

.REM

IDENTIFICATION

PRODUCT CODE: AC-E998B-MC
PRODUCT NAME: CXDLBB0 DL11-E MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE

1.0 ABSTRACT

DXDLB IS AN IOMOD THAT EXERCISES ONE DL11-E ASYNCHRONOUS COMMUNICATIONS INTERFACE (M7800). THE PROGRAM CONSISTS OF TWO MAJOR SECTIONS AS DESCRIBED BELOW:

SECTION ONE:

THE FIRST SECTION CONSISTS OF A LOGICALLY SEQUENCED SET OF STATIC REGISTER TESTS TO VERIFY THE DL11-E HARDWARE REQUIRED TO PERFORM INPUT/OUTPUT DATA TRANSFERS IN INTERRUPT MODE. ERRORS DETECTED IN THIS SECTION THAT ARE DETERMINED TO BE FATAL ARE REPORTED VIA THE STANDARD DEC/X11 ERROR PRINTOUT AND THEN THE MODULE IS DROPPED FROM THE EXERCISE. NON-FATAL ERRORS ARE SIMPLY REPORTED AND THEN THE PROGRAM CONTINUES IN NORMAL SEQUENCE.

SECTION TWO:

THE SECOND SECTION TRANSFERS 256. BYTE BLOCKS OF DATA USING THE MAINTENANCE MODE TO TURN THE DATA AROUND. THE 256. BYTES OUTPUT ARE COMPARED WITH THE 256. BYTES INPUT FOR DATA COMPARISON ERRORS. ALL DATA COMPARISON ERRORS ARE REPORTED ON THE CONSOLE DEVICE. THE 256. BYTE TRANSFER IS REPEATED FOR FOUR DIFFERENT DATA BIT PATTERNS AS DESCRIBED BELOW:

- A. NULL-DEL-NUL SEQUENCE (000,377,000.....,000,377)
- B. BINARY UP-COUNT SEQUENCE (000,001,002.....,376,377)
- C. BINARY DOWN COUNT SEQUENCE (377,376,375.....,001,000)
- D. WORST CASE PATTERN (376,377,001,000.,000,200)

2.0 REQUIREMENTS

HARDWARE: A PDP11 COMPUTER WITH A DL11-E INTERFACE
STORAGE:: DLB REQUIRES:
1. DECIMAL WORDS: 1354
2. OCTAL WORDS: 02512
3. OCTAL BYTES: 5224

3.0 PASS DEFINITION

ONE PASS OF "DXDLB" CONSISTS OF TWO ITERATIONS OF SECTION TWO OF THE MODULE CODE WHICH RESULTS IN 2048(10) BYTES TRANSFERRED.

4.0 EXECUTION TIME

AT 300 BAUD RUNNING ALONE ON A PDP11/40 A SINGLE ERROR FREE

PASS TAKES APPROXIMATELY 40. SECONDS THIS TIME WILL VARY
DEPENDING UPON THE BAUD RATE AND CPU TYPE.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 175610 VCT: 300 BRI: 4 BR2: 0
DVC: 1 SRI: 0

REQUIRED PARAMETERS:

SRI TO EXERCISE THOSE STATIC TESTS REQUIRING THE USE
OF THE H315 MODEM TEST CONNECTOR (MODEM CONTROL LOGIC)
BIT 15 OF SRI MUST BE SET TO A "1". IE SRI=100000.

NOTE: IF SRI BIT 15=1 AND THE MODEM TEST CONNECTOR
IS NOT INSTALLED, FALSE ERRORS WILL BE REPORTED.

6.0 DEVICE/OPTION SETUP

IF THE MODEM CONTROL LOGIC IS TO BE TESTED, THE USER MUST
DISCONNECT THE MODEM AND CONNECT THE H315 TEST CONNECTOR TO
THE DL11-E DEVICE CABLE. SRI MUST BE SETUP AS DESCRIBED IN
(5.0) OR THE TESTS WILL BE SKIPPED.

7.0 MODULE OPERATION

7.1 TEST SEQUENCES

A. STATIC REGISTER TESTS

- DLT01: TEST THAT ALL BITS IN THE RCSR ARE CLEAR WHEN
THE MODULE IS INITIALIZED TO RUN.
- DLT02: TEST THAT ONLY THE "READY" BIT IS SET
IN THE XCSR WHEN THE MODULE IS INITIALIZED TO RUN.
- DLT03: TEST THAT THE "MAINT" BIT IN THE XCSR CAN BE
SET AND CLEARED.
- DLT04: TEST THAT THE "INTR ENAB" BIT IN THE XCSR CAN
CAUSE AN INTERRUPT TO THE PROPER VECTOR WHEN
SET AND ALSO THAT "INTR ENAB" CLEARS PROPERLY.
- DLT05: TEST THAT A RECEIVER INTERRUPT OCCURS TO THE
PROPER VECTOR WHEN "DONE" GETS SET WITH THE
"INTR ENAB" BIT IN THE RCSR SET TO A ONE.
ALSO TEST THAT THE CORRECT DATA IS RECEIVED.

TESTS DLT06 THRU DLT13 ASSUME THAT THE H315 MODEM

TEST CONNECTOR IS INSTALLED. THE USER INDICATES THIS BY SETTING BIT15 IN SR1. THE MODULE LOOKS AT SR1 AND WILL SKIP AROUND DLT06 THRU DLT13 IF BIT15=0.

DLT06: TEST THAT "REQ TO SEND" CAN ASSERT "RING" WHEN SET AND THAT BOTH "REQ TO SEND" AND "RING" CAN BE CLEARED PROPERLY.

DLT07: TEST THAT "SEC XMIT" WHEN SET ASSERTS "SEC REC" WHICH SETS "DATA SET INT" AND THAT READING THE RXSR CLEARS "DATA SET INT". ALSO TESTS THAT CLEARING "SEC XMIT" NEGATES "SEC REC" WHICH ALSO CAUSES "DATA SET INT" TO SET.

DLT10: TEST THAT "DTR" ASSERTS "CLR TO SEND" AND "CAR DET" WHICH IN TURN SET "DATA SET INT". ALSO TESTS THAT "CLR TO SEND" AND "CAR DET" CLEAR WHEN "DTR" IS CLEARED.

DLT11: TEST THAT "DATA SET INTR ENABLE" CAN BE SET AND CLEARED.

DLT12: TEST THAT "DATA SET INTR ENABLE" IN THE XCSR CAUSES AN INTR. WHEN ENABLED.

DLT13: TEST THAT THE BREAK BIT IN THE XCSR CAN BE SET AND CLEARED.

NOTE: BASIC TESTS DLT01 THRU DLT13 ARE EXECUTED ONLY ONCE WHEN THE MODULE IS FIRST INITIALIZED. IF ANY FATAL ERRORS ARE DETECTED THE MODULE IS DROPPED PRIOR TO THE DATA TRANSFER TESTS. AFTER PASS 1 THE MODULE IS RESTARTED AT THE ENTRY POINT TO THE DATA TRANSFER TESTS.

B. DATA TRANSFER TESTS

AFTER THE BASIC TESTS ARE RUN, FOUR 256(10) BYTE DATA TRANSFERS ARE EXECUTED IN THE MAINTENANCE MODE. EACH 256(10) BYTE BLOCK TRANSFER IS DIFFERENT IN THAT FOUR DIFFERENT DATA PATTERNS ARE XMITTED AND RECEIVED AS DESCRIBED IN PARA. 1.0.

THE TEST SEQUENCE FOR THE DATA TRANSFER TESTS IS AS FOLLOWS:

- 1.) CLEAR BOTH THE INPUT AND OUTPUT BUFFERS IN CORE (256(10) BYTES EACH).
- 2.) LOAD THE OUTPUT BUFFER WITH THE APPROPRIATE DATA PATTERN.
- 3.) ENABLE BOTH THE XMIT AND RCVR INTERRUPTS AND INITIATE THE DATA TRANSFERS.

- 4.) AFTER 256(10) BYTES HAVE BEEN OUTPUT AND INPUT COMPARE THE OUTPUT AND INPUT BUFFERS, BYTE BY BYTE FOR DATA COMPARE ERRORS. REPORT ALL DATA ERRORS ON THE CONSOLE DEVICE.
- 5.) IF ALL FOUR DATA PATTERNS HAVE BEEN TRANSFERRED, GO TO (6) BELOW - IF NOT REPEAT (1) THRU (4) FOR THE NEXT PATTERN.
- 6.) DECREMENT A PASS COUNTER (INITIALIZED TO 2.) AND TEST FOR ZERO. IF ZERO GO TO (7) - IF NOT REPEAT (1) THRU (5) AGAIN.
- 7.) REPORT END OF PASS TO THE MONITOR AND RESTART AT (1) WITH THE FIRST DATA PATTERN.

NOTES:

- (1) ON EACH "XMIT" INTERRUPT THE "READY" FLAG IS TESTED AND IF NOT SET, THE ERROR IS REPORTED AND THE MODULE IS DROPPED. (FALSE INTERRUPTS ARE CLASSIFIED AS FATAL ERRORS).
- (2) ON EACH "RCVR" INTERRUPT THE "DONE" FLAG IS TESTED AND IF NOT SET THE MODULE IS DROPPED THE SAME AS FOR A "XMIT FALSE INTERRUPT".
- (3) IF A SOFT ERROR (PARITY-FRAMING-OVERRUN) IS DETECTED IN RCVR INTR. SERVICE, THE OFFENDING BLOCK TRANSFER IS RESTARTED FROM THE BEGINNING OF THE BLOCK. IF AFTER THREE RETRIES THE ERROR PERSISTS, TRANSFER OF THE OFFENDING DATA PATTERN IS ABORTED AND THE PROGRAM GOES ON TO THE NEXT DATA PATTERN. ALL SOFT ERRORS ARE REPORTED ON THE CONSOLE DEVICE.

7.2 SUBROUTINE ABSTRACTS

SEGX: THIS SUBROUTINE SERVES AS A MINI-MONITOR THAT CONTROLS THE SEQUENCING OF THE FOUR DIFFERENT 256(10) BYTE BLOCK TRANSFERS. IT IS CALLED AFTER THE BASIC TESTS AND PERFORMS THE FOLLOWING FUNCTIONS:

1. CALLS A SUBROUTINE TO CLEAR THE DATA BUFFERS
2. CALLS THE APPROPRIATE SUBROUTINE TO SET UP THE OUTPUT BUFFER WITH THE REQUIRED DATA PATTERN.
3. CALLS A SUBROUTINE TO ENABLE INTERRUPTS AND INITIATE THE DATA TRANSFER.
4. SERVICES RETRIES REQUESTED BY SOFT ERRORS.
5. PERFORMS "BREAK" CALLS TO THE MONITOR TO PREVENT TIMEOUTS FROM HANGING THE MODULE.
6. CALLS THE SUBROUTINE TO CHECK THE DATA BUFFERS WHEN THE BLOCK TRANSFER IS COMPLETE.

KICKOF: THIS SUBROUTINE IS CALLED FROM "SEGX" AND CONTAINS THE
THE CODE TO ENABLE INTERRUPTS AND INITIATE THE BLOCK
TRANSFER FOR EACH 256(10) BYTE BLOCK TRANSFER.

CHKDAT: THIS SUBROUTINE IS CALLED FROM "SFGX" AND CHECKS FOR
DATA COMPARISON ERRORS AFTER EACH BLOCK TRANSFER.

STATR: THIS SUBROUTINE IS CALLED FROM THE BASIC TESTS
AND SETS UP THE ERROR INFORMATION FOR ALL ERRORS
RELATING TO THE RECEIVER CSR.

STATX: THIS ROUTINE IS CALLED FROM THE BASIC TESTS AND SETS
UP THE ERROR INFORMATION FOR ALL ERRORS RELATING TO
THE TRANSMITTER CSR.

CLDLBF: THIS ROUTINE IS CALLED FROM "SEGX" AND CLEARS
BOTH THE OUTPUT AND INPUT DATA BUFFERS IN CORE.

LDOUT1: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO
LOAD THE OUTPUT BUFFER WITH THE NULL-DEL-NULL PATTERN.

LDOUT2: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD
THE OUTPUT BUFFER WITH A BINARY UP-COUNT PATTERN.

LDOUT3: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD
THE OUTPUT BUFFER WITH A BINARY DOWN-COUNT PATTERN.

LDOUT4: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD
THE OUTPUT BUFFER WITH THE MONITOR'S WORST CASE PATTERN.

8.0 OPERATOR OPTIONS

- A. USE THE MOD COMMAND TO MODIFY LOCATION "DLB 16" TO
CHANGE SRI. REFER TO PARA. 5.0.
- B. MODIFYING THE CONTENTS OF MODULE LOCATION "RESTR +2"
ALLOWS THE USER TO VARY THE TOTAL NO. OF BYTES TRANS-
FERRED PER PASS. THIS IS DEFAULTED AT LOAD TIME TO
2 WHICH RESULTS IN 2048. BYTES TRANSFERRED.

9.0 NON-STANDARD ERROR PRINTOUTS

- A. IF ANY ONE OF THE FOUR DATA PATTERNS OUTPUT CANNOT BE SUCCESS-
FULLY COMPLETED DUE TO SOFT ERRORS (3 RETRIES ATTEMPTED)
OR A MONITOR "BREAK" TIMEOUT ONE OF THE FOLLOWING APPROP-
RIATE PRINTOUTS WILL OCCUR:

MSG1: "NULL-DEL-NULL SEQUENCE ABORTED"
MSG2: "BINARY UP-COUNT SEQUENCE ABORTED"
MSG3: "BINARY DOWN-COUNT SEQUENCE ABORTED"

MSG4: "WORST CASE SEQUENCE ABORTED"

B. IF ANY FATAL ERROR DETECTED IN SECTION ONE RESULTS IN A
DECISION TO DROP THE MODULE THE FOLLOWING MESSAGE IS
PRINTED:

"FATAL ERROR DETECTED IN THE STATIC REGISTER TESTS"
AND THE "END" CALL IS EXECUTED TO DROP THE MODULE.

```

;DL11-E DEC/X11 EXERCISER MODULE
000000 IONOD <DLBB > 175610,300,4,,,1024,,161
000000 MODULE 140000,DLBB,175610,300,4,,,1024,,161
; -TITLE DLBB DEC/X11 SYSTEM EXERCISER MODULE
; DDACOM VERSION 6 23-MAY-78
; ***** LIST BIN *****
000000 BEGIN:
000000 046104 041102 040 MODNAM: .ASCII /DLBB / ;MODULE NAME
000005 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006 175610 ADDR: 175610+0 ;1ST DEVICE ADDR.
000010 000300 VECTOR: 300+0 ;1ST DEVICE VECTOR.
000013 200 BR1: .BYTE PRTV4+0 ;1ST BR LEVEL.
000014 000 BR2: .BYTE PRTV+0 ;2ND BR LEVEL.
000016 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000018 000000 SR1: OPEN ;SWITCH REGISTER 1
000020 000000 SR2: OPEN ;SWITCH REGISTER 2
000022 000000 SR3: OPEN ;SWITCH REGISTER 3
000024 000000 SR4: OPEN ;SWITCH REGISTER 4
; *****
000026 140000 STAT: 140000 ;STATUS WORD.
000030 000224 INIT: START ;MODULE START ADDR.
000032 000224 SPOINT: MODDSP ;MODULE STACK POINTER.
000034 000000 PASCNT: 0 ;PASS COUNTER
000036 002000 ICOUNT: 1024. ;# OF ITERATIONS PER PASS=1024.
000040 000000 IONOD: 0 ;LOC TO COUNT ITERATIONS
000042 000000 SDFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000048 000000 HDSPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000050 000000 SVSCHK: 0 ;# OF SYS ERRORS ACCUMULATED
000052 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000054 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE
000056 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062 000000 SVR0: OPEN ;LOC TO SAVE R0.
000064 000000 SVR1: OPEN ;LOC TO SAVE R1.
000066 000000 SVR2: OPEN ;LOC TO SAVE R2.
000070 000000 SVR3: OPEN ;LOC TO SAVE R3.
000072 000000 SVR4: OPEN ;LOC TO SAVE R4.
000074 000000 SVR5: OPEN ;LOC TO SAVE R5.
000076 000000 SVR6: OPEN ;LOC TO SAVE R6.
00100 000000 CSRA: JPEN ;ADDR OF CURRENT CSR.
00102 000000 SBADR: ;ADDR OF GOOD DATA, OR
00104 000000 ACSR: OPEN ;CONTENTS OF CSR.
00106 000000 WASADR: ;ADDR OF BAD DATA, OR
00110 000000 ASTAT: OPEN ;STATUS READ CONTENTS.
00112 000000 ERRTP: ;TYPE OF ERROR
00114 000000 ASB: OPEN ;EXPECTED DATA.
00116 000000 AWAS: OPEN ;ACTUAL DATA.
00118 002334 RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
00120 000000 WDIO: OPEN ;WORDS TO MEMORY PER ITERATION
00122 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
00124 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
00126 000161 IDNUM: 161 ;MODULE IDENTIFICATION NUMBER=161

```

```

000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
; *****
000224 MODDSP:
407 ; *****

```


DLBB80.P11 12-OCT-78 12:01
520 000716 016700 177066 MOV VECTOR,RO ;GET THE BASE VECTOR ADDRESS
521 000752 012720 001162 MOV #45,(RO) ;GO TO 45 ON RCVR INTERRUPT
522 000732 052777 000100 002626 BIC #100,@DLRCSR ;SET PRIORITY
523 000740 032777 000100 002620 BIT #100,@DLRCSR ;SET I.E. IN RCVR CSR
524 000746 081015 BNE IS ;DID IT SET
525 000754 012767 002412 JSR PC,STATR ;BR IF IT DID
526 000762 104405 000000 000000 MOV #25,ERRTYP ;GO SET UP ERROR INFO
527 *****
528 HDRS,REGIN,NULL ;CAN'T SET BIT 06 IN RCSR I.E.
529 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
530 ENDS,REGIN
531 000770 104403 000000 003640 BIC #100,@DLRCSR ;NOW CLEAR THE I.E. BIT
532 000776 104410 000000 002556 BIT #100,@DLRCSR ;DID I.E. BIT GET CLEARED ??
533 001002 042777 000100 002550 BNE 2S ;BR IF YES
534 001016 001415 JSR PC,STATR ;GO SET UP ERROR INFO
535 001020 004767 002342 MOV #25,ERRTYP
536 001024 012767 000023 177054 *****
537 HDRS,REGIN,NULL ;CAN'T CLEAR RCSR INTR. ENAB. BIT
538 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
539 ENDS,REGIN
540 001040 104403 000000 003640 BIC #100,@DLRCSR ;NOW TURN IT ON FOR REAL
541 001046 104410 000000 002506 BIT #4,@DLXCSR ;TURN ON MAINT. MODE
542 001052 052777 000100 002504 CLR RI ;INIT BREAK TIMER
543 001060 052777 000004 002504 CLR R1 ;LOAD THE XMITTR OUTPUT DATA BUFFER
544 001066 005000 000252 002476 MOVB #252,@DLXDBR
545 001070 112777 *****
546 001076 104407 000000 3S: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
547 001102 104407 000000 000000 BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
548 001106 004767 002476 TST INTFLG ;DID RCVR INTR. YET ??
549 BNE 4S ;BR IF YES
550 DEC RI ;COUNT BREAK TIMER
551 001112 001036 BNE 3S ;BR IF NO TIMEOUT
552 001114 005301 JSR PC,STATR ;GO SET UP ERROR INFO
553 001120 004767 002242 CLR @DLRCSR ;CLEAR BOTH CSRS
554 001124 005077 002436 CLR @DLXCSR
555 001130 005077 002436 MOV #25,ERRTYP
556 001134 012767 000023 176744 *****
557 HDRS,REGIN,NULL ;RCVR FAILED TO INTR ON TIME
558 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
559 ENDS,REGIN
560 001150 104403 000000 003640 MOVB @DLRDBR,@WAS ;GET THE RECEIVED DATA
561 001156 104410 000000 002370 BIC #100,@DLRCSR ;TURN OFF I.E.
562 001162 117767 002402 176720 CLR @DLXCSR ;TURN OFF MAINTENANCE MODF
563 001170 042777 000100 002370 CLR @DLRCSR ;SET SOFTWARE INTR. FLAG
564 001176 005077 002402 CDB @DLR,SBADR ;RETURN TO OTHER CPU
565 001182 000000 002370 RTI ;DID INTR SERVICE CLEAR THE RCVR CSR ??
566 001206 000002 TST @DLRCSR ;BR IF IT DID
567 001210 005777 002352 BNE 6S ;GO SET UP ERROR INFO
568 001216 001421 JSR PC,STATR ;CLEAR BOTH CSRS
569 001222 005077 002340 CLR @DLRCSR
570 001226 005077 002340 CLR @DLXCSR
571 001232 012767 000025 176646 MOV #25,ERRTYP
572 *****
573 HDRS,REGIN,NULL ;RCVR INTR SERVICE FAILED TO CLEAR I.E. AND DONE
574 001240 104405 000000 000000

DLBB80.P11 12-OCT-78 12:01
575 *****
576 001246 104403 000000 003640 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
577 001254 104410 000000 002556 ENDS,REGIN
578 001260 122767 000252 176622 6S: CMPB #252,@WAS ;WAS DATA CORRECT ???
579 001268 001421 BREQ MODM ;BR IF YES
580 001274 017767 002272 176602 MOV @DLR,CSRA ;SET UP ERROR INFO
581 001278 017767 002264 176576 MOV @DLRCSR,@ACSR
582 001304 012767 000252 176574 MOV #252,@ASR
583 001310 018767 002252 176564 MOV @DLRDBR,@ASADR
584 001320 016767 002250 176554 *****
585 001326 104404 000000 000000 DATERS,REGIN ;DATA ERROR!!!
586 *****
587 ;THE FOLLOWING GROUP OF BASIC TESTS ASSUME THAT THE MODEM IS DISCONNECTED
588 ;FROM THE DL11-E AND AN R315 MODEM TEST CONNECTOR IS CONNECTED IN ITS
589 ;PLACC. ALL OF THESE TESTS ARE SKIPPED IF SR1(15) IS CLEARED.
590 *****
591 MODEM: TST SR1 ;IS SR1 BIT 15 = 1 ??
592 BMI DLT06 ;DO MODEM TESTS IF SET
593 JMP RSTRT ;SKP MODEM TESTS IF SR1(15)=0
594 *****
595 ;TEST THAT "REQ TO SEND" CAN ASSERT "RING"
596 *****
597 DLT06: BIT #4,@DLRCSR ;SET REQ TO SEND
598 BIT #4,@DLRCSR ;DID IT SPT ??
599 BNE IS ;BR IF YES
600 JSR PC,STATR ;GO SET UP ERROR INFO
601 MOV #25,ERRTYP
602 *****
603 HDRS,REGIN,NULL ;CAN'T SET REQ TO SEND
604 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
605 ENDS,REGIN
606 001374 104405 000000 000000 BIT #40000,@DLRCSR ;DID "RING" GET ASSERTED ??
607 001402 032777 040000 002156 BNE 2S ;BR IF YES
608 001410 001010 JSR PC,STATR ;GO SET UP ERROR INFO
609 001412 004767 001750 MOV #25,ERRTYP
610 001416 012767 000025 176462 *****
611 HDRS,REGIN,NULL ;"REQ TO SEND" DIDN'T ASSERT "RING"
612 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
613 ENDS,REGIN
614 001424 104405 000000 000000 BIC #4,@DLRCSR ;TURN OFF "REQ TO SEND"
615 001432 042777 000004 002126 TST @DLRCSR ;ARE ALL BITS NOW CLEAR ??
616 001440 005777 002122 BRQ DLT07 ;BR IF BOTH "RING" AND "REQ TO SEND" CLEARED
617 001444 001410 JSR PC,STATR ;GO SET UP ERROR INFO
618 001446 004767 001714 MOV #25,ERRTYP
619 001452 012767 000025 176426 *****
620 HDRS,REGIN,NULL ;"RING" OR "REQ TO SEND" FAILED TO CLEAR
621 MSGNS,REGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
622 ENDS,REGIN
623 *****
624 ;TEST THAT "SEC XMIT" ASSERTS "SEC REC" WHICH SETS "DATA SET INT"
625 *****
626 *****
627 *****
628 DLT07: BIT #10,@DLRCSR ;SET SEC XMIT
629 001478 005777 000010 002072 ;DID DATA SET INT GET SET ??
630 001500 100410 BMI IS ;BR IF YES
631 001502 004767 001660 JSR PC,STATR ;GO SET UP ERROR INFO

```

632 001506 012767 000025 176372
633
634 001514 104405 000000 000000
635
636 001522 022777 002010 002036 1S:
637 001530 001410
638 001532 004767 001630
639 001536 012767 000025 176342
640
641 001544 104405 000000 000000
642
643 001552 042777 000010 002006 2S:
644 001560 005777 002002
645 001564 100410
646 001566 004767 001574
647 001572 012767 000020 176306
648
649 001600 104405 000000 000000
650
651 001606 005777 001754 3S:
652 001612 001410
653 001614 004767 001546
654 001620 012767 000020 176260
655
656 001626 104405 000000 000000
657
658
659
660
661
662 001634 005077 001726 DLT10: CLR 0DLRCSR ;CLR THE RCVR CSR
663 001640 052777 000002 001720 BIS #2,0DLRCSR ;SET DATA TERM READY
664 001646 005777 001714 TST 0DLRCSR ;DID DATA SET INT SET ??
665 001652 100407
666 001654 004767 001506 JSR PC,STATR ;GO SET UP ERROR INFO
667 001660 005067 176222 CLR 0ERRTYP
668
669 001664 104405 000000 000000
670
671 001672 022777 030002 001666 1S:
672 001700 004767
673 001702 004767 001460
674 001706 005067 176174 CLR 0ERRTYP
675
676 001712 104405 000000 000000
677
678 001720 042777 000002 001640 2S:
679 001726 005777 001634 TST 0DLRCSR ;TURN OFF DTR
680 001732 100407
681 001734 004767 001426 JSR PC,STATR ;DATA SET INT SHOULD HAVE SET
682 001740 005067 176142 CLR 0ERRTYP
683
684 001744 104405 000000 000000
685
686 001752 005777 001610 3S:
687 001756 001407

```

TEST THAT "DTM" ASSERTS "CAR DET", "CLR TO SEND", AND "DATA SET INT"

```

688 001760 004767 001402 JSR PC,STATR ;GO SET UP ERROR INFO
689 001764 005067 176116 CLR 0ERRTYP
690
691 001770 104405 000000 000000
692
693
694
695
696
697
698 001776 052777 000040 001562 DLT11: BIS #40,0DLRCSR ;SET DATA SET I.E.
699 002000 052777 000040 001554 BIT #0,0DLRCSR ;DID IT SET ??
700 002012 001010
701 002014 004767 001346 JSR PC,STATR ;BR IF YES
702 002020 012767 000027 176060 MOV #2,0ERRTYP ;GO SET UP ERROR INFO
703
704 002026 104405 000000 000000
705
706 002034 042777 000040 001524 1S:
707 002042 005777 001520
708 002046 001411
709 002050 004767 001312 JSR PC,STATR ;CLR DATA SET I.E.
710 002054 012767 000027 176024 MOV #2,0ERRTYP ;DID IT CLEAR ??
711
712 002062 104405 000000 000000
713
714 002070 000463
715
716
717
718 002072 005067 001512 DLT12: CLR INTPLG ;INIT SOFTWARE INTR. FLAG
719 002076 016700 175706 MOV VECTOR,R0 ;GET BAS VECTOR ADDR.
720 002102 012720 002174 MOV #2S,(R0)+ ;GO TO 2S ON DATA SET INTERRUPT
721 002106 005091 CLR R1 ;INIT BREAK TIMER
722 002110 052777 000040 001450 BIS #40,0DLRCSR ;ENABLE DATA SET INTR.
723 002116 052777 000010 001442 BIS #10,0DLRCSR ;SET SEC XMIT
724
725 002124 104407 000000 000000 1S:
726 002130 104407 000000 BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
727 002134 005767 001450 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
728 002140 001023 TST INTPLG ;DID INTR OCCUR YET ??
729 002144 005301
730 002146 001361
731 002148 004767 001214
732 002152 005077 001410
733 002156 012767 000023 175722 MOV #2,0ERRTYP ;BR IF YES
734
735 002164 104405 000000 000000
736
737 002172 000422
738 002202 042777 000040 001364 2S:
739 002206 005167 001402
740 002208 000002
741 002210 032777 000040 001350 3S:
742 002216 001410
743 002220 004767 001142 JSR PC,STATR ;RETURN CONTROL TO OTHER GUY

```

TEST THAT DATA SET INTR CAN CAUSE A RCVR INTRERUPT WHEN ENABLED

```

744 002224* 012767 000023 175654      MOV     #23,ERRTYP
745                                     ;*****
746 002232* 104405 000000* 000000      HDRERS,BEGIN,NULL      ;INTR SERVICE FAILED TO CLR DATA SET I.F.
747                                     ;*****
748                                     ;TEST THAT "BREAK" BIT CAN SET AND CLEAR
749                                     ;-----
750
751 002240* 052777 000001 001324 DLT13: BIS     #1,ADLXCSR      ;SET BREAK BIT
752 002246* 032777 000001 001316      BIT     #1,ADLXCSR      ;DID IT SET ??
753 002254* 001010 001122      BNE     IS              ;BR IF YES
754 002258* 004767 000025 175616      JSR     PC,STATY        ;GO SET UP ERROR INFO
755 002262* 012767 000025 175616      MOV     #25,ERRTYP
756                                     ;*****
757 002270* 104405 000000* 000000      HDRERS,BEGIN,NULL      ;CAN'T SET BREAK BIT
758                                     ;*****
759 002276* 042777 000001 001266      IS:     BIT     #1,ADLXCSR      ;CLEAR THE BREAK BIT
760 002304* 032777 000001 001260      BIT     #1,ADLXCSR      ;DID IT CLEAR ?
761 002312* 001410 001054      BEQ     RPSRT          ;BR IF YES
762 002314* 004767 000025 175560      JSR     PC,STATY        ;GO SET UP ERROR INFO
763 002320* 012767 000025 175560      MOV     #25,ERRTYP
764                                     ;*****
765 002326* 104405 000000* 000000      HDRERS,BEGIN,NULL      ;BREAK BIT WON'T CLEAR
766                                     ;*****
767
  
```

```

768                                     ;
769                                     ; * SECTION TWO *
770                                     ;
771
772 002334* 005067 001242      RESTRT: CLR     XEND          ;CLEAR END FLAGS
773 002340* 005077 001226      CLR     ADLXCSR        ;CLEAR THE DL11 CONTROL RFGS
774 002344* 005077 001216      CLR     ADLXCSR        ;JUST IN CASE
775 002350* 016700 175434      MOV     VECTOR,RO      ;GET START VECTOR ADDRESS
776 002354* 012720 002522*      MOV     BPRINT,(RO)+   ;SET UP THE RCVR AND XMIT VECTORS
777 002360* 116710 175426      MOV     BPI,(RO)
778 002364* 005720 002452*      TST    (RO)+
779 002366* 012720 175414      MOV     BPI,(RO)
780 002372* 116710 175414      MOV     BPI,(RO)
781 002376* 012703 003612*      DOAGIN: MOV    LDTAB,R3   ;POINT TO TABLE OF LOAD SUBR. POINTERS
782 002402* 012704 003622*      MOV    MNTAB,R4        ;POINT TO TABLE OF MESSAGE POINTERS
783 002406* 005067 001174      IS:     CLR    RTRF      ;CLEAR RETRY FLAGS
784 002412* 005777 001152      TST    BDLDRR         ;FLUSH RCVR INPUT BUFFER REG
785 002416* 005777 001146      TST    BDLDRR
786 002422* 012367 001204      MOV    (R3)+,LDOUT     ;SET UP CORRECT LOAD BUF ADDRESS POINTER
787 002426* 012467 001202      MOV    (R4)+,AMESS     ;SET UP MESSAGE POINTER
788 002432* 004767 000436      JSR    PC,SECT        ;GO DO A SECFMT
789 002436* 022703 003622*      CMP    MNTAB,R3       ;DONE ALL FOUR SEGMENTS ??
790 002442* 001361 000000*      BNE    IS             ;BR IF NOT
791
792
793
794
795
796
797
798 002444* 104413 000000*      ENDRS,BEGIN          ;SIGNAL END OF ITERATION.
799 002450* 000752      BR     DOAGIN         ;MONITOR SHALL TEST END OF PASS
800
801
802
803
  
```

```
804 ;THIS ROUTINE SERVICES ALL XMITTR INTRPTS. FOR ALL 256. BYTE XFERS
805 ;
806
807 002452 105777 001114 XINT: TSTR @DLXCSR ;XMIT READY SET ??
808 002456 100403 BMI IS ;BR IF YES
809
810 002460 000004 000000 002516 PIRQS,BEGIN,4S ; QUEUE UP TO CONTINUE AT 4S AND RTI
811
812 002466 022767 004522 001104 1S: CMP #DLBUFI,OPTR ;OUTPUT 256. BYTES YET ??
813 002474 001405 BNE IS ;BR IF YES
814 002476 117777 001076 001070 2S: MOVB @PTR,@DLXDR ;OUTPUT A CHARACTER
815 002504 005267 001070 INC PTR ;POINT TO NEXT CHAR. IN BUFFER
816 002510 3S:
817
818 002510 000004 000000 002610 PIRQS,BEGIN,6S ; QUEUE UP TO CONTINUE AT 6S AND RTI
819
820
821 002516 105767 001061 4S: TSTR XEND+1 ;ANY FATAL RCVR. ERRORS PENDING ??
822 002522 001025 BNE 5S ;BR IF YES - STOP XMITTING
823 002524 016767 001042 175346 MOV DLXCSR,CSRA ;SAVE THE CSR ADDRESS
824 002532 017767 001034 175342 MOV @DLXCSR,ACSR ;SAVE THE CONTENTS OF THE CSR
825 002540 052777 00104 001024 BIC #104,@DLXCSR ;DISABLE XMITTR INTERRUPTS
826 002546 105167 001030 COMB XEND ;SET XMIT END FLAG
827
828 002552 012767 000011 175326 MOV #11,ERRTYP
829 *****
830 002560 104405 000000 000000 HRDRS,BEGIN,NULL ;XMITTER FALSE INTERRUPT - FATAL ERROR
831 *****
832
833 002566 005077 000774 CLR @DLXCSR ;TURN OFF RCVR INTR.
834 002572 104410 ENDS,BEGIN ;
835
836 002576 042777 000100 000766 5S: BIC #100,@DLXCSR ;DISABLE XMITTR. INTERRUPTS
837 002604 104400 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
838
839 002610 105767 000767 6S: TSTR XEND+1 ;ANY FATAL RCVR. ERRORS PENDING ??
840 002614 001370 BNE 5S ;BR IF YES
841 002616 104400 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
842
843 ;THIS ROUTINE SERVICES RECEIVER INTERRUPTS FOR ALL 256. BYTE XFERS
844 ;
845
846 002622 105777 000740 RINT: TSTR @DLRCSR ;RCVR DONE SET ??
847 002626 100403 BMI IS ;BR IF YES
848
849 002630 000004 000000 002702 PIRQS,BEGIN,3S ; QUEUE UP TO CONTINUE AT 3S AND RTI
850
851 002636 005777 000726 1S: TST @DLRDRR ;OVERRUN/PARITY/FRAMING ERRORS ??
852 002642 100003 BPL 2S ;BR IF NONE
853
854 002644 000004 000000 002774 PIRQS,BEGIN,5S ; QUEUE UP TO CONTINUE AT 5S AND RTI
855
856 002652 022767 005122 000716 2S: CMP #RUFEND,IPTR ;INPUT BUFFER FULL ??
857 002660 001405 BNE 7S ;BR IF YES
858 002662 117777 000702 000706 MOVB @DLRDRR,@IPTR ;READ THE DL INPUT BUFFER REG.
859 002670 005267 000702 INC IPTR ;POINT TO NEXT CHAR. POSITION
```

```
860 002674 7S:
861
862 002674 000004 000000 003062 PIRQS,BEGIN,6S ; QUEUE UP TO CONTINUE AT 6S AND RTI
863
864
865 002702 105767 000674 3S: TSTR XEND ;ANY FATAL XMITTR ERROR PENDING
866 002705 001025 BNE 4S ;BR IF YES
867 002710 016767 000652 175162 MOV DLRCSR,CSRA ;SAVE THE RCVR. CSR ADDRESS
868 002716 017767 000644 175156 MOV @DLRCSR,ACSR ;SAVE CONTENTS OF CSR
869 002724 042777 000100 000634 BIC #100,@DLRCSR ;TURN OFF THE RCVR.
870 002732 105167 000645 COMB XEND+1 ;SET FATAL RCVR ERROR FLAG
871
872 002736 012767 000011 175142 MOV #11,ERRTYP
873 *****
874 002744 104405 000000 000000 HRDRS,BEGIN,NULL ;RECEIVER FALSE INTERRUPT - FATAL ERROR
875 *****
876
877 002752 005077 000614 CLR @DLXCSR ;DISABLE XMITTR TOO
878 002756 104410 ENDS,BEGIN ;
879
880 002762 042777 000100 000576 4S: BIC #100,@DLRCSR ;DISABLE RCVR INTERRUPTS
881 002770 104400 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
882
883 002774 105767 000602 5S: TSTR XEND ;ANY FATAL XMITTR ERRORS PENDING ??
884 003000 001370 BNE 4S ;BR IF YES
885 003002 016767 000560 175070 MOV DLRCSR,CSRA ;SAVE CSR ADDRESS
886 003010 017767 000552 175064 MOV @DLRCSR,ACSR ;SAVE CONTENTS OF CSR
887 003016 017767 000546 175060 MOV @DLRDRR,ASTAT ;SAVE THE ERROR FLAGS
888 003024 042777 000100 000534 BIC #100,@DLRCSR ;DISABLE RCVR INTR.
889
890 003032 012767 000017 175046 MOV #17,ERRTYP
891 *****
892 003040 104405 000000 000000 HRDRS,BEGIN,NULL ;OVERRUN - PARITY - FRAMING ERROR
893 *****
894
895 003046 005077 000520 CLR @DLXCSR ;DISABLE XMITTR TOO
896 003049 105267 000510 INCR RTRY ;SET RETRY FLAG
897 003056 104400 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
898
899 003062 105767 000514 6S: TSTR XEND ;ANY FATAL XMITTR ERRORS PENDING ??
900 003066 001335 BNE 4S ;BR IF YES
901 003070 104400 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
902 ;THIS ROUTINE CONTROLS THE EXECUTION OF EACH OF THE FOUR DATA PATTERNS
903
904 003074- 000240 SEGX: NOP ;DO NOTHING FOR NOW
905 003076- 004767 000320 JSR PC,CDDLBP ;GO CLEAR BUFFERS
906 003107- 004777 000529 JSR PC,LDOOUT ;GO SET UP PATTERN
907 003106- 004767 000106 JSR PC,KICKOF ;GO KICK OFF XMITTR AND RCVR
908 003112- 005002 CLR R2 ;INITIALIZE BREAK TIMFR
909
910 003114- 104407 000000- BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.....
911 003120- 104407 000000- BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
912 003120- 005767 000452 TST XEND ;ANY FATAL ERRORS PENDING ??
913 003130- 001400 BEQ 3S ;RR IF NOT
914 003130- 104400 000000- BRITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
915 003136- 105767 000444 TSTB RTRV ;RETRY FLAG SET ??
916 003142- 001411 BPO 4S ;RR IF NOT
917 003144- 105767 000436 CLR RTRV ;CLEAR THIS FLAG
918 003150- 105767 000439 INCB RTRV+1 ;COUNT ONE RETRY
919 003154- 122767 000003 000425 CMPB #3,RTRV+1 ;TRIED THREE TIMES ??
920 003162- 001345 BNE 4S ;RR IF NOT - TRY IT AGAIN
921 003164- 000409 005122- 000402 4S: CMP #RUFEND,IPTR ;REPORT IT AND GO TO NEXT SEGMENT
922 003174- 001406 BEQ 7S ;RECEIVED 256. CHARS. ??
923 003176- 005302 DEC R2 ;RR IF YES
924 003200- 001345 BNE 2S ;DECREMENT BREAK COUNTER
925 003207- 104403 000000- 003634- 5S: MSGNS,BEGIN,AMESS ;ASCII MESSAGE CALL WITH COMMON HEADER
926 003210- 000207 6S: RTS PC ;GO TO NEXT SEGMENT
927 003212- 004767 000034 7S: JSR PC,CHKDAT ;GO COMPARE W/OUT DATA
928 003216- 000207 RTS ;GO TO NEXT SEGMENT
929
930 ;THIS ROUTINE KICKS OFF ALL 256. BYTE TRANSFERS
931
932 KICKOF: MOV #DLBUFO,OPTR ;POINT TO BEGINNING OF OUTPUT BUFFER
933 MOV #DLBUFI,IPTR ;POINT TO BEGINNING OF INPUT BUFFER
934 BIC #104,ADLXCSR ;TURN ON XMITTR
935 BIC #100,ADLRCSR ;TURN ON RCVR
936 RTS PC ;RETURN TO CALLING SEGMENT
937
938 ;THIS ROUTINE CHECKS FOR AND REPORTS DATA COMPARE ERRORS
939
940
941
942
943
944 003252- 042777 000100 000312 CHKDAT: BIC #100,ADLXCSR ;DISABLE XMITTR INTR.
945 003250- 042777 000100 000300 BT #104,ADLXCSR ;DISABLE RCVR INTR.
946 003266- 012700 004122- MOV #DLBUFO,R0 ;R0 POINTS TO OUTPUT BUFFER
947 003272- 012701 004522- MOV #DLBUFI,R1 ;R1 POINTS TO INPUT BUFFER
948 003276- 122021 1S: CMPR (R0),(R1)+ ;COMPARE INPUT WITH OUTPUT
949 003300- 001004 BNE 2S ;RR IF NOT EQUAL
950 003302- 022701 005122- CMP #RUFEND,R1 ;END OF THE BUFFERS ??
951 003306- 001373 BNE 1S ;RR IF NOT
952 003310- 000207 RTS PC ;RETURN TO CALLER
953
954 003312- 016767 000250 174560 2S: MOV DLRCSR,CSRA ;SAVE THE CSR ADDRESS
955 003320- 114067 174562 MOVB -(R0),ASB ;SAVE THE SHOULD BE DATA
956 003324- 042767 177400 174554 BIC #177400,ASB ;ZERO HI BYTE
957 003332- 010067 174544 MOV R0,SBADR ;SAVE THE SHOULD BE ADDRESS
```

```
958 003336- 114167 174546 174540 MOVB -(R1),AWAS ;SAVE THE WAS DATA
959 003342- 042767 177400 BIC #177400,AWAS ;CLEAR OUT HI BYTE
960 003350- 010167 174530 MOV R1,WASADR ;SAVE THE WAS ADDRESS
961
962 003354- 104404 000000- ;*****
963 DATFRS,BEGIN ;DATA ERROR!!!
964 ;*****
965
966 003360- 105720 TSTB (R0)+ ;POINT TO NEXT BYTE IN THE BUFFERS
967 003364- 105746 TSTB (R1)+ ;POINT TO NEXT BYTE IN THE BUFFERS
968 BR 3S ;GO CHECK NEXT BYTE
969
970 ;THESE ROUTINES SET UP THE ERROR INFORMATION FOR THE BASIC TESTS
971
972
973 003366- 016767 000174 174504 STATR: MOV DLRCSR,CSRA ;SAVE THE CSR ADDRESS
974 003374- 017767 000166 174500 MOV #DLRCSR,ACSR ;SAVE THE CONTENTS OF THE CSR
975 003402- 000207 RTS PC ;RETURN TO BASIC TESTS
976
977 003404- 016767 000162 174466 STATX: MOV DLXCSR,CSRA ;SAVE THE CSR ADDRESS
978 003412- 017767 000154 174462 MOV #DLXCSR,ACSR ;SAVE THE CONTENTS OF THE CSR
979 003420- 000207 RTS PC ;RETURN TO THE BASIC TESTS
980
981 ;THIS ROUTINE IS USED TO CLEAR THE INPUT/OUTPUT BUFFERS
982
983
984 003422- 012700 004122- CDDLBP: MOV #DLBUFO,R0 ;POINT R0 TO BEGINNING OF BUFFERS
985 003426- 005020 CLR (R0)+ ;CLEAR ONE WORD - UPDATE POINTER
986 003430- 022700 005122- CMP #RUFEND,R0 ;DONE 256. BYTES ??
987 003434- 001374 BNE 1S ;RR IF NOT
988 003436- 000207 RTS PC ;RETURN TO CALLING SEGMENT
989
990 ;THIS ROUTINE LOADS THE OUTPUT BUFFER WITH A NULL-DEL-NULP PATTERN
991
992
993 003440- 012700 004122- LDOU1: MOV #DLBUFO,R0 ;SET UP POINTER
994 003444- 105020 1S: CLR (R0)+ ;MOV A NULL CHAR
995 003448- 112700 000377 MOVB #37,(R0)+ ;MOV A DELETE CHAR.
996 003452- 012700 004522- CMP #DLBUFI,R0 ;BUFFER FULL ???
997 003456- 001372 BNE 1S ;RR IF NOT
998 003460- 000207 RTS PC ;RETURN TO CALLING SEGMENT
999
1000 ;THIS ROUTINE IS USED TO LOAD AN ASCENDING BINARY COUNT PATTERN
1001
1002
1003
1004 003462- 012700 004122- LDOU2: MOV #DLBUFO,R0 ;SET UP POINTER
1005 003466- 005001 CLR R1 ;DSP R1 TO GENERATE THE PATTERN
1006 003470- 110160 1S: MOVB R1,(R0)+ ;LOAD ONE CHAR.
1007 003474- 012700 004522- CMP #DLBUFI,R0 ;BUFFER FULL ???
1008 003478- 001402 BPO 2S ;RR IF YES
1009 003500- 105201 INCB R1 ;GENERATE NEXT CHAR.
1010 003502- 000772 BR 1S ;GO MOVE IT
1011 003504- 000207 RTS PC ;RETURN TO CALLING SEGMENT
1012
1013 ;THIS ROUTINE IS USED TO LOAD THE DSCENDING BINARY COUNT PATTERN
```

```

1014
1015
1016 003506 012700 004122
1017 003512 012701 000377
1018 003516 110120 004522
1019 003520 022700
1020 003524 011842
1021 003526 105301
1022 003530 000772
1023 003532 000207
1024
1025
1026
1027
1028 003534 012700 004122
1029 003540 012701 005124
1030 003544 012120
1031 003546 022700 004522
1032 003552 011404
1033 003554 022701 005224
1034 003560 001767
1035 003562 000170
1036 003564 000207

;-----
LDOUT3: MOV #DLBUFO,RO ;SET UP POINTER
          MOV #377,R1 ;START R1 AT 377
          MOVB R1,(RO)+ ;LOAD ONE CHAR.
          CMP #DLBUFI,RO ;AT END OF THE BUFFER ??
          BCC YES ;BR IF YES
          DECR R1 ;GENERATE NEXT CHAR.
          BR 1S ;GO MOVE IT
          RTS PC ;RETURN TO CALLING SEGMENT

;THIS ROUTINE LOADS THE WORST CASE PATTERN
;-----
LDOUT4: MOV #DLBUFO,RO ;SET UP POINTERS
          MOV #WCASE,R1 ;POINT TO MONITOR'S WORST CASE PATTERN
          MOVB (R1),(RO)+ ;LOAD ONE WORD
          CMP #DLBUFI,RO ;BUFFER FULL ???
          BCC YES ;BR IF YES
          CMP #WCASE,R1 ;END OF WORST CASE PATTERN ??
          BEQ 3S ;GO RESET R1
          BR 2S ;GO MOVE IT
          RTS PC ;RETURN TO CALLING SEGMENT
  
```

```

1037
1038
1039
1040 003566 000000
1041 003570 000000
1042 003572 000000
1043 003574 000000
1044
1045 003576 000000
1046 003600 000000
1047
1048 003602 000000
1049 003604 000000
1050 003606 000000
1051 003610 000000
1052
1053 003612 003440
1054 003614 003462
1055 003616 003506
1056 003620 003514
1057 003622 003544
1058 003624 003705
1059 003626 003750
1060 003630 004015
1061
1062 003632 000000
1063 003634 003644
1064 003636 177777
1065 003640 004053
1066 003642 177777
1067 003644 047045 046125 026514
1068 003652 042504 026514 052516
1069 003660 046114 051440 050505
1070 003666 042515 041516 020105
1071 003674 041101 051117 042524
1072 003702 022504 000
1073 003705 045 044502 040516
1074 003712 054522 052440 020120
1075 003720 047503 047125 020124
1076 003726 042523 052521 047105
1077 003734 042503 040440 047502
1078 003742 052122 042105 000045
1079 003750 041045 047111 051101
1080 003756 020131 047504 047127
1081 003764 041440 052517 052116
1082 003772 051440 050505 042522
1083 004000 041516 020105 041101
1084 004006 051117 042524 022504
1085 004014 000
1086 004015 045 047527 051522
1087 004022 020124 040503 042523
1088 004030 051440 050505 042525
1089 004036 041516 020105 041101
1090 004044 051117 042524 022504
1091 004052 000
1092 004053 045 040506 040524

;VARIABLES,FLAGS,MESSAGES,AND BUFFERS
;-----
DLRCSR: OPEN ;CONTAINS ADDRESS OF RCVR CSR
DLRDBR: OPEN ;CONTAINS ADDRESS OF RCVR DBR
DLXCSR: OPEN ;CONTAINS ADDRESS OF XMITR CSR
DLXDBR: OPEN ;CONTAINS ADDRESS OF XMITR DBR

IPTR: OPEN ;CONTAINS POINTER TO INPUT BUFFER
OPTR: OPEN ;CONTAINS POINTER TO OUTPUT BUFFER

XEND: OPEN ;FATAL ERROR END FLAGS
EPCR: OPEN ;END OF PASS COUNTER
STRY: OPEN ;RETRY FLAG AND COUNTER
INTPLG: OPEN ;SOFTWARE INTR. FLAG USED BY BASIC TESTS

LDTAB: LDOUT1 ;POINTER TO 1ST LOAD BUFFER SUBR.
        LDOUT2 ;POINTER TO 2ND LOAD BUFFER ROUTINE
        LDOUT3 ;POINTER TO 3RD LOAD BUFFER ROUTINE
        LDOUT4 ;POINTER TO 4TH LOAD BUFFER ROUTINE

MTAB: MSG1 ;POINTER TO MESSAGE 1
      MSG2 ;POINTER TO MESSAGE 2
      MSG3 ;POINTER TO MESSAGE 3
      MSG4 ;POINTER TO MESSAGE 4

LDOUT: OPEN ;CONTAINS POINTER TO LOAD BUFFER SUBR.
AMESS: MSG1 ;MESSAGE POINTERS
      -1 ;TERMINATOR
DRPMS: MSG5 ;MESSAGE POINTER
      -1 ;TERMINATOR

MSG1: .ASCIZ /*NULL-DEL-NUL NULL SEQUENCE ABORTED*/
MSG2: .ASCIZ /*BINARY UP COUNT SEQUENCE ABORTED*/
MSG3: .ASCIZ /*BINARY DOWN COUNT SEQUENCE ABORTED*/
MSG4: .ASCIZ /*WORST CASE SEQUENCE ABORTED*/
MSG5: .ASCIZ /*FATAL ERROR IN STATIC REGISTER TESTS*/
  
```

1093 004060 020114 051105 047522
1094 004066 020122 047111 051440
1095 004074 040524 044524 020103
1096 004102 042522 044567 052103
1097 004110 051105 052040 051505
1098 004116 051524 000045
1099
1100
1101
1102
1103
1104 004122 000400
1105 004522 000400
1106 005122 000000
1107
1108
1109 005124 177776 000001 177775
1110 005132 000002 177773 000004
1111 005140 177767 000019
1112 005144 177757 000020 177737
1113 005152 000030 177677 000100
1114 005166 177577 000200
1115 005164 177377 000200
1116 005172 001000 175777 002000
1117 005200 173777 004000
1118 005204 167777 010000
1119 005220 077777 040000
1120 005226 077777 100000
1121 005224
1122
1123 000001

- EVEN

;512 WORDS RESERVED FOR TWO 256. BYTE BUFFERS

DLBUFD: -BLKB 256- ;RSVD FOR OUTPUT BUFFER
DLBUFI: -BLKB 256- ;RSVD FOR INPUT BUFFER
BUFEND: 0 ;MARK END OF BUFFER AREA

WCASE: .WORD 177776,1,177775,2,177773,4,177767,10

.WORD 177757,20,177737,40,177677,100,177577,200

.WORD 177377,400,176777,1000,175777,2000,173777,4000

.WORD 167777,10000,157777,20000,137777,40000,77777,100000

WCASEE:

- END

ACSR 000102R 399# 582*
ADDR 000006R 355# 409
ADDR22= 001090 407#
AMSS 03634R 78#
ASH 000106R 393# 927 1063#
ASTAT 000104R 391# 887* 955*
AWAS 000108R 324# 579 958* 959*
BEGIN 000000R 490 439 440 450 452 453 464 466 467 474 476 477
491 499 501 502 511 513 514 529 531 532 539 541
542 548 549 559 561 562 575 577 578 587 607 614 622
514 641 649 656 669 676 684 691 703 711 725 726 735
746 758 766 798 810 818 830 834 837 841 849 854 862
874 881 892 897 901 910 911 914 927 963
BIT0 = 000001 407#
BIT1 = 000002 407#
BIT10 = 020000 407#
BIT11 = 004000 407#
BIT12 = 010000 407#
BIT13 = 020000 407#
BIT14 = 040000 407#
BIT15 = 100000 407#
BIT2 = 000004 407#
BIT3 = 000010 407#
BIT4 = 000020 407#
BIT5 = 000040 407#
BIT6 = 000100 407#
BIT7 = 000200 407#
BIT8 = 000400 407#
BIT9 = 001000 407#
BREAKS= 104407 407# 490 491 548 549 725 726 910 911
BR1 000012R 486 522 778 781
BR3 000013R 355#
BTODS = 104421 407#
BUFEND 005122R 856#
CDATAS= 104412 407#
CWDAT 003252R 929 944#
CLDLBP 003442R 905 984#
CONFIG 000056R 377#
CSRA 000100R 387# 581* 823* 867* 885* 954* 973* 977*
DATCKS= 104411 407#
DATERS= 104404 407# 587 963
DLBUFI 004522R 812 936 947 996 1007 1019 1031 1105#
DLBUFD 004122R 935 984 993 1004 1016 1028 1104#
DLWIT 00056R 423#
DLRCSR 003566R 410* 423* 424* 432 523* 524 533* 534 543* 555* 564* 568 571*
581 582 601* 602 609 616* 617 628* 629 636 643* 644 651*
665* 664 671 678* 679 686 697* 698 705* 706 722* 723*
738* 741 775* 833* 846 867 868 869* 880* 885 886
938* 945* 954 973 974 1040#
412* 426 427 563 584 785 786 851 858 887 1041#
DLRDBR 003570R
DLT01 000302R 433#
DLT02 000342R 446#
DLT03 000404R 446#
DLT04 000524R 470#
DLT05 000712R 507#
DLT06 001344R 595# 601#

824* 868* 896* 974* 978*
1063# 955*
579 958* 959*
450 452 453 464 466 467 474 476 477
501 502 511 513 514 529 531 532 539 541
548 549 559 561 562 575 577 578 587 607 614 622
641 649 656 669 676 684 691 703 711 725 726 735
746 758 766 798 810 818 830 834 837 841 849 854 862
874 881 892 897 901 910 911 914 927 963
490 491 548 549 725 726 910 911
486 522 778 781
856#
929 944#
905 984#
387# 581* 823* 867* 885* 954* 973* 977*
407# 587 963
812 936 947 996 1007 1019 1031 1105#
935 984 993 1004 1016 1028 1104#
423#
410* 423* 424* 432 523* 524 533* 534 543* 555* 564* 568 571*
581 582 601* 602 609 616* 617 628* 629 636 643* 644 651*
665* 664 671 678* 679 686 697* 698 705* 706 722* 723*
738* 741 775* 833* 846 867 868 869* 880* 885 886
938* 945* 954 973 974 1040#
412* 426 427 563 584 785 786 851 858 887 1041#

DLBR DEC/X11 SYSTEM EXERCISER MODULE
XDLBBO.P11 12-OCT-78 12:01

MACY11 30A(1052) 12-OCT-78 16:28 PAGE 30
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0028

WASADR	000104R	390#	584*	960*								
WCASE	005124R	1029	1109#									
WCASE	005224R	1033	1121#									
WDFR	000116R	397#										
WDT0	000114R	396#										
XEND	003602R	773*	821	826*	839	865	870*	883	899	912	1048#	
XFLAG	000005R	354#										
XINT	002452R	780	807#									
.	= 005224R	1104#	1105#									

. ABS. 000000 000
005224 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XDLBRO,XDLBBO/SOL/CRF:SYN=DDKCOM,XDLBBO
RUN-TIME: 2 3 3 SECONDS
RUN-TIME RATIO: 31/6=4.9
CORE USED: 7K (13 PAGES)

DIAGNOSTIC ENGINEERING

digital

DECO DEPO SUBMISSION

FOR RELEASE ENG. USE
 NEW CHANGE DELETE

PRODUCT IDENTIFICATION

MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE	STATUS	DISTRIBUTION	1ST COPY - RIGHT YEAR	LAST COPY - RIGHT YEAR
	ZZ	CXDLB	B	1	01	22 Jan 79	OBSOLETE	X G R	1976	1979
TITLE CXDLBB1 DL11-E MODULE										
AUTHOR D. RUTENHOF			MAINTAINER D. RUTENHOF			SPT GRP		MAINTAINER D. RUTENHOF		SUBMITTING ENGINEER RUTENHOF

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X	DEPO	AF-E998B-M1					

PRODUCTS OBSOLETE (other than previous version)

LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV
MD			MD			MD		

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)
 03 04 05 10 20 21 34 35 40 45 50 55 60 70

OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)
 02 03 04 06 50

ACT/APT/XXDP	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
INFORMATION FIELD			<input checked="" type="checkbox"/> Y <input type="checkbox"/> N	<input checked="" type="checkbox"/> Y <input type="checkbox"/> N	40 SECONDS	70 SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED: _____

DEVICE AFFECTED DEC/X11 MULTIMEDIA AFFECTED? YES NO

KIT NUMBERS	ZJ129-RZ	ZJ129-FR	ZJ130-RB
-------------	----------	----------	----------

PROBLEM:
 Module is intended for 2/40 FRONT-END interface only, but documentation does not say so.

SOLUTION:
 State in module header that this module is not intended for use on standalone PDP11 SYSTEMS.

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO

SUBMITTING ENGINEER <i>D. Rutenhof</i>	MANUFACTURING ENGINEER <i>Renton to Nancy</i>	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER Q99-05460
DATE: 23-Jan-79	DATE: 7-Feb-79	DATE:	
MAINTAINER <i>D. Rutenhof</i>	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MCF# 2838
DATE: 23-Jan-79	DATE:	DATE:	