

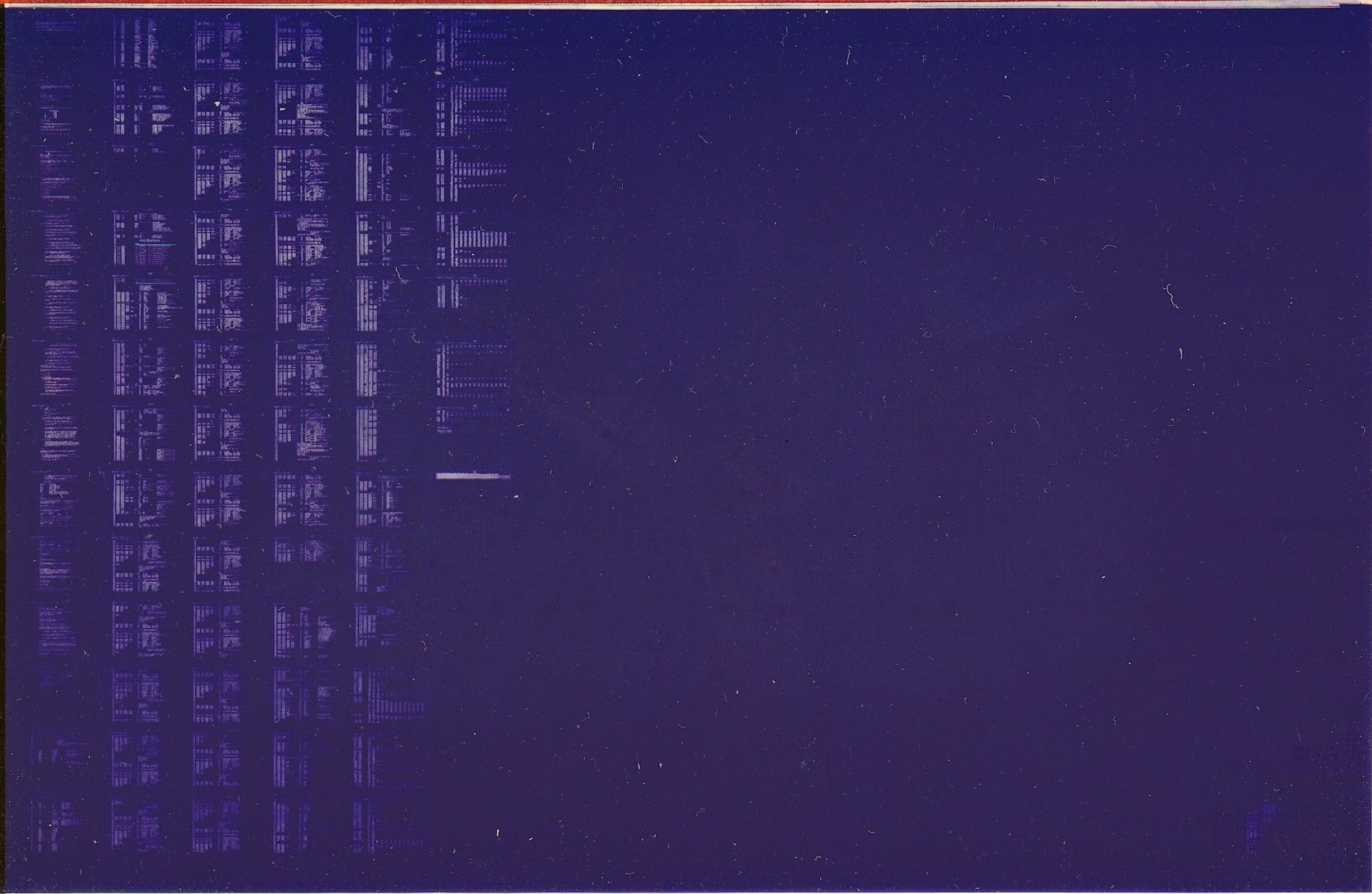
DU11

RECEIVER TIMING TESTS
MD-11-DZDUC-B

EP-DZDUC-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976

MADE IN USA



B01

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 1
HELLO.P11 03-AUG-76 00:00

SEQ 0001

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, 1976 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.

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

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

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 DOWN4.1.2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4.1.3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART

- (ONLY IN SINGLE DEVICE TESTS)
- SW01=1
- 4.1.4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
- SW02=1
- NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=1 IS USED
- NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1
- 4.2 STARTING ADDRESS
- THE STARTING ADDRESS FOR ALL TESTS IS 000200
- THE RETARTING ADDRESS FOR ALL TESTS IS 000200
- THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200
- THE STARTING ADDRESS TO LOCK ON TEST IS 000200
- 4.3 PROGRAM AND/OR OPERATOR ACTION
- 4.3.1 INITIAL PROGRAM START
- 4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER
- 4.3.1.2 LOAD ADDRESS 000200
- 4.3.1.3 CLEAR CONSOLE SWITCHES
- 4.3.1.4 PRESS START
- 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 2 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

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
SW02 =1	LOCK ON TEST
SW01 =1	RESTART PROGRAM AT SELECTED TEST
SW00 =1	RESELECT VECTOR AND CONTROL REGISTER ADDRESSES & PARAMETERS AFTER A PROGRAM RESTART

TO INHIBIT "END OF PASS" TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS THERE ARE FOUR DISTINCT ERROR TYPEOUTS

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

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

6.1.2 PC +2 = REGISTER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER
WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER
WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.3 PC +2 = RECEIVER ERROR PC

REGISTER	EXPECTED	ACTUAL
16XXXX	YYYYYY	ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

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

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6.1.5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6.2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7. RESTRICTIONS

7.1 MULTIPLE DEVICES

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

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

7.2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

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

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

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

7.2.2.2ORLOAD 000200 WITH SW00 =1 AND DEPRESS START....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
.....THE PROGRAM WILL CONTINUE WITH DEVICE 0

7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM
WILL TYPEOUT AN ERROR MESSAGE.....LOAD & START AT 000200

7.3 CABLE DELAYS

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

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

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

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

8. DEFAULT PARAMETERS:

1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160040

VECTOR ADDRESS-

DURIV: 770

L01

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 11
HELLO.P11 03-AUG-76 00:00

SEQ 0011

ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0
LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0
DU PRIORITY LEVEL- LEVEL 5 DUPRT: LEVEL 5
OF SYNC CHARS SELECTED - 2 SYNCNO: 377
IS SEC XMIT JUMPER # 6 IN ?- YES SEXMIT: 377
IS SEC REC JUMPER # 5 IN ?- YES 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

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

```

552 .ENABLE ABS
553
554 ;DU11 DZDUC-B TAPE C
555 ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
556
557 ;STARTING PROCEDURE
558 ;LOAD PROGRAM
559 ;PRESS START
560 ;PROGRAM WILL TYPE "DU11 DZDUC-B TAPE C "
561 ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
562 ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE C"
563 ;AND THEN RESUME TESTING
564
565
566 ;SWITCH REGISTER OPTIONS
567
568 100000 SW15=100000 ;=1,HALT ON ERROR
569 040000 SW14=40000 ;=1,LOOP ON CURRENT TEST
570 020000 SW13=20000 ;=1,INHIBIT ERROR TYPEOUT
571 010000 SW12=10000
572 004000 SW11=4000 ;=1,INHIBIT ITERATIONS
573 002000 SW10=2000 ;=1,ESCAPE TO NEXT TEST ON ERROR
574 001000 SW09=1000 ;=1,LOOP WITH CURRENT DATA
575 000400 SW08=400 ;=1,LOOP ON ERROR
576 000100 SW06=100
577 000040 SW05=40
578 000020 SW04=20
579 000010 SW03=10
580 000004 SW02=4 ;LOCK ON TEST SELECT
581 000002 SW01=2 ;RESTART PROGRAM AT SELECTED TEST
582 000001 SW00=1 ;RESELECT VECTOR AND CONTROL REGISTER
583 ;ADDRESS AFTER PROGRAM RESTART
584

```



```

585
586
587
588      000000      RO=%0      ;GENERAL REGISTER
589      000001      R1=%1      ;GENERAL REGISTER
590      000002      R2=%2      ;GENERAL REGISTER
591      000003      R3=%3      ;GENERAL REGISTER
592      000004      R4=%4      ;GENERAL REGISTER
593      000005      R5=%5      ;GENERAL REGISTER
594      000006      SP=%6      ;PROCESSOR STACK POINTER
595      000007      PC=%7      ;PROGRAM COUNTER
596
597
598
599      177570      DSWR=177570 ;HARDWARE SWITCH REGISTER LOC.
600      177570      DLIGHTS=177570 ;HARDWARE DISPLAY REGISTER LOC.
601      177776      PS=177776 ;PROCESSOR STATUS WORD
602      001100      STACK=1100 ;START OF PROCESSOR STACK
603
604
605
606      005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
607      005726      POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
608      010046      PUSHRO=10046 ;SAVE RO ON STACK =MOV RO, -(SP)
609      012600      POPRO=12600 ;RESTORE RO FROM STACK =MOV (SP)+,RO
610      024646      PUSH2SP=24646 ;DECREMENT STACK TWICE =CMP -(SP),-(SP)
611      022626      POP2SP=22626 ;INCREMENT STACK TWICE =CMP (SP)+,(SP)+
612      .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
613
614
615      100000      BIT15=100000
616      040000      BIT14=40000
617      020000      BIT13=20000
618      010000      BIT12=10000
619      004000      BIT11=4000
620      002000      BIT10=2000
621      001000      BIT9=1000
622      000400      BIT8=400
623      000200      BIT7=200
624      000100      BIT6=100
625      000040      BIT5=40
626      000020      BIT4=20
627      000010      BIT3=10
628      000004      BIT2=4
629      000002      BIT1=2
630      000001      BIT0=1
631
632
633
634
635
636
637
638
639
640

```

;PROCESSOR LEVELS

```

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

```

```

641      ;REGISTER DEFINITIONS
642      ;RXCSR BIT DEFINITIONS
643      100000 DSC=BIT15      ;DATA SET CHANGE
644      040000 RING=BIT14     ;RING
645      020000 CTS=BIT13      ;CLR TO SEND
646      010000 CARDET=BIT12   ;CARRIER DETECT
647      004000 RECACT=BIT11   ;REC ACTIVE
648      002000 SRD=BIT10      ;SEC REC DATA
649      001000 DSR=BIT9       ;DATA SET RDY
650      000400 STPSYN=BIT8    ;STRIP SYNC
651      000200 RXDONE=BIT7    ;REC DONE
652      000100 RINTEN=BIT6    ;REC INTR ENABLE
653      000040 DSINTE=BIT5    ;DSC INTR ENABLE
654      000020 SYNSCH=BIT4    ;SYNC SEARCH
655      000010 STD=BIT3       ;SEC XMIT DATA
656      000004 RTS=BIT2       ;REQ TO SEND
657      000002 DTR=BIT1       ;DATA TERM RDY
658      000001 VOID=BIT0
659
660      ;RXDBUF BIT DEFINITIONS
661      100000 RXERR=BIT15     ;REC ERROR
662      040000 OVRUN=BIT14    ;OVERRUN
663      020000 FRMERR=BIT13   ;FRAME ERROR
664      010000 PARER=BIT12    ;PARITY ERROR
665
666      ;PARCSR BIT DEFINITIONS
667      001000 PAREN=BIT9      ;PARITY ENABLE
668      000400 EVPAR=BIT8     ;EVEN PARITY SENSE
669
670      ;PARCSR WRD DEFINITIONS
671      030000 SYNINT=30000    ;SYNC EXTERNAL MODE
672      020000 SYNEXT=20000   ;SYNC INTERNAL MODE
673      000000 ISYMOD=0       ;ISOC MODE
674      000000 FIVE=0         ;WORD LENGTH 5 BITS
675      002000 SIX=2000       ;WORD LENGTH 6 BITS
676      004000 SEVEN=4000    ;WORD LENGTH 7 BITS
677      006000 EIGHT=6000   ;WORD LENGTH 8 BITS
678      000000 NOPAR=0       ;NO PARITY
679      001000 ODDPAR=1000   ;ODD PARITY
680      001400 EVEPAR=1400   ;EVEN PARITY
681
682      ;TXCSR BIT DEFINITIONS
683      100000 DNA=BIT15       ;DATA NOT AVAILABLE
684      040000 MTDATA=BIT14   ;MAINT DATA
685      020000 CLK=BIT13      ;CLK
686      002000 BITW=BIT10     ;BIT WINDOW
687      000400 MRESET=BIT8    ;MASTER RESET
688      000200 TXDONE=BIT7    ;XMIT DONE
689      000100 TXINTE=BIT6    ;XMIT INTR ENABLE
690      000040 DNAINTE=BIT5   ;DNA INTR ENAB
691      000020 SEND=BIT4      ;SEND
692      000010 HDXEN=BIT3     ;HDX/FDX
693      000001 BREAK=BIT0     ;BREAK
694
695      ;TXCSR WRD DEFINITIONS
696      000000 USER=0         ;USER MODE
697      004000 MINT=4000      ;MAINT INT MODE
698      010000 MEXT=10000    ;MAINT EXT MODE
699      014000 SYSTST=14000  ;SYSTEM TEST MODE
700
701      ;TRAPCATCHER FOR ILLEGAL INTERRUPTS

```



```

696                                     ;STANDARD INTERRUPT VECTORS
697
698
699                                     . =24
700 000024 014566                       .PFAIL                       ;POWER FAIL HANDLER
701 000026 000340                       340                          ;SERVICE AT LEVEL 7
702 000030 014316                       .HLT                          ;ERROR HANDLER
703 000032 000340                       340                          ;SERVICE AT LEVEL 7
704 000034 014264                       .TRPSRV                       ;GENERAL HANDLER DISPATCH SERVICE
705 000036 000340                       340                          ;SERVICE AT LEVEL 7
706
707                                     ;SOFTWARE SWITCH REGISTER
708
709                                     . =174
710 000174 000000                       DISPREG: .WORD 0              ;SOFTWARE DISPLAY REG.
711 000176 000000                       SWREG:   .WORD 0              ;SOFTWARE SWITCH REGISTER
712 000200 000167 001054                 JMP      .START                ;GO TO START OF PROGRAM
713
714
715                                     . =1100
716 001100                                ;INDIRECT POINTERS
717
718
719
720 001100 177570                       SWR:      177570                ;SWITCH REGISTER POINTER
721 001102 177570                       LIGHTS: 177570                ;DISPLAY REGISTER POINTER
722 001104 177560                       TKCSR:   177560                ;TELETYPE KEYBOARD CONTROL REGISTER
723 001106 177562                       TKDBR:   177562                ;TELETYPE KEYBOARD DATA BUFFER
724 001110 177564                       TPCSR:   177564                ;TELEPRINTER CONTROL REGISTER
725 001112 177566                       TPDBR:   177566                ;TELEPRINTER DATA BUFFER
726
727                                     ;PROGRAM CONTROL PARAMETERS
728
729 001114 000000                       RTRN:    0                      ;SCOPE ADDRESS FOR LOOP ON TEST
730 001116 000000                       NEXT:    0                      ;ADDRESS OF NEXT TEST TO BE EXECUTED
731 001120 000000                       LOCK:    0                      ;ADDRESS FOR LOCK ON CURRENT DATA
732 001122 000000                       ICOUNT:  0                      ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
733 001124 000000                       LPCNT:   0                      ;NUMBER OF ITERATIONS COMPLETED
734 001126 000000                       TSTNO:   0                      ;NUMBER OF TEST IN PROGRESS
735 001130 000000                       PASCNT:  0                      ;NUMBER OF PASSES COMPLETED
736 001132 000000                       ERRCNT:  0                      ;TOTAL NUMBER OF ERRORS
737 001134 000000                       LSTERR:  0                      ;PC OF LAST ERROR CALL
738
739                                     ;PROGRAM VARIABLES
740
741 001136 000020                       HOLD:    20                     ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
742 001140 000000                       SHIFT:   0                      ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
743 001142 000000                       COUNT:   0                      ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
744 001144 000000                       TEMP1:   0                      ;TEMPORARY STORAGE
745 001146 000000                       TEMP2:   0                      ;TEMPORARY STORAGE
746 001150 000000                       TEMP3:   0                      ;TEMPORARY STORAGE
747 001152 000000                       TEMP4:   0                      ;TEMPORARY STORAGE
748 001154 000000                       TEMP5:   0                      ;TEMPORARY STORAGE
749 001156 000000                       SAVR0:   0                      ;R0 STORAGE
750 001160 000000                       SAVR1:   0                      ;R1 STORAGE
751 001162 000000                       SAVR2:   0                      ;R2 STORAGE
    
```

752 001164 000000
753 001166 000000
754 001170 000000
755 001172 000000
756 001174 000000

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

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


```

757 ;PROGRAM CONVERSATIONAL PARAMETERS
758 001176 377 SYNCNO: .BYTE 377 ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
759 001177 377 SEXMIT: .BYTE 377 ;SEC XMIT JUMPER "IN"
760 001200 377 SEREC: .BYTE 377 ;SEC REC JUMPER "IN"
761 001201 377 OPTCLR: .BYTE 377 ;OPTIONAL JUMPER CLR "IN"
762 001202 000 MULTD: .BYTE 0 ;NO MULTIPLE DEVICE FLAG
763 001203 377 JMRBY: .BYTE 377 ;EXTERNAL MODEM BYPASS JUMPER "IN"
764 .EVEN
765
766 ;PROGRAM MULTIPLE DEVICE PARAMETERS
767 001204 000000 BASEADD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
768 001206 000000 KEEPADD: 0 ;SAVED 1ST DEVICE ADDR
769 001210 000000 LASTADD: 0 ;LAST DEVICE RXCSR ADDR
770 001212 000000 BASEIV: 0 ;PROG CONTROLLED IV
771 001214 000000 KEEPIV: 0 ;SAVED INTR VECTOR
772 001216 000000 ACTREG: 0 ;ACTIVE REGISTER , MODIFY THIS
773 ;LOCATION TO DISQUALIFY OR QUALIFY
774 ;DEVICES (1= RUN , 0= DON'T RUN)
775 001220 000000 ROTADD: 0 ;ROTATING POINTER FOR ACTREG..POINTS
776 ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DE
777
778 ;PROGRAM CONTROL FLAGS
779
780 001222 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
781 001223 000 STFLG: .BYTE 0 ;TEST START FLAG
782 001224 000 ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
783 001225 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
784
785 ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
786 ;POINTERS TO SUBROUTINES CAN BE FOUND
787 ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
788
789 001226 .TRPTAB:
790 ;*****
791 ;*****
792 104400 .SCOPE SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
793 001226 013050 .SCOPE SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
794 104401 .SCOPE1 TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
795 001230 013234 .TYPE INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
796 104402 .INSTR INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
797 001232 013254 .INSTER PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
798 104403 .PARAM SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
799 001234 013314 .SAVOS RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
800 104404 .RESOS CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
801 001236 013432 .CONVRT CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
802 104405 .CNVRT SETFLG=TRAP+12 ;CALL TO FLAG SET ROUTINE
803 001240 013464
804 104406
805 001242 013700
806 104407
807 001244 013740
808 104410
809 001246 013772
810 104411
811 001250 013776
812 104412

```

```

813 001252 014216 .SETFLG
814 104413
815 001254 014732 .CKSWR CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
816 104414 .CNTLU CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
817 001256 015006
818 ;*****
819 ;*****
820
821 ;PROGRAM INITIALIZATION
822 ;LOCK OUT INTERRUPTS
823 ;SET UP PROCESSOR STACK
824 ;SET UP POWER FAIL VECTOR
825 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
826 ;TYPE TITLE MESSAGE
827
828 001260 012767 000340 176510 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
829 001266 012706 001100 MOV #STACK,SP ;SET UP STACK
830 001272 012737 014566 000024 MOV #PFAIL,2#24 ;SET UP POWER FAIL VECTOR
831 001300 005067 177620 CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
832 001304 105067 177713 CLR STFLG ;CLEAR START FLAG
833 001310 005067 177614 CLR PASCNT ;CLEAR PASS COUNT
834 001314 105067 177704 CLRB ERRFLG ;CLEAR ERROR FLAG
835 001320 005067 177606 CLR ERRCNT ;CLEAR ERROR COUNT
836 001324 005067 177604 CLR LSTERR ;CLEAR LAST ERROR POINTER
837 001330 012767 000001 177570 MOV #1,TSTNO ;SET UP FOR TEST 1
838 001336 012767 001260 177550 MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
839 ;TESTING STARTS
840 001344 105767 177652 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
841 001350 001004 BNE ONCE
842 001352 104402 015106 TYPE #MTITLE ;TYPE TITLE MESSAGE
843 001356 105167 177640 COMB INIFLG ;IF NOT SET FLAG AND DO
844 001362 012767 177570 177510 ONCE: MOV #DSWR,SWR ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
845 001370 012767 177570 177504 MOV #DLIGHTS,LIGHTS ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
846 001376 013746 000006 MOV 2#6,-(SP) ;SAVE VECTORS
847 001402 013746 000004 MOV 2#4,-(SP)
848 001406 012737 001426 000004 MOV #64$,2#4 ;SET UP FOR TIMEOUT
849 001414 022777 177777 177456 CMP #-1,2#SWR ;REFERENCE HARDWARE SWITCH REGISTER
850 001422 001402 BEQ 65$
851 001424 000407 BR 66$
852 001426 022626 64$: CMP (SP)+,(SP)+ ;ADJUST STACK
853 001430 012767 000176 177442 65$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
854 001436 012767 000174 177436 MOV #DISPREG,LIGHTS ;POINT TO SOFT DISPLAY REG
855 001444 012637 000004 66$: MOV (SP)+,2#4 ;RESTORE VECTORS
856 001450 012637 000006 MOV (SP)+,2#6
857 001454 005737 000042 TST 2#42 ;UNDER MONITOR
858 001460 001005 BNE 67$
859 001462 022767 000176 177410 CMP #SWREG,SWR ;IS SWREG USED
860 001470 001001 BNE 67$
861 001472 104414 CNTLU
862 001474 032777 000001 177376 67$: BIT #SW00,2#SWR ;RESELECT VECTOR & CONTROL REG?
863 001502 001002 BNE 1$
864 001504 000167 000446 JMP .BEGIN
865 001510 012700 000300 1$: MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
866 001514 012701 000302 MOV #302,R1 ;START AT LOCATION 300
867 001520 012702 000004 MOV #4,R2
868 001524 010110 2$: MOV R1,(R0)
    
```


869	001526	005011			CLR	(R1)	
870	001530	060200			ADD	R2,R0	
871	001532	060201			ADD	R2,R1	
872	001534	022701	001000		CMP	#1000,R1	;END AT LOCATION 776
873	001540	002771			BLT	2\$	
874	001542	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
875	001544	015162			MREGAD		;MESSAGE
876	001546	104405			PARAM		;CONVERT STRING
877	001550	160000			160000		;LOW LIMIT
878	001552	167776			167776		;HIGH LIMIT
879	001554	017016			DUBASE		;STORE AT THIS LOCATION
880	001556	001			.BYTE	1	;MASK
881	001557	001			.BYTE	1	;HOW MANY TIMES + 2
882	001550	016767	015232	177420	MOV	DUBASE,KEEPADD	;SAVE
883	001566	004767	015072		JSR	PC,DUADDR	
884	001572	016767	177410	177404	MOV	KEEPADD,BASEADD	;RESTORE FOR ROTATION
885	001600	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
886	001602	015140			MVECTO		;MESSAGE
887	001604	104405			PARAM		;CONVERT STRING
888	001606	000300			300		;LOW LIMIT
889	001610	000776			776		;HIGH LIMIT
890	001612	017340			DURIV		;STORE AT THIS LOCATION
891	001614	001			.BYTE	1	;MASK
892	001615	004			.BYTE	4	;HOW MANY TIMES + 2
893	001616	016767	015516	177370	MOV	DURIV,KEEPIV	;SAVE
894	001624	016767	015510	177360	MOV	DURIV,BASEIV	;SET UP FOR ROTATION
895	001632	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
896	001634	015243			MMULT		;MESSAGE
897	001636	104412			SETFLG		;SET FLAG BASED UPON INPUT STRING
898	001640	001202			MULTD		;THIS FLAG
899	001642	105767	177334		TSTB	MULTD	;ARE THERE MULTIPLE DEVICES
900							;ON THE SYSTEM ?
901	001646	100406			BMI	BBB	;YES,ASK NEXT QUESTION
902	001650	005067	177342		CLR	ACTREG	
903	001654	005067	177340		CLR	ROTADD	
904	001660	000167	000140		JMP	OUTMUL	;JUMP AROUND NEXT QUESTION
905	001664				BBB:		
906	001664	104403			INSTR		;OUTPUT MESSAGE & GET INPUT STRING
907	001666	015322			MLASTD		;MESSAGE
908	001670	104405			PARAM		;CONVERT STRING
909	001672	160000			160000		;LOW LIMIT
910	001674	167776			167776		;HIGH LIMIT
911	001676	001210			LASTADD		;STORE AT THIS LOCATION
912	001700	001			.BYTE	1	;MASK
913	001701	001			.BYTE	1	;HOW MANY TIMES + 2
914					1\$:	THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME	
915	001702	012767	000001	177310	MOV	#1,ROTADD	;SET UP POINTER
916	001710	005067	177302		CLR	ACTREG	;CLR ACTIVE REGISTER
917	001714	056767	177300	177274	BIS	ROTADD,ACTREG	;MAKE THIS DEVICE ACTIVE
918	001722	000241			CLC		
919	001724	006167	177270		ROL	ROTADD	;SET UP POINTER
920	001730	103421			BCS	3\$;ARE YOU OUT OF RANGE ?
921	001732	062767	000010	177244	ADD	#10,BASEADD	;SET UP BASE ADDRESS
922	001740	026767	177244	177236	CMP	LASTADD,BASEADD	;IS THIS THE LAST DEVICE ?
923	001746	101362			BHI	2\$;NO DO IT AGAIN
924	001750	056767	177244	177240	BIS	ROTADD,ACTREG	;THIS ASSUMES THAT THERE ARE AT

```

925                                     ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
926                                     ;MULTIPLE DEVICE QUESTION
927 001756 012767 000001 177234 4$: MOV #1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
928 001764 016767 177216 177212 MOV KEEPADD,BASEADD ;DITTO
929 001772 000414 BR OUTMUL ;CONTINUE QUESTIONS
930 001774 016767 177206 177202 3$: MOV KEEPADD,BASEADD ;RESTORE
931 002002 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
932 002004 015505 MRANGE ;MESSAGE
933 002006 104405 PARAM ;CONVERT STRING
934 002010 160000 160000 ;LOW LIMIT
935 002012 167776 167776 ;HIGH LIMIT
936 002014 001210 LASTADD ;STORE AT THIS LOCATION
937 002016 001 .BYTE 1 ;MASK
938 002017 001 .BYTE 1 ;HOW MANY TIMES + 2
939 002020 000167 177656 JMP 1$ ;DO IT AGAIN
940 002024 OUTMUL:
941 002024 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
942 002026 015771 MLEVEL ;MESSAGE
943 002030 104405 PARAM ;CONVERT STRING
944 002032 000004 4 ;LOW LIMIT
945 002034 000007 7 ;HIGH LIMIT
946 002036 016660 DUPRT ;STORE AT THIS LOCATION
947 002040 000 .BYTE 0 ;MASK
948 002041 001 .BYTE 1 ;HOW MANY TIMES + 2
949 002042 004767 014542 JSR PC,DULEV
950                                     ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
951                                     ;BUFFER TO THE CHARACTERS "1" AND "2".
952                                     ;IF THE CHARACTER IS "1" CLEAR THE FLAG
953                                     ;IF THE CHARACTER IS "2" SET THE FLAG
954 002046 AAA:
955 002046 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
956 002050 016016 MSYNC ;MESSAGE
957 002052 122767 000061 014370 3$: CMPB #'1,INBUF ;IS IT "1" ?
958 002060 001003 BNE 1$
959 002062 105067 177110 CLRB SYNCNO ;000
960 002066 000412 BR 4$
961 002070 122767 000062 014352 1$: CMPB #'2,INBUF ;IS IT "2" ?
962 002076 001004 BNE 2$
963 002100 112767 177777 177070 MOVB #-1,SYNCNO ;377
964 002106 000402 BR 4$
965 002110 104404 2$: INSTR ;RETRY
966 002112 000757 BR 3$
967 002114 000240 4$: NOP
968 002116 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
969 002120 016064 MWIRE6 ;MESSAGE
970 002122 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
971 002124 001177 SEXMIT ;THIS FLAG
972 002126 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
973 002130 016132 MWIRES ;MESSAGE
974 002132 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
975 002134 001200 SEREC ;THIS FLAG
976 002136 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
977 002140 016177 MWIRE4 ;MESSAGE
978 002142 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
979 002144 001201 OPTCLR ;THIS FLAG
980 002146 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
    
```



```

981 002150 016253 MEXTJ ;MESSAGE
982 002152 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
983 002154 001203 JMRBY ;THIS FLAG
984
985 ;TEST START AND RESTART
986
987 002156 012767 000340 175612 .BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
988 002164 012706 001100 MOV #STACK,SP ;SET UP STACK
989 002170 005737 000042 TST @#42 ;IS PROGRAM UNDER MONITOR CONTROL
990 002174 001056 BNE 3$
991 002176 105767 177000 TSTB MULTD ;DON'T ALLOW LOCK ON TEST IF RUNNING
992 ;MULTIPLE DEVICES
993 002202 001407 BEQ 5$ ;IF NO TEST FOR LOCK ON TEST
994 002204 016767 011020 010720 MOV BRW,TTST ;RESTORE NORMAL SCOPE LOOP
995 002212 016767 011014 010714 MOV BRX,TTST+2 ;DITTO
996 002220 000444 BR 3$ ;JUMP AROUND IF YES
997 002222 032777 000004 176650 5$: BIT #BIT2,@SWR ;CHECK FOR LOCK ON TEST
998 002230 001416 BEQ 1$
999 002232 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1000 002234 015726 MLOCK ;MESSAGE
1001 002236 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1002 002240 001225 LOKFLG ;THIS FLAG
1003 002242 105767 176757 TSTB LOKFLG ;IS LOCK ON TEST OPTION SELECTED
1004 002246 001407 BEQ 1$
1005 002250 012767 000240 010654 MOV #NOP,TTST
1006 002256 012767 000240 010650 MOV #NOP,TTST+2 ;SET UP TO LOCK
1007 002264 000406 BR 2$
1008 002266 016767 010736 010636 1$: MOV BRW,TTST
1009 002274 016767 010732 010632 MOV BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1010 002302 032777 000002 176570 2$: BIT #SW01,@SWR ;IF SW01=1, GET STARTING PC
1011 002310 001410 BEQ 3$
1012 002312 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1013 002314 015713 MTSTPC ;MESSAGE
1014 002316 104405 PARAM ;CONVERT STRING
1015 002320 002350 TST1 ;LOW LIMIT
1016 002322 012116 TLAST ;HIGH LIMIT
1017 002324 001114 RTRN ;STORE AT THIS LOCATION
1018 002326 001 .BYTE 1 ;MASK
1019 002327 001 .BYTE 1 ;HOW MANY TIMES + 2
1020 002330 000403 BR 4$
1021 002332 012767 002350 176554 3$: MOV #TST1,RTRN ;START AT TEST 1
1022 002340 104402 015707 4$: TYPE MR ;TYPE R
1023 002344 000177 176544 JMP @RTRN ;START TESTING
1024
1025 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1026 ;: SECTION , IT USES THE ERROR FLAGS TO DETERMINE
1027 ;: THAT IT WAS SELECTED PROPERLY
1028 ;: FRAME ERROR (FRMERR,RXERR)
1029 ;: MODE: ISOC (ISYMOD)
1030 ;: LENGTH: FIVE
1031 ;: CHAR: 25
1032
1033 002350 012767 000001 176550 TST1: MOV #1,TSTNO ;SAVE THIS
1034 002356 012767 002524 176532 MOV #TST2,NEXT ;GO TO THIS TEST WHEN THRU
1035 002364 052777 000400 014736 BIS #MRESET,@TXCSR ;MASTER RESET
1036 002372 012777 000000 014724 MOV #ISYMOD,@PARCSR ;SET THE MODE

```

```

1037 002400 052777 000400 014722      BIS      #MRESET,@TXCSR ;MASTER RESET
1038
1039                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1040 002406 012777 064001 014714      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1041
1042                                     ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1043 002414 012777 000000 014702      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
1044 002422 052777 000020 014664      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1045                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1046 002430 042777 020000 014672      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1047 002436 052777 020000 014664      BIS      #CLK,@TXCSR ;POKE CLK UP
1048
1049                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1050 002444 042777 020000 014656      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1051 002452 052777 020000 014650      BIS      #CLK,@TXCSR ;POKE CLK UP
1052 002460 012767 000007 176452      MOV      #7,SHIFT ;# OF SHIFTS
1053 002466 012767 000052 176450      MOV      #52,TEMP1 ;DATA CHAR
1054                                     ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1054 002474 004767 014320      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1055 002500 016703 014614      MOV      RXDBUF,R3 ;FOR ERROR MESSAGE
1056 002504 012700 120025      MOV      #RXERR!FRMERR!25,R0 ;EXPECTED
1057 002510 017701 014604      MOV      @RXDBUF,R1 ;ACTUAL
1058 002514 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
1059 002516 001401      BEQ      64$
1060 002520 104002      HLT      2 ;FRAME ERROR & RX ERROR SHOULD BE SET
1061                                     ;IF LOWER BYTE DOES NOT MATCH IT
1062                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1063 002522                                     64$:
1064 002522 104400      SCOPE
1065                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1066                                     ;; SECTION , IT USES THE ERROR FLAGS TO DETERMINE
1067                                     ;; THAT IT WAS SELECTED PROPERLY
1068                                     ;; FRAME ERROR (FRMERR,RXERR)
1069                                     ;; MODE:ISOC (ISYMOD)
1070                                     ;; LENGTH:SIX
1071                                     ;; CHAR: 25
1072
1073 002524 012767 000002 176374      TST2:  MOV      #2,TSTNO ;SAVE THIS
1074 002532 012767 002700 176356      MOV      #TST, NEXT ;GO TO THIS TEST WHEN THRU
1075 002540 052777 000400 014562      BIS      #MRESET,@TXCSR ;MASTER RESET
1076 002546 012777 000000 014550      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1077 002554 052777 000400 014546      BIS      #MRESET,@TXCSR ;MASTER RESET
1078
1079                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1080 002562 012777 064001 014540      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1081
1082                                     ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1083 002570 012777 002000 014526      MOV      #ISYMOD!SIX!NOPAR!0,@PARCSR
1084 002576 052777 000020 014510      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1085                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1086 002604 042777 020000 014516      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1087 002612 052777 020000 014510      BIS      #CLK,@TXCSR ;POKE CLK UP
1088
1089                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1090 002620 042777 020000 014502      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1091 002626 052777 020000 014474      BIS      #CLK,@TXCSR ;POKE CLK UP
1092 002634 012767 000010 176276      MOV      #8,SHIFT ;# OF SHIFTS
1092 002642 012767 000052 176274      MOV      #52,TEMP1 ;DATA CHAR

```


K02

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 26
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0023

1093					
1094	002650	004767	014144		;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1095	002654	016703	014440		JSR PC,RPOKE ;SHIFT IN THIS CHAR
1096	002660	012700	120025		MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1097	002664	017701	014430		MOV #RXERR!FRMERR!25,R0 ;EXPECTED
1098	002670	020001			MOV @RXDBUF,R1 ;ACTUAL
1099	002672	001401			CMP R0,R1 ;COMPARE EXP VS ACT
1100	002674	104002			BEQ 64\$
1101					HLT 2 ;FRAME ERROR & RX ERROR SHOULD BE SET
1102					;IF LOWER BYTE DOES NOT MATCH IT
1103	002676				;PROBABLY IS A LENGTH SELECT PROBLEM
1104	002676	104400			64\$:
1105					SCOPE
1106					;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1107					;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1108					;;THAT IT WAS SELECTED PROPERLY
1109					;;FRAME ERROR (FRMERR,RXERR)
1110					;;MODE:ISOC (ISYMOD)
1111					;;LENGTH:SEVEN
1112					;;CHAR: 125
1113	002700	012767	000003	176220	TST3: MOV #3,TSTNO ;SAVE THIS
1114	002706	012767	003054	176202	MOV #TST4,NEXT ;GO TO THIS TEST WHEN THRU
1115	002714	052777	000400	014406	BIS #MRESET,@TXCSR ;MASTER RESET
1116	002722	012777	000000	014374	MOV #ISYMOD,@PARCSR ;SET THE MODE
1117	002730	052777	000400	014372	BIS #MRESET,@TXCSR ;MASTER RESET
1118					
1119					;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1120	002736	012777	064001	014364	MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1121					
1122					;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1123	002744	012777	004000	014352	MOV #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1124	002752	052777	000020	014334	BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1125					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1126	002760	042777	020000	014342	BIC #CLK,@TXCSR ;POKE CLK DOWN
1127	002766	052777	020000	014334	BIS #CLK,@TXCSR ;POKE CLK UP
1128					;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1129	002774	042777	020000	014326	BIC #CLK,@TXCSR ;POKE CLK DOWN
1130	003002	052777	020000	014320	BIS #CLK,@TXCSR ;POKE CLK UP
1131	003010	012767	000011	176122	MOV #9,SHIFT ;# OF SHIFTS
1132	003016	012767	000252	176120	MOV #252,TEMP1 ;DATA CHAR
1133					;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1134	003024	004767	013770		JSR PC,RPOKE ;SHIFT IN THIS CHAR
1135	003030	016703	014264		MOV RXDBUF,R3 ;FOR ERROR MESSAGE
1136	003034	012700	120125		MOV #RXERR!FRMERR!125,R0 ;EXPECTED
1137	003040	017701	014254		MOV @RXDBUF,R1 ;ACTUAL
1138	003044	020001			CMP R0,R1 ;COMPARE EXP VS ACT
1139	003046	001401			BEQ 64\$
1140	003050	104002			HLT 2 ;FRAME ERROR & RX ERROR SHOULD BE SET
1141					;IF LOWER BYTE DOES NOT MATCH IT
1142					;PROBABLY IS A LENGTH SELECT PROBLEM
1143	003052				64\$:
1144	003052	104400			SCOPE
1145					;;THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1146					;;SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1147					;;THAT IT WAS SELECTED PROPERLY
1148					;;FRAME ERROR (FRMERR,RXERR)

```

1149                                     ;;MODE:ISOC (ISYMOD)
1150                                     ;;LENGTH:EIGHT
1151                                     ;;CHAR: 125
1152
1153 003054 012767 000004 176044 TST4:  MOV    #4,TSTNO          ;SAVE THIS
1154 003062 012767 003230 176026      MOV    #TST5,NEXT        ;GO TO THIS TEST WHEN THRU
1155 003070 052777 000400 014232      BIS    #MRESET,@TXCSR   ;MASTER RESET
1156 003076 012777 000000 014220      MOV    #ISYMOD,@PARCSR  ;SET THE MODE
1157 003104 052777 000400 014216      BIS    #MRESET,@TXCSR   ;MASTER RESET
1158
1159                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1160 003112 012777 064001 014210      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1161
1162                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1163 003120 012777 006000 014176      MOV    #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1164 003126 052777 000020 014160      BIS    #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1165                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1166 003134 042777 020000 014166      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1167 003142 052777 020000 014160      BIS    #CLK,@TXCSR      ;POKE CLK UP
1168                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1169 003150 042777 020000 014152      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1170 003156 052777 020000 014144      BIS    #CLK,@TXCSR      ;POKE CLK UP
1171 003164 012767 000012 175746      MOV    #10,SHIFT        ;# OF SHIFTS
1172 003172 012767 000252 175744      MOV    #252,TEMP1       ;DATA CHAR
1173                                     ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1174 003200 004767 013614                JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
1175 003204 016703 014110                MOV    RXDBUF,R3        ;FOR ERROR MESSAGE
1176 003210 012700 120125                MOV    #RXERR!FRMERR!125,R0 ;EXPECTED
1177 003214 017701 014100                MOV    @RXDBUF,R1       ;ACTUAL
1178 003220 020001                CMP    R0,R1            ;COMPARE EXP VS ACT
1179 003222 001401                BEQ    64$
1180 003224 104002                HLT    2                ;FRAME ERROR & RX ERROR SHOULD BE SET
1181                                     ;IF LOWER BYTE DOES NOT MATCH IT
1182                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1183 003226                64$:
1184 003226 104400                SCOPE
1185                                     ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1186                                     ;;OF THE RECEIVER
1187                                     ;;MODE:ISOC (ISYMOD)
1188                                     ;;PARITY:EVEPAR
1189                                     ;;LENGTH:FIVE PLUS PARITY
1190                                     ;;CHAR: 25
1191
1192 003230 012767 000005 175670 TST5:  MOV    #5,TSTNO          ;SAVE THIS
1193 003236 012767 003414 175652      MOV    #TST6,NEXT        ;GO TO THIS TEST WHEN THRU
1194 003244 052777 000400 014056      BIS    #MRESET,@TXCSR   ;MASTER RESET
1195 003252 012777 000000 014044      MOV    #ISYMOD,@PARCSR  ;SET THE MODE
1196 003260 052777 000400 014042      BIS    #MRESET,@TXCSR   ;MASTER RESET
1197
1198                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1199 003266 012777 064001 014034      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1200
1201                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1202 003274 012777 001400 014022      MOV    #ISYMOD!FIVE!EVEPAR!0,@PARCSR
1203 003302 052777 000020 014004      BIS    #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1204                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....

```


M02

DZDUC-S MACY11 27(1006) 01-OCT-76 09:28 PAGE 28
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0025

```

1205 003310 042777 020000 014012      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1206 003316 052777 020000 014004      BIS      #CLK,@TXCSR      ;POKE CLK UP
1207      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1208 003324 042777 020000 013776      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1209 003332 052777 020000 013770      BIS      #CLK,@TXCSR      ;POKE CLK UP
1210 003340 016703 013754      MOV      RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1211 003344 012700 110025      MOV      #RXERR!PARER!25,R0 ;EXPECTED
1212 003350 012767 000010 175562      MOV      #8,SHIFT        ;# OF SHIFTS
1213 003356 012767 000252 175560      MOV      #252,TEMP1      ;DATA CHAR
1214 003364 004767 013430      JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1215 003370 105777 013720      TSTB    @RXCSR ;RXDONE ?
1216 003374 100401      BMI     1$
1217 003376 104000      HLT
1218 003400      ;RXDONE SHOULD BE ASSERTED
1219 003400 017701 013714      1$:    MOV      @RXDBUF,R1    ;ACTUAL
1220 003404 020001      CMP     R0,R1            ;COMPARE EXP VS. ACT
1221 003406 001401      BEQ    2$
1222 003410 104002      HLT     2
1223 003412      ;PARITY ERROR &RXERR SHOULD BE SET
1224      ;NOTE THAT THE PARITY BIT SHOULD
1225      ;SHOW UP IN THE DATA
1226      ;IE. BIT FIVE FOR FIVE LEVEL CODE
1227 003412 104400      SCOPE
1228      ;;THIS TEST VERIFYS EVEPAR PARITY SENSE
1229      ;;OF THE RECEIVER
1230      ;;MODE:ISOC (ISYMOD)
1231      ;;PARITY:EVEPAR
1232      ;;LENGTH:SIX PLUS PARITY
1233      ;;CHAR: 25
1234      ;
1235 003414 012767 000006 175504      TST6:  MOV      #6,TSTNO    ;SAVE THIS
1236 003422 012767 003600 175466      MOV      #TST7,NEXT      ;GO TO THIS TEST WHEN THRU
1237 003430 052777 000400 013672      BIS      #MRESET,@TXCSR ;MASTER RESET
1238 003436 012777 000000 013660      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1239 003444 052777 000400 013656      BIS      #MRESET,@TXCSR ;MASTER RESET
1240      ;
1241      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1242 003452 012777 064001 013650      MOV      #MCDATA!CLK!MINT!BREAK,@TXCSR
1243      ;
1244      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1245 003460 012777 003400 013636      MOV      #ISYMOD!SIX!EVEPAR!0,@PARCSR
1246 003466 052777 000020 013620      BIS      #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1247      ;POKE CLK TO GET RECEIVER INTO SYNCRONIZATION....
1248 003474 042777 020000 013626      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1249 003502 052777 020000 013620      BIS      #CLK,@TXCSR      ;POKE CLK UP
1250      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1251 003510 042777 020000 013612      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1252 003516 052777 020000 013604      BIS      #CLK,@TXCSR      ;POKE CLK UP
1253 003524 016703 013570      MOV      RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1254 003530 012700 110025      MOV      #RXERR!PARER!25,R0 ;EXPECTED
1255 003534 012767 000011 175376      MOV      #9,SHIFT        ;# OF SHIFTS
1256 003542 012767 000452 175374      MOV      #452,TEMP1      ;DATA CHAR
1257 003550 004767 013244      JSR      PC,RPOKE        ;SHIFT IN THIS CHAR
1258 003554 105777 013534      TSTB    @RXCSR ;RXDONE ?
1259 003560 100401      BMI     1$
1260 003562 104000      HLT
      ;RXDONE SHOULD BE ASSERTED

```



```

1317                                     ;; PARITY: EVEPAR
1318                                     ;; LENGTH: EIGHT PLUS PARITY
1319                                     ;; CHAR: 125
1320
1321 003764 012767 000010 175134 TST8:  MOV    #8, TSTNO          ; SAVE THIS
1322 003772 012767 004150 175116      MOV    #TST9, NEXT        ; GO TO THIS TEST WHEN THRU
1323 004000 052777 000400 013322      BIS    #MRESET, @TXCSR   ; MASTER RESET
1324 004006 012777 000000 013310      MOV    #ISYMOD, @PARCSR  ; SET THE MODE
1325 004014 052777 000400 013306      BIS    #MRESET, @TXCSR   ; MASTER RESET
1326
1327                                     ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1328 004022 012777 064001 013300      MOV    #MNTDATA!CLK!MINT!BREAK, @TXCSR
1329
1330                                     ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1331 004030 012777 007400 013266      MOV    #ISYMOD!EIGHT!EVEPAR!0, @PARCSR
1332 004036 052777 000020 013250      BIS    #SYNSCH, @RXCSR   ; SET SYNC SEARCH
1333                                     ; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1334 004044 042777 020000 013256      BIC    #CLK, @TXCSR      ; POKE CLK DOWN
1335 004052 052777 020000 013250      BIS    #CLK, @TXCSR      ; POKE CLK UP
1336                                     ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1337 004060 042777 020000 013242      BIC    #CLK, @TXCSR      ; POKE CLK DOWN
1338 004066 052777 020000 013234      BIS    #CLK, @TXCSR      ; POKE CLK UP
1339 004074 016703 013220                MOV    RXDBUF, R3         ; SET UP FOR ERROR MESSAGE
1340 004100 012700 110125                MOV    #RXERR!PARER!125, R0 ; EXPECTED
1341 004104 012767 000013 175026      MOV    #11, SHIFT        ; # OF SHIFTS
1342 004112 012767 003252 175024      MOV    #3252, TEMP1      ; DATA CHAR
1343 004120 004767 012674                JSR    PC, RPOKE          ; SHIFT IN THIS CHAR
1344 004124 105777 013164                TSTB   @RXCSR ; RXDONE ?
1345 004130 100401                BMI    1$
1346 004132 104000                HLT
1347                                     1$:
1348 004134 017701 013160                MOV    @RXDBUF, R1        ; ACTUAL
1349 004140 020001                CMP    R0, R1            ; COMPARE EXP VS. ACT
1350 004142 001401                BEQ    2$
1351 004144 104002                HLT    2
1352                                     2$:
1353 004146 104400                SCOPE
1354                                     ;; THIS TEST VERIFYS ODDPAR PARITY SENSE
1355                                     ;; OF THE RECEIVER
1356                                     ;; MODE: ISOC (ISYMOD)
1357                                     ;; PARITY: ODDPAR
1358                                     ;; LENGTH: FIVE PLUS PARITY
1359                                     ;; CHAR: 65
1360
1361 004150 012767 000011 174750 TST9:  MOV    #9, TSTNO          ; SAVE THIS
1362 004156 012767 004334 174732      MOV    #TST10, NEXT      ; GO TO THIS TEST WHEN THRU
1363 004164 052777 000400 013136      BIS    #MRESET, @TXCSR   ; MASTER RESET
1364 004172 012777 000000 013124      MOV    #ISYMOD, @PARCSR  ; SET THE MODE
1365 004200 052777 000400 013122      BIS    #MRESET, @TXCSR   ; MASTER RESET
1366
1367                                     ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1368 004206 012777 064001 013114      MOV    #MNTDATA!CLK!MINT!BREAK, @TXCSR
1369
1370                                     ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1371 004214 012777 001000 013102      MOV    #ISYMOD!FIVE!ODDPAR!0, @PARCSR
1372 004222 052777 000020 013064      BIS    #SYNSCH, @RXCSR   ; SET SYNC SEARCH
  
```

```

1373      :POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1374 004230 042777 020000 013072      BIC  #CLK,@TXCSR      :POKE CLK DOWN
1375 004236 052777 020000 013064      BIS  #CLK,@TXCSR      :POKE CLK UP
1376      :POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1377 004244 042777 020000 013056      BIC  #CLK,@TXCSR      :POKE CLK DOWN
1378 004252 052777 020000 013050      BIS  #CLK,@TXCSR      :POKE CLK UP
1379 004260 016703 013034      MOV  RXDBUF,R3        :SET UP FOR ERROR MESSAGE
1380 004264 012700 110065      MOV  #RXERR!PARER!65,RO :EXPECTED
1381 004270 012767 000010 174642      MOV  #8,SHIFT         :# OF SHIFTS
1382 004276 012767 000352 174640      MOV  #352,TEMP1       :DATA CHAR
1383 004304 004767 012510      JSR  PC,RPOKE         :SHIFT IN THIS CHAR
1384 004310 105777 013000      TSTB @RXCSR ;RXDONE ?
1385 004314 100401      BMI  1$
1386 004316 104000      HLT
1387 004320      :RXDONE SHOULD BE ASSERTED
1388 004320 017701 012774      1$: MOV  @RXDBUF,R1       :ACTUAL
1389 004324 020001      CMP  RO,R1           :COMPARE EXP VS. ACT
1390 004326 001401      BEQ  2$
1391 004330 104002      HLT  2
1392 004332      :PARITY ERROR &RXERR SHOULD BE SET
1393      :NOTE THAT THE PARITY BIT SHOULD
1394      :SHOW UP IN THE DATA
1395      :IE. BIT FIVE FOR FIVE LEVEL CODE
1396 004332 104400      SCOPE
1397      ::THIS TEST VERIFYS ODDPAR PARITY SENSE
1398      ::OF THE RECEIVER
1399      ::MODE:ISOC (ISYMOD)
1400      ::PARITY:ODDPAR
1401      ::LENGTH:SIX PLUS PARITY
1402      ::CHAR: 125
1403      :
1404 004334 012767 000012 174564      TST10: MOV  #10,TSTNO       ;SAVE THIS
1405 004342 012767 004520 174546      MOV  #TST11,NEXT      ;GO TO THIS TEST WHEN THRU
1406 004350 052777 000400 012752      BIS  #MRESET,@TXCSR   ;MASTER RESET
1407 004356 012777 000000 012740      MOV  #ISYMOD,@PARCSR  ;SET THE MODE
1408 004364 052777 000400 012736      BIS  #MRESET,@TXCSR   ;MASTER RESET
1409
1410      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1411 004372 012777 064001 012730      MOV  #MNTDATA!CLK!MINT!BREAK,@TXCSR
1412
1413      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1414 004400 012777 003000 012716      MOV  #ISYMOD!SIX!ODDPAR!0,@PARCSR
1415 004406 052777 000020 012700      BIS  #SYNSCH,@RXCSR   :SET SYNC SEARCH
1416      :POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1417 004414 042777 020000 012706      BIC  #CLK,@TXCSR      :POKE CLK DOWN
1418 004422 052777 020000 012700      BIS  #CLK,@TXCSR      :POKE CLK UP
1419      :POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1420 004430 042777 020000 012672      BIC  #CLK,@TXCSR      :POKE CLK DOWN
1421 004436 052777 020000 012664      BIS  #CLK,@TXCSR      :POKE CLK UP
1422 004444 016703 012650      MOV  RXDBUF,R3        :SET UP FOR ERROR MESSAGE
1423 004450 012700 110125      MOV  #RXERR!PARER!125,RO :EXPECTED
1424 004454 012767 000011 174456      MOV  #9,SHIFT         :# OF SHIFTS
1425 004462 012767 000652 174454      MOV  #652,TEMP1       :DATA CHAR
1426 004470 004767 012324      JSR  PC,RPOKE         :SHIFT IN THIS CHAR
1427 004474 105777 012614      TSTB @RXCSR ;RXDONE ?
1428 004500 100401      BMI  1$

```



```

1429 004502 104000          HLT          ;RXDONE SHOULD BE ASSERTED
1430 004504          1$:          MOV          @RXDBUF,R1          ;ACTUAL
1431 004504 017701 012610    CMP          R0,R1          ;COMPARE EXP VS. ACT
1432 004510 020001          BEQ          2$
1433 004512 001401          HLT          2$          ;PARITY ERROR &RXERR SHOULD BE SET
1434 004514 104002          2$:          HLT          ;NOTE THAT THE PARITY BIT SHOULD
1435 004516          ;SHOW UP IN THE DATA
1436          ;IE. BIT SIX FOR SIX LEVEL CODE
1437
1438
1439 004516 104400          SCOPE
1440          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
1441          ;;OF THE RECEIVER
1442          ;;MODE:ISOC (ISYMOD)
1443          ;;PARITY:ODDPAR
1444          ;;LENGTH:SEVEN PLUS PARITY
1445          ;;CHAR: 125
1446
1447 004520 012767 000013 174400  TST11:  MOV          #11,TSTNO          ;SAVE THIS
1448 004526 012767 004704 174362  MOV          #TST12,NEXT          ;GO TO THIS TEST WHEN THRU
1449 004534 052777 000400 012566  BIS          #MRESET,@TXCSR      ;MASTER RESET
1450 004542 012777 000000 012554  MOV          #ISYMOD,@PARCSR     ;SET THE MODE
1451 004550 052777 000400 012552  BIS          #MRESET,@TXCSR      ;MASTER RESET
1452
1453          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1454 004556 012777 064001 012544  MOV          #MTDATA!CLK!MINT!BREAK,@TXCSR
1455
1456          ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1457 004564 012777 005000 012532  MOV          #ISYMOD!SEVEN!ODDPAR!0,@PARCSR
1458 004572 052777 000020 012514  BIS          #SYNSCH,@RXCSR      ;SET SYNC SEARCH
1459          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1460 004600 042777 020000 012522  BIC          #CLK,@TXCSR          ;POKE CLK DOWN
1461 004606 052777 020000 012514  BIS          #CLK,@TXCSR          ;POKE CLK UP
1462          ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1463 004614 042777 020000 012506  BIC          #CLK,@TXCSR          ;POKE CLK DOWN
1464 004622 052777 020000 012500  BIS          #CLK,@TXCSR          ;POKE CLK UP
1465 004630 016703 012464          MOV          RXDBUF,R3          ;SET UP FOR ERROR MESSAGE
1466 004634 012700 110125          MOV          #RXERR!PARER!125,R0 ;EXPECTED
1467 004640 012767 000012 174272  MOV          #10,SHIFT           ;# OF SHIFTS
1468 004646 012767 001252 174270  MOV          #1252,TEMP1         ;DATA CHAR
1469 004654 004767 012140          JSR          PC,RPOKE           ;SHIFT IN THIS CHAR
1470 004660 105777 012430          TSTB         @RXCSR          ;RXDONE ?
1471 004664 100401          BMI          1$
1472 004666 104000          HLT          ;RXDONE SHOULD BE ASSERTED
1473 004670          1$:          MOV          @RXDBUF,R1          ;ACTUAL
1474 004670 017701 012424    CMP          R0,R1          ;COMPARE EXP VS. ACT
1475 004674 020001          BEQ          2$
1476 004676 001401          HLT          2$          ;PARITY ERROR &RXERR SHOULD BE SET
1477 004700 104002          2$:          HLT          ;NOTE THAT THE PARITY BIT SHOULD
1478 004702          ;SHOW UP IN THE DATA
1479          ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
1480
1481
1482 004702 104400          SCOPE
1483          ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
1484          ;;OF THE RECEIVER
    
```

E03

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 33
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0030

```

1485                                     ;;MODE:ISOC (ISYMOD)
1486                                     ;;PARITY:ODDPAR
1487                                     ;;LENGTH:EIGHT PLUS PARITY
1488                                     ;;CHAR: 125
1489                                     ;;
1490 004704 012767 000014 174214 TST12: MOV #12,TSTNO ;SAVE THIS
1491 004712 012767 005070 174176 MOV #TST13,NEXT ;GO TO THIS TEST WHEN THRU
1492 004720 052777 000400 012402 BIS #MRESET,@TXCSR ;MASTER RESET
1493 004726 012777 000000 012370 MOV #ISYMOD,@PARCSR ;SET THE MODE
1494 004734 052777 000400 012366 BIS #MRESET,@TXCSR ;MASTER RESET
1495
1496                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1497 004742 012777 064001 012360 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1498
1499                                     ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
1500 004750 012777 007000 012346 MOV #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
1501 004756 052777 000020 012330 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1502                                     ;POKE CLK TO GET RECEIVER INTO SYNCHRONIZATION....
1503 004764 042777 020000 012336 BIC #CLK,@TXCSR ;POKE CLK DOWN
1504 004772 052777 020000 012330 BIS #CLK,@TXCSR ;POKE CLK UP
1505                                     ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1506 005000 042777 020000 012322 BIC #CLK,@TXCSR ;POKE CLK DOWN
1507 005006 052777 020000 012314 BIS #CLK,@TXCSR ;POKE CLK UP
1508 005014 016703 012300 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1509 005020 012700 110125 MOV #RXERR!PARER!125,R0 ;EXPECTED
1510 005024 012767 000013 174106 MOV #11,SHIFT ;# OF SHIFTS
1511 005032 012767 002252 174104 MOV #2252,TEMP1 ;DATA CHAR
1512 005040 004767 011754 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1513 005044 105777 012244 TSTB @RXCSR ;RXDONE
1514 005050 100401 BMI IS
1515 005052 104000 HLT ;RXDONE SHOULD BE ASSERTED
1516 005054
1517 005054 017701 012240 IS: MOV @RXDBUF,R1 ;ACTUAL
1518 005060 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1519 005062 001401 BEQ 2$
1520 005064 104002 HLT 2 ;PARITY ERROR &RXERR SHOULD BE SET
1521 005066
1522 005066 104400 2$:
1523                                     SCOPE
1524                                     ;;THIS TEST PERFORMS BINARY DATA CHECK ON THE
1525                                     ;;RECEIVER
1526                                     ;;LENGTH:EIGHT PLUS PARITY
1527                                     ;;MODE:ISYMOD
1528                                     ;;PARITY:EVEPAR
1529 005070 012767 000015 174030 TST13: MOV #13,TSTNO ;SAVE THIS
1530 005076 012767 005270 174012 MOV #TST14,NEXT ;GO TO THIS TEST WHEN THRU
1531 005104 052777 000400 012216 BIS #MRESET,@TXCSR ;MASTER RESET
1532 005112 012777 000000 012204 MOV #ISYMOD,@PARCSR ;SET THE MODE
1533 005120 052777 000400 012202 BIS #MRESET,@TXCSR ;MASTER RESET
1534
1535                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1536 005126 012777 064001 012174 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1537
1538                                     ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
1539 005134 012777 007400 012162 MOV #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1540 005142 052777 000020 012144 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
  
```


F03

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 34
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0031

```

1541                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1542 005150 042777 020000 012152      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1543 005156 052777 020000 012144      BIS   #CLK,@TXCSR      ;POKE CLK UP
1544                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1545 005164 042777 020000 012136      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1546 005172 052777 020000 012130      BIS   #CLK,@TXCSR      ;POKE CLK UP
1547 005200 016703 012114                MOV   RXDBUF,R3        ;SET UP ERROR MESSAGE
1548 005204 005004                CLR   R4               ;DATA CHAR
1549 005206 010400                MOV   R4,R0            ;EXPECTED
1550 005210 012767 000013 173722      1$:  MOV   #11,SHIFT      ;# OF SHIFTS
1551 005216 010467 173722                MOV   R4,TEMP1        ;"TO BE SHIFTED CHARACTER"
1552 005222 004767 011744                JSR   PC,EVENB        ;CALC PARITY
1553 005226 000241                CLC
1554 005230 006167 173710                ROL   TEMP1           ;GENERATE START BIT
1555 005234 052767 002000 173702      2$:  BIS   #BIT10,TEMP1    ;GENERATE STOP BIT
1556                                     ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1557 005242 004767 011552                JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
1558 005246 017701 012046                MOV   #RXDBUF,R1      ;ACTUAL
1559 005252 020001                CMP   R0,R1           ;COMPARE EXP VS ACT
1560 005254 001401                BEQ   2$
1561 005256 104002                HLT
1562                                     ;DATA CHARS SHOULD MATCH
1563                                     ;THERE SHOULD BE NO PARITY ERROR
1564 005260                2$:  INC   R4               ;UPGRADE NEXT CHAR
1565 005262 105704                TSTB  R4               ;LAST CHAR ?
1566 005264 001350                BNE   1$
1567 005266 104400                SCOPE
1568                                     ;; THIS TEST PERFORMS BINARY DATA CHECK ON THE
1569                                     ;; RECEIVER
1570                                     ;; LENGTH:EIGHT PLUS PARITY
1571                                     ;; MODE:ISYMOD
1572                                     ;; PARITY:ODDPAR
1573                                     ;;
1574 005270 012767 000016 173630      TST14: MOV   #14,TSTNO    ;SAVE THIS
1575 005276 012767 005470 173612      MOV   #TST15,NEXT     ;GO TO THIS TEST WHEN THRU
1576 005304 052777 000400 012016      BIS   #MRESET,@TXCSR  ;MASTER RESET
1577 005312 012777 000000 012004      MOV   #ISYMOD,@PARCSR ;SET THE MODE
1578 005320 052777 000400 012002      BIS   #MRESET,@TXCSR  ;MASTER RESET
1579
1580                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1581 005326 012777 064001 011774      MOV   #MNTDATA!CLK!MINT!BREAK,@TXCSR
1582
1583                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1584 005334 012777 007000 011762      MOV   #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
1585 005342 052777 000020 011744      BIS   #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1586                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1587 005350 042777 020000 011752      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1588 005356 052777 020000 011744      BIS   #CLK,@TXCSR      ;POKE CLK UP
1589                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1590 005364 042777 020000 011736      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1591 005372 052777 020000 011730      BIS   #CLK,@TXCSR      ;POKE CLK UP
1592 005400 016703 011714                MOV   RXDBUF,R3        ;SET UP ERROR MESSAGE
1593 005404 005004                CLR   R4               ;DATA CHAR
1594 005406 010400                MOV   R4,R0            ;EXPECTED
1595 005410 012767 000013 173522      1$:  MOV   #11,SHIFT      ;# OF SHIFTS
1596 005416 010467 173522                MOV   R4,TEMP1        ;"TO BE SHIFTED CHARACTER"

```

G03

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 35
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0032

```

1597 005422 004767 011460      JSR      PC,ODDB ;CALC PARITY
1598 005426 000241              CLC
1599 005430 006167 173510      ROL      TEMP1 ;GENERATE START BIT
1600 005434 052767 002000 173502  BIS      #BIT10,TEMP1 ;GENERATE STOP BIT
1601                                ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1602 005442 004767 011352      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1603 005446 017701 011646      MOV      @RXDBUF,R1 ;ACTUAL
1604 005452 020001              CMP      R0,R1 ;COMPARE EXP VS ACT
1605 005454 001401              BEQ      2$
1606 005456 104002              HLT      2 ;DATA CHARS SHOULD MATCH
1607                                ;THERE SHOULD BE NO PARITY ERROR
1608                                2$:
1609 005460              INC      R4 ;UPGRADE NEXT CHAR
1610 005462 105704              TSTB    R4 ;LAST CHAR ?
1611 005464 001350              BNE     1$
1612 005466 104400              SCOPE
1613                                ;; THIS TEST PERFORMS BINARY DATA CHECK ON THE
1614                                ;; RECEIVER
1615                                ;; LENGTH:EIGHT PLUS PARITY
1616                                ;; MODE:SYNEXT
1617                                ;; PARITY:EVEPAR
1618                                ;;
1619 005470 012767 000017 173430  TST15: MOV     #15,TSTNO ;SAVE THIS
1620 005476 012767 005640 173412  MOV     #TST16,NEXT ;GO TO THIS TEST WHEN THRU
1621 005504 052777 000400 011616  BIS     #MRESET,@TXCSR ;MASTER RESET
1622 005512 012777 020000 011604  MOV     #SYNEXT,@PARCSR ;SET THE MODE
1623 005520 052777 000400 011602  BIS     #MRESET,@TXCSR ;MASTER RESET
1624
1625                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1626 005526 012777 064001 011574  MOV     #MCDATA!CLK!MINT!BREAK,@TXCSR
1627
1628                                ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
1629 005534 012777 027400 011562  MOV     #SYNEXT!EIGHT!EVEPAR!0,@PARCSR
1630 005542 052777 000020 011544  BIS     #SYNSCH,@RXCSR ;SET SEARCH SYNC
1631                                ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1632 005550 042777 020000 011552  BIC     #CLK,@TXCSR ;POKE CLK DOWN
1633 005556 052777 020000 011544  BIS     #CLK,@TXCSR ;POKE CLK UP
1634 005564 016703 011530      MOV     RXDBUF,R3 ;SET UP ERROR MESSAGE
1635 005570 005004              CLR     R4 ;DATA CHAR
1636 005572 010400              1$: MOV     R4,R0 ;EXPECTED
1637 005574 012767 000011 173336  MOV     #9,SHIFT ;# OF SHIFTS
1638 005602 010467 173336      MOV     R4,TEMP1 ;"TO BE SHIFTED CHARACTER"
1639 005606 004767 011360      JSR     PC,EVENB ;CALC PARITY
1640                                ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1641 005612 004767 011202      JSR     PC,RPOKE ;SHIFT IN THIS CHAR
1642 005616 017701 011476      MOV     @RXDBUF,R1 ;ACTUAL
1643 005622 020001              CMP     R0,R1 ;COMPARE EXP VS ACT
1644 005624 001401              BEQ     2$
1645 005626 104002              HLT     2 ;DATA CHARS SHOULD MATCH
1646                                ;THERE SHOULD BE NO PARITY ERROR
1647                                2$:
1648 005630              INC     R4 ;UPGRADE NEXT CHAR
1649 005632 105704              TSTB    R4 ;LAST CHAR ?
1650 005634 001356              BNE     1$
1651 005636 104400              SCOPE
1652                                ;; THIS TEST PERFORMS BINARY DATA CHECK ON THE
  
```



```

1653                                     ::RECEIVER
1654                                     ::LENGTH:EIGHT PLUS PARITY
1655                                     ::MODE:SYNEXT
1656                                     ::PARITY:ODDPAR
1657
1658 005640 012767 000020 173260 TST16: MOV #16,TSTNO ;SAVE THIS
1659 005646 012767 006010 173242 MOV #TST17,NEXT ;GO TO THIS TEST WHEN THRU
1660 005654 052777 000400 011446 BIS #MRESET,@TXCSR ;MASTER RESET
1661 005662 012777 020000 011434 MOV #SYNEXT,@PARCSR ;SET THE MODE
1662 005670 052777 000400 011432 BIS #MRESET,@TXCSR ;MASTER RESET
1663
1664 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1665 005676 012777 064001 011424 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1666
1667 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1668 005704 012777 027000 011412 MOV #SYNEXT!EIGHT!ODDPAR!0,@PARCSR
1669 005712 052777 000020 011374 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1670 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1671 005720 042777 020000 011402 BIC #CLK,@TXCSR ;POKE CLK DOWN
1672 005726 052777 020000 011374 BIS #CLK,@TXCSR ;POKE CLK UP
1673 005734 016703 011360 MOV RXDBUF,R3 ;SET UP ERROR MESSAGE
1674 005740 005004 CLR R4 ;DATA CHAR
1675 005742 010400 1S: MOV R4,R0 ;EXPECTED
1676 005744 012767 000011 173166 MOV #9,SHIFT ;# OF SHIFTS
1677 005752 010467 173166 MOV R4,TEMP1 ;"TO BE SHIFTED CHARACTER"
1678 005756 004767 011124 JSR PC,ODDB ;CALC PARITY
1679 ;TEMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
1680 005762 004767 011032 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1681 005766 017701 011326 MOV @RXDBUF,R1 ;ACTUAL
1682 005772 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1683 005774 001401 BEQ 2$
1684 005776 104002 HLT 2 ;DATA CHARS SHOULD MATCH
1685 ;THERE SHOULD BE NO PARITY ERROR
1686 006000 2$: INC R4 ;UPGRADE NEXT CHAR
1687 006000 005204 TSTB R4 ;LAST CHAR ?
1688 006002 105704 BNE 1$
1689 006004 001356 SCOPE
1690 006006 104400
1691 ::THIS TEST CHECKS THE STRIP SYNC FUNCTION
1692 ::OF THE RECEIVER LOGIC
1693 ::MODE:ISYMOD
1694 ::LENGTH:FIVE
1695 ::NOTE: RXDONE SHOULD NEVER ASSERT
1696 ::CHAR: 26 (SYNC)
1697
1698 006010 012767 000021 173110 TST17: MOV #17,TSTNO ;SAVE THIS
1699 006016 012767 006174 173072 MOV #TST18,NEXT ;GO TO THIS TEST WHEN THRU
1700 006024 052777 000400 011276 BIS #MRESET,@TXCSR ;MASTER RESET
1701 006032 012777 000000 011264 MOV #ISYMOD,@PARCSR ;SET THE MODE
1702 006040 052777 000400 011262 BIS #MRESET,@TXCSR ;MASTER RESET
1703
1704 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1705 006046 012777 064001 011254 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1706
1707 ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1708 006054 012777 000026 011242 MOV #ISYMOD!FIVE!NOPAR!26,@PARCSR
    
```

1709	006062	052777	000020	011224	BIS	#SYNSCH,@RXCSR	:SET SYNC SEARCH	
1710							:POKE CLK TO GET RECEIVER INTO SYNCROIZATION....	
1711	006070	042777	020000	011232	BIC	#CLK,@TXCSR	:POKE CLK DOWN	
1712	006076	052777	020000	011224	BIS	#CLK,@TXCSR	:POKE CLK UP	
1713							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION	
1714	006104	042777	020000	011216	BIC	#CLK,@TXCSR	:POKE CLK DOWN	
1715	006112	052777	020000	011210	BIS	#CLK,@TXCSR	:POKE CLK UP	
1716	006120	052777	000400	011166	BIS	#STPSYN,@RXCSR	:SET STRIP SYNC	
1717	006126	012767	000003	173006	MOV	#3,COUNT	:# OF SYNC CHARS	
1718	006134	012767	000154	173002	1\$:	MOV	#154,TEMP1	:CHAR TO BE SHIFTED
1719	006142	012767	000007	172770	MOV	#7,SHIFT	:# OF SHIFTS	
1720	006150	004767	010644		JSR	PC,RPOKE	:SHIFT IN THIS CHAR	
1721	006154	105777	011134		TSTB	@RXCSR ;RXDONE ?		
1722	006160	100001			BPL	2\$		
1723	006162	104000			HLT	;RXDONE SHOULD NOT BE ASSERTED		
1724	006164				2\$:			
1725	006164	005367	172752		DEC	COUNT ;# OF SYNC CHARS		
1726	006170	001361			BNE	1\$		
1727	006172	104400			SCOPE			
1728							::THIS TEST CHECKS THE STRIP SYNC FUNCTION	
1729							::OF THE RECEIVER LOGIC	
1730							::MODE:ISYMOD	
1731							::LENGTH:SIX	
1732							::NOTE:RXDONE SHOULD NEVER ASSERT	
1733							::CHAR:26 (SYNC)	
1734							::	
1735	006174	012767	000022	172724	TST18:	MOV	#18,TSTNO ;SAVE THIS	
1736	006202	012767	006360	172706	MOV	#TST19,NEXT ;GO TO THIS TEST WHEN THRU		
1737	006210	052777	000400	011112	BIS	#MRESET,@TXCSR ;MASTER RESET		
1738	006216	012777	000000	011100	MOV	#ISYMOD,@PARCSR ;SET THE MODE		
1739	006224	052777	000400	011076	BIS	#MRESET,@TXCSR ;MASTER RESET		
1740								
1741							;SET MAINT DATA,CLK BREAK,&MAINTENANCE MODE	
1742	006232	012777	064001	011070	MOV	#MNTDATA!CLK!MINT!BREAK,@TXCSR		
1743								
1744							;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG	
1745	006240	012777	002026	011056	MOV	#ISYMOD!SIX!NOPAR!26,@PARCSR		
1746	006246	052777	000020	011040	BIS	#SYNSCH,@RXCSR ;SET SYNC SEARCH		
1747							:POKE CLK TO GET RECEIVER INTO SYNCROIZATION....	
1748	006254	042777	020000	011046	BIC	#CLK,@TXCSR ;POKE CLK DOWN		
1749	006262	052777	020000	011040	BIS	#CLK,@TXCSR ;POKE CLK UP		
1750							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION	
1751	006270	042777	020000	011032	BIC	#CLK,@TXCSR ;POKE CLK DOWN		
1752	006276	052777	020000	011024	BIS	#CLK,@TXCSR ;POKE CLK UP		
1753	006304	052777	000400	011002	BIS	#STPSYN,@RXCSR ;SET STRIP SYNC		
1754	006312	012767	000003	172622	MOV	#3,COUNT	:# OF SYNC CHARS	
1755	006320	012767	000254	172616	1\$:	MOV	#254,TEMP1	:CHAR TO BE SHIFTED
1756	006326	012767	000010	172604	MOV	#8,SHIFT	:# OF SHIFTS	
1757	006334	004767	010460		JSR	PC,RPOKE	:SHIFT IN THIS CHAR	
1758	006340	105777	010750		TSTB	@RXCSR ;RXDONE ?		
1759	006344	100001			BPL	2\$		
1760	006346	104000			HLT	;RXDONE SHOULD NOT BE ASSERTED		
1761	006350				2\$:			
1762	006350	005367	172566		DEC	COUNT ;# OF SYNC CHARS		
1763	006354	001361			BNE	1\$		
1764	006356	104400			SCOPE			


```

1765      ;: THIS TEST CHECKS THE STRIP SYNC FUNCTION
1766      ;: OF THE RECEIVER LOGIC
1767      ;: MODE: ISYMOD
1768      ;: LENGTH: SEVEN
1769      ;: NOTE: RXDONE SHOULD NEVER ASSERT
1770      ;: CHAR: 26 (SYNC)
1771
1772 006360 012767 000023 172540 TST19: MOV     #19,TSTNO      ;SAVE THIS
1773 006366 012767 006544 172522      MOV     #TST20,NEXT      ;GO TO THIS TEST WHEN THRU
1774 006374 052777 000400 010726      BIS     #MRESET,@TXCSR  ;MASTER RESET
1775 006402 012777 000000 010714      MOV     #ISYMOD,@PARCSR ;SET THE MODE
1776 006410 052777 000400 010712      BIS     #MRESET,@TXCSR  ;MASTER RESET
1777
1778      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1779 006416 012777 064001 010704      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1780
1781      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1782 006424 012777 004026 010672      MOV     #ISYMOD!SEVEN!NOPAR!26,@PARCSR
1783 006432 052777 000020 010654      BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1784      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1785 006440 042777 020000 010662      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1786 006446 052777 020000 010654      BIS     #CLK,@TXCSR     ;POKE CLK UP
1787      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1788 006454 042777 020000 010646      BIC     #CLK,@TXCSR     ;POKE CLK DOWN
1789 006462 052777 020000 010640      BIS     #CLK,@TXCSR     ;POKE CLK UP
1790 006470 052777 000400 010616      BIS     #STPSYN,@RXCSR  ;SET STRIP SYNC
1791 006476 012767 000003 172436      MOV     #3,COUNT        ;# OF SYNC CHARS
1792 006504 012767 000454 172432 1$:  MOV     #454,TEMP1      ;CHAR TO BE SHIFTED
1793 006512 012767 000011 172420      MOV     #9,SHIFT        ;# OF SHIFTS
1794 006520 004767 010274      JSR     PC,@POKE        ;SHIFT IN THIS CHAR
1795 006524 105777 010564      TSTB   @RXCSR ;RXDONE ?
1796 006530 100001      BPL     2$
1797 006532 104000      HLT     ;RXDONE SHOULD NOT BE ASSERTED
1798
1799 006534 005367 172402 2$:  DEC     COUNT ;# OF SYNC CHARS
1800 006540 001361      BNE     1$
1801 006542 104400      SCOPE
1802      ;: THIS TEST CHECKS THE STRIP SYNC FUNCTION
1803      ;: OF THE RECEIVER LOGIC
1804      ;: MODE: ISYMOD
1805      ;: LENGTH: EIGHT
1806      ;: NOTE: RXDONE SHOULD NEVER ASSERT
1807      ;: CHAR: 26 (SYNC)
1808
1809 006544 012767 000024 172354 TST20: MOV     #20,TSTNO      ;SAVE THIS
1810 006552 012767 006730 172336      MOV     #TST21,NEXT      ;GO TO THIS TEST WHEN THRU
1811 006560 052777 000400 010542      BIS     #MRESET,@TXCSR  ;MASTER RESET
1812 006566 012777 000000 010530      MOV     #ISYMOD,@PARCSR ;SET THE MODE
1813 006574 052777 000400 010526      BIS     #MRESET,@TXCSR  ;MASTER RESET
1814
1815      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1816 006602 012777 064001 010520      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1817
1818      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1819 006610 012777 006026 010506      MOV     #ISYMOD!EIGHT!NOPAR!26,@PARCSR
1820 006616 052777 000020 010470      BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
    
```

K03

```

1821                                     ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1822 006624 042777 020000 010476      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1823 006632 052777 020000 010470      BIS   #CLK,@TXCSR      ;POKE CLK UP
1824                                     ;POKE CLK TO GET LOGIC INTO SYNCRIZATION
1825 006640 042777 020000 010462      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1826 006646 052777 020000 010454      BIS   #CLK,@TXCSR      ;POKE CLK UP
1827 006654 052777 000400 010432      BIS   #STPSYN,@RXCSR   ;SET STRIP SYNC
1828 006662 012767 000003 172252      MOV   #3,COUNT         ;# OF SYNC CHARS
1829 006670 012767 001054 172246      1$:  MOV   #1054,TEMP1   ;CHAR TO BE SHIFTED
1830 006676 012767 000012 172234      MOV   #10,SHIFT        ;# OF SHIFTS
1831 006704 004767 010110                JSR   PC,RPOKE          ;SHIFT IN THIS CHAR
1832 006710 105777 010400                TSTB  @RXCSR ;RXDONE ?
1833 006714 100001                BPL   2$
1834 006716 104000                HLT
1835                                     2$:
1836 006720 005367 172216                DEC   COUNT ;# OF SYNC CHARS
1837 006724 001361                BNE   1$
1838 006726 104400                SCOPE
1839                                     ;; THIS TEST CHECKS THE STRIP SYNC FUNCTION
1840                                     ;; OF THE RECEIVER LOGIC
1841                                     ;; MODE:SYNEXT
1842                                     ;; LENGTH:FIVE
1843                                     ;; NOTE: RXDONE SHOULD NEVER ASSERT
1844                                     ;; CHAR: 26 (SYNC)
1845
1846 006730 012767 000025 172170      TST21: MOV   #21,TSTNO    ;SAVE THIS
1847 006736 012767 007100 172152      MOV   #TST22,NEXT      ;GO TO THIS TEST WHEN THRU
1848 006744 052777 000400 010356      BIS   #MRESET,@TXCSR   ;MASTER RESET
1849 006752 012777 020000 010344      MOV   #SYNEXT,@PARCSR  ;SET THE MODE
1850 006760 052777 000400 010342      BIS   #MRESET,@TXCSR   ;MASTER RESET
1851
1852                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1853 006766 012777 064001 010334      MOV   #MTDATA!CLK!MINT!BREAK,@TXCSR
1854
1855                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1856 006774 012777 020026 010322      MOV   #SYNEXT!FIVE!NOPAR!26,@PARCSR
1857 007002 052777 000020 010304      BIS   #SYNSCH,@RXCSR   ;SET SEARCH SYNC
1858                                     ;POKE CLK TO GET LOGIC INTO SYNCRIZATION
1859 007010 042777 020000 010312      BIC   #CLK,@TXCSR      ;POKE CLK DOWN
1860 007016 052777 020000 010304      BIS   #CLK,@TXCSR      ;POKE CLK UP
1861 007024 052777 000400 010262      BIS   #STPSYN,@RXCSR   ;SET STRIP SYNC
1862 007032 012767 000003 172102      MOV   #3,COUNT         ;# OF SYNC CHARS
1863 007040 012767 000026 172076      1$:  MOV   #26,TEMP1        ;CHAR TO BE SHIFTED
1864 007046 012767 000005 172064      MOV   #5,SHIFT         ;# OF SHIFTS
1865 007054 004767 007740                JSR   PC,RPOKE          ;SHIFT IN THIS CHAR
1866 007060 105777 010230                TSTB  @RXCSR ;RXDONE ?
1867 007064 100001                BPL   2$
1868 007066 104000                HLT
1869                                     2$:
1870 007070 005367 172046                DEC   COUNT ;# OF SYNC CHARS
1871 007074 001361                BNE   1$
1872 007076 104400                SCOPE
1873                                     ;; THIS TEST CHECKS THE STRIP SYNC FUNCTION
1874                                     ;; OF THE RECEIVER LOGIC
1875                                     ;; MODE:SYNEXT
1876                                     ;; LENGTH:SIX

```



```

1877      ;;NOTE: RXDONE SHOULD NEVER ASSERT
1878      ;;CHAR: 26 (SYNC)
1879
1880 007100 012767 000026 172020 TST22: MOV    #22,TSTNO      ;SAVE THIS
1881 007106 012767 007250 172002      MOV    #TST23,NEXT      ;GO TO THIS TEST WHEN THRU
1882 007114 052777 000400 010206      BIS    #MRESET,@TXCSR  ;MASTER RESET
1883 007122 012777 020000 010174      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1884 007130 052777 000400 010172      BIS    #MRESET,@TXCSR  ;MASTER RESET
1885
1886      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1887 007136 012777 064001 010164      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1888
1889      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1890 007144 012777 022026 010152      MOV    #SYNEXT!SIX!NOPAR!26,@PARCSR
1891 007152 052777 000020 010134      BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1892      ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1893 007160 042777 020000 010142      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1894 007166 052777 020000 010134      BIS    #CLK,@TXCSR     ;POKE CLK UP
1895 007174 052777 000400 010112      BIS    #STPSYN,@RXCSR  ;SET STRIP SYNC
1896 007202 012767 000003 171732      MOV    #3,COUNT        ;# OF SYNC CHARS
1897 007210 012767 000026 171726 1$:  MOV    #26,TEMP1       ;CHAR TO BE SHIFTED
1898 007216 012767 000006 171714      MOV    #6,SHIFT        ;# OF SHIFTS
1899 007224 004767 007570      JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
1900 007230 105777 010060      TSTB   @RXCSR ;RXDONE ?
1901 007234 100001      BPL    2$
1902 007236 104000      HLT
1903      2$:
1904 007240 005367 171676      DEC    COUNT ;# OF SYNC CHARS
1905 007244 001361      BNE    1$
1906 007246 104400      SCOPE
1907      ;;THIS TEST CHECKS THE STRIP SYNC FUNCTION
1908      ;;OF THE RECEIVER LOGIC
1909      ;;MODE:SYNEXT
1910      ;;LENGTH:SEVEN
1911      ;;NOTE: RXDONE SHOULD NEVER ASSERT
1912      ;;CHAR: 26 (SYNC)
1913
1914 007250 012767 000027 171650 TST23: MOV    #23,TSTNO      ;SAVE THIS
1915 007256 012767 007420 171632      MOV    #TST24,NEXT      ;GO TO THIS TEST WHEN THRU
1916 007264 052777 000400 010036      BIS    #MRESET,@TXCSR  ;MASTER RESET
1917 007272 012777 020000 010024      MOV    #SYNEXT,@PARCSR ;SET THE MODE
1918 007300 052777 000400 010022      BIS    #MRESET,@TXCSR  ;MASTER RESET
1919
1920      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1921 007306 012777 064001 010014      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1922
1923      ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1924 007314 012777 024026 010002      MOV    #SYNEXT!SEVEN!NOPAR!26,@PARCSR
1925 007322 052777 000020 007764      BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
1926      ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1927 007330 042777 020000 007772      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1928 007336 052777 020000 007764      BIS    #CLK,@TXCSR     ;POKE CLK UP
1929 007344 052777 000400 007742      BIS    #STPSYN,@RXCSR  ;SET STRIP SYNC
1930 007352 012767 000003 171562      MOV    #3,COUNT        ;# OF SYNC CHARS
1931 007360 012767 000026 171556 1$:  MOV    #26,TEMP1       ;CHAR TO BE SHIFTED
1932 007366 012767 000007 171544      MOV    #7,SHIFT        ;# OF SHIFTS
  
```



```

1989 007626 012777 064001 007474      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1990
1991                                     ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1992 007634 012777 030026 007462      MOV      #SYNINT!FIVE!NOPAR!26,@PARCSR
1993 007642 052777 000020 007444      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1994                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1995 007650 042777 020000 007452      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1996 007656 052777 020000 007444      BIS      #CLK,@TXCSR ;POKE CLK UP
1997                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1998 007664 042777 020000 007436      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1999 007672 052777 020000 007430      BIS      #CLK,@TXCSR ;POKE CLK UP
2000 007700 052777 000400 007406      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2001 007706 012767 000003 171226      MOV      #3,COUNT ;# OF SYNC CHARS
2002 007714 012767 000026 171222 1$:      MOV      #26,TEMP1 ;CHAR TO BE SHIFTED
2003 007722 012767 000005 171210      MOV      #5,SHIFT ;# OF SHIFTS
2004 007730 004767 007064      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2005 007734 105777 007354      TSTB    @RXCSR ;RXDONE ?
2006 007740 100001      BPL     2$
2007 007742 104000      HLT
2008                                     2$:
2009 007744 005367 171172      DEC     COUNT ;# OF SYNC CHARS
2010 007750 001361      BNE     1$
2011 007752 104400      SCOPE
2012                                     ;; THIS TEST CHECKS THE STRIP SYNC FUNCTION
2013                                     ;; OF THE RECEIVER LOGIC
2014                                     ;; MODE:SYNINT
2015                                     ;; LENGTH:SIX
2016                                     ;; NOTE: RXDONE SHOULD NEVER ASSERT
2017                                     ;; CHAR: 26 (SYNC)
2018
2019 007754 012767 000032 171144  TST26:  MOV     #26,TSTNO ;SAVE THIS
2020 007762 012767 010140 171126      MOV     #TST27,NEXT ;GO TO THIS TEST WHEN THRU
2021 007770 052777 000400 007332      BIS     #MRESET,@TXCSR ;MASTER RESET
2022 007776 012777 030000 007320      MOV     #SYNINT,@PARCSR ;SET THE MODE
2023 010004 052777 000400 007316      BIS     #MRESET,@TXCSR ;MASTER RESET
2024
2025                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2026 010012 012777 064001 007310      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2027
2028                                     ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2029 010020 012777 032026 007276      MOV     #SYNINT!SIX!NOPAR!26,@PARCSR
2030 010026 052777 000020 007260      BIS     #SYNSCH,@RXCSR ;SET SYNC SEARCH
2031                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2032 010034 042777 020000 007266      BIC     #CLK,@TXCSR ;POKE CLK DOWN
2033 010042 052777 020000 007260      BIS     #CLK,@TXCSR ;POKE CLK UP
2034                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2035 010050 042777 020000 007252      BIC     #CLK,@TXCSR ;POKE CLK DOWN
2036 010056 052777 020000 007244      BIS     #CLK,@TXCSR ;POKE CLK UP
2037 010064 052777 000400 007222      BIS     #STPSYN,@RXCSR ;SET STRIP SYNC
2038 010072 012767 000003 171042      MOV     #3,COUNT ;# OF SYNC CHARS
2039 010100 012767 000026 171036 1$:      MOV     #26,TEMP1 ;CHAR TO BE SHIFTED
2040 010106 012767 000006 171024      MOV     #6,SHIFT ;# OF SHIFTS
2041 010114 004767 006700      JSR     PC,RPOKE ;SHIFT IN THIS CHAR
2042 010120 105777 007170      TSTB   @RXCSR ;RXDONE ?
2043 010124 100001      BPL     2$
2044 010126 104000      HLT     ;RXDONE SHOULD NOT BE ASSERTED
    
```



```

2045 010130          2$:
2046 010130 005367 171006      DEC    COUNT    ;# OF SYNC CHARS
2047 010134 001361          BNE    1$
2048 010136 104400          SCOPE
2049          ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2050          ;:OF THE RECEIVER LOGIC
2051          ;:MODE:SYNINT
2052          ;:LENGTH:SEVEN
2053          ;:NOTE: RXDONE SHOULD NEVER ASSERT
2054          ;:CHAR: 26 (SYNC)
2055          ;:
2056 010140 012767 000033 170760 TST27: MOV    #27,TSTNO    ;SAVE THIS
2057 010146 012767 010324 170742      MOV    #TST28,NEXT    ;GO TO THIS TEST WHEN THRU
2058 010154 052777 000400 007146      BIS    #MRESET,@TXCSR ;MASTER RESET
2059 010162 012777 030000 007134      MOV    #SYNINT,@PARCSR ;SET THE MODE
2060 010170 052777 000400 007132      BIS    #MRESET,@TXCSR ;MASTER RESET
2061
2062          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2063 010176 012777 064001 007124      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2064
2065          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2066 010204 012777 034026 007112      MOV    #SYNINT!SEVEN!NOPAR!26,@PARCSR
2067 010212 052777 000020 007074      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2068          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2069 010220 042777 020000 007102      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
2070 010226 052777 020000 007074      BIS    #CLK,@TXCSR    ;POKE CLK UP
2071          ;POKE CLK TO GET LOGIC INTO SYNCROIZATION
2072 010234 042777 020000 007066      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
2073 010242 052777 020000 007060      BIS    #CLK,@TXCSR    ;POKE CLK UP
2074 010250 052777 000400 007036      BIS    #STPSYN,@RXCSR ;SET STRIP SYNC
2075 010256 012767 000003 170656      MOV    #3,COUNT      ;# OF SYNC CHARS
2076 010264 012767 000026 170652      1$:  MOV    #26,TEMP1     ;CHAR TO BE SHIFTED
2077 010272 012767 000007 170640      MOV    #7,SHIFT      ;# OF SHIFTS
2078 010300 004767 006514          JSR    PC,RPOKE       ;SHIFT IN THIS CHAR
2079 010304 105777 007004          TSTB   @RXCSR ;RXDONE ?
2080 010310 100001          BPL    2$
2081 010312 104000          HLT
2082          2$:
2083 010314 005367 170622      DEC    COUNT    ;# OF SYNC CHARS
2084 010320 001361          BNE    1$
2085 010322 104400          SCOPE
2086          ;:THIS TEST CHECKS THE STRIP SYNC FUNCTION
2087          ;:OF THE RECEIVER LOGIC
2088          ;:MODE:SYNINT
2089          ;:LENGTH:EIGHT
2090          ;:NOTE: RXDONE SHOULD NEVER ASSERT
2091          ;:CHAR: 26 (SYNC)
2092          ;:
2093 010324 012767 000034 170574 TST28: MOV    #28,TSTNO    ;SAVE THIS
2094 010332 012767 010510 170556      MOV    #TST29,NEXT    ;GO TO THIS TEST WHEN THRU
2095 010340 052777 000400 006762      BIS    #MRESET,@TXCSR ;MASTER RESET
2096 010346 012777 030000 006750      MOV    #SYNINT,@PARCSR ;SET THE MODE
2097 010354 052777 000400 006746      BIS    #MRESET,@TXCSR ;MASTER RESET
2098
2099          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2100 010362 012777 064001 006740      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR

```


2101					
2102					;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2103	010370	012777	036026	006726	MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2104	010376	052777	000020	006710	BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2105					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2106	010404	042777	020000	006716	BIC #CLK,@TXCSR ;POKE CLK DOWN
2107	010412	052777	020000	006710	BIS #CLK,@TXCSR ;POKE CLK UP
2108					;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2109	010420	042777	020000	006702	BIC #CLK,@TXCSR ;POKE CLK DOWN
2110	010426	052777	020000	006674	BIS #CLK,@TXCSR ;POKE CLK UP
2111	010434	052777	000400	006652	BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2112	010442	012767	000003	170472	MOV #3,COUNT ;# OF SYNC CHARS
2113	010450	012767	000026	170466	15: MOV #26,TEMP1 ;CHAR TO BE SHIFTED
2114	010456	012767	000010	170454	MOV #8,SHIFT ;# OF SHIFTS
2115	010464	004767	006330		JSR PC,RPOKE ;SHIFT IN THIS CHAR
2116	010470	105777	006620		TSTB @RXCSR ;RXDONE ?
2117	010474	100001			BPL 25
2118	010476	104000			HLT ;RXDONE SHOULD NOT BE ASSERTED
2119	010500				25:
2120	010500	005367	170436		DEC COUNT ;# OF SYNC CHARS
2121	010504	001361			BNE 15
2122	010506	104400			SCOPE
2123					::THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
2124					::WHILE IN STRIP SYNC MODE
2125					::THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2126					::STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2127					::... BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2128					::IS DISCOMBOBULATED
2129					::IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2130					::NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2131					::PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT.
2132					::BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
2133					::MODE: ISOC (ISYMOD)
2134					::LENGTH: EIGHT
2135					::PARITY: EVEPAR
2136					::CHARACTER EXPECTED:26
2137					::CHARACTER SENT: SYNC CHARACTER
2138					::NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2139					
2140	010510	012767	000035	170410	↑ST29: MOV #29,TSTNO ;SAVE THIS
2141	010516	012767	011112	170372	MOV #TST30,NEXT ;GO TO THIS TEST WHEN THRU
2142	010524	052777	000400	006576	BIS #MRESET,@TXCSR ;MASTER RESET
2143	010532	012777	000000	006564	MOV #ISYMOD,@PARCSR ;SET THE MODE
2144	010540	052777	000400	006562	BIS #MRESET,@TXCSR ;MASTER RESET
2145					
2146					;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2147	010546	012777	064001	006554	MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2148					
2149					;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2150	010554	012777	007426	006542	MOV #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
2151	010562	016703	006532		MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2152	010566	012767	000003	170346	MOV #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
2153	010574	052777	000020	006512	BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2154					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2155	010602	042777	020000	006520	BIC #CLK,@TXCSR ;POKE CLK DOWN
2156	010610	052777	020000	006512	BIS #CLK,@TXCSR ;POKE CLK UP

```

2157 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2158 010616 042777 020000 006504 BIC #CLK, JTXCSR ;POKE CLK DOWN
2159 010624 052777 020000 006476 BIS #CLK, JTXCSR ;POKE CLK UP
2160 010632 052777 000400 006454 BIS #STPSYN, JRXCSR ;SET STRIP SYNC
2161 010640 012767 000013 170272 2$: MOV #11, SHIFT ;# OF SHIFTS
2162 010646 012767 003054 170270 MOV #3054, TEMP1 ;SYNC CHAR + START&STOP+ PARITY
2163 010654 004767 006140 1$: JSR PC, RPOKE ;SHIFT IN THIS CHARACTER
2164 010660 105777 006430 TSTB JRXCSR ;RXDONE = 0 ?
2165 010664 100001 BPL 64$
2166 010666 104000 HLT ;RXDONE SHOULD NOT BE SET
2167 010670 64$:
2168 010670 005367 170246 DEC COUNT ;# OF SYNC CHARS
2169 010674 001361 BNE 2$ ;GO AGAIN ?
2170 010676 012700 000026 MOV #26, R0 ;EXPECTED
2171 010702 017701 006412 MOV JRXDBUF, R1 ;ACTUAL
2172 ;NOTE THAT THIS IS THE FIRST TIME
2173 ;RXDBUF IS READ.....THERE SHOULD BE
2174 ;NO OVER RUN ERRORS
2175 010706 020001 CMP R0, R1 ;COMPARE EXPECTED VS ACTUAL
2176 010710 001401 BEQ 65$
2177 010712 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2178 ;THERE SHOULD BE NO RXERR'S
2179 010714 65$:
2180 010714 012767 000004 170220 MOV #4, COUNT ;# OF TIMES
2181 010722 012700 110026 MOV JRXERR!PARER!26, R0 ;EXPECTED
2182 010726 012767 002054 170210 MOV #2054, TEMP1 ;BAD SYNC CHAR (WRONG PARITY)
2183 010734 012767 000013 170176 3$: MOV #11, SHIFT ;# OF SHIFTS
2184 010742 004767 006052 JSR PC, RPOKE ;SHIFT IN THIS CHAR
2185 010746 105777 006342 TSTB JRXCSR ;RXDONE = 1?
2186 010752 100401 BMI 66$
2187 010754 104000 HLT ;RXDONE SHOULD BE SET
2188 010756 66$:
2189 010756 017701 006336 MOV JRXDBUF, R1 ;ACTUAL DATA
2190 010762 020001 CMP R0, R1 ;COMPARE EXP VS ACT
2191 010764 001401 BEQ 67$
2192 010766 104002 HLT 2 ;DID THE RESPECTIVE ERROR STOP THE
2193 ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2194 ;.....CHECK THIS.....
2195 010770 67$:
2196 010770 005367 170146 DEC COUNT ;# OF SYNC CHARS
2197 010774 001445 BEQ 5$ ;FINISHED ? GET OUT OF TEST
2198 010776 022767 000003 170136 CMP #3, COUNT ;# OF SYNC CHARS
2199 011004 001423 BEQ 6$ ;CHECK FRAME ERROR ?
2200 011006 022767 000002 170126 CMP #2, COUNT ;# OF SYNC CHARS
2201 011014 001426 BEQ 7$ ;CHECK FRAME ERROR & BAD PARITY ?
2202 ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2203 011016 012767 000013 170114 MOV #11, SHIFT ;# OF SHIFTS
2204 011024 012767 000054 170112 MOV #54, TEMP1 ;FRAME & PARITY ERROR
2205 011032 004767 005762 JSR PC, RPOKE ;SHIFT IN THIS CHAR
2206 ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2207 011036 012767 000054 170100 MOV #54, TEMP1 ;FRAME & PARITY ERROR
2208 011044 012700 170026 MOV JRXERR!OVRUN!FRMERR!PARER!26, R0 ;EXPECTED
2209 011050 000167 177660 JMP 3$ ;DO IT AGAIN
2210 011054 012767 001054 170062 6$: MOV #1054, TEMP1 ;BAD STOP BIT FOR FRAME ERROR
2211 011062 012700 120026 MOV JRXERR!FRMERR!26, R0 ;EXPECTED
2212 011066 000167 177642 JMP 3$ ;DO IT AGAIN

```


E04

```

2213 011072 012767 000054 170044 7$:  MOV    #54,TEMP1      ;BAD STOP BIT & PARITY
2214 011100 012700 130026          MOV    #RXERR!FRMERR!PARER!26,RO  ;EXPECTED
2215 011104 000167 177624          JMP    3$           ;DO IT AGAIN
2216 011110 104400          5$:  SCOPE
2217          ;; THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
2218          ;; WHILE IN STRIP SYNC MODE
2219          ;; THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2220          ;; STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2221          ;; ... BUT IF AN ERROR SHOULD OCCUR... THIS AUTOMATIC RESET
2222          ;; IS DISCOMBOBULATED
2223          ;; IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2224          ;; NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2225          ;; PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT
2226          ;; BUT, IF AN RXERR OCCURS  RXDONE PLUS RXERR ARE ASSERTED
2227          ;; MODE: ISOC (ISYMOD)
2228          ;; LENGTH: SEVEN
2229          ;; PARITY: EVEPAR
2230          ;; CHARACTER EXPECTED: 226
2231          ;; NOTE THAT THE PARITY BIT SHOULD SHOW
2232          ;; UP IN THE DATA  IE. BIT SEVEN FOR
2233          ;; SEVEN LEVEL CODE
2234          ;; CHARACTER SENT: SYNC CHARACTER
2235          ;; NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2236
2237 011112 012767 000036 170006 †ST30: MOV    #30,TSTNO      ;SAVE THIS
2238 011120 012767 011514 167770          MOV    #TST31,NEXT      ;GO TO THIS TEST WHEN THRU
2239 011126 052777 000400 006174          BIS    #MRESET,@TXCSR  ;MASTER RESET
2240 011134 012777 000000 006162          MOV    #ISYMOD,@PARCSR ;SET THE MODE
2241 011142 052777 000400 006160          BIS    #MRESET,@TXCSR  ;MASTER RESET
2242
2243          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2244 011150 012777 064001 006152          MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2245
2246          ;SET MODE , # OF BITS,PARITY SENSE & LOAD SYNC REG
2247 011156 012777 005626 006140          MOV    #ISYMOD!SEVEN!EVEPAR!226,@PARCSR
2248 011164 016703 006130          MOV    RXDBUF,R3       ;SET UP FOR ERROR MSG
2249 011170 012767 000003 167744          MOV    #3,COUNT        ;# OF TIMES SYNC CHAR WILL BE SENT
2250 011176 052777 000020 006110          BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2251          ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2252 011204 042777 020000 006116          BIC    #CLK,@TXCSR     ;POKE CLK DOWN
2253 011212 052777 020000 006110          BIS    #CLK,@TXCSR     ;POKE CLK UP
2254          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2255 011220 042777 020000 006102          BIC    #CLK,@TXCSR     ;POKE CLK DOWN
2256 011226 052777 020000 006074          BIS    #CLK,@TXCSR     ;POKE CLK UP
2257 011234 052777 000400 006052          BIS    #STPSYN,@RXCSR ;SET STRIP SYNC
2258 011242 012767 000012 167670 2$:  MOV    #10,SHIFT      ;# OF SHIFTS
2259 011250 012767 001454 167666          MOV    #1454,TEMP1     ;SYNC CHAR + START&STOP+ PARITY
2260 011256 004767 005536          JSR    PC,RPOKE        ;SHIFT IN THIS CHARACTER
2261 011262 105777 006026          TSTB   @RXCSR ;RXDONE = 0 ?
2262 011266 100001          BPL    64$
2263 011270 104000          HLT
2264          64$:
2265 011272 005367 167644          DEC    COUNT           ;# OF SYNC CHARS
2266 011276 001361          BNE    2$             ;GO AGAIN ?
2267 011300 012700 000226          MOV    #226,RO        ;EXPECTED
2268 011304 017701 006010          MOV    @RXDBUF,R1     ;ACTUAL
  
```

F04

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 47
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0044

```

2269                                     ;NOTE THAT THIS IS THE FIRST TIME
2270                                     ;RXDBUF IS READ.....THERE SHOULD BE
2271                                     ;NO OVER RUN ERRORS
2272 011310 020001          CMP          RO,R1          ;COMPARE EXPECTED VS ACTUAL
2273 011312 071401          BEQ          65$
2274 011314 104002          HLT          2
2275                                     ;DATA CHARS SHOULD COMPARE
2276 011316                                     ;THERE SHOULD BE NO RXERR'S
2277 011316 012767 000004 167616 65$:      MOV          #4,COUNT          ;# OF TIMES
2278 011324 012700 110026          MOV          #RXERR!PARER!26,RO          ;EXPECTED
2279 011330 012767 001054 167606          MOV          #1054,TEMP1          ;BAD SYNC CHAR (WRONG PARITY)
2280 011336 012767 000012 167574 3$:      MOV          #10,SHIFT          ;# OF SHIFTS
2281 011344 004767 005450          JSR          PC,RPOKE          ;SHIFT IN THIS CHAR
2282 011350 105777 005740          TSTB         #RXCSR ;RXDONE = 1?
2283 011354 100401          BMI          66$
2284 011356 104000          HLT
2285 011360                                     66$:
2286 011360 017701 005734          MOV          #RXDBUF,R1          ;ACTUAL DATA
2287 011364 020001          CMP          RO,R1          ;COMPARE EXP VS ACT
2288 011366 001401          BEQ          67$
2289 011370 104002          HLT          2
2290                                     ;DID THE RESPECTIVE ERROR STOP THE
2291                                     ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2292 011372                                     ;.....CHECK THIS.....
2293                                     67$:
2294                                     ;NOTE THAT THE PARITY BIT SHOULD
2295                                     ;SHOW UP IN THE DATA
2296 011372 005367 167544          DEC          COUNT          ;# OF SYNC CHARS
2297 011376 001445          BEQ          5$          ;FINISHED ? GET OUT OF TEST
2298 011400 022767 000003 167534          CMP          #3,COUNT          ;# OF SYNC CHARS
2299 011406 001423          BEQ          6$          ;CHECK FRAME ERROR ?
2300 011410 022767 000002 167524          CMP          #2,COUNT          ;# OF SYNC CHARS
2301 011416 001426          BEQ          7$          ;CHECK FRAME ERROR & BAD PARITY ?
2302                                     ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2303 011420 012767 000012 167512          MOV          #10,SHIFT          ;# OF SHIFTS
2304 011426 012767 000054 167510          MOV          #54,TEMP1          ;FRAME & PARITY ERROR
2305 011434 004767 005360          JSR          PC,RPOKE          ;SHIFT IN THIS CHAR
2306                                     ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2307 011440 012767 000054 167476          MOV          #54,TEMP1          ;FRAME & PARITY ERROR
2308 011446 012700 170026          MOV          #RXERR!OVRUN!FRMERR!PARER!26,RO          ;EXPECTED
2309 011452 000167 177660          JMP          3$          ;DO IT AGAIN
2310 011456 012767 000454 167460 6$:      MOV          #454,TEMP1          ;BAD STOP BIT FOR FRAME ERROR
2311 011464 012700 120226          MOV          #RXERR!FRMERR!226,RO          ;EXPECTED
2312 011470 000167 177642          JMP          3$          ;DO IT AGAIN
2313 011474 012767 000054 167442 7$:      MOV          #54,TEMP1          ;BAD STOP BIT & PARITY
2314 011502 012700 130026          MOV          #RXERR!FRMERR!PARER!26,RO          ;EXPECTED
2315 011506 000167 177624          JMP          3$          ;DO IT AGAIN
2316 011512 104400          5$:      SCOPE
2317                                     ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
2318                                     ;;WHILE IN STRIP SYNC MODE
2319                                     ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2320                                     ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2321                                     ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2322                                     ;;IS DISCOMBOBULATED
2323                                     ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2324                                     ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS

```



```

2325      ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT.
2326      ;;BUT, IF AN RXERR OCCURS  RXDONE PLUS RXERR ARE ASSERTEED
2327      ;;MODE: ISOC (ISYMOD)
2328      ;;LENGTH: SIX
2329      ;;PARITY: EVEPAR
2330      ;;CHARACTER EXPECTED:126
2331
2332      ;;NOTE THAT THE PARITY BIT SHOULD SHOW
2333      ;;UP IN THE DATA  IE. BIT SIX FOR
2334      ;;SIX LEVEL CODE.
2335      ;;CHARACTER SENT: SYNC CHARACTER
2336      ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2337 011514 012767 000037 167404 †TST31: MOV      #31,TSTNO      ;SAVE THIS
2338 011522 012767 012116 167366      MOV      #TST32,NEXT      ;GO TO THIS TEST WHEN THRU
2339 011530 052777 000400 005572      BIS      #MRESET,@TXCSR  ;MASTER RESET
2340 011536 012777 000000 005560      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2341 011544 052777 000400 005556      BIS      #MRESET,@TXCSR  ;MASTER RESET
2342
2343      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2344 011552 012777 064001 005550      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2345
2346      ;SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
2347 011560 012777 003526 005536      MOV      #ISYMOD!SIX!EVEPAR!126,@PARCSR
2348 011566 016703 005526      MOV      RXDBUF,R3      ;SET UP FOR ERROR MSG
2349 011572 012767 000003 167342      MOV      #3,COUNT      ;# OF TIMES SYNC CHAR WILL BE SENT
2350 011600 052777 000020 005506      BIS      #SYNSCH,@RXCSR  ;SET SYNC SEARCH
2351      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2352 011606 042777 020000 005514      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
2353 011614 052777 020000 005506      BIS      #CLK,@TXCSR    ;POKE CLK UP
2354      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2355 011622 042777 020000 005500      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
2356 011630 052777 020000 005472      BIS      #CLK,@TXCSR    ;POKE CLK UP
2357 011636 052777 000400 005450      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2358 011644 012767 000011 167266 25:  MOV      #9,SHIFT      ;# OF SHIFTS
2359 011652 012767 000654 167264      MOV      #654,TEMP1    ;SYNC CHAR + START&STOP+ PARITY
2360 011660 004767 005134 15:  JSR      PC,RPOKE      ;SHIFT IN THIS CHARACTER
2361 011664 105777 005424      TSTB    @RXCSR ;RXDONE = 0 ?
2362 011670 100001      BPL     64$
2363 011672 104000      HLT
2364 011674      64$:
2365 011674 005367 167242      DEC     COUNT ;# OF SYNC CHARS
2366 011700 001361      BNE     25 ;GO AGAIN ?
2367 011702 012700 000126      MOV     #126,R0 ;EXPECTED
2368 011706 017701 005406      MOV     @RXDBUF,R1 ;ACTUAL
2369
2370      ;NOTE THAT THIS IS THE FIRST TIME
2371      ;RXDBUF IS READ.....THERE SHOULD BE
2372      ;NO OVER RUN ERRORS
2373 011712 020001      CMP     R0,R1 ;COMPARE EXPECTED VS ACTUAL
2374 011714 001401      BEQ     65$
2375 011716 104002      HLT     2 ;DATA CHARS SHOULD COMPARE
2376      ;THERE SHOULD BE NO RXERR'S
2377 011720 012767 000004 167214 65$:  MOV     #4,COUNT ;# OF TIMES
2378 011726 012700 110026      MOV     #RXERR!PARER!26,R0 ;EXPECTED
2379 011732 012767 000454 167204      MOV     #454,TEMP1 ;BAD SYNC CHAR (WRONG PARITY)
2380 011740 012767 000011 167172 35:  MOV     #9,SHIFT ;# OF SHIFTS
    
```

```

2381 011746 004767 005046      JSR      PC,RPOKE      ;SHIFT IN THIS CHAR
2382 011752 105777 005336      TSTB    @RXCSR      ;RXDONE = 1?
2383 011756 100401                BMI     66$
2384 011760 104000                HLT     ;RXDONE SHOULD BE SET
2385 011762                66$:
2386 011762 017701 005332      MOV     @RXDBUF,R1    ;ACTUAL DATA
2387 011766 020001                CMP     R0,R1        ;COMPARE EXP VS ACT
2388 011770 001401                BEQ     67$
2389 011772 104002                HLT     2            ;DID THE RESPECTIVE ERROR STOP THE
2390                                ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2391                                ;.....CHECK THIS.....
2392 011774                67$:
2393                                ;NOTE THAT THE PARITY BIT SHOULD
2394                                ;SHOW UP IN THE DATA
2395                                ;IE. BIT SIX FOR SIX LEVEL CODE
2396 011774 005367 167142      DEC     COUNT        ;# OF SYNC CHARS
2397 012000 001445                BEQ     5$           ;FINISHED ? GET OUT OF TEST
2398 012002 022767 000003 167132  CMP     #3,COUNT     ;# OF SYNC CHARS
2399 012010 001423                BEQ     6$           ;CHECK FRAME ERROR ?
2400 012012 022767 000002 167122  CMP     #2,COUNT     ;# OF SYNC CHARS
2401 012020 001426                BEQ     7$           ;CHECK FRAME ERROR & BAD PARITY ?
2402                                ;NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE....
2403 012022 012767 000011 167110  MOV     #9,SHIFT     ;# OF SHIFTS
2404 012030 012767 000054 167106  MOV     #54,TEMP1    ;FRAME & PARITY ERROR
2405 012036 004767 004756      JSR     PC,RPOKE     ;SHIFT IN THIS CHAR
2406                                ;NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
2407 012042 012767 000054 167074  MOV     #54,TEMP1    ;FRAME & PARITY ERROR
2408 012050 012700 170026      MOV     @RXERR!OVRRUN!FRMERR!PARER!26,R0 ;EXPECTED
2409 012054 000167 177660      JMP     3$           ;DO IT AGAIN
2410 012060 012767 000254 167056  6$:    MOV     #254,TEMP1   ;BAD STOP BIT FOR FRAME ERROR
2411 012066 012700 120126      MOV     @RXERR!FRMERR!126,R0 ;EXPECTED
2412 012072 000167 177642      JMP     3$           ;DO IT AGAIN
2413 012076 012767 000054 167040  7$:    MOV     #54,TEMP1   ;BAD STOP BIT & PARITY
2414 012104 012700 130026      MOV     @RXERR!FRMERR!PARER!26,R0 ;EXPECTED
2415 012110 000167 177624      JMP     3$           ;DO IT AGAIN
2416 012114 104400                5$:    SCOPE
2417                                ;;THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
2418                                ;;WHILE IN STRIP SYNC MODE
2419                                ;;THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
2420                                ;;STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
2421                                ;;.....BUT IF AN ERROR SHOULD OCCUR....THIS AUTOMATIC RESET
2422                                ;;IS DISCOMBOBULATED
2423                                ;;IE. FORCE PARITY ERROR WHILE STRIP SYNC IS SET
2424                                ;;NOTE: NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
2425                                ;;PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT....
2426                                ;;BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
2427                                ;;MODE: ISOC (ISYMOD)
2428                                ;;LENGTH: FIVE
2429                                ;;PARITY: EVEPAR
2430                                ;;CHARACTER EXPECTED:66
2431                                ;NOTE THAT THE PARITY BIT SHOULD SHOW
2432                                ;UP IN THE DATA IE. BIT FIVE FOR
2433                                ;FIVE LEVEL CODE
2434                                ;;CHARACTER SENT: SYNC CHARACTER
2435                                ;;NOTE: THIS TEST USES ONLY THE RECEIVER LOGIC
2436                                ;;
    
```



```

2437 012116 012767 000040 167002 TST32: MOV #32,TSTNO ;SAVE THIS
2438 012124 012767 012520 166764 MOV #.EOP,NEXT ;GO TO THIS TEST WHEN THRU
2439 012132 052777 000400 005170 BIS #MRESET,@TXCSR ;MASTER RESET
2440 012140 012777 000000 005156 MOV #ISYMOD,@PARCSR ;SET THE MODE
2441 012146 052777 000400 005154 BIS #MRESET,@TXCSR ;MASTER RESET
2442
2443 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2444 012154 012777 064001 005146 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2445
2446 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
2447 012162 012777 001466 005134 MOV #ISYMOD!FIVE!EVEPAR!66,@PARCSR
2448 012170 016703 005124 MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
2449 012174 012767 000003 166740 MOV #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
2450 012202 052777 000020 005104 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2451 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2452 012210 042777 020000 005112 BIC #CLK,@TXCSR ;POKE CLK DOWN
2453 012216 052777 020000 005104 BIS #CLK,@TXCSR ;POKE CLK UP
2454 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2455 012224 042777 020000 005076 BIC #CLK,@TXCSR ;POKE CLK DOWN
2456 012232 052777 020000 005070 BIS #CLK,@TXCSR ;POKE CLK UP
2457 012240 052777 000400 005046 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2458 012246 012767 000010 166664 2$: MOV #8,SHIFT ;# OF SHIFTS
2459 012254 012767 000354 166662 MOV #354,TEMP1 ;SYNC CHAR + START&STOP+ PARITY
2460 012262 004767 004532 1$: JSR PC,RPOKE ;SHIFT IN THIS CHARACTER
2461 012266 105777 005022 TSTB @RXCSR ;RXDONE = 0 ?
2462 012272 100001 BPL 64$
2463 012274 104000 HLT ;RXDONE SHOULD NOT BE SET
2464 012276 64$:
2465 012276 005367 166640 DEC COUNT ;# OF SYNC CHARS
2466 012302 001361 BNE 2$ ;GO AGAIN ?
2467 012304 012700 000066 MOV #66,R0 ;EXPECTED
2468 012310 017701 005004 MOV @RXDBUF,R1 ;ACTUAL
2469 ;NOTE THAT THIS IS THE FIRST TIME
2470 ;RXDBUF IS READ.....THERE SHOULD BE
2471 ;NO OVER RUN ERRORS
2472 012314 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
2473 012316 001401 BEQ 65$
2474 012320 104002 HLT 2 ;DATA CHARS SHOULD COMPARE
2475 ;THERE SHOULD BE NO RXERR'S
2476 012322 65$:
2477 012322 012767 000004 166612 MOV #4,COUNT ;# OF TIMES
2478 012330 012700 110026 MOV #RXERR!PARER!26,R0 ;EXPECTED
2479 012334 012767 000254 166602 MOV #254,TEMP1 ;BAD SYNC CHAR (WRONG PARITY)
2480 012342 012767 000010 166570 3$: MOV #8,SHIFT ;# OF SHIFTS
2481 012350 004767 004444 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2482 012354 105777 004734 TSTB @RXCSR ;RXDONE = 1?
2483 012360 100401 BMI 66$
2484 012362 104000 HLT ;RXDONE SHOULD BE SET
2485 012364 66$:
2486 012364 017701 004730 MOV @RXDBUF,R1 ;ACTUAL DATA
2487 012370 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2488 012372 001401 BEQ 67$
2489 012374 104002 HLT 2 ;DID THE RESPECTIVE ERROR STOP THE
2490 ;AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
2491 ;.....CHECK THIS.....
2492 012376 67$:
    
```



```

2517
2518 ;END OF PASS
2519 ;TYPE NAME OF TEST
2520 ;UPDATE PASS COUNT
2521 ;CHECK FOR EXIT TO ACT-11
2522 ;RESTART TEST
2523
2524 012520 104402 .EOP: TYPE ;TYPE NAME OF TEST
2525 012522 015662 MEPASS
2526 012524 104410 012756 CONVRT ,OUTCRY
2527 012530 104402 015403 TYPE ,DEVICE
2528 012534 105767 166442 TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
2529 012540 001511 BEQ CCC ;NO, JUMP AROUND
2530 012542 005767 166450 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
2531 012546 001007 BNE RUNIT ;YES
2532 012550 104402 015415 TYPE ,MCOV ;NO
2533 012554 016700 166436 MOV ,ACTREG,RO ;DISPLAY ACTREG
2534 012560 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
2535 ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2536 012562 000167 166472 JMP .START ;START OVER AGAIN..... YOU DESELECTED EVERYTHING
2537 012566 062767 000010 166410 RUNIT: ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2538 012574 062767 000010 166410 ZERO: ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
2539 012602 000241 CLC
2540 012604 006167 166410 ROL ROTADD ;UP DATE ROTATING POINTER
2541 012610 103410 BCS 2$ ;IS IT THE LAST DEVICE
2542 ;TO BE TESTED IN THIS PASS ?
2543 012612 036767 166402 166376 BIT ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2544 012620 001762 BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
2545 012622 004767 000034 JSR PC,REPLAY ;CALCULATE NEW PARAMETERS
2546 012626 000167 000174 JMP RESTRT ;YES IT WAS ACTIVE, TEST THIS DEVICE
2547 012632 012767 000001 166360 2$: MOV #1,ROTADD ;OK! NOW SET UP ROTATING
2548 ;POINTER FOR NEXT MULTIPLE PASS
2549 012640 016767 166342 166336 MOV KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2550 012646 016767 166342 166336 MOV KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
2551 012654 004767 000002 JSR PC,REPLAY ;CALC NEW PARAMETERS
2552 012660 000441 BR CCC ;JUMP AROUND REPLAY
2553 012662 016767 166316 004126 REPLAY: MOV BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
2554 012670 004767 003770 JSR PC,DUADDR ;CREATE NEW ADDRESSES
2555 012674 016767 166312 004436 MOV BASEIV,DURIV ;CREATE DURIV
2556 012702 062767 000002 166302 ADD #2,BASEIV
2557 012710 016767 166276 004424 MOV BASEIV,DURIS ;CREATE DURIS
2558 012716 062767 000002 166266 ADD #2,BASEIV
2559 012724 016767 166262 004412 MOV BASEIV,DUTIV ;CREATE DUTIV
2560 012732 062767 000002 166252 ADD #2,BASEIV
2561 012740 016767 166246 004400 MOV BASEIV,DUTIS ;CREATE DUTIS
2562 012746 016767 004366 166236 MOV DURIV,BASEIV ;RESTORE
2563 012754 000207 RTS PC
2564
2565 012756 000001 OUTCRY: 1
2566 012760 006 002 .BYTE 6,2
2567 012762 017314 RXCSR
2568
2569 012764 CCC:
2570 012764 005067 166144 CLR LSTERR ;CLEAR LAST ERROR PC
2571 012770 005067 166230 CLR ERRFLG ;CLEAR ERROR FLAG
2572 012774 005267 166130 INC PASCNT ;UPDATE PASS COUNT
    
```

```

2573 013000 016777 166124 166074      MOV      PASCNT,ALIGHTS      ;DISPLAY PASS COUNT
2574 013006 013701 000042              MOV      @#42,R1            ;CHECK FOR ACT-11 OR DDP
2575 013012 001405              BEQ      RESTRT              ;IF NOT, CONTINUE TESTING
2576 013014 000005              RESET
2577 013016 004711              LOGICAL: JSR      PC,(R1)
2578 013020 000240              NOP
2579 013022 000240              NOP
2580 013024 000240              NOP
2581 013026 012767 000340 164742  RESTRT: MOV      #340,PS      ;PREVENT INTERRUPTS (PRIO: 7)
2582 013034 104413              CKSWR      ;CHECK FOR †G
2583 013036 012767 002350 166050      MOV      #TST1,RTRN
2584 013044 000167 167300              JMP      TST1
2585
2586              ;SCOPE LOOP AND INTERATION HANDLER
2587
2588 013050      .SCOPE:
2589              ;**** START OF CODE FOR THE X OR TESTER *****
2590 013050 000424      BR      4$
2591
2592 013052 013746 000004      MOV      @#4,-(SP)          ;IF RUNNING ON THE X OR TESTER CHANGE
2593 013056 012737 013076 000004      MOV      #1$,@#4            ;THIS INSTRUCTION TO A "NOP"(NOP=240)
2594 013064 005737 177060              TST      @#177060          ;SAVE CONTENTS OF ERROR VECTOR
2595 013070 012637 000004      MOV      (SP)+,@#4          ;SET FOR TIME OUT
2596 013074 000404              BR      2$                  ;TIME OUT ON X OR ?
2597 013076 022626      1$:      CMP      (SP)+,(SP)+        ;RESTORE ERROR VECTOR
2598 013100 012637 000004      MOV      (SP)+,@#4          ;GO TO NEXT TEST
2599 013104 000403              BR      3$                  ;CLEAR THE STACK AFTER A TIMEOUT
2600 013106 016767 166004 166000  2$:      MOV      NEXT,RTRN          ;RESTORE ERROR VECTOR
2601 013114 016716 165774      3$:      MOV      RTRN,(SP)        ;LOOP ON PRESENT TEST
2602 013120 000002              RTI                          ;SET UP NEXT TEST IN RTRN
2603 013122              4$:      ;**** END OF CODE FOR THE X OR TESTER ***** ;SET UP STACK FOR RTI
2604 013122 104413              CKSWR      ;CHECK FOR †G
2605 013124 032777 040000 165746  TTST:  BIT      #SW14,@SWR          ;LOOP ON CURRENT TEST ?
2606 013132 001407              BEQ      1$
2607 013134 000432              BR      3$
2608 013136 105777 165742              TSTB     @TKCSR              ;TEST TTY FLAG
2609 013142 100027              BPL      3$
2610 013144 017700 165736              MOV      @TKDBR,RO
2611 013150 000412              BR      2$                  ;CLR DONE BIT
2612 013152 032777 004000 165720  1$:      BIT      #SW11,@SWR          ;IF A TTY KEY IS STRUCK GO TO NEXT TST
2613 013160 001006              BNE      2$                  ;INHIBIT ITERATIONS ?
2614 013162 005267 165736              INC      LPCNT
2615 013166 026767 165732 165726      CMP      LPCNT,ICOUNT        ;CHECK FOR ITERATION CNT FINISH
2616 013174 101412              BLOS     3$
2617 013176 105067 166022      2$:      CLRB     ERRFLG
2618 013202 005067 165716              CLR      LPCNT
2619 013206 012767 000005 165706      MOV      #5,ICOUNT          ;SET UP ITERATION COUNT
2620 013214 016767 165676 165672      MOV      NEXT,RTRN          ;SET UP NEXT TEST IN RTRN
2621 013222 016716 165666      3$:      MOV      RTRN,(SP)        ;SET UP STACK FOR RTI
2622 013226 000002              RTI
2623 013230 001407      BRW:    1407                ;RESTORE "BEQ 1$" INSTRUCTION
2624 013232 000432      BRX:    432                  ;RESTORE "BR 3$" INSTRUCTION
2625
2626              ;CHECK FOR FREEZE ON CURRENT DATA
2627
2628 013234 104413      .SCOPE1: CKSWR              ;CHECK FOR †G

```



```

2629 013236 032777 001000 165634      BIT      #SW09, @SWR
2630 013244 001402                BEQ      1$
2631 013246 016716 165646                MOV      LOCK, (SP)
2632 013252 000002                1$:     RTI
2633
2634                ;TELETYPE OUTPUT ROUTINE
2635
2636 013254 010546                .TYPE:  MOV      R5, -(SP)
2637 013256 017605 000002                MOV      @2(SP), R5
2638 013262 062766 000002 000002                ADD      #2, 2(SP)
2639 013270 105715                1$:     TSTB     (R5)                ;LOOK FOR "0"
2640 013272 001406                BEQ      3$
2641 013274 105777 165610                2$:     TSTB     @TPCSR                ;TEST DONE BIT
2642 013300 100375                BPL      2$
2643 013302 112577 165604                MOVB     (R5)+, @TPDBR                ;TYPE CHAR
2644 013306 000770                BR       1$                ;DO IT AGAIN UNTIL "0" IS SEEN
2645 013310 012605                3$:     MOV      (SP)+, R5
2646 013312 000002                RTI
2647
2648                ;ASCII STRING INPUT ROUTINE
2649
2650 013314 010346                .INSTR: MOV      R3, -(SP)
2651 013316 010446                MOV      R4, -(SP)
2652 013320 017667 000004 000010                MOV      @4(SP), .MSG
2653 013326 062766 000002 000004                ADD      #2, 4(SP)
2654 013334 104402                .INST1: TYPE
2655 013336 000000                .MSG:   0
2656 013340 012704 016450                MOV      #INBUF, R4
2657 013344 012703 000007                MOV      #7, R3
2658 013350 105777 165530                1$:     TSTB     @TKCSR
2659 013354 100375                BPL      1$
2660 013356 117714 165524                MOVB     @TKDBR, (R4)
2661 013362 142714 000200                BICB     #200, (R4)
2662 013366 121427 000025                CMPB     (R4), #25                ;IS IT <↑U>
2663 013372 001003                BNE      200$
2664 013374 104402 015572                TYPE, MCRLF
2665 013400 000755                BR       .INST1
2666 013402 122427 000015                200$:   CMPB     (R4)+, #15
2667 013406 001423                BEQ      INSTR2
2668 013410 117777 165472 165474                MOVB     @TKDBR, @TPDBR
2669 013416 105777 165466                2$:     TSTB     @TPCSR
2670 013422 100375                BPL      2$
2671 013424 005303                DEC      R3
2672 013426 001350                BNE      1$
2673 013430 000402                BR       .INSTG
2674 013432 010346                .INSTE: MOV      R3, -(SP)
2675 013434 010446                MOV      R4, -(SP)
2676 013436 104402                .INSTG: TYPE
2677 013440 015566                MQM
2678 013442 005737 014730                TST      @#RDSW
2679 013446 001402                BEQ      400$
2680 013450 104402 015572                TYPE, MCRLF
2681 013454 000727                400$:   BR       .INST1
2682 013456 012604                INSTR2: MOV      (SP)+, R4
2683 013460 012603                MOV      (SP)+, R3
2684 013462 000002                RTI
    
```

```

2685
2686
2687
2688 013464 010546
2689 013466 010446
2690 013470 016605 000004
2691 013474 012567 000170
2692 013500 012567 000166
2693 013504 012567 000164
2694 013510 112567 000162
2695 013514 112567 000157
2696 013520 010566 000004
2697 013524 005005
2698 013526 012704 016450
2699 013532 122714 000015
2700 013536 001420
2701 013540 121427 000060
2702 013544 002415
2703 013546 121427 000067
2704 013552 003012
2705 013554 142714 000060
2706 013560 152405
2707 013562 122714 000015
2708 013566 001414
2709 013570 006305
2710 013572 006305
2711 013574 006305
2712 013576 000760
2713 013600 122714 000015
2714 013604 001003
2715 013606 005737 014730
2716 013612 001023
2717 013614 104404
2718 013616 000742
2719
2720
2721
2722 013620 020567 000046
2723 013624 101365
2724 013626 020567 000036
2725 013632 103762
2726 013634 136705 000036
2727 013640 001357
2728
2729
2730
2731 013642 016704 000026
2732 013646 010524
2733 013650 062705 000002
2734 013654 105367 000017
2735 013660 001372
2736 013662 012604
2737 013664 012605
2738 013666 000002
2739 013670 000000
2740 013672 000000

```

;CONVERT ASCII STRING TO OCTAL

```

.PARAM: MOV R5, -(SP)
        MOV R4, -(SP)
        MOV 4(SP), R5
        MOV (R5)+, LOLIM
        MOV (R5)+, HILIM
        MOV (R5)+, DEVADR
        MOV (R5)+, LOBITS
        MOV (R5)+, ADCRCNT
        MOV R5, 4(SP)
PARAM1: CLR R5
        MOV #INBUF, R4
        CMPB #15, (R4)
        BEQ PARERR
1$:     CMPB (R4), #60
        BLT PARERR
        CMPB (R4), #67
        BGT PARERR
        BICB #60, (R4)
        BISB (R4)+, R5
        CMPB #15, (R4)
        BEQ LIMITS
        ASL R5
        ASL R5
        ASL R5
        BR 1$
PARERR: CMPB #15, (R4)
        BNE 120$
        TST @#RDSW
        BNE PARTI
120$:   INSTER
        BR PARAM1

```

;IS FIRST CHARACTER A <CR>

;IS CKSWR ROUTINE BEING USED

;TEST TO SEE IF NUMBER IS WITHIN LIMITS

```

LIMITS: CMP R5, HILIM
        BHI PARERR
        CMP R5, LOLIM
        BLO PARERR
        BITB LOBITS, R5
        BNE PARERR

```

;STORE NUMBER AT SPECIFIED ADDRESS

```

1$:     MOV DEVADR, R4
        MOV R5, (R4)+
        ADD #2, R5
        DECB ADCRCNT
        BNE 1$
PARTI:  MOV (SP)+, R4
        MOV (SP)+, R5
        RTI
LOLIM:  0
HILIM:  0

```



```

2741 013674 000000          DEVADR: 0
2742 013676 000000          LOBITS: 0
2743          013677          ADRCNT=LOBITS+1
2744
2745          ;SAVE PC OF TEST THAT FAILED AND RO-R5
2746
2747 013700 016667 000004 165266 .SAVOS: MOV    4(SP),SAVPC
2748
2749          ;SAVE RO-R5
2750
2751 013706 010567 165256      SVOS:  MOV    R5,SAVR5
2752 013712 010467 165250      MOV    R4,SAVR4
2753 013716 010367 165242      MOV    R3,SAVR3
2754 013722 010267 165234      MOV    R2,SAVR2
2755 013726 010167 165226      MOV    R1,SAVR1
2756 013732 010067 165220      MOV    R0,SAVR0
2757 013736 000002          RTI
2758
2759          ;RESTORE RO-R5
2760
2761 013740 016700 165212      .RESGS: MOV    SAVR0,R0
2762 013744 016701 165210      MOV    SAVR1,R1
2763 013750 016702 165206      MOV    SAVR2,R2
2764 013754 016703 165204      MOV    SAVR3,R3
2765 013760 016704 165202      MOV    SAVR4,R4
2766 013764 016705 165200      MOV    SAVR5,R5
2767 013770 000002          RTI
2768
2769          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2770
2771 013772 104402          .CONVR: TYPE
2772 013774 015572          MCRLF
2773 013776 010046          .CNVRT: MOV    R0,-(SP)
2774 014000 010146          MOV    R1,-(SP)
2775 014002 010346          MOV    R3,-(SP)
2776 014004 010446          MOV    R4,-(SP)
2777 014006 010546          MOV    R5,-(SP)
2778 014010 017601 000012      MOV    2(12(SP),R1
2779 014014 016767 002470 165126      MOV    TEMP,TEMP3
2780 014022 062766 000002 000012      ADD    #2,12(SP)
2781 014030 012167 000154          MOV    (R1)+,WRDCNT
2782 014034 112167 000152      15:  MOVB  (R1)+,CHRCNT
2783 014040 112167 000147          MOVB  (R1)+,SPACNT
2784 014044 013167 000144          MOV    2(R1)+,BINWRD
2785 014050 016704 000140      25:  MOV    BINWRD,R4
2786 014054 116705 000132          MOVB  CHRCNT,R5
2787 014060 012700 016510          MOV    #TEMP,R0
2788 014064 010403          35:  MOV    R4,R3
2789 014066 042703 177770          BIC    #177770,R3
2790 014072 062703 000060          ADD    #060,R3
2791 014076 110320          MOVB  R3,(R0)+
2792 014100 006204          ASR    R4
2793 014102 042704 100000          BIC    #100000,R4
2794 014106 006204          ASR    R4
2795 014110 006204          ASR    R4
2796 014112 005305          DEC    R5
    
```

```

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

2797	014114	001363				BNE	3\$	
2798	014116	012703	016550			MOV	#MDATA,R3	
2799	014122	114023			4\$:	MOVB	-(R0),(R3)+	
2800	014124	105367	000062			DECB	CHRCNT	
2801	014130	001374				BNE	4\$	
2802	014132	105767	000055			TSTB	SPACNT	
2803	014136	001405				BEQ	6\$	
2804	014140	112723	000040		5\$:	MOVB	#040,(R3)+	
2805	014144	105367	000043			DECB	SPACNT	
2806	014150	001373				BNE	5\$	
2807	014152	105013			6\$:	CLRB	(R3)	
2808	014154	104402				TYPE		
2809	014156	016550				MDATA		
2810	014160	005367	000024			DEC	WRDCNT	
2811	014164	001323				BNE	1\$	
2812	014166	016767	164756	002314		MOV	TEMP3,TEMP	
2813	014174	012605				MOV	(SP)+,R5	
2814	014176	012604				MOV	(SP)+,R4	
2815	014200	012603				MOV	(SP)+,R3	
2816	014202	012601				MOV	(SP)+,R1	
2817	014204	012600				MOV	(SP)+,R0	
2818	014206	000002				RTI		
2819	014210	000000				WRDCNT:	0	
2820	014212	000000				CHRCNT:	0	
2821		014213				SPACNT=	CHRCNT+1	
2822	014214	000000				BINWRD:	0	
2823								
2824								
2825								
2826								
2827								
2828								
2829	014216	017605	000000			.SETFLG:MOV	@(SP),R5	
2830	014222	122767	000116	002220		CMPB	#'N,INBUF	;IS IT "N" ?
2831	014230	001002				BNE	1\$	
2832	014232	105015				CLRB	(R5) ;000	
2833	014234	000406				BR	2\$	
2834	014236	122767	000131	002204	1\$:	CMPB	#'Y,INBUF	;IS IT "Y" ?
2835	014244	001005				BNE	3\$	
2836	014246	112715	177777			MOVB	#-1,(R5)	;377
2837	014252	062716	000002		2\$:	ADD	#2,(SP)	
2838	014256	000002				RTI		
2839	014260	104404			3\$:	INSTR		;RETRY
2840	014262	000755				BR	.SETFLG	
2841								
2842								
2843								
2844								
2845								
2846	014264	011646				.TRPSR:MOV	(SP),-(SP)	;GET PC OF RETURN
2847	014266	162716	000002			SUB	#2,(SP)	;=PC OF TRAP
2848	014272	017616	000000			MOV	@(SP),(SP)	;GET TRP
2849	014276	006316			TRPOK:	ASL	(SP)	;MULTIPLY TRAP ARG BY 2
2850	014300	042716	177001			BIC	#177001,(SP)	;CLEAR UNWANTED BITS
2851	014304	062716	001226			ADD	#.TRPTAB,(SP)	;POINTER TO SUBROUTINE ADDRESS
2852	014310	017616	000000			MOV	@(SP),(SP)	;SUBROUTINE ADDRESS


```

2853 014314 000136          JMP      2(SP)+          ;GO TO SUBROUTINE
2854
2855                          ;ERROR HANDLER
2856
2857 014316 104413          .HLT:   CKSWR          ;CHECK FOR 1G
2858 014320 032777 020000 164552  BIT      #SW13,2SWR    ;INHIBIT ERROR TYPE OUT ?
2859 014326 001061          BNE     HALTS
2860 014330 021667 164600  CMP     (SP),LSTERR
2861 014334 001404          BEQ     1$
2862 014336 011667 164572  MOV     (SP),LSTERR
2863 014342 105067 164656  CLRB   ERRFLG
2864 014346 104406          1$:    SAVOS
2865 014350 011605          MOV     (SP),R5
2866 014352 162705 000002  SUB     #2,R5
2867 014356 011504          MOV     (R5),R4
2868 014360 006304          ASL    R4
2869 014362 061504          ADD    (R5),R4
2870 014364 006304          ASL    R4
2871 014366 042704 177001  BIC    #177001,R4
2872 014372 062704 017264  ADD    #.ERRTAB,R4
2873 014376 012467 000040  MOV    (R4)+,ERRMSG
2874 014402 012467 000046  MOV    (R4)+,DATAHD
2875 014406 011467 000054  MOV    (R4),DATABP
2876 014412 105767 164606  TSTB   ERRFLG
2877 014416 001403          BEQ    TYPMSG
2878 014420 005767 000042  TST    DATABP
2879 014424 001014          BNE    TYPDAT
2880 014426 104410          TYPMSG: CONVRT
2881 014430 014560          ERTABO
2882 014432 112767 177777 164564  MOVB   #-1,ERRFLG
2883 014440 104402          TYPE
2884 014442 000000          ERRMSG: 0
2885 014444 005767 000004  TST    DATAHD
2886 014450 001402          BEQ    TYPDAT
2887 014452 104402          TYPE
2888 014454 000000          DATAHD: 0
2889 014456 005767 000004  TYPDAT: TST    DATABP
2890 014462 001402          BEQ    RESREG
2891 014464 104410          CONVRT
2892 014466 000000          DATABP: 0
2893 014470 104407          RESREG: RESOS
2894 014472 005777 164402  HALTS: TST    2SWR
2895 014476 100005          BPL    EXITER
2896 014500 010046          PUSHRO
2897 014502 016600 000002  MOV    2(SP),R0
2898 014506 000000          HALT
2899 014510 012600          POPRO
2900 014512 104413          EXITER: CKSWR          ;CHECK FOR 1G
2901 014514 005267 164412          INC    ERRCNT
2902 014520 032777 000400 164352  BIT    #SW08,2SWR    ;LOOP ON ERROR ?
2903 014526 001007          BNE    1$
2904 014530 032777 002000 164342  BIT    #SW10,2SWR    ;ESCAPE TO NEXT ON ERROR ?
2905 014536 001407          BEQ    2$
2906 014540 016767 164352 164346  MOV    NEXT,RTRN    ;SET UP FOR NEXT TEST
2907 014546 012706 001100          1$:   MOV    #STACK,SP    ;REINITIALIZE SP
2908 014552 000177 164336          JMP    2$RTRN
    
```

E05

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 59
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0056

```

2909 014556 000002          2S: RTI
2910 014560 000001          ERTABO: 1
2911 014562 006          002 .BYTE 6,2
2912 014564 001174          SAVPC
                                ;ENTER HERE ON POWER FAILURE
2913
2914
2915
2916 014566 010046          .PFAIL: MOV R0,-(SP) ;SAVE R0-R5 ON PROCESSOR STACK
2917 014570 010146          MOV R1,-(SP)
2918 014572 010246          MOV R2,-(SP)
2919 014574 010346          MOV R3,-(SP)
2920 014576 010446          MOV R4,-(SP)
2921 014600 010546          MOV R5,-(SP)
2922 014602 016746 163216  MOV 24,-(SP)
2923 014606 010667 164360  MOV SP,SAVSP ;SAVE STACK POINTER
2924 014612 012767 014624 163204 MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
2925 014620 000000          HALT ;HALT ON POWER DOWN NORMAL
2926 014622 000777          1S: BR 1S
2927
2928          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2929
2930 014624 016706 164342  RESTAR: MOV SAVSP,SP ;RESTORE STACK POINTER
2931 014630 012605          MOV (SP)+,R5 ;RESTORE R0-R5
2932 014632 012604          MOV (SP)+,R4
2933 014634 012603          MOV (SP)+,R3
2934 014636 012602          MOV (SP)+,R2
2935 014640 012601          MOV (SP)+,R1
2936 014642 012600          MOV (SP)+,R0
2937 014644 012767 014566 163152  MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
2938 014652 012767 000340 163116  MOV #340,PS
2939 014660 012706 001100          MOV #STACK,SP
2940 014664 005067 001620          CLR TEMP
2941 014670 005267 001614          1S: INC TEMP
2942 014674 001375          BNE 1S
2943 014676 104410          CONVRT
2944 014700 014722          PFTAB
2945 014702 104402          TYPE
2946 014704 015575          MPFAIL
2947 014706 005067 164312  CLR ERRFLG
2948 014712 005067 164216  CLR LSTERR
2949 014716 000177 164172  JMP #RTRN
2950 014722 000001          PFTAB: 1
2951 014724 006          002 .BYTE 6,2
2952 014726 001114          RTRN
2953
2954
2955          ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 1G TO ALLOW CHANGING
2956          ;OF LOC.176.
2957          ;LOCATIONS USED:
2958 014730 000000          RDSW: .WORD 0
2959
2960
2961 014732 005737 000042          .CKSWR: TST #42
2962 014736 001042          BNE OUT
2963 014740 022767 000176 164132  CMP #SWREG,SWR ;SOFTWARE SWITCH REGISTER PRESENT
2964 014746 001036          BNE OUT ;NO, GET OUT
  
```


2965	014750	105777	164130		
2966	014754	100033			
2967	014756	017767	164124	176352	
2968	014764	042767	177600	176344	
2969	014772	122767	000007	176336	
2970	015000	001021			
2971	015002	104402	015060		
2972	015006	005137	014730		
2973	015012	104402	015065		
2974	015016	104411	015052		
2975	015022	104403	015075		
2976	015026	104405			
2977	015030	000000			
2978	015032	177777			
2979	015034	000176			
2980	015036	000	001		
2981	015040	104402	015572		
2982	015044	005037	014730		
2983	015050	000002			
2984	015052	000001			
2985	015054	006	002		
2986	015056	000176			
2987	015060	005015	043536	000	
2988	015065	015	051412	051127	
2989	015072	020075	000		
2990	015075	040	047040	053505	
2991	015102	020075	000		
2992		015106			
2993	015106	005015	042012	030525	
2994	015114	020061	055104	052504	
2995	015122	026503	020102	040524	
2996	015130	042520	041440	006440	
2997	015136	000012			
2998	015140	005015	042526	052103	
2999	015146	051117	040440	042104	
3000	015154	042522	051523	000055	
3001	015162	005015	051461	020124	
3002	015170	042504	044526	042503	
3003	015176	020072	042522	042503	
3004	015204	053111	051105	041440	
3005	015212	047117	051124	046117	
3006	015220	051040	043505	051511	
3007	015226	042524	020122	042101	
3008	015234	051104	051505	026523	
3009	015242	000			
3010	015243	015	040412	042522	
3011	015250	054440	052517	051040	
3012	015256	047125	044516	043516	
3013	015264	046440	046125	044524	
3014	015272	046120	020105	042504	
3015	015300	044526	042503	020123	
3016	015306	020077	054450	047440	
3017	015314	020122	024516	000055	
3018	015322	005015	040514	052123	
3019	015330	042040	053105	041511	
3020	015336	035105	042522	042503	

```

TSTB      @TKCSR      ;YES, WAIT FOR
BPL       OUT         ;READY, GET CHARACTER
MOV       @TKDBR, .MSG ;AND STRIP OFF
BIC       #177600, .MSG ;THE GARBAGE
CMPB     #7, .MSG     ;IS IT A <IG>
BNE       OUT
          TYPE, $CNTG
.CNTLU:   COM         @#RDSW
          TYPE, $MSWR
          CNVRT, $WREGC
          INSTR, $MNEW
          PARAM
          0
          177777
          SWREG
.BYTE     0, 1
          TYPE, MCRLF
OUT:      CLR         @#RDSW
          RTI
SWREGC:   1
.BYTE     6, 2
          SWREG
$CNTG:    .ASCIZ <15><12>/IG/
$MSWR:    .ASCIZ <15><12>/SWR= /
$MNEW:    .ASCIZ / NEW= /
.EVEN
MTITLE:   .ASCIZ <15><12><12>/DU11 DZDUC-B TAPE C /<15><12>
MVECTO:   .ASCIZ <15><12>/VECTOR ADDRESS-/
MREGAD:   .ASCIZ <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/
MMULT:    .ASCIZ <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/
MLASTD:   .ASCIZ <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/

```

G05

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 61
DZDUCC.P11 10-JUL-74 00:00

SEQ 0058

3021	015344	053111	051105	041440	
3022	015352	047117	051124	046117	
3023	015360	051040	043505	051511	
3024	015366	042524	020122	042101	
3025	015374	051104	051505	026523	
3026	015402	000			
3027	015403	075	042504	044526	DEVICE: .ASCIZ /=DEVICE /
3028	015410	042503	020040	000	
3029	015415	015	044012	053517	MCOW: .ASCIZ <15><12>/HOW NOW BROWN COW? ...SELECT SOMETHING TO RUN JACTREG/
3030	015422	047040	053517	041040	
3031	015430	047522	047127	041440	
3032	015436	053517	020077	027056	
3033	015444	051456	046105	041505	
3034	015452	020124	047523	042515	
3035	015460	044124	047111	020107	
3036	015466	047524	051040	047125	
3037	015474	040040	041501	051124	
3038	015502	043505	000		
3039	015505	015	047412	052125	MRANGE: .ASCIZ <15><12>/OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-/ MGM: .ASCIZ / ?/ MCRLF: .ASCIZ <15><12> MPFAIL: .ASCIZ / POWER FAILURE, PROGRAM RESTART AT TEST IN PROGRESS/
3040	015512	047440	020106	040522	
3041	015520	043516	035105	042522	
3042	015526	054524	042520	046040	
3043	015534	051501	020124	042504	
3044	015542	044526	042503	051040	
3045	015550	041530	051123	040440	
3046	015556	042104	042522	051523	
3047	015564	000055			
3048	015566	020040	000077		
3049	015572	005015	000		
3050	015575	040	050040	053517	
3051	015602	051105	043040	044501	
3052	015610	052514	042522	020054	
3053	015616	051120	043517	040522	
3054	015624	020115	042522	052123	
3055	015632	051101	020124	052101	
3056	015640	052040	051505	020124	
3057	015646	047111	050040	047522	
3058	015654	051107	051505	000123	
3059	015662	005015	047105	020104	MEPASS: .ASCIZ <15><12>/END OF PASS TAPE C/
3060	015670	043117	050040	051501	
3061	015676	020123	040524	042520	
3062	015704	041440	000		
3063	015707	015	051012	000	MR: .ASCIZ <15><12>/R/
3064	015713	015	052012	051505	MTSTPC: .ASCIZ <15><12>/TEST PC-/ MLOCK: .ASCIZ <15><12>/LOCK ON SELECTED TEST? (Y OR N)-/
3065	015720	020124	041520	000055	
3066	015726	005015	047514	045503	
3067	015734	047440	020116	042523	
3068	015742	042514	052103	042105	
3069	015750	052040	051505	037524	
3070	015756	024040	020131	051117	
3071	015764	047040	026451	000	
3072	015771	015	042012	020125	MLEVEL: .ASCIZ <15><12>/DU PRIORITY LEVEL-/ MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3073	015776	051120	047511	044522	
3074	016004	054524	046040	053105	
3075	016012	046105	000055		
3076	016016	005015	020043	043117	

3077	016024	051440	047131	020103
3078	016032	044103	051101	020123
3079	016040	042523	042514	052103
3080	016046	042105	024040	030440
3081	016054	047440	020122	024462
3082	016062	000055		
3083	016064	005015	051511	051440
3084	016072	041505	054040	044515
3085	016100	020124	052512	050115
3086	016106	051105	021440	020066
3087	016114	047111	020077	054450
3088	016122	047440	020122	024516
3089	016130	000055		
3090	016132	005015	051511	051440
3091	016140	041505	051040	041505
3092	016146	045040	046525	042520
3093	016154	020122	032443	044440
3094	016162	037516	024040	020131
3095	016170	051117	047040	026451
3096	016176	000		
3097	016177	015	044412	020123
3098	016204	050117	020124	046103
3099	016212	020122	047105	041101
3100	016220	042514	045040	046525
3101	016226	042520	020122	032043
3102	016234	044440	037516	024040
3103	016242	020131	051117	047040
3104	016250	026451	000	
3105	016253	015	040412	042522
3106	016260	054440	052517	051040
3107	016266	047125	044516	043516
3108	016274	044440	020116	040515
3109	016302	047111	020124	047515
3110	016310	042504	042440	052130
3111	016316	051105	040516	037514
3112	016324	005015	040401	042116
3113	016332	027040	027056	027056
3114	016340	042040	020117	047531
3115	016346	020125	040510	042526
3116	016354	052040	042510	042440
3117	016362	052130	051105	040516
3118	016370	020114	047515	042504
3119	016376	020115	054502	040520
3120	016404	051523		
3121	016406	005015	045001	046525
3122	016414	042520	020122	047503
3123	016422	047116	041505	047524
3124	016430	020122	047117	037440
3125	016436	054450	047440	020122
3126	016444	024516	000055	
3127				
3128				
3129				
3130				
3131	016450	000040		
3132	016510	000040		

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: .ASCII <15><12>/ARE YOU RUNNING IN MAINT MODE EXTERNAL?/

.ASCII <15><12><1>/AND DO YOU HAVE THE EXTERNAL MODEM BYPASS/

.ASCIZ <15><12><1>/JUMPER CONNECTOR ON ?(Y OR N)-/

.EVEN

;BUFFERS FOR INPUT-OUTPUT

INBUF: .BLKB 40
TEMP: .BLKB 40

```

3133 016550 000040          MDATA: .BLKB 40
3134                          ;*****
3135                          ;UTILITIES
3136                          ;*****
3137
3138      :THIS UTILITY CALCULATES PRIORITY LEVEL
3139 016610 006367 000044  DULEV: ASL     DUPRT     ;SHIFT LEFT
3140 016614 006367 000040          ASL     DUPRT     ;
3141 016620 006367 000034          ASL     DUPRT     ;
3142 016624 006367 000030          ASL     DUPRT     ;
3143 016630 006367 000024          ASL     DUPRT     ;
3144 016634 016767 000020 000020 MOV     DUPRT,LESS1 ;MOVE THIS TO LESS1
3145 016642 162767 000001 000012 SUB     #1,LESS1   ;CREATE LESS1
3146 016650 042767 000037 000004 BIC     #37,LESS1 ;CLEAR TNZVC
3147 016656 000207
3148 016660 000240          DUPRT: LEVEL5
3149 016662 000200          LESS1: LEVEL4 ;LEVEL TO ALLOW INTERRUPTS
3150
3151      :NEW DU ADDRESSES
3152 016664 016767 000126 000422 DUADDR: MOV     DUBASE,RXCSR ;XXX0
3153 016672 005267 000120          INC     DUBASE
3154 016676 016767 000114 000412 MOV     DUBASE,HRXCSR ;XXX1
3155 016704 005267 000106          INC     DUBASE
3156 016710 016767 000102 000402 MOV     DUBASE,RXDBUF ;XXX2
3157 016716 016767 000074 000400 MOV     DUBASE,PARCSR ;XXX2
3158 016724 005267 000066          INC     DUBASE
3159 016730 016767 000062 000364 MOV     DUBASE,HRXDBUF ;XXX3
3160 016736 016767 000054 000362 MOV     DUBASE,HPARCSR ;XXX3
3161 016744 005267 000046          INC     DUBASE
3162 016750 016767 000042 000352 MOV     DUBASE,TXCSR  ;XXX4
3163 016756 005267 000034          INC     DUBASE
3164 016762 016767 000030 000342 MOV     DUBASE,HTXCSR ;XXX5
3165 016770 005267 000022          INC     DUBASE
3166 016774 016767 000016 000332 MOV     DUBASE,TXDBUF ;XXX6
3167 017002 005267 000010          INC     DUBASE
3168 017006 016767 000004 000322 MOV     DUBASE,HTXDBUF ;XXX7
3169 017014 000207
3170 017016 000000          DUBASE: 0
3171
3172      :THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3173      :INFORMATION CONTAINED IN TEMP1 AND IT IS
3174      :SHIFTED IN BY THE CONTENTS OF SHIFT
3175 017020 042777 040000 000302 RPOKE: BIC     #MTDATA,@TXCSR
3176 017026 005067 162114          CLR     TEMP2
3177 017032 006067 162106          ROR     TEMP1     ;FORCE CARRY
3178 017036 006067 162104          ROR     TEMP2     ;PICK UP CARRY IN BIT 15
3179 017042 006267 162100          ASR     TEMP2     ;SHIFT INTO BIT 14
3180 017046 042767 100000 162072 BIC     #BIT15,TEMP2 ;CLR BIT 15
3181 017054 056777 162066 000246 BIS     TEMP2,@TXCSR ;POKE MAINT DATA
3182 017062 042777 020000 000240 BIC     #CLK,@TXCSR ;POKE CLK
3183 017070 052777 020000 000232 BIS     #CLK,@TXCSR ;
3184 017076 005367 162036          DEC     SHIFT
3185 017102 001346          BNE     RPOKE
3186 017104 000207          RTS     PC
3187
3188 017106 016767 162032 162032 ODD8: MOV     TEMP1,TEMP2 ;SAVE TEMP1

```



```

3189 017114 005067 162030 CLR TEMP3
3190 017120 012727 000010 MOV #8.,(PC)+
3191 017124 000000 4$: 0
3192 017126 006067 162014 1$: ROR TEMP2
3193 017132 005567 162012 ADC TEMP3
3194 017136 005367 177762 DEC 4$
3195 017142 001371 BNE 1$
3196 017144 006067 162000 ROR TEMP3
3197 017150 103404 BCS 2$
3198 017152 052767 000400 161764 BIS #BIT8,TEMP1 ;SET ODD PARITY
3199 017160 000403 BR 3$
3200 017162 042767 000400 161754 2$: BIC #BIT8,TEMP1 ;CLR EVEN PARITY
3201 :TEMP1 NOW HAS ODD PARITY CHARACTER
3202 017170 000207 3$: RTS PC
3203
3204 :THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3205 017172 016767 161746 161746 EVEN8: MOV TEMP1,TEMP2 ;SAVE TEMP1
3206 017200 005067 161744 CLR TEMP3
3207 017204 012727 000010 MOV #8.,(PC)+
3208 017210 000000 4$: 0
3209 017212 006067 161730 1$: ROR TEMP2
3210 017216 005567 161726 ADC TEMP3
3211 017222 005367 177762 DEC 4$
3212 017226 001371 BNE 1$
3213 017230 006067 161714 ROR TEMP3
3214 017234 103004 BCC 2$
3215 017236 052767 000400 161700 BIS #BIT8,TEMP1 ;SET EVEN PARITY
3216 017244 000403 BR 3$
3217 017246 042767 000400 161670 2$: BIC #BIT8,TEMP1 ;CLR ODD PARITY
3218 :TEMP1 NOW HAS EVEN PARITY CHARACTER
3219 017254 000207 3$: RTS PC
3220
3221 017256 062716 000002 TRPREG: ADD #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3222 ;IN MAIN PART OF THE PROGRAM
3223 017262 000002 RTI
3224 ;ERROR HLT TABLE
3225 017264 017350 .ERRTAB: EM0 ;HLT 0 BIT ERROR (GENERAL)
3226 017266 000000 0
3227 017270 000000 0
3228 017272 017364 EM1 ;HLT 1 REGISTER ERROR
3229 017274 017535 DH1
3230 017276 017556 DT1
3231 017300 017426 EM2 ;HLT 2 RECEIVER ERROR
3232 017302 017535 DH1
3233 017304 017556 DT1
3234 017306 017470 EM3 ;HLT 3 TRANSMITTER ERROR
3235 017310 017535 DH1
3236 017312 017556 DT1
3237
3238 017314 160040 RXCSR: 160040 ;DEFAULT DU ADDRESSES
3239 017316 160041 HRXCSR: 160041
3240 017320 160042 RXDBUF: 160042
3241 017322 160043 HRXDBUF: 160043
3242 017324 160042 PARCSR: 160042
3243 017326 160043 HPARCSR: 160043
3244 017330 160044 TXCSR: 160044
  
```

K05

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 65
 DZDUCC.P11 10-JUL-74 00:00

SEQ 0062

3245	017332	160045			HTXCSR: 160045
3246	017334	160046			TXDBUF: 160046
3247	017336	160047			HTXDBUF: 160047
3248					;DEFAULT DU VECTORS
3249	017340	000770			DURIV: 770 ;REC INTR VECTOR
3250	017342	000772			DURIS: 772 ;REC INTR STATUS
3251	017344	000774			DUTIV: 774 ;XMIT INTR VECTOR
3252	017346	000776			DUTIS: 776 ;XMIT INTR STATUS
3253					;ERROR MESSAGES
3254	017350	036440	042440	051122	EMO: .ASCIZ / = ERROR PC/
3255	017356	051117	050040	000103	
3256	017364	036440	051040	043505	EM1: .ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
3257	017372	051511	042524	020122	
3258	017400	051105	047522	020122	
3259	017406	041520	005015	051001	
3260	017414	043505	051511	042524	
3261	017422	020122	000040		
3262	017426	036440	051040	041505	EM2: .ASCIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
3263	017434	044505	042526	020122	
3264	017442	051105	047522	020122	
3265	017450	041520	005015	051001	
3266	017456	043505	051511	042524	
3267	017464	020122	000040		
3268	017470	036440	052040	040522	EM3: .ASCIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
3269	017476	051516	044515	052124	
3270	017504	051105	042440	051122	
3271	017512	051117	050040	006503	
3272	017520	000412	042522	044507	
3273	017526	052123	051105	020040	
3274	017534	000			
3275					;DATA HEADERS FOR ERROR MESSAGES
3276	017535	105	050130	041505	DH1: .ASCIZ /EXPECTED ACTUAL/
3277	017542	042524	020104	040440	
3278	017550	052103	040525	000114	
3279					.EVEN
3280					;DATA TABLES FOR ERROR MESSAGES
3281	017556	000003			DT1: 3
3282	017560	006	004		.BYTE 6,4
3283	017562	001164			SAVR3 ;REGISTER
3284	017564	006	004		.BYTE 6,4
3285	017566	001156			SAVR0 ;EXPECTED DATA
3286	017570	006	002		.BYTE 6,2
3287	017572	001160			SAVR1 ;ACTUAL DATA
3288		000001			.END

M05

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 68
DZDUCC.P11 10-JUL-74 00:00 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0064

DEVICE	015403	2527	3027#											
DH1	017535	3229	3232	3235	3276#									
DISPRE	000174	710#	854											
DLIGHT=	177570	600#	845											
DNA =	100000	679#												
DNAINT=	000040	686#												
DSC =	100000	643#												
DSINTE=	000040	653#												
DSR =	001000	649#												
DSWR =	177570	599#	844											
DTR =	000002	657#												
DT1	017556	3230	3233	3236	3281#									
DUADDR	016664	883	2554	3152#										
DUBASE	017016	879	882	2553*	3152	3153*	3154	3155*	3156	3157	3158*	3159	3160	3161*
		3162	3163*	3164	3165*	3166	3167*	3168	3170#					
DULEV	016610	949	3139#											
DUPRT	016660	946	3139*	3140*	3141*	3142*	3143*	3144	3148#					
DURIS	017342	2557*	3250#											
DURIV	017340	890	893	894	2555*	2562	3249#							
DUTIS	017346	2561*	3252#											
DUTIV	017344	2559*	3251#											
EIGHT =	006000	674#	1163	1331	1500	1539	1584	1629	1668	1819	1958	2103	2150	
EM0	017350	3225	3254#											
EM1	017364	3228	3256#											
EM2	017426	3231	3262#											
EM3	017470	3234	3268#											
ERRCNT	001132	736#	835*	2901*										
ERRFLG	001224	782#	834*	2571*	2617*	2863*	2876	2882*	2947*					
ERRMSG	014442	2873*	2884#											
ERTAB0	014560	2881	2910#											
EVEN8	017172	1552	1639	3205#										
EVEPAR=	001400	677#	1202	1245	1288	1331	1539	1629	2150	2247	2347	2447		
EVPAR =	000400	666#												
EXITER	014512	2895	2900#											
FIVE =	000000	671#	1043	1202	1371	1708	1856	1992	2447					
FRMERR=	020000	662#	1056	1096	1136	1176	2208	2211	2214	2308	2311	2314	2408	2411
		2414	2508	2511	2514									
HALTS	014472	2859	2894#											
HDXEN =	000010	688#												
HILIM	013672	2692*	2722	2740#										
HOLD	001136	741#												
HPARCS	017326	3160*	3243#											
HRXCSR	017316	3154*	3239#											
HRXDBU	017322	3159*	3241#											
HTXCSR	017332	3164*	3245#											
HTXDBU	017336	3168*	3247#											
ICOUNT	001122	732#	2615	2619*										
INBUF	016450	957	961	2656	2698	2830	2834	3131#						
INIFLG	001222	780#	840	843*										
INSTER=	104404	800#	965	2717	2839									
INSTR =	104403	798#	874	885	895	906	931	941	955	968	972	976	980	999
		1012	2975											
INSTR2	013456	2667	2682#											
ISYMOD=	000000	670#	1036	1043	1076	1083	1116	1123	1156	1163	1195	1202	1238	1245
		1281	1288	1324	1331	1364	1371	1407	1414	1450	1457	1493	1500	1532
		1539	1577	1584	1701	1708	1738	1745	1775	1782	1812	1819	2143	2150

E06

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 73
 DZDUCC.P11 10-JUL-74 00:00

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0069

TST24	007420	1915	1948#											
TST25	007570	1949	1982#											
TST26	007754	1983	2019#											
TST27	010140	2020	2056#											
TST28	010324	2057	2093#											
TST29	010510	2094	2140#											
TST3	002700	1074	1113#											
TST30	011112	2141	2237#											
TST31	011514	2238	2337#											
TST32	012116	2338	2437#	2518										
TST33	= ***** U	2438												
TST4	003054	1114	1153#											
TST5	003230	1154	1192#											
TST6	003414	1193	1235#											
TST7	003600	1236	1278#											
TST8	003764	1279	1321#											
TST9	004150	1322	1361#											
TTST	013132	994#	995#	1005#	1006#	1008#	1009#	2606#						
TXCSR	017330	1035#	1037#	1040#	1046#	1047#	1049#	1050#	1075#	1077#	1080#	1086#	1087#	1089#
		1090#	1115#	1117#	1120#	1126#	1127#	1129#	1130#	1155#	1157#	1160#	1166#	1167#
		1169#	1170#	1194#	1196#	1199#	1205#	1206#	1208#	1209#	1237#	1239#	1242#	1248#
		1249#	1251#	1252#	1280#	1282#	1285#	1291#	1292#	1294#	1295#	1323#	1325#	1328#
		1334#	1335#	1337#	1338#	1363#	1365#	1368#	1374#	1375#	1377#	1378#	1406#	1408#
		1411#	1417#	1418#	1420#	1421#	1449#	1451#	1454#	1460#	1461#	1463#	1464#	1492#
		1494#	1497#	1503#	1504#	1506#	1507#	1531#	1533#	1536#	1542#	1543#	1545#	1546#
		1576#	1578#	1581#	1587#	1588#	1590#	1591#	1621#	1623#	1626#	1632#	1633#	1660#
		1662#	1665#	1671#	1672#	1700#	1702#	1705#	1711#	1712#	1714#	1715#	1737#	1739#
		1742#	1748#	1749#	1751#	1752#	1774#	1776#	1779#	1785#	1786#	1788#	1789#	1811#
		1813#	1816#	1822#	1823#	1825#	1826#	1848#	1850#	1853#	1859#	1860#	1882#	1884#
		1887#	1893#	1894#	1916#	1918#	1921#	1927#	1928#	1950#	1952#	1955#	1961#	1962#
		1984#	1986#	1989#	1995#	1996#	1998#	1999#	2021#	2023#	2026#	2032#	2033#	2035#
		2036#	2058#	2060#	2063#	2069#	2070#	2072#	2073#	2095#	2097#	2100#	2106#	2107#
		2109#	2110#	2142#	2144#	2147#	2155#	2156#	2158#	2159#	2239#	2241#	2244#	2252#
		2253#	2255#	2256#	2339#	2341#	2344#	2352#	2353#	2355#	2356#	2439#	2441#	2444#
		2452#	2453#	2455#	2456#	3162#	3175#	3181#	3182#	3183#	3244#			
		3166#	3246#											
TXDBUF	017334	684#												
TXDONE=	000200	685#												
TXINTE=	000100	2879	2886	2889#										
TYPDAT	014456	796#	842	1022	2524	2527	2532	2654	2664	2676	2680	2771	2808	2883
TYPE =	104402	2887	2945	2971	2973	2981								
		2877	2880#											
TYPMSG	014426	691#												
USER =	000000	658#												
VOID =	000001	2781#	2810#	2819#										
MRDCNT	014210	2538#												
ZERO	012574	2971	2987#											
SCNTG	015060	552#	1034	1035#	1074	1075#	1114	1115#	1154	1155#	1193	1194#	1236	1237#
SE =	000042	1279	1280#	1322	1323#	1362	1363#	1405	1406#	1448	1449#	1491	1492#	1530
		1531#	1575	1576#	1620	1621#	1659	1660#	1699	1700#	1736	1737#	1773	1774#
		1810	1811#	1847	1848#	1881	1882#	1915	1916#	1949	1950#	1983	1984#	2020
		2021#	2057	2058#	2094	2095#	2141	2142#	2238	2239#	2338	2339#	2438	2439#
SMNEW	015075	2975	2990#											
SMSWR	015065	2973	2988#											
SN =	000040	552#	1033	1035#	1073	1075#	1113	1115#	1153	1155#	1192	1194#	1235	1237#
		1278	1280#	1321	1323#	1361	1363#	1404	1406#	1447	1449#	1490	1492#	1529

G06

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 76
 DZDUCC.P11 10-JUL-74 00:00

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0071

HLT	612# 1429 1902 2363	1060 1434 1936 2374	1100 1472 1970 2384	1140 1477 2007 2389	1180 1515 2044 2463	1217 1520 2081 2474	1222 1561 2118 2484	1260 1606 2166 2489	1265 1645 2177	1303 1684 2187	1308 1723 2192	1346 1760 2263	1351 1797 2274	1386 1834 2284	1391 1868 2289
PRGEND	552#	2517													
PRGFRT	552#	553													
PUSYF	552#														
RSETUP	552# 1621 2239	1035 1660 2339	1075 1700 2439	1115 1737	1155 1774	1194 1811	1237 1848	1280 1882	1323 1916	1363 1950	1406 1984	1449 2021	1492 2058	1531 2095	1576 2142
TSETUP	552#														
\$BEGIN	552#	984													
\$BINAR	552#														
\$BUFFE	552#	3128													
\$CABLE	552#														
\$CATCH	552#	695													
\$CLRVE	552#	844													
\$CONVR	552#	2768													
\$DNA	552#														
\$EOP	552#	2517													
\$GETFL	552#	895	968	972	976	980	999								
\$GETPA	552#	874	885	905	931	940	1012								
\$GETSY	552#	950													
\$HEADE	552#	553													
\$HLT	552#	2854													
\$INSTR	552#	2647													
\$ISOB	552#	1523	1568	1613	1652										
\$MATCH	552#														
\$MRR	552#														
\$MRRW	552#														
\$MRW	552#														
\$MSG	552#	2993													
\$PARAM	552#	2685													
\$PFAIL	552#	2913													
\$POKE	552#														
\$POKER	552# 1631 2254	1048 1670 2354	1088 1713 2454	1128 1750	1168 1787	1207 1824	1250 1858	1293 1892	1336 1926	1376 1960	1419 1997	1462 2034	1505 2071	1544 2108	1589 2157
\$RCNET	552#	2123	2217	2317	2417										
\$RECAC	552#														
\$REG	552#	2744													
\$RESET	552# 1323 1623 1916 2241	1035 1325 1660 1918 2339	1037 1363 1662 1950 2341	1075 1365 1700 1952 2439	1077 1406 1702 1984 2441	1115 1408 1737 1986	1117 1449 1739 2021	1155 1451 1774 2023	1157 1492 1776 2058	1194 1494 1811 2060	1196 1531 1813 2095	1237 1533 1848 2097	1239 1576 1850 2142	1280 1578 1882 2144	1282 1621 1884 2239
\$RXACT	552#														
\$SCOPE	552#	2585													
\$SCOPI	552#	2625													
\$SETFL	552#	2823													
\$SETVE	552#	696													
\$START	552#	820													
\$STRIP	552#	1691	1728	1765	1802	1839	1873	1907	1941	1975	2012	2049	2086		
\$SYMBO	552#	565													
\$SYNCR	552# 1709	1044 1746	1084 1783	1124 1820	1164 1993	1203 2030	1246 2067	1289 2104	1332 2153	1372 2250	1415 2350	1458 2450	1501	1540	1585

H06

DZDUC-B MACY11 27(1006) 01-OCT-76 09:28 PAGE 77

DZDUCC.P11 10-JUL-74 00:00

CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0072

STRAPS	552#	784													
STRPAR	552#														
STRPDE	552#	792	794	796	798	800	802	804	806	808	810	812	814	816	
STRPSR	552#	2841													
STSTNO	552#	1033	1073	1113	1153	1192	1235	1278	1321	1361	1404	1447	1490	1529	1574
	1619	1658	1698	1735	1772	1809	1846	1880	1914	1948	1992	2019	2056	2093	2140
	2237	2337	2437												
STYPE	552#	2633													
SUNIBU	552#														
SVARIA	552#	715													
SWORDF	552#	1025	1065	1105	1145										
SWCRDO	552#														
SWORDP	552#	1185	1228	1271	1314	1354	1397	1440	1483						

. ABS. 017574 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDUCC, DZDUCC/CRF/SOL=HELLO.P11, PARA.P11, KEET.P11, DZDUCC.P11
RUN-TIME: 22 32 3 SECONDS
RUN-TIME RATIO: 210/58=3.6
CORE USED: 18K (35 PAGES)

