```
                .REM    _
```

IDENTIFICATION

PRODUCT CODE: AC-E965C-MC

PRODUCT NAME: CXRLAC0 RL11/RL01 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RLA IS AN IOMODX THAT EXERCISES RL01 DISK DRIVES ON AN RL11 CONTROLLER. 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 RL DISK DRIVES WITH AN RL11 CONTROLLER.
STORAGE:: RLA REQUIRES:
1. DECIMAL WORDS: 1327
2. OCTAL WORDS: 02457
3. OCTAL BYTES: 5136

3. PASS DEFINITION

ONE PASS OF THE RLA MODULE CONSISTS OF 1024 CYCLES OF THE BASIC TEST SEQUENCE. (READ HEADER, SEEK, READ HEADER, WRITE, WRITE CHECK, READ). THE TEST SEQUENCE WRITES 1024 WORDS, READS 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:
DEVADP: 174400, VECTOR: 160, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:
NONE

6. DEVICE/OPTION SETUP

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

7. SR1 OPTIONAL SETUP

BIT 0 - DROP DRIVE ON ERROR

BIT 1 - RANDOM SEEKS

```
                        BIT 2 - DON'T PRINT SOFT ERRORS
```

```
                                        IOMODX  <RLAC >,174400,160,5,0,0,5,146,RUFIN,256.,1024.
                                        MODULE  150000,RLAC,174400,160,5,0,0,5,146,RUFIN,256.,1024.
                                        .TITLE  RLAC DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDFCOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046122  041501      040 MODNAM: .ASCII  /RLAC / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0       ;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' 000005                  ICONT:  5               ;# OF ITERATIONS PER PASS=5
000040' 000000                  ICOMNT: 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'                         SRADR:                  ;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' 000404'                 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' 093276'                 RBUFVA: BUFIN           ;READ BUFFER VIRTUAL ADDRESS
```



```
000126' 000000                  RBUFPA: OPEN            ;READ BUFFER PHYSICAL ADDRESS
000130' 000000                  RBUFEA: OPEN            ;READ BUFFER EA BITS
000132' 000400                  RBUFSZ: 256.            ;SIZE OF THE READ BUFFER
000134' 000000                  WBUFPA: OPEN            ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000                  WBUFEA: OPEN            ;WRITE BUFFER EA BITS
000140' 002000                  WBUFRQ: 1024.           ;WRITE BUFFER SIZE REQUESTED
000142' 000000                  WBUFSZ: OPEN            ;WRITE BUFFER SIZE AVAILABLE
000144' 000000                  CDERCT: OPEN            ;CDATA/DATCK ERROR COUNT
000146' 000000                  CDWDCT: OPEN            ;CDATA/DATCK WORD COUNT
000150' 000000                  FREE:   OPEN            ;RESERVED FOR FUTURE USE
        000040                          .REPT   SPSIZ   ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*****************************************************************************
000252' 012767  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  000005  177624          MOV     #5,INTR         ;5 INTERRUPTS/ITERATION
000274' 005067  002754                  CLR     DLYCNT          ;CLEAR DATA LATE COUNT
000300' 004767  001144                  JSR     PC,SETUP        ;GO SET UP REGISTERS
000304' 016767  177504  002756          MOV     DVID1,DVICE     ;COPY DRIVE SELECTION
000312' 122737  000014  000041  3$:     CMPB    #14,@#41        ;WAS RL LOAD DEVICE?
000320' 001020                          BNE     7$              ;N-BRANCH; Y-SEE IF LOAD UNIT SELECTED
000322' 012702  000001                  MOV     #1,R2           ;SET UP FOR MASK
000326' 113761  000040                  MOVB    @#40,R1         ;GET LOAD UNIT
000332' 001403                          BEQ     5$              ;IF ZERO GO MASK OUT UNIT
000334' 006302                  4$:     ASL     R2              ;SHIFT MASK
000336' 105301                          DECB    R1              ;DEC COUNT
000340' 001375                          BNE     4$              ;KEEP CHECKING
000342' 030267  002722          5$:     BIT     R2,DVICE        ;WAS THAT DRIVE SELECTED?
000346' 001405                          BEQ     7$              ;N-BRANCH; Y-CONTINUE
000350' 040267  002714                  BIC     R2,DVICE        ;DELETE UNIT FROM DEVICE MAP
000354' 104403  000000' 005116'         MSGNS,BEGIN,DROPLD      ;ASCII MESSAGE CALL WITH COMMON HEADER
000362' 005767  002702          7$:     TST     DVICE           ;ANY DRIVES SELECTED?
000366' 001005                          BNE     8$              ;YES, CONTINUE
000370' 104403  000000' 005046'         MSGNS,BEGIN,ABORT       ;ASCII MESSAGE CALL WITH COMMON HEADER
000376' 000167  000714                  JMP     FINI            ;MESSAGE, DROP MODULE
000402' 000404                  8$:     BR      RSTRT1          ;+ SUPPORT - DT03
000404' 005767  002640          RESTRT: TST     CNT             ;+    / SUPPORT
000410' 001001                          BNE     RSTRT1          ;+   /  FOR
000412' 000717                          BR      START           ;+  /   DT03
000414'                         RSTRT1:                         ;+/
000414' 104415  000000' 000124'         GETPAS,BEGIN,RBUFVA             ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000422' 005067  002624                  CLR     MULDRV          ;CLEAR MULTIPLE DRIVE INDICATOR
000426' 012767  000057  003732          MOV     #57,NUMB
000434' 012767  177400  002624          MOV     #-400,DRIVE     ;SET DRIVE SELECT
000442' 012767  000001  002574          MOV     #1,DRVMSK       ;SETUP DRIVE SELECT MASK
000450' 000402                          BR      CHKDRV
000452' 006367  002566          LOOPL:  ASL     DRVMSK          ;SHIFT MASK FOR NEXT DRIVE
000456'                         OLPSK:
000456'                         CHKDRV:
000456' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
```

```
000462- 062767  000400  002576  1$:     ADD     #400,DRIVE      ;NEXT DRIVE
000470- 005267  003672                   INC     NUMR
000474- 036767  002544  002566           BIT     DRVMSK,DVICE    ;IS THAT DRIVE PRESENT
000502- 001763                           BEQ     LOOPL           ;NO, GO FOR NEXT ONE
000504- 005067  002474                   CLR     RETRY           ;CLEAR A FEW LOCATIONS
000510- 005067  002524                   CLR     RWER            ;READ WRITE ERROR FLAG
000514- 005067  002530                   CLR     CNT             ;COUNT

                                         ;
                                         ;WE HAVE A DRIVE, START TESTING
                                         ;

000520- 004767  000576                   JSR     PC,WTRDY
000524- 004567  001152                   JSR     R5,DRVRTS       ;ISSUE DRIVE RESET, CLEAR VOLUME
000530- 012767  000201  002526           MOV     #201,DIFWD
000536- 004567  001112           LOOP:   JSR     R5,RDHDR        ;READ HEADER ON DISK
000542- 016767  002464  002512           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
                                         ;

000550- 032767  000002  177240   TAG:    BIT      #BIT1,SR1      ;INCREMENTAL OR RANDOM SEEKS?
000556- 001446                           BEQ     TAG1            ;INCREMENTAL, TAG1
000560- 042767  000177  002474           BIC     #177,HDRWD      ;CLEAR HEAD AND SECTOR BITS
000566- 104417  000000-                  RANDS,BEGIN
000572- 016700  177256                   MOV     RANNUM,R0       ;STORE IT AWAY
000576- 010001                           MOV     R0,R1           ;SAVE A COPY
000600- 042700  100177                   BIC     #100177,R0      ;CLEAR BIT 15, HEAD AND SECTOR
000604- 010067  002454                   MOV     R0,DIFWD        ;LET'S CALCULATE DIFFERENCE WORD
000610- 166767  002446  002446           SUB     HDRWD,DIFWD     ;GET DIFFERENCE TO SEEK
000616- 100003                           BPL     1$
000620- 005467  002440                   NEG     DIFWD           ;MAKE DIFF ABSOLUTE
000624- 000403                           BR      2$
000626- 052767  000004  002430   1$:     BIS     #4,DIFWD        ;SET DIRECTION BIT
000634- 052767  000001  002422   2$:     BIS     #1,DIFWD        ;SET MARKER
000642- 032701  000100                   BIT     #100,R1         ;TEST HEAD
000646- 001403                           BEQ     3$              ;IF 0, DON'T SET HEAD IN DIFF
000650- 052767  000020  002406           BIS     #20,DIFWD       ;SET HEAD
000656- 016167  002400           3$:     MOV     R1,HDRWD        ;GET EXPECTED HEADER
000662- 042767  100177  002372           BIC     #100177,HDRWD   ;CLEAR SECTOR BITS
000670- 000167  000126                   JMP     TAG2

000674- 042767  000177  002360   TAG1:   BIC     #177,HDRWD      ;CLEAR OUT SECTOR BITS & HEAD
000702- 032767  077600  002352           BIT     #77600,HDRWD    ;ON TRACK 0?
000710- 001007                           BNE     1$              ;NO, GO CHECK FOR CYLINDER 77600
000712- 012767  000200  002342           MOV     #200,HDRWD      ;SET NEXT ADDRESS=CYL 1
000720- 012767  000205  002336           MOV     #205,DIFWD      ;DIF WD 1, MARKER, SEEK IN, HS=0
                                                                 ;SET CURRENT HD=0, SEEK IN
000726- 000435                           BR      TAG2
000730- 022767  077600  002324   1$:     CMP     #77600,HDRWD    ;CURRENT ADDRESS=LAST TRACK?
000736- 001007                           BNE     2$              ;NO, CONTINUE
000740- 012767  077500  002314           MOV     #77500,HDRWD    ;NEXT ADDRESS=LAST CYL, HS=1
000746- 012767  000221  002310           MOV     #221,DIFWD      ;DIF WD 1, MARKER, SEEK OUT, HS=1
                                                                 ;SET CURRENT HD=1, SEEK OUT
```



```
000754- 000422                           BR      TAG2
000756- 032767  000004  002300   2$:     BIT     #4,DIFWD        ;SN SET IN DIF WORD
000764- 001404                           BEQ     3$              ;NO, 3$
000766- 062767  000200  002266           ADD     #200,HDRWD      ;YES, CYL WILL INCREMENT
000774- 000403                           BR      4$              ;SKIP OVER
000776- 162767  000200  002256   3$:     SUB     #200,HDRWD      ;NO, CYL WILL DECREMENT
001004- 032767  000020  002252   4$:     BIT     #20,DIFWD       ;HEAD SET?
001012- 001403                           BEQ     TAG2            ;NO, LEAVE EXPECTED ALONE
001014- 052767  000100  002240           BIS     #100,HDRWD      ;YES, SET HEAD SELECT BIT
001022- 004567  000002           TAG2:   JSR     R5,SEEK         ;PERFORM SEEK
001026- 004767  000270                   JSR     PC,WTRDY        ;WAIT FOR SEEK TO FINISH
001032- 004567  000616                   JSR     R5,RDHDR        ;READ HEADER VERIFY CORRECT
                                                                 ;SEEK
001036- 016767  002170  003316           MOV     T.MP,CURADR     ;READ HEADER
001044- 042767  000077  003310           BIC     #77,CURADR      ;CLEAR OUT SECTOR BITS
001052- 026767  003304  002202           CMP     CURADR,HDRWD    ;WAS SEEK CORRECT?
001060- 001425                           BEQ     6$              ;YES, CONTINUE
                                                                 ;NO REPORT ERROR
001062- 016767  002122  177010           MOV     RLCS,CSRA
001070- 017767  002114  177004           MOV     @RLCS,ACSR
001076- 017767  002106  177000           MOV     @RLCS,ASTAT
001104- 104403  000000- 005112-          MSGNS,BEGIN,BDSEEK      ;ASCII MESSAGE CALL WITH COMMON HEADER

001112- 012767  000051  176766           MOV     #51,ERRTYP      ;BAD SEEK
                                         ;*****************************************************************
001120- 104405  000000- 000000           ERDERS,BEGIN,NULL       ;SEEK WAS BAD
                                         ;*****************************************************************

001126- 016767  003230  002126           MOV     CURADR,HDRWD    ;MAKE MISTAKE NEW HDRWD

001134- 026727  002122  077700   6$:     CMP     HDRWD,#77700    ;ARE WE ON LAST TRACK
001142- 001002                           BNE     7$              ;NO, CONTINUE
001144- 000167  177400                   JMP     TAG             ;YES, GO GET ANOTHER CAUSE ITS THE BAD SECTOR TRACK

001150- 016767  176756  002116   7$:     MOV     RBUFSZ,WCNT2    ;GET BUFFER SIZE (READ)
001156- 005467  002112                   NEG     WCNT2           ;NEGATE FOR RLMP
001162- 104414  000000-                  GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
001166- 016767  176750  002076           MOV     WBUFSZ,WCNT1    ;GET BUFFER SIZE (WRITE)
001174- 005467  002072                   NEG     WCNT1           ;NEGATE FOR RLMP

001200- 004567  000324                   JSR     R5,WRITE        ;WRITE DATA
001204- 005767  002030                   TST     RWER            ;ERROR???
001210- 001016                           BNE     5$              ;YES, SKIP READ
001212- 004567  000302                   JSR     R5,WRCHK
001216- 005767  002016                   TST     RWEP
001222- 001011                           BNE     5$
001224- 004567  000340                   JSR     R5,READ         ;READ DATA

001230- 005767  002004                   TST     RWER
001234- 001004                           BNE     5$
001236- 104412  000000- 000126-          CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
001244- 001246-                          .+2                     ; IF ERROR, CONTINUE

001246- 005067  001766           5$:     CLR     RWER
001252- 005267  001772                   INC     CNT
```

```
001256' 022767  002000  001764          CMP     #1024.,CNT
001264' 001002                          BNE     1$
001266' 000167  177160                  JMP     LOOPL
001272' 000167  177240          1$:     JMP     LOOP

001276' 032767  000001  176512  CKDROP: BIT     #BIT0,SR1
001304' 001402                          BEQ     1$
001306' 004567  000102                  JSR     R5,DROP
001312' 000167  177134          1$:     JMP     LOOPL

001316'                         FINI:
001316' 104410  000000'                 ENDS,BEGIN              ;DROP THE MODULE

                                ;WAIT FOR DRIVE READY

001322' 042777  001400  001660  WTRDY:  BIC     #1400,@RLCS
001330' 052777  000132  001652          BIS     DRIVE,@RLCS
001336' 012767  177777  001702          MOV     #77777,CLK      ;SET UP TIMEOUT
001344' 032777  000001  001636  1$:     BIT     #1,@RLCS        ;DRIVE READY?
001352' 001017                          BNE     2$              ;YES, EXIT
001354' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001360' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001364' 005367  001656                  DEC     CLK             ;CHECK TIMEOUT
001370' 001365                          BNE     1$              ;GO BACK IF TIMEOUT OKAY

001372' 012767  000006  176506          MOV     #6,ERRTYP       ;DRIVE NOT READY
                                        ;*****************************************************
001400' 104405  000000' 003210'         HRDER$,BEGIN,RLCS       ;DRIVE NOT READY
                                        ;*****************************************************

001406' 004567  000002                  JSR     R5,DROP         ;CLEAR THAT DRIVE FROM LIST
                                                                ;DROP THE DRIVE

001412' 000207                  2$:     RTS     PC

                                ;DROP THE DRIVE

001414'                         DROP:
001414' 104403  000000' 005032' MSGN$,BEGIN,DROPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
001422' 046767  001616  001640          BIC     DRVMSK,DVICE    ;CLEAR THAT DRIVE
001430' 001006                          BNE     1$              ;ANY LEFT, YES 1$

001432' 104403  000000' 005042' MSGN$,BEGIN,NOLEFT      ;ASCII MESSAGE CALL WITH COMMON HEADER
001440' 012605                          MOV     (SP)+,R5
001442' 000167  177650                  JMP     FINI            ;DROP THE MODULE

001446' 000205                  1$:     RTS     R5              ;EXIT

                                ;
                                ;ROUTINE TO SET UP RL11 REGISTERS, VECTOR AND BR LEVEL
                                ;
```



```
001450' 016700  176332  SETUP:  MOV     ADDR,R0         ;GET BASE ADDRESS
001454' 010067  001530          MOV     R0,RLCS         ;CONTROL REGISTER
001460' 005720                  TST     (R0)+           ;INCREMENT FOR NEXT
001462' 010067  001524          MOV     R0,RLBA         ;BUS ADDRESS
001466' 005720                  TST     (R0)+           ;INCREMENT FOR NEXT
001470' 010067  001520          MOV     R0,RLDA         ;DISK ADDRESS
001474' 005720                  TST     (R0)+           ;INCREMENT FOR NEXT
001476' 010067  001514          MOV     R0,RLMP         ;DATA BUFFER
001502' 016700  176302          MOV     VECTOR,R0       ;GET VECTOR ADDRESS
001506' 012720  000252'         MOV     #START,(R0)+    ;SET POINTER
001512' 116710  176274          MOVB    BR1,(R0)        ;SET PRIORITY
001516' 000207                  RTS     PC              ;RETURN

                                ;DRIVERS (INTERRUPT)
001520' 012767  000102  001530  WRCHK:  MOV     #102,FUNC
001526' 000403                  BR      WRCOM

001530' 012767  000112  001520  WRITE:  MOV     #112,FUNC       ;WRITE FUNCTION
001536' 016777  001530  001452  WRCOM:  MOV     WCNT1,@RLMP     ;WORD COUNT
001544' 016777  001512  001442          MOV     HDRWD,@RLDA     ;DISK ADDRESS
001552' 016777  176356  001432          MOV     WBUFEA,@RLBA    ;BUFFER ADDRESS
001560' 016767  176352  001472          MOV     WBUFEA,XMEM     ;EXT. MEM. BITS
001566' 000453                  BR      EXEC
001570' 012767  000114  001460  READ:   MOV     #114,FUNC       ;READ FUNCTION
001576' 016777  001472  001412          MOV     WCNT2,@RLMP     ;WORD COUNT
001604' 016777  001452  001402          MOV     HDRWD,@RLDA     ;DISK ADDRESS
001612' 016777  176310  001422          MOV     RBUFEA,@RLBA    ;BUFFER ADDRESS
001620' 016767  176304  001432          MOV     RBUFEA,XMEM     ;EXT. MEM. BITS
001626' 000433                  BR      EXEC
001630' 012767  000106  001420  SEEK:   MOV     #106,FUNC       ;SEEK FUNCTION
001636' 016777  001422  001350          MOV     DIFWD,@RLDA     ;DIFFERENCE WORD
001644' 052777  000001  001342          BIS     #1,@RLDA        ;SET MARKER BIT
001652' 000421                  BR      EXEC
001654' 012767  000110  001374  RDHDR:  MOV     #110,FUNC       ;READ HEADER FUNCTION
001662' 000415                  BR      EXEC
001664' 012777  000003  001322  GSTAT:  MOV     #3,@RLDA        ;GET STATUS, MARKER
001672' 012767  000104  001356          MOV     #104,FUNC       ;GET STATUS FUNCTION
001700' 000406                  BR      EXEC
001702' 012777  000013  001304  DRVRTS: MOV     #13,@RLDA       ;RESET, GET STATUS, MARKER
001710' 012767  000104  001340          MOV     #104,FUNC       ;GET STATUS
001716' 056767  001316  001332  EXEC:   BIS     XMEM,FUNC       ;SET EA BITS
001724' 056767  001336  001324          BIS     DRIVE,FUNC      ;SELECT DRIVE
001732' 012777  001752' 176050          MOV     #INTSRV,@VECTOR ;SET UP INTERRUPT VECTOR
001740' 016777  001312  001242          MOV     FUNC,@RLCS      ;ISSUE FUNCTION
001746' 104400  000000'                 EXIT$S,BEGIN                    ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

001752'                         INTSRV:
                                ;-------------------------------------------------------------
001752' 000004  000000' 001760' QIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;-------------------------------------------------------------

001760' 005067  001250  1$:     CLR     T.STAT
001764' 016767  001220  176106  MOV     RLCS,CSRA       ;LOAD ADDR OF CSR
001772' 017767  001212  176102  MOV     @RLCS,ACSR      ;LOAD CONTENTS OF CSR
002000' 016767  176076  001216  MOV     ACSR,T.CS
002006' 017767  001200  001212  MOV     @RLBA,T.BA
```

```
002014' 017767 001174 001206        MOV     @RLDA,T.DA
002022' 017767 001170 001202        MOV     @RLMP,T.MP
002030' 005767 001170               TST     T.CS                ;ANY ERRORS
002034' 100403                      BMI     11$                 ;YES, CONTINUE TO CHECK
002036' 005067 001142               CLR     RETRY
002042' 000205                      RTS     R5                  ;NO, RETURN CALL+4 SKIP RETRY
002044' 005267 001170       11$:    INC     RWER
002050' 012767 005025' 003026       MOV     #NULLX,HTYPE        ;SETUP FOR NULL PRINT
002056' 032767 040000 001140        BIT     #BIT14,T.CS         ;DRIVE ERROR
002064' 001457                      BEQ     2$                  ;NO,BRANCH
002066' 012777 000003 001120        MOV     #3,@RLDA            ;GET STATUS
002074' 016767 000004 001112        MOV     #4,TMP
002102' 056767 001160 001112        BIS     DRIVE,TMP
002110' 016777 001106 001072        MOV     TMP,@RLCS
002116'                     99$:
002116' 104407 000000'              BREAK$,BEGIN                ;TEMPORARY RETURN TO MONITOR....
002122' 104407 000000'              BREAK$,BEGIN                ;THEN CONTINUE AT NEXT INSTRUCTION.
002126' 032777 000200 001054        BIT     #200,@RLCS
002134' 001770                      BEQ     99$
002136' 017767 001054 001070        MOV     @RLMP,T.STAT
002144' 104403 000000' 005076'      MSGN$,BEGIN,DRVERR          ;ASCII MESSAGE CALL WITH COMMON HEADER
002152' 012767 000006 175726        MOV     #6,ERRTYP           ;DRIVE ERROR
                                    ;*******************************************************************
002160' 104405 000000' 005122'      ORDER$,BEGIN,TABLE          ;
                                    ;*******************************************************************
002166' 012777 000013 001020        MOV     #13,@RLDA
002174' 016777 001022 001006        MOV     TMP,@RLCS
002202'                     98$:
002202' 104407 000000'              BREAK$,BEGIN                ;TEMPORARY RETURN TO MONITOR....
002206' 104407 000000'              BREAK$,BEGIN                ;THEN CONTINUE AT NEXT INSTRUCTION.
002212' 032777 000200 000770        BIT     #200,@RLCS
002220' 001770                      BEQ     98$
002222' 000530                      BR      NORPT
002224' 032767 020000 000772 2$:    BIT     #BIT13,T.CS         ;NXM SET
002232' 001404                      BEQ     21$
002234' 012767 005021' 002642       MOV     #NXM,HTYPE
002242' 000444                      BR      RPTERR
002244' 032767 002000 000752 21$:   BIT     #BIT10,T.CS         ;OPI SET
002252' 001423                      BEQ     4$                  ;NO, CHECK DCRC,DCK
002254' 012767 004774' 002622       MOV     #OPI,HTYPE          ;INITIAL SET FOR OPI
002262' 032767 004000 000734        BIT     #BIT11,T.CS         ;HCRC?
002270' 001404                      BEQ     3$                  ;NO, BRANCH
002272' 012767 005014' 002604       MOV     #HCRC,HTYPE         ;HCRC ERROR
002300' 000504                      BR      FNDRSC              ;FIND BAD SECTOR
002302' 032767 010000 000714 3$:    BIT     #BIT12,T.CS         ;HNF
002310' 001421                      BEQ     RPTERR              ;NO, IT'S OPI GO REPORT
002312' 012767 005010' 002564       MOV     #HNF,HTYPE          ;HNF ERROR
002320' 000474                      BR      FNDRSC              ;GO CHECK BAD SECTOR FILE
002322' 032767 004000 000674 4$:    BIT     #BIT11,T.CS         ;DCK?
002330' 001406                      BEQ     5$                  ;NO,MUST BE DLT
002332' 005367 000672               DEC     T.DA                ;BACK UP TO SECTOR THAT WAS BAD
002336' 012767 005004' 002540       MOV     #DCK,HTYPE          ;DCK ERROR
002344' 000462                      BR      FNDRSC              ;GO CHECK BAD SECTOR FILE
002346' 012767 005000' 002530 5$:   MOV     #DLT,HTYPE          ;SETUP DLT ERROR

002354' 032767 000004 175434 RPTERR: BIT    #BIT2,SR1           ;PRINTING SOFTERRORS
```



```
002362' 001011                      BNE     55$                 ;NO, SKIP PRINT

002364' 104403 000000' 005102'      MSGN$,BEGIN,SOFT            ;ASCII MESSAGE CALL WITH COMMON HEADER
002372' 012767 000001 175506        MOV     #1,ERRTYP           ;DATA ERROR
                                    ;*******************************************************************
002400' 104406 000000' 005122'      SOFER$,BEGIN,TABLE          ;
                                    ;*******************************************************************

002406' 026767 000572 000572 55$:   CMP     RETRY,LIMIT         ;RETRY EXHAUSTED
002414' 001405                      BEQ     6$                  ;YES, NO MORE RETRIES
002416' 005267 000562               INC     RETRY
002422' 162705 000004               SUB     #4,R5
002426' 000205                      RTS     R5

002430' 012700 004370'      6$:     MOV     #FNCLST,R0          ;SET LIST TO GET FUNCTION
002434' 016701 000616               MOV     FUNC,R1
002440' 006201                      ASR     R1
002442' 042701 177770               BIC     #177770,R1          ;LEAVE ONLY FUNCTION
002446' 005301                      DEC     R1
002450' 005720              7$:     TST     (R0)+               ;BUMP POINTER
002452' 005301                      DEC     R1
002454' 001375                      BNE     7$                  ;STAY IN LOOP TIL FOUND

002456' 011067 002370               MOV     (R0),EXCEED         ;GET FUNCTION MESSAGE
002462' 016767 002416 002374        MOV     HTYPE,TER1          ;GET ERROR TYPE
002470' 104403 000000' 005062'      MSGN$,BEGIN,HARD            ;ASCII MESSAGE CALL WITH COMMON HEADER
002476' 104403 000000' 005052'      MSGN$,BEGIN,EXCEED          ;ASCII MESSAGE CALL WITH COMMON HEADER
002504' 005067 000474       NORPT:  CLR     RETRY
002510' 000205                      RTS     R5

                                    ;
                                    ;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 PDHDR IF READ HDR THEN CHECK ONLY TRACK AND
                                    ;CYLINDER.
                                    ;
002512' 012701 077600       FNDRSC: MOV     #77600,R1           ;LAST TRACK
002516' 016700 000540               MOV     HDRWD,R0            ;PRESENT POSITION
002522' 042700 000160               BIC     #160,R0             ;CLEAR OUT HEAD
002526' 160001                      SUB     R0,R1               ;CALC SEEK DIFFERENCE
002530' 010177 000460               MOV     R1,@RLDA            ;LOAD SEEK DIFFERENCE
002534' 052777 000025 000452        BIS     #25,@RLDA           ;SET HEAD 1, SEEK IN
002542' 016767 000520 000466        MOV     DRIVE,MFLG          ;SELECT DRIVE. (MFLG USED)
002550' 052767 000006 000460        BIS     #6,MFLG             ;SET UP SEEK
002556' 016777 000454 000424        MOV     MFLG,@RLCS          ;SEEK
002564' 004767 176532               JSR     PC,WTRDY            ;WAIT FOR SEEK TO FINISH

                                    ;NOW SITTING ON LAST TRACK, RECOVER BAD SECTOR FILES AND COMPARE

002570' 012700 077700               MOV     #77700,R0           ;STARTING SECTOR 0
002574' 005067 000436               CLR     MFLG                ;SWITCH TO TELL US MANUF OR FIELD FILE
002600' 005067 000436               CLR     FND                 ;FLAG TO INDICATE HEADER FOUND IN LIST
002604' 010077 000404       2$:     MOV     R0,@RLDA            ;LOAD SECTOR TO READ
002610' 012777 177400 000400        MOV     #-256.,@RLMP        ;TWO SECTOR READ
002616' 016777 175304 000366        MOV     RBUFPA,@RLBA        ;WHERE TO READ TO
002624' 012777 000215 000356        MOV     #215,@RLCS          ;READ FUNCTION
```

```
                                BIS     DRIVE,@RLCS     ;BIT SET IN DRIVE SELECT
                                BIS     RROFEA,@RLCS    ;SET IN EA BITS IF NEEDED
                                BIC     #200,@RLCS      ;ISSUE FUNCTION
                                JSR     PC,WTRDY        ;WAIT FO DRIVE
                                TST     @RLCS           ;READ SUCCESSFUL??
                                BPL     4$              ;YES, GO CHECK FOR SECTOR

                                ADD     #4,R0           ;NO, NEXT SECTOR
                                TST     MFLG            ;WHICH WE READING, MAUF OR FIELD
                                BNE     3$              ;FIELD COMPARE AGAINST 77750
                                CMP     #77724,R0       ;MANUFACTURING, AT END
                                BNE     2$              ;NO, GO BACK AND READ NEXT

                        99$:
                                MSGNS,BEGIN,NOSEC       ;ASCII MESSAGE CALL WITH COMMON HEADER
                                JSR     R5,DROP
                                JMP     OLPSK

                        3$:     CMP     #77750,R0       ;AT END OF FIEL BAD
                                BNE     2$              ;NO, GO BACK
                                BR      99$             ;YES GO DROP DRIVE

                        4$:     MOV     RBUFVA,R1       ;GET WHERE WE READ
                                ADD     #10,R1          ;SKIP PAST I.D. ETC.....
                                MOV     #126.,R2        ;ONLY 126 ENTRIES
                        44$:    MOV     (R1)+,R3        ;GET CYLINDER
                                BMI     88$             ;MINUS WE'RE DONE
                                MOV     (R1)+,R4        ;GET TRACK AND SECTOR
                                SWAB    R3              ;ALIGN PROPERLY
                                ASR     R3
                                BISB    R4,R3
                                BIT     #400,R4
                                BEQ     5$
                                BIS     #100,R3
                        5$:     CMP     #HCRC,HTYPE     ;IS ERROR HCRC?
                                BNE     6$              ;NO, GO LOOK FOR BAD SECTOR
                                CMP     #110,FUNC       ;WE'RE WE DOING READ HEADER
                                BNE     6$              ;NO, GO LOOK FOR BAD SECTOR
                                BIC     #77,R3          ;YES, CLEAR SCETOR BITS
                                CMP     R3,HDRWD        ;BAD SECTOR
                                BEQ     7$
                                BR      8$
                        6$:     CMP     R3,T.DA         ;IS THIS ONE IT???????
                                BNE     8$              ;NO
                        7$:     INC     FND
                                BR      9$
                        8$:     DEC     R2              ;CHECKED WHOLE FILE
                                BNE     44$             ;NO
                        88$:    TST     MFLG            ;WHICH WE DOING
                                BNE     9$              ;FIELD, WE'RE DONE
                                INC     MFLG            ;MANUFACT. THEN SET UP FIELD
                                MOV     #77724,R0
                                BR      2$

                        9$:     MOV     HDRWD,R0
                                MOV     #77600,R1
```



```
                                BIC     #100,R0
                                SUB     R0,R1
                                MOV     R1,@RLDA
                                BIS     #1,@RLDA
                                BIT     #160,HDRWD
                                BEQ     10$
                                BIS     #20,@RLDA
                        10$:    MOV     DRIVE,MFLG
                                BIS     #,MFLG
                                MOV     MFLG,@RLCS
                                JSR     PC,WTRDY
                                TST     FND
                                BNE     11$
                                JMP     RPTERR
                        11$:    JMP     NORPT



                        ;LOCATIONS USED BY MODULE

                        RETRY:  .WORD   0
                        LIMIT:  .WORD   3
                        RLCS:   .WORD   0
                        RLBA:   .WORD   0
                        RLDA:   .WORD   0
                        RLMP:   .WORD   0
                                .WORD   177777
                        TMP:    .WORD   0
                        T.CS:   .WORD   0
                        T.BA:   .WORD   0
                        T.DA:   .WORD   0
                        T.MP:   .WORD   0
                        T.STAT: .WORD   0
                        MFLG:   .WORD   0
                        RWER:   .WORD   0
                        FND:    .WORD   0
                        DRVMSK: .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
                        XMEM:   .WORD   0               ;EA BITS FOR R/W
                        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 "DVIDIN"
                        WCNT1:  .WORD   0               ;WORD COUNT (WRITE)
                        WCNT2:  .WORD   0               ;WORD COUNT (READ)
                        BUFIN:  .BLKW   256.
                        BSECBF: .BLKW   25.             ;BAD SECTOR LIST
                        CURWSC: .WORD   0
                        CURADR: .WORD   0
                        NXTADR: .WORD   0
                        NUMR:   .WORD   0
                        ENCLST: .WORD   0
```

SEQ 0014

```
004372' 004527'                         MES9
004374' 004507'                         MES6
004376' 004473'                         MES5
004400' 004521'                         MES8
004402' 004514'                         MES7

004404' 047516  042040  044522  MES1:   .ASCIZ  "NO DRIVES PRESENT %"
004412' 042524  020123  051120
004420' 051505  047105  020124
004426' 000045
004430' 047516  042040  044522  MES2:   .ASCIZ  "NO DRIVES LEFT %"
004436' 042526  020123  042514
004444' 052106  022440     000
004451'    104  044522  042526  MES3:   .ASCIZ  "DRIVE "
004456' 000040
004460' 042040  047522  050120  MES4:   .ASCIZ  " DROPPED %"
004466' 042105  022440     000
004473'    122  040505  020104  MES5:   .ASCIZ  "READ HEADER"
004500' 042510  042101  051105
004506'    000
004507'    123  042505  000113  MES6:   .ASCIZ  "SEEK"
004514' 042522  042101     000  MES7:   .ASCIZ  "READ"
004521'    127  044522  042524  MES8:   .ASCIZ  "WRITE"
004526'    000
004527'    104  044522  042526  MES9:   .ASCIZ  "DRIVE RESET"
004534' 051040  051505  052105
004541'    040
004543'    040  042522  051124  MES10:  .ASCIZ  " RETRY LIMIT EXCEEDED%"
004550' 020131  044514  044515
004556' 020124  054105  042503
004564' 042105  042105  000045
004572' 051445  042505  020113  MES11:  .ASCIZ  "?SEEK TO WRONG CYLINDER%"
004600' 047524  053440  047522
004606' 043516  041440  046131
004614' 047111  042504  022522
004622'    000
004623'    104  052101  020101  MES12:  .ASCIZ  "DATA LATE%"
004630' 040514  042524  000045
004636' 051104  053111  020105  MES13:  .ASCIZ  "DRIVE ERROR%"
004644' 051105  047522  022522
004652'    000
004653'    123  043117  020124  MES14:  .ASCIZ  "SOFT ERROR   "
004660' 051105  047522  020122
004666' 020040     000
004671'    122  030114  020061  MES15:  .ASCIZ  "RL01 LOAD UNIT DROPPED"
004676' 047514  042101  052440
004704' 044516  020124  051104
004712' 050117  043520  000104
004720' 040510  042122  042440  MES16:  .ASCIZ  "HARD ERROR   "
004726' 051122  051117  020040
004734' 000040
004736' 040503  052116  051040  NBDSC:  .ASCIZ  "CANT RECOVER BAD SECTOR FILE%"
004744' 041505  053117  051105
004752' 041040  042101  051440
004760' 041505  047524  020122
```


SEQ 0015

```
004766' 044506  042514  000045
004774' 050117  000111
005000' 046104  000124
005004' 041504  000123
005010' 047110  000106
005014' 041510  041522     000  HCRC:   .ASCIZ  "HCRC"
005021'    116  046530     000  NXM:    .ASCIZ  "NXM"
005025'    040     000  NULLX:  .ASCIZ  " "
005027'    045     000  CR:     .ASCIZ  "%"

005032'                         .EVEN

005032' 004451'                 DROPMS: MES3
005034' 004366'                         NUMR
005036' 004460'                         MES4
005040' 177777                          177777

005042' 004430'                 NOLEFT: MES2
005044' 177777                          177777

005046' 004404'                 ABORT:  MES1
005050' 177777                          177777

005052' 000000                  EXCEED: .WORD   0
005054' 004543'                         MES10
005056' 005027'                         CR
005060' 177777                          177777

005062' 004720'                 HARD:   MES16
005064' 000000                  TERI:   .WORD   0
005066' 005027'                         CR
005070' 177777                          177777

005072' 004736'                 NOSEC:  NBDSC
005074' 177777                          177777

005076' 004636'                 DRVERR: MES13
005100' 177777                          177777

005102' 004653'                 SOFT:   MES14
005104' 000006                  HTYPE:  .WORD   6
005106' 005027'                         CR
005110' 177777                          177777

005112' 004572'                 BDSEEK: MES11
005114' 177777                          177777

005116' 004671'                 DROPLD: MES15
005120' 177777                          177777

                                ;REGISTERS OF RL11
005122'                         .EVEN
                                TABLE:
```

SEQ 0016

```
779  005122' 003224'                .WORD   T.CS            ;CONTROL AND STATUS REGISTER
780  005124' 003226'                .WORD   T.BA            ;BUS ADDRESS REGISTER
781  005126' 003230'                .WORD   T.DA
782  005130' 003232'                .WORD   T.MP
783  005132' 003234'                .WORD   T.STAT          ;HAS STATUS ON DRIVE ERROR
784  005134' 177777                 .WORD   177777          ;TERMINATOR
785          000001                 .END
```


SEQ 0017

```
ABORT   005046R          201     743#
ACSR    000102R          151#    294*    440*    441
ADDR    000006R          117#    387
ADDR22= 001000           190#
ASB     000106R          155#
ASTAT   000104R          153#    295*
AWAS    000110R          156#
BDSEEK  005112R          298     769#
BEGIN   000000R          114#    198     201     208     217     245     296     300     312     326     346     355     356
                         362     373     377     431     435     458     459     463     466     471     472     501     504
                         524     525     569
BIT0    = 000001         190#    337
BIT1    = 000002         190#    242
BIT10   = 002000         190#    490
BIT11   = 004000         190#    483     491
BIT12   = 010000         190#    487
BIT13   = 020000         190#    476
BIT14   = 040000         190#    451
BIT15   = 100000         190#
BIT2    = 000004         190#    498
BIT3    = 000010         190#
BIT4    = 000020         190#
BIT5    = 000040         190#
BIT6    = 000100         190#
BIT7    = 000200         190#
BIT8    = 000400         190#
BIT9    = 001000         190#
BREAK$= 104407           190#    355     356     458     459     471     472
BR1     000012R          119#    397
BR2     000013R          126#
BSECBF  004276R          661#
BTODS = 104421           190#
BUFIN   003276R          162     660#
CDATA$= 104412           190#    326
CDERCT  000144R          170#
CDWDCT  000146R          171#
CHKDRV  000456R          213     216#
CKDROP  001276R          337#
CLK     003246R          352*    357*    648#
CNT     003250R          294     225*    330*    331     649#
CONFIG  000056R          139#
CR      005027R          731#    750     755     766
CSRA    000100R          149#    293*    439*
CURADR  004362R          288*    289*    290     303     663#
CURMSG  004360R          662#
DATCK$= 104411           190#
DATER$= 104404           190#
DCK     005004R          404     726#
DIFWD   003264R          233*    249*    250*    252*    254*    255*    258*    267*    273*    276     281     417     655#
DLT     005006R          406     725#
DLTCNT  003254R          194*    651#
DRIVE   003266R          211*    219*    351     428     455     541     555     618     656#
DROP    001414R          339     365     372#    570
DROPLD  005116R          198     772#
DROPMS  005032R          373     735#
DRVERR  005076R          463     761#
```

```
DRVMSK  003744R          212*  214*   221   374   647#
DRVRTS  001702R          232   425#
DVICE   003270R          186*  195    197*  199   221   374*  657#
DVIDI   000014R          121#  186
EMDITS= 104413           180#  217
EMDS    = 104410         180#  346
ERRTYP  000106R          154#  298*   360*  464*  502*
EXCEED  000652R          522*  525    748#
EXEC    001716R          406   415    419   421   424   427#
EXITS$  = 104400         180#  431
FINI    001316R          202   345#   379
FNCLST  003370R          513   666#
FND     003242R          550*  599#   622   646#
FNDBSC  002512R          486   490    495   535#
FREE    000150R          172#
FUNC    003256R          401*  404*   410*  416*  420*  423*  426*  427*  428*  430   514   591   652#
GETPA$= 104415           180#  208
GETA$   001664R          422#
GWBUF$= 104414           180#  312
HARD    005062R          524   753#
HCRC    005014R          485   589    728#
HDRWD   003262R          235*  244*   260   259*  260*  263*  264   266*  270   272*  278*  280*  283*
                         290   302*   305   406   412   536   594   609   615   654#
HNF     005010R          489   727#
HDRCNT  000044R          134#
HDRPS=  104405           180#  300    362   466
HDRPAS  000050R          136#
HTYPE   005104R          450*  478*   482*  485*  489*  494*  496*  523   589   765#
ICOUNT  000036R          131#
ICOUNT  000040R          132#
IBNUM   000122R          161#
IMODX.= 000000           173#  313
INIT    000030R          128#
INTR    000120R          160#  183*
INTSRV  001752R          429   433#
LIMIT   003206R          507   632#
LOOP    000536R          334#  337
LOOPL   000452R          214#  222    333   340
MAP22$= 104416           180#
MES1    004404R          674#  743
MES10   004543R          695#  749
MES11   004572R          699#  769
MES12   004623R          704#
MES13   004636R          706#  761
MES14   004653R          709#  764
MES15   004671R          712#  772
MES16   004720R          716#  753
MES2    004430R          678#  740
MES3    004451R          681#  735
MES4    004460R          683#  737
MES5    004473R          669   685#
MES6    004507R          668   688#
MES7    004514R          671   689#
MES8    004521R          670   690#
MES9    004527R          667   692#
MFLG    003236R          541*  542*   543   549*  563   603   605*  618*  619*  620   644#
```



```
MODNAM  000000R          115#
MODSP   000252R          129   178#
MSG$$ = 104403           180#  198    201   296   373   377   463   501   524   525   569
MSG$ =  104402           180#
MSG =   104401           180#
MULDRV  003252R          409#  650#
NBDSC   004736R          719#  758
NOLEFT  005042R          377   740#
NORET   002504R          475   526#   625
NOSEC   005072R          569   756#
NULL    000000           180#  300
NULLX   005072R          450   730#
NUMB    004366R          210*  220*   665#  736
NXM     005021R          478   729#
NXTADR  004364R          664#
OLPSK   000456R          215#  571
OPEN    = 000000         116   122    123   124   125   142   143   144   145   146   147   148   149
                         151   153    155   156   158   159   160   163   164   166   167   169   170
                         171   172    180#
OPI     004774R          482   724#
OTOA$ = 104420           190#
PASCNT  000034R          130#
PIRQ$ = 000004           180#  435
PRCSP = 005726           180#
PRCSP2= 025626           180#
PRTY    = 000000         190#
PRTY0 = 000000           120   180#
PRTY1 = 000040           190#
PRTY2 = 000100           180#
PRTY3 = 000140           190#
PRTY4 = 000200           180#
PRTY5 = 000240           119   180#
PRTY6 = 000300           180#
PRTY7 = 000340           190#
PS    = 177776           180#
PSW   = 177776           180#
PUSH  = 005746           180#
PUSH2 = 024646           180#
RAND$ = 104417           180#  245
RANNUM  000054R          138#  246
RBUFEA  000130R          164#  414    556
RBUFPA  000126R          163#  326    413   553
RBUFSZ  000132R          165#  310
RBUFVA  000124R          162#  208    577
RDHDR   001654R          234   286    420#
READ    001570R          322   410#
RESTRT  000404R          157   204#
RES1    000056R          140#
RES2    000060R          141#
RETRY   003204R          223*  447*   507   509*  526*  631#
RLBA    003212R          390*  407*   413*  442   553*  634#
RLCS    003210R          393   294    395   356*  351*  353   362   388*  430*  439   440   456*  460
                         469*  473    543*  554*  555*  556*  557*  559   620*  633#
RLDA    003214R          392*  406*   412*  417*  418*  422*  425*  443   453*  468*  539*  540*  551*
RLMP    003216R          613*  614*   617*  635#
                         304*  405*   411*  444   462   552*  636#
```

```
RPTERR  002354R          479     488     498#    624
RSTRT   006112R          157#
RSTRT1  006414R          203     225     207#
RWER    003240R          224*    317     320     324     329*    449*    645#
SBADR   000102R          150#
SEEK    001630R          284     416#
SETUP   001450R          185     387#
SOFCNT  000042R          133#
SOFERS= 104406           180#    504
SOFPAS  000046R          135#
SOFT    005102R          501     764#
SPOINT  000032R          120#
SPSIZ =  000040            1#    173
SR1     000016R          122#    242     337     498
SR2     000020R          123#
SR3     000022R          124#
SR4     000024R          125#
START   002252R          128     181#    206     396
STAT    000026R          127#
SVR0    000062R          142#
SVR1    000064R          143#
SVR2    000066R          144#
SVR3    000070R          145#
SVR4    000072R          146#
SVR5    000074R          147#
SVR6    000076R          148#
SYSCNT  000052R          137#
TABLE   005122R          466     504     777#
TAG     003550R          242#    327
TAG1    003674R          243     263#
TAG2    001022R          261     269     275     282     284#
TER1    005064R          523*    754#
TMP     003222R          454*    455*    456     469     638#
TRPDFD= 000022           180#
T.BA    003226R          442*    640#    780
T.CS    003224R          441*    445     451     476     480     483     487     491     639#    779
T.DA    003230R          443*    483*    587     641#    781
T.MP    003232R          235     286     444*    642#    782
T.STAT  003234R          438*    462*    643#    783
VECTOR  000100R          118#    395     429*
WASADR  000104R          152#
WBUFEA  000136R          167#    408
WBUFPA  000134R          166#    407
WBUFRQ  000140R          168#
WBUFSZ  000142R          169#    313
WCNT1   003272R          313*    314*    425     658#
WCNT2   003274R          310*    311*    411     659#
WDER    000116R          159#    181*
WDTO    000114R          158#    182*
WRCHK   001520R          319     461#
WRCOM   001536R          402     465#
WRITE   001530R          317     434#
WTRDY   001322R          231     285     350#    544     558     621
XFLAG   000056R          116#
XMEM    003260R          408*    414*    427     653#
.       = 005136R        327     660#    661#    733#
```



```
. ABS.  000000      000
        005136      001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XRLAC0,XRLAC0/SOL/CRF:SYM=DDXCOM,XRLAC0
RUN-TIME:  1  2  .4  SECONDS
RUN-TIME RATIO: 174=3.0
CORE USED:  7K  (13 PAGES)
```