

# DUV-11

OFFLINE TRANSMITTER TESTS  
MD-11-DZDUT-B

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

DEC 1977  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames, each representing a frame of test data. The frames are arranged in approximately 10 rows and 10 columns. Each frame contains a small table or set of data points, likely representing test results for different transmitter parameters. The data is too small to read clearly but appears to be organized in a structured format.



REM \*

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

PRODUCT CODE MAINDEC-11-DZDUT-B-D

PRODUCT NAME DUV11 OFFLINE TRANSMITTER 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 TRANSMITTER TESTS VERIFY THAT THE TRANSMITTER  
SECTION PROVIDES THE CORRECT ERROR FLAGS, AND THAT IT  
TRANSMITS CHARACTERS THRU THE BIT WINDOW AT THE CORRECT  
NUMBER OF BITS PER CHARACTER

\* REM \*

2 REQUIREMENTS

PDP-11/03 COMPUTER (LS )

DUV11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

\* REM \*

2 2 STORAGE  
THE PROGRAM LOADS INTO 4K OF MEMOPY 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 SUSWR REFLECTS THIS STATUS, A 0 = CONNECTOR

PRESENT, A 1 = CONNECTOR NOT AVAILBLE  
THE USER CHANGES THE CONTENTS OF THIS LOCATION  
WHEN BUILDING THE E TABLE, BY ANSWERING THE  
PROMPT "SWITCH 2"

4 1 1 AFTER PROGRAM LOAD (INITIAL PROGRAM START)  
ALL CONSOLE SWITCHES DOWN

4 1 2 TO MODIFY DEVICE VECTOR AND CONTROL REGISTER ADDRESSES  
AFTER PROGRAM RESTART OR TO RUN MULTIPLE DEVICES

SW00=1

4 1 3 TO START PROGRAM AT SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)

SW01=1

4 1 4 TO LOCK ON SELECTED TEST AFTER A PROGRAM RESTART  
(ONLY IN SINGLE DEVICE TESTS)

SW14=1

NOTE1 IN GENERAL SW01 WILL BE USED WHEN SW14=1 IS USED  
NOTE2 WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1  
STARTING ADDRESS

4 2

THE STARTING ADDRESS FOR ALL TESTS IS 000200

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

4 3 PROGRAM AND/OR OPERATOR ACTION

4 3 1 INITIAL PROGRAM START

4 3 1 1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER

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

4 3 1 3 TYPE 200G

4 3 1 4 PROGRAM WILL START

\*

REM \*

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

\*

REM \*

4 3 1 6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT  
TO START TESTING ,AND THEN TESTING WILL BEGIN

4 3 2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4 3 2 1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

4 3 3 PROGRAM RESTART WITH SW00=1

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

4 3 3 2 TYPE 200G

4 3 3 3 PROGRAM WILL START

4 3 3 4 THE PROGRAM WILL TYPE " 1ST DEVICE RECEIVER CONTROL REGISTER ADDRESS" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 5 TYPE IN THE ADDRESS OF THE FIRST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4 3 3 4

4 3 3 6 THE PROGRAM WILL TYPE "VECTOR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 7 TYPE IN THE BASE RECEIVER INTERRUPT VECTOR ADDRESS FOR THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4 3 3 6

4 3 3 8 THE PROGRAM WILL TYPE "ARE YOU RUNNING MULTIPLE DEVICES ?" (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 9 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS GIVEN, THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4 3 3 8

IF A "NO" ANSWER IS GIVEN JUMP TO SECTION 4 3 3 12  
IF A "YES" ANSWER IS GIVEN THE NEXT QUESTION IS ASKED

4 3 3 10 THE PROGRAM WILL TYPE "LAST DEVICE RECEIVER CONTROL REGISTER ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 11 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4 3 3 10  
NOTE ALL ADDRESSES SHALL BE CONTIGUOUS

4 3 3 11 1 IF AN "OUT OF RANGE" ADDRESS IS TYPED  
IE MORE THAN 16 (10) DEVICES AWAY (UPWARDS) THE

PROGRAM WILL TYPE "OUT OF RANGE RETYPE LAST DEVICE RXCSR ADDRESS-"  
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 11 2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL  
REGISTER ADDRESS OF THE DUV11 TO BE TESTED FOLLOWED  
BY A <CARRIAGE RETURN>

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4 3 3 11 1

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

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

4 3 3 13 TYPE IN THE APPROPRIATE ANSWER "1" OR "2" FOLLOWED  
BY A <CARRIAGE RETURN> (NOTE ALL MULTIPLE DEVICES MUST  
BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4 3 3 12

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

4 3 3 15 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN> (NOTE THAT ALL MULTIPLE DEVICES  
MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4 3 3 14

4 3 3 16 THE PROGRAM WILL TYPE "IS SEC REC SWITCH E55-3 ON?  
(Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 17 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN> (NOTE ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4 3 3 16

4 3 3 18 THE PROGRAM WILL TYPE "IS OPT CLR ENABLE SWITCH  
E55-1 ON? (Y OR N)-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 3 19 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED  
BY A <CARRIAGE RETURN>. (NOTE ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4.3 3 18

4 3 3 20 THE PROGRAM WILL TYPE "ARE YOU RUNNING IN MAINT  
MODE EXTERNAL ? AND DO YOU HAVE THE EXTERNAL MODEM  
BYPASS JUMPER CONNECTOR ON ? (Y OR N)-" AND WAIT FOR AN  
INPUT FROM THE TELETYPE KEYBOARD

4 3 3 21 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY  
A <CARRIAGE RETURN> (NOTE ALL MULTIPLE DEVICES MUST BE THE SAME)

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?"  
AND WILL REPEAT THE MESSAGE OF 4 3 3 20

4 3 3 22 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT  
HAS STARTED AND WILL COMMENCE TESTING AT TEST 1

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

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

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

4 3 4 2 TYPE 200G

4 3 4 3 PROGRAM WILL START

4 3 4 4 THE PROGRAM WILL TYPE "TEST PC-" AND WAIT FOR AN INPUT FROM  
THE TELETYPE KEYBOARD

4 3 4 5 TYPE IN THE ADDRESS OF THE TEST AT WHICH THE PROGRAM IS TO  
BE STARTED FOLLOWED BY A <CARRIAGE RETURN>

4 3 4 6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED  
TESTING AT THE SELECTED TEST

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

4 3 5 PROGRAM RESTART WITH SW14 =1  
NOTE THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED  
SEE NOTE IN 4 3 4 FOR MORE DETAILS

4 3 5 1 SET SW14=1 IN SWITCH REG (LOC 176)

4 3 5 2 TYPE 200G

4 3 5 3 PROGRAM WILL START

4 3 5 4 THE PROGRAM WILL TYPE "LOCK ON SELECTED TEST ? (Y OR N)-"  
AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4 3 5 5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A  
<CARRIAGE RETURN>

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

4 3 5 6 IF A YES ANSWER WAS GIVEN THE PROGRAM WILL ACT AS FOLLOWS  
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED  
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED  
OR IF ANY KEY IS STRUCK ON THE TELETYPE, THE PROGRAM  
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON  
THE TELETYPE AND SO FORTH THRU THE PROGRAM IF SW01 =1 IT  
WILL PERFORM AS IN SECTION 4 3 4 ALLOWING ONE TO FREEZE  
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5 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  
16XXXX YYYYYY ZZZZZZ

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

WHERE YYYYYY IS THE EXPECTED CONTENTS OF THAT REGISTER

WHERE ZZZZZZ IS THE ACTUAL CONTENTS OF THAT REGISTER

6 1 5 ERROR DESCRIPTIONS  
SEE LISTINGS FOR DETAILS OF ERRORS

6 2 ERROR RECOVERY

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

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

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

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

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

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

END OF PASS TAPE Y  
16XXXX = DEVICE

WHERE Y IS THE TAPE LOADED

WHERE 16XXXX IS THE DEVICE'S BASE REGISTER ADDRESS

TO INHIBIT THIS TYPEOUT - TURN TELETYPE OFF

7 RESTRICTIONS

7 1 MULTIPLE DEVICES  
UP TO 16(10) DEVICES MAY BE TESTED HOWEVER, THEY  
MUST HAVE CONTIGUOUS ADDRESSES AND VECTORS

NOTE IF ALL DEVICES UNDER TEST HAVE THE SAME INTERRUPT VECTOR  
YOU CAN CHANGE "ZERO ADD #10, 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 TYPEOUT 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 TRANSMITTER SECTION TESTING  
OF THE DEVICE  
SEE LISTING FOR DETAILS

10 FLOW CHARTS: RECEIVER FLOW, TRANSM TTER FLOW, TRANSMITTER & PECEIVER 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 DZDUT-B TAPE D
562          , COPYRIGHT 1977, DIGITAL EQUIPMENT CORP , MAYNARD, MASS 01754 ;
563
564          , STARTING PROCEDURE
565          , TYPE 200G
566          , PROGRAM WILL TYPE "DUV11 DZDUT-B TAPE D "
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 D"
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 DNISP= 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

,XDATA 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

,XBASIC "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) **ERPOR**
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 DEV CES
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 001176 EVEN
748 001400 =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      RECACT=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      SYNCH=BIT4      , SYNC SEARCH
776      000010      STD=BIT3      , SEC XMIT DATA
777      000004      RTS=BIT2      , REQ TO SEND
778      000002      DTR=BIT1      , DATA TERM PDY
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      EUPAR=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

```

DZDUT-B MACY11 30(1046) 21-SEP-77 09 18 PAGE 20  
DZDUTB M11 31-MAY-77 09 49 BASIC DEFINITIONS

SEQ 0018

807	000040	DNAINTE=BIT5	, DNR 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

Line	Address	Value	Label	Type	Value	Description
816			SBTTL COMMON TAGS			
817						
818			*****			
819			*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS			
820			*USED IN THE PROGRAM			
821						
822		001400	=			
823	001400		\$CMTAG			.. START OF COMMON TAGS
824	001400	000000		WORD	0	
825	001402	000	\$TSTNM	BYTE	0	.. CONTAINS THE TEST NUMBER
826	001403	000	\$ERFLG	BYTE	0	.. CONTAINS ERROR FLAG
827	001404	000000	\$ICNT	WORD	0	.. CONTAINS SUBTEST ITERATION COUNT
828	001406	000000	\$LPADR	WORD	0	.. CONTAINS SCOPE LOOP ADDRESS
829	001410	000000	\$LPERR	WORD	0	.. CONTAINS SCOPE RETURN FOR ERRORS
830	001412	000000	\$ERTTL	WORD	0	.. CONTAINS TOTAL ERRORS DETECTED
831	001414	000	\$ITEMB	BYTE	0	.. CONTAINS ITEM CONTROL BYTE
832	001415	001	\$ERMAX	BYTE	1	.. CONTAINS MAX ERRORS PER TEST
833	001416	000000	\$ERRPC	WORD	0	.. CONTAINS PC OF LAST ERROR INSTRUCTION
834	001420	000000	\$GDADR	WORD	0	.. CONTAINS ADDRESS OF 'GOOD' DATA
835	001422	000000	\$BDADR	WORD	0	.. CONTAINS ADDRESS OF 'BAD' DATA
836	001424	000000	\$GDDAT	WORD	0	.. CONTAINS 'GOOD' DATA
837	001426	000000	\$BDDAT	WORD	0	.. CONTAINS 'BAD' DATA
838	001430	000000		WORD	0	.. RESERVED--NOT TO BE USED
839	001432	000000		WORD	0	
840	001434	000	\$AUTOB	BYTE	0	.. AUTOMATIC MODE INDICATOR
841	001435	000	\$NTAG	BYTE	0	.. INTERRUPT MODE INDICATOR
842	001436	000000		WORD	0	
843	001440	177570	\$SWR	WORD	DSWR	.. ADDRESS OF SWITCH REGISTER
844	001442	177570	\$DISPLAY	WORD	DDISP	.. ADDRESS OF DISPLAY REGISTER
845	001444	177560	\$TKS			.. TTY KBD STATUS
846	001446	177562	\$TKB			.. TTY KBD BUFFER
847	001450	177564	\$TPS			.. TTY PRINTER STATUS REG ADDRESS
848	001452	177566	\$TPB			.. TTY PRINTER BUFFER REG ADDRESS
849	001454	000	\$NULL	BYTE	0	.. CONTAINS NULL CHARACTER FOR FILLS
850	001455	002	\$FILLS	BYTE	2	.. CONTAINS # OF FILLER CHARACTERS REQUIRED
851	001456	012	\$FILLC	BYTE	12	.. INSERT FILL CHARS AFTER A "LINE FEED"
852	001457	000	\$TPFLG	BYTE	0	.. "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
853	001460	000000	\$REGAD	WORD	0	.. CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED
854						
855	001462	000000	\$REG0	WORD	0	.. CONTAINS ((\$REGAD)+0)
856	001464	000000	\$REG1	WORD	0	.. CONTAINS ((\$REGAD)+2)
857	001466	000000	\$REG2	WORD	0	.. CONTAINS ((\$REGAD)+4)
858	001470	000000	\$REG3	WORD	0	.. CONTAINS ((\$REGAD)+6)
859	001472	000000	\$REG4	WORD	0	.. CONTAINS ((\$REGAD)+10)
860	001474	000000	\$REG5	WORD	0	.. CONTAINS ((\$REGAD)+12)
861	001476	000000	\$TMP0	WORD	0	.. USER DEFINED
862	001500	000000	\$TMP1	WORD	0	.. USER DEFINED
863	001502	000000	\$TMP2	WORD	0	.. USER DEFINED
864	001504	000000	\$TMP3	WORD	0	.. USER DEFINED
865	001506	000000	\$TMP4	WORD	0	.. USER DEFINED
866	001510	000000	\$TMP5	WORD	0	.. USER DEFINED
867	001512	000000	\$TIMES	0		.. MAX NUMBER OF ITERATIONS
868	001514	000000	\$ESCAPE	0		.. ESCAPE ON ERROR ADDRESS
869	001516	177607 000377	\$BELL	ASCII	<207><377><377>	.. CODE FOR BELL
870	001522	077	\$QUES	ASCII	/?	.. QUESTION MARK
871	001523	015	\$CRLF	ASCII	<15>	.. CARRIAGE RETURN



```
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
```

928	001626	000000	\$DDW6	WORD	ADDW6	:: DEVICE DESCRIPTOR WORD#6
929	001630	000000	\$DDW7	WORD	ADDW7	:: DEVICE DESCRIPTOR WORD#7
930	001632	000000	\$DDW8	WORD	ADDW8	:: DEVICE DESCRIPTOR WORD#8
931	001634	000000	\$DDW9	WORD	ADDW9	:: DEVICE DESCRIPTOR WORD#9
932	001636	000000	\$DDW10	WORD	ADDW10	:: DEVICE DESCRIPTOR WORD#10
933	001640	000000	\$DDW11	WORD	ADDW11	:: DEVICE DESCRIPTOR WORD#11
934	001642	000000	\$DDW12	WORD	ADDW12	:: DEVICE DESCRIPTOR WORD#12
935	001644	000000	\$DDW13	WORD	ADDW13	:: DEVICE DESCRIPTOR WORD#13
936	001646	000000	\$DDW14	WORD	ADDW14	:: DEVICE DESCRIPTOR WORD#14
937	001650	000000	\$DDW15	WORD	ADDW15	:: DEVICE DESCRIPTOR WORD#15
938						
939						
940	001652		SETEND			
941						
942						
943						

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)  
 POPRO=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  
 RECACT=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  
 SYN SCH=BIT4 ; SYNC SEARCH  
 STD=BIT3 ; SEC XMIT DATA  
 RTS=BIT2 ; REQ TO SEND  
 DTR=BIT1 ; DATA TERM RDY  
 VOID=BIT0

, RXDBUF BIT DEFINITIONS

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

, PARCSR BIT DEFINITIONS

PAREN=BIT9 ; PARITY ENABLE  
 EVPAR=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=BIT5	,DNA INTR ENAB
1001	000020	SEND=BIT4	,SEND
1002	000010	HDXEN=BIT3	,HDX/FDX
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

:

1009 SBTTL ERROR POINTER TABLE  
 1010  
 1011 ,\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR  
 1012 ,\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
 1013 ,\*LOCATION \$ITEMB THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT  
 1014 ,\*NOTE1 IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC)  
 1015 ,\*NOTE2 EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS  
 1016  
 1017 ,\* EM ;,POINTS TO THE ERROR MESSAGE  
 1018 ,\* DH ;,POINTS TO THE DATA HEADER  
 1019 ,\* DT ;,POINTS TO THE DATA  
 1020 ,\* DF ;,POINTS TO THE DATA FORMAT  
 1021  
 1022

1023 001652 SERRTB  
 1024 , ERROR TABLE  
 1025 001652 001762 EM1 , ERROR 1 REGISTER ERROR  
 1026 001654 002067 DH1  
 1027 001656 002116 DT1  
 1028 001660 002132 DF1  
 1029 001662 002022 EM2 , ERROR 2 RECEIVER ERROR  
 1030 001664 002067 DH1  
 1031 001666 002116 DT1  
 1032 001670 002132 DF1  
 1033 001672 002043 EM3 , ERROR 3 TRANSMITTER ERROR  
 1034 001674 002067 DH1  
 1035 001676 002116 DT1  
 1036 001700 002132 DF1  
 1037 001702 001746 EM4 , ERROR 4 BIT ERROR (GENERAL)  
 1038 001704 000000 0  
 1039 001706 002126 DT4  
 1040 001710 002132 DF1  
 1041  
 1042

1043 001712 160010 RXCSR 160010 , DEFAULT DU ADDRESSES  
 1044 001714 160011 HRXCSR 160011  
 1045 001716 160012 RXOBUF 160012  
 1046 001720 160013 HRXOBUF 160013  
 1047 001722 160012 PARCSR 160012  
 1048 001724 160013 HPARCSR 160013  
 1049 001726 160014 TXCSR 160014  
 1050 001730 160015 HTXCSR 160015  
 1051 001732 160016 TXOBUF 160016  
 1052 001734 160017 HTXOBUF 160017

1053 , DEFAULT DU VECTORS  
 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

1058 , ERROR MESSAGES  
 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	000123						
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						
1080							. DATA TABLES FOR ERROR MESSAGES	
1081	002116	001416	001130	001132	DT1	WORD	\$ERRPC, HLDD, HLD1, 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								
1089								
1090								
1091								
1092								
1093		002136						
1094		000046						
1095	000046	012670						
1096		000052						
1097	000052	000000						
1098		002136						
1099								
1100								
1101								
1102								
1103								
1104		002136						
1105		000024						
1106	000024	000200						
1107		000044						
1108	000044	002136						
1109		002136						
1110								
1111								
1112								
1113								
1114	002136							
1115	002136	000000						
1116	002140	001526						
1117	002142	000010						
1118	002144	000010						
1119	002146	000000						
1120	002150	000052						

. DATA HEADERS FOR ERROR MESSAGES  
 . ASCIZ /ERRPC WANTED ACTUAL/

. DATA TABLES FOR ERROR MESSAGES  
 WORD \$ERRPC, HLDD, HLD1, 0

WORD \$ERRPC, 0

BYTE 0, 0, 0, 0

EVEN  
 SBTTL ACT11 HOOKS

\*\*\*\*\*  
 . HOOKS REQUIRED BY ACT11  
 \$SVP= . SAVE PC  
 =46  
 SENDAD . 1)SET LOC 46 TO ADDRESS OF SENDAD IN SEOP  
 =52  
 WORD 0 . 2)SET LOC 52 TO ZERO  
 =\$SVP . RESTORE PC  
 SBTTL APT PARAMETER BLOCK

\*\*\*\*\*  
 . SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
 \*\*\*\*\*  
 \$X= . SAVE CURRENT LOCATION  
 =24 . SET POWER FAIL TO POINT TO START OF PROGRAM  
 200 . FOR APT START UP  
 =44 . POINT TO APT INDIRECT ADDRESS PNTR  
 \$APTHDR . POINT TO APT HEADER BLOCK  
 = \$X . RESET LOCATION COUNTER

\*\*\*\*\*  
 . SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
 . INTERFACE SPEC

\$APTHD  
 \$HIBTS WORD 0 . TWO HIGH BITS OF 18 BIT MAILBOX ADDR  
 \$MBADR WORD \$MAIL . ADDRESS OF APT MAILBOX (BITS 0-15)  
 \$TSTM WORD 10 . RUN TIM OF LONGEST TEST  
 \$PASTM WORD 10 . RUN TIME IN SECS OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
 \$UNITM WORD . ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
 SETEND-\$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 016314 000020    MOV    # $SCOPE, @ $IOTVEC ;; IOT VECTOR FOR SCOPE ROUTINE
1140 002200    012737 000340 000022    MOV    # 340, @ $IOTVEC+2 ;; LEVEL 7
1141 002206    012737 014204 000030    MOV    # $ERROR, @ $EMTVEC ;; EMT VECTOR FOR ERROR ROUTINE
1142 002214    012737 000340 000032    MOV    # 340, @ $EMTVEC+2 ;; LEVEL 7
1143 002222    012737 016650 000034    MOV    # $TRAP, @ $TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
1144 002230    012737 000340 000036    MOV    # 340, @ $TRAPVEC+2; LEVEL 7
1145 002236    012737 015006 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    MOV    # 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    # $SWR, SWR      ;; SETUP FOR A HARDWARE SWICH REGISTER
1157 002324    012767 177570 177110    MOV    # $DISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
1158 002332    022777 177777 177100    CMP    # -1, @ SWR      ;; TRY TO REFERENCE HARDWARE SWR
1159 002340    001012          BNE    66$             ;; 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    BIT    # $APTSIZE, $ENUM ;; TEST USER SIZE UNDER APT
1170 002404    001403          BEQ    67$            ;; YES, USE NON-APT SWITCH
1171 002406    012767 001550 177024    MOV    # $SWREG, SWR   ;; NO, USE APT SWITCH REGISTER
1172 002414          67$
1173 002414    012706 001100          MOV    # $STACK, SP    , SET STACK
1174 002420    106427 000340          MTPS   # 340          ; LOCK INTERRUPTS
1175 002424    012737 015006 000024    MOV    # $PFAIL, @ $24 ; SET UP POWER FAIL VECTOR
1176 002432    105067 176535          CLR    $STFLG         ; CLEAR START FLAG
  
```



INITIALIZE THE COMMON TAGS

1177	002436	005067	176450			CLR	PASCNT	, CLEAR PASS COUNT
1178	002442	105067	176735			CLRB	SERFLG	, CLEAR ERROR FLAG
1179	002446	005067	176740			CLR	SERTTL	, CLEAR ERROR COUNT
1180	002452	005067	176740			CLR	SERRPC	, CLEAR LAST EPROR POINTER
1181	002456	012767	000001	176716		MOV	#1, \$TSTNM	, SET UP FOR TEST 1
1182	002464	012767	002152	176412		MOV	# START, RETURN	, SET UP FOR POWER FAIL BEFORE TESTING STARTS
1183								
1184	002472	013746	000006			MOV	@#6, -(SP)	
1185	002476	013746	000004			MOV	@#4, -(SP)	
1186	002502	012737	002515	000004		MOV	#15, @#4	
1187	002510	005777	176724			TST	@SWR	
1188	002514	000407				BR	25	
1189	002516	012767	000176	176714	15	MOV	#SWREG, SWR	
1190	002524	012767	000174	176710		MOV	#DISPREG, DISPLAY	
1191	002532	022626				CMP	(SP)+, (SP)+	
1192	002534	012637	000004		25	MOV	(SP)+, @#4	
1193	002540	012637	000006			MOV	(SP)+, @#6	
1194	002544	022767	000176	176666		CMP	#SWREG, SWR	
1195	002552	001007				BNE	35	
1196	002554	005737	000042			TST	@#42	, CHECK FOR CHA N
1197	002560	001402				BEQ	335	
1198	002562	000167	000522			JMP	BEGIN	
1199	002566	004767	010200		335	JSR	PC, CNTLU	
1200	002572	105767	176374		35	TSTB	INIFLG	, HAS INITIALIZATION BEEN PERFORMED
1201	002576	001004				BNE	ONCE	
1202	002600	104401	015146			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	115	, BR IF NO
1206	002616	032767	000001	176726		BIT	#1, \$USWR	, EXTENAL JUMPER ON?
1207	002624	001002				BNE	125	, NO
1208	002626	105067	176321			CLRB	JMRBY	, CLEAR FLAG
1209	002632	000167	000452		125	JMP	BEGIN	, GO DO IT
1210	002636	032777	000001	176574	115	BIT	#SW00, @SWR	, RESELECT VECTOR & CONTROL REG?
1211	002644	001002				BNE	15	
1212	002646	000167	000436			JMP	BEGIN	
1213	002652	012700	000300		15	MOV	#300, R0	, RESTORE VECTOF AREA TO TRAPCATCHER
1214	002656	012701	000302			MOV	#302, R1	START AT LOCATION 300
1215	002662	012702	000004			MOV	#4, R2	
1216	002666	010110			25	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	25	
1222	002704	104406				INSTR		, OUTPUT MESSAGE & GET INPUT STRING
1223	002706	015214				MREGAD		, MESSAGE
1224	002710	104410				PARAM		, CONVERT STRING
1225	002712	160000				160000		, LOW LIMIT
1226	002714	167776				167776		, HIGH LIMIT
1227	002716	017144				DUBASE		, STORE AT THIS LOCATION
1228	002720	001			BYTE	1		, MASK
1229	002721	001			BYTE	1		, HOW MANY TIMES + 2
1230	002722	016767	014216	176226		MOV	DUBASE, KEEPADD	, SAVE
1231	002730	004767	014056			JSR	PC, DUADDR	
1232	002734	016767	176216	176212		MOV	KEEPADD, BASEADD	, RESTORE FOR ROTATION

1233	002742	104406				INSTR	, OUTPUT MESSAGE & GET INPUT STRING
1234	002744	015201				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			BYTE	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 STR NG
1244	002776	015244				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 , ON THE SYSTEM ?
1248							
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	00302f	104406				INSTR	, OUTPUT MESSAGE & GET INPUT STRING
1255	003030	015273				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	1\$	MOV	#1, ROTADD , SET UP POINTER
1264	003052	005067	176110			CLR	ACTREG , CLR ACTIVE REGISTER
1265	003056	056767	176106	176102	2\$	BIS	ROTADD, ACTREG , MAKE THIS DEVICE ACTIVE
1266	003064	000241				CLC	
1267	003066	006167	176076			ROL	ROTADD , SET UP POINTER
1268	003072	103421				BCS	3\$ , 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	2\$ , NO DO IT AGAIN
1272	003112	056767	176052	176046		BIS	ROTADD, ACTREG ; THIS ASSUMES THAT THERE ARE AT , LEAST TWO DEVICES WHEN YOU ANSWER YES TO , MULTIPLE DEVICE QUESTION
1273							
1274							
1275	003120	012767	000001	176042	4\$	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	3\$	MOV	KEEPADD, BASEADD , RESTORE
1279	003144	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1280	003146	015367				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	1\$ , DO IT AGAIN
1288	003166	012767	000340	013612		OUTMUL	MOV #340, DUPRT

```

1289 003174 004767 013536 JSP 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 015605 MSYNC ,MESSAGE
1297 003204 122767 000061 012734 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 012716 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$ INSTER ,RETRY
1306 003244 000757 BR 3$
1307 003246 000240 4$ NOP
1308 003250 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1309 003252 015653 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 015724 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 015774 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 016053 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 STPING
1332 003332 015537 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 015533 4$ TYPE ,MR ,TYPE R
1343 003370 000177 175510 JMP @RETURN ,START TESTING
1344

```

```
1345          , , THIS TEST CHECKS THE STRIP SYNC FUNCTION
1346          , , OF THE RECEIVER LOGIC
1347          , , MODE SYNINT
1348          , , LENGTH EIGHT
1349          , , NOTE: RXDONE SHOULD NEVER ASSERT
1350          , , CHAR 26 (SYNC)
1351          , ,
1352          , , *****
1353 003374 000004 TST1 SCOPE
1354 003376 052777 000400 176322 BIS #MRESET,@TXCSR ; MASTER RESET
1355 003404 012777 030000 176310 MOV #SYNINT,@PARCSR ; SET THE MODE
1356 003412 052777 000400 176306 BIS #MRESET,@TXCSR ; MASTER RESET
1357
1358          , SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1359 003420 012777 064001 176300 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1360
1361          , SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
1362 003426 012777 036026 176266 MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
1363 003434 052777 000020 176250 BIS #SYNSCH,@RXCSR ; SET SYNC SEARCH
1364          , POKE CLK TO GET RECEIVER INTO SYNCRIZATION
1365 003442 042777 020000 176256 BIC #CLK,@TXCSR ; POKE CLK DOWN
1366 003450 052777 020000 176250 BIS #CLK,@TXCSR ; POKE CLK UP
1367          , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1368 003456 042777 020000 176242 BIC #CLK,@TXCSR ; POKE CLK DOWN
1369 003464 052777 020000 176234 BIS #CLK,@TXCSR ; POKE CLK UP
1370 003472 052777 000400 176212 BIS #STPSYN,@RXCSR ; SET STRIP SYNC
1371 003500 012767 000003 175416 MOV #3,COUNT ; # OF SYNC CHARS
1372 003506 012767 000026 175764 15 MOV #26,$TMP1 ; CHAR TO BE SHIFTED
1373 003514 012767 000010 175400 MOV #8,SHIFT ; # OF SHIFTS
1374 003522 004767 013420 JSR PC,RPOKE ; SHIFT IN THIS CHAR
1375 003526 105777 176160 TSTB @RXCSR ,RXDONE ?
1376 003532 100001 BPL +4
1377 003534 104004 ERROR 4 ; RXDONE SHOULD NOT BE ASSERTED
1378 003536 005367 175362 DEC COUNT ; # OF SYNC CHARS
1379 003542 001361 BNE 15
1380
1381          , , THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1382          , , WHILE IN STRIP SYNC MODE
1383          , , THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1384          , , STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1385          , , BUT IF AN ERROR SHOULD OCCUR THIS AUTOMATIC RESET
1386          , , IS DISCOMBOB LATED
1387          , , IE FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1388          , , NOTE NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
1389          , , PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT
1390          , , BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
1391          , , MODE ISOC (ISY:100)
1392          , , LENGTH EIGHT
1393          , , PARITY EVEPAR
1394          , , CHARACTER EXPECTED 26
1395          , , CHARACTER SENT SYNC CHARACTER
1396          , , NOTE THIS TEST USES ONLY THE RECEIVER LOGIC
1397          , ,
1398          , , *****
1399 003544 000004 TST2 SCOPE
1400 003546 052777 000400 176152 BIS #MRESET,@TXCSR ; MASTER RESET
```

```

1401 003554 012777 000000 176140      MOV      #ISYMOD,@PARCSR ,SET THE MODE
1402 003562 052777 000400 176136      BIS      #MRESET,@TXCSR ,MASTER RESET
1403
1404                                     ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1405 003570 012777 064001 176130      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1406
1407                                     ,SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1408 003576 012777 007426 176116      MOV      #ISYMOD'EIGHT!EVEPAR'26,@PARCSR
1409 003604 016703 176106      MOV      @RXDBUF,R3 ;SET UP FOR ERROR MSG
1410 003610 012767 000003 175306      MOV      #3,COUNT , # OF TIMES SYNC CHAR WILL BE SENT
1411 003616 052777 000020 176066      BIS      #SYNSCH,@RXCSR ,SET SYNC SEARCH
1412                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1413 003624 042777 020000 176074      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1414 003632 052777 020000 176066      BIS      #CLK,@TXCSR ;POKE CLK UP
1415                                     ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1416 003640 042777 020000 176060      BIC      #CLK,@TXCSR ,POKE CLK DOWN
1417 003646 052777 020000 176052      BIS      #CLK,@TXCSR ,POKE CLK UP
1418 003654 052777 000400 176030      BIS      #STPSYN,@RXCSR ,SET STRIP SYNC
1419 003662 012767 000013 175232 25     MOV      #11 ,SHIFT , # OF SHIFTS
1420 003670 012767 003054 175602      MOV      #3054,$TMP1 ,SYNC CHAR + START&STOP+ PARITY
1421 003676 004767 013244 19     JSR      PC,RPOKE ,SHIFT IN THIS CHARACTER
1422 003702 105777 176004      TSTB    @RXCSR ,RXDONE = 0 ?
1423 003706 100001      BPL     +4
1424 003710 104004      ERROR   4 ,RXDONE SHOULD NOT BE SET
1425 003712 005367 175206      DEC     COUNT , # OF SYNC CHARS
1426 003716 001361      BNE     25 ,GO AGAIN ?
1427 003720 012700 000026      MOV     #26,R0 ,EXPECTED
1428 003724 017701 175766      MOV     @RXDBUF,R1 ,ACTUAL
1429                                     ,NOTE THAT THIS IS THE FIRST TIME
1430                                     ,RXDBUF IS READ THERE SHOULD BE
1431                                     ,NO OVER RUN ERROR +5
1432 003730 020001      CMP     R0,R1 ,COMPARE EXPECTED VS ACTUAL
1433 003732 001401      BEQ     +4
1434 003734 104002      ERROR   2 ,DATA CHARS SHOULD COMPARE
1435                                     ,THERE SHOULD BE NO RXERR'S
1436 003736 012767 000004 175160      MOV     #4,COUNT , # OF TIMES
1437 003744 012700 110026      MOV     #RXERR'PARER'26,R0 ,EXPECTED
1438 003750 012767 002054 175522      MOV     #2054,$TMP1 ,BAD SYNC CHAR (WONG PAPITY)
1439 003756 012767 000013 175136 35     MOV     #11 ,SHIFT , # OF SHIFTS
1440 003764 004767 013156      JSR     PC,RPOKE ,SHIFT IN THIS CHAR
1441 003770 105777 175716      TSTB    @RXCSR ,RXDONE = 1 ?
1442 003774 100401      BMI     +4
1443 003776 104004      ERROR   4 ,RXDONE SHOULD BE SET
1444 004000 017701 175712      MOV     @RXDBUF,R1 ,ACTUAL DATA
1445 004004 020001      CMP     R0,R1 ,COMPARE EXP VS ACT
1446 004006 001401      BEQ     +4
1447 004010 104000      ERROR   ,DID THE RESPECTIVE ERROR 4 STOP THE
1448                                     ,AUTOMATIC RESSETING OF RXDONE & ERROP FLAGS
1449                                     ,CHECK THIS
1450 004012 005367 175106      DEC     COUNT , # OF SYNC CHARS
1451 004016 001445      BEQ     55 ,FINISHED ? GET OUT OF TEST
1452 004020 022767 000003 175076      CMP     #3,COUNT , # OF SYNC CHARS
1453 004026 001423      BEQ     65 ,CHECK FRAME ERROR ?
1454 004030 022767 000002 175066      CMP     #2,COUNT , # OF SYNC CHARS
1455 004036 001426      BEQ     75 ,CHECK FRAME ERROR & BAD PARITY ?
1456                                     NOPE THEN IT (COUNT) MUST BE = 1 THEREFOPE

```

```

1457 004040 012767 000013 175054      MOV      #11,SHIFT      ,# OF SHIFTS
1458 004046 012767 000054 175424      MOV      #54,$TMP1      ,FRAME & PARITY ERROR
1459 004054 004767 013066      JSR      PC,RPOKE      ,SHIFT IN THIS CHAR
1460      ,NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1461 004060 012767 000054 175412      MOV      #54,$TMP1      ,FRAME & PARITY ERROR
1462 004066 012700 170026      MOV      #RXERR!OVRRUN!FMERR!PARER!26,RO      ,EXPECTED
1463 004072 000167 177660      JMP      3$      ,DO IT AGAIN
1464 004076 012767 001054 175374 6$      MOV      #1054,$TMP1      ,BAD STOP BIT FOR FRAME ERROR
1465 004104 012700 120026      MOV      #RXERR!FMERR!26,RO      ,EXPECTED
1466 004110 000167 177642      JMP      3$      ,DO IT AGAIN
1467 004114 012767 000054 175356 7$      MOV      #54,$TMP1      ,BAD STOP BIT & PARITY
1468 004122 012700 130026      MOV      #RXERR!FMERR!PARER!26,RO      ,EXPECTED
1469 004126 000167 177624      JMP      3$      ,DO IT AGAIN
1470 004132
1471      5$
1472      ,,TH'S TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1473      ,,WHILE IN STRIP SYNC MODE
1474      ,,THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1475      ,,STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1476      ,, BUT IF AN ERROR SHOULD OCCUR THIS AUTOMATIC RESET
1477      ,, IS DISCOMBOBULATED
1478      ,, IE FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1479      ,,NOTE NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
1480      ,,PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT
1481      ,,BUT, IF AN FXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
1482      ,,MODE ISOC ISYMOD)
1483      ,,LENGTH SEVEN
1484      ,,PARITY EVEPAR
1485      ,,CHAPACTER EXPECTED 226
1486      ,,NOTE THAT THE PARITY BIT SHOULD SHOW
1487      ,,UP IN THE DATA IE BIT SEVEN FOR
1488      ,,SEVEN LEVEL CODE
1489      ,,CHARACTER SENT SYNC CHAPACTER
1490      ,,NOTE THIS TEST USES ONLY THE RECEIVER LOGIC
1491      ,,
1492      ,,*****
1492 004132 000004      TST3     SCOPE
1493 004134 052777 000400 175564      BIS      #MRESET,@TXCSR ,MASTER RESET
1494 004142 012777 000000 175552      MOV      #ISYMOD,@PARCSR ,SET THE MODE
1495 004150 052777 000400 175550      BIS      #MRESET,@TXCSR ,MASTER RESET
1496
1497      ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1498 004156 012777 064001 175542      MOV      #MTDATA'CLK'MINT'BREAK,@TXCSP
1499
1500      ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1501 004164 012777 005626 175530      MOV      #ISYMOD'SEVEN'EVEPAR'226,@PARCSR
1502 004172 016703 175520      MOV      RXDBUF,R3      ,SET UP FOR ERROR MSG
1503 004176 012767 000003 174720      MOV      #3,COUNT      ,# OF TIMES SYNC CHAR WILL BE SENT
1504 004204 052777 000020 175500      BIS      #SYNSCH,@RXCSR ,SET SYNC SEARCH
1505      ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1506 004212 042777 020000 175506      BIC      #CLK,@TXCSR      ,POKE CLK DOWN
1507 004220 052777 020000 175500      BIS      #CLK,@TXCSR      ,POKE CLK UP
1508      ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1509 004226 042777 020000 175472      BIC      #CLK,@TXCSR      ,POKE CLK DOWN
1510 004234 052777 020000 175464      BIS      #CLK,@TXCSR      ,POKE CLK UP
1511 004242 052777 000400 175442      BIS      #STPSYN,@RXCSR ,SET STRIP SYNC
1512 004250 012767 000012 174644 2$      MOV      #10,SHIFT      ,# OF SHIFTS
    
```





INITIALIZE THE COMMON TAGS

```

1569      , , THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1570      , , STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1571      , , BUT IF AN ERROR SHOULD OCCUR THIS AUTOMATIC RESET
1572      , , IS DISCOMBOBULATED
1573      , , IE FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1574      , , NOTE NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
1575      , , PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT
1576      , , BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
1577      , , MODE ISOC (ISYMOD)
1578      , , LENGTH SIX
1579      , , PARITY EVEPAR
1580      , , CHARACTER EXPECTED 126
1581      , , NOTE THAT THE PARITY BIT SHOULD SHOW
1582      , , UP IN THE DATA IE BIT SIX FOR
1583      , , SIX LEVEL CODE
1584      , , CHARACTER SENT SYNC CHARACTER
1585      , , NOTE THIS TEST USES ONLY THE RECEIVER LOGIC
1586      , ,
1587      , , *****
1588      004520 000004 TST4 SCOPE
1589      004522 052777 000400 175176 BIS #MRESET,@TXCSR ,MASTER RESET
1590      004530 012777 000000 175164 MOV #ISYMOD,@PARCSR ,SET THE MODE
1591      004536 052777 000400 175162 BIS #MRESET,@TXCSR ,MASTER RESET
1592
1593      , SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1594      004544 012777 064001 175154 MOV #MTDATA'CLK'MINT'BREAK,@TXCSR
1595
1596      , SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
1597      004552 012777 003526 175142 MOV #ISYMOD'SIX'EVEPAR'126,@PARCSR
1598      004560 016703 175132 MOV RXDBUF,R3 ,SET UP FOR ERROR MSG
1599      004564 012767 000003 174332 MOV #3,COUNT ,# OF TIMES SYNC CHAP WILL BE SENT
1600      004572 052777 000020 175112 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1601      , POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1602      004600 042777 020000 175120 BIC #CLK,@TXCSR ,POKE CLK DOWN
1603      004606 052777 020000 175112 BIS #CLK,@TXCSR ,POKE CLK UP
1604      , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1605      004614 042777 020000 175104 BIC #CLK,@TXCSR ,POKE CLK DOWN
1606      004622 052777 020000 175076 BIS #CLK,@TXCSR ,POKE CLK UP
1607      004630 052777 000400 175054 BIS #STPSYN,@RXCSR ,SET STRIP SYNC
1608      004636 012767 000011 174256 25 MOV #9,SHIFT ,# OF SHIFTS
1609      004644 012767 000654 174626 MOV #654,$TMP1 ,SYNC CHAR + START&STOP+ PAPITY
1610      004652 004767 012270 15 JSR PC,RPOKE ,SHIFT IN THIS CHARACTER
1611      004656 105777 175030 TSTB @RXCSR ,RXDONE = 0 ?
1612      004662 100001 BPL +4
1613      004664 104004 ERROR 4 ,RXDONE SHOULD NOT BE SET
1614      004666 005367 174232 DEC COUNT ,# OF SYNC CHARS
1615      004672 001361 BNE 25 ,GO AGAIN ?
1616      004674 012700 000126 MOV #126,RD ,EXPECTED
1617      004700 017701 175012 MOV @RXDBUF,R1 ,ACTUAL
1618      , NOTE THAT THIS IS THE FIRST TIME
1619      , RXDBUF IS READ THEPE SHOULD BE
1620      , NO OVER RUN ERROR 45
1621      004704 020001 CMP RD,R1 ,COMPARE EXPECTED VS ACTUAL
1622      004706 001401 BEQ 4
1623      004710 104002 ERROR 2 ,DATA CHAPS SHOULD COMPARE
1624      , THERE SHOULD BE NO RXERR'S
  
```

1625	004712	012767	000004	174204		MOV	#4,COUNT	, # OF TIMES
1626	004720	012700	110026			MOV	#RXERR'PARER'26,RO	, EXPECTED
1627	004724	012767	000454	174546		MOV	#454,\$TMP1	, BAD SYNC CHAR (WRONG PARITY)
1628	004732	012767	000011	174162	35	MOV	#9,SHIFT	, # OF SHIFTS
1629	004740	004767	012202			JSR	PC,RPOKE	, SHIFT IN THIS CHAR
1630	004744	105777	174742			TSTB	DRXCSR ,RXDONE = 1?	
1631	004750	100401				BMI	+4	
1632	004752	104004				ERROR	4	, RXDONE SHOULD BE SET
1633	004754	017701	174736			MOV	DRXDBUF,R1	, ACTUAL DATA
1634	004760	020001				CMP	RO,R1	, COMPARE EXP VS ACT
1635	004762	001401				BEQ	+4	
1636	004764	104000				ERROR		, DID THE RESPECTIVE ERROR 4 STOP THE
1637								, AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
1638								, CHECK THIS
1639								, NOTE THAT THE PARITY BIT SHOULD
1640								, SHOW UP IN THE DATA
1641								, IE BIT SIX FOR SIX LEVEL CODE
1642	004766	005367	174132			DEC	COUNT	, # OF SYNC CHARS
1643	004772	001445				BEQ	55	, FINISHED ? GET OUT OF TEST
1644	004774	022767	090003	174122		CMP	#3,COUNT	, # OF SYNC CHARS
1645	005002	001423				BEQ	65	, CHECK FRAME ERROR ?
1646	005004	022767	000002	174112		CMP	#2,COUNT	, # OF SYNC CHARS
1647	005012	001426				BEQ	75	, CHECK FRAME ERROR & BAD PARITY ?
1648								, NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE
1649	005014	012767	000011	174100		MOV	#9,SHIFT	, # OF SHIFTS
1650	005022	012767	000054	174450		MOV	#54,\$TMP1	, FRAME & PARITY ERROR
1651	005030	004767	012112			JSR	PC,RPOKE	, SHIFT IN THIS CHAR
1652								, NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1653	005034	012767	000054	174436		MOV	#54,\$TMP1	, FRAME & PARITY ERROR
1654	005042	012700	170026			MOV	#RXERR'OURRUN'FRMERR'PARER'26,RO	, EXPECTED
1655	005046	000167	177660			JMP	35	, DO IT AGAIN
1656	005052	012767	000254	174420	65	MOV	#254,\$TMP1	, BAD STOP BIT FOR FRAME ERROR
1657	005060	012700	120126			MOV	#RXERR'FRMERR'126,RO	, EXPECTED
1658	005064	000167	177642			JMP	35	, DO IT AGAIN
1659	005070	012767	000054	174402	75	MOV	#54,\$TMP1	, BAD STOP BIT & PARITY
1660	005076	012700	130026			MOV	#RXERR'FRMERR'PARER'26,RO	, EXPECTED
1661	005102	000167	177624			JMP	35	, DO IT AGAIN
1662	005106							55
1663								.. THIS TEST PROVES THAT RXERR FREEZES THE "RECEIVER RESET"
1664								.. WHILE IN STRIP SYNC MODE
1665								.. THIS TEST FIRST PROVES THAT AUTOMATIC RESETS OCCUR WHEN
1666								.. STRIP SYNC IS SET & SYNC CHARACTERS ARE SENT
1667								.. BUT IF AN ERROR SHOULD OCCUR THIS AUTOMATIC RESET
1668								.. IS DISCOMBOBULATED
1669								.. IE FORCE PARITY ERROR WHILE STRIP SYNC IS SET
1670								.. NOTE NORMALLY THE LOGIC RESETS THE RXDONE & ERROR FLAGS
1671								.. PROVIDING THAT ONLY GOOD SYNC CHARACTERS ARE SENT
1672								.. BUT, IF AN RXERR OCCURS RXDONE PLUS RXERR ARE ASSERTED
1673								.. MODE ISOC (ISYMOD)
1674								.. LENGTH FIVE
1675								.. PARITY EVEPAR
1676								.. CHARACTER EXPECTED 66
1677								.. NOTE THAT THE PARITY BIT SHOULD SHOW
1678								.. UP IN THE DATA 'E' BIT FIVE FOR
1679								.. FIVE LEVEL CODE
1680								.. CHARACTER SENT SYNC CHARACTER

INITIALIZE THE COMMON TAGS

```

1681      ,NOTE THIS TEST USES ONLY THE RECEIVER LOGIC
1682      ;;
1683      ,*****
1684 005106 000004      TST5  SCOPE
1685 005110 052777 000400 174610      BIS      #MRESET,@TXCSR ;MASTER RESET
1686 005116 012777 000000 174576      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1687 005124 052777 000400 174574      BIS      #MRESET,@TXCSR ;MASTER RESET
1688
1689      ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1690 005132 012777 0640 174566      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1691
1692      ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1693 005140 012777 001466 174554      MOV      #ISYMOD!FIVE!EVEPAR!66,@PARCSR
1694 005146 016703 174544      MOV      RXDBUF,R3 ;SET UP FOR ERROR MSG
1695 005152 012767 000003 173744      MOV      #3,COUNT ;# OF TIMES SYNC CHAR WILL BE SENT
1696 005160 052777 000020 174524      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1697      ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1698 005166 042777 020000 174532      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1699 005174 052777 020000 174524      BIS      #CLK,@TXCSR ;POKE CLK UP
1700      ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1701 005202 042777 020000 174516      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1702 005210 052777 020000 174510      BIS      #CLK,@TXCSR ;POKE CLK UP
1703 005216 052777 000400 174466      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
1704 005224 012767 000010 173670 25      MOV      #8,SHIFT ;# OF SHIFTS
1705 005232 012767 000354 174240 15      MOV      #354,$TMP1 ;SYNC CHAR + START&STOP+ PARITY
1706 005240 004767 011702      JSR      PC,RPOKE ;SHIFT IN THIS CHARACTER
1707 005244 105777 174442      TSTB    @RXCSR ,RXDONE = 0 ?
1708 005250 100001      BPL     +4
1709 005252 104004      ERROR   4 ;RXDONE SHOULD NOT BE SET
1710 005254 005367 173644      DEC     COUNT ;# OF SYNC CHARS
1711 005260 001361      BNE     25 ;GO AGAIN ?
1712 005262 012700 000066      MOV     #66,R0 ;EXPECTED
1713 005266 017701 174424      MOV     @RXDBUF,R1 ;ACTUAL
1714      ,NOTE THAT THIS IS THE FIRST TIME
1715      ,RXDBUF IS READ THERE SHOULD BE
1716      ,NO OVER RUN ERROR 45
1717 005272 020001      CMP     R0,R1 ;COMPARE EXPECTED VS ACTUAL
1718 005274 001401      BEQ     +4
1719 005276 104002      ERROR   2 ;DATA CHARS SHOULD COMPARE
1720      ,THERE SHOULD BE NO RXERR'S
1721 005300 012767 000004 173616      MOV     #4,COUNT ;# OF TIMES
1722 005306 012700 110026      MOV     #RXERR!PARER!26,R0 ;EXPECTED
1723 005312 012767 000254 174160      MOV     #254,$TMP1 ;BAD SYNC CHAR (WRONG PARITY)
1724 005320 012767 000010 173574 35      MOV     #8,SHIFT ;# OF SHIFTS
1725 005326 004767 011614      JSR     PC,RPOKE ;SHIFT IN THIS CHAR
1726 005332 105777 174354      TSTB    @RXCSR ,RXDONE = 1 ?
1727 005336 100401      BMI     +4
1728 005340 104004      ERROR   4 ;RXDONE SHOULD BE SET
1729 005342 017701 174350      MOV     @RXDBUF,R1 ;ACTUAL DATA
1730 005346 020001      CMP     R0,R1 ;COMPARE EXP VS ACT
1731 005350 001401      BEQ     +4
1732 005352 104000      ERROR   ,DID THE RESPECTIVE ERROR 4 STOP THE
1733      ,AUTOMATIC RESSETTING OF RXDONE & ERROR FLAGS
1734      ,CHECK THIS
1735      ,NOTE THAT THE PARITY BIT SHOULD
1736      ,SHOW UP IN THE DATA
  
```

```

1737                                     , IE BIT FIVE FOR FIVE LEVEL CODE
1738 005354 005367 173544             DEC COUNT ;# OF SYNC CHARS
1739 005360 001445                   BEQ 5$ ;FINISHED ? GET OUT OF TEST
1740 005362 022767 000003 173534     CMP #3,COUNT ;# OF SYNC CHARS
1741 005370 001423                   BEQ 6$ ;CHECK FRAME ERROR ?
1742 005372 022767 000002 173524     CMP #2,COUNT ;# OF SYNC CHARS
1743 005400 001426                   BEQ 7$ ;CHECK FRAME ERROR & BAD PARITY ?
1744                                     ,NOPE THEN IT (COUNT) MUST BE = 1 THEREFORE
1745 005402 012767 000010 173512     MOV #8,SHIFT ;# OF SHIFTS
1746 005410 012767 000054 174062     MOV #54,$TMP1 ;FRAME & PARITY ERROR
1747 005416 004767 011524           JSR PC,RPOKE ;SHIFT IN THIS CHAR
1748                                     ,NOW DON'T READ THE RXDBUF TO CREATE OVER RUN
1749 005422 012767 000054 174050     MOV #54,$TMP1 ;FRAME & PARITY ERROR
1750 005430 012700 170026           MOV #RXERR'OUVRUN'FMERR'PARER'26,RO ;EXPECTED
1751 005434 000167 177660           JMP 3$ ;DO IT AGAIN
1752 005440 012767 000154 174032 6$ MOV #154,$TMP1 ;BAD STOP BIT FOR FRAME ERROR
1753 005446 012700 120066           MOV #RXERR'FMERR'66,RO ;EXPECTED
1754 005452 000167 177642           JMP 3$ ;DO IT AGAIN
1755 005456 012767 000054 174014 7$ MOV #54,$TMP1 ;BAD STOP BIT & PARITY
1756 005464 012700 130026           MOV #RXERR'FMERR'PARER'26,RO ;EXPECTED
1757 005470 000167 177624           JMP 3$ ;DO IT AGAIN
1758 005474                           5$
1759                                     ;, THIS TEST VERIFYS WORD LENGTH SELECT OF
1760                                     ;, THE TRANSMITTER SECTION, IT USES THE DNA FLAG
1761                                     ;, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1762                                     ;, CORRECTLY
1763                                     ;, NOTE. DNA COMES UP ON THE FIRST RISING BIT
1764                                     ;, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1765                                     ;, LOADED INTO TXDBUF
1766                                     ;, MODE: SYNINT
1767                                     ;, PARITY: NO PARITY
1768                                     ;, LENGTH: FIVE
1769                                     ;,
1770                                     ;, *****
1771 005474 000004                       TST6 SCOPE
1772 005476 052777 000400 174222     BIS #MRESET,@TXCSR ;MASTER RESET
1773 005504 012777 030000 174210     MOV #SYNINT,@PARCSR ;SET THE MODE
1774 005512 052777 000400 174206     BIS #MRESET,@TXCSR ;MASTER RESET
1775
1776                                     ,SET MAINTENANCE MODE & SEND
1777                                     ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1778 005520 012777 004020 174200     MOV #MINT'SEND,@TXCSR
1779
1780                                     ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1781 005526 012777 030026 174166     MOV #SYNINT'FIVE'NOPAR'26,@PARCSR
1782 005534 016703 174166           MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1783 005540 112777 000021 174164     MOVB #21,@TXDBUF ;LOAD CHAR
1784 005546 012767 000021 173724     MOV #21,$TMP1 ;SHIFTED CHAR
1785 005554 012767 000005 173340     MOV #5,SHIFT ;# OF SHIFTS
1786                                     ,POKE CLK TO GET INTO SYNCHRONIZATION
1787 005562 052777 020000 174136     BIS #CLK,@TXCSR ;POKE CLK UP
1788 005570 042777 020000 174130     BIC #CLK,@TXCSR ;POKE CLK DOWN
1789 005576 005000                       1$ CLR RO
1790 005600 006067 173674           ROR $TMP1 ;FORCE CARRY
1791 005604 103002                       BCC 2$
1792 005606 052700 002000           BIS #BITW,RO ;EQUIV OF BIT WINDOW

```

```
1793 005612 25
1794 005612 052777 020000 174106 BIS #CLK,@TXCSR ;POKE CLK UP
1795 005620 042777 020000 174100 BIC #CLK,@TXCSR ;POKE CLK DOWN
1796 005626 017701 174074 MOV @TXCSR,R1 ;ACTUAL
1797 005632 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
1798 005636 020001 CMP RO,R1 ;COMPARE EXP VS ACT
1799 005640 001401 BEQ +4
1800 005642 104003 ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1801 ;BIT, ... ALSO CHECK DNA
1802 005644 005367 173252 DEC SHIFT ;# OF SHIFTS
1803 005650 001352 BNE 15 ;DO IT AGAIN ?
1804 ;NOW POKE CLK TO SEE DNA
1805 005652 052777 020000 174046 BIS #CLK,@TXCSR ;POKE CLK
1806 005660 012700 100000 MOV #100000,RO ;EXPECTED
1807 005664 017701 174036 MOV @TXCSR,R1 ;ACTUAL
1808 005670 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
1809 005674 020001 CMP RO,R1 ;COMPARE EXPECTED VS ACTUAL
1810 005676 001401 BEQ +4
1811 005700 104003 ERROR 3 ;DNA SHOULD BE SET
1812 ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1813 ;SELECT LOGIC OF THE TRANSMITTER
1814 005702 005777 174020 TST @TXCSR ;DNA ?
1815 005706 100001 BPL +4
1816 005710 104004 ERROR 4 ;DNA SHOULD NOT BE SET
1817 ;IT SHOULD HAVE BEEN CLEARED FROM
1818 ;PREVIOUS READ
1819
1820 ;, THIS TEST VERIFYS WORD LENGTH SELECT OF
1821 ;, THE TRANSMITTER SECTION, IT USES THE DNA FLAG
1822 ;, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1823 ;, CORRECTLY
1824 ;, NOTE DNA COMES UP ON THE FIRST RISING BIT
1825 ;, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1826 ;, LOADED INTO TXDBUF
1827 ;, MODE SYNINT
1828 ;, PARITY NO PARITY
1829 ;, LENGTH SIX
1830 ;,
1831 ;, *****
1832 005712 000004 TST7 SCOPE
1833 005714 052777 000400 174004 BIS #MRESET,@TXCSR ;MASTER RESET
1834 005722 012777 030000 173772 MOV #SYNINT,@PARCSR ;SET THE MODE
1835 005730 052777 000400 173770 BIS #MRESET,@TXCSR ;MASTER RESET
1836
1837 ; SET MAINTENANCE MODE & SEND
1838 ; NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1839 005736 012777 004020 173762 MOV #MINT!SEND,@TXCSR
1840
1841 ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1842 005744 012777 032026 173750 MOV #SYNINT!SIX!NOPAR!26,@PARCSR
1843 005752 016703 173750 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
1844 005756 112777 000021 173746 MOVB #21,@TXDBUF ;LOAD CHAR
1845 005764 012767 000021 173506 MOV #21,$TMP1 ;SHIFTED CHAR
1846 005772 012767 000006 173122 MOV #6,SHIFT ;# OF SHIFTS
1847 ;POKE CLK TO GET INTO SYNCHRONIZATION
1848 006000 052777 020000 173720 BIS #CLK,@TXCSR ;POKE CLK UP
```

```

1849 006006 042777 020000 173712      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1850 006014 005000                    CLR    RO             ;
1851 006016 006067 173456                ROR    $TMP1        ,FORCE CARRY
1852 006022 103002                    BCC    2$           ;
1853 006024 052700 002000                BIS    #BITW,RO      ;EQUIV OF BIT WINDOW
1854 006030                    2$
1855 006030 052777 020000 173670      BIS    #CLK,@TXCSR    ;POKE CLK UP
1856 006036 042777 020000 173662      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1857 006044 017701 173656                MOV    @TXCSR,R1     ;ACTUAL
1858 006050 042701 075777                BIC    #075777,R1    ;SAVE BITW & DNA
1859 006054 020001                    CMP    RO,R1         ;COMPARE EXP VS ACT
1860 006056 001401                    BEQ    +4            ;
1861 006060 104003                    ERROR  3             ;BIT WINDOW DID NOT MATCH ACTUAL DATA
1862                                ;BIT, ... ALSO CHECK DNA
1863 006062 005367 173034                DEC    SHIFT        ;# OF SHIFTS
1864 006066 001352                    BNE    1$           ;DO IT AGAIN ?
1865                                ;NOW POKE CLK TO SEE DNA
1866 006070 052777 020000 173630      BIS    #CLK,@TXCSR    ;POKE CLK
1867 006076 012700 100000                MOV    #100000,RO    ;EXPECTED
1868 006102 017701 173620                MOV    @TXCSR,R1     ;ACTUAL
1869 006106 042701 077777                BIC    #77777,R1     ;SAVE DNA ONLY
1870 006112 020001                    CMP    RO,R1         ;COMPARE EXPECTED VS ACTUAL
1871 006114 001401                    BEQ    +4            ;
1872 006116 104003                    ERROR  3             ;DNA SHOULD BE SET
1873                                ;IF DNA DID NOT SET ,CHECK WORD LENGTH
1874                                ;SELECT LOGIC OF THE TRANSMITTER
1875 006120 005777 173602                TST    @TXCSR        ;DNA ?
1876 006124 100001                    BPL    +4            ;
1877 006126 104004                    ERROR  4             ;DNA SHOULD NOT BE SET
1878                                ;IT SHOULD HAVE BEEN CLEARED FROM
1879                                ;PREVIOUS READ
1880
1881                                ;, THIS TEST VERIFYS WORD LENGTH SELECT OF
1882                                ;, THE TRANSMITTER SECTION, IT USES THE DNA FLAG
1883                                ;, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1884                                ;, CORRECTLY
1885                                ;, NOTE DNA COMES UP ON THE FIRST RISING BIT
1886                                ;, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1887                                ;, LOADED INTO TXDBUF
1888                                ;, MODE. SYNINT
1889                                ;, PARITY NO PARITY
1890                                ;, LENGTH SEVEN
1891                                ;,
1892                                ;, *****
1893 006130 000004                    TST10  SCOPE
1894 006132 052777 000400 173566                BIS    #MRESET,@TXCSR ;MASTER RESET
1895 006140 012777 030000 173554                MOV    #SYNINT,@PARCSR ;SET THE MODE
1896 006146 052777 000400 173552                BIS    #MRESET,@TXCSR ;MASTER RESET
1897
1898                                ;SET MAINTENANCE MODE & SEND
1899                                ;NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1900 006154 012777 004020 173544                MOV    #MINT!SEND,@TXCSR
1901
1902                                ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1903 006162 012777 034026 173532                MOV    #SYNINT!SEVEN!NOPAR!26 @PARCSR
1904 006170 016703 173532                MOV    TXCSR,R3      ;SET UP FOR ERROR MSG
  
```

INITIALIZE THE COMMON TAGS

```

1905 006174 112777 000021 173530      MOVB  #21,@TXDBUF      ,LOAD CHAR
1906 006202 012767 000021 173270      MOV   #21,$TMP1       ;SHIFTED CHAR
1907 006210 012767 000007 172704      MOV   #7,$SHIFT      ;# OF SHIFTS
1908                                     ,POKE CLK TO GET INTO SYNCHRONIZATION
1909 006216 052777 020000 173502      BIS   #CLK,@TXCSR     ;POKE CLK UP
1910 006224 042777 020000 173474      BIC   #CLK,@TXCSR     ;POKE CLK DOWN
1911 006232 005000                15      CLR   R0
1912 006234 006067 173240                ROR   $TMP1           ,FORCE CARRY
1913 006240 103002                BCC   25
1914 006242 052700 002000                BIS   #BITW,R0        ;EQUIV OF BIT WINDOW
1915 006246                25
1916 006246 052777 020000 173452      BIS   #CLK,@TXCSR     ,POKE CLK UP
1917 006254 042777 020000 173444      BIC   #CLK,@TXCSR     ,POKE CLK DOWN
1918 006262 017701 173440      MOV   @TXCSR,R1       ,ACTUAL
1919 006266 042701 075777      BIC   #075777,R1     ,SAVE BITW & DNA
1920 006272 020001      CMP   R0,R1           ,COMPARE EXP VS ACT
1921 006274 001401      BEQ   +4
1922 006276 104003      ERROR 3              ,BIT WINDOW DID NOT MATCH ACTUAL DATA
1923                                     ,BIT, ALSO CHECK DNA
1924 006300 005367 172616      DEC   $SHIFT         ,# OF SHIFTS
1925 006304 001352      BNE   15             ;DO IT AGAIN ?
1926                                     ,NOW POKE CLK TO SEE DNA
1927 006306 052777 020000 173412      BIS   #CLK,@TXCSR     ,POKE CLK
1928 006314 012700 100000      MOV   #100000,R0     ,EXPECTED
1929 006320 017701 173402      MOV   @TXCSR,R1     ,ACTUAL
1930 006324 042701 077777      BIC   #77777,R1     ,SAVE DNA ONLY
1931 006330 020001      CMP   R0,R1         ,COMPARE EXPECTED VS ACTUAL
1932 006332 001401      BEQ   +4
1933 006334 104003      ERROR 3              ,DNA SHOULD BE SET
1934                                     ,IF DNA DID NOT SET ,CHECK WORD LENGTH
1935                                     ,SELECT LOGIC OF THE TRANSMITTER
1936 006336 005777 173364      TST   @TXCSR         ,DNA ?
1937 006342 100001      BPL   +4
1938 006344 104004      ERROR 4              ,DNA SHOULD NOT BE SET
1939                                     ,IT SHOULD HAVE BEEN CLEARED FROM
1940                                     ,PREVIOUS READ
1941
1942                                     ,, THIS TEST VERIFYS WORD LENGTH SELECT OF
1943                                     ,, THE TRANSMITTER SECTION. IT USES THE DNA FLAG
1944                                     ,, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
1945                                     ,, CORRECTLY
1946                                     ,, NOTE DNA COMES UP ON THE FIRST RISING BIT
1947                                     ,, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
1948                                     ,, LOADED INTO TXDBUF
1949                                     ,, MODE SYNINT
1950                                     ,, PARITY NO PARITY
1951                                     ,, LENGTH EIGHT
1952                                     ,,
1953                                     ,, *****
1954 006346 000004      TST11 SCOPE
1955 006350 052777 000400 173350      BIS   #MRESET,@TXCSR ,MASTER RESET
1956 006356 012777 030000 173336      MOV   #SYNINT,@PARCSR ,SET THE MODE
1957 006364 052777 000400 173334      BIS   #MRESET,@TXCSR ,MASTER RESET
1958
1959                                     ,SET MAINTENANCE MODE & SEND
1960                                     ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)

```



INITIALIZE THE COMMON TAGS

```

1961 006372 012777 004020 173326      MOV      #MINT!SEND,@TXCSR
1962
1963      ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
1964 006400 012777 036026 173314      MOV      #SYNINT!EIGHT!NOPAR!26,@PARCSR
1965 006406 016703 173314      MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
1966 006412 112777 000021 173312      MOVB    #21,@TXDBUF   ;LOAD CHAR
1967 006420 012767 000021 173052      MOV      #21,$TMP1    ;SHIFTED CHAR
1968 006426 012767 000010 172466      MOV      #8,SHIFT     ;# OF SHIFTS
1969      ,POKE CLK TO GET INTO SYNCHRONIZATION
1970 006434 052777 020000 173264      BIS     #CLK,@TXCSR   ;POKE CLK UP
1971 006442 042777 020000 173256      BIC     #CLK,@TXCSR   ;POKE CLK DOWN
1972 006450 005000      15      CLR     R0
1973 006452 006067 173022      ROR     $TMP1 ,FORCE CARRY
1974 006456 103002      BCC     2$
1975 006460 052700 002000      BIS     #BITW,R0     ,EQUIV OF BIT WINDOW
1976 006464      2$
1977 006464 052777 020000 173234      BIS     #CLK,@TXCSR   ;POKE CLK UP
1978 006472 042777 020000 173226      BIC     #CLK,@TXCSR   ;POKE CLK DOWN
1979 006500 017701 173222      MOV     @TXCSR,R1    ,ACTUAL
1980 006504 042701 075777      BIC     #075777,R1    ,SAVE BITW & DNA
1981 006510 020001      CMP     R0,R1      ,COMPARE EXP VS ACT
1982 006512 001401      BEQ     +4
1983 006514 104003      ERROR   3      ,BIT WINDOW DID NOT MATCH ACTUAL DATA
1984      ,BIT, ALSO CHECK DNA
1985 006516 005367 172400      DEC     SHIFT      ,# OF SHIFTS
1986 006522 001352      BNE     1$      ,DO IT AGAIN ?
1987      ,NOW POKE CLK TO SEE DNA
1988 006524 052777 020000 173174      BIS     #CLK,@TXCSR   ;POKE CLK
1989 006532 012700 100000      MOV     #100000,R0    ;EXPECTED
1990 006536 017701 173164      MOV     @TXCSR,R1    ,ACTUAL
1991 006542 042701 077777      BIC     #77777,R1    ,SAVE DNA ONLY
1992 006546 020001      CMP     R0,R1      ,COMPARE EXPECTED VS ACTUAL
1993 006550 001401      BEQ     +4
1994 006552 104003      ERROR   3      ,DNA SHOULD BE SET
1995      ,IF DNA DID NOT SET ,CHECK WORD LENGTH
1996      ,SELECT LOGIC OF THE TRANSMITTER
1997 006554 005777 173146      TST     @TXCSR      ,DNA ?
1998 006560 100001      BPL     +4
1999 006562 104004      ERROR   4      ,DNA SHOULD NOT BE SET
2000      ,IT SHOULD HAVE BEEN CLEARED FROM
2001      ,PREVIOUS READ
2002
2003      ,, THIS TEST VERIFYS WORD LENGTH SELECT OF
2004      ,, THE TRANSMITTER SECTION, IT USES THE DNA FLAG
2005      ,, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
2006      ,, CORRECTLY
2007      ,, NOTE DNA COMES UP ON THE FIRST RISING BIT
2008      ,, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
2009      ,, LOADED INTO TXDBUF
2010      ,, MODE SYNEXT
2011      ,, PARITY NO PARITY
2012      ,, LENGTH FIVE
2013      ,,
2014      ,, *****
2015 006564 000004      TST12  SCOPE
2016 006566 052777 000400 173132      BIS     #MRESET,@TXCSR  MASTER PESET

```

INITIALIZE THE COMMON TAGS

```

2017 006574 012777 020000 173120      MOV    #SYNEXT,@PARCSR ,SET THE MODE
2018 006602 052777 000400 173116      BIS    #MRESET,@TXCSR ;MASTER RESET
2019
2020      ,SET MAINTENANCE MODE & SEND
2021      ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2022 006610 012777 004020 173110      MOV    #MINTSEND,@TXCSR
2023
2024      ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2025 006616 012777 020026 173076      MOV    #SYNEXT!FIVE!NOPAR!26,@PARCSR
2026 006624 016703 173075      MOV    TXCSR,R3      ,SET UP FOR ERROR MSC
2027 006630 112777 000021 173074      MOVB  #21,@TXDBUF    ,LOAD CHAR
2028 006636 012767 000021 172634      MOV    #21,$TMP1     ;SHIFTED CHAR
2029 006644 012767 000005 172250      MOV    #5,SHIFT      ,# OF SHIFTS
2030      ,POKE CLK TO GET INTO SYNCHRONIZATION
2031 006652 052777 020000 173046      BIS    #CLK,@TXCSR   ,POKE CLK UP
2032 006660 042777 020000 173040      BIC    #CLK,@TXCSR   ,POKE CLK DOWN
2033 006666 005000      1$    CLR    R0
2034 006670 006067 172604      ROR    $TMP1        ,FORCE CARRY
2035 006674 103002      BCC    2$
2036 006676 052700 002000      BIS    #BITW,R0      ,EQUIV OF BIT WINDOW
2037 006702      2$
2038 006702 052777 020000 173016      BIS    #CLK,@TXCSR   ,POKE CLK UP
2039 006710 042777 020000 173010      BIC    #CLK,@TXCSR   ,POKE CLK DOWN
2040 006716 017701 173004      MOV    @TXCSR,R1     ,ACTUAL
2041 006722 042701 075777      BIC    #075777,R1    ,SAVE BITW & DNA
2042 006726 020001      CMP    R0,R1        ,COMPARE EXP VS ACT
2043 006730 001401      BEQ    +4
2044 006732 104003      EPROR  3            ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2045      ,BIT, ALSO CHECK DNA
2046 006734 005367 172162      DEC    SHIFT        ,# OF SHIFTS
2047 006740 001352      BNE    1$           ,DO IT AGAIN ?
2048      ,NOW POKE CLK TO SEE DNA
2049 006742 052777 020000 172756      BIS    #CLK,@TXCSR   ,POKE CLK
2050 006750 012700 100000      MOV    #100000,R0    ,EXPECTED
2051 006754 017701 172746      MOV    @TXCSR,R1     ,ACTUAL
2052 006760 042701 077777      BIC    #77777,R1     ,SAVE DNA ONLY
2053 006764 020001      CMP    R0,R1        ,COMPARE EXPECTED VS ACTUAL
2054 006766 001401      BEQ    +4
2055 006770 104003      ERROR  3            ,DNA SHOULD BE SET
2056      ,IF DNA DID NOT SET ,CHECK WORD LENGTH
2057      ,SELECT LOGIC OF THE TRANSMITTER
2058 006772 005777 172730      TST    @TXCSR        ,DNA ?
2059 006776 100001      BPL    +4
2060 007000 104004      EPROR  4            ,DNA SHOULD NOT BE SET
2061      ,IT SHOULD HAVE BEEN CLEARED FROM
2062      ,PREVIOUS READ
2063
2064      ,, THIS TEST VERIFYS WORD LENGTH SELECT OF
2065      ,, THE TRANSMITTER SECTION, IT USES THE DNA FLAG
2066      ,, AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
2067      ,, CORRECTLY
2068      ,, NOTE DNA COMES UP ON THE FIRST RISING BIT
2069      ,, EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
2070      ,, LOADED INTO TXDBUF
2071      ,, MODE SYNEXT
2072      ,, PARITY NO PARITY
  
```

```

2073          ;,LENGTH SIX
2074          ;,
2075          ;,*****
2076 007002 000004 TST13 SCOPE
2077 007004 052777 000400 172714 BIS #MRESET,@TXCSR ;MASTER RESET
2078 007012 012777 020000 172702 MOV #SYNEXT,@PARCSR ;SET THE MODE
2079 007020 052777 000400 172700 BIS #MRESET,@TXCSR ;MASTER RESET
2080
2081          ;SET MAINTENANCE MODE & SEND
2082          ;NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2083 007026 012777 004020 172672 MOV #MINT!SEND,@TXCSR
2084
2085          ;SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2086 007034 012777 022026 172660 MOV #SYNEXT!SIX!NOPAR!26,@PARCSR
2087 007042 016703 172660 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2088 007046 112777 000021 172656 MOVB #21,@TXDBUF ;LOAD CHAR
2089 007054 012767 000021 172416 MOV #21,$TMP1 ;SHIFTED CHAR
2090 007062 012767 000006 172032 MOV #6,SHIFT ;# OF SHIFTS
2091          ;POKE CLK TO GET INTO SYNCHRONIZATION
2092 007070 052777 020000 172630 BIS #CLK,@TXCSR ;POKE CLK UP
2093 007076 042777 020000 172622 BIC #CLK,@TXCSR ;POKE CLK DOWN
2094 007104 005000 15 CLR R0
2095 007106 006067 172366 ROR $TMP1 ;FORCE CARRY
2096 007112 103002 BCC 2$
2097 007114 052700 002000 BIS #BITW,R0 ;EQUIV OF BIT WINDOW
2098 007120 25
2099 007120 052777 020000 172600 BIS #CLK,@TXCSR ;POKE CLK UP
2100 007126 042777 020000 172572 BIC #CLK,@TXCSR ;POKE CLK DOWN
2101 007134 017701 172566 MOV @TXCSR,R1 ;ACTUAL
2102 007140 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2103 007144 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2104 007146 001401 BEQ +4
2105 007150 104003 ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2106          ;BIT, ALSO CHECK DNA
2107 007152 005367 171744 DEC SHIFT ;# OF SHIFTS
2108 007156 001352 BNE 1$ ;DO IT AGAIN ?
2109          ;NOW POKE CLK TO SEE DNA
2110 007160 052777 020000 172540 BIS #CLK,@TXCSR ;POKE CLK
2111 007166 012700 100000 MOV #100000,R0 ;EXPECTED
2112 007172 017701 172530 MOV @TXCSR,R1 ;ACTUAL
2113 007176 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2114 007202 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
2115 007204 001401 BEQ +4
2116 007206 104003 ERROR 3 ;DNA SHOULD BE SET
2117          ;IF DNA DID NOT SET ,CHECK WORD LENGTH
2118          ;SELECT LOGIC OF THE TRANSMITTER
2119 007210 005777 172512 TST @TXCSR ;DNA ?
2120 007214 100001 BPL +4
2121 007216 104004 ERROR 4 ;DNA SHOULD NOT BE SET
2122          ;IT SHOULD HAVE BEEN CLEARED FROM
2123          ;PREVIOUS READ
2124
2125          ;,THIS TEST VERIFYS WORD LENGTH SELECT OF
2126          ;,THE TRANSMITTER SECTION,IT USES THE DNA FLAG
2127          ;,AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED
2128          ;,CORRECTLY
  
```

INITIALIZE THE COMMON TAGS

```

2129                                     ,,NOTE DNA COMES UP ON THE FIRST RISING BIT
2130                                     ,,EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS
2131                                     ,,LOADED INTO TXDBUF
2132                                     ,,MODE SYNEXT
2133                                     ,,PARITY NO PARITY
2134                                     ,,LENGTH SEVEN
2135                                     ,,
2136                                     ,,*****
2137 007220 000004 TST14 SCOPE
2138 007222 052777 000403 172476 BIS #MRESET,@TXCSR ,MASTER RESET
2139 007230 012777 020000 172464 MOV #SYNEXT,@PARCSR ,SET THE MODE
2140 007236 052777 000400 172462 BIS #MRESET,@TXCSR ,MASTER RESET
2141
2142                                     ,SET MAINTENANCE MODE & SEND
2143                                     ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2144 007244 012777 004020 172454 MOV #MINT'SEND,@TXCSR
2145
2146                                     ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2147 007252 012777 024026 172442 MOV #SYNEXT'SEVEN'NOPAR'26,@PARCSR
2148 007260 016703 172442 MOV TXCSR,R3 ,SET UP FOR ERROR MSG
2149 007264 112777 000021 172440 MOVB #21,@TXDBUF ,LOAD CHAR
2150 007272 012767 000021 172200 MOV #21,$TMP1 ,SHIFTED CHAR
2151 007300 012767 000007 171614 MOV #7,SHIFT ,# OF SHIFTS
2152                                     ,POKE CLK TO GET INTO SYNCHRONIZATION
2153 007306 052777 020000 172412 BIS #CLK,@TXCSR ,POKE CLK UP
2154 007314 042777 020000 172404 BIC #CLK,@TXCSR ,POKE CLK DOWN
2155 007322 005000 15 CLR R0
2156 007324 006067 172150 ROR $TMP1 ,FORCE CARRY
2157 007330 103002 BCC 2$
2158 007332 052700 002000 BIS #BITW,R0 ,EQUIV OF BIT WINDOW
2159 007336 25
2160 007336 052777 020000 172362 BIS #CLK,@TXCSP ,POKE CLK UP
2161 007344 042777 020000 172354 BIC #CLK,@TXCSR ,POKE CLK DOWN
2162 007352 017701 172350 MOV @TXCSR,R1 ,ACTUAL
2163 007356 042701 075777 BIC #075777,R1 ,SAVE BITW & DNA
2164 007362 020001 CMP R0,R1 ,COMPARE EXP VS ACT
2165 007364 001401 BEQ +4
2166 007366 104003 ERROR 3 ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2167                                     ,BIT, ALSO CHECK DNA
2168 007370 005367 171526 DEC SHIFT ,# OF SHIFTS
2169 007374 001352 BNE 1$ ,DO IT AGAIN ?
2170                                     ,NOW POKE CLK TO SEE DNA
2171 007376 052777 020000 172322 BIS #CLK,@TXCSR ,POKE CLK
2172 007404 012700 100000 MOV #100000,R0 ,EXPECTED
2173 007410 017701 172312 MOV @TXCSR,R1 ,ACTUAL
2174 007414 042701 077777 BIC #77777,R1 ,SAVE DNA ONLY
2175 007420 020001 CMP R0,R1 ,COMPARE EXPECTED VS ACTUAL
2176 007422 001401 BEQ +4
2177 007424 104003 ERROR 3 ,DNA SHOULD BE SET
2178                                     ,IF DNA DID NOT SET ,CHECK WORD LENGTH
2179                                     ,SELECT LOGIC OF THE TRANSMITTER
2180 007426 005777 172274 TST @TXCSR ,DNA ?
2181 007432 100001 BPL +4
2182 007434 104004 ERROR 4 ,DNA SHOULD NOT BE SET
2183                                     ,IT SHOULD HAVE BEEN CLEARED FROM
2184                                     ,PREVIOUS READ

```

```
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198 007436 000004
2199 007440 052777 000400 172260
2200 007446 012777 020000 172246
2201 007454 052777 000400 172244
2202
2203
2204
2205 007462 012777 004020 172236
2206
2207
2208 007470 012777 026026 172224
2209 007476 016703 172224
2210 007502 112777 000021 172222
2211 007510 012767 000021 171762
2212 007516 012767 000010 171376
2213
2214 007524 052777 020000 172174
2215 007532 042777 020000 172166
2216 007540 005000
2217 007542 006067 171732
2218 007546 103002
2219 007550 052700 002000
2220 007554
2221 007554 052777 020000 172144
2222 007562 042777 020000 172136
2223 007570 017701 172132
2224 007574 042701 075777
2225 007600 020001
2226 007602 001401
2227 007604 104003
2228
2229 007606 005367 171310
2230 007612 001352
2231
2232 007614 052777 020000 172104
2233 007622 012700 100000
2234 007626 017701 172074
2235 007632 042701 077777
2236 007636 020001
2237 007640 001401
2238 007642 104003
2239
2240
```

.. THIS TEST VERIFYS WORD LENGTH SELECT OF  
.. THE TRANSMITTER SECTION, IT USES THE DNA FLAG  
.. AND BIT WINDOW TO DETERMINE THAT IT WAS SELECTED  
.. CORRECTLY  
.. NOTE: DNA COMES UP ON THE FIRST RISING BIT  
.. EDGE OF THE NEXT CHARACTER IF NO NEW CHARACTER IS  
.. LOADED INTO TXDBUF  
.. MODE: SYNEXT  
.. PARITY: NO PARITY  
.. LENGTH: EIGHT

.. \*\*\*\*\*

TST15: SCOPE  
BIS #MRESET,@TXCSR ; MASTER RESET  
MOV #SYNEXT,@PARCSR ; SET THE MODE  
BIS #MRESET,@TXCSR ; MASTER RESET

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

; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG  
MOV #SYNEXT'EIGHT'NOPAR'26,@PARCSR  
MOV TXCSR,R3 ; SET UP FOR ERROR MSG  
MOVB #21,@TXDBUF ; LOAD CHAR  
MOV #21,\$TMP1 ; SHIFTED CHAR  
MOV #8,\$SHIFT ; # OF SHIFTS

; POKE CLK TO GET INTO SYNCHRONIZATION  
BIS #CLK,@TXCSR ; POKE CLK UP  
BIC #CLK,@TXCSR ; POKE CLK DOWN  
15 CLR R0  
ROR \$TMP1 ; FORCE CARRY  
BCC 25  
BIS #BITW,R0 ; EQUIV OF BIT WINDOW  
25 BIS #CLK,@TXCSR ; POKE CLK UP  
BIC #CLK,@TXCSR ; POKE CLK DOWN  
MOV @TXCSR,R1 ; ACTUAL  
BIC #075777,R1 ; SAVE BITW & DNA  
CMP R0,R1 ; COMPARE EXP VS ACT  
BEQ +4  
ERROR 3 ; BIT WINDOW DID NOT MATCH ACTUAL DATA  
; BIT, ALSO CHECK DNA  
DEC SHIFT ; # OF SHIFTS  
BNE 15 ; DO IT AGAIN ?

; NOW POKE CLK TO SEE DNA  
BIS #CLK,@TXCSR ; POKE CLK  
MOV #100000,R0 ; EXPECTED  
MOV @TXCSR,R1 ; ACTUAL  
BIC #77777,R1 ; SAVE DNA ONLY  
CMP R0,R1 ; COMPARE EXPECTED VS ACTUAL  
BEQ +4  
ERROR 3 ; DNA SHOULD BE SET  
; IF DNA DID NOT SET CHECK WORD LENGTH  
SELECT LOGIC OF THE TRANSMITTER

INITIALIZE THE COMMON TAGS

```

2241 007644 005777 172056 TST @TXCSR ,DNA ?
2242 007650 100001 BPL +4
2243 007652 104004 ERROR 4 ,DNA SHOULD NOT BE SET
2244 ,IT SHOULD HAVE BEEN CLEARED FROM
2245 ,PREVIOUS READ
2246
2247 ,, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERAT ON
2248 ,, OF THE TRANSMITTER SECTION
2249 ,, IT ALSO CHECKS DNA TIMING
2250 ,, MODE SYNINT
2251 ,, LENGTH FIVE PLUS PARITY
2252 ,, PARITY EVEPAR
2253 ,, CHARACTER 25
2254 ,,
2255 ,, *****
2256 007654 000004 TST16 SCOPE
2257 007656 052777 000400 172042 BIS #MRESET,@TXCSR ,MASTER RESET
2258 007664 012777 030000 172030 MOV #SYNINT,@PARCSR ,SET THE MODE
2259 007672 052777 000400 172026 BIS #MRESET,@TXCSR ,MASTER RESET
2260
2261 ,SET MAINTENANCE MODE & SEND
2262 ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2263 007700 012777 004020 172020 MOV #MINTISEND,@TXCSR
2264
2265 ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2266 007706 012777 031426 172006 MOV #SYNINT'FIVE'EVEPAR'26,@PARCSR
2267 007714 016703 172006 MOV TXCSR,R3 ,SET UP FOR ERROR MSG
2268 007720 112777 000025 172004 MOVB #25,@TXDBUF ,LOAD DATA CHAR
2269 007726 012767 000065 171544 MOV #65,$TMP1 ,TO BE SHIFTED CHAR
2270 007734 012767 000006 171160 MOV #6,SHIFT ,# OF SHIFTS
2271 ,POKE CLK TO GET INTO SYNCRONIZATION
2272 007742 052777 020000 171756 BIS #CLK,@TXCSP ,POKE CLK UP
2273 007750 042777 020000 171750 BIC #CLK,@TXCSR ,POKE CLK DOWN
2274 007756 005000 15 CLR R0
2275 007760 006067 171514 ROR $TMP1 ,FORCE CARRY
2276 007764 103002 BCC 25 ,BR IF CARRY CLR
2277 007766 052700 002000 BIS #BITW,R0 ,EQUIV OF BITW
2278 007772 25
2279 007772 052777 020000 171726 BIS #CLK,@TXCSR ,POKE CLK UP
2280 010000 042777 020000 171720 BIC #CLK,@TXCSR ,POKE CLK DOWN
2281 010006 017701 171714 MOV @TXCSR,R1 ,ACTUAL
2282 010012 042701 075777 BIC #075777,R1 ,SAVE BITW & DNA
2283 010016 020001 CMP R0,R1 ,COMPAE EXP VS ACT
2284 010020 001401 BEQ +4
2285 010022 104003 ERROR 3 ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2286 ,BIT. ALSO CHECK DNA
2287 010024 005367 171072 DEC SHIFT ,# OF SHIFTS
2288 010030 001352 BNE 15 ,DO IT AGAIN ?
2289 ,NOW POKE CLK TO SEE DNA
2290 010032 052777 020000 171666 BIS #CLK,@TXCSR ,POKE CLK
2291 010040 012700 100000 MOV #100000,R0 ,EXPECTED
2292 010044 017701 171656 MOV @TXCSR,P1 ,ACTUAL
2293 010050 042701 077777 BIC #77777,R1 ,SAVE DNA ONLY
2294 010054 020001 CMP R0,R1 ,COMPARE EXP VS ACT
2295 010056 001401 BEQ +4
2296 010060 104003 ERROR 3 ,DNA SHOULD BE SET
    
```

```

2297                                     , IF DNA DID NOT SET
2298                                     , CHECK WORD LENGTH SELECT LOGIC
2299
2300                                     ,, THIS TEST VERIFYS CHARACTER PLUS PARITY GENEPAATION
2301                                     ,, OF THE TRANSMITTER SECTION
2302                                     ,, IT ALSO CHECKS DNA TIMING
2303                                     ,, MODE SYNINT
2304                                     ,, LENGTH FIVE PLUS PARITY
2305                                     ,, PARITY ODDPAR
2306                                     ,, CHARACTER: 25
2307                                     ,,
2308                                     ,, *****
2309 010062 000004 TST17 SCOPE
2310 010064 052777 000400 171634 BIS #MRESET,@TXCSR , MASTER RESET
2311 010072 012777 030000 171622 MOV #SYNINT,@PARCSR , SET THE MODE
2312 010100 052777 000400 171620 BIS #MRESET,@TXCSR ; MASTER RESET
2313
2314                                     , SET MAINTENANCE MODE & SEND
2315                                     , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2316 010106 012777 004020 171612 MOV #MINT!SEND,@TXCSR
2317
2318                                     , SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2319 010114 012777 031026 171600 MOV #SYNINT!FIVE!ODDPAR!26,@PARCSR
2320 010122 016703 171600 MOV TXCSR,R3 , SET UP FOR ERROR MSG
2321 010126 112777 000025 171576 MOVB #25,@TXDBUF , LOAD DATA CHAR
2322 010134 012767 000025 171336 MOV #25,$TMP1 , TO BE SHIFTED CHAR
2323 010142 012767 000006 170752 MOV #6,SHIFT , # OF SHIFTS
2324                                     , POKE CLK TO GET INTO SYNCRONIZATION
2325 010150 052777 020000 171550 BIS #CLK,@TXCSR , POKE CLK UP
2326 010156 042777 020000 171542 BIC #CLK,@TXCSR , POKE CLK DOWN
2327 010164 005000 15 CLR R0
2328 010166 006067 171306 POP $TMP1 , FORCE CARRY
2329 010172 103002 BCC 2$ , BR IF CARRY CLR
2330 010174 052700 002000 BIS #BITW,R0 , EQUIV OF BITW
2331 010200 2$
2332 010200 052777 020000 171520 BIS #CLK,@TXCSR , POKE CLK UP
2333 010206 042777 020000 171512 BIC #CLK,@TXCSR , POKE CLK DOWN
2334 010214 017701 171506 MOV @TXCSR,R1 , ACTUAL
2335 010220 042701 075777 BIC #075777,R1 , SAVE BITW & DNA
2336 010224 020001 CMP R0,R1 , COMPARE EXP VS ACT
2337 010226 001401 BEQ +4
2338 010230 104003 ERROR 3 , BIT WINDOW DID NOT MATCH ACTUAL DATA
2339 , BIT, ALSO CHECK DNA
2340 010232 005367 170664 DEC SHIFT , # OF SHIFTS
2341 010236 001352 BNE 1$ , DO IT AGAIN ?
2342                                     , NOW POKE CLK TO SEE DNA
2343 010240 052777 020000 171460 BIS #CLK,@TXCSR , POKE CLK
2344 010246 012700 100000 MOV #100000,R0 , EXPECTED
2345 010252 017701 171450 MOV @TXCSR,R1 , ACTUAL
2346 010256 042701 077777 BIC #77777,R1 , SAVE DNA ONLY
2347 010262 020001 CMP R0,R1 , COMPARE EXP VS ACT
2348 010264 001401 BEQ +4
2349 010266 104003 ERROR 3 , DNA SHOULD BE SET
2350                                     , IF DNA DID NOT SET
2351                                     , CHECK WORD LENGTH SELECT LOGIC
2352
    
```



```

2353 ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2354 ; OF THE TRANSMITTER SECTION
2355 ; IT ALSO CHECKS DNA TIMING
2356 ; MODE: ISYMOD
2357 ; LENGTH: FIVE PLUS PARITY
2358 ; PARITY: EVEPAR
2359 ; CHARACTER: 25
2360 ;
2361 ; *****
2362 010270 000004 TST20. SCOPE
2363 010272 052777 000400 171426 BIS #MRESET,@TXCSR ; MASTER RESET
2364 010300 012777 000000 171414 MOV #ISYMOD,@PARCSR ; SET THE MODE
2365 010306 052777 000400 171412 BIS #MRESET,@TXCSR ; MASTER RESET
2366
2367 ; SET MAINTENANCE MODE & SEND
2368 ; NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2369 010314 012777 004020 171404 MOV #MINT!SEND,@TXCSR
2370
2371 ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2372 010322 012777 001426 171372 MOV #ISYMOD!FIVE!EVEPAR!26,@PARCSR
2373 010330 016703 171372 MOV TXCSR,R3 ; SET UP FOR ERROR MSG
2374 010334 112777 000025 171370 MOVB #25,@TXDBUF ; LOAD DATA CHAR
2375 010342 012767 000352 171130 MOV #352,$TMP1 ; TO BE SHIFTED CHAR
2376 010350 012767 000010 170544 MOV #8,SHIFT ; # OF SHIFTS
2377 ; POKE CLK TO GET INTO SYNCHRONIZATION
2378 010356 052777 020000 171342 BIS #CLK,@TXCSR ; POKE CLK UP
2379 010364 042777 020000 171334 BIC #CLK,@TXCSR ; POKE CLK DOWN
2380 010372 005000 15 CLR R0
2381 010374 006067 171100 ROR $TMP1 ; FORCE CARRY
2382 010400 103002 BCC 25 ; BR IF CARRY CLR
2383 010402 052700 002000 BIS #BITW,R0 ; EQUIV OF BITW
2384 010406 25
2385 010406 052777 020000 171312 BIS #CLK,@TXCSR ; POKE CLK UP
2386 010414 042777 020000 171304 BIC #CLK,@TXCSR ; POKE CLK DOWN
2387 010422 017701 171300 MOV @TXCSR,R1 ; ACTUAL
2388 010426 042701 075777 BIC #075777,R1 ; SAVE BITW & DNA
2389 010432 020001 CMP R0,R1 ; COMPARE EXP VS ACT
2390 010434 001401 BEQ +4
2391 010436 104003 ERROR 3 ; BIT WINDOW DID NOT MATCH ACTUAL DATA
2392 ; BIT, ALSO CHECK DNA
2393 010440 005367 170456 DEC SHIFT ; # OF SHIFTS
2394 010444 001352 BNE 15 ; DO IT AGAIN ?
2395 ; NOW POKE CLK TO SEE DNA
2396 010446 052777 020000 171252 BIS #CLK,@TXCSR ; POKE CLK
2397 010454 012700 000000 MOV #0,R0 ; EXPECTED
2398 010460 017701 171242 MOV @TXCSR,R1 ; ACTUAL
2399 010464 042701 077777 BIC #77777,R1 ; SAVE DNA ONLY
2400 010470 020001 CMP R0,R1 ; COMPARE EXP VS ACT
2401 010472 001401 BEQ +4
2402 010474 104003 ERROR 3 ; DNA SHOULD BE SET
2403 ; IF DNA DID NOT SET
2404 ; CHECK WORD LENGTH SELECT LOGIC
2405
2406 ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2407 ; OF THE TRANSMITTER SECTION
2408 ; IT ALSO CHECKS DNA TIMING
  
```

```

2409          ,MODE: ISYMOD
2410          ,LENGTH: FIVE PLUS PARITY
2411          ,PARITY ODDPAR
2412          ,CHARACTER 25
2413          ,
2414          ,*****
2415 010476 000004 TST21 SCOPE
2416 010500 052777 000400 171220 BIS #MRESET,@TXCSR ;MASTER RESET
2417 010506 012777 000000 171206 MOV #ISYMOD,@PARCSR ;SET THE MODE
2418 010514 052777 000400 171204 BIS #MRESET,@TXCSR ;MASTER RESET
2419
2420          ,SET MAINTENANCE MODE & SEND
2421          ,NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2422 010522 012777 004020 171176 MOV #MINT!SEND,@TXCSR
2423
2424          ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2425 010530 012777 001026 171164 MOV #ISYMOD!FIVE!ODDPAR!2@ @PARCSR
2426 010536 016703 171164 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2427 010542 112777 000025 171162 MOVB #25,@TXDBUF ;LOAD DATA CHAR
2428 010550 012767 000252 170722 MOV #252,$TMP1 ;TO BE SHIFTED CHAR
2429 010556 012767 000010 170336 MOV #8,SHIFT ;# OF SHIFTS
2430          ,POKE CLK TO GET INTO SYNCHRONIZATION
2431 010564 052777 020000 171134 BIS #CLK,@TXCSR ;POKE CLK UP
2432 010572 042777 020000 171126 BIC #CLK,@TXCSR ;POKE CLK DOWN
2433 010600 005000 15 CLR R0
2434 010602 006067 170672 ROR $TMP1 ;FORCE CARRY
2435 010606 103002 BCC 25 ;BR IF CARRY CLR
2436 010610 052700 002000 BIS #BITW,R0 ;EQUIV OF BITW
2437 010614 25
2438 010614 052777 020000 171104 BIS #CLK,@TXCSR ;POKE CLK UP
2439 010622 042777 020000 171076 BIC #CLK,@TXCSR ;POKE CLK DOWN
2440 010630 017701 171072 MOV @TXCSR,R1 ;ACTUAL
2441 010634 042701 075777 BIC #075777,R1 ;SAVE BITW & DNA
2442 010640 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2443 010642 001401 BEQ +4
2444 010644 104003 ERROR 3 ;BIT WINDOW DID NOT MATCH ACTUAL DATA
2445          ;BIT, ALSO CHECK DNA
2446 010646 005367 170250 DEC SHIFT ;# OF SHIFTS
2447 010652 001352 BNE 15 ;DO IT AGAIN ?
2448          ,NOW POKE CLK TO SEE DNA
2449 010654 052777 020000 171044 BIS #CLK,@TXCSR ;POKE CLK
2450 010662 012700 000000 MOV #0,R0 ;EXPECTED
2451 010666 017701 171034 MOV @TXCSR,R1 ;ACTUAL
2452 010672 042701 077777 BIC #77777,R1 ;SAVE DNA ONLY
2453 010676 020001 CMP R0,R1 ;COMPARE EXP VS ACT
2454 010700 001401 BEQ +4
2455 010702 104003 ERROR 3 ;DNA SHOULD BE SET
2456          ;IF DNA DID NOT SET
2457          ;CHECK WORD LENGTH SELECT LOGIC
2458
2459
2460          ,THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2461          ,OF THE TRANSMITTER SECTION
2462          ,IT ALSO CHECKS DNA TIMING
2463          ,MODE SYNINT
2464          ,LENGTH SIX PLUS PARITY
    
```

```

2465          ,. PARITY: EVEPAR
2466          ,. CHARACTER: 25
2467          ,.
2468          ,. *****
2469 010704 000004 TST22 SCOPE
2470 010706 052777 000400 171012 BIS #MRESET,@TXCSR ; MASTER RESET
2471 010714 012777 030000 171000 MOV #SYNINT,@PARCSR ; SET THE MODE
2472 010722 052777 000400 170776 BIS #MRESET,@TXCSR ; MASTER RESET
2473
2474          , SET MAINTENANCE MODE & SEND
2475          , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2476 010730 012777 004020 170770 MOV #MINT!SEND,@TXCSR
2477
2478          , SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2479 010736 012777 033426 170756 MOV #SYNINT!SIX!EVEPAR!26,@PARCSR
2480 010744 016703 170756 MOV TXCSR,R3 ; SET UP FOR ERROR MSG
2481 010750 112777 000025 170754 MOVB #25,@TXDBUF ; LOAD DATA CHAR
2482 010756 012767 000125 170514 MOV #125,$TMP1 ; TO BE SHIFTED CHAR
2483 010764 012767 000007 170130 MOV #7,SHIFT ; # OF SHIFTS
2484          , POKE CLK TO GET INTO SYNCHRONIZATION
2485 010772 052777 020000 170726 BIS #CLK,@TXCSR ; POKE CLK UP
2486 011000 042777 020000 170720 BIC #CLK,@TXCSR ; POKE CLK DOWN
2487 011006 005000 15 CLR R0
2488 011010 006067 170464 ROR $TMP1 ; FORCE CARRY
2489 011014 103002 BCC 25 , BR IF CARRY CLR
2490 011016 052700 002000 BIS #BITW,R0 ; EQUIV OF BITW
2491 011022 25
2492 011022 052777 020000 170676 BIS #CLK,@TXCSR ; POKE CLK UP
2493 011030 042777 020000 170670 BIC #CLK,@TXCSR ; POKE CLK DOWN
2494 011036 017701 170664 MOV @TXCSR,R1 ; ACTUAL
2495 011042 042701 075777 BIC #075777,R1 ; SAVE BITW & DNA
2496 011046 020001 CMP R0,R1 ; COMPARE EXP VS ACT
2497 011050 001401 BEQ +4
2498 011052 104003 ERROR 3 ; BIT WINDOW DID NOT MATCH ACTUAL DATA
2499          ; BIT, ALSO CHECK DNA
2500 011054 005367 170042 DEC SHIFT ; # OF SHIFTS
2501 011060 001352 BNE 15 ; DO IT AGAIN ?
2502          , NOW POKE CLK TO SEE DNA
2503 011062 052777 020000 170636 BIS #CLK,@TXCSR ; POKE CLK
2504 011070 012700 100000 MOV #100000,R0 ; EXPECTED
2505 011074 017701 170626 MOV @TXCSR,R1 ; ACTUAL
2506 011100 042701 077777 BIC #77777,R1 ; SAVE DNA ONLY
2507 011104 020001 CMP R0,R1 ; COMPARE EXP VS ACT
2508 011106 001401 BEQ +4
2509 011110 104003 ERROR 3 ; DNA SHOULD BE SET
2510          ; IF DNA DID NOT SET
2511          ; CHECK WORD LENGTH SELECT LOGIC
2512
2513          ,. THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2514          ,. OF THE TRANSMITTER SECTION
2515          ,. IT ALSO CHECKS DNA TIMING
2516          ,. MODE SYNINT
2517          ,. LENGTH SIX PLUS PARITY
2518          ,. PARITY ODDPAR
2519          ,. CHARACTER 25
2520          ,.
    
```

```

2521      , , *****
2522 011112 000004      TST23  SCOPE
2523 011114 052777 000400 170604      BIS      #MRESET,@TXCSR ; MASTER RESET
2524 011122 012777 030000 170572      MOV      #SYNINT,@PARCSR ; SET THE MODE
2525 011130 052777 000400 170570      BIS      #MRESET,@TXCSR ; MASTER RESET
2526
2527      ; SET MAINTENANCE MODE & SEND
2528      , NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2529 011136 012777 004020 170562      MOV      #MINT!SEND,@TXCSR
2530
2531      ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2532 011144 012777 033026 170550      MOV      #SYNINT!SIX!ODDPAR!26,@PARCSR
2533 011152 016703 170550      MOV      TXCSR,R3 ; SET UP FOR ERROR MSG
2534 011156 112777 000025 170546      MOV      #25,@TXDBUF ; LOAD DATA CHAR
2535 011164 012767 000025 170306      MOV      #25,$TMP1 ; TO BE SHIFTED CHAR
2536 011172 012767 000007 167722      MOV      #7,SHIFT ; # OF SHIFTS
2537      ; POKE CLK TO GET INTO SYNCHRONIZATION
2538 011200 052777 020000 170520      BIS      #CLK,@TXCSR ; POKE CLK UP
2539 011206 042777 020000 170512      BIC      #CLK,@TXCSR ; POKE CLK DOWN
2540 011214 005000      15      CLR      RO
2541 011216 006067 170256      ROR      $TMP1 ; FORCE CARRY
2542 011222 103002      BCC      25 ; BR IF CARRY CLR
2543 011224 052700 002000      BIS      #BITW,RO ; EQUIV OF BITW
2544 011230      25
2545 011230 052777 020000 170470      BIS      #CLK,@TXCSR ; POKE CLK UP
2546 011236 042777 020000 170462      BIC      #CLK,@TXCSR ; POKE CLK DOWN
2547 011244 017701 170456      MOV      @TXCSR,R1 ; ACTUAL
2548 011250 042701 075777      BIC      #075777,R1 ; SAVE BITW & DNA
2549 011254 020001      CMP      RO,R1 ; COMPARE EXP VS ACT
2550 011256 001401      BEQ      +4
2551 011260 104003      ERROR   3 ; BIT WINDOW DID NOT MATCH ACTUAL DATA
2552      ; BIT, ALSO CHECK DNA
2553 011262 005367 167634      DEC      SHIFT ; # OF SHIFTS
2554 011266 001352      BNE      15 ; DO IT AGAIN ?
2555      ; NOW POKE CLK TO SEE DNA
2556 011270 052777 020000 170430      BIS      #CLK,@TXCSR ; POKE CLK
2557 011276 012700 100000      MOV      #100000,RO ; EXPECTED
2558 011302 017701 170420      MOV      @TXCSR,R1 ; ACTUAL
2559 011306 042701 077777      BIC      #77777,R1 ; SAVE DNA ONLY
2560 011312 020001      CMP      RO,R1 ; COMPARE EXP VS ACT
2561 011314 001401      BEQ      +4
2562 011316 104003      ERROR   3 ; DNA SHOULD BE SET
2563      ; IF DNA DID NOT SET
2564      ; CHECK WORD LENGTH SELECT LOGIC
2565
2566      ; THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2567      ; OF THE TRANSMITTER SECTION
2568      ; IT ALSO CHECKS DNA TIMING
2569      ; MODE. ISYMOD
2570      ; LENGTH SIX PLUS PARITY
2571      ; PARITY. EVEPAR
2572      ; CHARACTER 25
2573      ;
2574      , , *****
2575 011320 000004      TST24  SCOPE
2576 011322 052777 000400 170376      BIS      #MRESET,@TXCSR ; MASTER RESET
  
```

INITIALIZE THE COMMON TAGS

```

2577 011330 012777 000000 170364      MOV    #ISYMOD,@PARCSR ,SET THE MODE
2578 011336 052777 000400 170362      BIS    #MRESET,@TXCSR ,MASTER RESET
2579
2580      ,SET MAINTENANCE MODE & SEND
2581      ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2582 011344 012777 004020 170354      MOV    #MINT'SEND,@TXCSR
2583
2584      ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2585 011352 012777 003426 170342      MOV    #ISYMOD!SIX!EVEPAR!26,@PARCSR
2586 011360 016703 170342      MOV    TXCSR,R3      ;SET UP FOR ERROR MSG
2587 011364 112777 000025 170340      MOVB  #25,@TXDBUF   ;LOAD DATA CHAR
2588 011372 012767 000652 170100      MOV    #652,$TMP1   ;TO BE SHIFTED CHAR
2589 011400 012767 000011 167514      MOV    #9,SHIFT     ,# OF SHIFTS
2590      ,POKE CLK TO GET INTO SYNCHRONIZATION
2591 011406 052777 020000 170312      BIS    #CLK,@TXCSR  ,POKE CLK UP
2592 011414 042777 020000 170304      BIC    #CLK,@TXCSR  ,POKE CLK DOWN
2593 011422 005000      1$    CLR    R0
2594 011424 006067 170050      ROR    $TMP1 ,FORCE CARRY
2595 011430 103002      BCC   2$ ,BR IF CARRY CLR
2596 011432 052700 002000      BIS    #BITW,R0     ,EQUIV OF BITW
2597 011436      2$
2598 011436 052777 020000 170262      B S    #CLK,@TXCSR  ,POKE CLK UP
2599 011444 042777 020000 170254      BIC    #CLK,@TXCSR  ,POKE CLK DOWN
2600 011452 017701 170250      MOV    @TXCSR,R1    ,ACTUAL
2601 011456 042701 075777      BIC    #075777,R1   ,SAVE BITW & DNA
2602 011462 020001      CMP    R0,R1      ;COMPARE EXP VS ACT
2603 011464 001401      BEQ    +4
2604 011466 104003      EPROR  3 ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2605      ,BIT, . ALSO CHECK DNA
2606 011470 005367 167426      DEC    SHIFT      ;# OF SHIFTS
2607 011474 001352      BNE   1$ ,DO IT AGAIN ?
2608      ,NOW POKE CLK TO SEE DNA
2609 011476 052777 020000 170222      BIS    #CLK,@TXCSR  ,POKE CLK
2610 011504 012700 000000      MOV    #0,R0 ,EXPECTED
2611 011510 017701 170212      MOV    @TXCSR,R1    ,ACTUAL
2612 011514 042701 077777      BIC    #77777,R1    ,SAVE DNA ONLY
2613 011520 020001      CMP    R0,R1      ,COMPARE EXP VS ACT
2614 011522 001401      BEQ    +4
2615 011524 104003      ERROR  3 ,DNA SHOULD BE SET
2616      ,IF DNA DID NOT SET
2617      ,CHECK WORD LENGTH SELECT LOGIC
2618
2619      ;, THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2620      ;, OF THE TRANSMITTER SECTION
2621      ;, IT ALSO CHECKS DNA TIMING
2622      ;, MODE ISYMOD
2623      ;, LENGTH SIX PLUS PARITY
2624      ;, PARITY ODDPAR
2625      ;, CHARACTER 25
2626      ;,
2627      ;, *****
2628 011526 000004      TST25 SCOPE
2629 011530 052777 000400 170170      BIS    #MRESET,@TXCSR ,MASTER RESET
2630 011536 012777 000000 170156      MOV    #ISYMOD,@PARCSR ,SET THE MODE
2631 011544 052777 000400 170154      BIS    #MRESET,@TXCSR ,MASTER RESET
2632

```

INITIALIZE THE COMMON TAGS

```
2633      ,SET MAINTENANCE MODE & SEND
2634      ,NOTE. BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2635 011552 012777 004020 170146      MOV      #MINT'SEND,@TXCSR
2636
2637      ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2638 011560 012777 003026 170134      MOV      #1SYMOD!SIX!ODDPAR!26,@PARCSR
2639 011566 016703 170134              MOV      TXCSR,R3      ;SET UP FOR ERROR MSG
2640 011572 112777 000025 170132      MOV      #25,@TXDBUF  ;LOAD DATA CHAR
2641 011600 012767 000452 167672      MOV      #452,$TMP1   ;TO BE SHIFTED CHAR
2642 011606 012767 000011 167306      MOV      #9,SHIFT     ;# OF SHIFTS
2643
2644 011614 052777 020000 170104      ,POKE CLK TO GET INTO SYNCHRONIZATION
2645 011622 042777 020000 170076      BIS      #CLK,@TXCSR  ;POKE CLK UP
2646 011630 005000              BIC      #CLK,@TXCSR  ;POKE CLK DOWN
2647 011632 006067 167642      1$      CLR      R0
2648 011636 103002              ROR      $TMP1      ,FORCE CARRY
2649 011640 052700 002000              BCC      2$          ,BR IF CARRY CLR
2650 011644              BIS      #BITW,R0     ,EQUIV OF BITW
2651 011644 052777 020000 170054      2$      BIS      #CLK,@TXCSR  ;POKE CLK UP
2652 011652 042777 020000 170046      BIC      #CLK,@TXCSR  ;POKE CLK DOWN
2653 011660 017701 170042      MOV      @TXCSR,R1    ,ACTUAL
2654 011664 042701 075777      BIC      #075777,R1   ,SAVE BITW & DNA
2655 011670 020001      CMP      R0,R1      ,COMPARE EXP VS ACT
2656 011672 001401      BEQ      +4
2657 011674 104003      ERROR   3          ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2658              ,BIT,.. ALSO CHECK DNA
2659 011676 005367 167220      DEC      SHIFT      ,# OF SHIFTS
2660 011702 001352      BNE      1$          ,DO IT AGAIN ?
2661      ,NOW POKE CLK TO SEE DNA
2662 011704 052777 020000 170014      BIS      #CLK,@TXCSR  ;POKE CLK
2663 011712 012700 000000      MOV      #0,R0      ;EXPECTED
2664 011716 017701 170004      MOV      @TXCSR,R1   ,ACTUAL
2665 011722 042701 077777      BIC      #77777,R1   ,SAVE DNA ONLY
2666 011726 020001      CMP      R0,R1      ,COMPARE EXP VS ACT
2667 011730 001401      BEQ      +4
2668 011732 104003      ERROR   3          ,DNA SHOULD BE SET
2669              ,IF DNA DID NOT SET
2670              ,CHECK WORD LENGTH SELECT LOGIC
2671
2672
2673
2674      ;,THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2675      ;,OF THE TRANSMITTER SECTION
2676      ;,IT ALSO CHECKS DNA TIMING
2677      ;,MODE SYNINT
2678      ;,LENGTH SEVEN PLUS PARITY
2679      ;,PARITY EVEPAR
2680      ;,CHARACTER: 125
2681      ;,
2682      ;,*****
2683 011734 000004      TST26  SCOPE
2684 011736 052777 000400 167762      BIS      #MRESET,@TXCSR ,MASTER RESET
2685 011744 012777 030000 167750      MOV      #SYNINT,@PARCSR ,SET THE MODE
2686 011752 052777 000400 167746      BIS      #MRESET,@TXCSR ,MASTER RESET
2687
2688      ,SET MAINTENANCE MODE & SEND
```

```

2689 ;NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2690 011760 012777 004020 167740 MOV #MINT!SEND,@TXCSR
2691
2692 ,SET MODE,# OF BITS,PARITY SENSE,& LOAD SYNC REG
2693 011766 012777 035426 167726 MOV #SYNINT!SEVEN!EVEPAR!26,@PARCSR
2694 011774 016703 167726 MOV TXCSR,R3 ;SET UP FOR ERROR MSG
2695 012000 112777 000125 167724 MOVB #125,@TXDBUF ;LOAD DATA CHAR
2696 012006 012767 000125 167464 MOV #125,$TMP1 ;TO BE SHIFTED CHAR
2697 012014 012767 000010 167100 MOV #8,SHIFT ;# OF SHIFTS
2698 ,POKE CLK TO GET INTO SYNCRONIZATION
2699 012022 052777 020000 167676 BIS #CLK,@TXCSR ,POKE CLK UP
2700 012030 042777 020000 167670 BIC #CLK,@TXCSR ,POKE CLK DOWN
2701 012036 005000 15 CLR R0
2702 012040 006067 167434 ROR $TMP1 ,FORCE CARRY
2703 012044 103002 BCC $S ,BR IF CARRY CLR
2704 012046 052700 002000 BIS #BITW,R0 ,EQUIV OF BITW
2705 25
2706 012052 052777 020000 167646 BIS #CLK,@TXCSR ,POKE CLK UP
2707 012060 042777 020000 167640 BIC #CLK,@TXCSR ,POKE CLK DOWN
2708 012066 017701 167634 MOV @TXCSR,R1 ,ACTUAL
2709 012072 042701 075777 BIC #075777,R1 ,SAVE BITW & DNA
2710 012076 020001 CMP R0,R1 ,COMPARE EXP VS ACT
2711 012100 001401 BEQ +4
2712 012102 104003 ERROR 3 ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2713 ,BIT, ALSO CHECK DNA
2714 012104 005367 167012 DEC SHIFT ,# OF SHIFTS
2715 012110 001352 BNE 15 ,DO IT AGAIN ?
2716 ,NOW POKE CLK TO SEE DNA
2717 012112 052777 020000 167606 BIS #CLK,@TXCSR ,POKE CLK
2718 012120 012700 100000 MOV #100000,R0 ,EXPECTED
2719 012124 017701 167576 MOV @TXCSR,R1 ,ACTUAL
2720 012130 042701 077777 BIC #77777,R1 ,SAVE DNA ONLY
2721 012134 020001 CMP R0,R1 ,COMPARE EXP VS ACT
2722 012136 001401 BEQ +4
2723 012140 104003 ERROR 3 ,DNA SHOULD BE SET
2724 ,IF DNA DID NOT SET
2725 ,CHECK WORD LENGTH SELECT LOG C
2726
2727 ,,THIS TEST VERIFYS CHARACTER PLUS PARITY GENERATION
2728 ,,OF THE TRANSMITTER SECTION
2729 ,,IT ALSO CHECKS DNA TIMING
2730 ,,MODE SYNINT
2731 ,,LENGTH SEVEN PLUS PARITY
2732 ,,PARITY ODDPAR
2733 ,,CHARACTER 125
2734 ,,
2735 ,,*****
2736 012142 000004 TST27 SCOPE
2737 012144 052777 000400 167554 BIS #MRESET,@TXCSR ,MASTER PESET
2738 012152 012777 030000 167542 MOV #SYNINT,@PARCSR ,SET THE MODE
2739 012160 052777 000400 167540 BIS #MRESET,@TXCSR ,MASTER RESET
2740
2741 ,SET MAINTENANCE MODE & SEND
2742 ,NOTE BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2743 012166 012777 004020 167532 MOV #MINT!SEND,@TXCSR
2744

```

```

2745      ,SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2746 012174 012777 035026 167520      MOV      #SYNINT!SEVEN!ODDPAR!26, @PARCSR
2747 012202 016703 167520      MOV      TXCSR, R3      ,SET UP FOR ERROR MSG
2748 012206 112777 000125 167516      MOVB    #125, @TXDBUF  ,LOAD DATA CHAR
2749 012214 012767 000325 167256      MOV      #325, STMP1   ,TO BE SHIFTED CHAR
2750 012222 012767 000010 166672      MOV      #8, SHIFT     ,# OF SHIFTS
2751      ,POKE CLK TO GET INTO SYNCHRONIZATION
2752 012230 052777 020000 167470      BIS      #CLK, @TXCSR  ,POKE CLK UP
2753 012236 042777 020000 167462      BIC      #CLK, @TXCSR  ,POKE CLK DOWN
2754 012244 005000      1$      CLR      RO
2755 012246 006067 167226      ROR      STMP1      ,FORCE CARRY
2756 012252 103002      BCC     2$      ,BR IF CARRY CLR
2757 012254 052700 002000      B S      #BITW, RO     ,EQUIV OF BITW
2758 012260      2$
2759 012260 052777 020000 167440      B S      #CLK, @TXCSR  ,POKE CLK UP
2760 012266 042777 020000 167432      BIC      #CLK, @TXCSR  ,POKE CLK DOWN
2761 012274 017701 167426      MOV      @TXCSR, R1   ,ACTUAL
2762 012300 042701 075777      BIC      #075777, R1   ,SAVE BITW & DNA
2763 012304 020001      CMP     RO, R1      ,COMPARE EXP VS ACT
2764 012306 001401      BEQ     +4
2765 012310 104003      ERROR   3      ,BIT WINDOW DID NOT MATCH ACTUAL DATA
2766      ,BIT, ALSO CHECK DNA
2767 012312 005367 166604      DEC     SHIFT      ,# OF SHIFTS
2768 012316 001352      BNE     1$      ,DO IT AGAIN ?
2769      ,NOW POKE CLK TO SEE DNA
2770 012320 052777 020000 167400      BIS      #CLK, @TXCSR  ,POKE CLK
2771 012326 012700 100000      MOV      #100000, RO   ,EXPECTED
2772 012332 017701 167370      MOV      @TXCSR, R1   ,ACTUAL
2773 012336 042701 077777      BIC      #77777, R1   ,SAVE DNA ONLY
2774 012342 020001      CMP     RO, R1      ,COMPARE EXP VS ACT
2775 012344 001401      BEQ     +4
2776 012346 104003      ERROR   3      ,DNA SHOULD BE SET
2777      ,IF DNA DID NOT SET
2778      ,CHECK WORD LENGTH SELECT LOGIC
2779
2780
2781
2782      ,END OF PASS
2783      ,TYPE NAME OF TEST
2784      ,UPDATE PASS COUNT
2785      ,CHECK FOR EXIT TO ACT-11
2786      ,RESTART TEST
2787
2788 012350 000004      EOP     SCOPE
2789 012352 004767 000340      JSR     PC, CKSWR
2790 012356 104401      TYPE
2791 012360 015506      MEPASS      ,TYPE NAME OF TEST
2792 012362 104413 012614      CONVRT  ,OUTCRY
2793 012366 104401 015325      TYPE    ,DEVICE
2794 012372 105767 166554      TSTB    MULTD      ,ARE YOU RUNNING MULTIPLE DEVICES ?
2795 012376 001511      BEQ     CCC      ,NO, JUMP AROUND
2796 012400 005767 166562      TST     ACTREG    ,ARE ANY DEVICES ACTIVE ?
2797 012404 001007      BNE     RUNIT     ,YES
2798 012406 104401 015337      TYPE    ,MCOW      ,NO
2799 012412 016700 166550      MOV     ACTREG, RO  ,DISPLAY ACTREG
2800 012416 000000      HALT    ,SELECT SOMETHING TO RUN @ ACTREG
  
```



```

2801      , SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2802 012420 000167 167526      JMP      START , START OVER AGAIN... YOU DESELECTED EVERYTHING
2803 012424 062767 000010 166522 RUNIT  ADD      #10, BASEADD , NEXT BLOCK (ADDRESSES)
2804 012432 062767 000010 166522 ZERO   ADD      #10, BASEIV  , NEXT BLOCK (VECTORS)
2805 012440 000241                CLC
2806 012442 006167 166522      ROL      ROTADD ; UP DATE ROTATING POINTER
2807 012446 103410                BCS     25      , IS IT THE LAST DEVICE
2808                        , TO BE TESTED IN THIS PASS ?
2809 012450 036767 166514 166510      BIT     ROTADD, ACTREG ; TEST THIS DEVICE FOR ACTIVE STATUS
2810 012456 001762                BEQ     RUNIT ; IF NOT ACTIVE, TRY NEXT ADDRESS
2811 012460 004767 000034                JSR     PC, REPLAY , CALCULATE NEW PARAMETERS
2812 012464 000167 000210                JMP     RESTRT , YES IT WAS ACTIVE, TEST THIS DEVICE
2813 012470 012767 000001 166472 25     MOV     #1, ROTADD ; OK!, NOW SET UP ROTATING
2814                        , POINTER FOR NEXT MULTIPLE PASS
2815 012476 016767 166454 166450      MOV     KEEPADD, BASEADD , RESTORE BASE ADDRESS
2816 012504 016767 166454 166450      MOV     KEEPIV, BASEIV , RESTORE BASE INTERRUPT VECTORS
2817 012512 004767 000002                JSR     PC, REPLAY , CALC NEW PARAMETERS
2818 012516 000441                BR      CCC , JUMP AROUND REPLAY
2819 012520 016767 166430 004416 REPLAY MOV     BASEADD, DUBASE , SET UP FOR NEW ADDRESSES
2820 012526 004767 004260                JSR     PC, DUADDR , CREATE NEW ADDRESSES
2821 012532 016767 166424 167176      MOV     BASEIV, DURIV , CREATE DURIV
2822 012540 062767 000002 166414      ADD     #2, BASEIV
2823 012546 016767 166410 167164      MOV     BASEIV, DURIS , CREATE DURIS
2824 012554 062767 000002 166400      ADD     #2, BASEIV
2825 012562 016767 166374 167152      MOV     BASEIV, DUTIV , CREATE DUTIV
2826 012570 062767 000002 166364      ADD     #2, BASEIV
2827 012576 016767 166360 167140      MOV     BASEIV, DUTIS , CREATE DUTIS
2828 012604 016767 167126 166350      MOV     DURIV, BASEIV , RESTORE
2829 012612 000207                RTS     PC
2830
2831 012614 000001                OUTCRY 1
2832 012616 006 002                BYTE   6, 2
2833 012620 001712                RXCSR
2834
2835                        CCC
2836 012622 005067 166554                CLR     $TSTNM , CLEAR TEST NUMBER
2837 012626 005067 166564                CLR     $ERRPC , CLEAR LAST ERROR PC
2838 012632 005067 166545                CLR     $ERFLG , CLEAR ERROR FLAG
2839 012636 005267 166250                INC     PASCNT , UPDATE PASS COUNT
2840 012642 016767 166244 166232      MOV     PASCNT, LIGHTS , DISPLAY PASS COUNT
2841 012650 016767 166236 166656      MOV     PASCNT, $PASS , PASS COUNT TO APT
2842 012656 013701 000042                MOV     @#42, R1 , CHECK FOR ACT-11 OR DDP
2843 012662 001406                BEQ     RESTRT , IF NO CONTINUE TESTING
2844 012664 000005                RESET
2845 012666 000005                RESET
2846 012670 004711                SENDAD JSR     PC, (R1)
2847 012672 000240                NOP
2848 012674 000240                NOP
2849 012676 000240                NOP
2850 012700                RESTRT
2851 012700 012767 003376 166500      MOV     #TST1+2, $LPADR , LOAD LAST ADDR
2852 012706 004767 000004                JSR     PC, CKSWR
2853 012712 000167 170372                JMP     BEGIN
2854
2855                        , CHECK SWITCH REGISTER ROUTINE
2856                        , CHECKS TO ALLOW FOR < G> TO ALLOW

```

```

2857                                     , THE CHANGING OF LOCATION 176
2858
2859 012716 005737 000042          CKSWR  TST      @#42
2860 012722 001040                  BNE      OUT
2861 012724 022767 000176 166506    CMP      #SWREG, SWR      , SOFTWARE SWR PRESENT?
2862 012732 001034                  BNE      OUT              , NO--LEAVE
2863 012734 105777 166504          TSTB     @STKS           , CHECK TTY READY
2864 012740 100031                  BPL      OUT              , NO--LEAVE
2865 012742 017767 166500 000422    MOV      @STKB, MSG      , GET CHARACTER
2866 012750 042767 177600 000414    BIC      #177600, MSG    , STRIP JUNK
2867 012756 122767 000007 000406    CMPB     #7, MSG         , IS IT < G> ?
2868 012764 001017                  BNE      OUT              , NO
2869 012766 104401 016113          TYPE     , MCNTG
2870 012772 005137 013032          CNTLU   COM      @#RDSW
2871 012776 104401 016123          TYPE     , MMSWR
2872 013002 104413                  CONVRT
2873 013004 013034                  SWREGL
2874 013006 104406 016134          INSTR, MMNEW
2875 013012 104410                  PARAM
2876 013014 000000                  0
2877 013016 177777                  177777
2878 013020 000176                  SWREG
2879 013022      000      001      BYTE  0, 1
2880 013024 005037 013032          OUT     CLR      @#RDSW
2881 013030 000207                  RTS      PC
2882 013032 000000          RDSW   WORD     0
2883 013034 000001          SWREGL 1
2884 013036      006      002      BYTE  6, 2
2885 013040 000176                  SWREG
2886
2887 013042 000005                  5
2888
2889                                     , CHECK FOR FREEZE ON CURRENT DATA
2890
2891 013044 004767 177646          SCOP1   JSR      PC, CKSWR
2892 013050 032777 001000 166362    BIT      #SW09, @SWR
2893 013056 001402                  BEQ      15
2894 013060 016716 166024          MOV      LOCK, (SP)
2895 013064 000002          15    RTI
2896          SBTTL  TYPE ROUTINE
2897
2898                                     , *****
2899                                     , *ROUTINE TO TYPE ASCIZ MESSAGE MESSAGE MUST TERMINATE WITH A 0 BYTE
2900                                     , *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED
2901                                     , *NOTE1          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER
2902                                     , *NOTE2          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2903                                     , *NOTE3          $r ILLC CONTAINS THE CHARACTER TO FILL AFTER
2904                                     , *
2905                                     , *CALL
2906                                     , *1) USING A TRAP INSTRUCTION
2907                                     , *      TYPE      , MESADR      , MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2908                                     , *
2909                                     , *      TYPE
2910                                     , *      MESADR
2911                                     , *
2912

```

2913	013066	105767	166365		\$TYPE	TSTB	\$TPFLG	.. IS THERE A TERMINAL?
2914	013072	100002				BPL	1\$	.. BR IF YES
2915	013074	000000				HALT		.. HALT HERE IF NO TERMINAL
2916	013076	000430				BR	3\$	.. LEAVE
2917	013100	010046			1\$	MOV	RO, -(SP)	.. SAVE RO
2918	013102	017600	000002			MOV	2(SP), RO	.. GET ADDRESS OF ASCIZ STRING
2919	013106	122767	000001	166432		CMPB	#APTENV, \$ENV	.. RUNNING IN APT MODE
2920	013114	001011				BNE	62\$	.. NO, GO CHECK FOR APT CONSOLE
2921	013116	132767	000100	166423		BITB	#APTPOOL, \$ENV	.. SPOOL MESSAGE TO APT
2922	013124	001405				BEQ	62\$	.. NO, GO CHECK FOR CONSOLE
2923	013126	010067	000004			MOV	RO, 61\$	.. SETUP MESSAGE ADDRESS FOR APT
2924	013132	004767	000006			JSR	PC, \$ATY3	.. SPOOL MESSAGE TO APT
2925	013136	000000			61\$	WORD	0	.. MESSAGE ADDRESS
2926	013140	132767	000040	166401	62\$	BITB	#APTCSUP, \$ENV	.. APT CONSOLE SUPPRESSED
2927	013146	001003				BNE	60\$	.. YES, SKIP TYPE OUT
2928	013150	112046			2\$	MOVB	(RO)+, -(SP)	.. PUSH CHARACTER TO BE TYPED ONTO STACK
2929	013152	001005				BNE	4\$	.. BR IF IT ISN'T THE TERMINATOR
2930	013154	005726				TST	(SP)+	.. IF TERMINATOR POP IT OFF THE STACK
2931	013156	012600			60\$	MOV	(SP)+, RO	.. RESTORE RO
2932	013160	062716	000002		3\$	ADD	#2, (SP)	.. ADJUST RETURN PC
2933	013164	000002				RTI		.. RETURN
2934	013166	122716	000011		4\$	CMPB	#HT, (SP)	.. BRANCH IF <HT>
2935	013172	001430				BEQ	8\$	
2936	013174	122716	000200			CMPB	#CRLF, (SP)	.. BRANCH IF NOT <CRLF>
2937	013200	001006				BNE	5\$	
2938	013202	005726				TST	(SP)+	.. POP <CR><LF> EQUIV
2939	013204	104401				TYPE		.. TYPE A CR AND LF
2940	013206	001523				\$CRLF		
2941	013210	105067	000130			CLRB	\$CHARCNT	.. CLEAR CHARACTER COUNT
2942	013214	000755				BR	2\$	.. GET NEXT CHARACTER
2943	013216	004767	000056		5\$	JSR	PC, \$TYPEC	.. GO TYPE THIS CHARACTER
2944	013222	126726	166230		6\$	CMPB	\$FILLC, (SP)+	.. IS IT TIME FOR FILLER CHARS ?
2945	013226	001350				BNE	2\$	.. IF NO GO GET NEXT CHAR
2946	013230	016746	166220			MOI	\$NULL, -(SP)	.. GET # OF FILLER CHARS NEEDED
2947								.. AND THE NULL CHAR
2948	013234	105366	000001		7\$	DECB	1(SP)	.. DOES A NULL NEED TO BE TYPED?
2949	013240	002770				BLT	6\$	.. BR IF NO--GO POP THE NULL OFF OF STACK
2950	013242	004767	000032			JSR	PC, \$TYPEC	.. GO TYPE A NULL
2951	013246	105367	000072			DECB	\$CHARCNT	.. DO NOT COUNT AS A COUNT
2952	013252	000770				BR	7\$	.. LOOP
2953								
2954								
2955								
2956	013254	112716	000040		8\$	MOVB	#', (SP)	.. REPLACE TAB WITH SPACE
2957	013260	004767	000014		9\$	JSR	PC, \$TYPEC	.. TYPE A SPACE
2958	013264	132767	000007	000052		BITB	#7, \$CHARCNT	.. BRANCH IF NOT AT
2959	013272	001372				BNE	9\$	.. TAB STOP
2960	013274	005726				TST	(SP)+	.. POP SPACE OFF STACK
2961	013276	000724				BR	2\$	.. GET NEXT CHARACTER
2962	013300	105777	166144		\$TYPEC	TSTB	2\$TPS	.. WAIT UNTIL PRINTER IS READY
2963	013304	100375				BPL	\$TYPEC	
2964	013306	116677	000002	166136		MOVB	2(SP), 2\$TPB	.. LOAD CHAR TO BE TYPED INTO DATA REG
2965	013314	122766	000015	000002		CMPB	#CR, 2(SP)	.. IS CHARACTER A CARRIAGE RETURN?
2966	013322	001003				BNE	1\$	.. BRANCH IF NO
2967	013324	105067	000014			CLRB	\$CHARCNT	.. YES--CLEAR CHARACTER COUNT
2968	013330	000406				BR	\$TYPEX	.. EXIT

```

2969 013332 122766 000012 000002 15  CMPB  #LF,2(SP)  , IS CHARACTER A LINE FEED?
2970 013340 001402  BEQ  STYPEX  , BRANCH IF YES
2971 013342 105227  INCB  (PC)+  , COUNT THE CHARACTER
2972 013344 000000  SCHARCNT WORD 0  , CHARACTER COUNT STORAGE
2973 013346 000207  STYPEX RTS  PC
2974
2975
2976 , ASCII STRING INPUT ROUTINE
2977
2978 013350 017667 000003 000014 INSTR MOV  @ (SP), MSG  , PICK UP MESSAGE
2979 013356 062716 000002  ADD  #2, (SP)  , JUMP AROUND MESSAGE FOR RTI
2980 013362 105767 166160  TSTB $ENV  , APT CONTROL
2981 013366 001036  BNE  INSTR2  , YES NO TYPE
2982 013370 104401  INST1 TYPE
2983 013372 000000  MSG  0
2984 013374 012704 016146  MOV  #INBUF, R4  , GET STARTING LOC OF INBUF
2985 013400 012703 000007  MOV  #7, R3  , MAX # OF CHARS
2986 013404 105777 166034 15  TSTB @ $TKS  , TTY FLAG
2987 013410 100375  BPL  15
2988 013412 117714 166030  MOVB @ $TKB, (R4)  , TAKE CHAR
2989 013416 142714 000200  BICB #200, (R4)  , STRIP
2990 013422 121427 000025  CMPB (R4), #25  , IS IT 'G'
2991 013426 001760  BEQ  INST1
2992 013430 122427 000015  CMPB (R4)+, #15  , CHECK FOR CR
2993 013434 001413  BEQ  INSTR2
2994 013436 105777 166006 25  TSTB @ $TPS  , TEST FLAG
2995 013442 100375  BPL  25
2996 013444 117777 165776 166000  MOVB @ $TKB, @ $TPB  , ECHO CHARACTER
2997 013452 005303  DEC  R3  , DID YOU TYPE TOO MANY CHAPS ?
2998 013454 001353  BNE  15
2999 013456 104401  INSTE TYPE
3000 013460 015433  MQM  , ?
3001 013462 000742  BR  INST1  , RETRY
3002 013464 000002  INSTR2 RTI
3003
3004 , CONVERT ASCII STRING TO OCTAL
3005
3006 013466 011605  PARAM MOV  (SP), R5  , PUT CONTENTS OF SP INTO R5
3007 013470 012567 000162  MOV  (R5)+, LOLIM  , PUT LOW LIMIT INTO LOLIM
3008 013474 012567 000160  MOV  (R5)+, HILIM  , PUT HIGH LIMIT INTO HILIM
3009 013500 012567 000156  MOV  (R5)+, DEVADR  , PUT STORE LOC INTO DEVADR
3010 013504 112567 000154  MOVB (R5)+, LOBITS  , PUT MASK INTO LOBITS
3011 013510 112567 000151  MOVB (R5)+, ADCNT  , PUT COUNT INTO ADCNT
3012 013514 010516  MOV  R5, (SP)  , RESTORE RETURN ADDR ON STACK FOR RTI
3013 013516 005005  PARAM1 CLR  R5
3014 013520 012704 016146  MOV  #INBUF, R4
3015 013524 122714 000015  CMPB #15, (R4)  , CR ?
3016 013530 001420  BEQ  PARERR  , YOU TYPED CR TOO SOON !
3017 013532 121427 000060 15  CMPB (R4), #60  , LOW LIMIT ASCII 0
3018 013536 002415  BLT  PARERR
3019 013540 121427 000067  CMPB (R4), #67  , HIGH LIMIT ASCII ?
3020 013544 003012  BGT  PARERR
3021 013546 142714 000060  BICB #60, (R4)  , CONVERT TO OCTAL
3022 013552 152405  BISB (R4)+, R5  , STORE AWAY ITS AN OK CHAR
3023 013554 122714 000015  CMPB #15, (R4)  , CR ?
3024 013560 001414  BEQ  LIMITS  , NOW CHECK FOR HIGH & LOW LIMIT CONDS

```

```

3025 013562 006305          ASL    R5      , ALLOCATE ROOM FOR NEXT CHAR
3026 013564 006305          ASL    R5
3027 013566 006305          ASL    R5
3028 013570 000760          BR     1$
3029 013572 122714 000015      PARERR CMPB   #15, (R4)      , C?
3030 013576 001003          BNE   120$
3031 013600 005737 013032      TST   @#RDSW      , CK SWR USED
3032 013604 001023          BNE   PARTI
3033 013606 104407      120$  INSTER , RETRY
3034 013610 000742          BR     PARAM1
3035
3036          , TEST TO SEE IF NUMBER IS WITHIN LIMITS
3037
3038 013612 020567 000042      LIMITS CMP    R5, HILIM
3039 013616 101365          BHI   PARERR , THE # IS TOO HIGH
3040 013620 020567 000032      CMP    R5, LOLIM
3041 013624 103762          BLO   PARERR , THE # IS TOO LOW
3042 013626 136705 000032      BITB  LOBITS, R5      , TEST BY MASKING THE #
3043 013632 001357          BNE   PAPERR
3044
3045          , STORE NUMBER AT SPECIFIED ADDRESS
3046
3047 013634 016704 000022      MOV    DEVAR, R4      , GET STARTING ADDR OF
3048 013640 010524      1$  MOV    R5, (R4)+      , STORE AT THIS ADDR
3049 013642 062705 000002      ADD    #2, R5
3050 013646 105367 000013      DECB  ADRCNT , HOW MANY TIMES + 2 ?
3051 013652 001372          BNE   1$
3052 013654 000002      PARTI RTI
3053 013656 000000      LOLIM 0
3054 013660 000000      HILIM 0
3055 013662 000000      DEVAR 0
3056 013664 000000      LOBITS 0
3057          ADRCNT=LOBITS+1
3058
3059          , SAVE PC OF TEST THAT FAILED AND R0-R5
3060
3061 013666 016667 000004 165232  SAV05 MOV    4(SP), SAVPC
3062
3063          , SAVE R0-R5
3064
3065 013674 010567 165574      SV05  MOV    R5, $REG5
3066 013700 010467 165566      MOV    R4, $REG4
3067 013704 010367 165560      MOV    R3, $REG3
3068 013710 010267 165552      MOV    R2, $REG2
3069 013714 010167 165544      MOV    R1, $REG1
3070 013720 010067 165536      MOV    R0, $REG0
3071 013724 000002          RTI
3072
3073          , RESTORE R0-R5
3074
3075 013726 016700 165530      RES05 MOV    $REG0, R0
3076 013732 016701 165526      MOV    $REG1, R1
3077 013736 016702 165524      MOV    $REG2, R2
3078 013742 016703 165522      MOV    $REG3, R3
3079 013746 016704 165520      MOV    $REG4, R4
3080 013752 016705 165516      MOV    $REG5, R5
  
```

```

3081 013756 000002          RTI
3082
3083          , CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3084
3085 013760 104401          CONVR TYPE
3086 013762 015437          MCRLF , CR LF
3087 013764 017601 000000  MOV @ (SP), R1 , PICK UP DATA POINTER
3088 013770 062716 000002  ADD #2, (SP) , SET UP SP FOR RTI
3089 013774 012167 000130  MOV (R1)+, WRDCNT ; PICK UP # OF WORDS FROM TABLE
3090 014000 112167 000126 15  MOVB (R1)+, CHRCNT , PICK UP # OF CHARS FROM TABLE
3091 014004 112167 000123  MOVB (R1)+, SPACNT , PICK UP # OF SPACES FROM TABLE
3092 014010 013167 000120  MOV @ (R1)+, BINWRD , PICK UP ADDRESS OF MSG
3093          , FROM TABLE
3094 014014 016704 000114 25  MOV BINWRD, R4 , SAVE
3095 014020 116705 000106  MOVB CHRCNT, R5 ; SAVE
3096 014024 012700 016210  MOV #TEMP, R0 , STARTING ADDRESS OF TEMP BLOCK
3097 014030 010403 35  MOV R4, R3 , SAVE
3098 014032 042703 177770  BIC #177770, R3 , CLR OUT UPPER BITS SAVE CHAR
3099 014036 062703 000260  ADD #260, R3 , CONVERT TO ASCII
3100 014042 110320          MOVB R3, (R0)+ , STORE AWAY
3101 014044 006204          ASR R4 , SHIFT FOR NEXT #
3102 014046 006204          ASR R4 , DITTO
3103 014050 006204          ASR R4 , DITTO
3104 014052 005305          DEC R5 , DEC CHAR COUNT
3105 014054 001365          BNE 35 , DO IT AGAIN ?
3106 014056 012703 016252  MOV #MDATA, R3 , STARTING ADDRESS OF MDATA BLOCK
3107 014062 114023 45  MOVB -(R0), (R3)+ , REVERSE THE ORDER OF NUMBERS
3108 014064 105367 000042  DECB CHRCNT , DEC CHAR COUNT
3109 014070 001374          BNE 45 , DO IT AGAIN ?
3110 014072 105767 000035  TSTB SPACNT , HOW MANY SPACES ?
3111 014076 001405          BEQ 65 ; TYPE # IF BR = 0
3112 014100 112723 000240 55  MOVB #240, (R3)+ , "SPACE" IN ASCII
3113 014104 105367 000023  DECB SPACNT , DEC # OF SPACE COUNT
3114 014110 001373          BNE 55 , DO IT AGAIN ?
3115 014112 105013 65  CLRB (R3) , INSERT "0" FOR TTY OUTPUT ROUTINE
3116 014114 104401          TYPE
3117 014116 016252          MDATA , THIS MESSAGE
3118 014120 005367 000004  DEC WRDCNT , HOW MANY #'S ?
3119 014124 001325          BNE 15 , DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3120 014126 000002          RTI , RETURN TO PROGRAM
3121 014130 000000          WRDCNT 0
3122 014132 000000          CHRCNT 0
3123          014133          SPACNT=CHRCNT+1
3124 014134 000000          BINWRD 0
3125
3126          , COMPARE THE FIRST CHARACTER IN THE TELETYPE NPUT
3127          , BUFFER TO THE CHARACTERS "N" AND "Y"
3128          , IF THE CHARACTER IS "N" CLEAR THE FLAG
3129          , IF THE CHARACTER IS "Y" SET THE FLAG
3130
3131 014136 017605 000000          SETFLG MOV @ (SP), R5
3132 014142 122767 000116 001776  CMPB #'N, INBUF , IS IT "N" ?
3133 014150 001002          BNE 15
3134 014152 105015          CLRB (R5) , 000
3135 014154 000406          BR 25
3136 014156 122767 000131 001762 15  CMPB #'Y, INBUF , IS IT "Y" ?
  
```

```

3137 014164 001005          BNE      3$
3138 014166 112715 177777  MOVB    #-1,(R5)      ,377
3139 014172 062716 000002 2$      ADD     #2,(SP)
3140 014176 000002          RTI
3141 014200 104407 3$      INSTER ,RETRY
3142 014202 000755          BR      .SETFLG
3143          SBTTL  ERROR HANDLER ROUTINE
3144
3145          ,,*****
3146          ,*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3147          ,*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3148          ,*AND GO TO SAVIT ON ERROR
3149          ,*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3150          ,*SW15=1      HALT ON ERROR
3151          ,*SW13=1      INHIBIT ERROR TYPEOUTS
3152          ,*SW10=1      BELL ON ERROR
3153          ,*SW09=1      LOOP ON ERROR
3154          ,*CALL
3155          ,*      ERROR  N      ,,ERROR=EMT AND N=ERROR ITEM NUMBER
3156
3157 014204          SERROR
3158 014204 105267 165173 7$      INCB   $ERFLG      ,,SET THE ERROR FLAG
3159 014210 001775          BEQ     7$          ,,DON'T LET THE FLAG GO TO ZERO
3160 014212 016777 165164 165222  MOV     $TSTNM,@DISPLAY ,,DISPLAY TEST NUMBER AND ERROR FLAG
3161 014220 032777 002000 165212  BIT     #BIT10,@SWR    ,,BELL ON ERROR?
3162 014226 001402          BEQ     1$          ,,NO - SKIP
3163 014230 104401 001516          TYPE   ,SBELL      ,,RING BELL
3164 014234 005267 165152 1$      INC     $ERTTL      ,,COUNT THE NUMBER OF ERRORS
3165 014240 011667 165152          MOV     (SP), $ERRPC ,,GET ADDRESS OF ERROR INSTRUCTION
3166 014244 162767 000002 165144  SUB     #2,$ERRPC
3167 014252 117767 165140 165134  MOVB   @ $ERRPC,$ITEMB ,,STRIP AND SAVE THE ERROR ITEM CODE
3168 014260 032777 020000 165152  BIT     #BIT13,@SWP    ,,SKIP TYPEOUT IF SET
3169 014266 001004          BNE     20$         ,,SKIP TYPEOUTS
3170 014270 004767 000072          JSR    PC,SAVIT      ,,GO TO USER ERROR ROUTINE
3171 014274 104401 001523          TYPE   , $CRLF
3172 014300          20$
3173 014300 122767 000001 165240  CMPB   #APTENV,$ENV   ,,RUNNING IN APT MODE
3174 014306 001007          BNE     2$          ,,NO,SKIP APT ERROR REPORT
3175 014310 116767 165100 000004  MOVB   $ITEMB,21$    ,,SET ITEM NUMBER AS ERROR NUMBER
3176 014316 004767 000016          JSR    PC,$ATY4      ,,REPORT FATAL ERROR TO APT
3177 014322 000          21$  BYTE   0
3178 014323 000          BYTE   0
3179 014324 000777          22$  BR     22$          ,,APT ERROR LOOP
3180 014326 005777 165106 2$      TST    @SWR          ,,HALT ON ERROR
3181 014332 100001          BPL     3$          ,,SKIP IF CONTINUE
3182 014334 000000          HALT   ,,HALT ON ERROR!
3183 014336 032777 001000 165074 3$      BIT     #BIT09,@SWR   ,,LOOP ON ERROR SWITCH SET?
3184 014344 001402          BEQ     4$          ,,BR IF NO
3185 014346 016716 165036          MOV     $LPERR,(SP)  ,,FUDGE RETURN FOR LOOPING
3186 014352 005767 165136 4$      TST    $ESCAPE      ,,CHECK FOR AN ESCAPE ADDRESS
3187 014356 001402          BEQ     5$          ,,BR IF NONE
3188 014360 016716 165130          MOV     $ESCAPE,(SP) ,,FUDGE RETURN ADDRESS FOR ESCAPE
3189 014364          5$
3190 014364 000002          RTI     ,,RETURN
3191 014366 010067 164536  SAVIT  MOV     R0,HLD0
3192 014372 010167 164534          MOV     R1,HLD1

```

```

3193 014376 010267 164532      MOV    R2,HL02
3194 014402 010367 164530      MOV    R3,HL03
3195 014406 010467 164526      MOV    R4,HL04
3196 014412 010567 164524      MOV    R5,HL05
3197 014416 016767 164760 164520  MOV    STSTNM,HL06
    
```

SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

, , *****
, *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
, *ERROR IS TO BE REPORTED IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
, *AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR
    
```

```

3206 014424                                $ERRTYP
3207 014424 104401 001523                TYPE    , $CRLF                , , "CARRIAGE RETURN" & "LINE FEED"
3208 014430 010046                        MOV    RO, -(SP)                , , SAVE RO
3209 014432 005000                        CLR    RO                        , , PICKUP THE ITEM INDEX
3210 014434 153700 001414                BISB   @#$ITEMB,RO
3211 014440 001004                        BNE    1$                        , , IF ITEM NUMBER IS ZERO, JUST
3212                                     , , TYPE THE PC OF THE ERROR
3213 014442 016746 164750                MOV    $ERRPC, -(SP)            , , SAVE $ERRPC FOR TYPEOUT
3214                                     , , ERROR ADDRESS
3215 014446 104402                        TYPOC                               , , GO TYPE--OCTAL ASCII(ALL DIGITS)
3216 014450 000426                        BR     6$                        , , GET OUT
3217 014452 005300 1$                    DEC    RO                        , , ADJUST THE INDEX SO THAT IT WILL
3218 014454 006300                        ASL   RO                        , , WORK FOR THE ERROR TABLE
3219 014456 006300                        ASL   RO
3220 014460 006300                        ASL   RO
3221 014462 062700 001652                ADD    #$ERRTB,RO                , , FOPM TABLE POINTER
3222 014466 012067 000004                MOV    (RO)+, 2$                , , PICKUP "ERROR MESSAGE" POINTER
3223 014472 001404                        BEQ    3$                        , , SKIP TYPEOUT IF NO POINTER
3224 014474 104401                        TYPE                               , , TYPE THE "ERROR MESSAGE"
3225 014476 000000 2$                    WORD   0                        , , "ERROR MESSAGE" POINTER GOES HERE
3226 014500 104401 001523                TYPE    , $CRLF                , , "CARRIAGE RETURN" & "LINE FEED"
3227 014504 012067 000004 3$            MOV    (RO)+, 4$                , , PICKUP "DATA HEADER" POINTER
3228 014510 001404                        BEQ    5$                        , , SKIP TYPEOUT IF 0
3229 014512 104401                        TYPE                               , , TYPE THE "DATA HEADER"
3230 014514 000000 4$                    WORD   0                        , , "DATA HEADER" POINTER GOES HERE
3231 014516 104401 001523                TYPE    , $CRLF                , , "CARRIAGE RETURN" & "LINE FEED"
3232 014522 011000 5$                    MOV    (RO),RO                  , , PICKUP "DATA TABLE" POINTER
3233 014524 001004                        BNE    7$                        , , GO TYPE THE DATA
3234 014526 012600 6$                    MOV    (SP)+,RO                 , , RESTORE RO
3235 014530 104401 001523                TYPE    , $CRLF                , , "CARRIAGE RETURN" & "LINE FEED"
3236 014534 000207                        RTS    PC                        , , RETURN
3237 014536 7$
3238 014536 013046                        MOV    @ (RO)+, -(SP)            , , SAVE @ (RO)+ FOR TYPEOUT
3239 014540 104402                        TYPOC                               , , GO TYPE--OCTAL ASCII(ALL DIGITS)
3240 014542 005710                        TST   (RO)                       , , IS THERE ANOTHER NUMBER?
3241 014544 001770                        BEQ    6$                        , , BR IF NO
3242 014546 104401 014554                TYPE    , 8$                    , , TYPE TWO(2) SPACES
3243 014552 000771                        BR     7$                        , , LOOP
3244 014554 020040 000 8$                ASCIIZ / /                        , , TWO(2) SPACES
3245 014560                        EVEN
    
```

SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

, , *****
    
```

3248



```

3249 ,*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3250 ,*OCTAL (ASCII) NUMBER AND TYPE IT
3251 ,*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3252 ,*CALL
3253 ,*      MOV      NUM, -(SP)      ,,NUMBER TO BE TYPED
3254 ,*      TYPOS    ,,CALL FOR TYPEOUT
3255 ,*      BYTE    N                ,,N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3256 ,*      BYTE    M                ,,M=1 OR 0
3257 ,*                                     ,,1=TYPE LEADING ZEROS
3258 ,*                                     ,,0=SUPPRESS LEADING ZEROS
3259 ,*
3260 ,*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3261 ,*$TYPOS OR $TYPOC
3262 ,*CALL
3263 ,*      MOV      NUM, -(SP)      ,,NUMBER TO BE TYPED
3264 ,*      TYPON    ,,CALL FOR TYPEOUT
3265 ,*
3266 ,*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3267 ,*CALL
3268 ,*      MOV      NUM, -(SP)      ,,NUMBER TO BE TYPED
3269 ,*      TYPOC    ,,CALL FOR TYPEOUT
3270
3271 014560 017646 000000      $TYPOS MOV      2(SP), -(SP)      ,, PICKUP THE MODE
3272 014564 116667 000001 000211 MOVB     1(SP), $OFILL    ,, LOAD ZERO FILL SWITCH
3273 014572 112667 000207      MOVB     (SP)+, $OMODE+1  ,, NUMBER OF DIGITS TO TYPE
3274 014576 062716 000002      ADD      #2, (SP)      ,, ADJUST RETURN ADDRESS
3275 014602 000406      BR      $TYPON
3276 014604 112767 000001 000171 $TYPOC MOVB     #1, $OFILL    ,, SET THE ZERO FILL SWITCH
3277 014612 112767 000006 000165 MOVB     #6, $OMODE+1    ,, SET FOR SIX(6) DIGITS
3278 014620 112767 000005 000154 $TYPON MOVB     #5, $OCNT      ,, SET THE ITERATION COUNT
3279 014626 010346      MOV      R3, -(SP)      ,, SAVE R3
3280 014630 010446      MOV      R4, -(SP)      ,, SAVE R4
3281 014632 010546      MOV      R5, -(SP)      ,, SAVE R5
3282 014634 116704 000145      MOVB     $OMODE+1, R4   ,, GET THE NUMBER OF DIGITS TO TYPE
3283 014640 005404      NEG      R4
3284 014642 062704 000006      ADD      #6, R4        ,, SUBTRACT IT FOR MAX ALLOWED
3285 014646 110467 000132      MOVB     R4, $OMODE    ,, SAVE IT FOR USE
3286 014652 116704 000125      MOVB     $OFILL, R4    ,, GET THE ZERO FILL SWITCH
3287 014656 016605 000012      MOV      12(SP), R5    ,, PICKUP THE INPUT NUMBER
3288 014662 005003      CLR      R3           ,, CLEAR THE OUTPUT WORD
3289 014664 006105      1$     ROL      R5     ,, ROTATE MSB INTO "C"
3290 014666 000404      BR      3$           ,, GO DO MSB
3291 014670 006105      2$     ROL      R5     ,, FORM THIS DIGIT
3292 014672 006105      ROL      R5
3293 014674 006105      ROL      R5
3294 014676 010503      MOV      R5, R3
3295 014700 006103      3$     ROL      R3     ,, GET LSB OF THIS DIGIT
3296 014702 105367 000076      DECB     $OMODE       ,, TYPE THIS DIGIT?
3297 014706 100016      BPL      7$           ,, BR IF NO
3298 014710 042703 177770      BIC      #177770, R3   ,, GET RID OF JUNK
3299 014714 001002      BNE      4$           ,, TEST FOR 0
3300 014716 005704      TST      R4           ,, SUPPRESS THIS 0?
3301 014720 001403      BEQ      5$           ,, BR IF YES
3302 014722 005204      4$     INC      R4     ,, DON'T SUPPRESS ANYMORE 0'S
3303 014724 052703 000060      BIS      #'0, R3     ,, MAKE THIS DIGIT ASCII
3304 014730 052703 000040      5$     BIS      #' , R3  ,, MAKE ASCII IF NOT ALREADY
  
```

3305	014734	110367	000040		MOVB	R3, 85	,, SAVE FOR TYPING
3306	014740	104401	015000		TYPE	, 85	,, GO TYPE THIS DIGIT
3307	014744	105367	000032	75	DECB	%OCNT	,, COUNT BY 1
3308	014750	003347			BGT	25	,, BR IF MORE TO DO
3309	014752	002402			BLT	65	,, BR IF DONE
3310	014754	005204			INC	R4	,, INSURE LAST DIGIT ISN'T A BLANK
3311	014756	000744			BR	25	,, GO DO THE LAST DIGIT
3312	014760	012605		65	MOV	(SP)+, R5	,, RESTORE R5
3313	014762	012604			MOV	(SP)+, R4	,, RESTORE R4
3314	014764	012603			MOV	(SP)+, R3	,, RESTORE R3
3315	014766	016666	000002 000004		MOV	2(SP), 4(SP)	,, SET THE STACK FOR RETURNING
3316	014774	012616			MOV	(SP)+, (SP)	
3317	014776	000002			RTI		,, RETURN
3318	015000	000		85	BYTE	0	,, STORAGE FOR ASCII DIGIT
3319	015001	000			BYTE	0	,, TERMINATOR FOR TYPE ROUTINE
3320	015002	000		%OCNT	BYTE	0	,, OCTAL DIGIT COUNTER
3321	015003	000		%OFILL	BYTE	0	,, ZERO FILL SWITCH
3322	015004	000000		%OMODE	WORD	0	,, NUMBER OF DIGITS TO TYPE
3323							, ENTER HERE ON POWER FAILURE
3324							
3325							
3326	015006				SPWRDN		
3327	015006	010046			PFAIL	MOL R0, -(SP)	, SAVE R0-R5 ON PROCESSOR STACK
3328	015010	010146			MOV	R1, -(SP)	
3329	015012	010246			MOV	R2, -(SP)	
3330	015014	010346			MOV	R3, -(SP)	
3331	015016	010446			MOV	R4, -(SP)	
3332	015020	010546			MOV	R5, -(SP)	
3333	015022	016746	162776		MOV	24, -(SP)	
3334	015026	010667	164064		MOV	SP, SAVSP	, SAVE STACK POINTER
3335	015032	012767	015044 162764		MOV	#RESTART, 24	, SET UP FOR POWER UP TRAP
3336	015040	000000			HALT		, HALT ON POWER DOWN NORMAL
3337	015042	000777			BR		
3338							
3339							, PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3340							
3341	015044	016706	164046		RESTAR	MOV SAVSP, SP	, RESTORE STACK POINTER
3342	015050	012605			MOV	(SP)+, R5	, RESTORE R0-R5
3343	015052	012604			MOV	(SP)+, R4	
3344	015054	012603			MOV	(SP)+, R3	
3345	015056	012602			MOV	(SP)+, R2	
3346	015060	012601			MOV	(SP)+, R1	
3347	015062	012600			MOV	(SP)+, R0	
3348	015064	012767	015006 162732		MOV	# PFAIL, 24	, SET UP FOR POWER FAILURE
3349	015072	106427	000340		MTPS	#340	
3350	015076	012706	001100		MOV	#STACK, SP	
3351	015102	005067	001102		CLR	TEMP	
3352	015106	005267	001076		INC	TEMP	
3353	015112	001375			BNE	-4	
3354	015114	104413			CONVRT		
3355	015116	015140			PFTAB		
3356	015120	104401			TYPE		
3357	015122	015442			MPFAIL		
3358	015124	005067	164253		CLR	%ERFLG	
3359	015130	005067	164262		CLR	%ERRPC	
3360	015134	000177	163744		JMP	@RETURN	

3361	015140	000001			PFTAB	1	
3362	015142	006	002			BYTE	6.2
3363	015144	000207				RETURN	
3364	015146	005015	042012	053125	MTITLE	ASCIZ	<15><12><12>/DUV11 OZDUT-B TAPE D /<15><12>
3365	015154	030461	042040	042132			
3366	015162	052125	041055	052040			
3367	015170	050101	020105	020104			
3368	015176	005015	000				
3369	015201	015	053012	041505	MVECTG	ASCIZ	<15><12>/VEC ADD-/
3370	015206	040440	042104	000055			
3371	015214	005015	051461	020124	MREGAD	ASCIZ	<15><12>/1ST DEV REC CSR ADD-/
3372	015222	042504	035126	051040			
3373	015230	041505	041440	051123			
3374	015236	040440	042104	000055			
3375	015244	005015	052515	052114	MMULT	ASCIZ	<15><12>/MULT DEV ? (Y OR N)-/
3376	015252	042040	053105	037440			
3377	015260	024040	020131	051117			
3378	015266	047040	026451	000			
3379	015273	015	046012	051501	MLASTD	ASCIZ	<15><12>/LAST DEV REC CSR ADDR-/
3380	015300	020124	042504	035126			
3381	015306	051040	041505	041440			
3382	015314	051123	040440	042104			
3383	015322	026522	000				
3384	015325	075	042504	044526	DEVICE	ASCIZ	/=DEVICE /
3385	015332	042503	020040	000			
3386	015337	015	051412	046105	MCOW	ASCIZ	<15><12>/SELECT TO RUN DACTREG/
3387	015344	041505	020124	047524			
3388	015352	051040	047125	040040			
3389	015360	041501	051124	043505			
3390	015366	000					
3391	015367	015	047412	043126	MRANGE	ASCIZ	<15><12>/OVFLO RETYPE LAST DEV RXCSR ADDS-/
3392	015374	047514	051072	052105			
3393	015402	050131	020105	040514			
3394	015410	052123	042040	053105			
3395	015416	051040	041530	051123			
3396	015424	040440	042104	026523			
3397	015432	000					
3398	015433	040	037440	000	MQM	ASCIZ	/ ?/
3399	015437	015	000012		MCRLF	ASCIZ	<15><12>
3400	015442	043120	044501	026114	MPFAIL	ASCIZ	/PFAIL, RESTART AT TEST IN PROGRESS/
3401	015450	020040	042522	052123			
3402	015456	051101	020124	052101			
3403	015464	052040	051505	020124			
3404	015472	047111	050040	047522			
3405	015500	051107	051505	000123			
3406	015506	005015	047105	020104	MEPASS	ASCIZ	<15><12>/END OF PASS TAPE D/
3407	015514	043117	050040	051501			
3408	015522	020123	040524	042520			
3409	015530	042040	000				
3410	015533	015	051012	000	MR	ASCIZ	<15><12>/R/
3411	015537	015	052012	051505	MTSTPC	ASCIZ	<15><12>/TEST PC-/
3412	015544	020124	041520	000055			
3413	015552	005015	047514	045503	MLOCK	ASCIZ	<15><12>/LOCK ON TEST? (Y OR N)-
3414	015560	047440	020116	052040			
3415	015566	051505	037524	024040			
3416	015574	020131	051117	047040			

3417	015602	026451	000		
3418	015605	015	021412	047440	MSYNC .ASCIZ <15><12>/# OF SYNC CHARS SELECTED ( 1 OR 2)-/
3419	015612	020106	054523	041516	
3420	015620	041440	040510	051522	
3421	015626	051440	046105	041505	
3422	015634	042524	020104	020050	
3423	015642	020061	051117	031040	
3424	015650	026451	000		
3425	015653	015	044412	020123	MWIRE6 ASCIZ <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3426	015660	042523	020103	046530	
3427	015666	052111	051440	044527	
3428	015674	041524	020110	032505	
3429	015702	026465	020062	047111	
3430	015710	020077	054450	047440	
3431	015716	020122	024516	000055	
3432	015724	005015	051511	051440	MWIRE5 ASCIZ <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3433	015732	041505	051040	041505	
3434	015740	051440	044527	041524	
3435	015746	020110	032505	026465	
3436	015754	020063	047111	020077	
3437	015762	054450	047440	020122	
3438	015770	024516	000055		
3439	015774	005015	051511	047440	MWIRE4 ASCIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3440	016002	052120	041440	051114	
3441	016010	042440	040516	046102	
3442	016016	020105	053523	052111	
3443	016024	044103	042440	032465	
3444	016032	030455	044440	037516	
3445	016040	024040	020131	051117	
3446	016046	047040	026451	000	
3447	016053	015	005012	031510	MEXTJ ASCIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3448	016060	032461	041440	047117	
3449	016066	042516	052103	051117	
3450	016074	047440	020116	024077	
3451	016102	020131	051117	047040	
3452	016110	026451	000		
3453	016113	015	020012	043536	MCNTG ASCIZ <15><12>/ G /
3454	016120	020040	000		
3455	016123	040	053523	036522	MMSWR ASCIZ / SWR= /
3456	016130	020040	000040		
3457	016134	020040	047040	053505	MMNEW ASCIZ / NEW= /
3458	016142	020075	000040		
3459					EVEN
3460					
3461					BUFFERS FOR INPUT-OUTPUT
3462					
3463	016146	000000			INBUF 0
3464		016210			= +40
3465	016210	000000			TEMP 0
3466		016252			= +40
3467	016252	000000			MDATA 0
3468		016314			= +40
3469					SBTTL SCOPE HANDLER ROUTINE
3470					
3471					*****
3472					*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS IT WILL INCREMENT

```

3473 , *AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG (DISPLAY<7 0>)
3474 , *AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15 08>
3475 , *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3476 , *SW14=1 LOOP ON TEST
3477 , *SW11=1 INHIBIT ITERATIONS
3478 , *SW09=1 LOOP ON ERROR
3479 , *SW08=1 LOOP ON TEST IN SWR<7 0>
3480 , *CALL
3481 , * SCOPE , ,SCOPE=10T
3482
3483 016314 SSCOPE
3484
3485 , SCOPE LOOP AND INTERATION HANDLER
3486
3487 SCOPE
3488 016314 004767 174376 JSR PC,CKSWR
3489 016320 005067 163072 CLR $ERRPC , CLEAR LAST ERROR PC
3490 016324 022716 003376 CMP #TST1+2, (SP) , IS SCOPE AT BEGINING OF TEST 1?
3491 016330 001422 BEQ $XTSTR , YES NO LOOP
3492
3493 016332 032777 040000 163100 TTST B T #BIT14, @SWR , THIS CODE IS FOR TESTING FOR BIT 14
3494 016340 001412 BEQ 1$ , ON LSI WHICH SYSMAC CANNOT HANDLE
3495 016342 016767 163034 163036 MOV $TSTNM, $LPADR
3496 016350 000406 BR 1$
3497 016352 105777 163066 TSTB @STKS , KEYBOARD DONE?
3498 016356 100123 BPL $OVER , BR IF NO
3499 016360 017766 163062 177776 MOV @STKB, -2(SP) , CLEAR DONE BIT
3500 016366 032777 040000 163044 1$ BIT #BIT14, @SWR , LOOP ON PRESENT TEST?
3501 016374 001114 BNE $OVER , YES IF SW14=1
3502 , #####START OF CODE FOR THE XOR TESTER#####
3503 016376 000416 SXTSTR BR 6$ , IF RUNNING ON THE "XOR" TESTER CHANGE
3504 , THIS INSTRUCTION TO A "NOP" (NOP=240)
3505 016400 013746 000004 MOV @#ERRVEC, -(SP) , SAVE THE CONTENTS OF THE ERROR VECTOR
3506 016404 012737 016424 000004 MOV #5$, @#ERRVEC , SET FOR TIMEOUT
3507 016412 005737 177060 TST @#177060 , TIME OUT ON XOR?
3508 016416 012637 000004 MOV (SP)+, @#ERRVEC , RESTORE THE ERROR VECTOR
3509 016422 000463 BR $SVLAD , GO TO THE NEXT TEST
3510 016424 022626 S$ CMP (SP)+, (SP)+ , CLEAR THE STACK AFTER A TIME OUT
3511 016426 012637 000004 MOV (SP)+, @#ERRVEC , RESTORE THE ERROR VECTOR
3512 016432 000423 BR 7$ , LOOP ON THE PRESENT TEST
3513 016434 6$ , #####END OF CODE FOR THE XOR TESTER#####
3514 016434 032777 000400 162776 BIT #BIT08, @SWR , LOOP ON SPEC TEST?
3515 016442 001404 BEQ 2$ , BR IF NO
3516 016444 127767 162770 162730 CMPB @SWR, $TSTNM , ON THE RIGHT TEST? SWR<7 0>
3517 016452 001465 BEQ $OVER , BR IF YES
3518 016454 105767 162723 2$ TSTB $ERFLG , HAS AN ERROR OCCURRED?
3519 016460 001421 BEQ 3$ , BR IF NO
3520 016462 126767 162727 162713 CMPB $ERMAX, $ERFLG , MAX ERRORS FOR THIS TEST OCCJRED?
3521 016470 101015 BHI 3$ , BR IF NO
3522 016472 032777 001000 162740 BIT #BIT09, @SWR , LOOP ON ERROR?
3523 016500 001404 BEQ 4$ , BR IF NO
3524 016502 016767 162702 162676 7$ MOV $LPERR, $LPADR , SET LOOP ADDRESS TO LAST SCOPE
3525 016510 000446 BR $OVER
3526 016512 105067 162665 4$ CLRB $ERFLG , ZERO THE ERROR FLAG
3527 016516 005067 162770 CLR $TIMES , CLEAR THE NUMBER OF ITERATIONS TO MAKE
3528 016522 000415 BR 1$ , ESCAPE TO THE NEXT TEST
  
```

```

3529 016524 032777 004000 162706 35 BIT #BIT11, @SWR // INHIBIT ITERATIONS?
3530 016532 001011 BNE 15 // BR IF YES
3531 016534 005767 162774 TST $PASS // IF FIRST PASS OF PROGRAM
3532 016540 001406 BEQ 15 // INHIBIT ITERATIONS
3533 016542 005267 162636 INC $ICNT // INCREMENT ITERATION COUNT
3534 016546 026767 162740 162630 CMP $TIMES, $ICNT // CHECK THE NUMBER OF ITERATIONS MADE
3535 016554 002024 BGE $OVER // BR IF MORE ITERATION REQUIRED
3536 016556 012767 000001 162620 15 MOV #1, $ICNT // REINITIALIZE THE ITERATION COUNTER
3537 016564 016767 000056 162720 MOV $MXCNT, $TIMES // SET NUMBER OF ITERATIONS TO DO
3538 016572 105267 162604 $SVLAD INCB $STNM // COUNT TEST NUMBERS
3539 016576 116767 162600 162726 MOV $STNM, $TESTN // SET TEST NUMBER IN APT MAILBOX
3540 016604 011667 162576 MOV (SP), $LPADR // SAVE SCOPE LOOP ADDRESS
3541 016610 011667 162574 MOV (SP), $LPERR // SAVE ERROR LOOP ADDRESS
3542 016614 005067 162674 CLR $ESCAPE // CLEAR THE ESCAPE FROM ERROR ADDRESS
3543 016620 112767 000001 162567 MOV #1, $ERMAX // ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3544 016626 016777 162550 162606 $OVER MOV $STNM, @DISPLAY // DISPLAY TEST NUMBER
3545 016634 016716 162546 MOV $LPADR, (SP) // FUDGE RETURN ADDRESS
3546 016640 000002 45 RTI
3547 016642 001407 BRW 1407
3548 016644 000432 BRX 432
3549 016646 C00005 $MXCNT 5 // MAX NUMBER OF ITERATIONS
3550 SBTTL TRAP DECODER
3551
3552 // *****
3553 // *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3554 // *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3555 // *OF THE DESIRED ROUTINE THEN USING THE ADDRESS OBTAINED IT WILL
3556 // *GO TO THAT ROUTINE
3557
3558 016650 010046 STRAP MOV RO, -(SP) // SAVE RO
3559 016652 016600 000002 MOV 2(SP), RO // GET TRAP ADDRESS
3560 016656 005740 TST -(RO) // BACKUP BY 2
3561 016660 111000 MOV (RO), RO // GET RIGHT BYTE OF TRAP
3562 016662 006300 ASL RO // POSITION FOR INDEXING
3563 016664 016000 016704 MOV $TRPAD(RO), RO // INDEX TO TABLE
3564 016670 000200 RTS RO // GO TO ROUTINE
3565
3566
3567 // THIS IS USE TO HANDLE THE "GETPRI" MACRO
3568
3569 016672 011646 STRAP2 MOV (SP), -(SP) // MOVE THE PC DOWN
3570 016674 016666 000004 000002 MOV 4(SP), 2(SP) // MOVE THE PSW DOWN
3571 016702 000002 RTI // RESTORE THE PSW
3572
3573 SBTTL TRAP TABLE
3574
3575 // *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3576 // *BY THE "TRAP" INSTRUCTION
3577
3578 // ROUTINE
3579 // -----
3580 016704 016672 STRPAD .WORD STRAP2
3581 016706 013066 $TYPE // CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
3582 016710 014604 $TYPOC // CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3583 016712 014560 $TYPOS // CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
3584 016714 014620 $TYPON // CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

```

```
3585
3586
3587 016716 013044          SCOP1  ;,CALL=SCOP1    TRAP+5(104405)
3588 016720 013350          INSTR  ;,CALL=INSTR     TRAP+6(104406)
3589 016722 013456          INSTER ;,CALL=INSTER    TRAP+7(104407)
3590 016724 013466          PARAM  ;,CALL=PARAM     TRAP+10(104410)
3591 016726 013666          SAV05  ;,CALL=SAV05    TRAP+11(104411)
3592 016730 013726          RES05  ;,CALL=RES05    TRAP+12(104412)
3593 016732 013760          CONVRT ;,CALL=CONVRT    TRAP+13(104413)
3594 016734 014136          SETFLG ;,CALL=SETFLG    TRAP+14(104414)
3595
3596
3597
3598
3599
3600 016736 006367 000044          DULEV  ASL      DUPRT  ,SHIFT LEFT
3601 016742 006367 000040          ASL      DUPRT
3602 016746 006367 000034          ASL      DUPRT
3603 016752 006367 000030          ASL      DUPRT
3604 016756 006367 000024          ASL      DUPRT
3605 016762 016767 000020 000020          MOV      DUPRT,LESS1  ,MOVE THIS TO LESS1
3606 016770 162767 000001 000012          SUB      #1,LESS1    ,CREATE LESS1
3607 016776 042767 000037 000004          BIC      #37,LESS1  ,CLEAR TNZVC
3608 017004 000207          RTS      PC
3609 017006 000240          DUPRT   PR5
3610 017010 000200          LESS1  PR4  ,LEVEL TO ALLOW INTERPUPTS
3611
3612
3613 017012 016767 000126 162672          DUAADR  MOV      DUBASE,RXCSR  ,XXX0
3614 017020 005267 000120          INC      DUBASE
3615 017024 016767 000114 162662          MOV      DUBASE,HRXCSR  ,XXX1
3616 017032 005267 000106          INC      DUBASE
3617 017036 016767 000102 162652          MOV      DUBASE,RXDDBUF ,XXX2
3618 017044 016767 000074 162650          MOV      DUBASE,PARCSR  ,XXX2
3619 017052 005267 000066          INC      DUBASE
3620 017056 016767 000062 162634          MOV      DUBASE,HRXDDBUF,XXX3
3621 017064 016767 000054 162632          MOV      DUBASE,HPARCSR ,XXX3
3622 017072 005267 000046          INC      DUBASE
3623 017076 016767 000042 162622          MOV      DUBASE,TXCSR   ,XXX4
3624 017104 005267 000034          INC      DUBASE
3625 017110 016767 000030 162612          MOV      DUBASE,HTXCSR  ,XXX5
3626 017116 005267 000022          INC      DUBASE
3627 017122 016767 000016 162602          MOV      DUBASE,TXDDBUF,XXX6
3628 017130 005267 000010          INC      DUBASE
3629 017134 016767 000004 162572          MOV      DUBASE,HTXDDBUF,XXX7
3630 017142 000207          RTS      PC
3631 017144 000000          DUBASE  0
3632
3633
3634
3635
3636 017146 042777 040000 162552          PPOKE  BIC      #MTDATA,@TXCSR
3637 017154 005067 162322          CLR      $TMP2
3638 017160 006067 162314          ROR      $TMP1  ,FORCE CARRY
3639 017164 006067 162312          ROR      $TMP2  ,PICK UP CARRY IN B T 15
3640 017170 006267 162306          ASR      $TMP2  ,SHIFT INTO BIT 14
```

```

3641 017174 042767 100000 162300      BIC  #BIT15,STMP2  ,CLR BIT 15
3642 017202 056777 162274 162516      BIS  STMP2,@TXCSR  ,POKE MAINT DATA
3643 017210 042777 020000 162510      BIC  #CLK,@TXCSR  ,POKE CLK
3644 017216 052777 020000 162502      BIS  #CLK,@TXCSR  ,
3645 017224 005367 161672      DEC  SHIFT
3646 017230 001346      BNE  RPOKE
3647 017232 000207      RTS  PC
3648
3649
3650 017234 016767 162240 162240  ODD8  , THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3651 017242 005067 162236      MOV  STMP1,STMP2  ,SAVE TEMP1
3652 017246 012727 000010      CLR  STMP3
3653 017252 000000      MOV  #8,(PC)+
3654 017254 006067 162222      4$  0
3655 017260 005567 162220      1$  ROR  STMP2
3656 017264 005367 177762      ADC  STMP3
3657 017270 001371      DEC  4$
3658 017272 006067 162206      BNE  1$
3659 017276 103404      ROR  STMP3
3660 017300 052767 000400 162172      BCS  2$
3661 017306 000403      BIS  #BIT8,STMP1  ,SET ODD PARITY
3662 017310 042767 000400 162162  2$  BR   3$
3663      BIC  #BIT8,STMP1  ,CLR EVEN PARITY
3664 017316 000207      ,STMP1 NOW HAS ODD PARITY CHARACTER
3665      RTS  PC
3666
3667 017320 016767 162154 162154  EVEN8  , THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3668 017326 005067 162152      MOV  STMP1,STMP2  ,SAVE TEMP1
3669 017332 012727 000010      CLR  STMP3
3670 017336 000000      MOV  #8,(PC)+
3671 017340 006067 162136      4$  0
3672 017344 005567 162134      1$  ROR  STMP2
3673 017350 005367 177762      ADC  STMP3
3674 017354 001371      DEC  4$
3675 017356 006067 162122      BNE  1$
3676 017362 103004      ROR  STMP3
3677 017364 052767 000400 162106      BCC  2$
3678 017372 000403      BIS  #BIT8,STMP1  ,SET EVEN PARITY
3679 017374 042767 000400 162076  2$  BR   3$
3680      BIC  #BIT8,STMP1  ,CLR ODD PARITY
3681 017402 000207      ,STMP1 NOW HAS EVEN PARITY CHARACTER
3682 017404 062716 000002      3$  RTS  PC
3683      TRPREG ADD #2,(SP) ,ALLOW IT TO "CRUNCH" INTO HLT BACK
3684 017410 000002      , IN MAIN PART OF THE PROGRAM
3685      RTI
      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*	2796	2799	2809	
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=	013665	3011*	3050*	3057*						
ADENV =	000000	877	888							
ADENV =	000000	877	889							
ADFATAL=	000000	877	880							
ADADR1=	000000	877	905							
ADADR2=	000000	877	909							
ADADR3=	000000	877	912							
ADADR4=	000000	877	915							
ADAMS1=	000000	877	899							
ADAMS2=	000000	877	907							
ADAMS3=	000000	877	910							
ADAMS4=	000000	877	913							
ADMSGAD=	000000	877	885							
ADMSGLG=	000000	877	886							
ADMSGTY=	000000	877	879							
ADMTYP1=	000000	877	900							
ADMTYP2=	000000	877	908							
ADMTYP3=	000000	877	911							
ADMTYP4=	000000	877	914							
ADPASS =	000000	877	882							
ADPRIOR=	000000	877								
ADPTCSU=	000040	537*	2926							
ADPTENV=	000001	537*	2919	3173						
ADPTSIZ=	000200	537*	1169							
ADPTSPO=	000100	537*	2921							
ADSHREG=	000000	877	890							
ADTESTN=	000000	877	881							
ADUNIT =	000000	877	884							
ADUSHR =	000000	877	891							
ADVECT1=	000000	877	916							
ADVECT2=	000000	877	917							
ADBASED	001154	731*	1232*	1269*	1270	1276*	1278*	2803*	2815*	2819











CROSS REFERENCE TABLE -- USER SYMBOLS

SW15 = 100000	613#													
SW2 = 000004	636#													
SW3 = 000010	635#													
SW4 = 000020	634#													
SW5 = 000040	633#													
SW6 = 000100	632#													
SW7 = 000200	631#													
SW8 = 000400	630#													
SW9 = 001000	629#													
SYNCNO 001146	722#	1299*	1303*											
SYNEXT= 020000	790#	983#	2017	2025	2078	2086	2139	2147	2200	2208				
SYNINT= 030000	789#	982#	1355	1362	1773	1781	1834	1842	1895	1903	1956	1964	2258	
	2266	2311	2319	2471	2479	2524	2532	2685	2693	2738	2746			
SYNSCH= 000020	775#	968#	1363	1411	1504	1600	1696							
SYSTST= 014000	815#	1008#												
TBITVE= 000014	671#													
TEMP 016210	3096	3351*	3352*	3465#										
TKVEC = 000060	678#													
TPVEC = 000064	679#													
TRAPVE= 000034	677#	1143*	1144*											
TRPREG 017404	3682#													
TRTVEC= 000014	672#													
TST1 003374	1334	1341	1353#	2851	3490									
TST10 006130	1893#													
TST11 006346	1954#													
TST12 006564	2015#													
TST13 007002	2076#													
TST14 007220	2137#													
TST15 007436	2198#													
TST16 007654	2256#													
TST17 010062	2309#													
TST2 003544	1399#													
TST20 010270	2362#													
TST21 010476	2415#													
TST22 010704	2469#													
TST23 011112	2522#													
TST24 011320	2575#													
TST25 011526	2628#													
TST26 011734	2683#													
TST27 012142	2736#													
TST3 004132	1492#													
TST4 004520	1588#													
TST5 005106	1684#													
TST6 005474	1771#													
TST7 005712	1832#													
TTST 016332	3493#													
TXCSR 001726	1049#	1354*	1356*	1359*	1365*	1366*	1368*	1369*	1400*	1402*	1405*	1413*	1414*	
	1416*	1417*	1493*	1495*	1498*	1506*	1507*	1509*	1510*	1589*	1591*	1594*	1602*	
	1603*	1605*	1606*	1685*	1687*	1690*	1698*	1699*	1701*	1702*	1772*	1774*	1778*	
	1782	1787*	1788*	1794*	1795*	1796	1805*	1807	1814	1833*	1835*	1839*	1843	
	1848*	1849*	1855*	1856*	1857	1866*	1868	1875	1894*	1896*	1900*	1904	1909*	
	1910*	1916*	1917*	1918	1927*	1929	1936	1955*	1957*	1961*	1965	1970*	1971*	
	1977*	1978*	1979	1988*	1990	1997	2016*	2018*	2022*	2026	2031*	2032*	2038*	
	2039*	2040	2049*	2051	2058	2077*	2079*	2083*	2087	2092*	2093*	2099*	2100*	
	2101	2110*	2112	2119	2138*	2140*	2144*	2148	2153*	2154*	2160*	2161*	2162	
	2171*	2173	2180	2199*	2201*	2205*	2209	2214*	2215*	2221*	2222*	2223	2232*	















SCMTA	526#	816
SEOP	526#	
SEPRO	526#	3143
SERRT	526#	3199
SPOWE	526#	
SSCOP	526#	3469
STRAP	526#	3550
STYPE	526#	2896
STYPO	526#	3246

ABS 017412 000

ERRORS DETECTED 0

DZDUTB.DZDUTB/SOL/CRF=DZDUT1/EQ RUND.DZDUT2.DZDUTB  
RUN-TIME 21 12 1 SECONDS  
RUN-TIME RATIO 255/35=7 1  
CORE USED 30K (59 PAGES)