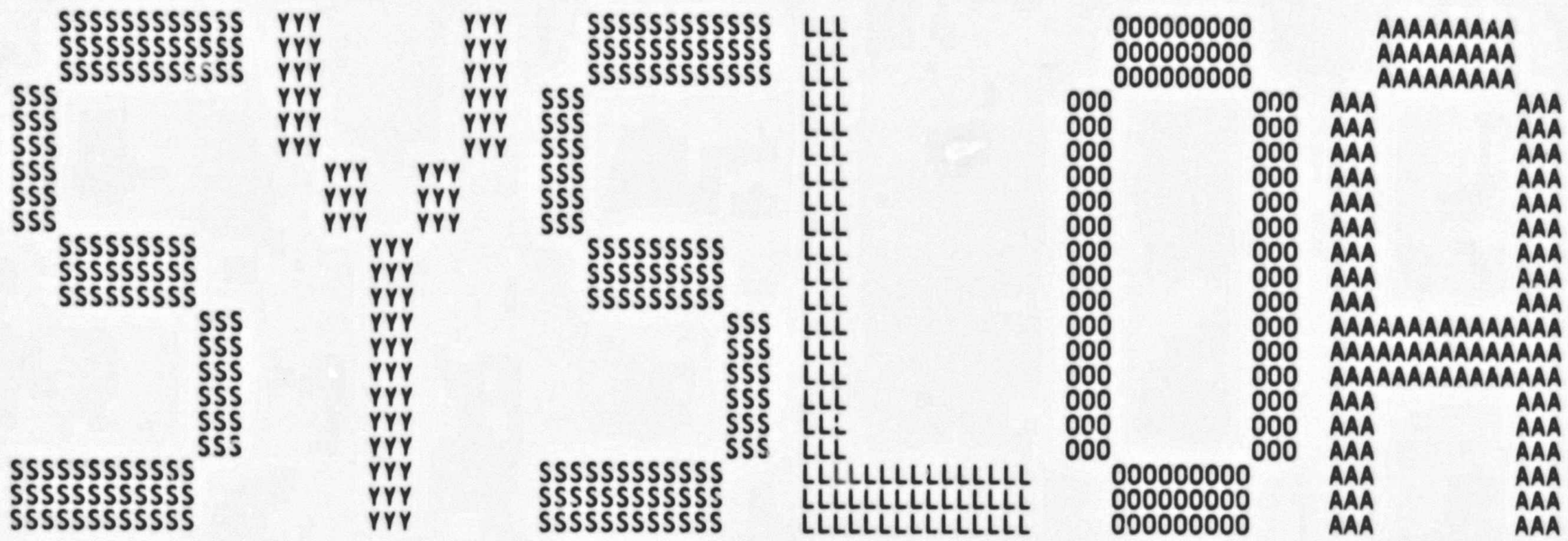
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL OOOOOOOOO AAAAAAAAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSS YYY YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAAAAAAAAAAAAAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSS YYY SSS LLL OOO OOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL OOOOOOOOO AAA AAA

```
  OOOOOO    PPPPPPPP    DDDDDDD    RRRRRRR    VV      VV  WW      WW   SSSSSSS
  OOOOOO    PPPPPPPP    DDDDDDDD   RRRRRRRR   VV      VV  WW      WW   SSSSSSS
OO      OO  PP      PP  DD      DD  RR     RR  VV      VV  WW      WW  SS
OO      OO  PP      PP  DD      DD  RR     RR  VV      VV  WW      WW  SS
OO      OO  PP      PP  DD      DD  RR     RR  VV      VV  WW      WW  SS
OO      OO  PP      PP  DD      DD  RR     RR  VV      VV  WW      WW  SS
OO      OO  PPPPPPPP    DD      DD  RRRRRRRR   VV      VV  WW      WW   SSSSSS
OO      OO  PPPPPPPP    DD      DD  RRRRRRRR   VV      VV  WW      WW   SSSSSS
OO      OO  PP          DD      DD  RR   RR     VV    VV   WW  WW  WW        SS
OO      OO  PP          DD      DD  RR   RR     VV    VV   WW  WW  WW        SS
OO      OO  PP          DD      DD  RR    RR     VV  VV     WWWW  WWWW       SS     ....
OO      OO  PP          DD      DD  RR    RR     VV  VV     WWWW  WWWW       SS     ....
  OOOOOO    PP          DDDDDDDD   RR     RR      VV       WW    WW   SSSSSSS      ....
  OOOOOO    PP          DDDDDDD    RR     RR      VV       WW    WW   SSSSSSS      ....

LL            IIIIII      SSSSSSS
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    SSSSSSS
```

Table of contents

```
0000     1          .TITLE  OPDRVWS - VAX/VMS QVSS CONSOLE TERMINAL DRIVER
0000     2          .IDENT  'V04-002'
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 ;
0000    40 ;       V04-002 KDM0110         Kathleen D. Morse       11-Dec-1984
0000    41 ;               Change name of module to OPDRVWS instead of OPDRVWS1,
0000    42 ;               since module is same for MicroVAX II and I.
0000    43 ;
0000    44 ;       V04-001 WHM0002         Bill Matthews           09-Oct-1984
0000    45 ;               Added clear input FIFO logic to CON$RELEASECTY.
0000    46 ;
0000    47 ;       V03-001 WHM0001         Bill Matthews           01-Aug-1984
0000    48 ;               Initialize the saved scan map. Save r0 across call to remap.
0000    49 ;               Initialize the permanent terminal device characteristics.
0000    50 ;-
```

```
0000     52 ;
0000     53 ; SYMBOL DEFINITIONS
0000     54 ;
0000     55
0000     56          $ADPDEF                         ; DEFINE ADAPTER CONTROL BLOCK
0000     57          $CRBDEF                         ; DFFINE CRB
0000     58          $CONDEF                         ; DEFINE CONSOLE FUNCTION CODES
0000     59          $DCDEF                          ; DEFINE DEVICE CLASSES
0000     60          $DDBDEF                         ; DEFINE DDB
0000     61          $DEVDEF                         ; DEFINE DEVICE CHARACTERISTICS
0000     62          $DPTDEF                         ; DEFINE DPT
0000     63          $DYNDEF                         ; STRUCTURE TYPE CODE DEFINITIONS
0000     64          $IDBDEF                         ; DEFINE IDB
0000     65          $IPLDEF                         ; DEFINE IPL LEVELS
0000     66          $IRPDEF                         ; DEFINE IRP OFFSETS
0000     67          $PRDEF                          ; DEFINE PROCESSOR REGISTERS
0000     68          $TTDEF                          ; DEFINE TERMINAL CHARACTERISTICS
0000     69          $TT2DEF                         ; DEFINE MORE TERMINAL CHARACTERISTICS
0000     70          $UCBDEF                         ; DEFINE UCB
0000     71          $TTYDEFS                        ; TTY UCB extension (must FOLLOW $UCBDEF)
0000     72          $TTYMACS                        ; TTY macro definitions
0000     73          $VADEF                          ; DEFINE VIRTUAL ADDRESS CONSTANTS
0000     74          $VECDEF                         ; DEFINE CRB VECTOR
0000     75          $WCBDEF                         ; Define WCB
0000     76          $CINDEF                         ; Connect to interrupt offsets
0000     77          $RBMDEF                         ; real time bitmap offsets
0000     78          $PTEDEF                         ; DEFINE PTE
```

```
                     0000     80
00000014             0000     81 CRB$L_SCAN_MAP = CRB$L_TIMELINK          ; ADDRESS OF SCAN MAP SAVE AREA
0000001C             0000     82 CRB$L_VIDEO_BASE = CRB$L_TOUTROUT        ; VIRT ADDR OF BASE OF VIDEO MEMORY
00000010             0000     83 CRB$L_OPFLAGS = CRB$L_AUXSTRUC           ; HANDSHAKE FLAGS BETWEEN OPDRVWS1 AND
                     0000     84                                          ; VCDRIVER
                     0000     85 ;       CRB$L_OPFLAGS DEFINITIONS
                     0000     86
                     0000     87         $VIELD  OP,0,<-
                     0000     88                 <REINIT,,M>-            ; First 24 scan lines must be reinited
                     0000     89                 <REMAP,,M>-             ; First 24 scan lines not on screen
                     0000     90                 <OPACTIVE,,M>-          ; OPDRVWS1 is using the first 24 scan lines
                     0000     91                 <VCACTIVE,,M>-          ; VCDRIVER has been initialized
                     0000     92                 >
                     0000     93
```

```
                        95
                        96 ;       UCBSW_QV_KEYSTATE DEFINITIONS
                        97
                        98         $VIELD  KEY,0,<-
                        99                 <APPKEYPAD,,M>-
                       100                 <HOLD,,M>-
                       101                 <LOCK,,M>-
                       102                 <SHIFT,,M>-
                       103                 <CTRL,,M>-
                       104                 <BUTTOG,,M>-            ; MOUSE BUTTON SAMPLE TOGGLE
                       105                 >
                       106
                       107 ;
                       108 ; OUTPUT INTERRUPT QUEUE
                       109 ;
                       110
00000000               111         .PSECT  SYSLOA,LONG
                       112
```

```
                                0000   114          .SBTTL  CONSOLE CONTROLLER INITIALIZATION
                                0000   115 ;++
                                0000   116 ; CON$INITIAL - INITIALIZE CONSOLE CONTROLLER
                                0000   117 ;
                                0000   118 ; FUNCTIONAL DESCRIPTION:
                                0000   119 ;
                                0000   120 ; THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE CONTROLLER.
                                0000   121 ;
                                0000   122 ; INPUTS:
                                0000   123 ;
                                0000   124 ;       R4 = CSR ADDRESS
                                0000   125 ;       R5 = UCB ADDRESS
                                0000   126 ;       R9 = CRB ADDRESS
                                0000   127 ;
                                0000   128 ; OUTPUTS:
                                0000   129 ;
                                0000   130 ;       ALL REGISTERS ARE PRESERVED.
                                0000   131 ;--
                                0000   132 CON$INITIAL::                                 ; INITIALIZE CONSOLE INTERFACE
                                0000   133
00000000'EF   FFFFFF90'EF   DE  0000   134          MOVAL   QVSS$KEY-112,QVSS$KEYTABLE  ; INITIALIZE THE KEYBOARD TRANSLATIO
                                000B   135
                            52  DD  000B   136          PUSHL   R2                      ; SAVE R2
                 52 00000000'GF  D0  000D   137          MOVL    G^IOC$GL_ADPLIST,R2     ; GET ADP ADDRESS
                 52   30   10 A2  C1  0014   138          ADDL3   ADP$L_VECTOR(R2),#^O60,R2 ; GET ADDR OF VECTOR TABLE ENTRY
                 62 00000025'GF  9E  0019   139          MOVAB   G^OPA$CRB+CRB$L_INTD+VEC$Q_DISPATCH+1,(R2); CONNECT THE VECTOR
                                0020   140 ;
                                0020   141 ;       SET UP INTERRUPTS
                                0020   142 ;
                   0E A4    00  90  0020   143          MOVB    #0,QVCSR_INTCTL(R4)     ; RESET INTERRUPT CONTROLLER
                0E A4    40 8F  90  0024   144          MOVB    #^X40,QVCSR_INTCTL(R4)  ; RESET IRR
                0E A4    80 8F  90  0029   145          MOVB    #^X80,QVCSR_INTCTL(R4)  ; SPECIFY INDIVIDUAL VECTORS
                0E A4    C0 8F  90  002E   146          MOVB    #^XC0,QVCSR_INTCTL(R4)  ; PRESET AUTOCLEAR DATA
                0C A4    FF 8F  90  0033   147          MOVB    #^XFF,QVCSR_INTDATA(R4) ; ALL ARE AUTO CLEAR
                                0038   148
                                0038   149 ;       VECTOR SPECIFIC
                                0038   150
                0E A4    E0 8F  90  0038   151          MOVB    #^XE0,QVCSR_INTCTL(R4)  ; PRESET VECTOR ADDRESS (ONE)
                   0C A4    30  90  003D   152          MOVB    #^O60,QVCSR_INTDATA(R4) ; USE SPECIAL VECTOR
                   0E A4    28  90  0041   153          MOVB    #^X28,QVCSR_INTCTL(R4)  ; ENABLE TX/RX INTERRUPT
                0E A4    A1 8F  90  0045   154          MOVB    #^XA1,QVCSR_INTCTL(R4)  ; ARM THE INTERRUPT CONTROLER CHIP
                                004A   155
                                004A   156
                                004A   157 ;
                                004A   158 ; SET UP UART
                                004A   159 ;
                                004A   160
                   24 A4    19  B0  004A   161          MOVW    #^X19,QVCSR_URTCMDA(R4) ; RESET MODE POINTER, ENABLE RCV, DI
                   20 A4    17  B0  004E   162          MOVW    #^X17,QVCSR_URTMODEA(R4) ; SET MODE 1 ,NOPARITY, 8 BIT
                   20 A4    07  B0  0052   163          MOVW    #^X07,QVCSR_URTMODEA(R4) ; SET MODE 2 , 1 STOP BIT
             22 A4    0099 8F  B0  0056   164          MOVW    #^X99,QVCSR_URTSTATA(R4) ; 4800 BAUD XMIT, RCV
                   2A A4    02  B0  005C   165          MOVW    #^X02,QVCSR_URTINT(R4)  ; ENABLE REC INTERRUPTS
                                0060   166
                      0044 8F  A8  0060   167          BISW    #<QVCSR$M_ENA_VIDEO!QVCSR$M_ENA_INT>,-; ENABLE VIDEO
                           64      0064   168                  QVCSR_CTL(R4)           ; INTERRUPTS AND CURSOR=AND
                         52 8ED0   0065   169          POPL    R2                      ; RESTORE R2
                              05   0068   170          RSB
```

```
                                0069   172          .SBTTL  CONSOLE UNIT INITIALIZATION
                                0069   173 ;++
                                0069   174 ; CON$INITIAL - INITIALIZE CONSOLE UNIT
                                0069   175 ;
                                0069   176 ; FUNCTIONAL DESCRIPTION:
                                0069   177 ;
                                0069   178 ; THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.
                                0069   179 ;
                                0069   180 ; INPUTS:
                                0069   181 ;
                                0069   182 ;       R5 = UCB ADDRESS
                                0069   183 ;       R9 = CRB ADDRESS
                                0069   184 ;
                                0069   185 ; OUTPUTS:
                                0069   186 ;
                                0069   187 ;       ALL REGISTERS ARE PRESFRVED.
                                0069   188 ;--
                                0069   189 CON$INITLINE::
                          50    DD    0069   190          PUSHL   R0                      ; SAVE R0
              50  00000000'GF    DE    006B   191          MOVAL   G^OPA$VECTOR,R0         ; GET THE VECTOR ADDRESS
                                0072   192          CLASS_UNIT_INIT                 ; AND INIT THIS UNIT
                   50  0114 C5  D0    00BB   193          MOVL    UCB$L_TT_CLASS(R5),R0   ; ADDRESS OF CLASS VECTOR TABLE
                       08 B0    16    00C0   194          JSB     @CLASS_SETUP_UCB(R0)    ; INITIALIZE THE UCB FOR CONSOLE TERMINAL
                08 64 A5    05    E1    00C3   195 30$:     BBC     #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
                   50  0114 C5  D0    00C8   196          MOVL    UCB$L_TT_CLASS(R5),R0   ; GET THE CLASS VECTOR TABLE ADDRESS
                       20 B0    16    00CD   197          JSB     @CLASS_POWERFAIL(R0)    ; AND GOTO THE POWERFAIL CODE
                                00D0   198
                          50 8ED0    00D0   199 40$:     POPL    R0                      ; RESTORE R0
                                00D3   200
                      41 A5    00    90    00D3   201          MOVB    #TT$_UNKNOWN,UCB$B_DEVTYPE(R5); SET UNKNOWN TERMINAL TYPE
          44 A5   00001000 8F    C8    00D7   202          BISL    #TT$M_SCOPE,UCB$L_DEVDEPEND(R5); QVSS IS SCOPE
          48 A5   00001000 8F    C8    00DF   203          BISL    #TT2$M_EDITING,UCB$L_DEVDEPND2(R5); ENABLE LINE EDITING
                  21000000 8F    CA    00E7   204          BICL    #<TT2$M_ANSICRT!TT2$M_DECCRT>,-; THIS DRIVER DOES NOT
                         48 A5          00ED   205                  UCB$L_DEVDEPND2(R5)     ; EMULATE VT100'S
              00C4 C5    44 A5    7D    00EF   206          MOVQ    UCB$L_DEVDEPEND(R5),UCB$L_TT_DECHAR(R5); MAKE PERMANENT
                                00F5   207
                                00F5   208 CON$SET_LINE::
                                00F5   209 CON$DS_SET::
                                00F5   210 CON$SET_MODEM::
                                00F5   211 CON$NULL::
                                05    00F5   212          RSB                             ;
                                00F6   213 CON$DISCONNECT::                            ; CALLED ON LAST DEASSIGN
                          3F    BB    00F6   214          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; SAVE REGISTERS
              53  00000000'GF    DE    00F8   215          MOVAL   G^OPA$CRB,R3            ; GET CRB ADDRESS
                   04 10 A3    03    E1    00FF   216          BBC     #OP$V_VCACTIVE,CRB$L_OPFLAGS(R3),20$;BC IF VCDRIVER NOT INITED
                      10 A3    04    CA    0104   217 10$:     BICL    #OP$M_OPACTIVE,CRB$L_OPFLAGS(R3); CLEAR OPACTIVE FLAG
                          3F    BA    0108   218 20$:     POPR    #^M<R0,R1,R2,R3,R4,R5>  ; RESTORE REGISTERS
                                05    010A   219          RSB                             ; RETURN
```

```
                     010B   221          .SBTTL CONSOLE RECIEVER INTERRUPT DISPATCHER
                     010B   222 ;++
                     010B   223 ; CON$INTINP - CONSOLE INTERRUPT ON INPUT READY
                     010B   224 ;
                     010B   225 ; FUNCTIONAL DESCRIPTION:
                     010B   226 ;
                     010B   227 ; THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
                     010B   228 ; QVSS KEBOARD.
                     010B   229 ;
                     010B   230 ; QVSS TERMINAL:        ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
                     010B   231 ;                       UNSOLICITED AND RESULT IN AN ENTRY INTO THE
                     010B   232 ;                       TERMINAL DRIVER COMMON CHARACTER BUFFERING
                     010B   233 ;                       ROUTINE "@UCB$L_TT_PUTNXT(R5)".
                     010B   234 ;
                     010B   235 ; INPUTS:
                     010B   236 ;
                     010B   237 ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
                     010B   238 ;
                     010B   239 ;       00(SP) = ADDRESS OF THE IDB
                     010B   240 ;
                     010B   241 ; OUTPUTS:
                     010B   242 ;
                     010B   243 ;       THE SAVED REGISTERS ARE RESTORED BEFORE REI.
                     010B   244 ;--
                     010B   245 CON$INTINP::
                     010B   246
    54    9E   D0    010B   247          MOVL    @(SP)+,R4               ; GET IDB ADDRESS
    50    64   D0    010E   248          MOVL    IDB$L_CSR(R4),R0        ; GET CSR ADDRESS
                     0111   249 ;
                     0111   250 ; GET THE ASSOCIATED UCB
                     0111   251 ;
          50   26 A0   9A   0111   252          MOVZBL  QVCSR_URTBUFA(R0),R0    ; GET INPUT DATA FROM LK201
                     0115   253 ;
                     0115   254 ; CALLED HERE FROM VCDRIVER WITH CHARACTER IN R0 AND IDB ADDRESS IN R4
                     0115   255 ;
                     0115   256 CON$VCINP::
          55   18 A4   D0   0115   257          MOVL    IDB$L_UCBLST(R4),R5     ; GET UCB 0 ADDRESS
 64 A5    0080 8F   A8   0119   258          BISW    #UCB$M_INTTYPE,UCB$W_STS(R5); SET RECEIVER INTERRUPT
             FEDE'  30   011F   259          BSBW    QVSS$KEYDECODE          ; DECODE THE KEYBOARD CHARACTER
                     0122   260 ;
                     0122   261 ; CONSOLE TERMINAL INTERRUPT
                     0122   262 ;
          53   50   9A   0122   263 10$:     MOVZBL  R0,R3                   ; ZERO TOP 3 BYTES
                    08   13   0125   264          BEQL    30$             ; DON'T PASS NULLS THRU
               0110 D5   16   0127   265          JSB     @UCB$L_TT_PUTNXT(R5)    ; BUFFER THE CHARACTER
                    02   13   012B   266          BEQL    30$             ; IF EQL THEN NO CHARACTER TO OUTPUT
                    0A   10   012D   267 20$:     BSBB    CON$STARTIO             ; OUTPUT THE CHARACTER
          50   8E   7D   012F   268 30$:     MOVQ    (SP)+,R0                ; RESTORE REGISTERS
          52   8E   7D   0132   269          MOVQ    (SP)+,R2                ;
          54   8E   7D   0135   270          MOVQ    (SP)+,R4                ;
                    02   0138   271          REI                             ;
                     0139   272
```

```
                     0139   274          .SBTTL  START I/O ON CONSOLE INTERFACE
                     0139   275 ;++
                     0139   276 ; CON$STARTIO - START I/O ON CONSOLE INTERFACE
                     0139   277 ;
                     0139   278 ; FUNCTIONAL DESCRIPTION:
                     0139   279 ;
                     0139   280 ; THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
                     0139   281 ; IF THE INTERFACE IS READY THE DATA IS OUTPUT DIRECTLY. IF THE INTERFACE
                     0139   282 ; IS NOT READY THEN THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE
                     0139   283 ; NEXT READY INTERRUPT.
                     0139   284 ;
                     0139   285 ; IN EITHER CASE, A RETURN TO THE CALLER IS DONE TO ENTER A "WAIT FOR
                     0139   286 ; INTERRUPT" STATE.
                     0139   287 ;
                     0139   288 ; INPUTS:
                     0139   289 ;
                     0139   290 ;       R3 = DATA TO OUTPUT
                     0139   291 ;       R5 = UCB ADDRESS
                     0139   292 ;
                     0139   293 ; OUTPUTS:
                     0139   294 ;
                     0139   295 ;       R3,R4,R5 ARE PRESERVED.
                     0139   296 ;--
                     0139   297
                     0139   298 CON$STARTIO::                                ;
                     0139   299
      08    19       0139   300 10$:    BLSS    20$                         ; BRANCH IF BURST MODE
50    53    9A       013B   301         MOVZBL  R3,R0                       ; GET CHARACTER TO OUTPUT
    00A8    30       013E   302         BSBW    CON$PUTCHAR                 ; OUTPUT CHARACTER
      12    11       0141   303         BRB     30$                         ; CHECK FOR MORE TO OUTPUT
                     0143   304
                     0143   305 20$:
                     0143   306 ;
                     0143   307 ; TAKE CHARACTER OUT OF BURST BUFFER AND TRY TO OUTPUT IT IMMEDIATELY
                     0143   308 ;
50  011C D5    9A    0143   309         MOVZBL  @UCB$L_TT_OUTADR(R5),R0 ; OUTPUT NEXT BYTE
      009E    30     0148   310         BSBW    CON$PUTCHAR                 ; OUTPUT CHARACTER
  011C C5    D6     014B   311         INCL    UCB$L_TT_OUTADR(R5)         ; UPDATE POINTER
  0120 C5    B7     014F   312         DECW    UCB$W_TT_OUTLEN(R5)         ; UPDATE COUNT
        EE    12     0153   313         BNEQ    20$                         ; NOT LAST CHARACTER
64  A5    03    8A   0155   314 30$:    BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
  010C D5    16     0159   315         JSB     @UCB$L_TT_GETNXT(R5)        ; GET THE NEXT CHARACTER
        DA    12     015D   316         BNEQ    10$                         ; IF NEQ MORE CHARACTERS TO OUTPUT
              05     015F   317         RSB                                 ; EXIT
                     0160   318
```

```
                     0160   320          .SBTTL  CONSOLE TRANSMITTER INTERRUPT SERVICE
                     0160   321 ;++
                     0160   322 ; CON$INTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE
                     0160   323 ;
                     0160   324 ; FUNCTIONAL DESCRIPTION:
                     0160   325 ;
                     0160   326 ; THIS ROUTINE IS A NOP FOR QVSS.
                     0160   327 ;--
                     0160   328 CON$INTOUT::                                    ;
50   8E   7D        0160   329          MOVQ    (SP)+,R0                       ; RESTORE REGISTERS
52   8E   7D        0163   330          MOVQ    (SP)+,R2                       ;
54   8E   7D        0166   331          MOVQ    (SP)+,R4                       ;
                02   0169   332          REI                                    ;
                     016A   333
```

```
                        335 .SBTTL  CONSOLE PORT ACTION ROUTINES
                        336 ;++
                        337 ; CON$XOFF -    SEND XOFF
                        338 ; CON$XON -     SEND XON
                        339 ; CON$STOP -    STOP OUTPUT
                        340 ; CON$STOP2 -   ALTERNATE STOP
                        341 ; CON$ABORT -   ABORT CURRENT OUTPUT
                        342 ; CON$RESUME -  RESUME STOPPED OUTPUT
                        343 ;
                        344 ; FUNCTIONAL DESCRIPTION:
                        345 ;
                        346 ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                        347 ; CONTROL OUTPUT ON THE PORT
                        348 ;
                        349 ; INPUTS:
                        350 ;
                        351 ;       R5 = UCB ADDRESS
                        352 ;
                        353 ; OUTPUTS:
                        354 ;
                        355 ;       R5 = UCB ADDRESS
                        356 ;--
                        357
                        358 CON$XOFF::
                        359 CON$XON::
                        360 CON$STOP::
                        361 CON$ABORT::
                        362 CON$RESUME::
05                      363         RSB
                        364
```

```
                     016B   366          .SBTTL  SEND COMMAND TO CONSOLE
                     016B   367
                     016B   368 ;++
                     016B   369 ; CON$SENDCONSCMD - SEND CPU-DEPENDENT COMMAND TO CONSOLE
                     016B   370 ;
                     016B   371 ; FUNCTIONAL DESCRIPTION:
                     016B   372 ;
                     016B   373 ;       INITIATE FUNCTION ON CONSOLE
                     016B   374 ;
                     016B   375 ; INPUTS:
                     016B   376 ;
                     016B   377 ;       R0 = CONSOLE FUNCTION TO PERFORM:
                     016B   378 ;               CON$C_BOOTCPU = SEND REBOOT SIGNAL TO CONSOLE AND THEN HALT
                     016B   379 ;               CON$C_CLRWARM = CLEAR CONSOLE WARMSTART FLAG
                     016B   380 ;               CON$C_CLRCOLD = CLEAR CONSOLE COLDSTART FLAG
                     016B   381 ;       R2 = NUMBER OF BYTES OF DATA TO BE RETURNED (= 0 IF NO DATA EXPECTED)
                     016B   382 ;               (CURRENTLY ONLY IMPLEMENTED IN 11/790 VERSION OF THIS ROUTINE)
                     016B   383 ;       R3 = ADDRESS OF BUFFER TO HOLD RETURNED DATA (ONLY IF R2 IS NON-ZERO)
                     016B   384 ;               (CURRENTLY IMPLEMENTED ONLY IN 11/790 VERSION OF THIS ROUTINE)
                     016B   385 ;
                     016B   386 ; OUTPUTS:
                     016B   387 ;
                     016B   388 ;       CONSOLE STATE MODIFIED
                     016B   389 ;       R1 DESTROYED
                     016B   390 ;--
                     016B   391
                     016B   392 CON$SENDCONSCMD::
50   0F00 8F    A8   016B   393          BISW    #^XF00,R0                       ; SELECT MISCELLANEOUS CONSOLE COMM.
     51     22    DB   0170   394 10$:     MFPR    #PR$_TXCS,R1                    ; GET TRANSMITTER STATUS
  F9 51     07    E1   0173   395          BBC     #7,R1,10$                       ; WAIT FOR CONSOLE READY
     02     50    91   0177   396          CMPB    R0,#CON$C_BOOTCPU               ; REBOOT CPU?
            0B    13   017A   397          BEQL    30$                             ; IF SO BRANCH TO HALT AFTER COMMAND
     23     50    DA   017C   398          MTPR    R0,#PR$_TXDB                    ; OTHERWISE ASSERT COMMAND
     51     22    DB   017F   399 20$:     MFPR    #PR$_TXCS,R1                    ; GET TRANSMITTER STATUS
  F9 51     07    E1   0182   400          BBC     #7,R1,20$                       ; WAIT FOR CONSOLE DONE
                  05   0186   401          RSB                                     ; RETURN
                     0187   402
     23     50    DA   0187   403 30$:     MTPR    R0,#PR$_TXDB                    ; SEND REBOOT COMMAND TO CONSOLE
                  00   018A   404          HALT
```

```
                                                     H 6
                     018B   406          .SBTTL  "ALLOCATE" CONSOLE TERMINAL
                     018B   407
                     018B   408 ;++
                     018B   409 ; CON$OWNCTY - "ALLOCATE" CONSOLE TERMINAL
                     018B   410 ;
                     018B   411 ; FUNCTIONAL DESCRIPTION:
                     018B   412 ;
                     018B   413 ;       THIS ROUTINE SHOULD BE CALLED WHEN PERFORMING NON-INTERRUPT DRIVEN
                     018B   414 ;       I/O TO THE CONSOLE TERMINAL.  IT DISABLES INTERRUPTS AND DOES ANY
                     018B   415 ;       CPU-SPECIFIC INITIALIZATION OF THE CONSOLE TERMINAL REGISTERS.
                     018B   416 ;       CON$RELEASECTY SHOULD BE CALLED TO RESTORE THE STATE OF THE CONSOLE
                     018B   417 ;       TERMINAL INTERFACE REGISTERS.
                     018B   418 ;
                     018B   419 ; INPUTS:
                     018B   420 ;       NONE
                     018B   421 ;
                     018B   422 ; OUTPUTS:
                     018B   423 ;
                     018B   424 ;       R0:     VALUE TO BE RESTORED TO OPACTIVE FLAG WHEN RELEASING CONSOLE TTY
                     018B   425 ;       R1:     VALUE TO BE RESTORED TO INTERRUPT CSR WHEN RELEASING CONSOLE TTY
                     018B   426 ;
                     018B   427 ;       QVSS IS SET UP SO THAT NON-INTERRUPT I/O CAN BE
                     018B   428 ;       PERFORMED TO THE CONSOLE TERMINAL.
                     018B   429 ;
                     018B   430 ;--
                     018B   431 CON$OWNCTY::
      50  00000000'GF   D0   018B   432          MOVL    G^OPA$IDB+IDB$L_CSR,R0          ;GET CSR ADDRESS
          51    60      B0   0192   433          MOVW    QVCSR_CTL(R0),R1                ;SAVE INTERRUPT STATE
      60    0040  8F    AA   0195   434          BICW    #QVCSR$M_ENA_INT,QVCSR_CTL(R0)  ;DISABLE INTERRUPTS
        2A  A0    00    B0   019A   435          MOVW    #0,QVCSR_URTINT(R0)             ;DISABLE UART INTERRUPTS
        24  A0    2A    B0   019E   436          MOVW    #^X2A,QVCSR_URTCMDA(R0)         ;PURGE FIFO
        24  A0    09    B0   01A2   437          MOVW    #9,QVCSR_URTCMDA(R0)            ;ENABLE RECEIVER
  50  00000010'GF    01    02   EF   01A6   438          EXTZV   #OP$V_OPACTIVE,#1,G^OPA$CRB+CRB$L_OPFLAGS,R0;SAVE OPACTIVE FLAG
                                  05   01AF   439          RSB
```

```
                                01B0   441          .SBTTL  RELEASE CONSOLE TERMINAL
                                01B0   442
                                01B0   443 ;++
                                01B0   444 ; CON$RELEASECTY - RELEASE CONSOLE TERMINAL
                                01B0   445 ;
                                01B0   446 ; FUNCTIONAL DESCRIPTION:
                                01B0   447 ;
                                01B0   448 ;       THIS ROUTINE SHOULD BE CALLED TO RELINQUISH EXCLUSIVE USE OF THE
                                01B0   449 ;       CONSOLE TERMINAL OBTAINED BY CALLING CON$OWNCTY.  IT RESTORES THE
                                01B0   450 ;       STATE OF THE CONSOLE.
                                01B0   451 ;
                                01B0   452 ; INPUTS:
                                01B0   453 ;       R0:     VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO OPACTIVE FLAG
                                01B0   454 ;       R1:     VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO INTERRUPT CSR
                                01B0   455 ;
                                01B0   456 ; OUTPUTS:
                                01B0   457 ;       QVSS AND OPACTIVE FLAG ARE RESTORED TO THEIR ORIGINAL STATE.
                                01B0   458 ;
                                01B0   459 ;--
                                01B0   460 CON$RELEASECTY::
00000010'GF   01    02    50    F0  01B0   461          INSV    R0,#OP$V_OPACTIVE,#1,G^OPA$CRB+CRB$L_OPFLAGS; RESTORE OPACTIVE FLAG
         50    00000000'GF    D0  01B9   462          MOVL    G^OPA$IDB+IDB$L_CSR,R0          ;GET CSR ADDRESS
               24    A0    2A    B0  01C0   463          MOVW    #^X2A,QVCSR_URTCMDA(R0)         ;PURGE INPUT FIFO
               24    A0    09    B0  01C4   464          MOVW    #9,QVCSR_URTCMDA(R0)            ;ENABLE RECEIVER
               2A    A0    02    B0  01C8   465          MOVW    #02,QVCSR_URTINT(R0)            ;ENABLE UART INTERRUPTS
                  60    51    B0  01CC   466          MOVW    R1,QVCSR_CTL(R0)                ;RESTORE INTERRUPT STATE
                        05    01CF   467          RSB
                              01D0   468
```

```
                                                J 6
                     01D0   470          .SBTTL  - GET A CHARACTER FROM THE CONSOLE TERMINAL
                     01D0   471 ;++
                     01D0   472 ; CON$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
                     01D0   473 ;
                     01D0   474 ; FUNCTIONAL DESCRIPTION:
                     01D0   475 ;
                     01D0   476 ;       THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
                     01D0   477 ;       DIRECTLY TO THE CONSOLE TERMINAL
                     01D0   478 ;
                     01D0   479 ; INPUTS:
                     01D0   480 ;       None
                     01D0   481 ;
                     01D0   482 ; OUTPUTS:
                     01D0   483 ;       R0 contains the character.
                     01D0   484 ;
                     01D0   485 ;--
           00000013  01D0   486          control_s = 19                  ; control s (xoff)
           00000011  01D0   487          control_q = 17                  ; control q (xon)
           00000001  01D0   488          qvuart$m_rxrdy = 1              ; receiver ready bit
                     01D0   489
                     01D0   490 CON$GETCHAR::
50 00000000'GF   D0  01D0   491 5$:     movl    g^opa$idb+idb$l_csr,r0  ;get qvss csr address
    22 A0    01   B3  01D7   492 10$:    bitw    #qvuart$m_rxrdy,qvcsr_urtstata(r0);receiver ready?
             FA   13  01DB   493         beql    10$                     ;if eql not ready
    50   26 A0   9A  01DD   494         movzbl  qvcsr_urtbufa(r0),r0    ;get character scan code
          FE1C'  30  01E1   495         bsbw    qvss$keydecode          ;decode the lk201 input data
            50   D5  01E4   496         tstl    r0                      ;need more input?
            E8   13  01E6   497         beql    5$                      ;if eql yes
                 05  01E8   498         rsb                             ;return
                     01E9   499
```

```
                                01E9    501          .SBTTL  - PUT A CHARACTER OUT ON THE CONSOLE TERMINAL
                                01E9    502 ;++
                                01E9    503 ; CON$PUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL
                                01E9    504 ;
                                01E9    505 ; FUNCTIONAL DESCRIPTION:
                                01E9    506 ;
                                01E9    507 ;       THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
                                01E9    508 ;       DIRECTLY TO THE CONSOLE TERMINAL
                                01E9    509 ;
                                01E9    510 ; INPUTS:
                                01E9    511 ;       R0 - Character to be output
                                01E9    512 ;
                                01E9    513 ; OUTPUTS:
                                01E9    514 ;       Character written to the console terminal.
                                01E9    515 ;
                                01E9    516 ;--
                                01E9    517          .enabl  lsb
                                01E9    518 CON$PUTCHAR::
                          3E    BB    01E9    519          pushr   #^m<r1,r2,r3,r4,r5>     ;save registers
              53  00000000'GF    DE    01EB    520          moval   g^opa$crb,r3           ;get crb address
           0F 10 A3    02     E0    01F2    521          bbs     #op$v_opactive,crb$l_opflags(r3),1$;continue if we have control of t
 03 00000000'GF  00000000'8F     E0    01F7    522          bbs     #exe$v_opa0,g^exe$gl_wsflags,1$;output to opa0 enabled? bs yes
                       0030    31    0203    523          brw     80$                     ;return
                         50    DD    0206    524 1$:      pushl   r0                      ;save r0
           10 10 A3    00     E5    0208    525          bbcc    #op$v_reinit,crb$l_opflags(r3),2$;reinit the scan lines?
1C B3   7800 8F     00    6E    00    2C    020D    526          movc5   #0,(sp),#0,#24*128*10,@crb$l_video_base(r3); init memory
              53  00000000'GF    DE    0216    527          moval   g^opa$crb,r3            ;get crb address
           03 10 A3    01     E5    021D    528 2$:      bbcc    #op$v_remap,crb$l_opflags(r3),3$;remap scan lines to screen?
                       0096    30    0222    529          bsbw    remap                   ;remap scan lines to the screen
                         50  8ED0    0225    530 3$:      popl    r0                      ;restore r0
              53    1C A3     D0    0228    531          movl    crb$l_video_base(r3),r3 ;get va of video memory
          54  00000000'GF     D0    022C    532          movi    g^opa$idb+idb$l_csr,r4  ;get va of qvss csr
                    FDCA'    30    0233    533          bsbw    qvss$putchar            ;output the character
                       3E    BA    0236    534 80$:     popr    #^m<r1,r2,r3,r4,r5>     ;restore registers
                       05    0238    535          rsb                             ;
                             0239    536          .dsabl  lsb
```

```
                                0239   538          .SBTTL  - INITIALIZE CONSOLE TERMINAL FOR NON-INTERRUPT DRIVEN I/O
                                0239   539 ;++
                                0239   540 ; CON$INIT_CTY - INITIALIZE QVSS FOR NON-INTERRUPT DRIVEN I/O
                                0239   541 ;
                                0239   542 ; FUNCTIONAL DESCRIPTION:
                                0239   543 ;
                                0239   544 ;       THIS ROUTINE MUST BE CALLED FROM INIT BEFORE ANY CONSOLE TERMINAL I/O
                                0239   545 ;       CAN OCCUR.
                                0239   546 ;
                                0239   547 ; INPUTS:
                                0239   548 ;
                                0239   549 ; OUTPUTS:
                                0239   550 ;       VIDEO MEMORY MAPPED.
                                0239   551 ;       I/O SPACE THAT CONATINS THE CSRS FOR QVSS MAPPED.
                                0239   552 ;
                                0239   553 ;--
                                0239   554 CON$INIT_CTY::                          ; INITIALIZE QVSS CONTROLLER
                                0239   555
                                0239   556          ;
                                0239   557          ; SAVE REGISTERS
                                0239   558          ;
                          1E    0239   559          PUSHR   #^M<R1,R2,R3,R4>
                                023B   560
                                023B   561          ;
                                023B   562          ; INITIALIZE THE KEYBOARD TRANSLATION TABLE
                                023B   563          ;
00000000'EF   FFFFFF90'EF   DE  023B   564          MOVAL   QVSS$KEY-112,QVSS$KEYTABLE
                                0246   565
                                0246   566          ;
                                0246   567          ; Initialize OPDRVWS1 state flags
                                0246   568          ;
          53  00000000'GF      DE  0246   569          MOVAL   G^OPA$CRB,R3            ; GET THE CRB ADDRESS
              10 A3     04      D0  024D   570          MOVL    #OP$M_OPACTIVE,CRB$L_OPFLAGS(R3); OPACTIVE,NOREINIT,NOREMAP,NOVCACTI
                                0251   571
                                0251   572          ;
                                0251   573          ; GET VIRTUAL ADDRESS OF CSR IN R4
                                0251   574          ;
          52  00000000'GF      D0  0251   575          MOVL    G^BOO$GL_SPTFREL,R2     ; GET A FREE SPT
              00000000'GF      D6  0258   576          INCL    G^BOO$GL_SPTFREL        ;
          51  00000000'GF      D0  025E   577          MOVL    G^MMG$GL_SPTBASE,R1     ; GET VA OF SYSTEM SPT BASE
                51    6142    DE  0265   578          MOVAL   (R1)[R2],R1             ; GET VA OF SPT PTE
            52    52    09    78  0269   579          ASHL    #9,R2,R2                ; MAKE VA
          52  80000000  8F      C8  026D   580          BISL    #VA$M_SYSTEM,R2         ; SET SYSTEM SPACE BIT
    54    52  00000080  8F      C1  0274   581          ADDL3   #QVCSR_BOFF,R2,R4       ; CALC CSR VIRTUAL ADDRESS
          00000000'GF     54    D0  027C   582          MOVL    R4,G^OPA$IDB+IDB$L_CSR  ; SAVE IN IDB
          61  9010000F    8F    D0  0283   583          MOVL    #<PTE$M_VALID!PTE$C_KW!QVCSR_PFN>,(R1); MAP PA OF CSR TO SYS VA
                                028A   584
                                028A   585          ;
                                028A   586          ; Map Video Memory
                                028A   587          ;
                      51        D4  028A   588          CLRL    R1                      ;
        51    64    87FF      8F  AB  028C   589          BICW3   #^C<QVCSR$M_MEMBANK>,(R4),R1; GET BASE QVSS MEMORY BANK
          51    51    F5    8F  78  0292   590          ASHL    #-QVCSR$V_MEMBANK,R1,R1 ; MAKE IT ZERO BASE
          51  00040000      8F  C4  0297   591          MULL2   #^X40000,R1             ; COMPUTE 256K BANK
          50    51    F7    8F  78  029E   592          ASHL    #-9,R1,R0               ; AND ISOLATE PFN
                51    0200  8F  3C  02A3   593          MOVZWL  #512,R1                 ; # OF PAGES
                          2C  10  02A8   594          BSBB    MAP_PAGES               ; MAP VIDEO RAM
```

```
                 08 50   E9  02AA   595          BLBC    R0,100$                         ; NO SPTS THEN EXIT
              1C A3 52   D0  02AD   596          MOVL    R2,CRB$L_VIDEO_BASE(R3) ; SAVE STARTING VA OF BITMAP
                 08      10  02B1   597          BSBB    REMAP                           ; MAP THE SCREEN
           50    0000'8F   3C  02B3   598          MOVZWL  #SS$_NORMAL,R0                  ; INDICATE SUCCESS
                 1E      BA  02B6   599 100$:     POPR    #^M<R1,R2,R3,R4>
                 05      10  02BA   600          RSB
                         02BB   601
```

```
                                02BB   603          .SBTTL  REMAP - MAP VIDEO RAM TO THE SCREEN
                                02BB   604 ;++
                                02BB   605 ; REMAP
                                02BB   606 ;
                                02BB   607 ; Map the first 24 scan lines to the screen.
                                02BB   608 ;
                                02BB   609 ; Inputs:
                                02BB   610 ;
                                02BB   611 ;       R3 - CRB Address for OPA0
                                02BB   612 ;
                                02BB   613 ; Outputs:
                                02BB   614 ;       R0,R1,R2 destroyed
                                02BB   615 ;
                                02BB   616 ; Implicit Outputs:
                                02BB   617 ;       Scan lines map video ram to the screen.
                                02BB   618 ;
                                02BB   619 ; Side Effects:
                                02BB   620 ;       None.
                                02BB   621 ;
                                02BB   622 ;--
                                02BB   623 REMAP:
                                02BB   624
                                02BB   625 ;       INIT THE SCAN MAP AND MAP SCREEN FULL OF LINES
                                02BB   626
52   1C A3   0003F800 8F   C1   02BB   627          ADDL3   #QVSCAN_MAP,CRB$L_VIDEO_BASE(R3),R2; COMPUTE ADDRESS OF SCAN MAP
                                02C4   628
                          50    D4   02C4   629 5$:     CLRL    R0
                  51    000000F0 8F   D0   02C6   630          MOVL    #24*10,R1               ; NUMBER OF SCAN MAPS
                        82      50    B0   02CD   631 10$:    MOVW    R0,(R2)+
                                50    D6   02D0   632          INCL    R0
                              F8 51   F5   02D2   633          SOBGTR  R1,10$
                                      05   02D5   634          RSB
                                           02D6   635
```

```
                     02D6   637          .SBTTL  MAP_PAGES - MAP PHYSICALLY-CONTIGUOUS PAGES
                     02D6   638 ;++
                     02D6   639 ; MAP_PAGES
                     02D6   640 ;
                     02D6   641 ; Map to system virtual address space N physically-contiguous pages.
                     02D6   642 ;
                     02D6   643 ; Inputs:
                     02D6   644 ;       R1 = N = number of physically-contiguous pages
                     02D6   645 ;       R0 = Starting PFN
                     02D6   646 ;
                     02D6   647 ; Outputs:
                     02D6   648 ;       R0 = status: SUCCESS, INSFMEM, INSFSPTS
                     02D6   649 ;       R1 = preserved
                     02D6   650 ;       R2 = system virtual address of N pages of memory if success
                     02D6   651 ;       all other registers preserved
                     02D6   652 ;
                     02D6   653 ; Implicit Outputs:
                     02D6   654 ;       None.
                     02D6   655 ;
                     02D6   656 ; Side Effects:
                     02D6   657 ;       IOC$ALLOSPT called - so SPTs are allocated
                     02D6   658 ;
                     02D6   659 ;--
                     02D6   660 MAP_PAGES:
              3A BB  02D6   661          PUSHR   #^M<R1,R3,R4,R5>        ; save work registers
                     02D8   662                                          ;  r1 = input used as loop counter
                     02D8   663                                          ;  r3 = address of SPT
                     02D8   664                                          ;  r4 = index into PFN database
                     02D8   665                                          ;  r5 = temp storage
           55 50 D0  02D8   666          MOVL    R0,R5                   ; Save starting PFN
 00000000'GF 16      02DB   667          JSB     G^IOC$ALLOSPT           ; allocate N SPTs to map VAs
         49 50 E9    02E1   668          BLBC    R0,30$                  ; if LBC, no system page table slots
                     02E4   669 ;
                     02E4   670 ; IOC$ALLOSPT returns:
                     02E4   671 ;       R1 = preserved, R2 = SVPN (index into SPT), R3 = address of SPT
                     02E4   672 ;
                     02E4   673 ; The main loop indexes backwards through the system page table entries
                     02E4   674 ; and backwards through the PFN database.  It goes backwards so that the
                     02E4   675 ; last system virtual address calculated can be returned to the caller.
                     02E4   676 ;
                     02E4   677 ;                                          ;  r0 = index into SPT
      50 52 51 C1    02E4   678          ADDL3   R1,R2,R0                ; start at last SPT and go backwards
      54 51 55 C1    02E8   679          ADDL3   R5,R1,R4                ; start PFNs at end in loop
                     02EC   680 10$:                                     ; set up system page-table entry
            50 D7    02EC   681          DECL    R0                      ; back up SVPN index
            54 D7    02EE   682          DECL    R4                      ; back up PFN index
         6340 54 D0  02F0   683          MOVL    R4,(R3)[R0]             ; fill PFN in SPT
6340 A0000000 8F C8  02F4   684          BISL2   #<PTE$C_UW!PTE$M_VALID>,(R3)[R0] ; user mode access, valid
                     02FC   685          CPUDISP <<UV1,12$>,-            ; *Dispatch on CPU type*
                     02FC   686                  <UV2,11$>>
6340 00180000 8F C8  030C   687 11$:    BISL2   #^X180000,(R3)[R0]      ; indicate QBUS memory for UV2
                     0314   688 ;
                     0314   689 ; Invalidate system virtual address
                     0314   690 ;
      52 50 09 78    0314   691 12$:    ASHL    #9,R0,R2                ; turn SVPN into VA
   52 80000000 8F C8 0318   692          BISL2   #<1@31>,R2              ; make VA a system VA
                     031F   693          INVALID R2                      ; and clear translation buffer
```

```
                C7 51    F5  0322   694          SOBGTR  R1,10$                  ; loop N times
                         0325   695
50   0000'8F    3C       0325   696          MOVZWL  #SS$_NORMAL,R0           ; indicates success (R2 has system VA)
           3A    BA       032A   697 15$:     POPR    #^M<R1,R3,R4,R5>         ; restore work registers
                 05       032C   698          RSB                              ;
50   0000'8F    3C       032D   699 30$:     MOVZWL  #SS$_INSFSPTS,R0         ; no SPTs left
           F6    11       0332   700          BRB     15$                      ; return
                          0334   701
                          0334   702 CON_END:
                          0334   703          .END
```

```
$$BASE                  =  00000007
$$DISPL                 =  00000009
$$GENSW                 =  00000001
$$HIGH                  =  00000008
$$LIMIT                 =  00000001
$$LOW                   =  00000007
$$MNSW                  =  00000001
$$MXSW                  =  00000001
ADP$L_VECTOR            =  00000010
BIT...                  =  00000006
BOO$GL_SPTFREL             ********    X   02
BUG$_UNSUPRTCPU            ********    X   02
CLASS_DDT               =  00000010
CLASS_GETNXT            =  00000000
CLASS_POWERFAIL         =  00000020
CLASS_PUTNXT            =  00000004
CLASS_SETUP_UCB         =  00000008
CON$ABORT                  0000016A RG     02
CON$C_BOOTCPU           =  00000002
CON$DISCONNECT             000000F6 RG     02
CON$DS_SET                 000000F5 RG     02
CON$GETCHAR                000001D0 RG     02
CON$INITIAL                00000000 RG     02
CON$INITLINE               00000069 RG     02
CON$INIT_CTY               00000239 RG     02
CON$INTINP                 0000010B RG     02
CON$INTOUT                 00000160 RG     02
CON$NULL                   000000F5 RG     02
CON$OWNCTY                 0000018B RG     02
CON$PUTCHAR                000001E9 RG     02
CON$RELEASECTY             000001B0 RG     02
CON$RESUME                 0000016A RG     02
CON$SENDCONSCMD            0000016B RG     02
CON$SET_LINE               000000F5 RG     02
CON$SET_MODEM              000000F5 RG     02
CON$STARTIO                00000139 RG     02
CON$STOP                   0000016A RG     02
CON$VCINP                  00000115 RG     02
CON$XOFF                   0000016A RG     02
CON$XON                    0000016A RG     02
CONTROL_Q               =  00000011
CONTROL_S               =  00000013
CON_END                    00000334 R      02
CRB$L_AUXSTRUC          =  00000010
CRB$L_INTD              =  00000024
CRB$L_OPFLAGS           =  00000010
CRB$L_SCAN_MAP          =  00000014
CRB$L_TIMELINK          =  00000014
CRB$L_TOUTROUT          =  0000001C
CRB$L_VIDEO_BASE        =  0000001C
DDB$L_DDT               =  0000000C
DPT$W_VECTOR            =  0000001E
EXE$GB_CPUTYPE             ********    X   02
EXE$GL_WSFLAGS             ********    X   02
EXE$V_OPA0                 ********    X   02
IDB$L_CSR               =  00000000
IDB$L_UCBLST            =  00000018
IOC$ALLOSPT                ********    X   02
IOC$GL_ADPLIST             ********    X   02
IOUV1$AL_QBOSP          =  20000000
KEY$M_APPKEYPAD         =  00000001
KEY$M_BUTTOG            =  00000020
KEY$M_CTRL              =  00000010
KEY$M_HOLD              =  00000002
KEY$M_LOCK              =  00000004
KEY$M_SHIFT             =  00000008
KEY$V_APPKEYPAD         =  00000000
KEY$V_BUTTOG            =  00000005
KEY$V_CTRL              =  00000004
KEY$V_HOLD              =  00000001
KEY$V_LOCK              =  00000002
KEY$V_SHIFT             =  00000003
MAP_PAGES                  000002D6 R      02
MMG$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
OPA$CRB                    ********    X   02
OPA$IDB                    ********    X   02
OPA$VECTOR                 ********    X   02
PR$_SID_TYPUV1          =  00000007
PR$_SID_TYPUV2          =  00000008
PR$_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$V_MODE19                  = 00000000
QVCSR_BOFF                      = 00000080  G
QVCSR_CRTADDR                   = 00000008  G
QVCSR_CRTDATA                   = 0G00000A  G
QVCSR_CTL                       = 00000000  G
QVCSR_CURPOS                    = 00000002  G
QVCSR_INTCTL                    = 0000000E  G
QVCSR_INTDATA                   = 0000000C  G
QVCSR_MOUSE                     = 00000004  G
QVCSR_OFFSET                    = 00001E80  G
QVCSR_PA                        = 20001E80  G
QVCSR_PFN                       = 0010000F  G
QVCSR_SPARE                     = 00000006  G
QVCSR_URTBUFA                   = 00000026  G
QVCSR_URTCMDA                   = 00000024  G
QVCSR_URTINT                    = 0000002A  G
QVCSR_URTMODEA                  = 00000020  G
QVCSR_URTSTATA                  = 00000022  G
QVSCAN_MAP                      = 0003F800  G
QVSCTLBLOCK                     = 0003F700  G
QVSCUR_RAM                      = 0003FFE0  G
QVSS$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                           000002BB R      02
SCAN_MAP                        = 0003F800  G
SIZ...                          = 00000001
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$B_TT_DEPARI                 = 000000EC
UCB$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
VA$M_SYSTEM                     = 80000000
VEC$Q_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
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000334 (820.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	71	00:00:00.32	00:00:01.40
Command processing	92	00:00:01.27	00:00:02.91
Pass 1	535	00:00:23.21	00:00:55.37
Symbol table sort	0	00:00:03.34	00:00:06.85
Pass 2	132	00:00:04.09	00:00:09.87
Symbol table output	21	00:00:00.21	00:00:00.22
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	855	00:00:32.49	00:01:16.70

The working set limit was 1800 pages.
120663 bytes (236 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2220 non-local and 32 local symbols.
759 source lines were read in Pass 1, producing 17 object records in Pass 2.
69 pages of virtual memory were used to define 64 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA18:[SYS.OBJ]LIB.MLB;1	30
_255DUA18:[SYSLIB]STARLET.MLB;3	9
TOTALS (all libraries)	39

2640 GETS were required to define 39 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:OPDRVWS/OBJ=OBJ$:OPDRVWS MSRC$:QVSCSR/UPDATE=(BUG$:QVSCSR)+MSRC$:OPDRVWS/UPDATE=(BUG$:OPDRVWS)+EXECML$/LIB
```

0448 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

CSP
LIS

TTYCHARI
LIS

TTDRVR

OPDRVWS
LIS

TTDRIVER
MAP

YCDRIVER
MAP

CSPQUORUM
LIS

TTYCHARO
LIS