

DUV/LSI-11

OFFLINE COMBINED TESTS
MD-11-DZDUV-A

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

APR 1977
digital
MADE IN USA

The image displays a grid of 48 small tables, arranged in 8 rows and 6 columns. Each table contains test data, including headers, numerical values, and alphanumeric strings. The data appears to be organized into columns, possibly representing different test parameters or components. The tables are densely packed and cover the left side of the page.

DR:DZDUVASEQ
DZDUV1.M11

000:0000
02-FEB-77 08:18

770419

801
PDP10 411

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 1

.REM *

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

PRODUCT CODE: MAINDEC-11-DZDUV-A-D

PRODUCT NAME: DUV11 OFFLINE COMBINED TESTS

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.

*

.REM *

GENERAL DESCRIPTION

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

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

* .REM *

2. REQUIREMENTS

* .REM *

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

- 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 DZDUV-A TAPE F" (ONCE ONLY)

*
 .REM *
 *
 .REM *

4.3.1.6 THE PROGRAM WILL TYPE "R" TO INDICATE THAT IT IS ABOUT
 TO START TESTING ,AND THEN TESTING WILL BEGIN

4.3.2 PROGRAM RESTART WITH ALL SWITCHES DOWN

4.3.2.1 THE PROGRAM WILL TYPE "R" AND WILL COMMENCE TESTING

4.3.3 PROGRAM RESTART WITH SW00=1

4.3.3.1 SET SWITCH REGISTER (LOC. 176) TO A 000001.

4.3.3.2 TYPE 200G.

4.3.3.3 PROGRAM WILL START.

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

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

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.4

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

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

IF AN INCORRECT ADDRESS IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.6

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

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

IF AN INCORRECT ANSWER IS GIVEN ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.8

IF A "NO" ANSWER IS GIVEN: JUMP TO SECTION 4.3.3.12
IF A "YES" ANSWER IS GIVEN:THE NEXT QUESTION IS ASKED

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

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

IF AN INCORRECT ANSWER IS TYPED ,THE PROGRAM WILL TYPE "?" AND WILL THEN REPEAT THE MESSAGE OF 4.3.3.10
NOTE:ALL ADDRESSES SHALL BE CONTIGUOUS

4.3.3.11.1 IF AN "OUT OF RANGE" ADDRESS IS TYPED IE. MORE THAN 16 (10) DEVICES AWAY (UPWARDS).....THE PROGRAM WILL TYPE "OUT OF RANGE:RETYPE LAST DEVICE RXCSR ADDRESS-" AND WAIT FOR AN INPUT FROM THE TELETYPE KEYBOARD

4.3.3.11.2 TYPE IN THE ADDRESS OF THE LAST RECEIVER CONTROL

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 5
 DZDUV1.M11 02-FEB-77 09:18

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

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 10
DZDUV1.M11 02-FEB-77 08:18

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

9. PROGRAM DESCRIPTION

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

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

11. LISTINGS

*
.REM *
*
.REM *
*

L01

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 12
DZDUV2.M11 02-FEB-77 08:20

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
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

000001

STN=1

M01

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 14
 DZDUVA.M11 13-OCT-76 09:26 APT COMMUNICATIONS ROUTINE

```

556 .ENABLE ABS
557
558 :DUV11 DZDUV-A TAPE F
559 ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
560
561 ;STARTING PROCEDURE
562 ;TYPE 200G
563 ;PROGRAM WILL TYPE "DUV11 DZDUV-A TAPE F "
564 ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
565 ;AT THE END OF A PASS, PROGRAM WILL TYPE "END OF PASS TAPE F"
566 ;AND THEN RESUME TESTING
567
568 .SBTTL BASIC DEFINITIONS
569
570 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
571 CO1100 STACK= 1100
572 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
573 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
574
575 ;*MISCELLANEOUS DEFINITIONS
576 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
577 000012 LF= 12 ;;CODE FOR LINE FEED
578 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
579 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
580 177776 PS= 177776 ;;PROCESSOR STATUS WORD
581 .EQUIV PS,PSW
582 177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
583 177772 PIRG= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
584 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
585 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
586
587 ;*GENERAL PURPOSE REGISTER DEFINITIONS
588 000000 R0= %0 ;;GENERAL REGISTER
589 000001 R1= %1 ;;GENERAL REGISTER
590 000002 R2= %2 ;;GENERAL REGISTER
591 000003 R3= %3 ;;GENERAL REGISTER
592 000004 R4= %4 ;;GENERAL REGISTER
593 000005 R5= %5 ;;GENERAL REGISTER
594 000006 R6= %6 ;;GENERAL REGISTER
595 000007 R7= %7 ;;GENERAL REGISTER
596 000006 SP= %6 ;;STACK POINTER
597 000007 PC= %7 ;;PROGRAM COUNTER
598
599 ;*PRIORITY LEVEL DEFINITIONS
600 000000 PR0= 0 ;;PRIORITY LEVEL 0
601 000040 PR1= 40 ;;PRIORITY LEVEL 1
602 000100 PR2= 100 ;;PRIORITY LEVEL 2
603 000140 PR3= 140 ;;PRIORITY LEVEL 3
604 000200 PR4= 200 ;;PRIORITY LEVEL 4
605 000240 PR5= 240 ;;PRIORITY LEVEL 5
606 000300 PR6= 300 ;;PRIORITY LEVEL 6
607 000340 PR7= 340 ;;PRIORITY LEVEL 7
608
609 ;*"SWITCH REGISTER" SWITCH DEFINITIONS
610 100000 SW15= 100000
611 040000 SW14= 40000
  
```

612	020000	SW13=	20000
613	010000	SW12=	10000
614	004000	SW11=	4000
615	002000	SW10=	2000
616	001000	SW09=	1000
617	000400	SW08=	400
618	000200	SW07=	200
619	000100	SW06=	100
620	000040	SW05=	40
621	000020	SW04=	20
622	000010	SW03=	10
623	000004	SW02=	4
624	000002	SW01=	2
625	000001	SW00=	1
626		.EQUIV	SW09,SW9
627		.EQUIV	SW08,SW8
628		.EQUIV	SW07,SW7
629		.EQUIV	SW06,SW6
630		.EQUIV	SW05,SW5
631		.EQUIV	SW04,SW4
632		.EQUIV	SW03,SW3
633		.EQUIV	SW02,SW2
634		.EQUIV	SW01,SW1
635		.EQUIV	SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

638	100000	BIT15=	100000
639	040000	BIT14=	40000
640	020000	BIT13=	20000
641	010000	BIT12=	10000
642	004000	BIT11=	4000
643	002000	BIT10=	2000
644	001000	BIT09=	1000
645	000400	BIT08=	400
646	000200	BIT07=	200
647	000100	BIT06=	100
648	000040	BIT05=	40
649	000020	BIT04=	20
650	000010	BIT03=	10
651	000004	BIT02=	4
652	000002	BIT01=	2
653	000001	BIT00=	1
654		.EQUIV	BIT09,BIT9
655		.EQUIV	BIT08,BIT8
656		.EQUIV	BIT07,BIT7
657		.EQUIV	BIT06,BIT6
658		.EQUIV	BIT05,BIT5
659		.EQUIV	BIT04,BIT4
660		.EQUIV	BIT03,BIT3
661		.EQUIV	BIT02,BIT2
662		.EQUIV	BIT01,BIT1
663		.EQUIV	BIT00,BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

666	000004	ERRVEC=	4	:: TIME OUT AND OTHER ERRORS
667	000010	RESVEC=	10	:: RESERVED AND ILLEGAL INSTRUCTIONS

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 16
DZDUVA.M11 13-OCT-76 08:26 BASIC DEFINITIONS

668	000014	TBITVEC=14	:: "T" BIT
669	000014	TRTVEC= 14	:: TRACE TRAP
670	000014	BPTVEC= 14	:: BREAKPOINT TRAP (BPT)
671	000020	IOTVEC= 20	:: INPUT/OUTPUT TRAP (IOT) **SCOPE**
672	000024	PWRVEC= 24	:: POWER FAIL
673	000030	EMTVEC= 30	:: EMULATOR TRAP (EMT) **ERROR**
674	000034	TRAPVEC=34	:: "TRAP" TRAP
675	000060	TKVEC= 60	:: TTY KEYBOARD VECTOR
676	000064	TPVEC= 64	:: TTY PRINTER VECTOR
677	000240	PIRQVEC=240	:: PROGRAM INTERRUPT REQUEST VECTOR

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 17
 DZDUVA.M11 13-OCT-76 08:26 BASIC DEFINITIONS

```

678                                     ;STANDARD INTERRUPT VECTORS
679
680
681      000174      000174      . =174
682      000174      000000      DISPREG:0
683      000176      000000      SWREG:0
684      000200      000200      . =200
685      000200      000167      001746      JMP      .START      ;GO TO START OF PROGRAM
686
687
688
689      001100      001100      . =1100
690      001100      000000      .WORD 0
691      001102      177570      LIGHTS:177570
692
693
694
695                                     ;PROGRAM CONTROL PARAMETERS
696
697      001104      000000      RETURN: 0
698      001106      000000      NEXT: 0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
699      001110      000000      LOCK: 0      ;ADDRESS FOR LOCK ON CURRENT DATA
700      001112      000000      PASCNT: 0      ;ADDRESS CONTAINING PASS COUNT
701      001114      000000      ERRCNT: 0      ;ERROR COUNT
702      001116      000000      SAVSP: 0      ;STACK POINTER STORAGE
703
704                                     ;PROGRAM VARIABLES
705
706      001120      000020      HOLD: 20      ;TEMPORARY STORAGE=DELAY TIME FOR CABLES
707      001122      000000      SHIFT: 0      ;TEMPORARY STORAGE= # OF SHIFTS PER CHAR
708      001124      000000      COUNT: 0      ;TEMPORARY STORAGE= # OF TIMES A CHAR WILL BE SENT
709      001126      000000      SAVPC: 0      ;PROGRAM COUNTER STORAGE
710      001130      000000      HLD0: 0
711      001132      000000      HLD1: 0
712      001134      000000      HLD2: 0
713      001136      000000      HLD3: 0
714      001140      000000      HLD4: 0
715      001142      000000      HLD5: 0
716      001144      000000      HLD6: 0
717

```


DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 18
 DZDUVA.M11 13-OCT-76 08:26 BASIC DEFINITIONS

```

718                                     ;PROGRAM CONVERSATIONAL PARAMETERS
719 001146      377      SYNCNO: .BYTE 377      ;# OF SYNC CHARS REQ'D FOR SYNC'ZATION
720 001147      377      SEXMIT: .BYTE 377      ;SEC XMIT JUMPER "IN"
721 001150      377      SEREC:  .BYTE 377      ;SEC REC JUMPER "IN"
722 001151      377      OPTCLR: .BYTE 377      ;OPTIONAL JUMPER CLR "IN"
723 001152      000      MULTD:  .BYTE 0        ;NO MULTIPLE DEVICE FLAG
724 001153      377      JMRBY:  .BYTE 377      ;EXTERNAL MODEM BYPASS JUMPER "IN"
725                                     .EVEN
726
727                                     ;PROGRAM MULTIPLE DEVICE PARAMETERS
728 001154 000000  BASEADD:      0        ;PROG CONTROLLED 1ST DEVICE ADDR
729 001156 000000  KEEPADD:     0        ;SAVED 1ST DEVICE ADDR
730 001160 000000  LASTADD:     0        ;LAST DEVICE RXCSR ADDR
731 001162 000000  BASEIV:      0        ;PROG CONTROLLED IV
732 001164 000000  KEEPIV:      0        ;SAVED INTR VECTOR
733 001166 000000  ACTREG:      0        ;ACTIVE REGISTER , , MODIFY THIS
734                                     ;LOCATION TO DISQUALIFY OR QUALIFY
735                                     ;DEVICES (1= RUN , 0= DON'T RUN)
736 001170 000000  ROTADD:      0        ;ROTATING POINTER FOR ACTREG. POINTS
737                                     ;TO DEVICE PRESENTLY UNDER TEST WHEN RUNNING MULTIPLE DEVICES
738
739                                     ;PROGRAM CONTROL FLAGS
740
741 001172      000      INIFLG: .BYTE 0        ;PROGRAM INITIALIZATION FLAG
742 001173      000      STFLG:  .BYTE 0        ;TEST START FLAG
743 001174      000      LOKFLG: .BYTE 0        ;LOCK ON CURRENT TEST FLAG
744                                     .EVEN
745                                     . =1400
746
747

```

```

748
749
750
751      ; INSTRUCTION DEFINITIONS
752
753      005746      PUSH1SP=5746      ; DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
754      005726      POP1SP=5726      ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
755      010046      PUSHRO=10046     ; SAVE RO ON STACK =MOV RO, -(SP)
756      012600      POPRO=12600     ; RESTORE RO FROM STACK =MOV (SP)+, RO
757      024646      PUSH2SP=24646   ; DECREMENT STACK TWICE =CMP -(SP), -(SP)
758      022626      POP2SP=22626    ; INCREMENT STACK TWICE =CMP (SP)+, (SP)+
759      ; REGISTER DEFINITIONS
760      ; RXCSR BIT DEFINITIONS
761      100000      DSC=BIT15      ; DATA SET CHANGE
762      040000      RING=BIT14     ; RING
763      020000      CTS=BIT13      ; CLR TO SEND
764      010000      CARDET=BIT12   ; CARRIER DETECT
765      004000      RECACT=BIT11   ; REC ACTIVE
766      002000      SRD=BIT10      ; SEC REC DATA
767      001000      DSR=BIT9       ; DATA SET RDY
768      000400      STPSYN=BIT8    ; STRIP SYNC
769      000200      RXDONE=BIT7    ; REC DONE
770      000100      RINTEN=BIT6    ; REC INTR ENABLE
771      000040      DSINTE=BIT5    ; DSC INTR ENABLE
772      000020      SYN SCH=BIT4   ; SYNC SEARCH
773      000010      STD=BIT3       ; SEC XMIT DATA
774      000004      RTS=BIT2       ; REQ TO SEND
775      000002      DTR=BIT1       ; DATA TERM RDY
776      000001      VOID=BIT0
777      ; RXDBUF BIT DEFINITIONS
778      100000      RXERR=BIT15     ; REC ERROR
779      040000      OVRUN=BIT14    ; OVERRUN
780      020000      FRMERR=BIT13   ; FRAME ERROR
781      010000      PARER=BIT12    ; PARITY ERROR
782      ; PARCSR BIT DEFINITIONS
783      001000      PAREN=BIT9      ; PARITY ENABLE
784      000400      EVPAR=BIT8     ; EVEN PARITY SENSE
785      ; PARCSR WRD DEFINITIONS
786      030000      SYNINT=30000    ; SYNC EXTERNAL MODE
787      020000      SYNEXT=20000   ; SYNC INTERNAL MODE
788      000000      ISYMOD=0       ; ISOC MODE
789      000000      FIVE=0         ; WORD LENGTH 5 BITS
790      002000      SIX=2000       ; WORD LENGTH 6 BITS
791      004000      SEVEN=4000     ; WORD LENGTH 7 BITS
792      006000      EIGHT=6000    ; WORD LENGTH 8 BITS
793      000000      NOPAR=0        ; NO PARITY
794      001000      ODDPAR=1000    ; ODD PARITY
795      001400      EVEPAR=1400    ; EVEN PARITY
796      ; TXCSR BIT DEFINITIONS
797      100000      DNA=BIT15       ; DATA NOT AVAILABLE
798      040000      MTDATA=BIT14   ; MAINT DATA
799      020000      CLK=BIT13      ; CLK
800      002000      BITW=BIT10     ; BIT WINDOW
801      000400      MRESET=BIT8    ; MASTER RESET
802      000200      TXDONE=BIT7   ; XMIT DONE
803      000100      TXINTE=BIT6    ; XMIT INTR ENABLE

```

F02

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 20
DZDUVA.M11 13-OCT-76 08:26 BASIC DEFINITIONS

804	000040	DNAINTE=BIT5	;DNA INTR ENAB
805	000020	SEND=BIT4	;SEND
806	000010	HDXEN=BIT3	;HDX/FDX
807	000001	BREAK=BIT0	;BREAK
808		;TXCSR WRD DEFINITIONS	
809	000000	USER=0	;USER MODE
810	004000	MINT=4000	;MAINT INT MODE
811	010000	MEXT=10000	;MAINT EXT MODE
812	014000	SYSTST=14000	;SYSTEM TEST MODE

813
814
815
816
817
818
819 001400
820 001400 001400
821 001400 000000
822 001402 000
823 001403 000
824 001404 000000
825 001406 000000
826 001410 000000
827 001412 000000
828 001414 000
829 001415 001
830 001416 000000
831 001420 000000
832 001422 000000
833 001424 000000
834 001426 000000
835 001430 000000
836 001432 000000
837 001434 000
838 001435 000
839 001436 000000
840 001440 177570
841 001442 177570
842 001444 177560
843 001446 177562
844 001450 177564
845 001452 177566
846 001454 000
847 001455 002
848 001456 012
849 001457 000
850 001460 000000
851
852 001462 000000
853 001464 000000
854 001466 000000
855 001470 000000
856 001472 000000
857 001474 000000
858 001476 000000
859 001500 000000
860 001502 000000
861 001504 000000
862 001506 000000
863 001510 000000
864 001512 000000
865 001514 000000
866 001516 177607 000377
867 001522 077
868 001523 015

.SBTTL COMMON TAGS

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

SCMTAG: =.
; ; START OF COMMON TAGS
\$TSTNM: .WORD 0 ; CONTAINS THE TEST NUMBER
\$ERFLG: .BYTE 0 ; CONTAINS ERROR FLAG
\$ICNT: .WORD 0 ; CONTAINS SUBTEST ITERATION COUNT
\$LPADR: .WORD 0 ; CONTAINS SCOPE LOOP ADDRESS
\$LPERR: .WORD 0 ; CONTAINS SCOPE RETURN FOR ERRORS
\$ERTTL: .WORD 0 ; CONTAINS TOTAL ERRORS DETECTED
\$ITEMB: .BYTE 0 ; CONTAINS ITEM CONTROL BYTE
\$ERMAX: .BYTE 1 ; CONTAINS MAX. ERRORS PER TEST
\$ERRPC: .WORD 0 ; CONTAINS PC OF LAST ERROR INSTRUCTION
\$GDADR: .WORD 0 ; CONTAINS ADDRESS OF 'GOOD' DATA
\$BDADR: .WORD 0 ; CONTAINS ADDRESS OF 'BAD' DATA
\$GDDAT: .WORD 0 ; CONTAINS 'GOOD' DATA
\$BDDAT: .WORD 0 ; CONTAINS 'BAD' DATA
; ; RESERVED--NOT TO BE USED
\$AUTOB: .BYTE 0 ; AUTOMATIC MODE INDICATOR
\$INTAG: .BYTE 0 ; INTERRUPT MODE INDICATOR
\$SWR: .WORD 0 ; ADDRESS OF SWITCH REGISTER
\$DISP: .WORD 0 ; ADDRESS OF DISPLAY REGISTER
\$TKS: 177560 ; TTY KBD STATUS
\$TKB: 177562 ; TTY KBD BUFFER
\$TPS: 177564 ; TTY PRINTER STATUS REG. ADDRESS
\$TPB: 177566 ; TTY PRINTER BUFFER REG. ADDRESS
\$NULL: .BYTE 0 ; CONTAINS NULL CHARACTER FOR FILLS
\$FILLS: .BYTE 2 ; CONTAINS # OF FILLER CHARACTERS REQUIRED
\$FILLC: .BYTE 12 ; INSERT FILL CHARS. AFTER A "LINE FEED"
\$STPFLG: .BYTE 0 ; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
\$REGAD: .WORD 0 ; CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED
\$REGO: .WORD 0 ; CONTAINS ((\$REGAD)+0)
\$REG1: .WORD 0 ; CONTAINS ((\$REGAD)+2)
\$REG2: .WORD 0 ; CONTAINS ((\$REGAD)+4)
\$REG3: .WORD 0 ; CONTAINS ((\$REGAD)+6)
\$REG4: .WORD 0 ; CONTAINS ((\$REGAD)+10)
\$REG5: .WORD 0 ; CONTAINS ((\$REGAD)+12)
\$TMP0: .WORD 0 ; USER DEFINED
\$TMP1: .WORD 0 ; USER DEFINED
\$TMP2: .WORD 0 ; USER DEFINED
\$TMP3: .WORD 0 ; USER DEFINED
\$TMP4: .WORD 0 ; USER DEFINED
\$TMP5: .WORD 0 ; USER DEFINED
\$TIMES: 0 ; MAX. NUMBER OF ITERATIONS
\$ESCAPE: 0 ; ESCAPE ON ERROR ADDRESS
\$BELL: .ASCIZ <207><377><377> ; CODE FOR BELL
\$QUES: .ASCII '?' ; QUESTION MARK
\$CRLF: .ASCII <15> ; CARRIAGE RETURN

H02

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 22
DZDUVA.M11 13-OCT-76 08:26 COMMON TAGS

869 001524 000012
870
871
872
873
874
875 001526
876 001526 000000
877 001530 000000
878 001532 000000
879 001534 000000
880 001536 000000
881 001540 000000
882 001542 000000
883 001544 000000
884 001546
885 001546 000
886 001547 000
887 001550 000000
888 001552 000000
889 001554 000000
890
891
892
893
894
895
896 001556 000
897 001557 000
898
899
900
901
902 001560 000000
903
904 001562 000
905 001563 000
906 001564 000000
907 001566 000
908 001567 000
909 001570 000000
910 001572 000
911 001573 000
912 001574 000000
913 001576 000000
914 001600 000000
915 001602 000000
916 001604 000000
917 001606 000000
918 001610 000000
919 001612 000000
920 001614 000000
921 001616 000000
922 001620 000000
923 001622 000000
924 001624 000000

\$LF: .ASCIZ <12> ; ;LINE FEED
;*****
\$BTTL 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


```

941
942
943
944
945
946      005746      PUSH1SP=5746      ; DECREMENT PROCESSOR STACK 1 WORD =TST -(SP)
947      005726      POP1SP=5726       ; INCREMENT PROCESSOR STACK 1 WORD =TST (SP)+
948      010046      PUSHRO=10046     ; SAVE RO ON STACK =MOV RO, -(SP)
949      012600      POPRO=12600      ; RESTORE RO FROM STACK =MOV (SP)+, RO
950      024646      PUSH2SP=24646   ; DECREMENT STACK TWICE =CMP -(SP), -(SP)
951      022626      POP2SP=22626    ; INCREMENT STACK TWICE =CMP (SP)+, (SP)+
952
953      ; REGISTER DEFINITIONS
954      100000      DSC=BIT15        ; RXCSR BIT DEFINITIONS
955      040000      RING=BIT14       ; DATA SET CHANGE
956      020000      CTS=BIT13        ; RING
957      010000      CARDET=BIT12     ; CLR TO SEND
958      004000      REACT=BIT11     ; CARRIER DETECT
959      002000      SPD=BIT10       ; REC ACTIVE
960      001000      DSR=BIT9        ; SEC REC DATA
961      000400      STPSYN=BIT8     ; DATA SET RDY
962      000200      RXDONE=BIT7     ; STRIP SYNC
963      000100      RINTEN=BIT6     ; REC DONE
964      000040      DSINTE=BITS     ; REC INTR ENABLE
965      000020      SYN SCH=BIT4     ; DSC INTR ENABLE
966      000010      STD=BIT3        ; SYNC SEARCH
967      000004      RTS=BIT2        ; SEC XMIT DATA
968      000002      DTR=BIT1        ; REQ TO SEND
969      000001      VOID=BIT0       ; DATA TERM RDY
970
971      100000      RXERR=BIT15      ; RXDBUF BIT DEFINITIONS
972      040000      OVRUN=BIT14     ; REC ERROR
973      020000      FRMERR=BIT13    ; OVERRUN
974      010000      PARER=BIT12     ; FRAME ERROR
975
976      001000      PAREN=BIT9       ; PARCSR BIT DEFINITIONS
977      000400      EVPAR=BIT8      ; PARITY ENABLE
978
979      030000      SYNINT=30000     ; PARCSR WRD DEFINITIONS
980      020000      SYNEXT=20000    ; SYNC EXTERNAL MODE
981      000000      ISYMOD=0        ; SYNC INTERNAL MODE
982      000000      FIVE=0         ; ISOC MODE
983      002000      SIX=2000       ; WORD LENGTH 5 BITS
984      004000      SEVEN=4000     ; WORD LENGTH 6 BITS
985      006000      EIGHT=6000    ; WORD LENGTH 7 BITS
986      000000      NOPAR=0        ; WORD LENGTH 8 BITS
987      001000      ODDPAR=1000    ; NO PARITY
988      001400      EVEPAR=1400    ; ODD PARITY
989
990      100000      DNA=BIT15       ; TXCSR BIT DEFINITIONS
991      040000      MTDATA=BIT14    ; DATA NOT AVAILABLE
992      020000      CLK=BIT13      ; MAINT DATA
993      002000      BITW=BIT10     ; CLK
994      000400      MRESET=BIT8    ; BIT WINDOW
995      000200      TXDONE=BIT7    ; MASTER RESET
996      000100      TXINTE=BIT6    ; XMIT DONE
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
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
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200

```

K02

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 25
DZDUVA.M11 13-OCT-76 09:26 APT MAILBOX-ETABLE

997	000040	DNAINTE=BIT5	;DNA INTR ENAB
998	000020	SEND=BIT4	;SEND
999	000010	HDXEN=BIT3	;HDX/FDX
1000	000001	BREAK=BIT0	;BREAK
1001		;TXCSR WRD DEFINITIONS	
1002	000000	USER=0	;USER MODE
1003	004000	MINT=4000	;MAINT INT MODE
1004	010000	MEXT=10000	;MAINT EXT MODE
1005	014000	SYSTST=14000	;SYSTEM TEST MODE

1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061

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 \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

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

\$ERRTB:
;ERROR TABLE
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/

M02

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 27
 DZDUVA.M11 13-OCT-76 08:26 ERROR POINTER TABLE

1062	002012	044507	052123	051105		
1063	002020	000123				
1064	002022	020040	042522	042503	EM2:	.ASCIZ / RECEIVER ERROR/
1065	002030	053111	051105	042440		
1066	002036	051122	051117	000		
1067	002043	040	052040	040522	EM3:	.ASCIZ / TRANSMITTER ERROR/
1068	002050	051516	044515	052124		
1069	002056	051105	042440	051122		
1070	002064	051117	000			
1071						;DATA HEADERS FOR ERROR MESSAGES
1072	002067	105	051122	041520	DH1:	.ASCIZ /ERRPC WANTED ACTUAL/
1073	002074	020040	040527	052116		
1074	002102	042105	020040	041501		
1075	002110	052524	046101	000		
1076		002116				.EVEN
1077						;DATA TABLES FOR ERROR MESSAGES
1078	002116	001416	001130	001132	DT1:	.WORD \$ERRPC,HLDD,HLDI,0
1079	002124	000000				
1080						
1081	002126	001416	000000		DT4:	.WORD \$ERRPC,0
1082						
1083	002132	000	000	000	DF1:	.BYTE 0,0,0,0
1084	002135	000				
1085						.EVEN
1086						.SBTTL ACT11 HOOKS
1087						
1088						;;*****
1089						;HOOKS REQUIRED BY ACT11
1090		002136				\$SVPC=.
1091		000046				;;SAVE PC
1092	000046	012662				.=46
1093		000052				\$ENDAD
1094	000052	000000				;;1)SET LOC.46 TO ADDRESS OF \$ENDAD IN \$.EOP
1095		002136				.=52
1096						.WORD 0
1097						;;2)SET LOC.52 TO ZERO
1098						;; RESTORE PC
1099						.SBTTL APT PARAMETER BLOCK
1100						;;*****
1101		002136				;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1102		000024				;;*****
1103	000024	000200				.\$X=.
1104		000044				;;SAVE CURRENT LOCATION
1105	000044	002136				.=24
1106		002136				200
1107						;;SET POWER FAIL TO POINT TO START OF PROGRAM
1108						FOR APT START UP
1109						.=44
1110						POINT TO APT INDIRECT ADDRESS PNTR.
1111	002136					\$APTHDR
1112	002136	000000				POINT TO APT HEADER BLOCK
1113	002140	001526				.\$X
1114	002142	000010				;;RESET LOCATION COUNTER
1115	002144	000010				;;*****
1116	002146	000000				;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1117	002150	000052				;INTERFACE SPEC.
						\$APTHD:
						\$SHIBTS: .WORD 0
						;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
						\$MADDR: .WORD \$MAIL
						;;ADDRESS OF APT MAILBOX (BITS 0-15)
						\$TSTM: .WORD 10
						;;RUN TIM OF LONGEST TEST
						\$PASTM: .WORD 10
						;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
						\$UNITM: .WORD
						;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
						.WORD \$ETEND-\$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

1118
1119
1120                                     :PROGRAM INITIALIZATION
1121                                     :LOCK OUT INTERRUPTS
1122                                     :SET UP PROCESSOR STACK
1123                                     :SET UP POWER FAIL VECTOR
1124                                     :CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1125                                     :TYPE TITLE MESSAGE
1126
1127 002152 .START:
1128 .SBTTL INITIALIZE THE COMMON TAGS
1129 ;;CLEAR THE COMMON TAGS (%CMTAG) AREA
1130 002152 012706 001400 MOV    %CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1131 002156 005026 CLR    (R6)+          ;;CLEAR MEMORY LOCATION
1132 002160 022706 001440 CMP    %SWR,R6      ;;DONE?
1133 002164 001374 BNE    -6             ;;LOOP BACK IF NO
1134 002166 012706 001100 MOV    %%STACK,SP    ;;SETUP THE STACK POINTER
1135 ;;INITIALIZE A FEW VECTORS
1136 002172 012737 016312 000020 MOV    %%SCOPE,%IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1137 002200 012737 000340 000022 MOV    #340,%IOTVEC+2 ;;LEVEL 7
1138 002206 012737 014202 000030 MOV    %SEROR,%EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1139 002214 012737 000340 000032 MOV    #340,%EMTVEC+2 ;;LEVEL 7
1140 002222 012737 016630 000034 MOV    %STRAP,%TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1141 002230 012737 000340 000036 MOV    #340,%TRAPVEC+2;LEVEL 7
1142 002236 012737 015004 000024 MOV    %SPWRDN,%PWRVEC ;;POWER FAILURE VECTOR
1143 002244 012737 000340 000026 MOV    #340,%PWRVEC+2 ;;LEVEL 7
1144 002252 005067 177234 CLR    $TIMES        ;;INITIALIZE NUMBER OF ITERATIONS
1145 002256 005067 177232 CLR    %ESCAPE       ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1146 002262 112767 000001 177125 MOVB  #1,%$ERMAX     ;;ALLOW ONE ERROR PER TEST
1147 002270 012767 002270 177110 MOV    #,%$LPADR     ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1148 002276 012767 002276 177104 MOV    #,%$LPERR     ;;SETUP THE ERROR LOOP ADDRESS
1149 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1150 ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1151 002304 013746 000004 MOV    %ERRVEC,-(SP) ;;SAVE ERROR VECTOR
1152 002310 012737 002344 000004 MOV    #64,%ERRVEC  ;;SET UP ERROR VECTOR
1153 002316 012767 177570 177114 MOV    %DSWR,%SWR   ;;SETUP FOR A HARDWARE SWICH REGISTER
1154 002324 012767 177570 177110 MOV    %DDISP,%DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1155 002332 022777 177777 177100 CMP    #-1,%SWR     ;;TRY TO REFERENCE HARDWARE SWR
1156 002340 001012 BNE    66$         ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1157 ;;AND THE HARDWARE SWR IS NOT = -1
1158 002342 000403 BR    65$         ;;BRANCH IF NO TIMEOUT
1159 002344 012716 002352 64$: MOV    #65,(SP)      ;;SET UP FOR TRAP RETURN
1160 002350 000002 RTI
1161 002352 012767 000176 177060 65$: MOV    %SWREG,%SWR   ;;POINT TO SOFTWARE SWR
1162 002360 012767 000174 177054 MOV    %DISPREG,%DISPLAY
1163 002366 012637 000004 66$: MOV    (SP)+,%ERRVEC ;;RESTORE ERROR VECTOR
1164
1165 002372 005067 177136 CLR    %PASS        ;;CLEAR PASS COUNT
1166 002376 132767 000200 177143 BITB  %APTSIZE,%SENVM ;;TEST USER SIZE UNDER APT
1167 002404 001403 BEQ    67$         ;;YES,USE NON-APT SWITCH
1168 002406 012767 001550 177024 MOV    %%SWREG,%SWR ;;NO,USE APT SWITCH REGISTER
1169 67$:
1170 002414 012706 001100 MOV    %STACK,SP    ;;SET STACK
1171 002420 106427 000340 MTPS  #340          ;;LOCK INTERRUPTS
1172 002424 012737 015004 000024 MOV    #,%PFAIL,%24 ;;SET UP POWER FAIL VECTOR
1173 002432 105067 176535 CLRB  $STFLG        ;;CLEAR START FLAG

```

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 29
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1174 002436 005067 176450 CLR PASCNT ;CLEAR PASS COUNT
1175 002442 105067 176735 CLRB SERFLG ;CLEAR ERROR FLAG
1176 002446 005067 176740 CLR SERTTL ;CLEAR ERROR COUNT
1177 002452 005067 176740 CLR ?ERRPC ;CLEAR LAST ERROR POINTER
1178 002456 012767 000001 176716 MOV #1,$STSTNM ;SET UP FOR TEST 1
1179 002464 012767 002152 176412 MOV #,$START,RETURN ;SET UP FOR POWER FAIL BEFORE
1180 ;TESTING STARTS
1181 002472 013746 000006 MOV @#6,-(SP)
1182 002476 013746 000004 MOV @#4,-(SP)
1183 002502 012737 002516 000004 MOV #15,@#4
1184 002510 005777 176724 TST @SWR
1185 002514 000407 BR 2$
1186 002516 012767 000176 176714 15: MOV #SWREG,SWR
1187 002524 012767 000174 176710 MOV #DISPREG,DISPLAY
1188 002532 022626 CMP (SP)+,(SP)+
1189 002534 012637 000004 25: MOV (SP)+,@#4
1190 002540 012637 000006 MOV (SP)+,@#6
1191 002544 022767 000176 176666 CMP #SWREG,SWR
1192 002552 001007 BNE 3$
1193 002554 005737 000042 TST @#42 ;CHECK FOR CHAIN
1194 002560 001402 BEQ 33$
1195 002562 000167 000522 JMP .BEGIN
1196 002566 004767 010176 33$: JSR PC,CNTLU
1197 002572 105767 176374 35: TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1198 002576 001004 BNE ONCE
1199 002600 104401 015144 TYPE #TITLE ;TYPE TITLE MESSAGE
1200 002604 105167 176362 COMB #INIFLG ;IF NOT SET FLAG AND DO
1201 002610 105767 176732 ONCE: TSTB #ENV ;APT CONTROL?
1202 002614 001410 BEQ 11$ ;BR IF NO
1203 002616 032767 000001 176726 BIT #1,$USWR ;EXTENAL JUMPER ON?
1204 002624 001002 BNE 12$ ;NO
1205 002626 105067 176321 CLRB #JMRBY ;CLEAR FLAG
1206 002632 000167 000452 12$: JMP .BEGIN ;GO DO IT
1207 002636 032777 000001 176574 11$: BIT #SW00,@SWR ;RESELECT VECTOR & CONTROL REG?
1208 002644 001002 BNE 1$
1209 002646 000167 000436 JMP .BEGIN
1210 002652 012700 000300 15: MOV #300,R0 ;RESTORE VECTOR AREA TO TRAPCATCHER
1211 002656 012701 000302 MOV #302,R1 ;START AT LOCATION 300
1212 002662 012702 000004 MOV #4,R2
1213 002666 010110 25: MOV R1,(R0)
1214 002670 005011 CLR (R1)
1215 002672 060200 ADD R2,R0
1216 002674 060201 ADD R2,R1
1217 002676 022701 001000 CMP #1000,R1 ;END AT LOCATION 776
1218 002702 002771 BLT 2$
1219 002704 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1220 002706 015212 MREGAD ;MESSAGE
1221 002710 104410 PARAM ;CONVERT STRING
1222 002712 160000 ;LOW LIMIT
1223 002714 167776 ;HIGH LIMIT
1224 002716 017124 DUBASE ;STORE AT THIS LOCATION
1225 002720 001 .BYTE 1 ;MASK
1226 002721 001 .BYTE 1 ;HOW MANY TIMES + 2
1227 002722 016767 014176 176226 MOV DUBASE,KEEPADD ;SAVE
1228 002730 004767 014036 JSR PC,DUADDR
1229 002734 016767 176216 176212 MOV KEEPADD,BASEADD ;RESTORE FOR ROTATION

```

1230	002742	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1231	002744	015177				MVECTO	: MESSAGE
1232	002746	104410				PARAM	: CONVERT STRING
1233	002750	000300				300	: LOW LIMIT
1234	002752	000776				776	: HIGH LIMIT
1235	002754	001736				DURIV	: STORE AT THIS LOCATION
1236	002756	001			.BYTE	1	: MASK
1237	002757	004			.BYTE	4	: HOW MANY TIMES + 2
1238	002760	016767	176752	176176		MOV	DURIV,KEEPIV ;SAVE
1239	002766	016767	176744	176166		MOV	DURIV,BASEIV ;SET UP FOR ROTATION
1240	002774	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1241	002776	015242				MMULT	: MESSAGE
1242	003000	104414				SETFLG	: SET FLAG BASED UPON INPUT STRING
1243	003002	001152				MULTD	: THIS FLAG
1244	003004	105767	176142			TSTB	MULTD ;ARE THERE MULTIPLE DEVICES
1245							; ON THE SYSTEM ?
1246	003010	100406				BMI	BBB ;YES,ASK NEXT QUESTION
1247	003012	005067	176150			CLR	ACTREG
1248	003016	005067	176146			CLR	ROTADD
1249	003022	000167	000140			JMP	OUTMUL ;JUMP AROUND NEXT QUESTION
1250	003026				BBB:		
1251	003026	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1252	003030	015271				MLASTD	: MESSAGE
1253	003032	104410				PARAM	: CONVERT STRING
1254	003034	160000				160000	: LOW LIMIT
1255	003036	167776				167776	: HIGH LIMIT
1256	003040	001160				LASTADD	: STORE AT THIS LOCATION
1257	003042	001			.BYTE	1	: MASK
1258	003043	001			.BYTE	1	: HOW MANY TIMES + 2
1259							: THE FOLLOWING ROUTINE SETS UP ACTREG FOR THE FIRST TIME
1260	003044	012767	000001	176116	1\$:	MOV	#1,ROTADD ;SET UP POINTER
1261	003052	005067	176110			CLR	ACTREG ;CLR ACTIVE REGISTER
1262	003056	056767	176106	176102	2\$:	BIS	ROTADD,ACTREG ;MAKE THIS DEVICE ACTIVE
1263	003064	000241				CLC	
1264	003066	006167	176076			ROL	ROTADD ;SET UP POINTER
1265	003072	103421				BCS	3\$;ARE YOU OUT OF RANGE ?
1266	003074	062767	000010	176052		ADD	#10,BASEADJ ;SET UP BASE ADDRESS
1267	003102	026767	176052	176044		LASTADD,BASEADD	: IS THIS THE LAST DEVICE ?
1268	003110	101362				BHI	2\$;NO DO IT AGAIN
1269	003112	056767	176052	176046		BIS	ROTADD,ACTREG ;THIS ASSUMES THAT THERE ARE AT
1270							: LEAST TWO DEVICES WHEN YOU ANSWER YES TO
1271							: MULTIPLE DEVICE QUESTION
1272	003120	012767	000001	176042	4\$:	MOV	#1,ROTADD ;SET UP FOR LATER USE IN END OF PASS ROUTINE
1273	003126	016767	176024	176020		MCV	KEEPADD,BASEADD ;DITTO
1274	003134	000414				BR	OUTMUL ;CONTINUE QUESTIONS
1275	003136	016767	176014	176010	3\$:	MOV	KEEPADD,BASEADD ;RESTORE
1276	003144	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1277	003146	015365				MRANGE	: MESSAGE
1278	003150	104410				PARAM	: CONVERT STRING
1279	003152	160000				160000	: LOW LIMIT
1280	003154	167776				167776	: HIGH LIMIT
1281	003156	001160				LASTADD	: STORE AT THIS LOCATION
1282	003160	001			.BYTE	1	: MASK
1283	003161	001			.BYTE	1	: HOW MANY TIMES + 2
1284	003162	000167	177656			JMP	1\$;DO IT AGAIN
1285	003166	012767	000340	013572	OUTMUL:	MOV	#340,DUPRT

```

1286 003174 004767 013516 JSR PC.DULEV
1287 ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
1288 ;BUFFER TO THE CHARACTERS "1" AND "2".
1289 ;IF THE CHARACTER IS "1" CLEAR THE FLAG
1290 ;IF THE CHARACTER IS "2" SET THE FLAG
1291 003200 AAA:
1292 003200 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1293 003202 015603 MSYNC ;MESSAGE
1294 003204 122767 000061 012732 3$: CMPB #'1,INBUF ;IS IT "1" ?
1295 003212 001003 BNE 1$
1296 003214 105067 175726 CLRB SYNCNO ;000
1297 003220 000412 BR 4$
1298 003222 122767 000062 012714 1$: CMPB #'2,INBUF ;IS IT "2" ?
1299 003230 001004 BNE 2$
1300 003232 112767 177777 175706 MOVB #-1,SYNCNO ;377
1301 003240 000402 BR 4$
1302 003242 104407 2$: INSTER ;RETRY
1303 003244 000757 BR 3$
1304 003246 000240 4$: NOP
1305 003250 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1306 003252 015651 MWIRE6 ;MESSAGE
1307 003254 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1308 003256 001147 SEXMIT ;THIS FLAG
1309 003260 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1310 003262 015722 MWIRES ;MESSAGE
1311 003264 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1312 003266 001150 SEREC ;THIS FLAG
1313 003270 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1314 003272 015772 MWIRE4 ;MESSAGE
1315 003274 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1316 003276 001151 OPTCLR ;THIS FLAG
1317 003300 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1318 003302 016051 MEXTJ ;MESSAGE
1319 003304 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1320 003306 001153 JMRBY ;THIS FLAG
1321
1322 ;TEST START AND RESTART
1323
1324 003310 012706 001100 .BEGIN: MOV #STACK,SP ;SET UP STACK
1325 003314 106427 000340 MTPS #340 ;LOCK OUT INTERRUPTS
1326 003320 032777 000002 176112 BIT #SW01,2SWR ;IF SW01=1, GET STARTING PC
1327 003326 001416 BEQ 3$
1328 003330 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1329 003332 015535 MTSTPC ;MESSAGE
1330 003334 104410 PARAM ;CONVERT STRING
1331 003336 003362 TST1 ;LOW LIMIT
1332 ;HIGH LIMIT
1333 ;STORE AT THIS LOCATION
1334 003340 001 .BYTE 1 ;MASK
1335 003341 001 .BYTE 1 ;HOW MANY TIMES + 2
1336 003342 000403 BR 4$
1337 003344 012767 003362 175532 3$: MOV #TST1,RETURN ;START AT TEST 1
1338 003352 104401 015531 4$: TYPE MR ;TYPE R
1339 003356 000177 175522 JMP 2RETURN ;START TESTING
1340
1341

```

```

1342      :: THIS TEST VERIFYS THAT RXDONE CAUSES AN INTERRUPT
1343      :: MODE: SYNC EXTERNAL
1344      :: INTERRUPT VECTOR: DURIV
1345      :: LENGTH: EIGHT
1346      :: *****
1347 003362 000004      TSTI: SCOPE
1348
1349 003364 052777 000400 176334      BIS      #MRESET,@TXCSR ; MASTER RESET
1350 003372 012777 020000 176322      MOV      #SYNEXT,@PARCSR ; SET THE MODE
1351 003400 052777 000400 176320      BIS      #MRESET,@TXCSR ; MASTER RESET
1352
1353      ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1354 00340E 012777 064001 176312      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1355
1356      ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1357 003414 012777 026026 176300      MOV      #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1358 003422 052777 000020 176262      BIS      #SYNSCH,@RXCSR ; SET SEARCH SYNC
1359      ; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1360 003430 042777 020000 176270      BIC      #CLK,@TXCSR ; POKE CLK DOWN
1361 003436 052777 020000 176262      BIS      #CLK,@TXCSR ; POKE CLK UP
1362 003444 012777 003466 176264      MOV      #15,@DURIV ; SET UP TRAPCATCHER
1363 003452 016777 013310 176260      MOV      DUPRT,@DURIS
1364 003460 106427 000000
1365 003464 000424
1366      ; THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1367 003466 106427 000340      IS: MTPS #340 ; DON'T ALLOW ANYMORE INTERRUPTS
1368 003472 042777 000100 176212      BIC      #RINTEN,@RXCSR ; CLEAR INTERRUPT ENABLE
1369 003500 105777 176206      TSTB    @RXCSR ; RXDONE=1?
1370 003504 100401
1371 003506 104004
1372 003510 012716 003702      MOV      #35(SP) ; SET UP RETURN LOCATION
1373 003514 016777 176220 176214      MOV      DURIS,@DURIV ; RESTORE TRAPCATCHER
1374 003522 012777 000000 176210      MOV      #0,@DURIS
1375 003530 017701 176162      MOV      @RXDBUF,R1 ; CLEAR INTERRUPT
1376 003534 000002      RTI
1377
1378 003536 052777 000100 176146      25: BIS      #RINTEN,@RXCSR ; SET INTERRUPT ENABLE
1379 003544 012767 000010 175350      MOV      #8,SHIFT ; # OF SHIFTS
1380 003552 012767 000025 175720      MOV      #25,$TMP1 ; TO BE SHIFTED CHARACTER
1381      ; THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
1382      ; INFORMATION CONTAINED IN $TMP1 AND IT IS
1383      ; SHIFTED IN BY THE CONTENTS OF SHIFT
1384 003560 042777 040000 176140      55: BIC      #MTDATA,@TXCSR
1385 003566 000241
1386 003570 006067 175704      ROR      $TMP1 ; FORCE CARRY
1387 003574 103003
1388 003576 052777 040000 176122      BCC     45
1389 003604 042777 020000 176114      45: BIC      #CLK,@TXCSR
1390 003612 052777 020000 176106      BIS      #CLK,@TXCSR
1391 003620 005367 175276      DEC     SHIFT
1392 003624 001355      BNE     55
1393      ; INTERRUPT SHOULD NOW OCCUR
1394 003626 005000      C_R    RD
1395 003630 005200      INC     RD ; WAIT FOR INTERRUPT
1396 003632 001376      BNE     -2
1397 003634 016777 176100 176074      MOV      DURIS,@DURIV ; RESTORE TRAPCATCHER
    
```

F03

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 33
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1398 003642 012777 000000 176070      MOV      #0, @DURIS      ;
1399 003650 016703 176042      MOV      RXDBUF, R3      ; FOR ERROR MESSAGE
1400 003654 012700 000025      MOV      #25, R0        ; EXPECTED
1401 003660 017701 176032      MOV      @RXDBUF, R1
1402 003664 042777 000100 176020      BIC      @RINTEN, @RXCSR ; CLR INTR ENABLE
1403 003672 020001      CMP      R0, R1
1404 003674 001401      BEQ      .+4
1405 003676 104002      ERROR   2              ; CHARACTERS SHOULD COMPARE
1406 003700 104004      ERROR   4              ; INTERRUPT FAILED TO OCCUR
1407
1408 003702 106427 000340      3$:      MTPS      #340
1409
1410
1411
1412      ;: THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP TO
1413      ;: THE SAME VECTOR ARE BOTH EXECUTED
1414      ;: INTERRUPT VECTOR: DURIV
1415      ;: THIS TEST ONLY WORKS IN MAINT EXTERNAL MODE
1416
1417      ;: *****
1418 003706 000004      1$:      SCOPE
1419 003710 105767 175237      TSTB     JMRBY          ; IN MAINT. EXTERNAL?
1420 003714 100402      BMI     .+6            ; IF ANSWER WAS YES SO THIS TEST
1421 003716 000167 000402      JMP     IS             ; IF ANSWER WAS NO JUMP AROUND TEST
1422 003722 052777 000400 175776      BIS     @MRESET, @TXCSR ; MASTER RESET
1423 003730 012777 020000 175764      MOV     @SYNEXT, @PARCSR ; SET THE MODE
1424 003736 052777 000400 175762      BIS     @MRESET, @TXCSR ; MASTER RESET
1425
1426      ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1427 003744 012777 064001 175754      MOV     @MTDATA!CLK!MINT!BREAK, @TXCSR
1428
1429      ; SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
1430 003752 012777 026026 175742      MOV     @SYNEXT!EIGHT!NOPAR!26, @PARCSR
1431 003760 052777 000020 175724      BIS     @SYNSCH, @RXCSR ; SET SEARCH SYNC
1432      ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1433 003766 042777 020000 175732      BIC     @CLK, @TXCSR    ; POKE CLK DOWN
1434 003774 052777 020000 175724      BIS     @CLK, @TXCSR    ; POKE CLK UP
1435 004002 012777 004024 175726      MOV     #25, @DURIV     ; SET UP TRAPCATCHER
1436 004010 016777 012752 175722      MOV     DUPRT, @DURIS
1437 004016 106427 000000      MTPS   #0              ; ALLOW INTERRUPT
1438 004022 000454      BR      3$             ; JUMP AROUND SVC ROUTINE
1439
1440      ; THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
1441 004024 106427 000340      2$:      MTPS      #340      ; DON'T ALLOW ANY MORE INTERRUPTS
1442 004030 105777 175656      TSTB     @RXCSR          ; RXDONE = 1 ?
1443 004034 100401      BMI     .+4
1444 004036 104004      ERROR   4              ; FALSE INTERRUPT
1445 004040 012716 004320      MOV     #55, (SP)       ; SET UP RETURN LOCATION
1446 004044 012777 004124 175664      MOV     #45, @DURIV     ; SET UP TRAPCATCHER FOR SECOND
1447      ; INTERRUPT
1448 004052 052777 000002 175632      BIS     @DTR, @RXCSR    ; TRY TO CAUSE SECOND INTERRUPT
1449 004060 017701 175632      MOV     @RXDBUF, R1     ; JUST READ RXDBUF TO CLR RXDONE
1450      ; TO ALLOW SECOND INTERRUPT
1451 004064 106427 000000      MTPS   #0              ; ALLOW INTERRUPT
1452 004070 005000      CLR     R0
1453 004072 005200      INC     R0              ; WAIT FOR INTERRUPT

```



```

1454 004074 001376      BNE      -2
1455 004076 042777 000140 175606      BIC      #RINTEN!DSINTE, @RXCSR ;CLR INTR ENABLES
1456 004104 104004      ERROR    4 ;2ND INTERRUPT FAILED TO OCCUR
1457
1458 004106 016777 175626 175622 6$:      MOV      DURIS, @DURIV ;RESTORE TRAPCATCHER
1459 004114 012777 000000 175616      MOV      #0, @DURIS ;
1460 004122 000002      RTI
1461
1462 ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1463 004124 106427 000340 4$:      MTPS     #340 ;DON'T ALLOW ANYMORE INTERRUPTS
1464 004130 005777 175556      TST      @RXCSR ;DSC = 1 ?
1465 004134 100401      BMI      +4
1466 004136 104004      ERROR    4 ;FALSE INTERRUPT
1467 004140 042777 000140 175544      BIC      #RINTEN!DSINTE, @RXCSR ;CLR BOTH INTR ENABLES
1468 004146 012716 004106      MOV      #6$, (SP) ;SET UP RETURN LOCATION
1469 004152 000002      RTI
1470
1471 004154 052777 000140 175530 3$:      BIS      #RINTEN!DSINTE, @RXCSR ;SET INTERRUPT ENABLES
1472 004162 012767 000010 174732      MOV      #8, SHIFT ;# OF SHIFTS
1473 004170 012767 000025 175302      MOV      #25, $TMP1
1474
1475 ;THE FOLLOWING POKES THE MAINT DATA BASED UPON THE
1476 ;INFORMATION CONTAINED IN $TMP1 AND IT IS
1477 ;SHIFTED IN BY THE CONTENTS OF SHIFT
1478 004176 042777 040000 175522 8$:      BIC      #MTDATA, @TXCSR
1479 004204 000241      CLC
1480 004206 006067 175266      ROR      $TMP1 ;FORCE CARRY
1481 004212 103003      BCC      7$
1482 004214 052777 040000 175504      BIS      #MTDATA, @TXCSR
1483 004222 042777 020000 175476 7$:      BIC      #CLK, @TXCSR
1484 004230 052777 020000 175470      BIS      #CLK, @TXCSR
1485 004236 005367 174660      DEC      SHIFT
1486 004242 001355      BNE      8$
1487 ;1ST INTERRUPT SHOULD NOW OCCUR
1488 004244 005000      CLR      RO
1489 004246 005200      INC      RO ;WAIT FOR INTERRUPT
1490 004250 001376      BNE      -2
1491 004252 016777 175462 175456      MOV      DURIS, @DURIV ;RESTORE TRAPCATCHER
1492 004260 012777 000000 175452      MOV      #0, @DURIS ;
1493 004266 016703 175424      MOV      @RX0BUF, R3 ;FOR ERROR MESSAGE
1494 004272 012700 000025      MOV      #25, RO ;EXPECTED
1495 004276 017701 175414      MOV      @RX0BUF, R1
1496 004302 042777 000140 175402      BIC      #RINTEN!DSINTE, @RXCSR ;CLR INTERRUPT ENABLES
1497 004310 020001      CMP      RO, R1
1498 004312 001401      BEQ      +4
1499 004314 104002      ERROR    2 ;CHARACTERS SHOULD COMPARE
1500 004316 104004      ERROR    4 ;INTERRUPT FAILED TO OCCUR
1501
1502 004320 106427 000340 5$:      MTPS     #340 ;DON'T ALLOW ANY MORE INTERRUPTS
1503 004324
1504
1505 ;: THIS TEST VERIFYS THAT DNA CAUSES AN INTERRUPT
1506 ;: MODE: SYNC EXTERNAL
1507 ;: INTERRUPT VECTOR: DUTIV
1508 ;: *****
1509 004324 000004 1$T3:      SCOPE
    
```

H03

DZDUV-A MACY11 27(1006) 03-FEB-77 09:30 PAGE 35
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1510
1511 004326 052777 000400 175372      BIS      #MRESET,@TXCSR  ;MASTER RESET
1512 004334 012777 020000 175360      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1513 004342 052777 000400 175356      BIS      #MRESET,@TXCSR  ;MASTER RESET
1514
1515      ;SET MAINTENANCE MODE & SEND
1516      ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1517 004350 012777 004020 175350      MOV      #MINT!SEND,@TXCSR
1518
1519      ;SET MODE # OF BITS,PARITY SENSE & LOAD SYNC REG
1520 004356 012777 026026 175336      MOV      #SYNEXT!EIGHT!NOPAR!26,@PARCSR
1521 004364 112777 000025 175340      MOV      #25,@TXDBUF    ;LOAD CHARACTER
1522 004372 012767 000010 174522      MOV      #8,SHIFT
1523      ;POKE CLK TO GET INTO SYNCRONIZATION
1524 004400 052777 020000 175320      BIS      #CLK,@TXCSR    ;POKE CLK UP
1525 004406 042777 020000 175312      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
1526
1527      1$:
1528 004414 052777 020000 175304      BIS      #CLK,@TXCSR    ;POKE CLK UP
1529 004422 042777 020000 175276      BIC      #CLK,@TXCSR    ;POKE CLK DOWN
1530 004430 005367 174466      DEC      SHIFT          ;LAST SHIFT?
1531 004434 001367      BNE      1$
1532 004436 012777 004504 175276      MOV      #2$,@DUTIV     ;SET UP TRAPCATCHER
1533 004444 016777 012316 175272      MOV      DUPRT,@DUTIS
1534 004452 106427 000000      MTPS     #0              ;ALLOW INTERRUPTS
1535 004456 052777 000040 175242      BIS      #DNAINTE,@TXCSR ;ENABLE INTERRUPT
1536      ;NOW POKE CLK TO GET DNA
1537 004464 052777 020000 175234      BIS      #CLK,@TXCSR    ;POKE CLK
1538 004472 005000      CLR      R0
1539 004474 005200      INC      R0              ;WAIT FOR INTERRUPT
1540 004476 001376      BNE      -2
1541 004500 104004      ERROR   4              ;INTERRUPT FAILED TO OCCUR
1542 004502 000422      BR      3$              ;JUMP AROUND SVC ROUTINE
1543      ;THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1544 004504 106427 000340 3$:      MTPS     #340           ;DON'T ALLOW ANYMORE INTERRUPTS
1545 004510 005777 175212      TST      @TXCSR         ;DNA?
1546 004514 100401      BMI     +4
1547 004516 104004      ERROR   4              ;FALSE INTERRUPT
1548 004520 042777 000040 175200      BIC      #DNAINTE,@TXCSR ;CLR INTR ENABLE
1549 004526 012716 004572      MOV      #4$(SP)        ;SET UP RETURN LOCATION
1550 004532 016777 175206 175202      MOV      DUTIS,@DUTIV   ;RESTORE TRAPCATCHER
1551 004540 012777 000000 175176      MOV      #0,@DUTIS
1552 004546 000002      RTI
1553
1554 004550 016777 175170 175164 3$:      MOV      DUTIS,@DUTIV   ;RESTORE TRAPCATCHER
1555 004556 012777 000000 175160      MOV      #0,@DUTIS
1556
1557 004564 042777 000040 175134 4$:      BIC      #DNAINTE,@TXCSR ;CLR INTERRUPT ENABLE
1558 004572 106427 000340      MTPS     #340           ;RESTORE NO INTERRUPT STATUS
1559
1560      ;: THIS TEST VERIFYS THAT TXDONE CAUSES AN INTERRUPT
1561      ;: INTERRUPT VECTOR: DUTIV
1562      ;: NOTE: TXDONE = 1 AFTER A MASTER RESET
1563      ;:
1564      ;: *****
1565

```

```

1566 004576 000004          TST4:  SCOPE
1567
1568 004600 052777 000400 175120      BIS      #MRESET, @TXCSR  ; MASTER RESET
1569 004606 012777 004654 175126      MOV      #15, @DUTIV   ; SET UP TRAPCATCHER
1570 004614 016777 012146 175122      MOV      DUPRT, @DUTIS ;
1571 004622 106427 000000          MTPS     #0           ; ALLOW INTERPUTS
1572 004626 052777 000100 175072      BIS      #TXINTE, @TXCSR ; ENABLE INTERRUPT
1573 004634 005000          CLR      R0           ;
1574 004636 005200          INC      R0           ; WAIT FOR INTERRUPT
1575 004640 001376          BNE     .-2           ;
1576 004642 042777 000100 175056      BIC      #TXINTE, @TXCSR ; CLR INTERRUPT ENABLE
1577 004650 104004          ERPOR   4           ; INTERRUPT FAILED TO OCCUR
1578 004652 000422          BR      2$           ; JUMP AROUND SVC ROUTINE
1579
1580          ; THE FOLLOWING IS THE INTERRUPT SERVICE ROUTINE
1581 004654 106427 000340          1$:     MTPS     #340    ; DON'T ALLOW ANYMORE INTERRUPTS
1582 004660 042777 000100 175040      BIC      #TXINTE, @TXCSR ; CLR INTR ENABLE
1583 004666 105777 175034          TSTB    @TXCSR       ; TXDONE?
1584 004672 100401          BMI     .+4          ;
1585 004674 104004          ERROR  4           ; FALSE INTERRUPT
1586 004676 012716 004734          MOV     #3$, (SP)    ; SET UP RETURN LOCATION
1587 004702 016777 175036 175032      MOV     DUTIS, @DUTIV ; RESTORE TRAPCATCHER
1588 004710 012777 000000 175026      MOV     #0, @DUTIS   ;
1589 004716 000002          RTI
1590
1591 004720 016777 175020 175014      2$:     MOV     DUTIS, @DUTIV ; RESTORE TRAPCATCHER
1592 004726 012777 000000 175010      MOV     #0, @DUTIS   ;
1593
1594 004734 106427 000340          3$:     MTPS     #340    ; RESTORE NO INTERRUPT STATUS
1595
1596
1597
1598          ;; THIS TEST VERIFYS THAT TXDONE DOES NOT CAUSE AN INTERRUPT
1599          ;; WHEN PROCESSOR PRIORITY LEVEL IS TOO HIGH
1600          ;; INTERRUPT VECTOR: DUTIV
1601          ;; NOTE: TXDONE = 1 AFTER A MASTER RESET
1602          ;;
1603          ;*****
1604 004740 000004          TST5:  SCOPE
1605 004742 052777 000400 174756      BIS      #MRESET, @TXCSR ; MASTER RESET
1606 004750 012777 005034 174764      MOV      #15, @DUTIV   ; SET UP TRAPCATCHER
1607 004756 016777 012004 174760      MOV      DUPRT, @DUTIS ;
1608 004764 106427 000340          MTPS     #340        ; SET PS LEVEL TOO HIGH
1609 004770 052777 000100 174730      BIS      #TXINTE, @TXCSR ; ENABLE INTERRUPT
1610 004776 005000          CLR      R0           ; WAIT FOR INTERRUPT
1611 005000 005200          INC      R0           ;
1612 005002 001376          BNE     .-2           ;
1613 005004 042777 000100 174714      BIC      #TXINTE, @TXCSR ; CLR INTR ENABLE
1614 005012 106427 000340          MTPS     #340        ; DON'T ALLOW INTERRUPTS
1615 005016 016777 174722 174716      MOV     DUTIS, @DUTIV ; RESTORE TRAPCATCHER
1616 005024 012777 000000 174712      MOV     #0, @DUTIS   ;
1617 005032 000421          BR      2$           ; TEST IS OK...GET OUT OF TEST
1618          ; THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1619 005034 106427 000340          1$:     MTPS     #340    ; DONT ALLOW ANYMORE INTERRUPTS
1620 005040 042777 000100 174660      BIC      #TXINTE, @TXCSR ; CLR INTR ENABLE
1621 005046 012716 005070          MOV     #3$, (SP)    ; SET UP RETURN LOCATION
  
```

```

1622
1623 005052 016777 174666 174662      MOV    DUTIS, DUTIV      ; TO REPORT ERROR
1624 005060 012777 000000 174656      MOV    #0, DUTIS        ; RESTORE TRAPCATCHER
1625 005066 000002                                RTI
1626                                ; END OF INTERRUPT SVC ROUTINE
1627
1628
1629
1630                                ; YOU SHOULD NOT GET INTO THIS FOLLOWING CODE UNLESS THERE
1631                                ; WAS AN ERROR
1632 005070 106427 000340      3$:   MTPS    #340        ; DON'T ALLOW ANYMORE INTERRUPTS
1633 005074 104004                                ERROR    4                ; INTERRUPT SHOULD NOT OF OCCURED, CHECK
1634                                ; THE INTERRUPT LEVEL SELECTED OR CHECK
1635                                ; INTERRUPT LOGIC OR BOTH
1636 005076      2$:
1637
1638                                ; THIS TEST VERIFYS THAT TXDONE CAUSES ONLY ONE INTERRUPT
1639                                ; PROVIDING THAT TXCSR IS NOT READ
1640                                ; AND TXDBUF IS NOT LOADED (WRITTEN)
1641                                ; THIS TEST CHECKS THE ONCE ONLY FLIP/FLOP (V2)
1642                                ; OF THE INTERRUPT CONTROL LOGIC
1643                                ; INTERRUPT VECTOR: DUTIV
1644                                ; NOTE: TXDONE = 1 AFTER A MASTER RESET
1645
1646                                ; *****
1647 005076 000004      *ST6:  SCOPE
1648 005100 052777 000400 174620      BIS    #MRESET, TXCSR   ; MASTER RESET
1649 005106 012777 005146 174626      MOV    #15, DUTIV       ; SET UP TRAPCATCHER
1650 005114 016777 011646 174622      MOV    DUPRT, DUTIS
1651 005122 106427 000000      MTPS   #0                ; ALLOW INTERRUPTS
1652 005126 052777 000100 174572      BIS    #TXINTE, TXCSR   ; ENABLE INTR ENABLE
1653 005134 005000      CLR    RO
1654 005136 005200      INC    RO
1655 005140 001376      BNE    .-2
1656 005142 104004      ERROR  4                ; INTERRUPT FAILED TO OCCUR
1657 005144 000425      BR     4$
1658                                ; THE FOLLOWING IS THE INTR SVC ROUTINE
1659 005146 106427 000340      1$:   MTPS    #340        ; DON'T ALLOW ANYMORE INTR
1660 005152 012716 005212      MOV    #3$, (SP)        ; SET UP RETURN LOCATION
1661 00515E 012777 005166 174556      MOV    #2$, DUTIV       ; SET UP TRAPCATCHER TO
1662                                ; PROVE THAT THE INTERRUPT DOES NOT OCCUR
1663                                ; TWICE (AFTER RTI 'ING FROM THIS
1664                                ; SVC ROUTINE
1665 005164 000002                                RTI
1666                                ; THE FOLLOWING INTERRUPT SVC ROUTINE WILL CATCH THE SECOND INTR
1667 005166 106427 000340      2$:   MTPS    #340        ; DON'T ALLOW INTER
1668 005172 012716 005220      MOV    #4$, (SP)        ; SET UP RETURN LOCATION
1669 005176 105777 174524      TSTB  TXCSR             ; TXDONE = 1?
1670 005202 100401      BMI    .+4
1671 005204 104004      ERROR  4                ; TXDONE SHOULD BE SET
1672 005206 104004      ERROR  4                ; THE INTERRUPT WAS TAKEN TWICE.....
1673                                ; CHECK OUT THE V2 FLIP/FLOP LOGIC
1674                                ; IN THE INTERRUPT CONTROL LOGIC
1675 005210 000002                                RTI
1676 005212 005000      3$:   CLR    RO                ; ALLOW TIME TO CATCH SECOND
1677 005214 005200      INC    RO                ; IF IT WERE TO OCCUR
    
```

K03

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 38
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1678 005216 001376          BNE      -2
1679 005220 016777 174520 174514 4$:  MOV     @DUTIS,@DUTIV ;RESTORE TRAPCATCHER
1680 005226 012777 000000 174510      MOV     #0,@DUTIS
1681 005234 042777 000100 174464      BIC     @TXINTE,@TXCSR ;CLR INTERRUPT ENABLE
1682 005242 106427 000340      MTPS    #340 ;RESTORE NO INTERRUPT STATUS
1683
1684
1685 ;:THIS TEST VERIFYS THAT TWO INTERRUPTS THAT TRAP
1686 ;:TO THE SAME VECTC ARE BOTH EXECUTED
1687 ;:INTERRUPT VECTOR: DUTIV
1688 ;:MODE: SY' EXTERNAL
1689
1690 ;:*****
1691 005246 000004      †ST7:  SCOPE
1692
1693 005250 052777 000400 174450      BIS     @MRESET,@TXCSR ;MASTER RESET
1694 005256 012777 020000 174436      MOV     @SYNEXT,@PARCSR ;SET THE MODE
1695 005264 052777 000400 174434      BIS     @MRESET,@TXCSR ;MASTER RESET
1696
1697 ;SET MAINTENANCE MODE & SEND
1698 ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
1699 005272 012777 004020 174426      MOV     @MINT!SEND,@TXCSR
1700
1701 ;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
1702 005300 012777 026026 174414      MOV     @SYNEXT!EIGHT!NOPAR!26,@PARCSR
1703 005306 112777 000025 174416      MOV     #25,@TXDBUF ;LOAD CHARACTER
1704 005314 012767 000010 173600      MOV     #8,SHIFT
1705 ;POKE CLK TO GET INTO SYNCHRONIZATION
1706 005322 052777 020000 174376      BIS     @CLK,@TXCSR ;POKE CLK UP
1707 005330 042777 020000 174370      BIC     @CLK,@TXCSR ;POKE CLK DOWN
1708
1709 005336
1710 005336 052777 020000 174362      1$:  BIS     @CLK,@TXCSR ;POKE CLK UP
1711 005344 042777 020000 174354      BIC     @CLK,@TXCSR ;POKE CLK DOWN
1712 005352 005367 173544      DEC     SHIFT ;LAST SHIFT?
1713 005356 001367          BNE     1$
1714 005360 012777 005420 174354      MOV     @2$,@DUTIV ;SET UP TRAPCATCHER
1715 005366 016777 011374 174350      MOV     @DUPAT,@DUTIS
1716 005374 106427 000000          MTPS    #0 ;ALLOW INTERRUPTS
1717 005400 052777 000140 174320      BIS     @TXINTE!DNAINTE,@TXCSR ;ENABLE INTERRUPTS
1718 005406 005000          CLR     RO
1719 005410 005200          INC     RO ;WAIT FOR INTERRUPT
1720 005412 001376          BNE     -2
1721 005414 104004          ERROR  4 ;INTERRUPT FAILED TO OCCUR
1722 005416 000461          BR     3$ ;JUMP AROUND SVC ROUTINES
1723
1724 ;THE FOLLOWING IS THE 1ST INTERRUPT SVC ROUTINE
1725 005420 106427 000340      2$:  MTPS    #340 ;DON'T ALLOW ANYMORE INTERRUPTS
1726 005424 005777 174276      TST     @TXCSR ;DNA=0 ?
1727 005430 100001          BPL     .+4
1728 005432 104004          ERROR  4 ;DNA SHOULD NOT BE ASSERTED
1729 005434 105777 174266      TSTB   @TXCSR ;TXDONE = 1?
1730 005440 100401          BMI     .+4
1731 005442 104004          ERROR  4 ;FALSE INTERRUPT
1732 005444 012716 005604      MOV     #4$(,SP) ;SET UP RETURN LOCATION
1733 005450 012777 005532 174264      MOV     #5$,@DUTIV ;SET UP TRAPCATCHER
  
```

```

1734                                     ;NOW POKE CLK TO BRING UP DNA
1735 005456 052777 020000 174242      BIS      #CLK,@TXCSR      ;POKE CLK
1736 005464 112777 000025 174240      MOV      #25,@TXDBUF    ;JUST LOAD ANY CHAR TO CLR
1737                                     ;TXDONE TO ALLOW SECOND INTERRUPT
1738 005472 106427 000000              MTPS     #0             ;ALLOW INTERRUPTS
1739 005476 005000                      CLR      RO
1740 005500 005200                      INC      RO             ;WAIT FOR INTERRUPT
1741 005502 001376                      BNE     .-2
1742 005504 042777 000140 174214      BIC      #DNAINTE!TXINTE,@TXCSR ;CLR INTR ENABLES
1743 005512 104004                      ERROR   4              ;2ND INTERRUPT FAILED TO OCCUR
1744
1745 005514 016777 174224 174220 6$:   MOV      DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1746 005522 012777 000000 174214      MOV      #0,@DUTIS     ;
1747 005530 000002                      RTI
1748
1749                                     ;THE FOLLOWING IS THE 2ND INTERRUPT SVC ROUTINE
1750 005532 106427 000340 5$:         MTPS     #340
1751 005536 005777 174164              TST      @TXCSR        ;DNA
1752 005542 100401                      BMI     .+4
1753 005544 104004                      ERROR   4              ;FALSE INTERRUPT
1754 005546 042777 000140 174152      BIC      #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTR ENABLES
1755 005554 012716 005514              MOV      #6$, (SP)    ;SETUP RETURN LOCATION
1756 005560 000002                      RTI
1757
1758 005562 016777 174156 174152 3$:   MOV      DUTIS,@DUTIV    ;RESTORE TRAPCATCHER
1759 005570 012777 000000 174146      MOV      #0,@DUTIS
1760
1761 005576 042777 000140 174122      BIC      #DNAINTE!TXINTE,@TXCSR ;CLR BOTH INTERRUPT
1762                                     ;ENABLES
1763 005604 106427 000340 4$:         MTPS     #340          ;RESTORE NO INTERRUPT STATUS
1764
1765
1766
1767                                     ;: THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1768                                     ;: IT BASICALLY CHECKS THE EXISTANCE OF
1769                                     ;: THE FREE RUNNING OSCILLATOR
1770                                     ;: MODE: SYNEXT
1771                                     ;: LENGTH: EIGHT
1772                                     ;: THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1773
1774                                     ;: *****
1775 005610 000004 7ST10: SCOPE
1776                                     ;: *****
1777 005612 000167 000262              JMP      1$            ;NOP THIS TEST
1778                                     ;: *****
1779 005616 052777 000400 174102      BIS      #MRESET,@TXCSR ;MASTER RESET
1780 005624 012777 020000 174070      MOV      #SYNEXT,@PARCSR ;LOAD THE MODE
1781 005632 052777 000400 174066      BIS      #MRESET,@TXCSR ;MASTER RESET
1782 005640 012777 026026 174054      MOV      #SYNEXT!EIGHT!NOPAR!26,@PARCSR ;LOAD THE MODE.
1783                                     ;: # OF BITS PER CHAR, PARITY SENSE(NO PARITY),
1784                                     ;: & SYNC CHARACTER (26)
1785 005646 112777 000025 174056      MOV      #25,@TXDBUF   ;LOAD THE CHAR
1786 005654 012777 005752 174054      MOV      #25,@DURIV    ;SET UP TRAPCATCHER
1787 005662 016777 011100 174050      MOV      DUPRT,@DURIS
1788 005670 106427 000000              MTPS     #0             ;ALLOW INTERRUPTS
1789 005674 016703 174016      MOV      RXDBUF,R3     ;SET UP FOR ERROR MESSAGE
    
```

M03

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 40
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1790 005700 012700 000025      MOV      #25,R0      ;EXPECTED
1791 005704 012777 014020 174014  MOV      #SYSTST!SEND,@TXCSR ;OK NOW LOAD SEND &
1792                                ;MAINT. MODE
1793 005712 052777 000127 173772  BIS      #SYNSCH!RINTEN,@RXCSR ;SET SEARCH SYNC &
1794                                ;RECEIVER INTERRUPT
1795                                ;ENABLE & WAIT FOR INTERRUPT
1796 005720 005067 173564          CLR      $TMP5
1797 005724 005002          3$: CLR      R2
1798 005726 005202          INC      R2      ;WAIT FOR INTERRUPT
1799 005730 001376          BNE     .-2
1800 005732 005267 173552          INC      $TMP5
1801 005736 022767 000003 173544  CMP      #3,$TMP5
1802 005744 002367          BGE     3$
1803 005746 104004          ERROR   4      ;INTERRUPT DID NOT OCCUR
1804 005750 000422          BR      4$
1805
1806          ;THE FOLLOWING IS THE INTERRUPT SVC ROUTINE
1807 005752 106427 000340          2$: MTPS   #340 ;PREVENT INTERRUPTS
1808 005756 017704 173730          MOV     @RXCSR,R4 ;SAVE
1809 005762 017701 173730          MOV     @RXDBUF,R1 ;ACTUAL
1810 005766 016777 173746 173742  MOV     @DURIS,@DURIV ;RESTORE TRAPCATCHER
1811 005774 012777 000000 173736  MOV     #0,@DURIS
1812 006002 012716 006046          MOV     #5,(SP) ;SET UP RETURN
1813 006006 042777 000100 173676  BIC     #RINTEN,@RXCSR ;CLR INTERRUPT ENABLE
1814 006014 000002          RTI
1815
1816 006016 042777 000100 173666          4$: BIC     #RINTEN,@RXCSR ;CLR INTERRUPT ENABLE
1817 006024 106427 000340          MTPS   #340 ;PREVENT INTERRUPTS
1818 006030 016777 173704 173700  MOV     @DURIS,@DURIV ;RESTORE TRAPCATCHER
1819 006036 012777 000000 173674  MOV     #0,@DURIS
1820 006044 000415          BR      1$
1821
1822 006046 020001          5$: CMP     R0,R1
1823 006050 001401          BEQ     .+4
1824 006052 104002          ERROR   2      ;CHARACTERS DID NOT MATCH
1825 006054 016703 173632          MOV     @RXCSR,R3 ;SETUP FOR ERROR MESSAGE
1826 006060 012700 000200          MOV     #200,R0 ;EXPECTED
1827 006064 010401          MOV     R4,R1 ;ACTUAL
1828 006066 042701 177577          BIC     #177577,R1 ;SAVE ONLY RXDONE
1829 006072 020001          CMP     R0,R1
1830 006074 001401          BEQ     .+4
1831 006076 104001          ERROR   1      ;FALSE INTERRUPT
1832
1833 006100          1$:
1834          ;THIS TEST VERIFYS CTP MODE (IE SYSTST MODE)
1835          ;IT BASICALLY CHECKS THE EXISTANCE OF
1836          ;THE FREE RUNNING OSCILLATOR
1837          ;MODE: SYNINT
1838          ;LENGTH: EIGHT
1839          ;THIS TEST USES BOTH THE RECEIVER & TRANSMITTER LOGIC
1840          ;
1841          ;*****
1842 006100 000004          ST11: SCOPE
1843 006102 052777 000400 173616  BIS     #MRESET,@TXCSR ;MASTER RESET
1844 006110 012777 030000 173604  MOV     #SYNINT,@PARCSR ;SET THE MODE
1845 006116 052777 000400 173602  BIS     #MRESET,@TXCSR ;MASTER RESET

```

N03

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 41
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

1846
1847
1848
1849 006124 012777 014020 173574
1850
1851
1852 006132 012777 036026 173562
1853 006140 052777 000420 173544
1854
1855
1856
1857
1858
1859 006146 005067 173336
1860 006152 005002
1861 006154 005202
1862 006156 001376
1863 006160 005267 173324
1864 006164 022767 000003 173315
1865 006172 002367
1866 006174 012777 006410 173534
1867 006202 016777 010560 173530
1868 006210 012777 006502 173524
1869 006216 016777 010544 173520
1870 006224 106427 000000
1871 006230 016703 173462
1872 006234 012700 000025
1873 006240 012767 000002 172656
1874
1875 006246 105767 172674
1876 006252 100402
1877 006254 005367 172644
1878 006260 052777 000100 173424 9$:
1879 006266 052777 000100 173432
1880 006274 000167 000012
1881
1882 006300 112777 000026 173424 1$:
1883 006306 005067 173176
1884 006312 005002 8$:
1885 006314 005202
1886 006316 001376
1887 006320 005267 173164
1888 006324 022767 000003 173156
1889 006332 002367
1890 006334 106427 000340
1891 006340 042777 000100 173360
1892 006346 042777 000100 173336
1893 006354 016777 173360 173354
1894 006362 012777 000000 173350
1895 006370 016777 173350 173344
1896 006376 012777 000000 173340
1897 006404 104004
1898 006406 000540
1899
1900
1901 006410 106427 000340
  
```

```

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

;SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
MOV #SYNINT!EIGHT!NOPAR!26,@PARCSR
BIS #SYNSCH!STPSYN,@RXCSR ;SET SEARCH SYNC &
;STRIP SYNC SO THAT RXDONE ASSERTS
;WHEN CHAR "25" ARRIVES AND NOT BEFORE...
;...THEREFORE, SET STRIP SYNC
;WAIT FOR SYNSCH TO BE
;CLOCKED IN BY SYSTST CLK
CLR $TMP5
CLR R2
INC R2 ;WAIT
BNE #-2
INC $TMP5
CMP #3,$TMP5
BGE #-20 ;GO BACK TO CLR R2 AND WAIT SOME MORE
MOV #25,@DURIV ;SET UP TRAPCATCHER
MOV DUPRT,@DURIS
MOV #35,@DUTIV
MOV DUPRT,@DUTIS
MTPS #0 ;ALLOW INTERRUPTS
MOV RXDBUF,R3 ;SET UP FOR ERROR MSG
MOV #25,R0 ;EXPECTED CHAR
MOV #2,COUNT ;# OF SYNC CHARS TO GET INTO
;SYNCHRONIZATION
TSTB SYNCNO ;TEST TO SEE HOW MANY SYNC CHARS NEEDED
BMI 9$
DEC COUNT ;MAKE IT ONE LESS
BIS #RINTEN,@RXCSR ;SET INTERRUPT ENABLES
BIS #TXINTE,@TXCSR
JMP 8$ ;THE FIRST XMIT INTERRUPT SHOULD COME
;FROM TXDONE = 1 AFTER A MASTER RESET
MOV #26,@TXDBUF ;LOAD SYNC CHAR
CLR $TMP5
CLR R2 ;WAIT FOR INTERRUPT
INC R2
BNE #-2
INC $TMP5
CMP #3,$TMP5
BGE 8$
MTPS #340 ;PREVENT INTERRUPTS
BIC #TXINTE,@TXCSR ;CLR INTR ENABLES
BIC #RINTEN,@RXCSR
MOV DURIS,@DURIV ;RESTORE TRAPCATCHER
MOV #0,@DURIS
MOV DUTIS,@DUTIV
MOV #0,@DUTIS
ERROR 4 ;TXDONE INTERRUPT FAILED TO OCCUR
BR 7$ ;GET OUT OF THE TEST

;THE FOLLOWING IS THE RECEIVER INTERRUPT SVC ROUTINE
2$: MTPS #340 ;PREVENT INTERRUPTS
  
```



```

1902 006414 017704 173272      MOV      @RXCSR,R4      ;SAVE
1903 006420 017701 173272      MOV      @RXDBUF,R1    ;ACTUAL
1904 006424 016777 173310 173304  MOV      @DURIS,@DURIV ;RESTORE TRAPCATCHER
1905 006432 012777 000000 173300  MOV      #0,@DURIS    ;
1906 006440 016777 173300 173274  MOV      @DUTIS,@DUTIV ;
1907 006446 012777 000000 173270  MOV      #0,@DUTIS    ;
1908 006454 012716 006634      MOV      #4,(SP)      ;SET UP RETURN LOCATION
1909 006460 042777 000100 173224  BIC      @RINTEN,@RXCSR ;CLR INTERRUPT ENABLES
1910 006466 042777 000100 173232  BIC      @TXINTE,@TXCSR ;
1911 006474 016705 172424      MOV      COUNT,R5     ;SAVE COUNT
1912 006500 000002      RTI
1913      ;END OF RECEIVER INTERRUPT SVC ROUTINE
1914      ;.....THE FOLLOWING IS THE XMITTER INTERRUPT SVC ROUTINE
1915 006502 005367 172416      3$: DEC      COUNT
1916 006506 100403      BMI      5$
1917 006510 012716 006300      MOV      #1,(SP)      ;SET UP RETURN LOCATION
1918      ;(LOAD SYNC CHARACTER AGAIN)
1919 006514 000002      RTI
1920 006516 012716 006524      5$: MOV      #6,(SP)      ;SET UP RETURN LOCATION
1921 006522 000002      RTI
1922      ;END OF XMITTER INTERRUPT SVC ROUTINE
1923 006524 112777 000025 173200  6$: MOV      @25,@TXDBUF ;LOAD CHARACTER
1924 006532 042777 000100 173166  BIC      @TXINTE,@TXCSR ;CLR INTR ENABLE
1925 006540 005067 172744      CLR      $TMP5
1926 006544 005002      10$: CLR      R2        ;WAIT FOR INTERRUPT(RECEIVER)
1927 006546 005202      INC      R2
1928 006550 001376      BNE      #-2          ;
1929 006552 005267 172732      INC      $TMP5
1930 006556 022767 000003 172724  CMP      #3,$TMP5
1931 006564 002367      BGE      10$
1932 006566 106427 000340      MTPS    #340        ;PREVENT INTERRUPTS
1933 006572 042777 000100 173112  BIC      @RINTEN,@RXCSR ;CLR INTR ENABLE
1934 006600 016777 173134 173130  MOV      @DURIS,@DURIV ;RESTORE TRAPCATCHER
1935 006606 012777 000000 173124  MOV      #0,@DURIS    ;
1936 006614 016777 173124 173120  MOV      @DUTIS,@DUTIV ;
1937 006622 012777 000000 173114  MOV      #0,@DUTIS    ;
1938 006630 104004      ERROR   4          ;RECEIVER INTR FAILED TO OCCUR
1939 006632 000426      BR      7$          ;GET OUT OF TEST
1940 006634 020001      4$: CMP      R0,R1
1941 006636 001401      BEQ     .+4
1942 006640 104002      ERROR   2          ;CHARACTERS DID NOT MATCH
1943 006642 016703 173044      MOV      @RXCSR,R3    ;SET UP FOR ERROR MSG
1944 006646 012700 000200      MOV      #200,R0     ;EXPECTED RXDONE
1945 006652 010401      MOV      R4,R1       ;ACTUAL
1946 006654 042701 177577      BIC      #177577,R1   ;SAVE ONLY RXDONE
1947 006660 020001      CMP      R0,R1
1948 006662 001401      BEQ     .+4
1949 006664 104001      ERROR   1          ;FALSE INTERRUPT
1950 006666 020527 177777      CMP      R5,#-1      ;WAS COUNT =-1 WHEN RECEIVER
1951      ;INTERRUPTED ?
1952 006672 001401      BEQ     .+4
1953 006674 104004      ERROR   4          ;IF R5 IS GREATER THAN -1.....
1954      ;THEN EITHER THE # OF SYNC STRAP IS WRONG...
1955      ;OR RXDONE IS OCCURING TOO SOON
1956 006676 026727 172222 177777      CMP      COUNT,#-1
1957 006704 001401      BEQ     .+4
    
```

```

1958 006706 104004          ERROR 4          ;IF THIS TEST FAILS,BUT THE ABOVE TEST
1959                                     ;DOESN'T.....IT MAY BE THAT CLEARING
1960                                     ;TXINTE IN THE RECEIVER SVC ROUTINE
1961                                     ;IS NOT STOPPING TXDONE INTERRUPTS
1962 006710 106427 000340    75:  MTPS  #340      ;INHIBIT INTERRUPTS
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978

```

```

;: THIS TEST VERIFYS MATCH DETECT & DATA RDY
;: FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
;: BY OBSERVING RECACT BIT
;: IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
;: * DEPENDENT ON MONITOR
;: IF ONE SYNC STRAP IS SELECTED ,IT WILL
;: ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
;: ASSERT
;: MODE: SYNC INTERNAL
;: LENGTH: FIVE
;: SYNC CHARACTER FOR MATCH: B/C
;: THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS

```

```

1979 006714 000004          †ST12: SCOPE
1980 006716 052777 000400 173002  BIS      #MRESET,@TXCSR ;MASTER RESET
1981 006724 016703 172766  MOV      RXDBUF,R3    ;SET UP FOR ERROR MESSAGE
1982                                     ;SET SYNC INTERNAL,FIVE NO PARITY 0 SYNC REGISTER
1983 006730 012704 030000  MOV      #SYNINT!FIVE!NOPAR,R4 ;CREATE PARAMETERS
1984 006734 012777 004020 172764  65:  MOV      #MINT!SEND,@TXCSR ;SET SEND & MAINT INTER
1985 006742 010477 172754  MOV      R4,@PARCSR  ;LOAD CSR
1986 006746 052777 000020 172736  BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1987                                     ;POKE CLK TO GET INTO SYNCRONIZATION
1988                                     ;BOTH THE LOGIC & RECEIVER
1989 006754 052777 020000 172744  BIS      #CLK,@TXCSR  ;POKE CLK UP
1990 006762 042777 020000 172736  BIC      #CLK,@TXCSR  ;POKE CLK DOWN
1991 006770 110477 172736  MOV      R4,@TXDBUF  ;LOAD DATA CHARACTER
1992                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCRONIZATION
1993 006774 052777 020000 172724  BIS      #CLK,@TXCSR  ;POKE CLK UP
1994 007002 042777 020000 172716  BIC      #CLK,@TXCSR  ;POKE CLK DOWN
1995 007010 032777 004000 172674  BIT      #RECACT,@RXCSR ;RECACT ?
1996 007016 001401  BEQ      .+4
1997 007020 104034  ERROR 4          ;RECACT SHOULD NOT BE SET
1998 007022 000404  BR      45
1999 007024 010477 172672  55:  MOV      R4,@PARCSR ;LOAD PARCSR WITH PARAMETERS
2000 007030 110477 172676  MOV      R4,@TXDBUF  ;LOAD SYNC CHAR
2001 007034 012767 000002 172062  45:  MOV      #2,COUNT  ;# OF SYNC CHARS
2002 007042 005777 172660  25:  TST      @TXCSR ;DNA ?
2003 007046 100001  BPL      .+4 ;BR IF NOT SET
2004 007050 104004  ERROR 4          ;DNA SHOULD NOT BE SET OR...
2005                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2006 007052 012767 000005 172042  MOV      #5,SHIFT   ;# OF SHIFTS
2007 007060
2008 007060 052777 020000 172640  15:  BIS      #CLK,@TXCSR ;POKE CLK UP
2009 007066 042777 020000 172632  BIC      #CLK,@TXCSR ;POKE CLK DOWN
2010 007074 005367 172022  DEC      SHIFT ;# OF SHIFTS
2011 007100 001367  BNE      15
2012 007102 005367 172016  DEC      COUNT ;# OF SYNC CHARS
2013 007106 001403  BEQ      35

```

INITIALIZE THE COMMON TAGS

```

2014 ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2015 007110 105767 172032 TSTB SYNCNO
2016 007114 100752 BMI 25 ;TWO SYNC CHARACTERS..
2017 007116 032777 004000 172566 3$: BIT #REACT, @RXCSR ;REACT ?
2018 007124 001001 BNE +4
2019 007126 104004 ERROR 4 ;REACT FAILED TO SET, POSSIBLE
2020 ; THAT THE RECEIVER FAILED TO MATCH
2021 ; THE SYNC CHARACTER
2022 007130 017701 172562 MOV @RXDBUF, R1 ;SAVE ACTUAL
2023 007134 010400 MOV R4, R0 ;SAVE EXPECTED
2024 007136 042700 177400 BIC #177400, R0 ;CLR UPPER BYTE
2025 007142 020001 CMP R0, R1 ;DO THEY COMPARE ?
2026 007144 001401 BEQ +4
2027 007146 104002 ERROR 2 ;IF REACT FAILED ALONG WITH THIS
2028 ; IT PROBABLY IS A TRANSMITTER ERROR
2029 ; HOWEVER, ... IF ONLY THIS FAILED IT
2030 ; PROBABLY IS A RECEIVER ERROR
2031 007150 104405 SCOPI
2032 ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2033 ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2034 ;TXDBUF
2035 007152 052777 020000 172546 BIS #CLK, @TXCSR ;POKE CLK UP
2036 007160 005777 172542 TST @TXCSR ;DNA?
2037 007164 100401 BMI +4
2038 007166 104004 ERROR 4 ;DNA DID NOT ASSERT
2039 ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2040 007170 052777 000400 172530 BIS #MRESET, @TXCSR ;MASTER RESET
2041 007176 032777 000020 172506 BIT #SYNSCH, @RXCSR ;SYNC SEARCH = 0 ?
2042 007204 001401 BEQ +4
2043 007206 104004 ERROR 4 ;SYNC SEARCH SHOULD BE NOT SET
2044 007210 005204 INC R4
2045 007212 122704 000040 CMPB #40, R4 ;IS THIS THE LAST CHARACTER ?
2046 007216 001246 BNE 6$ ;NO
2047
2048 ;; THIS TEST VERIFYS MATCH DETECT & DATA RDY
2049 ;; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2050 ;; BY OBSERVING REACT BIT
2051 ;; IT WILL TAKE TWO SYNC * CHARACTERS TO GET REACT BIT
2052 ;; * DEPENDENT ON MONITOR
2053 ;; IF ONE SYNC STRAP IS SELECTED, IT WILL
2054 ;; ONLY TAKE ONE SYNC CHARACTER BEFORE REACT TO
2055 ;; ASSERT
2056 ;; MODE: SYNC INTERNAL
2057 ;; LENGTH: SIX
2058 ;; SYNC CHARACTER FOR MATCH: B/C
2059 ;; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2060
2061 *****
2062 007220 000004 $T13: SCOPE
2063 007222 052777 000400 172476 BIS #MRESET, @TXCSR ;MASTER RESET
2064 007230 016703 172462 MOV @RXDBUF, R3 ;SET UP FOR ERROR MESSAGE
2065 ;SET SYNC INTERNAL, SIX, NO PARITY, 0 SYNC REGISTER
2066 007234 012704 032000 MOV #SYNINT!SIX!NOPAR, R4 ;CREATE PARAMETERS
2067 007240 012777 004020 172460 6$: MOV #MINT!SEND, @TXCSR ;SET SEND & MAINT INTER
2068 007246 010477 172450 MOV R4, @PARCSR ;LOAD CSR
2069 007252 052777 000020 172432 BIS #SYNSCH, @RXCSR ;SET SYNC SEARCH

```

E04

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 45
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

2070      :POKE CLK TO GET INTO SYNCHRONIZATION
2071      :BOTH THE LOGIC & RECEIVER
2072      007260 052777 020000 172440      BIS      #CLK,@TXCSR      ;POKE CLK UP
2073      007266 042777 020000 172432      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2074      007274 110477 172432      MOVB     R4,@TXDBUF      ;LOAD DATA CHARACTER
2075      :POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
2076      007300 052777 020000 172420      BIS      #CLK,@TXCSR      ;POKE CLK UP
2077      007306 042777 020000 172412      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2078      007314 032777 004000 172370      BIT      #REACT,@RXCSR    ;REACT ?
2079      007322 001401      BEQ      .+4
2080      007324 104004      ERROR    4              ;REACT SHOULD NOT BE SET
2081      007326 000404      BR
2082      007330 010477 172366      55:     MOV      R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2083      007334 110477 172372      MOVB     R4,@TXDBUF      ;LOAD SYNC CHAR
2084      007340 012767 000002 171556      45:     MOV      #2,COUNT      ;# OF SYNC CHARS
2085      007346 005777 172354      25:     TST      @TXCSR      ;DNA ?
2086      007352 100001      BPL      .+4            ;BR IF NOT SET
2087      007354 104004      ERROR    4              ;DNA SHOULD NOT BE SET OR...
2088      :IT SHOULD BE CLEARED FROM PREVIOUS READ
2089      007356 012767 000006 171536      15:     MOV      #6,SHIFT      ;# OF SHIFTS
2090      007364
2091      007364 052777 020000 172334      BIS      #CLK,@TXCSR      ;POKE CLK UP
2092      007372 042777 020000 172326      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2093      007400 005367 171516      DEC      SHIFT          ;# OF SHIFTS
2094      007404 001367      BNE      15
2095      007406 005367 171512      DEC      COUNT          ;# OF SYNC CHARS
2096      007412 001403      BEQ      35
2097      :TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2098      007414 105767 171526      TSTB     SYNCNO
2099      007420 100752      BMI      25            ;TWO SYNC CHARACTERS..
2100      007422 032777 004000 172262      35:     BIT      #REACT,@RXCSR    ;REACT ?
2101      007430 001001      BNE      .+4
2102      007432 104004      ERROR    4              ;REACT FAILED TO SET POSSIBLE
2103      :THAT THE RECEIVER FAILED TO MATCH
2104      :THE SYNC CHARACTER
2105      007434 017701 172256      MOV      @RXDBUF,R1      ;SAVE ACTUAL
2106      007440 010400      MOV      R4,R0           ;SAVE EXPECTED
2107      007442 042700 177400      BIC      #177400,R0      ;CLR UPPER BYTE
2108      007446 020001      CMP      R0,R1          ;DO THEY COMPARE ?
2109      007450 001401      BEQ      .+4
2110      007452 104002      ERROR    2              ;IF REACT FAILED ALONG WITH THIS
2111      :...IT PROBABLY IS A TRANSMITTER ERROR
2112      :HOWEVER... IF ONLY THIS FAILED IT
2113      :PROBABLY IS A RECEIVER ERROR
2114      007454 104405      SCOPI
2115      :POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2116      :BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2117      :TXDBUF
2118      007456 052777 020000 172242      BIS      #CLK,@TXCSR      ;POKE CLK UP
2119      007464 005777 172236      TST      @TXCSR      ;DNA?
2120      007470 100401      BMI      .+4
2121      007472 104004      ERROR    4              ;DNA DID NOT ASSERT
2122      :SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2123      007474 052777 000400 172224      BIS      #MRESET,@TXCSR    ;MASTER RESET
2124      007502 032777 000020 172202      BIT      #SYNSCH,@RXCSR    ;SYNC SEARCH = 0 ?
2125      007510 001401      BEQ      .+4
  
```

F04

DZDUV-A MACY11 P7(1006) 03-FEB-77 08:30 PAGE 46
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

2126 007512 104004          ERROR 4      ; SYNC SEARCH SHOULD BE NOT SET
2127 007514 005204          INC    R4
2128 007516 122704 000100  CMPB  #100,R4 ; IS THIS THE LAST CHARACTER ?
2129 007522 001246          BNE   B$      ; NO
2130
2131          ; THIS TEST VERIFYS MATCH DETECT & DATA RDY
2132          ; FLAGS FOR EVERY POSSIBLE MATCH CHARACTER
2133          ; BY OBSERVING RECACT BIT
2134          ; IT WILL TAKE TWO SYNC * CHARACTERS TO GET RECACT BIT
2135          ; * DEPENDENT ON MONITOR
2136          ; IF ONE SYNC STRAP IS SELECTED IT WILL
2137          ; ONLY TAKE ONE SYNC CHARACTER BEFORE RECACT TO
2138          ; ASSERT
2139          ; MODE: SYNC INTERNAL
2140          ; LENGTH: SEVEN
2141          ; SYNC CHARACTER FOR MATCH: B/C
2142          ; THIS TEST USES THE TRANSMITTER AND RECEIVER CHIPS
2143
2144          ; *****
2145 007524 003004          †ST14: SCOPE
2146 007526 052777 000400 172172  BIS   #MRESET,@TXCSR ; MASTER RESET
2147 007534 016703 172156          MOV   RXDBUF,R3      ; SET UP FOR ERROR MESSAGE
2148          ; SET SYNC INTERNAL, SEVEN NO PARITY 0 SYNC REGISTER
2149 007540 012704 034000          MOV   #SYNINT!SEVEN!NOPAR,R4 ; CREATE PARAMETERS
2150 007544 012777 004020 172154  B$:  MOV   #MINT!SEND,@TXCSR ; SET SEND & MAINT INTER
2151 007552 010477 172144          MOV   R4,@PARCSR    ; LOAD CSR
2152 007556 052777 000020 172126  BIS   #SYNSCH,@RXCSR ; SET SYNC SEARCH
2153          ; POKE CLK TO GET INTO SYNCHRONIZATION
2154          ; BOTH THE LOGIC & RECEIVER
2155 007564 052777 020000 172134  BIS   #CLK,@TXCSR    ; POKE CLK UP
2156 007572 042777 020000 172126  BIC   #CLK,@TXCSR    ; POKE CLK DOWN
2157 007600 110477 172126          MOVB  R4,@TXDBUF     ; LOAD DATA CHARACTER
2158          ; POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
2159 007604 052777 020000 172114  BIS   #CLK,@TXCSR    ; POKE CLK UP
2160 007612 042777 020000 172106  BIC   #CLK,@TXCSR    ; POKE CLK DOWN
2161 007620 032777 004000 172064  BIT   #RECACT,@RXCSR ; RECACT ?
2162 007626 001401          BEQ   .+4
2163 007630 104004          ERROR 4      ; RECACT SHOULD NOT BE SET
2164 007632 000404          BR   4$
2165 007634 010477 172062  5$:  MOV   R4,@PARCSR    ; LOAD PARCSR WITH PARAMETERS
2166 007640 110477 172066          MOVB  R4,@TXDBUF     ; LOAD SYNC CHAR
2167 007644 012767 000002 171252  4$:  MOV   #2,COUNT      ; # OF SYNC CHARS
2168 007652 005777 172050  2$:  TST   @TXCSR        ; DNA ?
2169 007656 100001          BPL   .+4           ; BR IF NOT SET
2170 007660 104004          ERROR 4      ; DNA SHOULD NOT BE SET OR...
2171          ; IT SHOULD BE CLEARED FROM PREVIOUS READ
2172 007662 012767 000007 171232  MOV   #7,SHIFT      ; # OF SHIFTS
2173 007670
2174 007670 052777 020000 172030  1$:  BIS   #CLK,@TXCSR    ; POKE CLK UP
2175 007676 042777 020000 172022  BIC   #CLK,@TXCSR    ; POKE CLK DOWN
2176 007704 005367 171212          DEC   SHIFT         ; # OF SHIFTS
2177 007710 001367          BNE   1$
2178 007712 005367 171206          DEC   COUNT         ; # OF SYNC CHARS
2179 007716 001403          BEQ   3$
2180          ; TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2181 007720 105767 171222          TSTB  SYNCNO
  
```



```

2238 010070 052777 020000 171630      BIS      #CLK,@TXCSR      ;POKE CLK UP
2239 010076 042777 020000 171622      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2240 010104 110477 171622      MOV      R4,@TXDBUF      ;LOAD DATA CHARACTER
2241                                     ;POKE CLK TO GET TRANSMITTER & RECEIVER INTO SYNCHRONIZATION
2242 010110 052777 020000 171610      BIS      #CLK,@TXCSR      ;POKE CLK UP
2243 010116 042777 020000 171602      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2244 010124 032777 004000 171560      BIT      #REACT,@RXCSR    ;REACT ?
2245 010132 001401                                     BEQ      .+4
2246 010134 104004                                     ERROR    4      ;REACT SHOULD NOT BE SET
2247 010136 000404                                     BR      4$
2248 010140 010477 171556      5$:     MOV      R4,@PARCSR      ;LOAD PARCSR WITH PARAMETERS
2249 010144 110477 171562      MOV      R4,@TXDBUF      ;LOAD SYNC CHAR
2250 010150 012767 000002 170746      4$:     MOV      #2,COUNT      ;# OF SYNC CHARS
2251 010156 005777 171544      2$:     TST      @TXCSR      ;DNA ?
2252 010162 100001                                     BPL      .+4      ;BR IF NOT SET
2253 010164 104004                                     ERROR    4      ;DNA SHOULD NOT BE SET OR...
2254                                     ;IT SHOULD BE CLEARED FROM PREVIOUS READ
2255 010166 012767 000010 170726      MOV      #8.,SHIFT      ;# OF SHIFTS
2256 010174                                     1$:
2257 010174 052777 020000 171524      BIS      #CLK,@TXCSR      ;POKE CLK UP
2258 010202 042777 020000 171516      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2259 010210 005367 170706      DEC      SHIFT      ;# OF SHIFTS
2260 010214 001367                                     BNE     1$
2261 010216 005367 170702      DEC      COUNT      ;# OF SYNC CHARS
2262 010222 001403                                     BEQ     3$
2263                                     ;TEST SYNCNO TO SEE HOW MANY SYNC CHARACTERS NEEDED
2264 010224 105767 170716      TSTB     SYNCNO
2265 010230 100752                                     BMI     2$      ;TWO SYNC CHARACTERS..
2266 010232 032777 004000 171452      3$:     BIT      #REACT,@RXCSR    ;REACT ?
2267 010240 001001                                     BNE     .+4
2268 010242 104004                                     ERROR    4      ;REACT FAILED TO SET,POSSIBLE
2269                                     ;THAT THE RECEIVER FAILED TO MATCH
2270                                     ;THE SYNC CHARACTER
2271 010244 017701 171446      MOV      @RXDBUF,R1      ;SAVE ACTUAL
2272 010250 010400      MOV      R4,R0      ;SAVE EXPECTED
2273 010252 042700 177400      BIC      #177400,R0      ;CLR UPPER BYTE
2274 010256 020001      CMP      R0,R1      ;DO THEY COMPARE ?
2275 010260 001401                                     BEQ     .+4
2276 010262 104002      ERROR    2      ;IF REACT FAILED ALONG WITH THIS
2277                                     ;...IT PROBABLY IS A TRANSMITTER ERROR
2278                                     ;HOWEVER,...IF ONLY THIS FAILED IT
2279                                     ;PROBABLY IS A RECEIVER ERROR
2280 010264 104405      SCOPI
2281                                     ;POKE CLK TO SEE DNA...DNA COMES UP ON THE FIRST
2282                                     ;BIT OF THE NEXT CHARACTER IF NO CHARACTER IS LOADED INTO
2283                                     ;TXDBUF
2284 010266 052777 020000 171432      BIS      #CLK,@TXCSR      ;POKE CLK UP
2285 010274 005777 171426      TST      @TXCSR      ;DNA?
2286 010300 100401                                     BMI     .+4
2287 010302 104004      ERROR    4      ;DNA DID NOT ASSERT
2288                                     ;SET UP CONDITIONS FOR NEXT SYNC CHARACTER MATCH
2289 010304 052777 000400 171414      BIS      #MRESET,@TXCSR    ;MASTER RESET
2290 010312 032777 000020 171372      BIT      #SYNSCH,@RXCSR    ;SYNC SEARCH = 0 ?
2291 010320 001401                                     BEQ     .+4
2292 010322 104004      ERROR    4      ;SYNC SEARCH SHOULD BE NOT SET
2293 010324 005204      INC      R4

```

```

2294 010326 122704 000000      CMPB  #0,R4  ;IS THIS THE LAST CHARACTER ?
2295 010332 001246      BNE    6$    ;NO
2296
2297      ;: THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2298      ;: BOTH THE TRANSMITTER AND RECEIVER LOGIC
2299      ;: MODE: SYNC EXTERNAL (SYNEXT)
2300      ;: LENGTH: EIGHT PLUS PARITY
2301      ;: PARITY: EVEPAR
2302      ;: MAINT. MODE: MINT
2303
2304      ;: *****
2305 010334 000004      †ST16: SCOPE
2306 010336 052777 000400 171362      BIS    #MRESET,@TXCSR ;MASTER RESET
2307 010344 012777 020000 171350      MOV    #SYNEXT,@PARCSR ;SET THE MODE
2308 010352 052777 000400 171346      BIS    #MRESET,@TXCSR ;MASTER RESET
2309
2310      ;SET MAINTENANCE MODE & SEND
2311      ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2312 010360 012777 004020 171340      MOV    #MINT!SEND,@TXCSR
2313
2314      ;SET MODE # OF BITS, PARITY SENSE, & LOAD SYNC REG
2315 010366 012777 027426 171326      MOV    #SYNEXT!EIGHT!EVEPAR!26,@PARCSR
2316 010374 016703 171316      MOV    RXDBUF,R3      ;SETUP FOR ERROR MSG
2317 010400 005004      CLR    R4              ;FOR DATA CHAR CREATION
2318 010402 110477 171324      MOVB   R4,@TXDBUF     ;LOAD CHARACTER
2319 010406 052777 000020 171276      BIS    #SYNSCH,@RXCSR ;SET SEARCH SYNC
2320      ;GET INTO SYNCHRONIZATION
2321 010414 052777 020000 171304      BIS    #CLK,@TXCSR    ;POKE CLK UP
2322 010422 042777 020000 171276      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
2323 010430 012767 000011 170464      1$:   MOV    #9,SHIFT    ;# OF SHIFTS
2324 010436 010400      MOV    R4,R0          ;EXPECTED
2325
2326 010440 052777 020000 171260      2$:   BIS    #CLK,@TXCSR    ;POKE CLK UP
2327 010446 042777 020000 171252      BIC    #CLK,@TXCSR    ;POKE CLK DOWN
2328 010454 005367 170442      DEC    SHIFT          ;# OF SHIFTS
2329 010460 022767 000003 170434      CMP    #3,SHIFT       ;TIME TO LOAD NEXT CHAR ?
2330 010466 001003      BNE    3$             ;NO ?
2331 010470 005204      INC    R4             ;GENERATE NEXT CHAR
2332 010472 110477 171234      MOVB   R4,@TXDBUF     ;LOAD NEXT CHARACTER
2333 010476 005767 170420      3$:   TST    SHIFT         ;IS IT 0 ?
2334 010502 001356      BNE    2$
2335 010504 105777 171202      TSTB   @RXCSR         ;RXDONE = 1 ?
2336 010510 100401      BMI    .+4
2337 010512 104004      ERROR  4              ;RXDONE SHOULD BE SET
2338 010514 017701 171176      MOV    @RXDBUF,R1     ;ACTUAL
2339 010520 020001      CMP    R0,R1          ;COMPARE EXP VS ACT
2340 010522 001401      BEQ    .+4
2341 010524 104002      ERROR  2              ;CHARACTERS SHOULD COMPARE
2342 010526 105704      TSTB   R4             ;LAST CHARACTER ?
2343 010530 001337      BNE    1$             ;NO
2344
2345      4$:
2346      ;: THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2347      ;: BOTH THE TRANSMITTER AND RECEIVER LOGIC
2348      ;: MODE: SYNC EXTERNAL (SYNEXT)
2349      ;: LENGTH: EIGHT PLUS PARITY
2350      ;: PARITY: ODDPAR
  
```



```

2350          ;;MAINT. MODE:MINT
2351          ;
2352          ;*****
2353 010532 000004          †ST17: SCOPE
2354 010534 052777 000400 171164      BIS      #MRESET,@TXCSR ;MASTER RESET
2355 010542 012777 020000 171152      MOV      #SYNEXT,@PARCSR ;SET THE MODE
2356 010550 052777 000400 171150      BIS      #MRESET,@TXCSR ;MASTER RESET
2357          ;
2358          ;SET MAINTENANCE MODE & SEND
2359          ;NOTE:BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2360 010556 012777 004020 171142      MOV      #MINT!SEND,@TXCSR
2361          ;
2362          ;SET MODE # OF BITS,PARITY SENSE & LOAD SYNC REG
2363 010564 012777 027026 171130      MOV      #SYNEXT!EIGHT!ODD!PAR!26,@PARCSR
2364 010572 016703 171120          MOV      RXDBUF,R3 ;SETUP FOR ERROR MSG
2365 010576 005004          CLR      R4 ;FOR DATA CHAR CREATION
2366 010600 110477 171126          MOV      R4,@TXDBUF ;LOAD CHARACTER
2367 010604 052777 000020 171100      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2368          ;GET INTO SYNCHRONIZATION
2369 010612 052777 020000 171106      BIS      #CLK,@TXCSR ;POKE CLK UP
2370 010620 042777 020000 171100      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2371 010626 012767 000011 170266 1$: MOV      #9,SHIFT ;# OF SHIFTS
2372 010634 010400          MOV      R4,R0 ;EXPECTED
2373          ;
2374 010636 052777 020000 171062      BIS      #CLK,@TXCSR ;POKE CLK UP
2375 010644 042777 020000 171054      BIC      #CLK,@TXCSR ;POKE CLK DOWN
2376 010652 005367 170244          DEC      SHIFT ;# OF SHIFTS
2377 010656 022767 000003 170236      CMP      #3,SHIFT ;TIME TO LOAD NEXT CHAR ?
2378 010664 001003          BNE      3$ ;NO ?
2379 010666 005204          INC      R4 ;GENERATE NEXT CHAR
2380 010670 110477 171036          MOV      R4,@TXDBUF ;LOAD NEXT CHARACTER
2381 010674 005767 170222 3$: TST      SHIFT ;IS IT 0 ?
2382 010700 001356          BNE      2$
2383 010702 105777 171004          TSTB    @RXCSR ;RXDONE = 1 ?
2384 010706 100401          BMI      +4
2385 010710 104004          ERROR   4 ;RXDONE SHOULD BE SET
2386 010712 017701 171000          MOV      @RXDBUF,R1 ;ACTUAL
2387 010716 020001          CMP      R0,R1 ;COMPARE EXP VS ACT
2388 010720 001401          BEQ      +4
2389 010722 104002          ERROR   2 ;CHARACTERS SHOULD COMPARE
2390 010724 105704          TSTB    R4 ;LAST CHARACTER ?
2391 010726 001337          BNE      1$ ;NO
2392          ;
2393          ;
2394          ;
2395          ;
2396          ;
2397          ;
2398          ;
2399          ;
2400          ;
2401 010730 000004          †ST20: SCOPE
2402 010732 105767 170215          TSTB    JMRBY ;JUMP AROUND TEST ?
2403 010736 100116          BPL      4$ ;YES ?
2404 010740 052777 000400 170760      BIS      #MRESET,@TXCSR ;MASTER RESET
2405 010746 012777 020000 170746      MOV      #SYNEXT,@PARCSR ;SET THE MODE
  
```

K04

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 51
 DZDUVA.M11 13-OCT-76 08:26

INITIALIZE THE COMMON TAGS

```

2406 010754 052777 000400 170744      BIS      #MRESET, @TXCSR ; MASTER RESET
2407
2408      ;SET MAINTENANCE MODE & SEND
2409      ;NOTE: BIT WINDOW&CLK ARE CLEARED (MTDATA=0)
2410 010762 012777 010020 170736      MOV      #NEXT!SEND, @TXCSR
2411
2412      ;SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
2413 010770 012777 027426 170724      MOV      #SYNEXT!EIGHT!EVEPAR!26, @PARCSR
2414 010776 016703 170714      MOV      @RXDBUF, R3 ; SETUP FOR ERROR MSG
2415 011002 005004      CLR      R4 ; FOR DATA CHAR CREATION
2416 011004 110477 170722      MOV      @TXDBUF, R4 ; LOAD CHARACTER
2417 011010 052777 000020 170674      BIS      #SYNSCH, @RXCSR ; SET SEARCH SYNC
2418      ;GET INTO SYNCRONIZATION
2419 011016 052777 020000 170702      BIS      #CLK, @TXCSR ; POKE CLK UP
2420      ;WAIT FOR CABLE & DRIVER DELAYS
2421 011024 016702 170070      MOV      HOLD, R2 ; WAIT THIS AMT
2422 011030 005302      DEC      R2 ; WAIT
2423 011032 001376      BNE      .-2
2424      ;EXIT...
2425 011034 042777 020000 170664      BIC      #CLK, @TXCSR ; POKE CLK DOWN
2426      ;WAIT FOR CABLE & DRIVER DELAYS
2427 011042 016702 170052      MOV      HOLD, R2 ; WAIT THIS AMT
2428 011046 005302      DEC      R2 ; WAIT
2429 011050 001376      BNE      .-2
2430      ;EXIT...
2431 011052 012767 000011 170042 1$:  MOV      #9, SHIFT ; # OF SHIFTS
2432 011060 010400      MOV      R4, R0 ; EXPECTED
2433 011062
2434 011062 052777 020000 170636 2$:  BIS      #CLK, @TXCSR ; POKE CLK UP
2435      ;WAIT FOR CABLE & DRIVER DELAYS
2436 011070 016702 170024      MOV      HOLD, R2 ; WAIT THIS AMT
2437 011074 005302      DEC      R2 ; WAIT
2438 011076 001376      BNE      .-2
2439      ;EXIT...
2440 011100 042777 020000 170620      BIC      #CLK, @TXCSR ; POKE CLK DOWN
2441      ;WAIT FOR CABLE & DRIVER DELAYS
2442 011106 016702 170006      MOV      HOLD, R2 ; WAIT THIS AMT
2443 011112 005302      DEC      R2 ; WAIT
2444 011114 001376      BNE      .-2
2445      ;EXIT...
2446 011116 005367 170000      DEC      SHIFT ; # OF SHIFTS
2447 011122 022767 000003 167772 3$:  CMP      #3, SHIFT ; TIME TO LOAD NEXT CHAR ?
2448 011130 001003      BNE      3$ ; NO ?
2449 011132 005204      INC      R4 ; GENERATE NEXT CHAR
2450 011134 110477 170572      MOV      @TXDBUF, R4 ; LOAD NEXT CHARACTER
2451 011140 005767 167756      TST      SHIFT ; IS IT 0 ?
2452 011144 001346      BNE      2$
2453 011146 105777 170540      TSTB    @RXCSR ; RXDONE = 1 ?
2454 011152 100401      BMI      .+4
2455 011154 104004      ERROR   4 ; RXDONE SHOULD BE SET
2456 011156 017701 170534      MOV      @RXDBUF, R1 ; ACTUAL
2457 011162 020001      CMP      R0, R1 ; COMPARE EXP VS ACT
2458 011164 001401      BEQ     .+4
2459 011166 104002      ERROR   2 ; CHARACTERS SHOULD COMPARE
2460      ;CHECK OUT MODEM BYPASS JUMPER
2461 011170 105704      TSTB    R4 ; LAST CHARACTER ?
  
```

L04

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 52
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

2462 011172 001327          BNE      1$      ;NO
2463 011174          4$:
2464          ;: THIS TEST PERFORMS A BINARY COUNT DATA PATTERN ON
2465          ;: BOTH THE TRANSMITTER AND RECEIVER LOGIC
2466          ;: MODE: SYNC EXTERNAL (SYNEXT)
2467          ;: LENGTH: EIGHT PLUS PARITY
2468          ;: PARITY: ODDPAR
2469          ;: MAINT. MODE: MEXT
2470          ;:
2471          ;: *****
2472 011174 000004          †ST21: SCOPE
2473 011176 105767 167751  TSTB     JMRBY    ; JUMP AROUND TEST ?
2474 011202 100116          BPL     4$      ; YES ?
2475 011204 052777 000400 170514  BIS     #MRESET, @TXCSR ; MASTER RESET
2476 011212 012777 020000 170502  MOV     #SYNEXT, @PARCSR ; SET THE MODE
2477 011220 052777 000400 170500  BIS     #MRESET, @TXCSR ; MASTER RESET
2478
2479          ; SET MAINTENANCE MODE & SEND
2480          ; NOTE: BIT WINDOW & CLK ARE CLEARED (MTDATA=0)
2481 011226 012777 010020 170472  MOV     #MEXT!SEND, @TXCSR
2482
2483          ; SET MODE # OF BITS, PARITY SENSE, & LOAD SYNC REG
2484 011234 012777 027026 170460  MOV     #SYNEXT!EIGHT!ODDPAR!26, @PARCSR
2485 011242 016703 170450          MOV     RXDBUF, R3      ; SETUP FOR ERROR MSG
2486 011246 005004          CLR     R4            ; FOR DATA CHAR CREATION
2487 011250 110477 170456          MOV     R4, @TXDBUF    ; LOAD CHARACTER
2488 011254 052777 000020 170430  BIS     #SYNSCH, @RXCSR ; SET SEARCH SYNC
2489
2490          ; GET INTO SYNCHRONIZATION
2491 011262 052777 020000 170436  BIS     #CLK, @TXCSR   ; POKE CLK UP
2492          ; WAIT FOR CABLE & DRIVER DELAYS
2493 011270 016702 167624          MOV     HOLD, R2      ; WAIT THIS AMT
2494 011274 005302          DEC     R2            ; WAIT
2495 011276 001376          BNE     .-2
2496          ; EXIT...
2497 011300 042777 020000 170420  BIC     #CLK, @TXCSR   ; POKE CLK DOWN
2498          ; WAIT FOR CABLE & DRIVER DELAYS
2499 011306 016702 167606          MOV     HOLD, R2      ; WAIT THIS AMT
2500 011312 005302          DEC     R2            ; WAIT
2501 011314 001376          BNE     .-2
2502          ; EXIT...
2502 011316 012767 000011 167576 1$: MOV     #9, SHIFT     ; # OF SHIFTS
2503 011324 010400          MOV     R4, R0        ; EXPECTED
2504 011326
2505 011326 052777 020000 170372 2$: BIS     #CLK, @TXCSR   ; POKE CLK UP
2506          ; WAIT FOR CABLE & DRIVER DELAYS
2507 011334 016702 167560          MOV     HOLD, R2      ; WAIT THIS AMT
2508 011340 005302          DEC     R2            ; WAIT
2509 011342 001376          BNE     .-2
2510          ; EXIT...
2511 011344 042777 020000 170354  BIC     #CLK, @TXCSR   ; POKE CLK DOWN
2512          ; WAIT FOR CABLE & DRIVER DELAYS
2513 011352 016702 167542          MOV     HOLD, R2      ; WAIT THIS AMT
2514 011356 005302          DEC     R2            ; WAIT
2515 011360 001376          BNE     .-2
2516          ; EXIT...
2517 011362 005367 167534          DEC     SHIFT        ; # OF SHIFTS
  
```

M04

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 53
 DZDUVA.M11 13-OCT-76 08:26 INITIALIZE THE COMMON TAGS

```

2518 011366 022767 000003 167526      CMP    #3,SHIFT      ;TIME TO LOAD NEXT CHAR ?
2519 011374 001003      BNE    3$            ;NO ?
2520 011376 005204      INC    R4            ;GENERATE NEXT CHAR
2521 011400 110477 170326      MOV    R4,@TXDBUF    ;LOAD NEXT CHARACTER
2522 011404 005767 167512      3$:  TST    SHIFT      ;IS IT 0 ?
2523 011410 001346      BNE    2$            ;
2524 011412 105777 170274      TST    @RXCSR      ;RXDONE = 1 ?
2525 011416 100401      BMI    .+4          ;
2526 011420 104004      ERROR  4            ;RXDONE SHOULD BE SET
2527 011422 017701 170270      MOV    @RXDBUF,R1    ;ACTUAL
2528 011426 020001      CMP    R0,R1        ;COMPARE EXP VS ACT
2529 011430 001401      BEQ    .+4          ;
2530 011432 104002      ERROR  2            ;CHARACTERS SHOULD COMPARE
2531      ;CHECK OUT MODEM BYPASS JUMPER
2532 011434 105704      TST    R4            ;LAST CHARACTER ?
2533 011436 001327      BNE    1$            ;NO
2534 011440      4$:
2535      ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2536      ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
2537      ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2538      ;; (OVRUN, RXERR)
2539      ;; MODE: ISYMOD
2540      ;; LENGTH: FIVE
2541      ;; CHAR: 12
2542      ;;
2543      ;*****
2544 011440 000004      ST22: SCOPE
2545 011442 052777 000400 170256      BIS    #MRESET,@TXCSR ;MASTER RESET
2546 011450 012777 000000 170244      MOV    #ISYMOD,@PARCSR ;SET THE MODE
2547 011456 052777 000400 170242      BIS    #MRESET,@TXCSR ;MASTER RESET
2548
2549      ;SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2550 011464 012777 064001 170234      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2551
2552      ;SET MODE, # OF BITS, PARITY SENSE & LOAD SYNC REG
2553 011472 012777 000000 170222      MOV    #ISYMOD!FIVE!NOPAR!0,@PARCSR
2554 011500 052777 000020 170204      BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2555      ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2556 011506 042777 020000 170212      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2557 011514 052777 020000 170204      BIS    #CLK,@TXCSR   ;POKE CLK UP
2558      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2559 011522 042777 020000 170176      BIC    #CLK,@TXCSR   ;POKE CLK DOWN
2560 011530 052777 020000 170170      BIS    #CLK,@TXCSR   ;POKE CLK UP
2561 011536 016703 170154      MOV    @RXDBUF,R3    ;SET UP FOR ERROR MESSAGE
2562 011542 012700 000012      MOV    #12,R0        ;EXPECTED
2563 011546 012767 000007 167346      MOV    #7,SHIFT      ;# OF SHIFTS
2564 011554 012767 000124 167716      MOV    #124,$TMP1    ;DATA CHAR
2565 011562 004767 005340      JSR    PC,RPOKE      ;SHIFT IN THIS CHAR
2566 011566 105777 170120      TST    @RXCSR      ;RXDONE
2567 011572 100401      BMI    .+4          ;
2568 011574 104004      ERROR  4            ;RXDONE SHOULD BE SET
2569 011576 017701 170114      MOV    @RXDBUF,R1    ;ACTUAL
2570 011602 020001      CMP    R0,R1        ;COMPARE EXPECTED VS. ACTUAL
2571 011604 001401      BEQ    .+4          ;
2572 011606 104002      ERROR  2            ;RECEIVED DATA DID NOT MATCH
2573      ;EXPECTED DATA - CHECK MAINT DATA

```



```

2630 012036 012767 000007 167056      MOV      #7,SHIFT          ;# OF SHIFTS
2631 012044 012767 000176 167426      MOV      #176,$TMP1       ;DATA CHAR
2632 012052 004767 005050              JSR      PC,RPOKE         ;SHIFT IN THIS CHAR
2633                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2634 012056 012767 000007 167036      MOV      #7,SHIFT          ;# OF SHIFTS
2635 012064 012767 000176 167406      MOV      #176,$TMP1       ;DATA CHAR
2636 012072 004767 005030              JSR      PC,RPOKE         ;SHIFT IN THIS CHAR
2637 012076 012700 140037              MOV      #140000!37,RO    ;EXPECTED DATA PLUS
2638                                ;RXERR & OVRUN
2639 012102 017701 167610              MOV      @RXDBUF,R1       ;ACTUAL
2640 012106 020001                    CMP      RO,R1            ;COMPARE EXP VS. ACT
2641 012110 001401                    BEQ      .+4
2642 012112 104002                    ERROR    2                ;SPECIFICALLY LOOK AT RXERR &
2643                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2644
2645                                ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
2646                                ;: RECEIVER SECTION, IT USES THE ERROR FLAGS
2647                                ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
2648                                ;: (OVRUN,RXERR)
2649                                ;: MODE: ISYMOD
2650                                ;: LENGTH: FIVE
2651                                ;: CHAR: 0
2652
2653                                ;*****
2654 012114 000004                    †ST24: SCOPE
2655 012116 052777 000400 167602      BIS      #MRESET,@TXCSR   ;MASTER RESET
2656 012124 012777 000000 167570      MOV      #ISYMOD,@PARCSR  ;SET THE MODE
2657 012132 052777 000400 167566      BIS      #MRESET,@TXCSR   ;MASTER RESET
2658
2659                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2660 012140 012777 064001 167560      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2661
2662                                ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
2663 012146 012777 000000 167546      MOV      #ISYMOD!FIVE!NOPAR!0,@PARCSR
2664 012154 052777 000020 167530      BIS      #SYNSCH,@RXCSR   ;SET SYNC SEARCH
2665                                ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2666 012162 042777 020000 167536      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2667 012170 052777 020000 167530      BIS      #CLK,@TXCSR      ;POKE CLK UP
2668                                ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2669 012176 042777 020000 167522      BIC      #CLK,@TXCSR      ;POKE CLK DOWN
2670 012204 052777 020000 167514      BIS      #CLK,@TXCSR      ;POKE CLK UP
2671 012212 016703 167500              MOV      RXDBUF,R3        ;SET UP FOR ERROR MESSAGE
2672 012216 012700 000000              MOV      #0,RO            ;EXPECTED
2673 012222 012767 000007 166672      MOV      #7,SHIFT          ;# OF SHIFTS
2674 012230 012767 000100 167242      MOV      #100,$TMP1       ;DATA CHAR
2675 012236 004767 004664              JSR      PC,RPOKE         ;SHIFT IN THIS CHAR
2676 012242 105777 167444              TSTB    @RXCSR            ;RXDONE
2677 012246 100401                    BMI      .+4
2678 012250 104004                    ERROR    4                ;RXDONE SHOULD BE SET
2679 012252 017701 167440              MOV      @RXDBUF,R1       ;ACTUAL
2680 012256 020001                    CMP      RO,R1            ;COMPARE EXPECTED VS. ACTUAL
2681 012260 001401                    BEQ      .+4
2682 012262 104002                    ERROR    2                ;RECEIVED DATA DID NOT MATCH
2683                                ;EXPECTED DATA - CHECK MAINT DATA
2684                                ;OR RECEIVER LOGIC
2685 012264 012767 000007 166630      MOV      #7,SHIFT          ;# OF SHIFTS

```

```

2686 012272 012767 000100 167200      MOV      #100,$TMP1      ;DATA CHAR
2687 012300 004767 004622              JSR      PC,RPOKE       ;SHIFT IN THIS CHAR
2688                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
2689 012304 012767 000007 166610      MOV      #7,SHIFT      ;# OF SHIFTS
2690 012312 012767 000100 167160      MOV      #100,$TMP1    ;DATA CHAR
2691 012320 004767 004602              JSR      PC,RPOKE       ;SHIFT IN THIS CHAR
2692 012324 012700 140000              MOV      #140000!0,RO  ;EXPECTED DATA PLUS
2693                                ;RXERR & OVRUN
2694 012330 017701 167362              MOV      @RXDBUF,R1    ;ACTUAL
2695 012334 020001                      CMP      RO,R1         ;COMPARE EXP VS. ACT
2696 012336 001401                      BEQ      +4
2697 012340 104002                      ERROR    2             ;SPECIFICALLY LOOK AT RXERR &
2698                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
2699
2700
2701                                ;END OF PASS
2702                                ;TYPE NAME OF TEST
2703                                ;UPDATE PASS COUNT
2704                                ;CHECK FOR EXIT TO ACT-11
2705                                ;RESTART TEST
2706
2707 012342 000004      .EOP: SCOPE
2708 012344 004767 000344              JSR      PC,CKSWR
2709 012350 104401                      TYPE
2710 012352 015504                      MEPASS                ;TYPE NAME OF TEST
2711 012354 104413 012606              CONVRT    ,OUTCRY
2712 012360 104401 015323              TYPE      ,DEVICE
2713 012364 105767 166562              TSTB     MULTD        ;ARE YOU RUNNING MULTIPLE DEVICES ?
2714 012370 001511                      BEQ      CCC          ;NO JUMP AROUND
2715 012372 005767 166570              TST      ACTREG      ;ARE ANY DEVICES ACTIVE ?
2716 012376 001007                      BNE      RUNIT        ;YES
2717 012400 104401 015335              TYPE      MCOW        ;NO
2718 012404 016700 166556              MOV      @ACTREG,RO   ;DISPLAY ACTREG
2719 012410 000000                      HALT
2720                                ;SELECT SOMETHING TO RUN @ ACTREG:
2721 012412 000167 167534              ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2722 012416 062767 000010 166530      JMP      .START       ;START OVER AGAIN.....YOU DESELECTED EVERYTHING
2723 012424 062767 000010 166530      RUNIT: ADD    #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2724 012432 000241                      ZERC:  ADD    #10,BASEIV  ;NEXT BLOCK (VECTORS)
2725 012434 006167 166530              CLC
2726 012440 103410                      ROTADD   ;UP DATE ROTATING POINTER
2727                                ;IS IT THE LAST DEVICE
2728 012442 036767 166522 166516      BCS     2$           ;TO BE TESTED IN THIS PASS ?
2729 012450 001762                      BIT      ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2730 012452 004767 000034              BEQ      RUNIT        ;IF NOT ACTIVE, TRY NEXT ADDRESS
2731 012456 000167 000210              JSR      PC,REPLAY    ;CALCULATE NEW PARAMETERS
2732 012462 012767 000001 166500      JMP      RE$TRT       ;YES IT WAS ACTIVE TEST THIS DEVICE
2733                                ;OK!, NOW SET UP ROTATING
2734 012470 016767 166462 166456      MOV      #1,ROTADD    ;POINTER FOR NEXT MULTIPLE PASS
2735 012476 016767 166462 166456      MOV      KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2736 012504 004767 000002              MOV      KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
2737 012510 000441                      JSR      PC,REPLAY    ;CALC NEW PARAMETERS
2738 012512 016767 166436 004404      BR       CCC          ;JUMP AROUND REPLAY
2739 012520 004767 004246              REPLAY: MOV     BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
2740 012524 016767 166432 167204      JSR      PC,DUADD     ;CREATE NEW ADDRESSES
2741 012532 062767 000002 166422      MOV      BASEIV,DURIV ;CREATE DURIV
                ADD    #2,BASEIV
  
```

```

2742 012540 016767 166416 167172      MOV    BASEIV,DURIS      ;CREATE DURIS
2743 012546 062767 000002 166406      ADD    #2,BASEIV
2744 012554 016767 166402 167160      MOV    BASEIV,DUTIV     ;CREATE DUTIV
2745 012562 062767 000002 166372      ADD    #2,BASEIV
2746 012570 016767 166366 167146      MOV    BASEIV,DUTIS     ;CREATE DUTIS
2747 012576 016767 167134 166356      MOV    DURIV,BASEIV     ;RESTORE
2748 012604 000207
2749
2750 012606 000001      OUTCRY: 1
2751 012610      006      002      .BYTE 6,2
2752 012612 001712      RXCSR
2753
2754 012614      CCC:
2755 012614 005067 166562      CLR    $TSTNM           ;CLEAR TEST NUMBER
2756 012620 005067 166572      CLR    $ERRPC          ;CLEAR LAST ERROR PC
2757 012624 005067 166553      CLR    $ERFLG         ;CLEAR ERROR FLAG
2758 012630 005267 166256      INC    PASCNT          ;UPDATE PASS COUNT
2759 012634 016767 166252 166240      MOV    PASCNT,LIGHTS  ;DISPLAY PASS COUNT
2760 012642 016767 166244 166664      MOV    PASCNT,$PASS   ;PASS COUNT TO APT
2761 012650 013701 000042      MOV    #42,R1         ;CHECK FOR ACT-11 OR DDP
2762 012654 001406      BEQ    RESTAT         ;IF NO CONTINUE TESTING
2763 012656 000005      RESET
2764 012660 000005      RESET
2765 012662 004711      $ENDAD: JSR    PC,(R1)
2766 012664 000240      NOP
2767 012666 000240      NOP
2768 012670 000240      NOP
2769 012672 106427 000340      RESTRT: MTPS   #340    ;PREVENT INTERRUPTS (PRIO: 7)
2770 012676 004767 000012      JSR    PC,CKSWR
2771 012702 012767 003364 166476      MOV    #TST1+2,$LPADR ;SET LAST ADDRESS POINTER
2772 012710 000167 170446      JMP    TST1
2773
2774      ;CHECK SWITCH REGISTER ROUTINE.
2775      ;CHECKS TO ALLOW FOR <↑G> TO ALLOW
2776      ;THE CHANGING OF LOCATION 176
2777
2778 012714 005737 000042      CKSWR: TST    #42
2779 012720 001040      BNE    OUT
2780 012722 022767 000176 166510      CMP    #SWREG,SWR     ;SOFTWARE SWR PRESENT?
2781 012730 001034      BNE    OUT            ;NO--LEAVE
2782 012732 105777 166506      TSTB  $STKS          ;CHECK TTY READY
2783 012736 100031      BPL    OUT            ;NO--LEAVE
2784 012740 017767 166502 000422      MOV    $STKB,MSG     ;GET CHARACTER
2785 012746 042767 177600 000414      BIC    #177600,.MSG  ;STRIP JUNK
2786 012754 122767 000007 000406      CMPB  #7,MSG        ;IS IT <↑G> ?
2787 012762 001017      BNE    OUT            ;NO
2788 012764 104401 016111      TYPE  MCNTG
2789 012770 005137 013030      CNTLU: COM   #RDSW
2790 012774 104401 016121      TYPE  ,MMSWR
2791 013000 104413      CONVRT
2792 013002 013032      SWREGL
2793 013004 104406 016132      INSTR,MMNEW
2794 013010 104410      PARAM
2795 013012 000000      0
2796 013014 177777      177777
2797 013016 000176      SWREG
  
```


E05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 58
 DZDUVA.M11 13-OCT-76 09:26 INITIALIZE THE COMMON TAGS

2798	013020	000	001		.BYTE	0,1		
2799	013022	005037	013030		OUT:	CLR	#RDSW	
2800	013026	000207				RTS	PC	
2801	013030	000000			RDSW:	.WORD	0	
2802	013032	000001			SWREGL:	1		
2803	013034	006	002			.BYTE	6.2	
2804	013036	000176				SWREG		
2805								
2806	013040	000005				5		
2807								
2808								;CHECK FOR FREEZE ON CURRENT DATA
2809								
2810	013042	004767	177646		.SCOPI:	JSR	PC,CKSWR	
2811	013046	032777	001000	16E364		BIT	#SW09,#SWR	
2812	013054	001402				BEQ	1\$	
2813	013056	016716	166026			MOV	LOCK,(SP)	
2814	013062	000002			1\$:	RTI		
2815					.SBTTL	TYPE	ROUTINE	
2816								
2817								
2818								*****
2819								*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2820								*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2821								*NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2822								*NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2823								*NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2824								*
2825								*CALL:
2826								*1) USING A TRAP INSTRUCTION
2827								* TYPE ,MESADR ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2828								*OR
2829								* TYPE
2830								* MESADR
2831								*
2832	013064	105767	166367		\$TYPE:	TSTB	\$TPFLG	;; IS THERE A TERMINAL?
2833	013070	100002				BPL	1\$;; BR IF YES
2834	013072	000000				HALT		;; HALT HERE IF NO TERMINAL
2835	013074	000430				BR	3\$;; LEAVE
2836	013076	010045			1\$:	MOV	RO,-(SP)	;; SAVE RO
2837	013100	017600	000002			MOV	#2(SP),RO	;; GET ADDRESS OF ASCIZ STRING
2838	013104	122767	000001	166434		CMPB	#APTENV,\$ENV	;; RUNNING IN APT MODE
2839	013112	001011				BNE	62\$;; NO GO CHECK FOR APT CONSOLE
2840	013114	132767	000100	166425		BITB	#APTSPool,\$ENVm	;; SPOOL MESSAGE TO APT
2841	013122	001405				BEQ	62\$;; NO GO CHECK FOR CONSOLE
2842	013124	010067	000004			MOV	RO,61\$;; SETUP MESSAGE ADDRESS FOR APT
2843	013130	004767	000006			JSR	PC,\$ATY3	;; SPOOL MESSAGE TO APT
2844	013134	000000			61\$:	.WORD	0	;; MESSAGE ADDRESS
2845	013136	132767	000040	166403	62\$:	BITB	#APTCSUP,\$ENVm	;; APT CONSOLE SUPPRESSED
2846	013144	001003				BNE	60\$;; YES, SKIP TYPE OUT
2847	013146	112046			2\$:	MOVB	(RO)+,-(SP)	;; PUSH CHARACTER TO BE TYPED ONTO STACK
2848	013150	001005				BNE	4\$;; BR IF IT ISN'T THE TERMINATOR
2849	013152	005726				TST	(SP)+	;; IF TERMINATOR POP IT OFF THE STACK
2850	013154	012600			60\$:	MOV	(SP)+,RO	;; RESTORE RO
2851	013156	062716	000002		3\$:	ADD	#2,(SP)	;; ADJUST RETURN PC
2852	013162	000002				RTI		;; RETURN
2853	013164	122716	000011		4\$:	CMPB	#HT,(SP)	;; BRANCH IF <HT>

```

2854 013170 001430      BEQ      8$
2855 013172 122716 000200  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
2856 013176 001006      BNE     5$
2857 013200 005726      TST     (SP)+          ;;POP <CR><LF> EQUIV
2858 013202 104401      TYPE    ;;TYPE A CR AND LF
2859 013204 001523      $CRLF
2860 013206 105067 000130  CLRB    $CHARCNT      ;;CLEAR CHARACTER COUNT
2861 013212 000755      BR      2$            ;;GET NEXT CHARACTER
2862 013214 004767 000056  5$:    JSR    PC,$TYPEC   ;;GO TYPE THIS CHARACTER
2863 013220 126726 166232  6$:    CMPB    $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
2864 013224 001350      BNE     2$            ;;IF NO GO GET NEXT CHAR.
2865 013226 016746 166222  MOV     $NULL,-(SP)   ;;GET # OF FILLER CHARS. NEEDED
2866                                ;;AND THE NULL CHAR.
2867 013232 105366 000001  7$:    DECB    1(SP)    ;;DOES A NULL NEED TO BE TYPED?
2868 013236 002770      BLT     6$            ;;BR IF NO--GO POP THE NULL OFF OF STACK
2869 013240 004767 000032  JSR    PC,$TYPEC   ;;GO TYPE A NULL
2870 013244 105367 000072  DECB    $CHARCNT    ;;DO NOT COUNT AS A COUNT
2871 013250 000770      BR      7$            ;;LOOP
2872
2873                                ;HORIZONTAL TAB PROCESSOR
2874
2875 013252 112716 000040  8$:    MOVB    #' (SP)   ;;REPLACE TAB WITH SPACE
2876 013256 004767 000014  9$:    JSR    PC,$TYPEC   ;;TYPE A SPACE
2877 013262 132767 000007 000052  BITB    #',$CHARCNT  ;;BRANCH IF NOT AT
2878 013270 001372      BNE     9$            ;;TAB STOP
2879 013272 005726      TST     (SP)+          ;;POP SPACE OFF STACK
2880 013274 000724      BR      2$            ;;GET NEXT CHARACTER
2881 013276 105777 166146  $TYPEC: TSTB    2$TPS     ;;WAIT UNTIL PRINTER IS READY
2882 013302 100375      BPL     $TYPEC
2883 013304 116677 000002 166140  MOVB    2(SP),2$TPB   ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2884 013312 122766 000015 000002  CMPB    #CR,2(SP)    ;;IS CHARACTER A CARRIAGE RETURN?
2885 013320 001003      BNE     1$            ;;BRANCH IF NO
2886 013322 105067 000014      CLRB    $CHARCNT    ;;YES--CLEAR CHARACTER COUNT
2887 013326 000406      BR      $TYPEX      ;;EXIT
2888 013330 122766 000012 000002  1$:    CMPB    #LF,2(SP) ;;IS CHARACTER A LINE FEED?
2889 013336 001402      BEQ     $TYPEX      ;;BRANCH IF YES
2890 013340 105227      INCB    (PC)+        ;;COUNT THE CHARACTER
2891 013342 000000  $CHARCNT: .WORD    0 ;;CHARACTER COUNT STORAGE
2892 013344 000207  $TYPEX: RTS     PC
2893
2894                                ;ASCII STRING INPUT ROUTINE
2895
2896
2897 013346 017667 000000 000014 .INSTR: MOV     2(SP),MSG ;;PICK UP MESSAGE
2898 013354 062716 000002      ADD     #2,(SP)      ;;JUMP AROUND MESSAGE FOR RTI
2899 013360 105767 166162      TSTB    $ENV        ;;APT CONTROL
2900 013364 001036      BNE     INSTR2      ;;YES NO TYPE
2901 013366 104401      .INST1: TYPE
2902 013370 000000      .MSG: 0
2903 013372 012704 016144      MOV     #INBUF,R4   ;;GET STARTING LOC OF INBUF
2904 013376 012703 000007      MOV     #7,R3       ;;MAX # OF CHARS
2905 013402 105777 166036  1$:    TSTB    2$TKS     ;TTY FLAG
2906 013406 100375      BPL     1$
2907 013410 117714 166032      MOVB    2$TKB,(R4)  ;;TAKE CHAR
2908 013414 142714 000200      BICB    #200,(R4)  ;;STRIP
2909 013420 121427 000025      CMPB    (R4),#25   ;;IS IT <G>
    
```

G05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 60
 DZDUVA.M11 13-OCT-76 08:26 TYPE ROUTINE

```

2910 013424 001760      BEQ      .INST1
2911 013426 122427 000015  CMPB    (R4)+,#15      ;CHECK FOR CR
2912 013432 001413      BEQ      INSTR2
2913 013434 105777 166010      2$:    TSTB    @STPS      ;TEST FLAG
2914 013440 100375      BPL     2$
2915 013442 117777 166000 166002  MOVB   @STKB,@STPB    ;ECHO CHARACTER
2916 013450 005303      DEC     R3            ;DID YOU TYPE TOO MANY CHARS ?
2917 013452 001353      BNE     1$
2918 013454 104401      .INSTE: TYPE
2919 013456 015431      MQM
2920 013460 000742      BR     .INST1 ;RETRY
2921 013462 000002      INSTR2: RTI
2922
2923                ;CONVERT ASCII STRING TO OCTAL
2924
2925 013464 011605      .PARAM: MOV    (SP),R5 ;PUT CONTENTS OF SP INTO R5
2926 013466 012567 000162  MOV    (R5)+,LOLIM   ;PUT LOW LIMIT INTO LOLIM
2927 013472 012567 000160  MOV    (R5)+,HILIM   ;PUT HIGH LIMIT INTO HILIM
2928 013476 012567 000156  MOV    (R5)+,DEVADR   ;PUT STORE LOC INTO DEVADR
2929 013502 112567 000154  MOVB   (R5)+,LOBITS   ;PUT MASK INTO LOBITS
2930 013506 112567 000151  MOVB   (R5)+,ADRCNT   ;PUT COUNT INTO ADRCNT
2931 013512 010516      MOV    R5,(SP) ;RESTORE RETURN ADDR ON STACK FOR RTI
2932 013514 005005      PARAM1: CLR    R5
2933 013516 012704 016144  MOV    #INBUF,R4
2934 013522 122714 000015  CMPB   #15,(R4)      ;CR ?
2935 013526 001420      BEQ    PARERR ;YOU TYPED CR TOO SOON !
2936 013530 121427 000060  1$:    CMPB   (R4),#60      ;LOW LIMIT ASCII 0
2937 013534 002415      BLT    PARERR
2938 013536 121427 000067  CMPB   (R4),#67      ;HIGH LIMIT ASCII 7
2939 013542 003012      BGT    PARERR
2940 013544 142714 000060  BICB   #60,(R4)      ;CONVERT TO OCTAL
2941 013550 152405      BISB   (R4)+,R5      ;STORE AWAY ITS AN OK CHAR
2942 013552 122714 000015  CMPB   #15,(R4)      ;CR ?
2943 013556 001414      BEQ    LIMITS ;NOW CHECK FOR HIGH &LOW LIMIT CONDS
2944 013560 006305      ASL    R5            ;ALLOCATE ROOM FOR NEXT CHAR
2945 013562 006305      ASL    R5
2946 013564 006305      ASL    R5
2947 013566 000760      BR     1$
2948 013570 122714 000015  PARERR: CMPB   #15,(R4) ;CR?
2949 013574 001003      BNE    120$
2950 013576 005737 013030  TST    @RDSW        ;CK SWR USED
2951 013602 001023      BNE    PARTI
2952 013604 104407      120$: INSTR  ;RETRY
2953 013606 000742      BR     PARAM1
2954
2955                ;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2956
2957 013610 020567 000042  LIMITS: CMP    R5,HILIM
2958 013614 101365      BHI    PARERR ;THE # IS TOO HIGH
2959 013616 020567 000032  CMP    R5,LOLIM
2960 013622 103762      BLO    PARERR ;THE # IS TOO LOW
2961 013624 136705 000032  BITB   LOBITS,R5    ;TEST BY MASKINGTHE #
2962 013630 001357      BNE    PARERR
2963
2964                ;STORE NUMBER AT SPECIFIED ADDRESS
2965

```

H05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 61
 DZDUVA.M11 13-OCT-76 08:26 TYPE ROUTINE

```

2966 013632 016704 000022          MOV    DEVADR,R4          ;GET STARTING ADDR OF
2967 013636 010524          1$:   MOV    R5,(R4)+      ;STORE AT THIS ADDR
2968 013640 062705 000002          ADD    #2,R5
2969 013644 105367 000013          DECB   ADRCNT ;HOW MANY TIMES + 2 ?
2970 013650 001372          BNE    1$
2971 013652 000002          PARTI: RTI
2972 013654 000000          LOLIM: 0
2973 013656 000000          HILIM: 0
2974 013660 000000          DEVADR: 0
2975 013662 000000          LOBITS: 0
2976          013663          ADRCNT=LOBITS+1
2977
2978          ;SAVE PC OF TEST THAT FAILED AND RO-R5
2979
2980 013664 016667 000004 165234 .SAV05: MOV    4(SP),SAVPC
2981
2982          ;SAVE RO-R5
2983
2984 013672 010567 165576          SV05:  MOV    R5,$REG5
2985 013676 010467 165570          MOV    R4,$REG4
2986 013702 010367 165562          MOV    R3,$REG3
2987 013706 010267 165554          MOV    R2,$REG2
2988 013712 010167 165546          MOV    R1,$REG1
2989 013716 010067 165540          MOV    R0,$REG0
2990 013722 000002          RTI
2991
2992          ;RESTORE RO-R5
2993
2994 013724 016700 165532          .RES05: MOV    $REG0,R0
2995 013730 016701 165530          MOV    $REG1,R1
2996 013734 016702 165526          MOV    $REG2,R2
2997 013740 016703 165524          MOV    $REG3,R3
2998 013744 016704 165522          MOV    $REG4,R4
2999 013750 016705 165520          MOV    $REG5,R5
3000 013754 000002          RTI
3001
3002          ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
3003
3004 013756 104401          .CONVR: TYPE
3005 013760 015435          MCRLF   ;CR LF
3006 013762 017601 000000          MOV    2(SP),R1          ;PICK UP DATA POINTER
3007 013766 062716 000002          ADD    #2(SP) ;SET UP SP FOR RTI
3008 013772 012167 000130          MOV    (R1)+,WRDCNT      ;PICK UP # OF WORDS FROM TABLE
3009 013776 112167 000126          1$:   MOV    (R1)+,CHRCNT      ;PICK UP # OF CHARS FROM TABLE
3010 014002 112167 000123          MOV    (R1)+,SPACNT      ;PICK UP # OF SPACES FROM TABLE
3011 014006 013167 000120          MOV    2(R1)+,BINWRD     ;PICK UP ADDRESS OF MSG
3012          ;FROM TABLE
3013 014012 016704 000114          2$:   MOV    BINWRD,R4          ;SAVE
3014 014016 116705 000106          MOV    CHRCNT,R5          ;SAVE
3015 014022 012700 016206          MOV    #TEMP,R0          ;STARTING ADDRESS OF TEMP BLOCK
3016 014026 010403          3$:   MOV    R4,R3          ;SAVE
3017 014030 042703 177770          BIC    #177770,R3          ;CLR OUT UPPER BITS .. SAVE CHAR
3018 014034 062703 000260          ADD    #260,R3 ;CONVERT TO ASCII
3019 014040 110320          MOV    R3,(R0)+          ;STORE AWAY
3020 014042 006204          ASR    R4          ;SHIFT FOR NEXT #
3021 014044 006204          ASR    R4          ;DITTO

```

```

3022 014046 006204      ASP      R4      ;DITTO
3023 014050 005305      DEC      R5      ;DEC CHAR COUNT
3024 014052 001365      BNE      3$      ;DO IT AGAIN ?
3025 014054 012703 016250  MOV      #MDATA,R3 ;STARTING ADDRESS OF MDATA BLOCK
3026 014060 114023      4$: MOVB  -(R0),(R3)+ ;REVERSE THE ORDER OF NUMBERS
3027 014062 105367 000042  DECB    CHRCNT ;DEC CHAR COUNT
3028 014066 001374      BNE      4$      ;DO IT AGAIN ?
3029 014070 105767 000035  TSTB    SPACNT ;HOW MANY SPACES ?
3030 014074 001405      BEQ      6$      ;TYPE # IF BR =0
3031 014076 112723 000240  5$: MOVB  #240,(R3)+ ;"SPACE" IN ASCII
3032 014102 105367 000023  DECB    SPACNT ;DEC # OF SPACE COUNT
3033 014106 001373      BNE      5$      ;DO IT AGAIN ?
3034 014110 105013      6$: CLRB  (R3)    ;INSERT "0" FOR TTY OUTPUT ROUTINE
3035 014112 104401      TYPE
3036 014114 016250      MDATA    ;THIS MESSAGE
3037 014116 005367 000004  DEC     WRDCNT ;HOW MANY #'S ?
3038 014122 001325      BNE     1$      ;DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3039 014124 000002      RTI     ;RETURN TO PROGRAM
3040 014126 000000      WRDCNT: 0
3041 014130 000000      CHRCNT: 0
3042 014131 014131      SPACNT=CHRCNT+1
3043 014132 000000      BINWRD: 0
3044
3045
3046 ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3047 ;BUFFER TO THE CHARACTERS "N" AND "Y".
3048 ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3049 ;IF THE CHARACTER IS "Y" SET THE FLAG
3050 014134 017605 000000  .SETFLG: MOV  2(SP),R5
3051 014140 122767 000116 001776  CMPB   #'N,INBUF ;IS IT "N" ?
3052 014146 001002      BNE     1$
3053 014150 105015      CLRB   (R5) ;000
3054 014152 000406      BR     2$
3055 014154 122767 000131 001762  1$: CMPB  #'Y,INBUF ;IS IT "Y" ?
3056 014162 001005      BNE     3$
3057 014164 112715 177777  MOVB   #-1,(R5) ;377
3058 014170 062716 000002  2$: ADD   #2,(SP)
3059 014174 000002      RTI
3060 014176 104407      3$: INSTER ;RETRY
3061 014200 000755      BR     .SETFLG
3062 .SBTTL  ERROR HANDLER ROUTINE
3063
3064 ;*****
3065 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3066 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3067 ;*AND GO TO SAVIT ON ERROR
3068 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3069 ;*SW15=1      HALT ON ERROR
3070 ;*SW13=1      INHIBIT ERROR TYPEOUTS
3071 ;*SW10=1      BELL ON ERROR
3072 ;*SW09=1      LOOP ON ERROR
3073 ;*CALL
3074 ;*      ERROR  N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3075
3076 014202      $ERROR:
3077 014202 105267 165175  7$: INCB  $ERFLG      ;;SET THE ERROR FLAG
    
```

```

3078 014206 001775          BEQ      7$
3079 014210 016777 165166 165224  MOV     $STNM,$DISPLAY ;: DON'T LET THE FLAG GO TO ZERO
3080 014216 032777 002000 165214  BIT     #BIT10,$SWR ;: DISPLAY TEST NUMBER AND ERROR FLAG
3081 014224 001402          BEQ     1$ ;: BELL ON ERROR?
3082 014226 104401 001516          TYPE   $BELL ;: NO - SKIP
3083 014232 005267 165154 1$: INC     $ERTTL ;: RING BELL
3084 014236 011667 165154          MOV    (SP),$ERRPC ;: COUNT THE NUMBER OF ERRORS
3085 014242 162767 000002 165146  SUB     #2,$ERRPC ;: GET ADDRESS OF ERROR INSTRUCTION
3086 014250 117767 165142 165136  MOVB   $ERRPC,$ITEMB ;: STRIP AND SAVE THE ERROR ITEM CODE
3087 014256 032777 020000 165154  BIT     #BIT13,$SWR ;: SKIP TYPEOUT IF SET
3088 014264 001004          BNE    20$ ;: SKIP TYPEOUTS
3089 014266 004767 000072          JSR    PC,SAVIT ;: GO TO USER ERROR ROUTINE
3090 014272 104401 001523          TYPE   $CRLF
3091 014276
3092 014276 122767 000001 165242 20$: CMPB   #APTENV,$ENV ;: RUNNING IN APT MODE
3093 014304 001007          BNE    2$ ;: NO SKIP APT ERROR REPORT
3094 014306 116767 165102 000004  MOVB   $ITEMB,21$ ;: SET ITEM NUMBER AS ERROR NUMBER
3095 014314 004767 000016          JSR    PC,$ATY4 ;: REPORT FATAL ERROR TO APT
3096 014320          000
3097 014321          000
3098 014322 000777          .BYTE 0
3099 014324 005777 165110 22$: BR     22$ ;: APT ERROR LOOP
3100 014330 100001          .BYTE 0 ;: HALT ON ERROR
3101 014332 000000          .BYTE 0 ;: SKIP IF CONTINUE
3102 014334 032777 001000 165076 3$: HALT ;: HALT ON ERROR!
3103 014342 001402          BIT     #BIT09,$SWR ;: LOOP ON ERROR SWITCH SET?
3104 014344 016716 165040          BEQ    4$ ;: BR IF NO
3105 014350 005767 165140          MOV    $LPERR,(SP) ;: FUDGE RETURN FOR LOOPING
3106 014354 001402          TST   $ESCAPE ;: CHECK FOR AN ESCAPE ADDRESS
3107 014356 016716 165132          BEQ    5$ ;: BR IF NONE
3108 014362          MOV    $ESCAPE,(SP) ;: FUDGE RETURN ADDRESS FOR ESCAPE
3109 014362 000002          5$: RTI ;: RETURN
3110 014364 010067 164540 SAVIT: MOV    R0,HL00
3111 014370 010167 164536          MOV    R1,HL01
3112 014374 010267 164534          MOV    R2,HL02
3113 014400 010367 164532          MOV    R3,HL03
3114 014404 010467 164530          MOV    R4,HL04
3115 014410 010567 164526          MOV    R5,HL05
3116 014414 016767 164762 164522  MOV    $STNM,HL06

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

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

$ERRTYP:
3125 014422          TYPE   $CRLF ;: "CARRIAGE RETURN" & "LINE FEED"
3126 014422 104401 001523  MOV     R0,-(SP) ;: SAVE R0
3127 014426 010046          CLR    R0 ;: PICKUP THE ITEM INDEX
3128 014430 005000          BISB  #2,$ITEMB,R0
3129 014432 153700 001414  BNE    1$ ;: IF ITEM NUMBER IS ZERO, JUST
3130 014436 001004          ;: TYPE THE PC OF THE ERROR
3131          ;: SAVE $ERRPC FOR TYPEOUT
3132 014440 016746 164752  MOV     $ERRPC,-(SP) ;: ERROR ADDRESS
3133

```

```

3134 014444 104402          TYP0C          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
3135 014446 000426          BR           6$          ;; GET OUT
3136 014450 005300          1$: DEC        RO          ;; ADJUST THE INDEX SO THAT IT WILL
3137 014452 006300          ASL        RO          ;; WORK FOR THE ERROR TABLE
3138 014454 006300          ASL        RO
3139 014456 006300          ASL        RO
3140 014460 062700 001652    ADD        #SERRTB,RO    ;; FORM TABLE POINTER
3141 014464 012067 000004    MOV        (RO)+,2$    ;; PICKUP "ERROR MESSAGE" POINTER
3142 014470 001404          BEQ        3$          ;; SKIP TYPEOUT IF NO POINTER
3143 014472 104401          TYPE          ;; TYPE THE "ERROR MESSAGE"
3144 014474 000000          2$: .WORD      0          ;; "ERROR MESSAGE" POINTER GOES HERE
3145 014476 104401 001523    TYPE      $CRLF        ;; "CARRIAGE RETURN" & "LINE FEED"
3146 014502 012067 000004    3$: MOV        (RO)+,4$    ;; PICKUP "DATA HEADER" POINTER
3147 014506 001404          BEQ        5$          ;; SKIP TYPEOUT IF 0
3148 014510 104401          TYPE          ;; TYPE THE "DATA HEADER"
3149 014512 000000          4$: .WORD      0          ;; "DATA HEADER" POINTER GOES HERE
3150 014514 104401 001523    TYPE      $CRLF        ;; "CARRIAGE RETURN" & "LINE FEED"
3151 014520 011000          5$: MOV        (RO),RO    ;; PICKUP "DATA TABLE" POINTER
3152 014522 001004          BNE        7$          ;; GO TYPE THE DATA
3153 014524 012600          6$: MOV        (SP)+,RO    ;; RESTORE RO
3154 014526 104401 001523    TYPE      $CRLF        ;; "CARRIAGE RETURN" & "LINE FEED"
3155 014532 000207          RTS        PC          ;; RETURN
3156 014534
3157 014534 013046          7$: MOV        2(RO)+,-(SP) ;; SAVE 2(RO)+ FOR TYPEOUT
3158 014536 104402          TYP0C          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
3159 014540 005710          TST        (RO)        ;; IS THERE ANOTHER NUMBER?
3160 014542 001770          BEQ        6$          ;; BR IF NO
3161 014544 104401 014552    TYPE      2$          ;; TYPE TWO(2) SPACES
3162 014550 000771          BR         7$          ;; LOOP
3163 014552 020040 000          8$: .ASCIZ    / /          ;; TWO(2) SPACES
3164 014556
3165          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3166
3167          ;*****
3168          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3169          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3170          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3171          ;*CALL:
3172          ;*      MOV      NUM,-(SP)          ;; NUMBER TO BE TYPED
3173          ;*      TYPOS          ;; CALL FOR TYPEOUT
3174          ;*      .BYTE  N          ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3175          ;*      .BYTE  M          ;; M=1 OR 0
3176          ;*
3177          ;*          ;; 1=TYPE LEADING ZEROS
3178          ;*          ;; 0=SUPPRESS LEADING ZEROS
3179          ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3180          ;*$TYPOS OR $TYPOC
3181          ;*CALL:
3182          ;*      MOV      NUM,-(SP)          ;; NUMBER TO BE TYPED
3183          ;*      TYPON          ;; CALL FOR TYPEOUT
3184          ;*
3185          ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3186          ;*CALL:
3187          ;*      MOV      NUM,-(SP)          ;; NUMBER TO BE TYPED
3188          ;*      TYP0C          ;; CALL FOR TYPEOUT
3189

```

L05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 65
 DZDUVA.M11 13-OCT-76 08:26 BINARY TO OCTAL (ASCII) AND TYPE

3190	014556	017646	000000		\$TYPOS:	MOV	2(SP),-(SP)	:: PICKUP THE MODE
3191	014562	116667	000001	000211		MOV	1(SP),\$OFILL	:: LOAD ZERO FILL SWITCH
3192	014570	112667	000207			MOV	(SP)+,\$SOMODE+1	:: NUMBER OF DIGITS TO TYPE
3193	014574	062716	000002			ADD	#2,(SP)	:: ADJUST RETURN ADDRESS
3194	014600	000406				BR	\$TYPON	
3195	014602	112767	000001	000171	\$TYPOC:	MOV	#1,\$OFILL	:: SET THE ZERO FILL SWITCH
3196	014610	112767	000006	000165		MOV	#6,\$SOMODE+1	:: SET FOR SIX(6) DIGITS
3197	014616	112767	000005	000154	\$TYPON:	MOV	#5,\$SOCNT	:: SET THE ITERATION COUNT
3198	014624	010346				MOV	R3,-(SP)	:: SAVE R3
3199	014626	010446				MOV	R4,-(SP)	:: SAVE R4
3200	014630	010546				MOV	R5,-(SP)	:: SAVE R5
3201	014632	116704	000145			MOV	\$SOMODE+1,R4	:: GET THE NUMBER OF DIGITS TO TYPE
3202	014636	005404				NEG	R4	
3203	014640	062704	000006			ADD	#6,R4	:: SUBTRACT IT FOR MAX. ALLOWED
3204	014644	110467	000132			MOV	R4,\$SOMODE	:: SAVE IT FOR USE
3205	014650	116704	000125			MOV	\$OFILL,R4	:: GET THE ZERO FILL SWITCH
3206	014654	016605	000012			MOV	12(SP),R5	:: PICKUP THE INPUT NUMBER
3207	014660	005003				CLR	R3	:: CLEAR THE OUTPUT WORD
3208	014662	006105			1\$:	ROL	R5	:: ROTATE MSB INTO "C"
3209	014664	000404				BR	3\$:: GO DO MSB
3210	014666	006105			2\$:	ROL	R5	:: FORM THIS DIGIT
3211	014670	006105				ROL	R5	
3212	014672	006105				ROL	R5	
3213	014674	010503				MOV	R5,R3	
3214	014676	006103			3\$:	ROL	R3	:: GET LSB OF THIS DIGIT
3215	014700	105367	000076			DECB	\$SOMODE	:: TYPE THIS DIGIT?
3216	014704	100016				BPL	7\$:: BR IF NO
3217	014706	042703	177770			BIC	#177770,R3	:: GET RID OF JUNK
3218	014712	001002				BNE	4\$:: TEST FOR 0
3219	014714	005704				TST	R4	:: SUPPRESS THIS 0?
3220	014716	001403				BEQ	5\$:: BR IF YES
3221	014720	005204			4\$:	INC	R4	:: DON'T SUPPRESS ANYMORE 0'S
3222	014722	052703	000060			BTS	#'0,R3	:: MAKE THIS DIGIT ASCII
3223	014726	052703	000040		5\$:	BIS	#'R3	:: MAKE ASCII IF NOT ALREADY
3224	014732	110367	000040			MOV	R3,8\$:: SAVE FOR TYPING
3225	014736	104401	014776			TYPE	8\$:: GO TYPE THIS DIGIT
3226	014742	105367	000032		7\$:	DECB	\$SOCNT	:: COUNT BY 1
3227	014746	003347				BGT	2\$:: BR IF MORE TO DO
3228	014750	002402				BLT	6\$:: BR IF DONE
3229	014752	005204				INC	R4	:: INSURE LAST DIGIT ISN'T A BLANK
3230	014754	000744				BR	2\$:: GO DO THE LAST DIGIT
3231	014756	012605			6\$:	MOV	(SP)+,R5	:: RESTORE R5
3232	014760	012604				MOV	(SP)+,R4	:: RESTORE R4
3233	014762	012603				MOV	(SP)+,R3	:: RESTORE R3
3234	014764	016666	000002	000004		MOV	2(SP),4(SP)	:: SET THE STACK FOR RETURNING
3235	014772	012616				MOV	(SP)+,(SP)	
3236	014774	000002				RTI		:: RETURN
3237	014776	000			8\$:	.BYTE	0	:: STORAGE FOR ASCII DIGIT
3238	014777	000				.BYTE	0	:: TERMINATOR FOR TYPE ROUTINE
3239	015000	000			\$SOCNT:	.BYTE	0	:: OCTAL DIGIT COUNTER
3240	015001	000			\$OFILL:	.BYTE	0	:: ZERO FILL SWITCH
3241	015002	000000			\$SOMODE:	.WORD	0	:: NUMBER OF DIGITS TO TYPE
3242								:: ENTER HERE ON POWER FAILURE
3243								
3244								
3245	015004				\$PWRDN:			

M05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 66
 DZDUVA.M11 13-OCT-76 08:26 BINARY TO OCTAL (ASCII) AND TYPE

3246	015004	010046			.PFAIL: MOV	R0,-(SP)		;SAVE R0-R5 ON PROCESSOR STACK
3247	015006	010146			MOV	R1,-(SP)		
3248	015010	010246			MOV	R2,-(SP)		
3249	015012	010346			MOV	R3,-(SP)		
3250	015014	010446			MOV	R4,-(SP)		
3251	015016	010546			MOV	R5,-(SP)		
3252	015020	016746	163000		MOV	24,-(SP)		
3253	015024	010667	164066		MOV	SP,SAVSP		;SAVE STACK POINTER
3254	015030	012767	015042	162766	MOV	#RESTART,24		;SET UP FOR POWER UP TRAP
3255	015036	000000			HALT			;HALT ON POWER DOWN NORMAL
3256	015040	000777			BR	.		
3257								
3258								;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3259								
3260	015042	016706	164050		RESTAR: MOV	SAVSP,SP		;RESTORE STACK POINTER
3261	015046	012605			MOV	(SP)+,R5		;RESTORE R0-R5
3262	015050	012604			MOV	(SP)+,R4		
3263	015052	012603			MOV	(SP)+,R3		
3264	015054	012602			MOV	(SP)+,R2		
3265	015056	012601			MOV	(SP)+,R1		
3266	015060	012600			MOV	(SP)+,R0		
3267	015062	012767	015004	162734	MOV	#.PFAIL,24		;SET UP FOR POWER FAILURE
3268	015070	106427	000340		MTPS	#340		
3269	015074	012706	001100		MOV	#STACK,SP		
3270	015100	005067	001102		CLR	TEMP		
3271	015104	005267	001076		INC	TEMP		
3272	015110	001375			BNE	.-4		
3273	015112	104413			CONVRT			
3274	015114	015136			PFTAB			
3275	015116	104401			TYPE			
3276	015120	015440			MPFAIL			
3277	015122	005067	164255		CLR	\$ERFLG		
3278	015126	005067	164264		CLR	\$ERRPC		
3279	015132	000177	163746		JMP	\$RETURN		
3280	015136	000001			PFTAB:	1		
3281	015140	006	002		.BYTE	6,2		
3282	015142	000207			RETURN			
3283	015144	005015	042012	053125	MTITLE: .ASCIZ	<15><12><12>/DUV11 DZDUV-A TAPE F /<15><12>		
3284	015152	030461	042040	042132				
3285	015160	053125	040455	052040				
3286	015166	050101	020105	020106				
3287	015174	005015	000					
3288	015177	015	053012	041505	MVECTO: .ASCIZ	<15><12>/VEC ADD- /		
3289	015204	040440	042104	000055				
3290	015212	005015	051461	020124	MREGAD: .ASCIZ	<15><12>/1ST DEV: REC CSR ADD- /		
3291	015220	042504	035126	051040				
3292	015226	041505	041440	051123				
3293	015234	040440	042104	000055				
3294	015242	005015	052515	052114	MMULT: .ASCIZ	<15><12>/MULT DEV ? (Y OR N)- /		
3295	015250	042040	053105	037440				
3296	015256	024040	020131	051117				
3297	015264	047040	026451	000				
3298	015271	015	046012	051501	MLASTD: .ASCIZ	<15><12>/LAST DEV: REC CSR ADDR- /		
3299	015276	020124	042504	035126				
3300	015304	051040	041505	041440				
3301	015312	051123	040440	042104				

N05

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 67
 DZDUVA.M11 13-OCT-76 08:26 BINARY TO OCTAL (ASCII) AND TYPE

3302	015320	026522	000		
3303	015323	075	042504	044526	DEVICE: .ASCIZ /=DEVICE /
3304	015330	042503	020040	000	
3305	015335	015	051412	046105	MCOW: .ASCIZ <15><12>/SELECT TO RUN DACTREG/
3306	015342	041505	020124	047524	
3307	015350	051040	047125	040040	
3308	015356	041501	051124	043505	
3309	015364	000			
3310	015365	015	047412	043126	MRANGE: .ASCIZ <15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS-/
3311	015372	047514	051072	052105	
3312	015400	050131	020105	040514	
3313	015406	052123	042040	053105	
3314	015414	051040	041530	051123	
3315	015422	040440	042104	026523	
3316	015430	000			
3317	015431	040	037440	000	MQM: .ASCIZ / ?/
3318	015435	015	000012		MCRLF: .ASCIZ <15><12>
3319	015440	043120	044501	026114	MPFAIL: .ASCIZ /PFAIL, RESTART AT TEST IN PROGRESS/
3320	015446	020040	042522	052123	
3321	015454	051101	020124	052101	
3322	015462	052040	051505	020124	
3323	015470	047111	050040	047522	
3324	015476	051107	051505	000123	
3325	015504	005015	047105	020104	MEPASS: .ASCIZ <15><12>/END OF PASS TAPE F/
3326	015512	043117	050040	051501	
3327	015520	020123	040524	042520	
3328	015526	043040	000		
3329	015531	015	051012	000	MR: .ASCIZ <15><12>/R/
3330	015535	015	052012	051505	MTSTPC: .ASCIZ <15><12>/TEST PC-/
3331	015542	020124	041520	000055	
3332	015550	005015	047514	045503	MLOCK: .ASCIZ <15><12>/LOCK ON TEST? (Y OR N)-/
3333	015556	047440	020116	052040	
3334	015564	051505	037524	024040	
3335	015572	020131	051117	047040	
3336	015600	026451	000		
3337	015603	015	021412	047440	MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3338	015610	020106	054523	041516	
3339	015616	041440	040510	051522	
3340	015624	051440	046105	041505	
3341	015632	042524	020104	020050	
3342	015640	020061	051117	031040	
3343	015646	026451	000		
3344	015651	015	044412	020123	MWIRE6: .ASCIZ <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3345	015656	042523	020103	046530	
3346	015664	052111	051440	044527	
3347	015672	041524	020110	032505	
3348	015700	026465	020062	047111	
3349	015706	020077	054450	047440	
3350	015714	020122	024516	000055	
3351	015722	005015	051511	051440	MWIRES: .ASCIZ <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3352	015730	041505	051040	041505	
3353	015736	051440	044527	041524	
3354	015744	020110	032505	026465	
3355	015752	020063	047111	020077	
3356	015760	054450	047440	020122	
3357	015766	024516	000055		

```

3358 015772 005015 051511 047440 MWIRE4: .ASCIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3359 016000 052120 041440 051114
3360 016006 042440 040516 046102
3361 016014 020105 053523 052111
3362 016022 044103 042440 032465
3363 016030 030455 044440 037516
3364 016036 024040 020131 051117
3365 016044 047040 026451 000
3366 016051 015 005012 031510 MEXTJ: .ASCIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-
3367 016056 032461 041440 047117
3368 016064 042516 052103 051117
3369 016072 047440 020116 024077
3370 016100 020131 051117 047040
3371 016106 026451 000
3372 016111 015 020012 043536 MCNTG: .ASCIZ <15><12>/ IG /
3373 016116 020040 000
3374 016121 040 053523 036522 MMSWR: .ASCIZ / SWR= /
3375 016126 020040 000040
3376 016132 020040 047040 053505 MMNEW: .ASCIZ / NEW= /
3377 016140 020075 000040
3378 .EVEN
3379
3380 ;BUFFERS FOR INPUT-OUTPUT
3381
3382 016144 000000 INBUF: 0
3383 016206 016206 .=. +40
3384 016206 000000 TEMP: 0
3385 016250 016250 .=. +40
3386 016250 000000 MDATA: 0
3387 016312 016312 .=. +40
3388 .SBTTL SCOPE HANDLER ROUTINE
3389
3390 ;*****
3391 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3392 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3393 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3394 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3395 ;*SW14=1 LOOP ON TEST
3396 ;*SW11=1 INHIBIT ITERATIONS
3397 ;*SW09=1 LOOP ON ERROR
3398 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
3399 ;*CALL
3400 ;* SCOPE ;;SCOPE=IOT
3401
3402 016312 $$SCOPE:
3403
3404 ;SCOPE LOOP AND INTERATION HANDLER
3405
3406 .SCOPE:
3407 016312 004767 174376 JSR PC_CKSWR
3408 016316 005067 163074 CLR $ERRPC ;CLEAR LAST ERROR PC
3409 016322 022716 003364 CMP #TST1+2,(SP) ;IS SCOPE AT BEGINING OF TEST 1?
3410 016326 001413 BEQ $XTSTR ;YES NO LOOP.
3411
3412 016330 000406 TTST: BR 1$ ;GO TO 1$ (IF LOCK SW02=1)
3413 016332 105777 163106 TSTB $T$KS ;KEYBOARD DONE?

```

```

3414 016336 100123          BPL      $OVER      ;BR IF NO
3415 016340 017766 163102 177776  MOV      @STKB,-2(SP) ;CLEAR DONE BIT
3416 016346 032777 040000 163064 1$:      BIT      @BIT14,@SWR ;LOOP ON PRESENT TEST?
3417 016354 001114          BNE      $OVER      ;YES IF SW14=1
3418          ;*****START OF CODE FOR THE XOR TESTER*****
3419 016356 000416          $XTSTR: BR      L$      ;IF RUNNING ON THE "XOR" TESTER CHANGE
3420          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
3421 016360 013746 000004          MOV      @ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3422 016364 012737 016404 000004  MOV      @SS,@ERRVEC  ;SET FOR TIMEOUT
3423 016372 005737 177060          TST      @177060      ;TIME OUT ON XOR?
3424 016376 012637 000004          MOV      (SP)+,@ERRVEC ;RESTORE THE ERROR VECTOR
3425 016402 000463          BR      $SVLAD       ;GO TO THE NEXT TEST
3426 016404 022626          5$:      CMP      (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
3427 016406 012637 000004          MOV      (SP)+,@ERRVEC ;RESTORE THE ERROR VECTOR
3428 016412 000423          BR      7$          ;LOOP ON THE PRESENT TEST
3429 016414          6$:;*****END OF CODE FOR THE XOR TESTER*****
3430 016414 032777 000400 163016  BIT      @BIT08,@SWR  ;LOOP ON SPEC. TEST?
3431 016422 001404          BEQ      2$          ;BR IF NO
3432 016424 127767 163010 162750  CMPB     @SWR,$STNM   ;ON THE RIGHT TEST? SWR<7:0>
3433 016432 001465          BEQ      $OVER      ;BR IF YES
3434 016434 105767 162743          2$:      TSTB     $ERFLG  ;HAS AN ERROR OCCURRED?
3435 016440 001421          BEQ      3$          ;BR IF NO
3436 016442 126767 162747 162733  CMPB     $ERMAX,$ERFLG ;MAX. ERRORS FOR THIS TEST OCCURRED?
3437 016450 101015          BHI      3$          ;BR IF NO
3438 016452 032777 001000 162750  BIT      @BIT09,@SWR ;LOOP ON ERROR?
3439 016460 001404          BEC      4$          ;BR IF NO
3440 016462 016767 162722 162716  7$:      MOV      $LPERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
3441 016470 000446          BR      $OVER      ;
3442 016472 105067 162705          4$:      CLRB     $ERFLG  ;ZERO THE ERROR FLAG
3443 016476 005067 163010          CLR      $TIMES     ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
3444 016502 000415          BR      1$          ;ESCAPE TO THE NEXT TEST
3445 016504 032777 004000 162726  3$:      BIT      @BIT11,@SWR ;INHIBIT ITERATIONS?
3446 016512 001011          BNE      1$          ;BR IF YES
3447 016514 005767 163014          TST      $PASS      ;IF FIRST PASS OF PROGRAM
3448 016520 001406          BEQ      1$          ;INHIBIT ITERATIONS
3449 016522 005267 162656          INC      $ICNT      ;INCREMENT ITERATION COUNT
3450 016526 026767 162760 162650  CMP      $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
3451 016534 002024          BGE      $OVER      ;BR IF MORE ITERATION REQUIRED
3452 016536 012767 000001 162640  1$:      MOV      #1,$ICNT  ;REINITIALIZE THE ITERATION COUNTER
3453 016544 016767 000056 162740  MOV      $MXCNT,$TIMES ;SET NUMBER OF ITERATIONS TO DO
3454 016552 105267 162624          $SVLAD: INCB     $TIMES  ;COUNT TEST NUMBERS
3455 016556 116767 162620 162746  MOVB     $TIM,$STESTN ;SET TEST NUMBER IN APT MAILBOX
3456 016564 011667 162616          MOV      (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
3457 016570 011667 162614          MOV      (SP),$LPERR ;SAVE ERROR LOOP ADDRESS
3458 016574 005067 162714          CLR      $ESCAPE    ;CLEAR THE ESCAPE FROM ERROR ADDRESS
3459 016600 112767 000001 162607  MOVB     #1,$ERMAX   ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3460 016606 016777 162570 162626  $OVER:  MOV      $STNM,$DISPLAY ;DISPLAY TEST NUMBER
3461 016614 016716 162566          MOV      $LPADR,(SP) ;FUDGE RETURN ADDRESS
3462 016620 000002          4$:      RTI
3463 016622 001407          BRW:     1407
3464 016624 000432          BRX:     432
3465 016626 000005          $MXCNT:  5          ;:MAX. NUMBER OF ITERATIONS
3466          .SBTTL TRAP DECODER
3467
3468          ;*****
3469          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION

```

3470
3471
3472
3473
3474 016630 010046
3475 016632 016500 000002
3476 016636 005740
3477 016640 111000
3478 016642 006300
3479 016644 016000 016664
3480 016650 000200
3481
3482
3483
3484
3485 016652 011646
3486 016654 016666 000004 000002
3487 016662 000002
3488
3489
3490
3491
3492
3493
3494
3495
3496 016664 016652
3497 016666 013064
3498 016670 014602
3499 016672 014556
3500 016674 014616
3501
3502
3503 016676 013042
3504 016700 013346
3505 016702 013454
3506 016704 013464
3507 016706 013664
3508 016710 013724
3509 016712 013756
3510 016714 014134
3511
3512
3513
3514
3515
3516 016716 006367 000044
3517 016722 006367 000040
3518 016726 006367 000034
3519 016732 006367 000030
3520 016736 006367 000024
3521 016742 016767 000020 000020
3522 016750 162767 000001 000012
3523 016756 042767 000037 000004
3524 016764 000207
3525 016766 006240

;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;*GO TO THAT ROUTINE.

\$TRAP: MOV RO, -(SP) ;; SAVE RO
MOV 2(SP), RO ;; GET TRAP ADDRESS
TST -(RO) ;; BACKUP BY 2
MOVB (RO), RO ;; GET RIGHT BYTE OF TRAP
ASL RO ;; POSITION FOR INDEXING
MOV \$TRPAD(RO), RO ;; INDEX TO TABLE
RTS RO ;; GO TO ROUTINE

:: 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
CLPRT: PRS

E06

D7DUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 71
 DZDUVA.M11 13-OCT-76 09:26 TRAP TABLE

```

3526 016770 000200 LESS1: PR4 ;LEVEL TO ALLOW INTERRUPTS
3527
3528 ;NEW DU ADDRESSES
3529 016772 016767 000126 162712 DUADDR: MOV DUBASE,RXCSR ;XXX0
3530 017000 005267 000120 INC DUBASE
3531 017004 016767 000114 162702 MOV DUBASE,HRXCSR ;XXX1
3532 017012 005267 000106 INC DUBASE
3533 017016 016767 000102 162672 MOV DUBASE,RXDBUF ;XXX2
3534 017024 016767 000074 162670 MOV DUBASE,PARCSR ;XXX2
3535 017032 005267 000066 INC DUBASE
3536 017036 016767 000062 162654 MOV DUBASE,HRXDBUF ;XXX3
3537 017044 016767 000054 162652 MOV DUBASE,HPARCSR ;XXX3
3538 017052 005267 000046 INC DUBASE
3539 017056 016767 000042 162642 MOV DUBASE,TXCSR ;XXX4
3540 017064 005267 000034 INC DUBASE
3541 017070 016767 000030 162632 MOV DUBASE,HTXCSR ;XXX5
3542 017076 005267 000022 INC DUBASE
3543 017102 016767 000016 162622 MOV DUBASE,TXDBUF ;XXX6
3544 017110 005267 000010 INC DUBASE
3545 017114 016767 000004 162612 MOV DUBASE,HTXDBUF ;XXX7
3546 017122 000207 RTS PC
3547 017124 000000 DUBASE: 0
3548
3549 ;THIS UTILITY POKES THE MAINT DATA BASED UPON THE
3550 ;INFORMATION CONTAINED IN STMP1 AND IT IS
3551 ;SHIFTED IN BY THE CONTENTS OF SHIFT
3552 017126 042777 040000 162572 RPOKE. BIC #MTDATA,@TXCSR
3553 017134 005067 162342 CLR STMP2
3554 017140 006067 162334 ROR STMP1 ;FORCE CARRY
3555 017144 006067 162332 ROR STMP2 ;PICK UP CARRY IN BIT 15
3556 017150 006267 162326 ASR STMP2 ;SHIFT INTO BIT 14
3557 017154 042767 100000 162320 BIC #BIT15,STMP2 ;CLR BIT 15
3558 017162 056777 162314 162536 BIS STMP2,@TXCSR ;POKE MAINT DATA
3559 017170 042777 020000 162530 BIC #CLK,@TXCSR ;POKE CLK
3560 017176 052777 020000 162522 BIS #CLK,@TXCSR ;
3561 017204 005367 161712 DEC SHIFT
3562 017210 001346 BNE RPOKE
3563 017212 000207 RTS PC
3564
3565 ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
3566 017214 016767 162260 162260 OOD8: MOV STMP1,STMP2 ;SAVE TEMP1
3567 017222 005067 162256 CLR STMP3
3568 017226 012727 000010 MOV #8,(PC)+
3569 017232 000000 4$: 0
3570 017234 006067 162242 1$: ROR STMP2
3571 017240 005567 162240 ADC STMP3
3572 017244 005367 177762 DEC 4$
3573 017250 001371 BNE 1$
3574 017252 006067 162226 ROR STMP3
3575 017256 103404 BCS 2$
3576 017260 052767 000400 162212 BIS #BIT8,STMP1 ;SET ODD PARITY
3577 017266 000403 BR 3$
3578 017270 042767 000400 162202 2$: BIC #BIT8,STMP1 ;CLR EVEN PARITY
3579 ;STMP1 NOW HAS ODD PARITY CHARACTER
3580 017276 000207 3$: RTS PC
3581

```

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 72
 DZDUVA.M11 13-OCT-76 08:26 TRAP TABLE

```

3582                                     ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3583 017300 016767 162174 162174 EVEN8: MOV STMP1,STMP2 ;SAVE TEMP1
3584 017306 005067 162172          CLR STMP3
3585 017312 012727 000010          MOV #8,(PC)+
3586 017316 000000          4$: 0
3587 017320 006067 162156          1$: ROR STMP2
3588 017324 005567 162154          ADC STMP3
3589 017330 005367 177762          DEC 4$
3590 017334 001371          BNE 1$
3591 017336 006067 162142          ROR STMP3
3592 017342 103004          BCC 2$
3593 017344 052767 000400 162126  BIS #BIT8,STMP1 ;SET EVEN PARITY
3594 017352 000403          BR 3$
3595 017354 042767 000400 162116 2$: BIC #BIT8,STMP1 ;CLR ODD PARITY
3596                                     ;STMP1 NOW HAS EVEN PARITY CHARACTER
3597 017362 000207          3$: RTS PC
3598 017364 062716 000002  TRPREG: ADD #2,(SP) ;ALLOW IT TO "CRUNCH" INTO HLT BACK
3599                                     ;IN MAIN PART OF THE PROGRAM
3600 017370 000002          RTI
3601 000001          .END

```

AAA	003200	1291*							
ABASE =	000000	874	915						
ACDW1 =	000000	874	917						
ACDW2 =	000000	874	918						
ACPUOP =	000000	874	889						
ACTREG	001166	133#	1247*	1261*	1262*	1269*	2715	2718	2728
ADDW0 =	000000	874	919						
ADDW1 =	000000	874	920						
ADDW10 =	000000	874	929						
ADDW11 =	000000	874	930						
ADDW12 =	000000	874	931						
ADDW13 =	000000	874	932						
ADDW14 =	000000	874	933						
ADDW15 =	000000	874	934						
ADDW2 =	000000	874	921						
ADDW3 =	000000	874	922						
ADDW4 =	000000	874	923						
ADDW5 =	000000	874	924						
ADDW6 =	000000	874	925						
ADDW7 =	000000	874	926						
ADDW8 =	000000	874	927						
ADDW9 =	000000	874	928						
ADEVCT =	000000	874	880						
ADEVN =	000000	874	916						
ADRCNT =	013663	2930*	2969*	2976*					
AENV =	000000	874	885						
AENVN =	000000	874	886						
AFATAL =	000000	874	877						
AMADR1 =	000000	874	902						
AMADR2 =	000000	874	906						
AMADR3 =	000000	874	909						
AMADR4 =	000000	874	912						
AMAMS1 =	000000	874	896						
AMAMS2 =	000000	874	904						
AMAMS3 =	000000	874	907						
AMAMS4 =	000000	874	910						
AMSGAO =	000000	874	882						
AMSQLG =	000000	874	883						
AMSGTY =	000000	874	876						
AMTYP1 =	000000	874	897						
AMTYP2 =	000000	874	905						
AMTYP3 =	000000	874	908						
AMTYP4 =	000000	874	911						
APASS =	000000	874	879						
APRIOR =	000000	874							
APTCSU =	000040	534#	2945						
APTEMV =	000001	534#	2838	3092					
APTSIZ =	000200	534#	1166						
APTSPO =	000100	534#	2840						
ASWREG =	000000	874	887						
ATESTN =	000000	874	878						
AUNIT =	000000	874	881						
AUSWR =	000000	874	888						
AVECT1 =	000000	874	913						
AVECT2 =	000000	874	914						
BASEAD	001154	728#	1229*	1266*	1267	1273*	1275*	2722*	2734* 2738

H06

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 75
 CZDUVA.M11 13-OCT-76 08:26 CROSS REFERENCE TABLE -- USER SYMBOLS

BASEIV	001162	731#	1239#	2723*	2735*	2740	2741*	2742	2743*	2744	2745*	2746	2747*
EBB	003026	1246	1250#										
BINWRD	014132	3011*	3013	3043#									
BITW =	002000	800#	993#										
BIT0 =	000001	663#	776	807	969	1000							
BIT00 =	000001	653#	663										
BIT01 =	000002	652#	662										
BIT02 =	000004	651#	661										
BIT03 =	000010	650#	660										
BIT04 =	000020	649#	659										
BIT05 =	000040	648#	658										
BIT06 =	000100	647#	657										
BIT07 =	000200	646#	656										
BIT08 =	000400	645#	655	3430									
BIT09 =	001000	644#	654	3102	3438								
BIT1 =	000002	662#	775	968									
BIT10 =	002000	643#	766	800	959	993	3080						
BIT11 =	004000	642#	765	958	3445								
BIT12 =	010000	641#	764	781	957	974							
BIT13 =	020000	640#	763	780	799	956	973	992	3087				
BIT14 =	040000	639#	762	779	798	955	972	991	3416				
BIT15 =	100000	638#	761	778	797	954	971	990	3557				
BIT2 =	000004	661#	774	967									
BIT3 =	000010	660#	773	806	966	999							
BIT4 =	000020	659#	772	805	965	998							
BIT5 =	000040	658#	771	804	964	997							
BIT6 =	000100	657#	770	803	963	996							
BIT7 =	000200	656#	769	802	962	995							
BIT8 =	000400	655#	768	784	801	961	977	994	3576	3578	3593	3595	
BIT9 =	001000	654#	767	783	960	976							
BPTVEC =	000014	670#											
BREAK =	000001	807#	1000#	1354	1427	2550	2605	2660					
BRW	016622	3463#											
BRX	016624	3464#											
CARDET =	010000	764#	957#										
CCC	012614	2714	2737	2754#									
CHRCNT	014130	3009#	3014	3027*	3041#	3042							
CKSWR	012714	2708	2770	2778#	2810	3407							
CLK =	020000	799#	992#	1354	1360	1361	1389	1390	1427	1433	1434	1482	1483
		1525	1528	1529	1537	1706	1707	1710	1711	1735	1989	1990	1993
		2008	2009	2035	2072	2073	2076	2077	2091	2092	2118	2155	2156
		2160	2174	2175	2201	2238	2239	2242	2243	2257	2258	2284	2321
		2326	2327	2369	2370	2374	2375	2419	2425	2434	2440	2490	2496
		2511	2550	2556	2557	2559	2560	2605	2611	2612	2614	2615	2660
		2667	2669	2670	3559	3560							2666
CNTLJ	012770	1196	2789#										
CONVRT =	104413	2711	2791	3273	3509#								
COUNT	001124	708#	1873*	1877*	1911	1915*	1956	2001*	2012*	2084*	2095*	2167*	2178*
		2261*											2250*
CR =	000015	578#	2884	2894									
CRLF =	000200	579#	2855	2894									
CTS =	020000	763#	956#										
DDISP =	177570	585#	841	1154									
DEVADR	013660	2928#	2966	2974#									
DEVICE	015323	2712	3303#										
DF1	002132	1025	1029	1033	1037	1083#							

M06

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 80
 DZDUVA.M11 13-OCT-76 08:26 CROSS REFERENCE TABLE -- USER SYMBOLS

SW15 =	100000	610#												
SW2 =	000004	633#												
SW3 =	000010	632#												
SW4 =	000020	631#												
SW5 =	000040	630#												
SW6 =	000100	629#												
SW7 =	000200	628#												
SW8 =	000400	627#												
SW9 =	001000	626#												
SYNCR0	001146	719#	1296*	1300*	1875	2015	2098	2181	2264					
SYNEXT=	020000	787#	980#	1350	1357	1423	1430	1512	1520	1694	1702	1780	1782	2307
		2315	2355	2363	2405	2413	2476	2484						
SYNINT=	030000	786#	979#	1844	1852	1983	2066	2149	2232					
SYNSCH=	000020	772#	965#	1358	1431	1793	1853	1986	2041	2069	2124	2152	2207	2235
		2290	2319	2367	2417	2488	2554	2609	2664					
SYSTST=	014000	812#	1005#	1791	1849									
TBITVE=	000014	668#												
TEMP	016206	3015	3270*	3271*	3384*									
TKVEC =	000060	675#												
TPVEC =	000064	676#												
TRAPVE=	000034	674#	1140*	1141*										
TRPREG	017364	3598#												
TRTVEC=	000014	669#												
TST1	003362	1331	1337	1347*	2771	2772	3409							
TST10	005610	1775#												
TST11	006100	1842#												
TST12	006714	1979#												
TST13	007220	2062#												
TST14	007524	2145#												
TST15	010030	2228#												
TST16	010334	2305#												
TST17	010532	2353#												
TST2	003706	1418#												
TST20	010730	2401#												
TST21	011174	2472#												
TST22	011440	2544#												
TST23	011666	2599#												
TST24	012114	2654#												
TST3	004324	1509#												
TST4	004576	1566#												
TST5	004740	1604#												
TST6	005076	1647#												
TST7	005246	1691#												
TTST	016330	3412#												
TXCSR	001726	1046#	1349*	1351*	1354*	1360*	1361*	1384*	1388*	1389*	1390*	1422*	1424*	1427*
		1433#	1434*	1477*	1481*	1482*	1483*	1511*	1513*	1517*	1524*	1525*	1528*	1529*
		1535#	1537*	1545	1548*	1557*	1568*	1572*	1576*	1582*	1583	1605*	1609*	1613*
		1620#	1648*	1652*	1669	1681*	1693*	1695*	1699*	1706*	1707*	1710*	1711*	1717*
		1726	1729	1735*	1742*	1751	1754*	1761*	1779*	1781*	1791*	1843*	1845*	1849*
		1879*	1891*	1910*	1924*	1980*	1984*	1989*	1990*	1993*	1994*	2002	2008*	2009*
		2035*	2036	2040*	2063*	2067*	2072*	2073*	2076*	2077*	2085	2091*	2092*	2118*
		2119	2123*	2146*	2150*	2155*	2156*	2159*	2160*	2168	2174*	2175*	2201*	2202
		2206*	2229*	2233*	2238*	2239*	2242*	2243*	2251	2257*	2258*	2284*	2285	2289*
		2306*	2308*	2312*	2321*	2322*	2326*	2327*	2354*	2356*	2360*	2369*	2370*	2374*
		2375*	2404*	2406*	2410*	2419*	2425*	2434*	2440*	2475*	2477*	2481*	2490*	2496*
		2505*	2511*	2545*	2547*	2550*	2556*	2557*	2559*	2560*	2600*	2602*	2605*	2611*

G07

DZDUV-A MACY11 27(1006) 03-FEB-77 08:30 PAGE 88
DZDUVA.M11 13-OCT-76 09:26 CROSS REFERENCE TABLE -- MACRO NAMES

.SERRT	523#	3118
.SPOWE	523#	
.SSCOP	523#	3388
.STRAP	523#	3466
.STYPE	523#	2815
.STYPO	523#	3165

. ABS. 017372 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDUVA, DZDUVA.SEG/SOL/CRF+DZDUV1/EG:RUNF, DZDUV2, DZDUVA
RUN-TIME: 18 12 1 SECONDS
RUN-TIME RATIO: 150/31=4.7
CORE USED: 31K (62 PAGES)