

Micro Fiche Scan

Name of device(s) tested:

DEC/X11,DL11-E,DLV11-E

Test description:

DL11-E, DLV11-E MOD

MAINDEC Number or Package Identifier (after SEP 1977):

CXDLBC0

Fiche Document Part Number:

AH-F000C-MC

Fiche preparation date unknown, using copyright year:

1985

Image resolution:

1-bit black&white, compressed for minimal file size

COPYRIGHT (C) 1976-85 by d|i|g|i|t|a|l

.REM

IDENTIFICATION  
-----

PRODUCT CODE: AC-E998C-MC  
PRODUCT NAME: CXDLBCO DL11-E,DLV11-E MODULE  
PRODUCT DATE: JULY 1985  
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) 1976,1980,1985 DIGITAL EQUIPMENT CORPORATION

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

## 1.0 ABSTRACT

-----

CXDLB IS AN IOMOD THAT EXERCISES ONE DL11-E (OR DLV11-E) ASYNCHRONOUS COMMUNICATIONS INTERFACE (M7800). THE PROGRAM CONSISTS OF TWO MAJOR SECTIONS AS DESCRIBED BELOW:

## SECTION ONE:

-----

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

## SECTION TWO:

-----

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

A. NULL-DEL 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 UNIBUS PDP 11 COMPUTER WITH A DL11-E INTERFACE  
OR A Q-BUS PDP-11 COMPUTER WITH A DLV11-E INTERFACE

STORAGE: DLB REQUIRES:

1. DECIMAL WORDS: 1356
2. OCTAL WORDS: 2514
3. OCTAL BYTES: 5230

## 3.0 PASS DEFINITION

-----

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

## 4.0 EXECUTION TIME

-----

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

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      SR2: 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.
- SR2: THIS LOCATION CAN BE LOADED WITH A VALUE TO SET THE BAUD RATE FOR A DLV11E THAT IS WIRED FOR A PROGRAMMABLE BAUD RATE THESE VALUES ARE SHOWN IN THE PERIPHERAL HANDBOOK. SOME EXAMPLES ARE:

00000	=	9600 BAUD	004000	=	0050 BAUD
02400	=	0110 BAUD	044000	=	0300 BAUD
07400	=	1200 BAUD	124000	=	2400 BAUD
14400	=	4800 BAUD	174000	=	19200 BAUD
16400	=	ALSO GIVES 9600 BAUD			

## 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.

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

- 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:

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

- 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-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

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

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

KICKOF: THIS SUBROUTINE IS CALLED FROM "SEGX" AND CONTAINS THE THE CODE TO ENABLE INTERRUPTS AND INITIAIE 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.

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

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

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

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

LDOUT4: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH THE 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 "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)



## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

:DL11-E DEC/X11 EXERCISER MODULE

```

000000      IOMOD <DLBC >,175610,300,4,,,2,,161
000000      MODULE 140000,DLBC ,175610,300,4,,,2,,161,,,,
          .TITLE DLBC DEC/X11 SYSTEM EXERCISER MODULE
          ;      DDXCOM VERSION 6.4   28 JAN-82
          .LIST  BIN
          ;*****
000000      BEGIN:
000000      104      114      102 MODNAM: .ASCII /DLBC / ;MODULE NAME.
000003      103      040
000005      000      XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006      175610     ADDR: 175610+0 ;1ST DEVICE ADDR.
000010      000300     VECTOR: 300+0 ;1ST DEVICE VECTOR.
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      000226'     INIT: START ;MODULE START ADDR.
000032      000224'     SPOINT: MODSP ;MODULE STACK POINTER.
000034      000000     PASCNT: 0 ;PASS COUNTER.
000036      000002     ICONT: 2. ;# OF ITERATIONS PER PASS=2.
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      000000     CONFIG: ;RESERVED FOR MONITOR USE
000056      000000     RES1: 0 ;RESERVED FOR MONITOR USE
000060      000000     PES2: 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      000000     SBADR: ;ADDR OF GOOD DATA, OR
000102      000000     ACSR: OPEN ;CONTENTS OF CSR.
000104      000000     WASADR: ;ADDR OF BAD DATA, OR
000104      000000     ASTAT: OPEN ;STATUS REG CONTENTS.
000106      000000     ERRTP: ;TYPE OF ERROR
000106      000000     ASB: OPEN ;EXPECTED DATA.
000110      000000     AWAS: OPEN ;ACTUAL DATA.
000112      002406'     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

```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

000040

.REPT SPSIZ  
.NLIST  
.WORD 0  
.LIST  
.ENDR

;MODULE STACK STARTS HERE.

000224

MODSP:

\*\*\*\*\*

368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379

000224 000020

;\*  
; THE NEXT LOCATION CONTAINS A NUMBER THAT CAN BE EXAMINED IN CORE TO BE  
; SURE THAT THE LISTING AND ACTUAL CODE ARE THE SAME. THIS LOCATION IS  
; TO BE UPDATED EACH TIME THE SOURCE CODE IS UPDATED. LOCATION 'MODREV'  
; IS NOT USED BY THE PROGRAM.

MODREV: 1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1

\*\*\*\*\*

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

381
382 000226 016700 177554      START:  MOV      ADDR,RO      ;GET BASE DEVICE ADDRESS
383 000232 010067 003402      MOV      RO,DLRCSR      ;SET UP RCVR CSR ADDRESS
384 000236 005720              TST      (RO)+
385 000240 010067 003376      MOV      RO,DLRDBR      ;SET UP RCVR DBR ADDRESS
386 000244 005720              TST      (RO)+
387 000246 010067 003372      MOV      RO,DLXCSR      ;SET UP XMITTR CSR ADDRESS
388 000252 005720              TST      (RO)+
389 000254 010067 003366      MOV      RO,DLXDBR      ;SET UP XMITTR DBR ADDRESS
390
391
392                               ;
393                               ;*****
394                               ; * SECTION ONE *
395                               ;*****
396 000260 005077 003354      DLINIT: CLR      @DLRCSR      ;CLEAR OUT BOTH CSR'S
397 000264 005077 003350      CLR      @DLRCSR      ;MAKE SURE ALL DATA COMM BITS CLEARED
398 000270 005077 003350      CLR      @DLXCSR
399 000274 005777 003342      TST      @DLRDBR      ;FLUSH RCVR DONE BIT
400 000300 005777 003336      TST      @DLRDBR
401
402
403                               ;IN CASE IT'S PROGRAMMABLE, LOAD DLXCSR WITH DESIRED BAUD RATE FROM SR2.
404                               ;IF SR2 IS ZERO THEN LOAD BAUD RATE OF 9600
405                               ;-----
406
407 000304 012700 164000              MOV      #164000,RO      ;SETUP BAUD RATE                ;RG01
408 000310 032767 004000 177502      BIT      #BIT11,SR2      ;BAUD RATE LOADED?            ;RG01
409 000316 001404              BEQ      1$              ;BRANCH IF NOT                ;RG01
410 000320 016700 177474              MOV      SR2,RO          ;GET RATE                      ;RG01
411 000324 042700 003777              BIC      #3777,RO        ;CLR UNWANTED BITS            ;RG01
412 000330 010077 003310      1$:  MOV      RO,@DLXCSR      ;LOAD IT                      ;RG01
413
414
415                               ;THIS TEST VERIFIES THAT RCVR CSR GOT CLEARED UPON ENTRY
416                               ;-----
417
418 000334 005777 003300      DLT01: TST      @DLRCSR      ;IS RCVR CSR ALL ZEROES ??
419 000340 001415              BEQ      DLT02           ;BR IF YES
420 000342 004767 003072              JSR      PC,STATR        ;GO SET UP ERROR INFO
421 000346 012767 000025 177532      MOV      #25,ERRTYP      ;
422                               ;*****
423                               ;HRDR$,BEGIN,NULL          ;CANT'T CLEAR OUT RCVR CSR
424                               ;*****
425                               ;MSGN$,BEGIN,DRPMS        ;ASCII MESSAGE CALL WITH COMMON HEADER
426                               ;END$,BEGIN
427
428                               ;TEST THAT READY BIT IS ONLY BIT SET IN XMITTR CSR
429                               ;-----
430 000374 022777 000200 003242      DLT02: CMP      #200,@DLXCSR  ;READY SET ??
431 000402 001415              BEQ      DLT03           ;BR IF YES
432 000404 004767 003046              JSR      PC,STATX        ;GO SET UP ERROR INFO
433 000410 012767 000025 177470      MOV      #25,ERRTYP      ;
434                               ;*****
435                               ;HRDR$,BEGIN,NULL          ;READY NOT SET OR OTHER BITS DIDN'T CLEAR IN XMIT CSR
436                               ;*****

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

434 000424 104403 000000' 003712'      MSGN$,BEGIN,DRPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
435 000432 104410 000000'      END$,BEGIN            ;
436
437      ;TEST THAT MAINT. BIT CAN BE SET AND CLEARED IN XMIT CSR
438      ;-----
439
440 000436 052777 000004 003200 DLT03: BIS      #4,@DLXCSR      ;SET THE MAINT. BIT
441 000444 022777 000204 003172      CMP      #204,@DLXCSR  ;DID IT SET ??
442 000452 001415          BEQ      1$           ;BR IF YES
443 000454 004767 002776          JSR      PC,STATX     ;GO SET UP ERROR INFO
444 000460 012767 000033 177420      MOV      #33,ERRTYP
445      ;*****
      HRDR$,BEGIN,NULL      ;MAINT BIT WON'T SET OR IT CLEARED READY
      ;*****
446 000474 104403 000000' 003712'      MSGN$,BEGIN,DRPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
447 000502 104410 000000'      END$,BEGIN            ;
448 000506 042777 000004 003130 1$: BIC      #4,@DLXCSR      ;NOW CLR THE MAINT BIT
449 000514 022777 000200 003122      CMP      #200,@DLXCSR  ;DID IT CLEAR ??
450 000522 001415          BEQ      DLT04         ;BR IF YES
451 000524 004767 002726          JSR      PC,STATX     ;GO SET UP ERROR INFO
452 000530 012767 000025 177350      MOV      #25,ERRTYP
453      ;*****
      HRDR$,BEGIN,NULL      ;MAINT. BIT WON'T CLEAR OR READY CLEARED
      ;*****
454 000544 104403 000000' 003712'      MSGN$,BEGIN,DRPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
455 000552 104410 000000'      END$,BEGIN            ;
456
457      ;TEST THAT BIT 06 IN XCSR CAN CAUSE AN INTERRUPT
458      ;-----
459
460 000556 005067 003100          DLT04: CLR      INTFLG      ;INIT THE SOFTWARE INTR. FLAG
461 000562 016700 177222          MOV      VECTOR,R0     ;GET BASE VECTOR ADDRESS
462 000566 062700 000004          ADD      #4,R0         ;GENERATE ADDR OF XMIT VECTOR
463 000572 012720 000666'        MOV      #2$(R0)+      ;GO TO 2$ ON XMIT INTERRUPT
464 000576 116710 177210          MOVB     BR1,(R0)      ;PRIORITY LEVEL = BR1
465 000602 005001          CLR      R1           ;INIT BREAK T. 1ER
466 000604 052777 000100 003032      BIS      #100,@DLXCSR  ;SET INTR. ENAB
467 000612          1$:
      000612 104407 000000'        BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
      000616 104407 000000'        BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
468 000622 005767 003034          TST      INTFLG       ;DID XMIT INTR OCCUR YET ??
469 000626 001025          BNE      3$          ;BR IF IT DID
470 000630 005301          DEC      R1          ;COUNT BREAK TIMER
471 000632 001367          BNE      1$          ;BR IF NO TIMEOUT
472 000634 004767 002616          JSR      PC,STATX     ;GO SET UP ERROR INFO
473 000640 012767 000023 177240      MOV      #23,ERRTYP
474      ;*****
      000646 104405 000000' 000000      HRDR$,BEGIN,NULL      ;XMITTR FAILED TO GENERATE INTERRUPT
      ;*****
475 000654 104403 000000' 003712'      MSGN$,BEGIN,DRPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
476 000662 104410 000000'      END$,BEGIN            ;
477 000666 042777 000100 002750 2$: BIC      #100,@DLXCSR  ;DISABLE XMITTR INTR ENABLE
478 000674 005167 002762          COM      INTFLG       ;SET THE INTR. FLAG
479 000700 000002          RTI                    ;RETURN CONTROL TO OTHER GUY
480 000702 022777 000200 002734 3$: CMP      #200,@DLXCSR  ;DID I.E. GET CLEARED IN INTR. SERVICE
481 000710 001415          BEQ      DLT05         ;BR IF YES
482 000712 004767 002540          JSR      PC,STATX     ;GO SET UP ERROR INFO

```

## DEC/X11 SYS .1 EXERSIZER MACRO DEFINITION MODULE

```

483 000716 012767 000027 177162      MOV      #27,ERRTYP
484                                     ;*****
000724 104405 000000' 000000      HRDR$,BEGIN,NULL      ;"DONE" OR RCVR INTR. ENAB FAILED TO CLEAR
                                     ;*****
485 000732 104403 000000' 003712'    MSGN$,BEGIN,DRPMS     ;ASC.I MESSAGE CALL WITH COMMON HEADER
486 000740 104410 000000'          END$,BEGIN            ;
487
488                                     ;TEST THAT A RCVR INTR CAN OCCUR WHEN "DONE" GETS SET
489                                     ;-----
490
491 000744 005067 002712          DLT05: CLR      INTFLG      ;INIT SOFTWARE INTR. FLAG
492 000750 016700 177034          MOV      VECTOR,RO    ;GET THE BASE VECTOR ADDRESS
493 000754 012720 001214'        MOV      #4,(RO)+     ;GO TO 4$ ON RCVR INTERRUPT
494 000760 116710 177026          MOVB     BR1,(RO)     ;SET PRIORITY
495 000764 052777 000100 002646    BIS      #100,@DLRCSR ;SET I.E. IN RCVR CSR
496 000772 032777 000100 002640    BIT      #100,@DLRCSR ;DID IT SET
497 001000 001015          BNE     1$           ;BR IF IT DID
498 001002 004767 002432          JSR      PC,STATR    ;GO SET UP ERROR INFO
499 001006 012767 000033 177072    MOV      #33,ERRTYP
500                                     ;*****
001014 104405 000000' 000000      HRDR$,BEGIN,NULL      ;CAN'T SET BIT 06 IN RCSR I.E.
                                     ;*****
501 001022 104403 000000' 003712'    MSGN$,BEGIN,DRPMS     ;ASCII MESSAGE CALL WITH COMMON HEADER
502 001030 104410 000000'          END$,BEGIN            ;
503 001034 042777 000100 002576 1$: BIC      #100,@DLRCSR ;NOW CLEAR THE I.E. BIT
504 001042 032777 000100 002570    BIT      #100,@DLRCSR ;DID I.E. BIT GET CLEARED ??
505 001050 001415          BEQ     2$           ;BR IF YES
506 001052 004767 002362          JSR      PC,STATR    ;GO SET UP ERROR INFO
507 001056 012767 000023 177022    MOV      #23,ERRTYP
508                                     ;*****
001064 104405 000000' 000000      HRDR$,BEGIN,NULL      ;CAN'T CLEAR RCSR INTR. ENAB. BIT
                                     ;*****
509 001072 104403 000000' 003712'    MSGN$,BEGIN,DRPMS     ;ASCII MESSAGE CALL WITH COMMON HEADER
510 001100 104410 000000'          END$,BEGIN            ;
511 001104 052777 000100 002526 2$: BIS      #100,@DLRCSR ;NOW TURN IT ON FOR REAL
512 001112 052777 000004 002524    BIS      #4,@DLXCSR   ;TURN ON MAINT. MODE
513 001120 005001          CLR     R1           ;INIT BREAK TIMER
514 001122 112777 000252 002516    MOVB     #252,@DLXDBR ;LOAD THE XMITTR OUTPUT DATA BUFFER
515 001130          3$: BREAK$,BEGIN      ;TEMPORARY RETURN TO MONITOR....
001130 104407 000000'          BREAK$,BEGIN        ;THEN CONTINUE AT NEXT INSTRUCTION.
001134 104407 000000'          TST      INTFLG     ;DID RCVR INTR. YET ??
516 001140 005767 002516          BNE     5$           ;BR IF IT DID
517 001144 001036          DEC     R1           ;COUNT BREAK TIMER
518 001146 005301          BNE     3$           ;BR IF NO TIMEOUT
519 001150 001367          JSR      PC,STATR    ;GO SET UP ERROR INFO
520 001152 004767 002262          CLR     @DLRCSR     ;CLEAR BOTH CSR'S
521 001156 005077 002456          CLR     @DLXCSR
522 001162 005077 002456          MOV      #23,ERRTYP
523 001166 012767 000023 176712    ;*****
524 001174 104405 000000' 000000      HRDR$,BEGIN,NULL      ;RCVR FAILED TO INTR ON TIME
                                     ;*****
525 001202 104403 000000' 003712'    MSGN$,BEGIN,DRPMS     ;ASCII MESSAGE CALL WITH COMMON HEADER
526 001210 104410 000000'          END$,BEGIN            ;
527 001214 117767 002422 176666 4$: MOVB     @DLRDBR,AWAS ;GET THE RECEIVED DATA
528 001222 042777 000100 002410    BIC      #100,@DLRCSR ;TURN OFF I.E.
529 001230 005077 002410          CLR     @DLXCSR     ;TURN OFF MAINTENANCE MODE

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

530 001234 005167 002422          COM      INTFLG          ;SET SOFTWARE INTR. FLAG
531 001240 000002                    RTI              ;RETURN TO OTHER GUY
532 001242 005777 002372          5:      IST      @DLRCSR      ;DID INTR SERVICE CLEAR THE RCVR CSR ??
533 001246 001421                    BEQ        6:          ;BR IF IT DID
534 001250 004767 002164          JSR      PC,STATR      ;GO SET UP ERROR INFO
535 001254 005077 002360          CLR      @DLRCSR      ;CLEAR BOTH CSR'S
536 001260 005077 002360          CLR      @DLXCSR
537 001264 012767 000025 176614    MOV      #25,ERRTYF
538                                ;*****
                                HRDR$,BEGIN,NULL      ;RCVR INTR SERVICE FAILED TO CLEAR I.E. AND DONE
                                ;*****
539 001300 104403 000000' 003712'  MSGN$,BEGIN,DRPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
540 001306 104410 000000'                    END$,BEGIN              ;
541 001312 122767 000252 176570    6:      CMPB      #252,AWAS      ;WAS DATA CORRECT ???
542 001320 001421                    BEQ        MODEM        ;BR IF YES
543 001322 016767 002312 176550    MOV      DLRCSR,CSRA    ;SET UP ERROR INFO
544 001330 017767 002304 176544    MOV      @DLRCSR,ACSR
545 001336 012767 000252 176542    MOV      #252,ASB
546 001344 016767 002272 176532    MOV      DLRDBR,WASADR
547 001352 016767 002270 176522    MOV      DLXDBR,SBADR
548                                ;*****
                                DATER$,BEGIN          ;DATA ERROR!!!
                                ;*****

549
550                                ;THE FOLLOWING GROUP OF BASIC TESTS ASSUME THAT THE MODEM IS DISCONNECT-
551                                ;ED FROM THE DL11-E AND AN H315 MODEM TEST CONNECTOR IS CONNECTED IN ITS
552                                ;PLACE. ALL OF THESE TESTS ARE SKIPPED IF SR1(15) IS CLEARED.
553
554 001364 005767 176426          MODEM:  TST      SR1          ;IS SR1 BIT 15 = 1 ??
555 001370 100402                    BMI      DLT06          ;DO MODEM TESTS IF SET
556 001372 000167 001010          JMP      RESTR          ;SKIP MODEM TESTS IF SR1(15)=0

557
558                                ;TEST THAT "REQ TO SEND" CAN ASSERT "RING"
559                                ;-----
560
561 001376 052777 000004 002234    DLT06:  BIS      #4,@DLRCSR      ;SET REQ TO SEND
562 001404 032777 000004 002:26    BIT      #4,@DLRCSR      ;DID IT SET ??
563 001412 001010                    BNE      1:          ;BR IF YES
564 001414 004767 002020          JSR      PC,STATR      ;GO SET UP ERROR INFO
565 001420 012767 000025 176460    MOV      #25,ERRTYF
566                                ;*****
                                HRDR$,BEGIN,NULL      ;CAN'T SET REQ TO SEND
                                ;*****
567 001434 032777 040000 002176    1:      BIT      #40000,@DLRCSR      ;DID "RING" GET ASSERTED ??
568 001442 001010                    BNE      2:          ;BR IF YES
569 001444 004767 001770          JSR      PC,STATR      ;GO SET UP ERROR INFO
570 001450 012767 000025 176430    MOV      #25,ERRTYF
571                                ;*****
                                HRDR$,BEGIN,NULL      ;"REQ TO SEND" DIDN'T ASSERT "RING"
                                ;*****
572 001464 042777 000004 002146    2:      BIC      #4,@DLRCSR      ;TURN OFF "REQ TO SEND"
573                                ;
574                                ; ** REQUEST FROM MRO FOR DL11 LINE FILTER DELAY **
575                                ;
576 001472 012705 000005          3:      MOV      #5,R5          ;SET UP A COUNTER FOR LINE FILTER DELAY ;RG002
577 001476 005305                    DEC      R5              ;COUNT IT DOWN... ;RG002
578 001500 001376                    BNE      3:          ;BRANCH IF NOT DONE ;RG002

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

579
580 001502 005777 002132          TST    @DLRCSR          ;ARE ALL BITS NOW CLEAR ??
581 001506 001410                  BEQ    DLT07            ;BR IF BOTH "RING" AND "REQ TO SEND " CLEARED
582 001510 004767 0C1724          JSR    PC,STATR        ;GO SET UP ERROR INFO
583 001514 012767 000025 176364  MOV    #25,ERRTYP
584                                ;*****
001522 104405 000000' 000000  HRDR$,BEGIN,NULL      ;"RING" OR "REQ TO SEND" FAILED TO CLEAR
                                ;*****

585                                ;TEST THAT "SEC XMIT" ASSERTS "SEC REC" WHICH SETS "DATA SET INT"
586                                ;-----
587
588
589 001530 052777 000010 002102  DLT07: BIS    #10,@DLRCSR    ;SET SEC XMIT
590 001536 005777 002076          TST    @DLRCSR        ;DID DATA SET INT GET SET ??
591 001542 100410                  BMI    1$             ;BR IF YES
592 001544 004767 001670          JSR    PC,STATR        ;GO SET UP ERROR INFO
593 001550 012767 000025 176330  MOV    #25,ERRTYP
594                                ;*****
001556 104405 000000' 000000  HRDR$,BEGIN,NULL      ;DATA SET INT FAILED TO SET
                                ;*****
595 001564 022777 002010 002046  1$:  CMP    #2010,@DLRCSR   ;SEC XMIT AND REC SET - DATA SET INT CLEAR
596 001572 001410                  BEQ    2$             ;BR IF YES
597 001574 004767 001640          JSR    PC,STATR        ;GO SET UP ERROR INFO
598 0C1600 012767 000025 176300  MOV    #25,ERRTYP
599                                ;*****
001606 104405 000000' 000000  HRDR$,BEGIN,NULL      ;SEC XMIT OR SEC REC FAILED TO SET - DATA SET INT FAILED TO
CLEAR
                                ;*****
600 001614 042777 000010 002016  2$:  BIC    #10,@DLRCSR    ;TURN OFF SEC XMIT
601                                ;
602                                ; ** REQUEST FROM MRO FOR DL11 LINE FILTER DELAY **
603                                ;
604 001622 012705 000005          MOV    #5,R5          ;SET UP A COUNTER FOR LINE FILTER DELAY ;RG002
605 001626 005305                  DEC    R5              ;COUNT IT DOWN... ;RG002
606 001630 001376                  BNE    3$             ;BRANCH IF NOT DONE ;RG002
607
608 001632 005777 002002          TST    @DLRCSR        ;DID SEC XMIT GOING OFF SET DATA SET INT ?
609 001636 100410                  BMI    4$             ;BR IF YES
610 001640 004767 001574          JSR    PC,STATR        ;GO SET UP ERROR INFO
611 001644 012767 000020 176234  MOV    #20,ERRTYP
612                                ;*****
001652 104405 000000' 000000  HRDR$,BEGIN,NULL      ;CLEARING SEC XMIT FAILED TO SET DATA SET INT
                                ;*****
613 001660 005777 001754          4$:  TST    @DLRCSR        ;ALL BITS NOW CLEAR
614 001664 001410                  BEQ    DLT10          ;BR IF YES
615 001666 004767 001546          JSR    PC,STATR        ;GO SET UP ERROR INFO
616 001672 012767 000020 176206  MOV    #20,ERRTYP
617                                ;*****
001700 104405 000000' 000000  HRDR$,BEGIN,NULL      ;SEC XMIT,SEC REC, OR DATA SET INT FAILED TO CLEAR
                                ;*****

618                                ;TEST THAT "DTR" ASSERTS "CAR DET", "CLR TO SEND", AND "DATA SET INT"
619                                ;-----
620
621
622 001706 005077 001726          DLT10: CLR    @DLRCSR    ;CLR THE RCVR CSR
623 001712 052777 000002 001720  BIS    #2,@DLRCSR    ;SET DATA TERM READY
624 001720 005777 001714          TST    @DLRCSR        ;DID DATA SET INT SET ??
625 001724 100407                  BMI    1$             ;BR IF YES

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

626 001726 004767 001506      JSR    PC,STATR      ;GO SET UP ERROR INFO
627 001732 005067 176150      CLR    ERRTP
628                                ;*****
                                HRDR$,BEGIN,NULL      ;"DTR" FAILED TO SET "DATA SET INT"
                                ;*****
629 001744 022777 030002 001666 1$: CMP    #30002,#DLRCSR;DTR,CLR TO SEND, CAR DET, SET AND DATA SET INT CLEAR ?
630 001752 001407                                BEQ    2$              ;BR IF YES
631 001754 004767 001460      JSR    PC,STATR      ;GO SET UP ERROR INFO
632 001760 005067 176122      CLR    ERRTP
633                                ;*****
                                HRDR$,BEGIN,NULL      ;DTR,CAR DET, OR CLR TO SEND FAILED TO SET OR DATA SET INT F
AILFD TO CLR
                                ;*****
634 001772 042777 000002 001640 2$: BIC    #2,#DLRCSR      ;TURN OFF DTR
635 002000 005777 001634      TST    #DLRCSR        ;DATA SET INT SHOULD HAVE SET
636 002004 100407                                BMI    3$              ;BR IF IT DID
637 002006 004767 001426      JSR    PC,STATR      ;GO SET UP ERROR INFO
638 002012 005067 176070      CLR    ERRTP
639                                ;*****
                                HRDR$,BEGIN,NULL      ;DATA SET INT FAILED TO SET ON CLEARING DTR
                                ;*****
640 002024 005777 001610      3$: TST    #DLRCSR        ;ALL BITS NOW CLEAR ??
641 002030 001407                                BEQ    DLT11           ;BR IF YES
642 002032 004767 001402      JSR    PC,STATR      ;GO SET UP ERROR INFO
643 002036 005067 176044      CLR    ERRTP
644                                ;*****
                                HRDR$,BEGIN,NULL      ;CLEARING DTR FAILED TO CLEAR OTHER BITS
                                ;*****
645                                ;TEST THAT DATA SET INTR ENAB CAN SET AND CLEAR
646                                ;-----
647
648
649 002050 052777 000040 001562 DLT11: BIS    #40,#DLRCSR      ;SET DATA SET I.E.
650 002056 032777 000040 001554 BIT    #40,#DLRCSR      ;DID IT SET ??
651 002064 001010                                BNE    1$              ;BR IF YES
652 002066 004767 001346      JSR    PC,STATR      ;GO SET UP ERROR INFO
653 002072 012767 000027 176006 MOV    #27,ERRTP
654                                ;*****
                                HRDR$,BEGIN,NULL      ;CAN'T SET DATA SET INTR.
                                ;*****
655 002106 042777 000040 001524 1$: BIC    #40,#DLRCSR      ;CLEAR DATA SET I.E.
656 002114 005777 001520      TST    #DLRCSR        ;DID IT CLEAR ??
657 002120 001411                                BEQ    DLT12           ;BR IF YES
658 002122 004767 001312      JSR    PC,STATR      ;GO SET UP ERROR INFO
659 002126 012767 000027 175752 MOV    #27,ERRTP
660                                ;*****
                                HRDR$,BEGIN,NULL      ;CAN'T CLEAR DATA SET I.E.
                                ;*****
661 002142 000463                                BR     DLT13           ;SKIP NEXT TEST
662
663                                ;TEST THAT DATA SET INT CAN CAUSE A RCVR INTERRUPT WHEN ENABLED
664                                ;-----
665
666 002144 005067 001512      DLT12: CLR    INTFLG        ;INIT SOFTWARE INTR. FLAG
667 002150 016700 175634      MOV    VECTOR,R0      ;GET BAS VECTOR ADDR.
668 002154 012720 002246      MOV    #2#,(R0)+      ;GO TO 2$ ON DATA SET INTERRUPT
669 002160 005001                                CLR    R1              ;INIT BREAK TIMER
670 002162 052777 000040 001450 BIS    #40,#DLRCSR      ;ENABLE DATA SET INTR.

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

671 002170 052777 000010 001442      BIS      #10,DLRCSR      ;SET SEC XMIT
672 002176 104407 000000' 1#:      BREAK$,BEGIN          ;TEMPORARY RETURN TO MONITOR....
      002176 104407 000000'          BREAK$,BEGIN          ;THEN CONTINUE AT NEXT INSTRUCTION.
      002202 104407 000000'          TST      INTFLG        ;DID INTR OCCUR YET ??
673 002206 005767 001450          BNE      3$           ;BR IF YES
674 002212 001023          DEC      R1           ;COUNT THE BREAK TIMER
675 002214 005301          BNE      1$           ;BR IF NO TIMEOUT
676 002216 001367          JSR      PC,STATR     ;GO SET UP ERROR INFO
677 002220 004767 001214          CLR      @DLRCSR     ;TURN OFF THE I.E. BIT
678 002224 005077 001410          MOV      #23,ERRTYP
679 002230 012767 000023 175650      ;*****
680 002236 104405 000000' 000000      HRDR$,BEGIN,NULL     ;DATA SET INTR. FAILED TO OCCUR
      ;*****
681 002244 000422          BR       DLT13       ;GO TO NEXT TEST
682 002246 042777 ^00040 001364 2#:      BIC      #40,DLRCSR   ;TURN OFF DATA SET I.E.
683 002254 005167 001402          COM      INTFLG     ;SET SOFTWARE INTR. FLAG
684 002260 000002          RTI                     ;RETURN CONTROL TO OTHER GUY
685 002262 032777 000040 001350 3#:      BIT      #40,DLRCSR ;DID INTR SERVICE TURN OFF I.E. ?
686 002270 001410          BEQ      DLT13     ;BR IF YES
687 002272 004767 001142          JSR      PC,STATR   ;GO SET UP ERROR INFO
688 002276 012767 000023 175602      MOV      #23,ERRTYP
689 0C2304 104405 000000' 000000      ;*****
      HRDR$,BEGIN,NULL     ;INTR SERVICE FAILED TO CLR DATA SET I.E.
      ;*****
690
691      ;TEST THAT "BREAK" BIT CAN SET AND CLEAR
692      ;-----
693
694 002312 052777 000001 001324 DLT13: BIS      #1,@DLXCSR   ;SET BREAK BIT
695 002320 032777 000001 001316      BIT      #1,@DLXCSR ;DID IT SET ??
696 002326 001010          BNE      1$           ;BR IF YES
697 002330 004767 001122          JSR      PC,STATX   ;GO SET UP ERROR INFO
698 002334 012767 000025 175544      MOV      #25,ERRTYP
699 002342 104405 000000' 000000      ;*****
      HRDR$,BEGIN,NULL     ;CAN'T SET BREAK BIT
      ;*****
700 002350 042777 000001 001266 1#:      BIC      #1,@DLXCSR   ;CLEAR THE BREAK BIT
701 002356 032777 000001 001260      BIT      #1,@DLXCSR ;DID IT CLEAR ?
702 002364 001410          BEQ      RESTRT     ;BR IF YES
703 002366 004767 001064          JSR      PC,STATX   ;GO SET UP ERROR INFO
704 002372 012767 000025 175506      MOV      #25,ERRTYP
705 002400 104405 000000' 000000      ;*****
      HRDR$,BEGIN,NULL     ;BREAK BIT WON'T CLEAR
      ;*****

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

707      ;
708      ;
709      ;
710      ;
711      002406      RESTRT:
712      002406      005067      001242      CLR      XEND      ;CLEAR END FLAGS
713      002412      005077      001226      CLR      @DLXCSR    ;CLEAR THE DL11 CONTROL REGS
714      002416      005077      001216      CLR      @DLRCSR    ;JUST IN CASE
715      002422      016700      175362      MOV      VECTOR,R0  ;GET START VECTOR ADDRESS
716      002426      012720      002674      MOV      @RINT,(R0)+ ;SET UP THE RCVR AND XMIT VECTORS
717      002432      116710      175354      MOV      BR1,(R0)
718      002436      005720      TST      (R0)+
719      002440      012720      002524      MOV      @XINT,(R0)+
720      002444      116710      175342      MOV      BR1,(R0)
721      002450      012703      003664      DOAGIN: MOV      @LDTAB,R3      ;POINT TO TABLE OF LOAD SUBR. POINTERS
722      002454      012704      003674      MOV      @MTAB,R4      ;POINT TO TABLE OF MESSAGE POINTERS
723      002460      005067      001174      1$:  CLR      RTRY      ;CLEAR RETRY FLAGS
724      002464      005777      001152      TST      @DLRDBR     ;FLUSH RCVR INPUT BUFFER REG
725      002470      005777      001146      TST      @DLRDBR
726      002474      012367      001204      MOV      (R3)+,LDOUT  ;SET UP CORRECT LOAD BUF ADDRESS POINTER
727      002500      012467      001202      MOV      (R4)+,AMESS  ;SET UP MESSAGE POINTER
728      002504      004767      000436      JSR      PC,SEGX     ;GO DO A SEGMENT
729      002510      022703      003674      CMP      @MTAB,R3     ;DONE ALL FOUR SEGMENTS ??
730      002514      001361      BNE      1$          ;BR IF NOT
731
732
733
734
735
736
737      002516      104413      000000      ENDIT$,BEGIN      ;SIGNAL END OF ITERATION.
738      002522      000752      BR      DOAGIN     ;MONITOR SHALL TEST END OF PASS
739
740
741

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

743          ;THIS ROUTINE SERVICES ALL XMITTR INTRPTS. FOR ALL 256. BYTE XFERS
744          ;-----
745
746 002524 105777 001114 XINT: TSTB  @DLXCSR      ;XMIT READY SET ??
747 002530 100403      BMI    1$          ;BR IF YES
748          ;-----
       002532 000004 000000' 002570' PIRQ$,BEGIN,4$      ; QUEUE UP TO CONTINUE AT 4$ AND RTI
       ;-----
749 002540 022767 004574' 001104 1$:  CMP    @DLBUFI,OPTR  ;OUTPUT 256. BYTES YET ??
750 002546 001405      BEQ    3$          ;BR IF YES
751 002550 117777 001076 001070 2$:  MOVB  @OPTR,@DLXDBR ;OUTPUT A CHARACTER
752 002556 005267 001070      INC    OPTR        ;POINT TO NEXT CHAR. IN BUFFER
753 002562      3$:
       ;-----
       002562 000004 000000' 002662' PIRQ$,BEGIN,6$      ; QUEUE UP TO CONTINUE AT 6$ AND RTI
       ;-----
754
755 002570 105767 001061 4$:  TSTB  XEND+1      ;ANY FATAL RCVR. ERRORS PENDING ??
756 002574 001025      BNE    5$          ;BR IF YES - STOP XMITTING
757 002576 016767 001042 175274  MOV    DLXCSR,CSRA ;SAVE THE CSR ADDRESS
758 002604 017767 001034 175270  MOV    @DLXCSR,ACSR ;SAVE THE CONTENTS OF THE CSR
759 002612 042777 000104 001024  BIC    @104,@DLXCSR ;DISABLE XMITR INTERRUPTS
760 002620 105167 001030      COMB   XEND          ;SET XMIT END FLAG
761
762 002624 012767 000011 175254  MOV    @11,ERRTYP
763 002632 104405 000000' 000000  ;*****
HRDR$,BEGIN,NULL      ;XMITTER FALSE INTERRUPT - FATAL ERROR
;*****
764
765 002640 005077 000774      CLR    @DLRCSR      ;TURN OFF RCVR INTR.
766 002644 104410 000000'      END$,BEGIN          ;
767
768 002650 042777 000100 000766 5$:  BIC    @100,@DLXCSR ;DISABLE XMITTR. INTERRUPTS
769 002656 104400 000000'      EXIT$,BEGIN        ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
770
771 002662 105767 000767 6$:  TSTB  XEND+1      ;ANY FATAL RCVR. ERRORS PENDING ??
772 002666 001370      BNE    5$          ;BR IF YES
773 002670 104400 000000'      EXIT$,BEGIN        ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
774
775          ;THIS ROUTINE SERVICES RECEIVER INTERRUPTS FOR ALL 256. BYTE XFERS
776          ;-----
777
778 002674 105777 000740 RINT: TSTB  @DLRCSR      ;RCVR DONE SET ??
779 002700 100403      BMI    1$          ;BR IF YES
780          ;-----
       002702 000004 000000' 002754' PIRQ$,BEGIN,3$      ; QUEUE UP TO CONTINUE AT 3$ AND RTI
       ;-----
781 002710 005777 000726 1$:  TST   @DLRDBR      ;OVERRUN/PARITY/FRAMING ERRORS ??
782 002714 100003      BPL    2$          ;BR IF NONE
783          ;-----
       002716 000004 000000' 003046' PIRQ$,BEGIN,5$      ; QUEUE UP TO CONTINUE AT 5$ AND RTI
       ;-----
784 002724 022767 005174' 000716 2$:  CMP    @BUFEND,IPTR ;INPUT BUFFER FULL ??
785 002732 001405      BEQ    7$          ;BR IF YES
786 002734 117777 000702 000706  MOVB  @DLRDBR,@IPTR ;READ THE DL INPUT BUFFER REG.
787 002742 005267 000702      INC    IPTR        ;POINT TO NEXT CHAR. POSITION
788 002746      7$:

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

002746 000004 000000' 003134'      ;-----
PIRQ$,BEGIN,6$                      ; QUEUE UP TO CONTINUE AT 6$ AND RTI
;-----

789
790 002754 105767 000674      3$:  TSTB   XEND                      ;ANY FATAL XMITTR ERROR PENDING
791 002760 001025                      BNE    4$                      ;BR IF YES
792 002762 016767 000652 175110  MOV    DLRCR,CSRA              ;SAVE THE RCVR. CSR ADDRESS
793 002770 017767 000644 175104  MOV    @DLRCR,ACSR            ;SAVE CONTENTS OF CSR
794 002776 042777 000100 000634  BIC    #100,@DLRCR           ;TURN OFF THE RCVR.
795 003004 105167 000645                      COMB   XEND+1                  ;SET FATAL RCVR ERROR FLAG
796
797 003010 012767 000011 175070  MOV    #11,ERRTYP
798 003016 104405 000000' 000000  ;*****
HRDR$,BEGIN,NULL                    ;RECEIVER FALSE INTERRUPT - FATAL ERROR
;*****

799
800 003024 005077 000614      CLR    @DLXCSR                ;DISABLE XMITTR TOO
801 003030 104410 000000'      END$,BEGIN                      ;
802
803 003034 042777 000100 000576  4$:  BIC    #100,@DLRCR           ;DISABLE RCVR INTERRUPTS
804 003042 104400 000000'      EXIT$,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
805
806 003046 105767 000602      5$:  TSTB   XEND                      ;ANY FATAL XMITTR ERRORS PENDING ??
807 003052 001370                      BNE    4$                      ;BR IF YES
808 003054 016767 000560 175016  MOV    DLRCR,CSRA              ;SAVE CSR ADDRESS
809 003062 017767 000552 175012  MOV    @DLRCR,ACSR            ;SAVE CONTENTS OF CSR
810 003070 017767 000546 175006  MOV    @DLRDBR,ASTAT          ;SAVE THE ERROR FLAGS
811 003076 042777 000100 000534  BIC    #100,@DLRCR           ;DISABLE RCVR INTR.
812
813 003104 012767 000017 174774  MOV    #17,ERRTYP
814 003112 104405 000000' 000000  ;*****
HRDR$,BEGIN,NULL                    ;OVERRUN - PARITY - FRAMING ERROR
;*****

815
816 003120 005077 000520      CLR    @DLXCSR                ;DISABLE XMITTR TOO
817 003124 105267 000530      INCB   RTRY                   ;SET RETRY FLAG
818 003130 104400 000000'      EXIT$,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
819
820 003134 105767 000514      6$:  TSTB   XEND                      ;ANY FATAL XMITTR ERRORS PENDING ??
821 003140 001335                      BNE    4$                      ;BR IF YES
822 003142 104400 000000'      EXIT$,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

824                                     ;THIS ROUTINE CONTROLS THE EXECUTION OF EACH OF THE FOUR DATA PATTERNS
825
826 003146 000240                      SEGX:  NOP                    ;DO NOTHING FOR NOW
827 003150 004767 000320                1$:  JSR      PC,CLDLBF          ;GO CLEAR BUFFERS
828 003154 004777 000524                JSR      PC,@LDOUT          ;GO SET UP PATTERN
829 003160 004767 000106                JSR      PC,KICKOF         ;GO KICK OFF XMITTR AND RCVR
830 003164 005002                      CLR      R2                 ;INITIALIZE BREAK TIMER
831 003166                               2$:
   003166 104407 000000'                BREAK$,BEGIN              ;TEMPORARY RETURN TO MONITOR....
   003172 104407 000000'                BREAK$,BEGIN              ;THEN CONTINUE AT NEXT INSTRUCTION.
832 003176 005767 000452                TST     XEND               ;ANY FATAL ERRORS PENDING ??
833 003202 001402                      BEQ     3$                 ;BR IF NOT
834 003204 104400 000000'                EXIT$,BEGIN                ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
835 003210 105767 000444                3$:  TSTB    RTRY              ;RETRY FLAG SET ??
836 003214 001411                      BEQ     4$                 ;BR IF NOT
837 003216 105067 000436                CLRB   RTRY                ;CLEAR THE FLAG
838 003222 105267 000433                INCB   RTRY+1              ;COUNT ONE RETRY
839 003226 122767 000003 000425        CMPB   #3,RTRY+1           ;TRIED THREE TIMES ??
840 003234 001345                      BNE    1$                 ;BR IF NOT - TRY IT AGAIN
841 003236 000406                      BR     5$                 ;REPORT IT AND GO TO NEXT SEGMENT
842 003240 022767 005174' 000402        4$:  CMP     @BUFEND,IPTR      ;RECEIVED 256. CHARS. ??
843 003246 001406                      BEQ     7$                 ;BR IF YES
844 003250 005302                      DEC     R2                  ;DECREMENT BREAK COUNTER
845 003252 001345                      BNE    2$                 ;BR IF NO TIMEOUT
846 003254                               5$:
   003254 104403 000000' 003706'        MSGN$,BEGIN,AMESS         ;ASCII MESSAGE CALL WITH COMMON HEADER
847 003262 000207                      6$:  RTS     PC                ;GO TO NEXT SEGMENT
848 003264 004767 000034                7$:  JSR      PC,CHKDAT         ;GO COMPARE IN/OUT DATA
849 003270 000207                      RTS     PC                  ;GO TO NEXT SEGMENT
850
851                                     ;THIS ROUTINE KICKS OFF ALL 256. BYTE TRANSFERS
852                                     ;-----
853
854 003272 012767 004174' 000352        KICKOF: MOV    #DLBUF0,OPTR   ;POINT TO BEGINNING OF OUTPUT BUFFER
855 003300 012767 004574' 000342        MOV    #DLBUFI,IPTR        ;POINT TO BEGINNING OF INPUT BUFFER
856 003306 052777 000104 000330        BIS    #104,@DLXCSR        ;TURN ON XMITTER
857 003314 052777 000100 000316        BIS    #100,@DLRCSR        ;TURN ON RCVR.
858 003322 000207                      RTS     PC                  ;RETURN TO CALLING SEGMENT
859
860                                     ;THIS ROUTINE CHECKS FOR AND REPORTS DATA COMPARE ERRORS
861                                     ;-----
862
863 003324 042777 000100 000312        CHKDAT: BIC    #100,@DLXCSR   ;DISABLE XMITTR INTR.
864 003332 042777 000100 000300        BIC    #100,@DLRCSR        ;DISABLE RCVR INTR.
865 003340 012700 004174'                MOV    #DLBUF0,R0          ;R0 POINTS TO OUTPUT BUFFER
866 003344 012701 004574'                MOV    #DLBUFI,R1          ;R1 POINTS TO INPUT BUFFER
867 003350 122021                      1$:  CMPB   (R0)+,(R1)+        ;COMPARE INPUT WITH OUTPUT
868 003352 001004                      BNE    2$                 ;BR IF NOT EQUAL
869 003354 022701 005174'                3$:  CMP     @BUFEND,R1        ;END OF THE BUFFERS ??
870 003360 001373                      BNE    1$                 ;BR IF NOT
871 003362 000207                      RTS     PC                  ;RETURN TO CALLER
872
873 003364 016767 000250 174506        2$:  MOV     DLRCSR,CSRA        ;SAVE THE CSR ADDRESS
874 003372 114067 174510                MOVB   -(R0),ASB           ;SAVE THE SHOULD BE DATA
875 003376 042767 177400 174502        BIC    #177400,ASB         ;ZERO HI BYTE
876 003404 010067 174472                MOV    R0,SBADR            ;SAVE THE SHOULD BE ADDRESS
877 003410 114167 174474                MOVB   -(R1),AWAS         ;SAVE THE WAS DATA

```

## DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```

878 003414 042767 177400 174466      BIC      #177400,AWAS      ;CLEAR OUT HI BYTE
879 003422 010167 174456              MOV      R1,WASADR        ;SAVE THE WAS ADDRESS
880
881      003426 104404 000000'          ;*****
      DATER$,BEGIN          ;DATA ERROR!!!
      ;*****

882
883 003432 105720              TSTB    (R0)+            ;POINT TO NEXT BYTE IN THE BUFFERS
884 003434 105721              TSTB    (R1)+
885 003436 000746              BR      3$              ;GO CHECK NEXT BYTE
886
887      ;THESE ROUTINES SET UP THE ERROR INFORMATION FOR THE BASIC TESTS
888      -----
889
890 003440 016767 000174 174432  STATR:  MOV      DLRCR$,CSRA      ;SAVE THE CSR ADDRESS
891 003446 017767 000166 174426              MOV      @DLRCR$,ACSR      ;SAVE THE CONTENTS OF THE CSR
892 003454 000207              RTS      PC              ;RETURN TO BASIC TESTS
893
894 003456 016767 000162 174414  STATX:  MOV      DLXCSR$,CSRA      ;SAVE THE CSR ADDRESS
895 003464 017767 000154 174410              MOV      @DLXCSR$,ACSR      ;SAVE THE CONTENTS OF THE CSR
896 003472 000207              RTS      PC              ;RETURN TO THE BASIC TESTS
897
898      ;THIS ROUTINE IS USED TO CLEAR THE INPUT/OUTPUT BUFFERS
899      -----
900
901 003474 012700 004174'          CLDLBF: MOV      #DLBUF0,R0      ;POINT R0 TO BEGINNING OF BUFFERS
902 003500 005020              1$:    CLR      (R0)+        ;CLEAR ONE WORD - UPDATE POINTER
903 003502 022700 005174'          CMP      #BUFEND,R0          ;DONE 256. BYTES ??
904 003506 001374              BNE     1$                  ;BR IF NOT
905 003510 000207              RTS      PC              ;RETURN TO CALLING SEGMENT
906
907      ;THIS ROUTINE LOADS THE OUTPUT BUFFER WITH A NULL-DEL-NULL PATTERN
908      -----
909
910 003512 012700 004174'          LDOUT1: MOV      #DLBUF0,R0      ;SET UP POINTER
911 003516 105020              1$:    CLR      (R0)+        ;MOV A NULL CHAR
912 003520 112720 000377          MOV      #377,(R0)+        ;MOV A DELETE CHAR.
913 003524 022700 004574'          CMP      #DLBUFI,R0          ;BUFFER FULL ???
914 003530 001372              BNE     1$                  ;BR IF NOT
915 003532 000207              RTS      PC              ;RETURN TO CALLING SEGMENT
916
917      ;THIS ROUTINE IS USED TO LOAD AN ASCENDING BINARY COUNT PATTERN
918      -----
919
920
921 003534 012700 004174'          LDOUT2: MOV      #DLBUF0,R0      ;SET UP POINTER
922 003540 005001              CLR      R1              ;USE R1 TO GENERATE THE PATTERN
923 003542 110120              1$:    MOV      R1,(R0)+        ;LOAD ONE CHAR.
924 003544 022700 004574'          CMP      #DLBUFI,R0          ;BUFFER FULL ???
925 003550 001402              BEQ     2$                  ;BR IF YES
926 003552 105201              INCB    R1              ;GENERATE NEXT CHAR.
927 003554 000772              BR      1$                  ;GO MOVE IT
928 003556 000207              2$:    RTS      PC              ;RETURN TO CALLING SEGMENT
929
930      ;THIS ROUTINE IS USED TO LOAD THE DESCENDING BINARY COUNT PATTERN
931      -----
932

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

933 003560 012700 004174'      LDOUT3: MOV      #DLBUFO,R0      ;SET UP POINTER
934 003564 012701 000377      MOV      #377,R1          ;START R1 AT 377
935 003570 110120              1$:   MOVB     R1,(R0)+      ;LOAD ONE CHAR.
936 003572 022700 004574'      CMP      #DLBUFI,R0      ;AT END OF THE BUFFER ??
937 003576 001402              BEQ      2$              ;BR IF YES
938 003600 105301              DECB     R1              ;GENERATE NEXT CHAR.
939 003602 000772              BR       1$              ;GO MOVE IT
940 003604 000207              2$:   RTS      PC          ;RETURN TO CALLING SEGMENT
941
942                          ;THIS ROUTINE LOADS THE WORST CASE PATTERN
943                          ;-----
944
945 003606 012700 004174'      LDOUT4: MOV      #DLBUFO,R0      ;SET UP POINTERS
946 003612 012701 005176'      1$:   MOV      #WCASE,R1        ;POINT TO MONITOR'S WORST CASE PATTERN
947 003616 012120              2$:   MOV      (R1)+,(R0)+      ;LOAD ONE WORD
948 003620 022700 004574'      CMP      #DLBUFI,R0      ;BUFFER FULL ???
949 003624 001404              BEQ      3$              ;BR IF YES
950 003626 022701 005276'      CMP      #WCASEE,R1        ;END OF WORST CASE PATTERN ??
951 003632 001767              BEQ      1$              ;GO RESET R1
952 003634 000770              BR       2$              ;GO RESET R1
953 003636 000207              3$:   RTS      PC          ;RETURN TO CALLING SEGMENT

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

955          ;VARIABLES, FLAGS, MESSAGES, AND BUFFERS
956          ;-----
957
958 003640 000000      DLRCR: OPEN          ;CONTAINS ADDRESS OF RCVR CSR
959 003642 000000      DLRDBR: OPEN         ;CONTAINS ADDRESS OF RCVR DBR
960 003644 000000      DLXCSR: OPEN        ;CONTAINS ADDRESS OF XMITR CSR
961 003646 000000      DLXDBR: OPEN        ;CONTAINS ADDRESS OF XMITR DBR
962
963 003650 000000      IPTR: OPEN           ;CONTAINS POINTER TO INPUT BUFFER
964 003652 000000      OPTR: OPEN          ;CONTAINS POINTER TO OUTPUT BUFFER
965
966 003654 000000      XEND: OPEN           ;FATAL ERROR END FLAGS
967 003656 000000      EPCTR: OPEN         ;END OF PASS COUNTER
968 003660 000000      RTRY: OPEN          ;RETRY FLAG AND COUNTER
969 003662 000000      INTFLG: OPEN        ;SOFTWARE INTR. FLAG USED BY BASIC TESTS
970
971 003664 003512'     LDOUT: LDOUT1        ;POINTER TO 1ST LOAD BUFFER SUBR.
972 003666 003534'     LDOUT: LDOUT2        ;POINTER TO 2ND LOAD BUFFER ROUTINE
973 003670 003560'     LDOUT: LDOUT3        ;POINTER TO 3RD LOAD BUFFER ROUTINE
974 003672 003606'     LDOUT: LDOUT4        ;POINTER TO 4TH LOAD BUFFER ROUTINE
975 003674 003716'     MTAB: MSG1           ;POINTER TO MESSAGE 1
976 003676 003757'     MTAB: MSG2           ;POINTER TO MESSAGE 2
977 003700 004022'     MTAB: MSG3           ;POINTER TO MESSAGE 3
978 003702 004067'     MTAB: MSG4           ;POINTER TO MESSAGE 4
979
980 003704 000000      LDOUT: OPEN          ;CONTAINS POINTER TO LOAD BUFFER SUBR.
981 003706 003716'     AMESS: MSG1          ;MESSAGE POINTERS
982 003710 177777      -1                  ;TERMINATOR
983 003712 004125'     DRPMS: MSG5          ;MESSAGE POINTER
984 003714 177777      -1                  ;TERMINATOR
985 003716 045 116 125 MSG1: .ASCIZ /*NULL-DEL-NULL SEQUENCE ABORTED*/
    003721 114 114 055
    003724 104 105 114
    003727 055 116 125
    003732 114 114 040
    003735 123 105 121
    003740 125 105 116
    003743 103 105 040
    003746 101 102 117
    003751 122 124 105
    003754 104 045 000
986 003757 045 102 111 MSG2: .ASCIZ /*BINARY UP COUNT SEQUENCE ABORTED*/
    003762 116 101 122
    003765 131 040 125
    003770 120 040 103
    003773 117 125 116
    003776 124 040 123
    004001 105 121 125
    004004 105 116 103
    004007 105 040 101
    004012 102 117 122
    004015 124 105 104
    004020 045 000
987 004022 045 102 111 MSG3: .ASCIZ /*BINARY DOWN COUNT SEQUENCE ABORTED*/
    004025 116 101 122
    004030 131 040 104
    004033 117 127 116

```

## DEC/X11 SYSTEM EXERCISER MACRO DEFINITION MODULE

```

004036      040      103      117
004041      125      116      124
004044      040      123      105
004047      121      125      105
004052      116      103      105
004055      040      101      102
004060      117      122      124
004063      105      104      045
004066      000
988 004067      045      127      117 MSG4: .ASCIZ /#WORST CASE SEQUENCE ABORTED#/
004072      122      123      124
004075      040      103      101
004100      123      105      040
004103      123      105      121
004106      125      105      116
004111      103      105      040
004114      101      102      117
004117      122      124      105
004122      104      045      000
989 004125      045      106      101 MSG5: .ASCIZ /#FATAL ERROR IN STATIC REGISTER TESTS#/
004130      124      101      114
004133      040      105      122
004136      122      117      122
004141      040      111      116
004144      040      123      124
004147      101      124      111
004152      103      040      122
004155      105      107      111
004160      123      124      105
004163      122      040      124
004166      105      123      124
004171      123      045      000
990          .EVEN
991
992          ;512 WORDS RESERVED FOR TWO 256. BYTE BUFFERS
993          ;-----
994
995 004174          DLBUFO: .BLKB 256.      ;RSVD FOR OUTPUT BUFFER
996 004574          DLBUFI: .BLKB 256.      ;RSVD FOR INPUT BUFFER
997 005174 000000  BUFEND: 0          ;MARK END OF BUFFER AREA
998
999
1000 005176 177776 000001 177775 WCASE: .WORD 177776.1,177775.2,177773.4,177767.10
005204 000002 177773 000004
005212 177767 000010
1001 005216 177757 000020 177737 .WORD 177757.20,177737.40,177677.100,177577.200
005224 000040 177677 000100
005232 177577 000200
1002 005236 177377 000400 176777 .WORD 177377.400,176777.1000,175777.2000,173777.4000
005244 001000 175777 002000
005252 173777 004000
1003 005256 167777 010000 157777 .WORD 167777.10000,157777.20000,137777.40000,77777.100000
005264 020000 137777 040000
005272 077777 100000
1004 005276          WCASEE:
1005
1006          000001          .END

```

## Symbol table

ACSR	000102R	CLKSP#	= 104422	ICONT	000036R	OTOA#	= 104420	R6	=#000006
ADDR	000006R	CONFIG	000056R	ICOUNT	000040R	PARPRE	= 002000	R7	=#000007
ADDR22	= 001000	CSRA	000100R	IDNUM	000122R	PASCNT	000034R	SBADR	000102R
AMESS	003706R	DATCK#	= 104411	INDPAR	= 000040	PDPF11	= 000002	SEGX	003146R
APTPRE	= 000200	DATER#	= 104404	INIT	000030R	PDPLSI	= 020000	SOFcnt	000042R
ASB	000106R	DLBUFI	004574R	INTFLG	003662R	PDP44	= 100000	SOFER#	= 104406
ASTAT	000104R	DLBUFO	004174R	INTR	000120R	PDP60	= 004000	SOFPAS	000046R
AUTO	= 000010	DLINIT	000260R	IPTR	003650R	PDP70	= 010000	SPOINT	000032R
AWAS	000110R	DLRCSR	003640R	KICKOF	003272R	PIRQ#	= 000004	SPSIZ	= 000040
BEGIN	000000R	DLRDBR	003642R	KTPRES	= 000400	POPSP	= 005726	SR1	000016R
BIT0	= 000001	DLT01	000334R	KTXTND	= 040000	POPSP2	= 022626	SR2	000020R
BIT1	= 000002	DLT02	000374R	LDOUT	003704R	PRHMS#	= 000002	SR3	000022R
BIT10	= 002000	DLT03	000436R	LDOUT1	003512R	PRTY	= 000000	SR4	000024R
BIT11	= 004000	DLT04	000556R	LDOUT2	003534R	PRTY0	= 000000	START	000226R
BIT12	= 010000	DLT05	000744R	LDOUT3	003560R	PRTY1	= 000040	STAT	000026R
BIT13	= 020000	DLT06	001376R	LDOUT4	003606R	PRTY2	= 000100	STATR	003440R
BIT14	= 040000	DLT07	001530R	LDTAB	003664R	PRTY3	= 000140	STATX	003456R
BIT15	= 100000	DLT10	001706R	M^P22#	= 104416	PRTY4	= 000200	SVRO	000062R
BIT2	= 000004	DLT11	002050R	MODEM	001364R	PRTY5	= 000240	SVR1	000064R
BIT3	= 000010	DLT12	002144R	MODNAM	000000R	PRTY6	= 000300	SVR2	000066R
BIT4	= 000020	DLT13	002312R	MODREV	000224R	PRTY7	= 000340	SVR3	000070R
BIT5	= 000040	DLXCSR	003644R	MODSP	000224R	PS	= 177776	SVR4	000072R
BIT6	= 000100	DLXDBR	003646R	MSGN#	= 104403	PSW	= 177776	SVR5	000074R
BIT7	= 000200	DOAGIN	002450R	MSG#	= 104402	PUSH	= 005746	SVR6	000076R
BIT8	= 000400	DRPMS	003712R	MSG1	003716R	PUSH2	= 024646	SYSCNT	000052R
BIT9	= 001000	DVID1	000014R	MSG2	003757R	PWRFLG	= 000002	TRPDFD	= 000023
BREAK#	= 104407	ECCMEM	= 000100	MSG3	004022R	QMON2#	= 000010	USTACK	= 000001
BR1	000012R	ENDIT#	= 104413	MSG4	004067R	RAND#	= 104417	VECTOR	000010R
BR2	000013R	END#	= 104410	MSG5	004125R	RANNUM	000054R	WASADR	000104R
BTOD#	= 104421	EPCTR	003656R	MTAB	003674R	RESTRT	002406R	WCASE	005176R
BUFEND	005174R	ERRTYP	000106R	NCPUOP	= 000020	RES1	000056R	WCASEE	005276R
CAPRES	= 000004	EXIT#	= 104400	NOPTY	= 000002	RES2	000060R	WDFR	000116R
CDATA#	= 104412	GETPA#	= 104415	NULL	= 000000	RH70	= 001000	WDT0	000114R
CHKDAT	003324R	GWBUF#	= 104414	OPEN	= 000000	RINT	002674R	XEND	003654R
CKHNG#	= 000001	HRDCNT	000044R	OPTR	003652R	RSTRT	000112R	XFLAG	000005R
CLDLBF	003474R	HRDER#	= 104405			RTRY	003660R	XINT	002524R
CLKPRE	= 000001	HRDPAS	000050R						

. ABS. 000000 000 (F,I,GBL,ABS,OVR)  
 005276 001 (RW,I,LCL,REL,CON)

Errors detected: 0

## \*\*\* Assembler statistics

Work file reads: 0  
 Work file writes: 0  
 Size of work file: 13285 Words ( 52 Pages)  
 Size of core pool: 19402 Words ( 74 Pages)  
 Operating system: RSX-11M/PLUS (Under VAX/VMS)

Elapsed time: 00:01:36.98  
 DLB.OBJ,DLB.LST/CR/-SP=DDXCOM,DLB

## SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES									
ACSR	000102 R	04-367	*5-544	*7-758	*7-793	*7-809	*8-891	*8-895			
ADDR	000006 R	04-367	5-382								
ADDR22	= 001000	04-367									
AMESS	003706 R	*6-727	8-846	*9-981							
APTPRE	= 000200	04-367									
ASB	000106 R	04-367	*5-545	*8-874	*8-875						
ASTAT	000104 R	04-367	*7-810								
AUTO	= 000010	04-367									
AWAS	000110 R	04-367	*5-527	5-541	*8-877	*8-878					
BEGIN	000000 R	04-367	5-422	5-423	5-424	5-433	5-434	5-435	5-445	5-446	
		5-447	5-453	5-454	5-455	5-467	5-467	5-474	5-475	5-476	
		5-484	5-485	5-486	5-500	5-501	5-502	5-508	5-509	5-510	
		5-515	5-515	5-524	5-525	5-526	5-538	5-539	5-540	5-548	
		5-566	5-571	5-584	5-594	5-599	5-612	5-617	5-628	5-633	
		5-639	5-644	5-654	5-660	5-672	5-672	5-680	5-689	5-699	
		5-705	6-737	7-748	7-753	7-763	7-766	7-769	7-773	7-780	
		7-783	7-788	7-798	7-801	7-804	7-814	7-818	7-822	8-831	
		8-831	8-834	8-846	8-881						
BIT0	= 000001	04-367									
BIT1	= 000002	04-367									
BIT10	= 002000	04-367									
BIT11	= 004000	04-367	5-408								
BIT12	= 010000	04-367									
BIT13	= 020000	04-367									
BIT14	= 040000	04-367									
BIT15	= 100000	04-367									
BIT2	= 000004	04-367									
BIT3	= 000010	04-367									
BIT4	= 000020	04-367									
BIT5	= 000040	04-367									
BIT6	= 000100	04-367									
BIT7	= 000200	04-367									
BIT8	= 000400	04-367									
BIT9	= 001000	04-367									
BREAK#	= 104407	04-367	5-467	5-467	5-515	5-515	5-672	5-672	8-831	8-831	
BR1	000012 R	04-367	5-464	5-494	6-717	6-720					
BR2	000013 R	04-367									
BTOD#	= 104421	04-367									
BUFEND	005174 R	7-784	8-842	8-869	8-903	*9-997					
CAPRES	= 000004	04-367									
CDATA#	= 104412	04-367									
CHKDAT	003324 R	8-848	*8-863								
CKMNG#	= 000001	04-367									
CLDLBF	003474 R	8-827	*8-901								
CLKPRE	= 000001	04-367									
CLKSP#	= 104422	04-367									
CONFIG	000056 R	04-367									
CSRA	000100 R	04-367	*5-543	*7-757	*7-792	*7-808	*8-873	*8-890	*8-894		
DATCK#	= 104411	04-367									
DATER#	= 104404	04-367	5-548	8-881							
DLBUF1	004574 R	7-749	8-855	8-866	8-913	8-924	8-936	8-948	*9-996		
DLBUFO	004174 R	8-854	8-865	8-901	8-910	8-921	8-933	8-945	*9-995		

## SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
DLINIT	000260 R	#5-396
DLRCSR	003640 R	*5-383 5-396 5-397 5-418 5-495 5-496 5-503 5-504 5-511
		5-521 5-528 5-532 5-535 5-543 5-544 5-561 5-562 5-567
		5-572 5-580 5-589 5-590 5-595 5-600 5-608 5-613 5-622
		5-623 5-624 5-629 5-634 5-635 5-640 5-649 5-650 5-655
		5-656 5-670 5-671 5-678 5-682 5-685 6-714 7-765 7-778
		7-792 7-793 7-794 7-803 7-808 7-809 7-811 8-857 8-864
		8-873 8-890 8-891 #9-958
DLRDBR	003642 R	*5-385 5-399 5-400 5-527 5-546 6-724 6-725 7-781 7-786
		7-810 #9-959
DLT01	000334 R	#5 418
DLT02	000374 R	5-419 #5-429
DLT03	000436 R	5-430 #5-440
DLT04	000556 R	5-450 #5-460
DLT05	000744 R	5-481 #5-491
DLT06	001376 R	5-555 #5-561
DLT07	001530 R	5-581 #5-589
DLT10	001706 R	5-614 #5-622
DLT11	002050 R	5-641 #5-649
DLT12	002144 R	5-657 #5-666
DLT13	002312 R	5-661 5-681 5-686 #5-694
DLXCSR	003644 R	*5-387 5-398 5-412 5-429 5-440 5-441 5-448 5-449 5-466
		5-477 5-480 5-512 5-522 5-529 5-536 5-694 5-695 5-700
		5-701 6-713 7-746 7-757 7-758 7-759 7-768 7-800 7-816
		8-856 8-863 8-894 8-895 #9-960
DLXDBR	003646 R	*5-389 5-514 5-547 7-751 #9-961
DOAGIN	002450 R	#6-721 6-738
DRPMS	003712 R	5-423 5-434 5-446 5-454 5-475 5-485 5-501 5-509 5-525
		5-539 #9-983
DVID1	000014 R	#4-367
ECCMEM	= 000100	#4-367
ENDIT#	= 104413	#4-367 6-737
END#	= 104410	#4-367 5-424 5-435 5-447 5-455 5-476 5-486 5 502 5 510
		5-526 5-540 7-766 7-801
EPCTR	003656 R	#9-967
ERRTYP	000106 R	#4-367 *5-421 *5-432 *5-444 *5-452 *5-473 *5-483 *5-499 *5-507
		*5-523 *5-537 *5-565 *5-570 *5-583 *5-593 *5-598 *5-611 *5-616
		*5-627 *5-632 *5-638 *5-643 *5-653 *5-659 *5-679 *5-688 *5-698
		*5-704 *7-762 *7-797 *7-813
EXIT#	= 104400	#4-367 7 769 7-773 7-804 7-818 7-822 8-834
GETPA#	= 104415	#4-367
GMBUF#	= 104414	#4-367
HRDCNT	000044 R	#4-367
HRDER#	= 104405	#4-367 5-422 5-433 5-445 5-453 5-474 5-484 5-500 5 508
		5-524 5-538 5-566 5-571 5-584 5-594 5-599 5-612 5-617
		5-628 5-633 5-639 5-644 5-654 5-660 5-680 5-689 5-699
		5-705 7-763 7-798 7-814
HRDPAS	000050 R	#4-367
ICONT	000036 R	#4 367
ICOUNT	000040 R	#4-367
IDNUM	000122 R	#4-367
INDPAR	= 000040	#4-367

## SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
INIT	000030 R	*4-367
INTFLG	003662 R	*5-460 5 468 *5-478 *5-491 5-516 *5-530 *5-666 5-673 *5-683
		*9-969
INTR	000120 R	*4-367
IPTR	003650 R	7-784 7 786 *7-787 8-842 *8-855 *9-963
KICKOF	003272 R	8-829 *8-854
KTPRES	= 000400	*4-367
KTXTND	= 040000	*4-367
LDOUT	003704 R	*6-726 8-828 *9-980
LDOUT1	003512 R	*8-910 9-971
LDOUT2	003534 R	*8-921 9-972
LDOUT3	003560 R	*8-933 9-973
LDOUT4	003606 R	*8-945 9-974
LDTAB	003664 R	6-721 *9-971
MAP22	= 104416	*4-367
MODEM	001364 R	5-542 *5-554
MODNAM	000000 R	*4-367
MODREV	000224 R	*4-377
MODSP	000224 R	4-367 *4-367
MSGN	= 104403	*4-367 5-423 5-434 5-446 5 454 5-475 5-485 5 501 5 509
		5-525 5-539 8-846
MSG	= 104402	*4-367
MSG1	= 104401	*4-367
MSG1	003716 R	9-975 9-981 *9-985
MSG2	003757 R	9-976 *9-986
MSG3	004022 R	9-977 *9-987
MSG4	004067 R	9-978 *9-988
MSG5	004125 R	9-983 *9-989
MTAB	003674 R	6-722 6-729 *9-975
NCPUOP	= 000020	*4-367
NOPTY	= 000002	*4-367
NULL	= 000000	*4-367 5-422 5-433 5-445 5-453 5-474 5-484 5-500 5 508
		5-524 5-538 5-566 5-571 5-584 5-594 5-599 5-612 5-617
		5-628 5-633 5-639 5-644 5-654 5-660 5-680 5-689 5 699
		5-705 7-763 7-798 7-814
OPEN	= 000000	4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367
		4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367 4-367
		4-367 4-367 *4-367 9-958 9-959 9-960 9-961 9-963 9 964
		9-966 9-967 9-968 9-969 9-980
OPTR	003 52 R	7 749 7-751 *7-752 *8-854 *9-964
OTOA	= 104420	*4-367
PARPRE	= 002000	*4-367
PASCNT	000034 P	*4-367
PDPF11	= 000002	*4-367
PDPLSI	= 020000	*4-367
PDP44	= 100000	*4-367
PDP60	= 004000	*4-367
PDP70	= 010000	*4-367
PIRQ	= 000004	*4-367 7-748 7-753 7-780 7-783 7-788
POPSP	= 005726	*4-367
POPSP2	= 022626	*4-367
PRHMS	= 000002	*4-367

## SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
PRTY	= 000000	4-367 #4-367
PRTY0	= 000000	#4-367
PRTY1	= 000040	#4-367
PRTY2	= 000100	#4-367
PRTY3	= 000140	#4-367
PRTY4	= 000200	4-367 #4 367
PRTY5	= 000240	#4-367
PRTY6	= 000300	#4-367
PRTY7	= 000340	#4-367
PS	= 177776	#4-367
PSW	= 177776	#4-367
PUSH	= 005746	#4-367
PUSH2	= 024646	#4-367
PWRFLG	= 000002	#4-367
QMON22	= 000010	#4-367
RAND\$	= 104417	#4-367
RANNUM	000054 R	#4 367
RESTR1	002406 R	4-367 5-556 5-702 #6-711
RES1	000056 R	#4-367
RES2	000060 R	#4-367
RH70	= 001000	#4-367
RINT	002674 R	6-716 #7-778
RSTRT	000112 R	#4-367
RTRY	003660 R	*6-723 *7-817 8-835 *8-837 *8-838 8-839 #9-968
R6	=#000006	#4-367
R7	=#000007	#4-367
SBADR	000102 R	#4-367 *5-547 *8-876
SEGX	003146 R	6-728 #8-826
SOFcnt	000042 R	#4-367
SOFER\$	= 104406	#4-367
SFPAS	000046 R	#4-367
SPOINT	000032 R	#4-367
SPSIZ	= 000040	#1-29 4-367
SR1	000016 R	#4-367 5-554
SR2	000020 R	#4-367 5-408 5-410
SR3	000022 R	#4-367
SR4	000024 R	#4-367
START	000226 R	4-367 #5-382
STAT	000026 R	#4-367
STATR	003440 R	5-420 5-498 5-506 5-520 5-534 5-564 5-569 5-582 5-592
		5-597 5-610 5-615 5-626 5-631 5-637 5-642 5-652 5 658
		5-677 5-687 #8-890
		5-431 5 443 5-451 5-472 5-482 5-697 5-703 #8-894
STATX	003456 R	5-431 5 443 5-451 5-472 5-482 5-697 5-703 #8-894
SVR0	000062 R	#4-367
SVR1	000064 R	#4-367
SVR2	000066 R	#4-367
SVR3	000070 R	#4-367
SVR4	000072 R	#4-367
SVR5	000074 R	#4-367
SVR6	000076 R	#4-367
SYSCNT	000052 R	#4-367
TRPDFD	= 000023	#4-367 4-367 4 367 #4-367 4-367 4-367 #4-367 4-367 4-367

SYMBOL CROSS REFERENCE

CREF V02

SYMBOL	VALUE	REFERENCES
		04-367 4-367 4-367 04-367 4-367 4-367 04-367 4-367 4-367
		04-367 4-367 4-367 04-367 4-367 4-367 04-367 4-367 4-367
		04-367 4-367 4-367 04-367 4-367 4-367 04-367 4-367 4-367
		04-367 4-367 4-367 04-367 4-367 4-367 04-367 4-367 4-367
		04-367 4-367 4-367 04-367 4-367 4-367 04-367 4-367 4-367
USTACK	* 000001	04-367
VECTOR	000010 R	04-367 5-461 5-492 5-667 6-715
WASADR	000104 R	04-367 *5-546 *8-879
WCASE	005176 R	8-946 09-1000
WCASEE	005276 R	8-950 09-1004
WDFR	000116 R	04-367
WDTO	000114 R	04-367
XEND	003654 R	*6-712 7-755 *7-760 7-771 7-790 *7-795 7-806 7-820 8-832
		09-966
XFLAG	000005 R	04-367
XINT	002524 R	6 719 07-746

