```
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEEEEEEEEEEEEEE  RRRRRRRRRRRR     FFFFFFFFFFFFFFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEE              RRR         RRR  FFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEEEEEEEEEEE     RRRRRRRRRRRR     FFFFFFFFFFFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR   RRR        FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEE              RRR      RRR     FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
EEEEEEEEEEEEEEE  RRR         RRR  FFF
```

FILEID**RKDISK   N 10

```
RRRRRRRR    KK      KK  DDDDDDDD      IIIIII      SSSSSSSS  KK      KK
RRRRRRRR    KK      KK  DDDDDDDD      IIIIII      SSSSSSSS  KK      KK
RR      RR  KK      KK  DD      DD      II      SS          KK      KK
RR      RR  KK      KK  DD      DD      II      SS          KK      KK
RR      RR  KK    KK    DD      DD      II      SS          KK    KK
RR      RR  KK    KK    DD      DD      II      SS          KK    KK
RRRRRRRR    KKKKKK      DD      DD      II        SSSSSS    KKKKKK
RRRRRRRR    KKKKKK      DD      DD      II        SSSSSS    KKKKKK
RR  RR      KK    KK    DD      DD      II              SS  KK    KK
RR  RR      KK    KK    DD      DD      II              SS  KK    KK
RR    RR    KK      KK  DD      DD      II              SS  KK      KK      ....
RR    RR    KK      KK  DD      DD      II              SS  KK      KK      ....
RR      RR  KK      KK  DDDDDDDD      IIIIII    SSSSSSSS    KK      KK      ....
RR      RR  KK      KK  DDDDDDDD      IIIIII    SSSSSSSS    KK      KK      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
        SUBROUTINE RKDISK (LUN)
C
C Version:      'V04-000'
C
C*****************************************************************************
C*                                                                           *
C*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
C*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
C*  ALL RIGHTS RESERVED.                                                     *
C*                                                                           *
C*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
C*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
C*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
C*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
C*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
C*  TRANSFERRED.                                                             *
C*                                                                           *
C*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
C*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
C*  CORPORATION.                                                             *
C*                                                                           *
C*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
C*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
C*                                                                           *
C*                                                                           *
C*****************************************************************************
C
C
C
C       AUTHOR  BRIAN PORTER            CREATION DATE   31-MAR-1979
C

c++
c       Functional description:
c
c       This module displays entries made for the RK611 controller.
c       The format of the error log packet after the 4 longword header
c       is as follows.
c
c       +-----------------------------------+
c       :               rkcs1               :
c       +-----------------------------------+
c       :               rkwc                :
c       +-----------------------------------+
c       :               rkba                :
c       +-----------------------------------+
c       :               rkda                :
c       +-----------------------------------+
c       :               rkcs2               :
c       +-----------------------------------+
c       :               rkds                :
c       +-----------------------------------+
c       :               rker                :
c       +-----------------------------------+
c       :               rkas                :
c       +-----------------------------------+
c       :               rkdc                :
```

```
c       +-------------------------------------+
c       :               rkmr1                 :
c       +-------------------------------------+
c       :               rkec1                 :
c       +-------------------------------------+
c       :               rkec2                 :
c       +-------------------------------------+
c       :               rkmr2                 :
c       +-------------------------------------+
c       :               rkmr3                 :
c       +-------------------------------------+
c       :         uba datapath number         :
c       +-------------------------------------+
c       :        uba datapath register        :
c       +-------------------------------------+
c       :     uba final mapping register      :
c       +-------------------------------------+
c       :    uba previous mapping register    :
c       +-------------------------------------+
c
c       Modified by:
c
C       V03-003 SAR0231         Sharon A. Reynolds,     28-Mar-1984
C               Changed the call to UCB$L_OWNUIC to ORB$L_OWNER.
C
C       V03-002 SAR0092         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
C       V03-001 SAR0049         Sharon A. Reynolds,     13-Jun-1983
C               Removed brief/cryptic support.
C
c       v02-005 BP0005          Brian Porter,           23-NOV-1981
c               minor edit.
c
c       v02-004 BP0004          Brian Porter,           03-NOV-1981
c               Added device attention support.
c
c       v02-003 BP0003          Brian Porter,           30-SEP-1981
c               Corrected call to uba_mapping.
c
c       v02-002 BP0002          Brian Porter,           23-JUL-1981
c               Added new uba handling routines.
c
c       v02-001 BP0001          Brian Porter,           29-JUN-1981
c               Made the default register output 16-bit.  Added call
c               to new DHEAD and LOGGER modules.  Removed call to
c               UNUSED_BITS, ILLEGAL_BITS and REGCHK.
c**
c--

        INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
        INCLUDE 'SRC$:DEVERR.FOR /NOLIST'

        BYTE            LUN

        INTEGER*4       RKCS1
        INTEGER*4       RKWC
        INTEGER*4       RKBA
        INTEGER*4       RKDA
        INTEGER*4       RKCS2
        INTEGER*4       RKDS
        INTEGER*4       RKER
        INTEGER*4       RKAS
        INTEGER*4       RKDC
        INTEGER*4       RKMR1
        INTEGER*4       RKEC1
        INTEGER*4       RKEC2
        INTEGER*4       RKMR2
        INTEGER*4       RKMR3
        INTEGER*4       UBA_REGS(4)
        INTEGER*4       FIELD
        INTEGER*4       DRIVE_FUNC
        INTEGER*4       COMPRESSC
        INTEGER*4       COMPRESS4

        logical*1       diagnostic_mode

        PARAMETER       TIMEOUT  = 96
        PARAMETER       XFER_ERR = 32
        PARAMETER       XFER_CMD = 8
        PARAMETER       RK06     = 1
        PARAMETER       RK07     = 2
        PARAMETER       SEEK     = 7

        EQUIVALENCE     (RKCS1,EMB$L_DV_REGSAV(0))
        EQUIVALENCE     (RKWC,EMB$L_DV_REGSAV(1))
        EQUIVALENCE     (RKBA,EMB$L_DV_REGSAV(2))
        EQUIVALENCE     (RKDA,EMB$L_DV_REGSAV(3))
        EQUIVALENCE     (RKCS2,EMB$L_DV_REGSAV(4))
        EQUIVALENCE     (RKDS,EMB$L_DV_REGSAV(5))
        EQUIVALENCE     (RKER,EMB$L_DV_REGSAV(6))
        EQUIVALENCE     (RKAS,EMB$L_DV_REGSAV(7))
        EQUIVALENCE     (RKDC,EMB$L_DV_REGSAV(8))
        EQUIVALENCE     (RKMR1,EMB$L_DV_REGSAV(9))
        EQUIVALENCE     (RKEC1,EMB$L_DV_REGSAV(10))
        EQUIVALENCE     (RKEC2,EMB$L_DV_REGSAV(11))
        EQUIVALENCE     (RKMR2,EMB$L_DV_REGSAV(12))
        EQUIVALENCE     (RKMR3,EMB$L_DV_REGSAV(13))
        EQUIVALENCE     (UBA_REGS,EMB$L_DV_REGSAV(14))

        CHARACTER*8     OFFSET_DIR(0:1)
        DATA    OFFSET_DIR(0)   /'FORWARD*'/
        DATA    OFFSET_DIR(1)   /'REVERSE*'/

        CHARACTER*5     RK06_OFFSET(1:5)
        DATA    RK06_OFFSET(1)  /'25*'/
        DATA    RK06_OFFSET(2)  /'200*'/
        DATA    RK06_OFFSET(3)  /'400*'/
        DATA    RK06_OFFSET(4)  /'800*'/
        DATA    RK06_OFFSET(5)  /'1200*'/

        CHARACTER*5     RK07_OFFSET(1:5)
        DATA    RK07_OFFSET(1)  /'12.5*'/
```

```
      DATA    RK07_OFFSET(2)  /'100*'/
      DATA    RK07_OFFSET(3)  /'200*'/
      DATA    RK07_OFFSET(4)  /'400*'/
      DATA    RK07_OFFSET(5)  /'600*'/

      CHARACTER*17    RK_FUNC(0:15)
      DATA    RK_FUNC(0)      /'SELECT DRIVE*'/
      DATA    RK_FUNC(1)      /'PACK ACKNOWLEDGE*'/
      DATA    RK_FUNC(2)      /'DRIVE CLEAR*'/
      DATA    RK_FUNC(3)      /'UNLOAD*'/
      DATA    RK_FUNC(4)      /'START SPINDLE*'/
      DATA    RK_FUNC(5)      /'RECALIBRATE*'/
      DATA    RK_FUNC(6)      /'OFFSET*'/
      DATA    RK_FUNC(7)      /'SEEK*'/
      DATA    RK_FUNC(8)      /'READ DATA*'/
      DATA    RK_FUNC(9)      /'WRITE DATA*'/
      DATA    RK_FUNC(10)     /'READ HEADER*'/
      DATA    RK_FUNC(11)     /'WRITE HEADER*'/
      DATA    RK_FUNC(12)     /'WRITE CHECK*'/
      DATA    RK_FUNC(13)     /'ILLEGAL FUNCTION*'/
      DATA    RK_FUNC(14)     /'ILLEGAL FUNCTION*'/
      DATA    RK_FUNC(15)     /'ILLEGAL FUNCTION*'/

      CHARACTER*7     RKCS1_1(0:0)
      DATA    RKCS1_1(0)      /'GO BIT*'/

      CHARACTER*28    RKCS1_2(5:9)
      DATA    RKCS1_2(5)      /'TRANSFER ERROR (VMS)*'/
      DATA    RKCS1_2(6)      /'INTERRUPT ENABLE*'/
      DATA    RKCS1_2(7)      /'CONTROLLER READY*'/
      DATA    RKCS1_2(8)      /'EXTENDED BUS ADDRESS BIT 16*'/
      DATA    RKCS1_2(9)      /'EXTENDED BUS ADDRESS BIT 17*'/

      CHARACTER*5     RKCS1_10(0:1)
      DATA    RKCS1_10(0)     /'RK06*'/
      DATA    RKCS1_10(1)     /'RK07*'/

      CHARACTER*20    RKCS1_3(11:11)
      DATA    RKCS1_3(11)     /'CONTROLLER TIME-OUT*'/

      CHARACTER*14    RK_FORMAT(0:1)
      DATA    RK_FORMAT(0)    /'16-BIT FORMAT*'/
      DATA    RK_FORMAT(1)    /'18-BIT FORMAT*'/

      CHARACTER*33    RKCS1_4(13:15)
      DATA    RKCS1_4(13)     /'DRIVE-TO-CONTROLLER PARITY ERROR*'/
      DATA    RKCS1_4(14)     /'DRIVE INTERRUPT*'/
      DATA    RKCS1_4(15)     /'COMBINED ERROR*'/

      CHARACTER*30    RKCS2_1(3:15)
      DATA    RKCS2_1(3)      /'RELEASE*'/
      DATA    RKCS2_1(4)      /'BUS ADDRESS INCREMENT INHIBIT*'/
      DATA    RKCS2_1(5)      /'SUBSYSTEM CLEAR*'/
      DATA    RKCS2_1(6)      /'INPUT READY*'/
      DATA    RKCS2_1(7)      /'OUTPUT READY*'/
      DATA    RKCS2_1(8)      /'UNIT FIELD ERROR*'/
      DATA    RKCS2_1(9)      /'MULTIPLE DRIVE SELECT*'/
```

```
        DATA    RKCS2_1(10)     /'PROGRAMMING ERROR*'/
        DATA    RKCS2_1(11)     /'NON-EXISTENT MEMORY*'/
        DATA    RKCS2_1(12)     /'NON-EXISTENT DRIVE*'/
        DATA    RKCS2_1(13)     /'UNIBUS PARITY ERROR*'/
        DATA    RKCS2_1(14)     /'WRITE CHECK ERROR*'/
        DATA    RKCS2_1(15)     /'DATA LATE ERROR*'/

        CHARACTER*16    RKDS_1(0:0)
        DATA    RKDS_1(0)       /'DRIVE AVAILABLE*'/

        CHARACTER*16    RKDS_2(2:7)
        DATA    RKDS_2(2)       /'OFFSET MODE*'/
        DATA    RKDS_2(3)       /'DRIVE AC LO*'/
        DATA    RKDS_2(4)       /'SPEED LOSS*'/
        DATA    RKDS_2(5)       /'DRIVE OFF TRACK*'/
        DATA    RKDS_2(6)       /'VOLUME VALID*'/
        DATA    RKDS_2(7)       /'DRIVE READY*'/

        CHARACTER*14    RK_DRIVETYPE(0:1)
        DATA    RK_DRIVETYPE(0) /'DRIVE IS RK06*'/
        DATA    RK_DRIVETYPE(1) /'DRIVE IS RK07*'/

        CHARACTER*16    RKDS_3(11:11)
        DATA    RKDS_3(11)      /'WRITE PROTECTED*'/

        CHARACTER*24    RKDS_4(13:15)
        DATA    RKDS_4(13)      /'POSITIONING IN PROGRESS*'/
        DATA    RKDS_4(14)      /'CURRENT DRIVE ATTENTION*'/
        DATA    RKDS_4(15)      /'STATUS VALID*'/

        CHARACTER*33    RKER_1(0:15)
        DATA    RKER_1(0)       /'ILLEGAL FUNCTION*'/
        DATA    RKER_1(1)       /'SEEK INCOMPLETE*'/
        DATA    RKER_1(2)       /'NON-EXECUTABLE FUNCTION*'/
        DATA    RKER_1(3)       /'CONTROLLER-TO-DRIVE PARITY ERROR*'/
        DATA    RKER_1(4)       /'FORMAT ERROR*'/
        DATA    RKER_1(5)       /'DRIVE TYPE ERROR*'/
        DATA    RKER_1(6)       /'ERROR CORRECTION HARD*'/
        DATA    RKER_1(7)       /'BAD SECTOR ERROR*'/
        DATA    RKER_1(8)       /'HEADER VERTICAL CHECK ERROR*'/
        DATA    RKER_1(9)       /'CYLINDER OVERFLOW ERROR*'/
        DATA    RKER_1(10)      /'INVALID DISK ADDRESS ERROR*'/
        DATA    RKER_1(11)      /'WRITE LOCK ERROR*'/
        DATA    RKER_1(12)      /'DRIVE TIMING ERROR*'/
        DATA    RKER_1(13)      /'OPERATION INCOMPLETE*'/
        DATA    RKER_1(14)      /'DRIVE UNSAFE*'/
        DATA    RKER_1(15)      /'DATA CHECK*'/

        CHARACTER*21    V1RKMR2(4:15)
        DATA    V1RKMR2(4)      /'SERVO SIGNAL PRESENT*'/
        DATA    V1RKMR2(5)      /'HEADS HOME*'/
        DATA    V1RKMR2(6)      /'BRUSHES HOME*'/
        DATA    V1RKMR2(7)      /'DOOR LATCHED*'/
        DATA    V1RKMR2(8)      /'CARTRIDGE PRESENT*'/
        DATA    V1RKMR2(9)      /'SPEED OK*'/
        DATA    V1RKMR2(10)     /'FORWARD*'/
        DATA    V1RKMR2(11)     /'REVERSE*'/
```

```
              DATA    V1RKMR2(12)     /'HEADS LOADING*'/
              DATA    V1RKMR2(13)     /'RETURN TO ZERO*'/
              DATA    V1RKMR2(14)     /'UNLOADING HEADS*'/
              DATA    V1RKMR2(15)     /'ODD PARITY BIT*'/

              CHARACTER*29    V1RKMR3(4:15)
              DATA    V1RKMR3(4)      /'SECTOR ERROR*'/
              DATA    V1RKMR3(5)      /'WRITE CURRENT, NO WRITE GATE*'/
              DATA    V1RKMR3(6)      /'WRITE GATE, NO TRANSITIONS*'/
              DATA    V1RKMR3(7)      /'HEAD FAULT*'/
              DATA    V1RKMR3(8)      /'MULTIPLE HEAD SELECT*'/
              DATA    V1RKMR3(9)      /'INDEX ERROR*'/
              DATA    V1RKMR3(10)     /'TRIBIT ERROR*'/
              DATA    V1RKMR3(11)     /'SERVO SIGNAL ERROR*'/
              DATA    V1RKMR3(12)     /'SEEK AND NO MOTION*'/
              DATA    V1RKMR3(13)     /'LIMIT DETECT ON SEEK*'/
              DATA    V1RKMR3(14)     /'SERVO UNSAFE*'/
              DATA    V1RKMR3(15)     /'ODD PARITY BIT*'/


              CALL FRCTOF (LUN)

              call dhead1 (lun,'UBA RK611')

              diagnostic_mode = .false.

              if (lib$extzv(5,1,rkmr1) .eq. 1) diagnostic_mode = .true.

              call linchk (lun,2)

              write(lun,20) rkcs1
     20       format(/' ',T8,'RKCS1',t24,z8.4)

              if (.not. diagnostic_mode) then

              CALL OUTPUT (LUN,RKCS1,RKCS1_1,0,0,0,'0')

              DRIVE_FUNC=LIB$EXTZV(1,4,RKCS1)

              CALL LINCHK (LUN,1)

              WRITE(LUN,30) RK_FUNC(DRIVE_FUNC)
     30       FORMAT(' ',T40,A<COMPRESSC (RK_FUNC(DRIVE_FUNC))>)

              CALL OUTPUT (LUN,RKCS1,RKCS1_2,5,5,9,'0')

              FIELD=LIB$EXTZV(10,1,RKCS1)

              CALL LINCHK (LUN,1)

              WRITE(LUN,40) RKCS1_10(FIELD)
     40       FORMAT(' ',T40,'CONTROLLER DRIVE TYPE ',
              1 A<COMPRESSC (RKCS1_10(FIELD))>)

              CALL OUTPUT (LUN,RKCS1,RKCS1_3,11,11,11,'0')

              FIELD=LIB$EXTZV(12,1,RKCS1)
```

```

                CALL LINCHK (LUN,1)

                WRITE(LUN,44) RK_FORMAT(FIELD)
        44      FORMAT(' ',T40,A<COMPRESSC (RK_FORMAT(FIELD))>)

                CALL OUTPUT (LUN,RKCS1,RKCS1_4,13,13,15,'0')
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,50) RKWC
        50      FORMAT(' ',T8,'RKWC',T24,Z8.4)

                CALL LINCHK (LUN,1)

                WRITE(LUN,60) RKBA
        60      FORMAT(' ',T8,'RKBA',T24,Z8.4)

                if (.not. diagnostic_mode) then

                IF (DRIVE_FUNC .GE. XFER_CMD) THEN

                CALL CALC_MAP (LUN,8,RKCS1,RKBA)
                ENDIF
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,70) RKDA
        70      FORMAT(' ',T8,'RKDA',T24,Z8.4)

                if (.not. diagnostic_mode) then

                CALL LINCHK (LUN,2)

                FIELD=LIB$EXTZV(0,5,RKDA)

                WRITE(LUN,80) FIELD
        80      FORMAT(' ',T40,'SECTOR = ',I<COMPRESS4 (FIELD)>,'.')

                FIELD=LIB$EXTZV(8,3,RKDA)

                WRITE(LUN,90) FIELD
        90      FORMAT(' ',T40,'TRACK  = ',I<COMPRESS4 (FIELD)>,'.')
                endif

                CALL LINCHK (LUN,1)

                WRITE(LUN,100) RKCS2
        100     FORMAT(' ',T8,'RKCS2',T24,Z8.4)

                if (.not. diagnostic_mode) then

                CALL LINCHK (LUN,1)

                FIELD=LIB$EXTZV(0,3,RKCS2)
```

```
          WRITE(LUN,110) FIELD
  110     FORMAT(' ',T40,'SELECTED DRIVE = ',I1,'.')

          CALL OUTPUT (LUN,RKCS2,RKCS2_1,3,3,15,'0')
          endif

          CALL LINCHK (LUN,1)

          WRITE(LUN,120) RKDS
  120     FORMAT(' ',T8,'RKDS',T24,Z8.4)

          if (.not. diagnostic_mode) then

          CALL OUTPUT (LUN,RKDS,RKDS_1,0,0,0,'0')

          CALL OUTPUT (LUN,RKDS,RKDS_2,2,2,7,'0')

          CALL LINCHK (LUN,1)

          FIELD=LIB$EXTZV(8,1,RKDS)

          WRITE(LUN,130) RK_DRIVETYPE(FIELD)
  130     FORMAT(' ',T40,A<COMPRESSC (RK_DRIVETYPE(FIELD))>)

          CALL OUTPUT (LUN,RKDS,RKDS_3,11,11,11,'0')

          CALL OUTPUT (LUN,RKDS,RKDS_4,13,13,15,'0')
          endif

          CALL LINCHK (LUN,1)

          WRITE(LUN,140) RKER
  140     FORMAT(' ',T8,'RKER',T24,Z8.4)

          if (.not. diagnostic_mode) then

          IF (JIAND(RKDS,'8000'X) .NE. 0) THEN

          CALL OUTPUT (LUN,RKER,RKER_1,0,0,15,'0')
          ENDIF
          endif

          CALL LINCHK (LUN,1)

          WRITE(LUN,150) RKAS
  150     FORMAT(' ',T8,'RKAS/OF',T24,Z8.4)

          if (.not. diagnostic_mode) then

          FIELD=LIB$EXTZV(0,6,RKAS)

          IF (FIELD .NE. 0) THEN

          CALL LINCHK (LUN,1)

          IF (FIELD .NE. 1 .AND.
             1   FIELD .NE. 8 .AND.
             2   FIELD .NE. 16 .AND.
             3   FIELD .NE. 32 .AND.
             4   FIELD .NE. 48) THEN

             WRITE(LUN,155)
    155      FORMAT(' ',T40,'INVALID OFFSET')

             GOTO 185

             ELSE IF (EMB$B_DV_TYPE .EQ. RK06) THEN

             FIELD=FIELD/8

             WRITE(LUN,160) RK06_OFFSET(FIELD)
    160      FORMAT(' ',T40,'OFFSET = ',
             1 A<COMPRESSC (RK06_OFFSET(FIELD))>,' MICRO INCHES')

             ELSE IF (EMB$B_DV_TYPE .EQ. RK07) THEN

             FIELD=FIELD/8

             WRITE(LUN,165) RK07_OFFSET(FIELD)
    165      FORMAT(' ',T40,'OFFSET = ',
             1 A<COMPRESSC (RK07_OFFSET(FIELD))>,' MICRO INCHES')
             ENDIF

             FIELD=LIB$EXTZV(7,1,RKAS)

             CALL LINCHK (LUN,1)

             WRITE(LUN,170) OFFSET_DIR(FIELD)
    170      FORMAT(' ',T40,'OFFSET DIRECTION = ',
             1 A<COMPRESSC (OFFSET_DIR(FIELD))>)
             ENDIF

    185      DO 195 I=8,15

             FIELD=LIB$EXTZV(I,1,RKAS)

             IF (FIELD .NE. 0) THEN

             CALL LINCHK (LUN,1)

             WRITE(LUN,190) (I-8)
    190      FORMAT(' ',T40,'ATTENTION DRIVE ',I1,'.')
             ENDIF

    195      CONTINUE
             endif

             CALL LINCHK (LUN,1)

             WRITE(LUN,200) RKDC
    200      FORMAT(' ',T8,'RKDC',T24,Z8.4)

             if (.not. diagnostic_mode) then
```

```

              FIELD=LIB$EXTZV(0,10,RKDC)

              IF (FIELD .NE. 0) THEN

              CALL LINCHK (LUN,1)

              WRITE(LUN,210) FIELD
      210     FORMAT(' ',T40,'DESIRED CYLINDER = ',I<COMPRESS4 (FIELD)>,'.')
              ENDIF
              endif

              CALL LINCHK (LUN,1)

              WRITE(LUN,230) RKMR1
      230     FORMAT(' ',T8,'RKMR1',T24,Z8.4)

              if (diagnostic_mode) then

              call linchk (lun,1)

              WRITE(LUN,235)
      235     FORMAT(' ',T40,'DIAGNOSTIC MODE')
              endif

              CALL LINCHK (LUN,1)

              WRITE(LUN,240) RKEC1
      240     FORMAT(' ',T8,'RKEC1',T24,Z8.4)

              CALL LINCHK (LUN,1)

              WRITE(LUN,250) RKEC2
      250     FORMAT(' ',T8,'RKEC2',T24,Z8.4)

              CALL LINCHK (LUN,1)

              WRITE(LUN,260) RKMR2
      260     FORMAT(' ',T8,'RKMR2',T24,Z8.4)

              if (.not. diagnostic_mode) then

              IF ((JIAND(RKMR1,'01'X) .NE. 0)
              1 .AND.
              2 (EMB$W_HD_ENTRY .EQ. TIMEOUT)
              3 .AND.
              4 (JIAND(RKMR3,'01'X) .NE. 0)) THEN

              CALL LINCHK (LUN,1)

              WRITE(LUN,261)
      261     FORMAT(' ',T40,'*** MESSAGE A1 ***')

              FIELD = LIB$EXTZV(0,3,RKMR2)

              CALL LINCHK (LUN,1)

              WRITE(LUN,262) FIELD
      262     FORMAT(' ',T40,'SELECTED DRIVE = ',I<COMPRESS4 (FIELD)>,'.')

              CALL OUTPUT (LUN,RKMR2,V1RKMR2,4,4,15,'0')
              ENDIF
              endif

              CALL LINCHK (LUN,1)

              WRITE(LUN,270) RKMR3
      270     FORMAT(' ',T8,'RKMR3',T24,Z8.4)

              if (.not. diagnostic_mode) then

              IF ((JIAND(RKMR1,'01'X) .NE. 0)
              1 .AND.
              2 (EMB$W_HD_ENTRY .EQ. TIMEOUT)
              3 .AND.
              4 (JIAND(RKMR3,'01'X) .NE. 0)
              5 .AND.
              6 (JIAND(RKMR3,'FFFC'X) .NE.0)) THEN

              CALL LINCHK (LUN,1)

              WRITE(LUN,272)
      272     FORMAT(' ',T40,'*** MESSAGE B1 ***')

              CALL OUTPUT (LUN,RKMR3,V1RKMR3,4,4,15,'0')
              ENDIF
              endif

              if (
              1 drive_func .ge. xfer_cmd
              1 .and.
              1 emb$w_hd_entry .ne. 96
              1 .and.
              1 emb$w_hd_entry .ne. 98
              1 ) then

              if (uba_regs(1)  .ne. 0) then

              call uba_datapath (lun,uba_regs(1),uba_regs(2))
              endif

              call calc_map2 (8,rkcs1,rkba,field)

              call uba_mapping (lun,field,uba_regs(3))

              if (
              1 lib$extzv(16,16,emb$l_dv_iosb1) .gt. 512
              1 .and.
              1 field .ne. 0
              1 ) then

              call uba_mapping (lun,(field-1),uba_regs(4))
              endif
              endif
```

```

        call linchk (lun,1)

        write(lun,275)
275     format(' ',:)

        if (emb$w_hd_entry .ne. 98) then

        call ucb$b_ertcnt (lun,emb$b_dv_ertcnt)

        call ucb$b_ertmax (lun,emb$b_dv_ertmax)
        endif

        call orb$l_owner (lun,emb$l_dv_ownuic)

        call ucb$l_char (lun,emb$l_dv_char)

        call ucb$w_sts (lun,emb$w_dv_sts)

        call ucb$l_opcnt (lun,emb$l_dv_opcnt)

        call ucb$w_errcnt (lun,emb$w_dv_errcnt)

        if (emb$w_hd_entry .ne. 98) then

        call ucb$l_media (lun,emb$l_dv_media)

        call linchk (lun,1)

        write(lun,275)

        call rkdisk_qio (lun,emb$w_dv_func)

        call irp$w_bcnt (lun,emb$w_dv_bcnt)

        call irp$w_boff (lun,emb$w_dv_boff)

        call irp$l_pid (lun,emb$l_dv_rqpid)

        call irp$q_iosb (lun,emb$l_dv_iosb1)
        endif

        RETURN
        END
```

```
PROGRAM SECTIONS

    Name                                          Bytes   Attributes

  0 $CODE                                          2873   PIC CON REL LCL   SHR   EXE    RD NOWRT LONG
  1 $PDATA                                          789   PIC CON REL LCL   SHR NOEXE    RD NOWRT LONG
  2 $LOCAL                                         3780   PIC CON REL LCL NOSHR NOEXE    RD   WRT LONG
  3 EMB                                             512   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG

    Total Space Allocated                          7954

ENTRY POINTS

    Address   Type  Name

  0-00000000        RKDISK

VARIABLES

    Address   Type  Name                                     Address   Type  Name

  2-00000954  L*1   DIAGNOSTIC_MODE                        2-0000095C  I*4   DRIVE_FUNC
  3-0000001C  L*1   EMB$B_DV_CLASS                         3-00000010  L*1   EMB$B_DV_ERTCNT
  3-00000011  L*1   EMB$B_DV_ERTMAX                        3-0000003E  L*1   EMB$B_DV_NAMLNG
  3-0000003A  L*1   EMB$B_DV_SLAVE                         3-0000001D  L*1   EMB$B_DV_TYPE
  3-00000036  I*4   EMB$L_DV_CHAR                          3-00000012  I*4   EMB$L_DV_IOSB1
  3-00000016  I*4   EMB$L_DV_IOSB2                         3-00000026  I*4   EMB$L_DV_MEDIA
  3-0000004E  I*4   EMB$L_DV_NUMREG                        3-0000002E  I*4   EMB$L_DV_OPCNT
  3-00000032  I*4   EMB$L_DV_OWNUIC                        3-0000001E  I*4   EMB$L_DV_RQPID
  3-00000000  I*4   EMB$L_HD_SID                           3-0000003F  CHAR  EMB$T_DV_NAME
  3-00000024  I*2   EMB$W_DV_BCNT                          3-00000022  I*2   EMB$W_DV_BOFF
  3-0000002C  I*2   EMB$W_DV_ERRCNT                        3-0000003C  I*2   EMB$W_DV_FUNC
  3-0000001A  I*2   EMB$W_DV_STS                           3-0000002A  I*2   EMB$W_DV_UNIT
  3-00000004  I*2   EMB$W_HD_ENTRY                         3-0000000E  I*2   EMB$W_HD_ERRSEQ
  2-00000958  I*4   FIELD                                  2-00000960  I*4   I
 AP-00000004@ L*1   LUN                                    3-0000006E  I*4   RKAS
  3-0000005A  I*4   RKBA                                   3-00000052  I*4   RKCS1
  3-00000062  I*4   RKCS2                                  3-0000005E  I*4   RKDA
  3-00000072  I*4   RKDC                                   3-00000066  I*4   RKDS
  3-0000007A  I*4   RKEC1                                  3-0000007E  I*4   RKEC2
  3-0000006A  I*4   RKER                                   3-00000076  I*4   RKMR1
  3-00000082  I*4   RKMR2                                  3-00000086  I*4   RKMR3
  3-00000056  I*4   RKWC

ARRAYS

    Address   Type  Name                                      Bytes  Dimensions

  3-00000000  L*1   EMB                                         512  (0:511)
  3-00000052  I*4   EMB$L_DV_REGSAV                             420  (0:104)
  3-00000006  I*4   EMB$Q_HD_TIME                                 8  (2)
  2-00000000  CHAR  OFFSET_DIR                                   16  (0:1)
  2-00000010  CHAR  RK06_OFFSET                                  25  (5)
  2-00000029  CHAR  RK07_OFFSET                                  25  (5)
```

```
                                                   B 12
RKDISK                                             16-Sep-1984 00:13:19    VAX-11 FORTRAN V3.4-56              Page  14
                                                    5-Sep-1984 14:21:27    DISK$VMSMASTER:[ERF.SRC]RKDISK.FOR;1

  2-00000152  CHAR RKCS1_1                                  7  (0:0)
  2-000001E5  CHAR RKCS1_10                                10  (0:1)
  2-00000159  CHAR RKCS1_2                                140  (5:9)
  2-000001EF  CHAR RKCS1_3                                 20  (11:11)
  2-0000021F  CHAR RKCS1_4                                 99  (13:15)
  2-00000282  CHAR RKCS2_1                                390  (3:15)
  2-00000408  CHAR RKDS_1                                  16  (0:0)
  2-00000418  CHAR RKDS_2                                  96  (2:7)
  2-00000494  CHAR RKDS_3                                  16  (11:11)
  2-000004A4  CHAR RKDS_4                                  72  (13:15)
  2-000004EC  CHAR RKER_1                                 528  (0:15)
  2-00000478  CHAR RK_DRIVETYPE                            28  (0:1)
  2-00000203  CHAR RK_FORMAT                               28  (0:1)
  2-00000042  CHAR RK_FUNC                                272  (0:15)
  3-0000008A  I*4  UBA_REGS                                16  (4)
  2-000006FC  CHAR V1RKMR2                                252  (4:15)
  2-000007F8  CHAR V1RKMR3                                348  (4:15)

LABELS

    Address     Label       Address     Label       Address     Label       Address     Label       Address     Label       Address     Label

  1-0000004C  20'        1-0000005F  30'        1-0000006B  40'        1-0000008F  44'        1-0000009B  50'        1-000000AC  60'
  1-000000BD  70'        1-000000CE  80'        1-000000E8  90'        1-00000102  100'       1-00000114  110'       1-00000132  120'
  1-00000143  130'       1-0000014F  140'       1-00000160  150'       1-00000174  155'       1-0000018A  160'       1-000001B0  165'
  1-000001D6  170'       0-000005A9  185        1-000001F7  190'           **      195        1-00000214  200'       1-00000225  210'
  1-00000249  230'       1-0000025B  235'       1-00000272  240'       1-00000284  250'       1-00000296  260'       1-000002A8  261'
  1-000002C2  262'       1-000002E4  270'       1-000002F6  272'       1-00000310  275'

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name               Type  Name               Type  Name               Type  Name               Type  Name               Type  Name

        CALC_MAP                 CALC_MAP2          I*4   COMPRESS4          I*4   COMPRESSC                DHEAD1                   FRCTOF
        IRP$L_PID                IRP$Q_IOSB               IRP$W_BCNT               IRP$W_BOFF         I*4   LIB$EXTZV                LINCHK
        ORB$L_OWNER              OUTPUT                   RKDISK_QIO               UBA_DATAPATH             UBA_MAPPING              UCB$B_ERTCNT
        UCB$B_ERTMAX             UCB$L_CHAR               UCB$L_MEDIA              UCB$L_OPCNT              UCB$W_ERRCNT             UCB$W_STS
        Subroutine RKDISK_QIO (lun,emb$w_dv_func)

        include 'src$:qiocommon.for /nolist'

        byte            lun

        integer*2       emb$w_dv_func

        integer*4       qiocode(0:1,0:63)


        if (qiocode(0,0) .eq. 0) then

        qiocode(1,00) = %loc(io$_nop)

        qiocode(1,01) = %loc(io$_unload)

        qiocode(1,02) = %loc(io$_seek)

        qiocode(1,03) = %loc(io$_recal)

        qiocode(1,04) = %loc(io$_drvclr)

        qiocode(1,05) = %loc(io$_release)

        qiocode(1,06) = %loc(io$_offset)

        qiocode(1,07) = %loc(io$_retcenter)

        qiocode(1,08) = %loc(io$_packack)

        qiocode(1,10) = %loc(io$_writecheck)

        qiocode(1,11) = %loc(io$_writepblk)

        qiocode(1,12) = %loc(io$_readpblk)

        qiocode(1,13) = %loc(io$_writehead)

        qiocode(1,14) = %loc(io$_readhead)

        qiocode(1,25) = %loc(io$_startspndl)

        qiocode(1,26) = %loc(io$_setchar)

        qiocode(1,27) = %loc(io$_sensechar)

        qiocode(1,32) = %loc(io$_writelblk)

        qiocode(1,33) = %loc(io$_readlblk)

        qiocode(1,35) = %loc(io$_setmode)

        qiocode(1,39) = %loc(io$_sensemode)
```

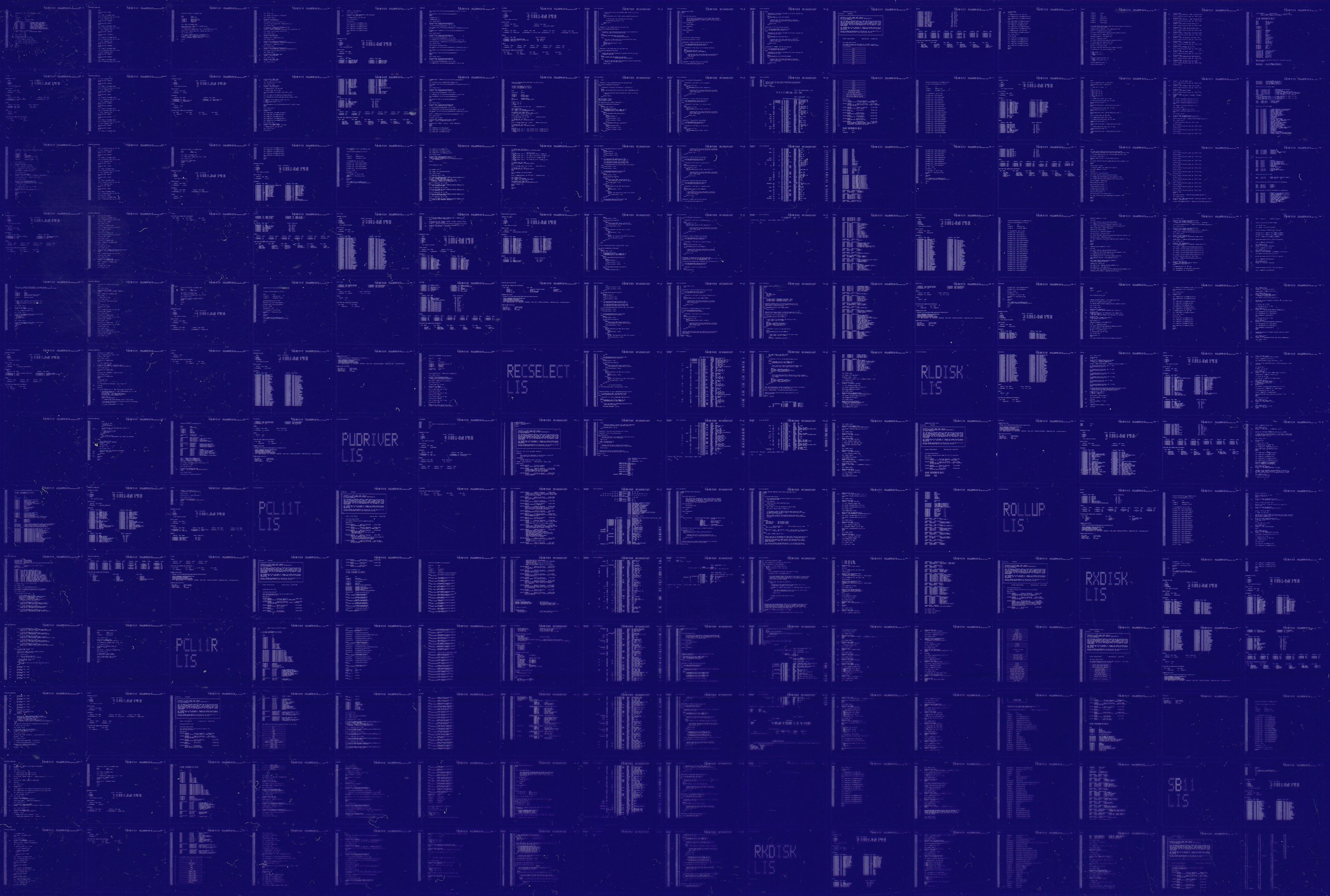
RL
PR
EN
VA
AR

```
        qiocode(1,48) = %loc(io$_writevblk)

        qiocode(1,49) = %loc(io$_readvblk)

        qiocode(1,50) = %loc(io$_access)

        qiocode(1,51) = %loc(io$_create)

        qiocode(1,52) = %loc(io$_deaccess)

        qiocode(1,53) = %loc(io$_delete)

        qiocode(1,54) = %loc(io$_modify)

        qiocode(1,56) = %loc(io$_acpcontrol)

        qiocode(1,57) = %loc(io$_mount)

        do  10,i = 0,63

        qiocode(0,i) = 33

        if (qiocode(1,i) .eq. 0) then

        qiocode(1,i) = %loc(qio_string)
        endif

10      continue
        endif

        call irp$w_func (lun,emb$w_dv_func,
        1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))

        return

        end
```

```
PROGRAM SECTIONS

    Name                                   Bytes   Attributes

  0 $CODE                                    295   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                     8   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                   548   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 QIOCOMMON                               1247   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                   2098


ENTRY POINTS

    Address   Type  Name

  0-00000000        RKDISK_QIO


VARIABLES

    Address   Type  Name                                  Address   Type  Name

 AP-00000008@ I*2   EMB$W_DV_FUNC                      2-00000200  I*4   I
  3-00000442  CHAR  IO$_ABORT                          3-0000034D  CHAR  IO$_ACCESS
  3-000003C2  CHAR  IO$_ACPCONTROL                     3-000004B3  CHAR  IO$_AVAILABLE
  3-00000297  CHAR  IO$_CLEAN                          3-00000369  CHAR  IO$_CREATE
  3-00000385  CHAR  IO$_DEACCESS                       3-00000393  CHAR  IO$_DELETE
  3-0000026D  CHAR  IO$_DIAGNOSE                       3-00000065  CHAR  IO$_DRVCLR
  3-000004CB  CHAR  IO$_DSE                            3-000000A9  CHAR  IO$_ERASETAPE
  3-00000276  CHAR  IO$_FORMAT                         3-00000071  CHAR  IO$_INITIALIZE
  3-00000014  CHAR  IO$_LOADMCODE                      3-000003A1  CHAR  IO$_MODIFY
  3-000003E2  CHAR  IO$_MOUNT                          3-00000000  CHAR  IO$_NOP
  3-0000009D  CHAR  IO$_OFFSET                         3-000000EB  CHAR  IO$_PACKACK
  3-000000E0  CHAR  IO$_QSTOP                          3-000003EF  CHAR  IO$_RDSTATS
  3-00000421  CHAR  IO$_READCSR                        3-00000169  CHAR  IO$_READHEAD
  3-000002B6  CHAR  IO$_READLBLK                       3-0000013F  CHAR  IO$_READPBLK
  3-00000200  CHAR  IO$_READPRESET                     3-00000195  CHAR  IO$_READTRACKD
  3-0000033A  CHAR  IO$_READVBLK                       3-0000045A  CHAR  IO$_READWTHBUF
  3-00000484  CHAR  IO$_READWTHXBUF                    3-0000004D  CHAR  IO$_RECAL
  3-0000007C  CHAR  IO$_RELEASE                        3-000001AB  CHAR  IO$_REREADN
  3-000001B8  CHAR  IO$_REREADP                        3-000000CA  CHAR  IO$_RETCENTER
  3-000002E6  CHAR  IO$_REWIND                         3-000002C9  CHAR  IO$_REWINDOFF
  3-000000FC  CHAR  IO$_SEARCH                         3-00000024  CHAR  IO$_SEEK
  3-00000231  CHAR  IO$_SENSECHAR                      3-00000309  CHAR  IO$_SENSEMODE
  3-0000021D  CHAR  IO$_SETCHAR                        3-000003B8  CHAR  IO$_SETCLOCK
  3-00000088  CHAR  IO$_SETCLOCKP                      3-000002DD  CHAR  IO$_SETMODE
  3-000002ED  CHAR  IO$_SKIPFILE                       3-000002FA  CHAR  IO$_SKIPRECORD
  3-00000029  CHAR  IO$_SPACEFILE                      3-0000010E  CHAR  IO$_SPACERECORD
  3-000003D7  CHAR  IO$_STARTDATA                      3-000000B4  CHAR  IO$_STARTDATAP
  3-00000037  CHAR  IO$_STARTMPROC                     3-0000020F  CHAR  IO$_STARTSPNDL
  3-00000059  CHAR  IO$_STOP                           3-0000000D  CHAR  IO$_UNLOAD
  3-0000046B  CHAR  IO$_WRITEBUFNCRC                   3-0000011E  CHAR  IO$_WRITECHECK
  3-000001E4  CHAR  IO$_WRITECHECKH                    3-000003FF  CHAR  IO$_WRITECSR
  3-00000153  CHAR  IO$_WRITEHEAD                      3-000002A2  CHAR  IO$_WRITELBLK
  3-00000247  CHAR  IO$_WRITEMARK                      3-00000314  CHAR  IO$_WRITEOF
  3-0000012A  CHAR  IO$_WRITEPBLK                      3-000001C9  CHAR  IO$_WRITERET
  3-0000017E  CHAR IO$_WRITETRACKD                      3-00000326  CHAR IO$_WRITEVBLK
  3-00000448  CHAR IO$_WRITEWTHBUF                      3-00000257  CHAR IO$_WRTTMKR
 AP-0000004a L*1  LUN                                   3-000004A1  CHAR QIO_STRING


ARRAYS

    Address   Type  Name                  Bytes  Dimensions

  2-00000000  I*4   QIOCODE                 512  (0:1, 0:63)


LABELS

    Address   Label

      **      10


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name              Type  Name

        IRP$W_FUNC         I*4  LIB$EXTZV


COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:RKDISK/OBJ=OBJ$:RKDISK MSRC$:RKDISK

  /CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
  /F77  /NOG_FLOATING  /I4  /OPTIMIZE  /WARNINGS  /NOD_LINES  /NOCROSS_REFERENCE  /NOMACHINE_CODE  /CONTINUATIONS=19


COMPILATION STATISTICS

  Run Time:           11.39 seconds
  Elapsed Time:       27.02 seconds
  Page Faults:        371
  Dynamic Memory:     245 pages
```

0153 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


RECSELECT
LIS

RLDISK
LIS

PUDRIVER
LIS

PCL11T
LIS

ROLLUP
LIS

RXDISK
LIS

PCL11R
LIS

SB11
LIS

RKDISK
LIS