K 6

```
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
```

```
                                                                        B 14
**FILE**ID**BUGCHK
```

```
BBBBBBBB    UU      UU    GGGGGGGG    CCCCCCCC  HH      HH  KK      KK
BBBBBBBB    UU      UU    GGGGGGGG    CCCCCCCC  HH      HH  KK      KK
BB      BB  UU      UU  GG          CC          HH      HH  KK      KK
BB      BB  UU      UU  GG          CC          HH      HH  KK      KK
BB      BB  UU      UU  GG          CC          HH      HH  KK    KK
BB      BB  UU      UU  GG          CC          HH      HH  KK    KK
BBBBBBBB    UU      UU  GG          CC          HHHHHHHHHH  KKKKKK
BBBBBBBB    UU      UU  GG          CC          HHHHHHHHHH  KKKKKK
BB      BB  UU      UU  GG  GGGGGG  CC          HH      HH  KK    KK
BB      BB  UU      UU  GG  GGGGGG  CC          HH      HH  KK    KK
BB      BB  UU      UU  GG      GG  CC          HH      HH  KK      KK      ....
BB      BB  UU      UU  GG      GG  CC          HH      HH  KK      KK      ....
BBBBBBBB    UUUUUUUUUU    GGGGGG      CCCCCCCC  HH      HH  KK      KK      ....
BBBBBBBB    UUUUUUUUUU    GGGGGG      CCCCCCCC  HH      HH  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 BUGCHK (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 29-JUL-1979
C
C       Functional description:
C
C       Display for fatal,non-fatal and user bugchecks.
C
C       Modified by:
C
C       V03-002 SAR0066         Sharon A. Reynolds,     20-Jun-1983
C               Changed the carriage control in the 'format' statements
C               for use with ERF.
C
C       v03-001 BP0001          Brian Porter,           17-AUG-1982
C               Removed cpu dependant register for operater requested
C               shutdowns.
C**
C--

        INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'

        INCLUDE 'SRC$:SYECOM.FOR /NOLIST'

        INCLUDE 'SRC$:MODES.FOR /NOLIST'

        BYTE            LUN

        INTEGER         I
        INTEGER         J
        INTEGER         K

        INTEGER*2       MTH$IIAND

        INTEGER*4       FIELD

        INTEGER*4       FIELD1

        INTEGER*4       COMPRESSC

        INTEGER*4       BUGMSG_LENGTH

        LOGICAL*4       SYS$ASCTIM
        EXTERNAL        SYS$ASCTIM

        logical*4       sys$fao
        external        sys$fao

        REAL*8          TIME2

        CHARACTER*73    BUGMSG_TEXT

        character*16    process_logical_name

        CHARACTER*11    TIME1

        CHARACTER*3     V1STK_PTR(0:4)

        CHARACTER*4     V1VAX_ICCS(0:0)
        CHARACTER*9     V2VAX_ICCS(4:4)
        CHARACTER*17    V3VAX_ICCS(6:7)
        CHARACTER*6     V4VAX_ICCS(31:31)

        CHARACTER*3     GPR(0:14)

        CHARACTER*5     V1SYSREG_LABEL(0:7)

        CHARACTER*31    V1SYSREG_COMENT(0:7)

        CHARACTER*11    AST_MODES(0:3)
        CHARACTER*11    V1ASTLVL(0:3)
        CHARACTER*11    EXCEPT_ENABLE(6:9)

        EQUIVALENCE     (MODES,AST_MODES)
        EQUIVALENCE     (MODES,V1ASTLVL)
        EQUIVALENCE     (MODES,EXCEPT_ENABLE)

        PARAMETER       FATAL_BUG = 37

        PARAMETER       KSP = 16
        PARAMETER       ESP = 20
        PARAMETER       SSP = 24
        PARAMETER       USP = 28
        PARAMETER       ISP = 32

        PARAMETER       R0 = 36
        PARAMETER       R1 = 40
        PARAMETER       R2 = 44
        PARAMETER       R3 = 48
        PARAMETER       R4 = 52
        PARAMETER       R5 = 56
        PARAMETER       R6 = 60
        PARAMETER       R7 = 64
        PARAMETER       R8 = 68
        PARAMETER       R9 = 72
        PARAMETER       R10 = 76
        PARAMETER       R11 = 80
        PARAMETER       AP = 84
        PARAMETER       FP = 88
        PARAMETER       SP = 92
        PARAMETER       PC = 96

        PARAMETER       PSL = 100

        PARAMETER       P0BR = 104
        PARAMETER       P0LR = 108
        PARAMETER       P1BR = 112
        PARAMETER       P1LR = 116

        PARAMETER       SBR = 120
        PARAMETER       SLR = 124

        PARAMETER       PCBB = 128
        PARAMETER       SCBB = 132

        PARAMETER       ASTLVL = 136

        PARAMETER       SISR = 140

        PARAMETER       ICCS = 144
        PARAMETER       ICR = 148

        PARAMETER       TODR = 152

        PARAMETER       ACCS = 156

        PARAMETER       SBIFS = 160

        PARAMETER       TBDR = 160

        PARAMETER       SBISC = 164

        PARAMETER       CADR = 164

        PARAMETER       SBIMT = 168

        PARAMETER       MCESR = 168

        PARAMETER       SBIER = 172

        PARAMETER       CAER = 172

        PARAMETER       CMIER = 176

        PARAMETER       SBITA = 176

        PARAMETER       SILO = 180

        PARAMETER       BUGCHK_CODE = 104


        PARAMETER       PID = 108

        PARAMETER       LNAME = 112



        DATA    V1STK_PTR(0)    /'KSP'/

        DATA    V1STK_PTR(1)    /'ESP'/

        DATA    V1STK_PTR(2)    /'SSP'/

        DATA    V1STK_PTR(3)    /'USP'/

        DATA    V1STK_PTR(4)    /'ISP'/






        DATA    V1VAX_ICCS(0)   /'RUN*'/



        DATA    V2VAX_ICCS(4)   /'TRANSFER*'/




        DATA    V3VAX_ICCS(6)   /'INTERRUPT ENABLE*'/

        DATA    V3VAX_ICCS(7)   /'INTERRUPT*'/




        DATA    V4VAX_ICCS(31)  /'ERROR*'/




        DATA    GPR(0)          /'R0 '/

        DATA    GPR(1)          /'R1 '/

        DATA    GPR(2)          /'R2 '/

        DATA    GPR(3)          /'R3 '/

        DATA    GPR(4)          /'R4 '/

        DATA    GPR(5)          /'R5 '/

        DATA    GPR(6)          /'R6 '/

        DATA    GPR(7)          /'R7 '/

        DATA    GPR(8)          /'R8 '/

        DATA    GPR(9)          /'R9 '/

        DATA    GPR(10)         /'R10'/

        DATA    GPR(11)         /'R11'/

        DATA    GPR(12)         /'AP '/

        DATA    GPR(13)         /'FP '/

        DATA    GPR(14)         /'SP '/


        DATA    V1SYSREG_LABEL(0)/'P0BR*'/

        DATA    V1SYSREG_LABEL(1)/'P0LR*'/

        DATA    V1SYSREG_LABEL(2)/'P1BR*'/

        DATA    V1SYSREG_LABEL(3)/'P1LR*'/

        DATA    V1SYSREG_LABEL(4)/'SBR*'/

        DATA    V1SYSREG_LABEL(5)/'SLR*'/

        DATA    V1SYSREG_LABEL(6)/'PCBB*'/

        DATA    V1SYSREG_LABEL(7)/'SCBB*'/




        DATA    V1SYSREG_COMENT(0)/'P0 PTE BASE (VIRT ADDR)*'/

        DATA    V1SYSREG_COMENT(1)/'TOTAL P0 PAGES*'/

        DATA    V1SYSREG_COMENT(2)/'P1 PTE BASE (VIRT ADDR)*'/

        DATA    V1SYSREG_COMENT(3)/'TOTAL NON-EXISTENT P1 PAGES*'/

        DATA    V1SYSREG_COMENT(4)/'SYSTEM PTE BASE (PHY ADDR)*'/

        DATA    V1SYSREG_COMENT(5)/'TOTAL PAGES ''SYSTEM'' VIRT MEM*'/

        DATA    V1SYSREG_COMENT(6)/'PCB BASE (PHY ADDR)*'/

        DATA    V1SYSREG_COMENT(7)/'SCB BASE (PHY ADDR)*'/
        CALL FRCTOF (LUN)

        call header (lun)

        I = 0

        IF (EMB$W_HD_ENTRY .EQ. '25'X) THEN

        call logger (lun,'FATAL BUGCHECK')

        I = 140

        ELSE IF (EMB$W_HD_ENTRY .EQ. '28'X) THEN

        call logger (lun,'NON-FATAL BUGCHECK')

        ELSE IF (EMB$W_HD_ENTRY .EQ. '70'X) THEN

        call logger (lun,'USER BUGCHECK')
        ENDIF

        CALL CSTRING (EMB(I+BUGCHK_CODE),%REF(BUGMSG_TEXT),BUGMSG_LENGTH)

        call linchk (lun,2)

        if (bugmsg_length .ne. 0) then

        WRITE(LUN,10) (BUGMSG_TEXT(K:K),K = 1,BUGMSG_LENGTH)
10      FORMAT(/' ',<BUGMSG_LENGTH>A1)
        else

        WRITE(LUN,12) (EMB(I+BUGCHK_CODE+K),K = 1,0,-1)
12      FORMAT(/' ',T8,'BUGCHECK CODE',T28,2Z2.2)
        endif

        if (
        1 (emb$w_hd_entry .eq. '25'x
        1 .and.
        1 emb(244) .ne. '74'x)
        1 .or.
        1 emb$w_hd_entry .eq. '28'x
        1 .or.
        1 emb$w_hd_entry .eq. '70'x
        1 ) then

        if (sys$fao ('!AF',,process_logical_name,%val(15),
        1 %ref(emb(i+lname)))) then

        CALL LINCHK (LUN,2)

        WRITE(LUN,5) process_logical_name
```

```
5       FORMAT(/' ',t8,'PROCESS NAME',t24,a)
        endif

        CALL LINCHK (LUN,2)

        WRITE(LUN,20) (EMB(I+PID+K),K = 3,0,-1)
20      FORMAT(/' ',t8,'PROCESS ID',T24,4Z2.2)


        I = 0

        CALL LINCHK (LUN,2)

        WRITE(LUN,25) (EMB(I+PC+K),K = 3,0,-1)
25      FORMAT(/' ',T8,'ERROR PC',T24,4Z2.2)

        CALL VAXPSL (LUN,EMB(I+PSL))

        CALL LINCHK (LUN,4)

        write(lun,27) 'STACK POINTERS'
27      format(/' 'a,/)

        write(lun,30) (v1stk_ptr(j/4),(emb(ksp+i+j+k),k = 3,0,-1),j = 0,19,4)
30      format(' ',5(a3,' ',4z2.2,'  '))

        CALL LINCHK (LUN,6)

        write(lun,27) 'GENERAL REGISTERS'

        write(lun,40) ((gpr((j/4)+l*5),(emb(r0+i+j+k+(l*20)),k = 3,0,-1),
        1 j = 0,19,4),l = 0,2)
40      format(3(' ',5(a,' ',4z2.2,'  '),/))

        IF (EMB$W_HD_ENTRY .EQ. FATAL_BUG) I = 140

        CALL LINCHK (LUN,2)

        IF (EMB$W_HD_ENTRY .EQ. FATAL_BUG) THEN

        I = 0

        CALL LINCHK (LUN,3)

        WRITE(LUN,290)
290     FORMAT(/' ','SYSTEM REGISTERS',/)

        DO 300,J = P0BR,SCBB,4

        CALL LINCHK (LUN,2)

        FIELD = (J-P0BR)/4

        WRITE(LUN,295) V1SYSREG_LABEL(FIELD),(EMB(I+J+K),K = 3,0,-1),
        1 V1SYSREG_COMENT(FIELD)
295     FORMAT(' ',T8,A<COMPRESSC (V1SYSREG_LABEL(FIELD))>,T24,4Z2.2,/,
        1 T40,A<COMPRESSC (V1SYSREG_COMENT(FIELD))>)
```

```

      300     CONTINUE

              CALL LINCHK (LUN,2)

              WRITE(LUN,323) (EMB(I+ASTLVL+K),K = 3,0,-1)
      323     FORMAT(' ',T8,'ASTLVL',T24,4Z2.2)

              FIELD = LIB$EXTZV(0,3,EMB(I+ASTLVL))

              IF (FIELD .LE. 3) THEN

              WRITE(LUN,325) V1ASTLVL(FIELD)
      325     FORMAT(' ',T40,A<COMPRESSC (V1ASTLVL(FIELD))>,
              1 ' MODE AST PENDING')

              ELSE IF (FIELD .EQ. 4) THEN

              WRITE(LUN,327)
      327     FORMAT(' ',T40,'NO AST''S PENDING')
              ENDIF

              CALL LINCHK (LUN,1)

              WRITE(LUN,330) (EMB(I+SISR+K),K = 3,0,-1)
      330     FORMAT(' ',T8,'SISR',T24,4Z2.2)

              DO 340,J = 1,15

              FIELD = LIB$EXTZV(J,1,EMB(I+SISR))

              IF (FIELD .NE. 0) THEN

              CALL LINCHK (LUN,1)

              WRITE(LUN,335) J
      335     FORMAT(' ',T40,'PENDING INTERRUPT AT LEVEL ',Z1)
              ENDIF

      340     CONTINUE

              FIELD = LIB$EXTZV(16,5,EMB(I+SISR))

              CALL LINCHK (LUN,1)

              WRITE(LUN,343) FIELD
      343     FORMAT(' ',T40,'INTERRUPT REQUEST ACTIVE = ',I2,'.')

              CALL LINCHK (LUN,1)

              WRITE(LUN,345) (EMB(I+ICCS+K),K = 3,0,-1)
      345     FORMAT(' ',T8,'ICCS',T24,4Z2.2)

              CALL OUTPUT (LUN,EMB(I+ICCS),V1VAX_ICCS,0,0,0,'0')

              CALL OUTPUT (LUN,EMB(I+ICCS),V2VAX_ICCS,4,4,4,'0')

              CALL OUTPUT (LUN,EMB(I+ICCS),V3VAX_ICCS,6,6,7,'0')

              CALL OUTPUT (LUN,EMB(I+ICCS),V4VAX_ICCS,31,31,31,'0')

              CALL LINCHK (LUN,2)

              WRITE(LUN,350) (EMB(I+ICR+K),K = 3,0,-1)
      350     FORMAT(' ',T8,'ICR',T24,4Z2.2,/,T40,'INTERVAL COUNT REGISTER')

              CALL LINCHK (LUN,1)

              WRITE(LUN,355) (EMB(I+TODR+K),K = 3,0,-1)
      355     FORMAT(' ',T8,'TODR',T24,4Z2.2)

              IF (CP_11780) THEN

              CALL LINCHK (LUN,3)

              WRITE(LUN,400)
      400     FORMAT(/' ','CPU-SPECIFIC REGISTERS',/)

              CALL ACCS_780 (LUN,EMB(I+ACCS))

              CALL SBI_FAULTREG (LUN,EMB(I+SBIFS))

              CALL SBI_COMPARATOR (LUN,EMB(I+SBISC))

              CALL SBI_MAINTENANCE (LUN,EMB(I+SBIMT))

              CALL SBI_ERROR (LUN,EMB(I+SBIER))

      C
      C       'JIAND' IS CALLED DIRECTLY TO FOOL THE FORTRAN COMPILER INTO
      C       PASSING MORE THAN ONE ELEMENT OF THE BYTE ARRAY 'EMB'.
      C

              FIELD = MTH$JIAND(EMB(I+SBIER),'1000'X)

              IF (FIELD .NE. 0) THEN

              CALL SBI_TIMEOUT (LUN,EMB(I+SBITA))
              ELSE

              CALL LINCHK (LUN,1)

              WRITE(LUN,410) (EMB(I+SBITA+K),K = 3,0,-1)
      410     FORMAT(' ',T8,'SBITA',T24,4Z2.2)
              ENDIF

      C
      C       'JIAND' IS CALLED DIRECTLY TO FOOL THE FORTRAN COMPILER INTO
      C       PASSING MORE THAN ONE ELEMENT OF THE BYTE ARRAY 'EMB'.
      C

              FIELD = MTH$JIAND(EMB(I+SBISC),'A0000000'X)

              FIELD1 = MTH$JIAND(EMB(I+SBIFS),'10000'X)
```

```

        IF (FIELD .NE. 0 .OR. FIELD1 .NE. 0) THEN

        CALL LINCHK (LUN,3)

        WRITE(LUN,420)
420     FORMAT(/' ','SBI SILO LOCKED, DETAILED SUMMARY',/)

        DO 430,J = 0,60,4

        CALL SBI_SILO (LUN,EMB(I+SILO+J))

430     CONTINUE
        ENDIF

        ELSE IF (CP_11750) THEN

c
c       The cpu registers are not output for the 11/750 because
c       they are only valid after a machine check.
c

        ENDIF
        ENDIF
        endif

        RETURN



        ENTRY B_BUGCHK (LUN)


        call header (lun)

        I = 0

        IF (EMB$W_HD_ENTRY .EQ. '25'X) THEN

        call logger (lun,'FATAL BUGCHECK')

        I = 140

        ELSE IF (EMB$W_HD_ENTRY .EQ. '28'X) THEN

        call logger (lun,'NON-FATAL BUGCHECK')

        ELSE IF (EMB$W_HD_ENTRY .EQ. '70'X) THEN

        call logger (lun,'USER BUGCHECK')
        ENDIF


        CALL CSTRING (EMB(I+BUGCHK_CODE),%REF(BUGMSG_TEXT),BUGMSG_LENGTH)

```

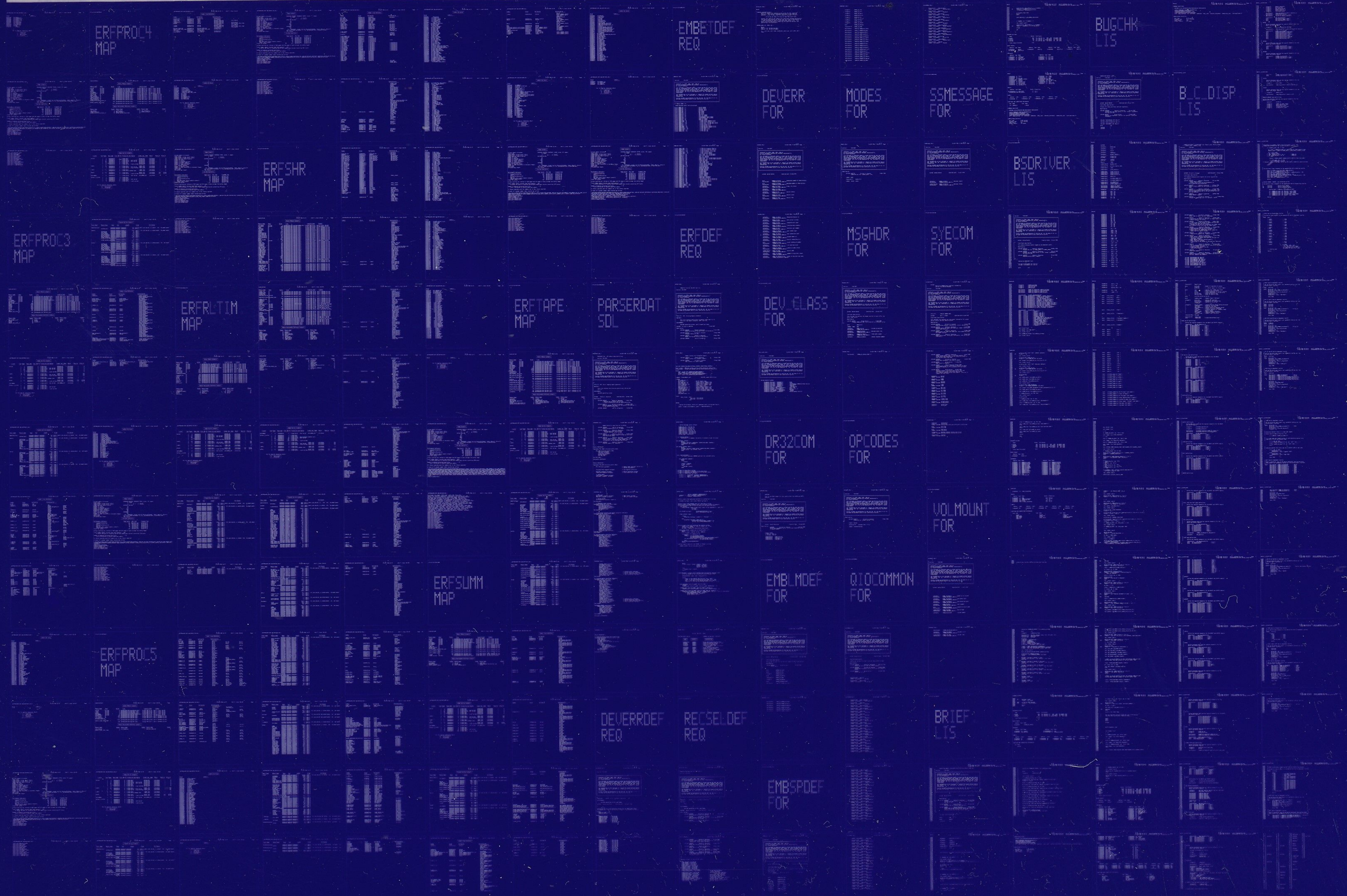
```
            IF (BUGMSG_LENGTH .NE. 0) THEN

            CALL LINCHK (LUN,2)

            WRITE(LUN,10) (BUGMSG_TEXT(K:K),K = 1,BUGMSG_LENGTH)
            else

            CALL LINCHK (LUN,2)

            WRITE(LUN,12) (EMB(I+BUGCHK_CODE+K),K = 1,0,-1)
            ENDIF

            RETURN

            END



PROGRAM SECTIONS

   Name                                 Bytes   Attributes

 0 $CODE                                 2668   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
 1 $PDATA                                 708   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
 2 $LOCAL                                1216   PIC CON REL LCL NOSHR NOEXE   RD   WRT QUAD
 3 EMB                                    512   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG
 4 SYECOM                                  44   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG
 5 MODE                                    55   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

   Total Space Allocated                 5203

ENTRY POINTS

   Address   Type  Name             Address   Type  Name

 0-00000000        BUGCHK         0-000008D5        B_BUGCHK

VARIABLES

   Address   Type  Name                                   Address   Type  Name

 2-00000214  I*4   BUGMSG_LENGTH                        2-00000191  CHAR BUGMSG_TEXT
 4-00000012  L*1   CP_11750                             4-00000011  L*1  CP_11780
 4-00000013  L*1   CP_117ZZ                             4-00000014  L*4  CRYPTK_FLAG
 4-0000000D  I*4   DEV_CHAR                             3-00000000  I*4  EMB$L_HD_SID
 3-00000004  I*2   EMB$W_HD_ENTRY                       3-0000000E  I*2  EMB$W_HD_ERRSEQ
 4-0000001E  L*1   END_VALUE                            4-0000001D  L*1  EOF_F[AG
 2-0000020C  I*4   FIE[D                                2-00000210  I*4  FIE[D1
 4-00000004  L*4   FORMS                                2-00000200  I*4  I
 2-00000204  I*4   J                                    2-00000208  I*4  K
 2-00000218  I*4   L                                    4-0000000C  L*1  LINES
 4-00000027  I*4   LSTLUN                              AP-00000004@ L*1  LUN
 4-0000001F  I*4   MAILBOX_CHANNEL                      2-000001F6  I*2  MTH$IIAND
 4-0000002B  CHAR  OPTIONS                              4-00000008  L*4  PRINTER
```

```
  2-000001DA  CHAR PROCESS_LOGICAL_NAME                     4-00000000  I*4  RECCNT
  4-00000023  I*4  RECORD_SIZE                              2-000001EA  CHAR TIME1
  2-000001F8  R*8  TIME2                                    4-00000019  L*1  VALID_CLASS
  4-0000001A  L*1  VALID_CPU                                4-0000001B  L*1  VALID_ENTRY
  4-0000001C  L*1  VALID_TYPE                               4-00000018  L*1  VOLUME_OUTPUT


ARRAYS

    Address   Type  Name                                      Bytes  Dimensions

  5-00000000  CHAR AST_MODES                                    44  (0:3)
  3-00000000  L*1  EMB                                         512  (0:511)
  3-00000006  I*4  EMB$Q_HD_TIME                                 8  (2)
  5-00000000  CHAR EXCEPT_ENABLE                                44  (6:9)
  2-00000044  CHAR GPR                                          45  (0:14)
  5-00000000  CHAR MODES                                        55  (0:4)
  5-00000000  CHAR V1ASTLVL                                     44  (0:3)
  2-00000000  CHAR V1STK_PTR                                    15  (0:4)
  2-00000099  CHAR V1SYSREG_COMENT                             248  (0:7)
  2-00000071  CHAR V1SYSREG_LABEL                               40  (0:7)
  2-0000000F  CHAR V1VAX_ICCS                                    4  (0:0)
  2-00000013  CHAR V2VAX_ICCS                                    9  (4:4)
  2-0000001C  CHAR V3VAX_ICCS                                   34  (6:7)
  2-0000003E  CHAR V4VAX_ICCS                                    6  (31:31)


LABELS

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

  1-000000B2   5'         1-00000089   10'        1-00000095   12'        1-000000CA   20'        1-000000E4   25'        1-000000FC   27'
  1-00000103   30'        1-0000011A   40'        1-00000135   290'       1-0000014D   295'           **        300        1-00000169   323'
  1-0000017E   325'       1-0000019D   327'       1-000001B5   330'       1-000001C8   335'           **        340        1-000001ED   343'
  1-00000215   345'       1-00000228   350'       1-00000256   355'       1-00000269   400'       1-00000287   410'       1-0000029B   420'
      **       430


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                                 Type  Name                                 Type  Name

        ACCS_780                             I*4   COMPRESSC                                  CSTRING
        FRCTOF                                     HEADER                               I*4   LIB$EXTZV
        LINCHK                                     LOGGER                               I*4   MTH$JIAND
        OUTPUT                                     SBI_COMPARATOR                             SBI_ERROR
        SBI_FAULTREG                               SBI_MAINTENANCE                            SBI_SILO
        SBI_TIMEOUT                          L*4   SYS$FAO                                    VAXPSL
```

```
COMMAND QUALIFIERS

  FORTRAN /LIS=LIS$:B  CHK/OBJ=OBJ$:BUGCHK MSRC$:BUGCHK

  /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:           8.53 seconds
  Elapsed Time:       21.13 seconds
  Page Faults:        235
  Dynamic Memory:     243 pages
```

0146 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



ERFPROC4
MAP

EMBETDEF
REQ

BUGCHK
LIS

DEVERR
FOR

MODES
FOR

SSMESSAGE
FOR

B_C_DISP
LIS

ERFSHR
MAP

BSDRIVER
LIS

ERFPROC3
MAP

ERFDEF
REQ

MSGHDR
FOR

SYECOM
FOR

ERFRLTIM
MAP

ERFTAPE
MAP

PARSERDAT
SDL

DEV_CLASS
FOR

DR32COM
FOR

OPCODES
FOR

VOLMOUNT
FOR

ERFSUMM
MAP

EMBLMDEF
FOR

QIOCOMMON
FOR

ERFPROC5
MAP

DEVERRDEF
REQ

RECSELDEF
REQ

BRIEF
LIS

EMBSPDEF
FOR