.9EM

IDENTIFICATION

PRODUCT CODE: AC-E998B-MC

PRODUCT NAME: CXDL890 DL11-E MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

DXDL8 IS AN IOMOD THAT EXERCISES ONE DL11-E ASYNCHRONOUS COMMUNICATIONS INTERFACE (W7800). 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 HARDCORE 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-NULL 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:: DL8 REQUIRES:
1. DECIMAL WORDS: 1354
2. OCTAL WORDS: 02512
3. OCTAL BYTES: 5224

3.0 PASS DEFINITION

ONE PASS OF "DXDL88" 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 BR1: 4 BR2: 0
DVC: 1 SR1: 0

REQUIRED PARAMETERS:

SR1 TO EXERCISE THOSE STATIC TESTS REQUIRING THE USE OF THE H315 MODEM TEST CONNECTOR (MODEM CONTROL LOGIC) BIT 15 OF SR1 MUST BE SET TO A "1". IE SR1=100000.

NOTE: IF SR1 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. SR1 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.

SEQ 0005

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 (INITIALIAZED 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-MONOTOR 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.

SEQ 0006

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 "SEGX" 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.

CLDL8F: 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 MONOTOR'S WORST CASE PATTERN.

8.0 OPERATOR OPTIONS

A. USE THE MOD COMMAND TO MODIFY LOCATION "DLB 16" TO CHANGE SR1. REFER TO PARA. 5.0.

B. MODIFYING THE CONTENTS OF MODULE LOCATION "RESTRT +2" ALLOWS THE USER TO VARY THE TOTAL NO. OF BYTES TRANSFERED 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

MSG4: "WORST CASE SEQUENCE ABORTED"

8. 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'                 IDWORD  <DLBR >,175610,300,4,,,1024.,<161
000000'                 MODULE  140000,DLBR ,175610,300,4,,,1024.,,161
                        .TITLE  DLBR DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION   6      23-MAY-78
                                .LIST     BIN
                                ;*****************************************************************
000000'                         BEGIN:
000000' 046104  041102      040 MODNAM: .ASCII  /DLBB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0PEN        ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 175610                  ADDR1:  175610+0            ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+0               ;1ST DEVICE VECTOR.
000012'     200                 BR1:    .BYTE   PRTY4+0     ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY+0      ;2ND BR 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                  ICONT:  1024.               ;# OF ITERATIONS PER PASS=1024.
000040' 000000                  ICOUNT: 0                   ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 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
000050' 000000                  HRDPAS: 0                   ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                   ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                   ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:
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.
000100' 000000                  CSRA:   OPEN                ;ADDR OF CURRENT CSR.
000102'                         SRADR:                      ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                ;CONTENTS OF CSR.
000104'                         #ASADR:                     ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                ;STATUS REG CONTENTS.
000106'                         ERRTYP:                     ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                ;ACTUAL DATA.
000112' 002334'                 RSTRT:  RESTRT              ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                ;# OF INTERRUPTS PER ITERATION
000122' 000161                  IDNUM:  161                 ;MODULE IDENTIFICATION NUMBER=161
```

```
                000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                                .NLIST
                                                .WORD   0
                                                .LIST
                                                .ENDR
          000224'                       MODSP:
    407                                 ;*****************************************************************
```



```
    408
    409  000224' 016700  177556          START:  MOV     ADDR,R0         ;GET BASE DEVICE ADDRESS
    410  000230' 010067  003332                  MOV     R0,DLRCSR       ;SET UP RCVR CSR ADDRESS
    411  000234' 005720                          TST     (R0)+
    412  000236' 010067  003326                  MOV     R0,DLRDBR       ;SET UP RCVR DBR ADDRESS
    413  000242' 005720                          TST     (R0)+
    414  000244' 010067  003322                  MOV     R0,DLXCSR       ;SET UP XMITTR CSR ADDRESS
    415  000250' 005720                          TST     (R0)+
    416  000252' 010067  003316                  MOV     R0,DLXDBR       ;SET UP XMITTR DBR ADDRESS
    417
    418
    419                                 ;               *******************************
    420                                 ;               *  SECTION ONE                *
    421                                 ;               *******************************
    422
    423  000256' 005077  003304          DLINIT: CLR     @DLRCSR         ;CLEAR OUT BOTH CSR'S
    424  000262' 005077  003300                  CLR     @DLRCSR         ;MAKE SURE ALL DATA COMM BITS CLEARED
    425  000266' 005077  003300                  CLR     @DLXCSR
    426  000272' 005777  003272                  TST     @DLRDBR         ;FLUSH RCVR DONE BIT
    427  000276' 005777  003266                  TST     @DLRDBR
    428
    429                                 ;THIS TEST VERIFIES THAT RCVR CSR GOT CLEARED UPON ENTRY
    430                                 ;-----------------------------------------------------
    431
    432  000302' 005777  003260          DLT01:  TST     @DLRCSR         ;IS RCVR CSR ALL ZEROES ??
    433  000306' 001415                  BEQ     DLT02           ;BR IF YES
    434  000310' 004767  003052                  JSR     PC,STATR        ;GO SET UP ERROR INFO
    435  000314' 012767  000025  177564          MOV     #25,ERRTYP      ;
    436                                 ;*****************************************************************
    437  000322' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CANT'T CLEAR OUT RCVR CSR
    438                                 ;*****************************************************************
    439  000330' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
    440  000336' 104410  000000'                  ENDS,BEGIN              ;
    441
    442                                 ;TEST THAT READY BIT IS ONLY BIT SET IN XMITTR CSR
    443                                 ;------------------------------------------------
    444
    445  000342' 022777  000200  003222  DLT02:  CMP     #200,@DLXCSR    ;READY SET ??
    446  000350' 001415                  BEQ     DLT03           ;BR IF YES
    447  000352' 004767  003026                  JSR     PC,STATX        ;GO SET UP ERROR INFO
    448  000356' 012767  000025  177522          MOV     #25,ERRTYP      ;
    449                                 ;*****************************************************************
    450  000364' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;READY NOT SET OR OTHER BITS DIDN'T CLEAR IN XMIT CSR
    451                                 ;*****************************************************************
    452  000372' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
    453  000400' 104410  000000'                  ENDS,BEGIN              ;
    454
    455                                 ;TEST THAT MAINT. BIT CAN BE SET AND CLEARED IN XMIT CSR
    456                                 ;------------------------------------------------------
    457
    458  000404' 052777  000004  003160  DLT03:  BIS     #4,@DLXCSR      ;SET THE MAINT. BIT
    459  000412' 022777  000204  003152          CMP     #204,@DLXCSR    ;DID IT SET ??
    460  000420' 001415                  BEQ     IS              ;BR IF YES
    461  000422' 004767  002756                  JSR     PC,STATX        ;GO SET UP ERROR INFO
    462  000426' 012767  000033  177452          MOV     #33,ERRTYP      ;
    463                                 ;*****************************************************************
```

```
000434' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MAINT BIT WON'T SET OR IT CLEARED READY
                                        ;*****************************************************************
000442' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000450' 104410  000000'                 ENDS,BEGIN              ;
000454' 042777  000004  003110  1$:     BIC     #4,@DLXCSR      ;NOW CLR THE MAINT BIT
000462' 022777  000200  003102          CMP     #200,@DLXCSR    ;DID IT CLEAR ??
000470' 001415                          BEQ     DLT04           ;BR IF YES
000472' 004767  002706                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000476' 012767  000025  177402          MOV     #25,ERRTYP
                                        ;*****************************************************************
000504' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MAINT. BIT WON'T CLEAR OR READY CLEARED
                                        ;*****************************************************************
000512' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000520' 104410  000000'                 ENDS,BEGIN              ;


                                ;TEST THAT BIT 06 IN XCSR CAN CAUSE AN INTERRUPT
                                ;-----------------------------------------------

000524' 005067  003060          DLT04:  CLR     INTFLG          ;INIT THE SOFTWARE INTR. FLAG
000530' 016700  177254                  MOV     VECTOR,R0       ;GET BASE VECTOR ADDRESS
000534' 062700  000004                  ADD     #4,R0           ;GENERATE ADDR OF XMIT VECTOR
000540' 012720  000634'                 MOV     #2$,(R0)+       ;GO TO 2S ON XMIT INTERRUPT
000544' 116710  177242                  MOVB    BR1,(R0)        ;PRIORITY LEVEL = BR1
000550' 005001                          CLR     R1              ;INIT BREAK TIMER
000552' 052777  000100  003012          BIS     #100,@DLXCSR    ;SET INTR. ENAB
000560'                         1$:
000560' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000564' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000570' 005767  003014                  TST     INTFLG          ;DID XMIT INTR OCCUR YET ??
000574' 001025                          BNE     3$              ;BR IF IT DID
000576' 005301                          DEC     R1              ;COUNT BREAK TIMER
000600' 001367                          BNE     1$              ;BR IF NO TIMEOUT
000602' 004767  002576                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000606' 012767  000023  177272          MOV     #23,ERRTYP
                                        ;*****************************************************************
000614' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;XMITTER FAILED TO GENERATE INTERRUPT
                                        ;*****************************************************************
000622' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000630' 104410  000000'                 ENDS,BEGIN              ;
000634' 042777  000100  002730  2$:     BIC     #100,@DLXCSR    ;DISABLE XMITTR INTR ENABLE
000642' 005167  002742                  COM     INTFLG          ;SET THE INTR. FLAG
000646' 000002                          RTI                     ;RETURN CONTROL TO OTHER GUY
000650' 022777  000200  002714  3$:     CMP     #200,@DLXCSR    ;DID I.E. GET CLEARED IN INTR. SERVICE
000656' 001415                          BEQ     DLT05           ;BR IF YES
000660' 004767  002520                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000664' 012767  000027  177214          MOV     #27,ERRTYP
                                        ;*****************************************************************
000672' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"DONE" OR RCVR INTR. ENAB FAILED TO CLEAR
                                        ;*****************************************************************
000700' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000706' 104410  000000'                 ENDS,BEGIN              ;


                                ;TEST THAT A RCVR INTR CAN OCCUR WHEN "DONE" GETS SET
                                ;----------------------------------------------------

000712' 005067  002672          DLT05:  CLR     INTFLG          ;INIT SOFTWARE INTR. FLAG
```



```
000716' 016700  177066                  MOV     VECTOR,R0       ;GET THE BASE VECTOR ADDRESS
000722' 012720  001162'                 MOV     #4$,(R0)+       ;GO TO 4S ON RCVR INTERRUPT
000726' 116710  177060                  MOVB    BR1,(R0)        ;SET PRIORITY
000732' 052777  000100  002626          BIS     #100,@DLRCSR    ;SET I.E. IN RCVR CSR
000740' 032777  000100  002620          BIT     #100,@DLRCSR    ;DID IT SET
000746' 001015                          BNE     1$              ;BR IF IT DID
000750' 004767  002412                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000754' 012767  000033  177124          MOV     #33,ERRTYP
                                        ;*****************************************************************
000762' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET BIT 06 IN RCSR I.E.
                                        ;*****************************************************************
000770' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000776' 104410  000000'                 ENDS,BEGIN              ;
001002' 042777  000100  002556  1$:     BIC     #100,@DLRCSR    ;NOW CLEAR THE I.E. BIT
001010' 032777  000100  002550          BIT     #100,@DLRCSR    ;DID I.E. BIT GET CLEARED ??
001016' 001415                          BEQ     2$              ;BR IF YES
001020' 004767  002342                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001024' 012767  000023  177054          MOV     #23,ERRTYP
                                        ;*****************************************************************
001032' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T CLEAR RCSR INTR. ENAB. BIT
                                        ;*****************************************************************
001040' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001046' 104410  000000'                 ENDS,BEGIN              ;
001052' 052777  000100  002506  2$:     BIS     #100,@DLRCSR    ;NOW TURN IT ON FOR REAL
001060' 052777  000004  002504          BIS     #4,@DLXCSR      ;TURN ON MAINT. MODE
001066' 005001                          CLR     R1              ;INIT BREAK TIMER
001070' 112777  000252  002476          MOVB    #252,@DLXDBR    ;LOAD THE XMITTR OUTPUT DATA BUFFER
001076'                         3$:
001076' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001102' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001106' 005767  002476                  TST     INTFLG          ;DID RCVR INTR. YET ??
001112' 001036                          BNE     5$              ;BR IF IT DID
001114' 005301                          DEC     R1              ;COUNT BREAK TIMER
001116' 001367                          BNE     3$              ;BR IF NO TIMEOUT
001120' 004767  002242                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001124' 005077  002436                  CLR     @DLRCSR         ;CLEAR BOTH CSR'S
001130' 005077  002436                  CLR     @DLXCSR
001134' 012767  000023  176744          MOV     #23,ERRTYP
                                        ;*****************************************************************
001142' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RCVR FAILED TO INTR ON TIME
                                        ;*****************************************************************
001150' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001156' 104410  000000'                 ENDS,BEGIN              ;
001162' 117767  002402  176720  4$:     MOVB    @DLRDBR,AWAS    ;GET THE RECEIVED DATA
001170' 042777  000100  002370          BIC     #100,@DLRCSR    ;TURN OFF I.E.
001176' 005077  002370                  CLR     @DLXCSR         ;TURN OFF MAINTENANCE MODE
001202' 005167  002402                  COM     INTFLG          ;SET SOFTWARE INTR. FLAG
001206' 000002                          RTI                     ;RETURN TO OTHER GUY
001210' 005777  002352          5$:     TST     @DLRCSR         ;DID INTR SERVICE CLEAR THE RCVR CSR ??
001214' 001421                          BEQ     6$              ;BR IF IT DID
001216' 004767  002144                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001222' 005077  002340                  CLR     @DLRCSR         ;CLEAR BOTH CSR'S
001226' 005077  002340                  CLR     @DLXCSR
001232' 012767  000025  176646          MOV     #25,ERRTYP
                                        ;*****************************************************************
001240' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RCVR INTR SERVICE FAILED TO CLEAR I.E. AND DONE
```

```
                                        ;*****************************************************************
001246' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001254' 104410  000000'                 ENDS,BEGIN              ;
001260' 122767  000252  176622  6S:     CMPB    #252,AWAS       ;WAS DATA CORRECT ???
001266' 001421                          BEQ     MODEM           ;BR IF YES
001270' 016767  002272  176602          MOV     DLRCSR,CSRA     ;SET UP ERROR INFO
001276' 017767  002264  176576          MOV     @DLRCSR,ACSR
001304' 012767  000252  176574          MOV     #252,ASR
001312' 016767  002252  176564          MOV     DLRDBR,WASADR
001320' 016767  002250  176554          MOV     DLXDBR,SBADR
                                        ;*****************************************************************
001326' 104404  000000'                 DATERS,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************


                                ;THE FOLLOWING GROUP OF BASIC TESTS ASSUME THAT THE MODEM IS DISCONNECT-
                                ;ED FROM THE DLII-E AND AN H315 MODEM TEST CONNECTOR IS CONNECTED IN ITS
                                ;PLACE. ALL OF THESE TESTS ARE SKIPPED IF SR1(15) IS CLEARED.

001332' 005767  176460          MODEM:  TST     SR1             ;IS SR1 BIT 15 = 1 ??
001336' 100402                          BMI     DLT06           ;DO MODEM TESTS IF SET
001340' 000167  000770                  JMP     RESTRT          ;SKIP MODEM TESTS IF SR1(15)=0


                                ;TEST THAT "REQ TO SEND" CAN ASSERT "RING"
                                ;-----------------------------------------

001344' 052777  000004  002214  DLT06:  BIS     #4,@DLRCSR      ;SET REQ TO SEND
001352' 032777  000004  002206          BIT     #4,@DLRCSR      ;DID IT SET ??
001360' 001010                          BNE     1$              ;BR IF YES
001362' 004767  002000                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001366' 012767  000025  176512          MOV     #25,ERRTYP
                                        ;*****************************************************************
001374' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET REQ TO SEND
                                        ;*****************************************************************
001402' 032777  040000  002156  1S:     BIT     #40000,@DLRCSR  ;DID "RING" GET ASSERTED ??
001410' 001010                          BNE     2$              ;BR IF YES
001412' 004767  001750                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001416' 012767  000025  176462          MOV     #25,ERRTYP
                                        ;*****************************************************************
001424' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"REQ TO SEND" DIDN'T ASSERT "RING"
                                        ;*****************************************************************
001432' 042777  000004  002126  2S:     BIC     #4,@DLRCSR      ;TURN OFF "REQ TO SEND"
001440' 005777  002122                  TST     @DLRCSR         ;ARE ALL BITS NOW CLEAR ??
001444' 001410                          BEQ     DLT07           ;BR IF BOTH "RING" AND "REQ TO SEND " CLEARED
001446' 004767  001714                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001452' 012767  000025  176426          MOV     #25,ERRTYP
                                        ;*****************************************************************
001460' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"RING" OR "REQ TO SEND" FAILED TO CLEAR
                                        ;*****************************************************************


                                ;TEST THAT "SEC XMIT" ASSERTS "SEC REC" WHICH SETS "DATA SET INT"
                                ;---------------------------------------------------------------

001466' 052777  000010  002072  DLT07:  BIS     #10,@DLRCSR     ;SET SEC XMIT
001474' 005777  002066                  TST     @DLRCSR         ;DID DATA SET INT GET SET ??
001500' 100410                          BMI     1S              ;BR IF YES
001502' 004767  001660                  JSR     PC,STATR        ;GO SET UP ERROR INFO
```



```
001506' 012767  000025  176372          MOV     #25,ERRTYP
                                        ;*****************************************************************
001514' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DATA SET INT FAILED TO SET
                                        ;*****************************************************************
001522' 022777  002010  002036  1S:     CMP     #2010,@DLRCSR   ;SEC XMIT AND REC SET - DATA SET INT CLEAR
001530' 001410                          BEQ     2$              ;BR IF YES
001532' 004767  001630                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001536' 012767  000025  176342          MOV     #25,ERRTYP
                                        ;*****************************************************************
001544' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SEC XMIT OR SEC REC FAILED TO SET - DATA SET INT FAILED
                                        ;*****************************************************************
001552' 042777  000010  002006  2S:     BIC     #10,@DLRCSR     ;TURN OFF SEC XMIT
001560' 005777  002002                  TST     @DLRCSR         ;DID SEC XMIT GOING OFF SET DATA SET INT ?
001564' 100410                          BMI     3$              ;BR IF YES
001566' 004767  001574                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001572' 012767  000020  176306          MOV     #20,ERRTYP
                                        ;*****************************************************************
001600' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CLEARING SEC XMIT FAILED TO SET DATA SET INT
                                        ;*****************************************************************
001606' 005777  001754          3S:     TST     @DLRCSR         ;ALL BITS NOW CLEAR
001612' 001410                          BEQ     DLT10           ;BR IF YES
001614' 004767  001546                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001620' 012767  000020  176260          MOV     #20,ERRTYP
                                        ;*****************************************************************
001626' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SEC XMIT,SEC REC, OR DATA SET INT FAILED TO CLEAR
                                        ;*****************************************************************


                                ;TEST THAT "DTR" ASSERTS "CAR DET", "CLR TO SEND", AND "DATA SET INT"
                                ;-------------------------------------------------------------------

001634' 005077  001726          DLT10:  CLR     @DLRCSR         ;CLR THE RCVR CSR
001640' 052777  000002  001720          BIS     #2,@DLRCSR      ;SET DATA TERM READY
001646' 005777  001714                  TST     @DLRCSR         ;DID DATA SET INT SET ??
001652' 100407                          BMI     1S              ;BR IF YES
001654' 004767  001506                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001660' 005067  176222                  CLR     ERRTYP
                                        ;*****************************************************************
001664' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"DTR" FAILED TO SET "DATA SET INT"
                                        ;*****************************************************************
001672' 022777  030002  001666  1S:     CMP     #30002,@DLRCSR;DTR,CLR TO SEND, CAR DET, SET AND DATA SET INT CLEAR ?
001700' 001407                          BEQ     2$              ;BR IF YES
001702' 004767  001460                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001706' 005067  176174                  CLR     ERRTYP
                                        ;*****************************************************************
001712' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DTR,CAR DET, OR CLR TO SEND FAILED TO SET OR DATA SET I
                                        ;*****************************************************************
001720' 042777  000002  001640  2S:     BIC     #2,@DLRCSR      ;TURN OFF DTR
001726' 005777  001634                  TST     @DLRCSR         ;DATA SET INT SHOULD HAVE SET
001732' 100407                          BMI     3$              ;BR IF IT DID
001734' 004767  001426                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001740' 005067  176142                  CLR     ERRTYP
                                        ;*****************************************************************
001744' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DATA SET INT FAILED TO SET ON CLEARING DTR
                                        ;*****************************************************************
001752' 005777  001610          3S:     TST     @DLRCSR         ;ALL BITS NOW CLEAR ??
001756' 001407                          BEQ     DLT11           ;BR IF YES
```

```
001760' 004767  001402                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001764' 005067  176116                  CLR     ERRTYP
                                        ;*****************************************************************
001770' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CLEARING DTR FAILED TO CLEAR OTHER BITS
                                        ;*****************************************************************


                                ;TEST THAT DATA SET INTR ENAB CAN SET AND CLEAR
                                ;----------------------------------------------

001776' 052777  000040  001562  DLT11:  BIS     #40,@DLRCSR     ;SET DATA SET I.E.
002004' 032777  000040  001554          BIT     #40,@DLRCSR     ;DID IT SET ??
002012' 001010                          BNE     1$              ;BR IF YES
002014' 004767  001346                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002020' 012767  000027  176060          MOV     #27,ERRTYP
                                        ;*****************************************************************
002026' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET DATA SET INTR.
                                        ;*****************************************************************
002034' 042777  000040  001524  1$:     BIC     #40,@DLRCSR     ;CLEAR DATA SET I.E.
002042' 005777  001520                  TST     @DLRCSR         ;DID IT CLEAR ??
002046' 001411                          BEQ     DLT12           ;BR IF YES
002050' 004767  001312                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002054' 012767  000027  176024          MOV     #27,ERRTYP
                                        ;*****************************************************************
002062' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T CLEAR DATA SET I.E.
                                        ;*****************************************************************
002070' 000463                          BR      DLT13           ;SKIP NEXT TEST


                                ;TEST THAT DATA SET INT CAN CAUSE A RCVR INTERRUPT WHEN ENABLED
                                ;--------------------------------------------------------------


002072' 005067  001512  DLT12:  CLR     INTFLG          ;INIT SOFTWARE INTR. FLAG
002076' 016700  175706          MOV     VECTOR,R0       ;GET RAS VECTOR ADDR.
002102' 012720  002174'         MOV     #2$,(R0)+       ;GO TO 2$ ON DATA SET INTERRUPT
002106' 005001                  CLR     R1              ;INIT BREAK TIMER
002110' 052777  000040  001450          BIS     #40,@DLRCSR     ;ENABLE DATA SET INTR.
002116' 052777  000010  001442          BIS     #10,@DLRCSR     ;SET SEC XMIT
002124'                 1$:
002124' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002130' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002134' 005767  001450                  TST     INTFLG          ;DID INTR OCCUR YET ??
002140' 001023                          BNE     3$              ;BR IF YES
002142' 005301                  2$:     DEC     R1              ;COUNT THE BREAK TIMER
002144' 001367                          BNE     1$              ;BR IF NO TIMEOUT
002146' 004767  001214                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002152' 005077  001410                  CLR     @DLRCSR         ;TURN OFF THE I.E. BIT
002156' 012767  000023  175722          MOV     #23,ERRTYP
                                        ;*****************************************************************
002164' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DATA SET INTR. FAILED TO OCCUR
                                        ;*****************************************************************
002172' 000422                          BR      DLT13           ;GO TO NEXT TEST
002174' 042777  000040  001364  2$:     BIC     #40,@DLRCSR     ;TURN OFF DATA SET I.E.
002202' 005167  001402                  COM     INTFLG          ;SET SOFTWARE INTR. FLAG
002206' 000002                          RTI                     ;RETURN CONTROL TO OTHER GUY
002210' 032777  000040  001350  3$:     BIT     #40,@DLRCSR     ;DID INTR SERVICE TURN OFF I.E. ?
002216' 001410                          BEQ     DLT13           ;BR IF YES
002220' 004767  001142                  JSR     PC,STATR        ;GO SET UP ERROR INFO
```



```
002224' 012767  000023  175654          MOV     #23,ERRTYP
                                        ;*****************************************************************
002232' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;INTR SERVICE FAILED TO CLR DATA SET I.E.
                                        ;*****************************************************************


                                ;TEST THAT "BREAK" BIT CAN SET AND CLEAR
                                ;---------------------------------------


002240' 052777  000001  001324  DLT13:  BIS     #1,@DLXCSR      ;SET BREAK BIT
002246' 032777  000001  001316          BIT     #1,@DLXCSR      ;DID IT SET ??
002254' 001010                          BNE     1$              ;BR IF YES
002256' 004767  001122                  JSR     PC,STATX        ;GO SET UP ERROR INFO
002262' 012767  000025  175616          MOV     #25,ERRTYP
                                        ;*****************************************************************
002270' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET BREAK BIT
                                        ;*****************************************************************
002276' 042777  000001  001266  1$:     BIC     #1,@DLXCSR      ;CLEAR THE BREAK BIT
002304' 032777  000001  001260          BIT     #1,@DLXCSR      ;DID IT CLEAR ?
002312' 001410                          BEQ     XFSTRT          ;BR IF YES
002314' 004767  001064                  JSR     PC,STATX        ;GO SET UP ERROR INFO
002320' 012767  000025  175560          MOV     #25,ERRTYP
                                        ;*****************************************************************
002326' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;BREAK BIT WON'T CLEAR
                                        ;*****************************************************************
```

```
                        ;               **********************
                        ;               *  SECTION TWO       *
                        ;               **********************
                        ;
                        RESTRT:
002334'                 CLR     XEND            ;CLEAR END FLAGS
002334' 005067 001242   CLR     @DLXCSR         ;CLEAR THE DL11 CONTROL REGS
002340' 005077 001226   CLR     @DLRCSR         ;JUST IN CASE
002344' 005077 001216   MOV     VECTOR,R0       ;GET START VECTOR ADDRESS
002350' 016700 175434   MOV     #RINT,(R0)+     ;SET UP THE RCVR AND XMIT VECTORS
002354' 012720 002422'  MOVB    #R1,(R0)
002360' 116710 175426   TST     (R0)+
002364' 005720          MOV     #XINT,(R0)+
002366' 012720 002452'  MOVB    #R1,(R0)
002372' 116710 175414
002376' 012703 003612'  DOAGIN: MOV     #LDTAB,R3       ;POINT TO TABLE OF LOAD SUBR. POINTERS
002402' 012704 003622'          MOV     #MTAB,R4        ;POINT TO TABLE OF MESSAGE POINTERS
002406' 005067 001174   1S:     CLR     RTRY            ;CLEAR RETRY FLAGS
002412' 005777 001152           TST     @DLRDBR         ;FLUSH RCVR INPUT BUFFER REG
002416' 005777 001146           TST     @DLRDBR
002422' 012367 001204           MOV     (R3)+,LDOUT     ;SET UP CORRECT LOAD BUF ADDRESS POINTER
002426' 012467 001202           MOV     (R4)+,AMESS     ;SET UP MESSAGE POINTER
002432' 004767 000436           JSR     PC,SEGX         ;GO DO A SEGMENT
002436' 022703 003622'          CMP     #MTAB,R3        ;DONE ALL FOUR SEGMENTS ??
002442' 001361                  BNE     1S              ;BR IF NOT
                        ;
                        ;
                        ;
                        ;
                        ;
002444' 104413 000000'          ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
002450' 000752                  BR      DOAGIN
                        ;
                        ;
                        ;
```



```
                        ;THIS ROUTINE SERVICES ALL XMITTR INTRPTS. FOR ALL 256. BYTE XFERS
                        ;-----------------------------------------------------------------
002452' 105777 001114   XINT:   TSTB    @DLXCSR         ;XMIT READY SET ??
002456' 100403          BMI     1S              ;BR IF YES
                        ;-----------------------------------------------------------------
002460' 000004 000000' 002516'  PIRQS,BEGIN,4S          ; QUEUE UP TO CONTINUE AT 4S AND RTI
                        ;-----------------------------------------------------------------
002466' 022767 004522' 001104   1S:     CMP     #DLXBUFI,OPTR   ;OUTPUT 256. BYTES YET ??
002474' 001405          BEQ     3S              ;BR IF YES
002476' 117777 001076 001070    2S:     MOVB    @OPTR,@DLXDBR   ;OUTPUT A CHARACTER
002504' 005267 001070           INC     OPTR            ;POINT TO NEXT CHAR. IN BUFFER
002510'                         3S:
                        ;-----------------------------------------------------------------
002510' 000004 000000' 002610'  PIRQS,BEGIN,6S          ; QUEUE UP TO CONTINUE AT 6S AND RTI
                        ;-----------------------------------------------------------------
002516' 105767 001061   4S:     TSTB    XEND+1          ;ANY FATAL RCVR. ERRORS PENDING ??
002522' 001025          BNE     5S              ;BR IF YES - STOP XMITTING
002524' 016767 001042 175146    MOV     DLXCSR,CSRA     ;SAVE THE CSR ADDRESS
002532' 017767 001034 175342    MOV     @DLXCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
002540' 042777 000104 001024    BIC     #104,@DLXCSR    ;DISABLE XMITTR INTERRUPTS
002546' 105167 001030           COMB    XEND            ;SET XMIT END FLAG
                        ;
002552' 012767 000011 175326    MOV     #11,ERRTYP
                        ;*****************************************************************
002560' 104405 000000' 000000   ERRORS,BEGIN,NULL       ;XMITTER FALSE INTERRUPT - FATAL ERROR
                        ;*****************************************************************
                        ;
002566' 005077 000774   CLR     @DLRCSR         ;TURN OFF RCVR INTR.
002572' 104410 000000'  ENDS,BEGIN              ;
                        ;
002576' 042777 000100 000766    5S:     BIC     #100,@DLXCSR    ;DISABLE XMITTR. INTERRUPTS
002604' 104400 000000'          EXITS,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                        ;
002610' 105767 000767   6S:     TSTB    XEND+1          ;ANY FATAL RCVR. ERRORS PENDING ??
002614' 001370          BNE     5S              ;BR IF YES
002616' 104400 000000'          EXITS,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                        ;
                        ;THIS ROUTINE SERVICES RECEIVER INTERRUPTS FOR ALL 256. BYTE XFERS
                        ;-----------------------------------------------------------------
002622' 105777 000740   RINT:   TSTB    @DLRCSR         ;RCVR DONE SET ??
002626' 100403          BMI     1S              ;BR IF YES
                        ;-----------------------------------------------------------------
002630' 000004 000000' 002702'  PIRQS,BEGIN,3S          ; QUEUE UP TO CONTINUE AT 3S AND RTI
                        ;-----------------------------------------------------------------
002636' 005777 000726   1S:     TST     @DLRDBR         ;OVERRUN/PARITY/FRAMING ERRORS ??
002642' 100003          BPL     2S              ;BR IF NONE
                        ;-----------------------------------------------------------------
002644' 000004 000000' 002774'  PIRQS,BEGIN,5S          ; QUEUE UP TO CONTINUE AT 5S AND RTI
                        ;-----------------------------------------------------------------
002652' 022767 005122' 000716   2S:     CMP     #BUFEND,IPTR    ;INPUT BUFFER FULL ??
002660' 001405          BEQ     7S              ;BR IF YES
002662' 117777 000702 000706    MOVB    @DLRDBR,@IPTR   ;READ THE DL INPUT BUFFER REG.
002670' 005267 000702           INC     IPTR            ;POINT TO NEXT CHAR. POSITION
```

```
                                7$:
                                ;-------------------------------------------------------------
002674' 000004  000000' 003062' PIRQ$,BEGIN,6S          ; QUEUE UP TO CONTINUE AT 6S AND RTI
                                ;-------------------------------------------------------------

002702' 105767  000674          3$:     TSTB    XEND            ;ANY FATAL XMITTR ERROR PENDING
002706' 001025                  BNE     4$              ;BR IF YES
002710' 016767  000652  175162  MOV     @LRCSR,CSRA     ;SAVE THE RCVR. CSR ADDRESS
002716' 017767  000644  175156  MOV     @LRCSR,ACSR     ;SAVE CONTENTS OF CSR
002724' 042777  000100  000634  BIC     #100,@LRCSR     ;TURN OFF THE RCVR.
002732' 105167  000645          COMB    XEND+1          ;SET FATAL RCVR ERROR FLAG

002736' 012767  000011  175142  MOV     #11,ERRTYP
                                ;*************************************************************
002744' 104405  000000' 000000  ERROR$,BEGIN,NULL       ;RECEIVER FALSE INTERRUPT - FATAL ERROR
                                ;*************************************************************

002752' 005077  000614          CLR     @LXCSR          ;DISABLE XMITTR TOO
002756' 104410  000000'         ENDS,BEGIN              ;

002762' 042777  000100  000576  4$:     BIC     #100,@LRCSR     ;DISABLE RCVR INTERRUPTS
002770' 104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

002774' 105767  000602          5$:     TSTB    XEND            ;ANY FATAL XMITTR ERRORS PENDING ??
003000' 001370                  BNE     4$              ;BR IF YES
003002' 016767  000560  175070  MOV     LRCSR,CSRA      ;SAVE CSR ADDRESS
003010' 017767  000552  175064  MOV     @LRCSR,ACSR     ;SAVE CONTENTS OF CSR
003016' 017767  000546  175060  MOV     @LRDBR,ASTAT    ;SAVE THE ERROR FLAGS
003024' 042777  000100  000534  BIC     #100,@LRCSR     ;DISABLE RCVR INTS.

003032' 012767  000017  175046  MOV     #17,ERRTYP
                                ;*************************************************************
003040' 104405  000000' 000000  ERROR$,BEGIN,NULL       ;OVERRUN - PARITY - FRAMING ERROR
                                ;*************************************************************

003046' 005077  000520          CLR     @LXCSR          ;DISABLE XMITTR TOO
003052' 105267  000530          INCB    RTRY            ;SET RETRY FLAG
003056' 104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

003062' 105767  000514          6$:     TSTB    XEND            ;ANY FATAL XMITTR ERRORS PENDING ??
003066' 001335                  BNE     4$              ;BR IF YES
003070' 104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                                ;THIS ROUTINE CONTROLS THE EXECUTION OF EACH OF THE FOUR DATA PATTERNS

003074' 000240                  SEQX:   NOP                     ;DO NOTHING FOR NOW
003076' 004767  000320          1$:     JSR     PC,CLDLBF       ;GO CLEAR BUFFERS
003102' 004777  000524                  JSR     PC,@LDOUT       ;GO SET UP  PATTERN
003106' 004767  000106                  JSR     PC,KICKOF       ;GO KICK OFF XMITTR AND RCVR
003112' 005002                          CLR     R2              ;INITIALIZE BREAK TIMER
003114'                         2$:
003114' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003120' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003124' 005767  000452                  TST     XEND            ;ANY FATAL ERRORS PENDING ??
003130' 001402                          BEQ     3$              ;BR IF NOT
003132' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003136' 105767  000444          3$:     TSTB    RTRY            ;RETRY FLAG SET ??
003142' 001411                          BEQ     4$              ;BR IF NOT
003144' 105067  000436                  CLRB    RTRY            ;CLEAR THE FLAG
003150' 105267  000431                  INCB    RTRY+1          ;COUNT ONE RETRY
003154' 122767  000003  000425          CMPB    #3,RTRY+1       ;TRIED THREE TIMES ??
003162' 001345                          BNE     1$              ;BR IF NOT - TRY IT AGAIN
003164' 000406                          BR      5$              ;REPORT IT AND GO TO NEXT SEGMENT
003166' 022767  005122' 000402  4$:     CMP     #BUFEND,IPTR    ;RECEIVED 256. CHARS. ??
003174' 001406                          BEQ     7$              ;BR IF YES
003176' 005302                          DEC     R2              ;DECREMENT BREAK COUNTER
003200' 001345                          BNE     2$              ;BR IF NO TIMEOUT
003202'                         5$:
003202' 104403  000000' 003634' MSGN$,BEGIN,AMESS        ;ASCII MESSAGE CALL WITH COMMON HEADER
003210' 000207                  6$:     RTS     PC              ;GO TO NEXT SEGMENT
003212' 004767  000034          7$:     JSR     PC,CHKDAT       ;GO COMPARE IN/OUT DATA
003216' 000207                          RTS     PC              ;GO TO NEXT SEGMENT

                                ;THIS ROUTINE KICKS OFF ALL 256. BYTE TRANSFERS
                                ;----------------------------------------------

003220' 012767  004122' 000352  KICKOF: MOV     #DLBUFO,OPTR    ;POINT TO BEGINNING OF OUTPUT BUFFER
003226' 012767  004522' 000342          MOV     #DLBUFI,IPTR    ;POINT TO BEGINNING OF INPUT BUFFER
003234' 052777  000104  000330          BIS     #104,@LXCSR     ;TURN ON XMITTER
003242' 052777  000100  000316          BIS     #100,@LRCSR     ;TURN ON RCVR
003250' 000207                          RTS     PC              ;RETURN TO CALLING SEGMENT

                                ;THIS ROUTINE CHECKS FOR AND REPORTS DATA COMPARE ERRORS
                                ;------------------------------------------------------

003252' 042777  000100  000312  CHKDAT: BIC     #100,@LXCSR     ;DISABLE XMITTR INTR.
003260' 042777  000100  000300          BIC     #100,@LRCSR     ;DISABLE RCVR INTR.
003266' 012700  004122'                 MOV     #DLBUFO,R0      ;R0 POINTS TO OUTPUT BUFFER
003272' 012701  004522'                 MOV     #DLBUFI,R1      ;R1 POINTS TO INPUT BUFFER
003276' 122021                  1$:     CMPB    (R0)+,(R1)+     ;COMPARE INPUT WITH OUTPUT
003300' 001004                          BNE     2$              ;BR IF NOT EQUAL
003302' 022701  005122'         3$:     CMP     #BUFEND,R1      ;END OF THE BUFFERS ??
003306' 001373                          BNE     1$              ;BR IF NOT
003310' 000207                          RTS     PC              ;RETURN TO CALLER

003312' 016767  000250  174560  2$:     MOV     DLRCSR,CSRA     ;SAVE THE CSR ADDRESS
003320' 114067  174562                  MOVB    -(R0),ASB       ;SAVE THE SHOULD BE DATA
003324' 042767  177400  174554          BIC     #177400,ASB     ;ZERO HI BYTE
003332' 010067  174544                  MOV     R0,SBADR        ;SAVE THE SHOULD BE ADDRESS
```

```
                                ;SAVE THE WAS DATA
                                ;CLEAR OUT HI BYTE
                                ;SAVE THE WAS ADDRESS
                        ;*******************************************************************
                        DATERS,BEGIN            ;DATA ERROR!!!
                        ;*******************************************************************

                                ;POINT TO NEXT BYTE IN THE BUFFERS

                                ;GO CHECK NEXT BYTE

;THESE ROUTINES SET UP THE ERROR INFORMATION FOR THE BASIC TESTS
;---------------------------------------------------------------

STATR:  MOV     DLRCSR,CSRA     ;SAVE THE CSR ADDRESS
        MOV     @DLRCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
        RTS     PC              ;RETURN TO BASIC TESTS

STATX:  MOV     DLXCSR,CSRA     ;SAVE THE CSR ADDRESS
        MOV     @DLXCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
        RTS     PC              ;RETURN TO THE BASIC TESTS

;THIS ROUTINE IS USED TO CLEAR THE INPUT/OUTPUT BUFFERS
;-----------------------------------------------------

CLDLBF: MOV     #DLBUFO,R0      ;POINT R0 TO BEGINNING OF BUFFERS
1$:     CLR     (R0)+           ;CLEAR ONE WORD - UPDATE POINTER
        CMP     #BUFEND,R0      ;DONE 256. BYTES ??
        BNE     1$              ;BR IF NOT
        RTS     PC              ;RETURN TO CALLING SEGMENT

;THIS ROUTINE LOADS THE OUTPUT BUFFER WITH A NULL-DEL-NULL PATTERN
;----------------------------------------------------------------

LDOUT1: MOV     #DLBUFO,R0      ;SET UP POINTER
1$:     CLRB    (R0)+           ;MOV A NULL CHAR
        MOVB    #377,(R0)+      ;MOV A DELETE CHAR.
        CMP     #DLBUFI,R0      ;BUFFER FULL ???
        BNE     1$              ;BR IF NOT
        RTS     PC              ;RETURN TO CALLING SEGMENT

;THIS ROUTINE IS USED TO LOAD AN ASCENDING BINARY COUNT PATTERN
;--------------------------------------------------------------

LDOUT2: MOV     #DLBUFO,R0      ;SET UP POINTER
        CLR     R1              ;USE R1 TO GENERATE THE PATTERN
1$:     MOVB    R1,(R0)+        ;LOAD ONE CHAR.
        CMP     #DLBUFI,R0      ;BUFFER FULL ???
        BEQ     2$              ;BR IF YES
        INCB    R1              ;GENERATE NEXT CHAR.
        BR      1$              ;GO MOVE IT
2$:     RTS     PC              ;RETURN TO CALLING SEGMENT

;THIS ROUTINE IS USED TO LOAD THE DESCENDING BINARY COUNT PATTERN
```



```
;-------------------------------------------------------------------

LDOUT3: MOV     #DLBUFO,R0      ;SET UP POINTER
        MOV     #377,R1         ;START R1 AT 377
1$:     MOVB    R1,(R0)+        ;LOAD ONE CHAR.
        CMP     #DLBUFI,R0      ;AT END OF THE BUFFER ??
        BEQ     2$              ;BR IF YES
        DECB    R1              ;GENERATE NEXT CHAR.
        BR      1$              ;GO MOVE IT
2$:     RTS     PC              ;RETURN TO CALLING SEGMENT

;THIS ROUTINE LOADS THE WORST CASE PATTERN
;-----------------------------------------

LDOUT4: MOV     #DLBUFO,R0      ;SET UP POINTERS
1$:     MOV     #WCASE,R1       ;POINT TO MONITOR'S WORST CASE PATTERN
2$:     MOV     (R1)+,(R0)+     ;LOAD ONE WORD
        CMP     #DLBUFI,R0      ;BUFFER FULL ???
        BEQ     3$              ;BR IF YES
        CMP     #WCASEE,R1      ;END OF WORST CASE PATTERN ??
        BEQ     1$              ;GO RESET R1
        BR      2$
3$:     RTS     PC              ;RETURN TO CALLING SEGMENT
```

```
                        ;VARIABLES,FLAGS,MESSAGES,AND BUFFERS
                        ;-------------------------------------

003566' 000000          DLRCSR: OPEN            ;CONTAINS ADDRESS OF RCVR CSR
003570' 000000          DLRDBR: OPEN            ;CONTAINS ADDRESS OF RCVR DBR
003572' 000000          DLXCSR: OPEN            ;CONTAINS ADDRESS OF XMITTR CSR
003574' 000000          DLXDBR: OPEN            ;CONTAINS ADDRESS OF XMITTR DBR

003576' 000000          IPTR:   OPEN            ;CONTAINS POINTER TO INPUT BUFFER
003600' 000000          OPTR:   OPEN            ;CONTAINS POINTER TO OUTPUT BUFFER

003602' 000000          XFND:   OPEN            ;FATAL ERROR END FLAGS
003604' 000000          EPCTR:  OPEN            ;END OF PASS COUNTER
003606' 000000          RTRY:   OPEN            ;RETRY FLAG AND COUNTER
003610' 000000          INTFLG: OPEN                    ;SOFTWARE INTR. FLAG USED BY BASIC TESTS

003612' 003440'         LDTAB:  LDOUT1          ;POINTER TO 1ST LOAD BUFFER SUBR.
003614' 003462'                 LDOUT2          ;POINTER TO 2ND LOAD BUFFER ROUTINE
003616' 003506'                 LDOUT3          ;POINTER TO 3RD LOAD BUFFER ROUTINE
003620' 003534'                 LDOUT4          ;POINTER TO 4TH LOAD BUFFER ROUTINE
003622' 003644'         MTAB:   MSG1            ;POINTER TO MESSAGE 1
003624' 003705'                 MSG2            ;POINTER TO MESSAGE 2
003626' 003750'                 MSG3            ;POINTER TO MESSAGE 3
003630' 004015'                 MSG4            ;POINTER TO MESSAGE 4

003632' 000000          LDOUT:  OPEN            ;CONTAINS POINTER TO LOAD BUFFER SUBR.
003634' 003644'         AMESS:  MSG1            ;MESSAGE POINTERS
003636' 177777                  -1                      ;TERMINATOR
003640' 004053'         DRPMS:  MSG5                    ;MESSAGE POINTER
003642' 177777                  -1                      ;TERMINATOR

003644' 047045  046125  026514  MSG1:   .ASCIZ  /%NULL-DEL-NULL SEQUENCE ABORTED%/
003652' 042504  026514  052516
003660' 046114  051440  050505
003666' 042525  041516  020105
003674' 041101  051117  042524
003702' 022504      000
003705'     045 044502  040516  MSG2:   .ASCIZ  /%BINARY UP COUNT SEQUENCE ABORTED%/
003712' 054522  052440  020120
003720' 047503  047125  020124
003726' 042523  052521  047105
003734' 042503  040440  047502
003742' 052122  042105  000045
003750' 041045  047111  051101  MSG3:   .ASCIZ  /%BINARY DOWN COUNT SEQUENCE ABORTED%/
003756' 020131  047504  047127
003764' 041440  052517  052116
003772' 051440  050505  042525
004000' 041516  020105  041101
004006' 051117  042524  022504
004014'     000
004015'     045 047527  051522  MSG4:   .ASCIZ  /%WORST CASE SEQUENCE ABORTED%/
004022' 020124  040503  042523
004030' 051440  050505  042525
004036' 041516  020105  041101
004044' 051117  042524  022504
004052'     000
004053'     045 040506  040524  MSG5:   .ASCIZ  /%FATAL ERROR IN STATIC REGISTER TESTS%/
```



```
004060' 020114  051105  047522
004066' 020122  047111  051440
004074' 040524  044524  020103
004102' 042522  044507  052123
004110' 051105  052040  051505
004116' 051524  000045

                        .EVEN

                        ;512 WORDS RESERVED FOR TWO 256. BYTE BUFFERS
                        ;--------------------------------------------

004122' 000400          DLBUFO: .BLKB   256.    ;RSVD FOR OUTPUT BUFFER
004522' 000400          DLBUFI: .BLKB   256.    ;RSVD FOR INPUT BUFFER
005122' 000000          BUFEND: 0               ;MARK END OF BUFFER AREA

005124' 177776  000001  177775  WCASE:  .WORD   177776,1,177775,2,177773,4,177767,10
005132' 000002  177773  000004
005140' 177767  000010
005144' 177757  000020  177737          .WORD   177757,20,177737,40,177677,100,177577,200
005152' 000040  177677  000100
005160' 177577  000200
005164' 177377  000400  176777          .WORD   177377,400,176777,1000,175777,2000,173777,4000
005172' 001000  175777  002000
005200' 173777  004000
005204' 167777  010000  157777          .WORD   167777,10000,157777,20000,137777,40000,77777,100000
005212' 020000  137777  040000
005220' 077777  100000
005224'                 WCASE8:

000001                  .END
```

```
ACSR    000102R   389#   582*   824*   868*   896*   974*   978*
A0DR    000006R   355#   409
A0DR22= 001000    407#
AMESS   003634R   786*   927    1063#
ASR     000106R   393#   583*   955*   956*
ASTAT   000104R   391#   847*
AWAS    000110R   394#   563*   579    958*   959*
BEGIN   000000R   352#   437    439    440    450    452    453    464    466    467    474    476    477
                  490    491    499    501    502    511    513    514    529    531    537    539    541
                  542    548    549    559    561    562    575    577    578    587    607    614    622
                  634    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    878    881    892    897    901    910    911    914    927    963
BIT0  = 000001    407#
BIT1  = 000002    407#
BIT10 = 002000    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   357#   486    522    778    781
BR2     000013R   358#
BTOD$ = 104421    407#
BUFEND  005122R   856    922    950    986    1106#
CDATA$= 104412    407#
CHKDAT  003252R   929    944#
CLDLBF  003422R   905    944#
CONFIG  000056R   377#
CSRA    000100R   387#   581*   823*   867*   885*   954*   973*   977*
DATCK$= 104411    407#
DATERR$= 104404   407#   587    963
DLBUFF  004522R   812    936    947    996    1007   1019   1031   1105#
DLBUFO  004122R   935    946    984    993    1004   1016   1028   1104#
DLINIT  000256R   373#   410*   423*   424*   432    523*   524    533*   534    543*   555*   564*   568    571*
DLRCSR  003566R   581    582    601*   602    609    610*   617    626*   629    636    643*   644    651
                  662*   663*   664    671    678*   679    686    697*   698    705*   706    722*   723*
                  732*   738*   741    775*   833*   846    867    868    869*   880*   885    886    888*
                  938*   945*   954    973    974    1040#  785    786    851    858    887    1041#
DLRDBP  003570R   412*   426    427    563    584    785    786    851    858    887    1041#
DLT01   000302R   432#
DLT02   000342R   433    445#
DLT03   000404R   446    458#
DLT04   000524R   470    482#
DLT05   000712R   507    519#
DLT06   001344R   595    601#
```



```
DLT07   001466R   618    628#
DLT10   001634R   652    652#
DLT11   001776R   687    697#
DLT12   002072R   707    718#
DLT13   002240R   713    717    742    752#
DLXCSR  003572R   414*   425*   445    458*   459    468*   469    488*   503*   506    544*   556*   565*
                  572*   752*   753    760*   761    774*   807    823    824    825*   836*   877*   895*
                  937*   944*   977    978    1042#
DLXDBR  003574R   416*   546*   585    814*   1043#
DOACIN  002376R   772#   800
DRPMS   003640R   452    466    476    501    513    531    541    561    577    1065#
DVIDI   000014R   359#
ENDITS= 104413    407#   798
END$  = 104410    407#   440    453    467    477    502    514    532    542    562    578    834    878
EPCTR   003604R   1049#
ERRTYP  000106R   392#   435*   448*   462*   472*   497*   509*   527*   537*   557*   573*   605*   612*
                  620*   632*   639*   647*   654*   667*   674*   682*   689*   701*   709*   733*   744*
                  756*   764*   828*   872*   890*
EXITS = 104400    407#   837    841    891    897    901    914
GETPA$= 104415    407#
GWBUF$= 104414    407#
HRDCNT  000044R   372#
HRDERR$= 104405   407#   437    450    464    474    499    511    529    539    559    575    607    614
                  622    634    641    649    656    669    676    684    691    703    711    735    746
                  758    766    830    874    892
HRDPAS  000050R   374#
ICONT   000036R   369#
ICONT   000040R   370#
IDNUM   000122R   399#
INIT    000030R   366#
INTPLC  003610R   492#   492    504*   519*   550    566*   718*   727    739*   1051#
INTP    000120R   398#
IPTR    003576R   856    858*   859*   922    936*   1045#
KICKOF  003220R   907    935#
LDOUT   003632R   787*   906    1062#
LDOUT1  003440R   993#   1053
LDOUT2  003462R   1004#  1054
LDOUT3  003506R   1016#  1055
LDOUT4  003514R   1028#  1056
LDTAB   003612R   782    1053#
MAP22$= 104416    407#
MODEM   001332R   580    594#
MODNAM  000000R   353#
MODSP   000224R   367    405#
MSG4$ = 104403    407#   439    452    466    476    501    513    531    541    561    577    927
MSG5$ = 104402    407#
MSG5  = 104401    407#
MSG1    003644R   1057   1063   1067#
MSG2    003705R   1058   1073#
MSG3    003750R   1059   1079#
MSG4    004015R   1060   1086#
MSG5    004053R   1065   1092#
NTAB    003622R   783    790    1057#
NULL  = 000000    407#   437    450    464    474    499    511    529    539    559    575    607    614
                  622    634    641    649    656    669    676    684    691    703    711    735    746
                  758    766    830    874    892
```

```
OPEN  = 000000            354     360     361     362     363     380     381     382     383     384     385     386     387
                          389     391     393     394     396     397     398    407#    1040    1041    1042    1043    1045
                         1046    1048    1049    1050    1051    1062
OPTR    003600R           812     814     815*    935*   1046#
OTOAS = 104420            407#
PASCNT  000034R           368#
PIRQS = 000004            407#    810     818     849     854     862
POPSP = 005726            407#
POPSP2= 022626            407#
PRTY  = 000000            358     407#
PRTY0 = 000000            407#
PRTY1 = 000040            407#
PRTY2 = 000100            407#
PRTY3 = 000140            407#
PRTY4 = 000200            357     407#
PRTY5 = 000240            407#
PRTY6 = 000300            407#
PRTY7 = 000340            407#
PS    = 177776            407#
PSW   = 177776            407#
PUSH  = 005746            407#
PUSH2 = 024646            407#
RANDS = 104417            407#
RANNUM  000054R           376#
RESTRT  002334R           395     596     762     772#
RES1    000056R           378#
RES2    000060R           379#
RINT    002622R           777     846#
RSTRT   000112R           395#
RTRY    003606R           784*    896*    915     917*    918*    919    1050#
SBADR   000102R           388#    585*    957*
SEGT    003074R           789     904#
SOFCNT  000042R           371#
SOFERS= 104406            407#
SOFPAS  000046R           373#
SPOINT  000032R           367#
SPSIZ = 000040              1#    400
SR1     000016R           360#    594
SR2     000020R           361#
SR3     000022R           362#
SR4     000024R           363#
START   000224R           366     409#
STAT    000026R           365#
STATR   003366R           434     526     536     554     570     604     611     619     631     638     646     653     666
                          673     681     688     700     708     731     743    973#
STATX   003404R           447     461     471     496     508     755     763    977#
SVR0    000062R           380#
SVR1    000064R           381#
SVR2    000066R           382#
SVR3    000070R           383#
SVR4    000072R           384#
SVR5    000074R           385#
SVR6    000076R           386#
SVSCNT  000052R           375#
TRPDFD= 000022            407#
VECTOR  000010R           356#    483     520     719     776
```



```
WASADR  000104R           390#    584*    960*
WCASE   005124R          1029    1109#
WCASEF  005224R          1033    1121#
WDER    000116R           397#
WDTO    000114R           396#
XEND    003602R           773*    821     826*    839     865     870*    883     899     912    1048#
XFLAG   000056R           354#
XINT    002452R           780     807#
.       = 005224R          1104#   1105#

. ABS.  000000      000
        005224      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDLRB0,XDLRB0/SOL/CRF:SYM=DDXCOM,XDLRB0
 RUN-TIME:  2  3.3 SECONDS
 RUN-TIME RATIO: 31/6=4.9
 CORE USED:  7K  (13 PAGES)
```

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐**

☐ NEW FOR RELEASE ENG. USE ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	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 | MAINTAIN GROUP SPT GRP | MAINTAINER D.BUTENHOF | SUBMITTING ENGINEER D.BUTENHOF

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 OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

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 INFORMATION FIELD		EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
				☒ Y ☐ N	☒ Y ☐ N	40 SECONDS	40 SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

DEVICE AFFECTED DEC/X11 | MULTIMEDIA AFFECTED? ☒ YES ☐ NO

KIT NUMBERS							
	ZJ129-RZ	ZJ129-FR	ZJ13Ø-RB				

PROBLEM:

Module is intented for 2Ø4Ø 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	MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
DATE: 23-jan-79	DATE: 1-Feb-79	DATE:	Q99-05460
MAINTAIN ER	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MCF2838
23-jan-79	DATE:	DATE:	