTED
1278	003250	001365				BNE UP	: GO PRINT NEXT ADDRESS/CONTENTS
1279	003252	016700	175746			MOV HOLD0, R0	
1280	003256	016701	175744			MOV HOLD1, R1	
1281						;*****	
1282							
1283	003262	012706	001100		THRU:	MOV #STACK, SP	: SET UP STACK
1284	003266	005737	000042			TST @#42	: IS PROGRAM UNDER MONITOR CONTROL
1285	003272	001056				BNE 3\$	
1286	003274	105767	175702			TSTB MULTD	: DON'T ALLOW LOCK ON TEST IF RUNNING
1287							: MULTIPLE DEVICES
1288	003300	001407				BEQ 5\$	: IF NO, TEST FOR LOCK ON TEST
1289	003302	016767	011404	011304		MOV BRW, TTST	: RESTORE NORMAL SCOPE LOOP
1290	003310	016767	011400	011300		MOV BRX, TTST+2	: DITTO
1291	003316	000444				BR 3\$	: JUMP AROUND IF YES
1292	003320	032777	000004	175552	5\$:	BIT #BIT2, @SWR	: CHECK FOR LOCK ON TEST
1293	003326	001416				BEQ 1\$	
1294	003330	104403				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1295	003332	017410				MLOCK	: MESSAGE
1296	003334	104412				SETFLG	: SET FLAG BASED UPON INPUT STRING
1297	003336	001247				LOKFLG	: THIS FLAG
1298	003340	105767	175703			TSTB LOKFLG	: IS LOCK ON TEST OPTION SELECTED
1299	003344	001407				BEQ 1\$	
1300	003346	012767	000240	011240		MOV #NOP, TTST	
1301	003354	012767	000240	011234		MOV #NOP, TTST+2	: SET UP TO LOCK
1302	003362	000406				BR 2\$	
1303	003364	016767	011322	011222	1\$:	MOV BRW, TTST	
1304	003372	016767	011316	011216		MOV BRX, TTST+2	: LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1305	003400	032777	000002	175472	2\$:	BIT #SW01, @SWR	: IF SW01=1, GET STARTING PC
1306	003406	001410				BEQ 3\$	
1307	003410	104403				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1308	003412	017375				MTSTPC	: MESSAGE
1309	003414	104405				PARAM	: CONVERT STRING
1310	003416	003446				TST1	: LOW LIMIT
1311	003420	013740				TLAST	: HIGH LIMIT
1312	003422	001114				RTRN	: STORE AT THIS LOCATION
1313	003424	001			.BYTE	1	: MASK
1314	003425	001			.BYTE	1	: HOW MANY TIMES + 2
1315	003426	000403				BR 4\$	
1316	003430	012767	003446	175456	3\$:	MOV #TST1, RTRN	: START AT TEST 1
1317	003436	104402	017371		4\$:	TYPE, MR	: TYPE R
1318	003442	000177	175446			JMP @RTRN	: START TESTING
1319							
1320						:: THIS TEST PROVES EXISTANCE OF DEVICE REGISTERS	
1321							
1322	003446	012767	000001	175452	TST1:	MOV #1, TSTNO	: SAVE THIS
1323	003454	012767	003534	175434		MOV #TST2, NEXT	: GO TO THIS TEST WHEN THRU
1324	003462	012737	020652	000004		MOV #TRPREG, @#4	: SETUP TRAPCATCHER
1325	003470	012737	000340	000006		MOV #LEVEL7, @#6	
1326	003476	105277	015206			INCB @RXCSR	: TEST THIS REG
1327	003502	000401				BR 64\$	: IF OK JMP AROUND HLT
1328	003504	104000				HLT	: CHECK DEVICE REG ADDRESSES
1329	003506	105277	015200		64\$:	INCB @HRXCSR	: TEST UPPER BYTE THIS REGISTER

1330	003512	000401				BR	65\$		:IF OK JMP AROUND HLT
1331	003514	104000				HLT			:CHECK DEVICE REG ADDRESSES
1332	003516	012737	000006	000004	65\$:	MOV	#6,@#4		:RESTORE TRAPCATCHER
1333	003524	012737	000000	000006		MOV	#0,@#6		:
1334	003532	104400				SCOPE			:
1335						::THIS		TEST PROVES EXISTANCE OF DEVICE REGISTERS	
1336						::			
1337	003534	012767	000002	175364	TST2:	MOV	#2,TSTNO		:SAVE THIS
1338	003542	012767	003622	175346		MOV	#TST3,NEXT		:GO TO THIS TEST WHEN THRU
1339	003550	012737	020652	000004		MOV	#TRPRÉG,@#4		:SETUP TRAPCATCHER
1340	003556	012737	000340	000006		MOV	#LEVEL7,@#6		:
1341	003564	105277	015124			INCB	@RXDBUF		:TEST THIS REG
1342	003570	000401				BR	64\$		:IF OK JMP AROUND HLT
1343	003572	104000				HLT			:CHECK DEVICE REG ADDRESSES
1344	003574	105277	015116		64\$:	INCB	@HRXDBUF		:TEST UPPER BYTE THIS REGISTER
1345	003600	000401				BR	65\$		:IF OK JMP AROUND HLT
1346	003602	104000				HLT			:CHECK DEVICE REG ADDRESSES
1347	003604	012737	000006	000004	65\$:	MOV	#6,@#4		:RESTORE TRAPCATCHER
1348	003612	012737	000000	000006		MOV	#0,@#6		:
1349	003620	104400				SCOPE			:
1350						::THIS		TEST PROVES EXISTANCE OF DEVICE REGISTERS	
1351						::			
1352	003622	012767	000003	175276	TST3:	MOV	#3,TSTNO		:SAVE THIS
1353	003630	012767	003710	175260		MOV	#TST4,NEXT		:GO TO THIS TEST WHEN THRU
1354	003636	012737	020652	000004		MOV	#TRPRÉG,@#4		:SETUP TRAPCATCHER
1355	003644	012737	000340	000006		MOV	#LEVEL7,@#6		:
1356	003652	105277	015042			INCB	@PARCSR		:TEST THIS REG
1357	003656	000401				BR	64\$		:IF OK JMP AROUND HLT
1358	003660	104000				HLT			:CHECK DEVICE REG ADDRESSES
1359	003662	105277	015034		64\$:	INCB	@HPARCSR		:TEST UPPER BYTE THIS REGISTER
1360	003666	000401				BR	65\$		:IF OK JMP AROUND HLT
1361	003670	104000				HLT			:CHECK DEVICE REG ADDRESSES
1362	003672	012737	000006	000004	65\$:	MOV	#6,@#4		:RESTORE TRAPCATCHER
1363	003700	012737	000000	000006		MOV	#0,@#6		:
1364	003706	104400				SCOPE			:
1365						::THIS		TEST PROVES EXISTANCE OF DEVICE REGISTERS	
1366						::			
1367	003710	012767	000004	175270	TST4:	MOV	#4,TSTNO		:SAVE THIS
1368	003716	012767	003776	175172		MOV	#TST5,NEXT		:GO TO THIS TEST WHEN THRU
1369	003724	012737	020652	000004		MOV	#TRPRÉG,@#4		:SETUP TRAPCATCHER
1370	003732	012737	000340	000006		MOV	#LEVEL7,@#6		:
1371	003740	105277	014760			INCB	@TXCSR		:TEST THIS REG
1372	003744	000401				BR	64\$		:IF OK JMP AROUND HLT
1373	003746	104000				HLT			:CHECK DEVICE REG ADDRESSES
1374	003750	105277	014752		64\$:	INCB	@HTXCSR ;TEST UPPER BYTE THIS REGISTER		
1375	003754	000401				BR	65\$		:IF OK JMP AROUND HLT
1376	003756	104000				HLT			:CHECK DEVICE REG ADDRESSES
1377	003760	012737	000006	000004	65\$:	MOV	#6,@#4		:RESTORE TRAPCATCHER
1378	003766	012737	000000	000006		MOV	#0,@#6		:
1379	003774	104400				SCOPE			:
1380						::THIS		TEST PROVES EXISTANCE OF DEVICE REGISTERS	
1381						::			
1382	003776	012767	000005	175122	TST5:	MOV	#5,TSTNO		:SAVE THIS
1383	004004	012767	004064	175104		MOV	#TST6,NEXT		:GO TO THIS TEST WHEN THRU
1384	004012	012737	020652	000004		MOV	#TRPRÉG,@#4		:SETUP TRAPCATCHER
1385	004020	012737	000340	000006		MOV	#LEVEL7,@#6		:

1386	004026	105277	014676		INCB	@TXDBUF		:TEST THIS REG
1387	004032	000401			BR	64\$		:IF OK JMP AROUND HLT
1388	004034	104000			HLT			:CHECK DEVICE REG ADDRESSES
1389	004036	105277	014670	64\$:	INCB	@HTXDBUF		:TEST UPPER BYTE THIS REGISTER
1390	004042	000401			BR	65\$		:IF OK JMP AROUND HLT
1391	004044	104000			HLT			:CHECK DEVICE REG ADDRESSES
1392	004046	012737	000006	000004	65\$:	MOV	#6,@#4	:RESTORE TRAPCATCHER
1393	004054	012737	000000	000006		MOV	#0,@#6	
1394	004062	104400				SCOPE		
1395						::BUS DRIVER TEST		
1396								
1397	004064	012767	000006	175034	TST6:	MOV	#6,TSTNO	:SAVE THIS
1398	004072	012767	004114	175016		MOV	#TST7,NEXT	:GO TO THIS TEST WHEN THRU
1399	004100	022777	177777	014622		CMP	#177777,@TXDBUF	
1400	004106	001401				BEQ	.+4	
1401	004110	104000				HLT		:READING TXDBUF SHOULD BE ALL 1'S
1402	004112	104400				SCOPE		
1403						::THIS TEST PERFORMS MASTER RESET TESTING &		
1404						::TESTING OF READ/WRITE BIT DTR		
1405								
1406	004114	012767	000007	175004	TST7:	MOV	#7,TSTNO	:SAVE THIS
1407	004122	012767	004244	174766		MOV	#TST8,NEXT	:GO TO THIS TEST WHEN THRU
1408	004130	052777	000002	014552		BIS	#DTR,@RXCSR	:SET THIS BIT
1409	004136	032777	000002	014544		BIT	#DTR,@RXCSR	:TEST THIS BIT
1410	004144	001001				BNE	64\$	:BR IF '1'
1411	004146	104000				HLT		:THIS BIT SHOULD BE SET
1412	004150				64\$:			
1413	004150	042777	000002	014532		BIC	#DTR,@RXCSR	:CLR THIS BIT
1414	004156	032777	000002	014524		BIT	#DTR,@RXCSR	:TEST THIS BIT
1415	004164	001401				BEQ	65\$	:BR IF '0'
1416	004166	104000				HLT		:THIS BIT SHOULD BE CLR
1417	004170				65\$:			
1418						:NOW SET THIS BIT		
1419	004170	052777	000002	014512		BIS	#DTR,@RXCSR	
1420	004176	052777	000400	014520		BIS	#MRESET,@TXCSR	:MASTER RESET
1421						::CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER		
1422								
1423	004204	105767	174771			TSTB	OPTCLR	:TEST FLAG
1424	004210	100006				BPL	1\$	:OPTIONAL CLR JUMPER IS NOT IN
1425	004212	032777	000002	014470		BIT	#DTR,@RXCSR	:TEST THIS BIT
1426	004220	001401				BEQ	66\$	:BR IF '0'
1427	004222	104000				HLT		:CHECK OUT MASTER RESET LOGIC
1428	004224				66\$:			
1429	004224	000405				BR	2\$	:JMP AROUND
1430	004226	032777	000002	014454	1\$:	BIT	#DTR,@RXCSR	:TEST THIS BIT
1431	004234	001001				BNE	67\$	:BR IF '1'
1432	004236	104000				HLT		:CHECK OUT OPTIONAL CLR JUMPER
1433	004240				67\$:			
1434	004240	000240			2\$:			
1435	004242	104400				NOP		
1436						SCOPE		
1437						::THIS TEST PERFORMS MASTER RESET TESTING &		
1438						::TESTING OF READ/WRITE BIT RTS		
1439	004244	012767	000010	174654	TST8:	MOV	#8,TSTNO	:SAVE THIS
1440	004252	012767	004374	174636		MOV	#TST9,NEXT	:GO TO THIS TEST WHEN THRU
1441	004260	052777	000004	014422		BIS	#RTS,@RXCSR	:SET THIS BIT

1442	004266	032777	000004	014414		BIT	#RTS,@RXCSR	:TEST THIS BIT
1443	004274	001001				BNE	64\$	:BR IF '1'
1444	004276	104000				HLT		:THIS BIT SHOULD BE SET
1445	004300				64\$:			
1446	004300	042777	000004	014402		BIC	#RTS,@RXCSR	:CLR THIS BIT
1447	004306	032777	000004	014374		BIT	#RTS,@RXCSR	:TEST THIS BIT
1448	004314	001401				BEQ	65\$	:BR IF '0'
1449	004316	104000				HLT		:THIS BIT SHOULD BE CLR
1450	004320				65\$:			
1451								:NOW SET THIS BIT
1452	004320	052777	000004	014362		BIS	#RTS,@RXCSR	
1453	004326	052777	000400	014370		BIS	#MRESET,@TXCSR	:MASTER RESET
1454								:CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1455								
1456	004334	105767	174641			TSTB	OPTCLR	:TEST FLAG
1457	004340	100006				BPL	1\$	:OPTIONAL CLR JUMPER IS NOT IN
1458	004342	032777	000004	014340		BIT	#RTS,@RXCSR	:TEST THIS BIT
1459	004350	001401				BEQ	66\$	:BR IF '0'
1460	004352	104000				HLT		:CHECK OUT MASTER RESET LOGIC
1461	004354				66\$:			
1462	004354	000405				BR	2\$	:JMP AROUND
1463	004356	032777	000004	014324	1\$:	BIT	#RTS,@RXCSR	:TEST THIS BIT
1464	004364	001001				BNE	67\$	:BR IF '1'
1465	004366	104000				HLT		:CHECK OUT OPTIONAL CLR JUMPER
1466	004370				67\$:			
1467	004370	000240			2\$:			
1468	004372	104400						:NOP
1469								:SCOPE
1470								:THIS TEST PERFORMS MASTER RESET TESTING &
1471								:TESTING OF READ/WRITE BIT STD
1472	004374	012767	000011	174524	TST9:	MOV	#9,TSTNO	:SAVE THIS
1473	004402	012767	004524	174506		MOV	#TST10,NEXT	:GO TO THIS TEST WHEN THRU
1474	004410	052777	000010	014272		BIS	#STD,@RXCSR	:SET THIS BIT
1475	004416	032777	000010	014264		BIT	#STD,@RXCSR	:TEST THIS BIT
1476	004424	001001				BNE	64\$	:BR IF '1'
1477	004426	104000				HLT		:THIS BIT SHOULD BE SET
1478	004430				64\$:			
1479	004430	042777	000010	014252		BIC	#STD,@RXCSR	:CLR THIS BIT
1480	004436	032777	000010	014244		BIT	#STD,@RXCSR	:TEST THIS BIT
1481	004444	001401				BEQ	65\$	:BR IF '0'
1482	004446	104000				HLT		:THIS BIT SHOULD BE CLR
1483	004450				65\$:			
1484								:NOW SET THIS BIT
1485	004450	052777	000010	014232		BIS	#STD,@RXCSR	
1486	004456	052777	000400	014240		BIS	#MRESET,@TXCSR	:MASTER RESET
1487								:CHECK EXISTANCE OF OPTIONAL CLEAR JUMPER
1488								
1489	004464	105767	174511			TSTB	OPTCLR	:TEST FLAG
1490	004470	100006				BPL	1\$	:OPTIONAL CLR JUMPER IS NOT IN
1491	004472	032777	000010	014210		BIT	#STD,@RXCSR	:TEST THIS BIT
1492	004500	001401				BEQ	66\$	:BR IF '0'
1493	004502	104000				HLT		:CHECK OUT MASTER RESET LOGIC
1494	004504				66\$:			
1495	004504	000405				BR	2\$	:JMP AROUND
1496	004506	032777	000010	014174	1\$:	BIT	#STD,@RXCSR	:TEST THIS BIT
1497	004514	001001				BNE	67\$	:BR IF '1'

```

1498 004516 104000          HLT          ;CHECK OUT OPTIONAL CLR JUMPER
1499 004520          67$:
1500 004520 000240          2$: NOP
1501 004522 104400          SCOPE
1502          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1503          ;;TESTING OF READ/WRITE BIT SYNSCH
1504          ;;
1505 004524 012767 000012 174374 TST10: MOV    #10,TSTNO      ;SAVE THIS
1506 004532 012767 004630 174356      MOV    #TST11,NEXT      ;GO TO THIS TEST WHEN THRU
1507 004540 052777 000020 014142      BIS    #SYNSCH,@RXCSR    ;SET THIS BIT
1508 004546 032777 000020 014134      BIT    #SYNSCH,@RXCSR    ;TEST THIS BIT
1509 004554 001001          BNE    64$      ;BR IF '1'
1510 004556 104000          HLT          ;THIS BIT SHOULD BE SET
1511 004560          64$:
1512 004560 042777 000020 014122      BIC    #SYNSCH,@RXCSR    ;CLR THIS BIT
1513 004566 032777 000020 014114      BIT    #SYNSCH,@RXCSR    ;TEST THIS BIT
1514 004574 001401          BEQ    65$      ;BR IF '0'
1515 004576 104000          HLT          ;THIS BIT SHOULD BE CLR
1516 004600          65$:
1517          ;NOW SET THIS BIT
1518 004600 052777 000020 014102      BIS    #SYNSCH,@RXCSR
1519 004606 052777 000400 014110      BIS    #MRESET,@TXCSR    ;MASTER RESET
1520 004614 032777 000020 014066      BIT    #SYNSCH,@RXCSR    ;TEST THIS BIT
1521 004622 001401          BEQ    66$      ;BR IF '0'
1522 004624 104000          HLT          ;CHECK OUT MASTER RESET LOGIC
1523 004626          66$:
1524 004626 104400          SCOPE
1525          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1526          ;;TESTING OF READ/WRITE BIT DSINTE
1527          ;;
1528 004630 012767 000013 174270 TST11: MOV    #11,TSTNO      ;SAVE THIS
1529 004636 012767 004734 174252      MOV    #TST12,NEXT      ;GO TO THIS TEST WHEN THRU
1530 004644 052777 000040 014036      BIS    #DSINTE,@RXCSR    ;SET THIS BIT
1531 004652 032777 000040 014030      BIT    #DSINTE,@RXCSR    ;TEST THIS BIT
1532 004660 001001          BNE    64$      ;BR IF '1'
1533 004662 104000          HLT          ;THIS BIT SHOULD BE SET
1534 004664          64$:
1535 004664 042777 000040 014016      BIC    #DSINTE,@RXCSR    ;CLR THIS BIT
1536 004672 032777 000040 014010      BIT    #DSINTE,@RXCSR    ;TEST THIS BIT
1537 004700 001401          BEQ    65$      ;BR IF '0'
1538 004702 104000          HLT          ;THIS BIT SHOULD BE CLR
1539 004704          65$:
1540          ;NOW SET THIS BIT
1541 004704 052777 000040 013776      BIS    #DSINTE,@RXCSR
1542 004712 052777 000400 014004      BIS    #MRESET,@TXCSR    ;MASTER RESET
1543 004720 032777 000040 013762      BIT    #DSINTE,@RXCSR    ;TEST THIS BIT
1544 004726 001401          BEQ    66$      ;BR IF '0'
1545 004730 104000          HLT          ;CHECK OUT MASTER RESET LOGIC
1546 004732          66$:
1547 004732 104400          SCOPE
1548          ;;THIS TEST PERFORMS MASTER RESET TESTING &
1549          ;;TESTING OF READ/WRITE BIT RINTEN
1550          ;;
1551 004734 012767 000014 174164 TST12: MOV    #12,TSTNO      ;SAVE THIS
1552 004742 012767 005040 174146      MOV    #TST13,NEXT      ;GO TO THIS TEST WHEN THRU
1553 004750 052777 000100 013732      BIS    #RINTEN,@RXCSR    ;SET THIS BIT
  
```

```

1554 004756 032777 000100 013724      BIT      #RINTEN,@RXCSR ;TEST THIS BIT
1555 004764 001001                      BNE      64$      ;BR IF '1'
1556 004766 104000                      HLT                      ;THIS BIT SHOULD BE SET
1557 004770
1558 004770 042777 000100 013712      BIC      #RINTEN,@RXCSR ;CLR THIS BIT
1559 004776 032777 000100 013704      BIT      #RINTEN,@RXCSR ;TEST THIS BIT
1560 005004 001401                      BEQ      65$      ;BR IF '0'
1561 005006 104000                      HLT                      ;THIS BIT SHOULD BE CLR
1562 005010
1563                                ;NOW SET THIS BIT
1564 005010 052777 000100 013672      BIS      #RINTEN,@RXCSR
1565 005016 052777 000400 013700      BIS      #MRESET,@TXCSR ;MASTER RESET
1566 005024 032777 000100 013656      BIT      #RINTEN,@RXCSR ;TEST THIS BIT
1567 005032 001401                      BEQ      66$      ;BR IF '0'
1568 005034 104000                      HLT                      ;CHECK OUT MASTER RESET LOGIC
1569 005036
1570 005036 104400
1571                                SCOPE
1572                                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1573                                ;;TESTING OF READ/WRITE BIT STPSYN
1574 005040 012767 000015 174060      TST13: MOV      #13,TSTNO      ;SAVE THIS
1575 005046 012767 005144 174042      MOV      #TST14,NEXT      ;GO TO THIS TEST WHEN THRU
1576 005054 052777 000400 013626      BIS      #STPSYN,@RXCSR ;SET THIS BIT
1577 005062 032777 000400 013620      BIT      #STPSYN,@RXCSR ;TEST THIS BIT
1578 005070 001001                      BNE      64$      ;BR IF '1'
1579 005072 104000                      HLT                      ;THIS BIT SHOULD BE SET
1580 005074
1581 005074 042777 000400 013606      BIC      #STPSYN,@RXCSR ;CLR THIS BIT
1582 005102 032777 000400 013600      BIT      #STPSYN,@RXCSR ;TEST THIS BIT
1583 005110 001401                      BEQ      65$      ;BR IF '0'
1584 005112 104000                      HLT                      ;THIS BIT SHOULD BE CLR
1585 005114
1586                                ;NOW SET THIS BIT
1587 005114 052777 000400 013566      BIS      #STPSYN,@RXCSR
1588 005122 052777 000400 013574      BIS      #MRESET,@TXCSR ;MASTER RESET
1589 005130 032777 000400 013552      BIT      #STPSYN,@RXCSR ;TEST THIS BIT
1590 005136 001401                      BEQ      66$      ;BR IF '0'
1591 005140 104000                      HLT                      ;CHECK OUT MASTER RESET LOGIC
1592 005142
1593 005142 104400
1594                                SCOPE
1595                                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1596                                ;;TESTING OF READ/WRITE BIT BREAK
1597 005144 012767 000016 173754      TST14: MOV      #14,TSTNO      ;SAVE THIS
1598 005152 012767 005250 173736      MOV      #TST15,NEXT      ;GO TO THIS TEST WHEN THRU
1599 005160 052777 000001 013536      BIS      #BREAK,@TXCSR  ;SET THIS BIT
1600 005166 032777 000001 013530      BIT      #BREAK,@TXCSR  ;TEST THIS BIT
1601 005174 001001                      BNE      64$      ;BR IF '1'
1602 005176 104000                      HLT                      ;THIS BIT SHOULD BE SET
1603 005200
1604 005200 042777 000001 013516      BIC      #BREAK,@TXCSR  ;CLR THIS BIT
1605 005206 032777 000001 013510      BIT      #BREAK,@TXCSR  ;TEST THIS BIT
1606 005214 001401                      BEQ      65$      ;BR IF '0'
1607 005216 104000                      HLT                      ;THIS BIT SHOULD BE CLR
1608 005220
1609                                ;NOW SET THIS BIT

```

```

1610 005220 052777 000001 013476 BIS #BREAK,@TXCSR
1611 005226 052777 000400 013470 BIS #MRESET,@TXCSR ;MASTER RESET
1612 005234 032777 000001 013462 BIT #BREAK,@TXCSR ;TEST THIS BIT
1613 005242 001401 BEQ 66$ ;BR IF '0'
1614 005244 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1615 005246 66$:
1616 005246 104400 SCOPE
1617 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1618 ;:TESTING OF READ/WRITE BIT HDXEN
1619 ;:
1620 005250 012767 000017 173650 TST15: MOV #15,TSTNO ;SAVE THIS
1621 005256 012767 005354 173632 MOV #TST16,NEXT ;GO TO THIS TEST WHEN THRU
1622 005264 052777 000010 013432 BIS #HDXEN,@TXCSR ;SET THIS BIT
1623 005272 032777 000010 013424 BIT #HDXEN,@TXCSR ;TEST THIS BIT
1624 005300 001001 BNE 64$ ;BR IF '1'
1625 005302 104000 HLT ;THIS BIT SHOULD BE SET
1626 005304 64$:
1627 005304 042777 000010 013412 BIC #HDXEN,@TXCSR ;CLR THIS BIT
1628 005312 032777 000010 013404 BIT #HDXEN,@TXCSR ;TEST THIS BIT
1629 005320 001401 BEQ 65$ ;BR IF '0'
1630 005322 104000 HLT ;THIS BIT SHOULD BE CLR
1631 005324 65$:
1632 ;:NOW SET THIS BIT
1633 005324 052777 000010 013372 BIS #HDXEN,@TXCSR
1634 005332 052777 000400 013364 BIS #MRESET,@TXCSR ;MASTER RESET
1635 005340 032777 000010 013356 BIT #HDXEN,@TXCSR ;TEST THIS BIT
1636 005346 001401 BEQ 66$ ;BR IF '0'
1637 005350 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1638 005352 66$:
1639 005352 104400 SCOPE
1640 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1641 ;:TESTING OF READ/WRITE BIT SEND
1642 ;:
1643 005354 012767 000020 173544 TST16: MOV #16,TSTNO ;SAVE THIS
1644 005362 012767 005460 173526 MOV #TST17,NEXT ;GO TO THIS TEST WHEN THRU
1645 005370 052777 000020 013326 BIS #SEND,@TXCSR ;SET THIS BIT
1646 005376 032777 000020 013320 BIT #SEND,@TXCSR ;TEST THIS BIT
1647 005404 001001 BNE 64$ ;BR IF '1'
1648 005406 104000 HLT ;THIS BIT SHOULD BE SET
1649 005410 64$:
1650 005410 042777 000020 013306 BIC #SEND,@TXCSR ;CLR THIS BIT
1651 005416 032777 000020 013300 BIT #SEND,@TXCSR ;TEST THIS BIT
1652 005424 001401 BEQ 65$ ;BR IF '0'
1653 005426 104000 HLT ;THIS BIT SHOULD BE CLR
1654 005430 65$:
1655 ;:NOW SET THIS BIT
1656 005430 052777 000020 013266 BIS #SEND,@TXCSR
1657 005436 052777 000400 013260 BIS #MRESET,@TXCSR ;MASTER RESET
1658 005444 032777 000020 013252 BIT #SEND,@TXCSR ;TEST THIS BIT
1659 005452 001401 BEQ 66$ ;BR IF '0'
1660 005454 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1661 005456 66$:
1662 005456 104400 SCOPE
1663 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1664 ;:TESTING OF READ/WRITE BIT DNAINTE
1665 ;:

```

```

1666 005460 012767 000021 173440 TST17: MOV #17,TSTNO ;SAVE THIS
1667 005466 012767 005564 173422 MOV #TST18,NEXT ;GO TO THIS TEST WHEN THRU
1668 005474 052777 000040 013222 BIS #DNAINTE,@TXCSR ;SET THIS BIT
1669 005502 032777 000040 013214 BIT #DNAINTE,@TXCSR ;TEST THIS BIT
1670 005510 001001 BNE 64$ ;BR IF '1'
1671 005512 104000 HLT ;THIS BIT SHOULD BE SET
1672 005514 64$:
1673 005514 042777 000040 013202 BIC #DNAINTE,@TXCSR ;CLR THIS BIT
1674 005522 032777 000040 013174 BIT #DNAINTE,@TXCSR ;TEST THIS BIT
1675 005530 001401 BEQ 65$ ;BR IF '0'
1676 005532 104000 HLT ;THIS BIT SHOULD BE CLR
1677 005534 65$:
1678 ;NOW SET THIS BIT
1679 005534 052777 000040 013162 BIS #DNAINTE,@TXCSR
1680 005542 052777 000400 013154 BIS #MRESET,@TXCSR ;MASTER RESET
1681 005550 032777 000040 013146 BIT #DNAINTE,@TXCSR ;TEST THIS BIT
1682 005556 001401 BEQ 66$ ;BR IF '0'
1683 005560 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1684 005562 66$:
1685 005562 104400 SCOPE
1686 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1687 ;:TESTING OF READ/WRITE BIT TXINTE
1688 ;:
1689 005564 012767 000022 173334 TST18: MOV #18,TSTNO ;SAVE THIS
1690 005572 012767 005670 173316 MOV #TST19,NEXT ;GO TO THIS TEST WHEN THRU
1691 005600 052777 000100 013116 BIS #TXINTE,@TXCSR ;SET THIS BIT
1692 005606 032777 000100 013110 BIT #TXINTE,@TXCSR ;TEST THIS BIT
1693 005614 001001 BNE 64$ ;BR IF '1'
1694 005616 104000 HLT ;THIS BIT SHOULD BE SET
1695 005620 64$:
1696 005620 042777 000100 013076 BIC #TXINTE,@TXCSR ;CLR THIS BIT
1697 005626 032777 000100 013070 BIT #TXINTE,@TXCSR ;TEST THIS BIT
1698 005634 001401 BEQ 65$ ;BR IF '0'
1699 005636 104000 HLT ;THIS BIT SHOULD BE CLR
1700 005640 65$:
1701 ;NOW SET THIS BIT
1702 005640 052777 000100 013056 BIS #TXINTE,@TXCSR
1703 005646 052777 000400 013050 BIS #MRESET,@TXCSR ;MASTER RESET
1704 005654 032777 000100 013042 BIT #TXINTE,@TXCSR ;TEST THIS BIT
1705 005662 001401 BEQ 66$ ;BR IF '0'
1706 005664 104000 HLT ;CHECK OUT MASTER RESET LOGIC
1707 005666 66$:
1708 005666 104400 SCOPE
1709 ;:TEST MAINT MODE BIT 0
1710 ;:
1711 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1712 ;:TESTING OF READ/WRITE BIT BIT11
1713 ;:
1714 005670 012767 000023 173230 TST19: MOV #19,TSTNO ;SAVE THIS
1715 005676 012767 005774 173212 MOV #TST20,NEXT ;GO TO THIS TEST WHEN THRU
1716 005704 052777 004000 013012 BIS #BIT11,@TXCSR ;SET THIS BIT
1717 005712 032777 004000 013004 BIT #BIT11,@TXCSR ;TEST THIS BIT
1718 005720 001001 BNE 64$ ;BR IF '1'
1719 005722 104000 HLT ;THIS BIT SHOULD BE SET
1720 005724 64$:
1721 005724 042777 004000 012712 BIC #BIT11,@TXCSR ;CLR THIS BIT

```

1722	005732	032777	004000	012764		BIT	#BIT11,@TXCSR	;TEST THIS BIT
1723	005740	001401				BEQ	65\$	;BR IF '0'
1724	005742	104000				HLT		;THIS BIT SHOULD BE CLR
1725	005744				65\$:			
1726							;NOW SET THIS BIT	
1727	005744	052777	004000	012752		BIS	#BIT11,@TXCSR	
1728	005752	052777	000400	012744		BIS	#MRESET,@TXCSR	;MASTER RESET
1729	005760	032777	004000	012736		BIT	#BIT11,@TXCSR	;TEST THIS BIT
1730	005766	001401				BEQ	66\$	;BR IF '0'
1731	005770	104000				HLT		;CHECK OUT MASTER RESET LOGIC
1732	005772				66\$:			
1733	005772	104400					SCOPE	
1734							:::TEST MAINT MODE BIT 1	
1735							:::	
1736							:::THIS TEST PERFORMS MASTER RESET TESTING &	
1737							:::TESTING OF READ/WRITE BIT BIT12	
1738							:::	
1739	005774	012767	000024	173124	TST20:	MOV	#20,TSTNO	;SAVE THIS
1740	006002	012767	006100	173106		MOV	#TST21,NEXT	;GO TO THIS TEST WHEN THRU
1741	006010	052777	010000	012706		BIS	#BIT12,@TXCSR	;SET THIS BIT
1742	006016	032777	010000	012700		BIT	#BIT12,@TXCSR	;TEST THIS BIT
1743	006024	001001				BNE	64\$	;BR IF '1'
1744	006026	104000				HLT		;THIS BIT SHOULD BE SET
1745	006030				64\$:			
1746	006030	042777	010000	012666		BIC	#BIT12,@TXCSR	;CLR THIS BIT
1747	006036	032777	010000	012660		BIT	#BIT12,@TXCSR	;TEST THIS BIT
1748	006044	001401				BEQ	65\$	;BR IF '0'
1749	006046	104000				HLT		;THIS BIT SHOULD BE CLR
1750	006050				65\$:			
1751							;NOW SET THIS BIT	
1752	006050	052777	010000	012646		BIS	#BIT12,@TXCSR	
1753	006056	052777	000400	012640		BIS	#MRESET,@TXCSR	;MASTER RESET
1754	006064	032777	010000	012632		BIT	#BIT12,@TXCSR	;TEST THIS BIT
1755	006072	001401				BEQ	66\$	;BR IF '0'
1756	006074	104000				HLT		;CHECK OUT MASTER RESET LOGIC
1757	006076				66\$:			
1758	006076	104400					SCOPE	
1759							:::THIS TEST PERFORMS MASTER RESET TESTING &	
1760							:::TESTING OF READ/WRITE BIT CLK	
1761							:::	
1762	006100	012767	000025	173020	TST21:	MOV	#21,TSTNO	;SAVE THIS
1763	006106	012767	006204	173002		MOV	#TST22,NEXT	;GO TO THIS TEST WHEN THRU
1764	006114	052777	020000	012602		BIS	#CLK,@TXCSR	;SET THIS BIT
1765	006122	032777	020000	012574		BIT	#CLK,@TXCSR	;TEST THIS BIT
1766	006130	001001				BNE	64\$	;BR IF '1'
1767	006132	104000				HLT		;THIS BIT SHOULD BE SET
1768	006134				64\$:			
1769	006134	042777	020000	012562		BIC	#CLK,@TXCSR	;CLR THIS BIT
1770	006142	032777	020000	012554		BIT	#CLK,@TXCSR	;TEST THIS BIT
1771	006150	001401				BEQ	65\$	;BR IF '0'
1772	006152	104000				HLT		;THIS BIT SHOULD BE CLR
1773	006154				65\$:			
1774							;NOW SET THIS BIT	
1775	006154	052777	020000	012542		BIS	#CLK,@TXCSR	
1776	006162	052777	000400	012534		BIS	#MRESET,@TXCSR	;MASTER RESET
1777	006170	032777	020000	012526		BIT	#CLK,@TXCSR	;TEST THIS BIT

```

1778 006176 001401      BEQ      66$      ;BR IF '0'
1779 006200 104000      HLT              ;CHECK OUT MASTER RESET LOGIC
1780 006202
1781 006202 104400      66$:
1782      SCOPE
1783      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1784      ;;TESTING OF READ/WRITE BIT MTDATA
1785 006204 012767 000026 172714 TST22: MOV      #22,TSTNO      ;SAVE THIS
1786 006212 012767 006310 172676      MOV      #TST23,NEXT      ;GO TO THIS TEST WHEN THRU
1787 006220 052777 040000 012476      BIS      #MTDATA,@TXCSR      ;SET THIS BIT
1788 006226 032777 040000 012470      BIT      #MTDATA,@TXCSR      ;TEST THIS BIT
1789 006234 001001      BNE      64$      ;BR IF '1'
1790 006236 104000      HLT              ;THIS BIT SHOULD BE SET
1791 006240
1792 006240 042777 040000 012456      64$: BIC      #MTDATA,@TXCSR      ;CLR THIS BIT
1793 006246 032777 040000 012450      BIT      #MTDATA,@TXCSR      ;TEST THIS BIT
1794 006254 001401      BEQ      65$      ;BR IF '0'
1795 006256 104000      HLT              ;THIS BIT SHOULD BE CLR
1796 006260
1797      65$: ;NOW SET THIS BIT
1798 006260 052777 040000 012436      BIS      #MTDATA,@TXCSR
1799 006266 052777 000400 012430      BIS      #MRESET,@TXCSR      ;MASTER RESET
1800 006274 032777 040000 012422      BIT      #MTDATA,@TXCSR      ;TEST THIS BIT
1801 006302 001401      BEQ      66$      ;BR IF '0'
1802 006304 104000      HLT              ;CHECK OUT MASTER RESET LOGIC
1803 006306
1804 006306 104400      66$:
1805      SCOPE
1806      ;;THIS TEST VERIFYS THAT INIT (RESET) CLEARS BITS IN THE
1807      ;;RXCSR & TXCSR
1808 006310 012767 000027 172610 TST23: MOV      #23,TSTNO      ;SAVE THIS
1809 006316 012767 006420 172572      MOV      #TST24,NEXT      ;GO TO THIS TEST WHEN THRU
1810 006324 012777 177777 012356      MOV      #177777,@RXCSR      ;SET ALL POSSIBLE BITS
1811 006332 012777 177777 012364      MOV      #177777,@TXCSR      ;DITTO
1812 006340 000005      RESET
1813 006342 012767 000340 171426      MOV      #LEVEL7,PS      ;RESTORE NON INTERRUPT STATUS
1814 006350 017701 012334      MOV      @RXCSR,R1      ;SAVE
1815 006354 017702 012344      MOV      @TXCSR,R2      ;SAVE
1816 006360 105767 172615      TSTB     OPTCLR      ;IS THE OPTIONAL CLR JUMPER ON ?
1817 006364 100402      BMI      1$      ;YES
1818 006366 042701 000016      BIC      #16,R1      ;CLR THE NON RESETABLE BITS
1819 006372 042701 073000      1$: BIC      #073000,R1      ;CLR ALL NON-CLEARABLE BITS
1820 006376 005701      TST      R1      ;ARE THEY ALL 0 ?
1821 006400 001401      BEQ      .+4
1822 006402 104000      HLT              ;ALL SPECIFIED BITS SHOULD BE CLEAR
1823 006404 042702 002200      BIC      #002200,R2      ;CLEAR ALL NON-CLEARABLE BITS
1824 006410 005702      TST      R2      ;ARE THEY ALL 0 ?
1825 006412 001401      BEQ      .+4
1826 006414 104000      HLT              ;ALL SPECIFIED BITS SHOULD BE CLEAR
1827 006416 104400      SCOPE
1828      ;;THIS TEST PERFORMS MASTER RESET TESTING &
1829      ;;TESTING OF WRITE ONLY BIT MRESET
1830
1831 006420 012767 000030 172500 TST24: MOV      #24,TSTNO      ;SAVE THIS
1832 006426 012767 006476 172462      MOV      #TST25,NEXT      ;GO TO THIS TEST WHEN THRU
1833 006434 052777 000400 012262      BIS      #MRESET,@TXCSR      ;TRY TO SET THIS BIT

```

```

1834 006442 032777 000400 012254      BIT      #MRESET,@TXCSR ;TEST THIS BIT
1835 006450 001401                BEQ      64$      ;BR IF '0'
1836 006452 104000                HLT                ;THIS BIT SHOULD NOT BE SET
1837 006454                64$:
1838 006454 052777 000400 012242      BIS      #MRESET,@TXCSR ;MASTER RESET
1839 006462 032777 000400 012234      BIT      #MRESET,@TXCSR ;TEST THIS BIT
1840 006470 001401                BEQ      65$      ;BR IF '0'
1841 006472 104000                HLT                ;THIS BIT SHOULD NOT BE SET
1842 006474                65$:
1843                                ;CHECK MASTER RESET LOGIC
1844 006474 104400                SCOPE
1845                                ;;THIS TEST VERIFYS THAT THE RXCSR & TXCSR CAN BE BYTE ADDRESSED (DATOB)
1846                                ;;
1847 006476 012767 000031 172422      TST25: MOV      #25,TSTNO      ;SAVE THIS
1848 006504 012767 006664 172404      MOV      #TST26,NEXT      ;GO TO THIS TEST WHEN THRU
1849 006512 052777 000400 012204      BIS      #MRESET,@TXCSR ;MASTER RESET
1850 006520 105767 172455      TSTB      OPTCLR ;IS THE OPTIONAL CLR JUMPER ON ?
1851 006524 100405                BMI      1$      ;YES
1852 006526 012777 000000 012154      MOV      #0,@RXCSR      ;CLR OUT NON RESETABLE BITS
1853 006534 005777 012150      TST      @RXCSR      ;CLR OUT DSC BY READING RXCSR
1854 006540 152777 000001 012144      BISB     #BIT0,@RXCSR      ;SET STRIP SYNC UPPER BYTE
1855 006546 017701 012136      MOV      @RXCSR,R1      ;SAVE RXCSR
1856 006552 022701 000400      CMP      #400,R1 ;TEST RXCSR
1857 006556 001401                BEQ      .+4
1858 006560 104000                HLT      ;ONLY STRIP SYNC SHOULD BE SET
1859 006562 105077 012122      CLRB      @RXCSR      ;CLR LOWER BYTE
1860 006566 017701 012116      MOV      @RXCSR,R1      ;SAVE RXCSR
1861 006572 022701 000400      CMP      #400,R1 ;TEST RXCSR
1862 006576 001401                BEQ      .+4
1863 006600 104000                HLT      ;ONLY STRIP SYNC SHOULD BE SET
1864 006602 052777 000400 012114      BIS      #MRESET,@TXCSR ;MASTER RESET
1865 006610 152777 000040 012110      BISB     #BIT5,@TXCSR      ;SET MAINT CLK UPPER BYTE
1866 006616 017701 012102      MOV      @TXCSR,R1      ;SAVE TXCSR
1867 006622 042701 002000      BIC      #BITW,R1      ;CLR BIT WINDOW (DEPENDENT
1868                                ;ON H315 CONNECTOR EXISTANCE)
1869 006626 022701 020200      CMP      #20200,R1      ;TEST TXCSR
1870 006632 001401                BEQ      .+4
1871 006634 104000                HLT      ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
1872 006636 105077 012062      CLRB      @TXCSR      ;CLR LOWER BYTE
1873 006642 017701 012056      MOV      @TXCSR,R1      ;SAVE TXCSR
1874 006646 042701 002000      BIC      #BITW,R1      ;CLR BIT WINDOW (DITTO)
1875 006652 022701 020200      CMP      #20200,R1      ;TEST TXCSR
1876 006656 001401                BEQ      .+4
1877 006660 104000                HLT      ;ONLY MAINT CLK BIT & TXDONE SHOULD BE SET
1878 006662 104400                SCOPE
1879                                ;;THIS TEST PERFORMS MASTER RESET TESTING &
1880                                ;;TESTING OF READ ONLY BIT BITW
1881                                ;;MAINT INTERNAL
1882                                ;;
1883 006664 012767 000032 172234      TST26: MOV      #26,TSTNO      ;SAVE THIS
1884 006672 012767 007016 172216      MOV      #TST27,NEXT      ;GO TO THIS TEST WHEN THRU
1885 006700 012777 044001 012016      MOV      #MINT!MTDATA.BREAK,@TXCSR ;SET MAINT INT.,BREAK,
1886                                ;&MTDATA
1887 006706 032777 002000 012010      BIT      #BITW,@TXCSR      ;TEST BITW
1888 006714 001001                BNE      .
1889 006716 104000                HLT      ;BIT WINDOW SHOULD BE SFT

```

```

1890 006720 042777 040000 011776      BIC      #MTDATA,@TXCSR
1891 006726 013702 001136              MOV      @#HOLD,R2
1892 006732 005302              DEC      R2
1893 006734 001376              BNE      1$
1894 006736 032777 002000 011760      BIT      #BITW,@TXCSR
1895 006744 001401              BEQ      .+4
1896 006746 104000              HLT      ;BIT SHOULD BE CLR
1897              ;NOW SET THE MTDATA
1898 006750 052777 040000 011746      BIS      #MTDATA,@TXCSR
1899 006756 052777 000400 011740      BIS      #MRESET,@TXCSR ;MASTER RESET
1900 006764 052777 004001 011732      BIS      #MINT!BREAK,@TXCSR
1901 006772 013702 001136              MOV      @#HOLD,R2
1902 006776 005302              DEC      R2
1903 007000 001376              BNE      2$
1904 007002 032777 002000 011714      BIT      #BITW,@TXCSR
1905 007010 001401              BEQ      .+4
1906 007012 104000              HLT      ;BITW SHOULD BE CLR BY MASTER RESET
1907 007014 104400              SCOPE
1908              ::THIS TEST PERFORMS MASTER RESET TESTING &
1909              ::TESTING OF READ ONLY BIT BITW
1910              ::MAINT EXTERNAL
1911              ::
1912 007016 012767 000033 172102 TST27: MOV      #27,TSTNO ;SAVE THIS
1913 007024 012767 007156 172064      MOV      #TST28,NEXT ;GO TO THIS TEST WHEN THRU
1914              ;TEST TO SEE IF EXTERNAL MODEM BYPASS CONNECTOR
1915              ;IS ON (H315)....IF 'NO' JUMP AROUND TEST
1916 007032 105767 172145      TSTB     JMRBY
1917 007036 100046              BPL      1$ ;IT IS NOT ON
1918 007040 012777 050001 011656      MOV      #MEXT!MTDATA.BREAK,@TXCSR ;SET MAINT EXT.,BREAK,
1919              ;&MTDATA
1920 007046 032777 002000 011650      BIT      #BITW,@TXCSR ;TEST BITW
1921 007054 001001              BNE      .+4
1922 007056 104000              HLT      ;BIT WINDOW SHOULD BE SET
1923 007060 042777 040000 011636      BIC      #MTDATA,@TXCSR
1924 007066 013702 001136              MOV      @#HOLD,R2
1925 007072 005302              DEC      R2
1926 007074 001376              BNE      2$
1927 007076 032777 002000 011620      BIT      #BITW,@TXCSR
1928 007104 001401              BEQ      .+4
1929 007106 104000              HLT      ;BIT SHOULD BE CLR
1930              ;NOW SET THE MTDATA
1931 007110 052777 040000 011606      BIS      #MTDATA,@TXCSR
1932 007116 052777 000400 011600      BIS      #MRESET,@TXCSR ;MASTER RESET
1933 007124 052777 010001 011572      BIS      #MEXT!BREAK,@TXCSR
1934 007132 013702 001136              MOV      @#HOLD,R2
1935 007136 005302              DEC      R2
1936 007140 001376              BNE      3$
1937 007142 032777 002000 011554      BIT      #BITW,@TXCSR
1938 007150 001401              BEQ      .+4
1939 007152 104000              HLT      ;BITW SHOULD BE CLR BY MASTER RESET
1940 007154 104400              SCOPE
1941              1$:
1942              ::THIS TEST PERFORMS MASTER RESET TESTING &
1943              ::TESTING OF READ ONLY BIT RXDONE
1944              ::
1945

```

```

1946 007156 012767 000034 171742 TST28: MOV #28,TSTNO ;SAVE THIS
1947 007164 012767 007214 171724 MOV #TST29,NEXT ;GO TO THIS TEST WHEN THRU
1948 007172 02777 000400 011524 BIS #MRESET,@TXCSR ;MASTER RESET
1949 007200 032777 000200 011502 BIT #RXDONE,@RXCSR ;TEST THIS BIT
1950 007206 001401 BEQ 64$ ;BR IF '0'
1951 007210 104000 HLT ;CHECK MASTER RESET LOGIC
1952 007212 64$: ;OR SHORT ON THIS BIT
1953 SCOPE
1954 007212 104400 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1955 ;:TESTING OF READ ONLY BIT RECACT
1956
1957
1958 007214 012767 000035 171704 TST29: MOV #29,TSTNO ;SAVE THIS
1959 007222 012767 007252 171666 MOV #TST30,NEXT ;GO TO THIS TEST WHEN THRU
1960 007230 052777 000400 011466 BIS #MRESET,@TXCSR ;MASTER RESET
1961 007236 032777 004000 011444 BIT #RECACT,@RXCSR ;TEST THIS BIT
1962 007244 001401 BEQ 64$ ;BR IF '0'
1963 007246 104000 HLT ;CHECK MASTER RESET LOGIC
1964 007250 64$: ;OR SHORT ON THIS BIT
1965 SCOPE
1966 007250 104400 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1967 ;:TESTING OF READ ONLY BIT DSC
1968
1969
1970 007252 012767 000036 171646 TST30: MOV #30,TSTNO ;SAVE THIS
1971 007260 012767 007310 171630 MOV #TST31,NEXT ;GO TO THIS TEST WHEN THRU
1972 007266 052777 000400 011430 BIS #MRESET,@TXCSR ;MASTER RESET
1973 007274 032777 100000 011406 BIT #DSC,@RXCSR ;TEST THIS BIT
1974 007302 001401 BEQ 64$ ;BR IF '0'
1975 007304 104000 HLT ;CHECK MASTER RESET LOGIC
1976 007306 64$: ;OR SHORT ON THIS BIT
1977 SCOPE
1978 007306 104400 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1979 ;:TESTING OF READ ONLY BIT TXDONE
1980
1981
1982 007310 012767 000037 171610 TST31: MOV #31,TSTNO ;SAVE THIS
1983 007316 012767 007346 171572 MOV #TST32,NEXT ;GO TO THIS TEST WHEN THRU
1984 007324 052777 000400 011372 BIS #MRESET,@TXCSR ;MASTER RESET
1985 007332 032777 000200 011364 BIT #TXDONE,@TXCSR ;TEST THIS BIT
1986 007340 001001 BNE .+4 ;BR IF '1'
1987 007342 104000 HLT ;CHECK MASTER RESET LOGIC
1988 ;OR SHORT ON THIS BIT
1989 SCOPE
1990 007344 104400 ;:THIS TEST PERFORMS MASTER RESET TESTING &
1991 ;:TESTING OF READ ONLY BIT DNA
1992
1993
1993 007346 012767 000040 171552 TST32: MOV #32,TSTNO ;SAVE THIS
1994 007354 012767 007404 171534 MOV #TST33,NEXT ;GO TO THIS TEST WHEN THRU
1995 007362 052777 000400 011334 BIS #MRESET,@TXCSR ;MASTER RESET
1996 007370 032777 100000 011326 BIT #DNA,@TXCSR ;TEST THIS BIT
1997 007376 001401 BEQ 64$ ;BR IF '0'
1998 007400 104000 HLT ;CHECK MASTER RESET LOGIC
1999 007402 64$: ;OR SHORT ON THIS BIT
2000 SCOPE
2001 007402 104400

```

```

2002                                     ::THIS TEST PERFORMS MASTER RESET TESTING &
2003                                     ::TESTING OF READ ONLY WORD RECEIVE DATA
2004                                     ::
2005 007404 012767 000041 171514 TST33: MOV #33,TSTNO ;SAVE THIS
2006 007412 012767 007452 171476 MOV #TST34,NEXT ;GO TO THIS TEST WHEN THRU
2007 007420 052777 000400 011276 BIS #MRESET,@TXCSR ;MASTER RESET
2008 007426 016703 011262 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
2009 007432 012700 000377 MOV #377,R0 ;EXPECTED
2010 007436 017701 011252 MOV @RXDBUF,R1 ;ACTUAL
2011 007442 120001 CMPB R0,R1
2012 007444 001401 BEQ .+4 ;BR IF '0'
2013 007446 104002 HLT 2 ;REC DATA SHOULD BE ALL 1'S
2014 007450 104400 SCOPE
2015                                     ::THIS TEST PERFORMS MASTER RESET TESTING &
2016                                     ::TESTING OF READ ONLY BIT PARER
2017                                     ::
2018 007452 012767 000042 171446 TST34: MOV #34,TSTNO ;SAVE THIS
2019 007460 012767 007510 171430 MOV #TST35,NEXT ;GO TO THIS TEST WHEN THRU
2020 007466 052777 000400 011230 BIS #MRESET,@TXCSR ;MASTER RESET
2021 007474 032777 010000 011212 BIT #PARER,@RXDBUF ;TEST THIS BIT
2022 007502 001401 BEQ 64$ ;BR IF '0'
2023 007504 104000 HLT ;CHECK MASTER RESET LOGIC
2024 007506 64$:
2025                                     ;OR SHORT ON THIS BIT
2026 007506 104400 SCOPE
2027                                     ::THIS TEST PERFORMS MASTER RESET TESTING &
2028                                     ::TESTING OF READ ONLY BIT FRMERR
2029                                     ::
2030 007510 012767 000043 171410 TST35: MOV #35,TSTNO ;SAVE THIS
2031 007516 012767 007546 171372 MOV #TST36,NEXT ;GO TO THIS TEST WHEN THRU
2032 007524 052777 000400 011172 BIS #MRESET,@TXCSR ;MASTER RESET
2033 007532 032777 020000 011154 BIT #FRMERR,@RXDBUF ;TEST THIS BIT
2034 007540 001401 BEQ 64$ ;BR IF '0'
2035 007542 104000 HLT ;CHECK MASTER RESET LOGIC
2036 007544 64$:
2037                                     ;OR SHORT ON THIS BIT
2038 007544 104400 SCOPE
2039                                     ::THIS TEST PERFORMS MASTER RESET TESTING &
2040                                     ::TESTING OF READ ONLY BIT OVRUN
2041                                     ::
2042 007546 012767 000044 171352 TST36: MOV #36,TSTNO ;SAVE THIS
2043 007554 012767 007604 171334 MOV #TST37,NEXT ;GO TO THIS TEST WHEN THRU
2044 007562 052777 000400 011134 BIS #MRESET,@TXCSR ;MASTER RESET
2045 007570 032777 040000 011116 BIT #OVRUN,@RXDBUF ;TEST THIS BIT
2046 007576 001401 BEQ 64$ ;BR IF '0'
2047 007600 104000 HLT ;CHECK MASTER RESET LOGIC
2048 007602 64$:
2049                                     ;OR SHORT ON THIS BIT
2050 007602 104400 SCOPE
2051                                     ::THIS TEST PERFORMS MASTER RESET TESTING &
2052                                     ::TESTING OF READ ONLY BIT RXERR
2053                                     ::
2054 007604 012767 000045 171314 TST37: MOV #37,TSTNO ;SAVE THIS
2055 007612 012767 007642 171276 MOV #TST38,NEXT ;GO TO THIS TEST WHEN THRU
2056 007620 052777 000400 011076 BIS #MRESET,@TXCSR ;MASTER RESET
2057 007626 032777 100000 011060 BIT #RXERR,@RXDBUF ;TEST THIS BIT

```

```

2058 007634 001401      BEQ      64$      ;BR IF '0'
2059 007636 104000      HLT          ;CHECK MASTER RESET LOGIC
2060 007640              64$:          ;OR SHORT ON THIS BIT
2061                                ;
2062 007640 104400      SCOPE
2063      ::THIS TEST VERIFYS THAT THE DEVICE REGISTER RXCSR
2064      ::IS CLEARED BY MASTER RESET
2065 007642 012767 000046 171256 TST38: MOV      #38,TSTNO      ;SAVE THIS
2066 007650 012767 007750 171240      MOV      #TST39,NEXT      ;GO TO THIS TEST WHEN THRU
2067 007656 012777 177777 011024      MOV      #177777,@RXCSR ;SET ALL POSSIBLE BITS
2068 007664 052777 000400 011032      BIS      #MRESET,@TXCSR ;MASTER RESET
2069 007672 016703 011012      MOV      RXCSR,R3      ;FOR ERROR MESSAGE
2070 007676 017701 011006      MOV      @RXCSR,R1      ;SAVE ACTUAL
2071 007702 105767 171273      TSTB     OPTCLR      ;TEST THE OPT CLR JUMPER FLAG
2072 007706 100010      BPL      1$      ;NO ,ITS NOT IN
2073 007710 042701 173000      BIC      #173000,R1      ;CLR NON-MASTER RESETTABLE
2074      ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
2075 007714 012700 000000      MOV      #0,R0      ;EXPECTED
2076 007720 020001      CMP      R0,R1      ;EXPECTED VS ACTUAL
2077 007722 001401      BEQ      .+4
2078 007724 104001      HLT      1      ;ALL MASTER RESETABLE BITS SHOULD BE CLR
2079 007726 000407      BR      2$      ;JUMP AROUND
2080 007730 042701 073000      BIC      #73000,R1      ;CLR NON-MASTER RESETABLE
2081      ;BITS(SINCE THESE ARE DEPENDENT ON H315 CONNECTORS EXISTANCE)
2082 007734 012700 000016      MOV      #16,R0      ;EXPECTED
2083 007740 020001      CMP      R0,R1      ;EXPECTED VS ACTUAL
2084 007742 001401      BEQ      .+4
2085 007744 104001      HLT      1      ;ONLY STD,RTS,DTR BITS SHOULD BE SET
2086      ;NOTE THAT STD IS READ =1 INDEPENDENT OF
2087      ;SEC XMIT #6 STRAP
2088 007746 104400      2$:      SCOPE
2089
2090      ::THIS TEST VERIFYS THAT THE DEVICE REGISTER TXCSR
2091      ::IS CLEARED BY MASTER RESET
2092
2093 007750 012767 000047 171150 TST39: MOV      #39,TSTNO      ;SAVE THIS
2094 007756 012767 010024 171132      MOV      #TST40,NEXT      ;GO TO THIS TEST WHEN THRU
2095 007764 012777 177777 010732      MOV      #177777,@TXCSR ;SET ALL POSSIBLE BITS
2096 007772 052777 000400 010724      BIS      #MRESET,@TXCSR ;MASTER RESET
2097 010000 016703 010720      MOV      TXCSR,R3      ;FOR ERROR MESSAGE
2098 010004 017701 010714      MOV      @TXCSR,R1      ;SAVE ACTUAL
2099 010010 012700 000200      MOV      #200,R0      ;EXPECTED
2100 010014 020001      CMP      R0,R1      ;EXPECTED VS ACTUAL
2101 010016 001401      BEQ      .+4
2102 010020 104001      HLT      1      ;ONLY TXDONE SHOULD BE SET
2103 010022 104400      SCOPE
2104
2105      ::THIS TEST VERIFYS THAT THE DEVICE REGISTER RXDBUF
2106      ::IS CLEARED BY MASTER RESET
2107
2108 010024 012767 000050 171074 TST40: MOV      #40,TSTNO      ;SAVE THIS
2109 010032 012767 010072 171056      MOV      #TST41,NEXT      ;GO TO THIS TEST WHEN THRU
2110 010040 052777 000400 010656      BIS      #MRESET,@TXCSR ;MASTER RESET
2111 010046 016703 010642      MOV      RXDBUF,R3      ;FOR ERROR MESSAGE
2112 010052 017701 010636      MOV      @RXDBUF,R1      ;SAVE
2113 010056 012700 000377      MOV      #377,R0      ;EXPECTED
  
```

```

CMP      R0,R1      ;EXPECTED VS ACTUAL
BEQ      +4
HLT      2           ;ONLY REC DATA BITS SHOULD BE SET
SCOPE
;;THIS TEST VERIFYS BITS RING,CTS,CARDET,SRD,DSR
;;ALSO DSC IS GENERATED WHEN ANY OF THESE BITS ARE SET
;;OR CLEARED.....IT ALSO CHECKS THE MODEM BYPASS
;;JUMPER AND THAT THESE BITS CAN BE READ
;;NOTE: THE MODEM BYPASS JUMPER MUST BE ON (H315)
;;
MOV      #41,TSTNO   ;SAVE THIS
MOV      #TST42,NEXT ;GO TO THIS TEST WHEN THRU
CLR      @RXCSR      ;TO GET RID OF STD ,RTS,DTR IF OPTCLR JUMPER #4 IS NOT ON
BIS      #MRESET,@TXCSR ;MASTER RESET
;TEST THAT A 'YES' ANSWER WAS GIVEN TO QUESTION IN
;THE MONITOR OR BY DEFAULT
;THIS TEST WILL BE BYPASSED IF THE EXTERNAL BYPASS
;JUMPER IS NOT INSTALLED
TSTB     JMRBY
BMI      +6         ;THE ANSWER WAS YES.....
;PERFORM THIS TEST
JMP      OUT1       ;JUMP AROUND THIS TEST IF THE ANSWER
;WAS NO
MOV      RXCSR,R3    ;SET UP FOR ERROR MESSAGE
MOV      @RXCSR,R1   ;ACTUAL
CLR      R0          ;EXPECTED
TST      R1          ;IS IT = 0 ?
BEQ      +4
HLT      1           ;RXCSR SHOULD BE CLR
BIS      #DTR,@RXCSR ;SET DTR
;WAIT FOR CABLE DELAYS
;*****
;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
;*****
MOV      HOLD,R2     ;SET DELAY TIME
DEC      R2
BNE      64$         ;WAIT THIS TIME
;OK NOW FALL THRU AND CONTINUE TESTING.....
;EXIT STAGE LEFT....CHINING!
MOV      @RXCSR,R1   ;ACTUAL
MOV      #130002,R0  ;DSC,CTS,CARDET,DTR
CMP      R0,R1      ;EXPECTED VS ACTUAL
BEQ      +4
HLT      1           ;CHECK BYPASS CONNECTOR
MOV      @RXCSR,R1   ;ACTUAL
MOV      #30002,R0   ;CTS,CARDET,DTR
CMP      R0,R1      ;EXPECTED VS ACTUAL
BEQ      +4
HLT      1           ;PREVIOUS READING OF RXCSR SHOULD
;HAVE CLEARED DSC
BIS      #RTS,@RXCSR
;WAIT FOR CABLE DELAYS
;*****
;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
;*****

```

2170	010232	016702	170700		MOV	HOLD,R2 ;SET DELAY TIME
2171	010236			65\$:	DEC	R2
2172	010236	005302			BNE	65\$;WAIT THIS TIME
2173	010240	001376				;OK NOW FALL THRU AND CONTINUE TESTING.....
2174						;EXIT STAGE LEFT....CHINNING!
2175					MOV	@RXCSR,R1
2176	010242	017701	010442		MOV	#170006,R0 ;DSC,RING,CTS,CARDET,RTS,DTR
2177	010246	012700	170006		CMP	R0,R1 ;EXPECTED VS ACTUAL
2178	010252	020001			BEQ	.+4
2179	010254	001401			HLT	1 ;CHECK BYPASS CONNECTOR
2180	010256	104001			MOV	@RXCSR,R1
2181	010260	017701	010424		MOV	#70006,R0 ;RING,CTS,CARDET,RTS,DTR
2182	010264	012700	070006		CMP	R0,R1 ;EXPECTED VS ACTUAL
2183	010270	020001			BEQ	.+4
2184	010272	001401			HLT	1 ;PREVIOUS READING OF RXCSR SHOULD
2185	010274	104001				;HAVE CLEARED DSC
2186					TSTB	SEXMIT ;IS SEC XMIT JUMPER IN ?
2187	010276	105767	170675		BPL	OUT2 ;NO
2188	010302	100112			TSTB	SEREC ;IS SEC REC JUMPER IN ?
2189	010304	105767	170670		BPL	OUT3 ;NO
2190	010310	100163			BIS	#STD,@RXCSR
2191	010312	052777	000010	010370		;WAIT FOR CABLE DELAYS
2192						*****
2193						;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2194						*****
2195					MOV	HOLD,R2 ;SET DELAY TIME
2196	010320	016702	170612		66\$:	DEC
2197	010324				BNE	66\$;WAIT THIS TIME
2198	010324	005302				;OK NOW FALL THRU AND CONTINUE TESTING.....
2199	010326	001376				;EXIT STAGE LEFT....CHINNING!
2200					MOV	@RXCSR,R1
2201					MOV	#173016,R0 ;DSC,RING,CTS,CARDET,SRD,DSR
2202	010330	017701	010354			;STD,RTS,DTR
2203	010334	012700	173016	MASK1:	CMP	R0,R1 ;EXPECTED VS ACTUAL
2204					BEQ	.+4
2205	010340	020001			HLT	1 ;CHECK BYPASS CONNECTOR
2206	010342	001401			MOV	@RXCSR,R1
2207	010344	104001			MOV	#73016,R0 ;RING,CTS,CARDET,SRD,DSR,STD
2208	010346	017701	010336	MASK2:		;RTS,DTR
2209	010352	012700	073016		CMP	R0,R1 ;EXPECTED VS ACTUAL
2210					BEQ	.+4
2211	010356	020001			HLT	1 ;PREVIOUS READING OF RXCSR SHOULD
2212	010360	001401				;HAVE CLEARED DSC
2213	010362	104001			BIC	#DTR,@RXCSR
2214						;WAIT FOR CABLE DELAYS
2215	010364	042777	000002	010316		*****
2216						;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2217						*****
2218					MOV	HOLD,R2 ;SET DELAY TIME
2219					64\$:	DEC
2220	010372	016702	170540		BNE	64\$;WAIT THIS TIME
2221	010376					;OK NOW FALL THRU AND CONTINUE TESTING.....
2222	010376	005302				;EXIT STAGE LEFT....CHINNING.
2223	010400	001376				
2224						
2225						

2226	010402	017701	010302		MOV	@RXCSR,R1	
2227	010406	012700	143014		MOV	#143014,R0	;DSC,RING,SRD,DSR,STD,RTS
2228	010412	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
2229	010414	001401			BEQ	.+4	
2230	010416	104001			HLT	1	;DSC SHOULD BE SET
2231	010420	042777	000004	010262	BIC	#RTS,@RXCSR	
2232							;WAIT FOR CABLE DELAYS
2233							*****
2234							;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2235							*****
2236	010426	016702	170504		MOV	HOLD,R2	;SET DELAY TIME
2237	010432			65\$:			
2238	010432	005302			DEC	R2	
2239	010434	001376			BNE	65\$	;WAIT THIS TIME
2240							;OK NOW FALL THRU AND CONTINUE TESTING.....
2241							;EXIT STAGE LEFT....CHINING!
2242	010436	017701	010246		MOV	@RXCSR,R1	
2243	010442	012700	103010	MASK3:	MOV	#103010,R0	;DSC,SRD,DSR,STD
2244	010446	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
2245	010450	001401			BEQ	.+4	
2246	010452	104001			HLT	1	;DSC SHOULD BE SET
2247	010454	042777	000010	010226	BIC	#STD,@RXCSR	
2248							;WAIT FOR CABLE DELAYS
2249							*****
2250							;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2251							*****
2252	010462	016702	170450		MOV	HOLD,R2	;SET DELAY TIME
2253	010466			64\$:			
2254	010466	005302			DEC	R2	
2255	010470	001376			BNE	64\$	;WAIT THIS TIME
2256							;OK NOW FALL THRU AND CONTINUE TESTING.....
2257							;EXIT STAGE LEFT....CHINING!
2258	010472	017701	010212		MOV	@RXCSR,R1	
2259	010476	012700	100000		MOV	#100000,R0	;DSC
2260	010502	020001			CMP	R0,R1	;EXPECTED VS ACTUAL
2261	010504	001401			BEQ	.+4	
2262	010506	104001			HLT	1	;DSC SHOULD BE SET
2263	010510	017701	010174		MOV	@RXCSR,R1	
2264	010514	005000			CLR	R0	;NONE
2265	010516	005701			TST	R1	
2266	010520	001401			BEQ	.+4	
2267	010522	104001			HLT	1	;DSC SHOULD BE CLEARED FROM PREVIOUS
2268							;READING OF RXCSR
2269	010524	000167	000254		JMP	OUT1	;JUMP AROUND
2270							;THE FOLLOWING ROUTINE HANDLES THE SITUATION WHERE SEC XMIT
2271							;AND SEC REC JUMPERS ARE NOT ON
2272	010530	052777	000010	010152	OUT2:	BIS	#STD,@RXCSR
2273							;WAIT FOR CABLE DELAYS
2274							*****
2275							;MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2276							*****
2277	010536	016702	170374		MOV	HOLD,R2	;SET DELAY TIME
2278	010542			64\$:			
2279	010542	005302			DEC	R2	
2280	010544	001376			BNE	64\$	;WAIT THIS TIME
2281							;OK NOW FALL THRU AND CONTINUE TESTING.....

Address	Hex	Hex	Hex	Hex	Label	Assembly
2282						;EXIT STAGE LEFT....CHINNING.
2283	010546	017701	010136			MOV @RXCSP,R1 ;ACTUAL
2284	010552	012700	070016			MOV #70016,R0 ;EXPECTED: RING ,CTS,CARDET,STD,RTS,DTR
2285	010556	020001				CMP R0,R1 ;EXPECTED VS ACTUAL
2286	010560	001401				BEQ .+4
2287	010562	104001				HLT 1 ;CHECK SEC XMIT & SEC REC JUMPERS
2288	010564	042777	000004	010116		BIC #RTS,@RXCSP
2289						;WAIT FOR CABLE DELAYS
2290						*****
2291						;MODIFY 'HOLD:'' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2292						*****
2293	010572	016702	170340			MOV HOLD,R2 ;SET DELAY TIME
2294	010576			65\$:		
2295	010576	005302				DEC R2
2296	010600	001376				BNE 65\$;WAIT THIS TIME
2297						;OK NOW FALL THRU AND CONTINUE TESTING.....
2298						;EXIT STAGE LEFT....CHINNING!
2299	010602	017701	010102			MOV @RXCSP,R1 ;ACTUAL
2300	010606	012700	130012			MOV #130012,R0 ;DSC,CTS,CARDET,DTR,STD
2301						;NOTE THAT DSC STILL ASSERTS EVEN THO THE SEC XMIT JUMPER # 6 IS NOT ON
2302	010612	020001				CMP R0,R1 ;EXPECTED VS ACTUAL
2303	010614	001401				BEQ .+4
2304	010616	104001				HLT 1 ;CHECK BYPASS CONNECTOR
2305	010620	042777	000002	010062		BIC #DTR,@RXCSP
2306						;WAIT FOR CABLE DELAYS
2307						*****
2308						;MODIFY 'HOLD:'' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2309						*****
2310	010626	016702	170304			MOV HOLD,R2 ;SET DELAY TIME
2311	010632			66\$:		
2312	010632	005302				DEC R2
2313	010634	001376				BNE 66\$;WAIT THIS TIME
2314						;OK NOW FALL THRU AND CONTINUE TESTING.....
2315						;EXIT STAGE LEFT....CHINNING!
2316	010636	017701	010046			MOV @RXCSP,R1 ;ACTUAL
2317	010642	012700	100010			MOV #100010,R0 ;DSC,STD
2318	010646	020001				CMP R0,R1 ;EXPECTED VS ACTUAL
2319	010650	001401				BEQ .+4
2320	010652	104001				HLT 1 ;ONLY DSC & STD SHOULD BE SET
2321	010654	000167	000124			JMP OUT1 ;JUMP AROUND
2322	010660	052777	000010	010022	OUT3:	BIS #STD,@RXCSP
2323						;WAIT FOR CABLE DELAYS
2324						*****
2325						;MODIFY 'HOLD:'' ACCORDINGLY FOR FASTER OR SLOWER MACHINE
2326						*****
2327	010666	016702	170244			MOV HOLD,R2 ;SET DELAY TIME
2328	010672			64\$:		
2329	010672	005302				DEC R2
2330	010674	001376				BNE 64\$;WAIT THIS TIME
2331						;OK NOW FALL THRU AND CONTINUE TESTING.....
2332						;EXIT STAGE LEFT....CHINNING!
2333	010676	017701	010006			MOV @RXCSP,R1 ;ACTUAL
2334	010702	012700	171016			MOV #171016,R0 ;EXPECTED: DSC,RING,CTS,CARDET,DSR,STD,RTS,DTR
2335	010706	020001				CMP R0,R1 ;EXPECTED VS ACTUAL
2336	010710	001401				BEQ .+4
2337	010712	104001				HLT 1 ;CHECK SEC REC JUMPER

2338	010714	042777	000004	007766		BIC	#RTS,@RXCSR	
2339						:	WAIT FOR CABLE DELAYS	
2340						:	*****	
2341						:	MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE	
2342						:	*****	
2343	010722	016702	170210			MOV	HOLD,R2 ;SET DELAY TIME	
2344	010726				65\$:			
2345	010726	005302				DEC	R2	
2346	010730	001376				BNE	65\$;WAIT THIS TIME	
2347						:	OK NOW FALL THRU AND CONTINUE TESTING.....	
2348						:	EXIT STAGE LEFT....CHINING!	
2349	010732	017701	007752			MOV	@RXCSR,R1 ;ACTUAL	
2350	010736	012700	131012			MOV	#131012,R0 ;EXPECTED: DSC,CTS,CARDET,DSR,STD,DTR	
2351	010742	020001				CMP	R0,R1 ;EXPECTED VS ACTUAL	
2352	010744	001401				BEQ	+.4	
2353	010746	104001				HLT	1 ;CHECK H315 CONNECTOR	
2354	010750	042777	000002	007732		BIC	#DTR,@RXCSR	
2355						:	WAIT FOR CABLE DELAYS	
2356						:	*****	
2357						:	MODIFY 'HOLD:' ACCORDINGLY FOR FASTER OR SLOWER MACHINE	
2358						:	*****	
2359	010756	016702	170154			MOV	HOLD,R2 ;SET DELAY TIME	
2360	010762				66\$:			
2361	010762	005302				DEC	R2	
2362	010764	001376				BNE	66\$;WAIT THIS TIME	
2363						:	OK NOW FALL THRU AND CONTINUE TESTING.....	
2364						:	EXIT STAGE LEFT....CHINING.	
2365	010766	017701	007716			MOV	@RXCSR,R1 ;ACTUAL	
2366	010772	012700	101010			MOV	#101010,R0 ;EXPECTED: DSC,DSR,STD	
2367	010776	020001				CMP	R0,R1 ;EXPECTED VS ACTUAL	
2368	011000	001401				BEQ	+.4	
2369	011002	104001				HLT	1 ;CHECK H315 CONNECTOR	
2370	011004	104400			OUT1:	SCOPE		
2371								
2372						:	THIS TEST VERIFYS THAT REACT (REC ACTIVE) ASSERTS	
2373						:	IMMED. WHEN SYNC EXTERNAL MODE IS SELECTED	
2374						:	AND SYNC SEARCH IS SET	
2375						:		
2376	011006	012767	000052	170112	TST42:	MOV	#42,TSTNO ;SAVE THIS	
2377	011014	012767	011134	170074		MOV	#TST43,NEXT ;GO TO THIS TEST WHEN THRU	
2378	011022	052777	000400	007674		BIS	#MRESET,@TXCSR ;MASTER RESET	
2379	011030	012777	020000	007662		MOV	#SYNEXT,@PARCSR ;SET THE MODE	
2380	011036	052777	000400	007660		BIS	#MRESET,@TXCSR ;MASTER RESET	
2381								
2382						:	SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE	
2383	011044	012777	064001	007652		MOV	#MTDATA!CLK!MINT!BREAK,@TXCSR	
2384								
2385						:	SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG	
2386	011052	012777	026026	007640		MOV	#SYNEXT!EIGHT!NOPAR!26,@PARCSR	
2387	011060	032777	004000	007622		BIT	#REACT,@RXCSR	
2388	011066	001401				BEQ	64\$	
2389	011070	104000				HLT	;REACT SHOULD NOT BE SET	
2390	011072				64\$:			
2391	011072	052777	000020	007610		BIS	#SYNSCH,@RXCSR ;SET SYNC SEARCH	
2392	011100	032777	004000	007602		BIT	#REACT,@RXCSR	
2393	011106	001001				BNF	65\$	

```

2394 011110 104000
2395 011112
2396 011112 042777 000020 007570
2397 011120 032777 004000 007562
2398 011126 001401
2399 011130 104000
2400 011132
2401 011132 104400
2402
2403
2404
2405
2406 011134 012767 000053 167764 TST43: MOV #43,TSTNO ;SAVE THIS
2407 011142 012767 011262 167746 MOV #TST44,NEXT ;GO TO THIS TEST WHEN THRU
2408 011150 052777 000400 007546 BIS #MRESET,@TXCSR ;MASTER RESET
2409 011156 012777 000000 007534 MOV #ISYMOD,@PARCSR ;SET THE MODE
2410 011164 052777 000400 007532 BIS #MRESET,@TXCSR ;MASTER RESET
2411
2412 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2413 011172 012777 064001 007524 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2414
2415 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2416 011200 012777 006026 007512 MOV #ISYMOD!EIGHT!NOPAR!26,@PARCSR
2417 011206 032777 004000 007474 BIT #REACT,@RXCSR
2418 011214 001401 BEQ 64$
2419 011216 104000 HLT ;REACT SHOULD NOT BE SET
2420 011220
2421 011220 052777 000020 007462 64$: BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2422 011226 032777 004000 007454 BIT #REACT,@RXCSR
2423 011234 001001 BNE 65$
2424 011236 104000 HLT ;REACT DID NOT ASSERT
2425 011240
2426 011240 042777 000020 007442 65$: BIC #SYNSCH,@RXCSR ;DROP SEARCH SYNC
2427 011246 032777 004000 007434 BIT #REACT,@RXCSR ;IS IT =0?
2428 011254 001401 BEQ 66$
2429 011256 104000 HLT ;REACT SHOULD BE 0
2430 011260
2431 011260 104400 66$:
2432 SCOPE
2433 ;;VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2434 ;;IN TWO * SYNC CHARS THRU MAINT DATA BIT
2435 ;;WATCH THE REACT BIT
2436 ;;ON THE THIRD * CHARACTER IT SHOULD SET RXDONE
2437 ;;*: DEPENDENT ON MONITOR.....
2438 ;;IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2439 ;;TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2440 ;;ON THE SECOND CHARACTER
2441 ;;ALSO CHECK THIS CHARACTER IN RXDBUF
2442 ;;AND CHECK OPERATION OF SYNSCH
2443 ;;MODE: SYNC INTERNAL
2444 ;;LENGTH:FIVE
2445 011262 012767 000054 167636 TST44: MOV #44,TSTNO ;SAVE THIS
2446 011270 012767 011604 167620 MOV #TST45,NEXT ;GO TO THIS TEST WHEN THRU
2447 011276 052777 000400 007420 BIS #MRESET,@TXCSR ;MASTER RESET
2448 011304 012777 030000 007406 MOV #SYNINT,@PARCSR ;SET THE MODE
2449 011312 052777 000400 007404 BIS #MRESET,@TXCSR ;MASTER RESET

```

```

2450
2451
2452 011320 012777 064001 007376 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2453      MOV      #MTDATA,CLK!MINT.BREAK,@TXCSR
2454
2455 011326 012777 030026 007364 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2456 011334 016703 007354      MOV      #SYNINT.FIVE!NOPAR!26,@PARCSR
2457 011340 052777 000020 007342      MOV      RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
2458      BIS      #SYNSCH,@RXCSR      ;SET SYNC SEARCH
2459 011346 042777 020000 007350      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2460 011354 052777 020000 007342      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2461      BIS      #CLK,@TXCSR      ;POKE CLK UP
2462 011362 042777 020000 007334      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2463 011370 052777 020000 007326      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2464 011376 012767 000002 167536      BIS      #CLK,@TXCSR      ;POKE CLK UP
2465 011404 012767 000005 167526      MOV      #2,COUNT
2466 011412 012767 000026 167524      1$: MOV      #5,SHIFT      ;# OF SHIFTS
2467 011420 004767 006770      MOV      #26,TEMP1      ;SYNC CHARACTER
2468 011424 005367 167512      JSR      PC,RPOKE
2469 011430 001403      DEC      COUNT
2470      BEQ      2$
2471 011432 105767 167540      ;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2472 011436 100762      TSTB     SYNCNO
2473 011440 105777 007244      BMI      1$      ;TWO SYNC CHARS
2474 011444 100001      2$: TSTB     @RXCSR      ;CHECK REC DONE BIT
2475 011446 104000      BPL      64$
2476 011450      HLT
2477 011450 032777 004000 007232      64$: BIT      #REACT,@RXCSR
2478 011456 001001      BNE      65$
2479 011460 104000      HLT      ;REACT SHOULD BE ASSERTED
2480 011462
2481 011462 012767 000005 167450      65$: MOV      #5,SHIFT
2482 011470 012767 000021 167446      MOV      #21,TEMP1      ;ANY CHARACTER
2483 011476 004767 006712      JSR      PC,RPOKE
2484 011502 105777 007207      TSTB     @RXCSR      ;CHECK RXDONE
2485 011506 100401      BMI      66$
2486 011510 104000      HLT      ;RXDONE SHOULD BE ASSERTED
2487 011512
2488 011512 032777 004000 007170      66$: BIT      #REACT,@RXCSR
2489 011520 001001      BNE      67$
2490 011522 104000      HLT      ;REACT SHOULD STILL BE ASSERTED
2491 011524
2492 011524 042777 000020 007156      67$: BIC      #SYNSCH,@RXCSR      ;CLR SYNC SEARCH
2493 011532 032777 004000 007150      BIT      #REACT,@RXCSR      ;IT SHOULD DROP IMMEDIATELY
2494 011540 001401      BEQ      68$
2495 011542 104000      HLT      ;REACT SHOULD BE CLR
2496 011544
2497 011544 105777 007140      68$: TSTB     @RXCSR      ;RXDONE
2498 011550 100401      BMI      69$
2499 011552 104000      HLT      ;RXDONE SHOULD STILL BE ASSERTED
2500 011554
2501 011554 012700 000021      69$: MOV      #21,R0      ;EXPECTED DATA
2502 011560 017701 007130      MOV      @RXDBUF,R1      ;ACTUAL DATA
2503 011564 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2504 011566 001401      BEQ      70$
2505 011570 104002      HLT      2      ;DATA CHARS SHOULD COMPARE
  
```

2506	011572				70\$:	
2507	011572	105777	007112			TSTB @RXCSR ;CHECK RXDONE
2508	011576	100001				BPL 71\$
2509	011600	104000				HLT ;RXDONE SHOULD BE CLR FROM
2510	011602				71\$:	
2511						;PREVIOUS READING OF RXDBUF
2512	011602	104400				SCOPE
2513						::VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2514						::IN TWO * SYNC CHARS THRU MAINT DATA BIT
2515						::WATCH THE RECACT BIT
2516						::ON THE THIRD * CHARACTER IT SHOULD SET RXDONE
2517						::* DEPENDENT ON MONITOR.....
2518						::IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2519						::TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2520						::ON THE SECOND CHARACTER
2521						::ALSO CHECK THIS CHARACTER IN RXDBUF
2522						::AND CHECK OPERATION OF SYN SCH
2523						::MODE: SYNC INTERNAL
2524						::LENGTH: SIX
2525						::
2526	011604	012767	000055	167314	TST45:	MOV #45,TSTNO ;SAVE THIS
2527	011612	012767	012126	167276		MOV #TST46,NEXT ;GO TO THIS TEST WHEN THRU
2528	011620	052777	000400	007076		BIS #MRESET,@TXCSR ;MASTER RESET
2529	011626	012777	030000	007064		MOV #SYNINT,@PARCSR ;SET THE MODE
2530	011634	052777	000400	007062		BIS #MRESET,@TXCSR ;MASTER RESET
2531						
2532						;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2533	011642	012777	064001	007054		MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2534						
2535						;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2536	011650	012777	032026	007042		MOV #SYNINT!SIX!NOPAR!26,@PARCSR
2537	011656	016703	007032			MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2538	011662	052777	000020	007020		BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2539						;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2540	011670	042777	020000	007026		BIC #CLK,@TXCSR ;POKE CLK DOWN
2541	011676	052777	020000	007020		BIS #CLK,@TXCSR ;POKE CLK UP
2542						;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2543	011704	042777	020000	007012		BIC #CLK,@TXCSR ;POKE CLK DOWN
2544	011712	052777	020000	007004		BIS #CLK,@TXCSR ;POKE CLK UP
2545	011720	012767	000002	167214		MOV #2,COUNT
2546	011726	012767	000006	167204	1\$:	MOV #6,SHIFT ;# OF SHIFTS
2547	011734	012767	000026	167202		MOV #26,TEMP1 ;SYNC CHARACTER
2548	011742	004767	006446			JSR PC,RPOKE
2549	011746	005367	167170			DEC COUNT
2550	011752	001403				BEQ 2\$
2551						;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2552	011754	105767	167216			TSTB SYNCNO
2553	011760	100762				BMI 1\$;TWO SYNC CHARS
2554	011762	105777	006722		2\$:	TSTB @RXCSR ;CHECK REC DONE BIT
2555	011766	100001				BPL 64\$
2556	011770	104000				HLT ;RXDONE SHOULD NOT BE ASSERTED
2557	011772				64\$:	
2558	011772	032777	004000	006710		BIT #RECACT,@RXCSR
2559	012000	001001				BNE 65\$
2560	012002	104000				HLT ;RECACT SHOULD BE ASSERTED
2561	012004				65\$:	

2562	012004	012767	000006	167126	MOV	#6,SHIFT	
2563	012012	012767	000021	167124	MOV	#21,TEMP1	;ANY CHARACTER
2564	012020	004767	006370		JSR	PC,RPOKE	
2565	012024	105777	006660		TSTB	@RXCSR	;CHECK RXDONE
2566	012030	100401			BMI	66\$	
2567	012032	104000			HLT		;RXDONE SHOULD BE ASSERTED
2568	012034						
2569	012034	032777	004000	006646	66\$: BIT	#REACT,@RXCSR	
2570	012042	001001			BNE	67\$	
2571	012044	104000			HLT		;REACT SHOULD STILL BE ASSERTED
2572	012046						
2573	012046	042777	000020	006634	67\$: BIC	#SYNSCH,@RXCSR	;CLR SYNC SEARCH
2574	012054	032777	004000	006626	BIT	#REACT,@RXCSR	;IT SHOULD DROP IMMEDIATELY
2575	012062	001401			BEQ	68\$	
2576	012064	104000			HLT		;REACT SHOULD BE CLR
2577	012066						
2578	012066	105777	006616		68\$: TSTB	@RXCSR	;RXDONE
2579	012072	100401			BMI	69\$	
2580	012074	104000			HLT		;RXDONE SHOULD STILL BE ASSERTED
2581	012076						
2582	012076	012700	000021		69\$: MOV	#21,R0	;EXPECTED DATA
2583	012102	017701	006606		MOV	@RXDBUF,R1	;ACTUAL DATA
2584	012106	020001			CMP	R0,R1	;COMPARE EXP VS ACT
2585	012110	001401			BEQ	70\$	
2586	012112	104002			HLT	2	;DATA CHARS SHOULD COMPARE
2587	012114						
2588	012114	105777	006570		70\$: TSTB	@RXCSR	;CHECK RXDONE
2589	012120	100001			BPL	71\$	
2590	012122	104000			HLT		;RXDONE SHOULD BE CLR FROM
2591	012124				71\$:		
2592							;PREVIOUS READING OF RXDBUF
2593	012124	104400					
2594					SCOPE		
2595					:::VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING		
2596					:::IN TWO * SYNC CHARS THRU MAINT DATA BIT		
2597					:::WATCH THE REACT BIT		
2598					:::ON THE THIRD * CHARACTER IT SHOULD SET RXDONE		
2599					:::* DEPENDENT ON MONITOR.....		
2600					:::IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY		
2601					:::TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT		
2602					:::ON THE SECOND CHARACTER		
2603					:::ALSO CHECK THIS CHARACTER IN RXDBUF		
2604					:::AND CHECK OPERATION OF SYNSCH		
2605					:::MODE: SYNC INTERNAL		
2606					:::LENGTH:SEVEN		
2607	012126	012767	000056	166772	TST46: MOV	#46,TSTNO	;SAVE THIS
2608	012134	012767	012450	166754	MOV	#TST47,NEXT	;GO TO THIS TEST WHEN THRU
2609	012142	052777	000400	006554	BIS	#MRESET,@TXCSR	;MASTER RESET
2610	012150	012777	030000	006542	MOV	#S/NINT,@PARCSR	;SET THE MODE
2611	012156	052777	000400	006540	BIS	#MRESET,@TXCSR	;MASTER RESET
2612							
2613							
2614	012164	012777	064001	006532	;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE		
2615					MOV	#MTDATA!CLK.MINT.BREAK,@TXCSR	
2616							
2617	012172	012777	034026	006520	;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG		
					MOV	#SYNINT.SEVEN NOPAR.26,@PARCSR	

2618	012200	016703	006510		MOV	RXDBUF,R3	;SET UP FOR ERROR MESSAGE
2619	012204	052777	000020	006476	BIS	#SYNSCH,@RXCSR	;SET SYNC SEARCH
2620							;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2621	012212	042777	020000	006504	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2622	012220	052777	020000	006476	BIS	#CLK,@TXCSR	;POKE CLK UP
2623							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2624	012226	042777	020000	006470	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2625	012234	052777	020000	006462	BIS	#CLK,@TXCSR	;POKE CLK UP
2626	012242	012767	000002	166672	MOV	#2,COUNT	
2627	012250	012767	000007	166662	1\$: MOV	#7,SHIFT	;# OF SHIFTS
2628	012256	012767	000026	166660	MOV	#26,TEMP1	;SYNC CHARACTER
2629	012264	004767	006124		JSR	PC,RPOKE	
2630	012270	005367	166646		DEC	COUNT	
2631	012274	001403			BEQ	2\$	
2632							;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2633	012276	105767	166674		TSTB	SYNCNO	
2634	012302	100762			BMI	1\$	;TWO SYNC CHARS
2635	012304	105777	006400		2\$: TSTB	@RXCSR	;CHECK REC DONE BIT
2636	012310	100001			BPL	64\$	
2637	012312	104000			HLT		;RXDONE SHOULD NOT BE ASSERTED
2638	012314				64\$: BIT	#REACT,@RXCSR	
2639	012314	032777	004000	006366	BNE	65\$	
2640	012322	001001			HLT		;REACT SHOULD BE ASSERTED
2641	012324	104000			65\$: MOV	#7,SHIFT	
2642	012326				MOV	#21,TEMP1	;ANY CHARACTER
2643	012326	012767	000007	166604	JSR	PC,RPOKE	
2644	012334	012767	000021	166602	TSTB	@RXCSR	;CHECK RXDONE
2645	012342	004767	006046		BMI	66\$	
2646	012346	105777	006336		HLT		;RXDONE SHOULD BE ASSERTED
2647	012352	100401			66\$: BIT	#REACT,@RXCSR	
2648	012354	104000			BNE	67\$	
2649	012356				HLT		;REACT SHOULD STILL BE ASSERTED
2650	012356	032777	004000	006324	67\$: BIC	#SYNSCH,@RXCSR	;CLR SYNC SEARCH
2651	012364	001001			BIT	#REACT,@RXCSR	;IT SHOULD DROP IMMEDIATELY
2652	012366	104000			BEQ	68\$	
2653	012370				HLT		;REACT SHOULD BE CLR
2654	012370	042777	000020	006312	68\$: TSTB	@RXCSR	;RXDONE
2655	012376	032777	004000	006304	BMI	69\$	
2656	012404	001401			HLT		;RXDONE SHOULD STILL BE ASSERTED
2657	012406	104000			69\$: MOV	#21,R0	;EXPECTED DATA
2658	012410				MOV	@RXDBUF,R1	;ACTUAL DATA
2659	012410	105777	006274		CMP	R0,R1	;COMPARE EXP VS ACT
2660	012414	100401			BEQ	70\$	
2661	012416	104000			HLT	2	;DATA CHARS SHOULD COMPARE
2662	012420				70\$: TSTB	@RXCSR	;CHECK RXDONE
2663	012420	012700	000021		BPL	71\$	
2664	012424	017701	006264		HLT		;RXDONE SHOULD BE CLR FROM
2665	012430	020001			71\$:		;PREVIOUS READING OF RXDBUF
2666	012432	001401					
2667	012434	104002					
2668	012436						
2669	012436	105777	006246				
2670	012442	100001					
2671	012444	104000					
2672	012446						
2673							

2674	012446	104400			SCOPE
2675					:::VERIFY THE MATCH DETECT & DATA RDY FLAGS BY PUMPING
2676					:::IN TWO * SYNC CHARS THRU MAINT DATA BIT
2677					:::WATCH THE RECACT BIT
2678					:::ON THE THIRD * CHARACTER IT SHOULD SET RXDONE
2679					:::* DEPENDENT ON MONITOR.....
2680					:::IF ONE SYNC STRAP IS SELECTED THEN IT WILL ONLY
2681					:::TAKE ONE SYNC CHARACTER FOR RXDONE TO ASSERT
2682					:::ON THE SECOND CHARACTER
2683					:::ALSO CHECK THIS CHARACTER IN RXDBUF
2684					:::AND CHECK OPERATION OF SYNSCH
2685					:::MODE: SYNC INTERNAL
2686					:::LENGTH:EIGHT
2687					:::
2688	012450	012767	000057	166450	TST47: MOV #47,TSTNO ;SAVE THIS
2689	012456	012767	012772	166432	MOV #TST48,NEXT ;GO TO THIS TEST WHEN THRU
2690	012464	052777	000400	006232	BIS #MRESET,@TXCSR ;MASTER RESET
2691	012472	012777	030000	006220	MOV #SYNINT,@PARCSR ;SET THE MODE
2692	012500	052777	000400	006216	BIS #MRESET,@TXCSR ;MASTER RESET
2693					
2694					;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2695	012506	012777	064001	006210	MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2696					
2697					;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2698	012514	012777	036026	006176	MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2699	012522	016703	006166		MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2700	012526	052777	000020	006154	BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2701					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2702	012534	042777	020000	006162	BIC #CLK,@TXCSR ;POKE CLK DOWN
2703	012542	052777	020000	006154	BIS #CLK,@TXCSR ;POKE CLK UP
2704					;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2705	012550	042777	020000	006146	BIC #CLK,@TXCSR ;POKE CLK DOWN
2706	012556	052777	020000	006140	BIS #CLK,@TXCSR ;POKE CLK UP
2707	012564	012767	000002	166350	MOV #2,COUNT
2708	012572	012767	000010	166340	1\$: MOV #8,SHIFT ;# OF SHIFTS
2709	012600	012767	000026	166336	MOV #26,TEMP1 ;SYNC CHARACTER
2710	012606	004767	005602		JSR PC,RPOKE
2711	012612	005367	166324		DEC COUNT
2712	012616	001403			BEQ 2\$
2713					;TEST SYNCNO TO SEE HOW MANY SYNC CHARS WERE SELECTED
2714	012620	105767	166352		TSTB SYNCNO
2715	012624	100762			BMI 1\$;TWO SYNC CHARS
2716	012626	105777	006056		2\$: TSTB @RXCSR ;CHECK REC DONE BIT
2717	012632	100001			BPL 64\$
2718	012634	104000			HLT ;RXDONE SHOULD NOT BE ASSERTED
2719	012636				64\$:
2720	012636	032777	004000	006044	BIT #RECACT,@RXCSR
2721	012644	001001			BNE 65\$
2722	012646	104000			HLT ;RECACT SHOULD BE ASSERTED
2723	012650				65\$:
2724	012650	012767	000010	166262	MOV #8,SHIFT
2725	012656	012767	000021	166260	MOV #21,TEMP1 ;ANY CHARACTER
2726	012664	004767	005524		JSR PC,RPOKE
2727	012670	105777	006014		TSTB @RXCSR ;CHECK RXDONE
2728	012674	100401			BMI 6C\$
2729	012676	104000			HLT ;RXDONE SHOULD BE ASSERTED

```

2730 012700
2731 012700 032777 004000 006002 66$: BIT #REACT,@RXCSR
2732 012706 001001 BNE 67$
2733 012710 104000 HLT ;REACT SHOULD STILL BE ASSERTED
2734 012712
2735 012712 042777 000020 005770 67$: BIC #SYNSCH,@RXCSR ;CLR SYNC SEARCH
2736 012720 032777 004000 005762 BIT #REACT,@RXCSR ;IT SHOULD DROP IMMEDIATELY
2737 012726 001401 BEQ 68$
2738 012730 104000 HLT ;REACT SHOULD BE CLR
2739 012732
2740 012732 105777 005752 68$: TSTB @RXCSR ;RXDONE
2741 012736 100401 BMI 69$
2742 012740 104000 HLT ;RXDONE SHOULD STILL BE ASSERTED
2743 012742
2744 012742 012700 000021 69$: MOV #21,R0 ;EXPECTED DATA
2745 012746 017701 005742 MOV @RXDBUF,R1 ;ACTUAL DATA
2746 012752 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2747 012754 001401 BFQ 70$
2748 012756 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2749 012760
2750 012760 105777 005724 70$: TSTB @RXCSR ;CHECK RXDONE
2751 012764 100001 BPL 71$
2752 012766 104000 HLT ;RXDONE SHOULD BE CLR FROM
2753 012770 71$:
2754 ;PREVIOUS READING OF RXDBUF
2755 012770 104400 SCOPE
2756 ;:THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2757 ;:RECEIVER SECTION,IT USES THE ERROR FLAGS
2758 ;:TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2759 ;:(OVRUN,RXERR)
2760 ;:MODE:ISYMOD
2761 ;:LENGTH:FIVE
2762 ;:CHAR:25
2763 ;:
2764 012772 012767 000060 166126 TST48: MOV #48,TSTNO ;SAVE THIS
2765 013000 012767 013234 166110 MOV #TST49,NEXT ;GO TO THIS TEST WHEN THRU
2766 013006 052777 000400 005710 BIS #MRESET,@TXCSR ;MASTER RESET
2767 013014 012777 000000 005676 MOV #ISYMOD,@PARCSR ;SET THE MODE
2768 013022 052777 000400 005674 BIS #MRESET,@TXCSR ;MASTER RESET
2769
2770 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2771 013030 012777 064001 005666 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2772
2773 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2774 013036 012777 000000 005654 MOV #ISYMOD!FIVE!NOPAR!0,@PARCSR
2775 013044 052777 000020 005636 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2776 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2777 013052 042777 020000 005644 BIC #CLK,@TXCSR ;POKE CLK DOWN
2778 013060 052777 020000 005636 BIS #CLK,@TXCSR ;POKE CLK UP
2779 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2780 013066 042777 020000 005630 BIC #CLK,@TXCSR ;POKE CLK DOWN
2781 013074 052777 020000 005622 BIS #CLK,@TXCSR ;POKE CLK UP
2782 013102 016703 005606 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2783 013106 012700 000025 MOV #25,R0 ;EXPECTED
2784 013112 012767 000007 166020 MOV #7,SHIFT ;# OF SHIFTS
2785 013120 012767 000152 166016 MOV #152,TEMP1 ;DATA CHAR
  
```

2786	013126	004767	005262		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2787	013132	105777	005552		TSTB	@RXCSR	;RXDONE ?
2788	013136	100401			BMI	64\$	
2789	013140	104000			HLT		;RXDONE SHOULD BE SET
2790	013142			64\$:			
2791	013142	017701	005546		MOV	@RXDBUF,R1	;ACTUAL
2792	013146	020001			CMP	R0,R1	;COMPARE EXPECTED VS. ACTUAL
2793	013150	001401			BEQ	65\$	
2794	013152	104002			HLT	2	;RECEIVED DATA DID NOT MATCH
2795							;EXPECTED DATA - CHECK MAINT DATA
2796							;OR RECEIVER LOGIC
2797	013154			65\$:			
2798	013154	012767	000007	165756	MOV	#7,SHIFT	;# OF SHIFTS
2799	013162	012767	000152	165754	MOV	#152,TEMP1	;DATA CHAR
2800	013170	004767	005220		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2801							;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2802	013174	012767	000007	165736	MOV	#7,SHIFT	;# OF SHIFTS
2803	013202	012767	000152	165734	MOV	#152,TEMP1	;DATA CHAR
2804	013210	004767	005200		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2805	013214	012700	140025		MOV	#140000!25,R0	;EXPECTED DATA PLUS
2806							;RXERR & OVRUN
2807	013220	017701	005470		MOV	@RXDBUF,R1	;ACTUAL
2808	013224	020001			CMP	R0,R1	;COMPARE EXP VS. ACT
2809	013226	001401			BEQ	66\$	
2810	013230	104002			HLT	2	;SPECIFICALLY LOOK AT RXERR &
2811							;OVRUN BITS...THEY BOTH SHOULD BE SET
2812	013232			66\$:			
2813	013232	104400					
2814							SCOPE
2815							::THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2816							::RECEIVER SECTION,IT USES THE ERROR FLAGS
2817							::TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2818							::(OVRUN,RXERR)
2819							::MODE:ISYMOD
2820							::LENGTH:FIVE
2821							::CHAR:12
2822	013234	012767	000061	165664	TST49: MOV	#49,TSTNO	;SAVE THIS
2823	013242	012767	013476	165646	MOV	#TST50,NEXT	;GO TO THIS TEST WHEN THRU
2824	013250	052777	000400	005446	BIS	#MRESET,@TXCSR	;MASTER RESET
2825	013256	012777	000000	005434	MOV	#ISYMOD,@PARCSR	;SET THE MODE
2826	013264	052777	000400	005432	BIS	#MRESET,@TXCSR	;MASTER RESET
2827							
2828							;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2829	013272	012777	064001	005424	MOV	#MTDATA!CLK!MINT!BREAK,@TXCSR	
2830							
2831							;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2832	013300	012777	000000	005412	MOV	#ISYMOD!FIVE!NOPAR!0,@PARCSR	
2833	013306	052777	000020	005374	BIS	#SYNSCH,@RXCSR	;SET SYNC SEARCH
2834							;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2835	013314	042777	020000	005402	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2836	013322	052777	020000	005374	BIS	#CLK,@TXCSR	;POKE CLK UP
2837							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2838	013330	042777	020000	005366	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2839	013336	052777	020000	005360	BIS	#CLK,@TXCSR	;POKE CLK UP
2840	013344	016703	005344		MOV	R0,0BUF,R3	;SET UP FOR ERROR MESSAGE
2841	013350	012700	000012		MOV	#12,R0	;EXPECTED

```

2842 013354 012767 000007 165556      MOV    #7,SHIFT      ;# OF SHIFTS
2843 013362 012767 000124 165554      MOV    #124,TEMP1    ;DATA CHAR
2844 013370 004767 005020              JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
2845 013374 105777 005310              TSTB   @RXCSR ;RXDONE ?
2846 013400 100401                    BMI    64$
2847 013402 104000                    HLT
2848 013404                    64$:
2849 013404 017701 005304      MOV    @RXDBUF,R1      ;ACTUAL
2850 013410 020001      CMP    R0,R1      ;COMPARE EXPECTED VS. ACTUAL
2851 013412 001401      BEQ    65$
2852 013414 104002      HLT    2          ;RECEIVED DATA DID NOT MATCH
2853                                ;EXPECTED DATA - CHECK MAINT DATA
2854                                ;OR RECEIVER LOGIC
2855 013416                    65$:
2856 013416 012767 000007 165514      MOV    #7,SHIFT      ;# OF SHIFTS
2857 013424 012767 000124 165512      MOV    #124,TEMP1    ;DATA CHAR
2858 013432 004767 004756      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
2859                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2860 013436 012767 000007 165474      MOV    #7,SHIFT      ;# OF SHIFTS
2861 013444 012767 000124 165472      MOV    #124,TEMP1    ;DATA CHAR
2862 013452 004767 004736      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
2863 013456 012700 140012      MOV    #140000!12,R0      ;EXPECTED DATA PLUS
2864                                ;RXERR & OVRUN
2865 013462 017701 005226      MOV    @RXDBUF,R1      ;ACTUAL
2866 013466 020001      CMP    R0,R1      ;COMPARE EXP VS. ACT
2867 013470 001401      BEQ    66$
2868 013472 104002      HLT    2          ;SPECIFICALLY LOOK AT RXERR &
2869                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2870                    66$:
2871 013474 104400      SCOPE
2872      ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2873      ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2874      ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2875      ;;(OVRUN,RXERR)
2876      ;;MODE:ISYMOD
2877      ;;LENGTH:FIVE
2878      ;;CHAR:37
2879      ;;
2880 013476 012767 000062 165422  TST50: MOV    #50,TSTNO      ;SAVE THIS
2881 013504 012767 013740 165404      MOV    #TST51,NEXT      ;GO TO THIS TEST WHEN THRU
2882 013512 052777 000400 005204      BIS    #MRESET,@TXCSR ;MASTER RESET
2883 013520 012777 000000 005172      MOV    #ISYMOD,@PARCSR ;SET THE MODE
2884 013526 052777 000400 005170      BIS    #MRESET,@TXCSR ;MASTER RESET
2885
2886      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2887 013534 012777 064001 005162      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2888
2889      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2890 013542 012777 000000 005150      MOV    #ISYMOD!FIVE!NOPAR!0,@PARCSR
2891 013550 052777 000020 005132      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2892      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2893 013556 042777 020000 005140      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2894 013564 052777 020000 005132      BIS    #CLK,@TXCSR      ;POKE CLK UP
2895      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2896 013572 042777 020000 005124      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2897 013600 052777 020000 005116      BIS    #CLK,@TXCSR      ;POKE CLK UP

```

```

2898 013606 016703 005102      MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
2899 013612 012700 000037      MOV    #37,R0      ;EXPECTED
2900 013616 012767 000007 165314  MOV    #7,SHIFT    ;# OF SHIFTS
2901 013624 012767 000176 165312  MOV    #176,TEMP1  ;DATA CHAR
2902 013632 004767 004556      JSR    PC,RPOKE     ;SHIFT IN THIS CHAR
2903 013636 105777 005046      TSTB   @RXCSR      ;RXDONE ?
2904 013642 100401 64$      BMI     64$
2905 013644 104000      HLT     ;RXDONE SHOULD BE SET
2906 013646 64$:
2907 013646 017701 005042      MOV    @RXDBUF,R1   ;ACTUAL
2908 013652 020001      CMP     R0,R1      ;COMPARE EXPECTED VS. ACTUAL
2909 013654 001401      BEQ     65$
2910 013656 104002      HLT     2      ;RECEIVED DATA DID NOT MATCH
2911      ;EXPECTED DATA - CHECK MAINT DATA
2912      ;OR RECEIVER LOGIC
2913 013660 65$:
2914 013660 012767 000007 165252  MOV    #7,SHIFT    ;# OF SHIFTS
2915 013666 012767 000176 165250  MOV    #176,TEMP1  ;DATA CHAR
2916 013674 004767 004514      JSR    PC,RPOKE     ;SHIFT IN THIS CHAR
2917      ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2918 013700 012767 000007 165232  MOV    #7,SHIFT    ;# OF SHIFTS
2919 013706 012767 000176 165230  MOV    #176,TEMP1  ;DATA CHAR
2920 013714 004767 004474      JSR    PC,RPOKE     ;SHIFT IN THIS CHAR
2921 013720 012700 140037      MOV    #140000!37,R0 ;EXPECTED DATA PLUS
2922      ;RXERR & OVRUN
2923 013724 017701 004764      MOV    @RXDBUF,R1   ;ACTUAL
2924 013730 020001      CMP     R0,R1      ;COMPARE EXP VS. ACT
2925 013732 001401      BEQ     66$
2926 013734 104002      HLT     2      ;SPECIFICALLY LOOK AT RXERR &
2927      ;OVRUN BITS...THEY BOTH SHOULD BE SET
2928 013736 66$:
2929 013736 104400      SCOPE
2930      ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2931      ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
2932      ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2933      ;;(OVRUN,RXERR)
2934      ;;MODE:ISYMOD
2935      ;;LENGTH:FIVE
2936      ;;CHAR:0
2937      ;;
2938 013740 012767 000063 165160  TST51: MOV    #51,TSTNO    ;SAVE THIS
2939 013746 012767 014202 165142  MOV    #.EOP,NEXT    ;GO TO THIS TEST WHEN THRU
2940 013754 052777 000400 004742  BIS    #MRESET,@TXCSR ;MASTER RESET
2941 013762 012777 000000 004730  MOV    #ISYMOD,@PARCSR ;SET THE MODE
2942 013770 052777 000400 004726  BIS    #MRESET,@TXCSR ;MASTER RESET
2943
2944      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2945 013776 012777 064001 004720  MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2946
2947      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2948 014004 012777 000000 004706  MOV    #ISYMOD!FIVE!NOPAR!0,@PARCSR
2949 014012 052777 000020 004670  BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2950      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2951 014020 042777 020000 004676  BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2952 014026 052777 020000 004670  BIS    #CLK,@TXCSR   ;POKE CLK UP
2953      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION

```

2954	014034	042777	020000	004662	BIC	#CLK,@TXCSR	:POKE CLK DOWN
2955	014042	052777	020000	004654	BIS	#CLK,@TXCSR	:POKE CLK UP
2956	014050	016703	004640		MOV	RXDBUF,R3	:SET UP FOR ERROR MESSAGE
2957	014054	012700	000000		MOV	#0,R0	:EXPECTED
2958	014060	012767	000007	165052	MOV	#7,SHIFT	:# OF SHIFTS
2959	014066	012767	000100	165050	MOV	#100,TEMP1	:DATA CHAR
2960	014074	004767	004314		JSR	PC,RPOKE	:SHIFT IN THIS CHAR
2961	014100	105777	004604		TSTB	@RXCSR	:RXDONE ?
2962	014104	100401			BMI	64\$	
2963	014106	104000			HLT		:RXDONE SHOULD BE SET
2964	014110						64\$:
2965	014110	017701	004600		MOV	@RXDBUF,R1	:ACTUAL
2966	014114	020001			CMP	R0,R1	:COMPARE EXPECTED VS. ACTUAL
2967	014116	001401			BEQ	65\$	
2968	014120	104002			HLT	2	:RECEIVED DATA DID NOT MATCH
2969							:EXPECTED DATA - CHECK MAINT DATA
2970							:OR RECEIVER LOGIC
2971	014122						65\$:
2972	014122	012767	000007	165010	MOV	#7,SHIFT	:# OF SHIFTS
2973	014130	012767	000100	165006	MOV	#100,TEMP1	:DATA CHAR
2974	014136	004767	004252		JSR	PC,RPOKE	:SHIFT IN THIS CHAR
2975							:NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2976	014142	012767	000007	164770	MOV	#7,SHIFT	:# OF SHIFTS
2977	014150	012767	000100	164766	MOV	#100,TEMP1	:DATA CHAR
2978	014156	004767	004232		JSR	PC,RPOKE	:SHIFT IN THIS CHAR
2979	014162	012700	140000		MOV	#140000.0,R0	:EXPECTED DATA PLUS
2980							:RXERR & OVRUN
2981	014166	017701	004522		MOV	@RXDBUF,R1	:ACTUAL
2982	014172	020001			CMP	R0,R1	:COMPARE EXP VS. ACT
2983	014174	001401			BEQ	66\$	
2984	014176	104002			HLT	2	:SPECIFICALLY LOOK AT RXERR &
2985							:OVRUN BITS...THEY BOTH SHOULD BE SET
2986	014200						66\$:
2987	014200	104400					SCOPE
2988							

```

2989
2990
2991
2992
2993
2994
2995
2996 014202 104402
2997 014204 0 7344
2998 014206 104410 014440
2999 014212 104402 017065
3000 014216 105767 164760
3001 014222 001511
3002 014224 005767 164766
3003 014230 001007
3004 014232 104402 017077
3005 014236 016700 164754
3006 014242 000000
3007
3008 014244 000167 165150
3009 014250 062767 000010 164726 RUNIT:
3010 014256 062767 000010 164726 ZERO:
3011 014264 000241
3012 014266 006167 164726
3013 014272 103410
3014
3015 014274 036767 164720 164714
3016 014302 001762
3017 014304 004767 000034
3018 014310 000167 000174
3019 014314 012767 000001 164676 2$:
3020
3021 014322 016767 164660 164654
3022 014330 016767 164660 164654
3023 014336 004767 000002
3024 014342 000441
3025 014344 016767 164634 004040 REPLAY:
3026 014352 004767 003702
3027 014356 016767 164630 004350
3028 014364 062767 000002 164620
3029 014372 016767 164614 004336
3030 014400 062767 000002 164604
3031 014406 016767 164600 004324
3032 014414 062767 000002 164570
3033 014422 016767 164564 004312
3034 014430 016767 004300 164554
3035 014436 000207
3036
3037 014440 000001
3038 014442 006 002 OUTCRY: 1
3039 014444 020710 .BYTE 6,2
3040
3041 014446
3042 014446 005067 164462
3043 014452 005067 164570
3044 014456 005267 164446
  
```

```

;END OF PASS
;TYPE NAME OF TEST
;UPDATE PASS COUNT
;CHECK FOR EXIT TO ACT-11
;RESTART TEST
  
```

```

.EOP: TYPE ;TYPE NAME OF TEST
  
```

```

MEPASS
CONVRT ,OUTCRY
TYPE ,DEVICE
TSTB MULTD
BEQ CCC
IST ACTREG
BNE RUNIT
TYPE ,MCOW
MOV ACTREG,RO
HALT
  
```

```

;ARE YOU RUNNING MULTIPLE DEVICES ?
;NO,JUMP AROUND
;ARE ANY DEVICES ACTIVE ?
;YES
;NO
;DISPLAY ACTREG
;SELECT SOMETHING TO RUN @ ACTREG:
(PUT SW00 =1)
;START OVER AGAIN.....YOU DESELECTED EVERYTHING
;NEXT BLOCK (ADDRESSES)
;NEXT BLOCK (VECTORS)
  
```

```

;SELECT SWITCHES & HIT (CONTINUE)
JMP .START
ADD #10,BASEADD
ADD #10,BASEIV
CLC
ROL ROTADD
BCS 2$
  
```

```

;UP DATE ROTATING POINTER
;IS IT THE LAST DEVICE
;TO BE TESTED IN THIS PASS ?
;TEST THIS DEVICE FOR ACTIVE STATUS
;IF NOT ACTIVE, TRY NEXT ADDRESS
;CALCULATE NEW PARAMETERS
;YES IT WAS ACTIVE,TEST THIS DEVICE
;OK!,NOW SET UP ROTATING
;POINTER FOR NEXT MULTIPLE PASS
;RESTORE BASE ADDRESS
;RESTORE BASE INTERRUPT VECTORS
;CALC NEW PARAMETERS
;JUMP AROUND REPLAY
;SET UP FOR NEW ADDRESSES
;CREATE NEW ADDRESSES
;CREATE DURIV
;CREATE DURIS
;CREATE DUTIV
;CREATE DUTIS
;RESTORE
  
```

```

BIT ROTADD,ACTREG
BEQ RUNIT
JSR PC,REPLAY
JMP RESTRT
MOV #1,ROTADD
MOV KEEPADD,BASEADD
MOV KEEPIV,BASEIV
JSR PC,REPLAY
BR CCC
MOV BASEADD,DUBASE
JSR PC,DUADDR
MOV BASEIV,DURIV
ADD #2,BASEIV
MOV BASEIV,DURIS
ADD #2,BASEIV
MOV BASEIV,DUTIV
ADD #2,BASEIV
MOV BASEIV,DUTIS
MOV DURIV,BASEIV
RTS PC
  
```

```

OUTCRY: 1
.BYTE 6,2
RXCSR
  
```

```

CCC:
CLR LSTERR
CLR ER?FLG
INC PASCNT
;CLEAR LAST ERROR PC
;CLEAR ERROR FLAG
;UPDATE PASS COUNT
  
```

```

3045 014462 016777 164442 164412      MOV    PASCNT,@LIGHTS      ;DISPLAY PASS COUNT
3046 014470 013701 000042              MOV    @#42,R1          ;CHECK FOR ACT-11 OR DDP
3047 014474 001405                      BEQ    RESTRT          ;IF NOT, CONTINUE TESTING
3048 014476 000005                      RESET
3049 014500 004711                      LOGICAL: JSR    PC,(R1)
3050 014502 000240                      NOP
3051 014504 000240                      NOP
3052 014506 000240                      NOP
3053 014510 012767 000340 163260 RESTRT: MOV    #340,PS          ;PREVENT INTERRUPTS (PRIO: 7)
3054 014516 104413                      CKSWR          ;CHECK FOR ^G
3055 014520 012767 003446 164366      MOV    #TST1,RTRN
3056 014526 000167 166714              JMP     TST1
3057
3058                                ;SCOPE LOOP AND INTERATION HANDLER
3059
3060 014532                                .SCOPE:
3061                                ;**** START OF CODE FOR THE X OR TESTER ****
3062 014532 000424                      BR     4$
3063                                ;IF RUNNING ON THE X OR TESTER CHANGE
3064 014534 013746 000004              MOV    @#4,-(SP)          ;THIS INSTRUCTION TO A 'NOP'(NOP=240)
3065 014540 012737 014560 000004      MOV    #1$,@#4          ;SAVE CONTENTS OF ERROR VECTOR
3066 014546 005737 177060              TST    @#177060        ;SET FOR TIME OUT
3067 014552 012637 000004              MOV    (SP)+,@#4        ;TIME OUT ON X OR ?
3068 014556 000404                      BR     2$
3069 014560 022626                      1$: CMP    (SP)+,(SP)+      ;RESTORE ERROR VECTOR
3070 014562 012637 000004              MOV    (SP)+,@#4        ;GO TO NEXT TEST
3071 014566 000403                      BR     3$
3072 014570 016767 164322 164316      2$: MOV    NEXT,RTRN      ;CLEAR THE STACK AFTER A TIMEOUT
3073 014576 016716 164312              3$: MOV    RTRN,(SP)      ;RESTORE ERROR VECTOR
3074 014602 000002                      RTI          ;LOOP ON PRESENT TEST
3075 014604                                4$: ;**** END OF CODE FOR THE X OR TESTER ****
3076 014604 104413                      CKSWR          ;CHECK FOR ^G
3077 014606 032777 040000 164264      TTST: BIT    #SW14,@SWR      ;LOOP ON CURRENT TEST ?
3078 014614 001407                      BEQ    1$
3079 014616 000432                      BR     3$
3080 014620 105777 164260              TSTB    @TKCSR          ;TEST TTY FLAG
3081 014624 100027                      BPL     3$
3082 014626 017700 164254              MOV    @TKDBR,R0
3083 014632 000412                      BR     2$
3084 014634 032777 004000 164236      1$: BIT    #SW11,@SWR      ;CLR DONE BIT
3085 014642 001006                      BNE     2$
3086 014644 005267 164254              INC     LPCNT          ;IF A TTY KEY IS STRUCK GO TO NEXT TST
3087 014650 026767 164250 164244      CMP     LPCNT,ICOUNT    ;INHIBIT ITERATIONS ?
3088 014656 101412                      BLOS    3$
3089 014660 105067 164362              2$: CLRB    ERRFLG
3090 014664 005067 164234              CLR     LPCNT
3091 014670 012767 000005 164224      MOV     #5,ICOUNT      ;SET UP ITERATION COUNT
3092 014676 016767 164214 164210      MOV     NEXT,RTRN      ;SET UP NEXT TEST IN RTRN
3093 014704 016716 164204              3$: MOV     RTRN,(SP)    ;SET UP STACK FOR RTI
3094 014710 000002                      RTI
3095 014712 001407                      BRW:    1407
3096 014714 000432                      BRX:    432
3097                                ;RESTORE 'BEQ 1$' INSTRUCTION
3098                                ;RESTORE 'BR 3$' INSTRUCTION
3099                                ;CHECK FOR FREEZE ON CURRENT DATA
3100 014716 104413                      .SCOPE1: CKSWR          ;CHECK FOR ^G

```

3101	014720	032777	001000	164152	BIT	#SW09,@SWR	
3102	014726	001402			BEQ	1\$	
3103	014730	016716	164164		MOV	LOCK,(SP)	
3104	014734	000002			RTI		
3105							
3106							
3107							
3108	014736	010546					
3109	014740	017605	000002		.TYPE: MOV	R5,-(SP)	
3110	014744	062766	000002	000002	MOV	@2(SP),R5	
3111	014752	105715			ADD	#2,2(SP)	
3112	014754	001406			1\$: TSTB	(R5)	;LOOK FOR '0'
3113	014756	105777	164126		BEQ	3\$	
3114	014762	100375			2\$: TSTB	@TPCSR	;TEST DONE BIT
3115	014764	112577	164122		BPL	2\$	
3116	014770	000770			MOVB	(R5)+,@TPDBR	;TYPE CHAR
3117	014772	012605			BR	1\$	;DO IT AGAIN UNTIL '0' IS SEEN
3118	014774	000002			5\$: MOV	(SP)+,R5	
3119					RTI		
3120							
3121							
3122	014776	010346			.INSTR: MOV	R3,-(SP)	
3123	015000	010446			MOV	R4,-(SP)	
3124	015002	017667	000004	000010	MOV	@4(SP),.MSG	
3125	015010	062766	000002	000004	ADD	#2,4(SP)	
3126	015016	104402			.INST1: TYPE		
3127	015020	000000			.MSG: 0		
3128	015022	012704	020044		MOV	#INBUF,R4	
3129	015026	012703	000007		MOV	#7,R3	
3130	015032	105777	164046		1\$: TSTB	@TKCSR	
3131	015036	100375			BPL	1\$	
3132	015040	117714	164042		MOVB	@TKDBR,(R4)	
3133	015044	142714	000200		BICB	#200,(R4)	
3134	015050	121427	000025		CMPS	(R4),#25	
3135	015054	001003			BNE	200\$	;IS IT <^U>
3136	015056	104402	017254		TYPE,MCRLF		
3137	015062	000755			BR	.INST1	
3138	015064	122427	000015		200\$: CMPS	(R4)+,#15	
3139	015070	001423			BEQ	INSTR2	
3140	015072	117777	164010	164012	MOVB	@TKDBR,@TPDBR	
3141	015100	105777	164004		2\$: TSTB	@TPCSR	
3142	015104	100375			BPL	2\$	
3143	015106	005303			DEC	R3	
3144	015110	001350			BNE	1\$	
3145	015112	000402			BR	.INSTG	
3146	015114	010346			.INSTE: MOV	R3,-(SP)	
3147	015116	010446			MOV	R4,-(SP)	
3148	015120	104402			.INSTG: TYPE		
3149	015122	017250			MOVM		
3150	015124	005737	016412		TST	@WRDSW	
3151	015130	001402			BEQ	400\$	
3152	015132	104402	017254		TYPE,MCRLF		
3153	015136	000727			400\$: BR	.INST1	
3154	015140	012604			INSTR2: MOV	(SP)+,R4	
3155	015142	012603			MOV	(R4)+,R3	
3156	015144	000002			RTI		

```

3157
3158
3159                                ;CONVERT ASCII STRING TO OCTAL
3160 015146 010546 .PARAM: MOV R5, -(SP)
3161 015150 010446 MOV R4, -(SP)
3162 015152 016605 MOV 4(SP), R5
3163 015156 012567 000004 MOV (R5)+, LOLIM
3164 015162 012567 000170 MOV (R5)+, HILIM
3165 015166 012567 000166 MOV (R5)+, DEVADR
3166 015172 112567 000164 MOV (R5)+, LOBITS
3167 015176 112567 000157 MOV (R5)+, ADCNT
3168 015202 010566 000004 MOV R5, 4(SP)
3169 015206 005005 PARAM1: CLR R5
3170 015210 012704 MOV #INBUF, R4
3171 015214 122714 000015 CMPB #15, (R4)
3172 015220 001420 BEQ PARERR
3173 015222 121427 000060 $: CMPB (R4), #60
3174 015226 002415 BLT PARERR
3175 015230 121427 000067 CMPB (R4), #67
3176 015234 003012 BGT PARERR
3177 015236 142714 000060 BICB #60, (R4)
3178 015242 152405 BISB (R4)+, R5
3179 015244 122714 000015 CMPB #15, (R4)
3180 015250 001414 BEQ LIMITS
3181 015252 006305 ASL R5
3182 015254 006305 ASL R5
3183 015256 006305 ASL R5
3184 015260 000760 BR 1$
3185 015262 122714 000015 PARERR: CMPB #15, (R4) ;IS FIRST CHARACTER A <CR>
3186 015266 001003 BNE 120$
3187 015270 005737 016412 TST 2*RD SW ;IS CKSW ROUTINE BEING USED
3188 015274 001023 BNE PARTI
3189 015276 104404 120$: INSTER
3190 015300 000742 BR PARAM1
3191
3192                                ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3193
3194 015302 020567 000046 LIMITS: CMP R5, HILIM
3195 015306 101365 BHI PARERR
3196 015310 020567 000036 CMP R5, LOLIM
3197 015314 103762 BLO PARERR
3198 015316 136705 000036 BITB LOBITS, R5
3199 015322 001357 BNE PARERR
3200
3201                                ;STORE NUMBER AT SPECIFIED ADDRESS
3202
3203 015324 016704 000026 1$: MOV DEVADR, R4
3204 015330 010524 MOV R5, (R4)+
3205 015332 062705 000002 ADD #2, R5
3206 015336 105367 000017 DECB ADCNT
3207 015342 001372 BNE 1$
3208 015344 012604 PARTI: MOV (SP)+, R4
3209 015346 012605 MOV (SP)+, R5
3210 015350 000002 RTI
3211 015352 000000 CLIM: 0
3212 015354 000000 HILIM: 0

```

```

3213 015356 000000          DEVADR: 0
3214 015360 000000          LOBITS: 0
3215          015361          ADRCNT=LOBITS+1
3216
3217          ;SAVE PC OF TEST THAT FAILED AND R0-R5
3218
3219 015362 016667 000004 163604 .SAV05: MOV      4(SP),SAVPC
3220
3221          ;SAVE R0-R5
3222
3223 015370 010567 163574      SV05:  MOV      R5,SAVR5
3224 015374 010467 163566      MOV      R4,SAVR4
3225 015400 010367 163560      MOV      R3,SAVR3
3226 015404 010267 163552      MOV      R2,SAVR2
3227 015410 010167 163544      MOV      R1,SAVR1
3228 015414 010067 163536      MOV      R0,SAVR0
3229 015420 000002          RTI
3230
3231          ;RESTORE R0-R5
3232
3233 015422 016700 163530      .RES05: MOV      SAVR0,R0
3234 015426 016701 163526      MOV      SAVR1,R1
3235 015432 016702 163524      MOV      SAVR2,R2
3236 015436 016703 163522      MOV      SAVR3,R3
3237 015442 016704 163520      MOV      SAVR4,R4
3238 015446 016705 163516      MOV      SAVR5,R5
3239 015452 000002          RTI
3240
3241          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3242
3243 015454 104402          .CONVR: TYPE
3244 015456 017254          MCRLF
3245 015460 010046          .CNVRT: MOV      R0,-(SP)
3246 015462 010146          MOV      R1,-(SP)
3247 015464 010346          MOV      R3,-(SP)
3248 015466 010446          MOV      R4,-(SP)
3249 015470 010546          MOV      R5,-(SP)
3250 015472 017601 000012      MOV      @12(SP),R1
3251 015476 016767 002402 163444 MOV      TEMP,TEMP3
3252 015504 062766 000002 000012 ADD      #2,12(SP)
3253 015512 012167 000154      MOV      (R1)+,WRDCNT
3254 015516 112167 000152      1$:  MOVB   (R1)+,CHRCNT
3255 015522 112167 000147      MOVB   (R1)+,SPACNT
3256 015526 013167 000144      MOV      @ (R1)+,BINWRD
3257 015532 016704 000140      2$:  MOV      BINWRD,R4
3258 015536 116705 000132      MOVB   CHRCNT,R5
3259 015542 012700 020104      MOV      #TEMP,R0
3260 015546 010403      3$:  MOV      R4,R3
3261 015550 042703 177770      BIC      #177770,R3
3262 015554 062703 000060      ADD      #060,R3
3263 015560 110320      MOVB   R3,(R0)+
3264 015562 006204      ASR      R4
3265 015564 042704 100000      BIC      #100000,R4
3266 015570 006204      ASR      R4
3267 015572 006204      ASR      R4
3268 015574 005305      DEC      R5

```

```

;SHIFT FOR NEXT #
;CLUGE TO STOP BIT 15 PROPAGATING.
;DITTO
;DITTO

```

3269	015576	001363			BNE	3\$	
3270	015600	012703	020144		MOV	#MDATA,R3	
3271	015604	114023		4\$:	MOVB	-(R0),(R3)+	
3272	015606	105367	000062		DECB	CHRCNT	
3273	015612	001374			BNE	4\$	
3274	015614	105767	000055		TSTB	SPACNT	
3275	015620	001405			BEQ	6\$	
3276	015622	112723	000040	5\$:	MOVB	#040,(R3)+	
3277	015626	105367	000043		DECB	SPACNT	
3278	015632	001373			BNE	5\$	
3279	015634	105013		6\$:	CLRB	(R3)	
3280	015636	104402			TYPE		
3281	015640	020144			MDATA		
3282	015642	005367	000024		DEC	WRDCNT	
3283	015646	001323			BNE	1\$	
3284	015650	016767	163274	002226	MOV	TEMP3,TEMP	
3285	015656	012605			MOV	(SP)+,R5	
3286	015660	012604			MOV	(SP)+,R4	
3287	015662	012603			MOV	(SP)+,R3	
3288	015664	012601			MOV	(SP)+,R1	
3289	015666	012600			MOV	(SP)+,R0	
3290	015670	000002			RTI		
3291	015672	000000			WRDCNT:	0	
3292	015674	000000			CHRCNT:	0	
3293		015675			SPACNT	CHRCNT+1	
3294	015676	000000			BINWRD:	0	
3295							
3296							
3297							
3298							
3299							
3300							
3301	015700	017605	000000		.SETFLG:MOV	@(SP),R5	
3302	015704	122767	000116	002132	CMPS	#'N',INBUF	;IS IT 'N'?
3303	015712	001002			BNE	1\$	
3304	015714	105015			CLRB	(R5)	;000
3305	015716	000406			BR	2\$	
3306	015720	122767	000131	002116	1\$:	CMPS	#'Y',INBUF ;IS IT 'Y'?
3307	015726	001005			BNE	3\$	
3308	015730	112715	177777		MOVB	#-1,(R5)	;377
3309	015734	062716	000002		2\$:	ADD	#2,(SP)
3310	015740	000002			RTI		
3311	015742	104404			3\$:	INSTR	;RETRY
3312	015744	000755			BR	.SETFLG	
3313							
3314							
3315							
3316							
3317							
3318	015746	011646			.TRPSR:MOV	(SP),-(SP)	;GET PC OF RETURN
3319	015750	162716	000002		SUB	#2,(SP)	;=PC OF TRAP
3320	015754	017616	000000		MOV	@(SP),(SP)	;GET TRP
3321	015760	006316			TRPOK:ASL	(SP)	;MULTIPLY TRAP ARG BY 2
3322	015762	042716	177001		BIC	#177001,(SP)	;CLEAR UNWANTED BITS
3323	015766	062716	001366		ADD	#.TRPTAB,(SP)	;POINTER TO SUBROUTINE ADDRESS
3324	015772	017616	000000		MOV	@(SP),(SP)	;SUBROUTINE ADDRESS

3325	015776	000136			JMP	@(SP)+		;GO TO SUBROUTINE
3326								
3327								
3328								
3329	016000	104413			.HLT:	CKSWR		;CHECK FOR ^G
3330	016002	032777	020000	163070		BIT	#SW13,@SWR	;INHIBIT ERROR TYPE OUT ?
3331	016010	001061				BNE	HALTS	
3332	016012	021667	163116			CMP	(SP),LSTERR	
3333	016016	001404				BEQ	1\$	
3334	016020	011667	163110			MOV	(SP),LSTERR	
3335	016024	105067	163216			CLRB	ERRFLG	
3336	016030	104406			1\$:	SAV05		
3337	016032	011605				MOV	(SP),R5	
3338	016034	162705	000002			SUB	#2,R5	
3339	016040	011504				MOV	(R5),R4	
3340	016042	006304				ASL	R4	
3341	016044	061504				ADD	(R5),R4	
3342	016046	006304				ASL	R4	
3343	016050	042704	177001			BIC	#177001,R4	
3344	016054	062704	020660			ADD	#.ERRTAB,R4	
3345	016060	012467	000040			MOV	(R4)+,ERRMSG	
3346	016064	012467	000046			MOV	(R4)+,DATAHD	
3347	016070	011467	000054			MOV	(R4),DATABP	
3348	016074	105767	163146			TSTB	ERRFLG	
3349	016100	001403				BEQ	TYPMSG	
3350	016102	005767	000042			TST	DATABP	
3351	016106	001014				BNE	TYPDAT	
3352	016110	104410			TYPMSG:	CONVRT		
3353	016112	016242				ERTAB0		
3354	016114	112767	177777	163124		MOVB	#-1,ERRFLG	
3355	016122	104402				TYPE		
3356	016124	000000			ERRMSG:	0		
3357	016126	005767	000004			TST	DATAHD	
3358	016132	001402				BEQ	TYPDAT	
3359	016134	104402				TYPE		
3360	016136	000000			DATAHD:	0		
3361	016140	005767	000004		TYPDAT:	TST	DATABP	
3362	016144	001402				BEQ	RESREG	
3363	016146	104410				CONVRT		
3364	016150	000000			DATABP:	0		
3365	016152	104407			RESREG:	RES05		
3366	016154	005777	162720		HALTS:	TST	@SWR	
3367	016160	100005				BPL	EXITER	
3368	016162	010046				PUSHRO		
3369	016164	016600	000002			MOV	2(SP),R0	
3370	016170	000000				HALT		
3371	016172	012600				POPRO		
3372	016174	104413			EXITER:	CKSWR		;CHECK FOR ^G
3373	016176	005267	162730			INC	ERRCNT	
3374	016202	032777	000400	162670		BIT	#SW08,@SWR	;LOOP ON ERROR ?
3375	016210	001007				BNE	1\$	
3376	016212	032777	002000	162660		BIT	#SW10,@SWR	;ESCAPE TO NEXT ON ERROR ?
3377	016220	001407				BEQ	2\$	
3378	016222	016767	162670	162664		MOV	NEXT,RTRN	;SET UP FOR NEXT TEST
3379	016230	012706	001100		1\$:	MOV	#STACK,SP	;REINITIALIZE SP
3380	016234	000177	162654			JMP	@RTRN	

```

3381 016240 000002      2$: RTI
3382 016242 000001      ERTAB: 1
3383 016244 006        002      .BYTE 6,2
3384 016246 001174      SAVPC
3385                                     ;ENTER HERE ON POWER FAILURE
3386
3387
3388 016250 010046      .PFAIL: MOV R0,-(SP)      ;SAVE R0-R5 ON PROCESSOR STACK
3389 016252 010146      MOV R1,-(SP)
3390 016254 010246      MOV R2,-(SP)
3391 016256 010346      MOV R3,-(SP)
3392 016260 010446      MOV R4,-(SP)
3393 016262 010546      MOV R5,-(SP)
3394 016264 016746 161534 MOV 24,-(SP)
3395 016270 010667 162676 MOV SP,SAVSP      ;SAVE STACK POINTER
3396 016274 012767 016306 161522 MOV #RESTART,24      ;SET UP FOR POWER UP TRAP
3397 016302 000000      HALT      ;HALT ON POWER DOWN NORMAL
3398 016304 000777      1$: BR 1$
3399
3400                                     ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3401
3402 016306 016706 162660 RESTAR: MOV SAVSP,SP      ;RESTORE STACK POINTER
3403 016312 012605      MOV (SP)+,R5      ;RESTORE R0-R5
3404 016314 012604      MOV (SP)+,R4
3405 016316 012603      MOV (SP)+,R3
3406 016320 012602      MOV (SP)+,R2
3407 016322 012601      MOV (SP)+,R1
3408 016324 012600      MOV (SP)+,R0
3409 016326 012767 016250 161470 MOV #.PFAIL,24      ;SET UP FOR POWER FAILURE
3410 016334 012767 000340 161434 MOV #340,PS
3411 016342 012706 001100 MOV #STACK,SP
3412 016346 005067 001532      CLR TEMP
3413 016352 005267 001526      1$: INC TEMP
3414 016356 001375      BNE 1$
3415 016360 104410      CONVRT
3416 016362 016404      PFTAB
3417 016364 104402      TYPE
3418 016366 017257      MPFAIL
3419 016370 005067 162652      CLR ERRFLG
3420 016374 005067 162534      CLR I STERR
3421 016400 000177 162510      JMP @RTRN
3422 016404 000001      PFTAB: 1
3423 016406 006        002      .BYTE 6,2
3424 016410 001114      RTRN
3425
3426
3427                                     ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3428                                     ;OF LOC.176.
3429                                     ;LOCATIONS USED:
3430 016412 000000      RDSW: .WORD 0
3431
3432
3433 016414 005737 000042      .CKSWR: TST @42
3434 016420 001042      BNE OUT
3435 016422 022767 000176 162450 CMP #CJREG,SWR      ;SOFTWARE SWITCH REGISTER PRESENT
3436 016430 001036      BNF OUT      ;NO, GET OUT

```

3437	016432	105777	162446			TSTB	@TKCSR		:YES, WAIT FOR
3438	016436	100033				BPL	OUT		:READY, GET CHARACTER
3439	016440	017767	162442	176352		MOV	@TKDBR, .MSG		:AND STRIP OFF
3440	016446	042767	177600	176344		BIC	#177600, .MSG		:THE GARBAGE
3441	016454	122767	000007	176336		CMPS	#7, .MSG		:IS IT A <^G>
3442	016462	001021				3NE	OUT		
3443	016464	104402	016542			TYPE, SCNTG			
3444	016470	005137	016412			.CNTLU: COM	@WRDSW		
3445	016474	104402	016547			TYPE, SMSWR			
3446	016500	104411	016534			CNVRT, SWREGC			
3447	016504	104403	016557			INSTR, SMNEW			
3448	016510	104405				PARAM			
3449	016512	000000				0			
3450	016514	177777				177777			
3451	016516	000176				SWREG			
3452	016520	000	001			.BYTE	0,1		
3453	016522	104402	017254			TYPE, MCR, F			
3454	016526	005037	016412			OUT: CLR	@WRDSW		
3455	016532	000002				RTI			
3456	016534	000001				SWREGC: 1			
3457	016536	006	002			.BYTE	6,2		
3458	016540	000176				SWREG			
3459	016542	005015	043536	000		SCNTG: .ASCIZ	<15><12>/^G/		
3460	016547	015	051412	051127		SMSWR: .ASCIZ	<15><12>/SWR= /		
3461	016554	020075	000			SMNEW: .ASCIZ	/ NEW= /		
3462	016557	040	047040	053505					
3463	016564	020075	000						
3464	016570	016570				.EVEN			
3465	016570	005015	042012	030525		MTITLE: .ASCIZ	<15><12><12>/DU11 CZDUA-E TAPE A /<15><12>		
3466	016576	020061	055103	052504					
3467	016604	026501	020105	040524					
3468	016612	042520	040440	006440					
3469	016620	000012							
3470	016622	005015	042526	052103		MVFCO: .ASCIZ	<15><12>/VECTOR ADDRESS-/		
3471	016630	051117	040440	042104					
3472	016636	042522	051523	000055					
3473	016644	005015	051461	020124		MREGAD: .ASCIZ	<15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/		
3474	016652	042504	044526	042503					
3475	016660	020072	042522	042503					
3476	016666	053111	051105	041440					
3477	016674	047117	051124	046117					
3478	016702	051040	043505	051511					
3479	016710	042524	020122	042101					
3480	016716	051104	051505	026523					
3481	016724	000							
3482	016725	015	040412	042522		MMULT: .ASCIZ	<15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/		
3483	016732	054440	052517	051040					
3484	016740	047125	044516	043516					
3485	016746	046440	046125	044524					
3486	016754	046120	020105	042504					
3487	016762	044526	042503	020123					
3488	016770	020077	054450	047440					
3489	016776	020122	024516	000055					
3490	017004	005015	040514	052123		MLASTD: .ASCIZ	<15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/		
3491	017012	042040	053105	041511					
3492	017020	035105	042522	042503					

3493	017026	053111	051105	041440	
3494	017034	047117	051124	046117	
3495	017042	051040	043505	051511	
3496	017050	042524	020122	042101	
3497	017056	051104	051505	026523	
3498	017064	000			
3499	017065	075	042504	044526	DEVICE: .ASCIIZ /=DEVICE /
3500	017072	042503	020040	000	
3501	017077	015	044012	053517	MCOW: .ASCIIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
3502	017104	047040	053517	041040	
3503	017112	047522	047127	041440	
3504	017120	053517	020077	027056	
3505	017126	051456	046105	041505	
3506	017134	020124	047523	042515	
3507	017142	044124	047111	020107	
3508	017150	047524	051040	047125	
3509	017156	040040	041501	051124	
3510	017164	043505	000		
3511	017167	015	047412	052125	MRANGE: .ASCIIZ <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
3512	017174	047440	020106	040522	
3513	017202	043516	035105	042522	
3514	017210	054524	042520	046040	
3515	017216	051501	020124	042504	
3516	017224	044526	042503	051040	
3517	017232	041530	051123	040440	
3518	017240	042104	042522	051523	
3519	017246	000055			
3520	017250	020040	000077		MQM: .ASCIIZ / ?/
3521	017254	005015	000		MCRLF: .ASCIIZ <15><12>
3522	017257	040	050040	053517	MPFAIL: .ASCIIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
3523	017264	051105	043040	044501	
3524	017272	052514	042522	020054	
3525	017300	051120	043517	040522	
3526	017306	020115	042522	052123	
3527	017314	051101	020124	052101	
3528	017322	052040	051505	020124	
3529	017330	047111	050040	047522	
3530	017336	051107	051505	000123	
3531	017344	005015	047105	020104	MEPASS: .ASCIIZ <15><12>/END OF PASS TAPE A/
3532	017352	043117	050040	051501	
3533	017360	020123	040524	042520	
3534	017366	040440	000		
3535	017371	015	051012	000	MR: .ASCIIZ <15><12>/R/
3536	017375	015	052012	051505	MTSTPC: .ASCIIZ <15><12>/TEST PC-/
3537	017402	020124	041520	000055	
3538	017410	005015	047514	045503	MLOCK: .ASCIIZ <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
3539	017416	047440	020116	042523	
3540	017424	042514	052103	042105	
3541	017432	052040	051505	037524	
3542	017440	024040	020131	051117	
3543	017446	047040	026451	000	
3544	017453	015	042012	020125	MLEVEL: .ASCIIZ <15><12>/DU PRIORITY LEVEL-/
3545	017460	051120	047511	044522	
3546	017466	054524	046040	053105	
3547	017474	046105	000055		
3548	017500	005015	020043	043117	MSYNC: .ASCIIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/

3549	017506	051440	047131	020103	
3550	017514	044103	051101	020123	
3551	017522	042523	042514	052103	
3552	017530	042105	024040	030440	
3553	017536	047440	020122	024462	
3554	017544	000055			
3555	017546	005015	051511	051440	MWIRE6: .ASCIIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
3556	017554	041505	054040	044515	
3557	017562	020124	052512	050115	
3558	017570	051105	021440	020066	
3559	017576	047111	020077	054450	
3560	017604	047440	020122	024516	
3561	017612	000055			
3562	017614	005015	051511	051440	MWIRE5: .ASCIIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
3563	017622	041505	051040	041505	
3564	017630	045040	046525	042520	
3565	017636	020122	032443	044440	
3566	017644	037516	024040	020131	
3567	017652	051117	047040	026451	
3568	017660	000			
3569	017661	015	044412	020123	MWIRE4: .ASCIIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
3570	017666	050117	020124	046103	
3571	017674	020122	047105	041101	
3572	017702	042514	045040	046525	
3573	017710	042520	020122	032043	
3574	017716	044440	037516	024040	
3575	017724	020131	051117	047040	
3576	017732	026451	000		
3577	017735	015	044412	020123	MEXTJ: .ASCIIZ <15><12>/IS THE TEST CONNECTOR INSTALLED ?(Y OR N)-/
3578	017742	044124	020105	042524	
3579	017750	052123	041440	047117	
3580	017756	042516	052103	051117	
3581	017764	044440	051516	040524	
3582	017772	046114	042105	037440	
3583	020000	054450	047440	020122	
3584	020006	024516	000055		
3585	020012	006412	020040	020040	MSTATUS: .ASCIIZ <12> <15>/ STATUS MAP / <12> <15>
3586	020020	052123	052101	051525	
3587	020026	020040	046440	050101	
3588	020034	020040	020040	005040	
3589	020042	000015			
3590					.EVEN
3591					
3592					;BUFFERS FOR INPUT-OUTPUT
3593					
3594	020044	000040			INBUF: .BLKB 40
3595	020104	000040			TEMP: .BLKB 40
3596	020144	000040			MDATA: .BLKB 40
3597					;*****
3598					;UTILITIES
3599					;*****
3600					
3601					;THIS UTILITY CALCULATES PRIORITY LEVEL
3602	020204	006367	000044		DULEV: ASL DUPRT ;SHIFT LEFT
3603	020210	006367	000040		ASL DUPRT ;
3604	020214	006367	000034		ASL DUPRT ;

```

3605 020220 006367 000030      ASL      DUPRT      ;
3606 020224 006367 000024      ASL      DUPRT      ;
3607 020230 016767 000020 000020  MOV      DUPRT,LESS1 ;MOVE THIS TO LESS1
3608 020236 162767 000001 000012  SUB      #1,LESS1    ;CREATE LESS1
3609 020244 042767 000037 000004  BIC      #37,LESS1   ;CLEAR TNZVC
3610 020252 000207      RTS      PC
3611 020254 000240      DUPRT:  LEVEL5
3612 020256 000200      LESS1:  LEVEL4 ;LEVEL TO ALLOW INTERRUPTS
3613
3614      ;NEW DU ADDRESSES
3615 020260 016767 000126 000422  DUADDR:  MOV      DUBASE,RXCSR ;XXX0
3616 020266 005267 000120      INC      DUBASE
3617 020272 016767 000114 000412  MOV      DUBASE,HRXCSR ;XXX1
3618 020300 005267 000106      INC      DUBASE
3619 020304 016767 000102 000402  MOV      DUBASE,RXDBUF ;XXX2
3620 020312 016767 000074 000400  MOV      DUBASE,PARCSR ;XXX2
3621 020320 005267 000066      INC      DUBASE
3622 020324 016767 000062 000364  MOV      DUBASE,HRXDBUF ;XXX3
3623 020332 016767 000054 000362  MOV      DUBASE,HPARCSR ;XXX3
3624 020340 005267 000046      INC      DUBASE
3625 020344 016767 000042 000352  MOV      DUBASE,TXCSR ;XXX4
3626 020352 005267 000034      INC      DUBASE
3627 020356 016767 000030 000342  MOV      DUBASE,HTXCSR ;XXX5
3628 020364 005267 000022      INC      DUBASE
3629 020370 016767 000016 000332  MOV      DUBASE,TXDBUF ;XXX6
3630 020376 005267 000010      INC      DUBASE
3631 020402 016767 000004 000322  MOV      DUBASE,HTXDBUF ;XXX7
3632 020410 000207      RTS      PC
3633 020412 000000      DUBASE: 0
3634
3635      ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3636      ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3637      ;SHIFTED IN BY THE CONTENTS OF SHIFT
3638 020414 042777 040000 000302  RPOKE:  BIC      #MTDATA,@TXCSR
3639 020422 005067 160520      CLR      TEMP2
3640 020426 006067 160512      ROR      TEMP1 ;FORCE CARRY
3641 020432 006067 160510      ROR      TEMP2 ;PICK UP CARRY IN BIT 15
3642 020436 006267 160504      ASR      TEMP2 ;SHIFT INTO BIT 14
3643 020442 042767 100000 160476  BIC      #BIT15,TEMP2 ;CLR BIT 15
3644 020450 056777 160472 000246  BIS      TEMP2,@TXCSR ;POKE MAINT DATA
3645 020456 042777 020000 000240  BIC      #CLK,@TXCSR ;POKE CLK
3646 020464 052777 020000 000232  BIS      #CLK,@TXCSR ;
3647 020472 005367 160442      DEC      SHIFT
3648 020476 001346      BNE      RPOKE
3649 020500 000207      RTS      PC
3650
3651      ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3652 020502 016767 160436 160436  ODD8:  MOV      TEMP1,TEMP2 ;SAVE TEMP1
3653 020510 005067 160434      CLR      TEMP3
3654 020514 012727 000010      MOV      #8,(PC)+
3655 020520 000000      1$:      0
3656 020522 006067 160420      2$:      ROR      TEMP2
3657 020526 005567 160416      ADC      TEMP3
3658 020532 005367 177762      DEC      1$
3659 020536 001371      BNE      2$
3660 020540 006067 160404      ROR      TEMP3
  
```

```

3661 020544 103404      BCS      3$
3662 020546 052767 000400 160370  BIS      #BIT8,TEMP1      ;SET ODD PARITY
3663 020554 000403      BR      4$
3664 020556 042767 000400 160360 3$:    BIC      #BIT8,TEMP1      ;CLR EVEN PARITY
3665 020564 000207      ;TEMP1 NOW HAS ODD PARITY CHARACTER
3666 020564 000207      4$:    RTS      PC
3667
3668 ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3669 020566 016767 160352 160352  EVEN8:  MOV     TEMP1,TEMP2      ;SAVE TEMP1
3670 020574 005067 160350      CLR     TEMP3
3671 020600 012727 000010      MOV     #8.,(PC)+
3672 020604 000000      1$:    0
3673 020606 006067 160334      2$:    ROR     TEMP2
3674 020612 005567 160332      ADC     TEMP3
3675 020616 005367 177762      DEC     1$
3676 020622 001371      BNE     2$
3677 020624 006067 160320      ROR     TEMP3
3678 020630 103004      BCC     3$
3679 020632 052767 000400 160304  BIS      #BIT8,TEMP1      ;SET EVEN PARITY
3680 020640 000403      BR      4$
3681 020642 042767 000400 160274 3$:    BIC      #BIT8,TEMP1      ;CLR ODD PARITY
3682 020650 000207      ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3683 020652 062716 000002      4$:    RTS      PC
3684 020652 062716 000002  TRPREG: ADD     #2,(SP) ;ALLOW IT TO 'CRUNCH' INTO HLT BACK
3685 020656 000002      ;IN MAIN PART OF THE PROGRAM
3686 020656 000002      RTI
3687 ;ERROR HLT TABLE
3688 020660 020744      .ERRTAB: EMO      ;HLT 0 BIT ERROR (GENERAL)
3689 020662 000000      0
3690 020664 000000      0
3691 020666 020760      EM1      ;HLT 1 REGISTER ERROR
3692 020670 021131      DH1
3693 020672 021152      DT1
3694 020674 021022      EM2      ;HLT 2 RECEIVER ERROR
3695 020676 021131      DH1
3696 020700 021152      DT1
3697 020702 021064      EM3      ;HLT 3 TRANSMITTER ERROR
3698 020704 021131      DH1
3699 020706 021152      DT1
3700 ;DEFAULT DU ADDRESSES
3701 020710 160040      RXCSR: 160040
3702 020712 160041      HRXCSR: 160041
3703 020714 160042      RXDBUF: 160042
3704 020716 160043      HPXDBUF: 160043
3705 020720 160042      PARCSR: 160042
3706 020722 160043      HPARCSR: 160043
3707 020724 160044      TXCSR: 160044
3708 020726 160045      HTXCSR: 160045
3709 020730 160046      TXDBUF: 160046
3710 020732 160047      HTXDBUF: 160047
3711 ;DEFAULT DU VECTORS
3712 020734 000770      DURIV: 770      ;REC INTR VECTOR
3713 020736 000772      DURIS: 772      ;REC INTR STATUS
3714 020740 000774      DUTIV: 774      ;XMIT INTR VECTOR
3715 020742 000776      DUTIS: 776      ;XMIT INTR STATUS
3716 ;ERROR MESSAGES

```

3717	020744	036440	042440	051122	EM0:	.ASCIZ / ERROR PC/
3718	020752	051117	050040	000103		
3719	020760	036440	051040	043505	EM1:	.ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER ,/
3720	020766	051511	042524	020122		
3721	020774	051105	047522	020122		
3722	021002	041520	005015	051001		
3723	021010	043505	051511	042524		
3724	021016	020122	000040			
3725	021022	036440	051040	041505	EM2:	.ASCIZ / - RECEIVER ERROR PC/<15><12><1>/REGISTER /
3726	021030	044505	042526	020122		
3727	021036	051105	047522	020122		
3728	021044	041520	005015	051001		
3729	021052	043505	051511	042524		
3730	021060	020122	000040			
3731	021064	036440	052040	040522	EM3:	.ASCIZ / TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3732	021072	051516	044515	052124		
3733	021100	051105	042440	051122		
3734	021106	051117	050040	006503		
3735	021114	000412	042522	044507		
3736	021122	052123	051105	020040		
3737	021130	000				
3738						
3739	021131	105	050130	041505	DH1:	:DATA HEADERS FOR ERROR MESSAGES .ASCIZ /EXPECTED ACTUAL,
3740	021136	042524	020104	040440		
3741	021144	052103	040525	000114		
3742					.EVEN	
3743						
3744	021152	000003			D11:	:DATA TABLES FOR ERROR MESSAGES
3745	021154	006	004			.BYTE 6,4
3746	021156	001164				SAVR3 :REGISTER
3747	021160	006	004			.BYTE 6,4
3748	021162	001156				SAVR0 :EXPECTED DATA
3749	021164	006	002			.BYTE 6,2
3750	021166	001160				SAVR1 :ACTUAL DATA
3751		000001			.END	

[illegible]

CSHRX	001342	951#	1073	1239*															
CSRHTX	001356	957#	1079	1245*															
CSRP	001350	954#	1076	1242*															
CSRRX	001340	950#	1072	1238*															
CSRTX	001354	956#	1078	1244*															
CTS	= 020000	770#																	
D	= *****	675																	
DATA	016150	3347*	3350	3361	3364#														
DATA	016136	3346*	3357	3360#															
DEVADR	015356	3165*	3203	3213#															
DEVICE	017065	2999	3499#																
DH1	021131	3692	3695	3698	3739#														
DISPRE	000174	835#	1032																
DLIGHT	= 177570	725#	1023																
DMULT	001304	932#	1055	1222*															
DNA	= 100000	804#	1996																
DNAINT	= 000040	811#	1668	1669	1673	1674	1679	1691											
DSC	= 100000	768#	1973																
DSINTE	= 000040	778#	1530	1531	1535	1536	1541	1543											
DSR	= 001000	774#																	
DSWR	= 177570	724#	1022																
DTR	= 000002	782#	1408	1409	1413	1414	1419	1425	1430	2143	2215	2305	2354						
DT1	021152	3693	3696	3699	3744#														
DUADDR	020260	1106	3026	3615#															
DUBASE	020412	1071*	1102	1105	1237	3025*	3615	3616*	3617	3618*	3619	3620	3621*	3622					
		3623	3624*	3625	3626*	3627	3628*	3629	3630*	3631	3633#								
DULEV	020204	1172	3602#																
DUPRT	020254	1070*	1169	1231	3602*	3603*	3604*	3605*	3606*	3607	3611#								
DURIS	020736	1066*	1233	3029*	3713#														
DURIV	020734	1065*	1113	1116	1117	1232	3027*	3034	3712#										
DUTIS	020742	1068*	1235	3033*	3715#														
DUTIV	020740	1067*	1234	3031*	3714#														
EIGHT	= 006000	799#	2386	2416	2698														
EMO	020744	3688	3717#																
EM1	020760	3691	3719#																
EM2	021022	3694	3725#																
EM3	021064	3697	3731#																
ERRCNT	001132	861#	1013*	3373*															
ERRFLG	001246	920#	1012*	3043*	3089*	3335*	3348	3354*	3419*										
ERRMSG	016124	3345*	3356#																
ERTABO	016242	3353	3382#																
EVEN8	020566	3669#																	
EVEPAR	= 001400	802#																	
EVPAR	= 000400	791#																	
FXITER	016174	3367	3372#																
FILBUF	003204	1268#	1271																
FIVE	= 000000	796#	2455	2774	2832	2890	2948												
FLAG	001222	905#	1087*	1258	1262*														
FRMERR	= 020000	787#	2053																
HALTS	016154	3331	3366#																
HDXEN	= 000010	813#	1622	1623	1627	1628	1633	1635											
HEREU	003132	1217	1258#																
HILIM	015354	3164*	3194	3212#															
HOLD	001136	866#	1891	1901	1924	1934	2148	2170	2196	2220	2236	2252	2277	2293					
		2310	2327	2343	2359														
HOLDO	001224	906#	1263*	1279															

[illegible]

MQM	017250	3149	3520#											
MR	017371	1317	3535#											
MRANGE	017167	1155	3511#											
MREGAD	016644	1098	3473#											
MRESET=	000400	808#	1420	1453	1486	1519	1542	1565	1588	1611	1634	1657	1680	1703
		1728	1753	1776	1799	1833	1834	1838	1839	1849	1864	1899	1932	1948
		1960	1972	1984	1995	2007	2020	2032	2044	2056	2068	2096	2110	2127
		2378	2380	2408	2410	2447	2449	2528	2530	2609	2611	2690	2692	2766
		2768	2824	2826	2882	2884	2940	2942						
MSTATU	020012	1261	3585#											
MSYNC	017500	1179	3548#											
MTDATA=	040000	805#	1787	1788	1792	1793	1798	1800	1885	1890	1898	1918	1923	1931
		2383	2413	2452	2533	2614	2695	2771	2829	2887	2945	3638		
MTITLE	016570	1020	3465#											
MTSTPC	017375	1308	3536#											
MULTD	001202	887#	1055*	1121	1122	1222	1286	3000						
MVECTO	016622	1109	3470#											
MWIRE4	017661	1200	3569#											
MWIRE5	017614	1196	3562#											
MWIRE6	017546	1192	3555#											
NEXT	001116	855#	1323*	338*	1353*	1368*	1383*	1398*	1407*	1440*	1473*	1506*	1529*	1552*
		1575*	1598*	1621*	1644*	1667*	1690*	1715*	1740*	1763*	1786*	1809*	1832*	1848*
		1884*	1913*	1947*	1959*	1971*	1983*	1994*	2006*	2019*	2031*	2043*	2055*	2066*
		2094*	2109*	2125*	2377*	2407*	2446*	2527*	2608*	2689*	2765*	2823*	2881*	2939*
		3072	3092	3378										
NOPAR =	000000	800#	2386	2416	2455	2536	2617	2698	2774	2832	2890	2948		
NOSYNC	001300	928#	1051	1218*										
ODDPAR=	001000	801#												
ODD8	020502	3652#												
ONCE	001522	1019	1022#											
OPTCLR	001201	886#	1054*	1202	1221	1423	1456	1489	1816	1850	2071			
OUT	016526	3434	3436	3438	3442	3454#								
OUTCRY	014440	2998	3037#											
OUTMUL	002476	1127	1152	1163#										
OUT1	011004	2135	2269	2321	2370#									
OUT2	010530	2188	2272#											
OUT3	010660	2190	2322#											
OVRRUN=	040000	786#	2045											
PARAM -	104405	980#	1099	1110	1131	1156	1166	1309	3448					
PARAM1	015206	3169#	3190											
PARCSR	020720	1076*	1242	1356*	2379*	2								

RDSW	016412	3150	3187	3430#	3444*	3454*								
REACT=	004000	772#	1961	2387	2392	2397	2417	2422	2427	2477	2488	2493	2558	2569
		2574	2639	2650	2655	2720	2731	2736						
REGACT	001320	942#	1061	1229*										
REPLAY	014344	3017	3023	3025#										
RESEC	001302	930#	1053	1220*										
RESREG	016152	3362	3365#											
RESTAR	016306	3396	3402#											
RESTR	014510	3018	3047	3053#										
RES05 =	104407	984#	3365											
RING =	040000	769#												
RINTEN=	000100	777#	1553	1554	1558	1559	1564	1566						
RISDU	001334	948#	1066	1233*										
RIVDU	001326	945#	1065	1232*										
ROTADD	001220	900#	1062*	1126*	1138*	1140	1142*	1147	1150*	1230	3012*	3015	3019*	
RPOKE	020414	2467	2483	2548	2564	2629	2645	2710	2726	2786	2800	2804	2844	2858
		2862	2902	2916	2920	2960	2974	2978	3638#	3648				
RTRN	001114	854#	1016*	1312	1316*	1318	3055*	3072*	3073	3092*	3093	3378*	3380	3421
		3424												
RTS =	000004	781#	1441	1442	1446	1447	1452	1458	1463	2165	2231	2288	2338	
RUNA =	000000	1	52	69	147	663								
RUNB =	***** U	20	57	69	150	668								
RUNC =	***** U	20	57	69	150	668								
RUND =	***** U	20	57	69	150	668								
RUNE =	***** U	20	57	72	150	668								
RUNF =	***** U	20	57	72	150	668								
RUNIT	014250	3003	3009#	3016										
RXCSR	020710	1072*	1238	1326*	1408*	1409	1413*	1414	1419*	1425	1430	1441*	1442	1446*
		1447	1452*	1458	1463	1474*	1475	1479*	1480	1485*	1491	1496	1507*	1508
		1512*	1513	1518*	1520	1530*	1531	1535*	1536	1541*	1543	1553*	1554	1558*
		1559	1564*	1566	1576*	1577	1581*	1582	1587*	1589	1810*	1814	1852*	1853
		1855	1859*	1860	1949	1961	1973	2067*	2069	2070	2126*	2137	2138	2143*
		2154	2159	2165*	2176	2181	2191*	2202	2208	2215*	2226	2231*	2242	2247*
		2258	2263	2272*	2283	2288*	2299	2305*	2316	2322*	2333	2338*	2349	2354*
		2365	2387	2391*	2392	2396*	2397	2417	2421*	2422	2426*	2427	2457*	2473
		2477	2484	2488	2492*	2493	2497	2507	2538*	2554	2558	2565	2569	2573*
		2574	2578	2588	2619*	2635	2639	2646	2650	2654*	2655	2659	2669	2700*
		2716	2720	2727	2731	2735*	2736	2740	2750	2775*	2787	2833*	2845	2891*
		2903	2949*	2961	3039	3615*	3701#							
RXDBUF	020714	1074*	1240	1341*	2008	2010	2021	2033	2045	2057	2111	2112	2456	2502
		2537	2583	2618	2664	2699	2745	2782	2791	2807	2840	2849	2865	2898
		2907	2923	2956	2965	2981	3619*	3703#						
		776#	1949											
RXDONE=	000200	785#	2057											
RXERR =	100000	881#	3219*	3384										
SAVPC	001174	874#	3228*	3233	3748									
SAVR0	001156	875#	3227*	3234	3750									
SAVR1	001160	876#	3226*	3235										
SAVR2	001162	877#	3225*	3236	3746									
SAVR3	001164	878#	3224*	3237										
SAVR4	001166	879#	3223*	3238										
SAVR5	001170	880#	3395*	3402										
SAVSP	001172	982#	3336											
SAV05 =	104406	970#	1334	1349	1364	1379	1394	1402	1435	1468	1501	1524	1547	1570
SCOPE =	104400	1593	1616	1639	1662	1685	1708	1733	1758	1781	1804	1827	1844	1878
		1907	1940	1954	1966	1978	1989	2001	2014	2026	2038	2050	2062	2088

SCOP1 = 104401	2103	2117	2370	2401	2431	2512	2593	2674	2755	2813	2871	2929	2987
SEND = 000020	972#												
SEREC 001200	812#	1645	1646	1650	1651	1656	1658						
SETFG 003144	885#	1053*	1198	1220	2189								
SETFLG= 104412	1259	1261#											
SEVEN = 004000	990#	1120	1193	1197	1201	1205	1296						
SEXMIT 001177	798#	2617											
SHIFT 001140	884#	1052*	1194	1219	2187								
	867#	2465*	2481*	2546*	2562*	2627*	2643*	2708*	2724*	2784*	2798*	2802*	2842*
	2856*	2860*	2900*	2914*	2918*	2958*	2972*	2976*	3647*				
SIX = 002000	797#	2536											
SPACNT= 015675	3255*	3274	3277*	3293#									
SRD = 002000	773#												
STACK = 001100	727#	1007	1283	3379	3411								
STATUS 001300	928#	1267	1270										
STD = 000010	780#	1474	1475	1479	1480	1485	1491	1496	2191	2247	2272	2322	
STFLG 001245	919#	1010*											
STPSYN= 000400	775#	1576	1577	1581	1582	1587	1589						
SV05 015370	3223#												
SWR 001100	845#	1022*	1027	1031*	1037	1046	1084	1216	1292	1305	3077	3084	3101
	3330	3366	3374	3376	3435								
SWREG 000176	836#	1031	1037	3435	3451	3458							
SWREGC 016534	3446	3456#											
SW00 = 000001	707#	1084											
SW01 = 000002	706#	1305											
SW02 = 000004	705#												
SW03 = 000010	704#												
SW04 = 000020	703#												
SW05 = 000040	702#												
SW06 = 000100	701#												
SW07 = 000200	700#	1046	1216										
SW08 = 000400	699#	3374											
SW09 = 001000	698#	3101											
SW10 = 002000	697#	3376											
SW11 = 004000	696#	3084											
SW12 = 010000	695#												
SW13 = 020000	694#	3330											
SW14 = 040000	693#	3077											
SW15 = 100000	692#												
SYNCD 001176	883#	1051*	1182*	1186*	1218	2471	2552	2633	2714				
SYNEXT= 020000	794#	2379	2386										
SYNINT- 030000	793#	2448	2455	2529	2536	2610	2617	2691	2698				
SYNSCH- 000020	779#	1507	1508	1512	1513	1518	1520	2391	2396	2421	2426	2457	2492
	2538	2573	2619	2654	2700	2735	2775	2833	2891	2949			
SYSTST= 014000	819#												
TABLE 001232	909#	1273*	1274*	1275									
TEMP 020104	3251	3259	3284*	3412*	3413*	3595#							
TEMP1 001144	869#	2466*	2482*	2547*	2563*	2628*	2644*	2709*	2725*	2785*	2799*	2803*	2843*
	2857*	2861*	2901*	2915*	2919*	2959*	2973*	2977*	3640*	3652	3662*	3664*	3669
	3679*	3681*											
TEMP2 001146	870#	3639*	3641*	3642*	3643*	3644	3652*	3656*	3669*	3673*			
TEMP3 001150	871#	3251*	3284	3653*	3657*	3660*	3670*	3674*	3677*				
TEMP4 001152	872#												
TEMP5 001154	873#												
THRU 003262	1260	1283#											
TISDU 001332	947#	1068	1235*										

[illegible]

TST49	013234	2765	2822#											
TST5	003776	1368	1382#											
TST50	013476	2823	2880#											
TST51	013740	2881	2938#	2990										
TST52 =	***** U	2939												
TST6	004064	1383	1397#											
TST7	004114	1398	1406#											
TST8	004244	1407	1439#											
TST9	004374	1440	1472#											
TTST	014614	1289*	1290*	1300*	1301*	1303*	1304*	3078#						
TXCSR	020724	1078*	1244	1371*	1420*	1453*	1486*	1519*	1542*	1565*	1588*	1599*	1600	1604*
		1605	1610*	1611*	1612	1622*	1623	1627*	1628	1633*	1634*	1635	1645*	1646
		1650*	1651	1656*	1657*	1658	1668*	1669	1673*	1674	1679*	1680*	1681	1691*
		1692	1696*	1697	1702*	1703*	1704	1716*	1717	1721*	1722	1727*	1728*	1729
		1741*	1742	1746*	1747	1752*	1753*	1754	1764*	1765	1769*	1770	1775*	1776*
		1777	1787*	1788	1792*	1793	1798*	1799*	1800	1811*	1815	1833*	1834	1838*
		1839	1849*	1864*	1866	1872*	1873	1885*	1887	1890*	1894	1898*	1899*	1900*
		1904	1918*	1920	1923*	1927	1931*	1932*	1933*	1937	1948*	1960*	1972*	1984*
		1985	1995*	1996	2007*	2020*	2032*	2044*	2056*	2068*	2095*	2096*	2097	2098
		2110*	2127*	2378*	2380*	2383*	2408*	2410*	2413*	2447*	2449*	2452*	2459*	2460*
		2462*	2463*	2528*	2530*	2533*	2540*	2541*	2543*	2544*	2609*	2611*	2614*	2621*
		2622*	2624*	2625*	2690*	2692*	2695*	2702*	2703*	2705*	2706*	2766*	2768*	2771*
		2777*	2778*	2780*	2781*	2824*	2826*	2829*	2835*	2836*	2838*	2839*	2882*	2884*
		2887*	28,3*	2894*	2896*	2897*	2940*	2942*	2945*	2951*	2952*	2954*	2955*	3625*
		3638*	3644*	3645*	3646*	3707#								
TXDBUF	020730	1080*	1246	1386*	1399	3629*	3709#							
TXDONE=	000200	809#	1985											
TXINTE=	000100	810#	1691	1692	1696	1697	1702	1704						
TYPDAT	016140	3351	3358	3361#										
TYPE =	104402	974#	1020	1261	1276	1317	2996	2999	3004	3126	3136	3148	3152	3243
		3280	3355	3359	3417	3443	3445	3453						
TYPMSG	016110	3349	3352#											
UP	003224	1273#	1278											
USER =	000000	816#												
VOID =	000001	783#												
WRDCNT	015672	3253*	3282*	3291#										
ZERO	014256	3010#												
\$CNTG	016542	3443	3459#											
\$E =	000065	676#	1323	1324#	1338	1339#	1353	1354#	1368	1369				

\$SYMB0	676#	689													
\$SYNCR	676#	2457	2538	2619	2700	2775	2833	2891	2949						
\$TRAPS	676#	962													
\$TRPAR	676#														
\$TRPDE	676#	970	972	974	976	978	980	982	984	986	988	990	992	994	
\$TRPSR	676#	3313													
\$TSTNO	676#	1322	1337	1352	1367	1382	1397	1406	1439	1472	1505	1528	1551	1574	1597
	1620	1643	1666	1689	1714	1739	1762	1785	1808	1831	1847	1883	1912	1946	1958
	1970	1982	1993	2005	2018	2030	2042	2054	2065	2093	2108	2124	2376	2406	2445
	2526	2607	2688	2764	2822	2880	2938								
\$TYPE	676#	3105													
\$UNIBU	676#	1320	1335	1350	1365	1380									
\$VARIA	676#	840													
\$WORDF	676#														
\$WORDO	676#	2756	2814	2872	2930										
\$WORDP	676#														

. ABS. 021170 000

ERRORS DETECTED: 0

CZDUA.E.BIN,CZDUA.E.SEQ/CRF/SOL/NL:TOC CZDU11.MLO/EQ:RUNA,EZDU22.PAR,CZDU11.KET,CZDUA.E.P11
 RUN-TIME: 9 13 1 SECONDS
 RUN-TIME RATIO: 100/25 3.9
 CORE USED: 19K (37 PAGES)