H 6

```
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD     RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRkR
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD     RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
TTTTTTTTTTTTTTT  TTTTTTTTTTTTTTT  DDDDDDDDDDDD     RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRP
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD  RRR         RRR  VVV         VVV  RRR         RRR
      TTT              TTT        DDD         DDD  RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD  RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD  RRRRRRRRRRRR     VVV         VVV  RRRRRRRRRRRR
      TTT              TTT        DDD         DDD  RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD  RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD  RRR   RRR        VVV         VVV  RRR   RRR
      TTT              TTT        DDD         DDD  RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDD         DDD  RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDD         DDD  RRR      RRR        VVV   VVV     RRR      RRR
      TTT              TTT        DDDDDDDDDDDD     RRR         RRR        VVV        RRR         RRR
      TTT              TTT        DDDDDDDDDDDD     RRR         RRR        VVV        RRR         RRR
      TTT              TTT        DDDDDDDDDDDD     RRR         RRR        VVV        RRR         RRR
```

FILEID**YFDRIVER

J 5

```
YY       YY  FFFFFFFFFF  DDDDDDDD    RRRRRRRR     IIIIII   VV      VV  EEEEEEEEEE  RRRRRRRR
YY       YY  FFFFFFFFFF  DDDDDDDD    RRRRRRRR     IIIIII   VV      VV  EEEEEEEEEE  RRRRRRRR
YY       YY  FF          DD      DD  RR      RR     II     VV      VV  EE          RR      RR
YY       YY  FF          DD      DD  RR      RR     II     VV      VV  EE          RR      RR
  YY   YY    FF          DD      DD  RR      RR     II     VV      VV  EE          RR      RR
  YY   YY    FF          DD      DD  RR      RR     II     VV      VV  EE          RR      RR
    YY       FFFFFFFF    DD      DD  RRRRRRRR       II     VV      VV  EEEEEEEE    RRRRRRRR
    YY       FFFFFFFF    DD      DD  RRRRRRRR       II     VV      VV  EEEEEEEE    RRRRRRRR
    YY       FF          DD      DD  RR  RR         II     VV      VV  EE          RR  RR
    YY       FF          DD      DD  RR  RR         II     VV      VV  EE          RR  RR
    YY       FF          DD      DD  RR    RR       II       VV  VV    EE          RR    RR
    YY       FF          DD      DD  RR    RR       II       VV  VV    EE          RR    RR      ....
    YY       FF          DDDDDDDD    RR      RR   IIIIII       VV      EEEEEEEEEE  RR      RR    ....
    YY       FF          DDDDDDDD    RR      RR   IIIIII       VV      EEEEEEEEEE  RR      RR    ....
                                                                                                 ....

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

```
    (1)       62        DECLARATIONS
    (1)      200        REGISTER DEFINITIONS
    (1)      332        SPEED CONVERSION TABLES
    (1)      390        CONTROLLER INITIALIZATION
    (1)      440        CONTROLLER INITIALIZATION
    (1)      516        UNIT INITIALIZATION
    (1)      668        MAINTENANCE ROUTINES
    (1)      742        OUTPUT MODEM CONTROL
    (1)      783        RECEIVER INTERRUPT SERVICE
    (1)      900        START I/O ROUTINE
    (1)      994        PORT DMA ROUTINES
    (1)     1290        PORT ROUTINES  STOP,RESUME,XON,XOFF
    (1)     1437        OUTPUT INTERRUPT SERVICE
    (1)     1653        SET SPEED, PARITY PARAMETERS
```

YFD
V04

```
0000          .TITLE  YFDRIVER - Port Driver for DHU/DHV
0000          .IDENT  'V04-000'
0000    ;
0000    ;*************************************************************************
0000    ;*                                                                       *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.               *
0000    ;*  ALL RIGHTS RESERVED.                                                 *
0000    ;*                                                                       *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE  THIS SOFTWARE OR  ANY  OTHER *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000    ;*  TRANSFERRED.                                                         *
0000    ;*                                                                       *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000    ;*  CORPORATION.                                                         *
0000    ;*                                                                       *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.              *
0000    ;*                                                                       *
0000    ;*                                                                       *
0000    ;*************************************************************************
0000    ;
0000    ;++
0000    ; FACILITY:
0000    ;
0000    ;       VAX/VMS TERMINAL DRIVER
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       DHU/DHV ASYNC PORT DRIVER
0000    ;
0000    ; AUTHOR:
0000    ;
0000    ;       RICK SPITZ/ANDREW PALKA
0000    ;
0000    ; Revision history:
0000    ;
0000    ;       V03-004 LMP0275         L. Mark Pilant,         12-Jul-1984  21:05
0000    ;               Initialize the ACL info in the ORB to be a null descriptor
0000    ;               list rather than an empty queue.  This avoids the overhead
0000    ;               of locking and unlocking the ACL mutex, only to find out
0000    ;               that the ACL was empty.
0000    ;
0000    ;       V03-003 EMD0097         Ellen M. Dusseault      30-Apr-1984
0000    ;               Fix a few bugs - device timeout problem and test the
0000    ;               abort flag first to see if it is necessary to clear it.
0000    ;               Also add DEV$M_NNM characteristic to DEVCHAR2 so that
0000    ;               these devices will have the prefix "node$".
0000    ;
0000    ;       V03-002 TMH0002         Tim Halvorsen           14-Apr-1984
0000    ;               Fix references to UCB$L_OWNUIC and UCB$W_VPROT to use ORB.
0000    ;
0000    ;       V03-001 EMD0070         Ellen M. Dusseault              30-MAR-1984
```

```
  0000    58 ;              Modify to make code more efficient.
  0000    59 ;
  0000    60 ;--
```

```
    0000          .SBTTL  DECLARATIONS
    0000
    0000  ;
    0000  ; EXTERNAL DEFINITIONS:
    0000  ;
    0000          $PRDEF
    0000          $ACBDEF                                 ; DEFINE ACB
    0000          $CRBDEF                                 ; DEFINE CRB
    0000          $DCDEF                                  ; DEFINE ADAPTER TYPES
    0000          $DDBDEF                                 ; DEFINE DDB
    0000          $DEVDEF                                 ; DEFINE DEVICE TYPES
    0000          $DYNDEF                                 ; DEFINE DYNAMIC STRUCTURE TYPES
    0000          $IDBDEF                                 ; DEFINE IDB OFFSETS
    0000          $IODEF                                  ; DEFINE I/O FUNCTION CODES
    0000          $IRPDEF                                 ; DEFINE IRP
    0000          $ORBDEF                                 ; DEFINE OBJECT RIGHTS BLOCK
    0000          $SSDEF                                  ; DEFINE SYSTEM STATUS CODES
    0000          $TTYDEF                                 ; DEFINE TERMINAL DRIVER SYMBOLS
    0000          $TTDEF                                  ; DEFINE TERMINAL TYPES
    0000          $TT2DEF                                 ; DEFINE EXTENDED CHARACTERISTICS
    0000          $TQEDEF                                 ; DEFINE TIMER QUEUE OFFSETS
    0000          $UCBDEF                                 ; DEFINE UCB
    0000          $UBADEF                                 ; DEFINE UBA
    0000          $VECDEF                                 ; DEFINE VECTOR FOR CRB
    0000          $TTYMACS                                ; DEFINE TERMINAL DRIVER MACROS
    0000          $TTYDEFS                                ; DEFINE TERMINAL DRIVER SYMBOLS
    0000          $TTYMODEM                               ; DEFINE MODEM DEFINITIONS
    0000
    0000
    0000
    0000  ;
    0000  ; LOCAL STORAGE
    0000  ;
00000000          .PSECT  $$$105_PROLOGUE
    0000
    0000  ;
    0000  ; Driver prologue table:
    0000  ;
    0000
    0000  ;
    0000  ; Note. The DPT says that this driver has a UBA adapter type.
    0000  ; In fact it will work with a Q-bus VAX, either mapped or unmapped
    0000  ;
    0000  YF$DPT::                                        ; DRIVER START
    0000          DPTAB   -                               ; DRIVER PROLOGUE TABLE
    0000                  END=YF$END,-                    ; End and offset to INIT's vectors
    0000                  UCBSIZE=UCB$C_TT_LENGTH+4,-     ; SIZE OF UCB
    0000                  FLAGS=DPT$M_NOUNLOAD,-          ; NO UNLOAD ALLOWED
    0000                  ADAPTER=UBA,-                   ; ADAPTER TYPE
    0000                  MAXUNITS=16,-                   ; Number of units to create
    0000                  DEFUNITS=16,-                   ; Number of units to create
    0000                  NAME=YFDRIVER,-                 ; NAME OF DRIVER
    0000                  VECTOR=PORT_VECTOR,-            ; PORT DRIVER VECTOR TABLE
    0000                  DELIVER=YF$DELIVER              ; Unit delivery routine
    0038          DPT_STORE INIT
    0038          DPT_STORE UCB,UCB$B_FIPL,B,8            ; FORK IPL
    003C          DPT_STORE UCB,UCB$L_DEVCHAR,L,<-;       CHARACTERISTICS
```

```
                003C                     DEV$M_REC!-          ;
                003C                     DEV$M_AVL!-          ;
                003C                     DEV$M_IDV!-          ;
                003C                     DEV$M_ODV!-          ;
                003C                     DEV$M_TRM!-          ;
                003C                     DEV$M_CCL>
                0043          DPT_STORE UCB,UCB$L_DEVCHAR2,L,- ; DEVICE CHARACTERISTICS
                0043                     <DEV$M_NNM>          ; PREFIX WITH "NODE$"
                004A          DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_TERM;
                004E          DPT_STORE UCB,UCB$B_TT_DETYPE,B,TT$_UNKNOWN       ; TYPE
                0052          DPT_STORE UCB,UCB$W_TT_DESIZE,@W,TTY$GW_DEFBUF    ; BUFFER SIZE
                0059          DPT_STORE UCB,UCB$L_TT_DECHAR,@L,TTY$GL_DEFCHAR   ; DEFAULT CHARACTERS
                0060          DPT_STORE UCB,UCB$L_TT_DECHA1,@L,TTY$GL_DEFCHAR2; DEFAULT CHARACTERS
                0067          DPT_STORE UCB,UCB$W_TT_DESPEE,@B,TTY$GB_DEFSPEED; DEFAULT SPEED
                006E          DPT_STORE UCB,UCB$W_TT_DESPEE+1,@B,TTY$GB_RSPEED; DEFAULT RSPEED
                0075          DPT_STORE UCB,UCB$B_TT_DEPARI,@B,TTY$GB_PARITY    ; DEFAULT PARITY
                007C          DPT_STORE UCB,UCB$B_DEVTYPE,B,TT$_UNKNOWN         ; TYPE
                0080          DPT_STORE UCB,UCB$W_DEVBUFSIZ,@W,TTY$GW_DEFBUF    ; BUFFER SIZE
                0087          DPT_STORE UCB,UCB$L_DEVDEPEND,@L,TTY$GL_DEFCHAR   ; DEFAULT CHARACTERS
                008E          DPT_STORE UCB,UCB$L_DEVDEPND2,@L,TTY$GL_DEFCHAR2; DEFAULT CHARACTERS
                0095          DPT_STORE UCB,UCB$W_TT_SPEED,@B,TTY$GB_DEFSPEED  ; DEFAULT SPEED
                009C          DPT_STORE UCB,UCB$W_TT_SPEED+1,@B,TTY$GB_RSPEED  ; DEFAULT RSPEED
                00A3          DPT_STORE UCB,UCB$B_DIPL,B,21                     ; DEVICE IPL
                00A7          DPT_STORE UCB,UCB$L_TT_WFLINK,L,0                 ; Zero write queue.
                00AE          DPT_STORE UCB,UCB$L_TT_WBLINK,L,0                 ; Zero write queue.
                00B5          DPT_STORE UCB,UCB$L_TT_RTIMOU,L,0                 ; Zero read timed out disp.
                00BC          DPT_STORE ORB,ORB$B_FLAGS,B,-                     ; Protection block flags
                00BC                   <ORB$M_PROT_16>                         ; SOGW protection word
                00C0          DPT_STORE ORB,ORB$W_PROT,@W,TTY$GW_PROT           ; Default allocation protect
                00C7          DPT_STORE ORB,ORB$L_OWNER,@L,TTY$GL_OWNUIC        ; Default owner UIC
                00CE          DPT_STORE DDB,DDB$L_DDT,D,YF$DDT
                00D3          DPT_STORE REINIT
                00D3          DPT_STORE CRB,CRB$L_INTD+4,D,YF$INTINP  ; RECEIVER INTERRUPT
                00D8          DPT_STORE CRB,CRB$L_INTD2+4,D,YF$INTOUT ; TRANSMITTER INTERRUPT
                00DD          DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,D,YF$INITIAL  ; CONTROLLER INIT
                00E2          DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,YF$INITLINE; UNIT INIT
                00E7
                00E7          DPT_STORE END
                0000
                0000          DDTAB   DEVNAM = YF,-    ; DUMMY DHU PORT DRIVER DISPATCH TABLE
                0000                  START  = 0,-
                0000                  FUNCTB = 0
            00000038          .PSECT  $$$115_DRIVER,LONG
                0038
                0038      ;
                0038      ; THE ASSOCIATED CLASS DRIVER USES THIS TABLE TO COMMAND THE PORT DRIVER.
                0038      ; THE ADDRESS OF THIS TABLE IS CONTAINED IN THE TERMINAL UCB EXTENSION AREA.
                0038      ; THE OFFSET DEFINITONS ARE DEFINED BY TTYDEFS.
                0038      ;
                0038      YF$L_SIL_ERROR::
     00000000   0038              .LONG   0                       ; Indicates Silo not empty at interrupt
                003C      YF$L_ERROR::
     00000000   003C              .LONG   0                       ; Indicates DMA count non zero at
                0040                                              ; interrupt
                0040      YF$L_DMAXMT_ERROR::
     00000000   0040              .LONG   0                       ; Indicates DMA error bit set by DHU
                0044      YF$L_INACT_ERROR::
                 00000000  0044          .LONG   0                         ; Indicates UCB$M_INT clear at interrupt
                           0048  PORT_VECTOR:
                           0048
                           0048  ;
                           0048  ; DHU/DHV SPECIFIC DISPATCH TABLE
                           0048  ;
                           0048          $VECINI YF,YF$NULL
                           0080          $VEC    STARTIO,YF$STARTIO              ; START NEW OUTPUT
                           004C          $VEC    SET_LINE,YF$SET_LINE            ; SET NEW SPEED/PARITY
                           0054          $VEC    DS_SET,YF$DS_SET                ; SET OUTPUT MODEM SIGNALS
                           0058          $VEC    XON,YF$XON                      ; SEND XON SEQUENCE
                           005C          $VEC    XOFF,YF$XOFF                    ; SEND XOFF SEQUENCE
                           0060          $VEC    STOP,YF$STOP                    ; STOP OUTPUT
                           0064          $VEC    ABORT,YF$ABORT                  ; ABORT OUTPUT IN PROGRESS
                           006C          $VEC    RESUME,YF$RESUME                ; RESUME STOPPED OUTPUT
                           0070          $VEC    MAINT,YF$MAINT                  ; INVOKE MAINTENANCE FUNCTION
                           007C          $VEC    FORKRET,YF$FORK                 ; PORT FORK CALLBACK
                           0080          $VECEND
                           0084
                           0084  ;
                           0084  YF$NULL:                                        ; NULL PORT ROUTINE
                       05  0084          RSB
                           0085
                 0085           .SBTTL  REGISTER DEFINITIONS
                 0085
                 0085
       00000000  0085   DHUCSR  =       0                               ; Base CSR
       00000002  0085   DHURBF  =       2                               ; Received data fifo
       00000002  0085   DHUTXC  =       2                               ; DHV single char register
       00000002  0085   DHUTCR  =       2                               ; DHU timer control
       00000004  0085   DHULPR  =       4                               ; Line parameters
       00000006  0085   DHUTXF  =       6                               ; Transmit FIFO
       00000006  0085   DHUTFS  =       6                               ; Transmit FIFO size
       00000007  0085   DHUSTT  =       7                               ; Modem status
       00000008  0085   DHULCT  =       8                               ; Line control
       0000000A  0085   DHUTBF1 =       10                              ; low bits of address
       0000000C  0085   DHUTBF2 =       12                              ; High bits of address
       0000000E  0085   DHUTCT  =       14                              ; Byte count
                 0085
                 0085   ;
                 0085   ; CSR BIT DEFINITIONS ( CSR ) ( READ/WRITE )
                 0085   ; (NOTE: THIS REGISTER MUST ONLY BE READ IN RESPONSE TO A TRANSMIT INTERRUPT.
                 0085   ;       TO LOAD AN INDIRECT REGISTER, ONLY WRITE OPERATIONS MAY BE USED.
                 0085   ;       READ-MODIFY-WRITE OPERATIONS MUST NEVER BE USED)
                 0085
                 0085           $VIELD  DHUCSR,0,<-
                 0085                   <IADDR,4,M>,-   ; INDIRECT REGISTER ADDRESS
                 0085                   <,1,>,-         ; DON'T USE THIS BIT
                 0085                   <CLEAR,1,M>,-   ; MASTER RESET
                 0085                   <RCVINT,1,M>,-  ; RECEIVER INTERRUPT ENABLE
                 0085                   <,1,>,-         ; DON'T USE THIS BIT
                 0085                   <LINE,4,M>,-    ; LINE NUMBER (0 - 15)
                 0085                   <DMAERR,1,M>,-  ; DMA error
                 0085                   <DIGFAL,1,M>,-  ; Diagnostic failed
                 0085                   <SNDINT,1,M>,-  ; TRANSMIT INTERRUPT ENABLE
                 0085                   <SNDRDY,1,M>-   ; TRANSMITTER READY
                 0085                   >
                 0085   ;
                 0085   ; RECEIVER BUFFER ( CSR+2 ) ( READ ONLY )
                 0085   ;
                 0085           $VIELD  DHURCV,0,<-
                 0085                   <BUF,8,M>,-     ; RECEIVER DATA
                 0085                   <LINE,4,M>,-    ; LINE NUMBER (0 - 7)
                 0085                   <PARERR,1,M>,-  ; PARITY ERROR
                 0085                   <FRAMER,1,M>,-  ; FRAME ERROR
                 0085                   <OVERRUN,1,M>,- ; OVERRUN ERROR
                 0085                   <VALID,1,M>-    ; DATA VALID
                 0085                   >
                 0085   ;
                 0085   ; LINE PARAMETER REGISTER ( CSR+4 , Indirect register)
                 0085   ;
                 0085
                 0085           $VIELD  DHULPR,0,<-
                 0085                   <,1,>,-         ; DON'T USE THIS BIT
                 0085                   <DIAG1,1,M>,-   ; Diagnostics control
                 0085                   <DIAG2,1,M>,-   ; Diagnostics control
                 0085                   <SIZE,2,M>,-    ; CHARACTER SIZE
                 0085                   <PARITY,1,M>,-  ; PARITY ENABLE
                 0085                   <ODD,1,M>,-     ; ODD PARITY
                 0085                   <STOP,1,M>,-    ; NUMBER STOP BITS
                 0085                      <RSPEED,4,M>,-  ; RECEIVER LINE SPEED
                 0085                      <TSPEED,4,M>,-  ; TRANSMITTER LINE SPEED (BOTH RVC/TX FOR LINES 2-7)
                 0085                      >
                 0085   ;
                 0085   ; LINE CONTROL INDIRECT REGISTER (CSR + 8, Indirect register)
                 0085   ;
                 0085
                 0085           $VIELD     DHULCT,0,<-
                 0085                      <ABORT,1,M>,-    ; Output abort
                 0085                      <OAUTO,1,M>,-    ; AUTO XON/OFF
                 0085                      <RCV,1,M>,-      ; RECEIVER ENABLE
                 0085                      <BREAK,1,M>,-    ; SEND BREAK
                 0085                      <IAUTO,1,M>,-    ; Incoming Auto flow enable
                 0085                      <SNDOFF,1,M>,-   ; Send XOFF
                 0085                      <MAINT,2,M>,-    ; Maintenance
                 0085                      <MODEM,1,M>,-    ; Modem control line
                 0085                      <DTR,1,M>,-      ; Data terminal ready
                 0085                      <,2,>,-
                 0085                      <RTS,1,M>,-      ; Request to send
                 0085                      >
                 0085   ;
                 0085   ; Status register
                 0085           $VIELD     DHUSTT,0,<-
                 0085                      <,3,>,-
                 0085                      <CTS,1,M>,-
                 0085                      <DCD,1,M>,-
                 0085                      <RI,1,M>,-
                 0085                      <,1,>,-
                 0085                      <DSR,1,M>,-
                 0085                      >
                 0085
                 0085   ; Constant value for Base CSR
                 0085   ; Enables both receive and transmit interrrupts
                 0085
        00004040 0085   DHUCSR$C_BASE = <DHUCSR$M_RCVINT!DHUCSR$M_SNDINT>
                 0085   ;
                 0085   ; MACRO USED TO ACCESS INDIRECT REGISTERS
                 0085   ;
                 0085
                 0085           .MACRO     SETIND  REG
                 0085           .IF        B,      REG
                 0085           MOVL       UCB$L_CRB(R5),R0                      ; GET CRB ADDRESS
                 0085           MOVL       @CRB$L_INTD+VEC$L_IDB(R0),R0          ; GET CSR ADDRESS
                 0085           BISW3      #<DHUCSR$C_BASE>,-
                 0085                                UCB$W_UNIT(R5),DHUCSR(R0); SELECT INDIRECT FIELD
                 0085           .IFF
                 0085           BISW3      #<DHUCSR$C_BASE>,-
                 0085                                UCB$W_UNIT(R5),DHUCSR(REG); SELECT INDIRECT FIELD
                 0085           .ENDC
                 0085           .ENDM      SETIND
                 0085
                 0085   ;
                 0085   ; Extra field in UCB
                 0085   ; This field could be removed if 2 spare bits can be found in the UCB for
                 0085   ; the use of the port driver. (This would save 4 bytes from the declared length
                 0085   ; of the UCB)
                 0085
```

```
00000134  0085   314 UCB$B_DHUFLG     = UCB$C_TT_LENGTH         ; Allocate at end of previous ucb
          0085   315
          0085   316             $VIELD  UCB,0,<-
          0085   317                     <MAP,1,M>,-     ; Map registers available if 1
          0085   318                     <DHU,1,M>,-     ; DHU if 1, DHV if 0
          0085   319                     <XOFF,1,M>,-    ; XOFF if 1, XON if 0
          0085   320                     >
          0085   321
          0085   322 ;           OUTPUT DELAY MACRO
          0085   323 ;           TO RELEASE THE UNIBUS
          0085   324
          0085   325             .MACRO  DELAY
          0085   326             .REPEAT 3
          0085   327             NOP
          0085   328             .ENDR
          0085   329             .ENDM
          0085   330
```

```
                  0085          .sbttl  SPEED CONVERSION TABLES
                  0085
                  0085    ;
                  0085    ; macro to generate table of acceptable speeds for DHU/DHV
                  0085    ;
                  0085          .MACRO  SPDCONV BAUD
                  0085    .=yf$vms_speeds+tt$c_'baud'
                  0085          .byte dhuspd$c_'baud'
                  0085    .=yf$dhu_speeds+dhuspd$c_'baud'
                  0085          .byte tt$c_'baud'
                  0085          .ENDM
                  0085
                  0085    ;
                  0085    ; speed values recognized by the DHU/DHV
                  0085    ;
        00000000  0085    DHUSPD$C_BAUD_50=0
        00000001  0085    DHUSPD$C_BAUD_75=1
        00000002  0085    DHUSPD$C_BAUD_110=2
        00000003  0085    DHUSPD$C_BAUD_134=3
        00000004  0085    DHUSPD$C_BAUD_150=4
        00000005  0085    DHUSPD$C_BAUD_300=5
        00000006  0085    DHUSPD$C_BAUD_600=6
        00000007  0085    DHUSPD$C_BAUD_1200=7
        00000008  0085    DHUSPD$C_BAUD_1800=8
        00000009  0085    DHUSPD$C_BAUD_2000=9
        0000000A  0085    DHUSPD$C_BAUD_2400=10
        0000000B  0085    DHUSPD$C_BAUD_4800=11
        0000000C  0085    DHUSPD$C_BAUD_7200=12
        0000000D  0085    DHUSPD$C_BAUD_9600=13
        0000000E  0085    DHUSPD$C_BAUD_19200=14
        0000000F  0085    DHUSPD$C_BAUD_38400=15
                  0085
                  0085    ; Allocate and initialize table of speed values
                  0085    YF$VMS_SPEEDS:
                  0085          .REPEAT 16
                  0085          .BYTE   -1                      ; Initial to illegal value
              FF  0085          .ENDR
                  0095    YF$DHU_SPEEDS:
                  0095          .REPEAT 16
                  0095          .BYTE   0                       ;  Initial to zero
              00  0095          .ENDR
                  00A5
                  00A5    ; Now build up the table of acceptable speed values
                  00A5
                  00A5          SPDCONV BAUD_75
                  0097          SPDCONV BAUD_110
                  0098          SPDCONV BAUD_134
                  0099          SPDCONV BAUD_150
                  009A          SPDCONV BAUD_300
                  009B          SPDCONV BAUD_600
                  009C          SPDCONV BAUD_1200
                  009D          SPDCONV BAUD_1800
                  009E          SPDCONV BAUD_2000
                  009F          SPDCONV BAUD_2400
                  00A0          SPDCONV BAUD_4800
                  00A1          SPDCONV BAUD_9600
                  00A3          SPDCONV BAUD_19200
```

```
                        00A4           .SBTTL  CONTROLLER INITIALIZATION
                        00A4
                        00A4
                        00A4
                        00A4     ;++
                        00A4     ; YF$DELIVER - Unit delivery routine
                        00A4     ;
                        00A4     ; FUNCTIONAL DESCRIPTION:
                        00A4     ;
                        00A4     ; THIS ROUTINE IS ENTERED AT SYSTEM STARTUP
                        00A4     ;
                        00A4     ; It checks the device to see if the specified unit number exists.
                        00A4     ; This is necessary as the DHV has 8 lines, while the DHU has 16
                        00A4     ;
                        00A4     ; INPUTS:
                        00A4     ;
                        00A4     ;         R4 = ADDRESS OF THE CONTROLLER CSR
                        00A4     ;         R5 = Unit number
                        00A4     ;
                        00A4     ; OUTPUTS:
                        00A4     ;
                        00A4     ;         R0 = 1 if unit exists
                        00A4     ;            = 0 if unit does not exist
                        00A4     ;
                        00A4     ; IMPLICIT INPUTS:
                        00A4     ;
                        00A4     ;
                        00A4     ;--
                        00A4     YF$DELIVER::
   10      55     D1    00A4           CMPL    R5,#16
           1C     18    00A7           BGEQ    10$                           ; exit if unit too large
   50      64     B0    00A9           MOVW    (R4),R0                       ; Read base CSR
15 50      05     E0    00AC           BBS     #DHUCSR$V_CLEAR,R0,10$        ; controller bad
11 50      0D     E0    00B0           BBS     #DHUCSR$V_DIGFAL,R0,10$       ; controller bad
                        00B4     ;
                        00B4     ; Note. If controller has not passed self test then we cannot rely on
                        00B4     ; the value read from DHUSTT
                        00B4
50      07 A4     90    00B4           MOVB    DHUSTT(R4),R0                 ; get status byte
        05 50     E8    00B8           BLBS    R0,5$                         ; low bit indicates DHU
   08      55     D1    00BB           CMPL    R5,#8                         ; DHV has only 8 lines
           05     18    00BE           BGEQ    10$
                        00C0     5$:
   50      01     D0    00C0           MOVL    #1,R0                         ; unit exists
           02     11    00C3           BRB     20$
                        00C5     10$:
           50     D4    00C5           CLRL    R0                            ; unit does not exist
                        00C7     20$:
                  05    00C7           RSB
```

```
                     00C8          .SBTTL  CONTROLLER INITIALIZATION
                     00C8
                     00C8
                     00C8
                     00C8    ;++
                     00C8    ; YF$INITIAL - INITIALIZE INTERFACE
                     00C8    ;
                     00C8    ; FUNCTIONAL DESCRIPTION:
                     00C8    ;
                     00C8    ; THIS ROUTINE IS ENTERED AT SYSTEM STARTUP AND POWER RECOVERY.
                     00C8    ;
                     00C8    ; INPUTS:
                     00C8    ;
                     00C8    ;         R4 = ADDRESS OF THE UNIT CSR
                     00C8    ;         R5 = IDB OF UNIT
                     00C8    ;         R8 = ADDRESS OF THE UNIT CRB
                     00C8    ;
                     00C8    ; OUTPUTS:
                     00C8    ;
                     00C8    ;         R2 is destroyed.
                     00C8    ;
                     00C8    ; IMPLICIT INPUTS:
                     00C8    ;
                     00C8    ;         IPL = IPL$_POWER
                     00C8    ;
                     00C8    ;--
                     00C8    YF$INITIAL::                               ; INITIALIZE DHU UNIT
                     00C8    ;
                     00C8    ; SET UP CONTROLLER
                     00C8    ;
                     00C8            CLASS_CTRL_INIT YF$DPT,PORT_VECTOR; RELOCATE THE NECESSARY TABLES
                     00F5    25$:
                     00F5    ;
                     00F5    ; Note. The DHV takes about 2 seconds to initialise
                     00F5    ; and the DHU up to 5
                     00F5    ; so we assume that initialization has taken place
                     00F5    ; We could start the initialization and then do a 'skip self test'
                     00F5    ; operation to bring the self test time down to a few milliseconds,
                     00F5    ; however we would not then know if the board was good or not.
                     00F5    ;
                     00F5
     64    20    B3  00F5            BITW    #DHUCSR$M_CLEAR,(R4)
           03    12  00F8            BNEQ    26$                        ; dont do reset if still there
     64    20    B0  00FA            MOVW    #DHUCSR$M_CLEAR,(R4)       ; CONTROLLER RESET
                     00FD    26$:
                     00FD    ;
                     00FD    ;       WAIT TILL CONTROLLER INITIALIZATION IS COMPLETE
                     00FD    ;
                     00FD    ;       5 second wait here !!!
                     00FD    ;
                     00FD            TIMEWAIT        #500000,#DHUCSR$M_CLEAR,(R4),W,.FALSE.
                     0124
                     0124
        2E 50    E9  0124            BLBC    R0,YF$CTRL_ERROR
 64   2000 8F    B3  0127            BITW    #DHUCSR$M_DIGFAL,(R4)
           05    13  012C            BEQL    90$
     50    00    B0  012E            MOVW    #0,R0                      ; failed self test, dont use
```

```
                  22  11  0131   497          BRB     YF$CTRL_ERROR
                          0133   498 90$:
       64     4040 8F  B0  0133   499          MOVW    #DHUCSR$C_BASE,(R4)      ; set up base csr value
                          0138   500
                          0138   501
                          0138   502 100$:
     0B A8      46 8F  90  0138   503          MOVB    #DT$_DHV,CRB$B_TT_TYPE(R8)       ; CONTROLLER IS DHV
        50      07 A4  90  013D   504          MOVB    DHUSTT(R4),R0                    ; test for DHU or DHV
                0D 50  E9  0141   505          BLBC    R0,110$                          ; DHV does not have silo timeout
  02 A4   00000000'GF  90  0144   506          MOVB    G^TTY$GB_SILOTIME,DHUTCR(R4)     ; INIT INPUT SILO TIMEOUT VALUE
     0B A8      47 8F  90  014C   507          MOVB    #DT$_DHU,CRB$B_TT_TYPE(R8)       ; CONTROLLER IS DHU
                          0151   508 110$:
           50      01  D0  0151   509          MOVL    #SS$_NORMAL,R0
                       05  0154   510          RSB
                          0155   511
                          0155   512 YF$CTRL_ERROR:
                       05  0155   513          RSB
                          0156   514
                          0156           .SBTTL  UNIT INITIALIZATION
                          0156
                          0156  ; YF$INITLINE - UNIT INITIALIZATION
                          0156  ;
                          0156  ; FUNCTIONAL DESCRIPTION:
                          0156  ;
                          0156  ; THIS ROUTINE PERFORMS A SIMPLE UNIT INITIALIZATION.
                          0156  ;
                          0156  ; INPUTS:
                          0156  ;
                          0156  ;        R5 = UCB ADDRESS
                          0156  ;
                          0156  ; OUTPUTS:
                          0156  ;
                          0156  ;        R4,R5 ARE PRESERVED.
                          0156  ;--
                          0156
                          0156  YF$INITLINE::
      50   FEEE CF   DE   0156           MOVAL   YF$VEC,R0                 ; GET THE DISPATCH TABLE ADDRESS
                          015B           CLASS_UNIT_INIT
        64 A5    10  A8   01A4           BISW    #UCB$M_ONLINE,UCB$W_STS(R5); SET ONLINE
                          01A8
    53   01    54 A5  78  01A8           ASHL    UCB$W_UNIT(R5),#1,R3      ; BUILD UNIT'S BIT MASK
      0106 C5    53   B0  01AD           MOVW    R3,UCB$W_TT_UNITBIT(R5)   ; SAVE IT
 00F8 C5   00000000'GF  90  01B2           MOVB    G^TTY$GB_PARITY,UCB$B_TT_PARITY(R5)
                          01BB                                             ; STORE TERMINAL'S PARITY VALUE IN UCB
                 24   A8  01BB           BISW    #TTY$M_PC_DMAAVL!TTY$M_PC_XOFAVL,-; SHOW DMA FEATURE AVAILABLE FOR U
           0122 C5        01BD                   UCB$W_TT_PRTCTL(R5)       ; IN PORT LEVEL
                          01C0
      51   0114 C5   D0   01C0           MOVL    UCB$L_TT_CLASS(R5),R1     ; ADDRESS CLASS VECTOR TABLE
             08 B1   16   01C5           JSB     @CLASS_SETUP_UCB(R1)      ; INIT UCB FIELDS
                          01C8
                          01C8  ;
                          01C8  ; Perform check to see if device is good
                          01C8  ;
                 30   BB  01C8           PUSHR   #^M<R4,R5>
        54    24 A5   D0  01CA           MOVL    UCB$L_CRB(R5),R4                           ; GET CRB ADDRESS
        54    2C B4   D0  01CE           MOVL    @CRB$[_INTD+VEC$L_IDB(R4),R4               ; GET CSR ADDRESS
        55    54 A5   3C  01D2           MOVZWL  UCB$W_ONIT(R5),R5
           FECA CF    16  01D6           JSB     YF$DE[IVER                           ; test this device is ok
                 30   BA  01DA           POPR    #^M<R4,R5>
             04 50    E8  01DC           BLBS    R0,11$
                          01DF  ;
                          01DF  ; DHU is in a bad way, set device offline
                          01DF  ;
        64 A5    10   AA  01DF           BICW    #UCB$M_ONLINE,UCB$W_STS(R5); SET OFFLINE
                          01E3  11$:
                          01E3  ; Test to see the type of configuration
                          01E3  ;
           0134 C5    94  01E3           CLRB    UCB$B_DHUFLG(R5)
                          01E7           SETIND
      0134 C5    02   88  01F6           BISB    #UCB$M_DHU,UCB$B_DHUFLG(R5); Assume DHU
        50    07 A0   90  01FB           MOVB    DHUSTT(R0),R0             ; test if DHU or DHV
             09 50    E8  01FF           BLBS    R0,12$
                          0202  ;
                          0202  ; Note. The DHV always interrupts at BR4
                          0202  ; This becomes IPL 20 on a VAX
                          0202   ;
        5E A5     14   90 0202           MOVB    #20,UCB$B_DIPL(R5)
      0134 C5     02   8A 0206           BICB    #UCB$M_DHU,UCB$B_DHUFLG(R5); Actually its a DHV
                          020B
                          020B   12$:
                          020B   ;
                          020B   ; Find out what kind of machine the device is connected to.
                          020B   ; This tells us what kind of mapping registers we have
                          020B   ;
                          020B           CPUDISP <-
                          020B                   YF_INITMAP,-            ; 11/780
                          020B                   YF_INITMAP,-            ; 11/750
                          020B                   YF_INITMAP,-            ; 11/730
                          020B                   YF_INITMAP,-            ; 11/790
                          020B                   YF_INITNULL,-           ; unknown
                          020B                   YF_INITNULL,-           ; unknown
                          020B                   YF_NOMAP,-              ; Seahorse
                          020B                   YF_INITMAP-             ; Mayflower
                          020B                   >
                          0227   ; Unknown processor type
                          0227   YF_INITNULL:
                          0227   ;
                          0227   ; prevent DMA being used, because we do not understand how the adapter
                          0227   ; works.
                          0227   ;
               04      AA 0227           BICW    #TTY$M_PC_DMAAVL,-; SHOW DMA FEATURE NOT AVAILABLE FOR USE
          0122 C5         0229                   UCB$W_TT_PRTCTL(R5)     ; IN PORT LEVEL
                          022C
               07      11 022C           BRB     INIT_CONTINUE
                          022E   ; These processors have map registers for DMA operation
                          022E   YF_INITMAP:
                          022E
      0134 C5     01   88 022E           BISB    #UCB$M_MAP,UCB$B_DHUFLG(R5)
                  00   11 0233           BRB     INIT_CONTINUE
                          0235
                          0235   ; These processors do not have map registers
                          0235
                          0235   YF_NOMAP:
                          0235   ; No bits set
                          0235
                          0235   INIT_CONTINUE:
                          0235   ;
                          0235   ;SET MODE CODE NEEDS TO TOGGLE THESE BITS
                          0235
        000009EB'EF    16 0235           JSB     YF$SET_LINE             ; INIT SPEED/PARITY
                          023B
                          023B   ;
                          023B   ; ENABLE LINE RECEIVER , TRANSMITTER AND MODEM INTERRUPTS
                          023B   ;
                          023B
     23 64 A5     04   E1 023B           BBC     #UCB$V_ONLINE,UCB$W_STS(R5),20$ ; TEST ONLINE
                          0240           SETIND  R4
           0D A4       94 0247           CLRB    DHUTBF2+1(R4)                   ; assume transmitter should
                          024A                                                   ; be disabled
        08 A4     01   A8 024A           BISW    #DHULCT$M_ABORT,DHULCT(R4)      ; assume abort to be set
                          024E
   0F 012A C5     07   E0 024E           BBS     #UCB$V_TT_DSBL,UCB$B_TT_MAINT(R5),20$   ; Test if disabled
```

```
                             0254
         08 A4   0104 8F   A8 0254          BISW    #DHULCT$M_RCV!DHULCT$M_MODEM,DHULCT(R4) ; enable receiver and modem
              08 A4    01   AA 025A          BICW    #DHULCT$M_ABORT,DHULCT(R4)        ; clear abort bit
         0D A4     80 8F   90 025E          MOVB    #^X80,DHUTBF2+1(R4)               ; enable transmitter
                             0263    20$:
                             0263    ;
                             0263    ; INIT RECEIVER MODEM STATUS FOR DHU
                             0263    ;
                             0263
              50    07 A4   90 0263          MOVB    DHUSTT(R4),R0
                             0267    ;
                             0267    ; Unfortunately the DHU/DHV dont put the modem bits where we want them
                             0267    ; To get the correct bits we have to move RI,DCD,CTS up one place, while not
                             0267    ; changing DSR. spare bits have to be ignored.
                 50    50   C0 0267          ADDL    R0,R0                       ; (This shifts DSR out of bottom byte)
              50    8F 8F   8A 026A          BICB    #^C<TT$M_DS_RING!TT$M_DS_CTS!TT$M_DS_CARRIER>,R0
            0124 C5    50   90 026E          MOVB    R0,UCB$B_TT_DS_RCV(R5)      ; UPDATE CURRENT INPUT MODEM SIGNALS
              06 50    08   E1 0273          BBC     #DHUSTT$V_DSR+T,R0,25$
         00 0124 C5    07   E2 0277          BBSS    #TT$V_DS_DSR,UCB$B_TT_DS_RCV(R5),25$
                             027D    25$:
                 51    00   9A 027D          MOVZBL  #MODEM$C_INIT,R1            ; ASSUME INIT MODEM PROTOCOL
            50    0114 C5   D0 0280          MOVL    UCB$L_TT_CLASS(R5),R0       ; ADDRESS CLASS VECTOR TABLE
                 0C B0      16 0285          JSB     @CLASS_DS_TRAN(R0)          ; INVOKE TO INIT MODEM PROTOCOL
                             0288    30$:
          08 64 A5    05    E1 0288          BBC     #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
            50    0114 C5   D0 028D          MOVL    UCB$L_TT_CLASS(R5),R0       ; GET THE CLASS VECTOR TABLE ADDRESS
                 20 B0      17 0292          JMP     @CLASS_POWERFAIL(R0)        ; AND GOTO THE POWERFAIL CODE
                             0295
                            05 0295  40$:    RSB
                             0296    ;
                             0296    ; ERROR DETECTED DURING INITIALIZATION
                             0296    ;
                             0296
                             0296    YF$UNIT_ERROR:
              64 A5    10   AA 0296          BICW    #UCB$M_ONLINE,UCB$W_STS(R5)        ; UNIT NOT ON LINE
                            05 029A          RSB
                             029B
                             029B                .SBTTL  MAINTENANCE ROUTINES
                             029B        ;++
                             029B        ; YF$MAINT - MAINTENANCE FUNCTIONS
                             029B        ;
                             029B        ; FUNCTIONAL DESCRIPTION:
                             029B        ; THIS ROUTINE PERFORMS MAINTENANCE FUNCTIONS FOR THE DHU
                             029B        ; (LOOPBACK IS ONLY ALLOWED ON LINES 0 AND 1)
                             029B        ;
                             029B        ; INPUTS:
                             029B        ;
                             029B        ;       R5 = UCB ADDRESS
                             029B        ;       UCB$B_TT_MAINT = FUNCTION TO BE PERFORMED
                             029B        ;
                             029B        ; OUTPUTS:
                             029B        ;       R0-R4 SCRATCH
                             029B        ;--
                             029B
                             029B        YF$MAINT:
                01      93   029B                BITB    #IO$M_LOOP@-7,-                 ; LOOPBACK FUNCTION
           012A C5           029D                        UCB$B_TT_MAINT(R5)
                05      13   02A0                BEQL    5$                              ; NO
           52   02      3C   02A2                MOVZWL  #^X02,R2                        ; SPECIFY LOOPBACK CODE
                0A      11   02A5                BRB     10$
                             02A7        5$:
                02      93   02A7                BITB    #IO$M_UNLOOP@-7,-               ; RESET LOOPBACK FUNCTION
           012A C5           02A9                        UCB$B_TT_MAINT(R5)
                26      13   02AC                BEQL    15$                             ; NO
           52   00      3C   02AE                MOVZWL  #^X00,R2                        ; SPECIFY UNLOOP CODE
                             02B1        10$:
                             02B1                SETIND
          51    08 A0   B0   02C0                MOVW    DHULCT(R0),R1
 51    02       06 52   F0   02C4                INSV    R2,#DHULCT$V_MAINT,#2,R1        ; SET MAINT FIELD
       08 A0    51      B0   02C9                MOVW    R1,DHULCT(R0)                   ; Update
           50   01      9A   02CD                MOVZBL  #1,R0                           ; INDICATE SUCCESS
                        05   02D0                RSB
                             02D1        50$:
                50      D4   02D1                CLRL    R0
                        05   02D3                RSB
                             02D4
                             02D4        15$:
             40 8F      93   02D4                BITB    #IO$M_AUTXOF_ENA@-7,-
           012A C5           02D7                        UCB$B_TT_MAINT(R5)              ;AUTOXON ENABLED
                05      13   02DA                BEQL    17$                             ; NO THEN MAYBE DISABLE
                20      A8   02DC                BISW    #TTY$M_PC_XOFAVL,-
           0122 C5           02DE                        UCB$W_TT_PRTCTL(R5)             ; SET THE BIT AVAILABLE
                             02E1
             80 8F      93   02E1        17$:    BITB    #IO$M_AUTXOF_DIS@-7,-
           012A C5           02E4                        UCB$B_TT_MAINT(R5)              ;AUTOXON disabled
                05      13   02E7                BEQL    19$                             ; no then don't disable it
                20      AA   02E9                BICW    #TTY$M_PC_XOFAVL,-
           0122 C5           02EB                        UCB$W_TT_PRTCTL(R5)
                             02EE
                             02EE        19$:
                04      93   02EE                BITB    #IO$M_LINE_OFF@-7,-             ; LINE OFF
           012A C5           02F0                        UCB$B_TT_MAINT(R5)
                1E      13   02F3                BEQL    30$                             ; NO
                             02F5                SETIND
```

```
08 A0    0104 8F   AA  0304             BICW    #DHULCT$M_RCV!DHULCT$M_MODEM,DHULCT(R0) ; Disable Receive and modem
     08 A0    01   A8  030A             BISW    #DHULCT$M_ABORT,DHULCT(R0)      ; Abort any current activity
        0D A0      94  030E             CLRB    DHUTBF2+1(R0)                   ; Disable Transmit
           25      11  0311             BRB     40$
                       0313       30$:
           10      93  0313             BITB    #IO$M_LINE_ON@-7,-              ; LINE ON
      012A C5          0315                     UCB$B_TT_MAINT(R5)
           B7      13  0318             BEQL    50$                             ; NO
                       031A             SETIND
08 A0    0104 8F   A8  0329             BISW    #DHULCT$M_RCV!DHULCT$M_MODEM,DHULCT(R0) ; Enable Receive and modem
     08 A0    01   AA  032F             BICW    #DHULCT$M_ABORT,DHULCT(R0)
  0D A0    80 8F   90  0333             MOVB    #^X80,DHUTBF2+1(R0)             ; Enable Transmit
        50    01   9A  0338       40$:  MOVZBL  #1,R0
                   05  033B             RSB
                       033C
                       033C
```

```
                            033C         .SBTTL  OUTPUT MODEM CONTROL
                            033C   ;++
                            033C   ; YF$DS_SET - SET OUTPUT MODEM SIGNALS
                            033C   ;
                            033C   ; FUNCTIONAL DESCRIPTION:
                            033C   ;
                            033C   ; THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT
                            033C   ;
                            033C   ; INPUTS:
                            033C   ;
                            033C   ;       R2 = LOW BYTE - SIGNALS TO ACTIVATE
                            033C   ;            HIGH BYTE- SIGNALS TO DEACTIVATE
                            033C   ;
                            033C   ;       R5 = UCB ADDRESS
                            033C   ;
                            033C   ; OUTPUTS:
                            033C   ;
                            033C   ;       R0-R3 ARE USED.
                            033C   ;--
                            033C   YF$DS_SET:
                            033C   ;
                            033C   ; Check that the DHU/DHV modem control signals have the standard values
                            033C   ;
      0125 C5    52    88   033C         BISB    R2,UCB$B_TT_DS_TX(R5)    ; SET NEW OUTPUT SIGNALS
   52   52    F8 8F    78   0341         ASHL    #-8,R2,R2                ; ACCESS SIGNALS TO RESET
      0125 C5    52    8A   0346         BICB    R2,UCB$B_TT_DS_TX(R5)    ; RESET THEM
                            034B         SETIND
7E   0125 C5    ED 8F  8B   035A         BICB3   #^C<<DHULCT$M_DTR!DHULCT$M_RTS>/256>,UCB$B_TT_DS_TX(R5),-(SP);
                            0361   ;
                            0361   ; The DHU/DHV is set to report modem change events in the receive fifo
                            0361   ; (Unless the line is disabled)
                            0361   ;
   03 012A C5    07    E0   0361         BBS     #UCB$V_TT_DSBL,UCB$B_TT_MAINT(R5),10$    ; Test if disabled
           6E    01    88   0367         BISB    #<DHULCT$M_MODEM/256>,(SP)
                            036A   10$:
        09 A0    8E    90   036A         MOVB    (SP)+,DHULCT+1(R0)               ; OUTPUT NEW VALUE
                            036E
                      05    036E         RSB
                            036F
                            036F
```

```
                          036F          .SBTTL  RECEIVER INTERRUPT SERVICE
                          036F    ;++
                          036F    ; YF$INTINP - DHU RECEIVER READY INTERRUPTS
                          036F    ;
                          036F    ; FUNCTIONAL DESCRIPTION:
                          036F    ;
                          036F    ; THIS ROUTINE IS ENTERED WHEN A CHARACTER IS AVAILABLE IN THE UNIT'S
                          036F    ; SILO. THE CHARACTER IS EXTRACTED AND IS PASSED TO THE ASSOCIATED
                          036F    ; CLASS DRIVER. IF THE CLASS DRIVER RETURNS CHARACTERS(S) THEN NEW
                          036F    ; OUTPUT IT INITIATED (NORMALLY ECHO).
                          036F    ;
                          036F    ; INPUTS:
                          036F    ;
                          036F    ;       00(SP) = ADDRESS OF IDB
                          036F    ;
                          036F    ; IMPLICIT INPUTS:
                          036F    ;
                          036F    ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON STACK.
                          036F    ;
                          036F    ; OUTPUTS:
                          036F    ;
                          036F    ;       THE INTERRUPT IS DISMISSED WHEN THE SILO IS EMPTY.
                          036F    ;
                          036F    ;--
                          036F    YF$INTINP::                                 ; DHU/DHV INPUT INTERRUPTS
                          036F    ;
                          036F    ; GET THE CSR ADDRESS
                          036F    ;
          54    9E    D0  036F          MOVL    @(SP)+,R4                   ; GET THE IDB ADDRESS
                54    DD  0372          PUSHL   R4                          ; SAVE IDB ADDRESS
          50    64    D0  0374          MOVL    (R4),R0                     ; GET THE CSR ADDRESS
                          0377    ;
                          0377    ; GET THE CHARACTER FROM THE INTERFACE
                          0377    ;
       53    02 A0    B0  0377    25$:  MOVW    DHURBF(R0),R3               ; Get the silo entry
                6D    18  037B          BGEQ    45$                         ; Silo empty (== BRW 100$)
    52    53    F8 8F  78  037D          ASHL    #-8, R3, R2                 ; shift the line number
    52   FFFFFFF0 8F  CA  0382          BICL    #^C<15>, R2                 ; use mask to obtain line number
       55    18 A442  D0  0389          MOVL    IDB$L_UCBLST(R4)[R2],R5     ; GET THE UCB FOR THAT LINE
                E7    13  038E          BEQL    25$                         ; IF EQL THEN NOT THERE
       53    7000 8F  B3  0390          BITW    #<DHURCV$M_PARERR>!-
                          0395                  <DHURCV$M_OVERRUN>!-
                          0395                  <DHURCV$M_FRAMER>,R3        ;ERRORS OR MODEM TRANSITION ?
                          0395
                55    12  0395          BNEQ    50$                         ;YES,PROCESS THEM
                          0397    27$:
          53    53    9A  0397          MOVZBL  R3,R3                       ; CLEAR THE HIGH BYTES OF CHARACTER
          0110  D5    16  039A          JSB     @UCB$L_TT_PUTNXT(R5)        ; BUFFER THE CHARACTER
                40    15  039E          BLEQ    40$                         ; NONE OR STRING OUTPUT
                          03A0          TIMSET  #1,R1,LOCKOUTPUT            ; SET TIMEOUT AND INTERRUPT BIT
                          03C1          SETIND  R0
    09 0134 C5    01    E0  03C8          BBS     #UCB$V_DHU,UCB$B_DHUFLG(R5),28$
 02 A0    53    8000 8F  A9  03CE          BISW3   #^X8000,R3,DHUTXC(R0)       ; DHV single char output
                04    11  03D5          BRB     30$
                          03D7    28$:
       06 A0    53    90  03D7          MOVB    R3,DHUTXF(R0)               ; DHU fifo output
          54    6E    D0  03DB    30$:  MOVL    (SP),R4                     ; GET IDB ADDRESS
```

```
                 97    11  03DE            BRB      25$                        ; CONTINUE
                           03E0  40$:
                 F9    13  03E0            BEQL     30$                        ; NO CHARACTER
      00000443'EF      16  03E2            JSB      BURST_OUTPUT               ; START BURST
                 F1    11  03E8            BRB      30$
                           03EA
                           03EA  45$:
                 18    11  03EA            BRB      100$
                           03EC  ;
                           03EC  50$:
                           03EC  ;
                           03EC  ; PROCESS PARITY, FRAME OVERRUN ERROR OR MODEM TRANSITION
                           03EC  ;
                           03EC  ;
                           03EC  ; The DHU indicates modem transition by setting all the
                           03EC  ; error bits
         08 53   0C    E1  03EC            BBC      #DHURCV$V_PARERR,R3,60$
         04 53   0E    E1  03F0            BBC      #DHURCV$V_OVERRUN,R3,60$
         19 53   0D    E0  03F4            BBS      #DHURCV$V_FRAMER,R3,200$; Modem transition if all set
                           03F8  60$:
      52    0114 C5    D0  03F8            MOVL     UCB$L_TT_CLASS(R5),R2      ; GET CLASS DISPATCH
              14 B2    16  03FD            JSB      @CLASS_READERROR(R2)       ; SIGNAL ERROR
                 95    12  0400            BNEQ     27$                        ; CHRRACTER TO ECHO
                           0402  70$:
                 D7    11  0402            BRB      30$
                           0404
            5E   04    C0  0404  100$:     ADDL     #4,SP                      ; REMOVE IDB ADDRESS
            50   8E    7D  0407            MOVQ     (SP)+,R0                   ; RESTORE REGISTERS
            52   8E    7D  040A            MOVQ     (SP)+,R2                   ;
            54   8E    7D  040D            MOVQ     (SP)+,R4                   ;
                       02  0410            REI
                           0411  ;
                           0411  ; Modem transition routine
                           0411  ;
                           0411  200$:
                           0411  ; If lsb set then it is a self test code
                           0411  ; We currently ignore self test codes, although these could be used
                           0411  ; for error logging purposes
                           0411
              C7 53    E8  0411            BLBS     R3,30$
                           0414  ;
                           0414  ; Unfortunately the DHU and DHV dont put the modem bits where we want them
                           0414  ; To get the correct bits we have to move RI,DCD,CTS up one place, while not
                           0414  ; changing DSR. spare bits have to be ignored.
            53   53    C0  0414            ADDL     R3,R3                      ; (This shifts DSR out of bottom byte)
      53    8F   8F    8A  0417            BICB     #^C<TT$M_DS_RING!TT$M_DS_CTS!TT$M_DS_CARRIER>,R3
         04 53   08    E1  041B            BBC      #DHUSTT$V_DSR+1,R3,210$
         00 53   07    E2  041F            BBSS     #TT$V_DS_DSR,R3,210$
                           0423  210$:
   0124 C5  53         90  0423            MOVB     R3,UCB$B_TT_DS_RCV(R5)     ; UPDATE CURRENT INPUT MODEM SIGNALS
         52  53        90  0428            MOVB     R3, R2                     ; PASS CURRENT INPUT MODEM SIGNALS IN R2
         51  03        9A  042B            MOVZBL   #MODEM$C_DATASET,R1        ; TRANSITION TYPE IS DATASET
             50        DD  042E            PUSHL    R0                         ; SAVE CSR ADDRESS
      54    0114 C5    D0  0430            MOVL     UCB$L_TT_CLASS(R5),R4      ; GET CLASS DISPATCH
              0C B4    16  0435            JSB      @CLASS_DS_TRAN(R4)         ; INVOKE TRANSITION ROUTINE
                 50  8ED0  0438            POPL     R0                         ; RESTORE CSR ADDRESS
               FF9D    31  043B            BRW      30$
```

```
                                        043E   897
                                        043E   898
```

```
                            043E            .SBTTL  START I/O ROUTINE
                            043E    ;++
                            043E    ; YF$STARTIO - START I/O OPERATION ON DHU
                            043E    ;
                            043E    ; FUNCTIONAL DESCRIPTION:
                            043E    ;
                            043E    ; THIS ROUTINE IS ENTERED FROM THE DEVICE INDEPENDENT TERMINAL STARTIO
                            043E    ; ROUTINE TO ENABLE OUTPUT INTERRUPTS ON AN IDLE DHU UNIT.
                            043E    ;
                            043E    ; INPUTS:
                            043E    ;
                            043E    ;       R3 =    CHARACTER       AND     CC = PLUS
                            043E    ;               ADDRESS         AND     CC = NEGATIVE
                            043E    ;
                            043E    ;       R5 = UCB ADDRESS
                            043E    ;
                            043E    ; OUTPUTS:
                            043E    ;
                            043E    ;       R5 = UCB ADDRESS
                            043E    ;--
                            043E            .ENABLE LSB
                            043E    YF$STARTIO::                            ; START I/O ON UNIT
                 03    19   043E            BLSS    BURST_OUTPUT
               006B    31   0440            BRW     90$
                            0443
                            0443    BURST_OUTPUT:
                            0443            SETIND
     52    0120 C5     3C   0452            MOVZWL  UCB$W_TT_OUTLEN(R5),R2  ; GET LENGTH
                 01    E1   0457            BBC     #TTY$V_PC_DMAENA,-      ; USE SILO IF DMA NOT ENABLED ON THIS LINE
        0C 0122 C5          0459                    UCB$W_TT_PRTCTL(R5),SILO_OUTPUT
  00000000'GF    52    B1   045D            CMPW    R2,G^TTY$GW_DMASIZE     ; LARGE ENOUGH FOR DMA
                 03    19   0464            BLSS    SILO_OUTPUT
               0088    31   0466    10$:    BRW     DMA_START               ; YES SO DO DMA
                            0469
                            0469    SILO_OUTPUT:
   64 0134 C5    01    E1   0469            BBC     #UCB$V_DHU,UCB$B_DHUFLG(R5),200$        ; DHV has single char output
     51    06 A0       9A   046F            MOVZBL  DHUTFS(R0),R1                   ; get number of slots
            51    52   B1   0473            CMPW    R2,R1                   ; BURST LARGER THAN SILO?
                  03   1B   0476            BLEQU   50$                     ; NO
            52    51   9A   0478            MOVZBL  R1,R2                   ; SLOTS AVAILABLE IS MAXIMUM
                            047B    50$:
     53    011C C5     D0   047B            MOVL    UCB$L_TT_OUTADR(R5),R3  ; GET ADDRESS
      011C C5    52    C0   0480            ADDL    R2,UCB$L_TT_OUTADR(R5)  ; UPDATE POINTER
      0120 C5    52    A2   0485            SUBW    R2,UCB$W_TT_OUTLEN(R5)  ; AND COUNT
                 07    13   048A            BEQL    60$                     ; ALL DONE, NO NEED FOR BURST
         0800 8F       A8   048C            BISW    #TTY$M_TANK_BURST,-     ; SIGNAL BURST ACTIVE
         0108 C5            0490                    UCB$W_TT_HOLD(R5)
                            0493    60$:
            08    52   E9   0493            BLBC    R2,70$                  ; EVEN TRANSFER
        06 A0     83   90   0496            MOVB    (R3)+,DHUTXF(R0)        ; OUTPUT ODD BYTE
                  52   D7   049A            DECL    R2                      ; UPDATE COUNT
                  0F   13   049C            BEQL    80$                     ; DONE
                            049E    70$:
    52    52    FF 8F  78   049E            ASHL    #-1,R2,R2               ; CONVERT TO WORD COUNT
                            04A3    75$:
        06 A0     83   B0   04A3            MOVW    (R3)+,DHUTXF(R0)
                            04A7            DELAY                           ; TO RELEASE THE UNIBUS
```

```
                    F6 52    F5  04AA   957          SOBGTR  R2,75$                    ; LOOP TILL DONE
                                 04AD   958
                             05  04AD   959 80$:     RSB                               ; RETURN TO CALLER
                                 04AE   960
                                 04AE   961 90$:
                       22    13  04AE   962          BEQL    100$                      ; SKIP IF NONE
                                 04B0   963          SETIND
          09 0134 C5   01    E0  04BF   964          BBS     #UCB$V_DHU,UCB$B_DHUFLG(R5),95$
                                 04C5   965 ;
                                 04C5   966 ; DHV single character output
                                 04C5   967 ;
     02 A0    53   8000 8F   A9  04C5   968          BISW3   #^X8000,R3,DHUTXC(R0)
                       04    11  04CC   969          BRB     100$
                                 04CE   970 ;
                                 04CE   971 ; DHU fifo output
                                 04CE   972 ;
                                 04CE   973 95$:
               06 A0   53    90  04CE   974          MOVB    R3,DHUTXF(R0)
                                 04D2   975 100$:
                             05  04D2   976          RSB
                                 04D3   977 ;
                                 04D3   978 ; DHV 'silo' output, uses single character mode
                                 04D3   979 ;
                                 04D3   980 200$:
               53   011C D5  9A  04D3   981          MOVZBL  @UCB$L_TT_OUTADR(R5),R3 ; GET character
                    011C C5  D6  04D8   982          INCL    UCB$L_TT_OUTADR(R5)     ; UPDATE POINTER
                    0120 C5  B7  04DC   983          DECW    UCB$W_TT_OUTLEN(R5)     ; AND COUNT
                         07  13  04E0   984          BEQL    260$                    ; ALL DONE, NO NEED FOR BURST
                    0800 8F  A8  04E2   985          BISW    #TTY$M_TANK_BURST,-     ; SIGNAL BURST ACTIVE
                    0108 C5      04E6   986                  UCB$W_TT_HOLD(R5)
                                 04E9   987 260$:
     02 A0    53   8000 8F   A9  04E9   988          BISW3   #^X8000,R3,DHUTXC(R0)   ; output the character
                             05  04F0   989          RSB
                                 04F1   990
                                 04F1   991          .DISABLE        LSB
                                 04F1   992
```

```
                      04F1          .SBTTL  PORT DMA ROUTINES
                      04F1    ;++
                      04F1    ; DMA_START     -       INITIATE DMA OUTPUT
                      04F1    ;
                      04F1    ; FUNCTIONAL DESCRIPTION:
                      04F1    ;
                      04F1    ; THESE ROUTINES ARE CALLED BY THE PORT INPUT INTERRUPT, OUTPUT
                      04F1    ; INTERRUPT, AND STARTIO TO INITIATE NEW DMA OUTPUT.
                      04F1    ; THEY HANDLE ALLOCATION AND LOADING OF MAP REGISTERS
                      04F1    ; TO HANDLE DMA OUTPUT. MAP REGISTERS ARE ALLOCATED IN PAIRS
                      04F1    ; TO ALLOW OUTPUT BURSTS UP TO 512 BYTES. TRANSFERS LARGER THAN
                      04F1    ; THAT ARE DONE IN SEGMENTS. IF INSUFFICIENT MAPS ARE AVAILABLE,
                      04F1    ; THE TRANSFER IS DONE SILO MODE.
                      04F1    ;
                      04F1    ;
                      04F1    ; INPUTS:
                      04F1    ;
                      04F1    ;       R5 = UCB ADDRESS
                      04F1    ;
                      04F1    ; OUTPUTS:
                      04F1    ;
                      04F1    ;       R5 = UCB ADDRESS
                      04F1    ;
                      04F1    ;       R0 PRESERVED
                      04F1    ;       R1,R2,R3,R4 DESTROYED
                      04F1    ;--
                      04F1    DMA_START:
     1000 8F      A8  04F1            BISW    #TTY$M_TANK_DMA,-       ; SHOW DMA MODE ACTIVE
     0108 C5          04F5                    UCB$W_TT_HOLD(R5)
          00      E5  04F8            BBCC    #TTY$V_TP_ABORT,-       ; RESET ANY OLD ABORT REQUESTS
  04 0130 C5          04FA                    UCB$B_TP_STAT(R5),3$
    08 A0 01      AA  04FE            BICW    #DHULCT$M_ABORT, DHULCT(R0)
                      0502    ;
                      0502    ; If there are no mapping registers for the device then don't allocate them
                      0502    ;
55 0134 C5  00    E1  0502    3$:     BBC     #UCB$V_MAP, UCB$B_DHUFLG(R5),DMA_CONTINUE
                      0508    ;               CHECK IF UNIT HAS PERMANENT MAP REGISTERS
                      0508
          03      E1  0508            BBC     #TTY$V_PC_PRMMAP,-      ; SKIP IF NOT AUTHORIZED FOR PERM MAPS
  06 0122 C5          050A                    UCB$W_TT_PRTCTL(R5),5$
          04      E0  050E            BBS     #TTY$V_PC_MAPAVL,-      ; SKIP FORK IF MAPS ALLOCATED ALREADY
  49 0122 C5          0510                    UCB$W_TT_PRTCTL(R5),DMA_CONTINUE
                      0514    5$:
          02      88  0514            BISB    #TTY$M_TP_ALLOC,-       ; SHOW ALLOC FORK ACTIVE
     0130 C5          0516                    UCB$B_TP_STAT(R5)
  51 0114 C5      D0  0519            MOVL    UCB$L_TT_CLASS(R5),R1   ; GET CLASS VECTOR ADDRESS
       1C B1      16  051E            JSB     @CLASS_FORK(R1)         ; FORK TO FIPL FOR MAP
                      0521                                            ; REGISTER ALLOCATION
                  05  0521            RSB                             ; RETURN TO CALLER WITH
                      0522                                            ; "INT" LEFT ON TO INTERLOCK
                      0522                                            ; OUTPUT. FORK ROUTINE WILL
                      0522                                            ; RESUME AT DMA_ALLOC.
                      0522    DMA_ALLOC:
                      0522
                      0522
                      0522    5$:
       53 02      9A  0522            MOVZBL  #2,R3                   ; REQUEST 2 MAP REGISTERS
```

```
      00000000'GF  16  0525            JSB     G^IOC$ALOUBAMAPN          ;
                       052B
                       052B            SETIPL  UCB$B_DIPL(R5)            ; INTERLOCK TO DEVICE IPL
               10  A8  052F            BISW    #TTY$M_PC_MAPAVL,-        ; SHOW MAP ALLOCATED
          0122 C5      0531                    UCB$W_TT_PRTCTL(R5)
               02  8A  0534            BICB    #TTY$M_TP_ALLOC,-         ; SHOW ALLOC FORK DONE
          0130 C5      0536                    UCB$B_TP_STAT(R5)
            17 50  E8  0539            BLBS    R0,20$                    ; SUCCESS
                       053C
     50    24 A5   D0  053C            MOVL    UCB$L_CRB(R5),R0          ; GET CRB OF UNIT
     50    2C B0   D0  0540            MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0; GET CSR
          100C 8F  AA  0544            BICW    #TTY$M_TANK_DMA,-         ; RESET DMA MODE
          0108 C5      0548                    UCB$W_TT_HOLD(R5)
   52     0120 C5  3C  054B            MOVZWL  UCB$W_TT_OUTLEN(R5),R2    ; RESTORE OUTPUT LENGTH
             FF16  31  0550            BRW     SILO_OUTPUT               ; USE SILO FOR OUTPUT
                       0553
                       0553    20$:
     50    24 A5   D0  0553            MOVL    UCB$L_CRB(R5),R0          ; GET CRB ADDRESS
           34 A0   D0  0557            MOVL    CRB$L_INTD+VEC$W_MAPREG(R0),-
          012C C5      055A                    UCB$L_TP_MAP(R5)          ; SAVE  MAP FIELD IN UCB
                       055D
                       055D
                       055D    DMA_CONTINUE:
               00  E0  055D            BBS     #TTY$V_TP_ABORT,-         ; BRANCH IF DMA TO BE ABORTED
       0C 0130 C5      055F                    UCB$B_TP_STAT(R5),2$
                       0563
   53     011C C5  D0  0563            MOVL    UCB$L_TT_OUTADR(R5),R3    ; GET ADDRESS OF NEXT STRING
   52     0120 C5  3C  0568            MOVZWL  UCB$W_TT_OUTLEN(R5),R2    ; LENGTH OF OUTPUT
               03  12  056D            BNEQ    4$                        ; SKIP IF MORE TO DC
             0162  31  056F    2$:     BRW     DMA_DONE                  ; BRANCH IF TRANSFER IS DONE
                       0572    4$:
               50  DD  0572            PUSHL   R0                        ; SAVE INPUT VOLITAL REGISTER "CSR"
                       0574
                       0574    ;
                       0574    ; If there are no mapping registers for the device then
                       0574    ; start dma directly
                       0574    ;
  03 0134 C5   00  E1  0574            BBC     #UCB$V_MAP, UCB$B_DHUFLG(R5),DMA_NOMAP
             00A0  31  057A            BRW     DMA_MAP
```

```
                                  057D  DMA_NOMAP:
                                  057D  ;
                                  057D  ; Special code to work with an unmapped DHU/DHV
                                  057D  ; (I.e. Seahorse configuration)
                                  057D  ;
                                  057D  ; this code examines the DMA buffer to find the maximum physically
                                  057D  ; contiguous area, and starts a DMA on that part.
                                  057D  ;
                                  057D  ; It assumes that the buffer is in non-paged pool
                                  057D  ; (I.e. that it is in system address space and that the PTE
                                  057D  ; will always contain the PFN for the page)
                                  057D  ;
                                  057D  ; Find physical address of first page
                                  057D  ;
             53   011C C5     D0  057D          MOVL    UCB$L_TT_OUTADR(R5),R3
     54   53  80000000 8F     CB  0582          BICL3   #^X80000000,R3,R4        ; CALC SVAPTE OF BUFFER
           54   54   F7 8F    78  058A          ASHL    #-9,R4,R4                ; ISOLATE PAGE
      50    00000000'GF       D0  058F          MOVL    G^MMG$GL_SPTBASE,R0      ; GETS SVAPTE OF BUFFER
                54   6044     DE  0596          MOVAL   (R0)[R4],R4              ; INTO R4
      53   FFFFFE00 8F        CA  059A          BICL    #^C^X1FF,R3              ; COMPUTE BYTE OFFSET IN PAGE
                                  05A1  ;
                                  05A1  ; calculate length of DMA in this page
                                  05A1  ;
     51   00000200 8F    53   C3  05A1          SUBL3   R3,#512,R1
                                  05A9  ;
                                  05A9  ; See if this is longer than buffer
                                  05A9  ;
           0120 C5       51   B1  05A9          CMPW    R1,UCB$W_TT_OUTLEN(R5)
                         05   15  05AE          BLEQ    10$
           51   0120 C5       B0  05B0          MOVW    UCB$W_TT_OUTLEN(R5),R1  ; use actual buffer length
                                  05B5  10$:
           0120 C5       51   A2  05B5          SUBW    R1,UCB$W_TT_OUTLEN(R5)
                                  05BA  ;
                                  05BA  ; Get the PFN for the first page
                                  05BA  ;
                   52    84   D0  05BA          MOVL    (R4)+,R2
                                  05BD
                                  05BD  ; calculate physical address from PTE
                                  05BD  ; (assuming that it fits into a 22-bit address)
                                  05BD
        52   52    15    00   EF  05BD          EXTZV   #0,#21,R2,R2            ; get PFN only
        53   0D    09    52   F0  05C2          INSV    R2,#9,#13,R3
                                  05C7  ;
                                  05C7  ; Loop through remaining pages to see if they are contiguous with this one
                                  05C7  ;
                                  05C7  20$:
                 0120 C5     B5   05C7          TSTW    UCB$W_TT_OUTLEN(R5)
                         2D   13  05CB          BEQL    50$                     ; No more DMA
                         52   D6  05CD          INCL    R2                      ; get expected PTE
        52   64    15    00   ED  05CF          CMPZV   #0,#21,(R4),R2          ; test next PTE
                         24   12  05D4          BNEQ    50$                     ; not contiguous
                         84   D5  05D6          TSTL    (R4)+                   ; step to next PTE
                                  05D8  ;
                                  05D8  ; concatenate this page to previous DMA
                                  05D8  ;
      0200 8F    0120 C5     B1   05D8          CMPW    UCB$W_TT_OUTLEN(R5),#512
                         10   15  05DF          BLEQ    40$                     ; partial page
```

```
                                 05E1
        51   00000200 8F    C0   05E1            ADDL    #512,R1                       ; whole page gets added
      0120 C5    0200 8F    A2   05E8            SUBW    #512,UCB$W_TT_OUTLEN(R5)
                      D6    11   05EF            BRB     20$
                                 05F1
                                 05F1     ; partial page gets added, to complete the DMA
                                 05F1     40$:
             51    0120 C5  A0   05F1            ADDW    UCB$W_TT_OUTLEN(R5),R1
                   0120 C5  B4   05F6            CLRW    UCB$W_TT_OUTLEN(R5)
                                 05FA     50$:
                                 05FA     ;
                                 05FA     ; Start up the DMA
                                 05FA     ;
             011C C5   51   C0   05FA            ADDL    R1,UCB$L_TT_OUTADR(R5)        ; step to next DMA address
                       50 8ED0   05FF            POPL    R0                            ; RESTORE CSR ADDRESS
                                 0602            SETIND  R0
               0E A0   51   B0   0609            MOVW    R1,DHUTCT(R0)                 ; load the count
               0A A0   53   B0   060D            MOVW    R3,DHUTBF1(R0)                ; load low address
        53    53   06  10   EF   0611            EXTZV   #16,#6,R3,R3                  ; get high address
     0C A0    53   80 8F    89   0616            BISB3   #^X80,R3,DHUTBF2(R0)          ; load high address and start
                            05   061C            RSB
```

```
                                 061D  ;
                                 061D  ; Code to do mapped DMA transfers
                                 061D  ;
                                 061D  DMA_MAP:
         00000200 8F    52   D1  061D               CMPL    R2,#512                   ; NEXT BURST TOO LONG FOR MAPS?
                        05   15  0624               BLEQ    5$                        ; NO
              52   0200 8F   3C  0626               MOVZWL  #512,R2
                                 062B
                                 062B
              011C C5   52   C0  062B  5$:          ADDL    R2,UCB$L_TT_OUTADR(R5)    ; UPDATE CHARACTER POINTER FOR NEXT TIME
              0120 C5   52   A2  0630               SUBW    R2,UCB$W_TT_OUTLEN(R5)    ; UPDATE COUNT FOR NEXT TIME
                                 0635
           21 0134 C5   02   E0  0635               BBS     #U(B$V_XOFF,UCB$B_DHUFLG(R5),6$ ; IF XOFF, DON'T SET TIMER
                                 063B               TIMSET  R2,R1,[OCKOUTPUT           ;RECOMPUTE TIMEOUT VALUE FOR THIS
                                 065C                                                  ;PORTION OF THE DMA BURST
                                 065C
                                 065C
                                 065C  ;                    R3 - STRING ADDRESS
                                 065C  ;                    R2 - LENGTH
                                 065C  ;                    R5 - UCB
                                 065C
                        24   BB  065C  6$:          PUSHR   #^M<R2,R5>
               50    24 A5   D0  065E               MOVL    UCB$L_CRB(R5),R0          ; GET CRB ADDRESS
               51    38 B0   D0  0662               MOVL    @CRB$L_INTD+VEC$L_ADP(R0),R1    ; CONFIG REGISTER
                  0F    00   EF  0666               EXTZV   #VEC$V_MAPREG,#VEC$S_MAPREG,-
              50   012C C5       0669                       UCB$L_TP_MAP(R5),R0       ; GET STARTING MAP REGISTER
            51   0800 C140   DE  066D               MOVAL   UBA$L_MAP(R1)[R0],R1      ; GET 1ST MAP REGISTER ADDRESS
                                 0673
      54    53   80000000 8F  CB  0673               BICL3   #^X80000000,R3,R4         ; CALC SVAPTE OF BUFFER
            54    54   F7 8F  78  067B               ASHL    #-9,R4,R4                 ; ISOLATE PAGE
            55    00000000'GF  D0  0680               MOVL    G^MMG$GL_SPTBASE,R5       ; GETS SVAPTE OF BUFFER
                  54    6544  DE  0687               MOVAL   (R5)[R4],R4               ; INTO R4
            53    FFFFFE00 8F  CA  068B               BICL    #^C^X1FF,R3               ; COMPUTE BYTE OFFSET IN PAGE
                                 0692
                                 0692  ;            LOAD MAP REGISTERS
                                 0692  ;                    R0 - MAP REGISTER NUMB[illegible]
                                 0692  ;                    R1 - ADDRESS OF FIRST [illegible] REGISTER
                                 0692  ;                    R2 - BUFFER LENGTH
                                 0692  ;                    R3 - BYTE OFFSET IN PAGE
                                 0692  ;                    R4 - SVAPTE OF BUFFER
                                 0692
                   52   02   9A  0692               MOVZBL  #2,R2
                   55   84   D0  0695  10$:         MOVL    (R4)+,R5                  ; GET CONTENTS OF NEXT PTE
                                 0698
                                 0698  ;            THIS CODE ASSUMES THAT DMA IS FROM NONPAGED POOL
                                 0698
  55    0B    15   00000400 8F   F0  0698               INSV    #^X400,#21,#11,R5         ; SET VALID BIT, DATA PATH 0
                   81   55   D0  06A1               MOVL    R5,(R1)+                  ; LOAD INTO MAP REGISTER
                   EE 52     F5  06A4               SOBGTR  R2,10$
                                 06A7
                        24   BA  06A7               POPR    #^M<R2,R5>                ; RESTORE LENGTH,WRITE BUFFER, UCB
                                 06A9
                                 06A9  ;
                                 06A9  ; Note that the following code works with 22 bit Qbus addresses
                                 06A9  ; As well as with 18 bit Unibus addresses
                                 06A9
        53    0D    09    50     F0  06A9               INSV    R0,#9,#13,R3              ; COMPUTE UNIBUS ADDRESS
```

```
                          06AE                  SETIND
          0E A0   52   B0 06BD                  MOVW    R2,DHUTCT(R0)             ; Load count
          0A A0   53   B0 06C1                  MOVW    R3,DHUTBF1(R0)            ; Load low bits
    53   53   06   10   EF 06C5                  EXTZV   #16,#6,R3,R3              ; get high bits
 0C A0   53   80 8F   89 06CA                  BISB3   #^X80,R3,DHUTBF2(R0)      ; load high bits and initiate DMA
             50 8ED0    06D0                  POPL    R0                        ; RESTORE CSR ADDRESS
                   05    06D3                  RSB                               ; RETURN TO CALLER
                          06D4                                                    ; FORK DISPATCHER, ISR, OR STARTIO
                          06D4
                          06D4  DMA_DONE:                                         ; DMA COMPLETION
                          06D4
                          06D4
              03   E0    06D4                  BBS     #TTY$V_PC_PRMMAP,-        ; SKIP FORK IF MAPS PERMANENT
      2E 0122 C5         06D6                          UCB$W_TT_PRTCTL(R5),DMA_POST
              04   88    06DA                  BISB    #TTY$M_TP_DLLOC,-         ; SHOW DEALLOC FORK ACTIVE
         0130 C5         06DC                          UCB$B_TP_STAT(R5)
   51    0114 C5   D0    06DF                  MOVL    UCB$L_TT_CLASS(R5),R1     ; GET CLASS VECTOR ADDRESS
            1C B1   16   06E4                  JSB     @CLASS_FORK(R1)           ; SCHEDULE FORK TO FIPL FOR MAP
                          06E7                                                    ; REGISTER DEALLOCATION
                   05    06E7                  RSB                               ; RETURN TO CALLER, FORK WILL RESUME
                          06E8                                                    ; AT DMA_DEALLOC
                          06E8
                          06E8  DMA_DEALLOC:
      50    24 A5   D0   06E8                  MOVL    UCB$L_CRB(R5),R0          ; GET CRB ADDRESS
         012C C5   D0    06EC                  MOVL    UCB$L_TP_MAP(R5),-
            34 A0        06F0                          CRB$L_INTD+VEC$W_MAPREG(R0); RESTORE MAP FIELD IN CRB
              06   13    06F2                  BEQL    5$                        ; SKIP IF NONE
      00000000'GF   16   06F4                  JSB     G^IOC$RELMAPREG           ; RELEASE MAP REGISTERS
                          06FA
                          06FA  5$:            SETIPL  UCB$B_DIPL(R5)            ; INTERLOCK TO DEVICE IPL
              10   AA    06FE                  BICW    #TTY$M_PC_MAPAVL,-        ; SHOW MAP ALLOCATED
         0122 C5         0700                          UCB$W_TT_PRTCTL(R5)
              04   8A    0703                  BICB    #TTY$M_TP_DLLOC,-         ; SHOW DEALLOC FORK DONE
         0130 C5         0705                          UCB$B_TP_STAT(R5)
                          0708
                          0708  DMA_POST:
              01   8A    0708                  BICB    #TTY$M_TP_ABORT,-         ; RESET ABORT REQUEST
         0130 C5         070A                          UCB$B_TP_STAT(R5)
         1000 8F   AA    070D                  BICW    #TTY$M_TANK_DMA,-         ; RESET DMA MODE
         0108 C5         0711                          UCB$W_TT_HOLD(R5)
                          0714
                          0714  ;              CALL GETNEXT TO CONTINUE PROCESSING
                          0714  ;
              03   8A    0714                  BICB    #UCB$M_TIM!UCB$M_INT,-   ;CLEAR TIMEOUT AND INT EXPECTED
            64 A5        0716                          UCB$W_STS(R5)
         010C D5   16    0718                  JSB     @UCB$L_TT_GETNXT(R5)      ; GET NEXT BURST
            FD1F   31    071C                  BRW     YF$STARTIO                ; AND PROCEED
                          071F  YF$FORK:                                          ;
                          071F                 SAVIPL                            ; SAVE CURRENT IPL ON THE STACK
      0000073A'EF   DF   0722                  PUSHAL  20$                       ; BUILD RETURN ADDRESS ON STACK
              01   E1    0728                  BBC     #TTY$V_TP_ALLOC,-         ; SKIP IF NOT ALLOCATE FORK
      03 0130 C5         072A                          UCB$B_TP_STAT(R5),10$
            FDF1   31    072E                  BRW     DMA_ALLOC                 ; RESUME AT ALLOCATE CODE THREAD
                          0731  10$:
              02   E1    0731                  BBC     #TTY$V_TP_DLLOC,-         ; CHECK FOR DEALLOCATE
      03 0130 C5         0733                          UCB$B_TP_STAT(R5),20$
            FFAE   31    0737                  BRW     DMA_DEALLOC
```

```
YFD
Sym

$$$
$$O
AT$
BIT
BUG
BUR
CLA
CLA
CLA
CLA
CLA
CLA
CLA
CLA
CRB
CRB
CRB
DC$
DDB
DEV
DEV
DEV
DEV
DEV
DEV
DEV
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
```

```
        073A  1284 20$:
        073A  1285          ENBINT                          ; RESTORE SAVED FORK IPL FROM STACK
        073D  1286
    05  073D  1287          RSB
        073E  1288
```

YFD
Sym

DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DHU
DMA
DMA
DMA
DMA
DMA
DMA
DMA
DMA

```
                        073E          .SBTTL  PORT ROUTINES  STOP,RESUME,XON,XOFF
                        073E    ;++
                        073E    ; YF$XOFF -       SEND XOFF
                        073E    ; YF$XON -        SEND XON
                        073E    ; YF$STOP -       STOP OUTPUT
                        073E    ; YF$ABORT -      ABORT CURRENT OUTPUT
                        073E    ; YF$RESUME -     RESUME STOPPED OUTPUT
                        073E    ;
                        073E    ; FUNCTIONAL DESCRIPTION:
                        073E    ;
                        073E    ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                        073E    ; CONTROL OUTPUT ON THE PORT
                        073E    ;
                        073E    ; INPUTS:
                        073E    ;
                        073E    ;         R5 = UCB ADDRESS
                        073E    ;
                        073E    ; OUTPUTS:
                        073E    ;
                        073E    ;         R5 = UCB ADDRESS
                        073E    ;--
                        073E    ;
                        073E    ; SCHEDULE XOFF OR XON TO BE SEND
                        073E    ;
                        073E    ; INPUTS:
                        073E    ;
                        073E    ;         R3 - CONTAINS THE CHARACTER TO SEND AS FLOW CONTROL.
                        073E    ;
                        073E    ;
                        073E    ; To send an XON we just clear the Force XOFF bit
                        073E    YF$XON:
                        073E    ;
                        073E    ; Forget any stored 'XOFF' character
     0100 8F       AA   073E          BICW    #TTY$M_TANK_PREMPT,-    ; RESET XOFF STATE
     0108 C5            0742                  UCB$W_TT_HOLD(R5)
                        0745          SETIND
                        0754    ;
                        0754    ; Clearing this bit will make the device send an XON asap
                        0754    ;
  08 A0    20      AA   0754          BICW    #DHULCT$M_SNDOFF,DHULCT(R0)
     11    53      91   0758          CMPB    R3,#^X11                ; Is it XOFF ?
           1A      12   075B          BNEQ    YF$PREEMPT
                   05   075D          RSB
                        075E    ;
                        075E    ; To send an XOFF we just set the Force XOFF bit,
                        075E    ;
                        075E    YF$XOFF:
                        075E          SETIND
     13    53      91   076D          CMPB    R3,#^X13                ; Is it XOFF ?
           05      12   0770          BNEQ    YF$PREEMPT              ; Not XOFF, have to do it the hard way
  08 A0    20      A8   0772          BISW    #DHULCT$M_SNDOFF,DHULCT(R0)
                   05   0776          RSB
                        0777    ;
                        0777    ; we have to send a character here (other than normal XON/XOFF),
                        0777    ; so see if the device is idle
                        0777    ;
                        0777    YF$PREEMPT:
```

```
      3B 64 A5   01    E2  0777          BBSS    #UCB$V_INT,UCB$W_STS(R5),30$     ; Branch if active
            0D A0    95  077C          TSTB    DHUTBF2+1(R0)
               21    18  077F          BGEQ    10$                              ; XOFFED, don't set timer
                         0781          TIMSET  #1,R1,LOCKOUTPUT
                         07A2  ;
                         07A2  ; If this is a DHV then we send the char by single character,
                         07A2  ; else by fifo
                         07A2  ;
                         07A2  10$:
   09 0134 C5    01    E0  07A2          BBS     #UCB$V_DHU,UCB$B_DHUFLG(R5),20$
02 A0   53   8000 8F   A9  07A8          BISW3   #^X8000,R3,DHUTXC(R0)
               12    11  07AF          BRB     90$
                         07B1  20$:
       06 A0     53    90  07B1          MOVB    R3,DHUTXF(R0)
               0C    11  07B5          BRB     90$
                         07B7  ;
                         07B7  ; Transmission is in progress, save the character till
                         07B7  ; the transmission completes
                         07B7  30$:
        0100 8F        A8  07B7          BISW    #TTY$M_TANK_PREMPT,-     ; Set the flag
        0108 C5            07BB                  UCB$W_TT_HOLD(R5)
     010A C5     53    90  07BE          MOVB    R3,UCB$B_TT_PREMPT(R5)   ; Save the character
                         07C3
                         07C3  90$:
                     05  07C3          RSB
                         07C4
                         07C4  ;
                         07C4  ; STOP PORT OUTPUT
                         07C4  ;
                         07C4  YF$STOP:
                 50    DD  07C4          PUSHL   R0
                         07C6          SETIND
            0D A0      94  07D5          CLRB    DHUTBF2+1(R0)
                         07D8  ;
                         07D8  ; Note. We dont reset UCB$M_INT for the DHU,DHV in case the
                         07D8  ; device finishes transmitting
                         07D8  ;
        0134 C5  04    88  07D8          BISB    #UCB$M_XOFF,UCB$B_DHUFLG(R5)    ; set to indicate xoff
          64 A5  01    8A  07DD          BICB    #UCB$M_TIM,UCB$W_STS(R5)        ; Reset timer
                 01    BA  07E1          POPR    #^M<R0>
                       05  07E3          RSB
                         07E4  ;
                         07E4  ; ABORT ANY CURRENT PORT OUTPUT ACTIVITY
                         07E4  ;
                         07E4  YF$ABORT:
                 50    DD  07E4          PUSHL   R0
      18 64 A5   01    E1  07E6          BBC     #UCB$V_INT,UCB$W_STS(R5),15$          ; SKIP IF NOT BUSY.
        0130 C5  01    88  07EB          BISB    #TTY$M_TP_ABORT,UCB$B_TP_STAT(R5)     ; REQUEST DMA ABORT
                         07F0          SETIND
          08 A0  01    A8  07FF          BISW    #DHULCT$M_ABORT,DHULCT(R0)
                         0803  15$:
                 01    BA  0803          POPR    #^M<R0>
                       05  0805          RSB
                         0806  ;
                         0806  ; RESUME PREVIOUSLY STOPPED PORT OUTPUT
                         0806  ;
                         0806  YF$RESUME:
```

```
                 0F    BB  0806          PUSHR   #^M<R0,R1,R2,R3>
                           0808          SETIND
      0D A0   80 8F    90  0817          MOVB    #^X80,DHUTBF2+1(R0)                    ; ENABLE TRANSMIT
     0134 C5     04    8A  081C          BICB    #UCB$M_XOFF, UCB$B_DHUFLG(R5)          ; clear to indicate xon
     33 64 A5    01    E0  0821          BBS     #UCB$V_INT,UCB$W_STS(R5),50$           ; Recalculate timeout
   2A 0108 C5    0B    E5  0826          BBCC    #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),40$ ; NO BURST IN PROGRESS
                           082C                                                         ; (RESET ANYWAY. WILL BE
                           082C                                                         ; SET IF NEEDED BY BURST_OUT
       51    0120 C5   3C  082C          MOVZWL  UCB$W_TT_OUTLEN(R5),R1                 ; GET NUMBER CHARACTERS
                           0831          TIMSET  R1,R1,LOCKOUTPUT                       ; COMPUTE TIMEOUT AND
                           0852                                                         ; SET INTERRUPT EXPECTED
             FBED CF   16  0852          JSB     BURST_OUTPUT                           ; RESTART OUTPUT
                           0856  40$:
                 0F    BA  0856          POPR    #^M<R0,R1,R2,R3>
                       05  0858          RSB
                           0859
                           0859  50$:
                           0859  ;
                           0859  ; reset timeout for the DHU/DHV
                           0859  ;
                           0859  ; We use the number of characters that the DHU has got left to send
                           0859  ; plus the number in the remaining part of the buffer
                           0859  ; Note that the byte count may not be updated by the DHU/DHV
                           0859  ; so we will over estimate the timeout value
                           0859  ; We have to add 64 because these characters may be in the Transmit fifo
       51    0040 8F   3C  0859          MOVZWL  #64,R1
         51    0E A0   A0  085E          ADDW    DHUTCT(R0),R1
       51    0120 C5   A0  0862          ADDW    UCB$W_TT_OUTLEN(R5),R1
                           0867          TIMSET  R1,R1,LOCKOUTPUT                       ; COMPUTE TIMEOUT AND
                           0888                                                         ; SET INTERUPT EXPTECTEDD
                 CC    11  0888          BRB     40$
                           088A
```

```
YFD
VAX

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
198
The
176
80

Mac
---
_$2
_$2
TOT

357

The

MAC
```

```
                        088A            .SBTTL  OUTPUT INTERRUPT SERVICE
                        088A    ;++
                        088A    ; YF$INTOUT - DHU OUTPUT INTERRUPT SERVICE
                        088A    ;
                        088A    ; FUNCTIONAL DESCRIPTION:
                        088A    ;
                        088A    ; THIS ROUTINE IS ENTERED WHEN THE DHU FINDS A LINE ENABLED
                        088A    ; AND AN EMPTY UART. THE CORRESPONDING UCB IS FOUND AND
                        088A    ; ANY OUTSTANDING PORT OUTPUT IS DONE. WHEN ALL OUTSTANDING PORT
                        088A    ; OUTPUT IS COMPLETED, THE CLASS DRIVER IS CALLED TO RETURN THE NEXT
                        088A    ; CHARACTER OR STRING TO BE OUTPUT. IF NO MORE OUTPUT IS FOUND, THEN
                        088A    ; THE LINE IS DISABLED.
                        088A    ;
                        088A    ; INPUTS:
                        088A    ;
                        088A    ;       SP(00) = ADDRESS OF THE IDB
                        088A    ;
                        088A    ; IMPLICIT INPUTS:
                        088A    ;
                        088A    ;       R0,R1,R2,R3,R4,R5 SAVED ON THE STACK.
                        088A    ;
                        088A    ; OUTPUTS:
                        088A    ;
                        088A    ;       THE INTERRUPT IS DISMISSED.
                        088A    ;
                        088A    ;--
                        088A    YF_OUT_EXIT:                            ; EXIT OUTPUT INTERRUPT
        5E    04    C0  088A            ADDL    #4,SP                   ; REMOVE IDB ADDRESS
        50    8E    7D  088D            MOVQ    (SP)+,R0                ; RESTORE REGISTERS
        52    8E    7D  0890            MOVQ    (SP)+,R2                ;
        54    8E    7D  0893            MOVQ    (SP)+,R4                ;
                    02  0896            REI                             ; DISMISS INTERRUPT
                        0897
                        0897    YF$INTOUT::                             ; DHU OUTPUT INTERRUPT SERVICE
                        0897
                        0897    YF_OUT_LOOP:
      54    00 BE   D0  0897            MOVL    @(SP),R4                ; GET THE IDB ADDRESS
        50    64    D0  089B            MOVL    (R4),R0                 ; GET THE CSR ADDRESS
                        089E    ;
                        089E    ; GET THE LINE INFO FROM THE CSR
                        089E    ;
                        089E
        52    60    B0  089E            MOVW    (R0),R2                 ; GET THE CSR VALUE
              E7    18  08A1            BGEQ    YF_OUT_EXIT
   51   52    F8 8F 78  08A3            ASHL    #-8,R2,R1               ; Get the line number
  51   FFFFFFF0 8F  CA  08A8            BICL    #^C<15>,R1
     55    18 A441  D0  08AF            MOVL    IDB$L_UCBLST(R4)[R1],R5 ; GET THE UCB ADDRESS
              E1    13  08B4            BEQL    YF_OUT_LOOP             ; IF EQL THEN DISMISS
                        08B6    ;
                        08B6    ;       CHECK FOR BURST OR DMA ACTIVE ON LINE
                        08B6    ;
                        08B6            SETIND  R0
      08 A0   01    B3  08BD            BITW    #DHULCT$M_ABORT, DHULCT(R0)     ; CHECK TO SEE IF ABORT IS SET
              04    13  08C1            BEQL    5$                              ; EQUAL, ABORT NOT SET
      08 A0   01    AA  08C3            BICW    #DHULCT$M_ABORT, DHULCT(R0)     ; CLEAR ABORT
              08    91  08C7    5$:     CMPB    #TTY$M_TANK_BURST@-8,-  ; ONLY BURST ACTIVE?
         0109 C5        08C9                    UCB$W_TT_HOLD+1(R5)
```

```
                  42    13  08CC  1494          BEQL    YF_SILO                 ; YES, CONTINUE SILO OUTPUT
            0100 8F     B3  08CE  1495          BITW    #TTY$M_TANK_PREMPT,-    ; Preempt required ?
            0108 C5         08D2  1496                  UCB$W_TT_HOLD(R5)
                  2C    12  08D5  1497          BNEQ    40$                     ; == BRW YF_PREEMPT
                            08D7  1498 9$:
            1000 8F     B3  08D7  1499          BITW    #TTY$M_TANK_DMA,-       ; DMA ACTIVE?
            0108 C5         08DB  1500                  UCB$W_TT_HOLD(R5)
                  26    12  08DE  1501          BNEQ    45$                 ; YES, PROCESS IT. (== BRW YF_DMA_INTERRUPT)
                            08E0  1502 ;
                            08E0  1503 ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
                            08E0  1504 ;
         64 A5    03    8A  08E0  1505 10$:     BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
                            08E4  1506 ;
                            08E4  1507 ; CALL CLASS DRIVER FOR MORE OUTPUT
                            08E4  1508 ;
            010C D5     16  08E4  1509          JSB     @UCB$L_TT_GETNXT(R5)    ; GET THE NEXT CHARACTER
                  1F    19  08E8  1510          BLSS    YF_START_BURST          ; BURST SPECIFIED
                  AB    13  08EA  1511          BEQL    YF_OUT_LOOP             ; NONE
                            08EC  1512 ;
                            08EC  1513 ; OUTPUT A CHARACTER TO THE DHU/DHV
                            08EC  1514 ;
                            08EC  1515 20$:
     0A 0134 C5   01    E0  08EC  1516          BBS     #UCB$V_DHU,UCB$B_DHUFLG(R5),30$
                            08F2  1517 ;
                            08F2  1518 ; Single char output to DHV
                            08F2  1519 ;
 02 A0    53    8000 8F     A9  08F2  1520          BISW3   #^X8000,R3,DHUTXC(R0)
                FF9B    31  08F9  1521          BRW     YF_OUT_LOOP
                            08FC  1522 ;
                            08FC  1523 ; Fifo output to DHU
                            08FC  1524 ;
                            08FC  1525 30$:
         06 A0    53    90  08FC  1526          MOVB    R3,DHUTXF(R0)
                            0900  1527 35$:
                FF94    31  0900  1528          BRW     YF_OUT_LOOP
                            0903  1529 40$:
                0089    31  0903  1530          BRW     YF_PREEMPT
                            0906  1531 45$:
                00A8    31  0906  1532          BRW     YF_DMA_INTERRUPT
                            0909  1533
                            0909  1534
```

```
                              0909
                              0909  YF_START_BURST:
           FB36 CF    16      0909          JSB     BURST_OUTPUT                ; START OUTPUT SILO OR DMA
               FF87   31      090D          BRW     YF_OUT_LOOP
                              0910  ;
                              0910  ;  CONTINUE SILO OUTPUT
                              0910  ;
                              0910  YF_SILO:
                              0910  ;
                              0910  ; Note. THE DHV does not have SILO output
                              0910  ;
     09 0134 C5    01   E1    0910          BBC     #UCB$V_DHU,UCB$B_DHUFLG(R5),5$
       40 8F    06 A0   91    0916          CMPB    DHUTFS(R0),#64              ; ANY SILO OUTPUT IN PROGRESS ?
                   22   12    091B          BNEQ    15$                         ; YES THEN LET IT COMPLETE
                   26   11    091D          BRB     20$                         ; OTHERWISE CONTINUE SILO OUTPUT
                              091F  ;
                              091F  ; DHV single char mode
                              091F  ;
                              091F  5$:
          53   011C D5  9A    091F          MOVZBL  @UCB$L_TT_OUTADR(R5),R3
   02 A0  53   8000 8F  A9    0924          BISW3   #^X8000,R3,DHUTXC(R0)
               011C C5  D6    092B          INCL    UCB$L_TT_OUTADR(R5)
               0120 C5  B7    092F          DECW    UCB$W_TT_OUTLEN(R5)
                   07   12    0933          BNEQ    10$                         ; NOT DONE
               0800 8F  AA    0935          BICW    #TTY$M_TANK_BURST,-         ; RESET BURST ACTIVE
               0108 C5        0939                  UCB$W_TT_HOLD(R5)
                              093C
                 FF58   31    093C  10$:    BRW     YF_OUT_LOOP
                              093F
                              093F  15$:
            F6F5 CF     D6    093F          INCL    YF$L_SIL_ERROR              ; INCREMENT ERROR COUNTER
                   F7   11    0943          BRB     10$                         ; == BRW YF_OUT_LOOP
                              0945  ;
                              0945  ; DHU silo output
                              0945  ;
                              0945  20$:
           51   06 A0   9A    0945          MOVZBL  DHUTFS(R0),R1               ; number of slots
                              0949
          52   0120 C5  3C    0949          MOVZWL  UCB$W_TT_OUTLEN(R5),R2      ; GET CURRENT LENGTH
          53   011C C5  D0    094E          MOVL    UCB$L_TT_OUTADR(R5),R3      ; GET CURRENT ADDRESS
               51   52  B1    0953          CMPW    R2,R1                       ; BURST LARGER THAN SILO?
                    03  1B    0956          BLEQU   50$                         ; NO
               52   51  9A    0958          MOVZBL  R1,R2                       ; MAXIMUM
                              095B  50$:
          011C C5   52  C0    095B          ADDL    R2,UCB$L_TT_OUTADR(R5)      ; UPDATE POINTER
          0120 C5   52  A2    0960          SUBW    R2,UCB$W_TT_OUTLEN(R5)      ; AND COUNT
                    07  12    0965          BNEQ    60$                         ; NOT DONE
               0800 8F  AA    0967          BICW    #TTY$M_TANK_BURST,-         ; RESET BURST ACTIVE
               0108 C5        096B                  UCB$W_TT_HOLD(R5)
                    52  D5    096E  60$:    TSTL    R2                          ; ANY ROOM AT ALL
                    1A  13    0970          BEQL    80$                         ; NO THEN EXIT
                              0972
               08   52  E9    0972          BLBC    R2,70$                      ; EVEN TRANSFER
            06 A0   83  90    0975          MOVB    (R3)+,DHUTXF(R0)            ; OUTPUT ODD BYTE
                    52  D7    0979          DECL    R2                          ; UPDATE COUNT
                    0F  13    097B          BEQL    80$                         ; DONE
                              097D  70$:
```

```
        52   52    FF 8F     78  097D           ASHL    #-1,R2,R2               ; CONVERT TO WORD COUNT
                                 0982   75$:
           06 A0      83     B0  0982           MOVW    (R3)+,DHUTXF(R0)
                                 0986           DELAY                           ; TO RELEASE THE UNIBUS
                  F6 52      F5  0989           SOBGTR  R2,75$                  ; LOOP TILL DONE
                                 098C
                   FF08      31  098C   80$:    BRW     YF_OUT_LOOP
                                 098F
                                 098F   YF_PREEMPT:
  0108 C5      0100 8F       AA  098F           BICW    #TTY$M_TANK_PREMPT, UCB$W_TT_HOLD(R5)     ; CLEAR PREEMPT BIT
       53      010A C5       9A  0996           MOVZBL  UCB$B_TT_PREMPT(R5),R3  ; GET CHARACTER
    09 0134 C5        01     E0  099B           BBS     #UCB$V_DHU,UCB$B_DHUFLG(R5),10$
                                 09A1   ;
                                 09A1   ; DHV single character
  02 A0    53    8000 8F     A9  09A1           BISW3   #^X8000,R3,DHUTXC(R0)
                      04     11  09A8           BRB     20$
                                 09AA   ;
                                 09AA   ; DHU FIFO output
                                 09AA   ;
                                 09AA   10$:
           06 A0      53     90  09AA           MOVB    R3,DHUTYF(R0)
                                 09AE   20$:
                   FEE6      31  09AE           BRW     YF_OUT_LOOP
                                 09B1
                                 09B1   YF_DMA_INTERRUPT:
                                 09B1   ;
                                 09B1   ; CHECK TO MAKE SURE NO DATA IS PENDING BEFORE ASKING FOR MORE
                                 09B1   ;
               FEE2 CF       DF  09B1           PUSHAL  YF_OUT_LOOP             ; BUILD RETURN ADDRESS ON STACK
            04 52    0C      E1  09B5           BBC     #DHUCSR$V_DMAERR,R2,5$  ; CHECK FOR A DMA ERROR
               F683 CF       D6  09B9           INCL    YF$L_DMAXMT_ERROR       ; ERROR OCCURED INCREMENT COUNTS
                                 09BD   5$:
                      06     93  09BD           BITB    #TTY$M_TP_ALLOC!TTY$M_TP_DLLOC,- ;CHECK FOR FORKS ACTIVE
                 0130 C5         09BF                   UCB$B_TP_STAT(R5)       ; AND IGNORE IF SO
                      1C     12  09C2           BNEQ    20$
         0130 C5      00     E0  09C4           BBS     #TTY$V_TP_ABORT, UCB$B_TP_STAT(R5),-
                      10         09C9                   10$                     ; ABORT ACTIVE DMA
                   0C A0     95  09CA           TSTB    DHUTBF2(R0)             ; ANY DMA IN PROGRESS ?
                      12     19  09CD           BLSS    30$                     ; YES, then let it complete
                   0E A0     B5  09CF           TSTW    DHUTCT(R0)              ; TEST DMA BYTE COUNT
                      06     13  09D2           BEQL    10$                     ; DMA BYTE COUNT DONE
               F664 CF       D6  09D4           INCL    YF$L_ERROR              ;
                      06     11  09D8           BRB     20$                     ; NO THEN CONTINUE
                                 09DA   10$:
                   0E A0     B4  09DA           CLRW    DHUTCT(R0)              ; CLEAR DMA BYTE COUNT (ABORT WAS CLEARED)
                    FB7D     31  09DD           BRW     DMA_CONTINUE            ; OTHERWISE, CONTINUE THE DMA
                                 09E0
                                 09E0                                           ; IF THIS INTERRUPT WAS THE RESULT
                                 09E0                                           ; OF AN ABORT, THIS WILL BE HANDLED
                                 09E0                                           ; BY DMA_CONTINUE
                                 09E0   20$:
                             05  09E0           RSB
                                 09E1
                                 09E1
                                 09E1   30$:
               F653 CF       D6  09E1           INCL    YF$L_SIL_ERROR
                 5E   04     C0  09E5           ADDL    #4, SP                  ; Pop off return address
```

```
                FEAC    31  09E8  1650          BRW     YF_OUT_LOOP
                            09EB  1651
```

```
_$2

Pse
----

$ADI

$DA

$GL(

$OWI

DIGI

REAI

SEEI

SYS

SYSI

$COI

$COI
```

```
                             09EB          .SBTTL  SET SPEED, PARITY PARAMETERS
                             09EB
                             09EB    ;++
                             09EB    ; YF$SET_LINE - RESET SPEED, PARITY
                             09EB    ;
                             09EB    ; FUNCTIONAL DESCRIPTION:
                             09EB    ;
                             09EB    ; INPUTS:
                             09EB    ;
                             09EB    ;         R5 - UCB ADDRESS
                             09EB    ;
                             09EB    ; OUTPUTS:
                             09EB    ;
                             09EB    ;         R4 USED
                             09EB    ;--
                             09EB
                             09EB    YF$SET_LINE:
                 0C     BB   09EB          PUSHR   #^M<R2,R3>
        54    24 A5     D0   09ED          MOVL    UCB$L_CRB(R5),R4           ; ADDRESS CRB
        54    2C B4     D0   09F1          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4     ; GET THE CSR ADDRESS VIA CRB
                             09F5          SETIND
        52    08 A0     3C   0A04          MOVZWL  DHULCT(R0),R2                       ; Fetch the line control reg
                             0A08
                             0A08    ;
                             0A08    ; The DHU/DHV have automatic detection of received XON/XOFF and also
                             0A08    ; automatic generation of XON/XOFF options
                             0A08    ;
   14 0122 C5    05     E1   0A08          BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),4$; AUTOXON XOFF AVAILABLE ON T
                             0A0E                                                  ; YES THEN IS
                             0A0E    ;
                             0A0E    ; Assume that both modes are required
                             0A0E    ;
                 12     A8   0A0E          BISW    #<DHULCT$M_OAUTO ! DHULCT$M_IAUTO>,-
                 52          0A10                  R2; Assume AUTOXOFF
                             0A11    ;
                             0A11    ; Disable detection of received XON/XOFF if not allowed
                             0A11    ;
   03 0122 C5    06     E0   0A11          BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),2$; AUTOXON XOFF ENABLED
          52    02      AA   0A17          BICW    #DHULCT$M_OAUTO,R2                  ; NO THEN CLEAR THE AUTOXOFF ENABLE
                             0A1A    2$:
                             0A1A    ;
                             0A1A    ; Disable sending of XON/XOFF is hostsync is not set
                             0A1A    ; (The DHU/DHV can send XON/XOFF automatically if the receive fifo fills up)
                             0A1A    ;
     03 44 A5    04     E0   0A1A          BBS     #TT$V_HOSTSYNC,UCB$L_DEVDEPEND(R5),4$; Host sync specified ?
          52    10      AA   0A1F          BICW    #DHULCT$M_IAUTO,R2                  ; No, then clear enable flag
                             0A22
     52   0100 8F       AA   0A22    4$:   BICW    #DHULCT$M_MODEM,R2
     05 44 A5    15     E1   0A27          BBC     #TT$V_MODEM, UCB$L_DEVDEPEND(R5), 6$
     52   0100 8F       A8   0A2C          BISW    #DHULCT$M_MODEM,R2
                             0A31
                             0A31    ;
                             0A31    ; move updated register back into device
                             0A31    ;
       08 A0     52     B0   0A31    6$:   MOVW    R2,DHULCT(R0)
                 7E     D4   0A35          CLRL    -(SP)                               ; RESET A TEMPORARY LOCATION
                             0A37    ;
```

_$2
Psec

$COD
$PLI
CLI
_CN
_CN
_LIE
_LIE
_LIE
_LIE
MSG
MSG
MSG
MSG

```
                              0A37     ;       SET UP LINE SPEED AND PARITY
                              0A37     ;
                              0A37
             00F5 C5     95   0A37             TSTB    UCB$W_TT_SPEED+1(R5)     ; RECEIVE SPEED SPECIFIED?
                  07     12   0A3B             BNEQ    8$                       ; YES
             00F4 C5     90   0A3D             MOVB    UCB$W_TT_SPEED(R5),-     ; NO, SO USE TRANSMITTER SPEED
             00F5 C5          0A41                     UCB$W_TT_SPEED+1(R5)
                              0A44     8$:
          53 00F4 C5     9A   0A44             MOVZBL  UCB$W_TT_SPEED(R5),R3            ; Get TRANSMIT SPEED
       53   F637 CF43    90   0A49             MOVB    YF$VMS_SPEEDS[R3],R3
                  19     19   0A4F             BLSS    10$                              ; Illegal speed
    6E    04    0C   53  F0   0A51             INSV    R3,#DHULPR$V_TSPEED,#4,(SP)      ; SET TRANSMIT SPEED
          53 00F5 C5     9A   0A56             MOVZBL  UCB$W_TT_SPEED+1(R5),R3
       53   F625 CF43    90   0A5B             MOVB    YF$VMS_SPEEDS[R3],R3
                  07     19   0A61             BLSS    10$                              ; Illegal speed
    6E    04    08   53  F0   0A63             INSV    R3,#DHULPR$V_RSPEED,#4,(SP)      ; SET RECEIVE SPEED
                  27     11   0A68             BRB     20$                ; set current speed
                              0A6A
                              0A6A     10$:
                              0A6A     ;
                              0A6A     ; This code restores the speed to its previous value
                              0A6A     ; !` is entered when an illegal speed combination is detected.
                              0A6A
            53    04 A0  B0   0A6A             MOVW    DHULPR(R0),R3            ; get line parameters
               6E    53  B0   0A6E             MOVW    R3,(SP)                  ; use previous speed as new
                     6E  94   0A71             CLRB    (SP)
       53    53    04 0C EF   0A73             EXTZV   #DHULPR$V_TSPEED,#4,R3,R3        ; extract speed
   00F4 C5    F618 CF43  90   0A78             MOVB    YF$DHU_SPEEDS[R3],UCB$W_TT_SPEED(R5); convert to VMS value
                              0A80
            53    04 A0  B0   0A80             MOVW    DHULPR(R0),R3                    ; get line parameters
       53    53    04 08 EF   0A84             EXTZV   #DHULPR$V_RSPEED,#4,R3,R3        ; extract speed
   00F5 C5    F607 CF43  90   0A89             MOVB    YF$DHU_SPEEDS[R3],UCB$W_TT_SPEED+1(R5); convert to VMS value
                              0A91
                              0A91     ; insert other parameters in the new LPR value
                              0A91     ;
                              0A91     20$:
                              0A91
  53    00F8 C5    02    03  EE   0A91             EXTV    #UCB$V_TT_LEN,#2,UCB$B_TT_PARITY(R5),R3          ; GET CHAR SIZE
        6E   02    03    53  F0   0A98             INSV    R3,#DHULPR$V_SIZE,#2,(SP)                        ; SET IT
                              0A9D
  53    00F8 C5    02    06  EE   0A9D             EXTV    #UCB$V_TT_PARTY,#2,UCB$B_TT_PARITY(R5),R3        ; GET PARITY/ODD
        6E   02    05    53  F0   0AA4             INSV    R3,#DHULPR$V_PARITY,#2,(SP)
         6E    00000040  8F  CC   0AA9             XORL    #DHULPR$M_ODD,(SP)                               ; Correct sense
                              0AB0
  53    00F8 C5    01    05  EE   0AB0             EXTV    #UCB$V_TT_STOP,#1,UCB$B_TT_PARITY(R5),R3         ; GET STOP
        6E   01    07    53  F0   0AB7             INSV    R3,#DHULPR$V_STOP,#1,(SP)
                              0ABC
            04 A0    6E  B1   0ABC             CMPW    (SP),DHULPR(R0)                          ; Any modifications to be made?
                     04  13   0AC0             BEQL    30$                                      ; EQL, no then return
            04 A0    6E  F7   0AC2             CVTLW   (SP),DHULPR(R0)                          ; INSERT AS LINE PARAMETER
               5E    04  C0   0AC6     30$:    ADDL    #4, SP                                   ; Restore stack
                     0C  BA   0AC9             POPR    #^M<R2,R3>                               ; Restore registers
                         05   0ACB             RSB
                              0ACC
                              0ACC
                              0ACC
                              0ACC     YF$END:                                  ; end of driver
```

```
                                          0ACC  1767          .END
```

```
_$25

Symb
----
FOUN
GENE
GETC
GETD
GETD
GETF
GETF
GETF
GETF
GETF
GETS
GETU
GETV
GET_
GRP
HELF
HOLD
IDEN
LBR$
LETT
LGI$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$
LIB$

LIB$
LIB$
LIB$
LIB$
LIST
LIST
MATC
MATC
MEM_
MOD
MTH
NAF
NETC
NETC
NSA
NSA
NSA
OUT
PAR
```

```
YFDRIVER                          - Port Driver for DHU/DHV                     16-SEP-1984 02:26:48  VAX/VMS Macro V04-00          Page  42
Symbol table                                                                     5-SEP-1984 04:17:43  [TTDRVR.SRC]YFDRIVER.MAR;1          (1)

$$$                              = 00000020 R     02        DHULCT$M_MAINT                   = 000000C0
$$OP                             = 00000002                 DHULCT$M_MODEM                   = 00000100
AT$_UBA                          = 00000001                 DHULCT$M_OAUTO                   = 00000002
BIT...                           = 00000003                 DHULCT$M_RCV                     = 00000004
BUG$_UNSUPRTCPU                    ********   X   03        DHULCT$M_RTS                     = 00001000
BURST_OUTPUT                       00000443 R     03        DHULCT$M_SNDOFF                  = 00000020
CLASS_DDT                        = 00000010                 DHULCT$S_ABORT                   = 00000001
CLASS_DS_TRAN                    = 0000000C                 DHULCT$S_BREAK                   = 00000001
CLASS_FORK                       = 0000001C                 DHULCT$S_DTR                     = 00000001
CLASS_GETNXT                     = 00000000                 DHULCT$S_IAUTO                   = 00000001
CLASS_POWERFAIL                  = 00000020                 DHULCT$S_MAINT                   = 00000002
CLASS_PUTNXT                     = 00000004                 DHULCT$S_MODEM                   = 00000001
CLASS_READERROR                  = 00000014                 DHULCT$S_OAUTO                   = 00000001
CLASS_SETUP_UCB                  = 00000008                 DHULCT$S_RCV                     = 00000001
CRB$B_TT_TYPE                    = 0000000B                 DHULCT$S_RTS                     = 00000001
CRB$L_INTD                       = 00000024                 DHULCT$S_SNDOFF                  = 00000001
CRB$L_INTD2                      = 00000048                 DHULCT$V_ABORT                   = 00000000
DC$_TERM                         = 00000042                 DHULCT$V_BREAK                   = 00000003
DDB$L_DDT                        = 0000000C                 DHULCT$V_DTR                     = 00000009
DEV$M_AVL                        = 00040000                 DHULCT$V_IAUTO                   = 00000004
DEV$M_CCL                        = 00000002                 DHULCT$V_MAINT                   = 00000006
DEV$M_IDV                        = 04000000                 DHULCT$V_MODEM                   = 00000008
DEV$M_NNM                        = 00000200                 DHULCT$V_OAUTO                   = 00000001
DEV$M_ODV                        = 08000000                 DHULCT$V_RCV                     = 00000002
DEV$M_REC                        = 00000001                 DHULCT$V_RTS                     = 0000000C
DEV$M_TRM                        = 00000004                 DHULCT$V_SNDOFF                  = 00000005
DHUCSR                           = 00000000                 DHULPR                           = 00000004
DHUCSR$C_BASE                    = 00004040                 DHULPR$M_DIAG1                   = 00000002
DHUCSR$M_CLEAR                   = 00000020                 DHULPR$M_DIAG2                   = 00000004
DHUCSR$M_DIGFAL                  = 00002000                 DHULPR$M_ODD                     = 00000040
DHUCSR$M_DMAERR                  = 00001000                 DHULPR$M_PARITY                  = 00000020
DHUCSR$M_IADDR                   = 0000000F                 DHULPR$M_RSPEED                  = 00000F00
DHUCSR$M_LINE                    = 00000F00                 DHULPR$M_SIZE                    = 00000018
DHUCSR$M_RCVINT                  = 00000040                 DHULPR$M_STOP                    = 00000080
DHUCSR$M_SNDINT                  = 00004000                 DHULPR$M_TSPEED                  = 0000F000
DHUCSR$M_SNDRDY                  = 00008000                 DHULPR$S_DIAG1                   = 00000001
DHUCSR$S_CLEAR                   = 00000001                 DHULPR$S_DIAG2                   = 00000001
DHUCSR$S_DIGFAL                  = 00000001                 DHULPR$S_ODD                     = 00000001
DHUCSR$S_DMAERR                  = 00000001                 DHULPR$S_PARITY                  = 00000001
DHUCSR$S_IADDR                   = 00000004                 DHULPR$S_RSPEED                  = 00000004
DHUCSR$S_LINE                    = 00000004                 DHULPR$S_SIZE                    = 00000002
DHUCSR$S_RCVINT                  = 00000001                 DHULPR$S_STOP                    = 00000001
DHUCSR$S_SNDINT                  = 00000001                 DHULPR$S_TSPEED                  = 00000004
DHUCSR$S_SNDRDY                  = 00000001                 DHULPR$V_DIAG1                   = 00000001
DHUCSR$V_CLEAR                   = 00000005                 DHULPR$V_DIAG2                   = 00000002
DHUCSR$V_DIGFAL                  = 0000000D                 DHULPR$V_ODD                     = 00000006
DHUCSR$V_DMAERR                  = 0000000C                 DHULPR$V_PARITY                  = 00000005
DHUCSR$V_IADDR                   = 00000000                 DHULPR$V_RSPEED                  = 00000008
DHUCSR$V_LINE                    = 00000008                 DHULPR$V_SIZE                    = 00000003
DHUCSR$V_RCVINT                  = 00000006                 DHULPR$V_STOP                    = 00000007
DHUCSR$V_SNDINT                  = 0000000E                 DHULPR$V_TSPEED                  = 0000000C
DHUCSR$V_SNDRDY                  = 0000000F                 DHURBF                           = 00000002
DHULCT                           = 00000008                 DHURCV$M_BUF                     = 000000FF
DHULCT$M_ABORT                   = 00000001                 DHURCV$M_FRAMER                  = 00002000
DHULCT$M_BREAK                   = 00000008                 DHURCV$M_LINE                    = 00000F00
DHULCT$M_DTR                     = 00000200                 DHURCV$M_OVERRUN                 = 00004000
DHULCT$M_IAUTO                   = 00000010                 DHURCV$M_PARERR                  = 00001000
```

```
_$2

Sym
---
PAR
PAR
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PLI
PRO
PRO
PRV
PRV
PRV
PRV
PRV
PRV
PUR
PWD
R
RAB
RAN
RAN
RDB
RDB
RDB
REC
REC
REC
REM
REM
REN
RES
RES
RIG
RMS
SET
SET
SHO
SHO
SIG
SS$
SS$
SS$
SS$
STR
STR
```

```
YFDRIVER                          - Port Driver for DHU/DHV                16-SEP-1984 02:26:48  VAX/VMS Macro V04-00          Page  43
Symbol table                                                                5-SEP-1984 04:17:43  [TTDRVR.SRC]YFDRIVER.MAR;1          (1)

DHURCV$M_VALID                    = 00008000               DPT$C_LENGTH                   = 00000038
DHURCV$S_BUF                      = 00000008               DPT$C_VERSION                  = 00000004
DHURCV$S_FRAMER                   = 00000001               DPT$INITAB                       00000038 R       02
DHURCV$S_LINE                     = 00000004               DPT$M_NOUNLOAD                 = 00000004
DHURCV$S_OVERRUN                  = 00000001               DPT$REINITAB                     000000D3 R       02
DHURCV$S_PARERR                   = 00000001               DPT$TAB                          00000000 R       02
DHURCV$S_VALID                    = 00000001               DPT$W_VECTOR                   = 0000001E
DHURCV$V_BUF                      = 00000000               DT$_DHU                        = 00000047
DHURCV$V_FRAMER                   = 0000000D               DT$_DHV                        = 00000046
DHURCV$V_LINE                     = 00000008               DYN$C_CRB                      = 00000005
DHURCV$V_OVERRUN                  = 0000000E               DYN$C_DDB                      = 00000006
DHURCV$V_PARERR                   = 0000000C               DYN$C_DPT                      = 0000001E
DHURCV$V_VALID                    = 0000000F               DYN$C_ORB                      = 00000049
DHUSPD$C_BAUD_110                 = 00000002               DYN$C_UCB                      = 00000010
DHUSPD$C_BAUD_1200                = 00000007               EXE$GB_CPUTYPE                   ********    X   03
DHUSPD$C_BAUD_134                 = 00000003               EXE$GL_ABSTIM                    ********    X   03
DHUSPD$C_BAUD_150                 = 00000004               EXE$GL_TENUSEC                   ********    X   03
DHUSPD$C_BAUD_1800                = 00000008               EXE$GL_UBDELAY                   ********    X   03
DHUSPD$C_BAUD_19200               = 0000000E               FUNCTAB_LEN                    = 00000000
DHUSPD$C_BAUD_2000                = 00000009               IDB$L_UCBLST                   = 00000018
DHUSPD$C_BAUD_2400                = 0000000A               INIT_CONTINUE                    00000235 R       03
DHUSPD$C_BAUD_300                 = 00000005               IO$M_AUTXOF_DIS                = 00004000
DHUSPD$C_BAUD_38400               = 0000000F               IO$M_AUTXOF_ENA                = 00002000
DHUSPD$C_BAUD_4800                = 0000000B               IO$M_LINE_OFF                  = 00000200
DHUSPD$C_BAUD_50                  = 00000000               IO$M_LINE_ON                   = 00000800
DHUSPD$C_BAUD_600                 = 00000006               IO$M_LOOP                      = 00000080
DHUSPD$C_BAUD_7200                = 0000000C               IO$M_UNLOOP                    = 00000100
DHUSPD$C_BAUD_75                  = 00000001               IOC$ALOUBAMAPN                   ********    X   03
DHUSPD$C_BAUD_9600                = 0000000D               IOC$MNTVER                       ********    X   03
DHUSTT                            = 00000007               IOC$RELMAPREG                    ********    X   03
DHUSTT$M_CTS                      = 00000008               IOC$RETURN                       ********    X   03
DHUSTT$M_DCD                      = 00000010               MMG$GL_SPTBASE                   ********    X   03
DHUSTT$M_DSR                      = 00000080               MODEM$C_DATASET                = 00000003
DHUSTT$M_RI                       = 00000020               MODEM$C_INIT                   = 00000000
DHUSTT$S_CTS                      = 00000001               ORB$B_FLAGS                    = 0000000B
DHUSTT$S_DCD                      = 00000001               ORB$L_OWNER                    = 00000000
DHUSTT$S_DSR                      = 00000001               ORB$M_PROT_16                  = 00000001
DHUSTT$S_RI                       = 00000001               ORB$W_PROT                     = 00000018
DHUSTT$V_CTS                      = 00000003               PORT_ABORT                     = 00000020
DHUSTT$V_DCD                      = 00000004               PORT_DS_SET                    = 0000000C
DHUSTT$V_DSR                      = 00000007               PORT_FORKRET                   = 00000034
DHUSTT$V_RI                       = 00000005               PORT_LENGTH                    = 00000038
DHUTBF1                           = 0000000A               PORT_MAINT                     = 00000030
DHUTBF2                           = 0000000C               PORT_RESUME                    = 00000024
DHUTCR                            = 00000002               PORT_SET_LINE                  = 00000008
DHUTCT                            = 0000000E               PORT_STARTIO                   = 00000000
DHUTFS                            = 00000006               PORT_STOP                      = 00000018
DHUTXC                            = 00000002               PORT_VECTOR                      00000048 R       03
DHUTXF                            = 00000006               PORT_XOFF                      = 00000014
DMA_ALLOC                           00000522 R       03    PORT_XON                       = 00000010
DMA_CONTINUE                        0000055D R       03    PR$_IPL                        = 00000012
DMA_DEALLOC                         000006E8 R       03    PR$_SID_TYP780                 = 00000001
DMA_DONE                            000006D4 R       03    SILO_OUTPUT                      00000469 R       03
DMA_MAP                             0000061D R       03    SIZ...                         = 00000001
DMA_NOMAP                           0000057D R       03    SS$_NORMAL                     = 00000001
DMA_POST                            00000708 R       03    TT$C_BAUD_110                  = 00000003
DMA_START                           000004F1 R       03    TT$C_BAUD_1200                 = 00000008
```

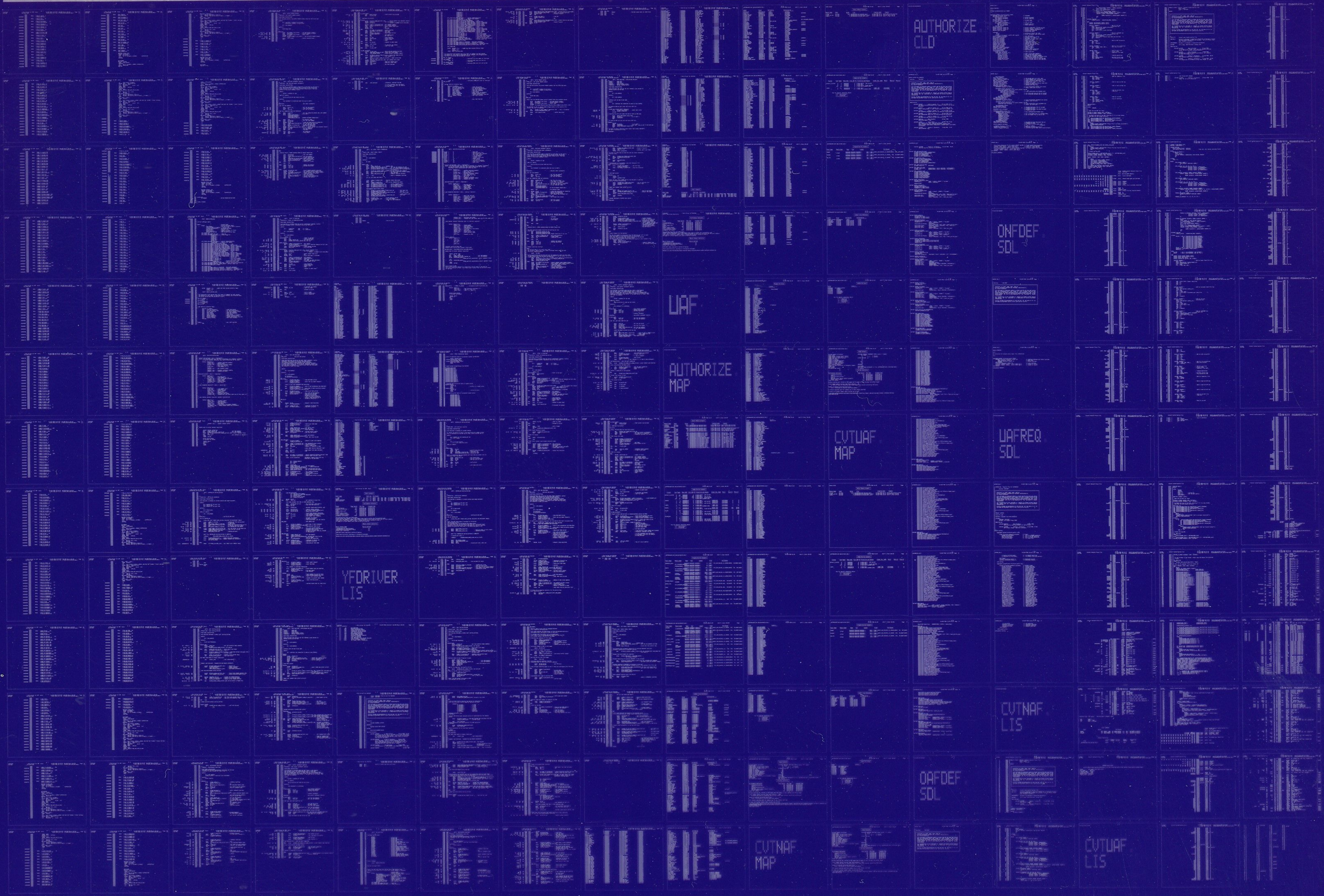
```
                                                           C 9
YFDRIVER                        - Port Driver for DHU/DHV               16-SEP-1984 02:26:48  VAX/VMS Macro V04-00        Page  44
Symbol table                                                             5-SEP-1984 04:17:43  [TTDRVR.SRC]YFDRIVER.MAR;1        (1)

TT$C_BAUD_134                   = 00000004                      UCB$B_TT_DS_RCV                 = 00000124
TT$C_BAUD_150                   = 00000005                      UCB$B_TT_DS_TX                  = 00000125
TT$C_BAUD_1800                  = 00000009                      UCB$B_TT_MAINT                  = 0000012A
TT$C_BAUD_19200                 = 00000010                      UCB$B_TT_PARITY                 = 000000F8
TT$C_BAUD_2000                  = 0000000A                      UCB$B_TT_PREMPT                 = 0000010A
TT$C_BAUD_2400                  = 0000000B                      UCB$C_TT_LENGTH                 = 00000134
TT$C_BAUD_300                   = 00000006                      UCB$L_CRB                       = 00000024
TT$C_BAUD_4800                  = 0000000D                      UCB$L_DDB                       = 00000028
TT$C_BAUD_600                   = 00000007                      UCB$L_DDT                       = 00000088
TT$C_BAUD_75                    = 00000002                      UCB$L_DEVCHAR                   = 00000038
TT$C_BAUD_9600                  = 0000000F                      UCB$L_DEVCHAR2                  = 0000003C
TT$M_DS_CARRIER                 = 00000020                      UCB$L_DEVDEPEND                 = 00000044
TT$M_DS_CTS                     = 00000010                      UCB$L_DEVDEPND2                 = 00000048
TT$M_DS_RING                    = 00000040                      UCB$L_DUETIM                    = 0000006C
TT$V_DS_DSR                     = 00000007                      UCB$L_TP_MAP                    = 0000012C
TT$V_HOSTSYNC                   = 00000004                      UCB$L_TT_CLASS                  = 00000114
TT$V_MODEM                      = 00000015                      UCB$L_TT_DECHA1                 = 000000C8
TT$_UNKNOWN                     = 00000000                      UCB$L_TT_DECHAR                 = 000000C4
TTY$GB_DEFSPEED                   ********    X    02           UCB$L_TT_GETNXT                 = 0000010C
TTY$GB_PARITY                     ********    X    02           UCB$L_TT_OUTADR                 = 0000011C
TTY$GB_RSPEED                     ********    X    02           UCB$L_TT_PORT                   = 00000118
TTY$GB_SILOTIME                   ********    X    03           UCB$L_TT_PUTNXT                 = 00000110
TTY$GL_DEFCHAR                    ********    X    02           UCB$L_TT_RTIMOU                 = 000000B4
TTY$GL_DEFCHAR2                   ********    X    02           UCB$L_TT_WBLINK                 = 000000D0
TTY$GL_DPT                        ********    X    03           UCB$L_TT_WFLINK                 = 000000CC
TTY$GL_OWNUIC                     ********    X    02           UCB$M_DHU                       = 00000002
TTY$GW_DEFBUF                     ********    X    02           UCB$M_INT                       = 00000002
TTY$GW_DMASIZE                    ********    X    03           UCB$M_MAP                       = 00000001
TTY$GW_PROT                       ********    X    02           UCB$M_ONLINE                    = 00000010
TTY$M_PC_DMAAVL                 = 00000004                      UCB$M_TIM                       = 00000001
TTY$M_PC_MAPAVL                 = 00000010                      UCB$M_XOFF                      = 00000004
TTY$M_PC_XOFAVL                 = 00000020                      UCB$S_DHU                       = 00000001
TTY$M_TANK_BURST                = 00000800                      UCB$S_MAP                       = 00000001
TTY$M_TANK_DMA                  = 00001000                      UCB$S_XOFF                      = 00000001
TTY$M_TANK_PREMPT               = 00000100                      UCB$V_DHU                       = 00000001
TTY$M_TP_ABORT                  = 00000001                      UCB$V_INT                       = 00000001
TTY$M_TP_ALLOC                  = 00000002                      UCB$V_MAP                       = 00000000
TTY$M_TP_DLLOC                  = 00000004                      UCB$V_ONLINE                    = 00000004
TTY$V_PC_DMAENA                 = 00000001                      UCB$V_POWER                     = 00000005
TTY$V_PC_MAPAVL                 = 00000004                      UCB$V_TT_DSBL                   = 00000007
TTY$V_PC_NOTIME                 = 00000000                      UCB$V_TT_LEN                    = 00000003
TTY$V_PC_PRMMAP                 = 00000003                      UCB$V_TT_PARTY                  = 00000006
TTY$V_PC_XOFAVL                 = 00000005                      UCB$V_TT_STOP                   = 00000005
TTY$V_PC_XOFENA                 = 00000006                      UCB$V_XOFF                      = 00000002
TTY$V_TANK_BURST                = 0000000B                      UCB$W_DEVBUFSIZ                 = 00000042
TTY$V_TP_ABORT                  = 00000000                      UCB$W_STS                       = 00000064
TTY$V_TP_ALLOC                  = 00000001                      UCB$W_TT_DESIZE                 = 000000F1
TTY$V_TP_DLLOC                  = 00000002                      UCB$W_TT_DESPEE                 = 000000E8
UBA$L_MAP                       = 00000800                      UCB$W_TT_HOLD                   = 00000108
UCB$B_DEVCLASS                  = 00000040                      UCB$W_TT_OUTLEN                 = 00000120
UCB$B_DEVTYPE                   = 00000041                      UCB$W_TT_PRTCTL                 = 00000122
UCB$B_DHUFLG                    = 00000134                      UCB$W_TT_SPEED                  = 000000F4
UCB$B_DIPL                      = 0000005E                      UCB$W_TT_UNITBIT                = 00000106
UCB$B_FIPL                      = 0000000B                      UCB$W_UNIT                      = 00000054
UCB$B_TP_STAT                   = 00000130                      VEC$L_ADP                       = 00000014
UCB$B_TT_DEPARI                 = 000000EC                      VEC$L_IDB                       = 00000008
UCB$B_TT_DETYPE                 = 000000F0                      VEC$L_INITIAL                   = 0000000C
```

```
YFDRIVER                      - Port Driver for DHU/DHV          16-SEP-1984 02:26:48  VAX/VMS Macro V04-00        Page  45
Symbol table                                                      5-SEP-1984 04:17:43  [TTDRVR.SRC]YFDRIVER.MAR;1        (1)

VEC$L_UNITINIT                = 00000018
VEC$S_MAPREG                  = 0000000F
VEC$V_MAPREG                  = 00000000
VEC$W_MAPREG                  = 00000010
YF$ABORT                        000007E4 R      03
YF$CTRL_ERROR                   00000155 R      03
YF$DDT                          00000000 RG     03
YF$DELIVER                      000000A4 RG     03
YF$DHU_SPEEDS                   00000095 R      03
YF$DPT                          00000000 RG     02
YF$DS_SET                       0000033C R      03
YF$END                          00000ACC R      03
YF$FORK                         0000071F R      03
YF$INITIAL                      000000C8 RG     03
YF$INITLINE                     00000156 RG     03
YF$INTINP                       0000036F RG     03
YF$INTOUT                       00000897 RG     03
YF$L_DMAXMT_ERROR               00000040 RG     03
YF$L_ERROR                      0000003C RG     03
YF$L_INACT_ERROR                00000044 RG     03
YF$L_SIL_ERROR                  00000038 RG     03
YF$MAINT                        0000029B R      03
YF$NULL                         00000084 R      03
YF$PREEMPT                      00000777 R      03
YF$RESUME                       00000806 R      03
YF$SET_LINE                     000009EB R      03
YF$STARTIO                      0000043E RG     03
YF$STOP                         000007C4 R      03
YF$UNIT_ERROR                   00000296 R      03
YF$VEC                          00000048 R      03
YF$VECEND                       00000080 R      03
YF$VMS_SPEEDS                   00000085 R      03
YF$XOFF                         0000075E R      03
YF$XON                          0000073E R      03
YF_DMA_INTERRUPT                000009B1 R      03
YF_INITMAP                      0000022E R      03
YF_INITNULL                     00000227 R      03
YF_NOMAP                        00000235 R      03
YF_OUT_EXIT                     0000088A R      03
YF_OUT_LOOP                     00000897 R      03
YF_PREEMPT                      0000098F R      03
YF_SILO                         00000910 R      03
YF_START_BURST                  00000909 R      03

                                                +----------------+
                                                ! Psect synopsis !
                                                +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$105_PROLOGUE                 000000E8  (  232.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$115_DRIVER                   00000ACC  ( 2764.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
```

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

Phase                        Page faults    CPU Time        Elapsed Time
-----                        -----------    --------        ------------
Initialization                        29    00:00:00.04     00:00:00.19
Command processing                   123    00:00:00.39     00:00:01.17
Pass 1                               754    00:00:23.82     00:00:27.56
Symbol table sort                      0    00:00:03.37     00:00:03.47
Pass 2                               296    00:00:04.93     00:00:05.45
Symbol table output                   44    00:00:00.23     00:00:00.45
Psect synopsis output                  2    00:00:00.01     00:00:00.01
Cross-reference output                 0    00:00:00.00     00:00:00.00
Assembler run totals                1250    00:00:32.79     00:00:38.30

The working set limit was 2550 pages.
198247 bytes (388 pages) of virtual memory were used to buffer the intermediate code.
There were 170 pages of symbol table space allocated to hold 3101 non-local and 128 local symbols.
1767 source lines were read in Pass 1, producing 24 object records in Pass 2.
80 pages of virtual memory were used to define 74 macros.

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

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                               35
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            12
TOTALS (all libraries)                                       47

3572 GETS were required to define 47 macros.

There were no errors, warnings or information messages.

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

0405 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



AUTHORIZE
CLD

ONFDEF
SDL

UAF

AUTHORIZE
MAP

CVTUAF
MAP

UAFREQ
SDL

YFDRIVER
LIS

CVTNAF
LIS

OAFDEF
SDL

CVTNAF
MAP

CVTUAF
LIS