```
.REM
```

IDENTIFICATION

PRODUCT CODE: AC E709L-MC

PRODUCT NAME: CXDLAL0 DL11 MODULE

PRODUCT DATE: NOVEMBER 1984

MAINTAINER: DEC/X11 SUPPORT GROUP

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

1. ABSTRACT:

DLA IS AN IOMOD THAT EXERCISES UP TO SIXTEEN DL11 ASYNCHRONOUS INTERFACES. IT IS CAPABLE OF EXERCISING ALL DL11 MODELS . IT USES MAINTENANCE MODE TO TRANSMIT AND RECEIVE A BINARY COUNT PATTERN OUTPUT AND RECEIVED IN 64. CHARACTER BURSTS. THE MAJOR PORTION OF THE ERROR CHECKING IS DEFERRED TO PRIORITY LEVEL 0. ALL LINES SELECTED FOR TEST (UP TO 16 DL11'S WITH CONTIGUOUS ADDRESSES AND VECTORS) ARE ACTIVATED AND RUN CONCURRENTLY. ALL TRANSMIT AND RECEIVE ERRORS ARE REPORTED ON THE CONSOLE TTY.

NOTE:
THIS MODULE IS NOT DESIGNED TO EXERCISE ANY TYPE OF TERMINAL DEVICE. IT IS DESIGNED ONLY TO EXERCISE THE DL. ANY INFORMATION DISPLAYED OR PRINTED IS INCIDENTAL AND SHOULD BE IGNORED. IN MOST CASES THE TERMINAL DEVICE SHOULD BE PLACED IN THE OFF-LINE MODE.

2 REQUIREMENTS·

HARDWARE: AT LEAST ONE DL11 INTERFACE
NOTE: FOR THE DLV11, THE TEST CONNECTOR MUST BE INSTALLED !

STORAGE:: DLA REQUIRES:
1. DECIMAL WORDS: 959 ;RG002
2. OCTAL WORDS: 1677 ;RG002
3. OCTAL BYTES: 3576 ;RG002

3 PASS DEFINITION:

ONE PASS OF THE DLA MODULE CONSISTS OF TRANSMITTING AND RECEIVING SOME CHARACTERS. THE EXACT NUMBER DEPENDS ON THE BAUD RATE

4. EXECUTION TIME:

VARIES WITH BAUD RATE BUT SHOULD TAKE ABOUT ONE MINUTE TO COMPLETE ONE PASS WHEN RUNNING ALONE. SR1 MUST BE SET TO MATCH THE BAUD RATE OR THE PASS TIMES WILL BE OFF. SR2 CAN BE SET TO THE ITERATION CONSTANT (ICONT) DESIRED REGARDLESS OF THE BAUD RATE IF DESIRED. (RS001)

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 176500, VCT: 1, BR1: 4, BR2: 0, DVC: 1

REQUIRED PARAMETERS:

AT CONFIGURATION TIME THE USER MUST SPECIFY:

VCT: VECTOR ADDRESS OF FIRST DL11
DVC: NO OF DL11'S IF GREATER THAN 1

6. DEVICE OPTION SETUP:

NONE REQUIRED

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

*. MODULE OPERATION:

*.1 TEST SEQUENCE:

A. START: USING THE DEVICE SELECTION PARAMETER "DVID1" THIS SECTION OF CODE SETS UP THE VECTORS OF ALL SELECTED LINES TO POINT TO THE APPROPRIATE JSR INSTRUCTION IN THE JSR LINKING TABLE.

B. SETCSR: THIS PIECE OF CODE INSERTS THE PROPER CSR ADDRESS OF EACH ACTIVE LINE INTO THE THIRD WORD OF EACH JSR TABLE ENTRY.

C. STUP: THIS ROUTINE INITIALIZES ALL TABLES, BUFFERS, FLAGS AND COUNTERS. THEN PROCEEDS TO TURN ON THE INTERRUPTS FOR ALL ACTIVE LINES. IT USES THE CONTENTS OF THE ACTIVE DEVICE TABLE TO FIND OUT WHICH LINES TO KICK OFF. AFTER INITIALIZING ALL LINES IT WAITS FOR COMPLETION OF 64 TRANSMITTER AND RECEIVER INTERRUPTS VIA A BREAK LOOP. IF THE 64 INTERRUPTS HAVE OCCURRED ON BOTH TRANSMITTER AND RECEIVER, OR IF THE BREAK LOOP TIMES OUT, CONTROL PASSES TO ERRCHK.

D. TINT: THE TRANSMITTER SERVICE ROUTINE SIMPLY QUEUES UP THE REQUEST FOR SERVICE IN A FIFO QUEUE, UPDATES THE POINTER, AND RETURNS CONTROL BACK TO THE MONITOR WITH A PIRQ. THE ELEMENT THAT GETS STORED IN THE QUEUE IS A POINTER TO THE INTERRUPTING CSR ADDRESS. THE ACTUAL SERVICING IS DONE LATER WHERE THE SERVICE CODE IS EXECUTED AT PRIORITY LEVEL 0.

E. TSERV: THIS CODE RETRIEVES A POINTER FROM THE FIFO QUEUE AND BUILDS THE CSR ADDRESS. THE FOLLOWING SEQUENCE IS EXECUTED:

1. TEST FOR END OF 64. CHAR BURST - IF END EXIT - IF NOT GO TO 2

2 TEST READY FLAG - IF NOT ASSERTED GO REPORT FALSE INTERRUPT - IF ASSERTED PROCEED TO STEP 3

3. COUNT THE INTERRUPT FOR INDIVIDUAL LINE

4. GENERATE AND OUTPUT NEXT CHARACTER, KEEP TRACK OF THE NUMBER OF CHARACTERS

OUTPUT ON THE LINE, AND THEN EXIT BACK TO THE MONITOR.

F. RINT: THE RECEIVER SERVICE ROUTINE STORES DATA AND STATUS INFORMATION IN A RE CEIVER STARTUP TABLE, TESTS FOR THE END OF A 64. CHAR XFR SEQUENCE AND THEN EXECUTES AN "RTI" . IT ALSO COUNTS RECEIVE INTERRUPTS IN A SEPARATE COUNTER FOR EACH LINE. SEPARATE COUNTER.

G. ERRCHK: THE BULK OF THE ERROR CHECKING AND RE-PORTING IS DONE HERE AT THE END OF EACH 64. CHAR. BURST. THE FOLLOWING SEQUENCE IS EXECUTED:

1. TURN OFF RCVR AND XMTR INTR. ENABLES FOR ALL ACTIVE LINES

2. SCAN THROUGH THE RECEIVER STATUS TABLE (64 ENTRIES OF TWO WORDS EACH) TO CHECK FOR AND REPORT:

   A.) PARITY, FRAMING AND OVER-RUN ERRORS.
   B.) RCVR FALSE INTERRUPTS
   C.) DATA COMPARE ERRORS. ONLY IF A AND B DID NOT OCCUR.

3. CHECK RECEIVER AND TRANSMITTER INTERRUPT COUNTS FOR EACH LINE TO BE SURE THAT NO LINES WERE DROPPED OR HAD TOO MANY INTERRUPTS.

4. GO TO THE ENPS ROUTINE AFTER CHECKING ALL 64 ENTRIES.

H. ENPS: COUNT THE 64. CHAR BURST AND TEST FOR 128. BURSTS (8192 CHARS). IF NOT END OF PASS GO TO I. IF END REPORT END OF PASS AND GO TO C.

I. RESYNC: RESYNC THE DATA BUFFERS AND THEN RESTART AT STEP C.

7.2 DESCRIPTION OF TABLES, QUEUES, AND BUFFERS

A. RSTAB: THIS IS A 128. WORD STATUS TABLE CON-SISTING OF 64. TWO WORD ENTRIES. IT GETS LOADED DURING RECEIVER INTERRUPT SERVICE AND CHECKED AT THE END OF EACH 64. CHAR BURST. EACH ENTRY HAS THE FOLLOWING FORMAT:

1ST WORD: CONTENTS OF RCSR

2ND WORD: LO BYTE = RCVD DATA BYTE
HI BYTE = LINE NUMBER

B. RCNT: 16 BYTE TABLE CONTAINING AN 8 BIT IN TERRUPT COUNTER FOR EACH RCVR. THE APPROPRIATE BYTE GETS INCREMENTED DURING RCVR INTR SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

C. TCNT: 16 BYTE TABLE CONTAINING AN 8-BIT IN TERRUPT COUNTER FOR EACH TRANSMITTER. THE APPROPRIATE BYTE GETS INCREMENTED DURING DEFERRED INTR. SERVICE AND CHECKED FOR EQUIVALENCE TO THE NUMBER OF CHARACTERS TRANSMITTED.

D. DLNT: 16 BYTE TABLE CONTAINING AN 8-BIT DATA COUNTER FOR EACH LINE. THE APPROPRIATE BYTE GETS INCREMENTED EACH TIME A CHARACTER IS TRANSMITTED ON THE LINE, AND CLEARED BEFORE THE BEGINNING OF EACH 64. WORD BURST.

E. TQ: 16 WORD FIFO QUEUE FOR TRANSMITTER SERVICE. LOADED DURING XMTR INTERRUPT SERVICE WITH A POINTER TO THE CSR ADDRESS AND UNLOADED DURING DEFERRED XMTR SERVICE.

F. XBUF: 16 BYTE XMTR DATA BUFFERS - ONE BYTE/XMTR

G. RBUF: 16 BYTE RCVR DATA BUFFERS - ONE BYTE/RCVR.

H. JSRTAB: A 128 WORD TABLE THAT CONTAINS 64 JSR INSTRUCTIONS WITH TWO TRAILING ARGUMENTS EACH RECEIVER AND EACH XMTR HAS AN ASSIGNED JSR IN THE TABLE OF THE FOLLOWING FORMAT:

```
JSR     R5,RINT(TINT)
  0
  N
```

WHERE THE 0 GETS OVERLAYED WITH THE ADDRESS OF THE CSR FOR LINE N AND N IS THE LINE NO. IN OCTAL (00 17)

OPERATOR OPTIONS:

A THE USER CAN USE THE "MOD" COMMAND TO DUMP THE TABLES BUFFERS DESCRIBED IN 7.2 TO OBTAIN MORE DETAILED ERROR INFORMATION.

B. THE USER CAN MODIFY (DLA 14) "DVID1 TO SELECT OR DESELECT INDIVIDUAL DL11'S.

C. SR1 IS USED TO SPECIFY THE BAUD RATE AND CHARACTER SIZE WITH WHICH THE DLA MODULE IS TO WORK. THE DEFAULT IS 9600 BAUD WITH AN 8-BIT CHARACTER SIZE.

1. CHAR. SIZE:

SR1	BIT1	BIT0	CHAR SIZE
	0	0	8-BIT
	0	1	7-BIT
	1	0	6-BIT
	1	1	5-BIT

2. BAUD RATE:
BITS 2-14 ARE USED. ONE AT MOST SHOULD BE SET; IF MORE THAN ONE BIT IS SET, THE RIGHT-MOST WILL BE USED.

BAUD	BIT SET
9600	NONE
7200	2
4800	3
2400	4
1800	5
1200	6
600	7
300	8
200	9
150	10
134.5	11
110	12
75	13
50	14

3. EXAMPLES:

SR1=5 MEANS 7-BIT CHARACTER AT 7200 BAUD
SR1=12 MEANS 6-BIT CHARACTER AT 4800 BAUD
SR1=1000 MEANS 8-BIT CHARACTER AT 200 BAUD
SR1=0 (DEFAULT) MEANS 8-BIT CHARACTER AT 9600 BAUD

4. CHANGE ITERATION CONSTANT (ICONT) USING SR2: ;RS001

BIT15 = 1 SET ITERATION CONSTANT (ICONT) WITH VALUE LOADED INTO BITS 14:0.

BIT15 = 0 ICONT SET FROM CNTTBL ACCORDING TO THE BAUD RATE SELECTED BY SR1.

*NOTE : CHARACTER SIZE UNAFFECTED BY SR2.

9. NON STANDARD PRINTOUTS:

THERE ARE TWO ERROR PRINTOUTS WHICH SUPPLY SPECIAL INFORMATION
IN THE CSRC AND STATC VALUES (CONSULT LISTING).

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
;..
;*****************************************************************************
;
;       EDIT    BY              DATE            REASON
;
;
;       00'     R. SOLER        6-JUL-83        COULD NOT ADJUST ITERATION
;                                               CONSTANT TO ALLOW FOR QUICK
;                                               TEST RUNS. THIS CHANGE AFFECTS
;                                               REV. K.
;
;       002     R. GAUDET       12-NOV-1984     FIX THE FRAMING ERROR PROBLEM.
;                                               INCLUDED CODE TO TEST FOR THE
;                                               OVERRUN, FRAMING, AND PARITY
;                                               ERRORS.
;
;*****************************************************************************
;
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
383                                     ;DL11 A-D DEC/X11 EXERCISER MODULE
384
385 000000                                      IOMOD   <DLAL >,176500,1,4,,,0,26
    000000                                      MODULE  140000,DLAL ,176500,1,4,,,0,26,,,,,
                                                .TITLE  DLAL DEC/X11 SYSTEM EXERCISER MODULE
                                        ;       DDXCOM  VERSION 6.4     28-JAN 82
                                                .LIST   BIN
                                        ;*****************************************************************************
    000000                              BEGIN:
    000000      104     114     101     MODNAM: .ASCII  /DLAL / ;MODULE NAME.
    000003      114     040
    000005      000                     XFLAG:  .BYTE   OPEN                    ;USED TO KEEP TRACK OF WBUFF USAGE
    000006  176500                      ADDR:   176500+0                        ;1ST DEVICE ADDR.
    000010  000001                      VECTOR: 1+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  000000                      ICON:   0                               ;# OF ITERATIONS PER PASS=0
    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:                                 ;RESERVED FOR MONITOR USE
    000056  000000                      RES1:   0                               ;RESERVED FOR MONITOR USE
    000060  000000                      RES2:   0                               ;RESERVED FOR MONITOR USE
    000062  000000                      SVR0:   OPEN                            ;LOC TO SAVE R0.
    000064  000000                      SVR1:   OPEN                            ;LOC TO SAVE R1.
    000066  000000                      SVR2:   OPEN                            ;LOC TO SAVE R2.
    000070  000000                      SVR3:   OPEN                            ;LOC TO SAVE R3.
    000072  000000                      SVR4:   OPEN                            ;LOC TO SAVE R4.
    000074  000000                      SVR5:   OPEN                            ;LOC TO SAVE R5.
    000076  000000                      SVR6:   OPEN                            ;LOC TO SAVE R6.
    000100  000000                      CSRA:   OPEN                            ;ADDR OF CURRENT CSR.
    000102                              SBADR:                                  ;ADDR OF GOOD DATA, OR
    000102  000000                      ACSR:   OPEN                            ;CONTENTS OF CSR.
    000104                              WASADR:                                 ;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  000552                      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  000026                      IDNUM:  26                              ;MODULE IDENTIFICATION NUMBER=26
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

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

    386
    387                                 ;+
    388                                 ; THE NEXT LOCATION  CONTAINS A NUMBER  THAT CAN BE EXAMINED IN CORE TO  BE
    389                                 ; SURE THAT THE LISTING AND ACTUAL CODE ARE THE SAME.  THIS  LOCATION IS TO
    390                                 ; BE INCREMENTED  EACH TIME THE SOURCE CODE IS UPDATED.  LOCATION  'MODREV'
    391                                 ; IS NOT USED BY THE PROGRAM.
    392                                 ;
    393
    394 000224  000014                  MODREV: 1.1.1.1.1.1.1.1.1.1.1.1.0
    395
```

```
                                        ;THIS ROUTINE SETS UP THE VECTORS FOR ALL SELECTED LINES TO POINT
                                        ;TO THE APPROPRIATE JSR IN THE JSR LINK TABLE AND SETS
                                        ;THE ITERATION COUNT TO MATCH BAUD RATE SPEC. IN SR1
000226  005000                  START:  CLR     R0              ;GET RID OF JUNK        (REV.J0)
000230  012767  000100  177656          MOV     #64.,WDTO       ;64 (10) WORDS PER ITERATION TO MEM
000236  012767  000100  177652          MOV     #64.,WDFR       ;64 (10) WORDS FROM MEM PER ITERATION
000244  012767  000200  177646          MOV     #128.,INTR      ;64 INTERRUPTS PER ITERATION
000252  016701  177540                  MOV     SR1,R1          ;GET SR1 VALUE
000256  042701  177774                  BIC     #177774,R1      ;MASK OUT SIZE PART
000262  001406                          BEQ     8$              ;DEFAULT SETTING
000264  112700  000200                  MOVB    #200,R0         ;SET UP MASK FOR 7-BIT
000270  005301                  7$:     DEC     R1              ;COUNT BITS MASKED
000272  001402                          BEQ     8$              ;EXIT WHEN DONE
000274  006200                          ASR     R0              ;SHIFT AND REPLICATE HIGH BIT
000276  000774                          BR      7$              ;AGAIN?
000300  010067  002524          8$:     MOV     R0,SIZE         ;SET SIZE MASK
000304  005000                          CLR     R0              ;INIT POINTER
000306  016701  177504                  MOV     SR1,R1          ;GET SR1
000312  006201                          ASR     R1              ;SHIFT SR1 TO
000314  006201                          ASR     R1              ;GET BAUD RATE
000316  001405                          BEQ     19$             ;9600 BAUD SET
000320  062700  000002          18$:    ADD     #2,R0           ;ELSE BUMP POINTER
000324  006001                          ROR     R1              ;LOOK FOR SR1 BIT
000326  103401                          BCS     19$             ;LEAVE IF WE FOUND IT
000330  000773                          BR      18$             ;ELSE DO IT AGAIN
000332  005767  177462          19$:    TST     SR2             ;IF BIT15 SET, LOAD ITERATION CONSTANT IN SR2   ;RS001
000336  001407                          BEQ     20$             ;ELSE LOAD FROM CNTTBL.                         ;RS001
000340  016704  177454                  MOV     SR2,R4          ;SAVE SR2 IN WORK AREA                          ;RS001
000344  042704  100000                  BIC     #100000,R4      ;MASK BIT15                                     ;RS001
000350  010467  177462                  MOV     R4,ICONT        ;LOAD SELECTED ITERATION CONSTANT               ;RS001
000354  000403                          BR      17$             ;BYPASS CNTTBL ICONT VALUES                     ;RS001
000356  016067  001650' 177452  20$:    MOV     CNTTBL(R0),ICONT                                                ;RS001
                                                                ;SET UP ITER COUNT
000364  016700  177420          17$:    MOV     VECTOR,R0       ;SET R0 TO POINT TO THE 1ST VECTOR
000370  016701  177420                  MOV     DVID1,R1        ;LOAD R1 WITH DEVICE SELECTION PARAMETER
000374  010167  002574                  MOV     R1,DVICE        ;INITIALIZE TO RECORD ANY LINES DROPPED
000400  012702  003176'                 MOV     #JSRTAB,R2      ;SET UP R2 TO POINT TO JSR TABLE
000404  006201                  1$:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
000406  103020                          BCC     3$              ;BR IF NOT SELECTED
000410  010220                          MOV     R2,(R0)+        ;SET UP RCVR INTR POINTER
000412  116720  177374                  MOVB    BR1,(R0)+       ;SET UP RCVR PRIORITY LEVEL
000416  105720                          TSTB    (R0)+           ;MOVE POINTER
000420  062702  000010                  ADD     #10,R2          ;POINT R2 TO XMTR ENTRY IN JSR TABLE
000424  010220                          MOV     R2,(R0)+        ;SET UP XMTR INTR POINTER
000426  116720  177360                  MOVB    BR1,(R0)+       ;SET UP XMTR PRIORITY LEVEL
000432  105720                          TSTB    (R0)+           ;MOVE POINTER
000434  062702  000010                  ADD     #10,R2          ;POINT R2 TO RCVR ENTRY FOR NEXT LINE
000440  022702  003576'          2$:    CMP     #JSRTAB+400,R2  ;IS THE POINTER AT THE END OF THE TABLE?
000444  001357                          BNE     1$              ;BR IF NOT
000446  000405                          BR      SETCSR          ;GO SET UP CSR ADDRESSES
000450  062700  000010          3$:     ADD     #10,R0          ;UPDATE VECTOR POINTER
000454  062702  000020                  ADD     #20,R2          ;UPDATE JSR TABLE POINTER
000460  000767                          BR      2$              ;GO CHECK FOR END OF TABLE

                                        ;THIS ROUTINE SETS UP THE JSR TABLE SUCH THAT THE APPROPRIATE
                                        ;CSR ADDRESS IS INCLUDED AS THE 3RD WORD OF EACH ENTRY
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
454 000462  016700  177320          SETCSR: MOV     ADDR,R0         ;GET THE FIRST CSR ADDRESS INTO R0
455 000466  016701  177322                  MOV     DVID1,R1        ;LOAD R1 WITH THE DEVICE SELECTION PARAMETER
456 000472  001002                          BNE     1$              ;BR IF DVC OK
457 000474  104410  000000                  END$,BEGIN              ;
458 000500  012702  003202          1$:     MOV     #JSRTAB+4,R2    ;POINT R2 TO CSR ADDRESS ENTRY
459 000504  006201                  2$:     ASR     R1              ;SHIFT SELECT BIT INTO "C"
460 000506  103014                          BCC     4$              ;BR IF LINE NOT SELECTED
461 000510  010012                          MOV     R0,(R2)         ;PUT RCVR CSR ADDRESS IN TABLE
462 000512  022020                          CMP     (R0)+,(R0)+     ;GENERATE XMTR CSR ADDRESS IN R0
463 000514  062702  000010                  ADD     #10,R2          ;POINT TO XMTR SLOT IN JSR TABLE
464 000520  010012                          MOV     R0,(R2)         ;PUT XMTR CSR ADDRESS IN THE TABLE
465 000522  022020                          CMP     (R0)+,(R0)+     ;GENERATE NEXT RCVR CSR ADDRESS IN R0
466 000524  062702  000010                  ADD     #10,R2          ;POINT TO RCVR SLOT IN JSR TABLE
467 000530  022702  003602          3$:     CMP     #JSRTAB+404,R2  ;IS POINTER BEYOND END OF TABLE?
468 000534  001363                          BNE     2$              ;BR IF NOT
469 000536  000411                          BR      REST            ;GO SET UP ACTIVE DEVICE TABLE.
470 000540  062700  000010          4$:     ADD     #10,R0          ;UPDATE CSR ADDRESS
471 000544  062702  000020                  ADD     #20,R2          ;UPDATE JSR TABLE POINTER
472 000550  000767                          BR      3$              ;GO TEST FOR END OF TABLE
473
474                                 ;THIS ROUTINE CLEARS BUFFERS AND TABLES, INITIALIZES FLAGS, AND STARTS
475                                 ;UP ALL SELECTED LINES.
476
477 000552  005000          RESTRT: CLR     R0              ;GET RID OF JUNK        (REV.J0)
478 000554  005767  177254          TST     PASCNT          ;HAVE WE BEEN THRU START?
479 000560  001622                  BEQ     START           ;BR THERE IF WE HAVEN'T
480 000562  004767  002262  REST:   JSR     PC,CLRBUF       ;GO CLEAR XMTR. AND RCVR. BUFFERS
481 000566  004767  002312  STUP1:  JSR     PC,DTAB         ;SET UP THE ACTIVE DEVICE TABLE
482 000572  004767  002270          JSR     PC,CLRTAB       ;GO CLEAR TABLES AND QUEUES
483 000576  005067  002232          CLR     TXCNT           ;CLEAR TX TOTAL INTERRUPT COUNTER.
484 000602  005067  002230          CLR     RXCNT           ;CLEAR RX TOTAL INTERRUPT COUNTER
485 000606  012767  002250' 002226  MOV     #RSTAB,SVPTR    ;INITIALIZE RCVR. STATUS TABLE POINTER
486 000614  012767  002730' 00 222  MOV     #TQ,QPTR1       ;SET UP XMTR FIFO QUEUE POINTERS
487 000622  012767  002730' 00 216  MOV     #TQ,QPTR2
488 000630  016700  002336          MOV     ACTDEV,R0       ;GET COUNT OF ACTIVE DEVICES
489 000634  116002  003152'  1$:     MOVB    DEVTAB(R0),R2   ;GET AN ACTIVE LINE NO.
490 000640  004767  001244          JSR     PC,GETADR       ;GO BUILD CSR ADDRESS IN R3
491 000644  005763  000002          TST     2(R3)           ;READ RCV DBR TO FLUSH DONE BI
492 000650  052713  000100          BIS     #100,(R3)       ;ENABLE RECEIVER INTERRUPTS
493 000654  052763  000004  000004  BIS     #4,4(R3)        ;ENABLE MAINT. MODE
494 000662  105262  002770'          INCB    XBUF(R2)        ;
495 000666  116263  002770' 000006  MOVB    XBUF(R2),6(R3)  ;OUTPUT CHAR ONTO TX.
496 000674  105267  002134          INCB    TXCNT           ;UP COUNT OF CHARS OUTPUT.
497 000700  105262  002710'          INCB    DCNT(R2)        ;COUNT CHARACTERS OUTPUT ON THIS LINE
498 000704  052763  000100  000004  BIS     #100,4(R3)      ;ENABLE TX INTERRUPTS.
499 000712  005300                  DEC     R0              ;COUNT ONE KICKED OFF
500 000714  100347                  BPL     1$              ;GO TEST FOR NEXT ONE
501 000716  012767  000006  002114  MOV     #6,CNTR         ;INITIALIZE COUNTER TO WAIT AT LEAST
502                                                         ;30 SECONDS BEFORE TIMING OUT
503 000724  005004          10$:    CLR     R4
504 000726                  2$:
    000726  104407  000000'          BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
    000732  104407  000000'          BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
505 000736  122767  000100  002071  CMPB    #64.,TXCNT+1    ;64 TRANSMITTER INTERRUPTS?
506 000744  003004                  BGT     3$              ;NO  BRANCH TO WAIT
507 000746  022767  000100  002062  CMP     #64 ,RXCNT      ;YES  64 RECEIVER INTERRUPTS?
508 000754  003405                  BLE     4$              ,YES  GO CHECK FOR ERRORS
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
000756  005304                  3$:     DEC     R4              ;TIMEOUT?
000760  001362                          BNE     2$              ;NO- WAIT SOME MORE
000762  005367  002052                  DEC     CNTR            ;EACH PASS OF THE SMALL LOOP TAKES
                                                                ;AT LEAST 5 SECONDS
000766  001356                          BNE     10$             ;BRANCH IF NOT DONE WITH 6 PASSES OF
                                                                ;THE SMALL COUNTER
000770  000167  000306          4$:     JMP ERRCHK

                                ;TRANSMITTER INTERRUPT SERVICE - ENTERED VIA APPROPRIATE JSR TABLE
                                ;ENTRY WITH R5 POINTING TO THE CSR ADDRESS. - CONTENTS OF R5
                                ;GETS QUEUED UP IN FIFO QUEUE AND ROUTINE RETURNS CONTROL BACK TO
                                ;THE MONITOR VIA A PIRQ TO DEFER SERVICING XMITTER AT LEVEL 0

000774  010577  002044          TINT:   MOV     R5,@QPTR1       ;STORE CONTENTS OF R5 IN THE QUEUE
001000  062767  000002  002036          ADD     #2,QPTR1        ;UPDATE THE QUEUE POINTER
001006  022767  002770' 002030          CMP     #TQ+40,QPTR1    ;POINTER AT END OF QUEUE?
001014  001003                          BNE     1$              ;BR IF NOT
001016  012767  002730' 002020          MOV     #TQ,QPTR1       ;RESET THE POINTER
001024  012605                  1$:     MOV     (R6)+,R5        ;RESTORE THE OTHER GUY'S R5
                                        ;-------------------------------------------- -- --  -- --
001026  000004  000000' 001034'         PIRQ$,BEGIN,TSERV               ; QUEUE UP TO CONTINUE AT TSERV AND RTI
                                        ;------------------- --- ---- --------- - --- ------  - --

                                ;DEFERRED XMTR SERVICE - THIS ROUTINE RETRIEVES POINTER TO CSR ADDRESS
                                ;FROM THE FIFO QUEUE AND SERVICES THE LINE AT LEVEL 0

001034  017700  002006          TSERV:  MOV     @QPTR2,R0       ;GET POINTER FROM THE QUEUE
001040  062767  000002  002000          ADD     #2,QPTR2        ;UPDATE THE QUEUE POINTER
001046  022767  002770' 001772          CMP     #TQ+40,QPTR2    ;POINTER AT HIGH LIMIT
001054  001003                          BNE     1$              ;BR IF NOT
001056  012767  002730' 001762          MOV     #TQ,QPTR2       ;RESET THE POINTER
001064  012001                  1$:     MOV     (R0)+,R1        ;MOV CSR ADDRESS INTO R1
001066  011000                          MOV     (R0),R0         ;MOV LINE # INTO R0
001070  105267  001741                  INCB    TXCNT+1         ;COUNT TOTAL TX INTERRUPTS.
001074  105260  002670'                 INCB    TCNT(R0)        ;COUNT THE INTERRUPT
001100  105711                          TSTB    (R1)            ;XMTR READY FLAG ASSERTED?
001102  100017                          BPL     4$              ;BR IF NOT
001104  122767  000100  001722          CMPB    #64.,TXCNT      ;64 CHARACTERS TRANSMITTED?
001112  001427                          BEQ     5$              ;YES- BRANCH TO EXIT
001114  105260  002770'                 INCB    XBUF(R0)        ;GENERATE NEXT DATA BYTE
001120  116061  002770' 000002          MOVB    XBUF(R0),2(R1)  ;LOAD THE XMTR BUFFER
001126  105267  001702                  INCB    TXCNT           ;UP TOTAL COUNT OF CHARS OUTPUT.
001132  105260  002710'                 INCB    DCNT(R0)        ;COUNT CHARACTERS OUTPUT ON THIS LINE
001136  104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001142  010167  176732          4$:     MOV     R1,CSRA         ;SAVE CSR ADDRESS
001146  011167  176730                  MOV     (R1),ACSR       ;SAVE CONTENTS OF THE CSR
001152  142711  000100                  BICB    #100,(R1)       ;DISABLE XMTTR INTERRUPT
001156  012767  000011  176722          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*****************************************************************
001164  104405  000000  000000         HRDER$,BEGIN,NULL       ;XMTTR FALSE INTERRUPT
                                        ;*****************************************************************
001172                          5$:
001172  104400  000000                  EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                                ;RECEIVER INTERRUPT SERVICE-ENTERED VIA APPROPRIATE JSR TABLE ENTRY
                                ;STORES PERTINENT INFORMATION IN THE RECEIVER STATUS TABLE THAT WILL
                                ;BE CHECKED IF 64. CHARACTERS HAVE BEEN RECEIVED
```

```
561
562 001176  010046                    RINT:   MOV     R0,-(R6)        ;SAVE R0 AND R1 ON THE STACK
563 001200  010146                            MOV     R1,-(R6)
564 001202  022767  000100  001626            CMP     #64.,RXCNT      ;64 CHAR'S RECEIVED?
565 001210  003422                            BLE     1$              ;YES- BRANCH
566 001212  016700  001624                    MOV     SVPTR,R0        ;NO- GET RCVR STATUS TABLE POINTER
567 001216  011501                            MOV     (R5),R1         ;GET RCVR CSR ADDRESS
568 001220  011120                            MOV     (R1),(R0)+      ;SAVE THE RCSR CONTENTS
569 001222  106260  177777                    ASRB    -1(R0)          ;SHIFT ERROR WORD
570 001226  156160  000003  177777            BISB    3(R1),-1(R0)    ;GET MORE ERROR STATUS
571 001234  116120  000002                    MOVB    2(R1),(R0)+     ;SAVE THE RCVD DATA
572 001240  156160  000002  177775            BISB    2(R1),-3(R0)    ;REREAD CSR IN CASE ERROR SET BETWEEN THE
573                                                                   ;LAST TWO INSTRUCTIONS
574 001246  116520  000002                    MOVB    2(R5),(R0)+     ;SAVE THE LINE NUMBER
575 001252  010067  001564                    MOV     R0,SVPTR        ;SAVE THE UPDATED STATUS TABLE POINTER
576 001256  016505  000002            1$:     MOV     2(R5),R5        ;GET LINE NO. INTO R5
577 001262  105265  002650                    INCB    RCNT(R5)        ;COUNT THE INTERRUPT FROM THIS LINE
578 001266  005267  001544                    INC     RXCNT           ;INCREMENT RX INTERRUPT TOTAL COUNT.
579 001272  012601                    2$:     MOV     (R6)+,R1        ;RESTORE THE OTHER GUY'S REGISTER
580 001274  012600                            MOV     (R6)+,R0
581 001276  012605                            MOV     (R6)+,R5
582 001300  000002                            RTI                     ;RETURN CONTROL BACK TO OTHER GUY
583
584                                   ;RECEIVER ERROR CHECKING AND CLEANUP ROUTINES
585
586                                   ;THIS ROUTINE DISABLES INTERRUPTS FROM ALL ACTIVE LINES
587
588 001302  016700  001664            ERRCHK: MOV     ACTDEV,R0       ;GET COUNT OF ACTIVE DEVICES
589 001306  116002  003152'           1$:     MOVB    DEVTAB(R0),R2   ;GET ACTIVE LINE NO.
590 001312  004767  000572                    JSR     PC,GETADR       ;GO BUILD ADDRESS IN R3
591 001316  042713  000100                    BIC     #100,(R3)       ;TURN OFF RECEIVER.
592 001322  042763  000100  000004            BIC     #100,4(R3)      ;TURN OFF TRANSMITTER.
593 001330  005300                            DEC     R0              ;COUNT ONE GUY OFF
594 001332  100365                            BPL     1$              ;BR TIL ALL OFF
595
596                                   ;THIS ROUTINE SCANS THROUGH THE 64 ENTRY RECEIVER STATUS TABLE
597                                   ;CHECKING FOR AND REPORTING ANY ERRORS
598
599 001334  026727  001476  000100    CHK1:   CMP     RXCNT,#64.      ;MAKE SURE COUNT IS NO LARGER THAN TABLE
600 001342  003403                            BLE     1$
601 001344  012767  000100  001464            MOV     #64.,RXCNT
602 001352  012701  002250'           1$:     MOV     #RSTAB,R1       ;GET STATUS TABLE POINTER
603 001356  005000                    2$:     CLR     R0              ;INDICATE NO HARDWARE FAILURES YET.
604 001360  116102  000003                    MOVB    3(R1),R2        ;GET LINE NO. INTO R2
605 001364  005711                            TST     (R1)            ;ERROR SET?
606 001366  100047                            BPL     5$              ;BRANCH IF NOT                          ;RG002
607 001370  011104                            MOV     (R1),R4         ;GET THE DEVICE STATUS INFORMATION      ;RG002
608 001372  004767  000602                    JSR     PC,RCVERR       ;GO SETUP TO REPORT ERROR
609 001376  005067  176504                    CLR     ERRTYP
610 001402  006104                            ROL     R4              ;PLACE THE OVERRUN BIT...               ;RG002
611 001404  006104                            ROL     R4              ;...IN THE CARRY BIT                    ;RG002
612 001406  103011                            BCC     3$              ;IF NO ERROR, GO CHECK THE NEXT BIT     ;RG002
613 001410  012767  000021  176470            MOV     #21,ERRTYP      ;OVERRUN ERROR                          ;RG002
614                                           ;*****************************************************************
    001416  104406  000000' 000000           SOFER$,BEGIN,NULL       ;OVERRUN ERROR
                                              ;*****************************************************************
615 001424  104403  000000  001706           MSGN$,BEGIN,OVERUN      ;ASCII MESSAGE CALL WITH COMMON HEADER
```

DEC/x11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
                                3$:     ROL     R4              ;PLACE THE FRAME BIT IN THE CARRY BIT            ;RG002
                                        BCC     4$              ;IF NO ERROR, GO CHECK THE NEXT BIT              ;RG002
                                        MOV     #22,ERRTYP      ;FRAMING ERROR                                   ;RG002
                                        ;*****************************************************************
001444  104406  000000' 000000        SOFER$,BEGIN,NULL       ;FRAMING ERROR
                                        ;*****************************************************************
                                        MSGN$,BEGIN,FRAME       ;ASCII MESSAGE CALL WITH COMMON HEADER
                                4$:     ROL     R4              ;CHECK FOR A PARITY ERROR                        ;RG002
                                        BCC     5$              ;IF NO ERROR, GO RESTORE THE REGISTERS           ;RG002
                                        MOV     #23,ERRTYP      ;PARITY ERROR                                    ;RG002
                                        ;*****************************************************************
001472  104406  000000' 000000        SOFER$,BEGIN,NULL       ;PARITY ERROR
                                        ;*****************************************************************
                                        MSGN$,BEGIN,PARITY      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                5$:     TSTB    (R1)            ;POINT TO LO BYTE OF CSR                         ;RG002
                                        BMI     6$              ;BR IF DONE WAS SET                              ;RG002
                                        JSR     PC,RCVERR       ;SETUP FOR ERROR REPORT
                                        MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*****************************************************************
001524  104405  000000' 000000        HRDER$,BEGIN,NULL       ;RECEIVER FALSE INTERRUPT
                                        ;*****************************************************************
                                6$:     INCB    RBUF(R2)        ;BUMP EXPECTED DATA                              ;RG002
                                        TST     R0              ;HARDWARE ERRORS?
                                        BNE     7$              ;DO NOT REPORT DATA ERRORS THEN.                 ;RG002
                                        BICB    SIZE,RBUF(R2)   ;MASK OFF BITS <7:5> TO CHECK ONLY
                                        BICB    SIZE,2(R1)      ;  SPECIFIED SIZE.
                                        CMPB    RBUF(R2),2(R1)  ;DID RCVD DATA CHECK OK?
                                        BEQ     7$              ;BR IF YES                                       ;RG002
                                        JSR     PC,DATBAD       ;GO REPORT DATA ERROR
                                7$:     CMP     (R1)+,(R1)+     ;POINT R1 TO NEXT TABLE ENTRY                    ;RG002
                                        DEC     RXCNT           ;ALL CHARS RECEIVED CHECKED?
                                        BNE     2$              ;NO- GO CHECK NEXT ENTRY

                                ;THIS ROUTINE REPORTS ANY LINE RECEIVING AN INCORRECT NUMBER OF INTERRUPTS

                                CKLINS: MOV     ACTDEV,R1       ;GET ACTIVE DEVICE COUNT
                                3$:     MOVB    DEVTAB(R1),R2   ;GET ACTIVE DEVICE LINE NO.
                                        CMPB    RCNT(R2),DCNT(R2)
                                                                ;CORRECT NUMBER OF RECVR INTERRUPTS?
                                        BEQ     4$              ;BR IF YES
                                        JSR     PC,BADR         ;GO REPORT BAD RCVR
                                4$:     CMPB    TCNT(R2),DCNT(R2)
                                                                ;CORRECT NUMBER OF XMTR INTERRUPTS?
                                        BEQ     5$              ;BR IF YES
                                        JSR     PC,BADT         ;GO REPORT BAD XMTR
                                5$:     DEC     R1              ;COUNT ONE GUY CHECKED
                                        BPL     3$              ;BR TIL ALL CHECKED
                                        BR      ENPS            ;GO CHECK FOR END OF PASS

                                ;TABLE USED FOR DIFFERENT BAUD RATES

                                CNTTBL: 1450
                                        1140
                                         621
                                         310
                                         230
                                         141
```

```
667 001664  000065                          65
668 001666  000030                          30
669 001670  000020                          20
670 001672  000014                          14
671 001674  000013                          13
672 001676  000011                          11
673 001700  000006                           6
674 001702  000004                           4
675 001704  000004                           4      ;NOT A LEGAL SR1 SETTING
676
677 001706  001712                  OVERUN: M.OVER                                          ;RG002
678 001710  177777                          -1                                              ;RG002
679                                                                                         ;RG002
680 001712     045     117     126  M.OVER: .ASCIZ  /%OVERRUN ERROR%/                       ;RG002
    001715     105     122     122
    001720     125     116     040
    001723     105     122     122
    001726     117     122     045
    001731     000
681                                         .EVEN                                           ;RG002
682                                                                                         ;RG002
683 001732  001736'                 FRAME:  M.FRAM                                          ;RG002
684 001734  177777                          -1                                              ;RG002
685                                                                                         ;RG002
686 001736     045     106     122  M.FRAM: .ASCIZ  /%FRAMING ERROR%/                       ;RG002
    001741     101     115     111
    001744     116     107     040
    001747     105     122     122
    001752     117     122     045
    001755     000
687                                         .EVEN                                           ;RG002
688                                                                                         ;RG002
689 001756  001762'                 PARITY: M.PAR                                           ;RG002
690 001760  177777                          -1                                              ;RG002
691                                                                                         ;RG002
692 001762     045     120     101  M.PAR:  .ASCIZ  /%PARITY ERROR%/                        ;RG002
    001765     122     111     124
    001770     131     040     105
    001773     122     122     117
    001776     122     045     000
693                                         .EVEN                                           ;RG002
694
695                                 ;ROUTINE TO REPORT BAD LINES (TOO MANY OR TOO FEW INTERRUPTS)
696
697 002002  004767  000102          BADR:   JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
698 002006  010367  176066                  MOV     R3,CSRA         ;SAVE CSR ADDRESS
699 002012  116267  002710' 176062         MOVB    DCNT(R2),ACSR   ;CHARACTERS XMTD
700 002020  116267  002650' 176056         MOVB    RCNT(R2),ASTAT  ;# OF RCVR INTERRUPTS
701
702 002026  012767  000014  176052         MOV     #14,ERRTYP      ;WRONG # OF INTERRUPTS
703                                         ;*****************************************************************
    002034  104405  000000' 000000         HRDER#,BEGIN,NULL       ;INCORRECT NUMBER OF RCVR INTERRUPTS
                                            ;*****************************************************************
704                                         ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
705                                         ;TRANSMITTED, AND STATC VALUE IS # OF
706                                         ;RECEIVER INTERRUPTS
```

```
                                        RTS     PC              ;RETURN TO CALLER
                                BADT:   JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
                                        CMP     (R3)+,(R3)+     ;MAKE IT A XMTR CSR ADDRESS
                                        MOV     R3,CSRA         ;SAVE CSR ADDRESS
                                        MOVB    DCNT(R2),ACSR   ;CHARACTERS XMITTED
                                        MOVB    TCNT(R2),ASTAT  ;# OF XMTR INTERRUPTS
                                        MOV     #14,ERRTYP      ;WRONG # OF INTERRUPTS
                                        ;*****************************************************************
002100  104405  000000' 000000          HRDER$,BEGIN,NULL       ;INCORRECT NUMBER OF XMTR INTERRUPTS
                                        ;*****************************************************************
                                        ;NOTE THAT CSRC VALUE IS # OF CHARACTERS
                                        ;TRANSMITTED, AND STATC VALUE IS # OF
                                        ;TRANSMITTER INTERRUPTS

002106  000207                          RTS     PC              ;RETURN TO CALLER

002110  010203                  GETADR: MOV     R2,R3           ;GET LINE NO.
002112  006303                          ASL     R3              ;BUILD CSR ADDRESS
002114  006303                          ASL     R3
002116  006303                          ASL     R3
002120  066703  175662                  ADD     ADDR,R3
002124  000207                          RTS     PC              ;RETURN TO CALLER

                                ;ROUTINE TO REPORT RCVR DATA COMPARE ERRORS

002126  004767  177756          DATBAD: JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
002132  010367  175742                  MOV     R3,CSRA         ;SAVE CSR ADDRESS
002136  116167  000002  175744          MOVB    2(R1),AWAS      ;SAVE BAD DATA
002144  005721                          TST     (R1)+           ;GENERATE RCVR DATA ADDRESS
002146  010167  175732                  MOV     R1,WASADR       ;SAVE ADDRESS OF BAD DATA
002152  005741                          TST     -(R1)           ;RESET R1
002154  012705  003010'                 MOV     #RBUF,R5        ;GENERATE ADDRESS OF GOOD DATA
002160  060205                          ADD     R2,R5
002162  111567  175720                  MOVB    (R5),ASB        ;SAVE GOOD DATA
002166  010567  175710                  MOV     R5,SBADR        ;SAVE ADDRESS OF GOOD DATA
                                        ;*****************************************************************
002172  104404  000000                  DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************
002176  000207                          RTS     PC              ;RETURN TO CALLER

                                ;ROUTINE TO SETUP FOR RECEIVER ERROR PRINTOUTS

002200  005200                  RCVERR: INC     R0              ;INDICATE HARDWARE ERROR.
002202  004767  177702                  JSR     PC,GETADR       ;GO BUILD CSR ADDRESS
002206  010367  175666                  MOV     R3,CSRA         ;STUFF IT IN CSRA
002212  011167  175664                  MOV     (R1),ACSR       ;GET CONTENTS IN ACSR
002216  000207                          RTS     PC              ;RETURN TO CALLER

                                ;THIS ROUTINE CHECKS FOR AND REPORTS END OF PASS
002220                          ENPS:
002220  104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS

                                ;THIS ROUTINE RESTARTS EACH 64 CHAR XFR SEQUENCE
```

```
  759 002224  012700  003010'         RESYNC: MOV     #RBUF,R0        ;RESYNC DATA FOR NEXT PASS
  760 002230  012701  002770                 MOV     #XBUF,R1
  761 002234  112021                  1$:    MOVB    (R0)+,(R1)+
  762 002236  022700  003030                 CMP     #RBUF+20,R0     ;DONE 16 BYTES?
  763 002242  001374                         BNE     1$              ;BR IF NOT
  764 002244  000167  176316                 JMP     STUP1           ;RESUME.
  765
  766                                 ;TABLES AND BUFFERS
  767
  768 002250                          RSTAB:  .BLKW   128.            ;128 WORD(64 ENTRIES)RCVR STATUS TABLE
  769 002650                          RCNT:   .BLKW   8.              ;RCVR INTERRUPT COUNTERS
  770 002670                          TCNT:   .BLKW   8.              ;XMTR INTERRUPT COUNTERS
  771 002710                          DCNT:   .BLKW   8.              ;CHARACTER COUNTERS
  772
  773 002730                          TQ:     .BLKW   16.             ;16 WORD XMTR FIFO QUEUE
  774
  775 002770                          XBUF:   .BLKW   8.              ;16 BYTE XMTR DATA BUFFERS
  776 003010                          RBUF:   .BLKW   8.              ;16 BYTE RCVR DATA BUFFERS
  777
  778                                 ;POINTERS, CONSTANTS, AND VARIABLES
  779
  780 003030  000000                  SIZE:   OPEN                    ;LOW BYTE IS MASK FOR VARIABLE 5-8 BIT COMPARE
  781 003032  000000                  COUNT:  OPEN                    ;END OF PASS COUNTER
  782 003034  000000                  TXCNT:  OPEN                    ;TX TOTAL INTERRUPTS COUNTER (HIGH BYTE).
  783                                                                 ;TOTAL CHARACTERS TRANSMITTED (LOW BYTE)
  784 003036  000000                  RXCNT:  OPEN                    ;RX TOTAL INTERRUPTS COUNTER.
  785 003040  000000                  CNTR:   OPEN                    ;BREAK LOOP COUNTER
  786 003042  000000                  SVPTR:  OPEN                    ;TEMP STORAGE FOR RSTAB POINTER
  787 003044  000000                  QPTR1:  OPEN                    ;XMTR FIFO QUEUE POINTER   LOAD
  788 003046  000000                  QPTR2:  OPEN                    ;XMTR FIFO QUEUE POINTER   UNLOAD
  789
  790
  791                                 ;SUBROUTINE TO CLEAR DATA BUFFERS AT BEGINING OF EACH NEW PASS
  792
  793 003050  012700  002770'         CLRBUF: MOV     #XBUF,R0        ;SET UP R0 TO POINT TO BEGINING
  794 003054  005020                  1$:     CLR     (R0)+           ;CLEAR A WORD
  795 003056  022700  003030'                 CMP     #RBUF+20,R0     ;END OF RCVR BUFFER?
  796 003062  001374                          BNE     1$              ;BR TIL ALL CLEAR
  797 003064  000207                          RTS     PC              ;RETURN TO CALLER
  798
  799                                 ;SUBROUTINE TO CLEAR TABLES AND QUEUES
  800
  801 003066  012700  002250'         CLRTAB: MOV     #RSTAB,R0       ;SET UP R0 TO POINT TO BEGINING
  802 003072  005020                  1$:     CLR     (R0)+           ;CLEAR A WORD
  803 003074  022700  002770'                 CMP     #TQ+40,R0       ;END?
  804 003100  001374                          BNE     1$              ;BR IF NOT
  805 003102  000207                          RTS     PC              ;RETURN TO CALLER
  806
  807                                 ;THIS ROUTINE SETS UP AN ACTIVE DEVICE TABLE TO REMEMBER HOW MANY
  808                                 ;AND WHICH LINES WERE ACTIVE DURING TEST - IT IS USED DURING THE
  809                                 ;ERROR CHECKING ROUTINES FOR VARIOUS PURPOSES
  810
  811 003104  005000                  DTAB:   CLR     R0              ;SET UP R0 AS TOTAL LINE COUNTER
  812 003106  005100                          COM     R0              ;INITIALLY SET TO MINUS ONE
  813 003110  005001                          CLR     R1              ;SET UP R1 AS ACTIVE LINE COUNTER
  814 003112  005101                          COM     R1              ;INITIALLY SET TO MINUS ONE
  815 003114  016702  000054                  MOV     DVICE,R2        ;GET DEVICE SELECTION PARAMETER
```

```
816 003120  005200            1$:     INC     R0              ;COUNT ONE DEVICE
817 003122  022700  000020            CMP     #20,R0          ;16 LINES CHECKED?
818 003126  001003                    BNE     2$              ;BR IF NOT
819 003130  010167  000036            MOV     R1,ACTDEV       ;SAVE THE COUNT OF ACTIVE LINES
820 003134  000207                    RTS     PC              ;RETURN TO CALLER
821 003136  006202            2$:     ASR     R2              ;SHIFT SELECT BIT INTO "C"
822 003140  103367                    BCC     1$              ;BR IF NOT SELECTED
823 003142  005201                    INC     R1              ;COUNT ACTIVE LINE
824 003144  110061  003152'           MOVB    R0,DEVTAB(R1)   ;STORE ACTIVE LINE NO. IN THE TABLE
825 003150  000763                    BR      1$              ;GO TEST NEXT LINE
826
827 003152                    DEVTAB: .BLKW   8.              ;16 BYTE ACTIVE DEVICE TABLE
828 003172  000000            ACTDEV: OPEN                    ;STORES COUNT OF NO. OF ACTIVE LINES MINUS ONE
829 003174  000000            DVICE:  OPEN                    ;DEVICE SELECTION INDICATOR
830
831                           ;JSR LINK TABLE CONSISTING OF 32 JSR'S (16 RCVR AND 16 XMTR) THAT
832                           ;LINK THE INTERRUPTS TO THE COMMON SERVICE ROUTINES
833
834 003176  004567  175774    JSRTAB: JSR     R5,RINT         ;RECEIVER LINK FOR LINE 0
835 003202  000000                    0                       ;SET UP WITH RCVR CSR ADDRESS
836 003204  000000                    0                       ;LINE NUMBER
837 003206  004567  175562            JSR     R5,TINT         ;XMTR LINK FOR LINE 0
838 003212  000000                    0                       ;SET UP WITH XMTR CSR ADDRESS
839 003214  000000                    0                       ;LINE NUMBER
840 003216  004567  175754            JSR     R5,RINT         ;LINK FOR LINE 1
841 003222  000000                    0
842 003224  000001                    1
843 003226  004567  175542            JSR     R5,TINT
844 003232  000000                    0
845 003234  000001                    1
846 003236  004567  175734            JSR     R5,RINT         ;LINK FOR LINE 2
847 003242  000000                    0
848 003244  000002                    2
849 003246  004567  175522            JSR     R5,TINT
850 003252  000000                    0
851 003254  000002                    2
852 003256  004567  175714            JSR     R5,RINT         ;LINK FOR LINE 3
853 003262  000000                    0
854 003264  000003                    3
855 003266  004567  175502            JSR     R5,TINT
856 003272  000000                    0
857 003274  000003                    3
858 003276  004567  175674            JSR     R5,RINT         ;LINK FOR LINE 4
859 003302  000000                    0
860 003304  000004                    4
861 003306  004567  175462            JSR     R5,TINT
862 003312  000000                    0
863 003314  000004                    4
864 003316  004567  175654            JSR     R5,RINT         ;LINK FOR LINE 5
865 003322  000000                    0
866 003324  000005                    5
867 003326  004567  175442            JSR     R5,TINT
868 003332  000000                    0
869 003334  000005                    5
870 003336  004567  175634            JSR     R5,RINT         ;LINK FOR LINE 6
871 003342  000000                    0
872 003344  000006                    6
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
873 003346  004567  175422          JSR     R5,TINT
874 003352  000000                  0
875 003354  000006                  6
876 003356  004567  175614          JSR     R5,RINT         ;LINK FOR LINE 7
877 003362  000000                  0
878 003364  000007                  7
879 003366  004567  175402          JSR     R5,TINT
880 003372  000000                  0
881 003374  000007                  7
882 003376  004567  175574          JSR     R5,RINT         ;LINK FOR LINE 10
883 003402  000000                  0
884 003404  000010                  10
885 003406  004567  175362          JSR     R5,TINT
886 003412  000000                  0
887 003414  000010                  10
888 003416  004567  175554          JSR     R5,RINT         ;LINK FOR LINE 11
889 003422  000000                  0
890 003424  000011                  11
891 003426  004567  175342          JSR     R5,TINT
892 003432  000000                  0
893 00343[illegible] 000011         11
894 003[illegible] 004567  175534   JSR     R5,RINT         ;LINK FOR LINE 12
895 003442  000000                  0
896 003444  000012                  12
897 003446  004567  175322          JSR     R5,TINT
898 003452  000000                  0
899 003454  000012                  12
900 003456  004567  175514          JSR     R5,RINT         ;LINK FOR LINE 13
901 003462  000000                  0
902 003464  000013                  13
903 003466  004567  175302          JSR     R5,TINT
904 003472  000000                  0
905 003474  000013                  13
906 003476  004567  175474          JSR     R5,RINT         ;LINK FOR LINE 14
907 003502  000000                  0
908 003504  000014                  14
909 003506  004567  175262          JSR     R5,TINT
910 003512  000000                  0
911 003514  000014                  14
912 003516  004567  175454          JSR     R5,RINT         ;LINK FOR LINE 15
913 003522  000000                  0
914 003524  000015                  15
915 003526  004567  175242          JSR     R5,TINT
916 003532  000000                  0
917 003534  000015                  15
918 003536  004567  175434          JSR     R5,RINT         ;LINK FOR LINE 16
919 003542  000000                  0
920 003544  000016                  16
921 003546  004567  175222          JSR     R5,TINT
922 003552  000000                  0
923 003554  000016                  16
924 003556  004567  175414          JSR     R5,RINT         ;LINK FOR LINE 17
925 003562  000000                  0
926 003564  000017                  17
927 003566  004567  175202          JSR     R5,TINT
928 003572  000000                  0
929 003574  000017                  17
```

```
930
931          000001                  .END
```

SYMBOL TABLE

```
ACSR    000102R   CKLINS  001602R   INDPAR= 000040    PRTY0 = 000000    SIZE    003030R
ACTDEV  003172R   CLKPRE= 000001    INIT    000030R   PRTY1 = 000040    SOFCNT  000042R
ADDR    000006R   CLKSP$= 104422    INTR    000120R   PRTY2 = 000100    SOFER$= 104406
ADDR22= 001000    CLRBUF  003050R   JSRTAB  003176R   PRTY3 = 000140    SOFPAS  000046R
APTPRE= 000200    CLRTAB  003066R   KTPRES= 000400    PRTY4 = 000200    SPOINT  000032R
ASB     000106R   CNTR    003040R   KTXTND= 040000    PRTY5 = 000240    SPSIZ = 000040
ASTAT   000104R   CNTTBL  001650R   MAP22$= 104416    PRTY6 = 000300    SR1     000016R
AUTO  = 000010    CONFIG  000056R   MODNAM  000000R   PRTY7 = 000340    SR2     000020R
AWAS    000110R   COUNT   003032R   MODREV  000224R   PS    = 177776    SR3     000022R
BADR    002002R   CSRA    000100R   MODSP   000224R   PSW   = 177776    SR4     000024R
BADT    002044R   DATBAD  002126R   MSGN$ = 104403    PUSH  = 005746    START   000226R
BEGIN   000000R   DATCK$= 104411    MSGS$ = 104402    PUSH2 = 024646    STAT    000026R
BIT0  = 000001    DATER$= 104404    MSG$  = 104401    PWRFLG= 000002    STUP1   000566R
BIT1  = 000002    DCNT    002710R   M.FRAM  001736R   QMON22= 000010    SVPTR   003042R
BIT10 = 002000    DEVTAB  003152R   M.OVER  001712R   QPTR1   003044R   SVR0    000062R
BIT11 = 004000    DTAB    003104R   M.PAR   001762R   QPTR2   003046R   SVR1    000064R
BIT12 = 010000    DVICE   003174R   NCPUOP= 000020    RAND$ = 104417    SVR2    000066R
BIT13 = 020000    DVID1   000014R   NOAPTY= 000002    RANNUM  000054R   SVR3    000070R
BIT14 = 040000    ECCMEM= 000100    NULL  = 000000    RBUF    003010R   SVR4    000072R
BIT15 = 100000    ENDIT$= 104413    OPEN  = 000000    RCNT    002650R   SVR5    000074R
BIT2  = 000004    END$  = 104410    OTOA$ = 104420    RCVERR  002200R   SVR6    000076R
BIT3  = 000010    ENPS    002220R   OVERUN  001706R   REST    000562R   SYSCNT  000052R
BIT4  = 000020    ERRCHK  001302R   PARITY  001756R   RESTRT  000552R   TCNT    002670R
BIT5  = 000040    ERRTYP  000106R   PARPRE= 002000    RESYNC  002224R   TINT    000774R
BIT6  = 000100    EXIT$ = 104400    PASCNT  000034R   RES1    000056R   TQ      002730R
BIT7  = 000200    FRAME   001732R   PDPF11= 000002    RES2    000060R   TRPDFD= 000023
BIT8  = 000400    GETADR  002110R   PDPLSI= 020000    RH70  = 001000    TSERV   001034R
BIT9  = 001000    GETPA$= 104415    PDP44 = 100000    RINT    001176R   TXCNT   003034R
BREAK$= 104407    GWBUF$= 104414    PDP60 = 004000    RSTAB   002250R   USTACK= 000001
BR1     000012R   HRDCNT  000044R   PDP70 = 010000    RSTRT   000112R   VECTOR  000010R
BR2     000013R   HRDER$= 104405    PIRQ$ = 000004    RXCNT   003036R   WASADR  000104R
BTOD$ = 104421    HRDPAS  000050R   POPSP = 005726    R6    =%000006    WDFR    000116R
CAPRES= 000004    ICONT   000036R   POPSP2= 022626    R7    =%000007    WDTO    000114R
CDATA$= 104412    ICOUNT  000040R   PRHMS$= 000002    SBADR   000102R   XBUF    002770R
CHK1    001334R   IDNUM   000122R   PRTY  = 000000    SETCSR  000462R   XFLAG   000005R
CKHNG$= 000001

. ABS.  000000     000
        003576     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13133 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  19748 WORDS  ( 75 PAGES)
ELAPSED TIME:  00:00:56
DLA,DLA/CR/-SP=DDXCOM,DLA
```

SYMBOL CROSS REFERENCE CREF V02

SYMBOL	VALUE		REFERENCES							
ACSR	000102	R	#8-385	*9-552	*9-699	*9 713	*9-751			
ACTDEV	003172	R	9-488	9-588	9-645	*9-819	#9-828			
ADDR	000006	R	#8-385	9-454	9-728					
ADDR22	• 001000		#8-385							
APTPRE	• 000200		#8-385							
ASB	000106	R	#8-385	*9-741						
ASTAT	000104	R	#8-385	*9-700	*9-714					
AUTO	• 000010		#8-385							
AWAS	000110	R	#8-385	*9-735						
BADR	002002	R	9-650	#9-697						
BADT	002044	R	9-654	#9-710						
BEGIN	000000	R	#8-385	9-457	9-504	9 504	9-528	9-550	9-555	9-556	9-614
			9-615	9-619	9-620	9-624	9-625	9-630	9-703	9-717	9-743
			9-755								
BIT0	• 000001		#8-385							
BIT1	• 000002		#8-385							
BIT10	• 002000		#8-385							
BIT11	• 004000		#8-385							
BIT12	• 010000		#8-385							
BIT13	• 020000		#8-385							
BIT14	• 040000		#8-385							
BIT15	• 100000		#8-385							
BIT2	• 000004		#8-385							
BIT3	• 000010		#8-385							
BIT4	• 000020		#8-385							
BIT5	• 000040		#8-385							
BIT6	• 000100		#8-385							
BIT7	• 000200		#8-385							
BIT8	• 000400		#8-385							
BIT9	• 001000		#8-385							
BREAK$	• 104407		#8-385	9-504	9-504					
BR1	000012	R	#8-385	9 437	9-441					
BR2	000013	R	#8-385							
BUOD$	• 104421		#8-385							
CAPRES	• 000004		#8 385							
CDATA$	• 104412		#8-385							
CHK1	001334	R	#9 599							
CKHNG$	• 000001		#8 385							
CKLINS	001602	R	#9-645							
CLKPRE	• 000001		#8-385							
CLKSP$	• 104422		#8 385							
CLRBUF	003050	R	9-480	#9 793						
CLRTAB	003066	R	9-482	#9-801						
CNTR	003040	R	*9-501	*9-511	#9-785					
CNTTBL	001650	R	9-428	#9-661						
CONFIG	000056	R	#8-385							
COUNT	003032	R	#9-781							
CSRA	000100	R	#8-385	*9-551	*9-698	*9-712	*9 734	*9 750		
DATBAD	002126	R	9-638	#9-733						
DATCK$	• 104411		#8-385							
DATER$	• 104404		#8-385	9-743						
DCNT	002710	R	*9-497	*9-549	9-647	9-651	9-699	9 713	#9-771	

SYMBOL CROSS REFERENCE CREF V02

SYMBOL	VALUE		REFERENCES							
DEVTAB	003152	R	9-489	9-589	9-646	*9-824	@9 827			
DTAB	003104	R	9-481	@9-811						
DVICE	003174	R	*9-432	9-815	@9-829					
DVID1	000014	R	@8-385	9-431	9-455					
ECCMEM	= 000100		@8-385							
ENDIT$	= 104413		@8-385	9-755						
END$	= 104410		@8-385	9-457						
ENPS	002220	R	9-657	@9-755						
ERRCHK	001302	R	9-515	@9-588						
ERRTYP	000106	R	@8-385	*9-554	*9-609	*9-613	*9-618	*9-623	*9-629	*9-702	*9 716
EXIT$	= 104400		@8-385	9-550	9-556					
FRAME	001732	R	9-620	@9-683						
GETADR	002110	R	9-490	9-590	9-697	9-710	@9-724	9-733	9-749	
GETPA$	= 104415		@8-385							
GWBUF$	= 104414		@8-385							
HRDCNT	000044	R	@8-385							
HRDER$	= 104405		@8-385	9-555	9-630	9-703	9-717			
HRDPAS	000050	R	@8-385							
ICONT	000036	R	@8-385	*9-426	*9-428					
ICOUNT	000040	R	@8-385							
IDNUM	000122	R	@8-385							
INDPAR	= 000040		@8-385							
INIT	000030	R	@8-385							
INTR	000120	R	@8-385	*9-403						
JSRTAB	003176	R	9-433	9-444	9-458	9-467	@9-834			
KTPRES	= 000400		@8-385							
KTXTND	= 040000		@8-385							
MAP22$	= 104416		@8-385							
MODNAM	000000	R	@8-385							
MODREV	000224	R	@8-394							
MODSP	000224	R	8-385	@8-385						
MSGN$	= 104403		@8-385	9-615	9-620	9-625				
MSGS$	= 104402		@8-385							
MSG$	= 104401		@8-385							
M.FRAM	001736	R	9-683	@9-686						
M.OVER	001712	R	9-677	@9-680						
M.PAR	001762	R	9-689	@9-692						
NCPUOP	= 000020		@8-385							
NOAPTY	= 000002		@8-385							
NULL	= 000000		@8-385	9-555	9-614	9-619	9-624	9-630	9-703	9 717	
OPEN	= 000000		8-385	8-385	8-385	8-385	8-385	8-385	8-385	8-385	8 385
			8-385	8-385	8-385	8-385	8-385	8-385	8-385	8-385	8 385
			8-385	8-385	@8-385	9-780	9-781	9-782	9-784	9 785	9 786
			9-787	9-788	9-828	9-829					
OTOA$	= 104420		@8-385							
OVERUN	001706	R	9-615	@9-677						
PARITY	001756	R	9-625	@9-689						
PARPRE	= 002000		@8-385							
PASCNT	000034	R	@8-385	9-478						
PDPF11	= 000002		@8-385							
PDPLSI	= 020000		@8-385							
PDP44	= 100000		@8-385							

SYMBOL CROSS REFERENCE CREF V02

SYMBOL	VALUE		REFERENCES								
PDP60	= 004000		#8-385								
PDP70	= 010000		#8 385								
PIRQ$	= 000004		#8-385	9 528							
POPSP	= 005726		#8-385								
POPSP2	= 022626		#8-385								
PRHMS$	= 000002		#8-385								
PRTY	= 000000		8-385	#8-385							
PRTY0	= 000000		#8-385								
PRTY1	= 000040		#8-385								
PRTY2	= 000100		#8-385								
PRTY3	= 000140		#8-385								
PRTY4	= 000200		8-385	#8-385							
PRTY5	= 000240		#8-385								
PRTY6	= 000300		#8-385								
PRTY7	= 000340		#8-385								
PS	= 177776		#8-385								
PSW	= 177776		#8-385								
PUSH	= 005746		#8-385								
PUSH2	= 024646		#8-385								
PWRFLG	= 000002		#8-385								
QMON22	= 000010		#8-385								
QPTR1	003044	R	*9-486	9-522	*9-523	9-524	*9-526	#9 787			
QPTR2	003046	R	*9-487	9-533	*9-534	9-535	*9-537	#9-788			
RAND$	= 104417		#8-385								
RANNUM	000054	R	#8-385								
RBUF	003010	R	*9-631	*9-634	9-636	9-739	9-759	9-762	#9-776	9 795	
RCNT	002650	R	*9-577	9-647	9-700	#9-769					
RCVERR	002200	R	9-608	9-628	#9-748						
REST	000562	R	9-469	#9-480							
RESTRT	000552	R	8-385	#9-477							
RESYNC	002224	R	#9-759								
RES1	000056	R	#8-385								
RES2	000060	R	#8-385								
RH70	= 001000		#8-385								
RINT	001176	R	#9-562	9-834	9-840	9-846	9-852	9-858	9-864	9-870	9 876
			9-882	9-888	9-894	9-900	9-906	9-912	9-918	9-924	
RSTAB	002250	R	9-485	9-602	#9-768	9-801					
RSTRT	000112	R	#8-385								
RXCNT	003036	R	*9-484	9-507	9-564	*9-578	9-599	*9-601	*9-640	#9 784	
R6	=#000006		#8-385	*9-527	*9-562	*9-563	*9-579	*9-580	*9-581		
R7	=#000007		#8-385								
SBADR	000102	R	#8-385	*9-742							
SETCSR	000462	R	9-446	#9-454							
SIZE	003030	R	*9-412	9-634	9-635	#9 780					
SOFCNT	000042	R	#8-385								
SOFER$	= 104406		#8-385	9-614	9-619	9-624					
SOFPAS	000046	R	#8-385								
SPOINT	000032	R	#8-385								
SPSIZ	= 000040		#1-29	8-385							
SR1	000016	R	#8-385	9-404	9-414						
SR2	000020	R	#8-385	9-422	9-424						
SR3	000022	R	#8-385								

SYMBOL	VALUE		REFERENCES								
SR4	000024	R	#8-385								
START	000226	R	8-385	#9 400	9 479						
STAT	000026	R	#8-385								
STUP1	000566	R	#9 481	9 764							
SVPTR	003042	R	*9-485	9 566	*9 575	#9 786					
SVR0	000062	R	#8-385								
SVR1	000064	R	#8-385								
SVR2	000066	R	#8-385								
SVR3	000070	R	#8-385								
SVR4	000072	R	#8-385								
SVR5	000074	R	#8-385								
SVR6	000076	R	#8-385								
SYSCNT	000052	R	#8-385								
TCNT	002670	R	*9-541	9-651	9-714	#9-770					
TINT	000774	R	#9-522	9 837	9-843	9-849	9-855	9-861	9-867	9-873	9 879
			9-885	9 891	9-897	9-903	9-909	9-915	9-921	9-927	
TQ	002730	R	9-486	9-487	9-524	9-526	9-535	9-537	#9-773	9-803	
TRPDFD	* 000023		#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8-385	8-385
			#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8-385	8-385
			#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8-385	8-385
			#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8-385	8 385
			#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8-385	8-385
			#8-385	8-385	8-385	#8-385	8-385	8-385	#8-385	8 385	8-385
			#8-385	8-385	8-385	#8-385					
TSERV	001034	R	9-528	#9-533							
TXCNT	003034	R	*9-483	*9-496	9 505	*9-540	9-544	*9-548	#9 782		
USTACK	* 000001		#8-385								
VECTOR	000010	R	#8-385	9-430							
WASADR	000104	R	#8-385	*9-737							
WDFR	000116	R	#8-385	*9-402							
WDTO	000114	R	#8-385	*9-401							
XBUF	002770	R	*9-494	9-495	*9-546	9-547	9-760	#9 775	9 793		
XFLAG	000005	R	#8-385								

MACRO NAME	REFERENCES								
BKMOD	#1-125								
BREAK	#1-223	9 504							
BTOD	#1 247								
CKDATA	#1-283								
CLKSP	#1-150								
DATACK	#1 292								
DATERR	#1-176	9-743							
DFSEVN	#1-315	8-385	8-385	8 385	8-385	8 385	8 385	8-385	8 385	8 385
	8-385	8-385	8-385						
DSEVNT	#1-325	8-385	8-385	8-385	8-385	8-385	8-385	8 385	
END	#1-213	9-457							
ENDIT	#1-204	9-755							
ENDMOD	#1-209								
EQUATS	#1-331	8-385							
EXIT	#1-158	9-550	9-556						
GETPA	#1-274								
GMBUFF	#1-262								
HLFBRK	#1-228								
HRDER	#1-166	9-555	9-630	9-703	9-717				
IOMOD	#1-121	8-385							
IOMODP	#1-145								
IOMODR	#1-141								
IOMODX	#1 137								
MAP22	#1-278								
MODULE	#1-30	8-385							
MSG	#1-192								
MSGN	#1-196	9-615	9-620	9-625					
MSGS	#1-200								
NBKMOD	#1-133								
OTOA	#1-233								
PIRQ	#1-217	9-528							
RAND	#1-162								
SBKMOD	#1-129								
SOFER	#1-182	9-614	9-619	9-624					