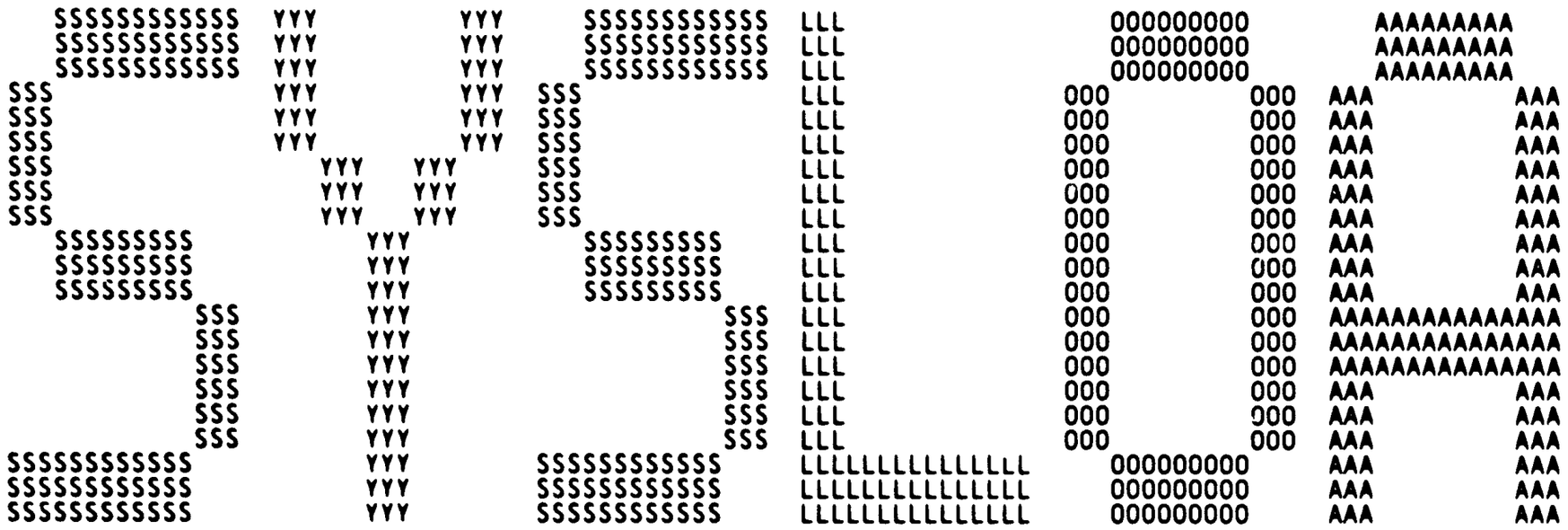
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSSSSSSSSSS YYY YYY SSSSSSSSSSS LLL 000000000 AAAAAAAAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 AAA AAA
SSS YYY YYY SSS LLL 000 000 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**OPDRV790

```
  000000    PPPPPPPP    DDDDDDDD    RRRRRRRR    VV      VV  77777777      999999      000000
  000000    PPPPPPPP    DDDDDDDD    RRRRRRRR    VV      VV  77777777      999999      000000
00      00  PP      PP  DD      DD  RR      RR  VV      VV        77  99      99  00      00
00      00  PP      PP  DD      DD  RR      RR  VV      VV        77  99      99  00      00
00      00  PP      PP  DD      DD  RR      RR  VV      VV        77  99      99  00    0000
00      00  PP      PP  DD      DD  RR      RR  VV      VV        77  99      99  00    0000
00      00  PPPPPPPP    DD      DD  RRRRRRRR    VV      VV      77    99999999  00  00  00
00      00  PPPPPPPP    DD      DD  RRRRRRRR    VV      VV      77    99999999  00  00  00
00      00  PP          DD      DD  RR  RR      VV      VV    77            99  0000    00
00      00  PP          DD      DD  RR  RR      VV      VV    77            99  0000    00
00      00  PP          DD      DD  RR    RR      VV  VV    77              99  00      00
00      00  PP          DD      DD  RR    RR      VV  VV    77              99  00      00
  000000    PP          DDDDDDDD    RR      RR      VV      77        999999      000000
  000000    PP          DDDDDDDD    RR      RR      VV      77        999999      000000

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  OPDRV790 - VAX/VMS 11/790 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 ;
0000    35 ; AUTHOR: Benn Schreiber, Trudy Matthews
0000    36 ;
0000    37 ; MODIFIED BY:
0000    38 ;
0000    39 ;       V03-011 TCM0009         Trudy C. Matthews       20-Aug-1984
0000    40 ;               Fix input interrupt dispatching through the dispatch table.
0000    41 ;
0000    42 ;       V03-010 TCM0008         Trudy C. Matthews       31-Jul-1984
0000    43 ;               Move routine CON$KEEPALIVE to module MCHECK790.
0000    44 ;
0000    45 ;       V03-009 TCM0007         Trudy C. Matthews       24-Jul-1984
0000    46 ;               Fix some undefined symbols in TCM0006.
0000    47 ;
0000    48 ;       V03-008 TCM0006         Trudy C. Matthews       19-Jul-1984
0000    49 ;               Add routine CON$KEEPALIVE, which is called periodically to
0000    50 ;               determine if the VENUS console software is still functioning.
0000    51 ;               Update the CON$C_xxx console function code definitions.
0000    52 ;               Add TXDB$C_xxx and RXDB$C_xxx definitions for EMM and logical
0000    53 ;               console lines.
0000    54 ;
0000    55 ;       V03-007 WHM0001         Bill Matthews           09-Jul-1984
0000    56 ;               Add routines CON$PUTCHAR, CON$GETCHAR, and CON$INIT_CTY to do
0000    57 ;               non-interrupt driven I/O to the console terminal.
```

```
0000     58 ;
0000     59 ;       V03-006 TCM0005         Trudy C. Matthews       20-Jun-1984
0000     60 ;               Add a first pass at handling venus environmental monitor
0000     61 ;               alerts.
0000     62 ;
0000     63 ;       V03-005 TCM0004         Trudy C. Matthews       04-Jun-1984
0000     64 ;               Track changes in the console interface protocol: we can not
0000     65 ;               read or write TXCS unless the "READY" bit is set; also, the
0000     66 ;               "write enable" bit position has changed.  Add routine
0000     67 ;               CON$RELEASECTY.
0000     68 ;
0000     69 ;       V03-004 TCM0003         Trudy C. Matthews       09-Apr-1984
0000     70 ;               Add dummy entry point for CON$RELEASECTY.  The full routine
0000     71 ;               will be added later.
0000     72 ;
0000     73 ;       V03-003 TCM0002         Trudy C. Matthews       13-Dec-1983
0000     74 ;               Add support for the "write enable bit" for line enable mask
0000     75 ;               in VENUS' TXCS in routines CON$SENDCONSCMD and CON$OWNCTY.
0000     76 ;               Modify interface to CON$OWNCTY: it now returns the values
0000     77 ;               that should be restored to TXCS and RXCS when the caller
0000     78 ;               is done with "exclusive" use of the console terminal.
0000     79 ;
0000     80 ;       V03-002 MIR0084         Michael I. Rosenblum    03-Aug-1983
0000     81 ;               Change references to TT_CARRIER and TT_DTR to
0000     82 ;               DZ.  Remove STOP2 entry point and all references to
0000     83 ;               STOP2.
0000     84 ;
0000     85 ;       V03-001 TCM0001         Trudy C. Matthews       25-Mar-1983
0000     86 ;               Modify CON$OWNCTY to work with 11/790 console interface.
0000     87 ;
0000     88 ;-
```

```
                     ;
                     ; SYMBOL DEFINITIONS
                     ;

                             $CONDEF                         ;CONSOLE DEFINITIONS
                             $CRBDEF                         ;DEFINE CRB
                             $DCDEF                          ;DEFINE DEVICE CLASSES
                             $DDBDEF                         ;DEFINE DDB
                             $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS
                             $DPTDEF                         ;DRIVER PROLOGUE TABLE
                             $DYNDEF                         ;STRUCTURE TYPE CODE DEFINITIONS
                             $EMBDEF <HD>                    ;DEFINE ERRORLOG ENTRY FORMATS
                             $IDBDEF                         ;DEFINE IDB
                             $IPLDEF                         ;DEFINE IPL LEVELS
                             $IRPDEF                         ;DEFINE IRP OFFSETS
                             $PRDEF                          ;DEFINE PROCESSOR REGISTERS
                             $PR790DEF                       ;DEFINE 11/790 PROCESSOR REGISTERS
                             $TTDEF                          ;DEFINE TERMINAL CHARACTERISTICS
                             $UCBDEF                         ;DEFINE UCB
                             $TTYDEFS                        ;TTY UCB extension(must FOLLOW $UCBDEF)
                             $TTYMODEM                       ;TTY modem definitions
                             $TTYMACS                        ;TTY macro definitions
                             $VECDEF                         ;DEFINE CRB VECTOR
                             $WCBDEF                         ;Define WCB
                     ;
                     ; TXCS REGISTER DEFINITIONS
                     ;
                             $DEFINI TXCS

                     $DEF    TXS             .BLKL   1       ;TXCS REGISTER

                             _VIELD  TXS,0,<-                ;DEFINE FIELDS TXS_M_XXX
                                     <,6>,-                  ;MBZ BITS
                                     <IE,,M>,-               ;INTERRUPT ENABLE BIT
                                     <RDY,,M>,-              ;READY BIT
                                     <ID,4,M>,-              ;ID FIELD
                                     <,3>,-                  ;MBZ BITS
                                     <WRTENA,,M>,-           ;ENABLE WRITE TO TXCS MASK FIELD
                                     <TEM,8,M>,-             ;TRANSMITTER ENABLE MASK
                                     >
00000000             .=TXS
                             _VIELD  TXS,0,<-                ;TRANSMITTER ENABLE MASK BITS
                                     <,16>,-                 ;SKIP TO TRANSMIT ENABLE MASK FIELD
                                     <CONSTERM,,M>,-         ;CONSOLE TERMINAL
                                     <REMPORT,,M>,-          ;REMOTE SERVICES PORT
                                     <EMM,,M>,-              ;ENVIRONMENTAL MONITOR
                                     <LOGCONS,,M>,-          ;LOGICAL CONSOLE
                                     >
                             $DEFEND TXCS

                     ;
                     ; DEFINE CONSTANTS FOR TXDB DATA:
                     ;
                     ; THE FOLLOWING VALUES CAN BE SENT ON THE EMM LINE.
                     ;
00000000             TXDB$C_EMM_STATUS = 0                   ;REQUEST FOR EMM STATUS
00000001             TXDB$C_EMM_ENV = 1                      ;REQUEST FOR SYSTEM ENVIRONMENT INFO
```

```
00000010  0000   147 TXDB$C_EMM_MARGIN = 16                  ;COMMAND TO SET MARGIN REGULATORS
00000011  0000   148 TXDB$C_EMM_CANCEL = 17                  ;CANCEL EMM REQUESTS ON INTERFACE
          0000   149 ;
          0000   150 ; THE FOLLOWING VALUES CAN BE SENT ON THE LOGICAL CONSOLE LINE.
          0000   151 ;
00000010  0000   152 TXDB$C_LOG_EXWARM = ^X10                ; EXAMINE WARMSTART FLAG
00000011  0000   153 TXDB$C_LOG_EXCOLD = ^X11                ; EXAMINE COLDSTART FLAG
00000012  0000   154 TXDB$C_LOG_EXUCODE = ^X12               ; EXAMINE MICROCODE VERSION LEVEL
00000020  0000   155 TXDB$C_LOG_SNDDIAG = ^X20               ; SEND DIAGNOSTIC COMMAND STRING
00000030  0000   156 TXDB$C_LOG_REQERL = ^X30                ; REQUEST ERRORLOG DATA
00000031  0000   157 TXDB$C_LOG_INVSNP1 = ^X31               ; INVALIDATE ERRORLOG SNAPSHOT 1
00000032  0000   158 TXDB$C_LOG_INVSNP2 = ^X32               ; INVALIDATE ERRORLOG SNAPSHOT 2
00000070  0000   159 TXDB$C_LOG_CANCEL = ^X70                ; CANCEL CURRENT AND QUEUED REQUESTS
          0000   160
          0000   161 ;
          0000   162 ; RXCS REGISTER DEFINITIONS
          0000   163 ;
          0000   164         $DEFINI RXCS
          0000   165
          0000   166 $DEF     RXS     .BLKL   1               ;RXCS REGISTER
          0004   167         _VIELD  RXS,0,<-
          0004   168                 <,6>,-                  ;MBZ BITS
          0004   169                 <IE,,M>,-               ;INTERRUPT ENABLE BIT
          0004   170                 <DONE,,M>,-             ;DONE BIT
          0004   171                 <,8>,-                  ;MBZ BITS
          0004   172                 <DTR,8,M>,-             ;LOGICAL DTR BITS
          0004   173                 >
00000000  0004   174         .=RXS
          0000   175         _VIELD  RXS,0,<-                ;LOGICAL DTR BITS
          0000   176                 <,16>,-                 ;SKIP TO THE LOGICAL DTR FIELD
          0000   177                 <CONSTERM,,M>,-         ;CONSOLE TERMINAL
          0000   178                 <REMPORT,,M>,-          ;REMOTE SERVICES PORT
          0000   179                 <EMM,,M>,-              ;ENVIRONMENTAL MONITOR
          0000   180                 <LOGCONS,,M>,-          ;LOGICAL CONSOLE
          0000   181                 >
          0000   182         $DEFEND RXCS
          0000   183 ;
          0000   184 ; CONSOLE RXDB REGISTER
          0000   185 ;
          0000   186         $DEFINI RXDB
          0000   187
          0000   188 $DEF     RXD             .BLKL   1       ;RXDB RECEIVER DATA REGISTER
          0004   189         _VIELD  RXD,0,<-
          0004   190                 <DATA,8>,-              ;8 BITS OF DATA
          0004   191                 <ID,4>,-                ;4 BITS OF ID
          0004   192                 <,4>,-                  ;SKIP 4 BITS OF MBZ
          0004   193                 <CARRIER,8>,-           ;LOGICAL CARRIER
          0004   194                 >
          0004   195         $DEFEND RXDB
          0000   196 ;
          0000   197 ; DEFINE DATA VALUES THAT CAN BE RECIEVED ON THE EMM AND LOGICAL CONSOLE
          0000   198 ; LINES.
          0000   199 ;
          0000   200 ; THE FOLLOWING RXDB VALUES MAY BE RECEIVED ON THE LOGICAL CONSOLE LINE:
          0000   201 ;
00000010  0000   202 RXDB$C_LOG_WRMFLG = 16                  ;RETURNING VALUE FOR WARMSTART FLAG
00000011  0000   203 RXDB$C_LOG_CLDFLG = 17                  ;RETURNING VALUE FOR COLDSTART FLAG
```

```
00000012  0000    204 RXDB$C_LOG_UCODE = 18                   ;RETURNING MICROCODE VERSION
00000020  0000    205 RXDB$C_LOG_CMDCMP = 32                  ;CONSOLE COMMAND STRING COMPLETE
00000082  0000    206 RXDB$C_LOG_CMDERR = 130                 ;CONSOLE COMMAND STRING ERROR
00000030  0000    207 RXDB$C_LOG_SNAP = 48                    ;SNAPFILE STATUS RETURNED
00000040  0000    208 RXDB$C_LOG_REBOOT = 64                  ;CONSOLE REBOOT SUCCESSFUL
```

```
                             0000   210          .SBTTL    CONSOLE RECEIVE DISPATCH VECTOR
                             0000   211
                         00000000   212          .PSECT  $YSLOA,LONG
                             0000   213
                             0000   214 ;
                             0000   215 ; THE INTERRUPT ROUTINE ADDRESSES FOR EACH OF THE DEVICES THAT SHARE THE
                             0000   216 ; RXCS/RXDB REGISTER ARE LOADED INTO THIS TABLE BY THE UNIT INITIALIZATION
                             0000   217 ; ROUTINE FOR EACH DEVICE.  CON$INITIAL SETS THE CELLS FOR THE CONSOLE
                             0000   218 ; TERMINAL, THE ENVIRONMENTAL MONITOR, THE UNUSED VECTORS, AND THE CARRIER
                             0000   219 ; CHANGE INTERRUPT TYPE.
                             0000   220 ;
                             0000   221 CON$INPDISTAB:
                 00000000    0000   222          .LONG   0                       ;ID 0 - CONSOLE TERMINAL
                 00000000    0004   223          .LONG   0                       ;ID 1 - REMOTE SERVICE PORT
                 00000000    0008   224          .LONG   0                       ;ID 2 - ENVIRONMENTAL MONITOR
                 00000000    000C   225          .LONG   0                       ;ID 3 - LOGICAL CONSOLE
00000000'00000000'00000000'00000000' 0010   226          .LONG   0[11]                   ;IDS 4-14 - UNUSED, RESERVED TO DIGITAL
00000000'00000000'00000000'00000000' 0020
        00000000'00000000'00000000' 0030
                 00000000    003C   227          .LONG   0                       ;ID 15 - CARRIER CHANGE NOTIFICATION
```

```
                        0040   229          .SBTTL  CONSOLE CONTROLLER INITIALIZATION
                        0040   230 ;++
                        0040   231 ; CON$INITIAL - INITIALIZE CONSOLE CONTROLLER
                        0040   232 ;
                        0040   233 ; FUNCTIONAL DESCRIPTION:
                        0040   234 ;
                        0040   235 ; THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE CONTROLLER.
                        0040   236 ;
                        0040   237 ; INPUTS:
                        0040   238 ;
                        0040   239 ;       R5 = IDB ADDRESS
                        0040   240 ;       R6 = DDB ADDRESS
                        0040   241 ;       R8 = CRB ADDRESS
                        0040   242 ;
                        0040   243 ; OUTPUTS:
                        0040   244 ;
                        0040   245 ;       ALL REGISTERS ARE PRESERVED, EXCEPT R0, R1.
                        0040   246 ;--
                        0040   247 CON$INITIAL::                                ;INITIALIZE CONSOLE INTERFACE
          10 A8   D5    0040   248          TSTL    CRB$L_AUXSTRUC(R8)         ;HAVE WE PASSED THIS WAY BEFORE?
             50   12    0043   249          BNEQ    30$                        ;BRANCH IF SO
                        0045   250
     50   FFB7 CF   9E  0045   251          MOVAB   W^CON$INPDISTAB,R0         ;POINT TO THE CONSOLE DISPATCH TABLE
       10 A8   50   D0  004A   252          MOVL    R0,CRB$L_AUXSTRUC(R8)      ;SET POINTER IN CRB
     60   01AA'CF   9E  004E   253          MOVAB   W^INT_CONSOLINP,(R0)       ;STORE CONSOLE INTERRUPT ROUTINE
          80   80   D0  0053   254          MOVL    (R0)+,(R0)+                ;SAME ROUTINE FOR REMOTE SERVICES PORT
     80   0281'CF   9E  0056   255          MOVAB   W^INT_EMMINP,(R0)+         ;STORE EMM INTERRUPT RTN
     80   030C'CF   9E  005B   256          MOVAB   W^INT_LOGINP,(R0)+         ;STORE LOGICAL CONSOLE INTERRUPT RTN
          51   0B   D0  0060   257          MOVL    #11,R1                     ;SET NUMBER OF VECTORS TO INITIALIZE
     80   0402'CF   9E  0063   258 10$:     MOVAB   W^INT_IGNOREINP,(R0)+      ;STORE INGORE INTERRUPT ROUTINE
            F8 51   F5  0068   259          SOBGTR  R1,10$                     ;DO ALL
     60   037B'CF   9E  006B   260          MOVAB   W^INT_CARCHANGE,(R0)       ;STORE CARRIER CHANGE INTERRUPT ADDR
                        0070   261
          50   21   DB  0070   262          MFPR    #PR$_RXDB,R0               ;READ CURRENT RECEIVER STATUS
50   50     F0 8F   78  0073   263          ASHL    #<-RXD_V_CARRIER>,R0,R0    ;POSITION CARRIER BYTE TO LOW BYTE
       1D A8   50   90  0078   264          MOVB    R0,CRB$B_DZ_CARRIER(R8)    ;SAVE INITIAL STATUS
            0E A5   94  007C   265          CLRB    IDB$B_TT_ENABLE(R5)        ;CLEAR TRANSMIT ENABLE MASK
               0C   90  007F   266          MOVB    #<RXS_M_EMM!RXS_M_LOGCONS>@<-RXS_V_DTR>,-
               1E A8        0081   267                  CRB$B_DZ_DTR(R8)       ;SETUP INITIAL LOGICAL DTR
          50   1E A8   9A  0083   268          MOVZBL  CRB$B_DZ_DTR(R8),R0        ;GET LOGICAL DTR
     50   50     10 78   87  0087   269          ASHL    #RXS_V_DTR,R0,R0           ;POSITION IN REGISTER
50   00000040 8F   C8  008B   270          BISL2   #RXS_M_IE,R0               ;SET INTERRUPT ENABLE
          20   50   DA  0092   271          MTPR    R0,#PR$_RXCS               ;ENABLE THE LINES
                        0095   272 30$:
                        0095   273 CON_RETURN:
                     05 0095   274          RSB
```

```
                                0096   276          .SBTTL  CONSOLE UNIT INITIALIZATION
                                0096   277 ;++
                                0096   278 ; CON$INITLINE - INITIALIZE CONSOLE UNIT
                                0096   279 ;
                                0096   280 ; FUNCTIONAL DESCRIPTION:
                                0096   281 ;
                                0096   282 ; THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.
                                0096   283 ;
                                0096   284 ; INPUTS:
                                0096   285 ;
                                0096   286 ;       R5 = UCB ADDRESS
                                0096   287 ;       R9 = CRB ADDRESS
                                0096   288 ;
                                0096   289 ; OUTPUTS:
                                0096   290 ;
                                0096   291 ;       REGISTERS R4,R5 PRESERVED
                                0096   292 ;--
                                0096   293 CON$INITLINE::
   00 44 A5    15    E2         0096   294          BBSS    #TT$V_MODEM,UCB$L_DEVDEPEND(R5),2$ ;ENSURE DEVICE IS MODEM
 00 00C4 C5    15    E2         009B   295 2$:      BBSS    #TT$V_MODEM,UCB$L_TT_DECHAR(R5),4$
51    00000000'GF    D0         00A1   296 4$:      MOVL    G^TTY$GL_DPT,R1         ;ADDRESS OF CLASS DPT
      50    1E A1    3C         00A8   297          MOVZWL  DPT$W_VECTOR(R1),R0     ;LOCATE CLASS DRIVER VECTOR TABLE
         51    50    C0         00AC   298          ADDL    R0,R1                   ;RELOCATE BASE ADDRESS
    0114 C5    51    D0         00AF   299          MOVL    R1,UCB$L_TT_CLASS(R5)   ;SET TERMINAL CLASS DRIVER VECTOR
    010C C5    61    D0         00B4   300          MOVL    CLASS_GETNXT(R1),UCB$L_TT_GETNXT(R5)
 0110 C5    04 A1    D0         00B9   301          MOVL    CLASS_PUTNXT(R1),UCB$L_TT_PUTNXT(R5)
      50    28 A5    D0         00BF   302          MOVL    UCB$L_DDB(R5),R0        ;GET DDB ADDRESS
   0C A0    10 A1    D0         00C3   303          MOVL    CLASS_DDT(R1),DDB$L_DDT(R0)
 0088 C5    10 A1    D0         00C8   304          MOVL    CLASS_DDT(R1),UCB$L_DDT(R5) ;SET DDT ADDRESS IN UCB
                                00CE   305
50    00000000'GF    9E         00CE   306          MOVAB   G^OP$DPT,R0             ;GET ADDRESS OF CONSOLE DPT
      51    1E A0    3C         00D5   307          MOVZWL  DPT$W_VECTOR(R0),R1     ;OFFSET TO CONSOLE PORT VECTOR DISPATCH TABL
0118 C5    50    51    C1         00D9   308          ADDL3   R1,R0,UCB$L_TT_PORT(R5) ;SET ADDRESS IN UCB
                                00DF   309
      64 A5    10    A8         00DF   310          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET ONLINE
50    01    54 A5    78         00E3   311          ASHL    UCB$W_UNIT(R5),#1,R0    ;BUILD UNIT'S BIT MASK
   0106 C5    50    B0         00E8   312          MOVW    R0,UCB$W_TT_UNITBIT(R5) ;SAVE IT IN UCB
         5C A5    B5         00ED   313          TSTW    UCB$W_REFC(R5)          ;REFCOUNT 0?
            08    12         00F0   314          BNEQ    20$                     ;IF NEQ THEN NO SET UP
                                00F2   315
   50   0114 C5    D0         00F2   316          MOVL    UCB$L_TT_CLASS(R5),R0   ;ADDRESS OF CLASS VECTOR TABLE
         08 B0    16         00F7   317          JSB     @CLASS_SETUP_UCB(R0)    ;INITIALIZE THE UCB FOR CONSOLE TERM.
                                00FA   318
         0124 C5    94         00FA   319 20$:     CLRB    UCB$B_TT_DS_RCV(R5)     ;CLEAR RECEIVE STATUS
         50    21    DB         00FE   320          MFPR    #PR$_RXDB,R0            ;READ CURRENT STATUS
50    50    F0 8F    78         0101   321          ASHL    #<-RXS_V_DTR>,R0,R0     ;POSITION CARRIER BYTE
   0106 C5    50    93         0106   322          BITB    R0,UCB$W_TT_UNITBIT(R5) ;CARRIER UP FOR THIS LINE?
            06    13         010B   323          BEQL    25$                     ;BRANCH IF NOT
         F0 8F    90         010D   324          MOVB    #<TT$M_DS_CARRIER!TT$M_DS_DSR!TT$M_DS_CTS!TT$M_DS_RING>,-
         0124 C5               0110   325                      UCB$B_TT_DS_RCV(R5) ;YES, SET IT AND BITS ALWAYS SET
         51    00    9A         0113   326 25$:     MOVZBL  #MODEM$C_INIT,R1        ;SET TO INIT MODEM PROTOCOL
         5C A5    B5         0116   327          TSTW    UCB$W_REFC(R5)          ;ANY CHANNELS ATTACHED?
            03    13         0119   328          BEQL    30$                     ;IF EQL NO
         51    01    9A         011B   329          MOVZBL  #MODEM$C_SHUTDWN,R1     ;FORCE MODEM HANGUP
   50   0114 C5    D0         011E   330 30$:     MOVL    UCB$L_TT_CLASS(R5),R0   ;ADDRESS CLASS VECTOR TABLE
            52    DD         0123   331          PUSHL   R2                      ;SAVE R2 OVER DS_TRAN
         0C B0    16         0125   332          JSB     @CLASS_DS_TRAN(R0)      ;INVOKE DATA SET TRANSITION
```

```
                52 8ED0  0128   333          POPL    R2                          ;RESTORE R2
      08 64 A5    05   E1  012B   334 40$:    BBC     #UCB$V_POWER,UCB$W_STS(R5),50$ ;DID WE DETECT A POWERFAIL?
     50   0114 C5    D0  0130   335          MOVL    UCB$L_TT_CLASS(R5),R0       ;YES, GET CLASS VECTOR TABLE ADDRESS
           20 B0    17  0135   336          JMP     @CLASS_POWERFAIL(R0)        ;AND GO TO THE POWERFAIL CODE
                        0138   337 50$:
                        0138   338 CON$SET_LINE::
                        0138   339 CON$SET_MODEM::
                        0138   340 CON$NULL::
                        0138   341 CON$INIT_CTY::
                    05  0138   342          RSB
```

```
                     0139   344          .SBTTL  CON$DISCONNECT DISCONNECT LINE
                     0139   345 ;++
                     0139   346 ; CON$DISCONNECT - SHUT OFF UNIT
                     0139   347 ;
                     0139   348 ; FUNCTIONAL DESCRIPTION:
                     0139   349 ;
                     0139   350 ;       THIS ROUTINE IS USED WHEN FOR SOME REASON THE UNIT MUST BE DISCONNECTED
                     0139   351 ;
                     0139   352 ; INPUTS:
                     0139   353 ;
                     0139   354 ;       R5 = UCB ADDRESS
                     0139   355 ;
                     0139   356 ; OUTPUTS:
                     0139   357 ;
                     0139   358 ;       R3,R4 ARE USED
                     0139   359 ;--
                     0139   360
                     0139   361 CON$DISCONNECT::
          07   BB   0139   362          PUSHR   #^M<R0,R1,R2>
     51   01   DO   013B   363          MOVL    #MODEM$C_SHUTDWN,R1     ;SET MODEM SHUTDOWN
50   0114 C5   D0   013E   364          MOVL    UCB$L_TT_CLASS(R5),R0   ;ACCESS CLASS VECTOR TABLE
       0C B0   16   0143   365          JSB     @CLASS_DS_TRAN(R0)      ;INVOKE MODEM TRANSITION ROUTINE
          07   BA   0146   366          POPR    #^M<R0,R1,R2>           ;RESTORE REGISTERS
               05   0148   367          RSB
```

```
                     0149   369          .SBTTL  OUTPUT MODEM CONTROL
                     0149   370 ;++
                     0149   371 ; CON$DS_SET - SET OUTPUT MODEM SIGNALS
                     0149   372 ;
                     0149   373 ; FUNCTIONAL DESCRIPTION:
                     0149   374 ;
                     0149   375 ; THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT
                     0149   376 ;
                     0149   377 ; INPUTS:
                     0149   378 ;
                     0149   379 ;       R2 = LOW BYTE - SIGNALS TO ACTIVATE
                     0149   380 ;            HIGH BYTE - SIGNALS TO DEACTIVATE
                     0149   381 ;
                     0149   382 ;       R5 = UCB ADDRESS
                     0149   383 ;
                     0149   384 ; OUTPUTS:
                     0149   385 ;
                     0149   386 ;       R0-R3 ARE USED.
                     0149   387 ;--
                     0149   388 CON$DS_SET::
        0125 C5   52   88  0149   389          BISB2   R2,UCB$B_TT_DS_TX(R5)   ;SET NEW OUTPUT SIGNALS
     52   52    F8 8F   78  014E   390          ASHL    #-8,R2,R2               ;ACCESS SIGNALS TO RESET
        0125 C5   52   8A  0153   391          BICB2   R2,UCB$B_TT_DS_TX(R5)   ;RESET THEM
          53    24 A5   D0  0158   392          MOVL    UCB$L_CRB(R5),R3        ;GET CRB ADDRESS
51   0125 C5   01   01   EF  015C   393          EXTZV   #TT$V_DS_DTR,#1,UCB$B_TT_DS_TX(R5),R1 ;GET CURRENT DTR FOR LINE
     51   51    54 A5   78  0163   394          ASHL    UCB$W_UNIT(R5),R1,R1    ;SHIFT TO RELATIVE LINE POSITION
     1E A3   0106 C5   8A  0168   395          BICB2   UCB$W_TT_UNITBIT(R5),CRB$B_DZ_DTR(R3) ;RESET CURRENT DTR FOR LINE
        1E A3   51   88  016E   396          BISB2   R1,CRB$B_DZ_DTR(R3)      ;SET IT IF NEED BE
        50   1E A3   9A  0172   397          MOVZBL  CRB$B_DZ_DTR(R3),R0     ;GET NEW DTR BITS
     50   50    10   78  0176   398          ASHL    #RXS_V_DTR,R0,R0        ;SHIFT INTO POSITION
   50    00000040 8F   C8  017A   399          BISL2   #RXS_M_IE,R0            ;SET INTERRUPT ENABLE ALWAYS
           20    50   DA  0181   400          MTPR    R0,#PR$_RXCS            ;SET NEW DTR SIGNALS
                 05   0184   401          RSB
```

```
                                        .SBTTL CONSOLE RECEIVER INTERRUPT DISPATCHER
                                ;++
                                ; CON$INTINP - CONSOLE INTERRUPT ON INPUT READY
                                ;
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ; THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
                                ; CONSOLE INTERFACE. THE INTERRUPT CAN BE GENERATED BY THE CONSOLE
                                ; TERMINAL, REMOTE SERVICES PORT, ENVIRONMENTAL MONITOR, OR THE
                                ; 11/790 LOGICAL CONSOLE.
                                ;
                                ; CONSOLE TERMINAL:
                                ; REMOTE SERVICES PORT:
                                ;                       ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
                                ;                       UNSOLICITED AND RESULT IN AN ENTRY INTO THE
                                ;                       TERMINAL DRIVER COMMON CHARACTER BUFFERING
                                ;                       ROUTINE "@UCB$L_TT_PUTNXT(R5)".
                                ;
                                ; ENVIRONMENTAL MONITOR:
                                ;
                                ;                       INTERRUPT CAN EITHER BE AN ALERT FOR TEMPERATURE
                                ;                       OR BLOWER, OR A PERIODIC SAMPLING.
                                ;
                                ; LOGICAL CONSOLE:
                                ;
                                ;                       TEST FOR NOTIFICATION THAT THE CONSOLE WAS REBOOTED
                                ;                       SUCCESSFULLY; IF SO, LOG THE EVENT.  THERE IS
                                ;                       CURRENTLY NO OTHER FULL DRIVER SUPPORT FOR THE LOGICAL
                                ;                       CONSOLE LINE.
                                ;
                                ; INPUTS:
                                ;
                                ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
                                ;
                                ;       00(SP) = ADDRESS OF THE IDB
                                ;
                                ; OUTPUTS:
                                ;
                                ;       THE SAVED REGISTERS ARE RESTORED BEFORE REI.
                                ;--
                                        .ENABLE LOCAL_BLOCK

                                CON$INTINP::
 53   21   DB                            MFPR    #PR$_RXDB,R3            ;MOVE DATA FROM INTERFACE
                                ;
                                ; GET THE ASSOCIATED UCB
                                ;
 54   9E   D0                            MOVL    @(SP)+,R4               ;GET IDB ADDRESS
52  53   F8 8F   78                      ASHL    #-8,R3,R2               ;GET LINE NUMBER
   52   FFF0 8F   AA                     BICW    #^C<^XF>,R2             ;
                                ;
                                ; SEE IF A UCB IS ASSOCIATED WITH THE LINE.  CURRENTLY, THE LOCAL CONSOLE
                                ; UCB IS ALWAYS PRESENT, AND SYSGEN SUPPORTS ADDING A UCB FOR THE REMOTE
                                ; CONSOLE, ENVIRONMENTAL MONITOR (EMM), AND LOGICAL CONSOLE LINES.  HOWEVER,
                                ; THERE IS CURRENTLY NO USER-REQUESTED TRANSFER SUPPORT FOR THE EMM OR
                                ; LOGICAL CONSOLE IN OPDRV790.  BUT WE MUST DISPATCH THE EMM AND LOGICAL
                                ; CONSOLE INTERRUPTS TO ROUTINES THAT CHECK FOR CERTAIN UNSOLICITED MESSAGES,
```

```
                          0195   460 ; SUCH AS ENVIRONMENTAL ALERT CONDITIONS AND "CONSOLE REBOOT SUCCESSFUL"
                          0195   461 ; MESSAGES.
                          0195   462 ;
          52    0E   B3   0195   463          BITW    #^XE,R2                 ;IS IT A DEVICE INTERRUPT?
                          0198   464                                          ; (IDS 0-1 MAY HAVE A UCB ASSOCIATED)
             07   12   0198   465          BNEQ    7$                      ;IF NEQ NO. SKIP UCB TEST
    55    18 A442   D0   019A   466 5$:      MOVL    IDB$L_UCBLST(R4)[R2],R5 ;GET THE UCB ADDRESS
             15   13   019F   467          BEQL    30$                     ;NO UCB - DISMISS INTERRUPT
 50   FE5A CF42   DE   01A1   468 7$:      MOVAL   W^CON$INPDISTAB[R2],R0  ;GET ADDRESS TO VECTOR TO
          00 B0   17   01A7   469          JMP     @(R0)                   ;DISPATCH TO PROCESS INTERRUPT
                          01AA   470 ;
                          01AA   471 ;CONSOLE TERMINAL INTERRUPT
                          01AA   472 ;
                          01AA   473 INT_CONSOLINP:
          53    53   9A   01AA   474          MOVZBL  R3,R3                   ;ZERO TOP 3 BYTES
        0110 D5   16   01AD   475          JSB     @UCB$L_TT_PUTNXT(R5)    ;BUFFER THE CHARACTER
             03   13   01B1   476          BEQL    30$                     ;IF EQL THEN NO CHARACTER TO OUTPUT
           0179   30   01B3   477 20$:     BSBW    CON$STARTIO             ;OUTPUT THE CHARACTER
           0249   31   01B6   478 30$:     BRW     DISMIS                  ;GO
                          01B9   479
                          01B9   480          .DISABLE LOCAL_BLOCK
```

```
                                01B9    482          .SBTTL  ENVIRONMENTAL MONITOR INPUT INTERRUPT
                                01B9    483 ;++
                                01B9    484 ; FUNCTIONAL DESCRIPTION:
                                01B9    485 ;
                                01B9    486 ; THE CONSOLE INTERRUPTS ON THIS LINE TO WARN OF ENVIRONMENTAL CONDITIONS
                                01B9    487 ; SUCH AS TEMPERATURE OR VOLTAGE OUT OF ACCEPTABLE RANGES.  LOG THE
                                01B9    488 ; WARNING AND SEND A MESSAGE TO THE CONSOLE TERMINAL.
                                01B9    489 ;
                                01B9    490 ; INPUTS:
                                01B9    491 ;       R3      - CONTENTS OF PR$_RXDB REGISTER
                                01B9    492 ;--
                                01B9    493
                                01B9    494 ;
                                01B9    495 ; LOCAL DATA USED FOR EMM ALERT MESSAGES.
                                01B9    496 ;
                                01B9    497 EMM_MESSAGE:
45 54 53 59 53 25 07 07 07 0A 0A 0D 01B9    498          .ASCII  <13><10><10><7><7><7>-
65 6D 6E 6F 72 69 76 6E 45 20 2C 4D 01C5
2D 20 74 72 65 6C 41 20 6C 61 74 6E 01D1
74 6E 65 6D 6E 6F 72 69 76 6E 45 20 01DD
68 20 72 6F 74 69 6E 6F 4D 20 6C 61 01E9
20 64 65 74 63 65 74 65 64 20 73 61 01F5
                                0201    499 \%SYSTEM, Environmental Alert - Environmental Monitor has detected \-
                             0D 0201    500 <13><10>-
6F 63 20 74 72 65 6C 61 20 6E 61 0A 0202    501 \an alert condition.  Please check the error log.\-
6C 50 20 20 2E 6E 6F 69 74 69 64 6E 020E
74 20 6B 63 65 68 63 20 65 73 61 65 021A
67 6F 6C 20 72 6F 72 72 65 20 65 68 0226
                             2E 0232
                          0A 0D 0233    502 <13><10>
                                0235    503
                                0235    504 SHUTDOWN_MESSAGE:
                                0235    505          .ASCII  -
6D 65 74 73 79 73 20 6C 61 74 6F 54 0235    506 \Total system power shutdown pending if condition is not corrected.\<13><10>
64 74 75 68 73 20 72 65 77 6F 70 20 0241
20 67 6E 69 64 6E 65 70 20 6E 77 6F 024D
6E 6F 69 74 69 64 6E 6F 63 20 66 69 0259
72 72 6F 63 20 74 6F 6E 20 73 69 20 0265
             0A 0D 2E 64 65 74 63 65 0271
                                0279    507
                                0279    508 SHUTDOWN_MSGLEN:
                          0044  0279    509          .WORD   SHUTDOWN_MSGLEN - SHUTDOWN_MESSAGE
                                027B    510 EMM_MSGLEN:
                          007C  027B    511          .WORD   SHUTDOWN_MESSAGE - EMM_MESSAGE
                                027D    512 EMM_BUFFER:                                 ; LENGTH OF ALERT MESSAGES IS 2 BYTES
                          0000  027D    513          .WORD   0
                                027F    514 EMM_IGNORECNT:                              ; HOW MANY BYTES TO IGNORE
                            00  027F    515          .BYTE   0
                                0280    516
                                0280    517 ;
                                0280    518 ; EMM FLAGS BYTE AND THE FLAG BIT DEFINITIONS
                                0280    519 ;
                                0280    520 EMM_FLAGS:                                  ; TO HELP KEEP TRACK OF WHERE WE ARE IN
                            00  0280    521          .BYTE   0                          ; THE PROTOCOL
                    00000000    0281    522 EMM_IN_PROGRESS = 0
                    00000001    0281    523 EMM_V_SHUTDOWN = 1
                    00000002    0281    524 EMM_V_IGNOREINP = 2
```

```
                                                K 8
00000003  0281    525 EMM_V_BYTECOUNT = 3
                  0281    526 ;
                  0281    527 ; DEFINE FORMAT OF FIRST BYTE OF EMM ALERT MESSAGE:
                  0281    528 ;
                  0281    529 ;       <07>    - WHEN SET, THIS IS AN EMM ALERT CONDITION, AS OPPOSED TO
                  0281    530 ;                 DATA RETURNED IN RESPONSE TO A REQUEST
                  0281    531 ;       <06>    - WHEN SET, THIS IS AN AUTOMATIC SHUTDOWN CONDITION.  IF THE
                  0281    532 ;                 CONDITION IS NOT CLEARED IN A SMALL NUMBER OF MINUTES (1-2),
                  0281    533 ;                 THE CPU WILL BE POWERED DOWN.
                  0281    534 ;       <05>    - RESERVED FOR FUTURE USE.  NOT GUARANTEED TO BE ZERO.
                  0281    535 ;       <04:00> - IDENTIFIES WHICH ALERT CONDITION IS BEING SIGNALLED
                  0281    536 ;
00000007  0281    537 EMM_V_ALERT = 7
00000006  0281    538 EMM_V_ASD = 6
00000000  0281    539 EMM_V_CONDITION = 0
00000005  0281    540 EMM_S_CONDITION = 5
                  0281    541 ;
                  0281    542 ; DISPATCH HERE FOR INPUT INTERRUPT FROM EMM
                  0281    543 ;
                  0281    544 INT_EMMINP:
6D FB AF   02  E0 0281    545         BBS     #EMM_V_IGNOREINP, -     ; BRANCH IF WE ARE IGNORING THIS
                  0286    546                 EMM_FLAGS, -            ; MESSAGE
                  0286    547                 EMM_IGNORE_DATA
16 F6 AF   00  E2 0286    548         BBSS    #EMM_IN_PROGRESS, -     ; BRANCH IF THIS IS NOT THE FIRST BYTE
                  028B    549                 EMM_FLAGS, -            ; OF AN EMM MESSAGE
                  028B    550                 EMM_LAST_BYTE
                  028B    551 ;
                  028B    552 ; HANDLE THE FIRST BYTE OF AN ENVIRONMENTAL MONITOR ALERT MESSAGE
                  028B    553 ;
                  028B    554 EMM_FIRST_BYTE:
58 53      07  E1 028B    555         BBC     #EMM_V_ALERT,R3,-       ; DON'T HANDLE RESPONSES TO EMM REQUESTS
                  028F    556                 EMM_RESPONSE            ; YET
   EB AF       B4 028F    557         CLRW    EMM_BUFFER              ; ZERO THE BUFFER
05 53      06  E1 0292    558         BBC     #EMM_V_ASD,R3,10$       ; BRANCH IF NOT AUTOMATIC SHUTDOWN
                  0296    559                                         ; CONDITION
      01   E2 0296    560         BBSS    #EMM_V_SHUTDOWN,-       ; FLAG AN AUTOMATIC SHUTDOWN SO IT
   00 E6 AF       0298    561                 EMM_FLAGS,10$           ; CAN BE REPORTED TO THE CONSOLE TERM
DE AF      53  90 029B    562 10$:    MOVB    R3,EMM_BUFFER           ; BUFFER THIS BYTE OF DATA
      43   11 029F    563         BRB     DISMIS_EMM              ; DISMISS THIS INTERRUPT
                  02A1    564 ;
                  02A1    565 ; HANDLE LAST BYTE OF AN EMM MESSAGE HERE.  WRITE AN ERRORLOG ENTRY AND
                  02A1    566 ; BROADCAST A WARNING TO THE CONSOLE TERMINAL.
                  02A1    567 ;
                  02A1    568 EMM_LAST_BYTE:
D9 AF      53  90 02A1    569         MOVB    R3,EMM_BUFFER+1         ; PUT THE DATA IN 2ND BYTE OF THE BUFFER
   51      12  D0 02A5    570         MOVL    #EMB$C_HD_LENGTH+2,R1   ; SIZE OF ERRORLOG BUFFER TO ALLOCATE
00000000'GF    16 02A8    571         JSB     G^ERL$ALLOCEMB          ; ALLOCATE ERRORLOG BUFFER
   0F 50       E9 02AE    572         BLBC    R0,BROADCAST            ; BRANCH IF NO BUFFER AVAILABLE
04 A2      0F  B0 02B1    573         MOVW    #EMB$C_EMM,EMB$W_HD_ENTRY(R2)
                  02B5    574                                         ; SET THE ERRORLOG TYPE
10 A2      C5 AF  B0 02B5    575         MOVW    EMM_BUFFER,EMB$C_HD_LENGTH(R2)
                  02BA    576                                         ; MOVE THE DATA INTO THE ERRORLOG BUFFER
00000000'GF    16 02BA    577         JSB     G^ERL$RELEASEMB         ; RELEASE THE ERRORLOG DATA
                  02C0    578 BROADCAST:
   51   B8 AF  3C 02C0    579         MOVZWL  EMM_MSGLEN,R1           ; LENGTH OF THE MESSAGE
04 B8 AF      01  E5 02C4    580         BBCC    #EMM_V_SHUTDOWN, -      ; BRANCH IF THIS IS NOT AN AUTOMATIC
                  02C9    581                 EMM_FLAGS,10$           ; SHUTDOWN CONDITION
```

```
                51    AD AF    A0   02C9    582              ADDW    SHUTDOWN_MSGLEN,R1      ; ADD SHUTDOWN MESSAGE TO BROADCAST
              52    FEE8 CF    9E   02CD    583 10$:         MOVAB   EMM_MESSAGE,R2          ; ADDRESS OF MESSAGE
        55   00000000'GF       9E   02D2    584              MOVAB   G^OPA$UCB0,R5           ; SEND IT TO THE CONSOLE TERMINAL
             00000000'GF       16   02D9    585              JSB     G^IOC$BROADCAST         ; BROADCAST THE MESSAGE
                               02DF    586 EMM_ALERT_DONE:
                         00    E5   02DF    587              BBCC    #EMM_IN_PROGRESS,-      ; FINISHED WITH THIS EMM MESSAGE
                   00 9D AF          02E1    588                      EMM_FLAGS,DISMIS_EMM
                               02E4    589 DISMIS_EMM:
                      011B     31   02E4    590              BRW     DISMIS
                               02E7    591 ;
                               02E7    592 ; WE GET HERE ONLY IF WE'RE FAIRLY MIXED UP: WE DON'T SUPPORT REQUESTING DATA
                               02E7    593 ; FROM THE EMM BUT THE EMM LINE IS GIVING US A RESPONSE TO A DATA REQUEST.
                               02E7    594 ; READ THE BYTECOUNT AND IGNORE THAT MANY BYTES OF DATA FROM THE EMM LINE.
                               02E7    595 ;
                               02E7    596 EMM_RESPONSE:
            00 95 AF    02     E2   02E7    597              BBSS    #EMM_V_IGNOREINP, -     ; SIGNAL WE'RE GETTING AN EMM RESPONSE
                               02EC    598                      EMM_FLAGS,10$
            00 90 AF    03     E2   02EC    599 10$:         BBSS    #EMM_V_BYTECOUNT, -     ; SIGNAL THE NEXT BYTE SHOULD BE THE
                               02F1    600                      EMM_FLAGS,20$           ; MESSAGE BYTE COUNT
                        F1     11   02F1    601 20$:         BRB     DISMIS_EMM
                               02F3    602
                               02F3    603 EMM_IGNORE_DATA:
            06 89 AF    03     E5   02F3    604              BBCC    #EMM_V_BYTECOUNT, -     ; BRANCH IF THIS IS NOT THE BYTECOUNT
                               02F8    605                      EMM_FLAGS,10$           ; BYTE
               83 AF    53     90   02F8    606              MOVB    R3,EMM_IGNORECNT        ; SAVE NUMBER OF BYTES TO IGNORE
                        E6     11   02FC    607              BRB     DISMIS_EMM              ; THAT'S IT FOR THIS BYTE
                               02FE    608 10$:
                  FF7D CF      97   02FE    609              DECB    EMM_IGNORECNT           ; COUNT DOWN ONE MORE BYTE
                        E0     12   0302    610              BNEQ    DISMIS_EMM              ; BRANCH IF STILL MORE TO IGNORE
         00 FF77 CF    02     E5   0304    611              BBCC    #EMM_V_IGNOREINP, -     ; CLEAR THE "IGNORE EMM DATA" FLAG
                               030A    612                      EMM_FLAGS,20$           ;
                        D8     11   030A    613 20$:         BRB     DISMIS_EMM              ; THAT'S IT
```

```
                     030C   615          .SBTTL  LOGICAL CONSOLE INPUT INTERRUPTS
                     030C   616 ;++
                     030C   617 ;INT_LOGINP - HANDLE LOGICAL CONSOLE INPUT INTERRUPTS
                     030C   618 ;
                     030C   619 ; FUNCTIONAL DESCRIPTION:
                     030C   620 ;
                     030C   621 ; ONLY ONE TYPE OF LOGICAL CONSOLE INTERRUPT-LEVEL INPUT IS CURRENTLY HANDLED:
                     030C   622 ; AN UNSOLICITED MESSAGE SIGNALLING THAT THE CONSOLE WAS RE-BOOTED SUCCESSFULLY.
                     030C   623 ; THE CONSOLE CAN BE RE-BOOTED BY THE ROUTINE CON$KEEPALIVE IF IT DETECTS THAT
                     030C   624 ; THE CONSOLE HAS DIED.  CON$KEEPALIVE WILL WRITE A MESSAGE TO THE SYSTEM
                     030C   625 ; ERRORLOG NOTING THAT IT HAS ATTEMPTED TO RE-BOOT THE CONSOLE.  IF THE
                     030C   626 ; REBOOT IS SUCCESSFUL, THIS ROUTINE WILL WRITE ANOTHER MESSAGE TO THE
                     030C   627 ; ERRORLOG SIGNALLING THE SUCCESSFUL REBOOT.
                     030C   628 ;
                     030C   629 ; INPUTS:
                     030C   630 ;       R3      - CONTENTS OF PR$_RXDB REGISTER
                     030C   631 ;--
                     030C   632 INT_LOGINP:
 40 8F    53   91    030C   633          CMPB    R3,#RXDB$C_LOG_REBOOT   ; Console reboot message?
          1A   12    0310   634          BNEQ    DISMIS_LOG              ; Branch if not.
    51    11   D0    0312   635          MOVL    #EMB$C_HD_LENGTH+1,R1   ; Allocate a header plus one byte.
00000000'GF    16    0315   636          JSB     G^ERL$ALLOCEMB          ; Allocate space in the errorlog buffer.
      0E   50   E9    031B   637          BLBC    R0,DISMIS_LOG           ; Branch if unable to allocate.
 04 A2    11   B0    031E   638          MOVW    #EMB$C_CRBT, -          ; Set entry type = console reboot.
                     0322   639                  EMB$W_HD_ENTRY(R2)      ;
 05 A2    01   90    0322   640          MOVB    #1,EMB$W_HD_ENTRY+1(R2) ; Set flag = reboot successful.
00000000'EF    16    0326   641          JSB     ERL$RELEASEMB           ; Release the errorlog data.
                     032C   642 DISMIS_LOG:
       00D3   31    032C   643          BRW     DISMIS
```

```
                                032F    645          .SBTTL  START I/O ON CONSOLE INTERFACE
                                032F    646 ;++
                                032F    647 ;CON$STARTIO - START I/O ON CONSOLE INTERFACE
                                032F    648 ;
                                032F    649 ; FUNCTIONAL DESCRIPTION:
                                032F    650 ;
                                032F    651 ; THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
                                032F    652 ; THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE NEXT READY INTERRUPT.
                                032F    653 ;
                                032F    654 ; A RETURN TO THE CALLER IS DONE TO ENTER A ''WAIT FOR INTERRUPT'' STATE.
                                032F    655 ;
                                032F    656 ; INPUTS:
                                032F    657 ;
                                032F    658 ;       R3 = DATA TO OUTPUT
                                032F    659 ;       R5 = UCB ADDRESS
                                032F    660 ;
                                032F    661 ; OUTPUTS:
                                032F    662 ;
                                032F    663 ;       R3,R4,R5 ARE PRESERVED.
                                032F    664 ;--
                                032F    665
                                032F    666          .ENABLE LOCAL_BLOCK
                                032F    667 CON$STARTIO::
                          3C    032F    668          BGEQ    20$                     ;BRANCH IF SINGLE CHARACTER
                0800 8F   A8    0331    669          BISW2   #TTY$M_TANK_BURST,-     ;FLAG BURST MODE ACTIVE
                0108 C5         0335    670                  UCB$W_TT_HOLD(R5)
                                0338    671 START_TERM_IO:
            51    24 A5   D0    0338    672 10$:     MOVL    UCB$L_CRB(R5),R1        ;GET CRB ADDRESS
            51    2C A1   D0    033C    673          MOVL    CRB$L_INTD+VEC$L_IDB(R1),R1 ;GET IDB ADDRESS
      0E A1     0106 C5   88    0340    674          BISB2   UCB$W_TT_UNITBIT(R5),IDB$B_TT_ENABLE(R1) ;CREATE NEW XMT ENABLE MSK
            51    0E A1   9A    0346    675          MOVZBL  IDB$B_TT_ENABLE(R1),R1  ;GET ENABLE MASK FOR THIS DEVICE
         51    51    10    78    034A    676          ASHL    #TXS_V_TEM,R1,R1        ;POSITION TO FIELD
   51     00008040 8F   C8    034E    677          BISL2   #TXS_M_IE!TXS_M_WRTENA,R1 ;SET INTERRUPT ENABLE
            64 A5   02    88    0355    678          BISB2   #UCB$M_INT,UCB$W_STS(R5);FLAG INTERRUPT EXPECTED
                            0359    679
                            0359    680          DSBINT  #^X14                   ;MAKE FOLLOWING TEST AND SET ATOMIC
            50    22    DB    035F    681 15$:     MFPR    #PR$_TXCS,R0            ;READ TRANSMIT STATUS
            F9 50   07    E1    0362    682          BBC     #TXS_V_RDY,R0,15$       ;WAIT 'TIL WE HAVE A VALID COPY
               22   51    DA    0366    683          MTPR    R1,#PR$_TXCS            ;ENABLE THIS DEVICE
                            0369    684          ENBINT
                      05    036C    685          RSB                             ;RETURN WITH INTERRUPT EXPECTED
                            036D    686
         0108 C5    53    90    036D    687 20$:     MOVB    R3,UCB$W_TT_HOLD(R5)    ;SAVE OUTPUT CHARACTER
               0400 8F    A8    0372    688          BISW2   #TTY$M_TANK_HOLD,-      ;SIGNAL CHARACTER IN TANK
               0108 C5          0376    689                  UCB$W_TT_HOLD(R5)
                      BD    11    0379    690          BRB     10$                     ;GO ENABLE INTERRUPT
                            037B    691          .DISABLE LOCAL_BLOCK
```

```
                                037B   693          .SBTTL  CARRIER CHANGE SENT BY CONSOLE
                                037B   694 ;++
                                037B   695 ; FUNCTIONAL DESCRIPTION:
                                037B   696 ;
                                037B   697 ; DISPATCH TO HERE IF THE CONSOLE IS TELLING US ABOUT A CHANGE IN
                                037B   698 ; CARRIER STATUS
                                037B   699 ;
                                037B   700 ;--
                                037B   701
                                037B   702 INT_CARCHANGE:
          55   18 A4    D0      037B   703          MOVL    IDB$L_UCBLST(R4),R5      ;GET ADDRESS OF CONSOLE UCB
          54   24 A5    D0      037F   704          MOVL    UCB$L_CRB(R5),R4         ;GET CRB ADDRESS IN R4
    53    53   F0 8F    78      0383   705          ASHL    #-16,R3,R3               ;POSITION CARRIER BITS FROM RXDB REG
    52    53   1D A4    8D      0388   706          XORB3   CRB$B_DZ_CARRIER(R4),R3,R2 ;FIND CHANGED BITS
          1D A4  53      90      038D   707          MOVB    R3,CRB$B_DZ_CARRIER(R4) ;SAVE STATE OF LOGICAL CARRIER
                                0391   708 ;
                                0391   709 ; PROCESS LINES WITH CARRIER CHANGE
                                0391   710 ;
51  52   08    00      EA      0391   711 20$:    FFS     #0,#8,R2,R1              ;FIND A LINE
                   13   13      0396   712          BEQL    50$                      ;IF EQL NO MORE
          00 52    51   E1      0398   713          BBC     R1,R2,30$                ;CLEAR BIT
          55   2C A4    D0      039C   714 30$:    MOVL    CRB$L_INTD+VEC$L_IDB(R4),R5 ;GET IDB ADDRESS
          55   18 A541  D0      03A0   715          MOVL    IDB$L_UCBLST(R5)[R1],R5  ;GET CORRESPONDING UCB ADDR FOR UNIT
                   EA   13      03A5   716          BEQL    20$                      ;BRANCH IF NO UCB
                   05   10      03A7   717          BSBB    REPORT_CARCHANGE         ;REPORT THE CARRIER CHANGE
                   E6   11      03A9   718          BRB     20$                      ;CONTINUE SCANNING
                                03AB   719 50$:
                                03AB   720 BR_DISMIS:
                0054    31      03AB   721          BRW     DISMIS                   ;DISMISS THE INTERRUPT
                                03AE   722 ;++
                                03AE   723 ; FUNCTIONAL DESCRIPTION:
                                03AE   724 ;
                                03AE   725 ; THIS ROUTINE REPORTS A CHANGE IN CARRIER STATUS TO THE TERMINAL
                                03AE   726 ; CLASS DRIVER
                                03AE   727 ;
                                03AE   728 ; INPUTS:
                                03AE   729 ;
                                03AE   730 ;       R5 = UCB ADDRESS
                                03AE   731 ;       R4 = CRB ADDRESS
                                03AE   732 ;
                                03AE   733 ; OUTPUTS:
                                03AE   734 ;
                                03AE   735 ;       DATASET TRANSITION REPORTED TO CLASS DRIVER
                                03AE   736 ;
                                03AE   737 ;       REGISTERS R0,R1,R2 DESTROYED, ALL OTHERS PRESERVED
                                03AE   738 ;--
                                03AE   739 REPORT_CARCHANGE:
       2D 38 A5    02      E1   03AE   740          BBC     #DEV$V_TRM,UCB$L_DEVCHAR(R5),20$ ; BRANCH IF NOT TERMINAL
       28 44 A5    15      E1   03B3   741          BBC     #TT$V_MODEM,UCB$L_DEVDEPEND(R5),20$ ;BRANCH IF NOT MODEM
50  1D A4    01    51      EF   03B8   742          EXTZV   R1,#1,CRB$B_DZ_CARRIER(R4),R0 ;GET CARRIER BIT FOR THIS LINE
       01    05    50      F0   03BE   743          INSV    R0,#TT$V_DS_CARRIER,#1,-;SET/CLEAR IN UCB
                0124 C5         03C2   744                  UCB$B_TT_DS_RCV(R5)
                  D0 8F    89   03C5   745          BISB3   #<TT$M_DS_DSR!TT$M_DS_CTS!TT$M_DS_RING>,-
          0124 C5    51         03C8   746                  R1,UCB$B_TT_DS_RCV(R5) ;SET BITS WHICH ARE ALWAYS SET
          52   0124 C5    9A   03CC   747          MOVZBL  UCB$B_TT_DS_RCV(R5),R2  ;GET CURRENT RCV MODEM STATUS
               51   03    9A   03D1   748          MOVZBL  #MODEM$C_DATASET,R1     ;SIGNAL DATASET TRANSITION
          50   0114 C5    D0   03D4   749          MOVL    UCB$L_TT_CLASS(R5),R0   ;GET CLASS VECTOR TABLE
```

```
                1E   BB  03D9   750          PUSHR   #^M<R1,R2,R3,R4>        ;SAVE VOLATILE REGISTERS
            0C  B0   16  03DB   751          JSB     @CLASS_DS_TRAN(R0)      ;SIGNAL DATASET TRANSITION
                1E   BA  03DE   752          POPR    #^M<R1,R2,R3,R4>        ;RESTORE REGISTERS
                     05  03E0   753 20$:     RSB                             ;RETURN TO CALLER
```

```
                                03E1    755          .SBTTL  CONSOLE TRANSMITTER INTERRUPT SERVICE
                                03E1    756 ;++
                                03E1    757 ; CON$INTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE
                                03E1    758 ;
                                03E1    759 ; FUNCTIONAL DESCRIPTION:
                                03E1    760 ;
                                03E1    761 ; THIS ROUTINE IS ENTERED WHEN A CONSOLE UNIT READY INTERRUPT OCCURS.
                                03E1    762 ;
                                03E1    763 ; THE INTERRUPT STATE OF THE UNIT IS CHECKED FOR EXPECTED INTERRUPT.
                                03E1    764 ; IF NO INTERRUPT IS EXPECTED, THE INTERRUPT IS DISMISSED. IF AN INTERRUPT
                                03E1    765 ; IS EXPECTED THEN THE DRIVER IS ENTERED. IN THE CASE OF THE CONSOLE TERMINAL,
                                03E1    766 ; A SPECIFIC ROUTINE IS ENTERED TO GET THE NEXT CHARACTER AVAILABLE TO OUTPUT
                                03E1    767 ; ON THE UNIT.
                                03E1    768 ;
                                03E1    769 ; INPUTS:
                                03E1    770 ;
                                03E1    771 ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
                                03E1    772 ;
                                03E1    773 ;       00(SP) = ADDRESS OF THE IDB
                                03E1    774 ;
                                03E1    775 ; OUTPUTS:
                                03E1    776 ;
                                03E1    777 ;       THE SAVED REGISTERS ARE RESTORED BEFORE REI.
                                03E1    778 ;--
                                03E1    779 CON$INTOUT::                                ;
          50    9E    D0      03E1    780          MOVL    @(SP)+,R0                  ;GET ADDRESS OF IDB
                                03E4    781 ;
                                03E4    782 ; Note that R0 contains the address of the IDB throughout this routine.
                                03E4    783 ;
          53    2?    DB      03E4    784          MFPR    #PR$_TXCS,R3               ;READ STATUS REGISTER
    53    53     F8  8F    78  03E7    785          ASHL    #-8,R3,R3                  ;EXTRACT ID FIELD
    53  FFFFFFF0  8F    CA  03EC    786          BICL2   #^C<^XF>,R3                ;CLEAR UNINTERESTING BITS
    55    18 A043   D0      03F3    787          MOVL    IDB$L_UCBLST(R0)[R3],R5    ;GET ADDRESS OF UCB
             08     13      03F8    788          BEQL    DISMIS                     ;BRANCH IF NO UCB
                            03FA    789          CASE    R3,-                       ;DISPATCH TO PROCESS
                            03FA    790                  <INTOUT_TERM,INTOUT_TERM>,-
                            03FA    791                  TYPE=B
                            0402    792 INT_IGNOREINP:
                            0402    793 DISMIS:                                     ;
    50    8E    7D      0402    794          MOVQ    (SP)+,R0                   ;RESTORE REGISTERS
    52    8E    7D      0405    795          MOVQ    (SP)+,R2                   ;
    54    8E    7D      0408    796          MOVQ    (SP)+,R4                   ;
                02      040B    797          REI                                ;
                            040C    798 ;
                            040C    799 ; DEVICE IS A TERMINAL, EITHER THE CONSOLE OR THE REMOTE SERVICES PORT
                            040C    800 ;
                            040C    801 INTOUT_TERM:
                            040C    802 ;
                            040C    803 ;       CHECK FOR BURST ACTIVE ON LINE
                            040C    804 ;
             08     91      040C    805          CMPB    #TTY$M_TANK_BURST@-8,-     ;ONLY BURST ACTIVE?
    0109     C5             040E    806                  UCB$W_TT_HOLD+1(R5)
             54     13      0411    807          BEQL    CON_BURST                  ;YES, CONTINUE BURST
                            0413    808 ;
                            0413    809 ;       LOOK FOR NEXT OUTPUT STATE IN TANK
                            0413    810 ;
                            0413    811
```

```
                                0413   812          FFS     #0,#6,UCB$W_TT_HOLD+1(R5),R3
                                041A   813          CASE    R3,TYPE=B,<=                    ;DISPATCH
                                041A   814                  CON_PREMPT,-                    ;SEND PREMPT CHARACTER
                                041A   815                  CON_STOP,-                      ;STOP OUTPUT
                                041A   816                  CON_CHAR,-                      ;CHAR IN TANK
                                041A   817                  CON_BURST,-                     ;BURST IN PROGRESS
                                041A   818                  >
                                0426   819 ;
                                0426   820 ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
                                0426   821 ;
                   64 A5   03   8A  0426   822          BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5) ;CLEAR TIMEOUT AND EXPECTED
                                042A   823 ;
                                042A   824 ; CALL CLASS DRIVER FOR MORE OUTPUT
                                042A   825 ;
                010C D5   16   042A   826          JSB     @UCB$L_TT_GETNXT(R5)    ;GET THE NEXT CHARACTER
          01    FF 8F   010B C5   8F   042E   827          CASEB   UCB$B_TT_OUTYPE(R5),#-1,#1 ;OPTIMIZE FOR THE SINGLE CHARACTER
                                0435   828                                          ;BY SETTING THE LIMIT TO 1
                     002B' 0435   829 40$:    .WORD   CON_START_BURST-40$     ;BURST SPECIFIED
                     000C' 0437   830          .WORD   CON_RESET_IE-40$        ;NONE
                                0439   831 ;
                                0439   832 ; OUTPUT A CHARACTER TO THE CONSOLE
                                0439   833 ;
                53   53   9A   0439   834 20$:    MOVZBL  R3,R3                   ;ENSURE ALL ZEROES
                23   53   DA   043C   835          MTPR    R3,#PR$_TXDB            ;OUTPUT CHARACTER
                     C1   11   043F   836          BRB     DISMIS
                                0441   837 ;
                                0441   838 ; DISABLE OUTPUT ON THIS LINE
                                0441   839 ;
                                0441   840 50$:
                                0441   841 CON_RESET_IE:
                     01   E0   0441   842          BBS     #UCB$V_INT,-            ;IF INT EXP, THEN DON'T RESET,
             BC 64 A5                   0443   843                  UCB$W_STS(R5),DISMIS    ; COULD HAVE BEEN SET DURING CALLBACK
    0E A0    0106 C5   8A   0446   844          BICB2   UCB$W_TT_UNITBIT(R5),IDB$B_TT_ENABLE(R0) ;CLEAR THIS DEVICE
       51    0E A0   9A   044C   845          MOVZBL  IDB$B_TT_ENABLE(R0),R1  ;GET NEW ENABLE MASK BITS
    51 51    10   78   0450   846          ASHL    #TXS_V_TEM,R1,R1        ;POSITION
51  00008040 8F   C8   0454   847          BISL2   #TXS_M_IE!TXS_M_WRTENA,R1 ;SET INTERRUPT ENABLE
          22 51   DA   045B   848          MTPR    R1,#PR$_TXCS            ;TELL CONSOLE
             A2   11   045E   849          BRB     DISMIS
                                0460   850
                                0460   851 CON_START_BURST:
          0800 8F   A8   0460   852          BISW    #TTY$M_TANK_BURST,-     ;SIGNAL BURST ACTIVE
          0108 C5              0464   853                  UCB$W_TT_HOLD(R5)
                                0467   854 ;
                                0467   855 ;  CONTINUE BURST OUTPUT
                                0467   856 ;
                                0467   857 CON_BURST:
    53    011C D5   9A   0467   858          MOVZBL  @UCB$L_TT_OUTADR(R5),R3 ;OUTPUT NEXT BYTE
          23   53   DA   046C   859          MTPR    R3,#PR$_TXDB
          011C C5   D6   046F   860          INCL    UCB$L_TT_OUTADR(R5)     ;UPDATE POINTER
          0120 C5   B7   0473   861          DECW    UCB$W_TT_OUTLEN(R5)     ;UPDATE COUNT
               07   12   0477   862          BNEQ    10$                     ;NOT LAST CHARACTER
          0800 8F   AA   0479   863          BICW    #TTY$M_TANK_BURST,-     ;RESET BURST ACTIVE
          0108 C5              047D   864                  UCB$W_TT_HOLD(R5)
             FF7F   31   0480   865 10$:    BRW     DISMIS
                                0483   866 ;
                                0483   867 ; OUTPUT SINGLE CHARACTER
                                0483   868 ;
```

```
                     0483   869 CON_CHAR:
53   0108 C5   9A   0483   870          MOVZBL  UCB$W_TT_HOLD(R5),R3    ;OUTPUT CHAR IN TANK
     23   53   DA   0488   871          MTPR    R3,#PR$_TXDB
     0400 8F   AA   048B   872          BICW    #TTY$M_TANK_HOLD,-      ;SHOW TANK EMPTY
     0108 C5        048F   873                  UCB$W_TT_HOLD(R5)
        FF6D   31   0492   874          BRW     DISMIS
                    0495   875 ;
                    0495   876 ; STOP THE OUTPUT
                    0495   877 ;
                    0495   878 CON_STOP:
          03   8A   0495   879          BICB    #UCB$M_INT!UCB$M_TIM,-
        64 A5        0497   880                  UCB$W_STS(R5)           ;RESET OUTPUT ACTIVE
           A6   11   0499   881          BRB     CON_RESET_IE            ;DISABLE INTERRUPT THIS DEVICE
                    049B   882 ;
                    049B   883 ; SEND XON OR XOFF CHARACTER
                    049B   884 ;
                    049B   885 CON_PREMPT:
     0100 8F   AA   049B   886          BICW    #TTY$M_TANK_PREMPT,-            ;RESET XOFF STATE
     0108 C5        049F   887                  UCB$W_TT_HOLD(R5)
23   010A C5   DA   04A2   888          MTPR    UCB$B_TT_PREMPT(R5),#PR$_TXDB   ;OUTPUT THE CHARACTER
        FF58   31   04A7   889          BRW     DISMIS                          ;AND DISMISS THE INTERRUPT
```

```
                                04AA   891 .SBTTL  CONSOLE PORT ACTION ROUTINES
                                04AA   892 ;++
                                04AA   893 ; CON$XOFF -    SEND XOFF
                                04AA   894 ; CON$XON -     SEND XON
                                04AA   895 ; CON$STOP -    STOP OUTPUT
                                04AA   896 ; CON$ABORT -   ABORT CURRENT OUTPUT
                                04AA   897 ; CON$RESUME -  RESUME STOPPED OUTPUT
                                04AA   898 ;
                                04AA   899 ; FUNCTIONAL DESCRIPTION:
                                04AA   900 ;
                                04AA   901 ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                                04AA   902 ; CONTROL OUTPUT ON THE PORT
                                04AA   903 ;
                                04AA   904 ; INPUTS:
                                04AA   905 ;
                                04AA   906 ;       R5 = UCB ADDRESS
                                04AA   907 ;
                                04AA   908 ; OUTPUTS:
                                04AA   909 ;
                                04AA   910 ;       R5 = UCB ADDRESS
                                04AA   911 ;--
                                04AA   912 CON$XOFF::
                                04AA   913 CON$XON::
        0108 C5   0100 8F   A8  04AA   914         BISW    #TTY$M_TANK_PREMPT,UCB$W_TT_HOLD(R5) ;SCHEDULE XOFF/XON
             010A C5    53   90  04B1   915         MOVB    R3,UCB$B_TT_PREMPT(R5)  ;SAVE THE CHARACTER IN THE PREMPT SLOT
          03 64 A5    01   E0  04B6   916         BBS     #UCB$V_INT,UCB$W_STS(R5),10$ ;IF OUTPUT ACTIVE, THEN DONE
                                04BB   917
                      FE7A   30  04BB   918         BSBW    START_TERM_IO           ;ENABLE OUTPUT INTERRUPTS
                             05  04BE   919 10$:    RSB
                                04BF   920
                                04BF   921 CON$STOP::
                0200 8F    A8  04BF   922         BISW    #TTY$M_TANK_STOP,-      ;SCHEDULE STOP
                0108 C5        04C3   923                 UCB$W_TT_HOLD(R5)
                             05  04C6   924         RSB
                                04C7   925
                                04C7   926 CON$ABORT::
             0108 C5    0B   E5  04C7   927         BBCC    #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),- ;RESET BURST ACTIVE
                             00 04CC   928                 10$
                                04CD   929 10$:
                                04CD   930         TIMSET  1                       ;SET A TIMEOUT
                                04E0   931                                         ;IN CASE OUTPUT ACTIVE
                             05  04E0   932         RSB
                                04E1   933
                                04E1   934 CON$RESUME::
                             51 DD  04E1   935         PUSHL   R1                      ;SAVE A REGISTER
        0108 C5    0200 8F    AA  04E3   936         BICW    #TTY$M_TANK_STOP-
                                04EA   937                 ,UCB$W_TT_HOLD(R5)      ;RESET STOP CONDITION
          21 0108 C5    0B   E1  04EA   938         BBC     #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),20$ ;BRANCH IF NO BURST IN PROGR
             51   0120 C5    3C  04F0   939         MOVZWL  UCB$W_TT_OUTLEN(R5),R1  ;NUMBER OF BURST CHARS
                                04F5   940         TIMSET  R1,R1                   ;SET THE TIMER
                          13   11  050F   941         BRB     30$
                                0511   942
                                0511   943 20$:     TIMSET  2                       ;CHAR IN TANK OR OTHER TIMEOUT
             03 64 A5    01   E0  0524   944 30$:     BBS     #UCB$V_INT,UCB$W_STS(R5),40$ ;SKIP IF OUTPUT ON
                      FE0C   30  0529   945         BSBW    START_TERM_IO           ;ENABLE OUTPUT INTERRUPT
                                052C   946 40$:
                          51 8ED0  052C   947         POPL    R1
```

05 052F 948 RSB

```
                                0530   950          .SBTTL  CON$SENDCONSCMD - Send command to 11/790 console
                                0530   951 ;++
                                0530   952 ; Functional Description:
                                0530   953 ;       CON$SENDCONSCMD is used to send a command to the "logical" console,
                                0530   954 ;       such as "Examine warm start flag" or "Reboot CPU".  It is also
                                0530   955 ;       used by code at IPL 31 to read data in console memory
                                0530   956 ;       (as opposed to using the logical console QIO interface).
                                0530   957 ;
                                0530   958 ; Inputs:
                                0530   959 ;       R0 = code for console command (see TXDB$C_LOG_xxx definitions)
                                0530   960 ;       R2 = # of bytes of data expected (if 0 then just send command)
                                0530   961 ;       R3 = address of buffer to store data in (only if R2 is non-zero)
                                0530   962 ;
                                0530   963 ; Outputs:
                                0530   964 ;       Data is stored in the buffer.
                                0530   965 ;       All registers preserved.
                                0530   966 ;--
                                0530   967
                                0530   968 CON$SENDCONSCMD::
                     0F   BB    0530   969          PUSHR   #^M<R0,R1,R2,R3>        ; Save working registers.
                                0532   970 ;
                                0532   971 ; Enable transmit/receive on logical console line.
                                0532   972 ;
                     7E   7C    0532   973          CLRQ    -(SP)                   ; Save RXCS and TXCS on stack.
               04 AE 20   DB    0534   974          MFPR    #PR$_RXCS,4(SP)         ; Save receive status register.
                  6E 22   DB    0538   975 20$:     MFPR    #PR$_TXCS,(SP)          ; Save transmit status register.
               F9 6E 07   E1    053B   976          BBC     #TXS_V_RDY,(SP),20$     ; Make sure we have a valid copy.
           FF00FFBF 8F    CA    053F   977          BICL    #^C<TXS_M_IEM!TXS_M_IE>,- ; Only save writable fields of TXCS
                  6E            0545   978                  (SP)                    ; (transmit enable mask and int enable).
         6E 00008000 8F    C8  0546   979          BISL    #TXS_M_WRTENA,(SP)      ; Set "write enable" in saved TXCS.
         22 00088000 8F    DA  054D   980 30$:     MTPR    #TXS_M_LOGCONS!TXS_M_WRTENA, - ; Disable all console lines except
                               0554   981                  #PR$_TXCS               ; logical console; also disable interrupts.
                     20 00    DA  0554   982          MTPR    #0,#PR$_RXCS            ; Disable interrupts in RXCS.
                                0557   983 ;
                                0557   984 ; Send command to logical console.
                                0557   985 ;
                  51 22    DB    0557   986 40$:     MFPR    #PR$_TXCS,R1            ; Get transmit status.
               F9 51 07    E1    055A   987          BBC     #TXS_V_RDY,R1,40$       ; Loop until ready bit is set.
         03    51 04 08    ED    055E   988          CMPZV   #TXS_V_ID,#TXS_S_ID,R1,#3 ; ID = logical console data?
                     E8 12    0563   989          BNEQ    30$                     ; If not, try again.
                  02 50 91    0565   990          CMPB    R0,#CON$C_BOOTCPU       ; Console function = boot CPU?
                     2E 13    0568   991          BEQL    90$                     ; Branch if so.
                  23 50 DA    056A   992          MTPR    R0,#PR$_TXDB            ; Else send request code to console.
                                056D   993 ;
                                056D   994 ; Get returned data (if any is expected).
                                056D   995 ;
                     52 D5    056D   996          TSTL    R2                      ; Any return data expected?
                     17 13    056F   997          BEQL    60$                     ; None; we're all through.
                  51 20    DB    0571   998 50$:     MFPR    #PR$_RXCS,R1            ; Get receiver status.
               F9 51 07    E1    0574   999          BBC     #RXS_V_DONE,R1,50$      ; Loop until done bit is set.
                  51 21    DB    0578  1000          MFPR    #PR$_RXDB,R1            ; Get received data.
         03    51 04 08    ED    057B  1001          CMPZV   #RXD_V_ID,#RXD_S_ID,R1,#3 ; ID = logical console data?
                     EF 12    0580  1002          BNEQ    50$                     ; If not, throw data away and try again.
                               0582  1003 ;       CMPB    R1,R0                   ; Code = requested data?
                               0582  1004 ;       BNEQ    CONSOLE_ERROR           ; No recovery from protocol error.
                  83 51 90     0582  1005          MOVB    R1,(R3)+                ; Put data byte in user's buffer.
                     E9 52 F5  0585  1006          SOBGTR  R2,50$                  ; Branch back to get another byte.
```

```
                     0588  1007
                     0588  1008 ;
                     0588  1009 ; Restore console IPRs and registers, and return.
                     0588  1010 ;
    51   22   DB     0588  1011 60$:     MFPR    #PR$_TXCS,R1            ; Read transmit status.
F9  51   07   E1     058B  1012         BBC     #TXS_V_RDY,R1,60$       ; Wait for ready.
    22   8E   DA     058F  1013         MTPR    (SP)+,#PR$_TXCS         ; Restore previous state of TXCS.
    20   8E   DA     0592  1014         MTPR    (SP)+,#PR$_RXCS         ; Restore previous state of RXCS.
         0F   BA     0595  1015         POPR    #^M<R0,R1,R2,R3>        ; Restore registers.
              05     0597  1016         RSB
                     0598  1017
                     0598  1018 ;
                     0598  1019 ; Come here to reboot CPU.
                     0598  1020 ;
    23   50   DA     0598  1021 90$:     MTPR    R0,#PR$_TXDB            ; Send reboot command.
              00     059B  1022         HALT                            ; Halt to let reboot occur.
```

```
                                059C  1024          .SBTTL  "ALLOCATE" CONSOLE TERMINAL
                                059C  1025
                                059C  1026 ;++
                                059C  1027 ; CON$OWNCTY - "ALLOCATE" CONSOLE TERMINAL
                                059C  1028 ;
                                059C  1029 ; FUNCTIONAL DESCRIPTION:
                                059C  1030 ;
                                059C  1031 ;       THIS ROUTINE SHOULD BE CALLED WHEN PERFORMING NON-INTERRUPT DRIVEN
                                059C  1032 ;       I/O TO THE CONSOLE TERMINAL.  IT DISABLES INTERRUPTS AND DOES ANY
                                059C  1033 ;       CPU-SPECIFIC INITIALIZATION OF THE CONSOLE TERMINAL REGISTERS.
                                059C  1034 ;       THE CONSOLE TERMINAL IS RESTORED TO ITS PREVIOUS STATE BY CALLING
                                059C  1035 ;       CON$RELEASECTY.
                                059C  1036 ;
                                059C  1037 ;       THIS ROUTINE SHOULD BE CALLED AT OR ABOVE IPL 20.
                                059C  1038 ;
                                059C  1039 ;       *** NOTE ***    THE CALLER IS RESPONSIBLE FOR RESTORING THE STATE
                                059C  1040 ;                       OF THE CONSOLE TERMINAL STATUS REGISTERS (RXCS AND
                                059C  1041 ;                       TXCS) BY CALLING CON$RELEASECTY WITH THE VALUES
                                059C  1042 ;                       RETURNED BY THIS ROUTINE.
                                059C  1043 ;
                                059C  1044 ; INPUTS:
                                059C  1045 ;       NONE
                                059C  1046 ;
                                059C  1047 ; OUTPUTS:
                                059C  1048 ;
                                059C  1049 ;       R0:     VALUE TO BE RESTORED TO TXCS WHEN CALLING CON$RELEASECTY
                                059C  1050 ;       R1:     VALUE TO BE RESTORED TO RXCS WHEN CALLING CON$RELEASECTY
                                059C  1051 ;
                                059C  1052 ;       PR$_RXCS AND PR$_TXCS ARE SET UP SO THAT NON-INTERRUPT I/O CAN BE
                                059C  1053 ;       PERFORMED TO THE CONSOLE TERMINAL.
                                059C  1054 ;
                                059C  1055 ;       11/780, 11/750, AND 11/730:
                                059C  1056 ;               CONSOLE INTERRUPTS ARE DISABLED
                                059C  1057 ;
                                059C  1058 ;       11/790:
                                059C  1059 ;               CONSOLE TRANSMIT AND RECEIVE MASKS ARE SET UP SO THAT ONLY
                                059C  1060 ;               I/O TO THE CONSOLE TERMINAL IS PERMITTED.  INTERRUPTS ARE
                                059C  1061 ;               DISABLED.
                                059C  1062 ;--
                                059C  1063 CON$OWNCTY::
      50    22    DB            059C  1064 10$:    MFPR    #PR$_TXCS,R0            ; GET VALUE TO BE RESTORED TO TXCS.
   F9 50    07    E1            059F  1065         BBC     #TXS_V_RDY,R0,10$       ; WAIT FOR VALID COPY
FF00FFBF    8F    CA            05A3  1066         BICL    #^C<TXS_M_TEM!TXS_M_IE>,- ; ONLY SAVE WRITABLE FIELDS OF TXCS
          50                    05A9  1067                 R0                      ; (TRANSMIT ENABLE MASK AND INT ENABLE).
50  00008000    8F    C8        05AA  1068         BISL    #TXS_M_WRTENA,R0        ; SET WRITE ENABLE BIT FOR MASK.
    00018000    8F    DA        05B1  1069         MTPR    #TXS_M_CONSTERM!TXS_M_WRTENA,-  ; ENABLE LOCAL TERM OUTPUT AND
          22                    05B7  1070                 #PR$_TXCS               ; DISABLE TRANSMIT INTERRUPTS.
20  00010000    8F    DA        05B8  1071         MTPR    #RXS_M_CONSTERM,#PR$_RXCS ; ENABLE LOCAL TERM INPUT AND DISABLE
                                05BF  1072                                         ; RECEIVE INTERRUPTS.
                      05        05BF  1073         RSB
```

```
                     05C0  1075          .SBTTL  RELEASE CONSOLE TERMINAL
                     05C0  1076 ;++
                     05C0  1077 ; CON$RELEASECTY - RELEASE CONSOLE TERMINAL
                     05C0  1078 ;
                     05C0  1079 ; FUNCTIONAL DESCRIPTION:
                     05C0  1080 ;
                     05C0  1081 ;       THIS ROUTINE SHOULD BE CALLED TO RELEASE THE CONSOLE TERMINAL AFTER
                     05C0  1082 ;       ALLOCATING IT WITH ROUTINE CON$OWNCTY.  IT RESTORES THE STATE OF THE
                     05C0  1083 ;       CONSOLE TERMINAL INTERFACE REGISTERS.
                     05C0  1084 ;
                     05C0  1085 ;       THIS ROUTINE SHOULD BE CALLED AT OR ABOVE IPL 20.
                     05C0  1086 ;
                     05C0  1087 ; INPUTS:
                     05C0  1088 ;       R0:     VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO TXCS
                     05C0  1089 ;       R1:     VALUE RETURNED BY CON$OWNCTY TO BE RESTORED TO RXCS
                     05C0  1090 ;
                     05C0  1091 ; OUTPUTS:
                     05C0  1092 ;       NONE.
                     05C0  1093 ;
                     05C0  1094 ;--
                     05C0  1095 CON$RELEASECTY::
        52   DD      05C0  1096          PUSHL   R2                      ; SAVE A REGISTER.
    52  22   DB      05C2  1097 10$:     MFPR    #PR$_TXCS,R2            ; READ TRANSMIT STATUS.
F9  52  07   E1      05C5  1098          BBC     #TXS_V_RDY,R2,10$       ; WAIT TIL INTERFACE IS READY.
    22  50   DA      05C9  1099          MTPR    R0,#PR$_TXCS            ; RESTORE TXCS TO PREVIOUS STATE.
    20  51   DA      05CC  1100          MTPR    R1,#PR$_RXCS            ; RESTORE RXCS TO PREVIOUS STATE.
    52  8E   D0      05CF  1101          MOVL    (SP)+,R2                ; RESTORE THE REGISTER.
             05      05D2  1102          RSB
```

```
                     05D3  1104          .SBTTL  CON$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
                     05D3  1105 ;++
                     05D3  1106 ; CON$GETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
                     05D3  1107 ;
                     05D3  1108 ; FUNCTIONAL DESCRIPTION:
                     05D3  1109 ;
                     05D3  1110 ;       THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
                     05D3  1111 ;       DIRECTLY TO THE CONSOLE TERMINAL
                     05D3  1112 ;
                     05D3  1113 ; INPUTS:
                     05D3  1114 ;       None
                     05D3  1115 ;
                     05D3  1116 ; OUTPUTS:
                     05D3  1117 ;       R0 contains the character.
                     05D3  1118 ;
                     05D3  1119 ;--
           00000013  05D3  1120          control_s = 19                  ; control s (xoff)
           00000011  05D3  1121          control_q = 17                  ; control q (xon)
                     05D3  1122
                     05D3  1123 CON$GETCHAR::
     50    20   DB  05D3  1124 10$:      mfpr    #pr$_rxcs,r0            ;receiver ready?
 F9  50    07   E1  05D6  1125          bbc     #rxs_v_done,r0,10$      ;if clr, receiver not ready
     50    21   DB  05DA  1126          mfpr    #pr$_rxdb,r0            ;read input character
                05  05DD  1127          rsb                             ;return
```

```
                                05DE  1129          .SBTTL  CON$PUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL
                                05DE  1130 ;++
                                05DE  1131 ; CON$PUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL
                                05DE  1132 ;
                                05DE  1133 ; FUNCTIONAL DESCRIPTION:
                                05DE  1134 ;
                                05DE  1135 ;       THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
                                05DE  1136 ;       DIRECTLY TO THE CONSOLE TERMINAL
                                05DE  1137 ;
                                05DE  1138 ; INPUTS:
                                05DE  1139 ;       R0 - Character to be output
                                05DE  1140 ;
                                05DE  1141 ; OUTPUTS:
                                05DE  1142 ;       Character written to the console terminal.
                                05DE  1143 ;
                                05DE  1144 ;--
                                05DE  1145 CON$PUTCHAR::
                          51    DD  05DE  1146          pushl   r1                      ;save a register
                      51  20    DB  05E0  1147 10$:     mfpr    #pr$_rxcs,r1            ;receiver ready?
                   1B 51  07    E1  05E3  1148          bbc     #rxs_v_done,r1,30$      ;if clr, receiver not ready
                      51  21    DB  05E7  1149          mfpr    #pr$_rxdb,r1            ;read input character.
                13    51  07    ED  05EA  1150          cmpzv   #0,#7,r1,#control_s     ;control-s?
                          11    12  05EF  1151          bneq    30$                     ;if neq no
                      51  20    DB  05F1  1152 20$:     mfpr    #pr$_rxcs,r1            ;receiver ready?
                   F9 51  07    E1  05F4  1153          bbc     #rxs_v_done,r1,20$      ;if clr, receiver not ready
                      51  21    DB  05F8  1154          mfpr    #pr$_rxdb,r1            ;read input character
                11    51  07    ED  05FB  1155          cmpzv   #0,#7,r1,#control_q     ;is it a control-q?
                          EF    12  0600  1156          bneq    20$                     ;no, wait for another character.
                      51  22    DB  0602  1157 30$:     mfpr    #pr$_txcs,r1            ;transmitter done?
                   F9 51  07    E1  0605  1158          bbc     #txs_v_rdy,r1,30$       ;if clr, transmitter not done
                      23  50    DA  0609  1159          mtpr    r0,#pr$_txdb            ;write output character
                          51 8ED0  060C  1160          popl    r1                      ;restore a register
                             05    060F  1161          rsb                             ;return
                                   0610  1162
                                   0610  1163          .END
```

```
BROADCAST                       000002C0 R      02
BR_DISMIS                       000003AB R      02
CLASS_DDT                     = 00000010
CLASS_DS_TRAN                 = 0000000C
CLASS_GETNXT                  = 00000000
CLASS_POWERFAIL               = 00000020
CLASS_PUTNXT                  = 00000004
CLASS_SETUP_UCB               = 00000008
CON$ABORT                       000004C7 RG     02
CON$C_BOOTCPU                 = 00000002
CON$DISCONNECT                  00000139 RG     02
CON$DS_SET                      00000149 RG     02
CON$GETCHAR                     000005D3 RG     02
CON$INITIAL                     00000040 RG     02
CON$INITLINE                    00000096 RG     02
CON$INIT_CTY                    00000138 RG     02
CON$INPDISTAB                   00000000 R      02
CON$INTINP                      00000185 RG     02
CON$INTOUT                      000003E1 RG     02
CON$NULL                        00000138 RG     02
CON$OWNCTY                      0000059C RG     02
CON$PUTCHAR                     000005DE RG     02
CON$RELEASECTY                  000005C0 RG     02
CON$RESUME                      000004E1 RG     02
CON$SENDCONSCMD                 00000530 RG     02
CON$SET_LINE                    00000138 RG     02
CON$SET_MODEM                   00000138 RG     02
CON$STARTIO                     0000032F RG     02
CON$STOP                        000004BF RG     02
CON$XOFF                        000004AA RG     02
CON$XON                         000004AA RG     02
CONTROL_Q                     = 00000011
CONTROL_S                     = 00000013
CON_BURST                       00000467 R      02
CON_CHAR                        00000483 R      02
CON_PREMPT                      0000049B R      02
CON_RESET_IE                    00000441 R      02
CON_RETURN                      00000095 R      02
CON_START_BURST                 00000460 R      02
CON_STOP                        00000495 R      02
CRB$B_DZ_CARRIER              = 0000001D
CRB$B_DZ_DTR                  = 0000001E
CRB$L_AUXSTRUC                = 00000010
CRB$L_INTD                    = 00000024
DDB$L_DDT                     = 0000000C
DEV$V_TRM                     = 00000002
DISMIS                          00000402 R      02
DISMIS_EMM                      000002E4 R      02
DISMIS_LOG                      0000032C R      02
DPT$W_VECTOR                  = 0000001E
EMB$C_CRBT                    = 00000011
EMB$C_EMM                     = 0000000F
EMB$C_HD_LENGTH               = 00000010
EMB$W_HD_ENTRY                = 00000004
EMM_ALERT_DONE                  000002DF R      02
EMM_BUFFER                      0000027D R      02
EMM_FIRST_BYTE                  0000028B R      02
EMM_FLAGS                       00000280 R      02
EMM_IGNORECNT                   0000027F R      02
EMM_IGNORE_DATA                 000002F3 R      02
EMM_IN_PROGRESS               = 00000000
EMM_LAST_BYTE                   000002A1 R      02
EMM_MESSAGE                     00001B9 R       02
EMM_MSGLEN                      0000027B R      02
EMM_RESPONSE                    000002E7 R      02
EMM_S_CONDITION               = 00000005
EMM_V_ALERT                   = 00000007
EMM_V_ASD                     = 00000006
EMM_V_BYTECOUNT               = 00000003
EMM_V_CONDITION               = 00000000
EMM_V_IGNOREINP               = 00000002
EMM_V_SHUTDOWN                = 00000001
ERL$ALLOCEMB                    ********    X   02
ERL$RELEASEMB                   ********    X   02
EXE$GL_ABSTIM                   ********    X   02
IDB$B_TT_ENABLE               = 0000000E
IDB$L_UCBLST                  = 00000018
INTOUT_TERM                     0000040C R      02
INT_CARCHANGE                   0000037B R      02
INT_CONSOLINP                   000001AA R      02
INT_EMMINP                      00000281 R      02
INT_IGNOREINP                   00000402 R      02
INT_LOGINP                      0000030C R      02
IOC$BROADCAST                   ********    X   02
MODEM$C_DATASET               = 00000003
MODEM$C_INIT                  = 00000000
MODEM$C_SHUTDWN               = 00000001
OP$DPT                          ********    X   02
OPA$UCB0                        ********    X   02
PR$_IPL                       = 00000012
PR$_RXCS                      = 00000020
PR$_RXDB                      = 00000021
PR$_TXCS                      = 00000022
PR$_TXDB                      = 00000023
REPORT_CARCHANGE                000003AE R      02
RXD                             00000000
RXDB$C_LOG_CLDFLG             = 00000011
RXDB$C_LOG_CMDCMP             = 00000020
RXDB$C_LOG_CMDERR             = 00000082
RXDB$C_LOG_REBOOT             = 00000040
RXDB$C_LOG_SNAP               = 00000030
RXDB$C_LOG_UCODE              = 00000012
RXDB$C_LOG_WRMFLG             = 00000010
RXD_S_ID                      = 00000004
RXD_V_CARRIER                 = 00000010
RXD_V_ID                      = 00000008
RXS                             00000000
RXS_M_CONSTERM                = 00010000
RXS_M_EMM                     = 00040000
RXS_M_IE                      = 00000040
RXS_M_LOGCONS                 = 00080000
RXS_V_DONE                    = 00000007
RXS_V_DTR                     = 00000010
SHUTDOWN_MESSAGE                00000235 R      02
```

```
SHUTDOWN_MSGLEN                 00000279 R      02
SIZ...                        = 00000008
START_TERM_IO                   00000338 R      02
TT$M_DS_CARRIER               = 00000020
TT$M_DS_CTS                   = 00000010
TT$M_DS_DSR                   = 00000080
TT$M_DS_RING                  = 00000040
TT$V_DS_CARRIER               = 00000005
TT$V_DS_DTR                   = 00000001
TT$V_MODEM                    = 00000015
TTY$GL_DPT                      ********    X   02
TTY$M_TANK_BURST              = 00000800
TTY$M_TANK_HOLD               = 00000400
TTY$M_TANK_PREMPT             = 00000100
TTY$M_TANK_STOP               = 00000200
TTY$V_PC_NOTIME               = 00000000
TTY$V_TANK_BURST              = 0000000B
TXDB$C_EMM_CANCEL             = 00000011
TXDB$C_EMM_ENV                = 00000001
TXDB$C_EMM_MARGIN             = 00000010
TXDB$C_EMM_STATUS             = 00000000
TXDB$C_LOG_CANCEL             = 00000070
TXDB$C_LOG_EXCOLD             = 00000011
TXDB$C_LOG_EXUCODE            = 00000012
TXDB$C_LOG_EXWARM             = 00000010
TXDB$C_LOG_INVSNP1            = 00000031
TXDB$C_LOG_INVSNP2            = 00000032
TXDB$C_LOG_REQERL             = 00000030
TXDB$C_LOG_SNDDIAG            = 00000020
TXS                             00000000
TXS_M_CONSTERM                = 00010000
TXS_M_IE                      = 00000040
TXS_M_LOGCONS                 = 00080000
TXS_M_TEM                     = 00FF0000
TXS_M_WRTENA                  = 00008000
TXS_S_ID                      = 00000004
TXS_V_ID                      = 00000008
TXS_V_RDY                     = 00000007
TXS_V_TEM                     = 00000010
UCB$B_TT_DS_RCV               = 00000124
UCB$B_TT_DS_TX                = 00000125
UCB$B_TT_OUTYPE               = 0000010B
UCB$B_TT_PREMPT               = 0000010A
UCB$L_CRB                     = 00000024
UCB$L_DDB                     = 00000028
UCB$L_DDT                     = 00000088
UCB$L_DEVCHAR                 = 00000038
UCB$L_DEVDEPEND               = 00000044
UCB$L_DUETIM                  = 0000006C
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_ONLINE                  = 00000010
UCB$M_TIM                     = 00000001
UCB$V_INT                     = 00000001
UCB$V_POWER                   = 00000005
UCB$W_REFC                    = 0000005C
UCB$W_STS                     = 00000064
UCB$W_TT_HOLD                 = 00000108
UCB$W_TT_OUTLEN               = 00000120
UCB$W_TT_PRTCTL               = 00000122
UCB$W_TT_UNITBIT              = 00000106
UCB$W_UNIT                    = 00000054
VEC$L_IDB                     = 00000008
```

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	00000004 (4.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000610 (1552.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:03.38
Command processing	114	00:00:00.45	00:00:03.01
Pass 1	542	00:00:14.97	00:01:10.61
Symbol table sort	0	00:00:02.30	00:00:08.14
Pass 2	211	00:00:03.19	00:00:12.01
Symbol table output	22	00:00:00.14	00:00:00.94
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	923	00:00:21.16	00:01:38.11

The working set limit was 1800 pages.
123886 bytes (242 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2143 non-local and 57 local symbols.
1163 source lines were read in Pass 1, producing 17 object records in Pass 2.
64 pages of virtual memory were used to define 61 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYSLOA.OBJ]790DEF.MLB;1	0
_255DUA28:[SYS.OBJ]LIB.MLB;1	28
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	37

2479 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:OPDRV790/OBJ=OBJ$:OPDRV790 MSRC$:OPDRV790/UPDATE=(ENH$:OPDRV790)+EXECML$/LIB+LIB$:790DEF/LIB
```

MOUNTVER
LIS

OPDRVWS1
LIS

QUORUM
LIS

OPDRV790
LIS

OPDRIVER
CIS

REBLDLOCK
LIS