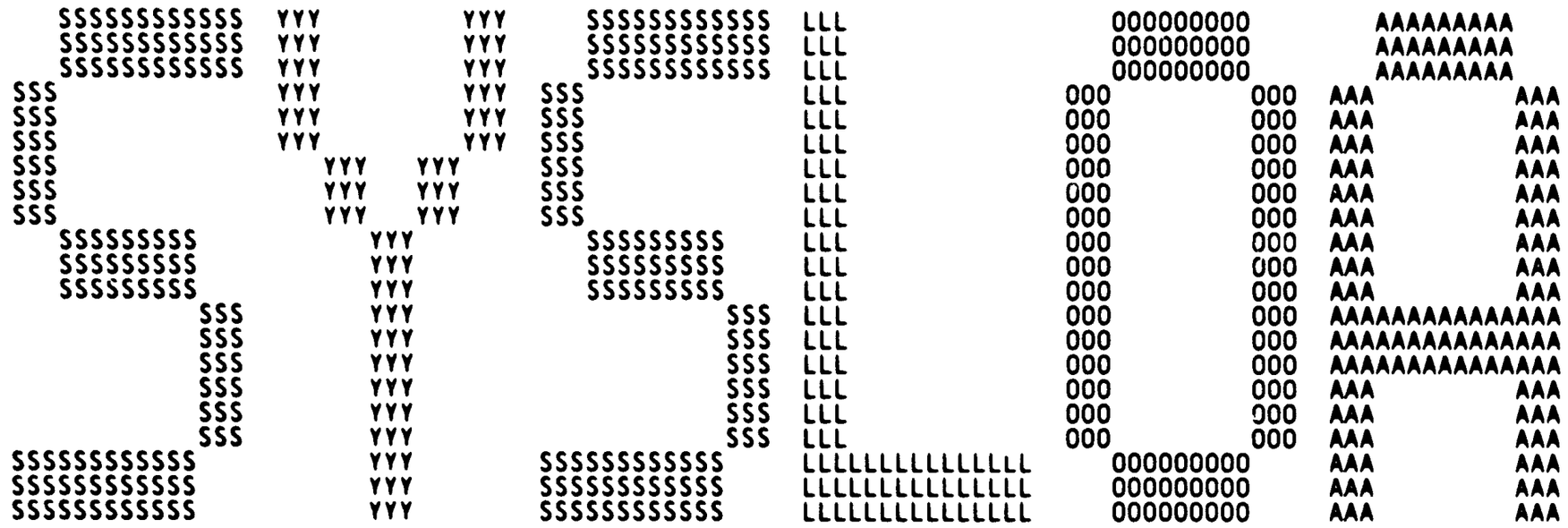
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSSSSSSSS YYY SSSSSSSSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAAAAAAAAAAAAAA
SSS YYY SSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAA AAA
SSS YYY SSS LLL 000 000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA
SSSSSSSSSSS YYY SSSSSSSSSSS LLLLLLLLLLLLLLL 000000000 AAA AAA

FILEID**OPDRVWS1

```
  OOOOOO    PPPPPPPP    DDDDDDDD    RRRRRRRR    VV      VV  WW      WW    SSSSSSSS      11
  OOOOOO    PPPPPPPP    DDDDDDDD    RRRRRRRR    VV      VV  WW      WW    SSSSSSSS      11
OO      OO  PP      PP  DD      DD  RR      RR  VV      VV  WW      WW  SS            1111
OO      OO  PP      PP  DD      DD  RR      RR  VV      VV  WW      WW  SS            1111
OO      OO  PP      PP  DD      DD  RR      RR  VV      VV  WW      WW  SS              11
OO      OO  PP      PP  DD      DD  RR      RR  VV      VV  WW      WW  SS              11
OO      OO  PPPPPPPP    DD      DD  RRRRRRRR    VV      VV  WW      WW    SSSSSS        11
OO      OO  PPPPPPPP    DD      DD  RRRRRRRR    VV      VV  WW      WW    SSSSSS        11
OO      OO  PP          DD      DD  RR  RR      VV      VV  WW  WW  WW          SS      11
OO      OO  PP          DD      DD  RR  RR      VV      VV  WW  WW  WW          SS      11
OO      OO  PP          DD      DD  RR    RR      VV  VV    WWWW  WWWW          SS      11          ....
OO      OO  PP          DD      DD  RR    RR      VV  VV    WWWW  WWWW          SS      11          ....
  OOOOOO    PP          DDDDDDDD    RR      RR      VV      WW      WW  SSSSSSSS      111111        ....
  OOOOOO    PP          DDDDDDDD    RR      RR      VV      WW      WW  SSSSSSSS      111111        ....


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     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
```

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

```
0000     88
0000     89 ;       UCB$W_QV_KEYSTATE DEFINITIONS
0000     90
0000     91         $VIELD  KEY,0,<-
0000     92                 <APPKEYPAD,,M>-
0000     93                 <HOLD,,M>-
0000     94                 <,,M>-
0000     95                 <SHIFT,,M>-
0000     96                 <CTRL,,M>-
0000     97                 <BUTTOG,,M>-            ; MOUSE BUTTON SAMPLE TOGGLE
0000     98                 >
0000     99
0000    100 ;       MAIN QVSS CSR BIT DEFINITIONS
0000    101
0000    102         $VIELD  QVCSR,0,<-              ;
0000    103                 <MODE19,1,M>-           ; (R)   15 OR 19 INCH MONITOR (1=19)
0000    104                 <,1>-                   ;       SPARE
0000    105                 <ENA_VIDEO,1,M>-        ; (RW)  ENABLE VIDEO
0000    106                 <CURS_FNC,1,M>-         ; (RW)  CURSOR (1=OR, 0=AND)
0000    107                 <,2>-                   ; (RW)  DIAG FUNCTIONS
0000    108                 <ENA_INT,1,M>-          ; (RW)  ENABLE INTERRUPT
0000    109                 <,1>-                   ; (R)   DIAG
0000    110                 <BUTA,1,M>-             ; (R)   MOUSE BUTTON A (RIGHT)
0000    111                 <BUTB,1,M>-             ; (R)   MOUSE BUTTON B (MIDDLE)
0000    112                 <BUTC,1,M>-             ; (R)   MOUSE BUTTON C (LEFT)
0000    113                 <MEMBANK,4,M>-          ; (R)   VIDEO MEMORY BASE SELECT
0000    114                 >
0000    115
0000    116
```

```
                     0000   118
                     0000   119          .SBTTL  REGISTER DEFINITIONS
                     0000   120
                     0000   121 ;        VIDEO RAM LAYOUT (0-3FFFF)
                     0000   122
0003F700             0000   123 QVSVIDEO_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_URTBUFA   == 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    == ^O17200                ; 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   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
00000000'EF   FFFFFF90'EF   DE  0000   179          MOVAL   QVSS$KEY-112,QVSS$KEYTABLE  ; INITIALIZE THE KEYBOARD TRANSLATIO
                                000B   180
                          52    DD  000B   181          PUSHL   R2                          ; SAVE R2
                00000000'GF   D0  000D   182          MOVL    G^IOC$GL_ADPLIST,R2         ; GET ADP ADDRESS
                  30    10 A2   C1  0014   183          ADDL3   ADP$L_VECTOR(R2),#^060,R2   ; GET ADDR OF VECTOR TABLE ENTRY
                00000025'GF   9E  0019   184          MOVAB   G^OPA$CRB+CRB$L_INTD+VEC$Q_DISPATCH+1,(R2); CONNECT THE VECTOR
                                0020   185 ;
                                0020   186 ;       SET UP INTERRUPTS
                                0020   187 ;
                00    A4    90  0020   188          MOVB    #0,QVCSR_INTCTL(R4)         ; RESET INTERRUPT CONTROLLER
                40    A4    90  0024   189          MOVB    #^X40,QVCSR_INTCTL(R4)      ; RESET IRR
                80    A4    90  0029   190          MOVB    #^X80,QVCSR_INTCTL(R4)      ; SPECIFY INDIVIDUAL VECTORS
                C0    A4    90  002E   191          MOVB    #^XC0,QVCSR_INTCTL(R4)      ; PRESET AUTOCLEAR DATA
                FF    A4    90  0033   192          MOVB    #^XFF,QVCSR_INTDATA(R4)     ; ALL ARE AUTO CLEAR
                                0038   193
                                0038   194 ;       VECTOR SPECIFIC
                                0038   195
                E0    A4    90  0038   196          MOVB    #^XE0,QVCSR_INTCTL(R4)      ; PRESET VECTOR ADDRESS (ONE)
                30    A4    90  003D   197          MOVB    #^060,QVCSR_INTDATA(R4)     ; USE SPECIAL VECTOR
                28    A4    90  0041   198          MOVB    #^X28,QVCSR_INTCTL(R4)      ; ENABLE TX/RX INTERRUPT
                A1    A4    90  0045   199          MOVB    #^XA1,QVCSR_INTCTL(R4)      ; ARM THE INTERRUPT CONTROLER CHIP
                                004A   200
                                004A   201
                                004A   202 ;
                                004A   203 ; SET UP UART
                                004A   204 ;
                                004A   205
                19    A4    B0  004A   206          MOVW    #^X19,QVCSR_URTCMDA(R4)     ; RESET MODE POINTER, ENABLE RCV, DI
                17    A4    B0  004E   207          MOVW    #^X17,QVCSR_URTMODEA(R4)    ; SET MODE 1 ,NOPARITY, 8 BIT
                07    A4    B0  0052   208          MOVW    #^X07,QVCSR_URTMODEA(R4)    ; SET MODE 2 , 1 STOP BIT
          0099    8F 22 A4    B0  0056   209          MOVW    #^X99,QVCSR_URTSTATA(R4)    ; 4800 BAUD XMIT, RCV
                02    A4    B0  005C   210          MOVW    #^X02,QVCSR_URTINT(R4)      ; ENABLE REC INTERRUPTS
                                0060   211
          0044    8F    A8  0060   212          BISW    #<QVCSR$M_ENA_VIDEO!QVCSR$M_ENA_INT>,-; ENABLE VIDEO
                64          0064   213                  QVCSR_CTL(R4)                  ; INTERRUPTS AND CURSOR=AND
                      52 8ED0   0065   214          POPL    R2                          ; RESTORE R2
                         05     0068   215          RSB
```

```
                                0069   217          .SBTTL  CONSOLE UNIT INITIALIZATION
                                0069   218 ;++
                                0069   219 ; CON$INITIAL - INITIALIZE CONSOLE UNIT
                                0069   220 ;
                                0069   221 ; FUNCTIONAL DESCRIPTION:
                                0069   222 ;
                                0069   223 ; THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.
                                0069   224 ;
                                0069   225 ; INPUTS:
                                0069   226 ;
                                0069   227 ;       R5 = UCB ADDRESS
                                0069   228 ;       R9 = CRB ADDRESS
                                0069   229 ;
                                0069   230 ; OUTPUTS:
                                0069   231 ;
                                0069   232 ;       ALL REGISTERS ARE PRESERVED.
                                0069   233 ;--
                                0069   234 CON$INITLINE::
                          50    DD  0069   235          PUSHL   R0                      ; SAVE R0
          50    00000000'GF   DE  006B   236          MOVAL   G^OPA$VECTOR,R0         ; GET THE VECTOR ADDRESS
                                0072   237          CLASS_UNIT_INIT                 ; AND INIT THIS UNIT
           50    0114 C5    D0  00BB   238          MOVL    UCB$L_TT_CLASS(R5),R0   ; ADDRESS OF CLASS VECTOR TABLE
                   08 B0    16  00C0   239          JSB     @CLASS_SETUP_UCB(R0)    ; INITIALIZE THE UCB FOR CONSOLE TERMINAL
        08 64 A5    05    E1  00C3   240 30$:     BBC     #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
           50    0114 C5    D0  00C8   241          MOVL    UCB$L_TT_CLASS(R5),R0   ; GET THE CLASS VECTOR TABLE ADDRESS
                   20 B0    16  00CD   242          JSB     @CLASS_POWERFAIL(R0)    ; AND GOTO THE POWERFAIL CODE
                                00D0   243
                          50 8ED0  00D0   244 40$:     POPL    R0                      ; RESTORE R0
                                00D3   245
                    41 A5    00    90  00D3   246          MOVB    #TT$_UNKNOWN,UCB$B_DEVTYPE(R5); SET UNKNOWN TERMINAL TYPE
                44 A5   00001000 8F    C8  00D7   247          BISL    #TT$M_SCOPE,UCB$L_DEVDEPEND(R5); QVSS IS SCOPE
                48 A5   00001000 8F    C8  00DF   248          BISL    #TT2$M_EDITING,UCB$L_DEVDEPND2(R5); ENABLE LINE EDITING
                       21000000 8F    CA  00E7   249          BICL    #<TT2$M_ANSICRT!TT2$M_DECCRT>,-; THIS DRIVER DOES NOT
                              48 A5        00ED   250                  UCB$L_DEVDEPND2(R5)     ; EMULATE VT100'S
                00C4 C5    44 A5    7D  00EF   251          MOVQ    UCB$L_DEVDEPEND(R5),UCB$L_TT_DECHAR(R5); MAKE PERMANENT
                                00F5   252
                                00F5   253 CON$SET_LINE::
                                00F5   254 CON$DS_SET::
                                00F5   255 CON$SET_MODEM::
                                00F5   256 CON$NULL::
                                   05  00F5   257          RSB                     ;
                                00F6   258 CON$DISCONNECT::                        ; CALLED ON LAST DEASSIGN
                                   3F  BB  00F6   259          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; SAVE REGISTERS
                53    00000000'GF   DE  00F8   260          MOVAL   G^OPA$CRB,R3            ; GET CRB ADDRESS
                   3D 10 A3    03    E1  00FF   261          BBC     #OP$V_VCACTIVE,CRB$L_OPFLAGS(R3),20$;BC IF VCDRIVER NOT INITED
2D 00000000'GF    00000000'8F    E1  0104   262          BBC     #EXE$V_OPA0,G^EXE$GL_WSFLAGS,10$;  ALL DONE WITH OPA0? IF BC YES
                         14 A3    D5  0110   263          TSTL    CRB$L_SCAN_MAP(R3)      ; SCAN MAP SAVE AREA ALREADY ALLOCATED?
                            28    12  0113   264          BNEQ    10$                     ; IF NEQ YES ALL DONE
                      51   01E0 8F    3C  0115   265          MOVZWL  #24*10*2,R1             ; ALLOCATE SAVE AREA FOR 24 X 10 SCAN LINE M
                       00000000'GF    16  011A   266          JSB     G^EXE$ALONONPAGED       ; GET THE MEMORY
                                1A    50  E9  0120   267          BLBC    R0,10$                  ; BRANCH IF ERROR
                         14 A3       52  D0  0123   268          MOVL    R2,CRB$L_SCAN_MAP(R3)   ; SAVE SCAN MAP SAVE AREA ADDRESS
          51    1C A3   0003F800 8F    C1  0127   269          ADDL3   #QVSCAN_MAP,CRB$L_VIDEO_BASE(R3),R1; COMPUTE ADDRESS OF SCAN MAP
                62    61   01E0 8F    28  0130   270          MOVC3   #24*10*2,(R1),(R2)      ; INIT SCAN MAP
                53    00000000'GF    DE  0136   271          MOVAL   G^OPA$CRB,R3            ; GET CRB ADDRESS
                      10 A3    04    CA  013D   272 10$:     BICL    #OP$M_OPACTIVE,CRB$L_OPFLAGS(R3); CLEAR OPACTIVE FLAG
                                3F    BA  0141   273 20$:     POPR    #^M<R0,R1,R2,R3,R4,R5>  ; RESTORE REGISTERS
```

```
05  0143   274          RSB                          ; RETURN
```

```
                                                B 11
                     0144   276          .SBTTL CONSOLE RECIEVER INTERRUPT DISPATCHER
                     0144   277 ;++
                     0144   278 ; CON$INTINP - CONSOLE INTERRUPT ON INPUT READY
                     0144   279 ;
                     0144   280 ; FUNCTIONAL DESCRIPTION:
                     0144   281 ;
                     0144   282 ; THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
                     0144   283 ; QVSS KEBOARD.
                     0144   284 ;
                     0144   285 ; QVSS TERMINAL:        ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
                     0144   286 ;                       UNSOLICITED AND RESULT IN AN ENTRY INTO THE
                     0144   287 ;                       TERMINAL DRIVER COMMON CHARACTER BUFFERING
                     0144   288 ;                       ROUTINE "@UCB$L_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 CON$INTINP::
                     0144   301
          54   9E   D0  0144   302          MOVL    @(SP)+,R4               ; GET IDB ADDRESS
          50   64   D0  0147   303          MOVL    IDB$L_CSR(R4),R0        ; GET CSR ADDRESS
                     014A   304 ;
                     014A   305 ; GET THE ASSOCIATED UCB
                     014A   306 ;
      55   18 A4   D0  014A   307 5$:       MOVL    IDB$L_UCBLST(R4),R5     ; GET UCB 0 ADDRESS
64 A5    0080 8F   A8  014E   308          BISW    #UCB$M_INTTYPE,UCB$W_STS(R5); SET RECEIVER INTERRUPT
                     0154   309 ;
      50   26 A0   9A  0154   310          MOVZBL  QVCSR_URTBUFA(R0),R0    ; GET INPUT DATA FROM LK201
           FEA5'   30  0158   311          BSBW    QVSS$KEYDECODE          ; DECODE THE KEYBOARD CHARACTER
                     015B   312 ;
                     015B   313 ; CONSOLE TERMINAL INTERRUPT
                     015B   314 ;
          53   50   9A  015B   315 10$:      MOVZBL  R0,R3                   ; ZERO TOP 3 BYTES
               08   13  015E   316          BEQL    30$                     ; DON'T PASS NULLS THRU
          0110 D5   16  0160   317          JSB     @UCB$L_TT_PUTNXT(R5)    ; BUFFER THE CHARACTER
               02   13  0164   318          BEQL    30$                     ; IF EQL THEN NO CHARACTER TO OUTPUT
               0A   10  0166   319 20$:      BSBB    CON$STARTIO             ; OUTPUT THE CHARACTER
          50   8E   7D  0168   320 30$:      MOVQ    (SP)+,R0                ; RESTORE REGISTERS
          52   8E   7D  016B   321          MOVQ    (SP)+,R2                ;
          54   8E   7D  016E   322          MOVQ    (SP)+,R4                ;
                    02  0171   323          REI                             ;
                     0172   324
```

```
                     0172   326          .SBTTL  START I/O ON CONSOLE INTERFACE
                     0172   327 ;++
                     0172   328 ; CON$STARTIO - START I/O ON CONSOLE INTERFACE
                     0172   329 ;
                     0172   330 ; FUNCTIONAL DESCRIPTION:
                     0172   331 ;
                     0172   332 ; THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
                     0172   333 ; IF THE INTERFACE IS READY THE DATA IS OUTPUT DIRECTLY. IF THE INTERFACE
                     0172   334 ; IS NOT READY THEN THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE
                     0172   335 ; NEXT READY INTERRUPT.
                     0172   336 ;
                     0172   337 ; IN EITHER CASE, A RETURN TO THE CALLER IS DONE TO ENTER A 'WAIT FOR
                     0172   338 ; INTERRUPT'' STATE.
                     0172   339 ;
                     0172   340 ; INPUTS:
                     0172   341 ;
                     0172   342 ;       R3 = DATA TO OUTPUT
                     0172   343 ;       R5 = UCB ADDRESS
                     0172   344 ;
                     0172   345 ; OUTPUTS:
                     0172   346 ;
                     0172   347 ;       R3,R4,R5 ARE PRESERVED.
                     0172   348 ;--
                     0172   349
                     0172   350 CON$STARTIO::                                ;
                     0172   351
      08   19         0172   352 10$:    BLSS    20$                         ; BRANCH IF BURST MODE
50    53   9A         0174   353         MOVZBL  R3,R0                       ; GET CHARACTER TO OUTPUT
   0090   30         0177   354         BSBW    CON$PUTCHAR                 ; OUTPUT CHARACTER
      12   11         017A   355         BRB     30$                         ; CHECK FOR MORE TO OUTPUT
                     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    CON$PUTCHAR                 ; OUTPUT CHARACTER
     011C C5   D6    0184   363         INCL    UCB$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
```

```
                     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
```

```
                        .SBTTL  CONSOLE PORT ACTION ROUTINES
                        ;++
                        ; CON$XOFF -    SEND XOFF
                        ; CON$XON -     SEND XON
                        ; CON$STOP -    STOP OUTPUT
                        ; CON$STOP2 -   ALTERNATE STOP
                        ; CON$ABORT -   ABORT CURRENT OUTPUT
                        ; CON$RESUME -  RESUME STOPPED OUTPUT
                        ;
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                        ; CONTROL OUTPUT ON THE PORT
                        ;
                        ; INPUTS:
                        ;
                        ;       R5 = UCB ADDRESS
                        ;
                        ; OUTPUTS:
                        ;
                        ;       R5 = UCB ADDRESS
                        ;--
                        CON$XOFF::
                        CON$XON::
                        CON$STOP::
                        CON$ABORT::
                        CON$RESUME::
05                      RSB
```

```
                     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$SENDCONSCMD::
          50  0F00 8F   A8  01A4    445          BISW    #^XF00,R0               ; SELECT MISCELLANEOUS CONSOLE COMM.
              51   22   DB  01A9    446 10$:     MFPR    #PR$_TXCS,R1            ; GET TRANSMITTER STATUS
            F9 51   07   E1  01AC    447          BBC     #7,R1,10$               ; WAIT FOR CONSOLE READY
              02   50   91  01B0    448          CMPB    R0,#CON$C_BOOTCPU       ; REBOOT CPU?
                   0B   13  01B3    449          BEQL    30$                     ; IF SO BRANCH TO HALT AFTER COMMAND
              23   50   DA  01B5    450          MTPR    R0,#PR$_TXDB            ; OTHERWISE ASSERT COMMAND
              51   22   DB  01B8    451 20$:     MFPR    #PR$_TXCS,R1            ; GET TRANSMITTER STATUS
            F9 51   07   E1  01BB    452          BBC     #7,R1,20$               ; WAIT FOR CONSOLE DONE
                         05  01BF    453          RSB                             ; RETURN
                             01C0    454
              23   50   DA  01C0    455 30$:     MTPR    R0,#PR$_TXDB            ; SEND REBOOT COMMAND TO CONSOLE
                         00  01C3    456          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$OWNCTY::
          50   00000000'GF    D0  01C4    484          MOVL    G^OPA$IDB+IDB$L_CSR,R0          ;GET CSR ADDRESS
                    51   60    B0  01CB    485          MOVW    QVCSR_CTL(R0),R1                ;SAVE INTERRUPT STATE
                         60 0040 8F  AA  01CE    486          BICW    #QVCSR$M_ENA_INT,QVCSR_CTL(R0)  ;DISABLE INTERRUPTS
    50   00000010'GF    01    02  EF  01D3    487          EXTZV   #OP$V_OPACTIVE,#1,G^OPA$CRB+CRB$L_OPFLAGS,R0;SAVE OPACTIVE FLAG
                                05  01DC    488          RSB
```

```
                                                H 11
                                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$RELEASECTY::
00000010'GF   01   02   50   F0 01DD   510          INSV    R0,#OP$V_OPACTIVE,#1,G^OPA$CRB+CRB$L_OPFLAGS; RESTORE OPACTIVE FLAG
          50    00000000'GF   D0 01E6   511          MOVL    G^OPA$IDB+IDB$L_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 ; CON$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
                                01F1    519 ;
                                01F1    520 ; FUNCTIONAL DESCRIPTION:
                                01F1    521 ;
                                01F1    522 ;       THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
                                01F1    523 ;       DIRECTLY TO THE CONSOLE TERMINAL
                                01F1    524 ;
                                01F1    525 ; INPUTS:
                                01F1    526 ;       None
                                01F1    527 ;
                                01F1    528 ; OUTPUTS:
                                01F1    529 ;       R0 contains the character.
                                01F1    530 ;
                                01F1    531 ;--
                     00000013   01F1    532          control_s = 19                  ; control s (xoff)
                     00000011   01F1    533          control_q = 17                  ; control q (xon)
                     00000001   01F1    534          qvuart$m_rxrdy = 1              ; receiver ready bit
                                01F1    535
                                01F1    536 CON$GETCHAR::
     50    00000000'GF    D0    01F1    537 5$:      movl    g^opa$idb+idb$l_csr,r0  ;get qvss csr address
           22 A0    01    B3    01F8    538 10$:     bitw    #qvuart$m_rxrdy,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::
                     3E   BB    020A   565          pushr   #^m<r1,r2,r3,r4,r5>     ;save registers
           53   00000000'GF   DE    020C   566          moval   g^opa$crb,r3            ;get crb address
        0F 10 A3    02   E0    0213   567          bbs     #op$v_opactive,crb$l_opflags(r3),1$;continue if we have control of t
  03 00000000'GF    00000000'8F   E0    0218   568          bbs     #exe$v_opa0,g^exe$gl_wsflags,1$;output to opa0 enabled? bs yes
                  0030   31    0224   569          brw     80$                     ;return
                    50   DD    0227   570 1$:      pushl   r0                      ;save r0
        10 10 A3    00   E5    0229   571          bbcc    #op$v_reinit,crb$l_opflags(r3),2$;reinit the scan lines?
1C B3   7800 8F    00   6C    022E   572          movc5   #0,(sp),#0,#24*128*10,@crb$l_video_base(r3); init memory
           53   00000000'GF   DE    0237   573          moval   g^opa$crb,r3            ;get crb address
        03 10 A3    01   E5    023E   574 2$:      bbcc    #op$v_remap,crb$l_opflags(r3),3$;remap scan lines to screen?
                  00\4   30    0243   575          bsbw    remap                   ;remap scan lines to the screen
                    5( 8ED0   0246   576 3$:      popl    r0                      ;restore r0
                 53   1C A3    D0    0249   577          movl    crb$l_video_base(r3),r3 ;get va of video memory
           54   00000000'GF   D0    024D   578          movl    g^opa$idb+idb$l_csr,r4  ;get va of qvss csr
                FDA9'   30    0254   579          bsbw    qvss$putchar            ;output the character
                  3E    BA    0257   580 80$:      popr    #^m<r1,r2,r3,r4,r5>     ;restore registers
                        \5    0259   581          rsb                             ;
                              025A   582          .dsabl  lsb
```

```
                                025A   584          .SBTTL  - INITIALIZE CONSOLE TERMINAL FOR NON-INTERRUPT DRIVEN I/O
                                025A   585 ;+
                                025A   586 ;  )N$INIT_CTY - INITIALIZE QVSS FOR NON-INTERRUPT DRIVEN I/O
                                025A   587 ;
                                025A   588 ; F NCTIONAL DESCRIPTION:
                                025A   589 ;
                                025A   590 ;       THIS ROUTINE MUST BE CALLED FROM INIT BEFORE ANY CONSOLE TERMINAL I/O
                                025A   591 ;       CAN OCCUR.
                                025A   592 ;
                                025A   593 ; IN JTS:
                                025A   594 ;
                                025A   595 ; OU' 'UTS:
                                025A   596 ;       VIDEO MEMORY MAPPED.
                                025A   597 ;       I/O SPACE THAT CONATINS THE CSRS FOR QVSS MAPPED.
                                025A   598 ;
                                025A   599 ;--
                                025A   600 CON$I( IT_CTY::                          ; INITIALIZE QVSS CONTROLLER
                                025A   601
                                025A   602          ;
                                025A   603          ; SAVE REGISTERS
                                025A   604          ;
                      1E    BB  025A   605          PUSHR   #^M<R1,R2,R3,R4>
                                025C   606
                                025C   607          ;
                                025C   608          ; INITIALIZE THE KEYBOARD TRANSLATION TABLE
                                025C   609          ;
    00000000'EF   FFFFFF90'EF   DE  025C   610          MOVAL   QVSS$KEY-112,QVSS$KEYTABLE
                                0267   611
                                0267   612          ;
                                0267   613          ; Initialize OPDRVWS1 state flags
                                0267   614          ;
          53    00000000'GF   DE  0267   615          MOVAL   G^OPA$CRB,R3                ; GET THE CRB ADDRESS
                 10 A3    04  D0  026E   616          MOVL    #OP$M_OPACTIVE,CRB$L_OPFLAGS(R3); OPACTIVE,NOREINIT.NOREMAP,NOVCACTI
                                0272   617
                                0272   618          ;
                                0272   619          ; GET VIRTUAL ADDRESS OF CSR IN R4
                                0272   620          ;
          52    00000000'GF   D0  0272   621          MOVL    G^BOO$GL_SPTFREL,R2         ; GET A FREE SPT
                00000000'GF   D6  0279   622          INCL    G^BOO$GL_SPTFREL            ;
          51    00000000'GF   D0  027F   623          MOVL    G^MMG$GL_SPTBASE,R1         ; GET VA OF SYSTEM SPT BASE
                   51    6142   DE  0286   624          MOVAL   (R1)[R2],R1                 ; GET VA OF SPT PTE
             52    52      09   78  028A   625          ASHL    #9,R2,R2                    ; MAKE VA
          52    80000000   8F   C8  028E   626          BISL    #VA$M_SYSTEM,R2             ; SET SYSTEM SPACE BIT
    54    52    00000080   8F   C1  0295   627          ADDL3   #QVCSR_BOFF,R2,R4           ; CALC CSR VIRTUAL ADDRESS
          00000000'GF      54   D0  029D   628          MOVL    R4,G^OPA$IDB+IDB$L_CSR      ; SAVE IN IDB
          61    9010000F   8F   D0  02A4   629          MOVL    #<PTE$M_VALID!PTE$C_KW!QVCSR_PFN>,(R1); MAP PA OF CSR TO SYS VA
                                02AB   630
                                02AB   631          ;
                                02AB   632          ; Map Video Memory
                                02AB   633          ;
                      51    D4  02AB   634          CLRL    R1                          ;
          51    64    87FF   8F   AB  02AD   635          BICW3   #^C<QVCSR$M_MEMBANK>,(R4),R1; GET BASE QVSS MEMORY BANK
            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
```

```
                OB 50    E9  02CB   641          BLBC    R0,100$                         ; NO SPTS THEN EXIT
          1C A3    52    D0  02CE   642          MOVL    R2,CRB$L_VIDEO_BASE(R3) ; SAVE STARTING VA OF BITMAP
                   08    10  02D2   643          BSBB    REMAP                           ; MAP THE SCREEN
     50    0000'8F    3C  02D4   644          MOVZWL  #SS$_NORMAL,R0                  ; INDICATE SUCCESS
                1E    BA  02D9   645 100$:    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$L_VIDEO_BASE(R3),R2; COMPUTE ADDRESS OF SCAN MAP
                                    02E5   674
                      14 A3     D5  02E5   675          TSTL    CRB$L_SCAN_MAP(R3)      ; SCAN MAP SAVE AREA?
                         0B     13  02E8   676          BEQL    5$                      ; IF EQL NO
                         3C     BB  02EA   677          PUSHR   #^M<R2,R3,R4,R5>        ; SAVE REGISTERS
        14 B3    62    01E0 8F   28  02EC   678          MOVC3   #24*10*2,(R2),@CRB$L_SCAN_MAP(R3); SAVE SCAN MAP
                         3C     BA  02F3   679          POPR    #^M<R2,R3,R4,R5>        ; RESTORE REGISTERS
                                02F5   680
                         50     D4  02F5   681 5$:      CLRL    R0
    51   000000F0 8F           D0  02F7   682          MOVL    #24*10,R1               ; NUMBER OF SCAN MAPS
                82       50     B0  02FE   683 10$:     MOVW    R0,(R2)+
                         50     D6  0301   684          INCL    R0
                      F8 51     F5  0303   685          SOBGTR  R1,10$
                                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$ALLOSPT called - so SPTs are allocated
                     0307   710 ;
                     0307   711 ;--
                     0307   712 MAP_PAGES:
             3A   BB 0307   713          PUSHR   #^M<R1,R3,R4,R5>        ; save work registers
                     0309   714                                          ;  r1 = input used as loop counter
                     0309   715                                          ;  r3 = address of SPT
                     0309   716                                          ;  r4 = index into PFN database
                     0309   717                                          ;  r5 = temp storage
         55   50  D0 0309   718          MOVL    R0,R5                   ; Save starting PFN
00000000'GF   16  030C 719          JSB     G^IOC$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$ALLOSPT returns:
                     0315   723 ;       R1 = preserved, R2 = SVPN (index into SPT), R3 = address of SPT
                     0315   724 ;
                     0315   725 ; The main loop indexes backwards through the system page table entries
                     0315   726 ; and backwards through the PFN database.  It goes backwards so that the
                     0315   727 ; last system virtual address calculated can be returned to the caller.
                     0315   728 ;
                     0315   729 ;                                          ;  r0 = index into SPT
      50   52   51  C1 0315   730          ADDL3   R1,R2,R0                ; start at last SPT and go backwards
      54   51   55  C1 0319   731          ADDL3   R5,R1,R4                ; start PFNs at end in loop
                     031D   732 10$:                                      ; set up system page-table entry
                50  D7 031D   733          DECL    R0                      ; back up SVPN index
                54  D7 031F   734          DECL    R4                      ; back up PFN index
           6340   54  D0 0321   735          MOVL    R4,(R3)[R0]             ; fill PFN in SPT
6340  A0000000 8F  C8 0325   736          BISL2   #<PTE$C_UW!PTE$M_VALID>,(R3)[R0] ; user mode access, valid
                     032D   737 ;
                     032D   738 ; Invalidate system virtual address
                     032D   739 ;
      52   50   09  78 032D   740          ASHL    #9,R0,R2                ; turn SVPN into VA
  52   80000000 8F  C8 0331   741          BISL2   #<1@31>,R2              ; make VA a system VA
                     0338   742          INVALID R2                      ; and clear translation buffer
              DF 51  F5 033B   743          SOBGTR  R1,10$                  ; loop N times
                     033E   744
      50   0000'8F  3C 033E   745          MOVZWL  #SS$_NORMAL,R0          ; indicates success (R2 has system VA)
```

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

```
ADP$L_VECTOR                    = 00000010
BIT...                          = 0000000F
BOO$GL_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
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$W_VECTOR                    = 0000001E
EXE$ALONONPAGED                   ********    X   02
EXE$GL_WSFLAGS                    ********    X   02
EXE$V_OPA0                        ********    X   02
IDB$L_CSR                       = 00000000
IDB$L_UCBLST                    = 00000018
IOC$ALOSPT                        ********    X   02
IOC$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$V_SHIFT                     = 00000003
MAP_PAGES                         00000307 R      02
MMG$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
OPA$CRB                           ********    X   02
OPA$IDB                           ********    X   02
OPA$VECTOR                        ********    X   02
PR$_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_BOFF                      = 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
```

```
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$M_RXRDY                  = 00000001
REMAP                             000002DC  R       02
SCAN_MAP                        = 0003F800  G
SIZ...                          = 00000004
SS$_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_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	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
_255DUA28:[SYS.OBJ]LIB.MLB;1	27
_255DUA28:[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)+EXECML$/LIB
```

MOUNTVER
LIS

OPDRVWS1
LIS

QUORUM
LIS

OPDRV790
LIS

OPDRIVER
CIS

REBLDLOCK
LIS