

DUV/LSI11

DUV11 OFFLINE RCVR TESTS
MD-11-DZDUR-A

EP-DZDUR-A-DL-A

APR 1977

COPYRIGHT © 1977

30-0000

FICHE 1 OF 1

MADE IN USA

The microfiche card displays a grid of 48 frames, arranged in 8 rows and 6 columns. Each frame contains a small table or chart with various numerical and graphical information, likely representing test results for different components or conditions. The data is presented in a structured, tabular format, with some frames showing what appears to be a bar chart or a similar graphical representation alongside numerical values. The overall layout is consistent across all frames, suggesting a systematic recording of test data.

30-0000

HDR1DZDURASEQ
DZDUR1.M11

00010000
02-FEB-77 08:16

770419

801
PDP10 411

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 2
SEQ 0001

.REM *

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

PRODUCT NAME: DUV11 OFFLINE RECEIVER TESTS

PRODUCT CODE:MAINDEC-11-DZDUR-A

RELEASE DATE:FEB 1977

MAINTAINER :DIAGNOSTICS

*
.REM *

COPYRIGHT (C) 1977
DIGITAL EQUIPMENT CORPORATION, MAYNARD MASS.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OF RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

*

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.

.REM *

- 1. THE DUV11 OFFLINE RECEIVER TESTS VERIFY THAT THE RECEIVER CHIP/LOGIC WORKS PROPERLY.

* .REM *

- 2. REQUIREMENTS

* .REM *

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

- 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 "SUSWR". IN ORDER TO BE FLEXIBLE ON THE AVAILABILITY OF THE H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11. BIT 0 IN SUSWR REFLECTS THIS STATUS, A 0 = CONNECTOR PRESENT, A 1 = CONNECTOR NOT AVAILIBLE.

- 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)
SW02=1
NOTE1: IN GENERAL SW01 WILL BE USED WHEN SW02=1 IS USED
NOTE2: WITHOUT SW01=1 "LOCK ON TEST" WILL DEFAULT TO TEST 1
- 4.2 STARTING ADDRESS

THE STARTING ADDRESS FOR ALL TESTS IS 000200

THE RETARTING ADDRESS FOR ALL TESTS IS 000200
THE STARTING ADDRESS TO ENTER A SELECTED TEST IS 000200
THE STARTING ADDRESS TO LOCK ON TEST IS 000200
- 4.3 PROGRAM AND/OR OPERATOR ACTION
- 4.3.1 INITIAL PROGRAM START
 - 4.3.1.1 LOAD PROGRAM INTO MEMORY WITH ABSOLUTE LOADER
 - 4.3.1.2 SET SWITCH REGISTER (LOC. 176) TO ZERO.
 - 4.3.1.3 TYPE 200G.
 - 4.3.1.4 PROGRAM WILL START.
 - 4.3.1.5 THE PROGRAM WILL TYPE "DUV11 DZDUR-A TAPE B" (ONCE ONLY)
 - 4.3.1.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT
TO START TESTING ,AND THEN TESTING WILL BEGIN
- 4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN
 - 4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING
- 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.

```
*
*.REM *
*
*.REM *
```

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 2 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 ? ANDDO 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 SW02 =1
NOTE: THIS WILL ONLY WORK WHEN A SINGLE DEVICE IS SELECTED
SEE NOTE IN 4.3.4 FOR MORE DETAILS

4.3.5.1 SET SW02=1 IN SWITCH REG. (LOC. 176)

4.3.5.2 TYPE 200G.

4.3.5.3 PROGRAM WILL START.

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

4.3.5.5 TYPE IN THE APPROPRIATE ANSWER YES OR NO FOLLOWED BY A
<CARRIAGE RETURN>

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

4.3.5.6 IF A YES ANSWER WAS GIVEN: THE PROGRAM WILL ACT AS FOLLOWS...
THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT HAS STARTED
TESTING AT TEST 1 AND WILL REMAIN IN TEST 1 UNTIL HALTED
OR IF ANY KEY IS STRUCK ON THE TELETYPE, THE PROGRAM
WILL FREEZE ON THE NEXT TEST UNTIL A KEY IS STRUCK ON
THE TELETYPE AND SO FORTH THRU THE PROGRAM. IF SW01 = 1 IT
WILL PERFORM AS IN SECTION 4.3.4 ALLOWING ONE TO FREEZE
ON A SELECTED TEST RATHER THAN DEFAULTING TO TEST 1

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS (INTERNAL TO THE CPU, ACCESSED VIA LOC. 176).

SW15 = 1 HALT ON ERROR
SW14 = 1 LOOP ON CURRENT TEST
SW13 = 1 INHIBIT ERROR TYPEOUT
SW11 = 1 INHIBIT ITERATIONS
SW10 = 1 ESCAPE TO NEXT TEST ON ERROR
SW09 = 1 LOOP ON ERROR
SW02 = 1 LOCK ON TEST
SW01 = 1 RESTART PROGRAM AT SELECTED TEST
SW00 = 1 RESELECT VECTOR AND CONTROL REGISTER ADDRESSES
& PARAMETERS AFTER A PROGRAM RESTART
TO INHIBIT "END OF 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,BASEIV ;NEXT BLOCK (VECTORS)" TO "ZERO: ADD #0,BASEIV"; 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.

7.2.2.2ORSET SW00=1 IN SWITCH REG (LOC. 176) AND TYPE 200G....
ANSWER THE QUESTION :1ST DEVICE : ETC.....
.....THE PROGRAM WILL CONTINUE WITH DEVICE 0

7.2.2.3 IF ALL DEVICES ARE DISQUALIFIED BY MISTAKE THE PROGRAM WILL TYPEOUT AN ERROR MESSAGE.....TYPE 200G.

7.3 CABLE DELAYS

NOTE: EXTERNAL LOOP BACK TESTS ONLY (MODEM CABLE WITH H315 CONNECTOR ON)

7.3.1 TO PROVIDE SUFFICIENT DELAY FOR CLOCK SIGNAL OVER THE CABLE, LOCATION "HOLD:" MUST BE MODIFIED TO ACCOMODATE FOR FASTER MACHINES. PRESENTLY "HOLD:" =20 IS SUFFICIENT TIME ON AN 11/03 MACHINE.

BASICALLY DON'T TRY TO EXCEED 10K TO 12K RATE USING THE EIA DRIVERS

7.4 TO USE THE "XOR" TESTER, THE BRANCH AROUND THE "XOR" CODE MUST BE PATCHED TO A "NOP". (SEE LISTINGS FOR DETAILS)

K01

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 11
DZDUR1.M11 02-FEB-77 08:16

SEQ 0010

8. DEFAULT PARAMETERS:
 1ST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- RXCSR: 160010
 VECTOR ADDRESS- DURIV: 770
 ARE YOU RUNNING MULTIPLE DEVICES ?- NO MULTD: 0
 LAST DEVICE: RECEIVER CONTROL REGISTER ADDRESS- LASTADD: 0
 # OF SYNC CHARS SELECTED - 2 SYNCNO: 377
 IS SEC XMIT SWITCH E55-2 ON?- YES SEXMIT: 377
 IS SEC REC SWITCH E55-3 ON?- YES SEREC: 377
 IS OPT CLR ENABLE SWITCH E55-1 ON?- YES OPTCLR: 377
 DO YOU HAVE THE EXTERNAL MODEM BYPASS JUMPER
 CONNECTOR ON (H315)- YES JMRBY: 377

9. PROGRAM DESCRIPTION

9.1 THIS PROGRAM PERFORMS THE OFFLINE RECEIVER SECTION TESTING
 OF THE DEVICE
 SEE LISTING FOR DETAILS

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

11. LISTINGS

*
 .REM *
 *
 .REM *
 *

521
522
523
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
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576

.SBTTL APT COMMUNICATIONS ROUTINE

::*****

000002	112767	000001	000236	\$ATY1:	MOVB	#1,\$FFLG	::TO REPORT FATAL ERROR
000010	112767	000001	000226	\$ATY3:	MOVB	#1,\$MFLG	::TO TYPE A MESSAGE
000016	000403				BR	\$ATYC	
000020	112767	000001	000220	\$ATY4:	MOVB	#1,\$FFLG	::TO ONLY REPORT FATAL ERROR
000026				\$ATYC:			
000026	010046				MOV	RO,-(SP)	::PUSH RO ON STACK
000030	010146				MOV	R1,-(SP)	::PUSH R1 ON STACK
000032	105767	000206			TSTB	\$MFLG	::SHOULD TYPE A MESSAGE?
000036	001450				REQ	\$S	::IF NOT: BR
000040	122767	000001	001500		CMPB	#APTENV,\$ENV	::OPERATING UNDER APT?
000046	001031				BNE	\$S	::IF NOT: BR
000050	132767	000100	001471		RTTB	#APTPOOL,\$ENVM	::SHOULD SPOOL MESSAGES?
000056	001425				BEQ	\$S	::IF NOT: BR
000060	017600	000004			MOV	#4(SP),RO	::GET MESSAGE ADDR.
000064	062766	000002	000004		ADD	#2,4(SP)	::BUMP RETURN ADDR.
000072	005767	001430		1\$:	TST	\$MSGTYPE	::SEE IF DONE W/ LAST XMISSION?
000076	001375				BNE	1\$::IF NOT: WAIT
000100	010067	001436			MOV	RO,\$MSGAD	::PUT ADDR IN MAILBOX
000104	105720			2\$:	TSTB	(RO)+	::FIND END OF MESSAGE
000106	001376				BNE	2\$	
000110	166700	001426			SUB	\$MSGAD,RO	::SUB START OF MESSAGE
000114	006200				ASR	RO	::GET MESSAGE LNGLTH IN WORDS
000116	010067	001422			MOV	RO,\$MSGLGT	::PUT LENGTH IN MAILBOX
000122	012767	000004	001376		MOV	#4,\$MSGTYPE	::TELL APT TO TAKE MSG.
000130	000413				BR	\$S	
000132	017667	000004	000016	3\$:	MOV	#4(SP),4\$::PUT MSG ADDR IN JSR LINKAGE
000140	062766	000002	000004		ADD	#2,4(SP)	::BUMP RETURN ADDRESS
000146	016746	177624			MOV	177776,-(SP)	::PUSH 177776 ON STACK
000152	004767	012556			JSR	PC,\$TYPE	::CALL TYPE MACRO
000156	000000			4\$:	.WORD	0	
000160				5\$:			
000160	105767	000062		10\$:	TSTB	\$FFLG	::SHOULD REPORT FATAL ERROR?
000164	001416				BEQ	12\$::IF NOT: BR
000166	005767	001354			TST	\$ENV	::RUNNING UNDER APT?
000172	001413				BEQ	12\$::IF NOT: BR
000174	005767	001326		11\$:	TST	\$MSGTYPE	::FINISHED LAST MESSAGE?
000200	001375				BNE	11\$::IF NOT: WAIT
000202	017667	000004	001320		MOV	#4(SP),\$FATAL	::GET ERROR #
000210	062766	000002	000004		ADD	#2,4(SP)	::BUMP RETURN ADDR.
000216	005267	001304			INC	\$MSGTYPE	::TELL APT TO TAKE ERROR
000222	105067	000020		12\$:	CLRB	\$FFLG	::CLEAR FATAL FLAG

M01

DZDUR-A MACY11 27(1006) 19-APR-77
DZDUR2.M11 02-FEB-77 08:19

10:06 PAGE 14
APT COMMUNICATIONS ROUTINE

SEQ 0012

577	000226	105067	000013
578	000232	105067	000006
579	000236	012601	
580	000240	012600	
581	000242	000207	
582	000244	000	
583	000245	000	
584	000246	000	
585		000250	
586		000200	
587		000001	
588		000100	
589		000040	
590		000001	
591			
592			
593			
594			
595			
596			
597			
598			
599			
600			
601			
602			
603			
604			
605			
606			
607			
608			
609			
610			
611			

CLRB	\$LFLG	:: CLEAR LOG FLAG
CLFB	\$MFLG	:: CLEAR MESSAGE FLAG
MOV	(SP)+,R1	:: POP STACK INTO R1
MOV	(SP)+,R0	:: POP STACK INTO R0
RTS	PC	:: RETURN
\$MFLG:	.BYTE	0
\$LFLG:	.BYTE	0
\$FFLG:	.BYTE	0
	.EVEN	
APTSIZE=200		
APTENV=001		
APTSPool=100		
APTCSUP=040		
\$TN=1		

```

:: CLEAR LOG FLAG
:: CLEAR MESSAGE FLAG
:: POP STACK INTO R1
:: POP STACK INTO R0
:: RETURN
:: MESSG. FLAG
:: LOG FLAG
:: FATAL FLAG

```

612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667

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 DZDUR-A TAPE B
;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

;STARTING PROCEDURE
;TYPE 200G
;PROGRAM WILL TYPE "DUV11 DZDUR-A TAPE B "
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE B"
;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 IOT,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
DDISP= 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
PR0= 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
```


668	020000	SW13=	20000
669	010000	SW12=	10000
670	004000	SW11=	4000
671	002000	SW10=	2000
672	001000	SW09=	1000
673	000400	SW08=	400
674	000200	SW07=	200
675	000100	SW06=	100
676	000040	SW05=	40
677	000020	SW04=	20
678	000010	SW03=	10
679	000004	SW02=	4
680	000002	SW01=	2
681	000001	SW00=	1
682		.EQUIV	SW09, SW9
683		.EQUIV	SW08, SW8
684		.EQUIV	SW07, SW7
685		.EQUIV	SW06, SW6
686		.EQUIV	SW05, SW5
687		.EQUIV	SW04, SW4
688		.EQUIV	SW03, SW3
689		.EQUIV	SW02, SW2
690		.EQUIV	SW01, SW1
691		.EQUIV	SW00, SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

694	100000	BIT15=	100000
695	040000	BIT14=	40000
696	020000	BIT13=	20000
697	010000	BIT12=	10000
698	004000	BIT11=	4000
699	002000	BIT10=	2000
700	001000	BIT09=	1000
701	000400	BIT08=	400
702	000200	BIT07=	200
703	000100	BIT06=	100
704	000040	BIT05=	40
705	000020	BIT04=	20
706	000010	BIT03=	10
707	000004	BIT02=	4
708	000002	BIT01=	2
709	000001	BIT00=	1
710		.EQUIV	BIT09, BIT9
711		.EQUIV	BIT08, BIT8
712		.EQUIV	BIT07, BIT7
713		.EQUIV	BIT06, BIT6
714		.EQUIV	BIT05, BIT5
715		.EQUIV	BIT04, BIT4
716		.EQUIV	BIT03, BIT3
717		.EQUIV	BIT02, BIT2
718		.EQUIV	BIT01, BIT1
719		.EQUIV	BIT00, BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

721		ERRVEC=	4	:: TIME OUT AND OTHER ERRORS
722	000004	RESVEC=	10	:: RESERVED AND ILLEGAL INSTRUCTIONS
723	000010			

724	000014	TBITVEC=14	:: "T" BIT
725	000014	TRTVEC= 14	:: TRACE TRAP
726	000014	BPTVEC= 14	:: BREAKPOINT TRAP (BPT)
727	000020	IOTVEC= 20	:: INPUT/OUTPUT TRAP (IOT) **SCOPE**
728	000024	PWRVEC= 24	:: POWER FAIL
729	000030	EMTVEC= 30	:: EMULATOR TRAP (EMT) **ERROR**
730	000034	TRAPVEC=34	:: "TRAP" TRAP
731	000060	TKVEC= 60	:: TTY KEYBOARD VECTOR
732	000064	TPVEC= 64	:: TTY PRINTER VECTOR
733	000240	PIRQVEC=240	:: PROGRAM INTERRUPT REQUEST VECTOR

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 19
 DZDURA.M11 12-OCT-76 10:29 BASIC DEFINITIONS

SEQ 0016

```

734                                     ;STANDARD INTERRUPT VECTORS
735
736
737                                     .=174
738 000174 000000  DISPREG:0
739 000176 000000  SWREG:0
740                                     .=200
741 000200 000167 001746  JMP      .START      ;GO TO START OF PROGRAM
742
743
744
745                                     .=1100
746 001100 000000  .WORD 0
747 001102 177570  LIGHTS:177570
748
749
750                                     ;PROGRAM CONTROL PARAMETERS
751
752
753 001104 000000  RETURN: 0
754 001106 000000  NEXT: 0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
755 001110 000000  LOCK: 0      ;ADDRESS FOR LOCK ON CURRENT DATA
756 001112 000000  PASCNT: 0   ;ADDRESS CONTAINING PASS COUNT
757 001114 000000  ERRCNT: 0   ;ERROR COUNT
758 001116 000000  SAVSP: 0    ;STACK POINTER STORAGE
759
760                                     ;PROGRAM VARIABLES
761
762 001120 000020  HOLD: 20    ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
763 001122 000000  SHIFT: 0    ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
764 001124 000000  COUNT: 0    ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
765 001126 000000  SAVPC: 0    ;PROGRAM COUNTER STORAGE
766 001130 000000  HLD0: 0
767 001132 000000  HLD1: 0
768 001134 000000  HLD2: 0
769 001136 000000  HLD3: 0
770 001140 000000  HLD4: 0
771 001142 000000  HLD5: 0
772 001144 000000  HLD6: 0
773

```



```

774 ;PROGRAM CONVERSATIONAL PARAMETERS
775 001146 377 SYNCNO: .BYTE 377 ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
776 001147 377 SEXMIT: .BYTE 377 ;SEC XMIT JUMPER "IN"
777 001150 377 SEREC: .BYTE 377 ;SEC REC JUMPER "IN"
778 001151 377 OPTCLR: .BYTE 377 ;OPTIONAL JUMPER CLR "IN"
779 001152 000 MULTD: .BYTE 0 ;NO MULTIPLE DEVICE FLAG
780 001153 377 JMRBY: .BYTE 377 ;EXTERNAL MODEM BYPASS JUMPER "IN"
781 .EVEN
782
783 ;PROGRAM MULTIPLE DEVICE PARAMETERS
784 001154 000000 BASEADD: 0 ;PROG CONTROLLED 1ST DEVICE ADDR
785 001156 000000 KEEPADD: 0 ;SAVED 1ST DEVICE ADDR
786 001160 000000 LASTADD: 0 ;LAST DEVICE RXCSR ADDR
787 001162 000000 BASEIV: 0 ;PROG CONTROLLED IV
788 001164 000000 KEEPIV: 0 ;SAVED INTR VECTOR
789 001166 000000 ACTREG: 0 ;ACTIVE REGISTER , MODIFY THIS
790 ;LOCATION TO DISQUALIFY OR QUALIFY
791 ;DEVICES (1= RUN , 0= DON'T RUN)
792 001170 000000 ROTADD: 0 ;ROTATING POINTER FOR ACTREG..POINTS
793 ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES
794
795 ;PROGRAM CONTROL FLAGS
796
797 001172 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
798 001173 000 STFLG: .BYTE 0 ;TEST START FLAG
799 001174 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
800 001176
801 001400
802
803

```

```

804
805
806
807           ; INSTRUCTION DEFINITIONS
808
809           005746   PUSH1SP=5746   ; DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
810           005726   POP1SP=5726   ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
811           010046   PUSHRO=10046  ; SAVE RO ON STACK =MOV RO, -(SP)
812           012600   POPRO=12600   ; RESTORE RO FROM STACK =MOV (SP)+, RO
813           024646   PUSH2SP=24646 ; DECREMENT STACK TWICE =CMP -(SP), -(SP)
814           022626   POP2SP=22626  ; INCREMENT STACK TWICE =CMP (SP)+, (SP)+
815           ; REGISTER DEFINITIONS
816           ; RXCSR BIT DEFINITIONS
817           100000   DSC=BIT15   ; DATA SET CHANGE
818           040000   RING=BIT14   ; RING
819           020000   CTS=BIT13   ; CLR TO SEND
820           010000   CARDET=BIT12 ; CARRIER DETECT
821           004000   RECACT=BIT11 ; REC ACTIVE
822           002000   SRD=BIT10   ; SEC REC DATA
823           001000   DSR=BIT9    ; DATA SET RDY
824           000400   STPSYN=BIT8  ; STRIP SYNC
825           000200   RXDONE=BIT7  ; REC DONE
826           000100   RINTEN=BIT6  ; REC INTR ENABLE
827           000040   DSINTE=BIT5  ; DSC INTR ENABLE
828           000020   SYN SCH=BIT4  ; SYNC SEARCH
829           000010   STD=BIT3    ; SEC XMIT DATA
830           000004   RTS=BIT2    ; REQ TO SEND
831           000002   DTR=BIT1    ; DATA TERM RDY
832           000001   VOID=BIT0
833           ; RXDBUF BIT DEFINITIONS
834           100000   RXERR=BIT15  ; REC ERROR
835           040000   OVRUN=BIT14  ; OVERRUN
836           020000   FRMERR=BIT13 ; FRAME ERROR
837           010000   PARER=BIT12  ; PARITY ERROR
838           ; PARCSR BIT DEFINITIONS
839           001000   PAREN=BIT9   ; PARITY ENABLE
840           000400   EVPAR=BIT8  ; EVEN PARITY SENSE
841           ; PARCSR WRD DEFINITIONS
842           030000   SYNINT=30000 ; SYNC EXTERNAL MODE
843           020000   SYNEXT=20000 ; SYNC INTERNAL MODE
844           000000   ISYMOD=0    ; ISOC MODE
845           000000   FIVE=0     ; WORD LENGTH 5 BITS
846           002000   SIX=2000   ; WORD LENGTH 6 BITS
847           004000   SEVEN=4000 ; WORD LENGTH 7 BITS
848           006000   EIGHT=6000 ; WORD LENGTH 8 BITS
849           000000   NOPAR=0    ; NO PARITY
850           001000   ODDPAR=1000 ; ODD PARITY
851           001400   EVEPAR=1400 ; EVEN PARITY
852           ; TXCSR BIT DEFINITIONS
853           100000   DNA=BIT15   ; DATA NOT AVAILABLE
854           040000   MTDATA=BIT14 ; MAINT DATA
855           020000   CLK=BIT13   ; CLK
856           002000   BITW=BIT10  ; BIT WINDOW
857           000400   MRESET=BIT8  ; MASTER RESET
858           000200   TXDONE=BIT7  ; XMIT DONE
859           000100   TXINTE=BIT6 ; XMIT INTR ENABLE

```

G02

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 22
DZDURA.M11 12-OCT-76 10:29 BASIC DEFINITIONS

SEQ 0019

860	000040	ONAINTE=BITS	;DNA INTR ENAB
861	000020	SEND=BIT4	;SEND
862	000010	HDXEN=BIT3	;HDX/FDX
863	000001	BREAK=BIT0	;BREAK
864		;TXCSR WRD DEFINITIONS	
865	000000	USER=0	;USER MODE
866	004000	MINT=4000	;MAINT INT MODE
867	010000	MEXT=10000	;MAINT EXT MODE
868	014000	SYSTST=14000	;SYSTEM TEST MODE

869
870
871
872
873
874
875 001400
876 001400
877 001400 000000
878 001402 000
879 001403 000
880 001404 000000
881 001406 000000
882 001410 000000
883 001412 000000
884 001414 000
885 001415 001
886 001416 000000
887 001420 000000
888 001422 000000
889 001424 000000
890 001426 000000
891 001430 000000
892 001432 000000
893 001434 000
894 001435 000
895 001436 000000
896 001440 177570
897 001442 177570
898 001444 177560
899 001446 177562
900 001450 177564
901 001452 177566
902 001454 000
903 001455 002
904 001456 012
905 001457 000
906 001460 000000
907
908 001462 000000
909 001464 000000
910 001466 000000
911 001470 000000
912 001472 000000
913 001474 000000
914 001476 000000
915 001500 000000
916 001502 000000
917 001504 000000
918 001506 000000
919 001510 000000
920 001512 000000
921 001514 000000
922 001516 177607 000377
923 001522 077
924 001523 015

.SBTTL COMMON TAGS

; THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
; USED IN THE PROGRAM.

```

SCMTAG:      =.                ;; START OF COMMON TAGS
              .WORD           0
STSTNM:      .BYTE           00    ;; CONTAINS THE TEST NUMBER
SERFLG:      .BYTE           00    ;; CONTAINS ERROR FLAG
SICHT:       .WORD           00    ;; CONTAINS SUBTEST ITERATION COUNT
SLPADR:      .WORD           00    ;; CONTAINS SCOPE LOOP ADDRESS
SLPERR:      .WORD           00    ;; CONTAINS SCOPE RETURN FOR ERRORS
SERTTL:      .WORD           00    ;; CONTAINS TOTAL ERRORS DETECTED
SITEMB:      .BYTE           00    ;; CONTAINS ITEM CONTROL BYTE
SERMAX:      .BYTE           01    ;; CONTAINS MAX. ERRORS PER TEST
SERRPC:      .WORD           00    ;; CONTAINS PC OF LAST ERROR INSTRUCTION
SGDADR:      .WORD           00    ;; CONTAINS ADDRESS OF 'GOOD' DATA
SBDADR:      .WORD           00    ;; CONTAINS ADDRESS OF 'BAD' DATA
SGDDAT:      .WORD           00    ;; CONTAINS 'GOOD' DATA
SBDDAT:      .WORD           00    ;; CONTAINS 'BAD' DATA
              .WORD           00    ;; RESERVED--NOT TO BE USED
              .WORD           00
SAUTOB:      .BYTE           00    ;; AUTOMATIC MODE INDICATOR
SINTAG:      .BYTE           00    ;; INTERRUPT MODE INDICATOR
              .WORD           0
SWR:         .WORD           0SWR  ;; ADDRESS OF SWITCH REGISTER
DISPLAY:     .WORD           0DISP  ;; ADDRESS OF DISPLAY REGISTER
STKS:        177560           ;; TTY KBD STATUS
STKB:        177562           ;; TTY KBD BUFFER
STPS:        177564           ;; TTY PRINTER STATUS REG. ADDRESS
STPB:        177566           ;; TTY PRINTER BUFFER REG. ADDRESS
SNUL:        .BYTE           0     ;; CONTAINS NULL CHARACTER FOR FILLS
SFILLS:      .BYTE           2     ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
SFILLC:      .BYTE           12    ;; INSERT FILL CHARS. AFTER A "LINE FEED"
STPFLG:      .BYTE           0     ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
SREGAD:      .WORD           0     ;; CONTAINS THE ADDRESS FROM WHICH ($REGO) WAS OBTAINED
SREGO:       .WORD           0     ;; CONTAINS ((SREGAD)+0)
SREG1:       .WORD           0     ;; CONTAINS ((SREGAD)+2)
SREG2:       .WORD           0     ;; CONTAINS ((SREGAD)+4)
SREG3:       .WORD           0     ;; CONTAINS ((SREGAD)+6)
SREG4:       .WORD           0     ;; CONTAINS ((SREGAD)+10)
SREG5:       .WORD           0     ;; CONTAINS ((SREGAD)+12)
STMP0:       .WORD           0     ;; USER DEFINED
STMP1:       .WORD           0     ;; USER DEFINED
STMP2:       .WORD           0     ;; USER DEFINED
STMP3:       .WORD           0     ;; USER DEFINED
STMP4:       .WORD           0     ;; USER DEFINED
STMP5:       .WORD           0     ;; USER DEFINED
STIMES:      0                ;; MAX. NUMBER OF ITERATIONS
SESCAPE:     0                ;; ESCAPE ON ERROR ADDRESS
SBELL:       .ASCIZ <207><377><377> ;; CODE FOR BELL
SQUES:       .ASCII /?/       ;; QUESTION MARK
SCRLF:       .ASCII <15>      ;; CARRIAGE RETURN

```

925 001524 000012
 926
 927
 928
 929
 930
 931 001526
 932 001526 000000
 933 001530 000000
 934 001532 000000
 935 001534 000000
 936 001536 000000
 937 001540 000000
 938 001542 000000
 939 001544 000000
 940 001546
 941 001546 000
 942 001547 000
 943 001550 000000
 944 001552 000000
 945 001554 000000
 946
 947
 948
 949
 950
 951
 952 001556 000
 953 001557 000
 954
 955
 956
 957
 958 001560 000000
 959
 960 001562 000
 961 001563 000
 962 001564 000000
 963 001566 000
 964 001567 000
 965 001570 000000
 966 001572 000
 967 001573 000
 968 001574 000000
 969 001576 000000
 970 001600 000000
 971 001602 000000
 972 001604 000000
 973 001606 000000
 974 001610 000000
 975 001612 000000
 976 001614 000000
 977 001616 000000
 978 001620 000000
 979 001622 000000
 980 001624 000000

\$LF: .ASCIZ <12> ;:LINE FEED
 ;:*****
 ;SBTTL APT MAILBOX-ETABLE
 ;:*****
 ;EVEN
 \$MAIL: ;:APT MAILBOX
 \$MSGTY: .WORD AMSGTY ;:MESSAGE TYPE CODE
 \$FATAL: .WORD AFATAL ;:FATAL ERROR NUMBER
 \$TESTN: .WORD ATESTN ;:TEST NUMBER
 \$PASS: .WORD APASS ;:PASS COUNT
 \$DEVCT: .WORD ADEVCT ;:DEVICE COUNT
 \$UNIT: .WORD AUNIT ;:I/O UNIT NUMBER
 \$MSGAD: .WORD AMSGAD ;:MESSAGE ADDRESS
 \$MSGLG: .WORD AMSGLG ;:MESSAGE LENGTH
 \$ETABLE: ;:APT ENVIRONMENT TABLE
 \$ENV: .BYTE AENV ;:ENVIRONMENT BYTE
 \$ENVM: .BYTE AENVM ;:ENVIRONMENT MODE BITS
 \$SWREG: .WORD ASWREG ;:APT SWITCH REGISTER
 \$USWR: .WORD AUSWR ;:USER SWITCHES
 \$CPUOP: .WORD ACPUOP ;:CPU TYPE, OPTIONS
 ;:BITS 15-11=CPU TYPE
 ;: 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
 ;: 11/70=06, PDQ=07, Q=10
 ;:BIT 10=REAL TIME CLOCK
 ;:BIT 9=FLOATING POINT PROCESSOR
 ;:BIT 8=MEMORY MANAGEMENT
 \$MAMS1: .BYTE AMAMS1 ;:HIGH ADDRESS, M.S. BYTE
 \$MTYP1: .BYTE AMTYP1 ;:MEM. TYPE, BLK#1
 ;:MEM. TYPE BYTE -- (HIGH BYTE)
 ;: 900 NSEC CORE=001
 ;: 300 NSEC BIPOLAR=002
 ;: 500 NSEC MOS=003
 \$MADR1: .WORD AMADR1 ;:HIGH ADDRESS, BLK#1
 ;:MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
 \$MAMS2: .BYTE AMAMS2 ;:HIGH ADDRESS, M.S. BYTE
 \$MTYP2: .BYTE AMTYP2 ;:MEM. TYPE, BLK#2
 \$MADR2: .WORD AMADR2 ;:MEM. LAST ADDRESS, BLK#2
 \$MAMS3: .BYTE AMAMS3 ;:HIGH ADDRESS, M.S. BYTE
 \$MTYP3: .BYTE AMTYP3 ;:MEM. TYPE, BLK#3
 \$MADR3: .WORD AMADR3 ;:MEM. LAST ADDRESS, BLK#3
 \$MAMS4: .BYTE AMAMS4 ;:HIGH ADDRESS, M.S. BYTE
 \$MTYP4: .BYTE AMTYP4 ;:MEM. TYPE, BLK#4
 \$MADR4: .WORD AMADR4 ;:MEM. LAST ADDRESS, BLK#4
 \$VECT1: .WORD AVECT1 ;:INTERRUPT VECTOR#1, BUS PRIORITY#1
 \$VECT2: .WORD AVECT2 ;:INTERRUPT VECTOR#2, BUS PRIORITY#2
 \$BASE: .WORD ABASE ;:BASE ADDRESS OF EQUIPMENT UNDER TEST
 \$DEVN: .WORD ADEVN ;:DEVICE MAP
 \$CDW1: .WORD ACDW1 ;:CONTROLLER DESCRIPTION WORD#1
 \$CDW2: .WORD ACDW2 ;:CONTROLLER DESCRIPTION WORD#2
 \$DDW0: .WORD ADDW0 ;:DEVICE DESCRIPTOR WORD#0
 \$DDW1: .WORD ADDW1 ;:DEVICE DESCRIPTOR WORD#1
 \$DDW2: .WORD ADDW2 ;:DEVICE DESCRIPTOR WORD#2
 \$DDW3: .WORD ADDW3 ;:DEVICE DESCRIPTOR WORD#3
 \$DDW4: .WORD ADDW4 ;:DEVICE DESCRIPTOR WORD#4
 \$DDW5: .WORD ADDW5 ;:DEVICE DESCRIPTOR WORD#5


```

997
998
999
1000
1001
1002      005746      PUSH1SP=5746      :DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
1003      005726      POP1SP=5726       :INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
1004      010046      PUSHRO=10046     :SAVE RO ON STACK =MOV RO, -(SP)
1005      012600      POPRO=12600      :RESTORE RO FROM STACK =MOV (SP)+, RO
1006      024646      PUSH2SP=24646    :DECREMENT STACK TWICE =CMP -(SP), -(SP)
1007      022626      POP2SP=22626     :INCREMENT STACK TWICE =CMP (SP)+, (SP)+
1008
1009      ;REGISTER DEFINITIONS
1010      ;RXCSR BIT DEFINITIONS
1011      100000      DSC=BIT 5        :DATA SET CHANGE
1012      040000      RING=BIT14       :RING
1013      020000      CTS=BIT13        :CLR TO SEND
1014      010000      CARDET=BIT12    :CARRIER DETECT
1015      004000      RECACT=BIT11    :REC ACTIVE
1016      002000      SRD=BIT10       :SEC REC DATA
1017      001000      DSR=BIT9        :DATA SET RDY
1018      000400      STPSYN=BIT8     :STRIP SYNC
1019      000200      RXDONE=BIT7     :REC DONE
1020      000100      RINTEN=BIT6     :REC INTR ENABLE
1021      000040      DSINTE=BIT5     :DSC INTR ENABLE
1022      000020      SYN SCH=BIT4    :SYNC SEARCH
1023      000010      STD=BIT3       :SEC XMIT DATA
1024      000004      RTS=BIT2       :REG TO SEND
1025      000002      DTR=BIT1       :DATA TERM RDY
1026      000001      VOID=BIT0
1027      ;RXDBUF BIT DEFINITIONS
1028      100000      RXERR=BIT15     :REC ERROR
1029      040000      OVRUN=BIT14    :OVERRUN
1030      020000      FRMERR=BIT13   :FRAME ERROR
1031      010000      PARER=BIT12    :PARITY ERROR
1032      ;PARCSR BIT DEFINITIONS
1033      001000      PAREN=BIT9     :PARITY ENABLE
1034      000400      EVPAR=BIT8     :EVEN PARITY SENSE
1035      ;PARCSR WRD DEFINITIONS
1036      030000      SYNINT=30000   :SYNC EXTERNAL MODE
1037      020000      SYNEXT=20000   :SYNC INTERNAL MODE
1038      000000      ISYMOD=0       :ISOC MODE
1039      000000      FIVE=0         :WORD LENGTH 5 BITS
1040      002000      SIX=2000       :WORD LENGTH 6 BITS
1041      004000      SEVEN=4000     :WORD LENGTH 7 BITS
1042      006000      EIGHT=6000     :WORD LENGTH 8 BITS
1043      000000      NOPAR=0        :NO PARITY
1044      001000      ODDPAR=1000    :ODD PARITY
1045      001400      EVEPAR=1400    :EVEN PARITY
1046      ;TXCSR BIT DEFINITIONS
1047      100000      DNA=BIT15      :DATA NOT AVAILABLE
1048      040000      MTDATA=BIT14   :MAINT DATA
1049      020000      CLK=BIT13      :CLK
1050      002000      BITW=BIT10     :BIT WINDOW
1051      000400      MRESET=BIT8    :MASTER RESET
1052      000200      TXDONE=BIT7    :XMIT DONE
1053      000100      TXINTE=BIT6    :XMIT INTR ENABLE

```

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 27
DZDURA.M11 12-OCT-76 10:29 APT MAILBOX-ETABLE

SEQ 0024

1053	000040	DNAINTE=BITS	;DNA INTR ENAB
1054	000020	SEND=BIT4	;SEND
1055	000010	HDXEN=BIT3	;HDX/FDX
1056	000001	BREAK=BIT0	;BREAK
1057		;TXCSR WRD DEFINITIONS	
1058	000000	USER=0	;USER MODE
1059	004000	MINT=4000	;MAINT INT MODE
1060	010000	MEXT=10000	;MAINT EXT MODE
1061	014000	SYSTST=14000	;SYSTEM TEST MODE

1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117

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

001712 160010
001714 160011
001716 160012
001720 160013
001722 160012
001724 160013
001726 160014
001730 160015
001732 160016
001734 160017

001736 000770
001740 000772
001742 000774
001744 000776

001746 020040 051105 047522
001754 020122 041520 000040
001762 020040 047503 050115
001770 051101 051511 047117
001776 042440 051122 051117
002004 047440 020116 042522

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

;ERROR TABLE
EM1 ;ERROR 1 REGISTER ERROR
DH1
DT1
DF1
EM2 ;ERROR 2 RECEIVER ERROR
DH1
DT1
DF1
EM3 ;ERROR 3 TRANSMITTER ERROR
DH1
DT1
DF1
EM4 ;ERROR 4 BIT ERROR (GENERAL)
0
DT4
DF1

;DEFAULT DU ADDRESSES

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

;DEFAULT DU VECTORS

DURIV: 770 ;REC INTR VECTOR
DURIS: 772 ;REC INTR STATUS
DUTIV: 774 ;XMIT INTR VECTOR
DUTIS: 776 ;XMIT INTR STATUS

;ERROR MESSAGES

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

```

1118 002012 044507 052123 051105
1119 002020 000123
1120 002022 020040 042522 042503 EM2: .ASCIZ / RECEIVER ERROR/
1121 002030 053111 051105 042440
1122 002036 051122 051117 000
1123 002043 040 052040 040522 EM3: .ASCIZ / TRANSMITTER ERROR/
1124 002050 051516 044515 052124
1125 002056 051105 042440 051122
1126 002064 051117 000
1127 ;DATA HEADERS FOR ERROR MESSAGES
1128 002067 105 051122 041520 DH1: .ASCIZ /ERRPC WANTED ACTUAL/
1129 002074 020040 040527 052116
1130 002102 042105 020040 041501
1131 002110 052524 046101 000
1132 .EVEN
1133 ;DATA TABLES FOR ERROR MESSAGES
1134 002116 001416 001130 001132 DT1: .WORD $ERRPC,HLD0,HLD1,0
1135 002124 000000
1136
1137 002126 001416 000000 DT4: .WORD $ERRPC,0
1138
1139 002132 000 000 000 DF1: .BYTE 0,0,0,0
1140 002135 000
1141 .EVEN
1142 .SBTTL ACT11 HOOKS
1143
1144 ;:*****
1145 ;HOOKS REQUIRED BY ACT11
1146 002136 $SVPC= ;SAVE PC
1147 003046 =46
1148 000046 012532 $ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
1149 000052 =52
1150 000052 000000 .WORD 0 ;;2)SET LOC.52 TO ZERO
1151 002136 =$SVPC ;; RESTORE PC
1152 .SBTTL APT PARAMETER BLOCK
1153
1154 ;:*****
1155 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1156 ;:*****
1157 002136 .SX= ;SAVE CUF ENT LOCATION
1158 000024 =24 ;SET POWER FAIL TO POINT TO START OF PROGRAM
1159 000024 200 ;FOR APT START UP
1160 000044 =44 ;POINT TO APT INDIRECT ADDRESS PNTR.
1161 000044 $APTHDR ;POINT TO APT HEADER BLOCK
1162 002136 =.SX ;RESET LOCATION COUNTER
1163 ;:*****
1164 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC
1165 ;INTERFACE SPEC.
1166
1167 002136 $APTHD:
1168 002136 $SHFTS: .WORD 0 ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1169 002140 $ENDAD: .WORD $MAIL ;ADDRESS OF APT MAILBOX (BITS 0-15)
1170 002142 $SYSTH: .WORD 10 ;RUN TIME OF LONGEST TEST
1171 002144 $PASTH: .WORD 10 ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1172 002146 $UNITH: .WORD ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1173 002150 ;SETENC-$MAIL/? ;LENGTH MAILBOX-ETABLE(WORDS)

```

```

1174
1175
1176 ;PROGRAM INITIALIZATION
1177 ;LOCK OUT INTERRUPTS
1178 ;SET UP PROCESSOR STACK
1179 ;SET UP POWER FAIL VECTOR
1180 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1181 ;TYPE TITLE MESSAGE
1182
1183 002152 .START:
1184 .SBTTL INITIALIZE THE COMMON TAGS
1185 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1186 002152 012706 001400 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
1187 002156 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
1188 002160 022706 001440 CMP #SWR,R6 ;;DONE?
1189 002164 001374 BNE -6 ;;LOOP BACK IF NO
1190 002166 012706 001100 MOV ##STACK,SP ;;SETUP THE STACK POINTER
1191 ;;INITIALIZE A FEW VECTORS
1192 002172 012737 016162 000020 MOV $$SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1193 002200 012737 000340 000022 MOV #340,@IOTVEC+2 ;;LEVEL 7
1194 002206 012737 014052 000030 MOV #ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1195 002214 012737 000340 000032 MOV #340,@EMTVEC+2 ;;LEVEL 7
1196 002222 012737 016500 000034 MOV #TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1197 002230 012737 000340 000036 MOV #340,@TRAPVEC+2 ;;LEVEL 7
1198 002236 012737 014654 000024 MOV #SPWRDN,@PWRVEC ;;POWER FAILURE VECTOR
1199 002244 012737 000340 000026 MOV #340,@PWRVEC+2 ;;LEVEL 7
1200 002252 005067 177234 CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
1201 002256 005067 177232 CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1202 002262 112767 000001 177125 MOVB #1,$SERMAX ;;ALLOW ONE ERROR PER TEST
1203 002270 012767 002270 177110 MOV #,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1204 002276 012767 002276 177104 MOV #,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS
1205 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1206 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
1207 002304 013746 000004 MOV @ERRVEC-(SP) ;;SAVE ERROR VECTOR
1208 002310 012737 002344 000004 MOV #64,$ERRVEC ;;SET UP ERROR VECTOR
1209 002316 012767 177570 177114 MOV #DSWR,$SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
1210 002324 012767 177570 177110 MOV #DDISP,$DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1211 002332 022777 177777 177100 CMP #-1,$SWR ;;TRY TO REFERENCE HARDWARE SWR
1212 002340 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1213 ;;AND THE HARDWARE SWR IS NOT = -1
1214 002342 000403 BR 65$ ;;BRANCH IF NO TIMEOUT
1215 002344 012716 002352 64$: MOV #65,$(SP) ;;SET UP FOR TRAP RETURN
1216 002350 000002 RTI
1217 002352 012767 000176 177060 65$: MOV #SWREG,$SWR ;;POINT TO SOFTWARE SWR
1218 002360 012767 000174 177054 MOV #DISPREG,$DISPLAY
1219 002366 012637 000004 66$: MOV (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
1220
1221 002372 005067 177136 CLR $PASS ;;CLEAR PASS COUNT
1222 002376 132767 000200 177143 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
1223 002404 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
1224 002406 012767 001550 177024 MOV #$$SWREG,$SWR ;;NO,USE APT SWITCH REGISTER
1225 002414 67$:
1226 002414 012706 001100 MOV #STACK,SP ;;SET STACK
1227 002420 106427 000340 MTPS #340 ;;LOCK INTERRUPTS
1228 002424 012737 014654 000024 MOV #.PFAIL,@#24 ;;SET UP POWER FAIL VECTOR
1229 002432 105067 176535 CLRB $TFLG ;;CLEAR START FLAG
    
```

```

1230 002436 005067 176450 CLR PASCNT ;CLEAR PASS COUNT
1231 002442 105067 176735 CLRB SERFLG ;CLEAR ERROR FLAG
1232 002446 005067 176740 CLR SERTTL ;CLEAR ERROR COUNT
1233 002452 005067 176740 CLR SERRPC ;CLEAR LAST ERROR POINTER
1234 002456 012767 000001 176716 MOV #1,STSTNM ;SET UP FOR TEST 1
1235 002464 012767 002152 176412 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
1236 ;TESTING STARTS
1237 002472 013746 000006 MOV @#6,-(SP)
1238 002476 013746 000004 MOV @#4,-(SP)
1239 002502 012737 002516 000004 MOV #15,@#4
1240 002510 005777 176724 TST @SWR
1241 002514 000407 BR 2$
1242 002516 012767 000176 176714 1$: MOV #SWREG,SWR
1243 002524 012767 000174 176710 MOV #DISPREG,DISPLAY
1244 002532 022626 CMP (SP)+,(SP)+
1245 002534 012637 000004 2$: MOV (SP)+,@#4
1246 002540 012637 000006 MOV (SP)+,@#6
1247 002544 022767 000176 176666 CMP #SWREG,SWR
1248 002552 001007 BNE 3$
1249 002554 005737 000042 TST @#42 ;CHECK FOR CHAIN
1250 002560 001402 BEQ 33$
1251 002562 000167 000522 JMP .BEGIN
1252 002566 004767 010046 33$: JSR PC,CNTLU
1253 002572 105767 176374 3$: TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1254 002576 001004 BNE ONCE
1255 002600 104401 015014 TYPE #TITLE ;TYPE TITLE MESSAGE
1256 002604 105167 176362 COMB INIFLG ;IF NOT SET FLAG AND DO
1257 002610 105767 176732 ONCE: TSTB SENV ;APT CONTROL?
1258 002614 001410 BEQ 11$ ;BR IF NO
1259 0026 5 032767 000001 176726 BIT #1,SUSWR ;EXTENAL JUMPER ON?
1260 002624 001002 BNE 12$ ;NO
1261 002626 105067 176321 CLRB JMRBY ;CLEAR FLAG
1262 002632 000167 000452 12$: JMP .BEGIN ;GO DO IT
1263 002636 032777 000001 176574 11$: BIT #SW00,@SWR ;RESELECT VECTOR & CONTROL REG?
1264 002644 001002 BNE 1$
1265 002646 000167 000436 JMP .BEGIN
1266 002652 012700 000300 1$: MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1267 002656 012701 000302 MOV #302,R1 ;START AT LOCATION 300
1268 002662 012702 000004 MOV #4,R2
1269 002666 010110 2$: MOV R1,(R0)
1270 002670 005011 CLR (R1)
1271 002672 060200 ADD R2,R0
1272 002674 060201 ADD R2,R1
1273 002676 022701 001000 CMP #1000,R1 ;END AT LOCATION 776
1274 002702 002771 BLT 2$
1275 002704 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1276 002706 015062 MREGAD ;MESSAGE
1277 002710 104410 PARAM ;CONVERT STRING
1278 002712 160000 ;LOW LIMIT
1279 002714 167776 ;HIGH LIMIT
1280 002716 016774 DUBASE ;STORE AT THIS LOCATION
1281 002720 001 .BYTE 1 ;MASK
1282 002721 001 .BYTE 1 ;HOW MANY TIMES + 2
1283 002722 016767 014046 176226 MOV DUBASE,KEEPADD ;SAVE
1284 002730 004767 013706 JSR PC,DUADDR
1285 002734 016767 176216 176212 MOV KEEPADD,BASEADD ;RESTORE FOR ROTATION
    
```


1286	002742	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1287	002744	015047				MVECTO	: MESSAGE
1288	002746	104410				PARAM	: CONVERT STRING
1289	002750	000300				300	: LOW LIMIT
1290	002752	000776				776	: HIGH LIMIT
1291	002754	001736				DURIV	: STORE AT THIS LOCATION
1292	002756	001			.BYTE	1	: MASK
1293	002757	004			.BYTE	4	: HOW MANY TIMES + 2
1294	002760	016767	176752	176176		MOV	DURIV,KEEPIV ;SAVE
1295	002766	016767	176744	176166		MOV	DURIV,BASEIV ;SET UP FOR ROTATION
1296	002774	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1297	002776	015112				MMULT	: MESSAGE
1298	003000	104414				SETFLG	: SET FLAG BASED UPON INPUT STRING
1299	003002	001152				MULTD	: THIS FLAG
1300	003004	105767	176142			TSTB	MULTD ;ARE THERE MULTIPLE DEVICES
1301							: ON THE SYSTEM ?
1302	003010	100406				BMI	BBB ;YES,ASK NEXT QUESTION
1303	003012	005067	176150			CLR	ACTREG
1304	003016	005067	176146			CLR	ROTADD
1305	003022	000167	000140			JMP	OUTMUL ;JUMP AROUND NEXT QUESTION
1306	003026				BBB:		
1307	003026	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1308	003030	015141				MLASTD	: MESSAGE
1309	003032	104410				PARAM	: CONVERT STRING
1310	003034	160000				160000	: LOW LIMIT
1311	003036	167776				167776	: HIGH LIMIT
1312	003040	001160				LASTADD	: STORE AT THIS LOCATION
1313	003042	001			.BYTE	1	: MASK
1314	003043	001			.BYTE	1	: HOW MANY TIMES + 2
1315							: THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1316	003044	012767	000001	176116	1\$:	MOV	#1,ROTADD ;SET UP POINTER
1317	003052	005067	176110			CLR	ACTREG ;CLR ACTIVE REGISTER
1318	003056	056767	176106	176102	2\$:	BIS	ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1319	003064	000241				CLC	
1320	003066	006167	176076			ROL	ROTADD ;SET UP POINTER
1321	003072	103421				BCS	3\$;ARE YOU OUT OF RANGE ?
1322	003074	062767	000010	176052		ADD	#10,BASEADD ;SET UP BASE ADDRESS
1323	003102	026767	176052	176044		CMP	LASTADD,BASEADD ;IS THIS THE LAST DEVICE ?
1324	003110	101362				BHI	2\$;NO DO IT AGAIN
1325	003112	056767	176052	176046		BIS	ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1326							: LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1327							: MULTIPLE DEVICE QUESTION
1328	003120	012767	000001	176042	4\$:	MOV	#1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1329	003126	016767	176024	176020		MOV	KEEPADD,BASEADD ;DITTO
1330	003134	000414				BR	OUTMUL ;CONTINUE QUESTIONS
1331	003136	016767	176014	176010	3\$:	MOV	KEEPADD,BASEADD ;RESTORE
1332	003144	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1333	003146	015235				MRANGE	: MESSAGE
1334	003150	104410				PARAM	: CONVERT STRING
1335	003152	160000				160000	: LOW LIMIT
1336	003154	167776				167776	: HIGH LIMIT
1337	003156	001160				LASTADD	: STORE AT THIS LOCATION
1338	003160	001			.BYTE	1	: MASK
1339	003161	001			.BYTE	1	: HOW MANY TIMES + 2
1340	003162	000167	177656			JMP	1\$;DO IT AGAIN
1341	003166	012767	000340	013442	OUTMUL:	MOV	#340,DUPRT

E03

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 33
 DZDURA.M11 12-OCT-76 10:29

INITIALIZE THE COMMON TAGS

SEQ 0030

```

1342 003174 004767 013366 JSR PC,DULEV
1343 ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1344 ;BUFFER TO THE CHARACTERS "1" AND "2".
1345 ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1346 ;IF THE CHARACTER IS "2" SET THE FLAG
1347 003200 AAA:
1348 003200 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1349 003202 015453 MSYNC ;MESSAGE
1350 003204 122767 000061 012602 3$: CMPB #'1,INBUF ;IS IT "1" ?
1351 003212 001003 BNE 1$
1352 003214 105067 175726 CLRB SYNCNO ;000
1353 003220 000412 BR 4$
1354 003222 122767 000062 012564 1$: CMPB #'2,INBUF ;IS IT "2" ?
1355 003230 001004 BNE 2$
1356 003232 112767 177777 175706 MOVB #-1,SYNCNO ;377
1357 003240 000402 BR 4$
1358 003242 104407 2$: INSTR ;RETRY
1359 003244 000757 BR 3$
1360 003246 000240 4$: NOP
1361 003250 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1362 003252 015521 MWIRE6 ;MESSAGE
1363 003254 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1364 003256 001147 SEXMIT ;THIS FLAG
1365 003260 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1366 003262 015572 MWIRE5 ;MESSAGE
1367 003264 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1368 003266 001150 SEREC ;THIS FLAG
1369 003270 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1370 003272 015642 MWIRE4 ;MESSAGE
1371 003274 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1372 003276 001151 OPTCLR ;THIS FLAG
1373 003300 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1374 003302 015721 MEXTJ ;MESSAGE
1375 003304 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1376 003306 001153 JMRBY ;THIS FLAG
1377
1378 ;TEST START AND RESTART
1379
1380 003310 012706 001100 .BEGIN: MOV #STACK,SP ;SET UP STACK
1381 003314 106427 000340 MTPS #340 ;LOCK OUT INTERRUPTS
1382 003320 032777 000002 176112 BIT #SW01,2SWR ;IF SW01=1, GET STARTING PC
1383 003326 001406 BEQ 3$
1384 003330 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1385 003332 015405 MTSTPC ;MESSAGE
1386 003334 104410 PARAM ;CONVERT STRING
1387 003336 003362 TST1 ;LOW LIMIT
1388 ;HIGH LIMIT
1389 ;STORE AT THIS LOCATION
1390 003340 001 .BYTE 1 ;MASK
1391 003341 001 .BYTE 1 ;HOW MANY TIMES + 2
1392 003342 000403 BR 4$
1393 003344 012767 003362 175532 3$: MOV #TST1,RETURN ;START AT TEST 1
1394 003352 104401 015401 4$: TYPE MR ;TYPE R
1395 003356 000177 175522 JMP @RETURN ;START TESTING
1396
1397 ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE

```

```

1398                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
1399                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1400                                     ;; (OVRUN, RXERR)
1401                                     ;; MODE: ISYMOD
1402                                     ;; LENGTH: FIVE
1403                                     ;; CHAR: 37
1404
1405                                     ;*****
1406 003362 000004                                †ST1: SCOPE
1407 003364 052777 000400 176334                BIS      #MRESET,@TXCSR ; MASTER RESET
1408 003372 012777 000000 176322                MOV      #ISYMOD,@PARCSR ; SET THE MODE
1409 003400 052777 000400 176320                BIS      #MRESET,@TXCSR ; MASTER RESET
1410
1411                                     ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1412 003406 012777 064001 176312                MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1413
1414                                     ; SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
1415 003414 012777 000000 176300                MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
1416 003422 052777 000020 176262                BIS      #SYNSCH,@RXCSR ; SET SYNC SEARCH
1417                                     ; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1418 003430 042777 020000 176270                BIC      #CLK,@TXCSR ; POKE CLK DOWN
1419 003436 052777 020000 176262                BIS      #CLK,@TXCSR ; POKE CLK UP
1420                                     ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1421 003444 042777 020000 176254                BIC      #CLK,@TXCSR ; POKE CLK DOWN
1422 003452 052777 020000 176246                BIS      #CLK,@TXCSR ; POKE CLK UP
1423 003460 016703 176232                MOV      RXDBUF,R3 ; SET UP FOR ERROR MESSAGE
1424 003464 012700 000037                MOV      #37,R0 ; EXPECTED
1425 003470 012767 000007 175424                MOV      #7,SHIFT ; # OF SHIFTS
1426 003476 012767 000176 175774                MOV      #176,$TMP1 ; DATA CHAR
1427 003504 004767 013266                JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1428 003510 105777 176176                TSTB    @RXCSR ; RXDONE
1429 003514 100401                BMI     .+4
1430 003516 104004                ERROR   4 ; RXDONE SHOULD BE SET
1431 003520 017701 176172                MOV      @RXDBUF,R1 ; ACTUAL
1432 003524 020001                CMP      R0,R1 ; COMPARE EXPECTED VS. ACTUAL
1433 003526 001401                BEQ     .+4
1434 003530 104002                ERROR   2 ; RECEIVED DATA DID NOT MATCH
1435                                     ; EXPECTED DATA - CHECK MAINT DATA
1436                                     ; OR RECEIVER LOGIC
1437 003532 012767 000007 175362                MOV      #7,SHIFT ; # OF SHIFTS
1438 003540 012767 000176 175732                MOV      #176,$TMP1 ; DATA CHAR
1439 003546 004767 013224                JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1440                                     ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1441 003552 012767 000007 175342                MOV      #7,SHIFT ; # OF SHIFTS
1442 003560 012767 000176 175712                MOV      #176,$TMP1 ; DATA CHAR
1443 003566 004767 013204                JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1444 003572 012700 140037                MOV      #140000!37,R0 ; EXPECTED DATA PLUS
1445                                     ; RXERR & OVRUN
1446 003576 017701 176114                MOV      @RXDBUF,R1 ; ACTUAL
1447 003602 020001                CMP      R0,R1 ; COMPARE EXP VS. ACT
1448 003604 001401                BEQ     .+4
1449 003606 104002                ERROR   2 ; SPECIFICALLY LOOK AT RXERR &
1450                                     ; OVRUN BITS...THEY BOTH SHOULD BE SET
1451
1452                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1453                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
    
```

```

1454      ; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1455      ; (OVRUN,RXERR)
1456      ; MODE: ISYMOD
1457      ; LENGTH: FIVE
1458      ; CHAR: 0
1459
1460      ;*****
1461      †ST2: SCOPE
1462      BIS      #MRESET,@TXCSR ; MASTER RESET
1463      MOV      #ISYMOD,@PARCSR ; SET THE MODE
1464      BIS      #MRESET,@TXCSR ; MASTER RESET
1465
1466      ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1467      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1468
1469      ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1470      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
1471      BIS      #SYNSCH,@TXCSR ; SET SYNC SEARCH
1472      ; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1473      BIC      #CLK,@TXCSR ; POKE CLK DOWN
1474      BIS      #CLK,@TXCSR ; POKE CLK UP
1475      ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1476      BIC      #CLK,@TXCSR ; POKE CLK DOWN
1477      BIS      #CLK,@TXCSR ; POKE CLK UP
1478      MOV      RXDBUF,R3 ; SET UP FOR ERROR MESSAGE
1479      MOV      #0,R0 ; EXPECTED
1480      MOV      #7,SHIFT ; # OF SHIFTS
1481      MOV      #100,$TMP1 ; DATA CHAR
1482      JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1483      TSTB    @TXCSR ; RXDONE
1484      BMI      .+4
1485      ERROR   4 ; RXDONE SHOULD BE SET
1486      MOV      @RXDBUF,R1 ; ACTUAL
1487      CMP      R0,R1 ; COMPARE EXPECTED VS. ACTUAL
1488      BEQ      .+4
1489      ERROR   2 ; RECEIVED DATA DID NOT MATCH
1490      ; EXPECTED DATA - CHECK MAINT DATA
1491      ; OR RECEIVER LOGIC
1492      MOV      #7,SHIFT ; # OF SHIFTS
1493      MOV      #100,$TMP1 ; DATA CHAR
1494      JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1495      ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1496      MOV      #7,SHIFT ; # OF SHIFTS
1497      MOV      #100,$TMP1 ; DATA CHAR
1498      JSR      PC,RPOKE ; SHIFT IN THIS CHAR
1499      MOV      #140000!0,R0 ; EXPECTED DATA PLUS
1500      ; RXERR & OVRUN
1501      MOV      @RXDBUF,R1 ; ACTUAL
1502      CMP      R0,R1 ; COMPARE EXP VS. ACT
1503      BEQ      .+4
1504      ERROR   2 ; SPECIFICALLY LOOK AT RXERR &
1505      ; OVRUN BITS...THEY BOTH SHOULD BE SET
1506
1507      ; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1508      ; RECEIVER SECTION, IT USES THE ERROR FLAGS
1509      ; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
    
```

H03

```

1510                                     ;; (OVRRUN, RXERR)
1511                                     ;; MODE: ISYMOD
1512                                     ;; LENGTH: SIX
1513                                     ;; CHAR: 25
1514                                     ;;
1515                                     ;*****
1516 004036 000004 †ST3: SCOPE
1517 004040 052777 000400 175660 BIS #MRESET, @TXCSR ; MASTER RESET
1518 004046 012777 000000 175646 MOV #ISYMOD, @PARCSR ; SET THE MODE
1519 004054 052777 000400 175644 BIS #MRESET, @TXCSR ; MASTER RESET
1520
1521 ; SET MAINT DATA, CLK BREAK, & MAINTENANCE MODE
1522 004062 012777 064001 175636 MOV #MTDATA!CLK!MINT!BREAK, @TXCSR
1523
1524 ; SET MODE # OF BITS, PARITY SENSE, & LOAD SYNC REG
1525 004070 012777 002000 175624 MOV #ISYMOD!SIX!NOPAR!0, @PARCSR
1526 004076 052777 000020 175606 BIS #SYNSCH, @RXCSR ; SET SYNC SEARCH
1527 ; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1528 004104 042777 020000 175614 BIC #CLK, @TXCSR ; POKE CLK DOWN
1529 004112 052777 020000 175606 BIS #CLK, @TXCSR ; POKE CLK UP
1530 ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1531 004120 042777 020000 175600 BIC #CLK, @TXCSR ; POKE CLK DOWN
1532 004126 052777 020000 175572 BIS #CLK, @TXCSR ; POKE CLK UP
1533 004134 016703 175556 MOV RXDBUF, R3 ; SET UP FOR ERROR MESSAGE
1534 004140 012700 000025 MOV #25, R0 ; EXPECTED
1535 004144 012767 000010 174750 MOV #8, SHIFT ; # OF SHIFTS
1536 004152 012767 000252 175320 MOV #252, STMP1 ; DATA CHAR
1537 004160 004767 012612 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1538 004164 105777 175522 TSTB @RXCSR ; RXDONE
1539 004170 100401 BMI +4
1540 004172 104004 ERROR 4 ; RXDONE SHOULD BE SET
1541 004174 017701 175516 MOV @RXDBUF, R1 ; ACTUAL
1542 004200 020001 CMP R0, R1 ; COMPARE EXPECTED VS. ACTUAL
1543 004202 001401 BEQ +4
1544 004204 104002 ERROR 2 ; RECEIVED DATA DID NOT MATCH
1545 ; EXPECTED DATA - CHECK MAINT DATA
1546 ; OR RECEIVER LOGIC
1547 004206 012767 000010 174706 MOV #8, SHIFT ; # OF SHIFTS
1548 004214 012767 000252 175256 MOV #252, STMP1 ; DATA CHAR
1549 004222 004767 012550 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1550 ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1551 004226 012767 000010 174666 MOV #8, SHIFT ; # OF SHIFTS
1552 004234 012767 000252 175236 MOV #252, STMP1 ; DATA CHAR
1553 004242 004767 012530 JSR PC, RPOKE ; SHIFT IN THIS CHAR
1554 004246 012700 140025 MOV #140000!25, R0 ; EXPECTED DATA PLUS
1555 ; RXERR & OVRRUN
1556 004252 017701 175440 MOV @RXDBUF, R1 ; ACTUAL
1557 004256 020001 CMP R0, R1 ; COMPARE EXP VS. ACT
1558 004260 001401 BEQ +4
1559 004262 104002 ERROR 2 ; SPECIFICALLY LOOK AT RXERR &
1560 ; OVRRUN BITS...THEY BOTH SHOULD BE SET
1561
1562 ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1563 ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
1564 ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1565 ;; (OVRRUN, RXERR)

```

```

1566          ;;MODE:ISYMOD
1567          ;;LENGTH:SIX
1568          ;;CHAR:52
1569          ;
1570          ;*****
1571 004264 000004          †ST4: SCOPE
1572 004266 052777 000400 175432  BIS      #MRESET,@TXCSR ;MASTER RESET
1573 004274 012777 000000 175420  MOV      #ISYMOD,@PARCSR ;SET THE MODE
1574 004302 052777 000400 175416  BIS      #MRESET,@TXCSR ;MASTER RESET
1575
1576          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1577 004310 012777 064001 175410  MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1578
1579          ;SET MODE, # OF BITS,PARITY SENSE &LOAD SYNC REG
1580 004316 012777 002000 175376  MOV      #ISYMOD!SIX!NOPAR!0,@PARCSR
1581 004324 052777 000020 175360  BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1582          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1583 004332 042777 020000 175366  BIC      #CLK,@TXCSR ;POKE CLK DOWN
1584 004340 052777 020000 175360  BIS      #CLK,@TXCSR ;POKE CLK UP
1585          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1586 004346 042777 020000 175352  BIC      #CLK,@TXCSR ;POKE CLK DOWN
1587 004354 052777 020000 175344  BIS      #CLK,@TXCSR ;POKE CLK UP
1588 004362 016703 175330  MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1589 004366 012700 000052  MOV      #52,R0 ;EXPECTED
1590 004372 012767 000010 174522  MOV      #8,SHIFT ;# OF SHIFTS
1591 004400 012767 000324 175072  MOV      #324,STMP1 ;DATA CHAR
1592 004406 004767 012364  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1593 004412 105777 175274  TSTB    @RXCSR ;RXDONE
1594 004416 100401  BMI      .+4
1595 004420 104004  ERROR   4 ;RXDONE SHOULD BE SET
1596 004422 017701 175270  MOV      @RXDBUF,R1 ;ACTUAL
1597 004426 020001  CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1598 004430 001401  BEQ     .+4
1599 004432 104002  ERROR   2 ;RECEIVED DATA DID NOT MATCH
1600          ;EXPECTED DATA - CHECK MAINT DATA
1601          ;OR RECEIVER LOGIC
1602 004434 012767 000010 174460  MOV      #8,SHIFT ;# OF SHIFTS
1603 004442 012767 000324 175030  MOV      #324,STMP1 ;DATA CHAR
1604 004450 004767 012322  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1605          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1606 004454 012767 000010 174440  MOV      #8,SHIFT ;# OF SHIFTS
1607 004462 012767 000324 175010  MOV      #324,STMP1 ;DATA CHAR
1608 004470 004767 012302  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1609 004474 012700 140052  MOV      #140000!52,R0 ;EXPECTED DATA PLUS
1610          ;RXERR & OVRRUN
1611 004500 017701 175212  MOV      @RXDBUF,R1 ;ACTUAL
1612 004504 020001  CMP      R0,R1 ;COMPARE EXP VS. ACT
1613 004506 001401  BEQ     .+4
1614 004510 104002  ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
1615          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1616
1617          ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1618          ;; RECEIVER SECTION,IT USES THE ERROR FLAGS
1619          ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1620          ;; (OVRRUN,RXERR)
1621          ;;MODE:ISYMOD

```



```

1622                ;;LENGTH:SIX
1623                ;;CHAR:77
1624                ;;
1625                ..*****
1626 004512 000004      TSTS: SCOPE
1627 004514 052777 000400 175204  BIS      #MRESET,@TXCSR ;MASTER RESET
1628 004522 012777 000000 175172  MOV      #ISYMOD,@PARCSR ;SET THE MODE
1629 004530 052777 000400 175170  BIS      #MRESET,@TXCSR ;MASTER RESET
1630
1631                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1632 004536 012777 064001 175162  MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1633
1634                ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
1635 004544 012777 002000 175150  MOV      #ISYMOD!SIX!NOPAR!0,@PARCSR
1636 004552 052777 000020 175132  BIS      #SYNSCH,@TXCSR ;SET SYNC SEARCH
1637                ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1638 004560 042777 020000 175140  BIC      #CLK,@TXCSR ;POKE CLK DOWN
1639 004566 052777 020000 175132  BIS      #CLK,@TXCSR ;POKE CLK UP
1640                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1641 004574 042777 020000 175124  BIC      #CLK,@TXCSR ;POKE CLK DOWN
1642 004602 052777 020000 175116  BIS      #CLK,@TXCSR ;POKE CLK UP
1643 004610 016703 175102  MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1644 004614 012700 000077  MOV      #77,R0 ;EXPECTED
1645 004620 012767 000010 174274  MOV      #8,SHIFT ;# OF SHIFTS
1646 004626 012767 000376 174644  MOV      #376,$TMP1 ;DATA CHAR
1647 004634 004767 012136  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1648 004640 105777 175046  TSTB    @RXCSR ;RXDONE
1649 004644 100401  BMI      .+4
1650 004646 104004  ERROR   4 ;RXDONE SHOULD BE SET
1651 004650 017701 175042  MOV      @RXDBUF,R1 ;ACTUAL
1652 004654 020001  CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1653 004656 001401  BEQ     .+4
1654 004660 104002  ERROR   2 ;RECEIVED DATA DID NOT MATCH
1655                ;EXPECTED DATA - CHECK MAINT DATA
1656                ;OR RECEIVER LOGIC
1657 004662 012767 000010 174232  MOV      #8,SHIFT ;# OF SHIFTS
1658 004670 012767 000376 174602  MOV      #376,$TMP1 ;DATA CHAR
1659 004676 004767 012074  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1660                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1661 004702 012767 000010 174212  MOV      #8,SHIFT ;# OF SHIFTS
1662 004710 012767 000376 174562  MOV      #376,$TMP1 ;DATA CHAR
1663 004716 004767 012054  JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1664 004722 012700 140077  MOV      #140000!77,R0 ;EXPECTED DATA PLUS
1665                ;RXERR & OVRRUN
1666 004726 017701 174764  MOV      @RXDBUF,R1 ;ACTUAL
1667 004732 020001  CMP      R0,R1 ;COMPARE EXP VS. ACT
1668 004734 001401  BEQ     .+4
1669 004736 104002  ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
1670                ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1671
1672                ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1673                ;; RECEIVER SECTION,IT USES THE ERROR FLAGS
1674                ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1675                ;; (OVRRUN,RXERR)
1676                ;; MODE:ISYMOD
1677                ;; LENGTH:SIX
    
```

K03

```

1678          ;;CHAR:0
1679          ;;
1680          ;*****
1681 004740 000004          †ST6: SCOPE
1682 004742 052777 000400 174756      BIS      #MRESET,@TXCSR ;MASTER RESET
1683 004750 012777 000000 174744      MOV      #ISYMOU,@PARCSR ;SET THE MODE
1684 004756 052777 000400 174742      BIS      #MRESET,@TXCSR ;MASTER RESET
1685
1686          ;SET MAINT DATA,CLK BREAK, & MAINTENANCE MODE
1687 004764 012777 064001 174734      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1688
1689          ;SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
1690 004772 012777 002000 174722      MOV      #ISYMOU!SIX!NOPAR!0,@PARCSR
1691 005000 052777 000020 174704      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1692          ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1693 005006 042777 020000 174712      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1694 005014 052777 020000 174704      BIS      #CLK,@TXCSR ;POKE CLK UP
1695          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1696 005022 042777 020000 174676      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1697 005030 052777 020000 174670      BIS      #CLK,@TXCSR ;POKE CLK UP
1698 005036 016703 174654      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1699 005042 012700 000000      MOV      #0,R0 ;EXPECTED
1700 005046 012767 000010 174046      MOV      #8,SHIFT ;# OF SHIFTS
1701 005054 012767 000200 174416      MOV      #200,STMP1 ;DATA CHAR
1702 005062 004767 011710      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1703 005066 105777 174620      TSTB    @RXCSR ;RXDONE
1704 005072 100401      BMI      +4
1705 005074 104004      ERROR   4 ;RXDONE SHOULD BE SET
1706 005076 017701 174614      MOV      @RXDBUF,R1 ;ACTUAL
1707 005102 020001      CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1708 005104 001401      BEQ     +4
1709 005106 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
1710          ;EXPECTED DATA - CHECK MAINT DATA
1711          ;OR RECEIVER LOGIC
1712 005110 012767 000010 174004      MOV      #8,SHIFT ;# OF SHIFTS
1713 005116 012767 000200 174354      MOV      #200,STMP1 ;DATA CHAR
1714 005124 004767 011646      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1715          ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1716 005130 012767 000010 173764      MOV      #8,SHIFT ;# OF SHIFTS
1717 005136 012767 000200 174334      MOV      #200,STMP1 ;DATA CHAR
1718 005144 004767 011626      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1719 005150 012700 140000      MOV      #140000!0,R0 ;EXPECTED DATA PLUS
1720          ;RXERR & OVRRUN
1721 005154 017701 174536      MOV      @RXDBUF,R1 ;ACTUAL
1722 005160 020001      CMP      R0,R1 ;COMPARE EXP VS. ACT
1723 005162 001401      BEQ     +4
1724 005164 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
1725          ;OVRRUN BITS...THEY BOTH SHOULD BE SET
1726
1727          ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1728          ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
1729          ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1730          ;; (OVRRUN, RXERR)
1731          ;; MODE: ISYMOU
1732          ;; LENGTH: SEVEN
1733          ;; CHAR: 125

```

```

1734
1735
1736 005166 000004
1737 005170 052777 000400 174530
1738 005176 012777 000000 174516
1739 005204 052777 000400 174514
1740
1741
1742 005212 012777 064001 174506
1743
1744
1745 005220 012777 004000 174474
1746 005226 052777 000020 174456
1747
1748 005234 042777 020000 174464
1749 005242 052777 020000 174456
1750
1751 005250 042777 020000 174450
1752 005256 052777 020000 174442
1753 005264 016703 174426
1754 005270 012700 000125
1755 005274 012767 000011 173620
1756 005302 012767 000652 174170
1757 005310 004767 011462
1758 005314 105777 174372
1759 005320 100401
1760 005322 104004
1761 005324 017701 174366
1762 005330 020001
1763 005332 001401
1764 005334 104002
1765
1766
1767 005336 012767 000011 173556
1768 005344 012767 000652 174126
1769 005352 004767 011420
1770
1771 005356 012767 000011 173536
1772 005364 012767 000652 174106
1773 005372 004767 011400
1774 005376 012700 140125
1775
1776 005402 017701 174310
1777 005406 020001
1778 005410 001401
1779 005412 104002
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789

;*****
;ST7: SCOPE
; BIS #MRESET,@TXCSR ;MASTER RESET
; MOV #ISYMOD,@PARCSR ;SET THE MODE
; BIS #MRESET,@TXCSR ;MASTER RESET

;SET MAINT DATA,CLK BREAK, & MAINTENANCE MODE
; MOV #MTDATA!CLK!MINT!BREAK,@TXCSR

;SET MODE # OF BITS,PARITY SENSE, & LOAD SYNC REG
; MOV #ISYMOD!SEVEN!NOPAR!D,@PARCSR
; BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
; BIC #CLK,@TXCSR ;POKE CLK DOWN
; BIS #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRIZATION
; BIC #CLK,@TXCSR ;POKE CLK DOWN
; BIS #CLK,@TXCSR ;POKE CLK UP
; MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
; MOV #125,R0 ;EXPECTED
; MOV #9,SHIFT ;# OF SHIFTS
; MOV #652,STMP1 ;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 #9,SHIFT ;# OF SHIFTS
; MOV #652,STMP1 ;DATA CHAR
; JSR PC,RPOKE ;SHIFT IN THIS CHAR
;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
; MOV #9,SHIFT ;# OF SHIFTS
; MOV #652,STMP1 ;DATA CHAR
; JSR PC,RPOKE ;SHIFT IN THIS CHAR
; MOV #140000!125,R0 ;EXPECTED DATA PLUS
; ;RXERR & OVRUN
; MOV @RXDBUF,R1 ;ACTUAL
; CMP R0,R1 ;COMPARE EXP VS. ACT
; BEQ .+4
; ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
; ;OVRUN 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
; ;(OVRUN,RXERR)
; ;MODE:ISYMOD
; ;LENGTH:SEVEN
; ;CHAR:52
; ;

```

```

1790
1791 005414 000004
1792 005416 052777 000400 174302
1793 005424 012777 000000 174270
1794 005432 052777 000400 174266
1795
1796
1797 005440 012777 064001 174260
1798
1799
1800 005446 012777 004000 174246
1801 005454 052777 000020 174230
1802
1803 005462 042777 020000 174236
1804 005470 052777 020000 174230
1805
1806 005476 042777 020000 174222
1807 005504 052777 020000 174214
1808 005512 016703 174200
1809 005516 012700 000052
1810 005522 012767 000011 173372
1811 005530 012767 000524 173742
1812 005536 004767 011234
1813 005542 105777 174144
1814 005546 100401
1815 005550 104004
1816 005552 017701 174140
1817 005556 020001
1818 005560 001401
1819 005562 104002
1820
1821
1822 005564 012767 000011 173330
1823 005572 012767 000524 173700
1824 005600 004767 011172
1825
1826 005604 012767 000011 173310
1827 005612 012767 000524 173660
1828 005620 004767 011152
1829 005624 012700 140052
1830
1831 005630 017701 174062
1832 005634 020001
1833 005636 001401
1834 005640 104002
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845

```

```

*****
TST10: SCOPE
      BIS      #MRESET,@TXCSR ;MASTER RESET
      MOV      #ISYMOD,@PARCSR ;SET THE MODE
      BIS      #MRESET,@TXCSR ;MASTER RESET

;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
      MOV      #MATA:CLK:MINT:BREAK,@TXCSR

;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
      MOV      #ISYMOD:SEVEN:NOPAR:0,@PARCSR
      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
      MOV      R,DBUF,R3 ;SET UP FOR ERROR MESSAGE
      MOV      #52,R0 ;EXPECTED
      MOV      #9,SHIFT ; # OF SHIFTS
      MOV      #524,STMP1 ;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      #9,SHIFT ; # OF SHIFTS
      MOV      #524,STMP1 ;DATA CHAR
      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
      MOV      #9,SHIFT ; # OF SHIFTS
      MOV      #524,STMP1 ;DATA CHAR
      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
      MOV      #140000:52,R0 ;EXPECTED DATA PLUS
                  ;RXERR & OVRRUN
      MOV      @RXDBUF,R1 ;ACTUAL
      CMP      R0,R1 ;COMPARE EXP VS. ACT
      BEQ      .+4
      ERROR    2 ;SPECIFICALLY LOOK AT RXERR &
                  ;OVRRUN 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
; ; (OVRRUN,RXERR)
; ; MODE: ISYMOD
; ; LENGTH: SEVEN
; ; CHAR: 177
*****

```

```

1846 005642 000004 TST11: SCOPE
1847 005644 052777 000400 174054 BIS #MRESET,@TXCSR ;MASTER RESET
1848 005652 012777 000000 174042 MOV #ISYMOD,@PARCSR ;SET THE MODE
1849 005660 052777 000400 174040 BIS #MRESET,@TXCSR ;MASTER RESET
1850
1851 ;SET MAINT DATA,CLK BREAK,&MAINTENANCE MODE
1852 005666 012777 064001 174032 MOV #MTOATA!CLK!MINT!BREAK,@TXCSR
1853
1854 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1855 005674 012777 004000 174020 MOV #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1856 005702 052777 000020 174002 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1857 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1858 005710 042777 020000 174010 BIC #CLK,@TXCSR ;POKE CLK DOWN
1859 005716 052777 020000 174002 BIS #CLK,@TXCSR ;POKE CLK UP
1860 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1861 005724 042777 020000 173774 BIC #CLK,@TXCSR ;POKE CLK DOWN
1862 005732 052777 020000 173766 BIS #CLK,@TXCSR ;POKE CLK UP
1863 005740 016703 173752 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1864 005744 012700 000177 MOV #177,R0 ;EXPECTED
1865 005750 012767 000011 173144 MOV #9,SHIFT ;# OF SHIFTS
1866 005756 012767 000776 173514 MOV #776,$TMP1 ;DATA CHAR
1867 005764 004767 01:006 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1868 005770 105777 173716 TSTB @RXCSR ;RXDONE
1869 005774 100401 BMI .+4
1870 005776 104004 ERROR 4 ;RXDONE SHOULD BE SET
1871 006000 017701 173712 MOV @RXDBUF,R1 ;ACTUAL
1872 006004 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1873 006006 001401 BEQ .+4
1874 006010 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1875 ;EXPECTED DATA - CHECK MAINT DATA
1876 ;OR RECEIVER LOGIC
1877 006012 012767 000011 173102 MOV #9,SHIFT ;# OF SHIFTS
1878 006020 012767 000776 173452 MOV #776,$TMP1 ;DATA CHAR
1879 006026 004767 010744 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1880 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1881 006032 012767 000011 173062 MOV #9,SHIFT ;# OF SHIFTS
1882 006040 012767 000776 173432 MOV #776,$TMP1 ;DATA CHAR
1883 006046 004767 010724 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1884 006052 012700 140177 MOV #140000!177,R0 ;EXPECTED DATA PLUS
1885 ;RXERR & OVRUN
1886 006056 017701 173634 MOV @RXDBUF,R1 ;ACTUAL
1887 006062 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1888 006064 001401 BEQ .+4
1889 006066 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1890 ;OVRUN BITS...THEY BOTH SHOULD BE SET
1891
1892 ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1893 ;; RECEIVER SECTION,IT USES THE ERROR FLAGS
1894 ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1895 ;; (OVRUN,RXERR)
1896 ;; MOVE:ISYMOD
1897 ;; LENGTH:SEVEN
1898 ;; CHAR:0
1899
1900 .....
1901 006070 000004 TST12: SCOPE

```

```

1902 006072 052777 000400 173626      BIS      #MRESET,@TXCSR ;MASTER RESET
1903 006100 012777 000000 173614      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1904 006106 052777 000400 173612      BIS      #MRESET,@TXCSR ;MASTER RESET
1905
1906                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1907 006114 012777 064001 173604      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1908
1909                                     ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
1910 006122 012777 004000 173572      MOV      #ISYMOD!SEVEN!NOPAR!D,@PARCSR
1911 006130 052777 000020 173554      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1912                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1913 006136 042777 020000 173562      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1914 006144 052777 020000 173554      BIS      #CLK,@TXCSR ;POKE CLK UP
1915                                     ;POKE CLK TO GET LOGIC INTO SYNCROIZATION
1916 006152 042777 020000 173546      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1917 006160 052777 020000 173540      BIS      #CLK,@TXCSR ;POKE CLK UP
1918 006166 016703 173524      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1919 006172 012700 000000      MOV      #0,R0 ;EXPECTED
1920 006176 012767 000011 172716      MOV      #9,SHIFT ;# OF SHIFTS
1921 006204 012767 000400 173266      MOV      #400,$TMP1 ;DATA CHAR
1922 006212 004767 010560      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1923 006216 105777 173470      TSTB    @RXCSR ;RXDONE
1924 006222 100401      BMI     .+4
1925 006224 104004      ERROR   4 ;RXDONE SHOULD BE SET
1926 006226 017701 173464      MOV      @RXDBUF,R1 ;ACTUAL
1927 006232 020001      CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1928 006234 001401      BEQ     .+4
1929 006236 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
1930                                     ;EXPECTED DATA - CHECK MAINT DATA
1931                                     ;OR RECEIVER LOGIC
1932 006240 012767 000011 172654      MOV      #9,SHIFT ;# OF SHIFTS
1933 006246 012767 000400 173224      MOV      #400,$TMP1 ;DATA CHAR
1934 006254 004767 010516      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1935                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1936 006260 012767 000011 172634      MOV      #9,SHIFT ;# OF SHIFTS
1937 006266 012767 000400 173204      MOV      #400,$TMP1 ;DATA CHAR
1938 006274 004767 010476      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1939 006300 012700 140000      MOV      #140000!0,R0 ;EXPECTED DATA PLUS
1940                                     ;RXERR & OVRUN
1941 006304 017701 173406      MOV      @RXDBUF,R1 ;ACTUAL
1942 006310 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
1943 006312 001401      BEQ     .+4
1944 006314 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
1945                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
1946
1947                                     ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1948                                     ;;RECEIVER SECTION,IT USES THE ERROR FLAGS
1949                                     ;;TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1950                                     ;;(OVRUN,RXERR)
1951                                     ;;MODE:ISYMOD
1952                                     ;;LENGTH:EIGHT
1953                                     ;;CHAR:125
1954
1955                                     ;*****
1956 006316 000004      †ST13: SCOPE
1957 006320 052777 000400 173400      BIS      #MRESET,@TXCSR ;MASTER RESET

```



```

1958 006326 012777 000000 173366      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1959 006334 052777 000400 173364      BIS      #MRESET,@TXCSR ;MASTER RESET
1960
1961                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1962 006342 012777 06+001 173356      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1963
1964                                     ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
1965 006350 012777 006000 173344      MOV      #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1966 006356 052777 000020 173326      BIS      #SYNSCH,@TXCSR ;SET SYNC SEARCH
1967                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1968 006364 042777 020000 173334      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1969 006372 052777 020000 173326      BIS      #CLK,@TXCSR ;POKE CLK UP
1970                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1971 006400 042777 020000 173320      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1972 006406 052777 020000 173312      BIS      #CLK,@TXCSR ;POKE CLK UP
1973 006414 016703 173276      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1974 006420 012700 000125      MOV      #125,R0 ;EXPECTED
1975 006424 012767 000012 172470      MOV      #10,SHIFT ;# OF SHIFTS
1976 006432 012767 001252 173040      MOV      #1252,STMP1 ;DATA CHAR
1977 006440 004767 010332      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1978 006444 105777 173242      TSTB    @RXCSR ;RXDONE
1979 006450 100401      BMI     .+4
1980 006452 104004      ERROR   4 ;RXDONE SHOULD BE SET
1981 006454 017701 173236      MOV      @RXDBUF,R1 ;ACTUAL
1982 006460 020001      CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1983 006462 001401      BEQ    .+4
1984 006464 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
1985                                     ;EXPECTED DATA - CHECK MAINT DATA
1986                                     ;OR RECEIVER LOGIC
1987 006466 012767 000012 172426      MOV      #10,SHIFT ;# OF SHIFTS
1988 006474 012767 001252 172776      MOV      #1252,STMP1 ;DATA CHAR
1989 006502 004767 010270      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1990                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1991 006506 012767 000012 172406      MOV      #10,SHIFT ;# OF SHIFTS
1992 006514 012767 001252 172756      MOV      #1252,STMP1 ;DATA CHAR
1993 006522 004767 010250      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1994 006526 012700 140125      MOV      #140000!125,R0 ;EXPECTED DATA PLUS
1995                                     ;RXERR & OVRUN
1996 006532 017701 173160      MOV      @RXDBUF,R1 ;ACTUAL
1997 006536 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
1998 006540 001401      BEQ    .+4
1999 006542 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
2000                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
2001
2002                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2003                                     ;; RECEIVER SECTION,IT USES THE ERROR FLAGS
2004                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2005                                     ;; (OVRUN,RXERR)
2006                                     ;; MODE:ISYMOD
2007                                     ;; LENGTH:EIGHT
2008                                     ;; CHAR:252
2009
2010                                     ;*****
2011 006544 000004      †ST14: SCOPE
2012 006546 052777 000400 173152      BIS      #MRESET,@TXCSR ;MASTER RESET
2013 006554 012777 000000 173140      MOV      #ISYMOD,@PARCSR ;SET THE MODE

```

```

2014 006562 052777 000400 173136      BIS      #MRESET,@TXCSR ;MASTER RESET
2015
2016 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2017 006570 012777 064001 173130      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2018
2019 ;SET MODE # OF BITS,PARITY SENSE & LOAD SYNC REG
2020 006576 012777 006000 173116      MOV      #ISYMOD!EIGHT!NOPAR!0,@PARCSR
2021 006604 052777 000020 173100      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2022 ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2023 006612 042777 020000 173106      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2024 006620 052777 020000 173100      BIS      #CLK,@TXCSR ;POKE CLK UP
2025 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2026 006626 042777 020000 173072      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2027 006634 052777 020000 173064      BIS      #CLK,@TXCSR ;POKE CLK UP
2028 006642 016703 173050      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2029 006646 012700 000252      MOV      #252,R0 ;EXPECTED
2030 006652 012767 000012 172242      MOV      #10,SHIFT ;# OF SHIFTS
2031 006660 012767 001524 172612      MOV      #1524,STMP1 ;DATA CHAR
2032 006666 004767 010104      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2033 006672 105777 173014      TSTB    @RXCSR ;RXDONE
2034 006676 100401      BMI     .+4
2035 006700 104004      ERROR   4 ;RXDONE SHOULD BE SET
2036 006702 017701 173010      MOV      @RXDBUF,R1 ;ACTUAL
2037 006706 020001      CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2038 006710 001401      BEQ     .+4
2039 006712 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
2040 ;EXPECTED DATA - CHECK MAINT DATA
2041 ;OR RECEIVER LOGIC
2042 006714 012767 000012 172200      MOV      #10,SHIFT ;# OF SHIFTS
2043 006722 012767 001524 172550      MOV      #1524,STMP1 ;DATA CHAR
2044 006730 004767 010042      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2045 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2046 006734 012767 000012 172160      MOV      #10,SHIFT ;# OF SHIFTS
2047 006742 012767 001524 172530      MOV      #1524,STMP1 ;DATA CHAR
2048 006750 004767 010022      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2049 006754 012700 140252      MOV      #140000!252,R0 ;EXPECTED DATA PLUS
2050 ;RXERR & OVRUN
2051 006760 017701 172732      MOV      @RXDBUF,R1 ;ACTUAL
2052 006764 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
2053 006766 001401      BEQ     .+4
2054 006770 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
2055 ;OVRUN BITS...THEY BOTH SHOULD BE SET
2056
2057 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2058 ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
2059 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2060 ;: (OVRUN,RXERR)
2061 ;: MODE:ISYMOD
2062 ;: LENGTH:EIGHT
2063 ;: CHAR:377
2064
2065 ;:*****
2066 006772 000004      †ST15: SCOPE
2067 006774 052777 000400 172724      BIS      #MRESET,@TXCSR ;MASTER RESET
2068 007002 012777 000000 172712      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2069 007010 052777 000400 172710      BIS      #MRESET,@TXCSR ;MASTER RESET

```

E04

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 46
 DZDURA.M11 12-OCT-76 10:29

INITIALIZE THE COMMON TAGS

SEQ 0043

```

2070
2071 ;SET MAINT DATA,CLK BREAK,&MAINTENANCE MODE
2072 007016 012777 064001 172702      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2073
2074 ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
2075 007024 012777 006000 172670      MOV      #ISYMOD!EIGHT!NOPAR!0,@PARCSR
2076 007032 052777 000020 172652      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2077 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2078 007040 042777 020000 172660      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2079 007046 052777 020000 172652      BIS      #CLK,@TXCSR ;POKE CLK UP
2080 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2081 007054 042777 020000 172644      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2082 007062 052777 020000 172636      BIS      #CLK,@TXCSR ;POKE CLK UP
2083 007070 016703 172622      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2084 007074 012700 000377      MOV      #377,R0 ;EXPECTED
2085 007100 012767 000012 172014      MOV      #10,SHIFT ;# OF SHIFTS
2086 007106 012767 001776 172364      MOV      #1776,$TMP1 ;DATA CHAR
2087 007114 004767 007656      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2088 007120 105777 172566      TSTB    @RXCSR ;RXDONE
2089 007124 100401      BMI     .+4
2090 007126 104004      ERROR   4 ;RXDONE SHOULD BE SET
2091 007130 017701 172562      MOV      @RXDBUF,R1 ;ACTUAL
2092 007134 020001      CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2093 007136 001401      BEQ     .+4
2094 007140 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
2095 ;EXPECTED DATA - CHECK MAINT DATA
2096 ;OR RECEIVER LOGIC
2097 007142 012767 000012 171752      MOV      #10,SHIFT ;# OF SHIFTS
2098 007150 012767 001776 172322      MOV      #1776,$TMP1 ;DATA CHAR
2099 007156 004767 007614      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2100 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2101 007162 012767 000012 171732      MOV      #10,SHIFT ;# OF SHIFTS
2102 007170 012767 001776 172302      MOV      #1776,$TMP1 ;DATA CHAR
2103 007176 004767 007574      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2104 007202 012700 140377      MOV      #140000!377,R0 ;EXPECTED DATA PLUS
2105 ;RXERR & OVRUN
2106 007206 017701 172504      MOV      @RXDBUF,R1 ;ACTUAL
2107 007212 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
2108 007214 001401      BEQ     .+4
2109 007216 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
2110 ;OVRUN BITS...THEY BOTH SHOULD BE SET
2111
2112 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2113 ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
2114 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2115 ;: (OVRUN,RXERR)
2116 ;: MODE:ISYMOD
2117 ;: LENGTH:EIGHT
2118 ;: CHAR:0
2119
2120 ;:*****
2121 007220 000004      †ST16: SCOPE
2122 007222 052777 000400 172476      BIS      #MRESET,@TXCSR ;MASTER RESET
2123 007230 012777 000000 172464      MOV      #ISYMOD,@PARCSR ;SET THE MODE
2124 007236 052777 000400 172462      BIS      #MRESET,@TXCSR ;MASTER RESET
2125

```

```

2126 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2127 007244 012777 064001 172454 MOV #MTDATA:CLK!MINT!BREAK,@TXCSR
2128
2129 ;SET MODE, # OF BITS,PARITY SENSE,&LOAD SYNC REG
2130 007252 012777 006000 172442 MOV #ISYMOD:EIGHT!NOPAR!D,@PARCSR
2131 007260 052777 000020 172424 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2132 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2133 007266 042777 020000 172432 BIC #CLK,@TXCSR ;POKE CLK DOWN
2134 007274 052777 020000 172424 BIS #CLK,@TXCSR ;POKE CLK UP
2135 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2136 007302 042777 020000 172416 BIC #CLK,@TXCSR ;POKE CLK DOWN
2137 007310 052777 020000 172410 BIS #CLK,@TXCSR ;POKE CLK UP
2138 007316 016703 172374 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2139 007322 012700 000000 MOV #D,R0 ;EXPECTED
2140 007326 012767 000012 171566 MOV #10,SHIFT ;# OF SHIFTS
2141 007334 012767 001000 172136 MOV #1000,STMP1 ;DATA CHAR
2142 007342 004767 007430 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2143 007346 105777 172340 TSTB @RXCSR ;RXDONE
2144 007352 100401 BMI .+4
2145 007354 104004 ERROR 4 ;RXDONE SHOULD BE SET
2146 007356 017701 172334 MOV @RXDBUF,R1 ;ACTUAL
2147 007362 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2148 007364 001401 BEQ .+4
2149 007366 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
2150 ;EXPECTED DATA - CHECK MAINT DATA
2151 ;OR RECEIVER LOGIC
2152 007370 012767 000012 171524 MOV #10,SHIFT ;# OF SHIFTS
2153 007376 012767 001000 172074 MOV #1000,STMP1 ;DATA CHAR
2154 007404 004767 007366 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2155 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2156 007410 012767 000012 171504 MOV #10,SHIFT ;# OF SHIFTS
2157 007416 012767 001000 172054 MOV #1000,STMP1 ;DATA CHAR
2158 007424 004767 007346 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2159 007430 012700 140000 MOV #140000!D,R0 ;EXPECTED DATA PLUS
2160 ;RXERR & OVRUN
2161 007434 017701 172256 MOV @RXDBUF,R1 ;ACTUAL
2162 007440 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
2163 007442 001401 BEQ .+4
2164 007444 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
2165 ;OVRUN BITS...THEY BOTH SHOULD BE SET
2166
2167 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2168 ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
2169 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2170 ;: (OVRUN,RXERR)
2171 ;: MODE:SYNEXT
2172 ;: LENGTH:FIVE
2173 ;: CHAR:25
2174
2175 ;:*****
2176 007446 000004 ST17: SCOPE
2177 007450 052777 000400 172250 BIS #MRESET,@TXCSR ;MASTER RESET
2178 007456 012777 020000 172236 MOV #SYNEXT,@PARCSR ;SET THE MODE
2179 007464 052777 000400 172234 BIS #MRESET,@TXCSR ;MASTER RESET
2180
2181 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE

```

```

2182 007472 012777 064001 172226      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2183                                     ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
2184                                     MOV      #SYNEXT!FIVE!NOPAR!0,@PARCSR
2185 007500 012777 020000 172214      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2186 007506 052777 000020 172176      ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2187                                     BIC      #CLK,@TXCSR ;POKE CLK DOWN
2188 007514 042777 020000 172204      BIS      #CLK,@TXCSR ;POKE CLK UP
2189 007522 052777 020000 172176      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2190 007530 016703 172162                                     ;EXPECTED
2191 007534 012700 000025                                     ;# OF SHIFTS
2192 007540 012767 000005 171354      MOV      #5,SHIFT ;# OF SHIFTS
2193 007546 012767 000025 171724      MOV      #25,$TMP1 ;DATA CHAR
2194 007554 004767 007216      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2195 007560 105777 172126      TSTB    @RXCSR ;RXDONE
2196 007564 100401                                     BMI      .+4
2197 007566 104004      ERROR   4 ;RXDONE SHOULD BE SET
2198 007570 017701 172122      MOV      @RXDBUF,R1 ;ACTUAL
2199 007574 020001      CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2200 007576 001401      BEQ     .+4
2201 007600 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
2202                                     ;EXPECTED DATA - CHECK MAINT DATA
2203                                     ;OR RECEIVER LOGIC
2204 007602 012767 000005 171312      MOV      #5,SHIFT ;# OF SHIFTS
2205 007610 012767 000025 171662      MOV      #25,$TMP1 ;DATA CHAR
2206 007616 004767 007154      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2207                                     ;NOW SHIFT IN A SECOND CHARACTER WITH OUT READING RXDBUF
2208 007622 012767 000005 171272      MOV      #5,SHIFT ;# OF SHIFTS
2209 007630 012767 000025 171642      MOV      #25,$TMP1 ;DATA CHAR
2210 007636 004767 007134      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2211 007642 012700 140025      MOV      #140000!25,R0 ;EXPECTED DATA PLUS
2212                                     ;RXERR & OVRUN
2213 007646 017701 172044      MOV      @RXDBUF,R1 ;ACTUAL
2214 007652 020001      CMP      R0,R1 ;COMPARE EXP VS. ACT
2215 007654 001401      BEQ     .+4
2216 007656 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
2217                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
2218
2219                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2220                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
2221                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2222                                     ;; (OVRUN,RXERR)
2223                                     ;; MODE:SYNEXT
2224                                     ;; LENGTH:FIVE
2225                                     ;; CHAR:12
2226                                     ;;
2227                                     ;*****
2228 007660 000004      †ST20: SCOPE
2229 007662 052777 000400 172036      BIS      #MRESET,@TXCSR ;MASTER RESET
2230 007670 012777 020000 172024      MOV      #SYNEXT,@PARCSR ;SET THE MODE
2231 007676 052777 000400 172022      BIS      #MRESET,@TXCSR ;MASTER RESET
2232
2233                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2234 007704 012777 064001 172014      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2235
2236                                     ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
2237 007712 012777 020000 172002      MOV      #SYNEXT!FIVE!NOPAR!0,@PARCSR
    
```

```

2238 007720 052777 000020 171764      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2239                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2240 007726 042777 020000 171772      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2241 007734 052777 020000 171764      BIS      #CLK,@TXCSR ;POKE CLK UP
2242 007742 016703 171750      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2243 007746 012700 000012                                MOV      #12,R0 ;EXPECTED
2244 007752 012767 000005 171142      MOV      #5,SHIFT ;# OF SHIFTS
2245 007760 012767 000012 171512      MOV      #12,$TMP1 ;DATA CHAR
2246 007766 004767 007004      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2247 007772 105777 171714      TSTB    @RXCSR ;RXDONE
2248 007776 104001      BMI     .+4
2249 010000 104004      ERROR   4 ;RXDONE SHOULD BE SET
2250 010002 017701 171710      MOV      @RXDBUF,R1 ;ACTUAL
2251 010006 020001      CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2252 010010 001401      BEQ     .+4
2253 010012 104002      ERROR   2 ;RECEIVED DATA DID NOT MATCH
2254                                ;EXPECTED DATA - CHECK MAINT DATA
2255                                ;OR RECEIVER LOGIC
2256 010014 012767 000005 171100      MOV      #5,SHIFT ;# OF SHIFTS
2257 010022 012767 000012 171450      MOV      #12,$TMP1 ;DATA CHAR
2258 010030 004767 006742      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2259                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2260 010034 012767 000005 171060      MOV      #5,SHIFT ;# OF SHIFTS
2261 010042 012767 000012 171430      MOV      #12,$TMP1 ;DATA CHAR
2262 010050 004767 006722      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2263 010054 012700 140012      MOV      #140000!12,R0 ;EXPECTED DATA PLUS
2264                                ;RXERR & OVRUN
2265 010060 017701 171632      MOV      @RXDBUF,R1 ;ACTUAL
2266 010064 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
2267 010066 001401      BEQ     .+4
2268 010070 104002      ERROR   2 ;SPECIFICALLY LOOK AT RXERR &
2269                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2270
2271                                ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2272                                ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2273                                ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2274                                ;: (OVRUN,RXERR)
2275                                ;: MODE:SYNEXT
2276                                ;: LENGTH:FIVE
2277                                ;: CHAR:37
2278
2279                                ;:*****
2280 010072 000004      †ST21: SCOPE
2281 010074 052777 000400 171624      BIS      #MRESET,@TXCSR ;MASTER RESET
2282 010102 012777 020000 171612      MOV      #SYNEXT,@PARCSR ;SET THE MODE
2283 010110 052777 000400 171610      BIS      #MRESET,@TXCSR ;MASTER RESET
2284
2285                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2286 010116 012777 064001 171602      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2287
2288                                ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
2289 010124 012777 020000 171570      MOV      #SYNEXT!FIVE!NOPAR!0,@PARCSR
2290 010132 052777 000020 171552      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2291                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2292 010140 042777 020000 171560      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2293 010146 052777 020000 171552      BIS      #CLK,@TXCSR ;POKE CLK UP

```

```

2294 010154 016703 171536      MOV     RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
2295 010160 012700 000037      MOV     #37,R0        ;EXPECTED
2296 010164 012767 000005 170730  MOV     #5,SHIFT      ;# OF SHIFTS
2297 010172 012767 000037 171300  MOV     #37,$TMP1     ;DATA CHAR
2298 010200 004767 006572      JSR     PC,RPOKE      ;SHIFT IN THIS CHAR
2299 010204 105777 171502      TSTB   @RXCSR ;RXDONE
2300 010210 100401      BMI    .+4
2301 010212 104004      ERROR  4              ;RXDONE SHOULD BE SET
2302 010214 017701 171476      MOV     @RXDBUF,R1   ;ACTUAL
2303 010220 020001      CMP    R0,R1        ;COMPARE EXPECTED VS. ACTUAL
2304 010222 001401      BEQ    .+4
2305 010224 104002      ERROR  2              ;RECEIVED DATA DID NOT MATCH
2306                                     ;EXPECTED DATA - CHECK MAINT DATA
2307                                     ;OR RECEIVER LOGIC
2308 010226 012767 000005 170666  MOV     #5,SHIFT      ;# OF SHIFTS
2309 010234 012767 000037 171236  MOV     #37,$TMP1     ;DATA CHAR
2310 010242 004767 006530      JSR     PC,RPOKE      ;SHIFT IN THIS CHAR
2311                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2312 010246 012767 000005 170646  MOV     #5,SHIFT      ;# OF SHIFTS
2313 010254 012767 000037 171216  MOV     #37,$TMP1     ;DATA CHAR
2314 010262 004767 006510      JSR     PC,RPOKE      ;SHIFT IN THIS CHAR
2315 010266 012700 140037      MOV     #140000!37,R0 ;EXPECTED DATA PLUS
2316                                     ;RXERR & OVRUN
2317 010272 017701 171420      MOV     @RXDBUF,R1   ;ACTUAL
2318 010276 020001      CMP    R0,R1        ;COMPARE EXP VS. ACT
2319 010300 001401      BEQ    .+4
2320 010302 104002      ERROR  2              ;SPECIFICALLY LOOK AT RXERR &
2321                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
2322
2323                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2324                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
2325                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2326                                     ;; (OVRUN,RXERR)
2327                                     ;; MODE:SYNEXT
2328                                     ;; LENGTH:FIVE
2329                                     ;; CHR:0
2330
2331                                     ;*****
2332 010304 000004      †ST22: SCOP'E
2333 010306 052777 000400 171412  BIS    #MRESET,@TXCSR ;MASTER RESET
2334 010314 012777 020000 171400  MOV    #SYNEXT,@PARCSR ;SET THE MODE
2335 010322 052777 000400 171376  BIS    #MRESET,@TXCSR ;MASTER RESET
2336
2337                                     ;SET MAINT DATA,CLK BREAK, & MAINTENANCE MODE
2338 010330 012777 064001 171370  MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2339
2340                                     ;SET MODE # OF BITS PARITY SENSE & LOAD SYNC REG
2341 010336 012777 020000 171356  MOV    #SYNEXT!FIVE!NOPAR!0,@PARCSR
2342 010344 052777 000020 171340  BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
2343                                     ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2344 010352 042777 020000 171346  BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2345 010360 052777 020000 171340  BIS    #CLK,@TXCSR   ;POKE CLK UP
2346 010366 016703 171324      MOV    RXDBUF,R3      ;SET UP FOR ERROR MESSAGE
2347 010372 012700 000000      MOV    #0,R0         ;EXPECTED
2348 010376 012767 000005 170516  MOV    #5,SHIFT      ;# OF SHIFTS
2349 010404 012767 000000 171066  MOV    #0,$TMP1      ;DATA CHAR

```

```

2350 010412 004767 006360 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2351 010416 105777 171270 TSTB @RXCSR ;RXDONE
2352 010422 100401 BMI .+4
2353 010424 104004 ERROR 4 ;RXDONE SHOULD BE SET
2354 010426 017701 171264 MOV @RXDBUF,R1 ;ACTUAL
2355 010432 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2356 010434 001401 BEQ .+4
2357 010436 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
2358 ;EXPECTED DATA - CHECK MAINT DATA
2359 ;OR RECEIVER LOGIC
2360 010440 012767 000005 170454 MOV #5,SHIFT ;# OF SHIFTS
2361 010446 012767 000000 171024 MOV #0,$TMP1 ;DATA CHAR
2362 010454 004767 006316 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2363 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2364 010460 012767 000005 170434 MOV #5,SHIFT ;# OF SHIFTS
2365 010466 012767 000000 171004 MOV #0,$TMP1 ;DATA CHAR
2366 010474 004767 006276 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2367 010500 012700 140000 MOV #140000!0,R0 ;EXPECTED DATA PLUS
2368 ;RXERR & OVRUN
2369 010504 017701 171206 MOV @RXDBUF,R1 ;ACTUAL
2370 010510 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
2371 010512 001401 BEQ .+4
2372 010514 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
2373 ;OVRUN BITS...THEY BOTH SHOULD BE SET
2374
2375 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2376 ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2377 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2378 ;: (OVRUN, RXERR)
2379 ;: MODE: SYNEXT
2380 ;: LENGTH: SIX
2381 ;: CHAR: 25
2382
2383 ;: *****
2384 010516 000004 ;$T23: SCOPE
2385 010520 052777 000400 171200 BIS #MRESET,@TXCSR ;MASTER RESET
2386 010526 012777 020000 171166 MOV #SYNEXT,@PARCSR ;SET THE MODE
2387 010534 052777 000400 171164 BIS #MRESET,@TXCSR ;MASTER RESET
2388
2389 ;SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2390 010542 012777 064001 171156 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2391
2392 ;SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
2393 010550 012777 022000 171144 MOV #SYNEXT!SIX!NOPAR!0,@PARCSR
2394 010556 052777 000020 171126 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2395
2396 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2396 010564 042777 020000 171134 BIC #CLK,@TXCSR ;POKE CLK DOWN
2397 010572 052777 020000 171126 BIS #CLK,@TXCSR ;POKE CLK UP
2398 010600 016703 171112 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2399 010604 012700 000025 MOV #25,R0 ;EXPECTED
2400 010610 012767 000006 170304 MOV #6,SHIFT ;# OF SHIFTS
2401 010616 012767 000025 170654 MOV #25,$TMP1 ;DATA CHAR
2402 010624 004767 006146 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2403 010630 105777 171056 TSTB @RXCSR ;RXDONE
2404 010634 100401 BMI .+4
2405 010636 104004 ERROR 4 ;RXDONE SHOULD BE SET

```


2406	010640	017701	171052	MOV	@RXDBUF,R1	:ACTUAL
2407	010644	020001		CMP	RO,R1	:COMPARE EXPECTED VS. ACTUAL
2408	010646	001401		BEQ	.+4	
2409	010650	104002		ERROR	2	:RECEIVED DATA DID NOT MATCH
2410						:EXPECTED DATA - CHECK MAINT DATA
2411						:OR RECEIVER LOGIC
2412	010652	012767	000006	MOV	#6,SHIFT	:# OF SHIFTS
2413	010660	012767	000025	MOV	#25,\$TMP1	:DATA CHAR
2414	010666	004767	006104	JSR	PC,@POKE	:SHIFT IN THIS CHAR
2415						:NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2416	010672	012767	000006	MOV	#6,SHIFT	:# OF SHIFTS
2417	010700	012767	000025	MOV	#25,\$TMP1	:DATA CHAR
2418	010706	004767	006064	JSR	PC,@POKE	:SHIFT IN THIS CHAR
2419	010712	012700	140025	MOV	#140000!25,RO	:EXPECTED DATA PLUS
2420						:RXERR & OVRRUN
2421	010716	017701	170774	MOV	@RXDBUF,R1	:ACTUAL
2422	010722	020001		CMP	RO,R1	:COMPARE EXP VS. ACT
2423	010724	001401		BEQ	.+4	
2424	010726	104002		ERROR	2	:SPECIFICALLY LOOK AT RXERR &
2425						:OVRRUN BITS...THEY BOTH SHOULD BE SET
2426						
2427						: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2428						: RECEIVER SECTION,IT USES THE ERROR FLAGS
2429						: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2430						: (OVRRUN,RXERR)
2431						: MODE:SYNEXT
2432						: LENGTH:SIX
2433						: CHAR:52
2434						
2435						:*****
2436	010730	000004		TST	24: SCOPE	
2437	010732	052777	000400	BIS	#MRESET,@TXCSR	:MASTER RESET
2438	010740	012777	020000	MOV	#SYNEXT,@PARCSR	:SET THE MODE
2439	010746	052777	000400	BIS	#MRESET,@TXCSR	:MASTER RESET
2440						:SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2441	010754	012777	064001	MOV	#MTDATA!CLK!MINT!BREAK,@TXCSR	
2442						:SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
2443	010762	012777	022000	MOV	#SYNEXT!SIX!N!PAR!0,@PARCSR	
2444	010770	052777	000020	BIS	#SYNSCH,@RXCSR	:SET SEARCH SYNC
2445						:POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2446	010776	042777	020000	BIC	#CLK,@TXCSR	:POKE CLK DOWN
2447	011004	052777	020000	BIS	#CLK,@TXCSR	:POKE CLK UP
2448	011012	016703	170700	MOV	RXDBUF,R3	:SET UP FOR ERROR MESSAGE
2449	011016	012700	000052	MOV	#52,RO	:EXPECTED
2450	011022	012767	000006	MOV	#6,SHIFT	:# OF SHIFTS
2451	011030	012767	000052	MOV	#52,\$TMP1	:DATA CHAR
2452	011036	004767	005734	JSR	PC,@POKE	:SHIFT IN THIS CHAR
2453	011042	105777	170644	TSTB	@RXCSR	:RXDONE
2454	011046	100401		BMI	.+4	
2455	011050	104004		ERROR	4	:RXDONE SHOULD BE SET
2456	011052	017701	170640	MOV	@RXDBUF,R1	:ACTUAL
2457	011056	020001		CMP	RO,R1	:COMPARE EXPECTED VS. ACTUAL
2458	011060	001401		BEQ	.+4	
2459	011062	104002		ERROR	2	:RECEIVED DATA DID NOT MATCH

```

2462                                     ; EXPECTED DATA - CHECK MAINT DATA
2463                                     ; OR RECEIVER LOGIC
2464 011064 012767 000006 170030      MOV    #6,SHIFT      ; # OF SHIFTS
2465 011072 012767 000052 170400      MOV    #52,$TMP1    ; DATA CHAR
2466 011100 004767 005672              JSR    PC,RPOKE     ; SHIFT IN THIS CHAR
2467                                     ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2468 011104 012767 000006 170010      MOV    #6,SHIFT      ; # OF SHIFTS
2469 011112 012767 000052 170360      MOV    #52,$TMP1    ; DATA CHAR
2470 011120 004767 005652              JSR    PC,RPOKE     ; SHIFT IN THIS CHAR
2471 011124 012700 140052              MOV    #140000!52,R0 ; EXPECTED DATA PLUS
2472                                     ; RXERR & OVRRUN
2473 011130 017701 170562              MOV    @RXDBUF,R1   ; ACTUAL
2474 011134 020001                    CMP    R0,R1        ; COMPARE EXP VS. ACT
2475 011136 001401                    BEQ    +4
2476 011140 104002                    ERROR  2            ; SPECIFICALLY LOOK AT RXERR &
2477                                     ; OVRRUN BITS...THEY BOTH SHOULD BE SET
2478
2479                                     ; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2480                                     ; RECEIVER SECTION, IT USES THE ERROR FLAGS
2481                                     ; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2482                                     ; (OVRRUN,RXERR)
2483                                     ; MODE:SYNEXT
2484                                     ; LENGTH:SIX
2485                                     ; CHAR:77
2486
2487                                     ; *****
2488 011142 000004                    †ST25: SCOPE
2489 011144 052777 000400 170554        BIS    #MRESET,@TXCSR ; MASTER RESET
2490 011152 012777 020000 170542        MOV    #SYNEXT,@PARCSR ; SET THE MODE
2491 011160 052777 000400 170540        BIS    #MRESET,@TXCSR ; MASTER RESET
2492
2493                                     ; SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2494 011166 012777 064001 170532        MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2495
2496                                     ; SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2497 011174 012777 022000 170520        MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2498 011202 052777 000020 170502        BIS    #SYNSCH,@RXCSR ; SET SEARCH SYNC
2499                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2500 011210 042777 020000 170510        BIC    #CLK,@TXCSR  ; POKE CLK DOWN
2501 011216 052777 020000 170502        BIS    #CLK,@TXCSR  ; POKE CLK UP
2502 011224 016703 170466              MOV    RXDBUF,R3    ; SET UP FOR ERROR MESSAGE
2503 011230 012700 000077              MOV    #77,R0      ; EXPECTED
2504 011234 012767 000006 167660      MOV    #6,SHIFT      ; # OF SHIFTS
2505 011242 012767 000077 170230      MOV    #77,$TMP1    ; DATA CHAR
2506 011250 004767 005522              JSR    PC,RPOKE     ; SHIFT IN THIS CHAR
2507 011254 105777 170432              TSTB  @RXCSR ;RXDONE
2508 011260 100401                    BMI    +4
2509 011262 104004                    ERROR  4            ; RXDONE SHOULD BE SET
2510 011264 017701 170426              MOV    @RXDBUF,R1   ; ACTUAL
2511 011270 020001                    CMP    R0,R1        ; COMPARE EXPECTED VS. ACTUAL
2512 011272 001401                    BEQ    +4
2513 011274 104002                    ERROR  2            ; RECEIVED DATA DID NOT MATCH
2514                                     ; EXPECTED DATA - CHECK MAINT DATA
2515                                     ; OR RECEIVER LOGIC
2516 011276 012767 000006 167616      MOV    #6,SHIFT      ; # OF SHIFTS
2517 011304 012767 000077 170166      MOV    #77,$TMP1    ; DATA CHAR

```

```

2518 011312 004767 005460          JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2519                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2520 011316 012767 000006 167576   MOV    #6,SHIFT          ;# OF SHIFTS
2521 011324 012767 000077 170146   MOV    #77,$TMP1         ;DATA CHAR
2522 011332 004767 005440          JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2523 011336 012700 140077          MOV    #140000!77,RO     ;EXPECTED DATA PLUS
2524                                ;RXERR & OVRUN
2525 011342 017701 170350          MOV    @RXDBUF,R1        ;ACTUAL
2526 011346 020001                CMP    RO,R1             ;COMPARE EXP VS. ACT
2527 011350 001401                BEQ    .+4
2528 011352 104002                ERROR  2                 ;SPECIFICALLY LOOK AT RXERR &
2529                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2530
2531                                ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2532                                ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2533                                ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2534                                ;: (OVRUN, RXERR)
2535                                ;: MODE: SYNEXT
2536                                ;: LENGTH: SIX
2537                                ;: CHAR: 0
2538
2539                                ;:*****
2540 011354 000004                tst26: SCOPE
2541 011356 052777 000400 170342   BIS    #MRESET,@TXCSR   ;MASTER RESET
2542 011364 012777 020000 170330   MOV    #SYNEXT,@PARCSR ;SET THE MODE
2543 011372 052777 000400 170326   BIS    #MRESET,@TXCSR   ;MASTER RESET
2544
2545                                ;SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2546 011400 012777 064001 170320   MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2547
2548                                ;SET MODE, # OF BITS, PARITY SENSE & LOAD SYNC REG
2549 011406 012777 022000 170306   MOV    #SYNEXT!SIX!NOPAR!0,@PARCSR
2550 011414 052777 000020 170270   BIS    #SYNSCH,@RXCSR   ;SET SEARCH SYNC
2551                                ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2552 011422 042777 020000 170276   BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2553 011430 052777 020000 170270   BIS    #CLK,@TXCSR      ;POKE CLK UP
2554 011436 016703 170254          MOV    RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2555 011442 012700 000000          MOV    #0,RO            ;EXPECTED
2556 011446 012767 000006 167446   MOV    #6,SHIFT         ;# OF SHIFTS
2557 011454 012767 000000 170016   MOV    #0,$TMP1         ;DATA CHAR
2558 011462 004767 005310          JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2559 011466 105777 170220          TSTB   @RXCSR           ;RXDONE
2560 011472 100401                BMI    .+4
2561 011474 104004                ERROR  4                 ;RXDONE SHOULD BE SET
2562 011476 017701 170214          MOV    @RXDBUF,R1        ;ACTUAL
2563 011502 020001                CMP    RO,R1             ;COMPARE EXPECTED VS. ACTUAL
2564 011504 001401                BEQ    .+4
2565 011506 104002                ERROR  2                 ;RECEIVED DATA DID NOT MATCH
2566                                ;EXPECTED DATA - CHECK MAINT DATA
2567                                ;OR RECEIVER LOGIC
2568 011510 012767 000006 167404   MOV    #6,SHIFT         ;# OF SHIFTS
2569 011516 012767 000000 167754   MOV    #0,$TMP1         ;DATA CHAR
2570 011524 004767 005246          JSR    PC,RPOKE          ;SHIFT IN THIS CHAR
2571                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2572 011530 012767 000006 167364   MOV    #6,SHIFT         ;# OF SHIFTS
2573 011536 012767 000000 167734   MOV    #0,$TMP1         ;DATA CHAR
    
```

```

2574 011544 004767 005226 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2575 011550 012700 140000 MOV #140000!0,RO ;EXPECTED DATA PLUS
2576 ;RXERR & OVRRUN
2577 011554 017701 170136 MOV @RXDBUF,R1 ;ACTUAL
2578 011560 020001 CMP RO,R1 ;COMPARE EXP VS. ACT
2579 011562 001401 BEQ .+4
2580 011564 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
;OVRRUN BITS...THEY BOTH SHOULD BE SET
2581
2582
2583 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2584 ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2585 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2586 ;: (OVRRUN,RXERR)
2587 ;: MODE:SYNEXT
2588 ;: LENGTH:SEVEN
2589 ;: CHAR:125
2590
2591 ;:*****
2592 011566 000004 TST27: SCOPE
2593 011570 052777 000400 170130 BIS #MRESET,@TXCSR ;MASTER RESET
2594 011576 012777 020000 170116 MOV #SYNEXT,@PARCSR ;SET THE MODE
2595 011604 052777 060400 170114 BIS #MRESET,@TXCSR ;MASTER RESET
2596
2597 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2598 011612 012777 064001 170106 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2599
2600 ;SET MODE, # OF BITS,PARITY SENSE,&LOAD SYNC REG
2601 011620 012777 024000 170074 MOV #SYNEXT!SEVEN,@PAR!D@PARCSR
2602 011626 052777 000020 170056 BIS #SYNSCH,@RXCSR ;SET ARCH SYNC
2603 ;POKE CLK TO GET LOG.C INTO SYNCRONIZATION
2604 011634 042777 020000 170064 BIC #CLK,@TXCSR ;POKE CLK DOWN
2605 011642 052777 020000 170056 BIS #CLK,@TXCSR ;POKE CLK UP
2606 011650 016703 170042 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2607 011654 012700 000125 MOV #125,RO ;EXPECTED
2608 011660 012767 000007 167234 MOV #7,SHIFT ;# OF SHIFTS
2609 011666 012767 000125 167604 MOV #125,$TMP1 ;DATA CHAR
2610 011674 004767 005076 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2611 011700 105777 170006 TSTB @RXCSR ;RXDONE
2612 011704 100401 BII .+4
2613 011706 104004 ERROR 4 ;RXDONE SHOULD BE SET
2614 011710 017701 170002 MOV @RXDBUF,R1 ;ACTUAL
2615 011714 020001 CMP RO,R1 ;COMPARE EXPECTED VS. ACTUAL
2616 011716 001401 BEQ .+4
2617 011720 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
;EXPECTED DATA - CHECK MAINT DATA
;OR RECEIVER LOGIC
2618
2619
2620 011722 012767 000007 167172 MOV #7,SHIFT ;# OF SHIFTS
2621 011730 012767 000125 167542 MOV #125,$TMP1 ;DATA CHAR
2622 011736 004767 005034 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2623 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2624 011742 012767 000007 167152 MOV #7,SHIFT ;# OF SHIFTS
2625 011750 012767 000125 167522 MOV #125,$TMP1 ;DATA CHAR
2626 011756 004767 005014 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2627 011762 012700 140125 MOV #140000!125,RO ;EXPECTED DATA PLUS
2628 ;RXERR & OVRRUN
2629 011766 017701 167724 MOV @RXDBUF,R1 ;ACTUAL

```

```

2630 011772 020001      CMP      R0,R1      ;COMPARE EXP VS. ACT
2631 011774 001401      BEQ      .+4
2632 011776 104002      ERROR    2          ;SPECIFICALLY LOOK AT RXERR &
2633                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
2634
2635                                     ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2636                                     ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2637                                     ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2638                                     ;: (OVRUN, RXERR)
2639                                     ;: MODE: SYNEXT
2640                                     ;: LENGTH: SEVEN
2641                                     ;: CHAR: 52
2642
2643                                     ;:*****
2644 012000 000004      †ST30: SCOPE
2645 012002 052777 000400 167716      BIS      #MRESET,@TXCSR ;MASTER RESET
2646 012010 012777 020000 167704      MOV      #SYNEXT,@PARCSR ;SET THE MODE
2647 012016 052777 000400 167702      BIS      #MRESET,@TXCSR ;MASTER RESET
2648
2649                                     ;SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2650 012024 012777 064001 167674      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2651
2652                                     ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2653 012032 012777 024000 167662      MOV      #SYNEXT!SEVEN!NOPAR!0,@PARCSR
2654 012040 052777 000020 167644      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2655
2656                                     ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
2657 012046 042777 020000 167652      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2658 012054 052777 020000 167644      BIS      #CLK,@TXCSR ;POKE CLK UP
2659 012062 016703 167630      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
2660 012072 012767 000007 167022      MOV      #52,R0 ;EXPECTED
2661 012100 012767 000052 167372      MOV      #7,SHIFT ;# OF SHIFTS
2662 012106 004767 004664      JSR      PC,RPOKE ;DATA CHAR
2663 012112 105777 167574      TSTB    @RXCSR ;SHIFT IN THIS CHAR
2664 012116 100401      BMI      .+4
2665 012120 104004      ERROR    4          ;RXDONE SHOULD BE SET
2666 012122 017701 167570      MOV      @RXDBUF,R1 ;ACTUAL
2667 012126 020001      CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
2668 012130 001401      BEQ      .+4
2669 012132 104002      ERROR    2          ;RECEIVED DATA DID NOT MATCH
2670                                     ;EXPECTED DATA - CHECK MAINT DATA
2671                                     ;OR RECEIVER LOGIC
2672 012134 012767 000007 166760      MOV      #7,SHIFT ;# OF SHIFTS
2673 012142 012767 000052 167330      MOV      #52,$TMP1 ;DATA CHAR
2674 012150 004767 004622      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2675
2676                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2677 012154 012767 000007 166740      MOV      #7,SHIFT ;# OF SHIFTS
2678 012162 012767 000052 167310      MOV      #52,$TMP1 ;DATA CHAR
2679 012170 004767 004602      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2680 012174 012700 140052      MOV      #140000!52,R0 ;EXPECTED DATA PLUS
2681                                     ;RXERR & OVRUN
2682 012203 017701 167512      MOV      @RXDBUF,R1 ;ACTUAL
2683 012204 020001      CMP      R0,R1 ;COMPARE EXP VS. ACT
2684 012206 001401      BEQ      .+4
2685 012210 104002      ERROR    2          ;SPECIFICALLY LOOK AT RXERR &
                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET

```

```

2686
2687
2688 ;END OF PASS
2689 ;TYPE NAME OF TEST
2690 ;UPDATE PASS COUNT
2691 ;CHECK FOR EXIT TO ACT-11
2692 ;RESTART TEST
2693
2694 012212 000004 .EOP: SCOPE
2695 012214 004767 000344 JSR PC,CKSWR
2696 012220 104401 TYPE ;TYPE NAME OF TEST
2697 012222 015354 MEPASS
2698 012224 104413 012456 CONVRT ,OUTCRY
2699 012230 104401 015173 TYPE ,DEVICE
2700 012234 105767 166712 TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
2701 012240 001511 BEQ CCC ;NO JUMP AROUND
2702 012242 005767 166720 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
2703 012246 001007 BNE RUNIT ;YES
2704 012250 104401 015205 TYPE MCON ;NO
2705 012254 016700 166706 MOV ACTREG,R0 ;DISPLAY ACTREG
2706 012260 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
2707 ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2708 012262 000167 167664 JMP .START ;START OVER AGAIN..... YOU Deselected EVERYTHING
2709 012266 062767 000010 166660 RUNIT: ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2710 012274 062767 000010 166660 ZERO: ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
2711 012302 000241 CLC
2712 012304 006167 166660 ROL ROTADD ;UP DATE ROTATING POINTER
2713 012310 103410 BCS 2$ ;IS IT THE LAST DEVICE
2714 ;TO BE TESTED IN THIS PASS ?
2715 012312 036767 166652 166646 BIT ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2716 012320 001762 BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
2717 012322 004767 000034 JSR PC,REPLAY ;CALCULATE NEW PARAMETERS
2718 012326 000167 000210 JMP RESTRT ;YES IT WAS ACTIVE, TEST THIS DEVICE
2719 012332 012767 000001 166630 2$: MOV #1,ROTADD ;OK!, NOW SET UP ROTATING
2720 ;POINTER FOR NEXT MULTIPLE PASS
2721 012340 016767 166612 166606 MOV KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2722 012346 016767 166612 166606 MOV KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
2723 012354 004767 000002 JSR PC,REPLAY ;CALC NEW PARAMETERS
2724 012360 000441 BR CCC ;JUMP AROUND REPLAY
2725 012362 016767 166566 004404 REPLAY: MOV BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
2726 012370 004767 004246 JSR PC,DUADR ;CREATE NEW ADDRESSES
2727 012374 016767 166562 167334 MOV BASEIV,DURIV ;CREATE DURIV
2728 012402 062767 000002 166552 ADD #2,BASEIV
2729 012410 016767 166546 167322 MOV BASEIV,DURIS ;CREATE DURIS
2730 012416 062767 000002 166536 ADD #2,BASEIV
2731 012424 016767 166532 167310 MOV BASEIV,DUTIV ;CREATE DUTIV
2732 012432 062767 000002 166522 ADD #2,BASEIV
2733 012440 016767 166516 167276 MOV BASEIV,DUTIS ;CREATE DUTIS
2734 012446 016767 167264 166506 MOV DURIV,BASEIV ;RESTORE
2735 012454 000207 RTS PC
2736
2737 012456 000001 OUTCRY: 1
2738 012460 006 002 .BYTE 6,2
2739 012462 001712 RXCSR
2740
2741 012464 CCC:

```

```

2742 012464 005067 166712 CLR $TSTNM ;CLEAR TEST NUMBER
2743 012470 005067 166722 CLR $ERRPC ;CLEAR LAST ERROR PC
2744 012474 005067 166703 CLR $ERFLG ;CLEAR ERROR FLAG
2745 012500 005267 166406 INC PASCNT ;UPDATE PASS COUNT
2746 012504 016767 166402 166370 MOV PASCNT,LIGHTS ;DISPLAY PASS COUNT
2747 012512 016767 166374 167014 MOV PASCNT,$PASS ;PASS COUNT TO APT
2748 012520 013701 000042 MOV @#42,R1 ;CHECK FOR ACT-11 OR DDP
2749 012524 001406 BEQ RESTRT ;IF NO CONTINUE TESTING
2750 012526 000005 RESET
2751 012530 000005 RESET
2752 012532 004711 SENDAD: JSR PC,(R1)
2753 012534 000240 NOP
2754 012536 000240 NOP
2755 012540 000240 NOP
2756 012542 106427 000340 RESTRT: MTPS #340 ;PREVENT INTERRUPTS (PRIO: 7)
2757 012546 004767 000012 JSR PC,CKSWR
2758 012552 012767 003364 166626 MOV #TST1+2,$LPADR ;SET LAST ADDRESS POINTER
2759 012560 000167 170576 JMP TST1
2760
2761 ;CHECK SWITCH REGISTER ROUTINE.
2762 ;CHECKS TO ALLOW FOR <↑G> TO ALLOW
2763 ;THE CHANGING OF LOCATION 176
2764
2765 012564 005737 000042 CKSWR: TST @#42
2766 012570 001040 BNE OUT
2767 012572 022767 000176 166640 CMP #SWREG,SWR ;SOFTWARE SWR PRESENT?
2768 012600 001034 BNE OUT ;NO--LEAVE
2769 012602 105777 166636 TSTB @STKS ;CHECK TTY READY
2770 012606 100031 BPL OUT ;NO--LEAVE
2771 012610 017767 166632 000422 MOV @STKB,.MSG ;GET CHARACTER
2772 012616 042767 177600 000414 BIC #177600,.MSG ;STRIP JUNK
2773 012624 122767 000007 000406 CMPB #7,.MSG ;IS IT <↑G> ?
2774 012632 001017 BNE OUT ;NO
2775 012634 104401 015761 CNTLU: TYPE MCNTG
2776 012640 005137 012700 COM @#RDSW
2777 012644 104401 015771 TYPE ,MMSWR
2778 012650 104413 CONVRT
2779 012652 012702 SWREGL
2780 012654 104406 016002 INSTR,MMNEW
2781 012660 104410 PARAM
2782 012662 000000 0
2783 012664 177777 177777
2784 012666 000176 SWREG
2785 012670 000 001 .BYTE 0,1
2786 012672 005037 012700 OUT: CLR @#RDSW
2787 012676 000207 RTS PC
2788 012700 000000 RDSW: .WORD 0
2789 012702 000001 SWREGL: 1
2790 012704 006 002 .BYTE 6,2
2791 012706 000176 SWREG
2792
2793 012710 000005 5
2794
2795 ;CHECK FOR FREEZE ON CURRENT DATA
2796
2797 012712 004767 177646 .SCOP1: JSR PC,CKSWR

```

```

2798 012716 032777 001000 166514 BIT #SW09,JSWR
2799 012724 001402 BEQ 1$
2800 012726 016716 166156 MOV LOCK,(SP)
2801 012732 000002 1$: RTI
2802 .SBTTL TYPE ROUTINE
2803
2804 ;*****
2805 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2806 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2807 ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2808 ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2809 ;NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2810 ;
2811 ;CALL:
2812 ;1) USING A TRAP INSTRUCTION
2813 ; TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2814 ;OR
2815 ; TYPE
2816 ; MESADR
2817 ;
2818
2819 012734 105767 166517 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
2820 012740 100002 BPL 1$ ;; BR IF YES
2821 012742 000000 HALT ;; HALT HERE IF NO TERMINAL
2822 012744 000430 BR 3$ ;; LEAVE
2823 012746 010046 1$: MOV RO, -(SP) ;; SAVE RO
2824 012750 017600 000002 MOV #2(SP),RO ;; GET ADDRESS OF ASCIZ STRING
2825 012754 122767 000001 166564 CMPB #APTENV,$ENV ;; RUNNING IN APT MODE
2826 012762 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
2827 012764 132767 000100 166555 BITB #APTPOOL,$ENVM ;; SPOOL MESSAGE TO APT
2828 012772 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
2829 012774 010067 000004 MOV RO,61$ ;; SETUP MESSAGE ADDRESS FOR APT
2830 013000 004767 165004 JSR PC,$ATY3 ;; SPOOL MESSAGE TO APT
2831 013004 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
2832 013006 132767 000040 166533 62$: BITB #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED
2833 013014 001003 BNE 60$ ;; YES, SKIP TYPE OUT
2834 013016 112046 2$: MOVB (RO)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2835 013020 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
2836 013022 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
2837 013024 012600 60$: MOV (SP)+,RO ;; RESTORE RO
2838 013026 062716 000002 3$: ADD #2,(SP) ;; ADJUST RETURN PC
2839 013032 000002 RTI ;; RETURN
2840 013034 122716 000011 4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
2841 013040 001430 BEQ 8$
2842 013042 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
2843 013046 001006 BNE 5$
2844 013050 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
2845 013052 104401 TYPE ;; TYPE A CR AND LF
2846 013054 001523 $CRLF
2847 013056 105067 000130 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
2848 013062 000755 BR 2$ ;; GET NEXT CHARACTER
2849 013064 004767 000056 5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
2850 013070 126726 166362 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
2851 013074 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
2852 013076 016746 166352 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
2853 ;; AND THE NULL CHAR.

```



```

2854 013102 105366 000001 7S:  DECB 1(SP)      ; DOES A NULL NEED TO BE TYPED?
2855 013106 002770          BLT  6S      ; BR IF NO--GO POP THE NULL OFF OF STACK
2856 013110 004767 000032  JSR  PC,$TYPEC ; GO TYPE A NULL
2857 013114 105367 000072  DECB $CHARCNT ; DO NOT COUNT AS A COUNT
2858 013120 000770          BR   7S      ; LOOP
2859
2860 ;HORIZONTAL TAB PROCESSOR
2861
2862 013122 112716 000040 8S:  MOVB #' (SP)      ; REPLACE TAB WITH SPACE
2863 013126 004767 000014 9S:  JSR  PC,$TYPEC ; TYPE A SPACE
2864 013132 132767 000007 000052 BITB #7,$CHARCNT ; BRANCH IF NOT AT
2865 013140 001372          BNE  9S      ; TAB STOP
2866 013142 005726          TST  (SP)+   ; POP SPACE OFF STACK
2867 013144 000724          BR   2S      ; GET NEXT CHARACTER
2868 013146 105777 166276 $TYPEC: TSTB $STPS ; WAIT UNTIL PRINTER IS READY
2869 013152 100375          BPL  $TYPEC
2870 013154 116677 000002 166270 MOVB 2(SP),2$TPB ; LOAD CHAR TO BE TYPED INTO DATA REG.
2871 013162 122766 000015 000002 CMPB #CR,2(SP) ; IS CHARACTER A CARRIAGE RETURN?
2872 013170 001003          BNE  1S      ; BRANCH IF NO
2873 013172 105067 000014          CLRB $CHARCNT ; YES--CLEAR CHARACTER COUNT
2874 013176 000406          BR   $TYPEX ; EXIT
2875 013200 122766 000012 000002 1S:  CMPB #LF,2(SP) ; IS CHARACTER A LINE FEED?
2876 013206 001402          BEQ  $TYPEX ; BRANCH IF YES
2877 013210 105227          INCB (PC)+   ; COUNT THE CHARACTER
2878 013212 000000 $CHARCNT: WORD 0 ; CHARACTER COUNT STORAGE
2879 013214 000207 $TYPEX: RTS  PC
2880
2881 ;ASCII STRING INPUT ROUTINE
2882
2883
2884 013216 017667 000000 000014 .INSTR: MOV 2(SP),MSG ; PICK UP MESSAGE
2885 013224 062716 000002          ADD  #2,(SP) ; JUMP AROUND MESSAGE FOR RTI
2886 013230 105767 166312          TSTB $ENV ; APT CONTROL
2887 013234 001036          BNE  INSTR2 ; YES NO TYPE
2888 013236 104401          .INST1: TYPE
2889 013240 000000          .MSG: 0
2890 013242 012704 016014          MOV  #INBUF,R4 ; GET STARTING LOC OF INBUF
2891 013246 012703 000007          MOV  #7,R3 ; MAX # OF CHARS
2892 013252 105777 166166 1S:  TSTB $STKS ; TTY FLAG
2893 013256 100375          APL  1S
2894 013260 117714 166162          MOVB $STKB,(R4) ; TAKE CHAR
2895 013264 142714 000200          BICB #200,(R4) ; STRIP
2896 013270 121427 000025          CMPB (R4),#25 ; IS IT <+G>
2897 013274 001760          BEQ  .INST1
2898 013276 122427 000015          CMPB (R4)+,#15 ; CHECK FOR CR
2899 013302 001413          BEQ  INSTR2
2900 013304 105777 166140 2S:  TSTB $STPS ; TEST FLAG
2901 013310 100375          BPL  2S
2902 013312 117777 166130 166132          MOVB $STKB,$STPB ; ECHO CHARACTER
2903 013320 005303          DEC  R3 ; DID YOU TYPE TOO MANY CHARS ?
2904 013322 001760          BNE  1S
2905 013324 104401          .INSTE: TYPE
2906 013326 015301          MCM  ; ?
2907 013330 000742          BR   .INST1 ; RETRY
2908 013332 000002 INSTR2: RTI
2909

```

```

2910                                     ;CONVERT ASCII STRING TO OCTAL
2911
2912 013334 011605 .PARAM: MOV (SP),R5 ;PUT CONTENTS OF SP INTO R5
2913 013336 012567 000162 MOV (R5)+,LOLIM ;PUT LOW LIMIT INTO LOLIM
2914 013342 012567 000160 MOV (R5)+,HILIM ;PUT HIGH LIMIT INTO HILIM
2915 013346 012567 000156 MOV (R5)+,DEVADR ;PUT STORE LOC INTO DEVADR
2916 013352 112567 000154 MOV (R5)+,LOBITS ;PUT MASK INTO LOBITS
2917 013356 112567 000151 MOV (R5)+,ADRCNT ;PUT COUNT INTO ADRCNT
2918 013362 010516 MOV R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2919 013364 005005 PARAM1: CLR R5
2920 013366 012704 016014 MOV #INBUF,R4
2921 013372 122714 000015 CMPB #15,(R4) ;CR ?
2922 013376 001420 BEQ PARERR ;YOU TYPED CR TOO SOON !
2923 013400 121427 000060 1$: CMPB (R4),#60 ;LOW LIMIT ASCII 0
2924 013404 002415 BLT PARERR
2925 013406 121427 000067 CMPB (R4),#67 ;HIGH LIMIT ASCII 7
2926 013412 003012 BGT PARERR
2927 013414 142714 BICB #60,(R4) ;CONVERT TO OCTAL
2928 013420 152405 BISB (R4)+,R5 ;STORE AWAY ITS AN OK CHAR
2929 013422 122714 000015 CMPB #15,(R4) ;CR ?
2930 013426 001414 BEQ LIMITS ;NOW CHECK FOR HIGH & LOW LIMIT CONDS
2931 013430 006305 ASL R5 ;ALLOCATE ROOM FOR NEXT CHAR
2932 013432 006305 ASL R5
2933 013434 006305 ASL R5
2934 013436 000760 BR 1$
2935 013440 122714 000015 PARERR: CMPB #15,(R4) ;CR?
2936 013444 001003 BNE 120$
2937 013446 005737 012700 TST #ARDSW ;CK SWR USED
2938 013452 001023 BNE PARTI
2939 013454 104407 120$: INSTER ;RETRY
2940 013456 000742 BR PARAM1
2941
2942                                     ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2943
2944 013460 020567 000042 LIMITS: CMP R5,HILIM
2945 013464 101365 BHI PARERR ;THE # IS TOO HIGH
2946 013466 020567 000032 CMP R5,LOLIM
2947 013472 103762 BLO PARERR ;THE # IS TOO LOW
2948 013474 136705 000032 BITB LOBITS,R5 ;TEST BY MASKINGTHE #
2949 013500 001357 BNE PARERR
2950
2951                                     ;STORE NUMBER AT SPECIFIED ADDRESS
2952
2953 013502 016704 000022 1$: MOV DEVADR,R4 ;GET STARTING ADDR OF
2954 013506 010524 000002 MOV R5,(R4)+ ;STORE AT THIS ADDR
2955 013510 062705 000002 ADD #2,R5
2956 013514 105367 000013 DECB ADRCNT ;HOW MANY TIMES + 2 ?
2957 013520 001372 BNE 1$
2958 013522 000002 PARTI: RTI
2959 013524 000000 LOLIM: 0
2960 013526 000000 HILIM: 0
2961 013530 000000 DEVADR: 0
2962 013532 000000 LOBITS: 0
2963 013533 013533 ADRCNT=LOBITS+1
2964
2965                                     ;SAVE PC OF TEST THAT FAILED AND RO-R5

```

H05

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 62
 DZDURA.M11 12-OCT-76 10:29 TYPE ROUTINE

SEQ 0059

```

2966
2967 013534 016667 000004 165364 .SAV05: MOV 4(SP), SAVPC
2968
2969 ;SAVE R0-R5
2970
2971 013542 010567 165726 SV05: MOV R5, $REG5
2972 013546 010467 165720 MOV R4, $REG4
2973 013552 010367 165712 MOV R3, $REG3
2974 013556 010267 165704 MOV R2, $REG2
2975 013562 010167 165676 MOV R1, $REG1
2976 013566 010067 165670 MOV R0, $REG0
2977 013572 000002 RTI
2978
2979 ;RESTORE R0-R5
2980
2981 013574 016700 165662 .RES05: MOV $REG0, R0
2982 013600 016701 165660 MOV $REG1, R1
2983 013604 016702 165656 MOV $REG2, R2
2984 013610 016703 165654 MOV $REG3, R3
2985 013614 016704 165652 MOV $REG4, R4
2986 013620 016705 165650 MOV $REG5, R5
2987 013624 000002 RTI
2988
2989 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2990
2991 013626 104401 .CONVR: TYPE
2992 013630 015305 MCRLF ;CR LF
2993 013632 017601 MOV 2(SP), R1 ;PICK UP DATA POINTER
2994 013636 062716 ADD #2(SP), WPCNT ;SET UP SP FOR RTI
2995 013642 012167 MOV (R1)+, WPCNT ;PICK UP # OF WORDS FROM TABLE
2996 013646 112167 1$: MOV (R1)+, CHRCNT ;PICK UP # OF CHARS FROM TABLE
2997 013652 112167 000123 MOV (R1)+, SPACNT ;PICK UP # OF SPACES FROM TABLE
2998 013656 013167 000120 MOV 2(R1)+, BINWRD ;PICK UP ADDRESS OF MSG
2999 ;FROM TABLE
3000 013662 016704 000114 2$: MOV BINWRD, R4 ;SAVE
3001 013666 116705 000106 MOV (R4), R5 ;SAVE
3002 013672 012700 016056 MOV #TEMP, R0 ;STARTING ADDRESS OF TEMP BLOCK
3003 013676 010403 3$: MOV R4, R3 ;SAVE
3004 013700 042703 177770 BIC #177770, R3 ;CLR OUT UPPER BITS .. SAVE CHAR
3005 013704 062703 000260 ADD #260, R3 ;CONVERT TO ASCII
3006 013710 110320 MOV R3, (R0)+ ;STORE AWAY
3007 013712 006204 ASR R4 ;SHIFT FOR NEXT #
3008 013714 006204 ASR R4 ;DITTO
3009 013716 006204 ASR R4 ;DITTO
3010 013720 005305 DEC R5 ;DEC CHAR COUNT
3011 013722 001365 BNE 3$ ;DO IT AGAIN ?
3012 013724 012703 016120 MOV #MDATA, R3 ;STARTING ADDRESS OF MDATA BLOCK
3013 013730 114023 4$: MOV (R0), (R3)+ ;REVERSE THE ORDER OF NUMBERS
3014 013732 105367 000042 DECB CHRCNT ;DEC CHAR COUNT
3015 013736 001374 BNE 4$ ;DO IT AGAIN ?
3016 013740 105767 000035 TSTB SPACNT ;HOW MANY SPACES ?
3017 013744 001405 BEQ 6$ ;TYPE # IF BR =0
3018 013746 112723 000240 5$: MOV (R3)+, #240, (R3)+ ;"SPACE" IN ASCII
3019 013752 105367 000023 DECB SPACNT ;DEC # OF SPACE COUNT
3020 013756 001373 BNE 5$ ;DO IT AGAIN ?
3021 013760 105013 6$: CLRB (R3) ;INSERT "0" FOR TTY OUTPUT ROUTINE

```

```

3022 013762 104401          TYPE
3023 013764 016120          MDATA ; THIS MESSAGE
3024 013766 005367 000004  DEC WRDCNT ; HOW MANY #'S ?
3025 013772 001325          BNE 1$ ; DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3026 013774 000002          RTI ; RETURN TO PROGRAM
3027 013776 000000          WRDCNT: 0
3028 014000 000000          CHRCNT: 0
3029 014001 014001          SPACNT=CHRCNT+1
3030 014002 000000          BINWRD: 0
3031
3032
3033 ; COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3034 ; BUFFER TO THE CHARACTERS "N" AND "Y".
3035 ; IF THE CHARACTER IS "N" CLEAR THE FLAG
3036 ; IF THE CHARACTER IS "Y" SET THE FLAG
3037 014004 017605 000000 .SETFLG:MOV @ (SP),R5
3038 014010 122767 000116 001776  CMPB #'N,INBUF ; IS IT "N" ?
3039 014016 001002          BNE 1$
3040 014020 105015          CLRB (R5) ; 000
3041 014022 000406          BR 2$
3042 014024 122767 000131 001762 1$: CMPB #'Y,INBUF ; IS IT "Y" ?
3043 014032 001005          BNE 3$
3044 014034 112715 177777  MOVB #-1 (R5) ; 377
3045 014040 062716 000002 2$: ADD #2, (SP)
3046 014044 000002          RTI
3047 014046 104407          3$: INSTER ; RETRY
3048 014050 000755          BR .SETFLG
3049 .SBTTL ERROR HANDLER ROUTINE
3050
3051 ; *****
3052 ; THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3053 ; SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3054 ; AND GO TO SAVIT ON ERROR
3055 ; THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3056 ; SW15=1 HALT ON ERROR
3057 ; SW13=1 INHIBIT ERROR TYPEOUTS
3058 ; SW10=1 BELL ON ERROR
3059 ; SW09=1 LOOP ON ERROR
3060 ; CALL
3061 ; * ERROR N ;; ERROR=EMT AND N=ERROR ITEM NUMBER
3062
3063 $ERROR:
3064 014052 105267 165325 7$: INCB $ERFLG ; SET THE ERROR FLAG
3065 014056 001775          BEQ 7$ ; DON'T LET THE FLAG GO TO ZERO
3066 014060 016777 165316 165354  MOV $STNM, @DISPLAY ; DISPLAY TEST NUMBER AND ERROR FLAG
3067 014066 032777 002000 165344  BIT #BIT10, @SWR ; BELL ON ERROR?
3068 014074 001402          BEQ 1$ ; NO - SKIP
3069 014076 104401 001516  TYPE $BELL ; RING BELL
3070 014102 005267 165304 1$: INC $ERTTL ; COUNT THE NUMBER OF ERRORS
3071 014106 011667 165304  MOV (SP), $ERRPC ; GET ADDRESS OF ERROR INSTRUCTION
3072 014112 162767 000002 165276  SUB #2, $ERRPC
3073 014120 117767 165272 165266  MOVB @ $ERRPC, $ITEMS ; STRIP AND SAVE THE ERROR ITEM CODE
3074 014126 032777 020000 165304  BIT #BIT13, @SWR ; SKIP TYPEOUT IF SET
3075 014134 001004          BNE 20$ ; SKIP TYPEOUTS
3076 014136 004767 000072  JSR PC, SAVIT ; GO TO USER ERROR ROUTINE
3077 014142 104401 001523  TYPE , $CRLF
    
```

```

3078 014146          20$:
3079 014146 122767 000001 165372  CMPB  #APTENV,SENV  ;;RUNNING IN APT MODE
3080 014154 001007          BNE  2$             ;;NO SKIP APT ERROR REPORT
3081 014156 116767 165232 000004  MOVB  $ITEMB,21$    ;;SET ITEM NUMBER AS ERROR NUMBER
3082 014164 004767 163630          JSR  PC,$ATY4       ;;REPORT FATAL ERROR TO APT
3083 014170          21$:  .BYTE  0
3084 014171          .BYTE  0
3085 014172 000777          BR   22$           ;; APT ERROR LOOP
3086 014174 005777 165240          22$:  TST  $SWR      ;; HALT ON ERROR
3087 014200 100001          BPL  3$           ;; SKIP IF CONTINUE
3088 014202 000000          HALT              ;; HALT ON ERROR!
3089 014204 032777 001000 165226  3$:  BIT  %BIT09,$SWR  ;; LOOP ON ERROR SWITCH SET?
3090 014212 001402          BEQ  4$           ;; BR IF NO
3091 014214 016716 165170          MOV  $LPERA,(SP)  ;; FUDGE RETURN FOR LOOPING
3092 014220 005767 165270          4$:  TST  $ESCAPE    ;; CHECK FOR AN ESCAPE ADDRESS
3093 014224 001402          BEQ  5$           ;; BR IF NONE
3094 014226 016716 165262          MOV  $ESCAPE,(SP) ;; FUDGE RETURN ADDRESS FOR ESCAPE
3095 014232          5$:
3096 014232 000002          RTI              ;; RETURN
3097 014234 010067 164670  SAVIT: MOV  R0,HLD0
3098 014240 010167 164666          MOV  R1,HLD1
3099 014244 010267 164664          MOV  R2,HLD2
3100 014250 010367 164662          MOV  R3,HLD3
3101 014254 010467 164660          MOV  R4,HLD4
3102 014260 010567 164656          MOV  R5,HLD5
3103 014264 016767 165112 164652  MOV  $TSTNM,HLD6

.SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

*****
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
*****
3112 014272          $ERRTYP:
3113 014272 104401 001523          TYPE  $CRLF        ;; "CARRIAGE RETURN" & "LINE FEED"
3114 014276 010046          MOV  R0,-(SP)     ;; SAVE R0
3115 014300 005000          CLR  R0           ;; PICKUP THE ITEM INDEX
3116 014302 153700 001414          BISB  2*$ITEMB,R0
3117 014306 001004          BNE  1$           ;; IF ITEM NUMBER IS ZERO, JUST
3118          MOV  $ERRPC,-(SP)  ;; TYPE THE PC OF THE ERROR
3119 014310 016746 165102          ;; SAVE $ERRPC FOR TYPEOUT
3120          ;; ERROR ADDRESS
3121 014314 104402          TYPOC            ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
3122 014316 000426          BR   6$           ;; GET OUT
3123 014320 005300          1$:  DEC  R0           ;; ADJUST THE INDEX SO THAT IT WILL
3124 014322 006300          ASL  R0           ;; WORK FOR THE ERROR TABLE
3125 014324 006300          ASL  R0
3126 014326 006300          ASL  R0
3127 014330 062700 001652          ADD  #$ERRTB,R0  ;; FORM TABLE POINTER
3128 014334 012067 000004          MOV  (R0)+,2$    ;; PICKUP "ERROR MESSAGE" POINTER
3129 014340 001404          BEQ  3$           ;; SKIP TYPEOUT IF NO POINTER
3130 014342 104401          TYPE  "ERROR MESSAGE"
3131 014344 000000          2$:  .WORD  0        ;; "ERROR MESSAGE" POINTER GOES HERE
3132 014346 104401 001523          TYPE  $CRLF      ;; "CARRIAGE RETURN" & "LINE FEED"
3133 014352 012067 000004          3$:  MOV  (R0)+,4$    ;; PICKUP "DATA HEADER" POINTER
    
```

```

3134 014356 001404      BEQ      5$      ;; SKIP TYPEOUT IF 0
3135 014360 104401      TYPE      ;; TYPE THE "DATA HEADER"
3136 014362 000000      4$: .WORD 0      ;; "DATA HEADER" POINTER GOES HERE
3137 014364 104401 001523  TYPE      $CRLF  ;; "CARRIAGE RETURN" & "LINE FEED"
3138 014370 011000      5$: MOV      (R0),R0  ;; PICKUP "DATA TABLE" POINTER
3139 014372 001004      BNE      7$      ;; GO TYPE THE DATA
3140 014374 012600      6$: MOV      (SP)+,R0  ;; RESTORE R0
3141 014376 104401 001523  TYPE      $CRLF  ;; "CARRIAGE RETURN" & "LINE FEED"
3142 014402 000207      RTS      PC      ;; RETURN
3143 014404
3144 014404 013046      7$: MOV      2(R0)+,-(SP)  ;; SAVE 2(R0)+ FOR TYPEOUT
3145 014406 104402      TYPOC     ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
3146 014410 005710      TST      (R0)    ;; IS THERE ANOTHER NUMBER?
3147 014412 001770      BEQ      6$      ;; BR IF NO
3148 014414 104401 014422  TYPE      8$      ;; TYPE TWO(2) SPACES
3149 014420 000771      BR       7$      ;; LOOP
3150 014422 020040 000      8$: .ASCIZ  / /    ;; TWO(2) SPACES
3151      014426      .EVEN
3152      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
3153
3154      ;*****
3155      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3156      ;OCTAL (ASCII) NUMBER AND TYPE IT.
3157      ;STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3158      ;CALL:
3159      ;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
3160      ;      TYPOS     ;; CALL FOR TYPEOUT
3161      ;      .BYTE   N      ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3162      ;      .BYTE   M      ;; M=1 OR 0
3163      ;      ;; 1=TYPE LEADING ZEROS
3164      ;      ;; 0=SUPPRESS LEADING ZEROS
3165      ;
3166      ;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3167      ;$TYPOS OR $TYPOC
3168      ;CALL:
3169      ;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
3170      ;      TYPON    ;; CALL FOR TYPEOUT
3171      ;
3172      ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3173      ;CALL:
3174      ;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
3175      ;      TYPOC    ;; CALL FOR TYPEOUT
3176      ;
3177 014426 017646 000000 000000 000000 $TYPOS: MOV      2(SP),-(SP)  ;; PICKUP THE MODE
3178 014432 116667 000001 000211  MOVB     1(SP),SOFILL  ;; LOAD ZERO FILL SWITCH
3179 014440 112667 000207  MOVB     (SP)+,SOMODE+1  ;; NUMBER OF DIGITS TO TYPE
3180 014444 062716 000002  ADD      #2,(SP)      ;; ADJUST RETURN ADDRESS
3181 014450 000406  BR       $TYPON
3182 014452 112767 000001 000171 $TYPOC: MOVB     #1,SOFILL  ;; SET THE ZERO FILL SWITCH
3183 014460 112767 000006 000165  MOVB     #6,SOMODE+1  ;; SET FOR SIX(6) DIGITS
3184 014466 112767 000005 000154 $TYPON: MOVB     #5,SXCNT  ;; SET THE ITERATION COUNT
3185 014474 010346  MOV      R3,-(SP)    ;; SAVE R3
3186 014476 010446  MOV      R4,-(SP)    ;; SAVE R4
3187 014500 010546  MOV      R5,-(SP)    ;; SAVE R5
3188 014502 116704 000145  MOVB     SOMODE+1,R4  ;; GET THE NUMBER OF DIGITS TO TYPE
3189 014506 005404  NEG      R4
    
```

L05

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 66
 DZDURA.M11 12-OCT-76 10:29 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0063

3190	014510	062704	000006		ADD	#6,R4	:: SUBTRACT IT FOR MAX. ALLOWED
3191	014514	110467	000132		MOV8	R4,\$OMODE	:: SAVE IT FOR USE
3192	014520	116704	000125		MOV8	\$OFILL,R4	:: GET THE ZERO FILL SWITCH
3193	014524	016605	000012		MOV	12(SP),R5	:: PICKUP THE INPUT NUMBER
3194	014530	005003			CLR	R3	:: CLEAR THE OUTPUT WORD
3195	014532	006105		15:	ROL	R5	:: ROTATE MSB INTO "C"
3196	014534	000404			BR	3\$:: GO DO MSB
3197	014536	006105		25:	ROL	R5	:: FORM THIS DIGIT
3198	014540	006105			ROL	R5	
3199	014542	006105			ROL	R5	
3200	014544	010503			MOV	R5,R3	
3201	014546	006103		35:	ROL	R3	:: GET LSB OF THIS DIGIT
3202	014550	105367	000076		DECB	\$OMODE	:: TYPE THIS DIGIT?
3203	014554	100016			BPL	7\$:: BR IF NO
3204	014556	042703	177770		BIC	#177770,R3	:: GET RID OF JUNK
3205	014562	001002			BNE	4\$:: TEST FOR 0
3206	014564	005704			TST	R4	:: SUPPRESS THIS 0'
3207	014566	001403			BEQ	5\$:: BR IF YES
3208	014570	005204		45:	INC	R4	:: DON'T SUPPRESS ANYMORE 0'S
3209	014572	052703	000060		BIS	#'0,R3	:: MAKE THIS DIGIT ASCII
3210	014576	052703	000040		BIS	#' ,R3	:: MAKE ASCII IF NOT ALREADY
3211	014602	110367	000040		MOV8	R3,8\$:: SAVE FOR TYPING
3212	014606	104401	014646		TYPE	8\$:: GO TYPE THIS DIGIT
3213	014612	105367	000032		75:	DECB	\$OCNT
3214	014616	003347			BGT	2\$:: COUNT BY 1
3215	014620	002402			BLT	6\$:: BR IF MORE TO DO
3216	014622	005204			INC	R4	:: BR IF DONE
3217	014624	000744			BR	2\$:: INSURE LAST DIGIT ISN'T A BLANK
3218	014626	012605		65:	MOV	(SP)+,R5	:: GO DO THE LAST DIGIT
3219	014630	012604			MOV	(SP)+,R4	:: RESTORE R5
3220	014632	012603			MOV	(SP)+,R3	:: RESTORE R4
3221	014634	016666	000002 000004		MOV	2(SP),4(SP)	:: RESTORE R3
3222	014642	012616			MOV	(SP)+,(SP)	:: SET THE STACK FOR RETURNING
3223	014644	000002			RTI		:: RETURN
3224	014646	000		85:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
3225	014647	000			.BYTE	0	:: TERMINATOR FOR TYPE ROUTINE
3226	014650	000		\$OCNT:	.BYTE	0	:: OCTAL DIGIT COUNTER
3227	014651	000		\$OFILL:	.BYTE	0	:: ZERO FILL SWITCH
3228	014652	000000		\$OMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE
3229							:: ENTER HERE ON POWER FAILURE
3230							
3231							
3232	014654			\$PWRDN:			
3233	014654	010046		.PFAIL:	MOV	R0,-(SP)	:: SAVE R0-R5 ON PROCESSOR STACK
3234	014656	010146			MOV	R1,-(SP)	
3235	014660	010246			MOV	R2,-(SP)	
3236	014662	010346			MOV	R3,-(SP)	
3237	014664	010446			MOV	R4,-(SP)	
3238	014666	010546			MOV	R5,-(SP)	
3239	014670	016746	163130		MOV	24,-(SP)	
3240	014674	010667	164216		MOV	SP,SAVSP	:: SAVE STACK POINTER
3241	014700	012767	014712 163116		MOV	#RESTART,24	:: SET UP FOR POWER UP TRAP
3242	014706	000000			HALT		:: HALT ON POWER DOWN NORMAL
3243	014710	000777			BR	.	
3244							
3245							:: PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

M05

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 67
 DZDURA.M11 12-OCT-76 10:29 BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0064

3246							
3247	014712	016706	164200	RESTAR:	MOV SAVSP, SP		;RESTORE STACK POINTER
3248	014716	012605			MOV (SP)+, R5		;RESTORE R0-R5
3249	014720	012604			MOV (SP)+, R4		
3250	014722	012603			MOV (SP)+, R3		
3251	014724	012602			MOV (SP)+, R2		
3252	014726	012601			MOV (SP)+, R1		
3253	014730	012600			MOV (SP)+, R0		
3254	014732	012767	014654 163064		MOV #PFAIL, 24		;SET UP FOR POWER FAILURE
3255	014740	106427	000340		MTPS #340		
3256	014744	012706	001100		MOV #STACK, SP		
3257	014750	005067	001102		CLR TEMP		
3258	014754	005267	001076		INC TEMP		
3259	014760	001375			BNE .-4		
3260	014762	104413			CONVRT		
3261	014764	015006			PFTAB		
3262	014766	104401			TYPE		
3263	014770	015310			MPFAIL		
3264	014772	005067	164405		CLR \$ERFLG		
3265	014776	005067	164414		CLR \$ERRPC		
3266	015002	000177	164076		JMP @RETURN		
3267	015006	000001		PFTAB:	1		
3268	015010	006	002		.BYTE 6,2		
3269	015012	000207			RETURN		
3270	015014	005015	042012 053125	MTITLE:	.ASCIZ <15><12><12>/DUV11 DZDUR-A TAPE B /<15><12>		
3271	015022	030461	042040 042132				
3272	015030	051125	040455 052040				
3273	015036	050101	020105 020102				
3274	015044	005015	000				
3275	015047	015	053012 041505	MVECTO:	.ASCIZ <15><12>/VEC ADD- /		
3276	015054	040440	042104 000055				
3277	015062	005015	051461 020124	MREGAD:	.ASCIZ <15><12>/1ST DEV: REC CSR ADD- /		
3278	015070	042504	035126 051040				
3279	015076	041505	041440 051123				
3280	015104	040440	042104 000055				
3281	015112	005015	052515 052114	MMULT:	.ASCIZ <15><12>/MULT DEV ? (Y OR N)- /		
3282	015120	042040	053105 037440				
3283	015126	024040	020131 051117				
3284	015134	047040	026451 000				
3285	015141	015	046012 051501	MLASTD:	.ASCIZ <15><12>/LAST DEV: REC CSR ADDR- /		
3286	015146	020124	042504 035126				
3287	015154	051040	041505 041440				
3288	015162	051123	040440 042104				
3289	015170	026522	000				
3290	015173	075	042504 044526	DEVICE:	.ASCIZ /=DEVICE /		
3291	015200	042503	020040 000				
3292	015205	015	051412 046105	MCOW:	.ASCIZ <15><12>/SELECT TO RUN @ACTREG /		
3293	015212	041505	020124 047524				
3294	015220	051040	047125 040040				
3295	015226	041501	051124 043505				
3296	015234	000					
3297	015235	015	047412 043126	MRANGE:	.ASCIZ <15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS- /		
3298	015242	047514	051072 052105				
3299	015250	050131	020105 040514				
3300	015256	052123	042040 053105				
3301	015264	051040	041530 051123				

N05

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 68
DZDURA.M11 12-OCT-76 10:29

BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0065

3302	015272	040440	042104	026523	
3303	015300	000			
3304	015301	040	037440	000	MQM: .ASCIZ / ?/
3305	015305	015	000012		MCRLF: .ASCIZ <15><12>
3306	015310	043120	044501	026114	MPFAIL: .ASCIZ /PFAIL, RESTART AT TEST IN PROGRESS/
3307	015316	020040	042522	052123	
3308	015324	051101	020124	052101	
3309	015332	052040	051505	020124	
3310	015340	047111	050040	047522	
3311	015346	051107	051505	000123	
3312	015354	005015	047105	020104	MEPASS: .ASCIZ <15><12>/END OF PASS TAPE B/
3313	015362	043117	050040	051501	
3314	015370	020123	040524	042520	
3315	015376	041040	000		
3316	015401	015	051012	000	MR: .ASCIZ <15><12>/R/
3317	015405	015	052012	051505	MTSTPC: .ASCIZ <15><12>/TEST PC-/
3318	015412	020124	041520	000055	
3319	015420	005015	047514	045503	MLOCK: .ASCIZ <15><12>/LOCK ON TEST? (Y OR N)-/
3320	015426	047440	020116	052040	
3321	015434	051505	037524	024040	
3322	015442	020131	051117	047040	
3323	015450	026451	000		
3324	015453	015	021412	047440	MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3325	015460	020106	054523	041516	
3326	015466	041440	040510	051522	
3327	015474	051440	046105	041505	
3328	015502	042524	020104	020050	
3329	015510	020061	051117	031040	
3330	015516	026451	000		
3331	015521	015	044412	020123	MWIRE6: .ASCIZ <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3332	015526	042523	020103	046530	
3333	015534	052111	051440	044507	
3334	015542	041524	020110	032505	
3335	015550	026465	020062	047111	
3336	015556	020077	054450	047440	
3337	015564	020122	024516	000055	
3338	015572	005015	051511	051440	MWIRE5: .ASCIZ <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3339	015600	041505	051040	041505	
3340	015606	051440	044527	041524	
3341	015614	020110	030505	026465	
3342	015622	020063	047111	020077	
3343	015630	054450	047440	020122	
3344	015636	024516	000055		
3345	015642	005015	051511	047440	MWIRE4: .ASCIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3346	015650	052120	041440	051114	
3347	015656	042440	040516	046102	
3348	015664	020105	053023	052111	
3349	015672	044103	042440	032465	
3350	015700	030455	044440	037516	
3351	015706	024040	020131	051117	
3352	015714	047040	026451	000	
3353	015721	015	005012	031510	MEXTJ: .ASCIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3354	015726	032461	041440	047117	
3355	015734	042516	052103	051117	
3356	015742	047440	020116	024077	
3357	015750	020131	051117	047040	

```

3358 015756 026451 000
3359 015761 015 020012 043536 MCNTG: .ASCIZ <15><12>/ ↑G /
3360 015766 020040 000
3361 015771 040 053523 036522 MMSWR: .ASCIZ / SWR= /
3362 015776 020040 000040
3363 016002 020040 047040 053505 MMNEW: .ASCIZ / NEW= /
3364 016010 020075 000040
3365 .EVEN
3366
3367 ;BUFFERS FOR INPUT-OUTPUT
3368
3369 016014 000000 INBUF: 0
3370 016056 .=. +40
3371 016056 000000 TEMP: 0
3372 016120 .=. +40
3373 016120 000000 MDATA: 0
3374 016162 .=. +40
3375 .SBTTL SCOPE HANDLER ROUTINE
3376
3377 ;*****
3378 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3379 ;AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3380 ;AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
3381 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3382 ;*SW14=1 LOOP ON TEST
3383 ;*SW11=1 INHIBIT ITERATIONS
3384 ;*SW09=1 LOOP ON ERROR
3385 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
3386 ;*CALL
3387 ;* SCOPE ;;SCOPE=IOT
3388
3389 016162 $SCOPE:
3390
3391 ;SCOPE LOOP AND INTERATION HANDLER
3392
3393 .SCOPE:
3394 016162 004767 174376 JSR PC,CKSWR
3395 016166 005067 163224 CLR _ERRPC ;CLEAR LAST ERROR PC
3396 016172 022716 003364 CMP #TST1+2,(SP) ;IS SCOPE AT BEGINING OF TEST 1?
3397 016176 001413 BEQ $XTSTR ;YES NO LOOP.
3398
3399 016200 000406 TTST: BR 1$ ;GO TO 1$ (IF LOCK SW02=1)
3400 016202 105777 163236 TSTB @STKS ;KEYBOARD DONE?
3401 016206 100123 BPL $OVER ;BR IF NO
3402 016210 017766 163232 177776 MOV @STKB,-2(SP) ;CLEAR DONE BIT
3403 016216 032777 040000 163214 1$: BIT #BIT14,@SWR ;LOOP ON PRESENT TEST?
3404 016224 001114 BNE $OVER ;YES IF SW14=1
3405 ;*****START OF CODE FOR THE XOR TESTER*****
3406 016226 000416 $XTSTR: BR 6$ ;IF RUNNING ON THE "XOR" TESTER CHANGE
3407 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
3408 016230 013746 000004 MOV @ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3409 016234 012737 016254 000004 MOV #5$,@ERRVEC ;SET FOR TIMEOUT
3410 016242 005737 177060 TST @177060 ;TIME OUT ON XOR?
3411 016246 012637 000004 MOV (SP)+,@ERRVEC ;RESTORE THE ERROR VECTOR
3412 016252 000463 BR $SVLAD ;GO TO THE NEXT TEST
3413 016254 022626 5$: CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
    
```

```

3414 016256 012637 000004      MOV      (SP)+, @ERRVEC      ;; RESTORE THE ERROR VECTOR
3415 016262 000423      BR       7$                ;; LOOP ON THE PRESENT TEST
3416 016264                6$:; *****END OF CODE FOR THE XOR TESTER*****
3417 016264 032777 000400 163146      BIT      @BIT08, @SWR        ;; LOOP ON SPEC. TEST?
3418 016272 001404                BEQ      2$                ;; BR IF NO
3419 016274 127767 163140 163100      CMPB    @SWR, $STNM         ;; ON THE RIGHT TEST?  SWR<7:0>
3420 016302 001465                BEQ      $OVER             ;; BR IF YES
3421 016304 105767 163073      2$:     TSTB    $ERFLG        ;; HAS AN ERROR OCCURRED?
3422 016310 001421                BEQ      3$                ;; BR IF NO
3423 016312 126767 163077 163063      CMPB    $ERMAX, $ERFLG     ;; MAX. ERRORS FOR THIS TEST OCCURRED?
3424 016320 101015                BHI      3$                ;; BR IF NO
3425 016322 032777 001000 163110      BIT      @BIT09, @SWR        ;; LOOP ON ERROR?
3426 016330 001404                BEQ      4$                ;; BR IF NO
3427 016332 016767 163052 163046      7$:     MOV      $LPERR, $LPADR   ;; SET LOOP ADDRESS TO LAST SCOPE
3428 016340 000446                BR       $OVER             ;;
3429 016342 105067 163035      4$:     CLRB    $ERFLG          ;; ZERO THE ERROR FLAG
3430 016346 005067 163140      CLR     $TIMES            ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
3431 016352 000415                BR       1$                ;; ESCAPE TO THE NEXT TEST
3432 016354 032777 004000 163056      3$:     BIT      @BIT11, @SWR     ;; INHIBIT ITERATIONS?
3433 016362 001011                BNE      1$                ;; BR IF YES
3434 016364 005767 163144      TST     $PASS            ;; IF FIRST PASS OF PROGRAM
3435 016370 001406                BEQ      1$                ;; INHIBIT ITERATIONS
3436 016372 005267 163006      INC     $ICNT            ;; INCREMENT ITERATION COUNT
3437 016376 025767 163110 163000      CMP     $TIMES, $ICNT      ;; CHECK THE NUMBER OF ITERATIONS MADE
3438 016404 002024                BGE     $OVER             ;; BR IF MORE ITERATION REQUIRED
3439 016406 012767 000001 162770      1$:     MOV      @1, $ICNT      ;; REINITIALIZE THE ITERATION COUNTER
3440 016414 016767 000056 163070      MOV     $MXCNT, $TIMES     ;; SET NUMBER OF ITERATIONS TO DO
3441 016422 105267 162754      $SVLAD: INCB    $STNM         ;; COUNT TEST NUMBERS
3442 016426 116767 162750 163076      MOVB   $STNM, $TESTN      ;; SET TEST NUMBER IN APT MAILBOX
3443 016434 011667 162746      MOV     (SP), $LPADR      ;; SAVE SCOPE LOOP ADDRESS
3444 016440 011667 162744      MOV     (SP), $LPERR      ;; SAVE ERROR LOOP ADDRESS
3445 016444 005067 163044      CLR     $ESCAPE          ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
3446 016450 112767 000001 162737      MOVB   @1, $ERMAX        ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3447 016456 016777 162720 162756      $OVER: MOV     $STNM, @DISPLAY ;; DISPLAY TEST NUMBER
3448 016464 016716 162716      MOV     $LPADR, (SP)      ;; FUDGE RETURN ADDRESS
3449 016470 000002      4$:     RTI
3450 016472 001407      BRW:    1407
3451 016474 000432      BRX:    432
3452 016476 000005      $MXCNT: 5                ;; MAX. NUMBER OF ITERATIONS
3453                .SBTTL TRAP DECODER
3454
3455                ; *****
3456                ; *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3457                ; *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3458                ; *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3459                ; *GO TO THAT ROUTINE.
3460
3461 016500 010046      STRAP: MOV     R0, -(SP)      ;; SAVE R0
3462 016502 016600 000002      MOV     2(SP), R0         ;; GET TRAP ADDRESS
3463 016506 005740                TST     -(R0)             ;; BACKUP BY 2
3464 016510 111000      MOVB   (R0), R0          ;; GET RIGHT BYTE OF TRAP
3465 016512 006300      ASL    R0                ;; POSITION FOR INDEXING
3466 016514 016000 016534      MOV     $TRPAD(R0), R0    ;; INDEX TO TABLE
3467 016520 000200      RTS     R0                ;; GO TO ROUTINE
3468
3469

```

3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525

016522 011646
016524 016666 000004 000002
016532 000002

016534 016522
016536 012734
016540 014452
016542 014426
016544 014466

016546 012712
016550 013216
016552 013324
016554 013334
016556 013534
016560 013574
016562 013626
016564 014004

016566 006367 000044
016572 006367 000040
016576 006367 000034
016602 006367 000030
016606 006367 000024
016612 016767 000020 000020
016620 162767 000001 000012
016626 042767 000037 000004
016634 0C0207
016636 000240
016640 000200

016642 016767 000126 163042
016650 005267 000120
016654 016767 000114 163032
016662 005267 000106
016666 016767 000102 163022
016674 016767 000074 163020
016702 005267 000066
016706 016767 000062 163004
016714 016767 000054 163002
016722 005267 000046

```
;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
$TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
RTI ;;RESTORE THE PSW

.SBTTL TRAP TABLE
;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
;*BY THE "TRAP" INSTRUCTION.

; ROUTINE
;-----
$TRPAD: .WORD $TRAP2
$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

.SCOPI ;;CALL=SCOPI TRAP+5(104405)
.INSTR ;;CALL=INSTR TRAP+6(104406)
.INSTER ;;CALL=INSTER TRAP+7(104407)
.PARAM ;;CALL=PARAM TRAP+10(104410)
.SAVOS ;;CALL=SAVOS TRAP+11(104411)
.RESOS ;;CALL=RESOS TRAP+12(104412)
.CONVRT ;;CALL=CONVRT TRAP+13(104413)
.SETFLG ;;CALL=SETFLG TRAP+14(104414)

;*****
;UTILITIES
;*****

;THIS UTILITY CALCULATES PRIORITY LEVEL
DULEV: ASL DUPRT ;SHIFT LEFT
ASL DUPRT ;
ASL DUPRT ;
ASL DUPRT ;
ASL DUPRT ;
MOV DUPRT,LESS1 ;MOVE THIS TO LESS1
SUB #1,LESS1 ;CREATE LESS1
BIC #37,LESS1 ;CLEAR TNZVC
RTS PC
DUPRT: PR5
LESS1: PR4 ;LEVEL TO ALLOW INTERRUPTS

;NEW DU ADDRESSES
DUADDR: MOV DUBASE,RXCSR ;XXX0
INC DUBASE
MOV DUBASE,HRXCSR ;XXX1
INC DUBASE
MOV DUBASE,RXDBUF ;XXX2
MOV DUBASE,PARCSR ;XXX2
INC DUBASE
MOV DUBASE,HRXDBUF ;XXX3
MOV DUBASE,HPARCSR ;XXX3
INC DUBASE
```

```

3526 016726 016767 000042 162772 MOV DUBASE, TXCSR ;XXX4
3527 016734 005267 000034 INC DUBASE
3528 016740 016767 000030 162762 MOV DUBASE, HTXCSR ;XXX5
3529 016746 005267 000022 INC DUBASE
3530 016752 016767 000016 162752 MOV DUBASE, TXDBUF ;XXX6
3531 016760 005267 000010 INC DUBASE
3532 016764 016767 000004 162742 MOV DUBASE, HTXDBUF ;XXX7
3533 016772 000207 RTS PC
3534 016774 000000 DUBASE: 0
3535
3536
3537 ; THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3538 ; INFORMATION CONTAINED IN STMP1 AND IT IS
3539 ; SHIFTED IN BY THE CONTENTS OF SHIFT
3539 016776 042777 040000 162722 RPOKE: BIC #MTDATA, @TXCSR
3540 017004 005067 162472 CLR STMP2
3541 017010 006067 162464 ROR STMP1 ; FORCE CARRY
3542 017014 006067 162462 ROR STMP2 ; PICK UP CARRY IN BIT 15
3543 017020 006267 162456 ASR STMP2 ; SHIFT INTO BIT 14
3544 017024 042767 100000 162450 BIC #BIT15, STMP2 ; CLR BIT 15
3545 017032 056777 162444 162666 BIS STMP2, @TXCSR ; POKE MAINT DATA
3546 017040 042777 020000 162660 BIC #CLK, @TXCSR ; POKE CLK
3547 017046 052777 020000 162652 BIS #CLK, @TXCSR ;
3548 017054 005367 162042 DEC SHIFT
3549 017060 001346 BNE RPOKE
3550 017062 000207 RTS PC
3551 ; THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3552 017064 016767 162410 162410 0008: MOV STMP1, STMP2 ; SAVE TEMP1
3553 017072 005067 162406 CLR STMP3
3554 017076 012727 000010 MOV #8., (PC)+
3555 017102 000000 4$: 0
3556 017104 006067 162372 1$: ROR STMP2
3557 017110 005567 162370 ADC STMP3
3558 017114 005367 177762 DEC 4$
3559 017120 001371 BNE 1$
3560 017122 006067 162356 ROR STMP3
3561 017126 103404 BCS 2$
3562 017130 052767 000400 162342 BIS #BIT8, STMP1 ; SET ODD PARITY
3563 017136 000403 BR 3$
3564 017140 042767 000400 162332 2$: BIC #BIT8, STMP1 ; CLR EVEN PARITY
3565 ; STMP1 NOW HAS ODD PARITY CHARACTER
3566 017146 000207 3$: RTS PC
3567
3568 ; THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3569 017150 016767 162324 162324 EVEN8: MOV STMP1, STMP2 ; SAVE TEMP1
3570 017156 005067 162322 CLR STMP3
3571 017162 012727 000010 MOV #8., (PC)+
3572 017166 000000 4$: 0
3573 017170 006067 162306 1$: ROR STMP2
3574 017174 005567 162304 ADC STMP3
3575 017200 005367 177762 DEC 4$
3576 017204 001371 BNE 1$
3577 017206 006067 162272 ROR STMP3
3578 017212 103004 BCC 2$
3579 017214 052767 000400 162256 BIS #BIT8, STMP1 ; SET EVEN PARITY
3580 017222 000403 BR 3$
3581 017224 042767 000400 162246 2$: BIC #BIT8, STMP1 ; CLR ODD PARITY

```

DZDUR-A MACY11 27(1006) 19-APR-77 10:06 PAGE 73
DZDURA.M11 12-OCT-76 10:29 TRAP TABLE

SEQ 0070

3582				.STMP1 NOW HAS EVEN PARITY CHARACTER
3583	017232	000207	3\$:	RTS PC
3584				
3585	017234	062716	TRPREG: ADD	*2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3586				;IN MAIN PART OF THE PROGRAM
3587	017240	000002		RTI
3588		000001	.END	

AAA	003200	1347*							
ABASE	= 000000	930	971						
ACDM1	= 000000	930	973						
ACDM2	= 000000	930	974						
ACPUOP	= 000000	930	945						
ACTREG	001166	789*	1303*	1317*	1318*	1325*	2702	2705	2715
ADD0	= 000000	930	975						
ADD1	= 000000	930	976						
ADD10	= 000000	930	985						
ADD11	= 000000	930	986						
ADD12	= 000000	930	987						
ADD13	= 000000	930	988						
ADD14	= 000000	930	989						
ADD15	= 000000	930	990						
ADD2	= 000000	930	977						
ADD3	= 000000	930	978						
ADD4	= 000000	930	979						
ADD5	= 000000	930	980						
ADD6	= 000000	930	981						
ADD7	= 000000	930	982						
ADD8	= 000000	930	983						
ADD9	= 000000	930	984						
ADEVCT	= 000000	930	936						
ADEVN	= 000000	930	972						
ADRCNT	013533	2917*	2956*	2963*					
RENV	= 000000	930	941						
REVM	= 000000	930	942						
AFATAL	= 000000	930	933						
AMADR1	= 000000	930	958						
AMADR2	= 000000	930	962						
AMADR3	= 000000	930	965						
AMADR4	= 000000	930	968						
AMAMS1	= 000000	930	952						
AMAMS2	= 000000	930	960						
AMAMS3	= 000000	930	963						
AMAMS4	= 000000	930	966						
AMSGAD	= 000000	930	938						
AMSLG	= 000000	930	939						
AMSGTY	= 000000	930	932						
AMTYP1	= 000000	930	953						
AMTYP2	= 000000	930	961						
AMTYP3	= 000000	930	964						
AMTYP4	= 000000	930	967						
APASS	= 000000	930	935						
APRIOR	= 000000	930							
APTCSU	000040	589*	2932						
APTENV	= 000001	545	587*	2825	3079				
APTSIZ	= 000200	586*	1222						
APTSPO	= 000100	547	588*	2827					
ASWREG	= 000000	930	943						
ATESTN	= 000000	930	934						
AUNIT	= 000000	930	937						
AUSWR	= 000000	930	944						
AVECT1	= 000000	930	969						
AVECT2	= 000000	930	970						
BASEAD	001154	784*	1285*	1322*	1323	1329*	1331*	2709*	2721* 2725

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0073

DF1	002132	1081	1085	1089	1093	1139#													
GH1	002067	1079	1083	1087	1128#														
DISPLA	001442	897#	1210*	1218*	1243*	3066*	3447*												
DISPRE	000174	738#	1218	1243															
DNA =	100000	853#	1046#																
DNAINT =	000040	860#	1053#																
DSC =	100000	817#	1010#																
DSINTE =	000040	827#	1020#																
DSR =	001000	823#	1016#																
DSWR =	177570	640#	896	1209															
DTR =	000002	831#	1024#																
DT1	002116	1080	1084	1088	1134#														
DT4	002126	1092	1137#																
DUADDR	016642	1284	2726	3516#															
DUBASE	016774	1280	1283	2725*	3516	3517*	3518	3519*	3520	3521	3522*	3523	3524	3525*					
		3526	3527*	3528	3529*	3530	3531*	3532	3534#										
DULEV	016566	1342	3503#																
DUPRT	016636	1341*	3503*	3504*	3505*	3506*	3507*	3508	3512#										
DURIS	001740	1108#	2729*																
DURIV	001736	1107#	1291	1294	1295	2727*	2734												
DUTIS	001744	1110#	2733*																
DUTIV	001742	1109#	2731*																
EIGHT =	006000	848#	1041#	1965	2020	2075	2130												
EMTVEC =	000030	729#	1194*	1195*															
EM1	001762	1078	1114#																
EM2	002022	1082	1120#																
EM3	002043	1086	1123#																
EM4	001746	1090	1112#																
ERRCNT	001114	757#																	
ERRVEC =	000004	722#	1207	1208*	1219*	3408	3409*	3411*	3414*										
EVEN8	017150	3569#																	
EVEPAR =	001400	851#	1044#																
EVPAR =	000400	840#	1033#																
FIVE =	000000	845#	1038#	1415	1470	2185	2237	2289	2341										
FRMERR =	020000	836#	1029#																
GNS =	***** U	3484	3485	3486	3487	3490	3491	3492	3493	3494	3495	3496	3497						
HDXEN =	000010	862#	1055#																
HILIM	013526	2914*	2944	2960#															
HLD0	001130	766#	1134	3097*															
HLD1	001132	767#	1134	3098*															
HLD2	001134	768#	3099*																
HLD3	001136	769#	3100*																
HLD4	001140	770#	3101*																
HLD5	001142	771#	3102*																
HLD6	001144	772#	3103*																
HOLD	001120	762#																	
HPARCS	001724	1101#	3524*																
HRXCSR	001714	1097#	3518*																
HRXDBU	001720	1099#	3523*																
HT =	000011	632#	2840	2881															
HTXCSR	001730	1103#	3528*																
HTXDBU	001734	1105#	3532*																
INBUF	016014	1350	1354	2890	2920	3038	3042	3369#											
INIFLG	001172	797#	1253	1256*															
INSTER =	104407	1358	2939	3047	3492#														
INSTR =	104406	1275	1286	1296	1307	1332	1348	1361	1365	1369	1373	1384	2780	3491#					

SN	= 000000	531#	2688#											
SNULL	001454	902#	2852	2881										
SNWTST=	000000	1405#	1460#	1515#	1570#	1625#	1680#	1735#	1790#	1845#	1900#	1955#	2010#	2065#
		2120#	2175#	2227#	2279#	2331#	2383#	2435#	2487#	2539#	2591#	2643#		
SOCNT	014650	3184#	3213#	3226#										
SOMODE	014652	3179#	3183#	3188	3191#	3202#	3228#							
SOVER	016456	3401	3404	3420	3428	3438	3447#							
SPASS	001534	935#	1221#	2747#	3434	3453								
SPASTH	002144	1171#												
SPWRDN	014654	1198	3232#											
SQUES	001522	923#	2881	3097										
SROCHR=	*****	3490												
SRODEC=	*****	3490												
SFDLIN=	*****	3490												
SROOCT=	*****	3490												
SREGAD	001460	906#												
SREGO	001462	908#	2976#	2981										
SREG1	001464	909#	2975#	2982										
SREG2	001466	910#	2974#	2983										
SREG3	001470	911#	2973#	2984										
SREG4	001472	912#	2972#	2985										
SREG5	001474	913#	2971#	2986										
SR2A =	*****	3490												
SSAVRE=	*****	3490												
SSCOPE	016162	1192	3389#											
SSETUP=	000017	1174#	1191	1192	1194	1196	1198	1200	1201	1203	3064	3089	3096	3390
SSTUP =	177777	1174#												
SSVLAD	016422	3412	3441#											
SSVPC =	002136	1146#	1151											
SSMR =	177400	522#	920	921	922	1200	1201	1203	1204	1407	1462	1517	1572	1627
		1682	1737	1792	1847	1902	1957	2012	2067	2122	2177	2229	2281	2333
		2385	2437	2489	2541	2593	2645	3055	3056	3057	3058	3059	3067	3074
		3086	3089	3097	3381	3382	3383	3384	3385	3403	3415	3417	3418	3421
		3422	3423	3430	3431	3432	3444	3447	3452					
SSWREG	001550	943#	1224											
SSWRMK=	000000	3385	3386	3419										
STESTN	001532	934#	3442#											
STIMES	001512	920#	1200#	3430#	3437	3440#	3452							
STKB	001446	899#	2771	2894	2902	3402								
STKS	001444	898#	2769	2892	3400									
STMP0	001476	914#												
STMP1	001500	915#	1426#	1438#	1442#	1481#	1493#	1497#	1536#	1548#	1552#	1591#	1603#	1607#
		1646#	1658#	1662#	1701#	1713#	1717#	1756#	1768#	1772#	1811#	1823#	1827#	1866#
		1878#	1882#	1921#	1933#	1937#	1976#	1988#	1992#	2031#	2043#	2047#	2086#	2098#
		2102#	2141#	2153#	2157#	2193#	2205#	2209#	2245#	2257#	2261#	2297#	2309#	2313#
		2349#	2361#	2365#	2401#	2413#	2417#	2453#	2465#	2469#	2505#	2517#	2521#	2557#
		2569#	2573#	2609#	2621#	2625#	2661#	2673#	2677#	3541#	3552	3562#	3564#	3569
		3579#	3581#											
STMP2	001502	916#	3540#	3542#	3543#	3544#	3545	3552#	3556#	3569#	3573#			
STMP3	001504	917#	3553#	3557#	3560#	3570#	3574#	3577#						
STMP4	001506	918#												
STMP5	001510	919#												
STN =	000031	590#	1405	1407#	1460	1462#	1515	1517#	1570	1572#	1625	1627#	1680	1682#
		1735	1737#	1790	1792#	1845	1847#	1900	1902#	1955	1957#	2010	2012#	2065
		2067#	2120	2122#	2175	2177#	2227	2229#	2279	2281#	2331	2333#	2383	2385#
		2435	2437#	2487	2489#	2539	2541#	2591	2593#	2643	2645#			

.SAPTY	1#	522#	533
.SASTA	1#		
.SCATC	1#	522#	
.SCHTA	1#	522#	869
.SDB2D	1#		
.SDB2D	1#		
.SDIV	1#		
.SEOP	1#	522#	
.SERRO	1#	522#	3049
.SERRT	1#	522#	3105
.SMULT	1#		
.SPOWE	1#	522#	
.SRAND	1#		
.SRDOE	1#		
.SRDOC	1#		
.SREAD	1#		
.SR2AZ	1#		
.SSAVE	1#		
.SSB2D	1#		
.SSB2D	1#		
.SSCOP	1#	522#	3375
.SSIZE	1#		
.SSUPR	1#		
.STRAP	1#	522#	3453
.STYPB	1#		
.STYPD	1#		
.STYPE	1#	522#	2802
.STYPO	1#	522#	3152
.S4OCA	1#		
.1170	1#		

. ABS. 017242 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

.DZDURA/SOL/CRF/NL: TOC=SYSMAC.C2, DZDUR1/EQ:RUNB, DZDUR2, DZDURA.M11
RUN-TIME: 20 18 1 SECONDS
RUN-TIME RATIO: 618/39=15.5
CORE USED: 43K (85 PAGES)

H07