

DUV/LSI-11

DUV11 OFF-LINE RCVR TIME
MD-11-DZDUS-A

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

APR 197777
digital
MADE IN USA

The microfiche card displays a grid of 48 frames of data, organized into 8 rows and 6 columns. Each frame contains a small portion of a larger document, likely a program listing or data dump. The text is dense and appears to be a mix of alphanumeric characters, possibly representing code or data points. The frames are arranged in a regular grid pattern, with some frames appearing to be blank or containing very faint text.

11

.REM *

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

PRODUCT CODE: MAINDEC-11-DZDUS-A-D
PRODUCT NAME: DUV11 OFFLINE RECEIVER TIMING 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 RECEIVER TIMING TESTS VERIFY THAT THE RECEIVER LOGIC AND ASSOCIATED ERROR FLAGS ASSERT AT THE PROPPER TIME

* .REM *

2. REQUIREMENTS

* .REM *

PDP-11/03 COMPUTER (LSI)

DUV11 SYNCHRONOUS/ISOCRONOUS OPTION

ONE CONSOLE TELETYPE OR EQUIVALENT

2.2 STORAGE

THE PROGRAM LOADS INTO 4K OF MEMORY WITH BOOTSTRAP

3. LOADING PROCEDURE

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

	STARTING ADDRESS FOR ABSOLUTE LOADER
4K	017500
8K	037500
12K	057500
16K	077500
20K	117500
24K	137500
28K	157500

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

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

NOTE: RUNNING UNDER APT-11, THERE IS A USER SWITCH REGISTER CALLED "SUSWR". IN ORDER TO BE FLEXIBLE ON THE AVAILABILITY OF THE H315 CONNECTOR, ONE BIT PASSES STATUS TO APT-11. BIT 0 IN SUSWR REFLECTS THIS STATUS, A 0 = CONNECTOR PRESENT, A 1 = CONNECTOR NOT AVAILBLE.

- 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 DZDUS-A TAPE C" (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.

E01

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 4
DZDUS1.M11 02-FEB-77 08:17

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

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 5
 DZDUS1.M11 02-FEB-77 08:17

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)

GO1

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 6
DZDUS1.M11 02-FEB-77 08:17

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

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 10
DZDUS1.M11 02-FEB-77 08:17

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

9. PROGRAM DESCRIPTION

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

*
 .REM *
 *
 .REM *

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

11. LISTINGS

*

L01

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 12
DZDUS2.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

000001

STN=1

MO1

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 14
 DZDUSA.M11 13-OCT-76 08:39 APT COMMUNICATIONS ROUTINE

```

556 .ENABLE ABS
557
558 ;DUV11 DZDUS-A TAPE C
559 ;COPYRIGHT 1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
560
561 ;STARTING PROCEDURE
562 ;TYPE 200G
563 ;PROGRAM WILL TYPE "DUV11 DZDUS-A TAPE C"
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 C"
566 ;AND THEN RESUME TESTING
567
568 .SBTTL BASIC DEFINITIONS
569
570 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
571      001100  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  STKLM= 177774 ;;STACK LIMIT REGISTER
583      177772  PIRQ= 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
 613 010000
 614 004000
 615 002000
 616 001000
 617 000400
 618 000200
 619 000100
 620 000040
 621 000020
 622 000010
 623 000004
 624 000002
 625 000001
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638 100000
 639 040000
 640 020000
 641 010000
 642 004000
 643 002000
 644 001000
 645 000400
 646 000200
 647 000100
 648 000040
 649 000020
 650 000010
 651 000004
 652 000002
 653 000001
 654
 655
 656
 657
 658
 659
 660
 661
 662
 663
 664
 665
 666 000004
 667 000010

SW13= 20000
 SW12= 10000
 SW11= 4000
 SW10= 2000
 SW09= 1000
 SW08= 400
 SW07= 200
 SW06= 100
 SW05= 40
 SW04= 20
 SW03= 10
 SW02= 4
 SW01= 2
 SW00= 1
 .EQUIV SW09,SW9
 .EQUIV SW08,SW8
 .EQUIV SW07,SW7
 .EQUIV SW06,SW6
 .EQUIV SW05,SW5
 .EQUIV SW04,SW4
 .EQUIV SW03,SW3
 .EQUIV SW02,SW2
 .EQUIV SW01,SW1
 .EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
 BIT14= 40000
 BIT13= 20000
 BIT12= 10000
 BIT11= 4000
 BIT10= 2000
 BIT09= 1000
 BIT08= 400
 BIT07= 200
 BIT06= 100
 BIT05= 40
 BIT04= 20
 BIT03= 10
 BIT02= 4
 BIT01= 2
 BIT00= 1
 .EQUIV BIT09,BIT9
 .EQUIV BIT08,BIT8
 .EQUIV BIT07,BIT7
 .EQUIV BIT06,BIT6
 .EQUIV BIT05,BIT5
 .EQUIV BIT04,BIT4
 .EQUIV BIT03,BIT3
 .EQUIV BIT02,BIT2
 .EQUIV BIT01,BIT1
 .EQUIV BIT00,BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

ERRVEC= 4 ;;TIME OUT AND OTHER ERRORS
 RESVEC= 10 ;;RESERVED AND ILLEGAL INSTRUCTIONS

568	000014	TBITVEC=14	;; "T" BIT
569	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

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 17
 DZDUSA.M11 13-OCT-76 08:39 BASIC DEFINITIONS

; STANDARD INTERRUPT VECTORS

```

678
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
690      001100 000000      .=1100
691      001102 177570      .WORD 0
692
693
694
695
696
697
698
699
700
701
702
703
704
705

```

; PROGRAM CONTROL PARAMETERS

```

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
705

```

; PROGRAM VARIABLES

```

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

```


DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 18
 DZDUSA.M11 13-OCT-76 08:39 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 001400 .=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 REACT=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 SYNSCH=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 ; RTC ERROR
779 040000 OVERRUN=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

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 20
DZDUSA.M11 13-OCT-76 09:39 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
          .WORD 0
$STNM:   .BYTE 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
$GDADR:  .WORD 0 ; CONTAINS 'GOOD' DATA
$BDADR:  .WORD 0 ; CONTAINS 'BAD' DATA
          .WORD 0 ; RESERVED--NOT TO BE USED
          .WORD 0
$AUTOB:  .BYTE 0 ; AUTOMATIC MODE INDICATOR
$INTAG:  .BYTE 0 ; INTERRUPT MODE INDICATOR
          .WORD 0
SWR:     .WORD DSWR ; ADDRESS OF SWITCH REGISTER
DISPLAY: .WORD DDISP ; 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"
$TPFLG:  .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
$CRFLF:  .ASCII <15> ; CARRIAGE RETURN
    
```

869	001524	000012	SLF:	.ASCIZ	<12>	::LINE FEED	
870			:	*****			
871			:	SBTTL APT MAILBOX-ETABLE			
872			:	*****			
873			:	*****			
874			:	EVEN			
875	001526		\$MAIL:			::APT MAILBOX	
876	001526	000000	\$MSGTY:	.WORD	AMSGTY	::MESSAGE TYPE CODE	
877	001530	000000	\$FATAL:	.WORD	AFATAL	::FATAL ERROR NUMBER	
878	001532	000000	\$TESTN:	.WORD	ATESTN	::TEST NUMBER	
879	001534	000000	\$PASS:	.WORD	APASS	::PASS COUNT	
880	001536	000000	\$DEVCT:	.WORD	ADEVCT	::DEVICE COUNT	
881	001540	000000	\$UNIT:	.WORD	AUNIT	::I/O UNIT NUMBER	
882	001542	000000	\$MSGAD:	.WORD	AMSGAD	::MESSAGE ADDRESS	
883	001544	000000	\$MSGLG:	.WORD	AMSGLG	::MESSAGE LENGTH	
884	001546		\$ETABLE:			::APT ENVIRONMENT TABLE	
885	001546	000	\$ENV:	.BYTE	AENV	::ENVIRONMENT BYTE	
886	001547	000	\$ENVM:	.BYTE	AENVM	::ENVIRONMENT MODE BITS	
887	001550	000000	\$SWREG:	.WORD	ASWREG	::APT SWITCH REGISTER	
888	001552	000000	\$USWR:	.WORD	AUSWR	::USER SWITCHES	
889	001554	000000	\$CPUOP:	.WORD	ACPUOP	::CPU TYPE, OPTIONS	
890			:	BITS 15-11=CPU TYPE			
891			:	11/04=01,11/05=02,11/20=03,11/40=04,11/45=05			
892			:	11/70=06,POQ=07,Q=10			
893			:	BIT 10=REAL TIME CLOCK			
894			:	BIT 9=FLOATING POINT PROCESSOR			
895			:	BIT 8=MEMORY MANAGEMENT			
896	001556	000	\$MAMS1:	.BYTE	AMAMS1	::HIGH ADDRESS, M.S. BYTE	
897	001557	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE BLK#1	
898			:	MEM. TYPE BYTE -- (HIGH BYTE)			
899			:	900 NSEC CORE=001			
900			:	300 NSEC BIPOLAR=002			
901			:	500 NSEC MOS=003			
902	001560	000000	\$MADR1:	.WORD	AMADR1	::HIGH ADDRESS, BLK#1	
903			:	MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE			
904	001562	000	\$MAMS2:	.BYTE	AMAMS2	::HIGH ADDRESS, M.S. BYTE	
905	001563	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE, BLK#2	
906	001564	000000	\$MADR2:	.WORD	AMADR2	::MEM. LAST ADDRESS, BLK#2	
907	001566	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS, M.S. BYTE	
908	001567	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE, BLK#3	
909	001570	000000	\$MADR3:	.WORD	AMADR3	::MEM. LAST ADDRESS, BLK#3	
910	001572	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS, M.S. BYTE	
911	001573	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE, BLK#4	
912	001574	000000	\$MADR4:	.WORD	AMADR4	::MEM. LAST ADDRESS, BLK#4	
913	001576	000000	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1, BUS PRIORITY#1	
914	001600	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2, BUS PRIORITY#2	
915	001602	000000	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST	
916	001604	000000	\$DEVN:	.WORD	ADEVN	::DEVICE MAP	
917	001606	000000	\$CDW1:	.WORD	ACDW1	::CONTROLLER DESCRIPTION WORD#1	
918	001610	000000	\$CDW2:	.WORD	ACDW2	::CONTROLLER DESCRIPTION WORD#2	
919	001612	000000	\$DDW0:	.WORD	ADDW0	::DEVICE DESCRIPTOR WORD#0	
920	001614	000000	\$DDW1:	.WORD	ADDW1	::DEVICE DESCRIPTOR WORD#1	
921	001616	000000	\$DDW2:	.WORD	ADDW2	::DEVICE DESCRIPTOR WORD#2	
922	001620	000000	\$DDW3:	.WORD	ADDW3	::DEVICE DESCRIPTOR WORD#3	
923	001622	000000	\$DDW4:	.WORD	ADDW4	::DEVICE DESCRIPTOR WORD#4	
924	001624	000000	\$DDW5:	.WORD	ADDW5	::DEVICE DESCRIPTOR WORD#5	

941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996

005746
005726
010046
012600
024646
022626

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

100000
040000
020000
010000

001000
000400

030000
020000
000000
000000
002000
004000
006000
000000
001000
001400

100000
040000
020000
002000
000400
000200
000100

; INSTRUCTION DEFINITIONS

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

; REGISTER DEFINITIONS

; RXCSR BIT DEFINITIONS

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

; RXDBUF BIT DEFINITIONS

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

; PARCSR BIT DEFINITIONS

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

; PARCSR WRD DEFINITIONS

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

; TXCSR BIT DEFINITIONS

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

K02

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 25
DZDUSA.M11 13-OCT-76 08:39 APT MAILBOX-ETABLE

997	000040	DNAINTE=BITS	;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

.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/

1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020 001652
1021
1022 001652 001762
1023 001654 002067
1024 001656 002116
1025 001660 002132
1026 001662 002022
1027 001664 002067
1028 001666 002116
1029 001670 002132
1030 001672 002043
1031 001674 002067
1032 001676 002116
1033 001700 002132
1034 001702 001746
1035 001704 000000
1036 001706 002126
1037 001710 002132
1038
1039
1040 001712 160010
1041 001714 160011
1042 001716 160012
1043 001720 160013
1044 001722 160012
1045 001724 160013
1046 001726 160014
1047 001730 160015
1048 001732 160016
1049 001734 160017
1050
1051 001736 000770
1052 001740 000772
1053 001742 000774
1054 001744 000776
1055
1056 001746 020040 051105 047522
1057 001754 020122 041520 000040
1058 001762 020040 047503 050115
1059 001770 051101 051511 047117
1060 001776 042440 051122 051117
1061 002004 047440 020116 042522

M02

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 27
 DZDUSA.M11 13-OCT-76 08:39 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,HLDD,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=.	;SAVE PC
1091		000046				.=46	
1092	000046	012646				\$ENDAD	;;1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .SEOP
1093		000052				.=52	
1094	000052	000000				.WORD 0	;;2)SET LOC.52 TO ZERO
1095		002136				.=\$SVPC	;; RESTORE PC
1096						.SBTTL APT PARAMETER BLOCK	
1097							
1098						;;*****	
1099						;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT	
1100						;;*****	
1101		002136				.SX=.	;SAVE CURRENT LOCATION
1102		000024				.=24	;SET POWER FAIL TO POINT TO START OF PROGRAM
1103	000024	000200				200	;FOR APT START UP
1104		000044				.=44	;POINT TO APT INDIRECT ADDRESS PNTR.
1105	000044	002136				\$APTHDR	;POINT TO APT HEADER BLOCK
1106		002136				.=\$SX	;RESET LOCATION COUNTER
1107						;;*****	
1108						;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC	
1109						;INTERFACE SPEC.	
1110							
1111	002136					\$APTHD:	
1112	002136	000000				\$HIBTS: .WORD 0	;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1113	002140	001526				\$MBADR: .WORD \$MAIL	;; ADDRESS OF APT MAILBOX (BITS 0-15)
1114	002142	000010				\$STMT: .WORD 10	;; RUN TIM OF LONGEST TEST
1115	002144	000010				\$PASTM: .WORD 10	;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1116	002146	000000				\$UNITM: .WORD	;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1117	002150	000052				.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    #SCMTAG,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 016276 000020 MOV    #SCOPE,#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1137 002200    012737 000340 000022 MOV    #340,#IOTVEC+2 ;;LEVEL 7
1138 002206    012737 014166 000030 MOV    #ERROR,#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1139 002214    012737 000340 000032 MOV    #340,#EMTVEC+2 ;;LEVEL 7
1140 002222    012737 016614 000034 MOV    #TRAP,#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1141 002230    012737 000340 000036 MOV    #340,#TRAPVEC+2;LEVEL 7
1142 002236    012737 014770 000024 MOV    #SPWRON,#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    $ESCAPE     ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1146 002262    112767 000001 177125 MOV    #1,$SERMAX   ;;ALLOW ONE ERROR PER TEST
1147 002270    012767 002270 177110 MOV    #,$SLPADR    ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1148 002276    012767 002276 177104 MOV    #,$SLPERR    ;;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$,#ERRVEC ;;SET UP ERROR VECTOR
1153 002316    012767 177570 177114 MOV    #USWR,SWR    ;;SETUP FOR A HARDWARE SWICH REGISTER
1154 002324    012767 177570 177110 MOV    #DISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1155 002332    022777 177777 177100 CMP    #-1,$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    $PASS       ;;CLEAR PASS COUNT
1166 002376    132767 000200 177143 BITB   #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
1167 002404    001403          BEQ   67$         ;;YES,USE NON-APT SWITCH
1168 002406    012767 001550 177024 MOV    #SSWREG,SWR  ;;NO,USE APT SWITCH REGISTER
1169 002414          67$:
1170 002414    012706 001100          MOV    #STACK,SP   ;;SET STACK
1171 002420    106427 000340          MTPS  #340         ;;LOCK INTERRUPTS
1172 002424    012737 014770 000024 MOV    #.PFAIL,#24  ;;SET UP POWER FAIL VECTOR
1173 002432    105067 176535          CLR   $STFLG      ;;CLEAR START FLAG
  
```

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	SERRPC		;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	25		
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	35		
1193	002554	005737	000042		TST	#42		;CHECK FOR CHAIN
1194	002560	001402			BEQ	335		
1195	002562	000167	000522		JMP	.BEGIN		
1196	002566	004767	010162		335:	JSR	PC,CNTLU	
1197	002572	105767	176374		35:	TSTB	INIFLG	;HAS INITIALIZATION BEEN PERFORMED
1198	002576	001004			BNE	ONCE		
1199	002600	104401	015130		TYPE	MTITLE		;TYPE TITLE MESSAGE
1200	002604	105167	176362		COMB	INIFLG		;IF NOT SET FLAG AND DO
1201	002610	105767	176732		ONCE:	TSTB	SENV	;APT CONTROL?
1202	002614	001410			BEQ	115		;BR IF NO
1203	002616	032767	000001	176726	BIT	#1,SUSWR		;EXTENAL JUMPER ON?
1204	002624	001002			BNE	125		;NO
1205	002626	105067	176321		CLRB	JMRBY		;CLEAR FLAG
1206	002632	000167	000452		125:	JMP	.BEGIN	;GO DO IT
1207	002636	032777	000001	176574	115:	BIT	SW00,SWR	;RESELECT VECTOR & CONTROL REG?
1208	002644	001002			BNE	15		
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	25		
1219	002704	104406			INSTR			;OUTPUT MESSAGE & GET INPUT STRING
1220	002706	015176			MREGAD			;MESSAGE
1221	002710	104410			PARAM			;CONVERT STRING
1222	002712	160000			160000			;LOW LIMIT
1223	002714	167776			167776			;HIGH LIMIT
1224	002716	017110			DUBASE			;STORE AT THIS LOCATION
1225	002720	001			.BYTE	1		;MASK
1226	002721	001			.BYTE	1		;HOW MANY TIMES + 2
1227	002722	016767	014162	176226	MOV	DUBASE,KEEPADD		;SAVE
1228	002730	004767	014022		JSR	PC,DJADDR		
1229	002734	016767	176216	176212	MOV	KEEPADD,BASEADD		;RESTORE FOR ROTATION

1230	002742	104406				INSTR	: OUTPUT MESSAGE & GET INPUT STRING
1231	002744	015163				MVECTOR	: 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	015226				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	015255				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,BASEADD :SET UP BASE ADDRESS
1267	003102	026767	176052	176044		CMP	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		MOV	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	015351				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	013556	OUTMUL:	MOV	#340,DUPRT

```

1286 003174 004767 013502 JSR PC,DLEV
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 AAA:
1292 003200 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1293 003202 015567 MSYNC ;MESSAGE
1294 003204 122767 000061 012716 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 012700 1$: CMPB #'2,INBUF ;IS IT "2" ?
1299 003230 001004 BNE 2$
1300 003232 112767 :77777 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$: NOP
1305 003250 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1306 003252 015635 MWIRE6 ;MESSAGE
1307 003254 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1308 003256 001147 SEXMIT ;THIS FLAG
1309 003260 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1310 003262 015706 MWIRE5 ;MESSAGE
1311 003264 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1312 003266 001150 SEREC ;THIS FLAG
1313 003270 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1314 003272 015756 MWIRE4 ;MESSAGE
1315 003274 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1316 003276 001151 OPTCLR ;THIS FLAG
1317 003300 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1318 003302 016035 NEXTJ ;MESSAGE
1319 003304 104414 SETFLG ;SET FLAG BASED UPON INPUT STRING
1320 003306 001153 JMRBY ;THIS FLAG
1321
1322 ;TEST START AND RESTART
1323
1324 003310 012706 001100 .BEGIN: MOV #'STACK,SP ;SET UP STACK
1325 003314 106427 000340 MTPS #'340 ;LOCK OUT INTERRUPTS
1326 003320 032777 000002 176112 BIT #'SW01,JSWR ;IF SW01=1, GET STARTING PC
1327 003326 001406 BEQ 3$
1328 003330 104406 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1329 003332 015521 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$: MOV #TST1,RETURN ;START AT TEST 1
1338 003352 104401 015515 4$: TYPE MR ;TYPE R
1339 003356 000177 175522 JMP #RETURN ;START TESTING
1340
1341 ;;THIS TEST VERIFYS WORD LENGTH SELECT OF THE
    
```

```

1342                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
1343                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1344                                     ;; (OVRUN, RXERR)
1345                                     ;; MODE: SYNEXT
1346                                     ;; LENGTH: SEVEN
1347                                     ;; CHAR: 177
1348
1349                                     ..*****
1350 003362 000004          †ST1: SCOPE
1351 003364 052777 000400 176334      BIS      #MRESET, @TXCSR ; MASTER RESET
1352 003372 012777 020000 176322      MOV      #SYNEXT, @PARCSR ; SET THE MODE
1353 003400 052777 000400 176320      BIS      #MRESET, @TXCSR ; MASTER RESET
1354
1355                                     ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
1356 003406 012777 064001 176312      MOV      #MTDATA!CLK!MINT!BREAK, @TXCSR
1357
1358                                     ; SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
1359 003414 012777 024000 176300      MOV      #SYNEXT!SEVEN!NOPAR!0, @PARCSR
1360 003422 052777 000020 176262      BIS      #SYNSCH, @RXCSR ; SET SEARCH SYNC
1361
1362                                     ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1363 003430 042777 020000 176270      BIC      #CLK, @TXCSR ; POKE CLK DOWN
1364 003436 052777 020000 176262      BIS      #CLK, @TXCSR ; POKE CLK UP
1365 003444 016703 176246      MOV      RXDBUF, R3 ; SET UP FOR ERROR MESSAGE
1366 003450 012700 000177          MOV      #177, R0 ; EXPECTED
1367 003454 012767 000007 175440      MOV      #7, SHIFT ; # OF SHIFTS
1368 003462 012767 000177 176010      MOV      #177, $TMP1 ; DATA CHAR
1369 003470 004767 013416      JSR      PC, RPOKE ; SHIFT IN THIS CHAR
1370 003474 105777 176212      TSTB    @RXCSR ; RXDONE
1371 003500 100401          BMI      +4
1372 003502 104004          ERROR 4 ; RXDONE SHOULD BE SET
1373 003504 017701 176206      MOV      @RXDBUF, R1 ; ACTUAL
1374 003510 020001          CMP      R0, R1 ; COMPARE EXPECTED VS. ACTUAL
1375 003512 001401          BEQ     +4
1376 003514 104002          ERROR 2 ; RECEIVED DATA DID NOT MATCH
1377                                     ; EXPECTED DATA - CHECK MAINT DATA
1378                                     ; OR RECEIVER LOGIC
1379 003516 012767 000007 175376      MOV      #7, SHIFT ; # OF SHIFTS
1380 003524 012767 000177 175746      MOV      #177, $TMP1 ; DATA CHAR
1381 003532 004767 013354      JSR      PC, RPOKE ; SHIFT IN THIS CHAR
1382                                     ; NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1383 003536 012767 000007 175356      MOV      #7, SHIFT ; # OF SHIFTS
1384 003544 012767 000177 175726      MOV      #177, $TMP1 ; DATA CHAR
1385 003552 004767 013334      JSR      PC, RPOKE ; SHIFT IN THIS CHAR
1386 003556 012700 140177      MOV      #140000!177, R0 ; EXPECTED DATA PLUS
1387                                     ; RXERR & OVRUN
1388 003562 017701 176130      MOV      @RXDBUF, R1 ; ACTUAL
1389 003566 020001          CMP      R0, R1 ; COMPARE EXP VS. ACT
1390 003570 001401          BEQ     +4
1391 003572 104002          ERROR 2 ; SPECIFICALLY LOOK AT RXERR &
1392                                     ; OVRUN BITS...THEY BOTH SHOULD BE SET
1393
1394                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1395                                     ;; RECEIVER SECTION, IT USES THE ERROR FLAGS
1396                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1397                                     ;; (OVRUN, RXERR)
1398                                     ;; MODE: SYNEXT

```

F03

```

1398                                     ;;LENGTH:SEVEN
1399                                     ;;CHAR:0
1400                                     ;;
1401                                     ;*****
1402 003574 000004          †ST2: SCOPE
1403 003576 052777 000400 176122      BIS      #MRESET,@TXCSR ;MASTER RESET
1404 003604 012777 020000 176110      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1405 003612 052777 000400 176106      BIS      #MRESET,@TXCSR ;MASTER RESET
1406
1407                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1408 003620 012777 064001 176100      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1409
1410                                     ;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
1411 003626 012777 024000 176066      MOV      #SYNEXT!SEVEN!NOPAR!0,@PARCSR
1412 003634 052777 000020 176050      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
1413                                     ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1414 003642 042777 020000 176056      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1415 003650 052777 020000 176050      BIS      #CLK,@TXCSR ;POKE CLK UP
1416 003656 016703 176034              MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1417 003662 012700 000000              MOV      #0,R0 ;EXPECTED
1418 003666 012767 000007 175226      MOV      #7,SHIFT ;# OF SHIFTS
1419 003674 012767 000000 175576      MOV      #0,STMP1 ;DATA CHAR
1420 003702 004767 013204              JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1421 003706 105777 176000              TSTB    @RXCSR ;RXDONE
1422 003712 100401 .+4
1423 003714 104004 ERROR 4 ;RXDONE SHOULD BE SET
1424 003716 017701 175774              MOV      @RXDBUF,R1 ;ACTUAL
1425 003722 020001 CMP      R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1426 003724 001401 BEQ      .+4
1427 003726 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1428                                     ;EXPECTED DATA - CHECK MAINT DATA
1429                                     ;OR RECEIVER LOGIC
1430 003730 012767 000007 175164      MOV      #7,SHIFT ;# OF SHIFTS
1431 003736 012767 000000 175534      MOV      #0,STMP1 ;DATA CHAR
1432 003744 004767 013142              JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1433                                     ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1434 003750 012767 000007 175144      MOV      #7,SHIFT ;# OF SHIFTS
1435 003756 012767 000000 175514      MOV      #0,STMP1 ;DATA CHAR
1436 003764 004767 013122              JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1437 003770 012700 140000              MOV      #140000!0,R0 ;EXPECTED DATA PLUS
1438                                     ;RXERR & OVRUN
1439 003774 017701 175716              MOV      @RXDBUF,R1 ;ACTUAL
1440 004000 020001 CMP      R0,R1 ;COMPARE EXP VS. ACT
1441 004002 001401 BEQ      .+4
1442 004004 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1443                                     ;OVRUN BITS...THEY BOTH SHOULD BE SET
1444
1445                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1446                                     ;; RECEIVER SECTION,IT USES THE ERROR FLAGS
1447                                     ;; TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1448                                     ;; (OVRUN,RXERR)
1449                                     ;; MODE:SYNEXT
1450                                     ;; LENGTH:EIGHT
1451                                     ;; CHAR:125
1452                                     ;*****
1453

```



```

1454 004006 000004          TST3:  SCOPE
1455 004010 052777 000400 175710      BIS      #MRESET,@TXCSR ;MASTER RESET
1456 004016 012777 020000 175676      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1457 004024 052777 000400 175674      BIS      #MRESET,@TXCSR ;MASTER RESET
1458
1459                                ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1460 004032 012777 064001 175666      MOV      #MNTDATA:CLK!MINT!BREAK,@TXCSR
1461
1462                                ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
1463 004040 012777 026000 175654      MOV      #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1464 004046 052777 000020 175636      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
1465                                ;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
1466 004054 042777 020000 175644      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1467 004062 052777 020000 175636      BIS      #CLK,@TXCSR ;POKE CLK UP
1468 004070 016703 175622          MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1469 004074 012700 000125          MOV      #125,R0 ;EXPECTED
1470 004100 012767 000010 175014      MOV      #8,SHIFT ;# OF SHIFTS
1471 004106 012767 000125 175364      MOV      #125,STMP1 ;DATA CHAR
1472 004114 004767 012772          JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1473 004120 105777 175566          TSTB    @RXCSR ;RXDONE
1474 004124 100401          BMI     .+4
1475 004126 104004          ERROR  4 ;RXDONE SHOULD BE SET
1476 004130 017701 175562          MOV      @RXDBUF,R1 ;ACTUAL
1477 004134 020001          CMP     R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1478 004136 001401          BEQ    .+4
1479 004140 104002          ERROR  2 ;RECEIVED DATA DID NOT MATCH
1480                                ;EXPECTED DATA - CHECK MAINT DATA
1481                                ;OR RECEIVER LOGIC
1482 004142 012767 000010 174752      MOV      #8,SHIFT ;# OF SHIFTS
1483 004150 012767 000125 175322      MOV      #125,STMP1 ;DATA CHAR
1484 004156 004767 012730          JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1485                                ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1486 004162 012767 000010 174732      MOV      #8,SHIFT ;# OF SHIFTS
1487 004170 012767 000125 175302      MOV      #125,STMP1 ;DATA CHAR
1488 004176 004767 012710          JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1489 004202 012700 140125          MOV      #140000!125,R0 ;EXPECTED DATA PLUS
1490                                ;RXERR & OVRUN
1491          MOV      @RXDBUF,R1 ;ACTUAL
1492          CMP     R0,R1 ;COMPARE EXP VS. ACT
1493          BEQ    .+4
1494          ERROR  2 ;SPECIFICALLY LOOK AT RXERR &
1495                                ;OVRUN BITS...THEY BOTH SHOULD BE SET
1496
1497                                ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1498                                ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
1499                                ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1500                                ;: (OVRUN,RXERR)
1501                                ;: MODE:SYNEXT
1502                                ;: LENGTH:EIGHT
1503                                ;: CHAR:252
1504
1505                                ;*****
1506 004220 000004          †ST4:  SCOPE
1507 004222 052777 000400 175476      BIS      #MRESET,@TXCSR ;MASTER RESET
1508 004230 012777 020000 175464      MOV      #SYNEXT,@PARCSR ;SET THE MODE
1509 004236 052777 000400 175462      BIS      #MRESET,@TXCSR ;MASTER RESET

```

H03

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 35
 DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

```

1510
1511 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1512 004244 012777 064001 175454 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1513
1514 ;SET MODE, # OF BITS,PARITY SENSE,&LOAD SYNC REG
1515 004252 012777 026000 175442 MOV #SYNEXT!EIGHT!NOPAR!0,@PARCSR
1516 004260 052777 000020 175424 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1517 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1518 004266 042777 020000 175432 BIC #CLK,@TXCSR ;POKE CLK DOWN
1519 004274 052777 020000 175424 BIS #CLK,@TXCSR ;POKE CLK UP
1520 004302 016703 175410 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1521 004306 012700 000252 MOV #252,R0 ;EXPECTED
1522 004312 012767 000010 174602 MOV #8,SHIFT ;# OF SHIFTS
1523 004320 012767 000252 175152 MOV #252,STMP1 ;DATA CHAR
1524 004326 004767 012560 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1525 004332 105777 175354 TSTB @RXCSR ;RXDONE
1526 004336 100401 BMI .+4
1527 004340 104004 ERROR 4 ;RXDONE SHOULD BE SET
1528 004342 017701 175350 MOV @RXDBUF,R1 ;ACTUAL
1529 004346 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1530 004350 001401 BEQ .+4
1531 004352 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1532 ;EXPECTED DATA - CHECK MAINT DATA
1533 ;OR RECEIVER LOGIC
1534 004354 012767 000010 174540 MOV #8,SHIFT ;# OF SHIFTS
1535 004362 012767 000252 175110 MOV #252,STMP1 ;DATA CHAR
1536 004370 004767 012516 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1537 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1538 004374 012767 000010 174520 MOV #8,SHIFT ;# OF SHIFTS
1539 004402 012767 000252 175070 MOV #252,STMP1 ;DATA CHAR
1540 004410 004767 012476 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1541 004414 012700 140252 MOV #140000!252,R0 ;EXPECTED DATA PLUS
1542 ;RXERR & OVRUN
1543 004420 017701 175272 MOV @RXDBUF,R1 ;ACTUAL
1544 004424 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1545 004426 001401 BEQ .+4
1546 004430 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1547 ;OVRUN BITS...THEY BOTH SHOULD BE SET
1548
1549 ;: THIS TEST VERIFYS WORC LENGTH SELECT OF THE
1550 ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
1551 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1552 ;: (OVRUN,RXERR)
1553 ;: MODE:SYNEXT
1554 ;: LENGTH:EIGHT
1555 ;: CHAR:377
1556
1557 ;:*****
1558 004432 000004 tSTS: SCOPE
1559 004434 052777 000400 175264 BIS #MRESET,@TXCSR ;MASTER RESET
1560 004442 012777 020000 175252 MOV #SYNEXT,@PARCSR ;SET THE MODE
1561 004450 052777 000400 175250 BIS #MRESET,@TXCSR ;MASTER RESET
1562
1563 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1564 004456 012777 064001 175242 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1565

```

```

1566 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
1567 004464 012777 026000 175230 MOV #SYNEXT,EIGHT!NOPAR!0,@PARCSR
1568 004472 052777 000020 175212 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1569 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1570 004500 042777 020000 175220 BIC #CLK,@TXCSR ;POKE CLK DOWN
1571 004506 052777 020000 175212 BIS #CLK,@TXCSR ;POKE CLK UP
1572 004514 016703 175176 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1573 004520 012700 000377 MOV #377,R0 ;EXPECTED
1574 004524 012767 000010 174370 MOV #8,SHIFT ;# OF SHIFTS
1575 004532 012767 000377 174740 MOV #377,$TMP1 ;DATA CHAR
1576 004540 004767 012346 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1577 004544 105777 175142 TSTB @RXCSR ;RXDONE
1578 004550 100401 BMI .+4
1579 004552 104004 ERROR 4 ;RXDONE SHOULD BE SET
1580 004554 017701 175136 MOV @RXDBUF,R1 ;ACTUAL
1581 004560 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1582 004562 001401 BEQ .+4
1583 004564 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1584 ;EXPECTED DATA - CHECK MAINT DATA
1585 ;OR RECEIVER LOGIC
1586 004566 012767 000010 174326 MOV #8,SHIFT ;# OF SHIFTS
1587 004574 012767 000377 174676 MOV #377,$TMP1 ;DATA CHAR
1588 004602 004767 012304 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1589 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1590 004606 012767 000010 174306 MOV #8,SHIFT ;# OF SHIFTS
1591 004614 012767 000377 174656 MOV #377,$TMP1 ;DATA CHAR
1592 004622 004767 012264 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1593 004626 012700 140377 MOV #140000!377,R0 ;EXPECTED DATA PLUS
1594 ;RXERR & OVRUN
1595 004632 017701 175060 MOV @RXDBUF,R1 ;ACTUAL
1596 004636 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1597 004640 001401 BEQ .+4
1598 004642 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1599 ;OVRUN BITS...THEY BOTH SHOULD BE SET
1600
1601 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF THE
1602 ;: RECEIVER SECTION,IT USES THE ERROR FLAGS
1603 ;: TO DETERMINE THAT IT WAS SELECTED CORRECTLY
1604 ;: (OVRUN,RXERR)
1605 ;: MODE:SYNEXT
1606 ;: LENGTH:EIGHT
1607 ;: CHAR:0
1608
1609 ;:*****
1610 004644 000004 †ST6: SCOPE
1611 004646 052777 000400 175052 BIS #MRESET,@TXCSR ;MASTER RESET
1612 004654 012777 020000 175040 MOV #SYNEXT,@PARCSR ;SET THE MODE
1613 004662 052777 000400 175036 BIS #MRESET,@TXCSR ;MASTER RESET
1614
1615 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1616 004670 012777 064001 175030 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1617
1618 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
1619 004676 012777 026000 175016 MOV #SYNEXT,EIGHT!NOPAR!0,@PARCSR
1620 004704 052777 000020 175000 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
1621 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION

```

```

1622 004712 042777 020000 175006 BIC #CLK,@TXCSR ;POKE CLK DOWN
1623 004720 052777 020000 175000 BIS #CLK,@TXCSR ;POKE CLK UP
1624 004726 016703 174764 MOV RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1625 004732 012700 000000 MOV #0,R0 ;EXPECTED
1626 004736 012767 000010 174156 MOV #8,SHIFT ;# OF SHIFTS
1627 004744 012767 000000 174526 MOV #0,$TMP1 ;DATA CHAR
1628 004752 004767 012134 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1629 004758 105777 174730 TSTB @RXCSR ;RXDONE
1630 004762 100401 BMI .+4
1631 004764 104004 ERROR 4 ;RXDONE SHOULD BE SET
1632 004766 017701 174724 MOV @RXDBUF,R1 ;ACTUAL
1633 004772 020001 CMP R0,R1 ;COMPARE EXPECTED VS. ACTUAL
1634 004774 001401 BEQ .+4
1635 004776 104002 ERROR 2 ;RECEIVED DATA DID NOT MATCH
1636 ;EXPECTED DATA - CHECK MAINT DATA
1637 ;OR RECEIVER LOGIC
1638 005000 012767 000010 174114 MOV #8,SHIFT ;# OF SHIFTS
1639 005006 012767 000000 174464 MOV #0,$TMP1 ;DATA CHAR
1640 005014 004767 012072 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1641 ;NOW SHIFT IN A SECOND CHARACTER WITHOUT READING RXDBUF
1642 005020 012767 000010 174074 MOV #8,SHIFT ;# OF SHIFTS
1643 005026 012767 000000 174444 MOV #0,$TMP1 ;DATA CHAR
1644 005034 004767 012052 JSR PC,RPOKE ;SHIFT IN THIS CHAR
1645 005040 012700 140000 MOV #140000!0,R0 ;EXPECTED DATA PLUS
1646 ;RXERR & OVRUN
1647 005044 017701 174646 MOV @RXDBUF,R1 ;ACTUAL
1648 005050 020001 CMP R0,R1 ;COMPARE EXP VS. ACT
1649 005052 001401 BEQ .+4
1650 005054 104002 ERROR 2 ;SPECIFICALLY LOOK AT RXERR &
1651 ;OVRUN BITS...THEY BOTH SHOULD BE SET
1652
1653 ;: THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1654 ;: SECTION, IT USES THE ERROR FLAGS TO DETERMINE
1655 ;: THAT IT WAS SELECTED PROPERLY
1656 ;: FRAME ERROR (FMERR,RXERR)
1657 ;: MODE: ISOC (ISYMOD)
1658 ;: LENGTH:FIVE
1659 ;: CHAR: 25
1660
1661 ;:*****
1662 005056 000004 †ST7: SCOPE
1663 005060 052777 000400 174640 BIS #MRESET,@TXCSR ;MASTER RESET
1664 005066 012777 000000 174626 MOV #ISYMOD,@PARCSR ;SET THE MODE
1665 005074 052777 000400 174624 BIS #MRESET,@TXCSR ;MASTER RESET
1666
1667 ;SET MAINT DATA,CLK BREAK,&MAINTENANCE MODE
1668 005102 012777 064001 174616 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
1669
1670 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
1671 005110 012777 000000 174604 MOV #ISYMOD!FIVE!NOPAR!0,@PARCSR
1672 005116 052777 000020 174566 BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
1673 ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1674 005124 042777 020000 174574 BIC #CLK,@TXCSR ;POKE CLK DOWN
1675 005132 052777 020000 174566 BIS #CLK,@TXCSR ;POKE CLK UP
1676 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1677 005140 042777 020000 174560 BIC #CLK,@TXCSR ;POKE CLK DOWN

```

K03

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 38
DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

1678 005146 052777 020000 174552
1679 005154 012767 003007 173740
1680 005162 012767 000052 174310
1681
1682 005170 004767 011716
1683 005174 016703 174516
1684 005200 012700 120025
1685 005204 017701 174506
1686 005210 020001
1687 005212 001401
1688 005214 104000
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701 005216 000004
1702 005220 052777 000400 174500
1703 005226 012777 000000 174466
1704 005234 052777 000400 174464
1705
1706
1707 005242 012777 064001 174456
1708
1709
1710 005250 012777 002000 174444
1711 005256 052777 000020 174426
1712
1713 005264 042777 020000 174434
1714 005272 052777 020000 174426
1715
1716 005300 042777 020000 174420
1717 005306 052777 020000 174412
1718 005314 012767 000010 173600
1719 005322 012767 000052 174150
1720
1721 005330 004767 011556
1722 005334 016703 174356
1723 005340 012700 120025
1724 005344 017701 174346
1725 005350 020001
1726 005352 001401
1727 005354 104000
1728
1729
1730
1731
1732
1733

BIS :CLK,@TXCSR ;POKE CLK UP
M#7,SHIFT ;# OF SHIFTS
FJV #52,STMP1 ;DATA CHAR
;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
JSR PC,RPOKE ;SHIFT IN THIS CHAR
MOV RXDBUF,R3 ;FOR ERROR MESSAGE
MOV #RXERR!FRMERR!25,R0 ;EXPECTED
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1 ;COMPARE EXP VS ACT
BEQ +4
ERROR ;FRAME ERROR 4 & RX ERROR SHOULD BE SET
;IF LOWER BYTE DOES NOT MATCH IT
;PROBABLY IS A LENGTH SELECT PROBLEM

:: THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
:: SECTION , IT USES THE ERROR FLAGS TO DETERMINE
:: THAT IT WAS SELECTED PROPERLY
:: FRAME ERROR (FRMERR,RXERR)
:: MODE: ISOC (ISYMOD)
:: LENGTH: SIX
:: CHAR: 25

::*****

↑ST10: SCOPE
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #ISYMOD,@PARCSR ;SET THE MODE
BIS #MRESET,@TXCSR ;MASTER RESET

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

;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
MOV #ISYMOD!SIX!NOPAR!0,@PARCSR
BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....

BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRIZATION

BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
MOV #8,SHIFT ;# OF SHIFTS
MOV #52,STMP1 ;DATA CHAR

;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
JSR PC,RPOKE ;SHIFT IN THIS CHAR
MOV RXDBUF,R3 ;FOR ERROR MESSAGE
MOV #RXERR!FRMERR!25,R0 ;EXPECTED
MOV @RXDBUF,R1 ;ACTUAL
CMP R0,R1 ;COMPARE EXP VS ACT
BEQ +4
ERROR ;FRAME ERROR 4 & RX ERROR SHOULD BE SET
;IF LOWER BYTE DOES NOT MATCH IT
;PROBABLY IS A LENGTH SELECT PROBLEM

:: THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
:: SECTION , IT USES THE ERROR FLAGS TO DETERMINE
:: THAT IT WAS SELECTED PROPERLY

```

1734                                     ;; FRAME ERROR (FRMERR,RXERR)
1735                                     ;; MODE:ISOC (ISYMOD)
1736                                     ;; LENGTH:SEVEN
1737                                     ;; CHAR: 125
1738                                     ;;
1739                                     ;:*****
1740 005356 000004                               †ST11: SCOPE
1741 005360 052777 000400 174340             BIS      #MRESET,@TXCSR ;MASTER RESET
1742 005366 012777 000000 174326             MOV      #ISYMOD,@PARCSR ;SET THE MODE
1743 005374 052777 000400 174324             BIS      #MRESET,@TXCSR ;MASTER RESET
1744
1745                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1746 005402 012777 064001 174316             MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1747
1748                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1749 005410 012777 004000 174304             MOV      #ISYMOD!SEVEN!NOPAR!0,@PARCSR
1750 005416 052777 000020 174266             BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1751                                     ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1752 005424 042777 020000 174274             BIC      #CLK,@TXCSR ;POKE CLK DOWN
1753 005432 052777 020000 174266             BIS      #CLK,@TXCSR ;POKE CLK UP
1754                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1755 005440 042777 020000 174260             BIC      #CLK,@TXCSR ;POKE CLK DOWN
1756 005446 052777 020000 174252             BIS      #CLK,@TXCSR ;POKE CLK UP
1757 005454 012767 000011 173440             MOV      #9,SHIFT ;# OF SHIFTS
1758 005462 012767 000252 174010             MOV      #252,$TMP1 ;DATA CHAR
1759                                     ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1760 005470 004767 011416                               JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1761 005474 016703 174216                               MOV      RXDBUF,R3 ;FOR ERROR MESSAGE
1762 005500 012700 120125                               MOV      #RXERR!FRMERR!125,R0 ;EXPECTED
1763 005504 017701 174206                               MOV      @RXDBUF,R1 ;ACTUAL
1764 005510 020001                               CMP      R0,R1 ;COMPARE EXP VS ACT
1765 005512 001401                               BEQ      +4
1766 005514 104000                               ERROR   ;FRAME ERROR 4 & RX ERROR SHOULD BE SET
1767                                     ;IF LOWER BYTE DOES NOT MATCH IT
1768                                     ;PROBABLY IS A LENGTH SELECT PROBLEM
1769
1770                                     ;; THIS TEST VERIFYS WORD LENGTH SELECT OF RECEIVER
1771                                     ;; SECTION ,IT USES THE ERROR FLAGS TO DETERMINE
1772                                     ;; THAT IT WAS SELECTED PROPERLY
1773                                     ;; FRAME ERROR (FRMERR,RXERR)
1774                                     ;; MODE:ISOC (ISYMOD)
1775                                     ;; LENGTH:EIGHT
1776                                     ;; CHAR: 125
1777                                     ;;
1778                                     ;:*****
1779 005516 000004                               †ST12: SCOPE
1780 005520 052777 000400 174200             BIS      #MRESET,@TXCSR ;MASTER RESET
1781 005526 012777 000000 174166             MOV      #ISYMOD,@PARCSR ;SET THE MODE
1782 005534 052777 000400 174164             BIS      #MRESET,@TXCSR ;MASTER RESET
1783
1784                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1785 005542 012777 064001 174156             MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1786
1787                                     ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1788 005550 012777 006000 174144             MOV      #ISYMOD!EIGHT!NOPAR!0,@PARCSR
1789 005556 052777 000020 174126             BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH

```

```

1790          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1791 005564 042777 020000 174134      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1792 005572 052777 020000 174126      BIS    #CLK,@TXCSR      ;POKE CLK UP
1793          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1794 005600 042777 020000 174120      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1795 005606 052777 020000 174112      BIS    #CLK,@TXCSR      ;POKE CLK UP
1796 005614 012767 000012 173300      MOV    #10,SHIFT        ;# OF SHIFTS
1797 005622 012767 000252 173650      MOV    #252,$TMP1       ;DATA CHAR
1798          ;NOTE: THE ABOVE CHARACTER IS MISSING STOP BIT
1799 005630 004767 011256          JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
1800 005634 016703 174056          MOV    @RXDBUF,R3       ;FOR ERROR MESSAGE
1801 005640 012700 120125          MOV    @RXERR!FAMERR!125,R0 ;EXPECTED
1802 005644 017701 174046          MOV    @RXDBUF,R1       ;ACTUAL
1803 005650 020001          CMP    R0,R1           ;COMPARE EXP VS ACT
1804 005652 001401          BEQ    .+4
1805 005654 104000          ERROR  ;FRAME ERROR    4 & RX ERROR SHOULD BE SET
1806          ;IF LOWER BYTE DOES NOT MATCH IT
1807          ;PROBABLY IS A LENGTH SELECT PROBLEM
1808
1809          ;; THIS TEST VERIFYS EVEPAR PARITY SENSE
1810          ;; OF THE RECEIVER
1811          ;; MODE: ISOC (ISYMOD)
1812          ;; PARITY: EVEPAR
1813          ;; LENGTH: FIVE PLUS PARITY
1814          ;; CHAR: 25
1815          ;
1816          ;*****
1817 005656 000004          ;ST13: SCOPE
1818 005660 052777 000400 174040      BIS    #MRESET,@TXCSR   ;MASTER RESET
1819 005666 012777 000000 174026      MOV    #ISYMOD,@PARCSR  ;SET THE MODE
1820 005674 052777 000400 174024      BIS    #MRESET,@TXCSR   ;MASTER RESET
1821
1822          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1823 005702 012777 064001 174016      MOV    @MTDATA!CLK!MINT!BREAK,@TXCSR
1824
1825          ;SET MODE ,# OF BITS,PARITY SENSE,&LOAD SYNC REG
1826 005710 012777 001400 174004      MOV    #ISYMOD!FIVE!EVEPAR!0,@PARCSR
1827 005716 052777 000020 173766      BIS    #SYNSCH,@RXCSR   ;SET SYNC SEARCH
1828          ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
1829 005724 042777 020000 173774      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1830 005732 052777 020000 173766      BIS    #CLK,@TXCSR      ;POKE CLK UP
1831          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1832 005740 042777 020000 173760      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
1833 005746 052777 020000 173752      BIS    #CLK,@TXCSR      ;POKE CLK UP
1834 005754 016703 173736          MOV    @RXDBUF,R3       ;SET UP FOR ERROR MESSAGE
1835 005760 012700 110025          MOV    @RXERR!PARER!25,R0 ;EXPECTED
1836 005764 012767 000010 173130      MOV    #8,SHIFT         ;# OF SHIFTS
1837 005772 012767 000252 173500      MOV    #252,$TMP1       ;DATA CHAR
1838 006000 004767 011106          JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
1839 006004 105777 173702          TSTB  @RXCSR ;RXDONE
1840 006010 100401          BMI    .+4
1841 006012 104004          ERROR  4 ;RXDONE SHOULD BE ASSERTED
1842 006014 017701 173676          MOV    @RXDBUF,R1       ;ACTUAL
1843 006020 020001          CMP    R0,R1           ;COMPARE EXP VS. ACT
1844 006022 001401          BEQ    .+4
1845 006024 104000          ERROR  ;PARITY ERROR    4 &RXERR SHOULD BE SET
    
```

N03

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 41
 DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

```

1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858 006026 000004
1859 006030 052777 000400 173670
1860 006036 012777 000000 173656
1861 006044 052777 000400 173654
1862
1863
1864 006052 012777 064001 173646
1865
1866
1867 006060 012777 003400 173634
1868 006066 052777 000020 173616
1869
1870 006074 042777 020000 173624
1871 006102 052777 020000 173616
1872
1873 006110 042777 020000 173610
1874 006116 052777 020000 173602
1875 006124 016703 173566
1876 006130 012700 110025
1877 006134 012767 000011 172760
1878 006142 012767 000452 173330
1879 006150 004767 010736
1880 006154 105777 173532
1881 006160 100401
1882 006162 104004
1883 006164 017701 173526
1884 006170 020001
1885 006172 001401
1886 006174 104000
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899 006176 000004
1900 006200 052777 000400 173520
1901 006206 012777 000000 173506
  
```

```

;NOTE THAT THE PARITY BIT SHOULD
;SHOW UP IN THE DATA
;IE. BIT FIVE FOR FIVE LEVEL CODE
  
```

```

; THIS TEST VERIFYS EVEPAR PARITY SENSE
; OF THE RECEIVER
; MODE: ISOC (ISYMOD)
; PARITY: EVEPAR
; LENGTH: SIX PLUS PARITY
; CHAR: 25
  
```

```

†ST14: SCOPE
      BIS      #MRESET,@TXCSR ;MASTER RESET
      MOV      #ISYMOD,@PARCSR ;SET THE MODE
      BIS      #MRESET,@TXCSR ;MASTER RESET

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

;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
      MOV      #ISYMOD!SIX!EVEPAR!0,@PARCSR
      BIS      #SYNSCH,@TXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
      MOV      #RXERR!PARER!25,R0 ;EXPECTED
      MOV      #9,SHIFT ;# OF SHIFTS
      MOV      #452,STMP1 ;DATA CHAR
      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
      TSTB    @RXCSR ;RXDONE
      BMI     .+4
      ERROR   4 ;RXDONE SHOULD BE ASSERTED
      MOV     @RXDBUF,R1 ;ACTUAL
      CMP     R0,R1 ;COMPARE EXP VS. ACT
      BEQ     .+4
      ERROR   ;PARITY ERROR 4 &RXERR SHOULD BE SET
  
```

```

;NOTE THAT THE PARITY BIT SHOULD
;SHOW UP IN THE DATA
;IE. BIT SIX FOR SIX LEVEL CODE
  
```

```

; THIS TEST VERIFYS EVEPAR PARITY SENSE
; OF THE RECEIVER
; MODE: ISOC (ISYMOD)
; PARITY: EVEPAR
; LENGTH: SEVEN PLUS PARITY
; CHAR: 325
  
```

```

†ST15: SCOPE
      BIS      #MRESET,@TXCSR ;MASTER RESET
      MOV      #ISYMOD,@PARCSR ;SET THE MODE
  
```



```

1902 006214 052777 000400 173504      BIS      #MRESET,@TXCSR ;MASTER RESET
1903
1904                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1905 006222 012777 064001 173476      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1906
1907                                     ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
1908 006230 012777 005400 173464      MOV      #ISYMOD!SEVEN!EVEPAR!0,@PARCSR
1909 006236 052777 000020 173446      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1910                                     ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1911 006244 042777 020000 173454      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1912 006252 052777 020000 173446      BIS      #CLK,@TXCSR ;POKE CLK UP
1913                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1914 006260 042777 020000 173440      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1915 006266 052777 020000 173432      BIS      #CLK,@TXCSR ;POKE CLK UP
1916 006274 016703 173416      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE
1917 006300 012700 110325      MOV      #RXERR!PARER!325,R0 ;EXPECTED
1918 006304 012767 000012 172610      MOV      #10,SHIFT ;# OF SHIFTS
1919 006312 012767 001652 173160      MOV      #1652,$TMP1 ;DATA CHAR
1920 006320 004767 010566      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
1921 006324 105777 173362      TSTB    @RXCSR ;RXDONE
1922 006330 100401      BMI     .+4
1923 006332 104004      ERROR   4 ;RXDONE SHOULD BE ASSERTED
1924 006334 017701 173356      MOV     @RXDBUF,R1 ;ACTUAL
1925 006340 020001      CMP     R0,R1 ;COMPARE EXP VS. ACT
1926 006342 001401      BEQ     .+4
1927 006344 104000      ERROR   ;PARITY ERROR 4 &RXERR SHOULD BE SET
1928                                     ;NOTE THAT THE PARITY BIT SHOULD
1929                                     ;SHOW UP IN THE DATA
1930                                     ;IE. BIT SEVEN FOR SEVEN LEVEL CODE
1931
1932                                     ;; THIS TEST VERIFYS EVEPAR PARITY SENSE
1933                                     ;; OF THE RECEIVER
1934                                     ;; MODE:ISOC (ISYMOD)
1935                                     ;; PARITY:EVEPAR
1936                                     ;; LENGTH:EIGHT PLUS PARITY
1937                                     ;; CHAR: 125
1938
1939                                     ;*****
1940 006346 000004      ST16:  SCOPE
1941 006350 052777 000400 173350      BIS      #MRESET,@TXCSR ;MASTER RESET
1942 006356 012777 000000 173336      MOV      #ISYMOD,@PARCSR ;SET THE MODE
1943 006364 052777 000400 173334      BIS      #MRESET,@TXCSR ;MASTER RESET
1944
1945                                     ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
1946 006372 012777 064001 173326      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
1947
1948                                     ;SET MODE , # OF BITS,PARITY SENSE &LOAD SYNC REG
1949 006400 012777 007400 173314      MOV      #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
1950 006406 052777 000020 173276      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
1951                                     ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
1952 006414 042777 020000 173304      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1953 006422 052777 020000 173276      BIS      #CLK,@TXCSR ;POKE CLK UP
1954                                     ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
1955 006430 042777 020000 173270      BIC      #CLK,@TXCSR ;POKE CLK DOWN
1956 006436 052777 020000 173262      BIS      #CLK,@TXCSR ;POKE CLK UP
1957 006444 016703 173246      MOV      RXDBUF,R3 ;SET UP FOR ERROR MESSAGE

```

1958	006450	012700	110125		MOV	#RXERR!PARER!125,RO	: EXPECTED
1959	006454	012767	000013	172440	MOV	#11,SHIFT	: # OF SHIFTS
1960	006462	012767	003252	173010	MOV	#3252,\$TMP1	: DATA CHAR
1961	006470	004767	010416		JSR	PC,RPOKE	: SHIFT IN THIS CHAR
1962	006474	105777	173212		TSTB	@RXCSR ;RXDONE	
1963	006500	100401			BMI	:+4	
1964	006502	104004			ERROR	4	: RXDONE SHOULD BE ASSERTED
1965	006504	017701	173206		MOV	@RXDBUF,R1	: ACTUAL
1966	006510	020001			CMP	RO,R1 ;COMPARE	EXP VS. ACT
1967	006512	001401			BEQ	:+4	
1968	006514	104000			ERROR	;PARITY ERROR 4 &RXERR	SHOULD BE SET
1969							
1970							
1971							
1972							
1973							
1974							
1975							
1976							
1977							
1978	006516	000004			*****		
1979	006520	052777	000400	173200	†ST17: SCOPE		
1980	006526	012777	000000	173166	BIS	#MRESET,@TXCSR	: MASTER RESET
1981	006534	052777	000400	173164	MOV	#ISYMOD,@PARCSR	: SET THE MODE
1982					BIS	#MRESET,@TXCSR	: MASTER RESET
1983							
1984	006542	012777	064001	173156			
1985							
1986							
1987	006550	012777	001000	173144			
1988	006556	052777	000020	173126			
1989							
1990	006564	042777	020000	173134			
1991	006572	052777	020000	173126			
1992							
1993	006600	042777	020000	173120			
1994	006606	052777	020000	173112			
1995	006614	016703	173076				
1996	006620	012700	110065				
1997	006624	012767	000010	172270			
1998	006632	012767	000352	172640			
1999	006640	004767	010246				
2000	006644	105777	173042				
2001	006650	100401					
2002	006652	104004					
2003	006654	017701	173036				
2004	006660	020001					
2005	006662	001401					
2006	006664	104000					
2007							
2008							
2009							
2010							
2011							
2012							
2013							

; THIS TEST VERIFYS ODDPAR PARITY SENSE
 ; OF THE RECEIVER
 ; MODE: ISOC (ISYMOD)
 ; PARITY: ODDPAR
 ; LENGTH: FIVE PLUS PARITY
 ; CHAR: 65
 ; *****
 ; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
 ; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
 ; POKE CLK TO GET RECEIVER INTO SYNCHRONIZATION....
 ; POKE CLK TO GET LOGIC INTO SYNCHRONIZATION
 ; SET UP FOR ERROR MESSAGE
 ; NOTE THAT THE PARITY BIT SHOULD
 ; SHOW UP IN THE DATA
 ; IE. BIT FIVE FOR FIVE LEVEL CODE
 ; THIS TEST VERIFYS ODDPAR PARITY SENSE
 ; OF THE RECEIVER
 ; MODE: ISOC (ISYMOD)

```

2014          ;; PARITY: ODDPAR
2015          ;; LENGTH: SIX PLUS PARITY
2016          ;; CHAR: 125
2017          ;;
2018          ;; *****
2019 006666 000004          †ST20: SCOPE
2020 006670 052777 000400 173030  BIS      #MRESET,@TXCSR ; MASTER RESET
2021 006676 012777 000000 173016  MOV      #ISYMOD,@PARCSR ; SET THE MODE
2022 006704 052777 000400 173014  BIS      #MRESET,@TXCSR ; MASTER RESET
2023          ;;
2024          ;; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2025 006712 012777 064001 173006  MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2026          ;;
2027          ;; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2028 006720 012777 003000 172774  MOV      #ISYMOD!SIX!ODDPAR!D,@PARCSR
2029 006726 052777 000020 172756  BIS      #SYNSCH,@RXCSR ; SET SYNC SEARCH
2030          ;; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2031 006734 042777 020000 172764  BIC      #CLK,@TXCSR ; POKE CLK DOWN
2032 006742 052777 020000 172756  BIS      #CLK,@TXCSR ; POKE CLK UP
2033          ;; POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2034 006750 042777 020000 172750  BIC      #CLK,@TXCSR ; POKE CLK DOWN
2035 006756 052777 020000 172742  BIS      #CLK,@TXCSR ; POKE CLK UP
2036 006764 016703 172726  MOV      RXDBUF,R3 ; SET UP FOR ERROR MESSAGE
2037 006770 012700 110125  MOV      #RXERR!PARER!125,R0 ; EXPECTED
2038 006774 012767 000011 172120  MOV      #9,SHIFT ; # OF SHIFTS
2039 007002 012767 000652 172470  MOV      #652,$TMP1 ; DATA CHAR
2040 007010 004767 010076  JSR      PC,RPOKE ; SHIFT IN THIS CHAR
2041 007014 105777 172672  TSTB    @RXCSR ; RXDONE
2042 007020 100401  BMI      .+4
2043 007022 104004  ERROR    4 ; RXDONE SHOULD BE ASSERTED
2044 007024 017701 172666  MOV      @RXDBUF,R1 ; ACTUAL
2045 007030 020001  CMP      R0,R1 ; COMPARE EXP VS. ACT
2046 007032 001401  BEQ      .+4
2047 007034 104000  ERROR    ; PARITY ERROR 4 & RXERR SHOULD BE SET
2048          ;; NOTE THAT THE PARITY BIT SHOULD
2049          ;; SHOW UP IN THE DATA
2050          ;; IE. BIT SIX FOR SIX LEVEL CODE
2051          ;;
2052          ;; THIS TEST VERIFYS ODDPAR PARITY SENSE
2053          ;; OF THE RECEIVER
2054          ;; MODE: ISOC (ISYMOD)
2055          ;; PARITY: ODDPAR
2056          ;; LENGTH: SEVEN PLUS PARITY
2057          ;; CHAR: 125
2058          ;;
2059          ;; *****
2060 007036 000004          †ST21: SCOPE
2061 007040 052777 000400 172660  BIS      #MRESET,@TXCSR ; MASTER RESET
2062 007046 012777 000000 172646  MOV      #ISYMOD,@PARCSR ; SET THE MODE
2063 007054 052777 000400 172644  BIS      #MRESET,@TXCSR ; MASTER RESET
2064          ;;
2065          ;; SET MAINT DATA, CLK, BREAK, & MAINTENANCE MODE
2066 007062 012777 064001 172636  MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2067          ;;
2068          ;; SET MODE, # OF BITS, PARITY SENSE, & LOAD SYNC REG
2069 007070 012777 005000 172624  MOV      #ISYMOD!SEVEN!ODDPAR!D,@PARCSR

```

E04

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 45
 DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

2070	007076	052777	000020	172606	BIS	#SYNSCH,@RXCSR	;SET SYNC SEARCH
2071							;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2072	007104	042777	020000	172614	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2073	007112	052777	020000	172606	BIS	#CLK,@TXCSR	;POKE CLK UP
2074							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2075	007120	042777	020000	172600	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2076	007126	052777	020000	172572	BIS	#CLK,@TXCSR	;POKE CLK UP
2077	007134	016703	172556		MOV	RXDBUF,R3	;SET UP FOR ERROR MESSAGE
2078	007140	012700	110125		MOV	#RXERR!PARER!125,R0	;EXPECTED
2079	007144	012767	000012	171750	MOV	#10,SHIFT	;# OF SHIFTS
2080	007152	012767	001252	172320	MOV	#1252,\$TMP1	;DATA CHAR
2081	007160	004767	007726		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2082	007164	105777	172522		TSTB	@RXCSR ;RXDONE	
2083	007170	100401			BMI	+.4	
2084	007172	104004			ERROR	4	;RXDONE SHOULD BE ASSERTED
2085	007174	017701	172516		MOV	@RXDBUF,R1	;ACTUAL
2086	007200	020001			CMP	R0,R1 ;COMPARE	EXP VS. ACT
2087	007202	001401			BEQ	+.4	
2088	007204	104000			ERRJR	;PARITY ERROR 4 &RXERR SHOULD BE SET	
2089							;NOTE THAT THE PARITY BIT SHOULD
2090							;SHOW UP IN THE DATA
2091							;IE. BIT SEVEN FOR SEVEN LEVEL CODE
2092							
2093							::THIS TEST VERIFYS ODDPAR PARITY SENSE
2094							:::OF THE RECEIVER
2095							:::MODE:ISOC (ISYMOD)
2096							:::PARITY:ODDPAR
2097							:::LENGTH:EIGHT PLUS PARITY
2098							:::CHAR: 125
2099							:::*****
2100							:::*****
2101	007206	000004			ST22:	SCOPE	
2102	007210	052777	000400	172510	BIS	#MRESET,@TXCSR	;MASTER RESET
2103	007216	012777	000000	172476	MOV	#ISYMOD,@PARCSR	;SET THE MODE
2104	007224	052777	000400	172474	BIS	#MRESET,@TXCSR	;MASTER RESET
2105							
2106							;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2107	007232	012777	064001	172466	MOV	#MTDATA!CLK!MINT!BREAK,@TXCSR	
2108							
2109							;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
2110	007240	012777	007000	172454	MOV	#ISYMOD!EIGHT!ODDPAR!0,@PARCSR	
2111	007246	052777	000020	172436	BIS	#SYNSCH,@RXCSR	;SET SYNC SEARCH
2112							;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2113	007254	042777	020000	172444	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2114	007262	052777	020000	172436	BIS	#CLK,@TXCSR	;POKE CLK UP
2115							;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2116	007270	042777	020000	172430	BIC	#CLK,@TXCSR	;POKE CLK DOWN
2117	007276	052777	020000	172422	BIS	#CLK,@TXCSR	;POKE CLK UP
2118	007304	016703	172406		MOV	RXDBUF,R3	;SET UP FOR ERROR MESSAGE
2119	007310	012700	110125		MOV	#RXERR!PARER!125,R0	;EXPECTED
2120	007314	012767	000013	171600	MOV	#11,SHIFT	;# OF SHIFTS
2121	007322	012767	002252	172150	MOV	#2252,\$TMP1	;DATA CHAR
2122	007330	004767	007556		JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2123	007334	105777	172352		TSTB	@RXCSR ;RXDONE	
2124	007340	100401			BMI	+.4	
2125	007342	104004			ERROR	4	;RXDONE SHOULD BE ASSERTED

F04

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 46
 DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

```

2126 007344 017701 172346      MOV    @RXDBUF,R1      ;ACTUAL
2127 007350 020001              CMP    R0,R1          ;COMPARE EXP VS. ACT
2128 007352 001401              BEQ    +4
2129 007354 104000              ERROR  ;PARITY ERROR 4 &RXERR SHOULD BE SET
2130
2131                      ;; THIS TEST PERFORMS BINARY DATA CHECK ON THE
2132                      ;; RECEIVER
2133                      ;; LENGTH:EIGHT PLUS PARITY
2134                      ;; MODE:ISYMOD
2135                      ;; PARITY:EVEPAR
2136
2137                      ;:*****
2138 007356 000004      †ST23: SCOPE
2139 007360 052777 000400 172340      BIS    #MRESET,@TXCSR ;MASTER RESET
2140 007366 012777 000000 172326      MOV    #ISYMOD,@PARCSR ;SET THE MODE
2141 007374 052777 000400 172324      BIS    #MRESET,@TXCSR ;MASTER RESET
2142
2143                      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2144 007402 012777 064001 172315      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2145
2146                      ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2147 007410 012777 007400 172304      MOV    #ISYMOD!EIGHT!EVEPAR!0,@PARCSR
2148 007416 052777 000020 172266      BIS    #SYNSCH,@TXCSR ;SET SYNC SEARCH
2149                      ;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2150 007424 042777 020000 172274      BIC    #CLK,@TXCSR ;POKE CLK DOWN
2151 007432 052777 020000 172266      BIS    #CLK,@TXCSR ;POKE CLK UP
2152                      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2153 007440 042777 020000 172260      BIC    #CLK,@TXCSR ;POKE CLK DOWN
2154 007446 052777 020000 172252      BIS    #CLK,@TXCSR ;POKE CLK UP
2155 007454 016703 172236      MOV    RXDBUF,R3 ;SET UP ERROR MESSAGE
2156 007460 005004              CLR    R4 ;DATA CHAR
2157 007462 010400              MOV    R4,R0 ;EXPECTED
2158 007464 012767 000013 171430      MOV    #11,SHIFT ;# OF SHIFTS
2159 007472 010467 172002              MOV    R4,$TMP1 ;"TO BE SHIFTED CHARACTER"
2160 007476 004767 007562              JSR    PC,EVENB ;CALC PARITY
2161 007502 000241              CLC
2162 007504 006167 171770              ROL    $TMP1 ;GENERATE START BIT
2163 007510 052767 002000 171762      BIS    #BIT10,$TMP1 ;GENERATE STOP BIT
2164                      ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2165 007516 004767 007370              JSR    PC,RPOKE ;SHIFT IN THIS CHAR
2166 007522 017701 172170              MOV    @RXDBUF,R1 ;ACTUAL
2167 007526 020001              CMP    R0,R1 ;COMPARE EXP VS ACT
2168 007530 001401              BEQ    +4
2169 007532 104002              ERROR  2 ;DATA CHARS SHOULD MATCH
2170                      ;THERE SHOULD BE NO PARITY ERROR
2171 007534 005204              INC    R4 ;UPGRADE NEXT CHAR
2172 007536 105704              TSTB  R4 ;LAST CHAR ?
2173 007540 001350              BNE    1$
2174
2175                      ;; THIS TEST PERFORMS BINARY DATA CHECK ON THE
2176                      ;; RECEIVER
2177                      ;; LENGTH:EIGHT PLUS PARITY
2178                      ;; MODE:ISYMOD
2179                      ;; PARITY:ODDPAR
2180
2181                      ;:*****

```

```

2182 007542 000004          TST24: SCOPE
2183 007544 052777 000400 172154   BIS      #MRESET,@TXCSR ;MASTER RESET
2184 007552 012777 000000 172142   MOV      #ISYMOD,@PARCSR ;SET THE MODE
2185 007560 052777 000400 172140   BIS      #MRESET,@TXCSR ;MASTER RESET
2186
2187 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2188 007566 012777 064001 172132   MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2189
2190 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2191 007574 012777 007000 172120   MOV      #ISYMOD!EIGHT!000PAR!0,@PARCSR
2192 007602 052777 000020 172102   BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
2193 ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2194 007610 042777 020000 172110   BIC      #CLK,@TXCSR ;POKE CLK DOWN
2195 007616 052777 020000 172102   BIS      #CLK,@TXCSR ;POKE CLK UP
2196 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2197 007624 042777 020000 172074   BIC      #CLK,@TXCSR ;POKE CLK DOWN
2198 007632 052777 020000 172066   BIS      #CLK,@TXCSR ;POKE CLK UP
2199 007640 016703 172052          MOV      RXDBUF,R3 ;SET UP ERROR MESSAGE
2200 007644 005004          CLR      R4 ;DATA CHAR
2201 007646 010400          MOV      R4,R0 ;EXPECTED
2202 007650 012767 000013 171244   MOV      #11,SHIFT ;# OF SHIFTS
2203 007656 010467 171616          MOV      R4,$TMP1 ;"TO BE SHIFTED CHARACTER"
2204 007662 004767 007312          JSR      PC,0008 ;CALC PARITY
2205 007666 000241          CLC
2206 007670 006167 171604          ROL      $TMP1 ;GENERATE START BIT
2207 007674 052767 002000 171576   BIS      #BIT10,$TMP1 ;GENERATE STOP BIT
2208 ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2209 007702 004767 007204          JSR      PC,RPOKE ;SHIFT IN THIS CHAR
2210 007706 017701 172004          MOV      @RXDBUF,R1 ;ACTUAL
2211 007712 020001          CMP      R0,R1 ;COMPARE EXP VS ACT
2212 007714 001401          BEQ      +4
2213 007716 104002          ERROR  2 ;DATA CHARS SHOULD MATCH
2214 ;THERE SHOULD BE NO PARITY ERROR
2215 007720 005204          INC      R4 ;UPGRADE NEXT CHAR
2216 007722 105704          TSTB    R4 ;LAST CHAR ?
2217 007724 001350          BNE      15
2218
2219 ;: THIS TEST PERFORMS BINARY DATA CHECK ON THE
2220 ;: RECEIVER
2221 ;: LENGTH:EIGHT PLUS PARITY
2222 ;: MODE:SYNEXT
2223 ;: PARITY:EVEPAR
2224 ;:
2225 ;:*****
2226 007726 000004          *ST25: SCOPE
2227 007730 052777 000400 171770   BIS      #MRESET,@TXCSR ;MASTER RESET
2228 007736 012777 020000 171756   MOV      #SYNEXT,@PARCSR ;SET THE MODE
2229 007744 052777 000400 171754   BIS      #MRESET,@TXCSR ;MASTER RESET
2230
2231 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2232 007752 012777 064001 171746   MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
2233
2234 ;SET MODE , # OF BITS,PARITY SENSE,&LOAD SYNC REG
2235 007760 012777 027400 171734   MOV      #SYNEXT!EIGHT!EVEPAR!0,@PARCSR
2236 007766 052777 000020 171716   BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
2237 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION

```

H04

```

2238 007774 042777 020000 171724      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2239 010002 052777 020000 171716      BIS    #CLK,@TXCSR      ;POKE CLK UP
2240 010010 016703 171702      MOV    RXDBUF,R3        ;SET UP ERROR MESSAGE
2241 010014 005004      CLR    R4               ;DATA CHAR
2242 010016 010400      MOV    R4,R0            ;EXPECTED
2243 010020 012767 000011 171074      MOV    #9,SHIFT         ;# OF SHIFTS
2244 010026 010467 171446      MOV    R4,$TMP1         ;"TO BE SHIFTED CHARACTER"
2245 010032 004767 007226      JSR    PC,EVENB         ;CALC PARITY
2246      ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2247 010036 004767 007050      JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2248 010042 017701 171650      MOV    @RXDBUF,R1       ;ACTUAL
2249 010046 020001      CMP    R0,R1            ;COMPARE EXP VS ACT
2250 010050 001401      BEQ    .+4
2251 010052 104002      ERROR  2               ;DATA CHARS SHOULD MATCH
2252      ;THERE SHOULD BE NO PARITY ERROR
2253 010054 005204      INC    R4               ;UPGRADE NEXT CHAR
2254 010056 105704      TSTB  R4               ;LAST CHAR ?
2255 010060 001356      BNE    1$
2256
2257      ;: THIS TEST PERFORMS BINARY DATA CHECK ON THE
2258      ;: RECEIVER
2259      ;: LENGTH:EIGHT PLUS PARITY
2260      ;: MODE:SYNEXT
2261      ;: PARITY:00DPAR
2262
2263      ;:*****
2264 010062 000004      $T26: SCOPE
2265 010064 052777 000400 171634      BIS    #MRESET,@TXCSR  ;MASTER RESET
2266 010072 012777 020000 171622      MOV    #SYNEXT,@PARCSR ;SET THE MODE
2267 010100 052777 000400 171620      BIS    #MRESET,@TXCSR  ;MASTER RESET
2268
2269      ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2270 010106 012777 064001 171612      MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2271
2272      ;SET MODE, # OF BITS,PARITY SENSE &LOAD SYNC REG
2273 010114 012777 027000 171600      MOV    #SYNEXT!EIGHT!00DPAR!0,@PARCSR
2274 010122 052777 000020 171562      BIS    #SYNSCH,@RXCSR  ;SET SEARCH SYNC
2275      ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2276 010130 042777 020000 171570      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2277 010136 052777 020000 171562      BIS    #CLK,@TXCSR      ;POKE CLK UP
2278 010144 016703 171546      MOV    RXDBUF,R3        ;SET UP ERROR MESSAGE
2279 010150 005004      CLR    R4               ;DATA CHAR
2280 010152 010400      MOV    R4,R0            ;EXPECTED
2281 010154 012767 000011 170740      MOV    #9,SHIFT         ;# OF SHIFTS
2282 010162 010467 171312      MOV    R4,$TMP1         ;"TO BE SHIFTED CHARACTER"
2283 010166 004767 007006      JSR    PC,ODDB         ;CALC PARITY
2284      ;$TMP1 NOW HAS CHARACTER TO BE POKED INTO RECEIVER
2285 010172 004767 006714      JSR    PC,RPOKE        ;SHIFT IN THIS CHAR
2286 010176 017701 171514      MOV    @RXDBUF,R1       ;ACTUAL
2287 010202 020001      CMP    R0,R1            ;COMPARE EXP VS ACT
2288 010204 001401      BEQ    .+4
2289 010206 104002      ERROR  2               ;DATA CHARS SHOULD MATCH
2290      ;THERE SHOULD BE NO PARITY ERROR
2291 010210 005204      INC    R4               ;UPGRADE NEXT CHAR
2292 010212 105704      TSTB  R4               ;LAST CHAR ?
2293 010214 001356      BNE    1$

```

```

2294
2295
2296
2297
2298
2299
2300
2301
2302
2303 010216 000004
2304 010220 052777 000400 171500
2305 010226 012777 000000 171466
2306 010234 052777 000400 171464
2307
2308
2309 010242 012777 064001 171456
2310
2311
2312 010250 012777 000026 171444
2313 010256 052777 000020 171426
2314
2315 010264 042777 020000 171434
2316 010272 052777 020000 171426
2317
2318 010300 042777 020000 171420
2319 010306 052777 020000 171412
2320 010314 052777 000400 171370
2321 010322 012767 000003 170574
2322 010330 012767 000154 171142
2323 010336 012767 000007 170556
2324 010344 004767 006542
2325 010350 105777 171336
2326 010354 100001
2327 010356 104004
2328 010360 005367 170540
2329 010364 001361
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339 010366 000004
2340 010370 052777 000400 171330
2341 010376 012777 000000 171316
2342 010404 052777 000400 171314
2343
2344
2345 010412 012777 064001 171306
2346
2347
2348 010420 012777 002026 171274
2349 010426 052777 000020 171256

```

```

:: THIS TEST CHECKS THE STRIP SYNC FUNCTION
:: OF THE RECEIVER LOGIC
:: MODE: ISYMOD
:: LENGTH: FIVE
:: NOTE: RXDONE SHOULD NEVER ASSERT
:: CHAR: 26 (SYNC)
:: *****
↑ST27: SCOPE
BIS #MRESET,@TXCSR ; MASTER RESET
MOV #ISYMOD,@PARCSR ; SET THE MODE
BIS #MRESET,@TXCSR ; MASTER RESET

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

; SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
MOV #ISYMOD!FIVE!NOPAR!26,@PARCSR
BIS #SYNSCH,@RXCSR ; SET SYNC SEARCH
; POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
BIC #CLK,@TXCSR ; POKE CLK DOWN
BIS #CLK,@TXCSR ; POKE CLK UP
; POKE CLK TO GET LOGIC INTO SYNCRIZATION
BIC #CLK,@TXCSR ; POKE CLK DOWN
BIS #CLK,@TXCSR ; POKE CLK UP
BIS #STPSYN,@RXCSR ; SET STRIP SYNC
MOV #3,COUNT ; # OF SYNC CHARS
15: MOV #154,$TMP1 ; CHAR TO BE SHIFTED
MOV #7,SHIFT ; # OF SHIFTS
JSR PC,RPOKE ; SHIFT IN THIS CHAR
TSTB @RXCSR ; RXDONE
BPL .+4
ERROR 4 ; RXDONE SHOULD NOT BE ASSERTED
DEC COUNT ; # OF SYNC CHARS
BNE 15

:: THIS TEST CHECKS THE STRIP SYNC FUNCTION
:: OF THE RECEIVER LOGIC
:: MODE: ISYMOD
:: LENGTH: SIX
:: NOTE: RXDONE SHOULD NEVER ASSERT
:: CHAR: 26 (SYNC)
:: *****
↑ST30: SCOPE
BIS #MRESET,@TXCSR ; MASTER RESET
MOV #ISYMOD,@PARCSR ; SET THE MODE
BIS #MRESET,@TXCSR ; MASTER RESET

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

; SET MODE # OF BITS, PARITY SENSE & LOAD SYNC REG
MOV #ISYMOD!SIX!NOPAR!26,@PARCSR
BIS #SYNSCH,@RXCSR ; SET SYNC SEARCH

```


J04

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 50
 DZDUSA.M11 13-OCT-76 08:39 INITIALIZE THE COMMON TAGS

2350					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2351	010434	042777	020000	171264	BIC #CLK,@TXCSR ;POKE CLK DOWN
2352	010442	052777	020000	171256	BIS #CLK,@TXCSR ;POKE CLK UP
2353					;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2354	010450	042777	020000	171250	BIC #CLK,@TXCSR ;POKE CLK DOWN
2355	010456	052777	020000	171242	BIS #CLK,@TXCSR ;POKE CLK UP
2356	010464	052777	000400	171220	BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2357	010472	012767	000003	170424	MOV #3,COUNT ;# OF SYNC CHARS
2358	010500	012767	000254	170772	15: MOV #254,STMP1 ;CHAR TO BE SHIFTED
2359	010506	012767	000010	170406	MOV #8,SHIFT ;# OF SHIFTS
2360	010514	004767	006372		JSR PC,@POKE ;SHIFT IN THIS CHAR
2361	010520	105777	171166		TSTB @RXCSR ;RXDONE
2362	010524	100001			BPL .+4
2363	010526	104004			ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2364	010530	005367	170370		DEC COUNT ;# OF SYNC CHARS
2365	010534	001361			BNE 15
2366					
2367					::THIS TEST CHECKS THE STRIP SYNC FUNCTION
2368					::OF THE RECEIVER LOGIC
2369					::MODE:ISYMOD
2370					::LENGTH:SEVEN
2371					::NOTE: RXDONE SHOULD NEVER ASSERT
2372					::CHAR: 26 (SYNC)
2373					::*****
2374					::*****
2375	010536	000004			↑ST31: SCOPE
2376	010540	052777	000400	171160	BIS #MRESET,@TXCSR ;MASTER RESET
2377	010546	012777	000000	171146	MOV #ISYMOD,@PARCSR ;SET THE MODE
2378	010554	052777	000400	171144	BIS #MRESET,@TXCSR ;MASTER RESET
2379					
2380					;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2381	010562	012777	064001	171136	MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2382					
2383					;SET MODE # OF BITS PARITY SENSE & LOAD SYNC REG
2384	010570	012777	004026	171124	MOV #ISYMOD!SEVEN!NOPAR!26,@PARCSR
2385	010576	052777	000020	171106	BIS #SYNSCH,@RXCSR ;SET SYNC SEARCH
2386					;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
2387	010604	042777	020000	171114	BIC #CLK,@TXCSR ;POKE CLK DOWN
2388	010612	052777	020000	171106	BIS #CLK,@TXCSR ;POKE CLK UP
2389					;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2390	010620	042777	020000	171100	BIC #CLK,@TXCSR ;POKE CLK DOWN
2391	010626	052777	020000	171072	BIS #CLK,@TXCSR ;POKE CLK UP
2392	010634	052777	000400	171050	BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2393	010642	012767	000003	170254	MOV #3,COUNT ;# OF SYNC CHARS
2394	010650	012767	000454	170622	15: MOV #454,STMP1 ;CHAR TO BE SHIFTED
2395	010656	012767	000011	170236	MOV #9,SHIFT ;# OF SHIFTS
2396	010664	004767	006222		JSR PC,@POKE ;SHIFT IN THIS CHAR
2397	010670	105777	171016		TSTB @RXCSR ;RXDONE
2398	010674	100001			BPL .+4
2399	010676	104004			ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2400	010700	005367	170220		DEC COUNT ;# OF SYNC CHARS
2401	010704	001361			BNE 15
2402					
2403					::THIS TEST CHECKS THE STRIP SYNC FUNCTION
2404					::OF THE RECEIVER LOGIC
2405					::MODE:ISYMOD

K04

2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461

```
010706 000004
010710 052777 000400 171010
010716 012777 000000 170776
010724 052777 000400 170774
010732 012777 064001 170766
010740 012777 006026 170754
010746 052777 000020 170736
010754 042777 020000 170744
010762 052777 020000 170736
010770 042777 020000 170730
010776 052777 020000 170722
011004 052777 000400 170700
011012 012767 000003 170104
011020 012767 001054 170452
011026 012767 000012 170066
011034 004767 006052
011040 105777 170646
011044 100001
011046 104004
011050 005367 170050
011054 001361
```

```
:::LENGTH:EIGHT
:::NOTE: RXDONE SHOULD NEVER ASSERT
:::CHAR: 26 (SYNC)
::*****
†ST32: SCOPE
      BIS      #MRESET,@TXCSR ;MASTER RESET
      MOV      #ISYMOD,@PARCSR ;SET THE MODE
      BIS      #MRESET,@TXCSR ;MASTER RESET
;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
      MOV      #ISYMOD!EIGHT!NOPAR!26,@PARCSR
      BIS      #SYNSCH,@RXCSR ;SET SYNC SEARCH
      .POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
      MOV      #3,COUNT ;# OF SYNC CHARS
1$:   MOV      #1054,STMP1 ;CHAR TO BE SHIFTED
      MOV      #10,SHIFT ;# OF SHIFTS
      JSR      PC,RPOKE ;SHIFT IN THIS CHAR
      TSTB    @RXCSR ;RXDONE
      BPL     .+4
      ERROR   4 ;RXDONE SHOULD NOT BE ASSERTED
      DEC     COUNT ;# OF SYNC CHARS
      BNE    1$
:::THIS TEST CHECKS THE STRIP SYNC FUNCTION
:::OF THE RECEIVER LOGIC
:::MODE:SYNEXT
:::LENGTH:FIVE
:::NOTE: RXDONE SHOULD NEVER ASSERT
:::CHAR: 26 (SYNC)
::*****
†ST33: SCOPE
      BIS      #MRESET,@TXCSR ;MASTER RESET
      MOV      #SYNEXT,@PARCSR ;SET THE MODE
      BIS      #MRESET,@TXCSR ;MASTER RESET
;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
      MOV      #MTDATA!CLK!MINT!BREAK,@TXCSR
;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
      MOV      #SYNEXT!FIVE!NOPAR!26,@PARCSR
      BIS      #SYNSCH,@RXCSR ;SET SEARCH SYNC
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
      BIC      #CLK,@TXCSR ;POKE CLK DOWN
      BIS      #CLK,@TXCSR ;POKE CLK UP
      BIS      #STPSYN,@RXCSR ;SET STRIP SYNC
```

2462	011146	012767	000003	167750		MOV	#3,COUNT	;# OF SYNC CHARS
2463	011154	012767	000026	170316	15:	MOV	#26,STMP1	;CHAR TO BE SHIFTED
2464	011162	012767	000005	167732		MOV	#5,SHIFT	;# OF SHIFTS
2465	011170	004767	005716			JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2466	011174	105777	170512			TSTB	@RXCSR	;RXDONE
2467	011200	100001				BPL	.+4	
2468	011202	104004				ERROR	4	;RXDONE SHOULD NOT BE ASSERTED
2469	011204	005367	167714			DEC	COUNT	;# OF SYNC CHARS
2470	011210	001361				BNE	15	

```

;: THIS TEST CHECKS THE STRIP SYNC FUNCTION
;: OF THE RECEIVER LOGIC
;: MODE:SYNEXT
;: LENGTH:SIX
;: NOTE: RXDONE SHOULD NEVER ASSERT
;: CHAR: 26 (SYNC)

```

2480	011212	000004			†ST34:	SCOPE		
2481	011214	052777	000400	170504		BIS	#MRESET,@TXCSR	;MASTER RESET
2482	011222	012777	020000	170472		MOV	#SYNEXT,@PARCSR	;SET THE MODE
2483	011230	052777	000400	170470		BIS	#MRESET,@TXCSR	;MASTER RESET

```

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

```

2486	011236	012777	064001	170462				
2489	011244	012777	022026	170450		MOV	#SYNEXT!SIX!NOPAR!26,@PARCSR	;SET MODE # OF BITS,PARITY SENSE,&LOAD SYNC REG
2490	011252	052777	000020	170432		BIS	#SYNSCH,@RXCSR	;SET SEARCH SYNC

;POKE CLK TO GET LOGIC INTO SYNCHRONIZATION

2492	011260	042777	020000	170440		BIC	#CLK,@TXCSR	;POKE CLK DOWN
2493	011266	052777	020000	170432		BIS	#CLK,@TXCSR	;POKE CLK UP
2494	011274	052777	000400	170410		BIS	#STPSYN,@RXCSR	;SET STRIP SYNC
2495	011302	012767	000003	167614		MOV	#3,COUNT	;# OF SYNC CHARS
2496	011310	012767	000026	170162	15:	MOV	#26,STMP1	;CHAR TO BE SHIFTED
2497	011316	012767	000006	167576		MOV	#6,SHIFT	;# OF SHIFTS
2498	011324	004767	005562			JSR	PC,RPOKE	;SHIFT IN THIS CHAR
2499	011330	105777	170356			TSTB	@RXCSR	;RXDONE

```

BPL .+4
ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
DEC COUNT ;# OF SYNC CHARS
BNE 15

```

```

;: THIS TEST CHECKS THE STRIP SYNC FUNCTION
;: OF THE RECEIVER LOGIC
;: MODE:SYNEXT
;: LENGTH:SEVEN
;: NOTE: RXDONE SHOULD NEVER ASSERT
;: CHAR: 26 (SYNC)

```

2513	011346	000004			†ST35:	SCOPE		
2514	011350	052777	000400	170350		BIS	#MRESET,@TXCSR	;MASTER RESET
2515	011356	012777	020000	170336		MOV	#SYNEXT,@PARCSR	;SET THE MODE
2516	011364	052777	000400	170334		BIS	#MRESET,@TXCSR	;MASTER RESET

2517

M04

```

2518 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2519 011372 012777 064001 170326 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2520
2521 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
2522 011400 012777 024026 170314 MOV #SYNEXT!SEVEN!NOPAR!26,@PARCSR
2523 011406 052777 000020 170276 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2524 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2525 011414 042777 020000 170304 BIC #CLK,@TXCSR ;POKE CLK DOWN
2526 011422 052777 020000 170276 BIS #CLK,@TXCSR ;POKE CLK UP
2527 011430 052777 000400 170254 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2528 011436 012767 000003 167460 MOV #3,COUNT ;# OF SYNC CHARS
2529 011444 012767 000026 170026 1$: MOV #26,$TMP1 ;CHAR TO BE SHIFTED
2530 011452 012767 000007 167442 MOV #7,SHIFT ;# OF SHIFTS
2531 011460 004767 005426 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2532 011464 105777 170222 TSTB @RXCSR ;RXDONE
2533 011470 100001 BPL .+4
2534 011472 104004 ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2535 011474 005367 167424 DEC COUNT ;# OF SYNC CHARS
2536 011500 001361 BNE 1$
  
```

```

;; THIS TEST CHECKS THE STRIP SYNC FUNCTION
;; OF THE RECEIVER LOGIC
;; MODE:SYNEXT
;; LENGTH:EIGHT
;; NOTE: RXDONE SHOULD NEVER ASSERT
;; CHAR: 26 (SYNC)
  
```

```

2545
2546 011502 000004 †ST36: SCOPE
2547 011504 052777 000400 170214 BIS #MRESET,@TXCSR ;MASTER RESET
2548 011512 012777 020000 170202 MOV #SYNEXT,@PARCSR ;SET THE MODE
2549 011520 052777 000400 170200 BIS #MRESET,@TXCSR ;MASTER RESET
2550
2551 ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2552 011526 012777 064001 170172 MOV #MTDATA!CLK!MINT!BREAK,@TXCSR
2553
2554 ;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
2555 011534 012777 026026 170160 MOV #SYNEXT!EIGHT!NOPAR!26,@PARCSR
2556 011542 052777 000020 170142 BIS #SYNSCH,@RXCSR ;SET SEARCH SYNC
2557 ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2558 011550 042777 020000 170150 BIC #CLK,@TXCSR ;POKE CLK DOWN
2559 011556 052777 020000 170142 BIS #CLK,@TXCSR ;POKE CLK UP
2560 011564 052777 000400 170120 BIS #STPSYN,@RXCSR ;SET STRIP SYNC
2561 011572 012767 000003 167324 MOV #3,COUNT ;# OF SYNC CHARS
2562 011600 012767 000026 167672 1$: MOV #26,$TMP1 ;CHAR TO BE SHIFTED
2563 011606 012767 000010 167306 MOV #8,SHIFT ;# OF SHIFTS
2564 011614 004767 005272 JSR PC,RPOKE ;SHIFT IN THIS CHAR
2565 011620 105777 170066 TSTB @RXCSR ;RXDONE
2566 011624 100001 BPL .+4
2567 011626 104004 ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2568 011630 005367 167270 DEC COUNT ;# OF SYNC CHARS
2569 011634 001361 BNE 1$
  
```

```

;; THIS TEST CHECKS THE STRIP SYNC FUNCTION
;; OF THE RECEIVER LOGIC
;; MODE:SYNINT
  
```

2570
2571
2572
2573

```

2574
2575
2576
2577
2578
2579 011636 000004
2580 011640 052777 000400 170060
2581 011646 012777 030000 170046
2582 011654 052777 000400 170044
2583
2584
2585 011662 012777 064001 170036
2586
2587
2588 011670 012777 030026 170024
2589 011676 052777 000020 170006
2590
2591 011704 042777 020000 170014
2592 011712 052777 020000 170006
2593
2594 011720 042777 020000 170000
2595 011726 052777 020000 167772
2596 011734 052777 000400 167750
2597 011742 012767 000003 167154
2598 011750 012767 000026 167522
2599 011756 012767 000005 167136
2600 011764 004767 005122
2601 011770 105777 167716
2602 011774 100001
2603 011776 104004
2604 012000 005367 167120
2605 012004 001361
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615 012006 000004
2616 012010 052777 000400 167710
2617 012016 012777 030000 167676
2618 012024 052777 000400 167674
2619
2620
2621 012032 012777 064001 167666
2622
2623
2624 012040 012777 032026 167654
2625 012046 052777 000020 167636
2626
2627 012054 042777 020000 167644
2628 012062 052777 020000 167636
2629

```

```

::LENGTH:FIVE
::NOTE: RXDONE SHOULD NEVER ASSERT
::CHAR: 26 (SYNC)
::*****
↑ST37: SCOPE
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #SYNINT,@PARCSR ;SET THE MODE
BIS #MRESET,@TXCSR ;MASTER RESET

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

;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
MOV #SYNINT!FIVE!NOPAR!26,@PARCSR
BIS #SYNSCH,@TXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
BIS #STPSYN,@TXCSR ;SET STRIP SYNC
MOV #3,COUNT ;# OF SYNC CHARS
1$: MOV #26,STMP1 ;CHAR TO BE SHIFTED
MOV #5,SHIFT ;# OF SHIFTS
JSR PC,RPOKE ;SHIFT IN THIS CHAR
TSTB @RXCSR ;RXDONE
BPL .+4
ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
DEC COUNT ;# OF SYNC CHARS
BNE 1$

::THIS TEST CHECKS THE STRIP SYNC FUNCTION
::OF THE RECEIVER LOGIC
::MODE:SYNINT
::LENGTH:SIX
::NOTE: RXDONE SHOULD NEVER ASSERT
::CHAR: 26 (SYNC)
::*****
↑ST40: SCOPE
BIS #MRESET,@TXCSR ;MASTER RESET
MOV #SYNINT,@PARCSR ;SET THE MODE
BIS #MRESET,@TXCSR ;MASTER RESET

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

;SET MODE # OF BITS,PARITY SENSE &LOAD SYNC REG
MOV #SYNINT!SIX!NOPAR!26,@PARCSR
BIS #SYNSCH,@TXCSR ;SET SYNC SEARCH
;POKE CLK TO GET RECEIVER INTO SYNCROIZATION....
BIC #CLK,@TXCSR ;POKE CLK DOWN
BIS #CLK,@TXCSR ;POKE CLK UP
;POKE CLK TO GET LOGIC INTO SYNCRONIZATION

```

```

2630 012070 042777 020000 167630      BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2631 012076 052777 020000 167622      BIS    #CLK,@TXCSR      ;POKE CLK UP
2632 012104 052777 000400 167600      BIS    #STPSYN,@RXCSR   ;SET STRIP SYNC
2633 012112 012767 000003 167004      MOV    #3,COUNT         ;# OF SYNC CHARS
2634 012120 012767 000026 167352 1$:  MOV    #26,$TMP1        ;CHAR TO BE SHIFTED
2635 012126 012767 000006 166766      MOV    #6,SHIFT         ;# OF SHIFTS
2636 012134 004767 004752          JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
2637 012140 105777 167546          TSTB  @RXCSR ;RXDONE
2638 012144 100001          BPL    .+4
2639 012146 104004          ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2640 012150 005367 166750          DEC   COUNT ;# OF SYNC CHARS
2641 012154 001361          BNE   1$
2642
2643          ;: THIS TEST CHECKS THE STRIP SYNC FUNCTION
2644          ;: OF THE RECEIVER LOGIC
2645          ;: MODE: SYNINT
2646          ;: LENGTH: SEVEN
2647          ;: NOTE: RXDONE SHOULD NEVER ASSERT
2648          ;: CHAR: 26 (SYNC)
2649
2650          ;: *****
2651 012156 000004          †ST41: SCOPE
2652 012160 052777 000400 167540          BIS    #MRESET,@TXCSR ;MASTER RESET
2653 012166 012777 030000 167526          MOV    #SYNINT,@PARCSR ;SET THE MODE
2654 012174 052777 000400 167524          BIS    #MRESET,@TXCSR ;MASTER RESET
2655
2656          ;SET MAINT DATA,CLK,BREAK,&MAINTENANCE MODE
2657 012202 012777 064001 167516          MOV    #MTDATA!CLK!MINT!BREAK,@TXCSR
2658
2659          ;SET MODE ,# OF BITS,PARITY SENSE &LOAD SYNC REG
2660 012210 012777 034026 167504          MOV    #SYNINT!SEVEN!NOPAR!26,@PARCSR
2661 012216 052777 000020 167466          BIS    #SYNSCH,@RXCSR ;SET SYNC SEARCH
2662          ;POKE CLK TO GET RECEIVER INTO SYNCRIZATION....
2663 012224 042777 020000 167474          BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2664 012232 052777 020000 167466          BIS    #CLK,@TXCSR      ;POKE CLK UP
2665          ;POKE CLK TO GET LOGIC INTO SYNCRONIZATION
2666 012240 042777 020000 167460          BIC    #CLK,@TXCSR      ;POKE CLK DOWN
2667 012246 052777 020000 167452          BIS    #CLK,@TXCSR      ;POKE CLK UP
2668 012254 052777 000400 167430          BIS    #STPSYN,@RXCSR   ;SET STRIP SYNC
2669 012262 012767 000003 166634          MOV    #3,COUNT         ;# OF SYNC CHARS
2670 012270 012767 000026 167202 1$:  MOV    #26,$TMP1        ;CHAR TO BE SHIFTED
2671 012276 012767 000007 166616          MOV    #7,SHIFT         ;# OF SHIFTS
2672 012304 004767 004602          JSR    PC,RPOKE         ;SHIFT IN THIS CHAR
2673 012310 105777 167376          TSTB  @RXCSR ;RXDONE
2674 012314 100001          BPL    .+4
2675 012316 104004          ERROR 4 ;RXDONE SHOULD NOT BE ASSERTED
2676 012320 005367 166600          DEC   COUNT ;# OF SYNC CHARS
2677 012324 001361          BNE   1$
2678
2679
2680          ;END OF PASS
2681          ;TYPE NAME OF TEST
2682          ;UPDATE PASS COUNT
2683          ;CHECK FOR EXIT TO ACT-11
2684          ;RESTART TEST
2685

```

```

2686 012326 000004 .EOP: SCOPE
2687 012330 004767 000344 JSR PC,CKSWR
2688 012334 104401 ;TYPE NAME OF TEST
2689 012336 015470 MEPASS
2690 012340 104413 012572 CONVRT ,OUTCRY
2691 012344 104401 015307 TYPE ,DEVICE
2692 012350 105767 166576 TSTB MULTD ;ARE YOU RUNNING MULTIPLE DEVICES ?
2693 012354 001511 BEQ CCC ;NO JUMP AROUND
2694 012356 005767 166604 TST ACTREG ;ARE ANY DEVICES ACTIVE ?
2695 012362 001007 BNE RUNIT ;YES
2696 012364 104401 015321 TYPE MCOV ;NO
2697 012370 016700 166572 MOV ACTREG,R0 ;DISPLAY ACTREG
2698 012374 000000 HALT ;SELECT SOMETHING TO RUN @ ACTREG:
2699 ;SELECT SWITCHES & HIT CONTINUE (PUT SW00 =1)
2700 012376 000167 167550 JMP .START ;START OVER AGAIN.....YOU Deselected EVERYTHING
2701 012402 062767 000010 166544 RUNIT: ADD #10,BASEADD ;NEXT BLOCK (ADDRESSES)
2702 012410 062767 000010 166544 ZERO: ADD #10,BASEIV ;NEXT BLOCK (VECTORS)
2703 012416 000241 CLC
2704 012420 006167 166544 ROL ROTADD ;UP DATE ROTATING POINTER
2705 012424 103410 BCS 2$ ;IS IT THE LAST DEVICE
2706 ;TO BE TESTED IN THIS PASS ?
2707 012426 036767 166536 166532 BIT ROTADD,ACTREG ;TEST THIS DEVICE FOR ACTIVE STATUS
2708 012434 001762 BEQ RUNIT ;IF NOT ACTIVE, TRY NEXT ADDRESS
2709 012436 004767 000034 JSR PC,REPLAY ;CALCULATE NEW PARAMETERS
2710 012442 000167 000210 JMP RESTRAT ;YES IT WAS ACTIVE, TEST THIS DEVICE
2711 012446 012767 000001 166514 2$: MOV #1,ROTADD ;OK!, NOW SET UP ROTATING
2712 ; POINTER FOR NEXT MULTIPLE PASS
2713 012454 016767 166476 166472 MOV KEEPADD,BASEADD ;RESTORE BASE ADDRESS
2714 012462 016767 166476 166472 MOV KEEPIV,BASEIV ;RESTORE BASE INTERRUPT VECTORS
2715 012470 004767 000002 JSR PC,REPLAY ;CALC NF 4 PARAMETERS
2716 012474 000441 BR CCC ;JUMP AROUND REPLAY
2717 012476 016767 166452 004404 REPLAY: MOV BASEADD,DUBASE ;SET UP FOR NEW ADDRESSES
2718 012504 004767 004246 JSR PC,DUADR ;CREATE NEW ADDRESSES
2719 012510 016767 166446 167220 MOV BASEIV,DURIV ;CREATE DURIV
2720 012516 062767 000002 166436 ADD #2,BASEIV
2721 012524 016767 166432 167206 MOV BASEIV,DURIS ;CREATE DURIS
2722 012532 062767 000002 166422 ADD #2,BASEIV
2723 012540 016767 166416 167174 MOV BASEIV,DUTIV ;CREATE DUTIV
2724 012546 062767 000002 166406 ADD #2,BASEIV
2725 012554 016767 166402 167162 MOV BASEIV,DUTIS ;CREATE DUTIS
2726 012562 016767 167150 166372 MOV DURIV,BASEIV ;RESTORE
2727 012570 000207 RTS PC
2728
2729 012572 000001 OUTCRY: 1
2730 012574 006 002 .BYTE 6,2
2731 012576 001712 RXCSR
2732
2733 012600 CCC:
2734 012600 005067 166576 CLR $STNM ;CLEAR TEST NUMBER
2735 012604 005067 166606 CLR $ERRPC ;CLEAR LAST ERROR PC
2736 012610 005067 166567 CLR $ERFLG ;CLEAR ERROR FLAG
2737 012614 005267 166272 INC PASCNT ;UPDATE PASS COUNT
2738 012620 016767 166266 166254 MOV PASCNT,LIGHTS ;DISPLAY PASS COUNT
2739 012626 016767 166260 166700 MOV PASCNT,$PASS ;PASS COUNT TO APT
2740 012634 013701 000042 MOV #42,R1 ;CHECK FOR ACT-11 OR DDP
2741 012640 001406 BEQ RESTRAT ;IF NO CONTINUE TESTING

```

```

2742 012642 000005          RESET
2743 012644 000005          RESET
2744 012646 004711          SENDAD: JSR      PC,(R1)
2745 012650 000240          NOP
2746 012652 000240          NOP
2747 012654 000240          NOP
2748 012656 106427 000340          RESTRT: MTPS   #340      ;PREVENT INTERRUPTS (PRIO: 7)
2749 012662 004767 000012          JSR      PC,CKSWR
2750 012666 012767 003364 166512          MOV     #TST1+2,SLPADR ;SET LAST ADDRESS POINTER
2751 012674 000167 170462          JMP     TST1
2752
2753          ;CHECK SWITCH REGISTER ROUTINE.
2754          ;CHECKS TO ALLOW FOR (↑G) TO ALLOW
2755          ;THE CHANGING OF LOCATION 176
2756
2757 012700 005737 000042          CKSWR: TST     #42
2758 012704 001040          BNE     OUT
2759 012706 022767 000176 166524          CMP     #SWREG,SWR ;SOFTWARE SWR PRESENT?
2760 012714 001034          BNE     OUT ;NO--LEAVE
2761 012716 105777 166522          TSTB   #STKS ;CHECK TTY READY
2762 012722 100031          BPL     OUT ;NO--LEAVE
2763 012724 017767 166516 000422          MOV     #STKB,MSG ;GET CHARACTER
2764 012732 042767 177600 000414          BIC     #177600,MSG ;STRIP JUNK
2765 012740 122767 000007 000406          CMPB   #7,MSG ;IS IT (↑G) ?
2766 012746 001017          BNE     OUT ;NO
2767 012750 104401 016075          TYPE   MCNTG
2768 012754 005137 013014          CNTLU: COM    #RDSW
2769 012760 104401 016105          TYPE   ,MMSWR
2770 012764 104413          CONVRT
2771 012766 013016          SWREGL
2772 012770 104406 016116          INSTR,MMNEW
2773 012774 104410          PARAM
2774 012776 000000          0
2775 013000 177777          177777
2776 013002 000176          SWREG
2777 013004 000          001
2778 013006 005037 013014          .BYTE 0,1
2779 013012 000207          OUT:   CLR     #RDSW
2780 013014 000000          RTS    PC
2781 013016 000001          RDSW:  .WORD 0
2782 013020 006          002
2783 013022 000176          SWREGL: .BYTE 6,2
2784
2785 013024 000005          SWREG
2786
2787          5
2788          ;CHECK FOR FREEZE ON CURRENT DATA
2789 013026 004767 177646          .SCOPI: JSR     PC,CKSWR
2790 013032 032777 001000 166400          BIT     #SW09,#SWR
2791 013040 001402          BEQ    IS
2792 013042 016716 166042          MOV     LOCK,(SP)
2793 013046 000002          IS:    RTI
2794          .SBTTL TYPE ROUTINE
2795
2796          ;*****
2797          ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.

```


DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 58
 DZDUSA.M11 13-OCT-76 08:39 TYPE ROUTINE

```

2798 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2799 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2800 ;*NOTE2: $FILLC CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2801 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2802 ;*
2803 ;*CALL:
2804 ;*1) USING A TRAP INSTRUCTION
2805 ;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2806 ;*OR
2807 ;* TYPE
2808 ;* MESADR
2809 ;*
2810
2811 013050 105767 166403 $TYPE: TSTB $TFPLG ;; IS THERE A TERMINAL?
2812 013054 100002 BPL 1$ ;; BR IF YES
2813 013056 000000 HALT ;; HALT HERE IF NO TERMINAL
2814 013060 000430 BR 3$ ;; LEAVE
2815 013062 010046 1$: MOV RO, -(SP) ;; SAVE RO
2816 013064 017600 000002 MOV 22(SP), RO ;; GET ADDRESS OF ASCIZ STRING
2817 013070 122767 000001 166 50 CMPB #APTENV, $ENV ;; RUNNING IN APT MODE
2818 013076 001011 BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
2819 013100 132767 000100 166441 BITB #APTPOOL, $ENVM ;; SPOOL MESSAGE TO APT
2820 013106 001405 BEQ 62$ ;; NO GO CHECK FOR CONSOLE
2821 013110 010067 000004 MOV RO, 61$ ;; SETUP MESSAGE ADDRESS FOR APT
2822 013114 004767 000006 JSR PC, $ATY3 ;; SPOOL MESSAGE TO APT
2823 013120 000000 61$: .WORD 0 ;; MESSAGE ADDRESS
2824 013122 132767 000040 166417 62$: BITB #APTCSUP, $ENVM ;; APT CONSOLE SUPPRESSED
2825 013130 001003 BNE 60$ ;; YES, SKIP TYPE OUT
2826 013132 112046 2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2827 013134 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
2828 013136 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
2829 013140 012600 60$: MOV (SP)+, RO ;; RESTORE RO
2830 013142 062716 000002 3$: ADD #2, (SP) ;; ADJUST RETURN PC
2831 013146 000002 RTI ;; RETURN
2832 013150 122716 000011 4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
2833 013154 001430 BEQ 8$
2834 013156 122716 000200 CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
2835 013162 001006 BNE 5$
2836 013164 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
2837 013166 104401 TYPE ;; TYPE A CR AND LF
2838 013170 001523 $CRLF
2839 013172 105067 000130 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
2840 013176 000755 BR 2$ ;; GET NEXT CHARACTER
2841 013200 004767 000056 5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
2842 013204 126726 166246 6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
2843 013210 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
2844 013212 016746 166236 MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
2845 ;; AND THE NULL CHAR.
2846 013216 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
2847 013222 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
2848 013224 004767 000032 JSR PC, $TYPEC ;; GO TYPE A NULL
2849 013230 105367 000072 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
2850 013234 000770 BR 7$ ;; LOOP
2851
2852 ;HORIZONTAL TAB PROCESSOR
2853

```

F05

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 59
 DZDUSA.M11 13-OCT-76 08:39 TYPE ROUTINE

```

2854 013236 112716 000040      8$:  MOVB  #' (SP)           ; REPLACE TAB WITH SPACE
2855 013242 004767 000014      9$:  JSR  PC,$TYPEPC        ; TYPE A SPACE
2856 013246 132767 000007 000052 BITB  #7,$SCHARCNT      ; BRANCH IF NOT AT
2857 013254 001372                BNE  9$                ; TAB STOP
2858 013256 005726                TST  (SP)+             ; POP SPACE OFF STACK
2859 013260 000724                BR   2$                ; GET NEXT CHARACTER
2860 013262 105777 166162      $TYPEPC: TSTB  2$TPS      ; WAIT UNTIL PRINTER IS READY
2861 013266 100375                BPL  $TYPEPC
2862 013270 116677 000002 166154 MOVB  2(SP),2$TPB      ; LOAD CHAR TO BE TYPED INTO DATA REG.
2863 013276 122766 000015 000002 CMPB  #CR,2(SP)       ; IS CHARACTER A CARRIAGE RETURN?
2864 013304 001003                BNE  1$                ; BRANCH IF NO
2865 013306 105067 000014                CLRB  $SCHARCNT      ; YES--CLEAR CHARACTER COUNT
2866 013312 000406                BR   $TYPEPC         ; EXIT
2867 013314 122766 000012 000002 1$:  CMPB  #LF,2(SP)      ; IS CHARACTER A LINE FEED?
2868 013322 001402                BEQ  $TYPEPC         ; BRANCH IF YES
2869 013324 105227                INCB  (PC)+           ; COUNT THE CHARACTER
2870 013326 000000      $SCHARCNT: WORD  0      ; CHARACTER COUNT STORAGE
2871 013330 000207      $TYPEPC:  RTS   PC
  
```

;ASCII STRING INPUT ROUTINE

```

2872
2873
2874
2875
2876 013332 017667 000000 000014 .INSTR: MOV  2(SP),MSG      ; PICK UP MESSAGE
2877 013340 062716 000002          ADD  #2,(SP)          ; JUMP AROUND MESSAGE FOR RTI
2878 013344 105767 166176          TSTB  $ENV           ; APT CONTROL
2879 013350 001036                BNE  INSTR2          ; YES NO TYPE
2880 013352 104401                .INST1: TYPE
2881 013354 000000                .MSG:  0
2882 013356 012704 016130          MOV  #INBUF,R4       ; GET STARTING LOC OF INBUF
2883 013362 012703 000007          MOV  #7,R3           ; MAX # OF CHARS
2884 013366 105777 166052      1$:  TSTB  2$TKS ; TTY FLAG
2885 013372 100375                BPL  1$
2886 013374 117714 166046          MOVB  2$TKB,(R4)     ; TAKE CHAR
2887 013400 142714 000200          BICB  #200,(R4)     ; STRIP
2888 013404 121427 000025          CMPB  (R4),#25      ; IS IT (<↑G)
2889 013410 001760                BEQ  .INST1
2890 013412 122427 000015          CMPB  (R4)+,#15     ; CHECK FOR CR
2891 013416 001413                BEQ  INSTR2
2892 013420 105777 166024      2$:  TSTB  2$TPS ; TEST FLAG
2893 013424 100375                BPL  2$
2894 013426 117777 166014 166016 MOVB  2$TKB,2$TPB    ; ECHO CHARACTER
2895 013434 005303                DEC  R3              ; DID YOU TYPE TOO MANY CHARS ?
2896 013436 001353                BNE  1$
2897 013440 104401                .INSTE: TYPE
2898 013442 015415                MQM  ;?
2899 013444 000742                BR   .INST1 ; RETRY
2900 013446 000002      INSTR2: RTI
  
```

;CONVERT ASCII STRING TO OCTAL

```

2901
2902
2903
2904 013450 011605                .PARAM: MOV  (SP),R5 ; PUT CONTENTS OF SP INTO R5
2905 013452 012567 000162          MOV  (R5)+,LOLIM    ; PUT LOW LIMIT INTO LOLIM
2906 013456 012567 000160          MOV  (R5)+,HILIM    ; PUT HIGH LIMIT INTO HILIM
2907 013462 012567 000156          MOV  (R5)+,DEVADR    ; PUT STORE LOC INTO DEVADR
2908 013466 112567 000154          MOVB  (R5)+,LOBITS   ; PUT MASK INTO LOBITS
2909 013472 112567 000151          MOVB  (R5)+,ADRCNT   ; PUT COUNT INTO ADRCNT
  
```

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 60
 DZDUSA.M11 13-OCT-76 08:39 TYPE ROUTINE

2910	013476	010516			MOV	R5,(SP)	;RESTORE RETURN ADDR ON STACK FOR RTI
2911	013500	005005			PARAM1: CLR	R5	
2912	013502	012704	016130		MOV	#INBUF,R4	
2913	013506	122714	000015		CMPB	#15,(R4)	;CR ?
2914	013512	001420			BEQ	PARERR	;YOU TYPED CR TOO SOON !
2915	013514	121427	000060		1\$: CMPB	(R4),#60	;LOW LIMIT ASCII 0
2916	013520	002415			BLT	PARERR	
2917	013522	121427	000067		CMPB	(R4),#67	;HIGH LIMIT ASCII 7
2918	013526	003012			BGT	PARERR	
2919	013530	142714	00006C		BICB	#60,(R4)	;CONVERT TO OCTAL
2920	013534	152405			BISB	(R4)+,R5	;STORE AWAY ITS AN OK CHAR
2921	013536	122714	000015		CMPB	#15,(R4)	;CR ?
2922	013542	001414			BEQ	LIMITS	;NOW CHECK FOR HIGH & LOW LIMIT CONDS
2923	013544	006305			ASL	R5	;ALLOCATE ROOM FOR NEXT CHAR
2924	013546	006305			ASL	R5	
2925	013550	006305			ASL	R5	
2926	013552	000760			BR	1\$	
2927	013554	122714	000015		PARERR: CMPB	#15,(R4)	;CR?
2928	013560	001003			BNE	120\$	
2929	013562	005737	013014		TST	2*RD SW	;CK SWR USED
2930	013566	001023			BNE	PARTI	
2931	013570	104407			120\$: INSTER	:RETRY	
2932	013572	000742			BR	PARAM1	
2933							
2934							;TEST TO SEE IF NUMBER IS WITHIN LIMITS
2935							
2936	013574	020567	000042		LIMITS: CMP	R5,HILIM	
2937	013600	101365			BHI	PARERR	;THE # IS TOO HIGH
2938	013602	020567	000032		CMP	R5,LOLIM	
2939	013606	103762			BLO	PARERR	;THE # IS TOO LOW
2940	013610	136705	000032		BITB	LOBITS,R5	;TEST BY MASKING THE #
2941	013614	001357			BNE	PARERR	
2942							
2943							;STORE NUMBER AT SPECIFIED ADDRESS
2944							
2945	013616	016704	000022		1\$: MOV	DEVADR,R4	;GET STARTING ADDR OF
2946	013622	010524			MOV	R5,(R4)+	;STORE AT THIS ADDR
2947	013624	062705	000002		ADD	#2,R5	
2948	013630	105367	000013		DECB	ADRCNT	;HOW MANY TIMES + 2 ?
2949	013634	001372			BNE	1\$	
2950	013636	000002			PARTI: RTI		
2951	013640	000000			LOLIM: 0		
2952	013642	000000			HILIM: 0		
2953	013644	000000			DEVADR: 0		
2954	013646	000000			LOBITS: 0		
2955		013647			ADRCNT=LOBITS+1		
2956							
2957							;SAVE PC OF TEST THAT FAILED AND R0-R5
2958							
2959	013650	016667	000004	165250	.SAVOS: MOV	4(SP),SAVPC	
2960							
2961							;SAVE R0-R5
2962							
2963	013656	010567	165612		SVOS: MOV	R5,\$REG5	
2964	013662	010467	165604		MOV	R4,\$REG4	
2965	013666	010367	165576		MOV	R3,\$REG3	

```

2966 013672 010267 165570      MOV     R2,$REG2
2967 013676 010167 165562      MOV     R1,$REG1
2968 013702 010067 165554      MOV     R0,$REG0
2969 013706 000002                RTI
2970
2971                                ;RESTORE R0-R5
2972
2973 013710 016700 165546      .RES05: MOV     $REG0,R0
2974 013714 016701 165544      MOV     $REG1,R1
2975 013720 016702 165542      MOV     $REG2,R2
2976 013724 016703 165540      MOV     $REG3,R3
2977 013730 016704 165536      MOV     $REG4,R4
2978 013734 016705 165534      MOV     $REG5,R5
2979 013740 000002                RTI
2980
2981                                ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
2982
2983 013742 104401      .CONVR: TYPE
2984 013744 015421      MCRLF      :CR LF
2985 013746 017601 000000      MOV     2(SP),R1      :PICK UP DATA POINTER
2986 013752 062716 000002      ADD     #2(SP)      :SET UP SP FOR RTI
2987 013756 012167 000130      MOV     (R1)+,WRDCNT  :PICK UP # OF WORDS FROM TABLE
2988 013762 112167 000126      1$:     MOV     (R1)+,CHRCNT  :PICK UP # OF CHARS FROM TABLE
2989 013766 112167 000123      MOV     (R1)+,SPACNT  :PICK UP # OF SPACES FROM TABLE
2990 013772 013167 000120      MOV     2(R1)+,BINWRD  :PICK UP ADDRESS OF MSG
2991                                :FROM TABLE
2992 013776 016704 000114      2$:     MOV     BINWRD,R4      :SAVE
2993 014002 116705 000106      MOV     CHRCNT,R5      :SAVE
2994 014006 012700 016172      MOV     #TEMP,R0      :STARTING ADDRESS OF TEMP BLOCK
2995 014012 010403      3$:     MOV     R4,R3      :SAVE
2996 014014 042703 17777C      BIC     #177770,R3      :CLR OUT UPPER BITS .. SAVE CHAR
2997 014020 062703 000260      ADD     #260,R3      :CONVERT TO ASCII
2998 014024 110320      MOV     R3,(R0)+      :STORE AWAY
2999 014026 006204      ASR     R4      :SHIFT FOR NEXT #
3000 014030 006204      ASR     R4      :DITTO
3001 014032 006204      ASR     R4      :DITTO
3002 014034 005305      DEC     R5      :DEC CHAR COUNT
3003 014036 001365      BNE     3$      :DO IT AGAIN ?
3004 014040 012703 016234      MOV     #MDATA,R3      :STARTING ADDRESS OF MDATA BLOCK
3005 014044 114023      4$:     MOV     -(R0),(R3)+      :REVERSE THE ORDER OF NUMBERS
3006 014046 105367 000042      DECB   CHRCNT      :DEC CHAR COUNT
3007 014052 001374      BNE     4$      :DO IT AGAIN ?
3008 014054 105767 000035      TSTB   SPACNT      :HOW MANY SPACES ?
3009 014060 001405      BEQ     6$      :TYPE # IF BR =0
3010 014062 112723 000240      5$:     MOV     #240,(R3)+      :"SPACE" IN ASCII
3011 014066 105367 000023      DECB   SPACNT      :DEC # OF SPACE COUNT
3012 014072 001373      BNE     5$      :DO IT AGAIN ?
3013 014074 105013      6$:     CLR     (R3)      :INSERT "0" FOR TTY OUTPUT ROUTINE
3014 014076 104401      TYPE
3015 014100 016234      MDATA
3016 014102 005367 000004      :THIS MESSAGE
3017 014106 001325      WRDCNT      :HOW MANY #'S ?
3018 014110 000002      1$:     BNE     1$      :DO THIS ROUTINE AGAIN IF NOT EQUAL TO 0
3019 014112 000000      RTI      :RETURN TO PROGRAM
3020 014114 000000
3021 014115
WRDCNT: 0
CHRCNT: 0
SPACNT=CHRCNT+1

```

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 62
 DZDUSA.M11 13-OCT-76 08:39 TYPE ROUTINE

```

3022 014116 000000          BINWRD: 0
3023
3024          ;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
3025          ;BUFFER TO THE CHARACTERS "N" AND "Y"
3026          ;IF THE CHARACTER IS "N" CLEAR THE FLAG
3027          ;IF THE CHARACTER IS "Y" SET THE FLAG
3028
3029 014120 017605 000000    .SETFLG:MOV    2(SP),R5
3030 014124 122767 000116 001776  CMPB    #'N,INBUF    ;IS IT "N" ?
3031 014132 001002          BNE     1$
3032 014134 105015          CLRB    (R5)    ;000
3033 014136 000406          BR      2$
3034 014140 122767 000131 001762 1$:  CMPB    #'Y,INBUF    ;IS IT "Y" ?
3035 014146 001005          BNE     3$
3036 014150 112715 177777    MOVB    #-1,(R5)    ;377
3037 014154 062716 000002    2$:  ADD     #2,(SP)
3038 014160 000002          RTI
3039 014162 104407          3$:  INSTER ;RETRY
3040 014164 000755          BR      .SETFLG
3041          .SBTTL  ERROR HANDLER ROUTINE
3042
3043          ;*****
3044          ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3045          ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3046          ;AND GO TO SAVIT ON ERROR
3047          ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3048          ;SW15=1    HALT ON ERROR
3049          ;SW13=1    INHIBIT ERROR TYPEOUTS
3050          ;SW10=1    BELL ON ERROR
3051          ;SW09=1    LOOP ON ERROR
3052          ;CALL
3053          ;*      ERROR  N          ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3054
3055 014166          $ERROR:
3056 014166 105267 165211    7$:  INCB    $ERFLG          ;; SET THE ERROR FLAG
3057 014172 001775          BEQ     7$              ;; DON'T LET THE FLAG GO TO ZERO
3058 014174 016777 165202 165240  MOV     $STNM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
3059 014202 032777 002000 165230  BIT     #BIT10,$SWR    ;; BELL ON ERROR?
3060 014210 001402          BEQ     1$              ;; NO - SKIP
3061 014212 104401 001516          TYPE   $BELL          ;; RING BELL
3062 014216 005267 165170    1$:  INC     $ERTTL        ;; COUNT THE NUMBER OF ERRORS
3063 014222 011667 165170          MOV     (SP),$ERRPC    ;; GET ADDRESS OF ERROR INSTRUCTION
3064 014226 162767 000002 165162  SUB     #2,$ERRPC
3065 014234 117767 165156 165152  MOVB    2,$ERRPC,$ITEMB ;; STRIP AND SAVE THE ERROR ITEM CODE
3066 014242 032777 020000 165170  BIT     #BIT13,$SWR    ;; SKIP TYPEOUT IF SET
3067 014250 001004          BNE     20$           ;; SKIP TYPEOUTS
3068 014252 004767 000072          JSR    PC,SAVIT        ;; GO TO USER ERROR ROUTINE
3069 014256 104401 001523          TYPE   ,SCLF
3070 014262
3071 014262 122767 000001 165256  20$:  CMPB    #APTENV,$ENV    ;; RUNNING IN APT MODE
3072 014270 001007          BNE     2$
3073 014272 116767 165116 000004  MOVB    $ITEMB,21$    ;; NO SKIP APT ERROR REPORT
3074 014300 004767 000016'  JSR    PC,$ATY4        ;; SET ITEM NUMBER AS ERROR NUMBER
3075 014304 000          21$:  .BYTE  0              ;; REPORT FATAL ERROR TO APT
3076 014305 000          .BYTE  0
3077 014306 000777          22$:  BR      22$          ;; APT ERROR LOOP
    
```

```

3078 014310 005777 165124 2$: TST 2SWR ;: HALT ON ERROR
3079 014314 100001 ;: BPL 3$ ;: SKIP IF CONTINUE
3080 014316 000000 ;: HALT ;: HALT ON ERROR!
3081 014320 032777 001000 165112 3$: BIT #BIT09,2SWR ;: LOOP ON ERROR SWITCH SET?
3082 014326 001402 ;: BEQ 4$ ;: BR IF NO
3083 014330 016716 165054 ;: MOV $LPERR,(SP) ;: FUDGE RETURN FOR LOOPING
3084 014334 005767 165154 4$: TST $ESCAPE ;: CHECK FOR AN ESCAPE ADDRESS
3085 014340 001402 ;: BEQ 5$ ;: BR IF NONE
3086 014342 016716 165146 ;: MOV $ESCAPE,(SP) ;: FUDGE RETURN ADDRESS FOR ESCAPE
3087 014346 ;: 5$: ;:
3088 014346 000002 ;: RTI ;: ;:RETURN
3089 014350 010067 164554 SAVIT: MOV R0,HLD0
3090 014354 010167 164552 MOV R1,HLD1
3091 014360 010267 164550 MOV R2,HLD2
3092 014364 010367 164546 MOV R3,HLD3
3093 014370 010467 164544 MOV R4,HLD4
3094 014374 010567 164542 MOV R5,HLD5
3095 014400 016767 164776 164536 MOV $STNM,HLD6
  
```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

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

```

3104 014406 ;: 3104: SERRTYP:
3105 014406 104401 001523 ;: TYPE $SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
3106 014412 010046 ;: MOV R0,-(SP) ;: SAVE R0
3107 014414 005000 ;: CLR R0 ;: PICKUP THE ITEM INDEX
3108 014416 153700 001414 ;: BISB 2*$ITEMB,R0
3109 014422 001004 ;: BNE 1$ ;: IF ITEM NUMBER IS ZERO, JUST
3110 ;: ;: TYPE THE PC OF THE ERROR
3111 014424 016746 164766 ;: MOV $ERR-C,-(SP) ;: SAVE $ERRPC FOR TYPEOUT
3112 ;: ;: ERROR ADDRESS
3113 014430 104402 ;: TYPOC ;: GO TYPE--OCTAL ASCII(ALL DIGITS)
3114 014432 000426 ;: BR 6$ ;: GET OUT
3115 014434 005300 1$: DEC R0 ;: ADJUST THE INDEX SO THAT IT WILL
3116 014436 006300 ;: ASL R0 ;: WORK FOR THE ERROR TABLE
3117 014440 006300 ;: ASL R0
3118 014442 006300 ;: ASL R0
3119 014444 062700 001652 ;: ADD #SERRTB,R0 ;: FORM TABLE POINTER
3120 014450 012067 000004 ;: MOV (R0)+,2$ ;: PICKUP "ERROR MESSAGE" POINTER
3121 014454 001404 ;: BEQ 3$ ;: SKIP TYPEOUT IF NO POINTER
3122 014456 104401 ;: TYPE ;: TYPE THE "ERROR MESSAGE"
3123 014460 000000 2$: .WORD 0 ;: "ERROR MESSAGE" POINTER GOES HERE
3124 014462 104401 001523 ;: TYPE $SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
3125 014466 012067 000004 3$: MOV (R0)+,4$ ;: PICKUP "DATA HEADER" POINTER
3126 014472 001404 ;: BEQ 5$ ;: SKIP TYPEOUT IF 0
3127 014474 104401 ;: TYPE ;: TYPE THE "DATA HEADER"
3128 014476 000000 4$: .WORD 0 ;: "DATA HEADER" POINTER GOES HERE
3129 014500 104401 001523 ;: TYPE $SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
3130 014504 011000 5$: MOV (R0),R0 ;: PICKUP "DATA TABLE" POINTER
3131 014506 001004 ;: BNE 7$ ;: GO TYPE THE DATA
3132 014510 012600 6$: MOV (SP)+,R0 ;: RESTORE R0
3133 014512 104401 001523 ;: TYPE $SCLF ;: "CARRIAGE RETURN" & "LINE FEED"
  
```

K05

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 64
 DZDUSA.M11 13-OCT-76 08:39 ERROR MESSAGE TYPEOUT ROUTINE

3134	014516	000207			RTS	PC	;;RETURN
3135	014520				7\$:		
3136	014520	013046			MOV	2(R0)+, -(SP)	;;SAVE 2(R0)+ FOR TYPEOUT
3137	014522	104402			TYPOC		;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3138	014524	005710			TST	(R0)	;;IS THERE ANOTHER NUMBER?
3139	014526	001770			BEQ	6\$;;BR IF NO
3140	014530	104401	014536		TYPE	8\$;;TYPE TWO(2) SPACES
3141	014534	000771			BR	7\$;;LOOP
3142	014536	020040	000		8\$:	.ASCIZ / /	;;TWO(2) SPACES
3143		014542				.EVEN	
3144						.SBTTL	BINARY TO OCTAL (ASCII) AND TYPE
3145							
3146							
3147							
3148							
3149							
3150							
3151							
3152							
3153							
3154							
3155							
3156							
3157							
3158							
3159							
3160							
3161							
3162							
3163							
3164							
3165							
3166							
3167							
3168							
3169	014542	017646	000000				
3170	014546	116667	000001	000211	\$TYPOS:	MOV	2(SP), -(SP) ; ; PICKUP THE MODE
3171	014554	112667	000207			MOVB	1(SP), \$OFILL ; ; LOAD ZERO FILL SWITCH
3172	014560	062716	000002			MOVB	(SP)+, \$OMODE+1 ; ; NUMBER OF DIGITS TO TYPE
3173	014564	000406				ADD	#2, (SP) ; ; ADJUST RETURN ADDRESS
3174	014566	112767	000001	000171	\$TYPOC:	BR	\$TYPON
3175	014574	112767	000006	000165		MOVB	#1, \$OFILL ; ; SET THE ZERO FILL SWITCH
3176	014602	112767	000005	000154	\$TYPON:	MOVB	#6, \$OMODE+1 ; ; SET FOR SIX(6) DIGITS
3177	014610	010346				MOVB	#5, \$OCNT ; ; SET THE ITERATION COUNT
3178	014612	010446				MOV	R3, -(SP) ; ; SAVE R3
3179	014614	010546				MOV	R4, -(SP) ; ; SAVE R4
3180	014616	116704	000145			MOV	R5, -(SP) ; ; SAVE R5
3181	014622	005404				MOVB	\$OMODE+1, R4 ; ; GET THE NUMBER OF DIGITS TO TYPE
3182	014624	062704	000006			NEG	R4
3183	014630	110467	000132			ADD	#6, R4 ; ; SUBTRACT IT FOR MAX. ALLOWED
3184	014634	116704	000125			MOVB	R4, \$OMODE ; ; SAVE IT FOR USE
3185	014640	016605	000012			MOVB	\$OFILL, R4 ; ; GET THE ZERO FILL SWITCH
3186	014644	005003				MOV	12(SP), R5 ; ; PICKUP THE INPUT NUMBER
3187	014646	006105			1\$:	CLR	R3 ; ; CLEAR THE OUTPUT WORD
3188	014650	000404				ROL	R5 ; ; ROTATE MSB INTO "C"
3189	014652	006105			2\$:	BR	3\$; ; GO DO MSB
						ROL	R5 ; ; FORM THIS DIGIT

3190	014654	006105			ROL	R5		
3191	014656	006105			ROL	R5		
3192	014660	010503			MOV	R5,R3		
3193	014662	006103			3\$: ROL	R3		:: GET LSB OF THIS DIGIT
3194	014664	105367	000076		DECB	\$OMODE		:: TYPE THIS DIGIT?
3195	014670	100016			BPL	7\$:: BR IF NO
3196	014672	042703	177770		BIC	#177770,R3		:: GET RID OF JUNK
3197	014676	001002			BNE	4\$:: TEST FOR 0
3198	014700	005704			TST	R4		:: SUPPRESS THIS 0?
3199	014702	001403			BEQ	5\$:: BR IF YES
3200	014704	005204			4\$: INC	R4		:: DON'T SUPPRESS ANYMORE 0'S
3201	014706	052703	000060		BIS	#'0,R3		:: MAKE THIS DIGIT ASCII
3202	014712	052703	000040		5\$: BIS	#' ,R3		:: MAKE ASCII IF NOT ALREADY
3203	014716	110367	000040		MOV8	R3,8\$:: SAVE FOR TYPING
3204	014722	104401	014762		TYPE	8\$:: GO TYPE THIS DIGIT
3205	014726	105367	000032		7\$: DECB	\$OCNT		:: COUNT BY 1
3206	014732	003347			BGT	2\$:: BR IF MORE TO DO
3207	014734	002402			BLT	6\$:: BR IF DONE
3208	014736	005204			INC	R4		:: INSURE LAST DIGIT ISN'T A BLANK
3209	014740	000744			BR	2\$:: GO DO THE LAST DIGIT
3210	014742	012605			6\$: MOV	(SP)+,R5		:: RESTORE R5
3211	014744	012604			MOV	(SP)+,R4		:: RESTORE R4
3212	014746	012603			MOV	(SP)+,R3		:: RESTORE R3
3213	014750	016666	000002	000004	MOV	2(SP),4(SP)		:: SET THE STACK FOR RETURNING
3214	014756	012616			MOV	(SP)+,(SP)		
3215	014760	000002			RTI			:: RETURN
3216	014762	000			8\$: .BYTE	0		:: STORAGE FOR ASCII DIGIT
3217	014763	000			.BYTE	0		:: TERMINATOR FOR TYPE ROUTINE
3218	014764	000			\$OCNT: .BYTE	0		:: OCTAL DIGIT COUNTER
3219	014765	000			\$OFILL: .BYTE	0		:: ZERO FILL SWITCH
3220	014766	000000			\$OMODE: .WORD	0		:: NUMBER OF DIGITS TO TYPE
3221								:: ENTER HERE ON POWER FAILURE
3222								
3223								
3224	014770				\$PWRDN:			
3225	014770	010046			.PFAIL: MOV	RO,-(SP)		:: SAVE RO-R5 ON PROCESSOR STACK
3226	014772	010146			MOV	R1,-(SP)		
3227	014774	010246			MOV	R2,-(SP)		
3228	014776	010346			MOV	R3,-(SP)		
3229	015000	010446			MOV	R4,-(SP)		
3230	015002	010546			MOV	R5,-(SP)		
3231	015004	016746	163014		MOV	24,-(SP)		
3232	015010	010667	164102		MOV	SP,SAVSP		:: SAVE STACK POINTER
3233	015014	012767	015026	163002	MOV	#RESTART,24		:: SET UP FOR POWER UP TRAP
3234	015022	000000			HALT			:: HALT ON POWER DOWN NORMAL
3235	015024	000777			BR	.		
3236								
3237								
3238								:: PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
3239	015026	016706	164064		RESTAR: MOV	SAVSP,SP		:: RESTORE STACK POINTER
3240	015032	012605			MOV	(SP)+,R5		:: RESTORE RO-R5
3241	015034	012604			MOV	(SP)+,R4		
3242	015036	012603			MOV	(SP)+,R3		
3243	015040	012602			MOV	(SP)+,R2		
3244	015042	012601			MOV	(SP)+,R1		
3245	015044	012600			MOV	(SP)+,R0		

M05

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 66
 DZDUSA.M11 13-OCT-76 08:39 BINARY TO OCTAL (ASCII) AND TYPE

3246	015046	012767	014770	162750	MOV	#.PFail,24	
3247	015054	106427	000340		MTPS	#340	;SET UP FOR POWER FAILURE
3248	015060	012706	001100		MOV	#STACK,SP	
3249	015064	005067	001102		CLR	TEMP	
3250	015070	005267	001076		INC	TEMP	
3251	015074	001375			BNE	.-4	
3252	015076	104413			CONVRT		
3253	015100	015122			PFTAB		
3254	015102	104401			TYPE		
3255	015104	015424			MPFAIL		
3256	015106	005067	164271		CLR	\$ERFLG	
3257	015112	005067	164300		CLR	\$ERRPC	
3258	015116	000177	163762		JMP	\$RETURN	
3259	015122	000001			PFTAB:	1	
3260	015124	006	002		.BYTE	6,2	
3261	015126	000207			RETURN		
3262	015130	005015	042012	053125	MTITLE:	.ASCIZ	<15><12><12>/DUV11 DZDUS-A TAPE C /<15><12>
3263	015136	030461	042040	042132			
3264	015144	051525	040455	052040			
3265	015152	050101	020105	020103			
3266	015160	005015	000				
3267	015163	015	053012	041505	MVECTO:	.ASCIZ	<15><12>/VEC ADD- /
3268	015170	040440	042104	000055			
3269	015176	005015	051461	020124	MREGAD:	.ASCIZ	<15><12>/1ST DEV: REC CSR ADD- /
3270	015204	042504	035126	051040			
3271	015212	041505	041440	051123			
3272	015220	040440	042104	000055			
3273	015226	005015	052515	052114	MMULT:	.ASCIZ	<15><12>/MULT DEV ? (Y OR N)- /
3274	015234	042040	053105	037440			
3275	015242	024040	020131	051117			
3276	015250	047040	026451	000			
3277	015255	015	046012	051501	MLASTD:	.ASCIZ	<15><12>/LAST DEV: REC CSR ADDR- /
3278	015262	020124	042504	035126			
3279	015270	051040	041505	041440			
3280	015276	051123	040440	042104			
3281	015304	026522	000				
3282	015307	075	042504	044526	DEVICE:	.ASCIZ	/=DEVICE /
3283	015314	042503	020040	000			
3284	015321	015	051412	046105	MCOW:	.ASCIZ	<15><12>/SELECT TO RUN \$ACTREG /
3285	015326	041505	020124	047524			
3286	015334	051040	047125	040040			
3287	015342	041501	051124	043505			
3288	015350	000					
3289	015351	015	047412	043126	MRANGE:	.ASCIZ	<15><12>/OVFLO:RETYPE LAST DEV RXCSR ADDS- /
3290	015356	047514	051072	052105			
3291	015364	050131	020105	040514			
3292	015372	052123	042040	053105			
3293	015400	051040	041530	051123			
3294	015406	040440	042104	026523			
3295	015414	000					
3296	015415	040	037440	000	MQM:	.ASCIZ	/ ? /
3297	015421	015	000012		MCRLF:	.ASCIZ	<15><12>
3298	015424	043120	044501	026114	MPFAIL:	.ASCIZ	/PFail, RESTART AT TEST IN PROGRESS /
3299	015432	020040	042522	052123			
3300	015440	051101	020124	052101			
3301	015446	052040	051505	020124			

N05

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 67
 DZDUSA.M11 13-OCT-76 08:39 BINARY TO OCTAL (ASCII) AND TYPE

3302	015454	047111	050040	047522	
3303	015462	051107	051505	000123	
3304	015470	005015	047105	020104	MEPASS: .ASCIZ <15><12>/END OF PASS TAPE C/
3305	015476	043117	050040	051501	
3306	015504	020123	040524	042520	
3307	015512	041440	000		
3308	015515	015	051012	000	MR: .ASCIZ <15><12>/R/
3309	015521	015	052012	051505	MTSTPC: .ASCIZ <15><12>/TEST PC-/
3310	015526	020124	041520	000055	
3311	015534	005015	047514	045503	MLOCK: .ASCIZ <15><12>/LOCK ON TEST? (Y OR N)-/
3312	015542	047440	020116	052040	
3313	015550	051505	037524	024040	
3314	015556	020131	051117	047040	
3315	015564	026451	000		
3316	015567	015	021412	047440	MSYNC: .ASCIZ <15><12>/# OF SYNC CHARS SELECTED (1 OR 2)-/
3317	015574	020106	054523	041516	
3318	015602	041440	040510	051522	
3319	015610	051440	046105	041505	
3320	015616	042524	020104	020050	
3321	015624	020061	051117	031040	
3322	015632	026451	000		
3323	015635	015	044412	020123	MWIRE6: .ASCIZ <15><12>/IS SEC XMIT SWITCH E55-2 IN? (Y OR N)-/
3324	015642	042523	020103	046530	
3325	015650	052111	051440	044527	
3326	015656	041524	020110	032505	
3327	015664	026465	020062	047111	
3328	015672	020077	054450	047440	
3329	015700	020122	024516	000055	
3330	015706	005015	051511	051440	MWIRE5: .ASCIZ <15><12>/IS SEC REC SWITCH E55-3 IN? (Y OR N)-/
3331	015714	041505	051040	041505	
3332	015722	051440	044527	041524	
3333	015730	020110	032505	026465	
3334	015736	020063	047111	020077	
3335	015744	054450	047440	020122	
3336	015752	024516	000055		
3337	015756	005015	051511	047440	MWIRE4: .ASCIZ <15><12>/IS OPT CLR ENABLE SWITCH E55-1 IN? (Y OR N)-/
3338	015764	052120	041440	051114	
3339	015772	042440	040516	046102	
3340	016000	020105	053523	052111	
3341	016006	044103	042440	032465	
3342	016014	030455	044440	037516	
3343	016022	024040	020131	051117	
3344	016030	047040	026451	000	
3345	016035	015	005012	031510	MEXTJ: .ASCIZ <15><12><12>/H315 CONNECTOR ON ?(Y OR N)-/
3346	016042	032461	041440	047117	
3347	016050	042516	052103	051117	
3348	016056	047440	020116	024077	
3349	016064	020131	051117	047040	
3350	016072	026451	000		
3351	016075	015	020012	043536	MCNTG: .ASCIZ <15><12>/ TG /
3352	016102	020040	000		
3353	016105	040	053523	036522	MMSWR: .ASCIZ / SWR= /
3354	016112	020040	002040		
3355	016116	020040	047040	053505	MMNEW: .ASCIZ / NEW= /
3356	016124	020075	000040		
3357					.EVEN

```

3358
3359 ;BUFFERS FOR INPUT-OUTPUT
3360
3361 016130 00000C INBUF: 0
3362 ;. +40
3363 016172 000000 TEMP: 0
3364 ;. +40
3365 016234 000000 MDATA: 0
3366 ;. +40
3367 ;SBTTL SCOPE HANDLER ROUTINE
3368
3369 ;*****
3370 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
3371 ;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
3372 ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
3373 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3374 ;*SW14=1 LOOP ON TEST
3375 ;*SW11=1 INHIBIT ITERATIONS
3376 ;*SW09=1 LOOP ON ERROR
3377 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
3378 ;*CALL
3379 ;* SCOPE ;;SCOPE=IOT
3380
3381 016276 $$SCOPE:
3382
3383 ;SCOPE LOOP AND INTERATION HANDLER
3384
3385 .SCOPE:
3386 016276 004767 174376 JSR PC,CKSWR
3387 016302 005067 163110 CLR $ERRPC ;CLEAR LAST ERROR PC
3388 016306 022716 003364 CMP #TST1+2,(SP) ;IS SCOPE AT BEGINING OF TEST 1?
3389 016312 001413 BEQ $XTSTR ;YES NO LOOP.
3390
3391 016314 000406 TTST: BR 1$ ;GO TO 1$ (IF LOCK SW02=1)
3392 016316 105777 163122 TSTB 2$TKS ;KEYBOARD DONE?
3393 016322 100123 BPL $OVER ;BR IF NO
3394 016324 017766 163116 177776 MOV 2$TKB,-2(SP) ;CLEAR DONE BIT
3395 016332 032777 040000 163100 1$: BIT #BIT14,2$SWR ;LOOP ON PRESENT TEST?
3396 016340 001114 BNE $OVER ;YES IF SW14=1
3397 ;*****START OF CODE FOR THE XOR TESTER*****
3398 016342 000416 $XTSTR: BR 6$ ;IF RUNNING ON THE "XOR" TESTER CHANGE
3399 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
3400 016344 013746 000004 MOV 2$ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3401 016350 012737 016370 000004 MOV #55,2$ERRVEC ;SET FOR TIMEOUT
3402 016356 005737 177060 TST 2$177060 ;TIME OUT ON XOR?
3403 016362 012637 000004 MOV (SP)+,2$ERRVEC ;RESTORE THE ERROR VECTOR
3404 016366 000463 BR $SVLAD ;GO TO THE NEXT TEST
3405 016370 022626 5$: CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
3406 016372 012637 000004 MOV (SP)+,2$ERRVEC ;RESTORE THE ERROR VECTOR
3407 016376 000423 BR 7$ ;LOOP ON THE PRESENT TEST
3408 016400 6$: ;*****END OF CODE FOR THE XOR TESTER*****
3409 016400 032777 000400 163032 BIT #BIT08,2$SWR ;LOOP ON SPEC. TEST?
3410 016406 001404 BEQ 2$ ;BR IF NO
3411 016410 127767 163024 162764 CMPB 2$SWR,$TSTNM ;ON THE RIGHT TEST? SWR<7:0>
3412 016416 001465 BEQ $OVER ;BR IF YES
3413 016420 105767 162757 2$: TSTB $ERFLG ;HAS AN ERROR OCCURRED?
    
```

```

3414 016424 001421          BEQ      35          ;; BR IF NO
3415 016426 126767 162763 162747  CMPB   $ERMAX,$ERFLG ;; MAX. ERRORS FOR THIS TEST OCCURRED?
3416 016434 101015          BHI     35          ;; BR IF NO
3417 016436 032777 001000 162774  BIT    #BIT09,$SWR   ;; LOOP ON ERROR?
3418 016444 001404          BEQ     45          ;; BR IF NO
3419 016446 016767 162736 162732 75:   MOV    $LPERR,$LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
3420 016454 000446          BR      $OVER
3421 016456 105067 162721          45:   CLRB  $ERFLG        ;; ZERO THE ERROR FLAG
3422 016462 005067 163024          CLR    $TIMES       ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
3423 016466 000415          BR      15         ;; ESCAPE TO THE NEXT TEST
3424 016470 032777 004000 162742 35:   BIT    #BIT11,$SWR   ;; INHIBIT ITERATIONS?
3425 016476 001011          BNE     15         ;; BR IF YES
3426 016500 005767 163030          TST    $PASS        ;; IF FIRST PASS OF PROGRAM
3427 016504 001406          BEQ     15         ;; INHIBIT ITERATIONS
3428 016506 005267 162672          INC    $ICNT        ;; INCREMENT ITERATION COUNT
3429 016512 026767 162774 162664  CMP    $TIMES,$ICNT  ;; CHECK THE NUMBER OF ITERATIONS MADE
3430 016520 002024          BGE    $OVER        ;; BR IF MORE ITERATION REQUIRED
3431 016522 012767 000001 162654 15:   MOV    #1,$ICNT     ;; REINITIALIZE THE ITERATION COUNTER
3432 016530 016767 000056 162754          MOV    $MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
3433 016536 105267 162640          $SVLAD: INCB  $STNM        ;; COUNT TEST NUMBERS
3434 016542 116767 162634 162762  MOVB  $STNM,$STNM   ;; SET TEST NUMBER IN APT MAILBOX
3435 016550 011667 162632          MOV    (SP),$LPADR  ;; SAVE SCOPE LOOP ADDRESS
3436 016554 011667 162630          MOV    (SP),$LPERR  ;; SAVE ERROR LOOP ADDRESS
3437 016560 005067 162730          CLR    $ESCAPE     ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
3438 016564 112767 000001 162623  MOVB  #1,$ERMAX     ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
3439 016572 016777 162604 162642  $OVER: MOV    $STNM,$DISPLAY ;; DISPLAY TEST NUMBER
3440 016600 016716 162602          MOV    $LPADR,(SP) ;; FUDGE RETURN ADDRESS
3441 016604 000002          45:   RTI
3442 016606 001407          BRW:   1407
3443 016610 000432          BRX:   432
3444 016612 000005          $MXCNT: 5          ;; MAX. NUMBER OF ITERATIONS
3445          .SBTTL TRAP DECODER
3446
3447          ;; *****
3448          ;; *THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3449          ;; *AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3450          ;; *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3451          ;; *GO TO THAT ROUTINE.
3452
3453 016614 010046          $TRAP: MOV    RO,-(SP)    ;; SAVE RO
3454 016616 016600 000002  MOV    2(SP),RO     ;; GET TRAP ADDRESS
3455 016622 005740  TST    -(RO)        ;; BACKUP BY 2
3456 016624 111000  MOVB   (RO),RO      ;; GET RIGHT BYTE OF TRAP
3457 016626 006300  ASL    RO           ;; POSITION FOR INDEXING
3458 016630 016000 016650  MOV    $TRPAD(RO),RO ;; INDEX TO TABLE
3459 016634 000200  RTS     RO          ;; GO TO ROUTINE
3460
3461
3462          ;; THIS IS USE TO HANDLE THE "GETPRI" MACRO
3463
3464 016636 011646          $TRAP2: MOV   (SP),-(SP) ;; MOVE THE PC DOWN
3465 016640 016666 000004 000002  MOV   4(SP),2(SP)    ;; MOVE THE PSW DOWN
3466 016646 000002  RTI                ;; RESTORE THE PSW
3467
3468          .SBTTL TRAP TABLE
3469

```

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

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

016650 016636
016652 013050
016654 014566
016656 014542
016660 014602

016662 013026
016664 013332
016666 013440
016670 013450
016672 013650
016674 013710
016676 013742
016700 014120

016702 006367 000044
016706 006367 000040
016712 006367 000034
016716 006367 000030
016722 006367 000024
016726 016767 000020 000020
016734 162767 000001 000012
016742 042767 000037 000004

016756 016767 000126 162726
016764 005267 000120
016770 016767 000114 162716
016776 005267 000106
017002 016767 000102 162706
017010 016767 000074 162704
017016 005267 000066
017022 016767 000062 162670
017030 016767 000054 162666
017036 005267 000046
017042 016767 000042 162656
017050 005267 000034
017054 016767 000030 162646
017062 005267 000022
017066 016767 000016 162636
017074 005267 000010
017100 016767 000004 162626
017106 000207

ROUTINE

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

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

;UTILITIES

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

;NEW DU ADDRESSES
DUADDR: MOV DUBASE,RXCSR ;XXX0
INC DUBASE
MOV DUBASE,HRXCSR ;XXX1
INC DUBASE
MOV DUBASE,RXDBUF ;XXX2
MOV DUBASE,PARCSR ;XXX2
INC DUBASE
MOV DUBASE,HRXDBUF ;XXX3
MOV DUBASE,HPARCSR ;XXX3
INC DUBASE
MOV DUBASE,TXCSR ;XXX4
INC DUBASE
MOV DUBASE,HTXCSR ;XXX5
INC DUBASE
MOV DUBASE,TXDBUF ;XXX6
INC DUBASE
MOV DUBASE,HTXDBUF ;XXX7
RTS PC

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 71
 DZDUSA.M11 13-OCT-76 08:39 TRAP TABLE

```

3526 017110 000000          DUBASE: 0
3527
3528
3529
3530
3531 017112 042777 040000 162606 RPOKE: BIC    #MTDATA,@TXCSR
3532 017120 005067 162356          CLR    $TMP2
3533 017124 006067 162350          ROR    $TMP1 ; FORCE CARRY
3534 017130 006067 162346          ROR    $TMP2 ; PICK UP CARRY IN BIT 15
3535 017134 006267 162342          ASR    $TMP2 ; SHIFT INTO BIT 14
3536 017140 042767 100000 162334 BIC    #BIT15,$TMP2 ; CLR BIT 15
3537 017146 056777 162330 162552 BIS    $TMP2,@TXCSR ; POKE MAINT DATA
3538 017154 042777 020000 162544 BIC    #CLK,@TXCSR ; POKE CLK
3539 017162 052777 020000 162536 BIS    #CLK,@TXCSR ;
3540 017170 005367 161726          DEC    SHIFT
3541 017174 001346          BNE    RPOKE
3542 017176 000207          RTS    PC
3543
3544 017200 016767 162274 162274 ODD8:  MOV    $TMP1,$TMP2 ; SAVE TEMP1
3545 017206 005067 162272          CLR    $TMP3
3546 017212 012727 000010          MOV    #8,(PC)+
3547 017216 000000          4$:   0
3548 017220 006067 162256          1$:   ROR    $TMP2
3549 017224 005567 162254          ADC    $TMP3
3550 017230 005367 177762          DEC    4$
3551 017234 001371          BNE    1$
3552 017236 006067 162242          ROR    $TMP3
3553 017242 103404          BCS    2$
3554 017244 052767 000400 162226 BIS    #BIT8,$TMP1 ; SET ODD PARITY
3555 017252 000403          BR     3$
3556 017254 042767 000400 162216 2$:   BIC    #BIT8,$TMP1 ; CLR EVEN PARITY
3557
3558 017262 000207          3$:   $TMP1 NOW HAS ODD PARITY CHARACTER
3559          RTS    PC
3560
3561 017264 016767 162210 162210 ; THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
3562 017272 005067 162206          EVEN8: MOV    $TMP1,$TMP2 ; SAVE TEMP1
3563 017276 012727 000010          CLR    $TMP3
3564 017302 000000          MOV    #8,(PC)+
3565 017304 006067 162172          4$:   0
3566 017310 005567 162170          1$:   ROR    $TMP2
3567 017314 005367 177762          ADC    $TMP3
3568 017320 001371          DEC    4$
3569 017322 006067 162156          BNE    1$
3570 017326 103004          ROR    $TMP3
3571 017330 052767 000400 162142 BCC    2$
3572 017336 000403          BIS    #BIT8,$TMP1 ; SET EVEN PARITY
3573 017340 042767 000400 162132 2$:   BR     3$
3574
3575 017346 000207          3$:   BIC    #BIT8,$TMP1 ; CLR ODD PARITY
3576
3577 017350 062716 000002          TRPREG: $TMP1 NOW HAS EVEN PARITY CHARACTER
3578
3579 017354 000002          RTS    PC
3580 000001          .END

```

F06

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 73
 DZCUSA.M11 13-OCT-76 08:39 CROSS REFERENCE TABLE -- USER SYMBOLS

AAA	003200	1291*								
RBASE =	000000	874	915							
ACDW1 =	000000	874	917							
ACDW2 =	000000	874	918							
ACPUOP=	000000	874	889							
ACTREG	001166	733*	1247*	1261*	1262*	1269*	2694	2697	2707	
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=	013647	2909*	2948*	2955*						
RENV =	000000	874	885							
REVM =	000000	874	886							
RFATAL=	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							
AMSGAD=	000000	874	882							
AMSGLG=	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*	2824							
APTENV=	000001	534*	2817	3071						
APTSIZ=	000200	534*	1166							
APTSPO=	000100	534*	2819							
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*	2701*	2713*	2717

BASEIV	001162	731#	1239*	2702*	2714*	2719	2720*	2721	2722*	2723	2724*	2725	2726*
888	003026	1246	1250#										
BINWRD	014116	2990*	2992	3022#									
BIT# =	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	3409									
BIT09 =	001000	644#	654	3081	3417								
BIT1 =	000002	662#	775	968									
BIT10 =	002000	643#	766	800	959	993	2163	2207	3059				
BIT11 =	004000	642#	765	958	3424								
BIT12 =	010000	641#	764	781	957	974							
BIT13 =	020000	640#	763	780	799	956	973	992	3066				
BIT14 =	040000	639#	762	779	798	955	972	991	3395				
BIT15 =	100000	638#	761	778	797	954	971	990	3536				
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	991	977	994	3554	3556	3571	3573	
BIT9 =	001000	654#	767	783	960	976							
BPTVEC =	000014	670#											
BREAK =	000001	807#	1000#	1356	1408	1460	1512	1564	1616	1668	1707	1746	1785
		1864	1905	1946	1984	2025	2066	2107	2144	2188	2232	2270	2309
		2381	2417	2453	2486	2519	2552	2585	2621	2657			1823
		3442#											2345
BRW	016606	3443#											
BRX	016610	764#	957#										
CARDET =	010000	2693	2716	2733#									
CCC	012600	2988#	2993	3006*	3020#	3021							
CHRCNT	014114	2687	2749	2757#	2789	3386							
CKSMR	01270C	799#	992#	1356	1362	1363	1408	1414	1415	1460	1466	1467	1512
CLK =	020000	1519	1564	1570	1571	1616	1622	1623	1668	1674	1675	1677	1678
		1713	1714	1716	1717	1746	1752	1753	1755	1756	1785	1791	1792
		1795	1823	1829	1830	1832	1833	1864	1870	1871	1873	1874	1905
		1912	1914	1915	1946	1952	1953	1955	1956	1984	1990	1991	1993
		2025	2031	2032	2034	2035	2066	2072	2073	2075	2076	2107	2113
		2116	2117	2144	2150	2151	2153	2154	2188	2194	2195	2197	2198
		2238	2239	2270	2276	2277	2309	2315	2316	2318	2319	2345	2351
		2354	2355	2381	2387	2388	2390	2391	2417	2423	2424	2426	2427
		2459	2460	2486	2492	2493	2519	2525	2526	2552	2558	2559	2585
		2592	2594	2595	2621	2627	2628	2630	2631	2657	2663	2664	2666
		3538	3539										2667
CNTLU	012754	1196	2768#										
CONVRT =	104413	2690	2770	3252	3488#								
COUNT	001124	708#	2321*	2328*	2357*	2364*	2393*	2400*	2429*	2436*	2462*	2469*	2495*
		2528*	2535*	2561*	2568*	2597*	2604*	2633*	2640*	2669*	2676*		2502*

\$ITEMB	001414	828#	3065*	3073	3089	3108												
\$LF	001524	869#	2873	3089														
\$LFLG	000243R	534#*																
\$LPADR	001406	825#	1147*	2750*	3419*	3435*	3440	3444										
\$LPERR	001410	826#	1148*	3083	3419	3436*	3444											
\$MADR1	001560	902#																
\$MADR2	001564	906#																
\$MADR3	001570	909#																
\$MADR4	001574	912#																
\$MAIL	001526	875#	1113	1117	1165	2817	3071	3434										
\$MAMS1	001556	896#																
\$MAMS2	001562	904#																
\$MAMS3	001566	907#																
\$MAMS4	001572	910#																
\$MBADR	002140	1113#																
\$MFLG	000242R	534#*																
\$MSGAD	001542	534#	882#															
\$MSGLG	001544	534#	883#															
\$MSGTY	001526	534#	876#															
\$MTYP1	001557	897#																
\$MTYP2	001563	905#																
\$MTYP3	001567	908#																
\$MTYP4	001573	911#																
\$MXCNT	016612	3432	3444#															
\$N =	000000	532#	2680#															
\$NULL	001454	846#	2844	2873														
\$NWTST =	000000	1349#	1401#	1453#	1505#	1557#	1609#	1661#	1700#	1739#	1778#	1816#	1857#	1898#				
		1939#	1977#	2018#	2059#	2100#	2137#	2181#	2225#	2263#	2302#	2338#	2374#	2410#				
		2446#	2479#	2512#	2545#	2578#	2614#	2650#										
\$OCNT	014764	3176#	3205#	3218#														
\$OMODE	014766	3171#	3175#	3180	3183#	3194#	3220#											
\$OVER	016572	3393	3396	3412	3420	3430	3439#											
\$PASS	001534	879#	1165*	2739*	3426	3445												
\$PASTH	002144	1115#																
\$PWRON	014770	1142	3224#															
\$QUES	001522	867#	2873	3089														
\$ROCHR =	***** U	3482																
\$RODEC =	***** U	3482																
\$ROLIN =	***** U	3482																
\$ROOCT =	***** U	3482																
\$REGAD	001460	850#																
\$REGO	001462	852#	2968#	2973														
\$REG1	001464	853#	2967#	2974														
\$REG2	001466	854#	2966#	2975														
\$REG3	001470	855#	2965#	2976														
\$REG4	001472	856#	2964#	2977														
\$REG5	001474	857#	2963#	2978														
\$R2A =	***** U	3482																
\$SAVRE =	***** U	3482																
\$SCOPE	016276	1136	3381#															
\$SETUP =	000017	1118#	1135	1136	1138	1140	1142	1144	1145	1147	3056	3081	3088	3382				
\$STUP =	177777	1118#																
\$SYAD	016536	3404	3433#															
\$SVPC =	002136	1090#	1095															
\$SWR =	177400	523#	864	865	866	1144	1145	1147	1148	1351	1403	1455	1507	1559				
		1611	1663	1702	1741	1780	1818	1859	1900	1941	1979	2020	2061	2102				

		2139	2183	2227	2265	2304	2340	2376	2412	2448	2481	2514	2547	2580
		2616	2652	3047	3048	3049	3050	3051	3059	3066	3078	3081	3089	3373
		3374	3375	3376	3377	3395	3407	3409	3410	3413	3414	3415	3422	3423
		3424	3436	3439	3444									
SSWREG	00.750	887#	1168											
SSWRNK=	0006J0	3377	3378	3411										
STESTN	001532	878#	3434#											
STINES	001512	864#	1144#	3422#	3429	3432#	3444							
STKB	001446	843#	2763	2886	2894	3394								
STKS	001444	842#	2761	2884	3392									
STMPO	001476	858#												
STMP1	001500	859#	1367#	1379#	1383#	1419#	1431#	1435#	1471#	1483#	1487#	1523#	1535#	1539#
		1575#	1587#	1591#	1627#	1639#	1643#	1680#	1719#	1758#	1797#	1837#	1878#	1919#
		1960#	1998#	2039#	2080#	2121#	2159#	2162#	2163#	2203#	220	2207#	2244#	2282#
		2322#	2358#	2394#	2430#	2463#	2496#	2529#	2562#	2598#	2634#	2670#	3533#	3544
		3554#	3556#	3561	3571#	3573#								
STMP2	001502	860#	3532#	3534#	3535#	3536#	3537	3544#	3548#	3561#	3565#			
STMP3	001504	861#	3545#	3549#	3552#	3562#	3566#	3569#						
STMP4	001506	862#												
STMP5	001510	863#												
STN =	000042	534#	1349	1351#	1401	1403#	1453	1455#	1505	1507#	1557	1559#	1609	1611#
		1661	1663#	1700	1702#	1739	1741#	1778	1780#	1816	1818#	1857	1859#	1898
		1900#	1939	1941#	1977	1979#	2018	2020#	2059	2061#	2100	2102#	2137	2139#
		2181	2183#	2225	2227#	2263	2265#	2302	2304#	2338	2340#	2374	2376#	2410
		2412#	2446	2448#	2479	2481#	2512	2514#	2545	2547#	2578	2580#	2614	2616#
		2650	2652#											
STPB	001452	845#	2862#	2873	2894#									
STPFLG	001457	849#	2811	2873										
STPS	001450	844#	2860	2873	2892									
STRAP	016614	1140	3453#											
STRAP2	016636	3464#	3475											
STRP =	000015	3468#	3477#	3478#	3479#	3480#	3482	3483#	3484#	3485#	3486#	3487#	3488#	3489#
		3490#												
STRPAD	016650	3458	3475#											
STSTM	002142	1114#												
STSTM	001402	822#	1178#	2734#	3058	3089	3095	3372	3411	3433#	3434	3439	3445	
STYPB=	*****	3480												
STYPDS=	*****	3480												
STYPE	013050	534	2811#	3468	3476									
STYPEC	013262	2841	2848	2855	2860#	2861								
STYPEX	013330	2866	2868	2871#										
STYPOC	014566	3174#	3477											
STYPOH	014602	3173	3176#	3479										
STYPOS	014542	3169#	3478											
SUNIT	001540	881#												
SUNITM	002146	1116#												
SUSWR	001552	888#	1203											
SVECT1	001576	913#												
SVECT2	001600	914#												
SXTSTR	016342	3389	3398#											
SOFILL	014765	3170#	3174#	3184	3219#									
S4OCAT=	*****	3068	3395											
	= 017356	534#	568#	681#	684#	689#	744#	745#	819#	870	1076#	1090	1091#	1093#
		1095#	1101	1102#	1104#	1106#	1133	1147	1148	1370	1374	1389	1422	1426
		1441	1474	1478	1493	1526	1530	1545	1578	1582	1597	1630	1634	1649
		1687	1726	1765	1804	1840	1844	1881	1885	1922	1926	1963	1967	2001

G07

DZDUS-A MACY11 27(1006) 03-FEB-77 07:52 PAGE 88
DZDUSA.M11 13-OCT-76 09:39 CROSS REFERENCE TABLE -- MACRO NAMES

.HEADE	523#	
.SETUP	523#	1118
.SACT1	523#	1086
.SAPT8	523#	871#
.SAPTH	523#	1096
.SAPTY	523#	534
.SCATC	523#	
.SCMTA	523#	813
.SEOP	523#	
.SERRO	523#	3041
.SERRT	523#	3097
.SPOWE	523#	
.SSCOP	523#	3367
.STRAP	523#	3445
.STYPE	523#	2794
.STYPO	523#	3144

. ABS. 017356 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDUSA, DZDUSA.SEQ/SOL/CRF+DZDUS1/EQ:RUNC, DZDUS2, DZDUSA
RUN-TIME: 18 12 1 SECONDS
RUN-TIME RATIO: 118/33=3.5
CORE USED: 31K (62 PAGES)