

DU 11

OFF-LINE TRANSMITTER TESTS

CZDUDDO

AH-8689D-MC

COPYRIGHT 73-78

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-8688D-MC

PRODUCT NAME: CZDUDDO DU11 OFFLINE TRANSMITTER 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 TRANSMITTER TESTS VERIFY THAT THE TRANSMITTER SECTION PROVIDES THE CORRECT ERROR FLAGS, AND THAT IT TRANSMITS CHARACTERS THRU THE BIT WINDOW AT THE CORRECT NUMBER OF BITS PER CHARACTER.

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
STARTING ADDRESS

4.2

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 CZDUD-D TAPE D' (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' TIMEOUT - 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 YYY.YYY 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 TRANSMITTER SECTION TESTING
OF THE DEVICE
SEE LISTING FOR DETAILS

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

11. LISTINGS

CZDUD-D MACY11 30A(1052) 21-NOV-78 15:46 PAGE 15
CZDU11.PAR 21-NOV-78 15:17

B 2

SEQ 0014

676
677 000000' 000000G

D


```
678      .ENABLE ABS
679
680      ;DU11 CZDUD-D TAPE D
681      ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
682
683      ;STARTING PROCEDURE
684      ;LOAD PROGRAM
685      ;PRESS START
686      ;PROGRAM WILL TYPE 'DU11 CZDUD-D TAPE D '
687      ;PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
688      ;AT THE END OF A PASS, PROGRAM WILL TYPE 'END OF PASS TAPE D'
689      ;AND THEN RESUME TESTING
690
691
692      ;SWITCH REGISTER OPTIONS
693
694      100000      SW15=100000      ;=1,HALT ON ERROR
695      040000      SW14=40000      ;=1,LOOP ON CURRENT TEST
696      020000      SW13=20000      ;=1,INHIBIT ERROR TYPEOUT
697      010000      SW12=10000
698      004000      SW11=4000      ;=1,INHIBIT ITERATIONS
699      002000      SW10=2000      ;=1,ESCAPE TO NEXT TEST ON ERROR
700      001000      SW09=1000      ;=1,LOOP WITH CURRENT DATA
701      000400      SW08=400      ;=1,LOOP ON ERROR
702      000200      SW07=200      ;+ =1, USE STATUS MAP
703      000100      SW06=100
704      000040      SW05=40
705      000020      SW04=20
706      000010      SW03=10
707      000004      SW02=4
708      000002      SW01=2
709      000001      SW00=1
710
711      ;LOCK ON TEST SELECT
712      ;RESTART PROGRAM AT SELECTED TEST
713      ;RESELECT VECTOR AND CONTROL REGISTER
714      ;ADDRESS AFTER PROGRAM RESTART
```

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
766
767

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


```

768      ;REGISTER DEFINITIONS
769      ;RXCSR BIT DEFINITIONS
770      100000 DSC=BIT15      ;DATA SET CHANGE
771      040000 RING=BIT14     ;RING
772      020000 CTS=BIT13     ;CLR TO SEND
773      010000 CARDET=BIT12  ;CARRIER DETECT
774      004000 REACT=BIT11   ;REC ACTIVE
775      002000 SRD=BIT10     ;SEC REC DATA
776      001000 DSR=BIT9      ;DATA SET RDY
777      000400 STPSYN=BIT8   ;STRIP SYNC
778      000200 RXDONE=BIT7   ;REC DONE
779      000100 RINTEN=BIT6   ;REC INTR ENABLE
780      000040 DSINTE=BIT5   ;DSC INTR ENABLE
781      000020 SYN SCH=BIT4  ;SYNC SEARCH
782      000010 STD=BIT3      ;SEC XMIT DATA
783      000004 RTS=BIT2      ;REQ TO SEND
784      000002 DTR=BIT1      ;DATA TERM RDY
785      000001 VOID=BIT0
786      ;RXDBUF BIT DEFINITIONS
787      100000 RXERR=BIT15    ;REC ERROR
788      040000 OVRUN=BIT14   ;OVERRUN
789      020000 FRMERR=BIT13  ;FRAME ERROR
790      010000 PARER=BIT12   ;PARITY ERROR
791      ;PARCSR BIT DEFINITIONS
792      001000 PAREN=BIT9     ;PARITY ENABLE
793      000400 EVPAR=BIT8    ;EVEN PARITY SENSE
794      ;PARCSR WRD DEFINITIONS
795      030000 SYNINT=30000   ;SYNC EXTERNAL MODE
796      020000 SYNEXT=20000  ;SYNC INTERNAL MODE
797      000000 ISYMOD=0      ;ISOC MODE
798      000000 FIVE=0        ;WORD LENGTH 5 BITS
799      002000 SIX=2000      ;WORD LENGTH 6 BITS
800      004000 SEVEN=4000   ;WORD LENGTH 7 BITS
801      006000 EIGHT=6000   ;WORD LENGTH 8 BITS
802      000000 NOPAR=0      ;NO PARITY
803      001000 ODDPAR=1000   ;ODD PARITY
804      001400 EVEPAR=1400   ;EVEN PARITY
805      ;TXCSR BIT DEFINITIONS
806      100000 DNA=BIT15     ;DATA NOT AVAILABLE
807      040000 MTDATA=BIT14  ;MAINT DATA
808      020000 CLK=BIT13     ;CLK
809      002000 BITW=BIT10    ;BIT WINDOW
810      000400 MRESET=BIT8   ;MASTER RESET
811      000200 TXDONE=BIT7   ;XMIT DONE
812      000100 TXINTE=BIT6   ;XMIT INTR ENABLE
813      000040 DNAINTE=BIT5  ;DNA INTR ENAB
814      000020 SEND=BIT4     ;SEND
815      000010 HDXEN=BIT3    ;HDX/FDX
816      000001 BREAK=BIT0    ;BREAK
817      ;TXCSR WRD DEFINITIONS
818      000000 USER=0        ;USER MODE
819      004000 MINT=4000     ;MAINT INT MODE
820      010000 MEXT=10000   ;MAINT EXT MODE
821      014000 SYSTST=14000 ;SYSTEM TEST MODE
822      ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
  
```

..=24

```

.PFAIL
340
.HLT
340
.TRPSRV
340

```

```
;POWER FAIL HANDLER
;SERVICE AT LEVEL 7
;ERROR HANDLER
;SERVICE AT LEVEL 7
;GENERAL HANDLER DISPATCH SERVICE
;SERVICE AT LEVEL 7
```

;SOFTWARE SWITCH REGISTER

..=174

```
DISPREG: .WORD 0
SWREG:   .WORD 0
JMP      .START
```

```

;SOFTWARE DISPLAY REG.
;SOFTWARE SWITCH REGISTER
;GO TO START OF PROGRAM

```

 $\therefore = 1100$

:INDIRECT POINTERS

SWR: 177570
LIGHTS: 177570
TKCSR: 177560
TKDBR: 177562
TPCSR: 177564
TPDBR: 177566

```
;SWITCH REGISTER POINTER
;DISPLAY REGISTER POINTER
;TELETYPE KEYBOARD CONTROL REGISTER
;TELETYPE KEYBOARD DATA BUFFER
;TELEPRINTER CONTROL REGISTER
;TELEPRINTER DATA BUFFER
```

:PROGRAM CONTROL PARAMETERS

```
RTRN: 0
NEXT: 0
LOCK: 0
ICOUNT: 0
LPCNT: 0
TSTNO: 0
PASCNT: 0
ERRCNT: 0
LSTERR: 0
```

```
;SCOPE ADDRESS FOR LOOP ON TEST
;ADDRESS OF NEXT TEST TO BE EXECUTED
;ADDRESS FOR LOCK ON CURRENT DATA
;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
;NUMBER OF ITERATIONS COMPLETED
;NUMBER OF TEST IN PROGRESS
;NUMBER OF PASSES COMPLETED
;TOTAL NUMBER OF ERRORS
;PC OF LAST ERROR CALL
```

```

PROGRAM VARIABLES

```

```
HOLD:      20
SHIFT:     0
COUNT:    0
TEMP1:     0
TEMP2:     0
TEMP3:     0
TEMP4:     0
TEMP5:     0
SAVR0:     0
SAVR1:     0
SAVR2:     0
```

```
;TEMPORARY STORAGE=DELAY TIME FOR CABLES
;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
;TEMPORARY STORAGE
;TEMPORARY STORAGE
;TEMPORARY STORAGE
;TEMPORARY STORAGE
;TEMPORARY STORAGE
;R0 STORAGE
;R1 STORAGE
;R2 STORAGE
```


879 001164 000000
880 001166 000000
881 001170 000000
882 001172 000000
883 001174 000000

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

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

```

884                                     ;PROGRAM CONVERSATIONAL PARAMETERS
885 001176 377 SYNCNO: .BYTE 377 ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
886 001177 377 SEXMIT: .BYTE 377 ;SEC XMIT JUMPER 'IN'
887 001200 377 SEREC: .BYTE 377 ;SEC REC JUMPER 'IN'
888 001201 377 OPTCLR: .BYTE 377 ;OPTIONAL JUMPER CLR 'IN'
889 001202 000 MULTD: .BYTE 0 ;NO MULTIPLE DEVICE FLAG
890 001203 377 JMRBY: .BYTE 377 ;EXTERNAL MODEM BYPASS JUMPER 'IN'
891 .EVEN
892
893                                     ;PROGRAM MULTIPLE DEVICE PARAMETERS
894 001204 000000 BASEADD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
895 001206 000000 KEEPADD: 0 ;SAVED 1ST DEVICE ADDR
896 001210 000000 LASTADD: 0 ;LAST DEVICE RXCSR ADDR
897 001212 000000 BASEIV: 0 ;PROG CONTROLLED IV
898 001214 000000 KEEPIV: 0 ;SAVED INTR VECTOR
899 001216 000000 ACTREG: 0 ;ACTIVE REGISTER ...MODIFY THIS
900 ;LOCATION TO DISQUALIFY OR QUALIFY
901 ;DEVICES (1= RUN,,0= DON'T RUN)
902 001220 000000 ROTADD: 0 ;ROTATING POINTER FOR ACTREG..POINTS
903 ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE
904 ;*****
905
906 ; THESE ARE STORAGE FOR THE STATUS MAP PRINT OUT
907 001222 000000 FLAG:0 ; FLAGS FOR STATUS MAP PRINT OUT (SSP)
908 001224 000000 HOLD0: 0 ; HOLDS R0 IN STATUS MAP PRINT
909 001226 000000 HOLD1:0 ; R1 ETC.
910 001230 000000 COUNT1:0 ; FOR COUNTING 3 WORDS
911 001232 000002 TABLE : 2 ; FOR CONVRT ROUTINE
912 001234 003006 3006
913 001236 000000 0
914 001240 003006 3006
915 001242 000000 0
916 ;*****
917 ;PROGRAM CONTROL FLAGS
918
919
920 001244 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
921 001245 000 STFLG: .BYTE 0 ;TEST START FLAG
922 001246 000 ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
923 001247 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
924
925 .EVEN
926
927 ;***** STATUS MAP *****
928
929 001300 001300 .=1300
930 001300 000001 STATUS: NOSYNC: .BLKB 1 ;SYNC CHARS
931 001301 000001 MITSEX: .BLKB 1 ;XMIT JUMPER
932 001302 000001 RESEC: .BLKB 1 ;REC SEC JUMPER
933 001303 000001 CLROPT: .BLKB 1 ;OPTIONAL JUMPER
934 001304 000001 DMULT: .BLKB 1 ;MULTIPLE DEVICE FLAG
935 001305 000001 BYJMR: .BLKB 1 ;EXTERNAL MODEM
936
937 ; MULTIPLE DEVICE PARAMETERS
938
939 001306 000001 ADDBASE: .BLKW 1 ;PROG CONTROLLED 1ST DEVICE ADDR

```


940 001310 000001
941 001312 000001
942 001314 000001
943 001316 000001
944 001320 000001
945 001322 000001
946 001324 000001
947 001326 000001
948 001330 000001
949 001332 000001
950 001334 000001
951 001336 000001
952 001340 000001
953 001342 000001
954 001344 000001
955 001346 000001
956 001350 000001
957 001352 000001
958 001354 000001
959 001356 000001
960 001360 000001
961 001362 000001
962 001364 000001

ADDKEEP: .BLKW 1
ADDLAST: .BLKW 1
IVBASE: .BLKW 1
IVKEEP: .BLKW 1
REGACT: .BLKW 1
ADDROT: .BLKW 1
PRTDU: .BLKW 1
RIVDU: .BLKW 1
TIVDU: .BLKW 1
TISDU: .BLKW 1
RISDU: .BLKW 1
L1ESS: .BLKW 1
CSRXX: .BLKW 1
CSRHRX: .BLKW 1
BUFRXD: .BLKW 1
BUFHRXD: .BLKW 1
CSRPAR: .BLKW 1
CSRHPAR: .BLKW 1
CSRTX: .BLKW 1
CSRHTX: .BLKW 1
BUFTXD: .BLKW 1
BUFHTXD: .BLKW 1
BASEDU: .BLKW 1
;SAVED 1ST DEVICE ADDR
;LAST DEVICE RXCSR ADDR
;PROG CONTROLLED IV
;SAVED INTR VECTOR
;ACTIVE REGISTER
;ROTATING POINTER
;DU11 PRIORITY
;DU11 REC INTR VECTOR
;DU11 XMIT INTR VECTOR
;DU11 XMIT INTR STATUS
;DU11 REC INTR STATUS
;PRIORITY TO ALLOW INTR
; DEFAULT OR ALTERED PARAMETERS
;DU11 RXCSR BASE ADDR
;EVEN

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

963
964
965
966
967
968
969 001366
970
971
972 104400
973 001366 014532
974 104401
975 001370 014716
976 104402
977 001372 014736
978 104403
979 001374 014776
980 104404
981 001376 015114
982 104405
983 001400 015146
984 104406
985 001402 015362
986 104407
987 001404 015422
988 104410
989 001406 015454
990 104411
991 001410 015460
992 104412
993 001412 015700
994 104413
995 001414 016414

.TRPTAB:

.SCOPE SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
.SCOP1 SCOP1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
.TYPE TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
.INSTR INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
.INSTER INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
.PARAM PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
.SAV05 SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
.RES05 RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
.CONVRT CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
.CNVRT CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
.SETFLG SETFLG=TRAP+12 ;CALL TO FLAG SET ROUTINE
.CKSWR CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY

```

996      104414
997 001416 016470
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008 001420 012767 000340 176350 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1009 001426 012706 001100 MOV #STACK,SP ;SET UP STACK
1010 001432 012737 016250 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
1011 001440 005067 177460 CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
1012 001444 105067 177575 CLRB STFLG ;CLEAR START FLAG
1013 001450 005067 177454 CLR PASCNT ;CLEAR PASS COUNT
1014 001454 105067 177566 CLRB ERRFLG ;CLEAR ERROR FLAG
1015 001460 005067 177446 CLR ERRCNT ;CLEAR ERROR COUNT
1016 001464 005067 177444 CLR LSTERR ;CLEAR LAST ERROR POINTER
1017 001470 012767 000001 177430 MOV #1,TSTNO ;SET UP FOR TEST 1
1018 001476 012767 001420 177410 MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
1019 ;TESTING STARTS
1020 001504 105767 177534 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1021 001510 001004 BNE ONCE
1022 001512 104402 016570 TYPE ,MTITLE ;TYPE TITLE MESSAGE
1023 001516 105167 177522 COMB INIFLG ;IF NOT SET FLAG AND DO
1024 001522 012767 177570 177350 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1025 001530 012767 177570 177344 MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1026 001536 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
1027 001542 013746 000004 MOV @#4,-(SP)
1028 001546 012737 001566 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
1029 001554 022777 177777 177316 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1030 001562 001402 BEQ 65$
1031 001564 000407 BR 66$
1032 001566 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
1033 001570 012767 000176 177302 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1034 001576 012767 000174 177276 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
1035 001604 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
1036 001610 012637 000006 MOV (SP)+,@#6
1037 001614 005737 000042 TST @#42 ;UNDER MONITOR
1038 001620 001005 BNE MAP
1039 001622 022767 000176 177250 CMP #SWREG,SWR ;IS SWREG USED
1040 001630 001001 BNE MAP ; BRANCH TO CHECK FOR STATUS MAP
1041 001632 104414 CNTLU
1042 ;*****
1043 ; CODE FOR STATUS MAP
1044 ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1045 ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1046 ; SETUP, OR REENTER QUESTIONING ROUTINE
1047
1048 001634 032777 000200 177236 MAP: BIT #SW07,@SWR ; IS SW07=1?
1049 001642 001537 BEQ $67 ; IF NOT, GO TO TEST FOR SW00=1
1050 ; NOW SET UP MAP VALUES FOR PROGRAM
1051 ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE

```



```

1052      ; OPERATION OF THIS PROGRAM.
1053      MOVB NOSYNC      ,SYNCRNO
1054      MOVB MITSEX      ,SEXMIT
1055      MOVB RESEC        ,SEREC
1056      MOVB CLROPT       ,OPTCLR
1057      MOVB DMULT        ,MULTD
1058      MOVB BYJMR        ,JMRBY
1059      MOV  ADDBASE       ,BASEADD
1060      MOV  ADDKEEP       ,KEEPADD
1061      MOV  ADDLAST      ,LASTADD
1062      MOV  IVKEEP       ,KEEPIV
1063      MOV  REGACT        ,ACTREG
1064      MOV  ADDROT        ,ROTADD
1065      MOV  IVBASE       ,BASEIV
1066      MOV  KEEPADD      ,BASEADD
1067      MOV  RIVDU        ,DURIV
1068      MOV  RISDU        ,DURIS
1069      MOV  TIVDU        ,DUTIV
1070      MOV  TISDU        ,DUTIS
1071      MOV  L1ESS        ,LESS1
1072      MOV  @#PRTDU      ,@#DUPRT
1073      MOV      BASEDU   ,DUBASE
1074      MOV  CSRRX        ,RXCSR
1075      MOV  CSRRHX       ,HRXCSR
1076      MOV      BUFRXD   ,RXDBUF
1077      MOV      BUFRXD   ,HRXDBUF
1078      MOV      CSRRPAR  ,PARCSR
1079      MOV      CSRRPAR  ,HPARCSR
1080      MOV      CSRTX    ,TXCSR
1081      MOV      CSRTX    ,HTXCSR
1082      MOV  BUFTXD       ,TXDBUF
1083      MOV  BUFTXD       ,HTXDBUF
1084      JMP  .BEGIN
1085      ;*****
1086      $67:  BIT      #SW00 ,@SWR      ; RESELECT VECTOR $ CONTROL REG?
1087      BNE      1$
1088      JMP  .BEGIN      ; BRANCH TO QUESTIONING
1089      1$:  CLR  @#FLAG   ; GO TO LOAD STATUS MAP ETC.
1090      MOV      #300,R0   ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1091      MOV      #302,R1   ; RESTORE VECTOR AREA TO TRAPCATCHER
1092      MOV      #4,R2     ; START AT LOCATION 300
1093      MOV      R1,(R0)
1094      CLR      (R1)
1095      ADD      R2,R0
1096      ADD      R2,R1
1097      CMP      #1000,R1
1098      BLT      2$
1099      INSTR
1100      MREGAD
1101      PARAM
1102      160000
1103      167776
1104      DUBASE
1105      1
1106      1
1107      MOV      DUBASE,KEEPADD ;SAVE
;*****
; SYNC CHAR
; XMIT JUMPER
; SEC REC JUMPER
; OPTIONAL JUMPER
; MULTIPLE DEVICE
; EXTERNAL MODEM
; PROG 1ST DEVICE ADDR
; SAVED 1ST DEVICE ADDR
; LAST DEVICE RXCSR ADDR
; SAVED INTR VECTOR
; ACTIVE REGISTER
; ROTATING POINTER
; BASE INTR VECTOR
; RELOAD BASEADD
; REC INTR VECTOR
; REC INTR STATUS
; XMIT INTR VECTOR
; XMIT INTR STATUS
; PRIORITY TO ALLOW INTR
; PRIORITY RELOADED
; BRANCH TO BEGIN TESTING
; END AT LOCATION 776
; OUTPUT MESSAGE & GET INPUT STRING
; MESSAGE
; CONVERT STRING
; LOW LIMIT
; HIGH LIMIT
; STORE AT THIS LOCATION
; MASK
; HOW MANY TIMES + 2

```

1108	002240	004767	016014		JSR	PC,DUADDR	
1109	002244	016767	176736	176732	MOV	KEEPADD,BASEADD	;RESTORE FOR ROTATION
1110	002252	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1111	002254	016622			MVECTO		;MESSAGE
1112	002256	104405			PARAM		;CONVERT STRING
1113	002260	000300			300		;LOW LIMIT
1114	002262	000776			776		;HIGH LIMIT
1115	002264	020734			DURIV		;STORE AT THIS LOCATION
1116	002266	001			1		;MASK
1117	002267	004			4		;HOW MANY TIMES + 2
1118	002270	016767	016440	176716	MOV	DURIV,KEEPIV	;SAVE
1119	002276	016767	016432	176706	MOV	DURIV,BASEIV	;SET UP FOR ROTATION
1120	002304	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1121	002306	016725			MMULT		;MESSAGE
1122	002310	104412			SETFLG		;SET FLAG BASED UPON INPUT STRING
1123	002312	001202			MULTD		;THIS FLAG
1124	002314	105767	176662		TSTB	MULTD	;ARE THERE MULTIPLE DEVICES
1125							;ON THE SYSTEM ?
1126	002320	100406			BMI	BBB	;YES,ASK NEXT QUESTION
1127	002322	005067	176670		CLR	ACTREG	
1128	002326	005067	176666		CLR	ROTADD	
1129	002332	000167	000140		JMP	OUTMUL	;JUMP AROUND NEXT QUESTION
1130	002336				BBB:		
1131	002336	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
1132	002340	017004			MLASTD		;MESSAGE
1133	002342	104405			PARAM		;CONVERT STRING
1134	002344	160000			160000		;LOW LIMIT
1135	002346	167776			167776		;HIGH LIMIT
1136	002350	001210			LASTADD		;STORE AT THIS LOCATION
1137	002352	001			1		;MASK
1138	002353	001			1		;HOW MANY TIMES + 2
1139							;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1140	002354	012767	000001	176636	1\$:	MOV	#1,ROTADD ;SET UP POINTER
1141	002362	005067	176630			CLR	ACTREG ;CLR ACTIVE REGISTER
1142	002366	056767	176626	176622	2\$:	BIS	ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1143	002374	000241				CLC	
1144	002376	006167	176616			ROL	ROTADD ;SET UP POINTER
1145	002402	103421				BCS	3\$: ARE YOU OUT OF RANGE ?
1146	002404	062767	000010	176572		ADD	#10,BASEADD ;SET UP BASE ADDRESS
1147	002412	026767	176572	176564		CMP	LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1148	002420	101362				BHI	2\$: NO DO IT AGAIN
1149	002422	056767	176572	176566		BIS	ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1150							;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1151							;MULTIPLE DEVICE QUESTION
1152	002430	012767	000001	176562	4\$:	MOV	#1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1153	002436	016767	176544	176540		MOV	KEEPADD,BASEADD ;DITTO
1154	002444	000414				BR	OUTMUL ;CONTINUE QUESTIONS
1155	002446	016767	176534	176530	3\$:	MOV	KEEPADD,BASEADD ;RESTORE
1156	002454	104403				INSTR	;OUTPUT MESSAGE & GET INPUT STRING
1157	002456	017167				MRANGE	;MESSAGE
1158	002460	104405				PARAM	;CONVERT STRING
1159	002462	160000				160000	;LOW LIMIT
1160	002464	167776				167776	;HIGH LIMIT
1161	002466	001210				LASTADD	;STORE AT THIS LOCATION
1162	002470	001				1	;MASK
1163	002471	001				1	;HOW MANY TIMES + 2


```

1164 002472 000167 177656      OUTMUL: JMP 1$      ;DO IT AGAIN
1165 002476                      INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1166 002476 104403              MLEVEL         ;MESSAGE
1167 002500 017453              PARAM          ;CONVERT STRING
1168 002502 104405              4              ;LOW LIMIT
1169 002504 000004              7              ;HIGH LIMIT
1170 002506 000007              DUPRT          ;STORE AT THIS LOCATION
1171 002510 020254              .BYTE 0        ;MASK
1172 002512 000              .BYTE 1          ;HOW MANY TIMES + 2
1173 002513 001
1174 002514 004767 015464      JSR PC,DULEV
1175                      ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1176                      ;BUFFER TO THE CHARACTERS '1' AND '2'
1177                      ;IF THE CHARACTER IS '1' CLEAR THE FLAG
1178                      ;IF THE CHARACTER IS '2' SET THE FLAG
1179                      AAA:
1180 002520 104403              INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1181 002522 017500              MSYNC          ;MESSAGE
1182 002524 122767 000061 015312 3$: CMPB #'1,INBUF ;IS IT '1' ?
1183 002532 001003              BNE 1$
1184 002534 105067 176436      CLRB SYNCNO      ;000
1185 002540 000412              BR 4$
1186 002542 122767 000062 015274 1$: CMPB #'2,INBUF ;IS IT '2' ?
1187 002550 001004              BNE 2$
1188 002552 112767 177777 176416      MOVB #-1,SYNCNO ;377
1189 002560 000402              BR 4$
1190 002562 104404              2$: INSTR
1191 002564 000757              BR 3$          ;RETRY
1192 002566 000240              4$: NOP
1193 002570 104403              INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1194 002572 017546              MWIRE6         ;MESSAGE
1195 002574 104412              SETFLG        ;SET FLAG BASED UPON INPUT STRING
1196 002576 001177              SEXMIT        ;THIS FLAG
1197 002600 104403              INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1198 002602 017614              MWIRE5         ;MESSAGE
1199 002604 104412              SETFLG        ;SET FLAG BASED UPON INPUT STRING
1200 002606 001200              SEREC        ;THIS FLAG
1201 002610 104403              INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1202 002612 017661              MWIRE4         ;MESSAGE
1203 002614 104412              SETFLG        ;SET FLAG BASED UPON INPUT STRING
1204 002616 001201              OPTCLR        ;THIS FLAG
1205 002620 104403              INSTR          ;OUTPUT MESSAGE & GET INPUT STRING
1206 002622 017735              MEXTJ         ;MESSAGE
1207 002624 104412              SETFLG        ;SET FLAG BASED UPON INPUT STRING
1208 002626 001203              JMRBY         ;THIS FLAG
1209
1210
1211                      ;TEST START AND RESTART
1212
1213 002630 012767 000340 175140 .BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
1214                      ; ***** LOAD STATUS MAP *****
1215                      ; THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1216                      ; USED IN THIS PROGRAM AND WILL BE PASSED TO ANY
1217                      ; OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1218 002636 032777 000200 176234 BIT #SW07 ;@SWR ; SW07 SET , IF YES BRANCH
1219 002644 001132              BNE HEREU

```

1220	002646	116767	176324	176424
1221	002654	116767	176317	176417
1222	002662	116767	176312	176412
1223	002670	116767	176305	176405
1224	002676	116767	176300	176400
1225	002704	116767	176273	176373
1226	002712	016767	176266	176366
1227	002720	016767	176262	176362
1228	002726	016767	176256	176356
1229	002734	016767	176254	176354
1230	002742	016767	176244	176344
1231	002750	016767	176242	176342
1232	002756	016767	176236	176336
1233	002764	013737	020254	001324
1234	002772	016767	015736	176326
1235	003000	016767	015732	176326
1236	003006	016767	015726	176314
1237	003014	016767	015722	176310
1238	003022	016767	015230	176306
1239	003030	016767	015356	176326
1240	003036	016767	015646	176274
1241	003044	016767	015642	176270
1242	003052	016767	015636	176264
1243	003060	016767	015632	176260
1244	003066	016767	015626	176254
1245	003074	016767	015622	176250
1246	003102	016767	015616	176244
1247	003110	016767	015612	176240
1248	003116	016767	015606	176234
1249	003124	016767	015602	176230
1250				
1251				
1252				
1253				
1254				
1255				
1256				
1257				
1258				
1259				
1260	003132	005737	001222	
1261	003136	001402		
1262	003140	000167	000116	
1263	003144	104402	020012	
1264	003150	062737	000001	001222
1265	003156	010067	176042	
1266	003162	010167	176040	
1267	003166	012767	000003	176034
1268	003174	012700	000002	
1269	003200	012701	001300	
1270	003204	010120		
1271	003206	062701	000002	
1272	003212	020127	001306	
1273	003216	001372		
1274	003220	012700	000002	
1275	003224	010067	176006	

MOVB SYNCNO	,NOSYNC	: SYNC CHARS
MOVB SEXMIT	,MITSEX	: XMIT JUMPER
MOVB SEREC	,RESEC	: SEC REC JUMPER
MOVB OPTCLR	,CLROPT	: OPTIONAL JUMPER
MOVB MULTD	,DMULT	: MULTIPLE DEVICES
MOVB JMRBY	,BYJMR	: EXTERNAL MODEM
MOV BASEADD	,ADDBASE	: PROG CONTROLLED 1ST ADDR
MOV KEEPADD	,ADDKEEP	: SAVED 1ST DEVICE ADDR
MOV LASTADD	,ADDLAST	: LAST DEVICE RXCSR ADDR
MOV KEEPIV	,IVKEEP	: SAVED INTR VECTOR
MOV BASEIV	,IVBASE	: RELOAD BASE INTR VECTOR
MOV ACTREG	,REGACT	: ACTIVE REGISTER
MOV ROTADD	,ADDROT	: ROTATING POINTER
MOV @#DUPRT	,@#PRTDU	: DU11 PRIORITY
MOV DURIV	,RIVDU	: REC INTR VECTOR
MOV DURIS	,RISDU	: REC INTR STATUS
MOV DUTIV	,TIVDU	: XMIT INTR VECTOR
MOV DUTIS	,TISDU	: XMIT INTR STATUS
MOV LESS1	,L1ESS	: PRIORITY TO ALLOW INTR
MOV DUBASE	,BASEDU	: RXCSR BASE ADDRESS
MOV RXCSR	,CSRXX	
MOV HRXCSR	,CSRHRX	
MOV RXDBUF	,BUFRXD	
MOV HRXDBUF	,BUFHRXD	
MOV PARCSR	,CSRPAR	
MOV HPARCSR	,CSRHPAR	
MOV TXCSR	,CSRTX	
MOV HTXCSR	,CSRHTX	
MOV TXDBUF	,BUFTXD	
MOV HTXDBUF	,BUFHTXD	

 : THE FOLLOWING CODE WILL PRINT
 : THE CONVERSATIONALLY SET JUMPER
 : SETTINGS FROM THE STATUS MAP
 : ON THE FIRST PASS OF
 : THIS DIAGNOSTIC OR
 : JUST AFTER THE QUESTIONING
 : AND ANSWERING .

HEREU:	TST @#FLAG	: TEST IF 1ST PASS
	BEQ SETFG	: IF FIRST PASS SET FLAG/PRINT
	JMP THRU	: AROUND IF PASS > 1
SETFG:	TYPE MSTATUS	: PRINT 'STATUS MAP'
	ADD #1, @#FLAG	: SET FLAG ON 1ST PASS
	MOV R0, HOLD0	: SAVE R0
	MOV R1, HOLD1	: SAVE R1
	MOV #3, COUNT1	: COUNTER FOR WORDS PRINTED
	MOV #BUFF1, R0	
FILBUF:	MOV #STATUS, R1	: (BUFF1)=STATUS ETC.
	MOV R1, (R0)+	: LOAD BUFF AS ABOVE
	ADD #2, R1	: PREPARE STATUS ADDRESS
	CMP R1, #STATUS+6	: CHECK IF 3 WORDS LOADED
	BNE FILBUF	: BACK TO LOAD NEXT ADDRESS
	MOV #BUFF1, R0	: LOAD FOR PRINT OUT
UP:	MOV R0, TABLE+4	: LOAD ADDRESS TO PRINT

1276	003230	012067	176006				MOV (R0)+, TABLE+10		: LOAD CONTENTS
1277	003234	104410	001232				CONVRT, TABLE		: PRINT ADDRESS/CONTENTS PAIR
1278	003240	104402	017254				TYPE, MCRLF		: CR AND LF
1279	003244	005367	175760				DEC COUNT1		: COUNT WORDS PRINTED
1280	003250	001365					BNE UP		: GO PRINT NEXT ADDRESS/CONTENTS
1281	003252	016700	175746				MOV HOLD0, R0		
1282	003256	016701	175744				MOV HOLD1, R1		
1283							:*****		
1284									
1285	003262	012706	001100			THRU:	MOV #STACK, SP		: SET UP STACK
1286	003266	005737	000042				TST @#42		: IS PROGRAM UNDER MONITOR CONTROL
1287	003272	001056					BNE 3\$		
1288	003274	105767	175702				TSTB MULTD		: DON'T ALLOW LOCK ON TEST IF RUNNING
1289									: MULTIPLE DEVICES
1290	003300	001407					BEQ 5\$		: IF NO, TEST FOR LOCK ON TEST
1291	003302	016767	011404	011304			MOV BRW, TTST		: RESTORE NORMAL SCOPE LOOP
1292	003310	016767	011400	011300			MOV BRX, TTST+2		: DITTO
1293	003316	000444					BR 3\$		: JUMP AROUND IF YES
1294	003320	032777	000004	175552	5\$:		BIT #BIT2, @SWR		: CHECK FOR LOCK ON TEST
1295	003326	001416					BEQ 1\$		
1296	003330	104403					INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1297	003332	017410					MLOCK		: MESSAGE
1298	003334	104412					SETFLG		: SET FLAG BASED UPON INPUT STRING
1299	003336	001247					LOKFLG		: THIS FLAG
1300	003340	105767	175703				TSTB LOKFLG		: IS LOCK ON TEST OPTION SELECTED
1301	003344	001407					BEQ 1\$		
1302	003346	012767	000240	011240			MOV #NOP, TTST		
1303	003354	012767	000240	011234			MOV #NOP, TTST+2		: SET UP TO LOCK
1304	003362	000406					BR 2\$		
1305	003364	016767	011322	011222	1\$:		MOV BRW, TTST		
1306	003372	016767	011316	011216			MOV BRX, TTST+2		: LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1307	003400	032777	000002	175472	2\$:		BIT #SW01, @SWR		: IF SW01=1, GET STARTING PC
1308	003406	001410					BEQ 3\$		
1309	003410	104403					INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1310	003412	017375					MTSTPC		: MESSAGE
1311	003414	104405					PARAM		: CONVERT STRING
1312	003416	003446					TST1		: LOW LIMIT
1313	003420	013760					TLAST		: HIGH LIMIT
1314	003422	001114					RTRN		: STORE AT THIS LOCATION
1315	003424	001				.BYTE	1		: MASK
1316	003425	001				.BYTE	1		: HOW MANY TIMES + 2
1317	003426	000403					BR 4\$		
1318	003430	012767	003446	175456	3\$:		MOV #TST1, RTRN		: START AT TEST 1
1319	003436	104402	017371		4\$:		TYPE ,MR		: TYPE R
1320	003442	000177	175446				JMP @RTRN		: START TESTING
1321									
1322									
1323									
1324									
1325									
1326									
1327									
1328									
1329									
1330									
1331									

:: THIS TEST VERIFYS WORD LENGTH SELECT OF
 :: THE TRANSMITTER SECTION, IT USES THE DNA FLAG
 :: AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
 :: CORRECTLY
 :: NOTE: DNA COMES UP ON THE FIRST RISING BIT
 :: EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
 :: LOADED INTO TXDBUF
 :: MODE: SYNINT
 :: PARITY: NO PARITY
 :: LENGTH: FIVE

```

1332
1333 003446 012767 000001 175452 TST1:  MOV    #1,TSTNO      ;SAVE THIS
1334 003454 012767 003700 175434      MOV    #TST2,NEXT    ;GO TO THIS TEST WHEN THRU
1335 003462 052777 000400 015234      BIS    #MRESET,@TXCSR ;MASTER RESET
1336 003470 012777 030000 015222      MOV    #SYNINT,@PARCSR ;SET THE MODE
1337 003476 052777 000400 015220      BIS    #MRESET,@TXCSR ;MASTER RESET
1338
1339      ;SET MAINTENANCE MODE & SEND
1340      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1341 003504 012777 004020 015212      MOV    #MINT!SEND,@TXCSR
1342
1343      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1344 003512 012777 030026 015200      MOV    #SYNINT!FIVE!NOPAR!26,@PARCSR
1345 003520 016703 015200      MOV    TXCSR,R3      ;SET UP FOR ERROR MSG
1346 003524 112777 000021 015176      MOVB   #21,@TXDBUF    ;LOAD CHAR
1347 003532 012767 000021 175404      MOV    #21,TEMP1     ;SHIFTED CHAR
1348 003540 012767 000005 175372      MOV    #5,SHIFT      ;# OF SHIFTS
1349      ;POKE CLK TO GET INTO SYNCHRONIZATION
1350 003546 052777 020000 015150      BIS    #CLK,@TXCSR    ;POKE CLK UP
1351 003554 042777 020000 015142      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1352 003562 005000      1$:    CLR    R0
1353 003564 006067 175354      ROR    TEMP1      ;FORCE CARRY
1354 003570 103002      BCC    2$
1355 003572 052700 002000      BIS    #BITW,R0      ;EQUIV OF BIT WINDOW
1356 003576      2$:    BIS    #CLK,@TXCSR    ;POKE CLK UP
1357 003576 052777 020000 015120      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1358 003604 042777 020000 015112      MOV    @TXCSR,R1     ;ACTUAL
1359 003612 017701 015106      BIC    #075777,R1    ;SAVE BITW & DNA
1360 003616 042701 075777      CMP    R0,R1      ;COMPARE EXP VS ACT
1361 003622 020001      BEQ    3$
1362 003624 001401      HLT    3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1363 003626 104003      ;BIT,.....ALSO CHECK DNA
1364
1365 003630      3$:    DEC    SHIFT      ;# OF SHIFTS
1366 003630 005367 175304      BNE    1$      ;DO IT AGAIN ?
1367 003634 001352
1368      ;NOW POKE CLK TO SEE DNA
1369 003636 052777 020000 015060      BIS    #CLK,@TXCSR    ;POKE CLK
1370 003644 012700 100000      MOV    #100000,R0    ;EXPECTED
1371 003650 017701 015050      MOV    @TXCSR,R1     ;ACTUAL
1372 003654 042701 077777      BIC    #77777,R1     ;SAVE DNA ONLY
1373 003660 020001      CMP    R0,R1      ;COMPARE EXPECTED VS ACTUAL
1374 003662 001401      BEQ    4$
1375 003664 104003      HLT    3      ;DNA SHOULD BE SET
1376      ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1377      ;SELECT LOGIC OF THE TRANSMITTER
1378 003666      4$:    TST    @TXCSR    ;DNA ?
1379 003666 005777 015032      BPL    5$
1380 003672 100001      HLT
1381 003674 104000      ;DNA SHOULD NOT BE SET
1382      ;IT SHOULD HAVE BEEN CLEARED FROM
1383      ;PREVIOUS READ
1384 003676      5$:    SCOPE
1385 003676 104400      ;:THIS TEST VERIFYS WORD LENGTH SELECT OF
1386      ;:THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1387

```



```

1388                                     ::AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1389                                     ::CORRECTLY
1390                                     ::NOTE: DNA COMES UP ON THE FIRST RISING BIT
1391                                     ::EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1392                                     ::LOADED INTO TXDBUF
1393                                     ::MODE:SYNINT
1394                                     ::PARITY:NO PARITY
1395                                     ::LENGTH:SIX
1396
1397 003700 012767 000002 175220 TST2: MOV #2,TSTNO ;SAVE THIS
1398 003706 012767 004132 175202 MOV #TST3,NEXT ;GO TO THIS TEST WHEN THRU
1399 003714 052777 000400 015002 BIS #MRESET,@TXCSR ;MASTER RESET
1400 003722 012777 030000 014770 MOV #SYNINT,@PARCSR ;SET THE MODE
1401 003730 052777 000400 014766 BIS #MRESET,@TXCSR ;MASTER RESET
1402
1403 ;SET MAINTENANCE MODE & SEND
1404 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1405 003736 012777 004020 014760 MOV #MINT!SEND,@TXCSR
1406
1407 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1408 003744 012777 032026 014746 MOV #SYNINT!SIX!NOPAR!26,@PARCSR
1409 003752 016703 014746 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1410 003756 112777 000021 014744 MOV #21,@TXDBUF ;LOAD CHAR
1411 003764 012767 000021 175152 MOV #21,TEMP1 ;SHIFTED CHAR
1412 003772 012767 000006 175140 MOV #6,SHIFT ;# OF SHIFTS
1413
1414 004000 052777 020000 014716 ;POKE CLK TO GET INTO SYNCHRONIZATION
1415 004006 042777 020000 014710 BIS #CLK,@TXCSR ;POKE CLK UP
1416 004014 005000 CLR R0 ;POKE CLK DOWN
1417 004016 006067 175122 1$: ROR TEMP1 ;FORCE CARRY
1418 004022 103002 BCC 2$
1419 004024 052700 002000 BIS #BITW,R0 ;EQUIV OF BIT WINDOW
1420 004030 2$:
1421 004030 052777 020000 014666 BIS #CLK,@TXCSR ;POKE CLK UP
1422 004036 042777 020000 014660 BIC #CLK,@TXCSR ;POKE CLK DOWN
1423 004044 017701 014654 MOV @TXCSR,R1 ;ACTUAL
1424 004050 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1425 004054 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1426 004056 001401 BEQ 3$
1427 004060 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1428 ;BIT,.....ALSO CHECK DNA
1429 004062 3$:
1430 004062 005367 175052 DEC SHIFT ;# OF SHIFTS
1431 004066 001352 BNE 1$ ;DO IT AGAIN ?
1432
1433 ;NOW POKE CLK TO SEE DNA
1434 004070 052777 020000 014626 BIS #CLK,@TXCSR ;POKE CLK
1435 004076 012700 100000 MOV #100000,R0 ;EXPECTED
1436 004102 017701 014616 MOV @TXCSR,R1 ;ACTUAL
1437 004106 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1438 004112 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
1439 004114 001401 BEQ 4$
1440 004116 104003 HLT 3 ;DNA SHOULD BE SET
1441 ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1442 ;SELECT LOGIC OF THE TRANSMITTER
1443 004120 005777 014600 4$: TST @TXCSR ;DNA ?

```

1444	004124	100001			BPL	5\$	
1445	004126	104000			HLT		:DNA SHOULD NOT BE SET
1446							:IT SHOULD HAVE BEEN CLEARED FROM
1447							:PREVIOUS READ
1448	004130				5\$:		
1449	004130	104400					
1450							SCOPE
1451							::THIS TEST VERIFYS WORD LENGTH SELECT OF
1452							::THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1453							::AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1454							::CORRECTLY
1455							::NOTE: DNA COMES UP ON THE FIRST RISING BIT
1456							::EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1457							::LOADED INTO TXDBUF
1458							::MODE:SYNINT
1459							::PARITY:NO PARITY
1460							::LENGTH:SEVEN
1461	004132	012767	000003	174766	TST3:	MOV	#3,TSTNO ;SAVE THIS
1462	004140	012767	004364	174750		MOV	#TST4,NEXT ;GO TO THIS TEST WHEN THRU
1463	004146	052777	000400	014550		BIS	#MRESET,@TXCSR ;MASTER RESET
1464	004154	012777	030000	014536		MOV	#SYNINT,@PARCSR ;SET THE MODE
1465	004162	052777	000400	014534		BIS	#MRESET,@TXCSR ;MASTER RESET
1466							
1467							:SET MAINTENANCE MODE & SEND
1468							:NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1469	004170	012777	004020	014526		MOV	#MINT!SEND,@TXCSR
1470							
1471							:SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1472	004176	012777	034026	014514		MOV	#SYNINT!SEVEN!NOPAR!26,@PARCSR
1473	004204	016703	014514			MOV	TXCSR,R3 ;SET UP FOR ERROR MSG
1474	004210	112777	000021	014512		MOVB	#21,@TXDBUF ;LOAD CHAR
1475	004216	012767	000021	174720		MOV	#21,TEMP1 ;SHIFTED CHAR
1476	004224	012767	000007	174706		MOV	#7,SHIFT ;# OF SHIFTS
1477							:POKE CLK TO GET INTO SYNCRONIZATION
1478	004232	052777	020000	014464		BIS	#CLK,@TXCSR ;POKE CLK UP
1479	004240	042777	020000	014456		BIC	#CLK,@TXCSR ;POKE CLK DOWN
1480	004246	005000			1\$:	CLR	R0
1481	004250	006067	174670			ROR	TEMP1 ;FORCE CARRY
1482	004254	103002				BCC	2\$
1483	004256	052700	002000			BIS	#BITW,R0 ;EQUIV OF BIT WINDOW
1484	004262				2\$:		
1485	004262	052777	020000	014434		BIS	#CLK,@TXCSR ;POKE CLK UP
1486	004270	042777	020000	014426		BIC	#CLK,@TXCSR ;POKE CLK DOWN
1487	004276	017701	014422			MOV	@TXCSR,R1 ;ACTUAL
1488	004302	042701	075777			BIC	#075777,R1 ;SAVE BITW & DNA
1489	004306	020001				CMP	R0,R1 ;COMPARE EXP VS ACT
1490	004310	001401				BEQ	3\$
1491	004312	104003				HLT	3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1492							:BIT,.....ALSO CHECK DNA
1493	004314				3\$:		
1494	004314	005367	174620			DEC	SHIFT ;# OF SHIFTS
1495	004320	001352				BNE	1\$;DO IT AGAIN ?
1496							:NOW POKE CLK TO SEE DNA
1497	004322	052777	020000	014374		BIS	#CLK,@TXCSR ;POKE CLK
1498	004330	012700	100000			MOV	#100000,R0 ;EXPECTED
1499	004334	017701	014364			MOV	@TXCSR,R1 ;ACTUAL


```

1500 004340 042701 077777      BIC    #77777,R1      ;SAVE DNA ONLY
1501 004344 020001      CMP    R0,R1      ;COMPARE EXPECTED VS ACTUAL
1502 004346 001401      BEQ    4$
1503 004350 104003      HLT
1504                                     ;DNA SHOULD BE SET
1505                                     ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1506                                     ;SELECT LOGIC OF THE TRANSMITTER
1506 004352                                     4$:
1507 004352 005777 014346      TST    @TXCSR      ;DNA ?
1508 004356 100001      BPL    5$
1509 004360 104000      HLT
1510                                     ;DNA SHOULD NOT BE SET
1511                                     ;IT SHOULD HAVE BEEN CLEARED FROM
1512                                     ;PREVIOUS READ
1512 004362                                     5$:
1513 004362 104400
1514      SCOPE
1515      ::THIS TEST VERIFYS WORD LENGTH SELECT OF
1516      ::THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1517      ::AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1518      ::CORRECTLY
1519      ::NOTE: DNA COMES UP ON THE FIRST RISING BIT
1520      ::EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1521      ::LOADED INTO TXDBUF
1522      ::MODE:SYNINT
1523      ::PARITY:NO PARITY
1524      ::LENGTH:EIGHT
1525 004364 012767 000004 174534 TST4: MOV    #4,TSTNO      ;SAVE THIS
1526 004372 012767 004616 174516      MOV    #TST5,NEXT      ;GO TO THIS TEST WHEN THRU
1527 004400 052777 000400 014316      BIS    #MRESET,@TXCSR ;MASTER RESET
1528 004406 012777 030000 014304      MOV    #SYNINT,@PARCSR ;SET THE MODE
1529 004414 052777 000400 014302      BIS    #MRESET,@TXCSR ;MASTER RESET
1530
1531      ;SET MAINTENANCE MODE & SEND
1532      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1533 004422 012777 004020 014274      MOV    #MINT!SEND,@TXCSR
1534
1535      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1536 004430 012777 036026 014262      MOV    #SYNINT!EIGHT!NOPAR!26,@PARCSR
1537 004436 016703 014262      MOV    TXCSR,R3      ;SET UP FOR ERROR MSG
1538 004442 112777 000021 014260      MOV    #21,@TXDBUF      ;LOAD CHAR
1539 004450 012767 000021 174466      MOV    #21,TEMP1      ;SHIFTED CHAR
1540 004456 012767 000010 174454      MOV    #8,SHIFT      ;# OF SHIFTS
1541      ;POKE CLK TO GET INTO SYNCHRONIZATION
1542 004464 052777 020000 014232      BIS    #CLK,@TXCSR      ;POKE CLK UP
1543 004472 042777 020000 014224      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1544 004500 005000      1$:
1545 004502 006067 174436      CLR    R0
1546 004506 103002      ROR    TEMP1      ;FORCE CARRY
1547 004510 052700 002000      BCC    2$
1548 004514      BIS    #BITW,R0      ;EQUIV OF BIT WINDOW
1549 004514 052777 020000 014202      BIS    #CLK,@TXCSR      ;POKE CLK UP
1550 004522 042777 020000 014174      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1551 004530 017701 014170      MOV    @TXCSR,R1      ;ACTUAL
1552 004534 042701 075777      BIC    #075777,R1      ;SAVE BITW & DNA
1553 004540 020001      CMP    R0,R1      ;COMPARE EXP VS ACT
1554 004542 001401      BEQ    3$
1555 004544 104003      HLT
                                     ;BIT WINDOW DID NOT MATCH ACTUAL DATA

```

```

1556                                     :BIT,.....ALSO CHECK DNA
1557 004546                               3$:
1558 004546 005367 174366                DEC      SHIFT  :# OF SHIFTS
1559 004552 001352                        BNE      1$      :DO IT AGAIN ?
1560                                     :NOW POKE CLK TO SEE DNA
1561 004554 052777 020000 014142        BIS      #CLK,@TXCSR :POKE CLK
1562 004562 012700 100000                MOV      #100000,R0 :EXPECTED
1563 004566 017701 014132                MOV      @TXCSR,R1 :ACTUAL
1564 004572 042701 077777                BIC      #77777,R1 :SAVE DNA ONLY
1565 004576 020001                        CMP      R0,R1 :COMPARE EXPECTED VS ACTUAL
1566 004600 001401                        BEQ      4$
1567 004602 104003                        HLT      3
1568                                     :DNA SHOULD BE SET
1569                                     :IF DNA DID NOT SET ,CHECK WORD LENGTH
1570                                     :SELECT LOGIC OF THE TRANSMITTER
1571 004604 005777 014114                4$:      TST      @TXCSR :DNA ?
1572 004610 100001                        BPL      5$
1573 004612 104000                        HLT
1574                                     :DNA SHOULD NOT BE SET
1575                                     :IT SHOULD HAVE BEEN CLEARED FROM
1576                                     :PREVIOUS READ
1577 004614 104400                5$:
1578                                     SCOPE
1579                                     ::THIS TEST VERIFYS WORD LENGTH SELECT OF
1580                                     ::THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1581                                     ::AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1582                                     ::CORRECTLY
1583                                     ::NOTE: DNA COMES UP ON THE FIRST RISING BIT
1584                                     ::EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1585                                     ::LOADED INTO TXDBUF
1586                                     ::MODE:SYNEXT
1587                                     ::PARITY:NO PARITY
1588                                     ::LENGTH:FIVE
1589 004616 012767 000005 174302        TST5:  MOV      #5,TSTNO :SAVE THIS
1590 004624 012767 005050 174264        MOV      #TST6,NEXT :GO TO THIS TEST WHEN THRU
1591 004632 052777 000400 014064        BIS      #MRESET,@TXCSR :MASTER RESET
1592 004640 012777 020000 014052        MOV      #SYNEXT,@PARCSR :SET THE MODE
1593 004646 052777 000400 014050        BIS      #MRESET,@TXCSR :MASTER RESET
1594
1595                                     :SET MAINTENANCE MODE & SEND
1596                                     :NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1597 004654 012777 004020 014042        MOV      #MINT!SEND,@TXCSR
1598
1599                                     :SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1600 004662 012777 020026 014030        MOV      #SYNEXT!FIVE!NOPAR!26,@PARCSR
1601 004670 016703 014030                MOV      TXCSR,R3 :SET UP FOR ERROR MSG
1602 004674 112777 000021 014026        MOV      #21,@TXDBUF :LOAD CHAR
1603 004702 012767 000021 174234        MOV      #21,TEMP1 :SHIFTED CHAR
1604 004710 012767 000005 174222        MOV      #5,SHIFT :# OF SHIFTS
1605                                     :POKE CLK TO GET INTO SYNCRONIZATION
1606 004716 052777 020000 014000        BIS      #CLK,@TXCSR :POKE CLK UP
1607 004724 042777 020000 013772        BIC      #CLK,@TXCSR :POKE CLK DOWN
1608 004732 005000                1$:      CLR      R0
1609 004734 006067 174204                ROR      TEMP1 :FORCE CARRY
1610 004740 103002                BCC      2$
1611 004742 052700 002000                BIS      #BITW,R0 :EQUIV OF BIT WINDOW

```



```

1612 004746
1613 004746 052777 020000 013750
1614 004754 042777 020000 013742
1615 004762 017701 013736
1616 004766 042701 075777
1617 004772 020001
1618 004774 001401
1619 004776 104003
1620
1621 005000
1622 005000 005367 174134
1623 005004 001352
1624
1625 005006 052777 020000 013710
1626 005014 012700 100000
1627 005020 017701 013700
1628 005024 042701 077777
1629 005030 020001
1630 005032 001401
1631 005034 104003
1632
1633
1634 005036
1635 005036 005777 013662
1636 005042 100001
1637 005044 104000
1638
1639
1640 005046
1641 005046 104400
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653 005050 012767 000006 174050
1654 005056 012767 005302 174032
1655 005064 052777 000400 013632
1656 005072 012777 020000 013620
1657 005100 052777 000400 013616
1658
1659
1660
1661 005106 012777 004020 013610
1662
1663
1664 005114 012777 022026 013576
1665 005122 016703 013576
1666 005126 112777 000021 013574
1667 005134 012767 000021 174002

2$:
BIS #CLK,@TXCSR ;POKE CLK UP
BIC #CLK,@TXCSR ;POKE CLK DOWN
MOV @TXCSR,R1 ;ACTUAL
BIC #075777,R1 ;SAVE BITW & DNA
CMP R0,R1 ;COMPARE EXP VS ACT
BEQ 3$
HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
;BIT,.....ALSO CHECK DNA

3$:
DEC SHIFT ;# OF SHIFTS
BNE 1$ ;DO IT AGAIN ?
;NOW POKE CLK TO SEE DNA
BIS #CLK,@TXCSR ;POKE CLK
MOV #100000,R0 ;EXPECTED
MOV @TXCSR,R1 ;ACTUAL
BIC #77777,R1 ;SAVE DNA ONLY
CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
BEQ 4$
HLT 3 ;DNA SHOULD BE SET
;IF DNA DID NOT SET ,CHECK WORD LENGTH
;SELECT LOGIC OF THE TRANSMITTER

4$:
TST @TXCSR ;DNA ?
BPL 5$
HLT ;DNA SHOULD NOT BE SET
;IT SHOULD HAVE BEEN CLEARED FROM
;PREVIOUS READ

5$:
SCOPE
;;THIS TEST VERIFYS WORD LENGTH SELECT OF
;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
;;CORRECTLY
;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
;;LOADED INTO TXDBUF
;;MODE:SYNEXT
;;PARITY:NO PARITY
;;LENGTH:SIX
TST6: MOV #6,TSTNO ;SAVE THIS
MOV #TST7,NEXT ;GO TO THIS TEST WHEN THRU
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #SYNEXT,@PARCSR ;SET THE MODE
BIS #MRESET,@TXCSR ;MASTER RESET

;SET MAINTENANCE MODE & SEND
;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
MOV #MINT!SEND,@TXCSR

;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
MOV #SYNEXT!SIX!NOPAR!26,@PARCSR
MOV TXCSR,R3 ;SET UP FOR ERROR MSG
MOVB #21,@TXDBUF ;LOAD CHAR
MOV #21,TEMP1 ;SHIFTED CHAR

```

```

1668 005142 012767 000006 173770      MOV      #6,SHIFT      ;# OF SHIFTS
1669      ;POKE CLK TO GET INTO SYNCRONIZATION
1670 005150 052777 020000 013546      BIS      #CLK,@TXCSR    ;POKE CLK UP
1671 005156 042777 020000 013540      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
1672 005164 005000      1$:      CLR      R0
1673 005166 006067 173752      ROR      TEMP1      ;FORCE CARRY
1674 005172 103002      2$:      BCC      2$
1675 005174 052700 002000      BIS      #BITW,R0      ;EQUIV OF BIT WINDOW
1676 005200      2$:      BIS      #CLK,@TXCSR    ;POKE CLK UP
1677 005200 052777 020000 013516      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
1678 005206 042777 020000 013510      MOV      @TXCSR,R1      ;ACTUAL
1679 005214 017701 013504      BIC      #075777,R1     ;SAVE BITW & DNA
1680 005220 042701 075777      CMP      R0,R1      ;COMPARE EXP VS ACT
1681 005224 020001      BEQ      3$
1682 005226 001401      HLT      3
1683 005230 104003      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1684      ;BIT,.....ALSO CHECK DNA
1685 005232      3$:      DEC      SHIFT      ;# OF SHIFTS
1686 005232 005367 173702      BNE      1$      ;DO IT AGAIN ?
1687 005236 001352      ;NOW POKE CLK TO SEE DNA
1688      BIS      #CLK,@TXCSR    ;POKE CLK
1689 005240 052777 020000 013456      MOV      #100000,R0     ;EXPECTED
1690 005246 012700 100000      MOV      @TXCSR,R1      ;ACTUAL
1691 005252 017701 013446      BIC      #77777,R1      ;SAVE DNA ONLY
1692 005256 042701 077777      CMP      R0,R1      ;COMPARE EXPECTED VS ACTUAL
1693 005262 020001      BEQ      4$
1694 005264 001401      HLT      3
1695 005266 104003      ;DNA SHOULD BE SET
1696      ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1697      ;SELECT LOGIC OF THE TRANSMITTER
1698 005270      4$:      TST      @TXCSR    ;DNA ?
1699 005270 005777 013430      BPL      5$
1700 005274 100001      HLT
1701 005276 104000      ;DNA SHOULD NOT BE SET
1702      ;IT SHOULD HAVE BEEN CLEARED FROM
1703      ;PREVIOUS READ
1704 005300      5$:
1705 005300 104400      SCOPE
1706      ::THIS TEST VERIFYS WORD LENGTH SELECT OF
1707      ::THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1708      ::AND BIT WINDOW TO DETEPMINE THAT IT WAS SELECTED
1709      ::CORRECTLY
1710      ::NOTE: DNA COMES UP ON THE FIRST RISING BIT
1711      ::EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1712      ::LOADED INTO TXDBUF
1713      ::MODE:SYNEXT
1714      ::PARITY:NO PARITY
1715      ::LENGTH:SEVEN
1716      ::
1717 005302 012767 000007 173616      TST7:  MOV      #7,TSTNO      ;SAVE THIS
1718 005310 012767 005534 173600      MOV      #TST8,NEXT      ;GO TO THIS TEST WHEN THRU
1719 005316 052777 000400 013400      BIS      #MRESET,@TXCSR ;MASTER RESET
1720 005324 012777 020000 013366      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1721 005332 052777 000400 013364      BIS      #MRESET,@TXCSR ;MASTER RESET
1722
1723      ;SET MAINTENANCE MODE & SEND

```



```

1724
1725 005340 012777 004020 013356      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1726      MOV      #MINT!SEND,@TXCSR
1727
1728 005346 012777 024026 013344      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1729 005354 016703 013344 013344      MOV      #SYNEXT!SEVEN!NOPAR!26,@PARCSR
1730 005360 112777 000021 013342      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
1731 005366 012767 000021 173550      MOV      #21,@TXDBUF   ;LOAD CHAR
1732 005374 012767 000007 173536      MOV      #21,TEMP1     ;SHIFTED CHAR
1733      MOV      #7,SHIFT ;# OF SHIFTS
1734 005402 052777 020000 013314      ;POKE CLK TO GET INTO SYNCRONIZATION
1735 005410 042777 020000 013306      BIS      #CLK,@TXCSR   ;POKE CLK UP
1736 005416 005000      173520      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
1737 005420 006067      173520      1$:      CLR      R0
1738 005424 103002      BCC      TEMP1 ;FORCE CARRY
1739 005426 052700 002000      BIS      #BITW,R0      ;EQUIV OF BIT WINDOW
1740 005432
1741 005432 052777 020000 013264      2$:      BIS      #CLK,@TXCSR   ;POKE CLK UP
1742 005440 042777 020000 013256      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
1743 005446 017701 013252      MOV      @TXCSR,R1     ;ACTUAL
1744 005452 042701 075777      BIC      #075777,R1    ;SAVE BITW & DNA
1745 005456 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
1746 005460 001401      BEQ      3$
1747 005462 104003      HLT      3
1748      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1749      ;BIT,.....ALSO CHECK DNA
1750 005464 005367 173450      3$:      DEC      SHIFT      ;# OF SHIFTS
1751 005470 001352      BNE      1$      ;DO IT AGAIN ?
1752
1753 005472 052777 020000 013224      ;NOW POKE CLK TO SEE DNA
1754 005500 012700 100000      BIS      #CLK,@TXCSR   ;POKE CLK
1755 005504 017701 013214      MOV      #100000,R0    ;EXPECTED
1756 005510 042701 077777      MOV      @TXCSR,R1     ;ACTUAL
1757 005514 020001      BIC      #77777,R1    ;SAVE DNA ONLY
1758 005516 001401      CMP      R0,R1      ;COMPARE EXPECTED VS ACTUAL
1759 005520 104003      BEQ      4$
1760      HLT      3
1761      ;DNA SHOULD BE SET
1762      ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1763      ;SELECT LOGIC OF THE TRANSMITTER
1764 005522
1765 005522 005777 013176      4$:      TST      @TXCSR ;DNA ?
1766 005526 100001      BPL      5$
1767 005530 104000      HLT
1768      ;DNA SHOULD NOT BE SET
1769      ;IT SHOULD HAVE BEEN CLEARED FROM
1770      ;PREVIOUS READ
1771 005532 104400      5$:
1772      SCOPE
1773      ;;THIS TEST VERIFYS WORD LENGTH SELECT OF
1774      ;;THE TRANSMITTER SECTION,IT USES THE DNA FLAG
1775      ;;AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1776      ;;CORRECTLY
1777      ;;NOTE: DNA COMES UP ON THE FIRST RISING BIT
1778      ;;EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1779      ;;LOADED INTO TXDBUF
1780      ;;MODE:SYNEXT
1781      ;;PARITY:NO PARITY
1782      ;;LENGTH:EIGHT

```

```

1780
1781 005534 012767 000010 173364 TST8: MOV #8,TSTNO ;SAVE THIS
1782 005542 012767 005766 173346 MOV #TST9,NEXT ;GO TO THIS TEST WHEN THRU
1783 005550 052777 000400 013146 BIS #MRESET,@TXCSR ;MASTER RESET
1784 005556 012777 020000 013134 MOV #SYNEXT,@PARCSR ;SET THE MODE
1785 005564 052777 000400 013132 BIS #MRESET,@TXCSR ;MASTER RESET
1786
1787 ;SET MAINTENANCE MODE & SEND
1788 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1789 005572 012777 004020 013124 MOV #MINT!SEND,@TXCSR
1790
1791 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1792 005600 012777 026026 013112 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1793 005606 016703 013112 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1794 005612 112777 000021 013110 MOV #21,@TXDBUF ;LOAD CHAR
1795 005620 012767 000021 173316 MOV #21,TEMP1 ;SHIFTED CHAR
1796 005626 012767 000010 173304 MOV #8,SHIFT ;# OF SHIFTS
1797 ;POKE CLK TO GET INTO SYNCHRONIZATION
1798 005634 052777 020000 013062 BIS #CLK,@TXCSR ;POKE CLK UP
1799 005642 042777 020000 013054 BIC #CLK,@TXCSR ;POKE CLK DOWN
1800 005650 005000 1$: CLR R0
1801 005652 006067 173266 ROR TEMP1 ;FORCE CARRY
1802 005656 103002 2$: BCC 2$
1803 005660 052700 002000 BIS #BITW,R0 ;EQUIV OF BIT WINDOW
1804 005664
1805 005664 052777 020000 013032 BIS #CLK,@TXCSR ;POKE CLK UP
1806 005672 042777 020000 013024 BIC #CLK,@TXCSR ;POKE CLK DOWN
1807 005700 017701 013020 MOV @TXCSR,R1 ;ACTUAL
1808 005704 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1809 005710 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1810 005712 001401 BEQ 3$
1811 005714 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1812 ;BIT,.....ALSO CHECK DNA
1813 005716 3$: DEC SHIFT ;# OF SHIFTS
1814 005716 005367 173216 BNE 1$ ;DO IT AGAIN ?
1815 005722 001352
1816 ;NOW POKE CLK TO SEE DNA
1817 005724 052777 020000 012772 BIS #CLK,@TXCSR ;POKE CLK
1818 005732 012700 100000 MOV #100000,R0 ;EXPECTED
1819 005736 017701 012762 MOV @TXCSR,R1 ;ACTUAL
1820 005742 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1821 005746 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
1822 005750 001401 BEQ 4$
1823 005752 104003 HLT 3 ;DNA SHOULD BE SET
1824 ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1825 ;SELECT LOGIC OF THE TRANSMITTER
1826 005754 4$: TST @TXCSR ;DNA ?
1827 005754 005777 012744 BPL 5$
1828 005760 100001 HLT ;DNA SHOULD NOT BE SET
1829 005762 104000 ;IT SHOULD HAVE BEEN CLEARED FROM
1830 ;PREVIOUS READ
1831
1832 005764 5$: SCOPE
1833 005764 104400 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1834 ;:OF THE TRANSMITTER SECTION.
1835

```



```

1836                                     ::IT ALSO CHECKS DNA TIMING
1837                                     ::MODE:SYNINT
1838                                     ::LENGTH:FIVE PLUS PARITY
1839                                     ::PARITY:EVEPAR
1840                                     ::CHARACTER:25
1841
1842 005766 012767 000011 173132 TST9: MOV #9,TSTNO ;SAVE THIS
1843 005774 012767 006210 173114 MOV #TST10,NEXT ;GO TO THIS TEST WHEN THRU
1844 006002 052777 000400 012714 BIS #MRESET,@TXCSR ;MASTER RESET
1845 006010 012777 030000 012702 MOV #SYNINT,@PARCSR ;SET THE MODE
1846 006016 052777 000400 012700 BIS #MRESET,@TXCSR ;MASTER RESET
1847
1848 ;SET MAINTENANCE MODE & SEND
1849 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1850 006024 012777 004020 012672 MOV #MINT!SEND,@TXCSR
1851
1852 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1853 006032 012777 031426 012660 MOV #SYNINT!FIVE!EVEPAR!26,@PARCSR
1854 006040 016703 012660 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1855 006044 112777 000025 012656 MOV #25,@TXDBUF ;LOAD DATA CHAR
1856 006052 012767 000065 173064 MOV #65,TEMP1 ;TO BE SHIFTED CHAR
1857 006060 012767 000006 173052 MOV #6,SHIFT ;# OF SHIFTS
1858
1859 ;POKE CLK TO GET INTO SYNCRONIZATION
1860 006074 042777 020000 012630 BIS #CLK,@TXCSR ;POKE CLK UP
1861 006102 005000 020000 012622 BIC #CLK,@TXCSR ;POKE CLK DOWN
1862 006104 006067 173034 1$: CLR R0
1863 006110 103002 ROR TEMP1 ;FORCE CARRY
1864 006112 052700 002000 BCC 2$ ;BR IF CARRY CLR
1865 006116 BIS #BITW,R0 ;EQUIV OF BITW
1866 006116 052777 020000 012600 2$: BIS #CLK,@TXCSR ;POKE CLK UP
1867 006124 042777 020000 012572 BIC #CLK,@TXCSR ;POKE CLK DOWN
1868 006132 017701 012566 MOV @TXCSR,R1 ;ACTUAL
1869 006136 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1870 006142 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1871 006144 001401 BEQ 3$
1872 006146 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1873 ;BIT,...ALSO CHECK DNA
1874 006150 3$: DEC SHIFT ;# OF SHIFTS
1875 006150 005367 172764 BNE 1$ ;DO IT AGAIN ?
1876 006154 001352
1877
1878 ;NOW POKE CLK TO SEE DNA
1879 006156 052777 020000 012540 BIS #CLK,@TXCSR ;POKE CLK
1880 006164 012700 100000 MOV #100000,R0 ;EXPECTED
1881 006170 017701 012530 MOV @TXCSR,R1 ;ACTUAL
1882 006174 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1883 006200 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1884 006202 001401 BEQ 4$
1885 006204 104003 HLT 3 ;DNA SHOULD BE SET
1886 ;IF DNA DID NOT SET
1887 006206 4$: SCOPE ;CHECK WORD LENGTH SELECT LOGIC
1888 006206 104400
1889
1890 ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1891 ::OF THE TRANSMITTER SECTION.
::IT ALSO CHECKS DNA TIMING

```

```

1892      ::MODE:SYNINT
1893      ::LENGTH:FIVE PLUS PARITY
1894      ::PARITY:ODDPAR
1895      ::CHARACTER:25
1896
1897 006210 012767 000012 172710 TST10: MOV #10,TSTNO ;SAVE THIS
1898 006216 012767 006432 172672      MOV #TST11,NEXT ;GO TO THIS TEST WHEN THRU
1899 006224 052777 000400 012472      BIS #MRESET,@TXCSR ;MASTER RESET
1900 006232 012777 030000 012460      MOV #SYNINT,@PARCSR ;SET THE MODE
1901 006240 052777 000400 012456      BIS #MRESET,@TXCSR ;MASTER RESET
1902
1903      ;SET MAINTENANCE MODE & SEND
1904      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1905 006246 012777 004020 012450      MOV #MINT!SEND,@TXCSR
1906
1907      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1908 006254 012777 031026 012436      MOV #SYNINT!FIVE!ODDPAR!26,@PARCSR
1909 006262 016703 012436      MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1910 006266 112777 000025 012434      MOV #25,@TXDBUF ;LOAD DATA CHAR
1911 006274 012767 000025 172642      MOV #25,TEMP1 ;TO BE SHIFTED CHAR
1912 006302 012767 000006 172630      MOV #6,SHIFT ;# OF SHIFTS
1913
1914 006310 052777 020000 012406      ;POKE CLK TO GET INTO SYNCHRONIZATION
1915 006316 042777 020000 012400      BIS #CLK,@TXCSR ;POKE CLK UP
1916 006324 005000      BIC #CLK,@TXCSR ;POKE CLK DOWN
1917 006326 006067 172612      1$: CLR R0
1918 006332 103002      ROR TEMP1 ;FORCE CARRY
1919 006334 052700 002000      BCC 2$ ;BR IF CARRY CLR
1920 006340      BIS #BITW,R0 ;EQUIV OF BITW
1921 006340 052777 020000 012356      2$: BIS #CLK,@TXCSR ;POKE CLK UP
1922 006346 042777 020000 012350      BIC #CLK,@TXCSR ;POKE CLK DOWN
1923 006354 017701 012344      MOV @TXCSR,R1 ;ACTUAL
1924 006360 042701 075777      BIC #075777,R1 ;SAVE BITW & DNA
1925 006364 020001      CMP R0,R1 ;COMPARE EXP VS ACT
1926 006366 001401      BEQ 3$
1927 006370 104003      HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1928      ;BIT,...ALSO CHECK DNA
1929 006372      3$: DEC SHIFT ;# OF SHIFTS
1930 006372 005367 172542      BNE 1$ ;DO IT AGAIN ?
1931 006376 001352
1932
1933 006400 052777 020000 012316      ;NOW POKE CLK TO SEE DNA
1934 006406 012700 100000      BIS #CLK,@TXCSR ;POKE CLK
1935 006412 017701 012306      MOV #100000,R0 ;EXPECTED
1936 006416 042701 077777      MOV @TXCSR,R1 ;ACTUAL
1937 006422 020001      BIC #77777,R1 ;SAVE DNA ONLY
1938 006424 001401      CMP R0,R1 ;COMPARE EXP VS ACT
1939 006426 104003      BEQ 4$
1940      HLT 3 ;DNA SHOULD BE SET
1941      ;IF DNA DID NOT SET
1942      ;CHECK WORD LENGTH SELECT LOGIC
1943 006430      4$:
1944 006430 104400
1945
1946      SCOPE
1947      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
      ::OF THE TRANSMITTER SECTION.
      ::IT ALSO CHECKS DNA TIMING
      ::MODE:ISYMOD

```



```

1948
1949
1950
1951
1952 006432 012767 000013 172466 TST11: MOV #11,TSTNO ;SAVE THIS
1953 006440 012767 006654 172450 MOV #TST12,NEXT ;GO TO THIS TEST WHEN THRU
1954 006446 052777 000400 012250 BIS #MRESET,@TXCSR ;MASTER RESET
1955 006454 012777 000000 012236 MOV #ISYMOD,@PARCSR ;SET THE MODE
1956 006462 052777 000400 012234 BIS #MRESET,@TXCSR ;MASTER RESET
1957
1958 ;SET MAINTENANCE MODE & SEND
1959 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1960 006470 012777 004020 012226 MOV #MINT!SEND,@TXCSR
1961
1962 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1963 006476 012777 001426 012214 MOV #ISYMOD!FIVE!EVEPAR!26,@PARCSR
1964 006504 016703 012214 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1965 006510 112777 000025 012212 MOV #25,@TXDBUF ;LOAD DATA CHAR
1966 006516 012767 000352 172420 MOV #352,TEMP1 ;TO BE SHIFTED CHAR
1967 006524 012767 000010 172406 MOV #8,,SHIFT ;# OF SHIFTS
1968
1969 ;POKE CLK TO GET INTO SYNCRONIZATION
1970 006540 042777 020000 012164 BIS #CLK,@TXCSR ;POKE CLK UP
1971 006546 005000 020000 012156 BIC #CLK,@TXCSR ;POKE CLK DOWN
1972 006550 006067 172370 1$: CLR R0
1973 006554 103002 ROR TEMP1 ;FORCE CARRY
1974 006556 052700 002000 BCC 2$ ;BR IF CARRY CLR
1975 006562 2$: BIS #BITW,R0 ;EQUIV OF BITW
1976 006562 052777 020000 012134 BIS #CLK,@TXCSR ;POKE CLK UP
1977 006570 042777 020000 012126 BIC #CLK,@TXCSR ;POKE CLK DOWN
1978 006576 017701 012122 MOV @TXCSR,R1 ;ACTUAL
1979 006602 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1980 006606 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1981 006610 001401 BEQ 3$
1982 006612 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1983 ;BIT,...ALSO CHECK DNA
1984 006614 3$: DEC SHIFT ;# OF SHIFTS
1985 006614 005367 172320 BNE 1$ ;DO IT AGAIN ?
1986 006620 001352
1987 ;NOW POKE CLK TO SEE DNA
1988 006622 052777 020000 012074 BIS #CLK,@TXCSR ;POKE CLK
1989 006630 012700 000000 MOV #0,R0 ;EXPECTED
1990 006634 017701 012064 MOV @TXCSR,R1 ;ACTUAL
1991 006640 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1992 006644 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1993 006646 001401 BEQ 4$
1994 006650 104003 HLT 3 ;DNA SHOULD BE SET
1995 ;IF DNA DID NOT SET
1996 ;CHECK WORD LENGTH SELECT LOGIC
1997 006652 4$:
1998 006652 104400
1999
2000 SCOPE
2001 ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2002 ;;OF THE TRANSMITTER SECTION.
2003 ;;IT ALSO CHECKS DNA TIMING
2004 ;;MODE:ISYMOD
2005 ;;LENGTH:FIVE PLUS PARITY

```

```

2004      :::PARITY:ODDPAR
2005      :::CHARACTER:25
2006      :::
2007      006654 012767 000014 172244 TST12: MOV #12,TSTNO ;SAVE THIS
2008      006662 012767 007076 172226      MOV #TST13,NEXT ;GO TO THIS TEST WHEN THRU
2009      006670 052777 000400 012026      BIS #MRESET,@TXCSR ;MASTER RESET
2010      006676 012777 000000 012014      MOV #ISYMOD,@PARCSR ;SET THE MODE
2011      006704 052777 000400 012012      BIS #MRESET,@TXCSR ;MASTER RESET
2012
2013      ;SET MAINTENANCE MODE & SEND
2014      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2015      006712 012777 004020 012004      MOV #MINT!SEND,@TXCSR
2016
2017      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2018      006720 012777 001026 011772      MOV #ISYMOD!FIVE!ODDPAR!26,@PARCSR
2019      006726 016703 011772      MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2020      006732 112777 000025 011770      MOV #25,@TXDBUF ;LOAD DATA CHAR
2021      006740 012767 000252 172176      MOV #252,TEMP1 ;TO BE SHIFTED CHAR
2022      006746 012767 000010 172164      MOV #8,SHIFT ;# OF SHIFTS
2023      ;POKE CLK TO GET INTO SYNCHRONIZATION
2024      006754 052777 020000 011742      BIS #CLK,@TXCSR ;POKE CLK UP
2025      006762 052777 020000 011734      BIC #CLK,@TXCSR ;POKE CLK DOWN
2026      006770 005000      1$: CLR R0
2027      006772 006067 172146      ROR TEMP1 ;FORCE CARRY
2028      006776 103002      BCC 2$ ;BR IF CARRY CLR
2029      007000 052700 002000      BIS #BITW,R0 ;EQUIV OF BITW
2030      007004      2$:
2031      007004 052777 020000 011712      BIS #CLK,@TXCSR ;POKE CLK UP
2032      007012 042777 020000 011704      BIC #CLK,@TXCSR ;POKE CLK DOWN
2033      007020 017701 011700      MOV @TXCSR,R1 ;ACTUAL
2034      007024 042701 075777      BIC #075777,R1 ;SAVE BITW & DNA
2035      007030 020001      CMP R0,R1 ;COMPARE EXP VS ACT
2036      007032 001401      BEQ 3$
2037      007034 104003      HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2038      ;BIT,...ALSO CHECK DNA
2039      007036      3$:
2040      007036 005367 172076      DEC SHIFT ;# OF SHIFTS
2041      007042 001352      BNE 1$ ;DO IT AGAIN ?
2042      ;NOW POKE CLK TO SEE DNA
2043      007044 052777 020000 011652      BIS #CLK,@TXCSR ;POKE CLK
2044      007052 012700 000000      MOV #0,R0 ;EXPECTED
2045      007056 017701 011642      MOV @TXCSR,R1 ;ACTUAL
2046      007062 042701 077777      BIC #77777,R1 ;SAVE DNA ONLY
2047      007066 020001      CMP R0,R1 ;COMPARE EXP VS ACT
2048      007070 001401      BEQ 4$
2049      007072 104003      HLT 3 ;DNA SHOULD BE SET
2050      ;IF DNA DID NOT SET
2051      ;CHECK WORD LENGTH SELECT LOGIC
2052      007074      4$:
2053      007074 104400      SCOPE
2054
2055      :::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2056      :::OF THE TRANSMITTER SECTION.
2057      :::IT ALSO CHECKS DNA TIMING
2058      :::MODE:SYN!NT
2059      :::LENGTH:SIX PLUS PARITY

```



```

2060                                     :::PARITY:EVEPAR
2061                                     :::CHARACTER:25
2062                                     :::
2063 007076 012767 000015 172022 TST13: MOV #13,TSTNO ;SAVE THIS
2064 007104 012767 007320 172004 MOV #TST14,NEXT ;GO TO THIS TEST WHEN THRU
2065 007112 052777 000400 011604 BIS #MRESET,@TXCSR ;MASTER RESET
2066 007120 012777 030000 011572 MOV #SYNINT,@PARCSR ;SET THE MODE
2067 007126 052777 000400 011570 BIS #MRESET,@TXCSR ;MASTER RESET
2068
2069 ;SET MAINTENANCE MODE & SEND
2070 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2071 007134 012777 004020 011562 MOV #MINT!SEND,@TXCSR
2072
2073 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2074 007142 012777 033426 011550 MOV #SYNINT!SIX!EVEPAR!26,@PARCSR
2075 007150 016703 011550 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2076 007154 112777 000025 011546 MOV #25,@TXDBUF ;LOAD DATA CHAR
2077 007162 012767 000125 171754 MOV #125,TEMP1 ;TO BE SHIFTED CHAR
2078 007170 012767 000007 171742 MOV #7,SHIFT ;# OF SHIFTS
2079 ;POKE CLK TO GET INTO SYNCHRONIZATION
2080 007176 052777 020000 011520 BIS #CLK,@TXCSR ;POKE CLK UP
2081 007204 042777 020000 011512 BIC #CLK,@TXCSR ;POKE CLK DOWN
2082 007212 005000 1$: CLR R0
2083 007214 006067 171724 ROR TEMP1 ;FORCE CARRY
2084 007220 103002 2$: BCC 2$ ;BR IF CARRY CLR
2085 007222 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2086 007226 2$:
2087 007226 052777 020000 011470 BIS #CLK,@TXCSR ;POKE CLK UP
2088 007234 042777 020000 011462 BIC #CLK,@TXCSR ;POKE CLK DOWN
2089 007242 017701 011456 MOV @TXCSR,R1 ;ACTUAL
2090 007246 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2091 007252 020001 R0,R1 ;COMPARE EXP VS ACT
2092 007254 001401 BEQ 3$
2093 007256 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2094 ;BIT,...ALSO CHECK DNA
2095 007260 3$:
2096 007260 005367 171654 DEC SHIFT ;# OF SHIFTS
2097 007264 001352 BNE 1$ ;DO IT AGAIN ?
2098 ;NOW POKE CLK TO SEE DNA
2099 007266 052777 020000 011430 BIS #CLK,@TXCSR ;POKE CLK
2100 007274 012700 100000 MOV #100000,R0 ;EXPECTED
2101 007300 017701 011420 MOV @TXCSR,R1 ;ACTUAL
2102 007304 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2103 007310 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2104 007312 001401 BEQ 4$
2105 007314 104003 HLT 3 ;DNA SHOULD BE SET
2106 ;IF DNA DID NOT SET
2107 ;CHECK WORD LENGTH SELECT LOGIC
2108 007316 4$:
2109 007316 104400
2110 SCOPE
2111 :::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2112 :::OF THE TRANSMITTER SECTION.
2113 :::IT ALSO CHECKS DNA TIMING
2114 :::MODE:SYNINT
2115 :::LENGTH:SIX PLUS PARITY
:::PARITY:ODDPAR

```

```

2116                                     :: CHARACTER:25
2117                                     ::
2118 007320 012767 000016 171600 TST14: MOV #14,TSTNO ;SAVE THIS
2119 007326 012767 007542 171562 MOV #TST15,NEXT ;GO TO THIS TEST WHEN THRU
2120 007334 052777 000400 011362 BIS #MRESET,@TXCSR ;MASTER RESET
2121 007342 012777 030000 011350 MOV #SYNINT,@PARCSR ;SET THE MODE
2122 007350 052777 000400 011346 BIS #MRESET,@TXCSR ;MASTER RESET
2123
2124 ;SET MAINTENANCE MODE & SEND
2125 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2126 007356 012777 004020 011340 MOV #MINT!SEND,@TXCSR
2127
2128 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2129 007364 012777 033026 011326 MOV #SYNINT!SIX!ODDPAR!26,@PARCSR
2130 007372 016703 011326 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2131 007376 112777 000025 011324 MOV #25,@TXDBUF ;LOAD DATA CHAR
2132 007404 012767 000025 171532 MOV #25,TEMP1 ;TO BE SHIFTED CHAR
2133 007412 012767 000007 171520 MOV #7,SHIFT ;# OF SHIFTS
2134
2135 ;POKE CLK TO GET INTO SYNCHRONIZATION
2136 007420 052777 020000 011276 BIS #CLK,@TXCSR ;POKE CLK UP
2137 007426 042777 020000 011270 BIC #CLK,@TXCSR ;POKE CLK DOWN
2138 007434 005000 171502 1$: CLR R0
2139 007436 006067 171502 ROR TEMP1 ;FORCE CARRY
2140 007442 103002 002000 BCC 2$ ;BR IF CARRY CLR
2141 007450 052777 020000 011246 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2142 007456 042777 020000 011240 BIC #CLK,@TXCSR ;POKE CLK DOWN
2143 007464 017701 011234 MOV @TXCSR,R1 ;ACTUAL
2144 007470 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2145 007474 020001 075777 CMP R0,R1 ;COMPARE EXP VS ACT
2146 007476 001401 075777 BEQ 3$
2147 007500 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2148 ;BIT,...ALSO CHECK DNA
2149
2150 007502 005367 171432 3$: DEC SHIFT ;# OF SHIFTS
2151 007502 001352 171432 BNE 1$ ;DO IT AGAIN ?
2152 007506 001352 171432
2153 ;NOW POKE CLK TO SEE DNA
2154 007510 052777 020000 011206 BIS #CLK,@TXCSR ;POKE CLK
2155 007516 012700 100000 MOV #100000,R0 ;EXPECTED
2156 007522 017701 011176 MOV @TXCSR,R1 ;ACTUAL
2157 007526 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2158 007532 020001 077777 CMP R0,R1 ;COMPARE EXP VS ACT
2159 007534 001401 077777 BEQ 4$
2160 007536 104003 HLT 3 ;DNA SHOULD BE SET
2161 ;IF DNA DID NOT SET
2162 ;CHECK WORD LENGTH SELECT LOGIC
2163 007540 104400 4$:
2164 007540 104400
2165 SCOPE
2166 ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2167 ::OF THE TRANSMITTER SECTION.
2168 ::IT ALSO CHECKS DNA TIMING
2169 ::MODE:ISYMOD
2170 ::LENGTH:SIX PLUS PARITY
2171 ::PARITY:EVEPAR
2172 ::CHARACTER:25

```



```

2172
2173 007542 012767 000017 171356 TST15: MOV #15,TSTNO ;SAVE THIS
2174 007550 012767 007764 171340 MOV #TST16,NEXT ;GO TO THIS TEST WHEN THRU
2175 007556 052777 000400 011140 BIS #MRESET,@TXCSR ;MASTER RESET
2176 007564 012777 000000 011126 MOV #ISYMOD,@PARCSR ;SET THE MODE
2177 007572 052777 000400 011124 BIS #MRESET,@TXCSR ;MASTER RESET
2178
2179 ;SET MAINTENANCE MODE & SEND
2180 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2181 007600 012777 004020 011116 MOV #MINT!SEND,@TXCSR
2182
2183 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2184 007606 012777 003426 011104 MOV #ISYMOD!SIX!EVEPAR!26,@PARCSR
2185 007614 016703 011104 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2186 007620 112777 000025 011102 MOV #25,@TXDBUF ;LOAD DATA CHAR
2187 007626 012767 000652 171310 MOV #652,TEMP1 ;TO BE SHIFTED CHAR
2188 007634 012767 000011 171276 MOV #9,SHIFT ;# OF SHIFTS
2189
2190 ;POKE CLK TO GET INTO SYNCHRONIZATION
2191 007642 052777 020000 011054 BIS #CLK,@TXCSR ;POKE CLK UP
2192 007650 042777 020000 011046 BIC #CLK,@TXCSR ;POKE CLK DOWN
2193 007656 005000 171260 1$: CLR R0
2194 007664 103002 ROR TEMP1 ;FORCE CARRY
2195 007666 052700 002000 BCC 2$ ;BR IF CARRY CLR
2196 007672 2$: BIS #BITW,R0 ;EQUIV OF BITW
2197 007672 052777 020000 011024 BIS #CLK,@TXCSR ;POKE CLK UP
2198 007700 042777 020000 011016 BIC #CLK,@TXCSR ;POKE CLK DOWN
2199 007706 017701 011012 MOV @TXCSR,R1 ;ACTUAL
2200 007712 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2201 007716 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2202 007720 001401 BEQ 3$
2203 007722 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2204 ;BIT,...ALSO CHECK DNA
2205 007724 3$: DEC SHIFT ;# OF SHIFTS
2206 007724 005367 171210 BNE 1$ ;DO IT AGAIN ?
2207 007730 001352
2208 ;NOW POKE CLK TO SEE DNA
2209 007732 052777 020000 010764 BIS #CLK,@TXCSR ;POKE CLK
2210 007740 012700 000000 MOV #0,R0 ;EXPECTED
2211 007744 017701 010754 MOV @TXCSR,R1 ;ACTUAL
2212 007750 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2213 007754 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2214 007756 001401 BEQ 4$
2215 007760 104003 HLT 3 ;DNA SHOULD BE SET
2216 ;IF DNA DID NOT SET
2217 ;CHECK WORD LENGTH SELECT LOGIC
2218 007762 4$:
2219 007762 104400 SCOPE
2220 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2221 ;:OF THE TRANSMITTER SECTION.
2222 ;:IT ALSO CHECKS DNA TIMING
2223 ;:MODE:ISYMOD
2224 ;:LENGTH:SIX PLUS PARITY
2225 ;:PARITY:ODDPAR
2226 ;:CHARACTER:25
2227 ;:

```

```

2228 007764 012767 000020 171134 TST16: MOV #16,TSTNO ;SAVE THIS
2229 007772 012767 010206 171116 MOV #TST17,NEXT ;GO TO THIS TEST WHEN THRU
2230 010000 052777 000400 010716 BIS #MRESET,@TXCSR ;MASTER RESET
2231 010006 012777 000000 010704 MOV #ISYMOD,@PARCSR ;SET THE MODE
2232 010014 052777 000400 010702 BIS #MRESET,@TXCSR ;MASTER RESET
2233
2234 ;SET MAINTENANCE MODE & SEND
2235 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2236 010022 012777 004020 010674 MOV #MINT!SEND,@TXCSR
2237
2238 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2239 010030 012777 003026 010662 MOV #ISYMOD!SIX!ODDPAR!26,@PARCSR
2240 010036 016703 010662 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2241 010042 112777 000025 010660 MOV #25,@TXDBUF ;LOAD DATA CHAR
2242 010050 012767 000452 171066 MOV #452,TEMP1 ;TO BE SHIFTED CHAR
2243 010056 012767 000011 171054 MOV #9,SHIFT ;# OF SHIFTS
2244
2245 ;POKE CLK TO GET INTO SYNCHRONIZATION
2246 010064 052777 020000 010632 BIS #CLK,@TXCSR ;POKE CLK UP
2247 010072 042777 020000 010624 BIC #CLK,@TXCSR ;POKE CLK DOWN
2248 010100 005000 171036 1$: CLR R0
2249 010102 006067 171036 ROR TEMP1 ;FORCE CARRY
2250 010110 052700 002000 BCC 2$ ;BR IF CARRY CLR
2251 010114 052777 020000 010602 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2252 010122 042777 020000 010574 BIC #CLK,@TXCSR ;POKE CLK DOWN
2253 010130 017701 010570 MOV @TXCSR,R1 ;ACTUAL
2254 010134 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2255 010140 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2256 010142 001401 BEQ 3$
2257 010144 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2258 ;BIT,...ALSO CHECK DNA
2259
2260 010146 005367 170766 3$: DEC SHIFT ;# OF SHIFTS
2261 010146 001352 BNE 1$ ;DO IT AGAIN ?
2262 010152
2263 ;NOW POKE CLK TO SEE DNA
2264 010154 052777 020000 010542 BIS #CLK,@TXCSR ;POKE CLK
2265 010162 012700 000000 MOV #0,R0 ;EXPECTED
2266 010166 017701 010532 MOV @TXCSR,R1 ;ACTUAL
2267 010172 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2268 010176 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2269 010200 001401 BEQ 4$
2270 010202 104003 HLT 3 ;DNA SHOULD BE SET
2271 ;IF DNA DID NOT SET
2272 ;CHECK WORD LENGTH SELECT LOGIC
2273 010204 104400 4$:
2274
2275 SCOPE
2276
2277 ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2278 ::OF THE TRANSMITTER SECTION.
2279 ::IT ALSO CHECKS DNA TIMING
2280 ::MODE:SYNINT
2281 ::LENGTH:SEVEN PLUS PARITY
2282 ::PARITY:EVEPAR
2283 ::CHARACTER:125
  
```



```

2284 010206 012767 000021 170712 TST17: MOV #17,TSTNO ;SAVE THIS
2285 010214 012767 010430 170674 MOV #TST18,NEXT ;GO TO THIS TEST WHEN THRU
2286 010222 052777 000400 010474 BIS #MRESET,@TXCSR ;MASTER RESET
2287 010230 012777 030000 010462 MOV #SYNINT,@PARCSR ;SET THE MODE
2288 010236 052777 000400 010460 BIS #MRESET,@TXCSR ;MASTER RESET
2289
2290 ;SET MAINTENANCE MODE & SEND
2291 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2292 010244 012777 004020 010452 MOV #MINT!SEND,@TXCSR
2293
2294 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2295 010252 012777 035426 010440 MOV #SYNINT!SEVEN!EVEPAR!26,@PARCSR
2296 010260 016703 010440 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2297 010264 112777 000125 010436 MOV #125,@TXDBUF ;LOAD DATA CHAR
2298 010272 012767 000125 170644 MOV #125,TEMP1 ;TO BE SHIFTED CHAR
2299 010300 012767 000010 170632 MOV #8,SHIFT ;# OF SHIFTS
2300 ;POKE CLK TO GET INTO SYNCHRONIZATION
2301 010306 052777 020000 010410 BIS #CLK,@TXCSR ;POKE CLK UP
2302 010314 042777 020000 010402 BIC #CLK,@TXCSR ;POKE CLK DOWN
2303 010322 005000 1$: CLR R0
2304 010324 006067 170614 ROR TEMP1 ;FORCE CARRY
2305 010330 103002 BCC 2$ ;BR IF CARRY CLR
2306 010332 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2307 010336 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2308 010336 052777 020000 010360 BIC #CLK,@TXCSR ;POKE CLK DOWN
2309 010344 042777 020000 010352 MOV @TXCSR,R1 ;ACTUAL
2310 010352 017701 010346 BIC #075777,R1 ;SAVE BITW & DNA
2311 010356 042701 075777 CMP R0,R1 ;COMPARE EXP VS ACT
2312 010362 020001 BEQ 3$
2313 010364 001401 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2314 010366 104003 ;BIT,...ALSO CHECK DNA
2315
2316 010370 3$: DEC SHIFT ;# OF SHIFTS
2317 010370 005367 170544 BNE 1$ ;DO IT AGAIN ?
2318 010374 001352
2319 ;NOW POKE CLK TO SEE DNA
2320 010376 052777 020000 010320 BIS #CLK,@TXCSR ;POKE CLK
2321 010404 012700 100000 MOV #100000,R0 ;EXPECTED
2322 010410 017701 010310 MOV @TXCSR,R1 ;ACTUAL
2323 010414 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2324 010420 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2325 010422 001401 BEQ 4$
2326 010424 104003 HLT 3 ;DNA SHOULD BE SET
2327 ;IF DNA DID NOT SET
2328 ;CHECK WORD LENGTH SELECT LOGIC
2329 010426 4$:
2330 010426 104400 SCOPE
2331 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2332 ;:OF THE TRANSMITTER SECTION.
2333 ;:IT ALSO CHECKS DNA TIMING
2334 ;:MODE:SYNINT
2335 ;:LENGTH:SEVEN PLUS PARITY
2336 ;:PARITY:ODDPAR
2337 ;:CHARACTER:125
2338
2339 010430 012767 000022 170470 TST18: MOV #18,TSTNO ;SAVE THIS

```

```

2340 010436 012767 010652 170452      MOV      #TST19,NEXT      ;GO TO THIS TEST WHEN THRU
2341 010444 052777 000400 010252      BIS      #MRESET,@TXCSR ;MASTER RESET
2342 010452 012777 030000 010240      MOV      #SYNINT,@PARCSR ;SET THE MODE
2343 010460 052777 000400 010236      BIS      #MRESET,@TXCSR ;MASTER RESET
2344
2345      ;SET MAINTENANCE MODE & SEND
2346      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2347 010466 012777 004020 010230      MOV      #MINT!SEND,@TXCSR
2348
2349      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2350 010474 012777 035026 010216      MOV      #SYNINT!SEVEN!ODDPAR!26,@PARCSR
2351 010502 016703 010216      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
2352 010506 112777 000125 010214      MOV      #125,@TXDBUF   ;LOAD DATA CHAR
2353 010514 012767 000325 170422      MOV      #325,TEMP1     ;TO BE SHIFTED CHAR
2354 010522 012767 000010 170410      MOV      #8,,SHIFT      ;# OF SHIFTS
2355      ;POKE CLK TO GET INTO SYNCRONIZATION
2356 010530 052777 020000 010166      BIS      #CLK,@TXCSR     ;POKE CLK UP
2357 010536 042777 020000 010160      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
2358 010544 005000      170372      1$:      CLR      R0
2359 010546 006067      170372      ROR      TEMP1      ;FORCE CARRY
2360 010552 103002      BCC      2$      ;BR IF CARRY CLR
2361 010554 052700 002000      BIS      #BITW,R0      ;EQUIV OF BITW
2362 010560      2$:
2363 010560 052777 020000 010136      BIS      #CLK,@TXCSR     ;POKE CLK UP
2364 010566 042777 020000 010130      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
2365 010574 017701 010124      MOV      @TXCSR,R1      ;ACTUAL
2366 010600 042701 075777      BIC      #075777,R1      ;SAVE BITW & DNA
2367 010604 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2368 010606 001401      BEQ      3$
2369 010610 104003      HLT      3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2370      ;BIT,...ALSO CHECK DNA
2371 010612      3$:
2372 010612 005367 170322      DEC      SHIFT      ;# OF SHIFTS
2373 010616 001352      BNE      1$      ;DO IT AGAIN ?
2374      ;NOW POKE CLK TO SEE DNA
2375 010620 052777 020000 010076      BIS      #CLK,@TXCSR     ;POKE CLK
2376 010626 012700 100000      MOV      #100000,R0     ;EXPECTED
2377 010632 017701 010066      MOV      @TXCSR,R1      ;ACTUAL
2378 010636 042701 077777      BIC      #777777,R1      ;SAVE DNA ONLY
2379 010642 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2380 010644 001401      BEQ      4$
2381 010646 104003      HLT      3      ;DNA SHOULD BE SET
2382      ;IF DNA DID NOT SET
2383      ;CHECK WORD LENGTH SELECT LOGIC
2384 010650      4$:
2385 010650 104400      SCOPE
2386      ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2387      ;;OF THE TRANSMITTER SECTION.
2388      ;;IT ALSO CHECKS DNA TIMING
2389      ;;MODE:ISYMOD
2390      ;;LENGTH:SEVEN PLUS PARITY
2391      ;;PARITY:EVEPAR
2392      ;;CHARACTER:125
2393      ;;
2394 010652 012767 000023 170246      TST19: MOV      #19,TSTNO      ;SAVE THIS
2395 010660 012767 011074 170230      MOV      #TST20,NEXT    ;GO TO THIS TEST WHEN THRU

```



```

2396 010666 052777 000400 010030      BIS      #MRESET,@TXCSR ;MASTER RESET
2397 010674 012777 000000 010016      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2398 010702 052777 000400 010014      BIS      #MRESET,@TXCSR ;MASTER RESET
2399
2400      ;SET MAINTENANCE MODE & SEND
2401      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2402 010710 012777 004020 010006      MOV      #MINT!SEND,@TXCSR
2403
2404      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2405 010716 012777 005426 007774      MOV      #ISYMOD!SEVEN!EVEPAR!26,@PARCSR
2406 010724 016703 007774      MOV      TXCSR,R3 ;SET UP FOR ERROR MSG
2407 010730 112777 000125 007772      MOV      #125,@TXDBUF ;LOAD DATA CHAR
2408 010736 012767 001252 170200      MOV      #1252,TEMP1 ;TO BE SHIFTED CHAR
2409 010744 012767 000012 170166      MOV      #10,SHIFT ;# OF SHIFTS
2410
2411      ;POKE CLK TO GET INTO SYNCHRONIZATION
2412 010752 052777 020000 007744      BIS      #CLK,@TXCSR ;POKE CLK UP
2413 010760 042777 020000 007736      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2414 010770 006067 170150      1$:      CLR      R0
2415 010774 103002      ROR      TEMP1 ;FORCE CARRY
2416 010776 052700 002000      BCC      2$ ;BR IF CARRY CLR
2417 011002      BIS      #BITW,R0 ;EQUIV OF BITW
2418 011002 052777 020000 007714      BIS      #CLK,@TXCSR ;POKE CLK UP
2419 011010 042777 020000 007706      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2420 011016 017701 007702      MOV      @TXCSR,R1 ;ACTUAL
2421 011022 042701 075777      BIC      #075777,R1 ;SAVE BITW & DNA
2422 011026 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
2423 011030 001401      BEQ      3$
2424 011032 104003      HLT      3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2425      ;BIT,...ALSO CHECK DNA
2426 011034      3$:
2427 011034 005367 170100      DEC      SHIFT ;# OF SHIFTS
2428 011040 001352      BNE      1$ ;DO IT AGAIN ?
2429
2430      ;NOW POKE CLK TO SEE DNA
2431 011042 052777 020000 007654      BIS      #CLK,@TXCSR ;POKE CLK
2432 011050 012700 000000      MOV      #0,R0 ;EXPECTED
2433 011054 017701 007644      MOV      @TXCSR,R1 ;ACTUAL
2434 011060 042701 077777      BIC      #77777,R1 ;SAVE DNA ONLY
2435 011064 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
2436 011066 001401      BEQ      4$
2437 011070 104003      HLT      3 ;DNA SHOULD BE SET
2438      ;IF DNA DID NOT SET
2439      ;CHECK WORD LENGTH SELECT LOGIC
2440      4$:
2441      SCOPE
2442      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2443      ::OF THE TRANSMITTER SECTION.
2444      ::IT ALSO CHECKS DNA TIMING
2445      ::MODE:ISYMOD
2446      ::LENGTH:SEVEN PLUS PARITY
2447      ::PARITY:ODDPAR
2448      ::CHARACTER:125
2449 011074 012767 000024 170024      TST20: MOV      #20,TSTNO ;SAVE THIS
2450 011102 012767 011316 170006      MOV      #TST21,NEXT ;GO TO THIS TEST WHEN THRU
2451 011110 052777 000400 007606      BIS      #MRESET,@TXCSR ;MASTER RESET

```

```

2452 011116 012777 000000 007574      MOV    #ISYMOD, @PARCSR ;SET THE MODE
2453 011124 052777 000400 007572      BIS    #MRESET, @TXCSR ;MASTER RESET
2454
2455      ;SET MAINTENANCE MODE & SEND
2456      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2457 011132 012777 004020 007564      MOV    #MINT!SEND, @TXCSR
2458
2459      ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2460 011140 012777 005026 007552      MOV    #ISYMOD!SEVEN!ODDPAR!26, @PARCSR
2461 011146 016703 007552      MOV    TXCSR, R3      ;SET UP FOR ERROR MSG
2462 011152 112777 000125 007550      MOV    #125, @TXDBUF ;LOAD DATA CHAR
2463 011160 012767 001652 167756      MOV    #1652, TEMP1  ;TO BE SHIFTED CHAR
2464 011166 012767 000012 167744      MOV    #10, SHIFT   ;# OF SHIFTS
2465      ;POKE CLK TO GET INTO SYNCHRONIZATION
2466 011174 052777 020000 007522      BIS    #CLK, @TXCSR  ;POKE CLK UP
2467 011202 042777 020000 007514      BIC    #CLK, @TXCSR  ;POKE CLK DOWN
2468 011210 005000      1$:      CLR    R0
2469 011212 006067 167726      ROR    TEMP1      ;FORCE CARRY
2470 011216 103002      BCC    2$      ;BR IF CARRY CLR
2471 011220 052700 002000      BIS    #BITW, R0     ;EQUIV OF BITW
2472      2$:
2473 011224 052777 020000 007472      BIS    #CLK, @TXCSR  ;POKE CLK UP
2474 011232 042777 020000 007464      BIC    #CLK, @TXCSR  ;POKE CLK DOWN
2475 011240 017701 007460      MOV    @TXCSR, R1    ;ACTUAL
2476 011244 042701 075777      BIC    #075777, R1    ;SAVE BITW & DNA
2477 011250 020001      CMP    R0, R1      ;COMPARE EXP VS ACT
2478 011252 001401      BEQ    3$
2479 011254 104003      HLT    3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2480      ;BIT,....ALSO CHECK DNA
2481 011256      3$:
2482 011256 005367 167656      DEC    SHIFT      ;# OF SHIFTS
2483 011262 001352      BNE    1$      ;DO IT AGAIN ?
2484      ;NOW POKE CLK TO SEE DNA
2485 011264 052777 020000 007432      BIS    #CLK, @TXCSR  ;POKE CLK
2486 011272 012700 000000      MOV    #0, R0      ;EXPECTED
2487 011276 017701 007422      MOV    @TXCSR, R1    ;ACTUAL
2488 011302 042701 077777      BIC    #77777, R1    ;SAVE DNA ONLY
2489 011306 020001      CMP    R0, R1      ;COMPARE EXP VS ACT
2490 011310 001401      BEQ    4$
2491 011312 104003      HLT    3      ;DNA SHOULD BE SET
2492      ;IF DNA DID NOT SET
2493      ;CHECK WORD LENGTH SELECT LOGIC
2494 011314      4$:
2495 011314 104400      SCOPE
2496
2497      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2498      ::OF THE TRANSMITTER SECTION.
2499      ::IT ALSO CHECKS DNA TIMING
2500      ::MODE:SYNINT
2501      ::LENGTH:EIGHT PLUS PARITY
2502      ::PARITY:EVEPAR
2503      ::CHARACTER:125
2504
2505 011316 012767 000025 167602      TST21: MOV    #21, TSTNO      ;SAVE THIS
2506 011324 012767 011540 167564      MOV    #TST22, NEXT    ;GO TO THIS TEST WHEN THRU
2507 011332 052777 000400 007364      BIS    #MRESET, @TXCSR ;MASTER RESET

```



```

2508 011340 012777 030000 007352      MOV    #SYNINT,@PARCSR ;SET THE MODE
2509 011346 052777 000400 007350      BIS    #MRESET,@TXCSR ;MASTER RESET
2510
2511      ;SET MAINTENANCE MODE & SEND
2512      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2513 011354 012777 004020 007342      MOV    #MINT!SEND,@TXCSR
2514
2515      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2516 011362 012777 037426 007330      MOV    #SYNINT!EIGHT!EVEPAR!26,@PARCSR
2517 011370 016703 007330      MOV    TXCSR,R3      ;SET UP FOR ERROR MSG
2518 011374 112777 000125 007326      MOVB   #125,@TXDBUF   ;LOAD DATA CHAR
2519 011402 012767 000125 167534      MOV    #125,TEMP1    ;TO BE SHIFTED CHAR
2520 011410 012767 000011 167522      MOV    #9,SHIFT      ;# OF SHIFTS
2521      ;POKE CLK TO GET INTO SYNCRONIZATION
2522 011416 052777 020000 007300      BIS    #CLK,@TXCSR   ;POKE CLK UP
2523 011424 042777 020000 007272      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2524 011432 005000      1$:      CLR    R0
2525 011434 006067 167504      ROR    TEMP1        ;FORCE CARRY
2526 011440 103002      BCC    2$      ;BR IF CARRY CLR
2527 011442 052700 002000      BIS    #BITW,R0      ;EQUIV OF BITW
2528 011446      2$:
2529 011446 052777 020000 007250      BIS    #CLK,@TXCSR   ;POKE CLK UP
2530 011454 042777 020000 007242      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2531 011462 017701 007236      MOV    @TXCSR,R1     ;ACTUAL
2532 011466 042701 075777      BIC    #075777,R1    ;SAVE BITW & DNA
2533 011472 020001      CMP    R0,R1      ;COMPARE EXP VS ACT
2534 011474 001401      BEQ    3$
2535 011476 104003      HLT    3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2536      ;BIT,...ALSO CHECK DNA
2537 011500      3$:
2538 011500 005367 167434      DEC    SHIFT        ;# OF SHIFTS
2539 011504 001352      BNE    1$      ;DO IT AGAIN ?
2540      ;NOW POKE CLK TO SEE DNA
2541 011506 052777 020000 007210      BIS    #CLK,@TXCSR   ;POKE CLK
2542 011514 012700 100000      MOV    #100000,R0    ;EXPECTED
2543 011520 017701 007200      MOV    @TXCSR,R1     ;ACTUAL
2544 011524 042701 077777      BIC    #77777,R1     ;SAVE DNA ONLY
2545 011530 020001      CMP    R0,R1      ;COMPARE EXP VS ACT
2546 011532 001401      BEQ    4$
2547 011534 104003      HLT    3      ;DNA SHOULD BE SET
2548      ;IF DNA DID NOT SET
2549      ;CHECK WORD LENGTH SELECT LOGIC
2550      4$:
2551 011536 104400      SCOPE
2552      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2553      ::OF THE TRANSMITTER SECTION.
2554      ::IT ALSO CHECKS DNA TIMING
2555      ::MODE:SYNINT
2556      ::LENGTH:EIGHT PLUS PARITY
2557      ::PARITY:ODDPAR
2558      ::CHARACTER:125
2559      ::
2560 011540 012767 000026 167360      TST22: MOV    #22,TSTNO      ;SAVE THIS
2561 011546 012767 011762 167342      MOV    #TST23,NEXT   ;GO TO THIS TEST WHEN THRU
2562 011554 052777 000400 007142      BIS    #MRESET,@TXCSR ;MASTER RESET
2563 011562 012777 030000 007130      MOV    #SYNINT,@PARCSR ;SET THE MODE
  
```

```

2564 011570 052777 000400 007126      BIS      #MRESET,@TXCSR ;MASTER RESET
2565
2566      ;SET MAINTENANCE MODE & SEND
2567      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2568 011576 012777 004020 007120      MOV      #MINT!SEND,@TXCSR
2569
2570      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2571 011604 012777 037026 007106      MOV      #SYNINT!EIGHT!ODDPAR!26,@PARCSR
2572 011612 016703 007106      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
2573 011616 112777 000125 007104      MOV      #125,@TXDBUF ;LOAD DATA CHAR
2574 011624 012767 000525 167312      MOV      #525,TEMP1 ;TO BE SHIFTED CHAR ;
2575 011632 012767 000011 167300      MOV      #9,SHIFT ;# OF SHIFTS
2576
2577      ;POKE CLK TO GET INTO SYNCRONIZATION
2578 011640 052777 020000 007056      BIS      #CLK,@TXCSR ;POKE CLK UP
2579 011646 042777 020000 007050      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2580 011654 005000      1$:      CLR      R0
2581 011656 006067 167262      ROR      TEMP1 ;FORCE CARRY
2582 011662 103002      BCC      2$ ;BR IF CARRY CLR
2583 011664 052700 002000      BIS      #BITW,R0 ;EQUIV OF BITW
2584 011670
2585 011670 052777 020000 007026      BIS      #CLK,@TXCSR ;POKE CLK UP
2586 011676 042777 020000 007020      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2587 011704 017701 007014      MOV      @TXCSR,R1 ;ACTUAL
2588 011710 042701 075777      BIC      #075777,R1 ;SAVE BITW & DNA
2589 011714 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
2590 011716 001401      BEQ      3$
2591 011720 104003      HLT      3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2592      ;BIT,...ALSO CHECK DNA
2593 011722
2594 011722 005367 167212      3$:      DEC      SHIFT ;# OF SHIFTS
2595 011726 001352      BNE      1$ ;DO IT AGAIN ?
2596      ;NOW POKE CLK TO SEE DNA
2597 011730 052777 020000 006766      BIS      #CLK,@TXCSR ;POKE CLK
2598 011736 012700 100000      MOV      #100000,R0 ;EXPECTED
2599 011742 017701 006756      MOV      @TXCSR,R1 ;ACTUAL
2600 011746 042701 077777      BIC      #77777,R1 ;SAVE DNA ONLY
2601 011752 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
2602 011754 001401      BEQ      4$
2603 011756 104003      HLT      3 ;DNA SHOULD BE SET
2604      ;IF DNA DID NOT SET
2605      ;CHECK WORD LENGTH SELECT LOGIC
2606 011760
2607 011760 104400      4$:
2608      SCOPE
2609      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2610      ::OF THE TRANSMITTER SECTION.
2611      ::IT ALSO CHECKS DNA TIMING
2612      ::MODE:ISYMOD
2613      ::LENGTH:EIGHT PLUS PARITY
2614      ::PARITY:EVEPAR
2615      ::CHARACTER:125
2616 011762 012767 000027 167136      TST23: MOV      #23,TSTNO ;SAVE THIS
2617 011770 012767 012204 167120      MOV      #TST24,NEXT ;GO TO THIS TEST WHEN THRU
2618 011776 052777 000400 006720      BIS      #MRESET,@TXCSR ;MASTER RESET
2619 012004 012777 000000 006706      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2619 012012 052777 000400 006704      BIS      #MRESET,@TXCSR ;MASTER RESET
  
```



```

2620
2621 ;SET MAINTENANCE MODE & SEND
2622 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2623 012020 012777 004020 006676 MOV #MINT!SEND,@TXCSR
2624
2625 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2626 012026 012777 007426 006664 MOV #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
2627 012034 016703 006664 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2628 012040 112777 000125 006662 MOV #125,@TXDBUF ;LOAD DATA CHAR
2629 012046 012767 002252 167070 MOV #2252,TEMP1 ;TO BE SHIFTED CHAR
2630 012054 012767 000013 167056 MOV #11,SHIFT ;# OF SHIFTS
2631
2632 ;POKE CLK TO GET INTO SYNCRONIZATION
2633 012062 052777 020000 006634 BIS #CLK,@TXCSR ;POKE CLK UP
2634 012070 042777 020000 006626 BIC #CLK,@TXCSR ;POKE CLK DOWN
2635 012076 005000 167040 1$: CLR R0
2636 012100 006067 167040 ROR TEMP1 ;FORCE CARRY
2637 012106 052700 002000 BCC 2$ ;BR IF CARRY CLR
2638 012112 2$: BIS #BITW,R0 ;EQUIV OF BITW
2639 012112 052777 020000 006604 BIS #CLK,@TXCSR ;POKE CLK UP
2640 012120 042777 020000 006576 BIC #CLK,@TXCSR ;POKE CLK DOWN
2641 012126 017701 006572 MOV @TXCSR,R1 ;ACTUAL
2642 012132 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2643 012136 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2644 012140 001401 BEQ 3$
2645 012142 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2646 ;BIT,....ALSO CHECK DNA
2647 012144 3$: DEC SHIFT ;# OF SHIFTS
2648 012144 005367 166770 BNE 1$ ;DO IT AGAIN ?
2649 012150 001352
2650 ;NOW POKE CLK TO SEE DNA
2651 012152 052777 020000 006544 BIS #CLK,@TXCSR ;POKE CLK
2652 012160 012700 000000 MOV #0,R0 ;EXPECTED
2653 012164 017701 006534 MOV @TXCSR,R1 ;ACTUAL
2654 012170 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2655 012174 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2656 012176 001401 BEQ 4$
2657 012200 104003 HLT 3 ;DNA SHOULD BE SET
2658 ;IF DNA DID NOT SET
2659 ;CHECK WORD LENGTH SELECT LOGIC
2660 012202 4$:
2661 012202 104400
2662 SCOPE
2663 ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2664 ::OF THE TRANSMITTER SECTION.
2665 ::IT ALSO CHECKS DNA TIMING
2666 ::MODE:ISYMOD
2667 ::LENGTH:EIGHT PLUS PARITY
2668 ::PARITY:ODDPAR
2669 ::CHARACTER:125
2670 012204 012767 000030 166714 TST24: MOV #24,TSTNO ;SAVE THIS
2671 012212 012767 012426 166676 MOV #TST25,NEXT ;GO TO THIS TEST WHEN THRU
2672 012220 052777 000400 006476 BIS #MRESET,@TXCSR ;MASTER RESET
2673 012226 012777 000000 006464 MOV #ISYMOD,@PARCSR ;SET THE MODE
2674 012234 052777 000400 006462 BIS #MRESET,@TXCSR ;MASTER RESET
2675

```

```

2676 ;SET MAINTENANCE MODE & SEND
2677 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2678 012242 012777 004020 006454 MOV #MINT!SEND,@TXCSR
2679
2680 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2681 012250 012777 007026 006442 MOV #ISYMOD!EIGHT!ODDPAR!26,@PARCSR
2682 012256 016703 006442 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2683 012262 112777 000125 006440 MOVB #125,@TXDBUF ;LOAD DATA CHAR
2684 012270 012767 003252 166646 MOV #3252,TEMP1 ;TO BE SHIFTED CHAR
2685 012276 012767 000013 166634 MOV #11,SHIFT ;# OF SHIFTS
2686 ;POKE CLK TO GET INTO SYNCRONIZATION
2687 012304 052777 020000 006412 BIS #CLK,@TXCSR ;POKE CLK UP
2688 012312 042777 020000 006404 BIC #CLK,@TXCSR ;POKE CLK DOWN
2689 012320 005000 1$: CLR R0
2690 012322 006067 166616 ROR TEMP1 ;FORCE CARRY
2691 012326 103002 BCC 2$ ;BR IF CARRY CLR
2692 012330 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2693 012334 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2694 012334 052777 020000 006362 BIC #CLK,@TXCSR ;POKE CLK DOWN
2695 012342 042777 020000 006354 MOV @TXCSR,R1 ;ACTUAL
2696 012350 017701 006350 BIC #075777,R1 ;SAVE BITW & DNA
2697 012354 042701 075777 CMP R0,R1 ;COMPARE EXP VS ACT
2698 012360 020001 BEQ 3$
2699 012362 001401 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2700 012364 104003 ;BIT,...ALSO CHECK DNA
2701
2702 012366 3$: DEC SHIFT ;# OF SHIFTS
2703 012366 005367 166546 BNE 1$ ;DO IT AGAIN ?
2704 012372 001352
2705 ;NOW POKE CLK TO SEE DNA
2706 012374 052777 020000 006322 BIS #CLK,@TXCSR ;POKE CLK
2707 012402 012700 000000 MOV #0,R0 ;EXPECTED
2708 012406 017701 006312 MOV @TXCSR,R1 ;ACTUAL
2709 012412 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2710 012416 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2711 012420 001401 BEQ 4$
2712 012422 104003 HLT 3 ;DNA SHOULD BE SET
2713 ;IF DNA DID NOT SET
2714 ;CHECK WORD LENGTH SELECT LOGIC
2715 012424 4$:
2716 012424 104400 SCOPE
2717
2718 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2719 ;:OF THE TRANSMITTER SECTION.
2720 ;:IT ALSO CHECKS DNA TIMING
2721 ;:MODE:SYNINT
2722 ;:LENGTH:EIGHT PLUS PARITY
2723 ;:PARITY:EVEPAR
2724 ;:CHARACTER:252
2725
2726 012426 012767 000031 166472 TST25: MOV #25,TSTNO ;SAVE THIS
2727 012434 012767 012650 166454 MOV #TST26,NEXT ;GO TO THIS TEST WHEN THRU
2728 012442 052777 000400 006254 BIS #MRESET,@TXCSR ;MASTER RESET
2729 012450 012777 030000 006242 MOV #SYNINT,@PARCSR ;SET THE MODE
2730 012456 052777 000400 006240 BIS #MRESET,@TXCSR ;MASTER RESET
2731

```



```

2732 ;SET MAINTENANCE MODE & SEND
2733 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2734 012464 012777 004020 006232 MOV #MINT!SEND,@TXCSR
2735
2736 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2737 012472 012777 037426 006220 MOV #SYNINT!EIGHT!EVEPAR!26,@PARCSR
2738 012500 016703 006220 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2739 012504 112777 000252 006216 MOV #252,@TXDBUF ;LOAD DATA CHAR
2740 012512 012767 000252 166424 MOV #252,TEMP1 ;TO BE SHIFTED CHAR
2741 012520 012767 000011 166412 MOV #9,SHIFT ;# OF SHIFTS
2742 ;POKE CLK TO GET INTO SYNCRONIZATION
2743 012526 052777 020000 006170 BIS #CLK,@TXCSR ;POKE CLK UP
2744 012534 042777 020000 006162 BIC #CLK,@TXCSR ;POKE CLK DOWN
2745 012542 005000 1$: CLR R0
2746 012544 006067 166374 ROR TEMP1 ;FORCE CARRY
2747 012550 103002 BCC 2$ ;BR IF CARRY CLR
2748 012552 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2749 012556 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2750 012556 052777 020000 006140 BIC #CLK,@TXCSR ;POKE CLK DOWN
2751 012564 042777 020000 006132 MOV @TXCSR,R1 ;ACTUAL
2752 012572 017701 006126 BIC #075777,R1 ;SAVE BITW & DNA
2753 012576 042701 075777 CMP R0,R1 ;COMPARE EXP VS ACT
2754 012602 020001 BEQ 3$
2755 012604 001401 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2756 012606 104003 ;BIT,...ALSO CHECK DNA
2757
2758 012610 3$: DEC SHIFT ;# OF SHIFTS
2759 012610 005367 166324 BNE 1$ ;DO IT AGAIN ?
2760 012614 001352
2761 ;NOW POKE CLK TO SEE DNA
2762 012616 052777 020000 006100 BIS #CLK,@TXCSR ;POKE CLK
2763 012624 012700 100000 MOV #100000,R0 ;EXPECTED
2764 012630 017701 006070 MOV @TXCSR,R1 ;ACTUAL
2765 012634 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2766 012640 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2767 012642 001401 BEQ 4$
2768 012644 104003 HLT 3 ;DNA SHOULD BE SET
2769 ;IF DNA DID NOT SET
2770 ;CHECK WORD LENGTH SELECT LOGIC
2771 012646 4$:
2772 012646 104400 SCOPE
2773 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2774 ;:OF THE TRANSMITTER SECTION.
2775 ;:IT ALSO CHECKS DNA TIMING
2776 ;:MODE:SYNINT
2777 ;:LENGTH:EIGHT PLUS PARITY
2778 ;:PARITY:ODDPAR
2779 ;:CHARACTER:252
2780 ;:
2781 012650 012767 000032 166250 TST26: MOV #26,TSTNO ;SAVE THIS
2782 012656 012767 013072 166232 MOV #TST27,NEXT ;GO TO THIS TEST WHEN THRU
2783 012664 052777 000400 006032 BIS #MRESET,@TXCSR ;MASTER RESET
2784 012672 012777 030000 006020 MOV #SYNINT,@PARCSR ;SET THE MODE
2785 012700 052777 000400 006016 BIS #MRESET,@TXCSR ;MASTER RESET
2786
2787 ;SET MAINTENANCE MODE & SEND

```

```

2788                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2789 012706 012777 004020 006010      MOV      #MINT!SEND,@TXCSR
2790
2791                                     ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2792 012714 012777 037026 005776      MOV      #SYNINT!EIGHT!ODDPAR!26,@PARCSR
2793 012722 016703 005776              MOV      TXCSR,R3          ;SET UP FOR ERROR MSG
2794 012726 112777 000252 005774      MOV      #252,@TXDBUF      ;LOAD DATA CHAR
2795 012734 012767 000652 166202      MOV      #652,TEMP1      ;TO BE SHIFTED CHAR
2796 012742 012767 000011 166170      MOV      #9,SHIFT        ;# OF SHIFTS
2797
2798 012750 052777 020000 005746      ;POKE CLK TO GET INTO SYNCHRONIZATION
2799 012756 042777 020000 005740      BIS      #CLK,@TXCSR      ;POKE CLK UP
2800 012764 005000                    BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2801 012766 006067 166152      1$:      CLR      R0
2802 012772 103002                    ROR      TEMP1      ;FORCE CARRY
2803 012774 052700 002000      BCC      2$      ;BR IF CARRY CLR
2804 013000                    BIS      #BITW,R0      ;EQUIV OF BITW
2805 013000 052777 020000 005716      2$:      BIS      #CLK,@TXCSR      ;POKE CLK UP
2806 013006 042777 020000 005710      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2807 013014 017701 005704      MOV      @TXCSR,R1      ;ACTUAL
2808 013020 042701 075777      BIC      #075777,R1      ;SAVE BITW & DNA
2809 013024 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2810 013026 001401      BEQ      3$
2811 013030 104003      HLT      3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2812                                     ;BIT,...ALSO CHECK DNA
2813 013032      3$:
2814 013032 005367 166102      DEC      SHIFT      ;# OF SHIFTS
2815 013036 001352      BNE      1$      ;DO IT AGAIN ?
2816
2817 013040 052777 020000 005656      ;NOW POKE CLK TO SEE DNA
2818 013046 012700 100000      BIS      #CLK,@TXCSR      ;POKE CLK
2819 013052 017701 005646      MOV      #100000,R0      ;EXPECTED
2820 013056 042701 077777      MOV      @TXCSR,R1      ;ACTUAL
2821 013062 020001      BIC      #77777,R1      ;SAVE DNA ONLY
2822 013064 001401      CMP      R0,R1      ;COMPARE EXP VS ACT
2823 013066 104003      BEQ      4$
2824      HLT      3      ;DNA SHOULD BE SET
2825                                     ;IF DNA DID NOT SET
2826                                     ;CHECK WORD LENGTH SELECT LOGIC
2827 013070 104400      4$:
2828      SCOPE
2829      ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2830      ::OF THE TRANSMITTER SECTION.
2831      ::IT ALSO CHECKS DNA TIMING
2832      ::MODE:SYNINT
2833      ::LENGTH:EIGHT PLUS PARITY
2834      ::PARITY:EVEPAR
2835      ::CHARACTER:0
2836 013072 012767 000033 166026      TST27: MOV      #27,TSTNO      ;SAVE THIS
2837 013100 012767 013314 166010      MOV      #TST28,NEXT      ;GO TO THIS TEST WHEN THRU
2838 013106 052777 000400 005610      BIS      #MRESET,@TXCSR      ;MASTER RESET
2839 013114 012777 030000 005576      MOV      #SYNINT,@PARCSR      ;SET THE MODE
2840 013122 052777 000400 005574      BJS      #MRESET,@TXCSR      ;MASTER RESET
2841
2842      ;SET MAINTENANCE MODE & SEND
2843      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)

```



```

2844 013130 012777 004020 005566      MOV      #MINT!SEND,@TXCSR
2845
2846      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2847 013136 012777 037426 005554      MOV      #SYNINT!EIGHT!EVEPAR!26,@PARCSR
2848 013144 016703 005554      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
2849 013150 112777 000000 005552      MOVB     #0,@TXDBUF      ;LOAD DATA CHAR
2850 013156 012767 000000 165760      MOV      #0,TEMP1      ;TO BE SHIFTED CHAR
2851 013164 012767 000011 165746      MOV      #9,SHIFT      ;# OF SHIFTS
2852      ;POKE CLK TO GET INTO SYNCRONIZATION
2853 013172 052777 020000 005524      BIS      #CLK,@TXCSR      ;POKE CLK UP
2854 013200 042777 020000 005516      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2855 013206 005000      1$:      CLR      R0
2856 013210 006067 165730      ROR      TEMP1      ;FORCE CARRY
2857 013214 103002      BCC      2$      ;BR IF CARRY CLR
2858 013216 052700 002000      BIS      #BITW,R0      ;EQUIV OF BITW
2859 013222      2$:
2860 013222 052777 020000 005474      BIS      #CLK,@TXCSR      ;POKE CLK UP
2861 013230 042777 020000 005466      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2862 013236 017701 005462      MOV      @TXCSR,R1      ;ACTUAL
2863 013242 042701 075777      BIC      #075777,R1      ;SAVE BITW & DNA
2864 013246 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2865 013250 001401      BEQ      3$
2866 013252 104003      HLT      3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2867      ;BIT,...ALSO CHECK DNA
2868 013254      3$:
2869 013254 005367 165660      DEC      SHIFT      ;# OF SHIFTS
2870 013260 001352      BNE      1$      ;DO IT AGAIN ?
2871      ;NOW POKE CLK TO SEE DNA
2872 013262 052777 020000 005434      BIS      #CLK,@TXCSR      ;POKE CLK
2873 013270 012700 100000      MOV      #100000,R0      ;EXPECTED
2874 013274 017701 005424      MOV      @TXCSR,R1      ;ACTUAL
2875 013300 042701 077777      BIC      #77777,R1      ;SAVE DNA ONLY
2876 013304 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
2877 013306 001401      BEQ      4$
2878 013310 104003      HLT      3      ;DNA SHOULD BE SET
2879      ;IF DNA DID NOT SET
2880      ;CHECK WORD LENGTH SELECT LOGIC
2881 013312      4$:
2882 013312 104400      SCOPE
2883      ;;THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2884      ;;OF THE TRANSMITTER SECTION.
2885      ;;IT ALSO CHECKS DNA TIMING
2886      ;;MODE:SYNINT
2887      ;;LENGTH:EIGHT PLUS PARITY
2888      ;;PARITY:ODDPAR
2889      ;;CHARACTER:0
2890      ;;
2891 013314 012767 000034 165604 TST28: MOV      #28,TSTNO      ;SAVE THIS
2892 013322 012767 013536 165566      MOV      #TST29,NEXT      ;GO TO THIS TEST WHEN THRU
2893 013330 052777 000400 005366      BIS      #MRESET,@TXCSR      ;MASTER RESET
2894 013336 012777 030000 005354      MOV      #SYNINT,@PARCSR      ;SET THE MODE
2895 013344 052777 000400 005352      BIS      #MRESET,@TXCSR      ;MASTER RESET
2896
2897      ;SET MAINTENANCE MODE & SEND
2898      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2899 013352 012777 004020 005344      MOV      #MINT!SEND,@TXCSR
  
```

```

2900
2901 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2902 013360 012777 037026 005332 MOV #SYNINT,EIGHT!ODDPAR!26,@PARCSR
2903 013366 016703 005332 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2904 013372 112777 000000 005330 MOVB #0,@TXDBUF ;LOAD DATA CHAR
2905 013400 012767 000400 165536 MOV #400,TEMP1 ;TO BE SHIFTED CHAR
2906 013406 012767 000011 165524 MOV #9,SHIFT ;# OF SHIFTS
2907 ;POKE CLK TO GET INTO SYNCRONIZATION
2908 013414 052777 020000 005302 BIS #CLK,@TXCSR ;POKE CLK UP
2909 013422 042777 020000 005274 BIC #CLK,@TXCSR ;POKE CLK DOWN
2910 013430 005000 1$: CLR R0
2911 013432 006067 165506 ROR TEMP1 ;FORCE CARRY
2912 013436 103002 BCC 2$ ;BR IF CARRY CLR
2913 013440 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2914 013444 2$:
2915 013444 052777 020000 005252 BIS #CLK,@TXCSR ;POKE CLK UP
2916 013452 042777 020000 005244 BIC #CLK,@TXCSR ;POKE CLK DOWN
2917 013460 017701 005240 MOV @TXCSR,R1 ;ACTUAL
2918 013464 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2919 013470 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2920 013472 001401 BEQ 3$
2921 013474 104003 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2922 ;BIT,...ALSO CHECK DNA
2923 013476 3$:
2924 013476 005367 165436 DEC SHIFT ;# OF SHIFTS
2925 013502 001352 BNE 1$ ;DO IT AGAIN ?
2926 ;NOW POKE CLK TO SEE DNA
2927 013504 052777 020000 005212 BIS #CLK,@TXCSR ;POKE CLK
2928 013512 012700 100000 MOV #100000,R0 ;EXPECTED
2929 013516 017701 005202 MOV @TXCSR,R1 ;ACTUAL
2930 013522 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2931 013526 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2932 013530 001401 BEQ 4$
2933 013532 104003 HLT 3 ;DNA SHOULD BE SET
2934 ;IF DNA DID NOT SET
2935 ;CHECK WORD LENGTH SELECT LOGIC
2936 013534 4$:
2937 013534 104400 SCOPE
2938 ::THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2939 ::OF THE TRANSMITTER SECTION.
2940 ::IT ALSO CHECKS DNA TIMING
2941 ::MODE:SYNINT
2942 ::LENGTH:EIGHT PLUS PARITY
2943 ::PARITY:EVEPAR
2944 ::CHARACTER:377
2945 ::
2946 013536 012767 000035 165362 TST29: MOV #29,TSTNO ;SAVE THIS
2947 013544 012767 013760 165344 MOV #TST30,NEXT ;GO TO THIS TEST WHEN THRU
2948 013552 052777 000400 005144 BIS #MRESET,@TXCSR ;MASTER RESET
2949 013560 012777 030000 005132 MOV #SYNINT,@PARCSR ;SET THE MODE
2950 013566 052777 000400 005130 BIS #MRESET,@TXCSR ;MASTER RESET
2951
2952 ;SET MAINTENANCE MODE & SEND
2953 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2954 013574 012777 004020 005122 MOV #!INT!SEND,@TXCSR
2955

```



```

2956 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2957 013602 012777 037426 005110 MOV #SYNINT!EIGHT!EVEPAR!26,@PARCSR
2958 013610 016703 005110 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2959 013614 112777 000377 005106 MOVB #377,@TXDBUF ;LOAD DATA CHAR
2960 013622 012767 000377 165314 MOV #377,TEMP1 ;TO BE SHIFTED CHAR
2961 013630 012767 000011 165302 MOV #9,SHIFT ;# OF SHIFTS
2962 ;POKE CLK TO GET INTO SYNCRONIZATION
2963 013636 052777 020000 005060 BIS #CLK,@TXCSR ;POKE CLK UP
2964 013644 042777 020000 005052 BIC #CLK,@TXCSR ;POKE CLK DOWN
2965 013652 005000 1$: CLR R0
2966 013654 006067 165264 ROR TEMP1 ;FORCE CARRY
2967 013660 103002 BCC 2$ ;BR IF CARRY CLR
2968 013662 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2969 013666 2$: BIS #CLK,@TXCSR ;POKE CLK UP
2970 013666 052777 020000 005030 BIC #CLK,@TXCSR ;POKE CLK DOWN
2971 013674 042777 020000 005022 MOV @TXCSR,R1 ;ACTUAL
2972 013702 017701 005016 BIC #075777,R1 ;SAVE BITW & DNA
2973 013706 042701 075777 CMP R0,R1 ;COMPARE EXP VS ACT
2974 013712 020001 BEQ 3$
2975 013714 001401 HLT 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2976 013716 104003 ;BIT,...ALSO CHECK DNA
2977 3$: DEC SHIFT ;# OF SHIFTS
2978 013720 BNE 1$ ;DO IT AGAIN ?
2979 013720 005367 165214 ;NOW POKE CLK TO SEE DNA
2980 013724 001352 BIS #CLK,@TXCSR ;POKE CLK
2981 013726 052777 020000 004770 MOV #100000,R0 ;EXPECTED
2982 013734 012700 100000 MOV @TXCSR,R1 ;ACTUAL
2983 013740 017701 004760 BIC #77777,R1 ;SAVE DNA ONLY
2984 013744 042701 077777 CMP R0,R1 ;COMPARE EXP VS ACT
2985 013750 020001 BEQ 4$
2986 013752 001401 HLT 3 ;DNA SHOULD BE SET
2987 013754 104003 ;IF DNA DID NOT SET
2988 ;CHECK WORD LENGTH SELECT LOGIC
2989 4$: SCOPE
2990 ;:THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2991 013756 ;:OF THE TRANSMITTER SECTION.
2992 013756 104400 ;:IT ALSO CHECKS DNA TIMING
2993 ;:MODE:SYNINT
2994 ;:LENGTH:EIGHT PLUS PARITY
2995 ;:PARITY:ODDPAR
2996 ;:CHARACTER:377
2997 ;:
3000 TST30: MOV #30,TSTNO ;SAVE THIS
3001 013760 012767 000036 165140 MOV #.EOP,NEXT ;GO TO THIS TEST WHEN THRU
3002 013766 012767 014202 165122 BIS #MRESET,@TXCSR ;MASTER RESET
3003 013774 052777 000400 004722 MOV #SYNINT,@PARCSR ;SET THE MODE
3004 014002 012777 030000 004710 BIS #MRESET,@TXCSR ;MASTER RESET
3005 014010 052777 000400 004706
3006 ;SET MAINTENANCE MODE & SEND
3007 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
3008 014016 012777 004020 004700 MOV #MINT!SEND,@TXCSR
3009 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
3010
3011

```

3012	014024	012777	037026	004666	MOV	#SYNINT!EIGHT!ODDPAR!26,@PARCSR	
3013	014032	016703	004666		MOV	TXCSR,R3	:SET UP FOR ERROR MSG
3014	014036	112777	000377	004664	MOVB	#377,@TXDBUF	:LOAD DATA CHAR
3015	014044	012767	000777	165072	MOV	#777,TEMP1	:TO BE SHIFTED CHAR
3016	014052	012767	000011	165060	MOV	#9,SHIFT	:# OF SHIFTS
3017							
3018	014060	052777	020000	004636			:POKE CLK TO GET INTO SYNCRONIZATION
3019	014066	042777	020000	004630	BIS	#CLK,@TXCSR	:POKE CLK UP
3020	014074	005000			BIC	#CLK,@TXCSR	:POKE CLK DOWN
3021	014076	006067	165042		1\$: CLR	R0	
3022	014102	103002			ROR	TEMP1	:FORCE CARRY
3023	014104	052700	002000		BCC	2\$:	:BR IF CARRY CLR
3024	014110				BIS	#BITW,R0	:EQUIV OF BITW
3025	014110	052777	020000	004606			
3026	014116	042777	020000	004600	BIS	#CLK,@TXCSR	:POKE CLK UP
3027	014124	017701	004574		BIC	#CLK,@TXCSR	:POKE CLK DOWN
3028	014130	042701	075777		MOV	@TXCSR,R1	:ACTUAL
3029	014134	020001			BIC	#075777,R1	:SAVE BITW & DNA
3030	014136	001401			CMP	R0,R1	:COMPARE EXP VS ACT
3031	014140	104003			BEQ	3\$:	
3032					HLT	3	:BIT WINDOW DID NOT MATCH ACTUAL DATA
3033	014142						:BIT....ALSO CHECK DNA
3034	014142	005367	164772		3\$: DEC	SHIFT	:# OF SHIFTS
3035	014146	001352			BNE	1\$:	:DO IT AGAIN ?
3036							
3037	014150	052777	020000	004546			:NOW POKE CLK TO SEE DNA
3038	014156	012700	100000		BIS	#CLK,@TXCSR	:POKE CLK
3039	014162	017701	004536		MOV	#100000,R0	:EXPECTED
3040	014166	042701	077777		MOV	@TXCSR,R1	:ACTUAL
3041	014172	020001			BIC	#77777,R1	:SAVE DNA ONLY
3042	014174	001401			CMP	R0,R1	:COMPARE EXP VS ACT
3043	014176	104003			BEQ	4\$:	
3044					HLT	3	:DNA SHOULD BE SET
3045							:IF DNA DID NOT SET
3046	014200						:CHECK WORD LENGTH SELECT LOGIC
3047	014200	104400			4\$: SCOPE		

3048						:END OF PASS	
3049						:TYPE NAME OF TEST	
3050						:UPDATE PASS COUNT	
3051						:CHECK FOR EXIT TO ACT-11	
3052						:RESTART TEST	
3053							
3054							
3055	014202	104402			.EOP:	TYPE	:TYPE NAME OF TEST
3056	014204	017344				MEPASS	
3057	014206	104410	014440			CONVRT	,OUTCRY
3058	014212	104402	017065			TYPE	,DEVICE
3059	014216	105767	164760			TSTB	MULTD
3060	014222	001511				BEQ	CCC
3061	014224	005767	164766			TST	ACTREG
3062	014230	001007				BNE	RUNIT
3063	014232	104402	017077			TYPE	,MCOW
3064	014236	016700	164754			MOV	ACTREG,R0
3065	014242	000000				HALT	
3066						;SELECT SWITCHES & HIT CONTINUE	(PUT SW00 =1)
3067	014244	000167	165150			JMP	.START
3068	014250	062767	000010	164726	RUNIT:	ADD	#10,BASEADD
3069	014256	062767	000010	164726	ZERO:	ADD	#10,BASEIV
3070	014264	000241				CLC	
3071	014266	006167	164726			ROL	ROTADD
3072	014272	103410				BCS	2\$
3073							:UP DATE ROTATING POINTER
3074	014274	036767	164720	164714		BIT	ROTADD,ACTREG
3075	014302	001762				BEQ	RUNIT
3076	014304	004767	000034			JSR	PC,REPLAY
3077	014310	000167	000174			JMP	RESTRT
3078	014314	012767	000001	164676	2\$:	MOV	#1,ROTADD
3079							:OK!,NOW SET UP ROTATING
3080	014322	016767	164660	164654		MOV	KEEPADD,BASEADD
3081	014330	016767	164660	164654		MOV	KEEPIV,BASEIV
3082	014336	004767	000002			JSR	PC,REPLAY
3083	014342	000441				BR	CCC
3084	014344	016767	164634	004040	REPLAY:	MOV	BASEADD,DUBASE
3085	014352	004767	003702			JSR	PC,DUADDR
3086	014356	016767	164630	004350		MOV	BASEIV,DURIV
3087	014364	062767	000002	164620		ADD	#2,BASEIV
3088	014372	016767	164614	004336		MOV	BASEIV,DURIS
3089	014400	062767	000002	164604		ADD	#2,BASEIV
3090	014406	016767	164600	004324		MOV	BASEIV,DUTIV
3091	014414	062767	000002	164570		ADD	#2,BASEIV
3092	014422	016767	164564	004312		MOV	BASEIV,DUTIS
3093	014430	016767	004300	164554		MOV	DURIV,BASEIV
3094	014436	000207				RTS	PC
3095							:CREATE DURIS
3096	014440	000001			OUTCRY: 1		
3097	014442	006	002			.BYTE	6.2
3098	014444	020710				RXCSR	
3099							
3100	014446				CCC:		
3101	014446	005067	164462			CLR	LSTERR
3102	014452	005067	164570			CLR	ERRFLG
3103	014456	005267	164446			INC	PASCNT

```

3104 014462 016777 164442 164412      MOV    PASCNT,@LIGHTS      ;DISPLAY PASS COUNT
3105 014470 013701 000042              MOV    @#42,R1          ;CHECK FOR ACT-11 OR DDP
3106 014474 001405                      BEQ    RESTRT          ;IF NOT, CONTINUE TESTING
3107 014476 000005                      RESET
3108 014500 004711                      LOGICAL: JSR    PC,(R1)
3109 014502 000240                      NOP
3110 014504 000240                      NOP
3111 014506 000240                      NOP
3112 014510 012767 000340 163260 RESTRT: MOV    #340,PS          ;PREVENT INTERRUPTS (PRIO: 7)
3113 014516 104413                      CKSWR          ;CHECK FOR ^G
3114 014520 012767 003446 164366      MOV    #TST1,RTRN
3115 014526 000167 166714              JMP     TST1
3116
3117                                ;SCOPE LOOP AND INTERATION HANDLER
3118
3119 014532                                .SCOPE:
3120                                ;***** START OF CODE FOR THE X OR TESTER *****
3121 014532 000424                      BR      4$
3122
3123 014534 013746 000004              MOV    @#4,-(SP)          ;IF RUNNING ON THE X OR TESTER CHANGE
3124 014540 012737 014560 000004      MOV    #1$,@#4          ;THIS INSTRUCTION TO A 'NOP'(NOP=240)
3125 014546 005737 177060              TST    @#177060        ;SAVE CONTENTS OF ERROR VECTOR
3126 014552 012637 000004              MOV    (SP)+,@#4        ;SET FOR TIME OUT
3127 014556 000404                      BR      2$              ;TIME OUT ON X OR ?
3128 014560 022626                      1$:  CMP    (SP)+,(SP)+      ;RESTORE ERROR VECTOR
3129 014562 012637 000004              MOV    (SP)+,@#4        ;GO TO NEXT TEST
3130 014566 000403                      BR      3$              ;CLEAR THE STACK AFTER A TIMEOUT
3131 014570 016767 164322 164316      2$:  MOV    NEXT,RTRN        ;RESTORE ERROR VECTOR
3132 014576 016716 164312      3$:  MOV    RTRN,(SP)      ;LOOP ON PRESENT TEST
3133 014602 000002                      RTI          ;SET UP NEXT TEST IN RTRN
3134 014604                      4$:  ;***** END OF CODE FOR THE X OR TESTER ***** ;SET UP STACK FOR RTI
3135 014604 104413                      CKSWR          ;CHECK FOR ^G
3136 014606 032777 040000 164264      BIT     #SW14,@SWR      ;LOOP ON CURRENT TEST ?
3137 014614 001407                      BEQ    1$
3138 014616 000432                      BR      3$
3139 014620 105777 164260              TSTB   @TKCSR          ;TEST TTY FLAG
3140 014624 100027                      BPL     3$
3141 014626 017700 164254              MOV    @TKDBR,R0        ;CLR DONE BIT
3142 014632 000412                      BR      2$              ;IF A TTY KEY IS STRUCK GO TO NEXT TST
3143 014634 032777 004000 164236      1$:  BIT     #SW11,@SWR      ;INHIBIT ITERATIONS ?
3144 014642 001006                      BNE     2$
3145 014644 005267 164254              INC     LPCNT
3146 014650 026767 164250 164244      CMP     LPCNT,ICOUNT    ;CHECK FOR ITERATION CNT FINISH
3147 014656 101412                      BLOS    3$
3148 014660 105067 164362      2$:  CLRB    ERRFLG
3149 014664 005067 164234              CLR     LPCNT
3150 014670 012767 000005 164224      MOV     #5,ICOUNT      ;SET UP ITERATION COUNT
3151 014676 016767 164214 164210      MOV     NEXT,RTRN      ;SET UP NEXT TEST IN RTRN
3152 014704 016716 164204      3$:  MOV     RTRN,(SP)    ;SET UP STACK FOR RTI
3153 014710 000002                      RTI
3154 014712 001407                      BRW:   1407          ;RESTORE 'BEQ 1$' INSTRUCTION
3155 014714 000432                      BRX:   432          ;RESTORE 'BR 3$' INSTRUCTION
3156
3157                                ;CHECK FOR FREEZE ON CURRENT DATA
3158
3159 014716 104413                                .SCOPE1: CKSWR          ;CHECK FOR ^G

```



```

3160 014720 032777 001000 164152      BIT      #SW09,@SWR
3161 014726 001402                BEQ      1$
3162 014730 016716 164164                MOV      LOCK,(SP)
3163 014734 000002                1$:      RTI
3164
3165                ;TELETYPE OUTPUT ROUTINE
3166
3167 014736 010546                .TYPE:    MOV      R5,-(SP)
3168 014740 017605 000002                MOV      @2(SP),R5
3169 014744 062766 000002 000002                ADD      #2,2(SP)
3170 014752 105715                1$:      TSTB      (R5)                ;LOOK FOR '0'
3171 014754 001406                BEQ      3$
3172 014756 105777 164126                2$:      TSTB      @TPCSR                ;TEST DONE BIT
3173 014762 100375                BPL      2$
3174 014764 112577 164122                MOV      (R5)+,@TPDBR                ;TYPE CHAR
3175 014770 000770                BR      1$                ;DO IT AGAIN UNTIL '0' IS SEEN
3176 014772 012605                3$:      MOV      (SP)+,R5
3177 014774 000002                RTI
3178
3179                ;ASCII STRING INPUT ROUTINE
3180
3181 014776 010346                .INSTR:  MOV      R3,-(SP)
3182 015000 010446                MOV      R4,-(SP)
3183 015002 017667 000004 000010                MOV      @4(SP),.MSG
3184 015010 062766 000002 000004                ADD      #2,4(SP)
3185 015016 104402                .INST1:  TYPE
3186 015020 000000                .MSG:    0
3187 015022 012704 020044                MOV      #INBUF,R4
3188 015026 012703 000007                MOV      #7,R3
3189 015032 105777 164046                1$:      TSTB      @TKCSR
3190 015036 100375                BPL      1$
3191 015040 117714 164042                MOV      @TKDBR,(R4)
3192 015044 142714 000200                BICB      #200,(R4)
3193 015050 121427 000025                CMPB      (R4),#25
3194 015054 001003                BNE      200$                ;IS IT <^U>
3195 015056 104402 017254                TYPE,MCRLF
3196 015062 000755                BR      .INST1
3197 015064 122427 000015                200$:    CMPB      (R4)+,#15
3198 015070 001423                BEQ      INSTR2
3199 015072 117777 164010 164012                MOV      @TKDBR,@TPDBR
3200 015100 105777 164004                2$:      TSTB      @TPCSR
3201 015104 100375                BPL      2$
3202 015106 005303                DEC      R3
3203 015110 001350                BNE      1$
3204 015112 000402                BR      .INSTG
3205 015114 010346                .INSTE:  MOV      R3,-(SP)
3206 015116 010446                MOV      R4,-(SP)
3207 015120 104402                .INSTG:  TYPE
3208 015122 017250                MQM
3209 015124 005737 016412                TST      @WRDSW
3210 015130 001402                BEQ      400$
3211 015132 104402 017254                TYPE,MCRLF
3212 015136 000727                400$:    BR      .INST1
3213 015140 012604                INSTR2:  MOV      (SP)+,R4
3214 015142 012603                MOV      (CP)+,R3
3215 015144 000002                RTI

```

```

3216
3217
3218 ;CONVERT ASCII STRING TO OCTAL
3219 015146 010546 .PARAM: MOV R5,-(SP)
3220 015150 010446 MOV R4,-(SP)
3221 015152 016605 000004 MOV 4(SP),R5
3222 015156 012567 000170 MOV (R5)+,LOLIM
3223 015162 012567 000166 MOV (R5)+,HILIM
3224 015166 012567 000164 MOV (R5)+,DEVADR
3225 015172 112567 000162 MOV (R5)+,LOBITS
3226 015176 112567 000157 MOV (R5)+,ADRCNT
3227 015202 010566 000004 MOV R5,4(SP)
3228 015206 005005 PARAM1: CLR R5
3229 015210 012704 020044 MOV #INBUF,R4
3230 015214 122714 000015 CMPB #15,(R4)
3231 015220 001420 BEQ PARERR
3232 015222 121427 000060 1$: CMPB (R4),#60
3233 015226 002415 BLT PARERR
3234 015230 121427 000067 CMPB (R4),#67
3235 015234 003012 BGT PARERR
3236 015236 142714 000060 BICB #60,(R4)
3237 015242 152405 BISB (R4)+,R5
3238 015244 122714 000015 CMPB #15,(R4)
3239 015250 001414 BEQ LIMITS
3240 015252 006305 ASL R5
3241 015254 006305 ASL R5
3242 015256 006305 ASL R5
3243 015260 000760 BR 1$
3244 015262 122714 000015 PARERR: CMPB #15,(R4) ;IS FIRST CHARACTER A <CR>
3245 015266 001003 BNE 120$
3246 015270 005737 016412 TST @#RDSW ;IS CKSWR ROUTINE BEING USED
3247 015274 001023 BNE PARTI
3248 015276 104404 120$: INSTER
3249 015300 000742 BR PARAM1
3250
3251 ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
3252
3253 015302 020567 000046 LIMITS: CMP R5,HILIM
3254 015306 101365 BHI PARERR
3255 015310 020567 000036 CMP R5,LOLIM
3256 015314 103762 BLO PARERR
3257 015316 136705 000036 BITB LOBITS,R5
3258 015322 001357 BNE PARERR
3259
3260 ;STORE NUMBER AT SPECIFIED ADDRESS
3261
3262 015324 016704 000026 1$: MOV DEVADR,R4
3263 015330 010524 MOV R5,(R4)+
3264 015332 062705 000002 ADD #2,R5
3265 015336 105367 000017 DECB ADRCNT
3266 015342 001372 BNE 1$
3267 015344 012604 PARTI: MOV (SP)+,R4
3268 015346 012605 MOV (SP)+,R5
3269 015350 000002 RTI
3270 015352 000000 LOLIM: 0
3271 015354 000000 HILIM: 0

```



```

3272 015356 000000      DEVADR: 0
3273 015360 000000      LOBITS: 0
3274      015361      ADRCNT=LOBITS+1
3275
3276      ;SAVE PC OF TEST THAT FAILED AND R0-R5
3277
3278 015362 016667 000004 163604 .SAV05: MOV 4(SP),SAVPC
3279
3280      ;SAVE R0-R5
3281
3282 015370 010567 163574      SV05:  MOV  R5,SAVR5
3283 015374 010467 163566      MOV  R4,SAVR4
3284 015400 010367 163560      MOV  R3,SAVR3
3285 015404 010267 163552      MOV  R2,SAVR2
3286 015410 010167 163544      MOV  R1,SAVR1
3287 015414 010067 163536      MOV  R0,SAVR0
3288 015420 000002      RTI
3289
3290      ;RESTORE R0-R5
3291
3292 015422 016700 163530      .RES05: MOV  SAVR0,R0
3293 015426 016701 163526      MOV  SAVR1,R1
3294 015432 016702 163524      MOV  SAVR2,R2
3295 015436 016703 163522      MOV  SAVR3,R3
3296 015442 016704 163520      MOV  SAVR4,R4
3297 015446 016705 163516      MOV  SAVR5,R5
3298 015452 000002      RTI
3299
3300      ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3301
3302 015454 104402      .CONVR: TYPE
3303 015456 017254      MCRLF
3304 015460 010046      .CNVRT: MOV  R0,-(SP)
3305 015462 010146      MOV  R1,-(SP)
3306 015464 010346      MOV  R3,-(SP)
3307 015466 010446      MOV  R4,-(SP)
3308 015470 010546      MOV  R5,-(SP)
3309 015472 017601 000012      MOV  @12(SP),R1
3310 015476 016767 002402 163444      MOV  TEMP,TEMP3
3311 015504 062766 000002 000012      ADD  #2,12(SP)
3312 015512 012167 000154      MOV  (R1)+,WRDCNT
3313 015516 112167 000152      1$:  MOVB (R1)+,CHRCNT
3314 015522 112167 000147      MOVB (R1)+,SPACNT
3315 015526 013167 000144      MOV  @ (R1)+,BINWRD
3316 015532 016704 000140      2$:  MOV  BINWRD,R4
3317 015536 116705 000132      MOVB CHRCNT,R5
3318 015542 012700 020104      MOV  #TEMP,R0
3319 015546 010403      3$:  MOV  R4,R3
3320 015550 042703 177770      BIC  #177770,R3
3321 015554 062703 000060      ADD  #060,R3
3322 015560 110320      MOVB R3,(R0)+
3323 015562 006204      ASR  R4
3324 015564 042704 100000      BIC  #100000,R4
3325 015570 006204      ASR  R4
3326 015572 006204      ASR  R4
3327 015574 005305      DEC  R5

```

```

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

```

3328	015576	001363			BNE	3\$	
3329	015600	012703	020144		MOV	#MDATA,R3	
3330	015604	114023		4\$:	MOVB	-(R0),(R3)+	
3331	015606	105367	000062		DECB	CHRCNT	
3332	015612	001374			BNE	4\$	
3333	015614	105767	000055		TSTB	SPACNT	
3334	015620	001405			BEQ	6\$	
3335	015622	112723	000040	5\$:	MOVB	#040,(R3)+	
3336	015626	105367	000043		DECB	SPACNT	
3337	015632	001373			BNE	5\$	
3338	015634	105013		6\$:	CLRB	(R3)	
3339	015636	104402			TYPE		
3340	015640	020144			MDATA		
3341	015642	005367	000024		DEC	WRDCNT	
3342	015646	001323			BNE	1\$	
3343	015650	016767	163274 002226		MOV	TEMP3,TEMP	
3344	015656	012605			MOV	(SP)+,R5	
3345	015660	012604			MOV	(SP)+,R4	
3346	015662	012603			MOV	(SP)+,R3	
3347	015664	012601			MOV	(SP)+,R1	
3348	015666	012600			MOV	(SP)+,R0	
3349	015670	000002			RTI		
3350	015672	000000			WRDCNT:	0	
3351	015674	000000			CHRCNT:	0	
3352		015675			SPACNT=CHRCNT+1		
3353	015676	000000			BINWRD:	0	
3354							
3355							
3356							
3357							
3358							
3359							
3360	015700	017605	000000		.SETFLG:MOV	@(SP),R5	
3361	015704	122767	000116 002132		CMPB	#'N',INBUF	:IS IT 'N' ?
3362	015712	001002			BNE	1\$	
3363	015714	105015			CLRB	(R5) ;000	
3364	015716	000406			BR	2\$	
3365	015720	122767	000131 002116	1\$:	CMPB	#'Y',INBUF	:IS IT 'Y' ?
3366	015726	001005			BNE	3\$	
3367	015730	112715	177777		MOVB	#-1,(R5)	:377
3368	015734	062716	000002	2\$:	ADD	#2,(SP)	
3369	015740	000002			RTI		
3370	015742	104404		3\$:	INSTR		:RETRY
3371	015744	000755			BR	.SETFLG	
3372							
3373							
3374							
3375							
3376							
3377	015746	011646			.TRPSR:MOV	(SP),-(SP)	:GET PC OF RETURN
3378	015750	162716	000002		SUB	#2,(SP)	:PC OF TRAP
3379	015754	017616	000000		MOV	@(SP),(SP)	:GET TRP
3380	015760	006316		TRPOK:	ASL	(SP)	:MULTIPLY TRAP ARG BY 2
3381	015762	042716	177001		BIC	#177001,(SP)	:CLEAR UNWANTED BITS
3382	015766	062716	001366		ADD	#.TRPTAB,(SP)	:POINTER TO SUBROUTINE ADDRESS
3383	015772	017616	000000		MOV	@(SP),(SP)	:SUBROUTINE ADDRESS

3384	015776	000136		JMP	@(SP)+	;GO TO SUBROUTINE
3385						
3386						
3387						
3388	016000	104413		.HLT:	CKSWR	;CHECK FOR ^G
3389	016002	032777	020000 163070	BIT	#SW13,@SWR	;INHIBIT ERROR TYPE OUT ?
3390	016010	001061		BNE	HALTS	
3391	016012	021667	163116	CMP	(SP),LSTERR	
3392	016016	001404		BEQ	1\$	
3393	016020	011667	163110	MOV	(SP),LSTERR	
3394	016024	105067	163216	CLRB	ERRFLG	
3395	016030	104406		1\$:	SAV05	
3396	016032	011605		MOV	(SP),R5	
3397	016034	162705	000002	SUB	#2,R5	
3398	016040	011504		MOV	(R5),R4	
3399	016042	006304		ASL	R4	
3400	016044	061504		ADD	(R5),R4	
3401	016046	006304		ASL	R4	
3402	016050	042704	177001	BIC	#177001,R4	
3403	016054	062704	020660	ADD	#.ERRTAB,R4	
3404	016060	012467	000040	MOV	(R4)+,ERRMSG	
3405	016064	012467	000046	MOV	(R4)+,DATAHD	
3406	016070	011467	000054	MOV	(R4),DATABP	
3407	016074	105767	163146	TSTB	ERRFLG	
3408	016100	001403		BEQ	TYPMSG	
3409	016102	005767	000042	TST	DATABP	
3410	016106	001014		BNE	TYPDAT	
3411	016110	104410		TYPMSG:	CONVRT	
3412	016112	016242		ERTAB0		
3413	016114	112767	177777 163124	MOVB	#-1,ERRFLG	
3414	016122	104402		TYPE		
3415	016124	000000		ERRMSG:	0	
3416	016126	005767	000004	TST	DATAHD	
3417	016132	001402		BEQ	TYPDAT	
3418	016134	104402		TYPE		
3419	016136	000000		DATAHD:	0	
3420	016140	005767	000004	TYPDAT:	TST	DATABP
3421	016144	001402		BEQ	RESREG	
3422	016146	104410		CONVRT		
3423	016150	000000		DATABP:	0	
3424	016152	104407		RESREG:	RES05	
3425	016154	005777	162720	HALTS:	TST	@SWR
3426	016160	100005		BPL	EXITER	
3427	016162	010046		PUSHRO		
3428	016164	016600	000002	MOV	2(SP),R0	
3429	016170	000000		HALT		
3430	016172	012600		POPPO		
3431	016174	104413		EXITER:	CKSWR	;CHECK FOR ^G
3432	016176	005267	162730	INC	ERRCNT	
3433	016202	032777	000400 162670	BIT	#SW08,@SWR	;LOOP ON ERROR ?
3434	016210	001007		BNE	1\$	
3435	016212	032777	002000 162660	BIT	#SW10,@SWR	;ESCAPE TO NEXT ON ERROR ?
3436	016220	001407		BEQ	2\$	
3437	016222	016767	162670 162664	MOV	NEXT,RTRN	;SET UP FOR NEXT TEST
3438	016230	012706	001100	1\$:	MOV	#CTACK,SP
3439	016234	000177	162654	JMP	@RTRN	;REINITIALIZE SP

```

3440 016240 000002      2$: RTI
3441 016242 000001      ERTAB0: 1
3442 016244 006        002      .BYTE 6,2
3443 016246 001174      SAVPC
3444      :ENTER HERE ON POWER FAILURE
3445
3446
3447 016250 010046      .PFAIL: MOV R0,-(SP)      ;SAVE R0-R5 ON PROCESSOR STACK
3448 016252 010146      MOV R1,-(SP)
3449 016254 010246      MOV R2,-(SP)
3450 016256 010346      MOV R3,-(SP)
3451 016260 010446      MOV R4,-(SP)
3452 016262 010546      MOV R5,-(SP)
3453 016264 016746 161534      MOV 24,-(SP)
3454 016270 010667 162676      MOV SP,SAVSP      ;SAVE STACK POINTER
3455 016274 012767 016306 161522      MOV #RESTART,24      ;SET UP FOR POWER UP TRAP
3456 016302 000000      HALT      ;HALT ON POWER DOWN NORMAL
3457 016304 000777      1$: BR 1$
3458
3459      :PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3460
3461 016306 016706 162660      RESTAR: MOV SAVSP,SP      ;RESTORE STACK POINTER
3462 016312 012605      MOV (SP)+,R5      ;RESTORE R0-R5
3463 016314 012604      MOV (SP)+,R4
3464 016316 012603      MOV (SP)+,R3
3465 016320 012602      MOV (SP)+,R2
3466 016322 012601      MOV (SP)+,R1
3467 016324 012600      MOV (SP)+,R0
3468 016326 012767 016250 161470      MOV #.PFAIL,24      ;SET UP FOR POWER FAILURE
3469 016334 012767 000340 161434      MOV #340,PS
3470 016342 012706 001100      MOV #STACK,SP
3471 016346 005067 001532      CLR TEMP
3472 016352 005267 001526      1$: INC TEMP
3473 016356 001375      BNE 1$
3474 016360 104410      CONVRT
3475 016362 016404      PFTAB
3476 016364 104402      TYPE
3477 016366 017257      MPFAIL
3478 016370 005067 162652      CLR ERRFLG
3479 016374 005067 162534      CLR LSTERR
3480 016400 000177 162510      JMP @RTRN
3481 016404 000001      PFTAB: 1
3482 016406 006        002      .BYTE 6,2
3483 016410 001114      RTRN
3484
3485
3486      :CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
3487      :OF LOC.176.
3488      :LOCATIONS USED:
3489 016412 000000      RDSW: .WORD 0
3490
3491
3492 016414 005737 000042      .CKSWR: TST @#42
3493 016420 001042      BNE OUT
3494 016422 022767 000176 162450      CMP #CJREG,SWR      ;SOFTWARE SWITCH REGISTER PRESENT
3495 016430 001036      BNE OUT      ;NO, GET OUT
  
```


3496	016432	105777	162446		TSTB	@TKCSR		:YES, WAIT FOR
3497	016436	100033			BPL	OUT		:READY, GET CHARACTER
3498	016440	017767	162442	176352	MOV	@TKDBR, .MSG		:AND STRIP OFF
3499	016446	042767	177600	176344	BIC	#177600, .MSG		:THE GARBAGE
3500	016454	122767	000007	176336	CMPB	#7, .MSG		:IS IT A <^G>
3501	016462	001021			BNE	OUT		
3502	016464	104402	016542		TYPE, \$CNTG			
3503	016470	005137	016412		.CNTLU: COM	@#RDSW		
3504	016474	104402	016547		TYPE, \$MSWR			
3505	016500	104411	016534		CNVRT, SWREGC			
3506	016504	104403	016557		INSTR, \$MNEW			
3507	016510	104405			PARAM			
3508	016512	000000			0			
3509	016514	177777			177777			
3510	016516	000176			SWREG			
3511	016520	000	001		.BYTE	0, 1		
3512	016522	104402	017254		TYPE, MCRLF			
3513	016526	005037	016412		OUT: CLR	@#RDSW		
3514	016532	000002			RTI			
3515	016534	000001			SWREGC: 1			
3516	016536	006	002		.BYTE	6, 2		
3517	016540	000176			SWREG			
3518	016542	005015	043536	000	\$CNTG: .ASCIZ	<15><12>/^G/		
3519	016547	015	051412	051127	\$MSWR: .ASCIZ	<15><12>/SWR= /		
3520	016554	020075	000					
3521	016557	040	047040	053505	\$MNEW: .ASCIZ	/ NEW= /		
3522	016564	020075	000					
3523		016570			.EVEN			
3524	016570	005015	042012	030525	MTITLE: .ASCIZ	<15><12><12>/DU11 CZDUD-D TAPE D /<15><12>		
3525	016576	020061	055103	052504				
3526	016604	026504	020104	040524				
3527	016612	042520	042040	006440				
3528	016620	000012						
3529	016622	005015	042526	052103	MVECTO: .ASCIZ	<15><12>/VECTOR ADDRESS-/		
3530	016630	051117	040440	042104				
3531	016636	042522	051523	000055				
3532	016644	005015	051461	020124	MREGAD: .ASCIZ	<15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/		
3533	016652	042504	044526	042503				
3534	016660	020072	042522	042503				
3535	016666	053111	051105	041440				
3536	016674	047117	051124	046117				
3537	016702	051040	043505	051511				
3538	016710	042524	020122	042101				
3539	016716	051104	051505	026523				
3540	016724	000						
3541	016725	015	040412	042522	MMULT: .ASCIZ	<15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/		
3542	016732	054440	052517	051040				
3543	016740	047125	044516	043516				
3544	016746	046440	046125	044524				
3545	016754	046120	020105	042504				
3546	016762	044526	042503	020123				
3547	016770	020077	054450	047440				
3548	016776	020122	024516	000055				
3549	017004	005015	040514	052123	MLASTD: .ASCIZ	<15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/		
3550	017012	042040	053105	041511				
3551	017020	035105	042522	042503				

3552 017026 053111 051105 041440
 3553 017034 047117 051124 046117
 3554 017042 051040 043505 051511
 3555 017050 042524 020122 042101
 3556 017056 051104 051505 026523
 3557 017064 000
 3558 017065 075 042504 044526
 3559 017072 042503 020040 000
 3560 017077 015 044012 053517
 3561 017104 047040 053517 041040
 3562 017112 047522 047127 041440
 3563 017120 053517 020077 027056
 3564 017126 051456 046105 041505
 3565 017134 020124 047523 042515
 3566 017142 044124 047111 020107
 3567 017150 047524 051040 047125
 3568 017156 040040 041501 051124
 3569 017164 043505 000
 3570 017167 015 047412 052125
 3571 017174 047440 020106 040522
 3572 017202 043516 035105 042522
 3573 017210 054524 042520 046040
 3574 017216 051501 020124 042504
 3575 017224 044526 042503 051040
 3576 017232 041530 051123 040440
 3577 017240 042104 042522 051523
 3578 017246 000055
 3579 017250 020040 000077
 3580 017254 005015 000
 3581 017257 040 050040 053517
 3582 017264 051105 043040 044501
 3583 017272 052514 042522 020054
 3584 017300 051120 043517 040522
 3585 017306 020115 042522 052123
 3586 017314 051101 020124 052101
 3587 017322 052040 051505 020124
 3588 017330 047111 050040 047522
 3589 017336 051107 051505 000123
 3590 017344 005015 047105 020104
 3591 017352 043117 050040 051501
 3592 017360 020123 040524 042520
 3593 017366 042040 000
 3594 017371 015 051012 000
 3595 017375 015 052012 051505
 3596 017402 020124 041520 000055
 3597 017410 005015 047514 045503
 3598 017416 047440 020116 042523
 3599 017424 042514 052103 042105
 3600 017432 052040 051505 037524
 3601 017440 024040 020131 051117
 3602 017446 047040 026451 000
 3603 017453 015 042012 020125
 3604 017460 051120 047511 044522
 3605 017466 054524 046040 053105
 3606 017474 046105 000055
 3607 017500 005015 020043 043117

DEVICE: .ASCIIZ /=DEVICE /

MCOW: .ASCIIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/

MRANGE: .ASCIIZ <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/

MQM: .ASCIIZ / ?/

MCRLF: .ASCIIZ <15><12>

MPFAIL: .ASCIIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/

MEPASS: .ASCIIZ <15><12>/END OF PASS TAPE D/

MR: .ASCIIZ <15><12>/R/

MTSTPC: .ASCIIZ <15><12>/TEST PC-/

MLOCK: .ASCIIZ <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/

MLEVEL: .ASCIIZ <15><12>/DU PRIORITY LEVEL-/

MSYNC: .ASCIIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/

3608 017506 051440 047131 020103
 3609 017514 044103 051101 020123
 3610 017522 042523 042514 052103
 3611 017530 042105 024040 030440
 3612 017536 047440 020122 024462
 3613 017544 000055
 3614 017546 005015 051511 051440
 3615 017554 041505 054040 044515
 3616 017562 020124 052512 050115
 3617 017570 051105 021440 020066
 3618 017576 047111 020077 054450
 3619 017604 047440 020122 024516
 3620 017612 000055
 3621 017614 005015 051511 051440
 3622 017622 041505 051040 041505
 3623 017630 045040 046525 042520
 3624 017636 020122 032443 044440
 3625 017644 037516 024040 020131
 3626 017652 051117 047040 026451
 3627 017660 000
 3628 017661 015 044412 020123
 3629 017666 050117 020124 046103
 3630 017674 020122 047105 041101
 3631 017702 042514 045040 046525
 3632 017710 042520 020122 032043
 3633 017716 044440 037516 024040
 3634 017724 020131 051117 047040
 3635 017732 026451 000
 3636 017735 015 044412 020123
 3637 017742 044124 020105 042524
 3638 017750 052123 041440 047117
 3639 017756 042516 052103 051117
 3640 017764 044440 051516 040524
 3641 017772 046114 042105 037440
 3642 020000 054450 047440 020122
 3643 020006 024516 000055
 3644 020012 006412 020040 020040
 3645 020020 052123 052101 051525
 3646 020026 020040 046440 050101
 3647 020034 020040 020040 005040
 3648 020042 000015
 3649
 3650
 3651
 3652
 3653 020044 000040
 3654 020104 000040
 3655 020144 000040
 3656
 3657
 3658
 3659
 3660
 3661 020204 006367 000044
 3662 020210 006367 000040
 3663 020214 006367 000034

MWIRE6: .ASCIIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/

MWIRE5: .ASCIIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/

MWIRE4: .ASCIIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/

MEXTJ: .ASCIIZ <15><12>/IS THE TEST CONNECTOR INSTALLED ?(Y OR N)-/

MSTATUS: .ASCIIZ <12> <15>/ STATUS MAP / <12> <15>

.EVEN

:BUFFERS FOR INPUT-OUTPUT

INBUF: .BLKB 40
 TEMP: .BLKB 40
 MDATA: .BLKB 40

:*****

:UTILITIES

:*****

:THIS UTILITY CALCULATES PRIORITY LEVEL

DULEV: ASL DUPRT :SHIFT LEFT
 ASL DUPRT :
 ASL DUPRT :

```

3664 020220 006367 000030      ASL      DUPRT      :
3665 020224 006367 000024      ASL      DUPRT      :
3666 020230 016767 000020 000020  MOV      DUPRT,LESS1  :MOVE THIS TO LESS1
3667 020236 162767 000001 000012  SUB      #1,LESS1      :CREATE LESS1
3668 020244 042767 000037 000004  BIC      #37,LESS1     :CLEAR TNZVC
3669 020252 000207                RTS      PC
3670 020254 000240                DUPRT: LEVEL5
3671 020256 000200                LESS1: LEVEL4 ;LEVEL TO ALLOW INTERRUPTS
3672
3673 ;NEW DU ADDRESSES
3674 020260 016767 000126 000422 DUADDR: MOV      DUBASE,RXCSR  :XXX0
3675 020266 005267 000120                INC      DUBASE
3676 020272 016767 000114 000412  MOV      DUBASE,HRXCSR  :XXX1
3677 020300 005267 000106                INC      DUBASE
3678 020304 016767 000102 000402  MOV      DUBASE,RXDBUF  :XXX2
3679 020312 016767 000074 000400  MOV      DUBASE,PARCSR  :XXX2
3680 020320 005267 000066                INC      DUBASE
3681 020324 016767 000062 000364  MOV      DUBASE,HRXDBUF :XXX3
3682 020332 016767 000054 000362  MOV      DUBASE,HPARCSR :XXX3
3683 020340 005267 000046                INC      DUBASE
3684 020344 016767 000042 000352  MOV      DUBASE,TXCSR   :XXX4
3685 020352 005267 000034                INC      DUBASE
3686 020356 016767 000030 000342  MOV      DUBASE,HTXCSR  :XXX5
3687 020364 005267 000022                INC      DUBASE
3688 020370 016767 000016 000332  MOV      DUBASE,TXDBUF  :XXX6
3689 020376 005267 000010                INC      DUBASE
3690 020402 016767 000004 000322  MOV      DUBASE,HTXDBUF :XXX7
3691 020410 000207                RTS      PC
3692 020412 000000                DUBASE: 0
3693
3694 ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3695 ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3696 ;SHIFTED IN BY THE CONTENTS OF SHIFT
3697 020414 042777 040000 000302 RPOKE: BIC      #MTDATA,@TXCSR
3698 020422 005067 160520                CLR      TEMP2
3699 020426 006067 160512                ROR      TEMP1      :FORCE CARRY
3700 020432 006067 160510                ROR      TEMP2      :PICK UP CARRY IN BIT 15
3701 020436 006267 160504                ASR      TEMP2      :SHIFT INTO BIT 14
3702 020442 042767 100000 160476  BIC      #BIT15,TEMP2  :CLR BIT 15
3703 020450 056777 160472 000246  BIS      TEMP2,@TXCSR  :POKE MAINT DATA
3704 020456 042777 020000 000240  BIC      #CLK,@TXCSR   :POKE CLK
3705 020464 052777 020000 000232  BIS      #CLK,@TXCSR   :
3706 020472 005367 160442                DEC      SHIFT
3707 020476 001346                BNE      RPOKE
3708 020500 000207                RTS      PC
3709
3710 ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3711 020502 016767 160436 160436 ODD8: MOV      TEMP1,TEMP2  :SAVE TEMP1
3712 020510 005067 160434                CLR      TEMP3
3713 020514 012727 000010                MOV      #8.,(PC)+
3714 020520 000000                4$: 0
3715 020522 006067 160420                1$: ROR      TEMP2
3716 020526 005567 160416                ADC      TEMP3
3717 020532 005367 177762                DEC      4$
3718 020536 001371                BNE      1$
3719 020540 006067 160404                ROR      TEMP3
  
```



```

3720 020544 103404      BCS      2$
3721 020546 052767 000400 160370  BIS      #BIT8,TEMP1      ;SET ODD PARITY
3722 020554 000403      BR        3$
3723 020556 042767 000400 160360 2$:    BIC      #BIT8,TEMP1      ;CLR EVEN PARITY
3724      :TEMP1 NOW HAS ODD PARITY CHARACTER
3725 020564 000207      3$:    RTS      PC
3726
3727      ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3728 020566 016767 160352 160352  EVEN8:  MOV      TEMP1,TEMP2      ;SAVE TEMP1
3729 020574 005067 160350      CLR      TEMP3
3730 020600 012727 000010      MOV      #8.,(PC)+
3731 020604 000000      4$:    0
3732 020606 006067 160334      1$:    ROR      TEMP2
3733 020612 005567 160332      ADC      TEMP3
3734 020616 005367 177762      DEC      4$
3735 020622 001371      BNE      1$
3736 020624 006067 160320      ROR      TEMP3
3737 020630 103004      BCC      2$
3738 020632 052767 000400 160304  BIS      #BIT8,TEMP1      ;SET EVEN PARITY
3739 020640 000403      BR        3$
3740 020642 042767 000400 160274 2$:    BIC      #BIT8,TEMP1      ;CLR ODD PARITY
3741      :TEMP1 NOW HAS EVEN PARITY CHARACTER
3742 020650 000207      3$:    RTS      PC
3743 020652 062716 000002  TRPREG:  ADD      #2,(SP) ;ALLOW IT TO 'CRUNCH' INTO HLT BACK
3744      ;IN MAIN PART OF THE PROGRAM
3745 020656 000002      RTI
3746      ;ERROR HLT TABLE
3747 020660 020744      .ERRTAB:  EMO      ;HLT 0 BIT ERROR (GENERAL)
3748 020662 000000      0
3749 020664 000000      0
3750 020666 020760      EM1      ;HLT 1 REGISTER ERROR
3751 020670 021131      DH1
3752 020672 021152      DT1
3753 020674 021022      EM2      ;HLT 2 RECEIVER ERROR
3754 020676 021131      DH1
3755 020700 021152      DT1
3756 020702 021064      EM3      ;HLT 3 TRANSMITTER ERROR
3757 020704 021131      DH1
3758 020706 021152      DT1
3759      ;DEFAULT DU ADDRESSES
3760 020710 160040      RXCSR: 160040
3761 020712 160041      HRXCSR: 160041
3762 020714 160042      RXDBUF: 160042
3763 020716 160043      HRXDBUF: 160043
3764 020720 160042      PARCSR: 160042
3765 020722 160043      HPARCSR: 160043
3766 020724 160044      TXCSR: 160044
3767 020726 160045      HTXCSR: 160045
3768 020730 160046      TXDBUF: 160046
3769 020732 160047      HTXDBUF: 160047
3770      ;DEFAULT DU VECTORS
3771 020734 000770      DURIV: 770      ;REC INTR VECTOR
3772 020736 000772      DURIS: 772      ;REC INTR STATUS
3773 020740 000774      DUTIV: 774      ;XMIT INTR VECTOR
3774 020742 000776      DUTIS: 776      ;XMIT INTR STATUS
3775      ;ERROR MESSAGES
  
```

3776	020744	036440	042440	051122	EM0:	.ASCIIZ / = ERROR PC/
3777	020752	051117	050040	000103		
3778	020760	036440	051040	043505	EM1:	.ASCIIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3779	020766	051511	042524	020122		
3780	020774	051105	047522	020122		
3781	021002	041520	005015	051001		
3782	021010	043505	051511	042524		
3783	021016	020122	000040			
3784	021022	036440	051040	041505	EM2:	.ASCIIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
3785	021030	044505	042526	020122		
3786	021036	051105	047522	020122		
3787	021044	041520	005015	051001		
3788	021052	043505	051511	042524		
3789	021060	020122	000040			
3790	021064	036440	052040	040522	EM3:	.ASCIIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3791	021072	051516	044515	052124		
3792	021100	051105	042440	051122		
3793	021106	051117	050040	006503		
3794	021114	000412	042522	044507		
3795	021122	052123	051105	020040		
3796	021130	000				
3797						
3798	021131	105	050130	041505	DH1:	:DATA HEADERS FOR ERROR MESSAGES
3799	021136	042524	020104	040440		.ASCIIZ /EXPECTED ACTUAL/
3800	021144	052103	040525	000114		
3801						.EVEN
3802						
3803	021152	000003			DT1:	:DATA TABLES FOR ERROR MESSAGES
3804	021154	006	004			3
3805	021156	001164				.BYTE 6.4
3806	021160	006	004			SAVR3 :REGISTER
3807	021162	001156				.BYTE 6.4
3808	021164	006	002			SAVR0 :EXPECTED DATA
3809	021166	001160				.BYTE 6.2
3810		000001				SAVR1 :ACTUAL DATA
						.END

AAA	002520	1179#																	
ACTREG	001216	899#	1063*	1127*	1141*	1142*	1149*	1231	3061	3064	3074								
ADDBAS	001306	939#	1059	1226*															
ADDKEE	001310	940#	1060	1227*															
ADDLAS	001312	941#	1061	1228*															
ADDROT	001322	945#	1064	1232*															
ADRCNT=	015361	3226*	3265*	3274#															
A10010=	***** GX	678																	
BASEAD	001204	894#	1059*	1066*	1109*	1146*	1147	1153*	1155*	1226	3068*	3080*	3084						
BASEDU	001364	962#	1073	1239*															
BASEIV	001212	897#	1065*	1119*	1230	3069*	3081*	3086	3087*	3088	3089*	3090	3091*	3092					
		3093*																	
BBB	002336	1126	1130#																
BINWRD	015676	3315*	3316	3353#															
BITW =	002000	809#	1355	1419	1483	1547	1611	1675	1739	1803	1864	1919	1974	2029					
		2085	2140	2195	2250	2306	2361	2416	2471	2527	2582	2637	2692	2748					
		2803	2858	2913	2968	3023													
BIT0 =	000001	757#	785	816															
BIT1 =	000002	756#	784																
BIT10 =	002000	747#	775	809															
BIT11 =	004000	746#	774																
BIT12 =	010000	745#	773	790															
BIT13 =	020000	744#	772	789	808														
BIT14 =	040000	743#	771	788	807														
BIT15 =	100000	742#	770	787	806	3702													
BIT2 =	000004	755#	783	1294															
BIT3 =	000010	754#	782	815															
BIT4 =	000020	753#	781	814															
BIT5 =	000040	752#	780	813															
BIT6 =	000100	751#	779	812															
BIT7 =	000200	750#	778	811															
BIT8 =	000400	749#	777	793	810	3721	3723	3738	3740										
BIT9 =	001000	748#	776	792															
BREAK =	000001	816#																	
BRW	014712	1291	1305	3154#															
BRX	014714	1292	1306	3155#															
BUFF1	000002R	678#	1268	1274															
BUFHRX	001346	955#	1077	1243*															
BUFHTX	001362	961#	1083	1249*															
BUFRXD	001344	954#	1076	1242*															
BUFTXD	001360	960#	1082	1248*															
BYJMR	001305	935#	1058	1225*															
CARDET=	010000	773#																	
CCC	014446	3060	3083	3100#															
CHRCNT	015674	3313*	3317	3331*	3351#	3352													
CKSWR =	104413	994#	3113	3135	3159	3388	3431												
CLK =	020000	808#	1350	1351	1357	1358	1369	1414	1415	1421	1422	1433	1478	1479					
		1485	1486	1497	1542	1543	1549	1550	1561	1606	1607	1613	1614	1625					
		1670	1671	1677	1678	1689	1734	1735	1741	1742	1753	1798	1799	1805					
		1806	1817	1859	1860	1866	1867	1878	1914	1915	1921	1922	1933	1969					
		1970	1976	1977	1988	2024	2025	2031	2032	2043	2080	2081	2087	2088					
		2099	2135	2136	2142	2143	2154	2190	2191	2197	2198	2209	2245	2246					
		2252	2253	2264	2301	2302	2308	2309	2320	2356	2357	2363	2364	2375					
		2411	2412	2418	2419	2430	2466	2467	2473	2474	2485	2522	2523	2529					
		2530	2541	2577	2578	2584	2585	2596	2632	2633	2639	2640	2651	2687					
		2688	2694	2695	2706	2743	2744	2750	2751	2762	2798	2799	2805	2806					

		2817	2853	2854	2860	2861	2872	2908	2909	2915	2916	2927	2963	2964
		2970	2971	2982	3018	3019	3025	3026	3037	3704	3705			
CLROPT	001303	933#	1056	1223*										
CNTLU =	104414	996#	1041											
CNVRT =	104411	990#	3505											
CONVRT=	104410	988#	1277	3057	3411	3422	3474							
COUNT	001142	870#												
COUNT1	001230	910#	1267*	1279*										
CSRHPA	001352	957#	1079	1245*										
CSRHRX	001342	953#	1075	1241*										
CSRHTX	001356	959#	1081	1247*										
CSRPAR	001350	956#	1078	1244*										
CSRRX	001340	952#	1074	1240*										
CSRTX	001354	958#	1080	1246*										
CTS =	020000	772#												
D =	***** GX	677												
DATABP	016150	3406*	3409	3420	3423#									
DATAHD	016136	3405*	3416	3419#										
DEVADR	015356	3224*		3262	3272#									
DEVICE	017065	3058	3558#											
DH1	021131	3751	3754	3757	3798#									
DISPRE	000174	837#	1034											
DLIGHT=	177570	727#	1025											
DMULT	001304	934#	1057	1224*										
DNA =	100000	806#												
DNAINT=	000040	813#												
DSC =	100000	770#												
DSINTE=	000040	780#												
DSR =	001000	776#												
DSWR =	177570	726#	1024											
DTR =	000002	784#												
DT1	021152	3752	3755	3758	3803#									
DUADDR	020260	1108	3085	3674#										
DUBASE	020412	1073*	1104	1107	1239	3084*	3674	3675*	3676	3677*	3678	3679	3680*	3681
		3682	3683*	3684	3685*	3686	3687*	3688	3689*	3690	3692#			
DULEV	020204	1174	3661#											
DUPRT	020254	1072*	1171	1233	3661*	3662*	3663*	3664*	3665*	3666	3670#			
DURIS	020736	1068*	1235	3088*	3772#									
DURIV	020734	1067*	1115	1118	1119	1234	3086*	3093	3771#					
DUTIS	020742	1070*	1237	3092*	3774#									
DUTIV	020740	1069*	1236	3090*	3773#									
EIGHT =	006000	801#	1536	1792	2516	2571	2626	2681	2737	2792	2847	2902	2957	3012
EMO	020744	3747	3776#											
EM1	020760	3750	3778#											
EM2	021022	3753	3784#											
EM3	021064	3756	3790#											
ERRCNT	001132	863#	1015*	3432*										
ERRFLG	001246	922#	1014*	3102*	3148*	3394*	3407	3413*	3478*					
ERRMSG	016124	3404*	3415#											
ERTABO	016242	3412	3441#											
EVENB	020566	3728#												
EVEPAR=	001400	804#	1853	1963	2074	2184	2295	2405	2516	2626	2737	2847	2957	
EVPAR =	000400	793#												
EXITER	016174	3426	3431#											
FILBUF	003204	1270#	1273											
FIVE =	000000	798#	1344	1600	1853	1908	1963	2018						

FLAG	001222	907#	1089*	1260	1264*														
FRMERR=	020000	789#																	
HALTS	016154	3390	3425#																
HDXEN =	000010	815#																	
HEREU	003132	1219	1260#																
HILIM	015354	3223*	3253	3271#															
HOLD	001136	868#																	
HOLDO	001224	908#	1265*	1281															
HOLD1	001226	909#	1266*	1282															
HPARCS	020722	1079*	1245	3682*	3765#														
HRXCSR	020712	1075*	1241	3676*	3761#														
HRXDBU	020716	1077*	1243	3681*	3763#														
HTXCSR	020726	1081*	1247	3686*	3767#														
HTXDBU	020732	1083*	1249	3690*	3769#														
ICOUNT	001122	859#	3146	3150*															
INBUF	020044	1182	1186	3187	3229	3361	3365	3653#											
INIFLG	001244	920#	1020	1023*															
INSTR=	104404	980#	1190	3248	3370														
INSTR =	104403	978#	1099	1110	1120	1131	1156	1166	1180	1193	1197	1201	1205	1296					
		1309	3506																
INSTR2	015140	3198	3213#																
ISYMOD=	000000	797#	1955	1963	2010	2018	2176	2184	2231	2239	2397	2405	2452	2460					
		2618	2626	2673	2681														
IVBASE	001314	942#	1065	1230*															
IVKEEP	001316	943#	1062	1229*															
JMRBY	001203	890#	1058*	1208	1225														
KEEPA	001206	895#	1060*	1066	1107*	1109	1153	1155	1227	3080									
KEEPIV	001214	898#	1062*	1118*	1229	3081													
LASTAD	001210	896#	1061*	1136	1147	1161	1228												
LESS1	020256	1071*	1238	3666*	3667*	3668*	3671#												
LEVEL0=	000000	767#																	
LEVEL1=	000040	766#																	
LEVEL2=	000100	765#																	
LEVEL3=	000140	764#																	
LEVEL4=	000200	763#	3671																
LEVEL5=	000240	762#	3670																
LEVEL6=	000300	761#																	
LEVEL7=	000340	760#																	
LIGHTS	001102	848#	1025*	1034*	3104*														
LIMITS	015302	3239	3253#																
LOBITS	015360	3225*	3257	3273#	3274														
LOCK	001120	858#	3162																
LOGICA	014500	3108#																	
LOKFLG	001247	923#	1299	1300															
LOLIM	015352	3222*	3255	3270#															
LPCNT	001124	860#	1011*	3145*	3146	3149*													
LSTERR	001134	864#	1016*	3101*	3391	3393*	3479*												
L1ESS	001336	951#	1071	1238*															
MAP	001634	1038	1040	1048#															
MCOW	017077	3063	3560#																
MCRLF	017254	1278	3195	3211	3303	3512	3580#												
MDATA	020144	3329	3340	3655#															
MEPASS	017344	3056	3590#																
MEXT =	010000	820#																	
MEXTJ	017735	1206	3636#																
MINT =	004000	819#	1341	1405	1469	1533	1597	1661	1725	1789	1850	1905	1960	2015					

		2071	2126	2181	2236	2292	2347	2402	2457	2513	2568	2623	2678	2734
MITSEX	001301	2789	2844	2899	2954	3009								
MLASTD	017004	931#	1054	1221*										
MLEVEL	017453	1132	3549#											
MLOCK	017410	1167	3603#											
MMULT	016725	1297	3597#											
MPFAIL	017257	1121	3541#											
MQM	017250	3477	3581#											
MR	017371	3208	3579#											
MRANGE	017167	1319	3594#											
MREGAD	016644	1157	3570#											
MRESET=	000400	1100	3532#											
		810#	1335	1337	1399	1401	1463	1465	1527	1529	1591	1593	1655	1657
		1719	1721	1783	1785	1844	1846	1899	1901	1954	1956	2009	2011	2065
		2067	2120	2122	2175	2177	2230	2232	2286	2288	2341	2343	2396	2398
		2451	2453	2507	2509	2562	2564	2617	2619	2672	2674	2728	2730	2783
		2785	2838	2840	2893	2895	2948	2950	3003	3005				
MSTATU	020012	1263	3644#											
MSYNC	017500	1181	3607#											
MTDATA=	040000	807#	3697											
MTITLE	016570	1022	3524#											
MTSTPC	017375	1310	3595#											
MULTD	001202	889#	1057*	1123	1124	1224	1288	3059						
MVECTO	016622	1111	3529#											
MWIRE4	017661	1202	3628#											
MWIRE5	017614	1198	3621#											
MWIRE6	017546	1194	3614#											
NEXT	001116	857#	1334*	1398*	1462*	1526*	1590*	1654*	1718*	1782*	1843*	1898*	1953*	2008*
		2064*	2119*	2174*	2229*	2285*	2340*	2395*	2450*	2506*	2561*	2616*	2671*	2727*
		2782*	2837*	2892*	2947*	3002*	3131	3151	3437					
NOPAR =	000000	802#	1344	1408	1472	1536	1600	1664	1728	1792				
NOSYNC	001300	930#	1053	1220*										
ODDPAR=	001000	803#	1908	2018	2129	2239	2350	2460	2571	2681	2792	2902	3012	
ODD8	020502	3711#												
ONCE	001522	1021	1024#											
OPTCLR	001201	888#	1056*	1204	1223									
OUT	016526	3493	3495	3497	3501	3513#								
OUTCRY	014440	3057	3096#											
OUTMUL	002476	1129	1154	1165#										
OVRUN=	040000	788#												
PARAM =	104405	982#	1101	1112	1133	1158	1168	1311	3507					
PARAM1	015206	3228#	3249											
PARCSR	020720	1078*	1244	1336*	1344*	1400*	1408*	1464*	1472*	1528*	1536*	1592*	1600*	1656*
		1664*	1720*	1728*	1784*	1792*	1845*	1853*	1900*	1908*	1955*	1963*	2010*	2018*
		2066*	2074*	2121*	2129*	2176*	2184*	2231*	2239*	2287*	2295*	2342*	2350*	2397*
		2405*	2452*	2460*	2508*	2516*	2563*	2571*	2618*	2626*	2673*	2681*	2729*	2737*
		2784*	2792*	2839*	2847*	2894*	2902*	2949*	2957*	3004*	3012*	3679*	3764#	
		792#												
PAREN =	001000	790#												
PARER =	010000													
PARERR	015262	3231	3233	3235	3244#	3254	3256	3258						
PARTI	015344	3247	3267#											
PASCNT	001130	862#	1013*	3103*	3104									
PF TAB	016404	3475	3481#											
POPPO =	012600	736#	3430											
POP1SP=	005726	734#												
POP2SP=	022626	738#												

TEMP2	001146
TEMP3	001150
TEMP4	001152
TEMP5	001154
THRU	003262
TISDU	001332
TIVDU	001330
TKCSR	001104

1732*	1750*	1796*	1814*	1857*	1875*	1912*	1930*	1967*	1985*	2022*	2040*	2078*
2096*	2133*	2151*	2188*	2206*	2243*	2261*	2299*	2317*	2354*	2372*	2409*	2427*
2464*	2482*	2520*	2538*	2575*	2593*	2630*	2648*	2685*	2703*	2741*	2759*	2796*
2814*	2851*	2869*	2906*	2924*	2961*	2979*	3016*	3034*	3706*			
799#	1408	1664	2074	2129	2184	2239						
3314*	3333	3336*	3352#									
775#												
729#	1009	1285	3438	3470								
930#	1269	1272										
782#												
921#	1012*											
777#												
3282#												
847#	1024*	1029	1033*	1039	1048	1086	1218	1294	1307	3136	3143	3160
3389	3425	3433	3435	3494								
838#	1033	1039	3494	3510	3517							
3505	3515#											
709#	1086											
708#	1307											
707#												
706#												
705#												
704#												
703#												
702#	1048	1218										
701#	3433											
700#	3160											
699#	3435											
698#	3143											
697#												
696#	3389											
695#	3136											
694#												
885#	1053*	1184*	1188*	1220								
796#	1592	1600	1656	1664	1720	1728	1784	1792				
795#	1336	1344	1400	1408	1464	1472	1528	1536	1845	1853	1900	1908
2066	2074	2121	2129	2287	2295	2342	2350	2508	2516	2563	2571	2729
2737	2784	2792	2839	2847	2894	2902	2949	2957	3004	3012		
781#												
821#												
911#	1275*	1276*	1277									
3310	3318	3343*	3471*	3472*	3654#							
871#	1347*	1353*	1411*	1417*	1475*	1481*	1539*	1545*	1603*	1609*	1667*	1673*
1731*	1737*	1795*	1801*	1856*	1862*	1911*	1917*	1966*	1972*	2021*	2027*	2077*
2083*	2132*	2138*	2187*	2193*	2242*	2248*	2298*	2304*	2353*	2359*	2408*	2414*
2463*	2469*	2519*	2525*	2574*	2580*	2629*	2635*	2684*	2690*	2740*	2746*	2795*
2801*	2850*	2856*	2905*	2911*	2960*	2966*	3015*	3021*	3699*	3711	3721*	3723*
3728	3738*	3740*										
872#	3698*	3700*	3701*	3702*	3703	3711						

TKDBR	001106	850#	3141	3191	3199	3498								
TLAST =	013760	1313	3049#											
TPCSR	001110	851#	3172	3200										
TPDBR	001112	852#	3174*	3199*										
TRPOK	015760	3380#												
TRPREG	020652	3743#												
TSTNO	001126	861#	1017*	1333*	1397*	1461*	1525*	1589*	1653*	1717*	1781*	1842*	1897*	1952*
		2007*	2063*	2118*	2173*	2228*	2284*	2339*	2394*	2449*	2505*	2560*	2615*	2670*
		2726*	2781*	2836*	2891*	2946*	3001*							
		1312	1318	1333#	3114	3115								
TST1	003446	1843	1897#											
TST10	006210	1898	1952#											
TST11	006432	1953	2007#											
TST12	006654	2008	2063#											
TST13	007076	2064	2118#											
TST14	007320	2119	2173#											
TST15	007542	2174	2228#											
TST16	007764	2229	2284#											
TST17	010206	2285	2339#											
TST18	010430	2340	2394#											
TST19	010652	1334	1397#											
TST2	003700	2395	2449#											
TST20	011074	2450	2505#											
TST21	011316	2506	2560#											
TST22	011540	2561	2615#											
TST23	011762	2616	2670#											
TST24	012204	2671	2726#											
TST25	012426	2727	2781#											
TST26	012650	2782	2836#											
TST27	013072	2837	2891#											
TST28	013314	2892	2946#											
TST29	013536	1398	1461#											
TST3	004132	2947	3001#	3049										
TST30	013760	3002												
TST31 =	***** U	1462	1525#											
TST4	004364	1526	1589#											
TST5	004616	1590	1653#											
TST6	005050	1654	1717#											
TST7	005302	1718	1781#											
TST8	005534	1782	1842#											
TST9	005766	1291*	1292*	1302*	1303*	1305*	1306*	3137#						
TTST	014614	1080*	1246	1335*	1337*	1341*	1345	1350*	1351*	1357*	1358*	1359	1369*	1371
TXCSR	020724	1379	1399*	1401*	1405*	1409	1414*	1415*	1421*	1422*	1423	1433*	1435	1443
		1463*	1465*	1469*	1473	1478*	1479*	1485*	1486*	1487	1497*	1499	1507	1527*
		1529*	1533*	1537	1542*	1543*	1549*	1550*	1551	1561*	1563	1571	1591*	1593*
		1597*	1601	1606*	1607*	1613*	1614*	1615	1625*	1627	1635	1655*	1657*	1661*
		1665	1670*	1671*	1677*	1678*	1679	1689*	1691	1699	1719*	1721*	1725*	1729
		1734*	1735*	1741*	1742*	1743	1753*	1755	1763	1783*	1785*	1789*	1793	1798*
		1799*	1805*	1806*	1807	1817*	1819	1827	1844*	1846*	1850*	1854	1859*	1860*
		1866*	1867*	1868	1878*	1880	1899*	1901*	1905*	1909	1914*	1915*	1921*	1922*
		1923	1933*	1935	1954*	1956*	1960*	1964	1969*	1970*	1976*	1977*	1978	1988*
		1990	2009*	2011*	2015*	2019	2024*	2025*	2031*	2032*	2033	2043*	2045	2065*
		2067*	2071*	2075	2080*	2081*	2087*	2088*	2089	2099*	2101	2120*	2122*	2126*
		2130	2135*	2136*	2142*	2143*	2144	2154*	2156	2175*	2177*	2181*	2185	2190*
		2191*	2197*	2198*	2199	2209*	2211	2230*	2232*	2236*	2240	2245*	2246*	2252*
		2253*	2254	2264*	2266	2286*	2288*	2292*	2296	2301*	2302*	2308*	2309*	2310

		2320*	2322	2341*	2343*	2347*	2351	2356*	2357*	2363*	2364*	2365	2375*	2377
		2396*	2398*	2402*	2406	2411*	2412*	2418*	2419*	2420	2430*	2432	2451*	2453*
		2457*	2461	2466*	2467*	2473*	2474*	2475	2485*	2487	2507*	2509*	2513*	2517
		2522*	2523*	2529*	2530*	2531	2541*	2543	2562*	2564*	2568*	2572	2577*	2578*
		2584*	2585*	2586	2596*	2598	2617*	2619*	2623*	2627	2632*	2633*	2639*	2640*
		2641	2651*	2653	2672*	2674*	2678*	2682	2687*	2688*	2694*	2695*	2696	2706*
		2708	2728*	2730*	2734*	2738	2743*	2744*	2750*	2751*	2752	2762*	2764	2783*
		2785*	2789*	2793	2798*	2799*	2805*	2806*	2807	2817*	2819	2838*	2840*	2844*
		2848	2853*	2854*	2860*	2861*	2862	2872*	2874	2893*	2895*	2899*	2903	2908*
		2909*	2915*	2916*	2917	2927*	2929	2948*	2950*	2954*	2958	2963*	2964*	2970*
		2971*	2972	2982*	2984	3003*	3005*	3009*	3013	3018*	3019*	3025*	3026*	3027
TXDBUF	020730	3037*	3039	3684*	3697*	3703*	3704*	3705*	3766#					
		1082*	1248	1346*	1410*	1474*	1538*	1602*	1666*	1730*	1794*	1855*	1910*	1965*
		2020*	2076*	2131*	2186*	2241*	2297*	2352*	2407*	2462*	2518*	2573*	2628*	2683*
		2739*	2794*	2849*	2904*	2959*	3014*	3688*	3768#					
TXDONE=	000200	811#												
TXINTE=	000100	812#												
TYPDAT	016140	3410	3417	3420#										
TYPE =	104402	976#	1022	1263	1278	1319	3055	3058	3063	3185	3195	3207	3211	3302
		3339	3414	3418	3476	3502	3504	3512						
TYPMSG	016110	3408	3411#											
UP	003224	1275#	1280											
USER =	000000	818#												
VOID =	000001	785#												
WRDCNT	015672	3312*	3341*	3350#										
ZERO	014256	3069#												
\$CNTG	016542	3502	3518#											
\$E =	000040	678#	1334	1335#	1398	1399#	1462	1463#	1526	1527#	1590	1591#	1654	1655#
		1718	1719#	1782	1783#	1843	1844#	1898	1899#	1953	1954#	2008	2009#	2064
		2065#	2119	2120#	2174	2175#	2229	2230#	2285	2286#	2340	2341#	2395	2396#
		2450	2451#	2506	2507#	2561	2562#	2616	2617#	2671	2672#	2727	2728#	2782

CZDUD-D MACY11 30A(1052) 21-NOV-78 15:46 PAGE 85
CZDUDD.P11 18-JUL-78 15:54

D 7

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0081

.INST1	015016	3185#	3196	3212		
.MSG	015020	3183*	3186#	3498*	3499*	3500
.PARAM	015146	983	3219#			
.PFAIL	016250	827	1010	3447#	3468	
.RES05	015422	987	3292#			
.SAV05	015362	985	3278#			
.SCOPE	014532	973	3119#			
.SCOP1	014716	975	3159#			
.SETFL	015700	993	3360#	3371		
.START	001420	839	1008#	1018	3067	
.TRPSR	015746	831	3377#			
.TRPTA	001366	969#	3382			
.TYPE	014736	977	3167#			

HLT	739# 1637 1994 2424 2823	1363 1683 2037 2436 2866	1375 1695 2049 2479 2878	1381 1701 2093 2491 2921	1427 1747 2105 2535 2933	1439 1759 2148 2547 2976	1445 1765 2160 2590 2988	1491 1811 2203 2602 3031	1503 1823 2215 2645 3043	1509 1829 2258 2657	1555 1872 2270 2700	1567 1884 2314 2712	1573 1927 2326 2756	1619 1939 2369 2768	1631 1982 2381 2811
PRGEND	678#	3048													
PRGFRT	678#	679													
PUSYF	678#														
RSETUP	678#														
TSETUP	678# 2175 3003	1335 2230	1399 2286	1463 2341	1527 2396	1591 2451	1655 2507	1719 2562	1783 2617	1844 2672	1899 2728	1954 2783	2009 2838	2065 2893	2120 2948
\$BEGIN	678#	1209													
\$BINAR	678#														
\$BUFFE	678#	3650													
\$CABLE	678#														
\$CATCH	678#	822													
\$CLRVE	678#	1024													
\$CONVR	678#	3299													
\$DNA	678#	1322	1386	1450	1514	1578	1642	1706	1770						
\$EOP	678#	3048													
\$GETFL	678#	1120	1193	1197	1201	1205	1296								
\$GETPA	678#	1099	1110	1130	1156	1165	1309								
\$GETSY	678#	1175													
\$HEADE	678#	679													
\$HLT	678#	3385													
\$INSTR	678#	3178													
\$ISOB	678#														
\$MATCH	678#														
\$MRR	678#														
\$MRRW	678#														
\$MRW	678#														
\$MSG	678#	3524													
\$PARAM	678#	3216													
\$PFAIL	678#	3444													
\$POKE	678# 1798 2196 2632 3024	1350 1804 2245 2638	1356 1859 2251 2687	1414 1865 2301 2693	1420 1914 2307 2743	1478 1920 2356 2749	1484 1969 2362 2798	1542 1975 2411 2804	1548 2024 2417 2853	1606 2030 2466 2859	1612 2080 2472 2908	1670 2086 2522 2914	1676 2135 2528 2963	1734 2141 2577 2969	1740 2190 2583 3018
\$POKER	678#														
\$RCNET	678#														
\$RECAC	678#														
\$REG	678#	3275													
\$RESET	678# 1783 2177 2617 3005	1335 1785 2230 2619	1337 1844 2232 2672	1399 1846 2286 2674	1401 1899 2288 2728	1463 1901 2341 2730	1465 1954 2343 2783	1527 1956 2396 2785	1529 2009 2398 2838	1591 2011 2451 2840	1593 2065 2453 2893	1655 2067 2507 2895	1657 2120 2509 2948	1719 2122 2562 2950	1721 2175 2564 3003
\$RXACT	678#														
\$SCOPE	678#	3116													
\$SCOP1	678#	3156													
\$SETFL	678#	3354													
\$SETVE	678#	823													
\$START	678#	1000													
\$STRIP	678#														

CZDUD-D MACY11 30A(1052) 21-NOV-78 15:46 PAGE 88
CZDUDD.P11 18-JUL-78 15:54

F 7

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0083

\$SYMB0	678#	691													
\$SYNCR	678#														
\$TRAPS	678#	964													
\$TRPAR	678#	1834	1889	1944	1999	2055	2110	2165	2220	2276	2331	2386	2441	2497	2552
	2607	2662	2718	2773	2828	2883	2938	2993							
\$TRPDE	678#	972	974	976	978	980	982	984	986	988	990	992	994	996	
\$TRPSR	678#	3372													
\$TSTNO	678#	1333	1397	1461	1525	1589	1653	1717	1781	1842	1897	1952	2007	2063	2118
	2173	2228	2284	2339	2394	2449	2505	2560	2615	2670	2726	2781	2836	2891	2946
	3001														
\$TYPE	678#	3164													
\$UNIBU	678#														
\$VARIA	678#	842													
\$WORDF	678#														
\$WORDO	678#														
\$WORDP	678#														

. ABS. 021170 000

ERRORS DETECTED: 0

CZDUDD.BIN,CZDUDD.SEQ/CRF/SOL/NL:TOC=CZDU11.HLO/EQ:RUND,CZDU11.PAR,CZDU11.KET,CZDUDD.P11

RUN-TIME: 8 12 1 SECONDS

RUN-TIME RATIO: 132/22=5.9

CORE USED: 19K (37 PAGES)