

BBBBBBBBBBBB	00000000	00000000	TTTTTTTTTTTT	SSSSSSSSSS
BBBBBBB9BBB	00000000	00000000	TTTTTTTTTTTT	SSSSSSSSSS
BBBBBBBBBBBB	00000000	00000000	TTTTTTTTTTTT	SSSSSSSSSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBBBBBBBBBBB	000	000	TTT	SSS
BBBBBBBBBBBB	000	000	TTT	SSS
BBBBBBBBBBBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBB	000	000	TTT	SSS
BBBBBBBBBBBB	00000000	00000000	TTT	SSSSSSSSSS
BBBBBBBBBBBB	00000000	00000000	TTT	SSSSSSSSSS
BBBBBBBBBBBB	00000000	00000000	TTT	SSSSSSSSSS

```

LL      000000      AAAAAA      DDDDDDDD      EEEEEEEEEE      RRRRRRRR      SSSSSSSS      UU      UU      BBBB8888
LL      000000      AAAAAA      DDDDDDDD      EEEEEEEEEE      RRRRRRRR      SSSSSSSS      UU      UU      BBBB8888
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AAAAAAAAAA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AAAAAAAAAA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LL      00      00      AA      AA      DD      DD      EE      RR      RR      SS      UU      UU      BB      BB
LLLLLLLLLLLL      000000      AA      AA      DDDDDDDD      EEEEEEEEEE      RR      RR      SSSSSSSS      UUUUUUUUUU      BBBB8888
LLLLLLLLLLLL      000000      AA      AA      DDDDDDDD      EEEEEEEEEE      RR      RR      SSSSSSSS      UUUUUUUUUU      BBBB8888

```

```

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```

LOADERSUB
Table of contents

- LOADER SUBROUTINES

I 15

15-SEP-1984 23:54:59 VAX/VMS Macro V04-00

Page 0

(1)	69	DECLARATIONS
(1)	108	IOGEN\$LOCK_IODB/IOGEN\$UNLK_IODB
(1)	143	IOGEN\$ALLOBLOCK
(1)	176	IOGEN\$CNTRL_INI
(1)	209	IOGEN\$CONN_VEC
(1)	263	IOGEN\$TEST_MEM
(1)	364	IOGEN\$CONSOLE

```
0000 1 .TITLE LOADERSUB - LOADER SUBROUTINES
0000 2 .IDENT 'V04-001'
0000 3 :
0000 4 :*****
0000 5 :*
0000 6 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :* ALL RIGHTS RESERVED.
0000 9 :*
0000 10 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :* TRANSFERRED.
0000 16 :*
0000 17 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :* CORPORATION.
0000 20 :*
0000 21 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :*
0000 24 :*
0000 25 :*****
0000 26 :
0000 27 :
0000 28 :++
0000 29 : FACILITY: I/O DATABASE GENERATOR
0000 30 :
0000 31 : ABSTRACT: LOADER SUBROUTINES
0000 32 :
0000 33 :
0000 34 : ENVIRONMENT: USER MODE PRIVILEGED CODE
0000 35 :
0000 36 : AUTHOR: LEN KAWELL, CREATION DATE:16-JUN-716
0000 37 :
0000 38 : MODIFIED BY:
0000 39 :
0000 40 : V04-001 WHM0004 Bill Matthews 04-Sep-1984
0000 41 : Correct branch in CONNECT CONSOLE/LOGICAL.
0000 42 :
0000 43 : V03-006 WHM0003 Bill Matthews 01-Aug-1984
0000 44 : Fix bugs in CONNECT CONSOLE support for VENUS. Add
0000 45 : CONNECT CONSOLE (terminal) support to MicroVAX I.
0000 46 :
0000 47 : V03-005 WHM0002 Bill Matthews 27-Feb-1984
0000 48 : Set maxunits to 4 for all VENUS CONNECT CONSOLE commands.
0000 49 :
0000 50 : V03-004 WHM0001 Bill Matthews 13-Dec-1983
0000 51 : Added CONNECT CONSOLE support for VENUS.
0000 52 : Converted all CPUDISP's in this module to use the new format
0000 53 : of the macro.
0000 54 :
0000 55 : V03-003 KDM0084 Kathleen D. Morse 23-Sep-1983
0000 56 : Add Micro-VAX I to CPUDISP and use new format of the macro.
0000 57 :
```

0000 58 :
0000 59 :
0000 60 :
0000 61 :
0000 62 :
0000 63 :
0000 64 :
0000 65 :
0000 66 :
0000 67 :--

V03-002 ROW0203 Ralph O. Weber 5-AUG-1983
Change IOGEN\$CNTRL_INI to use IOC\$CTRLINIT, the common,
system-wide routine for calling driver's controller
initialization routines.

V03-001 TCM0001 Trudy C. Matthews 27-Jul-1983
Removed local definition of CPUDISP macro. Updated some of
the CPUDISP invocations to include VENUS-specific code paths.

```

0000 69      .SBTTL  DECLARATIONS
0000 70
0000 71
0000 72 :
0000 73 : INCLUDE FILES:
0000 74 :
0000 75 :
0000 76 :
0000 77 : MACROS:
0000 78 :
0000 79 :
0000 80 :
0000 81 : EQUATED SYMBOLS:
0000 82 :
0000 83
0000 84      $DCDEF      ; DEFINE ADAPTER TYPE SYMBOLS
0000 85      $DYNDEF     ; DEFINE DYNAMIC MEMORY BLOCK TYPES
0000 86      $PRDEF      ; DEFINE PROCESSOR REGISTERS
0000 87      $UCBDEF     ; DEFINE UCB OFFSETS
0000 88      $DDBDEF     ; DEFINE DDB OFFSETS
0000 89      $CRBDEF     ; DEFINE CRB OFFSETS
0000 90      $IDBDEF     ; DEFINE IDB OFFSETS
0000 91      $VECDEF     ; DEFINE CRB DISPATCHER OFFSETS
0000 92      $UBADEF     ; DEFINE UBA REGISTERS
0000 93      $ADPDEF     ; DEFINE ADP OFFSETS
0000 94      $DPTDEF     ; DEFINE DPT OFFSETS
0000 95      $ACFDEF     ; DEFINE ACF OFFSETS
0000 96      $BOOCMDDEF  ; DEFINE SYSGEN COMMAND OPTIONS
0000 97
0000 98 :
0000 99 : OWN STORAGE:
0000 100 :
0000 101
00000000 102 .PSECT NONPAGED_DATA  rd,wrt,noexe,quad
0000 103
0000 104 UBINT_DISP: ;1ST INSTRUCTION IN INTERRUPT
3F BB 0000 105 PUSHR  #*M<R0,R1,R2,R3,R4,R5> ; DISPATCH BLOCK FOR UNIBUS DEVICE.
0002 106

```

```

0002 108      .SBTTL IOGEN$LOCK_IODB/IOGEN$UNLK_IODB
0002 109      :++
0002 110      : IOGEN$LOCK_IODB - LOCK THE SYSTEM I/O DATABASE FOR WRITE
0002 111      : IOGEN$UNLK_IODB - UNLOCK THE SYSTEM I/O DATABASE
0002 112      :
0002 113      : FUNCTIONAL DESCRIPTION:
0002 114      :
0002 115      :     LOCK - GETS THE I/O DATABASE MUTEX FOR WRITE
0002 116      :     UNLOCK - RETURNS THE I/O DATABASE MUTEX AND RETURNS IPL TO 1.
0002 117      :
0002 118      :
0002 119      : INPUTS:
0002 120      :
0002 121      :     NONE
0002 122      :
0002 123      : OUTPUTS:
0002 124      :
0002 125      :     I/O DATABASE LOCKED AND IPL RAISED OR UNLOCKED AND IPL LOWERED
0002 126      :--
0002 127      :
00000000 128 .PSECT NONPAGED_CODE   rd,nowrt,exe,long
0000 129
0000 130 IOGEN$LOCK_IODB::
50 00000000'GF DE 0000 131      MOVAL  G^IOC$GL_MUTEX,R0      ; GET I/O DATA BASE MUTEX
54 00000000'GF DO 0007 132      MOVL   G^SCH$GL_CURPCB,R4      ; GET OWN PCB ADDRESS
   00000000'GF 16 000E 133      JSB    G^SCH$LOCKW      ; AND LOCK IT
   05 0014 134      RSB
   0015 135
0000 136 IOGEN$UNLK_IODB::
50 00000000'GF DE 0015 137      MOVAL  G^IOC$GL_MUTEX,R0      ; GET I/O DATA BASE MUTEX
54 00000000'GF DO 001C 138      MOVL   G^SCH$GL_CURPCB,R4      ; AND OWN PCB ADDRESS
   00000000'GF 16 0023 139      JSB    G^SCH$UNLOCK      ; AND UNLOCK IT
   05 0029 140      SETIPL #0      ; ALSO LOWER IPL
   05 002C 141      RSB

```

```

002D 143 .SBTTL IOGEN$ALLOBLOCK
002D 144 :++
002D 145 :
002D 146 : IOGEN$ALLOBLOCK - ALLOCATE CONTROL BLOCKS
002D 147 :
002D 148 : ALLOCATES A BLOCK OF MEMORY FROM THE NON-PAGED POOL AND
002D 149 : ZEROS IT.
002D 150 :
002D 151 : INPUTS:
002D 152 :
002D 153 : R1 = SIZE OF BLOCK DESIRED
002D 154 :
002D 155 : OUTPUTS:
002D 156 :
002D 157 : R0 = STATUS OF ALLOCATION
002D 158 : R1 = SIZE OF ALLOCATED BLOCK
002D 159 : R2 = ADDRESS OF ALLOCATED BLOCK
002D 160 :
002D 161 :--
002D 162 IOGEN$ALLOBLOCK::
        38 BB 002D 163 PUSHR #^M<R3,R4,R5> ;SAVE REGISTERS
00000000'GF 16 002F 164 JSB G^EXE$ALONONPAGED ;ALLOCATE NON-PAGED POOL
        08 50 E8 0035 165 BLBS R0,10$ ;BR IF SUCCESS
50 0000'8F 3C 0038 166 MOVZWL #SS$ INSMEM,R0 ;SET ERROR STATUS
        38 BA 003D 167 POPR #^M<R3,R4,R5> ;RESTORE REGISTERS
        05 003F 168 RSB ;RETURN
0040 169 10$:
        07 BB 0040 170 PUSHR #^M<R0,R1,R2> ;SAVE REGISTERS
62 51 00 FC AF 00 2C 0042 171 20$: MOVCS #0,20$,#0,R1,(R2) ;ZERO THE BUFFER
        3F BA 0049 172 POPR #^M<R0,R1,R2,R3,R4,R5> ;RESTORE REGISTERS
        05 004B 173 RSB ;RETURN
004C 174

```



```

004C 176      .SBTTL IOGEN$CNTRL_INI
004C 177      :++
004C 178      :
004C 179      : IOGEN$CNTRL INI - INITIALIZE THE DEVICE CONTROLLER
004C 180      :
004C 181      : INPUTS - R6 = ADDR OF DDB
004C 182      :          R11 = ADDR OF DRIVER PROLOGUE TABLE
004C 183      :
004C 184      :--
004C 185      IOGEN$CNTRL_INI::
004C 186
004C 187      ; Limit types of adapters which are processed by this routine.
004C 188
004C 189      DISPATCH -
004C 190      DPT$B_ADPTYPE(R11), TYPE=B, PREFIX=AT$, -
004C 191      <-
004C 192      <UBA,50$>,- ; UNIBUS adapters are ok
004C 193      <MBA,50$>,- ; MASSBUS adapters are ok
004C 194      <CI,50$>,- ; CI adapters are ok
004C 195      <NULL,50$>- ; /NOADAPTER is also ok
004C 196      >
50 01 D0 005D 197      MOVL #1, R0 ; For the rest, set success status,
05 05 0060 198      RSB ; but don't really do anything.
0061 199
00FC 8F BB 0061 200 50$: PUSHR #*M<R2,R3,R4,R5,R6,R6,R7,R8,R9,R10,R11> ; Save registers.
50 04 A6 D0 0065 201      MOVL DDB$L_UCB(R6), R0 ; Get a UCB address.
58 24 A0 D0 0069 202      MOVL UCB$L_CRB(R0), R8 ; Use that to get a CRB address.
5B 56 D0 006D 203      MOVL R6, R1 ; Get DDB address in right place.
51 0C AC D0 0070 204      MOVL ACF$L_CONTRLREG(AP), R1 ; Get last resort CSR address value.
00000000 GF 16 0074 205      JSB G^IOC$CTRLINIT ; Call common controller init routine.
00FC 8F BA 007A 206      POPR #*M<R2,R3,R4,R5,R6,R6,R7,R8,R9,R10,R11> ; Restore registers.
05 05 007E 207      RSB ; Return with IOC$CTRLINIT status in R0.

```

```

007F 209      .SBTTL IOGEN$CONN_VEC
007F 210      :++
007F 211      :
007F 212      : IOGEN$CONN_VEC - CONNECT A UNIBUS INTERRUPT DISPATCHER TO A VECTOR
007F 213      :
007F 214      : THIS SUBROUTINE IS CPU-DEPENDENT. THE FOLLOWING CPU'S ARE SUPPORTED:
007F 215      :
007F 216      :     11/780 -CONNECT VEC$Q_DISPATCH+2 (JSB @#) TO VECTOR
007F 217      :     11/750 -CONNECT VEC$Q_DISPATCH (PUSHR) TO VECTOR
007F 218      :     11/730 -CONNECT VEC$Q_DISPATCH (PUSHR) TO VECTOR
007F 219      :     11/790 -CONNECT VEC$Q_DISPATCH+2 (JSB @#) TO VECTOR
007F 220      :
007F 221      : FOR ALL CPU'S, PUSH R #M<R2,R3,R4,R5> IN THE INTERRUPT DISPATCH BLOCK
007F 222      : IS CHANGED TO PUSH R #M<R0,R1,R2,R3,R4,R5>.
007F 223      :
007F 224      : INPUTS:
007F 225      :
007F 226      :     R0 = ADDRESS OF VECTOR TO CONNECT
007F 227      :     R4 = ADDRESS OF INTERRUPT DISPATCH BLOCK IN CRB
007F 228      :
007F 229      : OUTPUTS:
007F 230      :
007F 231      :     ALL REGISTERS PRESERVED
007F 232      :--
007F 233      :
007F 234      IOGEN$CONN_VEC::
06 01 00000000'GF 8F 007F 235      MOVW  UBINT_DISP,VEC$Q_DISPATCH(R4) ;CHANGE PUSH R TO PUSH R0-R5
64 00000000'EF 60 64 9E 0086 236      MOVAB  VEC$Q_DISPATCH(R4),(R0) ;CONNECT PUSH R TO VECTOR
0089 237      :
0089 238      :
0089 239      .list meb
0089 240      CPUDISP <<780,VEC_780>,- ; Dispatch on cpu type
0089 241      <<750,VEC_750>,-
0089 242      <<730,VEC_730>,-
0089 243      <<790,VEC_790>,-
0089 244      <<UV1,VEC_UV1>,-
0089 245      >
0089 246      CASEB  G^EXE$GB_CPUTYPE, $$BASE, $$LIMIT
0091 30012$:
0012' 0091      .SIGNED_WORD  VEC_780-30012$
0017' 0093      .SIGNED_WORD  VEC_750-30012$
0017' 0095      .SIGNED_WORD  VEC_730-30012$
0012' 0097      .SIGNED_WORD  VEC_790-30012$
00CE 0099      .IIF  EQ  $$GENSW, .WORD  2*<$$LIMIT+1>
000E 009B      .IIF  EQ  $$GENSW, .WORD  2*<$$LIMIT+1>
0017' 009D      .SIGNED_WORD  VEC_UV1-30012$
FEFF 009F      .WORD  ^XFEFF
0004' 00A1      .IIF  IDN <FATAL>,<FATAL> , .WORD  BUGS_UNSUPRTCPU!4
00A3 246      :
00A3 247      VEC_790: ;FOR 11/790 AND
00A3 248      VEC_780: ;FOR 11/780:
60 02 C0 00A3 249      ADDL  #2,(R0) ;STEP VECTOR TO VEC$Q_DISPATCH+2
02 02 11 00A6 250      BRB  VEC_END
00A8 251      :
00A8 252      VEC_750: ;11/750:
00A8 253      VEC_730: ;11/730:
00A8 254      VEC_UV1: ;FOR Micro-VAX I

```

LOADERSUB
V04-001

- LOADER SUBROUTINES
IOGEN\$CONN_VEC

D 16

15-SEP-1984 23:54:59 VAX/VMS Macro V04-00
7-SEP-1984 17:13:56 [BOOTS.SRC]LOADERSUB.MAR;2

Page 8
(1)

```
60  D6  00A8  255      INCL  (R0)      ;SET L.O. BIT TO INDICATE THAT
      COAA  256      ; INTERRUPT HANDLED ON INT STACK
      COAA  257      ;
      COAA  258 VEC_END: ; *END OF CPU-DEPENDENT CODE*
      COAA  259      ;
      COAA  260      ;
05  00AA  261      RSB      ;DONE
```

```
00AB 263 .SBTTL IOGEN$TEST_MEM
00AB 264 :++
00AB 265 :
00AB 266 : IOGEN$TEST_MEM - TEST A PAGE OF MEMORY
00AB 267 :
00AB 268 : THIS TEST IS CPU-DEPENDENT. THE FOLLOWING CPU'S ARE SUPPORTED:
00AB 269 :
00AB 270 : 11/780 -TEST A PAGE OF MEMORY A QUADWORD AT A TIME USING THE
00AB 271 : PR$ SBIQC PROCESSOR REGISTER. THIS CAN ALSO "FIX"
00AB 272 : SOME RDS ERRORS TO BE CRD ERRORS. IF A HARD RDS
00AB 273 : ERROR OCCURS, A MACHINE CHECK WILL RESULT AND THE
00AB 274 : PAGE WILL BE CONSIDERED BAD.
00AB 275 :
00AB 276 : 11/750 -TEST A PAGE OF MEMORY A QUADWORD AT A TIME USING THE
00AB 277 : MOVQ INSTRUCTION. IF A UNCORRECTABLE ECC ERROR OCCURS,
00AB 278 : A MACHINE CHECK WILL RESULT AND THE PAGE WILL BE
00AB 279 : CONSIDERED BAD.
00AB 280 :
00AB 281 : THIS SUBROUTINE SHOULD BE CALLED VIA BRANCH TO SUBROUTINE AT IPL 31.
00AB 282 :
00AB 283 : INPUTS :
00AB 284 :
00AB 285 : R3 = VIRTUAL ADDRESS OF FIRST BYTE OF PAGE.
00AB 286 : R4 = PHYSICAL ADDRESS OF FIRST BYTE OF PAGE.
00AB 287 :
00AB 288 : OUTPUTS:
00AB 289 :
00AB 290 : R0 = LOW BIT SET FOR GOOD PAGE.
00AB 291 : LOW BIT CLEAR FOR BAD PAGE.
00AB 292 : R3 = R3 + 512.
00AB 293 : R4 = R4 + 512.
00AB 294 : OTHER REGISTERS PRESERVED.
00AB 295 : --
00AB 296 IOGEN$TEST MEM: :
00AB 297 PUSHR #*M<R1,R2> : SAVE REGISTERS
51 00000000'EF 06 BB 00AB 298 MOVL MMG$A_SYSPARAM+<EXE$GL_SCB-EXE$A_SYSPARAM>,R1 : GET SYS.EXE COPY OF SCB ADDR
00AB 299 :
00AB 300 PUSHL 4(R1) : SAVE CURRENT MACHINE CHECK HANDLER
00AB 301 MOVL SP,R2 : MARK STACK POSITION
00AB 302 :
00AB 303 :
00AB 304 CPUDISP <<780,SET_MMCHK_780>,- : *DISPATCH ON CPU TYPE*
00AB 305 <750,SET_MMCHK_750>> :
01 01 00000000'GF 8F 00BA 306 CASEB G^EXE$GB_CPUYPE,###BASE,##$LIMIT :
00C2 30018$: :
00C2 .SIGNED_WORD SET_MMCHK_780-30018$ :
00C4 .SIGNED_WORD SET_MMCHK_750-30018$ :
00C6 .WORD ^XFEFF :
00C8 .IIF IDN <FATAL>,<FATAL> , .WORD BUG$_UNSUPRTCPU!4 :
00CA 306 :
00CA 307 SET_MMCHK_750: : FOR 11/750:
00CA 308 MOVAL MMCHK_750,4(R1) : CONNECT 11/750 MCHECK HANDLER
00D2 309 BRB SET_MMCHK_END :
00D4 310 :
00D4 311 SET_MMCHK_780: : FOR 11/780:
00D4 312 MOVAL MMCHK_780,4(R1) : CONNECT 11/780 MCHECK HANDLER
00DC 313 :
```

```

00DC 314 SET_MMCHK_END: ;*END OF CPU-DEPENDENT CODE*
00DC 315
50 40 8F 9A 00DC 316 MOVZBL #<512/8>,R0 ;SET LOOP COUNTER FOR 64 QUADWORDS
00E0 317
00E0 318 TEST_MEM_LOOP:
00E0 319
00E0 320 CPUDISP <<780,CLR_Q_780>,- ;*DISPATCH ON CPU TYPE*
00E0 321 <750,CLR_Q_750>>
01 01 00000000'GF 8F 00E0 CASEB G^EXE$GB_CPUTYPE,$$$BASE,$$$LIMIT
00E8 30024$:
00E8 00E8 .SIGNED_WORD CLR_Q_780-30024$
0010' 00EA .SIGNED_WORD CLR_Q_750-30024$
FEFF 00EC .WORD ^XFEFF
0004' 00EE .IF IDN <FATAL>,<FATAL> , .WORD BUGS_UNSUPRTCPU!4
00F0 322
00F0 323 CLR_Q_780: ;FOR 11/780:
36 54 DA 00F0 324 MTPR R4,#PRS_SBIQC ;CLEAR A QUADWORD
54 08 C0 00F3 325 ADDL #8,R4 ;INCREMENT PHYSICAL ADDRESS
02 11 00F6 326 BRB CLR_Q_END
00F8 327
00F8 328 CLR_Q_750: ;FOR 11/750:
63 7C 00F8 329 CLRQ (R3) ;CLEAR A QUADWORD
00FA 330
00FA 331 CLR_Q_END: ;*END OF CPU-DEPENDENT CODE*
83 83 D1 00FA 332 CMPL (R3)+,(R3)+ ;READ BOTH LONGWORDS, AND ADVANCE TO
00FD 333 ;TO NEXT QUADWORD
E0 50 F5 00FD 334 SOBGR R0,TEST_MEM_LOOP ;DECREMENT QUADWORD COUNTER AND LOOP
0100 335
50 01 9A 0100 336 MOVZBL #1,R0 ;SET STATUS TO SUCCESS
0012 31 0103 337 BRW TEST_MEM_DONE ;JOIN COMMON EXIT
0106 338
0106 339 ;
0106 340 ; LOCAL MACHINE CHECK HANDLER FOR 11/750:
0106 341 ;
0106 342 ;
0106 343 .ALIGN LONG
0108 344 MMCHK_750:
26 0F DA 0108 345 MTPR #^XF,#PRS_MCESR ;CLEAR ANY MEMORY ERROR
06 11 010B 346 BRB BAD_MEM
010D 347
010D 348 ;
010D 349 ; LOCAL MACHINE CHECK HANDLER FOR 11/780:
010D 350 ;
010D 351 ;
010D 352 .ALIGN LONG
0110 353 MMCHK_780:
30 00 DA 0110 354 MTPR #0,#PRS_SBIFS ;CLEAR ANY MEMORY ERROR
0113 355 BAD_MEM: ;WRITING/READING MEMORY CAUSED MCHECK
5E 52 D0 0113 356 MOVL R2,SP ;CLEAR MCHECK LOGOUT STACK
50 D4 0116 357 CLRL R0 ;SET STATUS TO FAILURE
0118 358 TEST_MEM_DONE: ;PREPARE FOR EXIT
04 A1 8ED0 0118 359 POPL 4(R1) ;RESTORE OLD MCHECK HANDLER
06 BA 011C 360 POPR #^M<R1,R2> ;RESTORE REGISTERS
011E 361
011F 362

```

```

011F 364 .SBTTL IOGEN$CONSOLE
011F 365 :++
011F 366 :
011F 367 : IOGEN$CONSOLE - SET UP FOR CONNECTING CONSOLE BLOCK STORAGE DEVICE
011F 368 :
011F 369 : THIS SUBROUTINE IS CPU DEPENDENT. THE FOLLOWING CPU'S ARE SUPPORTED:
011F 370 :
011F 371 : 11/780 -DRIVER NAME IS DXDRIVER. SET IDB BUILT FLAG IN
011F 372 : BOO$GL_CONFLAGS BECAUSE CONSOLE FLOPPY SHARES A
011F 373 : CRB AND IDB WITH THE CONSOLE TERMINAL (OPAO:).
011F 374 :
011F 375 : 11/750 -DRIVER NAME IS DDDRIVER. SET THE SCB VECTOR FLAG IN
011F 376 : BOO$GL_CONFLAGS BECAUSE CONSOLE TU58 INTERRUPTS
011F 377 : THROUGH THE SCB INSTEAD OF THE UBA.
011F 378 :
011F 379 : 11/730 - SAME AS 11/750
011F 380 :
011F 381 : 11/790 - DRIVER NAME IS CVDRIVER. REST IS THE SAME AS 11/750 AND 11/730
011F 382 :
011F 383 : INPUTS:
011F 384 :
011F 385 : NONE
011F 386 :
011F 387 : OUTPUTS:
011F 388 :
011F 389 : THE FOLLOWING GLOBAL LOCATIONS MAY BE MODIFIED:
011F 390 :
011F 391 : BOO$GL_CONDRV
011F 392 : BOO$GL_CONFLAGS
011F 393 : BOO$GL_CONCRB
011F 394 : BOO$GL_CONVECT
011F 395 : BOO$GL_CONUNITS
011F 396 :--
011F 397 :
011F 398 IOGEN$CONSOLE::
011F 399 CPUDISP <<780,CONSOLE_780>,- ;*DISPATCH ON CPU TYPE*
011F 400 <<750,CONSOLE_750>,-
011F 401 <<730,CONSOLE_730>,- ;FOR 11/730:
011F 402 <<UV1,CONSOLE_UV1>,- ;FOR Micro-VAX I
011F 403 <<790,CONSOLE_790>,- ;For 11/790
011F 404 >
06 01 00000000'GF 8F 011F CASEB G^EXE$GB_CPUTYPE, #$$BASE, #$$LIMIT
0127 30036$:
0012' 0127 .SIGNED_WORD CONSOLE_780-30036$
008D' 0129 .SIGNED_WORD CONSOLE_750-30036$
0083' 012B .SIGNED_WORD CONSOLE_730-30036$
0046' 012D .SIGNED_WORD CONSOLE_790-30036$
000E 012F .IIF EQ $$GENSW, .WORD 2*$$LIMIT+1>
000E 0131 .IIF EQ $$GENSW, .WORD 2*$$LIMIT+1>
0029' 0133 .SIGNED_WORD CONSOLE_UV1-30036$
FEFF' 0135 .WORD ^XFEFF
0004' 0137 .IIF IDN <FATAL>, <FATAL> , .WORD BUG$_UNSUPRTCPU!4
0139 405
0139 406 CONSOLE_780: ;FOR 11/780:
0407 MOVAB W^BOO$GT DXNAME, W^BOO$GL_CONDRV ; POINT TO DRIVERNAME
0408 MOVAB #ACFSM_CRBBLT, W^BOO$GL_CONFLAGS ; SET CRB BUILT FLAG
0000'CF 0000'CF 9E 0145 409 MOVAB G^OPASCRB, W^BOO$GL_CONCRB ; STORE CRB ADDRESS
0000'CF 00000000'GF 9E 0145

```

```

76 11 014E 410 BRB CONSOLE_END
      0150 411
      0150 412 CONSOLE_UV1:
0000'CF 01C7'CF 9E 0150 413 -MOVAB W^BOO$GT_CSNAME,W^BOO$GL_CONDRV ; SET DRIVER NAME TO 'CSDRIVER'
      0000'CF 0000'CF D4 0157 414 CLRL W^BOO$GL_CONCUNIT ; Connect unit 0
      0000'CF 01 01 0158 415 MOVL #1,W^BOO$GL_CONUNITS ; One for CSA0 console terminal
      0000'CF 04 90 0160 416 MOVB #ACF$M_SCBVEC,W^BOO$GL_CONFLAGS ; SET SCB VECTOR FLAG
      0000'CF F8 8F 9A 0165 417 MOVZBL #^XF8,W^BOO$GL_CONVECT ; SET SCB VECTOR OFFSET
      59 11 016B 418 BRB CONSOLE_END
      016D 419
      016D 420 CONSOLE_790:
      0000'CF 04 D0 016D 421 -MOVL #4,W^BOO$GL_CONUNITS ; SET MAXUNITS TO 4(OPA0-OPA3)
OF 0000'CF 13 E0 0172 422 BBS #BOOCMD$V_REMOTE,W^BOO$GL_CMDOPT,10$ ; BRANCH IF REMOTE CONSOLE
10 0000'CF 14 E0 0178 423 BBS #BOOCMD$V_LOGICAL,W^BOO$GL_CMDOPT,15$ ; BRANCH IF LOGICAL CONSOLE
0000'CF 0000'CF 9E 017E 424 -MOVAB W^BOO$GT_CSNAME,W^BOO$GL_CONDRV ; POINT TO DRIVER NAME FOR RLO2
      34 11 0185 425 BRB CONSOLE_COMMON
      0000'CF 01 D0 0187 426 10$: MOVL #1,W^BOO$GL_CONCUNIT ; SET UNIT TO 1(OPA1)REMOTE CONSOLE
      05 11 018C 427 20$: BRB ; JOIN COMMON CODE
      0000'CF 03 D0 018E 428 15$: MOVL #3,W^BOO$GL_CONCUNIT ; SET UNIT TO 3(OPA3)LOGICAL CONSOLE
0000'CF 0000'CF 9E 0193 429 20$: -MOVAB W^BOO$GT_OPNAME,W^BOO$GL_CONDEV ; SET DEVICE NAME TO 'OPA'
      0000'CF 02 90 019A 430 MOVB #ACF$M_CRBBLT,W^BOO$GL_CONFLAGS ; SET CRB BUILT FLAG
0000'CF 00000000'GF 9F 019F 431 -MOVAB G^OPAS$CRB,W^BOO$GL_CONCRB ; STORE CRB ADDRESS
      1C 11 01A8 432 BRB CONSOLE_END
      01AA 433 CONSOLE_730: ; For 11/730
      0000'CF 02 D0 01AA 434 -MOVL #2,W^BOO$GL_CONNUMU ; Set number of TUS8 units to 2
      0000'CF 03 D0 01AF 435 -MOVL #3,W^BOO$GL_CONUNITS ; Set max units (OPA0 is 1st unit)
      01B4 436 CONSOLE_750: ;FOR 11/750:
0000'CF 0000'CF 9E 01B4 437 -MOVAB W^BOO$GT_DDNAME,W^BOO$GL_CONDRV ; POINT TO DRIVERNAME
      01BB 438 CONSOLE_COMMON:
      0000'CF 04 90 01BB 439 -MOVB #ACF$M_SCBVEC,W^BOO$GL_CONFLAGS ; SET SCB VECTOR FLAG
      0000'CF F0 8F 9A 01C0 440 -MOVZBL #^XF0,W^BOO$GL_CONVECT ; SET SCB VECTOR OFFSET
      01C6 441
      01C6 442 CONSOLE_END: ;*END OF CPU DEPENDENT CODE*
      01C6 443
      05 01C6 444 RSB
      01C7 445
      01C7 446 BOO$GT_CSNAME:
52 45 56 49 52 44 53 43 00' 01C7 447 .ASCIC /CSDRIVER/ ; Console terminal driver
      08 01C7
      01D0 448
      01D0 449 .END

```

LOADERSUB
Symbol table

- LOADER SUBROUTINES

I 16

15-SEP-1984 23:54:59 VAX/VMS Macro V04-00
7-SEP-1984 17:13:56 [BOOTS.SRC]LOADERSUB.MAR;2

\$\$BASE	=	00000001		
\$\$DISPL	=	00000008		
\$\$GENSW	=	00000001		
\$\$HIGH	=	00000007		
\$\$LIMIT	=	00000006		
\$\$LOW	=	00000001		
\$\$MNSW	=	00000001		
\$\$MXSW	=	00000001		
ACF\$\$_CONTRLREG	=	0000000C		
ACF\$\$_CRBBLT	=	00000002		
ACF\$\$_SCBVEC	=	00000004		
ATS\$_CT	=	00000004		
ATS\$_MBA	=	00000000		
ATS\$_NULL	=	00000005		
ATS\$_UBA	=	00000001		
BAD\$_MEM		00000113	R	03
BOO\$\$_GL_CMDOPT	*****		X	03
BOO\$\$_GL_CONCRB	*****		X	03
BOO\$\$_GL_CONCUNIT	*****		X	03
BOO\$\$_GL_CONDEV	*****		X	03
BOO\$\$_GL_CONDRV	*****		X	03
BOO\$\$_GL_CONFLAGS	*****		X	03
BOO\$\$_GL_CONNUMU	*****		X	03
BOO\$\$_GL_CONUNITS	*****		X	03
BOO\$\$_GL_CONVECT	*****		X	03
BOO\$\$_GT_CSNAME	000001C7		R	03
BOO\$\$_GT_CVNAME	*****		X	03
BOO\$\$_GT_DDNAME	*****		X	03
BOO\$\$_GT_DXNAME	*****		X	03
BOO\$\$_GT_OPNAME	*****		X	03
BOOCMD\$\$_V_LOGICAL	=	00000014		
BOOCMD\$\$_V_REMOTE	=	00000013		
BUG\$\$_UN\$\$_OPRTCPU	*****		X	03
CLR_Q_750	000000F8		R	03
CLR_Q_780	000000F0		R	03
CLR_Q_END	000000FA		R	03
CONSOLE_730	000001AA		R	03
CONSOLE_750	000001B4		R	03
CONSOLE_780	00000139		R	03
CONSOLE_790	0000016D		R	03
CONSOLE_COMMON	000001BB		R	03
CONSOLE_END	000001C6		R	03
CONSOLE_UV1	00000150		R	03
DDB\$\$_L_UCB	=	00000004		
DPT\$\$_B_ADPTYPE	=	0000000C		
EXE\$\$_A_CONONPAGED	*****		X	03
EXE\$\$_A_SYSPARAM	*****		X	03
EXE\$\$_B_CPUTYPE	*****		X	03
EXE\$\$_GL_SCB	*****		X	03
IOC\$\$_CTRL_INIT	*****		X	03
IOC\$\$_GL_MUTEX	*****		X	03
IOGEN\$\$_ALLOBLOCK	0000002D		RG	03
IOGEN\$\$_CNTRL_INI	0000004C		RG	03
IOGEN\$\$_CONN_VEC	0000007F		RG	03
IOGEN\$\$_CONSOLE	0000011F		RG	03
IOGEN\$\$_LOCK_IODB	00000000		RG	03
IOGEN\$\$_TEST_MEM	000000AB		RG	03

IOGEN\$\$_UNLK_IODB	00000015	RG	03
MMCHK_750	00000108	R	03
MMCHK_780	00000110	R	03
MMG\$\$_A_SYSPARAM	*****	X	03
OPAS\$\$_CRB	*****	X	03
PR\$\$_IPL	=	00000012	
PR\$\$_MCE\$\$_SR	=	00000026	
PR\$\$_SBIFS	=	00000030	
PR\$\$_SBIQC	=	00000036	
PR\$\$_SID_TYP730	=	00000003	
PR\$\$_SID_TYP750	=	00000002	
PR\$\$_SID_TYP780	=	00000001	
PR\$\$_SID_TYP790	=	00000004	
PR\$\$_SID_TYPUV1	=	00000007	
SCH\$\$_GL_CURPCB	*****	X	03
SCH\$\$_LOCKW	*****	X	03
SCH\$\$_UNLOCK	*****	X	03
SET_MMCHK_750	000000CA	R	03
SET_MMCHK_780	000000D4	R	03
SET_MMCHK_END	000000DC	R	03
SS\$\$_INSFMEM	*****	X	03
TEST_MEM_DONE	00000118	R	03
TEST_MEM_LOOP	000000E0	R	03
UBINT_DISP	00000000	R	02
UCB\$\$_L_CRB	=	00000024	
VECSQ_DISPATCH	=	00000000	
VEC_730	000000A8	R	03
VEC_750	000000A8	R	03
VEC_780	000000A3	R	03
VEC_790	000000A3	R	03
VEC_END	000000AA	R	03
VEC_UV1	000000A8	R	03

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
NONPAGED_DATA	00000002 (2.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
NONPAGED_CODE	000001D0 (464.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.07	00:00:00.48
Command processing	111	00:00:00.63	00:00:02.04
Pass 1	441	00:00:14.22	00:00:26.61
Symbol table sort	6	00:00:01.83	00:00:03.96
Pass 2	100	00:00:02.69	00:00:05.79
Symbol table output	12	00:00:00.12	00:00:00.24
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	705	00:00:19.59	00:00:39.14

The working set limit was 1500 pages.
111471 bytes (218 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1195 non-local and 29 local symbols.
449 source lines were read in Pass 1, producing 18 object records in Pass 2.
31 pages of virtual memory were used to define 28 macros.

! Macro library statistics !

Macro library name	Macros defined
-\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	1
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	14
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	5
TOTALS (all libraries)	20

1334 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:LOADERSUB/OBJ=OBJ\$:LOADERSUB MSRC\$:LOADERSUB/UPDATE=(ENH\$:LOADERSUB)+EXECML\$/LIB+LIB\$:BOCTS.MLB/LIB

0038

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

Grid of 100 (10x10) small panels, each containing technical diagrams and code snippets. Some panels are highlighted with larger text labels:

- Row 3, Column 4: CUBTDRIUR LIS
- Row 4, Column 3: DOBTDRIV LIS
- Row 4, Column 4: DMBTDRIUR LIS
- Row 4, Column 5: DOBTDRIV LIS
- Row 5, Column 2: CONTO LIS
- Row 5, Column 4: DLBTDRIV LIS
- Row 5, Column 5: OXBTDRIUR LIS
- Row 5, Column 6: INITPGF IL LIS
- Row 6, Column 9: LOADERSUB LIS
- Row 7, Column 9: LOCKDATA LIS
- Row 8, Column 6: LOADDRIU LIS
- Row 8, Column 7: LOADER LIS