

DUV-11

OFFLINE RECEIVER TIMING TEST
MD-11-DZDUS-B

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

DEC 1977
digital
MADE IN USA

The microfiche card contains a grid of 48 frames of technical data, arranged in 8 rows and 6 columns. Each frame appears to contain a small table or diagram with alphanumeric characters. The data is too small to be legible. The right half of the card is blank.

REM *

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

PRODUCT CODE MAINDEC-11-DZDUS-B-D

PRODUCT NAME DUV11 OFFLINE RECEIVER TIMING 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 ERROPS 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 RECEIVER TIMING TESTS VERIFY THAT THE
RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT
AT THE PROPPER TIME

* REM *

2 REQUIREMENTS

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

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

* REM *

3 LOADING PROCEDURE

THE STANDARD PROCEDURE FOR LOADING ABSOLUTE BINARY TAPES
IS TO BE USED

STARTING ADDRESS
FOR ABSOLUTE LOADER

4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4 STARTING PROCEDURE

4 1 CONTROL SWITCH SETTINGS

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

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

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 DZDUS-B TAPE C" (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 TH'S 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 JIA 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"

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

000001

STN=1

558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613

001100

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

000000
000040
000100
000140
000200
000240
000300
000340

100000
040000

```
.ENABLE ABS

,DUV11 DZDUS-B TAPE C
,COPYRIGHT 1977, DIGITAL EQUIPMENT CORP , MAYNARD, MASS 01754

, STARTING PROCEDURE
, TYPE 200G
, PROGRAM WILL TYPE "DUV11 DZDUS-B TAPE C "
, PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
, AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE C"
, AND THEN RESUME TESTING

SBTTL BASIC DEFINITIONS

,*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
EQUIV EMT,ERROR      ,,BASIC DEFINITION OF ERROR CALL
EQUIV IGT,SCOPE      ,,BASIC DEFINITION OF SCOPE CALL

,*MISCELLANEOUS DEFINITIONS
HT= 11              ,,CODE FOR HORIZONTAL TAB
LF= 12              ,,CODE FOR LINE FEED
CR= 15              ,,CODE FOR CARRIAGE RETURN
CRLF= 200           ,,CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776         ,,PROCESSOR STATUS WORD
EQUIV PS,PSW
STKLMT= 177774     ,,STACK LIMIT REGISTER
PIRQ= 177772       ,,PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570       ,,HARDWARE SWITCH REGISTER
ODISP= 177570     ,,HARDWARE DISPLAY REGISTER

,*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0             ,,GENERAL REGISTER
R1= %1             ,,GENERAL REGISTER
R2= %2             ,,GENERAL REGISTER
R3= %3             ,,GENERAL REGISTER
R4= %4             ,,GENERAL REGISTER
R5= %5             ,,GENERAL REGISTER
R6= %6             ,,GENERAL REGISTER
R7= %7             ,,GENERAL REGISTER
SP= %6             ,,STACK POINTER
PC= %7             ,,PROGRAM COUNTER

,*PRIORITY LEVEL DEFINITIONS
PRO= 0             ,,PRIORITY LEVEL 0
PR1= 40            ,,PRIORITY LEVEL 1
PR2= 100           ,,PRIORITY LEVEL 2
PR3= 140           ,,PRIORITY LEVEL 3
PR4= 200           ,,PRIORITY LEVEL 4
PR5= 240           ,,PRIORITY LEVEL 5
PR6= 300           ,,PRIORITY LEVEL 6
PR7= 340           ,,PRIORITY LEVEL 7

,*"SWITCH REGISTER" SWITCH DEFINITIONS
SW15= 100000
SW14= 40000
```

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

, *DATA BIT DEFINITIONS (BIT00 TO BIT15)

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

, *BASIC "CPU" TRAP VECTOR ADDRESSES

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

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

. STANDARD INTERRUPT VECTORS

680						
681						
682						
683		000174		=174		
684	000174	000000		DISPREG 0		
685	000176	000000		SWREG 0		
686		000200		=200		
687	000200	000167	001746	JMP	START	. GO TO START OF PROGRAM
688						
689						
690						
691		001100		. =1100		
692	001100	000000		WORD 0		
693	001102	177570		LIGHTS 177570		
694						
695						
696						
697						

. PROGRAM CONTROL PARAMETERS

698						
699	001104	000000		RETURN	0	
700	001106	000000		NEXT	0	. ADDRESS OF NEXT TEST TO BE EXECUTED
701	001110	000000		LOCK	0	. ADDRESS FOR LOCK ON CURRENT DATA
702	001112	000000		PASCNT	0	. ADDRESS CONTAINING PASS COUNT
703	001114	000000		ERRCNT	0	. ERROR COUNT
704	001116	000000		SAVSP	0	. STACK POINTER STORAGE
705						

. PROGRAM VARIABLES

706						
707						
708	001120	000020		HOLD	20	. TEMPORARY STORAGE=DELAY TIME FOR CABLES
709	001122	000000		SHIFT	0	. TEMPORARY STORAGE= # OF SHIFTS PER CHAR
710	001124	000000		COUNT	0	. TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
711	001126	000000		SAVPC	0	. PROGRAM COUNTER STORAGE
712	001130	000000		HLD0	0	
713	001132	000000		HLD1	0	
714	001134	000000		HLD2	0	
715	001136	000000		HLD3	0	
716	001140	000000		HLD4	0	
717	001142	000000		HLD5	0	
718	001144	000000		HLD6	0	
719						


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

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

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

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

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


```

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

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

1008 SBTTL ERROR POINTER TABLE
 1009
 1010 ,*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
 1011 ,*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
 1012 ,*LOCATION \$ITEMB THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT
 1013 ,*NOTE1 IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC)
 1014 ,*NOTE2 EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS
 1015
 1016 ,* EM ;, POINTS TO THE ERROR MESSAGE
 1017 ,* DH ;, POINTS TO THE DATA HEADER
 1018 ,* DT ;, POINTS TO THE DATA
 1019 ,* DF ;, POINTS TO THE DATA FORMAT
 1020

1021
 1022 001652 SERRTB
 1023 , ERROR TABLE
 1024 001652 001762 EM1 , ERROR 1 REGISTER ERROR
 1025 001654 002067 DH1
 1026 001656 002116 DT1
 1027 001660 002132 DF1
 1028 001662 002022 EM2 , ERROR 2 RECEIVER ERROR
 1029 001664 002067 DH1
 1030 001666 002116 DT1
 1031 001670 002132 DF1
 1032 001672 002043 EM3 , ERROR 3 TRANSMITTER ERROR
 1033 001674 002067 DH1
 1034 001676 002116 DT1
 1035 001700 002132 DF1
 1036 001702 001746 EM4 , ERROR 4 BIT ERROR (GENERAL)
 1037 001704 000000 0
 1038 001706 002126 DT4
 1039 001710 002132 DF1
 1040

1041 , DEFAULT DU ADDRESSES
 1042 001712 160010 RXCSR 160010
 1043 001714 160011 HRXCSR 160011
 1044 001716 160012 RXDBUF 160012
 1045 001720 160013 HRXDBUF 160013
 1046 001722 160012 PARCSR 160012
 1047 001724 160013 HPARCSR 160013
 1048 001726 160014 TXCSR 160014
 1049 001730 160015 HTXCSR 160015
 1050 001732 160016 TXDBUF 160016
 1051 001734 160017 HTXDBUF 160017

1052 , DEFAULT DU VECTORS
 1053 001736 000770 DURIV 770 ;REC INTR VECTOR
 1054 001740 000772 DURIS 772 ;REC INTR STATUS
 1055 001742 000774 DUTIV 774 ,XMIT INTR VECTOR
 1056 001744 000776 DUTIS 776 ,XMIT INTR STATUS

1057 , ERROR MESSAGES
 1058 001746 020040 051105 047522 EM4 : ASCIZ / ERROR PC /
 1059 001754 020122 041520 000040
 1060 001762 020040 047503 050115 EM1 : ASCIZ / COMPARISON ERROR ON REGISTERS/
 1061 001770 051101 051511 047117
 1062 001776 042440 051122 051117
 1063 002004 047440 020116 042522

1064	002012	044507	052123	051105		
1065	002020	000123				
1066	002022	020040	042522	042503	EM2.	ASCIZ / RECEIVER ERROR/
1067	002030	053111	051105	042440		
1068	002036	051122	051117	000		
1069	002043	040	052040	040522	EM3	ASCIZ / TRANSMITTER ERROR/
1070	002050	051516	044515	052124		
1071	002056	051105	042440	051122		
1072	002064	051117	000			
1073						, DATA HEADERS FOR ERROR MESSAGES
1074	002067	105	051122	041520	DM1	ASCIZ /ERRPC WANTED ACTUAL/
1075	002074	020040	040527	052116		
1076	002102	042105	020040	041501		
1077	002110	052524	046101	000		
1078		002116			EVEN	
1079						, DATA TABLES FOR ERROR MESSAGES
1080	002116	001416	001130	001132	DT1	WORD \$ERRPC,HLDD,HLDD1,0
1081	002124	000000				
1082						
1083	002126	001416	000000		DT4	WORD \$ERRPC,0
1084						
1085	002132	000	000	000	DF1	BYTE 0,0,0,0
1086	002135	000				
1087					EVEN	
1088					SBTTL	ACT11 HOOKS
1089						
1090						*****
1091						,HOOKS REQUIRED BY ACT11
1092		002136				\$SVPC= ,SAVE PC
1093		000046				=46
1094	000046	012660				\$ENDAD ,,1)SET LOC 46 TO ADDRESS OF SENDAD IN \$EOP
1095		000052				=52
1096	000052	000000				WORD 0 ,,2)SET LOC 52 TO ZERO
1097		002136				=\$SVPC ,,RESTORE PC
1098					SBTTL	APT PARAMETER BLOCK
1099						
1100						*****
1101						,SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1102						*****
1103		002136				\$X= ;;SAVE CURRENT LOCATION
1104		000024				=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1105	000024	000200				200 ;;FOR APT START UP
1106		000044				=44 ;;POINT TO APT INDIRECT ADDRESS PNTR
1107	000044	002136				\$APTHDR ;;POINT TO APT HEADER BLOCK
1108		002136				= \$X ;;RESET LOCATION COUNTER
1109						*****
1110						,SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1111						,INTERFACE SPEC
1112						
1113	002136				\$APTHD	
1114	002136	000000			\$HIBTS	WORD 0 ,,TWO HIGH BITS OF 18 BIT MAILBOX ADDR
1115	002140	001526			\$MADR:	WORD \$MAIL ,,ADDRESS OF APT MAILBOX (BITS 0-15)
1116	002142	000010			\$TSTM.	WORD 10 ,,RUN TIM OF LONGEST TEST
1117	002144	000010			\$PASTM.	WORD 10 ,,RUN TIME IN SECS OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1118	002146	000000			\$UNITM	WORD ,,ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1119	002150	000052				SETEND-\$MAIL/2 ,,LENGTH MAILBOX-ETABLE(WORDS)

```

1120
1121
1122          , PROGRAM INITIALIZATION
1123          , LOCK OUT INTERRUPTS
1124          , SET UP PROCESSOR STACK
1125          , SET UP POWER FAIL VECTOR
1126          , CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1127          , TYPE TITLE MESSAGE
1128
1129 002152      START:
1130          SBTTL INITIALIZE THE COMMON TAGS
1131          ;; CLEAR THE COMMON TAGS (%$CMTAG) AREA
1132 002152 012706 001400      MOV    #%$CMTAG, R6      ;; FIRST LOCATION TO BE CLEARED
1133 002156 005026          CLR    (R6)+          ;; CLEAR MEMORY LOCATION
1134 002160 022706 001440      CMP    #SWR, R6      ;; DONE?
1135 002164 001374          BNE    -6              ;; LOOP BACK IF NO
1136 002166 012706 001100      MOV    ##STACK, SP      ;; SETUP THE STACK POINTER
1137          ;; INITIALIZE A FEW VECTORS
1138 002172 012737 016304 000020      MOV    %%SCOPE, @#IOTVEC  ;; IOT VECTOR FOR SCOPE ROUTINE
1139 002200 012737 000340 000022      MOV    #340, @#IOTVEC+2  ;; LEVEL 7
1140 002206 012737 014174 000030      MOV    #ERROR, @#EMTVEC  ;; EMT VECTOR FOR ERROR ROUTINE
1141 002214 012737 000340 000032      MOV    #340, @#EMTVEC+2  ;; LEVEL 7
1142 002222 012737 016640 000034      MOV    #STRAP, @#TRAPVEC ;; TRAP VECTOR FOR TRAP CALLS
1143 002230 012737 000340 000036      MOV    #340, @#TRAPVEC+2 ;; LEVEL 7
1144 002236 012737 014776 000024      MOV    #SPWRDN, @#PWRVEC ;; POWER FAILURE VECTOR
1145 002244 012737 000340 000026      MOV    #340, @#PWRVEC+2  ;; LEVEL 7
1146 002252 005067 177234          CLR    $TIMES          ;; INITIALIZE NUMBER OF ITERATIONS
1147 002256 005067 177232          CLR    $ESCAPE        ;; CLEAR THE ESCAPE ON ERROR ADDRESS
1148 002262 112767 000001 177125      MOV    #1, $ERMAX      ;; ALLOW ONE ERROR PER TEST
1149 002270 012767 002270 177110      MOV    #. , $LPADR     ;; INITIALIZE THE LOOP ADDRESS FOR SCOPE
1150 002276 012767 002276 177104      MOV    #. , $LPERR     ;; SETUP THE ERROR LOOP ADDRESS
1151          ;; SIZE FOR A HARDWARE SWITCH REGISTER IF NOT FOUND OR IT IS
1152          ;; EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER
1153 002304 013746 000004          MOV    @#ERRVEC, -(SP)  ;; SAVE ERROR VECTOR
1154 002310 012737 002344 000004      MOV    #64$, @#ERRVEC  ;; SET UP ERROR VECTOR
1155 002316 012767 177570 177114      MOV    #DSWR, SWR      ;; SETUP FOR A HARDWARE SWICH REGISTER
1156 002324 012767 177570 177110      MOV    #DDISP, DISPLAY ;; AND A HARDWARE DISPLAY REGISTER
1157 002332 022777 177777 177100      CMP    #-1, @SWR      ;; TRY TO REFERENCE HARDWARE SWR
1158 002340 001012          BNE    66$           ;; BRANCH IF NO TIMEOUT TRAP OCCURRED
1159          ;; AND THE HARDWARE SWR IS NOT = -1
1160 002342 000403          BR    65$           ;; BRANCH IF NO TIMEOUT
1161 002344 012716 002352          64$ MOV    #65$, (SP)    ;; SET UP FOR TRAP RETURN
1162 002350 000002          RTI
1163 002352 012767 000176 177060          65$ MOV    #SWREG, SWR    ;; POINT TO SOFTWARE SWR
1164 002360 012767 000174 177054          MOV    #DISPREG, DISPLAY
1165 002366 012637 000004          66$ MOV    (SP)+, @#ERRVEC ;; RESTORE ERROR VECTOR
1166
1167 002372 005067 177136          CLR    $PASS          ;; CLEAR PASS COUNT
1168 002376 132767 000200 177143      BITB  #APTSIZE, $ENVM  ;; TEST USER SIZE UNDER APT
1169 002404 001403          BEQ    67$           ;; YES, USE NON-APT SWITCH
1170 002406 012767 001550 177024          MOV    #$$SWREG, SWR  ;; NO, USE APT SWITCH REGISTER
1171 002414          67$
1172 002414 012706 001100          MOV    #STACK, SP    ;; SET STACK
1173 002420 106427 000340          MTPS  #340          ;; LOCK INTERPUPTS
1174 002424 012737 014776 000024      MOV    #. PFAIL, @#24  ;; SET UP POWER FAIL VECTOR
1175 002432 105067 176535          CLR    $STFLG        ;; CLEAR START FLAG
  
```

1176	002436	005067	176450		CLR	PASCNT	, CLEAR PASS COUNT
1177	002442	105067	176735		CLRB	SERFLG	, CLEAR ERROR FLAG
1178	002446	005067	176740		CLR	SERTTL	, CLEAR ERROR COUNT
1179	002452	005067	176740		CLR	SERRPC	, CLEAR LAST ERROR POINTER
1180	002456	012767	000001	176716	MOV	#1, STSTNM	, SET UP FOR TEST 1
1181	002464	012767	002152	176412	MOV	#. START, RETURN	, SET UP FOR POWER FAIL BEFORE TESTING STARTS
1182							
1183	002472	013746	000006		MOV	@#6, -(SP)	
1184	002476	013746	000004		MOV	@#4, -(SP)	
1185	002502	012737	002516	000004	MOV	#15, @#4	
1186	002510	005777	176724		TST	@SWR	
1187	002514	000407			BR	2\$	
1188	002516	012767	000176	176714	1\$	MOV	#SWREG, SWR
1189	002524	012767	000174	176710	MOV	#DISPREG, DISPLAY	
1190	002532	022626			CMP	(SP)+, (SP)+	
1191	002534	012637	000004		2\$	MOV	(SP)+, @#4
1192	002540	012637	000006		MOV	(SP)+, @#6	
1193	002544	022767	000176	176666	CMP	#SWREG, SWR	
1194	002552	001007			BNE	3\$	
1195	002554	005737	000042		TST	@#42	, CHECK FOR CHAIN
1196	002560	001402			BEQ	33\$	
1197	002562	000167	000522		JMP	BEGIN	
1198	002566	004767	010170		33\$	JSR	PC, CNTLU
1199	002572	105767	176374		3\$	TSTB	INIFLG
1200	002576	001004			BNE	ONCE	, HAS INITIALIZATION BEEN PERFORMED
1201	002600	104401	01E136		TYPE	, MTITLE	, TYPE TITLE MESSAGE
1202	002604	105167	176362		COMB	INIFLG	, IF NOT SET FLAG AND DO
1203	002610	105767	176732		ONCE	TSTB	SENV
1204	002614	001410			BEQ	11\$, APT CONTROL?
1205	002616	032767	000001	176726	BIT	#1, \$USWR	, BR IF NO
1206	002624	001002			BNE	12\$, EXTENAL JUMPER ON?
1207	002626	105067	176321		CLRB	JMRBY	, NO
1208	002632	000167	000452		12\$	JMP	BEGIN
1209	002636	032777	000001	176574	11\$	BIT	#SW00, @SWR
1210	002644	001002			BNE	1\$, RESELECT VECTOR & CONTROL REG?
1211	002646	000167	000436		JMP	BEGIN	
1212	002652	012700	000300		1\$	MOV	#300, RD , RESTORE VECTOR AREA TO TRAPCATCHER
1213	002656	012701	000302		MOV	#302, R1	, START AT LOCATION 300
1214	002662	012702	000004		MOV	#4, R2	
1215	002666	010110			2\$	MOV	R1, (RD)
1216	002670	005011			CLR	(R1)	
1217	002672	060200			ADD	R2, RD	
1218	002674	060201			ADD	R2, R1	
1219	002676	022701	001000		CMP	#1000, R1	, END AT LOCATION 776
1220	002702	002771			BLT	2\$	
1221	002704	104406			INSTR		, OUTPUT MESSAGE & GET INPUT STRING
1222	002706	015204			MREGAD		, MESSAGE
1223	002710	104410			PARAM		, CONVERT STRING
1224	002712	160000			160000		, LOW LIMIT
1225	002714	167776			167776		, HIGH LIMIT
1226	002716	017134			DUBASE		, STORE AT THIS LOCATION
1227	002720	001			1		, MASK
1228	002721	001			1		, HOW MANY TIMES + 2
1229	002722	016767	014206	176226	MOV	DUBASE, KEEPADD	, SAVE
1230	002730	004767	014046		JSR	PC, DUADDR	
1231	002734	016767	176216	176212	MOV	KEEPADD, BASEADD	, RESTORE FOR ROTATION

1232	002742	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1233	002744	015171				MVECTO	; MESSAGE
1234	002746	104410				PARAM	; CONVERT STRING
1235	002750	000300				300	; LOW LIMIT
1236	002752	000776				776	; HIGH LIMIT
1237	002754	001736				DURIV	; STORE AT THIS LOCATION
1238	002756	001			BYTE	1	; MASK
1239	002757	004			BYTE	4	; HOW MANY TIMES + 2
1240	002760	016767	176752	176176		MOV	DURIV,KEEPIV ; SAVE
1241	00276	016767	176744	176166		MOV	DURIV,BASEIV ; SET UP FOR ROTATION
1242	002774	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1243	002776	015234				MMULT	; MESSAGE
1244	003000	104414				SETFLG	; SET FLAG BASED UPON INPUT STRING
1245	003002	001152				MULTD	; THIS FLAG
1246	003004	105767	176142			TSTB	MULTD ; ARE THERE MULTIPLE DEVICES
1247							; ON THE SYSTEM ?
1248	003010	100406				BMI	BBB ; YES, ASK NEXT QUESTION
1249	003012	005067	176150			CLR	ACTREG
1250	003016	005067	176146			CLR	ROTADD
1251	003022	000167	000140			JMP	OUTMUL ; JUMP AROUND NEXT QUESTION
1252	003026				BBB		
1253	003026	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1254	003030	015263				MLASTD	; MESSAGE
1255	003032	104410				PARAM	; CONVERT STRING
1256	003034	160000				160000	; LOW LIMIT
1257	003036	167776				167776	; HIGH LIMIT
1258	003040	001160				LASTADD	; STORE AT THIS LOCATION
1259	003042	001			BYTE	1	; MASK
1260	003043	001			BYTE	1	; HOW MANY TIMES + 2
1261							; THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1262	003044	012767	000001	176116	1\$	MOV	#1,ROTADD ; SET UP POINTER
1263	003052	005067	176110			CLR	ACTREG ; CLR ACTIVE REGISTER
1264	003056	056767	176106	176102	2\$	BIS	ROTADD,ACTREG ; MAKE THIS DEVICE ACTIVE
1265	003064	000241				CLC	
1266	003066	006167	176076			ROL	ROTADD ; SET UP POINTER
1267	003072	103421				BCS	3\$; ARE YOU OUT OF RANGE ?
1268	003074	062767	000010	176052		ADD	#10,BASEADD ; SET UP BASE ADDRESS
1269	003102	026767	176052	176044		CMP	LASTADD,BASEADD ; IS THIS THE LAST DEVICE ?
1270	003110	101362				BHI	2\$; NO DO IT AGAIN
1271	003112	056767	176052	176046		BIS	ROTADD,ACTREG ; THIS ASSUMES THAT THERE ARE AT
1272							; LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1273							; MULTIPLE DEVICE QUESTION
1274	003120	012767	000001	176042	4\$	MOV	#1,ROTADD ; SET UP FOR LATER USE IN END OF PASS ROUTINE
1275	003126	016767	176024	176020		MOV	KEEPADD,BASEADD ; DITTO
1276	003134	000414				BR	OUTMUL ; CONTINUE QUESTIONS
1277	003136	016767	176014	176010	3\$	MOV	KEEPADD,BASEADD ; RESTORE
1278	003144	104406				INSTR	; OUTPUT MESSAGE & GET INPUT STRING
1279	003146	015357				MRANGE	; MESSAGE
1280	003150	104410				PARAM	; CONVERT STRING
1281	003152	160000				160000	; LOW LIMIT
1282	003154	167776				167776	; HIGH LIMIT
1283	003156	001160				LASTADD	; STORE AT THIS LOCATION
1284	003160	001			BYTE	1	; MASK
1285	003161	001			BYTE	1	; HOW MANY TIMES + 2
1286	003162	000167	177656			JMP	1\$; DO IT AGAIN
1287	003166	012767	000340	013602		OUTMUL	MOV #340,DUPRT

INITIALIZE THE COMMON TAGS

```

1288 003174 004767 013526 JSR PC,DULEV
1289 ,COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1290 ,BUFFER TO THE CHARACTERS "1" AND "2"
1291 ,IF THE CHARACTER IS "1" CLEAR THE FLAG
1292 ,IF THE CHARACTER IS "2" SET THE FLAG
1293 003200 AAA
1294 003200 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1295 003202 015575 MSYNC ;MESSAGE
1296 003204 122767 000061 012724 3$ CMPB #'1,INBUF ;IS IT "1" ?
1297 003212 001003 BNE 1$
1298 003214 105067 175726 CLRB SYNCNO ,000
1299 003220 000412 BR 4$
1300 003222 122767 000062 012706 1$ CMPB #'2,INBUF ;IS IT "2" ?
1301 003230 001004 BNE 2$
1302 003232 112767 177777 175706 MOVB #-1,SYNCNO ;377
1303 003240 000402 BR 4$
1304 003242 104407 2$ INSTER ,RETRY
1305 003244 000757 BR 3$
1306 003246 000240 4$ NOP
1307 003250 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1308 003252 015643 MWIRE6 ,MESSAGE
1309 003254 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1310 003256 001147 SEXMIT ,THIS FLAG
1311 003260 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1312 003262 015714 MWIRE5 ,MESSAGE
1313 003264 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1314 003266 001150 SEREC ,THIS FLAG
1315 003270 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1316 003272 015764 MWIRE4 ,MESSAGE
1317 003274 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1318 003276 001151 OPTCLR ,THIS FLAG
1319 003300 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STRING
1320 003302 016043 MEXTJ ,MESSAGE
1321 003304 104414 SETFLG ,SET FLAG BASED UPON INPUT STRING
1322 003306 001153 JMRBY ,THIS FLAG
1323
1324 ,TEST START AND RESTART
1325
1326 003310 012706 001100 BEGIN: MOV #STACK,SP ,SET UP STACK
1327 003314 106427 000340 MTPS #340 ,LOCK OUT INTERRUPTS
1328 003320 032777 000002 176112 BIT #SW01,@SWR ,IF SW01=1, GET STARTING PC
1329 003326 001413 BEQ 3$
1330 003330 104406 INSTR ,OUTPUT MESSAGE & GET INPUT STR NG
1331 003332 015527 MTSTPC ,MESSAGE
1332 003334 104410 PARAM ,CONVERT STRING
1333 003336 003374 TST1 ,LOW LIMIT
1334 003340 017500 17500 ,HIGH LIMIT
1335 003342 001402 $STSTM ,STORE AT THIS LOCATION
1336 003344 001 BYTE 1 ,MASK
1337 003345 001 BYTE 1 ,HOW MANY TIMES + 2
1338 003346 016767 176030 175530 MOV $STSTM,RETURN
1339 003354 000403 BR 4$
1340 003356 012767 003374 175520 3$ MOV #TST1,RETURN ,START AT TEST 1
1341 003364 104401 015523 4$ TYPE ,MR ,TYPE R
1342 003370 000177 175510 JMP @RETURN ,START TESTING
1343
    
```

INITIALIZE THE COMMON TAGS

```
1344 ; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1345 ; RECEIVER SECTION, IT USES THE ERROR FLAGS
1346 ; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1347 ; (OVERRUN, RXERR)
1348 ; MODE SYNEXT
1349 ; LENGTH: SEVEN
1350 ; CHAR. 177
1351 ;
1352 ; *****
1353 TST1 SCOPE
1354 003374 000004 BIS #MRESET, @TXCSR ; MASTER RESET
1355 003376 052777 000400 176322 MOV #SYNEXT, @PARCSR ; SET THE MODE
1356 003404 012777 020000 176310 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 024000 176266 MOV #SYNEXT!SEVEN!NOPAR!0, @PARCSR
1363 003434 052777 000020 176250 BIS #SYNSCH, @RXCSR ; SET SEARCH SYNC
1364 ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1365 003442 042777 020000 176256 BIC #CLK, @TXCSR ; POKE CLK DOWN
1366 003450 052777 020000 176250 BIS #CLK, @TXCSR ; POKE CLK UP
1367 003456 016703 176234 MOV #RXDBUF, R3 ; SET UP FOR ERROR MESSAGE
1368 003462 012700 000177 MOV #177, R0 ; EXPECTED
1369 003466 012767 000007 175426 MOV #7, SHIFT ; # OF SHIFTS
1370 003474 012767 000177 175776 MOV #177, $TMP1 ; DATA CHAR
1371 003502 004767 013430 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1372 003506 105777 176200 TSTB @RXCSR, RXDONE ?
1373 003512 100401 BMI +4
1374 003514 104004 ERROR 4 ; RXDONE SHOULD BE SET
1375 003516 017701 176174 MOV @RXDBUF, R1 ; ACTUAL
1376 003522 020001 CMP R0, R1 ; COMPARE EXPECTED VS ACTUAL
1377 003524 001401 BEQ +4
1378 003526 104002 ERROR 2 ; RECEIVED DATA DID NOT MATCH
1379 ; EXPECTED DATA - CHECK MAINT DATA
1380 ; OR RECEIVER LOGIC
1381 003530 012767 000007 175364 MOV #7, SHIFT ; # OF SHIFTS
1382 003536 012767 000177 175734 MOV #177, $TMP1 ; DATA CHAR
1383 003544 004767 013366 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1384 ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1385 003550 012767 000007 175344 MOV #7, SHIFT ; # OF SHIFTS
1386 003556 012767 000177 175714 MOV #177, $TMP1 ; DATA CHAR
1387 003564 004767 013346 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1388 003570 012700 140177 MOV #140000!177, R0 ; EXPECTED DATA PLUS
1389 ; RXERR & OVERRUN
1390 003574 017701 176116 MOV @RXDBUF, R1 ; ACTUAL
1391 003600 020001 CMP R0, R1 ; COMPARE EXP VS ACT
1392 003602 001401 BEQ +4
1393 003604 104002 ERROR 2 ; SPECIFICALLY LOOK AT RXERR &
1394 ; OVERRUN BITS THEY BOTH SHOULD BE SET
1395
1396 ; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1397 ; RECEIVER SECTION, IT USES THE ERROR FLAGS
1398 ; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1399 ; (OVERRUN, RXERR)
```

```

1400          ;,MODE:SYNEXT
1401          ;,LENGTH:SEVEN
1402          ;,CHAR:0
1403          ;,
1404          ;,*****
1405 003606 000004 TST2: SCOPE
1406 003610 052777 000400 176110 BIS #MRESET,@TXCSR ;MASTER RESET
1407 003616 012777 020000 176076 MOV #SYNEXT,@PARCSR ;SET THE MODE
1408 003624 052777 000400 176074 BIS #MRESET,@TXCSR ;MASTER RESET
1409
1410          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1411 003632 012777 064001 176066 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1412
1413          ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1414 003640 012777 024000 176054 MOV #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1415 003646 052777 000020 176036 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1416          ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1417 003654 042777 020000 176044 BIC #CLK,@TXCSR ;POKE CLK DOWN
1418 003662 052777 020000 176036 BIS #CLK,@TXCSR ;POKE CLK UP
1419 003670 016703 176022 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1420 003674 012700 000000 MOV #0,R0 ;EXPECTED
1421 003700 012767 000007 175214 MOV #7,SHIFT ;# OF SHIFTS
1422 003706 012767 000000 175564 MOV #0,$TMP1 ;DATA CHAR
1423 003714 004767 013216 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1424 003720 105777 175766 TSTB @RXCSR ,RXDONE ?
1425 003724 100401 BMI +4
1426 003726 104004 ERROR 4 ;RXDONE SHOULD BE SET
1427 003730 017701 175762 MOV @RXDBUF,R1 ;ACTUAL
1428 003734 020001 CMP R0,R1 ;COMPARE EXPECTED VS ACTUAL
1429 003736 001401 BEQ +4
1430 003740 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1431 ;EXPECTED DATA - CHECK MAINT DATA
1432 ;OR RECEIVER LOGIC
1433 003742 012767 000007 175152 MOV #7,SHIFT ;# OF SHIFTS
1434 003750 012767 000000 175522 MOV #0,$TMP1 ;DATA CHAR
1435 003756 004767 013154 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1436          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1437 003762 012767 000007 175132 MOV #7,SHIFT ;# OF SHIFTS
1438 003770 012767 000000 175502 MOV #0,$TMP1 ;DATA CHAR
1439 003776 004767 013134 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1440 004002 012700 140000 MOV #140000!0,R0 ;EXPECTED DATA PLUS
1441 ;RXERR & OVRUN
1442 004006 017701 175704 MOV @RXDBUF,R1 ;ACTUAL
1443 004012 020001 CMP R0,R1 ;COMPARE EXP VS ACT
1444 004014 001401 BEQ +4
1445 004016 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1446 ;OVRUN BITS THEY BOTH SHOULD BE SET
1447
1448          ;,THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1449          ;,RECEIVER SECTION.IT USES THE ERROR FLAGS
1450          ;,TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1451          ;,(OVRUN,RXERR)
1452          ;,MODE SYNEXT
1453          ;,LENGTH EIGHT
1454          ;,CHAR 125
1455          ;,

```



```
1456 ,, *****  
1457 004020 000004 TST3 SCOPE  
1458 004022 052777 000400 175676 BIS #MRESET,@TXCSR ,MASTER RESET  
1459 004030 012777 020000 175664 MOV #SYNEXT,@PP'CSR ;SET THE MODE  
1460 004036 052777 000400 175662 BIS #MRESET,@TXCSR ,MASTER RESET  
1461  
1462 ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE  
1463 004044 012777 064001 175654 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR  
1464  
1465 ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG  
1466 004052 012777 026000 175642 MOV #SYNEXT!EIGHT!NOPAR!0,@PARCSR  
1467 004060 052777 000020 175624 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC  
1468 ,POKE CLK TO GET LOGIC INTO SYNCHRONIZATION  
1469 004066 042777 020000 175632 BIC #CLK,@TXCSR ;POKE CLK DOWN  
1470 004074 052777 020000 175624 BIS #CLK,@TXCSR ;POKE CLK UP  
1471 004102 016703 175610 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE  
1472 004106 012700 000125 MOV #125,RO ,EXPECTED  
1473 004112 012767 000010 175002 MOV #8 ,SHIFT ,# OF SHIFTS  
1474 004120 012767 000125 175352 MOV #125,$TMP1 ,DATA CHAR  
1475 004126 004767 013004 JSR PC,RPOKE ,SHIFT IN THIS CHAR  
1476 004132 105777 175554 TSTB @RXCSR ,RXDONE ?  
1477 004136 100401 BMI +4  
1478 004140 104004 ERROR 4 ,RXDONE SHOULD BE SET  
1479 004142 017701 175550 MOV @RXDBUF,R1 ,ACTUAL  
1480 004146 020001 CMP RO,R1 ,COMPARE EXPECTED VS ACTUAL  
1481 004150 001401 BEQ +4  
1482 004152 104002 ERROR 2 ,RECEIVED DATA DID NOT MATCH  
1483 ,EXPECTED DATA - CHECK MAINT DATA  
1484 ,OR RECEIVER LOGIC  
1485 004154 012767 000010 174740 MOV #8 ,SHIFT ,# OF SHIFTS  
1486 004162 012767 000125 175310 MOV #125,$TMP1 ,DATA CHAR  
1487 004170 004767 012742 JSR PC,RPOKE ,SHIFT IN THIS CHAR  
1488 ,NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF  
1489 004174 012767 000010 174720 MOV #8 ,SHIFT ,# OF SHIFTS  
1490 004202 012767 000125 175270 MOV #125,$TMP1 ,DATA CHAR  
1491 004210 004767 012722 JSR PC,RPOKE ,SHIFT IN THIS CHAR  
1492 004214 012700 140125 MOV #140000!125,RO ,EXPECTED DATA PLUS  
1493 ,RXERR & OVRUN  
1494 004220 017701 175472 MOV @RXDBUF,R1 ,ACTUAL  
1495 004224 020001 CMP RO,R1 ,COMPARE EXP VS ACT  
1496 004226 001401 BEQ +4  
1497 004230 104002 ERROR 2 ,SPECIFICALLY LOOK AT RXERR &  
1498 ,OVRUN BITS THEY BOTH SHOULD BE SET  
1499  
1500 ,THIS TEST VERIFYS WORD LENGTH SELECT OF THE  
1501 ,,RECEIVER SECTION,IT USES THE ERROR FLAGS  
1502 ,,TO DETERMINE THAT IT WAS SELECTED CORRECTLY  
1503 ,, (OVRUN,RXERR)  
1504 ,,MODE SYNEXT  
1505 ,,LENGTH EIGHT  
1506 ,,CHAR. 252  
1507  
1508 ,, *****  
1509 004232 000004 TST4 SCOPE  
1510 004234 052777 000400 175464 BIS #MRESET,@TXCSR ,MASTER RESET  
1511 004242 012777 020000 175452 MOV #SYNEXT,@PARCSR ,SET THE MODE
```

INITIALIZE THE COMMON TAGS

```
1512 004250 052777 000400 175450      BIS      #MRESET,@TXCSR ,MASTER RESET
1513
1514      ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1515 004256 012777 064001 175442      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1516
1517      ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1518 004264 012777 026000 175430      MOV      #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1519 004272 052777 000020 175412      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
1520      ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1521 004300 042777 020000 175420      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1522 004306 052777 020000 175412      BIS      #CLK,@TXCSR ;POKE CLK UP
1523 004314 016703 175376      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1524 004320 012700 000252      MOV      #252,R0 ,EXPECTED
1525 004324 012767 000010 174570      MOV      #8 ,SHIFT ;# OF SHIFTS
1526 004332 012767 000252 175140      MOV      #252,$TMP1 ;DATA CHAR
1527 004340 004767 012572      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1528 004344 105777 175342      TSTB    @RXCSR ,RXDONE ?
1529 004350 100401      BMI      +4
1530 004352 104004      ERROR   4 ;RXDONE SHOULD BE SET
1531 004354 017701 175336      MOV      @RXDBUF,R1 ;ACTUAL
1532 004360 020001      CMP      R0,R1 ;COMPARE EXPECTED VS ACTUAL
1533 004362 001401      BEQ      +4
1534 004364 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
1535      ,EXPECTED DATA - CHECK MAINT DATA
1536      ,OR RECEIVER LOGIC
1537 004366 012767 000010 174526      MOV      #8 ,SHIFT ;# OF SHIFTS
1538 004374 012767 000252 175076      MOV      #252,$TMP1 ;DATA CHAR
1539 004402 004767 012530      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1540      ,NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1541 004406 012767 000010 174506      MOV      #8 ,SHIFT ;# OF SHIFTS
1542 004414 012767 000252 175056      MOV      #252,$TMP1 ;DATA CHAR
1543 004422 004767 012510      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1544 004426 012700 140252      MOV      #140000!252,R0 ;EXPECTED DATA PLUS
1545      ,RXERR & OVRUN
1546 004432 017701 175260      MOV      @RXDBUF,R1 ;ACTUAL
1547 004436 020001      CMP      R0,R1 ;COMPARE EXP VS ACT
1548 004440 001401      BEQ      +4
1549 004442 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERP &
1550      ,OVRUN BITS THEY BOTH SHOULD BE SET
1551
1552      ,, THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1553      ,, RECEIVER SECTION, IT USES THE ERROR FLAGS
1554      ,, TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1555      ,, (OVRUN,RXERR)
1556      ,, MODE SYNEXT
1557      ,, LENGTH EIGHT
1558      ,, CHAR 37?
1559      ,,
1560      ,, *****
1561 004444 000004      TSTB    SCOPE
1562 004446 052777 000400 175252      BIS      #MRESET,@TXCSR ,MASTER RESET
1563 004454 012777 020000 175240      MOV      #SYNEXT,@PARCSR ,SET THE MODE
1564 004462 052777 000400 175236      BIS      #MRESET,@TXCSR ,MASTER RESET
1565
1566      ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1567 004470 012777 064001 175230      MOV      #MTDATA!CLK!MINT!BREAK @TXCSR
```

```

1568
1569
1570 004476 012777 026000 175216
1571 004504 052777 000020 175200
1572
1573 004512 042777 020000 175206
1574 004520 052777 020000 175200
1575 004526 016703 175164
1576 004532 012700 000377
1577 004536 012767 000010 174356
1578 004544 012767 000377 174726
1579 004552 004767 012360
1580 004556 105777 175130
1581 004562 100401
1582 004564 104004
1583 004566 017701 175124
1584 004572 020001
1585 004574 001401
1586 004576 104002
1587
1588
1589 004600 012767 000010 174314
1590 004606 012767 000377 174664
1591 004614 004767 012316
1592
1593 004620 012767 000010 174274
1594 004626 012767 000377 174644
1595 004634 004767 012276
1596 004640 012700 140377
1597
1598 004644 017701 175046
1599 004650 020001
1600 004652 001401
1601 004654 104002
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613 004656 000704
1614 004660 052777 000400 175040
1615 004666 012777 020000 175026
1616 004674 052777 000400 175024
1617
1618
1619 004702 012777 064001 175016
1620
1621
1622 004710 012777 026000 175004
1623 004716 052777 000020 174766

```

, SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
 MOV #SYNEXT!EIGHT!NOPAR!0, @PARCSR
 BIS #SYNSCH, @RXCSR , SET SEARCH SYNC
 , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
 BIC #CLK, @TXCSR ; POKE CLK DOWN
 BIS #CLK, @TXCSR ; POKE CLK UP
 MOV RXDBUF, R3 ; SET UP FOR ERROR MESSAGE
 MOV #377, R0 , EXPECTED
 MOV #8 , SHIFT , # OF SHIFTS
 MOV #377, \$TMP1 , DATA CHAR
 JSR PC, RPOKE , SHIFT IN THIS CHAR
 TSTB @RXCSR , RXDONE ?
 BMI +4
 ERROR 4 , RXDONE SHOULD BE SET
 MOV @RXDBUF, R1 , ACTUAL
 CMP R0, R1 , COMPARE EXPECTED VS ACTUAL
 BEQ +4
 ERROR 2 , RECEIVED DATA DID NOT MATCH
 , EXPECTED DATA - CHECK MAINT DATA
 , OR RECEIVER LOGIC
 MOV #8 , SHIFT , # OF SHIFTS
 MOV #377, \$TMP1 , DATA CHAR
 JSR PC, RPOKE , SHIFT IN THIS CHAR
 , NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
 MOV #8 , SHIFT , # OF SHIFTS
 MOV #377, \$TMP1 , DATA CHAR
 JSR PC, RPOKE , SHIFT IN THIS CHAR
 MOV #140000!377, R0 , EXPECTED DATA PLUS
 , RXERR & OVERRUN
 MOV @RXDBUF, R1 , ACTUAL
 CMP R0, R1 , COMPARE EXP VS ACT
 BEQ +4
 ERROR 2 , SPECIFICALLY LOOK AT RXERR &
 , OVERRUN BITS THEY BOTH SHOULD BE SET
 , , THIS TEST VERIFYS WORD LENGTH SELECT OF THE
 , , RECEIVER SECTION, IT USES THE ERROR FLAGS
 , , TO DETERMINE THAT IT WAS SELECTED CORRECTLY
 , , (OVERRUN, RXERR)
 , ; MODE SYNEXT
 , , LENGTH EIGHT
 , ; CHAR 0
 , ,
 , , *****
 TST6 SCOPE
 BIS #MRESET, @TXCSR , MASTER RESET
 MOV #SYNEXT, @PARCSR , SET THE MODE
 BIS #MRESET, @TXCSR , MASTER RESET
 , SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
 MOV #MTOATA!CLK!MINT!BREAK, @TXCSR
 , SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
 MOV #SYNEXT!EIGHT!NOPAR!0, @PARCSR
 BIS #SYNSCH, @RXCSR , SET SEARCH SYNC

```

1624      .POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1625 004724 042777 020000 174774      BIC   #CLK,@TXCSR      ,POKE CLK DOWN
1626 004732 052777 020000 174766      BIS   #CLK,@TXCSR      ;POKE CLK UP
1627 004740 016703 174752              MOV   RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
1628 004744 012700 000000              MOV   #0,R0           ;EXPECTED
1629 004750 012767 000010 174144      MOV   #8,SHIFT        ;# OF SHIFTS
1630 004756 012767 000000 174514      MOV   #0,$TMP1        ;DATA CHAR
1631 004764 004767 012146              JSR   PC,RPOKE        ,SHIFT IN THIS CHAR
1632 004770 105777 174716              TSTB  @RXCSR ,RXDONE ?
1633 004774 100401                      BMI   .+4
1634 004776 104004                      ERROR 4              ,RXDONE SHOULD BE SET
1635 005000 017701 174712              MOV   @RXDBUF,R1     ;ACTUAL
1636 005004 020001                      CMP   R0,R1          ,COMPARE EXPECTED VS. ACTUAL
1637 005006 001401                      BEQ   .+4
1638 005010 104002                      ERROR 2              ;RECEIVED DATA DID NOT MATCH
1639                                     ;EXPECTED DATA - CHECK MAINT DATA
1640                                     ;OR RECEIVER LOGIC
1641 005012 012767 000010 174102      MOV   #8,SHIFT        ;# OF SHIFTS
1642 005020 012767 000000 174452      MOV   #0,$TMP1        ;DATA CHAR
1643 005026 004767 012104              JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
1644      .NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1645 005032 012767 000010 174062      MOV   #8,SHIFT        ;# OF SHIFTS
1646 005040 012767 000000 174432      MOV   #0,$TMP1        ;DATA CHAR
1647 005046 004767 012064              JSR   PC,RPOKE        ;SHIFT IN THIS CHAR
1648 005052 012700 140000              MOV   #140000!0,R0   ,EXPECTED DATA PLUS
1649                                     ,RXERR & OVRRUN
1650 005056 017701 174634              MOV   @RXDBUF,R1     ,ACTUAL
1651 005062 020001                      CMP   R0,R1          ;COMPARE EXP VS ACT
1652 005064 001401                      BEQ   +4
1653 005066 104002                      ERROR 2              ,SPECIFICALLY LOOK AT RXERR &
1654                                     ;OVRRUN BITS  THEY BOTH SHOULD BE SET
1655
1656      ., THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1657      ., SECTION , IT USES THE ERROR FLAGS TO DETERMINE
1658      ., THAT IT WAS SELECTED PROPERLY
1659      ., FRAME ERROR (FRMERR,RXERR)
1660      ., MODE ISOC (ISYMOD)
1661      ., LENGTH FIVE
1662      ., CHAR 25
1663      .,
1664      ., *****
1665 005070 000004      TST7  SCOPE
1666 005072 052777 000400 174626      BIS   #MRESET,@TXCSR ,MASTER RESET
1667 005100 012777 000000 174614      MOV   #ISYMOD,@PARCSR ,SET THE MODE
1668 005106 052777 000400 174612      BIS   #MRESET,@TXCSR ,MASTER RESET
1669
1670      .SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1671 005114 012777 064001 174604      MOV   #MTDATA'CLK'MINT'BREAK,@TXCSR
1672
1673      .SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1674 005122 012777 000000 174572      MOV   #ISYMOD'FIVE'NOPAR'0,@PARCSR
1675 005130 052777 000020 174554      BIS   #SYNSCH,@RXCSR ,SET SYNC SEARCH
1676      .POKE CLK TO GET RECEIVER INTO SYNCRONIZATION
1677 005136 042777 020000 174562      BIC   #CLK,@TXCSR      ,POKE CLK DOWN
1678 005144 052777 020000 174554      BIS   #CLK,@TXCSR      ,POKE CLK UP
1679      .POKE CLK TO GET LOGIC INTO SYNCRONIZATION

```

INITIALIZE THE COMMON TAGS

```
1680 005152 042777 020000 174546      BIC    #CLK,@TXCSR    ,POKE CLK DOWN
1681 005160 052777 020000 174540      BIS    #CLK,@TXCSR    ;POKE CLK UP
1682 005166 012767 000007 173726      MOV    #7,SHIFT      ;# OF SHIFTS
1683 005174 012767 000052 174276      MOV    #52,$TMP1     ;DATA CHAR
1684                                     ,NOTE THE ABOVE CHARACTER IS MISSING STOP BIT
1685 005202 004767 011730      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1686 005206 016703 174504      MOV    RXDBUF,R3     ;FOR ERROR MESSAGE
1687 005212 012700 120025      MOV    #RXERR!FRMERR!25,RO ;EXPECTED
1688 005216 017701 174474      MOV    @RXDBUF,R1    ;ACTUAL
1689 005222 020001      CMP    RO,R1        ;COMPARE EXP VS ACT
1690 005224 001401      BEQ    +4
1691 005226 104000      ERROR  ,FRAME ERROR 4 & RX ERROR SHOULD BE SET
1692                                     ,IF LOWER BYTE DOES NOT MATCH IT
1693                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1694
1695                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1696                                     ;; SECTION , IT USES THE ERROR FLAGS TO DETERMINE
1697                                     ;; THAT IT WAS SELECTED PROPERLY
1698                                     ;; FRAME ERROR (FRMERR,RXERR)
1699                                     ;; MODE: ISOC (ISYMOD)
1700                                     ;; LENGTH: SIX
1701                                     ;; CHAR. 25
1702                                     ;;
1703                                     ;; *****
1704 005230 000004      TST10 SCOPE
1705 005232 052777 000400 174466      BIS    #MRESET,@TXCSR ,MASTER RESET
1706 005240 012777 000J00 174454      MOV    #ISYMOD,@PARCSR ,SET THE MODE
1707 005246 052777 000400 174452      BIS    #MRESET,@TXCSR ,MASTER RESET
1708
1709                                     ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1710 005254 012777 064001 174444      MOV    #MTDATA'CLK'MINT'BREAK,@TXCSR
1711
1712                                     ,SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1713 005262 012777 002000 174432      MOV    #ISYMOD'SIX'NOPAR'0,@PARCSR
1714 005270 052777 000020 174414      BIS    #SYNSCH,@RXCSR ,SET SYNC SEARCH
1715                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1716 005276 042777 020000 174422      BIC    #CLK,@TXCSR    ,POKE CLK DOWN
1717 005304 052777 020000 174414      BIS    #CLK,@TXCSR    ;POKE CLK UP
1718                                     ,POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1719 005312 042777 020000 174406      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
1720 005320 052777 020000 174400      BIS    #CLK,@TXCSR    ,POKE CLK UP
1721 005326 012767 000010 173566      MOV    #8,SHIFT      ;# OF SHIFTS
1722 005334 012767 000052 174136      MOV    #52,$TMP1     ;DATA CHAR
1723                                     ,NOTE THE ABOVE CHARACTER IS MISSING STOP BIT
1724 005342 004767 011570      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
1725 005346 016703 174344      MOV    RXDBUF,R3     ;FOR ERROR MESSAGE
1726 005352 012700 120025      MOV    #RXERR!FRMERR!25,RO ;EXPECTED
1727 005356 017701 174334      MOV    @RXDBUF,R1    ;ACTUAL
1728 005362 020001      CMP    RO,R1        ;COMPARE EXP VS ACT
1729 005364 001401      BEQ    +4
1730 005366 104000      ERROR  ,FRAME ERROR 4 & RX ERROR SHOULD BE SET
1731                                     ,IF LOWER BYTE DOES NOT MATCH IT
1732                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1733
1734                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1735                                     ;; SECTION , IT USES THE ERROR FLAGS TO DETERMINE
```

```
1736      , , THAT IT WAS SELECTED PROPERLY
1737      , , FRAME ERROR (FRMERR,RXERR)
1738      , , MODE ISOC (ISYMOD)
1739      , , LENGTH: SEVEN
1740      , , CHAR. 125
1741      , ,
1742      , , *****
1743 005370 000004 TST11 SCOPE
1744 005372 052777 000400 174326 BIS #MRESET,@TXCSR ;MASTER RESET
1745 005400 012777 000000 174314 MOV #ISYMOD,@PARCSR ;SET THE MODE
1746 005406 052777 000400 174312 BIS #MRESET,@TXCSR ;MASTER RESET
1747
1748      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1749 005414 012777 064001 174304 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1750
1751      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1752 005422 012777 004000 174272 MOV #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1753 005430 052777 000020 174254 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1754      , POKE CLK TO GET RECEIVER INTO SYNCRIZATION
1755 005436 042777 020000 174262 BIC #CLK,@TXCSR ,POKE CLK DOWN
1756 005444 052777 020000 174254 BIS #CLK,@TXCSR ,POKE CLK UP
1757      , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1758 005452 042777 020000 174246 BIC #CLK,@TXCSR ,POKE CLK DOWN
1759 005460 052777 020000 174240 BIS #CLK,@TXCSR ;POKE CLK UP
1760 005466 012767 000011 173426 MOV #9 ,SHIFT ,# OF SHIFTS
1761 005474 012767 000252 173776 MOV #252,$TMP1 ,DATA CHAR
1762      , NOTE THE ABOVE CHARACTER IS MISSING STOP BIT
1763 005502 004767 011430 JSR PC,RPOKE ,SHIFT IN THIS CHAR
1764 005506 016703 174204 MOV RXDBUF,R3 ,FOR ERROR MESSAGE
1765 005512 012700 120125 MOV #RXERR!FRMERR!125,R0 ,EXPECTED
1766 005516 017701 174174 MOV @RXDBUF,R1 ,ACTUAL
1767 005522 020001 CMP R0,R1 ,COMPARE EXP VS ACT
1768 005524 001401 BEQ +4
1769 005526 104000 ERROR ,FRAME ERROR 4 & RX ERROR SHOULD BE SET
1770      , IF LOWER BYTE DOES NOT MATCH IT
1771      , PROBABLY IS A LENGTH SELECT PROBLEM
1772
1773      , , THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1774      , , SECTION , IT USES THE ERROR FLAGS TO DETERMINE
1775      , , THAT IT WAS SELECTED PROPERLY
1776      , , FRAME ERROR (FRMERR,RXERR)
1777      , , MODE ISOC (ISYMOD)
1778      , , LENGTH: EIGHT
1779      , , CHAR. 125
1780      , ,
1781      , , *****
1782 005530 000004 TST12 SCOPE
1783 005532 052777 000400 174166 BIS #MRESET,@TXCSR ,MASTER RESET
1784 005540 012777 000000 174154 MOV #ISYMOD,@PARCSR ,SET THE MODE
1785 005546 052777 000400 174152 BIS #MRESET,@TXCSR ;MASTER RESET
1786
1787      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1788 005554 012777 064001 174144 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1789
1790      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1791 005562 012777 006000 174132 MOV #ISYMOD!EIGHT!NOPAR!0,@PARCSR
```

INITIALIZE THE COMMON TAGS

```
1792 005570 052777 000020 174114 BIS #SYNSCH,@RXCSR ,SET SYNC SEARCH
1793 ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1794 005576 042777 020000 174122 BIC #CLK,@TXCSR ,POKE CLK DOWN
1795 005604 052777 020000 174114 BIS #CLK,@TXCSR ;POKE CLK UP
1796 ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1797 005612 042777 020000 174106 BIC #CLK,@TXCSR ,POKE CLK DOWN
1798 005620 052777 020000 174100 BIS #CLK,@TXCSR ,POKE CLK UP
1799 005626 012767 000012 173266 MOV #10,SHIFT ;# OF SHIFTS
1800 005634 012767 000252 173636 MOV #252,$TMP1 ,DATA CHAR
1801 ,NOTE THE ABOVE CHARACTER IS MISSING STOP BIT
1802 005642 004767 011270 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1803 005646 016703 174044 MOV RXDBUF,R3 ,FOR ERROR MESSAGE
1804 005652 012700 120125 MOV #RXERR!RMERR!125,RO ;EXPECTED
1805 005656 017701 174034 MOV @RXDBUF,R1 ,ACTUAL
1806 005662 020001 CMP RO,R1 ,COMPARE EXP VS ACT
1807 005664 001401 BEQ +4
1808 005666 104000 ERROR ,FRAME ERROR 4 & RX ERROR SHOULD BE SET
1809 ,IF LOWER BYTE DOES NOT MATCH IT
1810 ,PROBABLY IS A LENGTH SELECT PROBLEM
1811
1812 ,,THIS TEST VERIFYS EVEPAR PAP,ITY SENSE
1813 ,,OF THE RECEIVER
1814 ,,MODE: ISOC (ISYMOD)
1815 ,,PARITY: EVEPAR
1816 ,,LENGTH: FIVE PLUS PARITY
1817 ,,CHAR: 25
1818 ,,
1819 ,,*****
1820 005670 000004 TST13 SCOPE
1821 005672 052777 000400 174026 BIS #MRESET,@TXCSR ,MASTER RESET
1822 005700 012777 000000 174014 MOV #ISYMOD,@PARCSR ,SET THE MODE
1823 005706 052777 000400 174012 BIS #MRESET,@TXCSR ,MASTER RESET
1824
1825 ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1826 005714 012777 064001 174004 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1827
1828 ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1829 005722 012777 001400 173772 MOV #ISYMOD!FIVE!EVEPAR!0,@PARCSR
1830 005730 052777 000020 173754 BIS #SYNSCH,@RXCSR ,SET SYNC SEARCH
1831 ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1832 005736 042777 020000 173762 BIC #CLK,@TXCSR ,POKE CLK DOWN
1833 005744 052777 020000 173754 BIS #CLK,@TXCSR ,POKE CLK UP
1834 ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1835 005752 042777 020000 173746 BIC #CLK,@TXCSR ,POKE CLK DOWN
1836 005760 052777 020000 173740 BIS #CLK,@TXCSR ,POKE CLK UP
1837 005766 016703 173724 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1838 005772 012700 110025 MOV #RXERR!PARER!25,RO ,EXPECTED
1839 005776 012767 000010 173116 MOV #8,SHIFT ;# OF SHIFTS
1840 006004 012767 000252 173466 MOV #252,$TMP1 ,DATA CHAR
1841 006012 004767 011120 JSR PC,RPOKE ,SHIFT IN THIS CHAR
1842 006016 105777 173670 TSTB @RXCSR ,RXDONE ?
1843 006022 100401 BMI +4
1844 006024 104004 ERROR 4 ,RXDONE SHOULD BF ASSERTED
1845 006026 017701 173664 MOV @RXDBUF,R1 ,ACTUAL
1846 006032 020001 CMP RO,R1 ,COMPARE EXP VS ACT
1847 006034 001401 BEQ +4
```

INITIALIZE THE COMMON TAGS

```
1848 006036 104000          ERROR , PARITY ERROR 4 &RXERR SHOULD BE SET
1849                                     , NOTE THAT THE PARITY BIT SHOULD
1850                                     ; SHOW UP IN THE DATA
1851                                     ; IE. BIT FIVE FOR FIVE LEVEL CODE
1852
1853                                     .. THIS TEST VERIFYS EVEPAR PARITY SENSE
1854                                     .. OF THE RECEIVER
1855                                     .. MODE ISOC (ISYMOD)
1856                                     .. PARITY EVEPAR
1857                                     .. LENGTH SIX PLUS PARITY
1858                                     .. CHAR 25
1859
1860                                     .. *****
1861 006040 000004          TST14 SCOPE
1862 006042 052777 000400 173656      BIS #MRESET,@TXCSR ; MASTER RESET
1863 006050 012777 000000 173644      MOV #ISYMOD,@PARCSR ; SET THE MODE
1864 006056 052777 000400 173642      BIS #MRESET,@TXCSR ; MASTER RESET
1865
1866                                     . SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1867 006064 012777 064001 173634      MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1868
1869                                     . SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
1870 006072 012777 003400 173622      MOV #ISYMOD!SIX!EVEPAR!D,@PARCSR
1871 006100 052777 000020 173604      BIS #SYNSCH,@RXCSR ; SET SYNC SEARCH
1872                                     . POKE CLK TO GET RECEIVER INTO SYNCRIZATION
1873 006106 042777 020000 173612      BIC #CLK,@TXCSR ; POKE CLK DOWN
1874 006114 052777 020000 173604      BIS #CLK,@TXCSR ; POKE CLK UP
1875                                     . POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1876 006122 042777 020000 173576      BIC #CLK,@TXCSR ; POKE CLK DOWN
1877 006130 052777 020000 173570      BIS #CLK,@TXCSR ; POKE CLK UP
1878 006136 016703 173554          MOV RXDBUF,R3 ; SET UP FOR ERROR MESSAGE
1879 006142 012700 110025          MOV #RXERR!PARER!25,R0 ; EXPECTED
1880 006146 012767 000011 172746      MOV #9,SHIFT ; # OF SHIFTS
1881 006154 012767 000452 173316      MOV #452,$TMP1 ; DATA CHAR
1882 006162 004767 010750          JSR PC,RPOKE ; SHIFT IN THIS CHAR
1883 006166 105777 173520          TSTB @RXCSR ,RXDONE ?
1884 006172 100401          BMI +4
1885 006174 104004          ERROR 4 ; RXDONE SHOULD BE ASSERTED
1886 006176 017701 173514          MOV @RXDBUF,R1 ; ACTUAL
1887 006202 020001          CMP R0,R1 ; COMPARE EXP VS. ACT
1888 006204 001401          BEQ +4
1889 006206 104000          ERROR , PARITY ERROR 4 &RXERR SHOULD BE SET
1890                                     , NOTE THAT THE PARITY BIT SHOULD
1891                                     ; SHOW UP IN THE DATA
1892                                     ; IE BIT SIX FOR SIX LEVEL CODE
1893
1894                                     .. THIS TEST VERIFYS EVEPAR PARITY SENSE
1895                                     .. OF THE RECEIVER
1896                                     .. MODE ISOC (ISYMOD)
1897                                     .. PARITY EVEPAR
1898                                     .. LENGTH SEVEN PLUS PARITY
1899                                     .. CHAR 325
1900
1901                                     .. *****
1902 006210 000004          TST15 SCOPE
1903 006212 052777 000400 173506      BIS #MRESET,@TXCSR ; MASTER RESET
```



```
1904 006220 012777 000000 173474      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1905 006226 052777 000400 173472      BIS    #MRESET,@TXCSR ;MASTER RESET
1906
1907                                     ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1908 006234 012777 064001 173464      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1909
1910                                     ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1911 006242 012777 005400 173452      MOV    #ISYMOD!SEVEN!EVEPAR!0,@PARCSR
1912 006250 052777 000020 173434      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
1913                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1914 006256 042777 020000 173442      BIC    #CLK,@TXCSR ;POKE CLK DOWN
1915 006264 052777 020000 173434      BIS    #CLK,@TXCSR ;POKE CLK UP
1916                                     ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1917 006272 042777 020000 173426      BIC    #CLK,@TXCSR ;POKE CLK DOWN
1918 006300 052777 020000 173420      BIS    #CLK,@TXCSR ;POKE CLK UP
1919 006306 016703 173404      MOV    RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1920 006312 012700 110325      MOV    #RXERR!PARER!325,R0 ;EXPECTED
1921 006316 012767 000012 172576      MOV    #10,SHIFT ;# OF SHIFTS
1922 006324 012767 001652 173146      MOV    #1652,$TMP1 ;DATA CHAR
1923 006332 004767 010600      JSR    PC,RPOKE ;SHIFT IN THIS CHAR
1924 006336 105777 173350      TSTB  @RXCSR ,RXDONE ?
1925 006342 100401      BMI   +4
1926 006344 104004      ERROR 4 ;RXDONE SHOULD BE ASSERTED
1927 006346 017701 173344      MOV    @RXDBUF,R1 ;ACTUAL
1928 006352 020001      CMP    R0,R1 ;COMPARE EXP VS ACT
1929 006354 001401      BEQ   +4
1930 006356 104000      ERROR ;PARITY ERROR 4 &RXERR SHOULD BE SET
1931                                     ,NOTE THAT THE PARITY BIT SHOULD
1932                                     ,SHOW UP IN THE DATA
1933                                     ,IE BIT SEVEN FOR SEVEN LEVEL CODE
1934
1935                                     ,,THIS TEST VERIFYS EVEPAR PARITY SENSE
1936                                     ,,OF THE RECEIVER
1937                                     ,,MODE ISOC (ISYMOD)
1938                                     ,,PARITY EVEPAR
1939                                     ,,LENGTH: EIGHT PLUS PARITY
1940                                     ,,CHAR: 125
1941                                     ,,
1942                                     ,,*****
1943 006360 000004      TST16: SCOPE
1944 006362 052777 000400 173336      BIS    #MRESET,@TXCSR ;MASTER RESET
1945 006370 012777 000000 173324      MOV    #ISYMOD,@PARCSR ;SET THE MODE
1946 006376 052777 000400 173322      BIS    #MRESET,@TXCSR ;MASTER RESET
1947
1948                                     ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1949 006404 012777 064001 173314      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
1950
1951                                     ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1952 006412 012777 007400 173302      MOV    #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1953 006420 052777 000020 173264      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
1954                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
1955 006426 042777 020000 173272      BIC    #CLK,@TXCSR ;POKE CLK DOWN
1956 006434 052777 020000 173264      BIS    #CLK,@TXCSR ;POKE CLK UP
1957                                     ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1958 006442 042777 020000 173256      BIC    #CLK,@TXCSR ;POKE CLK DOWN
1959 006450 052777 020000 173250      BIS    #CLK,@TXCSR ;POKE CLK UP
```

```
1960 006456 016703 173234      MOV     RXDBUF,R3      ,SET UP FOR ERROR MESSAGE
1961 006462 012700 110125      MOV     #RXERR!PARER!125,R0  ;EXPECTED
1962 006466 012767 000013 172426  MOV     #11,SHIFT      ,# OF SHIFTS
1963 006474 012767 003252 172776  MOV     #3252,$TMP1     ,DATA CHAR
1964 006502 004767 010430      JSR     PC,RPOKE       ,SHIFT IN THIS CHAR
1965 006506 105777 173200      TSTB   @RXCSR ;RXDONE ?
1966 006512 100401      BMI   +4
1967 006514 104004      ERROR  4              ,RXDONE SHOULD BE ASSERTED
1968 006516 017701 173174      MOV     @RXDBUF,R1     ,ACTUAL
1969 006522 020001      CMP     R0,R1         ,COMPARE EXP VS ACT
1970 006524 001401      BEQ    +4
1971 006526 104000      ERROR  ,PARITY ERROR  4 &RXERR SHOULD BE SET
1972
1973      ;, THIS TEST VERIFYS ODDPAR PARITY SENSE
1974      ;, OF THE RECEIVER
1975      ;, MODE: ISOC (ISYMOD)
1976      ;, PARITY: ODDPAR
1977      ;, LENGTH: FIVE PLUS PARITY
1978      ;, CHAR: 65
1979      ;,
1980      ;, *****
1981 006530 000004      TST17  SCOPE
1982 006532 052777 000400 173166  BIS     #MRESET,@TXCSR  ,MASTER RESET
1983 006540 012777 000000 173154  MOV     #ISYMOD,@PARCSR ,SET THE MODE
1984 006546 052777 000400 173152  BIS     #MRESET,@TXCSR  ;MASTER RESET
1985
1986      ;, SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1987 006554 012777 064001 173144  MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
1988
1989      ;, SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
1990 006562 012777 001000 173132  MOV     #ISYMOD!FIVE!ODDPAR!0,@PARCSR
1991 006570 052777 000020 173114  BIS     #SYNSCH,@RXCSR  ;SET SYNC SEARCH
1992      ;, POKE CLK TO GET RECEIVER INTO SYNCRIZATION
1993 006576 042777 020000 173122  BIC     #CLK,@TXCSR    ,POKE CLK DOWN
1994 006604 052777 020000 173114  BIS     #CLK,@TXCSR    ;POKE CLK UP
1995      ;, POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1996 006612 042777 020000 173106  BIC     #CLK,@TXCSR    ;POKE CLK DOWN
1997 006620 052777 020000 173100  BIS     #CLK,@TXCSR    ,POKE CLK UP
1998 006626 016703 173064      MOV     RXDBUF,R3      ,SET UP FOR ERROR MESSAGE
1999 006632 012700 110065      MOV     #RXERR!PARER!65,R0  ;EXPECTED
2000 006636 012767 000010 172256  MOV     #8,SHIFT       ,# OF SHIFTS
2001 006644 012767 000352 172626  MOV     #352,$TMP1     ,DATA CHAR
2002 006652 004767 010260      JSR     PC,RPOKE       ,SHIFT IN THIS CHAR
2003 006656 105777 173030      TSTB   @RXCSR ;RXDONE ?
2004 006662 100401      BMI   +4
2005 006664 104004      ERROR  4              ,RXDONE SHOULD BE ASSERTED
2006 006666 017701 173024      MOV     @RXDBUF,R1     ,ACTUAL
2007 006672 020001      CMP     R0,R1         ,COMPARE EXP VS ACT
2008 006674 001401      BEQ    +4
2009 006676 104000      ERROR  ,PARITY ERROR  4 &RXERR SHOULD BE SET
2010      ;, NOTE THAT THE PARITY BIT SHOULD
2011      ;, SHOW UP IN THE DATA
2012      ;, IE BIT FIVE FOR FIVE LEVEL CODE
2013
2014      ;, THIS TEST VERIFYS ODDPAR PARITY SENSE
2015      ;, OF THE RECEIVER
```

```
2016                                     ;;MODE: ISOC (ISYMOD)
2017                                     ;;PARITY: ODDPAR
2018                                     ;;LENGTH: SIX PLUS PARITY
2019                                     ;;CHAR: 125
2020                                     ;;
2021                                     ;;*****
2022 006700 000004 TST20: SCOPE
2023 006702 052777 000400 173016     BIS #MRESET,@TXCSR ;MASTER RESET
2024 006710 012777 000000 173004     MOV #ISYMOD,@PARCSR ;SET THE MODE
2025 006716 052777 000400 173002     BIS #MRESET,@TXCSR ;MASTER RESET
2026
2027                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2028 006724 012777 064001 172774     MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2029
2030                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2031 006732 012777 003000 172762     MOV #ISYMOD!SIX!ODDPAR!0,@PARCSR
2032 006740 052777 000020 172744     BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2033                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION...
2034 006746 042777 020000 172752     BIC #CLK,@TXCSR ;POKE CLK DOWN
2035 006754 052777 020000 172744     BIS #CLK,@TXCSR ;POKE CLK UP
2036                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2037 006762 042777 020000 172736     BIC #CLK,@TXCSR ;POKE CLK DOWN
2038 006770 052777 020000 172730     BIS #CLK,@TXCSR ;POKE CLK UP
2039 006776 016703 172714             MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2040 007002 012700 110125             MOV #RXERR!PARER!125,RO ;EXPECTED
2041 007006 012767 000011 172106     MOV #9,SHIFT ;# OF SHIFTS
2042 007014 012767 000052 172456     MOV #652,$TMP1 ;DATA CHAR
2043 007022 004767 010110             JSR PC,RPOKE ;SHIFT IN THIS CHAR
2044 007026 105777 172660             TSTB @RXCSR ;RXDONE ?
2045 007032 100401                     BMI .+4
2046 007034 104004                     ERROR 4 ;RXDONE SHOULD BE ASSERTED
2047 007036 017701 172654             MOV @RXDBUF,R1 ;ACTUAL
2048 007042 020001                     CMP RO,R1 ;COMPARE EXP VS. ACT
2049 007044 001401                     BEQ .+4
2050 007046 104000                     ERROR ;PARITY ERROR 4 &RXERR SHOULD BE SET
2051                                     ;NOTE THAT THE PARITY BIT SHOULD
2052                                     ;SHOW UP IN THE DATA
2053                                     ;IE. BIT SIX FOR SIX LEVEL CODE
2054
2055                                     ;;THIS TEST VERIFYS ODDPAR PARITY SENSE
2056                                     ;;OF THE RECEIVER
2057                                     ;;MODE: ISOC (ISYMOD)
2058                                     ;;PARITY: ODDPAR
2059                                     ;;LENGTH: SEVEN PLUS PARITY
2060                                     ;;CHAR: 125
2061                                     ;;
2062                                     ;;*****
2063 007050 000004 TST21: SCOPE
2064 007052 052777 000400 172646     BIS #MRESET,@TXCSR ;MASTER RESET
2065 007060 012777 000000 172634     MOV #ISYMOD,@PARCSR ;SET THE MODE
2066 007066 052777 000400 172632     BIS #MRESET,@TXCSR ;MASTER RESET
2067
2068                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2069 007074 012777 064001 172624     MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2070
2071                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
```

INITIALIZE THE COMMON TAGS

```

2072 007102 012777 005000 172612      MOV      #ISYMOD!SEVEN!ODDPAR!0,@PARCSR
2073 007110 052777 000020 172574      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2074                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2075 007116 042777 020000 172602      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2076 007124 052777 020000 172574      BIS      #CLK,@TXCSR ;POKE CLK UP
2077                                     ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2078 007132 042777 020000 172566      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2079 007140 052777 020000 172560      BIS      #CLK,@TXCSR ;POKE CLK UP
2080 007146 016703 172544      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2081 007152 012700 110125      MOV      #RXERR!PARER!125,RO ;EXPECTED
2082 007156 012767 000012 171736      MOV      #10,SHIFT ;# OF SHIFTS
2083 007164 012767 001252 172306      MOV      #1252,$TMP1 ;DATA CHAR
2084 007172 004767 007740      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2085 007176 105777 172510      TSTB    @RXCSR ,RXDONE ?
2086 007202 100401      BMI     +4
2087 007204 104004      ERROR   4 ;RXDONE SHOULD BE ASSERTED
2088 007206 017701 172504      MOV      @RXDBUF,R1 ;ACTUAL
2089 007212 020001      CMP     RO,R1 ;COMPARE EXP VS ACT
2090 007214 001401      BEQ     +4
2091 007216 104000      ERROR   ,PARITY ERROR 4 &RXERR SHOULD BE SET
2092                                     ,NOTE THAT THE PARITY BIT SHOULD
2093                                     ,SHOW UP IN THE DATA
2094                                     ,IE BIT SEVEN FOR SEVEN LEVEL CODE
2095
2096                                     ,, THIS TEST VERIFYS ODDPAR PARITY SENSE
2097                                     ,, OF THE RECEIVER
2098                                     ,, MODE ISOC (ISYMOD)
2099                                     ,, PARITY ODDPAR
2100                                     ,, LENGTH EIGHT PLUS PARITY
2101                                     ,, CHAR 125
2102                                     ,,
2103                                     ,, *****
2104 007220 000004      TST22   SCOPE
2105 007222 052777 000400 172476      BIS      #MRESET,@TXCSR ;MASTER RESET
2106 007230 012777 000000 172464      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2107 007236 052777 000400 172462      BIS      #MRESET,@TXCSR ;MASTER RESET
2108
2109                                     ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2110 007244 012777 064001 172454      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2111
2112                                     SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2113 007252 012777 007000 172442      MOV      #ISYMOD!EIGHT!ODDPAR!0,@PARCSR
2114 007260 052777 000020 172424      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2115                                     ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2116 007266 042777 020000 172432      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2117 007274 052777 020000 172424      BIS      #CLK,@TXCSR ;POKE CLK UP
2118                                     ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2119 007302 042777 020000 172416      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2120 007310 052777 020000 172410      BIS      #CLK,@TXCSR ;POKE CLK UP
2121 007316 016703 172374      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2122 007322 012700 110125      MOV      #RXERR!PARER!125,RO ;EXPECTED
2123 007326 012767 000013 171566      MOV      #11,SHIFT ;# OF SHIFTS
2124 007334 012767 002252 172136      MOV      #2252,$TMP1 ;DATA CHAR
2125 007342 004767 007570      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2126 007346 105777 172340      TSTB    @RXCSR ,RXDONE ?
2127 007352 100401      BMI     +4

```

```

2128 007354 104004          ERROR 4          ,RXDONE SHOULD BE ASSERTED
2129 007356 017701 172334  MOV    @RXDBUF,R1      ,ACTUAL
2130 007362 020001          CMP    R0,R1          ,COMPARE EXP VS ACT
2131 007364 001401          BEQ    +4
2132 007366 104000          ERROR ;PARITY ERROR 4 &RXERR SHOULD BE SET
2133
2134          ,, THIS TEST PERFORMS BINARY DATA CHECK ON THE
2135          ,, RECEIVER
2136          ,, LENGTH EIGHT PLUS PARITY
2137          ,, MODE ISYMOD
2138          ,, PARITY: EVEPAR
2139          ;;
2140          ,, *****
2141 007370 000004          TST23 SCOPE
2142 007372 052777 000400 172326  BIS    #MRESET,@TXCSR ,MASTER RESET
2143 007400 012777 000000 172314  MOV    #ISYMOD,@PARCSR ,SET THE MODE
2144 007406 052777 000400 172312  BIS    #MRESET,@TXCSR ;MASTER RESET
2145
2146          ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2147 007414 012777 064001 172304  MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2148
2149          ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2150 007422 012777 007400 172272  MOV    #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
2151 007430 052777 000020 172254  BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2152          ,POKE CLK TO GET RECEIVER INTO SYNCRIZATION
2153 007436 042777 020000 172262  BIC    #CLK,@TXCSR    ,POKE CLK DOWN
2154 007444 052777 020000 172254  BIS    #CLK,@TXCSR    ,POKE CLK UP
2155          ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2156 007452 042777 020000 172246  BIC    #CLK,@TXCSR    ,POKE CLK DOWN
2157 007460 052777 020000 172240  BIS    #CLK,@TXCSR    ,POKE CLK UP
2158 007466 016703 172224          MOV    RXDBUF,R3      ,SET UP ERROR MESSAGE
2159 007472 005004          CLR    R4             ,DATA CHAR
2160 007474 010400          15  MOV    R4,R0          ,EXPECTED
2161 007476 012767 000013 171416  MOV    #11 ,SHIFT     ,# OF SHIFTS
2162 007504 010467 171770          MOV    R4,$TMP1       , "TO BE SHIFTED CHARACTER"
2163 007510 004767 007574          JSR    PC,EVEN8       ,CALC PARITY
2164 007514 000241          CLC
2165 007516 006167 171756          ROL    $TMP1          ,GENERATE START BIT
2166 007522 052767 002000 171750  BIS    #BIT10,$TMP1   ,GENERATE STOP BIT
2167          , $TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2168 007530 004767 007402          JSR    PC,RPOKE      ,SHIFT IN THIS CHAR
2169 007534 017701 172156          MOV    @RXDBUF,R1    ,ACTUAL
2170 007540 020001          CMP    R0,R1          ,COMPARE EXP VS ACT
2171 007542 001401          BEQ    +4
2172 007544 104002          ERROR 2             ,DATA CHARS SHOULD MATCH
2173          , THERE SHOULD BE NO PARITY ERPOR
2174 007546 005204          INC    R4             ,UPGRADE NEXT CHAR
2175 007550 105704          TSTB   R4             ,LAST CHAR ?
2176 007552 001350          BNE    15
2177
2178          ,, THIS TEST PERFORMS BINARY DATA CHECK ON THE
2179          ,, RECEIVER
2180          ,, LENGTH EIGHT PLUS PARITY
2181          ,, MODE ISYMOD
2182          ,, PARITY ODDPAR
2183          ,,

```

```
2184 // *****  
2185 007554 000004 TST24 SCOPE  
2186 007556 052777 000400 172142 BIS #MRESET,@TXCSR ,MASTER RESET  
2187 007564 012777 000000 172130 MOV #ISYMOD,@PARCSR ;SET THE MODE  
2188 007572 052777 000400 172126 BIS #MRESET,@TXCSR ;MASTER RESET  
2189  
2190 ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE  
2191 C07600 012777 064001 172120 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR  
2192  
2193 ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG  
2194 007606 012777 007000 172106 MOV #ISYMOD!EIGHT!ODDPAR!O,@PARCSR  
2195 007614 052777 000020 172070 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH  
2196 ,POKE CLK TO GET RECEIVER INTO SYNCRIZATION  
2197 007622 042777 020000 172076 BIC #CLK,@TXCSR ,POKE CLK DOWN  
2198 007630 052777 020000 172070 BIS #CLK,@TXCSR ;POKE CLK UP  
2199 ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION  
2200 007636 042777 020000 172062 BIC #CLK,@TXCSR ,POKE CLK DOWN  
2201 007644 052777 020000 172054 BIS #CLK,@TXCSR ,POKE CLK UP  
2202 007652 016703 172040 MOV RXDBUF,R3 ,SET UP ERROR MESSAGE  
2203 007656 005004 CLR R4 ,DATA CHAR  
2204 007660 010400 15 MOV R4,R0 ,EXPECTED  
2205 007662 012767 000013 171232 MOV #11 ,SHIFT ,# OF SHIFTS  
2206 007670 010467 171604 MOV R4,$TMP1 , "TO BE SHIFTED CHARACTER"  
2207 007674 004767 007324 JSR PC,ODD8 ,CALC PARITY  
2208 007700 000241 CLC  
2209 007702 006167 171572 ROL $TMP1 ,GENERATE START BIT  
2210 007706 052767 002000 171564 BIS #BIT10,$TMP1 ,GENERATE STOP BIT  
2211 STMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER  
2212 007714 004767 007216 JSR PC,RPOKE ,SHIFT IN THIS CHAR  
2213 007720 017701 171772 MOV @RXDBUF,R1 ,ACTUAL  
2214 007724 020001 CMP R0,R1 ,COMPARE EXP VS ACT  
2215 007726 001401 BEQ +4  
2216 007730 104002 ERROR 2 ,DATA CHARS SHOULD MATCH  
2217 ,THERE SHOULD BE NO PARITY ERROR  
2218 007732 005204 INC R4 ,UPGRADE NEXT CHAR  
2219 007734 105704 TSTB R4 ,LAST CHAR ?  
2220 007736 001350 BNE 1$  
2221  
2222 // THIS TEST PERFORMS BINARY DATA CHECK ON THE  
2223 // RECEIVER  
2224 // LENGTH EIGHT PLUS PARITY  
2225 // MODE SYNEXT  
2226 // PARITY EVEPAR  
2227 //  
2228 // *****  
2229 007740 000004 TST25 SCOPE  
2230 007742 052777 000400 171756 BIS #MRESET,@TXCSR ,MASTER RESET  
2231 007750 012777 020000 171744 MOV #SYNEXT,@PARCSR ,SET THE MODE  
2232 007756 052777 000400 171742 BIS #MRESET,@TXCSR ,MASTER RESET  
2233  
2234 ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE  
2235 007764 012777 064001 171734 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR  
2236  
2237 ,SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG  
2238 007772 012777 027400 171722 MOV #SYNEXT!EIGHT!EVEPAR!O,@PARCSR  
2239 010000 052777 000020 171704 BIS #SYNSCH,@RXCSR ,SET SEARCH SYNC
```

INITIALIZE THE COMMON TAGS

```
2240      ,POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2241 010006 042777 020000 171712      BIC  #CLK,@TXCSR      ,POKE CLK DOWN
2242 010014 052777 020000 171704      BIS  #CLK,@TXCSR      ,POKE CLK UP
2243 010022 016703 171670      MOV  RXDBUF,R3        ,SET UP ERROR MESSAGE
2244 010026 005004      CLR  R4              ,DATA CHAR
2245 010030 010400      MOV  R4,R0           ,EXPECTED
2246 010032 012767 000011 171062 1$   MOV  #9,SHIFT        ,# OF SHIFTS
2247 010040 010467 171434      MOV  R4,$TMP1        , "TO BE SHIFTED CHARACTER"
2248 010044 004767 007240      JSR  PC,EVEN8        ,CALC PARITY
2249      , $TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2250 010050 004767 007062      JSR  PC,RPOKE        ,SHIFT IN THIS CHAR
2251 010054 017701 171636      MOV  @RXDBUF,R1      ,ACTUAL
2252 010060 020001      CMP  R0,R1          ,COMPARE EXP VS ACT
2253 010062 001401      BEQ  +4
2254 010064 104002      ERROR 2            ,DATA CHARS SHOULD MATCH
2255      ,THERE SHOULD BE NO PARITY ERROR
2256 010066 005204      INC  R4              ,UPGRADE NEXT CHAR
2257 010070 105704      TSTB R4             ,LAST CHAR ?
2258 010072 001356      BNE  1$
2259
2260      ,, THIS TEST PERFORMS BINARY DATA CHECK ON THE
2261      ,, RECEIVER
2262      ,, LENGTH EIGHT PLUS PARITY
2263      ,, MODE. SYNEXT
2264      ,, PARITY. ODDPAR
2265
2266      ,, *****
2267 010074 000004      TST26 SCOPE
2268 010076 052777 000400 171622      BIS  #MRESET,@TXCSR ,MASTER RESET
2269 010104 012777 020000 171610      MOV  #SYNEXT,@PARCSR ,SET THE MODE
2270 010112 052777 000400 171606      BIS  #MRESET,@TXCSR ,MASTER RESET
2271
2272      ,SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2273 010120 012777 064001 171600      MOV  #MTDATA!CLK!MINT!BREAK,@TXCSR
2274
2275      ,SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2276 010126 012777 027000 171566      MOV  #SYNEXT'EIGHT'ODDPAR'0,@PARCSR
2277 010134 052777 000020 171550      BIS  #SYNSCH,@RXCSR ,SET SEARCH SYNC
2278      ,POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2279 010142 042777 020000 171556      BIC  #CLK,@TXCSR      ,POKE CLK DOWN
2280 010150 052777 020000 171550      BIS  #CLK,@TXCSR      ,POKE CLK UP
2281 010156 016703 171534      MOV  RXDBUF,R3        ,SET UP ERROR MESSAGE
2282 010162 005004      CLR  R4              ,DATA CHAR
2283 010164 010400      MOV  R4,R0           ,EXPECTED
2284 010166 012767 000011 170726 1$   MOV  #9,SHIFT        ,# OF SHIFTS
2285 010174 010467 171300      MOV  R4,$TMP1        , "TO BE SHIFTED CHARACTER"
2286 010200 004767 007020      JSR  PC,ODD8         ,CALC PARITY
2287      $TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2288 010204 004767 006726      JSR  PC,RPOKE        ,SHIFT IN THIS CHAR
2289 010210 017701 171502      MOV  @RXDBUF,R1      ,ACTUAL
2290 010214 020001      CMP  R0,R1          ,COMPARE EXP VS ACT
2291 010216 001401      BEQ  +4
2292 010220 104002      ERROR 2            ,DATA CHARS SHOULD MATCH
2293      ,THERE SHOULD BE NO PARITY ERROR
2294 010222 005204      INC  R4              ,UPGRADE NEXT CHAR
2295 010224 105704      TSTB R4             ,LAST CHAR ?
```

```
2296 010226 001356          BNE      15
2297
2298          ..THIS TEST CHECKS THE STRIP SYNC FUNCTION
2299          ..OF THE RECEIVER LOGIC
2300          ..MODE ISYMOD
2301          ..LENGTH FIVE
2302          ..NOTE RXDONE SHOULD NEVER ASSERT
2303          ..CHAR. 26 (SYNC)
2304          ..
2305          ..*****
2306 010230 000004          TST27  SCOPE
2307 010232 052777 000400 171466          BIS      #MRESET,@TXCSR ;MASTER RESET
2308 010240 012777 000000 171454          MOV      #ISYMOD,@PARCSR ;SET THE MODE
2309 010246 052777 000400 171452          BIS      #MRESET,@TXCSR ;MASTER RESET
2310
2311          .SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2312 010254 012777 064001 171444          MOV      #MTDATA:CLK!MINT!BREAK,@TXCSR
2313
2314          .SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2315 010262 012777 000026 171432          MOV      #ISYMOD'FIVE'NOPAR'26,@PARCSR
2316 010270 052777 000020 171414          BIS      #SYNSCH,@RXCSR ,SET SYNC SEARCH
2317          .POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2318 010276 042777 020000 171422          BIC      #CLK,@TXCSR ,POKE CLK DOWN
2319 010304 052777 020000 171414          BIS      #CLK,@TXCSR ,POKE CLK UP
2320          .POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2321 010312 042777 020000 171406          BIC      #CLK,@TXCSR ,POKE CLK DOWN
2322 010320 052777 020000 171400          BIS      #CLK,@TXCSR ,POKE CLK UP
2323 010326 052777 000400 171356          BIS      #STPSYN,@RXCSR ,SET STRIP SYNC
2324 010334 012767 000003 170562          MOV      #3,COUNT ,# OF SYNC CHARS
2325 010342 012767 000154 171130          15 MOV      #154,$TMP1 ,CHAR TO BE SHIFTED
2326 010350 012767 000007 170544          MOV      #7,SHIFT ,# OF SHIFTS
2327 010356 004767 006554          JSR      PC,RPOKE ,SHIFT IN THIS CHAR
2328 010362 105777 171324          TSTB    @RXCSR ,RXDONE ?
2329 010366 100001          BPL      +4
2330 010370 104004          ERROR   4 ,RXDONE SHOULD NOT BE ASSERTED
2331 010372 005367 170526          DEC     COUNT ,# OF SYNC CHARS
2332 010376 001361          BNE     15
2333
2334          ..THIS TEST CHECKS THE STRIP SYNC FUNCTION
2335          ..OF THE RECEIVER LOGIC
2336          ..MODE ISYMOD
2337          ..LENGTH SIX
2338          ..NOTE RXDONE SHOULD NEVER ASSERT
2339          ..CHAR 26 (SYNC)
2340          ..
2341          ..*****
2342 010400 000004          TST30  SCOPE
2343 010402 052777 000400 171316          BIS      #MRESET,@TXCSR ,MASTER RESET
2344 010410 012777 000000 171304          MOV      #ISYMOD,@PARCSR ,SET THE MODE
2345 010416 052777 000400 171302          BIS      #MRESET,@TXCSR ;MASTER RESET
2346
2347          .SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2348 010424 012777 064001 171274          MOV      #MTDATA:CLK!MINT!BREAK,@TXCSR
2349
2350          .SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2351 010432 012777 002026 171262          MOV      #ISYMOD'SIX'NOPAR'26,@PARCSR
```


INITIALIZE THE COMMON TAGS

```
2408      ;,MODE: ISYMOD
2409      ;,LENGTH: EIGHT
2410      ;,NOTE: RXDONE SHOULD NEVER ASSERT
2411      ;,CHAR: 26 (SYNC)
2412      ;,
2413      ;,*****
2414 010720 000004 TST32. SCOPE
2415 010722 052777 000400 170776   BIS    #MRESET,@TXCSR ;MASTER RESET
2416 010730 012777 000000 170764   MOV    #ISYMOD,@PARCSR ;SET THE MODE
2417 010736 052777 000400 170762   BIS    #MRESET,@TXCSR ;MASTER RESET
2418
2419      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2420 010744 012777 064001 170754   MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2421
2422      ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2423 010752 012777 006026 170742   MOV    #ISYMOD!EIGHT!NOPAR!26,@PARCSR
2424 010760 052777 000020 170724   BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2425      ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION
2426 010766 042777 020000 170732   BIC    #CLK,@TXCSR ;POKE CLK DOWN
2427 010774 052777 020000 170724   BIS    #CLK,@TXCSR ;POKE CLK L
2428      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2429 011002 042777 020000 170716   BIC    #CLK,@TXCSR ;POKE CLK DOWN
2430 011010 052777 020000 170710   BIS    #CLK,@TXCSR ;POKE CLK UP
2431 011016 052777 000400 170666   BIS    #STPSYN,@RXCSR ;SET STRIP SYNC
2432 011024 012767 000003 170072   MOV    #3,COUNT ;# OF SYNC CHARS
2433 011032 012767 001054 170440 1$   MOV    #1054,$TMP1 ;CHAR TO BE SHIFTED
2434 011040 012767 000012 170054   MOV    #10,$SHIFT ;# OF SHIFTS
2435 011046 004767 006064           JSR    PC,RPOKE ;SHIFT IN THIS CHAR
2436 011052 105777 170634           TSTB  @RXCSR ;RXDONE ?
2437 011056 100001           BPL    +4
2438 011060 104004           ERROR  4 ;RXDONE SHOULD NOT BE ASSERTED
2439 011062 005367 170036           DEC    COUNT ;# OF SYNC CHARS
2440 011066 001361           BNE    1$
2441
2442      ;,THIS TEST CHECKS THE STRIP SYNC FUNCTION
2443      ;,OF THE RECEIVER LOGIC
2444      ;,MODE SYNEXT
2445      ;,LENGTH: FIVE
2446      ;,NOTE RXDONE SHOULD NEVER ASSERT
2447      ;,CHAR: 26 (SYNC)
2448      ;,
2449      ;,*****
2450 011070 000004 TST33. SCOPE
2451 011072 052777 000400 170626   BIS    #MRESET,@TXCSR ;MASTER RESET
2452 011100 012777 020000 170614   MOV    #SYNEXT,@PARCSR ;SET THE MODE
2453 011106 052777 000400 170612   BIS    #MRESET,@TXCSR ;MASTER RESET
2454
2455      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2456 011114 012777 064001 170604   MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2457
2458      ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2459 011122 012777 020026 170572   MOV    #SYNEXT!FIVE!NOPAR!26,@PARCSR
2460 011130 052777 000020 170554   BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
2461      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2462 011136 042777 020000 170562   BIC    #CLK,@TXCSR ;POKE CLK DOWN
2463 011144 052777 020000 170554   BIS    #CLK,@TXCSR ;POKE CLK UP
```

```

2464 011152 052777 000400 170532      BIS      #STPSYN,@RXCSR ,SET STRIP SYNC
2465 011160 012767 000003 167736      MOV      #3,COUNT      ,# OF SYNC CHARS
2466 011166 012767 000026 170304 15      MOV      #26,$TMP1     ,CHAR TO BE SHIFTED
2467 011174 012767 000005 167720      MOV      #5,SHIFT      ,# OF SHIFTS
2468 011202 004767 005730      JSR      PC,RPOKE      ,SHIFT IN THIS CHAR
2469 011206 105777 170500      TSTB    @RXCSR ,RXDONE ?
2470 011212 100001      BPL      +4
2471 011214 104004      ERROR   4              ,RXDONE SHOULD NOT BE ASSERTED
2472 011216 005367 167702      DEC     COUNT          ,# OF SYNC CHARS
2473 011222 001361      BNE     15
2474
2475      , , THIS TEST CHECKS THE STRIP SYNC FUNCTION
2476      , , OF THE RECEIVER LOGIC
2477      , , MODE SYNEXT
2478      , , LENGTH SIX
2479      , , NOTE RXDONE SHOULD NEVER ASSERT
2480      , , CHAR 26 (SYNC)
2481      , ,
2482      , , *****
2483 011224 000004      TST34  SCOPE
2484 011226 052777 000400 170472      BIS     #MRESET,@TXCSR ;MASTER RESET
2485 011234 012777 020000 170460      MOV     #SYNEXT,@PARCSR ;SET THE MODE
2486 011242 052777 000400 170456      BIS     #MRESET,@TXCSR ;MASTER RESET
2487
2488      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2489 011250 012777 064001 170450      MOV     #MTDATA!CLK!MINT!BREAK,@TXCSR
2490
2491      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2492 011256 012777 022026 170436      MOV     #SYNEXT!SIX!NOPAR!26,@PARCSR
2493 011264 052777 000020 170420      BIS     #SYNSCH,@RXCSR ,SET SEARCH SYNC
2494
2495      , POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2496 011272 042777 020000 170426      BIC     #CLK,@TXCSP    ;POKE CLK DOWN
2497 011300 052777 020000 170420      BIS     #CLK,@TXCSR    ,POKE CLK UP
2498 011306 052777 000400 170376      BIS     #STPSYN,@RXCSR ,SET STRIP SYNC
2499 011314 012767 000003 167602      MOV     #3,COUNT      ,# OF SYNC CHARS
2500 011322 012767 000026 170150 15      MOV     #26,$TMP1     ,CHAR TO BE SHIFTED
2501 011330 012767 000006 167564      MOV     #6,SHIFT      ,# OF SHIFTS
2502 011336 004767 005574      JSR     PC,RPOKE      ,SHIFT IN THIS CHAR
2503 011342 105777 170344      TSTB    @RXCSR ,RXDONE ?
2504 011346 100001      BPL     +4
2505 011350 104004      ERROR   4              ,RXDONE SHOULD NOT BE ASSERTED
2506 011352 005367 167546      DEC     COUNT          ,# OF SYNC CHARS
2507 011356 001361      BNE     15
2508
2509      , , THIS TEST CHECKS THE STRIP SYNC FUNCTION
2510      , , OF THE RECEIVER LOGIC
2511      , , MODE SYNEXT
2512      , , LENGTH SEVEN
2513      , , NOTE RXDONE SHOULD NEVER ASSERT
2514      , , CHAR 26 (SYNC)
2515      , ,
2516 011360 000004      TST35  SCOPE
2517 011362 052777 000400 170336      BIS     #MRESET,@TXCSR ;MASTER RESET
2518 011370 012777 020000 170324      MOV     #SYNEXT,@PARCSR ;SET THE MODE
2519 011376 052777 000400 170322      BIS     #MRESET,@TXCSR ;MASTER RESET
  
```

```
2520
2521
2522 011404 012777 064001 170314 .SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2523
2524
2525 011412 012777 024026 170302 .SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
      MOV      #SYNEXT!SEVEN!NOPAR!26,@PARCSR
2526 011420 052777 000020 170264      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2527
2528 011426 042777 020000 170272 .POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2529 011434 052777 020000 170264      BIS      #CLK,@TXCSR ;POKE CLK UP
2530 011442 052777 000400 170242      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2531 011450 012767 000003 167446      MOV      #3,COUNT ;# OF SYNC CHARS
2532 011456 012767 000026 170014 1$      MOV      #26,$TMP1 ;CHAR TO BE SHIFTED
2533 011464 012767 000007 167430      MOV      #7,SHIFT ;# OF SHIFTS
2534 011472 004767 005440      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2535 011476 105777 170210      TSTB    @RXCSR ,RXDONE ?
2536 011502 100001      BPL     +4
2537 011504 104004      ERROR   4 ;RXDONE SHOULD NOT BE ASSERTED
2538 011506 005367 167412      DEC     COUNT ;# OF SYNC CHARS
2539 011512 001361      BNE     1$
2540
2541      ,, THIS TEST CHECKS THE STRIP SYNC FUNCTION
2542      ,, OF THE RECEIVER LOGIC
2543      ,, MODE SYNEXT
2544      ,, LENGTH EIGHT
2545      ,, NOTE RXDONE SHOULD NEVER ASSERT
2546      ,, CHAR 26 (SYNC)
2547      ,,
2548      ,, *****
2549 011514 000004      TST36   SCOPE
2550 011516 052777 000400 170202      BIS     #MRESET,@TXCSR ;MASTER RESET
2551 011524 012777 020000 170170      MOV     #SYNEXT,@PARCSR ;SET THE MODE
2552 011532 052777 000400 170166      BIS     #MRESET,@TXCSR ;MASTER RESET
2553
2554
2555 011540 012777 064001 170160 .SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2556
2557
2558 011546 012777 026026 170146 .SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
      MOV      #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2559 011554 052777 000020 170130      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2560
2561 011562 042777 020000 170136 .POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2562 011570 052777 020000 170130      BIS      #CLK,@TXCSR ;POKE CLK UP
2563 011576 052777 000400 170106      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
2564 011604 012767 000003 167312      MOV      #3,COUNT ;# OF SYNC CHARS
2565 011612 012767 000026 167660 1$      MOV      #26,$TMP1 ;CHAR TO BE SHIFTED
2566 011620 012767 000010 167274      MOV      #3,SHIFT ;# OF SHIFTS
2567 011626 004767 005304      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2568 011632 105777 170054      TSTB    @RXCSR ,RXDONE ?
2569 011636 100001      BPL     +4
2570 011640 104004      ERROR   4 ;RXDONE SHOULD NOT BE ASSERTED
2571 011642 005367 167256      DEC     COUNT ;# OF SYNC CHARS
2572 011646 001361      BNE     1$
2573
2574      ,, THIS TEST CHECKS THE STRIP SYNC FUNCTION
2575      ,, OF THE RECEIVER LOGIC
```

```
2576      , , MODE SYNINT
2577      , , LENGTH FIVE
2578      , , NOTE RXDONE SHOULD NEVER ASSERT
2579      , , CHAR 26 (SYNC)
2580      ;
2581      ; ; *****
2582 011650 000004 TST37 SCOPE
2583 011652 052777 000400 170046 BIS #MRESET,@TXCSR ,MASTER RESET
2584 011660 012777 030000 170034 MOV #SYNINT,@PARCSR ;SET THE MODE
2585 011666 052777 000400 170032 BIS #MRESET,@TXCSR ;MASTER RESET
2586
2587      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2588 011674 012777 064001 170024 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2589
2590      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2591 011702 012777 030026 170012 MOV #SYNINT!FIVE!NOPAR!26,@PARCSR
2592 011710 052777 000020 167774 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2593      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2594 011716 042777 020000 170002 BIC #CLK,@TXCSR ;POKE CLK DOWN
2595 011724 052777 020000 167774 BIS #CLK,@TXCSR ;POKE CLK UP
2596      ,POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2597 011732 042777 020000 167766 BIC #CLK,@TXCSR ;POKE CLK DOWN
2598 011740 052777 020000 167760 BIS #CLK,@TXCSR ;POKE CLK UP
2599 011746 052777 000400 167736 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2600 011754 012767 000003 167142 MOV #3,COUNT ;# OF SYNC CHARS
2601 011762 012767 000026 167510 15 MOV #26,STMP1 ;CHAR TO BE SHIFTED
2602 011770 012767 000005 167124 MOV #5,SHIFT ;# OF SHIFTS
2603 011776 004767 005134 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2604 012002 105777 167704 TSTB @RXCSR ;RXDONE ?
2605 012006 100001 BPL .+4
2606 012010 104004 ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2607 012012 005367 167106 DEC COUNT ;# OF SYNC CHARS
2608 012016 001361 BNE 15
2609
2610      , , THIS TEST CHECKS THE STRIP SYNC FUNCTION
2611      , , OF THE RECEIVER LOGIC
2612      , , MODE SYNINT
2613      , , LENGTH SIX
2614      , , NOTE RXDONE SHOULD NEVER ASSERT
2615      , , CHAR 26 (SYNC)
2616      ;
2617      ; ; *****
2618 012020 000004 TST40 SCOPE
2619 012022 052777 000400 167676 BIS #MRESET,@TXCSR ,MASTER RESET
2620 012030 012777 030000 167664 MOV #SYNINT,@PARCSR ,SET THE MODE
2621 012036 052777 000400 167662 BIS #MRESET,@TXCSR ;MASTER RESET
2622
2623      , SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2624 012044 012777 064001 167654 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2625
2626      , SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2627 012052 012777 032026 167642 MOV #SYNINT!SIX!NOPAR!26,@PARCSR
2628 012060 052777 000020 167624 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2629      ,POKE CLK TO GET RECEIVER INTO SYNCROIZATION
2630 012066 042777 020000 167632 BIC #CLK,@TXCSR ;POKE CLK DOWN
2631 012074 052777 020000 167624 BIS #CLK,@TXCSR ;POKE CLK UP
```

INITIALIZE THE COMMON TAGS

```

2632 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2633 012102 042777 020000 167616 BIC #CLK,@TXCSR ;POKE CLK DOWN
2634 012110 052777 020000 167610 BIS #CLK,@TXCSR ;POKE CLK UP
2635 012116 052777 000400 167566 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2636 012124 012767 000003 166772 MOV #3,COUNT ;# OF SYNC CHARS
2637 012132 012767 000026 167340 1$ MOV #26,$TMP1 ;CHAR TO BE SHIFTED
2638 012140 012767 000006 166754 MOV #6,SHIFT ;# OF SHIFTS
2639 012146 004767 004764 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2640 012152 105777 167534 TSTB @RXCSR :RXDONE ?
2641 012156 100001 BPL +4
2642 012160 104004 ERROR 4 ,RXDONE SHOULD NOT BE ASSERTED
2643 012162 005367 166736 DEC COUNT ,# OF SYNC CHARS
2644 012166 001361 BNE 1$

```

```

2645
2646 ;, THIS TEST CHECKS THE STRIP SYNC FUNCTION
2647 ;, OF THE RECEIVER LOGIC
2648 ;, MODE: SYNINT
2649 ;, LENGTH: SEVEN
2650 ;, NOTE: RXDONE SHOULD NEVER ASSERT
2651 ;, CHAR 26 (SYNC)
2652 ;,

```

```

2653 ;, *****
2654 012170 000004 TST41 SCOPE
2655 012172 052777 000400 167526 BIS #MRESET,@TXCSR ,MASTER RESET
2656 012200 012777 030000 167514 MOV #SYNINT,@PARCSR ,SET THE MODE
2657 012206 052777 000400 167512 BIS #MRESET,@TXCSR ;MASTER RESET

```

```

2658
2659 ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2660 012214 012777 064001 167504 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR

```

```

2661
2662 ; SET MODE , # OF BITS, PARITY SENSE, & LOAD SYNC REG
2663 012222 012777 034026 167472 MOV #SYNINT!SEVEN!NOPAR!26,@PARCSR
2664 012230 052777 000020 167454 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2665 ;POKE CLK TO GET RECEIVER INTO SYNCRONIZATION
2666 012236 042777 020000 167462 BIC #CLK,@TXCSR ;POKE CLK DOWN
2667 012244 052777 020000 167454 BIS #CLK,@TXCSR ;POKE CLK UP

```

```

2668 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2669 012252 042777 020000 167446 BIC #CLK,@TXCSR ;POKE CLK DOWN
2670 012260 052777 020000 167440 BIS #CLK,@TXCSR ;POKE CLK UP
2671 012266 052777 000400 167416 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2672 012274 012767 000003 166622 MOV #3,COUNT ;# OF SYNC CHARS
2673 012302 012767 000026 167170 1$ MOV #26,$TMP1 ;CHAR TO BE SHIFTED
2674 012310 012767 000007 166604 MOV #7,SHIFT ;# OF SHIFTS
2675 012316 004767 004614 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2676 012322 105777 167364 TSTB @RXCSR ,RXDONE ?
2677 012326 100001 BPL +4
2678 012330 104004 ERROR 4 ,RXDONE SHOULD NOT BE ASSERTED
2679 012332 005367 166566 DEC COUNT ,# OF SYNC CHARS
2680 012336 001361 BNE 1$

```

```

2681
2682 ; END OF PASS
2683 ; TYPE NAME OF TEST
2684 ; UPDATE PASS COUNT
2685 ; CHECK FOR EXIT TO ACT-11
2686 ; RESTART TEST
2687

```

```

2688
2689 012340 000004          EOP  SCOPE
2690 012342 004767 000340  JSR    PC,CKSWR
2691 012346 104401          TYPE          ,TYPE NAME OF TEST
2692 012350 015476          MEPASS
2693 012352 104413 012604  CONVRT ,OUTCRY
2694 012356 104401 015315  TYPE    ,DEVICE
2695 012362 105767 166564  TSTB   MULTD  ,ARE YOU RUNNING MULTIPLE DEVICES ?
2696 012366 001511          BEQ    CCC    ,NO, JUMP AROUND
2697 012370 005767 166572  TST    ACTREG ;ARE ANY DEVICES ACTIVE ?
2698 012374 001007          BNE    RUNIT  ,YES
2699 012376 104401 015327  TYPE    ,MCOW  ,NO
2700 012402 016700 166560  MOV    ACTREG,RO ,DISPLAY ACTREG
2701 012406 000000          HALT   ,SELECT SOMETHING TO RUN @ ACTREG
2702          ,SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2703 012410 000167 167536  JMP    START  ;START OVER AGAIN... YOU DESELECTED EVERYTHING
2704 012414 062767 000010 166532 RUNIT  ADD    #10,BASEADD ,NEXT BLOCK (ADDRESSES)
2705 012422 062767 000010 166532 ZERO  ADD    #10,BASEIV ,NEXT BLOCK (VECTORS)
2706 012430 000241          CLC
2707 012432 006167 166532  ROL    ROTADD ,UP DATE ROTATING POINTER
2708 012436 103410          BCS   2$     ,IS IT THE LAST DEVICE
2709          ,TO BE TESTED IN THIS PASS ?
2710 012440 036767 166524 166520  BIT    ROTADD,ACTREG ,TEST THIS DEVICE FOR ACTIVE STATUS
2711 012446 001762          BEQ    RUNIT  ,IF NOT ACTIVE, TRY NEXT ADDRESS
2712 012450 004767 000034  JSR    PC,REPLAY ,CALCULATE NEW PARAMETERS
2713 012454 000167 000210  JMP    RESTRT ;YES IT WAS ACTIVE, TEST THIS DEVICE
2714 012460 012767 000J01 166502 2$   MOV    #1,ROTADD ,OK!,NOW SET UP ROTATING
2715          ,POINTER FOR NEXT MULTIPLE PASS
2716 012466 016767 166464 166460  MOV    KEEPADD,BASEADD ,RESTORE BASE ADDRESS
2717 012474 016767 166464 166460  MOV    KEEPIV,BASEIV ,RESTORE BASE INTERRUPT VECTORS
2718 012502 004767 000002  JSR    PC,REPLAY ,CALC NEW PARAMETERS
2719 012506 000441          BR    CCC    ;JUMP AROUND REPLAY
2720 012510 016767 166440 004416  REPLAY MOV    BASEADD,DUBASE ,SET UP FOR NEW ADDRESSES
2721 012516 004767 004260  JSR    PC,DUADDR ,CREATE NEW ADDRESSES
2722 012522 016767 166434 167206  MOV    BASEIV,DURIV ,CREATE DURIV
2723 012530 062767 000002 166424  ADD    #2,BASEIV
2724 012536 016767 166420 167174  MOV    BASEIV,DURIS ,CREATE DURIS
2725 012544 062767 000002 166410  ADD    #2,BASEIV
2726 012552 016767 166404 167162  MOV    BASEIV,DUTIV ,CREATE DUTIV
2727 012560 062767 000002 166374  ADD    #2,BASEIV
2728 012566 016767 166370 167150  MOV    BASEIV,DUTIS ,CREATE DUT S
2729 012574 016767 167136 166360  MOV    DURIV,BASEIV ,RESTORE
2730 012602 000207          RTS    PC
2731
2732 012604 000001          OUTCRY: 1
2733 012606 006 002      . BYTE 6,2
2734 012610 001712          RXCSR
2735
2736          CCC
2737 012612 005067 166564          CLR    $STNM ,CLEAR TEST NUMBER
2738 012616 005067 166574          CLR    $ERRPC ,CLEAR LAST ERROR PC
2739 012622 005067 166555          CLR    $ERFLG ,CLEAR ERROR FLAG
2740 012626 005267 166260          INC    PASCNT ,UPDATE PASS COUNT
2741 012632 016767 166254 166242  MOV    PASCNT,LIGHTS ,DISPLAY PASS COUNT
2742 012640 016767 166246 166666  MOV    PASCNT,$PASS ,PASS COUNT TO APT
2743 012646 013701 000042  MOV    @#42,R1 ,CHECK FOR ACT-11 OR DDP

```

```

2744 012652 001406      BEQ      RESTRT      , IF NO CONTINUE TESTING
2745 012654 000005      RESET
2746 012656 000005      RESET
2747 012660 004711      SENDAD JSR      PC, (R1)
2748 012662 000240      NOP
2749 012664 000240      NOP
2750 012666 000240      NOP
2751 012670      RESTRT
2752 012670 012767 003376 166510  MOV      #TST1+2, $LPADR , LOAD LAST ADDR
2753 012676 004767 000004      JSR      PC, CKSWR
2754 012702 000167 170402      JMP      BEGIN
2755
2756      , CHECK SWITCH REGISTER ROUTINE
2757      , CHECKS TO ALLOW FOR < G > TO ALLOW
2758      , THE CHANGING OF LOCATION 176
2759
2760 012706 005737 000042      CKSWR  TST      @#42
2761 012712 001040      BNE      OUT
2762 012714 022767 000176 166516  CMP      #SWREG, SWR      , SOFTWARE SWR PRESENT?
2763 012722 001034      BNE      OUT      , NO--LEAVE
2764 012724 105777 166514      TSTB    @STKS      , CHECK TTY READY
2765 012730 100031      BPL      OUT      , NO--LEAVE
2766 012732 017767 166510 000422  MOV      @STKB, MSG      , GET CHARACTER
2767 012740 042767 177600 000414  BIC      #177600, MSG      , STRIP JUNK
2768 012746 122767 000007 000406  CMPB    #7, MSG      , IS IT < G > ?
2769 012754 001017      BNE      OUT      , NO
2770 012756 104401 016103      TYPE    , MCNTG
2771 012762 005137 013022      CNTLU  COM      @#RDSW
2772 012766 104401 016113      TYPE    , MMSWR
2773 012772 104413      CONVRT
2774 012774 013024      SWREGL
2775 012776 104406 016124      INSTR, MMNEW
2776 013002 104410      PARAM
2777 013004 000000      0
2778 013006 177777      177777
2779 013010 000176      SWREG
2780 013012 000      001      BYTE    0, 1
2781 013014 005037 013022      OUT    CLR      @#RDSW
2782 013020 000207      RTS      PC
2783 013022 000000      RDSW   WORD    0
2784 013024 000001      SWREGL 1
2785 013026 006      002      BYTE    6, 2
2786 013030 000176      SWREG
2787
2788 013032 000005      5
2789
2790      , CHECK FOR FREEZE ON CURPENT DATA
2791
2792 013034 004767 177646      SCOP1 JSR      PC, CKSWR
2793 013040 032777 001000 166372  BIT      #SW09, @SWR
2794 013046 001402      BEQ      15
2795 013050 016716 166034      MOV      LOCK, (SP)
2796 013054 000002      15    RTI
2797      SBTTL  TYPE ROUTINE
2798
2799      , , *****

```


DZDUS-B MACY11 30(1046) 21-SEP-77 09 12 PAGE 58
 UZDUSB M11 31-MAY-77 09 49 TYPE ROUTINE

SEQ 0056

```

2800 ,*ROUTINE TO TYPE ASCIZ MESSAGE MESSAGE MUST TERMINATE WITH A 0 BYTE
2801 ,*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2802 ,*NOTE1 $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER
2803 ,*NOTE2 $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
2804 ,*NOTE3 $FILLC CONTAINS THE CHARACTER TO FILL AFTER
2805 ,*
2806 ,*CALL
2807 ,*1) USING A TRAP INSTRUCTION
2808 ,* TYPE ,MESADR ,.MESADR IS FIRST ADDRESS OF AN ASCIZ STPING
2809 ,*OR
2810 ,* TYPE
2811 ,* MESADR
2812 ,*
2813
2814 013056 105767 166375 $TYPE TSTB $TPFLG ,.IS THERE A TERMINAL?
2815 013062 100002 BPL 1$ ,.BR IF YES
2816 013064 000000 HALT ,.HALT HERE IF NO TERMINAL
2817 013066 000430 BR 3$ ,.LEAVE
2818 013070 010046 1$ MOV RO,-(SP) ,.SAVE RO
2819 013072 017600 000002 MOV @2(SP),RO ,.GET ADDRESS OF ASCIZ STRING
2820 013076 122767 000001 166442 CMPB #APTENV,$ENV ,.RUNNING IN APT MODE
2821 013104 001011 BNE 62$ ,.NO,GO CHECK FOR APT CONSOLE
2822 013106 132767 000100 166433 BITB #APTPOOL,$ENVM ,.SPOOL MESSAGE TO APT
2823 013114 001405 BEQ 62$ ,.NO,GO CHECK FOR CONSOLE
2824 013116 010067 000004 MOV RO,61$ ,.SETUP MESSAGE ADDRESS FOR APT
2825 013122 004767 000006 JSR PC,$ATY3 ,.SPOOL MESSAGE TO APT
2826 013126 000000 61$ WORD 0 ,.MESSAGE ADDRESS
2827 013130 132767 000040 166411 62$ BITB #APTCSUP,$ENVM ,.APT CONSOLE SUPPRESSED
2828 013136 001003 BNE 60$ ,.YES,SKIP TYPE OUT
2829 013140 112046 2$ MOVB (RO)+,-(SP) ,.PUSH CHARACTER TO BE TYPED ONTO STACK
2830 013142 001005 BNE 4$ ,.BR IF IT ISN'T THE TERMINATOR
2831 013144 005726 TST (SP)+ ,.IF TERMINATOR POP IT OFF THE STACK
2832 013146 012600 60$ MOV (SP)+,RO ,.RESTORE RO
2833 013150 062716 000002 3$ ADD #2,(SP) ,.ADJUST RETURN PC
2834 013154 000002 RTI ,.RETURN
2835 013156 122716 000011 4$ CMPB #HT,(SP) ,.BRANCH IF <HT>
2836 013162 001430 BEQ 8$
2837 013164 122716 000200 CMPB #CRLF,(SP) ,.BRANCH IF NOT <CRLF>
2838 013170 001006 BNE 5$
2839 013172 005726 TST (SP)+ ,.POP <CR><LF> EQUIV
2840 013174 104401 TYPE ,.TYPE A CR AND LF
2841 013176 001523 $CRLF
2842 013200 105067 000130 CLRB $CHARCNT ,.CLEAR CHARACTER COUNT
2843 013204 000755 BR 2$ ,.GET NEXT CHARACTER
2844 013206 004767 000056 5$ JSR PC,$TYPEC ,.GO TYPE THIS CHARACTER
2845 013212 126726 166240 6$ CMPB $FILLC,(SP)+ ,.IS IT TIME FOR FILLER CHARS ?
2846 013216 001350 BNE 2$ ,.IF NO GO GET NEXT CHAR
2847 013220 016746 166230 MOV $NULL,-(SP) ,.GET # OF FILLER CHARS NEEDED
2848 ,.AND THE NULL CHAR
2849 013224 105366 000001 7$ DECB 1(SP) ,.DOES A NULL NEED TO BE TYPED?
2850 013230 002770 BLT 6$ ,.BR IF NO--GO POP THE NULL OFF OF STACK
2851 013232 004767 000032 JSR PC,$TYPEC ,.GO TYPE A NULL
2852 013236 105367 000072 DECB $CHARCNT ,.DO NOT COUNT AS A COUNT
2853 013242 000770 BR 7$ ,.LOOP
2854
2855 ,HORIZONTAL TAB PROCESSOR

```

```

2856
2857 013244 112716 000040      8$  MOV#  #' , (SP)      ;, REPLACE TAB WITH SPACE
2858 013250 004767 000014      9$  JSR   PC, $TYPEC    ;, TYPE A SPACE
2859 013254 132767 000007 000052  BIT#  #7, $CHARCNT   ;, BRANCH IF NOT AT
2860 013262 001372                BNE   9$             ;, TAB STOP
2861 013264 005726                TST   (SP)+         ;, POP SPACE OFF STACK
2862 013266 000724                BR    2$             ;, GET NEXT CHARACTER
2863 013270 105777 166154      $TYPEC TST#  @ $STPS    ;, WAIT UNTIL PRINTER IS READY
2864 013274 100375                BPL   $TYPEC
2865 013276 116677 000002 166146  MOV#  2(SP), @ $STPB ;, LOAD CHAR TO BE TYPED INTO DATA REG
2866 013304 122766 000015 000002  CMP#  #CR, 2(SP)    ;, IS CHARACTER A CARRIAGE RETURN?
2867 013312 001003                BNE   1$             ;, BRANCH IF NO
2868 013314 105067 000014      CLRB  $CHARCNT      ;, YES--CLEAR CHARACTER COUNT
2869 013320 000406                BR    $TYPEX        ;, EXIT
2870 013322 122766 000012 000002  1$  CMP#  #LF, 2(SP)    ;, IS CHARACTER A LINE FEED?
2871 013330 001402                BEQ   $TYPEX        ;, BRANCH IF YES
2872 013332 105227                INCB  (PC)+         ;, COUNT THE CHARACTER
2873 013334 000000      $CHARCNT WORD 0    ;, CHARACTER COUNT STORAGE
2874 013336 000207      $TYPEX RTS          PC
2875
2876
2877                                ; ASCII STRING INPUT ROUTINE
2878
2879 013340 017667 000000 000014  INSTR MOV  @ (SP), MSG    ;, PICK UP MESSAGE
2880 013346 062716 000002          ADD  #2, (SP)        ;, JUMP AROUND MESSAGE FOR RTI
2881 013352 105767 166170          TST#  $ENV          ;, APT CONTROL
2882 013356 001036          BNE   INSTR2        ;, YES NO TYPE
2883 013360 104401          INSTR1 TYPE
2884 013362 000000      MSG  0
2885 013364 012704 016136          MOV  #INBUF, R4     ;, GET STARTING LOC OF INBUF
2886 013370 012703 000007          MOV  #7, R3         ;, MAX # OF CHARS
2887 013374 105777 166044      1$  TST#  @ $TKS , TTY FLAG
2888 013400 100375          BPL   1$
2889 013402 117714 166040          MOV#  @ $TKB, (R4)   ;, TAKE CHAR
2890 013406 142714 000200          BIC#  #200, (R4)    ;, STRIP
2891 013412 121427 000025          CMP#  (R4), #25     ;, IS IT < G >
2892 013416 001760          BEQ   INSTR1
2893 013420 122427 000015          CMP#  (R4)+, #15    ;, CHECK FOR CR
2894 013424 001413          BEQ   INSTR2
2895 013426 105777 166016      2$  TST#  @ $TPS , TEST FLAG
2896 013432 100375          BPL   2$
2897 013434 117777 166006 166010  MOV#  @ $TKB, @ $TPB ;, ECHO CHARACTER
2898 013442 005303          DEC  R3             ;, DID YOU TYPE TOO MANY CHARS ?
2899 013444 001353          BNE   1$
2900 013446 104401          INSTR1 TYPE
2901 013450 015423          MQM  , ?
2902 013452 000742          BR   INSTR1 , RETRY
2903 013454 000002      INSTR2 RTI
2904
2905                                ; CONVERT ASCII STRING TO OCTAL
2906
2907 013456 011605      PARAM MOV  (SP), R5 , PUT CONTENTS OF SP INTO R5
2908 013460 012567 000162          MOV  (R5)+, LOLIM  ;, PUT LOW LIMIT INTO LOLIM
2909 013464 012567 000160          MOV  (R5)+, HILIM  ;, PUT HIGH LIMIT INTO HILIM
2910 013470 012567 000156          MOV  (R5)+, DEVADR  ;, PUT STORE LOC INTO DEVADR
2911 013474 112567 000154          MOV#  (R5)+, LOBITS ;, PUT MASK INTO LOBITS
  
```

DZDUS-B MACY11 30(1046) 21-SEP-77 09 12 PAGE 60
 UZDUSB M11 31-MAY-77 09 49 TYPE ROUTINE

SEQ 0058

```

2912 013500 112567 000151          MOVB   (R5)+,ADRCNT      ,PUT COUNT INTO ADRCNT
2913 013504 010516          MOV    R5,(SP)      ,RESTORE RETURN ADDR ON STACK FOR RTI
2914 013506 005005          PARAM1 CLR    R5
2915 013510 012704 016136          MOV    #INBUF,R4
2916 013514 122714 000015          CMPB   #15,(R4)      ,CR ?
2917 013520 001420          BEQ    PARERR      ,YOU TYPED CR TOO SOON !
2918 013522 121427 000060          1$    CMPB   (R4),#60      ;LOW LIMIT ASCII 0
2919 013526 002415          BLT    PARERR
2920 013530 121427 000067          CMPB   (R4),#67      ,HIGH LIMIT ASCII 7
2921 013534 003012          BGT    PARERR
2922 013536 142714 000060          BICB   #60,(R4)      ;CONVERT TO OCTAL
2923 013542 152405          BISB   (R4)+,R5      ,STORE AWAY ITS AN OK CHAR
2924 013544 122714 000015          CMPB   #15,(R4)      ,CR ?
2925 013550 001414          BEQ    LIMITS      ,NOW CHECK FOR HIGH & LOW LIM T CONDS
2926 013552 006305          ASL    R5          ,ALLOCATE ROOM FOR NEXT CHAR
2927 013554 006305          ASL    R5
2928 013556 006305          ASL    R5
2929 013560 000760          BR     1$
2930 013562 122714 000015          PARERR CMPB   #15,(R4)      ,CR?
2931 013566 001003          BNE    120$
2932 013570 005737 013022          TST    @#RDSW      ,CK SWP USED
2933 013574 001023          BNE    PARTI
2934 013576 104407          120$  INSTER ,RETRY
2935 013600 000742          BR     PARAM1
2936
2937          ,TEST TO SEE IF NUMBER IS WITHIN LIMITS
2938
2939 013602 020567 000042          LIMITS CMP    R5,HILIM
2940 013606 101365          BHI    PARERR      ,THE # IS TOO HIGH
2941 013610 020567 000032          CMP    R5,LOLIM
2942 013614 103762          BLO    PARERR      ;THE # IS TOO LOW
2943 013616 136705 000032          BITB   LOBITS,R5      ,TEST BY MASKINGTHE #
2944 013622 001357          BNE    PARERR
2945
2946          ,STORE NUMBER AT SPECIFIED ADDRESS
2947
2948 013624 016704 000022          1$    MOV    DEVADR,R4      ,GET STARTING ADDR OF
2949 013630 010524          MOV    R5,(R4)+      ,STORE AT THIS ADDR
2950 013632 062705 000002          ADD    #2,R5
2951 013636 105367 000013          DECB   ADRCNT      ,HOW MANY TIMES + 2 ?
2952 013642 001372          BNE    1$
2953 013644 000002          PARTI RTI
2954 013646 000000          LOLIM 0
2955 013650 000000          HILIM 0
2956 013652 000000          DEVADR 0
2957 013654 000000          LOBITS 0
2958          ADRCNT=LOBITS+1
2959
2960          ,SAVE PC OF TEST THAT FAILED AND R0-R5
2961
2962 013656 016667 000004 165242  SAV05 MOV    4(SP),SAVPC
2963
2964          ,SAVE R0-R5
2965
2966 013664 010567 165604  SAV05 MOV    R5,$REG5
2967 013670 010467 165576          MOV    R4,$REG4

```

```

2968 013674 010367 165570      MOV      R3,$REG3
2969 013700 010267 165562      MOV      R2,$REG2
2970 013704 010167 165554      MOV      R1,$REG1
2971 013710 010067 165546      MOV      R0,$REG0
2972 013714 000002      RTI
2973
2974      ,RESTORE R0-R5
2975
2976 013716 016700 165540      RES05   MOV      $REG0,R0
2977 013722 016701 165535      MOV      $REG1,R1
2978 013726 016702 165534      MOV      $REG2,R2
2979 013732 016703 165532      MOV      $REG3,R3
2980 013736 016704 165530      MOV      $REG4,R4
2981 013742 016705 165526      MOV      $REG5,R5
2982 013746 000002      RTI
2983
2984      ,CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2985
2986 013750 104401      CONVR   TYPE
2987 013752 015427      MCPLF   ,CR LF
2988 013754 017601 000000      MOV      @($P),R1      ,PICK UP DATA POINTER
2989 013760 062716 000002      ADD      #2,$P      ,SET UP $P FOR PTI
2990 013764 012167 000130      MOV      (R1)+,WRDCNT   ,PICK UP # OF WORDS FROM TABLE
2991 013770 112167 000126      1$     MOV      (R1)+,CHRCNT   ,PICK UP # OF CHARS FROM TABLE
2992 013774 112167 000123      MOV      (R1)+,SPACNT   ,PICK UP # OF SPACES FROM TABLE
2993 014000 013167 000120      MOV      @($R1)+,BINWRD ,PICK UP ADDRESS OF MSG
2994      ,FROM TABLE
2995 014004 016704 000114      2$     MOV      BINWRD,R4      ,SAVE
2996 014010 116705 000106      MOV      CHRCNT,R5      ,SAVE
2997 014014 012700 016200      MOV      #TEMP,R0      ,STARTING ADDRESS OF TEMP BLOCK
2998 014020 010403 3$     MOV      R4,R3      ,SAVE
2999 014022 042703 177770      BIC      #177770,R3     ,CLR OUT UPPER BITS      SAVE CHAR
3000 014026 062703 000260      ADD      #260,R3      ,CONVERT TO ASCII
3001 014032 110320      MOV      R3,(R0)+      ,STORE AWAY
3002 014034 006204      ASR      R4      ,SHIFT FOR NEXT #
3003 014036 006204      ASR      R4      ,DITTO
3004 014040 006204      ASR      R4      ,DITTO
3005 014042 005305      DEC      R5      ,DEC CHAR COUNT
3006 014044 001365      BNE      3$      ,DO IT AGAIN ?
3007 014046 012703 016242      MOV      #MDATA,R3     ,STARTING ADDRESS OF MDATA BLOCK
3008 014052 114023 4$     MOV      -(R0),(R3)+    ,REVERSE THE ORDER OF NUMBERS
3009 014054 105367 000042      DECB     CHRCNT      ,DEC CHAR COUNT
3010 014060 001374      BNE      4$      ,DO IT AGAIN ?
3011 014062 105767 000035      TSTB     SPACNT      ,HOW MANY SPACES ?
3012 014066 001405      BEQ      6$      ,TYPE # IF BR =0
3013 014070 112723 000240      5$     MOV      #240,(R3)+    , "SPACE" IN ASCII
3014 014074 105367 000023      DECB     SPACNT      ,DEC # OF SPACE COUNT
3015 014100 001373      BNE      5$      ,DO IT AGAIN ?
3016 014102 105013 6$     CLRB     (R3)      ,INSERT "0" FOR TTY OUTPUT ROUTINE
3017 014104 104401      TYPE
3018 014106 016242      MDATA   ,THIS MESSAGE
3019 014110 005367 000004      DEC      WRDCNT      ,HOW MANY #'S ?
3020 014114 001325      BNE      1$      ,DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3021 014116 000002      RTI      ,RETURN TO PROGRAM
3022 014120 000000      WPCNT   0
3023 014122 000000      CHPCNT  0
    
```

```

3024          014123          SPACNT=CHRCNT+1
3025 014124 000000          BINWRD 0
3026
3027          ,COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3028          ,BUFFER TO THE CHARACTERS "N" AND "Y"
3029          ,IF THE CHARACTER IS "N" CLEAR THE FLAG
3030          ,IF THE CHARACTER IS "Y" SET THE FLAG
3031
3032 014126 017605 000000          SETFLG MOV @ (SP),R5
3033 014132 122767 000115 001776          CMPB #'N,INBUF ,IS IT "N" ?
3034 014140 001002          BNE 1$
3035 014142 105015          CLRB (R5) ,000
3036 014144 000406          BR 2$
3037 014146 122767 000131 001762 1$          CMPB #'Y,INBUF ,IS IT "Y" ?
3038 014154 001005          BNE 3$
3039 014156 112715 177777          MOVB #-1,(R5) ,377
3040 014162 062716 000002          2$ ADD #2,(SP)
3041 014166 000002          RTI
3042 014170 104407          3$ INSTER ,RETRY
3043 014172 000755          BR SETFLG
3044          SBTTL ERROR HANDLER ROUTINE
3045
3046          ,*****
3047          ,*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3048          ,*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3049          ,*AND GO TO SAVIT ON ERROR
3050          ,*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE
3051          ,*SW15=1 HALT ON ERROR
3052          ,*SW13=1 INHIBIT ERROR TYPEOUTS
3053          ,*SW10=1 BELL ON ERROR
3054          ,*SW09=1 LOOP ON ERROR
3055          ,*CALL
3056          ,* ERROR N , ,ERROR=EMT AND N=ERROR ITEM NUMBER
3057
3058          SERROR
3059 014174 105267 165203          7$ INCB SERRFLG , ,SET THE ERROR FLAG
3060 014200 001775          BEQ 7$ , ,DON'T LET THE FLAG GO TO ZERO
3061 014202 016777 165174 165232          MOV $STNM,@DISPLAY , ,DISPLAY TEST NUMBER AND ERPOR FLAG
3062 014210 032777 002000 165222          BIT #BIT10,@SWR , ,BELL ON ERROR?
3063 014216 001402          BEQ 1$ , ,NO - SKIP
3064 014220 104401 001516          TYPE ,SBELL , ,RING BELL
3065 014224 005267 165162          1$ INC SERTTL , ,COUNT THE NUMBER OF ERRORS
3066 014230 011667 165162          MOV (SP),SERRPC , ,GET ADDRESS OF ERROR INSTRUCTION
3067 014234 162767 000002 165154          SUB #2,SERRPC
3068 014242 117767 165150 165144          MOVB @SERRPC,$ITEMB , ,STRIP AND SAVE THE ERROR ITEM CODE
3069 014250 032777 020000 165162          BIT #BIT13,@SWR , ,SKIP TYPEOUT IF SET
3070 014256 001004          BNE 20$ , ,SKIP TYPEOUTS
3071 014260 004767 000072          JSR PC,SAVIT , ,GO TO USER ERROR ROUTINE
3072 014264 104401 001523          TYPE ,SCLF
3073 014270          20$
3074 014270 122767 000001 165250          CMPB #APTENV,$ENV , ,RUNNING IN APT MODE
3075 014276 001007          BNE 2$ , ,NO,SKIP APT ERROR REPORT
3076 014300 116767 165110 000004          MOVB $ITEMB,21$ , ,SET ITEM NUMBER AS ERROR NUMBER
3077 014306 004767 000016          JSR PC,$SATY4 , ,REPORT FATAL ERROR TO APT
3078 014312 000          21$ BYTE 0
3079 014313 000          BYTE 0

```

```

3080 014314 000777          22$ BR 22$          ;;APT ERROR LOOP
3081 014316 005777 165116  2$  TST @SWR          ;;HALT ON ERROR
3082 014322 100001          3$  BPL 3$          ;;SKIP IF CONTINUE
3083 014324 000000          HALT          ;;HALT ON ERROR'
3084 014326 032777 001000 165104 3$ BIT #BIT09,@SWR  ;;LOOP ON ERROR SWITCH SET?
3085 014334 001402          4$  BEQ 4$          ;;BR IF NO
3086 014336 016716 165046  4$  MOV $LPERR,(SP)  ;;FUDGE RETURN FOR LOOPING
3087 014342 005767 165146  4$  TST $ESCAPE      ;;CHECK FOR AN ESCAPE ADDRESS
3088 014346 001402          5$  BEQ 5$          ;;BR IF NONE
3089 014350 016716 165140  5$  MOV $ESCAPE,(SP)  ;;FUDGE RETURN ADDRESS FOR ESCAPE
3090 014354          5$
3091 014354 000002          RTI          ;;RETURN
3092 014356 010067 164546  SAVIT MOV R0,HL00
3093 014362 010167 164544  MOV R1,HL01
3094 014366 010267 164542  MOV R2,HL02
3095 014372 010367 164540  MOV R3,HL03
3096 014376 010467 164536  MOV R4,HL04
3097 014402 010567 164534  MOV R5,HL05
3098 014406 016767 164770 164530 MOV $TSTNM,HL06
3099
3100          SBTTL ERROR MESSAGE TYPEOUT ROUTINE
3101
3102          ;;*****
3103          ;;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
3104          ;;*ERROR IS TO BE REPORTED IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
3105          ;;*AND REPORTS THE APPROPRIATE INFORMATION CONCERN'ING THE ERROR
3106
3107          SERRTYP
3108 014414 104401 001523  TYPE , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
3109 014420 010046  MOV RO,-(SP)          ;; SAVE RO
3110 014422 005000  CLR RO          ;; PICKUP THE ITEM INDEX
3111 014424 153700 001414  BISB @#$ITEMB,RO
3112 014430 001004  BNE 1$          ;; IF ITEM NUMBER IS ZERO, JUST
3113          ;; TYPE THE PC OF THE ERROR
3114 014432 016746 164760  MOV $ERRPC,-(SP)  ;; SAVE $ERRPC FOR TYPEOUT
3115          ;; ERROR ADDRESS
3116 014436 104402  TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
3117 014440 000426  BR 6$          ;; GET OUT
3118 014442 005300 1$  DEC RO          ;; ADJUST THE INDEX SO THAT IT WILL
3119 014444 006300  ASL RO          ;; WORK FOR THE ERROR TABLE
3120 014446 006300  ASL RO
3121 014450 006300  ASL RO
3122 014452 062700 001652  ADD #$ERRTB,RO  ;; FORM TABLE POINTER
3123 014456 012067 000004  MOV (RO)+,2$  ;; PICKUP "ERROR MESSAGE" POINTER
3124 014462 001404  BEQ 3$          ;; SKIP TYPEOUT IF NO POINTER
3125 014464 104401  TYPE          ;; TYPE THE "ERROR MESSAGE"
3126 014466 000000 2$  WORD 0          ;; "ERROR MESSAGE" POINTER GOES HERE
3127 014470 104401 001523  TYPE , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
3128 014474 012067 000004 3$  MOV (RO)+,4$  ;; PICKUP "DATA HEADER" POINTER
3129 014500 001404  BEQ 5$          ;; SKIP TYPEOUT IF 0
3130 014502 104401  TYPE          ;; TYPE THE "DATA HEADER"
3131 014504 000000 4$  WORD 0          ;; "DATA HEADER" POINTER GOES HERE
3132 014506 104401 001523  TYPE , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
3133 014512 011000 5$  MOV (RO),RO  ;; PICKUP "DATA TABLE" POINTER
3134 014514 001004  BNE 7$          ;; GO TYPE THE DATA
3135 014516 012600 6$  MOV (SP)+,RO  ;; RESTORE RO

```

```

3136 014520 104401 001523          TYPE  , $CRLF          ;; "CARRIAGE RETURN" & "LINE FEED"
3137 014524 000207          RTS      PC              ;; RETURN
3138 014526          7$          MOV      @ (RO)+, -(SP)    ;; SAVE @ (RO)+ FOR TYPEOUT
3139 014526 013046          TYPOC          ;; GO TYPE--OCTAL ASCII (ALL DIGITS)
3140 014530 104402          TST      (RO)          ;; IS THERE ANOTHER NUMBER?
3141 014532 005710          BEQ      6$              ;; BR IF NO
3142 014534 001770          TYPE  , 8$              ;; TYPE TWO(2) SPACES
3143 014536 104401 014544          BR      7$              ;; LOOP
3144 014542 000771          ASCIZ  / /              ;; TWO(2) SPACES
3145 014544 020040 000          EVEN
3146          014550          SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3147
3148
3149          ;; *****
3150          ;; THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3151          ;; OCTAL (ASCII) NUMBER AND TYPE IT.
3152          ;; $TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3153          ;; $CALL
3154          ;;      MOV      NUM, -(SP)          ;; NUMBER TO BE TYPED
3155          ;;      TYPOS          ;; CALL FOR TYPEOUT
3156          ;;      BYTE  N          ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3157          ;;      BYTE  M          ;; M=1 OR 0
3158          ;;
3159          ;;          ;; 1=TYPE LEADING ZEROS
3160          ;;          ;; 0=SUPPRESS LEADING ZEROS
3161          ;;
3162          ;; $STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3163          ;; $TYPOS OR $TYPOC
3164          ;; $CALL
3165          ;;      MOV      NUM, -(SP)          ;; NUMBER TO BE TYPED
3166          ;;      TYPON          ;; CALL FOR TYPEOUT
3167          ;;
3168          ;; $TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3169          ;; $CALL
3170          ;;      MOV      NUM, -(SP)          ;; NUMBER TO BE TYPED
3171          ;;      TYPOC          ;; CALL FOR TYPEOUT
3172 014550 017646 000000 000000 000211 $TYPOS MOV      @ (SP), -(SP)    ;; PICKUP THE MODE
3173 014554 116667 000001          MOVVB   1 (SP), $OFILL    ;; LOAD ZERO FILL SWITCH
3174 014562 112667 000207          MOVVB   (SP)+, $OMODE+1  ;; NUMBER OF DIGITS TO TYPE
3175 014566 062716 000002          ADD     #2, (SP)        ;; ADJUST RETURN ADDRESS
3176 014572 000406          BR      $TYPON
3177 014574 112767 000001 000171 $TYPOC. MOVVB   #1, $OFILL    ;; SET THE ZERO FILL SWITCH
3178 014602 112767 000006 000165          MOVVB   #6, $OMODE+1  ;; SET FOR SIX(6) DIGITS
3179 014610 112767 000005 000154 $TYPON MCVB   #5, $OCNT    ;; SET THE ITERATION COUNT
3180 014616 010346          MOV     R3, -(SP)      ;; SAVE R3
3181 014620 010446          MOV     R4, -(SP)      ;; SAVE R4
3182 014622 010546          MOV     R5, -(SP)      ;; SAVE R5
3183 014624 116704 000145          MOVVB   $OMODE+1, R4   ;; GET THE NUMBER OF DIGITS TO TYPE
3184 014630 005404          NEG     R4              ;;
3185 014632 062704 000006          ADD     #6, R4          ;; SUBTRACT IT FOR MAX ALLOWED
3186 014636 110467 000132          MOVVB   R4, $OMODE    ;; SAVE IT FOR USE
3187 014642 116704 000125          MOVVB   $OFILL, R4    ;; GET THE ZERO FILL SWITCH
3188 014646 016605 000012          MOV     12 (SP), R5   ;; PICKUP THE INPUT NUMBER
3189 014652 005003          CLR     R3              ;; CLEAR THE OUTPUT WORD
3190 014654 006105          1$      ROL     R5      ;; ROTATE MSB INTO "C"
3191 014656 000404          BR      3$              ;; GO DO MSB

```

```

3192 014660 006105      2$    ROL    R5      ,,FORM THIS DIGIT
3193 014662 006105      ROL    R5
3194 014664 006105      ROL    R5
3195 014666 010503      MOV    R5,R3
3196 014670 006103      3$    ROL    R3      ,,GET LSB OF THIS DIGIT
3197 014672 105367 000076  DECB  $OMODE  ,,TYPE THIS DIGIT?
3198 014676 100016      BPL    7$      ,,BR IF NO
3199 014700 042703 177770  BIC    #177770,R3  ,,GET RID OF JUNK
3200 014704 001002      BNE    4$      ,,TEST FOR 0
3201 014706 005704      TST    R4      ,,SUPPRESS THIS 0?
3202 014710 001403      BEQ    5$      ,,BR IF YES
3203 014712 005204      4$    INC    R4      ,,DON'T SUPPRESS ANYMORE 0'S
3204 014714 052703 000060  BIS    #'0,R3    ,,MAKE THIS DIGIT ASCII
3205 014720 052703 000040  5$    BIS    #' ,R3  ,,MAKE ASCII IF NOT ALREADY
3206 014724 110367 000040  MOV    R3,8$    ,,SAVE FOR TYPING
3207 014730 104401 014770  TYPE  ,8$      ,,GO TYPE THIS DIGIT
3208 014734 105367 000032  7$    DECB  $OCNT    ,,COUNT BY 1
3209 014740 003347      BGT    2$      ,,BR IF MORE TO DO
3210 014742 002402      BLT    6$      ,,BR IF DONE
3211 014744 005204      INC    R4      ,,INSURE LAST DIGIT ISN'T A BLANK
3212 014746 000744      BR     2$      ,,GO DO THE LAST DIGIT
3213 014750 012605      6$    MOV    (SP)+,R5  ,,RESTORE R5
3214 014752 012604      MOV    (SP)+,R4  ,,RESTORE R4
3215 014754 012603      MOV    (SP)+,R3  ,,RESTORE R3
3216 014756 016666 000002 000004  MOV    2(SP),4(SP)  ,,SET THE STACK FOR RETURNING
3217 014764 012616      MOV    (SP)+,(SP)
3218 014766 000002      RTI     ,,RETURN
3219 014770      000      8$    BYTE  0      ,,STORAGE FOR ASCII DIGIT
3220 014771      000      BYTE  0      ,,TERMINATOR FOR TYPE ROUTINE
3221 014772      000      $OCNT  BYTE  0      ,,OCTAL DIGIT COUNTER
3222 014773      000      $OFILL BYTE  0      ,,ZERO FILL SWITCH
3223 014774 000000      $OMODE WORD  0      ,,NUMBER OF DIGITS TO TYPE
3224      ,ENTER HERE ON POWER FAILURE
3225
3226
3227 014776      SPWRDN
3228 014776 010046      PFAIL  MOV    R0,-(SP)  ,,SAVE R0-R5 ON PPROCESSOR STACK
3229 015000 010146      MOV    R1,-(SP)
3230 015002 010246      MOV    R2,-(SP)
3231 015004 010346      MOV    R3,-(SP)
3232 015006 010446      MOV    R4,-(SP)
3233 015010 010546      MOV    R5,-(SP)
3234 015012 016746 163006  MOV    24,-(SP)
3235 015016 010667 164074  MOV    SP,SAVSP  ,,SAVE STACK POINTER
3236 015022 012767 015034 162774  MOV    #RESTART,24  ,,SET UP FOR POWER UP TRAP
3237 015030 000000      HALT   ,,HALT ON POWER DOWN NORIAL
3238 015032 000777      BR
3239
3240      ,PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3241
3242 015034 016706 164056  RESTAR MOV    SAVSP,SP  ,,RESTORE STACK POINTER
3243 015040 012605      MOV    (SP)+,R5  ,,RESTORE R0-R5
3244 015042 012604      MOV    (SP)+,R4
3245 015044 012603      MOV    (SP)+,R3
3246 015046 012602      MOV    (SP)+,R2
3247 015050 012601      MOV    (SP)+,R1

```


3248	015052	012600				MOV	(SP)+,RO	
3249	015054	012767	014776	162742		MOV	# PFAIL,24	,SET UP FOR POWER FAILURE
3250	015062	106427	000340			MTPS	#340	
3251	015066	012706	001100			MOV	#STACK,SP	
3252	015072	005067	001102			CLR	TEMP	
3253	015076	005267	001076			INC	TEMP	
3254	015102	001375				BNE	-4	
3255	015104	104413				CONVRT		
3256	015106	015130				PFTAB		
3257	015110	104401				TYPE		
3258	015112	015432				MPFAIL		
3259	015114	005067	164263			CLR	\$ERFLG	
3260	015120	005067	164272			CLR	\$ERRPC	
3261	015124	000177	163754			JMP	@RETURN	
3262	015130	000001			PFTAB	1		
3263	015132	006	002			BYTE	6,2	
3264	015134	000207				RETURN		
3265	015136	005015	042012	053125	MTITLE	ASCIZ	<15><12><12>/DUV11 DZDUS-B TAPE C /<15><12>	
3266	015144	030461	042040	042132				
3267	015152	051525	041055	052040				
3268	015160	050101	020105	020103				
3269	015166	005015	000					
3270	015171	015	053012	041505	MVECTO	ASCIZ	<15><12>/VEC ADD- /	
3271	015176	040440	042104	000055				
3272	015204	005015	051461	020124	MREGAD	ASCIZ	<15><12>/1ST DEV REC CSR ADD- /	
3273	015212	042504	03E126	051040				
3274	015220	041505	041440	051123				
3275	015226	040440	042104	000055				
3276	015234	005015	052515	052114	MMULT	ASCIZ	<15><12>/MULT DEV ? (Y OR N)- /	
3277	015242	042040	053105	037440				
3278	015250	024040	020131	051117				
3279	015256	047040	026451	000				
3280	015263	015	046012	051501	MLASTD	ASCIZ	<15><12>/LAST DEV REC CSR ADDP- /	
3281	015270	020124	042504	035126				
3282	015276	051040	041505	041440				
3283	015304	051123	040440	042104				
3284	015312	026522	000					
3285	015315	075	042504	044526	DEVICE	ASCIZ	/=DEVICE /	
3286	015322	042503	020040	000				
3287	015327	015	051412	046105	MCOW	ASCIZ	<15><12>/SELECT TO RUN @ACTREG /	
3288	015334	041505	020124	047524				
3289	015342	051040	047125	040040				
3290	015350	041501	051124	043505				
3291	015356	000						
3292	015357	015	047412	043126	MRANGE	ASCIZ	<15><12>/OVFLG: RETYPE LAST DEV RXCSR ADDS- /	
3293	015364	047514	051072	052105				
3294	015372	050131	020105	040514				
3295	015400	052123	042040	053105				
3296	015406	051040	041530	051123				
3297	015414	040440	042104	026523				
3298	015422	000						
3299	015423	040	037440	000	MQM	ASCIZ	/ ? /	
3300	015427	015	000012		MCRLF:	ASCIZ	<15><12>	
3301	015432	043120	044501	026114	MPFAIL	ASCIZ	/PFAIL, RESTART AT TEST IN PROGRESS /	
3302	015440	020040	042522	052123				
3303	015446	051101	020124	052101				

3304	015454	052040	051505	020124			
3305	015462	047111	050040	047522			
3306	015470	051107	051505	000123			
3307	015476	005015	047105	020104	MEPASS	ASCIZ	<15><12>/END OF PASS TAPE (/
3308	015504	043117	050040	051501			
3309	015512	020123	040524	042520			
3310	015520	041440	000				
3311	015523	015	051012	000	MR	ASCIZ	<15><12>/R/
3312	015527	015	052012	051505	MTSTPC	ASCIZ	<15><12>/TEST PC-/
3313	015534	020124	041520	000055			
3314	015542	005015	047514	045503	MLOCK	ASCIZ	<15><12>/LOCK ON TEST? (Y OR N)-/
3315	015550	047440	020116	052040			
3316	015556	051505	037524	024040			
3317	015564	020131	051117	047040			
3318	015572	026451	000				
3319	015575	015	021412	047440	MSYNC	ASCIZ	<15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3320	015602	020106	054523	041516			
3321	015610	041440	040510	051522			
3322	015616	051440	046105	041505			
3323	015624	042524	020104	020050			
3324	015632	020061	051117	031040			
3325	015640	026451	000				
3326	015643	015	044412	020123	MWIRE6	ASCIZ	<15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3327	015650	042523	020103	046530			
3328	015656	052111	051411	044527			
3329	015664	041524	020110	032505			
3330	015672	026465	020062	047111			
3331	015700	020077	054450	047440			
3332	015706	020122	024516	000055			
3333	015714	005015	051511	051440	MWIRE5	ASCIZ	<15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3334	015722	041505	051040	041505			
3335	015730	051440	044527	041524			
3336	015736	020110	032505	026465			
3337	015744	020063	047111	020077			
3338	015752	054450	047440	020122			
3339	015760	024516	000055				
3340	015764	005015	051511	047440	MWIRE4	ASCIZ	<15><12>/IS OPT CLR ENABLE SWITCH E55 1 IN? (Y OR N)-/
3341	015772	052120	041440	051114			
3342	016000	042440	040516	046102			
3343	016006	020105	053523	052111			
3344	016014	044103	042440	032465			
3345	016022	030455	044440	037516			
3346	016030	024040	020131	051117			
3347	016036	047040	026451	000			
3348	016043	015	005012	031510	MEXTJ	ASCIZ	<15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3349	016050	032461	041440	047117			
3350	016056	042516	052103	051117			
3351	016064	047440	020116	024077			
3352	016072	020131	051117	047040			
3353	016100	026451	000				
3354	016103	015	020012	043536	MCNTG	ASCIZ	<15><12>/ G /
3355	016110	020040	000				
3356	016113	040	053523	036522	MMSWR	ASCIZ	/ SWR= /
3357	016120	020040	000040				
3358	016124	020040	047040	053505	MMNEW	ASCIZ	/ NEW= /
3359	016132	020075	000040				

```

3360 .EVEN
3361
3362 .BUFFERS FOR INPUT-OUTPUT
3363
3364 016136 000000 INBUF 0
3365 = +40 016200 = +40
3366 016200 000000 TEMP: 0
3367 = +40 016242 = +40
3368 016242 000000 MDATA 0
3369 = +40 016304 = +40
3370 .SBTTL SCOPE HANDLER ROUTINE
3371
3372 .,*****
3373 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS IT WILL INCREMENT
3374 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG (DISPLAY<7 0>)
3375 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15 08>
3376 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE.
3377 ;*SW14=1 LOOP ON TEST
3378 ;*SW11=1 INHIBIT ITERATIONS
3379 ;*SW09=1 LOOP ON ERROR
3380 ;*SW08=1 LOOP ON TEST IN SWR<7 0>
3381 ;*CALL
3382 ;* SCOPE .,SCOPE=10T
3383
3384 016304 $SCOPE
3385
3386 .SCOPE LOOP AND INTERATION HANDLER
3387
3388 SCOPE
3389 016304 004767 174376 JSR PC,CKSWR
3390 016310 005067 163102 CLR $ERRPC ;CLEAR LAST ERROR PC
3391 016314 022716 003376 CMP #TST1+2,(SP) ;IS SCOPE AT BEGINING OF TEST 1?
3392 016320 001422 BEQ $XTSTR ;YES NO LOOP
3393
3394 016322 032777 040000 163110 TTST BIT #BIT14,@SWR ;THIS CODE IS FOR TESTING FOR BIT 14
3395 016330 001412 BEQ 1$ ;ON LSI WHICH SYSMAC CANNOT HANDLE
3396 016332 016767 163044 163046 MOV $TSTNM,$LPADR
3397 016340 000406 BR 1$
3398 016342 105777 163076 TSTB @STKS ;KEYBOARD DONE?
3399 016346 100123 BPL $OVER ;BR IF NO
3400 016350 017766 163072 177776 MOV @STKB,-2(SP) ;CLEAR DONE BIT
3401 016356 032777 040000 163054 1$ BIT #BIT14,@SWR ;LOOP ON PRESENT TEST?
3402 016364 001114 BNE $OVER ;YES IF SW14=1
3403 .#####START OF CODE FOR THE XOR TESTER#####
3404 016366 000416 $XTSTR BR 6$ ;IF RUNNING ON THE "XOR" TESTER CHANGE
3405 ;THIS INSTRUCTION TO A "NOP" (NOP-240)
3406 016370 013746 000004 MOV @ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3407 016374 012737 016414 000004 MOV #5$,@ERRVEC ;SET FOR TIMEOUT
3408 016402 005737 177060 TST @177060 ;TIME OUT ON XOR?
3409 016406 012637 000004 MOV (SP)+,@ERRVEC ;RESTORE THE ERROR VECTOR
3410 016412 000463 BR $$VLAD ;GO TO THE NEXT TEST
3411 016414 022626 5$ CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
3412 016416 012637 000004 MOV (SP)+,@ERRVEC ;RESTORE THE ERROR VECTOR
3413 016422 000423 BR 7$ ;LOOP ON THE PRESENT TEST
3414 016424 6$ .#####END OF CODE FOR THE XOR TESTER#####
3415 016424 032777 000400 163006 BIT #BIT08,@SWR ;LOOP ON SPEC TEST?

```

```

3416 016432 001404          BEQ      2$          ;; BR IF NO
3417 016434 127767 163000 162740  CMPB   @SWR, $TSTNM  ;; ON THE RIGHT TEST?  SWR<7 0>
3418 016442 001465          BEQ      $OVER      ;; BR IF YES
3419 016444 105767 162733          TSTB   $ERFLG      ;; HAS AN ERROR OCCURRED?
3420 016450 001421          BEQ      3$          ;; BR IF NO
3421 016452 126767 162737 162723  CMPB   $ERMAX, $ERFLG  ;; MAX ERRORS FOR THIS TEST OCCURRED?
3422 016460 101015          BHI     3$          ;; BR IF NO
3423 016462 032777 001000 162750  BIT    #BIT09, @SWR   ;; LOOP ON ERROR?
3424 016470 001404          BEQ      4$          ;; BR IF NO
3425 016472 016767 162712 162706  7$    MOV    $LPERR, $LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
3426 016500 000446          BR      $OVER      ;;
3427 016502 105067 162675          CLRB   $ERFLG      ;; ZERO THE ERROR FLAG
3428 016506 005067 163000          CLR    $TIMES      ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
3429 016512 000415          BR      1$          ;; ESCAPE TO THE NEXT TEST
3430 016514 032777 004000 162716  3$    BIT    #BIT11, @SWR  ;; INHIBIT ITERATIONS?
3431 016522 001011          BNE    1$          ;; BR IF YES
3432 016524 005767 163004          TST    $PASS      ;; IF FIRST PASS OF PROGRAM
3433 016530 001406          BEQ    1$          ;; INHIBIT ITERATIONS
3434 016532 005267 162646          INC    $ICNT      ;; INCREMENT ITERATION COUNT
3435 016536 026767 162750 162640  CMP    $TIMES, $ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
3436 016544 002024          BGE    $OVER      ;; BR IF MORE ITERATION REQUIRED
3437 016546 012767 000001 162630  1$    MOV    #1, $ICNT   ;; REINITIALIZE THE ITERATION COUNTER
3438 016554 016767 000056 162730          MOV    $MXCNT, $TIMES ;; SET NUMBER OF ITERATIONS TO DO
3439 016562 105267 162614          $SVLAD INCB   $TSTNM  ;; COUNT TEST NUMBERS
3440 016566 116767 162610 162736  MOVB   $TSTNM, $TESTN ;; SET TEST NUMBER IN APT MAILBOX
3441 016574 011667 162606          MOV    (SP), $LPADR  ;; SAVE SCOPE LOOP ADDRESS
3442 016600 011667 162604          MOV    (SP), $LPERR  ;; SAVE ERROR LOOP ADDRESS
3443 016604 005067 162704          CLR    $ESCAPE     ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
3444 016610 112767 000001 162577  MOVB   #1, $ERMAX   ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3445 016616 016777 162560 162616  $OVER  MOV    $TSTNM, @DISPLAY ;; DISPLAY TEST NUMBER
3446 016624 016716 162556          MOV    $LPADR, (SP)  ;; FUDGE RETURN ADDRESS
3447 016630 000002          4$    RTI          ;;
3448 016632 001407          BRW    1407        ;;
3449 016634 000432          BRX    432        ;;
3450 016636 000005          $MXCNT 5          ;; MAX NUMBER OF ITERATIONS
3451          SBTTL TRAP DECODER
3452
3453          ;; *****
3454          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3455          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3456          ;*OF THE DESIRED ROUTINE THEN USING THE ADDRESS OBTAINED IT WILL
3457          ;*GO TO THAT ROUTINE
3458
3459 016640 010046          STRAP  MOV    RO, -(SP)  ;; SAVE RO
3460 016642 016600 000002          MOV    2(SP), RO    ;; GET TRAP ADDRESS
3461 016646 005740          TST    -(RO)        ;; BACKUP BY 2
3462 016650 111000          MOVB   (RO), RO     ;; GET RIGHT BYTE OF TRAP
3463 016652 006300          ASL    RO           ;; POSITION FOR INDEXING
3464 016654 016000 016674          MOV    $TRPAD(RO), RO ;; INDEX TO TABLE
3465 016660 000200          RTS    RO          ;; GO TO ROUTINE
3466
3467
3468          ;; THIS IS USE TO HANDLE THE "GETPRI" MACRO
3469
3470 016662 011646          $TRAP2 MOV    (SP), -(SP)  ;; MOVE THE PC DOWN
3471 016664 016666 000004 000002          MOV    4(SP), 2(SP) ;; MOVE THE PSW DOWN

```

```

3472 016672 000002          RTI          ,,RESTORE THE PSW
3473
3474          SBTTL  TRAP TABLE
3475
3476          ,*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3477          ,*BY THE "TRAP" INSTRUCTION.
3478
3479          ,          ROUTINE
3480          ,          -----
3481 016674 016662          $TRPAD  WORD  $TRAP2
3482 016676 013056          $TYPE   ,,CALL=TYPE   TRAP+1(104401) TTY TYPEOUT ROUTINE
3483 016700 014574          $TYPOC  ,,CALL=TYPOC  TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3484 016702 014550          $TYPOS  ,,CALL=TYPOS  TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
3485 016704 014610          $TYPON  ,,CALL=TYPON  TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
3486
3487
3488 016706 013034          SCOP1   ,,CALL=SCOP1  TRAP+5(104405)
3489 016710 013340          INSTR  ,,CALL=INSTR  TRAP+6(104406)
3490 016712 013446          INSTER ,,CALL=INSTER TRAP+7(104407)
3491 016714 013456          PARAM  ,,CALL=PARAM  TRAP+10(104410)
3492 016716 013656          SAVO5  ,,CALL=SAVO5   TRAP+11(104411)
3493 016720 013716          RESO5  ,,CALL=RESO5   TRAP+12(104412)
3494 016722 013750          CONVRT ,,CALL=CONVRT   TRAP+13(104413)
3495 016724 014126          SETFLG ,,CALL=SETFLG  TRAP+14(104414)
3496          ,*****
3497          ,UTILITIES
3498          ,*****
3499
3500          THIS UTILITY CALCULATES PRIORITY LEVEL
3501 016726 006367 000044          DULEV  ASL      DUPRT  ,SHIFT LEFT
3502 016732 006367 000040          ASL      DUPRT  ;
3503 016736 006367 000034          ASL      DUPRT  ,
3504 016742 006367 000030          ASL      DUPRT  ,
3505 016746 006367 000024          ASL      DUPRT  ,
3506 016752 016767 000020 000020          MOV      DUPRT,LESS1 ,MOVE THIS TO LESS1
3507 016760 162767 000001 000012          SUB      #1,LESS1 ,CREATE LESS1
3508 016766 042767 000037 000004          BIC      #37,LESS1 ,CLEAR TNZVC
3509 016774 000207          RTS      PC
3510 016776 000240          DUPRT  PR5
3511 017000 000200          LESS1  PR4 ,LEVEL TO ALLOW INTERRUPTS
3512
3513          ,NEW DU ADDRESSES
3514 017002 016767 000126 162702          DUADDR  MOV      DUBASE,RXCSR ,XXX0
3515 017010 005267 000120          INC      DUBASE
3516 017014 016767 000114 162672          MOV      DUBASE,HRXCSR ,XXX1
3517 017022 005267 000106          INC      DUBASE
3518 017026 016767 000102 162662          MOV      DUBASE,RXDBUF ,XXX2
3519 017034 016767 000074 162660          MOV      DUBASE,PARCSR ,XXX2
3520 017042 005267 000066          INC      DUBASE
3521 017046 016767 000062 162644          MOV      DUBASE,HRXDBUF ;XXX3
3522 017054 016767 000054 162642          MOV      DUBASE,HPARCSR ,XXX3
3523 017062 005267 000046          INC      DUBASE
3524 017066 016767 000042 162632          MOV      DUBASE,TXCSR ,XXX4
3525 017074 005267 000034          INC      DUBASE
3526 017100 016767 000030 162622          MOV      DUBASE,HTXCSR ,XXX5
3527 017106 005267 000022          INC      DUBASE
  
```

```

3528 017112 016767 000016 162612      MOV    DUBASE, TXDBUF  ,XXX6
3529 017120 005267 000010          INC    DUBASE
3530 017124 016767 000004 162602      MOV    DUBASE, HTXDBUF ,XXX7
3531 017132 000207          RTS    PC
3532 017134 000000          DUBASE 0
3533
3534          , THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3535          , INFORMATION CONTAINED IN $TMP1 AND IT IS
3536          , SHIFTED IN BY THE CONTENTS OF SHIFT
3537 017136 042777 040000 162562  RPOKE  BIC    #MTDATA, @TXCSR
3538 017144 005067 162332          CLR    $TMP2
3539 017150 006067 162324          ROR    $TMP1  ; FORCE CARRY
3540 017154 006067 162322          ROR    $TMP2  , PICK UP CARRY IN BIT 15
3541 017160 006267 162316          ASR    $TMP2  , SHIFT INTO BIT 14
3542 017164 042767 100000 162310      BIC    #BIT15, $TMP2  , CLR BIT 15
3543 017172 056777 162304 162526      BIS    $TMP2, @TXCSR  , POKE MAINT DATA
3544 017200 042777 020000 162520      BIC    #CLK, @TXCSR  , POKE CLK
3545 017206 052777 020000 162512      BIS    #CLK, @TXCSR  ,
3546 017214 005367 161702          DEC    SHIFT
3547 017220 001346          BNE    RPOKE
3548 017222 000207          RTS    PC
3549          , THIS ROUTINE CALCULATES ODD PARIITY FOR AN 8 BIT CHAR
3550 017224 016767 162250 162250  ODD8  MOV    $TMP1, $TMP2  , SAVE TEMP1
3551 017232 005067 162246          CLR    $TMP3
3552 017236 012727 000010          MOV    #8 , (PC)+
3553 017242 000000          4$   0
3554 017244 006067 162232          1$   ROR    $TMP2
3555 017250 005567 162230          ADC    $TMP3
3556 017254 005367 177762          DEC    4$
3557 017260 001371          BNE    1$
3558 017262 006067 162216          ROR    $TMP3
3559 017266 103404          BCS    2$
3560 017270 052767 000400 162202      BIS    #BIT8, $TMP1  , SET ODD PARITY
3561 017276 000403          BR     3$
3562 017300 042767 000400 162172  2$   BIC    #BIT8, $TMP1  , CLR EVEN PARITY
3563          , $TMP1 NOW HAS ODD PARITY CHARACTER
3564 017306 000207          3$   RTS    PC
3565
3566          , THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3567 017310 016767 162164 162164  EVEN8 MOV    $TMP1, $TMP2  , SAVE TEMP1
3568 017316 005067 162162          CLR    $TMP3
3569 017322 012727 000010          MOV    #8 , (PC)+
3570 017326 000000          4$   0
3571 017330 006067 162146          1$   ROR    $TMP2
3572 017334 005567 162144          ADC    $TMP3
3573 017340 005367 177762          DEC    4$
3574 017344 001371          BNE    1$
3575 017346 006067 162132          ROR    $TMP3
3576 017352 103004          BCC    2$
3577 017354 052767 000400 162116      BIS    #BITS, $TMP1  , SET EVEN PARITY
3578 017362 000403          BR     3$
3579 017364 042767 000400 162106  2$   BIC    #BIT8, $TMP1  , CLR ODD PARITY
3580          , $TMP1 NOW HAS EVEN PARITY CHARACTER
3581 017372 000207          3$   RTS    PC
3582
3583 017374 062716 000002          TRPREG ADD    #2, (SP) , ALLOW IT TO 'CRUNCH' INTO HLT BACK
  
```

DZDUS-B MACY11 30(1046) 21-SEP-77 09 12 PAGE 72
UZDUSB M11 31-MAY-77 09 49 TRAP TABLE

F 6

SEQ 0070

3584
3585 017400 000002
3586 000001
END RTI

, IN MAIN PART OF THE PROGRAM

AAA	003200	1293*								
ABASE =	000000	876	917							
ACDW1 =	000000	876	919							
ACDW2 =	000000	876	920							
ACPUOP=	000000	876	891							
ACTREG	001166	735*	1249*	1263*	1264*	1271*	2697	2700	2710	
ADDW0 =	000000	876	921							
ADDW1 =	000000	876	922							
ADDW10=	000000	876	931							
ADDW11=	000000	876	932							
ADDW12=	000000	876	933							
ADDW13=	000000	876	934							
ADDW14=	000000	876	935							
ADDW15=	000000	876	936							
ADDW2 =	000000	876	923							
ADDW3 =	000000	876	924							
ADDW4 =	000000	876	925							
ADDW5 =	000000	876	926							
ADDW6 =	000000	876	927							
ADDW7 =	000000	876	928							
ADDW8 =	000000	876	929							
ADDW9 =	000000	876	930							
ADEVCT=	000000	876	882							
ADEVN =	000000	876	918							
ADRCNT=	013655	2912*	2951*	2958*						
AENV =	000000	876	887							
AENVN =	000000	876	888							
AFATAL=	000000	876	879							
AMADR1=	000000	876	904							
AMADR2=	000000	876	908							
AMADR3=	000000	876	911							
AMADR4=	000000	876	914							
AMAMS1=	000000	876	898							
AMAMS2=	000000	876	906							
AMAMS3=	000000	876	909							
AMAMS4=	000000	876	912							
AMSGAD=	000000	876	884							
AMSGLG=	000000	876	885							
AMSGTY=	000000	876	878							
AMTYP1=	000000	876	899							
AMTYP2=	000000	876	907							
AMTYP3=	000000	876	910							
AMTYP4=	000000	876	913							
APASS =	000000	876	881							
APRIOR=	000000	876								
APTCSU=	000040	536*	2827							
APTENV=	000001	536*	2820	3074						
APTSIZ=	000200	536*	1168							
APTSP0=	000100	536*	2822							
ASWREG=	000000	876	889							
ATESTN=	000000	876	880							
AUNIT =	000000	876	883							
AUSWR =	000000	876	890							
AVECT1=	000000	876	915							
AVECT2=	000000	876	916							
BASEAD	001154	730*	1231*	1268*	1269	1275*	1277*	2704*	2716*	2720

BASEIV	001162	733#	1241*	2705*	2717*	2722	2723*	2724	2725*	2726	2727*	2728	2729*
BBB	003026	1248	1252#										
BINWRD	014124	2993*	2995	3025#									
BITW =	002000	802#	995#										
BITO =	000001	665#	778	809	971	1002							
BIT00 =	000001	655#	665										
BIT01 =	000002	654#	664										
BIT02 =	000004	653#	663										
BIT03 =	000010	652#	662										
BIT04 =	000020	651#	661										
BIT05 =	000040	650#	660										
BIT06 =	000100	649#	659										
BIT07 =	000200	648#	658										
BIT08 =	000400	647#	657	3415									
BIT09 =	001000	646#	656	3084	3423								
BIT1 =	000002	664#	777	970									
BIT10 =	002000	645#	768	802	961	995	2166	2210	3062				
BIT11 =	004000	644#	767	960	3430								
BIT12 =	010000	643#	766	783	959	976							
BIT13 =	020000	642#	765	782	801	958	975	994	3069				
BIT14 =	040000	641#	764	781	800	957	974	993	3394	3401			
BIT15 =	100000	640#	763	780	799	956	973	992	3542				
BIT2 =	000004	663#	776	969									
BIT3 =	000010	662#	775	808	968	1001							
BIT4 =	000020	661#	774	807	967	1000							
BIT5 =	000040	660#	773	806	966	999							
BIT6 =	000100	659#	772	805	965	998							
BIT7 =	000200	658#	771	804	964	997							
BIT8 =	000400	657#	770	803	963	979	996	3560	3562	3577	3579		
BIT9 =	001000	656#	769	785	962	978							
BPTVEC =	000014	672#											
BREAK =	000001	809#	1002#	1359	1411	1463	1515	1567	1619	1671	1710	1749	1788
		1867	1908	1949	1987	2028	2069	2110	2147	2191	2235	2273	2312
		2384	2420	2456	2489	2522	2555	2588	2624	2660			1826
		3448#											2348
BRW	016632	3449#											
BRX	016634	766#	959#										
CARDET =	010000	2696	2719	2736#									
CCC	012612	2991*	2996	3009*	3023#	3024							
CHRCNT	014122	2690	2753	2760#	2792	3389							
CKSWR	012706	801#	994#	1359	1365	1366	1411	1417	1418	1463	1469	1470	1515
CLK =	020000	1522	1567	1573	1574	1619	1625	1626	1671	1677	1678	1680	1681
		1716	1717	1719	1720	1749	1755	1756	1758	1759	1788	1794	1795
		1798	1826	1832	1833	1835	1836	1867	1873	1874	1876	1877	1908
		1915	1917	1918	1949	1955	1956	1958	1959	1987	1993	1994	1996
		2028	2034	2035	2037	2038	2069	2075	2076	2078	2079	2110	2116
		2119	2120	2147	2153	2154	2156	2157	2191	2197	2198	2200	2201
		2241	2242	2273	2279	2280	2312	2318	2319	2321	2322	2348	2354
		2357	2358	2384	2390	2391	2393	2394	2420	2426	2427	2429	2430
		2462	2463	2489	2495	2496	2522	2528	2529	2555	2561	2562	2588
		2595	2597	2598	2624	2630	2631	2633	2634	2660	2666	2667	2670
		3544	3545										
CNTLU	012762	1198	2771#										
CONVRT =	104413	2693	2773	3255	3494#								
COUNT	001124	710#	2324*	2331*	2360*	2367*	2396*	2403*	2432*	2439*	2465*	2472*	2498*
		2531*	2538*	2564*	2571*	2600*	2607*	2636*	2643*	2672*	2679*		2505*

CROSS REFERENCE TABLE -- USER SYMBOLS

\$ITEMB	001414	830#	3068*	3076	3092	3111													
\$LF	001524	871#	2876	3092															
\$LFLG	000243R	536#*																	
\$LPADR	001406	827#	1149*	2752*	3396*	3425*	3441*	3446	3450										
\$LPERR	001410	828#	1150*	3086	3425	3442*	3450												
\$MADR1	001560	904#																	
\$MADR2	001564	908#																	
\$MADR3	001570	911#																	
\$MADR4	001574	914#																	
\$MAIL	001526	877#	1115	1119	1167	2820	3074	3440											
\$MAMS1	001556	898#																	
\$MAMS2	001562	906#																	
\$MAMS3	001566	909#																	
\$MAMS4	001572	912#																	
\$MBADR	002140	1115#																	
\$MFLG	000242R	536#*																	
\$MSGAD	001542	536#	884#																
\$MSGLG	001544	536#	885#																
\$MSGTY	001526	536#	878#																
\$MTYP1	001557	899#																	
\$MTYP2	001563	907#																	
\$MTYP3	001567	910#																	
\$MTYP4	001573	913#																	
\$MXCNT	016636	3438	3450#																
\$N =	000000	534#	2683#																
\$NULL	001454	248#	2847	2876															
\$NWTST=	000000	1352#	1404#	1456#	1508#	1560#	1612#	1664#	1703#	1742#	1781#	1819#	1860#	1901#					
		1942#	1980#	2021#	2062#	2103#	2140#	2184#	2228#	2266#	2305#	2341#	2377#	2413#					
		2449#	2482#	2515#	2548#	2581#	2617#	2653#											
\$OCNT	014772	3179*	3208*	3221#															
\$OMODE	014774	3174*	3178*	3183	3186*	3197*	3223#												
\$OVER	016616	3399	3402	3418	3426	3436	3445#												
\$PASS	001534	881#	1167*	2742*	3432	3451													
\$PASTM	002144	1117#																	
\$PWDRN	014776	1144	3227#																
\$QUES	001522	869#	2876	3092															
\$RDOCHR=	***** U	3488																	
\$RDODEC=	***** U	3488																	
\$RDLIN=	***** U	3488																	
\$RDOCT=	***** U	3488																	
\$REGAD	001460	852#																	
\$REGO	001462	854#	2971*	2976															
\$REG1	001464	855#	2970*	2977															
\$REG2	001466	856#	2969*	2978															
\$REG3	001470	857#	2968*	2979															
\$REG4	001472	858#	2967*	2980															
\$REG5	001474	859#	2966*	2981															
\$R2A =	***** U	3488																	
\$SAVRE=	***** U	3488																	
\$SCOPE	016304	1138	3384#																
\$SETUP=	000017	1120#	1137	1138	1140	1142	1144	1146	1147	1149	3059	3084	3091	3385					
\$STUP =	177777	1120#																	
\$SVLAD	016562	3410	3439#																
\$SVPC =	002136	1092#	1097																
\$SWR =	177400	525#	866	867	868	1146	1147	1149	1150	1354	1406	1458	1510	1562					
		1614	1666	1705	1744	1783	1821	1862	1903	1944	1982	2023	2064	2105					

CROSS REFERENCE TABLE -- USER SYMBOLS

		2142	2186	2230	2268	2307	2343	2379	2415	2451	2484	2517	2550	2583
		2619	2655	3050	3051	3052	3053	3054	3062	3069	3081	3084	3092	3376
		3377	3378	3379	3380	3401	3413	3415	3416	3419	3420	3421	3428	3429
		3430	3442	3445	3450									
\$SWREG	001550	889#	1170											
\$SWRMK=	000000	3380	3381	3417										
\$TESTN	001532	880#	3440*											
\$TIMES	001512	866#	1146*	3428*	3435	3438*	3450							
\$TKB	001446	845#	2766	2889	2897	3400								
\$TKS	001444	844#	2764	2887	3398									
\$TMP0	001476	860#												
\$TMP1	001500	861#	1370*	1382*	1386*	1422*	1434*	1438*	1474*	1486*	1490*	1526*	1538*	1542*
		1578*	1590*	1594*	1630*	1642*	1646*	1683*	1722*	1761*	1800*	1840*	1881*	1922*
		1963*	2001*	2042*	2083*	2124*	2162*	2165*	2166*	2206*	2209*	2210*	2247*	2285*
		2325*	2361*	2397*	2433*	2466*	2499*	2532*	2565*	2601*	2637*	2673*	3539*	3550
		3560*	3562*	3567	3577*	3579*								
\$TMP2	001502	862#	3538*	3540*	3541*	3542*	3543	3550*	3554*	3567*	3571*			
\$TMP3	001504	863#	3551*	3555*	3558*	3568*	3572*	3575*						
\$TMP4	001506	864#												
\$TMP5	001510	865#												
\$TN =	000042	536#	1352	1354#	1404	1406#	1456	1458#	1508	1510#	1560	1562#	1612	1614#
		1664	1666#	1703	1705#	1742	1744#	1781	1783#	1819	1821#	1860	1862#	1901
		1903#	1942	1944#	1980	1982#	2021	2023#	2062	2064#	2103	2105#	2140	2142#
		2184	2186#	2228	2230#	2266	2268#	2305	2307#	2341	2343#	2377	2379#	2413
		2415#	2449	2451#	2482	2484#	2515	2517#	2548	2550#	2581	2583#	2617	2619#
		2653	2655#											
\$TPB	001452	847#	2865*	2876	2897*									
\$TPFLG	001457	851#	2814	2876										
\$TPS	001450	846#	2863	2876	2895									
\$TRAP	016640	1142	3459#											
\$TRAP2	016662	3470#	3481											
\$TRP =	000015	3474#	3483#	3484#	3485#	3486#	3488	3489#	3490#	3491#	3492#	3493#	3494#	3495#
		3496#												
\$TRPAD	016674	3464	3481#											
\$TSTM	002142	1116#												
\$TSTM	001402	824#	1180*	1335	1338	2737*	3061	3092	3098	3375	3396	3417	3439*	3440
		3445	3451											
\$TYPBN=	***** U	3486												
\$TYPDS=	***** U	3486												
\$TYPE	013056	536	2814#	3474	3482									
\$TYPEC	013270	2844	2851	2858	2863#	2864								
\$TYPEX	013336	2869	2871	2874#										
\$TYPOC	014574	3177#	3483											
\$TYPON	014610	3176	3179#	3485										
\$TYPOS	014550	3172#	3484											
\$UNIT	001540	883#												
\$UNITM	002146	1118#												
\$USWR	001552	890#	1205											
\$VECT1	001576	915#												
\$VECT2	001600	916#												
\$XTSTR	016366	3392	3404#											
\$OFILL	014773	3173*	3177*	3187	3222#									
\$4OCAT=	***** U	3071	3401											
	= 017402	536#	570#	683#	686#	691#	746#	747#	821#	872	1078#	1092	1093#	1095#
		1097#	1103	1104#	1106#	1108#	1135	1149	1150	1373	1377	1392	1425	1429
		1444	1477	1481	1496	1529	1533	1548	1581	1585	1600	1633	1637	1652

HEADE	525#	
SETUP	525#	1120
SACT1	525#	1088
SAPT8	525#	873#
SAPTH	525#	1098
SAPTY	525#	536
SCATC	525#	
SCMTA	525#	815
SEOP	525#	
SERRO	525#	3044
SERRT	525#	3100
SPOWE	525#	
SSCOP	525#	3370
STRAP	525#	3451
STYPE	525#	2797
STYPO	525#	3147

ABS 017402 000

ERRORS DETECTED 0

DZDUSB, DZDUSB./SOL /CRF=DZDUS1./EQ. RUNC. DZDUS2. DZDUSB
RUN-TIME 22 12 1 SECONDS
RUN-TIME RATIO 319/36=8 6
CORE USED 30K (59 PAGES)