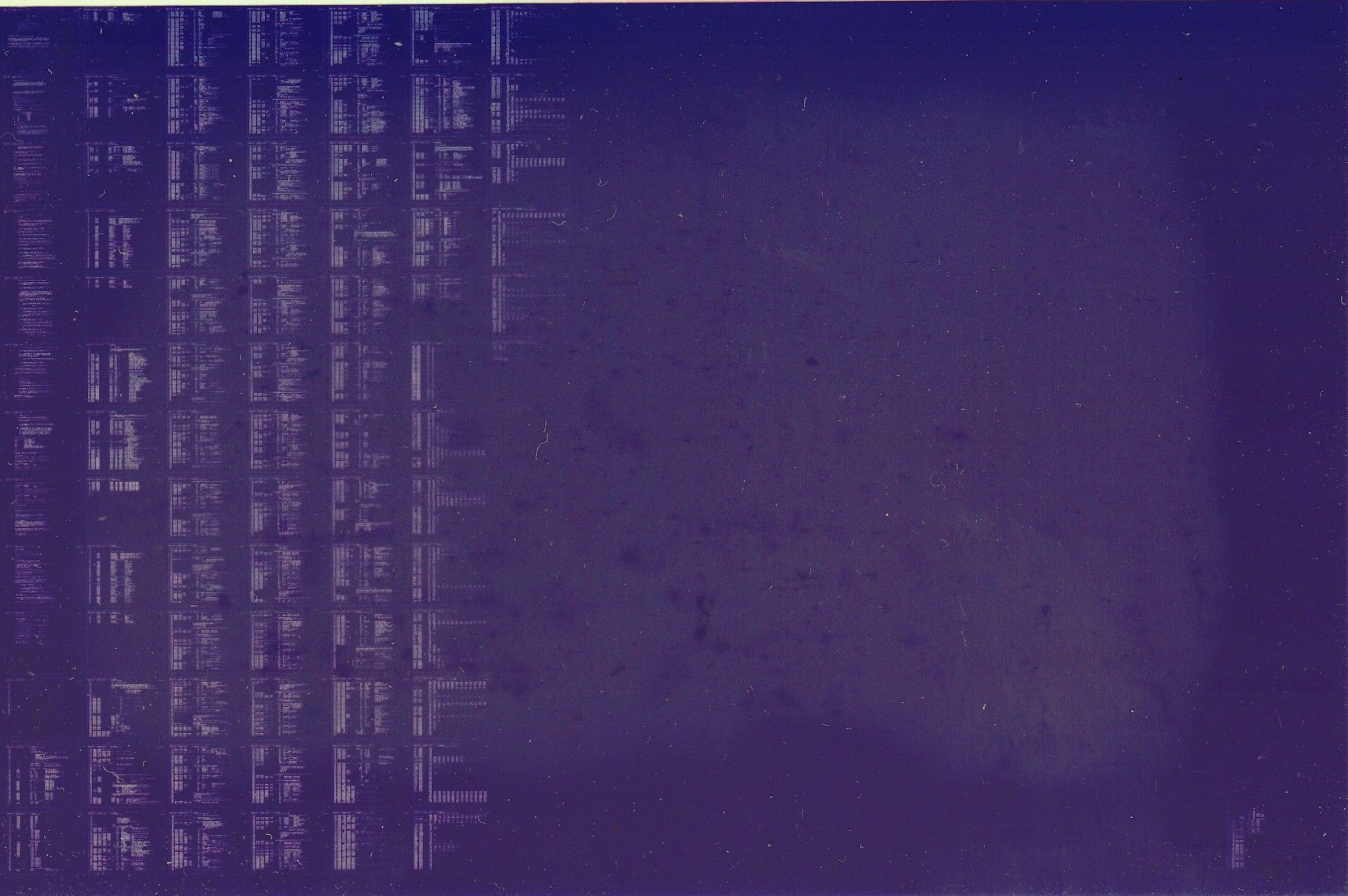


DUV-11

OFFLINE COMBINED TESTS
MD-11-DZDUV-B

EP-DZDUV-B-DL-B
COPYRIGHT © 1977
FICHE 1 OF 1

DEC 1977
digital
MADE IN USA



REM *

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

PRODUCT CODE MAINDEC-11-DZDUV-B-D

PRODUCT NAME DUV11 OFFLINE COMBINED TESTS

RELEASE DATE NOV 1977

MAINTAINER DIAGNOSTICS

*
REM *

COPYRIGHT (C) 1977
DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS

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

*

REM *

GENERAL DESCRIPTION

THIS DIAGNOSTIC CAN CHAIN 16 DUV11'S. THIS MEANS THAT 16 DEVICES CAN BE SEQUENTIALLY EXERCISED. THE DIAGNOSTIC MAKES ONE PASS BEFORE PROCEEDING TO THE NEXT DEVICE, AND CONTINUES EXERCISING ALL DEVICES IN THIS FASHION UNTIL HALTED

1 THE DUV11 OFFLINE COMB NED TESTS VERIFY THAT THE TRANSMITTER AND RECEIVER CAN TALK THRU THE EXTERNAL MODEM CABLE PROVIDING THAT THE H315 CONNECTOR IS ON

* REM *

2 REQUIREMENTS

PDF-11/03 COMPUTEP (LSI)

DUV11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

* REM *

2 2 STORAGE
THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

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 ALL SWITCHES RESIDE INTERNAL TO THE CPU AT ADDRESS 176. THESE MAY BE SET VIA THE CONSOLE TTY BY DIRECTLY MODIFYING LOC 176

NOTE RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER CALLED "\$USWR" IN ORDER TO BE FLEXIBLE ON THE AVAILABILITY OF THE H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11. BIT 0 IN \$USWR REFLECTS THIS STATUS, A 0 = CONNECTOR PRESENT, A 1 = CONNECTOR NOT AVAILBLE

THE USER CHANGES THE CONTENTS OF THIS LOCATION
WHEN BUILDING THE E TABLE, BY ANSWERING THE
PROMPT "SWITCH 2"

- 4 1 1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)
ALL CONSOLE SWITCHES DOWN
- 4 1 2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1
- 4 1 3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)
SW01=1
- 4 1 4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART
(ONLY IN SINGLE DEVICE TESTS)

SW14=1

NOTE1 IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED

NOTE2 WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1

- 4 2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200

THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200

THE STARTING ADDRESS TO LOCK ON TEST IS 000200

- 4 3 PROGRAM AND/OR OPERATOR ACTION

- 4 3 1 INITIAL PROGRAM START

4 3 1 1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

4 3 1 2 SET SWITCH REGISTER (LOC 176) TO ZERO

4 3 1 3 TYPE 200G

4 3 1 4 PROGRAM WILL START

*

REM *

4 3 1 5 THE PROGRAM WILL TYPE "DUV11 DZDUV-B TAPE F" (ONCE ONLY)

*

REM *

4 3 1 6 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 SET SWITCH REGISTER (LOC 176) TO A 000001

4 3 3 2 TYPE 200G

4 3 3 3 PROGRAM WILL START

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 DUV11 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 DUV11 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 DUV11 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 DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL REPEAT THE MESSAGE OF 4 3 3.11 1

IF A DEVICE ADDRESS LOWER THAN 1ST DEVICE ADDRESS IS TYPED SCHOOLS OUT . THERE IS NO PROTECTION FOR THIS THE PROGRAM WILL DEFAULT TO TWO DEVICES ACTIVE (UPWARDS FROM 1ST DEVICE ADDRESS) THE SAME APPLIES TO IDENTICAL ADDRESSES TYPED FOR FIRST AND LAST DEVICE OBSERVE LOCATION @ ACTREG SEE SECTION 7 2

4 3 3 12 THE PROGRAM WILL TYPE "# OF SYNC CHARS SELECTED (1 OR 2)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD REFER TO MANUAL FOR PROPER SWITCH SETTINGS OF SWITCH E55-4

4 3 3 13 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 12

4 3 3 14 THE PROGRAM WILL TYPE " IS SEC XMIT SWITCH E55-2 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 15 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 14

4 3 3 16 THE PROGRAM WILL TYPE "IS SEC REC SWITCH E55-3 ON? (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 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 OPT CLR ENABLE SWITCH E55-1 ON? (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 "ARE YOU RUNNING IN MAINT
MODE EXTERNAL ? AND DO YOU HAVE THE EXTERNAL MODEM
BYPASS JUMPER CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN
INPUT FROM THE TELETYPE KEYBOARD

4 3 3 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 "R" TO INDICATE THAT IT
HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

4 3 4 PROGRAM RESTART WITH SW01=1
NOTE THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
...IT WILL NOT WORK IF MULTIPLE DEVICES ARE SELECTED

IF MULTIPLE DEVICES WERE PREVIOUSLY SELECTED, LOAD 000200.
AND SELECT SW00=1 AND ANSWER "NO" TO THE MULTIPLE DEVICE QUESTION
SEE 4 3 3

4 3 4 1 SET SW01=1 IN SWITCH REG (LOC 176)

4 3 4 2 TYPE 200G

4 3 4 3 PROGRAM WILL START

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 SW14 =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 SET SW14=1 IN SWITCH REG (LOC 176)

4 3 5 2 TYPE 200G

4 3 5 3 PROGRAM WILL START

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 OPEATING PROCEDURE

5 1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC 176)

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
SW09 =1 LOOP ON ERROR
SW01 =1 RESTART PROGRAM AT SELECTED TEST
SW00 =1 RESELECT VECTOR AND CONTROL REGISTER ADRESSES
&PARAMETERS AFTER A PROGRAM RESTART

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

6 ERRORS

6 1 ERROR HALTS (UNDER LSI ALL HALT ERRORS RETURN CONTROL TO O D T)
THERE ARE FOUR DISTINCT ERROR TYPEOUTS

6 1 1 PC+2 = ERROR PC
WHERE PC +2 IS THE ADDRESS OF THE CALL TO THE ERROR HANDLER +2

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

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

6 1 2 PC +2 = REGISTER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ

WHERE 16XXXX IS THE ADDRESS OF THE FAILING DEVICE REGISTER

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6 1 3 PC +2 = RECEIVER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED DATA CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL DATA CONTENTS OF THAT REGISTER

6 1 4 PC +2 = TRANSMITTER ERROR PC
REGISTER EXPECTED ACTUAL
16XXXX YYYYYY ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6 1 5 ERROR DESCRIPTIONS
SEE LISTINGS FOR DETAILS OF ERRORS

6 2 ERROR RECOVERY

6 2 1 SW15 =0
IF THE PROGRAM IS RUN WITH SW15 =0 ,NO OPERATOR ACTION IS
REQUIRED TO CONTINUE TESTING

6 2 2 SW15 =1
IF THE PROGRAM IS RUN WITH SW15 =1 ,TO CONTINUE TESTING
AFTER THE PROGRAM HAS HALTED ,PRESS THE PROCESSOR
CONSOLE "CONTINUE SWITCH"

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

6 2 3 ILLEGAL INTERRUPTS
IF AN INTERRUPT OCCURS TO A VECTOR ADDRESS NOT SELECTED
DURING PROGRAM INITIALIZATION, THE PROGRAM WILL HALT IN
THE TRAPCATCHER THE ADDRESS AT WHICH THE PROGRAM
HALTS IS 2 GREATER THAN THE ADDRESS TO WHICH THE INTERRUPT
OCCURED THE PROGRAM MUST BE RESTARTED AT 000200 TO
RECOVER FROM THIS ERROR

6 2 4 ADDITIONAL TROUBLESHOOTING AIDS ERRCNT & PASCNT
CHECK THESE TWO TAG LOCATIONS FOR TOTAL # OF ERRORS AND PASSES RESPECTIVELY
LOADING 000200 AND RESTARTING WILL CLEAR THESE LOCATIONS

6 3 END OF PASS ROUTINE
THIS TYPEOUT IS MENTIONED HERE FOR CONVENIENCE
IT IS IN THE FORM

END OF PASS TAPE Y
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TIMEOUT - TURN TELETYPE OFF

7 RESTRICTIONS

7 1 MULTIPLE DEVICES

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

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

7 2 DISQUALIFYING DEVICES WHEN RUNNING MULTIPLE DEVICES

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

7 2 1 IF DEVICE 0 IS TO BE DISQUALIFIED, SIMPLY RESTART
PROGRAM WITH SW00 = 1 AND OMIT THE FIRST DEVICE

7 2 2 IF HOWEVER, DEVICES 1 THRU 15 OR ANY COMBINATION THEREOF
ARE TO BE DISQUALIFIED LOAD THE LOCATION OF ACTREG
OBSERVE THE ACTIVE BITS (ACTIVE = 1 NONACTIVE = 0)
AND DEPOSIT 0 WHERE THOSE DEVICES ARE TO BE DISQUALIFIED

7 2 2 1 TO RESTART TYPE 200G
THE PROGRAM WILL CONTINUE WITH THE DEVICE IT WAS IN BEFORE HALTING

7 2 2 2 OR SET SW00=1 IN SWITCH REG (LOC 176) AND TYPE 200G
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 TIMEOUT AN ERROR MESSAGE TYPE 200G

7 3 CABLE DELAYS
NOTE EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH M315 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 11'03 MACHINE

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 RECV CONTROL REGISTER ADDRESS- RXCSR 160010
 VECTOR ADDRESS- DURIV 770
 ARE YOU RUNNING MULTIPLE DEVICES?- NO MULTD 0
 LAST DEVICE RECEIVER CONTROL REGISTER ADDRESS- LASTADD 0
 # OF SYNC CHARS SELECTED - 2 SYNCNO 377
 IS SEC XMIT SWITCH ESS-2 ON?- YES SEXMIT 377
 IS SEC REC SWITCH ESS-3 ON?- YES SEREC 377
 IS OPT CLR ENABLE SWITCH ESS-1 ON?- YES OPTCLR 377
 DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
 CONNECTOR ON (H315)- YES JMPBY 377

9 PROGRAM DESCRIPTION

9 1 THIS PROGRAM PERFORMS THE OFFLINE COMBINED (TRANSMITTER & RECEIVER)
 CABLE TESTING OF THE DEVICE
 SEE LISTING FOR DETAILS

REM *
REM *

10 FLOW CHARTS RECEIVER FLOW TRANSMITTER FLOW, TRANSMITTER & RECEIVER FLOW

11 LISTINGS

524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557

000001

STN=1

```

558 .ENABLE ABS
559
560 .DUV11 DZDUV-B TAPE F
561 .COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
562
563 .STARTING PROCEDURE
564 .TYPE 200G
565 .PROGRAM WILL TYPE "DUV11 DZDUV-B TAPE F "
566 .PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
567 .AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE F"
568 .AND THEN RESUME TESTING
569
570 SBTTL BASIC DEFINITIONS
571
572 .*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
573      G01100
574      STACK= 1100
575      EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
576      EQUIV IOT,SCOPE     ;;BASIC DEFINITION OF SCOPE CALL
577
578 .*MISCELLANEOUS DEFINITIONS
579      HT= 11              ;;CODE FOR HORIZONTAL TAB
580      LF= 12              ;;CODE FOR LINE FEED
581      CR= 15              ;;CODE FOR CARRIAGE RETURN
582      CRLF= 200           ;;CODE FOR CARRIAGE RETURN-LINE FEED
583      PS= 177776         ;;PROCESSOR STATUS WORD
584      .EQUIV PS,PSW
585      STKLMT= 177774     ;;STACK LIMIT REGISTER
586      PIRQ= 177772      ;;PROGRAM INTERRUPT REQUEST REGISTER
587      DSWR= 177570     ;;HARDWARE SWITCH REGISTER
588      DDISP= 177570    ;;HARDWARE DISPLAY REGISTER
589
590 .*GENERAL PURPOSE REGISTER DEFINITIONS
591      R0= %0             ;;GENERAL REGISTER
592      R1= %1             ;;GENERAL REGISTER
593      R2= %2             ;;GENERAL REGISTER
594      R3= %3             ;;GENERAL REGISTER
595      R4= %4             ;;GENERAL REGISTER
596      R5= %5             ;;GENERAL REGISTER
597      R6= %6             ;;GENERAL REGISTER
598      R7= %7             ;;GENERAL REGISTER
599      SP= %6             ;;STACK POINTER
600      PC= %7             ;;PROGRAM COUNTER
601
602 .*PRIORITY LEVEL DEFINITIONS
603      PR0= 0             ;;PRIORITY LEVEL 0
604      PR1= 40            ;;PRIORITY LEVEL 1
605      PR2= 100           ;;PRIORITY LEVEL 2
606      PR3= 140           ;;PRIORITY LEVEL 3
607      PR4= 200           ;;PRIORITY LEVEL 4
608      PR5= 240           ;;PRIORITY LEVEL 5
609      PR6= 300           ;;PRIORITY LEVEL 6
610      PR7= 340           ;;PRIORITY LEVEL 7
611
612 .*"SWITCH REGISTER" SWITCH DEFINITIONS
613      SW15= 100000
614      SW14= 40000

```

614	020000	SW13=	20000
615	010000	SW12=	10000
616	004000	SW11=	4000
617	002000	SW10=	2000
618	001000	SW09=	1000
619	000400	SW08=	400
620	000200	SW07=	200
621	000100	SW06=	100
622	000040	SW05=	40
623	000020	SW04=	20
624	000010	SW03=	10
625	000004	SW02=	4
626	000002	SW01=	2
627	000001	SW00=	1
628		EQUIV	SW09, SW9
629		EQUIV	SW08, SW8
630		EQUIV	SW07, SW7
631		EQUIV	SW06, SW6
632		EQUIV	SW05, SW5
633		EQUIV	SW04, SW4
634		EQUIV	SW03, SW3
635		EQUIV	SW02, SW2
636		EQUIV	SW01, SW1
637		EQUIV	SW00, SW0

*DATA BIT DEFINITIONS (BIT00 TO BIT15)

640	100000	BIT15=	100000
641	040000	BIT14=	40000
642	020000	BIT13=	20000
643	010000	BIT12=	10000
644	004000	BIT11=	4000
645	002000	BIT10=	2000
646	001000	BIT09=	1000
647	000400	BIT08=	400
648	000200	BIT07=	200
649	000100	BIT06=	100
650	000040	BIT05=	40
651	000020	BIT04=	20
652	000010	BIT03=	10
653	000004	BIT02=	4
654	000002	BIT01=	2
655	000001	BIT00=	1
656		EQUIV	BIT09, BIT9
657		EQUIV	BIT08, BIT8
658		EQUIV	BIT07, BIT7
659		EQUIV	BIT06, BIT6
660		EQUIV	BIT05, BIT5
661		EQUIV	BIT04, BIT4
662		EQUIV	BIT03, BIT3
663		EQUIV	BIT02, BIT2
664		EQUIV	BIT01, BIT1
665		EQUIV	BIT00, BIT0

*BASIC "CPU" TRAP VECTOR ADDRESSES

667		ERRVEC=	4	TIME OUT AND OTHER ERRORS
668	000004	RESVEC=	10	RESERVED AND ILLEGAL INSTRUCTIONS
669	000010			

670	000014	TBITVEC=14	.. "T" BIT
671	000014	TRTVEC= 14	.. TRACE TRAP
672	000014	BPTVEC= 14	.. BREAKPOINT TRAP (BPT)
673	000020	IOTVEC= 20	.. INPUT/OUTPUT TRAP (IOT) **SCOPE**
674	000024	PWRVEC= 24	.. POWER FAIL
675	000030	EMTVEC= 30	.. EMULATOR TRAP (EMT) **ERPOR**
676	000034	TRAPVEC=34	.. "TRAP" TRAP
677	000060	TKVEC= 60	.. TTY KEYBOARD VECTOR
678	000064	TPVEC= 64	.. TTY PRINTER VECTOR
679	000240	PIRQVEC=240	.. PROGRAM INTERRUPT REQUEST VECTOR

```

680          , STANDARD INTERRUPT VECTORS
681
682
683          =174
684 000174 000000  DISPREG 0
685 000176 000000  SWREG 0
686          =200
687 000200 000167 001746  JMP      START      , GO TO START OF PROGRAM
688
689
690
691          =1100
692 001100 000000  WORD 0
693 001102 177570  LIGHTS 177570
694
695
696          , PROGRAM CONTROL PARAMETERS
697
698
699 001104 000000  RETURN 0
700 001106 000000  NEXT 0          , ADDRESS OF NEXT TEST TO BE EXECUTED
701 001110 000000  LOCK 0         , ADDRESS FOR LOCK ON CURRENT DATA
702 001112 000000  PASCNT 0      , ADDRESS CONTAINING PASS COUNT
703 001114 000000  ERRCNT 0     , ERROR COUNT
704 001116 000000  SAVSP 0      , STACK POINTER STORAGE
705
706          , PROGRAM VARIABLES
707
708 001120 000020  HOLD 20      , TEMPORARY STORAGE=DELAY TIME FOR CABLES
709 001122 000000  SHIFT 0     , TEMPORARY STORAGE= # OF SHIFTS PER CHAR
710 001124 000000  COUNT 0    , TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
711 001126 000000  SAVPC 0     , PROGRAM COUNTER STORAGE
712 001130 000000  HLD0 0
713 001132 000000  HLD1 0
714 001134 000000  HLD2 0
715 001136 000000  HLD3 0
716 001140 000000  HLD4 0
717 001142 000000  HLD5 0
718 001144 000000  HLD6 0
719
  
```



```
720          ,PROGRAM CONVERSATIONAL PARAMETERS
721 001146    377      SYNCNO  BYTE  377      , # OF SYNC CHARS REQ'D FOR SYNC ZATION
722 001147    377      SEXMIT  BYTE  377      , SEC XMIT JUMPER "IN"
723 001150    377      SEREC   BYTE  377      , SEC REC JUMPER "IN"
724 001151    377      OPTCLR  BYTE  377      , OPTIONAL JUMPER CLR "IN"
725 001152    000      MULTD   BYTE  0        , NO MULTIPLE DEVICE FLAG
726 001153    377      JMRBY   BYTE  377      , EXTERNAL MODEM BYPASS JUMPER " N"
727          EVEN
728
729          ,PROGRAM MULTIPLE DEVICE PARAMETERS
730 001154    000000   BASEADD  0        , PROG CONTROLLED 1ST DEVICE ADDR
731 001156    000000   KEEPADD  0        , SAVED 1ST DEVICE ADDR
732 001160    000000   LASTADD  0        , LAST DEVICE RXCSR ADDR
733 001162    000000   BASEIV   0        , PROG CONTROLLED IV
734 001164    000000   KEEPIV   0        , SAVED INTR VECTOR
735 001166    000000   ACTREG   0        , ACTIVE REGISTER ,,,MODIFY THIS
736          ,LOCATION TO DISQUALIFY OR QUALIFY
737          ,DEVICES (1= RUN,,,0= DON'T RUN)
738 001170    000000   ROTADD   0        , ROTATING POINTER FOR ACTREG POINTS
739          , TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES
740
741          ,PROGRAM CONTROL FLAGS
742
743 001172    000      INIFLG   BYTE  0        , PROGRAM INITIALIZATION FLAG
744 001173    000      STFLG    BYTE  0        , TEST START FLAG
745 001174    000      LOKFLG   BYTE  0        , LOCK ON CURRENT TEST FLAG
746          001176    EVEN
747          001400    =1400
748
749
```

```

750
751
752
753      , INSTRUCTION DEFINITIONS
754
755      005746      PUSH1SP=5746      , DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
756      005726      POP1SP=5726      , INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
757      010046      PUSHRO=10046      ; SAVE RO ON STACK =MOV RO, -(SP)
758      012600      POPRO=12600      ; RESTORE RO FROM STACK =MOV (SP)+, RO
759      024646      PUSH2SP=24646      , DECREMENT STACK TWICE =CMP -(SP), -(SP)
760      022626      POP2SP=22626      , INCREMENT STACK TWICE =CMP (SP)+, (SP)+
761      , REGISTER DEFINITIONS
762      , RXCSR BIT DEFINITIONS
763      100000      DSC=BIT15      , DATA SET CHANGE
764      040000      RING=BIT14      , RING
765      020000      CTS=BIT13      , CLR TO SEND
766      010000      CARDET=BIT12      , CARRIER DETECT
767      004000      RECACT=BIT11      , REC ACTIVE
768      002000      SRD=BIT10      , SEC REC DATA
769      001000      DSR=BIT9      , DATA SET RDY
770      000400      STPSYN=BIT8      , STRIP SYNC
771      000200      RXDONE=BIT7      , REC DONE
772      000100      RINTEN=BIT6      , REC INTR ENABLE
773      000040      DSINTE=BIT5      , DSC INTR ENABLE
774      000020      SYN SCH=BIT4      , SYNC SEARCH
775      000010      STD=BIT3      , SEC XMIT DATA
776      000004      RTS=BIT2      , REQ TO SEND
777      000002      DTR=BIT1      , DATA TERM RDY
778      000001      VOID=BIT0
779      , RXDBUF BIT DEFINITIONS
780      100000      RXERR=BIT15      , REC ERROR
781      040000      OVRUN=BIT14      , OVERRUN
782      020000      FRMERR=BIT13      , FRAME ERROR
783      010000      PARER=BIT12      , PARITY ERROR
784      , PARCSR BIT DEFINITIONS
785      001000      PAREN=BIT9      , PARITY ENABLE
786      000400      EUPAR=BIT8      , EVEN PARITY SENSE
787      , PARCSR WRD DEFINITIONS
788      030000      SYNINT=30000      , SYNC EXTERNAL MODE
789      020000      SYNEXT=20000      , SYNC INTERNAL MODE
790      000000      ISYMOD=0      , ISOC MODE
791      000000      FIVE=0      , WORD LENGTH 5 BITS
792      002000      SIX=2000      , WORD LENGTH 6 BITS
793      004000      SEVEN=4000      , WORD LENGTH 7 BITS
794      006000      EIGHT=6000      , WORD LENGTH 8 BITS
795      000000      NOPAR=0      , NO PARITY
796      001000      ODDPAR=1000      , ODD PARITY
797      001400      EVEPAR=1400      , EVEN PARITY
798      , TXCSR BIT DEFINITIONS
799      100000      DNA=BIT15      , DATA NOT AVAILA&LE
800      040000      MTDATA=BIT14      , MAINT DATA
801      020000      CLK=BIT13      , CLK
802      002000      BITW=BIT10      , BIT WINDOW
803      000400      MRESET=BIT8      , MASTER RESET
804      000200      TXDONE=BIT7      , XMIT DONE
805      000100      TXINTE=BIT6      , XMIT INTR ENABLE
  
```

806	000040	DNAINTE=BIT5	,DNA INTR ENAB
807	000020	SEND=BIT4	,SEND
808	000010	HDXEN=BIT3	,HDX/FDX
809	000001	BREAK=BIT0	,BREAK
810			,TXCSR WRD DEFINITIONS
811	000000	USER=0	,USER MODE
812	004000	MINT=4000	,MAINT INT MODE
813	010000	MEXT=10000	,MAINT EXT MODE
814	014000	SYSTST=14000	,SYSTEM TEST MODE

Line	Address	Value	Label	Type	Value	Description
815			SBTTL COMMON TAGS			
817			*****			
818			*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS			
819			*USED IN THE PROGRAM			
821	001400		=			
822	001400		\$CMTAG			.. START OF COMMON TAGS
823	001400	000000		WORD	0	
824	001402	000	\$TSTNM	BYTE	0	.. CONTAINS THE TEST NUMBER
825	001403	000	\$ERFLG	BYTE	0	.. CONTAINS ERROR FLAG
826	001404	000000	\$ICNT	WORD	0	.. CONTAINS SUBTEST ITERATION COUNT
827	001406	000000	\$LPADR	WORD	0	.. CONTAINS SCOPE LOOP ADDRESS
828	001410	000000	\$LPERR	WORD	0	.. CONTAINS SCOPE RETURN FOR ERRORS
829	001412	000000	\$ERTTL	WORD	0	.. CONTAINS TOTAL ERRORS DETECTED
830	001414	000	\$ITEMB	BYTE	0	.. CONTAINS ITEM CONTROL BYTE
831	001415	001	\$ERMAX	BYTE	1	.. CONTAINS MAX. ERRORS PER TEST
832	001416	000000	\$ERRPC	WORD	0	.. CONTAINS PC OF LAST ERROR INSTRUCTION
833	001420	000000	\$GDADR	WORD	0	.. CONTAINS ADDRESS OF 'GOOD' DATA
834	001422	000000	\$BDADR	WORD	0	.. CONTAINS ADDRESS OF 'BAD' DATA
835	001424	000000	\$GDDAT	WORD	0	.. CONTAINS 'GOOD' DATA
836	001426	000000	\$BDDAT	WORD	0	.. CONTAINS 'BAD' DATA
837	001430	000000		WORD	0	.. RESERVED--NOT TO BE USED
838	001432	000000		WORD	0	
839	001434	000	\$AUTOB	BYTE	0	.. AUTOMATIC MODE INDICATOR
840	001435	000	\$NTAG	BYTE	0	.. INTERRUPT MODE INDICATOR
841	001436	000000		WORD	0	
842	001440	177570	\$SWR	WORD	DSWR	.. ADDRESS OF SWITCH REGISTER
843	001442	177570	\$D SPLAY	WORD	DDISP	.. ADDRESS OF DISPLAY REGISTER
844	001444	177560	\$TKS	177560		.. TTY KBD STATUS
845	001446	177562	\$TKB	177562		.. TTY KBD BUFFER
846	001450	177564	\$TPS	177564		.. TTY PRINTER STATUS REG ADDRESS
847	001452	177566	\$TPB	177566		.. TTY PRINTER BUFFER REG ADDRESS
848	001454	000	\$NULL	BYTE	0	.. CONTAINS NULL CHARACTER FOR FILLS
849	001455	002	\$FILLS	BYTE	2	.. CONTAINS # OF FILLER CHARACTERS REQUIRED
850	001456	012	\$FILLC	BYTE	12	.. INSERT FILL CHARS AFTER A "LINE FEED"
851	001457	000	\$TPFLG	BYTE	0	.. "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
852	001460	000000	\$REGAD	WORD	0	.. CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED
853						.. CONTAINS ((\$REGAD)+0)
854	001462	000000	\$REG0	WORD	0	.. CONTAINS ((\$REGAD)+2)
855	001464	000000	\$REG1	WORD	0	.. CONTAINS ((\$REGAD)+4)
856	001466	000000	\$REG2	WORD	0	.. CONTAINS ((\$REGAD)+6)
857	001470	000000	\$REG3	WORD	0	.. CONTAINS ((\$REGAD)+10)
858	001472	000000	\$REG4	WORD	0	.. CONTAINS ((\$REGAD)+12)
859	001474	000000	\$REG5	WORD	0	.. USER DEFINED
860	001476	000000	\$TMP0	WORD	0	.. USER DEFINED
861	001500	000000	\$TMP1	WORD	0	.. USER DEFINED
862	001502	000000	\$TMP2	WORD	0	.. USER DEFINED
863	001504	000000	\$TMP3	WORD	0	.. USER DEFINED
864	001506	000000	\$TMP4	WORD	0	.. USER DEFINED
865	001510	000000	\$TMP5	WORD	0	.. MAX NUMBER OF ITERATIONS
866	001512	000000	\$TIMES	0		.. ESCAPE ON ERROR ADDRESS
867	001514	000000	\$ESCAPE	0		.. CODE FOR BELL
868	001516	177607 000377	\$BELL	ASCII	<207><377><377>	.. QUESTION MARK
869	001522	077	\$QUES	ASCII	/?	.. CARRIAGE RETURN
870	001523	015	\$CRLF	ASCII	<15>	

871	001524	000012	\$LF	ASCIZ	<12>	..L NE FEED
872			..	*****		
873			SBTTL	AP	MAILBOX-ETABLE	
874						
875			..	*****		
876			EVEN			
877	001526		\$MAIL			.. APT MAILBOX
878	001526	000000	\$MSGTY	WORD	AMSGTY	.. MESSAGE TYPE CODE
879	001530	000000	\$FATAL	WORD	AFATAL	.. FATAL ERROR NUMBER
880	001532	000000	\$TESTN	WORD	ATESTN	.. TEST NUMBER
881	001534	000000	\$PASS	WORD	APASS	.. PASS COUNT
882	001536	000000	\$DEVCT	WORD	ADEVCT	.. DEVICE COUNT
883	001540	000000	\$UNIT	WORD	AUNIT	.. I/O UNIT NUMBER
884	001542	000000	\$MSGAD	WORD	AMSGAD	.. MESSAGE ADDRESS
885	001544	000000	\$MSGLG	WORD	AMSGLG	.. MESSAGE LENGTH
886	001546		\$ETABLE			.. APT ENVIRONMENT TABLE
887	001546	000	\$ENV	BYTE	AENV	.. ENVIRONMENT BYTE
888	001547	000	\$ENVM	BYTE	AENVM	.. ENVIRONMENT MODE BITS
889	001550	000000	\$SWREG	WORD	ASWREG	.. APT SWITCH REGISTER
890	001552	000000	\$USWR	WORD	AUSWR	.. USER SWITCHES
891	001554	000000	\$CPUOP	WORD	ACPUOP	.. CPU TYPE, OPTIONS
892			..			BITS 15-11=CPU TYPE
893			..			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
894			..			11/70=06, PDQ=07, Q=10
895			..			BIT 10=REAL TIME CLOCK
896			..			BIT 9=FLOATING POINT PROCESSOR
897			..			BIT 8=MEMORY MANAGEMENT
898	001556	000	\$MAMS1	BYTE	AMAMS1	.. HIGH ADDRESS, M S BYTE
899	001557	000	\$MTYP1	BYTE	AMTYP1	.. MEM TYPE, BLK#1
900			..			MEM TYPE BYTE -- (HIGH BYTE)
901			..			900 NSEC CORE=001
902			..			300 NSEC BIPOLAR=002
903			..			500 NSEC MOS=003
904	001560	000000	\$MADR1	WORD	AMADR1	.. HIGH ADDRESS, BLK#1
905			..			MEM LAST ADDR =3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
906	001562	000	\$MAMS2	BYTE	AMAMS2	.. HIGH ADDRESS, M S BYTE
907	001563	000	\$MTYP2	BYTE	AMTYP2	.. MEM TYPE, BLK#2
908	001564	000000	\$MADR2	WORD	AMADR2	.. MEM LAST ADDRESS, BLK#2
909	001566	000	\$MAMS3	BYTE	AMAMS3	.. HIGH ADDRESS, M S BYTE
910	001567	000	\$MTYP3	BYTE	AMTYP3	.. MEM TYPE, BLK#3
911	001570	000000	\$MADR3	WORD	AMADR3	.. MEM LAST ADDRESS, BLK#3
912	001572	000	\$MAMS4	BYTE	AMAMS4	.. HIGH ADDRESS, M. S. BYTE
913	001573	000	\$MTYP4	BYTE	AMTYP4	.. MEM TYPE, BLK#4
914	001574	000000	\$MADR4	WORD	AMADR4	.. MEM. LAST ADDRESS, BLK#4
915	001576	000000	\$VECT1	WORD	AVECT1	.. INTERRUPT VECTOR#1, BUS PRIORITY#1
916	001600	000000	\$VECT2	WORD	AVECT2	.. INTERRUPT VECTOR#2BUS PRIORITY#2
917	001602	000000	\$BASE	WORD	ABASE	.. BASE ADDRESS OF EQUIPMENT UNDER TEST
918	001604	000000	\$DEVN	WORD	ADEVN	.. DEVICE MAP
919	001606	000000	\$CDW1	WORD	ACDW1	.. CONTROLLER DESCRIPTION WORD#1
920	001610	000000	\$CDW2	WORD	ACDW2	.. CONTROLLER DESCRIPTION WORD#2
921	001612	000000	\$DDW0	WORD	ADDW0	.. DEVICE DESCRIPTOR WORD#0
922	001614	000000	\$DDW1	WORD	ADDW1	.. DEVICE DESCRIPTOR WORD#1
923	001616	000000	\$DDW2	WORD	ADDW2	.. DEVICE DESCRIPTOR WORD#2
924	001620	000000	\$DDW3	WORD	ADDW3	.. DEVICE DESCRIPTOR WORD#3
925	001622	000000	\$DDW4	WORD	ADDW4	.. DEVICE DESCRIPTOR WORD#4
926	001624	000000	\$DDW5	WORD	ADDW5	.. DEVICE DESCRIPTOR WORD#5

927	001626	000000	\$DDW6	WORD	ADDW6	.. DEVICE DESCRIPTOR	WORD#6
928	001630	000000	\$DDW7	WORD	ADDW7	.. DEVICE DESCRIPTOR	WORD#7
929	001632	000000	\$DDW8	WORD	ADDW8	.. DEVICE DESCRIPTOR	WORD#8
930	001634	000000	\$DDW9	WORD	ADDW9	.. DEVICE DESCRIPTOR	WORD#9
931	001636	000000	\$DDW10	WORD	ADDW10	.. DEVICE DESCRIPTOR	WORD#10
932	001640	000000	\$DDW11	WORD	ADDW11	.. DEVICE DESCRIPTOR	WORD#11
933	001642	000000	\$DDW12	WORD	ADDW12	.. DEVICE DESCRIPTOR	WORD#12
934	001644	000000	\$DDW13	WORD	ADDW13	.. DEVICE DESCRIPTOR	WORD#13
935	001646	000000	\$DDW14	WORD	ADDW14	.. DEVICE DESCRIPTOR	WORD#14
936	001650	000000	\$DDW15	WORD	ADDW15	.. DEVICE DESCRIPTOR	WORD#15
937							
938							
939	001652		SETEND				
940							
941							
942							

```

943
944
945
946      , INSTRUCTION DEFINITIONS
947
948      005746      PUSH1SP=5746      , DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
949      005726      POP1SP=5726      ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
950      010046      PUSHRO=10046      , SAVE RO ON STACK =MOV RO, -(SP)
951      012600      POPRO=12600      , RESTORE RO FROM STACK =MOV (SP)+, RO
952      024646      PUSH2SP=24646      , DECREMENT STACK TWICE =CMP -(SP), -(SP)
953      022626      POP2SP=22626      , INCREMENT STACK TWICE =CMP (SP)+, (SP)+
954
955      , REGISTER DEFINITIONS
956      , RXCSR BIT DEFINITIONS
956      100000      DSC=BIT15      , DATA SET CHANGE
957      040000      RING=BIT14      , RING
958      020000      CTS=BIT13      , CLR TO SEND
959      010000      CARDET=BIT12      , CARRIER DETECT
960      004000      RECACT=BIT11      , REC ACTIVE
961      002000      SRD=BIT10      , SEC REC DATA
962      001000      DSR=BIT9      , DATA SET RDY
963      000400      STPSYN=BIT8      , STRIP SYNC
964      000200      RXDONE=BIT7      , REC DONE
965      000100      RINTEN=BIT6      , REC INTR ENABLE
966      000040      DSINTE=BIT5      , DSC INTR ENABLE
967      000020      SYNSCH=BIT4      , SYNC SEARCH
968      000010      STD=BIT3      , SEC XMIT DATA
969      000004      RTS=BIT2      , REQ TO SEND
970      000002      DTR=BIT1      , DATA TERM PDY
971      000001      VOID=BIT0
972
973      , RXDBUF BIT DEFINITIONS
973      100000      RXERR=BIT15      , REC ERROR
974      040000      OVERRUN=BIT14      , OVERRUN
975      020000      FRMERR=BIT13      , FRAME ERROR
976      010000      PARER=BIT12      , PARITY ERROR
977
978      , PARCSR BIT DEFINITIONS
978      001000      PAREN=BIT9      , PARITY ENABLE
979      000400      EUPAR=BIT8      , EVEN PARITY SENSE
980
981      , PARCSR WRD DEFINITIONS
981      030000      SYNINT=30000      , SYNC EXTERNAL MODE
982      020000      SYNEXT=20000      , SYNC INTERNAL MODE
983      000000      ISYMOD=0      , ISOC MODE
984      000000      FIVE=0      , WORD LENGTH 5 BITS
985      002000      SIX=2000      , WORD LENGTH 6 BITS
986      004000      SEVEN=4000      , WORD LENGTH 7 BITS
987      006000      EIGHT=6000      , WORD LENGTH 8 BITS
988      000000      NOPAR=0      , NO PARITY
989      001000      ODDPAR=1000      , ODD PARITY
990      001400      EVEPAR=1400      , EVEN PARITY
991
992      , TXCSR BIT DEFINITIONS
992      100000      DNA=BIT15      , DATA NOT AVAILA&LE
993      040000      MTDATA=BIT14      , MAINT DATA
994      020000      CLK=BIT13      , CLK
995      002000      BITW=BIT10      , BIT WINDOW
996      000400      MRESET=BIT8      , MASTER RESET
997      000200      TXDONE=BIT7      , XMIT DONE
998      000100      TXINTE=BIT6      , XMIT INTR ENABLE
  
```

999	000040	DNAINTE=BIT5	,DNA INTR ENAB
1000	000020	SEND=BIT4	,SEND
1001	000010	HDXEN=BIT3	,HDX/FDX
1002	000001	BREAK=BIT0	,BREAK
1003		;TXCSR WRD DEFINITIONS	
1004	000000	USER=0	,USER MODE
1005	004000	MINT=4000	,MAINT INT MODE
1006	010000	MEXT=10000	,MAINT EXT MODE
1007	014000	SYSTST=14000	,SYSTEM TEST MODE

1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063

001652

001652 001762
001654 002067
001656 002116
001660 002132
001662 002022
001664 002067
001666 002116
001670 002132
001672 002043
001674 002067
001676 002116
001700 002132
001702 001746
001704 000000
001706 002126
001710 002132

160010
160011
160012
160013
160012
160013
160014
160015
160016
160017

000770
000772
000774
000776

020040 051105 047522
020122 041520 000040
020040 047503 050115
051101 051511 047117
042440 051122 051117
047440 020116 042522

SBTTL ERROR POINTER TABLE

*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR
 *THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 *LOCATION \$ITEMB THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT
 *NOTE1 IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC)
 *NOTE2 EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS

* EM ..POINTS TO THE ERROR MESSAGE
 * DH ..POINTS TO THE DATA HEADER
 * DT ..POINTS TO THE DATA
 * DF ..POINTS TO THE DATA FORMAT

SERRTB

.ERROR TABLE

EM1	.ERROR 1	REGISTER ERROR
DH1		
DT1		
DF1		
EM2	.ERROR 2	RECEIVER ERROR
DH1		
DT1		
DF1		
EM3	.ERROR 3	TRANSMITTER ERROR
DH1		
DT1		
DF1		
EM4	.ERROR 4	BIT ERROR (GENERAL)
O		
DT4		
DF1		

.DEFAULT DU ADDRESSES

RXCSR 160010
 HRXCSR 160011
 RXDBUF 160012
 HRXDBUF 160013
 PARCSR 160012
 HPARCSR 160013
 TXCSR 160014
 HTXCSR 160015
 TXDBUF 160016
 HTXDBUF 160017

.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

EM4 ASCIZ / ERROR PC /
 EM1 ASCIZ / COMPARISON ERROR ON REGISTERS /

1064	002012	044507	052123	051105			
1065	002020	000123					
1066	002022	020040	042522	042503	EM2	ASCIZ / RECEIVER ERROR/	
1067	002030	053111	051105	042440			
1068	002036	051122	051117	000			
1069	002043	040	052040	040522	EM3	ASCIZ / TRANSMITTER ERROR/	
1070	002050	051516	044515	052124			
1071	002056	051105	042440	051122			
1072	002064	051117	000				
1073						.DATA HEADERS FOR ERROR MESSAGES	
1074	002067	105	051122	041520	DH1	ASCIZ /ERRPC WANTED ACTUAL/	
1075	002074	020040	040527	052116			
1076	002102	042105	020040	041501			
1077	002110	052524	046101	000			
1078		002116			EVEN		
1079						.DATA TABLES FOR ERROR MESSAGES	
1080	002116	001416	001130	001132	DT1	WORD \$ERRPC,HLDD,HL01,0	
1081	002124	000000					
1082							
1083	002126	001416	000000		DT4	WORD \$ERRPC,0	
1084							
1085	002132	000	000	000	DF1	BYTE 0,0,0,0	
1086	002135	000					
1087					EVEN		
1088					SBTTL	ACT11 HOOKS	
1089							
1090						..*****	
1091						HOOKS REQUIRED BY ACT11	
1092		002136				\$SVPC .	SAVE PC
1093		000046				=46	
1094	000046	012674				\$ENDAD	..1)SET LOC 46 TO ADDRESS OF SENDAD IN SEOP
1095		000052				=52	
1096	000052	000000				WORD 0	..2)SET LOC 52 TO ZERO
1097		002136				=\$SVPC	.. RESTORE PC
1098					SBTTL	APT PARAMETER BLOCK	
1099							
1100						..*****	
1101						SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT	
1102						..*****	
1103		002136				\$X=.	;;SAVE CURRENT LOCATION
1104		000024				=24	;;SET POWER FAIL TO POINT TO START OF PROGRAM
1105	000024	000200				200	;;FOR APT START UP
1106		000044				=44	;;POINT TO APT INDIRECT ADDRESS PNTR
1107	000044	002136				\$APTHDR	;;POINT TO APT HEADER BLOCK
1108		002136				=\$X	;;RESET LOCATION COUNTER
1109						..*****	
1110						SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC	
1111						INTERFACE SPEC	
1112							
1113	002136				\$APTHD.		
1114	002136	000000			\$HIBTS	WORD 0	.. TWO HIGH BITS OF 18 BIT MAILBOX ADDR
1115	002140	001526			\$MBADR	WORD \$MAIL	.. ADDRESS OF APT MAILBOX (BITS 0-15)
1116	002142	000010			\$TSTM	WORD 10	.. RUN TIM OF LONGEST TEST
1117	002144	000010			\$PASTM	WORD 10	.. RUN TIME IN SECS OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1118	002146	000000			\$UNITM	WORD	.. ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1119	002150	000052				WORD	SETEND-\$MAIL/2 ..LENGTH MAILBOX-ETABLE(WORDS)

```

1120
1121
1122          , PROGRAM INITIALIZATION
1123          , LOCK OUT INTERRUPTS
1124          , SET UP PROCESSOR STACK
1125          , SET UP POWER FAIL VECTOR
1126          , CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1127          , TYPE TITLE MESSAGE
1128
1129 002152    START
1130          SBTTL  INITIALIZE THE COMMON TAGS
1131          ,, CLEAR THE COMMON TAGS (%CMTAG) AREA
1132 002152    012706 001400    MOV    #%CMTAG, R6      ,, FIRST LOCATION TO BE CLEARED
1133 002156    005026          CLR    (R6)+           ,, CLEAR MEMORY LOCATION
1134 002160    022706 001440    CMP    #SWR, R6      ,, DONE?
1135 002164    001374          BNE   -6             ,, LOOP BACK IF NO
1136 002166    012706 001100    MOV    ##STACK, SP   ,, SETUP THE STACK POINTER
1137          ,, INITIALIZE A FEW VECTORS
1138 002172    012737 016320 000020  MOV    %%SCOPE, @#IOTVEC  ,, IOT VECTOR FOR SCOPE ROUTINE
1139 002200    012737 000340 000022  MOV    #340, @#IOTVEC+2  ,, LEVEL 7
1140 002206    012737 014210 000030  MOV    #%ERROR, @#EMTVEC  ,, EMT VECTOR FOR ERROR ROUTINE
1141 002214    012737 000340 000032  MOV    #340, @#EMTVEC+2  ,, LEVEL 7
1142 002222    012737 016654 000034  MOV    #STRAP, @#TRAPVEC  ,, TRAP VECTOR FOR TRAP CALLS
1143 002230    012737 000340 000036  MOV    #340, @#TRAPVEC+2, LEVEL 7
1144 002236    012737 015012 000024  MOV    #%PWRDN, @#PWRVEC  ,, POWER FAILURE VECTOR
1145 002244    012737 000340 000026  MOV    #340, @#PWRVEC+2  ,, LEVEL 7
1146 002252    005067 177234          CLR    $TIMES          ,, INITIALIZE NUMBER OF ITERATIONS
1147 002256    005067 177232          CLR    $ESCAPE        ,, CLEAR THE ESCAPE ON ERROR ADDRESS
1148 002262    112767 000001 177125  MOV    #1, $ERMAX       ,, ALLOW ONE ERROR PER TEST
1149 002270    012767 002270 177110  MOV    # , $LPADR       ,, INITIALIZE THE LOOP ADDRESS FOR SCOPE
1150 002276    012767 002276 177104  MOV    # , $LPERR       ,, SETUP THE ERROR LOOP ADDRESS
1151          ,, SIZE FOR A HARDWARE SWITCH REGISTER IF NOT FOUND OR IT IS
1152          ,, EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER
1153 002304    013746 000004          MOV    @#ERRVEC, -(SP)  ,, SAVE ERROR VECTOR
1154 002310    012737 002344 000004  MOV    #64$, @#ERRVEC   ,, SET UP ERROR VECTOR
1155 002316    012767 177570 177114  MOV    #DSWR, SWR       ,, SETUP FOR A HARDWARE SWICH REGISTER
1156 002324    012767 177570 177110  MOV    #DDISP, DISPLAY  ,, AND A HARDWARE DISPLAY REGISTER
1157 002332    022777 177777 177100  CMP    #-1, @SWR       ,, TRY TO REFERENCE HARDWARE SWR
1158 002340    001012          BNE   66$             ,, BRANCH IF NO TIMEOUT TRAP OCCURRED
1159          ,, AND THE HARDWARE SWR IS NOT = -1
1160 002342    000403          BR    65$            ,, BRANCH IF NO TIMEOUT
1161 002344    012716 002352    64$    MOV    #65$, (SP)       ,, SET UP FOR TRAP RETURN
1162 002350    000002          RTI
1163 002352    012767 000176 177060 65$    MOV    #SWREG, SWR     ,, POINT TO SOFTWARE SWR
1164 002360    012767 000174 177054  MOV    #DISPREG, DISPLAY
1165 002366    012637 000004    66$    MOV    (SP)+, @#ERRVEC  ,, RESTORE ERROR VECTOR
1166
1167 002372    005067 177136          CLR    $PASS          ,, CLEAR PASS COUNT
1168 002376    132767 000200 1771+3  BITB  #APTSIZE, $ENVM   ,, TEST USER SIZE UNDER APT
1169 002404    001403          BEQ   67$            ,, YES, USE NON-APT SWITCH
1170 002406    012767 001550 177024  MOV    #SSWREG, SWR    ,, NO, USE APT SWITCH REGISTER
1171 002414          67$
1172 002414    012706 001100          MOV    #STACK, SP     ,, SET STACK
1173 002420    106427 000340          MTPS  #340           ,, LOCK INTERRUPTS
1174 002424    012737 015012 000024  MOV    # PFAIL, @#24   ,, SET UP POWER FAIL VECTOR
1175 002432    105067 176535          CLR   $STFLG         ,, CLEAR START FLAG
  
```

INITIALIZE THE COMMON TAGS

1176	002436	005067	176450			CLR	PASCNT	, CLEAR PASS COUNT
1177	002442	105067	176735			CLRB	\$ERFLG	, CLEAR ERROR FLAG
1178	002446	005067	176740			CLR	\$ERTTL	, CLEAR ERROR COUNT
1179	002452	005067	176740			CLR	\$ERRPC	, CLEAR LAST EPROR POINTER
1180	002456	012767	000001	176716		MOV	#1, \$STSTM	, SET UP FOR TEST 1
1181	002464	012767	002152	176412		MOV	#. START, RETURN	, SET UP FOP POWER FAIL BEFORE
1182								, TESTING STARTS
1183	002472	013746	000006			MOV	@#6, -(SP)	
1184	002476	013746	000004			MOV	@#4, -(SP)	
1185	002502	012737	002516	000004		MOV	#15, @#4	
1186	002510	005777	176724			TST	@SWR	
1187	002514	000407				BR	2\$	
1188	002516	012767	000176	176714	1\$	MOV	#SWREG, SWR	
1189	002524	012767	000174	176710		MOV	#DISPREG, DISPLAY	
1190	002532	022626				CMP	(SP)+, (SP)+	
1191	002534	012637	000004		2\$	MOV	(SP)+, @#4	
1192	002540	012637	000006			MOV	(SP)+, @#6	
1193	002544	022767	000176	176666		CMP	#SWREG, SWR	
1194	002552	001007				BNE	3\$	
1195	002554	005737	000042			TST	@#42	, CHECK FOR CHAIN
1196	002560	001402				BEQ	33\$	
1197	002562	000167	000522			JMP	. BEGIN	
1198	002566	004767	010204		33\$	JSR	PC, CNTLU	
1199	002572	105767	176374		3\$	TSTB	INIFLG	, HAS INITIALIZATION BEEN PERFORMED
1200	002576	001004				BNE	ONCE	
1201	002600	104401	015152			TYPE	, MTITLE	, TYPE TITLE MESSAGE
1202	002604	105167	176362			COMB	INIFLG	, IF NOT SET FLAG AND DO
1203	002610	105767	176732		ONCE	TSTB	\$ENV	, APT CONTROL?
1204	002614	001410				BEQ	11\$, BR IF NO
1205	002616	032767	000001	176726		BIT	#1, \$USWR	, EXTENAL JUMPER ON?
1206	002624	001002				BNE	12\$, NO
1207	002626	105067	176321			CLRB	JMRBY	, CLEAR FLAG
1208	002632	000167	000452		12\$	JMP	. BEGIN	, GO DO IT
1209	002636	032777	000001	176574	11\$	BIT	#SW00, @SWR	, RESELECT VECTOR & CONTROL REG?
1210	002644	001002				BNE	1\$	
1211	002646	000167	000436			JMP	. BEGIN	
1212	002652	012700	000300		1\$	MOV	#300, R0	, RESTORE VECTOR AREA TO TRAPCATCHER
1213	002656	012701	000302			MOV	#302, R1	, START AT LOCATION 300
1214	002662	012702	000004			MOV	#4, R2	
1215	002666	010110			2\$	MOV	R1, (R0)	
1216	002670	005011				CLR	(R1)	
1217	002672	060200				ADD	R2, R0	
1218	002674	060201				ADD	R2, R1	
1219	002676	022701	001000			CMP	#1000, R1	, END AT LOCATION 776
1220	002702	002771				BLT	2\$	
1221	002704	104406				INSTR		, OUTPUT MESSAGE & GET INPUT STRING
1222	002706	015220				MREGAD	; MESSAGE	
1223	002710	104410				PARAM	, CONVERT STRING	
1224	002712	160000				160000	, LOW LIMIT	
1225	002714	167776				167776	; HIGH LIMIT	
1226	002716	017150				DUBASE	, STORE AT THIS LOCATION	
1227	002720	001			BYTE	1	; MASK	
1228	002721	001			BYTE	1	, HOW MANY TIMES + 2	
1229	002722	016767	014222	176226		MOV	DUBASE, KEEPADD	, SAVE
1230	002730	004767	014062			JSR	PC, DUADDR	
1231	002734	016767	176216	176212		MOV	KEEPADD, BASEADD	, RESTORE FOR ROTAT CN

INITIALIZE THE COMMON TAGS

1232	002742	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1233	002744	015205				MVELTO	; MESSAGE
1234	002746	104410				PARAM	; CONVERT STRING
1235	002750	000300				300	; LOW LIMIT
1236	002752	000776				776	; HIGH LIMIT
1237	002754	001736				DURIV	; STORE AT THIS LOCATION
1238	002756	001			BYTE	1	; MASK
1239	002757	004			BYTE	4	; HOW MANY TIMES + 2
1240	002760	016767	176752	176176		MOV	DURIV,KEEPIV ,SAVE
1241	002766	016767	176744	176166		MOV	DURIV,BASEIV ,SET UP FOR ROTATION
1242	002774	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1243	002776	015250				MMULT	; MESSAGE
1244	003000	104414				SETFLG	; SET FLAG BASED UPON INPUT STRING
1245	003002	001152				MULTD	; THIS FLAG
1246	003004	105767	176142			TSTB	MULTD ,ARE THERE MULTIPLE DEVICES
1247							, ON THE SYSTEM ?
1248	003010	100406				BMI	BBB ,YES, ASK NEXT QUESTION
1249	003012	005067	176150			CLR	ACTREG
1250	003016	005067	176146			CLR	ROTADD
1251	003022	000167	000140			JMP	OUTMUL , JUMP AROUND NEXT QUESTION
1252	003026				BBB		
1253	003026	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1254	003030	015277				MLASTD	; MESSAGE
1255	003032	104410				PARAM	; CONVERT STRING
1256	003034	160000				160000	; LOW LIMIT
1257	003036	167776				167776	; HIGH LIMIT
1258	003040	001160				LASTADD	; STORE AT THIS LOCATION
1259	003042	001			BYTE	1	; MASK
1260	003043	001			BYTE	1	; HOW MANY TIMES + 2
1261							, THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1262	003044	012767	000001	176116	1\$	MOV	#1,ROTADD ,SET UP POINTER
1263	003052	005067	176110			CLR	ACTREG ,CLR ACTIVE REGISTER
1264	003056	056767	176106	176102	2\$	BIS	ROTADD,ACTREG ,MAKE THIS DEVICE ACTIVE
1265	003064	000241				CLC	
1266	003066	006167	176076			ROL	ROTADD ,SET UP POINTER
1267	003072	103421				BCS	3\$,ARE YOU OUT OF RANGE ?
1268	003074	062767	000010	176052		ADD	#10,BASEADD ,SET UP BASE ADDRESS
1269	003102	026767	176052	176044		CMP	LASTADD,BASEADD ,IS THIS THE LAST DEVICE ?
1270	003110	101362				BHI	2\$,NO DO IT AGAIN
1271	003112	056767	176052	176046		BIS	ROTADD,ACTREG ,THIS ASSUMES THAT THERE ARE AT
1272							,LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1273							,MULTIPLE DEVICE QUESTION
1274	003120	012767	000001	176042	4\$	MOV	#1,ROTADD ,SET UP FOR LATER USE IN END OF PASS ROUTINE
1275	003126	016767	176024	176020		MOV	KEEPADD,BASEADD ,DITTO
1276	003134	000414				BR	OUTMUL ,CONTINUE QUESTIONS
1277	003136	016767	176014	176010	3\$	MOV	KEEPADD,BASEADD ,RESTORE
1278	003144	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1279	003146	015373				MRANGE	; MESSAGE
1280	003150	104410				PARAM	; CONVERT STRING
1281	003152	160000				160000	; LOW LIMIT
1282	003154	167776				167776	; HIGH LIMIT
1283	003156	001160				LASTADD	; STORE AT THIS LOCATION
1284	003160	001			BYTE	1	; MASK
1285	003161	001			BYTE	1	; HOW MANY TIMES + 2
1286	003162	000167	177656			JMP	1\$,DO IT AGAIN
1287	003166	012767	000340	013616	OUTMUL	MOV	#340,DUPRT

```

1288 003174 004767 013542 JSR PC,DULEV
1289 ,COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1290 ,BUFFER TO THE CHARACTERS "1" AND "2"
1291 ,IF THE CHARACTER IS "1" CLEAR THE FLAG
1292 ,IF THE CHARACTER IS "2" SET THE FLAG
1293 003200 AAA
1294 003200 104406 NSTR ,OUTPUT MESSAGE & GET INPUT STRING
1295 003202 015611 MSYNC ;MESSAGE
1296 003204 122767 000061 012740 3$ CMPB #'1,INBUF ,IS IT "1" ?
1297 003212 001003 BNE 1$
1298 003214 105067 175726 CLRB SYNCNO ,000
1299 003220 000412 BR 4$
1300 003222 122767 000062 012722 1$ CMPB #'2,INBUF ,IS IT "2" ?
1301 003230 001004 BNE 2$
1302 003232 112767 177777 175706 MOVB #-1,SYNCNO ,377
1303 003240 000402 BR 4$
1304 003242 104407 2$ INSTR ,RETRY
1305 003244 000757 BR 3$
1306 003246 000240 4$ NOP
1307 003250 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1308 003252 015657 MWIRE6 ,MESSAGE
1309 003254 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1310 003256 001147 SEXMIT ,THIS FLAG
1311 003260 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1312 003262 015730 MWIRE5 ,MESSAGE
1313 003264 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1314 003266 001150 SEREC ,THIS FLAG
1315 003270 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1316 003272 016000 MWIRE4 ,MESSAGE
1317 003274 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1318 003276 001151 OPTCLR ,THIS FLAG
1319 003300 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1320 003302 016057 MEXTJ ,MESSAGE
1321 003304 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1322 003306 001153 JMRBY ,THIS FLAG
1323
1324 ,TEST START AND RESTART
1325
1326 003310 012706 001100 BEGIN MOV #STACK,SP ,SET UP STACK
1327 003314 106427 000340 MTPS #340 ,LOCK OUT INTERRUPTS
1328 003320 032777 000002 176112 BIT #SW01,DSWR IF SW01=1, GET STARTING PC
1329 003326 001413 BEQ 3$
1330 003330 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1331 003332 015543 MTSTPC ,MESSAGE
1332 003334 104410 PARAM ;CONVERT STRING
1333 003336 003374 TST1 ,LOW LIMIT
1334 003340 017500 17500 ,HIGH LIMIT
1335 003342 001402 $TSTNM ,STORE AT THIS LOCATION
1336 003344 001 BYTE 1 ,MASK
1337 003345 001 BYTE 1 ,HOW MANY TIMES + 2
1338 003346 016767 176030 175530 MOV $TSTNM,RETURN
1339 003354 000403 BR 4$
1340 003356 012767 003374 175520 3$ MOV #TST1,RETURN ,START AT TEST 1
1341 003364 104401 015537 4$ TYPE ,MR ,TYPE R
1342 003370 000177 175510 JMP @RETURN ,START TESTING
1343

```

```

1344
1345      ;, THIS TEST VERIFYS THAT RXDONE CAUSES AN INTERRUPT
1346      ;, MODE SYNC EXTERNAL
1347      ;, INTERRUPT VECTOR DURIV
1348      ;, LENGTH EIGHT
1349      ;, *****
1350 003374 000004      TST1 SCOPE
1351
1352 003376 052777 000400 176322      BIS #MRESET, @TXCSR ; MASTER RESET
1353 003404 012777 020000 176310      MOV #SYNEXT, @PARCSR ; SET THE MODE
1354 003412 052777 000400 176306      BIS #MRESET, @TXCSR ; MASTER RESET
1355
1356      ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1357 003420 012777 064001 176300      MOV #MTDATA!CLK!MINT!BREAK, @TXCSR
1358
1359      ; SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
1360 003426 012777 026026 176266      MOV #SYNEXT!EIGHT!NOPAR!26, @PARCSR
1361 003434 052777 000020 176250      BIS #SYNSCH, @RXCSR ; SET SEARCH SYNC
1362      ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1363 003442 042777 020000 176256      BIC #CLK, @TXCSR ; POKE CLK DOWN
1364 003450 052777 020000 176250      BIS #CLK, @TXCSR ; POKE CLK UP
1365 003456 012777 003500 176252      MOV #15, @DURIV ; SET UP TRAPCATCHER
1366 003464 016777 013322 176246      MOV DUPRT, @DURIS
1367 003472 106427 000000      MTPS #0 ; ALLOW INTERRUPTS
1368 003476 000424      BR 25 ; JUMP AROUND INTERRUPT SVC ROUTINE
1369      ; THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1370 003500 106427 000340      15 MTPS #340 ; DON'T ALLOW ANYMORE INTERRUPTS
1371 003504 042777 000100 176200      BIC #RINTEN, @RXCSR ; CLEAR INTERRUPT ENABLE
1372 003512 105777 176174      TSTB @RXCSR ; RXDONE=1?
1373 003516 100401      BMI +4
1374 003520 104004      ERROR 4 ; FALSE INTERRUPT
1375 003522 012716 003714      MOV #3$, (SP) ; SET UP RETURN LOCATION
1376 003526 016777 176206 176202      MOV DURIS, @DURIV ; RESTORE TRAPCATCHER
1377 003534 012777 000000 176176      MOJ #0, @DURIS
1378 003542 017701 176150      MOV @RXDBUF, R1 ; CLEAR INTERRUPT
1379 003546 000002      RTI
1380
1381 003550 052777 000100 176134 25 BIS #RINTEN, @RXCSR ; SET INTERRUPT ENABLE
1382 003556 012767 000010 175336      MOV #8, SHIFT ; # OF SHIFTS
1383 003564 012767 000025 175706      MOV #25, $TMP1 ; TO BE SHIFTED CHARACTER
1384
1385      ; THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
1386      ; INFORMATION CONTAINED IN $TMP1 AND IT IS
1387      ; SHIFTED IN BY THE CONTENTS OF SHIFT
1387 003572 042777 040000 176126 55 BIC #MTDATA, @TXCSR
1388 003600 000241      CLC
1389 003602 006067 175672      ROR $TMP1 ; FORCE CARRY
1390 003606 103003      BCC 45
1391 003610 052777 040000 176110      BIS #MTDATA, @TXCSR
1392 003616 042777 020000 176102 45 BIC #CLK, @TXCSR
1393 003624 052777 020000 176074      BIS #CLK, @TXCSR
1394 003632 005367 175264      DEC SHIFT
1395 003636 001355      BNE 55
1396      ; INTERRUPT SHOULD NOW OCCUR
1397 003640 005000      CLR R0
1398 003642 005200      INC R0 ; WA T FOR INTERRUPT
1399 003644 001376      BNE -2
  
```

INITIALIZE THE COMMON TAGS

```
1400 003646 016777 176066 176062      MOV    DURIS,@DURIV    ,RESTORE TRAPCATCHER
1401 003654 012777 000000 176056      MOV    #0,@DURIS      /
1402 003662 016703 176030      MOV    RXDBUF,R3      ,FOR ERROR MESSAGE
1403 003666 012700 000025      MOV    #25,R0         ,EXPECTED
1404 003672 017701 176020      MOV    @RXDBUF,R1
1405 003676 042777 000100 176006      BIC    #RINTEN,@RXCSR ,CLR INTR ENABLE
1406 003704 020001      CMP    R0,R1
1407 003706 001401      BEQ    +4
1408 003710 104002      ERROR  2              ,CHARACTERS SHOULD COMPARE
1409 003712 104004      ERROR  4              , INTERRUPT FAILED TO OCCUR
1410
1411 003714 106427 000340      35    MTPS    #340
1412
1413
1414
1415      , , THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP TO
1416      , , THE SAME VECTOR ARE BOTH EXECUTED
1417      , , INTERRUPT VECTOR DURIV
1418      , , THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
1419
1420      , , *****
1421 003720 000004      TST2   SCOPE
1422 003722 105767 175225      TSTB   JMRBY          , IN MAINT EXTERNAL?
1423 003726 100402      BMI    +6            , IF ANSWER WAS YES DO THIS TEST
1424 003730 000167 000402      JMP    1$           , IF ANSWER WAS NO JUMP AROUND TEST
1425 003734 052777 000400 175764      BIS    #MRESET,@TXCSR ,MASTER RESET
1426 003742 012777 020000 175752      MOV    #SYNEXT,@PARCSR ,SET THE MODE
1427 003750 052777 000400 175750      BIS    #MRESET,@TXCSR ,MASTER RESET
1428
1429      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1430 003756 012777 064001 175742      MOV    #MTDATA'CLK'MINT'BREAK @TXCSR
1431
1432      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC PEG
1433 003764 012777 026026 175730      MOV    #SYNEXT'EIGHT'NOPAR'26,@PARCSR
1434 003772 052777 000020 175712      BIS    #SYNSCH,@RXCSR  ,SET SEARCH SYNC
1435      , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1436 004000 042777 020000 175720      BIC    #CLK,@TXCSR    ,POKE CLK DOWN
1437 004006 052777 020000 175712      BIS    #CLK,@TXCSR    ,POKE CLK UP
1438 004014 012777 004036 175714      MOV    #25,@DURIV    ,SET UP TRAPCATCHER
1439 004022 016777 012764 175710      MOV    DUPRT,@DURIS  /
1440 004030 106427 000000      MTPS   #0            ,ALLOW INTERRUPT
1441 004034 000454      BR     3$            ,JLMP AROUND SVC ROUTINE
1442
1443      , THE FOLLOWING IS THE 1ST INTERRUPT SVC RCUTINE
1444 004036 106427 000340      2$    MTPS   #340     ,DON'T ALLOW ANY MORE INTERRUPTS
1445 004042 105777 175644      TSTB   @RXCSR        ,RXDONE = 1 ?
1446 004046 100401      BMI    +4
1447 004050 104004      ERROR  4              , FALSE INTERRUPT
1448 004052 012716 004332      MOV    #5$,(SP)      ,SET UP RETURN LOCATION
1449 004056 012777 004136 175652      MOV    #4$,@DURIV    ,SET UP TRAPCATCHER FOR SECOND
1450      , INTERRUPT
1451 004064 052777 000002 175620      BIS    #DTR,@RXCSR   ,TRY TO CAUSE SECOND INTERRUPT
1452 004072 017701 175620      MOV    @RXDBUF,R1    ,JUST READ RXDBUF TO CLR RXDONE
1453      , TO ALLOW SECOND INTERRUPT
1454 004076 106427 000000      MTPS   #0            ,ALLOW INTERRUPT
1455 004102 005000      CLP    R0
```


INITIALIZE THE COMMON TAGS

```
1456 004104 005200          INC      R0          ,WAIT FOR INTERRUPT
1457 004106 001376          BNE     -2          ,
1458 004110 042777 000140 175574 BIC     #RINTEN'DSINTE,@RXCSR ,CLR INTR ENABLES
1459 004116 104004          ERROR   4          ,2ND INTERRUPT FAILED TO OCCUR
1460
1461 004120 016777 175614 175610 65     MOV     @DURIS,@DURIV ,RESTORE TRAPCATCHER
1462 004126 012777 000000 175604          MOV     #0,@DURIS ,
1463 004134 000002          RTI
1464
1465          ,THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1466 004136 106427 000340 45     MTPS   #340 ,DON'T ALLOW ANYMORE INTERRUPTS
1467 004142 005777 175544          TST     @RXCSR ,DSC = 1 ?
1468 004146 100401          BMI     +4
1469 004150 104004          ERROR   4          ,FALSE INTERRUPT
1470 004152 042777 000140 175532 BIC     #RINTEN'DSINTE,@RXCSR ,CLR BOTH INTR ENABLES
1471 004160 012716 004120          MOV     #65,(SP) ,SET UP RETURN LOCATION
1472 004164 000002          RTI
1473
1474 004166 052777 000140 175516 35     BIS     #RINTEN'DSINTE,@RXCSR ,SET INTERRUPT ENABLES
1475 004174 012767 000010 174720          MOV     #8,SHIFT ,# OF SHIFTS
1476 004202 012767 000025 175270          MOV     #25,$TMP1
1477
1478          ,THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
1479          ,INFORMATION CONTAINED IN $TMP1 AND IT IS
1480          ,SHIFTED IN BY THE CONTENTS OF SHIFT
1480 004210 042777 040000 175510 85     BIC     #MTDATA,@TXCSR
1481 004216 000241          CLC
1482 004220 006067 175254          ROR     $TMP1 ,FORCE CARRY
1483 004224 103003          BCC     75
1484 004226 052777 040000 175472          BIC     #MTDATA,@TXCSR
1485 004234 042777 020000 175464 75     BIC     #CLK,@TXCSR
1486 004242 052777 020000 175456          BIS     #CLK,@TXCSR
1487 004250 005367 174646          DEC     SHIFT
1488 004254 001355          BNE     85
1489          ,1ST INTERRUPT SHOULD NOW OCCUR
1490 004256 005000          CLR     R0
1491 004260 005200          INC     R0          ,WAIT FOR INTERRUPT
1492 004262 001376          BNE     -2          ,
1493 004264 016777 175450 175444          MOV     @DURIS,@DURIV ,RESTORE TRAPCATCHER
1494 004272 012777 000000 175440          MOV     #0,@DURIS ,
1495 004300 016703 175412          MOV     @RXDBUF,R3 ,FOR ERROR MESSAGE
1496 004304 012700 000025          MOV     #25,R0 ,EXPECTED
1497 004310 017701 175402          MOV     @RXDBUF,R1
1498 004314 042777 000140 175370 BIC     #RINTEN'DSINTE,@RXCSR ,CLR INTERRUPT ENABLES
1499 004322 020001          CMP     R0,R1
1500 004324 001401          BEQ     +4
1501 004326 104002          ERROR   2          ,CHARACTERS SHOULD COMPARE
1502 004330 104004          ERROR   4          ,INTERRUPT FAILED TO OCCUR
1503
1504 004332 106427 000340 55     MTPS   #340 ,DON'T ALLOW AN' MORE INTERRUPTS
1505 004336 15
1506
1507          THIS TEST VERIFYS THAT DNA CAUSES AN INTERRUPT
1508          ..MODE SYNC EXTERNAL
1509          INTERRUPT VECTOR DUTIV
1510
1511          *****
```

```

1512 004336 000004          TST3  SCOPE
1513
1514 004340 052777 000400 175360      BIS    #MRESET,@TXCSR ,MASTER RESET
1515 004346 012777 020000 175346      MOV    #SYNEXT,@PARCSR ,SET THE MODE
1516 004354 052777 000400 175344      BIS    #MRESET,@TXCSR ,MASTER RESET
1517
1518          ,SET MA NTENANCE MODE & SEND
1519          ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1520 004362 012777 004020 175336      MOV    #MINT'SEND,@TXCSR
1521
1522          ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1523 004370 012777 026026 175324      MOV    #SYNEXT'EIGHT'NOPAR'26,@PARCSR
1524 004376 112777 000025 175326      MOV    #25,@TXDBUF ,LOAD CHARACTER
1525 004404 012767 000010 174510      MOV    #8 ,SHIFT
1526          ,POKE CLK TO GET INTO SYNCRONIZATION
1527 004412 052777 020000 175306      BIS    #CLK,@TXCSR ,POKE CLK UP
1528 004420 042777 020000 175300      BIC    #CLK,@TXCSR ,POKE CLK DOWN
1529
1530 004426          15
1531 004426 052777 020000 175272      BIS    #CLK,@TXCSR ,POKE CLK UP
1532 004434 042777 020000 175264      BIC    #CLK,@TXCSR ,POKE CLK DOWN
1533 004442 005367 174454          DEC    SHIFT ,LAST SHIFT?
1534 004446 001367          BNE    15
1535 004450 012777 004516 175264      MOV    #25,@DUTIV ,SET UP TRAPCATCHER
1536 004456 016777 012330 175260      MOV    DUPRT,@DUTIS ,
1537 004464 106427 000000          MTPS   #0 ,ALLOW INTERRUPTS
1538 004470 052777 000040 175230      BIS    #DNAINTE,@TXCSR ,ENABLE INTERRUPT
1539          ,NOW POKE CLK TO GET DNA
1540 004476 052777 020000 175222      BIC    #CLK,@TXCSR ,POKE CLK
1541 004504 005000          CLR    R0
1542 004506 005200          INC    R0 ,WAIT FOR INTERRUPT
1543 004510 001376          BNE    -2
1544 004512 104004          ERROR  4 ,INTERPUPT FAILED TO OCCUR
1545 004514 000422          BR     35 ,JUMP AROUND SVC ROUTINE
1546          ,THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1547 004516 106427 000340 25      MTPS   #340 ,DON'T ALLOW ANYMORE INTERRUPTS
1548 004522 005777 175200          TST   @TXCSR ,DNA?
1549 004526 100401          BMI   +4
1550 004530 104004          ERROR  4 ,FALSE INTERRUPT
1551 004532 042777 000040 175166      BIC    #DNAINTE,@TXCSR ,CLR INTR ENABLE
1552 004540 012716 004604          MOV    #4$(SP) ,SET UP RETURN LOCATION
1553 004544 016777 175174 175170      MOV    DUTIS,@DUTIV ,RESTORE TRAPCATCHER
1554 004552 012777 000000 175164      MOV    #0,@DUTIS ,
1555 004560 000002          RTI
1556
1557 004562 016777 175156 175152 35      MOV    DUTIS,@DUTIV ,RESTOPE TRAPCATCHER
1558 004570 012777 000000 175146      MOV    #0,@DUTIS ,
1559
1560 004576 042777 000040 175122          BIC    #DNAINTE,@TXCSR ,CLR INTERRUPT ENABLE
1561 004604 106427 000340 45      MTPS   #340 ,RESTORE NO INTERRUPT STATUS
1562
1563          ,, THIS TEST VERIFYS THAT TXDONE CAUSES AN INTERRUPT
1564          ,, INTERRUPT VECTOR DUTIV
1565          ,, NOTE TXDONE = 1 AFTER A MASTER PESET
1566          ,,
1567

```

```

1568
1569 004610 000004
1570
1571 004612 052777 000400 175106      BIS      #MRESET,@TXCSR  ,MASTER RESET
1572 004620 012777 004666 175114      MOV      #15,@DUTIV ,SET UP TRAPCATCHER
1573 004626 016777 012160 175110      MOV      DUPRT,@DUTIS ,
1574 004634 106427 000000      MTPS     #0          ,ALLOW INTERPUTS
1575 004640 052777 000100 175060      BIS      #TXINTE,@TXCSR ,ENABLE INTERRUPT
1576 004646 005000      CLR      R0          ,WAIT FOR INTERRUPT
1577 004650 005200      INC      R0          ,
1578 004652 001376      BNE      -2          ,
1579 004654 042777 000100 175044      BIC      #TXINTE,@TXCSR ,CLR INTERRUPT ENABLE
1580 004662 104004      ERROR    4          ,INTERRUPT FAILED TO OCCUR
1581 004664 000422      BR       25         ,JUMP AROUND SVC ROUTINE
1582
1583      ,THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1584 004666 106427 000340      15      MTPS     #340      ,DON'T ALLOW ANYMORE INTERRUPTS
1585 004672 042777 000100 175026      BIC      #TXINTE,@TXCSR ,CLR INTR ENABLE
1586 004700 105777 175022      TSTB     @TXCSR      ,TXDONE?
1587 004704 100401      BMI      +4          ,
1588 004706 104004      ERROR    4          ,FALSE INTERRUPT
1589 004710 012716 004746      MOV      #35,(SP)    ,SET UP RETURN LOCATION
1590 004714 016777 175024 175020      MOV      DUTIS,@DUTIV ,RESTORE TRAPCATCHER
1591 004722 012777 000000 175014      MOV      #0,@DUTIS  ,
1592 004730 000002      RTI
1593
1594 004732 016777 175006 175002      25      MOV      DUTIS,@DUTIV ,RESTORE TRAPCATCHER
1595 004740 012777 000000 174776      MOV      #0,@DUTIS  ,
1596
1597 004746 106427 000340      35      MTPS     #340      ,RESTORE NO INTERRUPT STATUS
1598
1599
1600
1601      ,,THIS TEST VERIFYS THAT TXDONE DOES NOT CAUSE AN INTERPUPT
1602      ,,WHEN PROCESSOR PRIORITY LEVEL IS TOO HIGH
1603      ,,INTERRUPT VECTOR DUTIV
1604      ,,NOTE TXDONE = 1 AFTER A MASTER RESET
1605
1606      ,
1607 004752 000004      TST5     SCOPE
1608 004754 052777 000400 174744      BIS      #MRESET,@TXCSR ,MASTER RESET
1609 004762 012777 005046 174752      MOV      #15,@DUTIV ,SET UP TRAPCATCHER
1610 004770 016777 012016 174746      MOV      DUPRT,@DUTIS ,
1611 004776 106427 000340      MTPS     #340      ,SET PS LEVEL TOO HIGH
1612 005002 052777 000100 174716      BIS      #TXINTE,@TXCSR ,ENABLE INTERRUPT
1613 005010 005000      CLR      R0          ,WAIT FOR INTERRUPT
1614 005012 005200      INC      R0          ,
1615 005014 001376      BNE      -2          ,
1616 005016 042777 000100 174702      BIC      #TXINTE,@TXCSR ,CLR INTR ENABLE
1617 005024 106427 000340      MTPS     #340      ,DON'T ALLOW INTERRUPTS
1618 005030 016777 174710 174704      MOV      DUTIS,@DUTIV ,RESTORE TRAPCATCHER
1619 005036 012777 000000 174700      MOV      #0,@DUTIS  ,
1620 005044 000421      BR       25         ,TEST IS OK GET OUT OF TEST
1621      ,THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1622 005046 106427 000340      15      MTPS     #340      ,DONT ALLOW ANYMORE INTERRUPTS
1623 005052 042777 000100 174646      BIC      #TXINTE,@TXCSR ,CLR INTR ENABLE
  
```

```

1624 005060 012716 005102          MOV    #3$, (SP)      , SET UP RETURN LOCATION
1625                                     , TO REPORT ERROR
1626 005064 016777 174654 174650    MOV    DUTIS, @DUTIV , RESTORE TRAPCATCHER
1627 005072 012777 000000 174644    MOV    #0, @DUTIS   ,
1628 005100 000002                                     RTI
1629                                     , END OF INTERRUPT SVC ROUTINE
1630
1631
1632
1633                                     , YOU SHOULD NOT GET INTO THIS FOLLOWING CODE UNLESS THERE
1634                                     , WAS AN ERROR
1635 005102 106427 000340    3$    MTPS    #340    , DON'T ALLOW ANYMORE INTERRUPTS
1636 005106 104004                                     ERROR    4          , INTERRUPT SHOULD NOT OF OCCURED, CHECK
1637                                     , THE INTERRUPT LEVEL SELECTED OP CHECK
1638                                     , INTERRUPT LOGIC OR BOTH
1639 005110    2$
1640
1641                                     , THIS TEST VERIFYS THAT TXDONE CAUSES ONLY ONE INTERRUPT
1642                                     , PROVIDING THAT TXCSR IS NOT READ
1643                                     , AND TXDBUF IS NOT LOADED (WRITTEN)
1644                                     , THIS TEST CHECKS THE ONCE ONLY FLIP/FLOP (V2)
1645                                     , OF THE INTERRUPT CONTROL LOGIC
1646                                     , INTERRUPT VECTOR DUTIV
1647                                     , NOTE TXDONE = 1 AFTER A MASTER RESET
1648
1649                                     , *****
1650 005110 000004    TST6   SCOPE
1651 005112 052777 000400 174606    BIS    #MRESET, @TXCSR , MASTER RESET
1652 005120 012777 005160 174614    MOV    #1$, @DUTIV    , SET UP TRAPCATCHER
1653 005126 016777 011660 174610    MOV    DUPRT, @DUTIS ,
1654 005134 106427 000000    MTPS   #0            , ALLOW INTERRUPTS
1655 005140 052777 000100 174560    BIS    #TXINTE, @TXCSR , ENABLE INTR ENABLE
1656 005146 005000    CLR    RD
1657 005150 005200    INC    RD
1658 005152 001376    BNE    -2
1659 005154 104004    ERROR  4            , INTERRUPT FAILED TO OCCUR
1660 005156 000425    BR     4$
1661                                     , THE FOLLOWING IS THE INTR SVC ROUTINE
1662 005160 106427 000340    1$    MTPS    #340    , DON'T ALLOW ANYMORE INTR
1663 005164 012716 005224    MOV    #3$, (SP)    , SET UP RETURN LOCATION
1664 005170 012777 005200 174544    MOV    #2$, @DUTIV  , SET UP TRAPCATCHER TO
1665                                     , PROVE THAT THE INTERRUPT DOES NOT OCCUR
1666                                     , TWICE (AFTER RTI 'ING FROM THIS
1667                                     , SVC ROUTINE
1668 005176 000002    RTI
1669                                     , THE FOLLOWING INTERRUPT SVC ROUTINE WILL CATCH THE SECOND INTR
1670 005200 106427 000340    2$    MTPS    #340    , DON'T ALLOW INTER
1671 005204 012716 005232    MOV    #4$, (SP)    , SET UP RETURN LOCATION
1672 005210 105777 174512    TSTB  @TXCSR    , TXDONE = 1?
1673 005214 100401    BMI    +4
1674 005216 104004    ERROR  4            , TXDONE SHOULD BE SET
1675 005220 104004    ERROR  4            , THE INTERRUPT WAS TAKEN TWICE
1676                                     , CHECK OUT THE V2 FLIP/FLOP LOGIC
1677                                     , IN THE INTERRUPT CONTROL LOGIC
1678 005222 000002    RTI
1679 005224 005000    3$    CLR    RD        , ALLOW TIME TO CATCH SECOND

```

```

1680 005226 005200          INC      RO          , IF IT WERE TO OCCUR
1681 005230 001376          BNE      -2          ,
1682 005232 016777 174506 174502 4$  MOV      DUTIS, @DUTIV , RESTORE TRAPCATCHER
1683 005240 012777 000000 174476  MOV      #0, @DUTIS ,
1684 005246 042777 000100 174452  BIC      #TXINTE, @TXCSR , CLR INTERRUPT ENABLE
1685 005254 106427 000340  MTPS     #340      , RESTORE NO INTERRUPT STATUS
1686
1687
1688      , , THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP
1689      , , TO THE SAME VECTOR ARE BOTH EXECUTED
1690      , , INTERRUPT VECTOR DUTIV
1691      , , MODE SYNC EXTERNAL
1692
1693      , , *****
1694 005260 000004          TST7     SCOPE
1695
1696 005262 052777 000400 174436  BIS      #MRESET, @TXCSR , MASTER RESET
1697 005270 012777 020000 174424  MOV      #SYNEXT, @PARCSR , SET THE MODE
1698 005276 052777 000400 174422  BIS      #MRESET, @TXCSR , MASTER RESET
1699
1700      , SET MAINTENANCE MODE & SEND
1701      , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1702 005304 012777 004020 174414  MOV      #MINT'SEND, @TXCSR
1703
1704      , SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1705 005312 012777 02E026 174402  MOV      #SYNEXT'EIGHT'NOPAR'26, @PARCSR
1706 005320 112777 000025 174404  MOVB     #25, @TXDBUF , LOAD CHARACTER
1707 005326 012767 000010 173566  MOV      #8 , SHIFT
1708
1709      , POKE CLK TO GET INTO SYNCHRONIZATION
1709 005334 052777 020000 174364  BIS      #CLK, @TXCSR , POKE CLK UP
1710 005342 042777 020000 174356  BIC      #CLK, @TXCSR , POKE CLK DOWN
1711
1712      1$
1713 005350 052777 020000 174350  BIS      #CLK, @TXCSR , POKE CLK UP
1714 005356 042777 020000 174342  BIC      #CLK, @TXCSR , POKE CLK DOWN
1715 005364 005367 173532  DEC      SHIFT      , LAST SHIFT?
1716 005370 001367  BNE      1$
1717 005372 012777 005432 174342  MOV      #25, @DUTIV , SET UP TRAPCATCHER
1718 005400 016777 011406 174336  MOV      DUPRT, @DUTIS ,
1719 005406 106427 000000  MTPS     #0          , ALLOW INTERRUPTS
1720 005412 052777 000140 174306  BIS      #TXINTE'DNAINTE, @TXCSR , ENABLE INTERRUPTS
1721 005420 005000  CLR      RO
1722 005422 005200  INC      RO          , WAIT FOR INTERRUPT
1723 005424 001376  BNE      -2          ,
1724 005426 104004  ERROR    4          , INTERRUPT FAILED TO OCCUR
1725 005430 000461  BR       3$          , JUMP AROUND SVC ROUTINES
1726
1727      , THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
1728 005432 106427 000340 2$  MTPS     #340      , DON'T ALLOW AN'MORE INTERRUPTS
1729 005436 005777 174264  TST      @TXCSR , DNA=0 ?
1730 005442 100001  BPL      +4
1731 005444 104004  ERROR    4          , DNA SHOULD NOT BE ASSERTED
1732 005446 105777 174254  TSTB     @TXCSR , TXDONE = 1?
1733 005452 100401  BMI      +4
1734 005454 104004  ERROR    4          , FALSE INTERRUPT
1735 005456 012716 005616  MOV      #4$, (SP) , SET UP RETURN LOCATION

```

```
1736 005462 012777 005544 174252      MOV      #55, @DUTIV      , SET UP TRAPCATCHER
1737                                     , NOW POKE CLK TO BRING UP DNA
1738 005470 052777 020000 174230      BIS      #CLK, @TXCSR     , POKE CLK
1739 005476 112777 000025 174226      MOV      #25, @TXDBUF     , JUST LOAD ANY CHAR TO CLR
1740                                     ; TXDONE TO ALLOW SECOND INTERRUPT
1741 005504 106427 000000      MTPS    #0              ; ALLOW INTERRUPTS
1742 005510 005000      CLR      R0
1743 005512 005200      INC      R0              ; WAIT FOR INTERRUPT
1744 005514 001376      BNE      -2
1745 005516 042777 000140 174202      BIC      #DNAINTE!TXINTE, @TXCSR , CLR INTR ENABLES
1746 005524 104004      ERROR   4              , 2ND INTERRUPT FAILED TO OCCUR
1747
1748 005526 016777 174212 174206 65      MOV      DUTIS, @DUTIV    , RESTORE TRAPCATCHER
1749 005534 012777 000000 174202      MOV      #0, @DUTIS
1750 005542 000002      RTI
1751
1752                                     , THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1753 005544 106427 000340 55      MTPS    #340
1754 005550 005777 174152      TST      @TXCSR          , DNA
1755 005554 100401      BMI      +4
1756 005556 104004      ERROR   4              , FALSE INTERRUPT
1757 005560 042777 000140 174140      BIC      #DNAINTE!TXINTE, @TXCSR , CLR BOTH INTR ENABLES
1758 005566 012716 005526      MOV      #65, (SP)      , SETUP RETURN LOCATION
1759 005572 000002      RTI
1760
1761 005574 016777 174144 174140 35      MOV      DUTIS, @DUTIV    , RESTORE TRAPCATCHER
1762 005602 012777 000000 174134      MOV      #0, @DUTIS
1763
1764 005610 042777 000140 174110      BIC      #DNAINTE!TXINTE, @TXCSR , CLR BOTH INTERRUPT
1765                                     , ENABLES
1766 005616 106427 000340 45      MTPS    #340          ; RESTORE NC INTERRUPT STATUS
1767
1768
1769
1770                                     , , THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1771                                     , , IT BASICALLY CHECKS THE EXISTANCE OF
1772                                     , , THE FREE RUNNING OSCILLATOR
1773                                     , , MODE SYNEXT
1774                                     , , LENGTH: EIGHT
1775                                     , , THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1776
1777                                     , , *****
1778 005622 000004      TST10   SCOPE
1779                                     , , *****
1780 005624 000167 000262      JMP      15              , NOP THIS TEST
1781                                     , , *****
1782 005630 052777 000400 174070      BIS      #MRESET, @TXCSR , MASTER RESET
1783 005636 012777 020000 174056      MOV      #SYNEXT, @PARCSR , LOAD THE MODE
1784 005644 052777 000400 174054      BIS      #MRESET, @TXCSR , MASTER RESET
1785 005652 012777 026026 174042      MOV      #SYNEXT!EIGHT!NOPAR!26, @FARCSR , LOAD THE MODE,
1786                                     , # OF BITS PER CHAR, PARITY SENSE(NO PARITY),
1787                                     , & SYNC CHARACTER (26)
1788 005660 112777 000025 174044      MOV      #25, @TXDBUF     , LOAD THE CHAR
1789 005666 012777 005764 174042      MOV      #25, @DUIV       , SET UP TRAPCATCHER
1790 005674 016777 011112 174036      MOV      DUPRT, @DURIS
1791 005702 106427 000000      MTPS    #0              , ALLOW INTERRUPTS
```

```

1792 005706 016703 174004      MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
1793 005712 012700 000025      MOV    #25,R0        ;EXPECTED
1794 005716 012777 014020 174002  MOV    #SYSTST!SEND,@TXCSR ;OK NOW LOAD SEND &
1795                                     ,MAINT MODE
1796 005724 052777 000120 173760  BIS    #SYNSCH!RINTEN,@RXCSR ;SET SEARCH SYNC &
1797                                     ,RECEIVER INTERRUPT
1798                                     ,ENABLE & WAIT FOR INTERRUPT
1799 005732 005067 173552          CLR    $TMP5
1800 005736 005002          3$ CLR    R2
1801 005740 005202          INC    R2            ,WAIT FOR INTERRUPT
1802 005742 001376          BNE    -2
1803 005744 005267 173540          INC    $TMP5
1804 005750 022767 000003 173532  CMP    #3,$TMP5
1805 005756 002367          BGE    3$
1806 005760 104004          ERROR  4            , INTERRUPT DID NOT OCCUR
1807 005762 000422          BR     4$
1808
1809                                     , THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1810 005764 106427 000340          2$ MTPS #340 , PREVENT INTERRUPTS
1811 005770 017704 173716          MOV    @RXCSR,R4    , SAVE
1812 005774 017701 173716          MOV    @RXDBUF,R1  , ACTUAL
1813 006000 016777 173734 173730  MOV    DURIS,@DURIV , RESTORE TRAPCATCHER
1814 006006 012777 000000 173724  MOV    #0,@DURIS
1815 006014 012716 006060          MOV    #5,(SP)     , SET UP RETURN
1816 006020 042777 000100 173664  BIC    #RINTEN,@RXCSR , CLR INTERRUPT ENABLE
1817 006026 000002          RTI
1818
1819 006030 042777 000100 173654  4$ BIC    #RINTEN,@RXCSR , CLR INTERRUPT ENABLE
1820 006036 106427 000340          MTPS #340 , PREVENT INTERRUPTS
1821 006042 016777 173672 173666  MOV    DURIS,@DURIV , RESTORE TRAPCATCHER
1822 006050 012777 000000 173662  MOV    #0,@DURIS
1823 006056 000415          BR     1$
1824
1825 006060 020001          5$ CMP    R0,R1
1826 006062 001401          BEQ   +4
1827 006064 104002          ERROR  2            , CHARACTERS DID NOT MATCH
1828 006066 016703 173620          MOV    RXCSR,R3    , SETUP FOR ERROR MESSAGE
1829 006072 012700 000200          MOV    #200,R0     , EXPECTED
1830 006076 010401          MOV    R4,R1       , ACTUAL
1831 006100 042701 177577          BIC    #177577,R1  , SAVE ONLY PxDONE
1832 006104 020001          CMP    R0,R1
1833 006106 001401          BEQ   +4
1834 006110 104001          ERROR  1            , FALSE INTERRUPT
1835
1836 006112          1$
1837                                     ,, THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1838                                     ,, IT BASICALLY CHECKS THE EXISTANCE OF
1839                                     ,, THE FREE RUNNING OSCILLATOR
1840                                     ,, MODE SYNINT
1841                                     ,, LENGTH EIGHT
1842                                     ,, THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1843                                     ,,
1844                                     ,, *****
1845 006112 000004          TST11 SCOPE
1846 006114 052777 000400 173604  BIS    #MRESET,@TXCSR , MASTER RESET
1847 006122 012777 030000 173572  MOV    #SYNINT,@PARCSR , SET THE MODE

```

```

1848 006130 052777 000400 173570      BIS      #MRESET,@TXCSR ;MASTER RESET
1849
1850                                     ;SET MAINTENANCE MODE & SEND
1851                                     ;NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1852 006136 012777 014020 173562      MOV      #SYSTST!SEND,@TXCSR
1853
1854                                     ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1855 006144 012777 036026 173550      MOV      #SYNINT!EIGHT!NOPAR!26,@PARCSR
1856 006152 052777 000420 173532      BIS      #SYNSCH!STPSYN,@RXCSR ;SET SEARCH SYNC &
1857                                     ;STRIP SYNC SO THAT RXDONE ASSERTS
1858                                     ;WHEN CHAR "25" ARRIVES AND NOT BEFORE
1859                                     ;...THEREFORE, SET STRIP SYNC
1860                                     ;WAIT FOR SYNSCH TO BE
1861                                     ;CLOCKED IN BY SYSTST CLK
1862 006160 005067 173324      CLR      $TMP5
1863 006164 005002      CLR      R2
1864 006166 005202      INC      R2 ;WAIT
1865 006170 001376      BNE      -2
1866 006172 005267 173312      INC      $TMP5
1867 006176 022767 000003 173304      CMP      #3,$TMP5
1868 006204 002367      BGE      -20 ;GO BACK TO CLP R2 AND WAIT SOME MORE
1869 006206 012777 006422 173522      MOV      #25,@DURIV ;SET UP TRAPCATCHER
1870 006214 016777 010572 173516      MOV      DUPRT,@DURIS
1871 006222 012777 006514 173512      MOV      #35,@DUTIV
1872 006230 016777 010556 173506      MOV      DUPRT,@DUTIS
1873 006236 106427 000000      MTPS     #0 ;ALLOW INTERRUPTS
1874 006242 016703 173450      MOV      RXDBUF,R3 ;SET UP FOR ERROR MSG
1875 006246 012700 000025      MOV      #25,R0 ;EXPECTED CHAR
1876 006252 012767 000002 172644      MOV      #2,COUNT ;# OF SYNC CHARS TO GET INTO
1877                                     ;SYNCHRONIZATION
1878 006260 105767 172662      TSTB    SYNCNO ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
1879 006264 100402      BMI     95
1880 006266 005367 172632      DEC     COUNT ;MAKE IT ONE LESS
1881 006272 052777 000100 173412 95      BIS     #RINTEN,@RXCSR ;SET INTERPUPT ENABLES
1882 006300 052777 000100 173420      BIS     #TXINTE,@TXCSR
1883 006306 000167 000012      JMP     85 ;THE FIRST XMIT INTERRUPT SHOULD COME
1884                                     ;FROM TXDONE = 1 AFTER A MASTER RESET
1885 006312 112777 000026 173412 15      MOVB   #26,@TXDBUF ;LOAD SYNC CHAR
1886 006320 005067 173164      CLR     $TMP5
1887 006324 005002      CLR     R2 ;WAIT FOR INTERRUPT
1888 006326 005202      INC     R2
1889 006330 001376      BNE     -2
1890 006332 005267 173152      INC     $TMP5
1891 006336 022767 000003 173144      CMP     #3,$TMP5
1892 006344 002367      BGE     85
1893 006346 106427 000340      MTPS   #340 ;PREVENT INTERRUPTS
1894 006352 042777 000100 173346      BIC     #TXINTE,@TXCSR ;CLR INTR ENABLES
1895 006360 042777 000100 173324      BIC     #RINTEN,@RXCSR
1896 006366 016777 173346 173342      MOV     DURIS,@DURIV ;RESTORE TRAPCATCHER
1897 006374 012777 000000 173336      MOV     #0,@DURIS
1898 006402 016777 173336 173332      MOV     DUTIS,@DUTIV
1899 006410 012777 000000 173326      MOV     #0,@DUTIS
1900 006416 104004      ERROR  4 ;TXDONE INTERRUPT FAILED TO OCCUR
1901 006420 000540      BR     75 ;GET OUT OF THE TEST

```

. THE FOLLOWING IS THE RECEIVER INTEPRUPT SVC ROUTINE

1902
1903

INITIALIZE THE COMMON TAGS

```

1904 006422 106427 000340      25  MTPS      #340      ,PREVENT INTERRUPTS
1905 006426 017704 173260      MOV      @RXCSR,R4      ,SAVE
1906 006432 017701 173260      MOV      @RXDBUF,R1     ,ACTUAL
1907 006436 016777 173276 173272  MOV      DURIS,@DURIV   ,RESTORE TRAPCATCHER
1908 006444 012777 000000 173266  MOV      #0,@DURIS     ,
1909 006452 016777 173266 173262  MOV      DUTIS,@DUTIV   ,
1910 006460 012777 000000 173256  MOV      #0,@DUTIS     ,
1911 006466 012716 006646      MOV      #4$, (SP)     ,SET UP RETURN LOCATION
1912 006472 042777 000100 173212  BIC      #RINTEN,@RXCSR ,CLR INTERRUPT ENABLES
1913 006500 042777 000100 173220  BIC      #TXINTE,@TXCSR ,
1914 006506 016705 172412      MOV      COUNT,R5      ,SAVE COUNT
1915 006512 000002      RTI
1916      ,END OF RECEIVER INTERRUPT SVC ROUTINE
1917      , THE FOLLOWING IS THE XMITTER INTERRUPT SVC ROUTINE
1918 006514 005367 172404      35  DEC      COUNT
1919 006520 100403      BMI      5$
1920 006522 012716 006312      MOV      #1$, (SP)     ,SET UP RETURN LOCATION
1921      , (LOAD SYNC CHARACTER AGAIN)
1922 006526 000002      RTI
1923 006530 012716 006536      55  MOV      #6$, (SP)     ,SET UP RETURN LOCATION
1924 006534 000002      RTI
1925      ,END OF XMITTER INTERRUPT SVC ROUTINE
1926 006536 112777 000025 173166  65  MOV      #25,@TXDBUF   ,LOAD CHARACTER
1927 006544 042777 000100 173154  BIC      #TXINTE,@TXCSR ,CLR INTR ENABLE
1928 006552 005067 172732      CLR      $TMP5
1929 006556 005002      105  CLR      R2           ,WAIT FOR INTERRUPT(RECEIVER)
1930 006560 005202      INC      R2
1931 006562 001376      BNE      -2           ,
1932 006564 005267 172720      INC      $TMP5
1933 006570 022767 000003 172712  CMP      #3,$TMP5
1934 006576 002367      BGE      10$
1935 006600 106427 000340      MTPS     #340      ,PPEVENT INTERRUPTS
1936 006604 042777 000100 173100  BIC      #RINTEN,@RXCSR ,CLR INTR ENABLE
1937 006612 016777 173122 173116  MOV      DURIS,@DURIV   ,RESTORE TRAPCATCHER
1938 006620 012777 000000 173112  MOV      #0,@DURIS     ,
1939 006626 016777 173112 173106  MOV      DUTIS,@DUTIV   ,
1940 006634 012777 000000 173102  MOV      #0,@DUTIS     ,
1941 006642 104004      ERROR   4           ,RECEIVER INTR FAILED TO OCCUR
1942 006644 000426      BR      7$           ,GET OUT OF TEST
1943 006646 020001      45  CMP      R0,R1
1944 006650 001401      BEQ     +4
1945 006652 104002      ERROR   2           ,CHARACTERS DID NOT MATCH
1946 006654 016703 173032  MOV      RXCSR,R3      ,SET UP FOR ERROR MSG
1947 006660 012700 000200  MOV      #200,R0      ,EXPECTED RXDONE
1948 006664 010401      MOV      R4,R1      ,ACTUAL
1949 006666 042701 177577  BIC      #177577,R1    ,SAVE ONLY RXDONE
1950 006672 020001      CMP     R0,R1
1951 006674 001401      BEQ     +4
1952 006676 104001      ERROR   1           ,FALSE INTERRUPT
1953 006700 020527 177777  CMP      R5,#-1      ,WAS COUNT ==-1 WHEN RECEIVER
1954      , INTERRUPTED ?
1955 006704 001401      BEQ     +4
1956 006706 104004      ERROR   4           , IF R5 IS GREATER THAN -1
1957      , THEN EITHER THE # OF SYNC STRAP IS WRONG
1958      , OR RXDONE IS OCCURING TOO SOON
1959 006710 026727 172210 177777  CMP      COUNT,#-1

```

```

1960 006716 001401      BEQ      +4
1961 006720 104004      ERROR    4      , IF THIS TEST FAILS, BUT THE ABOVE TEST
1962                                     , DOESN'T . . . IT MAY BE THAT CLEARING
1963                                     , TXINTE IN THE RECEIVER SVC ROUTINE
1964                                     , IS NOT STOPPING TXDONE INTERRUPTS
1965 006722 106427 000340 7$      MTPS     #340      ; INHIBIT INTERRUPTS
1966
1967
1968                                     .. THIS TEST VERIFYS MATCH DETECT & DATA RDY
1969                                     .. FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
1970                                     .. BY OBSERVING RECACT BIT
1971                                     .. IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
1972                                     .. * DEPENDENT ON MONITOR
1973                                     .. IF ONE SYNC STRAP IS SELECTED , IT WILL
1974                                     .. ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
1975                                     .. ASSERT
1976                                     .. MODE SYNC INTERNAL
1977                                     .. LENGTH FIVE
1978                                     .. SYNC CHARACTER FOR MATCH B/C
1979                                     .. THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
1980
1981                                     .. *****
1982 006726 000004      TST12   SCOPE
1983 006730 052777 000400 172770      BIS      #MRESET, @TXCSR , MASTER RESET
1984 006736 016703 172754      MOV      RXDBUF, R3      , SET UP FOR ERROR MESSAGE
1985                                     , SET SYNC INTERNAL, FIVE, NO PARITY, 0 SYNC REGISTER
1986 006742 012704 030J00      MOV      #SYNINT'FIVE'INOPAR, R4 , CREATE PARAMETERS
1987 006746 012777 004020 172752 6$      MOV      #MINT'SEND, @TXCSR , SET SEND & MAINT INTER
1988 006754 010477 172742      MOV      R4, @PARCSR , LOAD CSR
1989 006760 052777 000020 172724      BIS      #SYNSCH, @RXCSR , SET SYNC SEARCH
1990                                     , POKE CLK TO GET INTO SYNCHRONIZATION
1991                                     BOTH THE LOGIC & RECEIVER
1992 006766 052777 020000 172732      BIS      #CLK, @TXCSR , POKE CLK UP
1993 006774 042777 020000 172724      BIC      #CLK, @TXCSR , POKE CLK DOWN
1994 007002 110477 172724      MOV      R4, @TXDBUF , LOAD DATA CHARACTER
1995                                     , POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
1996 007006 052777 020000 172712      BIS      #CLK, @TXCSR , POKE CLK UP
1997 007014 042777 020000 172704      BIC      #CLK, @TXCSR , POKE CLK DOWN
1998 007022 032777 004000 172662      BIT      #RECACT, @RXCSR , RECACT ?
1999 007030 001401      BEQ      +4
2000 007032 104004      ERROR    4      , RECACT SHOULD NOT BE SET
2001 007034 000404      BR      4$
2002 007036 010477 172660 5$      MOV      R4, @PARCSR , LOAD PARCSR WITH PAPAMETERS
2003 007042 110477 172664      MOV      R4, @TXDBUF , LOAD SYNC CHAR
2004 007046 012767 000002 172050 4$      MOV      #2, COUNT , # OF SYNC CHARS
2005 007054 005777 172646 2$      TST      @TXCSR , DNA ?
2006 007060 100001      BPL      +4      ; BR IF NOT SET
2007 007062 104004      ERROR    4      , DNA SHOULD NOT BE SET OR
2008                                     , IT SHOULD BE CLEARED FROM PREVIOUS READ
2009 007064 012767 000005 172030      MOV      #5, SHIFT , # OF SHIFTS
2010 007072                                     1$
2011 007072 052777 020000 172626      BIS      #CLK, @TXCSR , POKE CLK UP
2012 007100 042777 020000 172620      BIC      #CLK, @TXCSR , POKE CLK DOWN
2013 007106 005367 172010      DEC      SHIFT , # OF SHIFTS
2014 007112 001367      BNE      1$
2015 007114 005367 172004      DEC      COUNT , # OF SYNC CHARS
    
```

INITIALIZE THE COMMON TAGS

```
2016 007120 001403 BEQ 3$  
2017 , TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED  
2018 007122 105767 172020 TSTB SYNCNO  
2019 007126 100752 BMI 2$ , TWO SYNC CHARACTERS  
2020 007130 032777 004000 172554 3$ BIT #REACT,@RXCSR , REACT ?  
2021 007136 001001 BNE +4  
2022 007140 104004 ERROR 4 ; REACT FAILED TO SET, POSSIBLE  
2023 , THAT THE RECEIVER FAILED TO MATCH  
2024 , THE SYNC CHARACTER  
2025 007142 017701 172553 MOV @RXDBUF,R1 , SAVE ACTUAL  
2026 007146 010400 MOV R4,R0 , SAVE EXPECTED  
2027 007150 042700 177400 BIC #177400,R0 , CLR UPPER BYTE  
2028 007154 020001 CMP R0,R1 , DO THEY COMPARE ?  
2029 007156 001401 BEQ +4  
2030 007160 104002 ERROR 2 , IF REACT FAILED ALONG WITH THIS  
2031 , IT PROBABLY IS A TRANSMITTER ERROR  
2032 , HOWEVER, IF ONLY THIS FAILED IT  
2033 , PROBABLY IS A RECEIVER ERROR  
2034 007162 104405 SCOPE1  
2035 , POKE CLK TO SEE DNA DNA COMES UP ON THE FIRST  
2036 , BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO  
2037 , TXDBUF  
2038 007164 052777 020000 172534 BIS #CLK,@TXCSR , POKE CLK UP  
2039 007172 005777 172530 TST @TXCSR , DNA?  
2040 007176 100401 BMI +4  
2041 007200 104004 ERROR 4 , DNA DID NOT ASSERT  
2042 , SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH  
2043 007202 052777 000400 172516 BIS #MRESET,@TXCSR , MASTER RESET  
2044 007210 032777 000020 172474 BIT #SYNSCH,@RXCSR , SYNC SEARCH = 0 ?  
2045 007216 001401 BEQ +4  
2046 007220 104004 ERROR 4 , SYNC SEARCH SHOULD BE NOT SET  
2047 007222 005204 INC R4  
2048 007224 122704 000040 CMPB #40,R4 , IS THIS THE LAST CHARACTER ?  
2049 007230 001246 BNE 6$ , NO  
2050  
2051 , , THIS TEST VERIFYS MATCH DETECT & DATA RDY  
2052 , , FLAGS FOR EVERY POSSIBLE MATCH CHARACTER  
2053 , , BY OBSERVING REACT BIT  
2054 , , IT WILL TAKE TWO SYNC * CHARACTERS TO GET REACT BIT  
2055 , , * DEPENDENT ON MONITOR  
2056 , , IF ONE SYNC STRAP IS SELECTED , IT WILL  
2057 , , ONLY TAKE ONE SYNC CHARACTER BEFORE REACT TO  
2058 , , ASSERT  
2059 , , MODE SYNC INTERNAL  
2060 , , LENGTH SIX  
2061 , , SYNC CHARACTER FOR MATCH B/C  
2062 , , THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS  
2063 , ,  
2064 , , *****  
2065 007232 000004 TST13 SCOPE  
2066 007234 052777 000400 172464 BIS #MRESET,@TXCSR , MASTER RESET  
2067 007242 016703 172450 MOV RXDBUF,R3 , SET UP FOR ERROR MESSAGE  
2068 , SET SYNC INTERNAL, SIX, NO PARITY, 0 SYNC REGISTER  
2069 007246 012704 032000 MOV #SYNINT'SIX'NOPAR,R4 , CREATE PARAMETERS  
2070 007252 012777 004020 172446 6$ MOV #MINT'SEND,@TXCSR , SET SEND & MAINT INTERP  
2071 007260 010477 172436 MOV R4,@PARCSR , LOAD CSR
```

INITIALIZE THE COMMON TAGS

```

2072 007264 052777 000020 172420      BIS      #SYNSCH,@RXCSR  ,SET SYNC SEARCH
2073      ,POKE CLK TO GET INTO SYNCHRONIZATION
2074      ,BOTH THE LOGIC & RECEIVER
2075 007272 052777 020000 172426      BIS      #CLK,@TXCSR   ;POKE CLK UP
2076 007300 042777 020000 172420      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2077 007306 110477 172420      MOVB     R4,@TXDBUF    ;LOAD DATA CHARACTER
2078      ,POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZAT ON
2079 007312 052777 020000 172406      BIS      #CLK,@TXCSR   ;POKE CLK UP
2080 007320 042777 020000 172400      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2081 007326 032777 004000 172356      BIT      #REACT,@RXCSR ;REACT ?
2082 007334 001401      BEQ      +4
2083 007336 104004      ERROR    4           ,REACT SHOULD NOT BE SET
2084 007340 000404      BR       4$
2085 007342 010477 172354      5$      MOV      R4,@PARCSR    ,LOAD PARCSR WITH PARAMETERS
2086 007346 110477 172360      MOVB     R4,@TXDBUF    ,LOAD SYNC CHAR
2087 007352 012767 000002 171544  4$      MOV      #2,COUNT      ,# OF SYNC CHARS
2088 007360 005777 172342      2$      TST      @TXCSR ,DNA ?
2089 007364 100001      BPL      +4           ,BR IF NOT SET
2090 007366 104004      ERROR    4           ,DNA SHOULD NOT BE SET OR
2091      ,IT SHOULD BE CLEARED FROM PREVIOUS READ
2092 007370 012767 000006 171524      MOV      #6,SHIFT      ,# OF SHIFTS
2093      1$
2094 007376 052777 020000 172322      BIS      #CLK,@TXCSR   ,POKE CLK UP
2095 007404 042777 020000 172314      BIC      #CLK,@TXCSR   ,POKE CLK DOWN
2096 007412 005367 171504      DEC      SHIFT ,# OF SHIFTS
2097 007416 001367      BNE      1$
2098 007420 005367 171500      DEC      COUNT ,# OF SYNC CHARS
2099 007424 001403      BEQ      3$
2100      TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2101 007426 105767 171514      TSTB     SYNCNO
2102 007432 100752      BMI      2$           ,TWO SYNC CHARACTERS
2103 007434 032777 004000 172250  3$      BIT      #REACT,@RXCSR ,REACT ?
2104 007442 001001      BNE      +4
2105 007444 104004      ERROR    4           ,REACT FAILED TO SET,POSSIBLE
2106      ,THAT THE RECEIVER FAILED TO MATCH
2107      ,THE SYNC CHARACTER
2108 007446 017701 172244      MOV      @RXDBUF,R1    ,SAVE ACTUAL
2109 007452 010400      MOV      R4,R0         ,SAVE EXPECTED
2110 007454 042700 177400      BIC      #177400,R0    ,CLR UPPER BYTE
2111 007460 020001      CMP      R0,R1         ,DO THEY COMPARE ?
2112 007462 001401      BEQ      +4
2113 007464 104002      ERROR    2           ,IF REACT FAILED ALONG WITH THIS
2114      ,IT PROBABLY IS A TRANSMITTER ERROR
2115      ,HOWEVER, IF ONLY THIS FAILED IT
2116      ,PROBABLY IS A RECEIVER ERROR
2117 007466 104405      SCOP1
2118      ,POKE CLK TO SEE DNA DNA COMES UP ON THE FIRST
2119      ,BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2120      ,TXDBUF
2121 007470 052777 020000 172230      BIS      #CLK,@TXCSR   ,POKE CLK UP
2122 007476 005777 172224      TST      @TXCSR ,DNA?
2123 007502 100401      BMI      +4
2124 007504 104004      ERROR    4           ,DNA DID NOT ASSERT
2125      ,SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2126 007506 052777 000400 172212      BIS      #MRESET,@TXCSR ,MASTER RESET
2127 007514 032777 000020 172170      BIT      #SYNSCH,@RXCSR ,SYNC SEARCH = 0 ?
  
```

INITIALIZE THE COMMON TAGS

```

2128 007522 001401 BEQ +4
2129 007524 104004 ERROR 4 , SYNC SEARCH SHOULD BE NOT SET
2130 007526 005204 INC R4
2131 007530 122704 000100 CMPB #100,R4 , IS THIS THE LAST CHARACTER ?
2132 007534 001246 BNE 65 , NO
2133
2134 ,, THIS TEST VERIFYS MATCH DETECT & DATA RDY
2135 ,, FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2136 ,, BY OBSERVING RECACT BIT
2137 ,, IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2138 ,, * DEPENDENT ON MONITOR
2139 ,, IF ONE SYNC STRAP IS SELECTED , IT WILL
2140 ,, ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2141 ,, ASSERT
2142 ,, MODE SYNC INTERNAL
2143 ,, LENGTH SEVEN
2144 ,, SYNC CHARACTER FOR MATCH B/C
2145 ,, THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2146 ,,
2147 ,, *****
2148 007536 000004 TST14 SCOPE
2149 007540 052777 000400 172160 BIS #MRESET,@TXCSR , MASTER RESET
2150 007546 016703 172144 MOV RXDBUF,R3 , SET UP FOR ERROR MESSAGE
2151 , SET SYNC INTERNAL, SEVEN, NO PARITY, 0 SYNC REGISTER
2152 007552 012704 034000 MOV #SYNINT!SEVEN!NOPAR,R4 , CREATE PARAMETERS
2153 007556 012777 004020 172142 65 MOV #MINT!SEND,@TXCSR , SET SEND & MAINT NTER
2154 007564 010477 172132 MOV R4,@PARCSR , LOAD CSR
2155 007570 052777 000020 172114 BIS #SYNSCH,@RXCSR , SET SYNC SEARCH
2156 , POKE CLK TO GET INTO SYNCRONIZATION
2157 , BOTH THE LOGIC & RECEIVER
2158 007576 052777 020000 172122 BIS #CLK,@TXCSR , POKE CLK UP
2159 007604 042777 020000 172114 BIC #CLK,@TXCSP , POKE CLK DOWN
2160 007612 110477 172114 MOVB R4,@TXDBUF , LOAD DATA CHAPACTER
2161 , POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZAT ON
2162 007616 052777 020000 172102 BIS #CLK,@TXCSR , POKE CLK UP
2163 007624 042777 020000 172074 BIC #CLK,@TXCSR , POKE CLK DOWN
2164 007632 032777 004000 172052 BIT #RECACT,@RXCSR , RECACT ?
2165 007640 001401 BEQ +4
2166 007642 104004 ERROR 4 , RECACT SHOULD NOT BE SET
2167 007644 000404 BR 45
2168 007646 010477 172050 55 MOV R4,@PARCSR , LOAD PARCSR WITH PARAMETERS
2169 007652 110477 172054 MOVB R4,@TXDBUF , LOAD SYNC CHAR
2170 007656 012767 000002 171240 45 MOV #2,COUNT , # OF SYNC CHARS
2171 007664 005777 172036 25 TST @TXCSR , DNA ?
2172 007670 100001 BPL +4 , BR IF NOT SET
2173 007672 104004 ERROR 4 , DNA SHOULD NOT BE SET OR
2174 , IT SHOULD BE CLEARED FROM PREVIOUS READ
2175 007674 012767 000007 171220 MOV #7,SHIFT , # OF SHIFTS
2176 007702 15
2177 007702 052777 020000 172016 BIS #CLK,@TXCSR , POKE CLK UP
2178 007710 042777 020000 172010 BIC #CLK,@TXCSR , POKE CLK DOWN
2179 007716 005367 171200 DEC SHIFT , # OF SHIFTS
2180 007722 001367 BNE 15
2181 007724 005367 171174 DEC COUNT , # OF SYNC CHARS
2182 007730 001403 BEQ 35
2183 , TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
  
```

```

2184 007732 105767 171210 TSTB SYNCNO
2185 007736 100752 BMI 25 , TWO SYNC CHARACTERS
2186 007740 032777 004000 171744 35 BIT #RECACT,@RXCSR ,RECACT ?
2187 007746 001001 BNE +4
2188 007750 104004 ERROR 4 , RECACT FAILED TO SET, POSSIBLE
2189 , THAT THE RECEIVER FAILED TO MATCH
2190 , THE SYNC CHARACTER
2191 007752 017701 171740 MOV @RXDBUF,R1 , SAVE ACTUAL
2192 007756 010400 MOV R4,R0 , SAVE EXPECTED
2193 007760 042700 177403 BIC #177400,R0 , CLR UPPER BYTE
2194 007764 020001 CMP R0,R1 , DO THEY COMPARE ?
2195 007766 001401 BEQ +4
2196 007770 104002 ERROR 2 , IF RECACT FAILED ALONG WITH THIS
2197 , IT PROBABLY IS A TRANSMITTER ERROR
2198 , HOWEVER, IF ONLY THIS FAILED IT
2199 , PROBABLY IS A RECEIVER ERROR
2200 007772 104405 SCOPE1
2201 , POKE CLK TO SEE DNA DNA COMES UP ON THE FIRST
2202 , BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2203 , TXDBUF
2204 007774 052777 020000 171724 B S #CLK,@TXCSR , POKE CLK UP
2205 010002 005777 171720 TST @TXCSR ,DNA?
2206 010006 100401 BMI +4
2207 010010 104004 ERROR 4 , DNA DID NOT ASSERT
2208 , SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2209 010012 052777 000400 171706 BIS #MRESET,@TXCSR , MASTER RESET
2210 010020 032777 000020 171664 BIT #SYNSCH,@RXCSR , SYNC SEARCH = 0 ?
2211 010026 001401 BEQ +4
2212 010030 104004 ERROR 4 , SYNC SEARCH SHOULD BE NOT SET
2213 010032 005204 INC R4
2214 010034 122704 000200 CMPB #200,R4 , IS THIS THE LAST CHARACTER ?
2215 010040 001246 BNE 65 , NO
2216
2217 , THIS TEST VERIFYS MATCH DETECT & DATA RDY
2218 , FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2219 , BY OBSERVING RECACT BIT
2220 , IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2221 , * DEPENDENT ON MONITOR
2222 , IF ONE SYNC STRAP IS SELECTED , IT WILL
2223 , ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2224 , ASSERT
2225 , MODE SYNC INTERNAL
2226 , LENGTH EIGHT
2227 , SYNC CHARACTER FOR MATCH B/C
2228 , THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2229
2230 , *****
2231 010042 000004 TST15 SCOPE
2232 010044 052777 000400 171654 BIS #MRESET,@TXCSR , MASTER RESET
2233 010052 016703 171640 MOV RXDBUF,R3 , SET UP FOR ERROR MESSAGE
2234 , SET SYNC INTERNAL,EIGHT,NO PARITY,0 SYNC REGISTER
2235 010056 012704 036000 MOV #SYNINT'EIGHT'NOPAR,R4 , CREATE PARAMETERS
2236 010062 012777 004020 171636 65 MOV #MINT'SEND,@TXCSR , SET SEND & MAINT INTER
2237 010070 010477 171626 MOV R4,@PARCSR , LOAD CSR
2238 010074 052777 000020 171610 BIS #SYNSCH,@RXCSR , SET SYNC SEARCH
2239 , POKE CLK TO GET INTO SYNCRONIZATION
  
```

INITIALIZE THE COMMON TAGS

```

2240      , BOTH THE LOGIC & RECEIVER
2241 010102 052777 020000 171616      BIS      #CLK, @TXCSR      , POKE CLK UP
2242 010110 042777 020000 171610      BIC      #CLK, @TXCSR      ; POKE CLK DOWN
2243 010116 110477 171610      MOV      R4, @TXDBUF      , LOAD DATA CHARACTER
2244      , POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRON ZATION
2245 010122 052777 020000 171576      BIS      #CLK, @TXCSR      ; POKE CLK UP
2246 010130 042777 020000 171570      BIC      #CLK, @TXCSR      , POKE CLK DOWN
2247 010136 032777 004000 171546      BIT      #REACT, @RXCSR      , REACT ?
2248 010144 001401      BEQ      +4
2249 010146 104004      ERROR    4      , REACT SHOULD NOT BE SET
2250 010150 000404      BR      4$
2251 010152 010477 171544      5$      MOV      R4, @PARCSR      , LOAD PARCSR WITH PARAMETERS
2252 010156 110477 171550      MOV      R4, @TXDBUF      , LOAD SYNC CHAR
2253 010162 012767 000002 170734 4$      MOV      #2, COUNT      , # OF SYNC CHARS
2254 010170 005777 171532      2$      TST      @TXCSR      , DNA ?
2255 010174 100001      BPL      +4      , BR IF NOT SET
2256 010176 104004      ERROR    4      , DNA SHOULD NOT BE SET OR
2257      , IT SHOULD BE CLEARED FROM PREVIOUS READ
2258 010200 012767 000010 170714      MOV      #8, SHIFT      , # OF SHIFTS
2259 010206      1$
2260 010206 052777 020000 171512      BIS      #CLK, @TXCSR      , POKE CLK UP
2261 010214 042777 020000 171504      BIC      #CLK, @TXCSR      , POKE CLK DOWN
2262 010222 005367 170674      DEC      SHIFT      , # OF SHIFTS
2263 010226 001367      BNE      1$
2264 010230 005367 170670      DEC      COUNT      , # OF SYNC CHARS
2265 010234 001403      BEQ      3$
2266      , TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2267 010236 105767 170704      TST      SYNCNO
2268 010242 100752      BMI      2$      , TWO SYNC CHARACTERS
2269 010244 032777 004000 171440 3$      BIT      #REACT, @RXCSR      , REACT ?
2270 010252 001001      BNE      +4
2271 010254 104004      ERROR    4      , REACT FAILED TO SET, POSSIBLE
2272      , THAT THE RECEIVER FAILED TO MATCH
2273      , THE SYNC CHARACTER
2274 010256 017701 171434      MOV      @RXDBUF, R1      , SAVE ACTUAL
2275 010262 010400      MOV      R4, R0      , SAVE EXPECTED
2276 010264 042700 177400      BIC      #177400, R0      , CLR UPPER BYTE
2277 010270 020001      CMP      R0, R1      , DO THEY COMPARE ?
2278 010272 001401      BEQ      +4
2279 010274 104002      ERROR    2      , IF REACT FAILED ALONG WITH THIS
2280      , IT PROBABLY IS A TRANSMITTER ERROR
2281      , HOWEVER, IF ONLY THIS FAILED IT
2282      , PROBABLY IS A RECEIVER ERROR
2283 010276 104405      SCOP1
2284      , POKE CLK TO SEE DNA      DNA COMES UP ON THE FIRST
2285      , BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2286      , TXDBUF
2287 010300 052777 020000 171420      BIS      #CLK, @TXCSR      , POKE CLK UP
2288 010306 005777 171414      TST      @TXCSR      , DNA?
2289 010312 100401      BMI      +4
2290 010314 104004      ERROR    4      , DNA DID NOT ASSERT
2291      , SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2292 010316 052777 000400 171402      BIS      #MRESET, @TXCSR      , MASTER RESET
2293 010324 032777 000020 171360      BIT      #SYNSCH, @RXCSR      , SYNC SEARCH = 0 ?
2294 010332 001401      BEQ      +4
2295 010334 104004      ERROR    4      , SYNC SEARCH SHOULD BE NOT SET
  
```

INITIALIZE THE COMMON TAGS

```

2296 010336 005204          INC      R4
2297 010340 122704 000000  CMPB    #0,R4 , IS THIS THE LAST CHARACTER ?
2298 010344 001246          BNE     65    , NO
2299
2300          .. THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2301          .. BOTH THE TRANSMITTER AND RECEIVER LOGIC
2302          .. MODE SYNC EXTERNAL (SYNEXT)
2303          .. LENGTH EIGHT PLUS PARITY
2304          .. PARITY EVEPAR
2305          .. MAINT MODE MINT
2306
2307          .. *****
2308 010346 000004          TST16   SCOPE
2309 010350 052777 000400 171350  BIS     #MRESET,@TXCSR , MASTER RESET
2310 010356 012777 020000 171336  MOV     #SYNEXT,@PARCSR , SET THE MODE
2311 010364 052777 000400 171334  BIS     #MRESET,@TXCSR , MASTER RESET
2312
2313          .SET MAINTENANCE MODE & SEND
2314          .NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2315 010372 012777 004020 171326  MOV     #MINT'SEND,@TXCSR
2316
2317          SET MODE # OF BITS, PARITY SENSE, & LOAD SYNC REG
2318 010400 012777 027426 171314  MOV     #SYNEXT'EIGHT'EVEPAR'26,@PARCSR
2319 010406 016703 171304          MOV     RXDBUF,R3 , SETUP FOR ERROR MSG
2320 010412 005004          CLR     R4 , FOR DATA CHAR CREATION
2321 010414 110477 171312          MOVB   R4,@TXDBUF , LOAD CHARACTER
2322 010420 052777 000020 171264  BIS     #SYNSCH,@RXCSR , SET SEARCH SYNC
2323          .GET INTO SYNCHRONIZATION
2324 010426 052777 020000 171272  BIC     #CLK,@TXCSR , POKE CLK UP
2325 010434 042777 020000 171264  BIC     #CLK,@TXCSR , POKE CLK DOWN
2326 010442 012767 000011 170452 15  MOV     #9,SHIFT , # OF SHIFTS
2327 010450 010400          MOV     R4,R0 , EXPECTED
2328 010452          25
2329 010452 052777 020000 171246  BIS     #CLK,@TXCSR , POKE CLK UP
2330 010460 042777 020000 171240  BIC     #CLK,@TXCSR , POKE CLK DOWN
2331 010466 005367 170430          DEC     SHIFT , # OF SHIFTS
2332 010472 022767 000003 170422  CMP     #3,SHIFT , TIME TO LOAD NEXT CHAR ?
2333 010500 001003          BNE     35    , NO ?
2334 010502 005204          INC     R4 , GENERATE NEXT CHAR
2335 010504 110477 171222          MOVB   R4,@TXDBUF , LOAD NEXT CHARACTER
2336 010510 005767 170406          35  TST     SHIFT , IS IT 0 ?
2337 010514 001356          BNE     25
2338 010516 105777 171170          TSTB   @RXCSR , RXDONE = 1 ?
2339 010522 100401          BMI     +4
2340 010524 104004          ERROR  4 , RXDONE SHOULD BE SET
2341 010526 017701 171164          MOV     @RXDBUF,R1 , ACTUAL
2342 010532 020001          CMP     R0,R1 , COMPARE EXP VS ACT
2343 010534 001401          BEQ     +4
2344 010536 104002          ERROR  2 , CHARACTERS SHOULD COMPARE
2345 010540 105704          TSTB   R4 , LAST CHARACTER ?
2346 010542 001337          BNE     15    , NO
2347 010544          45
2348          .. THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2349          .. BOTH THE TRANSMITTER AND RECEIVER LOGIC
2350          .. MODE SYNC EXTERNAL (SYNEXT)
2351          .. LENGTH EIGHT PLUS PARITY
  
```



```

2352          ,PARITY ODDPAR
2353          ,MAINT MODE MINT
2354          ,
2355          ,*****
2356 010544 000004 TST17 SCOPE
2357 010546 052777 000400 171152 BIS #MRESET,@TXCSR ,MASTER RESET
2358 010554 012777 020000 171140 MOV #SYNEXT,@PARCSR ,SET THE MODE
2359 010562 052777 000400 171136 BIS #MRESET,@TXCSR ,MASTER RESET
2360
2361          ,SET MAINTENANCE MODE & SEND
2362          ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2363 010570 012777 004020 171130 MOV #MINT'SEND,@TXCSR
2364
2365          ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2366 010576 012777 027026 171116 MOV #SYNEXT'EIGHT'ODDPAR'26,@PARCSR
2367 010604 016703 171106 MOV RXDBUF,R3 ,SETUP FOR ERROR MSG
2368 010610 005004 CLR R4 ;FOR DATA CHAR CREATION
2369 010612 110477 171114 MOVB R4,@TXDBUF ,LOAD CHARACTER
2370 010616 052777 000020 171066 BIS #SYNSCH,@RXCSR ,SET SEARCH SYNC
2371          ,GET INTO SYNCHRONIZATION
2372 010624 052777 020000 171074 BIS #CLK,@TXCSR ,POKE CLK UP
2373 010632 042777 020000 171066 BIC #CLK,@TXCSR ,POKE CLK DOWN
2374 010640 012767 000011 170254 1$ MOV #9,SHIFT ,# OF SHIFTS
2375 010646 010400 MOV R4,R0 ,EXPECTED
2376 010650 2$
2377 010650 052777 020000 171050 BIS #CLK,@TXCSR ,POKE CLK UP
2378 010656 042777 020000 171042 BIC #CLK,@TXCSR ,POKE CLK DOWN
2379 010664 005367 170232 DEC SHIFT ,# OF SHIFTS
2380 010670 022767 000003 170224 CMP #3,SHIFT ,TIME TO LOAD NEXT CHAR ?
2381 010676 001003 BNE 3$ ,NO ?
2382 010700 005204 INC R4 ,GENERATE NEXT CHAR
2383 010702 110477 171024 MOVB R4,@TXDBUF ,LOAD NEXT CHARACTER
2384 010706 005767 170210 3$ TST SHIFT ,IS IT 0 ?
2385 010712 001356 BNE 2$
2386 010714 105777 170772 TSTB @RXCSR ,RXDONE = 1 ?
2387 010720 100401 BMI +4
2388 010722 104004 ERROR 4 ,RXDONE SHOULD BE SET
2389 010724 017701 170766 MOV @RXDBUF,R1 ,ACTUAL
2390 010730 020001 CMP R0,R1 ,COMPARE EXP VS ACT
2391 010732 001401 BEQ +4
2392 010734 104002 ERROR 2 ,CHARACTERS SHOULD COMPARE
2393 010736 105704 TSTB R4 ,LAST CHARACTER ?
2394 010740 001337 BNE 1$ ,NO
2395 010742 4$
2396          ,THIS TEST PERFORMS A BINARY COUNT DATA PATTERN OF
2397          ,BOTH THE TRANSMITTER AND RECEIVER LOGIC
2398          ,MODE SYNC EXTERNAL (SYNEXT)
2399          ,LENGTH EIGHT PLUS PARITY
2400          ,PARITY EVEPAR
2401          ,MAINT MODE MEXT
2402          ,
2403          ,*****
2404 010742 000004 TST20 SCOPE
2405 010744 105767 170203 TSTB JMRBY ,JUMP AROUND TEST ?
2406 010750 100116 BPL 4$ ,YES ?
2407 010752 052777 000400 170746 BIS #MRESET,@TXCSR ,MASTER RESET
  
```

```

2408 010760 012777 020000 170734      MOV    #SYNEXT,@PARCSR ,SET THE MODE
2409 010766 052777 000400 170732      BIS    #MRESET,@TXCSR ,MASTER RESET
2410
2411      ,SET MAINTENANCE MODE & SEND
2412      ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2413 010774 012777 010020 170724      MOV    #NEXT!SEND,@TXCSR
2414
2415      ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2416 011002 012777 027426 170712      MOV    #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
2417 011010 016703 170702      MOV    RXDBUF,R3      ;SETUP FOR ERROR MSG
2418 011014 005004      CLR    R4      ;FOR DATA CHAR CREATION
2419 011016 110477 170710      MOVB  R4,@TXDBUF      ,LOAD CHARACTER
2420 011022 052777 000020 170662      BIS    #SYNSCH,@RXCSR ,SET SEARCH SYNC
2421      ,GET INTO SYNCHRONIZATION
2422 011030 052777 020000 170670      BIS    #CLK,@TXCSR      ,POKE CLK UP
2423      ,WAIT FOR CABLE & DRIVER DELAYS
2424 011036 016702 170056      MOV    HOLD,R2 ,WAIT THIS AMT
2425 011042 005302      DEC    R2      ,WAIT
2426 011044 001376      BNE    2
2427      ,EXIT
2428 011046 042777 020000 170652      BIC    #CLK,@TXCSR      ,POKE CLK DOWN
2429      ,WAIT FOR CABLE & DRIVER DELAYS
2430 011054 016702 170040      MOV    HOLD,R2 ,WAIT THIS AMT
2431 011060 005302      DEC    R2      ,WAIT
2432 011062 001376      BNE    -2
2433      ,EXIT
2434 011064 012767 000011 170030 15      MOV    #9 ,SHIFT      ,# OF SHIFTS
2435 011072 010400      MOV    R4,R0      ,EXPECTED
2436 011074      25
2437 011074 052777 020000 170624      BIS    #CLK,@TXCSR      ,POKE CLK UP
2438      ,WAIT FOR CABLE & DRIVER DELAYS
2439 011102 016702 170012      MOV    HOLD,R2 ,WAIT THIS AMT
2440 011106 005302      DEC    R2      ,WAIT
2441 011110 001376      BNE    -2
2442      ,EXIT
2443 011112 042777 020000 170606      BIC    #CLK,@TXCSR      ,POKE CLK DOWN
2444      ,WAIT FOR CABLE & DRIVER DELAYS
2445 011120 016702 167774      MOV    HOLD,R2 ,WAIT THIS AMT
2446 011124 005302      DEC    R2      ,WAIT
2447 011126 001376      BNE    -2
2448      ,EXIT
2449 011130 005367 167766      DEC    SHIFT      ,# OF SHIFTS
2450 011134 022767 000003 167760      CMP    #3,SHIFT      ,TIME TO LOAD NEXT CHAR ?
2451 011142 001003      BNE    3$      ,NO ?
2452 011144 005204      INC    R4      ,GENERATE NEXT CHAR
2453 011146 110477 170560      MOVB  R4,@TXDBUF      ,LOAD NEXT CHARACTER
2454 011152 005767 167744      TST    SHIFT      ,IS IT 0 ?
2455 011156 001346      BNE    2$
2456 011160 105777 170526      TSTB  @RXCSR ,RXDONE = 1 ?
2457 011164 100401      BMI    +4
2458 011166 104004      ERROR  4      ,RXDONE SHOULD BE SET
2459 011170 017701 170522      MOV    @RXDBUF,R1      ,ACTUAL
2460 011174 020001      CMP    R0,R1      ,COMPARE EXP VS ACT
2461 011176 001401      BEQ    +4
2462 011200 104002      ERROR  2      ,CHARACTERS SHOULD COMPARE
2463      ,CHECK OUT MODEM BYPASS JUMPER
  
```

INITIALIZE THE COMMON TAGS

2464	011202	105704			TSTB	R4	, LAST CHARACTER ?
2465	011204	001327			BNE	15	, NO
2466	011206			45			
2467							.. THIS TEST PERFORMS A BINARY COUNT DATA PATTEPN ON
2468							.. BOTH THE TRANSMITTER AND RECEIVER LOGIC
2469							.. MODE SYNC EXTERNAL (SYNEXT)
2470							.. LENGTH EIGHT PLUS PARITY
2471							.. PARITY ODDPAR
2472							.. MAINT MODE MEXT
2473							..
2474							.. *****
2475	011206	000004			TST21	SCOPE	
2476	011210	105767	167737		TSTB	JMPBY	, JUMP AROUND TEST ?
2477	011214	100116			BPL	45	, YES ?
2478	011216	052777	000400	170502	BIS	#MRESET, @TXCSR	, MASTER RESET
2479	011224	012777	020000	170470	MOV	#SYNEXT, @PARCSR	, SET THE MODE
2480	011232	052777	000400	170466	BIS	#MRESET, @TXCSR	, MASTER RESET
2481							
2482							. SET MAINTENANCE MODE & SEND
2483							. NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2484	011240	012777	010020	170460	MOV	#MEXTISEND, @TXCSR	
2485							
2486							. SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2487	011246	012777	027026	170446	MOV	#SYNEXT'EIGHT'ODDPAR'26, @PARCSR	
2488	011254	016703	170436		MOV	RXDBUF, R3	; SETUP FOR ERROR MSG
2489	011260	005004			CLR	R4	; FOR DATA CHAR CREATION
2490	011262	110477	170444		MOVB	R4, @TXDBUF	, LOAD CHARACTER
2491	011266	052777	000020	170416	BIS	#SYNSCH, @RXCSR	, SET SEARCH SYNC
2492							. GET INTO SYNCHRONIZATION
2493	011274	052777	020000	170424	BIS	#CLK, @TXCSR	, POKE CLK UP
2494							. WAIT FOR CABLE & DRIVER DELAYS
2495	011302	016702	167612		MOV	HOLD, R2	, WAIT THIS AMT
2496	011306	005302			DEC	R2	, WAIT
2497	011310	001376			BNE	-2	
2498							. EXIT
2499	011312	042777	020000	170406	BIC	#CLK, @TXCSR	, POKE CLK DOWN
2500							. WAIT FOR CABLE & DRIVER DELAYS
2501	011320	016702	167574		MOV	HOLD, R2	, WAIT THIS AMT
2502	011324	005302			DEC	R2	, WAIT
2503	011326	001376			BNE	-2	
2504							. EXIT
2505	011330	012767	000011	167564	15	MOV	#9, SHIFT
2506	011336	010400			MOV	R4, R0	, EXPECTED
2507	011340				25		
2508	011340	052777	020000	170360	BIS	#CLK, @TXCSR	, POKE CLK UP
2509							. WAIT FOR CABLE & DRIVER DELAYS
2510	011346	016702	167546		MOV	HOLD, R2	, WAIT THIS AMT
2511	011352	005302			DEC	R2	, WAIT
2512	011354	001376			BNE	-2	
2513							. EXIT
2514	011356	042777	020000	170342	BIC	#CLK, @TXCSR	, POKE CLK DOWN
2515							. WAIT FOR CABLE & DRIVER DELAYS
2516	011364	016702	167530		MOV	HOLD, R2	, WAIT THIS AMT
2517	011370	005302			DEC	R2	, WAIT
2518	011372	001376			BNE	-2	
2519							. EXIT

INITIALIZE THE COMMON TAGS

```

2520 011374 005367 167522          DEC    SHIFT    ; # OF SHIFTS
2521 011400 022767 000003 167514  CMP    #3,SHIFT ; TIME TO LOAD NEXT CHAR ?
2522 011406 001003          BNE    3$      ; NO ?
2523 011410 005204          INC    R4      , GENERATE NEXT CHAR
2524 011412 110477 170314          MOVB   R4,@TXDBUF , LOAD NEXT CHARACTER
2525 011416 005767 167500          3$    TST    SHIFT    , IS IT 0 ?
2526 011422 001346          BNE    2$
2527 011424 105777 170262          TSTB  @RXCSR   , RXDONE = 1 ?
2528 011430 100401          BMI    +4
2529 011432 104004          ERROR  4      , RXDONE SHOULD BE SET
2530 011434 017701 170256          MOV    @RXDBUF,R1 , ACTUAL
2531 011440 020001          CMP    R0,R1   , COMPARE EXP VS ACT
2532 011442 001401          BEQ    +4
2533 011444 104002          ERROR  2      , CHARACTERS SHOULD COMPARE
2534          , CHECK OUT MODEM BYPASS JUMPER
2535 011446 105704          TSTB  R4      , LAST CHARACTER ?
2536 011450 001327          BNE    1$      , NO
2537 011452          4$
2538          , , THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2539          , , RECEIVER SECTION, IT USES THE ERROR FLAGS
2540          , , TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2541          , , (OVRRUN,RXERR)
2542          , , MODE ISYMOD
2543          , , LENGTH FIVE
2544          , , CHAR 12
2545          , ,
2546          , , *****
2547 011452 000004          TST22 SCOPE
2548 011454 052777 000400 170244          BIS    #MRESET,@TXCSR ; MASTER RESET
2549 011462 012777 000000 170232          MOV    #ISYMOD,@PARCSR , SET THE MODE
2550 011470 052777 000400 170230          BIS    #MRESET,@TXCSR , MASTER RESET
2551          , SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2552          MOV    #MTDATA'CLK'MINT'BREAK,@TXCSR
2553 011476 012777 064001 170222          , SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
2554          MOV    #ISYMOD'FIVE'NOPAR'0,@PARCSR
2555 011504 012777 000000 170210          BIS    #SYNSCH,@RXCSR , SET SYNC SEARCH
2556 011512 052777 000020 170172          ; POKE CLK TO GET RECEIVER INTO SYNCRIZATION
2557          BIC    #CLK,@TXCSR , POKE CLK DOWN
2558          BIS    #CLK,@TXCSR , POKE CLK UP
2559 011520 042777 020000 170200          , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2560 011526 052777 020000 170172          BIC    #CLK,@TXCSR , POKE CLK DOWN
2561          BIS    #CLK,@TXCSR , POKE CLK UP
2562 011534 042777 020000 170164          MOV    RXDBUF,R3 , SET UP FOR ERROR MESSAGE
2563 011542 052777 020000 170156          MOV    #12,R0 , EXPECTED
2564 011550 016703 170142          MOV    #7,SHIFT , # OF SHIFTS
2565 011554 012700 000012          MOV    #124,$TMP1 , DATA CHAR
2566 011560 012767 000007 167334          JSR    PC,RPOKE , SHIFT 'N THIS CHAR
2567 011566 012767 000124 167704          TSTB  @RXCSR   , RXDONE ?
2568 011574 004767 005352          BMI    +4
2569 011600 105777 170106          ERROR  4      , RXDONE SHOULD BE SET
2570 011604 100401          MOV    @RXDBUF,R1 , ACTUAL
2571 011606 104004          CMP    R0,R1   , COMPARE EXPECTED VS ACTUAL
2572 011610 017701 170102          BEQ    +4
2573 011614 020001          ERROR  2      , RECEIVED DATA DID NOT MATCH
2574 011616 001401
2575 011620 104002

```


2688	012276	012767	000007	166616		MOV	#7,SHIFT	,# OF SHIFTS
2689	012304	012767	000100	167166		MOV	#100,\$TMP1	,DATA CHAR
2690	012312	004767	004634			JSR	PC,RPOKE	,SHIFT IN THIS CHAR
2691								,NOW SHIFT IN A SECONO CHARACTER WITHOUT READING RXDBUF
2692	012316	012767	000007	166576		MOV	#7,SHIFT	,# OF SHIFTS
2693	012324	012767	000100	167146		MOV	#100,\$TMP1	,DATA CHAR
2694	012332	004767	004614			JSR	PC,RPOKE	,SHIFT IN THIS CHAR
2695	012336	012700	140000			MOV	#140000'D,RO	,EXPECTED DATA PLUS
2696								,RXERR & OVRRUN
2697	012342	017701	167350			MOV	@RXDBUF,R1	,ACTUAL
2698	012346	020001				CMP	RO,R1	,COMPARE EXP VS ACT
2699	012350	001401				BEQ	+4	
2700	012352	104002				ERROR	2	,SPECIFICALLY LOOK AT RXERR & OVRRUN BITS THEY BOTH SHOULD BE SET
2701								
2702								
2703								
2704								,END OF PASS
2705								,TYPE NAME OF TEST
2706								,UPDATE PASS COUNT
2707								,CHECK FOR EXIT TO ACT-11
2708								,RESTART TEST
2709								
2710	012354	000004			EOP	SCOPE		
2711	012356	004767	000340			JSR	PC,CKSWR	
2712	012362	104401				TYPE		,TYPE NAME OF TEST
2713	012364	015512				MEPASS		
2714	012366	104413	012620			CONVRT	,OUTCRY	
2715	012372	104401	015331			TYPE	,DEVICE	
2716	012376	105767	166550			TSTB	MULTD	,ARE YOU RUNNING MULTIPLE DEVICES ?
2717	012402	001511				BEQ	CCC	,NO, JUMP AROUND
2718	012404	005767	166556			TST	ACTREG	,ARE ANY DEVICES ACTIVE ?
2719	012410	0010C7				BNE	RUNIT	,YES
2720	012412	104401	015343			TYPE	,MCOV	,NO
2721	012416	016700	166544			MOV	ACTREG,RO	,DISPLAY ACTREG
2722	012422	000000				HALT		,SELECT SOMETHING TO RUN @ ACTREG
2723								,SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2724	012424	000167	167522			JMP	START	,START OVER AGAIN. YOU DESELECTED EVERYTHING
2725	012430	062767	000010	166516	RUNIT	ADD	#10,BASEADD	,NEXT BLOCK (ADDRESSES)
2726	012436	062767	000010	166516	ZERO	ADD	#10,BASEIV	,NEXT BLOCK (VECTORS)
2727	012444	000241				CLC		
2728	012446	006167	166516			ROL	ROTADD	,UP DATE ROTATING POINTER
2729	012452	103410				BCS	25	,IS IT THE LAST DEVICE
2730								,TO BE TESTED IN THIS PASS ?
2731	012454	036767	166510	166504		BIT	ROTADD,ACTREG	,TEST THIS DEVICE FOR ACTIVE STATJS
2732	012462	001762				BEQ	RUNIT	,IF NOT ACTIVE, TRY NEXT ADDRESS
2733	012464	004767	000034			JSR	PC,REPLAY	,CALCULATE NEW PARAMETERS
2734	012470	000167	000210			JMP	RESTRT	,YES IT WAS ACTIVE,TEST THIS DEVICE
2735	012474	012767	000001	166466	25	MOV	#1,ROTADD	,OK!,NOW SET UP ROTATING
2736								,POINTER FOR NEXT MULTIPLE PASS
2737	012502	016767	166450	166444		MOV	KEEPADD,BASEADD	,RESTORE BASE ADDRESS
2738	012510	016767	166450	166444		MOV	KEEPIV,BASEIV	,RESTORE BASE INTERRUPT VECTORS
2739	012516	004767	000002			JSR	PC,REPLAY	,CALC NEW PARAMETERS
2740	012522	000441				BR	CCC	,JUMP AROUND REPLAY
2741	012524	016767	166424	004416	REPLAY	MOV	BASEADD,DUBASE	,SET UP FOR NEW ADDRESSES
2742	012532	004767	004260			JSR	PC,DUADDR	,CREATE NEW ADDRESSES
2743	012536	016767	166420	167172		MOV	BASEIV,DURIV	,CREATE DURIV

```

2744 012544 062767 000002 166410      ADD      #2, BASE IV
2745 012552 016767 166404 167160      MOV      BASE IV, DUR IS      , CREATE DUR IS
2746 012560 062767 000002 166374      ADD      #2, BASE IV
2747 012566 016767 166370 167146      MOV      BASE IV, DUT IV      , CREATE DUT IV
2748 012574 062767 000002 166360      ADD      #2, BASE IV
2749 012602 016767 166354 167134      MOV      BASE IV, DUT IS      , CREATE DUT IS
2750 012610 016767 167122 166344      MOV      DUR IV, BASE IV      , RESTORE
2751 012616 000207                                RTS      PC
2752
2753 012620 000001                                OUTCRY  1
2754 012622      006      002                                BYTE    6, 2
2755 012624 001712                                RXCSR
2756
2757 012626                                CCC
2758 012626 005067 166550      CLR      $TSTNM                , CLEAR TEST NUMBER
2759 012632 005067 166560      CLR      $ERRPC                , CLEAR LAST ERROR PC
2760 012636 005067 166541      CLR      $ERFLG                , CLEAR ERROR FLAG
2761 012642 005267 166244      INC      PASCNT                , UPDATE PASS COUNT
2762 012646 016767 166240 166226      MOV      PASCNT, LIGHTS        , DISPLAY PASS COUNT
2763 012654 016767 166232 166652      MOV      PASCNT, $PASS         , PASS COUNT TO APT
2764 012662 013701 000042      MOV      @#42, R1              , CHECK FOR ACT-11 OR DDP
2765 012666 001406                                BEQ      RESTRT                , IF NO CONTINUE TESTING
2766 012670 000005                                RESET
2767 012672 000005                                RESET
2768 012674 004711                                SENDAD  R      PC, (R1)
2769 012676 000240                                NOP
2770 012700 000240                                NOP
2771 012702 000240                                NOP
2772 012704                                RESTRT
2773 012704 012767 003376 166474      MOV      #TST1+2, $LPADR      LOAD LAST ADDR
2774 012712 004767 000004      JSR      PC, CKSWR
2775 012716 000167 170366      JMP      BEGIN
2776
2777                                , CHECK SWITCH REGISTER ROUTINE
2778                                , CHECKS TO ALLOW FOR < G > TO ALLOW
2779                                , THE CHANGING OF LOCATION 176
2780
2781 012722 005737 000042                                CKSWR  TST      @#42
2782 012726 001040                                BNE     OUT
2783 012730 022767 000176 166502      CMP     #SWREG, SWR           , SOFTWARE SWR PRESENT?
2784 012736 001034                                BNE     OUT                   , NO--LEAVE
2785 012740 105777 166500      TSTB   @$TKS                 , CHECK TTY READY
2786 012744 100031                                BPL     OUT                   , NO--LEAVE
2787 012746 017767 166474 000422      MOV     @$TKB, MSG           , GET CHARACTER
2788 012754 042767 177600 000414      BIC     #177600, MSG         , STRIP JUNK
2789 012762 122767 000007 000406      CMPB   #7, MSG              , IS IT < G > ?
2790 012770 001017                                BNE     OUT                   NO
2791 012772 104401 016117                                TYPE   , MCNTG
2792 012776 005137 013036                                CNTLU  COM     @#RDSW
2793 013002 104401 016127                                TYPE   , MMSWR
2794 013006 104413                                CONVRT
2795 013010 013040                                SWREGL
2796 013012 104406 016140                                INSTR, MMNEW
2797 013016 104410                                PARAM
2798 013020 000000                                O
2799 013022 177777                                177777

```


2800	013024	000176				SWREG			
2801	013026	000	001			BYTE	0,1		
2802	013030	005037	013036			OUT	CLR	#RDSW	
2803	013034	000207					RTS	PC	
2804	013036	000000				RDSW	WORD	0	
2805	013040	000001				SWREGL	1		
2806	013042	006	002				BYTE	6,2	
2807	013044	000176					SWREG		
2808									
2809	013046	000005					5		
2810									
2811									.CHECK FOR FREEZE ON CURRENT DATA
2812									
2813	013050	004767	177646			SCOP1	JSR	PC,CKSWR	
2814	013054	032777	001000	166356			BIT	#SW09,#SWR	
2815	013062	001402					BEQ	1\$	
2816	013064	016716	166020				MOV	LOCK,(SP)	
2817	013070	000002				1\$	RTI		
2818						SBTTL			TYPE ROUTINE
2819									
2820									*****
2821									*ROUTINE TO TYPE ASCIZ MESSAGE MESSAGE MUST TERMINATE WITH A 0 BYTE
2822									*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED
2823									*NOTE1 \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER
2824									*NOTE2 \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2825									*NOTE3 \$FILLC CONTAINS THE CHARACTER TO FILL AFTER
2826									*
2827									*CALL
2828									*1) USING A TRAP INSTRUCTION
2829									* TYPE ,MESADR ..MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2830									*OR
2831									* TYPE
2832									* MESADR
2833									*
2834									
2835	013072	105767	166361			STYPE	TSTB	\$TPFLG	.. IS THERE A TERMINAL?
2836	013076	100002					BPL	1\$.. BR IF YES
2837	013100	000000					HALT		.. HALT HERE IF NO TERMINAL
2838	013102	000430					BR	3\$.. LEAVE
2839	013104	010046				1\$	MOV	RO,-(SP)	.. SAVE RO
2840	013106	017600	000002				MOV	@2(SP),RO	.. GET ADDRESS OF ASCIZ STRING
2841	013112	122767	000001	166426			CMPB	#APTENV,\$ENV	.. RUNNING IN APT MODE
2842	013120	001011					BNE	62\$.. NO,GO CHECK FOR APT CONSOLE
2843	013122	132767	000100	166417			BITB	#APTPOOL,\$ENVM	.. SPOOL MESSAGE TO APT
2844	013130	001405					BEQ	62\$.. NO,GO CHECK FOR CONSOLE
2845	013132	010067	000004				MOV	RO,61\$.. SETUP MESSAGE ADDRESS FOR APT
2846	013136	004767	000006				JSR	PC,\$ATY3	.. SPOOL MESSAGE TO APT
2847	013142	000000				61\$	WORD	0	.. MESSAGE ADDRESS
2848	013144	132767	000040	166075		62\$	BITB	#APTCSUP,\$ENVM	.. APT CONSOLE SUPPRESSED
2849	013152	001003					BNE	60\$.. YES,SKIP TYPE OUT
2850	013154	112046				2\$	MOVB	(RO)+,-(SP)	.. PUSH CHARACTER TO BE TYPED ONTO STACK
2851	013156	001005					BNE	4\$.. BR IF IT ISN'T THE TERMINATOR
2852	013160	005726					TST	(SP)+	.. IF TERMINATOR POP IT OFF THE STACK
2853	013162	012600				60\$	MOV	(SP)+,RO	.. RESTORE RO
2854	013164	0+2716	000002			3\$	ADD	#2,(SP)	.. ADJUST RETURN PC
2855	013170	000002					RTI		.. RETURN

```

2856 013172 122716 000011      4$    CMPB    #HT, (SP)      ..BRANCH IF <HT>
2857 013176 001430              BEQ     8$
2858 013200 122716 000200      CMPB    #CRLF, (SP)    ..BRANCH IF NOT <CRLF>
2859 013204 001006              BNE     5$
2860 013206 005726              TST     (SP)+          ..POP <CR><LF> EQU U
2861 013210 104401              TYPE
2862 013212 001523              $CRLF
2863 013214 105067 000130      CLRBR   $CHARCNT      ..CLEAR CHARACTER COUNT
2864 013220 000755              BR      2$             ..GET NEXT CHARACTER
2865 013222 004767 000056      5$     JSR     PC, $TYPEC   ..GO TYPE THIS CHARACTER
2866 013226 126726 166224      6$     CMPB    $FILLC, (SP)+ ..IS IT TIME FOR FILLER CHARS ?
2867 013232 001350              BNE     2$             ..IF NO GO GET NEXT CHAR
2868 013234 016746 166214      MOV     $NULL, -(SP)   ..GET # OF FILLER CHARS NEEDED
2869                                ..AND THE NULL CHAR
2870 013240 105366 000001      7$     DECB    1(SP)         ..DOES A NULL NEED TO BE TYPED?
2871 013244 002770              BLT     6$             ..BR IF NO--GO POP THE NULL OFF OF STACK
2872 013246 004767 000032      JSR     PC, $TYPEC   ..GO TYPE A NULL
2873 013252 105367 000072      DECB    $CHARCNT      ..DO NOT COUNT AS A COUNT
2874 013256 000770              BR      7$             ..LOOP
2875
2876                                .HORIZONTAL TAB PROCESSOR
2877
2878 013260 112716 000040      8$     MOVB    #' , (SP)    ..REPLACE TAB WITH SPACE
2879 013264 004767 000014      9$     JSR     PC, $TYPEC   ..TYPE A SPACE
2880 013270 132767 000007 000052  BITB    #7, $CHARCNT  ..BRANCH IF NOT AT
2881 013276 001372              BNE     9$             ..TAB STOP
2882 013300 005726              TST     (SP)+          ..POP SPACE OFF STACK
2883 013302 000724              BR      2$             ..GET NEXT CHARACTER
2884 013304 105777 166140      $TYPEC TSTBR   @STPS   ..WAIT UNTIL PRINTER IS READY
2885 013310 100375              BPL     $TYPEC
2886 013312 116677 000002 166132  MOVB    2(SP), @STPB  ..LOAD CHAR TO BE TYPED INTO DATA REG
2887 013320 122766 000015 000002  CMPBR   #CR, 2(SP)    ..IS CHARACTER A CARRIAGE RETURN?
2888 013326 001003              BNE     1$             ..BRANCH IF NO
2889 013330 105067 000014      CLRBR   $CHARCNT      ..YES--CLEAR CHARACTER COUNT
2890 013334 000406              BR      $TYPEX        ..EXIT
2891 013336 122766 000012 000002  1$     CMPBR   #LF, 2(SP)    ..IS CHARACTER A LINE FEED?
2892 013344 001402              BEQ     $TYPEX        ..BRANCH IF YES
2893 013346 105227              INCB    (PC)+         ..COUNT THE CHARACTER
2894 013350 000000      $CHARCNT WORD 0      ..CHARACTER COUNT STOPAGE
2895 013352 000207      $TYPEX RTS    PC
2896
2897
2898                                .ASCII STRING INPUT ROUTINE
2899
2900 013354 017667 000000 000014  INSTR  MOV     @ (SP), MSG  .PICK UP MESSAGE
2901 013362 062716 000002              ADD     #2, (SP)      .JUMP AROUND MESSAGE FOR PT1
2902 013366 105767 166154              TSTBR   $ENV          .APT CONTROL
2903 013372 001036              BNE     INSTR2        .YES NO TYPE
2904 013374 104401              INSTR1 TYPE
2905 013376 000000      MSG     0
2906 013400 012704 016152      MOV     #INBUF, R4    .GET STARTING LOC OF INBUF
2907 013404 012703 000007      MOV     #7, R3        .MAX # OF CHARS
2908 013410 105777 166030      1$     TSTBR   @STKS , TTY FLAG
2909 013414 100375              BPL     1$
2910 013416 117714 166024      MOVB    @STKB, (R4)   .TAKE CHAR
2911 013422 142714 000200      BICBR   #200, (R4)   .STRIP

```

```

2912 013426 121427 000025      CMPB      (R4),#25      ,IS IT < G>
2913 013432 001760              BEQ        INST1
2914 013434 122427 000015      CMPB      (R4)+,#15    ,CHECK FOR CR
2915 013440 001413              BEQ        INSTR2
2916 013442 105777 166002      2$       TSTB      @STPS    ,TEST FLAG
2917 013446 100375              BPL        2$
2918 013450 117777 165772 165774      MOVB      @STKB,@STPB  ,ECHO CHARACTER
2919 013456 005303              DEC        R3        ,DID YOU TYPE TOO MANY CHARS ?
2920 013460 001353              BNE        1$
2921 013462 104401              INSTE     TYPE
2922 013464 015437              MQM       ,?
2923 013466 000742              BR        INST1    ,RETRY
2924 013470 000002      INSTR2    RTI
2925
2926              ,CONVERT ASCII STRING TO OCTAL
2927
2928 013472 011605      PARAM    MOV        (SP),R5    ,PUT CONTENTS OF SP INTO R5
2929 013474 012567 000162      MOV        (R5)+,LOLIM  ,PUT LOW LIMIT INTO LOLIM
2930 013500 012567 000160      MOV        (R5)+,HILIM  ,PUT HIGH LIMIT INTO HILIM
2931 013504 012567 000156      MOV        (R5)+,DEVADR  ,PUT STORE LOC INTO DEVADR
2932 013510 112567 000154      MOVB      (R5)+,LOBITS  ,PUT MASK INTO LOBITS
2933 013514 112567 000151      MOVB      (R5)+,ADRCNT  ,PUT COUNT INTO ADRCNT
2934 013520 010516      MOV        R5,(SP)    ,RESTORE RETURN ADDR ON STACK FOR RTI
2935 013522 005005      PARAM1   CLR        R5
2936 013524 012704 016152      MOV        #INBUF,R4
2937 013530 122714 000015      CMPB      #15,(R4)    ,CR ?
2938 013534 001420              BEQ        PARERR    ,YOU TYPED CR TOO SOON !
2939 013536 121427 000060      1$       CMPB      (R4),#60    ,LOW LIMIT ASCII 0
2940 013542 002415              BLT        PARERR
2941 013544 121427 000067      CMPB      (R4),#67    ,HIGH LIMIT ASCII ?
2942 013550 003012              BGT        PARERR
2943 013552 142714 000060      BICB      #60,(R4)    ,CONVERT TO OCTAL
2944 013556 152405              BISB      (R4)+,R5    ,STORE AWAY ITS AN OK CHAR
2945 013560 122714 000015      CMPB      #15,(R4)    ,CR ?
2946 013564 001414              BEQ        LIMITS    ,NOW CHECK FOR HIGH & LOW LIMIT CONDS
2947 013566 006305              ASL        R5        ,ALLOCATE ROOM FOR NEXT CHAR
2948 013570 006305              ASL        R5
2949 013572 006305              ASL        R5
2950 013574 000760              BR        1$
2951 013576 122714 000015      PARERR   CMPB      #15,(R4)    CR?
2952 013602 001003              BNE        120$
2953 013604 005737 013036      TST      @#RDSW    ,CK SWR USED
2954 013610 001023              BNE        PARTI
2955 013612 104407      120$     INSTER    ,RETRY
2956 013614 000742              BR        PARAM1
2957
2958              ,TEST TO SEE IF NUMBER IS WITHIN LIMITS
2959
2960 013616 020567 000042      LIMITS   CMP        R5,HILIM
2961 013622 101365              BHI        PARERR    ,THE # IS TOO HIGH
2962 013624 020567 000032      CMP        R5,LOLIM
2963 013630 103762              BLO        PARERR    ,THE # IS TOO LOW
2964 013632 136705 000032      BITB      LOBITS,R5  ,TEST BY MASKINGTHE #
2965 013636 001357              BNE        PARERR
2966
2967              ,STORE NUMBER AT SPECIFIED ADDRESS
  
```

```

2968
2969 013640 016704 000022          MOV    DEVADR,R4      ,GET STARTING ADDR OF
2970 013644 010524          1$    MOV    R5,(R4)+    ,STORE AT THIS ADDR
2971 013646 062705 000002          ADD    #2,R5
2972 013652 105367 000013          DECB  ADRCNT ,HOW MANY TIMES + 2 ?
2973 013656 001372          BNE   1$
2974 013660 000002          PARTI RTI
2975 013662 000000          LOLIM 0
2976 013664 000000          HILIM 0
2977 013666 000000          DEVADR 0
2978 013670 000000          LOBITS 0
2979          ADRCNT=LOBITS+1
2980
2981          ,SAVE PC OF TEST THAT FAILED AND RO-R5
2982
2983 013672 016667 000004 165226  SAV05 MOV    4(SP),SAVPC
2984
2985          ,SAVE RO-R5
2986
2987 013700 010567 165570          SV05  MOV    R5,$REG5
2988 013704 010467 165562          MOV    R4,$REG4
2989 013710 010367 165554          MOV    R3,$REG3
2990 013714 010267 165546          MOV    R2,$REG2
2991 013720 010167 165540          MOV    R1,$REG1
2992 013724 010067 165532          MOV    R0,$REG0
2993 013730 000002          RTI
2994
2995          ,RESTORE RO-R5
2996
2997 013732 016700 165524          RES05 MOV    $REG0,R0
2998 013736 016701 165522          MOV    $REG1,R1
2999 013742 016702 165520          MOV    $REG2,R2
3000 013746 016703 165516          MOV    $REG3,R3
3001 013752 016704 165514          MOV    $REG4,R4
3002 013756 016705 165512          MOV    $REG5,R5
3003 013762 000002          RTI
3004
3005          ,CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3006
3007 013764 104401          CONVR TYPE
3008 013766 015443          MCRLF ,CR LF
3009 013770 017601 000000          MOV    @ (SP),R1      ,PICK UP DATA POINTER
3010 013774 062716 000002          ADD    #2,(SP) ,SET UP SP FOR RTI
3011 014000 012167 000130          MOV    (R1)+,WRDCNT  ,PICK UP # OF WORDS FROM TABLE
3012 014004 112167 000126          1$    MOVB  (R1)+,CHRCNT  ,PICK UP # OF CHARS FROM TABLE
3013 014010 112167 000123          MOVB  (R1)+,SPACNT  ,PICK UP # OF SPACES FROM TABLE
3014 014014 013167 000120          MOV    @ (R1)+,BINWRD ,PICK UP ADDRESS OF MSG
3015          ,FROM TABLE
3016 014020 016704 000114          2$    MOV    BINWRD,R4      ,SAVE
3017 014024 116705 000106          MOVB  CHRCNT,R5      ,SAVE
3018 014030 012700 016214          MOV    #TEMP,R0      ,STARTING ADDRESS OF TEMP BLOCK
3019 014034 010403          3$    MOV    R4,R3          ,SAVE
3020 014036 042703 177770          BIC   #177770,R3     ,CLR OUT UPPER BITS   SAVE CHAR
3021 014042 062703 000260          ADD   #260,R3 ,CONVERT TO ASCII
3022 014046 110320          MOVB  R3,(R0)+      ,STORE AWAY
3023 014050 006204          ASR   R4          ,SHIFT FOR NEXT #
    
```

```

3024 014052 006204 ASR R4 ;DITTO
3025 014054 006204 ASR R4 ;DITTO
3026 014056 005305 DEC R5 ;DEC CHAR COUNT
3027 014060 001365 BNE 3$ ;DO IT AGAIN ?
3028 014062 012703 016256 MOV #MDATA,R3 ;STARTING ADDRESS OF MDATA BLOCK
3029 014066 114023 4$ MOVB -(R0),(R3)+ ;REVERSE THE ORDER OF NUMBERS
3030 014070 105367 000042 DECB CHRCNT ;DEC CHAR COUNT
3031 014074 001374 BNE 4$ ;DO IT AGAIN ?
3032 014076 105767 000035 TSTB SPACNT ;HOW MANY SPACES ?
3033 014102 001405 BEQ 6$ ;TYPE # IF BR =0
3034 014104 112723 000240 5$ MOVB #240,(R3)+ ;"SPACE" IN ASCI
3035 014110 105367 000023 DECB SPACNT ;DEC # OF SPACE COUNT
3036 014114 001373 BNE 5$ ;DO IT AGAIN ?
3037 014116 105013 6$ CLRB (R3) ;INSERT "0" FOR TTY OUTPUT ROUTINE
3038 014120 104401 TYPE
3039 014122 016256 MDATA ;THIS MESSAGE
3040 014124 005367 000004 DEC WRDCNT ;HOW MANY #'S ?
3041 014130 001325 BNE 1$ ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3042 014132 000002 RTI ;RETURN TO PROGRAM
3043 014134 000000 WRDCNT 0
3044 014136 000000 CHRCNT 0
3045 014137 014137 SPACNT=CHRCNT+1
3046 014140 000000 BINWRD 0
3047
3048 ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3049 ;BUFFER TO THE CHARACTERS 'N' AND 'Y'
3050 ;IF THE CHARACTER IS 'N' CLEAR THE FLAG
3051 ;IF THE CHARACTER IS 'Y' SET THE FLAG
3052
3053 014142 017605 000000 SETFLG MOV 3(SP),R5
3054 014146 122767 000116 001776 CMPB #'N,INBUF ;IS IT "N" ?
3055 014154 001002 BNE 1$
3056 014156 105015 CLRB (R5) ;000
3057 014160 000406 BR 2$
3058 014162 122767 000131 001762 1$ CMPB #'Y,INBUF ;IS IT "Y" ?
3059 014170 001005 BNE 3$
3060 014172 112715 177777 MOVB #-1,(R5) 377
3061 014176 062716 000002 2$ ADD #2,(SP)
3062 014202 000002 RTI
3063 014204 104407 3$ INSTER ;RETRY
3064 014206 000755 BR SETFLG
3065 SBTTL ERROR HANDLER ROUTINE
3066
3067 ;*****
3068 ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
3069 ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3070 ;AND GO TO SAVIT ON ERROR
3071 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3072 ;*SW15=1 HALT ON ERROR
3073 ;*SW13=1 INHIBIT ERROR TYPEOUTS
3074 ;*SW10=1 BELL ON ERROR
3075 ;*SW09=1 LOOP ON ERROR
3076 ;*CALL
3077 ;* ERROR N ;.ERROR=EMT AND N=ERROR ITEM NUMBER
3078
3079 014210 SEPROR
  
```

```

3080 014210 105267 165167 7$ INCB SERFLG // SET THE ERROR FLAG
3081 014214 001775 BEQ 7$ // DON'T LET THE FLAG GO TO ZERO
3082 014216 016777 165160 165216 MOV STSTNM, @DISPLAY // DISPLAY TEST NUMBER AND ERROR FLAG
3083 014224 032777 002000 165206 BIT #BIT10, @SWR // BELL ON ERROR?
3084 014232 001402 BEQ 1$ // NO - SKIP
3085 014234 104401 001516 TYPE ,SBELL // RING BELL
3086 014240 005267 165146 15 INC SERTTL // COUNT THE NUMBER OF ERRORS
3087 014244 011667 165146 MOV (SP), SERRPC // GET ADDRESS OF ERROR INSTRUCTION
3088 014250 162767 000002 165140 SUB #2, SERRPC
3089 014256 117767 165134 165130 MOVB @SERRPC, $ITEMB // STRIP AND SAVE THE ERROR ITEM CODE
3090 014264 032777 020000 165146 BIT #BIT13, @SWR // SKIP TYPEOUT IF SET
3091 014272 001004 BNE 20$ // SKIP TYPEOUTS
3092 014274 004767 000072 JSR PC, SAVIT // GO TO USER ERROR ROUTINE
3093 014300 104401 001523 TYPE ,SCLF
3094 014304 20$
3095 014304 122767 000001 165234 CMPB #APTENV, $ENV // RUNNING IN APT MODE
3096 014312 001007 BNE 2$ // NO, SKIP APT ERROR REPORT
3097 014314 116767 165074 000004 MOVB $ITEMB, 21$ // SET ITEM NUMBER AS ERROR NUMBER
3098 014322 004767 000016 JSR PC, $ATY4 // REPORT FATAL ERROR TO APT
3099 014326 000 21$ BYTE 0
3100 014327 000 BYTE 0
3101 014330 000777 22$ BR 22$ // APT EPROR LOOP
3102 014332 005777 165102 2$ TST @SWR // HALT ON ERROR
3103 014336 100001 BPL 3$ // SKIP IF CONTINUE
3104 014340 000000 HALT // HALT ON ERROR!
3105 014342 032777 001000 165070 3$ BIT #BIT09, @SWR // LOOP ON ERROR SWITCH SET?
3106 014350 001402 BEQ 4$ // BR IF NO
3107 014352 016716 165032 MOV $LPERR, (SP) // FUDGE RETURN FOR LOOPING
3108 014356 005767 165132 4$ TST $ESCAPE // CHECK FOR AN ESCAPE ADDRESS
3109 014362 001402 BEQ 5$ // BR IF NONE
3110 014364 016716 165124 MOV $ESCAPE, (SP) // FUDGE RETURN ADDRESS FOR ESCAPE
3111 014370 5$
3112 014370 000002 RTI // RETURN
3113 014372 010067 164532 SAVIT MOV R0, HLD0
3114 014376 010167 164530 MOV R1, HLD1
3115 014402 010267 164526 MOV R2, HLD2
3116 014406 010367 164524 MOV R3, HLD3
3117 014412 010467 164522 MOV R4, HLD4
3118 014416 010567 164520 MOV R5, HLD5
3119 014422 016767 164754 164514 MOV $TSTNM, HLD6

```

SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

3120
3121
3122
3123 // *****
3124 // *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3125 // *ERROR IS TO BE REPORTED IT THEN OBTAINS, FROM THE "ERROR TABLE" ($EPTB),
3126 // *AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR
3127

```

SERRTYP

```

3128 014430 TYPE ,SCLF // "CARRIAGE RETURN" & "LINE FEED"
3129 014430 104401 001523 MOV R0, -(SP) // SAVE R0
3130 014434 010046 CLR R0 // PICKUP THE ITEM INDEX
3131 014436 005000 BISB @SITEMB, R0
3132 014440 153700 001414 BNE 1$ // IF ITEM NUMBER IS ZERO, JUST
3133 014444 001004 // TYPE THE PC OF THE ERROR
3134 // SAVE SERRPC FOR TYPEOUT
3135 014446 016746 164744 MOV SERRPC, -(SP)

```

```

3136                                     .. ERROR ADDRESS
3137 014452 104402                       TYPUC .. GO TYPE--OCTAL ASCII(ALL DIGITS)
3138 014454 000426                       BR 6$ .. GET OUT
3139 014456 005300                       1$ DEC RO .. ADJUST THE INDEX SO THAT IT WILL
3140 014460 006300                       ASL RO .. WORK FOR THE ERROR TABLE
3141 014462 006300                       ASL RO
3142 014464 006300                       ASL RO
3143 014466 062700 001652                 ADD #SERRTB,RO .. FORM TABLE POINTER
3144 014472 012067 000004                 MOV (RO)+,2$ .. PICKUP "ERROR MESSAGE" POINTER
3145 014476 001404                       BEQ 3$ .. SKIP TYPEOUT IF NO POINTER
3146 014500 104401                       TYPE .. TYPE THE "ERROR MESSAGE"
3147 014502 000000                       2$ WORD 0 .. "ERROR MESSAGE" POINTER GOES HERE
3148 014504 104401 001523                 TYPE , $CRLF .. "CARRIAGE RETURN" & "LINE FEED"
3149 014510 012067 000004                 3$ MOV (RO)+,4$ .. PICKUP "DATA HEADER" POINTER
3150 014514 001404                       BEQ 5$ .. SKIP TYPEOUT IF 0
3151 014516 104401                       TYPE .. TYPE THE "DATA HEADER"
3152 014520 000000                       4$ WORD 0 .. "DATA HEADER" POINTER GOES HERE
3153 014522 104401 001523                 TYPE , $CRLF .. "CARRIAGE RETURN" & "LINE FEED"
3154 014526 011000                       5$ MOV (RO),RO .. PICKUP "DATA TABLE" POINTER
3155 014530 001004                       BNE 7$ .. GO TYPE THE DATA
3156 014532 012600                       6$ MOV (SP)+,RO .. RESTORE RO
3157 014534 104401 001523                 TYPE , $CRLF .. "CARRIAGE RETURN" & "LINE FEED"
3158 014540 000207                       RTS PC .. RETURN
3159 014542                               7$
3160 014542 013046                       MOV @ (RO)+, -(SP) .. SAVE @ (RO)+ FOR TYPEOUT
3161 014544 104402                       TYPUC .. GO TYPE--OCTAL ASCII(ALL DIGITS)
3162 014546 005710                       TST (RO) .. IS THERE ANOTHER NUMBER?
3163 014550 001770                       BEQ 6$ .. BR IF NO
3164 014552 104401 014560                 TYPE , 8$ .. TYPE TWO(2) SPACES
3165 014556 000771                       BR 7$ .. LOOP
3166 014560 020040 000                     8$ ASCIZ / / .. TWO(2) SPACES
3167 014564                               EVEN
3168 SBTTL BINARY TO OCTAL (ASCII) AND TYPE
3169
3170 .. *****
3171 .. *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3172 .. *OCTAL (ASCII) NUMBER AND TYPE IT
3173 .. *$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3174 .. *CALL
3175 .. * MOV NUM, -(SP) .. NUMBER TO BE TYPED
3176 .. * TYPOS .. CALL FOR TYPEOUT
3177 .. * BYTE N .. N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3178 .. * BYTE M .. M=1 OR 0
3179 .. * .. 1=TYPE LEADING ZEROS
3180 .. * .. 0=SUPPRESS LEADING ZEROS
3181 .. *
3182 .. *$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3183 .. *$TYPOS OR $TYPOC
3184 .. *CALL
3185 .. * MOV NUM, -(SP) .. NUMBER TO BE TYPED
3186 .. * TYPON .. CALL FOR TYPEOUT
3187 .. *
3188 .. *$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3189 .. *CALL
3190 .. * MOV NUM, -(SP) .. NUMBER TO BE TYPED
3191 .. * TYPUC .. CALL FOR TYPEOUT
    
```


3248	015012				\$PWRDN			
3249	015012	010046			PFAIL	MOV	R0, -(SP)	, SAVE R0-R5 ON PROCESSOR STACK
3250	015014	010146				MOV	R1, -(SP)	
3251	015016	010246				MOV	R2, -(SP)	
3252	015020	010346				MOV	R3, -(SP)	
3253	015022	010446				MOV	R4, -(SP)	
3254	015024	010546				MOV	R5, -(SP)	
3255	015026	016746	162772			MOV	24, -(SP)	
3256	015032	010667	164060			MOV	SP, SAVSP	, SAVE STACK POINTER
3257	015036	012767	015053	162760		MOV	#RESTART, 24	, SET UP FOR POWER UP TRAP
3258	015044	000000				HALT		, HALT ON POWER DOWN NORMAL
3259	015046	000777				BR		
3260								
3261								, PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3262								
3263	015050	016706	164042		RESTAR	MOV	SAVSP, SP	, RESTORE STACK POINTER
3264	015054	012605				MOV	(SP)+, R5	, RESTORE R0-R5
3265	015056	012604				MOV	(SP)+, R4	
3266	015060	012603				MOV	(SP)+, R3	
3267	015062	012602				MOV	(SP)+, R2	
3268	015064	012601				MOV	(SP)+, R1	
3269	015066	012600				MOV	(SP)+, R0	
3270	015070	012767	015012	162726		MOV	# PFAIL, 24	, SET UP FOR POWER FAILURE
3271	015076	106427	000340			MTPS	#340	
3272	015102	012706	001100			MOV	#STACK, SP	
3273	015106	005067	001102			CLR	TEMP	
3274	015112	005267	001076			INC	TEMP	
3275	015116	001375				BNE	-4	
3276	015120	104413				CONVRT		
3277	015122	015144				PFTAB		
3278	015124	104401				TYPE		
3279	015126	015446				MPFAIL		
3280	015130	005067	164247			CLR	\$ERFLG	
3281	015134	005067	164256			CLR	\$ERRPC	
3282	015140	000177	163740			JMP	@RETURN	
3283	015144	000001			PFTAB	1		
3284	015146	006	002			BYTE	6, 2	
3285	015150	000207				RETURN		
3286	015152	005015	042012	053125	MTITLE	ASCIZ	<15><12><12>/DUV11 DZDUV-B TAPE F /<15><12>	
3287	015160	030461	042040	042132				
3288	015166	053125	041055	052040				
3289	015174	050101	020105	020106				
3290	015202	005015	000					
3291	015205	015	053012	041505	MVECTO	ASCIZ	<15><12>/VEC ADD- /	
3292	015212	040440	042104	000055				
3293	015220	005015	051461	020124	MREGAD	ASCIZ	<15><12>/1ST DEV REC CSR ADD- /	
3294	015226	042504	035126	051040				
3295	015234	041505	041440	051123				
3296	015242	040440	042104	000055				
3297	015250	005015	052515	052114	MMULT	ASCIZ	<15><12>/MULT DEV ? (Y OR N)- /	
3298	015256	042040	053105	037440				
3299	015264	024040	020131	051117				
3300	015272	047040	026451	000				
3301	015277	015	046012	051501	MLASTD	ASCIZ	<15><12>/LAST DEV. REC CSR ADDR- /	
3302	015304	020124	042504	035126				
3303	015312	051040	041505	041440				

3304	015320	051123	040440	042104			
3305	015326	026522	000				
3306	015331	075	042504	044526	DEVICE	ASCIZ	/=DEVICE /
3307	015336	042503	020040	000			
3308	015343	015	051412	046105	MCOW	ASCIZ	<15><12>/SELECT TO RUN @ACTREG/
3309	015350	041505	020124	047524			
3310	015356	051040	047125	040040			
3311	015364	041501	051124	043505			
3312	015372	000					
3313	015373	015	047412	043126	MRANGE	ASCIZ	<15><12>/OVFLO. RETYPE LAST DEV RXCSR ADDS-/
3314	015400	047514	051072	052105			
3315	015406	050131	020105	040514			
3316	015414	052123	042040	053105			
3317	015422	051040	041530	051123			
3318	015430	040440	042104	026523			
3319	015436	000					
3320	015437	040	037440	000	MQM	ASCIZ	/ ?/
3321	015443	015	000012		MCRLF	ASCIZ	<15><12>
3322	015446	043120	044501	026114	MPFAIL	ASCIZ	/PFAIL, RESTART AT TEST IN PROGRESS/
3323	015454	020040	042522	052123			
3324	015462	051101	020124	052101			
3325	015470	052040	051505	020124			
3326	015476	047111	050040	047522			
3327	015504	051107	051505	000123			
3328	015512	005015	047105	020104	MEPASS	ASCIZ	<15><12>/END OF PASS TAPE F/
3329	015520	043117	050040	051501			
3330	015526	020123	040524	042520			
3331	015534	043040	000				
3332	015537	015	051012	000	MR	ASCIZ	<15><12>/R/
3333	015543	015	052012	051505	MTSTPC	ASCIZ	<15><12>/TEST PC-/
3334	015550	020124	041520	000055			
3335	015556	005015	047514	045503	MLOCK	ASCIZ	<15><12>/LOCK ON TEST? (Y OR N)-/
3336	015564	047440	020116	052040			
3337	015572	051505	037524	024040			
3338	015600	020131	051117	047040			
3339	015606	026451	000				
3340	015611	015	021412	047440	MSYNC	ASCIZ	<15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3341	015616	020106	054523	041516			
3342	015624	041440	040510	051522			
3343	015632	051440	046105	041505			
3344	015640	042524	020104	020050			
3345	015646	020061	051117	031040			
3346	015654	026451	000				
3347	015657	015	044412	020123	MWIRE6	ASCIZ	<15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3348	015664	042523	020103	046530			
3349	015672	052111	051440	044527			
3350	015700	04524	020110	032505			
3351	015706	026465	020062	047111			
3352	015714	020077	054450	047440			
3353	015722	020122	024516	000055			
3354	015730	005015	051511	051440	MWIRE5	ASCIZ	<15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3355	015736	041505	051040	041505			
3356	015744	051440	044527	041524			
3357	015752	020110	032505	026465			
3358	015760	020063	047111	020077			
3359	015766	054450	047440	020122			

```
3360 015774 024516 000055
3361 016000 005015 051511 047440 MWIRE4 ASCIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3362 016006 052120 041440 051114
3363 016014 042440 040516 046102
3364 016022 020105 053523 052111
3365 016030 044103 042440 032465
3366 016036 030455 044440 037516
3367 016044 024040 020131 051117
3368 016052 047040 026451 000
3369 016057 015 005012 031510 MEXTJ ASCIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3370 016064 032461 041440 047117
3371 016072 042516 052103 051117
3372 016100 047440 020116 024077
3373 016106 020131 051117 047040
3374 016114 026451 000
3375 016117 015 020012 043536 MCNTG ASCIZ <15><12>/ G /
3376 016124 020040 000
3377 016127 040 053523 036522 MMSWR ASCIZ / SWR= /
3378 016134 020040 000040
3379 016140 020040 047040 053505 MMNEW ASCIZ / NEW= /
3380 016146 020075 000040
3381 EVEN
3382
3383 .BUFFERS FOR INPUT-OUTPUT
3384
3385 016152 000000 INBUF 0
3386 016214 000000 = +40
3387 016214 000000 TEMP 0
3388 016256 000000 = +40
3389 016256 000000 MDATA 0
3390 016320 000000 = +40
3391 SBTTL SCOPE HANDLER ROUTINE
3392
3393 .,*****
3394 ., *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS IT WILL INCREMENT
3395 ., *AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG (DISPLAY<7 0>)
3396 ., *AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15 08>
3397 ., *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3398 ., *SW14=1 LOOP ON TEST
3399 ., *SW11=1 INHIBIT ITERATIONS
3400 ., *SW09=1 LOOP ON ERROR
3401 ., *SW08=1 LOOP ON TEST IN SWR<7 0>
3402 ., *CALL
3403 ., * SCOPE ., SCOPE=10T
3404
3405 016320 $SCOPE
3406
3407 ., SCOPE LOOP AND INTERATION HANDLER
3408
3409 016320 SCOPE
3410 016320 004767 174376 JSR PC,CKSWR
3411 016324 005067 163066 CLR $ERRPC ., CLEAR LAST ERROR PC
3412 016330 022716 003376 CMP #TST1+2, (SP) ., IS SCOPE AT BEGINING CF TEST 1?
3413 016334 001422 BEQ $XTSTR ., YES NO LOOP
3414
3415 016336 032777 040000 163074 TTST BIT #BIT14, @SWR ., THIS CODE IS FOR TESTING FOR BIT 14
```

```

3416 016344 001412          BEQ      1$          ,ON LSI WHICH SYSMAC CANNOT HANDLE
3417 016346 016767 163030 163032  MOV      $TSTNM,$LPADR
3418 016354 000406          BR       1$
3419 016356 105777 163062  TSTB    @STKS      ,KEYBOARD DONE?
3420 016362 100123          BPL     $OVER      ,BR IF NO
3421 016364 017766 163056 177776  MOV      @STKB,-2(SP) ;CLEAR DONE BIT
3422 016372 032777 040000 163040 1$  BIT     #BIT14,@SWR  ;LOOP ON PRESENT TEST?
3423 016400 001114          BNE     $OVER      ;YES IF SW14=1
3424          ,#####START OF CODE FOR THE XOR TESTER#####
3425 016402 000416          $XTSTR BR      6$      ;IF RUNNING ON THE "XOR" TESTER CHANGE
3426          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
3427 016404 013746 000004          MOV     @#ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3428 016410 012737 016430 000004  MOV     #55,@#ERRVEC  ;SET FOR TIMEOUT
3429 016416 005737 177060  TST     @#177060     ;TIME OUT ON XOR?
3430 016422 012637 000004          MOV     (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
3431 016426 000463          BR      $$VLAD      ;GO TO THE NEXT TEST
3432 016430 022626          5$  CMP     (SP)+,(SP)+  ;CLEAR THE STACK AFTER A TIME OUT
3433 016432 012637 000004          MOV     (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
3434 016436 000423          BR      7$          ;LOOP ON THE PRESENT TEST
3435 016440          6$  ,#####END OF CODE FOR THE XOR TESTER#####
3436 016440 032777 000400 162772  BIT     #BIT08,@SWR  ;LOOP ON SPEC TEST?
3437 016446 001404          BEQ     2$          ;BR IF NO
3438 016450 127767 162764 162724  CMPB   @SWR,$TSTNM  ;ON THE RIGHT TEST? SWR<7 0>
3439 016456 001465          BEQ     $OVER      ;BR IF YES
3440 016460 105767 162717          2$  TSTB   $ERFLG      ;HAS AN ERROR OCCURRED?
3441 016464 001421          BEQ     3$          ;BR IF NO
3442 016466 126767 162723 162707  CMPB   $ERMAX,$ERFLG ;MAX ERRORS FOR THIS TEST OCCURRED?
3443 016474 101015          BHI     3$          ;BR IF NO
3444 016476 032777 001000 162734  BIT     #BIT09,@SWR  ;LOOP ON ERROR?
3445 016504 001404          BEQ     4$          ;BR IF NO
3446 016506 016767 162676 162672  7$  MOV     $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
3447 016514 000446          BR      $OVER
3448 016516 105067 162661          4$  CLRB   $ERFLG      ;ZERO THE ERROR FLAG
3449 016522 005067 162764          CLR    $TIMES      ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3450 016526 000415          BR      1$          ;ESCAPE TO THE NEXT TEST
3451 016530 032777 004000 162702  3$  BIT     #BIT11,@SWR  ;INHIBIT ITERATIONS?
3452 016536 001011          BNE     1$          ;BR IF YES
3453 016540 00167 162770          TST    $PASS      ;IF FIRST PASS OF PROGRAM
3454 016544 001406          BEQ     1$          ;INHIBIT ITERATIONS
3455 016546 005267 162632          INC    $ICNT      ;INCREMENT ITERATION COUNT
3456 016552 026767 162734 162624  CMP    $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
3457 016560 002024          BGE    $OVER      ;BR IF MORE ITERATION REQUIRED
3458 016562 012767 000001 162614 1$  MOV    #1,$ICNT    ;REINITIALIZE THE ITERATION COUNTER
3459 016570 016767 000056 162714          MOV    $MXCNT,$TIMES ;SET NUMBER OF ITERATIONS TO DO
3460 016576 105267 162600          $$VLAD INCB    $TSTNM ;COUNT TEST NUMBERS
3461 016602 116767 162574 162722  MOVB   $TSTNM,$TESTN ;SET TEST NUMBER IN APT MAILBOX
3462 016610 011667 162572          MOV    (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
3463 016614 011667 162570          MOV    (SP),$LPERR ;SAVE ERROR LOOP ADDRESS
3464 016620 005067 162670          CLR    $ESCAPE    ;CLEAR THE ESCAPE FROM ERROR ADDRESS
3465 016624 112767 000001 162563          MOVB   #1,$ERMAX  ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3466 016632 016777 162544 162602  $OVER MOV    $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER
3467 016640 016716 162542          MOV    $LPADR,(SP) ;FUJGE RETURN ADDRESS
3468 016644 000002          4$  RTI
3469 016646 001407          BRW    1407
3470 016650 000432          BRX    432
3471 016652 000005          $MXCNT 5          ,MAX NUMBER OF ITEPAT NS
    
```

```

3472          SBTTL TRAP DECODER
3473
3474          , , *****
3475          , *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3476          , *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3477          , *OF THE DESIRED ROUTINE THEN USING THE ADDRESS OBTAINED IT WILL
3478          , *GO TO THAT ROUTINE
3479
3480 016654 010046          STRAP  MOV  RO, -(SP)          , , SAVE RO
3481 016656 016600 000002      MOV  2(SP), RO          , , GET TRAP ADDRESS
3482 016662 005740          TST  -(RO)            , , BACKUP BY 2
3483 016664 111000          MOVB (RO), RO         , , GET RIGHT BYTE OF TRAP
3484 016666 006300          ASL  RO              , , POSITION FOR INDEXING
3485 016670 016000 016710      MOV  STRPAD(RO), RO    , , INDEX TO TABLE
3486 016674 000200          RTS  RO              , , GO TO ROUTINE
3487
3488
3489          , , THIS IS USE TO HANDLE THE "GETPRI" MACRO
3490
3491 016676 011646          STRAP2 MOV  (SP), -(SP)      , , MOVE THE PC DOWN
3492 016700 016666 000004 000002      MOV  4(SP), 2(SP)     , , MOVE THE PSW DOWN
3493 016706 000J02          RTI                    , , RESTORE THE PSW
3494
3495          SBTTL TRAP TABLE
3496
3497          , *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3498          , *BY THE "TRAP" INSTRUCTION
3499
3500          ROUTINE
3501          -----
3502 016710 016676          STRPAD  WORD  STRAP2          TRAP+1(104401)  TTY TYPEOUT ROUTINE
3503 016712 013072          STYPE  , , CALL=TYPE          TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3504 016714 014610          STYPOC , , CALL=TYPOC          TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
3505 016716 014564          STYPOS , , CALL=TYPOS          TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
3506 016720 014624          STYPON , , CALL=TYPON
3507
3508
3509 016722 013050          SCOP1  , , CALL=SCOP1          TRAP+5(104405)
3510 016724 013354          INSTR , , CALL=INSTR          TRAP+6(104406)
3511 016726 013462          INSTER , , CALL=INSTER          TRAP+7(104407)
3512 016730 013472          PARAM , , CALL=PARAM          TRAP+10(104410)
3513 016732 013672          SAVOS , , CALL=SAVOS          TRAP+11(104411)
3514 016734 013732          RESOS , , CALL=RESOS          TRAP+12(104412)
3515 016736 013764          CONVRT , , CALL=CONVRT          TRAP+13(104413)
3516 016740 014142          SETFLG , , CALL=SETFLG          TRAP+14(104414)
3517          , *****
3518          , UTILITIES
3519          , *****
3520
3521          , THIS UTILITY CALCULATES PRIORITY LEVEL
3522 016742 006367 000044      DULEV ASL  DUPRT , SHIFT LEFT
3523 016746 006367 000040      ASL  DUPRT ,
3524 016752 006367 000034      ASL  DUPRT ,
3525 016756 006367 000030      ASL  DUPRT ,
3526 016762 006367 000024      ASL  DUPRT ,
3527 016766 016767 000020 000020      MOV  DUPRT, LESS1      , MOVE THIS TO LESS1
  
```

3528	016774	162767	000001	000012		SUB	#1, LESS1	, CREATE LESS1
3529	017002	042767	000037	000004		BIC	#37, LESS1	, CLEAR TNZVC
3530	017010	000207				RTS	PC	
3531	017012	000240			DUPRT	PR5		
3532	017014	000200			LESS1	PR4		, LEVEL TO ALLOW INTERRUPTS
3533								
3534								, NEW DU ADDRESSES
3535	017016	016767	000126	162666	DUADDR	MOV	DUBASE, RXCSR	, XXX0
3536	017024	005267	000120			INC	DUBASE	
3537	017030	016767	000114	162656		MOV	DUBASE, HRXCSR	, XXX1
3538	017036	005267	000106			INC	DUBASE	
3539	017042	016767	000102	162646		MOV	DUBASE, RXDBUF	, XXX2
3540	017050	016767	000074	162644		MOV	DUBASE, PARCSR	, XXX2
3541	017056	005267	000066			INC	DUBASE	
3542	017062	016767	000062	162630		MOV	DUBASE, HRXDBUF	, XXX3
3543	017070	016767	000054	162626		MOV	DUBASE, HPARCSR	, XXX3
3544	017076	005267	000046			INC	DUBASE	
3545	017102	016767	000042	162616		MOV	DUBASE, TXCSR	, XXX4
3546	017110	005267	000034			INC	DUBASE	
3547	017114	016767	000030	162606		MOV	DUBASE, HTXCSR	, XXX5
3548	017122	005267	000022			INC	DUBASE	
3549	017126	016767	000016	162576		MOV	DUBASE, TXDBUF	, XXX6
3550	017134	005267	000010			INC	DUBASE	
3551	017140	016767	000004	162566		MOV	DUBASE, HTXDBUF	, XXX7
3552	017146	000207				RTS	PC	
3553	017150	000000				DUBASE	0	
3554								
3555								, THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3556								, INFORMATION CONTAINED IN STMP1 AND IT S
3557								, SHIFTED IN BY THE CONTENTS OF SHIFT
3558	017152	042777	040000	162546	RPOKE	BIC	#MTDATA, @TXCSR	
3559	017160	005067	162316			CLR	STMP2	
3560	017164	006067	162310			ROR	STMP1	, FORCE CARRY
3561	017170	006067	162306			ROR	STMP2	, PICK UP CARRY IN BIT 15
3562	017174	006267	162302			ASR	STMP2	, SHIFT INTO BIT 14
3563	017200	042767	100000	162274		BIC	#BIT15, STMP2	, CLR BIT 15
3564	017206	056777	162270	162512		BIS	STMP2, @TXCSR	, POKE MAINT DATA
3565	017214	042777	020000	162504		BIC	#CLK, @TXCSR	, POKE CLK
3566	017222	052777	020000	162476		BIS	#CLK, @TXCSR	
3567	017230	005367	161666			DEC	SHIFT	
3568	017234	001346				BNE	RPOKE	
3569	017236	000207				RTS	PC	
3570								
3571								, THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3572	017240	016767	162234	162234	ODD8	MOV	STMP1, STMP2	, SAVE TEMP1
3573	017246	005067	162232			CLP	STMP3	
3574	017252	012727	000010			MOV	#8, (PC)+	
3575	017256	000000			45	0		
3576	017260	006067	162216		15	ROR	STMP2	
3577	017264	005567	162214			ADC	STMP3	
3578	017270	005367	177762			DEC	45	
3579	017274	001371				BNE	15	
3580	017276	006067	162202			ROR	STMP3	
3581	017302	103404				BCS	25	
3582	017304	052767	000400	162166		BIS	#BIT8, STMP1	, SET ODD PARITY
3583	017312	000403				BR	35	

DZDUV-B MACY11 30(1046) 21-SEP-77 09 27 PAGE 72
 DZDUV-B M11 31-MAY-77 09 51 TRAP TABLE

SEQ 0070

```

3584 017314 042767 000400 162156 2$ BIC #BIT8,$TMP1 ,CLR EVEN PAR TY
3585 , $TMP1 NOW HAS ODD PARITY CHARACTER
3586 017322 000207 3$ RTS PC
3587
3588 , THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHRRACTER
3589 017324 016767 162150 162150 EVEN8 MOV $TMP1,$TMP2 ,SAVE TEMP1
3590 017332 005067 162146 CLR $TMP3
3591 017336 012727 000010 MOV #8 ,(PC)+
3592 017342 000000 4$ 0
3593 017344 006067 162132 1$ ROR $TMP2
3594 017350 005567 162130 ADC $TMP3
3595 017354 005367 177762 DEC 4$
3596 017360 001371 BNE 1$
3597 017362 006067 162116 ROR $TMP3
3598 017366 103004 BCC 2$
3599 017370 052767 000400 162102 BIS #BIT8,$TMP1 ,SET EVEN PARITY
3600 017376 000403 BR 3$
3601 017400 042767 000400 162072 2$ BIC #BIT8,$TMP1 ,CLR ODD PARITY
3602 , $TMP1 NOW HAS EVEN PARITY CHARACTER
3603 017406 000207 3$ RTS PC
3604 017410 062716 000002 TRPREG ADD #2,(SP) ,ALLOW IT TO "CRUNCH" NTO HLT BACK
3605 , IN MAIN PART OF THE PROGRAM
3606 017414 000002 RTI
3607 000001 ENC
    
```

AAA	003200	1293*								
ABASE =	000000	876	917							
ACDW1 =	000000	876	919							
ACDW2 =	000000	876	920							
ACPUOP=	000000	876	891							
ACTREG	001166	735*	1249*	1263*	1264*	1271*	2718	2721	2731	
ADDW0 =	000000	876	921							
ADDW1 =	000000	876	922							
ADDW10=	000000	876	931							
ADDW11=	000000	876	932							
ADDW12=	000000	876	933							
ADDW13=	000000	876	934							
ADDW14=	000000	876	935							
ADDW15=	000000	876	936							
ADDW2 =	000000	876	923							
ADDW3 =	000000	876	924							
ADDW4 =	000000	876	925							
ADDW5 =	000000	876	926							
ADDW6 =	000000	876	927							
ADDW7 =	000000	876	928							
ADDW8 =	000000	876	929							
ADDW9 =	000000	876	930							
ADEVCT=	000000	876	882							
ADEVN =	000000	876	918							
ADR CNT=	013671	2933*	2972*	2979*						
AENV =	000000	876	887							
AENVN =	000000	876	888							
AFATAL=	000000	876	879							
AMADR1=	000000	876	904							
AMADR2=	000000	876	908							
AMADR3=	000000	876	911							
AMADR4=	000000	876	914							
AMAMS1=	000000	876	898							
AMAMS2=	000000	876	906							
AMAMS3=	000000	876	909							
AMAMS4=	000000	876	912							
AMSGAD=	000000	876	884							
AMSGLG=	000000	876	885							
AMSGTY=	000000	876	878							
AMTYP1=	000000	876	899							
AMTYP2=	000000	876	907							
AMTYP3=	000000	876	910							
AMTYP4=	000000	876	913							
APASS =	000000	876	881							
APRIOR=	000000	876								
APTCSU=	000040	536*	2848							
APTENV=	000001	536*	2841	3095						
APTSIZ=	000200	536*	1168							
APTSP0=	000100	536*	2847							
ASWREG=	000000	876	889							
ATESTN=	000000	876	880							
AUNIT =	000000	876	883							
AUSWR =	000000	876	890							
AVECT1=	000000	876	915							
AVECT2=	000000	876	916							
BASEAD	001154	730*	1231*	1268*	1269	1275*	1277*	2725*	2737*	2741

CROSS REFERENCE TABLE -- USER SYMBOLS

SW15 = 100000	612#													
SW2 = 000004	635#													
SW3 = 000010	634#													
SW4 = 000020	633#													
SW5 = 000040	632#													
SW6 = 000100	631#													
SW7 = 000200	630#													
SW8 = 000400	629#													
SW9 = 001000	628#													
SYNCNO 001146	721#	1298*	1302*	1878	2018	2101	2184	2267						
SYNEXT= 020000	789#	982#	1353	1360	1426	1433	1515	1523	1697	1705	1783	1785	2310	
	2318	2358	2366	2408	2416	2479	2487							
SYNINT= 030000	788#	981#	1847	1855	1986	2069	2152	2235						
SYNSCH= 000020	774#	967#	1361	1434	1796	1856	1989	2044	2072	2127	2155	2210	2238	
	2293	2322	2370	2420	2491	2557	2612	2667						
SYSTST= 014000	814#	1007#	1794	1852										
TBITVE= 000014	670#													
TEMP 016214	3018	3273*	3274*	3387#										
TKVEC = 000060	677#													
TPVEC = 000064	678#													
TRAPVE= 000034	676#	1142*	1143*											
TRPREG 017410	3604#													
TRTVEC= 000014	671#													
TST1 003374	1333	1340	1350#	2773	3412									
TST10 005622	1778#													
TST11 006112	1845#													
TST12 006726	1982#													
TST13 007232	2065#													
TST14 007536	2148#													
TST15 010042	2231#													
TST16 010346	2308#													
TST17 010544	2356#													
TST2 003720	1421#													
TST20 010742	2404#													
TST21 011206	2475#													
TST22 011452	2547#													
TST23 011700	2602#													
TST24 012126	2657#													
TST3 004336	1512#													
TST4 004610	1569#													
TST5 004752	1607#													
TST6 005110	1650#													
TST7 005260	1694#													
TTST 016336	3415#													
TXCSR 001726	1048#	1352*	1354*	1357*	1363*	1364*	1387*	1391*	1392*	1393*	1425*	1427*	1430*	
	1436*	1437*	1480*	1484*	1485*	1486*	1514*	1516*	1520*	1527*	1528*	1531*	1532*	
	1538*	1540*	1548	1551*	1560*	1571*	1575*	1579*	1585*	1586	1608*	1612*	1616*	
	1623*	1651*	1655*	1672	1684*	1696*	1698*	1702*	1709*	1710*	1713*	1714*	1720*	
	1729	1732	1738*	1745*	1754	1757*	1764*	1782*	1784*	1794*	1846*	1848*	1852*	
	1882*	1894*	1913*	1927*	1983*	1987*	1992*	1993*	1996*	1997*	2005	2011*	2012*	
	2038*	2039	2043*	2066*	2070*	2075*	2076*	2079*	2080*	2088	2094*	2095*	2121*	
	2122	2126*	2149*	2153*	2158*	2159*	2162*	2163*	2171	2177*	2178*	2204*	2205	
	2209*	2232*	2236*	2241*	2242*	2245*	2246*	2254	2260*	2261*	2287*	2288	2292*	
	2309*	2311*	2315*	2324*	2325*	2329*	2330*	2357*	2359*	2363*	2372*	2373*	2377*	
	2378*	2407*	2409*	2413*	2422*	2428*	2437*	2443*	2478*	2480*	2484*	2493*	2499*	
	2508*	2514*	2548*	2550*	2553*	2559*	2560*	2562*	2563*	2603*	2605*	2608*	2614*	

SERRT	525#	3121
SPOWE	525#	
SSCOP	525#	3391
STRAP	525#	3472
STYPE	525#	2818
STYPO	525#	3168

RBS 017416 000

ERRORS DETECTED 0

DZDUVB DZDUVB/SOL/CRF=DZDUV1/EQ RUNF, DZDUV2, DZDUVB

RUN-TIME 21 12 1 SECONDS

PUN-TIME PATIO 270/34=7 8

CORE USED 30K (59 PAGES)