.REM -

IDENTIFICATION

PRODUCT CODE: AC-E965F-MC

PRODUCT NAME: CXRLAF0 RL11/RLV11/RLV12/RL01/RL02

PRODUCT DATE: FEBRUARY 1981

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RLA IS AN IOMODX THAT EXERCISES RL01/RL02 DISK DRIVES ON RL11/RLV11/RLV12 CONTROLLERS. IT EXERCISES THE DRIVES BY DOING READ HEADERS, SEEKS, READS, WRITES AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE DEVICE.

2. REQUIREMENTS

HARDWARE: 1 TO 4 RL01/RL02 DISK DRIVES (WITH SCRATCH PACKS).
1 RL11, RLV11, OR RLV12 DISK CONTROLLER.

STORAGE:: RLA REQUIRES:
1. DECIMAL WORDS: 1445
2. OCTAL WORDS: 2645
3. OCTAL BYTES: 5512

3. PASS DEFINITION

ONE PASS OF THE RLA MODULE CONSISTS OF 20 X 100 CYCLES OF THE BASIC TEST SEQUENCE (READ HEADER, SEEK, READ HEADER, WRITE, WRITE CHECK, READ). THE TEST SEQUENCE WRITES 1024 WORDS, READS BACK THE FIRST 256, AND DATA CHECKS THE SAME.

4. EXECUTION TIME

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

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DEVADR: 174400, VECTOR: 160, BR1: 5, DEVCNT: 1

6. DEVICE/OPTION SETUP

INSURE THAT ALL DRIVES UNDER TEST ARE POWERED UP, WRITE ENABLED, AND READY -- AND OF COURSE, SCRATCH PACKS INSTALLED !!!

IF MORE THAN 1 DRIVE, CHANGE DVID1: (LOC 14') ACCORDINGLY.

IF RLV12 CONTROLLER, SET SR1: = 10 (LOC 16', BIT 3).

7. SR1 OPTIONAL SETUP

BIT 0 - DROP DRIVE ON ERROR

BIT 1 - RANDOM SEEKS

BIT 2 - DON'T PRINT SOFT ERRORS

BIT 3 - CONTROLLER IS AN RLV12

8. ERROR REPORTING

ON ERROR ALL REGISTERS ARE PRINTED IN THE FOLLOWING ORDER:

RL11/RLV11 RLCS RLBA RLDA RLMP DRIVE STATUS

RLV12 RLCS RLBA RLDA RLMP RLBAE DRIVE STATUS

EDIT:	BY:	DATE:	REASON:
1	G.PASQUINTONIO	MAY-81	"HNF" ERRORS WERE BEING REPORTED AS "OPI".
2	G.PASQUINTONIO	MAY-81	CONTROLLER ERROR ON WRITE-CHECK CAUSED A TERMINAL TABLE SEARCH (ULTIMATELY TRAPPING TO 4).
3	G.PASQUINTONIO	MAY-81	"DROP ON ERROR" OPTION (SR1<0>) NOT IMPLEMENTED CORRECTLY.
4	G.PASQUINTONIO	MAY-81	RLV12 22 BIT UPGRADE.

```
                        IOMODX  <RLAF >,174400,160,5,0,0,20.,146,BUFIN,256.,1024.
                        MODULE  150000,RLAF ,174400,160,5,0,0,20.,146,BUFIN,256.,1024.
                        .TITLE  RLAF DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046122  043101     040  MODNAM: .ASCII  /RLAF / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 174400                  ADDR:   174400+0                ;1ST DEVICE ADDR.
000010' 000160                  VECTOR: 160+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' 000024                  ICONT:  20.                     ;# OF ITERATIONS PER PASS=20.
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' 000424'                 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' 000146                  IDNUM:  146                     ;MODULE IDENTIFICATION NUMBER=146
000124' 003634'                 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:
                                ;*****************************************************************************
  220
  221 000252' 012767 002000 177636 START: MOV     #1024.,WDFR     ;1024. WORDS FROM MEM/ITERATION
  222 000260' 012767 000400 177626        MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
  223 000266' 012767 000005 177624        MOV     #5,INTR         ;5 INTERRUPTS/ITERATION
  224 000274' 012767 000057 004422        MOV     #57,NUMB
  225 000302' 012767 177400 003314        MOV     #-400,DRIVE     ;SET DRIVE SELECT
  226 000310' 012767 000001 003252        MOV     #1,DRVMSK       ;SETUP DRIVE SELECT MASK
  227 000316' 005067 003272               CLR     DLTCNT          ;CLEAR DATA LATE COUNT
  228 000322' 004767 001300               JSR     PC,SETUP        ;GO SET UP REGISTERS
  229 000326' 016767 177462 003272        MOV     DVID1,DVICE     ;COPY DRIVE SELECTION
  230 000334' 122737 000014 000041 3$:    CMPB    #14,@#41        ;WAS RL LOAD DEVICE?
  231 000342' 001020                      BNE     7$              ;N-BRANCH; Y-SEE IF LOAD UNIT SELECTED
  232 000344' 012702 000001               MOV     #1,R2           ;SET UP FOR MASK
  233 000350' 113701 000040               MOVB    @#40,R1         ;GET LOAD UNIT
  234 000354' 001403                      BEQ     5$              ;IF ZERO GO MASK OUT UNIT
  235 000356' 006302                4$:   ASL     R2              ;SHIFT MASK
  236 000360' 105301                      DECB    R1              ;DEC COUNT
  237 000362' 001375                      BNE     4$              ;KEEP CHECKING
  238 000364' 030267 003236         5$:   BIT     R2,DVICE        ;WAS THAT DRIVE SELECTED?
  239 000370' 001405                      BEQ     7$              ;N-BRANCH; Y-CONTINUE
  240 000372' 040267 003230               BIC     R2,DVICE        ;DELETE UNIT FROM DEVICE MAP
  241 000376' 104403 000000' 005472'      MSGN$,BEGIN,DROPLD      ;ASCII MESSAGE CALL WITH COMMON HEADER
  242 000404' 005767 003216         7$:   TST     DVICE           ;ANY DRIVES SELECTED?
  243 000410' 001011                      BNE     RSTRT1          ;YES, CONTINUE
  244 000412' 104403 000000' 005422'      MSGN$,BEGIN,ABORT       ;ASCII MESSAGE CALL WITH COMMON HEADER
  245 000420' 000167 001176               JMP     FINI            ;MESSAGE, DROP MODULE
  246 000424' 005767 003160         RESTRT: TST     CNT             ;+    / SUPPORT
  247 000430' 001001                      BNE     RSTRT1          ;+   /  FOR
  248 000432' 000707                      BR      START           ;+  /   DT03
  249 000434'                       RSTRT1:                       ;+/
  250 000434' 104415 000000' 000124'      GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
  251 000442' 000421                      BR      CHKDR1
  252 000444' 006367 003120         LOOPL: ASL    DRVMSK          ;SHIFT MASK FOR NEXT DRIVE
  253 000450' 022767 000020 003112        CMP     #20,DRVMSK      ;DRIVE MASK OVERSHIFT CHECK
  254 000456' 001013                      BNE     CHKDR1          ;BRANCH IF MASK OK
  255 000460' 012767 000001 003102        MOV     #1,DRVMSK       ;RESET DRIVE SELECT MASK
  256 000466' 012767 000057 004230        MOV     #57,NUMB
  257 000474' 012767 177400 003122        MOV     #-400,DRIVE     ;RESET DRIVE SELECT
  258 000502'                       CHKDRV:
  259 000502' 104413 000000'              ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
```

```
                                                                ;MONITOR SHALL TEST END OF PASS
000506'                                 CHKDR1:
000506' 062767  000400  003110  1$:     ADD     #400,DRIVE      ;NEXT DRIVE
000514' 005267  004204                  INC     NUMB
000520' 036767  003044  003100          BIT     DRVMSK,DVICE    ;IS THAT DRIVE PRESENT
000526' 001746                          BEQ     LOOPL           ;NO, GO FOR NEXT ONE
000530' 005067  002770                  CLR     RETRY           ;CLEAR A FEW LOCATIONS
000534' 005067  003024                  CLR     RWER            ;READ WRITE ERROR FLAG
000540' 005067  003044                  CLR     CNT             ;COUNT
                                        ;
                                        ;WE HAVE A DRIVE, START TESTING
                                        ;
000544' 004767  000722                  JSR     PC,WTRDY
000550' 004567  001430                  JSR     R5,DRVRTS       ;ISSUE DRIVE RESET, CLEAR VOLUME
000554' 012767  100077  003022          MOV     #100077,MASK2   ;
000562' 012767  077724  003002          MOV     #77724,FSTBSC   ;SET UP FOR RL01
000570' 012767  077750  002776          MOV     #77750,LSTBSC
000576' 012767  100177  002772          MOV     #100177,MASK
000604' 012767  077600  002766          MOV     #77600,LSTCYL
000612' 012767  077700  002762          MOV     #77700,LSTTRK
000620' 032767  000200  002726          BIT     #200,T.MP       ;TEST RL01 OR RL02
000626' 001422                          BEQ     2$              ;RL01 BRANCH
000630' 042767  100000  002740          BIC     #100000,MASK    ;FIX FOR RL02
000636' 052767  100000  002734          BIS     #100000,LSTCYL
000644' 052767  100000  002730          BIS     #100000,LSTTRK
000652' 052767  100000  002714          BIS     #100000,LSTBSC
000660' 042767  100000  002716          BIC     #100000,MASK2   ;
000666' 052767  100000  002676          BIS     #100000,FSTBSC
000674' 012767  000201  002720  2$:     MOV     #201,DIFWD
000702' 004567  001256          LOOP:   JSR     R5,RDHDR        ;READ HEADER ON DISK
000706' 016767  002642  002704          MOV     T.MP,HDRWD      ;GET HEADER
                                        ;
                                        ;CHECK TO SEE IF RANDOM SEEK IS REQUESTED, BIT 1 OF SR1
                                        ;SET INDICATES A RANDOM SEEK OTERWISE SEEK IS INCREMENTAL
                                        ;
000714' 032767  000002  177074  TAG:    BIT      #BIT1,SR1      ;INCREMENTAL OR RANDOM SEEKS?
000722' 001446                          BEQ     TAG1            ;INCREMENTAL, TAG1
000724' 042767  000177  002666          BIC     #177,HDRWD      ;CLEAR HEAD AND SECTOR BITS
000732' 104417  000000'                 RAND$,BEGIN
000736' 016700  177112                  MOV     RANNUM,R0       ;STORE IT AWAY
000742' 010001                          MOV     R0,R1           ;SAVE A COPY
000744' 046700  002626                  BIC     MASK,R0         ;CLEAR HEAD AND SECTOR
000750' 010067  002646                  MOV     R0,DIFWD        ;LET'S CALCULATE DIFFERENCE WORD
000754' 166767  002640  002640          SUB     HDRWD,DIFWD     ;GET DIFFERENCE TO SEEK
000762' 103003                          BCC     1$
000764' 005467  002632                  NEG     DIFWD           ;MAKE DIFF ABSOLUTE
000770' 000403                          BR      2$
000772' 052767  000004  002622  1$:     BIS     #4,DIFWD        ;SET DIRECTION BIT
001000' 052767  000001  002614  2$:     BIS     #1,DIFWD        ;SET MARKER
001006' 032701  000100                  BIT     #100,R1         ;TEST HEAD
001012' 001403                          BEQ     3$              ;IF 0, DON'T SET HEAD IN DIFF
001014' 052767  000020  002600          BIS     #20,DIFWD       ;SET HEAD
001022' 010167  002572          3$:     MOV     R1,HDRWD        ;GET EXPECTED HEADER
001026' 046767  002552  002564          BIC     MASK2,HDRWD     ;CLEAR SECTOR BITS
001034' 000167  000134                  JMP     TAG2
```

```
001040' 042767  000177  002552  TAG1:   BIC     #177,HDRWD          ;CLEAR OUT SECTOR BITS & HEAD
001046' 032767  177600  002544          BIT     #177600,HDRWD       ;ON TRACK 0?
001054' 001007                          BNE     1$                  ;NO, GO CHECK FOR LAST CYLINDER
001056' 012767  000200  002534          MOV     #200,HDRWD          ;SET NEXT ADDRESS=CYL 1
001064' 012767  000205  002530          MOV     #205,DIFWD          ;DIF WD 1, MARKER, SEEK IN, HS=0
                                                                    ;SET CURRENT HD=0, SEEK IN
001072' 000440                          BR      TAG2
001074' 026767  002500  002516  1$:     CMP     LSTCYL,HDRWD        ;CURRENT ADDRESS=LAST TRACK?
001102' 001012                          BNE     2$                  ;NO, CONTINUE
001104' 162767  000200  002506          SUB     #200,HDRWD          ;
001112' 052767  000100  002500          BIS     #100,HDRWD          ;
001120' 012767  000221  002474          MOV     #221,DIFWD          ;DIF WD 1, MARKER, SEEK OUT, HS=1
                                                                    ;SET CURRENT HD=1, SEEK OUT
001126' 000422                          BR      TAG2
001130' 032767  000004  002464  2$:     BIT     #4,DIFWD            ;SN SET IN DIF WORD
001136' 001404                          BEQ     3$                  ;NO, 3$
001140' 062767  000200  002452          ADD     #200,HDRWD          ;YES, CYL WILL INCREMENT
001146' 000403                          BR      4$                  ;SKIP OVER
001150' 162767  000200  002442  3$:     SUB     #200,HDRWD          ;NO, CYL WILL DECREMENT
001156' 032767  000020  002436  4$:     BIT     #20,DIFWD           ;HEAD SET?
001164' 001403                          BEQ     TAG2                ;NO, LEAVE EXPECTED ALONE
001166' 052767  000100  002424          BIS     #100,HDRWD          ;YES, SET HEAD SELECT BIT
001174' 004567  000740          TAG2:   JSR     R5,SEEK             ;PERFORM SEEK
001200' 004767  000266                  JSR     PC,WTRDY            ;WAIT FOR SEEK TO FINISH
001204' 004567  000754                  JSR     R5,RDHDR            ;READ HEADER VERIFY CORRECT
                                                                    ;SEEK
001210' 016767  002340  003502          MOV     T.MP,CURADR         ;READ HEADER
001216' 042767  000077  003474          BIC     #77,CURADR          ;CLEAR OUT SECTOR BITS
001224' 026767  003470  002366          CMP     CURADR,HDRWD        ;WAS SEEK CORRECT?
001232' 001425                          BEQ     6$                  ;YES, CONTINUE
                                                                    ;NO REPORT ERROR
001234' 016767  002270  176636          MOV     RLCS,CSRA
001242' 017767  002262  176632          MOV     @RLCS,ACSR
001250' 017767  002254  176626          MOV     @RLCS,ASTAT
001256' 104403  000000' 005466'         MSGN$,BEGIN,BDSEEK          ;ASCII MESSAGE CALL WITH COMMON HEADER

001264' 012767  000051  176614          MOV     #51,ERRTYP          ;BAD SEEK
                                        ;***********************************************************
001272' 104405  000000' 005476'         HRDER$,BEGIN,TABLE          ;SEEK WAS BAD
                                        ;***********************************************************

001300' 016767  003414  002312          MOV     CURADR,HDRWD        ;MAKE MISTAKE NEW HDRWD

001306' 026767  002306  002266  6$:     CMP     HDRWD,LSTTRK        ;ARE WE ON LAST TRACK
001314' 001002                          BNE     7$                  ;NO, CONTINUE
001316' 000167  177372                  JMP     TAG                 ;YES, GO GET ANOTHER CAUSE ITS THE BAD SECTOR TRACK

001322' 016767  176604  002302  7$:     MOV     RBUFSZ,WCNT2        ;GET BUFFER SIZE (READ)
001330' 005467  002276                  NEG     WCNT2               ;NEGATE FOR RLMP
001334' 104414  000000'                 GWBUF$, BEGIN               ;GET WRITE BUFFER INFORMATION
001340' 016767  176576  002262          MOV     WBUFSZ,WCNT1        ;GET BUFFER SIZE (WRITE)
001346' 005467  002256                  NEG     WCNT1               ;NEGATE FOR RLMP
001352' 004567  000502                  JSR     R5,WRITE            ;WRITE DATA
001356' 005767  002202                  TST     RWER                ; CONTROLLER ERROR ??
001362' 001017                          BNE     5$                  ; SKIP READS IF SO.
001364' 004567  000460                  JSR     R5,WRCHK
```

```
                                        TST     RWER            ; ERROR ??
                                        BNE     5$              ; SKIP READS IF SO.
                                        JSR     R5,READ         ;READ DATA
                                        TST     RWER            ; ERROR ??
                                        BNE     5$              ; SKIP DATA CHECK IF SO.
                                        CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
                                        5$                      ; IF ERROR, RETURN AT TAG 5$
                                        BR      4$              ; BR IF NO DATA ERRORS.                 ;GP-3
                                5$:     BIT     #BIT0,SR1       ; DROP ON ERROR ??                      ;GP-3
                                        BEQ     4$              ; NO                                    ;GP-3
                                        JSR     R5,DROP         ; YES, DROP IT.                         ;GP-3
                                        JMP     LOOPL           ;...AND TRY ANOTHER.                    ;GP-3

                                4$:     CLR     RWER
                                        INC     CNT
                                        CMP     #16.,CNT
                                        BEQ     3$
                                        JMP     LOOP            ; REITERATE 16 TIMES/DRIVE...
                                3$:     JMP     LOOPL           ;...THEN GET ANOTHER DRIVE.
                                ;
                                ;WAIT FOR DRIVE READY.  DROP DRIVE IF IT NEVFR COMES UP.
                                ;
                                WTRDY:  BIC     #1400,@RLCS
                                        BIS     DRIVE,@RLCS
                                        MOV     #77777,CLK      ;SET UP TIMEOUT
                                1$:     BIT     #1,@RLCS        ;DRIVE READY?
                                        BEQ     2$              ; NOT YET                               ;GP-3
                                        RTS     PC              ; YES, RETURN TO CALLER.                ;GP-3
                                2$:
                                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        DEC     CLK             ;CHECK TIMEOUT
                                        BNE     1$              ; LOOP TIL TIMER EXPIRES.
                                        MOV     #6,ERRTYP       ;DRIVE NOT READY
                                        ;*******************************************************************
                                        HRDER$,BEGIN,RLCS       ;DRIVE NOT READY
                                        ;*******************************************************************
                                        JSR     R5,DROP         ;CLEAR DRIVE (DROP) FROM LIST.
                                        TST     (SP)+           ; FIX THE STACK...
                                        JMP     LOOPL           ;...AND GO FOR ANOTHER.
                                ;
                                ;DROP CURRENT DRIVE.  DROP MODULE IF NO DRIVES LEFT.
                                ;
                                DROP:
                                        MSGN$,BEGIN,DROPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        BIC     DRVMSK,DVICE    ;CLEAR THIS DRIVE BIT.
                                        BEQ     1$              ; DROP MODULE IF NO MORE.               ;GP-3
                                        RTS     R5              ; OTHERWISE, RETURN TO CALLER.          ;GP-3
                                1$:
                                        MSGN$,BEGIN,NOLEFT      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        TST     (SP)+           ; FIX STACK.
                                FINI:
                                        END$,BEGIN              ;DROP THE MODULE
                                ;
                                ;ROUTINE TO SET UP RL11 REGISTERS, VECTOR AND BR LEVEL
                                ;
```

```
001626' 016700 176154           SETUP:  MOV     ADDR,R0         ;GET BASE ADDRESS
001632' 010067 001672                   MOV     R0,RLCS         ;CONTROL REGISTER
001636' 005720                          TST     (R0)+           ;INCREMENT FOR NEXT
001640' 010067 001666                   MOV     R0,RLBA         ;BUS ADDRESS
001644' 005720                          TST     (R0)+           ;INCREMENT FOR NEXT
001646' 010067 001662                   MOV     R0,RLDA         ;DISK ADDRESS
001652' 005720                          TST     (R0)+           ;INCREMENT FOR NEXT
001654' 010067 001656                   MOV     R0,RLMP         ;DATA BUFFER
001660' 032767 000010 176130            BIT     #BIT3,SR1       ;TEST IF AN RLV12 CONTROLER        ;GP-4
001666' 001411                          BEQ     1$              ;BR IF NOT                         ;GP-4
001670' 005720                          TST     (R0)+           ;INCREMENT FOR NEXT                ;GP-4
001672' 010067 001642                   MOV     R0,RLBAE        ;BUS EXTENDED ADDRESS REGISTER     ;GP-4
001676' 012767 003556' 003602           MOV     #T.BAE,TABLEY   ;FIX ERROR REPORT                  ;GP-4
001704' 012767 003560' 003576           MOV     #T.STAT,TABLEZ  ; FOR RLV12 CONTROLLER             ;GP-4
001712' 016700 176072           1$:     MOV     VECTOR,R0       ;GET VECTOR ADDRESS
001716' 012720 000252'                  MOV     #START,(R0)+    ;SET POINTER
001722' 116710 176064                   MOVB    BR1,(R0)        ;SET PRIORITY
001726' 000207                          RTS     PC              ;RETURN
                                ;
                                ; SUBROUTINE TO SET 18 OR 22 BIT BUS ADDRESS IN CONTROLLER.
                                ; THIS ENTIRE SUBROUTINE IS NEW.                                   ;GP-4
                                ;
001730' 016767 176200 000102    SETWBA: MOV     WBUFPA,PA18     ; GET 18 BIT WRITE BUFFER ADDRESS.
001736' 016767 176174 000076            MOV     WBUFEA,EA18
001744' 000406                          BR      SETCMN
001746' 016767 176154 000064    SETRBA: MOV     RBUFPA,PA18     ; GET 18 BIT READ READ BUFFER ADDRESS.
001754' 016767 176150 000060            MOV     RBUFEA,EA18
001762' 016777 000052 001542    SETCMN: MOV     PA18,@RLBA      ; SET BA<15:0>...
001770' 042777 000060 001532            BIC     #60,@RLCS
001776' 056777 000040 001524            BIS     EA18,@RLCS      ;...AND BA<17:16> IN CSR<5:4>.
002004' 032767 000010 176004            BIT     #BIT3,SR1       ; ARE WE RLV12 ??
002012' 001411                          BEQ     1$              ; BR IF NOT.
002014' 104416 000000' 002040'          MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
002022' 016777 000016 001502            MOV     PA22,@RLBA      ; LOAD BA<15:0>...
002030' 016777 000012 001502            MOV     EA22,@RLBAE     ;...AND BA<21:16>.
002036' 000207                  1$:     RTS     PC

002040' 000000                  PA18:   0               ; 18 BIT BA<15:0>...
002042' 000000                  EA18:   0               ;...AND EA<17:16> IN BITS <5:4>.
002044' 000000                  PA22:   0               ; 22 BIT BA<15:0>...
002046' 000000                  EA22:   0               ;...AND EA<21:16> IN BITS <5:0>.
                                ;
                                ;DRIVERS (INTERRUPT)
                                ;
002050' 012767 000102 001540    WRCHK:  MOV     #102,FUNC       ; WRITE-CHECK...
002056' 000403                          403                     ;...OR...
002060' 012767 000112 001530    WRITE:  MOV     #112,FUNC       ;...WRITE FUNCTION.
002066' 016777 001536 001442            MOV     WCNT1,@RLMP     ; WORD COUNT
002074' 016777 001520 001432            MOV     HDRWD,@RLDA     ; DISK ADDRESS
002102' 004767 177622                   JSR     PC,SETWBA       ; WRITE BUFFER ADDRESS             ;GP-4
002106' 000444                          BR      EXEC
002110' 012767 000114 001500    READ:   MOV     #114,FUNC       ; READ FUNCTION
002116' 016777 001510 001412            MOV     WCNT2,@RLMP     ; WORD COUNT
002124' 016777 001470 001402            MOV     HDRWD,@RLDA     ; DISK ADDRESS
002132' 004767 177610                   JSR     PC,SETRBA       ; READ BUFFER ADDRESS              ;GP-4
002136' 000430                          BR      EXEC
```

```
                                SEEK:   MOV     #106,FUNC       ;SEEK FUNCTION
                                        MOV     DIFWD,@RLDA     ;DIFFERENCE WORD
                                        BIS     #1,@RLDA        ;SET MARKER BIT
                                        BR      EXEC
                                RDHDR:  MOV     #110,FUNC       ;READ HEADER FUNCTION
                                        BR      EXEC
                                GSTAT:  MOV     #3,@RLDA        ;GET STATUS...
                                        403                     ;...OR...
                                DRVRTS: MOV     #13,@RLDA       ;...RESET AND GET STATUS.
                                        MOV     #104,FUNC       ;GET STATUS FUNCTION.

                                EXEC:   BIC     #1416,@RLCS     ; CLEAR THE OLD...                      ;GP-4
                                        BIS     DRIVE,@RLCS     ;...AND INSERT NEW DRIVE...             ;GP-4
                                        BIS     FUNC,@RLCS      ;...AND OPCODE BITS.                    ;GP-4
                                        MOV     #INTSRV,@VECTOR ;SET UP INTERRUPT VECTOR...             ;GP-4
                                        BIC     #200,@RLCS      ;...AND EXECUTE.                        ;GP-4
                                        EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                                ;
                                ; CONTINUE HERE ON RL INTERRUPT.
                                ;
                                INTSRV:
                                        ;-------------------------------------------------------------
                                        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-------------------------------------------------------------
                                1$:     CLR     T.STAT
                                        MOV     RLCS,CSRA       ;LOAD ADDR OF CSR
                                        MOV     @RLCS,ACSR      ;LOAD CONTENTS OF CSR
                                        MOV     ACSR,T.CS
                                        MOV     @RLBA,T.BA
                                        MOV     @RLDA,T.DA
                                        MOV     @RLMP,T.MP
                                        BIT     #BIT3,SR1       ;TEST IF RLV12 CONTROLLER               ;GP-4
                                        BEQ     20$             ;BR IF NOT                              ;GP-4
                                        MOV     @RLBAE,T.BAE    ;GET EA BITS                            ;GP-4
                                20$:    TST     T.CS            ;ANY ERRORS
                                        BMI     11$             ;YES, CONTINUE TO CHECK
                                        CLR     RETRY
                                        RTS     R5              ;NO, RETURN CALL+4 SKIP RETRY

                                11$:    INC     RWER
                                        MOV     #NULLX,HTYPE    ;SETUP FOR NULL PRINT
                                        BIT     #BIT14,T.CS     ;DRIVE ERROR
                                        BEQ     2$              ;NO,BRANCH
                                        MOV     #3,@RLDA        ;GET STATUS
                                        MOV     #4,TMP
                                        BIS     DRIVE,TMP
                                        MOV     TMP,@RLCS
                                99$:
                                        BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                                        BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                        BIT     #200,@RLCS
                                        BEQ     99$
                                        MOV     @RLMP,T.STAT
                                        MSGN$,BEGIN,DRVERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        MOV     #6,ERRTYP       ;DRIVE ERROR
                                        ;*************************************************************
```

```
                                HRDER$,BEGIN,TABLE      ;
                                ;*****************************************************************
                                MOV     #13,@RLDA
                                MOV     TMP,@RLCS
                        98$:
                                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
                                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                BIT     #200,@RLCS
                                BEQ     98$
                                BR      NORPT
                        2$:     BIT     #BIT13,T.CS     ;NXM SET
                                BEQ     21$
                                MOV     #NXM,HTYPE
                                BR      HRDRPT
                        21$:    BIT     #BIT10,T.CS     ;OPI SET
                                BEQ     4$              ;NO, CHECK DCRC,DCK
                                MOV     #OPI,HTYPE      ;INITIAL SET FOR OPI
                                BIT     #BIT11,T.CS     ;HCRC?
                                BEQ     3$              ;NO, BRANCH
                                MOV     #HCRC,HTYPE     ;HCRC ERROR
                                BR      FNDBSC          ;FIND BAD SECTOR
                        3$:     BIT     #BIT12,T.CS     ;HNF
                                BEQ     HRDRPT          ; NO, REPORT AS REPORT 'OPI''           ;GP-1
                                MOV     #HNF,HTYPE      ;HNF ERROR
                                BR      FNDBSC          ;GO CHECK BAD SECTOR FILE
                        4$:     BIT     #BIT11,T.CS     ;DCK?
                                BEQ     5$              ;NO,MUST BE DLT
                                DEC     T.DA            ;BACK UP TO SECTOR THAT WAS BAD
                                MOV     #DCK,HTYPE      ;DCK ERROR
                                BR      FNDBSC          ;GO CHECK BAD SECTOR FILE
                        5$:     MOV     #DLT,HTYPE      ;SETUP DLT ERROR

                        RPTERR: BIT     #BIT2,SR1       ;PRINTING SOFTERRORS ??
                                BNE     55$             ;NO, SKIP PRINT
                                MSGN$,BEGIN,SOFT        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                MOV     #1,ERRTYP       ;DATA ERROR
                                ;*****************************************************************
                                SOFER$,BEGIN,TABLE      ;
                                ;*****************************************************************
                        55$:    CMP     RETRY,LIMIT     ;RETRY EXHAUSTED
                                BEQ     6$              ;YES, NO MORE RETRIES
                                INC     RETRY
                                SUB     #4,R5           ; ADJUST RETURN PC...
                                RTS     R5              ;...AND TRY AGAIN.

                        6$:     MOV     FUNC,R0         ; GET FUNCTION CODE...                  ;GP-2
                                BIC     #^C16,R0        ;...STRIP IT...                         ;GP-2
                                MOV     FNCLST(R0),EXCEED ;...AND GET APPORPRIATE TEXT.         ;GP-2
                                MOV     HTYPE,TER1      ;GET ERROR TYPE
                                MSGN$,BEGIN,HARD        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                MSGN$,BEGIN,EXCEED      ;ASCII MESSAGE CALL WITH COMMON HEADER
                        NORPT:  CLR     RETRY
                                RTS     R5
                        HRDRPT: MOV     HTYPE,TER1
                                MSGN$,BEGIN,HARD        ;ASCII MESSAGE CALL WITH COMMON HEADER
                                BR      NORPT
```

```
                                        ;
                                        ;ERROR WAS HCRC OR HNF OR DCK, POSITION TO LAST TRACK AND RECOVER
                                        ;BAD SECTOR FILES. IF DCK/HNF CHECK WHOLE DA, IF HCRC CHECK IF
                                        ;WE WERE DOING A RDHDR IF READ HDR THEN CHECK ONLY TRACK AND
                                        ;CYLINDER.
                                        ;
003042' 016701  000532          FNDBSC: MOV     LSTCYL,R1       ;LAST TRACK
003046' 016700  000546                  MOV     HDRWD,R0        ;PRESENT POSITION
003052' 042700  000100                  BIC     #100,R0         ;CLEAR OUT HEAD
003056' 160001                          SUB     R0,R1           ;CALC SEEK DIFFERENCE
003060' 010177  000450                  MOV     R1,@RLDA        ;LOAD SEEK DIFFERENCE
003064' 052777  000025  000442          BIS     #25,@RLDA       ;SET HEAD 1, SEEK IN
003072' 016767  000526  000462          MOV     DRIVE,MFLG      ;SELECT DRIVE. (MFLG UESD)
003100' 052767  000006  000454          BIS     #6,MFLG         ;SET UP SEEK
003106' 016777  000450  000414          MOV     MFLG,@RLCS      ;SEEK
003114' 004767  176352                  JSR     PC,WTRDY        ;WAIT FOR SEEK TO FINISH
                                        ;
                                        ;NOW SITTING ON LAST TRACK, RECOVER BAD SECTOR FILES AND COMPARE
                                        ;
003120' 016700  000456                  MOV     LSTTRK,R0       ;STARTING SECTOR 0
003124' 005067  000432                  CLR     MFLG            ;SWITCH TO TELL US MANUF OR FIELD FILE
003130' 005067  000432                  CLR     FND             ;FLAG TO INDICATE HEADER FOUND IN LIST
003134' 010077  000374          2$:     MOV     R0,@RLDA        ;LOAD SECTOR TO READ
003140' 012777  177400  000370          MOV     #-256.,@RLMP    ;TWO SECTOR READ
003146' 012777  000215  000354          MOV     #215,@RLCS      ; READ COMMAND                  ;GP-4
003154' 004767  176566                  JSR     PC,SETRBA       ; BUFFER ADDRESS.               ;GP-4
003160' 056777  000440  000342          BIS     DRIVE,@RLCS     ; INSERT DRIVE                  ;GP-4
003166' 042777  000200  000334          BIC     #200,@RLCS      ; EXECUTE IT.
003174' 004767  176272                  JSR     PC,WTRDY        ;WAIT FOR DRIVE
003200' 005777  000324                  TST     @RLCS           ;READ SUCCESSFUL??
003204' 100023                          BPL     4$              ;YES, GO CHECK FOR SECTOR
                                        ;
003206' 062700  000004                  ADD     #4,R0           ;NO, NEXT SECTOR
003212' 005767  000344                  TST     MFLG            ;WHICH WE READING, MANUF OR FIELD
003216' 001012                          BNE     3$              ;FIELD COMPARE AGAINST 77750
003220' 026700  000346                  CMP     FSTBSC,R0       ;MANUFACTURING, AT END
003224' 001343                          BNE     2$              ;NO, GO BACK AND READ NEXT
                                        ;
003226'                         99$:
003226' 104403  000000' 005446'         MSGN$,BEGIN,NOSEC       ;ASCII MESSAGE CALL WITH COMMON HEADER
003234' 004567  176332                  JSR     R5,DROP
003240' 000167  175200                  JMP     LOOPL
                                        ;
003244' 026700  000324          3$:     CMP     LSTBSC,R0       ;AT END OF FIELD BAD
003250' 001331                          BNE     2$              ;NO, GO BACK
003252' 000765                          BR      99$             ;YES GO DROP DRIVE
                                        ;
003254' 016701  174644          4$:     MOV     RBUFVA,R1       ;GET WHERE WE READ
003260' 062701  000010                  ADD     #10,R1          ;SKIP PAST I.D. ETC.....
003264' 012702  000176                  MOV     #126.,R2        ;ONLY 126 ENTRIES
003270' 012103                  44$:    MOV     (R1)+,R3        ;GET CYLINDER
003272' 100437                          BMI     88$             ;MINUS WE'RE DONE
003274' 012104                          MOV     (R1)+,R4        ;GET TRACK AND SECTOR
003276' 000303                          SWAB    R3              ;ALIGN PROPERLY
003300' 006303                          ASL     R3
003302' 150403                          BISB    R4,R3
```

```
003304' 032704  000400                    BIT     #400,R4
003310' 001402                            BEQ     5$
003312' 052703  000100                    BIS     #100,R3
003316' 022767  005371' 002134   5$:      CMP     #HCRC,HTYPE        ;IS ERROR HCRC?
003324' 001012                            BNE     6$                 ;NO, GO LOOK FOR BAD SECTOR
003326' 022767  000110  000262            CMP     #110,FUNC          ;WE'RE WE DOING READ HEADER
003334' 001006                            BNE     6$                 ;NO, GO LOOK FOR BAD SECTOR
003336' 042703  000077                    BIC     #77,R3             ;YES, CLEAR SCETOR BITS
003342' 020367  000252                    CMP     R3,HDRWD           ;BAD SECTOR
003346' 001404                            BEQ     7$
003350' 000406                            BR      8$
003352' 020367  000174           6$:      CMP     R3,T.DA            ;IS THIS ONE IT???????
003356' 001003                            BNE     8$                 ;NO
003360' 005267  000202           7$:      INC     FND
003364' 000412                            BR      9$
003366' 005302                   8$:      DEC     R2                 ;CHECKED WHOLE FILE
003370' 001337                            BNE     44$                ;NO
003372' 005767  000164           88$:     TST     MFLG               ;WHICH WE DOING
003376' 001005                            BNE     9$                 ;FIELD , WE'RE DONE
003400' 005267  000156                    INC     MFLG               ;MANUFACT. THEN SET UP FIELD
003404' 016700  000162                    MOV     FSTBSC,R0
003410' 000651                            BR      2$

003412' 016700  000202           9$:      MOV     HDRWD,R0
003416' 016701  000156                    MOV     LSTCYL,R1
003422' 042700  000100                    BIC     #100,R0
003426' 160001                            SUB     R0,R1
003430' 010177  000100                    MOV     R1,@RLDA
003434' 052777  000001  000072            BIS     #1,@RLDA
003442' 032767  000100  000150            BIT     #100,HDRWD
003450' 001403                            BEQ     10$
003452' 052777  000020  000054            BIS     #20,@RLDA
003460' 016767  000140  000074   10$:     MOV     DRIVE,MFLG
003466' 052767  000006  000066            BIS     #6,MFLG
003474' 016777  000062  000026            MOV     MFLG,@RLCS         ; SEEK.
003502' 004767  175764                    JSR     PC,WTRDY
003506' 005767  000054                    TST     FND
003512' 001002                            BNE     11$
003514' 000167  177162                    JMP     RPTERR
003520' 000167  177272           11$:     JMP     NORPT
                                 ;
                                 ;LOCATIONS USED BY MODULE
                                 ;
003524' 000000                   RETRY:   .WORD   0
003526' 000003                   LIMIT:   .WORD   3
003530' 000000                   RLCS:    .WORD   0
003532' 000000                   RLBA:    .WORD   0
003534' 000000                   RLDA:    .WORD   0
003536' 000000                   RLMP:    .WORD   0
003540' 177777                   RLBAE:   .WORD   177777             ; RLV12 BAE              ;GP-4
003542' 177777                            .WORD   177777             ; TERMINATOR.
003544' 000000                   TMP:     .WORD   0
003546' 000000                   T.CS:    .WORD   0
003550' 000000                   T.BA:    .WORD   0
003552' 000000                   T.DA:    .WORD   0
003554' 000000                   T.MP:    .WORD   0
```

```
                                T.BAE:  .WORD   0               ; RLV12 BAE                             ;GP-4
                                T.STAT: .WORD   0
                                MFLG:   .WORD   0
                                RWER:   .WORD   0
                                FND:    .WORD   0
                                DRVMSK: .WORD   0
                                FSTBSC: .WORD   0
                                LSTBSC: .WORD   0
                                MASK:   .WORD   0
                                LSTCYL: .WORD   0
                                LSTTRK: .WORD   0
                                MASK2:  .WORD   0
                                CLK:    .WORD   0
                                CNT:    .WORD   0
                                MULDRV: .WORD   0
                                DLTCNT: .WORD   0               ;NUMBER OF DATA LATE ERRORS
                                FUNC:   .WORD   0               ;FUNCTION TO BE PERFORMED
                                HDRWD:  .WORD   0               ;HEADER WORD (RDHDR, R/W)
                                DIFWD:  .WORD   0               ;DIFFERENCE WORD (SEEK)
                                DRIVE:  .WORD   0               ;DRIVE UNDER TEST (BITS 8,9)
                                DVICE:  .WORD   0               ;WORKING 'DVID1"
                                WCNT1:  .WORD   0               ;WORD COUNT (WRITE)
                                WCNT2:  .WORD   0               ;WORD COUNT (READ)
                                BUFIN:  .BLKW   256.
                                BSECBF: .BLKW   25.             ;BAD SECTOR LIST
                                CURMSG: .WORD   0
                                CURADR: .WORD   0
                                NXTADR: .WORD   0
                                NUMB:   .WORD   0
                                FNCLST: NULLX                   ; OPCODE 0 UNUSED.                      ;GP-2
                                        MES8A                   ; 1 = WRT CHK.                          ;GP-2
                                        MES9                    ; 2 = RESET OR GET STATUS.
                                        MES6                    ; 3 = SEEK.
                                        MES5                    ; 4 = READ HEADER.
                                        MES8                    ; 5 = WRITE.
                                        MES7                    ; 6 = READ.
                                        NULLX                   ; 7 = READ NO-HEADER (UNUSED).          ;GP-2

004746' 047516 042040 044522    MES1:   .ASCIZ  'NO DRIVES PRESENT %'
004754' 042526 020123 051120
004762' 051505 047105 020124
004770' 000045
004772' 047516 042040 044522    MES2:   .ASCIZ  'NO DRIVES LEFT %'
005000' 042526 020123 042514
005006' 052106 022440 000
005013' 104 044522 040526       MES3:   .ASCIZ  'DRIVE '
005020' 000040
005022' 042040 047522 050120    MES4:   .ASCIZ  ' DROPPED %'
005030' 042105 022440 000
005035' 122 040505 020104       MES5:   .ASCIZ  'READ HEADER'
005042' 042510 042101 051105
005050' 000
005051' 123 042505 000113       MES6:   .ASCIZ  'SEEK'
005056' 042522 042101 000       MES7:   .ASCIZ  'READ'
005063' 127 044522 042524       MES8:   .ASCIZ  'WRITE'
005070' 000
```

```
764  005071'     127  044522  042524  MES8A:  .ASCIZ  'WRITE-CHECK'                    ;GP-2
765  005076' 041455  042510  045503
766  005104'     000
767  005105'     104  044522  042526  MES9:   .ASCIZ  'DRIVE RESET'
768  005112' 051040  051505  052105
769  005120'     000
770  005121'     040  042522  051124  MES10:  .ASCIZ  ' RETRY LIMIT EXCEEDED%'
771  005126' 020131  044514  044515
772  005134' 020124  054105  042503
773  005142' 042105  042105  000045
774  005150' 051445  042505  020113  MES11:  .ASCIZ  '%SEEK TO WRONG CYLINDER%'
775  005156' 047524  053440  047522
776  005164' 043516  041440  046131
777  005172' 047111  042504  022522
778  005200'     000
779  005201'     104  052101  020101  MES12:  .ASCIZ  'DATA LATE%'
780  005206' 040514  042524  000045
781  005214' 051104  053111  020105  MES13:  .ASCIZ  'DRIVE ERROR%'
782  005222' 051105  047522  022522
783  005230'     000
784  005231'     123  043117  020124  MES14:  .ASCIZ  'SOFT ERROR   '
785  005236' 051105  047522  020122
786  005244' 020040      000
787  005247'     122  054114  046040  MES15:  .ASCIZ  'RLX LOAD UNIT DROPPED'
788  005254' 040517  020104  047125
789  005262' 052111  042040  047522
790  005270' 050120  042105      000
791  005275'     110  051101  020104  MES16:  .ASCIZ  'HARD ERROR   '
792  005302' 051105  047522  020122
793  005310' 020040      000
794  005313'     103  047101  020124  NBDSC:  .ASCIZ  'CANT RECOVER BAD SECTOR FILE%'
795  005320' 042522  047503  042526
796  005326' 020122  040502  020104
797  005334' 042523  052103  051117
798  005342' 043040  046111  022505
799  005350'     000
800  005351'     117  044520      000  OPI:    .ASCIZ  'OPI'
801  005355'     104  052114      000  DLT:    .ASCIZ  'DLT'
802  005361'     104  045503      000  DCK:    .ASCIZ  'DCK'
803  005365'     110  043116      000  HNF:    .ASCIZ  'HNF'
804  005371'     110  051103  000103  HCRC:   .ASCIZ  'HCRC'
805  005376' 054116  000115           NXM:    .ASCIZ  'NXM'
806  005402' 000040                  NULLX:  .ASCIZ  ' '
807  005404' 000045                  CR:     .ASCIZ  '%'
808                                          .EVEN
809
810  005406' 005013'                 DROPMS: MES3
811  005410' 004724'                         NUMB
812  005412' 005022'                         MES4
813  005414' 177777                          177777
814
815  005416' 004772'                 NCLEFT: MES2
816  005420' 177777                          177777
817
818  005422' 004746'                 ABORT:  MES1
819  005424' 177777                          177777
```

```
        005426' 000000          EXCEED: .WORD   0
        005430' 005121'                 MES10
        005432' 005404'                 CR
        005434' 177777                  177777

        005436' 005275'         HARD:   MES16
        005440' 000000          TER1:   .WORD   0
        005442' 005404'                 CR
        005444' 177777                  177777

        005446' 005313'         NOSEC:  NBDSC
        005450' 177777                  177777

        005452' 005214'         DRVERR: MES13
        005454' 177777                  177777

        005456' 005231'         SOFT:   MES14
        005460' 000000          HTYPE:  .WORD   0
        005462' 005404'                 CR
        005464' 177777                  177777

        005466' 005150'         BDSEEK: MES11
        005470' 177777                  177777

        005472' 005247'         DROPLD: MES15
        005474' 177777                  177777
                                ;
                                ;REGISTERS OF RL11
                                ;
        005476' 003546'         TABLE:  .WORD   T.CS            ;CONTROL AND STATUS REGISTER
        005500' 003550'                 .WORD   T.BA            ;BUS ADDRESS REGISTER
        005502' 003552'                 .WORD   T.DA            ;DISK ADDRESS REGISTER
        005504' 003554'                 .WORD   T.MP            ;DISK DATA BUFFER ADDRESS
        005506' 003560'         TABLEY: .WORD   T.STAT          ;T.BAE  ;HAS STATUS ON DRIVE ERROR      ;GP-4
        005510' 177777          TABLEZ: .WORD   177777          ;T.STAT ;TERMINATOR                     ;GP-4
        005512' 177777                  .WORD   177777          ;TERMINATOR

                000001                  .END
```

```
ABORT   005422R          244     818#
ACSR    000102R          191#    348*    510*    511
ADDR    000006R          157#    428
ADDR22= 00100C           220#
ASR     000106R          195#
ASTAT   000104R          193#    349*
AWAS    000110R          196#
BDSEEK  005466R          350     842#
BEGIN   000000R          154#    241     244     250     259     298     350     354     365     377     401     402     407
                         416     421     424     460     500     506     532     533     537     540     545     546     574
                         577     589     590     594     635
BIT0  = 000001           220#    380
BIT1  = 000002           220#    295
BIT10 = 002000           220#    554
BIT11 = 004000           220#    557     565
BIT12 = 010000           220#    561
BIT13 = 020000           220#    550
BIT14 = 040000           220#    525
BIT15 = 100000           220#
BIT2  = 000004           220#    572
BIT3  = 000010           220#    436     458     515
BIT4  = 000020           220#
BIT5  = 000040           220#
BIT6  = 000100           220#
BIT7  = 000200           220#
BIT8  = 000400           220#
BIT9  = 001000           220#
BREAK$= 104407           220#    401     402     532     533     545     546
BR1     000012R          159#    444
BR2     000013R          160#
BSECBF  004634R          732#
BTOD$ = 104421           220#
BUFIN   003634R          202     731#
CDATA$= 104412           220#    377
CDERCT  000144R          210#
CDWDCT  000146R          211#
CHKDRV  000502R          258#
CHKDR1  000506R          251     254     261#
CLK     003606R          396*    403*    720#
CNT     003610R          246     268*    386*    387     721#
CONFIG  000056R          179#
CR      005404R          807#    823     828     839
CSRA    000100R          189#    347*    509*
CURADR  004720R          342*    343*    344     357     734#
CURMSG  004716R          733#
DATCK$= 104411           220#
DATER$= 104404           220#
DCK     005361R          568     802#
DIFWD   003622R          288*    302*    303*    305*    307*    308*    311*    320*    327*    330     335     485     726#
DLT     005355R          570     801#
DLTCNT  003614R          227*    723#
DRIVE   003624R          225*    257*    262*    395     496     529     608     622     684     727#
DROP    001572R          382     409     415#    636
DROPLD  005472R          241     845#
DROPMS  005406R          416     810#
DRVERR  005452R          537     834#
```

```
DRVMSK  003570R          226* 252* 253  255* 264  417  713#
DRVRTS  002204R          273  492#
DVICE   003626R          229* 238  240* 242  264  417* 728#
DVID1   000014R          161# 229
EA18    002042R          451* 454* 457  466#
EA22    002046R          462  468#
ENDIT$= 104413           220# 259
END$  = 104410           220# 424
ERRTYP  000106R          194# 352* 405* 538* 575*
EXCEED  005426R          587* 590  821#
EXEC    002220R          478  483  487  489  495#
EXIT$ = 104400           220# 500
FINI    001622R          245  423#
FNCLST  004726R          587  737#
FND     003566R          617* 665* 688  712#
FNDBSC  003042R          560  564  569  602#
FREE    000150R          212#
FSTBSC  003572R          275* 287* 631  672  714#
FUNC    003616R          472* 474* 479* 484* 488* 493* 497  585  657  724#
GETPA$= 104415           220# 250
GSTAT   002174R          490#
GWBUF$= 104414           220# 365
HARD    005436R          589  594  826#
HCRC    005371R          559  655  804#
HDRWD   003620R          290* 297* 303  312* 313* 316* 317  319* 323  325* 326* 332* 334*
                         337* 344  357* 359  476  481  603  660  675  681  725#
HNF     005365R          563  803#
HRDCNT  000044R          174#
HRDERS= 104405           220# 354  407  540
HRDPAS  000050R          176#
HRDRPT  003024R          553  562  593#
HTYPE   005460R          524* 552* 556* 559* 563* 568* 570* 588  593  655  838#
ICONT   000036R          171#
ICOUNT  000040R          172#
IDNUM   000122R          201#
IMODX.= 000000           213# 366
INIT    000030R          168#
INTR    000120R          200# 223*
INTSRV  002262R          498  504#
LIMIT   003526R          579  696#
LOOP    000702R          289# 389
LOOPL   000444R          252# 265  383  390  411  637
LSTBSC  003574R          276* 285* 639  715#
LSTCYL  003600R          278* 283* 323  602  676  717#
LSTTRK  003602R          279* 284* 359  615  718#
MAP22$= 104416           220# 460
MASK    003576R          277* 282* 301  716#
MASK2   003604R          274* 286* 313  719#
MES1    004746R          746# 818
MES10   005121R          770# 822
MES11   005150R          774# 842
MES12   005201R          779#
MES13   005214R          781# 834
MES14   005231R          784# 837
MES15   005247R          787# 845
MES16   005275R          791# 826
```

```
MES2      004772R          750#    815
MES3      005013R          753#    810
MES4      005022R          755#    812
MES5      005035R          741     757#
MES6      005051R          740     760#
MES7      005056R          743     761#
MES8      005063R          742     762#
MES8A     005071R          738     764#
MES9      005105R          739     767#
MFLG      003562R          608*    609*    610     616*    629     669     671*    684*    685*    686     710#
MODNAM    000000R          155#
MODSP     000252R          169     218#
MSGN$ =   104403           220#    241     244     350     416     421     537     574     589     590     594     635
MSGS$ =   104402           220#
MSG$  =   104401           220#
MULDRV    003612R          722#
NBDSC     005313R          794#    831
NOLEFT    005416R          421     815#
NORPT     003016R          549     591#    595     691
NOSEC     005446R          635     831#
NULL  =   000000           220#
NULLX     005402R          524     737     744     806#
NUMB      004724R          224*    256*    263*    736#    811
NXM       005376R          552     805#
NXTADR    004722R          735#
OPEN  =   000000           156     162     163     164     165     182     183     184     185     186     187     188     189
                           191     193     195     196     198     199     200     203     204     206     207     209     210
                           211     212     220#
OPI       005351R          556     800#
OTOA$ =   104420           220#
PASCNT    000034R          170#
PA18      002040R          450*    453*    455     460     465#
PA22      002044R          461     467#
PIRQ$ =   000004           220#    506
POPSP =   005726           220#
POPSP2=   022626           220#
PRTY  =   000000           220#
PRTY0 =   000000           160     220#
PRTY1 =   000040           220#
PRTY2 =   000100           220#
PRTY3 =   000140           220#
PRTY4 =   000200           220#
PRTY5 =   000240           159     220#
PRTY6 =   000300           220#
PRTY7 =   000340           220#
PS    =   177776           220#
PSW   =   177776           220#
PUSH  =   005746           220#
PUSH2 =   024646           220#
RAND$ =   104417           220#    298
RANNUM    000054R          178#    299
RBUFEA    000130R          204#    454
RBUFPA    000126R          203#    377     453
RBUFSZ    000132R          205#    363
RBUFVA    000124R          202#    250     643
RDHDR     002164R          289     340     488#
```

```
READ    002110R          374     479#
RESTRT  000424R          197     246#
RES1    000056R          180#
RES2    000060R          181#
RETRY   003524R          266*    520*    579     581*    591*    695#
RLBA    003532R          431*    455*    461*    512     698#
RLBAE   003540R          439*    462*    517     701#
RLCS    003530R          347     348     349     394*    395*    397     407     429*    456*    457*    495*    496*    497*
                         499*    509     510     530*    534     543*    547     610*    620*    622*    623*    625     686*
                         697#
RLDA    003534R          433*    476*    481*    485*    486*    490*    492*    513     527*    542*    606*    607*    618*
                         679*    680*    683*    699#
RLMP    003536R          435*    475*    480*    514     536     619*    700#
RPTERR  002702R          572#    690
RSTRT   000112R          197#
RSTRT1  000434R          243     247     249#
RWER    003564R          267*    369     372     375     385*    523*    711#
SBADR   000102R          190#
SEEK    002140R          338     484#
SETCMN  001762R          452     455#
SETRBA  001746R          453#    482     621
SETUP   001626R          228     428#
SETWBA  001730R          450#    477
SOFCNT  000042R          173#
SOFERS= 104406           220#    577
SOFPAS  000046R          175#
SOFT    005456R          574     837#
SPOINT  000032R          169#
SPSIZ = 000040             1#    213
SR1     000016R          162#    295     380     436     458     515     572
SR2     000020R          163#
SR3     000022R          164#
SR4     000024R          165#
START   000252R          168     221#    248     443
STAT    000026R          167#
SVR0    000062R          182#
SVR1    000064R          183#
SVR2    000066R          184#
SVR3    000070R          185#
SVR4    000072R          186#
SVR5    000074R          187#
SVR6    000076R          188#
SYSCNT  000052R          177#
TABLE   005476R          354     540     577     850#
TABLEY  005506R          440*    854#
TABLEZ  005510R          441*    855#
TAG     000714R          295#    361
TAG1    001040R          296     316#
TAG2    001174R          314     322     329     336     338#
TER1    005440R          588*    593*    827#
TMP     003544R          528*    529*    530     543     703#
TRPDFD= 000022           220#
T.BA    003550R          512*    705#    851
T.BAE   003556R          440     517*    708#
T.CS    003546R          511*    518     525     550     554     557     561     565     704#    850
T.DA    003552R          513*    567*    663     706#    852
```

```
T.MP    003554R          280     290     342     514*    707#    853
T.STAT  003560R          441     508*    536*    709#    854
VECTOR  000010R          158#    442     498*
WASADR  000104R          192#
WBUFEA  000136R          207#    451
WBUFPA  000134R          206#    450
WBUFRQ  000140R          208#
WBUFSZ  000142R          209#    366
WCNT1   003630R          366*    367*    475     729#
WCNT2   003632R          363*    364*    480     730#
WDFR    000116R          199#    221*
WDTO    000114R          198#    222*
WRCHK   002050R          371     472#
WRITE   002060R          368     474#
WTRDY   001472R          272     339     394#    611     624     687
XFLAG   000005R          156#
.     = 005514R          731#    732#
```

```
BKMOD           1#
BREAK           1#      400     531     544
BTOD            1#
CKDATA          1#      377
DATACK          1#
DATERR          1#
DFSEVN          1#      220
DSEVNT          1#      220
END             1#      423
ENDIT           1#      258
ENDMOD          1#
EQUATS          1#      220
EXIT            1#      500
GETPA           1#      250
GUBUFF          1#      365
HRDER           1#      353     406     539
IOMOD           1#
IOMODP          1#
IOMODR          1#
IOMODX          1#      148
MAP22           1#      460
MODULE          1#      149
MSG             1#
MSGN            1#      241     244     350     415     420     537     574     589     590     594     634
MSGS            1#
NBKMOD          1#
OTOA            1#
PIRQ            1#      504
RAND            1#      298
SBKMOD          1#
SOFER           1#      576

. ABS.  000000      000
        005514      001


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRLAF0,XRLAF0.SEQ/SOL/CRF/DOC=DDXCOM,XRLAF0
 RUN-TIME:  1  2  .5 SECONDS
 RUN-TIME RATIO:  30/4=6.6
 CORE USED:  10K  (20 PAGES)

 DOCUMENT PAGES:  22
```