

IDENTIFICATION  
-----

PRODUCT CODE: AC-E988-MC  
PRODUCT NAME: CXDL880 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 IOWD THAT EXERCISES ONE D11-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 EQUIPPED TO PERFORM INPUT/OUTPUT DATA TRANSFERS IN INTERRUPT MODE. ERRORS DETECTED IN THIS SECTION 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-NULL SEQUENCE (000,377,000,000,000,377)
- B. BINARY UP-COUNT SEQUENCE (000,001,002,003,004,005)
- C. BINARY DOWN COUNT SEQUENCE (377,376,375,374,373,372)
- 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 CPO TYPE.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 175610 VCT: 300 BR1: 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 DLT1-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 RCSPR ARE CLEAR WHEN  
THE MODULE IS INITIALIZED TO RUN.
- DLT02: TEST THAT ONLY THE "RFADV" 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 RCSPR 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 SRI. THE MODULE LOOKS AT SRI 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 INTR" AND THAT "CAR DET" WHICH SETS "DATA SET INTR" ALSO TESTS THAT CLEARING "SEC XMIT" NEGATES "SEC REC" WHICH ALSO CAUSES "DATA SET INTR" TO SET.

DLT10: TEST THAT "DTR" ASSERTS "CLR TO SEND" AND "CAR DET" WHICH IN TURN SETS "DATA SET INTR" 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 "BREAK" CALLS TO THE MONITOR TO PREVENT TIMEOUTS FROM HANGING THE MODULE.
5. CALLS THE SUBROUTINE TO CHECK THE DATA BUFFERS WHEN THE BLOCK TRANSFER IS COMPLETE.

KICKOF: THIS SUBROUTINE IS CALLED FROM "SEGX" AND CONTAINS 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.

LDOU1: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH THE NULL-DEL-NULL PATTERN.

LDOU2: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH A BINARY UP-COUNT PATTERN.

LDOU3: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH A BINARY DOWN-COUNT PATTERN.

LDOU4: 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 "DLR 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 TRANSFERRED 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 SUCCESSFULLY COMPLETED DUE TO SOFT ERRORS (3 RETRIES ATTEMPTED) OR A MONITOR "BREAK" TIMEOUT ONE OF THE FOLLOWING APPROPRIATE PRINTOUTS WILL OCCUR:

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

DL88 DEC/X11 SYSTEM EXERCISER MODULE MACY11 30A(1052) 12-OCT-78 16:28 PAGE 8  
XDLRBO.P11 12-OCT-78 12:01

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.

SEQ 0007

```
DL11-E DEC/X11 EXERCISER MODULE
000000- [OMOD <DLBB > 175610,300,4,1024,161
000000- MODULE 140000,DLBB,175610,300,4,1024,161
, -TITLE DLBB DEC/X11 SYSTEM EXERCISER MODULE
DDKCOM VERSION 6 23-MAY-78
- LIST BIN
*****
000000- 046104 041102 040 BGIN: *****
000000- 000005- 000 XFLAG: BYTE OPEN ;MODULE NAME.
000006- 175610 ADDR: 175610+0 ;USED TO KEEP TRACK OF WBUFF USAGE
000010- 000300 VECTOR: 300+0 ;1ST DEVICE ADDR.
000012- 200 BR1: -BYTE PRTV4+0 ;1ST DEVICE VECTOR.
000013- 000 BR2: -BYTE PRTV+0 ;1ST RR LEVEL.
000014- 000001 DVID1: +1 ;DEVICE INDICATOR 1.
000016- 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: MODSP ;MODULE STACK POINTER.
000034- 000000 PASCNT: 0 ;PASS COUNTER.
000036- 002000 ICOUNT: 1024. ;LOC TO COUNT ITERATIONS
000040- 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS PER PASS=1024.
000042- 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000044- 000000 SOFPAS: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000046- 000000 HRDPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS PER PASS
000050- 000000 SYSCNT: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000052- 000000 RANNUM: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000054- 000000 CONFIG: 0 ;# OF SYS ERRORS ACCUMULATED
000056- 000000 RES1: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000060- 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000062- 000000 SVR0: OPEN ;RESERVED FOR MONITOR USE
000064- 000000 SVR1: OPEN ;LOC TO SAVE R0.
000066- 000000 SVR2: OPEN ;LOC TO SAVE R1.
000070- 000000 SVR3: OPEN ;LOC TO SAVE R2.
000072- 000000 SVR4: OPEN ;LOC TO SAVE R3.
000074- 000000 SVR5: OPEN ;LOC TO SAVE R4.
000076- 000000 SVR6: OPEN ;LOC TO SAVE R5.
00100- 000000 CSRA: OPEN ;LOC TO SAVE R6.
00102- 000000 SBADR: OPEN ;ADDR OF CURRENT CSR.
00104- 000000 ACSR: OPEN ;ADDR OF GOOD DATA, OR
00106- 000000 WSAADR: OPEN ;CONTENTS OF CSR.
00110- 000000 ASTAT: OPEN ;ADDR OF BAD DATA, OR
00112- 002334- ERRTVP: OPEN ;STATUS REG CONTENTS.
00114- 000000 ASB: OPEN ;TYPE OF ERROR.
00116- 000000 AWAS: OPEN ;EXPECTED DATA.
00118- 000000 RSTRT: RSTRT ;ACTUAL DATA.
00120- 000000 WDTP: OPEN ;RESTART ADDRESS AFTER END OF PASS
00122- 000161 IDNUM: 161 ;WORDS TO MEMORY PER ITERATION
;WORDS FROM MEMORY PER ITERATION
;# OF INTERRUPTS PER ITERATION
;MODULE IDENTIFICATION NUMBER=161
```

```
000040 .REPT SPSIZ ;MODULE STACK STARTS HERE.
-MLIST 0
-WORD
- LIST
-ENDR
MODSP:
*****
407
```



```
408 000224 016700 177556 START: MOV ADDR,RO ;GET BASE DEVICE ADDRESS
409 000230 010067 003332 MOV RO,DLRCSR ;SET UP RCVR CSR ADDRESS
410 000230 010067 003326 TST (R0) ;
411 000236 010067 003326 MOV RO,DLRDBR ;SET UP RCVR DBR ADDRESS
412 000244 005720 003322 TST (R0)+ ;
413 000244 010067 003322 MOV RO,DLXCSR ;SET UP XMITTR CSR ADDRESS
414 000250 005720 003316 TST (R0)+ ;
415 000252 010067 003316 MOV RO,DLXDBR ;SET UP XMITTR DBR ADDRESS
416
417
418
419
420
421
422
423
424 000256 005077 003304 DLINIT: CLR ADLRCSR ;CLEAR OUT BOTH CSR'S
425 000262 005077 003300 CLR ADLRCSR ;MAKE SURE ALL DATA COMM BITS CLEARED
426 000266 005077 003300 CLR ADLXCSR ;
427 000272 005777 003276 TST ADLRDBR ;FLUSH RCVR DONE BIT
428
429
430
431
432
433
434 000302 005777 003260 DLT01: TST ADLRCSR ;IS RCVR CSR ALL ZEROES ??
435 000306 001415 000052 BEQ DLT02 ;BR IF YES
436 000314 004787 000025 JSR PC,STATX ;GO SET UP ERROR INFO
437 000322 104405 000000 000000 MOV #25,ERRTYP ;*****
438 000330 104403 000000 003640 HDRRS,BEGIN,NULL ;CANT'T CLEAR OUT RCVR CSR
439 000336 104410 000000 003640 MSGNS,BEGIN,DRPMS ;*****
440 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
441
442
443
444
445 000342 022777 000200 003222 DLT02: CMP #200,ADLXCSR ;READY SET ??
446 000350 001415 000025 BEQ DLT03 ;BR IF YES
447 000356 012767 000025 JSR PC,STATX ;GO SET UP ERROR INFO
448 000364 104405 000000 000000 MOV #25,ERRTYP ;*****
449 000372 104403 000000 003640 HDRRS,BEGIN,NULL ;READY NOT SET OR OTHER BITS DIDN'T CLEAR IN XMIT CSR
450 000378 104410 000000 003640 MSGNS,BEGIN,DRPMS ;*****
451 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
452
453
454
455
456
457
458 000404 052777 000004 003160 DLT03: BTS #4,ADLXCSR ;SET THE MAINT. BIT
459 000420 001415 000204 003152 CMP #4,ADLXCSR ;DID IT SET ??
460 000426 004787 000025 BEQ DLT04 ;BR IF YES
461 000432 004787 002756 JSR PC,STATX ;GO SET UP ERROR INFO
462 000438 012767 000033 177452 MOV #33,ERRTYP ;*****
463
```

```
464 000434 104405 000000 000000 HDRRS,BEGIN,NULL ;MAINT BIT WON'T SET OR IT CLEARED READY
465 000442 104403 000000 003640 MSGNS,BEGIN,DRPMS ;*****
466 000450 104410 000000 003640 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
467 000454 042777 000004 003110 1S: BIC #4,ADLXCSR ;NOW CLR THE MAINT BIT
468 000462 022777 000200 003102 CMP #200,ADLXCSR ;DID IT CLEAR ??
469 000470 001415 000000 003102 BEQ DLT04 ;BR IF YES
470 000472 004787 002706 JSR PC,STATX ;GO SET UP ERROR INFO
471 000478 012767 000025 177402 MOV #25,ERRTYP ;*****
472 000504 104405 000000 000000 HDRRS,BEGIN,NULL ;MAINT BIT WON'T CLEAR OR READY CLEARED
473 000512 104403 000000 003640 MSGNS,BEGIN,DRPMS ;*****
474 000520 104410 000000 003640 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
475
476
477
478
479
480
481
482 000524 005067 003060 DLT04: CLR INTPLG ;INIT THE SOFTWARE INTR. FLAG
483 000530 016700 177254 MOV VECTOR,RO ;GET BASE VECTOR ADDRESS
484 000534 062700 000004 ADD #4,RO ;GENERATE ADDR OF XMIT VECTOR
485 000540 172720 000634 MOV #25,(R0)+ ;GO TO 25 ON XMIT INTERRUPT
486 000544 116710 177242 MOV BR1,(R0) ;PRIORITY LEVEL = BR1
487 000550 005001 000000 CLR R1 ;INIT BREAK TIMER
488 000552 052777 000100 003012 BIS #100,ADLXCSR ;SET INTR. ENAB
489
490
491 1S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
492 000560 104407 000000 000000 BRS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
493 000570 005767 003014 TST INTPLG ;DID XMIT INTR OCCUR YET ??
494 000574 01025 000000 BNE 35 ;BR IF IT DID
495 000576 005367 000000 DEC R1 ;COUNT BREAK TIMER
496 000600 001367 000000 BNE 15 ;BR IF NO TIMEOUT
497 000602 004787 002576 JSR PC,STATX ;GO SET UP ERROR INFO
498 000606 012767 000023 177272 MOV #23,ERRTYP ;*****
499 000614 104405 000000 000000 HDRRS,BEGIN,NULL ;XMITTR FAILED TO GENERATE INTERRUPT
500 000622 104403 000000 003640 MSGNS,BEGIN,DRPMS ;*****
501 000630 104410 000000 003640 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
502 000634 042777 000100 002730 2S: BIC #100,ADLXCSR ;DISABLE XMITTR INTR ENABLE
503 000642 005167 000242 COM INTPLG ;SET THE INTR. FLAG
504 000646 000002 000000 RTD ;RETURN CONTROL TO OTHER GUY
505 000650 022777 000200 002714 3S: CMP #200,ADLXCSR ;DID I.E. GET CLEARED IN INTR. SERVICE
506 000656 001415 000000 002520 BEQ DLT05 ;BR IF YES
507 000660 004787 002520 JSR PC,STATX ;GO SET UP ERROR INFO
508 000664 012767 000027 177214 MOV #27,ERRTYP ;*****
509 000672 104405 000000 000000 HDRRS,BEGIN,NULL ;"DONE" OR RCVR INTR. ENAB FAILED TO CLEAR
510 000680 104403 000000 003640 MSGNS,BEGIN,DRPMS ;*****
511 000686 104410 000000 003640 ENDS,BEGIN ;ASCII MESSAGE CALL WITH COMMON HEADER
512
513
514
515
516
517
518
519 000712 005067 002672 DLT05: CLR INTPLG ;INIT SOFTWARE INTR. FLAG
```

```
520 000715 016700 177066- MOV VECTOR,R0 ;GET THE BASE VECTOR ADDRESS
521 000720 012720 001162- MOV #45,(R0) ;GO TO 45 ON RCVR INTERRUPT
522 000725 116710 177060 MOVB BR1,(R0) ;SET PRIORITY
523 000730 052777 000100 002626 BIS #100,ADLRCR ;SET I.E. IN RCVR CSR
524 000740 032777 000100 002620 BIT #100,ADLRCR ;DID IT SET
525 000745 001015 ;BNE IS ;BRR IF IT DID
526 000750 004767 002412 JSR PC,STATR ;GO SET UP ERROR INFO
527 000754 012767 000033 177124 MOV #25,ERRTYP
528 ;*****
529 000762 104405 000000 000000 HDRERS,BEGIN,NULL ;CAN'T SET BIT 06 IN RCVR I.E.
530 000770 104403 000000 003640 MSGNS,BEGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
531 000775 104410 000000 ENDS,BEGIN ;
532 001002 042777 000100 002556 1S: BIC #100,ADLRCR ;NOW CLEAR THE I.E. BIT
533 001010 032777 000100 002556 BEO #100,ADLRCR ;DID I.E. BIT GET CLEARED ??
534 001016 001415 ;BRR IF YES
535 001020 004767 002342 JSR PC,STATR ;GO SET UP ERROR INFO
536 001024 012767 000023 177054 MOV #23,ERRTYP
537 ;*****
538 001032 104405 000000 000000 HDRERS,BEGIN,NULL ;CAN'T CLEAR RCVR INTR. ENAB. BIT
539 001040 104403 000000 003640 MSGNS,BEGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
540 001046 104410 000000 ENDS,BEGIN ;
541 001052 052777 000100 002506 2S: BIS #100,ADLRCR ;NOW TURN IT ON FOR REAL
542 001060 052777 000004 002504 BIS #4,ADLRCR ;TURN ON MAINT. MODE
543 001066 004767 000023 002476 CLR RI ;INIT BREAK TIMER
544 001070 112777 000252 002476 MOVB #252,ADLXDBR ;LOAD THE XMITTR OUTPUT DATA BUFFER
545 001076 ;
546 001076 104407 000000 000000 3S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR.
547 001076 104407 000000 000000 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
548 001106 005767 002476 TST INTPLG ;DID RCVR INTR. YET ??
549 001112 001036 ;BNE AS ;BRR IF YES
550 001114 005301 DEC RI ;COUNT BREAK TIMER
551 001116 004767 ;BRR IF NO TIMEOUT
552 001120 004767 002242 JSR PC,STATR ;GO SET UP ERROR INFO
553 001124 005077 002436 CLR ADLRCR ;CLEAR BOTH CSRS
554 001130 005077 002436 CLR ADLXCSR
555 001134 012767 000023 176744 MOV #23,ERRTYP
556 ;*****
557 001142 104405 000000 000000 HDRERS,BEGIN,NULL ;RCVR FAILED TO INTR ON TIME
558 001150 104403 000000 003640 MSGNS,BEGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
559 001156 104410 000000 ENDS,BEGIN ;
560 001162 117767 002402 176720 4S: MOVB ADLXDBR,AWAS ;GET THE RECEIVED DATA
561 001170 042777 000100 002370 BIC #100,ADLRCR ;TURN OFF I.E.
562 001176 005077 000100 002370 CLR ADLXCSR ;TURN OFF MAINTENANCE MODF
563 001202 005167 002402 CDM INTPLG ;SET SOFTWARE INTR. FLAG
564 001206 000002 ;RTI ;RETURN TO OTHER CPU
565 001210 005777 002352 TST ADLRCR ;DID INTR SERVICE CLEAR THE RCVR CSR ??
566 001214 004767 ;BRR IF YES
567 001216 004767 002144 JSR PC,STATR ;GO SET UP ERROR INFO
568 001222 005077 002340 CLR ADLRCR ;CLEAR BOTH CSRS
569 001226 005077 002340 CLR ADLXCSR
570 001232 012767 000025 176646 MOV #25,ERRTYP
571 ;*****
572 001240 104405 000000 000000 HDRERS,BEGIN,NULL ;RCVR INTR SERVICE FAILED TO CLEAR I.E. AND DONE
```

```
575 ;*****
576 001246 104403 000000 003640 MSGNS,BEGIN,DRPMS ;ASCII MESSAGE CALL WITH COMMON HEADER
577 001254 104410 000000 ENDS,BEGIN ;
578 001260 122767 000252 176622 6S: CMPR #252,AWAS ;WAS DATA CORRECT ???
579 001260 104410 000000 000000 BRQ MODRM ;BRR IF YES
580 001260 016767 002272 176602 MOV DLRCSR,CSRA ;SET UP ERROR INFO
581 001270 017767 002264 176576 MOV ADLRCR,ACSR
582 001304 012767 000252 176574 MOV #252,ASR
583 001320 016767 002256 176554 MOV DLRDBR,WASADR
584 ;*****
585 001326 104404 000000 DATERS,BEGIN ;DATA ERROR!!!
586 ;*****
587 ;THE FOLLOWING GROUP OF BASIC TESTS ASSUME THAT THE MODEM IS DISCONNECTED
588 ;FROM THE DL11-E AND AN #315 MODEM TEST CONNECTOR IS CONNECTED IN ITS
589 ;PLACE. 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 ;SKPT MODEM TESTS IF SR1(15)=0
594
595 ;TEST THAT "REQ TO SEND" CAN ASSERT "RING"
596
597 DLT06: BIS #4,ADLRCR ;SET REQ TO SEND
598 BIT #4,ADLRCR ;DID IT SPT ??
599 BNE IS ;BRR IF YES
600 JSR PC,STATR ;GO SET UP ERROR INFO
601 MOV #25,ERRTYP
602 ;*****
603 001374 104405 000000 000000 HDRERS,BEGIN,NULL ;CAN'T SET REQ TO SEND
604 001402 032777 040000 002156 1S: BIT #4000,ADLRCR ;DID "RING" GET ASSERTED ??
605 001410 001010 ;BNE 2S ;BRR IF YES
606 001412 004767 001750 JSR PC,STATR ;GO SET UP ERROR INFO
607 001416 012767 000025 176462 MOV #25,ERRTYP
608 ;*****
609 001424 104405 000000 000000 HDRERS,BEGIN,NULL ;"REQ TO SEND" DIDN'T ASSERT "RING"
610 001432 042777 000004 002126 2S: BIC #4,ADLRCR ;TURN OFF "REQ TO SEND"
611 001440 005777 002122 TST ADLRCR ;ARE ALL BITS NOW CLEAR ??
612 001444 001410 BRQ DLT07 ;BRR IF BOTH "RING" AND "REQ TO SEND" CLEARED
613 001446 004767 001714 JSR PC,STATR ;GO SET UP ERROR INFO
614 001452 012767 000025 176426 MOV #25,ERRTYP
615 ;*****
616 001460 104405 000000 000000 HDRERS,BEGIN,NULL ;"RING" OR "REQ TO SEND" FAILED TO CLEAR
617 ;*****
618 ;TEST THAT "SEC XMIT" ASSERTS "SEC REC" WHICH SETS "DATA SET INT"
619
620 DLT07: BIS #10,ADLRCR ;SET SEC XMIT
621 TST ADLRCR ;DID DATA SET INT GET SET ??
622 BMI IS ;BRR IF YES
623 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 001614* 004767 001546
653 001620* 012767 000020 176260
654
655 001626* 104405 000000* 000000
656
657
658
659
660
661
662 001634* 005077 001726 DLT10: CLR QDLRCSR ;CLR THE RCVR CSR
663 001640* 052777 000002 001720 BIS #2, QDLRCSR ;SET DATA TERM READY
664 001652* 100407 001714 TST QDLRCSR ;DID DATA SET INT SET ??
665 001654* 004767 001506 JSR PC, STATR ;GO SET UP ERROR INFO
666 001660* 005067 176222 CLR ERRTYP
667
668 001664* 104405 000000* 000000
669
670
671 001672* 022777 030002 001666 1S:
672 001702* 004767 001460 CMP #30002, QDLRCSR; DTR CLR TO SEND, CAR DET, SET AND DATA SET INT CLEAR ?
673 001706* 005067 176174 BIC #1, QDLRCSR ;BR IF YES
674
675 001712* 104405 000000* 000000
676
677
678 001720* 042777 000002 001640 2S:
679 001726* 005777 001634 TST QDLRCSR ;TURN OFF DTR
680 001732* 100407 001426 BIC #1, QDLRCSR ;DATA SET INT SHOULD HAVE SET
681 001734* 004767 001426 JSR PC, STATR ;BR IF IT DID
682 001740* 005067 176142 CLR ERRTYP ;GO SET UP ERROR INFO
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 ERRTYP
690
691 001770* 104405 000000* 000000
692
693
694
695
696
697
698 001776* 052777 000040 001562 DLT11: BIS #40, QDLRCSR ;SET DATA SET I.E.
699 002002* 032777 000040 001554 BIT #5, QDLRCSR ;DID IT SET ??
700 002012* 001010 BNE #5 ;BR IF YES
701 002014* 004767 001346 JSR PC, STATR ;GO SET UP ERROR INFO
702 002020* 012767 000027 176060 MOV #21, ERRTYP
703
704 002026* 104405 000000* 000000
705
706
707 002034* 042777 000040 001524 1S:
708 002042* 005777 001520 BIC #40, QDLRCSR ;CLEAR DATA SET I.E.
709 002046* 001411 BIC #1, QDLRCSR ;DID IT CLEAR ??
710 002050* 004767 001312 JSR PC, STATR ;GO SET UP ERROR INFO
711 002054* 012767 000027 176024 MOV #21, ERRTYP
712
713 002062* 104405 000000* 000000
714
715
716
717
718 002070* 000463
719
720
721
722
723
724
725 002072* 005067 001512 DLT12: CLR INTPLG ;INIT SOFTWARE INTR. FLAG
726 002076* 016700 175706 MOV VECTOR, R0 ;GET RAS VECTOR ADDR.
727 002102* 012720 002174 MOV #2S, (R0)+ ;GO TO 2S ON DATA SET INTERRUPT
728 002106* 050901 CLR #1 ;INIT BREAK TIMER
729 002110* 052777 000040 001450 BIS #40, QDLRCSR ;ENABLE DATA SET INTR.
730 002116* 052777 000010 001442 BIS #10, QDLRCSR ;SET SEC XMIT
731
732 002124* 104407 000000* 000000
733 002130* 104407 000000* 000000
734 002134* 005767 001450 1S:
735 002140* 001023 BREAKS, BEGIN ;TEMPORARY RETURN TO MONITOR....
736 002142* 005391 TST INTPLG ;THEN CONTINUE AT NEXT INSTRUCTION.
737 002144* 001361 BNE #3S ;DID INTR OCCUR YET ??
738 002146* 004767 001214 ;BR IF YES
739 002152* 005077 001410 DEC #1 ;COUNT THE BREAK TIMER
740 002156* 012767 000023 175722 BNE #1S ;BR IF NO TIMEOUT
741
742 002164* 104405 000000* 000000
743
744
745
746
747
748
749 002172* 000422 DLT13: BR DLT13 ;GO TO NEXT TEST
750 002174* 042777 000040 001364 2S: BIC #40, QDLRCSR ;TURN OFF DATA SET I.E.
751 002202* 005167 001402 COM INTPLG ;SET SOFTWARE INTR. FLAG
752 002206* 000002 000040 001350 3S: RTI ;RETURN CONTROL TO OTHER GUY
753 002210* 031410 BIT #40, QDLRCSR ;DID INTR SERVICE TURN OFF I.E. ?
754 002220* 004767 001142 BIC #1, QDLRCSR ;BR IF YES
755 JSR PC, STATR ;GO SET UP ERROR INFO

```

```

744 002224 012767 000023 175654      MOV      #23,ERRTYP
745                                     ;*****
746 002232 104405 000000 000000      HRDRS,BEGIN,NULL      ;INTR SERVICE FAILED TO CLR DATA SET I.F.
747                                     ;*****
748                                     ;TEST THAT "BREAK" BIT CAN SET AND CLEAR
749                                     ;-----
750
751
752 002240 052777 000001 001324      DLT13:  BIS      #1,ADLXCSR      ;SET BREAK BIT
753 002246 032777 000001 001316      BIT      #1,ADLXCSR      ;DID IT SET ??
754 002254 001010 000001 001316      BNE      PC              ;BR IF YES
755 002256 004787 001122              JSR      PC,ERRTYP      ;GO SET UP ERROR INFO
756 002262 012767 000025 175616      MOV      #25,ERRTYP
757                                     ;*****
758 002270 104405 000000 000000      HRDRS,BEGIN,NULL      ;CAN'T SET BREAK BIT
759                                     ;*****
760 002276 042777 000001 001266      IS:     BIC      #1,ADLXCSR      ;CLEAR THE BREAK BIT
761 002304 032777 000001 001260      BIT      #1,ADLXCSR      ;DID IT CLEAR ?
762 002312 001410 000001 001260      BNE      PC              ;BR IF YES
763 002314 004787 001064              JSR      PC,ERRTYP      ;GO SET UP ERROR INFO
764 002320 012767 000025 175560      MOV      #25,ERRTYP
765                                     ;*****
766 002326 104405 000000 000000      HRDRS,BEGIN,NULL      ;BREAK BIT WON'T CLEAR
767                                     ;*****
  
```

```

768                                     ;
769                                     ;
770                                     ;
771                                     ;
772 002334 005067 001242      RESTRT: CLR      XEND      ;CLEAR END FLAGS
773 002340 005077 001226      CLR      ADLXCSR      ;CLEAR THE DLI1 CONTROL RFGS
774 002344 005077 001216      CLR      ADLXCSR      ;JUST IN CASE
775 002350 016700 175434      MOV      VECTOR,R0     ;GET START VECTOR ADDRESS
776 002354 012720 002622      MOV      #RINT,R0)+    ;SET UP THE RCVR AND XMIT VFACTORS
777 002360 116710 175426      MOV      BR1,(R0)
778 002364 005720              TST      (R0)+
779 002366 012720 002452      MOV      #XINT,(R0)+
780 002372 116710 175414      MOV      BR1,(R0)
781 002376 012703 003612      DDACIN: MOV      #LDTAB,R3 ;POINT TO TABLE OF LOAD SUBR. POINTERS
782 002402 017704 003622      MOV      #MTAB,R4      ;POINT TO TABLE OF MESSAGE POINTERS
783 002406 005067 001174      CLR      RTRV          ;CLEAR RETRY FLAGS
784 002412 005777 001152      IS:     TST      @DLRDRR ;FLUSH RCVR INPUT BUFFER REG
785 002416 005777 001146      TST      @DLRDRR
786 002422 012367 001204      MOV      (R3)+,LDOUT   ;SET UP CORRECT LOAD BUF ADDRESS POINTER
787 002426 012467 001202      JSR      (R4)+,AMESS   ;SET UP MESSAGE POINTER
788 002432 004767 000436      CMP      PC,SECT      ;GO DO A SEGMENT
789 002436 022703 003622      BNE      IS           ;DONE ALL FOUR SEGMENTS ??
790                                     ;BR IF NOT
791
792
793
794
795
796
797 002444 104413 000000      ENDTTS,BEGIN          ;SIGNAL END OF ITERATION.
798 002450 000752      BR      DDACIN        ;MONITOR SHALL TEST END OF PASS
799
800
801
802
803
  
```

```

804                                     ;THIS ROUTINE SERVICES ALL XMITTR INTRPTS. FOR ALL 256. BYTE XFERS
805
806
807 002452* 105777 001114 XINT: TSTB @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 BGE JS ;BR IF YES
814 002476* 117777 001076 001070 2S: MOVB OPTR,@DLXDBR ;OUTPUT CHARACTER
815 002504* 005267 001070 INC OPTR ;POINT TO NEXT CHAR. IN BUFFER
816
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 SS ;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* 042777 001084 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 002560* 104405 000000* 000000 HRDRS,BEGIN,NULL ;RECEIVER FALSE INTERRUPT - FATAL ERROR
830
831
832 002566* 005077 000774 CLR @DLRCSR ;TURN OFF RCVR INTR.
833 002572* 104410 000000* ENDS,BEGIN
834
835 002576* 042777 000100 000766 5S: BIC #100,@DLXCSR ;DISABLE XMITTR. INTERRUPTS
836 002604* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
837
838 002610* 105767 000767 6S: TSTR XEND+1 ;ANY FATAL RCVR. ERRORS PENDING ??
839 002614* 001370 BNE SS ;BR IF YES
840 002616* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
841
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 ZS ;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 BGE JS ;BR IF YES
858 002662* 117777 000702 000706 MOVB @DLRDBR,@IPTR ;READ THE DL INPUT BUFFER REG.
859 002670* 005267 000702 INC IPTR ;POINT TO NEXT CHAR. POSITION

```

```

860 002674*
861 002674* 000004 000000* 003062* PIRQS,BEGIN,6S ; QUEUE UP TO CONTINUE AT 6S AND RTI
862
863
864 002702* 105767 000674 3S: TSTB XEND ;ANY FATAL XMITTR ERROR PENDING
865 002706* 001925 BNE 4S ;BR IF YES
866 002710* 019767 000652 175162 MOV DLXCSR,CSRA ;SAVE THE RCVR. CSR ADDRESS
867 002716* 017767 000644 175156 MOV @DLXCSR,ACSR ;SAVE CONTENTS OF CSR
868 002724* 042777 000100 000634 BIC #100,@DLRCSR ;TURN OFF THE RCVR.
869 002732* 105167 000645 COMB XEND+1 ;SET FATAL RCVR ERROR FLAG
870
871
872 002736* 012767 000011 175142 MOV #11,ERRTYP
873 002744* 104405 000000* 000000 HRDRS,BEGIN,NULL ;RECEIVER FALSE INTERRUPT - FATAL ERROR
874
875
876 002752* 005077 000614 CLR @DLXCSR ;DISABLE XMITTR TOO
877 002756* 104410 000000* ENDS,BEGIN
878
879 002762* 042777 000100 000576 4S: BIC #100,@DLRCSR ;DISABLE RCVR INTERRUPTS
880 002770* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
881
882 002774* 105767 000602 5S: TSTB XEND ;ANY FATAL XMITTR ERRORS PENDING ??
883 003000* 001370 BNE 4S ;BR IF YES
884 003002* 016767 000560 175070 MOV DLRCSR,CSRA ;SAVE CSR ADDRESS
885 003010* 017767 000552 175064 MOV @DLRCSR,ACSR ;SAVE CONTENTS OF CSR
886 003018* 017767 000546 175060 MOV @DLRDRR,ASTAT ;SAVE THE ERROR FLAGS
887 003024* 042777 000100 000534 BIC #100,@DLRCSR ;DISABLE RCVR INTR.
888
889 003032* 012767 000017 175046 MOV #17,ERRTYP
890 003040* 104405 000000* 000000 HRDRS,BEGIN,NULL ;OVERRUN - PARITY - FRAMING ERROR
891
892
893 003046* 005077 000520 CLR @DLXCSR ;DISABLE XMITTR TOO
894 003052* 105267 000510 INCR RETRY ;SET RETRY FLAG
895 003056* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
896
897 003062* 105767 000514 6S: TSTB XEND ;ANY FATAL XMITTR ERRORS PENDING ??
898 003066* 001335 BNE 4S ;BR IF YES
899 003070* 104400 000000* EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
900
901

```

```

902 ;THIS ROUTINE CONTROLS THE EXECUTION OF EACH OF THE FOUR DATA PATTERNS
903
904 003074* 000740
905 003076* 004767 000320
906 003102* 004777 000524
907 003106* 004767 000106
908 003112* 005002
909
910 003114* 104407 000000*
911 003120* 104407 000000*
912 003130* 005767 000452
913
914 003132* 104400 000000*
915 003136* 105767 000444
916 003142* 007411
917 003144* 105067 000436
918 003150* 105267 000433
919 003154* 122767 000003 000425
920 003162* 001345
921 003164* 004767
922 003166* 022767 005122* 000402
923 003174* 001406
924 003176* 005302
925 003180* 001345
926 003202*
927 003202* 104403 000000* 003634*
928 003210* 000207
929 003212* 004767 000034
930 003216* 000207
931
932
933
934
935 003220* 012767 004122* 000352
936 003226* 022767 004522* 000342
937 003234* 004767 000104 000330
938 003242* 052777 000100 000316
939 003250* 000207
940
941
942 ;THIS ROUTINE CHECKS FOR AND REPORTS DATA COMPARE ERRORS
943
944 003252* 042777 000100 000312
945 003256* 042777 000100 000306
946 003266* 012700 004122*
947 003272* 012701 004522*
948 003306* 001064
949 003306* 001064
950 003302* 022701 005122*
951 003306* 001373
952 003310* 000207
953
954 003312* 016767 000250 174560
955 003320* 114067 174562
956 003324* 042767 174500
957 003332* 010667 174544

```

```

958 003336* 114167 174546
959 003342* 042767 174540
960 003350* 010167 174530
961
962
963 003354* 104404 000000*
964
965
966 003360* 105720
967 003362* 105721
968 003364* 000746
969
970
971
972
973 003396* 016767 000172 174504
974 003398* 016767 000166 174500
975 003402* 000207
976
977 003404* 016767 000162 174466
978 003412* 017767 000154 174462
979 003420* 000207
980
981
982 ;THIS ROUTINE IS USED TO CLEAR THE INPUT/OUTPUT BUFFERS
983
984 003422* 012700 004122*
985 003426* 005020
986 003430* 022700 005122*
987 003434* 001373
988 003436* 000207
989
990
991
992
993 003440* 012700 004122*
994 003444* 105020
995 003446* 112020 000377
996 003452* 022700 004522*
997 003456* 001373
998 003460* 000207
999
1000
1001
1002
1003
1004 003462* 012700 004122*
1005 003466* 005001
1006 003470* 110100
1007 003472* 022700 004522*
1008 003476* 001402
1009 003500* 105201
1010 003502* 000773
1011 003504* 000207
1012
1013

```

1014  
1015  
1016 003506 012700 004122  
1017 003512 012701 000377  
1018 003516 110120  
1019 003520 022700 004522  
1020 003524 001402  
1021 003528 105301  
1022 003532 000771  
1023 003536 000771  
1024 003532 000207  
1025  
1026  
1027  
1028 003534 012700 004122  
1029 003540 012701 005124  
1030 003544 000000  
1031 003548 022700 004522  
1032 003552 001404  
1033 003554 022701 005224  
1034 003560 001771  
1035 003564 000771  
1036 003564 000207

```

;-----
LDOUT3: MOV #DLBUF0,R0 ;SET UP POINTER
        MOV #377,R1 ;START R1 AT 377
1$: MOV #R1,R0 ;LOAD ONE CHAR-
        CMP #DLBUF1,R0 ;AT END OF THE BUFFER ??
        BEQ #R1 ;IF YES
        DECR R1 ;GENERATE NEXT CHAR.
        BR #R1 ;GO MOVE IT
2$: RTS PC ;RETURN TO CALLING SEGMENT
;-----
;THIS ROUTINE LOADS THE WORST CASE PATTERN
;-----
LDOUT4: MOV #DLBUF0,R0 ;SET UP POINTERS
1$: MOV #WCASE,R1 ;POINT TO MONITOR'S WORST CASE PATTERN
2$: MOV #R1,R0 ;LOAD ONE WORD
        CMP #DLBUF1,R0 ;BUFFER FULL ???
        BEQ #R1 ;IF YES
        CMP #WCASEE,R1 ;END OF WORST CASE PATTERN ??
        BEQ #R1 ;GO RESET R1
        BR #R1 ;GO RESET R1
3$: 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 003514  
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 042825 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 042503 052521 047105  
1077 003734 042503 040440 047502  
1078 003742 052122 042105 000045  
1079 003750 041045 047111 051101  
1080 003756 020131 047504 047174  
1081 003764 041440 052517 052116  
1082 003772 051440 050505 042525  
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
FEND: OPEN ;FATAL ERROR END FLAGS
EPCR: OPEN ;END OF PASS COUNTER
RTRY: 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-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*/

```





DLT07	001466R	618	628#															
DLT10	001974R	655#	664#															
DLT11	001974R	707	718#															
DLT12	002072R	713	737#	742	752#													
DLT13	002240R	414*	425*	445	458*	459	468*	469	488*	503*	506	544*	556*	565*				
DLKCSR	003572R	317#	327#	337#	347#	357#	367#	377#	387#	397#	407#	417#	427#	437#	447#	457#	467#	477#
DLXDBR	003574R	416*	425*	445	458*	459	468*	469	488*	503*	506	544*	556*	565*				
DDAGLN	003768R	782*	792*	802*	812*	822*	832*	842*	852*	862*	872*	882*	892*	902*	912*	922*	932*	942*
DDPMS	003848R	373#	383#	393#	403#	413#	423#	433#	443#	453#	463#	473#	483#	493#	503#	513#	523#	533#
DVIDI	000014R	359#	369#	379#	389#	399#	409#	419#	429#	439#	449#	459#	469#	479#	489#	499#	509#	519#
ENDIT3	104413	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
ENDS	104410	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
EPCR	003520R	1044#	1054#	1064#	1074#	1084#	1094#	1104#	1114#	1124#	1134#	1144#	1154#	1164#	1174#	1184#	1194#	1204#
ERRRTP	000106R	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
EXITS	104400	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
GETPA3	104415	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
GWBUP3	104414	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
HRDCNT	000040R	377#	387#	397#	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#
HRDSCR3	104405	622#	632#	642#	652#	662#	672#	682#	692#	702#	712#	722#	732#	742#	752#	762#	772#	782#
HRDPAS	000050R	758#	768#	778#	788#	798#	808#	818#	828#	838#	848#	858#	868#	878#	888#	898#	908#	918#
ICDNT	000398R	374#	384#	394#	404#	414#	424#	434#	444#	454#	464#	474#	484#	494#	504#	514#	524#	534#
ICDUMT	000040R	377#	387#	397#	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#
IDNUM	000122R	399#	409#	419#	429#	439#	449#	459#	469#	479#	489#	499#	509#	519#	529#	539#	549#	559#
INIT	000030R	366#	376#	386#	396#	406#	416#	426#	436#	446#	456#	466#	476#	486#	496#	506#	516#	526#
INTPLG	000040R	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
INTR	000170R	398#	408#	418#	428#	438#	448#	458#	468#	478#	488#	498#	508#	518#	528#	538#	548#	558#
IPTR	003576R	858#	868#	878#	888#	898#	908#	918#	928#	938#	948#	958#	968#	978#	988#	998#	1008#	1018#
ITKTOP	003220R	907#	917#	927#	937#	947#	957#	967#	977#	987#	997#	1007#	1017#	1027#	1037#	1047#	1057#	1067#
LDGMT	004632R	787#	797#	807#	817#	827#	837#	847#	857#	867#	877#	887#	897#	907#	917#	927#	937#	947#
LDOUT1	003440R	993#	1003#	1013#	1023#	1033#	1043#	1053#	1063#	1073#	1083#	1093#	1103#	1113#	1123#	1133#	1143#	1153#
LDOUT2	003462R	1004#	1014#	1024#	1034#	1044#	1054#	1064#	1074#	1084#	1094#	1104#	1114#	1124#	1134#	1144#	1154#	1164#
LDOUT3	003506R	1016#	1026#	1036#	1046#	1056#	1066#	1076#	1086#	1096#	1106#	1116#	1126#	1136#	1146#	1156#	1166#	1176#
LDOUT4	003510R	1028#	1038#	1048#	1058#	1068#	1078#	1088#	1098#	1108#	1118#	1128#	1138#	1148#	1158#	1168#	1178#	1188#
LDTAB	003612R	1025#	1035#	1045#	1055#	1065#	1075#	1085#	1095#	1105#	1115#	1125#	1135#	1145#	1155#	1165#	1175#	1185#
MAP223	104416	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
MODEM	001332R	580#	590#	600#	610#	620#	630#	640#	650#	660#	670#	680#	690#	700#	710#	720#	730#	740#
MODMAN	004002R	323#	333#	343#	353#	363#	373#	383#	393#	403#	413#	423#	433#	443#	453#	463#	473#	483#
MODSP	000224R	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
MSCN3	104403	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
MSCS	104402	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
MSCS	104401	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
MSCI	003644R	1057#	1067#	1077#	1087#	1097#	1107#	1117#	1127#	1137#	1147#	1157#	1167#	1177#	1187#	1197#	1207#	1217#
MSG1	003705R	1058#	1068#	1078#	1088#	1098#	1108#	1118#	1128#	1138#	1148#	1158#	1168#	1178#	1188#	1198#	1208#	1218#
MSG2	003750R	1059#	1069#	1079#	1089#	1099#	1109#	1119#	1129#	1139#	1149#	1159#	1169#	1179#	1189#	1199#	1209#	1219#
MSG3	003750R	1059#	1069#	1079#	1089#	1099#	1109#	1119#	1129#	1139#	1149#	1159#	1169#	1179#	1189#	1199#	1209#	1219#
MSG4	004053R	1059#	1069#	1079#	1089#	1099#	1109#	1119#	1129#	1139#	1149#	1159#	1169#	1179#	1189#	1199#	1209#	1219#
MSG5	004053R	1059#	1069#	1079#	1089#	1099#	1109#	1119#	1129#	1139#	1149#	1159#	1169#	1179#	1189#	1199#	1209#	1219#
MTAB	003622R	783#	793#	803#	813#	823#	833#	843#	853#	863#	873#	883#	893#	903#	913#	923#	933#	943#
NULL	000000	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
		956	966	976	986	996	1006	1016	1026	1036	1046	1056	1066	1076	1086	1096	1106	1116

OPEN	000000	354	360	361	362	363	380	381	382	383	384	385	386	387				
OPTR	003600R	1045#	1046#	1047#	1048#	1049#	1050#	1051#	1052#	1053#	1054#	1055#	1056#	1057#	1058#	1059#	1060#	1061#
DTAAS	104420	1045#	1046#	1047#	1048#	1049#	1050#	1051#	1052#	1053#	1054#	1055#	1056#	1057#	1058#	1059#	1060#	1061#
PAS CNT	000344R	369#	379#	389#	399#	409#	419#	429#	439#	449#	459#	469#	479#	489#	499#	509#	519#	529#
PIR0	000004	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
POPSP	005726	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
POPSP2	022626	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY0	000000	358#	368#	378#	388#	398#	408#	418#	428#	438#	448#	458#	468#	478#	488#	498#	508#	518#
PRY1	000040	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY2	000100	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY3	000160	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY4	000200	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY5	000240	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY6	000300	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PRY7	000340	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PS	002240	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PSW	177776	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PUSH	005746	407#	417#	427#	437#	447#	457#	467#	477#	487#	497#	507#	517#	527#	537#	547#	557#	567#
PUSH	024545	407#	417#	427#	437#	447#												

WASADR	000104R	390#	584*	960*																	
WCASE	005144R	1029	1109#																		
WCASE	005224R	1033	1121#																		
WDPR	000116R	397#																			
WDTD	000114R	396#																			
XEND	003602R	773*	821	826*	839	865	870*	883	899	912	1048#										
XPLAG	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=DDXCON,XDLB80  
RUN-TIME: 7 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

*mf*

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	XX G	R	1976 1979

TITLE CXDLBB1 DL11-E MODULE

AUTHOR D. BUTENHOF MAINTAINER D. BUTENHOF SPT GRP MAINTAINER D. BUTENHOF SUBMITTING ENGINEER D. BUTENHOF

PRODUCT COMPONENTS						
CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.
	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			Y N	Y N	40 SECONDS	10 SECONDS									

DECO/DEPO INFORMATION									
PROBLEM REPORTS CLOSED:									
DEVICE AFFECTED DEC/X11					MULTIMEDIA AFFECTED? <input checked="" type="checkbox"/> YES <input type="checkbox"/> 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. Butenhof</i>	MANUFACTURING ENGINEER <i>Robert M. M...</i>	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER Q99-05460
DATE: 23-jan-79	DATE: 7-Feb-79	DATE:	
MAINTAINER <i>D. Butenhof</i>	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MCF# 2838
DATE: 23-jan-79	DATE:	DATE:	