

REM *

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

PRODUCT CODE MAINDEC-11-DZDUU-B-D

PRODUCT NAME DUV1! OFFLINE COMBINED TIMING & INTERRUPT 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 COMBINED TIMING AND INTERRUPT TESTS VERIFY THAT THE TRANSMITTER AND RECEIVER CAN CORRECTLY TALK TO EACH OTHER INTERRUPT LOGIC AND PRIORITY LEVEL /VECTOR ADDRESSES ARE ALSO VERIFIED

* REM *

2 REQUIREMENTS

PDP-11/03 COMPUTER (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 PESIDE 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 "SUSWR" IN ORDER TO BE FLEXIBLE ON THE AVAILIBILITY OF THE H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11. BIT 0 IN \$USWR REFLECTS THIS STATUS, 9 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
 - 4 3 1 5 THE PROGRAM WILL TYPE "DUV11 DZDUU-B TAPE E" (ONCE ONLY)
 - 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

*
REM *
*
PEM *

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 OPERATING 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 ADDRESSES
&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
!6XXXX 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 IV",
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 H315 CONNECTOR ON)

7.3 1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE,
LOCATION "HOLD " MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES
PRESENTLY "HOLD " =20 IS SUFFICIENT TIME ON AN 11/03 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 RECEIVER 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 E55-2 ON?- YES SEXMIT 377
IS SEC REC SWITCH E55-3 ON?- YES SEREC 377
IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR 377
DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
CONNECTOR ON (H315)- YES JMRBY 377

9 PROGRAM DESCRIPTION

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

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

11 LISTINGS

*
REM *
*
REM *
*

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
558

000001

STN=1

```
559          ENABLE ABS
560
561          ,DUV11 DZDUU-B TAPE E
562          ,COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS 01754
563
564          ; STARTING PROCEDURE
565          ; TYPE 200G
566          ; PROGRAM WILL TYPE "DUV11 DZDUU-B TAPE E "
567          ; PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
568          ; AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE E"
569          ; AND THEN RESUME TESTING
570
571          .SBTTL BASIC DEFINITIONS
572
573          ,*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
574          001100 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          000011 HT= 11          ;; CODE FOR HORIZONTAL TAB
580          000012 LF= 12          ;; CODE FOR LINE FEED
581          000015 CR= 15          ;; CODE FOR CARRIAGE RETURN
582          000200 CRLF= 200       ;; CODE FOR CARRIAGE RETURN-LINE FEED
583          177776 PS= 177776     ;; PROCESSOR STATUS WORD
584          .EQUIV PS,PSW
585          177774 STKLMT= 177774  ;; STACK LIMIT REGISTER
586          177772 PIRQ= 177772   ;; PROGRAM INTERRUPT REQUEST REGISTER
587          177570 DSWR= 177570   ;; HARDWARE SWITCH REGISTER
588          177570 DDISP= 177570  ;; HARDWARE DISPLAY REGISTER
589
590          ,*GENERAL PURPOSE REGISTER DEFINITIONS
591          000000 R0= %0          ;; GENERAL REGISTER
592          000001 R1= %1          ;; GENERAL REGISTER
593          000002 R2= %2          ;; GENERAL REGISTER
594          000003 R3= %3          ;; GENERAL REGISTER
595          000004 R4= %4          ;; GENERAL REGISTER
596          000005 R5= %5          ;; GENERAL REGISTER
597          000006 R6= %6          ;; GENERAL REGISTER
598          000007 R7= %7          ;; GENERAL REGISTER
599          000006 SP= %6          ;; STACK POINTER
600          000007 PC= %7          ;; PROGRAM COUNTER
601
602          ,*PRIORITY LEVEL DEFINITIONS
603          000000 PRO= 0          ;; PRIORITY LEVEL 0
604          000040 PR1= 40         ;; PRIORITY LEVEL 1
605          000100 PR2= 100        ;; PRIORITY LEVEL 2
606          000140 PR3= 140        ;; PRIORITY LEVEL 3
607          000200 PR4= 200        ;; PRIORITY LEVEL 4
608          000240 PR5= 240        ;; PRIORITY LEVEL 5
609          000300 PR6= 300        ;; PRIORITY LEVEL 6
610          000340 PR7= 340        ;; PRIORITY LEVEL 7
611
612          ,*"SWITCH REGISTER" SWITCH DEFINITIONS
613          100000 SW15= 100000
614          040000 SW14= 40000
```

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

*DATA BIT DEFINITIONS (BIT00 TO BIT15)

640		BIT15=	100000
641	100000	BIT14=	40000
642	040000	BIT13=	20000
643	020000	BIT12=	10000
644	010000	BIT11=	4000
645	004000	BIT10=	2000
646	002000	BIT09=	1000
647	001000	BIT08=	400
648	000400	BIT07=	200
649	000200	BIT06=	100
650	000100	BIT05=	40
651	000040	BIT04=	20
652	000020	BIT03=	10
653	000010	BIT02=	4
654	000004	BIT01=	2
655	000002	BIT00=	1
656	000001	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

668		ERRVEC=	4	.. TIME OUT AND OTHER ERRORS
669	000004	RESVEC=	10	.. RESERVED AND ILLEGAL INSTRUCTIONS
670	000010			

671	000014	TBITVEC=14	; "T" BIT
672	000014	TRTVEC= 14	; TRACE TRAP
673	000014	BPTVEC= 14	; BREAKPOINT TRAP (BPT)
674	000020	IOTVEC= 20	; INPUT/OUTPUT TRAP (IOT) **SCOPE**
675	000024	PWRVEC= 24	; POWER FAIL
676	000030	EMTVEC= 30	; EMULATOR TRAP (EMT) **ERROR**
677	000034	TRAPVEC=34	; "TRAP" TRAP
678	000060	TKVEC= 60	; TTY KEYBOARD VECTOR
679	000064	TPVEC= 64	; TTY PRINTER VECTOR
680	000240	PIRQVEC=240	; PROGRAM INTERRUPT REQUEST VECTOR

```
681                                     ; STANDARD INTERRUPT VECTORS
682
683
684                                     . =174
685 000174 000000  DISPREG 0
686 000176 000000  SWREG 0
687                                     =200
688 000200 000167 001746  JMP      START      , GO TO START OF PROGRAM
689
690
691
692                                     . =1100
693 001100 000000  . WORD 0
694 001102 177570  LIGHTS 177570
695
696
697
698                                     , PROGRAM CONTROL PARAMETERS
699
700 001104 000000  RETURN 0
701 001106 000000  NEXT   0      , ADDRESS OF NEXT TEST TO BE EXECUTED
702 001110 000000  LOCK   0      , ADDRESS FOR LOCK ON CURRENT DATA
703 001112 000000  PASCNT 0      , ADDRESS CONTAINING PASS COUNT
704 001114 000000  ERRCNT 0      , ERROR COUNT
705 001116 000000  SAVSP  0      , STACK POINTER STORAGE
706
707                                     , PROGRAM VARIABLES
708
709 001120 000020  HOLD   20     , TEMPORARY STORAGE=DELAY TIME FOR CABLES
710 001122 000000  SHIFT  0     , TEMPORARY STORAGE= # OF SHIFTS PER CHAR
711 001124 000000  COUNT  0     , TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
712 001126 000000  SAVPC  0     , PROGRAM COUNTER STORAGE
713 001130 000000  HLD0   0
714 001132 000000  HLD1   0
715 001134 000000  HLD2   0
716 001136 000000  HLD3   0
717 001140 000000  HLD4   0
718 001142 000000  HLD5   0
719 001144 000000  HLD6   0
720
```



```

721                                     ;PROGRAM CONVERSATIONAL PARAMETERS
722 001146      377      SYNCNO: .BYTE 377      ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
723 001147      377      SEXMIT .BYTE 377      ;SEC XMIT JUMPER "IN"
724 001150      377      SEREC:  .BYTE 377      ;SEC REC JUMPER "IN"
725 001151      377      OPTCLR: .BYTE 377      ;OPTIONAL JUMPER CLR "IN"
726 001152      000      MULTD:  .BYTE 0        ;NO MULTIPLE DEVICE FLAG
727 001153      377      JMRBY:  .BYTE 377      ;EXTERNAL MODEM BYPASS JUMPER "IN"
728                                     . EVEN
729
730                                     ;PROGRAM MULTIPLE DEVICE PARAMETERS
731 001154      000000    BASEADD: 0          ;PROG CONTROLLED 1ST DEVICE ADDR
732 001156      000000    KEEPADD: 0         ;SAVED 1ST DEVICE ADDR
733 001160      000000    LASTADD: 0         ;LAST DEVICE RXCSR ADDR
734 001162      000000    BASEIV:  0         ;PROG CONTROLLED IV
735 001164      000000    KEEPIV:  0         ;SAVED INTR VECTOR
736 001166      000000    ACTREG:  0         ;ACTIVE REGISTER,,,MODIFY THIS
737                                     ;LOCATION TO DISQUALIFY OR QUALIFY
738                                     ;DEVICES (1= RUN,,,0= DON'T RUN)
739 001170      000000    ROTADD:  0         ;ROTATING POINTER FOR ACTREG POINTS
740                                     ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES
741
742                                     ;PROGRAM CONTROL FLAGS
743
744 001172      000      INIFLG .BYTE 0        ;PROGRAM INITIALIZATION FLAG
745 001173      000      STFLG  .BYTE 0        ;TEST START FLAG
746 001174      000      LOKFLG .BYTE 0        ;LOCK ON CURRENT TEST FLAG
747                                     EVEN
748                                     =1400
749
750

```

```
751
752
753
754      ; INSTRUCTION DEFINITIONS
755
756      005746      PUSH1SP=5746      ; DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
757      005726      POP1SP=5726      ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
758      010046      PUSHRO=10046      ; SAVE RO ON STACK =MOV RO, -(SP)
759      012600      POPRO=12600      ; RESTORE RO FROM STACK =MOV (SP)+, RO
760      024646      PUSH2SP=24646      ; DECREMENT STACK TWICE =CMP -(SP), -(SP)
761      022626      POP2SP=22626      ; INCREMENT STACK TWICE =CMP (SP)+, (SP)+
762      ; REGISTER DEFINITIONS
763      ; RXCSR BIT DEFINITIONS
764      100000      DSC=BIT15      ; DATA SET CHANGE
765      040000      RING=BIT14      ; RING
766      020000      CTS=BIT13      ; CLR TO SEND
767      010000      CARDET=BIT12      ; CARRIER DETECT
768      004000      REACT=BIT11      ; REC ACTIVE
769      002000      SRD=BIT10      ; SEC REC DATA
770      001000      DSR=BIT9      ; DATA SET RDY
771      000400      STPSYN=BIT8      ; STRIP SYNC
772      000200      RXDONE=BIT7      ; REC DONE
773      000100      RINTEN=BIT6      ; REC INTR ENABLE
774      000040      DSINTE=BIT5      ; DSC INTR ENABLE
775      000020      SYN SCH=BIT4      ; SYNC SEARCH
776      000010      STD=BIT3      ; SEC XMIT DATA
777      000004      RTS=BIT2      ; REQ TO SEND
778      000002      DTR=BIT1      ; DATA TERM RDY
779      000001      VOID=BIT0
780      ; RXDBUF BIT DEFINITIONS
781      100000      RXERR=BIT15      ; REC ERROR
782      040000      OVERRUN=BIT14      ; OVERRUN
783      020000      FRMERR=BIT13      ; FRAME ERROR
784      010000      PARER=BIT12      ; PARITY ERROR
785      ; PARCSR BIT DEFINITIONS
786      001000      PAREN=BIT9      ; PARITY ENABLE
787      000400      EVPAR=BIT8      ; EVEN PARITY SENSE
788      ; PARCSR WRD DEFINITIONS
789      030000      SYNINT=30000      ; SYNC EXTERNAL MODE
790      020000      SYNEXT=20000      ; SYNC INTERNAL MODE
791      000000      ISYMOD=0      ; ISOC MODE
792      000000      FIVE=0      ; WORD LENGTH 5 BITS
793      002000      SIX=2000      ; WORD LENGTH 6 BITS
794      004000      SEVEN=4000      ; WORD LENGTH 7 BITS
795      006000      EIGHT=6000      ; WORD LENGTH 8 BITS
796      000000      NOPAR=0      ; NO PARITY
797      001000      ODDPAR=1000      ; ODD PARITY
798      001400      EVEPAR=1400      ; EVEN PARITY
799      ; TXCSR BIT DEFINITIONS
800      100000      DNA=BIT15      ; DATA NOT AVAILABLE
801      040000      MTDATA=BIT14      ; MAINT DATA
802      020000      CLK=BIT13      ; CLK
803      002000      BITW=BIT10      ; BIT WINDOW
804      000400      MRESET=BIT8      ; MASTER RESET
805      000200      TXDONE=BIT7      ; XMIT DONE
806      000100      TXINTE=BIT6      ; XMIT INTR ENABLE
```

807	000040	DNAINTE=BITS	;DNA INTR ENAB
808	000020	SEND=BIT4	;SEND
809	000010	HDXEN=BIT3	;HDX/FDX
810	000001	BREAK=BIT0	;BREAK
811		,TXCSR WRD DEFINITIONS	
812	000000	USER=0	;USER MODE
813	004000	MINT=4000	;MAINT INT MODE
814	010000	MEXT=10000	;MAINT EXT MODE
815	014000	SYSTST=14000	;SYSTEM TEST MODE

		SBTTL COMMON TAGS			
816					
817					
818					
819					*****
820					*,THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
821					*,USED IN THE PROGRAM.
822		001400	=		
823	001400		\$CMTAG.		:: START OF COMMON TAGS
824	001400	000000		WORD	0
825	001402	000	\$TSTNM.	BYTE	0
826	001403	000	\$ERFLG.	BYTE	0
827	001404	000000	\$ICNT.	WORD	0
828	001406	000000	\$LPADR.	WORD	0
829	001410	000000	\$LPERR.	WORD	0
830	001412	000000	\$ERTTL.	WORD	0
831	001414	000	\$ITEMB.	BYTE	0
832	001415	001	\$ERMAX.	BYTE	1
833	001416	000000	\$ERRPC.	WORD	0
834	001420	000000	\$GDADR.	WORD	0
835	001422	000000	\$BDADR.	WORD	0
836	001424	000000	\$GDDAT.	WORD	0
837	001426	000000	\$BDDAT.	WORD	0
838	001430	000000		WORD	0
839	001432	000000		WORD	0
840	001434	000	\$AUTOB.	BYTE	0
841	001435	000	\$INTAG.	BYTE	0
842	001436	000000		WORD	0
843	001440	177570	\$SWR	WORD	\$DSWR
844	001442	177570	\$DISPLAY	WORD	\$DDISP
845	001444	177560	\$TKS		
846	001446	177562	\$TKB		
847	001450	177564	\$TPS		
848	001452	177566	\$TPB		
849	001454	000	\$NULL	BYTE	0
850	001455	002	\$FILLS	BYTE	2
851	001456	012	\$FILLC	BYTE	12
852	001457	000	\$TPFLG	BYTE	0
853	001460	000000	\$REGAD	WORD	0
854					
855	001462	000000	\$REG0	WORD	0
856	001464	000000	\$REG1	WORD	0
857	001466	000000	\$REG2	WORD	0
858	001470	000000	\$REG3	WORD	0
859	001472	000000	\$REG4	WORD	0
860	001474	000000	\$REG5	WORD	0
861	001476	000000	\$STMP0	WORD	0
862	001500	000000	\$STMP1	WORD	0
863	001502	000000	\$STMP2	WORD	0
864	001504	000000	\$STMP3	WORD	0
865	001506	000000	\$STMP4	WORD	0
866	001510	000000	\$STMP5	WORD	0
867	001512	000000	\$TIMES	0	
868	001514	000000	\$ESCAPE	0	
869	001516	177607	\$BELL	ASCII	<207><377><377>
870	001522	077	\$QUES	ASCII	/?/
871	001523	015	\$CRLF	ASCII	<15>

```
872 001524 000012 $LF .ASCIZ <12> ;,LINE FEED
873 ;,*****
874 SBTTL APT MAILBOX-ETABLE
875 ;,*****
876 ;,*****
877 EVEN
878 001526 $MAIL ;,APT MAILBOX
879 001526 000000 $MSGTY: .WORD AMSGTY ;,MESSAGE TYPE CODE
880 001530 000000 $FATAL: .WORD AFATAL ;,FATAL ERROR NUMBER
881 001532 000000 $TESTN: .WORD ATESTN ;,TEST NUMBER
882 001534 000000 $PASS: .WORD APASS ;,PASS COUNT
883 001536 000000 $DEVCT: .WORD ADEVCT ;,DEVICE COUNT
884 001540 000000 $UNIT: .WORD AUNIT ;,I/O UNIT NUMBER
885 001542 000000 $MSGAD: .WORD AMSGAD ;,MESSAGE ADDRESS
886 001544 000000 $MSGLG: .WORD AMSGLG ;,MESSAGE LENGTH
887 001546 $ETABLE. ;,APT ENVIRONMENT TABLE
888 001546 000 $ENV: .BYTE AENV ;,ENVIRONMENT BYTE
889 001547 000 $ENVM: .BYTE AENVM ;,ENVIRONMENT MODE BITS
890 001550 000000 $SWREG: .WORD ASWREG ;,APT SWITCH REGISTER
891 001552 000000 $USWR: .WORD AUSWR ;,USER SWITCHES
892 001554 000000 $CPUOP: .WORD ACPUOP ;,CPU TYPE, OPTIONS
893 ;,* BITS 15-11=CPU TYPE
894 ;,* 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
895 ;,* 11/70=06, PDQ=07, Q=10
896 ;,* BIT 10=REAL TIME CLOCK
897 ;,* BIT 9=FLOATING POINT PROCESSOR
898 ;,* BIT 8=MEMORY MANAGEMENT
899 001556 000 $MAMS1: .BYTE AMAMS1 ;,HIGH ADDRESS, M. S. BYTE
900 001557 000 $MTYP1: .BYTE AMTYP1 ;,MEM TYPE, BLK#1
901 ;,* MEM TYPE BYTE -- (HIGH BYTE)
902 ;,* 900 NSEC CORE=001
903 ;,* 300 NSEC BIPOLAR=002
904 ;,* 500 NSEC MOS=003
905 001560 000000 $MADR1: .WORD AMADR1 ;,HIGH ADDRESS, BLK#1
906 ;,* MEM LAST ADDR. =3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
907 001562 000 $MAMS2: .BYTE AMAMS2 ;,HIGH ADDRESS, M. S. BYTE
908 001563 000 $MTYP2: .BYTE AMTYP2 ;,MEM. TYPE, BLK#2
909 001564 000000 $MADR2: .WORD AMADR2 ;,MEM LAST ADDRESS, BLK#2
910 001566 000 $MAMS3: .BYTE AMAMS3 ;,HIGH ADDRESS, M. S. BYTE
911 001567 000 $MTYP3: .BYTE AMTYP3 ;,MEM. TYPE, BLK#3
912 001570 000000 $MADR3: .WORD AMADR3 ;,MEM. LAST ADDRESS, BLK#3
913 001572 000 $MAMS4: .BYTE AMAMS4 ;,HIGH ADDRESS, M. S. BYTE
914 001573 000 $MTYP4: .BYTE AMTYP4 ;,MEM TYPE, BLK#4
915 001574 000000 $MADR4: .WORD AMADR4 ;,MEM. LAST ADDRESS, BLK#4
916 001576 000000 $VECT1: .WORD AVECT1 ;,INTERRUPT VECTOR#1, BUS PRIORITY#1
917 001600 000000 $VECT2: .WORD AVECT2 ;,INTERRUPT VECTOR#2BUS PRIORITY#2
918 001602 000000 $BASE: .WORD ABASE ;,BASE ADDRESS OF EQUIPMENT UNDER TEST
919 001604 000000 $DEVN: .WORD ADEVN ;,DEVICE MAP
920 001606 000000 $CDW1: .WORD ACDW1 ;,CONTROLLER DESCRIPTION WORD#1
921 001610 000000 $CDW2: .WORD ACDW2 ;,CONTROLLER DESCRIPTION WORD#2
922 001612 000000 $DDW0: .WORD ADDW0 ;,DEVICE DESCRIPTOR WORD#0
923 001614 000000 $DDW1: .WORD ADDW1 ;,DEVICE DESCRIPTOR WORD#1
924 001616 000000 $DDW2: .WORD ADDW2 ;,DEVICE DESCRIPTOR WORD#2
925 001620 000000 $DDW3: .WORD ADDW3 ;,DEVICE DESCRIPTOR WORD#3
926 001622 000000 $DDW4: .WORD ADDW4 ;,DEVICE DESCRIPTOR WORD#4
927 001624 000000 $DDW5: .WORD ADDW5 ;,DEVICE DESCRIPTOR WORD#5
```


944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

005746
005726
010046
012600
024646
022626

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

100000
040000
020000
010000

001000
000400

030000
020000
000000
000000
002000
004000
006000
000000
001000
001400

100000
040000
020000
002000
000400
000200
000100

; INSTRUCTION DEFINITIONS

PUSH1SP=5746 ; DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
POP1SP=5726 ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
PUSHRO=10046 ; SAVE RO ON STACK =MOV RO,-(SP)
POPPO=12600 ; RESTORE RO FROM STACK =MOV (SP)+,RO
PUSH2SP=24646 ; DECREMENT STACK TWICE =CMP -(SP),-(SP)
POP2SP=22626 ; INCREMENT STACK TWICE =CMP (SP)+,(SP)+

; REGISTER DEFINITIONS

; RXCSR BIT DEFINITIONS

DSC=BIT15 ; DATA SET CHANGE
RING=BIT14 ; RING
CTS=BIT13 ; CLR TO SEND
CARDET=BIT12 ; CARRIER DETECT
REACT=BIT11 ; REC ACTIVE
SRD=BIT10 ; SEC REC DATA
DSR=BIT9 ; DATA SET RDY
STPSYN=BIT8 ; STRIP SYNC
RXDONE=BIT7 ; REC DONE
RINTEN=BIT6 ; REC INTR ENABLE
DSINTE=BIT5 ; DSC INTR ENABLE
SYNSCH=BIT4 ; SYNC SEARCH
STD=BIT3 ; SEC XMIT DATA
RTS=BIT2 ; REQ TO SEND
DTR=BIT1 ; DATA TERM RDY
VOID=BIT0

; RXOBUF BIT DEFINITIONS

RXERR=BIT15 ; REC ERROR
OVRUN=BIT14 ; OVERRUN
FRMERR=BIT13 ; FRAME ERROR
PARER=BIT12 ; PARITY ERROR

; PARCSR BIT DEFINITIONS

PAREN=BIT9 ; PARITY ENABLE
EUPAR=BIT8 ; EVEN PARITY SENSE

; PARCSR WRD DEFINITIONS

SYNINT=30000 ; SYNC EXTERNAL MODE
SYNEXT=20000 ; SYNC INTERNAL MODE
ISYMOD=0 ; ISOC MODE
FIVE=0 ; WORD LENGTH 5 BITS
SIX=2000 ; WORD LENGTH 6 BITS
SEVEN=4000 ; WORD LENGTH 7 BITS
EIGHT=6000 ; WORD LENGTH 8 BITS
NOPAR=0 ; NO PARITY
ODDPAR=1000 ; ODD PARITY
EVEPAR=1400 ; EVEN PARITY

; TXCSR BIT DEFINITIONS

DNA=BIT15 ; DATA NOT AVAILABLE
MTDATA=BIT14 ; MAINT DATA
CLK=BIT13 ; CLK
BITW=BIT10 ; BIT WINDOW
MRESET=BIT8 ; MASTER RESET
TXDONE=BIT7 ; XMIT DONE
TXINTE=BIT6 ; XMIT INTR ENABLE

1000	000040	DNAINTE=BITS	;DNA INTR ENAB
1001	000020	SEND=BIT4	;SEND
1002	000010	HDXEN=BIT3	;HDX/FOX
1003	000001	BREAK=BIT0	;BREAK
1004		;TXCSR WRD DEFINITIONS	
1005	000000	USER=0	;USER MODE
1006	004000	MINT=4000	;MAINT INT MODE
1007	010000	MEXT=10000	;MAINT EXT MODE
1008	014000	SYSTST=14000	;SYSTEM TEST MODE

ERROR POINTER TABLE

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

\$ERRTB

, ERROR TABLE

Index	Item 1	Item 2	Item 3	Item 4	Description
1023	001652				
1024					
1025	001652	001762			REGISTER ERROR
1026	001654	002067			
1027	001656	002116			
1028	001660	002132			
1029	001662	002022			RECEIVER ERROR
1030	001664	002067			
1031	001666	002116			
1032	001670	002132			
1033	001672	002043			TRANSMITTER ERROR
1034	001674	002067			
1035	001676	002116			
1036	001700	002132			
1037	001702	001746			BIT ERROR (GENERAL)
1038	001704	000000			
1039	001706	002126			
1040	001710	002132			

, DEFAULT DU ADDRESSES

1041					
1042					
1043	001712	160010			RXCSR 160010
1044	001714	160011			HRXCSR 160011
1045	001716	160012			RXDDBUF 160012
1046	001720	160013			HRXDDBUF 160013
1047	001722	160012			PARCSR 160012
1048	001724	160013			HPARCSR 160013
1049	001726	160014			TXCSR 160014
1050	001730	160015			HTXCSR 160015
1051	001732	160016			TXDDBUF 160016
1052	001734	160017			HTXDDBUF 160017

, DEFAULT DU VECTORS

1053					
1054	001736	000770			DURIV 770 ,REC INTR VECTOR
1055	001740	000772			DURIS 772 ,REC INTR STATUS
1056	001742	000774			DUTIV 774 ,XMIT INTR VECTOR
1057	001744	000776			DUTIS 776 ,XMIT INTR STATUS

, ERROR MESSAGES

1058					
1059	001746	020040	051105	047522	EM4 ASCIZ / ERROR PC /
1060	001754	020122	041520	000040	
1061	001762	020040	047503	050115	EM1 ASCIZ / COMPARISON ERROR ON REGISTERS/
1062	001770	051101	051511	047117	
1063	001776	042440	051122	051117	
1064	002004	047440	020116	042522	

1065	002012	044507	052123	051105		
1066	002020	000127				
1067	002022	020040	042522	042503	EM2.	ASCIZ / RECEIVER ERROR/
1068	002030	053111	051105	042440		
1069	002036	051122	051117	000		
1070	002043	040	052040	040522	EM3:	ASCIZ / TRANSMITTER ERROR/
1071	002050	051516	044515	052124		
1072	002056	051105	042440	051122		
1073	002064	051117	000			
1074						.DATA HEADERS FOR ERROR MESSAGES
1075	002067	105	051122	041520	DH1.	ASCIZ /ERRPC WANTED ACTUAL/
1076	002074	020040	040527	052116		
1077	002102	042105	020040	041501		
1078	002110	052524	046101	000		
1079		002116				. EVEN
1080						.DATA TABLES FOR ERROR MESSAGES
1081	002116	001416	001130	001132	DT1:	WORD \$ERRPC,HLDD,HLDD1,0
1082	002124	000000				
1083						
1084	002126	001416	000000		DT4.	WORD \$ERRPC,0
1085						
1086	002132	000	000	000	DF1	BYTE 0,0,0,0
1087	002135	000				
1088						EVEN
1089						SBTTL ACT11 HOOKS
1090						
1091						*****
1092						.HOOKS REQUIRED BY ACT11
1093		002136				\$SVPC= .SAVE PC
1094		000046				=46
1095	000046	012714				\$ENDAD ..1)SET LOC 46 TO ADDRESS OF \$ENDAD IN \$EOP
1096		000052				=52
1097	000052	000000				.WORD 0 ..2)SET LOC 52 TO ZERO
1098		002136				=\$SVPC ..RESTORE PC
1099						SBTTL APT PARAMETER BLOCK
1100						
1101						*****
1102						.SET LOCATIONS 24 AND 44 HS REQUIRED FOR APT
1103						*****
1104		002136				.SX= ;;SAVE CURRENT LOCATION
1105		000024				=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1106	000024	000200				200 ;;FOR APT START UP
1107		000044				=44 ;;POINT TO APT INDIRECT ADDRESS PNTR
1108	000044	002136				\$APTHDR ;;POINT TO APT HEADER BLOCK
1109		002136				=\$X ;;RESET LOCATION COUNTER
1110						*****
1111						.SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC
1112						.INTERFACE SPEC
1113						
1114	002136					\$APTHD
1115	002136	000000				\$HIBTS WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR
1116	002140	001526				\$MBADR: WORD \$MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1117	002142	000010				\$TSTM WORD 10 ;;RUN TIM OF LONGEST TEST
1118	002144	000010				\$PASTM WORD 10 ;;RUN TIME IN SECS OF 1ST PASS ON 1 LUNIT (QUICK VERIFY)
1119	002146	000000				\$UNITM WORD ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1120	002150	000052				WORD \$ETEND-\$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

1121
1122
1123           ;PROGRAM INITIALIZATION
1124           ;LOCK OUT INTERRUPTS
1125           ;SET UP PROCESSOR STACK
1126           ;SET UP POWER FAIL VECTOR
1127           ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1128           ;TYPE TITLE MESSAGE
1129
1130 002152     .START
1131           .SBTTL INITIALIZE THE COMMON TAGS
1132           ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1133 002152 012706 001400     MOV    # $CMTAG, R6           ;; FIRST LOCATION TO BE CLEARED
1134 002156 005026           CLR    (R6)+                ;; CLEAR MEMORY LOCATION
1135 002160 022706 001440     CMP    # SWR, R6           ;; DONE?
1136 002164 001374           BNE    -6                ;; LOOP BACK IF NO
1137 002166 012706 001100     MOV    ##STACK, SP        ;; SETUP THE STACK POINTER
1138           ;; INITIALIZE A FEW VECTORS
1139 002172 012737 016340 000020     MOV    # $SCOPE, @#IOTVEC  ;; IOT VECTOR FOR SCOPE ROUTINE
1140 002200 012737 000340 000022     MOV    #340, @#IOTVEC+2  ;; LEVEL 7
1141 002206 012737 014230 000030     MOV    # $ERROR, @#EMTVEC ;; EMT VECTOR FOR ERROR ROUTINE
1142 002214 012737 000340 000032     MOV    #340, @#EMTVEC+2  ;; LEVEL 7
1143 002222 012737 016674 000034     MOV    # $TRAP, @#TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
1144 002230 012737 000340 000036     MOV    #340, @#TRAPVEC+2 ;; LEVEL 7
1145 002236 012737 015032 000024     MOV    # $PWRDN, @#PWRVEC ;; POWER FAILURE VECTOR
1146 002244 012737 000340 000026     MOV    #340, @#PWRVEC+2  ;; LEVEL 7
1147 002252 005067 177234           CLR    $TIMES            ;; INITIALIZE NUMBER OF ITERATIONS
1148 002256 005067 177232           CLR    $ESCAPE          ;; CLEAR THE ESCAPE ON ERROR ADDRESS
1149 002262 112767 000001 177125     MOVB   #1, $ERMAX        ;; ALLOW ONE ERROR PER TEST
1150 002270 012767 002270 177110     MOV    #., $LPADR        ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
1151 002276 012767 002276 177104     MOV    #., $LPERR        ;; SETUP THE ERROR LOOP ADDRESS
1152           ;; SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1153           ;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1154 002304 013746 000004           MOV    @#ERRVEC, -(SP)   ;; SAVE ERROR VECTOR
1155 002310 012737 002344 000004     MOV    #64$, @#ERRVEC   ;; SET UP ERROR VECTOR
1156 002316 012767 177570 177114     MOV    #DSWR, SWR        ;; SETUP FOR A HARDWARE SWICH REGISTER
1157 002324 012767 177570 177110     MOV    #ODISP, DISPLAY   ;; AND A HARDWARE DISPLAY REGISTER
1158 002332 022777 177777 177100     CMP    #-1, @SWR         ;; TRY TO REFERENCE HARDWARE SWR
1159 002340 001012           BNE    66$              ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
1160           ;; AND THE HARDWARE SWR IS NOT = -1
1161 002342 000403           BR     65$              ;; BRANCH IF NO TIMEOUT
1162 002344 012716 002352 64$        MOV    #65$, (SP)        ;; SET UP FOR TRAP RETURN
1163 002350 000002           RTI
1164 002352 012767 000176 177060 65$   MOV    #SWREG, SWR       ;; POINT TO SOFTWARE SWR
1165 002360 012767 000174 177054     MOV    #DISPREG, DISPLAY
1166 002366 012637 000004 66$        MOV    (SP)+, @#ERRVEC   ;; RESTORE ERROR VECTOR
1167
1168 002372 005067 177136           CLR    $PASS            ;; CLEAR PASS COUNT
1169 002376 132767 000200 177143     BITB   #APTSIZE, $ENVM   ;; TEST USER SIZE UNDER APT
1170 002404 001403           BEQ    67$              ;; YES, USE NON-APT SWITCH
1171 002406 012767 001550 177024     MOV    #SSWREG, SWR      ;; NO, USE APT SWITCH REGISTER
1172 002414           67$
1173 002414 012706 001100           MOV    #STACK, SP        .SET STACK
1174 002420 106427 000340           MTPS   #340              .LOCK INTERRUPTS
1175 002424 012737 015032 000024     MOV    # PFAIL, @#24     .SET UP POWER FA L VECTOR
1176 002432 105067 176535           CLRB   $STFLG           .CLEAR START FLAG
  
```

INITIALIZE THE COMMON TAGS

1177	002436	005067	176450			CLR	PASCNT	; CLEAR PASS COUNT
1178	002442	105067	176735			CLRB	\$ERFLG	; CLEAR ERROR FLAG
1179	002446	005067	176740			CLR	\$ERTTL	; CLEAR ERROR COUNT
1180	002452	005067	176740			CLR	\$ERRPC	; CLEAR LAST ERROR POINTER
1181	002456	012767	000001	176716		MOV	#1, \$STNM	; SET UP FOR TEST 1
1182	002464	012767	002152	176412		MOV	#. START, RETURN	; SET UP FOR POWER FAIL BEFORE
1183								; TESTING STARTS
1184	002472	013746	000006			MOV	@#6, -(SP)	
1185	002476	013746	000004			MOV	@#4, -(SP)	
1186	002502	012737	002516	000004		MOV	#15, @#4	
1187	002510	005777	176724			TST	@SWR	
1188	002514	000407				BR	2\$	
1189	002516	012767	000176	176714	1\$	MOV	#SWREG, SWR	
1190	002524	012767	000174	176710		MOV	#DISPREG, DISPLAY	
1191	002532	022626				CMP	(SP)+, (SP)+	
1192	002534	012637	000004		2\$	MOV	(SP)+, @#4	
1193	002540	012637	000006			MOV	(SP)+, @#6	
1194	002544	022767	000176	176666		CMP	#SWREG, SWR	
1195	002552	001007				BNE	3\$	
1196	002554	005737	000042			TST	@#42	; CHECK FOR CHAIN
1197	002560	001402				BEQ	33\$	
1198	002562	000167	000522			JMP	. BEGIN	
1199	002566	004767	010224		33\$	JSR	PC, CNTLU	
1200	002572	105767	176374		3\$	TSTB	INIFLG	; HAS INITIALIZATION BEEN PERFORMED
1201	002576	001004				BNE	ONCE	
1202	002600	104401	015172			TYPE	. MTITLE	; TYPE TITLE MESSAGE
1203	002604	105167	176362			COMB	INIFLG	; IF NOT SET FLAG AND DO
1204	002610	105767	176732		ONCE	TSTB	\$ENV	; APT CONTROL?
1205	002614	001410				BEQ	11\$; BR IF NO
1206	002616	032767	000001	176726		BIT	#1, \$USWR	; EXTENAL JUMPER ON?
1207	002624	001002				BNE	12\$; NO
1208	002626	105067	176321			CLRB	JMRBY	; CLEAR FLAG
1209	002632	000167	000452		12\$	JMP	. BEGIN	; GO DO IT
1210	002636	032777	000001	176574	11\$	BIT	#SW00, @SWR	; RESELECT VECTOR & CONTROL REG?
1211	002644	001002				BNE	1\$	
1212	002646	000167	000436			JMP	. BEGIN	
1213	002652	012700	000300		1\$	MOV	#300, R0	; RESTORE VECTOR AREA TO TRAPCATCHER
1214	002656	012701	000302			MOV	#302, R1	; START AT LOCATION 300
1215	002662	012702	000004			MOV	#4, R2	
1216	002666	010110			2\$	MOV	R1, (R0)	
1217	002670	005011				CLR	(R1)	
1218	002672	060200				ADD	R2, R0	
1219	002674	060201				ADD	R2, R1	
1220	002676	022701	001000			CMP	#1000, R1	; END AT LOCATION 776
1221	002702	002771				BLT	2\$	
1222	002704	104406				INSTR		; OUTPUT MESSAGE & GET INPUT STRING
1223	002706	015240				MREGAD		; MESSAGE
1224	002710	104410				PARAM		; CONVERT STRING
1225	002712	160000				160000		; LOW LIMIT
1226	002714	167776				167776		; HIGH LIMIT
1227	002716	017170				DUBASE		; STORE AT THIS LOCATION
1228	002720	001			BYTE	1		; MASK
1229	002721	001			BYTE	1		; HOW MANY TIMES + 2
1230	002722	016767	014242	176226		MOV	DUBASE, KEEPADD	; SAVE
1231	002730	004767	014102			JSR	PC, DUADDR	
1232	002734	016767	176216	176212		MOV	KEEPADD, BASEADD	; RESTORE FOR ROTATION

INITIALIZE THE COMMON TAGS

1233	002742	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1234	002744	015225				MVECTO	; MESSAGE
1235	002746	104410				PARAM	; CONVERT STRING
1236	002750	000300				300	; LOW LIMIT
1237	002752	000776				776	; HIGH LIMIT
1238	002754	001736				DURIV	; STORE AT THIS LOCATION
1239	002756	001				1	; MASK
1240	002757	004			BYTE	4	; HOW MANY TIMES + 2
1241	002760	016767	176752	176176		MOV	DURIV,KEEPIV ;SAVE
1242	002766	016767	176744	176166		MOV	DURIV,BASEIV ;SET UP FOR ROTATION
1243	002774	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1244	002776	015270				MMULT	; MESSAGE
1245	003000	104414				SETFLG	; SET FLAG BASED UPON INPUT STRING
1246	003002	001152				MULTD	; THIS FLAG
1247	003004	105767	176142			TSTB	MULTD ,ARE THERE MULTIPLE DEVICES
1248							; ON THE SYSTEM ?
1249	003010	100406				BMI	BBB ,YES,ASK NEXT QUESTION
1250	003012	005067	176150			CLR	ACTREG
1251	003016	005067	176146			CLR	ROTADD
1252	003022	000167	000140			JMP	OUTMUL ; JUMP AROUND NEXT QUESTION
1253	003026				BBB		
1254	003026	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1255	003030	015317				MLASTD	; MESSAGE
1256	003032	104410				PARAM	; CONVERT STRING
1257	003034	160000				160000	; LOW LIMIT
1258	003036	167776				167776	; HIGH LIMIT
1259	003040	001160				LASTADD	; STORE AT THIS LOCATION
1260	003042	001			BYTE	1	; MASK
1261	003043	001			BYTE	1	; HOW MANY TIMES + 2
1262							; THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1263	003044	012767	000001	176116	15	MOV	#1,ROTADD ;SET UP POINTER
1264	003052	005067	176110			CLR	ACTREG ;CLR ACTIVE REGISTER
1265	003056	056767	176106	176102	25	BIS	ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1266	003064	000241				CLC	
1267	003066	006167	176076			ROL	ROTADD ;SET UP POINTER
1268	003072	103421				BCS	35 ;ARE YOU OUT OF RANGE ?
1269	003074	062767	000010	176052		ADD	#10,BASEADD ;SET UP BASE ADDRESS
1270	003102	026767	176052	176044		CMP	LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1271	003110	101362				BHI	25 ;NO DO IT AGAIN
1272	003112	056767	176052	176046		BIS	ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1273							; LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1274							; MULTIPLE DEVICE QUESTION
1275	003120	012767	000001	176042	45	MOV	#1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1276	003126	016767	176024	176020		MOV	KEEPADD,BASEADD ;DITTO
1277	003134	000414				BR	OUTMUL ;CONTINUE QUESTIONS
1278	003136	016767	176014	176010	35	MOV	KEEPADD,BASEADD ;RESTORE
1279	003144	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1280	003146	015413				MRANGE	; MESSAGE
1281	003150	104410				PARAM	; CONVERT STRING
1282	003152	160000				160000	; LOW LIMIT
1283	003154	167776				167776	; HIGH LIMIT
1284	003156	001160				LASTADD	; STORE AT THIS LOCATION
1285	003160	001			BYTE	1	; MASK
1286	003161	001			BYTE	1	; HOW MANY TIMES + 2
1287	003162	000167	177656			JMP	15 ,DO IT AGAIN
1288	003166	012767	000340	013636	OUTMUL	MOV	#340,DUPRT

```

1289 003174 004767 013562      JSR      PC,DULEV
1290                          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1291                          ;BUFFER TO THE CHARACTERS "1" AND "2"
1292                          ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1293                          ;IF THE CHARACTER IS "2" SET THE FLAG
1294 003200                      AAA
1295 003200 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1296 003202 015631              MSYNC ;MESSAGE
1297 003204 122767 000061 012760 3$ CMPB  #'1,INBUF ;IS IT "1" ?
1298 003212 001003              BNE     1$
1299 003214 105067 175726      CLRB   SYNCNO ;000
1300 003220 000412              BR      4$
1301 003222 122767 000062 012742 1$ CMPB  #'2,INBUF ;IS IT "2" ?
1302 003230 001004              BNE     2$
1303 003232 112767 177777 175706 MOVB  #-1,SYNCNO ;377
1304 003240 000402              BR      4$
1305 003242 104407              2$   INSTR ;RETRY
1306 003244 000757              BR      3$
1307 003246 000240              4$   NOP
1308 003250 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1309 003252 015677              MWIRE6 ;MESSAGE
1310 003254 104414              SETFLG ;SET FLAG BASED UPON INPUT STRING
1311 003256 001147              SEXMIT ;THIS FLAG
1312 003260 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1313 003262 015750              MWIRE5 ;MESSAGE
1314 003264 104414              SETFLG ;SET FLAG BASED UPON INPUT STRING
1315 003266 001150              SEREC  ;THIS FLAG
1316 003270 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1317 003272 016020              MWIRE4 ;MESSAGE
1318 003274 104414              SETFLG ;SET FLAG BASED UPON INPUT STRING
1319 003276 001151              OPTCLR ;THIS FLAG
1320 003300 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1321 003302 016077              MEXTJ  ;MESSAGE
1322 003304 104414              SETFLG ;SET FLAG BASED UPON INPUT STRING
1323 003306 001153              JMRBY  ;THIS FLAG
1324
1325                          ;TEST START AND RESTART
1326
1327 003310 012706 001100      BEGIN MOV   #STACK,SP ;SET UP STACK
1328 003314 106427 000340      MTPS  #340 ;LOCK OUT INTERRUPTS
1329 003320 032777 000002 176112 BIT   #SW01,@SWR ;IF SW01=1, GET STARTING PC
1330 003326 001413              BEQ    3$
1331 003330 104406              INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1332 003332 015563              MTSTPC ;MESSAGE
1333 003334 104410              PARAM ;CONVERT STRING
1334 003336 003374              TST1  ;LOW LIMIT
1335 003340 017500              17500 ;HIGH LIMIT
1336 003342 001402              $TSTNM ;STORE AT THIS LOCATION
1337 003344 001              BYTE  1 ;MASK
1338 003345 001              BYTE  1 ;HOW MANY TIMES + 2
1339 003346 016767 176030 175530 MOV   $TSTNM,RETURN
1340 003354 000403              BR     4$
1341 003356 012767 003374 175520 3$ MOV   #TST1,RETURN ;START AT TEST 1
1342 003364 104401 015557 4$   TYPE  ,MR ;TYPE R
1343 003370 000177 175510      JMP   @RETURN ;START TESTING
1344

```

```

1345      ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1346      ; OF THE TRANSMITTER SECTION
1347      ; IT ALSO CHECKS DNA TIMING
1348      ; MODE: ISYMOD
1349      ; LENGTH: SEVEN PLUS PARITY
1350      ; PARITY: EVEPAR
1351      ; CHARACTER: 125
1352      ;
1353      ; *****
1354 003374 000004      TST1  SCOPE
1355 003376 052777 000400 176322      BIS      #MRESET,@TXCSR  ,MASTER RESET
1356 003404 012777 000000 176310      MOV      #ISYMOD,@PARCSR ,SET THE MODE
1357 003412 052777 000400 176306      BIS      #MRESET,@TXCSR  ;MASTER RESET
1358
1359      ; SET MAINTENANCE MODE & SEND
1360      ; NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1361 003420 012777 004020 176300      MOV      #MINT!SEND,@TXCSR
1362
1363      ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1364 003426 012777 005426 176266      MOV      #ISYMOD!SEVEN!EVEPAR!26,@PARCSR
1365 003434 016703 176266      MOV      TXCSR,R3      ,SET UP FOR ERROR MSG
1366 003440 112777 000125 176264      MOV      #125,@TXDBUF  ,LOAD DATA CHAR
1367 003446 012767 001252 176024      MOV      #1252,$TMP1   ,TO BE SHIFTED CHAR
1368 003454 012767 000012 175440      MOV      #10,$SHIFT    ,# OF SHIFTS
1369
1370 003462 052777 020000 176236      ; POKE CLK TO GET INTO SYNCHRONIZATION
1371 003470 042777 020000 176230      BIS      #CLK,@TXCSR   ,POKE CLK UP
1372 003476 005000      BIC      #CLK,@TXCSR   ,POKE CLK DOWN
1373 003500 006067 175774      1$      CLR      R0
1374 003504 103002      ROR      $TMP1      ; FORCE CARRY
1375 003506 052700 002000      BCC      2$      ,BR IF CARRY CLR
1376 003512      BIS      #BITW,R0     ,EQUIV OF BITW
1377 003512 052777 020000 176206      2$      BIS      #CLK,@TXCSR   ,POKE CLK UP
1378 003520 042777 020000 176200      BIC      #CLK,@TXCSR   ,POKE CLK DOWN
1379 003526 017701 176174      MOV      @TXCSR,R1     ,ACTUAL
1380 003532 042701 075777      BIC      #075777,R1    ,SAVE BITW & DNA
1381 003536 020001      CMP      R0,R1      ,COMPARE EXP VS ACT
1382 003540 001401      BEQ      +4
1383 003542 104003      ERROR   3      ,BIT WINDOW DID NOT MATCH ACTUAL DATA
1384      ; BIT, ALSO CHECK DNA
1385 003544 005367 175352      DEC      SHIFT      ,# OF SHIFTS
1386 003550 001352      BNE      1$      ,DO IT AGAIN ?
1387
1388 003552 052777 020000 176146      ; NOW POKE CLK TO SEE DNA
1389 003560 012700 000000      BIS      #CLK,@TXCSR   ,POKE CLK
1390 003564 017701 176136      MOV      #0,R0      ,EXPECTED
1391 003570 042701 077777      MOV      @TXCSR,R1     ,ACTUAL
1392 003574 020001      BIC      #77777,R1    ,SAVE DNA ONLY
1393 003576 001401      CMP      R0,R1      ,COMPARE EXP VS ACT
1394 003600 104003      BEQ      +4
1395      ERROR   3      ,DNA SHOULD BE SET
1396      ; IF DNA DID NOT SET
1397      ; CHECK WORD LENGTH SELECT LOGIC
1398
1399      ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1400      ; OF THE TRANSMITTER SECTION
1400      ; IT ALSO CHECKS DNA TIMING

```

```

1401          ;;MODE: ISYMOD
1402          ;;LENGTH: SEVEN PLUS PARITY
1403          ;;PARITY: ODDPAR
1404          ;;CHARACTER: 125
1405          ;;
1406          ,, *****
1407 003602 000004          TST2  SCOPE
1408 003604 052777 000400 176114  BIS  #MRESET,@TXCSR ;MASTER RESET
1409 003612 012777 000000 176102  MOV  #ISYMOD,@PARCSR ;SET THE MODE
1410 003620 052777 000400 176100  BIS  #MRESET,@TXCSR ;MASTER RESET
1411
1412          ,SET MAINTENANCE MODE & SEND
1413          ,NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1414 003626 012777 004020 176072  MOV  #MINT!SEND,@TXCSR
1415
1416          ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1417 003634 012777 005026 176060  MOV  #ISYMOD!SEVEN!ODDPAR!26,@PARCSR
1418 003642 016703 176060          MOV  TXCSR,R3 ;SET UP FOR ERROR MSG
1419 003646 112777 000125 176056  MOV  #125,@TXDBUF ;LOAD DATA CHAR
1420 003654 012767 001652 175616  MOV  #1652,$TMP1 ;TO BE SHIFTED CHAR
1421 003662 012767 000012 175232  MOV  #10,SHIFT ;# OF SHIFTS
1422          ,POKE CLK TO GET INTO SYNCHRONIZATION
1423 003670 052777 020000 176030  BIS  #CLK,@TXCSR ;POKE CLK UP
1424 003676 042777 020000 176022  BIC  #CLK,@TXCSR ;POKE CLK DOWN
1425 003704 005000          1$ CLR  R0
1426 003706 006067 175566          ROR  $TMP1 ;FORCE CARRY
1427 003712 103002          BCC  2$ ,BR IF CARRY CLR
1428 003714 052700 002000          BIS  #BITW,R0 ;EQUIV OF BITW
1429 003720          2$
1430 003720 052777 020000 176000  BIS  #CLK,@TXCSR ;POKE CLK UP
1431 003726 042777 020000 175772  BIC  #CLK,@TXCSR ;POKE CLK DOWN
1432 003734 017701 175766          MOV  @TXCSR,R1 ;ACTUAL
1433 003740 042701 075777          BIC  #075777,R1 ;SAVE BITW & DNA
1434 003744 020001          CMP  R0,R1 ;COMPARE EXP VS ACT
1435 003746 001401          BEQ  +4
1436 003750 104003          ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1437          ;BIT, ALSO CHECK DNA
1438 003752 005367 175144          DEC  SHIFT ;# OF SHIFTS
1439 003756 001352          BNE  1$ ;DO IT AGAIN ?
1440          ,NOW POKE CLK TO SEE DNA
1441 003760 052777 020000 175740  BIS  #CLK,@TXCSR ;POKE CLK
1442 003766 012700 000000          MOV  #0,R0 ;EXPECTED
1443 003772 017701 175730          MOV  @TXCSR,R1 ;ACTUAL
1444 003776 042701 077777          BIC  #77777,R1 ;SAVE DNA ONLY
1445 004002 020001          CMP  R0,R1 ;COMPARE EXP VS ACT
1446 004004 001401          BEQ  +4
1447 004006 104003          ERROR 3 ;DNA SHOULD BE SET
1448          ;IF DNA DID NOT SET
1449          ;CHECK WORD LENGTH SELECT LOGIC
1450
1451          ,, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1452          ,, OF THE TRANSMITTER SECTION
1453          ,, IT ALSO CHECKS DNA TIMING
1454          ,, MODE SYNINT
1455          ,, LENGTH EIGHT PLUS PARITY
1456          ,, PARITY EVEPAR
  
```



```
1457          ;; CHARACTER: 125
1458          ;;
1459          ;; *****
1460 004010 000004          TST3: SCOPE
1461 004012 052777 000400 175706      BIS      #MRESET,@TXCSR ; MASTER RESET
1462 004020 012777 030000 175674      MOV      #SYNINT,@PARCSR ; SET THE MODE
1463 004026 052777 000400 175672      BIS      #MRESET,@TXCSR ; MASTER RESET
1464
1465          ; SET MAINTENANCE MODE & SEND
1466          , NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1467 004034 012777 004020 175664      MOV      #MINT!SEND,@TXCSR
1468
1469          ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1470 004042 012777 037426 175652      MOV      #SYNINT!EIGHT!EVEPAR!26,@PARCSR
1471 004050 016703 175652          MOV      TXCSR,R3          ; SET UP FOR ERROR MSG
1472 004054 112777 000125 175650      MOV      #125,@TXDBUF    ; LOAD DATA CHAR
1473 004062 012767 000125 175410      MOV      #125,$TMP1     ; TO BE SHIFTED CHAR
1474 004070 012767 000011 175024      MOV      #9,SHIFT       ; # OF SHIFTS
1475          ; POKE CLK TO GET INTO SYNCHRONIZATION
1476 004076 052777 020000 175622      BIS      #CLK,@TXCSR    ; POKE CLK UP
1477 004104 042777 020000 175614      BIC      #CLK,@TXCSR    ; POKE CLK DOWN
1478 004112 005000          1$ CLR      R0
1479 004114 006067 175360          ROR      $TMP1          ; FORCE CARRY
1480 004120 103002          BCC      2$            ; BR IF CARRY CLR
1481 004122 052700 002000          BIS      #BITW,R0       ; EQUIV OF BITW
1482 004126          2$
1483 004126 052777 020000 175572      BIS      #CLK,@TXCSR    ; POKE CLK UP
1484 004134 042777 020000 175564      BIC      #CLK,@TXCSR    ; POKE CLK DOWN
1485 004142 017701 175560          MOV      @TXCSR,R1      ; ACTUAL
1486 004146 042701 075777          BIC      #075777,R1     ; SAVE BITW & DNA
1487 004152 020001          CMP      R0,R1          ; COMPARE EXP VS ACT
1488 004154 001401          BEQ     +4
1489 004156 104003          ERROR  3            ; BIT WINDOW DID NOT MATCH ACTUAL DATA
1490          ; BIT... ALSO CHECK DNA
1491 004160 005367 174736          DEC     SHIFT          ; # OF SHIFTS
1492 004164 001352          BNE     1$            ; DO IT AGAIN ?
1493          ; NOW POKE CLK TO SEE DNA
1494 004166 052777 020000 175532      BIS      #CLK,@TXCSR    ; POKE CLK
1495 004174 012700 100000          MOV      #100000,R0     ; EXPECTED
1496 004200 017701 175522          MOV      @TXCSR,R1      ; ACTUAL
1497 004204 042701 077777          BIC      #77777,R1      ; SAVE DNA ONLY
1498 004210 020001          CMP      R0,R1          ; COMPARE EXP VS ACT
1499 004212 001401          BEQ     +4
1500 004214 104003          ERROR  3            ; DNA SHOULD BE SET
1501          ; IF DNA DID NOT SET
1502          ; CHECK WORD LENGTH SELECT LOGIC
1503
1504          ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1505          ; OF THE TRANSMITTER SECTION.
1506          ; IT ALSO CHECKS DNA TIMING
1507          ; MODE: SYNINT
1508          ; LENGTH: EIGHT PLUS PARITY
1509          ; PARITY ODDPAR
1510          ; CHARACTER: 125
1511          ;;
1512          ;; *****
```

```
1513 004216 000004 TST4 SCOPE
1514 004220 052777 000400 175500 BIS #MRESET,@TXCSR ;MASTER RESET
1515 004226 012777 030000 175466 MOV #SYNINT,@PARCSR ;SET THE MODE
1516 004234 052777 000400 175464 BIS #MRESET,@TXCSR ;MASTER RESET
1517
1518 ;SET MAINTENANCE MODE & SEND
1519 ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1520 004242 012777 004020 175456 MOV #MINT'SEND,@TXCSR
1521
1522 ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1523 004250 012777 037026 175444 MOV #SYNINT!EIGHT!ODDPAR!26,@PARCSR
1524 004256 016703 175444 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1525 004262 112777 000125 175442 MOVB #125,@TXDBUF ;LOAD DATA CHAR
1526 004270 012767 000525 175202 MOV #525,$TMP1 ;TO BE SHIFTED CHAR
1527 004276 012767 000011 174616 MOV #9,SHIFT ;# OF SHIFTS
1528 ;POKE CLK TO GET INTO SYNCHRONIZATION
1529 004304 052777 020000 175414 BIS #CLK,@TXCSR ;POKE CLK UP
1530 004312 042777 020000 175406 BIC #CLK,@TXCSR ;POKE CLK DOWN
1531 004320 005000 15 CLR R0
1532 004322 006067 175152 ROR $TMP1 ;FORCE CARRY
1533 004326 103002 BCC 2$ ,BR IF CARRY CLR
1534 004330 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
1535 004334 2$
1536 004334 052777 020000 175364 BIS #CLK,@TXCSR ;POKE CLK UP
1537 004342 042777 020000 175356 BIC #CLK,@TXCSR ;POKE CLK DOWN
1538 004350 017701 175352 MOV @TXCSR,R1 ;ACTUAL
1539 004354 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1540 004360 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1541 004362 001401 BEQ +4
1542 004364 104003 ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1543 ;BIT,... ALSO CHECK DNA
1544 004366 005367 174530 DEC SHIFT ;# OF SHIFTS
1545 004372 001352 BNE 1$ ,DO IT AGAIN ?
1546 ;NOW POKE CLK TO SEE DNA
1547 004374 052777 020000 175324 BIS #CLK,@TXCSR ;POKE CLK
1548 004402 012700 100000 MOV #100000,R0 ;EXPECTED
1549 004406 017701 175314 MOV @TXCSR,R1 ;ACTUAL
1550 004412 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1551 004416 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1552 004420 001401 BEQ +4
1553 004422 104003 ERROR 3 ;DNA SHOULD BE SET
1554 ;IF DNA DID NOT SET
1555 ;CHECK WORD LENGTH SELECT LOGIC
1556
1557 ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1558 ;, OF THE TRANSMITTER SECTION
1559 ;, IT ALSO CHECKS DNA TIMING
1560 ;, MODE: ISYMOD
1561 ;, LENGTH: EIGHT PLUS PARITY
1562 ;, PARITY: EVEPAR
1563 ;, CHARACTER 125
1564 ;,
1565 ;, *****
1566 004424 000004 TST5 SCOPE
1567 004426 052777 000400 175272 BIS #MRESET,@TXCSR ;MASTER RESET
1568 004434 012777 000000 175260 MOV #ISYMOD,@PARCSR ;SET THE MODE
```

```

1569 004442 052777 000400 175256      BIS      #MRESET,@TXCSR ,MASTER RESET
1570
1571      ;SET MAINTENANCE MODE & SEND
1572      ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1573 004450 012777 004020 175250      MOV      #MINT!SEND,@TXCSR
1574
1575      ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1576 004456 012777 007426 175236      MOV      #ISYMOD!EIGHT!EVEPAR!26,@PARCSR
1577 004464 016703 175236      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
1578 004470 112777 000125 175234      MOV      #125,@TXDBUF  ;LOAD DATA CHAR
1579 004476 012767 002252 174774      MOV      #2252,$TMP1   ;TO BE SHIFTED CHAR
1580 004504 012767 000013 174410      MOV      #11,SHIFT     ;# OF SHIFTS
1581
1582 004512 052777 020000 175206      .POKE CLK TO GET INTO SYNCHRONIZATION
1583 004520 042777 020000 175200      BIS      #CLK,@TXCSR   ,POKE CLK UP
1584 004526 005000      CLR      R0            ;POKE CLK DOWN
1585 004530 006067 174744      ROR      $TMP1 ,FORCE CARRY
1586 004534 103002      BCC      2$           ,BR IF CARRY CLR
1587 004536 052700 002000      BIS      #BITW,R0      ,EQUIV OF BITW
1588 004542
1589 004542 052777 020000 175156      BIS      #CLK,@TXCSR   ,POKE CLK UP
1590 004550 042777 020000 175150      BIC      #CLK,@TXCSR   ,POKE CLK DOWN
1591 004556 017701 175144      MOV      @TXCSR,R1     ,ACTUAL
1592 004562 042701 075777      BIC      #075777,R1    ,SAVE BITW & DNA
1593 004566 020001      CMP      R0,R1        ,COMPARE EXP VS ACT
1594 004570 001401      BEQ      +4
1595 004572 104003      ERROR   3            ,BIT WINDOW DID NOT MATCH ACTUAL DATA
1596
1597 004574 005367 174322      DEC      SHIFT        ,# OF SHIFTS
1598 004600 001352      BNE      1$           ,DO IT AGAIN ?
1599
1600 004602 052777 020000 175116      .NOW POKE CLK TO SEE DNA
1601 004610 012700 000000      BIS      #CLK,@TXCSP   ,POKE CLK
1602 004614 017701 175106      MOV      #0,R0        ,EXPECTED
1603 004620 042701 077777      MOV      @TXCSR,R1     ,ACTUAL
1604 004624 020001      BIC      #77777,R1    ,SAVE DNA ONLY
1605 004626 001401      CMP      R0,R1        ,COMPARE EXP VS ACT
1606 004630 104003      BEQ      +4
1607      ERROR   3            DNA SHOULD BE SET
1608      ,IF DNA DID NOT SET
1609      ,CHECK WORD LENGTH SELECT LOGIC
1610
1611      ,, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1612      ,, OF THE TRANSMITTER SECTION
1613      ,, IT ALSO CHECKS DNA TIMING
1614      ,, MODE ISYMOD
1615      ,, LENGTH EIGHT PLUS PARITY
1616      ,, PARITY JDDPAR
1617      ,, CHARACTER: 125
1618      ,,
1619 004632 000004      TST6   SCOPE
1620 004634 052777 000400 175064      BIS      #MRESET,@TXCSR ,MASTER RESET
1621 004642 012777 000000 175052      MOV      #ISYMOD,@PARCSR ,SET THE MODE
1622 004650 052777 000400 175050      BIS      #MRESET,@TXCSR ,MASTER RESET
1623
1624      ,SET MAINTENANCE MODE & SEND

```

```

1625                                     ,NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1626 004656 012777 004020 175042      MOV      #MINT!SEND,@TXCSR
1627
1628                                     ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1629 004664 012777 007026 175030      MOV      #1SYMOD!EIGHT!ODDPAR!26,@PARCSR
1630 004672 016703 175030              MOV      TXCSR,R3          ;SET UP FOR ERROR MSG
1631 004676 112777 000125 175026      MOV      #125,@TXDBUF     ;LOAD DATA CHAR
1632 004704 012767 003252 174566      MOV      #3252,$TMP1      ;TO BE SHIFTED CHAR
1633 004712 012767 000013 174202      MOV      #11,$SHIFT       ;# OF SHIFTS
1634                                     ,POKE CLK TO GET INTO SYNCHRONIZATION
1635 004720 052777 020000 175000      BIS      #CLK,@TXCSR      ;POKE CLK UP
1636 004726 042777 020000 174772      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1637 004734 005000                      CLR      R0
1638 004736 006067 174536              ROR      $TMP1           ;FORCE CARRY
1639 004742 103002                      BCC      $S              ;BR IF CARRY CLR
1640 004744 052700 002000              BIS      #BITW,R0         ;EQUIV OF BITW
1641                                     25
1642 004750 052777 020000 174750      BIS      #CLK,@TXCSR      ;POKE CLK UP
1643 004756 042777 020000 174742      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
1644 004764 017701 174736              MOV      @TXCSR,R1        ;ACTUAL
1645 004770 042701 075777              BIC      #075777,R1       ;SAVE BITW & DNA
1646 004774 020001                      CMP      R0,R1           ;COMPARE EXP VS ACT
1647 004776 001401                      BEQ      +4
1648 005000 104003                      ERROR    3               ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1649                                     ;BIT, ALSO CHECK DNA
1650 005002 005367 174114              DEC      SHIFT           ;# OF SHIFTS
1651 005006 001352                      BNE      $S              ;DO IT AGAIN ?
1652                                     ,NOW POKE CLK TO SEE DNA
1653 005010 052777 020000 174710      BIS      #CLK,@TXCSR      ;POKE CLK
1654 005016 012700 000000              MOV      #0,R0           ;EXPECTED
1655 005022 017701 174700              MOV      @TXCSR,R1        ;ACTUAL
1656 005026 042701 077777              BIC      #77777,R1       ;SAVE DNA ONLY
1657 005032 020001                      CMP      R0,R1           ;COMPARE EXP VS ACT
1658 005034 001401                      BEQ      +4
1659 005036 104003                      ERROR    3               ;DNA SHOULD BE SET
1660                                     ;IF DNA DID NOT SET
1661                                     ;CHECK WORD LENGTH SELECT LOGIC
1662
1663
1664                                     ,, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1665                                     ,, OF THE TRANSMITTER SECTION
1666                                     ,, IT ALSO CHECKS DNA TIMING
1667                                     ,, MODE. SYNINT
1668                                     ,, LENGTH. EIGHT PLUS PARITY
1669                                     ,, PARITY. EVEPAR
1670                                     ,, CHARACTER 252
1671                                     ;
1672                                     ,, *****
1673 005040 000004                      TST7    SCOPE
1674 005042 052777 000400 174656      BIS      #MRESET,@TXCSR  ;MASTER RESET
1675 005050 012777 030000 174644      MOV      #SYNINT,@PARCSR ;SET THE MODE
1676 005056 052777 000400 174642      BIS      #MRESET,@TXCSR  ;MASTER RESET
1677
1678                                     ,SET MAINTENANCE MODE & SEND
1679                                     ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1680 005064 012777 004020 174634      MOV      #MINT!SEND,@TXCSR
    
```

```

1681
1682      ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1683      005072 012777 037426 174622      MOV      #SYNINT'EIGHT'EVEPAR'26, @PARCSR
1684      005100 016703 174622      MOV      TXCSR, R3      ,SET UP FOR ERROR MSG
1685      005104 112777 000252 174620      MOV      #252, @TXDBUF      ;LOAD DATA CHAR
1686      005112 012767 000252 174360      MOV      #252, $TMP1      ;TO BE SHIFTED CHAR
1687      005120 012767 000011 173774      MOV      #9, SHIFT      ;# OF SHIFTS
1688      ;POKE CLK TO GET INTO SYNCHRONIZATION
1689      005126 052777 020000 174572      BIS      #CLK, @TXCSR      ;POKE CLK UP
1690      005134 042777 020000 174564      BIC      #CLK, @TXCSR      ;POKE CLK DOWN
1691      005142 005000      15      CLR      R0
1692      005144 006067 174330      ROR      $TMP1      ,FOR 'E CARRY
1693      005150 103002      BCC      25      ,BR 'F CARRY CLR
1694      005152 052700 002000      BIS      #BITW, R0      EQUIV OF BITW
1695      005156      25
1696      005156 052777 020000 174542      BIS      #CLK, @TXCSR      ,POKE CLK UP
1697      005164 042777 020000 174534      BIC      #CLK, @TXCSR      ,POKE CLK DOWN
1698      005172 017701 174530      MOV      @TXCSR, R1      ,ACTUAL
1699      005176 042701 075777      BIC      #075777, R1      ;SAVE BITW & DNA
1700      005202 020001      CMP      R0, R1      ,COMPARE EXP VS ACT
1701      005204 001401      BEQ      +4
1702      005206 104003      ERROR    3      ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1703      ;BIT, ALSO CHECK DNA
1704      005210 005367 173706      DEC      SHIFT      ,# OF SHIFTS
1705      005214 001352      BNE      15      ,DO IT AGAIN ?
1706      ;NOW POKE CLK TO SEE DNA
1707      005216 052777 020000 174502      BIS      #CLK, @TXCSR      ,POKE CLK
1708      005224 012700 100000      MOV      #100000, R0      ,EXPECTED
1709      005230 017701 174472      MOV      @TXCSR, R1      ,ACTUAL
1710      005234 042701 077777      BIC      #77777, R1      ,SAVE DNA ONLY
1711      005240 020001      CMP      R0, R1      ,COMPARE EXP VS ACT
1712      005242 001401      BEQ      +4
1713      005244 104003      ERROR    3      ,DNA SHOULD BE SET
1714      ;IF DNA DID NOT SET
1715      ;CHECK WORD LENGTH SELECT LOGIC
1716
1717      ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1718      ;, OF THE TRANSMITTER SECTION
1719      ;, IT ALSO CHECKS DNA TIMING
1720      ;, MODE: SYNINT
1721      ;, LENGTH: EIGHT PLUS PARITY
1722      ;, PARITY: ODDPAR
1723      ;, CHARACTER: 252
1724      ;,
1725      ;, *****
1726      005246 000004      TSTIO    SCOPE
1727      005250 052777 000400 174450      BIS      #MRESET, @TXCSR      ,MASTER RESET
1728      005256 012777 030000 174436      MOV      #SYNINT, @PARCSR      ,SET THE MODE
1729      005264 052777 000400 174434      BIS      #MRESET, @TXCSR      ;MASTER RESET
1730
1731      ;SET MAINTENANCE MODE & SEND
1732      ;NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1733      005272 012777 004020 174426      MOV      #MINT!SEND, @TXCSR
1734
1735      ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1736      005300 012777 037026 174414      MOV      #SYNINT'EIGHT'ODDPAR'26, @PARCSR
  
```

INITIALIZE THE COMMON TAGS

```

1737 005306 016703 174414          MOV    TXCSR,R3          ;SET UP FOR ERROR MSG
1738 005312 112777 000252 174412    MOVB   #252,@TXDBUF     ;LOAD DATA CHAR
1739 005320 012767 000652 174152    MOV    #652,$TMP1      ;TO BE SHIFTED CHAR
1740 005326 012767 000011 173566    MOV    #9,SHIFT        ;# OF SHIFTS
1741                                     ,POKE CLK TO GET INTO SYNCHRONIZATION
1742 005334 052777 020000 174364    BIS    #CLK,@TXCSR     ;POKE CLK UP
1743 005342 042777 020000 174356    BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1744 005350 005000                                     15    CLR    R0
1745 005352 006067 174122                                     ROR    $TMP1          ;FORCE CARRY
1746 005356 103002                                     BCC    25             ;BR IF CARRY CLR
1747 005360 052700 002000                                     BIS    #BITW,R0       ;EQUIV OF BITW
1748 005364                                     25
1749 005364 052777 020000 174334    BIS    #CLK,@TXCSR     ;POKE CLK UP
1750 005372 042777 020000 174326    BIC    #CLK,@TXCSR     ;POKE CLK DOWN
1751 005400 017701 174322          MOV    @TXCSR,R1        ;ACTUAL
1752 005404 042701 075777          BIC    #075777,R1      ;SAVE BITW & DNA
1753 005410 020001          CMP    R0,R1           ;COMPARE EXP VS ACT
1754 005412 001401          BEQ    +4
1755 005414 104003          ERROR  3              ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1756                                     ;BIT,... ALSO CHECK DNA
1757 005416 005367 173500          DEC    SHIFT          ;# OF SHIFTS
1758 005422 001352          BNE    15             ;DO IT AGAIN ?
1759                                     ,NOW POKE CLK TO SEE DNA
1760 005424 052777 020000 174274    BIS    #CLK,@TXCSR     ;POKE CLK
1761 005432 012700 100000          MOV    #100000,R0      ;EXPECTED
1762 005436 017701 174264          MOV    @TXCSR,R1        ;ACTUAL
1763 005442 042701 077777          BIC    #77777,R1       ;SAVE DNA ONLY
1764 005446 020001          CMP    R0,R1           ;COMPARE EXP VS ACT
1765 005450 001401          BEQ    +4
1766 005452 104003          ERROR  3              ;DNA SHOULD BE SET
1767                                     ;IF DNA DID NOT SET
1768                                     ;CHECK WORD LENGTH SELECT LOGIC
1769
1770                                     ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1771                                     ;, OF THE TRANSMITTER SECTION
1772                                     ;, IT ALSO CHECKS DNA TIMING
1773                                     ;, MODE: SYNINT
1774                                     ;, LENGTH: EIGHT PLUS PARITY
1775                                     ;, PARITY: EVEPAR
1776                                     ;, CHARACTER 0
1777                                     ;,
1778                                     ;, *****
1779 005454 000004          TST11  SCOPE
1780 005456 052777 000400 174242    BIS    #MRESET,@TXCSR ;MASTER RESET
1781 005464 012777 030000 174230    MOV    #SYNINT,@PARCSR ;SET THE MODE
1782 005472 052777 000400 174226    BIS    #MRESET,@TXCSR ;MASTER RESET
1783
1784                                     ,SET MAINTENANCE MODE & SEND
1785                                     ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1786 005500 012777 004020 174220    MOV    #MINT$SEND,@TXCSR
1787
1788                                     ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1789 005506 012777 037426 174206    MOV    #SYNINT'EIGHT'EVEPAR'26,@PARCSR
1790 005514 016703 174206          MOV    TXCSR,R3        ;SET UP FOR ERROR MSG
1791 005520 112777 000000 174204    MOVB   #0,@TXDBUF     ;LOAD DATA CHAR
1792 005526 012767 000000 173744    MOV    #0,$TMP1       ;TO BE SHIFTED CHAR
    
```

```

1793 005534 012767 000011 173360      MOV      #9,SHIFT      ;# OF SHIFTS
1794                                ;POKE CLK TO GET INTO SYNCHRONIZATION
1795 005542 052777 020000 174156      BIS      #CLK,@TXCSR   ;POKE CLK UP
1796 005550 042777 020000 174150      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
1797 005556 005000                                15:     CLR      RO
1798 005560 006067 173714      ROR      $TMP1        ;FORCE CARRY
1799 005564 103002      BCC      25          ;BR IF CARRY CLR
1800 005566 052700 002000      BIS      #BITW,RO     ;EQUIV OF BITW
1801 005572                                25:
1802 005572 052777 020000 174126      BIS      #CLK,@TXCSR   ;POKE CLK UP
1803 005600 042777 020000 174120      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
1804 005606 017701 174114      MOV      @TXCSR,R1     ;ACTUAL
1805 005612 042701 075777      BIC      #075777,R1    ;SAVE BITW & DNA
1806 005616 020001      CMP      RO,R1        ;COMPARE EXP VS ACT
1807 005620 001401      BEQ      +4
1808 005622 104003      ERROR   3            ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1809                                ;BIT,... ALSO CHECK DNA
1810 005624 005367 173272      DEC      SHIFT        ;# OF SHIFTS
1811 005630 001352      BNE      15          ;DO IT AGAIN ?
1812                                ;NOW POKE CLK TO SEE DNA
1813 005632 052777 020000 174066      BIS      #CLK,@TXCSR   ;POKE CLK
1814 005640 012700 100000      MOV      #100000,RO    ;EXPECTED
1815 005644 017701 174056      MOV      @TXCSR,R1     ;ACTUAL
1816 005650 042701 077777      BIC      #77777,R1     ;SAVE DNA ONLY
1817 005654 020001      CMP      RO,R1        ;COMPARE EXP VS ACT
1818 005656 001401      BEQ      +4
1819 005660 104003      ERROR   3            ;DNA SHOULD BE SET
1820                                ;IF DNA DID NOT SET
1821                                ;CHECK WORD LENGTH SELECT LOGIC
1822
1823                                ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1824                                ;, OF THE TRANSMITTER SECTION
1825                                ;, IT ALSO CHECKS DNA TIMING
1826                                ;, MODE: SYNINT
1827                                ;, LENGTH: EIGHT PLUS PARITY
1828                                ;, PARITY ODDPAR
1829                                ;, CHARACTER: 0
1830                                ;,
1831                                ;, *****
1832 005662 000004      TST12: SCOPE
1833 005664 052777 000400 174034      BIS      #MRESET,@TXCSR ;MASTER RESET
1834 005672 012777 030000 174022      MOV      #SYNINT,@PARCSR ;SET THE MODE
1835 005700 052777 000400 174020      BIS      #MRESET,@TXCSR ;MASTER RESET
1836
1837                                ;SET MAINTENANCE MODE & SEND
1838                                ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1839 005706 012777 004020 174012      MOV      #MINT$SEND,@TXCSR
1840
1841                                ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1842 005714 012777 037026 174000      MOV      #SYNINT$EIGHT$ODDPAR$26,@PARCSR
1843 005722 016703 174000      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
1844 005726 112777 000000 173776      MOVB    #0,@TXDBUF     ;LOAD DATA CHAR
1845 005734 012767 000400 173536      MOV      #400,$TMP1    ;TO BE SHIFTED CHAR
1846 005742 012767 000011 173152      MOV      #9,SHIFT     ;# OF SHIFTS
1847                                ;POKE CLK TO GET INTO SYNCHRONIZATION
1848 005750 052777 020000 173750      BIS      #CLK,@TXCSR   ;POKE CLK UP
    
```

```

1849 005756 042777 020000 173742      BIC    #CLK,@TXCSR    ,POKE CLK DOWN
1850 005764 005000                    15    CLR    RO
1851 005766 006067 173506                ROR    $TMP1    ,FORCE CARRY
1852 005772 103002                    BCC    2$      ,BR IF CARRY CLR
1853 005774 052700 002000                BIS    #BITW,RO    ,EQUIV OF BITW
1854 006000                                25
1855 006000 052777 020000 173720      BIS    #CLK,@TXCSR    ;POKE CLK UP
1856 006006 042777 020000 173712      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1857 006014 017701 173706                MOV    @TXCSR,R1     ,ACTUAL
1858 006020 042701 075777                BIC    #075777,R1    ;SAVE BITW & DNA
1859 006024 020001                    CMP    RO,R1        ,COMPARE EXP VS ACT
1860 006026 001401                    BEQ    .+4
1861 006030 104003                    ERROR  3            ,BIT WINDOW DID NOT MATCH ACTUAL DATA
1862                                ;BIT, ... ALSO CHECK DNA
1863 006032 005367 173064                DEC    SHIFT        ,# OF SHIFTS
1864 006036 001352                    BNE    1$          ;DO IT AGAIN ?
1865                                ;NOW POKE CLK TO SEE DNA
1866 006040 052777 020000 173660      BIS    #CLK,@TXCSR    ;POKE CLK
1867 006046 012700 100000                MOV    #100000,RO    ,EXPECTED
1868 006052 017701 173650                MOV    @TXCSR,R1     ,ACTUAL
1869 006056 042701 077777                BIC    #77777,R1    ,SAVE DNA ONLY
1870 006062 020001                    CMP    RO,R1        ,COMPARE EXP VS ACT
1871 006064 001401                    BEQ    .+4
1872 006066 104003                    ERROR  3            ,DNA SHOULD BE SET
1873                                ;IF DNA DID NOT SET
1874                                ;CHECK WORD LENGTH SELECT LOGIC
1875
1876                                ;; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1877                                ;; OF THE TRANSMITTER SECTION
1878                                ;; IT ALSO CHECKS DNA TIMING
1879                                ;; MODE: SYNINT
1880                                ;; LENGTH: EIGHT PLUS PARITY
1881                                ;; PARITY: EVEPAR
1882                                ;; CHARACTER: 377
1883                                ;;
1884                                ;; *****
1885 006070 000004                    TST13: SCOPE
1886 006072 052777 000400 173626      BIS    #MRESET,@TXCSR ;MASTER RESET
1887 006100 012777 030000 173614      MOV    #SYNINT,@PARCSR ;SET THE MODE
1888 006106 052777 000400 173612      BIS    #MRESET,@TXCSR ;MASTER RESET
1889
1890                                ;SET MAINTENANCE MODE & SEND
1891                                ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1892 006114 012777 004020 173604      MOV    #MINT!SEND,@TXCSR
1893
1894                                ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1895 006122 012777 037426 173572      MOV    #SYNINT!EIGHT!EVEPAR!26,@PARCSR
1896 006130 016703 173572                MOV    TXCSR,R3     ;SET UP FOR ERROR MSG
1897 006134 112777 000377 173570      MOVB  #377,@TXDBUF   ,LOAD DATA CHAR
1898 006142 012767 000377 173330      MOV    #377,$TMP1    ,TO BE SHIFTED CHAR
1899 006150 012767 000011 172744      MOV    #9,SHIFT     ,# OF SHIFTS
1900                                ;POKE CLK TO GET INTO SYNCHRONIZATION
1901 006156 052777 020000 173542      BIS    #CLK,@TXCSR    ,POKE CLK UP
1902 006164 042777 020000 173534      BIC    #CLK,@TXCSR    ,POKE CLK DOWN
1903 006172 005000                    15    CLR    RO
1904 006174 006067 173300                ROR    $TMP1    ,FORCE CARRY
  
```


INITIALIZE THE COMMON TAGS

```
1905 006200 103002          BCC 2$ ;BR IF CARRY CLR
1906 006202 052700 002000  BIS  #BITW,RO ;EQUIV OF BITW
1907 006206                2$
1908 006206 052777 020000 173512  BIS  #CLK,@TXCSR ;POKE CLK UP
1909 006214 042777 020000 173504  BIC  #CLK,@TXCSR ;POKE CLK DOWN
1910 006222 017701 173500      MOV  @TXCSR,R1 ;ACTUAL
1911 006226 042701 075777      BIC  #075777,R1 ;SAVE BITW & DNA
1912 006232 020001          CMP  RO,R1 ;COMPARE EXP VS ACT
1913 006234 001401          BEQ  +4
1914 006236 104003          ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1915                                ;BIT,... ALSO CHECK DNA
1916 006240 005367 172656      DEC  SHIFT ;# OF SHIFTS
1917 006244 001352          BNE  1$ ;DO IT AGAIN ?
1918                                ;NOW POKE CLK TO SEE DNA
1919 006246 052777 020000 173452  BIS  #CLK,@TXCSR ;POKE CLK
1920 006254 012700 100000      MOV  #100000,RO ;EXPECTED
1921 006260 017701 173442      MOV  @TXCSR,R1 ;ACTUAL
1922 006264 042701 077777      BIC  #77777,R1 ;SAVE DNA ONLY
1923 006270 020001          CMP  RO,R1 ;COMPARE EXP VS ACT
1924 006272 001401          BEQ  +4
1925 006274 104003          ERROR 3 ;DNA SHOULD BE SET
1926                                ;IF DNA DID NOT SET
1927                                ;CHECK WORD LENGTH SELECT LOGIC
1928
1929                                ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
1930                                ;, OF THE TRANSMITTER SECTION
1931                                ;, IT ALSO CHECKS DNA TIMING
1932                                ;, MODE. SYNINT
1933                                ;, LENGTH: EIGHT PLUS PARITY
1934                                ;, PARITY: ODDPAR
1935                                ;, CHARACTER 377
1936                                ;,
1937                                ;, *****
1938 006276 000004          TST14. SCOPE
1939 006300 052777 000400 173420  BIS  #MRESET,@TXCSR ;MASTER RESET
1940 006306 012777 030000 173406  MOV  #SYNINT,@PARCSR ;SET THE MODE
1941 006314 052777 000400 173404  BIS  #MRESET,@TXCSR ;MASTER RESET
1942
1943                                ; SET MAINTENANCE MODE & SEND
1944                                ; NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1945 006322 012777 004020 173376  MOV  #MINT!SEND,@TXCSR
1946
1947                                ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1948 006330 012777 037026 173364  MOV  #SYNINT!EIGHT!ODDPAR!26,@PARCSR
1949 006336 016703 173364      MOV  TXCSR,R3 ;SET UP FOR ERROR MSG
1950 006342 112777 000377 173362  MOV  #377,@TXDBUF ;LOAD DATA CHAR
1951 006350 012767 000777 173122  MOV  #777,$TMP1 ;TO BE SHIFTED CHAR
1952 006356 012767 000011 172536  MOV  #9,SHIFT ;# OF SHIFTS
1953                                ; POKE CLK TO GET INTO SYNCRONIZATION
1954 006364 052777 020000 173334  BIS  #CLK,@TXCSR ;POKE CLK UP
1955 006372 042777 020000 173326  BIC  #CLK,@TXCSR ;POKE CLK DOWN
1956 006400 005000          1$ CLR  RO
1957 006402 006067 173072      ROR  $TMP1 ;FORCE CARRY
1958 006406 103002          BCC  2$ ;BR IF CARRY CLR
1959 006410 052700 002000 2$ BIS  #BITW,RO ;EQUIV OF BITW
1960 006414
```

INITIALIZE THE COMMON TAGS

```
1961 006414 052777 020000 173304 BIS #CLK,@TXCSR ;POKE CLK UP
1962 006422 042777 020000 173276 BIC #CLK,@TXCSR ;POKE CLK DOWN
1963 006430 017701 173272 MOV @TXCSR,R1 ;ACTUAL
1964 006434 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1965 006440 020001 CMP RO,R1 ;COMPARE EXP VS ACT
1966 006442 001401 BEQ +4
1967 006444 104003 ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1968 ;BIT,... ALSO CHECK DNA
1969 006446 005367 172450 DEC SHIFT ;# OF SHIFTS
1970 006452 001352 BNE 1$ ;DO IT AGAIN ?
1971 ;NOW POKE CLK TO SEE DNA
1972 006454 052777 020000 173244 BIS #CLK,@TXCSR ;POKE CLK
1973 006462 012700 100000 MOV #100000,RO ;EXPECTED
1974 006466 017701 173234 MOV @TXCSR,R1 ;ACTUAL
1975 006472 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1976 006476 020001 CMP RO,R1 ;COMPARE EXP VS ACT
1977 006500 001401 BEQ +4
1978 006502 104003 ERROR 3 ;DNA SHOULD BE SET
1979 ;IF DNA DID NOT SET
1980 ;CHECK WORD LENGTH SELECT LOGIC
1981
1982
1983
1984 .. THIS TEST VERIFYS THAT BY SENDING ONLY ONE SYNC
1985 .. CHARACTER (TWO SELECTED BY STRAPPING ) REACT =0
1986 .. THEN SEND ONE ORDINARY CHARACTER (TO BREAK UP THE SEQUENCE)
1987 .. REACT =0.... IT WILL TAKE TWO MORE SYNC CHARS
1988 .. BEFORE REACT =1
1989 .. NOTE THIS TEST WILL ONLY WORK WHEN TWO SYNC CHARS
1990 .. HAS BEEN BEEN SELECTED . OTHERWISE JUMP AROUND THIS TEST
1991 .. MODE SYNC INTERNAL (SYNINT)
1992 .. PARITY: NOPAR
1993 .. LENGTH: EIGHT
1994
1995 .. THIS TEST CHECKS ONLY THE RECEIVER SECTION
1996 ..
1997 .. *****
1998 006504 000004 TST15 SCOPE
1999 006506 105767 172434 TSTB SYNCNO ;TEST FOR # OF SYNC CHARS REQUIRED
2000 006512 100127 BPL 2$ ;IF NOT TWO GET OUT OF TEST
2001 006514 052777 000400 173204 BIS #MRESET,@TXCSR ;MASTER RESET
2002 006522 012777 030000 173172 MOV #SYNINT,@PARCSR ;SET THE MODE
2003 006530 052777 000400 173170 BIS #MRESET,@TXCSR ;MASTER RESET
2004
2005 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2006 006536 012777 064001 173162 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2007
2008 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2009 006544 012777 036026 173150 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2010 006552 052777 000020 173132 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2011 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2012 006560 042777 020000 173140 BIC #CLK,@TXCSR ;POKE CLK DOWN
2013 006566 052777 020000 173132 BIS #CLK,@TXCSR ;POKE CLK UP
2014 ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2015 006574 042777 020000 173124 BIC #CLK,@TXCSR ;POKE CLK DOWN
2016 006602 052777 020000 173116 BIS #CLK,@TXCSR ;POKE CLK UP
```

INITIALIZE THE COMMON TAGS

```
2017 006610 012767 000010 172304      MOV      #8,SHIFT      ;# OF SHIFTS
2018 006616 012767 000026 172654      MOV      #26,$TMP1     ;SYNC CHAR TO BE SHIFTED IN
2019 006624 004767 010342              JSR      PC,RPOKE      ;SHIFT IN THIS SYNC CHAR
2020 006630 032777 004000 173054      BIT      #RECACT,@RXCSR ;RECACT = 0 ?
2021 006636 001401              BEQ      +4
2022 006640 104004              ERROR   4              ;RECACT SHOULD BE 0
2023 006642 012767 000010 172252      MOV      #8,SHIFT      ;# OF SHIFTS
2024 006650 012767 000025 172622      MOV      #25,$TMP1     ;ANY CHARACTER
2025 006656 004767 010310              JSR      PC,RPOKE      ;SHIFT IN THIS CHARACTER
2026                                     ; YOU HAVE JUST LOST SYNCHRONIZATION.
2027                                     ; POKE THE CLK TWICE TO GET INTO SYNCHRONIZATION
2028                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2029 006662 042777 020000 173036      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2030 006670 052777 020000 173030      BIS      #CLK,@TXCSR   ;POKE CLK UP
2031                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2032 006676 042777 020000 173022      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2033 006704 052777 020000 173014      BIS      #CLK,@TXCSR   ;POKE CLK UP
2034 006712 012767 000002 172204      MOV      #2,COUNT      ;# OF SYNC CHARS
2035 006720 032777 004000 172764 15      BIT      #RECACT,@RXCSR ;RECACT = 0 ?
2036 006726 001401              BEQ      +4
2037 006730 104004              ERROR   4              ;RECACT SHOULD BE 0
2038 006732 012767 000010 172162      MOV      #8,SHIFT      ;# OF SHIFTS
2039 006740 012767 000026 172532      MOV      #26,$TMP1     ;SYNC CHAR
2040 006746 004767 010220              JSR      PC,RPOKE      ;SHIFT IN THIS SYNC CHAR
2041 006752 005367 172146              DEC      COUNT
2042 006756 001360              BNE     15              ;IS COUNT = 0 ? NO GO AGAIN
2043 006760 032777 004000 172724      BIT      #RECACT,@RXCSR ;RECACT = 1 ?
2044 006766 001001              BNE     +4
2045 006770 104004              ERROR   4              ;RECACT SHOULD BE ASSERTED
2046 006772
2047                                     25:
2048                                     ; THIS TEST VERIFYS MODE SELECT
2049                                     ; SYNC EXTERNAL VS. SYNC INTERNAL
2050                                     ;
2051                                     ; BASICALLY THE TEST CHECKS THAT THE RECEIVED
2052                                     ; DATA FREEZES IN SYNC INTERNAL
2053                                     ; IN SYNC EXTERNAL THIS DATA IS TRANSPARENT
2054                                     ; THIS TEST ONLY APPLIES TO THE RECEIVER SECTION
2055                                     ; LENGTH: EIGHT
2056                                     ; NOTE: SEARCH SYNC IS NOT SET
2057                                     ; *****
2057 006772 000004      TST16.  SCOPE
2058 006774 052777 000400 172724      BIS      #MRESET,@TXCSR ;MASTER RESET
2059 007002 012777 030000 172712      MOV      #SYNINT,@PARCSR ;SET THE MODE
2060 007010 052777 000400 172710      BIS      #MRESET,@TXCSR ;MASTER RESET
2061
2062                                     ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2063 007016 012777 064001 172702      MOV      #MCDATA!CLK!MINT!BREAK,@TXCSR
2064
2065                                     ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2066 007024 012777 036026 172670      MOV      #SYNINT!EIGHT!NOPAR!26,@PARCSR
2067                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2068 007032 042777 020000 172666      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2069 007040 052777 020000 172660      BIS      #CLK,@TXCSR   ;POKE CLK UP
2070                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2071 007046 042777 020000 172652      BIC      #CLK,@TXCSR   ;POKE CLK DOWN
2072 007054 052777 020000 172644      BIS      #CLK,@TXCSR   ;POKE CLK UP
```



```
2129          ;,LENGTH  EIGHT
2130          ;,
2131          ;,*****
2132 007354 000004 TST17 SCOPE
2133
2134 007356 052777 000400 172342      BIS      #MRESET,@TXCSR ,MASTER RESET
2135 007364 012777 030000 172330      MOV      #SYNINT,@PARCSR ;SET THE MODE
2136 007372 052777 000400 172326      BIS      #MRESET,@TXCSR ;MASTER RESET
2137
2138          ; SET MAINTENANCE MODE & SEND
2139          ; NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2140 007400 012777 004020 172320      MOV      #MINT!SEND,@TXCSR
2141
2142          ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2143 007406 012777 036026 172306      MOV      #SYNINT!EIGHT!NOPAR!26,@PARCSR
2144
2145 007414 105777 172306          TSTB     @TXCSR          ,TXDONE?
2146 007420 100401          BMI      +4
2147 007422 104004          ERROR    4              ;TXDONE SHOULD BE SET
2148 007424 112777 000021 172300      MOVB     #21,@TXDBUF    ;LOAD ANY CHAR
2149 007432 052777 020000 172266      BIS      #CLK,@TXCSR    ,POKE CLK UP
2150 007440 042777 020000 172260      BIC      #CLK,@TXCSR    ,POKE CLK DOWN
2151 007446 105777 172254          TSTB     @TXCSR
2152 007452 100001          BPL      +4
2153 007454 104004          ERROR    4              ,TXDONE SHOULD BE CLR
2154
2155 007456 052777 020000 172242      BIS      #CLK,@TXCSR    ,POKE CLK UP
2156 007464 042777 020000 172234      BIC      #CLK,@TXCSR    ,POKE CLK DOWN
2157 007472 105777 172230          TSTB     @TXCSR
2158 007476 100401          BMI      +4
2159 007500 104004          ERROR    4              ,TXDONE SHOULD BE SET
2160
2161
2162
2163          ;, THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER
2164          ;, RESET" WHILE IN STRIP SYNC MODE
2165          ;, THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR
2166          ;, WHEN STRIP SYNC IS SET AND SYNC CHARACTERS ARE SENT
2167          ;, BUT IF AN ERROR SHOULD OCCUR THIS AUTOMATIC RESET
2168          ;, IS DISCOMBOBULATED
2169          ;, IE. FORCE OVERRUN (OVERRUN) WHILE STRIP SYNC IS SET
2170          ;, BY TRANSMITTING A DATA CHARACTER THEN TRANSMIT A SYNC CHARACTER
2171          ;, AND DON'T READ THAT DATA CHARACTER NOTE NORMALLY THE LOGIC
2172          ;, RESETS THE RXDONE & ERROR FLAGS PROVIDING THAT ONLY SYNC CHARACTERS ARE
2173          ;, STRIPPED
2174          ;, MODE SYNC EXTERNAL (SYNEXT)
2175          ;, LENGTH. EIGHT
2176          ;, NOTE THIS TEST USES BOTH RECEIVER AND TRANSMITTER LOGIC
2177          ;,
2178          ;,
2179          ;, *****
2180 007502 000004 TST20 SCOPE
2181
2182 007504 052777 000400 172214      BIS      #MRESET,@TXCSR ,MASTER RESET
2183 007512 012777 020000 172202      MOV      #SYNEXT,@PARCSR ,SET THE MODE
2184 007520 052777 000400 172200      BIS      #MRESET,@TXCSR ,MASTER RESET
```

```

2185
2186      .SET MAINTENANCE MODE & SEND
2187      ,NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2188 007526 012777 004020 172172      MOV      #MINT!SEND,@TXCSR
2189
2190      .SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2191 007534 012777 026026 172160      MOV      #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2192 007542 112777 000026 172162      MOV      #26,@TXDBUF      ;LOAD SYNC CHAR
2193 007550 052777 000420 172134      BIS      #SYNSCH!STPSYN,@RXCSR      ;SET SYNC SEARCH & STRIP SYNC
2194 007556 016703 172134      MOV      RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
2195 007562 012767 000003 171334      MOV      #3,COUNT      ;# OF TIMES SYNC WILL BE SENT
2196 007570 052777 020000 172130      BIS      #CLK,@TXCSR      ;POKE CLK UP
2197 007576 042777 020000 172122      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2198 007604 012767 000010 171310      MOV      #8,SHIFT      ;# OF SHIFTS
2199
2200 007612 052777 020000 172106      BIS      #CLK,@TXCSR      ;POKE CLK UP
2201 007620 042777 020000 172100      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2202 007626 005367 171270      DEC      SHIFT
2203 007632 001367      BNE      1$
2204 007634 105777 172052      TSTB     @RXCSR      ,RXDONE?
2205 007640 100001      BPL      +4
2206 007642 104004      ERROR    4      ,RXDONE SHOULD NOT ASSERT
2207 007644 005367 171254      DEC      COUNT
2208 007650 001355      BNE      2$
2209 007652 012700 000026      MOV      #26,R0      ,EXPECTED
2210 007656 017701 172034      MOV      @RXDBUF,R1      ,ACTUAL
2211 007662 020001      CMP      R0,R1
2212 007664 001401      BEQ      +4
2213 007666 104002      ERROR    2      ,NOTE THAT OVERRUN SHOULD NOT OCCUR, ALSO
2214      ,SECOND & 3RD SYNC CHARACTER CAME FROM
2215      ,SYNC HOLDING REGISTER
2216 007670 012767 000003 171226      MOV      #3,COUNT      ,# OF TIMES
2217 007676 112777 000025 172026      MOV      #25,@TXDBUF      ,LOAD ANY CHAR HOWEVER
2218      ,ONE MORE SYNC CHAR WILL BE SENT BEFORE
2219      ,THE "25" CHAR IS SENT (THE DMA BIT IS
2220      ,ALREADY UP)
2221
2222 007704 012767 000010 171210      MOV      #8,SHIFT      ,# OF SHIFTS
2223
2224      3$
2225 007712 052777 020000 172006      BIS      #CLK,@TXCSR      ;POKE CLK UP
2226 007720 042777 020000 172000      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2227 007726 005367 171170      DEC      SHIFT
2228 007732 001367      BNE      3$
2229 007734 005367 171164      DEC      COUNT
2230 007740 001361      BNE      4$
2231 007742 105777 171744      TSTB     @RXCSR      ,RXDONE = 1 ?
2232 007746 100401      BMI      +4
2233 007750 104004      ERROR    4      ,RXDONE SHOULD BE SET
2234 007752 012700 140026      MOV      #RXERR'OVRUN'26,R0      ,EXPECTED
2235 007756 017701 171734      MOV      @RXDBUF,R1      ;ACTUAL
2236 007762 020001      CMP      R0,R1
2237 007764 001401      BEQ      +4
2238 007766 104002      ERROR    2      ,NOTE THAT OVERRUN SHOULD OCCUR.
2239      ,ALSO SECOND SYNC CHARACTER CAME
2240      ,FROM SYNC HOLDING REGISTER

```

```
2241                                     , SUMMARY THE OVRUN STOPPED
2242                                     , THE AUTOMATIC RESETTING OF
2243                                     , RXDONE & ERROR FLAGS . CHECK THIS
2244
2245                                     ,, THIS TEST VERIFYS THAT EITHER SEQUENCE OF
2246                                     ,, LOADING TXDBUF AND SETTING SEND
2247                                     ,, DOES CAUSE TRANSMISSION
2248                                     ,, MODE SYNC EXT
2249                                     ,, LENGTH EIGHT
2250
2251                                     ,, *****
2252 007770 000004 TST21 SCOPE
2253
2254 007772 052777 000400 171726 BIS #MRESET, @TXCSR ; MASTER RESET
2255 010000 012777 020000 171714 MOV #SYNEXT, @PARCSR ; SET THE MODE
2256 010006 052777 000400 171712 BIS #MRESET, @TXCSR ; MASTER RESET
2257
2258                                     , SET MAINTENANCE MODE & SEND
2259                                     , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2260 010014 012777 004020 171704 MOV #MINT!SEND, @TXCSR
2261
2262                                     , SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2263 010022 012777 026026 171672 MOV #SYNEXT'EIGHT'NOPAR'26, @PARCSR
2264 010030 016703 171672 MOV TXCSR, R3 ; SET UP FOR ERROR MESSAGE
2265 010034 042777 000020 171664 BIC #SEND, @TXCSR ; DROP SEND
2266 010042 012767 000025 171430 MOV #25, $TMP1
2267 010050 112777 000025 171654 MOVB #25, @TXDBUF ; LOAD CHARACTER
2268 010056 052777 000020 171642 BIS #SEND, @TXCSR ; SET SEND
2269
2270 010064 052777 020000 171634 , GET INTO SYNCRONIZATION
2271 010072 042777 020000 171626 BIS #CLK, @TXCSR ; POKE CLK UP
2272 010100 012767 000010 171014 BIC #CLK, @TXCSR ; POKE CLK DOWN
2273 010106 005000 15 CLR R0 ; # OF SHIFTS
2274 010110 006067 171364 ROR $TMP1
2275 010114 103002 BCC 25
2276 010116 052700 002000 BIS #BITW, R0 ; EQUIV OF BIT WINDOW
2277
2278 010122 25
2279 010122 052777 020000 171576 BIS #CLK, @TXCSR ; POKE CLK UP
2280 010130 042777 020000 171570 BIC #CLK, @TXCSR ; POKE CLK DOWN
2281 010136 017701 171564 MOV @TXCSR, R1 ; ACTUAL
2282 010142 042701 075777 BIC #075777, R1 ; SAVE BIT WINDOW & DNA
2283 010146 020001 CMP R0, R1
2284 010150 001401 BEQ +4
2285 010152 104003 ERROR 3 ; BIT WINDOW DID NOT MATCH ACTUAL DATA BIT
2286 ; ALSO CHECK DNA
2287 010154 005367 170742 DEC SHIFT
2288 010160 001352 BNE 15
2289
2290
2291
2292                                     , THIS TEST VERIFYS THAT DROPPING OF SEND IN THE
2293                                     , MIDDLE OF TRANSMITTING A CHARACTER DOES INDEED
2294                                     , FINISH TRANSMITTING THAT CHARACTER
2295                                     , MODE SYNC EXT
2296                                     , LENGTH EIGHT
```

```

2297
2298
2299
2300 010162 000004 TST22 SCOPE
2301
2302 010164 052777 000400 171534 BIS #MRESET,@TXCSR ;MASTER RESET
2303 010172 012777 020000 171522 MOV #SYNEXT,@PARCSR ;SET THE MODE
2304 010200 052777 000400 171520 BIS #MRESET,@TXCSR ;MASTER RESET
2305
2306 ,SET MAINTENANCE MODE & SEND
2307 ,NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2308 010206 012777 004020 171512 MOV #MINT!SEND,@TXCSR
2309
2310 ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2311 010214 012777 026026 171500 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2312 010222 016703 171500 MOV TXCSR,R3 ,SETUP FOR ERROR MESSAGE
2313 010226 112777 000252 171476 MOVB #252,@TXDBUF ;LOAD DATA CHAR
2314 010234 012767 000252 171236 MOV #252,$TMP1 ,SHIFTED CHAR
2315 010242 012767 000010 170652 MOV #8,SHIFT ,# OF SHIFTS
2316 ;GET INTO SYNCRONIZATION
2317 010250 052777 020000 171450 BIS #CLK,@TXCSR ,POKE CLK UP
2318 010256 042777 020000 171442 BIC #CLK,@TXCSR ,POKE CLK DOWN
2319
2320 010264 005000 15 CLR R0
2321 010266 006067 171206 ROR $TMP1 ,FORCE CARRY
2322 010272 103002 BCC 25
2323 010274 052700 002000 BIS #BITW,R0 ,EQUIV OF BIT WINDOW
2324
2325 010300 25
2326 010300 052777 020000 171420 BIS #CLK,@TXCSR ,POKE CLK UP
2327 010306 042777 020000 171412 BIC #CLK,@TXCSR ,POKE CLK DOWN
2328 010314 017701 171406 MOV @TXCSR,R1 ;ACTUAL
2329 010320 042701 075777 BIC #075777,R1 ,SAVE ONLY BIT WINDOW & DNA
2330 010324 020001 CMP R0,R1
2331 010326 001401 BEQ +4
2332 010330 104003 ERROR 3 ,BIT WINDOW DID NOT MATCH
2333 ,ACTUAL DATA BIT
2334 010332 005367 170564 DEC SHIFT
2335 010336 022767 000003 170556 CMP #3,SHIFT
2336 010344 001003 BNE 35
2337 010346 042777 000020 171352 BIC #SEND,@TXCSR ,DROP SEND
2338 010354 005767 170542 35 TST SHIFT
2339 010360 001341 BNE 15 ,DO IT AGAIN?
2340
2341
2342
2343 ,, THIS TEST VERIF/S THAT RXDONE ASSERTS WHEN STRIP SYNC IS SET
2344 ,, MODE SYNC INTERNAL
2345 ,, LENGTH: EIGHT
2346
2347
2348 010362 000004 TST23 SCOPE
2349 010364 052777 000400 171334 BIS #MRESET,@TXCSR ;MASTER RESET
2350 010372 012777 030000 171322 MOV #SYNINT,@PARCSR ;SET THE MODE
2351 010400 052777 000400 171320 BIS #MRESET,@TXCSR ;MASTER RESET
2352

```



```

2353 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2354 010406 012777 064001 171312 MOV #MNTDATA!CLK!MINT!BREAK,@TXCSR
2355
2356 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2357 010414 012777 036026 171300 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2358
2359 010422 052777 000420 171262 BIS #SYNSCH!STPSYN,@RXCSR ;SET SYNC SEARCH &
2360 ;STRIP SYNC
2361 ,POKE CLK TO GET RECEIVER INTO SYNCRONIZATION
2362 010430 042777 020000 171270 BIC #CLK,@TXCSR ;POKE CLK
2363 010436 052777 020000 171262 BIS #CLK,@TXCSR ;
2364 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2365 010444 042777 020000 171254 BIC #CLK,@TXCSR ;POKE CLK DOWN
2366 010452 052777 020000 171246 BIS #CLK,@TXCSR ;POKE CLK UP
2367 010460 016703 171232 MOV RXDBUF,R3 ,FOR ERROR MESSAGE
2368 010464 012767 000002 170432 MOV #2,COUNT ,# OF TIMES OF SYNC CHARS
2369 ,TEST TO SEE HOW MANY SYNC CHARS NEEDED
2370 010472 105767 170450 TSTB SYNCNO
2371 010476 100402 BMI 3$ ;WILL IT BE ONE OR TWO ?
2372 010500 005367 170420 DEC COUNT ;MAKE IT ONE LESS
2373 010504 012767 000010 170410 3$ MOV #8,SHIFT ;#OF SHIFTS
2374 010512 012767 000026 170760 MOV #26,$TMP1 ;SYNC CHAR
2375 010520 004767 006446 JSR PC,RPOKE
2376 010524 005367 170374 DEC COUNT ;IS IT THE LAST SYNC CHAR ?
2377 010530 001365 BNE 3$ ,GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
2378 010532 032777 004000 171152 BIT #REACT,@RXCSR ;REACT=1?
2379 010540 001001 BNE .+4
2380 010542 104004 ERROR 4 ;REACT SHOULD BE ASSERTED
2381 010544 105777 171142 TSTB @RXCSR ;RXDONE=0?
2382 010550 100001 BPL .+4
2383 010552 104004 ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2384 010554 012767 000010 170340 MOV #8,SHIFT ;#OF SHIFTS
2385 010562 012767 000025 170710 MOV #25,$TMP1 ;ANY CHARACTER
2386 010570 004767 006376 JSR PC,RPOKE
2387 010574 105777 171112 TSTB @RXCSR ;RXDONE=1?
2388 010600 100401 BMI .+4
2389 010602 104004 ERROR 4 ;RXDONE SHOULD NOW BE ASSERTED
2390 010604 012700 000025 MOV #25,R0 ;EXPECTED
2391 010610 017701 171102 MOV @RXDBUF,R1 ;ACTUAL
2392 010614 020001 CMP R0,R1
2393 010616 001401 BEQ .+4
2394 010620 104002 ERROR 2 ;CHARACTERS SHOULD BE MATCHED
2395
2396
2397
2398 ;,THIS TEST VERIFYS THAT BY LOPPING SYNSCH
2399 ;,IN THE MIDDLE OF A CHARACTER, SYNC CHARACTER SEQUENCE
2400 ;,IS NEEDED BEFORE REACT, RXDONE ASSERT AGAIN
2401 ;,ALSO NOTE: SINCE REACT IS DEPENDENT ON MATCH DETECT,
2402 ;,AND IF SYNSCH IS DROPPED IN THE MIDDLE OF
2403 ;,A SYNC CHARACTER AND THEN RAISED AGAIN ;RXDONE SHOULD
2404 ;,NOT ASSERT UNTIL NEW SYNC CHARACTER SEQUENCE HOWEVER
2405 ;,ONE "PAD" CHARACTER MUST PRECEED THIS JUGGLING OF SEARCH SYNC (SYNSCH)
2406 ;,MODE SYNC INTERNAL (SYNINT)
2407 ;,LENGTH EIGHT
2408 ;,

```

```
2409
2410 ; ; *****
2411 010622 000004 TST24 SCOPE
2412
2413 010624 052777 000400 171074 BIS #MRESET,@TXCSR ; MASTER RESET
2414 010632 012777 030000 171062 MOV #SYNINT,@PARCSR ; SET THE MODE
2415 010640 052777 000400 171060 BIS #MRESET,@TXCSR ; MASTER RESET
2416
2417 ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2418 010646 012777 064001 171052 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2419
2420 ; SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
2421 010654 012777 036026 171040 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
2422
2423 010662 052777 000020 171022 BIS #SYNSCH,@RXCSR ; SET SYNC SEARCH
2424 ; POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2425 010670 042777 020000 171030 BIC #CLK,@TXCSR ; POKE CLK DOWN
2426 010676 052777 020000 171022 BIS #CLK,@TXCSR ; POKE CLK UP
2427 ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2428 010704 042777 020000 171014 BIC #CLK,@TXCSR ; POKE CLK DOWN
2429 010712 052777 020000 171006 BIS #CLK,@TXCSR ; POKE CLK UP
2430 010720 012767 000002 170176 MOV #2,COUNT ; # OF TIMES
2431 010726 016703 170764 MOV RXDBUF,R3 ; FOR ERROR MESSAGE
2432 010732 012767 000010 170162 15. MOV #8,SHIFT ; # OF SHIFTS
2433 010740 012767 000026 170532 MOV #26,$TMP1 ; SYNC CHAR.
2434 010746 004767 006220 JSR PC,RPOKE
2435 010752 005367 170146 DEC COUNT
2436 010756 001403 BEQ 25
2437 ; TEST TO SEE HOW MANY SYNC CHARACTERS NEEDED
2438 010760 105767 170162 TSTB SYNCNO
2439 010764 100762 BMI 15
2440
2441 010766 032777 004000 170716 25 BIT #REACT,@RXCSR ,REACT=1?
2442 010774 001001 BNE +4
2443 010776 104004 ERROR 4 ; REACT SHOULD BE SET
2444 011000 105777 170706 TSTB @RXCSR ,RXDONE = 0?
2445 011004 100001 BPL +4
2446 011006 104004 ERROR 4 ; RXDONE SHOULD = 0
2447 ; THE FOLLOWING ALLOWS THE RECEIVER "CHIP" TO RECOGNIZE THE DROPPING OF SYNSCH
2448 ; BASICALLY IT SIMULATES THE "PAD" CHARACTER REQUIREMENT AT THE END
2449 ; OF DATA TRANSFER.
2450 ; ONE "PAD" CHARACTER IS REQUIRED TO FLUSH OUT SYNC CHARACTER
2451 ; AND YOU HAVE TO DROP SYNSCH ON THE NEXT OR FOLLOWING CHARACTER
2452 ; FOR A "CLEAN" RESTART
2453 011010 012767 000010 170104 MOV #8,SHIFT ; # OF SHIFTS
2454 011016 012767 000025 170454 MOV #25,$TMP1 ; "PAD" CHARACTER
2455 011024 004767 006142 JSR PC,RPOKE ; SHIFT IN "PAD" CHARACTER
2456 011030 105777 170656 TSTB @RXCSR ,RXDONE = 1?
2457 011034 100401 BMI +4
2458 011036 104004 ERROR 4 ; RXDONE SHOULD = 1
2459 011040 017701 170652 MOV @RXDBUF,R1 ; STOPE ACTIUAL & CLEAR RXDONE
2460 011044 012700 000025 MOV #25,R0 ; EXPECTED
2461 011050 020001 CMP R0,R1 ; COMPARE EXPECTED DATA VS ACTUAL DATA
2462 011052 001401 BEQ +4
2463 011054 104002 ERROR 2 ; DATA SHOULD COMPARE WITHOUT ERROR FLAGS
2464 ; THE FOLLOWING IS THE SYNC CHARACTER AFTER THE "PAD" CHARACTER
```

```

2465      , IN WHICH SEARCH SYNC (SYNSCH) IS JUGGLED TO CREATE AN ABORT SITUATION
2466      ; AT THE END OF A DATA TRANSFER SEQUENCE
2467 011056 012767 000004 170036      MOV      #4, SHIFT      ; # OF SHIFTS
2468 011064 012767 000026 170406      MOV      #26, STMP1     ; SYNC CHAR.
2469 011072 004767 006074              JSR      PC, RPOKE
2470 011076 032777 004000 170606      BIT      #REACT, @RXCSR , REACT=1?
2471 011104 001001              BNE      +4
2472 011106 104004              ERROR    4              ; REACT SHOULD STILL BE SET
2473 011110 105777 170576      TSTB    @RXCSR ; RXDONE = 0 ?
2474 011114 100001              BPL      +4
2475 011116 104004              ERROR    4              ; RXDONE SHOULD = 0 FROM PREVIOUS READING OF RXDBUF
2476 011120 042777 000020 170564      BIC      #SYNSCH, @RXCSR , DROP SEARCH SYNC
2477 011126 032777 004000 170556      BIT      #REACT, @RXCSR ; REACT=0?
2478 011134 001401              BEQ      +4
2479 011136 104004              ERROR    4              ; REACT SHOULD NOT BE SET
2480      , NOW SHIFT TWO BITS TO ALLOW SEARCH SYNC =0 TO TAKE
2481      , EFFECT IN THE LOGIC (THIS ALLOWS THE RECEIVER CHIP TO SEE
2482      , THE DROPPING OF SEARCH SYNC)
2483 011140 012767 000002 167754      MOV      #2, SHIFT     ; # OF SHIFTS
2484 011146 004767 006020              JSR      PC, RPOKE
2485 011152 052777 000020 170532      BIS      #SYNSCH, @RXCSR , SET SEARCH SYNC
2486 011160 032777 004000 170524      BIT      #REACT, @RXCSR
2487 011166 001401              BEQ      +4
2488 011170 104004              ERROR    4              ; REACT = 0 ?
2489 011172 105777 170514      TSTB    @RXCSR ; RXDONE = 0 ?
2490 011176 100001              BPL      +4
2491 011200 104004              ERROR    4              ; RXDONE = 0 ?
2492 011202 012767 000002 167712      MOV      #2, SHIFT     ; # OF SHIFTS TO FINISH UP THE SYNC CHARACTER
2493 011210 004767 005756              JSR      PC, RPOKE
2494 011214 032777 004000 170470      BIT      #REACT, @RXCSR , REACT=0?
2495 011222 001401              BEQ      +4
2496 011224 104004              ERROR    4              ; REACT SHOULD NOT BE SET
2497 011226 105777 170460      TSTB    @RXCSR ; RXDONE=0?
2498 011232 100001              BPL      +4
2499 011234 104004              ERROR    4              ; RXDONE SHOULD NOT BE ASSERTED
2500 011236 012700 000025              MOV      #25, R0        ; EXPECTED
2501 011242 017701 170450              MOV      @RXDBUF, R1    ; ACTUAL
2502 011246 020001              CMP      R0, R1        ; COMPARE EXPECTED VS ACTUAL
2503 011250 001401              BEQ      +4
2504 011252 104002              ERROR    2              ; CHARACTERS SHOULD BE MATCHED
2505      ; REMEMBER THAT THE " 25 " FROZE WHEN SYNSCH WAS DROPPED
2506      ; THEREFORE . 25 WILL BE READ AND NOT 26
2507      ; THE FOLLOWING VERIFYS THAT THE RE SYNCHRONIZATION COMES UP "CLEAN"
2508 011254 012767 000002 167642      MOV      #2, COUNT     ; # OF TIMES OF SYNC CHARS
2509      ;
2510      ; TEST TO SEE HOW MANY SYNC CHARS NEEDED
2511 011262 105767 167660      TSTB    SYNCNO ; HOW MANY SYNCs ARE YOU STRAPPED FOR ?
2512 011266 100402              BMI     3$            ; WILL IT BE ONE OR TWO ?
2513 011270 005367 167630              DEC     COUNT ; IT WAS ONLY ONE NEEDED
2514 011274 012767 000010 167620 3$      MOV      #8, SHIFT     ; # OF SHIFTS
2515 011302 012767 000026 170170      MOV      #26, STMP1    ; SYNC CHAR
2516 011310 004767 005656              JSR      PC, RPOKE
2517 011314 005367 167604              DEC     COUNT ; IS IT THE LAST SYNC CHAR ?
2518 011320 001365              BNE     3$            ; GO AGAIN AND SHIFT IN ANOTHER SYNC CHAR
2519 011322 032777 004000 170362      BIT      #REACT, @RXCSR , REACT=1?
2520 011330 001001              BNE      +4
  
```

```
2521 011332 104004          ERROR 4 ;REACT SHOULD BE ASSERTED
2522 011334 105777 170352  TSTB @RXCSR ;RXDONE=0?
2523 011340 100001          BPL .+4
2524 011342 104004          ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2525 011344 012767 000010 167550 MOV #8,SHIFT ;#OF SHIFTS
2526 011352 012767 000025 170120 MOV #25,STMP1 ;ANY CHARACTER
2527 011360 004767 005606 JSR PC,RPOKE
2528 011364 105777 170322 TSTB @RXCSR ;RXDONE=1?
2529 011370 100401          BMI .+4
2530 011372 104004          ERROR 4 ;RXDONE SHOULD NOW BE ASSERTED
2531 011374 012700 000025 MOV #25,R0 ;EXPECTED
2532 011400 017701 170312 MOV @RXDBUF,R1 ;ACTUAL
2533 011404 020001          CMP R0,R1
2534 011406 001401          BEQ .+4
2535 011410 104002          ERROR 2 ;CHARACTERS SHOULD BE MATCHED
2536
2537
2538
2539 ;, THIS TEST VERIFYS THAT HDX MODE DISQUALIFIES THE
2540 ;, RECEIVER WHEN SEND IS ASSERTED
2541 ;, MODE: SYNC EXT
2542 ;, LENGTH: EIGHT
2543 ;, NOTE: THIS TEST WORKS ONLY IN MAINT EXTERNAL MODE
2544 ;, THIS TEST USES BOTH RECEIVER & TRANSMITTER LOGIC
2545 ;, *****
2546 011412 000004          TST25 SCOPE
2547 011414 105767 167533 TSTB JMRBY
2548 011420 100155          BPL 15 ;GET OUT OF THIS TEST IF "NO"
2549 011422 016703 170270 MOV RXDBUF,R3 ;FOR ERROR MESSAGE
2550 011426 052777 000400 170272 BIS #MRESET,@TXCSR ;MASTER RESET
2551 011434 012777 020000 170260 MOV #SYNEXT,@PARCSR ;SET THE MODE
2552 011442 052777 000400 170256 BIS #MRESET,@TXCSR ;MASTER RESET
2553
2554 ;, SET MAINTENANCE MODE & SEND
2555 ;, NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2556 011450 012777 010020 170250 MOV #NEXT!SEND,@TXCSR
2557
2558 ;, SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2559 011456 012777 026026 170236 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2560 011464 052777 000020 170220 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2561 011472 112777 000025 170232 MOVB #25,@TXDBUF ;ANY CHARACTER
2562 ;, POKE CLK FOR SYNCRONIZATION
2563 011500 052777 020000 170220 BIS #CLK,@TXCSR ;POKE CLK UP
2564 ;, WAIT FOR CABLE & DRIVER DELAYS
2565 011506 016702 167406 MOV HOLD,R2 ;WAIT THIS AMT
2566 011512 005302          DEC R2 ;WAIT
2567 011514 001376          BNE -2
2568 ;, EXIT.
2569 011516 042777 020000 170202 BIC #CLK,@TXCSR ;POKE CLK DOWN
2570 ;, WAIT FOR CABLE & DRIVER DELAYS
2571 011524 016702 167370 MOV HOLD,R2 ;WAIT THIS AMT
2572 011530 005302          DEC R2 ;WAIT
2573 011532 001376          BNE -2
2574 ;, EXIT
2575 011534 012767 000010 167360 MOV #8,SHIFT ;# OF SHIFTS
2576 011542
```

2577	011542	052777	020000	170156	BIS	#CLK,@TXCSR	;POKE CLK UP
2578							;WAIT FOR CABLE & DRIVER DELAYS
2579	011550	016702	167344		MOV	HOLD,R2	;WAIT THIS AMT
2580	011554	005302			DEC	R2	;WAIT
2581	011556	001376			BNE	-2	
2582							;EXIT...
2583	011560	042777	020000	170140	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2584							;WAIT FOR CABLE & DRIVER DELAYS
2585	011566	016702	167326		MOV	HOLD,R2	;WAIT THIS AMT
2586	011572	005302			DEC	R2	;WAIT
2587	011574	001376			BNE	-2	
2588							;EXIT...
2589	011576	005367	167320		DEC	SHIFT	;# OF SHIFTS
2590	011602	022767	000003	167312	CMP	#3,SHIFT	;IS IT TIME TO LOAD NEXT CHAR ?
2591	011610	001003			BNE	3\$	
2592	011612	112777	000024	170112	MOVB	#24,@TXDBUF	;LOAD NEXT CHAR
2593	011620	005767	167276		TST	SHIFT	
2594	011624	001346			BNE	2\$	
2595	011626	105777	170060		TSTB	@RXCSR	;RXDONE=1?
2596	011632	100401			BMI	+4	
2597	011634	104004			ERROR	4	;RXDONE SHOULD BE SET
2598	011636	012700	000025		MOV	#25,R0	;EXPECTED
2599	011642	017701	170050		MOV	@RXDBUF,R1	;ACTUAL
2600	011646	020001			CMP	R0,R1	
2601	011650	001401			BEQ	+4	
2602	011652	104002			ERROR	2	;CHARACTERS SHOULD COMPARE
2603	011654	052777	000010	170044	BIS	#HDXEN,@TXCSR	;SET HALF DUPLEX HDX
2604	011662	012767	000010	167232	MOV	#8,SHIFT	;# OF SHIFT
2605	011670						
2606	011670	052777	020000	170030	BIS	#CLK,@TXCSR	;POKE CLK UP
2607							;WAIT FOR CABLE & DRIVER DELAYS
2608	011676	016702	167216		MOV	HOLD,R2	;WAIT THIS AMT
2609	011702	005302			DEC	R2	;WAIT
2610	011704	001376			BNE	-2	
2611							;EXIT
2612	011706	042777	020000	170012	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2613							;WAIT FOR CABLE & DRIVER DELAYS
2614	011714	016702	167200		MOV	HOLD,R2	;WAIT THIS AMT
2615	011720	005302			DEC	R2	;WAIT
2616	011722	001376			BNE	-2	
2617							;EXIT
2618	011724	005367	167172		DEC	SHIFT	
2619	011730	001357			BNE	4\$	
2620	011732	105777	167754		TSTB	@RXCSR	;RXDONE=0?
2621	011736	100001			BPL	+4	
2622	011740	104004			ERROR	4	;RXDONE SHOULD NOT BE ASSERTED
2623							;CHECK OUT HDX LOGIC
2624	011742	017701	167750		MOV	@RXDBUF,R1	;ACTUAL
2625	011746	020001			CMP	R0,R1	
2626	011750	001401			BEQ	+4	
2627	011752	104002			ERROR	2	;CHARACTERS SHOULD COMPARE
2628							;NOTE THAT CHARACTER 25 WILL BE FROZEN
2629							;IN THE RXDBUF EVEN THOUGH CHARACTER 24 WAS
2630							;SENT TO THE RECEIVER
2631							
2632	011754						

```

2633
2634
2635      ;; THIS TEST VERIFYS THAT BREAK FORCES A SPACE CONDITION
2636      ;; ON THE LINE WHILE TRANSMITTING
2637      ;; THIS TEST USES BOTH THE RECEIVER AND TRANSMITTER LOGIC
2638      ;; MODE SYNC EXT (SYNEXT)
2639      ;; LENGTH: EIGHT
2640      ;;
2641
2642      ;; *****
2643 011754 000004      TST26 SCOPE
2644 011756 052777 000400 167742      BIS      #MRESET, @TXCSR      , MASTER RESET
2645 011764 012777 020000 167730      MOV      #SYNEXT, @PARCSR ; SET THE MODE
2646 011772 052777 000400 167726      BIS      #MRESET, @TXCSR      , MASTER RESET
2647
2648      , SET MAINTENANCE MODE & SEND
2649      , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2650 012000 012777 004020 167720      MOV      #MINT!SEND, @TXCSR
2651
2652      , SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2653 012006 012777 026026 167706      MOV      #SYNEXT!EIGHT!NOPAR!26, @PARCSR
2654 012014 052777 000020 167670      BIS      #SYNSCH, @RXCSR      , SET SEARCH SYNC
2655 012022 016703 167670      MOV      RXDBUF, R3      , FOR ERROR MESSAGE
2656 012026 012767 000002 167070      MOV      #2, COUNT      , # OF TIMES
2657 012034 112777 000025 167670      MOV      #25, @TXDBUF      ; ANY CHARACTER
2658      , POKE CLK FOR SYNCRONIZATION
2659 012042 052777 020000 167656      BIS      #CLK, @TXCSR      ; POKE CLK UP
2660 012050 042777 020000 167650      BIC      #CLK, @TXCSR      , POKE CLK DOWN
2661 012056 012700 000025      MOV      #25, R0      , EXPECTED
2662 012062 012767 000010 167032 1$      MOV      #8, SHIFT      , # OF SHIFTS
2663
2664      2$
2665 012070 052777 020000 167630      BIS      #CLK, @TXCSR      , POKE CLK UP
2666 012076 042777 020000 167622      BIC      #CLK, @TXCSR      , POKE CLK DOWN
2667 012104 005367 167012      DEC      SHIFT
2668 012110 022767 000003 167004      CMP      #3, SHIFT
2669 012116 001003      BNE      3$
2670 012120 112777 000024 167604      MOV      #24, @TXDBUF      , LOAD NEXT CHAR
2671 012126 005767 166770 3$      TST      SHIFT
2672 012132 001356      BNE      2$
2673 012134 105777 167552      TSTB     @RXCSR      , RXDONE=1?
2674 012140 100401      BMI     +4
2675 012142 104004      ERROR   4
2676 012144 017701 167546      MOV      @RXDBUF, R1      , ACTUAL
2677 012150 020001      CMP      R0, R1
2678 012152 001401      BEQ     +4
2679 012154 104003      ERROR   3
2680 012156 052777 000001 167542      BIS      #BREAK, @TXCSR      , SET BREAK
2681 012164 012700 000000      MOV      #0, R0      , EXPECTED
2682 012170 005367 166730      DEC      COUNT
2683 012174 001332      BNE     1$
2684
2685
2686      ;; THIS TEST VERIFYS THAT DSC CAUSES AN INTERRUPT
2687      ;; THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
2688      ;; INTERRUPT VECTOR DURIV
  
```

```
2689
2690 012176 000004
2691 012200 105767 166747
2692 012204 100073
2693 012206 052777 000400 167512
2694 012214 105767 166731
2695 012220 100405
2696 012222 012777 000000 167462
2697 012230 005777 167456
2698 012234 012777 012256 167474 55
2699 012242 016777 004564 167470
2700 012250 106427 000000
2701 012254 000422
2702
2703 012256 106427 000340
2704 012262 005777 167424
2705 012266 100401
2706 012270 104004
2707 012272 042777 000040 167412
2708 012300 012716 012370
2709 012304 016777 167430 167424
2710 012312 012777 000000 167420
2711 012320 000002
2712
2713 012322 052777 000040 167362 15
2714 012330 052777 000002 167354
2715 012336 005000
2716 012340 005200
2717 012342 001376
2718 012344 016777 167370 167364
2719 012352 012777 000000 167360
2720
2721 012360 042777 000040 167324
2722 012366 104004
2723 012370 106427 000340 25
2724 012374 35
2725
2726
2727
2728
2729
2730
2731
2732
2733 012374 000004 EOP SCOPE
2734 012376 004767 000340 JSR PC,CKSWR
2735 012402 104401 TYPE ,TYPE NAME OF TEST
2736 012404 015532 MEPASS
2737 012406 104413 012640 CONVRT ,OUTCRY
2738 012412 104401 015351 TYPE ,DEVICE
2739 012416 105767 166530 TSTB MULTD ,ARE YOU RUNNING MULTIPLE DEVICES ?
2740 012422 001511 BEQ CCC ,NO, JUMP AROUND
2741 012424 005767 166536 TST ACTREG ,ARE ANY DEVICES ACTIVE ?
2742 012430 001007 BNE RUNIT ,YES
2743 012432 104401 015363 TYPE ,MCOW ;NO
2744 012436 016700 166524 MOV ACTREG,RO ,DISPLAY ACTREG
```

```

2745 012442 000000          HALT          ;SELECT SOMETHING TO RUN @ ACTREG
2746          ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2747 012444 000167 167502    JMP          .START ;START OVER AGAIN..... YOU DESELECTED EVERYTHING
2748 012450 062767 000010 166476 RUNIT. ADD      #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2749 012456 062767 000010 166476 ZERO  ADD      #10,BASEIV  ;NEXT BLOCK (VECTORS)
2750 012464 000241          CLC
2751 012466 006167 166476    ROL        ROTADD ;UP DATE ROTATING POINTER
2752 012472 103410          BCS        25      ;IS IT THE LAST DEVICE
2753          ;TO BE TESTED IN THIS PASS ?
2754 012474 036767 166470 166464    BIT        ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2755 012502 001762          BEQ        RUNIT  ;IF NOT ACTIVE, TRY NEXT ADDRESS
2756 012504 004767 000034          JSR        PC,REPLAY ;CALCULATE NEW PARAMETERS
2757 012510 000167 000210          JMP        RESTRT  ;YES IT WAS ACTIVE, TEST THIS DEVICE
2758 012514 012767 000001 166446 25  MOV        #1,ROTADD ;OK!,NOW SET UP ROTATING
2759          ;POINTER FOR NEXT MULTIPLE PASS
2760 012522 016767 166430 166424    MOV        KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2761 012530 016767 166430 166424    MOV        KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
2762 012536 004767 000002          JSR        PC,REPLAY ;CALC NEW PARAMETERS
2763 012542 000441          BR         CCC     ;JUMP AROUND REPLAY
2764 012544 016767 166404 004416 REPLAY. MOV      BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
2765 012552 004767 004260          JSR        PC,DUADDR ;CREATE NEW ADDRESSES
2766 012556 016767 166400 167152    MOV        BASEIV,DURIV ;CREATE DURIV
2767 012564 062767 000002 166370    ADD        #2,BASEIV
2768 012572 016767 166364 167140    MOV        BASEIV,DURIS ;CREATE DURIS
2769 012600 062767 000002 166354    ADD        #2,BASEIV
2770 012606 016767 166350 167126    MOV        BASEIV,DUTIV ;CREATE DUTIV
2771 012614 062767 000002 166340    ADD        #2,BASEIV
2772 012622 016767 166334 167114    MOV        BASEIV,DUTIS ;CREATE DUTIS
2773 012630 016767 167102 166324    MOV        DURIV,BASEIV ;RESTORE
2774 012636 000207          RTS        PC
2775
2776 012640 000001          OUTCRY 1
2777 012642 006 002          BYTE 6,2
2778 012644 001712          RXCSR
2779
2780          CCC
2781 012646 005067 166530          CLR        $TSTNM ;CLEAR TEST NUMBER
2782 012652 005067 166540          CLR        $ERRPC ;CLEAR LAST ERROR PC
2783 012656 005067 166521          CLR        $ERFLG ;CLEAR ERROR FLAG
2784 012662 005267 166224          INC        PASCNT ;UPDATE PASS COUNT
2785 012666 016767 166220 166206    MOV        PASCNT,LIGHTS ;DISPLAY PASS COUNT
2786 012674 016767 166212 166632    MOV        PASCNT,$PASS ;PASS COUNT TO APT
2787 012702 013701 000042          MOV        #42,R1 ;CHECK FOR ACT-11 OR ODP
2788 012706 001406          BEQ        RESTRT ;IF NO CONTINUE TESTING
2789 012710 000005          RESET
2790 012712 000005          RESET
2791 012714 004711          SENDAD JSR      PC,(R1)
2792 012716 000240          NOP
2793 012720 000240          NOP
2794 012722 000240          NOP
2795 012724          RESTRT
2796 012724 012767 003376 166454    MOV        #TST1+2,$LPADR ;LOAD LAST ADDR
2797 012732 004767 000004          JSR        PC,CKSWR
2798 012736 000167 170346          JMP        BEGIN
2799
2800          ;CHECK SWITCH REGISTER ROUTINE
    
```



```

2801          ,CHECKS TO ALLOW FOR < G> TO ALLOW
2802          ;THE CHANGING OF LOCATION 176
2803
2804 012742 005737 000042          CKSWR  TST      @#42
2805 012746 001040          BNE      OUT
2806 012750 022767 000176 166462  CMP      #SWREG, SWR          ;SOFTWARE SWR PRESENT?
2807 012756 001034          BNE      OUT          ;NO--LEAVE
2808 012760 105777 166460  TSTB     @STKS          ;CHECK TTY READY
2809 012764 100031          BPL      OUT          ;NO--LEAVE
2810 012766 017767 166454 000422  MOV      @STKB, .MSG          ;GET CHARACTER
2811 012774 042767 177600 000414  BIC      #177600, .MSG          ;STRIP JUNK
2812 013002 122767 000007 000406  CMPB     #7, MSG          ;IS IT < G> ?
2813 013010 001017          BNE      OUT          ;NO
2814 013012 104401 016137          TYPE     ,MCNTG
2815 013016 005137 013056          CNTLU.  COM      @#RDSW
2816 013022 104401 016147          TYPE     ,MMSWR
2817 013026 104413          CONVRT
2818 013030 013060          SWREGL
2819 013032 104406 016160  INSTR, MMNEW
2820 013036 104410          PARAM
2821 013040 000000          0
2822 013042 177777          177777
2823 013044 000176          SWREG
2824 013046          000          001          BYTE
2825 013050 005037 013056          OUT     CLR      @#RDSW
2826 013054 000207          RTS      PC
2827 013056 000000          RDSW    WORD    0
2828 013060 000001          SWREGL  1
2829 013062          006          002          BYTE    6, 2
2830 013064 000176          SWREG
2831
2832 013066 000005          5
2833
2834          ,CHECK FOR FREEZE ON CURRENT DATA
2835
2836 013070 004767 177646          SCOP1  JSR      PC, CKSWR
2837 013074 032777 001000 166336  BIT      #SW09, @SWR
2838 013102 001402          BEQ     1$
2839 013104 016716 166000          MOV     LOCK, (SP)
2840 013110 000002          1$    RTI
2841          SBTTL  TYPE ROUTINE
2842
2843          , , *****
2844          , *ROUTINE TO TYPE ASCIZ MESSAGE MESSAGE MUST TERMINATE WITH A 0 BYTE
2845          , *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED
2846          , *NOTE1.          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER
2847          , *NOTE2          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2848          , *NOTE3          $FILLC CONTAINS THE CHARACTER TO FILL AFTER
2849          , *
2850          , *CALL
2851          , *1) USING A TRAP INSTRUCTION
2852          , *          TYPE     , MESADR          , , MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2853          , *OR
2854          , *          TYPE
2855          , *          MESADR
2856          , *
  
```

```

2857
2858 013112 105767 166341 $TYPE TSTB $TFPLG ;; IS THERE A TERMINAL?
2859 013116 100002 BPL 1$ ;; BR IF YES
2860 013120 000000 HALT ;; HALT HERE IF NO TERMINAL
2861 013122 000430 BR 3$ ;; LEAVE
2862 013124 010046 1$ MOV RO,-(SP) ;; SAVE RO
2863 013126 017600 000002 MOV @2(SP),RO ;; GET ADDRESS OF ASCIZ STRING
2864 013132 122767 000001 166406 CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
2865 013140 001011 BNE 62$ ;; NO, GO CHECK FOR APT CONSOLE
2866 013142 132767 000100 166377 BITB #APTPOOL,$ENVM ;; SPOOL MESSAGE TO APT
2867 013150 001405 BEQ 62$ ;; NO, GO CHECK FOR CONSOLE
2868 013152 010067 000004 MOV RO,61$ ;; SETUP MESSAGE ADDRESS FOR APT
2869 013156 004767 000006 JSR PC,$ATY3 ;; SPOOL MESSAGE TO APT
2870 013162 000000 61$ WORD 0 ;; MESSAGE ADDRESS
2871 013164 132767 000040 166355 62$ BITB #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
2872 013172 001003 BNE 60$ ;; YES, SKIP TYPE OUT
2873 013174 112046 2$ MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2874 013176 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
2875 013200 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
2876 013202 012600 60$ MOV (SP)+,RO ;; RESTORE RO
2877 013204 062716 000002 3$ ADD #2,(SP) ;; ADJUST RETURN PC
2878 013210 000002 RTI ;; RETURN
2879 013212 122716 000011 4$ CMPB #HT,(SP) ;; BRANCH IF <HT>
2880 013216 001430 BEQ 8$
2881 013220 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
2882 013224 001006 BNE 5$
2883 013226 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
2884 013230 104401 TYPE ;; TYPE A CR AND LF
2885 013232 001523 $CRLF
2886 013234 105067 000130 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
2887 013240 000755 BR 2$ ;; GET NEXT CHARACTER
2888 013242 004767 000056 5$ JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
2889 013246 126726 166204 6$ CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS ?
2890 013252 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR
2891 013254 016746 166174 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS NEEDED
2892 ;; AND THE NULL CHAR
2893 013260 105366 000001 7$ DECB 1( ) ;; DOES A NULL NEED TO BE TYPED?
2894 013264 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
2895 013266 004767 000032 JSR PC,$TYPEC ;; GO TYPE A NULL
2896 013272 105367 000072 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
2897 013276 000770 BR 7$ ;; LOOP
2898
2899 ,HORIZONTAL TAB PROCESSOR
2900
2901 013300 112716 000040 8$ MOVB #' ,(SP) ;; REPLACE TAB WITH SPACE
2902 013304 004767 000014 9$ JSR PC,$TYPEC ;; TYPE A SPACE
2903 013310 132767 000007 000052 BITB #?, $CHARCNT ;; BRANCH IF NOT AT
2904 013316 001372 BNE 9$ ;; TAB STOP
2905 013320 005726 TST (SP)+ ;; POP SPACE OFF STACK
2906 013322 000724 BR 2$ ;; GET NEXT CHARACTER
2907 013324 105777 166120 $TYPEC TSTB @5TPS ;; WAIT UNTIL PRINTER IS READY
2908 013330 100375 BPL $TYPEC
2909 013332 116677 000002 166112 MOVB 2(SP),@5TPB ;; LOAD CHAR TO BE TYPED INTO DATA REG
2910 013340 12.766 000015 000002 CMPB #CR,2(SP) ;; IS CHARACTER A CARRIAGE RETURN?
2911 013346 001003 BNE 1$ ;; BRANCH IF NO
2912 013350 105067 000014 CLRB $CHARCNT ;; YES--CLEAR CHARACTER COUNT

```

```

2913 013354 000406          BR      $TYPEX      ;EXIT
2914 013356 122766 000012 000002 1$    CMPB   #LF,2(SP)  ;IS CHARACTER A LINE FEED?
2915 013364 001402          BEQ    $TYPEX      ;BRANCH IF YES
2916 013366 105227          INCB  (PC)+      ;COUNT THE CHARACTER
2917 013370 000000          $CHARCNT.WORD 0   ;CHARACTER COUNT STORAGE
2918 013372 000207          $TYPEX RTS      PC
2919
2920
2921          ;ASCII STRING INPUT ROUTINE
2922
2923 013374 017667 000000 000014  INSTR  MOV    @ (SP), MSG      ;PICK UP MESSAGE
2924 013402 062716 000002          ADD    #2,(SP)      ;JUMP AROUND MESSAGE FOR RTI
2925 013406 105767 166134          TSTB  $ENV        ;APT CONTROL
2926 013412 001036          BNE   INSTR2      ;YES NO TYPE
2927 013414 104401          INST1 TYPE
2928 013416 000000          MSG    0
2929 013420 012704 016172          MOV    #INBUF,R4   ;GET STARTING LOC OF INBUF
2930 013424 012703 000007          MOV    #7,R3      ;MAX # OF CHARS
2931 013430 105777 166010 1$    TSTB  @ $TKS , TTY FLAG
2932 013434 100375          BPL   1$
2933 013436 117714 166004          MOVB  @ $TKB,(R4)  ;TAKE CHAR
2934 013442 142714 000200          BICB  #200,(R4)   ;STRIP
2935 013446 121427 000025          CMPB  (R4),#25    ;IS IT < G>
2936 013452 001760          BEQ   INST1
2937 013454 122427 000015          CMPB  (R4)+,#15   ;CHECK FOR CR
2938 013460 001413          BEQ   INSTR2
2939 013462 105777 165762 2$    TSTB  @ $TPS , TEST FLAG
2940 013466 100375          BPL   2$
2941 013470 117777 165752 165754  MOVB  @ $TKB,@ $TPB ;ECHO CHARACTER
2942 013476 005303          DEC   R3          ;DID YOU TYPE TOO MANY CHARS ?
2943 013500 001353          BNE   1$
2944 013502 104401          INSTE TYPE
2945 013504 015457          MQM  ,?
2946 013506 000742          BR   INST1 ,RETRY
2947 013510 000002          INSTR2 RTI
2948
2949          ;CONVERT ASCII STRING TO OCTAL
2950
2951 013512 011605          PARAM MOV    (SP),R5 ,PUT CONTENTS OF SP INTO R5
2952 013514 012567 000162          MOV    (R5)+,LOLIM ;PUT LOW LIMIT INTO LOLIM
2953 013520 012567 000160          MOV    (R5)+,HILIM ;PUT HIGH LIMIT INTO HILIM
2954 013524 012567 000156          MOV    (R5)+,DEVADR ;PUT STORE LOC INTO DEVADR
2955 013530 112567 000154          MOVB  (R5)+,LOBITS ;PUT MASK INTO LOBITS
2956 013534 112567 000151          MOVB  (R5)+,ADRCNT ;PUT COUNT INTO ADRCNT
2957 013540 010516          MOV    R5,(SP) ,RESTORE RETURN ADDR ON STACK FOR RTI
2958 013542 005005          PARAM1 CLR   R5
2959 013544 012704 016172          MOV    #INBUF,R4
2960 013550 122714 000015          CMPB  #15,(R4)   ;CR ?
2961 013554 001420          BEQ   PARERR ,YOU TYPED CR TOO SOON !
2962 013556 121427 000060 1$    CMPB  (R4),#60   ;LOW LIMIT ASCII 0
2963 013562 002415          BLT   PARERR
2964 013564 121427 000067          CMPB  (R4),#67   ;HIGH LIMIT ASCII 7
2965 013570 003012          BGT   PARERR
2966 013572 142714 000060          BICB  #60,(R4)   ;CONVERT TO OCTAL
2967 013576 152405          BISB  (R4)+,R5   ;STORE AWAY ITS AN OF CHAR
2968 013600 122714 000015          CMPB  #15,(R4)   ;CR ?

```

```

2969 013604 001414          BEQ     LIMITS ,NOW CHECK FOR HIGH &LOW LIMIT CONDS
2970 013606 006305          ASL     R5     ,ALLOCATE ROOM FOR NEXT CHAP
2971 013610 006305          ASL     R5
2972 013612 006305          ASL     R5
2973 013614 000760          BR      1$
2974 013616 122714 000015    PARERR  CMPB   #15,(R4) ,CR?
2975 013622 001003          BNE     120$
2976 013624 005737 013056    TST     @#RDSW ,CK SWR USED
2977 013630 001023          BNE     PARTI
2978 013632 104407          120$   INSTER ,RETRY
2979 013634 000742          BR      PARAM1
2980
2981                          ,TEST TO SEE IF NUMBER IS WITHIN LIMITS
2982
2983 013636 020567 000042    LIMITS  CMP     R5,HILIM
2984 013642 101365          BHI     PARERR ,THE # IS TOO HIGH
2985 013644 020567 000032    CMP     R5,LOLIM
2986 013650 103762          BLO     PARERR ,THE # IS TOO LOW
2987 013652 136705 000032    BITB   LOBITS,R5 ,TEST BY MASKINGTHE #
2988 013656 001357          BNE     PARERR
2989
2990                          ,STORE NUMBER AT SPECIFIED ADDRESS
2991
2992 013660 016704 000022          MOV     DEVADR,R4 ,GET STARTING ADDR OF
2993 013664 010524          1$     MOV     R5,(R4)+ ,STORE AT THIS ADDR
2994 013666 062705 000002    ADD     #2,R5
2995 013672 105367 000013    DECB   ADCNT ,HOW MANY TIMES + 2 ?
2996 013676 001372          BNE     1$
2997 013700 000002          PARTI  RTI
2998 013702 000000          LOLIM  0
2999 013704 000000          HILIM  0
3000 013706 000000          DEVADR 0
3001 013710 000000          LOBITS 0
3002                          ADCNT=LOBITS+1
3003
3004                          ,SAVE PC OF TEST THAT FAILED AND RO-R5
3005
3006 013712 016667 000004 165206  SAV05. MOV     4(SP),SAVPC
3007
3008                          ,SAVE RO-R5
3009
3010 013720 010567 165550          SV05   MOV     R5,$REG5
3011 013724 010467 165542          MOV     R4,$REG4
3012 013730 010367 165534          MOV     R3,$REG3
3013 013734 010267 165526          MOV     R2,$REG2
3014 013740 010167 165520          MOV     R1,$REG1
3015 013744 010067 165512          MOV     R0,$REG0
3016 013750 000002          RTI
3017
3018                          ,RESTORE RO-R5
3019
3020 013752 016700 165504          RES05  MOV     $REG0,R0
3021 013756 016701 165502          MOV     $REG1,R1
3022 013762 016702 165500          MOV     $REG2,R2
3023 013766 016703 165476          MOV     $REG3,R3
3024 013772 016704 165474          MOV     $REG4,R4
  
```

```

3025 013776 016705 165472      MOV      $REG5, R5
3026 014002 000002      RTI
3027
3028                          ; CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3029
3030 014004 104401      CONVR  TYPE
3031 014006 015463      MCRLF  ; CR LF
3032 014010 017601 000000      MOV      @ (SP), R1      , PICK UP DATA POINTER
3033 014014 062716 000002      ADD      #2, (SP)      , SET UP SP FOR RTI
3034 014020 012167 000130      MOV      (R1)+, WRDCNT  ; PICK UP # OF WORDS FROM TABLE
3035 014024 112167 000126      1$      MOV      (R1)+, CHRCNT  ; PICK UP # OF CHARS FROM TABLE
3036 014030 112167 000123      MOV      (R1)+, SPACNT  ; PICK UP # OF SPACES FROM TABLE
3037 014034 013167 000120      MOV      @ (R1)+, BINWRD ; PICK UP ADDRESS OF MSG
3038                          ; FROM TABLE
3039 014040 016704 000114      2$      MOV      BINWRD, R4      ; SAVE
3040 014044 116705 000106      MOV      CHRCNT, R5      ; SAVE
3041 014050 012700 016234      MOV      #TEMP, R0      , STARTING ADDRESS OF TEMP BLOCK
3042 014054 010403      3$      MOV      R4, R3          ; SAVE
3043 014056 042703 177770      BIC      #177770, R3     , CLR OUT UPPER BITS      SAVE CHAR
3044 014062 062703 000260      ADD      #260, R3      , CONVERT TO ASCII
3045 014066 110320      MOV      R3, (R0)+      ; STORE AWAY
3046 014070 006204      ASR      R4          ; SHIFT FOR NEXT #
3047 014072 006204      ASR      R4          ; DITTO
3048 014074 006204      ASR      R4          ; DITTO
3049 014076 005305      DEC      R5          ; DEC CHAR COUNT
3050 014100 001365      BNE      3$          ; DO IT AGAIN ?
3051 014102 012703 016276      MOV      #MDATA, R3     ; STARTING ADDRESS OF MDATA BLOCK
3052 014106 114023      4$      MOV      -(R0), (R3)+    , REVERSE THE ORDER OF NUMBERS
3053 014110 105367 000042      DECB    CHRCNT      , DEC CHAR COUNT
3054 014114 001374      BNE      4$          ; DO IT AGAIN ?
3055 014116 105767 000035      TSTB    SPACNT      ; HOW MANY SPACES ?
3056 014122 001405      BEQ      6$          ; TYPE # IF BR = 0
3057 014124 112723 000240      5$      MOV      #240, (R3)+    , "SPACE" IN ASCII
3058 014130 105367 000023      DECB    SPACNT      ; DEC # OF SPACE COUNT
3059 014134 001373      BNE      5$          ; DO IT AGAIN ?
3060 014136 105013      6$      CLRB    (R3)        ; INSERT "0" FOR TTY OUTPUT ROUTINE
3061 014140 104401      TYPE
3062 014142 016276      MDATA  , THIS MESSAGE
3063 014144 005367 000004      DEC      WRDCNT      , HOW MANY #'S ?
3064 014150 001325      BNE      1$          ; DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3065 014152 000002      RTI      , RETURN TO PROGRAM
3066 014154 000000      WRDCNT  0
3067 014156 000000      CHRCNT  0
3068 014157 014157      SPACNT=CHRCNT+1
3069 014160 000000      BINWRD  0
3070
3071                          ; COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3072                          ; BUFFER TO THE CHARACTERS "N" AND "Y"
3073                          ; IF THE CHARACTER IS "N" CLEAR THE FLAG
3074                          ; IF THE CHARACTER IS "Y" SET THE FLAG
3075
3076 014162 017605 000000      SETFLG MOV      @ (SP), R5
3077 014166 122767 000116 001776      CMPB    #'N, INBUF     , IS IT "N" ?
3078 014174 001002      BNE      1$
3079 014176 105015      CLRB    (R5)          , 000
3080 014200 000406      BR      2$

```

```

3081 014202 122767 000131 001762 15:  CMPB  #'Y, INBUF      ; IS IT "Y" ?
3082 014210 001005                BNE    35
3083 014212 112715 177777        MOVB   #-1, (R5)      ; 377
3084 014216 062716 000002        ADD    #2, (SP)
3085 014222 000002                RTI
3086 014224 104407        35:  INSTER ;RETRY
3087 014226 000755                BR     .SETFLG
3088                .SBTTL  ERROR HANDLER ROUTINE
3089
3090                ; *****
3091                ; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3092                ; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3093                ; *AND GO TO SAVIT ON ERROR
3094                ; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3095                ; *SW15=1      HALT ON ERROR
3096                ; *SW13=1      INHIBIT ERROR TYPEOUTS
3097                ; *SW10=1      BELL ON ERROR
3098                ; *SW09=1      LOOP ON ERROR
3099                ; *CALL
3100                ; *      ERROR  N      .. ERROR=EMT AND N=ERROR ITEM NUMBER
3101
3102 014230                $ERROR
3103 014230 105267 165147        75:  INCB   $ERFLG      .. SET THE ERROR FLAG
3104 014234 001775                BEQ    75            .. DON'T LET THE FLAG GO TO ZERO
3105 014236 016777 165140 165176  MOV    $STNM, @DISPLAY .. DISPLAY TEST NUMBER AND ERROR FLAG
3106 014244 032777 002000 165166  BIT    #BIT10, @SWR   .. BELL ON ERROR?
3107 014252 001402                BEQ    15           .. NO - SKIP
3108 014254 104401 001516        TYPE   , $BELL      .. RING BELL
3109 014260 005267 165126        15:  INC    $ERTTL     .. COUNT THE NUMBER OF ERRORS
3110 014264 011667 165126        MOV    (SP), $ERRPC  .. GET ADDRESS OF ERROR INSTRUCTION
3111 014270 162767 000002 165120  SUB    #2, $ERRPC
3112 014276 117767 165114 165110  MOVB   @ $ERRPC, $ITEMB .. STRIP AND SAVE THE ERROR ITEM CODE
3113 014304 032777 020000 165126  BIT    #BIT13, @SWR   .. SKIP TYPEOUT IF SET
3114 014312 001004                BNE    205         .. SKIP TYPEOUTS
3115 014314 004767 000072        JSR    PC, SAVIT     .. GO TO USER ERROR ROUTINE
3116 014320 104401 001523        TYPE   , $CRLF
3117 014324                205
3118 014324 122767 000001 165214  CMPB   #APTENV, $ENV  .. RUNNING IN APT MODE
3119 014332 001007                BNE    25           .. NO, SKIP APT ERROR REPORT
3120 014334 116767 165054 000004  MOVB   $ITEMB, 215   .. SET ITEM NUMBER AS ERROR NUMBER
3121 014342 004767 000016        JSR    PC, $ATY4     .. REPORT FATAL ERROR TO APT
3122 014346 000                215  BYTE   0
3123 014347 000                BYTE   0
3124 014350 000777                225  BR     225         .. APT ERROR LOOP
3125 014352 005777 165062        25:  TST    @SWR        .. HALT ON ERROR
3126 014356 100001                BPL    35           .. SKIP IF CONTINUE
3127 014360 000000                HALT   .. HALT ON ERROR!
3128 014362 032777 001000 165050  35:  BIT    #BIT09, @SWR .. LOOP ON ERROR SWITCH SET?
3129 014370 001402                BEQ    45           .. BR IF NO
3130 014372 016716 165012        MOV    $LPERR, (SP)  .. FUDGE RETURN FOR LOOPING
3131 014376 005767 165112        45:  TST    $ESCAPE   .. CHECK FOR AN ESCAPE ADDRESS
3132 014402 001402                BEQ    55           .. BR IF NONE
3133 014404 016716 165104        MOV    $ESCAPE, (SP) .. FUDGE RETURN ADDRESS FOR ESCAPE
3134 014410                55
3135 014410 000002                RTI     .. RETURN
3136 014412 010067 164512        SAVIT. MOV    RO, HLDO
  
```

```

3137 014416 010167 164510      MOV      R1,HL D1
3138 014422 010267 164506      MOV      R2,HL D2
3139 014426 010367 164504      MOV      R3,HL D3
3140 014432 010467 164502      MOV      R4,HL D4
3141 014436 010567 164500      MOV      R5,HL D5
3142 014442 016767 164734 164474  MOV      $TSTNM,HL D6
3143
3144      SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
3145
3146      ;,*****
3147      ;,THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3148      ;,ERROR IS TO BE REPORTED IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3149      ;,AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR
3150
3151 014450      SERRTYP
3152 014450 104401 001523      TYPE      , $CRLF      ;,"CARRIAGE RETURN" & "LINE FEED"
3153 014454 010046      MOV      RO,-(SP)      ;,SAVE RO
3154 014456 005000      CLR      RO      ;,PICKUP THE ITEM INDEX
3155 014460 153700 001414      BISB     @#$ITEMB,RO
3156 014464 001004      BNE      1$      ;,IF ITEM NUMBER IS ZERO, JUST
3157      ;,TYPE THE PC OF THE ERROR
3158 014466 016746 164724      MOV      $ERRPC,-(SP) ;,SAVE $ERRPC FOR TYPEOUT
3159      ;,ERROR ADDRESS
3160 014472 104402      TYPOC     ;,GO TYPE--OCTAL ASCII(ALL DIGITS)
3161 014474 000426      BR       6$      ;,GET OUT
3162 014476 005300 1$      DEC      RO      ;,ADJUST THE INDEX SO THAT IT WILL
3163 014500 006300      ASL      RO      ;,
3164 014502 006300      ASL      RO      ;,
3165 014504 006300      ASL      RO      ;,
3166 014506 062700 001652      ADD      #$ERRTB,RO ;,FORM TABLE POINTER
3167 014512 012067 000004      MOV      (RO)+,2$ ;,PICKUP "ERROR MESSAGE" POINTER
3168 014516 001404      BEQ      3$      ;,SKIP TYPEOUT IF NO POINTER
3169 014520 104401      TYPE     ;,TYPE THE "ERROR MESSAGE"
3170 014522 000000 2$      WORD    0      ;,"ERROR MESSAGE" POINTER GOES HERE
3171 014524 104401 001523      TYPE     , $CRLF ;,"CARRIAGE RETURN" & "LINE FEED"
3172 014530 012067 000004 3$      MOV      (RO)+,4$ ;,PICKUP "DATA HEADER" POINTER
3173 014534 001404      BEQ      5$      ;,SKIP TYPEOUT IF 0
3174 014536 104401      TYPE     ;,TYPE THE "DATA HEADER"
3175 014540 000000 4$      WORD    0      ;,"DATA HEADER" POINTER GOES HERE
3176 014542 104401 001523      TYPE     , $CRLF ;,"CARRIAGE RETURN" & "LINE FEED"
3177 014546 011000 5$      MOV      (RO),RO ;,PICKUP "DATA TABLE" POINTER
3178 014550 001004      BNE      7$      ;,GO TYPE THE DATA
3179 014552 012600 6$      MOV      (SP)+,RO ;,RESTORE RO
3180 014554 104401 001523      TYPE     , $CRLF ;,"CARRIAGE RETURN" & "LINE FEED"
3181 014560 000207      RTS      PC      ;,RETURN
3182 014562 7$
3183 014562 013046      MOV      @ (RO)+,-(SP) ;,SAVE @ (RO)+ FOR TYPEOUT
3184 014564 104402      TYPOC     ;,GO TYPE--OCTAL ASCII(ALL DIGITS)
3185 014566 005710      TST      (RO)      ;,IS THERE ANOTHER NUMBER?
3186 014570 001770      BEQ      6$      ;,BR IF NO
3187 014572 104401 014600      TYPE     ,8$      ;,TYPE TWO(2) SPACES
3188 014576 000771      BR       7$      ;,LOOP
3189 014600 020040 000      8$      ASCIIZ  / /      ;,TWO(2) SPACES
3190      EVEN
3191      SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3192

```

```

3193      , , *****
3194      , *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3195      , *OCTAL (ASCII) NUMBER AND TYPE IT.
3196      , *STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3197      , *CALL
3198      , *      MOV      NUM, -(SP)      ;, NUMBER TO BE TYPED
3199      , *      TYPOS    ;, CALL FOR TYPEOUT
3200      , *      BYTE    N                ;, N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3201      , *      BYTE    M                ;, M=1 OR 0
3202      , *                                     ;, 1=TYPE LEADING ZEROS
3203      , *                                     ;, 0=SUPPRESS LEADING ZEROS
3204      , *
3205      , *STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3206      , *STYPOS OR STYPOC
3207      , *CALL
3208      , *      MOV      NUM, -(SP)      ;, NUMBER TO BE TYPED
3209      , *      TYPON    ;, CALL FOR TYPEOUT
3210      , *
3211      , *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3212      , *CALL
3213      , *      MOV      NUM, -(SP)      ;, NUMBER TO BE TYPED
3214      , *      TYPOC    ;, CALL FOR TYPEOUT
3215
3216 014604 017646 000000      STYPOS  MOV      @ (SP), -(SP)      ;, PICKUP THE MODE
3217 014610 116667 000001 000211      MOVVB   1 (SP), $OFILL      ;, LOAD ZERO FILL SWITCH
3218 014616 112667 00C207      MOVVB   (SP)+, $OMODE+1    ;, NUMBER OF DIGITS TO TYPE
3219 014622 062716 000U02      ADD     #2, (SP)         ;, ADJUST RETURN ADDRESS
3220 014626 000406      BR      $TYPON
3221 014630 112767 000001 000171      STYPOC  MOVVB   #1, $OFILL      ;, SET THE ZERO FILL SWITCH
3222 014636 112767 000006 000165      MOVVB   #6, $OMODE+1    ;, SET FOR SIX(6) DIGITS
3223 014644 112767 000005 000154      STYPON  MOVVB   #5, $OCNT      ;, SET THE ITERATION COUNT
3224 014652 010346      MOV     R3, -(SP)       ;, SAVE R3
3225 014654 010446      MOV     R4, -(SP)       ;, SAVE R4
3226 014656 010546      MOV     R5, -(SP)       ;, SAVE R5
3227 014660 116704 000145      MOVVB   $OMODE+1, R4    ;, GET THE NUMBER OF DIGITS TO TYPE
3228 014664 005404      NEG     R4
3229 014666 062704 000006      ADD     #6, R4          ;, SUBTRACT IT FOR MAX ALLOWED
3230 014672 110467 000132      MOVVB   R4, $OMODE      ;, SAVE IT FOR USE
3231 014676 116704 000125      MOVVB   $OFILL, R4      ;, GET THE ZERO FILL SWITCH
3232 014702 016605 000012      MOV     12(SP), R5      ;, PICKUP THE INPUT NUMBER
3233 014706 005003      CLR     R3              ;, CLEAR THE OUTPUT WORD
3234 014710 006105      15     ROL     R5        ;, ROTATE MSB INTO "C"
3235 014712 000404      BR      3$             ;, GO DO MSB
3236 014714 006105      25     ROL     R5        ;, FORM THIS DIGIT
3237 014716 006105      ROL     R5
3238 014720 006105      ROL     R5
3239 014722 010503      MOV     R5, R3
3240 014724 006103      35     ROL     R3        ;, GET LSB OF THIS DIGIT
3241 014726 105367 000076      DECB   $OMODE          ;, TYPE THIS DIGIT?
3242 014732 100016      BPL    7$              ;, BR IF NO
3243 014734 042703 177770      BIC    #177770, R3     ;, GET RID OF JUNK
3244 014740 001002      BNE    4$              ;, TEST FOR 0
3245 014742 005704      TST    R4              ;, SUPPRESS THIS 0?
3246 014744 001403      BEQ    5$              ;, BR IF YES
3247 014746 005204      45     INC     R4        ;, DON'T SUPPRESS ANYMORE 0'S
3248 014750 052703 000060      BIS    #'0, R3        ;, MAKE THIS DIGIT ASCII

```



```

3249 014754 052703 000040      5$    BIS      #' ,R3      ;; MAKE ASCII IF NOT ALREADY
3250 014760 110367 000040      MOV8     R3,8$      ;; SAVE FOR TYPING
3251 014764 104401 015024      TYPE     ,8$       ;; GO TYPE THIS DIGIT
3252 014770 105367 000032      7$    DECB     $OCNT    ;; COUNT BY 1
3253 014774 003347      BGT      2$        ;; BR IF MORE TO DO
3254 014776 002402      BLT      6$        ;; BR IF DONE
3255 015000 005204      INC      R4        ;; INSURE LAST DIGIT ISN'T A BLANK
3256 015002 000744      BR       2$        ;; GO DO THE LAST DIGIT
3257 015004 012605      6$    MOV      (SP)+,R5  ;; RESTORE R5
3258 015006 012604      MOV      (SP)+,R4  ;; RESTORE R4
3259 015010 012603      MOV      (SP)+,R3  ;; RESTORE R3
3260 015012 016666 000002 000004      MOV      2(SP),4(SP) ;; SET THE STACK FOR RETURNING
3261 015020 012616      MOV      (SP)+,(SP)
3262 015022 000002      RTI                    ;; RETURN
3263 015024      000      8$    BYTE     0        ;; STORAGE FOR ASCII DIGIT
3264 015025      000      BYTE     0        ;; TERMINATOR FOR TYPE ROUTINE
3265 015026      000      $OCNT    BYTE     0        ;; OCTAL DIGIT COUNTER
3266 015027      000      $OFILL   BYTE     0        ;; ZERO FILL SWITCH
3267 015030 000000      $OMODE   WORD      0        ;; NUMBER OF DIGITS TO TYPE
3268
3269
3270
3271 015032      SPWRDN
3272 015032 010046      PFAIL   MOV      R0,-(SP)      ; SAVE R0-R5 ON PROCESSOR STACK
3273 015034 010146      MOV      R1,-(SP)
3274 015036 010246      MOV      R2,-(SP)
3275 015040 010346      MOV      R3,-(SP)
3276 015042 010446      MOV      R4,-(SP)
3277 015044 010546      MOV      R5,-(SP)
3278 015046 016746 162752      MOV      24,-(SP)
3279 015052 010667 164040      MOV      SP,SAVSP      ; SAVE STACK POINTER
3280 015056 012767 015070 162740      MOV      #RESTART,24  ; SET UP FOR POWER UP TRAP
3281 015064 000000      HALT                    ; HALT ON POWER DOWN NORMAL
3282 015066 000777      BR
3283
3284      ; PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3285
3286 015070 016706 164022      RESTAR  MOV      SAVSP,SP      ; RESTORE STACK POINTER
3287 015074 012605      MOV      (SP)+,R5      ; RESTORE R0-R5
3288 015076 012604      MOV      (SP)+,R4
3289 015100 012603      MOV      (SP)+,R3
3290 015102 012602      MOV      (SP)+,R2
3291 015104 012601      MOV      (SP)+,R1
3292 015106 012600      MOV      (SP)+,R0
3293 015110 012767 015032 162706      MOV      #.PFAIL,24  ; SET UP FOR POWER FAILURE
3294 015116 106427 000340      MTPS     #340
3295 015122 012706 001100      MOV      #STACK,SP
3296 015126 005067 001102      CLR      TEMP
3297 015132 005267 001076      INC      TEMP
3298 015136 001375      BNE      -4
3299 015140 104413      CONVRT
3300 015142 015164      PFTAB
3301 015144 104401      TYPE
3302 015146 015466      MPFAIL
3303 015150 005067 164227      CLR      $ERFLG
3304 015154 005067 164236      CLR      $ERRPC

```

3305	015160	000177	163720						
3306	015164	000001			PFTAB	JMP	RETURN		
3307	015166	006	002			1			
3308	015170	000207				BYTE	6.2		
3309	015172	005015	042012	053125	MTITLE.	RETURN			
3310	015200	030461	042040	042132		ASCIZ	<15><12><12>/DUV11 DZDUU-B TAPE E /<15><12>		
3311	015206	052525	041055	052040					
3312	015214	050101	020105	020105					
3313	015222	005015	000						
3314	015225	015	053012	041505	MVECTO.	ASCIZ	<15><12>/VEC ADD- /		
3315	015232	040440	042104	000055					
3316	015240	005015	051461	020124	MREGAD	ASCIZ	<15><12>/1ST DEV REC CSR ADD- /		
3317	015246	042504	035126	051040					
3318	015254	041505	041440	051123					
3319	015262	040440	042104	000055					
3320	015270	005015	052515	052114	MMULT	ASCIZ	<15><12>/MULT DEV ? (Y OR N)- /		
3321	015276	042040	053105	037440					
3322	015304	024040	020131	051117					
3323	015312	047040	026451	000					
3324	015317	015	046012	051501	MLASTD	ASCIZ	<15><12>/LAST DEV REC CSR ADDR- /		
3325	015324	020124	042504	035126					
3326	015332	051040	041505	041440					
3327	015340	051123	040440	042104					
3328	015346	026522	000						
3329	015351	075	042504	044526	DEVICE	ASCIZ	/=DEVICE /		
3330	015356	042503	020040	000					
3331	015363	015	051412	046105	MCOV	ASCIZ	<15><12>/SELECT TO RUN @ACTREG /		
3332	015370	041505	020124	047524					
3333	015376	051040	047125	040040					
3334	015404	041501	051124	043505					
3335	015412	000							
3336	015413	015	047412	043126	MRANGE	ASCIZ	<15><12>/OVFLO RETYPE LAST DEV RXCSR ADDS- /		
3337	015420	047514	051072	052105					
3338	015426	050131	020105	040514					
3339	015434	052123	042040	053105					
3340	015442	051040	041530	051123					
3341	015450	040440	042104	026523					
3342	015456	000							
3343	015457	040	037440	000	MQM:	ASCIZ	/ ? /		
3344	015463	015	000012		MCRLF:	ASCIZ	<15><12>		
3345	015466	043120	044501	026114	MPFAIL	ASCIZ	/PFAIL, RESTART AT TEST IN PROGRESS /		
3346	015474	020040	042522	052123					
3347	015502	051101	020124	052101					
3348	015510	052040	051505	020124					
3349	015516	047111	050040	047522					
3350	015524	051107	051505	000123					
3351	015532	005015	047105	020104	MEPASS	ASCIZ	<15><12>/END OF PASS TAPE E /		
3352	015540	043117	050040	051501					
3353	015546	020123	040524	042520					
3354	015554	042440	000						
3355	015557	015	051012	000	MR	ASCIZ	<15><12>/R /		
3356	015563	015	052012	051505	MTSTPC	ASCIZ	<15><12>/TEST PC- /		
3357	015570	020124	041520	000055					
3358	015576	005015	047514	045503	MLOCK	ASCIZ	<15><12>/LOCK ON TEST? (Y OR N)- /		
3359	015604	047440	020116	052040					
3360	015612	051505	037524	024040					

```

3361 015620 020131 051117 047040
3362 015626 026451 000
3363 015631 015 021412 047440 MSYNC. . ASCIIZ <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3364 015636 020106 054523 041516
3365 015644 041440 040510 051522
3366 015652 051440 046105 041505
3367 015660 042524 020104 020050
3368 015666 020061 051117 031040
3369 015674 026451 000
3370 015677 015 044412 020123 MWIRE6. . ASCIIZ <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3371 015704 042523 020103 046530
3372 015712 052111 051440 044527
3373 015720 041524 020110 032505
3374 015726 026465 020062 047111
3375 015734 020077 054450 047440
3376 015742 020122 024516 000055
3377 015750 005015 051511 051440 MWIRE5 ASCIIZ <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3378 015756 041505 051040 041505
3379 015764 051440 044527 041524
3380 015772 020110 032505 026465
3381 016000 020063 047111 020077
3382 016006 054450 047440 020122
3383 016014 024516 000055
3384 016020 005015 051511 047440 MWIRE4. . ASCIIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3385 016026 052120 041440 051114
3386 016034 042440 040516 046102
3387 016042 020105 053523 052111
3388 016050 044103 042440 032465
3389 016056 030455 044440 037516
3390 016064 024040 020131 051117
3391 016072 047040 026451 000
3392 016077 015 005012 031510 MEXTJ ASCIIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3393 016104 032461 041440 047117
3394 016112 042516 052103 051117
3395 016120 047440 020116 024077
3396 016126 020131 051117 047040
3397 016134 026451 000
3398 016137 015 020012 043536 MCNTG. . ASCIIZ <15><12>/ G /
3399 016144 020040 000
3400 016147 040 053523 036522 MMSWR . ASCIIZ / SWR= /
3401 016154 020040 000040
3402 016160 020040 047040 053505 MMNEW. ASCIIZ / NEW= /
3403 016166 020075 000040
3404 EVEN
3405
3406 , BUFFERS FOR INPUT-OUTPUT
3407
3408 016172 000000 INBUF 0
3409 016234 016234 = +40
3410 016234 000000 TEMP. 0
3411 016276 016276 = +40
3412 016276 000000 MDATA 0
3413 016340 016340 = +40
3414 SBTTL SCOPE HANDLER ROUTINE
3415
3416 , , *****

```

```

3417 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3418 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG (DISPLAY<7 0>)
3419 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15 08>
3420 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE.
3421 ;*SW14=1 LOOP ON TEST
3422 ;*SW11=1 INHIBIT ITERATIONS
3423 ;*SW09=1 LOOP ON ERROR
3424 ;*SW08=1 LOOP ON TEST IN SWR<7 0>
3425 ;*CALL
3426 ;* SCOPE ;:SCOPE=10T
3427
3428 016340 $$SCOPE
3429
3430 ; SCOPE LOOP AND INTERATION HANDLER
3431
3432 016340 SCOPE
3433 016340 004767 174376 JSR PC,CKSWR
3434 016344 005067 163046 CLR $ERRPC ;CLEAR LAST ERROR PC
3435 016350 022716 003376 CMP #TST1+2, (SP) ;IS SCOPE AT BEGINING OF TEST 1?
3436 016354 001422 BEQ $XTSTR ;YES NO LOOP.
3437
3438 016356 032777 040000 163054 TTST BIT #BIT14, @SWR ; THIS CODE IS FOR TESTING FOR BIT 14
3439 016364 001412 BEQ 1$ ; ON LSI WHICH SYSMAC CANNOT HANDLE
3440 016366 016767 163010 163012 MOV $TSTNM, $LPADR
3441 016374 000406 BR 1$
3442 016376 105777 163042 TSTB @STKS ; KEYBOARD DONE?
3443 016402 100123 BPL $OVER ; BR IF NO
3444 016404 017766 163036 177776 MOV @STKB, -2(SP) ; CLEAR DONE BIT
3445 016412 032777 040000 163020 1$. BIT #BIT14, @SWR ; LOOP ON PRESENT TEST?
3446 016420 001114 BNE $OVER ; YES IF SW14=1
3447 ;*****START OF CODE FOR THE XOR TESTER*****
3448 016422 000416 $XTSTR BR 6$ ; IF RUNNING ON THE "XOR" TESTER CHANGE
3449 ; THIS INSTRUCTION TO A "NOP" (NOP=240)
3450 016424 013746 000004 MOV @ERRVEC, -(SP) ; SAVE THE CONTENTS OF THE ERROR VECTOR
3451 016430 012737 016450 000004 MOV #5$, @ERRVEC ; SET FOR TIMEOUT
3452 016436 005737 177060 TST @#177060 ; TIME OUT ON XOR?
3453 016442 012637 000004 MOV (SP)+, @ERRVEC ; RESTORE THE ERROR VECTOR
3454 016446 000463 BR $$VLAD ; GO TO THE NEXT TEST
3455 016450 022626 5$ CMP (SP)+, (SP)+ ; CLEAR THE STACK AFTER A TIME OUT
3456 016452 012637 000004 MOV (SP)+, @ERRVEC ; RESTORE THE ERROR VECTOR
3457 016456 000423 BR 7$ ; LOOP ON THE PRESENT TEST
3458 016460 6$ ;*****END OF CODE FOR THE XOR TESTER*****
3459 016460 032777 000400 162752 BIT #BIT08, @SWR ; LOOP ON SPEC TEST?
3460 016466 001404 BEQ 2$ ; BR IF NO
3461 016470 127767 162744 162704 CMPB @SWR, $TSTNM ; ON THE RIGHT TEST? SWR<7 0>
3462 016476 001465 BEQ $OVER ; BR IF YES
3463 016500 105767 162677 2$ TSTB $ERFLG ; HAS AN ERROR OCCURRED?
3464 016504 001421 BEQ 3$ ; BR IF NO
3465 016506 126767 162703 162667 CMPB $ERMAX, $ERFLG ; MAX ERRORS FOR THIS TEST OCCURRED?
3466 016514 101015 BHI 3$ ; BR IF NO
3467 016516 032777 001000 162714 BIT #BIT09, @SWR ; LOOP ON ERROR?
3468 016524 001404 BEQ 4$ ; BR IF NO
3469 016526 016767 162656 162652 7$ MOV $LPERR, $LPADR ; SET LOOP ADDRESS TO LAST SCOPE
3470 016534 000446 BR $OVER
3471 016536 105067 162641 4$ CLRB $ERFLG ; ZERO THE ERROR FLAG
3472 016542 005067 162744 CLR $TIMES ; CLEAR THE NUMBER OF ITERATIONS TO MAKE

```

```

3473 016546 000415 BR 1$ // ESCAPE TO THE NEXT TEST
3474 016550 032777 004000 162662 3$ BIT #BIT11, @SWR // INHIBIT ITERATIONS?
3475 016556 001011 BNE 1$ // BR IF YES
3476 016560 005767 162750 TST $PASS // IF FIRST PASS OF PROGRAM
3477 016564 001406 BEQ 1$ // INHIBIT ITERATIONS
3478 016566 005267 162612 INC $ICNT // INCREMENT ITERATION COUNT
3479 016572 026767 162714 162604 CMP $TIMES, $ICNT // CHECK THE NUMBER OF ITERATIONS MADE
3480 016600 002024 BGE $OVER // BR IF MORE ITERATION REQUIRED
3481 016602 012767 000001 162574 1$ MOV #1, $ICNT // REINITIALIZE THE ITERATION COUNTER
3482 016610 016767 000056 162674 MOV $MXCNT, $TIMES // SET NUMBER OF ITERATIONS TO DO
3483 016616 105267 162560 $SVLAD INCB $STSTM // COUNT TEST NUMBERS
3484 016622 116767 162554 162702 MOV $STSTM, $TESTN // SET TEST NUMBER IN APT MAILBOX
3485 016630 011667 162552 MOV (SP), $LPADR // SAVE SCOPE LOOP ADDRESS
3486 016634 011667 162550 MOV (SP), $LPERR // SAVE ERROR LOOP ADDRESS
3487 016640 005067 162650 CLR $ESCAPE // CLEAR THE ESCAPE FROM ERROR ADDRESS
3488 016644 112767 000001 162543 MOV #1, $ERMAX // ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3489 016652 016777 162524 162562 $OVER MOV $STSTM, @DISPLAY // DISPLAY TEST NUMBER
3490 016660 016716 162522 MOV $LPADR, (SP) // FUDGE RETURN ADDRESS
3491 016664 000002 4$ RTI
3492 016666 001407 BRW 1407
3493 016670 000432 BRX 432
3494 016672 000005 $MXCNT 5 // MAX NUMBER OF ITERATIONS
3495 SBTTL TRAP DECODER
3496
3497 // *****
3498 // *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3499 // *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3500 // *OF THE DESIRED ROUTINE THEN USING THE ADDRESS OBTAINED IT WILL
3501 // *GO TO THAT ROUTINE
3502
3503 016674 010046 $TRAP MOV RO, -(SP) // SAVE RO
3504 016676 016600 000002 MOV 2(SP), RO // GET TRAP ADDRESS
3505 016702 005740 TST -(RO) // BACKUP BY 2
3506 016704 111000 MOV (RO), RO // GET RIGHT BYTE OF TRAP
3507 016706 006300 ASL RO // POSITION FOR INDEXING
3508 016710 016000 016730 MOV $TRPAD(RO), RO // INDEX TO TABLE
3509 016714 000200 RTS RO // GO TO ROUTINE
3510
3511
3512 // THIS IS USE TO HANDLE THE "GETPRI" MACRO
3513
3514 016716 011646 $TRAP2 MOV (SP), -(SP) // MOVE THE PC DOWN
3515 016720 016666 000004 000002 MOV 4(SP), 2(SP) // MOVE THE PSW DOWN
3516 016726 000002 RTI // RESTORE THE PSW
3517
3518 SBTTL TRAP TABLE
3519
3520 // *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3521 // *BY THE "TRAP" INSTRUCTION
3522
3523 ; ROUTINE
3524 / -----
3525 016730 016716 $TRPAD: .WORD $TRAP2
3526 016732 013112 $TYPE // CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
3527 016734 014630 $TYPOC // CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3528 016736 014604 $TYPOS // CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)

```

```

3529 016740 014644          $TYPON ;,CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
3530
3531
3532 016742 013070          . SCOP1 ;;CALL=SCOP1 TRAP+5(104405)
3533 016744 013374          INSTR ;;CALL=INSTR TRAP+6(104406)
3534 016746 013502          . INSTER ;;CALL=INSTER TRAP+7(104407)
3535 016750 013512          . PARAM ;;CALL=PARAM TRAP+10(104410)
3536 016752 013712          SAVOS ;;CALL=SAVOS TRAP+11(104411)
3537 016754 013752          . RESOS ;;CALL=RESOS TRAP+12(104412)
3538 016756 014004          . CONVRT ;;CALL=CONVRT TRAP+13(104413)
3539 016760 014162          . SETFLG ;;CALL=SETFLG TRAP+14(104414)
3540 ;*****
3541 ;UTILITIES
3542 ;*****
3543
3544 ,THIS UTILITY CALCULATES PRIORITY LEVEL
3545 016762 006367 000044 DULEV ASL DUPRT ,SHIFT LEFT
3546 016766 006367 000040 ASL DUPRT ;
3547 016772 006367 000034 ASL DUPRT ;
3548 016776 006367 000030 ASL DUPRT ;
3549 017002 006367 000024 ASL DUPRT ;
3550 017006 016767 000020 000020 MOV DUPRT,LESS1 ,MOVE THIS TO LESS1
3551 017014 162767 000001 000012 SUB #1,LESS1 ,CREATE LESS1
3552 017022 042767 000037 000004 BIC #37,LESS1 ,CLEAR TNZVC
3553 017030 000207 RTS PC
3554 017032 000240 DUPRT PR5
3555 017034 000200 LESS1 PR4 ,LEVEL TO ALLOW INTERRUPTS
3556
3557 ;NEW DU ADDRESSES
3558 017036 016767 000126 162646 DUADDR MOV DUBASE,RXCSR ,XXX0
3559 017044 005267 000120 INC DUBASE
3560 017050 016767 000114 162636 MOV DUBASE,HRXCSR ,XXX1
3561 017056 005267 000106 INC DUBASE
3562 017062 016767 000102 162626 MOV DUBASE,RXDBUF ,XXX2
3563 017070 016767 000074 162624 MOV DUBASE,PARCSR ,XXX2
3564 017076 005267 000066 INC DUBASE
3565 017102 016767 000062 162610 MOV DUBASE,HRXDBUF ,XXX3
3566 017110 016767 000054 162606 MOV DUBASE,HPARCSR ,XXX3
3567 017116 005267 000046 INC DUBASE
3568 017122 016767 000042 162576 MOV DUBASE,TXCSR ;XXX4
3569 017130 005267 000034 INC DUBASE
3570 017134 016767 000030 162566 MOV DUBASE,HTXCSR ,XXX5
3571 017142 005267 000022 INC DUBASE
3572 017146 016767 000016 162556 MOV DUBASE,TXDBUF ,XXX6
3573 017154 005267 000010 INC DUBASE
3574 017160 016767 000004 162546 MOV DUBASE,HTXDBUF ,XXX7
3575 017166 000207 RTS PC
3576 017170 000000 DUBASE 0
3577
3578 ,THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3579 ; INFORMATION CONTAINED IN $TMP1 AND IT IS
3580 ; SHIFTED IN BY THE CONTENTS OF SHIFT
3581 017172 042777 040000 162526 RPOKE BIC #MTDATA,@TXCSR
3582 017200 005067 162276 CLR $TMP2
3583 017204 006067 162270 ROR $TMP1 ,FORCE CARRY
3584 017210 006067 162266 ROR $TMP2 ,PICK UP CARRY IN BIT 15
  
```

```

3585 017214 006267 162262          ASR      $TMP2 , SHIFT INTO BIT 14
3586 017220 042767 100000 162254   BIC      #BIT15, $TMP2 , CLR BIT 15
3587 017226 056777 162250 162472   BIS      $TMP2, @TXCSR , POKE MAINT DATA
3588 017234 042777 020000 162464   BIC      #CLK, @TXCSR ; POKE CLK
3589 017242 052777 020000 162456   BIS      #CLK, @TXCSR ;
3590 017250 005367 161646          DEC      SHIFT
3591 017254 001346          BNE      RPOKE
3592 017256 000207          RTS      PC
3593
3594
3595 017260 016767 162214 162214 0DD8 , THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3596 017266 005067 162212          MOV      $TMP1, $TMP2 ; SAVE TEMP1
3597 017272 012727 000010          CLR      $TMP3
3598 017276 000000          MOV      #8 , (PC)+
3599 017300 006067 162176          4$      0
3600 017304 005567 162174          1$      ROR      $TMP2
3601 017310 005367 177762          ADC      $TMP3
3602 017314 001371          DEC      4$
3603 017316 006067 162162          BNE      1$
3604 017322 103404          ROR      $TMP3
3605 017324 052767 000400 162146   BCS      2$
3606 017332 000403          BIS      #BIT8, $TMP1 , SET ODD PARITY
3607 017334 042767 000400 162136 2$ BR      3$
3608          BIC      #BIT8, $TMP1 , CLR EVEN FARITY
3609 017342 000207          , $TMP1 NOW HAS ODD PARITY CHARACTER
3610          RTS      PC
3611          3$
3612          , THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3613 017344 016767 162130 162130 EVEN8 MOV      $TMP1, $TMP2 , SAVE TEMP1
3614 017352 005067 162126          CLR      $TMP3
3615 017356 012727 000010          MOV      #8 , (PC)+
3616 017362 000000          4$      0
3617 017364 006067 162112          1$      ROR      $TMP2
3618 017370 005567 162110          ADC      $TMP3
3619 017374 005367 177762          DEC      4$
3620 017400 001371          BNE      1$
3621 017402 006067 162076          ROR      $TMP3
3622 017406 103004          BCC      2$
3623 017410 052767 000400 162062   BIS      #BIT8, $TMP1 , SET EVEN PARITY
3624 017416 000403          BR      3$
3625 017420 042767 000400 162052 2$ BIC      #BIT8, $TMP1 , CLR ODD PARITY
3626          , $TMP1 NOW HAS EVEN PARITY CHARACTER
3627 017426 000207          3$      RTS      PC
3628 017430 062716 000002          TAPREG ADD      #2, (SP) , ALLOW IT TO "CRUNCH" INTO HLT BACK
3629          , IN MAIN PART OF THE PROGRAM
3630          RTI
3630          END
  
```

AAA	003200	1294*								
ABASE =	000000	877	918							
ACDW1 =	000000	877	920							
ACDW2 =	000000	877	921							
ACPUOP=	000000	877	892							
ACTREG	001166	736*	1250*	1264*	1265*	1272*	2741	2744	2754	
ADDW0 =	000000	877	922							
ADDW1 =	000000	877	923							
ADDW10=	000000	877	932							
ADDW11=	000000	877	933							
ADDW12=	000000	877	934							
ADDW13=	000000	877	935							
ADDW14=	000000	877	936							
ADDW15=	000000	877	937							
ADDW2 =	000000	877	924							
ADDW3 =	000000	877	925							
ADDW4 =	000000	877	926							
ADDW5 =	000000	877	927							
ADDW6 =	000000	877	928							
ADDW7 =	000000	877	929							
ADDW8 =	000000	877	930							
ADDW9 =	000000	877	931							
ADEVCT=	000000	877	883							
ADEVN =	000000	877	919							
ADRCNT=	013711	2956*	2995*	3002*						
AENV =	000000	877	888							
AENVN =	000000	877	889							
AFATAL =	000000	877	880							
AMADR1=	000000	877	905							
AMADR2=	000000	877	909							
AMADR3=	000000	877	912							
AMADR4=	000000	877	915							
AMAMS1=	000000	877	899							
AMAMS2=	000000	877	907							
AMAMS3=	000000	877	910							
AMAMS4=	000000	877	913							
AMSGAD=	000000	877	885							
AMSGLG=	000000	877	886							
AMSGTY=	000000	877	879							
AMTYP1=	000000	877	900							
AMTYP2=	000000	877	908							
AMTYP3=	000000	877	911							
AMTYP4=	000000	877	914							
APASS =	000000	877	882							
APRIOR=	000000	877								
APTCSU=	000040	537*	2871							
APTENV=	000001	537*	2864	3118						
APTSIZ=	000200	537*	1169							
APTSP0=	000100	537*	2866							
ASWREG=	000000	877	890							
ATESTN*	000000	877	881							
AUNIT =	000000	877	884							
AUSWR =	000000	877	891							
AVECT1=	000000	877	916							
AVECT2=	000000	877	917							
BASEAD	001154	731*	1232*	1269*	1270	1276*	1278*	2748*	2760*	2764

SCMTR	526#	816
SEOP	526#	
SERRO	526#	3088
SERRT	526#	3144
SPOWE	526#	
SSCOP	526#	3414
STRAP	526#	3495
STYPE	526#	2841
STYPO	526#	3191

ABS 017436 000

ERRORS DETECTED. 0

DZDUUB, DZDUUB/SOL/CRF=DZDUU1/EQ RUNE, DZDUU2, DZDUUB
RUN-TIME. 21 12 1 SECONDS
RUN-TIME RATIO 289/35=8 2
CORE USED 30K (59 PAGES)