

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 D11'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

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

D11 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
STARTING ADDRESS

4.2

THE STARTING ADDRESS FOR ALL TESTS IS 000200

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

- 4.3 PROGRAM AND/OR OPERATOR ACTION

- 4.3.1 INITIAL PROGRAM START

4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4.3.1.2 LOAD ADDRESS 000200

4.3.1.3 CLEAR CONSOLE SWITCHES

4.3.1.4 PRESS START

NOTE: IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
SWR=XXXXXX MEM= (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 MEM= (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 DUI 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 DUI 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 DUI 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 DUI 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.....
 SHOWS 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 ? AND DO YOU HAVE THE EXTERNAL MODEN 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 CONFERENCE 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 TO CHOOSE 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 (<PG>): 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 PRSS" TYPEOUT - TURN TELETYPE OFF

6. ERRORS

6.1 ERROR HALTS
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

NOTE: IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT THE OPERATOR IS REQUIRED TO TYPE A <IG> 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 2 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

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";
THEBY 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

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 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 JMBY: 377

9. PROGRAM DESCRIPTION

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

11. LISTINGS

MO1

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 15
DZDUF.C.P11 05-AUG-76 00:00

SEQ 0012

```
552 .ENABLE ABS
553
554 ;DU11 DZDUF-C TAPE F
555 ;COPYRIGHT 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
556
557 ;STARTING PROCEDURE
558 ;LOAD PROGRAM
559 ;PRESS START
560 ;PROGRAM WILL TYPE "DU11 DZDUF-C TAPE F "
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 "
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
581 000002 SW01=2
582 000001 SW00=1
583
584 ;LOCK ON TEST SELECT
585 ;RESTART PROGRAM AT SELECTED TEST
586 ;RESELECT VECTOR AND CONTROL REGISTER
587 ;ADDRESS AFTER PROGRAM RESTART
```

```

595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
;REGISTER DEFINITIONS
RO=%0      :GENERAL REGISTER
R1=%1      :GENERAL REGISTER
R2=%2      :GENERAL REGISTER
R3=%3      :GENERAL REGISTER
R4=%4      :GENERAL REGISTER
R5=%5      :GENERAL REGISTER
SP=%6      :PROCESSOR STACK POINTER
PC=%7      :PROGRAM COUNTER

;LOCATION EQUIVALENCIES
DSWR=177570 :HARDWARE SWITCH REGISTER LOC.
DLIGHTS=177570 :HARDWARE DISPLAY REGISTER LOC.
PS=177776   :PROCESSOR STATUS WORD
STACK=1100  :START OF PROCESSOR STACK

;INSTRUCTION DEFINITIONS
PUSH1SP=5746 :DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
POP1SP=5726  :INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
PUSHR0=10046 :SAVE R0 ON STACK =MOV R0,-(SP)
POPR0=12600  :RESTORE R0 FROM STACK =MOV (SP)+,R0
PUSH2SP=24646 :DECREMENT STACK TWICE =CMP -(SP),-(SP)
POP2SP=22626  :INCREMENT STACK TWICE =CMP (SP)+,(SP)+
.EQUIV ENT,HLT :BASIC DEFINITION OF ERROR CALL

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

;PROCESSER 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 CARRDET=BIT12 :CARRIER DETECT
647 004000 RECACT=BIT11 :REC ACTIVE
648 002000 SRO=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 SYNCH=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 ;RXDBUF BIT DEFINITIONS
660 100000 RXERR=BIT15 :REC ERROR
661 040000 OVRUN=BIT14 :OVERRUN
662 020000 FRMERR=BIT13 :FRAME ERROR
663 010000 PARERR=BIT12 :PARITY ERROR
664 ;PARCSR BIT DEFINITIONS
665 001000 PAREN=BIT9 :PARITY ENABLE
666 000400 EVPAR=BIT8 :EVEN PARITY SENSE
667 ;PARCSR WRD DEFINITIONS
668 030000 SYNINT=30000 :SYNC EXTERNAL MODE
669 020000 SYNEXT=20000 :SYNC INTERNAL MODE
670 000000 ISYNC=0 :ISOC MODE
671 000003 FIVE=0 :WORD LENGTH 5 BITS
672 002000 SIX=2000 :WORD LENGTH 6 BITS
673 004000 SEVEN=4000 :WORD LENGTH 7 BITS
674 000000 EIGHT=6000 :WORD LENGTH 8 BITS
675 000000 NOPAR=0 :NO PARITY
676 001000 OODPAR=1000 :ODD PARITY
677 001400 EVEPAR=1400 :EVEN PARITY
678 ;TXCSR BIT DEFINITIONS
679 100000 DNR=BIT15 :DATA NOT AVAILABLE
680 040000 MTDATA=BIT14 :MAINT DATA
681 020000 CLK=BIT13 :CLK
682 002000 BITW=BIT10 :BIT WINDOW
683 000400 MRESET=BIT8 :MASTER RESET
684 000200 TXDONE=BIT7 :XMIT DONE
685 000100 TXINTE=BIT6 :XMIT INTR ENABLE
686 000040 DNAINTE=BIT5 :DNR INTR ENAB
687 000020 SEND=BIT4 :SEND
688 000010 HUXEN=BIT3 :HUX/FDX
689 000001 BREAK=BIT0 :BREAK
690 ;TXCSR WRD DEFINITIONS
691 000000 USER=0 :USER MODE
692 004000 MINT=4000 :MAINT INT MODE
693 010000 MEXT=10000 :MAINT EXT MODE
694 014000 SYSTST=14000 :SYSTEM TEST MODE
695 ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
    
```

```

696                                     ;STANDARD INTERRUPT VECTORS
697
698
699                                     . = 24
700 000024 007332                .PFAIL                :POWER FAIL HANDLER
701 000026 007340                340                  :SERVICE AT LEVEL 7
702 000030 007062                .HLT                  :ERROR HANDLER
703 000032 007040                340                  :SERVICE AT LEVEL 7
704 000034 007030                .TRPSRV              :GENERAL HANDLER DISPATCH SERVICE
705 000036 007040                340                  :SERVICE AT LEVEL 7
706
707                                     ;SOFTWARE SWITCH REGISTER
708
709                                     . = 174
710 000174 000700                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
717
718                                     ;INDIRECT POINTERS
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                TPD8R: 177566        :TELEPRINTER DATA BUFFER
726
727                                     ;PROGRAM CONTROL PARAMETERS
728
729 001114 000000                RTRN: 0              :SCOPE ADDRESS FOR LOOP ON TEST
730 001116 000700                NEXT: 000700         :ADDRESS OF NEXT TEST TO BE EXECUTED
731 001120 000700                LOCK: 000700         :ADDRESS FOR LOCK ON CURRENT DATA
732 001122 000000                ICOUNT: 0            :NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
733 001124 000700                LPENT: 000700        :NUMBER OF ITERATIONS COMPLETED
734 001126 000700                TSTNO: 000700        :NUMBER OF TEST IN PROGRESS
735 001130 000700                PNT: 000700         :NUMBER OF PASSES COMPLETED
736 001132 000000                EMT: 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 000700                SHIFT: 000700       :TEMPORARY STORAGE=# OF SHIFTS PER CHAR
743 001142 000000                COUNT: 000000       :TEMPORARY STORAGE=# OF TIMES A CHAR WILL BE SENT
744 001144 000000                T1: 000000          :TEMPORARY STORAGE
745 001146 000000                T2: 000000          :TEMPORARY STORAGE
746 001148 000000                T3: 000000          :TEMPORARY STORAGE
747 001150 000000                T4: 000000          :TEMPORARY STORAGE
748 001152 000000                T5: 000000          :TEMPORARY STORAGE
749 001154 000000                S1: 000000          :R0 STORAGE
750 001156 000000                S2: 000000          :R1 STORAGE
751 001158 000000                S3: 000000          :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
:SP	STACK POINTER STORAGE
:PC	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 BOCFAD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
768 001206 000000 KEEL300: 0 ;SAVED 1ST DEVICE ADDR
769 001210 000000 LASTAD: 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 NOTIFY 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 001228 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
780 001229 000 STFLG: .BYTE 0 ;TEST START FLAG
781 001231 000 ENRFLG: .BYTE 0 ;ERROR OCCURRED FLAG
782 001232 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
783
784 ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
785 ;POINTERS TO SUBROUTINES CAN BE FOUND
786 ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS
787
788 .TRPTAB:
789 *****
790 *****
791
792 001226 104400 .SCOPE SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
793 001226 005614 .SCOPE1 SCOP1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
794 001230 104401 .SCOPE1 SCOP1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
795 001230 006000 .TYPE TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
796 104402
797 001232 007000 .INSTR INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
798 104403
799 001234 005060 .INSTR INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
800 104404
801 001236 001176 .INSTR INSTR=TRAP+4 ;CALL TO INPUT ERROR HANDLER
802 104405
803 001240 006230 .PARAM PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INPUT ROUTINE
804 104406
805 001242 007444 .SAVOS SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
806 104407
807 001244 006504 .RESOS RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
808 104410
809 001246 007536 .CONVRT CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
810 104411
811 001250 006542 .CONVRT CONVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF
812 104412

```

813 001252 006762
814 104413
815 001254 007476
816 104414
817 001256 007552
818
819
820
821
822
823
824
825
826
827
828 001260 012767 000340 176510
829 001256 012706 001100
830 001272 012737 007332 000024
831 001300 005067 177620
832 001304 105067 177713
833 001310 005067 177614
834 001314 105067 177704
835 001320 005067 177606
836 001324 005067 177604
837 001330 012767 000001 177570
838 001336 012767 001260 177550
839
840 001344 105767 177652
841 001350 001004
842 001352 104402 007652
843 001356 105167 177640
844 001362 012767 177570 177510
845 001370 012767 177570 177504
846 001376 013746 000006
847 001402 013746 000004
848 001406 012737 001426 000004
849 001414 022777 177777 177456
850 001422 001402
851 001424 000407
852 001426 022626 645:
853 001430 012767 000176 177442 655:
854 001436 012767 000174 177436
855 001444 012637 000004 665:
856 001450 012637 000006
857 001454 005737 000012
858 001460 001005
859 001462 022767 000176 177410
860 001470 001001
861 001472 104414
862 001474 032777 000001 177376 675:
863 001502 001002
864 001504 000167 000446
865 001510 012700 000300 18:
866 001514 012701 000302
867 001520 012702 000004
868 001524 010110 28:

```
.SETFLG
. CKSWR=TRAP+13 ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
. CKSWR
. CNTLU=TRAP+14 ;CALL TO ALLOW LOADING OF SWREG FROM TTY
. CNTLU
;*****
;*****
```

```
:PROGRAM INITIALIZATION
:LOCK OUT INTERRUPTS
:SET UP PROCESSOR STACK
:SET UP PC R FAIL VECTOR
:CLEAR PROGRAM CONTROL FLAGS AND COUNTS
:TYPE TITLE MESSAGE
```

```
.START: MOV #340,PS ;LOCK OUT INTERRUPTS
MOV #STACK,SP ;SET UP STACK
MOV #PFAIL,#2024 ;SET UP POWER FAIL VECTOR
CLR LPCNT ;CLEAR # OF ITERATION COMPLETED LOCATION
CLAB STFLG ;CLEAR START FLAG
CLR PASCNT ;CLEAR PASS COUNT
CLAB ERRFLG ;CLEAR ERROR FLAG
CLR ERRCNT ;CLEAR ERROR COUNT
CLR LSTERR ;CLEAR LAST ERROR POINTER
MOV #1,TSTNO ;SET UP FOR TEST 1
MOV #.START,RTRN ;SET UP FOR POWER FAIL BEFORE
;TESTING STARTS
;HAS INITIALIZATION BEEN PERFORMED

TSTB INIFLG
BNE ONCE
TYPE #TITLE
COMB INIFLG
ONCE: MOV #0,SWR ;IF NOT SET FLAG AND DO
MOV #0,LIGHTS,LIGHTS ;RELOAD HARDWARE SWITCH REGISTER INTO POINTER
MOV #206,-(SP) ;RELOAD HARDWARE DISPLAY REGISTER INTO POINTER
MOV #204,-(SP) ;SAVE VECTORS
MOV #648,#204 ;SET UP FOR TIMEOUT
CMP #1,#2,SWR ;REFERENCE HARDWARE SWITCH REGISTER
BEQ 658
BR 668
645: CMP (SP)+,(SP)+ ;ADJUST STACK
655: MOV #5,REG,SWR ;POINT TO SOFTWARE SWITCH REG
MOV #015,REG,LIGHTS ;POINT TO SOFT DISPLAY REG
665: MOV (SP)+,#204 ;RESTORE VECTORS
MOV (SP)+,#206
TST #2042 ;UNDER MONITOR
BNE 678 ;IS SWREG USED
CMP #5,REG,SWR
BNE 678
675: BIT #SH00,#2,SWR ;RESELECT VECTOR & CONTROL REG?
BNE 18
JMP .BEGIN
18: MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
MOV #302,R1 ;START AT LOCATION 300
MOV #4,R2
MOV R1,(R0)

```

```

869 001526 000011 CLR (R1)
870 001530 C 00 ADD R2,R0
871 001532 C 01 ADD R2,R1
872 001534 C 01 001000 CMP #1000,R1 ;END AT LOCATION 776
873 001536 C 771 BLT 28
874 001540 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
875 001544 007726 MREGAD ;MESSAGE
876 001546 104405 PARAM ;CONVERT STRING
877 001550 160000 160000 ;LOW LIMIT
878 001552 167776 167776 ;HIGH LIMIT
879 001554 011562 DUBASE ;STORE AT THIS LOCATION
880 001556 001 ;MASK
881 001557 001 ;HOW MANY TIMES + 2
882 001560 J16767 007776 177420 .BYTE 1
883 001566 004767 007636 MOV DUBASE,KEEPADD ;SAVE
884 001572 016767 177410 177404 JSR PC,DURADD
885 001600 104403 MOV KEEPADD,BASEADD ;RESTORE FOR ROTATION
886 001602 007704 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
887 001604 104405 MVECTO ;MESSAGE
888 001606 000300 PARAM ;CONVERT STRING
889 001610 000776 300 ;LOW LIMIT
890 001612 012104 776 ;HIGH LIMIT
891 001614 001 DURIV ;STORE AT THIS LOCATION
892 001615 004 .BYTE 1 ;MASK
893 001616 016767 010262 177370 .BYTE 4 ;HOW MANY TIMES + 2
894 001624 016767 010254 177360 MOV DURIV,KEEPIV ;SAVE
895 001632 104403 MOV DURIV,BASEIV ;SET UP FOR ROTATION
896 001634 010007 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
897 001636 104412 MMULT ;MESSAGE
898 001640 001202 SETFLG ;SET FLAG BASED UPON INPUT STRING
899 001642 105767 177334 MULTD ;THIS FLAG
900 ;ARE THERE MULTIPLE DEVICES
901 001646 100406 BMI BBB ;ON THE SYSTEM ?
902 001650 000767 177342 CLR ACTREG ;YES,ASK NEXT QUESTION
903 001654 000367 177340 CLR ROTADD
904 001660 000167 000140 JMP OUTMUL ;JUMP AROUND NEXT QUESTION
905 001664 000167 000140 BBB:
906 001664 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
907 001666 010066 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 ;MASK
913 001701 001 ;HOW MANY TIMES + 2
914 ;THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
915 001702 012767 000001 177310 18: MOV #1,ROTADD ;SET UP POINTER
916 001710 000767 177302 CLR ACTREG ;CLR ACTIVE REGISTER
917 001714 000767 177300 177274 28: BIS ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
918 001722 000241 CLC
919 001724 006167 177270 ROL ROTADD ;SET UP POINTER
920 001730 104421 BCS 38 ;ARE YOU OUT OF RANGE ?
921 001732 002767 000010 177244 ADD #10,BASEADD ;SET UP BASE ADDRESS
922 001740 002767 177244 177236 CMP LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
923 001746 101362 BNE 28 ;NO DO IT AGAIN
924 001750 056767 177244 177240 BIS ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT

```

H02

```

925          ;LEAST TWO DEVICES WHEN YOU ANSWER YES TO
926          ;MULTIPLE DEVICE QUESTION
927 001756 012767 000001 177234 4S:  MOV      ;SET UP FOR LATER USE IN END OF PASS ROUTINE
928 001764 016767 177216 177212  MOV      ;DITTO
929 001772 000414          BR      ;CONTINUE QUESTIONS
930 001774 016767 177206 177202 3S:  MOV      ;RESTORE
931 001772 104403          INSTR     ;OUTPUT MESSAGE & GET INPUT STRING
932 001772 010251          MRRANGE  ;MESSAGE
933 001772 104403          PARAM     ;CONVERT STRING
934 001772 160000          160000  ;LOW LIMIT
935 001772 167776          167776  ;HIGH LIMIT
936 001772 001210          LASTADD  ;STORE AT THIS LOCATION
937 001772 001          ;MASK
938 001772 001          ;HOW MANY TIMES + 2
939 001772 000167 177656          .BYTE   ;DO IT AGAIN
940 001772 001          .BYTE
941 001772 104403          INSTR     ;OUTPUT MESSAGE & GET INPUT STRING
942 001772 010535          MLEVEL  ;MESSAGE
943 001772 104403          PARAM     ;CONVERT STRING
944 001772 000004          4          ;LOW LIMIT
945 001772 000007          7          ;HIGH LIMIT
946 001772 011424          DUPRT    ;STORE AT THIS LOCATION
947 001772 000          ;MASK
948 001772 001          ;HOW MANY TIMES + 2
949 001772 004767 007306          JSR      PC,DLEV
950 001772 004767 007306          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
951 001772 004767 007306          ;BUFFER TO THE CHARACTERS "1" AND "2"
952 001772 004767 007306          ;IF THE CHARACTER IS "1" CLEAR THE FLAG
953 001772 004767 007306          ;IF THE CHARACTER IS "2" SET THE FLAG
954 001772 004767 007306          ;
955 001772 004767 007306          ;
956 001772 004767 007306          ;
957 001772 004767 007306          ;
958 001772 004767 007306          ;
959 001772 004767 007306          ;
960 001772 004767 007306          ;
961 001772 004767 007306          ;
962 001772 004767 007306          ;
963 001772 004767 007306          ;
964 001772 004767 007306          ;
965 001772 004767 007306          ;
966 001772 004767 007306          ;
967 001772 004767 007306          ;
968 001772 004767 007306          ;
969 001772 004767 007306          ;
970 001772 004767 007306          ;
971 001772 004767 007306          ;
972 001772 004767 007306          ;
973 001772 004767 007306          ;
974 001772 004767 007306          ;
975 001772 004767 007306          ;
976 001772 004767 007306          ;
977 001772 004767 007306          ;
978 001772 004767 007306          ;
979 001772 004767 007306          ;
980 001772 004767 007306          ;

```

```

981 002150 011017 NEXTJ ;MESSAGE
982 002152 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
983 002154 001203 JMRBY ;THIS FLAG
984 ;TEST START AND RESTART
985
986 002156 012767 000340 175612 .BEGIN: MOV #340,PS ;LOCK OUT INTERRUPTS
987 012164 012706 001100 MOV #STACK,SP ;SET UP STACK
988 012170 005737 000042 TST #42 ;IS PROGRAM UNDER MONITOR CONTROL
989 002174 001056 BNE #35
990 002176 105767 177000 TSTB MULTD ;DON'T ALLOW LOCK ON TEST IF RUNNING
991 ;MULTIPLE DEVICES
992
993 001407 BEQ #5 ;IF NO TEST FOR LOCK ON TEST
994 016767 003564 003464 MOV BRW,TTST ;RESTORE NORMAL SCOPE LOOP
995 015767 003560 003460 MOV BRX,TTST+2 ;DITTO
996 01444 BR #35 ;JUMP AROUND IF YES
997 012777 000004 176650 5S: BIT #BIT2,@SWR ;CHECK FOR LOCK ON TEST
998 001416 BEQ #15
999 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1000 010472 MLOCK ;MESSAGE
1001 104412 SETFLG ;SET FLAG BASED UPON INPUT STRING
1002 001225 LOKFLG ;THIS FLAG
1003 105767 176757 TSTB LOKFLG ;IS LOCK ON TEST OPTION SELECTED
1004 001407 BEQ #15
1005 012767 000340 003420 MOV #TP,TTST
1006 012767 000340 003414 MOV #TP,TTST+2 ;SET UP TO LOCK
1007 012767 BR #25
1008 016767 002502 003402 1S: MOV BRW,TTST
1009 016767 003376 003376 MOV BRX,TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1010 012777 000002 176570 2S: BIT #SM01,@SWR ;IF SM01=1, GET STARTING PC
1011 001410 BEQ #35
1012 312 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1013 314 012767 MLOCK ;MESSAGE
1014 316 104405 PARAM ;CONVERT STRING
1015 002767 TST1 ;LOW LIMIT
1016 003004 TLAST ;HIGH LIMIT
1017 001114 RTRN ;STORE AT THIS LOCATION
1018 001 .BYTE 1 ;MASK
1019 001 .BYTE 1 ;HOW MANY TIMES + 2
1020 000403 BR #45
1021 012767 002350 176554 3S: MOV #TST1,RTRN ;START AT TEST 1
1022 104402 010453 4S: TYPE #R ;TYPE R
1023 000177 176544 JMP @RTRN ;START TESTING
1024

```

```

1025      ; THIS TEST VERIFYS MATCH DETECT & DATA RDY
1026      ; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1027      ; BY OBSERVING RECACT BIT
1028      ; IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1029      ; * DEPENDENT ON MONITOR .....
1030      ; IF ONE SYNC STRAP IS SELECTED IT WILL
1031      ; ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1032      ; ASSERT
1033      ; MODE: SYNC INTERNAL
1034      ; LENGTH: FIVE
1035      ; SYNC CHARACTER FOR MATCH: B/C
1036      ; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1037
1038 002350 012767 000001 176550 TST1:  MOV    #1,TSTNO      ;SAVE THIS
1039 002356 012767 002676 176532      MOV    #TST2,NEXT    ;GO TO THIS TEST WHEN THRU
1040 002364 012767 000000 176526      MOV    #3$,LOCK      ;SET UP FOR SCOPE LOOP
1041 002372 052777 000000 007474      BIS    #MRESSET,@TXCSR ;MASTER RESET
1042 002400 016703 007460      MOV    #RXDBUF,R3    ;SET UP FOR ERROR MESSAGE
1043      ;SET SYNC INTERNAL,FIVE NO PARITY 0 SYNC REGISTER
1044 002404 012704 030000      MOV    #SYNINT!FIVE!NOPAR,R4 ;CREATE PARAMETERS
1045 002410 012777 000000 007456 15:    MOV    #MINT!SEND,@TXCSR ;SET SEND & MAINT INTER
1046 002416 010477 007446      MOV    R4,@PARCSR    ;LOAD CSR
1047 002422 052777 000020 007430      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
1048      ;POKE CLK TO GET INTO SYNCHRONIZATION
1049      ;BOTH THE LOGIC & RECEIVER
1050 002430 052777 020000 007436      BIS    #CLK,@TXCSR    ;POKE CLK UP
1051 002436 042777 000000 007430      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1052 002444 110477 007430      MOVB  R4,@TXDBUF     ;LOAD DATA CHARACTER
1053      ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1054 002450 052777 020000 007416      BIS    #CLK,@TXCSR    ;POKE CLK UP
1055 002456 042777 020000 007410      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1056 002464 032777 004000 007366      BIT    #RECACT,@RXCSR ;RECACT ?
1057 002472 001401      BEQ    #2$           ;RECACT SHOULD NOT BE SET
1058 002474 104000      M.T
1059 002476      2$:
1060 002476 000404      BR     4$
1061 002500 010477 007364      3$:  MOV    R4,@PARCSR    ;LOAD PARCSR WITH PARAMETERS
1062 002504 110477 007370      MOVB  R4,@TXDBUF     ;LOAD SYNC CHAR
1063 002510 012767 000002 176424 4$:  MOV    #2,COUNT      ;# OF SYNC CHARS
1064 002516 005777 007352      5$:  TST    @TXCSR        ;DNA ?
1065 002522 100001      BPL    #6$           ;BR IF NOT SET
1066 002524 104000      M.T
1067 002526      6$:
1068      ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1069 002526 012767 000005 176404      MOV    #5,SHIFT      ;# OF SHIFTS
1070 002534      7$:
1071 002534 052777 020000 007332      BIS    #CLK,@TXCSR    ;POKE CLK UP
1072 002542 042777 020000 007324      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1073 002550 005367 176364      DEC    SHIFT        ;# OF SHIFTS
1074 002554 001367      BNE    #7$
1075 002556 005367 176360      DEC    COUNT        ;# OF SYNC CHARS
1076 002562 001403      BEQ    #8$
1077      ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1078 002564 105767 176406      TSTB  SYNCNO
1079 002570 100752      BMI    #5$           ;TWO SYNC CHARACTERS..
1080 002572 032777 004000 007260 8$:  BIT    #RECACT,@RXCSR ;RECACT ?

```

K02

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 26
 DZDUF.C.P11 05-AUG-76 00:00

SEQ 0023

```

1081 002600 001001      BNE      95
1082 002602 104000      HLT
1083 002604              95:
1084                          ;REACT FAILED TO SET, POSSIBLE
1085                          ; THAT THE RECEIVER FAILED TO MATCH
1086                          ; THE SYNC CHARACTER
1086 002604 017701 007254  MOV      @RXDBUF,R1      ;SAVE ACTUAL
1087 002610 010400      MOV      R4,R0          ;SAVE EXPECTED
1088 002612 042700 177400  BIC      @177400,R0     ;CLR UPPER BYTE
1089 002616 020001      CMP      R0,R1         ;DO THEY COMPARE ?
1090 002620 0C1401      BEQ      105
1091 002622 104002      HLT
1092 002624              105:
1093                          ;...IT PROBABLY IS A TRANSMITTER ERROR
1094                          ; HOWEVER... IF ONLY THIS FAILED IT
1095                          ; PROBABLY IS A RECEIVER ERROR
1096 002624 104401      SCOPE1
1097                          ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1098                          ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1099                          ;TXDBUF
1100 002626 052777 020000 007240  BIS      @CLK,@TXCSR   ;POKE CLK UP
1101 002634 005777 007234      TST      @TXCSR       ;DNA?
1102 002640 100401      BMI      115
1103 002642 104000      HLT
1104 002644              115:
1105                          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1106 002644 052777 000400 007222  BIS      @MRESET,@TXCSR ;MASTER RESET
1107 002652 032777 000020 007200  BIT      @SYNSCH,@TXCSR ;SYNC SEARCH = 0 ?
1108 002660 001401      BEQ      125
1109 002662 104000      HLT
1110 002664              125:
1111 002664 005204      INC      R4
1112 002666 122704 000040  CMPB    @40,R4        ;IS THIS THE LAST CHARACTER ?
1113 002672 001246      BNE      15
1114 002674 104400      SCOPE
  
```



```

1115                                     ;; THIS TEST VERIFYS MATCH DETECT & DATA RDY
1116                                     ;; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1117                                     ;; BY OBSERVING RECACT BIT
1118                                     ;; IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1119                                     ;; * DEPENDENT ON MONITOR .....
1120                                     ;; IF ONE SYNC STRAP IS SELECTED IT WILL
1121                                     ;; ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1122                                     ;; ASSERT
1123                                     ;; MODE: SYNC INTERNAL
1124                                     ;; LENGTH: SIX
1125                                     ;; SYNC CHARACTER FOR MATCH: B/C
1126                                     ;; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1127
1128 002676 012767 000002 176222 TST2:  MOV    #2,TSTNO           ;SAVE THIS
1129 002704 012767 003224 176204      MOV    #TST3,NEXT        ;GO TO THIS TEST WHEN THRU
1130 002712 012767 003026 176200      MOV    #3$,LOCK         ;SET UP FOR SCOPE LOOP
1131 002720 052777 000400 007146      BIS    #MRES,ATXCSR     ;MASTER RESET
1132 002726 016703 007132              MOV    RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1133                                     ;SET SYNC INTERNAL, SIX NO PARITY, 0 SYNC REGISTER
1134 002732 012704 032000              MOV    #SYNINT!SIX!NOPAR,R4 ;CREATE PARAMETERS
1135 002736 012777 004020 007130 1$:   MOV    #MINT!SEND,ATXCSR ;SET SEND & MAINT INTER
1136 002744 010477 007120              MOV    R4,@PARCSR      ;LOAD CSR
1137 002750 052777 000020 007102      BIS    #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1138                                     ;POKE CLK TO GET INTO SYNCHRONIZATION
1139                                     ;BOTH THE LOGIC & RECEIVER
1140 002756 052777 020000 007110      BIS    #CLK,@TXCSR     ;POKE CLK UP
1141 002764 042777 020000 007102      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1142 002772 110477 007102              MOV    R4,@TXDBUF     ;LOAD DATA CHARACTER
1143                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1144 002776 052777 020000 007070      BIS    #CLK,@TXCSR     ;POKE CLK UP
1145 003004 042777 020000 007062      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1146 003012 032777 004000 007040      BIT    #RECACT,@RXCSR ;RECACT ?
1147 003020 001401              BEQ    2$
1148 003022 104000              HLT
1149                                     ;RECACT SHOULD NOT BE SET
1149 003024 2$:
1150 003024 000404              BR    4$
1151 003026 010477 007036 3$:   MOV    R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
1152 003032 110477 007042              MOV    R4,@TXDBUF     ;LOAD SYNC CHAR
1153 003036 012767 000002 176076 4$:   MOV    #2,COUNT        ;# OF SYNC CHARS
1154 003044 005777 007024 5$:   TST    @TXCSR ;DNA ?
1155 003050 100001              BPL    6$             ;BR IF NOT SET
1156 003052 104000              HLT
1157                                     ;DNA SHOULD NOT BE SET OR....
1157 003054 6$:
1158                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1159 003054 012767 000006 176056      MOV    #6,SHIFT        ;# OF SHIFTS
1160 003062 7$:
1161 003062 052777 020000 007004      BIS    #CLK,@TXCSR     ;POKE CLK UP
1162 003070 042777 020000 006776      BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1163 003076 005367 176036              DEC    SHIFT ;# OF SHIFTS
1164 003102 001367              SNE    7$
1165 003104 005367 176032              DEC    COUNT ;# OF SYNC CHARS
1166 003110 001403              BEQ    8$
1167                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1168 003112 105767 176060              TSTB   SYNCNO
1169 003116 100752              BMI    5$             ;TWO SYNC CHARACTERS..
1170 003120 032777 004000 006732 8$:   BIT    #RECACT,@RXCSR ;RECACT ?
    
```

M02

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 28
 DZDUF.C.P11 05-AUG-76 00:00

SEQ 0025

```

1171 003126 001001      BNE      9S
1172 003130 104000      HLT
1173 003132              9S:
1174                      ;REACT FAILED TO SET, POSSIBLE
1175                      ; THAT THE RECEIVER FAILED TO MATCH
1176                      ; THE SYNC CHARACTER
1176 003132 017701 006726  MOV      @RXDBUF,R1      ;SAVE ACTUAL
1177 003136 010400      MOVB     R4,R0          ;SAVE EXPECTED
1178 003140 042700 177400  BIC      @177400,R0     ;CLR UPPER BYTE
1179 003144 020001      CMP      R0,R1          ;DO THEY COMPARE ?
1180 003146 001401      BEQ     10S
1181 003150 104002      HLT      2
1182 003152              10S:
1183                      ;... IT PROBABLY IS A TRANSMITTER ERROR
1184                      ; HOWEVER,... IF ONLY THIS FAILED IT
1185                      ; PROBABLY IS A RECEIVER ERROR
1186 003152 104401      SCOPE1
1187                      ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1188                      ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1189                      ;TXDBUF
1190 003154 052777 020000 006712  BIS      @CLK,@TXCSR    ;POKE CLK UP
1191 003162 005777 006706      TST     @TXCSR ;DNA?
1192 003166 100401      BMI     11S
1193 003170 104000      HLT      ;DNA DID NOT ASSERT
1194 003172              11S:
1195                      ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1196 003172 052777 000400 006674  BIS      @MRESET,@TXCSR ;MASTER RESET
1197 003200 032777 000020 006652  BIT      @SYNSCH,@RXCSR ;SYNC SEARCH = 0 ?
1198 003206 001401      BEQ     12S
1199 003210 104000      HLT      ;SYNC SEARCH SHOULD BE NOT SET
1200 003212              12S:
1201 003212 005204      INC     R4
1202 003214 122704 000100  CMPB    @100,R4 ;IS THIS THE LAST CHARACTER ?
1203 003220 001246      BNE     1S
1204 003222 104400      HLT      ;NO
SCOPE
  
```

```

1205      ; THIS TEST VERIFYS MATCH DETECT & DATA RDY
1206      ; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1207      ; BY OBSERVING RECACT BIT
1208      ; IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1209      ; * DEPENDENT ON MONITOR
1210      ; IF ONE SYNC STRAP IS SELECTED IT WILL
1211      ; ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1212      ; ASSERT
1213      ; MODE: SYNC INTERNAL
1214      ; LENGTH: SEVEN
1215      ; SYNC CHARACTER FOR MATCH: B/C
1216      ; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1217
1218 003224 012767 000003 175674 TST3:  MOV    #3,TSTNO      ;SAVE THIS
1219 003232 012767 003552 175656      MOV    #TST4,NEXT    ;GO TO THIS TEST WHEN THRU
1220 003240 012767 003354 175652      MOV    #3$,LOCK      ;SET UP FOR SCOPE LOOP
1221 003246 052777 000400 006620      BIS    #MRESET,@TXCSR ;MASTER RESET
1222 003254 016703 006604      MOV    RXDBUF,R3     ;SET UP FOR ERROR MESSAGE
1223      ;SET SYNC INTERNAL, SEVEN, NO PARITY, 0 SYNC REGISTER
1224 003260 012704 034000      MOV    #SYNINT!SEVEN!NOPAR,R4 ;CREATE PARAMETERS
1225 003264 012777 004020 006602 1$:    MOV    #MINT!SEND,@TXCSR ;SET SEND & MAINT INTER
1226 003272 010477 006572      MOV    R4,@PARCSR    ;LOAD CSR
1227 003276 052777 000020 006554      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
1228      ;POKE CLK TO GET INTO SYNCHRONIZATION
1229      ;BOTH THE LOGIC & RECEIVER
1230 003304 052777 020000 006562      BIS    #CLK,@TXCSR   ;POKE CLK UP
1231 003312 042777 020000 006554      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
1232 003320 110477 006554      MOV    R4,@TXDBUF    ;LOAD DATA CHARACTER
1233      ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1234 003324 052777 020000 006542      BIS    #CLK,@TXCSR   ;POKE CLK UP
1235 003332 042777 020000 006534      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
1236 003340 032777 004000 006512      BIT    #RECACT,@RXCSR ;RECACT ?
1237 003346 001401      BEQ    2$
1238 003350 104000      HLT
1239      ;RECACT SHOULD NOT BE SET
1240 003352 000404 2$:    BR     4$
1241 003354 010477 006510 3$:    MOV    R4,@PARCSR    ;LOAD PARCSR WITH PARAMETERS
1242 003360 110477 006514      MOV    R4,@TXDBUF    ;LOAD SYNC CHAR
1243 003364 012767 000002 175550 4$:    MOV    #2,COUNT     ;# OF SYNC CHARS
1244 003372 005777 006476 5$:    TST    @TXCSR      ;DNA ?
1245 003376 100001      BPL    6$           ;BR IF NOT SET
1246 003400 104000      HLT
1247      ;DNA SHOULD NOT BE SET OR....
1248 6$:
1249 003402 012767 000007 175530      MOV    #7,SHIFT     ;# OF SHIFTS
1250 003410 7$:
1251 003410 052777 020000 006456      BIS    #CLK,@TXCSR   ;POKE CLK UP
1252 003416 042777 020000 006450      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
1253 003424 005367 175510      DEC    SHIFT        ;# OF SHIFTS
1254 003430 001367 7$:
1255 003432 005367 175504      DEC    COUNT        ;# OF SYNC CHARS
1256 003436 001403 8$:
1257      ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1258 003440 105767 175532      TST    SYNCNO
1259 003444 100752 5$:
1260 003446 032777 004000 006404 8$:    BIT    #RECACT,@RXCSR ;RECACT ?
    
```

```

1261 003454 001001      BNE      95
1262 003456 104000      HLT
1263 003460          95:
1264
1265
1266 003460 017701 006400      MOV      @RXDBUF,A1      ;SAVE ACTUAL
1267 003464 C10400      MOV      R4,R0          ;SAVE EXPECTED
1268 003466 042700 177400      BIC      @177400,R0      ;CLR UPPER BYTE
1269 003472 020001      CMP      R0,R1          ;DO THEY COMPARE ?
1270 003474 001401      BEQ     105
1271 003476 104002      HLT
1272 003500          105:
1273
1274
1275
1276 003500 104401      SCOPE1
1277
1278 ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
1279 ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1280 ;TXDBUF
1280 003512 052777 020000 006364      BIS      @CLK,@TXCSR      ;POKE CLK UP
1281 003510 C05777 006360      TST      @TXCSR      ;DNA?
1282 003514 105401      BMI     115
1283 003516 104000      HLT
1284 003520          115:
1285
1286 ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1286 003520 052777 000400 006346      BIS      @MRESET,@TXCSR ;MASTER RESET
1287 003526 032777 000020 006324      BIT      @SYNSCH,@TXCSR ;SYNC SEARCH = 0 ?
1288 003534 001401      BEQ     125
1289 003536 104000      HLT
1290
1291          125:
1291 003540 005204      INC      R4
1292 003542 122704 000200      CMPB    @200,R4 ;IS THIS THE LAST CHARACTER ?
1293 003546 001246      BNE     15
1294 003550 104400      SCOPE
  
```

```

1295      ; THIS TEST VERIFYS MATCH DETECT & DATA RDY
1296      ; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1297      ; BY OBSERVING RECACT BIT
1298      ; IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1299      ; * DEPENDENT ON MONITOR
1300      ; IF ONE SYNC STRAP IS SELECTED IT WILL
1301      ; ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1302      ; ASSERT
1303      ; MODE: SYNC INTERNAL
1304      ; LENGTH: EIGHT
1305      ; SYNC CHARACTER FOR MATCH: B/C
1306      ; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1307
1308 003552 012767 000004 175346 TST4: MOV     #4,TSTNO      ;SAVE THIS
1309 003553 012767 004100 175330      MOV     @TSTS NEXT    ;GO TO THIS TEST WHEN THRU
1310 003554 012767 003702 175324      MOV     #39,LOCK     ;SET UP FOR SCOPE LOOP
1311 003555 052777 000400 006272      BIS     @MRES, @TXCSR ;MASTER RESET
1312 003602 016703 006256      MOV     @RXDJF, R3    ;SET UP FOR ERROR MESSAGE
1313      ;SET SYNC INTERNAL, EIGHT NO PARITY 0 SYNC REGISTER
1314 003606 012704 035000      MOV     @SYNINT!EIGHT!NOPAR, R4 ;CREATE P. PARAMETERS
1315 003612 012777 004020 006254 18:      MOV     @MAINT!SEND, @TXCSR ;SET SEND & MAINT INTER
1316 003620 010477 005244      MOV     R4, @PARCSR   ;LOAD CSR
1317 003624 052777 000020 006226      BIS     @SYNSCH, @TXCSR ;SET SYNC SEARCH
1318      ;POKE CLK TO GET INTO SYNCHRONIZATION
1319      ;BOTH THE LOGIC & RECEIVER
1320 003632 052777 020000 006234      BIS     @CLK, @TXCSR  ;POKE CLK UP
1321 003640 042777 020000 006226      BIC     @CLK, @TXCSR  ;POKE CLK DOWN
1322 003646 110477 006226      MOV     R4, @TXCSUF   ;LOAD DATA CHARACTER
1323      ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1324 003652 052777 020000 006214      BIS     @CLK, @TXCSR  ;POKE CLK UP
1325 003653 042777 020000 005206      BIC     @CLK, @TXCSR  ;POKE CLK DOWN
1326 003656 032777 004000 006164      BIT     @RECACT, @TXCSR ;RECACT ?
1327 003657 001401      BEQ     Z8            ;RECACT SHOULD NOT BE SET
1328 003676 104000      HLT
1329
1330      Z8:      BR     #48
1331 003700 000404      BR     #48
1332 003702 010477 005162 38:      MOV     R4, @PARCSR   ;LOAD PARCSR WITH PARAMETERS
1333 003706 110477 006166      MOV     R4, @TXCSUF   ;LOAD SYNC CHAR
1334 003712 012767 000002 175222 48:      MOV     #2, COUNT    ;# OF SYNC CHARS
1335 003720 052777 006150 58:      TST     @TXCSR ;DNA ?
1336 003724 100701      BPL     #68          ;BR IF NOT SET
1337 003726 104000      HLT                ;DNA SHOULD NOT BE SET OR....
1338
1339      68:
1340      ;IT SHOULD BE CLEARED FROM PREVIOUS READ
1341 003730 012767 000010 175202      MOV     @8., SHIFT   ;# OF SHIFTS
1342
1343      78:
1344      BIS     @CLK, @TXCSR ;POKE CLK UP
1345      BIC     @CLK, @TXCSR ;POKE CLK DOWN
1346      DEC     SHIFT     ;# OF SHIFTS
1347      BNE     #78
1348      DEC     COUNT    ;# OF SYNC CHARS
1349      BEQ     #88
1350      ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
1351      TST     SYNCNO
1352      BMI     #58      ;TWO SYNC CHARACTERS..
1353      BIT     @RECACT, @TXCSR ;RECACT ?

```

```

1351 004002 001001      BNE      95
1352 004004 104000      HLT
1353 004006          95:
1354          ;REACT FAILED TO SET, POSSIBLE
1355          ; THAT THE RECEIVER FAILED TO MATCH
1356          ; THE SYNC CHARACTER
1357 004006 017701 006052      MOV      @RXDBUF, R1 ;SAVE ACTUAL
1358 004012 010400      MOV      R4, R0      ;SAVE EXPECTED
1359 004014 042700 177400      B       @177400, R0 ;CLR UPPER BYTE
1360 004020 020001      CMP      R0, R1      ;DO THEY COMPARE ?
1361 004022 001401      BEQ      105
1362 004024 104002      HLT
1363          105:
1364          ; IF REACT FAILED ALONG WITH THIS
1365          ; ... IT PROBABLY IS A TRANSMITTER ERROR
1366          ; HOWEVER, IF ONLY THIS FAILED IT
1367          ; PROBABLY IS A RECEIVER ERROR
1368 004026 104401      SCOPE
1369          ;POKE CLK TO SEE DNA... DNA COMES UP ON THE FIRST
1370          ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
1371          ;TXDBUF
1372 004030 052777 020000 006036      BIS      @CLK, @TXCSR ;POKE CLK UP
1373 004036 005777 006032      TST      @TXCSR ;DNA?
1374 004042 104401      BMI      115
1375 004044 104000      HLT ;DNA DID NOT ASSERT
1376          115:
1377          ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
1378 004046 052777 000400 006020      BIS      @SET, @TXCSR ;MASTER RESET
1379 004054 005777 000020 005776      BIT      @SYNSCH, @TXCSR ;SYNC SEARCH = 0 ?
1380 004062 001401      BEQ      125
1381 004064 104000      HLT ;SYNC SEARCH SHOULD BE NOT SET
1382          125:
1383 004066 005204      INC      R4
1384 004070 122704 000000      CMPB    @0, R4 ;IS THIS THE LAST CHARACTER ?
1385 004074 001246      BNE
1386 004076 104400      SCOPE

```

1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500

004100 012767 000005 175020
004106 012767 004312 175002
004114 012777 000400 005752
004122 012777 000000 005740
004130 052777 000400 005736

004136 012777 004020 005730

004144 012777 027426 005716
004152 016703 005706
004158 005304
004160 110477 005714
004164 052777 000020 005666

004172 052777 020000 005674
004200 042777 020000 005656
004206 012767 000011 174724
004214 010400
004216 052777 020000 005650
004224 042777 020000 005642
004232 005367 174702
004240 022767 000003 174674
004248 001003
004256 005204
004264 110477 005624
004272 005767 174660
004280 001356
004288 106777 005572
004296 100401
004304 104000

004310 017701 005566
020001
001401
104002

105704
001337
104400

THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
BOTH THE TRANSMITTER AND RECEIVER LOGIC
MODE: SYNC EXTERNAL (SYNEXT)
LENGTH: EIGHT PLUS PARITY
PARITY: EVEPAR
MAINT. MODE: MINT

TSTS: MOV #5, TSTNO ; SAVE THIS
MOV #7, 6, NEXT ; GO TO THIS TEST WHEN THRU
BIS #0, @TXCSR ; MASTER RESET
MOV #5, @TXCSR ; SET THE MODE
BIS @MRESET, @TXCSR ; MASTER RESET

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

; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
MOV #SYNEXT, @TXCSR ; EIGHT, EVEPAR, @PARCSR
MOV @RXDBUF, R3 ; SETUP FOR ERROR MSG
CLR R4 ; FOR DATA CHAR CREATION
MOV @RXDBUF, R4 ; LOAD CHARACTER
BIS #SYNEXT, @TXCSR ; SET SEARCH SYNC

; GET INTO SYNCHRONIZATION
BIS @CLK, @TXCSR ; POKE CLK UP
BIC @CLK, @TXCSR ; POKE CLK DOWN
18: MOV #9, SHIFT ; # OF SHIFTS
MOV R4, R0 ; EXPECTED
28: BIS @CLK, @TXCSR ; POKE CLK UP
BIC @CLK, @TXCSR ; POKE CLK DOWN
DEC SHIFT ; # OF SHIFTS
CMP #3, SHIFT ; TIME TO LOAD NEXT CHAR ?
BNE #0, NO ?
INC R4 ; GENERATE NEXT CHAR
MOV @RXDBUF, R4 ; LOAD NEXT CHARACTER
38: TST SHIFT ; IS IT 0 ?
BNE #0, NO ?
TSTB @TXCSR ; RXDONE = 1 ?
BMI #0, NO ?
HLT ; RXDONE SHOULD BE SET
58: MOV @RXDBUF, R1 ; ACTUAL
CMP R0, R1 ; COMPARE EXP VS ACT
BNE #0, NO ?
HLT ; CHARACTERS SHOULD COMPARE
68: TSTB R4 ; LAST CHARACTER ?
BNE #0, NO ?
48: SCOPE

F03

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 34
 DZDUF.C.P11 05-AUG-76 00:00

SEQ 0031

```

1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484

      ; THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
      ; BOTH THE TRANSMITTER AND RECEIVER LOGIC
      ; MODE: SYNC EXTERNAL (SYNEXT)
      ; LENGTH: EIGHT PLUS PARITY
      ; PARITY: 000PAR
      ; MAINT. MODE: MINT
TST6:  MOV    #6,TSTNO      ;SAVE THIS
      MOV    #TST7,NEXT    ;GO TO THIS TEST WHEN THRU
      BIS    #ESET,@TXCSR  ;MASTER RESET
      MOV    #SYNEXT,@PARCSR ;SET THE MODE
      BIS    #MRESET,@TXCSR ;MASTER RESET

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

      ;SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
      MOV    #SYNEXT!EIGHT!000PAR!26,@PARCSR
      MOV    @RXDBUF,R3    ;SETUP FOR ERROR MSG
      CLR    R4            ;FOR DATA CHAR CREATION
      MOVB   R4,@TXDBUF    ;LOAD CHARACTER
      BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
      ;GET INTO SYNCHRONIZATION
      BIS    #CLK,@TXCSR   ;POKE CLK UP
      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
18:    MOV    #9,SHIFT     ;# OF SHIFTS
      MOV    R4,R0        ;EXPECTED
28:    BIS    #CLK,@TXCSR  ;POKE CLK UP
      BIC    #CLK,@TXCSR  ;POKE CLK DOWN
      DEC    SHIFT        ;# OF SHIFTS
      CMP    #3,SHIFT     ;TIME TO LOAD NEXT CHAR ?
      BNE    #3           ;NO ?
      INC    R4           ;GENERATE NEXT CHAR
      MOVB   R4,@TXDBUF   ;LOAD NEXT CHARACTER
38:    TST    SHIFT       ;IS IT 0 ?
      BNE    #28
      TSTB   @RXCSR       ;RXDONE = 1 ?
      BMI    #58
      HLT                    ;RXDONE SHOULD BE SET
58:    MOV    @RXDBUF,R1   ;ACTUAL
      CMP    R0,R1        ;COMPARE EXP VS ACT
      BEQ    #68
      HLT                    ;CHARACTERS SHOULD COMPARE
68:    TSTB   R4           ;LAST CHARACTER ?
      BNE    #18         ;NO
48:    SCOPE

```



```

1495          : THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1496          : BOTH THE TRANSMITTER AND RECEIVER LOGIC
1497          : MODE: SYNC EXTERNAL (SYNEXT)
1498          : LENGTH: EIGHT PLUS PARITY
1499          : PARITY: EVEPAR
1500          : MAINT. MODE: NEXT
1501          :
1502 004524 012767 000007 174374 TST7: MOV      #7,TSTNO          ;SAVE THIS
1503 004532 012767 000004 174356      MOV      #TSTB,NEXT          ;GO TO THIS TEST WHEN THRU
1504 004540 105767 174437      TSTB     JMRBY          ;JUMP AROUND TEST ?
1505 004544 100116      BPL      #4          ;YES ?
1506 004546 052777 000400 005320      BIS      #MRESET,@TXCSR ;MASTER RESET
1507 004554 012777 020000 005306      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1508 004562 052777 000400 005304      BIS      #MRESET,@TXCSR ;MASTER RESET
1509          :
1510          ;SET MAINTENANCE MODE & SEND
1511          ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1512 004570 012777 010020 005276      MOV      #NEXT!SEND,@TXCSR
1513          :
1514          ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1515 004576 012777 027426 005264      MOV      #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
1516 004604 016703 005254      MOV      RXDBUF,R3          ;SETUP FOR ERROR MSG
1517 004610 005004      CLR      R4          ;FOR DATA CHAR CREATION
1518 004612 110477 005262      MOVB     R4,@TXDBUF        ;LOAD CHARACTER
1519 004616 052777 000020 005234      BIS      #SYNSCH,@RXCSR    ;SET SEARCH SYNC
1520          :
1521          ;GET INTO SYNCHRONIZATION
1522 004624 052777 020000 005242      BIS      #CLK,@TXCSR       ;POKE CLK UP
1523          ;WAIT FOR CABLE & DRIVER DELAYS
1524 004632 016702 174300      MOV      HOLD,R2          ;WAIT THIS AMT
1525          :
1526          64S:
1527 004636 005302      DEC      R2          ;WAIT
1528 004640 001376      BNE      64S
1529          ;EXIT...
1530 004642 042777 020000 005224      BIC      #CLK,@TXCSR       ;POKE CLK DOWN
1531          ;WAIT FOR CABLE & DRIVER DELAYS
1532 004650 016702 174262      MOV      HOLD,R2          ;WAIT THIS AMT
1533          :
1534          65S:
1535 004654 005302      DEC      R2          ;WAIT
1536 004656 001376      BNE      65S
1537          ;EXIT...
1538 004660 012767 000011 174252 1S:  MOV      #9,SHIFT          ;# OF SHIFTS
1539 004666 010400      MOV      R4,R0          ;EXPECTED
1540          :
1541          2S:
1542 004670 052777 020000 005176      BIS      #CLK,@TXCSR       ;POKE CLK UP
1543          ;WAIT FOR CABLE & DRIVER DELAYS
1544 004676 016702 174234      MOV      HOLD,R2          ;WAIT THIS AMT
1545          :
1546          66S:
1547 004702 005302      DEC      R2          ;WAIT
1548 004704 001376      BNE      66S
1549          ;EXIT...
1550 004706 042777 020000 005160      BIC      #CLK,@TXCSR       ;POKE CLK DOWN
1551          ;WAIT FOR CABLE & DRIVER DELAYS
1552 004714 016702 174216      MOV      HOLD,R2          ;WAIT THIS AMT
1553          :
1554          67S:
1555 004720 005302      DEC      R2          ;WAIT
1556 004722 001376      BNE      67S
    
```

H03

DZDUFC-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 36
DZDUFC.P11 05-AUG-76 00:00

SEQ 0033

```
1541                                     :EXIT...
1542 004724 005367 174210                DEC      SHIFT      ;# OF SHIFTS
1543 004730 022767 000003 174202        CMP      R3,SHIFT  ;TIME TO LOAD NEXT CHAR ?
1544 004736 011073                       BNE      R3        ;NO ?
1545 004740 000000                       INC      R4        ;GENERATE NEXT CHAR
1546 004742 110477 005132                MOVW    R4,@TXBUF  ;LOAD NEXT CHARACTER
1547 004746 005767 174166                TST     SHIFT     ;IS IT 0 ?
1548 004752 001345                       BNE      R3
1549 004754 105777 005100                TSTB   @RXCSR    ;RXDONE = 1 ?
1550 004760 100401                       BMI     R3
1551 004762 104000                       HLT
1552                                     ;RXDONE SHOULD BE SET
1553                                     55:
1554 004764 017701 005074                MOV     @RXBUF,R1 ;ACTUAL
1555 004770 000001                       CMP     R0,R1    ;COMPARE EXP VS ACT
1556 004772 001401                       BEQ     R3
1557 004774 104002                       HLT
1558                                     65:
1559 004776 105704                       ;CHECK OUT MODEM BYPASS JUMPER
1560 005000 001327                       TSTB   R4        ;LAST CHARACTER ?
1561 005002 104400                       BNE     R3
1561                                     45:
SCOPE
```

```

1563          ;: THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
1564          ;: BOTH THE TRANSMITTER AND RECEIVER LOGIC
1565          ;: MODE: SYNC EXTERNAL (SYNEXT)
1566          ;: LENGTH: EIGHT PLUS PARITY
1567          ;: PARITY: 000PAR
1568          ;: MODE: NEXT
1569          ;:
1570          ;:
1571          ;:
1572          ;:
1573          ;:
1574          ;:
1575          ;:
1576          ;:
1577          ;:
1578          ;:
1579          ;:
1580          ;:
1581          ;:
1582          ;:
1583          ;:
1584          ;:
1585          ;:
1586          ;:
1587          ;:
1588          ;:
1589          ;:
1590          ;:
1591          ;:
1592          ;:
1593          ;:
1594          ;:
1595          ;:
1596          ;:
1597          ;:
1598          ;:
1599          ;:
1600          ;:
1601          ;:
1602          ;:
1603          ;:
1604          ;:
1605          ;:
1606          ;:
1607          ;:
1608          ;:
1609          ;:
1610          ;:
1611          ;:
1612          ;:
1613          ;:
1614          ;:
1615          ;:
1616          ;:
1617          ;:
1618          ;:
1619          ;:
1620          ;:
1621          ;:
1622          ;:
1623          ;:
1624          ;:
1625          ;:
1626          ;:
1627          ;:
1628          ;:
1629          ;:
1630          ;:
1631          ;:
1632          ;:
1633          ;:
1634          ;:
1635          ;:
1636          ;:
1637          ;:
1638          ;:
1639          ;:
1640          ;:
1641          ;:
1642          ;:
1643          ;:
1644          ;:
1645          ;:
1646          ;:
1647          ;:
1648          ;:
1649          ;:
1650          ;:
1651          ;:
1652          ;:
1653          ;:
1654          ;:
1655          ;:
1656          ;:
1657          ;:
1658          ;:
1659          ;:
1660          ;:
1661          ;:
1662          ;:
1663          ;:
1664          ;:
1665          ;:
1666          ;:
1667          ;:
1668          ;:
1669          ;:
1670          ;:
1671          ;:
1672          ;:
1673          ;:
1674          ;:
1675          ;:
1676          ;:
1677          ;:
1678          ;:
1679          ;:
1680          ;:
1681          ;:
1682          ;:
1683          ;:
1684          ;:
1685          ;:
1686          ;:
1687          ;:
1688          ;:
1689          ;:
1690          ;:
1691          ;:
1692          ;:
1693          ;:
1694          ;:
1695          ;:
1696          ;:
1697          ;:
1698          ;:
1699          ;:
1700          ;:
1701          ;:
1702          ;:
1703          ;:
1704          ;:
1705          ;:
1706          ;:
1707          ;:
1708          ;:
1709          ;:
1710          ;:
1711          ;:
1712          ;:
1713          ;:
1714          ;:
1715          ;:
1716          ;:
1717          ;:
1718          ;:
1719          ;:
1720          ;:
1721          ;:
1722          ;:
1723          ;:
1724          ;:
1725          ;:
1726          ;:
1727          ;:
1728          ;:
1729          ;:
1730          ;:
1731          ;:
1732          ;:
1733          ;:
1734          ;:
1735          ;:
1736          ;:
1737          ;:
1738          ;:
1739          ;:
1740          ;:
1741          ;:
1742          ;:
1743          ;:
1744          ;:
1745          ;:
1746          ;:
1747          ;:
1748          ;:
1749          ;:
1750          ;:
1751          ;:
1752          ;:
1753          ;:
1754          ;:
1755          ;:
1756          ;:
1757          ;:
1758          ;:
1759          ;:
1760          ;:
1761          ;:
1762          ;:
1763          ;:
1764          ;:
1765          ;:
1766          ;:
1767          ;:
1768          ;:
1769          ;:
1770          ;:
1771          ;:
1772          ;:
1773          ;:
1774          ;:
1775          ;:
1776          ;:
1777          ;:
1778          ;:
1779          ;:
1780          ;:
1781          ;:
1782          ;:
1783          ;:
1784          ;:
1785          ;:
1786          ;:
1787          ;:
1788          ;:
1789          ;:
1790          ;:
1791          ;:
1792          ;:
1793          ;:
1794          ;:
1795          ;:
1796          ;:
1797          ;:
1798          ;:
1799          ;:
1800          ;:
1801          ;:
1802          ;:
1803          ;:
1804          ;:
1805          ;:
1806          ;:
1807          ;:
1808          ;:
1809          ;:
1810          ;:
1811          ;:
1812          ;:
1813          ;:
1814          ;:
1815          ;:
1816          ;:
1817          ;:
1818          ;:
1819          ;:
1820          ;:
1821          ;:
1822          ;:
1823          ;:
1824          ;:
1825          ;:
1826          ;:
1827          ;:
1828          ;:
1829          ;:
1830          ;:
1831          ;:
1832          ;:
1833          ;:
1834          ;:
1835          ;:
1836          ;:
1837          ;:
1838          ;:
1839          ;:
1840          ;:
1841          ;:
1842          ;:
1843          ;:
1844          ;:
1845          ;:
1846          ;:
1847          ;:
1848          ;:
1849          ;:
1850          ;:
1851          ;:
1852          ;:
1853          ;:
1854          ;:
1855          ;:
1856          ;:
1857          ;:
1858          ;:
1859          ;:
1860          ;:
1861          ;:
1862          ;:
1863          ;:
1864          ;:
1865          ;:
1866          ;:
1867          ;:
1868          ;:
1869          ;:
1870          ;:
1871          ;:
1872          ;:
1873          ;:
1874          ;:
1875          ;:
1876          ;:
1877          ;:
1878          ;:
1879          ;:
1880          ;:
1881          ;:
1882          ;:
1883          ;:
1884          ;:
1885          ;:
1886          ;:
1887          ;:
1888          ;:
1889          ;:
1890          ;:
1891          ;:
1892          ;:
1893          ;:
1894          ;:
1895          ;:
1896          ;:
1897          ;:
1898          ;:
1899          ;:
1900          ;:
1901          ;:
1902          ;:
1903          ;:
1904          ;:
1905          ;:
1906          ;:
1907          ;:
1908          ;:
1909          ;:
1910          ;:
1911          ;:
1912          ;:
1913          ;:
1914          ;:
1915          ;:
1916          ;:
1917          ;:
1918          ;:
1919          ;:
1920          ;:
1921          ;:
1922          ;:
1923          ;:
1924          ;:
1925          ;:
1926          ;:
1927          ;:
1928          ;:
1929          ;:
1930          ;:
1931          ;:
1932          ;:
1933          ;:
1934          ;:
1935          ;:
1936          ;:
1937          ;:
1938          ;:
1939          ;:
1940          ;:
1941          ;:
1942          ;:
1943          ;:
1944          ;:
1945          ;:
1946          ;:
1947          ;:
1948          ;:
1949          ;:
1950          ;:
1951          ;:
1952          ;:
1953          ;:
1954          ;:
1955          ;:
1956          ;:
1957          ;:
1958          ;:
1959          ;:
1960          ;:
1961          ;:
1962          ;:
1963          ;:
1964          ;:
1965          ;:
1966          ;:
1967          ;:
1968          ;:
1969          ;:
1970          ;:
1971          ;:
1972          ;:
1973          ;:
1974          ;:
1975          ;:
1976          ;:
1977          ;:
1978          ;:
1979          ;:
1980          ;:
1981          ;:
1982          ;:
1983          ;:
1984          ;:
1985          ;:
1986          ;:
1987          ;:
1988          ;:
1989          ;:
1990          ;:
1991          ;:
1992          ;:
1993          ;:
1994          ;:
1995          ;:
1996          ;:
1997          ;:
1998          ;:
1999          ;:
2000          ;:

```

J03

```

1618                                :EXIT...
1619 005204 005367 173730  DEC          SHIFT ;# OF SHIFTS
1620 00210 022767 000003 173722  CMP          R3,SHIFT ;TIME TO LOAD NEXT CHAR ?
1621 001003  BNE          3$ ;NO ?
1622 000104  INC          R4 ;GENERATE NEXT CHAR
1623 110477 004652  MOVB        R4,RXDBUF ;LOAD NEXT CHARACTER
1624 005767 173706 3$:  TST          SHIFT ;IS IT 0 ?
1625 001346  BNE          2$
1626 105777 004620  TSTB        RXCSR ;RXDONE = 1 ?
1627 104401  BMI          5$
1628 104000  HLT          ;RXDONE SHOULD BE SET
1629
1630 017701 004614 5$:  MOV          RXDBUF,R1 ;ACTUAL
1631 020001  CMP          R0,R1 ;COMPARE EXP VS ACT
1632 001401  BEQ          6$
1633 104002  HLT          2 ;CHARACTERS SHOULD COMPARE
1634 00256
1635
1636 005256 105704  :CHECK OUT MODEM BYPASS JUMPER
1637 001327  TSTB        R4 ;LAST CHARACTER ?
1638 005262 104400  BNE          1$ ;NO
SCOPE

```

K03

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 39
DZDUF.C.P11 05-AUG-76 00:00

SEQ 0036

```
1639
1640 ;END OF PASS
1641 ;TYPE NAME OF TEST
1642 ;UPDATE PASS COUNT
1643 ;CHECK FOR EXIT TO ACT-11
1644 ;RESTART TEST
1645
1646 005264 104402 .EOP: TYPE ;TYPE NAME OF TEST
1647 005266 010426 MEPASS
1648 005270 104410 005522 CONVRT ,OUTCRY
1649 005274 104402 010147 TYPE ,DEVICE
1650 005300 105767 173676 TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
1651 005304 001511 BEQ CCC ;NO, JUMP AROUND
1652 005306 005767 173704 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
1653 005312 001007 BNE RUNIT ;YES
1654 005314 104402 010161 TYPE ,MCON ;NO
1655 005320 016700 173672 MOV ACTREG,RO ;DISPLAY ACTREG
1656 005324 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
1657 ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
1658 005326 000167 173726 JMP .START ;START OVER AGAIN..... YOU DESELECTED EVERYTHING
1659 005332 012767 000010 173644 RUNIT: ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
1660 005340 012767 000010 173644 ZERO: ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
1661 005346 000241 CLC
1662 005350 006167 173644 ROL ROTADD ;UP DATE ROTATING POINTER
1663 005354 103410 BCS 2$ ;IS IT THE LAST DEVICE
1664 ;TO BE TESTED IN THIS PASS ?
1665 005356 036767 173636 173632 BIT ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
1666 005364 001762 BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
1667 005366 004767 000034 JSR PC,REPLAY ;CALCULATE NEW PARAMETERS
1668 005372 000167 000174 JMP RESTRT ;YES IT WAS ACTIVE, TEST THIS DEVICE
1669 005376 012767 000001 173614 2$: MOV #1,ROTADD ;OK! NOW SET UP ROTATING
1670 ;POINTER FOR NEXT MULTIPLE PASS
1671 005404 016767 173576 173572 MOV KEEPADD,BASEADD ;RESTORE BASE ADDRESS
1672 005412 016767 173576 173572 MOV KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
1673 005420 004767 000002 JSR PC,REPLAY ;CALC NEW PARAMETERS
1674 005424 000441 BR CCC ;JUMP AROUND REPLAY
1675 005426 016767 173552 004126 REPLAY: MOV BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
1676 005434 004767 003770 JSR PC,QUADOR ;CREATE NEW ADDRESSES
1677 005440 016767 173546 004476 MOV BASEIV,DURIV ;CREATE DURIV
1678 005446 062767 000002 173536 ADD #2,BASEIV
1679 005454 016767 173532 004434 MOV BASEIV,DURIS ;CREATE DURIS
1680 005462 062767 000002 173522 ADD #2,BASEIV
1681 005470 016767 173516 004412 MOV BASEIV,DUTIV ;CREATE DUTIV
1682 005476 062767 000002 173506 ADD #2,BASEIV
1683 005504 016767 173502 004400 MOV BASEIV,DUTIS ;CREATE DUTIS
1684 005512 016767 004366 173472 MOV DURIV,BASEIV ;RESTORE
1685 005520 000207 RTS PC
1686
1687 005522 000001 OUTCRY: 1
1688 005524 006 002 .BYTE 6,2
1689 005526 012060 RXCSR
1690
1691 005530 CCC:
1692 005530 005067 173400 CLR LSTERR ;CLEAR LAST ERROR PC
1693 005534 005067 173464 CLR ERRFLG ;CLEAR ERROR FLAG
1694 005540 005267 173364 INC PASCNT ;UPDATE PASS COUNT
```

L03

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 40
 DZDUFC.P11 05-AUG-76 00:00

SEQ 0037

```

1695 005544 016777 173360 173330      MOV      PASCNT,ALIGHTS      ;DISPLAY PASS COUNT
1696 005552 013701 000042                MOV      @#42,R1           ;CHECK FOR ACT-11 OR DOP
1697 005556 001405                BEQ      RESTRT           ;IF NOT, CONTINUE TESTING
1698 005560 000005                RESET
1699 005562 004711                LOGICAL: JSR      PC,(R1)
1700 005564 000240                NOP
1701 005566 000240                NOP
1702 005570 000240                NOP
1703 005572 012767 000340 172176  RESTRT: MOV      @#340,PS      ;PREVENT INTERRUPTS (PRIO: 7)
1704 005600 104413                CKSWR      ;CHECK FOR †G
1705 005602 012767 002350 173304      MOV      @TST1,RTRN
1706 005610 000167 174534                JMP      TST1
1707
1708                                ;SCOPE LOOP AND ITERATION HANDLER
1709
1710 005614                .SCOPE:
1711                ;**** START OF CODE FOR THE X OR TESTER ****
1712 005614 000424                BR      4$
1713                                ;IF RUNNING ON THE X OR TESTER CHANGE
1714 005616 013746 000004                MOV      @#4,-(SP)        ;THIS INSTRUCTION TO A "NOP"(NOP=240)
1715 005622 012737 005642 000004      MOV      @1$,@#4         ;SAVE CONTENTS OF ERROR VECTOR
1716 005630 005737 177060                TST      @#177060        ;SET FOR TIME OUT
1717 005634 012637 000004      MOV      (SP)+,@#4       ;TIME OUT ON X OR ?
1718 005640 000404                BR      2$              ;RESTORE ERROR VECTOR
1719 005642 022626                1$: CMP      (SP)+,(SP)+  ;GO TO NEXT TEST
1720 005644 012637 000004      MOV      (SP)+,@#4       ;CLEAR THE STACK AFTER A TIMEOUT
1721 005650 000403                BR      3$              ;RESTORE ERROR VECTOR
1722 005652 016767 173240 173234  2$: MOV      NEXT,RTRN      ;LOOP ON PRESENT TEST
1723 005660 016716 173230  3$: MOV      RTRN,(SP)     ;SET UP NEXT TEST IN RTRN
1724 005664 000002                RTI                    ;SET UP STACK FOR RTI
1725 005666                4$: ;**** END OF CODE FOR THE X OR TESTER ****
1726 005666 104413                CKSWR      ;CHECK FOR †G
1727 005670 032777 040000 173202  TTST: BIT      @SW14,@SWR    ;LOOP ON CURRENT TEST ?
1728 005676 001407                BEQ      1$
1729 005700 000432                BR      3$
1730 005702 105777 173176                TSTB     @TKCSR          ;TEST TTY FLAG
1731 005706 100027                BPL      3$
1732 005710 017700 173172                MOV      @TKDBR,R0      ;CLR DONE BIT
1733 005714 000412                BR      2$              ;IF A TTY KEY IS STRUCK GO TO NEXT TST
1734 005716 032777 004000 173154  1$: BIT      @SW11,@SWR    ;INHIBIT ITERATIONS ?
1735 005724 001006                BNE      2$
1736 005726 005267 173172                INC      LPCNT
1737 005732 026767 173166 173162  2$: CMP      LPCNT,ICOUNT  ;CHECK FOR ITERATION CNT FINISH
1738 005740 101412                BLOS     3$
1739 005742 105067 173256                CLR      ERRFLG
1740 005746 005067 173152                CLR      LPCNT
1741 005752 012767 000005 173142      MOV      @5,ICOUNT      ;SET UP ITERATION COUNT
1742 005760 016767 173132 173126      MOV      NEXT,RTRN      ;SET UP NEXT TEST IN RTRN
1743 005766 016716 173122  3$: MOV      RTRN,(SP)     ;SET UP STACK FOR RTI
1744 005772 000002                RTI
1745 005774 001407                BRW: 1407                ;RESTORE "BEQ 1$" INSTRUCTION
1746 005776 000432                BRX: 432                ;RESTORE "BR 3$" INSTRUCTION
1747
1748                                ;CHECK FOR FREEZE ON CURRENT DATA
1749
1750 006000 104413                .SCOPE: CKSWR          ;CHECK FOR †G

```

M03

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 41
 DZDUF.C.P11 05-AUG-76 00:00

SEQ 0038

```

1751 006002 032777 001000 173070      BIT      #SW09,@SWR
1752 006010 001402                BEQ      1$
1753 006012 016716 173102                MOV      LOCK,(SP)
1754 006016 000002                RTI
1755
1756                                ;TELETYPE OUTPUT ROUTINE
1757
1758 006020 010546                .TYPE:  MOV      R5,-(SP)
1759 006022 017605 000002                MOV      @2(SP),R5
1760 006026 062766 000002 000002                ADD      #2,2(SP)
1761 006034 105715                1$:      TSTB     (R5)                ;LOOK FOR "0"
1762 006036 001406                BEQ      3$
1763 006040 105777 173044                2$:      TSTB     @TPCSR                ;TEST DONE BIT
1764 006044 100375                BPL      2$
1765 006046 112577 173040                MOVB     (R5)+,@TPDBR                ;TYPE CHAR
1766 006052 000770                BR       1$                ;DO IT AGAIN UNTIL "0" IS SEEN
1767 006054 012605                3$:      MOV      (SP)+,R5
1768 006056 000002                RTI
1769
1770                                ;ASCII STRING INPUT ROUTINE
1771
1772 006060 010346                .INSTR: MOV      R3,-(SP)
1773 006062 010446                MOV      R4,-(SP)
1774 006064 017667 000004 000010                MOV      @4(SP),MSG
1775 006072 062766 000002 000004                ADD      #2,4(SP)
1776 006100 104402                .INST1: TYPE
1777 006102 000000                .MSG:   0
1778 006104 012704 011214                MOV      @INBUF,R4
1779 006110 012703 000007                MOV      #7,R3
1780 006114 105777 172764                1$:      TSTB     @TKCSR
1781 006120 100375                BPL      1$
1782 006122 117714 172760                MOVB     @TKDBR,(R4)
1783 006126 142714 000200                BICB     #200,(R4)
1784 006132 121427 000025                CMPB     (R4),#25                ;IS IT <U>
1785 006136 001003                BNE      200$
1786 006140 104402 010336                TYPE,MCRLF
1787 006144 000755                BR       .INST1
1788 006146 122427 000015                200$:   CMPB     (R4)+,#15
1789 006152 001423                BEQ      INSTR2
1790 006154 117777 172726 172730                MOVB     @TKDBR,@TPDBR
1791 006162 105777 172722                2$:      TSTB     @TPCSR
1792 006166 100375                BPL      2$
1793 006170 005303                DEC      R3
1794 006172 001350                BNE      1$
1795 006174 000702                BR       .INSTG
1796 006176 010346                .INSTE: MOV      R3,-(SP)
1797 006180 010446                .INSTG: MOV      R4,-(SP)
1798 006182 104402                .INSTG: TYPE
1799 006184 010332                MOV      R4,R3
1800 006186 005737 007474                TST      @RDSW
1801 006188 001402                BEQ      400$
1802 006190 104402 010336                TYPE,MCRLF
1803 006192 000727                400$:   BR       .INST1
1804 006194 012604                INSTR2: MOV      (SP)+,R4
1805 006196 012603                MOV      (SP)+,R3
1806 006198 000002                RTI

```

```

1807
1808
1809
1810 006230 010546
1811 006232 010446
1812 006234 016605 000004
1813 006240 012567 000170
1814 006244 012567 000166
1815 006250 012567 000164
1816 006254 112567 000162
1817 006260 112567 000157
1818 006264 010566 000004
1819 006270 005005
1820 006272 012704 011214
1821 006276 122714 000015
1822 006302 001420
1823 006304 121427 000060
1824 006310 002415
1825 006312 121427 000067
1826 006316 003012
1827 006320 142714 000060
1828 006324 152405
1829 006326 122714 000015
1830 006332 001414
1831 006334 006305
1832 006336 006305
1833 006340 006305
1834 006342 000760
1835 006344 122714 000015
1836 006350 001003
1837 006352 005737 007474
1838 006356 001023
1839 006360 104404
1840 006362 000742
1841
1842
1843
1844 006364 020567 000046
1845 006370 101365
1846 006372 020567 000036
1847 006376 103762
1848 006400 136705 000036
1849 006404 001357
1850
1851
1852
1853 006406 016704 000026
1854 006412 010524
1855 006414 002705 000002
1856 006420 100367 000017
1857 006424 001372
1858 006426 012604
1859 006430 012605
1860 006432 000002
1861 006434 000000
1862 006436 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)+, DEVAOR
MOV (R5)+, LOBITS
MOV (R5)+, AORCNT
MOV R5, 4(SP)
PARAM1: CLR R5
MOV #INBUF, R4
CMPB #15, (R4)
BEQ PARERR
IS: 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 IS
PARERR: CMPB #15, (R4) ; IS FIRST CHARACTER A <CR>
BNE 120$
TST #AORCNT ; IS CKSUM ROUTINE BEING USED
BNE PARTI
120$: INSTER
BR PARAM1
; TEST TO SEE IF NUMBER IS WITHIN LIMITS
LIMITS: CMP R5, HILIM
BHI PARERR
CMP R5, LOLIM
BLO PARERR
BITB LOBITS, R5
BNE PARERR
; STORE NUMBER AT SPECIFIED ADDRESS
IS: MOV DEVAOR, R4
MOV R5, (R4)+
ADD #2, R5
LECB AORCNT
BNE IS
PARTI: MOV (SP)+, R4
MOV (SP)+, R5
RTI
LOLIM: 0
HILIM: 0

```



```

1863 006440 000000          DEVAR: 0
1864 006442 000000          LOBITS: 0
1865          006443          ADRCNT=LOBITS+1
1866
1867          ;SAVE PC OF TEST THAT FAILED AND RO-RS
1868
1869 006444 016667 000004 172522 .SAV05: MOV    4(SP),SAVPC
1870
1871          ;SAVE RO-RS
1872
1873 006452 010567 172512          SV05:  MOV    R5,SAVR5
1874 006456 010467 172504          MOV    R4,SAVR4
1875 006462 010367 172476          MOV    R3,SAVR3
1876 006466 010267 172470          MOV    R2,SAVR2
1877 006472 010167 172462          MOV    R1,SAVR1
1878 006476 010067 172454          MOV    R0,SAVR0
1879 006502 000002          RTI
1880
1881          ;RESTORE RO-RS
1882
1883 006504 016700 172446          .RES05: MOV    SAVR0,R0
1884 006510 016701 172444          MOV    SAVR1,R1
1885 006514 016702 172442          MOV    SAVR2,R2
1886 006520 016703 172440          MOV    SAVR3,R3
1887 006524 016704 172436          MOV    SAVR4,R4
1888 006530 016705 172434          MOV    SAVR5,R5
1889 006534 000002          RTI
1890
1891          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1892
1893 006536 104402          .CONVR: TYPE
1894 006540 010336          MCRLF
1895 006542 010046          .CNVRT: MOV    R0,-(SP)
1896 006544 010146          MOV    R1,-(SP)
1897 006546 010346          MOV    R2,-(SP)
1898 006550 010446          MOV    R3,-(SP)
1899 006552 010546          MOV    R4,-(SP)
1900 006554 017601 000012          MOV    @12(SP),R1
1901 006556 016767 012470 172362          MOV    TEMP,TEMP3
1902 006566 062766 000002 000012          ADD    #2,12(SP)
1903 006574 012167 000154          MOV    (R1)+,WPCNT
1904 006600 112167 000152          15:  MOVB  (R1)+,CHRCNT
1905 006604 112167 000147          MOVB  (R1)+,SPACNT
1906 006610 013167 000144          MOV    2(R1)+,BINWRD
1907 006614 016704 000140          25:  MOV    BINWRD,R4
1908 006620 116705 000132          MOVB  CHRCNT,R5
1909 006624 012700 011254          MOV    @TEMP,R0
1910 006630 010403          35:  MOV    R4,R3
1911 006632 042703 177770          BIC   @177770,R3
1912 006636 062703 000060          ADD    @060,R3
1913 006642 110320          MOVB  R3,(R0)+
1914 006644 006204          RSR   R4
1915 006646 042704 100000          BIC   @10000,R4
1916 006652 006204          RSR   R4
1917 006654 006204          RSR   R4
1918 006656 005306          DEC   R5

```

```

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

```

```

1919 006660 001363
1920 006662 012703 011314
1921 006664 114723
1922 006670 105367 000062
1923 006674 001374
1924 006676 105767 000055
1925 006702 001405
1926 006704 112723 000040
1927 006710 105367 000043
1928 006714 001373
1929 006716 105013
1930 006720 104402
1931 006722 011314
1932 006724 005367 000024
1933 006730 001323
1934 006732 016767 172212 002314
1935 006740 012625
1936 006742 012604
1937 006744 012603
1938 006746 012601
1939 006750 012620
1940 006752 000002
1941 006754 000000
1942 006756 000000
1943 006757 000000
1944 006760 000000

```

```

BNE 35
MOV #MDATA,R3
45: MOVB -(R0),(R3)+
DECB CHRCNT
BNE 45
TSTB SPACNT
BEQ 65
55: MOVB #040,(R3)+
DECB SPACNT
BNE 55
65: CLRB (R3)
TYPE
MDATA
DEC WROCNT
BNE 15
MOV TEMP3,TEMP
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R1
MOV (SP)+,R0
RTI

```

```

WROCNT: 0
CHRCNT: 0
SPACNT=CHRCNT+1
BINR0: 0

```

```

;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
;BUFFER TO THE CHARACTERS "N" AND "Y"
;IF THE CHARACTER IS "N" CLEAR THE FLAG
;IF THE CHARACTER IS "Y" SET THE FLAG

```

```

1945 006762 017605 000000
1946 006764 122767 000116 002220
1947 006774 001002
1948 006776 105015
1949 007000 000406
1950 007002 122767 000131 002204
1951 007010 001005
1952 007012 112715 177777
1953 007016 062716 000002
1954 007022 007002
1955 007024 104404
1956 007026 000755
1957 007030 011646
1958 007032 162716 000002
1959 007036 017616 000000
1960 007042 005316
1961 007044 042716 177001
1962 007050 062716 001226
1963 007054 017616 000000

```

```

.SETFLG: MOV @ (SP),R5
CMPB @'N',INBUF ;IS IT "N" ?
BNE 15
CLRB (R5) ;000
BR 25
15: CMPB @'Y',INBUF ;IS IT "Y" ?
BNE 35
MOVB @-1,(R5) ;377
25: ADD @2,(SP)
RTI
35: INSTER ;RETRY
BR .SETFLG

```

```

;TRAP DISPATCH SERVICE
;ARGUMENT OF TRAP IS EXTRACTED
;AND USED AS OFFSET TO OBTAIN POINTER
;TO SELECTED SUBROUTINE

```

```

.TRPSR: MOV (SP)-,(SP) ;GET PC OF RETURN
SUB @2,(SP) ;PC OF TRAP
MOV @ (SP),(SP) ;GET TRAP
TRPOK: RSL (SP) ;MULTIPLY TRAP ARG BY 2
BIC @177001,(SP) ;CLEAR UNWANTED BITS
ADD @.TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
MOV @ (SP),(SP) ;SUBROUTINE ADDRESS

```

```

1975 007060 000136      JMP      2(SP)+          ;GO TO SUBROUTINE
1976
1977                      ;ERROR HANDLER
1978
1979 007062 104413      .HLT:   CKSWR          ;CHECK FOR IG
1980 007064 032777 020000 172006      BIT      #SW13,2SWR    ;INHIBIT ERROR TYPE OUT ?
1981 007072 001061      BNE     HALTS
1982 007074 011657 172034      CMP     (SP),LSTERR
1983 007100 011434      BEQ     IS
1984 007102 011657 172026      MOV     (SP),LSTERR
1985 007106 105 67 172112     CLR     ERRFLG
1986 007112 104406      IS:     SAVOS
1987 007114 011 35      MOV     (SP),RS
1988 007116 162 35 000002     SUB     #2,RS
1989 007122 011504      MOV     (RS),R4
1990 007124 006304      ASL     R4
1991 007126 051504      ADD     (RS),R4
1992 007130 011 304      ASL     R4
1993 007132 0177001 177001     BIC     #177001,R4
1994 007136 062704 012030     ADD     #ERRTAB,R4
1995 007142 012467 000040     MOV     (R4)+,ERRMSG
1996 007146 012467 010046     MOV     (R4)+,DATAHD
1997 007152 011467 000054     MOV     (R4),DATAPP
1998 007156 105767 172042     TST     ERRFLG
1999 007162 001403      BEQ     TYPMSG
2000 007164 005767 000042     TST     DATAPP
2001 007170 001014      BNE     TYPDAT
2002 007172 104410      TYPMSG: CONVRT
2003 007174 007324      ERTAB0
2004 007176 112767 177777 172020     MOV     #-1,ERRFLG
2005 007204 104402      TYPE
2006 007206 000000      ERRMSG: 0
2007 007210 005767 000004     TST     DATAHD
2008 007214 001402      BEQ     TYPDAT
2009 007216 104402      TYPE
2010 007220 000000      DATAHD: 0
2011 007222 005767 000004     TYPDAT: TST     DATAPP
2012 007226 001402      BEQ     RESREG
2013 007230 104410      CONVRT
2014 007232 000000      DATAPP: 0
2015 007234 104407      RESREG: RESOS
2016 007236 005777 171636     HALTS:  TST     #R
2017 007242 100005      BPL     EXITER
2018 007244 010046      PUSHRO
2019 007246 016600 000002     MOV     2(SP),R0
2020 007250 000000      HALT
2021 007252 012600      POPRO
2022 007254 104413      EXITER: CKSWR
2023 007256 005267 171646      INC     ERRCNT          ;CHECK FOR IG
2024 007264 032777 000400 171606     BIT     #SW08,2SWR    ;LOOP ON ERROR ?
2025 007272 001007      BNE     IS
2026 007274 032777 002000 171576     BIT     #SW10,2SWR
2027 007202 001407      BEQ     #2          ;ESCAPE TO NEXT ON ERROR ?
2028 007204 016767 171606 171602     MOV     NEXT,RTN
2029 007312 012706 001100      IS:     MOV     #STACK,SP
2030 007316 000177 171572     JMP     @RTN          ;REINITIALIZE SP

```

E04

DZDUF-C MACY11 27(1006) 01-OCT-76 09:48 PAGE 46
 DZDUF.C.P11 05-AUG-76 00:00

SEQ 0043

```

2031 007322 000002          26: RTI
2032 007324 000001          LRTAB0: 1
2033 007326 006          .BYTE 6,2
2034 007330 001174          SAVPC
2035          ;ENTER HERE ON POWER FAILURE
2036
2037
2038 007332 010046          .PFAIL: MOV R0, -(SP) ;SAVE R0-R5 ON PROCESSOR STACK
2039 007334 010146          MOV R1, -(SP)
2040 007336 010246          MOV R2, -(SP)
2041 007340 010346          MOV R3, -(SP)
2042 007342 010446          MOV R4, -(SP)
2043 007344 010546          MOV R5, -(SP)
2044 007346 016746 170452  MOV SP, -(SP)
2045 007352 010667 171614  MOV SP, SAVSP ;SAVE STACK POINTER
2046 007356 012767 007370 170440  MOV #RESTART, 24 ;SET UP FOR POWER UP TRAP
2047 007364 001000          HALT ;HALT ON POWER DOWN NORMAL
2048 007366 000777          IS: BR IS
2049
2050          ;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
2051
2052 007370 016706 171576  RESTAR: MOV SAVSP, SP ;RESTORE STACK POINTER
2053 007374 012605          MOV (SP)+, R5 ;RESTORE R0-R5
2054 007376 012634          MOV (SP)+, R4
2055 007378 012603          MOV (SP)+, R3
2056 007380 012602          MOV (SP)+, R2
2057 007382 012601          MOV (SP)+, R1
2058 007384 012600          MOV (SP)+, R0
2059 007386 012767 007332 170406  MOV #.PFAIL, 24 ;SET UP FOR POWER FAILURE
2060 007388 012767 000340 170352  MOV #340, PS
2061 007390 012706 001100  MOV #STACK, SP
2062 007392 005067 001620  CLR TEMP
2063 007394 005267 001614  IS: INC TEMP
2064 007396 001375          BNE IS
2065 007398 104410          CONVRT
2066 007400 007466          PFTAB
2067 007402 104402          TYPE
2068 007404 010341          MPFAIL
2069 007406 001367 171546  CLR ERPF LG
2070 007408 001367 171422  CLR LSTERR
2071 007410 001377 171426  JMP #RTN
2072 007412 000001          PFTAB: 1
2073 007414 006          .BYTE 6,2
2074 007416 001114          RTRN
2075
2076          ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 16 TO ALLOW CHANGING
2077          ;OF LOC. 176.
2078          ;LOCATIONS USED:
2079          RDSW: .WORD 0
2080
2081 007474 000000
2082
2083 007476 005737 000042          .CKSWR: TST #42
2084 007502 001042          BNE OUT
2085 007504 022767 000176 171366  CMP #S_REG, SWR ;SOFTWARE SWITCH REGISTER PRESENT
2086 007512 001036          BNE OUT ;NO, GET OUT
  
```

2087	007514	105777	171364			TSTB	ATKCSR		:YES WAIT FOR
2088	007520	100033				BPL	OUT		:READY GET CHARACTER
2089	007522	017767	171360	176352		MOV	ATKDBR .MSG		:AND STRIP OFF
2090	007530	042767	177600	176344		BIC	0177600 .MSG		:THE GARBAGE
2091	007536	122767	000007	176336		CMPB	07 .MSG		:IS IT A <IG>
2092	007544	001021				BNE	OUT		
2093	007546	104402	007624			TYPE, SCNTG			
2094	007552	005137	007474		.CNTLU:	COM	ATKDSH		
2095	007554	104402	007631			TYPE, SWSR			
2096	007556	104402	007631			CMVAT, SWSR			
2097	007558	104402	007616			INSTR, SWSR			
2098	007560	104402	007641			PARAM			
2099	007572	000000				0			
2100	007574	177777				177777			
2101	007576	000176				SWSR			
2102	007582	000	001			.BYTE	0,1		
2103	007584	104402	010336			TYPE, MCRLF			
2104	007586	005037	007474			OUT:	CLR	ATKDSH	
2105	007590	000002				RTI			
2106	007594	000001				SWSR:	1		
2107	007596	006	002			.BYTE	6,2		
2108	007598	000176				SWSR			
2109	007600	005015	043536	000		SCNTG:	.ASCIZ <15><12>/IG/		
2110	007602	015555	051412	051127		SWSR:	.ASCIZ <15><12>/SWSR= /		
2111	007604	020075	000			SWSR:	.ASCIZ / NEW= /		
2112	007606	040	047040	053505		SWSR:	.ASCIZ / NEW= /		
2113	007608	020075	000			.EVEN			
2114	007610	007652				HTITLE:	.ASCIZ <15><12><12>/DU11 DZDUF-C TAPE F /<15><12>		
2115	007612	007715	043012	030535					
2116	007614	007715	043104	041524					
2117	007616	007715	043103	041524					
2118	007618	007715	043040	006440					
2119	007620	007715							
2120	007622	007715	042526	052103		MVECTOR:	.ASCIZ <15><12>/VECTOR ADDRESS-/		
2121	007624	007715	040440	042104					
2122	007626	007715	051523	010005		MREGAD:	.ASCIZ <15><12>/1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS-/		
2123	007628	007715	051461	041124					
2124	007630	007715	051252	041124					
2125	007632	007715	051105	041440					
2126	007634	007715	051124	046117					
2127	007636	007715	051040	041511					
2128	007638	007715	051022	042101					
2129	007640	007715	051505	026523					
2130	010007	000							
2131	010009	015	040412	042522		MMULT:	.ASCIZ <15><12>/ARE YOU RUNNING MULTIPLE DEVICES ? (Y OR N)-/		
2132	010011	015	052517	051040					
2133	010013	054446	044516	043516					
2134	010015	047126	044516	044524					
2135	010017	046440	046124	044524					
2136	010019	046120	020106	042504					
2137	010021	044526	042503	020123					
2138	010023	046440	054450	047440					
2139	010025	020077	054450	047440					
2140	010027	020077	044516	000000					
2141	010029	020122	040514	052123		MLASTD:	.ASCIZ <15><12>/LAST DEVICE:RECEIVER CONTROL REGISTER ADDRESS-/		
2142	010031	005015	040514	052123					
2143	010033	042040	053106	041511					
2144	010035	042040	042522	042503					

2143	010110	053111	051105	041440
2144	010111	047117	051124	046117
2145	010112	051049	043505	051511
2146	010113	042524	020122	042101
2147	010114	051104	051505	026523
2148	010115	000		
2149	010116	075	042504	044526
2150	010117	042503	020040	000
2151	010118	015	044012	053517
2152	010119	047044	053517	041040
2153	010120	047522	047127	041440
2154	010121	053517	020077	042705
2155	010122	051505	047510	041505
2156	010123	020140	047510	042510
2157	010124	047510	047111	020107
2158	010125	047524	051040	047123
2159	010126	040040	041501	051124
2160	010127	043505	000	
2161	010128	015	047412	052123
2162	010129	047440	020105	040522
2163	010130	043516	035105	042522
2164	010131	054524	042520	046040
2165	010132	051501	020124	042504
2166	010133	044	047503	051040
2167	010134	041500	051123	040440
2168	010135	042104	042522	051523
2169	010136	047055		
2170	010137	047040	000077	
2171	010138	047015	000077	
2172	010139	047040	050077	053517
2173	010140	047040	047040	047040
2174	010141	047040	047040	047040
2175	010142	047040	047040	047040
2176	010143	047040	047040	047040
2177	010144	047040	047040	047040
2178	010145	047040	047040	047040
2179	010146	047040	047040	047040
2180	010147	047040	047040	047040
2181	010148	047040	047040	047040
2182	010149	047040	047040	047040
2183	010150	047040	047040	047040
2184	010151	047040	047040	047040
2185	010152	047040	047040	047040
2186	010153	047040	047040	047040
2187	010154	047040	047040	047040
2188	010155	047040	047040	047040
2189	010156	047040	047040	047040
2190	010157	047040	047040	047040
2191	010158	047040	047040	047040
2192	010159	047040	047040	047040
2193	010160	047040	047040	047040
2194	010161	047040	047040	047040
2195	010162	047040	047040	047040
2196	010163	047040	047040	047040
2197	010164	047040	047040	047040
2198	010165	047040	047040	047040
2199	010166	047040	047040	047040
2200	010167	047040	047040	047040
2201	010168	047040	047040	047040
2202	010169	047040	047040	047040
2203	010170	047040	047040	047040
2204	010171	047040	047040	047040
2205	010172	047040	047040	047040
2206	010173	047040	047040	047040
2207	010174	047040	047040	047040
2208	010175	047040	047040	047040
2209	010176	047040	047040	047040
2210	010177	047040	047040	047040
2211	010178	047040	047040	047040
2212	010179	047040	047040	047040
2213	010180	047040	047040	047040
2214	010181	047040	047040	047040
2215	010182	047040	047040	047040
2216	010183	047040	047040	047040
2217	010184	047040	047040	047040
2218	010185	047040	047040	047040
2219	010186	047040	047040	047040
2220	010187	047040	047040	047040
2221	010188	047040	047040	047040
2222	010189	047040	047040	047040
2223	010190	047040	047040	047040
2224	010191	047040	047040	047040
2225	010192	047040	047040	047040
2226	010193	047040	047040	047040
2227	010194	047040	047040	047040
2228	010195	047040	047040	047040
2229	010196	047040	047040	047040
2230	010197	047040	047040	047040
2231	010198	047040	047040	047040
2232	010199	047040	047040	047040
2233	010200	047040	047040	047040

DEVICE: .ASCIZ /=DEVICE /

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

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

MM: .ASCIZ / ?/

MCRLF: .ASCIZ <15><12>

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

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

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

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

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

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

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

```

011219 000000 020103
011220 000000 020103
011221 000000 020103
011222 000000 020103
011223 000000 020103
011224 000000 020103
011225 000000 020103
011226 000000 020103
011227 000000 020103
011228 000000 020103
011229 000000 020103
011230 000000 020103
011231 000000 020103
011232 000000 020103
011233 000000 020103
011234 000000 020103
011235 000000 020103
011236 000000 020103
011237 000000 020103
011238 000000 020103
011239 000000 020103
011240 000000 020103
011241 000000 020103
011242 000000 020103
011243 000000 020103
011244 000000 020103
011245 000000 020103
011246 000000 020103
011247 000000 020103
011248 000000 020103
011249 000000 020103
011250 000000 020103
011251 000000 020103
011252 000000 020103
011253 000000 020103
011254 000000 020103
011255 000000 020103
011256 000000 020103
011257 000000 020103
011258 000000 020103
011259 000000 020103
011260 000000 020103
011261 000000 020103
011262 000000 020103
011263 000000 020103
011264 000000 020103
011265 000000 020103
011266 000000 020103
011267 000000 020103
011268 000000 020103
011269 000000 020103
011270 000000 020103
011271 000000 020103
011272 000000 020103
011273 000000 020103
011274 000000 020103
011275 000000 020103
011276 000000 020103
011277 000000 020103
011278 000000 020103
011279 000000 020103
011280 000000 020103
011281 000000 020103
011282 000000 020103
011283 000000 020103
011284 000000 020103
011285 000000 020103
011286 000000 020103
011287 000000 020103
011288 000000 020103
011289 000000 020103
011290 000000 020103
011291 000000 020103
011292 000000 020103
011293 000000 020103
011294 000000 020103
011295 000000 020103
011296 000000 020103
011297 000000 020103
011298 000000 020103
011299 000000 020103
011300 000000 020103

```

HWIRE6: .ASCIZ <15><12>/IS SEC XMIT JUMPER #6 IN? (Y OR N)-/

HWIRE5: .ASCIZ <15><12>/IS SEC REC JUMPER #5 IN? (Y OR N)-/

HWIRE4: .ASCIZ <15><12>/IS OPT CLR ENABLE JUMPER #4 IN? (Y OR N)-/

NEXTJ: .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

```

2295 011314 000040
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310

```

```

MDATA: .BLKB 40
;*****
;UTILITIES
;*****

```

```

;THIS UTILITY CALCULATES PRIORITY LEVEL
DULEV: ASL DUPRT ;SHIFT LEFT
        ASL DUPRT
        ASL DUPRT
        ASL DUPRT
        ASL DUPRT
        MOV DUPRT,LESS1 ;MOVE THIS TO LESS1
        SUB #1,LESS1 ;CREATE LESS1
        BIC #37,LESS1 ;CLEAR TNZVC
        RTS
        PC
DUPRT: LEVEL5
LESS1: LEVEL4 ;LEVEL TO ALLOW INTERRUPTS

```

```

;NEW DU ADDRESSES
DUBADR: MOV DI ,RXCSR ;XXX0
        INC DI
        MOV DI ,HRXCSR ;XXX1
        INC DI
        MOV DI ,RXDBUF ;XXX2
        MOV DI ,PARCSR ;XXX2
        INC DI
        MOV DUBASE,HRXDBUF ;XXX3
        MOV DUBASE,HPARCSR ;XXX3
        INC DUBASE
        MOV DUBASE,TXCSR ;XXX4
        INC DUBASE
        MOV DUBASE,HTXCSR ;XXX5
        INC DUBASE
        MOV DUBASE,TXDBUF ;XXX6
        INC DUBASE
        MOV DUBASE,HTXDBUF ;XXX7
        RTS
        PC
DUBASE: 0

```

```

;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
;INFORMATION CONTAINED IN TEMP1 AND IT IS
;SHIFTED IN BY THE CONTENTS OF SHIFT
RPOKE: BIC @MDATA,@TXCSR
        CLR TEMP2
        ROR TEMP1 ;FORCE CARRY
        ROR TEMP2 ;PICK UP CARRY IN BIT 15
        ASR TEMP2 ;SHIFT INTO BIT 14
        BIC @BIT15,TEMP2 ;CLR BIT 15
        BIS TEMP2,@TXCSR ;POKE MAINT DATA
        BIC @CLK,@TXCSR ;POKE CLK
        BIS @CLK,@TXCSR
        DFC SHIFT
        B .E R-POKE
        RTS
        PC

```

;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR


```

2311 011652 016767 167266 167266 0008: MOV TEMP1,TEMP2 ;SAVE TEMP1
2312 011660 005067 167264 CLR TEMP3
2313 011664 012727 000010 MOV #8.,(PC)+
2314 011670 000000 18: 0
2315 011672 006067 167250 29: ROR TEMP2
2316 011676 005567 167246 ROR TEMP3
2317 011702 005367 177762 ROR TEMP3
2318 011706 001371 18: 0
2319 011710 006067 167234 29: ROR TEMP3
2320 011714 103404 BCS 39
2321 011716 052767 000400 167220 BIS #8IB,TEMP1 ;SET ODD PARITY
2322 011724 000400 BR 49
2323 011726 042767 000400 167210 39: BIC #8IB,TEMP1 ;CLR EVEN PARITY
:TEMP1 NOW HAS ODD PARITY CHARACTER
2324 011734 000207 49: RTS PC
:THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
2325 011736 016767 167202 167202 EVEN8: MOV TEMP1,TEMP2 ;SAVE TEMP1
2326 011744 005067 167200 CLR TEMP3
2327 011750 012727 000010 MOV #8.,(PC)+
2328 011754 000000 18: 0
2329 011756 006067 167164 29: ROR TEMP2
2330 011762 005567 167162 ROR TEMP3
2331 011766 005367 177762 ROR TEMP3
2332 011772 001371 18: 0
2333 011774 006067 167150 29: ROR TEMP3
2334 012000 103004 BCC 39
2335 012002 052767 000400 167134 BIS #8IB,TEMP1 ;SET EVEN PARITY
2336 012010 000403 BR 49
2337 012012 042767 000400 167124 39: BIC #8IB,TEMP1 ;CLR ODD PARITY
:TEMP1 NOW HAS EVEN PARITY CHARACTER
2338 012020 000207 49: RTS PC
2339 012022 062716 000002 TRPREG: ADC #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
;IN MAIN PART OF THE PROGRAM
2340 012026 000002 RTI
:ERROR HLT TABLE
2341 012030 012114 .ERRTAB: EM0 ;HLT 0 BIT ERROR (GENERAL)
2342 012032 000000 0
2343 012034 000000 0
2344 012036 012130 EM1 ;HLT 1 REGISTER ERROR
2345 012038 012101 DH1
2346 012040 012102 DT1
2347 012042 012172 EM2 ;HLT 2 RECEIVER ERROR
2348 012044 012172 DH1
2349 012046 012161 DT1
2350 012048 012102 EM3 ;HLT 3 TRANSMITTER ERROR
2351 012050 012234 DH1
2352 012052 012301 DT1
2353 012054 012301 DH1
2354 012056 012322 DT1
:DEFAULT DU ADDRESSES
2355 012060 160040 RXCSR: 160040
2356 012062 160041 HRXCSR: 160041
2357 012064 160042 RXDUF: 160042
2358 012066 160043 TXDUF: 160043
2359 012068 160042 PAR: 160042
2360 012070 160043 HFCSR: 160043
2361 012072 160043 TXCSR: 160043
2362 012074 160044

```

```

2367 012076 160045
2368 012100 160046
2369 012102 160047
2370
2371 012104 000770
2372 012106 000772
2373 012110 000774
2374 012112 000776
2375
2376 012114 036440 042440 051122
2377 012122 051117 050040 000103
2378 012130 036440 051040 043505
2379 012136 051511 042524 020122
2380 012144 051105 042522 020122
2381 012152 041520 050015 051001
2382 012160 043505 051511 042524
2383 012166 020122 000040
2384 012172 036440 051040 041505
2385 012200 044505 042526 020122
2386 012206 051105 047522 020122
2387 012214 041520 050015 051001
2388 012222 043505 051511 042524
2389 012230 020122 000040
2390 012234 036440 051040 040522
2391 012242 051516 044515 052124
2392 012250 051105 042440 051122
2393 012256 051117 051040 008503
2394 012264 000412 042522 044507
2395 012272 052123 051105 020040
2396 012300 000
2397
2398 012301 105 050130 041505
2399 012306 042524 020104 040440
2400 012314 052103 040525 000114
2401
2402
2403 012322 000003
2404 012324 006 004
2405 012326 001164
2406 012330 006 004
2407 012332 001156
2408 012334 006 002
2409 012336 001160
2410 000001

```

```

HTXCSR: 160045
TXDBUF: 160046
HTXDBUF: 160047
:DEFAULT DU VECTORS
DURIV: 770 :REC INTR VECTOR
DURIS: 772 :REC INTR STATUS
DUTIV: 774 :XMIT INTR VECTOR
DUTIS: 776 :XMIT INTR STATUS
:ERROR MESSAGES
EMO: .ASCIZ / = ERROR PC/
EM1: .ASCIZ / = REGISTER ERROR PC/<15><12><1>/REGISTER /
EM2: .ASCIZ / = RECEIVER ERROR PC/<15><12><1>/REGISTER /
EM3: .ASCIZ / = TRANSMITTER ERROR PC/<15><12><1>/REGISTER /
:DATA HEADERS FOR ERROR MESSAGES
DH1: .ASCIZ /EXPECTED ACTUAL/
.EVEN
DT1: :DATA TABLES FOR ERROR MESSAGES
:BYTE 6,4
SAVR3 :REGISTER
:BYTE 6,4
SAVR0 :EXPECTED DATA
:BYTE 6,2
SAVR1 :ACTUAL DATA
.END

```


PS	=	177776	601#	828*	987*	1703*	2060*									
PUSHRO	=	010746	603#	2018												
PUSHIS	=	005746	606#													
PUSH2S	=	024646	610#													
ROSM	=	007474	1800	1837	2080#	2094*	2104*									
REOACI	=	004003	647#	1056	1090	1146	1170	1236	1260	1326	1350					
REPLAY	=	005426	1667	1673	1675#											
RESREG	=	007234	2012	2015#												
RESTAR	=	007370	2046	2052#												
RESTRT	=	005572	1669	1697	1703#											
RESOS	=	104407	806#	2015												
RING	=	040000	644#													
RIMEN	=	000100	652#													
ROTAD	=	001223	775#	903*	915*	917	919*	924	927*	1662*	1665	1669*				
RPOKE	=	011564	2297#	2307												
RTRN	=	001114	729#	838#	1017	1021*	1023	1705*	1722*	1723	1742*	1743	2028*	2030	2071	
RTS	=	000004	2074													
RMA	=	####	656#													
RMB	=	####	1	33	45	108	546									
RMC	=	####	1	33	45	108	546									
RMD	=	####	1	33	45	108	546									
RME	=	####	1	33	45	108	546									
RMF	=	####	1	33	45	108	546									
RUNIT	=	005332	1653	1659#	1666											
RXCSR	=	012060	1047*	1056	1080	1107	1137*	1146	1170	1197	1227*	1236	1260	1287	1317*	
			1326	1350	1377	1407*	1423	1457*	1473	1509*	1549	1586*	1626	1689	2274*	
			200#													
RXDBUF	=	012064	1042	1086	1132	1176	1222	1266	1312	1356	1404	1427	1454	1477	1506	
			1553	1583	1630	2278#	2362#									
RXDONE	=	000700	651#													
RXC	=	105000	650#													
SAVFC	=	001174	756#	1859*	2034											
SAVRO	=	001156	749#	1878*	1893	2407										
SAVR1	=	001160	750#	1877*	1E 4	2409										
SAVR2	=	001162	751#	1876*	1E 5											
SAVR3	=	001164	752#	1875*	1E 6	2405										
SAVR4	=	001166	753#	1874*	1887											
SAVR5	=	001170	754#	1873*	1879											
SAVSP	=	001172	755#	2045*	2052											
SAVDS	=	104406	804#	1936												
SCOPE	=	104400	792#	1114	1204	1294	1384	1434	1484	1561	1638					
SCOPI	=	104401	794#	1076	1186	1276	1366									
SEND	=	000020	687#	1045	1135	1225	1315	1400	1450	1502	1579					
SEDEC	=	001300	760#	975												
SETFLG	=	104412	812#	897	970	974	978	982	1001							
SEVEN	=	004000	673#	1224												
SEXMIT	=	001177	759#	971												
SHIFT	=	001140	742#	1069*	1073*	1159*	1163*	1249*	1253*	1339*	1343*	1411*	1416*	1417	1421	
			1461*	1466*	1467	1471	1525*	1542*	1543	1547	1602*	1619*	1620	1624	2306*	
			672#	1134												
SIX	=	002000	1905#	1924	1927*	1943#										
SPACNT	=	000757	648#													
STD	=	000010	602#	829	988	2029	2061									
STFLG	=	001223	655#													
			781#	832#												

UUUUUU

C05

DZDUF-C MACY11 27(1005) 01-OCT-76 09:48 PAGE 58

DZDUF.C.P11 05-AUG-76 00:00

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0054

STPSYN# 000400	650#																
SVOS 006452	1873#																
SWR 001100	720#	844#	849	853#	859	862	997	1010	1727	1734	1751	1980	2016				
	2024	2026	2005														
SVREG 000176	711#	813	859	2085	2101	2108											
SVREGC 007616	2096	2106#															
SVMOO # 000001	582#	862															
SVMOO1 # 000002	581#	1010															
SVMOO2 # 000004	580#																
SVMOO3 # 000010	579#																
SVMOO4 # 000020	578#																
SVMOO5 # 000040	577#																
SVMOO6 # 000100	576#																
SVMOO7 # 000400	575#	2024															
SVMOO8 # 001000	574#	1751															
SVMOO9 # 002000	573#	2026															
SVMOO10 # 004000	572#	1734															
SVMOO11 # 010000	571#																
SVMOO12 # 020000	570#	1980															
SVMOO13 # 040000	569#	1727															
SVMOO14 # 100000	568#																
SYNCHO 001176	758#	959#	963#	1078	1168	1258	1348										
SYNEXT# 020000	669#	1395	1403	1445	1453	1497	1505	1574	1582								
SYNINT# 030000	669#	1044	1134	1224	1314												
SYNSCH# 000020	654#	1047	1107	1137	1197	1227	1287	1317	1377	1407	1457	1509	1586				
SYSTST# 014000	694#																
TEMP 011294	1901	1909	1934#	2062#	2063#	2254#											
TEMP1 001144	744#	2299#	2311	2321#	2223#	2328	2338#	2340#									
TEMP2 001146	745#	2293#	2300#	2301#	2302#	2303	2311#	2315#	2328#	2332#							
TEMP3 001158	746#	1901#	1934	2312#	2316#	2319#	2329#	2333#	2336#								
TEMP4 001152	747#																
TEMP5 001154	748#																
TKCSA 001104	722#	1730	1780	2097													
TKCSB 001106	723#	1732	1782	1790	2089												
TKCST # 005004	1016	1640#															
TKCSA 001110	724#	1763	1791														
TKCSB 001112	725#	1765#	1790#														
TKCSK 001114	1971#																
TKCSL 012026	2343#																
TSTNO 001126	734#	837#	1039#	1128#	1218#	1308#	1392#	1442#	1462#	1569#							
TST1 002226	1015	1021	1036#	1705	1706												
TST2 002228	1039	1129#															
TST3 002230	1129	1219#															
TST4 004100	1219	13															
TST5 004102	1309	13															
TST6 004312	1393	1442#															
TST7 004522	1443	1492#															
TST8 005004	1493	1569#	1640														
TST9 # #####	1570																
TTST 005676	994#	997#	1005#	1006#	1008#	1009#	1728#										
TXCSR 012074	1041#	1043#	1050#	1051#	1054#	1055#	1064	1071#	1072#	1100#	1101	1106#	1131#				
	1135#	1143#	1141#	1144#	1145#	1154	1161#	1162#	1190#	1191	1196#	1221#	1225#				
	1230#	1231#	1234#	1235#	1244	1251#	1273#	1273#	1281	1285#	1311#	1315#	1320#				
	1321#	1324#	1335#	1334	1341#	1342#	1340#	1371	1376#	1374#	1396#	1400#	1404#				
	1410#	1414#	1415#	1444#	1446#	1450#	1459#	1460#	1464#	1465#	1496#	1498#	1502#				
	1511#	1518#	1528#	1535#	1573#	1575#	1579#	1588#	1595#	1605#	1612#	2284#	2297#				

SWORDF 552#
SWORDO 552#
SWOROP 552#

. ABS. 012340 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDUFC, DZDUFC/CRF/SOL=HELLO.P11, PARA.P11, KEET.P11, DZDUFC.P11
RUN-TIME: 16 23 2 SECONDS
RUN-TIME RATIO: 117/42=2.7
CORE USED: 18K (35 PAGES)

