

[illegible]

OPT
VO4

```

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
LLLLLLLLLLLLLLLL      IIIIII                      SSSSSSSS
LLLLLLLLLLLLLLLL      IIIIII                      SSSSSSSS

```


(4)	119	REGISTER DEFINITIONS
(5)	159	CONSOLE CONTROLLER INITIALIZATION
(6)	217	CONSOLE UNIT INITIALIZATION
(7)	276	CONSOLE RECIEVER INTERRUPT DISPATCHER
(8)	326	START I/O ON CONSOLE INTERFACE
(9)	372	CONSOLE TRANSMITTER INTERRUPT SERVICE
(9)	387	CONSOLE PORT ACTION ROUTINES
(10)	418	SEND COMMAND TO CONSOLE
(11)	458	"ALLOCATE" CONSOLE TERMINAL
(12)	490	RELEASE CONSOLE TERMINAL
(12)	516	- GET A CHARACTER FROM THE CONSOLE TERMINAL
(12)	547	- PUT A CHARACTER OUT ON THE CONSOLE TERMINAL
(12)	584	- INITIALIZE CONSOLE TERMINAL FOR NON-INTERRUPT DRIVEN I/O
(12)	649	REMAP - MAP VIDEO RAM TO THE SCREEN
(12)	689	MAP_PAGES - MAP PHYSICALLY-CONTIGUOUS PAGES


```
0000 1 .TITLE OPDRVWS1 - VAX/VMS QVSS CONSOLE TERMINAL DRIVER
0000 2 .IDENT 'V04-000'
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 FACILITY:
0000 29
0000 30 VAX/VMS I/O SUBSYSTEM
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 AUTHOR: Bill Matthews
0000 35
0000 36 OPDRIVER AUTHOR: Trudy Matthews, Benn Schreiber
0000 37
0000 38 MODIFIED BY:
0000 39 V03-001 WHM0001 Bill Matthews 01-Aug-1984
0000 40 Initialize the saved scan map. Save r0 across call to remap.
0000 41 Initialize the permanent terminal device characteristics.
0000 42 :-
```


0000	44	:			
0000	45	:	SYMBOL DEFINITIONS		
0000	46	:			
0000	47	:			
0000	48		\$ADPDEF	:	DEFINE ADAPTER CONTROL BLOCK
0000	49		\$CRBDEF	:	DEFINE CRB
0000	50		\$CONDEF	:	DEFINE CONSOLE FUNCTION CODES
0000	51		\$DCDEF	:	DEFINE DEVICE CLASSES
0000	52		\$DDBDEF	:	DEFINE DDB
0000	53		\$DEVDEF	:	DEFINE DEVICE CHARACTERISTICS
0000	54		\$DPTDEF	:	DEFINE DPT
0000	55		\$DYNDEF	:	STRUCTURE TYPE CODE DEFINITIONS
0000	56		\$IDBDEF	:	DEFINE IDB
0000	57		\$IOUV1DEF	:	DEFINE MICROVAX I I/O SPACE
0000	58		\$IPLDEF	:	DEFINE IPL LEVELS
0000	59		\$IRPDEF	:	DEFINE IRP OFFSETS
0000	60		\$PRDEF	:	DEFINE PROCESSOR REGISTERS
0000	61		\$TTDEF	:	DEFINE TERMINAL CHARACTERISTICS
0000	62		\$TT2DEF	:	DEFINE MORE TERMINAL CHARACTERISTICS
0000	63		\$UCBDEF	:	DEFINE UCB
0000	64		\$TTYDEFS	:	TTY UCB extension (must FOLLOW \$UCBDEF)
0000	65		\$TTYMACS	:	TTY macro definitions
0000	66		\$VADEF	:	DEFINE VIRTUAL ADDRESS CONSTANTS
0000	67		\$VECDEF	:	DEFINE CRB VECTOR
0000	68		\$WCBDEF	:	Define WCB
0000	69		\$CINDEF	:	Connect to interrupt offsets
0000	70		\$RBMDEF	:	real time bitmap offsets
0000	71		\$PTEDEF	:	DEFINE PTE


```

00000014 0000 73
0000001C 0000 74 CRBSL_SCAN MAP = CRBSL_TIMELINK      ; ADDRESS OF SCAN MAP SAVE AREA
00000010 0000 75 CRBSL_VIDEO BASE = CRBSL_TOUTROUT    ; VIRT ADDR OF BASE OF VIDEO MEMORY
00000010 0000 76 CRBSL_OPFLAGS = CRBSL_AUXSTRUC      ; HANDSHAKE FLAGS BETWEEN OPDRVWS1 AND
00000010 0000 77                                     ; VCDRIVER
00000010 0000 78 ;      CRBSL_OPFLAGS DEFINITIONS
00000010 0000 79
00000010 0000 80 $VIELD OP,0,<-
00000010 0000 81 <REINIT,,M>-      ; First 24 scan lines must be reinitd
00000010 0000 82 <REMAP,,M>-      ; First 24 scan lines not on screen
00000010 0000 83 <OPACTIVE,,M>-    ; OPDRVWS1 is using the first 24 scan lines
00000010 0000 84 <VCACTIVE,,M>-    ; VCDRIVER has been initialized
00000010 0000 85 >
00000010 0000 86

```



```

0000 88
0000 89 : UCBSW_QV_KEYSTATE DEFINITIONS
0000 90
0000 91 $VIELD KEY,0,<-
0000 92 <APPKEYPAD,,M>-
0000 93 <HOLD,,M>-
0000 94 <CURS_FNC,,M>-
0000 95 <SHIFT,,M>-
0000 96 <CTRL,,M>-
0000 97 <BUTTON,,M>-
0000 98 ; MOUSE BUTTON SAMPLE TOGGLE
0000 99
0000 100 : MAIN QVSS CSR BIT DEFINITIONS
0000 101
0000 102 $VIELD QVCSR,0,<-
0000 103 <MODE19,1,M>-
0000 104 <,1>-
0000 105 <ENA_VIDEO,1,M>-
0000 106 <CURS_FNC,1,M>-
0000 107 <,2>-
0000 108 <ENA_INT,1,M>-
0000 109 <,1>-
0000 110 <BUTA,1,M>-
0000 111 <BUTB,1,M>-
0000 112 <BUTC,1,M>-
0000 113 <MEMBANK,4,M>-
0000 114 >
0000 115
0000 116

```

(R) 15 OR 19 INCH MONITOR (1=19)
SPARE
(RW) ENABLE VIDEO
(RW) CURSOR (1=OR, 0=AND)
(RW) DIAG FUNCTIONS
(RW) ENABLE INTERRUPT
(R) DIAG
(R) MOUSE BUTTON A (RIGHT)
(R) MOUSE BUTTON B (MIDDLE)
(R) MOUSE BUTTON C (LEFT)
(R) VIDEO MEMORY BASE SELECT

```
0000 118
0000 119      .SBTTL REGISTER DEFINITIONS
0000 120
0000 121 ;      VIDEO RAM LAYOUT (0-3FFFF)
0000 122
0003F700 0000 123 QVSVIDE0 SIZE == ^X3F700 ; SIZE OF VIDEO RAM AREA
0003F700 0000 124 QVSCTLBLOCK == ^X3F700 ; QVSS SYSTEM CONTROL BLOCK (QVB)
0003F7E0 0000 125 QVSUCODE == ^X3F7E0 ; UVAX I QVSS CONSOLE AREA
0003F800 0000 126 QVSCAN_MAP == ^X3F800 ; BASE OF SCAN MAP
0003FFE0 0000 127 QVSCUR_RAM == ^X3FFE0 ; CURSOR RAM REGION
0003F800 0000 128 SCAN_MAP == QVSCAN_MAP
0000 129
0000 130 ;      QVSS CONTROL CSRS
0000 131
00000000 0000 132 QVCSR_CTL == 0 ; CONTROL CSR
00000002 0000 133 QVCSR_CURPOS == 2 ; CURSOR POSITION (OUTPUT)
00000004 0000 134 QVCSR_MOUSE == 4 ; MOUSE INPUT (INPUT)
00000006 0000 135 QVCSR_SPARE == 6
0000 136
00000008 0000 137 QVCSR_CRTADDR == 8 ; CRT CONTROLLER (ADDRESS SELECT)
0000000A 0000 138 QVCSR_CRTDATA == 10 ; CRT CONTROLLER (DATA PORT)
0000000C 0000 139 QVCSR_INTDATA == 12 ; INTERRUPT CONTROLLER (DATA VALUES)
0000000E 0000 140 QVCSR_INTCTL == 14 ; INTERRUPT CONTROLLER (CONTROL FIELD)
00000026 0000 141 QVCSR_URTBUFFA == 38 ; UART DATA BUFFER
00000022 0000 142 QVCSR_URTSTATA == 34 ; UART STATUS
00000020 0000 143 QVCSR_URTMODEA == 32
00000024 0000 144 QVCSR_URTCMDA == 36
0000002A 0000 145 QVCSR_URTINT == 42
0000 146
00001E80 0000 147 QVCSR_OFFSET == ^017200 ; OFFSET OF QVSS CSR IN I/O SPACE
20001E80 0000 148 QVCSR_PA == IOUV1$AL_QBOSP+QVCSR_OFFSET; QVSS CSR PHY ADDR
0010000F 0000 149 QVCSR_PFN == QVCSR_PA7512 ; QVSS CSR PFN
00000080 0000 150 QVCSR_BOFF == QVCSR_PA - <QVCSR_PFN*512>; QVSS CSR BYTE OFFSET IN PAGE
0000 151
0000 152 ;
0000 153 ; OUTPUT INTERRUPT QUEUE
0000 154 ;
0000 155
00000000 0000 156      .PSECT SYSLOA, LONG
0000 157
```



```
0000 159 .SBTTL CONSOLE CONTROLLER INITIALIZATION
0000 160 :++
0000 161 : CON$INITIAL - INITIALIZE CONSOLE CONTROLLER
0000 162 :
0000 163 : FUNCTIONAL DESCRIPTION:
0000 164 :
0000 165 : THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE CONTROLLER.
0000 166 :
0000 167 : INPUTS:
0000 168 :
0000 169 : R4 = CSR ADDRESS
0000 170 : R5 = UCB ADDRESS
0000 171 : R9 = CRB ADDRESS
0000 172 :
0000 173 : OUTPUTS:
0000 174 :
0000 175 : ALL REGISTERS ARE PRESERVED.
0000 176 :--
0000 177 CON$INITIAL:: ; INITIALIZE CONSOLE INTERFACE
0000 178
0000 179 MOVAL QVSS$KEY-112,QVSS$KEYTABLE ; INITIALIZE THE KEYBOARD TRANSLATIO
000B 180
000B 181 PUSHL R2 ; SAVE R2
52 00000000'GF DD 000B 182 MOVL G^IOC$GL_ADPLIST,R2 ; GET ADP ADDRESS
52 30 10 A2 D0 000D 183 ADL3 ADP$ VECTOR(R2),#^060,R2 ; GET ADDR OF VECTOR TABLE ENTRY
62 00000025'GF 9E 0019 184 MOVAB G^OPA$CRB+CRB$ _INTD+VEC$Q_DISPATCH+1,(R2); CONNECT THE VECTOR
0020 185 :
0020 186 : SET UP INTERRUPTS
0020 187 :
0020 188 MOVB #0,QVCSR_INTCTL(R4) ; RESET INTERRUPT CONTROLLER
0E A4 40 8F 90 0024 189 MOVB #^X40,QVCSR_INTCTL(R4) ; RESET IRR
0E A4 80 8F 90 0029 190 MOVB #^X80,QVCSR_INTCTL(R4) ; SPECIFY INDIVIDUAL VECTORS
0E A4 C0 8F 90 002E 191 MOVB #^XC0,QVCSR_INTCTL(R4) ; PRESET AUTOCLEAR DATA
0C A4 FF 8F 90 0033 192 MOVB #^XFF,QVCSR_INTDATA(R4) ; ALL ARE AUTO CLEAR
0038 193
0038 194 : VECTOR SPECIFIC
0038 195
0038 196 MOVB #^XE0,QVCSR_INTCTL(R4) ; PRESET VECTOR ADDRESS (ONE)
0C A4 E0 8F 90 003D 197 MOVB #^060,QVCSR_INTDATA(R4) ; USE SPECIAL VECTOR
0E A4 30 8F 90 0041 198 MOVB #^X28,QVCSR_INTCTL(R4) ; ENABLE TX/RX INTERRUPT
0E A4 A1 8F 90 0045 199 MOVB #^XA1,QVCSR_INTCTL(R4) ; ARM THE INTERRUPT CONTROLER CHIP
004A 200
004A 201
004A 202 : SET UP UART
004A 203 :
004A 204 :
004A 205
004A 206 MOVW #^X19,QVCSR_URTCMDA(R4) ; RESET MODE POINTER, ENABLE RCV, DI
24 A4 19 B0 004E 207 MOVW #^X17,QVCSR_URTMODEA(R4) ; SET MODE 1 ,NOPARITY, 8 BIT
20 A4 17 B0 0052 208 MOVW #^X07,QVCSR_URTMODEA(R4) ; SET MODE 2 , 1 STOP BIT
20 A4 07 B0 0056 209 MOVW #^X99,QVCSR_URTSTATA(R4) ; 4800 BAUD XMIT, RCV
22 A4 0099 8F B0 005C 210 MOVW #^X02,QVCSR_URTINT(R4) ; ENABLE REC INTERRUPTS
2A A4 02 B0
0044 8F A8 0060 211
64 0064 212 BISW #<QVCSR$M_ENA_VIDEO!QVCSR$M_ENA_INT>,-; ENABLE VIDEO
52 8ED0 0065 213 QVCSR_CTL(R4) ; INTERRUPTS AND CURSOR=AND
05 0068 214 POPL R2 ; RESTORE R2
05 0068 215 RSB
```



```
0069 217 .SBTTL CONSOLE UNIT INITIALIZATION
0069 218 :++
0069 219 :CONSINITIAL - INITIALIZE CONSOLE UNIT
0069 220 :
0069 221 :FUNCTIONAL DESCRIPTION:
0069 222 :
0069 223 :THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.
0069 224 :
0069 225 :INPUTS:
0069 226 :
0069 227 :R5 = UCB ADDRESS
0069 228 :R9 = CRB ADDRESS
0069 229 :
0069 230 :OUTPUTS:
0069 231 :
0069 232 :ALL REGISTERS ARE PRESERVED.
0069 233 :--
0069 234 :CONSINITLINE::
50 00000000'GF DD 0069 235 PUSHL R0 ; SAVE R0
DE 0068 236 MOVAL G^OPAS$VECTOR,R0 ; GET THE VECTOR ADDRESS
0072 237 CLASS_UNIT_INIT ; AND INIT THIS UNIT
50 0114 C5 D0 00BB 238 MOVL UCBSL TT CLASS(R5),R0 ; ADDRESS OF CLASS VECTOR TABLE
08 64 A5 08 B0 16 00C0 239 JSB @CLASS_SETUP_UCB(R0) ; INITIALIZE THE UCB FOR CONSOLE TERMINAL
50 0114 C5 D0 00C3 240 30$: BBC #UCBSV_POWER_UCBSW_STS(R5),40$; DID WE DETECT A POWER FAIL
20 B0 16 00C8 241 MOVL UCBSL TT CLASS(R5),R0 ; GET THE CLASS VECTOR TABLE ADDRESS
00D0 242 JSB @CLASS_POWERFAIL(R0) ; AND GOTO THE POWERFAIL CODE
50 8ED0 00D0 243 40$: POPL R0 ; RESTORE R0
00D3 244
44 A5 41 A5 00 90 00D3 245 MOVB #TT$ UNKNOWN,UCBSB DEVTYPE(R5); SET UNKNOWN TERMINAL TYPE
48 A5 00001000 8F C8 00D7 246 BISL #TT$M SCOPE,UCBSL DEVDEPEND(R5); QVSS IS SCOPE
21000000 8F CA 00DF 247 BISL #TT2$M EDITING,UCBSL DEVDEPN2(R5); ENABLE LINE EDITING
48 A5 48 A5 00ED 248 BICL #<TT2$M ANSICRT!TT2$M_DECCRT>,-; THIS DRIVER DOES NOT
00C4 C5 44 A5 7D 00EF 249 UCBSL_DEVDEPN2(R5) ; EMULATE VT100'S
00F5 250 UCBSL_DEVDEPEND(R5),UCBSL TT_DECHAR(R5); MAKE PERMANENT
00F5 251
00F5 252 :CONSSET LINE::
00F5 253 :CONSDS SET::
00F5 254 :CONSSET MODEM::
00F5 255 :CONSNULL::
05 00F5 256 RSB
00F6 257 :CONSDISCONNECT::
53 00000000'GF 3F BB 00F6 258 :CALLED ON LAST DEASSIGN
3D 10 A3 03 DE 00F8 259 PUSHR #*M<R0,R1,R2,R3,R4,R5> ; SAVE REGISTERS
2D 00000000'GF 00000000'8F E1 00FF 260 MOVAL G^OPAS$CRB,R3 ; GET CRB ADDRESS
14 A3 D5 0110 261 BBC #OPSV_VACTIVE,CRBSL_OPFLAGS(R3),20$;BC IF VCDRIVER NOT INITED
51 01E0 8F 3C 0113 262 BBC #EXESV_OPA0,G^EXESGL_WSFLAGS,10$; ALL DONE WITH OPA0? IF BC YES
00000000'GF 1A 50 E9 0120 263 TSTL CRBSL_SCAN_MAP(R3) ; SCAN MAP SAVE AREA ALREADY ALLOCATED?
14 A3 52 D0 0123 264 BNEQ 10$ ; IF NEQ YES ALL DONE
0003F800 8F C1 0127 265 MOVZWL #24*10*2,R1 ; ALLOCATE SAVE AREA FOR 24 X 10 SCAN LINE M
61 01E0 8F 28 0130 266 JSB G^EXESALONONPAGED ; GET THE MEMORY
53 00000000'GF 04 CA 013D 267 BLBC R0,10$ ; BRANCH IF ERROR
10 A3 3F BA 0141 268 MOVL R2,CRBSL_SCAN_MAP(R3) ; SAVE SCAN MAP SAVE AREA ADDRESS
62 61 01E0 8F C1 0127 269 ADDL3 #QVSCAN_MAP,CRBSL_VIDEO_BASE(R3),R1; COMPUTE ADDRESS OF SCAN MAP
53 00000000'GF DE 0136 270 MOVCL #24*10*2,(R1),(R2) ; INIT SCAN MAP
10 A3 04 CA 013D 271 MOVAL G^OPAS$CRB,R3 ; GET CRB ADDRESS
20$: BICL #OPSM_OPACTIVE,CRBSL_OPFLAGS(R3); CLEAR OPACTIVE FLAG
10$: POPR #*M<R0,R1,R2,R3,R4,R5> ; RESTORE REGISTERS
```


OPDRVWS1
V04-000

- VAX/VMS QVSS CONSOLE TERMINAL DRIVER N 10
CONSOLE UNIT INITIALIZATION

05 0143 274 RSB

16-SEP-1984 01:08:42 VAX/VMS Macro V04-00
5-SEP-1984 04:11:14 [SYSLOA.SRC]OPDRVWS1.MAR;1

Page 8
(6)

; RETURN


```
0144 276 .SBTTL CONSOLE RECIEVER INTERRUPT DISPATCHER
0144 277 :++
0144 278 :CONSINTINP - CONSOLE INTERRUPT ON INPUT READY
0144 279 :
0144 280 :FUNCTIONAL DESCRIPTION:
0144 281 :
0144 282 :THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
0144 283 :QVSS KEBOARD.
0144 284 :
0144 285 :QVSS TERMINAL: ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
0144 286 : UNSOLICITED AND RESULT IN AN ENTRY INTO THE
0144 287 : TERMINAL DRIVER COMMON CHARACTER BUFFERING
0144 288 : ROUTINE '@UCBSL_TT_PUTNXT(R5)'.
0144 289 :
0144 290 :INPUTS:
0144 291 :
0144 292 : R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
0144 293 :
0144 294 : 00(SP) = ADDRESS OF THE IDB
0144 295 :
0144 296 :OUTPUTS:
0144 297 :
0144 298 : THE SAVED REGISTERS ARE RESTORED BEFORE REI.
0144 299 :--
0144 300 CONSINTINP::
0144 301
0144 302      MOVL    @ (SP)+,R4          ; GET IDB ADDRESS
54 9E D0 0144 302      MOVL    IDBSL_CSR(R4),R0      ; GET CSR ADDRESS
50 64 D0 0147 303
014A 304 : GET THE ASSOCIATED UCB
014A 305
014A 306 5$: MOVL    IDBSL_UCBLST(R4),R5      ; GET UCB 0 ADDRESS
64 A5 55 18 A4 D0 014A 307 5$: BISW    #UCBSM_INTTYPE,UCBSW_STS(R5); SET RECEIVER INTERRUPT
0080 8F A8 014E 308
0154 309 :
50 26 A0 9A 0154 310      MOVZBL  QVCSR_URTBUFFA(R0),R0      ; GET INPUT DATA FROM LK201
FEA5 30      BSBW    QVSS$KEYDECODE      ; DECODE THE KEYBOARD CHARACTER
015B 311
015B 312 : CONSOLE TERMINAL INTERRUPT
015B 313
015B 314 10$: MOVZBL  R0,R3          ; ZERO TOP 3 BYTES
53 50 9A 015B 315 10$: BEQL    30$          ; DON'T PASS NULLS THRU
08 13 015E 316      JSB     @UCBSL_TT_PUTNXT(R5)      ; BUFFER THE CHARACTER
0110 D5 16 0160 317      BEQL    30$          ; IF EQL THEN NO CHARACTER TO OUTPUT
02 13 0164 318      BSBB    CONS$STARTIO      ; OUTPUT THE CHARACTER
0A 10 0166 319 20$: MOVQ    (SP)+,R0      ; RESTORE REGISTERS
50 8E 7D 0168 320 30$: MOVQ    (SP)+,R2
52 8E 7D 016B 321      MOVQ    (SP)+,R4
54 8E 7D 016E 322      REI
02 0171 323
0172 324
```



```
0172 326 .SBTTL START I/O ON CONSOLE INTERFACE
0172 327 :++
0172 328 :CONS$STARTIO - START I/O ON CONSOLE INTERFACE
0172 329 :
0172 330 :FUNCTIONAL DESCRIPTION:
0172 331 :
0172 332 :THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
0172 333 :IF THE INTERFACE IS READY THE DATA IS OUTPUT DIRECTLY. IF THE INTERFACE
0172 334 :IS NOT READY THEN THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE
0172 335 :NEXT READY INTERRUPT.
0172 336 :
0172 337 :IN EITHER CASE, A RETURN TO THE CALLER IS DONE TO ENTER A 'WAIT FOR
0172 338 :INTERRUPT' STATE.
0172 339 :
0172 340 :INPUTS:
0172 341 :
0172 342 :R3 = DATA TO OUTPUT
0172 343 :R5 = UCB ADDRESS
0172 344 :
0172 345 :OUTPUTS:
0172 346 :
0172 347 :R3,R4,R5 ARE PRESERVED.
0172 348 :--
0172 349 :
0172 350 CONS$STARTIO::
0172 351 :
0172 352 10$: BLSS 20$ ; BRANCH IF BURST MODE
50 08 19 0174 353 MOVZBL R3,R0 ; GET CHARACTER TO OUTPUT
53 9A 0177 354 BSBW CONS$PUTCHAR ; OUTPUT CHARACTER
0090 30 017A 355 BRB 30$ ; CHECK FOR MORE TO OUTPUT
12 11 017C 356
017C 357 20$:
017C 358 :
017C 359 :TAKE CHARACTER OUT OF BURST BUFFER AND TRY TO OUTPUT IT IMMEDIATELY
017C 360 :
50 011C D5 9A 017C 361 MOVZBL @UCB$L_TT_OUTADR(R5),R0 ; OUTPUT NEXT BYTE
0086 30 0181 362 BSBW CONS$PUTCHAR ; OUTPUT CHARACTER
011C C5 D6 0184 363 INCL UCB$L_TT_OUTADR(R5) ; UPDATE POINTER
0120 C5 B7 0188 364 DECW UCB$W_TT_OUTLEN(R5) ; UPDATE COUNT
EE 12 018C 365 BNEQ 20$ ; NOT LAST CHARACTER
64 A5 03 8A 018E 366 30$: BICB #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
010C D5 16 0192 367 JSB @UCB$L_TT_GETNXT(R5) ; GET THE NEXT CHARACTER
DA 12 0196 368 BNEQ 10$ ; IF NEQ MORE CHARACTERS TO OUTPUT
05 0198 369 RSB ; EXIT
0199 370
```


D 11

- VAX/VMS QVSS CONSOLE TERMINAL DRIVER
CONSOLE TRANSMITTER INTERRUPT SERVICE

VAX/VMS Macro V04-00
[SYSLOA.SRC]OPDRVWS1.MAR;1

Page 11
(9)

```

0199      372      .SBTTL  CONSOLE TRANSMITTER INTERRUPT SERVICE
0199      373      :++
0199      374      : CON$INTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE
0199      375      :
0199      376      : FUNCTIONAL DESCRIPTION:
0199      377      :
0199      378      : THIS ROUTINE IS A NOP FOR QVSS.
0199      379      :--
0199      380      CON$INTOUT::
50      8E      7D      0199      381      MOVQ      (SP)+,R0      : RESTORE REGISTERS
52      8E      7D      019C      382      MOVQ      (SP)+,R2      :
54      8E      7D      019F      383      MOVQ      (SP)+,R4      :
                                02      01A2      384      REI      :
                                01A3      385      :

```



```

01A3 387 .SBTTL CONSOLE PORT ACTION ROUTINES
01A3 388 :++
01A3 389 : CON$XOFF - SEND XOFF
01A3 390 : CON$XON - SEND XON
01A3 391 : CON$STOP - STOP OUTPUT
01A3 392 : CON$STOP2 - ALTERNATE STOP
01A3 393 : CON$ABORT - ABORT CURRENT OUTPUT
01A3 394 : CON$RESUME - RESUME STOPPED OUTPUT
01A3 395 :
01A3 396 : FUNCTIONAL DESCRIPTION:
01A3 397 :
01A3 398 : THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
01A3 399 : CONTROL OUTPUT ON THE PORT
01A3 400 :
01A3 401 : INPUTS:
01A3 402 :
01A3 403 : R5 = UCB ADDRESS
01A3 404 :
01A3 405 : OUTPUTS:
01A3 406 :
01A3 407 : R5 = UCB ADDRESS
01A3 408 : --
01A3 409 :
01A3 410 CON$XOFF::
01A3 411 CON$XON::
01A3 412 CON$STOP::
01A3 413 CON$ABORT::
01A3 414 CON$RESUME::
05 01A3 415 RSB
01A4 416

```



```
01A4 418 .SBTTL SEND COMMAND TO CONSOLE
01A4 419
01A4 420 :++
01A4 421 : CON$SENDCONSCMD - SEND CPU-DEPENDENT COMMAND TO CONSOLE
01A4 422 :
01A4 423 : FUNCTIONAL DESCRIPTION:
01A4 424 :
01A4 425 : INITIATE FUNCTION ON CONSOLE
01A4 426 :
01A4 427 : INPUTS:
01A4 428 :
01A4 429 : R0 = CONSOLE FUNCTION TO PERFORM:
01A4 430 : CON$C_BOOTCPU = SEND REBOOT SIGNAL TO CONSOLE AND THEN HALT
01A4 431 : CON$C_CLRWARM = CLEAR CONSOLE WARMSTART FLAG
01A4 432 : CON$C_CLRCOLD = CLEAR CONSOLE COLDSTART FLAG
01A4 433 : R2 = NUMBER OF BYTES OF DATA TO BE RETURNED (= 0 IF NO DATA EXPECTED)
01A4 434 : (CURRENTLY ONLY IMPLEMENTED IN 11/790 VERSION OF THIS ROUTINE)
01A4 435 : R3 = ADDRESS OF BUFFER TO HOLD RETURNED DATA (ONLY IF R2 IS NON-ZERO)
01A4 436 : (CURRENTLY IMPLEMENTED ONLY IN 11/790 VERSION OF THIS ROUTINE)
01A4 437 :
01A4 438 : OUTPUTS:
01A4 439 :
01A4 440 : CONSOLE STATE MODIFIED
01A4 441 : R1 DESTROYED
01A4 442 :--
01A4 443
01A4 444 CON$SENDCONSCMD::
50 0F00 8F A8 01A4 445 B1SW #^XFO0,R0 ; SELECT MISCELLANEOUS CONSOLE COMM.
F9 51 22 DB 01A9 446 10$: MFPR #PR$ TXCS,R1 ; GET TRANSMITTER STATUS
02 50 91 01AC 447 BBC #7,RT,10$ ; WAIT FOR CONSOLE READY
23 50 13 01B0 448 CMPB R0,#CON$C_BOOTCPU ; REBOOT CPU?
51 22 DA 01B3 449 BEQL 30$ ; IF SO BRANCH TO HALT AFTER COMMAND
F9 51 07 DB 01B5 450 MTPR R0,#PR$ TXDB ; OTHERWISE ASSERT COMMAND
05 01B8 451 20$: MFPR #PR$ TXCS,R1 ; GET TRANSMITTER STATUS
01BF 452 BBC #7,RT,20$ ; WAIT FOR CONSOLE DONE
01C0 453 RSB ; RETURN
23 50 DA 01C0 454 30$: MTPR R0,#PR$ TXDB ; SEND REBOOT COMMAND TO CONSOLE
00 01C3 455 HALT
```



```

01C4 458      .SBTTL "ALLOCATE" CONSOLE TERMINAL
01C4 459
01C4 460      :++
01C4 461      : CON$OWNCTY - "ALLOCATE" CONSOLE TERMINAL
01C4 462      :
01C4 463      : FUNCTIONAL DESCRIPTION:
01C4 464      :
01C4 465      : THIS ROUTINE SHOULD BE CALLED WHEN PERFORMING NON-INTERRUPT DRIVEN
01C4 466      : I/O TO THE CONSOLE TERMINAL. IT DISABLES INTERRUPTS AND DOES ANY
01C4 467      : CPU-SPECIFIC INITIALIZATION OF THE CONSOLE TERMINAL REGISTERS.
01C4 468      : CON$RELEASECTY SHOULD BE CALLED TO RESTORE THE STATE OF THE CONSOLE
01C4 469      : TERMINAL INTERFACE REGISTERS.
01C4 470      :
01C4 471      : INPUTS:
01C4 472      : NONE
01C4 473      :
01C4 474      : OUTPUTS:
01C4 475      :
01C4 476      : R0:  VALUE TO BE RESTORED TO OPACTIVE FLAG WHEN RELEASING CONSOLE TTY
01C4 477      : R1:  VALUE TO BE RESTORED TO INTERRUPT CSR WHEN RELEASING CONSOLE TTY
01C4 478      :
01C4 479      : QVSS IS SET UP SO THAT NON-INTERRUPT I/O CAN BE
01C4 480      : PERFORMED TO THE CONSOLE TERMINAL.
01C4 481      :
01C4 482      :--
01C4 483      : CON$OWNCTY::
01C4 484      : MOVL  G^OPASIDB+IDB$L,CSR,R0      ;GET CSR ADDRESS
01C4 485      : MOVW  QVCSR CTL(R0),R1      ;SAVE INTERRUPT STATE
01C4 486      : BICW  #QVCSR$M ENA INT,QVCSR CTL(R0) ;DISABLE INTERRUPTS
01C4 487      : EXTZV #OP$V_OPACTIVE,#1,G^OPASCRB+CRB$L_OPFLAGS,R0;SAVE OPACTIVE FLAG
01C4 488      : RSB
  
```

```

50 00000000'GF 51 60
60 0040 8F
50 00000010'GF 01 02
  
```

```

D0 01C4 484
B0 01CB 485
AA 01CE 486
EF 01D3 487
05 01DC 488
  
```

```

MOVL  G^OPASIDB+IDB$L,CSR,R0      ;GET CSR ADDRESS
MOVW  QVCSR CTL(R0),R1      ;SAVE INTERRUPT STATE
BICW  #QVCSR$M ENA INT,QVCSR CTL(R0) ;DISABLE INTERRUPTS
EXTZV #OP$V_OPACTIVE,#1,G^OPASCRB+CRB$L_OPFLAGS,R0;SAVE OPACTIVE FLAG
RSB
  
```



```

01DD 490          .SBTTL RELEASE CONSOLE TERMINAL
01DD 491
01DD 492 :++
01DD 493 : CON$RELEASECTY - RELEASE CONSOLE TERMINAL
01DD 494 :
01DD 495 : FUNCTIONAL DESCRIPTION:
01DD 496 :
01DD 497 : THIS ROUTINE SHOULD BE CALLED TO RELINQUISH EXCLUSIVE USE OF THE
01DD 498 : CONSOLE TERMINAL OBTAINED BY CALLING CON$OWNCTY. IT RESTORES THE
01DD 499 : STATE OF THE CONSOLE.
01DD 500 :
01DD 501 : INPUTS:
01DD 502 : R0: VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO OPACTIVE FLAG
01DD 503 : R1: VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO INTERRUPT CSR
01DD 504 :
01DD 505 : OUTPUTS:
01DD 506 : QVSS AND OPACTIVE FLAG ARE RESTORED TO THEIR ORIGINAL STATE.
01DD 507 :
01DD 508 :--
01DD 509 CON$RELEASECTY::
01DD 510      INSV  R0,#OP$V OPACTIVE,#1,G^OPAS$CRB+CRB$$_OPFLAGS; RESTORE OPACTIVE FLAG
01DD 511      MOVL  G^OPAS$IDB+IDB$$_CSR,R0 ;GET_CSR ADDRESS
01DD 512      MOVW  R1,QVCSR_CTL(R0) ;RESTORE INTERRUPT STATE
01DD 513      RSB
01DD 514
00000010'GF 01 02 50 F0 01DD 510      INSV  R0,#OP$V OPACTIVE,#1,G^OPAS$CRB+CRB$$_OPFLAGS; RESTORE OPACTIVE FLAG
50 00000000'GF D0 01E6 511      MOVL  G^OPAS$IDB+IDB$$_CSR,R0 ;GET_CSR ADDRESS
60 51 B0 01ED 512      MOVW  R1,QVCSR_CTL(R0) ;RESTORE INTERRUPT STATE
05 01F0 513      RSB
01F1 514

```



```

01F1 516 .SBTTL - GET A CHARACTER FROM THE CONSOLE TERMINAL
01F1 517 :++
01F1 518 :CONSGETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
01F1 519 :
01F1 520 :FUNCTIONAL DESCRIPTION:
01F1 521 :
01F1 522 :THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
01F1 523 :DIRECTLY TO THE CONSOLE TERMINAL
01F1 524 :
01F1 525 :INPUTS:
01F1 526 :None
01F1 527 :
01F1 528 :OUTPUTS:
01F1 529 :R0 contains the character.
01F1 530 :
01F1 531 :--
00000013 01F1 532 :control_s = 19 ; control s (xoff)
00000011 01F1 533 :control_q = 17 ; control q (xon)
00000001 01F1 534 :qvuart$m_rxdy = 1 ; receiver ready bit
01F1 535 :
01F1 536 CONSGETCHAR::
50 00000000'GF D0 01F1 537 5$: movl q^opa$idb+idb$l_csr,r0 ;get qvss csr address
22 A0 01 B3 01F8 538 10$: bitw #qvuart$m_rxdy,qvcsr_urtstata(r0);receiver ready?
FA 13 01FC 539 beql 10$ ;if eql not ready
50 26 A0 9A 01FE 540 movzbl qvcsr_urtbufa(r0),r0 ;get character scan code
FDFB' 30 0202 541 bsbw qvss$keydecode ;decode the lk201 input data
50 D5 0205 542 tstl r0 ;need more input?
E8 13 0207 543 beql 5$ ;if eql yes
05 0209 544 rsb ;return
020A 545

```



```

020A 547 .SBTTL - PUT A CHARACTER OUT ON THE CONSOLE TERMINAL
020A 548 :++
020A 549 : CON$PUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL
020A 550 :
020A 551 : FUNCTIONAL DESCRIPTION:
020A 552 :
020A 553 : THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
020A 554 : DIRECTLY TO THE CONSOLE TERMINAL
020A 555 :
020A 556 : INPUTS:
020A 557 : R0 - Character to be output
020A 558 :
020A 559 : OUTPUTS:
020A 560 : Character written to the console terminal.
020A 561 :
020A 562 :--
020A 563 .enabl lsb
020A 564 CON$PUTCHAR::
020A 565 pushr #^m<r1,r2,r3,r4,r5> ;save registers
020C 566 movl g^opa$crb,r3 ;get crb address
0213 567 bbs #op$V_opactive,crb$l_opflags(r3),1$;continue if we have control of t
0218 568 bbs #exe$V_opa0,g^exe$gl_wsflags,1$;output to opa0 enabled? bs yes
0224 569 brw 80$ ;return
0227 570 1$: pushl r0 ;save r0
0229 571 bbcc #op$V_reinit,crb$l_opflags(r3),2$;reinit the scan lines?
022E 572 movc5 #0,(sp),#0,#24*128*10,acrb$l_video_base(r3); init memory
0237 573 movl g^opa$crb,r3 ;get crb address
023E 574 2$: bbcc #op$V_remap,crb$l_opflags(r3),3$;remap scan lines to screen?
0243 575 bsbw remap ;remap scan lines to the screen
0246 576 3$: popl r0 ;restore r0
0249 577 movl crb$l_video_base(r3),r3 ;get va of video memory
024D 578 movl g^opa$idb+idb$l_csr,r4 ;get va of qvss csr
0254 579 bsbw qvss$putchar ;output the character
0257 580 80$: popr #^m<r1,r2,r3,r4,r5> ;restore registers
0259 581 rsb ;
025A 582 .dsabl lsb

```

```

53 00000000'GF 3E BB
0F 10 A3 02 DE
03 00000000'GF 8F E0
0030 31
50 DD
10 10 3 00 E5
00 6E 00 2C
53 00000000'GF DE
03 10 A3 01 E5
0030 30
50 8ED0
53 1C A3 DO
54 00000000'GF DO
FDA9' 30
3E VA
5 S

```

1C B3 7800 8F


```
025A 584 .SBTTL - INITIALIZE CONSOLE TERMINAL FOR NON-INTERRUPT DRIVEN I/O
025A 585 :+
025A 586 :CONSINIT_CTY - INITIALIZE QVSS FOR NON-INTERRUPT DRIVEN I/O
025A 587 :
025A 588 :FUNCTIONAL DESCRIPTION:
025A 589 :
025A 590 :THIS ROUTINE MUST BE CALLED FROM INIT BEFORE ANY CONSOLE TERMINAL I/O
025A 591 :CAN OCCUR.
025A 592 :
025A 593 :INPUTS:
025A 594 :
025A 595 :OUTPUTS:
025A 596 :VIDEO MEMORY MAPPED.
025A 597 :I/O SPACE THAT CONTAINS THE CSRS FOR QVSS MAPPED.
025A 598 :
025A 599 :--
025A 600 CONSINIT_CTY:: ; INITIALIZE QVSS CONTROLLER
025A 601 :
025A 602 :
025A 603 :SAVE REGISTERS
025A 604 :
025A 605 PUSHR #^M<R1,R2,R3,R4>
025C 606 :
025C 607 :
025C 608 :INITIALIZE THE KEYBOARD TRANSLATION TABLE
025C 609 :
00000000'EF FFFFFFFF90'EF DE 025C 610 MOVAL QVSS$KEY-112,QVSS$KEYTABLE
0267 611 :
0267 612 :
0267 613 :Initialize OPDRVWS1 state flags
0267 614 :
0267 615 MOVAL G^OPA$CRB,R3 ; GET THE CRB ADDRESS
53 00000000'GF DE 0267 616 MOVL #OP$M_OPA$ACTIVE,CRB$OP$FLAGS(R3); OPA$ACTIVE,NOREINIT.NOREMAP,NOVCACTI
10 A3 04 DO 026E 617 :
0272 618 :
0272 619 :GET VIRTUAL ADDRESS OF CSR IN R4
0272 620 :
0272 621 MOVL G^BOO$GL_SPTFREL,R2 ; GET A FREE SPT
0279 622 INCL G^BOO$GL_SPTFREL ;
027F 623 MOVL G^MMG$GL_SPTBASE,R1 ; GET VA OF SYSTEM SPT BASE
51 00000000'GF DO 0286 624 MOVAL (R1)[R2],R1 ; GET VA OF SPT PTE
51 51 6142 DE 028A 625 ASHL #9,R2,R2 ; MAKE VA
52 52 52 09 78 028E 626 BISL #VASM_SYSTEM,R2 ; SET SYSTEM SPACE BIT
54 52 80000000 8F C8 0295 627 ADDL3 #QVCSR_BOFF,R2,R4 ; CALC CSR VIRTUAL ADDRESS
52 00000080 8F C1 029D 628 MOVL R4,G^OPA$IDB+IDB$CSR ; SAVE IN IDB
00000000'GF 54 DO 02A4 629 MOVL #<PTESM_VALID!PTESC_KW!QVCSR_PFN>,(R1); MAP PA OF CSR TO SYS VA
61 9010000F 8F DO 02AB 630 :
02AB 631 :
02AB 632 :Map Video Memory
02AB 633 :
02AB 634 CLRL R1 ;
51 64 87FF 8F D4 02AB 635 BICW3 #^C<QVCSR$M_MEMBANK>,(R4); R1; GET BASE QVSS MEMORY BANK
51 51 51 F5 8F 78 02B3 636 ASHL #-QVCSR$V_MEMBANK,R1,R1 ; MAKE IT ZERO BASE
51 00040000 8F C4 02B8 637 MULL2 #^X40000,R1 ; COMPUTE 256K BANK
50 51 F7 8F 78 02BF 638 ASHL #-9,R1,R0 ; AND ISOLATE PFN
51 0200 8F 3C 02C4 639 MOVZWL #512,R1 ; # OF PAGES
3C 10 02C9 640 BSBB MAP_PAGES ; MAP VIDEO RAM
```


OPDRVWS1
V04-000

L 11
- VAX/VMS QVSS CONSOLE TERMINAL DRIVER 16-SEP-1984 01:08:42 VAX/VMS Macro V04-00
- INITIALIZE CONSOLE TERMINAL FOR NON-IN 5-SEP-1984 04:11:14 [SYSLOA.SRC]OPDRVWS1.MAR;1 Page 19
(12)

OB	50	E9	02CB	641	BLBC	R0,100\$	:	NO SPTS THEN EXIT	
1C	A3	52	D0	02CE	642	MOVL	R2,CB\$SL_VIDEO_BASE(R3)	:	SAVE STARTING VA OF BITMAP
		08	10	02D2	643	BSBB	REMAP	:	MAP THE SCREEN
50	0000	8F	3C	02D4	644	MOVZWL	#SS\$ NORMAL,R0	:	INDICATE SUCCESS
		1E	BA	02D9	645	POPR	#^M<R1,R2,R3,R4>	:	
			05	02DB	646	RSB		:	
				02DC	647			:	


```

02DC 649 .SBTTL REMAP - MAP VIDEO RAM TO THE SCREEN
02DC 650 :++
02DC 651 : REMAP
02DC 652 :
02DC 653 : Map the first 24 scan lines to the screen.
02DC 654 :
02DC 655 : Inputs:
02DC 656 :
02DC 657 : R3 - CRB Address for OPA0
02DC 658 :
02DC 659 : Outputs:
02DC 660 : R0,R1,R2 destroyed
02DC 661 :
02DC 662 : Implicit Outputs:
02DC 663 : Scan lines map video ram to the screen.
02DC 664 :
02DC 665 : Side Effects:
02DC 666 : None.
02DC 667 :
02DC 668 :--
02DC 669 REMAP:
02DC 670
02DC 671 : INIT THE SCAN MAP AND MAP SCREEN FULL OF LINES
02DC 672 :
52 1C A3 0003F800 8F C1 02DC 673 ADDL3 #QVSCAN_MAP,CRB$L_VIDEO_BASE(R3),R2; COMPUTE ADDRESS OF SCAN MAP
      14 A3 D5 02E5 674
      0B 13 02E5 675 TSTL CRB$L_SCAN_MAP(R3) ; SCAN MAP SAVE AREA?
      3C BB 02E8 676 BEQL 5$ ; IF EQL NO
14 B3 62 01E0 8F 28 02EA 677 PUSHR #^M<R2,R3,R4,R5> ; SAVE REGISTERS
      3C BA 02EC 678 MOVC3 #24*10*2,(R2),@CRB$L_SCAN_MAP(R3); SAVE SCAN MAP
      50 D4 02F3 679 POPR #^M<R2,R3,R4,R5> ; RESTORE REGISTERS
      8F D0 02F5 680
      50 D0 02F5 681 5$: CLRL R0
      82 50 D0 02F7 682 MOVL #24*10,R1 ; NUMBER OF SCAN MAPS
      50 D0 02F7 683 10$: MOVW R0,(R2)+
      F8 51 D6 0301 684 INCL R0
      05 F5 0303 685 SOBGTR R1,10$
      05 05 0306 686 RSB
      0307 687

```



```
0307 689 .SBTTL MAP_PAGES - MAP PHYSICALLY-CONTIGUOUS PAGES
0307 690 :++
0307 691 : MAP_PAGES
0307 692 :
0307 693 : Map to system virtual address space N physically-contiguous pages.
0307 694 :
0307 695 : Inputs:
0307 696 :     R1 = N = number of physically-contiguous pages
0307 697 :     R0 = Starting PFN
0307 698 :
0307 699 : Outputs:
0307 700 :     R0 = status: SUCCESS, INSFMEM, INSFSPTS
0307 701 :     R1 = preserved
0307 702 :     R2 = system virtual address of N pages of memory if success
0307 703 :     all other registers preserved
0307 704 :
0307 705 : Implicit Outputs:
0307 706 :     None.
0307 707 :
0307 708 : Side Effects:
0307 709 :     IOC$ALLOSPT called - so SPTs are allocated
0307 710 :
0307 711 :--
0307 712 : MAP_PAGES:
2A  BB 0307 713 :     PUSHR    #^M<R1,R3,R4,R5>           : save work registers
0309 714 :                                           : r1 = input used as loop counter
0309 715 :                                           : r3 = address of SPT
0309 716 :                                           : r4 = index into PFN database
0309 717 :                                           : r5 = temp storage
55 50 0309 718 :     MOVL     R0,R5                       : Save starting PFN
00000000'GF 16 030C 719 :     JSB      G^IOC$ALLOSPT              : allocate N SPTs to map VAs
31 50  E9 0312 720 :     BLBC     R0,30$                     : if LBC, no system page table slots
0315 721 :
0315 722 : IOC$ALLOSPT returns:
0315 723 :     R1 = preserved, R2 = SVPN (index into SPT), R3 = address of SPT
0315 724 :
0315 725 : The main loop indexes backwards through the system page table entries
0315 726 : and backwards through the PFN database. It goes backwards so that the
0315 727 : last system virtual address calculated can be returned to the caller.
0315 728 :
0315 729 :
50 52 51 0315 730 :     ADDL3    R1,R2,R0                     : r0 = index into SPT
54 51 55 0319 731 :     ADDL3    R5,R1,R4                     : start at last SPT and go backwards
031D 732 :     10$:     DECL    R0                     : start PFNs at end in loop
50  D7 031D 733 :     DECL    R0                     : set up system page-table entry
54  D7 031F 734 :     DECL    R4                     : back up SVPN index
6340 6340 54 0321 735 :     MOVL     R4,(R3)[R0]                : back up PFN index
A0000000 8F C8 0325 736 :     BISL2    #<PTES$C_UW!PTES$M_VALID>,(R3)[R0] : fill PFN in SPT
032D 737 :                                           : user mode access, valid
032D 738 :     Invalidate system virtual address
032D 739 :
52 52 50 09 78 032D 740 :     ASHL     #9,R0,R2                     : turn SVPN into VA
80000000 8F C8 0331 741 :     BISL2    #<1@31>,R2                 : make VA a system VA
DF 51 F5 0338 742 :     INVALID  R2                         : and clear translation buffer
50 0000'8F 3C 0338 743 :     SOBGTR   R1,10$                     : loop N times
033E 744 :
033E 745 :     MOVZWL   #SS$ _NORMAL,R0             : indicates success (R2 has system VA)
```


OPDRVWS1
V04-000

50	3A	BA	0343	746	15\$:	POPR	#^M<R1,R3,R4,R5>	:	restore work registers
		05	0345	747		RSB		:	
	0000'8F	3C	0346	748	30\$:	MOVZWL	#SS\$_INSFSPTS,R0	:	no SPTs left
	F6	11	034B	749		BRB	15\$	:	return
			034D	750					
			034D	751	CON_END:				
			034D	752	.END				

ADPSL_VECTOR = 00000010
BIT... = 0000000F
BOOSGL_SPTFREL ***** X 02
CLASS_DDT = 00000010
CLASS_GETNXT = 00000000
CLASS_POWERFAIL = 00000020
CLASS_PUTNXT = 00000004
CLASS_SETUP_UCB = 00000008
CON\$ABORT = 000001A3 RG 02
CON\$C_BOOTCPU = 00000002
CON\$DISCONNECT = 000000F6 RG 02
CON\$DS_SET = 000000F5 RG 02
CON\$GETCHAR = 000001F1 RG 02
CON\$INITIAL = 00000000 RG 02
CON\$INITLINE = 00000069 RG 02
CON\$INIT CTY = 0000025A RG 02
CON\$INTINP = 00000144 RG 02
CON\$INTOUT = 00000199 RG 02
CON\$NULL = 000000F5 RG 02
CON\$OWNCTY = 000001C4 RG 02
CON\$PUTCHAR = 0000020A RG 02
CON\$RELEASECTY = 000001DD RG 02
CON\$RESUME = 000001A3 RG 02
CON\$SENDCONSCMD = 000001A4 RG 02
CON\$SET_LINE = 000000F5 RG 02
CON\$SET_MODEM = 000000F5 RG 02
CON\$STARTIO = 00000172 RG 02
CON\$STOP = 000001A3 RG 02
CON\$XOFF = 000001A3 RG 02
CON\$XON = 000001A3 RG 02
CONTROL_Q = 00000011
CONTROL_S = 00000013
CON_END = 0000034D R 02
CRBSL_AUXSTRUC = 00000010
CRBSL_INTD = 00000024
CRBSL_OPFLAGS = 00000010
CRBSL_SCAN_MAP = 00000014
CRBSL_TIMECINK = 00000014
CRBSL_TOUTROUT = 0000001C
CRBSL_VIDEO_BASE = 0000001C
DDBSL_DDT = 0000000C
DPT\$W_VECTOR = 0000001E
EXESA\$CONONPAGED ***** X 02
EXESGL_WSFLAGS ***** X 02
EXESV_OPA0 ***** X 02
IDB\$SL_CSR = 00000000
IDB\$SL_UCBLST = 00000018
IOCSA\$LOSPT ***** X 02
IOU\$GL_ADPLIST ***** X 02
IOU\$V\$AL_QBOSP = 20000000
KEY\$M_APPKEYPAD = 00000001
KEY\$M_BUTTONG = 00000020
KEY\$M_CTRL = 00000010
KEY\$M_HOLD = 00000002
KEY\$M_LOCK = 00000004
KEY\$M_SHIFT = 00000008
KEY\$V_APPKEYPAD = 00000000

KEY\$V_BUTTONG = 00000005
KEY\$V_CTRL = 00000004
KEY\$V_HOLD = 00000001
KEY\$V_LOCK = 00000002
KEY\$V_SHIFT = 00000003
MAP_PAGES = 00000307 R 02
MMG\$GL_SPTBASE ***** X 02
OP\$M_OPACTIVE = 00000004
OP\$M_REINIT = 00000001
OP\$M_REMAP = 00000002
OP\$M_VCACTIVE = 00000008
OP\$V_OPACTIVE = 00000002
OP\$V_REINIT = 00000000
OP\$V_REMAP = 00000001
OP\$V_VCACTIVE = 00000003
OPAS\$CRB ***** X 02
OPAS\$IDB ***** X 02
OPAS\$VECTOR ***** X 02
PR\$TBIS = 0000003A
PR\$TXCS = 00000022
PR\$TXDB = 00000023
PTE\$C_KW = 10000000
PTE\$C_UW = 20000000
PTE\$M_VALID = 80000000
QVCSR\$M_BUTA = 00000100
QVCSR\$M_BUTB = 00000200
QVCSR\$M_BUTC = 00000400
QVCSR\$M_CURS_FNC = 00000008
QVCSR\$M_ENA_INT = 00000040
QVCSR\$M_ENA_VIDEO = 00000004
QVCSR\$M_MEMBANK = 00007800
QVCSR\$M_MODE19 = 00000001
QVCSR\$S_BUTA = 00000001
QVCSR\$S_BUTB = 00000001
QVCSR\$S_BUTC = 00000001
QVCSR\$S_CURS_FNC = 00000001
QVCSR\$S_ENA_INT = 00000001
QVCSR\$S_ENA_VIDEO = 00000001
QVCSR\$S_MEMBANK = 00000004
QVCSR\$S_MODE19 = 00000001
QVCSR\$V_BUTA = 00000008
QVCSR\$V_BUTB = 00000009
QVCSR\$V_BUTC = 0000000A
QVCSR\$V_CURS_FNC = 00000003
QVCSR\$V_ENA_INT = 00000006
QVCSR\$V_ENA_VIDEO = 00000002
QVCSR\$V_MEMBANK = 0000000B
QVCSR\$V_MODE19 = 00000000
QVCSR_BUFF = 00000080 G
QVCSR_CRTADDR = 00000008 G
QVCSR_CRTDATA = 0000000A G
QVCSR_CTL = 00000000 G
QVCSR_CURPOS = 00000002 G
QVCSR_INTCTL = 0000000E G
QVCSR_INTDATA = 0000000C G
QVCSR_MOUSE = 00000004 G
QVCSR_OFFSET = 00001E80 G

OPDRVWS1
Symbol table

- VAX/VMS QVSS CONSOLE TERMINAL DRIVER

D 12

16-SEP-1984 01:08:42
5-SEP-1984 04:11:14

VAX/VMS Macro V04-00
[SYSLOA.SRC]OPDRVWS1.MAR;1

Page 24
(12)

QVCSR_PA	= 20001E80	G	
QVCSR_PFN	= 0010000F	G	
QVCSR_SPARE	= 00000006	G	
QVCSR_URTBUFFA	= 00000026	G	
QVCSR_URTCMDA	= 00000024	G	
QVCSR_URTINT	= 0000002A	G	
QVCSR_URTMODEA	= 00000020	G	
QVCSR_URTSTATA	= 00000022	G	
QVSCAN_MAP	= 0003F800	G	
QVSCTLBLOCK	= 0003F700	G	
QVSCUR_RAM	= 0003FFE0	G	
QVSS\$KEY	*****	X	02
QVSS\$KEYDECODE	*****	X	02
QVSS\$KEYTABLE	*****	X	02
QVSS\$PUTCHAR	*****	X	02
QVSUCODE	= 0003F7E0	G	
QVSVIDEO_SIZE	= 0003F700	G	
QVUART\$M_RXRDY	= 00000001		
REMAP	= 000002DC	R	02
SCAN_MAP	= 0003F800	G	
SIZ...	= 00000004		
SS\$INSFSPTS	*****	X	02
SS\$NORMAL	*****	X	02
TT\$M_SCOPE	= 00001000		
TT\$UNKNOWN	= 00000000		
TT2\$M_ANSICRT	= 01000000		
TT2\$M_DECCRT	= 20000000		
TT2\$M_EDITING	= 00001000		
TTY\$GB_PARITY	*****	X	02
TTY\$GL_DPT	*****	X	02
UCB\$B_DEVTYPE	= 00000041		
UCB\$B_TT_DEPAR1	= 000000EC		
UCB\$B_TT_PARITY	= 000000F8		
UCB\$L_DDB	= 00000028		
UCB\$L_DDT	= 00000088		
UCB\$L_DEVDEPEND	= 00000044		
UCB\$L_DEVDEPND2	= 00000048		
UCB\$L_TT_CLASS	= 00000114		
UCB\$L_TT_DECHAR	= 000000C4		
UCB\$L_TT_GETNXT	= 0000010C		
UCB\$L_TT_OUTADR	= 0000011C		
UCB\$L_TT_PORT	= 00000118		
UCB\$L_TT_PUTNXT	= 00000110		
UCB\$M_INT	= 00000002		
UCB\$M_INTTYPE	= 00000080		
UCB\$M_TIM	= 00000001		
UCB\$V_POWER	= 00000005		
UCB\$W_STS	= 00000064		
UCB\$W_TT_OUTLEN	= 00000120		
VASM_SYSTEM	= 80000000		
VECSQ_DISPATCH	= 00000000		

+-----+
! 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
SYSLOA	0000034D (845.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:02.79
Command processing	107	00:00:00.37	00:00:03.43
Pass 1	534	00:00:14.38	00:00:50.53
Symbol table sort	0	00:00:02.31	00:00:10.10
Pass 2	141	00:00:02.74	00:00:13.61
Symbol table output	21	00:00:00.12	00:00:00.38
Psect synopsis output	2	00:00:00.01	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	836	00:00:19.97	00:01:20.87

The working set limit was 1650 pages.

118953 bytes (233 pages) of virtual memory were used to buffer the intermediate code.

There were 120 pages of symbol table space allocated to hold 2210 non-local and 27 local symbols.

752 source lines were read in Pass 1, producing 17 object records in Pass 2.

59 pages of virtual memory were used to define 56 macros.

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

Macro library name	Macros defined
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	27
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	36

2548 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:OPDRVWS1/OBJ=OBJ\$:OPDRVWS1 MSRC\$:OPDRVWS1/UPDATE=(ENH\$:OPDRVWS1)+EXECMLS/LIB

0398 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

