

[illegible][illegible]

```
BBBBBBBBB  UU      UU      GGGGGGGG  CCCCCCCC  HH      HH  KK      KK
BBBBBBBBB  UU      UU      GGGGGGGG  CCCCCCCC  HH      HH  KK      KK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BBBBBBBBB  UU      UU      GG      GG      CC      CC  HHHHHHHHHH  KKKKKK
BBBBBBBBB  UU      UU      GG      GG      CC      CC  HHHHHHHHHH  KKKKKK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BB      BB  UU      UU      GG      GG      CC      CC  HH      HH  KK      KK
BBBBBBBBB  UUUUUUUUU  GGGGGG  CCCCCCCC  HH      HH  KK      KK
BBBBBBBBB  UUUUUUUUU  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
```


C 14
16-Sep-1984 00:18:13
5-Sep-1984 13:49:18

VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]BUGCHK.FOR;1 Page 1

```
0001      SUBROUTINE BUGCHK (LUN)
0002      C
0003      C Version:      'V04-000'
0004      C
0005      C*****
0006      C*
0007      C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008      C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009      C* ALL RIGHTS RESERVED.
0010      C*
0011      C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012      C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013      C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014      C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015      C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016      C* TRANSFERRED.
0017      C*
0018      C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019      C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020      C* CORPORATION.
0021      C*
0022      C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023      C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024      C*
0025      C*
0026      C*****
0027      C
0028      C
0029      C
0030      C      AUTHOR  BRIAN PORTER      CREATION DATE 29-JUL-1979
0031      C
0032      C      Functional description:
0033      C
0034      C      Display for fatal,non-fatal and user bugchecks.
0035      C
0036      C      Modified by:
0037      C
0038      C      V03-002 SAR0066      Sharon A. Reynolds,      20-Jun-1983
0039      C      Changed the carriage control in the 'format' statements
0040      C      for use with ERF.
0041      C
0042      C      V03-001 BP0001      Brian Porter,      17-AUG-1982
0043      C      Removed cpu dependant register for operator requested
0044      C      shutdowns.
0045      C**
0046      C--
0047
0048      INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0107
0108      INCLUDE 'SRC$:SYECOM.FOR /NOLIST'
0236
0237      INCLUDE 'SRC$:MODES.FOR /NOLIST'
0275
0276      BYTE      LUN
0277
0278      INTEGER      I
0279      INTEGER      J
```

0280	INTEGER	K
0281		
0282	INTEGER*2	MTH\$IIAND
0283		
0284	INTEGER*4	FIELD
0285		
0286	INTEGER*4	FIELD1
0287		
0288	INTEGER*4	COMPRESSC
0289		
0290	INTEGER*4	BUGMSG_LENGTH
0291		
0292	LOGICAL*4	SYSSASCTIM
0293	EXTERNAL	SYSSASCTIM
0294		
0295	logical*4	sys\$fao
0296	external	sys\$fao
0297		
0298	REAL*8	TIME2
0299		
0300	CHARACTER*73	BUGMSG_TEXT
0301		
0302	character*16	process_logical_name
0303		
0304	CHARACTER*11	TIME1
0305		
0306	CHARACTER*3	V1STK_PTR(0:4)
0307		
0308	CHARACTER*4	V1VAX_ICCS(0:0)
0309	CHARACTER*9	V2VAX_ICCS(4:4)
0310	CHARACTER*17	V3VAX_ICCS(6:7)
0311	CHARACTER*6	V4VAX_ICCS(31:31)
0312		
0313	CHARACTER*3	GPR(0:14)
0314		
0315	CHARACTER*5	V1SYSREG_LABEL(0:7)
0316		
0317	CHARACTER*31	V1SYSREG_COMENT(0:7)
0318		
0319	CHARACTER*11	AST_MODES(0:3)
0320	CHARACTER*11	V1ASTLVL(0:3)
0321	CHARACTER*11	EXCEPT_ENABLE(6:9)
0322		
0323	EQUIVALENCE	(MODES,AST_MODES)
0324	EQUIVALENCE	(MODES,V1ASTLVL)
0325	EQUIVALENCE	(MODES,EXCEPT_ENABLE)
0326		
0327	PARAMETER	FATAL_BUG = 37
0328		
0329	PARAMETER	KSP = 16
0330	PARAMETER	ESP = 20
0331	PARAMETER	SSP = 24
0332	PARAMETER	USP = 28
0333	PARAMETER	ISP = 32
0334		
0335	PARAMETER	R0 = 36
0336	PARAMETER	R1 = 40

0337	PARAMETER	R2 = 44
0338	PARAMETER	R3 = 48
0339	PARAMETER	R4 = 52
0340	PARAMETER	R5 = 56
0341	PARAMETER	R6 = 60
0342	PARAMETER	R7 = 64
0343	PARAMETER	R8 = 68
0344	PARAMETER	R9 = 72
0345	PARAMETER	R10 = 76
0346	PARAMETER	R11 = 80
0347	PARAMETER	AP = 84
0348	PARAMETER	FP = 88
0349	PARAMETER	SP = 92
0350	PARAMETER	PC = 96
0351		
0352	PARAMETER	PSL = 100
0353		
0354	PARAMETER	POBR = 104
0355	PARAMETER	POLR = 108
0356	PARAMETER	P1BR = 112
0357	PARAMETER	P1LR = 116
0358		
0359	PARAMETER	SBR = 120
0360	PARAMETER	SLR = 124
0361		
0362	PARAMETER	PCBB = 128
0363	PARAMETER	SCBB = 132
0364		
0365	PARAMETER	ASTLVL = 136
0366		
0367	PARAMETER	SISR = 140
0368		
0369	PARAMETER	ICCS = 144
0370	PARAMETER	ICR = 148
0371		
0372	PARAMETER	TODR = 152
0373		
0374	PARAMETER	ACCS = 156
0375		
0376	PARAMETER	SBIFS = 160
0377		
0378	PARAMETER	TBDR = 160
0379		
0380	PARAMETER	SBISC = 164
0381		
0382	PARAMETER	CADR = 164
0383		
0384	PARAMETER	SBIMT = 168
0385		
0386	PARAMETER	MCESR = 168
0387		
0388	PARAMETER	SBIER = 172
0389		
0390	PARAMETER	CAER = 172
0391		
0392	PARAMETER	CMIER = 176
0393		

[illegible]

BUGCHK

F 14
16-Sep-1984 00:18:13
5-Sep-1984 13:49:18VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]BUGCHK.FOR;1
Page 4

```
0394      PARAMETER      SBITA = 176
0395
0396      PARAMETER      SILO = 180
0397
0398      PARAMETER      BUGCHK_CODE = 104
0399
0400
0401      PARAMETER      PID = 108
0402
0403      PARAMETER      LNAME = 112
0404
0405
0406
0407      DATA      V1STK_PTR(0)      /'KSP'/
0408
0409      DATA      V1STK_PTR(1)      /'ESP'/
0410
0411      DATA      V1STK_PTR(2)      /'SSP'/
0412
0413      DATA      V1STK_PTR(3)      /'USP'/
0414
0415      DATA      V1STK_PTR(4)      /'ISP'/
0416
0417
0418
0419
0420
0421
0422      DATA      V1VAX_ICCS(0)      /'RUN*'/
0423
0424
0425
0426      DATA      V2VAX_ICCS(4)      /'TRANSFER*'/
0427
0428
0429
0430
0431      DATA      V3VAX_ICCS(6)      /'INTERRUPT ENABLE*'/
0432
0433      DATA      V3VAX_ICCS(7)      /'INTERRUPT*'/
0434
0435
0436
0437
0438      DATA      V4VAX_ICCS(31)     /'ERROR*'/
0439
0440
0441
0442
0443      DATA      GPR(0)              /'R0 '/
0444
0445      DATA      GPR(1)              /'R1 '/
0446
0447      DATA      GPR(2)              /'R2 '/
0448
0449      DATA      GPR(3)              /'R3 '/
0450
```

```

0451      DATA      GPR(4)          /'R4 '/
0452
0453      DATA      GPR(5)          /'R5 '/
0454
0455      DATA      GPR(6)          /'R6 '/
0456
0457      DATA      GPR(7)          /'R7 '/
0458
0459      DATA      GPR(8)          /'R8 '/
0460
0461      DATA      GPR(9)          /'R9 '/
0462
0463      DATA      GPR(10)         /'R10'/
0464
0465      DATA      GPR(11)         /'R11'/
0466
0467      DATA      GPR(12)         /'AP '/
0468
0469      DATA      GPR(13)         /'FP '/
0470
0471      DATA      GPR(14)         /'SP '/
0472
0473
0474      DATA      V1SYSREG_LABEL(0)/'POBR*'/
0475
0476      DATA      V1SYSREG_LABEL(1)/'POLR*'/
0477
0478      DATA      V1SYSREG_LABEL(2)/'P1BR*'/
0479
0480      DATA      V1SYSREG_LABEL(3)/'P1LR*'/
0481
0482      DATA      V1SYSREG_LABEL(4)/'SBR*'/
0483
0484      DATA      V1SYSREG_LABEL(5)/'SLR*'/
0485
0486      DATA      V1SYSREG_LABEL(6)/'PCBB*'/
0487
0488      DATA      V1SYSREG_LABEL(7)/'SCBB*'/
0489
0490
0491
0492
0493      DATA      V1SYSREG_COMENT(0)/'PO PTE BASE (VIRT ADDR)*'/
0494
0495      DATA      V1SYSREG_COMENT(1)/'TOTAL PO PAGES*'/
0496
0497      DATA      V1SYSREG_COMENT(2)/'P1 PTE BASE (VIRT ADDR)*'/
0498
0499      DATA      V1SYSREG_COMENT(3)/'TOTAL NON-EXISTENT P1 PAGES*'/
0500
0501      DATA      V1SYSREG_COMENT(4)/'SYSTEM PTE BASE (PHY ADDR)*'/
0502
0503      DATA      V1SYSREG_COMENT(5)/'TOTAL PAGES ''SYSTEM'' VIRT MEM*'/
0504
0505      DATA      V1SYSREG_COMENT(6)/'PCB BASE (PHY ADDR)*'/
0506
0507      DATA      V1SYSREG_COMENT(7)/'SCB BASE (PHY ADDR)*'/

```



```
0508
0509
0510
0511
0512
0513
0514      CALL FRCTOF (LUN)
0515      call header (lun)
0516
0517      I = 0
0518
0519      IF (EMB$W_HD_ENTRY .EQ. '25'X) THEN
0520
0521      call logger (lun,'FATAL BUGCHECK')
0522
0523      I = 140
0524
0525      ELSE IF (EMB$W_HD_ENTRY .EQ. '28'X) THEN
0526
0527      call logger (lun,'NON-FATAL BUGCHECK')
0528
0529      ELSE IF (EMB$W_HD_ENTRY .EQ. '70'X) THEN
0530
0531      call logger (lun,'USER BUGCHECK')
0532
0533      ENDIF
0534
0535      CALL CSTRING (EMB(I+BUGCHK_CODE),%REF(BUGMSG_TEXT),BUGMSG_LENGTH)
0536
0537      call linchk (lun,2)
0538
0539      if (bugmsg_length .ne. 0) then
0540
0541      10  WRITE(LUN,10) (BUGMSG_TEXT(K:K),K = 1,BUGMSG_LENGTH)
0542      FORMAT(/' ',<BUGMSG_LENGTH>A1)
0543      else
0544
0545      12  WRITE(LUN,12) (EMB(I+BUGCHK_CODE+K),K = 1,0,-1)
0546      FORMAT(/' ',T8,'BUGCHECK CODE',T28.2Z2.2)
0547      endif
0548
0549      if (
0550      1  (emb$w_hd_entry .eq. '25'x
0551      1  .and.
0552      1  emb(244) .ne. '74'x)
0553      1  .or.
0554      1  emb$w_hd_entry .eq. '28'x
0555      1  .or.
0556      1  emb$w_hd_entry .eq. '70'x
0557      1  ) then
0558
0559      if (sys$fao ('!AF',process_logical_name,%val(15),
0560      1 %ref(emb(i+lname)))) then
0561
0562      CALL LINCHK (LUN,2)
0563
0564      WRITE(LUN,5) process_logical_name
```



```

0565 5  FORMAT(/' ',t8,'PROCESS NAME',t24,a)
0566      endif
0567
0568      CALL LINCHK (LUN,2)
0569
0570      WRITE(LUN,20) (EMB(I+PID+K),K = 3,0,-1)
0571 20  FORMAT(/' ',t8,'PROCESS ID',t24,4z2.2)
0572
0573      I = 0
0574
0575      CALL LINCHK (LUN,2)
0576
0577      WRITE(LUN,25) (EMB(I+PC+K),K = 3,0,-1)
0578 25  FORMAT(/' ',t8,'ERROR PC',t24,4z2.2)
0579
0580      CALL VAXPSL (LUN,EMB(I+PSL))
0581
0582      CALL LINCHK (LUN,4)
0583
0584      write(lun,27) 'STACK POINTERS'
0585 27  format(/' ',a,/)
0586
0587      write(lun,30) (v1stk_ptr(j/4),(emb(ksp+i+j+k),k = 3,0,-1),j = 0,19,4)
0588 30  format(' ',5(a3,' ',4z2.2,' '),/)
0589
0590      CALL LINCHK (LUN,6)
0591
0592      write(lun,27) 'GENERAL REGISTERS'
0593
0594      write(lun,40) ((gpr((j/4)+l*5),(emb(r0+i+j+k+(l*20)),k = 3,0,-1),
0595 40  1 j = 0,19,4),l = 0,2)
0596      format(3(' ',5(a,' ',4z2.2,' '),/))
0597
0598      IF (EMBSW_HD_ENTRY .EQ. FATAL_BUG) I = 140
0599
0600      CALL LINCHK (LUN,2)
0601
0602      IF (EMBSW_HD_ENTRY .EQ. FATAL_BUG) THEN
0603
0604      I = 0
0605
0606      CALL LINCHK (LUN,3)
0607
0608      WRITE(LUN,290)
0609 290  FORMAT(/' ', 'SYSTEM REGISTERS',/)
0610
0611      DO 300,J = POBR,SCBB,4
0612
0613      CALL LINCHK (LUN,2)
0614
0615      FIELD = (J-POBR)/4
0616
0617      WRITE(LUN,295) V1SYSREG_LABEL(FIELD),(EMB(I+J+K),K = 3,0,-1),
0618 295  1 V1SYSREG_COMENT(FIELD)
0619      FORMAT(' ',t8,A<COMPRESSC (V1SYSREG_LABEL(FIELD))>,t24,4z2.2,/,
0620 1 t40,A<COMPRESSC (V1SYSREG_COMENT(FIELD))>)
0621

```

```

0622 300 CONTINUE
0623
0624 CALL LINCHK (LUN,2)
0625
0626
0627 WRITE(LUN,323) (EMB(I+ASTLVL+K),K = 3,0,-1)
0628 323 FORMAT(' ',T8,'ASTLVL',T24,4Z2.2)
0629
0630 FIELD = LIB$EXTZV(0,3,EMB(I+ASTLVL))
0631
0632 IF (FIELD .LE. 3) THEN
0633
0634 WRITE(LUN,325) V1ASTLVL(FIELD)
0635 325 FORMAT(' ',T40,A<COMPRESSC (V1ASTLVL(FIELD))>,
0636 1 ' MODE AST PENDING')
0637
0638 ELSE IF (FIELD .EQ. 4) THEN
0639
0640 WRITE(LUN,327)
0641 327 FORMAT(' ',T40,'NO AST''S PENDING')
0642
0643 ENDIF
0644
0645 CALL LINCHK (LUN,1)
0646
0647 WRITE(LUN,330) (EMB(I+SISR+K),K = 3,0,-1)
0648 330 FORMAT(' ',T8,'SISR',T24,4Z2.2)
0649
0650 DO 340,J = 1,15
0651
0652 FIELD = LIB$EXTZV(J,1,EMB(I+SISR))
0653
0654 IF (FIELD .NE. 0) THEN
0655
0656 CALL LINCHK (LUN,1)
0657
0658 WRITE(LUN,335) J
0659 335 FORMAT(' ',T40,'PENDING INTERRUPT AT LEVEL ',Z1)
0660
0661 ENDIF
0662
0663 340 CONTINUE
0664
0665 FIELD = LIB$EXTZV(16,5,EMB(I+SISR))
0666
0667 CALL LINCHK (LUN,1)
0668
0669 WRITE(LUN,343) FIELD
0670 343 FORMAT(' ',T40,'INTERRUPT REQUEST ACTIVE = ',I2,'.')
0671
0672 CALL LINCHK (LUN,1)
0673
0674 WRITE(LUN,345) (EMB(I+ICCS+K),K = 3,0,-1)
0675 345 FORMAT(' ',T8,'ICCS',T24,4Z2.2)
0676
0677 CALL OUTPUT (LUN,EMB(I+ICCS),V1VAX_ICCS,0,0,0,'0')
0678 CALL OUTPUT (LUN,EMB(I+ICCS),V2VAX_ICCS,4,4,4,'0')

```



```
0679      CALL OUTPUT (LUN,EMB(I+ICCS),V3VAX_ICCS,6,6,7,'0')
0680
0681      CALL OUTPUT (LUN,EMB(I+ICCS),V4VAX_ICCS,31,31,31,'0')
0682
0683      CALL LINCHK (LUN,2)
0684
0685      350  WRITE(LUN,350) (EMB(I+ICR+K),K = 3,0,-1)
0686          FORMAT(' ',T8,'ICR',T24,422.2,/,T40,'INTERVAL COUNT REGISTER')
0687
0688      CALL LINCHK (LUN,1)
0689
0690      355  WRITE(LUN,355) (EMB(I+TODR+K),K = 3,0,-1)
0691          FORMAT(' ',T8,'TODR',T24,422.2)
0692
0693      IF (CP_11780) THEN
0694
0695      CALL LINCHK (LUN,3)
0696
0697      400  WRITE(LUN,400)
0698          FORMAT(/' ', 'CPU-SPECIFIC REGISTERS',/)
0699
0700      CALL ACCS_780 (LUN,EMB(I+ACCS))
0701
0702      CALL SBI_FAULTREG (LUN,EMB(I+SBIFS))
0703
0704      CALL SBI_COMPARATOR (LUN,EMB(I+SBISC))
0705
0706      CALL SBI_MAINTENANCE (LUN,EMB(I+SBIMT))
0707
0708      CALL SBI_ERROR (LUN,EMB(I+SBIER))
0709
0710      C
0711      C      'JIAN' IS CALLED DIRECTLY TO FOOL THE FORTRAN COMPILER INTO
0712      C      PASSING MORE THAN ONE ELEMENT OF THE BYTE ARRAY 'EMB'.
0713      C
0714
0715      FIELD = MTH$JIAN(EMB(I+SBIER),'1000'X)
0716
0717      IF (FIELD .NE. 0) THEN
0718
0719      CALL SBI_TIMEOUT (LUN,EMB(I+SBITA))
0720      ELSE
0721
0722      CALL LINCHK (LUN,1)
0723
0724      410  WRITE(LUN,410) (EMB(I+SBITA+K),K = 3,0,-1)
0725          FORMAT(' ',T8,'SBITA',T24,422.2)
0726      ENDIF
0727
0728      C
0729      C      'JIAN' IS CALLED DIRECTLY TO FOOL THE FORTRAN COMPILER INTO
0730      C      PASSING MORE THAN ONE ELEMENT OF THE BYTE ARRAY 'EMB'.
0731      C
0732
0733      FIELD = MTH$JIAN(EMB(I+SBISC),'A0000000'X)
0734
0735      FIELD1 = MTH$JIAN(EMB(I+SBIFS),'10000'X)
```

```
0736
0737 IF (FIELD .NE. 0 .OR. FIELD1 .NE. 0) THEN
0738
0739 CALL LINCHK (LUN,3)
0740
0741 WRITE(LUN,420)
0742 420 FORMAT(/' ','SBI SILO LOCKED, DETAILED SUMMARY',/)
0743
0744 DO 430,J = 0,60,4
0745
0746 CALL SBI_SILO (LUN,EMB(I+SILO+J))
0747
0748 430 CONTINUE
0749 ENDF
0750
0751 ELSE IF (CP_11750) THEN
0752
0753 c The cpu registers are not output for the 11/750 because
0754 c they are only valid after a machine check.
0755 c
0756 c
0757 ENDF
0758 ENDF
0759 endif
0760
0761 RETURN
0762
0763
0764
0765
0766 ENTRY B_BUGCHK (LUN)
0767
0768
0769
0770 call header (lun)
0771
0772 I = 0
0773
0774 IF (EMB$W_HD_ENTRY .EQ. '25'X) THEN
0775
0776 call logger (lun,'FATAL BUGCHECK')
0777
0778 I = 140
0779
0780 ELSE IF (EMB$W_HD_ENTRY .EQ. '28'X) THEN
0781
0782 call logger (lun,'NON-FATAL BUGCHECK')
0783
0784 ELSE IF (EMB$W_HD_ENTRY .EQ. '70'X) THEN
0785
0786 call logger (lun,'USER BUGCHECK')
0787 ENDF
0788
0789
0790 CALL CSTRING (EMB(I+BUGCHK_CODE),%REF(BUGMSG_TEXT),BUGMSG_LENGTH)
0791
0792
```



```

0793      IF (BUGMSG_LENGTH .NE. 0) THEN
0794
0795      CALL LINCHK (LUN,2)
0796
0797      WRITE(LUN,10) (BUGMSG_TEXT(K:K),K = 1,BUGMSG_LENGTH)
0798      else
0799
0800      CALL LINCHK (LUN,2)
0801
0802      WRITE(LUN,12) (EMB(I+BUGCHK_CODE+K),K = 1,0,-1)
0803      ENDIF
0804
0805      RETURN
0806
0807      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_11722	4-00000014	L*4	CRYPTK_FLAG
4-0000000D	I*4	DEV_CHAR	3-00000000	I*4	EMBSL_RD_SID
3-00000004	I*2	EMBSW_HD_ENTRY	3-0000000E	I*2	EMBSW_HD_ERRSEQ
4-0000001E	L*1	END_VALUE	4-0000001D	L*1	EOF_FLAG
2-0000020C	I*4	FIECD	2-00000210	I*4	FIECD1
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-00000004a	L*1	LUN
4-0000001F	I*4	MAILBOX_CHANNEL	2-000001F6	I*2	MTH\$IIAND
4-0000002B	CHAR	OPTIONS	4-00000008	L*4	PRINTER

BUGCHK

N 14
16-Sep-1984 00:18:13
5-Sep-1984 13:49:18VAX-11 FORTRAN V3.4-56
DISK\$VMSMASTER:[ERF.SRC]BUGCHK.FOR;1 Page 12

2-000001DA CHAR PROCESS_LOGICAL_NAME
4-00000023 I*4 RECORD_SIZE
2-000001F8 R*8 TIME2
4-0000001A L*1 VALID_CPU
4-0000001C L*1 VALID_TYPE

4-00000000 I*4 RECCNT
2-000001EA CHAR TIME1
4-00000019 L*1 VALID_CLASS
4-0000001B L*1 VALID_ENTRY
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
1-000000B2	5'	1-00000089	10'	1-00000095	12'	1-000000CA	20'	1-000000E4	25'
1-00000103	30'	1-0000011A	40'	1-00000135	290'	1-0000014D	295'	**	300
1-0000017E	325'	1-0000019D	327'	1-000001B5	330'	1-000001C8	335'	**	340
1-00000215	345'	1-00000228	350'	1-00000256	355'	1-00000269	400'	1-00000287	410'
**	430							1-0000029B	420'

FUNCTIONS AND SUBROUTINES REFERENCED

Type Name
ACCS_780
FRCTOF
LINCHK
OUTPUT
SBI_FAULTREG
SBI_TIMEOUT

Type Name
I*4 COMPRESSC
HEADER
LOGGER
SBI_COMPARATOR
SBI_MAINTENANCE
L*4 SYS\$FAO

Type Name
CSTRING
I*4 LIB\$EXTZV
I*4 MTH\$JIAND
SBI_ERROR
SBI_SILO
VAXPSL

B 15
16-Sep-1984 00:18:13
5-Sep-1984 13:49:18

BR:

15

15

152

152

152

152

152

152

153

153

153

153

[illegible]

0146

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY