

DU11

RECEIVER TIMING TESTS CZDUCD0

AH-8685D MC

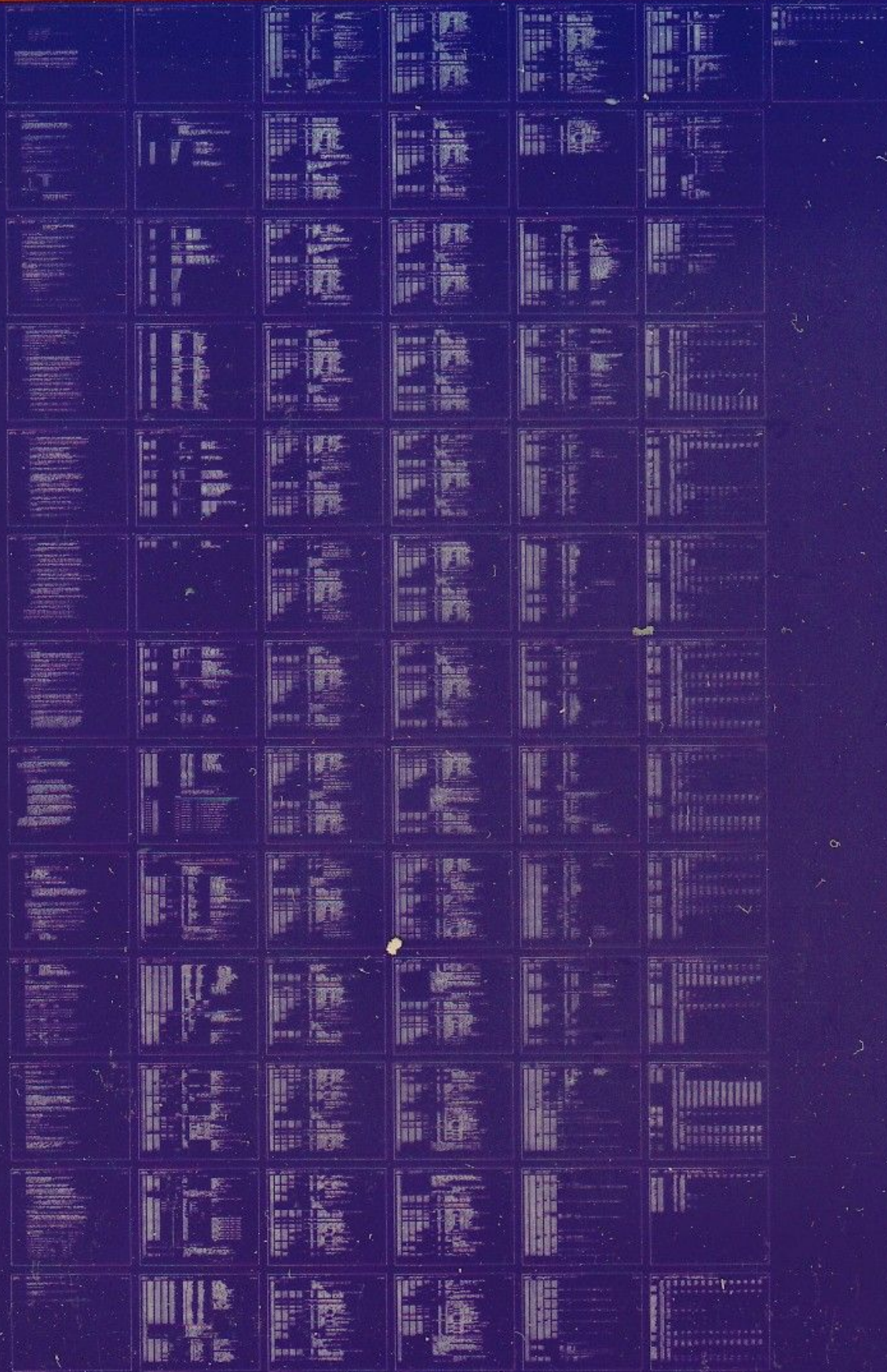
COPYRIGHT 73-78

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-8684D-MC

PRODUCT NAME: CZDUCDO DU11 OFFLINE RCVR TIMING 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 RECEIVER TIMING TESTS VERIFY THAT THE RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT AT THE PROPPER TIME

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 CZDUC-D TAPE C' (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 ALLWS
THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
REGISTER (LOC. 176) IS USED.

CONTROL :

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

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

5.1

OPERATIONAL SWITCH SETTINGS
SW15 =1 HALT ON ERROR
SW14 =1 LOOP ON CURRENT TEST
SW13 =1 INHIBIT ERROR TYPEOUT
SW11 =1 INHIBIT ITERATIONS

SW10 =1 ESCAPE TO NEXT TEST ON ERROR
SW08 =1 LOOP ON ERROR
SW07 =1 USE STATUS MAP PARAMETERS
SW02 =1 LOCK ON TEST
SW01 =1 RESTART PROGRAM AT SELECTED TEST
SW00 =1 RESELECT VECTOR AND CONTROL REGISTER ADDRESSES
&PARAMETERS AFTER A PROGRAM RESTART
TO INHIBIT 'END OF PASS' TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

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

6.1.1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2
REFER TO THE ABOVE 'HLT'' IN DIAGNOSTIC FOR ERROR DESCRIPTION
CHECK ADDRESS @ RXCSR: TO LOCATE THE DEVICE PRESENTLY UNDER
TEST WHEN RUNNING MULTIPLE DEVICES

6.1.2 PC +2 = REGISTER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ
WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER
WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER
WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ
WHERE 16XXXX IS THE ADDRESS OF THE FAILING RECEIVER (RXDBUF) REGISTER
WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER
WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6.1.4 PC +2 = TRANSMITTER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ
WHERE 16XXXX IS THE ADDRESS OF THE FAILING TRANSMITTER (TXCSR) REGISTER
WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER
WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

- 6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS
- 6.2 ERROR RECOVERY
 - 6.2.1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING
 - 6.2.2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

NOTE: THE PC + 2 OF THE 'HLT' WILL BE DISPLAYED IN THE DATA LIGHTS
 - 6.2.3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER. THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED. THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR.
 - 6.2.4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT: & PASCNT:
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY.
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS.
- 6.3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM:

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF
- 7. RESTRICTIONS
 - 7.1 MULTIPLE DEVICES
UP TO 16(10) DEVICES MAY BE TESTED. HOWEVER, THEY
MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

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

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

- 7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED, SIMPLY RESTART
PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.
- 7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED
 - 7.2.2.1 TO RESTART...LOAD 000200 IN SWR AND DEPRESS START....
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.
 - 7.2.2.2ORLOAD 000200 WITH SW00 =1 AND DEPRESS START....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
....THE PROGRAM WILL CONTINUE WITH DEVICE 0
 - 7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM
WILL TYPEOUT AN ERROR MESSAGE.....LOAD & START AT 000200
- 7.3 CABLE DELAYS
NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)
 - 7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
LOCATION 'HOLD:'' MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
PRESENTLY 'HOLD:'' =20 IS SUFFICIENT TIME ON AN 11/20 MACHINE.
IF RUNNING ON AN 11/40 OR AN 11/45 'HOLD:'' MUST BE PATCHED TO 40

BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS
- 7.4 TO USE THE 'XOR' TESTER ,THE BRANCH AROUND THE 'XOR'
CODE MUST BE PATCHED TO A 'NOP'. (SEE LISTINGS FOR DETAILS)
- 8. DEFAULT PARAMETERS:
1S: 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: C

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 SEREC: 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 RECEIVER TIMING TESTING
OF THE DEVICE
SEE LISTING FOR DETAILS
10. FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW
11. LISTINGS

CZDUC-D
CZDU11.PAR

MACY11 30A(1052) 21-NOV-78 15:40 PAGE 15
21-NOV-78 15:17

B 2

SEQ 0014

675
676 000000' 000000G

D

677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710

.ENABLE ABS

:DU11 CZDUC-D TAPE C
:COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

:STARTING PROCEDURE
:LOAD PROGRAM
:PRESS START
:PROGRAM WILL TYPE 'DU11 CZDUC-D TAPE C ''
:PROGRAM WILL TYPE 'R' TO INDICATE THAT TESTING HAS STARTED
:AT THE END OF A PASS, PROGRAM WILL TYPE 'END OF PASS TAPE C''
:AND THEN RESUME TESTING

:SWITCH REGISTER OPTIONS

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

SW15=100000
SW14=40000
SW13=20000
SW12=10000
SW11=4000
SW10=2000
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

=1,HALT ON ERROR
=1,LOOP ON CURRENT TEST
=1,INHIBIT ERROR TYPEOUT

=1,INHIBIT ITERATIONS
=1,ESCAPE TO NEXT TEST ON ERROR
=1,LOOP WITH CURRENT DATA
=1,LOOP ON ERROR
+ =1, USE STATUS MAP

:LOCK ON TEST SELECT
:RESTART PROGRAM AT SELECTED TEST
:RESELECT VECTOR AND CONTROL REGISTER
:ADDRESS AFTER PROGRAM RESTART

```
711
712
713 ;REGISTER DEFINITIONS
714 000000 R0=%0 ;GENERAL REGISTER
715 000001 R1=%1 ;GENERAL REGISTER
716 000002 R2=%2 ;GENERAL REGISTER
717 000003 R3=%3 ;GENERAL REGISTER
718 000004 R4=%4 ;GENERAL REGISTER
719 000005 R5=%5 ;GENERAL REGISTER
720 000006 SP=%6 ;PROCESSOR STACK POINTER
721 000007 PC=%7 ;PROGRAM COUNTER
722
723 ;LOCATION EQUIVALENCIES
724
725 177570 DSWR=177570 ;HARDWARE SWITCH REGISTER LOC.
726 177570 DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
727 177776 PS=177776 ;PROCESSOR STATUS WORD
728 001100 STACK=1100 ;START OF PROCESSOR STACK
729
730 ;INSTRUCTION DEFINITIONS
731
732 005746 PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
733 005726 POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
734 010046 PUSHRO=10046 ;SAVE R0 ON STACK =MOV R0,-(SP)
735 012600 POPRO=12600 ;RESTORE R0 FROM STACK =MOV (SP)+,R0
736 024646 PUSH2SP=24646 ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
737 022626 POP2SP=22626 ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
738 .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
739
740
741 100000 BIT15=100000
742 040000 BIT14=40000
743 020000 BIT13=20000
744 010000 BIT12=10000
745 004000 BIT11=4000
746 002000 BIT10=2000
747 001000 BIT9=1000
748 000400 BIT8=400
749 000200 BIT7=200
750 000100 BIT6=100
751 000040 BIT5=40
752 000020 BIT4=20
753 000010 BIT3=10
754 000004 BIT2=4
755 000002 BIT1=2
756 000001 BIT0=1
757
758 ;PROCESSER LEVELS
759 000340 LEVEL7=340
760 000300 LEVEL6=300
761 000240 LEVEL5=240
762 000200 LEVEL4=200
763 000140 LEVEL3=140
764 000100 LEVEL2=100
765 000040 LEVEL1=040
766 000000 LEVEL0=000
```



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

```
822                                     ;STANDARD INTERRUPT VECTORS
823
824
825                                     .=24
826 000024 015664                       .PFAIL                       ;POWER FAIL HANDLER
827 000026 000340                       340                          ;SERVICE AT LEVEL 7
828 000030 015414                       .HLT                          ;ERROR HANDLER
829 000032 000340                       340                          ;SERVICE AT LEVEL 7
830 000034 015362                       .TRPSRV                       ;GENERAL HANDLER DISPATCH SERVICE
831 000036 000340                       340                          ;SERVICE AT LEVEL 7
832
833                                     ;SOFTWARE SWITCH REGISTER
834
835                                     .=174
836 000174 000000                       DISPREG: .WORD 0             ;SOFTWARE DISPLAY REG.
837 000176 000000                       SWREG:  .WORD 0             ;SOFTWARE SWITCH REGISTER
838 000200 000167 001214                 JMP      .START              ;GO TO START OF PROGRAM
839
840
841
842                                     .=1100
843
844                                     ;INDIRECT POINTERS
845
846 001100 177570                       SWR:      177570             ;SWITCH REGISTER POINTER
847 001102 177570                       LIGHTS:177570             ;DISPLAY REGISTER POINTER
848 001104 177560                       TKCSR:   177560             ;TELETYPE KEYBOARD CONTROL REGISTER
849 001106 177562                       TKDBR:   177562             ;TELETYPE KEYBOARD DATA BUFFER
850 001110 177564                       TPCSR:   177564             ;TELEPRINTER CONTROL REGISTER
851 001112 177566                       TPDBR:   177566             ;TELEPRINTER DATA BUFFER
852
853                                     ;PROGRAM CONTROL PARAMETERS
854
855 001114 000000                       RTRN:    0                  ;SCOPE ADDRESS FOR LOOP ON TEST
856 001116 000000                       NEXT:    0                  ;ADDRESS OF NEXT TEST TO BE EXECUTED
857 001120 000000                       LOCK:    0                  ;ADDRESS FOR LOCK ON CURRENT DATA
858 001122 000000                       ICOUNT:  0                  ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
859 001124 000000                       LPCNT:   0                  ;NUMBER OF ITERATIONS COMPLETED
860 001126 000000                       TSTNO:   0                  ;NUMBER OF TEST IN PROGRESS
861 001130 000000                       PASCNT:  0                  ;NUMBER OF PASSES COMPLETED
862 001132 000000                       ERRCNT:  0                  ;TOTAL NUMBER OF ERRORS
863 001134 000000                       LSTERR:  0                  ;PC OF LAST ERROR CALL
864
865                                     ;PROGRAM VARIABLES
866
867 001136 000020                       HOLD:    20                 ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
868 001140 000000                       SHIFT:   0                  ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
869 001142 000000                       COUNT:   0                  ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
870 001144 000000                       TEMP1:   0                  ;TEMPORARY STORAGE
871 001146 000000                       TEMP2:   0                  ;TEMPORARY STORAGE
872 001150 000000                       TEMP3:   0                  ;TEMPORARY STORAGE
873 001152 000000                       TEMP4:   0                  ;TEMPORARY STORAGE
874 001154 000000                       TEMP5:   0                  ;TEMPORARY STORAGE
875 001156 000000                       SAVR0:   0                  ;R0 STORAGE
876 001160 000000                       SAVR1:   0                  ;R1 STORAGE
877 001162 000000                       SAVR2:   0                  ;R2 STORAGE
```


878 001164 000000
879 001166 000000
880 001170 000000
881 001172 000000
882 001174 000000

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

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

```

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

```


939 001310 000001
 940 001312 000001
 941 001314 000001
 942 001316 000001
 943 001320 000001
 944 001322 000001
 945 001324 000001
 946 001326 000001
 947 001330 000001
 948 001332 000001
 949 001334 000001
 950 001336 000001
 951 001340 000001
 952 001342 000001
 953 001344 000001
 954 001346 000001
 955 001350 000001
 956 001352 000001
 957 001354 000001
 958 001356 000001
 959 001360 000001
 960 001362 000001
 961 001364 000001

ADDKEEP: .BLKW 1 ;SAVED 1ST DEVICE ADDR
 ADDLAST: .BLKW 1 ;LAST DEVICE RXCSR ADDR
 IVBASE: .BLKW 1 ;PROG CONTROLLED IV
 IVKEEP: .BLKW 1 ;SAVED INTR VECTOR
 REGACT: .BLKW 1 ;ACTIVE REGISTER
 ADDR0T: .BLKW 1 ;ROTATING POINTER
 PRTDU: .BLKW 1 ;DU11 PRIORITY
 RIVDU: .BLKW 1 ;DU11 REC INTR VECTOR
 TIVDU: .BLKW 1 ;DU11 XMIT INTR VECTOR
 TISDU: .BLKW 1 ;DU11 XMIT INTR STATUS
 RISDU: .BLKW 1 ;DU11 REC INTR STATUS
 LISS: .BLKW 1 ;PRIORITY TO ALLOW INTR
 CSRRX: .BLKW 1 ; DEFAULT OR ALTERED PARAMETERS
 CSRRHX: .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 ;DU11 RXCSR BASE ADDR
 .EVEN

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

962
 963
 964
 965
 966
 967
 968 001366
 969
 970
 971 104400
 972 001366 014146
 973 104401
 974 001370 014332
 975 104402
 976 001372 014352
 977 104403
 978 001374 014412
 979 104404
 980 001376 014530
 981 104405
 982 001400 014562
 983 104406
 984 001402 014776
 985 104407
 986 001404 015036
 987 104410
 988 001406 015070
 989 104411
 990 001410 015074
 991 104412
 992 001412 015314
 993 104413
 994 001414 016030

.TRPTAB:
 ;*****
 ;*****
 .SCOPE SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
 .SCOPI 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 !NSTER=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

```

995          104414
996 001416 016104
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007 001420 012767 000340 176350 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1008 001426 012706 001100 MOV #STACK,SP ;SET UP STACK
1009 001432 012737 015664 000024 MOV #.PFAIL,@#24 ;SET UP POWER FAIL VECTOR
1010 001440 005067 177460 CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
1011 001444 105067 177575 CLR STFLG ;CLEAR START FLAG
1012 001450 005067 177454 CLR PASCNT ;CLEAR PASS COUNT
1013 001454 105067 177566 CLR ERFLG ;CLEAR ERROR FLAG
1014 001460 005067 177446 CLR ERRCNT ;CLEAR ERROR COUNT
1015 001464 005067 177444 CLR LSTERR ;CLEAR LAST ERROR POINTER
1016 001470 012767 000001 177430 MOV #1,TSTNO ;SET UP FOR TEST 1
1017 001476 012767 001420 177410 MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
1018 ;TESTING STARTS
1019 001504 105767 177534 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1020 001510 001004 BNE ONCE
1021 001512 104402 016204 TYPE ,MTITLE ;TYPE TITLE MESSAGE
1022 001516 105167 177522 COMB INIFLG ;IF NOT SET FLAG AND DO
1023 001522 012767 177570 177350 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1024 001530 012767 177570 177344 MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1025 001536 013746 000006 MOV @#6,-(SP) ;SAVE VECTORS
1026 001542 013746 000004 MOV @#4,-(SP)
1027 001546 012737 001566 000004 MOV #64$,@#4 ;SET UP FOR TIMEOUT
1028 001554 022777 177777 177316 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1029 001562 001402 BEQ 65$
1030 001564 000407 BR 66$
1031 001566 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
1032 001570 012767 000176 177302 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1033 001576 012767 000174 177276 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
1034 001604 012637 000004 66$: MOV (SP)+,@#4 ;RESTORE VECTORS
1035 001610 012637 000006 MOV (SP)+,@#6
1036 001614 005737 000042 TST @#42 ;UNDER MONITOR
1037 001620 001005 BNE MAP
1038 001622 022767 000176 177250 CMP #SWREG,SWR ;IS SWREG USED
1039 001630 001001 BNE MAP ; BRANCH TO CHECK FOR STATUS MAP
1040 001632 104414 CNTLU
1041 ;*****
1042 ; CODE FOR STATUS MAP
1043 ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1044 ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1045 ; SETUP, OR REENTER QUESTIONING ROUTINE
1046
1047 001634 032777 000200 177236 MAP: BIT #SW07,@SWR ; IS SW07=1?
1048 001642 001537 BEQ $67 ; IF NOT, GO TO TEST FOR SW00=1
1049 ; NOW SET UP MAP VALUES FOR PROGRAM
1050 ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE

```



```

1051                                     ; OPERATION OF THIS PROGRAM.
1052 001644 116767 177430 177324      MOVB NOSYNC      ,SYNCRD      ; SYNC CHAR
1053 001652 116767 177423 177317      MOVB MITSEX      ,SEXMIT      ; XMIT JUMPER
1054 001660 116767 177416 177312      MOVB RESEC       ,SEREC       ; SEC REC JUMPER
1055 001666 116767 177411 177305      MOVB CLROPT      ,OPTCLR      ; OPTIONAL JUMPER
1056 001674 116767 177404 177300      MOVB DMULT       ,MULTD      ; MULTIPLE DEVICE
1057 001702 116767 177377 177273      MOVB BYJMR       ,JMRBY      ; EXTERNAL MODEM
1058 001710 016767 177372 177266      MOV  ADDBASE     ,BASEADD     ; PROG 1ST DEVICE ADDR
1059 001716 016767 177366 177262      MOV  ADDKEEP     ,KEEPADD     ; SAVED 1ST DEVICE ADDR
1060 001724 016767 177362 177256      MOV  ADDLAST     ,LASTADD     ; LAST DEVICE RXCSR ADDR
1061 001732 016767 177360 177254      MOV  IVKEEP      ,KEEPIV      ; SAVED INTR VECTOR
1062 001740 016767 177354 177250      MOV  REGACT      ,ACTREG      ; ACTIVE REGISTER
1063 001746 016767 177350 177244      MOV  ADDRROT     ,ROTADD      ; ROTATING POINTER
1064 001754 016767 177334 177230      MOV  IVBASE      ,BASEIV      ; BASE INTR VECTOR
1065 001762 016767 177220 177214      MOV  KEEPADD     ,BASEADD     ; RELOAD BASEADD
1066 001770 016767 177332 016352      MOV  RIVDU       ,DURIV      ; REC INTR VECTOR
1067 001776 016767 177332 016346      MOV  RISDU       ,DURIS      ; REC INTR STATUS
1068 002004 016767 177320 016342      MOV  TIVDU       ,DUTIV      ; XMIT INTR VECTOR
1069 002012 016767 177314 016336      MOV  TISDU       ,DUTIS      ; XMIT INTR STATUS
1070 002020 016767 177312 015644      MOV  LISS        ,LESS1      ; PRIORITY TO ALLOW INTR
1071 002026 013737 001324 017670      MOV  @#PRTDU     ,@#DUPRT     ; PRIORITY RELOADED
1072 002034 016767 177324 015764      MOV  BASEDU      ,DUBASE     ;
1073 002042 016767 177272 016254      MOV  CSRRX       ,RXCSR      ;
1074 002050 016767 177266 016250      MOV  CSRHRX      ,HRXCSR     ;
1075 002056 016767 177262 016244      MOV  BUFRXD      ,RXDBUF     ;
1076 002064 016767 177256 016240      MOV  BUFHRXD     ,HRXDBUF    ;
1077 002072 016767 177252 016234      MOV  CSRPAR      ,PARCSR     ;
1078 002100 016767 177246 016230      MOV  CSRHPAR     ,HPARCSR    ;
1079 002106 016767 177242 016224      MOV  CSRTX       ,TXCSR      ;
1080 002114 016767 177236 016220      MOV  CSRHTX      ,HTXCSR     ;
1081 002122 016767 177232 016214      MOV  BUFTXD      ,TXDBUF     ;
1082 002130 016767 177226 016210      MOV  BUFHTXD     ,HTXDBUF    ;
1083 002136 000167 000466      JMP  .BEGIN      ; BRANCH TO BEGIN TESTING
1084                                     ; *****
1085 002142 032777 000001 176730 $67: BIT  #SW00 ,@SWR  ; RESELECT VECTOR $ CONTROL REG?
1086 002150 001002      BNE  1$         ; BRANCH TO QUESTIONING
1087 002152 000167 000452      JMP  .BEGIN     ; GO TO LOAD STATUS MAP ETC.
1088 002156 005037 001222 $1: CLR  @#FLAG   ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1089 002162 012700 000300      MOV  #300,R0   ; RESTORE VECTOR AREA TO TRAPCATCHER
1090 002166 012701 000302      MOV  #302,R1   ; START AT LOCATION 300
1091 002172 012702 000004      MOV  #4,R2
1092 002176 010110 $2: MOV  R1,(R0)
1093 002200 005011      CLR  (R1)
1094 002202 060200      ADD  R2,R0
1095 002204 060201      ADD  R2,R1
1096 002206 022701 001000      CMP  #1000,R1 ; END AT LOCATION 776
1097 002212 002771      BLT  2$
1098 002214 104403      INSTR          ; OUTPUT MESSAGE & GET INPUT STRING
1099 002216 016260      MREGAD        ; MESSAGE
1100 002220 104405      PARAM        ; CONVERT STRING
1101 002222 160000      160000       ; LOW LIMIT
1102 002224 167776      167776       ; HIGH LIMIT
1103 002226 020026      DUBASE        ; STORE AT THIS LOCATION
1104 002230      001         ; MASK
1105 002231      001         ; HOW MANY TIMES + 2
1106 002232 016767 015570 176746 .BYTE 1
      .BYTE 1
      MOV  DUBASE,KEEPADD ;SAVE

```

1107	002240	004767	015430		JSR	PC,DUADDR		
1108	002244	016767	176736	176732	MOV	KEEPPADD,BASEADD	:RESTORE FOR ROTATION	
1109	002252	104403			INSTR		:OUTPUT MESSAGE & GET INPUT STRING	
1110	002254	016236			MVECTO		:MESSAGE	
1111	002256	104405			PARAM		:CONVERT STRING	
1112	002260	000300			300		:LOW LIMIT	
1113	002262	000776			776		:HIGH LIMIT	
1114	002264	020350			DURIV		:STORE AT THIS LOCATION	
1115	002266	001			1		:MASK	
1116	002267	004			4		:HOW MANY TIMES + 2	
1117	002270	016767	016054	176716	MOV	DURIV,KEEPIV	:SAVE	
1118	002276	016767	016046	176706	MOV	DURIV,BASEIV	:SET UP FOR ROTATION	
1119	002304	104403			INSTR		:OUTPUT MESSAGE & GET INPUT STRING	
1120	002306	016341			MMULT		:MESSAGE	
1121	002310	104412			SETFLG		:SET FLAG BASED UPON INPUT STRING	
1122	002312	001202			MULTD		:THIS FLAG	
1123	002314	105767	176662		TSTB	MULTD	:ARE THERE MULTIPLE DEVICES	
1124							:ON THE SYSTEM ?	
1125	002320	100406			BMI	BBB	:YES,ASK NEXT QUESTION	
1126	002322	005067	176670		CLR	ACTREG		
1127	002326	005067	176666		CLR	ROTADD		
1128	002332	000167	000140		JMP	OUTMUL	:JUMP AROUND NEXT QUESTION	
1129	002336				BBB:			
1130	002336	104403			INSTR		:OUTPUT MESSAGE & GET INPUT STRING	
1131	002340	016420			MLASTD		:MESSAGE	
1132	002342	104405			PARAM		:CONVERT STRING	
1133	002344	160000			160000		:LOW LIMIT	
1134	002346	167776			167776		:HIGH LIMIT	
1135	002350	001210			LASTADD		:STORE AT THIS LOCATION	
1136	002352	001			1		:MASK	
1137	002353	001			1		:HOW MANY TIMES + 2	
1138							:THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME	
1139	002354	012767	000001	176636	1\$:	MOV	#1,ROTADD	:SET UP POINTER
1140	002362	005067	176630		CLR	ACTREG	:CLR ACTIVE REGISTER	
1141	002366	056767	176626	176622	2\$:	BIS	ROTADD,ACTREG	:MAKE THIS DEVICE ACTIVE
1142	002374	000241			CLC			
1143	002376	006167	176616		ROL	ROTADD	:SET UP POINTER	
1144	002402	103421			BCS	3\$:	:ARE YOU OUT OF RANGE ?	
1145	002404	062767	000010	176572	ADD	#10,BASEADD	:SET UP BASE ADDRESS	
1146	002412	026767	176572	176564	CMP	LASTADD,BASEADD	:IS THIS THE LAST DEVICE ?	
1147	002420	101362			BHI	2\$:	:NO DO IT AGAIN	
1148	002422	056767	176572	176566	BIS	ROTADD,ACTREG	:THIS ASSUMES THAT THERE ARE AT	
1149							:LEAST TWO DEVICES WHEN YOU ANSWER YES TO	
1150							:MULTIPLE DEVICE QUESTION	
1151	002430	012767	000001	176562	4\$:	MOV	#1,ROTADD	:SET UP FOR LATER USE IN END OF PASS ROUTINE
1152	002436	016767	176544	176540	MOV	KEEPPADD,BASEADD	:DITTO	
1153	002444	000414			BR	OUTMUL	:CONTINUE QUESTIONS	
1154	002446	016767	176534	176530	3\$:	MOV	KEEPPADD,BASEADD	:RESTORE
1155	002454	104403			INSTR		:OUTPUT MESSAGE & GET INPUT STRING	
1156	002456	016603			MRANGE		:MESSAGE	
1157	002460	104405			PARAM		:CONVERT STRING	
1158	002462	160000			160000		:LOW LIMIT	
1159	002464	167776			167776		:HIGH LIMIT	
1160	002466	001210			LASTADD		:STORE AT THIS LOCATION	
1161	002470	001			1		:MASK	
1162	002471	001			1		:HOW MANY TIMES + 2	


```
1163 002472 000167 177656      JMP      1$      ;DO IT AGAIN
1164 002476
1165 002476 104403      OUTMUL:  INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1166 002500 017067      MLEVEL      ;MESSAGE
1167 002502 104405      PARAM      ;CONVERT STRING
1168 002504 000004      4          ;LOW LIMIT
1169 002506 000007      7          ;HIGH LIMIT
1170 002510 017670      DUPRT      ;STORE AT THIS LOCATION
1171 002512      000      0          ;MASK
1172 002513      001      1          ;HOW MANY TIMES + 2
1173 002514 004767 015100      JSR      PC,DULEV
1174
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
1178 002520      AAA:
1179 002520 104403      INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1180 002522 017114      MSYNC      ;MESSAGE
1181 002524 122767 000061 014726 3$:  CMPB      #'1,INBUF ;IS IT '1' ?
1182 002532 001003      BNE      1$
1183 002534 105067 176436      CLRB      SYNCNO ;000
1184 002540 000412      BR      4$
1185 002542 122767 000062 014710 1$:  CMPB      #'2,INBUF ;IS IT '2' ?
1186 002550 001004      BNE      2$
1187 002552 112767 177777 176416      MOVB      #-1,SYNCNO ;377
1188 002560 000402      BR      4$
1189 002562 104404      2$:  INSTR      ;RETRY
1190 002564 000757      BR      3$
1191 002566 000240      4$:  NOP
1192 002570 104403      INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1193 002572 017162      MWIRE6     ;MESSAGE
1194 002574 104412      SETFLG     ;SET FLAG BASED UPON INPUT STRING
1195 002576 001177      SEXMIT     ;THIS FLAG
1196 002600 104403      INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1197 002602 017230      MWIRE5     ;MESSAGE
1198 002604 104412      SETFLG     ;SET FLAG BASED UPON INPUT STRING
1199 002606 001200      SEREC     ;THIS FLAG
1200 002610 104403      INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1201 002612 017275      MWIRE4     ;MESSAGE
1202 002614 104412      SETFLG     ;SET FLAG BASED UPON INPUT STRING
1203 002616 001201      OPTCLR     ;THIS FLAG
1204 002620 104403      INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1205 002622 017351      MEXTJ     ;MESSAGE
1206 002624 104412      SETFLG     ;SET FLAG BASED UPON INPUT STRING
1207 002626 001203      JMRBY     ;THIS FLAG
1208
1209
1210      ;TEST START AND RESTART
1211
1212 002630 012767 000340 175140 .BEGIN: MOV      #340,PS ;LOCK OUT INTERRUPTS
1213      ; ***** LOAD STATUS MAP *****
1214      ; THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1215      ; USED IN THIS PROGRAM AND WILL BE PASSED TO ANY
1216      ; OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1217 002636 032777 000200 176234 BIT #SW07 ;@SWR ; SW07 SET , IF YES BRANCH
1218 002644 001132      BNE HEREU
```

```

1219 002646 116767 176324 176424
1220 002654 116767 176317 176417
1221 002662 116767 176312 176412
1222 002670 116767 176305 176405
1223 002676 116767 176300 176400
1224 002704 116767 176273 176373
1225 002712 016767 176266 176366
1226 002720 016767 176262 176362
1227 002726 016767 176256 176356
1228 002734 016767 176254 176354
1229 002742 016767 176244 176344
1230 002750 016767 176242 176342
1231 002756 016767 176236 176336
1232 002764 013737 017670 001324
1233 002772 016767 015352 176326
1234 003000 016767 015346 176326
1235 003006 016767 015342 176314
1236 003014 016767 015336 176310
1237 003022 016767 014644 176306
1238 003030 016767 014772 176326
1239 003036 016767 015262 176274
1240 003044 016767 015256 176270
1241 003052 016767 015252 176264
1242 003060 016767 015246 176260
1243 003066 016767 015242 176254
1244 003074 016767 015236 176250
1245 003102 016767 015232 176244
1246 003110 016767 015226 176240
1247 003116 016767 015222 176234
1248 003124 016767 015216 176230
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259 003132 005737 001222
1260 003136 001402
1261 003140 000167 000116
1262 003144 104402 017426
1263 003150 062737 000001 001222
1264 003156 010067 176042
1265 003162 010167 176040
1266 003166 012767 000003 176034
1267 003174 012700 000002
1268 003200 012701 001300
1269 003204 010120
1270 003206 062701 000002
1271 003212 020127 001306
1272 003216 001372
1273 003220 012700 000002
1274 003224 010067 176006

```

```

MOV B SYNCNO ,NOSYNC ; SYNC CHARS
MOV B SEXMIT ,MITSEX ; XMIT JUMPER
MOV B SEREC ,RESEC ; SEC REC JUMPER
MOV B OPTCLR ,CLROPT ; OPTIONAL JUMPER
MOV B MULTD ,DMULT ; MULTIPLE DEVICES
MOV B 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 ,CSR RX
MOV HRXCSR ,CSR HRX
MOV RXDBUF ,BUF RXD
MOV HRXDBUF ,BUF HRXD
MOV PARCSR ,CSR PAR
MOV HPARCSR ,CSR HPAR
MOV TXCSR ,CSR TX
MOV HTXCSR ,CSR HTX
MOV TXDBUF ,BUF TXD
MOV HTXDBUF ,BUF HTXD

```

```

:*****
: 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
MOV #STATUS, R1
FILBUF: MOV R1, (R0)+ ; (BUFF1)=STATUS ETC.
ADD #2, R1 ; LOAD BUFF AS ABOVE
CMP R1, #STATUS+6 ; PREPARE STATUS ADDRESS
BNE FILBUF ; CHECK IF 3 WORDS LOADED
MOV #BUFF1, R0 ; BACK TO LOAD NEXT ADDRESS
UP: MOV R0, TABLE+4 ; LOAD FOR PRINT OUT
; LOAD ADDRESS TO PRINT

```



```

1275 003230 012067 176006          MOV (R0)+, TABLE+10          : LOAD CONTENTS
1276 003234 104410 001232          CONVRT , TABLE              : PRINT ADDRESS/CONTENTS PAIR
1277 003240 104402 016670          TYPE, MCRLF                  : CR AND LF
1278 003244 005367 175760          DEC COUNT1                   : COUNT WORDS PRINTED
1279 003250 001365                   BNE UP                        : GO PRINT NEXT ADDRESS/CONTENTS
1280 003252 016700 175746          MOV HOLD0,R0
1281 003256 016701 175744          MOV HOLD1, R1
1282                                     :*****
1283
1284 003262 012706 001100          THRU: MOV #STACK,SP          :SET UP STACK
1285 003266 005737 000042          TST @#42                     :IS PROGRAM UNDER MONITOR CONTROL
1286 003272 001056                   BNE 3$
1287 003274 105767 175702          TSTB MULTD ;DON'T ALLOW LOCK ON TEST IF RUNNING
1288                                     :MULTIPLE DEVICES
1289 003300 001407                   BEQ 5$                        :IF NO,TEST FOR LOCK ON TEST
1290 003302 016767 011020 010720    MOV BRW,TTST                 :RESTORE NORMAL SCOPE LOOP
1291 003310 016767 011014 010714    MOV BRX,TTST+2               :DITTO
1292 003316 000444                   BR 3$                         :JUMP AROUND IF YES
1293 003320 032777 000004 175552 5$: BIT #BIT2,@SWR              :CHECK FOR LOCK ON TEST
1294 003326 001416                   BEQ 1$
1295 003330 104403                   INSTR                        :OUTPUT MESSAGE & GET INPUT STRING
1296 003332 017024                   MLOCK                        :MESSAGE
1297 003334 104412                   SETFLG                       :SET FLAG BASED UPON INPUT STRING
1298 003336 001247                   LOKFLG                       :THIS FLAG
1299 003340 105767 175703          TSTB LOKFLG                  :IS LOCK ON TEST OPTION SELECTED
1300 003344 001407                   BEQ 1$
1301 003346 012767 000240 010654    MOV #NOP,TTST
1302 003354 012767 000240 010650    MOV #NOP,TTST+2              :SET UP TO LOCK
1303 003362 000406                   BR 2$
1304 003364 016767 010736 010636 1$: MOV BRW,TTST
1305 003372 016767 010732 010632    MOV BRX,TTST+2              :LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1306 003400 032777 000002 175472 2$: BIT #SW01,@SWR
1307 003406 001410                   BEQ 3$                        :IF SW01=1, GET STARTING PC
1308 003410 104403                   INSTR                        :OUTPUT MESSAGE & GET INPUT STRING
1309 003412 017011                   MTSTPC                       :MESSAGE
1310 003414 104405                   PARAM                        :CONVERT STRING
1311 003416 003446                   TST1                          :LOW LIMIT
1312 003420 013214                   TLAST                        :HIGH LIMIT
1313 003422 001114                   RTRN                          :STORE AT THIS LOCATION
1314 003424 001 .BYTE 1             :MASK
1315 003425 001 .BYTE 1             :HOW MANY TIMES + 2
1316 003426 000403                   BR 4$
1317 003430 012767 003446 175456 3$: MOV #TST1,RTRN              :START AT TEST 1
1318 003436 104402 017005 4$: TYPE ,MR                      :TYPE R
1319 003442 000177 175446          JMP @RTRN                    :START TESTING
1320
1321                                     ::THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1322                                     ::SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1323                                     ::THAT IT WAS SELECTED PROPERLY
1324                                     ::FRAME ERROR (FRMERR,RXERR)
1325                                     ::MODE:ISOC (ISYMOD)
1326                                     ::LENGTH:FIVE
1327                                     ::CHAR: 25
1328                                     :
1329 003446 012767 000001 175452 TST1: MOV #1,TSTNO                ;SAVE THIS
1330 003454 012767 003622 175434    MOV #TST2,NEXT              ;GO TO THIS TEST WHEN THRU

```

```

1331 003462 052777 000400 014650      BIS      #MRESET,@TXCSR  ;MASTER RESET
1332 003470 012777 000000 014636      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1333 003476 052777 000400 014634      BIS      #MRESET,@TXCSR  ;MASTER RESET
1334
1335                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1336 003504 012777 064001 014626      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1337
1338                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1339 003512 012777 000000 014614      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
1340 003520 052777 000020 014576      BIS      #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1341                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1342 003526 042777 020000 014604      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
1343 003534 052777 020000 014576      BIS      #CLK,@TXCSR     ;POKE CLK UP
1344                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1345 003542 042777 020000 014570      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
1346 003550 052777 020000 014562      BIS      #CLK,@TXCSR     ;POKE CLK UP
1347 003556 012767 000007 175354      MOV      #7,SHIFT        ;# OF SHIFTS
1348 003564 012767 000052 175352      MOV      #52,TEMP1       ;DATA CHAR
1349                                     ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1350 003572 004767 014232      JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1351 003576 016703 014526      MOV      RXDBUF,R3       ;FOR ERROR MESSAGE
1352 003602 012700 120025      MOV      #RXERR!FRMERR!25,R0 ;EXPECTED
1353 003606 017701 014516      MOV      @RXDBUF,R1      ;ACTUAL
1354 003612 020001      CMP      R0,R1           ;COMPARE EXP VS ACT
1355 003614 001401      BEQ      64$
1356 003616 104002      HLT      2               ;FRAME ERROR & RX ERROR SHOULD BE SET
1357                                     ;IF LOWER BYTE DOES NOT MATCH IT
1358                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1359 003620      64$:
1360 003620 104400      SCOPE
1361                                     ::THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1362                                     ::SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1363                                     ::THAT IT WAS SELECTED PROPERLY
1364                                     ::FRAME ERROR (FRMERR,RXERR)
1365                                     ::MODE:ISOC (ISYMOD)
1366                                     ::LENGTH:SIX
1367                                     ::CHAR: 25
1368
1369 003622 012767 000002 175276 TST2:  MOV      #2,TSTNO      ;SAVE THIS
1370 003630 012767 003776 175260      MOV      #TST3,NEXT      ;GO TO THIS TEST WHEN THRU
1371 003636 052777 000400 014474      BIS      #MRESET,@TXCSR  ;MASTER RESET
1372 003644 012777 000000 014462      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1373 003652 052777 000400 014460      BIS      #MRESET,@TXCSR  ;MASTER RESET
1374
1375                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1376 003660 012777 064001 014452      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1377
1378                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1379 003666 012777 002000 014440      MOV      #ISYMOD!SIX!NOPAR!0,@PARCSR
1380 003674 052777 000020 014422      BIS      #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1381                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1382 003702 042777 020000 014430      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
1383 003710 052777 020000 014422      BIS      #CLK,@TXCSR     ;POKE CLK UP
1384                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1385 003716 042777 020000 014414      BIC      #CLK,@TXCSR     ;POKE CLK DOWN
1386 003724 052777 020000 014406      BIS      #CLK,@TXCSR     ;POKE CLK UP

```



```

1387 003732 012767 000010 175200
1388 003740 012767 000052 175176
1389
1390 003746 004767 014056
1391 003752 016703 014352
1392 003756 012700 120025
1393 003762 017701 014342
1394 003766 020001
1395 003770 001401
1396 003772 104002
1397
1398
1399 003774
1400 003774 104400
1401
1402
1403
1404
1405
1406
1407
1408
1409 003776 012767 000003 175122
1410 004004 012767 004152 175104
1411 004012 052777 000400 014320
1412 004020 012777 000000 014306
1413 004026 052777 000400 014304
1414
1415
1416 004034 012777 064001 014276
1417
1418
1419 004042 012777 004000 014264
1420 004050 052777 000020 014246
1421
1422 004056 042777 020000 014254
1423 004064 052777 020000 014246
1424
1425 004072 042777 020000 014240
1426 004100 052777 020000 014232
1427 004106 012767 000011 175024
1428 004114 012767 000252 175022
1429
1430 004122 004767 013702
1431 004126 016703 014176
1432 004132 012700 120125
1433 004136 017701 014166
1434 004142 020001
1435 004144 001401
1436 004146 104002
1437
1438
1439 004150
1440 004150 104400
1441
1442

```

```

MOV #8,SHIFT ;# OF SHIFTS
MOV #52,TEMP1 ;DATA CHAR
;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
JSR PC,RPOKE ;SHIFT IN THIS CHAR
MOV RXDBUF,R3 ;FOR ERROR MESSAGE
MOV #RXERR!FRMERR!25,R0 ;EXPECTED
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1 ;COMPARE EXP VS ACT
BEQ 64$
HLT 2 ;FRAME ERROR & RX ERROR SHOULD BE SET
;IF LOWER BYTE DOES NOT MATCH IT
;PROBABLY IS A LENGTH SELECT PROBLEM

64$:
SCOPE
::THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
::SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
::THAT IT WAS SELECTED PROPERLY
::FRAME ERROR (FRMERR,RXERR)
::MODE:ISOC (ISYMOD)
::LENGTH:SEVEN
::CHAR: 125
::
TST3: MOV #3,TSTNO ;SAVE THIS
MOV #TST4,NEXT ;GO TO THIS TEST WHEN THRU
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #ISYMOD,@PARCSR ;SET THE MODE
BIS #MRESET,@TXCSR ;MASTER RESET

;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
MOV #MTDATA!CLK!MINT!BREAK,@TXCSR

;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
MOV #ISYMOD!SEVEN!NOPAR!0,@PARCSR
BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
MOV #9,SHIFT ;# OF SHIFTS
MOV #252,TEMP1 ;DATA CHAR
;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
JSR PC,RPOKE ;SHIFT IN THIS CHAR
MOV RXDBUF,R3 ;FOR ERROR MESSAGE
MOV #RXERR!FRMERR!125,R0 ;EXPECTED
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1 ;COMPARE EXP VS ACT
BEQ 64$
HLT 2 ;FRAME ERROR & RX ERROR SHOULD BE SET
;IF LOWER BYTE DOES NOT MATCH IT
;PROBABLY IS A LENGTH SELECT PROBLEM

64$:
SCOPE
::THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
::SECTION ,IT USES THE ERROR FLAGS TO DETERMINE

```

```
1443                                     :: THAT IT WAS SELECTED PROPERLY
1444                                     :: FRAME ERROR (FRMERR,RXERR)
1445                                     :: MODE:ISOC (ISYMOD)
1446                                     :: LENGTH:EIGHT
1447                                     :: CHAR: 125
1448                                     ::
1449 004152 012767 000004 174746 TST4:  MOV    #4,TSTNO      ;SAVE THIS
1450 004160 012767 004326 174730      MOV    #TST5,NEXT    ;GO TO THIS TEST WHEN THRU
1451 004166 052777 000400 014144      BIS    #MRESET,@TXCSR ;MASTER RESET
1452 004174 012777 000000 014132      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1453 004202 052777 000400 014130      BIS    #MRESET,@TXCSR ;MASTER RESET
1454
1455                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1456 004210 012777 064001 014122      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1457
1458                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1459 004216 012777 006000 014110      MOV    #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1460 004224 052777 000020 014072      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1461                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1462 004232 042777 020000 014100      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1463 004240 052777 020000 014072      BIS    #CLK,@TXCSR    ;POKE CLK UP
1464                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1465 004246 042777 020000 014064      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1466 004254 052777 020000 014056      BIS    #CLK,@TXCSR    ;POKE CLK UP
1467 004262 012767 000012 174650      MOV    #10,SHIFT      ;# OF SHIFTS
1468 004270 012767 000252 174646      MOV    #252,TEMP1     ;DATA CHAR
1469
1470                                     ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1471 004276 004767 013526                JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
1472 004302 016703 014022                MOV    RXDBUF,R3      ;FOR ERROR MESSAGE
1473 004306 012700 120125                MOV    #RXERR!FRMERR!125,R0 ;EXPECTED
1474 004312 017701 014012                MOV    @RXDBUF,R1     ;ACTUAL
1475 004316 020001                        CMP    R0,R1          ;COMPARE EXP VS ACT
1476 004320 001401                        BEQ    64$
1477 004322 104002                        HLT    2              ;FRAME ERROR & RX ERROR SHOULD BE SET
1478                                     ;IF LOWER BYTE DOES NOT MATCH IT
1479                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1479 004324                                64$:
1480 004324 104400                                SCOPE
1481                                     :: THIS TEST VERIFYS EVEPAR PARITY SENSE
1482                                     :: OF THE RECEIVER
1483                                     :: MODE:ISOC (ISYMOD)
1484                                     :: PARITY:EVEPAR
1485                                     :: LENGTH:FIVE PLUS PARITY
1486                                     :: CHAR: 25
1487                                     ::
1488 004326 012767 000005 174572 TST5:  MOV    #5,TSTNO      ;SAVE THIS
1489 004334 012767 004512 174554      MOV    #TST6,NEXT    ;GO TO THIS TEST WHEN THRU
1490 004342 052777 000400 013770      BIS    #MRESET,@TXCSR ;MASTER RESET
1491 004350 012777 000000 013756      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1492 004356 052777 000400 013754      BIS    #MRESET,@TXCSR ;MASTER RESET
1493
1494                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1495 004364 012777 064001 013746      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1496
1497                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1498 004372 012777 001400 013734      MOV    #ISYMOD!FIVE!EVEPAR!0,@PARCSR
```



```

1499 004400 052777 000020 013716      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1500                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1501 004406 042777 020000 013724      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1502 004414 052777 020000 013716      BIS    #CLK,@TXCSR    ;POKE CLK UP
1503                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1504 004422 042777 020000 013710      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1505 004430 052777 020000 013702      BIS    #CLK,@TXCSR    ;POKE CLK UP
1506 004436 016703 013666                 MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
1507 004442 012700 110025                 MOV    #RXERR!PARER!25,R0 ;EXPECTED
1508 004446 012767 000010 174464      MOV    #8,SHIFT      ;# OF SHIFTS
1509 004454 012767 000252 174462      MOV    #252,TEMP1    ;DATA CHAR
1510 004462 004767 013342                 JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1511 004466 105777 013632                 TSTB   @RXCSR ;RXDONE ?
1512 004472 100401                         BMI    1$
1513 004474 104000                         HLT
1514 004476                                     1$:
1515 004476 017701 013626                 MOV    @RXDBUF,R1    ;ACTUAL
1516 004502 020001                         CMP    R0,R1        ;COMPARE EXP VS. ACT
1517 004504 001401                         BEQ    2$
1518 004506 104002                         HLT
1519 004510                                     2$:
1520                                     ;NOTE THAT THE PARITY BIT SHOULD
1521                                     ;SHOW UP IN THE DATA
1522                                     ;IE. BIT FIVE FOR FIVE LEVEL CODE
1523 004510 104400                         SCOPE
1524                                     ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1525                                     ;;OF THE RECEIVER
1526                                     ;;MODE:ISOC (ISYMOD)
1527                                     ;;PARITY:EVEPAR
1528                                     ;;LENGTH:SIX PLUS PARITY
1529                                     ;;CHAR: 25
1530                                     ;;
1531 004512 012767 000006 174406      TST6:  MOV    #6,TSTNO    ;SAVE THIS
1532 004520 012767 004676 174370      MOV    #TST7,NEXT    ;GO TO THIS TEST WHEN THRU
1533 004526 052777 000400 013604      BIS    #MRESET,@TXCSR ;MASTER RESET
1534 004534 012777 000000 013572      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1535 004542 052777 000400 013570      BIS    #MRESET,@TXCSR ;MASTER RESET
1536
1537                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1538 004550 012777 064001 013562      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1539
1540                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1541 004556 012777 003400 013550      MOV    #ISYMOD!SIX!EVEPAR!0,@PARCSR
1542 004564 052777 000020 013532      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
1543                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1544 004572 042777 020000 013540      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1545 004600 052777 020000 013532      BIS    #CLK,@TXCSR    ;POKE CLK UP
1546                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1547 004606 042777 020000 013524      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1548 004614 052777 020000 013516      BIS    #CLK,@TXCSR    ;POKE CLK UP
1549 004622 016703 013502                 MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
1550 004626 012700 110025                 MOV    #RXERR!PARER!25,R0 ;EXPECTED
1551 004632 012767 000011 174300      MOV    #9,SHIFT      ;# OF SHIFTS
1552 004640 012767 000452 174276      MOV    #452,TEMP1    ;DATA CHAR
1553 004646 004767 013156                 JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1554 004652 105777 013446                 TSTB   @RXCSR ;RXDONE ?

```

```

1555 004656 100401          BMI      1$
1556 004660 104000          HLT              ;RXDONE SHOULD BE ASSERTED
1557 004662                1$:
1558 004662 017701 013442    MOV      @RXDBUF,R1 ;ACTUAL
1559 004666 020001          CMP      R0,R1    ;COMPARE EXP VS. ACT
1560 004670 001401          BEQ     2$
1561 004672 104002          HLT      2        ;PARITY ERROR &RXERR SHOULD BE SET
1562 004674                2$:
1563                                ;NOTE THAT THE PARITY BIT SHOULD
1564                                ;SHOW UP IN THE DATA
1565                                ;IE. BIT SIX FOR SIX LEVEL CODE
1566 004674 104400          SCOPE
1567                                ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1568                                ;;OF THE RECEIVER
1569                                ;;MODE:ISOC (ISYMOD)
1570                                ;;PARITY:EVEPAR
1571                                ;;LENGTH:SEVEN PLUS PARITY
1572                                ;;CHAR: 325
1573                                ;;
1574 004676 012767 000007 174222 TST7:  MOV      #7,TSTNO ;SAVE THIS
1575 004704 012767 005062 174204    MOV      #TST8,NEXT ;GO TO THIS TEST WHEN THRU
1576 004712 052777 000400 013420    BIS      #MRESET,@TXCSR ;MASTER RESET
1577 004720 012777 000000 013406    MOV      #ISYMOD,@PARCSR ;SET THE MODE
1578 004726 052777 000400 013404    BIS      #MRESET,@TXCSR ;MASTER RESET
1579
1580                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1581 004734 012777 064001 013376    MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1582
1583                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1584 004742 012777 005400 013364    MOV      #ISYMOD!SEVEN!EVEPAR!0,@PARCSR
1585 004750 052777 000020 013346    BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1586                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1587 004756 042777 020000 013354    BIC      #CLK,@TXCSR ;POKE CLK DOWN
1588 004764 052777 020000 013346    BIS      #CLK,@TXCSR ;POKE CLK UP
1589                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1590 004772 042777 020000 013340    BIC      #CLK,@TXCSR ;POKE CLK DOWN
1591 005000 052777 020000 013332    BIS      #CLK,@TXCSR ;POKE CLK UP
1592 005006 016703 013316          MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1593 005012 012700 110325          MOV      #RXERR!PARER!325,R0 ;EXPECTED
1594 005016 012767 000012 174114    MOV      #10,SHIFT ;# OF SHIFTS
1595 005024 012767 001652 174112    MOV      #1652,TEMP1 ;DATA CHAR
1596 005032 004767 012772          JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1597 005036 105777 013262          TSTB    @RXCSR ;RXDONE ?
1598 005042 100401          BMI      1$
1599 005044 104000          HLT              ;RXDONE SHOULD BE ASSERTED
1600                                1$:
1601 005046 017701 013256    MOV      @RXDBUF,R1 ;ACTUAL
1602 005052 020001          CMP      R0,R1    ;COMPARE EXP VS. ACT
1603 005054 001401          BEQ     2$
1604 005056 104002          HLT      2        ;PARITY ERROR &RXERR SHOULD BE SET
1605                                2$:
1606                                ;NOTE THAT THE PARITY BIT SHOULD
1607                                ;SHOW UP IN THE DATA
1608                                ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
1609 005060 104400          SCOPE
1610                                ;;THIS TEST VERIFYS EVEPAR PARITY SENSE

```



```
1611      ::OF THE RECEIVER
1612      ::MODE:ISOC (ISYMOD)
1613      ::PARITY:EVEPAR
1614      ::LENGTH:EIGHT PLUS PARITY
1615      ::CHAR: 125
1616      ::
1617 005062 012767 000010 174036 TST8:  MOV    #8,TSTNO      ;SAVE THIS
1618 005070 012767 005246 174020      MOV    #TST9,NEXT    ;GO TO THIS TEST WHEN THRU
1619 005076 052777 000400 013234      BIS    #MRESET,@TXCSR ;MASTER RESET
1620 005104 012777 000000 013222      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1621 005112 052777 000400 013220      BIS    #MRESET,@TXCSR ;MASTER RESET
1622      ::
1623      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1624 005120 012777 064001 013212      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1625      ::
1626      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1627 005126 012777 007400 013200      MOV    #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1628 005134 052777 000020 013162      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1629      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1630 005142 042777 020000 013170      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1631 005150 052777 020000 013162      BIS    #CLK,@TXCSR    ;POKE CLK UP
1632      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1633 005156 042777 020000 013154      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1634 005164 052777 020000 013146      BIS    #CLK,@TXCSR    ;POKE CLK UP
1635 005172 016703 013132      MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
1636 005176 012700 110125      MOV    #RXERR!PARER!125,R0 ;EXPECTED
1637 005202 012767 000013 173730      MOV    #11,SHIFT     ;# OF SHIFTS
1638 005210 012767 003252 173726      MOV    #3252,TEMP1   ;DATA CHAR
1639 005216 004767 012606      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1640 005222 105777 013076      TSTB   @RXCSR ;RXDONE ?
1641 005226 100401      BMI    1$
1642 005230 104000      HLT
1643 005232      ;RXDONE SHOULD BE ASSERTED
1644 005232 017701 013072 1$:  MOV    @RXDBUF,R1     ;ACTUAL
1645 005236 020001      CMP    R0,R1         ;COMPARE EXP VS. ACT
1646 005240 001401      BEQ   2$
1647 005242 104002      HLT
1648 005244      ;PARITY ERROR &RXERR SHOULD BE SET
1649 005244 104400 2$:
1650      SCOPE
1651      ::THIS TEST VERIFYS ODDPAR PARITY SENSE
1652      ::OF THE RECEIVER
1653      ::MODE:ISOC (ISYMOD)
1654      ::PARITY:ODDPAR
1655      ::LENGTH:FIVE PLUS PARITY
1656      ::CHAR: 65
1657      ::
1657 005246 012767 000011 173652 TST9:  MOV    #9,TSTNO      ;SAVE THIS
1658 005254 012767 005432 173634      MOV    #TST10,NEXT   ;GO TO THIS TEST WHEN THRU
1659 005262 052777 000400 013050      BIS    #MRESET,@TXCSR ;MASTER RESET
1660 005270 012777 000000 013036      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1661 005276 052777 000400 013034      BIS    #MRESET,@TXCSR ;MASTER RESET
1662      ::
1663      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1664 005304 012777 064001 013026      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1665      ::
1666      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
```

```

1667 005312 012777 001000 013014      MOV      #ISYMOD!FIVE!ODDPAR!0,@PARCSR
1668 005320 052777 000020 012776      BIS      #SYNSCH,@RXCSR      ;SET SYNC SEARCH
1669                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1670 005326 042777 020000 013004      BIC      #CLK,@TXCSR        ;POKE CLK DOWN
1671 005334 052777 020000 012776      BIS      #CLK,@TXCSR        ;POKE CLK UP
1672                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1673 005342 042777 020000 012770      BIC      #CLK,@TXCSR        ;POKE CLK DOWN
1674 005350 052777 020000 012762      BIS      #CLK,@TXCSR        ;POKE CLK UP
1675 005356 016703 012746      MOV      RXDBUF,R3          ;SET UP FOR ERROR MESSAGE
1676 005362 012700 110065      MOV      #RXERR!PARER!65,R0  ;EXPECTED
1677 005366 012767 000010 173544      MOV      #8,SHIFT          ;# OF SHIFTS
1678 005374 012767 000352 173542      MOV      #352,TEMP1        ;DATA CHAR
1679 005402 004767 012422      JSR      PC,RPOKE          ;SHIFT IN THIS CHAR
1680 005406 105777 012712      TSTB    @RXCSR ;RXDONE ?
1681 005412 100401      BMI     1$
1682 005414 104000      HLT
1683                                     $:
1684 005416 017701 012706      MOV      @RXDBUF,R1        ;ACTUAL
1685 005422 020001      CMP     R0,R1             ;COMPARE EXP VS. ACT
1686 005424 001401      BEQ     2$
1687 005426 104002      HLT     2
1688                                     2$:
1689                                     ;NOTE THAT THE PARITY BIT SHOULD
1690                                     ;SHOW UP IN THE DATA
1691                                     ;IE. BIT FIVE FOR FIVE LEVEL CODE
1692 005430 104400      SCOPE
1693      ::THIS TEST VERIFYS ODDPAR PARITY SENSE
1694      ::OF THE RECEIVER
1695      ::MODE:ISOC (ISYMOD)
1696      ::PARITY:ODDPAR
1697      ::LENGTH:SIX PLUS PARITY
1698      ::CHAR: 125
1699      ::
1700 005432 012767 000012 173466      TST10:  MOV      #10,TSTNO    ;SAVE THIS
1701 005440 012767 005616 173450      MOV      #TST11,NEXT      ;GO TO THIS TEST WHEN THRU
1702 005446 052777 000400 012664      BIS      #MRESET,@TXCSR   ;MASTER RESET
1703 005454 012777 000000 012652      MOV      #ISYMOD,@PARCSR  ;SET THE MODE
1704 005462 052777 000400 012650      BIS      #MRESET,@TXCSR   ;MASTER RESET
1705
1706      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1707 005470 012777 064001 012642      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1708
1709      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1710 005476 012777 003000 012630      MOV      #ISYMOD!SIX!ODDPAR!0,@PARCSR
1711 005504 052777 000020 012612      BIS      #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1712                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1713 005512 042777 020000 012620      BIC      #CLK,@TXCSR        ;POKE CLK DOWN
1714 005520 052777 020000 012612      BIS      #CLK,@TXCSR        ;POKE CLK UP
1715                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1716 005526 042777 020000 012604      BIC      #CLK,@TXCSR        ;POKE CLK DOWN
1717 005534 052777 020000 012576      BIS      #CLK,@TXCSR        ;POKE CLK UP
1718 005542 016703 012562      MOV      RXDBUF,R3          ;SET UP FOR ERROR MESSAGE
1719 005546 012700 110125      MOV      #RXERR!PARER!125,R0 ;EXPECTED
1720 005552 012767 000011 173360      MOV      #9,SHIFT          ;# OF SHIFTS
1721 005560 012767 000652 173356      MOV      #652,TEMP1        ;DATA CHAR
1722 005566 004767 012236      JSR      PC,RPOKE          ;SHIFT IN THIS CHAR

```



```

1723 005572 105777 012526          TSTB  @RXCSR ;RXDONE ?
1724 005576 100401                    BMI   1$
1725 005600 104000                    HLT   ;RXDONE SHOULD BE ASSERTED
1726 005602                    1$:
1727 005602 017701 012522          MOV   @RXDBUF,R1 ;ACTUAL
1728 005606 020001                    CMP   R0,R1 ;COMPARE EXP VS. ACT
1729 005610 001401                    BEQ   2$
1730 005612 104002                    HLT   2 ;PARITY ERROR &RXERR SHOULD BE SET
1731 005614                    2$:
1732                                ;NOTE THAT THE PARITY BIT SHOULD
1733                                ;SHOW UP IN THE DATA
1734                                ;IE. BIT SIX FOR SIX LEVEL CODE
1735 005614 104400          SCOPE
1736                                ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
1737                                ;;OF THE RECEIVER
1738                                ;;MODE:ISOC (ISYMOD)
1739                                ;;PARITY:ODDPAR
1740                                ;;LENGTH:SEVEN PLUS PARITY
1741                                ;;CHAR: 125
1742                                ;;
1743 005616 012767 000013 173302 TST11: MOV   #11,TSTNO ;SAVE THIS
1744 005624 012767 006002 173264      MOV   #TST12,NEXT ;GO TO THIS TEST WHEN THRU
1745 005632 052777 000400 012500      BIS   #MRESET,@TXCSR ;MASTER RESET
1746 005640 012777 000000 012466      MOV   #ISYMOD,@PARCSR ;SET THE MODE
1747 005646 052777 000400 012464      BIS   #MRESET,@TXCSR ;MASTER RESET
1748
1749                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1750 005654 012777 064001 012456      MOV   #MTDATA!CLK!MINT!BREAK,@TXCSR
1751
1752                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1753 005662 012777 005000 012444      MOV   #ISYMOD!SEVEN!ODDPAR!0,@PARCSR
1754 005670 052777 000020 012426      BIS   #SYNSCH,@RXCSR ;SET SYNC SEARCH
1755                                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1756 005676 042777 020000 012434      BIC   #CLK,@TXCSR ;POKE CLK DOWN
1757 005704 052777 020000 012426      BIS   #CLK,@TXCSR ;POKE CLK UP
1758                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1759 005712 042777 020000 012420      BIC   #CLK,@TXCSR ;POKE CLK DOWN
1760 005720 052777 020000 012412      BIS   #CLK,@TXCSR ;POKE CLK UP
1761 005726 016703 012376          MOV   RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1762 005732 012700 110125          MOV   #RXERR!PARER!125,R0 ;EXPECTED
1763 005736 012767 000012 173174      MOV   #10,SHIFT ;# OF SHIFTS
1764 005744 012767 001252 173172      MOV   #1252,TEMP1 ;DATA CHAR
1765 005752 004767 012052          JSR   PC,RPOKE ;SHIFT IN THIS CHAR
1766 005756 105777 012342          TSTB  @RXCSR ;RXDONE ?
1767 005762 100401                    BMI   1$
1768 005764 104000                    HLT   ;RXDONE SHGULD BE ASSERTED
1769 005766                    1$:
1770 005766 017701 012336          MOV   @RXDBUF,R1 ;ACTUAL
1771 005772 020001                    CMP   R0,R1 ;COMPARE EXP VS. ACT
1772 005774 001401                    BEQ   2$
1773 005776 104002                    HLT   2 ;PARITY ERROR &RXERR SHOULD BE SET
1774 006000                    2$:
1775                                ;NOTE THAT THE PARITY BIT SHOULD
1776                                ;SHOW UP IN THE DATA
1777                                ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
1778 006000 104400          SCOPE

```

```

1779                                     ::THIS TEST VERIFYS ODDPAR PARITY SENSE
1780                                     ::OF THE RECEIVER
1781                                     ::MODE:ISOC (ISYMOD)
1782                                     ::PARITY:ODDPAR
1783                                     ::LENGTH:EIGHT PLUS PARITY
1784                                     ::CHAR: 125
1785                                     ::
1786 006002 012767 000014 173116 TST12: MOV #12,TSTNO ;SAVE THIS
1787 006010 012767 006166 173100 MOV #TST13,NEXT ;GO TO THIS TEST WHEN THRU
1788 006016 052777 000400 012314 BIS #MRESET,@TXCSR ;MASTER R-SET
1789 006024 012777 000000 012302 MOV #ISYMOD,@PARCSR ;SET THE MODE
1790 006032 052777 000400 012300 BIS #MRESET,@TXCSR ;MASTER RESET
1791
1792                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1793 006040 012777 064001 012272 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1794
1795                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1796 006046 012777 007000 012260 MOV #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
1797 006054 052777 000020 012242 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1798                                     ;POKE CLK TO GET RECEIVER INTO SYNCKOIZATION....
1799 006062 042777 020000 012250 BIC #CLK,@TXCSR ;POKE CLK DOWN
1800 006070 052777 020000 012242 BIS #CLK,@TXCSR ;POKE CLK UP
1801                                     ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1802 006076 042777 020000 012234 BIC #CLK,@TXCSR ;POKE CLK DOWN
1803 006104 052777 020000 012226 BIS #CLK,@TXCSR ;POKE CLK UP
1804 006112 016703 012212 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1805 006116 012700 110125 MOV #RXERR!PARER!125,R0 ;EXPECTED
1806 006122 012767 000013 173010 MOV #11,SHIFT ;# OF SHIFTS
1807 006130 012767 002252 173006 MOV #2252,TEMP1 ;DATA CHAR
1808 006136 004767 011666 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1809 006142 105777 012156 TSTB @RXCSR ;RXDONE ?
1810 006146 100401 BMI 1$
1811 006150 104000 HLT ;RXDONE SHOULD BE ASSERTED
1812 006152 1$:
1813 006152 017701 012152 MOV @RXDBUF,R1 ;ACTUAL
1814 006156 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1815 006160 001401 BEQ 2$
1816 006162 104002 HLT 2 ;PARITY ERROR &RXERR SHOULD BE SET
1817 006164 2$:
1818 006164 104400
1819
1820                                     SCOPE
1821                                     ::THIS TEST PERFORMS BINARY DATA CHECK ON THE
1822                                     ::RECEIVER
1823                                     ::LENGTH:EIGHT PLUS PARITY
1824                                     ::MODE:ISYMOD
1825                                     ::PARITY:EVEPAR
1826                                     ::
1827 006166 012767 000015 172732 TST13: MOV #13,TSTNO ;SAVE THIS
1828 006174 012767 006366 172714 MOV #TST14,NEXT ;GO TO THIS TEST WHEN THRU
1829 006202 052777 000400 012130 BIS #MRESET,@TXCSR ;MASTER RESET
1830 006210 012777 000000 012116 MOV #ISYMOD,@PARCSR ;SET THE MODE
1831 006216 052777 000400 012114 BIS #MRESET,@TXCSR ;MASTER RESET
1832
1833                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1834 006224 012777 064001 012106 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR

```



```

1835 006232 012777 007400 012074      MOV      #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1836 006240 052777 000020 012056      BIS      #SYNSCH,@RXCSR      ;SET SYNC SEARCH
1837                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1838 006246 042777 020000 012064      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1839 006254 052777 020000 012056      BIS      #CLK,@TXCSR      ;POKE CLK UP
1840                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1841 006262 042777 020000 012050      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1842 006270 052777 020000 012042      BIS      #CLK,@TXCSR      ;POKE CLK UP
1843 006276 016703 012026      MOV      RXDBUF,R3      ;SET UP ERROR MESSAGE
1844 006302 005004                                     CLR      R4      ;DATA CHAR
1845 006304 010400      1$:      MOV      R4,R0      ;EXPECTED
1846 006306 012767 000013 172624      MOV      #11,SHIFT      ;# OF SHIFTS
1847 006314 010467 172624      MOV      R4,TEMP1      ;'TO BE SHIFTED CHARACTER''
1848 006320 004767 011656      JSR      PC,EVEN8      ;CALC PARITY
1849 006324 000241                                     CLC
1850 006326 006167 172612      ROL      TEMP1      ;GENERATE START BIT
1851 006332 052767 002000 172604      BIS      #BIT10,TEMP1      ;GENERATE STOP BIT
1852                                     ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1853 006340 004767 011464      JSR      PC,RPOKE      ;SHIFT IN THIS CHAR
1854 006344 017701 011760      MOV      @RXDBUF,R1      ;ACTUAL
1855 006350 020001      CMP      R0,R1      ;COMPARE EXP VS ACT
1856 006352 001401      BEQ      2$
1857 006354 104002      HLT      2      ;DATA CHARS SHOULD MATCH
1858                                     ;THERE SHOULD BE NO PARITY ERROR
1859      2$:
1860 006356 005204      INC      R4      ;UPGRADE NEXT CHAR
1861 006360 105704      TSTB    R4      ;LAST CHAR ?
1862 006362 001350      BNE     1$
1863 006364 104400      SCOPE
1864                                     ;:THIS TEST PERFORMS BINARY DATA CHECK ON THE
1865                                     ;:RECEIVER
1866                                     ;:LENGTH:EIGHT PLUS PARITY
1867                                     ;:MODE:ISYMOD
1868                                     ;:PARITY:ODDPAR
1869                                     ;:
1870 006366 012767 000016 172532      TST14:  MOV      #14,TSTNO      ;SAVE THIS
1871 006374 012767 006566 172514      MOV      #TST15,NEXT      ;GO TO THIS TEST WHEN THRU
1872 006402 052777 000400 011730      BIS      #MRESET,@TXCSR      ;MASTER RESET
1873 006410 012777 000000 011716      MOV      #ISYMOD,@PARCSR      ;SET THE MODE
1874 006416 052777 000400 011714      BIS      #MRESET,@TXCSR      ;MASTER RESET
1875
1876                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1877 006424 012777 064001 011706      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1878
1879                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1880 006432 012777 007000 011674      MOV      #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
1881 006440 052777 000020 011656      BIS      #SYNSCH,@RXCSR      ;SET SYNC SEARCH
1882                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1883 006446 042777 020000 011664      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1884 006454 052777 020000 011656      BIS      #CLK,@TXCSR      ;POKE CLK UP
1885                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1886 006462 042777 020000 011650      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1887 006470 052777 020000 011642      BIS      #CLK,@TXCSR      ;POKE CLK UP
1888 006476 016703 011626      MOV      RXDBUF,R3      ;SET UP ERROR MESSAGE
1889 006502 005004                                     CLR      R4      ;DATA CHAR
1890 006504 010400      1$:      MOV      R4,R0      ;EXPECTED

```

```

1891 006506 012767 000013 172424      MOV    #11,SHIFT      ;# OF SHIFTS
1892 006514 010467 172424      MOV    R4,TEMP1      ;'TO BE SHIFTED CHARACTER''
1893 006520 004767 011372      JSR    PC,ODD8      ;CALC PARITY
1894 006524 000241                CLC
1895 006526 006167 172412      ROL    TEMP1      ;GENERATE START BIT
1896 006532 052767 002000 172404      BIS    #BIT10,TEMP1  ;GENERATE STOP BIT
1897                ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1898 006540 004767 011264      JSR    PC,RPOKE     ;SHIFT IN THIS CHAR
1899 006544 017701 011560      MOV    @RXDBUF,R1   ;ACTUAL
1900 006550 020001                CMP    R0,R1      ;COMPARE EXP VS ACT
1901 006552 001401                BEQ    2$
1902 006554 104002                HLT    2          ;DATA CHARS SHOULD MATCH
1903                ;THERE SHOULD BE NO PARITY ERROR
1904                2$:
1905 006556 005204                INC    R4          ;UPGRADE NEXT CHAR
1906 006560 105704                TSTB   R4          ;LAST CHAR ?
1907 006562 001350                BNE    1$
1908 006564 104400                SCOPE
1909                ;:THIS TEST PERFORMS BINARY DATA CHECK ON THE
1910                ;:RECEIVER
1911                ;:LENGTH:EIGHT PLUS PARITY
1912                ;:MODE:SYNEXT
1913                ;:PARITY:EVEPAR
1914                ;:
1915 006566 012767 000017 172332      TST15: MOV    #15,TSTNO      ;SAVE THIS
1916 006574 012767 006736 172314      MOV    #TST16,NEXT  ;GO TO THIS TEST WHEN THRU
1917 006602 052777 000400 011530      BIS    #MRESET,@TXCSR ;MASTER RESET
1918 006610 012777 020000 011516      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1919 006616 052777 000400 011514      BIS    #MRESET,@TXCSR ;MASTER RESET
1920                ;
1921                ;:SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1922 006624 012777 064001 011506      MOV    #MMDATA!CLK!MINT!BREAK,@TXCSR
1923                ;
1924                ;:SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1925 006632 012777 027400 011474      MOV    #SYNEXT!EIGHT!EVEPAR!0,@PARCSR
1926 006640 052777 000020 011456      BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
1927                ;:POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1928 006646 042777 020000 011464      BIC    #CLK,@TXCSR  ;POKE CLK DOWN
1929 006654 052777 020000 011456      BIS    #CLK,@TXCSR  ;POKE CLK UP
1930 006662 016703 011442      MOV    RXDBUF,R3    ;SET UP ERROR MESSAGE
1931 006666 005004                CLR    R4          ;DATA CHAR
1932 006670 010400                MOV    R4,R0      ;EXPECTED
1933 006672 012767 000011 172240      MOV    #9,SHIFT     ;# OF SHIFTS
1934 006700 010467 172240      MOV    R4,TEMP1     ;'TO BE SHIFTED CHARACTER''
1935 006704 004767 011272      JSR    PC,EVEN8     ;CALC PARITY
1936                ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1937 006710 004767 011114      JSR    PC,RPOKE     ;SHIFT IN THIS CHAR
1938 006714 017701 011410      MOV    @RXDBUF,R1   ;ACTUAL
1939 006720 020001                CMP    R0,R1      ;COMPARE EXP VS ACT
1940 006722 001401                BEQ    2$
1941 006724 104002                HLT    2          ;DATA CHARS SHOULD MATCH
1942                ;THERE SHOULD BE NO PARITY ERROR
1943                2$:
1944 006726 005204                INC    R4          ;UPGRADE NEXT CHAR
1945 006730 105704                TSTB   R4          ;LAST CHAR ?
1946 006732 001356                BNE    1$

```



```

1947 006734 104400
1948
1949
1950
1951
1952
1953
1954 006736 012767 000020 172162 TST16: MOV #16,TSTNO ;SAVE THIS
1955 006744 012767 007106 172144 MOV #TST17,NEXT ;GO TO THIS TEST WHEN THRU
1956 006752 052777 000400 011360 BIS #MRESET,@TXCSR ;MASTER RESET
1957 006760 012777 020000 011346 MOV #SYNEXT,@PARCSR ;SET THE MODE
1958 006766 052777 000400 011344 BIS #MRESET,@TXCSR ;MASTER RESET
1959
1960 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1961 006774 012777 064001 011336 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1962
1963 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1964 007002 012777 027000 011324 MOV #SYNEXT!EIGHT!ODDPAR!0,@PARCSR
1965 007010 052777 000020 011306 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1966 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1967 007016 042777 020000 011314 BIC #CLK,@TXCSR ;POKE CLK DOWN
1968 007024 052777 020000 011306 BIS #CLK,@TXCSR ;POKE CLK UP
1969 007032 016703 011272 MOV RXDBUF,R3 ;SET UP ERROR MESSAGE
1970 007036 005004 CLR R4 ;DATA CHAR
1971 007040 010400 1$: MOV R4,R0 ;EXPECTED
1972 007042 012767 000011 172070 MOV #9,SHIFT ;# OF SHIFTS
1973 007050 010467 172070 MOV R4,TEMP1 ;'TO BE SHIFTED CHARACTER'
1974 007054 004767 011036 JSR PC,ODD8 ;CALC PARITY
1975 ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1976 007060 004767 010744 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1977 007064 017701 011240 MOV @RXDBUF,R1 ;ACTUAL
1978 007070 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1979 007072 001401 BEQ 2$
1980 007074 104002 HLT 2 ;DATA CHARS SHOULD MATCH
1981 ;THERE SHOULD BE NO PARITY ERROR
1982 007076 2$: INC R4 ;UPGRADE NEXT CHAR
1983 007076 005204 TSTB R4 ;LAST CHAR ?
1984 007100 105704 BNE 1$
1985 007102 001356
1986 007104 104400
1987
1988 SCOPE
1989 ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
1990 ;:OF THE RECEIVER LOGIC
1991 ;:MODE:ISYMOD
1992 ;:LENGTH:FIVE
1993 ;:NOTE: RXDONE SHOULD NEVER ASSERT
1994 ;:CHAR: 26 (SYNC)
1995
1996
1997
1998
1999
2000
2001 007144 012777 064001 011166 TST17: MOV #17,TSTNO ;SAVE THIS
2002 MOV #TST18,NEXT ;GO TO THIS TEST WHEN THRU
        BIS #MRESET,@TXCSR ;MASTER RESET
        MOV #ISYMOD,@PARCSR ;SET THE MODE
        BIS #MRESET,@TXCSR ;MASTER RESET
        ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
        MOV #i?DATA!CLK!MINT!BREAK,@TXCSR

```

```

2003      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2004 007152 012777 000026 011154      MOV      #ISYMOD!FIVE!NOPAR!26,@PARCSR
2005 007160 052777 000020 011136      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2006      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2007 007166 042777 020000 011144      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2008 007174 052777 020000 011136      BIS      #CLK,@TXCSR ;POKE CLK UP
2009      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2010 007202 042777 020000 011130      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2011 007210 052777 020000 011122      BIS      #CLK,@TXCSR ;POKE CLK UP
2012 007216 052777 000400 011100      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2013 007224 012767 000003 171710      MOV      #3,COUNT ;# OF SYNC CHARS
2014 007232 012767 000154 171704      1$:     MOV      #154,TEMP1 ;CHAR TO BE SHIFTED
2015 007240 012767 000007 171672      MOV      #7,SHIFT ;# OF SHIFTS
2016 007246 004767 010556      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2017 007252 105777 011046      TSTB    @RXCSR ;RXDONE ?
2018 007256 100001      BPL     2$
2019 007260 104000      HLT     ;RXDONE SHOULD NOT BE ASSERTED
2020 007262
2021 007262 005367 171654      2$:     DEC     COUNT ;# OF SYNC CHARS
2022 007266 001361      BNE     1$
2023 007270 104400      SCOPE
2024      ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
2025      ;;OF THE RECEIVER LOGIC
2026      ;;MODE:ISYMOD
2027      ;;LENGTH:SIX
2028      ;;NOTE: RXDONE SHOULD NEVER ASSERT
2029      ;;CHAR: 26 (SYNC)
2030
2031 007272 012767 000022 171626      TST18:  MOV      #18,TSTNO ;SAVE THIS
2032 007300 012767 007456 171610      MOV      #TST19,NEXT ;GO TO THIS TEST WHEN THRU
2033 007306 052777 000400 011024      BIS      #MRESET,@TXCSR ;MASTER RESET
2034 007314 012777 000000 011012      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2035 007322 052777 000400 011010      BIS      #MRESET,@TXCSR ;MASTER RESET
2036
2037      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2038 007330 012777 064001 011002      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2039
2040      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2041 007336 012777 002026 010770      MOV      #ISYMOD!SIX!NOPAR!26,@PARCSR
2042 007344 052777 000020 010752      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2043      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2044 007352 042777 020000 010760      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2045 007360 052777 020000 010752      BIS      #CLK,@TXCSR ;POKE CLK UP
2046      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2047 007366 042777 020000 010744      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2048 007374 052777 020000 010736      BIS      #CLK,@TXCSR ;POKE CLK UP
2049 007402 052777 000400 010714      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2050 007410 012767 000003 171524      MOV      #3,COUNT ;# OF SYNC CHARS
2051 007416 012767 000254 171520      1$:     MOV      #254,TEMP1 ;CHAR TO BE SHIFTED
2052 007424 012767 000010 171506      MOV      #8,SHIFT ;# OF SHIFTS
2053 007432 004767 010372      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2054 007436 105777 010662      TSTB    @RXCSR ;RXDONE ?
2055 007442 100001      BPL     2$
2056 007444 104000      HLT     ;RXDONE SHOULD NOT BE ASSERTED
2057 007446
2058 007446 005367 171470      2$:     DEC     COUNT ;# OF SYNC CHARS

```



```
2059 007452 001361          BNE      1$
2060 007454 104400          SCOPE
2061                          ::THIS TEST CHECKS THE STRIP SYNC FUNCTION
2062                          ::OF THE RECEIVER LOGIC
2063                          ::MODE:ISYMOD
2064                          ::LENGTH:SEVEN
2065                          ::NOTE: RXDONE SHOULD NEVER ASSERT
2066                          ::CHAR: 26 (SYNC)
2067                          ::
2068 007456 012767 000023 171442 TST19:  MOV     #19,TSTNO      ;SAVE THIS
2069 007464 012767 007642 171424      MOV     #TST20,NEXT      ;GO TO THIS TEST WHEN THRU
2070 007472 052777 000400 010640      BIS     #MRESET,@TXCSR  ;MASTER RESET
2071 007500 012777 000000 010626      MOV     #ISYMOD,@PARCSR ;SET THE MODE
2072 007506 052777 000400 010624      BIS     #MRESET,@TXCSR  ;MASTER RESET
2073
2074                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2075 007514 012777 064001 010616      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2076
2077                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2078 007522 012777 004026 010604      MOV     #ISYMOD!SEVEN!NOPAR!26,@PARCSR
2079 007530 052777 000020 010566      BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2080                          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2081 007536 042777 020000 010574      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2082 007544 052777 020000 010566      BIS     #CLK,@TXCSR     ;POKE CLK UP
2083                          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2084 007552 042777 020000 010560      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
2085 007560 052777 020000 010552      BIS     #CLK,@TXCSR     ;POKE CLK UP
2086 007566 052777 000400 010530      BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
2087 007574 012767 000003 171340      MOV     #3,COUNT        ;# OF SYNC CHARS
2088 007602 012767 000454 171334 1$:  MOV     #454,TEMP1      ;CHAR TO BE SHIFTED
2089 007610 012767 000011 171322      MOV     #9,SHIFT        ;# OF SHIFTS
2090 007616 004767 010206          JSR     PC,RPOKE        ;SHIFT IN THIS CHAR
2091 007622 105777 010476          TSTB   @RXCSR ;RXDONE ?
2092 007626 100001          BPL     2$
2093 007630 104000          HLT
2094 007632          2$:
2095 007632 005367 171304          DEC     COUNT ;# OF SYNC CHARS
2096 007636 001361          BNE     1$
2097 007640 104400          SCOPE
2098                          ::THIS TEST CHECKS THE STRIP SYNC FUNCTION
2099                          ::OF THE RECEIVER LOGIC
2100                          ::MODE:ISYMOD
2101                          ::LENGTH:EIGHT
2102                          ::NOTE: RXDONE SHOULD NEVER ASSERT
2103                          ::CHAR: 26 (SYNC)
2104                          ::
2105 007642 012767 000024 171256 TST20:  MOV     #20,TSTNO      ;SAVE THIS
2106 007650 012767 010026 171240      MOV     #TST21,NEXT      ;GO TO THIS TEST WHEN THRU
2107 007656 052777 000400 010454      BIS     #MRESET,@TXCSR  ;MASTER RESET
2108 007664 012777 000000 010442      MOV     #ISYMOD,@PARCSR ;SET THE MODE
2109 007672 052777 000400 010440      BIS     #MRESET,@TXCSR  ;MASTER RESET
2110
2111                          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2112 007700 012777 064001 010432      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2113
2114                          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
```



```

2171                                     ::MODE:SYNEXT
2172                                     ::LENGTH:SIX
2173                                     ::NOTE: RXDONE SHOULD NEVER ASSERT
2174                                     ::CHAR: 26 (SYNC)
2175                                     ::
2176 010176 012767 000026 170722 TST22: MOV #22,TSTNO ;SAVE THIS
2177 010204 012767 010346 170704 MOV #TST23,NEXT ;GO TO THIS TEST WHEN THRU
2178 010212 052777 000400 010120 BIS #MRESET,@TXCSR ;MASTER RESET
2179 010220 012777 020000 010106 MOV #SYNEXT,@PARCSR ;SET THE MODE
2180 010226 052777 000400 010104 BIS #MRESET,@TXCSR ;MASTER RESET
2181
2182 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2183 010234 012777 064001 010076 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2184
2185 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2186 010242 012777 022026 010064 MOV #SYNEXT!SIX!NOPAR!26,@PARCSR
2187 010250 052777 000020 010046 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2188 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2189 010256 042777 020000 010054 BIC #CLK,@TXCSR ;POKE CLK DOWN
2190 010264 052777 020000 010046 BIS #CLK,@TXCSR ;POKE CLK UP
2191 010272 052777 000400 010024 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2192 010300 012767 000003 170634 MOV #3,COUNT ;# OF SYNC CHARS
2193 010306 012767 000026 170630 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2194 010314 012767 000006 170616 MOV #6,SHIFT ;# OF SHIFTS
2195 010322 004767 007502 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2196 010326 105777 007772 TSTB @RXCSR ;RXDONE ?
2197 010332 100001 BPL 2$
2198 010334 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2199 010336
2200 010336 005367 170600 2$: DEC COUNT ;# OF SYNC CHARS
2201 010342 001361 BNE 1$
2202 010344 104400 SCOPE
2203 ::THIS TEST CHECKS THE STRIP SYNC FUNCTION
2204 ::OF THE RECEIVER LOGIC
2205 ::MODE:SYNEXT
2206 ::LENGTH:SEVEN
2207 ::NOTE: RXDONE SHOULD NEVER ASSERT
2208 ::CHAR: 26 (SYNC)
2209 ::
2210 010346 012767 000027 170552 TST23: MOV #23,TSTNO ;SAVE THIS
2211 010354 012767 010516 170534 MOV #TST24,NEXT ;GO TO THIS TEST WHEN THRU
2212 010362 052777 000400 007750 BIS #MRESET,@TXCSR ;MASTER RESET
2213 010370 012777 020000 007736 MOV #SYNEXT,@PARCSR ;SET THE MODE
2214 010376 052777 000400 007734 BIS #MRESET,@TXCSR ;MASTER RESET
2215
2216 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2217 010404 012777 064001 007726 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2218
2219 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2220 010412 012777 024026 007714 MOV #SYNEXT!SEVEN!NOPAR!26,@PARCSR
2221 010420 052777 000020 007676 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2222 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2223 010426 042777 020000 007704 BIC #CLK,@TXCSR ;POKE CLK DOWN
2224 010434 052777 020000 007676 BIS #CLK,@TXCSR ;POKE CLK UP
2225 010442 052777 000400 007654 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2226 010450 012767 000003 170464 MOV #3,COUNT ;# OF SYNC CHARS

```

```
2227 010456 012767 000026 170460 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2228 010464 012767 000007 170446 MOV #7,SHIFT ;# OF SHIFTS
2229 010472 004767 007332 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2230 010476 105777 007622 TSTB @RXCSR ;RXDONE ?
2231 010502 100001 BPL 2$
2232 010504 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2233 010506 2$:
2234 010506 005367 170430 DEC COUNT ;# OF SYNC CHARS
2235 010512 001361 BNE 1$
2236 010514 104400 SCOPE
2237 ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2238 ;:OF THE RECEIVER LOGIC
2239 ;:MODE:SYNEXT
2240 ;:LENGTH:EIGHT
2241 ;:NOTE: RXDONE SHOULD NEVER ASSERT
2242 ;:CHAR: 26 (SYNC)
2243 ;:
2244 010516 012767 000030 170402 TST24: MOV #24,TSTNO ;SAVE THIS
2245 010524 012767 010666 170364 MOV #TST25,NEXT ;GO TO THIS TEST WHEN THRU
2246 010532 052777 000400 007600 BIS #MRESET,@TXCSR ;MASTER RESET
2247 010540 012777 020000 007566 MOV #SYNEXT,@PARCSR ;SET THE MODE
2248 010546 052777 000400 007564 BIS #MRESET,@TXCSR ;MASTER RESET
2249 ;
2250 ;:SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2251 010554 012777 064001 007556 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2252 ;
2253 ;:SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2254 010562 012777 026026 007544 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2255 010570 052777 000020 007526 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2256 ;:POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2257 010576 042777 020000 007534 BIC #CLK,@TXCSR ;POKE CLK DOWN
2258 010604 052777 020000 007526 BIS #CLK,@TXCSR ;POKE CLK UP
2259 010612 052777 000400 007504 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2260 010620 012767 000003 170314 MOV #3,COUNT ;# OF SYNC CHARS
2261 010626 012767 000026 170310 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2262 010634 012767 000010 170276 MOV #8,SHIFT ;# OF SHIFTS
2263 010642 004767 007162 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2264 010646 105777 007452 TSTB @RXCSR ;RXDONE ?
2265 010652 100001 BPL 2$
2266 010654 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2267 010656 2$:
2268 010656 005367 170260 DEC COUNT ;# OF SYNC CHARS
2269 010662 001361 BNE 1$
2270 010664 104400 SCOPE
2271 ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2272 ;:OF THE RECEIVER LOGIC
2273 ;:MODE:SYNINT
2274 ;:LENGTH:FIVE
2275 ;:NOTE: RXDONE SHOULD NEVER ASSERT
2276 ;:CHAR: 26 (SYNC)
2277 ;:
2278 010666 012767 000031 170232 TST25: MOV #25,TSTNO ;SAVE THIS
2279 010674 012767 011052 170214 MOV #TST26,NEXT ;GO TO THIS TEST WHEN THRU
2280 010702 052777 000400 007430 BIS #MRESET,@TXCSR ;MASTER RESET
2281 010710 012777 030000 007416 MOV #SYNINT,@PARCSR ;SET THE MODE
2282 010716 052777 000400 007414 BIS #MRESET,@TXCSR ;MASTER RESET
```



```

2283
2284 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2285 010724 012777 064001 007406 MOV #MMDATA!CLK!MINT!BREAK,@TXCSR
2286
2287 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2288 010732 012777 030026 007374 MOV #SYNINT!FIVE!NOPAR!26,@PARCSR
2289 010740 052777 000020 007356 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2290 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2291 010746 042777 020000 007364 BIC #CLK,@TXCSR ;POKE CLK DOWN
2292 010754 052777 020000 007356 BIS #CLK,@TXCSR ;POKE CLK UP
2293 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2294 010762 042777 020000 007350 BIC #CLK,@TXCSR ;POKE CLK DOWN
2295 010770 052777 020000 007342 BIS #CLK,@TXCSR ;POKE CLK UP
2296 010776 052777 000400 007320 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2297 011004 012767 000003 170130 MOV #3,COUNT ;# OF SYNC CHARS
2298 011012 012767 000026 170124 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2299 011020 012767 000005 170112 MOV #5,SHIFT ;# OF SHIFTS
2300 011026 004767 006776 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2301 011032 105777 007266 TSTB @RXCSR ;RXDONE ?
2302 011036 100001 BPL 2$
2303 011040 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2304 011042
2305 011042 005367 170074 2$: DEC COUNT ;# OF SYNC CHARS
2306 011046 001361 BNE 1$
2307 011050 104400 SCOPE
2308 ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2309 ;:OF THE RECEIVER LOGIC
2310 ;:MODE:SYNINT
2311 ;:LENGTH:SIX
2312 ;:NOTE: RXDONE SHOULD NEVER ASSERT
2313 ;:CHAR: 26 (SYNC)
2314 ;:
2315 011052 012767 000032 170046 TST26: MOV #26,TSTNO ;SAVE THIS
2316 011060 012767 011236 170030 MOV #TST27,NEXT ;GO TO THIS TEST WHEN THRU
2317 011066 052777 000400 007244 BIS #MRESET,@TXCSR ;MASTER RESET
2318 011074 012777 030000 007232 MOV #SYNINT,@PARCSR ;SET THE MODE
2319 011102 052777 000400 007230 BIS #MRESET,@TXCSR ;MASTER RESET
2320
2321 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2322 011110 012777 064001 007222 MOV #MMDATA!CLK!MINT!BREAK,@TXCSR
2323
2324 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2325 011116 012777 032026 007210 MOV #SYNINT!SIX!NOPAR!26,@PARCSR
2326 011124 052777 000020 007172 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2327 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2328 011132 042777 020000 007200 BIC #CLK,@TXCSR ;PGKE CLK DOWN
2329 011140 052777 020000 007172 BIS #CLK,@TXCSR ;POKE CLK UP
2330 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2331 011146 042777 020000 007164 BIC #CLK,@TXCSR ;POKE CLK DOWN
2332 011154 052777 020000 007156 BIS #CLK,@TXCSR ;POKE CLK UP
2333 011162 052777 000400 007134 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2334 011170 012767 000003 167744 MOV #3,COUNT ;# OF SYNC CHARS
2335 011176 012767 000026 167740 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2336 011204 012767 000006 167726 MOV #6,SHIFT ;# OF SHIFTS
2337 011212 004767 006612 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2338 011216 105777 007102 TSTB @RXCSR ;RXDONE ?

```

```
2339 011222 100001          BPL      2$
2340 011224 104000          HLT      ;RXDONE SHOULD NOT BE ASSERTED
2341 011226                2$:
2342 011226 005367 167710     DEC      COUNT    ;# OF SYNC CHARS
2343 011232 001361          BNE      1$
2344 011234 104400          SCOPE
2345                ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2346                ;:OF THE RECEIVER LOGIC
2347                ;:MODE:SYNINT
2348                ;:LENGTH:SEVEN
2349                ;:NOTE: RXDONE SHOULD NEVER ASSERT
2350                ;:CHAR: 26 (SYNC)
2351                ;:
2352 011236 012767 000033 167662 TST27:  MOV     #27,TSTNO      ;SAVE THIS
2353 011244 012767 011422 167644     MOV     #TST28,NEXT    ;GO TO THIS TEST WHEN THRU
2354 011252 052777 000400 007060     BIS     #MRESET,@TXCSR ;MASTER RESET
2355 011260 012777 030000 007046     MOV     #SYNINT,@PARCSR ;SET THE MODE
2356 011266 052777 000400 007044     BIS     #MRESET,@TXCSR ;MASTER RESET
2357
2358                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2359 011274 012777 064001 007036     MOV     #MMDATA!CLK!MINT!BREAK,@TXCSR
2360
2361                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2362 011302 012777 034026 007024     MOV     #SYNINT!SEVEN!NOPAR!26,@PARCSR
2363 011310 052777 000020 007006     BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2364                ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2365 011316 042777 020000 007014     BIC     #CLK,@TXCSR    ;POKE CLK DOWN
2366 011324 052777 020000 007006     BIS     #CLK,@TXCSR    ;POKE CLK UP
2367
2368                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2369 011332 042777 020000 007000     BIC     #CLK,@TXCSR    ;POKE CLK DOWN
2370 011340 052777 020000 006772     BIS     #CLK,@TXCSR    ;POKE CLK UP
2371 011346 052777 000400 006750     BIS     #STPSYN,@RXCSR ;SET STRIP SYNC
2372 011354 012767 000003 167560     MOV     #3,COUNT      ;# OF SYNC CHARS
2373 011362 012767 000026 167554     1$:    MOV     #26,TEMP1     ;CHAR TO BE SHIFTED
2374 011370 012767 000007 167542     MOV     #7,SHIFT      ;# OF SHIFTS
2375 011376 004767 006426          JSR     PC,RPOKE      ;SHIFT IN THIS CHAR
2376 011402 105777 006716          TSTB   @RXCSR ;RXDONE ?
2377 011406 100001          BPL      2$
2378 011410 104000          HLT      ;RXDONE SHOULD NOT BE ASSERTED
2379 011412                2$:
2380 011412 005367 167524     DEC      COUNT    ;# OF SYNC CHARS
2381 011416 001361          BNE      1$
2382 011420 104400          SCOPE
2383                ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2384                ;:OF THE RECEIVER LOGIC
2385                ;:MODE:SYNINT
2386                ;:LENGTH:EIGHT
2387                ;:NOTE: RXDONE SHOULD NEVER ASSERT
2388                ;:CHAR: 26 (SYNC)
2389                ;:
2389 011422 012767 000034 167476 TST28:  MOV     #28,TSTNO      ;SAVE THIS
2390 011430 012767 011606 167460     MOV     #TST29,NEXT    ;GO TO THIS TEST WHEN THRU
2391 011436 052777 000400 006674     BIS     #MRESET,@TXCSR ;MASTER RESET
2392 011444 012777 030000 006662     MOV     #SYNINT,@PARCSR ;SET THE MODE
2393 011452 052777 000400 006660     BIS     #MRESET,@TXCSR ;MASTER RESET
2394
```



```
2395 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2396 011460 012777 064001 006652 MOV #MCDATA!CLK!MINT!BREAK,@TXCSR
2397
2398 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2399 011466 012777 036026 006640 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2400 011474 052777 000020 006622 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2401 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2402 011502 042777 020000 006630 BIC #CLK,@TXCSR ;POKE CLK DOWN
2403 011510 052777 020000 006622 BIS #CLK,@TXCSR ;POKE CLK UP
2404 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2405 011516 042777 020000 006614 BIC #CLK,@TXCSR ;POKE CLK DOWN
2406 011524 052777 020000 006606 BIS #CLK,@TXCSR ;POKE CLK UP
2407 011532 052777 000400 006564 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2408 011540 012767 000003 167374 MOV #3,COUNT ;# OF SYNC CHARS
2409 011546 012767 000026 167370 1$: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2410 011554 012767 000010 167356 MOV #8.,SHIFT ;# OF SHIFTS
2411 011562 004767 006242 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2412 011566 105777 006532 TSTB @RXCSR ;RXDONE ?
2413 011572 100001 BPL 2$
2414 011574 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
2415 011576
2416 011576 005367 167340 2$: DEC COUNT ;# OF SYNC CHARS
2417 011602 001361 BNE 1$
2418 011604 104400 SCOPE
2419 ;;THIS TEST PROVES THAT RXERR FREEZES THE 'RECEIVER RESET'
2420 ;;WHILE IN STRIP SYNC MODE
2421 ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2422 ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2423 ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2424 ;;IS DISCOMBOBULATED
2425 ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2426 ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE &ERROR FLAGS
2427 ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
2428 ;;BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
2429 ;;MODE: ISOC (ISYMOD)
2430 ;;LENGTH: EIGHT
2431 ;;PARITY: EVEPAR
2432 ;;CHARACTER EXPECTED:26
2433 ;;CHARACTER SENT: SYNC CHARACTER
2434 ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2435 ;;
2436 011606 012767 000035 167312 TST29: MOV #29,TSTNO ;SAVE THIS
2437 011614 012767 012210 167274 MOV #TST30,NEXT ;GO TO THIS TEST WHEN THRU
2438 011622 052777 000400 006510 BIS #MRESET,@TXCSR ;MASTER RESET
2439 011630 012777 000000 006476 MOV #ISYMOD,@PARCSR ;SET THE MODE
2440 011636 052777 000400 006474 BIS #MRESET,@TXCSR ;MASTER RESET
2441
2442 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2443 011644 012777 064001 006466 MOV #MCDATA!CLK!MINT!BREAK,@TXCSR
2444
2445 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2446 011652 012777 007426 006454 MOV #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
2447 011660 016703 006444 MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2448 011664 012767 000003 167250 MOV #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
2449 011672 052777 000020 006424 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2450 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
```

```

2451 011700 042777 020000 006432      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2452 011706 052777 020000 006424      BIS    #CLK,@TXCSR      ;POKE CLK UP
2453                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2454 011714 042777 020000 006416      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2455 011722 052777 020000 006410      BIS    #CLK,@TXCSR      ;POKE CLK UP
2456 011730 052777 000400 006366      BIS    #STPSYN,@RXCSR   ;SET STRIP SYNC
2457 011736 012767 000013 167174      2$:   MOV    #11,SHIFT    ;# OF SHIFTS
2458 011744 012767 003054 167172      MOV    #3054,TEMP1     ;SYNC CHAR + START&STOP+ PARITY
2459 011752 004767 006052          1$:   JSR    PC,RPOKE        ;SHIFT IN THIS CHARACTER
2460 011756 105777 006342          TSTB   @RXCSR ;RXDONE = 0 ?
2461 011762 100001          BPL    64$
2462 011764 104000          HLT    ;RXDONE SHOULD NOT BE SET
2463 011766          64$:
2464 011766 005367 167150          DEC    COUNT           ;# OF SYNC CHARS
2465 011772 001361          BNE    2$              ;GO AGAIN ?
2466 011774 012700 000026          MOV    #26,R0          ;EXPECTED
2467 012000 017701 006324          MOV    @RXDBUF,R1     ;ACTUAL
2468                                     ;NOTE THAT THIS IS THE FIRST TIME
2469                                     ;RXDBUF IS READ.....THERE SHOULD BE
2470                                     ;NO OVER RUN ERRORS
2471 012004 020001          CMP    R0,R1          ;COMPARE EXPECTED VS ACTUAL
2472 012006 001401          BEQ    65$
2473 012010 104002          HLT    2              ;DATA CHARS SHOULD COMPARE
2474                                     ;THERE SHOULD BE NO RXERR'S
2475 012012          65$:
2476 012012 012767 000004 167122      MOV    #4,COUNT        ;# OF TIMES
2477 012020 012700 110026          MOV    #RXERR!PARER!26,R0 ;EXPECTED
2478 012024 012767 002054 167112      MOV    #2054,TEMP1    ;BAD SYNC CHAR (WRONG PARITY)
2479 012032 012767 000013 167100      3$:   MOV    #11,SHIFT    ;# OF SHIFTS
2480 012040 004767 005764          JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2481 012044 105777 006254          TSTB   @RXCSR ;RXDONE = 1?
2482 012050 100401          BMI    66$
2483 012052 104000          HLT    ;RXDONE SHOULD BE SET
2484 012054          66$:
2485 012054 017701 006250          MOV    @RXDBUF,R1     ;ACTUAL DATA
2486 012060 020001          CMP    R0,R1          ;COMPARE EXP VS ACT
2487 012062 001401          BEQ    67$
2488 012064 104002          HLT    2              ;DID THE RESPECTIVE ERROR STOP THE
2489                                     ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2490                                     ;.....CHECK THIS.....
2491 012066          67$:
2492 012066 005367 167050          DEC    COUNT           ;# OF SYNC CHARS
2493 012072 001445          BEQ    5$              ;FINISHED ? GET OUT OF TEST
2494 012074 022767 000003 167040      CMP    #3,COUNT        ;# OF SYNC CHARS
2495 012102 001423          BEQ    6$              ;CHECK FRAME ERROR ?
2496 012104 022767 000002 167030      CMP    #2,COUNT        ;# OF SYNC CHARS
2497 012112 001426          BEQ    7$              ;CHECK FRAME ERROR & BAD PARITY ?
2498                                     ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2499 012114 012767 000013 167016      MOV    #11,SHIFT    ;# OF SHIFTS
2500 012122 012767 000054 167014      MOV    #54,TEMP1     ;FRAME & PARITY ERROR
2501 012130 004767 005674          JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2502                                     ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2503 012134 012767 000054 167002      MOV    #54,TEMP1     ;FRAME & PARITY ERROR
2504 012142 012700 170026          MOV    #RXERR!OVRRUN!FRMERR!PARER!26,R0 ;EXPECTED
2505 012146 000167 177660          JMP    3$              ;DO IT AGAIN
2506 012152 012767 001054 166764      6$:   MOV    #1054,TEMP1    ;BAD STOP BIT FOR FRAME ERROR

```



```

2507 012160 012700 120026      MOV      #RXERR!FRMERR!26,R0      ;EXPECTED
2508 012164 000167 177642      JMP      3$      ;DO IT AGAIN
2509 012170 012767 000054 166746 7$:  MOV      #54,TEMP1      ;BAD STOP BIT & PARITY
2510 012176 012700 130026      MOV      #RXERR!FRMERR!PARER!26,R0 ;EXPECTED
2511 012202 000167 177624      JMP      3$      ;DO IT AGAIN
2512 012206 104400      5$:      SCOPE
2513      ;:THIS TEST PROVES THAT RXERR FREEZES THE 'RECEIVER RESET'
2514      ;:WHILE IN STRIP SYNC MODE
2515      ;:THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2516      ;:STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2517      ;:.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2518      ;:IS DISCOMBOBULATED
2519      ;:IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2520      ;:NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2521      ;:PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
2522      ;:BUT, IF AN RXERR OCCURS  RXDONE PLUS RXERR ARE ASSERTED
2523      ;:MODE: ISOC (ISYMOD)
2524      ;:LENGTH: SEVEN
2525      ;:PARITY: EVEPAR
2526      ;:CHARACTER EXPECTED:226
2527      ;:NOTE THAT THE PARITY BIT SHOULD SHOW
2528      ;:UP IN THE DATA  IE. BIT SEVEN FOR
2529      ;:SEVEN LEVEL CODE
2530      ;:CHARACTER SENT: SYNC CHARACTER
2531      ;:NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2532      ;:
2533 012210 012767 000036 166710  TST30:  MOV      #30,TSTNO      ;SAVE THIS
2534 012216 012767 012612 166672      MOV      #TST31,NEXT      ;GO TO THIS TEST WHEN THRU
2535 012224 052777 000400 006106      BIS      #MRESET,@TXCSR ;MASTER RESET
2536 012232 012777 000000 006074      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2537 012240 052777 000400 006072      BIS      #MRESET,@TXCSR ;MASTER RESET
2538
2539      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2540 012246 012777 064001 006064      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2541
2542      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2543 012254 012777 005626 006052      MOV      #ISYMOD!SEVEN!EVEPAR!226,@PARCSR
2544 012262 016703 006042      MOV      RXDBUF,R3      ;SET UP FOR ERROR MSG
2545 012266 012767 000003 166646      MOV      #3,COUNT      ;# OF TIMES SYNC CHAR WILL BE SENT
2546 012274 052777 000020 006022      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2547      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2548 012302 042777 020000 006030      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2549 012310 052777 020000 006022      BIS      #CLK,@TXCSR ;POKE CLK UP
2550      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2551 012316 042777 020000 006014      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2552 012324 052777 020000 006006      BIS      #CLK,@TXCSR ;POKE CLK UP
2553 012332 052777 000400 005764      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2554 012340 012767 000012 166572 2$:  MOV      #10,SHIFT      ;# OF SHIFTS
2555 012346 012767 001454 166570      MOV      #1454,TEMP1 ;SYNC CHAR + START&STOP+ PARITY
2556 012354 004767 005450 1$:  JSR      PC,RPOKE      ;SHIFT IN THIS CHARACTER
2557 012360 105777 005740      TSTB    @RXCSR ;RXDONE = 0 ?
2558 012364 100001      BPL     64$
2559 012366 104000      HLT
2560 012370      64$:
2561 012370 005367 166546      DEC     COUNT ;# OF SYNC CHARS
2562 012374 001361      BNE    2$ ;GO AGAIN ?

```

```

2563 012376 012700 000226      MOV    #226,R0 ;EXPECTED
2564 012402 017701 005722      MOV    @RXDBUF,R1 ;ACTUAL
2565                                ;NOTE THAT THIS IS THE FIRST TIME
2566                                ;RXDBUF IS READ.....THERE SHOULD BE
2567                                ;NO OVER RUN ERRORS
2568 012406 020001      CMP    R0,R1 ;COMPARE EXPECTED VS ACTUAL
2569 012410 001401      BEQ    65$
2570 012412 104002      HLT    2 ;DATA CHARS SHOULD COMPARE
2571                                ;THERE SHOULD BE NO RXERR'S
2572 012414                                65$:
2573 012414 012767 000004 166520      MOV    #4,COUNT ;# OF TIMES
2574 012422 012700 110026      MOV    #RXERR!PARER!26,R0 ;EXPECTED
2575 012426 012767 001054 166510      MOV    #1054,TEMP1 ;BAD SYNC CHAR (WRONG PARITY)
2576 012434 012767 000012 166476 3$:      MOV    #10,,SHIFT ;# OF SHIFTS
2577 012442 004767 005362      JSR    PC,RPOKE ;SHIFT IN THIS CHAR
2578 012446 105777 005652      TSTB  @RXCSR ;RXDONE = 1?
2579 012452 100401      BMI    66$
2580 012454 104000      HLT    ;RXDONE SHOULD BE SET
2581 012456                                66$:
2582 012456 017701 005646      MOV    @RXDBUF,R1 ;ACTUAL DATA
2583 012462 020001      CMP    R0,R1 ;COMPARE EXP VS ACT
2584 012464 001401      BEQ    67$
2585 012466 104002      HLT    2 ;DID THE RESPECTIVE ERROR STOP THE
2586                                ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2587                                ;.....CHECK THIS.....
2588 012470                                67$:
2589                                ;NOTE THAT THE PARITY BIT SHOULD
2590                                ;SHOW UP IN THE DATA
2591                                ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
2592 012470 005367 166446      DEC    COUNT ;# OF SYNC CHARS
2593 012474 001445      BEQ    5$ ;FINISHED ? GET OUT OF TEST
2594 012476 022767 000003 166436      CMP    #3,COUNT ;# OF SYNC CHARS
2595 012504 001423      BEQ    6$ ;CHECK FRAME ERROR ?
2596 012506 022767 000002 166426      CMP    #2,COUNT ;# OF SYNC CHARS
2597 012514 001426      BEQ    7$ ;CHECK FRAME ERROR & BAD PARITY ?
2598                                ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2599 012516 012767 000012 166414      MOV    #10,,SHIFT ;# OF SHIFTS
2600 012524 012767 000054 166412      MOV    #54,TEMP1 ;FRAME & PARITY ERROR
2601 012532 004767 005272      JSR    PC,RPOKE ;SHIFT IN THIS CHAR
2602                                ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2603 012536 012767 000054 166400      MOV    #54,TEMP1 ;FRAME & PARITY ERROR
2604 012544 012700 170026      MOV    #RXERR!OVRUN!FRMERR!PARER!26,R0 ;EXPECTED
2605 012550 000167 177660      JMP    3$ ;DO IT AGAIN
2606 012554 012767 000454 166362 6$:      MOV    #454,TEMP1 ;BAD STOP BIT FOR FRAME ERROR
2607 012562 012700 120226      MOV    #RXERR!FRMERR!226,R0 ;EXPECTED
2608 012566 000167 177642      JMP    3$ ;DO IT AGAIN
2609 012572 012767 000054 166344 7$:      MOV    #54,TEMP1 ;BAD STOP BIT & PARITY
2610 012600 012700 130026      MOV    #RXERR!FRMERR!PARER!26,R0 ;EXPECTED
2611 012604 000167 177624      JMP    3$ ;DO IT AGAIN
2612                                5$:
2613                                SCOPE
2614                                ;;THIS TEST PROVES THAT RXERR FREEZES THE 'RECEIVER RESET'
2615                                ;;WHILE IN STRIP SYNC MODE
2616                                ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2617                                ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2618                                ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
                                ;;IS DISCOMBOBULATED

```



```

2619      :: IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2620      :: NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2621      :: PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT.....
2622      :: BUT, IF AN RXERR OCCURS  RXDONE PLUS RXERR ARE ASSERTED
2623      :: MODE: ISOC (ISYMOD)
2624      :: LENGTH: SIX
2625      :: PARITY: EVEPAR
2626      :: CHARACTER EXPECTED:126
2627      :: NOTE THAT THE PARITY BIT SHOULD SHOW
2628      :: UP IN THE DATA  IE. BIT SIX FOR
2629      :: SIX LEVEL CODE
2630      :: CHARACTER SENT: SYNC CHARACTER
2631      :: NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2632
2633 012612 012767 000037 166306 TST31: MOV #31,TSTNO ;SAVE THIS
2634 012620 012767 013214 166270 MOV #TST32,NEXT ;GO TO THIS TEST WHEN THRU
2635 012626 052777 000400 005504 BIS #MRESET,@TXCSR ;MASTER RESET
2636 012634 012777 000000 005472 MOV #ISYMOD,@PARCSR ;SET THE MODE
2637 012642 052777 000400 005470 BIS #MRESET,@TXCSR ;MASTER RESET
2638
2639 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2640 012650 012777 064001 005462 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2641
2642 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2643 012656 012777 003526 005450 MOV #ISYMOD!SIX!EVEPAR!126,@PARCSR
2644 012664 016703 005440 MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2645 012670 012767 000003 166244 MOV #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
2646 012676 052777 000020 005420 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2647 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2648 012704 042777 020000 005426 BIC #CLK,@TXCSR ;POKE CLK DOWN
2649 012712 052777 020000 005420 BIS #CLK,@TXCSR ;POKE CLK UP
2650 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2651 012720 042777 020000 005412 BIC #CLK,@TXCSR ;POKE CLK DOWN
2652 012726 052777 020000 005404 BIS #CLK,@TXCSR ;POKE CLK UP
2653 012734 052777 000400 005362 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2654 012742 012767 000011 166170 2$: MOV #9,SHIFT ;# OF SHIFTS
2655 012750 012767 000654 166166 MOV #654,TEMP1 ;SYNC CHAR + START&STOP+ PARITY
2656 012756 004767 005046 1$: JSR PC,RPOKE ;SHIFT IN THIS CHARACTER
2657 012762 105777 005336 TSTB @RXCSR ;RXDONE = 0 ?
2658 012766 100001 BPL 64$
2659 012770 104000 HLT ;RXDONE SHOULD NOT BE SET
2660 012772 64$:
2661 012772 005367 166144 DEC COUNT ;# OF SYNC CHARS
2662 012776 001361 BNE 2$ ;GO AGAIN ?
2663 013000 012700 000126 MOV #126,R0 ;EXPECTED
2664 013004 017701 005320 MOV @RXDBUF,R1 ;ACTUAL
2665 ;NOTE THAT THIS IS THE FIRST TIME
2666 ;RXDBUF IS READ.....THERE SHOULD BE
2667 ;NO OVER RUN ERRORS
2668 013010 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
2669 013012 001401 BEQ 65$
2670 013014 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2671 ;THERE SHOULD BE NO RXERR'S
2672 013016 65$:
2673 013016 012767 000004 166116 MOV #4,COUNT ;# OF TIMES
2674 013024 012700 110026 MOV #RXERR!PARER!26,R0 ;EXPECTED

```

```

2675 013030 012767 000454 166106      MOV    #454,TEMP1      ;BAD SYNC CHAR (WRONG PARITY)
2676 013036 012767 000011 166074 3$:  MOV    #9,,SHIFT      ;# OF SHIFTS
2677 013044 004767 004760      JSR    PC,RPOKE       ;SHIFT IN THIS CHAR
2678 013050 105777 005250      TSTB  @RXCSR ;RXDONE = 1?
2679 013054 100401      BMI   66$
2680 013056 104000      HLT   ;RXDONE SHOULD BE SET
2681 013060      66$:
2682 013060 017701 005244      MOV    @RXDBUF,R1     ;ACTUAL DATA
2683 013064 020001      CMP    R0,R1         ;COMPARE EXP VS ACT
2684 013066 001401      BEQ   67$
2685 013070 104002      HLT   2              ;DID THE RESPECTIVE ERROR STOP THE
2686      ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2687      ;.....CHECK THIS.....
2688 013072      67$:
2689      ;NOTE THAT THE PARITY BIT SHOULD
2690      ;SHOW UP IN THE DATA
2691      ;IE. BIT SIX FOR SIX LEVEL CODE
2692 013072 005367 166044      DEC    COUNT         ;# OF SYNC CHARS
2693 013076 001445      BEQ   5$             ;FINISHED ? GET OUT OF TEST
2694 013100 022767 000003 166034      CMP    #3,COUNT      ;# OF SYNC CHARS
2695 013106 001423      BEQ   6$             ;CHECK FRAME ERROR ?
2696 013110 022767 000002 166024      CMP    #2,COUNT      ;# OF SYNC CHARS
2697 013116 001426      BEQ   7$             ;CHECK FRAME ERROR & BAD PARITY ?
2698      ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2699 013120 012767 000011 166012      MOV    #9,,SHIFT      ;# OF SHIFTS
2700 013126 012767 000054 166010      MOV    #54,TEMP1     ;FRAME & PARITY ERROR
2701 013134 004767 004670      JSR    PC,RPOKE       ;SHIFT IN THIS CHAR
2702      ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2703 013140 012767 000054 165776      MOV    #54,TEMP1     ;FRAME & PARITY ERROR
2704 013146 012700 170026      MOV    #RXERR!OVRRUN!FRMERR!PARER!26,R0 ;EXPECTED
2705 013152 000167 177660      JMP   3$             ;DO IT AGAIN
2706 013156 012767 000254 165760 6$:  MOV    #254,TEMP1    ;BAD STOP BIT FOR FRAME ERROR
2707 013164 012700 120126      MOV    #RXERR!FRMERR!126,R0 ;EXPECTED
2708 013170 000167 177642      JMP   3$             ;DO IT AGAIN
2709 013174 012767 000054 165742 7$:  MOV    #54,TEMP1     ;BAD STOP BIT & PARITY
2710 013202 012700 130026      MOV    #RXERR!FRMERR!PARER!26,R0 ;EXPECTED
2711 013206 000167 177624      JMP   3$             ;DO IT AGAIN
2712      5$:  SCOPE
2713      ;;THIS TEST PROVES THAT RXERR FREEZES THE 'RECEIVER RESET'
2714      ;;WHILE IN STRIP SYNC MODE
2715      ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2716      ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2717      ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2718      ;;IS DISCOMBOBULATED
2719      ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2720      ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2721      ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT.....
2722      ;;BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
2723      ;;MODE: ISOC (ISYMOD)
2724      ;;LENGTH: FIVE
2725      ;;PARITY: EVEPAR
2726      ;;CHARACTER EXPECTED:66
2727      ;NOTE THAT THE PARITY BIT SHOULD SHOW
2728      ;UP IN THE DATA IE. BIT FIVE FOR
2729      ;FIVE LEVEL CODE
2730      ;;CHARACTER SENT: SYNC CHARACTER

```



```

2731                                     ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2732
2733 013214 012767 000040 165704 1ST32: MOV #32,TSTNO ;SAVE THIS
2734 013222 012767 013616 165666 MOV #.EOP,NEXT ;GO TO THIS TEST WHEN THRU
2735 013230 052777 000400 005102 BIS #MRESÉT,@TXCSR ;MASTER RESET
2736 013236 012777 000000 005070 MOV #ISYMOD,@PARCSR ;SET THE MODE
2737 013244 052777 000400 005066 BIS #MRESÉT,@TXCSR ;MASTER RESET
2738
2739 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2740 013252 012777 064001 005060 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2741
2742 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2743 013260 012777 001466 005046 MOV #ISYMOD!FIVE!EVEPAR!66,@PARCSR
2744 013266 016703 005036 MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2745 013272 012767 000003 165642 MOV #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
2746 013300 052777 000020 005016 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2747 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2748 013306 042777 020000 005024 BIC #CLK,@TXCSR ;POKE CLK DOWN
2749 013314 052777 020000 005016 BIS #CLK,@TXCSR ;POKE CLK UP
2750 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2751 013322 042777 020000 005010 BIC #CLK,@TXCSR ;POKE CLK DOWN
2752 013330 052777 020000 005002 BIS #CLK,@TXCSR ;POKE CLK UP
2753 013336 052777 000400 004760 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2754 013344 012767 000010 165566 2$: MOV #8,SHIFT ;# OF SHIFTS
2755 013352 012767 000354 165564 MOV #354,TEMP1 ;SYNC CHAR + START&STOP+ PARITY
2756 013360 004767 004444 1$: JSR PC,RPOKE ;SHIFT IN THIS CHARACTER
2757 013364 105777 004734 TSTB @RXCSR ;RXDONE = 0 ?
2758 013370 100001 BPL 64$
2759 013372 104000 HLT ;RXDONE SHOULD NOT BE SET
2760
2761 013374 005367 165542 64$: DEC COUNT ;# OF SYNC CHARS
2762 013400 001361 BNE 2$ ;GO AGAIN ?
2763 013402 012700 000066 MOV #66,R0 ;EXPECTED
2764 013406 017701 004716 MOV @RXDBUF,R1 ;ACTUAL
2765 ;NOTE THAT THIS IS THE FIRST TIME
2766 ;RXDBUF IS READ.....THERE SHOULD BE
2767 ;NO OVER RUN ERRORS
2768 013412 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
2769 013414 001401 BEQ 65$
2770 013416 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2771 ;THERE SHOULD BE NO RXERR'S
2772
2773 013420 65$:
2774 013420 012767 000004 165514 MOV #4,COUNT ;# OF TIMES
2775 013426 012700 110026 MOV #RXERR!PARER!26,R0 ;EXPECTED
2776 013432 012767 000254 165504 MOV #254,TEMP1 ;BAD SYNC CHAR (WRONG PARITY)
2777 013440 012767 000010 165472 3$: MOV #8,SHIFT ;# OF SHIFTS
2778 013446 004767 004356 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2779 013452 105777 004646 TSTB @RXCSR ;RXDONE = 1?
2780 013456 100401 BMI 66$
2781 013462 104000 HLT ;RXDONE SHOULD BE SET
2782
2782 013462 017701 004642 66$: MOV @RXDBUF,R1 ;ACTUAL DATA
2783 013466 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2784 013470 001401 BEQ 67$
2785 013472 104002 HLT 2 ;DID THE RESPECTIVE ERROR STOP THE
2786 ;AUTOMATIC RESSETING OF RXDONE & ERROR FLAGS

```

```

2787
2788 013474          67$:          ;.....CHECK THIS.....
2789
2790
2791
2792 013474 005367 165442          DEC
2793 013500 001445          BEQ
2794 013502 022767 000003 165432  CMP
2795 013510 001423          BEQ
2796 013512 022767 000002 165422  CMP
2797 013520 001426          BEQ
2798
2799 013522 012767 000010 165410  MOV
2800 013530 012767 000054 165406  MOV
2801 013536 004767 004266          JSR
2802
2803 013542 012767 000054 165374  ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2804 013550 012700 170026          MOV
2805 013554 000167 177660          MOV
2806 013560 012767 000154 165356 6$:  JMP
2807 013566 012700 120066          MOV
2808 013572 000167 177642          JMP
2809 013576 012767 000054 165340 7$:  MOV
2810 013604 012700 130026          MOV
2811 013610 000167 177624          JMP
2812 013614 104400          5$:  SCOPE

```

;NOTE THAT THE PARITY BIT SHOULD
;SHOW UP IN THE DATA
;IE. BIT FIVE FOR FIVE LEVEL CODE
COUNT ;# OF SYNC CHARS
5\$;FINISHED ? GET OUT OF TEST
#3,COUNT ;# OF SYNC CHARS
6\$;CHECK FRAME ERROR ?
#2,COUNT ;# OF SYNC CHARS
7\$;CHECK FRAME ERROR & BAD PARITY ?
;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
#8,SHIFT ;# OF SHIFTS
#54,TEMP1 ;FRAME & PARITY ERROR
PC,RPOKE ;SHIFT IN THIS CHAR
#54,TEMP1 ;FRAME & PARITY ERROR
#RXERR!OVRUN!FRMERR!PARER!26,R0 ;EXPECTED
3\$;DO IT AGAIN
#154,TEMP1 ;BAD STOP BIT FOR FRAME ERROR
#RXERR!FRMERR!66,R0 ;EXPECTED
3\$;DO IT AGAIN
#54,TEMP1 ;BAD STOP BIT & PARITY
#RXERR!FRMERR!PARER!26,R0 ;EXPECTED
3\$;DO IT AGAIN


```

2813
2814
2815
2816
2817
2818
2819
2820 013616 104402
2821 013620 016760
2822 013622 104410 014054
2823 013626 104402 016501
2824 013632 105767 165344
2825 013636 001511
2826 013640 005767 165352
2827 013644 001007
2828 013646 104402 016513
2829 013652 016700 165340
2830 013656 000000
2831
2832 013660 000167 165534
2833 013664 062767 000010 165312 RUNIT:
2834 013672 062767 000010 165312 ZERO:
2835 013700 000241
2836 013702 006167 165312
2837 013706 103410
2838
2839 013710 036767 165304 165300
2840 013716 001762
2841 013720 004767 000034
2842 013724 000167 000174
2843 013730 012767 000001 165262 2$:
2844
2845 013736 016767 165244 165240
2846 013744 016767 165244 165240
2847 013752 004767 000002
2848 013756 000441
2849 013760 016767 165220 004040 REPLAY:
2850 013766 004767 003702
2851 013772 016767 165214 004350
2852 014000 062767 000002 165204
2853 014006 016767 165200 004336
2854 014014 062767 000002 165170
2855 014022 016767 165164 004324
2856 014030 062767 000002 165154
2857 014036 016767 165150 004312
2858 014044 016767 004300 165140
2859 014052 000207
2860
2861 014054 000001 OUTCRY: 1
2862 014056 006 002 .BYTE 6,2
2863 014060 020324 RXCSR
2864
2865 014062 CCC:
2866 014062 005067 165046 CLR LSTERR ;CLEAR LAST ERROR PC
2867 014066 005067 165154 CLR ER?FLG ;CLEAR ERROR FLAG
2868 014072 005267 165032 INC PASCNT ;UPDATE PASS COUNT

```

```

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

```

```

;TYPE NAME OF TEST
;ARE YOU RUNNING MULTIPLE DEVICES ?
:NO, JUMP AROUND
;ARE ANY DEVICES ACTIVE ?
:YES
:NO
;DISPLAY ACTREG
;SELECT SOMETHING TO RUN @ ACTREG:
(PUT SW00 =1)
;START OVER AGAIN..... YOU DESELECTED EVERYTHING
;NEXT BLOCK (ADDRESSES)
;NEXT BLOCK (VECTORS)
;UP DATE ROTATING POINTER
;IS IT THE LAST DEVICE
;TO BE TESTED IN THIS PASS ?
;TEST THIS DEVICE FOR ACTIVE STATUS
;IF NOT ACTIVE, TRY NEXT ADDRESS
;CALCULATE NEW PARAMETERS
;YES IT WAS ACTIVE, TEST THIS DEVICE
;OK!, NOW SET UP ROTATING
;POINTER FOR NEXT MULTIPLE PASS
;RESTORE BASE ADDRESS
;RESTORE BASE INTERRUPT VECTORS
;CALC NEW PARAMETERS
;JUMP AROUND REPLAY
;SET UP FOR NEW ADDRESSES
;CREATE NEW ADDRESSES
;CREATE DURIV
;CREATE DURIS
;CREATE DUTIV
;CREATE DUTIS
;RESTORE

```

```
2869 014076 016777 165026 164776      MOV      PASCNT,@LIGHTS      ;DISPLAY PASS COUNT
2870 014104 013701 000042                MOV      @#42,R1            ;CHECK FOR ACT-11 OR DDP
2871 014110 001405                BEQ      RESTRT            ;IF NOT, CONTINUE TESTING
2872 014112 000005                RESET
2873 014114 004711                LOGICAL: JSR      PC,(R1)
2874 014116 000240                NOP
2875 014120 000240                NOP
2876 014122 000240                NOP
2877 014124 012767 000340 163644  RESTRT: MOV      #340,PS      ;PREVENT INTERRUPTS (PRIO: 7)
2878 014132 104413                CKSWR                      ;CHECK FOR ^G
2879 014134 012767 003446 164752      MOV      #TST1,RTRN
2880 014142 000167 167300                JMP      TST1
2881
2882                                ;SCOPE LOOP AND INTERATION HANDLER
2883
2884 014146                .SCOPE:
2885                ;***** START OF CODE FOR THE X OR TESTER *****
2886 014146 000424                BR      4$                ;IF RUNNING ON THE X OR TESTER CHANGE
2887                                ;THIS INSTRUCTION TO A 'NOP'(NOP=240)
2888 014150 013746 000004                MOV      @#4,-(SP)         ;SAVE CONTENTS OF ERROR VECTOR
2889 014154 012737 014174 000004      MOV      #1$,@#4          ;SET FOR TIME OUT
2890 014162 005737 177060                TST      @#177060         ;TIME OUT ON X OR ?
2891 014166 012637 000004      MOV      (SP)+,@#4        ;RESTORE ERROR VECTOR
2892 014172 000404                BR      2$                ;GO TO NEXT TEST
2893 014174 022626                1$:  CMP      (SP)+,(SP)+    ;CLEAR THE STACK AFTER A TIMEOUT
2894 014176 012637 000004      MOV      (SP)+,@#4        ;RESTORE ERROR VECTOR
2895 014202 000403                BR      3$                ;LOOP ON PRESENT TEST
2896 014204 016767 164706 164702  2$:  MOV      NEXT,RTRN        ;SET UP NEXT TEST IN RTRN
2897 014212 016716 164676                3$:  MOV      RTRN,(SP)       ;SET UP STACK FOR RTI
2898 014216 000002                RTI
2899 014220                4$:  ;***** END OF CODE FOR THE X OR TESTER *****
2900 014220 104413                CKSWR                      ;CHECK FOR ^G
2901 014222 032777 040000 164650      TTST: BIT      #SW14,@SWR    ;LOOP ON CURRENT TEST ?
2902 014230 001407                BEQ      1$
2903 014232 000432                BR      3$
2904 014234 105777 164644                TSTB    @TKCSR            ;TEST TTY FLAG
2905 014240 100027                BPL      3$
2906 014242 017700 164640                MOV      @TKDBR,R0        ;CLR DONE BIT
2907 014246 000412                BR      2$                ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2908 014250 032777 004000 164622  1$:  BIT      #SW11,@SWR      ;INHIBIT ITERATIONS ?
2909 014256 001006                BNE      2$
2910 014260 005267 164640                INC      LPCNT
2911 014264 026767 164634 164630      CMP      LPCNT,ICOUNT     ;CHECK FOR ITERATION CNT FINISH
2912 014272 101412                BLOS    3$
2913 014274 105067 164746                2$:  CLRB    ERRFLG
2914 014300 005067 164620                CLR      LPCNT
2915 014304 012767 000005 164610      MOV      #5,ICOUNT        ;SET UP ITERATION COUNT
2916 014312 016767 164600 164574      MOV      NEXT,RTRN        ;SET UP NEXT TEST IN RTRN
2917 014320 016716 164570                3$:  MOV      RTRN,(SP)       ;SET UP STACK FOR RTI
2918 014324 000002                RTI
2919 014326 001407                BRW:  1407                ;RESTORE 'BEQ 1$' INSTRUCTION
2920 014330 000432                BRX:  432                ;RESTORE 'BR 3$' INSTRUCTION
2921
2922                                ;CHECK FOR FREEZE ON CURRENT DATA
2923
2924 014332 104413                .SCOPE1: CKSWR            ;CHECK FOR ^G
```



```

2925 014334 032777 001000 164536      BIT      #SW09,@SWR
2926 014342 001402                BEQ      1$
2927 014344 016716 164550                MOV      LOCK,(SP)
2928 014350 000002                1$:     RTI
2929
2930                                ;TELETYPE OUTPUT ROUTINE
2931
2932 014352 010546                .TYPE:  MOV      R5,-(SP)
2933 014354 017605 000002                MOV      @2(SP),R5
2934 014360 062766 000002 000002                ADD      #2,2(SP)
2935 014366 105715                1$:     TSTB   (R5)                ;LOOK FOR '0'
2936 014370 001406                BEQ      3$
2937 014372 105777 164512                2$:     TSTB   @TPCSR                ;TEST DONE BIT
2938 014376 100375                BPL      2$
2939 014400 112577 164506                MOVB    (R5)+,@TPDBR                ;TYPE CHAR
2940 014404 000770                BR       1$                ;DO IT AGAIN UNTIL '0' IS SEEN
2941 014406 012605                3$:     MOV      (SP)+,R5
2942 014410 000002                RTI
2943
2944                                ;ASCII STRING INPUT ROUTINE
2945
2946 014412 010346                .INSTR: MOV      R3,-(SP)
2947 014414 010446                MOV      R4,-(SP)
2948 014416 017667 000004 000010                MOV      @4(SP),.MSG
2949 014424 062766 000002 000004                ADD      #2,4(SP)
2950 014432 104402                .INST1: TYPE
2951 014434 000000                .MSG:   0
2952 014436 012704 017460                MOV      #INBUF,R4
2953 014442 012703 000007                MOV      #7,R3
2954 014446 105777 164432                1$:     TSTB   @TKCSR
2955 014452 100375                BPL      1$
2956 014454 117714 164426                MOVB    @TKDBR,(R4)
2957 014460 142714 000200                BICB   #200,(R4)
2958 014464 121427 000025                CMPB   (R4),#25                ;IS IT <^U>
2959 014470 001003                BNE     200$
2960 014472 104402 016670                TYPE,MCRLF
2961 014476 000755                BR      .INST1
2962 014500 122427 000015                200$:  CMPB   (R4)+,#15
2963 014504 001423                BEQ     INSTR2
2964 014506 117777 164374 164376                MOVB    @TKDBR,@TPDBR
2965 014514 105777 164370                2$:     TSTB   @TPCSR
2966 014520 100375                BPL      2$
2967 014522 005303                DEC     R3
2968 014524 001350                BNE     1$
2969 014526 000402                BR      .INSTG
2970 014530 010346                .INSTE: MOV      R3,-(SP)
2971 014532 010446                MOV      R4,-(SP)
2972 014534 104402                .INSTG: TYPE
2973 014536 016664                MQM
2974 014540 005737 016026                TST     @#RDSW
2975 014544 001402                BEQ     400$
2976 014546 104402 016670                TYPE,MCRLF
2977 014552 000727                400$:  BR      .INST1
2978 014554 012604                INSTR2: MOV     (SP)+,R4
2979 014556 012603                MOV     (SP)+,R3
2980 014560 000002                RTI

```

2981
2982
2983
2984 014562 010546
2985 014564 010446
2986 014566 016605 000004
2987 014572 012567 000170
2988 014576 012567 000166
2989 014602 012567 000164
2990 014606 112567 000162
2991 014612 112567 000157
2992 014616 010566 000004
2993 014622 005005
2994 014624 012704 017460
2995 014630 122714 000015
2996 014634 001420
2997 014636 121427 000060
2998 014642 002415
2999 014644 121427 000067
3000 014650 003012
3001 014652 142714 000060
3002 014656 152405
3003 014660 122714 000015
3004 014664 001414
3005 014666 006305
3006 014670 006305
3007 014672 006305
3008 014674 000760
3009 014676 122714 000015
3010 014702 001003
3011 014704 005737 016026
3012 014710 001023
3013 014712 104404
3014 014714 000742
3015
3016
3017
3018 014716 020567 000046
3019 014722 101365
3020 014724 020567 000036
3021 014730 103762
3022 014732 136705 000036
3023 014736 001357
3024
3025
3026
3027 014740 016704 000026
3028 014744 010524
3029 014746 062705 000002
3030 014752 105367 000017
3031 014756 001372
3032 014760 012604
3033 014762 012605
3034 014764 000002
3035 014766 000000
3036 014770 000000

```

;CONVERT ASCII STRING TO OCTAL
.PARAM: MOV R5, -(SP)
MOV R4, -(SP)
MOV 4(SP), R5
MOV (R5)+, LOLIM
MOV (R5)+, HILIM
MOV (R5)+, DEVADR
MOVB (R5)+, LOBITS
MOVB (R5)+, ADCRCNT
MOV R5, 4(SP)
PARAM1: CLR R5
MOV #INBUF, R4
CMPB #15, (R4)
1$: BEQ PARERR
CMPB (R4), #60
BLT PARERR
CMPB (R4), #67
BGT PARERR
BICB #60, (R4)
BISB (R4)+, R5
CMPB #15, (R4)
BEQ LIMITS
ASL R5
ASL R5
ASL R5
BR 1$
PARERR: CMPB #15, (R4) ;IS FIRST CHARACTER A <CR>
BNE 120$
TST @#RDSW ;IS CKSWR ROUTINE BEING USED
BNE PARTI
120$: INSTER
BR PARAM1

;TEST TO SEE IF NUMBER IS WITHIN LIMITS
LIMITS: CMP R5, HILIM
BHI PARERR
CMP R5, LOLIM
BLO PARERR
BITB LOBITS, R5
BNE PARERR

;STORE NUMBER AT SPECIFIED ADDRESS
1$: MOV DEVADR, R4
MOV R5, (R4)+
ADD #2, R5
DECB ADCRCNT
BNE 1$
PARTI: MOV (SP)+, R4
MOV (SP)+, R5
RTI
LOLIM: 0
HILIM: 0
    
```



```

3037 014772 000000          DEVADR: 0
3038 014774 000000          LOBITS: 0
3039          014775          ADRCNT=LOBITS+1
3040
3041                                ;SAVE PC OF TEST THAT FAILED AND R0-R5
3042
3043 014776 016667 000004 164170 .SAV05: MOV     4(SP),SAVPC
3044
3045                                ;SAVE R0-R5
3046
3047 015004 010567 164160      SV05:  MOV     R5,SAVR5
3048 015010 010467 164152      MOV     R4,SAVR4
3049 015014 010367 164144      MOV     R3,SAVR3
3050 015020 010267 164136      MOV     R2,SAVR2
3051 015024 010167 164130      MOV     R1,SAVR1
3052 015030 010067 164122      MOV     R0,SAVR0
3053 015034 000002          RTI
3054
3055                                ;RESTORE R0-R5
3056
3057 015036 016700 164114      .RES05: MOV     SAVR0,R0
3058 015042 016701 164112      MOV     SAVR1,R1
3059 015046 016702 164110      MOV     SAVR2,R2
3060 015052 016703 164106      MOV     SAVR3,R3
3061 015056 016704 164104      MOV     SAVR4,R4
3062 015062 016705 164102      MOV     SAVR5,R5
3063 015066 000002          RTI
3064
3065                                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3066
3067 015070 104402          .CONVR: TYPE
3068 015072 016670          MCRLF
3069 015074 010046          .CNVRT: MOV     R0,-(SP)
3070 015076 010146          MOV     R1,-(SP)
3071 015100 010346          MOV     R3,-(SP)
3072 015102 010446          MOV     R4,-(SP)
3073 015104 010546          MOV     R5,-(SP)
3074 015106 017601 000012      MOV     @12(SP),R1
3075 015112 016767 002402 164030  MOV     TEMP,TEMP3
3076 015120 062766 000002 000012  ADD     #2,12(SP)
3077 015126 012167 000154      MOV     (R1)+,WRDCNT
3078 015132 112167 000152      1$:  MOVVB  (R1)+,CHRCNT
3079 015136 112167 000147      MOVVB  (R1)+,SPACNT
3080 015142 013167 000144      MOV     @(R1)+,BINWRD
3081 015146 016704 000140      2$:  MOV     BINWRD,R4
3082 015152 116705 000132      MOVVB  CHRCNT,R5
3083 015156 012700 017520      MOV     #TEMP,R0
3084 015162 010403      3$:  MOV     R4,R3
3085 015164 042703 177770      BIC     #177770,R3
3086 015170 062703 000060      ADD     #060,R3
3087 015174 110320      MOVVB  R3,(R0)+
3088 015176 006204      ASR     R4
3089 015200 042704 100000      BIC     #100000,R4
3090 015204 006204      ASR     R4
3091 015206 006204      ASR     R4
3092 015210 005305      DEC     R5

```

```

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

```

3093	015212	001363				BNE	3\$	
3094	015214	012703	017560			MOV	#MDATA,R3	
3095	015220	114023			4\$:	MOVB	-(R0),(R3)+	
3096	015222	105367	000062			DECB	CHRCNT	
3097	015226	001374				BNE	4\$	
3098	015230	105767	000055			TSTB	SPACNT	
3099	015234	001405				BEQ	6\$	
3100	015236	112723	000040		5\$:	MOVB	#040,(R3)+	
3101	015242	105367	000043			DECB	SPACNT	
3102	015246	001373				BNE	5\$	
3103	015250	105013			6\$:	CLRB	(R3)	
3104	015252	104402				TYPE		
3105	015254	017560				MDATA		
3106	015256	005367	000024			DEC	WRDCNT	
3107	015262	001323				BNE	1\$	
3108	015264	016767	163660	002226		MOV	TEMP3,TEMP	
3109	015272	012605				MOV	(SP)+,R5	
3110	015274	012604				MOV	(SP)+,R4	
3111	015276	012603				MOV	(SP)+,R3	
3112	015300	012601				MOV	(SP)+,R1	
3113	015302	012600				MOV	(SP)+,R0	
3114	015304	000002				RTI		
3115	015306	000000				WRDCNT:	0	
3116	015310	000000				CHRCNT:	0	
3117		015311				SPACNT=	CHRCNT+1	
3118	015312	000000				BINWRD:	0	
3119								
3120								
3121								
3122								
3123								
3124								
3125	015314	017605	000000			.SETFLG:MOV	@(SP),R5	
3126	015320	122767	000116	002132		CMPB	#'N',INBUF	:IS IT 'N' ?
3127	015326	001002				BNE	1\$	
3128	015330	105015				CLRB	(R5) ;000	
3129	015332	000406				BR	2\$	
3130	015334	122767	000131	002116	1\$:	CMPB	#'Y',INBUF	:IS IT 'Y' ?
3131	015342	001005				BNE	3\$	
3132	015344	112715	177777			MOVB	#-1,(R5)	:377
3133	015350	062716	000002		2\$:	ADD	#2,(SP)	
3134	015354	000002				RTI		
3135	015356	104404			3\$:	INSTER		:RETRY
3136	015360	000755				BR	.SETFLG	
3137								
3138								
3139								
3140								
3141								
3142	015362	011646				.TRPSR:MOV	(SP),-(SP)	:GET PC OF RETURN
3143	015364	162716	000002			SUB	#2,(SP)	:PC OF TRAP
3144	015370	017616	000000			MOV	@(SP),(SP)	:GET TRP
3145	015374	006316			TRPOK:	ASL	(SP)	:MULTIPLY TRAP ARG BY 2
3146	015376	042716	177001			BIC	#177001,(SP)	:CLEAR UNWANTED BITS
3147	015402	062716	001366			ADD	#.TRPTAB,(SP)	:POINTER TO SUBROUTINE ADDRESS
3148	015406	017616	000000			MOV	@(SP),(SP)	:SUBROUTINE ADDRESS


```
3149 015412 000136          JMP      @ (SP)+          ;GO TO SUBROUTINE
3150
3151                          ;ERROR HANDLER
3152
3153 015414 104413          .HLT:   CKSWR          ;CHECK FOR ^G
3154 015416 032777 020000 163454  BIT      #SW13,@SWR      ;INHIBIT ERROR TYPE OUT ?
3155 015424 001061          BNE     HALTS
3156 015426 021667 163502  CMP      (SP),LSTERR
3157 015432 001404          BEQ     1$
3158 015434 011667 163474  MOV      (SP),LSTERR
3159 015440 105067 163602  CLRB    ERRFLG
3160 015444 104406          1$:    SAVO5
3161 015446 011605          MOV      (SP),R5
3162 015450 162705 000002  SUB      #2,R5
3163 015454 011504          MOV      (R5),R4
3164 015456 006304          ASL     R4
3165 015460 061504          ADD     (R5),R4
3166 015462 006304          ASL     R4
3167 015464 042704 177001  BIC     #177001,R4
3168 015470 062704 020274  ADD     #.ERRTAB,R4
3169 015474 012467 000040  MOV     (R4)+,ERRMSG
3170 015500 012467 000046  MOV     (R4)+,DATAHD
3171 015504 011467 000054  MOV     (R4)+,DATABP
3172 015510 105767 163532  TSTB   ERRFLG
3173 015514 001403          BEQ     TYPMSG
3174 015516 005767 000042  TST    DATABP
3175 015522 001014          BNE     TYPDAT
3176 015524 104410          TYPMSG: CONVRT
3177 015526 015656          ERTAB0
3178 015530 112767 177777 163510  MOVB   #-1,ERRFLG
3179 015536 104402          TYPE
3180 015540 000000          ERRMSG: 0
3181 015542 005767 000004  TST    DATAHD
3182 015546 001402          BEQ     TYPDAT
3183 015550 104402          TYPE
3184 015552 000000          DATAHD: 0
3185 015554 005767 000004  TYPDAT: TST    DATABP
3186 015560 001402          BEQ     RESREG
3187 015562 104410          CONVRT
3188 015564 000000          DATABP: 0
3189 015566 104407          RESREG: RES05
3190 015570 005777 163304  HALTS:  TST    @SWR
3191 015574 100005          BPL    EXITER
3192 015576 010046          PUSHRO
3193 015600 016600 000002  MOV     2(SP),R0
3194 015604 000000          HALT
3195 015606 012600          POPRO
3196 015610 104413          EXITER: CKSWR          ;CHECK FOR ^G
3197 015612 005267 163314  INC     ERRCNT
3198 015616 032777 000400 163254  BIT     #SW08,@SWR      ;LOOP ON ERROR ?
3199 015624 001007          BNE     1$
3200 015626 032777 002000 163244  BIT     #SW10,@SWR      ;ESCAPE TO NEXT ON ERROR ?
3201 015634 001407          BFQ     2$
3202 015636 016767 163254 163250  MOV     NEXT,RTRN
3203 015644 012706 001100          1$:    MOV     #C'ACK,SP
3204 015650 000177 163240  JMP     @RTRN          ;REINITIALIZE SP
```

```

3205 015654 000002      2$: RTI
3206 015656 000001      ERTABC: 1
3207 015660 006      002      .BYTE 6,2
3208 015662 001174      SAVPC
3209      :ENTER HERE ON POWER FAILURE
3210
3211
3212 015664 010046      .PFAIL: MOV R0,-(SP)      ;SAVE R0-R5 ON PROCESSOR STACK
3213 015666 010146      MOV R1,-(SP)
3214 015670 010246      MOV R2,-(SP)
3215 015672 010346      MOV R3,-(SP)
3216 015674 010446      MOV R4,-(SP)
3217 015676 010546      MOV R5,-(SP)
3218 015700 016746 162120      MOV 24,-(SP)
3219 015704 010667 163262      MOV SP,SAVSP      ;SAVE STACK POINTER
3220 015710 012767 015722 162106      MOV #RESTART,24      ;SET UP FOR POWER UP TRAP
3221 015716 000000      HALT      ;HALT ON POWER DOWN NORMAL
3222 015720 000777      1$: BR 1$
3223
3224      ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3225
3226 015722 016706 163244      RESTAR: MOV SAVSP,SP      ;RESTORE STACK POINTER
3227 015726 012605      MOV (SP)+,R5      ;RESTORE R0-R5
3228 015730 012604      MOV (SP)+,R4
3229 015732 012603      MOV (SP)+,R3
3230 015734 012602      MOV (SP)+,R2
3231 015736 012601      MOV (SP)+,R1
3232 015740 012600      MOV (SP)+,R0
3233 015742 012767 015664 162054      MOV #.PFAIL,24      ;SET UP FOR POWER FAILURE
3234 015750 012767 000340 162020      MOV #340,PS
3235 015756 012706 001100      MOV #STACK,SP
3236 015762 005067 001532      1$: CLR TEMP
3237 015766 005267 001526      INC TEMP
3238 015772 001375      BNE 1$
3239 015774 104410      CONVRT
3240 015776 016020      PFTAB
3241 016000 104402      TYPE
3242 016002 016673      MPFAIL
3243 016004 005067 163236      CLR ERRFLG
3244 016010 005067 163120      CLR LSTERR
3245 016014 000177 163074      JMP @RTRN
3246 016020 000001      PFTAB: 1
3247 016022 006      002      .BYTE 6,2
3248 016024 001114      RTRN
3249
3250
3251      ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING .
3252      ;OF LOC.176.
3253      ;LOCATIONS USED:
3254 016026 000000      RDSW: .WORD 0
3255
3256
3257 016030 005737 000042      .CKSWR: TST @#42
3258 016034 001042      BNE OUT
3259 016036 022767 000176 163034      CMP #C:REG,SWR      ;SOFTWARE SWITCH REGISTER PRESENT
3260 016044 001036      BNE OUT      ;NO, GET OUT

```


3261	016046	105777	163032		TSTB	@TKCSR			:YES, WAIT FOR
3262	016052	100033			BPL	OUT			:READY, GET CHARACTER
3263	016054	017767	163026	176352	MOV	@TKDBR, .MSG			:AND STRIP OFF
3264	016062	042767	177600	176344	BIC	#177600, .MSG			:THE GARBAGE
3265	016070	122767	000007	176336	CMPB	#7, .MSG			:IS IT A <^G>
3266	016076	001021			BNE	OUT			
3267	016100	104402	016156		TYPE, \$CNTG				
3268	016104	005137	016026		.CNTLU: COM	@#RDSW			
3269	016110	104402	016163		TYPE, \$MSWR				
3270	016114	104411	016150		CNVRT, SWREGC				
3271	016120	104403	016173		INSTR, \$MNEW				
3272	016124	104405			PARAM				
3273	016126	000000			0				
3274	016130	177777			177777				
3275	016132	000176			SWREG				
3276	016134	000	001		.BYTE	0,1			
3277	016136	104402	016670		TYPE, MCRLF				
3278	016142	005037	016026		OUT: CLR	@#RDSW			
3279	016146	000002			RTI				
3280	016150	000001			SWREGC: 1				
3281	016152	006	002		.BYTE	6,2			
3282	016154	000176			SWREG				
3283	016156	005015	043536	000	\$.CNTG: .ASCIZ	<15><12>/^G/			
3284	016163	015	051412	051127	\$.MSWR: .ASCIZ	<15><12>/SWR= /			
3285	016170	020075	000		\$.MNEW: .ASCIZ	/ NEW= /			
3286	016173	040	047040	053505					
3287	016200	020075	000		.EVEN				
3288		016204			MTITLE: .ASCIZ	<15><12><12>/DU11 CZDUC-D TAPE C /<15><12>			
3289	016204	005015	042012	030525					
3290	016212	020061	055103	052504					
3291	016220	026503	020104	040524					
3292	016226	042520	041440	006440					
3293	016234	000012							
3294	016236	005015	042526	052103	MVECTO: .ASCIZ	<15><12>/VECTOR ADDRESS-/			
3295	016244	051117	040440	042104					
3296	016252	042522	051523	000055					
3297	016260	005015	051461	020124	MREGAD: .ASCIZ	<15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/			
3298	016266	042504	044526	042503					
3299	016274	020072	042522	042503					
3300	016302	053111	051105	041440					
3301	016310	047117	051124	046117					
3302	016316	051040	043505	051511					
3303	016324	042524	020122	042101					
3304	016332	051104	051505	026523					
3305	016340	000							
3306	016341	015	040412	042522	MMULT: .ASCIZ	<15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/			
3307	016346	054440	052517	051040					
3308	016354	047125	044516	043516					
3309	016362	046440	046125	044524					
3310	016370	046120	020105	042504					
3311	016376	044526	042503	020123					
3312	016404	020077	054450	047440					
3313	016412	020122	024516	000055					
3314	016420	005015	040514	052123	MLASTD: .ASCIZ	<15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/			
3315	016426	042040	053105	041511					
3316	016434	035105	042522	042503					

3317	016442	053111	051105	041440	
3318	016450	047117	051124	046117	
3319	016456	051040	043505	051511	
3320	016464	042524	020122	042101	
3321	016472	051104	051505	026523	
3322	016500	000			
3323	016501	075	042504	044526	DEVICE: .ASCIZ /=DEVICE /
3324	016506	042503	020040	000	
3325	016513	015	044012	053517	MCOW: .ASCIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN @ACTREG/
3326	016520	047040	053517	041040	
3327	016526	047522	047127	041440	
3328	016534	053517	020077	027056	
3329	016542	051456	046105	041505	
3330	016550	020124	047523	042515	
3331	016556	044124	047111	020107	
3332	016564	047524	051040	047125	
3333	016572	040040	041501	051124	
3334	016600	043505	000		
3335	016603	015	047412	052125	MRANGE: .ASCIZ <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/
3336	016610	047440	020106	040522	
3337	016616	043516	035105	042522	
3338	016624	054524	042520	046040	
3339	016632	051501	020124	042504	
3340	016640	044526	042503	051040	
3341	016646	041530	051123	040440	
3342	016654	042104	042522	051523	
3343	016662	000055			
3344	016664	020040	000077		MQM: .ASCIZ / ?/
3345	016670	005015	000		MCRLF: .ASCIZ <15><12>
3346	016673	040	050040	053517	MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
3347	016700	051105	043040	044501	
3348	016706	052514	042522	020054	
3349	016714	051120	043517	040522	
3350	016722	020115	042522	052123	
3351	016730	051101	020124	052101	
3352	016736	052040	051505	020124	
3353	016744	047111	050040	047522	
3354	016752	051107	051505	000123	
3355	016760	005015	047105	020104	MEPASS: .ASCIZ <15><12>/END OF PASS TAPE C/
3356	016766	043117	050040	051501	
3357	016774	020123	040524	042520	
3358	017002	041440	000		
3359	017005	015	051012	000	MR: .ASCIZ <15><12>/R/
3360	017011	015	052012	051505	MTSTPC: .ASCIZ <15><12>/TEST PC-/
3361	017016	020124	041520	000055	
3362	017024	005015	047514	045503	MLOCK: .ASCIZ <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
3363	017032	047440	020116	042523	
3364	017040	042514	052103	042105	
3365	017046	052040	051505	037524	
3366	017054	024040	020131	051117	
3367	017062	047040	026451	000	
3368	017067	015	042012	020125	MLEVEL: .ASCIZ <15><12>/DU PRIORITY LEVEL-/
3369	017074	051120	047511	044522	
3370	017102	054524	046040	053105	
3371	017110	046105	000055		
3372	017114	005015	020043	043117	MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/


```

3373 017122 051440 047131 020103
3374 017130 044103 051101 020123
3375 017136 042523 042514 052103
3376 017144 042105 024040 030440
3377 017152 047440 020122 024462
3378 017160 000055
3379 017162 005015 051511 051440
3380 017170 041505 054040 044515
3381 017176 020124 052512 050115
3382 017204 051105 021440 020066
3383 017212 047111 020077 054450
3384 017220 047440 020122 024516
3385 017226 000055
3386 017230 005015 051511 051440
3387 017236 041505 051040 041505
3388 017244 045040 046525 042520
3389 017252 020122 032443 044440
3390 017260 037516 024040 020131
3391 017266 051117 047040 026451
3392 017274 000
3393 017275 015 044412 020123
3394 017302 050117 020124 046103
3395 017310 020122 047105 041101
3396 017316 042514 045040 046525
3397 017324 042520 020122 032043
3398 017332 044440 037516 024040
3399 017340 020131 051117 047040
3400 017346 026451 000
3401 017351 015 044412 020123
3402 017356 044124 020105 042524
3403 017364 052123 041440 047117
3404 017372 042516 052103 051117
3405 017400 044440 051516 040524
3406 017406 046114 042105 037440
3407 017414 054450 047440 020122
3408 017422 024516 000055
3409 017426 006412 020040 020040
3410 017434 052123 052101 051525
3411 017442 020040 046440 050101
3412 017450 020040 020040 005040
3413 017456 000015
3414
3415
3416
3417
3418 017460 000040
3419 017520 000040
3420 017560 000040
3421
3422
3423
3424
3425
3426 017620 006367 000044
3427 017624 006367 000040
3428 017630 006367 000034

```

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

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

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

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

MSTATUS: .ASCIZ <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  ;

```

```

3429 017634 006367 000030      ASL      DUPRT      :
3430 017640 006367 000024      ASL      DUPRT      :
3431 017644 016767 000020 000020  MOV      DUPRT,LESS1  :MOVE THIS TO LESS1
3432 017652 162767 000001 000012  SUB      #1,LESS1     :CREATE LESS1
3433 017660 042767 000037 000004  BIC      #37,LESS1    :CLEAR TNZVC
3434 017666 000207      RTS      PC
3435 017670 000240      DUPRT:  LEVEL5
3436 017672 000200      LESS1:  LEVEL4  ;LEVEL TO ALLOW INTERRUPTS
3437
3438      ;NEW DU ADDRESSES
3439 017674 016767 000126 000422  DUADDR:  MOV      DUBASE,RXCSR  ;XXX0
3440 017702 005267 000120      INC      DUBASE
3441 017706 016767 000114 000412  MOV      DUBASE,HRXCSR  ;XXX1
3442 017714 005267 000106      INC      DUBASE
3443 017720 016767 000102 000402  MOV      DUBASE,RXDBUF  ;XXX2
3444 017726 016767 000074 000400  MOV      DUBASE,PARCSR  ;XXX2
3445 017734 005267 000066      INC      DUBASE
3446 017740 016767 000062 000364  MOV      DUBASE,HRXDBUF ;XXX3
3447 017746 016767 000054 000362  MOV      DUBASE,HPARCSR ;XXX3
3448 017754 005267 000046      INC      DUBASE
3449 017760 016767 000042 000352  MOV      DUBASE,TXCSR   ;XXX4
3450 017766 005267 000034      INC      DUBASE
3451 017772 016767 000030 000342  MOV      DUBASE,HTXCSR  ;XXX5
3452 020000 005267 000022      INC      DUBASE
3453 020004 016767 000016 000332  MOV      DUBASE,TXDBUF  ;XXX6
3454 020012 005267 000010      INC      DUBASE
3455 020016 016767 000004 000322  MOV      DUBASE,HTXDBUF ;XXX7
3456 020024 000207      RTS      PC
3457 020026 000000      DUBASE: 0
3458
3459      ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3460      ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3461      ;SHIFTED IN BY THE CONTENTS OF SHIFT
3462 020030 042777 040000 000302  RPOKE:  BIC      #MTDATA,@TXCSR
3463 020036 005067 161104      CLR      TEMP2
3464 020042 006067 161076      ROR      TEMP1      ;FORCE CARRY
3465 020046 006067 161074      ROR      TEMP2      ;PICK UP CARRY IN BIT 15
3466 020052 006267 161070      ASR      TEMP2      ;SHIFT INTO BIT 14
3467 020056 042767 100000 161062  BIC      #BIT15,TEMP2  ;CLR BIT 15
3468 020064 056777 161056 000246  BIS      TEMP2,@TXCSR  ;POKE MAINT DATA
3469 020072 042777 020000 000240  BIC      #CLK,@TXCSR   ;POKE CLK
3470 020100 052777 020000 000232  BIS      #CLK,@TXCSR   ;
3471 020106 005367 161026      DEC      SHIFT
3472 020112 001346      BNE      RPOKE
3473 020114 000207      RTS      PC
3474      ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3475 020116 016767 161022 161022  ODD8:  MOV      TEMP1,TEMP2  ;SAVE TEMP1
3476 020124 005067 161020      CLR      TEMP3
3477 020130 012727 000010      MOV      #8.,(PC)+
3478 020134 000000      4$:    0
3479 020136 006067 161004      1$:    ROR      TEMP2
3480 020142 005567 161002      ADC      TEMP3
3481 020146 005367 177762      DEC      4$
3482 020152 001371      BNE      1$
3483 020154 006067 160770      ROR      TEMP3
3484 020160 103404      BCS     2$

```



```

3485 020162 052767 000400 160754      BIS    #BIT8,TEMP1      ;SET ODD PARITY
3486 020170 000403          BR      3$
3487 020172 042767 000400 160744  2$:    BIC    #BIT8,TEMP1      ;CLR EVEN PARITY
3488          :TEMP1 NOW HAS ODD PARITY CHARACTER
3489 020200 000207          3$:    RTS     PC
3490
3491          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3492 020202 016767 160736 160736  EVEN8: MOV    TEMP1,TEMP2      ;SAVE TEMP1
3493 020210 005067 160734          CLR    TEMP3
3494 020214 012727 000010          MOV    #8.,(PC)+
3495 020220 000000          4$:    0
3496 020222 006067 160720          1$:    ROR    TEMP2
3497 020226 005567 160716          ADC    TEMP3
3498 020232 005367 177762          DEC    4$
3499 020236 001371          BNE    1$
3500 020240 006067 160704          ROR    TEMP3
3501 020244 103004          BCC    2$
3502 020246 052767 000400 160670      BIS    #BIT8,TEMP1      ;SET EVEN PARITY
3503 020254 000403          BR      3$
3504 020256 042767 000400 160660  2$:    BIC    #BIT8,TEMP1      ;CLR ODD PARITY
3505          :TEMP1 NOW HAS EVEN PARITY CHARACTER
3506 020264 000207          3$:    RTS     PC
3507
3508 020266 062716 000002      TRPREG: ADD    #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3509          ;IN MAIN PART OF THE PROGRAM
3510 020272 000002          RTI
3511          ;ERROR HLT TABLE
3512 020274 020360      .ERRTAB: EM0    ;HLT 0 BIT ERROR (GENERAL)
3513 020276 000000          0
3514 020300 000000          0
3515 020302 020374          EM1    ;HLT 1 REGISTER ERROR
3516 020304 020545          DH1
3517 020306 020566          DT1
3518 020310 020436          EM2    ;HLT 2 RECEIVER ERROR
3519 020312 020545          DH1
3520 020314 020566          DT1
3521 020316 020500          EM3    ;HLT 3 TRANSMITTER ERROR
3522 020320 020545          DH1
3523 020322 020566          DT1
3524          ;DEFAULT DU ADDRESSES
3525 020324 160040      RXCSR: 160040
3526 020326 160041      HRXCSR: 160041
3527 020330 160042      RXDBUF: 160042
3528 020332 160043      HRXDBUF: 160043
3529 020334 160042      PARCSR: 160042
3530 020336 160043      HPARCSR: 160043
3531 020340 160044      TXCSR: 160044
3532 020342 160045      HTXCSR: 160045
3533 020344 160046      TXDBUF: 160046
3534 020346 160047      HTXDBUF: 160047
3535          ;DEFAULT DU VECTORS
3536 020350 000770      DURIV: 770    ;REC INTR VECTOR
3537 020352 000772      DURIS: 772    ;REC INTR STATUS
3538 020354 000774      DUTIV: 774    ;XMIT INTR VECTOR
3539 020356 000776      DUTIS: 776    ;XMIT INTR STATUS
3540          ;ERROR MESSAGES

```

3541	020360	036440	042440	051122	EM0:	.ASCIZ / = ERROR PC/
3542	020366	051117	050040	000103		
3543	020374	036440	051040	043505	EM1:	.ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3544	020402	051511	042524	020122		
3545	020410	051105	047522	020122		
3546	020416	041520	005015	051001		
3547	020424	043505	051511	042524		
3548	020432	020122	000040			
3549	020436	036440	051040	041505	EM2:	.ASCIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
3550	020444	044505	042526	020122		
3551	020452	051105	047522	020122		
3552	020460	041520	005015	051001		
3553	020466	043505	051511	042524		
3554	020474	020122	000040			
3555	020500	036440	052040	040522	EM3:	.ASCIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3556	020506	051516	044515	052124		
3557	020514	051105	042440	051122		
3558	020522	051117	050040	006503		
3559	020530	000412	042522	044507		
3560	020536	052123	051105	020040		
3561	020544	000				
3562						
3563	020545	105	050130	041505	DH1:	:DATA HEADERS FOR ERROR MESSAGES .ASCIZ /EXPECTED ACTUAL/
3564	020552	042524	020104	040440		
3565	020560	052103	040525	000114		
3566						
3567					.EVEN	
3568	020566	000003			DT1:	:DATA TABLES FOR ERROR MESSAGES 3
3569	020570	006	004			.BYTE 6,4
3570	020572	001164				SAVR3 :REGISTER
3571	020574	006	004			.BYTE 6,4
3572	020576	001156				SAVR0 :EXPECTED DATA
3573	020600	006	002			.BYTE 6,2
3574	020602	001160				SAVR1 :ACTUAL DATA
3575		000001			.END	

AAA	002520	1178#																		
ACTREG	001216	898#	1062*	1126*	1140*	1141*	1148*	1230	2826	2829	2839									
ADDBAS	001306	938#	1058	1225*																
ADDKEE	001310	939#	1059	1226*																
ADDLAS	001312	940#	1060	1227*																
ADDR0T	001322	944#	1063	1231*																
ADRCNT=	014775	2991*	3030*	3039#																
A10010=	***** GX	677																		
BASEAD	001204	893#	1058*	1065*	1108*	1145*	1146	1152*	1154*	1225	2833*	2845*	2849							
BASEDU	001364	961#	1072	1238*																
BASEIV	001212	896#	1064*	1118*	1229	2834*	2846*	2851	2852*	2853	2854*	2855	2856*	2857						
BBB	002336	2858*																		
BINWRD	015312	1125	1129#																	
BITW =	002000	3080*	3081	3118#																
BIT0 =	000001	808#																		
BIT1 =	000002	756#	784	815																
BIT10 =	002000	755#	783																	
BIT11 =	004000	746#	774	808	1851	1896														
BIT12 =	010000	745#	773																	
BIT13 =	020000	744#	772	789																
BIT14 =	040000	743#	771	788	807															
BIT15 =	100000	742#	770	787	806															
BIT2 =	000004	741#	769	786	805	3467														
BIT3 =	000010	754#	782	1293																
BIT4 =	000020	753#	781	814																
BIT5 =	000040	752#	780	813																
BIT6 =	000100	751#	779	812																
BIT7 =	000200	750#	778	811																
BIT8 =	000400	749#	777	810																
BIT9 =	001000	748#	776	792	809	3485	3487	3502	3504											
BREAK =	000001	747#	775	791																
		815#	1336	1376	1416	1456	1495	1538	1581	1624	1664	1707	1750	1793						
		1832	1877	1922	1961	2001	2038	2075	2112	2149	2183	2217	2251	2285						
		2322	2359	2396	2443	2540	2640	2740												
BRW	014326	1290	1304	2919#																
BRX	014330	1291	1305	2920#																
BUFF1	000002R	677#	1267	1273																
BUFHRX	001346	954#	1076	1242*																
BUFHTX	001362	960#	1082	1248*																
BUFRXD	001344	953#	1075	1241*																
BUFTXD	001360	959#	1081	1247*																
BYJMR	001305	934#	1057	1224*																
CARDET=	010000	772#																		
CCC	014062	2825	2848	2865#																
CHRCNT	015310	3078*	3082	3096*	3116#	3117														
CKSWR =	104413	993#	2878	2900	2924	3153	3196													
CLK =	020000	807#	1336	1342	1343	1345	1346	1376	1382	1383	1385	1386	1416	1422						
		1423	1425	1426	1456	1462	1463	1465	1466	1495	1501	1502	1504	1505						
		1538	1544	1545	1547	1548	1581	1587	1588	1590	1591	1624	1630	1631						
		1633	1634	1664	1670	1671	1673	1674	1707	1713	1714	1716	1717	1750						
		1756	1757	1759	1760	1793	1799	1800	1802	1803	1832	1838	1839	1841						
		1842	1877	1883	1884	1886	1887	1922	1928	1929	1961	1967	1968	2001						
		2007	2008	2010	2011	2038	2044	2045	2047	2048	2075	2081	2082	2084						
		2085	2112	2118	2119	2121	2122	2149	2155	2156	2183	2189	2190	2217						
		2223	2224	2251	2257	2258	2285	2291	2292	2294	2295	2322	2328	2329						
		2331	2332	2359	2365	2366	2368	2369	2396	2402	2403	2405	2406	2443						

MCRLF	016670	1277	2960	2976	3068	3277	3345#							
MDATA	017560	3094	3105	3420#										
MEPASS	016760	2821	3355#											
MEXT =	010000	819#												
MEXTJ	017351	1205	3401#											
MINT =	004000	818#	1336	1376	1416	1456	1495	1538	1581	1624	1664	1707	1750	1793
		1832	1877	1922	1961	2001	2038	2075	2112	2149	2183	2217	2251	2285
		2322	2359	2396	2443	2540	2640	2740						
MITSEX	001301	930#	1053	1220*										
MLASTD	016420	1131	3314#											
MLEVEL	017067	1166	3368#											
MLOCK	017024	1296	3362#											
MMULT	016341	1120	3306#											
MPFAIL	016673	3242	3346#											
MQM	016664	2973	3344#											
MR	017005	1318	3359#											
MRANGE	016603	1156	3335#											
MREGAD	016260	1099	3297#											
MRESET=	000400	809#	1331	1333	1371	1373	1411	1413	1451	1453	1490	1492	1533	1535
		1576	1578	1619	1621	1659	1661	1702	1704	1745	1747	1788	1790	1827
		1829	1872	1874	1917	1919	1956	1958	1996	1998	2033	2035	2070	2072
		2107	2109	2144	2146	2178	2180	2212	2214	2246	2248	2280	2282	2317
		2319	2354	2356	2391	2393	2438	2440	2535	2537	2635	2637	2735	2737
MSTATU	017426	1262	3409#											
MSYNC	017114	1180	3372#											
MTDATA=	040000	806#	1336	1376	1416	1456	1495	1538	1581	1624	1664	1707	1750	1793
		1832	1877	1922	1961	2001	2038	2075	2112	2149	2183	2217	2251	2285
		2322	2359	2396	2443	2540	2640	2740	3462					
MTITLE	016204	1021	3289#											
MTSTPC	017011	1309	3360#											
MULTD	001202	888#	1056*	1122	1123	1223	1287	2824						
MVECTO	016236	1110	3294#											
MWIRE4	017275	1201	3393#											
MWIRE5	017230	1197	3386#											
MWIRE6	017162	1193	3379#											
NEXT	001116	856#	1330*	1370*	1410*	1450*	1489*	1532*	1575*	1618*	1658*	1701*	1744*	1787*
		1826*	1871*	1916*	1955*	1995*	2032*	2069*	2106*	2143*	2177*	2211*	2245*	2279*
		2316*	2353*	2390*	2437*	2534*	2634*	2734*	2896	2916	3202			
		801#	1339	1379	1419	1459	2004	2041	2078	2115	2152	2186	2220	2254
NOPAR =	000000	2288	2325	2362	2399									
		929#	1052	1219*										
NOSYNC	001300	802#	1667	1710	1753	1796	1880	1964						
ODDPAR=	001000	1893	1974	3475#										
ODD8	020116	1020	1023#											
ONCE	001522	887#	1055*	1203	1222									
OPTCLR	001201	3258	3260	3262	3266	3278#								
OUT	016142	2822	2861#											
OUTCRY	014054	1128	1153	1164#										
OUTMUL	002476	787#	2504	2604	2704	2304								
OVRRUN=	040000	981#	1100	1111	1132	1157	1167	1310	3272					
PARAM =	104405	2993#	3014											
PARAM1	014622	1077*	1243	1332*	1339*	1372*	1379*	1412*	1419*	1452*	1459*	1491*	1498*	1534*
PARCSR	020334	1541*	1577*	1584*	1620*	1627*	1660*	1667*	1703*	1710*	1746*	1755*	1789*	1796*
		1828*	1835*	1873*	1880*	1918*	1925*	1957*	1964*	1997*	2004*	2034*	2041*	2071*
		2078*	2108*	2115*	2145*	2152*	2179*	2186*	2213*	2220*	2247*	2254*	2281*	2288*
		2318*	2325*	2355*	2362*	2392*	2399*	2439*	2446*	2536*	2543*	2636*	2643*	2736*

HLT	738#	1356	1396	1436	1476	1513	1518	1556	1561	1599	1604	1642	1647	1682	1687
	1725	1730	1768	1773	1811	1816	1857	1902	1941	1980	2019	2056	2093	2130	2164
	2198	2232	2266	2303	2340	2377	2414	2462	2473	2483	2488	2559	2570	2580	2585
	2659	2670	2680	2685	2759	2770	2780	2785							
PRGEND	677#	2813													
PRGFRT	677#	678													
PUSSYF	677#														
RSETUP	677#	1331	1371	1411	1451	1490	1533	1576	1619	1659	1702	1745	1788	1827	1872
	1917	1956	1996	2033	2070	2107	2144	2178	2212	2246	2280	2317	2354	2391	2438
	2535	2635	2735												
TSETUP	677#														
\$BEGIN	677#	1208													
\$BINAR	677#														
\$BUFFE	677#	3415													
\$CABLE	677#														
\$CATCH	677#	821													
\$CLRVE	677#	1023													
\$CONVR	677#	3064													
\$DNA	677#														
\$EOP	677#	2813													
\$GETFL	677#	1119	1192	1196	1200	1204	1295								
\$GETPA	677#	1098	1109	1129	1155	1164	1308								
\$GETSY	677#	1174													
\$HEADE	677#	678													
\$HLT	677#	3150													
\$INSTR	677#	2943													
\$ISOB	677#	1819	1864	1909	1948										
\$MATCH	677#														
\$MRR	677#														
\$MRRW	677#														
\$MRW	677#														
\$MSG	677#	3289													
\$PARAM	677#	2981													
\$PFAIL	677#	3209													
\$POKE	677#														
\$POKER	677#	1344	1384	1424	1464	1503	1546	1589	1632	1672	1715	1758	1801	1840	1885
	1927	1966	2009	2046	2083	2120	2154	2188	2222	2256	2293	2330	2367	2404	2453
	2550	2650	2750												
\$RCNET	677#	2419	2513	2613	2713										
\$RECAC	677#														
\$REG	677#	3040													
\$RESET	677#	1331	1333	1371	1373	1411	1413	1451	1453	1490	1492	1533	1535	1576	1578
	1619	1621	1659	1661	1702	1704	1745	1747	1788	1790	1827	1829	1872	1874	1917
	1919	1956	1958	1996	1998	2033	2035	2070	2072	2107	2109	2144	2146	2178	2180
	2212	2214	2246	2248	2280	2282	2317	2319	2354	2356	2391	2393	2438	2440	2535
	2537	2635	2637	2735	2737										
\$RXACT	677#														
\$SCOPE	677#	2881													
\$SCOP1	677#	2921													
\$SETFL	677#	3119													
\$SETVE	677#	822													
\$START	677#	999													
\$STRIP	677#	1987	2024	2061	2098	2135	2169	2203	2237	2271	2308	2345	2382		
\$SYMBO	677#	690													
\$SYNCR	677#	1340	1380	1420	1460	1499	1542	1585	1628	1668	1711	1754	1797	1836	1881
	2005	2042	2079	2116	2289	2326	2363	2400	2449	2546	2646	2746			

CZDUC-D
CZDUCD.P11

MACY11 30A(1052)
18-JUL-78 15:52

21-NOV-78 15:40 PAGE 84

B 7

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0079

STRAPS	677#	963													
STRPAR	677#														
STRPDE	677#	971	973	975	977	979	981	983	985	987	989	991	993	995	
STRPSR	677#	3137													
STSTNO	677#	1329	1369	1409	1449	1488	1531	1574	1617	1657	1700	1743	1786	1825	1870
	1915	1954	1994	2031	2068	2105	2142	2176	2210	2244	2278	2315	2352	2389	2436
	2533	2633	2733												
STYPE	677#	2929													
SUNIBU	677#														
SVARIA	677#	841													
SWORDF	677#	1321	1361	1401	1441										
SWORDO	677#														
SWORDP	677#	1481	1524	1567	1610	1650	1693	1736	1779						

. ABS. 020604 000

ERRORS DETECTED: 0

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

RUN-TIME: 8 12 1 SECONDS

RUN-TIME RATIO: 100/22=4.4

CORE USED: 19K (37 PAGES)