```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E932D-MC

PRODUCT NAME: CXRCAD0 RC11 MOD

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RCA IS AN IOMOD THAT EXERCISES RS64 DISK DRIVES ON AN
RC11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES,
WRITE-CHECKS, READS, AND IN-CORE COMPARISONS.
ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 4 RS64 DISK DRIVES WITH AN RC11 CONTROLLER

STORAGE:: RCA REQUIRES:
1. DECIMAL WORDS: 830
2. OCTAL WORDS: 1476
3. OCTAL BYTES: 3174

3. PASS DEFINITION

ONE PASS OF THE RCA MODULE CONSISTS OF 600 CYCLES OF THE BASIC TEST
SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE
WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS,
AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF RCA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177440, VECTOR: 210, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

7. MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A STARTING SECTOR ADDRESS
D. GET A DRIVE ADDRESS
E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
I. IF END OF PASS, REPORT AND GO TO C
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS

NONE

9. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT

B. ERROR MESSAGES DUMP THE CONTENTS OF THE 8 RC11 REGISTERS IN THE FOLLOWING ORDER:

RCLA RCDA RCER RCCS RCWC RCBA RCMR RCDB

```
000000'  -
000000'          IOMODX  <RCAD  >,177440,210,5,0,0,600.,21,BUFIN,256.,1024.
000000'          MODULE  150000,RCAD ,177440,210,5,0,0,600.,21,BUFIN,256.,1024.
                 .TITLE  RCAD DEC/X11 SYSTEM EXERCISER MODULE
                 ;       DDXCOM  VERSION  6       23-MAY-78
                                 .LIST   BIN
                                 ;*****************************************************************
000000'                          BEGIN:
000000'  041122  042101     040 MODNAM: .ASCII  /RCAD / ;MODULE NAME.
000005'  000100                 XFLAG:  .BYTE   OPEN                ;USED TO KEEP TRACK OF WBUFF USAGE
000006'  177440                 ADDR:   177440+0                    ;1ST DEVICE ADDR.
000010'  000210                 VECTOR: 210+0                       ;1ST DEVICE VECTOR.
000012'      240                 BR1:    .BYTE   PRTY5+0             ;1ST BR LEVEL.
000013'      300                 BR2:    .BYTE   PRTY0+0             ;2ND BR LEVEL.
000014'  000001                 DVID1:  0+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'  150000                 STAT:   150000                      ;STATUS WORD.
000030'  000252'                 INIT:   START                       ;MODULE START ADDR.
000032'  000252'                 SPOINT: MODSP                       ;MODULE STACK POINTER.
000034'  000000                 PASCNT: 0                           ;PASS COUNTER.
000036'  001130                 ICONT:  600.                        ;# OF ITERATIONS PER PASS=600.
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'  000412'                 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'  002000                 INTR:   OPEN                        ;# OF INTERRUPTS PER ITERATION
000122'  000021                 IDNUM:  21                          ;MODULE IDENTIFICATION NUMBER=21
```



```
000124'  002002'                 RBUFVA: BUFIN                       ;READ BUFFER VIRTUAL ADDRESS
000126'  000000                 RBUFPA: OPEN                        ;READ BUFFER PHYSICAL ADDRESS
000130'  000000                 RBUFEA: OPEN                        ;READ BUFFER EA BITS
000132'  000400                 RBUFSZ: 256.                        ;SIZE OF THE READ BUFFER
000134'  000000                 WBUFPA: OPEN                        ;WRITE BUFFER PHYSICAL ADDRESS
000136'  000000                 WBUFEA: OPEN                        ;WRITE BUFFER EA BITS
000140'  002000                 WBUFRQ: 1024.                       ;WRITE BUFFER SIZE REQUESTED
000142'  000000                 WBUFSZ: OPEN                        ;WRITE BUFFER SIZE AVAILABLE
000144'  000000                 CDERCT: OPEN                        ;CDATA/DATCK ERROR COUNT
000146'  000000                 CDWDCT: OPEN                        ;CDATA/DATCK WORD COUNT
000150'  000000                 FREE:   OPEN                        ;RESERVED FOR FUTURE USE
         000040                          .REPT   SPSIZ              ;MODULE STACK STARTS HERE.
                                         .NLIST
                                         .WORD   0
                                         .LIST
                                         .ENDR
000252'                          MODSP:
                                 ;*****************************************************************
```

```
                                ;
000252' 012767  000003  177640  START:  MOV     #3,INTR         ;3 INTERRUPTS/ITERATION
000260' 012767  000400  177626          MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
000266' 012767  002667  177622          MOV     #1024.,WDFR     ;1024 WORDS FORM MEM/ITERATION
000274' 105067  002667                  CLRB    FLAG            ; CLEAR FLAGS
000300' 016767  177510  001452          MOV     DVID1,DVICE     ; GET DRIVE INDICATOR
000306' 123727  000041  000012          CMPB    @#41,#12        ;IF RS IS LOAD MEDIUM THEN
000314' 001124                          BNE     3$              ;BEGIN
000316' 113700  000040                  MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
000322' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
000326' 105700          1$:             TSTB    R0              ;   WHILE NOT POINTING AT LOAD-DEVICE DO
000330' 001403                          BEQ     2$              ;   BEGIN
000332' 006301                          ASL     R1              ;      POINT TO NEXT DEVICE
000334' 105300                          DECB    R0              ;      COUNT SHIFTS
000336' 000773                          BR      1$              ;   END
000340' 130167  001414          2$:     BITB    R1,DVICE        ;   IF LOAD-DEVICE SELECTED THEN
000344' 001410                          BEQ     3$              ;   BEGIN
000346' 113767  000040  001410          MOVB    @#40,DRVVE      ;      MOVE LOAD-DEVICE NUMBER TO  RVVE
000354' 004767  000536                  JSR     PC,DROP         ;      DROP THE DRIVE
000360' 104403  000000' 003150'         MSGMS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        ;   END
000366'                         3$:     ;END
000366' 012767  177740  001374          MOV     #-32.,BLK1      ; INITIALIZE BLOCK COUNTER
000374' 004767  001104                  JSR     PC,SETUP        ; GENERATE REGISTER ADDRESSES
000400' 004767  001200                  JSR     PC,REZET        ; INITIALIZE RC REGS. AND ALL DRIV S
000404' 005767  001350                  TST     DVICE           ; DROP THE MODULE ?
000410' 001502                          BEQ     FINI            ; YES
000412'                         RESTRT:
000412' 104415  000000' 000124'         GETPA$,BEGIN,,RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000420' 016767  177506  001352          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000426' 005467  001346                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
                                ;
000432' 004767  000620          STRT:   JSR     PC,BLOCK        ; GET NEXT DISK ADDRESS
000436' 016767  001326  001312          MOV     BLK1,DSKADR     ; SAVE DISK ADDRESS
000444' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
000450' 016767  177466  001320          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000456' 005467  001314                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
                                ;
000462' 004767  000622          NEXT:   JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
000466' 005767  001266                  TST     DVICE           ; ANY DRIVES LEFT ?
000472' 001451                          BEQ     FINI            ; NO, GO DROP THE MODULE
000474' 132767  000010  002466          BITB    #BIT3,F AG      ; ALL DRIVES DONE ?
000502' 001353                          BNE     STRT            ; YES, GO GET ANOTHER BLOCK
                                ;
000504' 042767  014000  001244          BIC     #14000,DSKADR   ; CLEAR DRIVE ADDRESS
000512' 056767  001256  001236          BIS     DRVSFT,DSKADR   ; SAVE DRIVE ADDRESS
000520' 005067  002444                  CLR     TRY1            ; ZERO RETRY COUNTERS
000524' 105067  002442                  CLRB    TRY3            ;
                                ;
000530' 004567  000172          GO:     JSR     R5,WRITE        ; WRITE SOME DATA
000534' 000432                          BR      RETRY1          ; IF ERRORS, TRY IT AGAIN
000536' 132764  000004  002423          BITB    #BIT2,FLAG      ; DID DISK OVERFLOW ?
000544' 001407                          BEQ     GOA             ; NO, CONTINUE
000546' 142767  000004  002413          BICB    #BIT2,FLAG      ; YES, CLEAR OVERFLOW FLAG
000554' 012767  177740  001206          MOV     #-32.,BLK1      ; RESET BLOCK NUMBER
000562' 000723                          BR      STRT            ; START OVER AT BEGINNING OF DISK
000564' 004567  000170          GOA:    JSR     R5,WRITCK       ; WRITE-CHECK THE DATA
```



```
000570' 000426                          BR      RETRY2          ; IF ERRORS, TRY AGAIN
000572' 004567  000214          GOB:    JSR     R5,READ         ; READ THE DATA WRITTEN
000576' 000435                          BR      RETRY3          ; IF ERRORS, TRY AGAIN
000600' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
000606' 000610'                         .+2                     ; IF ERROR, CONTINUE
                                ;
000610' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                ;                               ;MONITOR SHALL TEST END OF PASS
000614' 000722                          BR      NEXT            ;  CONTINUE
                                ;
000616'                         FINI:
000616' 104410  000000'                 END$,BEGIN              ; DROP THE MODULE
                                ;       -------------------------
```

```
000622' 105267 002342          RETRY1: INCB    TRY1            ; COUNT THE RETRYS
000626' 122767 000003 002334          CMPB    #3,TRY1         ; LIMIT EXCEEDED ?
000634' 001335                        BNE     GO1             ; NO, GO TRY IT AGAIN
000636' 104403 000000' 003126'        MSGN$,BEGIN,EXCED1      ;ASCII MESSAGE CALL WITH COMMON HEADER
000644' 000424                        BR      NEXTA           ; GO ON TO NEXT DRIVE
                                      ;-----------------------
000646' 105267 002317          RETRY2: INCB    TRY2            ; COUNT RETRYS
000652' 122767 000003 002311          CMPB    #3,TRY2         ; LIMIT EXCEEDED ?
000660' 001341                        BNE     GO1             ; NO, TRY AGAIN
000662' 104403 000000' 003134'        MSGN$,BEGIN,EXCED2      ;ASCII MESSAGE CALL WITH COMMON HEADER
000670' 000412                        BR      NEXTA           ; GO ON TO NEXT DRIVE
                                      ;-----------------------
000672' 105267 002274          RETRY3: INCB    TRY3            ; COUNT RETRYS
000676' 122767 000003 002266          CMPB    #3,TRY3         ; LIMIT EXCEEDED ?
000704' 001332                        BNE     GO1             ; NO, GO TRY AGAIN
000706' 104403 000000' 003142'        MSGN$,BEGIN,EXCED3      ;ASCII MESSAGE CALL WITH COMMON HEADER
000714' 000400                        BR      NEXTA           ; GO ON TO NEXT DRIVE
                                      ;-----------------------
000716' 004767 000662          NEXTA:  JSR     PC,REZET        ; GO CHECK ALL DRIVES FOR ON-LINE
000722' 000167 177534                  JMP     NEXT            ; GO ON TO NEXT DRIVE
                                      ;-----------------------


                                      ;               -----  RC11 DISK DRIVERS  -----

000726' 012767 000103 001016  WRITE:  MOV     #103,FUNC       ; LOAD WRITE FUNCTION
000734' 016777 001036 002052          MOV     WCNT1,@RCWC     ; LOAD WORD COUNT
000742' 016777 177166 002046          MOV     WBUFPA,@RCBA    ; LOAD BUFFER ADDRESS
000750' 016767 177162 000776          MOV     WBUFEA,XMEM     ; GET EXTENDED MEMORY BITS
000756' 000431                        BR      GOGO            ; CONTINUE
000760' 012767 000107 000764  WRITCK: MOV     #107,FUNC       ; LOAD WRITE-CHECK FUNCTION
000766' 016777 001004 002020          MOV     WCNT1,@RCWC     ; LOAD WORD COUNT
000774' 016777 177134 002014          MOV     WBUFPA,@RCBA    ; LOAD BUFFER ADDRESS
001002' 016767 177130 000744          MOV     WBUFEA,XMEM     ; GET EXTENDED MEMORY BITS
001010' 000414                        BR      GOGO            ; CONTINUE
001012' 012767 000105 000732  READ:   MOV     #105,FUNC       ; LOAD READ FUNCTION
001020' 016777 000754 001766          MOV     WCNT2,@RCWC     ; LOAD WORDCOUNT
001026' 016777 177074 001762          MOV     RBUFPA,@RCBA    ; LOAD BUFFER ADDRESS
001034' 016767 177070 000712          MOV     RBUFEA,XMEM     ; GET EXTENDED MEMORY BITS

001042' 012777 001076' 176740 GOGO:   MOV     #NTRUPT,@VECTOR ; SET INTERUPT ENTRY POINTER
001050' 016777 000702 001730          MOV     DSKADR,@RCDA    ; LOAD THE DISK ADDRESS
001056' 056767 000672 000666          BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
001064' 016777 000662 001720          MOV     FUNC,@RCCS      ; EXECUTE THE FUNCTION
001072' 104400 000000'                EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```

001076'                        NTRUPT:
                               ;-------------------------------------------------------------
001076' 0000C  000000' 001104'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND WAIT
                               ;-------------------------------------------------------------

001104' 004567 000334  1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
001110' 000205                 RTS     R5              ; ERRORS DETECTED, RETURN
001112' 005725                 TST     (R5)+           ; NO ERRORS, SKIP RETRY
001114' 000205                 RTS     R5              ; RETURN OK
                               ;-----------------------

001116' 012701 000001  DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
001122' 016700 000636          MOV     DRYVE,R0        ; GET THE DRIVE NUMBER
001126' 001403                 BEQ     2$              ; IF DRIVE 0 GO DROP IT
001130' 006301         1$:     ASL     R1              ; NO, AIM AT THE NEXT DRIVE
001132' 005300                 DEC     R0              ; IS THIS THE ONE ?
001134' 001375                 BNE     1$              ; NO, LOOK AGAIN
001136' 040167 000616  2$:     BIC     R1,DVICE        ; DROP THE DRIVE
                               ;*****************************************************
                                                       ;CONVERT DRYVE TO ASCII AND
                                                       ;STORE AT ADR1
001142' 104420 000000' 001764'         OTOA$,BEGIN,DRYVE,ADR1
001150' 003160'
                               ;*************************************************************
001152' 000207                 RTS     PC              ; RETURN
                               ;-----------------------


001154' 016700 000610  ROOM:   MOV     BLK1,R0         ; SAVE CURRENT BLOCK NUMBER
001160' 005700                 TST     R0              ; ON BLOCK ZERO ?
001162' 001003                 BNE     1$              ; NO, CONTINUE
001164' 012702 177777          MOV     #-1,R2          ; YES, SET REG. 2 FOR MAX. TRANSFER
001170' 000416                 BR      3$              ; CONTINUE
001172' 012701 003777  1$:     MOV     #2047.,R1       ; LOAD MAX. NUMBER OF BLOCKS
001176' 005002                 CLR     R2              ; ZERO REG. 2
001200' 160001                 SUB     R0,R1           ; GET # OF BLOCKS LEFT ON DISK
001202' 062702 000040  2$:     ADD     #32.,R2         ; GET TOTAL # OF WORDS LEFT
001206' 005301                 DEC     R1              ; ALL BLOCKS ADDED IN ?
001210' 001374                 BNE     2$              ; NO, KEEP ADDING
001212' 005702                 TST     R2              ; IS # OF WORDS LEFT ON DISK NEG. ?
001214' 100404                 BMI     3$              ; YES
001216' 005767 176720          TST     WBUFSZ          ; IS TRANSFER SIZE NEG. ?
001222' 100411                 BMI     6$              ; YES
001224' 000403                 BR      4$              ; NO, GO COMPARE
001226' 005767 176710  3$:     TST     WBUFSZ          ; IS TRANSFER SIZE POSITIVE ?
001232' 100003                 BPL     5$              ; YES
001234' 020267 176702  4$:     CMP     R2,WBUFSZ       ; WAS THERE ENOUGH ROOM FOR THE TRANSFER
001240' 002402                 BLT     6$              ; NO, RETURN OK
001242' 005725         5$:     TST     (R5)+           ; YES, MUST BE A REAL ERROR
001244' 000205                 RTS     R5              ; RETURN ERROR
001246' 152767 000004 001713   6$:     BISB    #BIT2,FLAG      ; SET OVERFLOW FLAG
001254' 000205                 RTS     R5              ; RETURN OK
                               ;-----------------------
```

```
                                                ;
001256' 062767  000040  000504  BLOCK:  ADD     #32.,BLK1       ; STEP TO NEXT BLOCK
001264' 022767  004000  000476          CMP     #2048.,BLK1     ; BLOCK LIMIT REACHED ?
001272' 100002                          BPL     1$              ; NO, CONTINUE
001274' 005067  000470                  CLR     BLK1            ; YES, RESET BLOCK #
001300' 016767  000464  000464  1$:     MOV     BLK1,BLK2       ; READ WHERE WRITE
001306' 000207                          RTS     PC              ; RETURN
                                        ;       ------------------------
001310' 005267  000450  DRVADR: INC     DRVVE                   ; COUNT A DRIVE
001314' 062767  004000  000444          ADD     #BIT11,DRVSFT   ; DRIVE # LINED UP WITH RFDA
001322' 142767  000010  001637          BICB    #BIT3,FLAG      ; CLEAR END OF DRIVES FLAG
001330' 022767  000004  000426          CMP     #4,DRVVE        ; ALL DRIVES CHECKED ?
001336' 001404                          BEQ     1$              ; YES, GO FLAG END OF DRIVES
001340' 006267  000416                  ASR     DRIVE           ; NO, IS NEXT DRIVE CHOSEN ?
001344' 103361                          BCC     DRVADR          ; NO, GO TRY ANOTHER DRIVE
001346' 000207                          RTS     PC              ; YES, RETURN
                                        ;
001350' 152767  000010  001611  1$:     BISB    #BIT3,FLAG      ; SET END OF DRIVES FLAG
001356' 012767  177777  000400          MOV     #-1,DRVVE       ; RESET DRIVE COUNTER
001364' 012767  174000  000374          MOV     #174000,DRVSFT  ; RESET SHIFTED DRIVE #
001372' 016767  000[illegible]  000362          MOV     DVICE,DRIVE     ; RESTORE CHOSEN DRIVES
001400' 000207                  2$:     RTS     PC              ; RETURN
                                        ;       ------------------------
001402' 014167  176500          ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
001406' 010167  176470                  MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
001412' 014267  176472                  MOV     -(R2),AWAS      ; LOAD THE DATA
001416' 010267  176462                  MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
001422' 005721                          TST     (R1)+           ; RESET REG. 1
001424' 005722                          TST     (R2)+           ; RESET REG. 2
                                        ;
001426' 016767  001360  176444  ERSUB1: MOV     RCCS,CSRA       ; LOAD ADR. OF CURRENT CSR
001434' 017767  001352  176440          MOV     @RCCS,ACSR      ; LOAD CONTENTS OF CURRENT CSR
001442' 000207                          RTS     PC              ; RETURN
                                        ;       ------------------------
```



```
                                                ;
001444' 005777  001342  ERRORS: TST     @RCCS           ; ANY ERRORS ?
001450' 100013                  BPL     1$              ; NO, RETURN
001452' 004567  177476          JSR     R5,ROOM         ; YES, IS IT A REAL ERROR ?
001456' 000410                  BR      1$              ; NO, CONTINUE
001460' 004767  177742          JSR     PC,ERSUB1       ; YES, LOAD ERROR INFORMATION
001464' 005067  176416          CLR     ERRTYP          ;UNKNOWN ERROR
                                        ;***************************************************************
001470' 104406  000000' 003004'         SOFERS,BEGIN,TABLE      ;
                                        ;***************************************************************
001476' 000205                  RTS     R5              ; RETURN, ERRORS
001500' 005725                  TST     (R5)+           ; SKIP RETRY
001502' 000205                  RTS     R5              ; RETURN OK
                                        ;       ------------------------
                                        ;
001504' 016700  176276  SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
001510' 010067  001278          MOV     R0,RCLA         ; GENERATE CONTROLLER REGS. ADDRESSES
001514' 005720                  TST     (R0)+
001516' 010067  001264          MOV     R0,RCDA
001522' 005720                  TST     (R0)+
001524' 010067  001262          MOV     R0,RCBA
001530' 005720                  TST     (R0)+
001532' 010067  001254          MOV     R0,RCCS
001536' 005720                  TST     (R0)+
001540' 010067  001250          MOV     R0,RCWC
001544' 005720                  TST     (R0)+
001546' 010067  001244          MOV     R0,RCBA
001552' 005720                  TST     (R0)+
001554' 010067  001240          MOV     R0,RCMA
001560' 005720                  TST     (R0)+
001562' 010067  001234          MOV     R0,RCDB
001566' 016700  176216          MOV     VECTOR,R0       ; GET THE VECTOR ADDRESS
001572' 012720  000437'         MOV     #$RTI,(R0)+     ; SET POINTER JUST IN CASE
001576' 116710  176210          MOVB    BR1,(R0)        ; SET PRIORITY
001602' 000207          1$:     RTS     PC              ; RETURN
                                        ;       ------------------------
                                        ;
001604' 012767  077777  001170  REZET:  MOV     #77777,CLK      ; SET THE TIMER
001612' 105777  001174          1$:     TSTB    @RCCS           ; CONTROLLER READY ?
001616' 100420                          BMI     2$              ; YES, CONTINUE
                                        ;
001620' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001624' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001630' 005367  001146                  DEC     CLK             ; WAIT SOME MORE ?
001634' 001366                          BNE     1$              ; YES
001636' 012767  000003  176242          MOV     #3,ERRTYP       ;CONTROLLER NOT READY
                                        ;***************************************************************
001644' 104405  000000' 003004'         HRDERS,BEGIN,TABLE      ; CONTROLLER NOT READY
                                        ;***************************************************************
001652' 005067  000102          2$:     CLR     DVICE           ; SET TO DROP THE MODULE
001656' 000207                          RTS     PC              ; RETURN
```

```
                                ;
                                ;
001660' 012767  177777  000076  2$:     MOV     #-1,DRVVE       ; INITIALIZE DRIVE COUNTERS
001666' 016767  000066  000066          MOV     DVICE,DRIVE     ; GET DRIVES NOW ACTIVE
001674' 012767  174000  000064          MOV     #174000,DRVSFT  ; INITIALIZE SHIFTED DRIVE #
001702' 004767  177402                  JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
001706' 132767  000010  001253  3$:     BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
001714' 001015                          BNE     4$              ; YES, RETURN
001716' 016777  000044  001062          MOV     DRVSFT,@RCDA    ; NO, LOAD DISK ADDRESS REG.
001724' 032777  004000  001060          BIT     #BIT11,@RCCS    ; DRIVE EXIST ?
001732' 001763                          BEQ     3$              ; YES, CONTINUE
001734' 004767  177156                  JSR     PC,DROP         ; NO, DROP THE DRIVE
001740' 104403  000000' 003150'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
001746' 000755                          BR      3$              ; MAKE SURE ALL GET CHECKED
001750' 000207  4$:     RTS     PC      ; RETURN
                ;       ------------------------
                ;
                ;
001752' 000000  FUNC:   0
001754' 000000  XMEM:   0
001756' 000000  DSKADR: 0
001760' 000000  DVICE:  0
001762' 000000  DRIVE:  0
001764' 000000  DRVVE:  0
001766' 000000  DRVSFT: 0
001770' 000000  BLK1:   0
001772' 000000  BLK2:   0
001774' 000000  TBUF:   0
001776' 000000  WCNT1:  0
002000' 000000  WCNT2:  0
002002' 000400  BUFIN:  .BLKW   256.
003002' 000000  CLK:    0
003004'         TABLE:
003004' 000200  RCLA:   0
003006' 000000  RCDA:   0
003010' 000000  RCER:   0
003012' 000000  RCCS:   0
003014' 000000  RCWC:   0
003016' 000000  RCBA:   0
003020' 000000  RCMR:   0
003022' 000000  RCDB:   0
003024' 177777          177777
```



```
003026' 020040  051104  053111  MES3:   .ASCIZ  "  DRIVE  "
003034' 020105  000040
003040' 020040  051104  050117  MES4:   .ASCIZ  "  DROPPED$"
003046' 042520  022504  000
003053'    040  042522  051124  MES5:   .ASCIZ  " RETRY EXCEEDED$"
003060' 020131  054105  042503
003066' 042105  042105  000045
003074' 053440  044522  042524  MES6:   .ASCIZ  " WRITE"
003102'    000
003103'    040  051127  052111  MES7:   .ASCIZ  " WRITE-CHECK"
003110' 026505  044103  041505
003116' 000113
003120' 051040  040505  000104  MES8:   .ASCIZ  " READ"
                                .EVEN
003126' 003074'                 EXCED1: MES6
003130' 003053'                         MES5
003132' 177777                          177777
003134' 003103'                 EXCED2: MES7
003136' 003053'                         MES5
003140' 177777                          177777
003142' 003120'                 EXCED3: MES8
003144' 003053'                         MES5
003146' 177777                          177777
003150' 003026'                 DRP:    MES3
003152' 003165'                         NUMB
003154' 003040'                         MES4
003156' 177777                          177777
003160' 000005                  ADR1:   .BLKB   5
003165'    000                  NUMB:   .BYTE   0
003166'    000                          .BYTE   0
003167'    000                  FLAG:   .BYTE   0
                                .EVEN
003170'    000                  TRY1:   .BYTE   0
003171'    000                  TRY2:   .BYTE   0
003172'    000                  TRY3:   .BYTE   0
                                .EVEN
003174'
000001                          .END
```

```
ACSR    000102R          182#    420*
ADDR    000006R          148#    440
ADDR22= 001000           211#
ADR1    003160R          352     549#
ASB     000106R          186#    412#
ASTAT   000104R          184#
AWAS    000110R          187#    414*
BEGIN   000000R          145#    229     238     244     270     273     278     283     290     297     327     332     352
                         432     467     468     473     489
BIT0  = 000001           211#
BIT1  = 000002           211#
BIT10 = 002000           211#
BIT11 = 004000           211#    396     486
BIT12 = 010000           211#
BIT13 = 020000           211#
BIT14 = 040000           211#
BIT15 = 100000           211#
BIT2  = 000004           211#    261     263     381
BIT3  = 000010           211#    251     397     404     483
BIT4  = 000020           211#
BIT5  = 000040           211#
BIT6  = 000100           211#
BIT7  = 000200           211#
BIT8  = 000400           211#
BIT9  = 001000           211#
BLK1    001770R          232*    243     264*    359     386*    387     389*    390     502#
BLK2    001772R          390*    503#
BLOCK   001256R          242     386#
BREAK$= 104407           211#    467     468
BR1     000012R          150#    458
BR2     000013R          151#
BTODS = 104421           211#
BUFIN   002002R          193     507#
CDATA$= 104412           211#    270
CDERCT  000144R          201#
CDWDCT  000146R          202#
CLK     003002R          463*    469*    508#
CONFIG  000056R          170#
CSRA    000100R          180#    419*
DATCK$= 104411           211#
DATER$= 104404           211#
DRIVE   001762R          400*    407*    480*    499#
DROP    001116R          228     342#    488
DRP     003150R          229     489     545#
DRVADR  001310R          248     395#    401     487
DRVSFT  001766R          255     396*    406*    481*    485     501#
DRVVE   001764R          247*    243     395*    398     405*    479*    500#
DSKADR  001756R          243*    254*    324     497#
DVICE   001760R          215*    225     235     249     348*    407     475*    480     498#
DVID1   000014R          152#    215
ENDI$= 104413            211#    273
END$  = 104410           211#    278
ERRORS  001444R          335     425#
ERRTYP  000106R          185#    430*    471*
ERSUB1  001426R          419#    429
ERSUB2  001402R          412#
```



```
EXCED1  003126R          283     536#
EXCED2  003134R          290     539#
EXCED3  003142R          297     542#
EXITS = 104400           211#    327
FINI    000616R          236     250     277#
FLAG    003167R          214*    251     261     263*    381*    397*    404*    483     552#
FREE    000150R          203#
FUNC    001752R          308*    313*    318*    325*    326     495#
GETPA$= 104415           211#    238
GC      000530R          259#    282
GOA     000564R          262     266#    289
GOB     000572R          268#    296
GOGO    001042R          312     317     323#
GWBUF$= 104414           211#    244
HDCNT   000044R          165#
HRDER$= 104405           211#    473
HRDPAS  000050R          167#
ICONT   000036R          162#
ICOUNT  000040R          163#
IDNUM   000122R          192#
IMODX.= 000000           204#    245
INIT    000030R          159#
INTR    000120R          191#    211*
MAP22$= 104416           211#
MES3    003026R          522#    545
MES4    003040R          524#    547
MES5    003053R          526#    537     540     543
MES6    003074R          529#    536
MES7    003103R          531#    539
MES8    003120R          534#    542
MODNAM  000000R          146#
MODSP   000252R          169     209#
MSGNS = 104403           211#    229     283     290     297     489
MSGSS = 104402           211#
MSGS  = 104401           211#
NEXT    000462R          248#    275     302
NEXTA   000716R          284     291     298     301#
NTRUPT  001076R          323     330#
NULL  = 000000           211#
NUMB    003165R          546     550#
OPEN  = 000000           147     153     154     155     156     173     174     175     176     177     178     179     180
                         182     184     186     187     189     190     191     194     195     197     198     200     201
                         202     203     211#
OTOAS = 104420           211#    352
PASCNT  000034R          161#
PIRQS = 000004           211#    332
POPSP = 005726           211#
POPSP2= 022626           211#
PRTY  = 000000           211#
PRTY0 = 000000           151     211#
PRTY1 = 000040           211#
PRTY2 = 000100           211#
PRTY3 = 000140           211#
PRTY4 = 000200           211#
PRTY5 = 000240           150     211#
PRTY6 = 000300           211#
```

```
PRTY7 = 000340          211#
PS    = 177776          211#
PSW   = 177776          211#
PUSH  = 005746          211#
PUSH2 = 024646          211#
RANDS = 104417          211#
RANNUM  000054R         169#
RBUFEA  000130R         195#    321
RBUFPA  000126R         194#    270     320
RBUFSZ  000132R         196#    239
RBUFVA  000124R         193#    238
RCBA    003016R         310*    315*    320*    451*    515#
RCCS    003012R         326*    419     420     426     447#    464     486     513#
RCDA    003006R         324*    423*    485*    511#
RCDB    003022R         455*    517#
RCER    003010R         445*    512#
RCLA    003004R         441*    510#
RCMR    003020R         451*    516#
RCWC    003014R         309*    314*    319*    449*    514#
READ    001012R         268     316#
RESTRT  000412R         188     237#
RES1    000056R         171#
RES2    000060R         172#
RETRY1  000622R         260     280#
RETRY2  000646R         267     287#
RETRY3  000672R         269     294#
REZET   001604R         234     301     463#
ROOM    001154R         359#    427
RSTRT   000112R         188#
SBADR   000102R         191#    413*
SETUP   001504R         233     440#
SOFCNT  000042R         164#
SOFERS= 174406          211#    432
SOFPAS  000046R         166#
SPOINT  000032R         160#
SPSIZ = 000040            1#    204
SR1     000016R         153#
SR2     000020R         154#
SR3     000022R         155#
SR4     000024R         156#
START   000252R         159     211#
STAT    000026R         158#
STRT    000432R         242#    252     265     457
SVR0    000062R         173#
SVR1    000064R         174#
SVR2    000066R         175#
SVR3    000070R         176#
SVR4    000072R         177#
SVR5    000074R         178#
SVR6    000076R         179#
SYSCNT  000052R         168#
TABLE   003004R         432     473     509#
TBUF    001774R         504#
TRPDFD= 000042          211#
TRY1    003170R         256*    280*    281     554#
TRY2    003171R         287*    288     555#
```



```
TRY3    003172R         257*    294*    295     556#
VECTOR  000010R         149#    323*    456
WASADR  000104R         143#    415*
WBUFEA  000136R         198#    311     316
WBUFPA  000134R         197#    310     315
WBUFRQ  000140R         199#
WBUFSZ  000142R         200#    245     372     375     377
WCNT1   001776R         245*    246*    309     314     505#
WCNT2   002000R         239*    240*    319     506#
WDFR    000116R         190#    213*
WDTO    000114R         189#    212*
WRITCK  000760R         266     313#
WRITE   000726R         259     308#
XFLAG   000005R         147#
XMEM    001754R         311*    316*    321*    325     496#
.     = 003174R         271     507#    549#    557#

. ABS.  000000      000
        003174      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRCAD0,XRCAD0/SOL/CRF:SYM=DDXCOM,XRCAD0
 RUN-TIME:  1  1  .3 SECONDS
 RUN-TIME RATIO: 22/3=6.0
 CORE USED:  7K  (13 PAGES)
```