

DU11

OFF-LINE COMBINED TEST
CZDUFD0

AH-8697D-MC

COPYRIGHT © 73-78

FICHE 1 OF 1

JAN 1979

digital

MADE IN USA

This microfiche card contains a grid of frames. The first column on the left contains frames with text, likely serving as a table of contents or index. The remaining columns contain frames with data, which appears to be a combined test report. The data is organized into columns and rows, with some sections containing what looks like a list or table of results. The text is very small and difficult to read due to the low resolution of the scan.

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

PRODUCT CODE: AC-8696D-MC

PRODUCT NAME: CZDUFDO DU11 OFFLINE COMBINED TESTS

RELEASE DATE: JUN 1978

MAINTAINER : DIAGNOSTICS

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

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

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

COPYRIGHT (C) 1973, 1978 BY DIGITAL EQUIPMENT CORPORATION

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DU11'S. THIS MEANS THAT 16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE, AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL HALTED.

1. THE DU11 OFFLINE COMBINED TESTS VERIFY THAT THE TRANSMITTER AND RECEIVER CAN TALK THRU THE EXTERNAL MODEM CABLE PROVIDING THAT THE H315 CONNECTOR IS ON

2. REQUIREMENTS

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

DU11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE

THE PROGRAM LOADS AND RUNS IN 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
STARTING ADDRESS

4.2

THE STARTING ADDRESS FOR ALL TESTS IS 000200

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

4.3 PROGRAM AND/OR OPERATOR ACTION

4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

4.3.1.5 THE PROGRAM WILL TYPE 'DU11 CZDUF-D TAPE F' (ONCE ONLY)

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

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

4.3.1.7 THE PROGRAM WILL TYPE 'R' TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE 'R' AND WILL COMMENCE TESTING

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 LOAD ADDRESS 000200

4.3.3.2 SET SW00=1

4.3.3.3 PRESS START

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

4.3.3.4 THE PROGRAM WILL TYPE "1ST DEVICE: RECEIVER CONTROL REGISTER
ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL
REGISTER ADDRESS OF THE DU11 TO BE TESTED
FOLLOWED BY A <CARRIAGE RETURN>

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

4.3.3.6 THE PROGRAM WILL TYPE 'VECTOR ADDRESS-' AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD

4.3.3.7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS
FOR THE DU11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

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

4.3.3.8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?"
(Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

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

4.3.3.10 THE PROGRAM WILL TYPE 'LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DU11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

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

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

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DU11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

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

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

4.3.3.12 THE PROGRAM WILL TYPE 'DU PRIORITY LEVEL-' AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

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

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

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

4.3.3.16 THE PROGRAM WILL TYPE " IS SEC XMIT JUMPER #6 IN ? (Y OR N)-"
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

4.3.3.18 THE PROGRAM WILL TYPE "IS SEC REC JUMPER # 5 IN ?
(Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

4.3.3.20 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE JUMPER
4 IN ? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

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

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

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

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

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"
AND WILL REPEAT THE MESSAGE OF 4.3.3.22

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

4.3.4 PROGRAM RESTART WITH SW01=1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
...IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED,LOAD 000200,
AND SELECT SW00=1 AND ANSWER 'NO' TO THE MULTIPLE DEVICE QUESTION
SEE 4.3.3

4.3.4.1 LOAD 000200

4.3.4.2 SET SW01=1

4.3.4.3 PRESS START

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

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

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

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

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

4.3.5 PROGRAM RESTART WITH SW02 =1

NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 LOAD ADDRESS 000200

4.3.5.2 SET SW02 =1

NOTE: IT MAY BE ADVANTAGEOUS TO SET SW01=1 (OPTIONAL)

4.3.5.3 PRESS START

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

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

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

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

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

4.4 STATUS MAP

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

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

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

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

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

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

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

THE MAP WILL LOOK LIKE:

STATUS MAP

1300/ 177777

1302/ 000000

1304/ 177777

THE BYTES ARE DEFINED AS FOLLOWS:

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

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

5. OPERATING PROCEDURE

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

CONTROL:

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

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

5.1

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

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

6. ERRORS

6.1 ERROR HALTS
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

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

6.1.1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

REFER TO THE ABOVE 'HLT' IN DIAGNOSTIC FOR ERROR DESCRIPTION

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

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

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

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

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

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

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

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

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

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

- 7.2.1 IF DEVICE 0 IS TO BE DISQUALIFIED, SIMPLY RESTART
PROGRAM WITH SW00 =1 AND OMIT THE FIRST DEVICE.
- 7.2.2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
ARE TO BE DISQUALIFIED....LOAD THE LOCATION OF ACTREG:
OBSERVE THE ACTIVE BITS (ACTIVE =1, NONACTIVE = 0)
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED
 - 7.2.2.1 TO RESTART...LOAD 000200 IN SWR AND DEPRESS START....
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING.
 - 7.2.2.2ORLOAD 000200 WITH SW00 =1 AND DEPRESS START....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
.....THE PROGRAM WILL CONTINUE WITH DEVICE 0
 - 7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM
WILL TYPEOUT AN ERROR MESSAGE.....LOAD & START AT 000200

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

- 7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
LOCATION 'HOLD:' MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES.
PRESENTLY 'HOLD:' =20 IS SUFFICIENT TIME ON AN 11/20 MACHINE.
IF RUNNING ON AN 11/40 OR AN 11/45 'HOLD:' MUST BE PATCHED TO 40

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

- 7.4 TO USE THE 'XOR' TESTER, THE BRANCH AROUND THE 'XOR'
CODE MUST BE PATCHED TO A 'NOP'. (SEE LISTINGS FOR DETAILS)

8. DEFAULT PARAMETERS:
1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160040
VECTOR ADDRESS- DURIV: 770
ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0
LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: C
DU PRIORITY LEVEL- LEVEL 5 DUPRT: LEVEL 5
OF SYNC CHARS SELECTED - 2 SYNCNO: 377
IS SEC XMIT JUMPER # 6 IN ?- YES SEXMIT: 377
IS SEC REC JUMPER # 5 IN ?- YES SEREC: 377
IS OPT CLR ENABLE JUMPER # 4 IN ?- YES OPTCLR: 377

DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
CONNECTOR ON (H315)- YES

JMRBY: 377

9. PROGRAM DESCRIPTION
- 9.1 THIS PROGRAM PERFORMS THE OFFLINE COMBINED (TRANSMITTER & RECEIVER)
CABLE TESTING OF THE DEVICE
SEE LISTING FOR DETAILS
10. FLOW CHARTS: RECEIVER FLOW, TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW
11. LISTINGS

675
676 000000' 000000G

D.


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

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



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

```

822                                     ;STANDARD INTERRUPT VECTORS
823
824
825                                     .=24
826 000024 010430                       .PFAIL                       ;POWER FAIL HANDLER
827 000026 000340                       340                          ;SERVICE AT LEVEL 7
828 000030 010160                       .HLT                          ;ERROR HANDLER
829 000032 000340                       340                          ;SERVICE AT LEVEL 7
830 000034 010126                       .TRPSRV                       ;GENERAL HANDLER DISPATCH SERVICE
831 000036 000340                       340                          ;SERVICE AT LEVEL 7
832
833                                     ;SOFTWARE SWITCH REGISTER
834
835                                     .=174
836 000174 000000                       DISPREG:.WORD 0              ;SOFTWARE DISPLAY REG.
837 000176 000000                       SWREG: .WORD 0              ;SOFTWARE SWITCH REGISTER
838 000200 000167 001214                 JMP .START                   ;GO TO START OF PROGRAM
839
840
841
842                                     .=1100
843
844                                     ;INDIRECT POINTERS
845
846 001100 177570                       SWR: 177570                  ;SWITCH REGISTER POINTER
847 001102 177570                       LIGHTS:177570               ;DISPLAY REGISTER POINTER
848 001104 177560                       TKCSR: 177560              ;TELETYPE KEYBOARD CONTROL REGISTER
849 001106 177562                       TKDBR: 177562              ;TELETYPE KEYBOARD DATA BUFFER
850 001110 177564                       TPCSR: 177564              ;TELEPRINTER CONTROL REGISTER
851 001112 177566                       TPDBR: 177566              ;TELEPRINTER DATA BUFFER
852
853                                     ;PROGRAM CONTROL PARAMETERS
854
855 001114 000000                       RTRN: 0                     ;SCOPE ADDRESS FOR LOOP ON TEST
856 001116 000000                       NEXT: 0                     ;ADDRESS OF NEXT TEST TO BE EXECUTED
857 001120 000000                       LOCK: 0                     ;ADDRESS FOR LOCK ON CURRENT DATA
858 001122 000000                       ICOUNT: 0                   ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
859 001124 000000                       LPCNT: 0                    ;NUMBER OF ITERATIONS COMPLETED
860 001126 000000                       TSTNO: 0                    ;NUMBER OF TEST IN PROGRESS
861 001130 000000                       PASCNT: 0                   ;NUMBER OF PASSES COMPLETED
862 001132 000000                       ERRCNT: 0                   ;TOTAL NUMBER OF ERRORS
863 001134 000000                       LSTERR: 0                   ;PC OF LAST ERROR CALL
864
865                                     ;PROGRAM VARIABLES
866
867 001136 000020                       HOLD: 20                    ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
868 001140 000000                       SHIFT: 0                    ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
869 001142 000000                       COUNT: 0                    ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
870 001144 000000                       TEMP1: 0                    ;TEMPORARY STORAGE
871 001146 000000                       TEMP2: 0                    ;TEMPORARY STORAGE
872 001150 000000                       TEMP3: 0                    ;TEMPORARY STORAGE
873 001152 000000                       TEMP4: 0                    ;TEMPORARY STORAGE
874 001154 000000                       TEMP5: 0                    ;TEMPORARY STORAGE
875 001156 000000                       SAVR0: 0                    ;R0 STORAGE
876 001160 000000                       SAVR1: 0                    ;R1 STORAGE
877 001162 000000                       SAVR2: 0                    ;R2 STORAGE

```


878 001164 000000
379 001166 000000
880 001170 000000
881 001172 000000
882 001174 000000

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

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

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


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

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

968	001366		.TRPTAB:	
969			*****	
970			*****	
971		104400	.SCOPE	SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
972	001366	006712	.SCOPE	
973		104401	.SCOPE1	SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
974	001370	007076	.SCOPE1	
975		104402	.TYPE	TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
976	001372	007116	.TYPE	
977		104403	.INSTR	INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
978	001374	007156	.INSTR	
979		104404	.INSTER	INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
980	001376	007274	.INSTER	
981		104405	.PARAM	PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
982	001400	007326	.PARAM	
983		104406	.SAV05	SAV05=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
984	001402	007542	.SAV05	
985		104407	.RES05	RES05=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
986	001404	007602	.RES05	
987		104410	.CONVRT	CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
988	001406	007634	.CONVRT	
989		104411	.CNVRT	CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
990	001410	007640	.CNVRT	
991		104412	.SETFLG	SETFLG=TRAP+12 ;CALL TO FLAG SET ROUTINE
992	001412	010060	.SETFLG	
993		104413	.CKSWR	CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
994	001414	010574	.CKSWR	

```
995          104414          CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
996 001416 010650          .CNTLU
997          :*****
998          :*****
999
1000          :PROGRAM INITIALIZATION
1001          :LOCK OUT INTERRUPTS
1002          :SET UP PROCESSOR STACK
1003          :SET UP POWER FAIL VECTOR
1004          :CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1005          :TYPE TITLE MESSAGE
1006
1007 001420 012767 000340 176350 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1008 001426 012706 001100          MOV #STACK,SP ;SET UP STACK
1009 001432 012737 010430 000024          MOV #,PFAIL,@#24 ;SET UP POWER FAIL VECTOR
1010 001440 005067 177460          CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
1011 001444 105067 177575          CLR STFLG ;CLEAR START FLAG
1012 001450 005067 177454          CLR PASCNT ;CLEAR PASS COUNT
1013 001454 105067 177566          CLR ERRFLG ;CLEAR ERROR FLAG
1014 001460 005067 177446          CLR ERRCNT ;CLEAR ERROR COUNT
1015 001464 005067 177444          CLR LSTERR ;CLEAR LAST ERROR POINTER
1016 001470 012767 000001 177430          MOV #1,TSTNO ;SET UP FOR TEST 1
1017 001476 012767 001420 177410          MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
1018          ;TESTING STARTS
1019 001504 105767 177534          TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1020 001510 001004          BNE ONCE
1021 001512 104402 010750          TYPE ,MTITLE ;TYPE TITLE MESSAGE
1022 001516 105167 177522          COMB INIFLG ;IF NOT SET FLAG AND DO
1023 001522 012767 177570 177350 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
1024 001530 012767 177570 177344          MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
1025 001536 013746 000006          MOV @#6,-(SP) ;SAVE VECTORS
1026 001542 013746 000004          MOV @#4,-(SP)
1027 001546 012737 001566 000004          MOV #64$,@#4 ;SET UP FOR TIMEOUT
1028 001554 022777 177777 177316          CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1029 001562 001402          BEQ 65$
1030 001564 000407          BR 66$
1031 001566 022626          64$: CMP (SP)+,(SP)+ ;ADJUST STACK
1032 001570 012767 000176 177302          65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1033 001576 012767 000174 177276          MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
1034 001604 012637 000004          66$: MOV (SP)+,@#4 ;RESTORE VECTORS
1035 001610 012637 000006          MOV (SP)+,@#6
1036 001614 005737 000042          TST @#42 ;UNDER MONITOR
1037 001620 001005          BNE MAP
1038 001622 022767 000176 177250          CMP #SWREG,SWR ;IS SWREG USED
1039 001630 001001          BNE MAP ; BRANCH TO CHECK FOR STATUS MAP
1040 001632 104414          CNTLU
1041          :*****
1042          ; CODE FOR STATUS MAP
1043          ; CODE ADDED FOR REV. E OF DIAGNOSTICS
1044          ; IF SW07= 1 ,THEN YOU USE THE STATUS MAP PREVIOUSLY
1045          ; SETUP, OR REENTER QUESTIONING ROUTINE
1046
1047 001634 032777 000200 177236 MAP: BIT #SW07,@SWR ; IS SW07=1?
1048 001642 001537          BEQ $67 ; IF NOT, GO TO TEST FOR SW00=1
1049          ; NOW SET UP MAP VALUES FOR PROGRAM
1050          ; THESE VALUES FROM THE STATUS MAP WILL BE USED IN THE
```



```

1051                                     ; OPERATION OF THIS PROGRAM.
1052 001644 116767 177430 177324      MOVB NOSYNC      ,SYNCRNO      ; SYNC CHAR
1053 001652 116767 177423 177317      MOVB MITSEX      ,SEXMIT      ; XMIT JUMPER
1054 001660 116767 177416 177312      MOVB RESEC       ,SEREC       ; SEC REC JUMPER
1055 001666 116767 177411 177305      MOVB CLROPT     ,OPTCLR     ; OPTIONAL JUMPER
1056 001674 116767 177404 177300      MOVB DMULT      ,MULTD      ; MULTIPLE DEVICE
1057 001702 116767 177377 177273      MOVB BYJMR      ,JMRBY      ; EXTERNAL MODEM
1058 001710 016767 177372 177266      MOV ADDBASE     ,BASEADD     ; PROG 1ST DEVICE ADDR
1059 001716 016767 177366 177262      MOV ADDKEEP     ,KEEPADD     ; SAVED 1ST DEVICE ADDR
1060 001724 016767 177362 177256      MOV ADDLAST     ,LASTADD     ; LAST DEVICE RXCSR ADDR
1061 001732 016767 177360 177254      MOV IVKEEP      ,KEEPIV     ; SAVED INTR VECTOR
1062 001740 016767 177354 177250      MOV REGACT      ,ACTREG     ; ACTIVE REGISTER
1063 001746 016767 177350 177244      MOV ADDROT      ,ROTADD     ; ROTATING POINTER
1064 001754 016767 177334 177230      MOV IVBASE      ,BASEIV     ; BASE INTR VECTOR
1065 001762 016767 177220 177214      MOV KEEPADD     ,BASEADD     ; RELOAD BASEADD
1066 001770 016767 177332 011116      MOV RIVDU      ,DURIV      ; REC INTR VECTOR
1067 001776 016767 177332 011112      MOV RISDU      ,DURIS      ; REC INTR STATUS
1068 002004 016767 177320 011106      MOV TIVDU      ,DUTIV      ; XMIT INTR VECTOR
1069 002012 016767 177314 011102      MOV TISDU      ,DUTIS      ; XMIT INTR STATUS
1070 002020 016767 177312 010410      MOV LISS       ,LESS1      ; PRIORITY TO ALLOW INTR
1071 002026 013737 001324 012434      MOV @#PRTDU    ,@#DUPRT    ; PRIORITY RELOADED
1072 002034 016767 177324 010530      MOV BASEDU     ,DUBASE     ;
1073 002042 016767 177272 011020      MOV CSRRX      ,RXCSR     ;
1074 002050 016767 177266 011014      MOV CSRHRX     ,HRXCSR    ;
1075 002056 016767 177262 011010      MOV BUFRXD     ,RXDBUF    ;
1076 002064 016767 177256 011004      MOV BUHRXD     ,HRXDBUF   ;
1077 002072 016767 177252 011000      MOV CSRPAR     ,PARCSR    ;
1078 002100 016767 177246 010774      MOV CSRHPAR    ,HPARCSR   ;
1079 002106 016767 177242 010770      MOV CSRTX      ,TXCSR     ;
1080 002114 016767 177236 010764      MOV CSRHTX     ,HTXCSR    ;
1081 002122 016767 177232 010760      MOV BUFTXD     ,TXDBUF    ;
1082 002130 016767 177226 010754      MOV BUFTXD     ,HTXDBUF   ;
1083 002136 000167 000466      JMP .BEGIN     ; BRANCH TO BEGIN TESTING
1084                                     ;*****
1085 002142 032777 000001 176730 $67: BIT #SW00 ,@SWR ; RESELECT VECTOR $ CONTROL REG?
1086 002150 001002      BNE 1$ ; BRANCH TO QUESTIONING
1087 002152 000167 000452      JMP .BEGIN ; GO TO LOAD STATUS MAP ETC.
1088 002156 005037 001222 1$: CLR @#FLAG ; CLEAR FLAG SO STATUS MAP PRINTS OUT
1089 002162 012700 000300      MOV #300,R0 ; RESTORE VECTOR AREA TO TRAPCATCHER
1090 002166 012701 000302      MOV #302,R1 ; START AT LOCATION 300
1091 002172 012702 000004      MOV #4,R2
1092 002176 010110 2$: MOV R1,(R0)
1093 002200 005011      CLR (R1)
1094 002202 060200      ADD R2,R0
1095 002204 060201      ADD R2,R1
1096 002206 022701 001000      CMP #1000,R1 ; END AT LOCATION 776
1097 002212 002771      BLT 2$
1098 002214 104403      INSTR ; OUTPUT MESSAGE & GET INPUT STRING
1099 002216 011024      MREGAD ; MESSAGE
1100 002220 104405      PARAM ; CONVERT STRING
1101 002222 160000      160000 ; LOW LIMIT
1102 002224 167776      167776 ; HIGH LIMIT
1103 002226 012572      DUBASE ; STORE AT THIS LOCATION
1104 002230 001 .BYTE 1 ; MASK
1105 002231 001 .BYTE 1 ; HOW MANY TIMES * 2
1106 002232 016767 010334 176746      MOV DUBASE,KEEPADD ; SAVE
    
```

```

1107 002240 004767 010174 JSR PC,DUADDR
1108 002244 016767 176736 176732 MOV KEEPADD,BASEADD ;RESTORE FOR ROTATION
1109 002252 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1110 002254 011002 MVECTO ;MESSAGE
1111 002256 104405 PARAM ;CONVERT STRING
1112 002260 000300 300 ;LOW LIMIT
1113 002262 000776 776 ;HIGH LIMIT
1114 002264 013114 DURIV ;STORE AT THIS LOCATION
1115 002266 001 .BYTE 1 ;MASK
1116 002267 004 .BYTE 4 ;HOW MANY TIMES + 2
1117 002270 016767 010620 176716 MOV DURIV,KEEPIV ;SAVE
1118 002276 016767 010612 176706 MOV DURIV,BASEIV ;SET UP FOR ROTATION
1119 002304 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1120 002306 011105 MMULT ;MESSAGE
1121 002310 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1122 002312 001202 MULTD ;THIS FLAG
1123 002314 105767 176662 TSTB MULTD ;ARE THERE MULTIPLE DEVICES
1124 ;ON THE SYSTEM ?
1125 002320 100406 BMI BBB ;YES,ASK NEXT QUESTION
1126 002322 005067 176670 CLR ACTREG
1127 002326 005067 176666 CLR ROTADD
1128 002332 000167 000140 JMP OUTMUL ;JUMP AROUND NEXT QUESTION
1129 002336 BBB:
1130 002336 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1131 002340 011164 MLASTD ;MESSAGE
1132 002342 104405 PARAM ;CONVERT STRING
1133 002344 160000 160000 ;LOW LIMIT
1134 002346 167776 167776 ;HIGH LIMIT
1135 002350 001210 LASTADD ;STORE AT THIS LOCATION
1136 002352 001 .BYTE 1 ;MASK
1137 002353 001 .BYTE 1 ;HOW MANY TIMES + 2
1138 ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1139 002354 012767 000001 176636 1$: MOV #1,ROTADD ;SET UP POINTER
1140 002362 005067 176630 CLR ACTREG ;CLR ACTIVE REGISTER
1141 002366 056767 176626 176622 2$: BIS ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1142 002374 000241 CLC
1143 002376 006167 176616 ROL ROTADD ;SET UP POINTER
1144 002402 103421 BCS 3$ ;ARE YOU OUT OF RANGE ?
1145 002404 062767 000010 176572 ADD #10,BASEADD ;SET UP BASE ADDRESS
1146 002412 026767 176572 176564 CMP LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1147 002420 101362 BHI 2$ ;NO DO IT AGAIN
1148 002422 056767 176572 176566 BIS ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1149 ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1150 ;MULTIPLE DEVICE QUESTION
1151 002430 012767 000001 176562 4$: MOV #1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1152 002436 016767 176544 176540 MOV KEEPADD,BASEADD ;DITTO
1153 002444 000414 BR OUTMUL ;CONTINUE QUESTIONS
1154 002446 016767 176534 176530 3$: MOV KEEPADD,BASEADD ;RESTORE
1155 002454 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1156 002456 011347 MRANGE ;MESSAGE
1157 002460 104405 PARAM ;CONVERT STRING
1158 002462 160000 160000 ;LOW LIMIT
1159 002464 167776 167776 ;HIGH LIMIT
1160 002466 001210 LASTADD ;STORE AT THIS LOCATION
1161 002470 001 .BYTE 1 ;MASK
1162 002471 001 .BYTE 1 ;HOW MANY TIMES + 2

```



```

1163 002472 000167 177656          JMP      1$      ;DO IT AGAIN
1164 002476                                OUTMUL:
1165 002476 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1166 002500 011633                    MLEVEL                               ;MESSAGE
1167 002502 104405                    PARAM                                ;CONVERT STRING
1168 002504 000004                    4                                    ;LOW LIMIT
1169 002506 000007                    7                                    ;HIGH LIMIT
1170 002510 012434                    DUPRT                                ;STORE AT THIS LOCATION
1171 002512      000                    .BYTE 0                               ;MASK
1172 002513      001                    .BYTE 1                               ;HOW MANY TIMES + 2
1173 002514 004767 007644          JSR      PC,DULEV
1174                                ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1175                                ;BUFFER TO THE CHARACTERS '1' AND '2'.
1176                                ;IF THE CHARACTER IS '1' CLEAR THE FLAG
1177                                ;IF THE CHARACTER IS '2' SET THE FLAG
1178 002520                                AAA:
1179 002520 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1180 002522 011660                    MSYNC                                ;MESSAGE
1181 002524 122767 000061 007472 3$:  CMPB    #'1',INBUF                ;IS IT '1' ?
1182 002532 001003                    BNE     1$
1183 002534 105067 176436                    CLRB   SYNCNO                      ;000
1184 002540 000412                    BR     4$
1185 002542 122767 000062 007454 1$:  CMPB    #'2',INBUF                ;IS IT '2' ?
1186 002550 001004                    BNE     2$
1187 002552 112767 177777 176416          MOVB   #-1,SYNCNO                  ;377
1188 002560 000402                    BR     4$
1189 002562 104404                    2$:  INSTER                          ;RETRY
1190 002564 000757                    BR     3$
1191 002566 000240                    4$:  NOP
1192 002570 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1193 002572 011726                    MWIRE6                               ;MESSAGE
1194 002574 104412                    SETFLG                              ;SET FLAG BASED UPON INPUT STRING
1195 002576 001177                    SEXMIT                              ;THIS FLAG
1196 002600 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1197 002602 011774                    MWIRE5                               ;MESSAGE
1198 002604 104412                    SETFLG                              ;SET FLAG BASED UPON INPUT STRING
1199 002606 001200                    SEREC                              ;THIS FLAG
1200 002610 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1201 002612 012041                    MWIRE4                               ;MESSAGE
1202 002614 104412                    SETFLG                              ;SET FLAG BASED UPON INPUT STRING
1203 002616 001201                    OPTCLR                              ;THIS FLAG
1204 002620 104403                    INSTR                                ;OUTPUT MESSAGE & GET INPUT STRING
1205 002622 012115                    MEXTJ                               ;MESSAGE
1206 002624 104412                    SETFLG                              ;SET FLAG BASED UPON INPUT STRING
1207 002626 001203                    JMRBY                              ;THIS FLAG
1208
1209
1210                                ;TEST START AND RESTART
1211
1212 002630 012767 000340 175140 .BEGIN: MOV    #340,PS                ;LOCK OUT INTERRUPTS
1213                                ; ***** LOAD STATUS MAP *****
1214                                ; THE VALUES NOW BEING LOADED INTO THE STATUS MAP WILL BE
1215                                ; USED IN THIS PROGRAM AND WILL BE PASSED TO ANY
1216                                ; OTHER DU11 PROGRAMS LOADED IMMEDIATELY FOLLOWING THIS PROG.
1217 002636 032777 000200 176234          BIT    #SW07                      ,@SWR ; SW07 SET , IF YES BRANCH
1218 002644 001132                    BNE    HEREU

```

1219	002646	116767	176324	176424	MOVB SYNCNO	,NOSYNC	:SYNC CHARS
1220	002654	116767	176317	176417	MOVB SEXMIT	,MITSEX	:XMIT JUMPER
1221	002662	116767	176312	176412	MOVB SEREC	,RESEC	:SEC REC JUMPER
1222	002670	116767	176305	176405	MOVB OPTCLR	,CLROPT	:OPTIONAL JUMPER
1223	002676	116767	176300	176400	MOVB MULTD	,DMULT	: MULTIPLE DEVICES
1224	002704	116767	176273	176373	MOVB JMRBY	,BYJMR	:EXTERNAL MODEM
1225	002712	016767	176266	176366	MOV BASEADD	,ADDBASE	:PROG CONTROLLED 1ST ADDR
1226	002720	016767	176262	176362	MOV KEEPADD	,ADDKEEP	:SAVED 1ST DEVICE ADDR
1227	002726	016767	176256	176356	MOV LASTADD	,ADDLAST	:LAST DEVICE RXCSR ADDR
1228	002734	016767	176254	176354	MOV KEEPIV	,IVKEEP	:SAVED INTR VECTOR
1229	002742	016767	176244	176344	MOV BASEIV	,IVBASE	: RELOAD BASE INTR VECTOR
1230	002750	016767	176242	176342	MOV ACTREG	,REGACT	:ACTIVE REGISTER
1231	002756	016767	176236	176336	MOV ROTADD	,ADDROT	:ROTATING POINTER
1232	002764	013737	012434	001324	MOV @#DUPRT	,@#PRTDU	:DU11 PRIORITY
1233	002772	016767	010116	176326	MOV DURIV	,RIVDU	:REC INTR VECTOR
1234	003000	016767	010112	176326	MOV DURIS	,RISDU	: REC INTR STATUS
1235	003006	016767	010106	176314	MOV DUTIV	,TIVDU	: XMIT INTR VECTOR
1236	003014	016767	010102	176310	MOV DUTIS	,TISDU	: XMIT INTR STATUS
1237	003022	016767	007410	176306	MOV LESS1	,L1ESS	: PRIORITY TO ALLOW INTR
1238	003030	016767	007536	176326	MOV DUBASE	,BASEDU	:RXCSR BASE ADDRESS
1239	003036	016767	010026	176274	MOV RXCSR,	CSRRX	
1240	003044	016767	010022	176270	MOV HRXCSR,	CSRHRX	
1241	003052	016767	010016	176264	MOV RXDBUF,	BUFRXD	
1242	003060	016767	010012	176260	MOV HRXDBUF,	BUFHRXD	
1243	003066	016767	010006	176254	MOV PARCSR,	CSRPAR	
1244	003074	016767	010002	176250	MOV HPARCSR,	CSRHPAR	
1245	003102	016767	007776	176244	MOV TXCSR,	CSRTX	
1246	003110	016767	007772	176240	MOV HTXCSR,	CSRHTX	
1247	003116	016767	007766	176234	MOV TXDBUF,	BUFTXD	
1248	003124	016767	007762	176230	MOV HTXDBUF,	BUFHTXD	
1249					:*****		
1250					: THE FOLLOWING CODE WILL PRINT		
1251					: THE CONVERSATIONALLY SET JUMPER		
1252					: SETTINGS FROM THE STATUS MAP		
1253					: ON THE FIRST PASS OF		
1254					: THIS DIAGNOSTIC OR		
1255					: JUST AFTER THE QUESTIONING		
1256					: AND ANSWERING .		
1257					:*****		
1258							
1259	003132	005737	001222		HEREU:	TST @#FLAG	: TEST IF 1ST PASS
1260	003136	001402				BEQ SETFG	: IF FIRST PASS SET FLAG/PRINT
1261	003140	000167	000116			JMP THRU	: AROUND IF PASS > 1
1262	003144	104402	012172		SETFG:	TYPE MSTATUS	: PRINT 'STATUS MAP'
1263	003150	062737	000001	001222		ADD #1, @#FLAG	: SET FLAG ON 1ST PASS
1264	003156	010067	176042			MOV R0, HOLD0	: SAVE R0
1265	003162	010167	176040			MOV R1, HOLD1	: SAVE R1
1266	003166	012767	000003	176034		MOV #3, COUNT1	:COUNTER FOR WORDS PRINTED
1267	003174	012700	000002			MOV #BUFF1, R0	
1268	003200	012701	001300			MOV #STATUS, R1	: (BUFF1)=STATUS ETC.
1269	003204	010120			FILBUF:	MOV R1, (R0)+	: LOAD BUFF AS ABOVE
1270	003206	062701	000002			ADD #2, R1	: PREPARE STATUS ADDRESS
1271	003212	020127	001306			CMP R1, #STATUS+6	: CHECK IF 3 WORDS LOADED
1272	003216	001372				BNE FILBUF	: BACK TO LOAD NEXT ADDRESS
1273	003220	012700	000002			MOV #BUFF1, R0	: LOAD FOR PRINT OUT
1274	003224	010067	176006		UP:	MOV R0, TABLE+4	:LOAD ADDRESS TO PRINT

1275	003230	012067	176006			MOV (R0)+, TABLE+10		: LOAD CONTENTS
1276	003234	104410	001232			CONVRT, TABLE		: PRINT ADDRESS/CONTENTS PAIR
1277	003240	104402	011434			TYPE, MCRLF		: CR AND LF
1278	003244	005367	175760			DEC COUNT1		: COUNT WORDS PRINTED
1279	003250	001365				BNE UP		: GO PRINT NEXT ADDRESS/CONTENTS
1280	003252	016700	175746			MOV HOLD0, R0		
1281	003256	016701	175744			MOV HOLD1, R1		
1282								
1283								
1284	003262	012706	001100		THRU:	MOV #STACK, SP		: SET UP STACK
1285	003266	005737	000042			TST @#42		: IS PROGRAM UNDER MONITOR CONTROL
1286	003272	001056				BNE 3\$		
1287	003274	105767	175702			TSTB MULTD		: DON'T ALLOW LOCK ON TEST IF RUNNING
1288								: MULTIPLE DEVICES
1289	003300	001407				BEQ 5\$: IF NO, TEST FOR LOCK ON TEST
1290	003302	016767	003564	003464		MOV BRW, TTST		: RESTORE NORMAL SCOPE LOOP
1291	003310	016767	003560	003460		MOV BRX, TTST+2		: DITTO
1292	003316	000444				BR 3\$: JUMP AROUND IF YES
1293	003320	032777	000004	175552	5\$:	BIT #BIT2, @SWR		: CHECK FOR LOCK ON TEST
1294	003326	001416				BEQ 1\$		
1295	003330	104403				INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1296	003332	011570				MLOCK		: MESSAGE
1297	003334	104412				SETFLG		: SET FLAG BASED UPON INPUT STRING
1298	003336	001247				LOKFLG		: THIS FLAG
1299	003340	105767	175703			TSTB LOKFLG		: IS LOCK ON TEST OPTION SELECTED
1300	003344	001407				BEQ 1\$		
1301	003346	012767	000240	003420		MOV #NOP, TTST		
1302	003354	012767	000240	003414		MOV #NOP, TTST+2		: SET UP TO LOCK
1303	003362	000406				BR 2\$		
1304	003364	016767	003502	003402	1\$:	MOV BRW, TTST		
1305	003372	016767	003476	003376		MOV BRX, TTST+2		: LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1306	003400	032777	000002	175472	2\$:	BIT #SW01, @SWR		: IF SW01=1, GET STARTING PC
1307	003406	001410				BEQ 3\$		
1308	003410	104403				INSTR		: OUTPUT MESSAGE & GET INPUT STRING
1309	003412	011555				MTSTPC		: MESSAGE
1310	003414	104405				PARAM		: CONVERT STRING
1311	003416	003446				TST1		: LOW LIMIT
1312	003420	006102				TLAST		: HIGH LIMIT
1313	003422	001114				RTRN		: STORE AT THIS LOCATION
1314	003424	001			.BYTE	1		: MASK
1315	003425	001			.BYTE	1		: HOW MANY TIMES + 2
1316	003426	000403				BR 4\$		
1317	003430	012767	003446	175456	3\$:	MOV #TST1, RTRN		: START AT TEST 1
1318	003436	104402	011551		4\$:	TYPE, MR		: TYPE R
1319	003442	000177	175446			JMP @RTRN		: START TESTING
1320								

```

1321                                     ::THIS TEST VERIFYS MATCH DETECT & DATA RDY
1322                                     ::FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1323                                     ::BY OBSERVING RECACT BIT
1324                                     ::IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1325                                     ::* DEPENDENT ON MONITOR .....
1326                                     ::IF ONE SYNC STRAP IS SELECTED ,IT WILL
1327                                     ::ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1328                                     ::ASSERT
1329                                     ::MODE: SYNC INTERNAL
1330                                     ::LENGTH: FIVE
1331                                     ::SYNC CHARACTER FOR MATCH: B/C
1332                                     ::THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1333
1334 003446 012767 000001 175452 TST1:  MOV    #1,TSTNO           ;SAVE THIS
1335 003454 012767 003774 175434      MOV    #TST2,NEXT         ;GO TO THIS TEST WHEN THRU
1336 003462 012767 003576 175430      MOV    #3$,LOCK          ;SET UP FOR SCOPE LOOP
1337 003470 052777 000400 007406      BIS    #MRESET,@TXCSR    ;MASTER RESET
1338 003476 016703 007372              MOV    RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
1339                                     ;SET SYNC INTERNAL,FIVE,NO PARITY,0 SYNC REGISTER
1340 003502 012704 030000              MOV    #SYNINT!FIVE!NOPAR,R4 ;CREATE PARAMETERS
1341 003506 012777 004020 007370 1$:  MOV    #MINT!SEND,@TXCSR  ;SET SEND & MAINT INTER
1342 003514 010477 007360              MOV    R4,@PARCSR        ;LOAD CSR
1343 003520 052777 000020 007342      BIS    #SYNSCH,@RXCSR    ;SET SYNC SEARCH
1344                                     ;POKE CLK TO GET INTO SYNCHRONIZATION
1345                                     ;BOTH THE LOGIC & RECEIVER
1346 003526 052777 020000 007350      BIS    #CLK,@TXCSR       ;POKE CLK UP
1347 003534 042777 020000 007342      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1348 003542 110477 007342              MOV    R4,@TXDBUF        ;LOAD DATA CHARACTER
1349                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1350 003546 052777 020000 007330      BIS    #CLK,@TXCSR       ;POKE CLK UP
1351 003554 042777 020000 007322      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1352 003562 032777 004000 007300      BIT    #RECACT,@RXCSR    ;RECACT ?
1353 003570 001401                    BEQ    2$
1354 003572 104000                    HLT
1355 003574                    2$:  HLT    ;RECACT SHOULD NOT BE SET
1356 003574 000404                    BR     4$
1357 003576 010477 007276 3$:  MOV    R4,@PARCSR        ;LOAD PARCSR WITH PARAMETERS
1358 003602 110477 007302              MOV    R4,@TXDBUF        ;LOAD SYNC CHAR
1359 003606 012767 000002 175326 4$:  MOV    #2,COUNT          ;# OF SYNC CHARS
1360 003614 005777 007264 5$:  TST    @TXCSR            ;DNA ?
1361 003620 100001                    BPL    6$                ;BR IF NOT SET
1362 003622 104000                    HLT    ;DNA SHOULD NOT BE SET OR....
1363 003624                    6$:
1364                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1365 003624 012767 000005 175306      MOV    #5,SHIFT          ;# OF SHIFTS
1366 003632                    7$:
1367 003632 052777 020000 007244      BIS    #CLK,@TXCSR       ;POKE CLK UP
1368 003640 042777 020000 007236      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1369 003646 005367 175266              DEC    SHIFT            ;# OF SHIFTS
1370 003652 001367                    BNE    7$
1371 003654 005367 175262              DEC    COUNT            ;# OF SYNC CHARS
1372 003660 001403                    BEQ    8$
1373                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1374 003662 105767 175310              TSTB   SYNCNO
1375 003666 100752                    BMI    5$                ;TWO SYNC CHARACTERS..
1376 003670 032777 004000 007172 8$:  BIT    #RECACT,@RXCSR    ;RECACT ?

```



```
1377 003676 001001          BNE      9$
1378 003700 104000          HLT
1379 003702          9$:
1380          ;RECACT FAILED TO SET,POSSIBLE
1381          ;THAT THE RECEIVER FAILED TO MATCH
1382 003702 017701 007166    MOV      @RXDBUF,R1      ;SAVE ACTUAL
1383 003706 010400          MOV      R4,R0          ;SAVE EXPECTED
1384 003710 042700 177400    BIC      #177400,R0     ;CLR UPPER BYTE
1385 003714 020001          CMP      R0,R1          ;DO THEY COMPARE ?
1386 003716 001401          BEQ     10$
1387 003720 104002          HLT      2              ;IF RECACT FAILED ALONG WITH THIS
1388 003722          10$:
1389          ;...IT PROBABLY IS A TRANSMITTER ERROR
1390          ;HOWEVER,...IF ONLY THIS FAILED IT
1391          ;PROBABLY IS A RECEIVER ERROR
1392 003722 104401          SCOPE1
1393          ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1394          ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1395          ;TXDBUF
1396 003724 052777 020000 007152  BIS      #CLK,@TXCSR    ;POKE CLK UP
1397 003732 005777 007146          TST      @TXCSR        ;DNA?
1398 003736 100401          BMI     11$
1399 003740 104000          HLT      ;DNA DID NOT ASSERT
1400 003742          11$:
1401          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1402 003742 052777 000400 007134  BIS      #MRESET,@TXCSR ;MASTER RESET
1403 003750 032777 000020 007112  BIT      #SYNSCH,@RXCSR ;SYNC SEARCH = 0 ?
1404 003756 001401          BEQ     12$
1405 003760 104000          HLT      ;SYNC SEARCH SHOULD BE NOT SET
1406 003762          12$:
1407 003762 005204          INC      R4
1408 003764 122704 000040          CMPB    #40,R4        ;IS THIS THE LAST CHARACTER ?
1409 003770 001246          BNE     1$            ;NO
1410 003772 104400          SCOPE
```

```

1411                                     ::THIS TEST VERIFYS MATCH DETECT & DATA RDY
1412                                     ::FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1413                                     ::BY OBSERVING RECACT BIT
1414                                     ::IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1415                                     ::* DEPENDENT ON MONITOR .....
1416                                     ::IF ONE SYNC STRAP IS SELECTED ,IT WILL
1417                                     ::ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1418                                     ::ASSERT
1419                                     ::MODE: SYNC INTERNAL
1420                                     ::LENGTH: SIX
1421                                     ::SYNC CHARACTER FOR MATCH: B/C
1422                                     ::THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1423
1424 003774 012767 000002 175124 TST2: MOV #2,TSTNO ;SAVE THIS
1425 004002 012767 004322 175106 MOV #TST3,NEXT ;GO TO THIS TEST WHEN THRU
1426 004010 012767 004124 175102 MOV #3$,LOCK ;SET UP FOR SCOPE LOOP
1427 004016 052777 000400 007060 BIS #MRESET,@TXCSR ;MASTER RESET
1428 004024 016703 007044 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1429 ;SET SYNC INTERNAL,SIX,NO PARITY,0 SYNC REGISTER
1430 004030 012704 032000 MOV #SYNINT!SIX!NOPAR,R4 ;CREATE PARAMETERS
1431 004034 012777 004020 007042 1$: MOV #MINT!SEND,@TXCSR ;SET SEND & MAINT INTER
1432 004042 010477 007032 MOV R4,@PARCSR ;LOAD CSR
1433 004046 052777 000020 007014 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1434 ;POKE CLK TO GET INTO SYNCRONIZATION
1435 ;BOTH THE LOGIC & RECEIVER
1436 004054 052777 020000 007022 BIS #CLK,@TXCSR ;POKE CLK UP
1437 004062 042777 020000 007014 BIC #CLK,@TXCSR ;POKE CLK DOWN
1438 004070 110477 007014 MOV R4,@TXDBUF ;LOAD DATA CHARACTER
1439 ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
1440 004074 052777 020000 007002 BIS #CLK,@TXCSR ;POKE CLK UP
1441 004102 042777 020000 006774 BIC #CLK,@TXCSR ;POKE CLK DOWN
1442 004110 032777 004000 006752 BIT #RECACT,@RXCSR ;RECACT ?
1443 004116 001401 BEQ 2$
1444 004120 104000 HLT ;RECACT SHOULD NOT BE SET
1445 004122 2$:
1446 004122 000404 BR 4$
1447 004124 010477 006750 3$: MOV R4,@PARCSR ;LOAD PARCSR WITH PARAMETERS
1448 004130 110477 006754 MOV R4,@TXDBUF ;LOAD SYNC CHAR
1449 004134 012767 000002 175000 4$: MOV #2,COUNT ;# OF SYNC CHARS
1450 004142 005777 006736 5$: TST @TXCSR ;DNA ?
1451 004146 100001 BPL 6$ ;BR IF NOT SET
1452 004150 104000 HLT ;DNA SHOULD NOT BE SET OR....
1453 004152 6$:
1454 ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1455 004152 012767 000006 174760 7$: MOV #6,SHIFT ;# OF SHIFTS
1456 004160 BIS #CLK,@TXCSR ;POKE CLK UP
1457 004160 052777 020000 006716 BIC #CLK,@TXCSR ;POKE CLK DOWN
1458 004166 042777 020000 006710 DEC SHIFT ;# OF SHIFTS
1459 004174 005367 174740 BNE 7$
1460 004200 001367 DEC COUNT ;# OF SYNC CHARS
1461 004202 005367 174734 BEQ 8$
1462 004206 001403 ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1463 TSTB SYNCNO
1464 004210 105767 174762 BMI 5$ ;TWO SYNC CHARACTERS..
1465 004214 100752 8$: BIT #RECACT,@RXCSR ;RECACT ?
1466 004216 032777 004000 006644

```



```

1467 004224 001001          BNE      9$
1468 004226 104000          HLT
1469 004230          9$:
1470
1471
1472 004230 017701 006640      MOV     @RXDBUF,R1      ;SAVE ACTUAL
1473 004234 010400      MOV     R4,R0          ;SAVE EXPECTED
1474 004236 042700 177400      BIC     #177400,R0     ;CLR UPPER BYTE
1475 004242 020001      CMP     R0,R1          ;DO THEY COMPARE ?
1476 004244 001401      BEQ    10$
1477 004246 104002      HLT
1478 004250          10$:
1479
1480
1481
1482 004250 104401          SCOPE1
1483
1484
1485
1486 004252 052777 020000 006624  ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1487 004260 005777 006620          ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1488 004264 100401          ;TXDBUF
1489 004266 104000          BIS     #CLK,@TXCSR   ;POKE CLK UP
1490 004270          TST     @TXCSR        ;DNA?
1491
1492 004270 052777 000400 006606  BMI    11$
1493 004276 032777 000020 006564  HLT     ;DNA DID NOT ASSERT
1494 004304 001401          11$:
1495 004306 104000          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1496 004310          BIS     #MRESET,@TXCSR ;MASTER RESET
1497 004310 005204          BIT     #SYNSCH,@RXCSR ;SYNC SEARCH = 0 ?
1498 004312 122704 000100      BEQ    12$
1499 004316 001246          HLT     ;SYNC SEARCH SHOULD BE NOT SET
1500 004320 104400          12$:
          INC     R4
          CMPB   #100,R4 ;IS THIS THE LAST CHARACTER ?
          BNE    1$      ;NO
          SCOPE
  
```

```

1501                                     ::THIS TEST VERIFYS MATCH DETECT & DATA RDY
1502                                     ::FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1503                                     ::BY OBSERVING RECACT BIT
1504                                     ::IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1505                                     ::*   DEPENDENT ON MONITOR .....
1506                                     ::IF ONE SYNC STRAP IS SELECTED ,IT WILL
1507                                     ::ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1508                                     ::ASSERT
1509                                     ::MODE: SYNC INTERNAL
1510                                     ::LENGTH: SEVEN
1511                                     ::SYNC CHARACTER FOR MATCH: B/C
1512                                     ::THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1513
1514 004322 012767 000003 174576 TST3:  MOV    #3,TSTNO           ;SAVE THIS
1515 004330 012767 004650 174560      MOV    #TST4,NEXT         ;GO TO THIS TEST WHEN THRU
1516 004336 012767 004452 174554      MOV    #3$,LOCK          ;SET UP FOR SCOPE LOOP
1517 004344 052777 000400 006532      BIS    #MRESET,@TXCSR    ;MASTER RESET
1518 004352 016703 006516              MOV    RXDBUF,R3         ;SET UP FOR ERROR MESSAGE
1519                                     ;SET SYNC INTERNAL,SEVEN,NO PARITY,0 SYNC REGISTER
1520 004356 012704 034000              MOV    #SYNINT!SEVEN!NOPAR,R4 ;CREATE PARAMETERS
1521 004362 012777 004020 006514 1$:   MOV    #MINT!SEND,@TXCSR    ;SET SEND & MAINT INTER
1522 004370 010477 006504              MOV    R4,@PARCSR        ;LOAD CSR
1523 004374 052777 000020 006466      BIS    #SYNSCH,@RXCSR    ;SET SYNC SEARCH
1524                                     ;POKE CLK TO GET INTO SYNCHRONIZATION
1525                                     ;BOTH THE LOGIC & RECEIVER
1526 004402 052777 020000 006474      BIS    #CLK,@TXCSR       ;POKE CLK UP
1527 004410 042777 020000 006466      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1528 004416 110477 006466              MOVVB  R4,@TXDBUF        ;LOAD DATA CHARACTER
1529                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1530 004422 052777 020000 006454      BIS    #CLK,@TXCSR       ;POKE CLK UP
1531 004430 042777 020000 006446      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1532 004436 032777 004000 006424      BIT    #RECACT,@RXCSR    ;RECACT ?
1533 004444 001401                      BEQ    2$
1534 004446 104000                      HLT    ;RECACT SHOULD NOT BE SET
1535 004450                              2$:
1536 004450 000404                      BR     4$
1537 004452 010477 006422 3$:      MOV    R4,@PARCSR        ;LOAD PARCSR WITH PARAMETERS
1538 004456 110477 006426              MOVVB  R4,@TXDBUF        ;LOAD SYNC CHAR
1539 004462 012767 000002 174452 4$:   MOV    #2,COUNT          ;# OF SYNC CHARS
1540 004470 005777 006410 5$:      TST    @TXCSR            ;DNA ?
1541 004474 100001                      BPL    6$                ;BR IF NOT SET
1542 004476 104000                      HLT    ;DNA SHOULD NOT BE SET OR....
1543 004500                              6$:
1544                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1545 004500 012767 000007 174432      MOV    #7,SHIFT          ;# OF SHIFTS
1546 004506                              7$:
1547 004506 052777 020000 006370      BIS    #CLK,@TXCSR       ;POKE CLK UP
1548 004514 042777 020000 006362      BIC    #CLK,@TXCSR       ;POKE CLK DOWN
1549 004522 005367 174412              DEC    SHIFT            ;# OF SHIFTS
1550 004526 001367                      BNE    7$
1551 004530 005367 174406              DEC    COUNT            ;# OF SYNC CHARS
1552 004534 001403                      BEQ    8$
1553                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1554 004536 105767 174434              TSTB   SYNCNO
1555 004542 100752                      BMI    5$                ;TWO SYNC CHARACTERS..
1556 004544 032777 004000 006316 8$:   BIT    #RECACT,@RXCSR    ;RECACT ?

```



```
1557 004552 001001      BNE      9$
1558 004554 104000      HLT
1559 004556              9$:
1560                      ;REACT FAILED TO SET,POSSIBLE
1561                      ;THAT THE RECEIVER FAILED TO MATCH
1562 004556 017701 006312  MOV      @RXDBUF,R1      ;SAVE ACTUAL
1563 004562 010400      MOV      R4,R0          ;SAVE EXPECTED
1564 004564 042700 177400  BIC      #177400,R0     ;CLR UPPER BYTE
1565 004570 020001      CMP      R0,R1          ;DO THEY COMPARE ?
1566 004572 001401      BEQ     10$
1567 004574 104002      HLT
1568 004576              10$:
1569                      ;IF REACT FAILED ALONG WITH THIS
1570                      ;...IT PROBABLY IS A TRANSMITTER ERROR
1571                      ;HOWEVER,...IF ONLY THIS FAILED IT
1572 004576 104401      SCOPE1
1573                      ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1574                      ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1575                      ;TXDBUF
1576 004600 052777 020000 006276  BIS      #CLK,@TXCSR    ;POKE CLK UP
1577 004606 005777 006272      TST      @TXCSR        ;DNA?
1578 004612 100401      BMI     11$
1579 004614 104000      HLT
1580 004616              11$:
1581                      ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1582 004616 052777 000400 006260  BIS      #MRESET,@TXCSR ;MASTER RESET
1583 004624 032777 000020 006236  BIT      #SYNSCH,@RXCSR ;SYNC SEARCH = 0 ?
1584 004632 001401      BEQ     12$
1585 004634 104000      HLT
1586 004636              12$:
1587 004636 005204      INC      R4
1588 004640 122704 000200  CMPB    #200,R4        ;IS THIS THE LAST CHARACTER ?
1589 004644 001246      BNE     1$            ;NO
1590 004646 104400      SCOPE
```

```

1591                                     ::THIS TEST VERIFYS MATCH DETECT & DATA RDY
1592                                     ::FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1593                                     ::BY OBSERVING RECACT BIT
1594                                     ::IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1595                                     ::* DEPENDENT ON MONITOR .....
1596                                     ::IF ONE SYNC STRAP IS SELECTED ,IT WILL
1597                                     ::ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1598                                     ::ASSERT
1599                                     ::MODE: SYNC INTERNAL
1600                                     ::LENGTH: EIGHT
1601                                     ::SYNC CHARACTER FOR MATCH: B/C
1602                                     ::THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1603
1604 004650 012767 000004 174250 TST4:  MOV    #4,TSTNO          ;SAVE THIS
1605 004656 012767 005176 174232      MOV    #TST5,NEXT        ;GO TO THIS TEST WHEN THRU
1606 004664 012767 005000 174226      MOV    #3$,LOCK         ;SET UP FOR SCOPE LOOP
1607 004672 052777 000400 006204      BIS    #MRESET,@TXCSR   ;MASTER RESET
1608 004700 016703 006170              MOV    RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1609                                     ;SET SYNC INTERNAL,EIGHT,NO PARITY,0 SYNC REGISTER
1610 004704 012704 036000              MOV    #SYNINT!EIGHT!NOPAR,R4 ;CREATE PARAMETERS
1611 004710 012777 004020 006166 1$:  MOV    #MINT!SEND,@TXCSR ;SET SEND & MAINT INTER
1612 004716 010477 006156              MOV    R4,@PARCSR       ;LOAD CSR
1613 004722 052777 000020 006140      BIS    #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1614                                     ;POKE CLK TO GET INTO SYNCHRONIZATION
1615                                     ;BOTH THE LOGIC & RECEIVER
1616 004730 052777 020000 006146      BIS    #CLK,@TXCSR      ;POKE CLK UP
1617 004736 042777 020000 006140      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1618 004744 110477 006140              MOV    R4,@TXDBUF       ;LOAD DATA CHARACTER
1619                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1620 004750 052777 020000 006126      BIS    #CLK,@TXCSR      ;POKE CLK UP
1621 004756 042777 020000 006120      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1622 004764 032777 004000 006076      BIT    #RECACT,@RXCSR   ;RECACT ?
1623 004772 001401                    BEQ    2$
1624 004774 104000                    HLT
1625 004776                            2$:
1626 004776 000404                    BR     4$
1627 005000 010477 006074 3$:  MOV    R4,@PARCSR        ;LOAD PARCSR WITH PARAMETERS
1628 005004 110477 006100              MOV    R4,@TXDBUF       ;LOAD SYNC CHAR
1629 005010 012767 000002 174124 4$:  MOV    #2,COUNT         ;# OF SYNC CHARS
1630 005016 005777 006062 5$:  TST    @TXCSR           ;DNA ?
1631 005022 100001                    BPL    6$               ;BR IF NOT SET
1632 005024 104000                    HLT
1633 005026                            6$:
1634                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1635 005026 012767 000010 174104 7$:  MOV    #8,,SHIFT        ;# OF SHIFTS
1636 005034
1637 005034 052777 020000 006042      BIS    #CLK,@TXCSR      ;POKE CLK UP
1638 005042 042777 020000 006034      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1639 005050 005367 174064              DEC    SHIFT           ;# OF SHIFTS
1640 005054 001367                    BNE    7$
1641 005056 005367 174060              DEC    COUNT           ;# OF SYNC CHARS
1642 005062 001403                    BEQ    8$
1643                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1644 005064 105767 174106              TSTB   SYNCNO
1645 005070 100752                    BMI    5$               ;TWO SYNC CHARACTERS..
1646 005072 032777 004000 005770 8$:  BIT    #RECACT,@RXCSR   ;RECACT ?

```



```

1647 005100 001001      BNE      9$
1648 005102 104000      HLT
1649 005104              9$:
1650                      ;RECACT FAILED TO SET,POSSIBLE
1651                      ;THAT THE RECEIVER FAILED TO MATCH
1652 005104 017701 005764  MOV      @RXDBUF,R1      ;SAVE ACTUAL
1653 005110 010400      MOV      R4,R0          ;SAVE EXPECTED
1654 005112 042700 177400  BIC      #177400,R0     ;CLR UPPER BYTE
1655 005116 020001      CMP      R0,R1          ;DO THEY COMPARE ?
1656 005120 001401      BEQ      10$
1657 005122 104002      HLT      2              ;IF RECACT FAILED ALONG WITH THIS
1658 005124              10$:
1659                      ;...IT PROBABLY IS A TRANSMITTER ERROR
1660                      ;HOWEVER,...IF ONLY THIS FAILED IT
1661                      ;PROBABLY IS A RECEIVER ERROR
1662 005124 104401      SCOPE1
1663                      ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1664                      ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1665                      ;TXDBUF
1666 005126 052777 020000 005750  BIS      #CLK,@TXCSR    ;POKE CLK UP
1667 005134 005777 005744      TST      @TXCSR        ;DNA?
1668 005140 100401      BMI      11$
1669 005142 104000      HLT      ;DNA DID NOT ASSERT
1670 005144              11$:
1671                      ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1672 005144 052777 000400 005732  BIS      #MRESET,@TXCSR ;MASTER RESET
1673 005152 032777 000020 005710  BIT      #SYNSCH,@RXCSR ;SYNC SEARCH = 0 ?
1674 005160 001401      BEQ      12$
1675 005162 104000      HLT      ;SYNC SEARCH SHOULD BE NOT SET
1676 005164              12$:
1677 005164 005204      INC      R4
1678 005166 122704 000000      CMPB    #0,R4          ;IS THIS THE LAST CHARACTER ?
1679 005172 001246      BNE      1$            ;NO
1680 005174 104400      SCOPE

```

```

1681                                     :: THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1682                                     :: BOTH THE TRANSMITTER AND RECEIVER LOGIC
1683                                     :: MODE: SYNC EXTERNAL (SYNEXT)
1684                                     :: LENGTH: EIGHT PLUS PARITY
1685                                     :: PARITY: EVEPAR
1686                                     :: MAINT. MODE: MINT
1687                                     ::
1688 005176 012767 000005 173722 TST5: MOV #5,TSTNO ;SAVE THIS
1689 005204 012767 005410 173704 MOV #TST6,NEXT ;GO TO THIS TEST WHEN THRU
1690 005212 052777 000400 005664 BIS #MRESET,@TXCSR ;MASTER RESET
1691 005220 012777 020000 005652 MOV #SYNEXT,@PARCSR ;SET THE MODE
1692 005226 052777 000400 005650 BIS #MRESET,@TXCSR ;MASTER RESET
1693
1694 ;SET MAINTENANCE MODE & SEND
1695 ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1696 005234 012777 004020 005642 MOV #MINT!SEND,@TXCSR
1697
1698 ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1699 005242 012777 027426 005630 MOV #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
1700 005250 016703 005620 MOV RXDBUF,R3 ;SETUP FOR ERROR MSG
1701 005254 005004 CLR R4 ;FOR DATA CHAR CREATION
1702 005256 110477 005626 MOV#B R4,@TXDBUF ;LOAD CHARACTER
1703 005262 052777 000020 005600 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1704 ;GET INTO SYNCRONIZATION
1705 005270 052777 020000 005606 BIS #CLK,@TXCSR ;POKE CLK UP
1706 005276 042777 020000 005600 BIC #CLK,@TXCSR ;POKE CLK DOWN
1707 005304 012767 000011 173626 1$: MOV #9,SHIFT ;# OF SHIFTS
1708 005312 010400 MOV R4,R0 ;EXPECTED
1709 005314 2$:
1710 005314 052777 020000 005562 BIS #CLK,@TXCSR ;POKE CLK UP
1711 005322 042777 020000 005554 BIC #CLK,@TXCSR ;POKE CLK DOWN
1712 005330 005367 173604 DEC SHIFT ;# OF SHIFTS
1713 005334 022767 000003 173576 CMP #3,SHIFT ;TIME TO LOAD NEXT CHAR ?
1714 005342 001003 BNE 3$ ;NO ?
1715 005344 005204 INC R4 ;GENERATE NEXT CHAR
1716 005346 110477 005536 MOV#B R4,@TXDBUF ;LOAD NEXT CHARACTER
1717 005352 005767 173562 3$: TST SHIFT ;IS IT 0 ?
1718 005356 001356 BNE 2$
1719 005360 105777 005504 TSTB @RXCSR ;RXDONE = 1 ?
1720 005364 100401 BMI 5$
1721 005366 104000 HLT ;RXDONE SHOULD BE SET
1722 005370 5$:
1723 005370 017701 005500 MOV @RXDBUF,R1 ;ACTUAL
1724 005374 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1725 005376 001401 BEQ 6$
1726 005400 104002 HLT 2 ;CHARACTERS SHOULD COMPARE
1727 005402 6$:
1728 005402 105704 TSTB R4 ;LAST CHARACTER ?
1729 005404 001337 BNE 1$ ;NO
1730 005406 104400 4$: SCOPE

```



```

1731                                     ::THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1732                                     ::BOTH THE TRANSMITTER AND RECEIVER LOGIC
1733                                     ::MODE:SYNC EXTERNAL (SYNEXT)
1734                                     ::LENGTH:EIGHT PLUS PARITY
1735                                     ::PARITY:ODDPAR
1736                                     ::MAINT. MODE:MINT
1737                                     ::
1738 005410 012767 000006 173510 TST6:  MOV    #6,TSTNO      ;SAVE THIS
1739 005416 012767 005622 173472      MOV    #TST7,NEXT    ;GO TO THIS TEST WHEN THRU
1740 005424 052777 000400 005452      BIS    #MRESET,@TXCSR ;MASTER RESET
1741 005432 012777 020000 005440      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1742 005440 052777 000400 005436      BIS    #MRESET,@TXCSR ;MASTER RESET
1743
1744                                     ;SET MAINTENANCE MODE & SEND
1745                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1746 005446 012777 004020 005430      MOV    #MINT!SEND,@TXCSR
1747
1748                                     ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1749 005454 012777 027026 005416      MOV    #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
1750 005462 016703 005406      MOV    RXDBUF,R3      ;SETUP FOR ERROR MSG
1751 005466 005004      CLR    R4              ;FOR DATA CHAR CREATION
1752 005470 110477 005414      MOVB   R4,@TXDBUF     ;LOAD CHARACTER
1753 005474 052777 000020 005366      BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
1754                                     ;GET INTO SYNCHRONIZATION
1755 005502 052777 020000 005374      BIS    #CLK,@TXCSR    ;POKE CLK UP
1756 005510 042777 020000 005366      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1757 005516 012767 000011 173414 1$:  MOV    #9,SHIFT        ;# OF SHIFTS
1758 005524 010400      MOV    R4,R0          ;EXPECTED
1759 005526
1760 005526 052777 020000 005350 2$:  BIS    #CLK,@TXCSR    ;POKE CLK UP
1761 005534 042777 020000 005342      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1762 005542 005367 173372      DEC    SHIFT          ;# OF SHIFTS
1763 005546 022767 000003 173364      CMP    #3,SHIFT       ;TIME TO LOAD NEXT CHAR ?
1764 005554 001003      BNE    3$             ;NO ?
1765 005556 005204      INC    R4             ;GENERATE NEXT CHAR
1766 005560 110477 005324      MOVB   R4,@TXDBUF     ;LOAD NEXT CHARACTER
1767 005564 005767 173350 3$:  TST    SHIFT          ;IS IT 0 ?
1768 005570 001356      BNE    2$
1769 005572 105777 005272      TSTB   @RXCSR         ;RXDONE = 1 ?
1770 005576 100401      BMI    5$
1771 005600 104000      HLT
1772 005602
1773 005602 017701 005266 5$:  MOV    @RXDBUF,R1     ;ACTUAL
1774 005606 020001      CMP    R0,R1          ;COMPARE EXP VS ACT
1775 005610 001401      BEQ    6$
1776 005612 104002      HLT    2             ;CHARACTERS SHOULD COMPARE
1777 005614
1778 005614 105704 6$:  TSTB   R4             ;LAST CHARACTER ?
1779 005616 001337      BNE    1$            ;NO
1780 005620 104400 4$:  SCOPE

```

```

1781                                     ::THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1782                                     ::BOTH THE TRANSMITTER AND RECEIVER LOGIC
1783                                     ::MODE:SYNC EXTERNAL (SYNEXT)
1784                                     ::LENGTH:EIGHT PLUS PARITY
1785                                     ::PARITY:EVEPAR
1786                                     ::MAINT. MODE:MEXT
1787                                     ::
1788 005622 012767 000007 173276 TST7: MOV #7,TSTNO ;SAVE THIS
1789 005630 012767 006102 173260 MOV #TST8,NEXT ;GO TO THIS TEST WHEN THRU
1790 005636 105767 173341 TSTB JMRBY ;JUMP AROUND TEST ?
1791 005642 100116 BPL 4$ ;YES ?
1792 005644 052777 000400 005232 BIS #MRESET,@TXCSR ;MASTER RESET
1793 005652 012777 020000 005220 MOV #SYNEXT,@PARCSR ;SET THE MODE
1794 005660 052777 000400 005216 BIS #MRESET,@TXCSR ;MASTER RESET
1795
1796                                     ;SET MAINTENANCE MODE & SEND
1797                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1798 005666 012777 010020 005210 MOV #MEXT!SEND,@TXCSR
1799
1800                                     ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1801 005674 012777 027426 005176 MOV #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
1802 005702 016703 005166 MOV RXDBUF,R3 ;SETUP FOR ERROR MSG
1803 005706 005004 CLR R4 ;FOR DATA CHAR CREATION
1804 005710 110477 005174 MOV#B R4,@TXDBUF ;LOAD CHARACTER
1805 005714 052777 000020 005146 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1806                                     ;GET INTO SYNCHRONIZATION
1807 005722 052777 020000 005154 BIS #CLK,@TXCSR ;POKE CLK UP
1808                                     ;WAIT FOR CABLE & DRIVER DELAYS
1809 005730 016702 173202 MOV HOLD,R2 ;WAIT THIS AMT
1810                                     64$:
1811 005734 005302 DEC R2 ;WAIT
1812 005736 001376 BNE 64$
1813                                     ;EXIT...
1814 005740 042777 020000 005136 BIC #CLK,@TXCSR ;POKE CLK DOWN
1815                                     ;WAIT FOR CABLE & DRIVER DELAYS
1816 005746 016702 173164 MOV HOLD,R2 ;WAIT THIS AMT
1817                                     65$:
1818 005752 005302 DEC R2 ;WAIT
1819 005754 001376 BNE 65$
1820                                     ;EXIT...
1821 005756 012767 000011 173154 1$: MOV #9,SHIFT ;# OF SHIFTS
1822 005764 010400 MOV R4,R0 ;EXPECTED
1823                                     2$:
1824 005766 052777 020000 005110 BIS #CLK,@TXCSR ;POKE CLK UP
1825                                     ;WAIT FOR CABLE & DRIVER DELAYS
1826 005774 016702 173136 MOV HOLD,R2 ;WAIT THIS AMT
1827                                     66$:
1828 006000 005302 DEC R2 ;WAIT
1829 006002 001376 BNE 66$
1830                                     ;EXIT...
1831 006004 042777 020000 005072 BIC #CLK,@TXCSR ;POKE CLK DOWN
1832                                     ;WAIT FOR CABLE & DRIVER DELAYS
1833 006012 016702 173120 MOV HOLD,R2 ;WAIT THIS AMT
1834                                     67$:
1835 006016 005302 DEC R2 ;WAIT
1836 006020 001376 BNE 67$

```


1837					:EXIT...
1838	006022	005367	173112		DEC SHIFT ;# OF SHIFTS
1839	006026	022767	000003	173104	CMP #3,SHIFT ;TIME TO LOAD NEXT CHAR ?
1840	006034	001003			BNE 3\$;NO ?
1841	006036	005204			INC R4 ;GENERATE NEXT CHAR
1842	006040	110477	005044		MOVB R4,@TXDBUF ;LOAD NEXT CHARACTER
1843	006044	005767	173070	3\$:	TST SHIFT ;IS IT 0 ?
1844	006050	001346			BNE 2\$
1845	006052	105777	005012		TSTB @RXCSR ;RXDONE = 1 ?
1846	006056	100401			BMI 5\$
1847	006060	104000			HLT ;RXDONE SHOULD BE SET
1848	006062			5\$:	
1849	006062	017701	005006		MOV @RXDBUF,R1 ;ACTUAL
1850	006066	020001			CMP R0,R1 ;COMPARE EXP VS ACT
1851	006070	001401			BEQ 6\$
1852	006072	104002			HLT 2 ;CHARACTERS SHOULD COMPARE
1853	006074			6\$:	
1854					:CHECK OUT MODEM BYPASS JUMPER
1855	006074	105704			TSTB R4 ;LAST CHARACTER ?
1856	006076	001327			BNE 1\$;NO
1857	006100	104400		4\$:	SCOPE

```

1858                                     ::THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1859                                     ::BOTH THE TRANSMITTER AND RECEIVER LOGIC
1860                                     ::MODE:SYNC EXTERNAL (SYNEXT)
1861                                     ::LENGTH:EIGHT PLUS PARITY
1862                                     ::PARITY:ODDPAR
1863                                     ::MAINT. MODE:MEXT
1864                                     ::
1865 006102 012767 000010 173016 TST8:  MOV    #8,TSTNO           ;SAVE THIS
1866 006110 012767 006362 173000      MOV    #.EOP,NEXT         ;GO TO THIS TEST WHEN THRU
1867 006116 105767 173061              TSTB   JMRBY             ;JUMP AROUND TEST ?
1868 006122 100116                    BPL    4$                ;YES ?
1869 006124 052777 000400 004752      BIS    #MRESET,@TXCSR   ;MASTER RESET
1870 006132 012777 020000 004740      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1871 006140 052777 000400 004736      BIS    #MRESET,@TXCSR   ;MASTER RESET
1872
1873                                     ;SET MAINTENANCE MODE & SEND
1874                                     ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1875 006146 012777 010020 004730      MOV    #MEXT!SEND,@TXCSR
1876
1877                                     ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1878 006154 012777 027026 004716      MOV    #SYNEXT!EIGHT!ODDPAR!26,@PARCSR
1879 006162 016703 004706      MOV    RXDBUF,R3        ;SETUP FOR ERROR MSG
1880 006166 005004                    CLR    R4               ;FOR DATA CHAR CREATION
1881 006170 110477 004714      MOV    R4,@TXDBUF       ;LOAD CHARACTER
1882 006174 052777 000020 004666      BIS    #SYNSCH,@RXCSR   ;SET SEARCH SYNC
1883
1884 006202 052777 020000 004674      ;GET INTO SYNCHRONIZATION
1885                                     BIS    #CLK,@TXCSR      ;POKE CLK UP
1886                                     ;WAIT FOR CABLE & DRIVER DELAYS
1886 006210 016702 172722      MOV    HOLD,R2          ;WAIT THIS AMT
1887 006214
1888 006214 005302      64$:  DEC    R2              ;WAIT
1889 006216 001376      BNE    64$
1890                                     ;EXIT...
1891 006220 042777 020000 004656      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1892                                     ;WAIT FOR CABLE & DRIVER DELAYS
1893 006226 016702 172704      MOV    HOLD,R2          ;WAIT THIS AMT
1894 006232
1895 006232 005302      65$:  DEC    R2              ;WAIT
1896 006234 001376      BNE    65$
1897                                     ;EXIT...
1898 006236 012767 000011 172674 1$:  MOV    #9,SHIFT        ;# OF SHIFTS
1899 006244 010400      MOV    R4,R0           ;EXPECTED
1900 006246
1901 006246 052777 020000 004630 2$:  BIS    #CLK,@TXCSR     ;POKE CLK UP
1902                                     ;WAIT FOR CABLE & DRIVER DELAYS
1903 006254 016702 172656      MOV    HOLD,R2          ;WAIT THIS AMT
1904 006260
1905 006260 005302      66$:  DEC    R2              ;WAIT
1906 006262 001376      BNE    66$
1907                                     ;EXIT...
1908 006264 042777 020000 004612      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1909                                     ;WAIT FOR CABLE & DRIVER DELAYS
1910 006272 016702 172640      MOV    HOLD,R2          ;WAIT THIS AMT
1911 006276
1912 006276 005302      67$:  DEC    R2              ;WAIT
1913 006300 001376      BNE    67$

```


1914					:EXIT...
1915	006302	005367	172632		DEC SHIFT ;# OF SHIFTS
1916	006306	022767	000003	172624	CMP #3,SHIFT ;TIME TO LOAD NEXT CHAR ?
1917	006314	001003			BNE 3\$;NO ?
1918	006316	005204			INC R4 ;GENERATE NEXT CHAR
1919	006320	110477	004564		MOVB R4,@TXDBUF ;LOAD NEXT CHARACTER
1920	006324	005767	172610	3\$:	TST SHIFT ;IS IT 0 ?
1921	006330	001346			BNE 2\$
1922	006332	105777	004532		TSTB @RXCSR ;RXDONE = 1 ?
1923	006336	100401			BMI 5\$
1924	006340	104000			HLT ;RXDONE SHOULD BE SET
1925	006342			5\$:	
1926	006342	017701	004526		MOV @RXDBUF,R1 ;ACTUAL
1927	006346	020001			CMP R0,R1 ;COMPARE EXP VS ACT
1928	006350	001401			BEQ 6\$
1929	006352	104002			HLT 2 ;CHARACTERS SHOULD COMPARE
1930	006354			6\$:	
1931					:CHECK OUT MODEM BYPASS JUMPER
1932	006354	105704			TSTB R4 ;LAST CHARACTER ?
1933	006356	001327			BNE 1\$;NO
1934	006360	104400		4\$:	SCOPE

```

1935
1936
1937
1938
1939
1940
1941
1942 006362 104402
1943 006364 011524
1944 006366 104410 006620
1945 006372 104402 011245
1946 006376 105767 172600
1947 006402 001511
1948 006404 005767 172606
1949 006410 001007
1950 006412 104402 011257
1951 006416 016700 172574
1952 006422 000000
1953
1954 006424 000167 172770
1955 006430 062767 000010 172546 RUNIT:
1956 006436 062767 000010 172546 ZERO:
1957 006444 000241
1958 006446 006167 172546
1959 006452 103410
1960
1961 006454 036767 172540 172534
1962 006462 001762
1963 006464 004767 000034
1964 006470 000167 000174
1965 006474 012767 000001 172516 2$:
1966
1967 006502 016767 172500 172474
1968 006510 016767 172500 172474
1969 006516 004767 000002
1970 006522 000441
1971 006524 016767 172454 004040 REPLAY:
1972 006532 004767 003702
1973 006536 016767 172450 004350
1974 006544 062767 000002 172440
1975 006552 016767 172434 004336
1976 006560 062767 000002 172424
1977 006566 016767 172420 004324
1978 006574 062767 000002 172410
1979 006602 016767 172404 004312
1980 006610 016767 004300 172374
1981 006616 000207
1982
1983 006620 000001
1984 006622 006 002
1985 006624 013070
1986
1987 006626
1988 006626 005067 172302
1989 006632 005067 172410
1990 006636 005267 172266
  
```

```

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

```

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

```

MEPASS
CONVRT ,OUTCRY
TYPE ,DEVICE
TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
BEQ CCC ;NO, JUMP AROUND
TST ACTREG ;ARE ANY DEVICES ACTIVE?
BNE RUNIT ;YES
TYPE ,MCOW ;NO
MOV ACTREG,RO ;DISPLAY ACTREG
HALT ;SELECT SOMETHING TO RUN @ ACTREG:
;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
JMP .START ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
CLC
ROL ROTADD ;UP DATE ROTATING POINTER
BCS 2$ ;IS IT THE LAST DEVICE
;TO BE TESTED IN THIS PASS ?
BIT ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
JSR PC,REPLAY ;CALCULATE NEW PARAMETERS
JMP RESTRT ;YES IT WAS ACTIVE,TEST THIS DEVICE
MOV #1,ROTADD ;OK!,NOW SET UP ROTATING
;POINTER FOR NEXT MULTIPLE PASS
MOV KEEPADD,BASEADD ;RESTORE BASE ADDRESS
MOV KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
JSR PC,REPLAY ;CALC NEW PARAMETERS
BR CCC ;JUMP AROUND REPLAY
REPLAY: MOV BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
JSR PC,DUADDR ;CREATE NEW ADDRESSES
MOV BASEIV,DURIV ;CREATE DURIV
ADD #2,BASEIV
MOV BASEIV,DURIS ;CREATE DURIS
ADD #2,BASEIV
MOV BASEIV,DUTIV ;CREATE DUTIV
ADD #2,BASEIV
MOV BASEIV,DUTIS ;CREATE DUTIS
MOV DURIV,BASEIV ;RESTORE
RTS PC
  
```

```

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

```

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



```

1991 006642 016777 172262 172232      MOV      PASCNT,@LIGHTS      ;DISPLAY PASS COUNT
1992 006650 013701 000042                MOV      @#42,R1           ;CHECK FOR ACT-11 OR DDP
1993 006654 001405                BEQ      RESTRT           ;IF NOT, CONTINUE TESTING
1994 006656 000005                RESET
1995 006660 004711                LOGICAL: JSR      PC,(R1)
1996 006662 000240                NOP
1997 006664 000240                NOP
1998 006666 000240                NOP
1999 006670 012767 000340 171100 RESTRT: MOV      #340,PS      ;PREVENT INTERRUPTS (PRIO: 7)
2000 006676 104413                CKSWR           ;CHECK FOR ^G
2001 006700 012767 003446 172206      MOV      #TST1,RTRN
2002 006706 000167 174534                JMP      TST1
2003
2004                ;SCOPE LOOP AND ITERATION HANDLER
2005
2006 006712                .SCOPE:
2007                ;***** START OF CODE FOR THE X OR TESTER *****
2008 006712 000424                BR      4$           ;IF RUNNING ON THE X OR TESTER CHANGE
2009                ;THIS INSTRUCTION TO A 'NOP'(NOP=240)
2010 006714 013746 000004                MOV      @#4,-(SP)       ;SAVE CONTENTS OF ERROR VECTOR
2011 006720 012737 006740 000004      MOV      #1$,@#4        ;SET FOR TIME OUT
2012 006726 005737 177060                TST      @#177060       ;TIME OUT ON X OR ?
2013 006732 012637 000004      MOV      (SP)+,@#4      ;RESTORE ERROR VECTOR
2014 006736 000404                BR      2$           ;GO TO NEXT TEST
2015 006740 022626                1$:  CMP      (SP)+,(SP)+  ;CLEAR THE STACK AFTER A TIMEOUT
2016 006742 012637 000004      MOV      (SP)+,@#4      ;RESTORE ERROR VECTOR
2017 006746 000403                BR      3$           ;LOOP ON PRESENT TEST
2018 006750 016767 172142 172136 2$:  MOV      NEXT,RTRN      ;SET UP NEXT TEST IN RTRN
2019 006756 016716 172132 3$:  MOV      RTRN,(SP)     ;SET UP STACK FOR RTI
2020 006762 000002                RTI
2021 006764                4$:  ;***** END OF CODE FOR THE X OR TESTER *****
2022 006764 104413                CKSWR           ;CHECK FOR ^G
2023 006766 032777 040000 172104 TTST: BIT      #SW14,@SWR    ;LOOP ON CURRENT TEST ?
2024 006774 001407                BEQ      1$
2025 006776 000432                BR      3$
2026 007000 105777 172100                TSTB    @TKCSR         ;TEST TTY FLAG
2027 007004 100027                BPL      3$
2028 007006 017700 172074                MOV      @TKDBR,R0     ;CLR DONE BIT
2029 007012 000412                BR      2$           ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2030 007014 032777 004000 172056 1$:  BIT      #SW11,@SWR    ;INHIBIT ITERATIONS ?
2031 007022 001006                BNE      2$
2032 007024 005267 172074                INC      LPCNT
2033 007030 026767 172070 172064      CMP      LPCNT,ICOUNT  ;CHECK FOR ITERATION CNT FINISH
2034 007036 101412                BLOS    3$
2035 007040 105067 172202 2$:  CLRB    ERRFLG
2036 007044 005067 172054                CLR      LPCNT
2037 007050 012767 000005 172044      MOV      #5,ICOUNT     ;SET UP ITERATION COUNT
2038 007056 016767 172034 172030      MOV      NEXT,RTRN    ;SET UP NEXT TEST IN RTRN
2039 007064 016716 172024 3$:  MOV      RTRN,(SP)     ;SET UP STACK FOR RTI
2040 007070 000002                RTI
2041 007072 001407                BRW:  1407           ;RESTORE 'BEQ 1$' INSTRUCTION
2042 007074 000432                BRX:  432           ;RESTORE 'BR 3$' INSTRUCTION
2043
2044                ;CHECK FOR FREEZE ON CURRENT DATA
2045
2046 007076 104413                .SCOPE1: CKSWR        ;CHECK FOR ^G
  
```

```

2047 007100 032777 001000 171772      BIT      #SW09,@SWR
2048 007106 001402                BEQ      1$
2049 007110 016716 172004                MOV      LOCK,(SP)
2050 007114 000002                1$:     RTI
2051
2052                                ;TELETYPE OUTPUT ROUTINE
2053
2054 007116 010546                .TYPE:  MOV      R5,-(SP)
2055 007120 017605 000002                MOV      @2(SP),R5
2056 007124 062766 000002 000002        ADD      #2,2(SP)
2057 007132 105715                1$:     TSTB   (R5)                ;LOOK FOR '0'
2058 007134 001406                BEQ      3$
2059 007136 105777 171746                2$:     TSTB   @TPCSR                ;TEST DONE BIT
2060 007142 100375                BPL      2$
2061 007144 112577 171742                MOV      (R5)+,@TPDBR                ;TYPE CHAR
2062 007150 000770                BR       1$                ;DO IT AGAIN UNTIL '0' IS SEEN
2063 007152 012605                3$:     MOV      (SP)+,R5
2064 007154 000002                RTI
2065
2066                                ;ASCII STRING INPUT ROUTINE
2067
2068 007156 010346                .INSTR: MOV      R3,-(SP)
2069 007160 010446                MOV      R4,-(SP)
2070 007162 017667 000004 000010        MOV      @4(SP),MSG
2071 007170 062766 000002 000004        ADD      #2,4(SP)
2072 007176 104402                .INST1: TYPE
2073 007200 000000                .MSG:   0
2074 007202 012704 012224                MOV      #INBUF,R4
2075 007206 012703 000007                MOV      #7,R3
2076 007212 105777 171666                1$:     TSTB   @TKCSR
2077 007216 100375                BPL      1$
2078 007220 117714 171662                MOV      @TKDBR,(R4)
2079 007224 142714 000200                BICB   #200,(R4)
2080 007230 121427 000025                CMPB   (R4),#25                ;IS IT <^U>
2081 007234 001003                BNE     200$
2082 007236 104402 011434                TYPE,MCRLF
2083 007242 000755                BR      .INST1
2084 007244 122427 000015                200$:  CMPB   (R4)+,#15
2085 007250 001423                BEQ     INSTR2
2086 007252 117777 171630 171632        MOV      @TKDBR,@TPDBR
2087 007260 105777 171624                2$:     TSTB   @TPCSR
2088 007264 100375                BPL      2$
2089 007266 005303                DEC     R3
2090 007270 001350                BNE     1$
2091 007272 000402                BR      .INSTG
2092 007274 010346                .INSTE: MOV      R3,-(SP)
2093 007276 010446                MOV      R4,-(SP)
2094 007300 104402                .INSTG: TYPE
2095 007302 011430                MQM
2096 007304 005737 010572                TST     @#RDSW
2097 007310 001402                BEQ     400$
2098 007312 104402 011434                TYPE,MCRLF
2099 007316 000727                400$:  BR      .INST1
2100 007320 012604                INSTR2: MOV      (SP)+,R4
2101 007322 012603                MOV      (SP)+,R3
2102 007324 000002                RTI

```



```

2103
2104                                     ;CONVERT ASCII STRING TO OCTAL
2105
2106 007326 010546      .PARAM: MOV     R5,-(SP)
2107 007330 010446      MOV     R4,-(SP)
2108 007332 016605 000004  MOV     4(SP),R5
2109 007336 012567 000170  MOV     (R5)+,LOLIM
2110 007342 012567 000166  MOV     (R5)+,HILIM
2111 007346 012567 000164  MOV     (R5)+,DEVADR
2112 007352 112567 000162  MOV     (R5)+,LOBITS
2113 007356 112567 000157  MOV     (R5)+,ADRCNT
2114 007362 010566 000004  MOV     R5,4(SP)
2115 007366 005005      PARAM1: CLR     R5
2116 007370 012704 012224  MOV     #INBUF,R4
2117 007374 122714 000015  CMPB   #15,(R4)
2118 007400 001420      BEQ    PARERR
2119 007402 121427 000060  1$:    CMPB   (R4),#60
2120 007406 002415      BLT    PARERR
2121 007410 121427 000067  CMPB   (R4),#67
2122 007414 003012      BGT    PARERR
2123 007416 142714 000060  BICB   #60,(R4)
2124 007422 152405      BISB   (R4)+,R5
2125 007424 122714 000015  CMPB   #15,(R4)
2126 007430 001414      BEQ    LIMITS
2127 007432 006305      ASL    R5
2128 007434 006305      ASL    R5
2129 007436 006305      ASL    R5
2130 007440 000760      BR     1$
2131 007442 122714 000015  PARERR: CMPB   #15,(R4)          ;IS FIRST CHARACTER A <CR>
2132 007446 001003      BNE    120$
2133 007450 005737 010572  TST    @#RDSW          ;IS CKSWR ROUTINE BEING USED
2134 007454 001023      BNE    PARTI
2135 007456 104404      120$: INSTER
2136 007460 000742      BR     PARAM1
2137
2138                                     ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2139
2140 007462 020567 000046  LIMITS: CMP    R5,HILIM
2141 007466 101365      BHI    PARERR
2142 007470 020567 000036  CMP    R5,LOLIM
2143 007474 103762      BLO    PARERR
2144 007476 136705 000036  BITB   LOBITS,R5
2145 007502 001357      BNE    PARERR
2146
2147                                     ;STORE NUMBER AT SPECIFIED ADDRESS
2148
2149 007504 016704 000026  1$:    MOV     DEVADR,R4
2150 007510 010524      MOV     R5,(R4)+
2151 007512 062705 000002  ADD     #2,R5
2152 007516 105367 000017  DECB   ADRCNT
2153 007522 001372      BNE    1$
2154 007524 012604      PARTI: MOV     (SP)+,R4
2155 007526 012605      MOV     (SP)+,R5
2156 007530 000002      RTI
2157 007532 000000      LOLIM: 0
2158 007534 000000      HILIM: 0

```

```

2159 007536 000000          DEVADR: 0
2160 007540 000000          LOBITS: 0
2161          007541          ADRCNT=LOBITS+1
2162
2163          ;SAVE PC OF TEST THAT FAILED AND R0-R5
2164
2165 007542 016667 000004 171424 .SAV05: MOV    4(SP),SAVPC
2166
2167          ;SAVE R0-R5
2168
2169 007550 010567 171414      SV05:  MOV    R5,SAVR5
2170 007554 010467 171406      MOV    R4,SAVR4
2171 007560 010367 171400      MOV    R3,SAVR3
2172 007564 010267 171372      MOV    R2,SAVR2
2173 007570 010167 171364      MOV    R1,SAVR1
2174 007574 010067 171356      MOV    R0,SAVR0
2175 007600 000002          RTI
2176
2177          ;RESTORE R0-R5
2178
2179 007602 016700 171350      .RES05: MOV    SAVR0,R0
2180 007606 016701 171346      MOV    SAVR1,R1
2181 007612 016702 171344      MOV    SAVR2,R2
2182 007616 016703 171342      MOV    SAVR3,R3
2183 007622 016704 171340      MOV    SAVR4,R4
2184 007626 016705 171336      MOV    SAVR5,R5
2185 007632 000002          RTI
2186
2187          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2188
2189 007634 104402          .CONVR: TYPE
2190 007636 011434          MCRLF
2191 007640 010046          .CNVRT: MOV    R0,-(SP)
2192 007642 010146          MOV    R1,-(SP)
2193 007644 010346          MOV    R3,-(SP)
2194 007646 010446          MOV    R4,-(SP)
2195 007650 010546          MOV    R5,-(SP)
2196 007652 017601 000012      MOV    @12(SP),R1
2197 007656 016767 002402 171264 MOV    TEMP,TEMP3
2198 007664 062766 000002 000012 ADD    #2,12(SP)
2199 007672 012167 000154      MOV    (R1)+,WRDCNT
2200 007676 112167 000152      1$:  MOV    (R1)+,CHRCNT
2201 007702 112167 000147      MOV    (R1)+,SPACNT
2202 007706 013167 000144      MOV    @(R1)+,BINWRD
2203 007712 016704 000140      2$:  MOV    BINWRD,R4
2204 007716 116705 000132      MOV    CHRCNT,R5
2205 007722 012700 012264      MOV    #TEMP,R0
2206 007726 010403          3$:  MOV    R4,R3
2207 007730 042703 177770      BIC    #177770,R3
2208 007734 062703 000060      ADD    #060,R3
2209 007740 110320          MOV    R3,(R0)+
2210 007742 006204          ASR    R4
2211 007744 042704 100000      BIC    #100000,R4
2212 007750 006204          ASR    R4
2213 007752 006204          ASR    R4
2214 007754 005305          DEC    R5

```

```

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

```



```

2215 007756 001363      BNE      3$
2216 007760 012703 012324  MOV      #MDATA,R3
2217 007764 114023      4$:      MOVB    -(R0),(R3)+
2218 007766 105367 000062  DECB    CHRCNT
2219 007772 001374      BNE      4$
2220 007774 105767 000055  TSTB    SPACNT
2221 010000 001405      BEQ      6$
2222 010002 112723 000040  5$:      MOVB    #040,(R3)+
2223 010006 105367 000043  DECB    SPACNT
2224 010012 001373      BNE      5$
2225 010014 105013      6$:      CLRB    (R3)
2226 010016 104402      TYPE
2227 010020 012324      MDATA
2228 010022 005367 000024  DEC      WRDCNT
2229 010026 001323      BNE      1$
2230 010030 016767 171114 002226  MOV      TEMP3,TEMP
2231 010036 012605      MOV      (SP)+,R5
2232 010040 012604      MOV      (SP)+,R4
2233 010042 012603      MOV      (SP)+,R3
2234 010044 012601      MOV      (SP)+,R1
2235 010046 012600      MOV      (SP)+,R0
2236 010050 000002      RTI
2237 010052 000000      WRDCNT: 0
2238 010054 000000      CHRCNT: 0
2239          010055      SPACNT=CHRCNT+1
2240 010056 000000      BINWRD: 0
2241
2242          :COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
2243          :BUFFER TO THE CHARACTERS 'N' AND 'Y'.
2244          :IF THE CHARACTER IS 'N' CLEAR THE FLAG
2245          :IF THE CHARACTER IS 'Y' SET THE FLAG
2246
2247 010060 017605 000000      .SETFLG:MOV    @ (SP),R5
2248 010064 122767 000116 002132  CMPB    #'N',INBUF      ;IS IT 'N' ?
2249 010072 001002      BNE      1$
2250 010074 105015      CLRB    (R5)      ;000
2251 010076 000406      BR      2$
2252 010100 122767 000131 002116  1$:      CMPB    #'Y',INBUF      ;IS IT 'Y' ?
2253 010106 001005      BNE      3$
2254 010110 112715 177777      MOVB    #-1,(R5)      ;377
2255 010114 062716 000002      2$:      ADD     #2,(SP)
2256 010120 000002      RTI
2257 010122 104404      3$:      INSTER      ;RETRY
2258 010124 000755      BR      .SETFLG
2259          :TRAP DISPATCH SERVICE
2260          :ARGUMENT OF TRAP IS EXTRACTED
2261          :AND USED AS OFFSET TO OBTAIN POINTER
2262          :TO SELECTED SUBROUTINE
2263
2264 010126 011646      .TRPSR:MOV    (SP),-(SP)      ;GET PC OF RETURN
2265 010130 162716 000002      SUB     #2,(SP)      ;=PC OF TRAP
2266 010134 017616 000000      MOV     @ (SP),(SP)      ;GET TRP
2267 010140 006316      TRPOK:ASL    (SP)      ;MULTIPLY TRAP ARG BY 2
2268 010142 042716 177001      BIC     #177001,(SP)      ;CLEAR UNWANTED BITS
2269 010146 062716 001366      ADD     #.TRPTAB,(SP)      ;POINTER TO SUBROUTINE ADDRESS
2270 010152 017616 000000      MOV     @ (SP),(SP)      ;SUBROUTINE ADDRESS
  
```

```

2271 010156 000136          JMP      @ (SP)+          ;GO TO SUBROUTINE
2272
2273          ;ERROR HANDLER
2274
2275 010160 104413          .HLT:   CKSWR          ;CHECK FOR ^G
2276 010162 032777 020000 170710 BIT      #SW13,@SWR      ;INHIBIT ERROR TYPE OUT ?
2277 010170 001061          BNE     HALTS
2278 010172 021667 170736   CMP      (SP),LSTERR
2279 010176 001404          BEQ     1$
2280 010200 011667 170730   MOV      (SP),LSTERR
2281 010204 105067 171036   CLRB    ERRFLG
2282 010210 104406          1$:    SAVO5
2283 010212 011605          MOV      (SP),R5
2284 010214 162705 000002   SUB      #2,R5
2285 010220 011504          MOV      (R5),R4
2286 010222 006304          ASL     R4
2287 010224 061504          ADD     (R5),R4
2288 010226 006304          ASL     R4
2289 010230 042704 177001   BIC     #177001,R4
2290 010234 062704 013040   ADD     #.ERRTAB,R4
2291 010240 012467 000040   MOV     (R4)+,ERRMSG
2292 010244 012467 000046   MOV     (R4)+,DATAHD
2293 010250 011467 000054   MOV     (R4),DATABP
2294 010254 105767 170766   TSTB   ERRFLG
2295 010260 001403          BEQ     TYPMSG
2296 010262 005767 000042   TST    DATABP
2297 010266 001014          BNE     TYPDAT
2298 010270 104410          TYPMSG: CONVRT
2299 010272 010422          ERTAB0
2300 010274 112767 177777 170744 MOVB    #-1,ERRFLG
2301 010302 104402          TYPE
2302 010304 000000          ERRMSG: 0
2303 010306 005767 000004   TST    DATAHD
2304 010312 001402          BEQ     TYPDAT
2305 010314 104402          TYPE
2306 010316 000000          DATAHD: 0
2307 010320 005767 000004   TYPDAT: TST    DATABP
2308 010324 001402          BEQ     RESREG
2309 010326 104410          CONVRT
2310 010330 000000          DATABP: 0
2311 010332 104407          RESREG: RES05
2312 010334 005777 170540   HALTS:  TST    @SWR
2313 010340 100005          BPL    EXITER
2314 010342 010046          PUSHRO
2315 010344 016600 000002   MOV     2(SP),R0
2316 010350 000000          HALT
2317 010352 012600          POPRO
2318 010354 104413          EXITER: CKSWR          ;CHECK FOR ^G
2319 010356 005267 170550   INC     ERRCNT
2320 010362 032777 000400 170510 BIT     #SW08,@SWR      ;LOOP ON ERROR ?
2321 010370 001007          BNE     1$
2322 010372 032777 002000 170500 BIT     #SW10,@SWR      ;ESCAPE TO NEXT ON ERROR ?
2323 010400 001407          BEQ     2$
2324 010402 016767 170510 170504 MOV     NEXT,RTRN      ;SET UP FOR NEXT TEST
2325 010410 012706 001100          1$:   MOV     #STACK,SP      ;REINITIALIZE SP
2326 010414 000177 170474          JMP     @RTRN
  
```



```

2327 010420 000002          2$: RTI
2328 010422 000001          ERTABO: 1
2329 010424 006          002      .BYTE 6,2
2330 010426 001174          SAVPC
2331                                     ;ENTER HERE ON POWER FAILURE
2332
2333
2334 010430 010046          .PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
2335 010432 010146          MOV R1,-(SP)
2336 010434 010246          MOV R2,-(SP)
2337 010436 010346          MOV R3,-(SP)
2338 010440 010446          MOV R4,-(SP)
2339 010442 010546          MOV R5,-(SP)
2340 010444 016746 167354  MOV 24,-(SP)
2341 010450 010667 170516  MOV SP,SAVSP
2342 010454 012767 010466 167342  MOV #RESTART,24 ;SAVE STACK POINTER
2343 010462 000000          HALT ;SET UP FOR POWER UP TRAP
2344 010464 000777          1$: BR 1$ ;HALT ON POWER DOWN NORMAL
2345
2346                                     ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2347
2348 010466 016706 170500  RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER
2349 010472 012605          MOV (SP)+,R5 ;RESTORE R0-R5
2350 010474 012604          MOV (SP)+,R4
2351 010476 012603          MOV (SP)+,R3
2352 010500 012602          MOV (SP)+,R2
2353 010502 012601          MOV (SP)+,R1
2354 010504 012600          MOV (SP)+,R0
2355 010506 012767 010430 167310  MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
2356 010514 012767 000340 167254  MOV #340,PS
2357 010522 012706 001100          MOV #STACK,SP
2358 010526 005067 001532          CLR TEMP
2359 010532 005267 001526          1$: INC TEMP
2360 010536 001375          BNE 1$
2361 010540 104410          CONVRT
2362 010542 010564          PFTAB
2363 010544 104402          TYPE
2364 010546 011437          MPFAIL
2365 010550 005067 170472          CLR ERRFLG
2366 010554 005067 170354          CLR LSTERR
2367 010560 000177 170330          JMP @RTRN
2368 010564 000001          PFTAB: 1
2369 010566 006          002      .BYTE 6,2
2370 010570 001114          RTRN
2371
2372
2373                                     ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
2374                                     ;OF LOC.176.
2375                                     ;LOCATIONS USED:
2376 010572 000000          RDSW: .WORD 0
2377
2378
2379 010574 005737 000042          .CKSWR: TST @#42
2380 010600 001042          BNE OUT
2381 010602 022767 000176 170270  CMP #CUREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
2382 010610 001036          BNE OUT ;NO, GET OUT

```

2383	010612	105777	170266		TSTB	@TKCSR	:YES, WAIT FOR
2384	010616	100033			BPL	OUT	:READY, GET CHARACTER
2385	010620	017767	170262	176352	MOV	@TKDBR, .MSG	:AND STRIP OFF
2386	010626	042767	177600	176344	BIC	#177600, .MSG	:THE GARBAGE
2387	010634	122767	000007	176336	CMPB	#7, .MSG	:IS IT A <^G>
2388	010642	001021			BNE	OUT	
2389	010644	104402	010722		TYPE, \$CNTG		
2390	010650	005137	010572		.CNTLU: COM	@#RDSW	
2391	010654	104402	010727		TYPE, \$MSWR		
2392	010660	104411	010714		CNVRT, SWREGC		
2393	010664	104403	010737		INSTR, \$MNEW		
2394	010670	104405			PARAM		
2395	010672	000000			0		
2396	010674	177777			177777		
2397	010676	000176			SWREG		
2398	010700	000	001		.BYTE	0,1	
2399	010702	104402	011434		TYPE, MCRIF		
2400	010706	005037	010572		OUT: CLR	@#RDSW	
2401	010712	000002			RTI		
2402	010714	000001			SWREGC: 1		
2403	010716	006	002		.BYTE	6,2	
2404	010720	000176			SWREG		
2405	010722	005015	043536	000	\$CNTG: .ASCIZ	<15><12>/^G/	
2406	010727	015	051412	051127	\$MSWR: .ASCIZ	<15><12>/SWR= /	
2407	010734	020075	000				
2408	010737	040	047040	053505	\$MNEW: .ASCIZ	/ NEW= /	
2409	010744	020075	000				
2410		010750			.EVEN		
2411	010750	005015	042012	030525	MTITLE: .ASCIZ	<15><12><12>/DU11 CZDUF-D TAPE F /<15><12>	
2412	010756	020061	055103	052504			
2413	010764	026506	020104	040524			
2414	010772	042520	043040	006440			
2415	011000	000012					
2416	011002	005015	042526	052103	MVECTO: .ASCIZ	<15><12>/VECTOR ADDRESS-/	
2417	011010	051117	040440	042104			
2418	011016	042522	051523	000055			
2419	011024	005015	051461	020124	MREGAD: .ASCIZ	<15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/	
2420	011032	042504	044526	042503			
2421	011040	020072	042522	042503			
2422	011046	053111	051105	041440			
2423	011054	047117	051124	046117			
2424	011062	051040	043505	051511			
2425	011070	042524	020122	042101			
2426	011076	051104	051505	026523			
2427	011104	000					
2428	011105	015	040412	042522	MMULT: .ASCIZ	<15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/	
2429	011112	054440	052517	051040			
2430	011120	047125	044516	043516			
2431	011126	046440	046125	044524			
2432	011134	046120	020105	042504			
2433	011142	044526	042503	020123			
2434	011150	020077	054450	047440			
2435	011156	020122	024516	000055			
2436	011164	005015	040514	052123	MLASTD: .ASCIZ	<15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/	
2437	011172	042040	053105	041511			
2438	011200	035105	042522	042503			

2439 011206 053111 051105 041440
2440 011214 047117 051124 046117
2441 011222 051040 043505 051511
2442 011230 042524 020122 042101
2443 011236 051104 051505 026523
2444 011244 000
2445 011245 075 042504 044526
2446 011252 042503 020040 000
2447 011257 015 044012 053517
2448 011264 047040 053517 041040
2449 011272 047522 047127 041440
2450 011300 053517 020077 027056
2451 011306 051456 046105 041505
2452 011314 020124 047523 042515
2453 011322 044124 047111 020107
2454 011330 047524 051040 047125
2455 011336 040040 041501 051124
2456 011344 043505 000
2457 011347 015 047412 052125
2458 011354 047440 020106 040522
2459 011362 043516 035105 042522
2460 011370 054524 042520 046040
2461 011376 051501 020124 042504
2462 011404 044526 042503 051040
2463 011412 041530 051123 040440
2464 011420 042104 042522 051523
2465 011426 000055
2466 011430 020040 000077
2467 011434 005015 000
2468 011437 040 050040 053517
2469 011444 051105 043040 044501
2470 011452 052514 042522 020054
2471 011460 051120 043517 040522
2472 011466 020115 042522 052123
2473 011474 051101 020124 052101
2474 011502 052040 051505 020124
2475 011510 047111 050040 047522
2476 011516 051107 051505 000123
2477 011524 005015 047105 020104
2478 011532 043117 050040 051501
2479 011540 020123 040524 042520
2480 011546 043040 000
2481 011551 015 051012 000
2482 011555 015 052012 051505
2483 011562 020124 041520 000055
2484 011570 005015 047514 045503
2485 011576 047440 020116 042523
2486 011604 042514 052103 042105
2487 011612 052040 051505 037524
2488 011620 024040 020131 051117
2489 011626 047040 026451 000
2490 011633 015 042012 020125
2491 011640 051120 047511 044522
2492 011646 054524 046040 053105
2493 011654 046105 000055
2494 011660 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-/

MQM: .ASCIZ / ?/

MCRLF: .ASCIZ <15><12>

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

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

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)-/

2495 011666 051440 047131 020103
2496 011674 044103 051101 020123
2497 011702 042523 042514 052103
2498 011710 042105 024040 030440
2499 011716 047440 020122 024462
2500 011724 000055
2501 011726 005015 051511 051440
2502 011734 041505 054040 044515
2503 011742 020124 052512 050115
2504 011750 051105 021440 020066
2505 011756 047111 020077 054450
2506 011764 047440 020122 024516
2507 011772 000055
2508 011774 005015 051511 051440
2509 012002 041505 051040 041505
2510 012010 045040 046525 042520
2511 012016 020122 032443 044440
2512 012024 037516 024040 020131
2513 012032 051117 047040 026451
2514 012040 000
2515 012041 015 044412 020123
2516 012046 050117 020124 046103
2517 012054 020122 047105 041101
2518 012062 042514 045040 046525
2519 012070 042520 020122 032043
2520 012076 044440 037516 024040
2521 012104 020131 051117 047040
2522 012112 026451 000
2523 012115 015 044412 020123
2524 012122 044124 020105 042524
2525 012130 052123 041440 047117
2526 012136 042516 052103 051117
2527 012144 044440 051516 040524
2528 012152 046114 042105 037440
2529 012160 054450 047440 020122
2530 012166 024516 000055
2531 012172 006412 020040 020040
2532 012200 052123 052101 051525
2533 012206 020040 046440 050101
2534 012214 020040 020040 005040
2535 012222 000015
2536
2537
2538
2539
2540 012224 000040
2541 012264 000040
2542 012324 000040
2543
2544
2545
2546
2547
2548 012364 006367 000044
2549 012370 006367 000040
2550 012374 006367 000034

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

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

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

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

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

.EVEN

;BUFFERS FOR INPUT-OUTPUT

INBUF: .BLKB 4)
TEMP: .BLKB 4)
MDATA: .BLKB 4)

;UTILITIES

;THIS UTILITY CALCULATES PRIORITY LEVEL

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


```

2551 012400 006367 000030 ASL DUPRT ;
2552 012404 006367 000024 ASL DUPRT ;
2553 012410 016767 000020 000020 MOV DUPRT,LESS1 ;MOVE THIS TO LESS1
2554 012416 162767 000001 000012 SUB #1,LESS1 ;CREATE LESS1
2555 012424 042767 000037 000004 BIC #37,LESS1 ;CLEAR TNZVC
2556 012432 000207 RTS PC
2557 012434 000240 DUPRT: LEVEL5
2558 012436 000200 LESS1: LEVEL4 ;LEVEL TO ALLOW INTERRUPTS
2559
2560 ;NEW DU ADDRESSES
2561 012440 016767 000126 000422 DUADDR: MOV DUBASE,RXCSR ;XXX0
2562 012446 005267 000120 INC DUBASE
2563 012452 016767 000114 000412 MOV DUBASE,HRXCSR ;XXX1
2564 012460 005267 000106 INC DUBASE
2565 012464 016767 000102 000402 MOV DUBASE,RXDBUF ;XXX2
2566 012472 016767 000074 000400 MOV DUBASE,PARCSR ;XXX2
2567 012500 005267 000066 INC DUBASE
2568 012504 016767 000062 000364 MOV DUBASE,HRXDBUF ;XXX3
2569 012512 016767 000054 000362 MOV DUBASE,HPARCSR ;XXX3
2570 012520 005267 000046 INC DUBASE
2571 012524 016767 000042 000352 MOV DUBASE,TXCSR ;XXX4
2572 012532 005267 000034 INC DUBASE
2573 012536 016767 000030 000342 MOV DUBASE,HTXCSR ;XXX5
2574 012544 005267 000022 INC DUBASE
2575 012550 016767 000016 000332 MOV DUBASE,TXDBUF ;XXX6
2576 012556 005267 000010 INC DUBASE
2577 012562 016767 000004 000322 MOV DUBASE,HTXDBUF ;XXX7
2578 012570 000207 RTS PC
2579 012572 000000 DUBASE: 0
2580
2581 ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
2582 ;INFORMATION CONTAINED IN TEMP1 AND IT IS
2583 ;SHIFTED IN BY THE CONTENTS OF SHIFT
2584 012574 042777 040000 000302 RPOKE: BIC #MTDATA,@TXCSR
2585 012602 005067 166340 CLR TEMP2
2586 012606 006067 166332 ROR TEMP1 ;FORCE CARRY
2587 012612 006067 166330 ROR TEMP2 ;PICK UP CARRY IN BIT 15
2588 012616 006267 166324 ASR TEMP2 ;SHIFT INTO BIT 14
2589 012622 042767 100000 166316 BIC #BIT15,TEMP2 ;CLR BIT 15
2590 012630 056777 166312 000246 BIS TEMP2,@TXCSR ;POKE MAINT DATA
2591 012636 042777 020000 000240 BIC #CLK,@TXCSR ;POKE CLK
2592 012644 052777 020000 000232 BIS #CLK,@TXCSR ;
2593 012652 005367 166262 DEC SHIFT
2594 012656 001346 BNE RPOKE
2595 012660 000207 RTS PC
2596
2597 ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
2598 012662 016767 166256 166256 ODD8: MOV TEMP1,TEMP2 ;SAVE TEMP1
2599 012670 005067 166254 CLR TEMP3
2600 012674 012727 000010 MOV #8,(PC)+
2601 012700 000000 1$: 0
2602 012702 006067 166240 2$: ROR TEMP2
2603 012706 005567 166236 ADC TEMP3
2604 012712 005367 177762 DEC 1$
2605 012716 001371 BNE 2$
2606 012720 006067 166224 ROR TEMP3

```

```

2607 012724 103404          BCS      3$
2608 012726 052767 000400 166210      BIS      #BIT8,TEMP1      ;SET ODD PARITY
2609 012734 000400          BR       3$
2610 012736 042767 000400 166200 3$:      BIC      #BIT8,TEMP1      ;CLR EVEN PARITY
2611          :TEMP1 NOW HAS ODD PARITY CHARACTER
2612 012744 000207          4$:      RTS      PC
2613
2614          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
2615 012746 016767 166172 166172  EVEN8:  MOV     TEMP1,TEMP2      ;SAVE TEMP1
2616 012754 005067 166170          CLR     TEMP3
2617 012760 012727 000010          MOV     #8.,(PC)+
2618 012764 000000          1$:      0
2619 012766 006067 166154          2$:      ROR     TEMP2
2620 012772 005567 166152          ADC     TEMP3
2621 012776 005367 177762          DEC     1$
2622 013002 001371          BNE     2$
2623 013004 006067 166140          ROR     TEMP3
2624 013010 103004          BCC     3$
2625 013012 052767 000400 166124      BIS      #BIT8,TEMP1      ;SET EVEN PARITY
2626 013020 000403          BR       4$
2627 013022 042767 000400 166114 3$:      BIC      #BIT8,TEMP1      ;CLR ODD PARITY
2628          :TEMP1 NOW HAS EVEN PARITY CHARACTER
2629 013030 000207          4$:      RTS      PC
2630 013032 062716 000002  TRPREG: ADD   #2,(SP) ;ALLOW IT TO 'CRUNCH' INTO HLT BACK
2631          ;IN MAIN PART OF THE PROGRAM
2632 013036 000002          RTI
2633          ;ERROR HLT TABLE
2634 013040 013124      .ERRTAB:  EMO      ;HLT 0 BIT ERROR (GENERAL)
2635 013042 000000          0
2636 013044 000000          0
2637 013046 013140          EM1      ;HLT 1 REGISTER ERROR
2638 013050 013311          DH1
2639 013052 013332          DT1
2640 013054 013202          EM2      ;HLT 2 RECEIVER ERROR
2641 013056 013311          DH1
2642 013060 013332          DT1
2643 013062 013244          EM3      ;HLT 3 TRANSMITTER ERROR
2644 013064 013311          DH1
2645 013066 013332          DT1
2646          ;DEFAULT DU ADDRESSES
2647 013070 160040      RXCSR: 160040
2648 013072 160041      HRXCSR: 160041
2649 013074 160042      RXDBUF: 160042
2650 013076 160043      HRXDBUF: 160043
2651 013100 160042      PARCSR: 160042
2652 013102 160043      HPARCSR: 160043
2653 013104 160044      TXCSR: 160044
2654 013106 160045      HTXCSR: 160045
2655 013110 160046      TXDBUF: 160046
2656 013112 160047      HTXDBUF: 160047
2657          ;DEFAULT DU VECTORS
2658 013114 000770      DURIV: 770      ;REC INTR VECTOR
2659 013116 000772      DURIS: 772      ;REC INTR STATUS
2660 013120 000774      DUTIV: 774      ;XMIT INTR VECTOR
2661 013122 000776      DUTIS: 776      ;XMIT INTR STATUS
2662          ;ERROR MESSAGES

```


2663	013124	036440	042440	051122	EM0:	.ASCIZ / = ERROR PC/
2664	013132	051117	050040	000103		
2665	013140	036440	051040	043505	EM1:	.ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
2666	013146	051511	042524	020122		
2667	013154	051105	047522	020122		
2668	013162	041520	005015	051001		
2669	013170	043505	051511	042524		
2670	013176	020122	000040			
2671	013202	036440	051040	041505	EM2:	.ASCIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
2672	013210	044505	042526	020122		
2673	013216	051105	047522	020122		
2674	013224	041520	005015	051001		
2675	013232	043505	051511	042524		
2676	013240	020122	000040			
2677	013244	036440	052040	040522	EM3:	.ASCIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
2678	013252	051516	044515	052124		
2679	013260	051105	042440	051122		
2680	013266	051117	050040	006503		
2681	013274	000412	042522	044507		
2682	013302	052123	051105	020040		
2683	013310	000				
2684						:DATA HEADERS FOR ERROR MESSAGES
2685	013311	105	050130	041505	DH1:	.ASCIZ /EXPECTED ACTUAL/
2686	013316	042524	020104	040440		
2687	013324	052103	040525	000114		
2688						.EVEN
2689						:DATA TABLES FOR ERROR MESSAGES
2690	013332	000003			DT1:	3
2691	013334	006	004			.BYTE 6.4
2692	013336	001164				:REGISTER
2693	013340	006	004			.BYTE 6.4
2694	013342	001156				:EXPECTED DATA
2695	013344	006	002			.BYTE 6.2
2696	013346	001160				:ACTUAL DATA
2697		000001				.END

CROSS REFERENCE TABLE -- USER SYMBOLS

HRXCSR	013072	1074*	1240	2563*	2648#													
HRXDBJ	013076	1076*	1242	2568*	2650#													
HTXCSR	013106	1080*	1246	2573*	2654#													
HTXDBJ	013112	1082*	1248	2577*	2656#													
ICOUNT	001122	858#	2033	2037*														
INBUF	012224	1181	1185	2074	2116	2248	2252	2540#										
INIFLG	001244	919#	1019	1022*														
INSTER=	104404	979#	1189	2135	2257													
INSTR =	104403	977#	1098	1109	1119	1130	1155	1165	1179	1192	1196	1200	1204	1295				
		1308	2393															
INSTR2	007320	2085	2100#															
ISYMOD=	000000	796#																
IVBASE	001314	941#	1064	1229*														
IVKEEP	001316	942#	1061	1228*														
JMRBY	001203	889#	1057*	1207	1224	1790	1867											
KEEPAD	001206	894#	1059*	1065	1106*	1108	1152	1154	1226	1967								
KEEPIV	001214	897#	1061*	1117*	1228	1968												
LASTAD	001210	895#	1060*	1135	1146	1160	1227											
LESS1	012436	1070*	1237	2553*	2554*	2555*	2558#											
LEVEL0=	000000	766#																
LEVEL1=	000040	765#																
LEVEL2=	000100	764#																
LEVEL3=	000140	763#																
LEVEL4=	000200	762#	2558															
LEVEL5=	000240	761#	2557															
LEVEL6=	000300	760#																
LEVEL7=	000340	759#																
LIGHTS	001102	847#	1024*	1033*	1991*													
LIMITS	007462	2126	2140#															
LOBITS	007540	2112*	2144	2160#	2161													
LOCK	001120	857#	1336*	1426*	1516*	1606*	2049											
LOGICA	006660	1995#																
LOKFLG	001247	922#	1298	1299														
LOLIM	007532	2109*	2142	2157#														
LPCNT	001124	859#	1010*	2032*	2033	2036*	2366*											
LSTERR	001134	863#	1015*	1988*	2278	2280*												
L1ESS	001336	950#	1070	1237*														
MAP	001634	1037	1039	1047#														
MCOw	011257	1950	2447#															
MCRLF	011434	1277	2082	2098	2190	2399	2467#											
MDATA	012324	2216	2227	2542#														
MEPASS	011524	1943	2477#															
MEXT =	010000	819#	1798	1875														
MEXTJ	012115	1205	2523#															
MINT =	004000	818#	1341	1431	1521	1611	1696	1746										
MITSEX	001301	930#	1053	1220*														
MLASTD	011164	1131	2436#															
MLEVEL	011633	1166	2490#															
MLOCK	011570	1296	2484#															
MMULT	011105	1120	2428#															
MPFAIL	011437	2364	2468#															
MQM	011430	2095	2466#															
MR	011551	1318	2481#															
MRANGE	011347	1156	2457#															
MREGAD	011024	1099	2419#															
MRESET=	000400	809#	1337	1402	1427	1492	1517	1582	1607	1672	1690	1692	1740	1742				

MSTATJ 012172	1792	1794	1869	1871															
MSYNC 011660	1262	2531#																	
MTDATA= 040000	1180	2494#																	
MTITLE 010750	806#	2584																	
MTSTPC 011555	1021	2411#																	
MULTD 001202	1309	2482#																	
MVECTO 011002	888#	1056*	1122	1123	1223	1287	1946												
MWIRE4 012041	1110	2416#																	
MWIRE5 011774	1201	2515#																	
MWIRE6 011726	1197	2508#																	
NEXT 001116	1193	2501#																	
NOPAR = 000000	856#	1335*	1425*	1515*	1605*	1689*	1739*	1789*	1866*	2018	2038	2324							
NOSYNC 001300	801#	1340	1430	1520	1610														
ODDPAR= 001000	929#	1052	1219*																
ODD8 012662	802#	1749	1878																
ONCE 001522	2598#																		
OPTCLR 001201	1020	1023#																	
OUT 010706	887#	1055*	1203	1222															
OUTCRY 006620	2380	2382	2384	2388	2400#														
OUTMUL 002476	1944	1983#																	
OVRRUN= 040000	1128	1153	1164#																
PARAM = 104405	787#																		
PARAM1 007366	981#	1100	1111	1132	1157	1167	1310	2394											
PARCSR 013100	2115#	2136																	
	1077*	1243	1342*	1357*	1432*	1447*	1522*	1537*	1612*	1627*	1691*	1699*	1741*						
	1749*	1793*	1801*	1870*	1878*	2566*	2651#												
PAREN = 001000	791#																		
PARER = 010000	789#																		
PARERR 007442	2118	2120	2122	2131#	2141	2143	2145												
PARTI 007524	2134	2154#																	
PASCNT 001130	861#	1012*	1990*	1991															
PFTAB 010564	2362	2368#																	
POPPO = 012600	735#	2317																	
POP1SP= 005726	733#																		
POP2SP= 022626	737#																		
PRTDU 001324	945#	1071	1232*																
PS = 177776	727#	1007*	1212*	1999*	2356*														
PUSHRO= 010046	734#	2314																	
PUSH1S= 005746	732#																		
PUSH2S= 024646	736#																		
RDSW 010572	2096	2133	2376#	2390*	2400*														
REACT= 004000	773#	1352	1376	1442	1466	1532	1556	1622	1646										
REGACT 001320	943#	1062	1230*																
REPLAY 006524	1963	1969	1971#																
RESEC 001302	931#	1054	1221*																
RESREG 010332	2308	2311#																	
RESTAR 010466	2342	2348#																	
RESTR 006670	1964	1993	1999#																
RES05 = 104407	985#	2311																	
RING = 040000	770#																		
RINTEN= 000100	778#																		
RISDU 001334	949#	1067	1234*																
RIVDU 001326	946#	1066	1233*																
ROTADD 001220	901#	1063*	1127*	1139*	1141	1143*	1148	1151*	1231	1958*	1961	1965*							
RPOKE 012574	2584#	2594																	
RTRN 001114	855#	1017*	1313	1317*	1319	2001*	2018*	2019	2038*	2039	2324*	2326	2367						

\$WORDF 677#
\$WORDO 677#
\$WORDP 677#

. ABS. 013350 000

ERRORS DETECTED: 0

CZDUFD.BIN,CZDUFD.SEQ/CRF/SOL/NL:TOC=CZDU11.HLO/EQ:RUNF,CZDU11.PAR,CZDU11.KET,CZDUFD.P11
RUN-TIME: 69.9 SECONDS
RUN-TIME RATIO: 103/16=6.1
CORE USED: 18K (35 PAGES)