F 5

```
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
```

```
_$

Sy
--
CT
CT
CT
CT
CU
CU
CU
DC
DC
DC
DC
DC
DC
DE
DE
DE
DE
DE
DE
DE
DE
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DU
DU
DU
DU
DU
DU
DU
DU
DV
DV
DV
DY
DY
DY
DY
DY
ER
ES
```

```
**FILE**ID**CRASH                                                    H  6
```

```
  CCCCCCCC  RRRRRRRR      AAAAAA      SSSSSSSS  HH      HH
  CCCCCCCC  RRRRRRRR      AAAAAA      SSSSSSSS  HH      HH
CC          RR      RR  AA      AA  SS          HH      HH
CC          RR      RR  AA      AA  SS          HH      HH
CC          RR      RR  AA      AA  SS          HH      HH
CC          RR      RR  AA      AA  SS          HH      HH
CC          RRRRRRRR    AA      AA    SSSSSS    HHHHHHHHHH
CC          RRRRRRRR    AA      AA    SSSSSS    HHHHHHHHHH
CC          RR  RR      AAAAAAAAAA          SS  HH      HH
CC          RR  RR      AAAAAAAAAA          SS  HH      HH
CC          RR    RR    AA      AA          SS  HH      HH      ....
CC          RR    RR    AA      AA          SS  HH      HH      ....
  CCCCCCCC  RR      RR  AA      AA  SSSSSSSS    HH      HH      ....
  CCCCCCCC  RR      RR  AA      AA  SSSSSSSS    HH      HH      ....

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
```

Table of contents

```
0000          .TITLE  CRASH   DISPLAY CRASH RELATED INFORMATION
0000          .SBTTL  COPYRIGHT NOTICE
0000          .IDENT  'V04-000'
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY  *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
```

```
0000          .SBTTL  PROGRAM DESCRIPTION
0000     ;++
0000     ; FACILITY
0000     ;
0000     ;       SYSTEM DUMP ANALYZER
0000     ;
0000     ; ABSTRACT
0000     ;
0000     ;       THIS MODULE CONTAINS ROUTINES TO FORMAT INFORMATION
0000     ;       SAVED AT CRASH TIME.
0000     ;
0000     ; ENVIRONMENT
0000     ;
0000     ;       NATIVE MODE, USER MODE
0000     ;
0000     ; AUTHOR
0000     ;
0000     ;       TIM HALVORSEN, JULY 1978
0000     ;
0000     ; MODIFIED BY
0000     ;
0000     ;       V03-003 EMB0106         Ellen M. Batbouta       7-Jun-1984
0000     ;               Increase the max size of a file name from 64 bytes to
0000     ;               255 bytes.
0000     ;
0000     ;       V03-002 ROW0257         Ralph O. Weber          17-NOV-1983
0000     ;               Cause SHOW CRASH to display the SCS node name, if one exists.
0000     ;
0000     ;       V03-001 TCM0001         Trudy C. Matthews       27-Jul-1983
0000     ;               Added comment saying why the CPUDISP macro can't be used for
0000     ;               some CPU-dependent code in this module.
0000     ;
0000     ;--
```

```
0000          .SBTTL  DECLARATIONS
0000    ;
0000    ;       SYMBOL DEFINTIONS
0000    ;
0000            $DMPDEF                         ; DUMP FILE DEFINITIONS
0000            $EMBDEF <CR,ET,HD>              ; CRASHDUMP ERROR LOG ENTRY
0000            $ERLDEF                         ; ERROR LOG DEFINITIONS
0000            $PCBDEF                         ; PROCESS CONTROL BLOCK
0000            $PHDDEF                         ; PROCESS HEADER DEFINITIONS
0000            $IFDDEF                         ; IMAGE FILE DESCRIPTOR
0000            $PSLDEF                         ; PROGRAM STATUS LONGWORD
0000            $SBDEF                          ; SYSTEM BLOCK
```

```
               0000          .SBTTL  STORAGE DEFINITIONS
               0000
               0000     ;
               0000     ;    STORAGE DEFINITIONS
               0000     ;
               0000
           00000000          .PSECT  SDADATA,NOEXE,WRT
               0000
               0000     ERLPTR::
 00000008      0000          .BLKL   2                      ; ADDRESS OF ERROR LOG ENTRY
               0008
               0008
           00000000          .PSECT  CRASH,EXE,NOWRT,LONG
               0000
               0000          .DEFAULT DISPLACEMENT,LONG
```

```
          0000            .SBTTL  READ-ONLY DATA DEFINITIONS
          0000
          0000    ;
          0000    ;       READ-ONLY DATA DEFINITIONS
          0000    ;
          0000
          0000    ;
          0000    ;       TABLE OF BUGCHECK CODES WHICH CAUSE THE CONSOLE TO
          0000    ;       WIPE OUT MOST OF THE REGISTERS.  THIS TABLE IS TAKEN
          0000    ;       DIRECTLY FROM THE CODE IN POWERFAIL.
          0000    ;
          0000    RESTART_BUGS:
00000000' 0000            .LONG   BUG$_IVLISTK            ; INVALID INTERRUPT STACK
00000000' 0004            .LONG   BUG$_DBLERR             ; DOUBLE ERROR HALT
00000000' 0008            .LONG   BUG$_HALT               ; HALT INSTRUCTION
00000000' 000C            .LONG   BUG$_ILLVEC             ; ILLEGAL VECTOR CODE
00000000' 0010            .LONG   BUG$_NOUSRWCS           ; NO USER WCS FOR VECTOR
00000000' 0014            .LONG   BUG$_ERRHALT            ; ERROR PENDING ON HALT
00000000' 0018            .LONG   BUG$_CHMONIS            ; CHM ON INTERRUPT STACK
00000000  001C            .LONG   0                       ; --- END OF TABLE
```

```
                                0020           .SBTTL  DISPLAY_CRASH -- DISPLAY CRASH INFORMATION
                                0020     ;---
                                0020     ;
                                0020     ;         DISPLAY_CRASH
                                0020     ;
                                0020     ;         THIS ROUTINE DISPLAYS ALL RELATED INFORMATION REGARDING
                                0020     ;         THE SAVED STATE OF THE PROCESSOR AT THE TIME OF THE
                                0020     ;         SYSTEM BUGCHECK EXCEPTION.
                                0020     ;
                                0020     ;    INPUTS:
                                0020     ;
                                0020     ;         NONE
                                0020     ;
                                0020     ;    OUTPUTS:
                                0020     ;
                                0020     ;         NONE
                                0020     ;
                                0020     ;---
                                0020
                                0020           .ENABL  LSB
                                0020
                          020C  0020           .ENTRY  DISPLAY_CRASH,-
                                0022           ^M<R2,R3,R9>
                                0022
                                0022           SUBHD   <System crash information>       ; SET NEW HEADING
                                002F           SKIP    PAGE
                                0036
        59   00000000'EF    D0  0036           MOVL    ERLPTR,R9                  ; ADDRESS OF ERROR LOG ENTRY
                                003D           ALLOC   24,R2                      ; ALLOCATE SPACE FOR DATE/TIME
                                0047           $ASCTIM_S TIMADR=EMB$Q_CR_TIME(R9),TIMBUF=(R2)
                       52   DD  0057           PUSHL   R2
                                0059           PRINT   1,<Time of system crash: !AS>
                                0066
                                0066           SKIP    2
                                006F           GETMEM  @SYS$GQ_VERSION,-(SP)    ; GET SYSTEM VERSION
                       5E   DD  007F           PUSHL   SP                         ; ADDRESS OF STRING
                       04   DD  0081           PUSHL   #4                         ; LENGTH OF STRING
                                0083           PRINT   3,<Version of system: VAX/VMS VERSION !AD>
        53   00000000'GF    D0  0090           MOVL    G^SCS$GA_LOCALSB, R3       ; Get address of local system block
                  5E   10   C2  0097           SUBL    #SB$S_NODENAME, SP         ; Make scratch space for node name
                  52   5E   D0  009A           MOVL    SP, R2                     ; Save address of scratch
                                009D           GETMEM  SB$T_NODENAME(R3), -       ; Get node name
                                009D                   (R2), #SB$S_NODENAME
                       62   95  00AB           TSTB    (R2)                       ; Is the node name null?
                       18   13  00AD           BEQL    15$                        ; Branch if null node name
                                00AF           SKIP    2
                       52   DD  00B8           PUSHL   R2                         ; Push node name copy address
                                00BA           PRINT   1,<VAXcluster node name: !AC>
                                00C7     15$:
                                00C7
   50   00F4 C9    FD 8F    78  00C7           ASHL    #-3,EMB$L_CR_CODE(R9),R0         ; MESSAGE NUMBER
                       28   13  00CE           BEQL    25$                        ; SKIP IF NO MESSAGE
        51   00000000'EF    9E  00D0           MOVAB   L^BUG$T_MESSAGES,R1        ; ADDRESS OF MESSAGES
                                00D7     20$:
                  52   81   9A  00D7           MOVZBL  (R1)+,R2                   ; LENGTH OF MESSAGE
                  51   52   C0  00DA           ADDL2   R2,R1                      ; SKIP TO NEXT MESSAGE
                  F7   50   F5  00DD           SOBGTR  R0,20$                     ; LOOP UNTIL FOUND
```

```
                    51   DD  00E0            PUSHL   R1                           ; ADDRESS OF BUGCHECK MESSAGE
                             00E2            SKIP    2
                             00EB            PRINT   1,<Reason for BUGCHECK exception: !AC>
                             00F8
                             00F8   25$:
                             00F8            GETMEM  @SCH$GL_CURPCB               ; GET CURRENT PROCESS'S PCB
                 30 50   E9  0105            BLBC    R0,26$                       ; BRANCH IF DATA NOT AVAILABLE
                    7E   7C  0108            CLRQ    -(SP)                        ; INITIALIZE A BUFFER TO HOLD
                    7E   7C  010A            CLRQ    -(SP)                        ; THE CURRENT PROCESS'S NAME
              52    5E   D0  010C            MOVL    SP,R2                        ; REMEMBER ADDRESS OF BUFFER
                             010F            GETMEM  PCB$T_LNAME(R1),(R2),#16     ; GET CURRENT PROCESS'S NAME
                    52   DD  011D            PUSHL   R2                           ; PROCESS NAME
                             011F            SKIP    2
                             0128            PRINT   1,<Process currently executing: !AC>
              5E    10   C0  0135            ADDL    #16,SP                       ; CLEAN BUFFER OFF STACK
                             0138   26$:
                             0138
       00000000'EF  00   FB  0138            CALLS   #0,CURPROC                   ; SET TO CURRENT PROCESS
       51   00000000'EF  D0  013F            MOVL    MMG$IMGHDRBUF,R1             ; ADDRESS OF HEADER BUFFER
                             0146            TRYMEM  4(R1),R2                     ; GET ADDRESS OF IFD
                 4B 50   E9  0153            BLBC    R0,30$                       ; IF NOT AVAILABLE
                             0156            TRYMEM  IFD$W_FILNAMOFF(R1)          ; GET OFFSET TO FILE NAME
                 3E 50   E9  0160            BLBC    R0,30$                       ; BRANCH IF NOT AVAILABLE
              51    51   32  0163            CVTWL   R1,R1                        ; CONVERT TO LONGWORD
       5E   000000FF 8F  C2  0166            SUBL    #255,SP                      ; ALLOCATE BUFFER FOR FILESPEC
              50    5E   D0  016D            MOVL    SP,R0
                             0170            TRYMEM  (R2)[R1],(R0),#255           ; GET ASCIC IMAGE FILE NAME
                 1C 50   E9  0182            BLBC    R0,30$                       ; BRANCH IF NOT AVAILABLE
                    5E   DD  0185            PUSHL   SP
                             0187            SKIP    2
                             0190            PRINT   1+<255/4>,<Current image file: !AC>
                             01A1
                             01A1   30$:
   7E   64 A9   05  10   EF  01A1            EXTZV   #PSL$V_IPL,#PSL$S_IPL,EMB$L_CR_PSL(R9),-(SP)
                             01A7            SKIP    2
                             01B0            PRINT   1,<Current IPL: !UL  (decimal)>
                             01BD
                             01BD            SKIP    5
                             01C6            ENSURE  7
                             01DE            PRINT   0,<General registers:>
                             01EB            SKIP    1
                             01F4   ;
                             01F4   ;        IF THE BUGCHECK CODE WAS ANY OF THE ONES THAT COME
                             01F4   ;        BACK FROM THE CONSOLE ROM VIA POWERFAIL RESTART, THEN
                             01F4   ;        DO NOT PRINT THE REGISTERS AS THEY HAVE BEEN WIPED OUT.
                             01F4   ;
     50   00F4 C9   07   CB  01F4            BICL3   #7,EMB$L_CR_CODE(R9),R0      ; BUGCHECK CODE (CLEAR FLAG)
          51   FE02 CF   DE  01FA            MOVAL   RESTART_BUGS,R1              ; TABLE OF RESTART BUGCHECKS
                             01FF   40$:
              81    50   D1  01FF            CMPL    R0,(R1)+                     ; CHECK IF MATCHES
                    06   13  0202            BEQL    45$                          ; BRANCH IF SO
                    61   D5  0204            TSTL    (R1)                         ; END OF TABLE?
                    F7   12  0206            BNEQ    40$                          ; CONTINUE UNTIL DONE
                    7C   11  0208            BRB     50$                          ; PRINT REGISTERS
                             020A   45$:
                 50 A9   DD  020A            PUSHL   EMB$L_CR_R11(R9)             ; PSL
                 4C A9   DD  020D            PUSHL   EMB$L_CR_R10(R9)             ; PC
                 0210          PRINT   2,<!_PC  = !XL   PSL = !XL>
                 021D          SKIP    1
                 0226          PRINT   0,<!_Remaining registers not available -- wiped out by console>
                 0233          SKIP    1
                 023C          PRINT   0,<Processor registers:>
                 0249          SKIP    1
0084 C9    DD    0252          PUSHL   EMB$L_CR_SCBB(R9)
  7C A9    DD    0256          PUSHL   EMB$L_CR_SLR(R9)
  78 A9    DD    0259          PUSHL   EMB$L_CR_SBR(R9)
                 025C          PRINT   1,<!_SBR     = !XL>
                 0269          PRINT   1,<!_SLR     = !XL>
                 0276          PRINT   1,<!_SCBB    = !XL>
   021F    31    0283          BRW     60$                        ; PRINT KSP-ISP REGISTERS
                 0286     50$:
  30 A9    DD    0286          PUSHL   EMB$L_CR_R3(R9)
  2C A9    DD    0289          PUSHL   EMB$L_CR_R2(R9)
  28 A9    DD    028C          PUSHL   EMB$L_CR_R1(R9)
  24 A9    DD    028F          PUSHL   EMB$L_CR_R0(R9)
                 0292          PRINT   4,<!_R0  = !XL   R1  = !XL   R2  = !XL   R3  = !XL>
  40 A9    DD    029F          PUSHL   EMB$L_CR_R7(R9)
  3C A9    DD    02A2          PUSHL   EMB$L_CR_R6(R9)
  38 A9    DD    02A5          PUSHL   EMB$L_CR_R5(R9)
  34 A9    DD    02A8          PUSHL   EMB$L_CR_R4(R9)
                 02AB          PRINT   4,<!_R4  = !XL   R5  = !XL   R6  = !XL   R7  = !XL>
  50 A9    DD    02B8          PUSHL   EMB$L_CR_R11(R9)
  4C A9    DD    02BB          PUSHL   EMB$L_CR_R10(R9)
  48 A9    DD    02BE          PUSHL   EMB$L_CR_R9(R9)
  44 A9    DD    02C1          PUSHL   EMB$L_CR_R8(R9)
                 02C4          PRINT   4,<!_R8  = !XL   R9  = !XL   R10 = !XL   R11 = !XL>
  60 A9    DD    02D1          PUSHL   EMB$L_CR_PC(R9)
  5C A9    DD    02D4          PUSHL   EMB$L_CR_SP(R9)
  58 A9    DD    02D7          PUSHL   EMB$L_CR_FP(R9)
  54 A9    DD    02DA          PUSHL   EMB$L_CR_AP(R9)
                 02DD          PRINT   4,<!_AP  = !XL   FP  = !XL   SP  = !XL   PC  = !XL>
  64 A9    DD    02EA          PUSHL   EMB$L_CR_PSL(R9)
                 02ED          PRINT   1,<!_PSL = !XL>
                 02FA          SKIP    4
                 0303          ENSURE  10
                 031B          PRINT   0,<Processor registers:>
                 0328          SKIP    1
                 0331     ;
                 0331     ; We can't use the CPUDISP macro here because we have to get the EXE$GB_CPUTYPE
                 0331     ; value from the appropriate dump file.
                 0331     ;
                 0331          GETMEM  @EXE$GB_CPUTYPE            ; GET TYPE OF CPU
  09 50    E9    033E          BLBC    R0,780$                    ; IF NOT FOUND, ASSUME 11/780
                 0341          CASE    R1,TYPE=B,-                ; DISPATCH ON CPU TYPE
                 0341                  LIMIT=#PR$_SID_TYP780,-
                 0341                  <780$,-                    ; 11/780
                 0341                  750$>                      ; 11/750
                 034A                                             ; ALL OTHERS USE 11/780
                 034A     ;
                 034A     ;            11/780 INTERNAL REGISTERS
                 034A     ;
0090 C9    DD    034A     780$:  PUSHL   EMB$L_CR_ACCS(R9)
0080 C9    DD    034E          PUSHL   EMB$L_CR_PCBB(R9)
  68 A9    DD    0352          PUSHL   EMB$L_CR_P0BR(R9)
```

```
                 0355             PRINT   3,<!_P0BR   = !XL    PCBB   = !XL     ACCS   = !XL>
00A0 C9     DD   0362             PUSHL   EMB$L_CR_SBIFS(R9)
0084 C9     DD   0366             PUSHL   EMB$L_CR_SCBB(R9)
  6C A9     DD   036A             PUSHL   EMB$L_CR_P0LR(R9)
                 036D             PRINT   3,<!_P0LR   = !XL    SCBB   = !XL     SBIFS  = !XL>
00A4 C9     DD   037A             PUSHL   EMB$L_CR_SBISC(R9)
0088 C9     DD   037E             PUSHL   EMB$L_CR_ASTLVL(R9)
  70 A9     DD   0382             PUSHL   EMB$L_CR_P1BR(R9)
                 0385             PRINT   3,<!_P1BR   = !XL    ASTLVL = !XL     SBISC  = !XL>
00A8 C9     DD   0392             PUSHL   EMB$L_CR_SBIMT(R9)
008C C9     DD   0396             PUSHL   EMB$L_CR_SISR(R9)
  74 A9     DD   039A             PUSHL   EMB$L_CR_P1LR(R9)
                 039D             PRINT   3,<!_P1LR   = !XL    SISR   = !XL     SBIMT  = !XL>
00AC C9     DD   03AA             PUSHL   EMB$L_CR_SBIER(R9)
0090 C9     DD   03AE             PUSHL   EMB$L_CR_ICCS(R9)
  78 A9     DD   03B2             PUSHL   EMB$L_CR_SBR(R9)
                 03B5             PRINT   3,<!_SBR    = !XL    ICCS   = !XL     SBIER  = !XL>
00B0 C9     DD   03C2             PUSHL   EMB$L_CR_SBITA(R9)
0094 C9     DD   03C6             PUSHL   EMB$L_CR_ICR(R9)
  7C A9     DD   03CA             PUSHL   EMB$L_CR_SLR(R9)
                 03CD             PRINT   3,<!_SLR    = !XL    ICR    = !XL     SBITA  = !XL>
00B4 C9     DD   03DA             PUSHL   EMB$L_CR_SBIS(R9)
0098 C9     DD   03DE             PUSHL   EMB$L_CR_TODR(R9)
                 03E2             PRINT   2,<!_                   TODR   = !XL     SBIS   = !XL>
                 03EF             SKIP    1
   00AA     31   03F8             BRW     60$
                 03FB       ;
                 03FB       ;     11/750 INTERNAL REGISTERS
                 03FB       ;
009C C9     DD   03FB  750$:      PUSHL   EMB$L_CR_ACCS(R9)
0080 C9     DD   03FF             PUSHL   EMB$L_CR_PCBB(R9)
  68 A9     DD   0403             PUSHL   EMB$L_CR_P0BR(R9)
                 0406             PRINT   3,<!_P0BR   = !XL    PCBB   = !XL     ACCS   = !XL>
00A0 C9     DD   0413             PUSHL   EMB$L_CR_TBDR(R9)
0084 C9     DD   0417             PUSHL   EMB$L_CR_SCBB(R9)
  6C A9     DD   041B             PUSHL   EMB$L_CR_P0LR(R9)
                 041E             PRINT   3,<!_P0LR   = !XL    SCBB   = !XL     TBDR   = !XL>
00A4 C9     DD   042B             PUSHL   EMB$L_CR_CADR(R9)
0088 C9     DD   042F             PUSHL   EMB$L_CR_ASTLVL(R9)
  70 A9     DD   0433             PUSHL   EMB$L_CR_P1BR(R9)
                 0436             PRINT   3,<!_P1BR   = !XL    ASTLVL = !XL     CADR   = !XL>
00A8 C9     DD   0443             PUSHL   EMB$L_CR_MCESR(R9)
008C C9     DD   0447             PUSHL   EMB$L_CR_SISR(R9)
  74 A9     DD   044B             PUSHL   EMB$L_CR_P1LR(R9)
                 044E             PRINT   3,<!_P1LR   = !XL    SISR   = !XL     MCESR  = !XL>
00AC C9     DD   045B             PUSHL   EMB$L_CR_CAER(R9)
0090 C9     DD   045F             PUSHL   EMB$L_CR_ICCS(R9)
  78 A9     DD   0463             PUSHL   EMB$L_CR_SBR(R9)
                 0466             PRINT   3,<!_SBR    = !XL    ICCS   = !XL     CAER   = !XL>
00B0 C9     DD   0473             PUSHL   EMB$L_CR_CMIERR(R9)
0094 C9     DD   0477             PUSHL   EMB$L_CR_ICR(R9)
  7C A9     DD   047B             PUSHL   EMB$L_CR_SLR(R9)
                 047E             PRINT   3,<!_SLR    = !XL    ICR    = !XL     CMIERR = !XL>
0098 C9     DD   048B             PUSHL   EMB$L_CR_TODR(R9)
                 048F             PRINT   1,<!_                   TODR   = !XL>
                 049C             SKIP    1
                 04A5       60$:
```

```
20 A9   DD  04A5    341          PUSHL   EMB$L_CR_ISP(R9)
            04A8    342          PRINT   1,<!_ISP      = !XL>
1C A9   DD  04B5    343          PUSHL   EMB$L_CR_USP(R9)
18 A9   DD  04B8    344          PUSHL   EMB$L_CR_SSP(R9)
14 A9   DD  04BB    345          PUSHL   EMB$L_CR_ESP(R9)
10 A9   DD  04BE    346          PUSHL   EMB$L_CR_KSP(R9)
            04C1    347          PRINT   1,<!_KSP      = !XL>
            04CE    348          PRINT   1,<!_ESP      = !XL>
            04DB    349          PRINT   1,<!_SSP      = !XL>
            04E8    350          PRINT   1,<!_USP      = !XL>
        04  04F5    351          RET
```

```
                            04F6         .SBTTL  PRINT_HEADER -- PRINT THE DUMP FILE HEADER BLOCKS
                            04F6  ;---
                            04F6  ;
                            04F6  ;       PRINT_HEADER
                            04F6  ;
                            04F6  ;       DUMP THE CONTENTS OF THE DUMP FILE HEADER BLOCKS
                            04F6  ;
                            04F6  ; INPUTS:
                            04F6  ;
                            04F6  ;       DUMP_HEADER = DUMP FILE HEADER (3 BLOCKS)
                            04F6  ;
                            04F6  ; OUTPUTS:
                            04F6  ;
                            04F6  ;       NONE
                            04F6  ;
                            04F6  ;---
                            04F6
                            04F6         .ENABL  LSB
                            04F6
                            04F6  PRINT_HEADER::
                      0004  04F6         .WORD   ^M<R2>
                            04F8
                            04F8         SUBHD   <Dump file header>
                            0505         SKIP    PAGE
52  00000000'EF       DE    050C         MOVAL   DUMP_HEADER,R2                  ; STARTING ADDRESS
53  00000600 8F       D0    0513         MOVL    #3*5T2,R3                       ; LENGTH TO DUMP
                            051A  10$:
                52    DD    051A         PUSHL   R2                              ; DUMP ADDRESS
                52    DD    051C         PUSHL   R2                              ; ADDRESS OF ASCII STRING
                20    DD    051E         PUSHL   #32                             ; LENGTH OF STRING
           50   08    D0    0520         MOVL    #8,R0                           ; REPEAT COUNT
                            0523  20$:
                82    DD    0523         PUSHL   (R2)+
             FB 50    F5    0525         SOBGTR  R0,20$                          ; PUSH NEXT 8 LONGWORDS
                            0528         PRINT   11,<!XL !XL !XL !XL !XL !XL !XL !XL  !AF      !XL>
           53   20    C2    0535         SUBL2   #32,R3                          ; DECREMENT LENGTH
                E0    14    0538         BGTR    10$                             ; CONTINUE UNTIL DONE
                      04    053A         RET
                            053B
                            053B         .DSABL  LSB
```

```
                                        053B          .SBTTL  GET_DUMP_INFO -- GET DUMP HEADER INFO
                                        053B     ;---
                                        053B     ;
                                        053B     ;        GET_DUMP_INFO
                                        053B     ;
                                        053B     ;        FIND THE ERROR LOG ENTRY AND SAVE IT.  IF THE
                                        053B     ;        ENTRY CANNOT BE FOUND, GENERATE A DUMMY ENTRY.
                                        053B     ;
                                        053B     ;    INPUTS:
                                        053B     ;
                                        053B     ;        NONE
                                        053B     ;
                                        053B     ;    OUTPUTS:
                                        053B     ;
                                        053B     ;        ERLPTR = ADDRESS OF ERROR LOG ENTRY
                                        053B     ;
                                        053B     ;---
                                        053B          .ENABL  LSB
                                        053B
                                        053B     GET_DUMP_INFO::
                                  0304  053B          .WORD   ^M<R2,R8,R9>
                                        053D
                                        053D     ;
                                        053D     ;        SET FLAGS DESCRIBING THE VERSION OF THE SYSTEM
                                        053D     ;
                   00000000'EF     D4   053D          CLRL    VERSION_FLAGS                 ; PRESET FLAGS LONGWORD
                                        0543          GETMEM  @SYS$GQ_VERSION               ; READ THE SYSTEM VERSION
              2E31 8F      51      B1   0550          CMPW    R1,#^A'T.'                    ; 1.0 EXECUTIVE? (1.01,1.5 ALSO)
                           07      13   0555          BEQL    2$                            ; BRANCH IF NOT
             00000000'EF   01      D0   0557          MOVL    #1,VERSION_FLAGS              ; INDICATE VERSION 2
                                        055E     2$:
          03 00000000'EF           E9   055E          BLBC    CURRENT_SYSTEM,3$             ; BRANCH IF EXAMINING DUMP FILE
                         009C      31   0565          BRW     30$                           ; SETUP FOR CURRENT SYSTEM
                                        0568     3$:
00000000'EF   0000006C'EF          9E   0568          MOVAB   DUMP_HEADER+DMP$L_CRASHERL,ERLPTR ; SET PTR TO ERROR LOG ENTRY
         02   00000006'EF          B1   0573          CMPW    DUMP_HEADER+DMP$W_DUMPVER,#2 ; IS ERR MSG HDR AT NEG OFFSETS?
                           07      19   057A          BLSS    1$                            ; BR IF NO, (VMS V2 FORMAT)
             00000000'EF   04      C0   057C          ADDL    #EMB$K_LENGTH,ERLPTR          ; POINT PAST THE ERRMSG HDR (VMS V3)
                                        0583     1$:
          59  00000000'EF          D0   0583          MOVL    ERLPTR,R9                     ; ADDRESS OF ERROR LOG ENTRY
                                        058A     ;
                                        058A     ;        THE FOLLOWING CODE ADJUSTS THE SAVED SP SINCE IT WAS
                                        058A     ;        NOT SAVED UNTIL AFTER THE PC,PSL WAS ADDED TO THE STACK.
                                        058A     ;
                  5C A9    08      C0   058A     4$:  ADDL2   #2*4,EMB$L_CR_SP(R9)          ; ADJUST DUE TO BUGCHECK
                                        058E     ;
                                        058E     ;        STORE THE CURRENT SP (R14) INTO THE CORRESPONDING PROCESSOR
                                        058E     ;        REGISTER CORRESPONDING TO THE ACCESS MODE ONLY IF KERNEL OR
                                        058E     ;        INTERRUPT STACK.  IF OTHER STACK, THEN USE THE PROCESSOR
                                        058E     ;        REGISTER BECAUSE BUGCHECK HAS TO MOVE THE INFO TO THE KERNEL
                                        058E     ;        MODE STACK AND IN THE PROCESS, WIPES THE SP REGISTER.
                                        058E     ;
        51   64 A9    02   18      EF   058E          EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,EMB$L_CR_PSL(R9),R1
                           08      13   0594          BEQL    10$                           ; BRANCH IF KERNEL/INTERRUPT
           5C A9    10 A941        D0   0596          MOVL    EMB$L_CR_KSP(R9)[R1],EMB$L_CR_SP(R9) ; FIX SP VALUE
                           12      11   059C          BRB     6$
             08 64 A9      1A      E0   059E     10$: BBS     #PSL$V_IS,EMB$L_CR_PSL(R9),5$   ; BRANCH IF ISP
```

```
     10 A941   5C A9   D0  05A3         MOVL    EMB$L_CR_SP(R9),EMB$L_CR_KSP(R9)[R1]     ; R14 TO PR[MODE+1]
                  05   11  05A9         BRB     6$
       20 A9   5C A9   D0  05AB    5$:  MOVL    EMB$L_CR_SP(R9),EMB$L_CR_ISP(R9)         ; R14 TO PR[0]
                           05B0    6$:
                           05B0        ;
                           05B0        ;        SIMULATE A SVPCTX INSTRUCTION IN ORDER TO MAKE THE HARDWARE
                           05B0        ;        PCB IN THE PROCESS HEADER CORRESPOND TO THE SAVED REGISTERS
                           05B0        ;        AT THE TIME OF THE CRASH.
                           05B0        ;
                           05B0         GETMEM  @SCH$GL_CURPCB              ; ADDRESS OF CURRENT PCB
               43 50   E9  05BD         BLBC    R0,21$                      ; BRANCH IF CANNOT READ
                           05C0         GETMEM  PCB$L_PHD(R1),R2            ; GET ADDRESS OF CURRENT PHD
               33 50   E9  05CD         BLBC    R0,21$                      ; BRANCH IF CANNOT READ
                  52   D5  05D0         TSTL    R2                          ; PHD VALID?
                  2F   13  05D2         BEQL    21$                         ; SKIP IF NOT
                           05D4         PUTMEM  PHD$L_KSP(R2),EMB$L_CR_KSP(R9),#4*4 ; SET KSP - USP
                           05E3         PUTMEM  PHD$L_R0(R2),EMB$L_CR_R0(R9),#14*4 ; SET R0 - R13
                           05F3         PUTMEM  PHD$L_PC(R2),EMB$L_CR_PC(R9),#2*4 ; SET PC,PSL
                       04  0603   21$:  RET
                           0604
                           0604        ;
                           0604        ;        EXAMINE CURRENT RUNNING SYSTEM -- CREATE A DUMMY
                           0604        ;        ERROR LOG ENTRY AND DUMP HEADER.
                           0604        ;
       58   00000000'EF    9E  0604   30$:  MOVAB   DUMP_HEADER,R8              ; ADDRESS OF DUMP HEADER
          00 04 A8    00   E2  060B         BBSS    #DMP$V_OLDDUMP,DMP$L_FLAGS(R8),32$ ; FLAG DUMP ANALYZED
            0000010C 8F    DD  0610   32$:  PUSHL   #EMB$C_CR_LENGTH            ; LENGTH OF CRASH LOG ENTRY
    00000000'EF       01   FB  0616         CALLS   #1,ALLOCATE                 ; ALLOCATE STORAGE
    00000000'EF       51   D0  061D         MOVL    R1,ERLPTR                   ; SAVE ADDRESS OF ERL BUFFER
             59       51   D0  0624         MOVL    R1,R9                       ; REFERENCE OFF R9
                           0627         GETMEM  @EXE$GQ_SYSTIME,EMB$Q_CR_TIME(R9),#8 ; SET DATE/TIME
                           0639         GETMEM  @MMG$GL_SPTLEN,R2           ; LENGTH OF SPT IN LONGWORDS
                           0649         GETMEM  @MMG$GL_SBR                 ; PHYSICAL ADDRESS OF SPT
             51     6142   DE  0656         MOVAL   (R1)[R2],R1                 ; COMPUTE PHYSICAL MEMORY SIZE
00000000'EF  51    F7 8F   78  065A         ASHL    #-9,R1,PHYS_PAGES           ; SAVE MEMORY SIZE IN PAGES
                           0663         GETMEM  @SCH$GL_CURPCB,R2           ; GET ADDRESS OF CURRENT PCB
               1F 50   E9  0673         BLBC    R0,35$                      ; BRANCH IF ERROR
                           0676         GETMEM  PCB$L_PID(R2),EMB$L_CR_PID(R9),#4 ; GET PID OF CURRENT PROCESS
                           0685         GETMEM  PCB$T_LNAME(R2),EMB$T_CR_LNAME(R9),#16 ; AND ALSO NAME
                           0695   35$:  STATUS  SUCCESS
                       04  069C         RET
                           069D
                           069D         .DSABL  LSB
                           069D
                           069D         .END
```

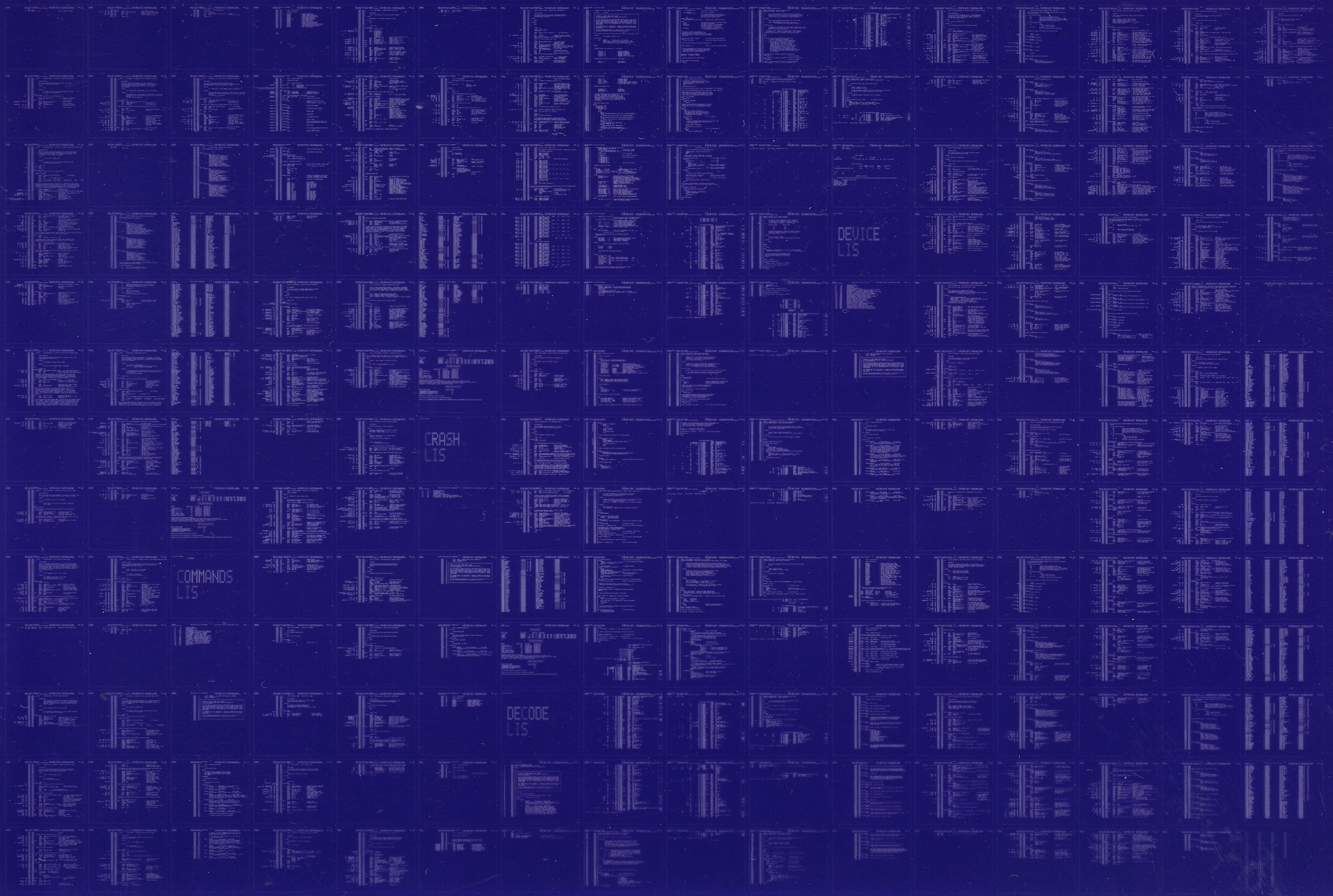
```
ALLOCATE                    *********  X   03     EMB$L_CR_SBIS            = 000000B4
ARGS                      = 00000003              EMB$L_CR_SBISC           = 000000A4
BUG$T_MESSAGES              *********  X   03     EMB$L_CR_SBITA           = 000000B0
BUG$_CHMONIS                *********  X   03     EMB$L_CR_SBR             = 00000078
BUG$_DBLERR                 *********  X   03     EMB$L_CR_SCBB            = 00000084
BUG$_ERRHALT                *********  X   03     EMB$L_CR_SISR            = 0000008C
BUG$_HALT                   *********  X   03     EMB$L_CR_SLR             = 0000007C
BUG$_ILLVEC                 *********  X   03     EMB$L_CR_SP              = 0000005C
BUG$_IVLISTK                *********  X   03     EMB$L_CR_SSP             = 00000018
BUG$_NOUSRWCS               *********  X   03     EMB$L_CR_TBDR            = 000000A0
CURPROC                     *********  X   03     EMB$L_CR_TODR            = 00000098
CURRENT_SYSTEM              *********  X   03     EMB$L_CR_USP             = 0000001C
DISPLAY_CRASH               00000020 RG    03     EMB$Q_CR_TIME            = 00000006
DMP$L_CRASHERL            = 0000006C              EMB$T_CR_LNAME           = 000000FC
DMP$L_FLAGS               = 00000004              ERLPTR                     00000000 RG     02
DMP$V_OLDDUMP             = 00000000              EXE$GB_CPUTYPE             *********  X    03
DMP$W_DUMPVER             = 00000006              EXE$GQ_SYSTIME             *********  X    03
DUMP_HEADER                 *********  X   03     GETMEM                     *********  X    03
EMB$C_CR_LENGTH           = 0000010C              GET_DUMP_INFO              0000053B RG     03
EMB$K_LENGTH              = 00000004              IFD$W_FILNAMOFF          = 00000002
EMB$L_CR_ACCS             = 0000009C              LINE_COUNT                 *********  X    03
EMB$L_CR_AP               = 00000054              MMG$GL_SBR                 *********  X    03
EMB$L_CR_ASTLVL           = 00000088              MMG$GL_SPTLEN              *********  X    03
EMB$L_CR_CADR             = 000000A4              MMG$IMGHDRBUF              *********  X    03
EMB$L_CR_CAER             = 000000AC              MSG$_SUCCESS               *********  X    03
EMB$L_CR_CMIERR           = 000000B0              NEW_PAGE                   *********  X    03
EMB$L_CR_CODE             = 000000F4              PAGE_SIZE                  *********  X    03
EMB$L_CR_ESP              = 00000014              PCB$L_PHD                = 0000006C
EMB$L_CR_FP               = 00000058              PCB$L_PID                = 00000060
EMB$L_CR_ICCS             = 00000090              PCB$T_LNAME              = 00000070
EMB$L_CR_ICR              = 00000094              PHD$L_KSP                = 00000078
EMB$L_CR_ISP              = 00000020              PHD$L_PC                 = 000000C0
EMB$L_CR_KSP              = 00000010              PHD$L_R0                 = 00000088
EMB$L_CR_MCESR            = 000000A8              PHYS_PAGES                 *********  X    03
EMB$L_CR_P0BR             = 00000068              PR$_SID_TYP780             *********  X    03
EMB$L_CR_P0LR             = 0000006C              PRINT                      *********  X    03
EMB$L_CR_P1BR             = 00000070              PRINT_HEADER               000004F6 RG     03
EMB$L_CR_P1LR             = 00000074              PSL$S_CURMOD             = 00000002
EMB$L_CR_PC               = 00000060              PSL$S_IPL                = 00000005
EMB$L_CR_PCBB             = 00000080              PSL$V_CURMOD             = 00000018
EMB$L_CR_PID              = 000000F8              PSL$V_IPL                = 00000010
EMB$L_CR_PSL              = 00000064              PSL$V_IS                 = 0000001A
EMB$L_CR_R0               = 00000024              PUTMEM                     *********  X    03
EMB$L_CR_R1               = 00000028              RESTART_BUGS               00000000 R      03
EMB$L_CR_R10              = 0000004C              SB$S_NODENAME            = 00000010
EMB$L_CR_R11              = 00000050              SB$T_NODENAME            = 00000044
EMB$L_CR_R2               = 0000002C              SCH$GL_CURPCB              *********  X    03
EMB$L_CR_R3               = 00000030              SCS$GA_LOCALSB             *********  X    03
EMB$L_CR_R4               = 00000034              SET_HEADING                *********  X    03
EMB$L_CR_R5               = 00000038              SKIP_LINES                 *********  X    03
EMB$L_CR_R6               = 0000003C              SYS$ASCTIM                 ********* GX    03
EMB$L_CR_R7               = 00000040              SYS$GQ_VERSION             *********  X    03
EMB$L_CR_R8               = 00000044              TRYMEM                     *********  X    03
EMB$L_CR_R9               = 00000048              VERSION_FLAGS              *********  X    03
EMB$L_CR_SBIER            = 000000AC
EMB$L_CR_SBIFS            = 000000A0
EMB$L_CR_SBIMT            = 000000A8
```

```
                                        +----------------+
                                        ! Psect synopsis !
                                        +----------------+

PSECT name                      Allocation           PSECT No.  Attributes
----------                      ----------           ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           00000000  (     0.)  01 (   1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
SDADATA                         00000008  (     8.)  02 (   2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC BYTE
CRASH                           0000069D  (  1693.)  03 (   3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
LITERALS                        000006CC  (  1740.)  04 (   4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

                                    +------------------------+
                                    ! Performance indicators !
                                    +------------------------+

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   35    00:00:00.06    00:00:00.93
Command processing              146    00:00:00.48    00:00:03.66
Pass 1                          297    00:00:05.83    00:00:24.15
Symbol table sort                 0    00:00:00.61    00:00:01.00
Pass 2                          111    00:00:01.50    00:00:05.27
Symbol table output              15    00:00:00.06    00:00:00.08
Psect synopsis output             2    00:00:00.02    00:00:00.02
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            608    00:00:08.56    00:00:35.11

The working set limit was 1650 pages.
52887 bytes (104 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 573 non-local and 70 local symbols.
495 source lines were read in Pass 1, producing 32 object records in Pass 2.
31 pages of virtual memory were used to define 30 macros.

                                   +--------------------------+
                                   ! Macro library statistics !
                                   +--------------------------+

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[SDA.OBJ]SDALIB.MLB;1                      10
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       7
TOTALS (all libraries)                                 27

745 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:CRASH/OBJ=OBJ$:CRASH MSRC$:CRASH/UPDATE=(ENH$:CRASH)+EXECML$/LIB+LIB$:SDALIB/LIB
```

```
LIB
V04
```

0351 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DEVICE
LIS

CRASH
LIS

COMMANDS
LIS

DECODE
LIS