

# DU11

OFF-LINE TIMING & INTER  
CZDUED0

AH-8693D-MC

COPYRIGHT ©73-78

FICHE 1 OF 1

JAN 1979

**digital**

MADE IN USA

The image shows a microfiche card with a grid of 12 columns and 12 rows of small, illegible data tables or charts. The data is too faint to be transcribed accurately, but it appears to be organized in a structured format, possibly representing timing and interconnect data for the DU11 component.



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

-----  
PRODUCT CODE: AC-8692D-MC

PRODUCT NAME: CZDUED0 DU11 OFFLINE TIMING & INTERRUPT 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 OR 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 COMBINED TIMING AND INTERRUPT TESTS VERIFY THAT THE TRANSMITTER AND RECEIVER CAN CORRECTLY TALK TO EACH OTHER. INTERRUPT LOGIC AND PRIORITY LEVEL /VECTOR ADDRESSES ARE ALSO VERIFIED

2. REQUIREMENTS

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

DJ11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE

THE PROGRAM LOADS AND RUNS IN 4K OF MEMORY.

3. LOADING PROCEDURE

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

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

.....

NOTE: BEFORE PROCEEDING IT IS IMPORTANT TO  
TO REALIZE IF ONE DOESNOT  
HAVE THE DU11 SET UP TO THE

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

\*\*\*\*\*

4.1 CONTROL SWITCH SETTINGS

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

4.1.1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)  
ALL CONSOLE SWITCHES DOWN

4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES  
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)  
SW01-1

4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)

SW02-1

NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02-1 IS USED  
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200  
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200  
THE STARTING ADDRESS TO LOCK ON TEST IS 000200

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE 'DU11 CZDUE-D TAPE E' (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 4.3.3.2

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



#### 4.4 STATUS MAP

---

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

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

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

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

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

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

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

THE MAP WILL LOOK LIKE:

STATUS MAP

1300/ 177777

1302/ 000000

1304/ 177777

THE BYTES ARE DEFINED AS FOLLOWS:

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

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

#### 5. OPERATING PROCEDURE

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

#### CONTROL :

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

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

#### 5.1

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

SW11 =1 INHIBIT ITERATIONS  
SW10 1 ESCAPE TO NEXT TEST ON ERROR  
SW08 -1 LOOP ON ERROR  
SW07 -1 USE STATUS MAP PARAMETERS  
SW02 1 LOCK ON TEST  
SW01 =1 RESTART PROGRAM AT SELECTED TEST  
SW00 =1 RESELECT VECTOR AND CONTROL REGISTER ADDRESSES  
&PARAMETERS AFTER A PROGRAM RESTART  
TO INHIBIT 'END OF PASS' TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS  
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

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

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

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

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

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

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

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

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

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS  
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y  
16XXXX - DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

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

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



7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

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

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

7.2.2.1 TO RESTART...LOAD 000200 IN SWR AND DEPRESS START....  
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.

7.2.2.2 .....OR .....LOAD 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)

P. DEFAULT PARAMETERS:  
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160040

VECTOR ADDRESS- DURIV: 770

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0

LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0

DJ PRIORITY LEVEL- LEVEL 5 DUFRT: 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 SEPEC: 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 COMBINED (TRANSMITTER & RECEIVER)  
TIMING & INTERRUPT TESTING OF THE DEVICE  
SEE LISTING FOR DETAILS

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

11. LISTINGS

676  
677 000000' 000000G

D

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



```
712
713           ;REGISTER DEFINITIONS
714
715           000000      R0=%0           ;GENERAL REGISTER
716           000001      R1=%1           ;GENERAL REGISTER
717           000002      R2=%2           ;GENERAL REGISTER
718           000003      R3=%3           ;GENERAL REGISTER
719           000004      R4=%4           ;GENERAL REGISTER
720           000005      R5=%5           ;GENERAL REGISTER
721           000006      SP=%6          ;PROCESSOR STACK POINTER
722           000007      PC=%7          ;PROGRAM COUNTER
723
724           ;LOCATION EQUIVALENCIES
725
726           177570      DSWR=177570     ;HARDWARE SWITCH REGISTER LOC.
727           177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
728           177776      PS=177776      ;PROCESSOR STATUS WORD
729           001100      STACK=1100     ;START OF PROCESSOR STACK
730
731           ;INSTRUCTION DEFINITIONS
732
733           005746      PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD -TST -(SP)
734           005726      POP1SP=5726    ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
735           010046      PUSHRO=10046    ;SAVE R0 ON STACK =MOV R0,-(SP)
736           012600      POPRO=12600    ;RESTORE R0 FROM STACK =MOV (SP)+,R0
737           024646      PUSH2SP=24646  ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
738           022626      POP2SP=22626   ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
739           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
740
741
742           100000      BIT15=100000
743           040000      BIT14=40000
744           020000      BIT13=20000
745           010000      BIT12=10000
746           004000      BIT11=4000
747           002000      BIT10=2000
748           001000      BIT9=1000
749           000400      BIT8=400
750           000200      BIT7=200
751           000100      BIT6=100
752           000040      BIT5=40
753           000020      BIT4=20
754           000010      BIT3=10
755           000004      BIT2=4
756           000002      BIT1=2
757           000001      BIT0=1
758
759           ;PROCESSER LEVELS
760           000340      LEVEL7=340
761           000300      LEVEL6=300
762           000240      LEVEL5=240
763           000200      LEVEL4=200
764           000140      LEVEL3=140
765           000100      LEVEL2=100
766           000040      LEVEL1=040
767           000000      LEVEL0=000
```

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

```

823                                     :STANDARD INTERRUPT VECTORS
824
825
826                                     .=24
827 000024 015520                       .PFAIL                       :POWER FAIL HANDLER
828 000026 000340                       340                          :SERVICE AT LEVEL 7
829 000030 015250                       .HLT                          :ERROR HANDLER
830 000032 000340                       340                          :SERVICE AT LEVEL 7
831 000034 015216                       .TRPSRV                       :GENERAL HANDLER DISPATCH SERVICE
832 000036 000340                       340                          :SERVICE AT LEVEL 7
833
834                                     :SOFTWARE SWITCH REGISTER
835
836                                     .=174
837 000174 000000                       DISPREG: .WORD 0              :SOFTWARE DISPLAY REG.
838 000176 000000                       SWREG:  .WORD 0              :SOFTWARE SWITCH REGISTER
839 000200 000167 001214                 JMP      .START              :GO TO START OF PROGRAM
840
841
842
843                                     . 1100
844
845                                     :INDIRECT POINTERS
846
847 001100 177570                       SWR: 177570                   :SWITCH REGISTER POINTER
848 001102 177570                       LIGHTS:177570                 :DISPLAY REGISTER POINTER
849 001104 177560                       TKCSR: 177560                 :TELETYPE KEYBOARD CONTROL REGISTER
850 001106 177562                       TKDBR: 177562                 :TELETYPE KEYBOARD DATA BUFFER
851 001110 177564                       TPCSR: 177564                 :TELEPRINTER CONTROL REGISTER
852 001112 177566                       TPDBR: 177566                 :TELEPRINTER DATA BUFFER
853
854                                     :PROGRAM CONTROL PARAMETERS
855
856 001114 000000                       RTRN: 0                       :SCOPE ADDRESS FOR LOOP ON TEST
857 001116 000000                       NEXT: 0                       :ADDRESS OF NEXT TEST TO BE EXECUTED
858 001120 000000                       LOCK: 0                       :ADDRESS FOR LOCK ON CURRENT DATA
859 001122 000000                       ICOUNT: 0                     :NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
860 001124 000000                       LPCNT: 0                     :NUMBER OF ITERATIONS COMPLETED
861 001126 000000                       TSTNO: 0                     :NUMBER OF TEST IN PROGRESS
862 001130 000000                       PASCNT: 0                    :NUMBER OF PASSES COMPLETED
863 001132 000000                       ERRCNT: 0                    :TOTAL NUMBER OF ERRORS
864 001134 000000                       LSTERR: 0                    :PC OF LAST ERROR CALL
865
866                                     :PROGRAM VARIABLES
867
868 001136 000020                       HOLD: 20                     :TEMPORARY STORAGE DELAY TIME FOR CABLES
869 001140 000000                       SHIFT: 0                     :TEMPORARY STORAGE= # OF SHIFTS PER CHAR
870 001142 000000                       COUNT: 0                     :TEMPORARY STORAGE- # OF TIMES A CHAR WILL BE SENT
871 001144 000000                       TEMP1: 0                     :TEMPORARY STORAGE
872 001146 000000                       TEMP2: 0                     :TEMPORARY STORAGE
873 001150 000000                       TEMP3: 0                     :TEMPORARY STORAGE
874 001152 000000                       TEMP4: 0                     :TEMPORARY STORAGE
875 001154 000000                       TEMP5: 0                     :TEMPORARY STORAGE
876 001156 000000                       SAVR0: C                     :R0 STORAGE
877 001160 000000                       SAVR1: 0                     :R1 STORAGE
878 001162 000000                       SAVR2: C                     :R2 STORAGE

```

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

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

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



```

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

```

940	001310	000001	ADDKEEP: .BLKW 1	:SAVED 1ST DEVICE ADDR
941	001312	000001	ADDLAST: .BLKW 1	:LAST DEVICE RXCSR ADDR
942	001314	000001	IVBASE: .BLKW 1	:PROG CONTROLLED IV
943	001316	000001	IVKEEP: .BLKW 1	:SAVED INTR VECTOR
944	001320	000001	REGACT: .BLKW 1	:ACTIVE REGISTER
945	001322	000001	ADDR0T: .BLKW 1	:ROTATING POINTER
946	001324	000001	PRTDU: .BLKW 1	:DU11 PRIORITY
947	001326	000001	RIVDU: .BLKW 1	:DU11 REC INTR VECTOR
948	001330	000001	TIVDU: .BLKW 1	:DU11 XMIT INTR VECTOR
949	001332	000001	TISDU: .BLKW 1	:DU11 XMIT INTR STATUS
950	001334	000001	RISDU: .BLKW 1	:DU11 REC INTR STATUS
951	001336	000001	LESS: .BLKW 1	:PRIORITY TO ALLOW INTR
952	001340	000001	CSRRX: .BLKW 1	: DEFAULT OR ALTERED PARAMETERS
953	001342	000001	CSRHRX: .BLKW 1	:
954	001344	000001	BUFRXD: .BLKW 1	:
955	001346	000001	BUFHRXD: .BLKW 1	:
956	001350	000001	CSRPAR: .BLKW 1	:
957	001352	000001	CSRHPAR: .BLKW 1	:
958	001354	000001	CSRTX: .BLKW 1	:
959	001356	000001	CSRHTX: .BLKW 1	:
960	001360	000001	BUFTXD: .BLKW 1	:
961	001362	000001	BUFHTXD: .BLKW 1	:
962	001364	000001	BASEDU: .BLKW 1	:DU11 RXCSR BASE ADDR
963			.EVEN	

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

968			.TRPTAB:	
969	001366		:*****	
970			:*****	
971			:*****	
972		104400	.SCOPE	SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
973	001366	014002	.SCOPE1	SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
974		104401	.TYPE	TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
975	001370	014166	.INSTR	INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
976		104402	.INSTR	INSTR=TRAP+4 ;CALL TO INPUT ERROR HANDLER
977	001372	014206	.PARAM	PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
978		104403	.SAV05	SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
979	001374	014246	.RES05	RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
980		104404	.CONVRT	CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
981	001376	014364	.CONVRT	CONVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
982		104405	.SETFLG	SETFLG=TRAP+12 ;CALL TO FLAG SET ROUTINE
983	001400	014416	.CKSWR	CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TV
984		104406		
985	001402	014632		
986		104407		
987	001404	014672		
988		104410		
989	001406	014724		
990		104411		
991	001410	014730		
992		104412		
993	001412	015150		
994		104413		
995	001414	015664		

```

996          104414          CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
997 001416 015740          .CNTLU
998          :*****
999          :*****
1000
1001          ;PROGRAM INITIALIZATION
1002          ;LOCK OUT INTERRUPTS
1003          ;SET UP PROCESSOR STACK
1004          ;SET UP POWER FAIL VECTOR
1005          ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1006          ;TYPE TITLE MESSAGE
1007
1008 001420 012767 000340 176350 .START: MOV A340 PS ;LOCK OUT INTERRUPTS
1009 001426 012706 001100          MOV #S1, K, SP ;SET UP STACK
1010 001432 012737 015520 000024          MOV #.PFAIL, @#24 ;SET UP POWER FAIL VECTOR
1011 001440 005067 177460          CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
1012 001444 105067 177575          CLR STFLG ;CLEAR START FLAG
1013 001450 005067 177454          CLR PASCNT ;CLEAR PASS COUNT
1014 001454 105067 177566          CLR ERRFLG ;CLEAR ERROR FLAG
1015 001460 005067 177446          CLR ERRCNT ;CLEAR ERROR COUNT
1016 001464 005067 177444          CLR LSTERR ;CLEAR LAST ERROR POINTER
1017 001470 012767 000001 177430          MOV #1, TSTNO ;SET UP FOR TEST 1
1018 001476 012767 001420 177410          MOV #.START, RTRN ;SET UP FOR POWER FAIL BEFORE
1019          ;TESTING STARTS
1020 001504 105767 177534          TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1021 001510 001004          BNE ONCE
1022 001512 104402 016040          TYPE ,MTITLE ;TYPE TITLE MESSAGE
1023 001516 105167 177522          COMB INIFLG ;IF NOT SET FLAG AND DO
1024 001522 012767 177570 177350 ONCE: MOV #DSWR, SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1025 001530 012767 177570 177344          MOV #DLIGHTS, LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1026 001536 013746 000006          MOV @#6, -(SP) ;SAVE VECTORS
1027 001542 013746 000004          MOV @#4, -(SP)
1028 001546 012737 001566 000004          MOV #64$, @#4 ;SET UP FOR TIMEOUT
1029 001554 022777 177777 177316          CMP #-1, @SWR ;REFERENCE HARDWARE SWITCH REGISTER
1030 001562 001402          BEQ 65$
1031 001564 000407          BR 66$
1032 001566 022626          64$: CMP (SP)+, (SP)+ ;ADJUST STACK
1033 001570 012767 000176 177302          65$: MOV #SWREG, SWR ;POINT TO SOFTWARE SWITCH REG
1034 001576 012767 000174 177276          MOV #DISPREG, LIGHTS ;POINT TO SOFT DISPLAY REG
1035 001604 012637 000004          60$: MOV (SP)+, @#4 ;RESTORE VECTORS
1036 001610 012637 000006          MOV (SP)+, @#6
1037 001614 005737 000042          TST @#42 ;UNDER MONITOR
1038 001620 001005          BNE MAP
1039 001622 022767 000176 177250          CMP #SWREG, SWR ;IS SWREG USED
1040 001630 001001          BNE MAP ; BRANCH TO CHECK FOR STATUS MAP
1041 001632 104414          CNTLU
1042          :*****
1043          ; CODE FOR STATUS MAP
1044          ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1045          ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1046          ; SETUP, OR REENTER QUESTIONING ROUTINE
1047
1048 001634 032777 000200 177236 MAP: BIT #SW07, @SWR ; IS SW07-1?
1049 001642 001537          BEQ $67 ; IF NOT, GO TO TEST FOR SW00-1
1050          ; NOW SET JP MAP VALUES FOR PROGRAM
1051          ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE
  
```

```

1052 ; OPERATION OF THIS PROGRAM.
1053 001644 116767 177430 177324 MOVB NOSYNC ,SYNCR0 ; SYNC CHAR
1054 001652 116767 177423 177317 MOVB MITSEX ,SEXMIT ; XMIT JUMPER
1055 001660 116767 177416 177312 MOVB RESEC ,SEREC ; SEC REC JUMPER
1056 001666 116767 177411 177305 MOVB CLROPT ,OPTCLR ; OPTIONAL JUMPER
1057 001674 116767 177404 177300 MOVB DMULT ,MULTD ; MULTIPLE DEVICE
1058 001702 116767 177377 177273 MOVB BYJMR ,JMRBY ; EXTERNAL MODEM
1059 001710 016767 177372 177266 MOV ADDBASE ,BASEADD ; PROG 1ST DEVICE ADDR
1060 001716 016767 177366 177262 MOV ADDKEEP ,KEEPADD ; SAVED 1ST DEVICE ADDR
1061 001724 016767 177362 177256 MOV ADDLAST ,LASTADD ; LAST DEVICE RXCSR ADDR
1062 001732 016767 177360 177254 MOV IVKEEP ,KEEPIV ; SAVED INIR VECTOR
1063 001740 016767 177354 177250 MOV REGACT ,ACTREG ; ACTIVE REGISTER
1064 001746 016767 177350 177244 MOV ADDR0T ,ROTADD ; ROTATING POINTER
1065 001754 016767 177334 177230 MOV IVBASE ,BASEIV ; BASE INTR VECTOR
1066 001762 016767 177220 177214 MOV KEEPADD ,BASEADD ; RELOAD BASEADD
1067 001770 016767 177332 016206 MOV RIVDU ,DURIV ; REC INTR VECTOR
1068 001776 016767 177332 016202 MOV RISDJ ,LIRIS ; REC INTR STATUS
1069 002004 016767 177320 016176 MOV TIVDU ,DUIV ; XMIT INTR VECTOR
1070 002012 016767 177314 016172 MOV TISDU ,DUTIS ; XMIT INTR STATUS
1071 002020 016767 177312 015500 MOV LESS ,LESS1 ; PRIORITY TO ALLOW INTR
1072 002026 013737 001324 017524 MOV @PRTDU ,@DUPRT ; PRIORITY RELOADED
1073 002034 016767 177324 015620 MOV BASEDU ,DUBASE ;
1074 002042 016767 177272 016110 MOV CSRRX ,RXCSR ;
1075 002050 016767 177266 016104 MOV CSRHRX ,HRXCSR ;
1076 002056 016767 177262 016100 MOV BUFRXD ,RXDBUF ;
1077 002064 016767 177256 016074 MOV BUFRXD ,HRXDBUF ;
1078 002072 016767 177252 016070 MOV CSRPAR ,PARCSR ;
1079 002100 016767 177246 016064 MOV CSRHPAR ,HPARCSR ;
1080 002106 016767 177242 016060 MOV CSRTX ,TXCSR ;
1081 002114 016767 177236 016054 MOV CSRHTX ,HTXCSR ;
1082 002122 016767 177232 016050 MOV BUFTXD ,TXDBUF ;
1083 002130 016767 177226 016044 MOV BUFTXD ,HTXDBUF ;
1084 002136 000167 000466 JMP .BEGIN ; BRANCH TO BEGIN TESTING
1085 *****
1086 002142 032777 000001 176730 $67: BIT #SW00 ,@SWR ; RESELECT VECTOR $ CONTROL REG?
1087 002150 001002 BNE 1$ ; BRANCH TO QUESTIONING
1088 002152 000167 000452 JMP .BEGIN ; GO TO LOAD STATUS MAP ETC.
1089 002156 005037 001222 1$: CLR @FLAG ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1090 002162 012700 000300 MOV #300,R0 ; RESTORE VECTOR AREA TO TRAPCATCHER
1091 002166 012701 000302 MOV #302,R1 ; START AT LOCATION 300
1092 002172 012702 000004 MOV #4,R2
1093 002176 010110 2$: MOV R1,(R0)
1094 002200 005011 CLR (R1)
1095 002202 060200 ADD R2,R0
1096 002204 060201 ADD R2,R1
1097 002206 022701 001000 CMP #1000,R1 ; END AT LOCATION 776
1098 002212 002771 BLT 2$
1099 002214 104403 INSTR ; OUTPUT MESSAGE & GET INPUT STRING
1100 002216 016114 MREGAD ; MESSAGE
1101 002220 104405 PARAM ; CONVERT STRING
1102 002222 160000 160000 ; LOW LIMIT
1103 002224 167776 167776 ; HIGH LIMIT
1104 002226 017662 DIBASE ; STORE AT THIS LOCATION
1105 002230 001 .BYTE 1 ; MASK
1106 002231 001 .BYTE 1 ; HOW MANY TIMES * 2
107 002232 016767 015424 176746 MOV DIBASE,KEEPADD ; SAVE

```



```

1108 002240 004767 015264 JSR PC,DUADDR
1109 002244 016767 176736 176732 MOV KEEPADD,BASEADD ;RESTORE FOR ROTATION
1110 002252 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1111 002254 016072 MVECTO ;MESSAGE
1112 002256 104405 PARAM ;CONVERT STRING
1113 002260 000300 300 ;LOW LIMIT
1114 002262 000776 776 ;HIGH LIMIT
1115 002264 020204 DURIV ;STORE AT THIS LOCATION
1116 002266 001 .BYTE 1 ;MASK
1117 002267 004 .BYTE 4 ;HOW MANY TIMES + 2
1118 002270 016767 015710 176716 MOV DURIV,KEEPIV ;SAVE
1119 002276 016767 015702 176706 MOV DURIV,BASEIV ;SET UP FOR ROTATION
1120 002304 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1121 002306 016175 MMULT ;MESSAGE
1122 002310 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1123 002312 001202 MULTD ;THIS FLAG
1124 002314 105767 176662 TSTB MULTD ;ARE THERE MULTIPLE DEVICES
1125 ;ON THE SYSTEM ?
1126 002320 100406 BMI BBB ;YES,ASK NEXT QUESTION
1127 002322 005067 176670 CLR ACTREG
1128 002326 005067 176666 CLR ROTADD
1129 002332 000167 000140 JMP OUTMUL ;JUMP AROUND NEXT QUESTION
1130 002336 BBB:
1131 002336 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1132 002340 016254 MLASTD ;MESSAGE
1133 002342 104405 PARAM ;CONVERT STRING
1134 002344 160000 160000 ;LOW LIMIT
1135 002346 167776 167776 ;HIGH LIMIT
1136 002350 001210 LASTADD ;STORE AT THIS LOCATION
1137 002352 001 .BYTE 1 ;MASK
1138 002353 001 .BYTE 1 ;HOW MANY TIMES + 2
1139 ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1140 002354 012767 000001 176636 1$: MOV #1,ROTADD ;SET UP POINTER
1141 002362 005067 176630 CLR ACTREG ;CLR ACTIVE REGISTER
1142 002366 056767 176626 176622 2$: BIS ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1143 002374 000241 CLC
1144 002376 006167 176616 ROL ROTADD ;SET UP POINTER
1145 002402 103421 BCS 3$ ;ARE YOU OUT OF RANGE ?
1146 002404 062767 000010 176572 ADD #10,BASEADD ;SET UP BASE ADDRESS
1147 002412 026767 176572 176564 CMP LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1148 002420 101362 BHI 2$ ;NO DO IT AGAIN
1149 002422 056767 176572 176566 BIS ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1150 ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1151 ;MULTIPLE DEVICE QUESTION
1152 002430 012767 000001 176562 4$: MOV #1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1153 002436 016767 176544 176540 MOV KEEPADD,BASEADD ;DITTO
1154 002444 000414 BR OUTMUL ;CONTINUE QUESTIONS
1155 002446 016767 176534 176530 3$: MOV KEEPADD,BASEADD ;RESTORE
1156 002454 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1157 002456 016437 MRANGE ;MESSAGE
1158 002460 104405 PARAM ;CONVERT STRING
1159 002462 160000 160000 ;LOW LIMIT
1160 002464 167776 167776 ;HIGH LIMIT
1161 002466 001210 LASTADD ;STORE AT THIS LOCATION
1162 002470 001 .BYTE 1 ;MASK
1163 002471 001 .BYTE 1 ;HOW MANY TIMES + 2

```

```

1164 002472 000167 177656          JMP      1$      ;DO IT AGAIN
1165 002476          OUTMUL:
1166 002476 104403          INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1167 002500 016723          MLEVEL     ;MESSAGE
1168 002502 104405          PARAM      ;CONVERT STRING
1169 002504 000004          4          ;LOW LIMIT
1170 002506 000007          7          ;HIGH LIMIT
1171 002510 017524          DUPRT      ;STORE AT THIS LOCATION
1172 002512          000          .BYTE     ;MASK
1173 002513          001          .BYTE     ;HOW MANY TIMES + 2
1174 002514 004767 014734          JSR      PC,DULEV
1175          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1176          ;BUFFER TO THE CHARACTERS '1' AND '2'
1177          ;IF THE CHARACTER IS '1' CLEAR THE FLAG
1178          ;IF THE CHARACTER IS '2' SET THE FLAG
1179 002520          AAA:
1180 002520 104403          INSTR      ;OUTPUT MESSAGE & GET IN-UT STRING
1181 002522 016750          MSYNC      ;MESSAGE
1182 002524 122767 000061 014562 3$:  CMPB      #'1',INBUF ;IS IT '1' ?
1183 002532 001003          BNE        1$
1184 002534 105067 176436          CLR      SYNCNO ;000
1185 002540 000412          BR         4$
1186 002542 122767 000062 014544 1$:  CMPB      #'2',INBUF ;IS IT '2' ?
1187 002550 001004          BNE        2$
1188 002552 112767 177777 176416          MOV      #-1,SYNCNO ;377
1189 002560 000402          BR         4$
1190 002562 104404          2$:  INSTER      ;RETRY
1191 002564 000757          BR         3$
1192 002566 000240          4$:  NOP
1193 002570 104403          INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1194 002572 017016          MWIRE6     ;MESSAGE
1195 002574 104412          SETFLG     ;SET FLAG BASED UPON INPUT STRING
1196 002576 001177          SEXMIT     ;THIS FLAG
1197 002600 104403          INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1198 002602 017064          MWIRE5     ;MESSAGE
1199 002604 104412          SETFLG     ;SET FLAG BASED UPON INPUT STRING
1200 002606 001200          SEREC     ;THIS FLAG
1201 002610 104403          INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1202 002612 017131          MWIRE4     ;MESSAGE
1203 002614 104412          SETFLG     ;SET FLAG BASED UPON INPUT STRING
1204 002616 001201          OPTCLR     ;THIS FLAG
1205 002620 104403          INSTR      ;OUTPUT MESSAGE & GET INPUT STRING
1206 002622 017205          MEXTJ     ;MESSAGE
1207 002624 104412          SETFLG     ;SET FLAG BASED UPON INPUT STRING
1208 002626 001203          JMRBY     ;THIS FLAG
1209
1210
1211          ;TEST START AND RESTART
1212
1213 002630 012767 000340 175140 .BEGIN: MOV      #340,PS ;LOCK OUT INTERRUPTS
1214          ; ***** LOAD STATUS MAP *****
1215          ; THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1216          ; USED IN THIS PROGRAM AND WILL BE PASSED TO ANY
1217          ; OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1218 002636 032777 000200 176234 BIT      #SW07 ;@SWR ; SW07 SET , IF YES BRANCH
1219 002644 001132          BNF      HEREU
  
```

1220	002646	116767	176324	176424	MOVB SYNCNO	,NOSYNC	: SYNC CHARS
1221	002654	116767	176317	176417	MOVB SEXMII	,MISEX	: XMIT JUMPER
1222	002662	116767	176312	176412	MOVB SEREC	,RESEC	: SEC REC JUMPER
1223	002670	116767	176305	176405	MOVB OPTCLR	,CLROPT	: OPTIONAL JUMPER
1224	002676	116767	176300	176400	MOVB MULTD	,DMULT	: MULTIPLE DEVICES
1225	002704	116767	176273	176373	MOVB JMRBY	,BYJMR	: EXTERNAL MODEM
1226	002712	016767	176266	176366	MOV BASEADD	,ADDBASE	: PROG CONTROLLED 1ST ADDR
1227	002720	016767	176262	176362	MOV KEEPADD	,ADDKEEP	: SAVED 1ST DEVICE ADDR
1228	002726	016767	176256	176356	MOV LASTADD	,ADDLAST	: LAST DEVICE RXCSR ADDR
1229	002734	016767	176254	176354	MOV KEEPIV	,IVKEEP	: SAVED INTR VECTOR
1230	002742	016767	176244	176344	MOV BASEIV	,IVBASE	: RELOAD BASE INTR VECTOR
1231	002750	016767	176242	176342	MOV ACTREG	,REGACT	: ACTIVE REGISTER
1232	002756	016767	176236	176336	MOV ROTADD	,ADDROT	: ROTATING POINTER
1233	002764	013737	017524	001324	MOV @#DUPRT	,@#PRTDU	: DUT1 PRIORITY
1234	002772	016767	015206	176326	MOV DURIV	,RIVDU	: REC INTR VECTOR
1235	003000	016767	015202	176326	MOV DJRIS	,RISDU	: REC INTR STATUS
1236	003006	016767	015176	176314	MOV DUTIV	,TIVDU	: XMIT INTR VECTOR
1237	003014	016767	015172	176310	MOV DUTIS	,TISDU	: XMIT INTR STATUS
1238	003022	016767	014500	176306	MOV LESS1	,L1ESS	: PRIORITY TO ALLOW INTR
1239	003030	016767	014626	176326	MOV DUBASE	,BASEDU	: RXCSR BASE ADDRESS
1240	003036	016767	015116	176274	MOV RXCSR,	CSRXR	
1241	003044	016767	015112	176270	MOV HRXCSR,	CSRHRX	
1242	003052	016767	015106	176264	MOV RXDBUF,	BUFRXD	
1243	003060	016767	015102	176260	MOV HRXDBUF,	BUFHRXD	
1244	003066	016767	015076	176254	MOV PARCSR,	CSRPAR	
1245	003074	016767	015072	176250	MOV HPARCSR,	CSRHPAR	
1246	003102	016767	015066	176244	MOV TXCSR,	CSRTX	
1247	003110	016767	015062	176240	MOV HTXCSR,	CSRHTX	
1248	003116	016767	015056	176234	MOV TXDBUF,	BUFTXD	
1249	003124	016767	015052	176230	MOV HTXDBUF,	BUFHTXD	
1250					:*****		
1251					: THE FOLLOWING CODE WILL PRINT		
1252					: THE CONVERSATIONALLY SET JUMPER		
1253					: SETTINGS FROM THE STATUS MAP		
1254					: ON THE FIRST PASS OF		
1255					: THIS DIAGNOSTIC OR		
1256					: JUST AFTER THE QUESTIONING		
1257					: AND ANSWERING .		
1258					:*****		
1259							
1260	003132	00573	001222		HEREU:	TST @#FLAG	: TEST IF 1ST PASS
1261	003136	001402				BEQ SETFG	: IF FIRST PASS SET FLAG/PRINT
1262	003140	000167	000116			JMP THRU	: AROUND IF PASS > 1
1263	003144	104402	017262		SETFG:	TYPE MSTATUS	: PRINT 'STATUS MAP'
1264	003150	062737	000001	001222		ADD #1, @#FLAG	: SET FLAG ON 1ST PASS
1265	003156	010067	176042			MOV R0, HOLD0	: SAVE R0
1266	003162	010167	176040			MOV R1, HOLD1	: SAVE R1
1267	003166	012767	000003	176034		MOV #3, COUNT1	: COUNTER FOR WORDS PRINTED
1268	003174	012700	000002			MOV #BUFF1, R0	
1269	003200	012701	001300			MOV #STATUS, R1	: (BUFF1)=STATUS ETC.
1270	003204	010120			FILBUF:	MOV R1, (R0)+	: LOAD BUFF AS ABOVE
1271	003206	062701	000002			ADD #2, R1	: PREPARE STATUS ADDRESS
1272	003212	020127	001306			CMP R1, #STATUS+6	: CHECK IF 3 WORDS LOADED
1273	003216	001372				BNE FILBUF	: BACK TO LOAD NEXT ADDRESS
1274	003220	012700	000002			MC #BUFF1, R0	: LOAD FOR PRINT OUT
1275	003224	010067	176006			MOV R0, TABLE+4	: LOAD ADDRESS TO PRINT

1276	003230	012067	176006			MOV (R0)+, TABLE+10		: LOAD CONTENTS
1277	003234	104410	001232			CONVRT, TABLE		: PRINT ADDRESS/CONTENTS PAIR
1278	003240	104402	016524			TYPE, MCRLF		: CR AND LF
1279	003244	005367	175760			DEC COUNT1		: COUNT WORDS PRINTED
1280	003250	001365				BNE UP		: GO PRINT NEXT ADDRESS/CONTENTS
1281	003252	016700	175746			MOV HOLD0, R0		
1282	003256	016701	175744			MOV HOLD1, R1		
1283								:*****
1284								
1285	003262	012706	001100		THRU:	MOV #STACK, SP		: SET UP STACK
1286	003266	005737	000042			TST @#42		: IS PROGRAM UNDER MONITOR CONTROL
1287	003272	001056				BNE 3\$		
1288	003274	105767	175702			TSTB MULTD		: DON'T ALLOW LOCK ON TEST IF RUNNING
1289								: MULTIPLE DEVICES
1290	003300	001407				BEQ 5\$		: IF NO, TEST FOR LOCK ON TEST
1291	003302	016767	010654	010554		MOV BRW, TTST		: RESTORE NORMAL SCOPE LOOP
1292	003310	016767	010650	010550		MOV BRX, TTST+2		: DITTO
1293	003316	000444				BR 3\$		: JUMP AROUND IF YES
1294	003320	032777	000004	175552	5\$:	BIT #BIT2, @SWR		: CHECK FOR LOCK ON TEST
1295	003326	001416				BEQ 1\$		
1296	003330	104403				INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1297	003332	016660				MLOCK		: MESSAGE
1298	003334	104412				SETFLG		: SET FLAG BASED UPON INPUT STRING
1299	003336	001247				LOKFLG		: THIS FLAG
1300	003340	105767	175703			TSTB LOKFLG		: IS LOCK ON TEST OPTION SELECTED
1301	003344	001407				BEQ 1\$		
1302	003346	012767	000240	010510		MOV #NOP, TTST		
1303	003354	012767	000240	010504		MOV #NOP, TTST+2		: SET UP TO LOCK
1304	003362	000406				BR 2\$		
1305	003364	016767	010572	010472	1\$:	MOV BRW, TTST		
1306	003372	016767	010566	010466		MOV BRX, TTST+2		: LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1307	003400	032777	000002	175472	2\$:	BIT #SW01, @SWR		: IF SW01=1, GET STARTING PC
1308	003406	001410				BEQ 3\$		
1309	003410	104403				INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1310	003412	016645				MTSTPC		: MESSAGE
1311	003414	104405				PARAM		: CONVERT STRING
1312	003416	003446				TST1		: LOW LIMIT
1313	003420	012610				TLAST		: HIGH LIMIT
1314	003422	001114				RTRN		: STORE AT THIS LOCATION
1315	003424	001				.BYTE 1		: MASK
1316	003425	001				.BYTE 1		: HOW MANY TIMES + 2
1317	003426	000403				BR 4\$		
1318	003430	012767	003446	175456	3\$:	MOV #TST1, RTRN		: START AT TEST 1
1319	003436	104402	016641		4\$:	TYPE ,MR		: TYPE R
1320	003442	000177	175446			JMP @RTRN		: START TESTING
1321								
1322								

```

1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336 003446 012767 000001 175452
1337 003454 012767 003750 175434
1338 003462 105767 175510
1339 003466 100127
1340 003470 052777 000400 014476
1341 003476 012777 030000 014464
1342 003504 052777 000400 014462
1343
1344
1345 003512 012777 064001 014454
1346
1347
1348 003520 012777 036026 014442
1349 003526 052777 000020 014424
1350
1351 003534 042777 020000 014432
1352 003542 052777 020000 014424
1353
1354 003550 042777 020000 014416
1355 003556 052777 020000 014410
1356 003564 012767 000010 175346
1357 003572 012767 000026 175344
1358 003600 004767 014060
1359 003604 032777 004000 014346
1360 003612 001401
1361 003614 104000
1362 003616
1363 003616 012767 000010 175314
1364 003624 012767 000025 175312
1365 003632 004767 014026
1366
1367
1368
1369 003636 042777 020000 014330
1370 003644 052777 020000 014322
1371
1372 003652 042777 020000 014314
1373 003660 052777 020000 014306
1374 003666 012767 000002 175246
1375 003674 032777 004000 014256
1376 003702 001401
1377 003704 104000
1378 003706
  
```

```

::THIS TEST VERIFYS THAT BY SENDING ONLY ONE SYNC
::CHARACTER (TWO SELECTED BY STRAPPING ) REACT =0
::THEN SEND ONE ORDINARY CHARACTER (TO BREAK UP THE SEQUENCE)
::REACT =0.....IT WILL TAKE TWO MORE SYNC CHARS
::BEFORE REACT =1
::NOTE: THIS TEST WILL ONLY WORK WHEN TWO SYNC CHARS
::HAS BEEN BEEN SELECTED ...OTHERWISE JUMP AROUND THIS TEST
::MODE:SYNC INTERNAL (SYNINT)
::PARITY: NOPAR
::LENGTH: EIGHT
::
::THIS TEST CHECKS ONLY THE RECEIVER SECTION
::
TST1:  MOV      #1,TSTNO          ;SAVE THIS
      MOV      #TST2,NEXT        ;GO TO THIS TEST WHEN THRU
      TSTB     SYNCNO           ;TEST FOR # OF SYNC CHARS REQUIRED
      BPL      4$               ;IF NOT TWO GET OUT OF TEST
      BIS      #MRESET,@TXCSR    ;MASTER RESET
      MOV      #SYNINT,@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      #SYNINT!EIGHT!NOPAR!26,@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      #8,,SHIFT        ;# OF SHIFTS
      MOV      #26,TEMP1        ;SYNC CHAR TO BE SHIFTED IN
      JSR      PC,RPOKE         ;SHIFT IN THIS SYNC CHAR
      BIT      #REACT,@RXCSR    ;REACT = 0 ?
      BEQ      1$
      HLT
1$:
      MOV      #8,,SHIFT        ;# OF SHIFTS
      MOV      #25,TEMP1        ;ANY CHARACTER
      JSR      PC,RPOKE         ;SHIFT IN THIS CHARACTER
      ;YOU HAVE JUST LOST SYNCRONIZATION.....
      ;POKE THE CLK TWICE TO GET INTO SYNCRONIZATION
      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
      BIS      #CLK,@TXCSR      ;POKE CLK UP
      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
      BIS      #CLK,@TXCSR      ;POKE CLK UP
      MOV      #2,COUNT         ;# OF SYNC CHARS
2$:  BIT      #REACT,@RXCSR    ;REACT = 0 ?
      BEQ      3$
      HLT
3$:
      ;REACT SHOULD BE 0
  
```

1379	003706	012767	000010	175224	MOV	#8,SHIFT	;	# OF SHIFTS
1380	003714	012767	000026	175222	MOV	#26,TEMP1	;	SYNC CHAR
1381	003722	004767	013736		JSR	PC,RPOKE	;	SHIFT IN THIS SYNC CHAR
1382	003726	005367	175210		DEC	COUNT		
1383	003732	001360			BNE	2\$	;	IS COUNT = 0 ? NO GO AGAIN
1384	003734	032777	004000	014216	BIT	#REACT,@RXCSR	;	REACT = 1 ?
1385	003742	001001			BNE	4\$		
1386	003744	104000			HLT		;	REACT SHOULD BE ASSERTED
1387	003746							
1388	003746	104400						

4\$: SLOPE

```

1389      ::THIS TEST VERIFYS MODE SELECT.....
1390      ::SYNC EXTERNAL VS. SYNC INTERNAL
1391      ::
1392      ::BASICALLY THE TEST CHECKS THAT THE RECEIVED
1393      ::DATA FREEZES IN SYNC INTERNAL
1394      ::IN SYNC EXTERNAL THIS DATA IS TRANSPARENT
1395      ::THIS TEST ONLY APPLIES TO THE RECEIVER SECTION
1396      ::LENGTH: EIGHT
1397      ::NOTE:SEARCH SYNC IS NOT SET
1398 003750 012767 000002 175150 TST2:  MOV    #2,TSTNO      ;SAVE THIS
1399 003756 012767 004346 175132      MOV    #TST3,NEXT      ;GO TO THIS TEST WHEN THRU
1400 003764 052777 000400 014202      BIS    #MRESET,@TXCSR ;MASTER RESET
1401 003772 012777 030000 014170      MOV    #SYNINT,@PARCSR ;SET THE MODE
1402 004000 052777 000400 014166      BIS    #MRESET,@TXCSR ;MASTER RESET
1403
1404      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1405 004006 012777 064001 014160      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1406
1407      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1408 004014 012777 036026 014146      MOV    #SYNINT!EIGHT!NOPAR!26,@PARCSR
1409      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1410 004022 042777 020000 014144      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1411 004030 052777 020000 014136      BIS    #CLK,@TXCSR    ;POKE CLK UP
1412      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1413 004036 042777 020000 014130      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1414 004044 052777 020000 014122      BIS    #CLK,@TXCSR    ;POKE CLK UP
1415 004052 012767 000010 175060      MOV    #8,SHIFT      ;# OF SHIFTS
1416 004060 012767 000125 175056      MOV    #125,TEMP1    ;DATA CHARACTER
1417 004066 016703 014072      MOV    RXDBUF,R3     ;FOR ERROR MESSAGE
1418 004072 042777 040000 014074 1$:   BIC    #MTDATA,@TXCSR ;CLEAR MAINT DATA
1419 004100 000241      CLC
1420 004102 006067 175036      RLC    TEMP1        ;FORCE CARRY
1421 004106 103003      BCC    2$
1422 004110 052777 040000 014056      BIS    #MTDATA,@TXCSR ;SET MTDATA
1423 004116 042777 020000 014050 2$:   BIC    #CLK,@TXCSR    ;POKE CLK
1424 004124 052777 020000 014042      BIS    #CLK,@TXCSR
1425 004132 012700 000377      MOV    #377,R0      ;EXPECTED
1426 004136 017701 014022      MOV    @RXDBUF,R1   ;ACTUAL
1427 004142 020001      CMP    R0,R1
1428 004144 001401      BFQ   3$
1429 004146 104002      HLT   2
1430      ;DATA CHARACTER SHOULD COMPARE.....
1431 004150      3$:
1432 004150 005367 174764      DEC    SHIFT        ;IS IT THE LAST SHIFT ?
1433 004154 001346      BNE   1$           ;NO ...SHIFT SOME MORE
1434 004156 052777 000400 014010      BIS    #MRESET,@TXCSR ;MASTER RESET
1435 004164 012777 020000 013776      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1436 004172 052777 000400 013774      BIS    #MRESET,@TXCSR ;MASTER RESET
1437
1438      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1439 004200 012777 064001 013766      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1440
1441      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1442 004206 012777 026026 013754      MOV    #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1443      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1444 004214 042777 020000 013752      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
  
```



1445	004222	052777	020000	013744		BIS	#CLK,@TXCSR	:POKE CLK UP
1446	004230	012767	000010	174702		MOV	#8,SHIFT	:# OF SHIFTS
1447	004236	012767	000125	174700		MOV	#125,TEMP1	:DATA CHARACTER
1448	004244	005067	174676			CLR	TEMP2	
1449	004250	105167	174672			COMB	TEMP2	:MAKE LOW BYTE ALL 1'S
1450								:TO MATCH RXDBUF'S CONTENTS AFTER A MASTER RESET
1451	004254	042777	040000	013712	4\$:	BIC	#MTDATA,@TXCSR	:CLR MAINT DATA
1452	004262	000241				CLC		
1453	004264	006067	174654			ROR	TEMP1	:FORCE CARRY
1454	004270	103003				BCC	5\$	
1455	004272	052777	040000	013674		BIS	#MTDATA,@TXCSR	
1456	004300	106067	174642		5\$:	RORB	TEMP2	:PICK UP CARRY BIT
1457	004304	042777	020000	013662		BIC	#CLK,@TXCSR	
1458	004312	052777	020000	013654		BIS	#CLK,@TXCSR	
1459	004320	016700	174622			MOV	TEMP2,R0	:EXPECTED
1460	004324	017701	013634			MOV	@RXDBUF,R1	:ACTUAL
1461	004330	020001				CMP	R0,R1	
1462	004332	001401				BEQ	6\$	
1463	004334	104002				HLT	2	:DATA CHARACTER SHOULD COMPARE...
1464								:THE DATA CHARACTER SHOULD BE SEEN AS IT
1465								:SHIFTS ACROSS THE RECEIVER DATA OUTPUT
1466	004336				6\$:			
1467	004336	005367	174576			DEC	SHIFT	
1468	004342	001344				BNE	4\$	
1469	004344	104400				SCOPE		

```

1470      ::THIS TEST VERIFYS TX DONE FUNCTION, DONE = 1
1471      ::ALSO VERIFYS THAT THE TRANSMITTER CHIP IDLES 'SYNC' CHARACTER
1472      ::WHEN NO NEW CHARACTER IS LOADED INTO TXDBUF('SYNC' = BINARY COUNT PATTERN)
1473      ::MODE: SYNC INTERNAL
1474      ::PARITY: NO PARITY (NOPAR)
1475      ::LENGTH: EIGHT
1476      ::
1477 004346 012767 000003 174552 1ST3: MOV #3,TSTNO ;SAVE THIS
1478 004354 012767 004644 174534 MOV #TST4,NEXT ;GO TO THIS TEST WHEN THRU
1479 004362 012767 004546 174530 MOV #6$,LOCK ;SET UP FOR SCOPE LOOP
1480 004370 012704 036000 MOV #SYNINT!EIGHT!NOPAR!0,R4 ;MODE ETC.
1481 004374 052777 000400 013572 1$: BIS #MRESET,@TXCSR ;MASTER RESET
1482 004402 012777 030000 013560 MOV #SYNINT,@PARCSR ;SET THE MODE
1483 004410 052777 000400 013556 BIS #MRESET,@TXCSR ;MASTER RESET
1484 004416 012777 004020 013550 MOV #MINT!SEND,@TXCSR
1485 004424 010477 013540 MOV R4,@PARCSR
1486 004430 105777 013540 TSTB @TXCSR ;TXDONE?
1487 004434 100401 BMI 2$ ;TXDONE SHOULD BE SET
1488 004436 104000 HLT
1489 004440
1490 004440 112777 000021 013532 2$: MOVB #21,@TXDBUF ;LOAD ANY CHAR
1491 ;POKE CLK TO GET INTO SYNCIRONIZATION
1492 004446 052777 020000 013520 BIS #CLK,@TXCSR ;POKE CLK UP
1493 004454 042777 020000 013512 BIC #CLK,@TXCSR ;POKE CLK DOWN
1494 004462 105777 013506 TSTB @TXCSR
1495 004466 100001 BPL 3$ ;TXDONE SHOULD BE CLR
1496 004470 104000 HLT
1497 004472
1498 004472 052777 020000 013474 3$: BIS #CLK,@TXCSR ;POKE CLK UP
1499 004500 042777 020000 013466 BIC #CLK,@TXCSR ;POKE CLK DOWN
1500 004506 105777 013462 TSTB @TXCSR
1501 004512 100401 BMI 4$ ;TXDONE SHOULD BE SET
1502 004514 104000 HLT
1503 004516
1504 004516 012767 000007 174414 4$: MOV #7,SHIFT
1505 004524 5$:
1506 004524 052777 020000 013442 BIS #CLK,@TXCSR ;POKE CLK UP
1507 004532 042777 020000 013434 BIC #CLK,@TXCSR ;POKE CLK DOWN
1508 004540 005367 174374 DEC SHIFT
1509 004544 001367 BNE 5$ ;SHIFT OUT THE '21'
1510 004546 016703 013422 6$: MOV TXCSR,R3 ;FOR ERROR MESSAGE
1511 ;THE BIT WINDOW IS RE GENERATED INTO
1512 ;A CHARACTER AND LEFT PRESENTED IN R1
1513 ;FOR THE COMPARE OPERATION. IF YOU WANT TO
1514 ;LOCK ON A PARTICULAR SYNC CHARACTER...
1515 ;SET SWR09-1
1516 004552 005000 CLR R0
1517 004554 150400 BISS R4,R0 ;EXPECT 'SYNC'
1518 004556 012767 000010 174354 MOV #8,SHIFT ;# OF SHIFTS
1519 004564 005001 CLR R1
1520 004566 7$:
1521 004566 052777 020000 013400 BIS #CLK,@TXCSR ;POKE CLK UP
1522 004574 042777 020000 013372 BIC #CLK,@TXCSR ;POKE CLK DOWN
1523 004602 000241 CLC
1524 004604 032777 002000 013362 BIT #01W,@TXCSR ;BIT W ?
1525 004612 001401 BEQ 8$

```

1526	004614	000261		SEC	:SET CARRY
1527	004616	106001		RORB	R1 :PICK UP CARRY
1528	004620	005367	174314	DEC	SHIFT
1529	004624	001360		BNE	7\$ :FINISH THAT CHARACTER
1530	004626	020001		CMP	RO,R1 :CMP EXPECTED VS ACTUAL
1531	004630	001401		BEQ	9\$
1532	004632	104003		HLT	3 :SYNC CHAR IS NOT CORRECT
1533	004634				
1534	004634	104401		SCOPI	
1535	004636	105204		INCB	R4 :SET UP FOR NEXT SYNC HOLDING REG.
1536	004640	001255		BNE	1\$ :FINISHED WITH BINARY COUNT PATTERN ?
1537					
1538	004642	104400		SCOPE	
1539					

```

1540 ::THIS TEST PROVES THAT RXERR FREEZES THE 'RECEIVER
1541 ::RESET' WHILE IN STRIP SYNC MODE
1542 ::THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR
1543 ::WHEN STRIP SYNC IS SET AND SYNC CHARACTERS ARE SENT
1544 ::BUT IF AN ERROR SHOULD OCCUR...THIS AUTOMATIC RESET
1545 ::IS DISCOMBOBULATED
1546 ::IE: FORCE OVERRUN (OVERRUN) WHILE STRIP SYNC IS SET
1547 ::BY TRANSMITTING A DATA CHARACTER THEN TRANSMIT A SYNC CHARACTER
1548 ::AND DON'T READ THAT DATA CHARACTER. NOTE: NORMALLY THE LOGIC
1549 ::RESETS THE RXDONE & ERROR FLAGS PROVIDING THAT ONLY SYNC CHARACTERS ARE
1550 ::STRIPPED
1551 ::MODE: SYNC EXTERNAL (SYNEXT)
1552 ::LENGTH: EIGHT
1553 ::NOTE: THIS TEST USES BOTH RECEIVER AND TRANSMITTER LOGIC
1554 ::
1555 ::
1556 004644 012767 000004 174254 ST4: MOV #4,TSTNO ;SAVE THIS
1557 004657 012767 005146 174236 MOV #TST5,NEXT ;GO TO THIS TEST WHEN THRU
1558
1559 004660 052777 000400 013306 BIS #MRESET,@TXCSR ;MASTER RESET
1560 004666 012777 020000 013274 MOV #SYNEXT,@PARCSR ;SET THE MODE
1561 004674 052777 000400 013272 BIS #MRESET,@TXCSR ;MASTER RESET
1562
1563 ;SET MAINTENANCE MODE & SEND
1564 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1565 004702 012777 004020 013264 MOV #MINT.SEND,@TXCSR
1566
1567 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1568 004710 012777 026026 013252 MOV #SYNEXT!EIGHT!NOPAR.26,@PARCSR
1569 004716 112777 000026 013254 MOVB #26,@TXDBUF ;LOAD SYNC CHAR
1570 004724 052777 000420 013226 BIS #SYNSCH.STPSYN,@RXCSR ;SET SYNC SEARCH & STRIP SYNC
1571 004732 016703 013226 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1572 004736 012767 000003 174176 MOV #3,COUNT ;# OF TIMES SYNC WILL BE SENT
1573 004744 052777 020000 013222 BIS #CLK,@TXCSR ;POKE CLK UP
1574 004752 042777 020000 013214 BIC #CLK,@TXCSR ;POKE CLK DOWN
1575 004760 012767 000010 174152 1$: MOV #8.,SHIFT ;# OF SHIFTS
1576 004766
2$:
1577 004766 052777 020000 01320C BIS #CLK,@TXCSR ;POKE CLK UP
1578 004774 042777 020000 013172 BIC #CLK,@TXCSR ;POKE CLK DOWN
1579 005002 005367 174132 DEC SHIFT
1580 005006 001367 BNE 2$
1581 005010 105777 013144 TSTB @RXCSR ;RXDONE?
1582 005014 100001 BPL 3$
1583 005016 104000 HLT ;RXDONE SHOULD NOT ASSERT
1584 005020
3$:
1585 005020 005367 174116 DEC COUNT
1586 005024 001355 BNE 1$
1587 005026 012700 000026 MOV #26,R0 ;EXPECTED
1588 005032 017701 013126 MOV @RXDBUF,R1 ;ACTUAL
1589 005036 020001 CMP R0,R1
1590 005040 001401 BEQ 4$
1591 005042 104002 HLT 2 ;NOTE THAT OVERRUN SHOULD NOT OCCUR, ALSO
1592 ;SECOND & 3RD SYNC CHARACTER CAME FROM
1593 ;SYNC HOLDING REGISTER
1594 005044
4$:
1595 005044 012767 000003 17407U MOV #3,COUNT ;# OF TIMES

```



```

1626                                     ::THIS TEST VERIFYS THAT EITHER SEQUENCE OF
1627                                     ::LOADING TXDBUF AND SETTING SEND
1628                                     ::DOES CAUSE TRANSMISSION
1629                                     ::MODE: SYNC EXT
1630                                     ::LENGTH: EIGHT
1631                                     ::
1632 005146 012767 000005 173752 1$T5:  MOV    #5,TSTNO      ;SAVE THIS
1633 005154 012767 005354 173734      MOV    #TST6,NEXT      ;GO TO THIS TEST WHEN THRU
1634
1635 005162 052777 000400 013004      BIS    #MRESET,@TXCSR ;MASTER RESET
1636 005170 012777 020000 012772      MOV    #SYNEXT,@PARCSR;SET THE MODE
1637 005176 052777 000400 012770      BIS    #MRESET,@TXCSR ;MASTER RESET
1638
1639                                     ;SET MAINTENANCE MODE & SEND
1640                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1641 005204 012777 004020 012762      MOV    #MINT!SEND,@TXCSR
1642
1643                                     ;SFT MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1644 005212 012777 026026 012750      MOV    #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1645 005220 016703 012750                MOV    TXCSR,R3        ;SET UP FOR ERROR MESSAGE
1646 005224 042777 000020 012742      BIC    #SEND,@TXCSR   ;DROP SEND
1647 005232 012767 000025 173704      MOV    #25,TEMP1
1648 005240 112777 000025 012732      MOVB   #25,@TXDBUF    ;LOAD CHARACTER
1649 005246 052777 000020 012720      BIS    #SEND,@TXCSR   ;SET SEND
1650                                     ;GET INTO SYNCRONIZATION
1651 005254 052777 020000 012712      BIS    #CLK,@TXCSR    ;POKE CLK UP
1652 005262 042777 020000 012704      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1653 005270 012767 000010 173642      MOV    #8.,SHIFT     ;# OF SHIFTS
1654 005276 005000                1$:  CLR    R0
1655 005300 006067 173640                ROR    TEMP1
1656 005304 103002                BCC    2$
1657 005306 052700 002000                BIS    #BITW,R0      ;EQUIV OF BIT WINDOW
1658
1659                                     2$:
1660 005312 052777 020000 012654      BIS    #CLK,@TXCSR    ;POKE CLK UP
1661 005320 042777 020000 012646      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1662 005326 017701 012642      MOV    @TXCSR,R1     ;ACTUAL
1663 005332 042701 075777      BIC    #075777,R1    ;SAVE BIT WINDOW & DNA
1664 005336 020001      CMP    R0,R1
1665 005340 001401      BEQ    3$
1666 005342 104003      HLT    3
1667                                     ;BIT WINDOW DID NOT MATCH ACTUAL DATA BIT
1668                                     ;ALSO CHECK DNA
1669 005344 005367 173577      3$:  DEC    SHIFT
1670 005350 001352      BNE    1$
1671
1672 005352 104400      SCOPE
1673

```

```

1674                                     ::THIS TEST VERIFYS THAT DROPPING OF SEND IN THE
1675                                     ::MIDDLE OF TRANSMITTING A CHARACTER DOES INDEED
1676                                     ::FINISH TRANSMITTING THAT CHARACTER
1677                                     ::MODE: SYNC EXT
1678                                     ::LENGTH: EIGHT
1679                                     ::
1680
1681 005354 012767 000006 173544 TST6:  MOV    #6,TSTNO      ;SAVE THIS
1682 005362 012767 005570 173526      MOV    #TST7,NEXT      ;GO TO THIS TEST WHEN THRU
1683
1684 005370 052777 000400 012576      BIS    #MRESET,@TXCSR ;MASTER RESET
1685 005376 012777 020000 012564      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1686 005404 052777 000400 012562      BIS    #MRESET,@TXCSR ;MASTER RESET
1687
1688                                     ;SET MAINTENANCE MODE & SEND
1689                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1690 005412 012777 004020 012554      MOV    #MINT!SEND,@TXCSR
1691
1692                                     ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1693 005420 012777 026026 012542      MOV    #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1694 005426 016703 012542                MOV    TXCSR,R3        ;SETUP FOR ERROR MESSAGE
1695 005432 112777 000252 012540      MOV    #252,@TXDBUF    ;LOAD DATA CHAR.
1696 005440 012767 000252 173776      MOV    #252,TEMP1      ;SHIFTED CHAR
1697 005446 012767 000000 173464      MOV    #8,SHIFT        ;# OF SHIFTS
1698                                     ;GET INTO SYNCRONIZATION
1699 005454 052777 020000 012512      BIS    #CLK,@TXCSR     ;POKE CLK UP
1700 005462 042777 020000 012504      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1701
1702 005470 005000                1$:   CLR    R0
1703 005472 006067 173446      ROR    TEMP1          ;FORCE CARRY
1704 005476 103002                BCC    2$
1705 005500 052700 002000      BIS    #BITW,R0        ;EQUIV OF BIT WINDOW
1706
1707 005504                2$:   BIS    #CLK,@TXCSR     ;POKE CLK UP
1708 005504 052777 020000 012462      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1709 005512 042777 020000 012454      MOV    @TXCSR,R1       ;ACTUAL
1710 005520 017701 012450      BIC    #075777,R1     ;SAVE ONLY BIT WINDOW & DNA
1711 005524 042701 075777      CMP    R0,R1
1712 005530 020001                BEQ    3$
1713 005532 001401                HLT
1714 005534 104003                3$:   HLT
1715                                     ;BIT WINDOW DID NOT MATCH
1716                                     ;ACTUAL DATA 3IT
1717 005536 005367 173376      DEC    SHIFT
1718 005542 022767 000003 173370      CMP    #3,SHIFT
1719 005550 001003                BNE    4$
1720 005552 042777 000020 012414      BIC    #SEND,@TXCSR    ;DROP SEND
1721 005560 005767 173354      TST   SHIFT
1722 005564 001341                BNE    1$              ;DO IT AGAIN?
1723
1724 005566 104400                SCOPE
1725

```



```

1726                                     :: THIS TEST VERIFYS THAT RXDONE ASSERTS WHEN STRIP SYNC IS SET
1727                                     :: MODE: SYNC INTERNAL
1728                                     :: LENGTH: EIGHT
1729                                     ::
1730 005570 012767 000007 173330 TST7: MOV #7,TSTNO ;SAVE THIS
1731 005576 012767 006044 173312 MOV #TST8,NEXT ;GO TO THIS TEST WHEN THRU
1732 005604 052777 000400 012362 BIS #MRESET,@TXCSR ;MASTER RESET
1733 005612 012777 030000 012350 MOV #SYNINT,@PARCSR ;SET THE MODE
1734 005620 052777 000400 012346 BIS #MRESET,@TXCSR ;MASTER RESET
1735
1736 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1737 005626 012777 064001 012340 MOV #MCDATA!CLK!MINT!BREAK,@TXCSR
1738
1739 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1740 005634 012777 036026 012326 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
1741
1742 005642 052777 000420 012310 BIS #SYNSCH!STPSYN,@RXCSR ;SET SYNC SEARCH &
1743 ;STRIP SYNC
1744 ;POKE CLK TO GET RECEIVER INTO SYNCRONIZATION.....
1745 005650 042777 020000 012316 BIC #CLK,@TXCSR ;POKE CLK
1746 005656 052777 020000 012310 BIS #CLK,@TXCSR ;
1747 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1748 005664 042777 020000 012302 BIC #CLK,@TXCSR ;POKE CLK DOWN
1749 005672 052777 020000 012274 BIS #CLK,@TXCSR ;POKE CLK UP
1750 005700 016703 012260 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1751 005704 012767 000002 173230 MOV #2,COUNT ;# OF TIMES OF SYNC CHARS.
1752 ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
1753 005712 105767 173260 TSTB SYNCNO
1754 005716 100402 BMI 1$ ;WILL IT BE ONE OR TWO ?
1755 005720 005367 173216 DEC COUNT ;MAKE IT ONE LESS
1756 005724 012767 000010 173206 1$: MOV #8,SHIFT ;#OF SHIFTS
1757 005732 012767 000026 173204 MOV #26,TEMP1 ;SYNC CHAR
1758 005740 004767 011720 JSR PC,RPOKE
1759 005744 005367 173172 DEC COUNT ;IS IT THE LAST SYNC CHAR ?
1760 005750 001365 BNE 1$ ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
1761 005752 032777 004000 012200 BIT #REACT,@RXCSR ;REACT=1?
1762 005760 001001 BNE 2$
1763 005762 104000 HLT ;REACT SHOULD BE ASSERTED
1764 005764 2$:
1765 005764 105777 012170 TSTB @RXCSR ;RXDONE=0?
1766 005770 100001 BPL 3$
1767 005772 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
1768 005774 3$:
1769 005774 012767 000010 173136 MOV #8,SHIFT ;#OF SHIFTS
1770 006002 012767 000025 173134 MOV #25,TEMP1 ;ANY CHARACTER
1771 006010 004767 011650 JSR PC,RPOKE
1772 006014 105777 012140 TSTB @RXCSR ;RXDONE-1?
1773 006020 100401 BMI 4$
1774 006022 104000 HLT ;RXDONE SHOULD NOW BE ASSERTED
1775 006024 4$:
1776 006024 012700 000025 MOV #25,R0 ;EXPECTED
1777 006030 017701 012130 MOV @RXDBUF,R ;ACTUAL
1778 006034 020001 CMP R0,R1
1779 006036 001401 BEQ 5$
1780 006040 104002 HLT
1781 006042 5$:

```

CZDUE-D MACY11 30A(1052) 21-NOV-78 15:50 PAGE 4?  
CZDUE.D.P11 18-JUL-78 15:56

N 3

SEQ 0039

1782 006042 104400  
1783

SCOPE

```

1784      ::THIS TEST VERIFYS THAT BY DROPPING SYNCSN
1785      ::IN THE MIDDLE OF A CHARACTER, SYNC CHARACTER SEQUENCE
1786      ::IS NEEDED BEFORE RECACT, RXDONE ASSERT AGAIN.
1787      ::ALSO NOTE: SINCE RECACT IS DEPENDENT ON MATCH DETECT,
1788      ::AND IF SYNCSCH IS DROPPED IN THE MIDDLE OF
1789      ::A SYNC CHARACTER AND THEN RAISED AGAIN ;RXDONE SHOULD
1790      ::NOT ASSERT UNTIL NEW SYNC CHARACTER SEQUENCE
1791      ::MODE: SYNC INTERNAL (SYNINT)
1792      ::LENGTH: EIGHT
1793      ::
1794      ::
1795 006044 012767 000010 173054 TSTB:  MOV    #8,TSTNO      ;SAVE THIS
1796 006052 012767 006574 173036      MOV    #TST9,NEXT      ;GO TO THIS TEST WHEN THRU
1797
1798 006060 105767 173112      TSTB   SYNCNO          ;TWO SYNC CHARACTERS SELECTED ?
1799 006064 100002      BPL    1$              ;IF ANSWER WAS NO DO THIS TEST
1800 006066 000167 000500      JMP    16$            ;IF ANSWER WAS YES JUMP OVER THIS TEST
1801 006072
1802 006072 052777 000400 012074 1$:    BIS    #MRESET,@TXCSR ;MASTER RESET
1803 006100 012777 030000 012062      MOV    #SYNINT,@PARCSR ;SET THE MODE
1804 006106 052777 000400 012060      BIS    #MRESET,@TXCSR ;MASTER RESET
1805
1806      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1807 006114 012777 064001 012052      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1808
1809      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1810 006122 012777 036026 012040      MOV    #SYNINT!EIGHT!NOPAR!26,@PARCSR
1811
1812 006130 052777 000020 012022      BIS    #SYNCSCH,@RXCSR ;SET SYNC SEARCH
1813      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1814 006136 042777 020000 012030      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1815 006144 052777 020000 012022      BIS    #CLK,@TXCSR     ;POKE CLK UP
1816      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1817 006152 042777 020000 012014      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1818 006160 052777 020000 012006      BIS    #CLK,@TXCSR     ;POKE CLK UP
1819 006166 012767 000002 172746      MOV    #2,COUNT        ;# OF TIMES
1820 006174 016703 011764      MOV    RXDBUF,R3       ;FOR ERROR MESSAGE
1821 006200 012767 000010 172732 2$:    MOV    #8,SHIFT        ;# OF SHIFTS
1822 006206 012767 000026 172730      MOV    #26,TEMP1       ;SYNC CHAR.
1823 006214 004767 011444      JSR    PC,RPOKE
1824 006220 005367 172716      DEC    COUNT
1825 006224 001403      BEQ    3$
1826
1827 006226 105767 172744      ;TEST TO SEE HOW MANY SYNC CHARACTERS NEEDED
1828 006232 100762      TSTB   SYNCNO
1829      BMI    2$
1830 006234 032777 004000 011716 3$:    BIT    #RECACT,@RXCSR ;RECACT-1?
1831 006242 001001      BNE    4$
1832 006244 104000      HLT
1833      ;RECACT SHOULD BE SET
1834 006246 012767 000004 172664 4$:    MOV    #4,SHIFT        ;# OF SHIFTS
1835 006254 012767 000026 172662      MOV    #26,TEMP1       ;SYNC CHAR.
1836 006262 004767 011376      JSR    PC,RPOKE
1837 006266 032777 004000 011664      BIT    #RECACT,@RXCSR ;RECACT=1?
1838 006274 001001      BNE    5$
1839 006276 104000      HLT
1839      ;RECACT SHOULD STILL BE SET

```

```

1840 006300
1841 006300 042777 000020 011652 5$: BIC #SYNSCH,@RXCSR ;DROP SEARCH SYNC
1842 006306 032777 004000 011644 BIT #RECACT,@RXCSR ;RECACT=0?
1843 006314 001401 BEQ 6$
1844 006316 104000 HLT ;RECACT SHOULD NOT BE SET
1845
1846 ;NOW SHIFT TWO BITS TO ALLOW SEARCH SYNC =0 TO TAKE
1847 ;EFFECT IN THE LOGIC(THIS ALLOWS THE RECEIVER CHIP TO SEE
1848 ;THE DROPPING OF SEARCH SYNC)...MATCH DETECT IN THE REC.CHIP SHOULD ALSO DROP
1849 006320 012767 000002 172612 6$: MOV #2,SHIFT ;# OF SHIFTS
1850 006326 004767 011332 JSR PC,RPOKE
1851 006332 052777 000020 011620 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1852 006340 032777 004000 011612 BIT #RECACT,@RXCSR
1853 006346 001401 BEQ 7$
1854 006350 104000 HLT ;RECACT =0 ?
1855 006352
1856 006352 105777 011602 7$: TSTB @RXCSR
1857 006356 100001 BPL 8$
1858 006360 104000 HLT ;RXDONE = 0 ?
1859 006362
1860 006362 012767 000002 172550 8$: MOV #2,SHIFT ;# OF SHIFTS
1861 006370 004767 011270 JSR PC,RPOKE
1862 006374 032777 004000 011556 BIT #RECACT,@RXCSR ;RECACT=0?
1863 006402 001401 BEQ 9$
1864 006404 104000 HLT ;RECACT SHOULD NOT BE SET
1865 006406
1866 006406 105777 011546 9$: TSTB @RXCSR ;RXDONE=0?
1867 006412 100001 BPL 10$
1868 006414 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
1869 006416
1870 006416 012700 000026 MOV #26,R0 ;EXPECTED
1871 006422 017701 011536 MOV @RXDBUF,R1 ;ACTUAL
1872 006426 020001 CMP R0,R1
1873 006430 001401 BEQ 11$
1874 006432 104002 HLT 2 ;CHARACTERS SHOULD BE MATCHED
1875 006434
1876 006434 012767 000002 172500 11$: MOV #2,COUNT ;# OF TIMES OF SYNC CHARS.
1877 ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
1878 006442 105767 172530 TSTB SYNCNO
1879 006446 100402 BMI 12$ ;WILL IT BE TWO OR ONE ?
1880 006450 005367 172466 DEC COUNT ;IT WAS ONLY ONE NEEDED
1881 006454 012767 000010 172456 12$: MOV #8,SHIFT ;#OF SHIFTS
1882 006462 012767 000026 172454 MOV #26,TEMP1 ;SYNC CHAR
1883 006470 004767 011170 JSR PC,RPOKE
1884 006474 005367 172442 DEC COUNT ;IS IT THE LAST SYNC CHAR ?
1885 006500 001365 BNE 12$ ;GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
1886 006502 032777 004000 011450 BIT #RECACT,@RXCSR ;RECACT-1?
1887 006510 001001 BNE 13$
1888 006512 104000 HLT ;RECACT SHOULD BE ASSERTED
1889 006514
1890 006514 105777 011440 13$: TSTB @RXCSR ;RXDONE=0?
1891 006520 100001 BPL 14$
1892 006522 104000 HLT ;RXDONE SHOULD NOT BE ASSERTED
1893 006524
1894 006524 012767 000010 172406 14$: MOV #2,SHIFT ;#OF SHIFTS
1895 006532 012767 000025 172404 MOV #25,TEMP1 ;ANY CHARACTER

```

```
1896 006540 004767 011120 JSR PC,PPOKE
1897 006544 105777 011410 TSTB @RXCSR ;RXDONE=1?
1898 006550 100401 BMI 15$
1899 006552 104000 HLT ;RXDONE SHOULD NOW BE ASSERTED
1900 006554 15$:
1901 006554 012700 000025 MOV #25,R0 ;EXPECTED
1902 006560 017701 011400 MOV @RXDBUF,R1 ;ACTUAL
1903 006564 020001 CMP R0,R1
1904 006566 001401 BEQ 16$
1905 006570 104002 HLT 2 ;CHARACTERS SHOULD BE MATCHED
1906 ;IF THIS FAILS THEN CHECK THAT THE CORRECT
1907 ;RECEIVER CHIP IS BEING USED...WHAT IS
1908 ;HAPPENING IS THAT MATCH DETECT IS ASSERTING
1909 ;BEFORE A NEW SYNC CHARACTER SEQUENCE
1910 ;TRANSPIRES THUS RXDONE ASSERTS TOO SOON
1911 ;AND OVER RUN OCCURS SINCE THE RECEIVER WAS NOT READ
1912 ;CONCLUSION::: IF OLDER RECEIVER CHIP
1913 ;IS BEING USED THEN REPLACE IT WITH A NEW
1914 ;RECEIVER CHIP IF 1 SYNC CHARACTER SEQUENCE IS DESIRED
1915 006572 15$:
1916 006572 104400 SCOPE
1917
```

```

1918                                     ;;THIS TEST VERIFYS THAT HDX MODE DISQUALIFIES THE
1919                                     ;;RECEIVER WHEN SEND IS ASSERTED
1920                                     ;;MODE: SYNC EXT
1921                                     ;;LENGTH: EIGHT
1922                                     ;;NOTE: THIS TEST WORKS ONLY IN MAINT. EXTERNAL MODE
1923                                     ;;THIS TEST USES BOTH RECEIVER & TRANSMITTER LOGIC
1924 006574 012767 000011 172324 TST9: MOV #9,TSTNO ;SAVE THIS
1925 006602 012767 007152 172306 MOV #TST10,NEXT ;GO TO THIS TEST WHEN THRU
1926 006610 105767 172367 TSTB JMRBY
1927 006614 100*55 BPL 7$ ;GET OUT OF THIS TEST IF 'NO'
1928 006616 016703 011342 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1929 006622 052777 000400 011344 BIS #MRESET,@TXCSR ;MASTER RESET
1930 006630 012777 020000 011332 MOV #SYNEXT,@PARCSR ;SET THE MODE
1931 006636 052777 000400 011330 BIS #MRESET,@TXCSR ;MASTER RESET
1932
1933 ;SET MAINTENANCE MODE & SEND
1934 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1935 006644 012777 010020 011322 MOV #MEXT!SEND,@TXCSR
1936
1937 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1938 006652 012777 026026 011310 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1939 006660 052777 000020 011272 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1940 006666 112777 000025 011304 MOVB #25,@TXDBUF ;ANY CHARACTER
1941 ;POKE CLK FOR SYNCRONIZATION
1942 006674 052777 020000 011272 BIS #CLK,@TXCSR ;POKE CLK UP
1943 ;WAIT FOR CABLE & DRIVER DELAYS
1944 006702 016702 172230 MOV HOLD,R2 ;WAIT THIS AMT
1945 006706 64$:
1946 006706 005302 DEC R2 ;WAIT
1947 006710 001376 BNE 64$
1948 ;EXIT...
1949 006712 042777 020000 011254 BIC #CLK,@TXCSR ;POKE CLK DOWN
1950 ;WAIT FOR CABLE & DRIVER DELAYS
1951 006720 016702 172212 MOV HOLD,R2 ;WAIT THIS AMT
1952 006724 65$:
1953 006724 005302 DEC R2 ;WAIT
1954 006726 001376 BNE 65$
1955 ;EXIT...
1956 006730 012767 000010 172202 MOV #8.,SHIFT ;# OF SHIFTS
1957 006736 1$:
1958 006736 052777 020000 011230 BIS #CLK,@TXCSR ;POKE CLK UP
1959 ;WAIT FOR CABLE & DRIVER DELAYS
1960 006744 016702 172166 MOV HOLD,R2 ;WAIT THIS AMT
1961 006750 66$:
1962 006750 005302 DEC R2 ;WAIT
1963 006752 001376 BNE 66$
1964 ;EXIT...
1965 006754 042777 020000 011212 BIC #CLK,@TXCSR ;POKE CLK DOWN
1966 ;WAIT FOR CABLE & DRIVER DELAYS
1967 006762 016702 172150 MOV HOLD,R2 ;WAIT THIS AMT
1968 006766 67$:
1969 006766 005302 DEC R2 ;WAIT
1970 006770 001376 BNE 67$
1971 ;EXIT...
1972 006772 005367 172142 DEC SHIFT ;# OF SHIFTS
1973 006776 022767 000003 172134 CMP #3,SHIFT ;IS IT TIME TO LOAD NEXT CHAR ?

```

```

1974 007004 001003
1975 007006 112777 J00024 011164
1976 007014 005767 172120 2$:
1977 007020 001346
1978 007022 105777 011132
1979 007026 100401
1980 007030 104000
1981 007032 3$:
1982 007032 012700 000025
1983 007036 017701 011122
1984 007042 020001
1985 007044 001401
1986 007046 104002
1987 007050 4$:
1988 007050 052777 000010 011116
1989 007056 012767 000010 172054
1990 007064 5$:
1991 007064 052777 020000 011102
1992
1993 007072 016702 172040
1994 007076 68$:
1995 007076 005302
1996 007100 001376
1997
1998 007102 042777 020000 011064
1999
2000 007110 016702 172022
2001 007114 69$:
2002 007114 005302
2003 007116 001376
2004
2005 007120 005367 172014
2006 007124 001357
2007 007126 105777 011026
2008 007132 100001
2009 007134 104000
2010
2011 007136 6$:
2012 007136 017701 011022
2013 007142 020001
2014 007144 001401
2015 007146 104002
2016
2017
2018
2019 007150 7$:
2020 007150 104400
2021
  
```

```

BNE 2$
MOVB #24,@TXDBUF ;LOAD NEXT CHAR.
TST SHIFT
BNE 1$
TSTB @RXCSR ;RXDNE-1?
BMI 3$
HLT ;RXDONE SHOULD BE SET
3$:
MOV #25,R0 ;EXPECTED
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1
BEQ 4$
HLT 2 ;CHARACTERS SHOULD COMPARE
4$:
BIS #HDXEN,@TXCSR ;SET HALF DUPLEX HDX
MOV #8.,SHIFT ;# OF SHIFT
5$:
BIS #CLK,@TXCSR ;POKE CLK UP
;WAIT FOR CABLE & DRIVER DELAYS
MOV HOLD,R2 ;WAIT THIS AMT
68$:
DEC R2 ;WAIT
BNE 68$
;EXIT...
BIC #CLK,@TXCSR ;POKE CLK DOWN
;WAIT FOR CABLE & DRIVER DELAYS
MOV HOLD,R2 ;WAIT THIS AMT
69$:
DEC R2 ;WAIT
BNE 69$
;EXIT...
DEC SHIFT
BNE 5$
TSTB @RXCSR ;RXDONE-0?
BPL 6$
HLT ;RXDONE SHOULD NOT BE ASSERTED
;CHECK OUT HDX LOGIC
6$:
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1
BEQ 7$
HLT 2 ;CHARACTERS SHOULD COMPARE
;NOTE THAT CHARACTER 25 WILL BE FROZEN
;IN THE RXDBUF EVEN THOUGH CHARACTER 24 WAS
;SENT TO THE RECEIVER
7$:
SCOPE
  
```

```

2022      ::THIS TEST VERIFYS THAT BREAK FORCES A SPACE CONDITION
2023      ::ON THE LINE WHILE TRANSMITTING
2024      ::THIS TEST USES BOTH THE RECEIVER AND TRANSMITTER LOGIC
2025      ::MODE: SYNC EXT (SYNEXT)
2026      ::LENGTH: EIGHT
2027      ::
2028      ::
2029      007152 012767 000012 171746 TST10: MOV #10,TSTNO ;SAVE THIS
2030      007160 012767 007410 171730      MOV #TST11,NEXT ;GO TO THIS TEST WHEN THRU
2031      007166 052777 000400 011000      BIS #MRESET,@TXCSR ;MASTER RESET
2032      007174 012777 020000 010766      MOV #SYNEXT,@PARCSR ;SET THE MODE
2033      007202 052777 000400 010764      BIS #MRESET,@TXCSR ;MASTER RESET
2034
2035      ;SET MAINTENANCE MODE & SEND
2036      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2037      007210 012777 004020 010756      MOV #MINT!SEND,@TXCSR
2038
2039      ;SFT MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2040      007216 012777 026026 010744      MOV #SYNEXT!EIGHT.NOPAR!26,@PARCSR
2041      007224 052777 000020 010726      BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2042      007232 016703 010726      MOV RXDBUF,R3 ;FOR ERROR MESSAGE
2043      007236 012767 000002 171676      MOV #2,COUNT ;# OF TIMES
2044      007244 112777 000025 010726      MOVB #25,@TXDBUF ;ANY CHARACTER
2045      ;POKE CLK FOR SYNCRONIZATION
2046      007252 052777 020000 010714      BIS #CLK,@TXCSR ;POKE CLK UP
2047      007260 042777 020000 010706      BIC #CLK,@TXCSR ;POKE CLK DOWN
2048      007266 012700 000025      MOV #25,R0 ;EXPECTED
2049      007272 012767 000010 171640 1$: MOV #8.,SHIFT ;# OF SHIFTS
2050
2051      007300      2$:
2052      007300 052777 020000 010666      BIS #CLK,@TXCSR ;POKE CLK UP
2053      007306 042777 020000 010660      BIC #CLK,@TXCSR ;POKE CLK DOWN
2054      007314 005367 171620      DEC SHIFT
2055      007320 022767 000003 171612      CMP #3,SHIFT
2056      007326 001003      BNE 3$
2057      007330 112777 000024 010642      MOVB #24,@TXDBUF ;LOAD NEXT CHAR
2058      007336 005767 171576 3$: TST SHIFT
2059      007342 001356      BNE 2$
2060      007344 105777 010610      TSTB @RXCSR ;RXDONE=1?
2061      007350 100401      BMI 4$
2062      007352 104000      HLT
2063      007354 4$:
2064      007354 017701 010604      MOV @RXDBUF,R1 ;ACTUAL
2065      007360 020001      CMP R0,R1
2066      007362 001401      BEQ 5$
2067      007364 104003      HLT 3
2068      007366 5$:
2069      007366 052777 000001 010600      BIS #BREAK,@TXCSR ;SET BREAK
2070      007374 012700 000000      MOV #0,R0 ;EXPECTED
2071      007400 005367 171536      DEC COUNT
2072      007404 001332      BNE 1$
2073      007406 104400      SCOPE
2074

```



```

2075                                     :: THIS TEST VERIFYS THAT DSC CAUSES AN INTERRUPT
2076                                     :: THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
2077                                     :: INTERRUPT VECTOR: DURIV
2078 007410 012767 000013 171510 TST11: MOV #11,TSTNO ;SAVE THIS
2079 007416 012767 007630 171472 MOV #TST12,NEXT ;GO TO THIS TEST WHEN THRU
2080 007424 105767 171553 TSTB JMRBY ;IN MAINT EXTERNAL?
2081 007430 100076 BPL 7$ ;IF ANSWER NO JUMP AROUND TEST
2082 007432 052777 000400 010534 BIS #MRESET,@TXCSR ;MASTER RESET
2083 007440 105767 171535 TSTB OPTCLR ;IS THE OPTIONAL CLR JUMPER IN ?
2084 007444 100405 BMI 1$ ;YES
2085 007446 012777 000000 010504 MOV #0,@RXCSR ;CLR THE UNRESETTABLE BITS
2086 007454 005777 010500 TST @RXCSR ;GET RID OF DSC BY READING RXCSR
2087 007460 012777 007504 010516 1$: MOV #2$,@DURIV ;SET UP TRAPCATCHER
2088 007466 016777 010032 010512 MOV DUPRT,@DURIS ;
2089 007474 016767 010026 170274 MOV LESS1,PS ;ALLOW INTERRUPTS
2090 007502 000423 BR 4$ ;JUMP AROUND INTERRUPT SVC ROUTINE
2091                                     : THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2092 007504 012767 000340 170264 2$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
2093 007512 005777 010442 TST @RXCSR ;DSC=1?
2094 007516 100401 BMI 3$
2095 007520 104000 HLT ;FALSE INTERRUPT
2096 007522 3$:
2097 007522 042777 000040 010430 BIC #DSINTE,@RXCSR ;CLEAR INTERRUPT ENABLE
2098 007530 012716 007620 MOV #6$(SP) ;SET UP RETURN LOCATION
2099 007534 016777 010446 010442 MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2100 007542 012777 000000 01043E MOV #0,@DURIS ;
2101 007550 000002 RTI ;
2102
2103 007552 052777 000040 010400 4$: BIS #DSINTE,@RXCSR ;SET INTERRUPT ENABLE
2104 007560 052777 000002 010372 BIS #DTR,@RXCSR ;TRY TO CAUSE INTERRUPT
2105 007566 005000 CLR R0 ;
2106 007570 5$:
2107 007570 005200 INC R0 ;WAIT FOR INTERRUPT
2108 007572 001376 BNE 5$ ;
2109 007574 016777 010406 010402 MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2110 007602 012777 000000 010376 MOV #0,@DURIS ;
2111
2112 007610 042777 000040 010342 BIC #DSINTE,@RXCSR ;CLEAR INTERRUPT ENABLE
2113 007616 104000 HLT ;INTERRUPT FAILED TO OCCUR
2114 007620 012767 000340 170150 6$: MOV #LEVEL7,PS ;
2115 007626 104400 7$: SCOPE ;
2116

```

```

2117                                     ::THIS TEST VERIFYS THAT RXDONE CAUSES AN INTERRUPT
2118                                     ::MODE: SYNC EXTERNAL
2119                                     ::INTERRUPT VECTOR: DURIV
2120                                     ::LENGTH: EIGHT
2121
2122 007630 012767 000014 171270 TST12: MOV #12,TSTNO ;SAVE THIS
2123 007636 012767 010176 171252 MOV #TST13,NEXT ;GO TO THIS TEST WHEN THRU
2124
2125 007644 052777 000400 010322 BIS #MRESET,@TXCSR ;MASTER RESET
2126 007652 012777 020000 010310 MOV #SYNEXT,@PARCSR ;SET THE MODE
2127 007660 052777 000400 010306 BIS #MRESET,@TXCSR ;MASTER RESET
2128
2129 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2130 007666 012777 064001 010300 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2131
2132 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2133 007674 012777 026026 010266 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2134 007702 052777 000020 010250 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2135 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2136 007710 042777 020000 010256 BIC #CLK,@TXCSR ;POKE CLK DOWN
2137 007716 052777 020000 010250 BIS #CLK,@TXCSR ;POKE CLK UP
2138 007724 012777 007750 010252 MOV #1$,@DURIV ;SET UP TRAPCATCHER
2139 007732 016777 007566 010246 MOV DUPRT,@DURIS ;
2140 007740 016767 007562 170030 MOV LESS1,PS ;ALLOW INTERRUPTS
2141 007746 000425 BR 3$ ;JUMP AROUND INTERRUPT SVC ROUTINE
2142 ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2143 007750 012767 000340 170020 1$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
2144 007756 042777 000100 010174 BIC #RINTEN,@RXCSR ;CLEAR INTERRUPT ENABLE
2145 007764 105777 010170 TSTB @RXCSR ;RXDONE=1?
2146 007770 100401 BMI 2$
2147 007772 104000 HLT ;FALSE INTERRUPT
2148 007774 2$:
2149 007774 012716 010166 MOV #8$, (SP) ;SET UP RETURN LOCATION
2150 010000 016777 010202 010176 MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2151 010006 012777 000000 010172 MOV #0,@DURIS ;
2152 010014 017701 010144 MOV @RXDBUF,R1 ;CLEAR INTERRUPT
2153 010020 000002 RTI
2154
2155 010022 052777 000100 010130 3$: BIS #RINTEN,@RXCSR ;SET INTERRUPT ENABLE
2156 010030 012767 000010 171102 MOV #8,SHIFT ;# OF SHIFTS
2157 010036 012767 000025 171100 MOV #25,TEMP1 ;TO BE SHIFTED CHARACTER
2158 ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
2159 ;INFORMATION CONTAINED IN TEMP1 AND IT IS
2160 ;SHIFTED IN BY THE CONTENTS OF SHIFT
2161 010044 042777 040000 010122 4$: BIC #MTDATA,@TXCSR
2162 010052 000241 CLC
2163 010054 006067 171064 ROR TEMP1 ;FORCE CARRY
2164 010060 103003 BCC 5$
2165 010062 052777 040000 010104 BIS #MTDATA,@TXCSR
2166 010070 042777 020000 010076 5$: BIC #CLK,@TXCSR
2167 010076 052777 020000 010070 BIS #CLK,@TXCSR
2168 010104 005367 171030 DEC SHIFT
2169 010110 001355 BNE 4$
2170 ;INTERRUPT SHOULD NOW OCCUR
2171 010112 005000 CLR R0
2172 010114 6$:

```

```
2173 010114 005200      INC      R0      ;WAIT FOR INTERRUPT
2174 010116 001376      BNE      6$
2175 010120 016777 010062 010056      MOV      DURIS,@DURIV ;RESTORE TRAPCATCHER
2176 010126 012777 000000 010052      MOV      #0,@DURIS
2177 010134 016703 010024      MOV      RXDBUF,R3 ;FOR ERROR MESSAGE
2178 010140 012700 000025      MOV      #25,R0 ;EXPECTED
2179 010144 017701 010014      MOV      @RXDBUF,R1
2180 010150 042777 000100 010002      BIC      #RINTEN,@RXCSR ;CLR INTR ENABLE
2181 010156 020001      CMP      R0,R1
2182 010160 001401      BEQ      7$
2183 010162 104002      HLT      2 ;CHARACTERS SHOULD COMPARE
2184 010164      ;$:
2185 010164 104000      HLT      ;INTERRUPT FAILED TO OCCUR
2186
2187 010166 012767 000340 167602 8$:      MOV      #LEVEL7,PS
2188 010174 104400      SCOPE
2189
```

```

2190      ::THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP TO
2191      ::THE SAME VECTOR ARE BOTH EXECUTED
2192      ::INTERRUPT VECTOR: DURIV
2193      ::THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
2194
2195 010176 012767 000015 170722 TST13: MOV #13,TSTNO ;SAVE THIS
2196 010204 012767 010654 170704 MOV #TST14,NEXT ;GO TO THIS TEST WHEN THRU
2197 010212 105767 170765 TSTB JMRBY ;IN MAINT. EXTERNAL?
2198 010216 100402 BMI 1$ ;IF ANSWER WAS YES DO THIS TEST
2199 010220 000167 000426 JMP 15$ ;IF ANSWER WAS NO JUMP AROUND TEST
2200 010224
2201 010224 105767 170751 1$: TSTB OPTCLR ;IS THE OPTIONAL CLEAR JUMPER IN ?
2202 010230 100402 BMI 2$ ;YES
2203 010232 005077 007722 CLR @RXCSR ;NO CLEAR UNRESETTABLE BITS
2204 010236
2205 010236 052777 000400 007730 2$: BIS #MRESET,@TXCSR ;MASTER RESET
2206 010244 012777 020000 007716 MOV #SYNEXT,@PARCSR ;SET THE MODE
2207 010252 052777 000400 007714 BIS #MRESET,@TXCSR ;MASTER RESET
2208
2209 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2210 010250 012777 064001 007706 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2211
2212 ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
2213 010266 012777 026026 007674 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2214 010274 052777 000020 007656 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2215 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2216 010302 042777 020000 007664 BIC #CLK,@TXCSR ;POKE CLK DOWN
2217 010310 052777 020000 007656 BIS #CLK,@TXCSR ;POKE CLK UP
2218 010316 012777 010342 007660 MOV #3$,@DURIV ;SET UP TRAPCATCHER
2219 010324 016777 007174 007654 MOV DUPRT,@DURIS
2220 010332 016767 007170 167436 MOV LESS1,PS ;ALLOW INTERRUPT
2221 010340 000457 BR 9$ ;JUMP AROUND SVC ROUTINE
2222
2223 ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
2224 010342 012767 000340 167426 3$: MOV #LEVEL7,PS ;DON'T ALLOW ANY MORE INTERRUPTS
2225 010350 105777 007604 TSTB @RXCSR ;RXDONE = 1 ?
2226 010354 100401 BMI 4$
2227 010356 104000 HLT ;FALSE INTERRUPT
2228 010360
2229 010360 012716 010644 4$: MOV #14$,SP ;SET UP RETURN LOCATION
2230 010364 012777 010446 007612 MOV #7$,@DURIV ;SET UP TRAPCATCHER FOR SECOND
2231 ;INTERRUPT
2232 010372 052777 000002 007560 BIS #DTR,@RXCSR ;TRY TO CAUSE SECOND INTERRUPT
2233 010400 017701 007560 MOV @RXDBUF,R1 ;JUST READ RXDBUF TO CLR RXDONE
2234 ;TO ALLOW SECOND INTERRUPT
2235 010404 016767 007116 167364 MOV LESS1,PS ;ALLOW INTERRUPT
2236 010412 005000 CLR R0
2237 010414 5$:
2238 010414 005200 INC R0 ;WAIT FOR INTERRUPT
2239 010416 001376 BNE 5$
2240 010420 042777 000140 007532 BIC #RINTEN.DSINTE,@RXCSR ;CLR INTR ENABLES
2241 010426 104000 HLT ;2ND INTERRUPT FAILED TO OCCUR
2242
2243 010430 016777 007552 007546 6$: MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2244 010436 012777 000000 007542 MOV #0,@DURIS
2245 010444 000002 RTI
  
```

```

2246
2247
2248 010446 012767 000340 167322 ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
2249 010454 005777 007500 7$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
2250 010460 100401 ;TST @RXCSR ;DSC = 1 ?
2251 010462 104000 ;BMI 8$
2252 010464 ;HLT ;FALSE INTERRUPT
2253 010464 042777 000140 007466 8$: BIC #RINTEN!DSINTE,@RXCSR ;CLR BOTH INTR ENABLES
2254 010472 012716 010430 ;MOV #6$(,SP) ;SET UP RETURN LOCATION
2255 010476 000002 ;RTI
2256
2257 010500 052777 000140 007452 9$: BIS #RINTEN!DSINTE,@RXCSR ;SET INTERRUPT ENABLES
2258 010506 012767 000010 170424 ;MOV #8$,SHIFT ;# OF SHIFTS
2259 010514 012767 000025 170422 ;MOV #25,TEMP1
2260 ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
2261 ;INFORMATION CONTAINED IN TEMP1 AND IT IS
2262 ;SHIFTED IN BY THE CONTENTS OF SHIFT
2263 010522 042777 040000 007444 10$: BIC #MTDATA,@TXCSR
2264 010530 000241 ;CLC
2265 010532 006067 170406 ;ROR TEMP1 ;FORCE CARRY
2266 010536 103003 ;BCC 11$
2267 010540 052777 040000 007426 ;BIS #MTDATA,@TXCSR
2268 010546 042777 020000 007420 11$: BIC #CLK,@TXCSR
2269 010554 052777 020000 007412 ;BIS #CLK,@TXCSR
2270 010562 005367 170352 ;DEC SHIFT
2271 010566 001355 ;BNE 10$
2272 ;1ST INTERRUPT SHOULD NOW OCCUR
2273 010570 005000 ;CLR R0
2274 C 0572 12$:
2275 010572 005200 ;INC R0 ;WAIT FOR INTERRUPT
2276 010574 001376 ;BNE 12$
2277 010576 016777 007404 007400 ;MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2278 010604 012777 000000 007374 ;MOV #0,@DURIS
2279 010612 016703 007346 ;MOV RXDBUF,R3 ;FOR ERROR MESSAGE
2280 010616 012700 000025 ;MOV #25,R0 ;EXPECTED
2281 010622 017701 007336 ;MOV @RXDBUF,R1
2282 010626 042777 000140 007324 ;BIC #RINTEN!DSINTE,@RXCSR ;CLR INTERRUPT ENABLES
2283 010634 020001 ;CMP R0,R1
2284 010636 001401 ;BEQ 13$
2285 010640 104002 ;HLT 2 ;CHARACTERS SHOULD COMPARE
2286 010642 13$:
2287 010642 104000 ;HLT ;INTERRUPT FAILED TO OCCUR
2288
2289 010644 012767 000340 167124 14$: MOV #LEVEL7,PS ;DON'T ALLOW ANY MORE INTERRUPTS
2290 010652 104400 15$: SCOPE
2291
  
```

```

2292      ;:THIS TEST VERIFYS THAT DNA CAUSES AN INTERRUPT
2293      ;:MODE: SYNC EXTERNAL
2294      ;:INTERRUPT VECTOR: DUTIV
2295
2296 010654 012767 000016 170244 TST14: MOV #14,TSTNO ;SAVE THIS
2297 010662 012767 011150 170226 MOV #TST15,NEXT ;GO TO THIS TEST WHEN THRU
2298
2299 010670 052777 000400 007276 BIS #MRESET,@TXCSR ;MASTER RESET
2300 010676 012777 020000 007264 MOV #SYNEXT,@PARCSR ;SET THE MODE
2301 010704 052777 000400 007262 BIS #MRESET,@TXCSR ;MASTER RESET
2302
2303 ;SET MAINTENANCE MODE & SEND
2304 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2305 010712 012777 004020 007254 MOV #MINT!SEND,@TXCSR
2306
2307 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2308 010720 012777 026026 007242 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2309 010726 112777 000010 007244 MOV #25,@TXDBUF ;LOAD CHARACTER
2310 010734 012767 000010 170176 MOV #8,SHIF
2311 ;POKE CLK TO GET INTO SYNCRONIZATION
2312 010742 052777 020000 007224 BIS #CLK,@TXCSR ;POKE CLK UP
2313 010750 042777 020000 007216 BIC #CLK,@TXCSR ;POKE CLK DOWN
2314
2315 010756 1$:
2316 010756 052777 020000 007210 BIS #CLK,@TXCSR ;POKE CLK UP
2317 010764 042777 020000 007202 BIC #CLK,@TXCSR ;POKE CLK DOWN
2318 010772 005367 170142 DEC SHIFT ;LAST SHIFT?
2319 010776 001367 BNE 1$
2320 011000 012777 011050 007202 MOV #3$,@DUTIV ;SET UP TRAPCATCHER
2321 011006 016777 006512 007176 MOV DUPRT,@DUTIS ;
2322 011014 016767 006506 166754 MOV LESS1,PS ;ALLOW INTERUPTS
2323 011022 052777 000040 007144 BIS #DNAINTE,@TXCSR ;ENABLE INTERRUPT
2324 ;NOW POKE CLK TO GET DNA
2325 011030 052777 020000 007136 BIS #CLK,@TXCSR ;POKE CLK
2326 011036 005000 CLR R0
2327 2$:
2328 011040 005200 INC R0 ;WAIT FOR INTERRUPT
2329 011042 001376 BNE 2$
2330 011044 104000 HLT ;INTERRUPT FAILED TO OCCUR
2331 011046 000423 BR 5$ ;JUMP AROUND SVC ROUTINE
2332 ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
2333 011050 012767 000340 166720 3$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
2334 011056 005777 007112 TST @TXCSR ;DNA?
2335 011062 100401 BMI 4$
2336 011064 104000 HLT ;FALSE INTERRUPT
2337 4$:
2338 011066 042777 000040 007100 BIC #DNAINTE,@TXCSR ;CLR INTR ENABLE
2339 011074 012716 011140 MOV #6$, (SP) ;SET UP RETURN LOCATION
2340 011100 016777 007106 007102 MOV DUTIS,@DUTIV ;RESTORE TRAPCATCHER
2341 011106 012777 000000 007076 MOV #0,@DUTIS ;
2342 011114 000002 RTI
2343
2344 011116 016777 007070 007064 5$: MOV DUTIS,@DUTIV ;RESTORE TRAPCATCHER
2345 011124 012777 000000 007060 MOV #0,@DUTIS ;
2346
2347 011132 042777 000040 007034 BIC #DNAINTE,@TXCSR ;CLR INTERRUPT ENABLE
  
```

2348 011140 012767 000340 '66630 6\$: MOV #LEVEL7,PS ;RESTORE NO INTERRUPT STATUS  
2349 011146 104400 SCOPE  
2350

```

2351      :: THIS TEST VERIFYS THAT TXDONE CAUSES AN INTERRUPT
2352      :: INTERRUPT VECTOR: DUTIV
2353      :: NOTE: TXDONE = 1      AFTER A MASTER RESET
2354      ::
2355      011150 012767 000017 167750 TST15: MOV    #15,TSTNO      ;SAVE THIS
2356      011156 012767 011334 167732      MOV    #TST16,NEXT      ;GO TO THIS TEST WHEN THRU
2357
2358      011164 052777 000400 007002      BIS    #MRESET,@TXCSR  ;MASTER RESET
2359      011172 012777 011242 007010      MOV    #2$,@DUTIV      ;SET UP TRAPCATCHER
2360      011200 016777 006320 007004      MOV    DUPRT,@DUTIS    ;
2361      011206 016767 006314 166562      MOV    LESS1,PS        ;ALLOW INTERPUTS
2362      011214 052777 000100 006752      BIS    #TXINTE,@TXCSR  ;ENABLE INTERRUPT
2363      011222 005000      CLR    RO              ;
2364      011224      1$:      INC    RO              ;WAIT FOR INTERRUPT
2365      011224 005200      BNE   1$              ;
2366      011226 001376      BIC   #TXINTE,@TXCSR  ;CLR INTERRUPT ENABLE
2367      011230 042777 000100 006736      HLT   ;INTERRUPT FAILED TO OCCUR
2368      011236 104000      BR    4$              ;JUMP AROUND SVC ROUTINE
2369      011240 000423
2370
2371      :: THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
2372      011242 012767 000340 166526 2$:  MOV    #LEVEL7,PS     ;DON'T ALLOW ANYMORE INTERRUPTS
2373      011250 042777 000100 006716      BIC   #TXINTE,@TXCSR  ;CLR INTR ENABLE
2374      011256 105777 006712      TSTB  @TXCSR          ;TXDONE?
2375      011262 100401      BMI   3$              ;
2376      011264 104000      HLT   ;FALSE INTERRUPT
2377      011266      3$:
2378      011266 012716 011324      MOV    #5$, (SP)      ;SET UP RETURN LOCATION
2379      011272 016777 006714 006710      MOV    DUTIS,@DUTIV   ;RESTORE TRAPCATCHER
2380      011300 012777 000000 006704      MOV    #0,@DUTIS     ;
2381      011306 000002      RTI
2382
2383      011310 016777 006676 006672 4$:  MOV    DUTIS,@DUTIV   ;RESTORE TRAPCATCHER
2384      011316 012777 000000 006666      MOV    #0,@DUTIS     ;
2385
2386      011324 012767 000340 166444 5$:  MOV    #LEVEL7,PS     ;RESTORE NO INTERRUPT STATUS
2387      011332 104400      SCOPE
2388

```



```
2389
2390
2391
2392
2393
2394 011334 012767 000020 167564
2395 011342 012767 011516 167546
2396 011350 052777 000400 006616
2397 011356 012777 011446 006624
2398 011364 016777 006134 006620
2399 011372 016767 006126 166376
2400 011400 052777 000100 006566
2401 011406 005000
2402 011410
2403 011410 005200
2404 011412 001376
2405 011414 042777 000100 006552
2406 011422 012767 000340 166346
2407 011430 016777 006556 006552
2408 011436 012777 000000 006546
2409 011444 000423
2410
2411 011446 012767 000340 166322
2412 011454 042777 000100 006512
2413 011462 012716 011504
2414
2415 011466 016777 006520 006514
2416 011474 012777 000000 006510
2417 011502 000002
2418
2419
2420
2421
2422
2423
2424 011504 012767 000340 166264
2425 011512 104000
2426
2427
2428 011514 104400
2429
2430

::THIS TEST VERIFYS THAT TXDONE DOES NOT CAUSE AN INTERRUPT
::WHEN PROCESSOR PRIORITY LEVEL IS TOO HIGH
::INTERRUPT VECTOR: DUTIV
::NOTE: TXDONE = 1 AFTER A MASTER RESET
::
TST16: MOV #16,TSTNO ;SAVE THIS
MOV #TST17,NEXT ;GO TO THIS TEST WHEN THRU
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #2$,@DUTIV ;SET UP TRAPCATCHER
MOV DUPRT,@DUTIS
MOV DUPRT,PS ;SET PS LEVEL TOO HIGH
BIS #TXINTE,@TXCSR ;ENABLE INTERRUPT
CLR RO ;WAIT FOR INTERRUPT

1$: INC RO
BNE 1$
BIC #TXINTE,@TXCSR ;CLR INTR ENABLE
MOV #LEVEL7,PS ;DON'T ALLOW INTERRUPTS
MOV DUTIS,@DUTIV ;RESTORE TRAPCATCHER
MOV #0,@DUTIS
BR 4$ ;TEST IS OK....GET OUT OF TEST

;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2$: MOV #LEVEL7,PS ;DONT ALLOW ANYMORE INTERRUPTS
BIC #TXINTE,@TXCSR ;CLR INTR ENABLE
MOV #3$, (SP) ;SET UP RETURN LOCATION
;TO REPORT ERROR
MOV DUTIS,@DUTIV ;RESTORE TRAPCATCHER
MOV #0,@DUTIS
RTI
;END OF INTERRUPT SVC ROUTINE

;YOU SHOULD NOT GET INTO THIS FOLLOWING CODE UNLESS THERE
;WAS AN ERROR
3$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
HLT ;INTERRUPT SHOULD NOT OF OCCURED,CHECK
;THE INTERRUPT LEVEL SELECTED OR CHECK
;INTERRUPT LOGIC OR BOTH

4$: SCOPE
```

```

2431
2432
2433
2434
2435
2436
2437
2438
2439 011516 012767 000021 167402
2440 011524 012767 011712 167364
2441 011532 052777 000400 006434
2442 011540 012777 011602 006442
2443 011546 016777 005752 006436
2444 011554 016767 005746 166214
2445 011562 052777 000100 006404
2446 011570 005000
2447 011572
2448 011572 005200
2449 011574 001376
2450 011576 104000
2451 011600 000427
2452
2453 011602 012767 000340 '66166
2454 011610 012716 011652
2455 011614 012777 011624 006366
2456
2457
2458
2459 011622 000002
2460
2461 011624 012767 000340 166144
2462 011632 012716 011660
2463 011636 105777 006332
2464 011642 100401
2465 011644 104000
2466 011646
2467 011646 104000
2468
2469
2470 011650 000002
2471 011652 005000
2472 011654
2473 011654 005200
2474 011656 001376
2475 011660 016777 006326 006322
2476 011666 012777 000000 006316
2477 011674 042777 000100 006272
2478 011702 012767 000340 166066
2479 011710 104400
2480

::THIS TEST VERIFYS THAT TXDONE CAUSES ONLY ONE INTERRUPT
::PROVIDING THAT TXCSR IS NOT READ
::AND TXDBUF IS NOT LOADED (WRITTEN)
::THIS TEST CHECKS THE ONCE ONLY FLIP/FLOP (V2)
::OF THE INTERRUPT CONTROL LOGIC
::INTERRUPT VECTOR: DUTIV
::NOTE: TXDONE = 1 AFTER A MASTER RESET
TST17: MOV #17,TSTNO ;SAVE THIS
MOV #TST18,NEXT ;GO TO THIS TEST WHEN THRU
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #2$,@DUTIV ;SET UP TRAPCATCHER
MOV DUPRT,@DUTIS ;
MOV LESS1,PS ;ALLOW INTERRUPTS
BIS #TXINTE,@TXCSR ;ENABLE INTR ENABLE
CLR RO
$: INC RO
BNE 1$
HLT ;INTERRUPT FAILED TO OCCUR
BR 7$
;THE FOLLOWING IS THE INTR SVC ROUTINE
2$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTR
MOV #5$, (SP) ;SET UP RETURN LOCATION
MOV #3$,@DUTIV ;SET UP TRAPCATCHER TO
;PROVE THAT THE INTERRUPT DOES NOT OCCUR
;TWICE (AFTER RTI 'ING FROM THIS
;SVC ROUTINE
RTI
;THE FOLLOWING INTERRUPT SVC ROUTINE WILL CATCH THE SECOND INTR
3$: MOV #LEVEL7,PS ;DON'T ALLOW INTER
MOV #7$, (SP) ;SET UP RETURN LOCATION
TSTB @TXCSR ;TXDONE = 1?
BMI 4$
HLT ;TXDONE SHOULD BE SET
4$: HLT ;THE INTERRUPT WAS TAKEN TWICE.....
;CHECK OUT THE V2 FLIP/FLOP LOGIC
;IN THE INTERRUPT CONTROL LOGIC
RTI
5$: CLR RO ;ALLOW TIME TO CATCH SECOND
6$: INC RO ;IF IT WERE TO OCCUR
BNE 6$
7$: MOV DUTIS,@DUTIV ;RESTORE TRAPCATCHER
MOV #0,@DUTIS ;
BIC #TXINTE,@TXCSR ;CLR INTERRUPT ENABLE
MOV #LEVEL7,PS ;RESTORE NO INTERRUPT STATUS
SCOPE

```

```

2481
2482      ;; THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP
2483      ;; TO THE SAME VECTOR ARE BOTH EXECUTED
2484      ;; INTERRUPT VECTOR: DUTIV
2485      ;; MODE: SYNC EXTERNAL
2486 011712 012767 000022 167206 TST18: MOV #18,TSTNO ;SAVE THIS
2487 011720 012767 012302 167170 MOV #TST19,NEXT ;GO TO THIS TEST WHEN THRU
2488
2489 011726 052777 000400 006240 B'IS #MRESET,@TXCSR ;MASTER RESET
2490 011734 012777 020000 006226 MOV #SYNEXT,@PARCSR ;SET THE MODE
2491 011742 052777 000400 006224 BIS #MRESET,@TXCSR ;MASTER RESET
2492
2493      ;SET MAINTENANCE MODE & SEND
2494      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2495 011750 012777 004020 006216 MOV #MINT!SEND,@TXCSR
2496
2497      ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2498 011756 012777 026026 006204 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2499 011764 112777 000025 006206 MOV#B #25,@TXDBUF ;LOAD CHARACTER
2500 011772 012767 000010 167140 MOV #8,,SHIFT
2501
2502      ;POKE CLK TO GET INTO SYNCRONIZATION
2503 012000 052777 020000 006166 BIS #CLK,@TXCSR ;POKE CLK UP
2504 012006 042777 020000 006160 BIC #CLK,@TXCSR ;POKE CLK DOWN
2505
2506 012014 052777 020000 006152 1$: BIS #CLK,@TXCSR ;POKE CLK UP
2507 012022 042777 020000 006144 BIC #CLK,@TXCSR ;POKE CLK DOWN
2508 012030 005367 167104 DEC SHIFT ;LAST SHIFT?
2509 012034 001367 BNE 1$
2510 012036 012777 012100 006144 MOV #3$,@DUTIV ;SET UP TRAPCATCHER
2511 012044 016777 005454 006140 MOV DUPRT,@DUTIS ;
2512 012052 016767 005450 165716 MOV LESS1,PS ;ALLOW INTERRUPTS
2513 012060 052777 000140 006106 BIS #TXINTE!DNAINTE,@TXCSR ;ENABLE INTERRUPTS
2514 012066 005000 CLR RO
2515 012070 2$:
2516 012070 005200 INC RO ;WAIT FOR INTERRUPT
2517 012072 001376 BNE 2$ ;
2518 012074 104000 HLT ;INTERRUPT FAILED TO OCCUR
2519 012076 000464 BR 10$ ;JUMP AROUND SVC ROUTINES
2520
2521      ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
2522 012100 012767 000340 165670 3$: MOV #LEVEL7,PS ;DON'T ALLOW ANYMORE INTERRUPTS
2523 012106 005777 006062 TST @TXCSR ;DNA=0 ?
2524 012112 100001 BPL 4$
2525 012114 104000 HLT ;DNA SHOULD NOT BE ASSERTED
2526 012116 4$:
2527 012116 105777 006052 TSTB @TXCSR ;TXDONE - 1?
2528 012122 100401 BMI 5$
2529 012124 104000 HLT ;FALSE INTERRUPT
2530 012126 5$:
2531 012126 012716 012272 MOV #11$, (SP) ;SET UP RETURN LOCATION
2532 012132 012777 012216 006050 MOV #8$,@DUTIV ;SET UP TRAPCATCHER
2533      ;NOW POKE CLK TO BRING UP DNA
2534 012140 052777 020000 006026 B'IS #CLK,@TXCSR ;POKE CLK
2535 012146 112777 000025 006024 MOV#B #2$,@TXDBUF ;JUST LOAD ANY CHAR TO CLR
2536      ;TXDONE TO ALLOW SECOND INTERRUPT
  
```

```

2537 012154 016767 005346 165614      MOV    LESS1,PS      ;ALLOW INTERRUPTS
2538 012162 005000                      CLR    R0
2539 012164                      6$:
2540 012164 005200                      INC    R0            ;WAIT FOR INTERRUPT
2541 012166 001376                      BNE   6$
2542 012170 042777 000140 005776      BIC   #DNAINTE!TXINTE,@TXCSR ;CLR INTR ENABLES
2543 012176 104000                      HLT   ;2ND INTERRUPT FAILED TO OCCUR
2544
2545 012200 016777 006006 006002 7$:   MOV    DUTIS,@DUTIV  ;RESTORE TRAPCATCHER
2546 012206 012777 000000 005776      MOV    #0,@DUTIS
2547 012214 000002                      RTI
2548
2549                                     ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
2550 012216 012767 000340 165552 8$:   MOV    #LEVEL7,PS
2551 012224 005777 005744                      ST    @TXCSR        ;DNA
2552 012230 100401                      BNE   9$
2553 012232 104000                      HLT   ;FALSE INTERRUPT
2554 012234
2555 012234 042777 000140 005732 9$:   BIC   #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTR ENABLES
2556 012242 012716 012200                      MOV    #7$(SP)      ;SETUP RETURN LOCATION
2557 012246 000002                      RTI
2558
2559 012250 016777 005736 005732 10$:  MOV    DUTIS,@DUTIV  ;RESTORE TRAPCATCHER
2560 012256 012777 000000 005726      MOV    #0,@DUTIS
2561
2562 012264 042777 000140 005702      BIC   #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTERRUPT
2563                                     ;ENABLES
2564 012272 012767 000340 165476 11$:  MOV    #LEVEL7,PS   ;RESTORE NO INTERRUPT STATUS
2565
2566 012300 104400
2567                                     SCOPE
  
```

```

2568      ::THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
2569      ::IT BASICALLY CHECKS THE EXISTANCE OF
2570      ::THE FREE RUNNING OSCILLATOR
2571      ::MODE: ISOCRONOUS (ISYMOD)
2572      ::LENGTH: EIGHT
2573      ::THIS TEST USES BOTH . = RECEIVER & TRANSMITTER LOGIC
2574
2575 012302 012767 000023 166616 TST19: MOV #19,TSTNO ;SAVE THIS
2576 012310 012767 012610 166600 MOV #TST20,NEXT ;GO TO THIS TEST WHEN THRU
2577 012316 052777 000400 005650 BIS #MRESET,@TXCSR ;MASTER RESET
2578 012324 012777 000000 005636 MOV #ISYMOD,@PARCSR ;LOAD THE MODE
2579 012332 052777 000400 005634 BIS #MRESET,@TXCSR ;MASTER RESET
2580 012340 012777 006026 005622 MOV #ISYMOD!EIGHT!NOPAR.26,@PARCSR ;LOAD THE MODE,
2581 ;# OF BITS PER CHAR,PARITY SENSE(NO PARITY),
2582 ;&SYNC CHARACTER (26)
2583 012346 112777 000025 005624 MOVB #25,@TXDBUF ;LOAD THE CHAR
2584 012354 012777 012454 005622 MOV #3$,@DURIV ;SET UP TRAPCATCHER
2585 012362 016777 005136 005616 MOV DUPRT,@DURIS ;
2586 012370 016767 005132 165400 MOV LESS1,PS ;ALLOW INTERRUPTS
2587 012376 016703 005562 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2588 012402 012700 000025 MOV #25,R0 ;EXPECTED
2589 012406 012777 014020 005560 MOV #SYSTST.SEND,@TXCSR ;OK NOW LOAD SEND &
2590 ;MAINT. MODE
2591 012414 052777 000120 005536 BIS #SYNSCH!RINTEN,@RXCSR ;SET SEARCH SYNC &
2592 ;RECEIVER INTERRUPT
2593 ;ENABLE & WAIT FOR INTERRUPT
2594 012422 005067 166526 CLR TEMPS
2595 012426 005002 1$: CLR R2
2596 012430 2$:
2597 012430 005202 INC R2 ;WAIT FOR INTERRUPT
2598 012432 001376 BNE 2$
2599 012434 005267 166514 INC TEMPS
2600 012440 022767 000003 166506 CMP #3,TEMPS
2601 012446 002367 BGE 1$
2602 012450 104000 HLT ;INTERUPT DID NOT OCCUR
2603 012452 000423 BR 4$
2604
2605 ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
2606 012454 012767 000340 165314 3$: MOV #LEVEL7,PS ;PREVENT INTERRUPTS
2607 012462 017704 005472 MOV @RXCSR,R4 ;SAVE
2608 012466 017701 005472 MOV @RXDBUF,R1 ;ACTUAL
2609 012472 016777 005510 005504 MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2610 012500 012777 000000 005500 MOV #0,@DURIS ;
2611 012506 012716 012554 MOV #5$, (SP) ;SET UP RETURN
2612 012512 042777 000100 005440 BIC #RINTEN,@RXCSR ;CLR INTERRUPT ENABLE
2613 012520 000002 RTI
2614
2615 012522 042777 000100 005430 4$: BIC #RINTEN,@RXCSR ;CLR INTERRUPT ENABLE
2616 012530 012767 000340 165240 MOV #LEVEL7,PS ;PREVENT INTERRUPTS
2617 012536 016777 005444 005440 MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
2618 012544 012777 000000 005434 MOV #0,@DURIS ;
2619 012552 000415 BR 7$
2620
2621 012554 020001 5$: CMP R0,R1
2622 012556 001401 BEQ 6$
2623 012560 104002 HLT ? ;CHARACTERS DID NOT MATCH

```

2624	012562			6\$:			
2625	012562	016703	005372		MOV	RXCSR,R3	: SETUP FOR ERROR MESSAGE
2626	012566	012700	000200		MOV	#200,R0	: EXPECTED
2627	012572	010401			MOV	R4,R1	: ACTUAL
2628	012574	042701	177577		BIC	#177577,R1	: SAVE ONLY RXDONE
2629	012600	020001			CMP	R0,R1	
2630	012602	001401			BEQ	7\$	
2631	012604	104001			HLT	1	: FALSE INTERRUPT
2632	012606			7\$:			
2633	012606	104400			SCOPE		

```

2634      ::THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
2635      ::IT BASICALLY CHECKS THE EXISTANCE OF
2636      ::THE FREE RUNNING OSCILLATOR
2637      ::MODE: SYNINT
2638      ::LENGTH: EIGHT
2639      ::THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
2640      ::
2641 012610 012767 000024 166310 TST20: MOV #20,TSTNO ;SAVE THIS
2642 012616 012767 013452 166272 MOV #.EOP,NEXT ;GO TO THIS TEST WHEN THRU
2643 012624 052777 000400 005342 BIS #MRESET,@TXCSR ;MASTER RESET
2644 012632 012777 030000 005330 MOV #SYNINT,@PARCSR ;SET THE MODE
2645 012640 052777 000400 005326 BIS #MRESET,@TXCSR ;MASTER RESET
2646
2647 ;SET MAINTENANCE MODE & SEND
2648 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2649 012646 012777 014020 005320 MOV #SYSTST!SEND,@TXCSR
2650
2651 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2652 012654 012777 036026 005306 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2653 012662 052777 000400 005270 BIS #SYNSCH!STPSYN,@RXCSR ;SET SEARCH SYNC &
2654 ;STRIP SYNC SO THAT RXDONE ASSERTS
2655 ;WHEN CHAR '25' ARRIVES AND NOT BEFORE...
2656 ;...THEREFORE,SET STRIP SYNC
2657 ;.....WAIT FOR SYNSCH TO BE
2658 ;CLOCKED IN BY SYSTST CLK
2659 012670 005067 166260 CLR TEMPS
2660 012674
2661 012674 005002 1$: CLR R2
2662 012676
2663 012676 005202 2$: INC R2 ;WAIT
2664 012700 001376 BNE 2$
2665 012702 005267 166246 INC TEMPS
2666 012706 022767 000003 166240 CMP #3,TEMPS
2667 012714 002367 BGE 1$ ;GO BACK TO CLR R2 AND WAIT SOME MORE
2668 012716 012777 013136 005260 MOV #7$,@DUR!V ;SET UP TRAPCATCHER
2669 012724 016777 004574 005254 MOV DUPRT,@DURIS
2670 012732 012777 013232 005250 MOV #8$,@DUT!V
2671 012740 016777 004560 005244 MOV DUPRT,@DUTIS
2672 012746 016767 004554 165022 MOV LESS1,PS ;ALLOW INTERRUPTS
2673 012754 016703 005204 MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2674 012760 012700 000025 MOV #25,R0 ;EXPECTED CHAR
2675 012764 012767 000002 166150 MOV #2,COUNT ;# OF SYNC CHARS TO GET INTO
2676 ;SYNCRONIZATION
2677 012772 105767 166200 TSTB SYNCNO ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
2678 012776 100402 BMI 3$
2679 013000 005367 166136 DEC COUNT ;MAKE IT ONE LESS
2680 013004 052777 000100 005146 3$: BIS #RINTEN,@RXCSR ;SET INTERRUPT ENABLES
2681 013012 052777 000100 005154 BIS #TXINTE,@TXCSR
2682 013020 000167 000012 JMP 5$ ;THE FIRST XMIT INTERRUPT SHOULD COME
2683 ;FROM TXDONE - 1 AFTER A MASTER RESET
2684 013024 112777 000026 005146 4$: MOVB #26,@TXDBUF ;LOAD SYNC CHAR
2685 013032 005067 166116 CLR TEMPS
2686 013036 005002 5$: CLR R2 ;WAIT FOR INTERRUPT
2687 013040 6$:
2688 013040 005202
2689 013042 001376 INC R.
BNE 6$
  
```

```

2690 013044 005267 166104      INC    TEMP5
2691 013050 022767 000003 166076  CMP    #3,TEMP5
2692 013056 002367          BGE    5$
2693 013060 012767 000340 164710  MOV    #LEVEL7,PS      ;PREVENT INTERRUPTS
2694 013066 042777 000100 005100  BIC    #TXINTE,@TXCSR  ;CLR INTR ENABLES
2695 013074 042777 000100 005066  BIC    #RINTEN,@RXCSR
2696 013102 016777 005100 005074  MOV    DURIS,@DURIV    ;RESTORE TRAPCATCHER
2697 013110 012777 000000 005070  MOV    #0,@DURIS
2698 013116 016777 005070 005064  MOV    DUTIS,@DUTIV
2699 013124 012777 000000 005060  MOV    #0,@DUTIS
2700 013132 104000          HLT
2701
2702
2703
2704
2705
2706
2707 013134 000542          BR     17$      ;GET OUT OF THE TEST
2708
2709
2710 013136 012767 000340 164632  ;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
2711 013144 017704 005010 7$: MOV    #LEVEL7,PS      ;PREVENT INTERRUPTS
2712 013150 017701 005010  MOV    @RXCSR,R4      ;SAVE
2713 013154 016777 005026 005022  MOV    @RXDBUF,R1    ;ACTUAL
2714 013162 012777 000000 005016  MOV    DURIS,@DURIV  ;RESTORE TRAPCATCHER
2715 013170 016777 005016 005012  MOV    #0,@DURIS
2716 013176 012777 000000 005006  MOV    DUTIS,@DUTIV
2717 013204 012716 013366  MOV    #13$(SP)      ;SET UP RETURN LOCATION
2718 013210 042777 000100 004742  BIC    #RINTEN,@RXCSR ;CLR INTERRUPT ENABLES
2719 013216 042777 000100 004750  BIC    #TXINTE,@TXCSR
2720 013224 016705 165712  MOV    COUNT,R5     ;SAVE COUNT
2721 013230 000002          RTI
2722
2723
2724 013232 005367 165704  ;END OF RECEIVER INTERRUPT SVC ROUTINE
2725 013236 100403 8$:
2726 013240 012716 013024  ;...THE FOLLOWING IS THE XMITTER INTERRUPT SVC ROUTINE
2727
2728 013244 000002          RTI
2729 013246 012716 013254 9$: MOV    #10$(SP)      ;SET UP RETURN LOCATION
2730 013252 000002          RTI
2731
2732 013254 112777 000025 004716  ;END OF XMITTER INTERRUPT SVC ROUTINE
2733 013262 042777 000100 004704 10$: MOV    #25,@TXDBUF   ;LOAD CHARACTER
2734 013270 005067 165660  BIC    #TXINTE,@TXCSR ;CLR INTR ENABLE
2735 013274 005002          CLR    TEMP5
2736 013276          11$: CLR    R2          ;WAIT FOR INTERRUPT(RECEIVER)
2737 013276 005202          12$:
2738 013300 001376          INC    R2
2739 013302 005267 165646  BNE    12$
2740 013306 022767 000003 165640  INC    TEMP5
2741 013314 002367          CMP    #3,TEMP5
2742 013316 012767 000340 164452  BGE    11$
2743 013324 042777 000100 004626  MOV    #LEVEL7,PS      ;PREVENT INTERRUPTS
2744 013332 016777 004650 004644  BIC    #RINTEN,@RXCSR  ;CLR INTR ENABLE
2745 013340 012777 000000 004640  MOV    DUTIS,@DURIV    ;RESTORE TRAPCATCHER
2745 013340 012777 000000 004640  MOV    #0,@DURIS

```



2746	013346	016777	004640	004634	MOV	DUTIS,@DUTIV ;
2747	013354	012777	000000	004630	MOV	#0,@DUTIS ;
2748	013362	104000			HLT	:RECEIVER INTR FAILED TO OCCUP
2749	013364	000426			BR	17\$ ;GET OUT OF TEST
2750	013366	020001			13\$: CMP	R0,R1
2751	013370	001401			BEQ	14\$
2752	013372	104002			HLT	2 ;CHARACTERS DID NOT MATCH
2753	013374				14\$:	
2754	013374	016703	004560		MOV	RXC SR,R3 ;SET UP FOR ERROR MSG
2755	013400	012700	000200		MOV	#200,R0 ;EXPECTED RXDONE
2756	013404	010401			MOV	R4,R1 ;ACTUAL
2757	013406	042701	177577		BIC	#177577,R1 ;SAVE ONLY RXDONE
2758	013412	020001			CMP	R0,R1
2759	013414	001401			BEQ	15\$
2760	013416	104001			HLT	1 ;FALSE INTERRUPT
2761	013420				15\$:	
2762	013420	020527	177777		CMP	R5,#-1 ;WAS COUNT --1 WHEN RECEIVER
2763						;INTERRUPTED ?
2764	013424	001401			BEQ	*\$
2765	013426	104000			HLT	;IF R5 IS GREATER THAN -1.....
2766						;THEN EITHER THE # OF SYNC STRAP IS WRONG
2767						;OR RXDONE IS OCCURING TOO SOON
2768	013430				16\$:	
2769	013430	026727	165506	177777	CMP	COUNT,#-1
2770	013436	001401			BEQ	17\$
2771	013440	104000			HLT	;IF THIS TEST FAILS,BUT THE ABOVE TEST
2772						;DOESN'T.....IT MAY BE THAT CLEARING
2773						;TXINTE IN THE RECEIVER SVC ROUTINE
2774						;IS NOT STOPPING TXDONE INTERRUPTS
2775	013442				17\$:	
2776	013442	012767	000340	164326	MOV	#LEVEL7,\$S ;INHIBIT INTERRUPTS
2777	013450	104400			SCOPE	

```

2778
2779
2780
2781
2782
2783
2784
2785 013452 104402 .EOP: TYPE ;TYPE NAME OF TEST
2786 013454 016614 MEPASS
2787 013456 104410 013710 CONVRT .OUTCRY
2788 013462 104402 016335 TYPE .DEVICE
2789 013466 105767 165510 TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
2790 013472 001511 BEQ CCC ;NO, JUMP AROUND
2791 013474 005767 165516 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
2792 013500 001007 BNE RUNIT ;YES
2793 013502 104402 016347 TYPE .MCOV ;NO
2794 013506 016700 165504 MOV ACTREG,RO ;DISPLAY ACTREG
2795 013512 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
2796 ;SELECT SWITCHES & HIT (CONTINUE (PUT SWOO =1)
2797 013514 000167 165700 JMP .START ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2798 013520 062767 000010 165456 RUNIT: ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2799 013526 062767 000010 165456 ZERO: ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
2800 013534 000241 CLC
2801 013536 006167 165456 ROL
2802 013542 103410 BCS ROTADD ;UP DATE ROTATING POINTER
2803 2$ ;IS IT THE LAST DEVICE
2804 013544 036767 165450 165444 BIT ROTADD,ACTREG ;TO BE TESTED IN THIS PASS ?
2805 013552 001762 BEQ RUNIT ;TEST THIS DEVICE FOR ACTIVE STATUS
2806 013554 004767 000034 JSR PC,REPLAY ;IF NOT ACTIVE, TRY NEXT ADDRESS
2807 013560 000167 000174 JMP RESTRT ;CALCULATE NEW PARAMETERS
2808 013564 012767 000001 165426 2$: MOV #1,ROTADD ;YES IT WAS ACTIVE,TEST THIS DEVICE
2809 ;OK!,NOW SET UP ROTATING
2810 013572 016767 165410 165404 MOV KEEPADD,BASEADD ;POINTER FOR NEXT MULTIPLE PASS
2811 013600 016767 165410 165404 MOV KEEPIV,BASEIV ;RESTORE BASE ADDRESS
2812 013606 004767 000002 JSR PC,REPLAY ;RESTORE BASE INTERRUPT VECTORS
2813 013612 000441 BR CCC ;CALC NEW PARAMETERS
2814 013614 016767 165364 004040 REPLAY: MOV BASEADD,DUBASE ;JUMP AROUND REPLAY
2815 013622 004767 003702 JSR PC,DUADDR ;SET UP FOR NEW ADDRESSES
2816 013626 016767 165360 004350 MOV BASEIV,DURIV ;CREATE NEW ADDRESSES
2817 013634 062767 000002 165350 ADD #2,BASEIV ;CREATE DURIV
2818 013642 016767 165344 004336 MOV BASEIV,DURIS ;CREATE DURIS
2819 013650 062767 000002 165334 ADD #2,BASEIV
2820 013656 016767 165330 004324 MOV BASEIV,DUTIV ;CREATE DUTIV
2821 013664 062767 000002 165320 ADD #2,BASEIV
2822 013672 016767 165314 004312 MOV BASEIV,DUTIS ;CREATE DUTIS
2823 013700 016767 004300 165304 MOV DURIV,BASEIV ;RESTORE
2824 013706 000207 RTS PC
2825
2826 013710 000001 OUTCRY: 1
2827 013712 006 002 .BYTE 6,2
2828 013714 070160 RXCSR
2829
2830 013716 CCC:
2831 013716 005067 165212 CLR LSTERR ;CLEAR LAST ERROR PC
2832 013722 005067 165320 CLR ER?FLG ;CLEAR ERROR FLAG
2833 013726 005267 165176 INC PASCNT ;UPDATE PASS COUNT
  
```

```

2834 013732 016777 165172 165142      MOV      PASCNT,@LIGHTS      ;DISPLAY PASS COUNT
2835 013740 013701 000042              MOV      @#42,R1             ;CHECK FOR ACT-11 OR DDP
2836 013744 001405              BEQ      RESTRT              ;IF NOT, CONTINUE TESTING
2837 013746 000005              RESET
2838 013750 004711              LOGICAL: JSR      PC,(R1)
2839 013752 000240              NOP
2840 013754 000240              NOP
2841 013756 000240              NOP
2842 013760 012767 000340 164010  RESTRT: MOV      #340,PS          ;PREVENT INTERRUPTS (PRIO: 7)
2843 013766 104413              CKSWR                          ;CHECK FOR ^G
2844 013770 012767 003446 165116      MOV      #TST1,RTRN
2845 013776 000167 167444              JMP      TST1
2846
2847              ;SCOPE LOOP AND INTERA*ION HANDLER
2848
2849 014002              .SCOPE:
2850              ;***** START OF CODE FOR THE X OR TESTER *****
2851 014002 000424              BR      4$                    ;IF RUNNING ON THE X OR TESTER CHANGE
2852              ;THIS INSTRUCTION TO A 'NOP'(NOP=240)
2853 014004 013746 000004              MOV      @#4,-(SP)           ;SAVE CONTENTS OF ERROR VECTOR
2854 014010 012737 014030 000004      MOV      #1$,@#4            ;SET FOR TIME OUT
2855 014016 005737 177060              TST     @#177060            ;TIME OUT ON X OR ?
2856 014022 012637 000004              MOV      (SP)+,@#4           ;RESTORE ERROR VECTOR
2857 014026 000404              BR      2$                    ;GO TO NEXT TEST
2858 014030 022626              1$:  CMP      (SP)+,(SP)+       ;CLEAR THE STACK AFTER A TIMEOUT
2859 014032 012637 000004              MOV      (SP)+,@#4           ;RESTORE ERROR VECTOR
2860 014036 000403              BR      3$                    ;LOOP ON PRESENT TEST
2861 014040 016767 165052 165046      2$:  MOV      NEXT,RTRN        ;SET UP NEXT TEST IN RTRN
2862 014046 016716 165042              3$:  MOV      RTRN,(SP)         ;SET UP STACK FOR RTI
2863 014052 000002              RTI
2864 014054              4$:  ;***** END OF CODE FOR THE X OR TESTER *****
2865 014054 104413              CKSWR                          ;CHECK FOR ^G
2866 014056 032777 040000 165014      TTST: BIT      #SW14,@SWR       ;LOOP ON CURRENT TEST ?
2867 014064 001407              BEQ      1$
2868 014066 000432              BR      3$
2869 014070 105777 165010              TSTB   @TKCSR                ;TEST TTY FLAG
2870 014074 100027              BPL      3$
2871 014076 017700 165004              MOV      @TKDBR,R0           ;CLR DONE BIT
2872 014102 000412              BR      2$                    ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2873 014104 032777 004000 164766      1$:  BIT      #SW11,@SWR         ;INHIBIT ITERATIONS ?
2874 014112 001006              BNF      2$
2875 014114 005267 165004              INC      LPCNT
2876 014120 026767 165000 164774      CMP      LPCNT,ICOUNT        ;CHECK FOR ITERATION CNT FINISH
2877 014126 101412              BLOS    3$
2878 014130 105067 165112              2$:  CLRB   ERRFLG
2879 014134 005067 164764              CLR      LPCNT
2880 014140 012767 000005 164754      MOV      #5,ICOUNT           ;SET UP ITERATION COUNT
2881 014146 016767 164744 164740      MOV      NEXT,RTRN          ;SET UP NEXT TEST IN RTRN
2882 014154 016716 164734              3$:  MOV      RTRN,(SP)         ;SET UP STACK FOR RTI
2883 014160 000002              RTI
2884 014162 001407              BRW:  1407                    ;RESTORE 'BEQ 1$' INSTRUCTION
2885 014164 000432              BRX:  432                    ;RESTORE 'BR 3$' INSTRUCTION
2886
2887              ;CHECK FOR FREEZE ON CURRENT DATA
2888
2889 014166 104413              .SCOPE: CKSWR                ;CHECK FOR ^G

```

```

2890 014170 032777 001000 164702      BIT      #SW09,@SWR
2891 014176 001402                      BEQ      1$
2892 014200 016716 164714                      MOV      LOCK,(SP)
2893 014204 000002                      1$: RTI
2894
2895                      ;TELETYPE OUTPUT ROUTINE
2896
2897 014206 010546                      .TYPE:  MOV      R5,-(SP)
2898 014210 017605 000002                      MOV      @2(SP),R5
2899 014214 062766 000002 000002          ADD      #2,2(SP)
2900 014222 105715                      1$:  TSTB     (R5)                      ;LOOK FOR '0'
2901 014224 001406                      BEQ      3$
2902 014226 105777 164656          2$:  TSTB     @TPCSR                      ;TEST DONE BIT
2903 014232 100375                      BPL      2$
2904 014234 112577 164652          MOVSB   (R5)+,@TPDBR                    ;TYPE CHAR
2905 014240 000770                      BR       1$                              ;DO IT AGAIN UNTIL '0' IS SEEN
2906 014242 012605                      3$:  MOV      (SP)+,R5
2907 014244 000002                      RTI
2908
2909                      ;ASCII STRING INPUT ROUTINE
2910
2911 014246 010346                      .INSTR: MOV      R3,-(SP)
2912 014250 010446                      MOV      R4,-(SP)
2913 014252 017667 000004 000010          MOV      @4(SP),MSG
2914 014260 062766 000002 000004          ADD      #2,4(SP)
2915 014266 104402                      .INST1: TYPE
2916 014270 000000                      .MSG:   0
2917 014272 012704 017314                      MOV      #INBUF,R4
2918 014276 012703 000007                      MOV      #7,R3
2919 014302 105777 164576          1$:  TSTR     @TKCSR
2920 014306 100375                      BPL      1$
2921 014310 117714 164572          MOVSB   @TKDBR,(R4)
2922 014314 142714 000200          BICB    #200,(R4)
2923 014320 121427 000025          CMPB    (R4),#25                      ;IS IT <'U'
2924 014324 001003                      BNE     200$
2925 014326 104402 016524          TYPE ,MCRLF
2926 014332 000755                      BR      .INST1
2927 014334 122427 000015          200$: CMPB    (R4)+,#15
2928 014340 001423                      BEQ     INSTR2
2929 014342 117777 164540 164542          MOVSB   @TKDBR,@TPDBR
2930 014350 105777 164534          2$:  TSTB     @TPCSR
2931 014354 100375                      BPL      2$
2932 014356 005303                      DEC     R3
2933 014360 001350                      BNE     1$
2934 014362 000402                      BR      .INSTG
2935 014364 010346                      .INSTE: MOV      R3,-(SP)
2936 014366 010446                      MOV      R4,-(SP)
2937 014370 104402                      .INSTG: TYPE
2938 014372 016520                      MOV
2939 014374 005737 015662          MOV
2940 014400 001402                      TST     @RDSW
2941 014402 104402 016524          BEQ     400$
2942 014406 000727                      TYPE ,MCRLF
2943 014410 012604                      400$: BR      .INST1
2944 014412 012603                      INSTR2: MOV     (SP)+,R4
2945 014414 000002                      MOV     (SP)+,R3
RTI

```

```

2946
2947
2948
2949 014416 010546
2950 014420 010446
2951 014422 016605 000004
2952 014426 012567 000170
2953 014432 012567 000166
2954 014436 012567 000164
2955 014442 112567 000162
2956 014446 112567 000157
2957 014452 010566 000004
2958 014456 005005
2959 014460 012704 017314
2960 014464 122714 000015
2961 014470 001420
2962 014472 121427 000060
2963 014476 002415
2964 014500 121427 000067
2965 014504 003012
2966 014506 142714 000060
2967 014512 152405
2968 014514 122714 000015
2969 014520 001414
2970 014522 006305
2971 014524 006305
2972 014526 006305
2973 014530 000760
2974 014532 122714 000015
2975 014536 001003
2976 014540 005737 015662
2977 014544 001023
2978 014546 104404
2979 014550 000742
2980
2981
2982
2983 014552 020567 000046
2984 014556 101365
2985 014560 020567 000036
2986 014564 103762
2987 014566 136705 000036
2988 014572 001357
2989
2990
2991
2992 014574 016704 000026
2993 014600 010524
2994 014602 062705 000002
2995 014606 105367 000017
2996 014612 001372
2997 014614 012604
2998 014616 012605
2999 014620 000002
3000 014622 000000
3001 014624 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)+,ADRCNT
        MOV R5,4(SP)
PARAM1: CLR R5
        MOV #INBUF,R4
        CMPB #15,(R4)
        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 @WARDSW ;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$:  MOV DEVADR,R4
        MOV R5,(R4)+
        ADD #2,R5
        DECB ADRCNT
        BNE 1$
PARTI: MOV (SP)+,R4
        MOV (SP)+,R5
        RTI
LOLIM: 0
HILIM: 0

```

```

3002 014626 000000          DEVADR: 0
3003 014630 000000          LOBITS: 0
3004          014631          ADRCNT=LOBITS+1
3005
3006                          ;SAVE PC OF TEST THAT FAILED AND R0-R5
3007
3008 014632 016667 000004 164334 .SAV05: MOV     4(SP),SAVPC
3009
3010                          ;SAVE R0-R5
3011
3012 014640 010567 164324      SV05:  MOV     R5,SAVR5
3013 014644 010467 164316      MOV     R4,SAVR4
3014 014650 010367 164310      MOV     R3,SAVR3
3015 014654 010267 164302      MOV     R2,SAVR2
3016 014660 010167 164274      MOV     R1,SAVR1
3017 014664 010067 164266      MOV     R0,SAVR0
3018 014670 000002          RTI
3019
3020                          ;RESTORE R0-R5
3021
3022 014672 016700 164260      .RES05: MOV     SAVR0,R0
3023 014676 016701 164256      MOV     SAVR1,R1
3024 014702 016702 164254      MOV     SAVR2,R2
3025 014706 016703 164252      MOV     SAVR3,R3
3026 014712 016704 164250      MOV     SAVR4,R4
3027 014716 016705 164246      MOV     SAVR5,R5
3028 014722 000002          RTI
3029
3030                          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3031
3032 014724 104402          .CONVR: TYPE
3033 014726 016524          MCRLF
3034 014730 010046          .CNVRT: MOV     R0,-(SP)
3035 014732 010146          MOV     R1,-(SP)
3036 014734 010346          MOV     R3,-(SP)
3037 014736 010446          MOV     R4,-(SP)
3038 014740 010546          MOV     R5,-(SP)
3039 014742 017601 000012      MOV     @12(SP),R1
3040 014746 016767 002402 164174 MOV     TEMP,TEMP3
3041 014754 062766 000002 000012 ADD     #2,12(SP)
3042 014762 012167 000154      MOV     (R1)+,WRDCNT
3043 014766 112167 000152      1$:  MOV     (R1)+,CHRCNT
3044 014772 112167 000147      MOV     (R1)+,SPACNT
3045 014776 013167 000144      MOV     @ (R1)+,BINWRD
3046 015002 016704 000140      2$:  MOV     BINWRD,R4
3047 015006 116705 000132      MOV     CHRCNT,R5
3048 015012 012700 017354      MOV     #TEMP,R0
3049 015016 010403          3$:  MOV     R4,R3
3050 015020 042703 177770      BIC     #177770,R3
3051 015024 062703 000060      ADD     #060,R3
3052 015030 110320          MOV     R3,(R0)+
3053 015032 006204          ASR     R4
3054 015034 042704 100000      BIC     #100000,R4
3055 015040 006204          ASR     R4
3056 015042 006204          ASR     R4
3057 015044 005305          DEC     R5

```

```

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

```

3058	015046	001363				BNE	3\$	
3059	015050	012702	017414			MOV	#MDATA,R3	
3060	015054	114023			4\$:	MOVB	-(R0),(R3)+	
3061	015056	105367	000062			DECB	CHRCNT	
3062	015062	001374				BNE	4\$	
3063	015064	105767	000055			TSTB	SPACNT	
3064	015070	001405				BEQ	6\$	
3065	015072	112723	000040		5\$:	MOVB	#040,(R3)+	
3066	015076	105367	000043			DECB	SPACNT	
3067	015102	001373				BNE	5\$	
3068	015104	105013			6\$:	CLRB	(R3)	
3069	015106	104402				TYPE		
3070	015110	017414				MDATA		
3071	015112	005367	000024			DEC	WRDCNT	
3072	015116	001323				BNE	1\$	
3073	015120	016767	164024	002226		MOV	TEMP3,TEMP	
3074	015126	012605				MOV	(SP)+,R5	
3075	015130	012604				MOV	(SP)+,R4	
3076	015132	012603				MOV	(SP)+,R3	
3077	015134	012601				MOV	(SP)+,R1	
3078	015136	012600				MOV	(SP)+,R0	
3079	015140	000002				RTI		
3080	015142	000000				WRDCNT: 0		
3081	015144	000000				CHRCNT: 0		
3082		015145				SPACNT-CHRCNT+1		
3083	015146	000000				BINWRD: 0		
3084								
3085								
3086								
3087								
3088								
3089								
3090	015150	017605	000000			.SETFLG:MOV	@(SP),R5	
3091	015154	122767	000116	002132		CMPB	#'N',INBUF	;IS IT 'N'?
3092	015162	001002				BNE	1\$	
3093	015164	105015				CLRB	(R5) ;000	
3094	015166	000406				BR	2\$	
3095	015170	122767	000131	002116	1\$:	CMPB	#'Y',INBUF	;IS IT 'Y'?
3096	015176	001005				BNE	3\$	
3097	015200	112715	177777			MOVB	#-1,(R5)	;377
3098	015204	062716	000002		2\$:	ADD	#2,(SP)	
3099	015210	000002				RTI		
3100	015212	104404			3\$:	INSTER		;RETRY
3101	015214	000755				BR	.SETFLG	
3102								
3103								
3104								
3105								
3106								
3107	015216	011646				.TRPSR:MOV	(SP),-(SP)	;GET PC OF RETURN
3108	015220	162716	000002			SUB	#2,(SP)	;=PC OF TRAP
3109	015224	017616	000000			MOV	@(SP),(SP)	;GET TRP
3110	015230	006316			TRPOK:	ASL	(SP)	;MULTIPLY TRAP ARG BY 2
3111	015232	042716	177001			BIC	#177001,(SP)	;CLEAR UNWANTED BITS
3112	015236	062716	001366			ADD	#.TRPTAB,(SP)	;POINTER TO SUBROUTINE ADDRESS
3113	015242	017616	000000			MOV	@(SP),(SP)	;SUBROUTINE ADDRESS

```

3114 015246 000136          JMP      @ (SP)+          ;GO TO SUBROUTINE
3115
3116                          ;ERROR HANDLER
3117
3118 015250 104413          .HLT:   CKSWR          ;CHECK FOR ^G
3119 015252 032777 020000 163620 BIT      #SW13,@SWR      ;INHIBIT ERROR TYPE OUT :
3120 015260 001061          BNE     HALTS
3121 015262 021667 163646  CMP     (SP),LSTERR
3122 015266 001404          BEQ     1$
3123 015270 011667 163640  MOV     (SP),LSTERR
3124 015274 105067 163746  CLR    ERRFLG
3125 015300 104406          1$:    SAVO5
3126 015302 011605          MOV     (SP),R5
3127 015304 162705 000002  SUB     #2,R5
3128 015310 011504          MOV     (R5),R4
3129 015312 006304          ASL    R4
3130 015314 061504          ADD    (R5),R4
3131 015316 006304          ASL    R4
3132 015320 042704 177001  BIC    #177001,R4
3133 015324 062704 020130  ADD    #.ERRTAB,R4
3134 015330 012467 000040  MOV    (R4)+,ERRMSG
3135 015334 012467 000046  MOV    (R4)+,DATAHD
3136 015340 011467 000054  MOV    (R4),DATABP
3137 015344 105767 163676  TSTB  ERRFLG
3138 015350 001403          BEQ    TYPMSG
3139 015352 005767 000042  TST   DATABP
3140 015356 001014          BNE    TYPDAT
3141 015360 104410          TYPMSG: CONVRT
3142 015362 015512          ERTAB0
3143 015364 112767 177777 163654 MOVB   #-1,ERRFLG
3144 015372 104402          TYPE
3145 015374 000000          ERRMSG: 0
3146 015376 005767 000004  TST   DATAHD
3147 015402 001402          BEQ   TYPDAT
3148 015404 104402          TYPE
3149 015406 000000          DATAHD: 0
3150 015410 005767 000004  TYPDAT: TST   DATABP
3151 015414 001402          BEQ   RESREG
3152 015416 104410          CONVRT
3153 015420 000000          DATABP: 0
3154 015422 104407          RESREG: RESO5
3155 015424 005777 163450  HALTS: TST   @SWR
3156 015430 100005          BPL   EXITER
3157 015432 010046          PUSHRO
3158 015434 016600 000002  MOV    2(SP),R0
3159 015440 000000          HALT
3160 015442 012600          POPRO
3161 015444 104413          EXITER: CKSWR          ;CHECK FOR ^G
3162 015446 005267 163460  INC    ERRCNT
3163 015452 032777 000400 163420 BIT    #SW08,@SWR      ;LOOP ON ERROR ?
3164 015460 001007          BNE    1$
3165 015462 032777 002000 163410 BIT    #SW10,@SWR      ;ESCAPE TO NEXT ON ERROR ?
3166 015470 001407          BEQ    2$
3167 015472 016767 163420 163414 MOV    NEXT,RTRN      ;SET UP FOR NEXT TEST
3168 015500 012706 001100 1$:    MOV    #STACK,SP    ;REINITIALIZE SP
3169 015504 000177 163404  JMP    @RTRN
  
```



```

3170 015510 000002          2$: RTI
3171 015512 000001          FRTABC: 1
3172 015514 006 002      .BYTE 6,2
3173 015516 001174      SAVPC
3174      ;ENTER HERE ON POWER FAILURE
3175
3176
3177 015520 010046      .PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
3178 015522 010146      MOV R1,-(SP)
3179 015524 010246      MOV R2,-(SP)
3180 015526 010346      MOV R3,-(SP)
3181 015530 010446      MOV R4,-(SP)
3182 015532 010546      MOV R5,-(SP)
3183 015534 016746 162264  MOV 24,-(SP)
3184 015540 010667 163426  MOV SP,SAVSP ;SAVE STACK POINTER
3185 015544 012767 015556 162252  MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
3186 015552 000000      HALT ;HALT ON POWER DOWN NORMAL
3187 015554 000777      1$: BR 1$
3188
3189      ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3190
3191 015556 016706 163410  RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER
3192 015562 012605      MOV (SP)+,R5 ;RESTORE R0-R5
3193 015564 012604      MOV (SP)+,R4
3194 015566 012603      MOV (SP)+,R3
3195 015570 012602      MOV (SP)+,R2
3196 015572 012601      MOV (SP)+,R1
3197 015574 012600      MOV (SP)+,R0
3198 015576 012767 015520 162220  MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
3199 015604 012767 000340 162164  MOV #340,PS
3200 015612 012706 001100      MOV #STACK,SP
3201 015616 005067 001532      CLR TEMP
3202 015622 005267 001526      1$: INC TEMP
3203 015626 001375      BNE 1$
3204 015630 104410      CONVRT
3205 015632 015654      PFTAB
3206 015634 104402      TYPE
3207 015636 016527      MPFAIL
3208 015640 005067 163402      CLR ERRFLG
3209 015644 005067 163264      CLR LSTERR
3210 015650 000177 163240      .MP @RTRN
3211 015654 000001      PFTAB: 1
3212 015656 006 002      .BYTE 6,2
3213 015660 001114      RTRN
3214
3215
3216      ;CHECK SWITCH REGISTER ROUTINE. (CHECKS FOR *G TO ALLOW CHANGING
3217      ;OF LOC.176.
3218      ;LOCATIONS USED:
3219 015662 000000      RDSW: .WORD 0
3220
3221
3222 015664 005737 000042      .KSWR: TST @#42
3223 015670 001042      BNE OUT
3224 015672 022767 000176 163200  CMP #C.REG.SWR ;SOFTWARE SWITCH REGISTER PRESENT
3225 015700 001036      BNF OUT ;NO, GET OUT
  
```

3226	015702	105777	163176			TSTB	@TKCSR		:YES, WAIT FOR
3227	015706	100033				BPL	OUT		:READY, GET CHARACTER
3228	015710	017767	163172	176352		MOV	@TKDBR, .MSG		:AND STRIP OFF
3229	015716	042767	177600	176344		BIC	#177600, .MSG		:THE GARBAGE
3230	015724	122767	000007	176336		CMPB	#7, .MSG		:IS IT A <^G>
3231	015732	001021				BNE	OUT		
3232	015734	104402	016012			TYPE, \$CNTG			
3233	015740	005137	015662		.CNTLU:	COM	@#RDSW		
3234	015744	104402	016017			TYPE, \$MSWR			
3235	015750	104411	016004			CNVRT, SWREGC			
3236	015754	104403	016027			INSTR, \$MNEW			
3237	015760	104405				PARAM			
3238	015762	000000				0			
3239	015764	177777				177777			
3240	015766	000176				SWREG			
3241	015770	000	001		.BYTE	0, 1			
3242	015772	104402	016524			TYPE, MCR, F			
3243	015776	005037	015662		OUT:	CLR	@#RDSW		
3244	016002	000002				RTI			
3245	016004	000001			SWREGC:	1			
3246	016006	006	002		.BYTE	6, 2			
3247	016010	000176				SWREG			
3248	016012	005015	043536	000	\$CNTG:	.ASCIZ	<15><12>/^G/		
3249	016017	015	051412	051127	\$MSWR:	.ASCIZ	<15><12>/SWR= /		
3250	016024	020075	000						
3251	016027	040	047040	053505	\$MNEW:	.ASCIZ	/ NEW= /		
3252	016034	020075	000						
3253		016040			.EVEN				
3254	016040	005015	042012	030525	MTITLE:	.ASCIZ	<15><12><12>/DU11 CZDUE-D TAPE E /<15><12>		
3255	016046	020061	055103	052504					
3256	016054	026505	020104	040524					
3257	016062	042520	042440	006440					
3258	016070	000012							
3259	016072	005015	042526	052103	MVECTOR:	.ASCIZ	<15><12>/VECTOR ADDRESS- /		
3260	016100	051117	040440	042104					
3261	016106	042522	051523	000055					
3262	016114	005015	051461	020124	MREGAD:	.ASCIZ	<15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- /		
3263	016122	042504	044526	042503					
3264	016130	020072	042522	042503					
3265	016136	053111	051105	041440					
3266	016144	047117	051124	046117					
3267	016152	051040	043505	051511					
3268	016160	042524	020122	042101					
3269	016166	051104	051505	026523					
3270	016174	000							
3271	016175	015	040412	042522	MMULT:	.ASCIZ	<15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)- /		
3272	016202	054440	052517	051010					
3273	016210	047125	044516	043516					
3274	016216	046440	046125	044524					
3275	016224	046120	020105	042504					
3276	016232	044526	042503	020123					
3277	016240	020077	054450	047440					
3278	016246	020122	024516	000055					
3279	016254	005015	040514	052123	MLASTD:	.ASCIZ	<15><12>/LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- /		
3280	016262	042040	053105	041511					
3281	016270	035105	042522	042503					

3282 016276 053111 051105 041440  
3283 016304 047117 051124 046117  
3284 016312 051040 043505 051511  
3285 016320 042524 020122 042101  
3286 016326 051104 051505 026523  
3287 016334 000  
3288 016335 075 042504 044526  
3289 016342 042503 020040 000  
3290 016347 015 044012 053517  
3291 016354 047040 053517 041040  
3292 016362 047522 047127 041440  
3293 016370 053517 020077 027056  
3294 016376 051456 046105 041505  
3295 016404 020124 047523 042515  
3296 016412 044124 047111 020107  
3297 016420 047524 051040 047125  
3298 016426 040040 041501 051124  
3299 016434 043505 000  
3300 016437 015 047412 052125  
3301 016444 047440 020106 040522  
3302 016452 043516 035105 042522  
3303 016460 054524 042520 046040  
3304 016466 051501 020124 042504  
3305 016474 044526 042503 051040  
3306 016502 041530 051123 040440  
3307 016510 042104 042122 051522  
3308 016516 000055  
3309 016520 020040 000077  
3310 016524 005015 000  
3311 016527 040 050040 053517  
3312 016534 051105 043040 044501  
3313 016542 052514 042522 020054  
3314 016550 051120 043517 040522  
3315 016556 020115 042522 052123  
3316 016564 051101 020124 052101  
3317 016572 052040 051505 020124  
3318 016600 047111 050040 047522  
3319 016606 051107 051505 000123  
3320 016614 005015 047105 020104  
3321 016622 043117 050040 051501  
3322 016630 020123 040524 042520  
3323 016636 042440 000  
3324 016641 015 051012 000  
3325 016645 015 052012 051505  
3326 016652 020124 041520 000055  
3327 016660 005015 047514 045503  
3328 016666 047440 020116 042523  
3329 016674 042514 052103 042105  
3330 016702 052040 051505 037524  
3331 016710 024040 020131 051117  
3332 016716 047040 026451 000  
3333 016723 015 042012 020125  
3334 016730 051120 047511 044522  
3335 016736 054524 046040 053105  
3336 016744 046105 000055  
3337 016750 005015 020043 043117

DEVICE: .ASCIZ /=DEVICE /

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

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

MOM: .ASCIZ / ?/

MCRLF: .ASCIZ <15><12>

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

MEPASS: .ASCIZ <15><12>/END OF PASS TAPE E/

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

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

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

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

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

```
3338 016756 051440 047131 020103
3339 016764 044103 051101 020123
3340 016772 042523 042514 052103
3341 017000 042105 024040 030440
3342 017006 047440 020122 024462
3343 017014 000055
3344 017016 005015 051511 051440 MWIRE6: .ASCIIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/
3345 017024 041505 054040 044515
3346 017032 020124 052512 050115
3347 017040 051105 021440 020066
3348 017046 047111 020077 054450
3349 017054 047440 020122 024516
3350 017062 000055
3351 017064 005015 051511 051440 MWIRE5: .ASCIIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/
3352 017072 041505 051040 041505
3353 017100 045040 046525 042520
3354 017106 020122 032443 044440
3355 017114 037516 024040 020131
3356 017122 051117 047040 026451
3357 017130 000
3358 017131 015 044412 020123 MWIRE4: .ASCIIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/
3359 017136 050117 020124 046103
3360 017144 020122 047105 041101
3361 017152 042514 045040 046525
3362 017160 042520 020122 032043
3363 017166 044440 037516 024040
3364 017174 020131 051117 047040
3365 017202 026451 000
3366 017205 015 044412 020123 NEXTJ: .ASCIIZ <15><12>/IS THE TEST CONNECTOR INSTALLED?(Y OR N)-/
3367 017212 044124 020105 042524
3368 017220 052123 041440 047117
3369 017226 042516 052103 051117
3370 017234 044440 051516 040524
3371 017242 046114 042105 037440
3372 017250 054450 047440 020122
3373 017256 024516 000055
3374 017262 006412 020040 020040 MSTATUS: .ASCIIZ <12> <15>/ STATUS MAP / <12> <15>
3375 017270 052123 052101 051525
3376 017276 020040 046440 050101
3377 017304 020040 020040 005040
3378 017312 000015
3379 .EVEN
3380
3381 ;BUFFERS FOR INPUT-OUTPUT
3382
3383 017314 000040 INBUF: .BLKB 40
3384 017354 000040 TEMP: .BLKB 40
3385 017414 000040 MDATA: .BLKB 40
3386 ;.....
3387 ;UTILITIES
3388 ;.....
3389
3390 ;THIS UTILITY CALCULATES PRIORITY LEVEL
3391 017454 006367 000044 DULEV: ASL DUPRT :SHIFT LEFT
3392 017460 006367 000040 ASL DUPRT :
3393 017464 006367 000034 ASL DUPRT :
```

```

3394 017470 006367 000030      ASL      DUPRT      ;
3395 017474 006367 000024      ASL      DUPRT      ;
3396 017500 016767 000020 000020  MOV      DUPRT,LESS1 ;MOVE THIS TO LESS1
3397 017506 162767 000001 000012  SUB      #1,LESS1    ;CREATE LESS1
3398 017514 042767 000037 000004  BIC      #37,LESS1   ;CLEAR TNZVC
3399 017522 000207      RTS      PC
3400 017524 000240      DUPRT:  LEVEL5
3401 017526 000200      LESS1:  LEVEL4 ;LEVEL TO ALLOW INTERRUPTS
3402
3403      ;NEW DU ADDRESSES
3404 017530 016767 000126 000422  DUADDR:  MOV      DUBASE,RXCSR ;XXX0
3405 017536 005267 000120      INC      DUBASE
3406 017542 016767 000114 000002  MOV      DUBASE,HRXCSR ;XXX1
3407 017550 005267 000106      INC      DUBASE
3408 017554 016767 000102 000402  MOV      DUBASE,RXDBUF ;XXX2
3409 017562 016767 000074 000400  MOV      DUBASE,PARCSR ;XXX2
3410 017570 005267 000066      INC      DUBASE
3411 017574 016767 000062 000364  MOV      DUBASE,HRXDBUF ;XXX3
3412 017602 016767 000054 000362  MOV      DUBASE,HPARCSR ;XXX3
3413 017610 005267 000046      INC      DUBASE
3414 017614 016767 000042 000352  MOV      DUBASE,TXCSR ;XXX4
3415 017622 005267 000034      INC      DUBASE
3416 017626 016767 000030 000342  MOV      DUBASE,HTXCSR ;XXX5
3417 017634 005267 000022      INC      DUBASE
3418 017640 016767 000016 000332  MOV      DUBASE,TXDBUF ;XXX6
3419 017646 005267 000010      INC      DUBASE
3420 017652 016767 000004 000322  MOV      DUBASE,HTXDBUF ;XXX7
3421 017660 000207      RTS      PC
3422 017662 000000      DUBASE: 0
3423
3424      ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3425      ;INFORMATION CONTAINED IN TEMP1 AND IT IS
3426      ;SHIFTED IN BY THE CONTENTS OF SHIFT
3427 017664 042777 040000 000302  RPOKE:  BIC      #MTDATA,@TXCSR
3428 017672 005067 161250      CLR      TEMP2
3429 017676 006067 161242      ROR      TEMP1 ;FORCE CARRY
3430 017702 006067 161240      ROR      TEMP2 ;PICK UP CARRY IN BIT 15
3431 017706 006267 161234      ASR      TEMP2 ;SHIFT INTO BIT 14
3432 017712 042767 100000 161226  BIC      #BIT15,TEMP2 ;CLR BIT 15
3433 017720 056777 161222 000246  BIS      TEMP2,@TXCSR ;POKE MAINT DATA
3434 017726 042777 020000 000240  BIC      #CLK,@TXCSR ;POKE CLK
3435 017734 052777 020000 000232  BIS      #CLK,@TXCSR ;
3436 017742 005367 161172      DEC      SHIFT
3437 017746 001346      BNE      RPOKE
3438 017750 000207      RTS      PC
3439
3440      ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3441 017752 016767 161166 161166  ODD8:  MOV      TEMP1,TEMP2 ;SAVE TEMP1
3442 017760 005067 161164      CLR      TEMP3
3443 017764 012727 000010      MOV      #8,(PC)+
3444 017770 000000      '8:  0
3445 017772 006067 161150      '8:  ROR      TEMP2
3446 017776 005567 161146      '8:  ADC      TEMP3
3447 020002 005367 177762      '8:  DEC      '8
3448 020006 001371      '8:  BNE      '8
3449 020010 006067 161134      '8:  ROR      TEMP2
  
```

```

3450 020014 103404          BCS      3$
3451 020016 052767 000400 161120      BIS      #BIT8,TEMP1      ;SET ODD PARITY
3452 020024 000403          BR       4$
3453 020026 042767 000400 161110 3$:      BIC      #BIT8,TEMP1      ;CLR EVEN PARITY
3454          000207          ;TEMP1 NOW HAS ODD PARITY CHARACTER
3455 020034 000207          4$:      RTS      PC
3456
3457          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3458 020036 016767 161102 161102  EVEN8:  MOV      TEMP1,TEMP2      ;SAVE TEMP1
3459 020044 005067 161100          CLR      TEMP3
3460 020050 012727 000010          MOV      #8,(PC)+
3461 020054 000000          1$:      0
3462 020056 006067 161064          2$:      ROR      TEMP2
3463 02006~ 005567 161062          ADC      TEMP3
3464 020066 0^5367 177762          DEC      1$
3465 020072 001371          BNE     2$
3466 020074 006067 16105C          ROR      TEMP3
3467 020100 103004          BCC     3$
3468 020102 052767 000400 161034      BIS      #BIT8,TEMP1      ;SET EVEN PARITY
3469 020110 000403          BR       4$
3470 020112 042767 000400 161024 3$:      BIC      #BIT8,TEMP1      ;CLR ODD PARITY
3471          ;TEMP1 NOW HAS EVEN PARITY CHARACTER
3472 020120 000207          4$:      RTS      PC
3473 020122 062716 000002  TRPREG: ADD     #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3474          ;IN MAIN PART OF THE PROGRAM
3475 020126 000002          RTI
3476          ;ERROR HLT TABLE
3477 020130 020214          .ERRTAB: EM0      ;HLT 0 BIT ERROR (GENERAL)
3478 020132 000000          0
3479 020134 000000          0
3480 020136 020230          EM1      ;HLT 1 REGISTER ERROR
3481 020140 020401          DH1
3482 020142 020422          DT1
3483 020144 020272          EM2      ;HLT 2 RECEIVER ERROR
3484 020146 020401          DH1
3485 020150 020422          DT1
3486 020152 020334          EM3      ;HLT 3 TRANSMITTER ERROR
3487 020154 020401          DH1
3488 020156 020422          DT1
3489          ;DEFAULT DU ADDRESSES
3490 020160 160040          RXCSR: 160040
3491 020162 160041          HRXCSR: 160041
3492 020164 160042          RXDBUF: 160042
3493 020166 160043          HPXDBUF: 160043
3494 020170 160042          PARCSR: 160042
3495 020172 160043          HPARCSR: 160043
3496 020174 160044          TXCSR: 160044
3497 020176 160045          HTXCSR: 160045
3498 020200 160046          TXDBUF: 160046
3499 020202 160047          HTXDBUF: 160047
3500          ;DEFAULT DU VECTORS
3501 020204 000770          DURIV: 770      ;REC INTR VECTOR
3502 020206 000772          DURIS: 772      ;REC INTR STATUS
3503 020210 000774          DUTIV: 774      ;XMIT INTR VECTOR
3504 020212 000776          DUTIS: 776      ;XMIT INTR STATUS
3505          ;ERROR MESSAGES
  
```

```
3506 020214 036440 042440 051122 EMO: .ASCIZ / = ERROR PC/
3507 020222 051117 050040 000103
3508 020230 036440 051040 043505 EM1: .ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3509 020236 051511 042524 020122
3510 020244 051105 047522 020122
3511 020252 041520 005015 051001
3512 020260 043505 051511 042524
3513 020266 020122 000040
3514 020272 036440 051040 041505 EM2: .ASCIZ / - RECEIVER ERROR PC/<15><12><1>/REGISTER /
3515 020300 044505 042526 020122
3516 020306 051105 047522 020122
3517 020314 041520 005015 051001
3518 020322 043505 051511 042524
3519 020330 020122 000040
3520 020334 036440 052040 040522 EM3: .ASCIZ / - TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3521 020342 051516 044515 052124
3522 020350 051105 042440 051122
3523 020356 051117 050040 006503
3524 020364 000412 042522 044507
3525 020372 052123 051105 020040
3526 020400 000
3527 :DATA HEADERS FOR ERROR MESSAGES
3528 020401 105 050130 041505 DH1: .ASCIZ /EXPECTED ACTUAL/
3529 020406 042524 020104 040440
3530 020414 052103 040525 000114
3531 .EVEN
3532 :DATA TABLES FOR ERROR MESSAGES
3533 020422 000003 DT1: {
3534 020424 006 004 .BYTE 6,4
3535 020426 001164 .SAVR3 :REGISTER
3536 020430 006 004 .BYTE 6,4
3537 020432 001156 .SAVR0 :EXPECTED DATA
3538 020434 006 002 .BYTE 6,2
3539 020436 001160 .SAVR1 :ACTUAL DATA
3540 000001 .END
```











PRTDU	001324	946#	1072	1233*										
PS	= 177776	728#	1008*	1213*	2089*	2092*	2114*	2140*	2143*	2187*	2220*	2224*	2235*	2248*
		2289*	2322*	2333*	2348*	2361*	2372*	2386*	2399*	2406*	2411*	2424*	2444*	2453*
		2461*	2478*	2512*	2522*	2537*	2550*	2564*	2586*	2606*	2616*	2672*	2693*	2710*
		2742*	2776*	2842*	3199*									
PUSHRO=	010046	735#	3157											
PUSH1S=	005746	733#												
PUSH2S=	024646	737#												
RDSW	015662	2939	2976	3219#	3233*	3243*								
REACT=	004000	774#	1359	1375	1384	1761	1830	1837	1842	1852	1862	1886		
REGACT	001320	944#	1063	1231*										
REPLAY	013614	2806	2812	2814#										
RESEC	001302	932#	1055	1222*										
RESREG	015422	3151	3154#											
RESTAR	015556	3185	3191#											
RESTRT	013760	2807	2836	2842#										
RESOS =	104407	986#	3154											
RING =	040000	771#												
RINTEN=	000100	779#	2144	2155	2180	2240	2253	2257	2282	2591	2612	2615	2680	2695
		2718	2743											
RISDU	001334	950#	1068	1235*										
RIVDU	001326	947#	1067	1234*										
ROTADD	001220	902#	1064*	1128*	1140*	1142	1144*	1149	1152*	1232	2801*	2804	2808*	
RPOKE	017664	1358	1365	1381	1758	1771	1823	1836	1850	1861	1883	1896	3427#	3437
RTRN	001114	856#	1018*	1314	1318*	1320	2844*	2861*	2862	2881*	2882	3167*	3169	3210
		3213												
RTS	= 000004	783#												
RUNA	- *****	1	53	71	149	665								
RUN3	- *****	1	53	71	149	665								
RUNC	- *****	1	53	71	149	665								
RUND	- *****	1	53	71	149	665								
RUNE	= 000000	1	53	71	149	665								
RUNF	= *****	21	59	71	152	670								
RUNIT	013520	2792	2798#	2805										
RXCSR	020160	1074*	1240	1349*	1359	1375	1384	1570*	1581	1610	1742*	1761	1765	1772
		1812*	1830	1837	1841*	1842	1851*	1852	1856	1862	1866	1886	1890	1897
		1939*	1978	2007	2041*	2060	2085*	2086	2093	2097*	2103*	2104*	2112*	2134*
		2144*	2145	2155*	2180*	2203*	2214*	2225	2232*	2240*	2249	2253*	2257*	2282*
		2591*	2607	2612*	2615*	2625	2653*	2680*	2695*	2711	2718*	2743*	2754	2828
		3404*	3490#											
RXDBUF	020164	1076*	1242	1417	1426	1460	1571	1588	1615	1750	1777	1820	1871	1902
		1928	1983	2012	2042	2064	2152	2177	2179	2233	2279	2281	2587	2608
		2673	2712	3408*	3492#									
		778#												
RXDONE-	000200	787#	1614											
RXERR -	100000	863#	3008*	3173										
SAVPC	001174	876#	3017*	3022	3537									
SAVRO	001156	877#	3016*	3023	3539									
SAVR1	001160	878#	3015*	3024										
SAVR2	001162	879#	3014*	3025	3535									
SAVR3	001164	880#	3013*	3026										
SAVR4	001166	881#	3012*	3027										
SAVR5	001170	882#	3184*	3191										
SAVSP	001172	984#	3125											
SAVOS -	104406	972#	1388	1469	1538	1625	1672	1724	1782	1916	2020	2073	2115	2188
SCOPE -	104400	2290	2349	2387	2428	2479	2566	2633	2777					







HLT	739#	1361	1377	1386	1429	1463	1488	1496	1502	1532	1583	1591	1612	1618	1666
	1714	1763	1767	1774	1780	1832	1839	1844	1854	1858	1864	1868	1874	1888	1892
	1899	1905	1980	1986	2009	2015	2062	2067	2095	2113	2147	2183	2185	2227	2241
	2251	2285	2287	2330	2336	2368	2376	2425	2450	2465	2467	2518	2525	2529	2543
	2553	2602	2623	2631	2700	2748	2752	2760	2765	2771					
PRGEND	678#	2778													
PRGFRT	678#	679													
PUSSYF	678#	1942	1957	1990											
RSETUP	678#	1340	1400	1434	1732	1802	2125	2205							
TSETUP	678#	1559	1635	1684	1929	2031	2299	2489	2643						
\$BEGIN	678#	1209													
\$BINAR	678#														
\$BUFFE	678#	3380													
\$CABLE	678#														
\$CATCH	678#	822													
\$CLRVE	678#	1024													
\$CONVR	678#	3029													
\$DNA	678#														
\$EOP	678#	2778													
\$GETFL	678#	1120	1193	1197	1201	1205	1296								
\$GETPA	678#	1099	1110	1130	1156	1165	1309								
\$GETSY	678#	1175													
\$HEADE	678#	679													
\$HLT	678#	3115													
\$INSTR	678#	2908													
\$ISOB	678#														
\$MATCH	678#														
\$MRR	678#														
\$MRRW	678#														
\$MRW	678#														
\$MSG	678#	3254													
\$PARAM	678#	2946													
\$PFAIL	678#	3174													
\$POKE	678#	1492	1498	1505	1520	1573	1576	1603	1651	1659	1699	1707	2046	2051	2312
	2315	2502	2505												
\$POKER	678#	1353	1368	1371	1409	1412	1443	1747	1816	2135	2215				
\$RCNET	678#														
\$RECAC	678#														
\$REG	678#	3005													
\$RESET	678#	1340	1342	1400	1402	1434	1436	1559	1561	1635	1637	1684	1686	1732	1734
	1802	1804	1929	1931	2031	2033	2082	2125	2127	2205	2207	2299	2301	2358	2396
	2441	2489	2491	2577	2579	2643	2645								
\$RXACT	678#														
\$SCOPE	678#	2846													
\$SCOPI	678#	2886													
\$SETFL	678#	3084													
\$SETVE	678#	823													
\$START	678#	1000													
\$STRIP	678#														
\$SYMBO	678#	691													
\$SYNCR	678#	1349	1812												
\$TRAPS	678#	964													
\$TRPAR	678#														
\$TRPDE	678#	972	974	976	978	980	982	984	986	988	990	992	994	996	
\$TRPSR	678#	3102													
\$TSTNO	678#	1336	1398	1477	1556	1632	1681	1730	1795	1924	2029	2078	2122	2195	2296



	2355	2394	2439	2486	2575	2641
\$TYPE	678#	2894				
\$UNIBU	678#					
\$VARIA	678#	842				
\$WORDF	678#					
\$WORDO	678#					
\$WORDP	678#					

. ABS. 020440 000

ERRORS DETECTED: 0

CZDUE.D.BIN,CZDUE.D.SEQ/CRF/SOL/NL:TOC-CZDU11.HLO/EQ:RUNE,CZDU11.PAR,CZDU11.KET,CZDUE.D.P11  
RUN-TIME: 7 11 1 SECONDS  
RUN-TIME RATIO: 72/20=3.4  
CORE USED: 19K (37 PAGES)