```
1
                .REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E748I-MC

PRODUCT NAME: CXRSAI0 DEC/X11 RH11/RS03, RS03/LA, RS04 MODULE

DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RSA IS AN IOMODX THAT EXERCISES RS03, RS03/LA AND RS04 DISK DRIVES ON AN RH11 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 8 RS03, RS03/LA AND/OR RS04'S WITH AN RH11 CONTROLLER

STORAGE:: RSA REQUIRES:
1. DECIMAL WORDS: 1223
2. OCTAL WORDS: 02367
3. OCTAL BYTES: 4616

3. PASS DEFINITION

ONE PASS OF THE RSA MODULE CONSISTS OF 1300 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 RSA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172040, VECTOR: 204, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP 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 (SOFTWARE SWITCH)

SR1 BIT0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE FUNCTION IS ABORTED AND TESTING CONTINUES

SR1 BIT0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE ERROR IS CONSIDERED FATAL AND THE DRIVE DROPPED

SR1 BIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM

SR1 BIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

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 12 RH11/RS REGISTERS IN THE FOLLOWING ORDER:

RHCS1 RHCS2 RHWC RHBA RSDA RSDS RSER RSAS
RSLA RHDB RSMR RSDT RHBAE RHCS3

SEQ 0005

```
000000'  -       IOMODX  <RSAI  >,172040,204,5,0,0,1300.,61,BUFIN,256.,1024.
000000'          MODULE  150000,RSAI  ,172040,,204,5,0,0,1300.,61,BUFIN,256.,1024.
                 .TITLE  RSAI DEC/X11 SYSTEM EXERCISER MODULE
          ;      DDXCOM  VERSION  6       23-MAY-78
                                          .LIST   BIN
                                 ;*****************************************************************************
000000'                          BEGIN:
000000' 051522  044501     040   MODNAM: .ASCII  /RSAI / ;MODULE NAME.
000005'     000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 172040                   ADDR:   172040+0                ;1ST DEVICE ADDR.
000010' 000204                   VECTOR: 204+0                   ;1ST DEVICE VECTOR.
000012'     240                  BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000                  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' 002424                   ICONT:  1300.                   ;# OF ITERATIONS PER PASS=1300.
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: 0                       ;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' 000434'                  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' 000061                   IDNUM:  61                      ;MODULE IDENTIFICATION NUMBER=61
```


SEQ 0006

```
000124' 003222'                  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                         .RPT    SPSIZ           ;MODULE STACK STARTS HERE.
                                         .NLIST
                                         .WORD   0
                                         .LIST
                                         .ENDR
000252'                          MODSP:
                                 ;*****************************************************************************
```

```
000252' 012767  002000  177636  START:  MOV     #1024.,WDFR     ;1024 WORDS FROM MEM/ITERATION
000260' 012767  000400  177626          MOV     #256.,WDTO      ;256 WORDS TO MEM.ITERATION
000266' 012767  000003  177624          MOV     #3,INTR         ;3 INTERRUPTS/ITERATION
000274' 005067  002670                  CLR     CNT             ; ZERO END OF PASS TESTER
000300' 105067  004311                  CLRB    FLAG            ; CLEAR FLAGS
000304' 005067  002646                  CLR     DLTCNT          ;CLEAR DATA LATE ERROR COUNTER
000310' 016767  177500  002672          MOV     DVID1,DVICE     ; GET DRIVE INDICATOR
000316' 016767  002666  002666          MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
000324' 122737  000015  000041          CMPB    #15,@#41        ;IF THIS DEVICE IS NOT THE LOAD MEDIA THEN
000332' 001023                          BNE     3$              ;GO TO 3$
000334' 012702  000000                  MOV     #0,R2           ;LOAD R2 WITH DRIVE NUMBER
000340' 113700  000040                  MOVB    @#40,R0         ;GET DRIVE NUMBER
000344' 012701  000001                  MOV     #1,R1           ;DRIVE MASK. INITIALIZE TO DRIVE 0
000350' 105700          1$:             TSTB    R0              ;IF DRIVE FOUND THEN
000352' 001404                          BEQ     2$              ;GO TO 2$
000354' 006301                          ASL     R1              ;ELSE SHIFT MASK TO NEXT DRIVE
000356' 105300                          DECB    R0              ;DOWNCOUNT DRIVE #
000360' 005202                          INC     R2              ;UPDATE DRIVE NUMBER
000362' 000772                          BR      1$              ;CHECK AGAIN
000364' 130167  002620          2$:     BITB    R1,DVICE        ;IF DRIVE NOT  SELECTED TO BE TESTED THEN
000370' 001404                          BEQ     3$              ;GO TO 3$
000372' 010267  002616                  MOV     R2,DRVVE        ;ELSE SET UP TO DROP DRIVE
000376' 004767  001712                  JSR     PC,DROP         ;AND GO DROP IT
000402'                         3$:
000402' 012767  177770  002554          MOV     #-8.,BLK1       ; INITIALIZE BLOCK COUNTER
000410' 012767  177777  002576          MOV     #-1,DRVVE       ; INITIALIZE DEVICE COUNTER
000416' 004767  002172                  JSR     PC,SETUP        ;GENERATE REGISTER ADDRESSES
000422' 004767  002342                  JSR     PC,REZET        ;INITIALIZE RH REGS AND DRIVES
000426' 005767  002556                  TST     DVICE           ; DROP THE MODULE ?
000432' 001477                          BEQ     FINI            ; YES
000434'                         RESTRT:
000434' 104415  000000' 000124'         GETPAS,BEGIN,,RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000442' 016767  177464  002550          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000450' 005467  002544                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
000454' 004767  001524          STRT:   JSR     PC,BLOCK        ; GET NEXT BLOCK NUMBER, DISK ADDRESS
000460' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
000464' 016767  177452  002524          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000472' 005467  002520                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
000476' 004767  001534          NEXT:   JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
000502' 005767  002502                  TST     DVICE           ; ANY DRIVES LEFT ?
000506' 001451                          BEQ     FINI            ; NO, GO DROP THE MODULE
000510' 132767  000010  004077          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
000516' 001356                          BNE     STRT            ; YES, GO GET ANOTHER BLOCK
000520' 004567  002030                  JSR     R5,READY        ; IS DRIVE READY ?
000524' 000403                          BR      1$              ; YES, CONTINUE
000526' 004767  001620                  JSR     PC,NOTRDY       ; NO, GO WAIT FOR IT
000532' 000761                          BR      NEXT            ; CONTINUE
000534' 005067  004052          1$:     CLR     TRY1            ; ZERO RETRY COUNTERS
000540' 105067  004050                  CLRB    TRY3            ;
000544' 004567  000210          GO:     JSR     R5,WRITE        ; WRITE SOME DATA
000550' 000432                          BR      RETRY1          ; IF ERRORS, TRY IT AGAIN
000552' 132767  000004  004035          BITB    #BIT2,FLAG      ; DID THE DISK OVERFLOW ?
000560' 001407                          BEQ     GOA             ; NO, CONTINUE
000562' 142767  000004  004025          BICB    #BIT2,FLAG      ; YES, CLEAR THE OVERFLOW FLAG
```



```
000570' 012767  177770  002366          MOV     #-8.,BLK1       ; RESET THE BLOCK NUMBER
000576' 000726                          BR      STRT            ; START OVER AT BEGGINING OF DISK
000600' 004567  000226          GOA:    JSR     R5,WRITCK       ; WRITE-CHECK THE DATA
000604' 000426                          BR      RETRY2          ; IF ERRORS, TRY AGAIN
000606' 004567  000272          GOB:    JSR     R5,READ         ; READ THE DATA WRITTEN
000612' 000435                          BR      RETRY3          ; IF ERRORS, TRY AGAIN
000614' 104412  000000' 000126'         CDATAS,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
000622' 000624'                         .+2                     ; IF ERROR, CONTINUE
000624' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
000630' 000722                          BR      NEXT            ; NO, CONTINUE
```

```


000632'                             FINI:
000632' 104410  000000'                     ENDS,BEGIN                  ; DROP THE MODULE
                                    ;       -------------------------



000636' 105267  003750              RETRY1: INCB    TRY1                ; COUNT THE RETRYS
000642' 122767  000003  003742              CMPB    #3,TRY1             ; LIMIT EXCEEDED ?
000650' 001335                              BNE     GO                  ; NO, GO TRY IT AGAIN
000652' 104403  000000' 004544'             MSGN$,BEGIN,EXCED1          ;ASCII MESSAGE CALL WITH COMMON HEADER
000660' 000424                              BP      NEXTA               ; GO ON TO NEXT DRIVE
                                    ;       -------------------------


000662' 105267  003725              RETRY2: INCB    TRY2                ; COUNT RETRYS
000666' 122767  000003  003717              CMPB    #3,TRY2             ; LIMIT EXCEEDED ?
000674' 001341                              BNE     GOA                 ; NO, TRY AGAIN
000676' 104403  000000' 004552'             MSGN$,BEGIN,EXCED2          ;ASCII MESSAGE CALL WITH COMMON HEADER
000704' 000412                              BR      NEXTA               ; GO ON TO NEXT DRIVE
                                    ;       -------------------------


000706' 105267  003702              RETRY3: INCB    TRY3                ; COUNT RETRYS
000712' 122767  000003  003674              CMPB    #3,TRY3             ; LIMIT EXCEEDED ?
000720' 001332                              BNE     GOB                 ; NO, GO TRY AGAIN
000722' 104403  000000' 004560'             MSGN$,BEGIN,EXCED3          ;ASCII MESSAGE CALL WITH COMMON HEADER
000730' 000400                              BR      NEXTA               ; GO ON TO NEXT DRIVE
                                    ;       -------------------------


000732' 032767  000001  177056      NEXTA:  BIT     #BIT0,SR1           ; DROP THE DRIVE
000740' 001405                              BEQ     1$                  ; NO, SKIP TO NEXT DRIVE
000742' 004767  001346                      JSR     PC,DROP             ; YES, DROP OFFENDING DRIVE
000746' 104403  000000' 004566'             MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
000754' 000167  177516              1$:     JMP     NEXT                ; GO ON TO NEXT DRIVE
                                    ;       -------------------------


                                            .MACRO  LINEUP  EABITS      ; LINE UP EA BITS FOR RHCS1
                                    ;       LINEUP  EABITS              ; LINE UP EA BITS FOR RHCS1
                                            .NLIST
                                            MOV     EABITS,R0           ; GET EXTENDED MEMORY BITS
                                            ASL     R0                  ; SHIFT 4 PLACES TO THE LEFT
                                            ASL     R0                  ; TO LINE UP WITH RHCS1
                                            ASL     R0                  ;
                                            ASL     R0                  ;
                                            MOV     R0,XMEM             ; SAVE THE SHIFTED BITS
                                            .LIST
                                            .ENDM   LINEUP
```



```
                                    ;               ----- RS DISK DRIVERS -----
000760' 012767  000161  002206      WRITE:  MOV     #161,FUNC           ; LOAD WRITE FUNCTION
000766' 016777  002224  003232              MOV     WCNT1,@RHWC         ; LOAD WORD COUNT
000774' 016777  177134  003226              MOV     WBUFPA,@RHBA        ; LOAD BUFFER ADDRESS
                                    ;       LINEUP  WBUFEA              ; LINE UP EA BITS FOR RHCS1
001022' 016777  002136  003202              MOV     BLK1,@RSDA          ; LOAD DISK ADDRESS
001030' 000467                              BR      GOGO                ; CONTINUE
001032' 012767  000151  002134      WRITCK: MOV     #151,FUNC           ; LOAD WRITE-CHECK FUNCTION
001040' 016777  002152  003160              MOV     WCNT1,@RHWC         ; LOAD WORD COUNT
001046' 016777  177062  003154              MOV     WBUFPA,@RHBA        ; LOAD BUFFER ADDRESS
                                    ;       LINEUP  WBUFEA              ; LINE UP EA BITS FOR RHCS1
001074' 016777  002064  003130              MOV     BLK1,@RSDA          ; LOAD DISK ADDRESS
001102' 000442                              BR      GOGO                ; CONTINUE
001104' 012767  000171  002062      READ:   MOV     #171,FUNC           ; LOAD READ FUNCTION
001112' 016777  002102  003106              MOV     WCNT2,@RHWC         ; LOAD WORD COUNT
001120' 016777  177002  003102              MOV     RBUFPA,@RHBA        ; LOAD BUFFER ADDRESS
                                    ;       LINEUP  RBUFEA              ; LINE UP EA BITS FOR RHCS1
001146' 016777  002014  003056              MOV     BLK2,@RSDA          ; LOAD DISK ADDRESS
001154' 000415                              BR      GOGO                ; CONTINUE

001156' 016777  002032  003040      CLEAR:  MOV     DRYVE,@RHCS2        ; LOAD UNIT ADDRESS
001164' 012777  000011  003030              MOV     #11,@RHCS1          ; ISSUE A DRIVE CLEAR
001172' 012777  000377  003040              MOV     #377,@RSAS          ; CLEAR ALL DISK ATTENTION SUMMARIES
001200' 012777  040000  003014              MOV     #BIT14,@RHCS1       ; CLEAR ANY CONTROLLER ERRORS
001206' 000205                              RTS     R5                  ; RETURN
001210' 016777  002000  003006      GOGO:   MOV     DRYVE,@RHCS2        ; LOAD UNIT SELECT
001216' 012777  001344' 176564              MOV     #NTRUPT,@VECTOR     ; SET INTERUPT ENTRY POINTER
001224' 032767  001000  176624              BIT     #ADDR22,RES1        ;22 BIT ADDRESSING?
001232' 001434                              BEQ     1$                  ;NO
001234' 017767  002770  001734              MOV     @RHBA,PA18          ;GET LOWER 18 BITS
001242' 006267  001732                      ASR     XMEM                ;SHIFT AE BITS INTO LOC 4+5
001246' 006267  001726                      ASR     XMEM
001252' 006267  001722                      ASR     XMEM
001256' 006267  001716                      ASR     XMEM
001262' 104416  000000' 003176'             MAP22$, BEGIN,PA18                  ; GET 22-BIT ADDR FROM 18-BIT ADDR
001270' 016777  001706  002732              MOV     PA22,@RHBA          ;LOAD BA REG
001276' 016777  001702  002746              MOV     EA22,@RSBAE         ;LOAD BAE REG
001304' 042767  000034  001672              BIC     #34,EA22            ;CLEAR UNWANTED BITS
001312' 000367  001666                      SWAB    EA22                ;LOCATE BITS 16,17
001316' 016767  001662  001654              MOV     EA22,XMEM           ;TO LOAD INTO CS1
001324' 056767  001650  001642      1$:     BIS     XMEM,FUNC           ; LOAD EXTENDED MEMORY BITS
001332' 016777  001636  002662              MOV     FUNC,@RHCS1         ; EXECUTE THE FUNCTION
001340' 104400  000000'                     EXIT$,BEGIN                 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001344'                             NTRUPT:
                                    ;-------------------------------------------------------
001344' 000004  000000' 001352'             PIRQ$,BEGIN,1$              ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                    ;-------------------------------------------------------
001352' 004567  000272              1$:     JSR     R5,ERRORS           ; GO CHECK FOR ERRORS
001356' 000205                              RTS     R5                  ; ERRORS DETECTED, RETURN
001360' 005725                              TST     (R5)+               ; NO ERRORS, SKIP RETRY
001362' 000205                              RTS     R5                  ; RETURN OK
```

```
001364' 016700 001574          ROOM:   MOV     RLK1,R0         ;GET THE CURRENT BLOCK NUMBER
001370' 012701 010000                  MOV     #4096.,R1       ;LOAD THE MAX. NUMBER OF BLOCKS
001374' 012767 000200  001570          MOV     #128.,BSIZ      ;SET BLOCK SIZE TO HI DENSITY
001402' 005002                         CLR     R2              ;ZERO REG. 2
001404' 142767 000001  003203          BICB    #BIT0,FLAG      ;CLEAR 32K INDICTOR
001412' 016703 176524                  MOV     WBUFS2,R3       ;GET THE TRANSFER SIZE.
001416' 160001                         SUB     R0,R1           ;GET # OF BLOCKS LEFT ON DISK
001420' 022777 000004  002622          CMP     #4,@RSDT        ;IS THIS A RS03/LA ?
001426' 001026                         BNE     7$              ;NO
001430' 006267 001536                  ASR     BSIZ            ;SET SECTOR SIZE
001434' 006267 001532                  ASR     BSIZ            ;TO 32 WD/SECTOR
001440' 022701 004000                  CMP     #2048.,R1       ;PLENTY OF ROOM LEFT?
001444' 003473                         BLE     5$              ;YES ERROR
001446' 022701 002000                  CMP     #1024.,R1       ;MORE THEN 32K LEFT?
001452' 003043                         BGT     2$              ;NO CONTINUE
001454' 152767 000001  003133          BISB    #BIT0,FLAG      ;YES SET THE INDICATOR
001462' 162701 002000                  SUB     #1024.,R1       ;SUB 32K OF BLOCKS
001466' 000435                         BR      2$
001470' 032777 000002  002552  7$:     BIT     #BIT1,@RSDT     ;HI DENSITY DRIVE?
001476' 001016                         BNE     1$              ;YES, CONTINUE
001500' 006267 001466                  ASR     BSIZ            ; NO, SET BLOCK SIZE TO 64.
001504' 022701 002000                  CMP     #1024.,R1       ;LOW DENSITY. PLENTY OF ROOM LEFT?
001510' 003451                         BLE     5$              ;YES, ERROR
001512' 022701 001000                  CMP     #512.,R1        ;MORE THAN 32K LEFT?
001516' 003021                         BGT     2$              ;NO, CONTINUE
001520' 152767 000001  003067          BISB    #BIT0,FLAG      ;YES, SET THE INDICATOR
001526' 162701 001000                  SUB     #512.,R1        ;SUBTRACT 32K WORTH OF BLOCKS
001532' 000413                         BR      2$              ;GO GET TOTAL NUMBER OF WORDS LEFT
001534' 022701 001000          1$:     CMP     #512.,R1        ;HIGH DENSITY. PLENTY OF ROOM LEFT?
001540' 003435                         BLE     5$              ;YES, ERROR
001542' 022701 000400                  CMP     #256.,R1        ;MORE THAN 32K LEFT?
001546' 003005                         BGT     2$              ;NO, CONTINUE
001550' 152767 000001  003037          BISB    #BIT0,FLAG      ;YES, SET THE INDICATOR
001556' 162701 000400                  SUB     #256.,R1        ;SUBTRACT 32K WORTH OF BLOCKS
001562' 005701                 2$:     TST     R1              ;ANY BLOCKS LEFT ON DISK?
001564' 003425                         BLE     6$              ;NO, RETURN OK
001566' 066702 001400                  ADD     BSIZ,R2         ;GET TOTAL # OF WORDS LEFT
001572' 005301                         DEC     R1              ;ALL BLOCKS ADDED IN?
001574' 003372                         BGT     2$              ;NO, KEEP ADDING
001576' 005703                         TST     R3              ;REQUEST LARGER THAN 32K?
001600' 100007                         BPL     3$              ; NO, GO CHECK THAT CONDITION
001602' 042703 100000                  BIC     #BIT15,R3       ;YES, GET RID OF 32K
001606' 132767 000001  003001          BITB    #BIT0,FLAG      ;MORE THAN 32K LEFT?
001614' 001411                         BEQ     6$              ;NO, RETURN OK
001616' 000404                         BR      4$              ;YES, GO COMPARE
001620' 132767 000001  002767  3$:     BITB    #BIT0,FLAG      ;MORE THAN 32K LEFT?
001626' 001002                         BNE     5$              ;YES, PLENTY OF ROOM, ERROR
001630' 020203                 4$:     CMP     R2,R3           ;ENOUGH ROOM FOR THE TRANSFER?
001632' 002402                         BLT     6$              ;NO, RETURN OK
001634' 005725                 5$:     TST     (R5)+           ;MUST BE A REAL ERROR
001636' 000205                         RTS     R5              ;RETURN INDICATING THE ERROR
001640' 152767 000004  002747  6$:     BISB    #BIT2,FLAG      ;SET OVERFLOW FLAG
001646' 000205                         RTS     R5              ;RETURN OK
```



```
001650' 005777 002346          ERRORS: TST     @RHCS1          ; ATTENTION OR ERROR ?
001654' 100124                         BPL     10$             ; NO, GO ON TO NEXT FUNCTION
001656' 032777 001000  002352          BIT     #BIT9,@RSER     ; ADDRESS OVERFLOW ?
001664' 001403                         BEQ     1$              ; NO, CONTINUE
001666' 004567 177472                  JSR     R5,RDOM         ; YES, IS IT A REAL ERROR ?
001672' 000513                         BR      9$              ; NO, CONTINUE
001674' 004767 000636          1$:     JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
001700' 005777 002320                  TST     @RHCS2          ;IS THIS A DATA LATE ERROR?
001704' 100012                         BPL     11$             ;NO
001706' 052767 001244                  INC     DLTCNT          ;ADD 1 TO DATA LATER ERROR COUNTER
001712' 032767 000004  176076          BIT     #BIT2,SR1       ;TYPE ERROR?
001720' 001075                         BNE     8$              ;NO
001722' 104403 000000' 004576'         MSGNS,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
001730' 000444                         BR      5$              ;CONT
001732' 032777 040000  002262  11$:    BIT     #BIT14,@RHCS1   ; TRANSFER ERROR ?
001740' 001025                         BNE     2$              ; YES
001742' 032777 020000  002252          BIT     #BIT13,@RHCS1   ; MASSBUS CONTROL PARITY ERROR ?
001750' 001025                         BNE     3$              ; YES
001752' 032777 000400  002244          BIT     #BIT8,@RHCS2    ; MASSBUS DATA PARITY ERROR ?
001760' 001025                         BNE     4$              ; YES
001762' 032777 040000  002244          BIT     #BIT14,@RSDS    ; ANY DRIVE ERRORS ?
001770' 001024                         BNE     5$              ; YES
001772' 005777 002242                  TST     @RSAS           ; ANY ATTENTIONS ACTIVE ?
001776' 001021                         BNE     5$              ; YES, CONTINUE
002000' 005067 176102                  CLR     ERRTYP  ;UNKNOWN ERROR
                                       ;*****************************************************************
002004' 104405 000000' 004222'         HRDERS,BEGIN,TABLE      ; SPECIAL CONDITION SET BUT NO REASON FOUND
                                       ;*****************************************************************
002012' 000440                         BR      8$              ; RETURN
002014'                        2$:
002014' 104403 000000' 004530'         MSGNS,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
002022' 000407                         BR      5$              ; GO DUMP REGISTERS
002024'                        3$:
002024' 104403 000000' 004534'         MSGNS,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
002032' 000403                         BR      5$              ; GO DUMP REGISTERS
002034'                        4$:
002034' 104403 000000' 004540'         MSGNS,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
002042' 005777 002172          5$:     TST     @RSAS           ; ANY ATTENTIONS ACTIVE ?
002046' 001402                         BEQ     6$              ; NO, CONTINUE
002050' 004767 000056                  JSR     PC,WHO          ; YES, FIND OUT WHICH DRIVE IT IS
002054' 016700 002164          6$:     MOV     RHDB,R0         ; SAVE ADDRESS OF DATA BUFFER
002060' 105777 002140                  TSTB    @RHCS2          ; CAN DATA BUFFER BE READ ?
002064' 100406                         BMI     7$              ; YES, CONTINUE
002066' 012767 003206' 002150          MOV     #ZERO,RHDB      ; NO, LOAD ADDRESS OF ZERO
002074' 012767 000030  176004          MOV     #30,ERRTYP      ;ERROR DURING DATA XFER
002102'                        7$:
                                       ;*****************************************************************
002102' 104405 000000' 004222'         HRDERS,BEGIN,TABLE      ; DUMP RH11 AND RS REGISTERS
                                       ;*****************************************************************
002110' 010067 002130          8$:     MOV     R0,RHDB         ; RESTORE DATA BUFFER ADDRESS
002114' 004567 177036                  JSR     R5,CLEAR        ; GO CLEAR OUT ERRORS
002120' 000205                         RTS     R5              ; ERRORS DETECTED, RETURN
002122' 004567 177030          9$:     JSR     R5,CLEAR        ; GO CLEAR OUT ANY ERRORS
002126' 005725                 10$:    TST     (R5)+           ; NO ERRORS, SKIP RETRY
002130' 000205                         RTS     R5              ; RETURN OK
```

```
002132' 017701  002122          WHO:    MOV     @RSAS,R1                ; GET THE ATTENTION SUMMARY
002136' 017704  002062                  MOV     @RHCS2,R4               ; SAVE THE STATUS REGISTER
002142' 012702  000001                  MOV     #BIT0,R2                ; SET POINTER TO DRIVE 0
002146' 005003                          CLR     R3                      ; ZERO THE DRIVE COUNTER

002150' 030201                  1$:     BIT     R2,R1                   ; IS THIS ATTENTION BIT SET ?
002152' 001006                          BNE     2$                      ; YES, CONTINUE
002154' 005203                          INC     R3                      ; NO, INCREMENT THE DRIVE COUNTER
002156' 006302                          ASL     R2                      ; SET POINTER TO NEXT DRIVE
002160' 032702  000400                  BIT     #BIT8,R2                ; ALL DONE ?
002164' 001771                          BEQ     1$                      ; NO, GO AGAIN
002166' 000207                          RTS     PC                      ; SOMEBODY LIED -- NO ATTENTIONS SET

002170' 042704  000007          2$:     BIC     #7,R4                   ; CLEAR OUT OLD UNIT NUMBER
002174' 050304                          BIS     R3,R4                   ; LOAD THE NEW UNIT NUMBER
002176' 010477  002022                  MOV     R4,@RHCS2               ; RESTORE THE STATUS REGISTER
002202' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------

002204' 062767  000010  000752  BLOCK:  ADD     #8.,BLK1                ; STEP TO NEXT BLOCK
002212' 022767  007777  000744          CMP     #4095.,BLK1             ; ALL BLOCKS DONE ?
002220' 100002                          BPL     1$                      ; NO, CONTINUE
002222' 005067  000736                  CLR     BLK1                    ; YES, START OVER
002226' 016767  000732  000732  1$:     MOV     BLK1,BLK2               ; SAME STARTING BLOCK FOR READ
002234' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------

002236' 005267  000752          DRVADR: INC     DRYVE                   ; COUNT A DRIVE
002242' 142767  000010  002345          BICB    #BIT3,FLAG              ; CLEAR END OF DRIVES FLAG
002250' 022767  000010  000736          CMP     #8.,DRYVE               ; ALL DRIVES CHECKED ?
002256' 001404                          BEQ     1$                      ; YES, GO FLAG END OF DRIVES
002260' 006267  000726                  ASR     DRIVE                   ; NO, IS NEXT DRIVE CHOSEN ?
002264' 103364                          BCC     DRVADR                  ; NO, GO TRY ANOTHER DRIVE
002266' 000411                          BR      2$                      ; RETURN
002270' 152767  000010  002317  1$:     BISB    #BIT3,FLAG              ; SET END OF DRIVES FLAG
002276' 012767  177777  000710          MOV     #-1,DRYVE               ; RESET DRIVE COUNTER
002304' 016767  000700  000700          MOV     DVICE,DRIVE             ; RESTORE CHOSEN DRIVES
002312' 000207                  2$:     RTS     PC                      ; RETURN
                                ;       ------------------------

002314' 012701  000001          DROP:   MOV     #1,R1                   ; INITIALIZE DROP PICKER
002320' 016700  000670                  MOV     DRYVE,R0                ; GET THE DRIVE NUMBER
002324' 001403                          BEQ     2$                      ; IF DRIVE 0 GO DROP IT
002326' 006301                  1$:     ASL     R1                      ; POINT TO NEXT DRIVE
002330' 005300                          DEC     R0                      ; IS THIS THE ONE ?
002332' 001375                          BNE     1$                      ; NO, LOOK AGAIN
002334' 040167  000650          2$:     BIC     R1,DVICE                ; DROP THE DRIVE
                                        ;**************************************************************
                                                                        ;CONVERT DRYVE TO ASCII AND
                                                                        ;STORE AT ADR1
002340' 104420  000000' 003214'         OTOA$,BEGIN,DRYVE,ADR1
002346' 004602'
                                        ;**************************************************************
002350' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------
```



```
002352' 012767  177777  000634  NOTRDY: MOV     #-1,DRYVE               ; START WITH FIRST DRIVE
002360' 016767  000624  000624          MOV     DVICE,DRIVE             ; RESET DRIVE SELECT
002366' 004767  177644          1$:     JSR     PC,DRVADR               ; GET A DRIVE ADDRESS
002372' 132767  000010  002215          BITB    #BIT3,FLAG              ; ALL DRIVES CHECKED ?
002400' 001043                          BNE     3$                      ; YES, RETURN
002402' 012767  000010  000552          MOV     #10,CLOCK               ;SETUP LARGE TIMEOUT ROUTINE
                                                                        ;TO GIVE DRIVES TIME TO GET UP TO SPEED
002410' 012767  077777  000542  4$:     MOV     #77777,CLK              ; SET THE TIMER
002416' 004567  176534          2$:     JSR     R5,CLEAR                ; CLEAR OUT ANY ERRORS
002422' 004567  000126                  JSR     R5,READY                ; IS THIS DRIVE READY ?
002426' 000757                          BR      1$                      ; YES, CONTINUE
002430' 104407  000000'                 BREAKS,BEGIN                    ;TEMPORARY RETURN TO MONITOR....
002434' 104407  000000'                 BREAKS,BEGIN                    ;THEN CONTINUE AT NEXT INSTRUCTION.
002440' 005367  000514                  DEC     CLK                     ; WAIT SOME MORE ?
002444' 001364                          BNE     2$                      ; YES
002446' 005367  000510                  DEC     CLOCK                   ;TIME?
002452' 001356                          BNE     4$                      ;NO CONTINUE WAITING
002454' 004767  000056                  JSR     PC,ERSUB1               ; LOAD ERROR INFORMATION
002460' 012767  000006  175420          MOV     #6,ERRTYP               ;DEVICE NOT READY
                                        ;**************************************************************
002466' 104405  000000' 004222'         HRDER$,BEGIN,TABLE              ; DRIVE NOT READY ... TIME-OUT
                                        ;**************************************************************
002474' 004767  177614                  JSR     PC,DROP                 ; NO, DROP THE DRIVE
002500' 104403  000000' 004566'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
002506' 000727                          BR      1$                      ; CHECK REST OF DRIVES
002510' 000207                  3$:     RTS     PC                      ; RETURN
                                ;       ------------------------

002512' 014167  175370          ERSUB2: MOV     -(R1),ASB               ; LOAD THE DATA
002516' 010167  175360                  MOV     R1,SBADR                ; LOAD ADDRESS OF DATA WRITTEN
002522' 014267  175362                  MOV     -(R2),AWAS              ; LOAD THE DATA
002526' 010267  175352                  MOV     R2,WASADR               ; LOAD ADDRESS OF DATA READ
002532' 005721                          TST     (R1)+                   ; RESET REG. 1
002534' 005722                          TST     (R2)+                   ; RESET REG. 2
002536' 016767  001460  175334  ERSUB1: MOV     RHCS1,CSRA              ; LOAD ADR OF CURRENT CSR
002544' 017767  001452  175330          MOV     @RHCS1,ACSR             ; LOAD CONTENTS OF CURRENT CSR
002552' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------

002554' 016777  000434  001442  READY:  MOV     DRYVE,@RHCS2            ; LOAD UNIT ADDRESS
002562' 017700  001446                  MOV     @RSDS,R0                ; SAVE STATUS IN REG. 0
002566' 105700                          TSTB    R0                      ; DRIVE READY ?
002570' 100007                          BPL     1$                      ; NO
002572' 032700  000400                  BIT     #BIT8,R0                ; DRIVE PRESENT ?
002576' 001404                          BEQ     1$                      ; NO
002600' 032700  010000                  BIT     #BIT12,R0               ; MEDIUM ON-LINE ?
002604' 001401                          BEQ     1$                      ; NO
002606' 000205                          RTS     R5                      ; RETURN, READY
002610' 005725                  1$:     TST     (R5)+                   ; SKIP INSTRUCTION FOLLOWING CALL
002612' 000205                          RTS     R5                      ; RETURN, NOT READY
                                ;       ------------------------
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
002614' 016700  175166          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
002620' 010067  001376                  MOV     R0,PHCS1        ; GENERATE REGISTER ADDRESSES
002624' 005720                          TST     (R0)+
002626' 010067  001374                  MOV     R0,PHWC
002632' 005720                          TST     (R0)+
002634' 010067  001370                  MOV     R0,PHBA
002640' 005720                          TST     (R0)+
002642' 010067  001364                  MOV     R0,PSDA
002646' 005720                          TST     (R0)+
002650' 010067  001350                  MOV     R0,PHCS2
002654' 005720                          TST     (R0)+
002656' 010067  001352                  MOV     R0,PSDS
002662' 005720                          TST     (R0)+
002664' 010067  001346                  MOV     R0,PSER
002670' 005720                          TST     (R0)+
002672' 010067  001342                  MOV     R0,PSAS
002676' 005720                          TST     (R0)+
002700' 010067  001336                  MOV     R0,PSLA
002704' 005720                          TST     (R0)+
002706' 010067  001332                  MOV     R0,PHDB
002712' 005720                          TST     (R0)+
002714' 010067  001326                  MOV     R0,PSMR
002720' 005720                          TST     (R0)+
002722' 010067  001322                  MOV     R0,PSDT
002726' 032767  001000  175122          BIT     #ADDR22,RES1    ;22 BIT ADDRESSING?
002734' 001406                          BEQ     1$              ;NO
002736' 005720                          TST     (R0)+
002740' 010067  001306                  MOV     R0,PSPAE
002744' 005720                          TST     (R0)+
002746' 010067  001302                  MOV     R0,PSCS3
                                ;
002752' 016700  175032          1$:     MOV     VECTOR,R0       ; GET VECTOR ADDRESS
002756' 012720  000454'                 MOV     #STPT,(R0)+     ; SET POINTER JUST IN CASE
002762' 116710  175024                  MOVB    PR1,(R0)        ; SET PRIORITY
                                ;
002766' 000207                  ;       RTS     PC              ; RETURN
                                ;       ----------------------------
```



```
                                ;
                                ;
                                ;
                                ;
002770' 012777  000040  001226  REZET:  MOV     #BIT5,@RHCS2    ; ISSUE AN RH11 INIT
002776' 012767  077777  000154          MOV     #77777,CLK      ; SET THE TIMER
003004' 105777  001212          1$:     TSTB    @RHCS1          ; CONTROLLER READY ?
003010' 100420                          BMI     2$              ; YES, CONTINUE
003012' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003016' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003022' 005367  000132                  DEC     CLK             ; WAIT SOME MORE ?
003026' 001366                          BNE     1$              ; YES
003030' 005067  000154                  CLR     DVICE           ; NO, SET TO DROP THE MODULE
003034' 012767  000003  175044          MOV     #3,ERRTYP       ;CONTROLLER NOT READY
                                        ;*****************************************************************
003042' 104405  000000' 004222'         ARDERS,BEGIN,TABLE      ; CONTROLLER NOT READY
                                        ;*****************************************************************
003050' 000207                          RTS     PC              ; RETURN
                                ;
003052' 004767  177274          2$:     JSR     PC,NOTRDY       ; MAKE SURE ALL CHOSEN DRIVES ARE READY
003056' 005767  000126                  TST     DVICE           ; ANY DRIVES LEFT ?
003062' 001434                          BEQ     6$              ; NO, RETURN
003064' 004767  177146          3$:     JSR     PC,DRVADR       ; GET A DRIVE NUMBER
003070' 132767  000010  001517          BITB    #BIT3,FLAG      ; ALL DRIVES RESET ?
003076' 001026                          BNE     6$              ; YES, RETURN
003100' 016777  000110  001116          MOV     DRVVE,@RHCS2    ; NO, LOAD NEXT DRIVE ADDRESS
003106' 032777  174000  001134          BIT     #174000,@RSDT   ; IS THIS AN RS DISK ?
003114' 001013                          BNE     5$              ; NO, GO DROP IT
003116' 004567  176034          4$:     JSR     R5,CLEAR        ; GO CLEAR OUT ANY ERRORS
003122' 000760                          BR      3$              ; KEEP GOING
003124' 004767  177164          5$:     JSR     PC,DROP         ; DROP THE NON-RS DRIVE
003130' 012767  000006  174750          MOV     #6,ERRTYP       ;NOT AN RS OR NOT READY
                                        ;*****************************************************************
003136' 104405  000000' 004222'         ARDERS,BEGIN,TABLE      ; NOT AN RS DRIVE
                                        ;*****************************************************************
003144' 104403  000000' 004566'         MSGNS,BEGIN,DPP ;ASCII MESSAGE CALL WITH COMMON HEADER
003152' 000744                          BR      3$              ; MAKE SURE ALL GET RESET
003154' 000207                  6$:     RTS     PC              ; RETURN
                                ;       ----------------------------
```

```
690  003156' 000000                  DLTCNT: 0
691  003160' 000000                  CLK:    0
692  003162' 000000                  CLOCK:  0
693  003164' 000000                  BLK1:   0
694  003166' 000000                  BLK2:   0
695  003170' 000000                  CNT:    0
696  003172' 000000                  BSIZ:   0
697  003174' 000000                  FUNC:   0
698                                  ;DO NOT  CHANGE THE ORDER OF THE NEXT FOUR LOCATIONS
699  003176' 000000                  PA18:   0
700  003200' 000000                  XMEM:   0
701  003202' 000000                  PA22:   0
702  003204' 000000                  EA22:   0
703  003206' 000000                  ZERO:   0
704  003210' 000000                  DVICE:  0
705  003212' 000000                  DRIVE:  0
706  003214' 000000                  DRYVE:  0
707  003216' 000000                  WCNT1:  0
708  003220' 000000                  WCNT2:  0
709  003222' 000400                  BUFIN:  .BLKW   256.
710  004222'                         TABLE:
711  004222' 000000                  RHCS1:  0
712  004224' 000000                  RHCS2:  0
713  004226' 000000                  RHWC:   0
714  004230' 000000                  RHBA:   0
715  004232' 000000                  RSDA:   0
716  004234' 000000                  RSDS:   0
717  004236' 000000                  RSER:   0
718  004240' 000000                  RSAS:   0
719  004242' 000000                  RSLA:   0
720  004244' 000000                  RHDB:   0
721  004246' 000000                  RSMR:   0
722  004250' 000000                  RSDT:   0
723  004252' 000000                  RSBAE:  0
724  004254' 000000                  RSCS3:  0
725  004256' 177777                  177777
726  004260' 020040  051124  047101  MES1:   .ASCIZ  "  TRANSFER  ERROR*"
727  004268' 043123  051105  020040
728  004274' 051105  047522  022522
729  004302'     000
730  004303'     040  046440  051501  MES2:   .ASCIZ  "  MASSBUS  PARITY  ERROR*"
731  004310' 041123  051525  020040
732  004316' 040520  044522  054524
733  004324' 020040  051105  047522
734  004332' 022522      000
735  004335'     040  046440  051501  MES3:   .ASCIZ  "  MASSBUS  DATA  PARITY  ERROR*"
736  004342' 041123  051525  020040
737  004350' 040504  040524  020040
738  004356' 040520  044522  054524
739  004364' 020040  051105  047522
740  004372' 022522      000
741  004375'     040  042040  044522  MES4:   .ASCIZ  "  DRIVE  "
742  004402' 042526  020040      000
```



```
743
744
745  004407'     040  042040  047522  MES5:   .ASCIZ  "  DROPPED*"
746  004414' 050120  042105  000045
747  004422' 051040  052105  054522  MES6:   .ASCIZ  " RETRY EXCEEDED*"
748  004430' 042440  041530  042505
749  004436' 042504  022504      000
750  004443'     040  020040  051127  MES7:   .ASCIZ  "   WRITE"
751  004450' 052111  000105
752  004454' 020040  053440  044522  MES8:   .ASCIZ  "   WRITE-CHECK"
753  004462' 042524  041455  042510
754  004470' 045503      000
755  004473'     040  020040  042522  MES9:   .ASCIZ  "   READ"
756  004500' 042101      000
757  004503'     040  020040  042040  MES10:  .ASCIZ  "    DATA LATE ERROR*"
758  004510' 052101  020101  040514
759  004516' 042524  042440  051122
760  004524' 051117  000045
761                                  .EVEN
762  004530' 004260'                 TRERR:  MES1
763  004532' 177777                  177777
764  004534' 004303'                 MCPERR: MES2
765  004536' 177777                  177777
766  004540' 004335'                 MDPERR: MES3
767  004542' 177777                  177777
768  004544' 004443'                 EXCED1: MES7
769  004546' 004422'                 MES6
770  004550' 177777                  177777
771  004552' 004454'                 EXCED2: MES8
772  004554' 004422'                 MES6
773  004556' 177777                  177777
774  004560' 004473'                 EXCED3: MES9
775  004562' 004422'                 MES6
776  004564' 177777                  177777
777  004566' 004375'                 DRP:    MES4
778  004570' 004607'                 NUMB
779  004572' 004407'                 MES5
780  004574' 177777                  177777
781  004576' 004503'                 DLTERR: MES10
782  004600' 177777                  177777
783  004602' 000005                  ADR1:   .BLKB   5
784  004607'     000                 NUMB:   .BYTE   0
785  004610'     000                 .BYTE   0
786          004612'                 .EVEN
787  004612'     000                 TRY1:   .BYTE   0
788  004613'     000                 TRY2:   .BYTE   0
789  004614'     000                 TRY3:   .BYTE   0
790  004615'     000                 FLAG:   .BYTE   0
791                                  ;       ------------------------
792
793          000001                          .END
```

```
ACSR    000162R  193#  589*
ADDR    000046R  153#  611
ADDR22= 001000   222#  367  635
ADR1    004662R  549   783#
ASR     000166R  197#  582*
ASTAT   000104R  195#
AWAS    000110R  198#  584*
BEGIN   000000R  156#  253  258  284  286  292  300  308  316  324  374  382  385
                 456   470  474  477  480  491  549  565  580  674  677  656  657
                 663   681  683
BIT0  = 000001   222#  321  395  406  416  423  433  436  501
BIT1  = 000002   222#  409
BIT10 = 002000   222#
BIT11 = 004000   222#
BIT12 = 010000   222#  599
BIT13 = 020000   222#  460
BIT14 = 040000   222#  363  458  464
BIT15 = 100000   222#  432
BIT2  = 000004   222#  275  277  442  454
BIT3  = 000010   222#  265  527  533  557  671
BIT4  = 000020   222#
BIT5  = 000040   222#  652
BIT6  = 000100   222#
BIT7  = 000200   222#
BIT8  = 000400   222#  462  508  597
BIT9  = 001000   222#  446
BLK1    003164R  246*  278*  345  351  391  518*  519  521*  522  693#
BLK2    003166R  537*  694#
BLOCK   002204R  257   518#
BREAK$= 104407   222#  565  656  657
BR1     000012R  161#  644
BR2     000013R  162#
BSIZ    003172R  393*  400*  401*  411*  427  696#
BTOD$ = 104421   222#
BUFIN   003222R  204   709#
CDATA$= 104412   222#  284
CDERCT  000144R  212#
CDWDCT  000146R  213#
CLEAR   001156R  360#  494  496  562  676
CLK     003160R  561*  567*  653*  658*  691#
CLOCK   003162R  550*  569*  692#
CNT     003170R  225*  695#
CONFIG  000056R  181#  588*
CSRA    000160R  191#
DATCK$= 104411   222#
DATER$= 104404   222#
DLTCNT  003156R  227*  453*  690#
DLTERR  004576R  456   781#
DRIVE   003212R  229*  530*  535*  705#
DROP    002314R  244   323   539#  576   678
DRP     004566R  324   577   683   777#
DRVADR  002236R  262   526#  531   556   670
DRYVE   003214R  243*  247*  360   365   526*  528   534*  540   549   554*  593   673   706#
DVICE   003210R  228*  229   241   250   263   535   545*  555   660*  668   704#
DVID1   000014R  163#  220
EA22    003204R  376   377*  378*  379   702#
```



```
ENDIT$= 104413   222#  286
END$  = 104410   222#  292
ERRORS  001650R  387   444#
ERRTVP  000106R  196#  468*  488*  572*  661*  679*
ERSUB1  002536R  450   571   588#
ERSUB2  002512R  592#
EXCED1  004544R  300   768#
EXCED2  004552R  308   771#
EXCED3  004560R  316   774#
EXITS = 104400   222#  382
FINI    000632R  251   264   291#
FLAG    004615R  225*  265   275   277*  395*  406*  416*  423*  433   436   442*  527*  533*
                 557   671   790#
FREE    000150R  214#
FUNC    003174R  341*  347*  353*  380*  381   697#
GETPA$= 104415   222#  253
GO      000544R  273#  299
GOA     000600R  276   280#  307
GOB     000606R  282#  315
GOGO    001210R  346   352   358   365#
GWBUF$= 104414   222#  258
HRDCNT  000044R  176#
HRDER$= 104405   222#  470   491   574   663   681
HRDPAS  000050R  178#
ICONT   000036R  173#
ICOUNT  000040R  174#
IDNUM   000122R  203#
IMODX.= 000000   215#  259
INIT    000030R  170#
INTR    000120R  202#  224*
MAP22$= 104416   222#  374
MCPERR  004534R  477   764#
MDPERR  004540R  480   765#
MES1    004260R  726#  752
MES10   004593R  757#  781
MES2    004303R  730#  764
MES3    004335R  735#  766
MES4    004375R  741#  777
MES5    004807R  745#  779
MES6    004422R  747#  769   772   775
MES7    004443R  750#  768
MES8    004454R  752#  771
MES9    004473R  755#  774
MODNAM  000007R  157#
MODSP   000252R  171   220#
MSGN$ = 104403   222#  300   308   316   324   456   474   477   480   577   683
MSG$  = 104402   222#
MSG$  = 104401   222#
NEXT    000476R  262#  270   288   325
NEXTA   000732R  301   309   317   321#
NOTRDY  002352R  269   554#  667
NTRUPT  001344R  366   383#
NULL  = 000000   222#
NUMB    004607R  778   784#
OPEN  = 000000   158   164   165   166   167   184   185   186   187   188   189   190   191
                 193   195   197   198   200   201   202   205   206   208   209   211   212
```

```
OTOAS = 104420                  213     214     222#
PASCNT  000034R                 222#    549
PA18    003176R                 366*    374     699#
PA22    003302R                 375     701#
PIRQS = 200004                  222#    385
PDPSP = 005726                  222#
PDPSP2= 022626                  222#
PRTY  = 000016                  222#
PRTY0 = 000000                  162     222#
PRTY1 = 000040                  222#
PRTY2 = 000100                  222#
PRTY3 = 000140                  222#
PRTY4 = 000200                  222#
PRTY5 = 000240                  161     222#
PRTY6 = 000300                  222#
PRTY7 = 000340                  222#
PS    = 177776                  222#
PSW   = 177776                  222#
PUSH  = 005746                  222#
PUSH2 = 024646                  222#
RAND$ = 104417                  222#
RANNUM  000254R                 186#
RBUFEA  001308R                 206#    357
RBUFPA  001268R                 205#    284     355
RBUFSZ  001328R                 207#    254
RBUFVA  001248R                 204#    253
READ  = 005404                  223#    353#
READY   002554R                 267     563     593#
RESTRT  000434R                 199     252#
RES1    000568R                 182#    367     635
RES2    000608R                 183#
RETRY1  000636R                 274     297#
RETRY2  000662R                 281     305#
RETRY3  000706R                 293     313#
REZET   002770R                 248     552#
RHBA    004236R                 343*    349*    355*    369     375*    616*    714#
RHCS1   004222R                 361*    363*    381*    444     458     460     588     589     612*    654     711#
RHCS2   004224R                 360*    365*    451     462     485     500     514*    593*    620*    652*    673*    712#
RHDB    004244R                 484     487*    493*    630*    720#
RHWC    004226R                 347*    348*    354*    614*    713#
ROOM    001364R                 301#    448
RSAS    004240R                 362*    466     481     499     626*    718#
RSBAE   004252R                 376*    638*    723#
RSCS3   004254R                 640*    724#
RSDA    004232R                 345*    351*    357*    618*    715#
RSDS    004234R                 464     594     622*    716#
RSDT    004250R                 398     409     634*    674     722#
RSER    004236R                 446     624*    717#
RSLA    004242R                 628*    719#
RSMR    004246R                 632*    721#
RSTRT   000112R                 199#
SBADR   000102R                 192#    583*
SETUP   002514R                 248     611#
SOFCNT  000042R                 175#
SOFERS= 104406                  222#
```



```
SOFPAS  000046R                 177#
SPDINT  000032R                 171#
SPSIZ = 000040                  174     215
SR1     000016R                 164#    321     454
SR2     000020R                 165#
SR3     000022R                 166#
SR4     000024R                 167#
START   000252R                 170     222#
STAT    000026R                 169#
STRT    004454R                 257#    266     279     643
SVR0    000062R                 194#
SVR1    000064R                 195#
SVR2    000066R                 196#
SVR3    000070R                 187#
SVR4    000072R                 188#
SVR5    000074R                 189#
SVR6    000076R                 190#
SYSCNT  000052R                 179#
TABLE   004222R                 470     491     574     663     681     710#
TRERR   004530R                 474     762#
TRPDFD= 000022                  222#
TRY1    004612R                 271*    297*    298     787#
TRY2    004613R                 305*    306     788#
TRY3    004614R                 272*    313*    314     789#
VECTOR  000010R                 160#    346*    642
WASADP  001104R                 194#    595*
WBUFEA  001360R                 209#    345     351
WBUFPA  001134R                 208#    343     349
WBUFRQ  001140R                 210#
WBUFSZ  001142R                 211#    259     396
WCNT1   003216R                 259*    260*    342     348     707#
WCNT2   003220R                 254*    255*    354     708#
WDFD    000116R                 201#    222*
WDTO    000114R                 200#    223*
WHO     002132R                 483     499#
WRITCK  001032R                 280     347#
WRITE   000760R                 273     341#
XFLAG   000056R                 158#
XMEM    003200R                 345*    351*    357*    370*    371*    372*    373*    379*    380     700#
ZERO    003248R                 497     703#
.       = 004616R               285     709#    783#    786#

. ABS.  000000      000
        004616      001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  2

XRSAIO,XRSAIO/SOL/CRF:SYM=DDXCOM,XRSAIO
RUN-TIME:  1  2  .4 SECONDS
RUN-TIME RATIO: 214=4.6
CORE USED:  7K  (13 PAGES)
```