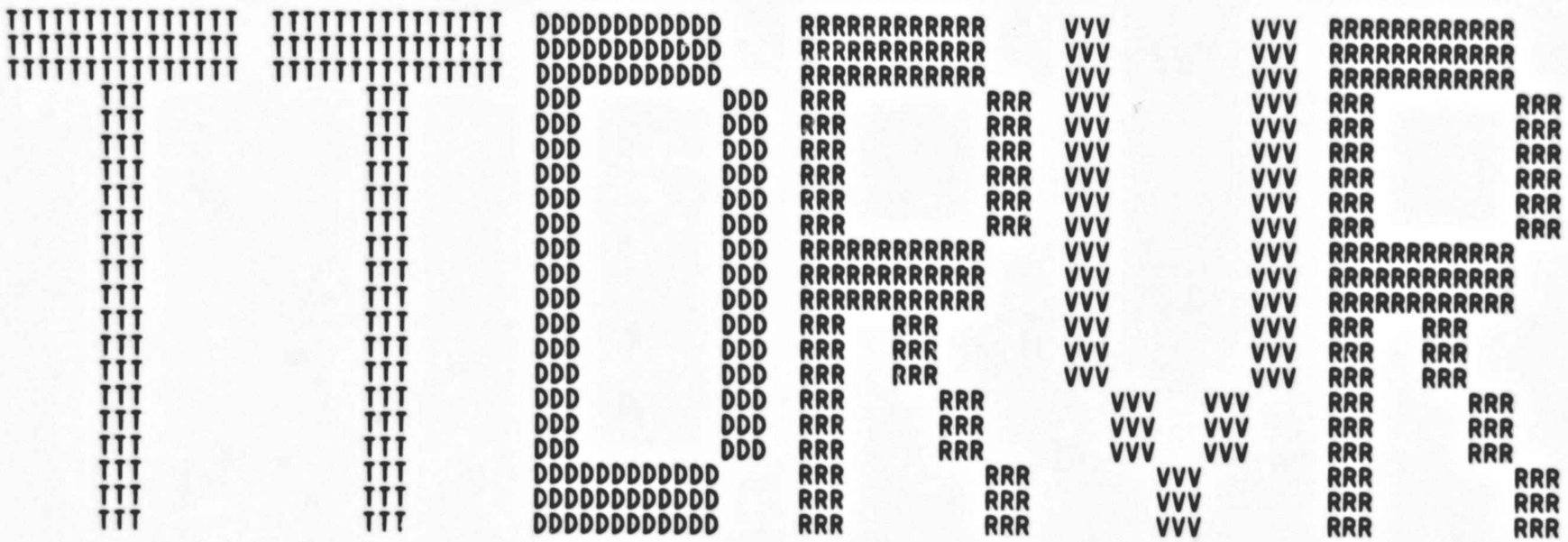
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR 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 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 DDDDDDDDDDDD RRR RRR VVV RRR RRR
TTT TTT DDDDDDDDDDDD RRR RRR VVV RRR RRR
TTT TTT DDDDDDDDDDDD RRR RRR VVV RRR RRR

FILEID**YCDRIVER

```
YY      YY    CCCCCCCC  DDDDDDDD    RRRRRRRR     IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
YY      YY    CCCCCCCC  DDDDDDDD    RRRRRRRR     IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
YY      YY  CC          DD      DD  RR      RR     II      VV      VV  EE          RR      RR
YY      YY  CC          DD      DD  RR      RR     II      VV      VV  EE          RR      RR
  YY  YY    CC          DD      DD  RR      RR     II      VV      VV  EE          RR      RR
  YY  YY    CC          DD      DD  RR      RR     II      VV      VV  EE          RR      RR
    YY      CC          DD      DD  RRRRRRRR       II      VV      VV  EEEEEEEE    RRRRRRRR
    YY      CC          DD      DD  RRRRRRRR       II      VV      VV  EEEEEEEE    RRRRRRRR
    YY      CC          DD      DD  RR  RR         II      VV      VV  EE          RR  RR
    YY      CC          DD      DD  RR  RR         II      VV      VV  EE          RR  RR
    YY      CC          DD      DD  RR    RR       II        VV  VV    EE          RR    RR      ....
    YY      CC          DD      DD  RR    RR       II        VV  VV    EE          RR    RR      ....
    YY        CCCCCCCC  DDDDDDDD    RR      RR   IIIIII        VV      EEEEEEEEEE  RR      RR    ....
    YY        CCCCCCCC  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

```
                        .TITLE  YCDRIVER - Port Driver for DMF  Async
;MIR0001                .IDENT  'V04-002'
-1
                        ;
                        ;*****************************************************************************
                        ;*                                                                           *
                        ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                        ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                        ;*  ALL RIGHTS RESERVED.                                                     *
                        ;*                                                                           *
                        ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
                        ;*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE           *
                        ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER      *
                        ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
                        ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY       *
                        ;*  TRANSFERRED.                                                             *
                        ;*                                                                           *
                        ;*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE     *
                        ;*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT         *
                        ;*  CORPORATION.                                                             *
                        ;*                                                                           *
                        ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS      *
                        ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
                        ;*                                                                           *
                        ;*****************************************************************************
                        ;
                        ;++
                        ; FACILITY:
                        ;
                        ;       VAX/VMS TERMINAL DRIVER
                        ;
                        ; ABSTRACT:
                        ;
                        ;       DMF  ASYNC PORT DRIVER
                        ;
                        ; AUTHOR:
                        ;
                        ;       RICK SPITZ
                        ;
                        ; Revision history:
                        ;
;MIR0001                 ;       V04-002 MIR0001         Michael I. Rosenblum    09-Nov-1984
;MIR0001                 ;               Some DMF lookalike board don't
;MIR00C1                 ;               Do Autoxon-xoff correctly.  Add code to disable the
;MIR0001                 ;               feature based on bit 0 of the sysgen parameter
;MIR0001                 ;               TTY_DEFPORT.
;MIR0001                 ;
;WHM0004                 ;       V04-001 WHM0004         Bill Matthews           30-Oct-1984
;WHM0004                 ;               Correct test for DMF32 vs DMZ32.
;WHM0004                 ;
                        ;       V03-032 WHM0003         Bill Matthews           24-Jul-1984
                        ;               Set the device type field in the CRB.
                        ;
                        ;       V03-231 LMP0275         L. Mark Pilant,         12-Jul-1984  21:04
                        ;               Initialize the ACL info in the ORB to be a null descriptor
                        ;               list rather than an empty queue.  This avoids the overhead
```

```
0000     49 ;                  of locking and unlocking the ACL mutex, only to find out
0000     50 ;                  that the ACL was empty.
0000     51 ;
0000     52 ;          V03-030 EMD0089         Ellen M. Dusseault      30-Apr-1984
0000     53 ;                  Add DEV$M_NNM characteristic to DEVCHAR2 so that these
0000     54 ;                  devices will have the prefix "node$".
0000     55 ;
0000     56 ;          V03-029 LMP0221         L. Mark Pilant,         31-Mar-1984  10:23
0000     57 ;                  Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
0000     58 ;                  ORB$W_PROT.
0000     59 ;
0000     60 ;          V03-028 MIR0360         Michael I. Rosenblum    20-Mar-1984
0000     61 ;                  Add code to check for DMF honoring XON and XOFF.
0000     62 ;                  Remove Callback to BURST_OUTPUT when resume is called
0000     63 ;                  This illiminates unnecessry interrupts when the silo
0000     64 ;                  Goes empty and we get a control-Q.
0000     65 ;
0000     66 ;          V03-027 WHM0002         Bill Matthews   29-Feb-1984
0000     67 ;                  Second part of change to control_init to support
0000     68 ;                  multiple comm devices on one board.
0000     69 ;
0000     70 ;          V03-026 WHM0001         Bill Matthews           28-Dec-1983
0000     71 ;                  Use IDB$B_COMBO_VECTOR and IDB$B_COMBO_CSR_OFFSET to find
0000     72 ;                  the start of the combo device's CSR and load the soft
0000     73 ;                  interrupt vector.
0000     74 ;
0000     75 ;          V03-025 JLV0318         Jake VanNoy             9-DEC-1983
0000     76 ;                  comment out DPT_STORE for parity to fix build problem.
0000     77 ;
0000     78 ;          V03-024 MIR0083         Michael I. Rosenblum            02-Aug-1983
0000     79 ;                  Add Field test DMZ-32 support.  Remove local def of
0000     80 ;                  IO$M_AUTXOF_ENA and IO$M_AUTXOF_DIS and move them into STARDEF.
0000     81 ;
0000     82 ;          V03-023 MIR0062         Michael I. Rosenblum            30-Jun-1983
0000     83 ;                  Change controler and unit init to use newly defined
0000     84 ;                  macros.  Remove code that looked at the logical UCB
0000     85 ;                  (code moved to the class drivers setup ucb and transition
0000     86 ;                  routine) Remove into the class driver on resume with nothing
0000     87 ;                  to resume.
0000     88 ;
0000     89 ;          V03-022 MIR0060         Michael I. Rosenblum            1-Jun-1983
0000     90 ;                  Add code to combat against silo full conditions and spurious
0000     91 ;                  DMA completeion interupts.
0000     92 ;
0000     93 ;          V03-021 MIR0050         Michael I. Rosenblum            11-May-1983
0000     94 ;                  Remove STOP2 entery point (no longer necessary)
0000     95 ;                  add code in set_line to allow the use of the auto xon xoff
0000     96 ;                  feature of the dmf.  Fix bug that would cause setting of the
0000     97 ;                  XON bit not to restart a write.
0000     98 ;
0000     99 ;          V03-020 RKS0020         RICK SPITZ                      14-MAR-1983
0000    100 ;                  ADD ENHANCEMENTS TO SUPPORT LOGICAL UCB
0000    101 ;
0000    102 ;          V03-019 RKS0019         RICK SPITZ                      1-FEB-1983
0000    103 ;                  DMA CODE HAS BEEN MOVED INTO THE DMF PORT DRIVER. THIS
0000    104 ;                  CODE HAS BEEN REMOVED FROM THE CLASS LEVEL. ALL OUTPUT
0000    105 ;                  FROM CLASS TO PORT IS NOW HANDLED USING THE SAME INTERFACE.
```

```
0000   106 ;
0000   107 ;               RESUME LOGIC HAS BEEN CHANGED TO INSURE INT IS ON DURING
0000   108 ;               DMA BURST AND BYTE COUNT = 0 TO INSURE NO NEW OUTPUT IS
0000   109 ;               STARTED DURING THAT WINDOW.
0000   110 ;
0000   111 ;       V03-018 MIR0022         Michael I. Rosenblum            18-Jan-1983
0000   112 ;               Replace old vector table with new vector table macro.
0000   113 ;               Remove references to UCB$L_DEVDEPEND and UCB$Q_TT_STATE
0000   114 ;               move these references into the class driver jacket routines
0000   115 ;
0000   116 ;       V03-017 MIR0017         Michael I. Rosenblum            04-Jan-1983
0000   117 ;               Add powerfail check in the Unit init routine to allow the
0000   118 ;               terminal class drier to take positive action on powerfail.
0000   119 ;
0000   120 ;       V03-016 MIR0016         Michael I. Rosenblum            29-Dec-1982
0000   121 ;               Replace time calculation code with TIMSET macro call.
0000   122 ;
0000   123 ;       V03-015 MIR0015         Michael I. Rosenblum    22-Dec-1982
0000   124 ;               Change code in YC$DISCONNECT to reflect the change
0000   125 ;               in functionality.  Remove code that referenced
0000   126 ;               DMAABO and DMA  state bits.
0000   127 ;
0000   128 ;       V03-014 MIR0014         Michael I. Rosenblum    17-Dec-1982
0000   129 ;               move code to figure out what flow control character
0000   130 ;               to send to TTYSUB.
0000   131 ;
0000   132 ;       V03-013 RKS0013         RICK SPITZ                      18-OCT-1982
0000   133 ;               CORRECT SENSE OF PARITY BIT
0000   134 ;
0000   135 ;       V03-010 RKS0012         RICK SPITZ                      16-SEP-1982
0000   136 ;               Remove check for powerfail in unit init which prevents
0000   137 ;               SETUP_UCB from being called. This is needed to allow
0000   138 ;               Unibus switches to connect a line and then switch the
0000   139 ;               device on the bus, followed by a psuedo powerfail recovery.
0000   140 ;
0000   141 ;               Check reference count in unit init to determine if
0000   142 ;               modem control should be initialized or hungup. This
0000   143 ;               is needed to insure that a hangup ^Y is posted on
0000   144 ;               powerfail on modem lines.
0000   145 ;
0000   146 ;               Refrain from clearing INT in the port abort logic.
0000   147 ;               This prevents output data following a flush from being
0000   148 ;               lost during a small time window.
0000   149 ;               The DMF will interrupt on silo flush complete, at
0000   150 ;               which point INT will be handled.
0000   151 ;
0000   152 ;       V03-009 KDM0002         Kathleen D. Morse       28-Jun-1982
0000   153 ;               Added $DEVDEF and $SSDEF.
0000   154 ;
0000   155 ;       V03-008 RKS008          RICK SPITZ                      23-MAR-1982
0000   156 ;               USE SYSGEN VALUE FOR COMBO INPUT SILO TIMER.
0000   157 ;               CHANGE OUTPUT DELAY.
0000   158 ;
0000   159 ;       V03-007 RKS007          RICK SPITZ                      13-FEB-1982
0000   160 ;               ADD DELAY LOGIC WHEN OUTPUTING TO TRANSMIT SILO
0000   161 ;
0000   162 ;       V03-006 RKS0006         RICK SPITZ                      8-FEB-1982
0000   163 ;               DO NOT ALLOW UNLOAD
```

```
0000   163 ;               FIX CHARACTER TIMER
0000   164 ;               DO UNSIGNED COMPARISON IN SILO OUTPUT.
0000   165 ;
0000   166 ;       V03-005 RKS0005         RICK SPITZ                  11-JAN-1982
0000   167 ;               REPAIR MODEM INTERRUPT LOGIC
0000   168 ;               ADD INPUT SILO TIMER.
0000   169 ;
0000   170 ;       V03-004 RKS0004         RICK SPITZ                  15-DEC-1981
0000   171 ;               FIX MAINTENANCE BIT OFFSETS, ADD EXTERNAL LOOP SUPPORT.
0000   172 ;               ENHANCE RESUME LOGIC IN DMA MODE TO AVOID WRITING 0 BYTE
0000   173 ;               COUNT.
0000   174 ;               USE SYSPARAM VALUES FOR CLASS DRIVER LOCATION,PARITY,RSPEED
0000   175 ;               CHANGE TYPE CODE TO DMF-32.
0000   176 ;
0000   177 ;       V03-003 JLV0104         Jake VanNoy                 27-Oct-1981
0000   178 ;               Changed TTYDEFS to $TTYDEFS.
0000   179 ;
0000   180 ;       V03-002 RKS002          RICK SPITZ                  8-OCT-1981
0000   181 ;               USE SUBTRACT TO COMPUTE VECTOR ADDRESS TO BE LOADED INTO
0000   182 ;               DMF-32 VECTOR REGISTER.
0000   183 ;
0000   184 ;
0000   185 ;--
```

```
0000    187          .SBTTL  DECLARATIONS
0000    188
0000    189 ;
0000    190 ; EXTERNAL DEFINITIONS:
0000    191 ;
0000    192          $ACBDEF                         ; DEFINE ACB
0000    193          $CRBDEF                         ; DEFINE CRB
0000    194          $DCDEF                          ; DEFINE ADAPTER TYPES
0000    195          $DDBDEF                         ; DEFINE DDB
0000    196          $DEVDEF                         ; DEFINE DEVICE TYPES
0000    197          $DYNDEF                         ; DEFINE DYNAMIC STRUCTURE TYPES
0000    198          $IDBDEF                         ; DEFINE IDB OFFSETS
0000    199          $IODEF                          ; DEFINE I/O FUNCTION CODES
0000    200          $IRPDEF                         ; DEFINE IRP
0000    201          $ORBDEF                         ; DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
0000    202          $SSDEF                          ; DEFINE SYSTEM STATUS CODES
0000    203          $TTYDEF                         ; DEFINE TERMINAL DRIVER SYMBOLS
0000    204          $TTDEF                          ; DEFINE TERMINAL TYPES
0000    205          $TT2DEF                         ; DEFINE EXTENDED CHARACTERISTICS
0000    206          $TQEDEF                         ; DEFINE TIMER QUEUE OFFSETS
0000    207          $UCBDEF                         ; DEFINE UCB
0000    208          $UBADEF                         ; DEFINE UBA
0000    209          $VECDEF                         ; DEFINE VECTOR FOR CRB
0000    210          $TTYMACS                        ; DEFINE TERMINAL DRIVER MACROS
0000    211 .show me
0000    212          $TTYDEFS                        ; DEFINE TERMINAL DRIVER SYMBOLS
0000                 $DEFINI TTYDEFS,
0000                 .SAVE   LOCAL_BLOCK
0000                 .NOCROSS
0000                 .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
0000                 .PSECT  $ABS$,ABS
0000                 $GBLINI
0000                 .IF     IDN <LOCAL> <GLOBAL>
0000                 .MACRO  $DEF    SYM,ALLOC,SIZ
0000                 .IIF    NB,SYM, SYM::
0000                 .IIF    NB,ALLOC,       ALLOC   SIZ
0000                 .ENDM   $DEF
0000                 .MACRO  $EQU    SYM,VAL
0000                 SYM==VAL
0000                 .ENDM   $EQU
0000                 .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000                 SIZ...=1
0000                 .IIF    NB,SIZ, SIZ...=SIZ
0000                 .IF     NB,SYM
0000                 MOD'SEP'V_'SYM==BIT...
0000                 .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
0000                 .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
0000                 .ENDC
0000                 BIT...=BIT...+SIZ...
0000                 .ENDM   $VIELD1
0000                 .IFF
0000                 .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
0000                 .MACRO  $DEF    SYM,ALLOC,SIZ
0000                 .IIF    NB,SYM, SYM:
0000                 .IIF    NB,ALLOC,       ALLOC   SIZ
0000                 .ENDM   $DEF
0000                 .MACRO  $EQU    SYM,VAL
```

```
                     0000               SYM=VAL
                     0000               .ENDM   $EQU
                     0000               .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000               SIZ...=1
                     0000               .IIF    NB,SIZ, SIZ...=SIZ
                     0000               .IF     NB,SYM
                     0000               MOD'SEP'V_'SYM=BIT...
                     0000               .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000               .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000               .ENDC
                     0000               BIT...=BIT...+SIZ...
                     0000               .ENDM   $VIELD1
                     0000               .ENDC
                     0000
00000000             0000               .=0
                     0000
                     0000               $UCBDEF
                     0000
                     0000               $CRBDEF
                     0000
                     0000               $IDBDEF
                     0000
                     0000               $TTYUCBDEF
                     0000               $DEFINI TTYUCB,
                     0000               .SAVE   LOCAL_BLOCK
                     0000               .NOCROSS
                     0000               .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
                     0000               .PSECT  $ABS$,ABS
                     0000               $GBLINI
                     0000               .IF     IDN <LOCAL> <GLOBAL>
                     0000               .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000               .IIF    NB,SYM, SYM::
                     0000               .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000               .ENDM   $DEF
                     0000               .MACRO  $EQU    SYM,VAL
                     0000               SYM==VAL
                     0000               .ENDM   $EQU
                     0000               .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000               SIZ...=1
                     0000               .IIF    NB,SIZ, SIZ...=SIZ
                     0000               .IF     NB,SYM
                     0000               MOD'SEP'V_'SYM==BIT...
                     0000               .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
                     0000               .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
                     0000               .ENDC
                     0000               BIT...=BIT...+SIZ...
                     0000               .ENDM   $VIELD1
                     0000               .IFF
                     0000               .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE ''GLOBAL'',''LOCAL'',OR NULL
                     0000               .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000               .IIF    NB,SYM, SYM:
                     0000               .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000               .ENDM   $DEF
                     0000               .MACRO  $EQU    SYM,VAL
                     0000               SYM=VAL
                     0000               .ENDM   $EQU
                     0000               .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
```

```
                     0000               SIZ...=1
                     0000               .IIF     NB,SIZ, SIZ...=SIZ
                     0000               .IF      NB,SYM
                     0000               MOD'SEP'V_'SYM=BIT...
                     0000               .IIF     NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000               .IIF     NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000               .ENDC
                     0000               BIT...=BIT...+SIZ...
                     0000               .ENDM   $VIELD1
                     0000               .ENDC
                     0000
00000000             0000               .=0
                     0000
                     0000     $EQU      UCB$C_TL_LENGTH 176
000000B0             0000               UCB$C_TL_LENGTH=176
                     0000
                     0000     $EQU      UCB$K_TL_LENGTH 176
000000B0             0000               UCB$K_TL_LENGTH=176
                     0000
                     0000     $EQU      TTY$M_ST_POWER  1
00000001             0000               TTY$M_ST_POWER=1
                     0000
                     0000     $EQU      TTY$M_ST_CTRLS  2
00000002             0000               TTY$M_ST_CTRLS=2
                     0000
                     0000     $EQU      TTY$M_ST_FILL   4
00000004             0000               TTY$M_ST_FILL=4
                     0000
                     0000     $EQU      TTY$M_ST_CURSOR 8
00000008             0000               TTY$M_ST_CURSOR=8
                     0000
                     0000     $EQU      TTY$M_ST_SENDLF 16
00000010             0000               TTY$M_ST_SENDLF=16
                     0000
                     0000     $EQU      TTY$M_ST_BACKSPACE      32
00000020             0000               TTY$M_ST_BACKSPACE=32
                     0000
                     0000     $EQU      TTY$M_ST_MULTI  64
00000040             0000               TTY$M_ST_MULTI=64
                     0000
                     0000     $EQU      TTY$M_ST_WRITE  128
00000080             0000               TTY$M_ST_WRITE=128
                     0000
                     0000     $EQU      TTY$M_ST_EOL    256
00000100             0000               TTY$M_ST_EOL=256
                     0000
                     0000     $EQU      TTY$M_ST_EDITREAD       512
00000200             0000               TTY$M_ST_EDITREAD=512
                     0000
                     0000     $EQU      TTY$M_ST_RDVERIFY       1024
00000400             0000               TTY$M_ST_RDVERIFY=1024
                     0000
                     0000     $EQU      TTY$M_ST_RECALL 2048
00000800             0000               TTY$M_ST_RECALL=2048
                     0000
                     0000     $EQU      TTY$M_ST_READ   4096
00001000             0000               TTY$M_ST_READ=4096
```

```
                     0000
                     0000     $EQU    TTY$M_ST_CTRLO  1
00000001             0000             TTY$M_ST_CTRLO=1
                     0000
                     0000     $EQU    TTY$M_ST_DEL    2
00000002             0000             TTY$M_ST_DEL=2
                     0000
                     0000     $EQU    TTY$M_ST_PASALL 4
00000004             0000             TTY$M_ST_PASALL=4
                     0000
                     0000     $EQU    TTY$M_ST_NOECHO 8
00000008             0000             TTY$M_ST_NOECHO=8
                     0000
                     0000     $EQU    TTY$M_ST_WRTALL 16
00000010             0000             TTY$M_ST_WRTALL=16
                     0000
                     0000     $EQU    TTY$M_ST_PROMPT 32
00000020             0000             TTY$M_ST_PROMPT=32
                     0000
                     0000     $EQU    TTY$M_ST_NOFLTR 64
00000040             0000             TTY$M_ST_NOFLTR=64
                     0000
                     0000     $EQU    TTY$M_ST_ESC    128
00000080             0000             TTY$M_ST_ESC=128
                     0000
                     0000     $EQU    TTY$M_ST_BADESC 256
00000100             0000             TTY$M_ST_BADESC=256
                     0000
                     0000     $EQU    TTY$M_ST_NL     512
00000200             0000             TTY$M_ST_NL=512
                     0000
                     0000     $EQU    TTY$M_ST_REFRSH 1024
00000400             0000             TTY$M_ST_REFRSH=1024
                     0000
                     0000     $EQU    TTY$M_ST_ESCAPE 2048
00000800             0000             TTY$M_ST_ESCAPE=2048
                     0000
                     0000     $EQU    TTY$M_ST_TYPFUL 4096
00001000             0000             TTY$M_ST_TYPFUL=4096
                     0000
                     0000     $EQU    TTY$M_ST_SKIPLF 8192
00002000             0000             TTY$M_ST_SKIPLF=8192
                     0000
                     0000     $EQU    TTY$M_ST_ESC_O  16384
00004000             0000             TTY$M_ST_ESC_O=16384
                     0000
                     0000     $EQU    TTY$M_ST_WRAP   32768
00008000             0000             TTY$M_ST_WRAP=32768
                     0000
                     0000     $EQU    TTY$M_ST_OVRFLO 65536
00010000             0000             TTY$M_ST_OVRFLO=65536
                     0000
                     0000     $EQU    TTY$M_ST_AUTOP  131072
00020000             0000             TTY$M_ST_AUTOP=131072
                     0000
                     0000     $EQU    TTY$M_ST_CTRLR  262144
00040000             0000             TTY$M_ST_CTRLR=262144
```

```
                     0000
                     0000     $EQU    TTY$M_ST_SKIPCRLF       524288
00080000             0000             TTY$M_ST_SKIPCRLF=524288
                     0000
                     0000     $EQU    TTY$M_ST_EDITING        1048576
00100000             0000             TTY$M_ST_EDITING=1048576
                     0000
                     0000     $EQU    TTY$M_ST_TABEXPAND      2097152
00200000             0000             TTY$M_ST_TABEXPAND=2097152
                     0000
                     0000     $EQU    TTY$M_ST_QUOTING        4194304
00400000             0000             TTY$M_ST_QUOTING=4194304
                     0000
                     0000     $EQU    TTY$M_ST_OVERSTRIKE     8388608
00800000             0000             TTY$M_ST_OVERSTRIKE=8388608
                     0000
                     0000     $EQU    TTY$M_ST_TERMNORM       16777216
01000000             0000             TTY$M_ST_TERMNORM=16777216
                     0000
                     0000     $EQU    TTY$M_ST_ECHAES 33554432
02000000             0000             TTY$M_ST_ECHAES=33554432
                     0000
                     0000     $EQU    TTY$M_ST_PRE    67108864
04000000             0000             TTY$M_ST_PRE=67108864
                     0000
                     0000     $EQU    TTY$M_ST_NINTMULTI      134217728
08000000             0000             TTY$M_ST_NINTMULTI=134217728
                     0000
                     0000     $EQU    TTY$M_ST_RECONNECT      268435456
10000000             0000             TTY$M_ST_RECONNECT=268435456
                     0000
                     0000     $EQU    TTY$M_ST_CTSLOW 536870912
20000000             0000             TTY$M_ST_CTSLOW=536870912
                     0000
                     0000     $EQU    TTY$M_ST_TABRIGHT       1073741824
40000000             0000             TTY$M_ST_TABRIGHT=1073741824
                     0000
                     0000     $EQU    UCB$M_TT_XXPARITY       1
00000001             0000             UCB$M_TT_XXPARITY=1
                     0000
                     0000     $EQU    UCB$M_TT_DISPARERR      2
00000002             0000             UCB$M_TT_DISPARERR=2
                     0000
                     0000     $EQU    UCB$M_TT_USERFRAME      4
00000004             0000             UCB$M_TT_USERFRAME=4
                     0000
                     0000     $EQU    UCB$M_TT_LEN    24
00000018             0000             UCB$M_TT_LEN=24
                     0000
                     0000     $EQU    UCB$M_TT_STOP   32
00000020             0000             UCB$M_TT_STOP=32
                     0000
                     0000     $EQU    UCB$M_TT_PARTY  64
00000040             0000             UCB$M_TT_PARTY=64
                     0000
                     0000     $EQU    UCB$M_TT_ODD    128
00000080             0000             UCB$M_TT_ODD=128
```

```
                        0000
                        0000    $EQU    TTY$M_TANK_PREMPT       256
00000100                0000            TTY$M_TANK_PREMPT=256
                        0000
                        0000    $EQU    TTY$M_TANK_STOP 512
00000200                0000            TTY$M_TANK_STOP=512
                        0000
                        0000    $EQU    TTY$M_TANK_HOLD 1024
00000400                0000            TTY$M_TANK_HOLD=1024
                        0000
                        0000    $EQU    TTY$M_TANK_BURST        2048
00000800                0000            TTY$M_TANK_BURST=2048
                        0000
                        0000    $EQU    TTY$M_TANK_DMA  4096
00001000                0000            TTY$M_TANK_DMA=4096
                        0000
                        0000    $EQU    TTY$M_PC_NOTIME 1
00000001                0000            TTY$M_PC_NOTIME=1
                        0000
                        0000    $EQU    TTY$M_PC_DMAENA 2
00000002                0000            TTY$M_PC_DMAENA=2
                        0000
                        0000    $EQU    TTY$M_PC_DMAAVL 4
00000004                0000            TTY$M_PC_DMAAVL=4
                        0000
                        0000    $EQU    TTY$M_PC_PRMMAP 8
00000008                0000            TTY$M_PC_PRMMAP=8
                        0000
                        0000    $EQU    TTY$M_PC_MAPAVL 16
00000010                0000            TTY$M_PC_MAPAVL=16
                        0000
                        0000    $EQU    TTY$M_PC_XOFAVL 32
00000020                0000            TTY$M_PC_XOFAVL=32
                        0000
                        0000    $EQU    TTY$M_PC_XOFENA 64
00000040                0000            TTY$M_PC_XOFENA=64
                        0000
                        0000    $EQU    TTY$M_PC_NOCRLF 128
00000080                0000            TTY$M_PC_NOCRLF=128
                        0000
                        0000    $EQU    UCB$M_TT_DSBL   128
00000080                0000            UCB$M_TT_DSBL=128
                        0000
                        0000    $EQU    UCB$C_TT_CLSLEN 300
0000012C                0000            UCB$C_TT_CLSLEN=300
                        0000
                        0000    $EQU    UCB$K_TT_CLSLEN 300
0000012C                0000            UCB$K_TT_CLSLEN=300
                        0000
                        0000    $EQU    TTY$M_TP_ABORT  1
00000001                0000            TTY$M_TP_ABORT=1
                        0000
                        0000    $EQU    TTY$M_TP_ALLOC  2
00000002                0000            TTY$M_TP_ALLOC=2
                        0000
                        0000    $EQU    TTY$M_TP_DLLOC  4
00000004                0000            TTY$M_TP_DLLOC=4
```

```
                        0000
                        0000    $EQU    UCB$C_TP_LENGTH 308
00000134                0000            UCB$C_TP_LENGTH=308
                        0000
                        0000    $EQU    UCB$K_TP_LENGTH 308
00000134                0000            UCB$K_TP_LENGTH=308
                        0000
                        0000    $EQU    UCB$C_TT_LENGTH 308
00000134                0000            UCB$C_TT_LENGTH=308
                        0000
                        0000    $EQU    UCB$K_TT_LENGTH 308
00000134                0000            UCB$K_TT_LENGTH=308
                        0000
                        0000    $EQU    FLG$M_WIIRP_BSY 1
00000001                0000            FLG$M_WIIRP_BSY=1
                        0000
                        0000    $EQU    FLG$M_CTRLO     2
00000002                0000            FLG$M_CTRLO=2
                        0000
                        0000    $EQU    FLG$M_CANCTRLO  4
00000004                0000            FLG$M_CANCTRLO=4
                        0000
                        0000    $EQU    FLG$M_INWRTFDT  8
00000008                0000            FLG$M_INWRTFDT=8
                        0000
                        0000    $EQU    FLG$M_QUOTA     16
00000010                0000            FLG$M_QUOTA=16
                        0000
                        0000    $EQU    FLG$M_VAXTOVAX  32
00000020                0000            FLG$M_VAXTOVAX=32
                        0000
                        0000    $EQU    FLG$M_BUFFER    64
00000040                0000            FLG$M_BUFFER=64
                        0000
                        0000    $EQU    UCB$C_RTT_LENGTH        312
00000138                0000            UCB$C_RTT_LENGTH=312
                        0000
                        0000    $EQU    UCB$K_RTT_LENGTH        312
00000138                0000            UCB$K_RTT_LENGTH=312
                        0000
                        0000    $EQU    UCB$S_TTYUCBDEF 316
0000013C                0000            UCB$S_TTYUCBDEF=316
                        0000
                        0000    $EQU    UCB$L_TL_CTRLY  144
00000090                0000            UCB$L_TL_CTRLY=144
                        0000
                        0000    $EQU    UCB$L_TL_CTRLC  148
00000094                0000            UCB$L_TL_CTRLC=148
                        0000
                        0000    $EQU    UCB$L_TL_OUTBAND        152
00000098                0000            UCB$L_TL_OUTBAND=152
                        0000
                        0000    $EQU    UCB$L_TL_BANDQUE        156
0000009C                0000            UCB$L_TL_BANDQUE=156
                        0000
                        0000    $EQU    UCB$L_TL_PHYUCB 160
000000A0                0000            UCB$L_TL_PHYUCB=160
```

```
                     0000
                     0000     $EQU    UCB$L_TL_CTLPID 164
000000A4             0000             UCB$L_TL_CTLPID=164
                     0000
                     0000     $EQU    UCB$S_TL_BRKTHRU        8
00000008             0000             UCB$S_TL_BRKTHRU=8
                     0000
                     0000     $EQU    UCB$Q_TL_BRKTHRU        168
000000A8             0000             UCB$Q_TL_BRKTHRU=168
                     0000
                     0000     $EQU    UCB$S_TTYRTTUCB 140
0000008C             0000             UCB$S_TTYRTTUCB=140
                     0000
                     0000     $EQU    UCB$R_TTYRTTUCB 176
000000B0             0000             UCB$R_TTYRTTUCB=176
                     0000
                     0000     $EQU    UCB$S_TTYUCB    140
0000008C             0000             UCB$S_TTYUCB=140
                     0000
                     0000     $EQU    UCB$R_TTYUCB    176
000000B0             0000             UCB$R_TTYUCB=176
                     0000
                     0000     $EQU    UCB$L_TT_RDUE   176
000000B0             0000             UCB$L_TT_RDUE=176
                     0000
                     0000     $EQU    UCB$L_TT_RTIMOU 180
000000B4             0000             UCB$L_TT_RTIMOU=180
                     0000
                     0000     $EQU    UCB$S_TT_STATE  8
00000008             0000             UCB$S_TT_STATE=8
                     0000
                     0000     $EQU    UCB$Q_TT_STATE  184
000000B8             0000             UCB$Q_TT_STATE=184
                     0000
                     0000     $EQU    UCB$L_TT_STATE1 184
000000B8             0000             UCB$L_TT_STATE1=184
                     0000
                     0000     $EQU    TTY$V_ST_POWER  0
00000000             0000             TTY$V_ST_POWER=0
                     0000
                     0000     $EQU    TTY$V_ST_CTRLS  1
00000001             0000             TTY$V_ST_CTRLS=1
                     0000
                     0000     $EQU    TTY$V_ST_FILL   2
00000002             0000             TTY$V_ST_FILL=2
                     0000
                     0000     $EQU    TTY$V_ST_CURSOR 3
00000003             0000             TTY$V_ST_CURSOR=3
                     0000
                     0000     $EQU    TTY$V_ST_SENDLF 4
00000004             0000             TTY$V_ST_SENDLF=4
                     0000
                     0000     $EQU    TTY$V_ST_BACKSPACE      5
00000005             0000             TTY$V_ST_BACKSPACE=5
                     0000
                     0000     $EQU    TTY$V_ST_MULTI  6
00000006             0000             TTY$V_ST_MULTI=6
```

```
                        0000
                        0000    $EQU    TTY$V_ST_WRITE  7
00000007                0000            TTY$V_ST_WRITE=7
                        0000
                        0000    $EQU    TTY$V_ST_EOL    8
00000008                0000            TTY$V_ST_EOL=8
                        0000
                        0000    $EQU    TTY$V_ST_EDITREAD       9
00000009                0000            TTY$V_ST_EDITREAD=9
                        0000
                        0000    $EQU    TTY$V_ST_RDVERIFY       10
0000000A                0000            TTY$V_ST_RDVERIFY=10
                        0000
                        0000    $EQU    TTY$V_ST_RECALL 11
0000000B                0000            TTY$V_ST_RECALL=11
                        0000
                        0000    $EQU    TTY$V_ST_READ   12
0000000C                0000            TTY$V_ST_READ=12
                        0000
                        0000    $EQU    UCB$L_TT_STATE2 188
000000BC                0000            UCB$L_TT_STATE2=188
                        0000
                        0000    $EQU    TTY$V_ST_CTRLO  0
00000000                0000            TTY$V_ST_CTRLO=0
                        0000
                        0000    $EQU    TTY$V_ST_DEL    1
00000001                0000            TTY$V_ST_DEL=1
                        0000
                        0000    $EQU    TTY$V_ST_PASALL 2
00000002                0000            TTY$V_ST_PASALL=2
                        0000
                        0000    $EQU    TTY$V_ST_NOECHO 3
00000003                0000            TTY$V_ST_NOECHO=3
                        0000
                        0000    $EQU    TTY$V_ST_WRTALL 4
00000004                0000            TTY$V_ST_WRTALL=4
                        0000
                        0000    $EQU    TTY$V_ST_PROMPT 5
00000005                0000            TTY$V_ST_PROMPT=5
                        0000
                        0000    $EQU    TTY$V_ST_NOFLTR 6
00000006                0000            TTY$V_ST_NOFLTR=6
                        0000
                        0000    $EQU    TTY$V_ST_ESC    7
00000007                0000            TTY$V_ST_ESC=7
                        0000
                        0000    $EQU    TTY$V_ST_BADESC 8
00000008                0000            TTY$V_ST_BADESC=8
                        0000
                        0000    $EQU    TTY$V_ST_NL     9
00000009                0000            TTY$V_ST_NL=9
                        0000
                        0000    $EQU    TTY$V_ST_REFRSH 10
0000000A                0000            TTY$V_ST_REFRSH=10
                        0000
                        0000    $EQU    TTY$V_ST_ESCAPE 11
0000000B                0000            TTY$V_ST_ESCAPE=11
```

```
                     0000
                     0000      $EQU    TTY$V_ST_TYPFUL 12
0000000C             0000              TTY$V_ST_TYPFUL=12
                     0000
                     0000      $EQU    TTY$V_ST_SKIPLF 13
0000000D             0000              TTY$V_ST_SKIPLF=13
                     0000
                     0000      $EQU    TTY$V_ST_ESC_O  14
0000000E             0000              TTY$V_ST_ESC_O=14
                     0000
                     0000      $EQU    TTY$V_ST_WRAP   15
0000000F             0000              TTY$V_ST_WRAP=15
                     0000
                     0000      $EQU    TTY$V_ST_OVRFLO 16
00000010             0000              TTY$V_ST_OVRFLO=16
                     0000
                     0000      $EQU    TTY$V_ST_AUTOP  17
00000011             0000              TTY$V_ST_AUTOP=17
                     0000
                     0000      $EQU    TTY$V_ST_CTRLR  18
00000012             0000              TTY$V_ST_CTRLR=18
                     0000
                     0000      $EQU    TTY$V_ST_SKIPCRLF       19
00000013             0000              TTY$V_ST_SKIPCRLF=19
                     0000
                     0000      $EQU    TTY$V_ST_EDITING        20
00000014             0000              TTY$V_ST_EDITING=20
                     0000
                     0000      $EQU    TTY$V_ST_TABEXPAND      21
00000015             0000              TTY$V_ST_TABEXPAND=21
                     0000
                     0000      $EQU    TTY$V_ST_QUOTING        22
00000016             0000              TTY$V_ST_QUOTING=22
                     0000
                     0000      $EQU    TTY$V_ST_OVERSTRIKE     23
00000017             0000              TTY$V_ST_OVERSTRIKE=23
                     0000
                     0000      $EQU    TTY$V_ST_TERMNORM       24
00000018             0000              TTY$V_ST_TERMNORM=24
                     0000
                     0000      $EQU    TTY$V_ST_ECHAES 25
00000019             0000              TTY$V_ST_ECHAES=25
                     0000
                     0000      $EQU    TTY$V_ST_PRE    26
0000001A             0000              TTY$V_ST_PRE=26
                     0000
                     0000      $EQU    TTY$V_ST_NINTMULTI      27
0000001B             0000              TTY$V_ST_NINTMULTI=27
                     0000
                     0000      $EQU    TTY$V_ST_RECONNECT      28
0000001C             0000              TTY$V_ST_RECONNECT=28
                     0000
                     0000      $EQU    TTY$V_ST_CTSLOW 29
0000001D             0000              TTY$V_ST_CTSLOW=29
                     0000
                     0000      $EQU    TTY$V_ST_TABRIGHT       30
0000001E             0000              TTY$V_ST_TABRIGHT=30
```

```
                     0000
                     0000     $EQU    UCB$L_TT_LOGUCB 192
000000C0             0000             UCB$L_TT_LOGUCB=192
                     0000
                     0000     $EQU    UCB$L_TT_DECHAR 196
000000C4             0000             UCB$L_TT_DECHAR=196
                     0000
                     0000     $EQU    UCB$L_TT_DECHA1 200
000000C8             0000             UCB$L_TT_DECHA1=200
                     0000
                     0000     $EQU    UCB$L_TT_WFLINK 204
000000CC             0000             UCB$L_TT_WFLINK=204
                     0000
                     0000     $EQU    UCB$L_TT_WBLINK 208
000000D0             0000             UCB$L_TT_WBLINK=208
                     0000
                     0000     $EQU    UCB$L_TT_WRTBUF 212
000000D4             0000             UCB$L_TT_WRTBUF=212
                     0000
                     0000     $EQU    UCB$L_TT_MULTI  216
000000D8             0000             UCB$L_TT_MULTI=216
                     0000
                     0000     $EQU    UCB$W_TT_MULTILEN       220
000000DC             0000             UCB$W_TT_MULTILEN=220
                     0000
                     0000     $EQU    UCB$W_TT_SMLTLEN        222
000000DE             0000             UCB$W_TT_SMLTLEN=222
                     0000
                     0000     $EQU    UCB$L_TT_SMLT   224
000000E0             0000             UCB$L_TT_SMLT=224
                     0000
                     0000     $EQU    UCB$L_TT_TYPAHD 228
000000E4             0000             UCB$L_TT_TYPAHD=228
                     0000
                     0000     $EQU    UCB$W_TT_DESPEE 232
000000E8             0000             UCB$W_TT_DESPEE=232
                     0000
                     0000     $EQU    UCB$B_TT_DECRF  234
000000EA             0000             UCB$B_TT_DECRF=234
                     0000
                     0000     $EQU    UCB$B_TT_DELFF  235
000000EB             0000             UCB$B_TT_DELFF=235
                     0000
                     0000     $EQU    UCB$B_TT_DEPARI 236
000000EC             0000             UCB$B_TT_DEPARI=236
                     0000
                     0000     $EQU    UCB$B_TT_DEFSPE_SPARE1  237
000000ED             0000             UCB$B_TT_DEFSPE_SPARE1=237
                     0000
                     0000     $EQU    UCB$W_TT_DEFSPE_SPARE2  238
000000EE             0000             UCB$W_TT_DEFSPE_SPARE2=238
                     0000
                     0000     $EQU    UCB$B_TT_DETYPE 240
000000F0             0000             UCB$B_TT_DETYPE=240
                     0000
                     0000     $EQU    UCB$W_TT_DESIZE 241
000000F1             0000             UCB$W_TT_DESIZE=241
```

```
                     0000
                     0000          $EQU    UCB$B_TT_SPARE1 243
000000F3             0000                  UCB$B_TT_SPARE1=243
                     0000
                     0000          $EQU    UCB$W_TT_SPEED  244
000000F4             0000                  UCB$W_TT_SPEED=244
                     0000
                     0000          $EQU    UCB$B_TT_TSPEED 244
000000F4             0000                  UCB$B_TT_TSPEED=244
                     0000
                     0000          $EQU    UCB$B_TT_RSPEED 245
000000F5             0000                  UCB$B_TT_RSPEED=245
                     0000
                     0000          $EQU    UCB$B_TT_CRFILL 246
000000F6             0000                  UCB$B_TT_CRFILL=246
                     0000
                     0000          $EQU    UCB$B_TT_LFFILL 247
000000F7             0000                  UCB$B_TT_LFFILL=247
                     0000
                     0000          $EQU    UCB$B_TT_PARITY 248
000000F8             0000                  UCB$B_TT_PARITY=248
                     0000
                     0000          $EQU    UCB$V_TT_XXPARITY       0
00000000             0000                  UCB$V_TT_XXPARITY=0
                     0000
                     0000          $EQU    UCB$V_TT_DISPARERR      1
00000001             0000                  UCB$V_TT_DISPARERR=1
                     0000
                     0000          $EQU    UCB$V_TT_USERFRAME      2
00000002             0000                  UCB$V_TT_USERFRAME=2
                     0000
                     0000          $EQU    UCB$S_TT_LEN    2
00000002             0000                  UCB$S_TT_LEN=2
                     0000
                     0000          $EQU    UCB$V_TT_LEN    3
00000003             0000                  UCB$V_TT_LEN=3
                     0000
                     0000          $EQU    UCB$V_TT_STOP   5
00000005             0000                  UCB$V_TT_STOP=5
                     0000
                     0000          $EQU    UCB$V_TT_PARTY  6
00000006             0000                  UCB$V_TT_PARTY=6
                     0000
                     0000          $EQU    UCB$V_TT_ODD    7
00000007             0000                  UCB$V_TT_ODD=7
                     0000
                     0000          $EQU    UCB$B_TT_PAR_SPARE1     249
000000F9             0000                  UCB$B_TT_PAR_SPARE1=249
                     0000
                     0000          $EQU    UCB$W_TT_PAR_SPARE2     250
000000FA             0000                  UCB$W_TT_PAR_SPARE2=250
                     0000
                     0000          $EQU    UCB$W_TT_CURSOR 252
000000FC             0000                  UCB$W_TT_CURSOR=252
                     0000
                     0000          $EQU    UCB$B_TT_LINE   254
000000FE             0000                  UCB$B_TT_LINE=254
```

```
                     0000
                     0000          $EQU    UCB$B_TT_LASTC  255
000000FF             0000                  UCB$B_TT_LASTC=255
                     0000
                     0000          $EQU    UCB$W_TT_BSPLEN 256
00000100             0000                  UCB$W_TT_BSPLEN=256
                     0000
                     0000          $EQU    UCB$B_TT_FILL   258
00000102             0000                  UCB$B_TT_FILL=258
                     0000
                     0000          $EQU    UCB$B_TT_ESC    259
00000103             0000                  UCB$B_TT_ESC=259
                     0000
                     0000          $EQU    UCB$B_TT_ESC_0  260
00000104             0000                  UCB$B_TT_ESC_0=260
                     0000
                     0000          $EQU    UCB$B_TT_INTCNT 261
00000105             0000                  UCB$B_TT_INTCNT=261
                     0000
                     0000          $EQU    UCB$W_TT_UNITBIT        262
00000106             0000                  UCB$W_TT_UNITBIT=262
                     0000
                     0000          $EQU    UCB$W_TT_HOLD   264
00000108             0000                  UCB$W_TT_HOLD=264
                     0000
                     0000          $EQU    TTY$B_TANK_CHAR 264
00000108             0000                  TTY$B_TANK_CHAR=264
                     0000
                     0000          $EQU    TTY$V_TANK_PREMPT       8
00000008             0000                  TTY$V_TANK_PREMPT=8
                     0000
                     0000          $EQU    TTY$V_TANK_STOP 9
00000009             0000                  TTY$V_TANK_STOP=9
                     0000
                     0000          $EQU    TTY$V_TANK_HOLD 10
0000000A             0000                  TTY$V_TANK_HOLD=10
                     0000
                     0000          $EQU    TTY$V_TANK_BURST        11
0000000B             0000                  TTY$V_TANK_BURST=11
                     0000
                     0000          $EQU    TTY$V_TANK_DMA  12
0000000C             0000                  TTY$V_TANK_DMA=12
                     0000
                     0000          $EQU    UCB$B_TT_PREMPT 266
0000010A             0000                  UCB$B_TT_PREMPT=266
                     0000
                     0000          $EQU    UCB$B_TT_OUTYPE 267
0000010B             0000                  UCB$B_TT_OUTYPE=267
                     0000
                     0000          $EQU    UCB$L_TT_GETNXT 268
0000010C             0000                  UCB$L_TT_GETNXT=268
                     0000
                     0000          $EQU    UCB$L_TT_PUTNXT 272
00000110             0000                  UCB$L_TT_PUTNXT=272
                     0000
                     0000          $EQU    UCB$L_TT_CLASS  276
00000114             0000                  UCB$L_TT_CLASS=276
```

```
                    0000
                    0000        $EQU    UCB$L_TT_PORT   280
00000118            0000                UCB$L_TT_PORT=280
                    0000
                    0000        $EQU    UCB$L_TT_OUTADR 284
0000011C            0000                UCB$L_TT_OUTADR=284
                    0000
                    0000        $EQU    UCB$W_TT_OUTLEN 288
00000120            0000                UCB$W_TT_OUTLEN=288
                    0000
                    0000        $EQU    UCB$W_TT_PRTCTL 290
00000122            0000                UCB$W_TT_PRTCTL=290
                    0000
                    0000        $EQU    TTY$V_PC_NOTIME 0
00000000            0000                TTY$V_PC_NOTIME=0
                    0000
                    0000        $EQU    TTY$V_PC_DMAENA 1
00000001            0000                TTY$V_PC_DMAENA=1
                    0000
                    0000        $EQU    TTY$V_PC_DMAAVL 2
00000002            0000                TTY$V_PC_DMAAVL=2
                    0000
                    0000        $EQU    TTY$V_PC_PRMMAP 3
00000003            0000                TTY$V_PC_PRMMAP=3
                    0000
                    0000        $EQU    TTY$V_PC_MAPAVL 4
00000004            0000                TTY$V_PC_MAPAVL=4
                    0000
                    0000        $EQU    TTY$V_PC_XOFAVL 5
00000005            0000                TTY$V_PC_XOFAVL=5
                    0000
                    0000        $EQU    TTY$V_PC_XOFENA 6
00000006            0000                TTY$V_PC_XOFENA=6
                    0000
                    0000        $EQU    TTY$V_PC_NOCRLF 7
00000007            0000                TTY$V_PC_NOCRLF=7
                    0000
                    0000        $EQU    UCB$B_TT_DS_RCV 292
00000124            0000                UCB$B_TT_DS_RCV=292
                    0000
                    0000        $EQU    UCB$B_TT_DS_TX  293
00000125            0000                UCB$B_TT_DS_TX=293
                    0000
                    0000        $EQU    UCB$W_TT_DS_ST  294
00000126            0000                UCB$W_TT_DS_ST=294
                    0000
                    0000        $EQU    UCB$W_TT_DS_TIM 296
00000128            0000                UCB$W_TT_DS_TIM=296
                    0000
                    0000        $EQU    UCB$B_TT_MAINT  298
0000012A            0000                UCB$B_TT_MAINT=298
                    0000
                    0000        $EQU    UCB$S_TT_MAINT_FILL     7
00000007            0000                UCB$S_TT_MAINT_FILL=7
                    0000
                    0000        $EQU    UCB$V_TT_MAINT_FILL     0
00000000            0000                UCB$V_TT_MAINT_FILL=0
```

```
                        0000
                        0000    $EQU    UCB$V_TT_DSBL   7
00000007                0000            UCB$V_TT_DSBL=7
                        0000
                        0000    $EQU    UCB$B_TT_OLDCPZORG      299
0000012B                0000            UCB$B_TT_OLDCPZORG=299
                        0000
                        0000    $EQU    UCB$L_TP_MAP    300
0000012C                0000            UCB$L_TP_MAP=300
                        0000
                        0000    $EQU    UCB$B_TP_STAT   304
00000130                0000            UCB$B_TP_STAT=304
                        0000
                        0000    $EQU    TTY$V_TP_ABORT  0
00000000                0000            TTY$V_TP_ABORT=0
                        0000
                        0000    $EQU    TTY$V_TP_ALLOC  1
00000001                0000            TTY$V_TP_ALLOC=1
                        0000
                        0000    $EQU    TTY$V_TP_DLLOC  2
00000002                0000            TTY$V_TP_DLLOC=2
                        0000
                        0000    $EQU    UCB$B_TP_SPARE1 305
00000131                0000            UCB$B_TP_SPARE1=305
                        0000
                        0000    $EQU    UCB$W_TP_SPARE2 306
00000132                0000            UCB$W_TP_SPARE2=306
                        0000
                        0000    $EQU    TTY$S_TT_STATE_SX       8
00000008                0000            TTY$S_TT_STATE_SX=8
                        0000
                        0000    $EQU    UCB$R_TT_STATE_SX       308
00000134                0000            UCB$R_TT_STATE_SX=308
                        0000
                        0000    $EQU    TTY$V_SX_POWER  0
00000000                0000            TTY$V_SX_POWER=0
                        0000
                        0000    $EQU    TTY$V_SX_CTRLS  1
00000001                0000            TTY$V_SX_CTRLS=1
                        0000
                        0000    $EQU    TTY$V_SX_FILL   2
00000002                0000            TTY$V_SX_FILL=2
                        0000
                        0000    $EQU    TTY$V_SX_CURSOR 3
00000003                0000            TTY$V_SX_CURSOR=3
                        0000
                        0000    $EQU    TTY$V_SX_SENDLF 4
00000004                0000            TTY$V_SX_SENDLF=4
                        0000
                        0000    $EQU    TTY$V_SX_BACKSPACE      5
00000005                0000            TTY$V_SX_BACKSPACE=5
                        0000
                        0000    $EQU    TTY$V_SX_MULTI  6
00000006                0000            TTY$V_SX_MULTI=6
                        0000
                        0000    $EQU    TTY$V_SX_WRITE  7
00000007                0000            TTY$V_SX_WRITE=7
```

```
                        0000
                        0000    $EQU    TTY$V_SX_EOL    8
00000008                0000            TTY$V_SX_EOL=8
                        0000
                        0000    $EQU    TTY$V_SX_EDITREAD       9
00000009                0000            TTY$V_SX_EDITREAD=9
                        0000
                        0000    $EQU    TTY$V_SX_RDVERIFY       10
0000000A                0000            TTY$V_SX_RDVERIFY=10
                        0000
                        0000    $EQU    TTY$V_SX_RECALL 11
0000000B                0000            TTY$V_SX_RECALL=11
                        0000
                        0000    $EQU    TTY$V_SX_READ   12
0000000C                0000            TTY$V_SX_READ=12
                        0000
                        0000    $EQU    TTY$S_SX_FILLBITS       19
00000013                0000            TTY$S_SX_FILLBITS=19
                        0000
                        0000    $EQU    TTY$V_SX_FILLBITS       13
0000000D                0000            TTY$V_SX_FILLBITS=13
                        0000
                        0000    $EQU    TTY$V_SX_CTRLO  32
00000020                0000            TTY$V_SX_CTRLO=32
                        0000
                        0000    $EQU    TTY$V_SX_DEL    33
00000021                0000            TTY$V_SX_DEL=33
                        0000
                        0000    $EQU    TTY$V_SX_PASALL 34
00000022                0000            TTY$V_SX_PASALL=34
                        0000
                        0000    $EQU    TTY$V_SX_NOECHO 35
00000023                0000            TTY$V_SX_NOECHO=35
                        0000
                        0000    $EQU    TTY$V_SX_WRTALL 36
00000024                0000            TTY$V_SX_WRTALL=36
                        0000
                        0000    $EQU    TTY$V_SX_PROMPT 37
00000025                0000            TTY$V_SX_PROMPT=37
                        0000
                        0000    $EQU    TTY$V_SX_NOFLTR 38
00000026                0000            TTY$V_SX_NOFLTR=38
                        0000
                        0000    $EQU    TTY$V_SX_ESC    39
00000027                0000            TTY$V_SX_ESC=39
                        0000
                        0000    $EQU    TTY$V_SX_BADESC 40
00000028                0000            TTY$V_SX_BADESC=40
                        0000
                        0000    $EQU    TTY$V_SX_NL     41
00000029                0000            TTY$V_SX_NL=41
                        0000
                        0000    $EQU    TTY$V_SX_REFRSH 42
0000002A                0000            TTY$V_SX_REFRSH=42
                        0000
                        0000    $EQU    TTY$V_SX_ESCAPE 43
0000002B                0000            TTY$V_SX_ESCAPE=43
```

```
                     0000
                     0000     $EQU    TTY$V_SX_TYPFUL 44
0000002C             0000             TTY$V_SX_TYPFUL=44
                     0000
                     0000     $EQU    TTY$V_SX_SKIPLF 45
0000002D             0000             TTY$V_SX_SKIPLF=45
                     0000
                     0000     $EQU    TTY$V_SX_ESC_O  46
0000002E             0000             TTY$V_SX_ESC_O=46
                     0000
                     0000     $EQU    TTY$V_SX_WRAP   47
0000002F             0000             TTY$V_SX_WRAP=47
                     0000
                     0000     $EQU    TTY$V_SX_OVRFLO 48
00000030             0000             TTY$V_SX_OVRFLO=48
                     0000
                     0000     $EQU    TTY$V_SX_AUTOP  49
00000031             0000             TTY$V_SX_AUTOP=49
                     0000
                     0000     $EQU    TTY$V_SX_CTRLR  50
00000032             0000             TTY$V_SX_CTRLR=50
                     0000
                     0000     $EQU    TTY$V_SX_SKIPCRLF       51
00000033             0000             TTY$V_SX_SKIPCRLF=51
                     0000
                     0000     $EQU    TTY$V_SX_EDITING        52
00000034             0000             TTY$V_SX_EDITING=52
                     0000
                     0000     $EQU    TTY$V_SX_TABEXPAND      53
00000035             0000             TTY$V_SX_TABEXPAND=53
                     0000
                     0000     $EQU    TTY$V_SX_QUOTING        54
00000036             0000             TTY$V_SX_QUOTING=54
                     0000
                     0000     $EQU    TTY$V_SX_OVERSTRIKE     55
00000037             0000             TTY$V_SX_OVERSTRIKE=55
                     0000
                     0000     $EQU    TTY$V_SX_TERMNORM       56
00000038             0000             TTY$V_SX_TERMNORM=56
                     0000
                     0000     $EQU    TTY$V_SX_ECHAES 57
00000039             0000             TTY$V_SX_ECHAES=57
                     0000
                     0000     $EQU    TTY$V_SX_PRE    58
0000003A             0000             TTY$V_SX_PRE=58
                     0000
                     0000     $EQU    TTY$V_SX_NINTMULTI      59
0000003B             0000             TTY$V_SX_NINTMULTI=59
                     0000
                     0000     $EQU    TTY$V_SX_RECONNECT      60
0000003C             0000             TTY$V_SX_RECONNECT=60
                     0000
                     0000     $EQU    TTY$V_SX_CTSLOW 61
0000003D             0000             TTY$V_SX_CTSLOW=61
                     0000
                     0000     $EQU    TTY$V_SX_TABRIGHT       62
0000003E             0000             TTY$V_SX_TABRIGHT=62
```

```
                     0000
                     0000          $EQU    UCB$S_RTTUCB    136
00000088             0000                  UCB$S_RTTUCB=136
                     0000
                     0000          $EQU    UCB$R_RTTUCB    176
000000B0             0000                  UCB$R_RTTUCB=176
                     0000
                     0000          $EQU    UCB$L_RTT_NETUCB        176
000000B0             0000                  UCB$L_RTT_NETUCB=176
                     0000
                     0000          $EQU    UCB$L_RTT_NETWIND       180
000000B4             0000                  UCB$L_RTT_NETWIND=180
                     0000
                     0000          $EQU    UCB$L_RTT_IRPFL 184
000000B8             0000                  UCB$L_RTT_IRPFL=184
                     0000
                     0000          $EQU    UCB$L_RTT_IRPBL 188
000000BC             0000                  UCB$L_RTT_IRPBL=188
                     0000
                     0000          $EQU    UCB$L_RTT_NETIRP        192
000000C0             0000                  UCB$L_RTT_NETIRP=192
                     0000
                     0000          $EQU    UCB$L_RTT_BANDINCL      196
000000C4             0000                  UCB$L_RTT_BANDINCL=196
                     0000
                     0000          $EQU    UCB$L_RTT_BANDINMSK     200
000000C8             0000                  UCB$L_RTT_BANDINMSK=200
                     0000
                     0000          $EQU    UCB$L_RTT_BANDEXCL      204
000000CC             0000                  UCB$L_RTT_BANDEXCL=204
                     0000
                     0000          $EQU    UCB$L_RTT_BANDEXMSK     208
000000D0             0000                  UCB$L_RTT_BANDEXMSK=208
                     0000
                     0000          $EQU    UCB$B_RTT_PROVRS        212
000000D4             0000                  UCB$B_RTT_PROVRS=212
                     0000
                     0000          $EQU    UCB$B_RTT_PROECO        213
000000D5             0000                  UCB$B_RTT_PROECO=213
                     0000
                     0000          $EQU    UCB$W_RTT_LINK  214
000000D6             0000                  UCB$W_RTT_LINK=214
                     0000
                     0000          $EQU    UCB$B_RTT_OBJ   216
000000D8             0000                  UCB$B_RTT_OBJ=216
                     0000
                     0000          $EQU    UCB$W_RTT_SYSTYPE       217
000000D9             0000                  UCB$W_RTT_SYSTYPE=217
                     0000
                     0000          $EQU    UCB$B_RTT_FILLBYTE      219
000000DB             0000                  UCB$B_RTT_FILLBYTE=219
                     0000
                     0000          $EQU    UCB$W_CT_FLAGS  220
000000DC             0000                  UCB$W_CT_FLAGS=220
                     0000
                     0000          $EQU    FLG$V_WIIRP_BSY 0
00000000             0000                  FLG$V_WIIRP_BSY=0
```

```
                     0000
                     0000          $EQU    FLG$V_CTRLO     1
00000001             0000                  FLG$V_CTRLO=1
                     0000
                     0000          $EQU    FLG$V_CANCTRLO  2
00000002             0000                  FLG$V_CANCTRLO=2
                     0000
                     0000          $EQU    FLG$V_INWRTFDT  3
00000003             0000                  FLG$V_INWRTFDT=3
                     0000
                     0000          $EQU    FLG$V_QUOTA     4
00000004             0000                  FLG$V_QUOTA=4
                     0000
                     0000          $EQU    FLG$V_VAXTOVAX  5
00000005             0000                  FLG$V_VAXTOVAX=5
                     0000
                     0000          $EQU    FLG$V_BUFFER    6
00000006             0000                  FLG$V_BUFFER=6
                     0000
                     0000          $EQU    UCB$W_CT_QCTPCNT        222
000000DE             0000                  UCB$W_CT_QCTPCNT=222
                     0000
                     0000          $EQU    UCB$L_CT_WIIRP  224
000000E0             0000                  UCB$L_CT_WIIRP=224
                     0000
                     0000          $EQU    UCB$L_CT_TQE    228
000000E4             0000                  UCB$L_CT_TQE=228
                     0000
                     0000          $EQU    UCB$L_CT_NETQFL 232
000000E8             0000                  UCB$L_CT_NETQFL=232
                     0000
                     0000          $EQU    UCB$L_CT_NETQBL 236
000000EC             0000                  UCB$L_CT_NETQBL=236
                     0000
                     0000          $EQU    UCB$L_CT_STALLQFL       240
000000F0             0000                  UCB$L_CT_STALLQFL=240
                     0000
                     0000          $EQU    UCB$L_CT_STALLQBL       244
000000F4             0000                  UCB$L_CT_STALLQBL=244
                     0000
                     0000          $EQU    UCB$L_CT_WRTCTP 248
000000F8             0000                  UCB$L_CT_WRTCTP=248
                     0000
                     0000          $EQU    UCB$L_CT_WRTCUR 252
000000FC             0000                  UCB$L_CT_WRTCUR=252
                     0000
                     0000          $EQU    UCB$W_CT_WRTSIZ 256
00000100             0000                  UCB$W_CT_WRTSIZ=256
                     0000
                     0000          $EQU    UCB$W_CT_WRTCNT 258
00000102             0000                  UCB$W_CT_WRTCNT=258
                     0000
                     0000          $EQU    UCB$W_CT_MAXMSG 260
00000104             0000                  UCB$W_CT_MAXMSG=260
                     0000
                     0000          $EQU    UCB$W_CT_MAXREAD        262
00000106             0000                  UCB$W_CT_MAXREAD=262
```

```
                     0000
                     0000          $EQU    UCB$L_CT_LEGALMSG       264
00000108             0000                  UCB$L_CT_LEGALMSG=264
                     0000
                     0000          $EQU    UCB$B_CT_VERSION        268
0000010C             0000                  UCB$B_CT_VERSION=268
                     0000
                     0000          $EQU    UCB$B_CT_ECO    269
0000010D             0000                  UCB$B_CT_ECO=269
                     0000
                     0000          $EQU    UCB$W_CT_FILLWORD       270
0000010E             0000                  UCB$W_CT_FILLWORD=270
                     0000
                     0000          $EQU    UCB$S_CT_DEBUG_FILL     40
00000028             0000                  UCB$S_CT_DEBUG_FILL=40
                     0000
                     0000          $EQU    UCB$T_CT_DEBUG_FILL     272
00000110             0000                  UCB$T_CT_DEBUG_FILL=272
                     0000
                     0000                  $DEFEND TTYUCB,,DEF
                     0000                  .MACRO  $TTYUCBDEF A
                     0000                  .ENDM   $TTYUCBDEF
                     0000                  .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                     0000                  .CROSS
                     0000                  .RESTORE
                     0000
                     0000
                     0000                  $TTYVECDEF
                     0000                  $DEFINI TTYVEC,
                     0000                  .SAVE   LOCAL_BLOCK
                     0000                  .NOCROSS
                     0000                  .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
                     0000                  .PSECT  $ABS$,ABS
                     0000                  $GBLINI
                     0000                  .IF     IDN <LOCAL> <GLOBAL>
                     0000                  .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000                  .IIF    NB,SYM, SYM::
                     0000                  .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000                  .ENDM   $DEF
                     0000                  .MACRO  $EQU    SYM,VAL
                     0000                  SYM==VAL
                     0000                  .ENDM   $EQU
                     0000                  .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000                  SIZ...=1
                     0000                  .IIF    NB,SIZ, SIZ...=SIZ
                     0000                  .IF     NB,SYM
                     0000                  MOD'SEP'V_'SYM==BIT...
                     0000                  .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
                     0000                  .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
                     0000                  .ENDC
                     0000                  BIT...=BIT...+SIZ...
                     0000                  .ENDM   $VIELD1
                     0000                  .IFF
                     0000                  .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
                     0000                  .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000                  .IIF    NB,SYM, SYM:
                     0000                  .IIF    NB,ALLOC,       ALLOC   SIZ
```

```
                     0000              .ENDM   $DEF
                     0000              .MACRO  $EQU    SYM,VAL
                     0000              SYM=VAL
                     0000              .ENDM   $EQU
                     0000              .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000              SIZ...=1
                     0000              .IIF    NB,SIZ, SIZ...=SIZ
                     0000              .IF     NB,SYM
                     0000              MOD'SEP'V_'SYM=BIT...
                     0000              .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000              .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000              .ENDC
                     0000              BIT...=BIT...+SIZ...
                     0000              .ENDM   $VIELD1
                     0000              .ENDC
                     0000
00000000             0000              .=0
                     0000
                     0000    $EQU      CLASS_CLASS_DEF         40
00000028             0000              CLASS_CLASS_DEF=40
                     0000
                     0000    $EQU      CLASS_GETNXT    0
00000000             0000              CLASS_GETNXT=0
                     0000
                     0000    $EQU      CLASS_PUTNXT    4
00000004             0000              CLASS_PUTNXT=4
                     0000
                     0000    $EQU      CLASS_SETUP_UCB 8
00000008             0000              CLASS_SETUP_UCB=8
                     0000
                     0000    $EQU      CLASS_DS_TRAN   12
0000000C             0000              CLASS_DS_TRAN=12
                     0000
                     0000    $EQU      CLASS_DDT       16
00000010             0000              CLASS_DDT=16
                     0000
                     0000    $EQU      CLASS_READERROR 20
00000014             0000              CLASS_READERROR=20
                     0000
                     0000    $EQU      CLASS_DISCONNECT        24
00000018             0000              CLASS_DISCONNECT=24
                     0000
                     0000    $EQU      CLASS_FORK      28
0000001C             0000              CLASS_FORK=28
                     0000
                     0000    $EQU      CLASS_POWERFAIL 32
00000020             0000              CLASS_POWERFAIL=32
                     0000
                     0000    $EQU      CLASS_TABLES    36
00000024             0000              CLASS_TABLES=36
                     0000
                     0000    $EQU      PORT_LENGTH     56
00000038             0000              PORT_LENGTH=56
                     0000
                     0000    $EQU      PORTS_PORT_DEF  56
00000038             0000              PORTS_PORT_DEF=56
                     0000
```

```
                     0000          $EQU    PORT_STARTIO    0
00000000             0000                  PORT_STARTIO=0
                     0000
                     0000          $EQU    PORT_DISCONNECT 4
00000004             0000                  PORT_DISCONNECT=4
                     0000
                     0000          $EQU    PORT_SET_LINE   8
00000008             0000                  PORT_SET_LINE=8
                     0000
                     0000          $EQU    PORT_DS_SET     12
0000000C             0000                  PORT_DS_SET=12
                     0000
                     0000          $EQU    PORT_XON        16
00000010             0000                  PORT_XON=16
                     0000
                     0000          $EQU    PORT_XOFF       20
00000014             0000                  PORT_XOFF=20
                     0000
                     0000          $EQU    PORT_STOP       24
00000018             0000                  PORT_STOP=24
                     0000
                     0000          $EQU    PORT_STOP2      28
0000001C             0000                  PORT_STOP2=28
                     0000
                     0000          $EQU    PORT_ABORT      32
00000020             0000                  PORT_ABORT=32
                     0000
                     0000          $EQU    PORT_RESUME     36
00000024             0000                  PORT_RESUME=36
                     0000
                     0000          $EQU    PORT_SET_MODEM  40
00000028             0000                  PORT_SET_MODEM=40
                     0000
                     0000          $EQU    PORT__DMA       44
0000002C             0000                  PORT__DMA=44
                     0000
                     0000          $EQU    PORT_MAINT      48
00000030             0000                  PORT_MAINT=48
                     0000
                     0000          $EQU    PORT_FORKRET    52
00000034             0000                  PORT_FORKRET=52
                     0000
                     0000                  $DEFEND TTYVEC,,DEF
                     0000                  .MACRO  $TTYVECDEF A
                     0000                  .ENDM   $TTYVECDEF
                     0000                  .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                     0000                  .CROSS
                     0000                  .RESTORE
                     0000
                     0000
                     0000                  $TTYSYMDEF
                     0000                  $DEFINI TTYSYM,
                     0000                  .SAVE   LOCAL_BLOCK
                     0000                  .NOCROSS
                     0000                  .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
                     0000                  .PSECT  $ABS$,ABS
                     0000                  $GBLINI
```

```
                0000            .IF     IDN <LOCAL> <GLOBAL>
                0000            .MACRO  $DEF    SYM,ALLOC,SIZ
                0000            .IIF    NB,SYM, SYM::
                0000            .IIF    NB,ALLOC,       ALLOC   SIZ
                0000            .ENDM   $DEF
                0000            .MACRO  $EQU    SYM,VAL
                0000            SYM==VAL
                0000            .ENDM   $EQU
                0000            .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                0000            SIZ...=1
                0000            .IIF    NB,SIZ, SIZ...=SIZ
                0000            .IF     NB,SYM
                0000            MOD'SEP'V_'SYM==BIT...
                0000            .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
                0000            .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
                0000            .ENDC
                0000            BIT...=BIT...+SIZ...
                0000            .ENDM   $VIELD1
                0000            .IFF
                0000            .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
                0000            .MACRO  $DEF    SYM,ALLOC,SIZ
                0000            .IIF    NB,SYM, SYM:
                0000            .IIF    NB,ALLOC,       ALLOC   SIZ
                0000            .ENDM   $DEF
                0000            .MACRO  $EQU    SYM,VAL
                0000            SYM=VAL
                0000            .ENDM   $EQU
                0000            .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                0000            SIZ...=1
                0000            .IIF    NB,SIZ, SIZ...=SIZ
                0000            .IF     NB,SYM
                0000            MOD'SEP'V_'SYM=BIT...
                0000            .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                0000            .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                0000            .ENDC
                0000            BIT...=BIT...+SIZ...
                0000            .ENDM   $VIELD1
                0000            .ENDC
                0000
00000000        0000            .=0
                0000
                0000    $EQU    TTY$M_FD_UNSOL  1
00000001        0000            TTY$M_FD_UNSOL=1
                0000
                0000    $EQU    TTY$M_FD_GETAHD 2
00000002        0000            TTY$M_FD_GETAHD=2
                0000
                0000    $EQU    TTY$M_FD_DISCONNECT     4
00000004        0000            TTY$M_FD_DISCONNECT=4
                0000
                0000    $EQU    TTY$M_FD_PORTFORK       8
00000008        0000            TTY$M_FD_PORTFORK=8
                0000
                0000    $EQU    TTY$M_FD_UNLINK 16
00000010        0000            TTY$M_FD_UNLINK=16
                0000
                0000    $EQU    TTY$M_FD_LINK   32
```

```
00000020  0000                    TTY$M_FD_LINK=32
          0000
          0000          $EQU      TTY$M_FD_BUSY   64
00000040  0000                    TTY$M_FD_BUSY=64
          0000
          0000          $EQU      TTY$S_FORK      1
00000001  0000                    TTY$S_FORK=1
          0000
          0000          $EQU      TTY$V_FD_UNSOL  0
00000000  0000                    TTY$V_FD_UNSOL=0
          0000
          0000          $EQU      TTY$V_FD_GETAHD 1
00000001  0000                    TTY$V_FD_GETAHD=1
          0000
          0000          $EQU      TTY$V_FD_DISCONNECT     2
00000002  0000                    TTY$V_FD_DISCONNECT=2
          0000
          0000          $EQU      TTY$V_FD_PORTFORK       3
00000003  0000                    TTY$V_FD_PORTFORK=3
          0000
          0000          $EQU      TTY$V_FD_UNLINK 4
00000004  0000                    TTY$V_FD_UNLINK=4
          0000
          0000          $EQU      TTY$V_FD_LINK   5
00000005  0000                    TTY$V_FD_LINK=5
          0000
          0000          $EQU      TTY$V_FD_BUSY   6
00000006  0000                    TTY$V_FD_BUSY=6
          0000
          0000          $EQU      TTY$C_CTRLA     1
00000001  0000                    TTY$C_CTRLA=1
          0000
          0000          $EQU      TTY$C_CTRLB     2
00000002  0000                    TTY$C_CTRLB=2
          0000
          0000          $EQU      TTY$C_CTRLC     3
00000003  0000                    TTY$C_CTRLC=3
          0000
          0000          $EQU      TTY$C_CTRLD     4
00000004  0000                    TTY$C_CTRLD=4
          0000
          0000          $EQU      TTY$C_CTRLE     5
00000005  0000                    TTY$C_CTRLE=5
          0000
          0000          $EQU      TTY$C_CTRLF     6
00000006  0000                    TTY$C_CTRLF=6
          0000
          0000          $EQU      TTY$C_BELL      7
00000007  0000                    TTY$C_BELL=7
          0000
          0000          $EQU      TTY$C_BS        8
00000008  0000                    TTY$C_BS=8
          0000
          0000          $EQU      TTY$C_TAB       9
00000009  0000                    TTY$C_TAB=9
          0000
          0000          $EQU      TTY$C_LF        10
```

```
                     0000
0000000A             0000          TTY$C_LF=10
                     0000
                     0000          $EQU    TTY$C_VT        11
0000000B             0000          TTY$C_VT=11
                     0000
                     0000          $EQU    TTY$C_FF        12
0000000C             0000          TTY$C_FF=12
                     0000
                     0000          $EQU    TTY$C_CR        13
0000000D             0000          TTY$C_CR=13
                     0000
                     0000          $EQU    TTY$C_CTRLN     14
0000000E             0000          TTY$C_CTRLN=14
                     0000
                     0000          $EQU    TTY$C_CTRLO     15
0000000F             0000          TTY$C_CTRLO=15
                     0000
                     0000          $EQU    TTY$C_CTRLP     16
00000010             0000          TTY$C_CTRLP=16
                     0000
                     0000          $EQU    TTY$C_CTRLQ     17
00000011             0000          TTY$C_CTRLQ=17
                     0000
                     0000          $EQU    TTY$C_CTRLR     18
00000012             0000          TTY$C_CTRLR=18
                     0000
                     0000          $EQU    TTY$C_CTRLS     19
00000013             0000          TTY$C_CTRLS=19
                     0000
                     0000          $EQU    TTY$C_CTRLT     20
00000014             0000          TTY$C_CTRLT=20
                     0000
                     0000          $EQU    TTY$C_CTRLU     21
00000015             0000          TTY$C_CTRLU=21
                     0000
                     0000          $EQU    TTY$C_CTRLV     22
00000016             0000          TTY$C_CTRLV=22
                     0000
                     0000          $EQU    TTY$C_CTRLW     23
00000017             0000          TTY$C_CTRLW=23
                     0000
                     0000          $EQU    TTY$C_CTRLX     24
00000018             0000          TTY$C_CTRLX=24
                     0000
                     0000          $EQU    TTY$C_CTRLY     25
00000019             0000          TTY$C_CTRLY=25
                     0000
                     0000          $EQU    TTY$C_CTRLZ     26
0000001A             0000          TTY$C_CTRLZ=26
                     0000
                     0000          $EQU    TTY$C_ESCAPE    27
0000001B             0000          TTY$C_ESCAPE=27
                     0000
                     0000          $EQU    TTY$C_XON       17
00000011             0000          TTY$C_XON=17
                     0000
                     0000          $EQU    TTY$C_XOFF      19
```

```
00000013  0000                    TTY$C_XOFF=19
          0000
          0000          $EQU      TTY$C_BLANK      32
00000020  0000                    TTY$C_BLANK=32
          0000
          0000          $EQU      TTY$C_DOLLAR     36
00000024  0000                    TTY$C_DOLLAR=36
          0000
          0000          $EQU      TTY$C_PLUS       43
0000002B  0000                    TTY$C_PLUS=43
          0000
          0000          $EQU      TTY$C_ZERO       48
00000030  0000                    TTY$C_ZERO=48
          0000
          0000          $EQU      TTY$C_ONE        49
00000031  0000                    TTY$C_ONE=49
          0000
          0000          $EQU      TTY$C_SCRIPT     96
00000060  0000                    TTY$C_SCRIPT=96
          0000
          0000          $EQU      TTY$C_LOWA       97
00000061  0000                    TTY$C_LOWA=97
          0000
          0000          $EQU      TTY$C_LOWZ       123
0000007B  0000                    TTY$C_LOWZ=123
          0000
          0000          $EQU      TTY$C_LOWESC1    125
0000007D  0000                    TTY$C_LOWESC1=125
          0000
          0000          $EQU      TTY$C_LOWESC2    126
0000007E  0000                    TTY$C_LOWESC2=126
          0000
          0000          $EQU      TTY$C_DELETE     127
0000007F  0000                    TTY$C_DELETE=127
          0000
          0000          $EQU      TTY$C_NL         128
00000080  0000                    TTY$C_NL=128
          0000
          0000          $EQU      TTY$C_CSI        155
0000009B  0000                    TTY$C_CSI=155
          0000
          0000          $EQU      TTY$C_MAXPAGLEN  255
000000FF  0000                    TTY$C_MAXPAGLEN=255
          0000
          0000          $EQU      TTY$C_MAXPAGWID  511
000001FF  0000                    TTY$C_MAXPAGWID=511
          0000
          0000          $EQU      TTY$C_HIGHIPL    22
00000016  0000                    TTY$C_HIGHIPL=22
          0000
          0000          $EQU      TTY$K_ER_NORMAL  0
00000000  0000                    TTY$K_ER_NORMAL=0
          0000
          0000          $EQU      TTY$K_ER_CLRECHO         1
00000001  0000                    TTY$K_ER_CLRECHO=1
          0000
          0000          $EQU      TTY$K_ER_ECHLINE         2
```

```
                     0000
00000002             0000          TTY$K_ER_ECHLINE=2
                     0000
                     0000     $EQU TTY$K_ER_UPDCURSOR      3
00000003             0000          TTY$K_ER_UPDCURSOR=3
                     0000
                     0000     $EQU TTY$K_ER_EXITING        4
00000004             0000          TTY$K_ER_EXITING=4
                     0000
                     0000     $EQU TTY$K_ER_MOVECURSOR     5
00000005             0000          TTY$K_ER_MOVECURSOR=5
                     0000
                     0000     $EQU TTY$K_ER_CLRREST        6
00000006             0000          TTY$K_ER_CLRREST=6
                     0000
                     0000     $EQU TTY$K_ER_PRMECHO        7
00000007             0000          TTY$K_ER_PRMECHO=7
                     0000
                     0000     $EQU TTY$K_ER_PRMECHO1       8
00000008             0000          TTY$K_ER_PRMECHO1=8
                     0000
                     0000     $EQU TTY$K_ER_AESECHO        9
00000009             0000          TTY$K_ER_AESECHO=9
                     0000
                     0000     $EQU TTY$K_ER_RVECHO 10
0000000A             0000          TTY$K_ER_RVECHO=10
                     0000
                     0000     $EQU TTY$K_ER_SIMCEOL        11
0000000B             0000          TTY$K_ER_SIMCEOL=11
                     0000
                     0000     $EQU TTY$K_ET_CTRLU  1
00000001             0000          TTY$K_ET_CTRLU=1
                     0000
                     0000     $EQU TTY$K_ET_CTRLR  2
00000002             0000          TTY$K_ET_CTRLR=2
                     0000
                     0000     $EQU TTY$K_ET_DELEFT 3
00000003             0000          TTY$K_ET_DELEFT=3
                     0000
                     0000     $EQU TTY$K_ET_ESCAPE 4
00000004             0000          TTY$K_ET_ESCAPE=4
                     0000
                     0000     $EQU TTY$K_ET_BACK_CHAR      5
00000005             0000          TTY$K_ET_BACK_CHAR=5
                     0000
                     0000     $EQU TTY$K_ET_FORWARD_CHAR   6
00000006             0000          TTY$K_ET_FORWARD_CHAR=6
                     0000
                     0000     $EQU TTY$K_ET_MOVE_EOL       7
00000007             0000          TTY$K_ET_MOVE_EOL=7
                     0000
                     0000     $EQU TTY$K_ET_MOVE_BOL       8
00000008             0000          TTY$K_ET_MOVE_BOL=8
                     0000
                     0000     $EQU TTY$K_ET_DELETE_WORD    9
00000009             0000          TTY$K_ET_DELETE_WORD=9
                     0000
                     0000     $EQU TTY$K_ET_QUOTING        10
```

```
                        0000
0000000A                0000            TTY$K_ET_QUOTING=10
                        0000
                        0000    $EQU    TTY$K_ET_RECALL 11
0000000B                0000            TTY$K_ET_RECALL=11
                        0000
                        0000    $EQU    TTY$K_ET_TOGGEL 12
0000000C                0000            TTY$K_ET_TOGGEL=12
                        0000
                        0000    $EQU    TTY$K_ET_UNUSED 13
0000000D                0000            TTY$K_ET_UNUSED=13
                        0000
                        0000    $EQU    TTY$K_ET_TERMINATE      14
0000000E                0000            TTY$K_ET_TERMINATE=14
                        0000
                        0000    $EQU    TTY$K_EDITNORMAL        4
00000004                0000            TTY$K_EDITNORMAL=4
                        0000
                        0000    $EQU    TTY$C_FC_READ   0
00000000                0000            TTY$C_FC_READ=0
                        0000
                        0000    $EQU    TTY$C_FC_WRITE  1
00000001                0000            TTY$C_FC_WRITE=1
                        0000
                        0000    $EQU    TTY$C_FC_SETM   2
00000002                0000            TTY$C_FC_SETM=2
                        0000
                        0000    $EQU    TTY$C_FC_SETC   3
00000003                0000            TTY$C_FC_SETC=3
                        0000
                        0000    $EQU    TTY$C_FC_HANGUP 4
00000004                0000            TTY$C_FC_HANGUP=4
                        0000
                        0000    $EQU    TTY$C_FC_MAINT  5
00000005                0000            TTY$C_FC_MAINT=5
                        0000
                        0000    $EQU    TTY$C_FC_CTRL   6
00000006                0000            TTY$C_FC_CTRL=6
                        0000
                        0000    $EQU    TTY$C_FC_CONNECT        7
00000007                0000            TTY$C_FC_CONNECT=7
                        0000
                        0000    $EQU    TTY$C_FC_DISCON 8
00000008                0000            TTY$C_FC_DISCON=8
                        0000
                        0000    $EQU    TTY$M_CH_LOWER  8
00000008                0000            TTY$M_CH_LOWER=8
                        0000
                        0000    $EQU    TTY$M_CH_SPEC   16
00000010                0000            TTY$M_CH_SPEC=16
                        0000
                        0000    $EQU    TTY$M_CH_CTRL   32
00000020                0000            TTY$M_CH_CTRL=32
                        0000
                        0000    $EQU    TTY$M_CH_CTRL3  64
00000040                0000            TTY$M_CH_CTRL3=64
                        0000
                        0000    $EQU    TTY$M_CH_CTRL2  128
```

```
00000080  0000              TTY$M_CH_CTRL2=128
          0000
          0000      $EQU    TTY$S_CHAR_CHAR 1
00000001  0000              TTY$S_CHAR_CHAR=1
          0000
          0000      $EQU    TTY$S_CH_FILL   3
00000003  0000              TTY$S_CH_FILL=3
          0000
          0000      $EQU    TTY$V_CH_FILL   0
00000000  0000              TTY$V_CH_FILL=0
          0000
          0000      $EQU    TTY$V_CH_LOWER  3
00000003  0000              TTY$V_CH_LOWER=3
          0000
          0000      $EQU    TTY$V_CH_SPEC   4
00000004  0000              TTY$V_CH_SPEC=4
          0000
          0000      $EQU    TTY$V_CH_CTRL   5
00000005  0000              TTY$V_CH_CTRL=5
          0000
          0000      $EQU    TTY$V_CH_CTRL3  6
00000006  0000              TTY$V_CH_CTRL3=6
          0000
          0000      $EQU    TTY$V_CH_CTRL2  7
00000007  0000              TTY$V_CH_CTRL2=7
          0000
          0000              $DEFEND TTYSYM,,DEF
          0000              .MACRO  $TTYSYMDEF A
          0000              .ENDM   $TTYSYMDEF
          0000              .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
          0000              .CROSS
          0000              .RESTORE
          0000
          0000
          0000              $TTYRBDEF
          0000              $DEFINI TTYRB,
          0000              .SAVE   LOCAL_BLOCK
          0000              .NOCROSS
          0000              .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
          0000              .PSECT  $ABS$,ABS
          0000              $GBLINI
          0000              .IF     IDN <LOCAL> <GLOBAL>
          0000              .MACRO  $DEF    SYM,ALLOC,SIZ
          0000              .IIF    NB,SYM, SYM::
          0000              .IIF    NB,ALLOC,       ALLOC   SIZ
          0000              .ENDM   $DEF
          0000              .MACRO  $EQU    SYM,VAL
          0000              SYM==VAL
          0000              .ENDM   $EQU
          0000              .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
          0000              SIZ...=1
          0000              .IIF    NB,SIZ, SIZ...=SIZ
          0000              .IF     NB,SYM
          0000              MOD'SEP'V_'SYM==BIT...
          0000              .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
          0000              .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
          0000              .ENDC
```

```
                     0000              BIT...=BIT...+SIZ...
                     0000              .ENDM   $VIELD1
                     0000              .IFF
                     0000              .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE ''GLOBAL'',''LOCAL'',OR NULL
                     0000              .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000              .IIF    NB,SYM, SYM:
                     0000              .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000              .ENDM   $DEF
                     0000              .MACRO  $EQU    SYM,VAL
                     0000              SYM=VAL
                     0000              .ENDM   $EQU
                     0000              .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000              SIZ...=1
                     0000              .IIF    NB,SIZ, SIZ...=SIZ
                     0000              .IF     NB,SYM
                     0000              MOD'SEP'V_'SYM=BIT...
                     0000              .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000              .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000              .ENDC
                     0000              BIT...=BIT...+SIZ...
                     0000              .ENDM   $VIELD1
                     0000              .ENDC
                     0000
00000000             0000              .=0
                     0000
                     0000    $EQU      TTY$M_RS_WRAP   1
00000001             0000              TTY$M_RS_WRAP=1
                     0000
                     0000    $EQU      TTY$S_TTYRBDEF  78
0000004E             0000              TTY$S_TTYRBDEF=78
                     0000
                     0000    $EQU      TTY$L_RB_TXT    0
00000000             0000              TTY$L_RB_TXT=0
                     0000
                     0000    $EQU      TTY$L_RB_UVA    4
00000004             0000              TTY$L_RB_UVA=4
                     0000
                     0000    $EQU      TTY$W_RB_SIZE   8
00000008             0000              TTY$W_RB_SIZE=8
                     0000
                     0000    $EQU      TTY$B_RB_SPARE1 10
0000000A             0000              TTY$B_RB_SPARE1=10
                     0000
                     0000    $EQU      TTY$B_RB_ECHLEN 11
0000000B             0000              TTY$B_RB_ECHLEN=11
                     0000
                     0000    $EQU      TTY$S_RB_ECHOAREA       8
00000008             0000              TTY$S_RB_ECHOAREA=8
                     0000
                     0000    $EQU      TTY$Q_RB_ECHOAREA       12
0000000C             0000              TTY$Q_RB_ECHOAREA=12
                     0000
                     0000    $EQU      TTY$L_RB_ECHSTR 20
00000014             0000              TTY$L_RB_ECHSTR=20
                     0000
                     0000    $EQU      TTY$L_RB_PIC    24
00000018             0000              TTY$L_RB_PIC=24
```

```
                        0000
                        0000        $EQU    TTY$L_RB_TERM   28
0000001C                0000                TTY$L_RB_TERM=28
                        0000
                        0000        $EQU    TTY$L_RB_MOD    32
00000020                0000                TTY$L_RB_MOD=32
                        0000
                        0000        $EQU    TTY$L_RB_AES    36
00000024                0000                TTY$L_RB_AES=36
                        0000
                        0000        $EQU    TTY$W_RB_AESLEN 40
00000028                0000                TTY$W_RB_AESLEN=40
                        0000
                        0000        $EQU    TTY$W_RB_RDSTATE        42
0000002A                0000                TTY$W_RB_RDSTATE=42
                        0000
                        0000        $EQU    TTY$V_RS_WRAP   0
00000000                0000                TTY$V_RS_WRAP=0
                        0000
                        0000        $EQU    TTY$L_RB_LIN    44
0000002C                0000                TTY$L_RB_LIN=44
                        0000
                        0000        $EQU    TTY$W_RB_LINOFF 48
00000030                0000                TTY$W_RB_LINOFF=48
                        0000
                        0000        $EQU    TTY$W_RB_LINREST        50
00000032                0000                TTY$W_RB_LINREST=50
                        0000
                        0000        $EQU    TTY$W_RB_PRMLEN 52
00000034                0000                TTY$W_RB_PRMLEN=52
                        0000
                        0000        $EQU    TTY$W_RB_TIMOS  54
00000036                0000                TTY$W_RB_TIMOS=54
                        0000
                        0000        $EQU    TTY$W_RB_CPZCUR 56
00000038                0000                TTY$W_RB_CPZCUR=56
                        0000
                        0000        $EQU    TTY$W_RB_CPZORG 58
0000003A                0000                TTY$W_RB_CPZORG=58
                        0000
                        0000        $EQU    TTY$W_RB_TXTOFF 60
0000003C                0000                TTY$W_RB_TXTOFF=60
                        0000
                        0000        $EQU    TTY$W_RB_PICLEN 62
0000003E                0000                TTY$W_RB_PICLEN=62
                        0000
                        0000        $EQU    TTY$W_RB_TXTSIZ 64
00000040                0000                TTY$W_RB_TXTSIZ=64
                        0000
                        0000        $EQU    TTY$W_RB_TXTECH 66
00000042                0000                TTY$W_RB_TXTECH=66
                        0000
                        0000        $EQU    TTY$W_RB_MODE   68
00000044                0000                TTY$W_RB_MODE=68
                        0000
                        0000        $EQU    TTY$B_RB_RVFCLR 70
00000046                0000                TTY$B_RB_RVFCLR=70
```

```
                     0000
                     0000          $EQU    TTY$B_RB_RVFFIL 71
00000047             0000                  TTY$B_RB_RVFFIL=71
                     0000
                     0000          $EQU    TTY$W_RB_ESCTKN 72
00000048             0000                  TTY$W_RB_ESCTKN=72
                     0000
                     0000          $EQU    TTY$A_RB_PRM    74
0000004A             0000                  TTY$A_RB_PRM=74
                     0000
                     0000          $EQU    TTY$L_RB_DATA   74
0000004A             0000                  TTY$L_RB_DATA=74
                     0000
                     0000                  $DEFEND TTYRB,,DEF
                     0000                  .MACRO  $TTYRBDEF A
                     0000                  .ENDM   $TTYRBDEF
                     0000                  .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                     0000                  .CROSS
                     0000                  .RESTORE
                     0000
                     0000
                     0000                  $TTYISDEF
                     0000                  $DEFINI TTYIS,
                     0000                  .SAVE   LOCAL_BLOCK
                     0000                  .NOCROSS
                     0000                  .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
                     0000                  .PSECT  $ABS$,ABS
                     0000                  $GBLINI
                     0000                  .IF     IDN <LOCAL> <GLOBAL>
                     0000                  .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000                  .IIF    NB,SYM, SYM::
                     0000                  .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000                  .ENDM   $DEF
                     0000                  .MACRO  $EQU    SYM,VAL
                     0000                  SYM==VAL
                     0000                  .ENDM   $EQU
                     0000                  .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000                  SIZ...=1
                     0000                  .IIF    NB,SIZ, SIZ...=SIZ
                     0000                  .IF     NB,SYM
                     0000                  MOD'SEP'V_'SYM==BIT...
                     0000                  .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
                     0000                  .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
                     0000                  .ENDC
                     0000                  BIT...=BIT...+SIZ...
                     0000                  .ENDM   $VIELD1
                     0000                  .IFF
                     0000                  .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
                     0000                  .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000                  .IIF    NB,SYM, SYM:
                     0000                  .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000                  .ENDM   $DEF
                     0000                  .MACRO  $EQU    SYM,VAL
                     0000                  SYM=VAL
                     0000                  .ENDM   $EQU
                     0000                  .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000                  SIZ...=1
```

```
                     0000              .IIF    NB,SIZ, SIZ...=SIZ
                     0000              .IF     NB,SYM
                     0000              MOD'SEP'V_'SYM=BIT...
                     0000              .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000              .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000              .ENDC
                     0000              BIT...=BIT...+SIZ...
                     0000              .ENDM   $VIELD1
                     0000              .ENDC
                     0000
00000000             0000              .=0
                     0000
                     0000      $EQU    TTY$K_IS_LENGTH 88
00000058             0000              TTY$K_IS_LENGTH=88
                     0000
                     0000      $EQU    TTY$S_TTYISDEF  88
00000058             0000              TTY$S_TTYISDEF=88
                     0000
                     0000      $EQU    TTY$L_IS_ACMODE 0
00000000             0000              TTY$L_IS_ACMODE=0
                     0000
                     0000      $EQU    TTY$L_IS_EDITMODE       4
00000004             0000              TTY$L_IS_EDITMODE=4
                     0000
                     0000      $EQU    TTY$L_IS_BUF    8
00000008             0000              TTY$L_IS_BUF=8
                     0000
                     0000      $EQU    TTY$L_IS_BUFLEN 12
0000000C             0000              TTY$L_IS_BUFLEN=12
                     0000
                     0000      $EQU    TTY$L_IS_INI    16
00000010             0000              TTY$L_IS_INI=16
                     0000
                     0000      $EQU    TTY$L_IS_INILEN 20
00000014             0000              TTY$L_IS_INILEN=20
                     0000
                     0000      $EQU    TTY$L_IS_INIBUF 24
00000018             0000              TTY$L_IS_INIBUF=24
                     0000
                     0000      $EQU    TTY$L_IS_ITMLST 28
0000001C             0000              TTY$L_IS_ITMLST=28
                     0000
                     0000      $EQU    TTY$L_IS_LASTITM        32
00000020             0000              TTY$L_IS_LASTITM=32
                     0000
                     0000      $EQU    TTY$L_IS_MODIFY 36
00000024             0000              TTY$L_IS_MODIFY=36
                     0000
                     0000      $EQU    TTY$L_IS_PIC    40
00000028             0000              TTY$L_IS_PIC=40
                     0000
                     0000      $EQU    TTY$L_IS_PICLEN 44
0000002C             0000              TTY$L_IS_PICLEN=44
                     0000
                     0000      $EQU    TTY$L_IS_PRM    48
00000030             0000              TTY$L_IS_PRM=48
                     0000
```

```
                     0000          $EQU    TTY$L_IS_PRMLEN 52
00000034             0000                  TTY$L_IS_PRMLEN=52
                     0000
                     0000          $EQU    TTY$L_IS_PRMBUF 56
00000038             0000                  TTY$L_IS_PRMBUF=56
                     0000
                     0000          $EQU    TTY$L_IS_SPECIFYED      60
0000003C             0000                  TTY$L_IS_SPECIFYED=60
                     0000
                     0000          $EQU    TTY$L_IS_TERM   64
00000040             0000                  TTY$L_IS_TERM=64
                     0000
                     0000          $EQU    TTY$L_IS_TERMLEN        68
00000044             0000                  TTY$L_IS_TERMLEN=68
                     0000
                     0000          $EQU    TTY$L_IS_AES    72
00000048             0000                  TTY$L_IS_AES=72
                     0000
                     0000          $EQU    TTY$L_IS_AESLEN 76
0000004C             0000                  TTY$L_IS_AESLEN=76
                     0000
                     0000          $EQU    TTY$L_IS_TIMEOUT        80
00000050             0000                  TTY$L_IS_TIMEOUT=80
                     0000
                     0000          $EQU    TTY$W_IS_FILLCHR        84
00000054             0000                  TTY$W_IS_FILLCHR=84
                     0000
                     0000          $EQU    TTY$W_IS_INIOFF 86
00000056             0000                  TTY$W_IS_INIOFF=86
                     0000
                     0000          $DEFEND TTYIS,,DEF
                     0000          .MACRO  $TTYISDEF A
                     0000          .ENDM   $TTYISDEF
                     0000          .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                     0000          .CROSS
                     0000          .RESTORE
                     0000
                     0000
                     0000          $TTYILDEF
                     0000          $DEFINI TTYIL,
                     0000          .SAVE   LOCAL_BLOCK
                     0000          .NOCROSS
                     0000          .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
                     0000          .PSECT  $ABS$,ABS
                     0000          $GBLINI
                     0000          .IF     IDN <LOCAL> <GLOBAL>
                     0000          .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000          .IIF    NB,SYM, SYM::
                     0000          .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000          .ENDM   $DEF
                     0000          .MACRO  $EQU    SYM,VAL
                     0000          SYM==VAL
                     0000          .ENDM   $EQU
                     0000          .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000          SIZ...=1
                     0000          .IIF    NB SIZ, SIZ...=SIZ
                     0000          .IF     NB,SYM
```

```
                     0000               MOD'SEP'V_'SYM==BIT...
                     0000               .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
                     0000               .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
                     0000               .ENDC
                     0000               BIT...=BIT...+SIZ...
                     0000               .ENDM   $VIELD1
                     0000               .IFF
                     0000               .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
                     0000               .MACRO  $DEF    SYM,ALLOC,SIZ
                     0000               .IIF    NB,SYM, SYM:
                     0000               .IIF    NB,ALLOC,       ALLOC   SIZ
                     0000               .ENDM   $DEF
                     0000               .MACRO  $EQU    SYM,VAL
                     0000               SYM=VAL
                     0000               .ENDM   $EQU
                     0000               .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
                     0000               SIZ...=1
                     0000               .IIF    NB,SIZ, SIZ...=SIZ
                     0000               .IF     NB,SYM
                     0000               MOD'SEP'V_'SYM=BIT...
                     0000               .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
                     0000               .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
                     0000               .ENDC
                     0000               BIT...=BIT...+SIZ...
                     0000               .ENDM   $VIELD1
                     0000               .ENDC
                     0000
00000000             0000               .=0
                     0000
                     0000       $EQU    TTY$K_IL_LENGTH 12
0000000C             0000               TTY$K_IL_LENGTH=12
                     0000
                     0000       $EQU    TTY$S_TTYILDEF  12
0000000C             0000               TTY$S_TTYILDEF=12
                     0000
                     0000       $EQU    TTY$W_IL_LEN    0
00000000             0000               TTY$W_IL_LEN=0
                     0000
                     0000       $EQU    TTY$W_IL_TYPE   2
00000002             0000               TTY$W_IL_TYPE=2
                     0000
                     0000       $EQU    TTY$L_IL_ADR    4
00000004             0000               TTY$L_IL_ADR=4
                     0000
                     0000       $EQU    TTY$L_IL_RETADR 8
00000008             0000               TTY$L_IL_RETADR=8
                     0000
                     0000               $DEFEND TTYIL,,DEF
                     0000               .MACRO  $TTYILDEF A
                     0000               .ENDM   $TTYILDEF
                     0000               .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                     0000               .CROSS
                     0000               .RESTORE
                     0000
                     0000
                     0000               $TTYTADEF
                     0000               $DEFINI TTYTA,
```

```
0000                    .SAVE   LOCAL_BLOCK
0000                    .NOCROSS
0000                    .IIF    DIF <> <GLOBAL>,.ENABLE SUPPRESSION
0000                    .PSECT  $ABS$,ABS
0000                    $GBLINI
0000                    .IF     IDN <LOCAL> <GLOBAL>
0000                    .MACRO  $DEF    SYM,ALLOC,SIZ
0000                    .IIF    NB,SYM, SYM::
0000                    .IIF    NB,ALLOC,       ALLOC   SIZ
0000                    .ENDM   $DEF
0000                    .MACRO  $EQU    SYM,VAL
0000                    SYM==VAL
0000                    .ENDM   $EQU
0000                    .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000                    SIZ...=1
0000                    .IIF    NB,SIZ, SIZ...=SIZ
0000                    .IF     NB,SYM
0000                    MOD'SEP'V_'SYM==BIT...
0000                    .IIF    NB,SIZ, MOD'SEP'S_'SYM==SIZ
0000                    .IIF    NB,MSK, MOD'SEP'M_'SYM==<<<1@SIZ...>-1>@BIT...>
0000                    .ENDC
0000                    BIT...=BIT...+SIZ...
0000                    .ENDM   $VIELD1
0000                    .IFF
0000                    .IIF    DIF <LOCAL> <LOCAL>,.ERROR ;ARG MUST BE ''GLOBAL'',''LOCAL'',OR NULL
0000                    .MACRO  $DEF    SYM,ALLOC,SIZ
0000                    .IIF    NB,SYM, SYM:
0000                    .IIF    NB,ALLOC,       ALLOC   SIZ
0000                    .ENDM   $DEF
0000                    .MACRO  $EQU    SYM,VAL
0000                    SYM=VAL
0000                    .ENDM   $EQU
0000                    .MACRO  $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000                    SIZ...=1
0000                    .IIF    NB,SIZ, SIZ...=SIZ
0000                    .IF     NB,SYM
0000                    MOD'SEP'V_'SYM=BIT...
0000                    .IIF    NB,SIZ, MOD'SEP'S_'SYM=SIZ
0000                    .IIF    NB,MSK, MOD'SEP'M_'SYM=<<<1@SIZ...>-1>@BIT...>
0000                    .ENDC
0000                    BIT...=BIT...+SIZ...
0000                    .ENDM   $VIELD1
0000                    .ENDC
0000
00000000 0000           .=0
0000
0000            $EQU    TTY$K_TA_RCLLEN 256
00000100 0000           TTY$K_TA_RCLLEN=256
0000
0000            $EQU    TTY$S_TTYTADEF  284
0000011C 0000           TTY$S_TTYTADEF=284
0000
0000            $EQU    TTY$L_TA_PUT    0
00000000 0000           TTY$L_TA_PUT=0
0000
0000            $EQU    TTY$L_TA_GET    4
00000004 0000           TTY$L_TA_GET=4
```

```
                        0000
                        0000        $EQU    TTY$W_TA_SIZE   8
00000008                0000                TTY$W_TA_SIZE=8
                        0000
                        0000        $EQU    TTY$B_TA_TYPE   10
0000000A                0000                TTY$B_TA_TYPE=10
                        0000
                        0000        $EQU    TTY$B_TA_SPARE1 11
0000000B                0000                TTY$B_TA_SPARE1=11
                        0000
                        0000        $EQU    TTY$W_TA_INAHD  12
0000000C                0000                TTY$W_TA_INAHD=12
                        0000
                        0000        $EQU    TTY$W_TA_RCLOFF 14
0000000E                0000                TTY$W_TA_RCLOFF=14
                        0000
                        0000        $EQU    TTY$L_TA_END    16
00000010                0000                TTY$L_TA_END=16
                        0000
                        0000        $EQU    TTY$W_TA_RCLSIZ 20
00000014                0000                TTY$W_TA_RCLSIZ=20
                        0000
                        0000        $EQU    TTY$W_TA_SPARE2 22
00000016                0000                TTY$W_TA_SPARE2=22
                        0000
                        0000        $EQU    TTY$S_TA_RCL    256
00000100                0000                TTY$S_TA_RCL=256
                        0000
                        0000        $EQU    TTY$A_TA_RCL    24
00000018                0000                TTY$A_TA_RCL=24
                        0000
                        0000        $EQU    TTY$L_TA_DATA   280
00000118                0000                TTY$L_TA_DATA=280
                        0000
                        0000                $DEFEND TTYTA,,DEF
                        0000                .MACRO  $TTYTADEF A
                        0000                .ENDM   $TTYTADEF
                        0000                .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                        0000                .CROSS
                        0000                .RESTORE
                        0000
                        0000
                        0000                $DEFEND TTYDEFS,,DEF
                        0000                .MACRO  $TTYDEFSDEF A
                        0000                .ENDM   $TTYDEFSDEF
                        0000                .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                        0000                .CROSS
                        0000                .RESTORE
                        0000
                        0000
                        0000    213 .noshow me
                        0000    214         $TTYMODEM                       ; DEFINE MODEM DEFINITIONS
                        0000    215
                        0000    216
                        0000    217 ;
                        0000    218 ; LOCAL STORAGE
                        0000    219 ;
```

```
00000000   220          .PSECT  $$$105_PROLOGUE
0000       221
0000       222 ;
0000       223 ; Driver prologue table:
0000       224 ;
0000       225
0000       226 YC$DPT::                                        ; DRIVER START
0000       227          DPTAB   -                              ; DRIVER PROLOGUE TABLE
0000       228                  END=YC$END,-                   ; End and offset to INIT's vectors
0000       229                  UCBSIZE=UCB$C_TT_LENGTH,-      ; SIZE OF UCB
0000       230                  FLAGS=DPT$M_NOUNLOAD,-         ; NO UNLOAD ALLOWED
0000       231                  ADAPTER=UBA,-                  ; ADAPTER TYPE
0000       232                  DEFUNITS=8,-                   ; Number of units to create
0000       233                  NAME=YCDRIVER,-                ; NAME OF DRIVER
0000       234                  VECTOR=PORT_VECTOR             ; PORT DRIVER VECTOR TABLE
0038       235          DPT_STORE INIT
0038       236          DPT_STORE UCB,UCB$B_FIPL,B,8           ; FORK IPL
003C       237          DPT_STORE UCB,UCB$L_DEVCHAR,L,<-       ; CHARACTERISTICS
003C       238                       DEV$M_REC!-               ;
003C       239                       DEV$M_AVL!-               ;
003C       240                       DEV$M_IDV!-               ;
003C       241                       DEV$M_ODV!-               ;
003C       242                       DEV$M_TRM!-               ;
003C       243                       DEV$M_CCL>
0043       244          DPT_STORE UCB,UCB$L_DEVCHAR2,L,-       ; DEVICE CHARACTERISTICS
0043       245                       <DEV$M_NNM>               ; PREFIX WITH "NODE$"
004A       246          DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_TERM;
004E       247          DPT_STORE UCB,UCB$B_TT_DETYPE,B,TT$_UNKNOWN     ; TYPE
0052       248          DPT_STORE UCB,UCB$W_TT_DESIZE,aW,TTY$GW_DEFBUF  ; BUFFER SIZE
0059       249          DPT_STORE UCB,UCB$L_TT_DECHAR,aL,TTY$GL_DEFCHAR ; DEFAULT CHARACTERS
0060       250          DPT_STORE UCB,UCB$L_TT_DECHA1,aL,TTY$GL_DEFCHAR2; DEFAULT CHARACTERS
0067       251          DPT_STORE UCB,UCB$W_TT_DESPEE,aB,TTY$GB_DEFSPEED; DEFAULT SPEED
006E       252          DPT_STORE UCB,UCB$W_TT_DESPEE+1,aB,TTY$GB_RSPEED; DEFAULT RSPEED
0075       253          DPT_STORE UCB,UCB$B_TT_DEPARI,aB,TTY$GB_PARITY  ; DEFAULT PARITY
007C       254 ;**      DPT_STORE UCB,UCB$B_TT_PARITY,aB,TTY$GB_PARITY  ; DEFAULT PARITY
007C       255          DPT_STORE UCB,UCB$B_DEVTYPE,B,TT$_UNKNOWN      ; TYPE
0080       256          DPT_STORE UCB,UCB$W_DEVBUFSIZ,aW,TTY$GW_DEFBUF  ; BUFFER SIZE
0087       257          DPT_STORE UCB,UCB$L_DEVDEPEND,aL,TTY$GL_DEFCHAR ; DEFAULT CHARACTERS
008E       258          DPT_STORE UCB,UCB$L_DEVDEPND2,aL,TTY$GL_DEFCHAR2; DEFAULT CHARACTERS
0095       259          DPT_STORE UCB,UCB$W_TT_SPEED,aB,TTY$GB_DEFSPEED ; DEFAULT SPEED
009C       260          DPT_STORE UCB,UCB$W_TT_SPEED+1,aB,TTY$GB_RSPEED ; DEFAULT RSPEED
00A3       261          DPT_STORE UCB,UCB$B_DIPL,B,21                   ; DEVICE IPL
00A7       262          DPT_STORE UCB,UCB$L_TT_WFLINK,L,0               ; Zero write queue.
00AE       263          DPT_STORE UCB,UCB$L_TT_WBLINK,L,0               ; Zero write queue.
00B5       264          DPT_STORE UCB,UCB$L_TT_RTIMOU,L,0               ; Zero read timed out disp.
00BC       265          DPT_STORE ORB,ORB$B_FLAGS,B,-          ; Protection block flags
00BC       266                       <ORB$M_PROT_16>           ; SOGW protection word
00C0       267          DPT_STORE ORB,ORB$W_PROT,aW,TTY$GW_PROT ; Default allocation protection
00C7       268          DPT_STORE ORB,ORB$L_OWNER,aL,TTY$GL_OWNUIC ; Default owner UIC
00CE       269          DPT_STORE DDB,DDB$L_DDT,D,YC$DDT
00D3       270
00D3       271          DPT_STORE REINIT
00D3       272          DPT_STORE CRB,CRB$L_INTD+4,D,YC$INTINP  ; RECEIVER INTERRUPT
00D8       273          DPT_STORE CRB,CRB$L_INTD2+4,D,YC$INTOUT ; TRANSMITTER INTERRUPT
00DD       274          DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,D,YC$INITIAL  ; CONTROLLER INIT
00E2       275          DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,YC$INITLINE; UNIT INIT
00E7       276          DPT_STORE END
```

```
                     0000   277
                     0000   278          DDTAB   DEVNAM = YC,-    ; DUMMY DMF PORT DRIVER DISPATCH TABLE
                     0000   279                  START  = 0,-
                     0000   280                  FUNCTB = 0
                     0038   281
                00000038    282          .PSECT  $$$115_DRIVER,LONG
                     0038   283
                     0038   284 ;
                     0038   285 ; THE ASSOCIATED CLASS DRIVER USES THIS TABLE TO COMMAND THE PORT DRIVER.
                     0038   286 ; THE ADDRESS OF THIS TABLE IS CONTAINED IN THE TERMINAL UCB EXTENSION AREA.
                     0038   287 ; THE OFFSET DEFINITONS ARE DEFINED BY TTYDEFS.
                     0038   288 ;
                     0038   289 YC$L_SIL_ERROR:
           00000000  0038   290          .LONG   0
                     003C   291 YC$L_ERROR:
           00000000  003C   292          .LONG   0
                     0040   293 YC$L_DMAXMT_ERROR:
           00000000  0040   294          .LONG   0
                     0044   295 PORT_VECTOR:
                     0044   296
                     0044   297 ;
                     0044   298 ; DMF SPECIFIC DISPATCH TABLE
                     0044   299 ;
                     0044   300          $VECINI YC,YC$NULL
                     007C   301          $VEC    STARTIO,YC$STARTIO              ; START NEW OUTPUT
                     0048   302          $VEC    SET_LINE,YC$SET_LINE            ; SET NEW SPEED/PARITY
                     0050   303          $VEC    DS_SET,YC$DS_SET                ; SET OUTPUT MODEM SIGNALS
                     0054   304          $VEC    XON,YC$XON                      ; SEND XON SEQUENCE
                     0058   305          $VEC    XOFF,YC$XOFF                    ; SEND XOFF SEQUENCE
                     005C   306          $VEC    STOP,YC$STOP                    ; STOP OUTPUT
                     0060   307          $VEC    ABORT,YC$ABORT                  ; ABORT OUTPUT IN PROGRESS
                     0068   308          $VEC    RESUME,YC$RESUME                ; RESUME STOPPED OUTPUT
                     006C   309          $VEC    MAINT,YC$MAINT                  ; INVOKE MAINTENANCE FUNCTION
                     0078   310          $VEC    FORKRET,YC$FORK                 ; PORT FORK CALLBACK
                     007C   311          $VECEND
                     0080   312
                     0080   313
                     0080   314 ;
                     0080   315 YC$NULL:                                         ; NULL PORT ROUTINE
                  05 0080   316          RSB
                     0081   317
```

```
0081    319          .SBTTL  REGISTER DEFINITIONS
0081    320
0081    321 ;
0081    322 ; CSR BIT DEFINITIONS ( CSR ) ( READ/WRITE )
0081    323 ; (NOTE: THIS REGISTER MUST ONLY BE READ IN RESPONSE TO A TRANSMIT INTERRUPT.
0081    324 ;       TO LOAD AN INDIRECT REGISTER, ONLY WRITE OPERATIONS MAY BE USED)
0081    325
0081    326          $VIELD  DMFCSR,0,<-
0081    327                  <IADDR,5,M>,-    ; INDERECT REGISTER ADDRESS
0081    328                  <CLEAR,1,M>,-    ; MASTER RESET
0081    329                  <RCVINT,1,M>,-   ; RECEIVER INTERRUPT ENABLE
0081    330                  <,1,>,-          ; DON'T USE THIS BIT
0081    331                  <LINE,3,M>,-     ; LINE NUMBAE (0 - 7)
0081    332                  <,1,>,-
0081    333                  <DMAERR,1,M>,-   ; DMA TRANSMIT ERROR
0081    334                  <,1,>,-
0081    335                  <SNDINT,1,M>,-   ; TRANSMIT INTERRUPT ENABLE
0081    336                  <SNDRDY,1,M>-    ; TRANSMITTER READY
0081    337                  >
0081    338 ;
0081    339 ; RECEIVER BUFFER ( CSR+4 ) ( READ ONLY )
0081    340 ;
0081    341          $VIELD  DMFRCV,0,<-
0081    342                  <BUF,8,M>,-      ; RECEIVER DATA
0081    343                  <LINE,3,M>,-     ; LINE NUMBER (0 - 7)
0081    344                  <DS_CHG,1,M>,-   ; DATA SET CHANGE INTERRUPT
0081    345                  <PARERR,1,M>,-   ; PARITY ERROR
0081    346                  <FRAMER,1,M>,-   ; FRAME ERROR
0081    347                  <OVERRUN,1,M>,-  ; OVERRUN ERROR
0081    348                  <VALID,1,M>-     ; DATA VALID
0081    349                  >
0081    350 ;
0081    351 ; LINE PARAMETER REGISTER ( CSR+2 ) ( WRITE ONLY )
0081    352 ;
0081    353
0081    354          $VIELD  DMFLPR,0,<-
0081    355                  <LINE,3,M>,-     ; LINE NUMBER (0-7)
0081    356                  <SIZE,2,M>,-     ; CHARACTER SIZE
0081    357                  <PARITY,1,M>,-   ; PARITY ENABLE
0081    358                  <ODD,1,M>,-      ; ODD PARITY
0081    359                  <STOP,1,M>,-     ; NUMBER STOP BITS
0081    360                  <RSPEED,4,M>,-   ; RECEIVER LINE SPEED
0081    361                  <TSPEED,4,M>,-   ; TRANSMITTER LINE SPEED (BOTH RVC/TX FOR LINES 2-7)
0081    362                  >
0081    363 ;
0081    364 ; LINE CONTROL INDIRECT REGISTER (INDIRECT REGISTFR #2)
0081    365 ;
0081    366
0081    367          $VIELD  DMFLCT,0,<-
0081    368                  <XMIT,1,M>,-     ; TRANSMIT ENABLE
0081    369                  <AUTOX,1,M>,-    ; AUTO XON/OFF
0081    370                  <RCV,1,M>,-      ; RECEIVER ENABLE
0081    371                  <BREAK,1,M>,-    ; SEND BREAK
0081    372                  <FLUSH,1,M>,-    ; FLUSH OUTPUT SILO/ABORT TRANSMIT
0081    373                  <DS_ENABLE,1,M>,- ; ENABLE MODEM TRANSITION INTERRUPTS
0081    374                  <MAINT,2,M>,-    ;
0081    375                  <,7,>,-
```

```
0081   376                  <PREMPT,1,M>,-  ; PREMPT CURRENT OUTPUT
0081   377                  >
0081   378
0081   379 ;
0081   380 ; MACRO USED TO ACCESS INDIRECT REGISTERS
0081   381 ;
0081   382
0081   383          .MACRO  GETIND  OFFSET,RX
0081   384          MOVL    UCB$L_CRB(R5),R0                        ; GET CRB ADDRESS
0081   385          MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0            ; GET CSR ADDRESS
0081   386          BISW3   #<<^X4040>+<OFFSET*8>>,UCB$W_UNIT(R5),(R0); SELECT INDIRECT FIELD
0081   387          .IF     NB      RX
0081   388          MOVW    6(R0),RX                                ; GET CONTENTS OF INDIRECT F
0081   389          .ENDC
0081   390          .ENDM   GETIND
0081   391
0081   392 ;        OUTPUT DELAY MACRO
0081   393 ;        TO RELEASE THE UNIBUS
0081   394
0081   395          .MACRO  DELAY
0081   396          .REPEAT 3
0081   397          NOP
0081   398          .ENDR
0081   399          .ENDM
0081   400
```

```
                     0081   402          .SBTTL  CONTROLLER INITIALIZATION
                     0081   403
                     0081   404
                     0081   405
                     0081   406 ;++
                     0081   407 ; YC$INITIAL - INITIALIZE INTERFACE
                     0081   408 ;
                     0081   409 ; FUNCTIONAL DESCRIPTION:
                     0081   410 ;
                     0081   411 ; THIS ROUTINE IS ENTERED AT SYSTEM STARTUP AND POWER RECOVERY.
                     0081   412 ;
                     0081   413 ; INPUTS:
                     0081   414 ;
                     0081   415 ;       R4 = ADDRESS OF THE UNIT CSR
                     0081   416 ;       R5 = IDB OF UNIT
                     0081   417 ;       R8 = ADDRESS OF THE UNIT CRB
                     0081   418 ;
                     0081   419 ; OUTPUTS:
                     0081   420 ;
                     0081   421 ;       R2 is destroyed.
                     0081   422 ;
                     0081   423 ; IMPLICIT INPUTS:
                     0081   424 ;
                     0081   425 ;       IPL = IPL$_POWER
                     0081   426 ;
                     0081   427 ;--
                     0081   428 YC$INITIAL::                                 ; INITIALIZE DMF UNIT
                     0081   429 ;
                     0081   430 ; SET UP CONTROLLER
                     0081   431 ;
                     0081   432          CLASS_CTRL_INIT YC$DPT,PORT_VECTOR; RELOCATE THE NECESSARY TABLES
                     00AD   433 25$:
                0B A8  95    00AD   434          TSTB    CRB$B_TT_TYPE(R8)                   ; CONTROLLER TYPE INITIALIZED?
                   15  12    00B0   435          BNEQ    30$                                 ; IF NEQ YES
:WHM0004        0B A8  45 8F 90    00B2    .1     MOVB    #DT$_DMZ32,CRB$B_TT_TYPE(R8)       ; ASSUME DMZ32
:WHM0004          10 A5  10  91    00B7    .2     CMPB    #16,IDB$B_COMBO_VECTOR_OFFSET(R5); DMF32 HAS A VECTOR OFFSET OF 16
:WHM0004                 0A  12    00BB    .3     BNEQ    30$                                 ; IF NEQ NO, MUST BE A DMZ32
:WHM0004          0F A5  0C  91    00BD    .4     CMPB    #12,IDB$B_COMBO_CSR_OFFSET(R5)      ; DMF32 HAS A CSR OFFSET OF 12
:WHM0004                 04  12    00C1    .5     BNEQ    30$                                 ; IF NEQ NO, MUST BE A DMZ32
:WHM0004          0B A8  0A  90    00C3    .6     MOVB    #DT$_DMF32,CRB$B_TT_TYPE(R8)       ; SET CONTROLLER TYPE DMF32
-4
                     00C7   440 30$:
                50   0F A5  98    00C7   441          CVTBL   IDB$B_COMBO_CSR_OFFSET(R5),R0       ; GET THE OFFSET TO THE MAIN CSR
                   10 A5  83    00CB   442          SUBB3   IDB$B_COMBO_VECTOR_OFFSET(R5),-     ; LOAD THE SOFT VECTOR
              6440   0B A5         00CE   443                  IDB$B_VECTOR(R5),(R4)[R0]           ;
                   64    20  B0    00D2   444          MOVW    #DMFCSR$M_CLEAR,(R4)                ; CONTROLLER RESET
                                    00D5   445
                                    00D5   446 ;
                                    00D5   447 ;       WAIT TILL CONTROLLER INITIALIZATION IS COMPLETE
                                    00D5   448 ;
                                    00D5   449          TIMEWAIT        #100,#DMFCSR$M_CLEAR,(R4),W,.FALSE.
                                    00FC   450
              64   4040 8F    B0    00FC   451          MOVW    #<<DMFCSR$M_RCVINT>!-       ; ENABLE RECEIVER INTERRUPTS
                                    0101   452                  <DMFCSR$M_SNDINT>>,(R4)     ; ENABLE TRANSMITTER INTERRUPTS
                      09 50  E9     0101   453          BLBC    R0,YC$CTRL_ERROR
                                    0104   454
                                    0104   455
                                    0104   456 100$:
```

```
00000000'GF   90  0104    457          MOVB    G^TTY$GB_SILOTIME,4(R4) ; INIT INPUT SILO TIMEOUT VALUE
      04 A4       010A
                  010C    458
                  010C    459
                  010C    460
              05  010C    461          RSB
                  010D    462
                  010D    463 YC$CTRL_ERROR:
              05  010D    464          RSB
                  010E    465
```

```
                                010E    467          .SBTTL  UNIT INITIALIZATION
                                010E    468
                                010E    469 ; YC$INITLINE - UNIT INITIALIZATION
                                010E    470 ;
                                010E    471 ; FUNCTIONAL DESCRIPTION:
                                010E    472 ;
                                010E    473 ; THIS ROUTINE PERFORMS A SIMPLE UNIT INITIALIZATION.
                                010E    474 ;
                                010E    475 ; INPUTS:
                                010E    476 ;
                                010E    477 ;       R5 = UCB ADDRESS
                                010E    478 ;
                                010E    479 ; OUTPUTS:
                                010E    480 ;
                                010E    481 ;       R4,R5 ARE PRESERVED.
                                010E    482 ;--
                                010E    483
                                010E    484 YC$INITLINE::
            50    FF32 CF     DE    010E    485          MOVAL   YC$VEC,R0               ; GET THE DISPATCH TABLE ADDRESS
                                0113    486          CLASS_UNIT_INIT
              64 A5    10     A8    015C    487          BISW    #UCB$M_ONLINE,UCB$W_STS(R5); SET ONLINE
                                0160    488
              01    54 A5     78    0160    489 10$:     ASHL    UCB$W_UNIT(R5),#1,R3    ; BUILD UNIT'S BIT MASK
                       53
                                0164
            0106 C5    53     B0    0165    490          MOVW    R3,UCB$W_TT_UNITBIT(R5) ; SAVE IT
                                016A    491
                       24     A8    016A    492          BISW    #TTY$M_PC_DMAAVL!TTY$M_PC_XOFAVL,-; SHOW DMA FEATURE AVAILABLE FOR U
                  0122 C5           016C    493                  UCB$W_TT_PRTCTL(R5)     ; IN PORT LEVEL
;MIR0001                 00     E1    016F     .1          BBC     #0,G^TTY$GL_DEFPORT,15$ ; DO WE REALY WANT AUTOXON?
                05 00000000'GF      0171
;MIR0001           0122 C5    20     AA    0177     .2          BICW    #TTY$M_PC_XOFAVL,UCB$W_TT_PRTCTL(R5); NO THEN TURN IT OFF
;MIR0001                             017C     .3
;MIR0001           51    0114 C5     D0    017C     .4 15$:     MOVL    UCB$L_TT_CLASS(R5),R1   ; ADDRESS CLASS VECTOR TABLE
-2                      08 B1     16    0181    496          JSB     @CLASS_SETUP_UCB(R1)    ; INIT UCB FIELDS
                                0184    497
                                0184    498 ;SET MODE CODE NEEDS TO TOGGLE THESE BITS
                                0184    499
                00000778'EF     16    0184    500          JSB     YC$SET_LINE             ; INIT SPEED/PARITY
                                018A    501
                                018A    502
                                018A    503 ;
                                018A    504 ; ENABLE LINE RECEIVER , TRANSMITTER AND MODEM INTERRUPTS
                                018A    505 ;
                                018A    506
              4048 8F    A9    018A    507          BISW3   #^X4048,UCB$W_UNIT(R5),(R4)     ; SELECT LCT
          64    54 A5           018E
            06 A4    25     B0    0191    508          MOVW    #<<DMFLCT$M_XMIT>!-             ; ENABLE TRANSMIT
                                0195    509                  <DMFLCT$M_RCV>!-                ; ENABLE RECEIVE
                                0195    510                  <DMFLCT$M_DS_ENABLE>>,6(R4)     ; AND MODEM INTERRUPTS
                                0195    511 ;
                                0195    512 ; INIT RECEIVER MODEM STATUS FOR DMF
                                0195    513 ;
                                0195    514
              4040 8F    A9    0195    515          BISW3   #^X4040,UCB$W_UNIT(R5),(R4)     ; SELECT RECV MODEM INDIRECT REG
          64    54 A5           0199
              07 A4    90     019C    516          MOVB    7(R4),UCB$B_TT_DS_RCV(R5)       ; UPDATE RECEIVER MODEM STATUS
            0124 C5           019F
```

```
                                                M 16
                        01A2    517
                        01A2    518 25$:
      51    00    9A    01A2    519             MOVZBL  #MODEM$C_INIT,R1          ; ASSUME INIT MODEM PROTOCOL
50    0114  C5    D0    01A5    520             MOVL    UCB$L_TT_CLASS(R5),R0     ; ADDRESS CLASS VECTOR TABLE
        0C  B0    16    01AA    521             JSB     @CLASS_D$_TRAN(R0)        ; INVOKE TO INIT MODEM PROTOCOL
                        01AD    522 30$:
08 64 A5    05    E1    01AD    523             BBC     #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
50    0114  C5    D0    01B2    524             MOVL    UCB$L_TT_CLASS(R5),R0     ; GET THE CLASS VECTOR TABLE ADDRESS
        20  B0    17    01B7    525             JMP     @CLASS_POWERFAIL(R0)      ; AND GOTO THE POWERFAIL CODE
                        01BA    526
                    05  01BA    527 40$:        RSB
                        01BB    528 ;
                        01BB    529 ; ERROR DETECTED DURING INITIALIZATION
                        01BB    530 ;
                        01BB    531
                        01BB    532 YC$UNIT_ERROR:
   64 A5    10    AA    01BB    533             BICW    #UCB$M_ONLINE,UCB$W_STS(R5)     ; UNIT NOT ON LINE
                    05  01BF    534             RSB
                        01C0    535
```

```
                     01C0   537          .SBTTL  MAINTENANCE ROUTINES
                     01C0   538 ;++
                     01C0   539 ; YC$MAINT - MAINTENANCE FUNCTIONS
                     01C0   540 ;
                     01C0   541 ; FUNCTIONAL DESCRIPTION:
                     01C0   542 ; THIS ROUTINE PERFORMS MAINTENANCE FUNCTIONS FOR THE DMF
                     01C0   543 ; (LOOPBACK IS ONLY ALLOWED ON LINES 0 AND 1)
                     01C0   544 ;
                     01C0   545 ; INPUTS:
                     01C0   546 ;
                     01C0   547 ;       R5 = UBC ADDRESS
                     01C0   548 ;       UCB$B_TT_MAINT = FUNCTION TO BE PERFORMED
                     01C0   549 ;
                     01C0   550 ; OUTPUTS:
                     01C0   551 ;       R0-R4 SCRATCH
                     01C0   552 ;--
                     01C0   553
                     01C0   554 YC$MAINT:
        01   93      01C0   555          BITB    #IO$M_LOOPA-7,-                 ; LOOPBACK FUNCTION
    012A C5          01C2   556                  UCB$B_TT_MAINT(R5)
          0E 13      01C5   557          BEQL    5$                              ; NO
          12 B0      01C7   558          MOVW    #TT$M_DS_DTR!TT$M_DS_RTS,-      ; SET REQUIRED MODEM LINES
          52         01C9   559                  R2
00000295'EF   16     01CA   560          JSB     YC$DS_SET                       ; SET THEM TO ALLOW LOOPBACK
       52   02      3C 01D0   561          MOVZWL  #^X02,R2                        ; SPECIFY LOOPBACK CODE
          15 11      01D3   562          BRB     10$
                     01D5   563 5$:
        02   93      01D5   564          BITB    #IO$M_UNLOOPA-7,-               ; RESET LOOPBACK FUNCTION
    012A C5          01D7   565                  UCB$B_TT_MAINT(R5)
          42 13      01DA   566          BEQL    15$                             ; NO
    1300 8F   B0     01DC   567          MOVW    #256*<TT$M_DS_DTR!TT$M_DS_RTS!^X01>,-; RESET REQUIRED MODEM LINES
          52         01E0   568                  R2
00000295'EF   16     01E1   569          JSB     YC$DS_SET                       ; SET THEM TO ALLOW LOOPBACK
       52   00      3C 01E7   570          MOVZWL  #^X00,R2                        ; SPECIFY UNLOOP CODE
                     01EA   571 10$:
    51   24 A5      D0 01EA   572          MOVL    UCB$L_CRB(R5),R1                ; GET THE CRB ADDRESS
0B  A1   45 8F      91 01EE   573          CMPB    #DT$_DMZ32,CRB$B_TT_TYPE(R1)    ; IS THIS A DMZ
          06 13      01F3   574          BEQL    12$                             ; YES THEN MODEM CONTROL ON ALL UNIT
    54 A5    02      B1 01F5   575          CMPW    #2,UCB$W_UNIT(R5)               ; ALLOWED ON THIS UNIT?
          20 15      01F9   576          BLEQ    50$                             ; NO
                     01FB   577 12$:     GETIND  1,R1                            ; GET INDIRECT FIELD
02   06    52 F0     020E   578          INSV    R2,#DMFLCT$V_MAINT,#2,R1        ; SET MAINT FIELD
          51         0212
  06 A0    51 B0     0213   579          MOVW    R1,6(R0)                        ; UPDATE
       50  01 9A     0217   580          MOVZBL  #1,R0                           ; INDICATE SUCCESS
          05 021A    581          RSB
                     021B   582 50$:
          50 D4      021B   583          CLRL    R0
          05 021D    584          RSB
                     021E   585
                     021E   586 15$:
     40 8F   93      021E   587          BITB    #IO$M_AUTXOF_ENA-7,-
    012A C5          0221   588                  UCB$B_TT_MAINT(R5)              ;AUTOXON ENABLED
          05 13      0224   589          BEQL    17$                             ; NO THEN MAYBE DISABLE
          20 A8      0226   590          BISW    #TTY$M_PC_XOFAVL,-
    0122 C5          0228   591                  UCB$W_TT_PRTCTL(R5)             ; SET THE BIT AVAILABLE
                     022B   592
```

```
                     80 8F   93   022B   593 17$:    BITB    #IO$M_AUTXOF_DIS@-7,-
                 012A C5          022E   594                 UCB$B_TT_MAINT(R5)          ;AUTOXON disabled
                        05   13   0231   595         BEQL    19$                         ; no then don't disable it
                        20   AA   0233   596         BICW    #TTY$M_PC_XOFAVL,-
                 0122 C5          0235   597                 UCB$W_TT_PRTCTL(R5)
                                  0238   598
                                  0238   599 19$:
                                  0238   600
                        20   93   0238   601         BITB    #IO$M_LOOP_EXT@-7,-         ; LOOPBACK FUNCTION
                 012A C5          023A   602                 UCB$B_TT_MAINT(R5)
                        0E   13   023D   603         BEQL    20$                         ; NO
                        13   B0   023F   604         MOVW    #<TT$M_DS_DTR!TT$M_DS_RTS!^X01>,-; SET REQUIRED MODEM LINES
                        52        0241   605                 R2
           00000295'    EF   16   0242   606         JSB     YC$DS_SET                   ; SET THEM TO ALLOW LOOPBACK
                52      00   3C   0248   607         MOVZWL  #^X00,R2                    ; SPECIFY  NO LOOPBACK CODE
                        9D   11   024B   608         BRB     10$
                                  024D   609 20$:                                       ; CHECK OTHER FUNCTIONS
                        04   93   024D   610         BITB    #IO$M_LINE_OFF@-7,-         ; LINE OFF
                 012A C5          024F   611                 UCB$B_TT_MAINT(R5)
                        1C   13   0252   612         BEQL    30$                         ; NO
                                  0254   613         GETIND  1,R1                        ; GET FIELD
                51      05   AA   0267   614         BICW    #<<DMFLCT$M_XMIT>!-         ; DISABLE TRANSMIT AND RECEIVE
                                  026A   615                 <DMFLCT$M_RCV>>,R1
             06 A0      51   B0   026A   616         MOVW    R1,6(R0)                    ; UPDATE
                        21   11   026E   617         BRB     40$
                                  0270   618 30$:
                        10   93   0270   619         BITB    #IO$M_LINE_ON@-7,-          ; LINE ON
                 012A C5          0272   620                 UCB$B_TT_MAINT(R5)
                        A4   13   0275   621         BEQL    50$                         ; NO
                                  0277   622         GETIND  1,R1                        ; GET FIELD
                51      05   A8   028A   623         BISW    #<<DMFLCT$M_XMIT>!-         ; ENABLE TRANSMIT AND RECEIVE
                                  028D   624                 <DMFLCT$M_RCV>>,R1
             06 A0      51   B0   028D   625         MOVW    R1,6(R0)                    ; UPDATE
                                  0291   626
                50      01   9A   0291   627 40$:    MOVZBL  #1,R0
                                  0294   628         RSB
                                  0295   629
                                  0295   630
```

```
                     0295   632          .SBTTL  OUTPUT MODEM CONTROL
                     0295   633 ;++
                     0295   634 ; YC$DS_SET - SET OUTPUT MODEM SIGNALS
                     0295   635 ;
                     0295   636 ; FUNCTIONAL DESCRIPTION:
                     0295   637 ;
                     0295   638 ; THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT
                     0295   639 ;
                     0295   640 ; INPUTS:
                     0295   641 ;
                     0295   642 ;       R2 = LOW BYTE - SIGNALS TO ACTIVATE
                     0295   643 ;            HIGH BYTE- SIGNALS TO DEACTIVATE
                     0295   644 ;
                     0295   645 ;       R5 = UCB ADDRESS
                     0295   646 ;
                     0295   647 ; OUTPUTS:
                     0295   648 ;
                     0295   649 ;       R0-R3 ARE USED.
                     0295   650 ;--
                     0295   651
                     0295   652 YC$DS_SET:
0125 C5    52   88  0295   653          BISB    R2,UCB$B_TT_DS_TX(R5)   ; SET NEW OUTPUT SIGNALS
  52    F8 8F   78  029A   654          ASHL    #-8,R2,R2               ; ACCESS SIGNALS TO RESET
               52
                     029E
0125 C5    52   8A  029F   655          BICB    R2,UCB$B_TT_DS_TX(R5)   ; RESET THEM
                     02A4   656          GETIND  1,-(SP)
     E0 8F    8B  02B7   657          BICB3   #^XOE0,UCB$B_TT_DS_TX(R5),- ; OR IN OUTPUT MODEM SIGNALS
   0125 C5         02BA
     01 AE         02BD   658                  1(SP)
 06 A0    8E   B0  02BF   659          MOVW    (SP)+,6(R0)             ; OUTPUT UPDATED VALUE
               05  02C3   660          RSB
                   02C4   661
                   02C4   662
```

```
                     02C4    664          .SBTTL  RECEIVER INTERRUPT SERIVCE
                     02C4    665 ;++
                     02C4    666 ; YC$INTINP - DMF RECEI ER READY INTERRUPTS
                     02C4    667 ;
                     02C4    668 ; FUNCTIONAL DESCRIPTION:
                     02C4    669 ;
                     02C4    670 ; THIS ROUTINE IS ENTERED WHEN A CHARACTER IS AVAILABLE IN THE UNIT'S
                     02C4    671 ; SILO. THE CHARACTER IS EXTRACTED AND IS PASSED TO THE ASSOCIATED
                     02C4    672 ; CLASS DRIVER. IF THE CLASS DRIVER RETURNS CHARACTERS(S) THEN NEW
                     02C4    673 ; OUTPUT IT INITIATED (NORMALLY ECHO).
                     02C4    674 ;
                     02C4    675 ; INPUTS:
                     02C4    676 ;
                     02C4    677 ;       00(SP) = ADDRESS OF IDB
                     02C4    678 ;
                     02C4    679 ; IMPLICIT INPUTS:
                     02C4    680 ;
                     02C4    681 ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON STACK.
                     02C4    682 ;
                     02C4    683 ; OUTPUTS:
                     02C4    684 ;
                     02C4    685 ;       THE INTERRUPT IS DISMISSED WHEN THE SILO IS EMPTY.
                     02C4    686 ;
                     02C4    687 ;--
                     02C4    688 YC$INTINP::                             ; DMF INPUT INTERRUPTS
                     02C4    689 ;
                     02C4    690 ; GET THE CSR ADDRESS
                     02C4    691 ;
54   9E   D0         02C4    692          MOVL    @(SP)+,R4              ; GET THE IDB ADDRESS
     54   DD         02C7    693          PUSHL   R4                     ; SAVE IDB ADDRESS
50   64   D0         02C9    694          MOVL    (R4),R0                ; GET THE CSR ADDRESS
                     02CC    695 ;
                     02CC    696 ; GET THE CHARACTER FROM THE INTERFACE
                     02CC    697 ;
  53   04 A0   B0    02CC    698 25$:     MOVW    4(R0),R3               ; GET THE CHARACTER,ERRORS AND LINE NUMBER
          60   18    02D0    699          BGEQ    100$                   ;SILO EMPTY
53   7800 8F   B3    02D2    700          BITW    #<DMFRCV$M_PARERR>!-
                     02D7    701                  <DMFRCV$M_OVERRUN>!-
                     02D7    702                  <DMFRCV$M_FRAMER>!-
                     02D7    703                  <DMFRCV$M_DS_CHG>,R3   ;ERRORS OR MODEM TRANSITION ?
                     02D7    704
          36   12    02D7    705          BNEQ    50$                    ;YES,PROCESS THEM
  53   F8 8F   78    02D9    706 27$:     ASHL    #-8,R3,R2              ; GET THE LINE NUMBER
          52         02DD
FFFFFFF8 8F   CA    02DE    707          BICL    #^C<7>,R2              .
          52         02E4
     53   53   9A    02E5    708          MOVZBL  R3,R3                  ; CLEAR THE HIGH BYTES OF CHARACTER
55   18 A442   D0    02E8    709          MOVL    IDB$L_UCBLST(R4)[R2],R5 ; GET THE UCB FOR THAT LINE
          DD   13    02ED    710          BEQL    25$                    ; IF EQL THEN NOT THERE
     0110 D5   16    02EF    711          JSB     @UCB$L_TT_PUTNXT(R5)   ; BUFFER THE CHARACTER
          10   15    02F3    712          BLEQ    40$                    ; NONE OR STRING OUTPUT
     4040 8F   A9    02F5    713          BISW3   #^X4040,UCB$W_UNIT(R5),(R0)     ; SELECT OUTPUT SILO INDIRECT REGIST
  60   54 A5         02F9
  06 A0   53   90    02FC    714          MOVB    R3,6(R0)               ; OUTPUT CHARACTER
     54   6E   D0    0300    715 30$:     MOVL    (SP),R4                ; GET IDB ADDRESS
          C7   11    0303    716          BRB     25$                    ; CONTINUE
                     0305    717 40$:
```

```
                F9   13  0305   718          BEQL    30$                             ; NO CHARACTER
0000036B'       EF   16  0307   719          JSB     BURST_OUTPUT                    ; START BURST
                F1   11  030D   720          BRB     30$
                     030F   721 ;
                     030F   722 50$:
                     030F   723 ;
                     030F   724 ; PROCESS PARITY, FRAME OVERRUN ERROR OR MODEM TRANSITION
                     030F   725 ;
  53     F8 8F       78  030F   726          ASHL    #-8,R3,R2                       ; GET LINE NUMBER
             52           0312
FFFFFFF8     8F       CA  0314   727          BICL    #^C<7>,R2                       ;
             52           031A
  55   18 A442       D0  031B   728          MOVL    IDB$L_UCBLST(R4)[R2],R5 ; GET UCB ADDRESS
               0E      13  0320   729          BEQL    70$                             ; IF EQL THEN NO UCB
    19 53     0B       EC  0322   730          BBS     #DMFRCV$V_DS_CHG,R3,200$; MODEM TRANSITION
  52   0114   C5       D0  0326   731          MOVL    UCB$L_TT_CLASS(R5),R2   ; GET CLASS DISPATCH
          14   B2       16  032B   732          JSB     @CLASS_READERROR(R2)            ; SIGNAL ERROR
               A9       12  032E   733          BNEQ    27$                             ; CHRRACTER TO ECHO
                     0330   734 70$:
               CE       11  0330   735          BRB     30$
                     0332   736
       5E      04       C0  0332   737 100$:    ADDL    #4,SP                           ; REMOVE IDB ADDRESS
       50      8E       7D  0335   738          MOVQ    (SP)+,R0                        ; RESTORE REGISTERS
       52      8E       7D  0338   739          MOVQ    (SP)+,R2                        ;
       54      8E       7D  033B   740          MOVQ    (SP)+,R4                        ;
                        02  033E   741          RET
                     033F   742 200$:
  52   4040   8F       A9  033F   743          BISW3   #^X4040,R2,(R0)                 ; SELECT MODEM INDIRECT REGISTER
               60           0344
    52     07   A0       9A  0345   744          MOVZBL  7(R0),R2                        ; GET CURRENT RECEIVE MODEM SIGNALS
0124   C5       52       90  0349   745          MOVB    R2,UCB$B_TT_DS_RCV(R5)  ; UPDATE CURRENT INPUT MODEM SIGNALS
         51       03       9A  034E   746          MOVZBL  #MODEM$C_DATASET,R1     ; TRANSITION TYPE IS DATASET
                  50       DD  0351   747          PUSHL   R0                              ; SAVE CSR ADDRESS
  54   0114   C5       D0  0353   748          MOVL    UCB$L_TT_CLASS(R5),R4   ; GET CLASS DISPATCH
            0C   B4       16  0358   749          JSB     @CLASS_DS_TRAN(R4)              ; INVOKE TRANSITION ROUTINE
                  50   8ED0  035B   750          POPL    R0                              ; RESTORE CSR ADDRESS
               FF9F       31  035E   751          BRW     30$
                     0361   752
                     0361   753
```

```
                     0361   755          .SBTTL   START I/O ROUTINE
                     0361   756 ;++
                     0361   757 ; YC$STARTIO - START I/O OERATION ON DMF
                     0361   758 ;
                     0361   759 ; FUNCTIONAL DESCRIPTION:
                     0361   760 ;
                     0361   761 ; THIS ROUTINE IS ENTERED FROM THE DEVICE INDEPENDENT TERMINAL STARTIO
                     0361   762 ; ROUTINE TO ENABLE OUTPUT INTERRUPTS ON AN IDLE DMF UNIT.
                     0361   763 ;
                     0361   764 ; INPUTS:
                     0361   765 ;
                     0361   766 ;       R3 =    CHARACTER       AND     CC = PLUS
                     0361   767 ;               ADDRESS         AND     CC = NEGATIVE
                     0361   768 ;
                     0361   769 ;       R5 = UCB ADDRESS
                     0361   770 ;
                     0361   771 ; OUTPUTS:
                     0361   772 ;
                     0361   773 ;       R5 = UCB ADDRESS
                     0361   774 ;--
                     0361   775          .ENABLE LSB
                     0361   776 YC$STARTIO::                                ; START I/O ON UNIT
           66    18  0361   777          BGEQ    90$                         ; SINGLE CHARACTER
     50    24 A5 D0  0363   778          MOVL    UCB$L_CRB(R5),R0            ; GET CRB OF UNIT
     50    2C B0 D0  0367   779          MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0; GET CSR
                     036B   780
                     036B   781 BURST_OUTPUT:
  52   0120 C5 3C    036B   782          MOVZWL  UCB$W_TT_OUTLEN(R5),R2      ; GET LENGTH
           01 E1     0370   783          BBC     #TTY$V_PC_DMAENA,-          ; USE SILO IF DMA NOT ENABLED ON THIS LINE
    09 0122 C5       0372   784                  UCB$W_TT_PRTCTL(R5),SILO_OUTPUT
           52 B1     0376   785          CMPW    R2,G^TTY$GW_DMASIZE         ; LARGE ENOUGH FOR DMA
 00000000'GF       0378
           60    18  037D   786          BGEQ    DMA_START                   ; YES SO DO DMA
                     037F   787
                     037F   788 SILO_OUTPUT:
     4040 8F A9      037F   789          BISW3   #^X4040,UCB$W_UNIT(R5),(R0); SELECT TRANSMIT SILO
   60   54 A5       0383
   51   06 A0 9A    0386   790          MOVZBL  6(R0),R1                    ; GET SILO DEPTH
51   20   51 C3     038A   791          SUBL3   R1,#32,R1                           ; CONVERT TO NUMBER SLOTS AVAILABLE
                     038E   792
      51   52 B1     038E   793          CMPW    R2,R1                       ; BURST LARGER THAN SILO?
           03 1B     0391   794          BLEQU   50$                         ; NO
      52   51 9A     0393   795          MOVZBL  R1,R2                       ; SLOTS AVAILABLE IS MAXIMUM
                     0396   796 50$:
53   011C C5 D0      0396   797          MOVL    UCB$L_TT_OUTADR(R5),R3      ; GET ADDRESS
011C C5   52 C0      039B   798          ADDL    R2,UCB$L_TT_OUTADR(R5)      ; UPDATE POINTER
0120 C5   52 A2      03A0   799          SUBW    R2,UCB$W_TT_OUTLEN(R5)      ; AND COUNT
           07 13     03A5   800          BEQL    60$                         ; ALL DONE, NO NEED FOR BURST
     0800 8F A8      03A7   801          BISW    #TTY$M_TANK_BURST,-         ; SIGNAL BURST ACTIVE
     0108 C5         03AB   802                  UCB$W_TT_HOLD(R5)
                     03AE   803 60$:
        08 52 E9     03AE   804          BLBC    R2,70$                      ; EVEN TRANSFER
  06 A0    83 90     03B1   805          MOVB    (R3)+,6(R0)                 ; OUTPUT ODD BYTE
           52 D7     03B5   806          DECL    R2                          ; UPDATE COUNT
           0F 13     03B7   807          BEQL    80$                         ; DONE
                     03B9   808 70$:
  52   FF 8F 78      03B9   809          ASHL    #-1,R2,R2                   ; CONVERT TO WORD COUNT
```

```
                52        03BD
                          03BE    810 75$:
06 A0    83     B0        03BE    811          MOVW    (R3)+,6(R0)
                          03C2    812          DELAY
      F6 52     F5        03C5    813          SOBGTR  R2,75$                ; TO RELEASE THE UNIBUS
                          03C8    814                                       ; LOOP TILL DONE
                05        03C8    815 80$:     RSB                         ; RETURN TO CALLER
                          03C9    816
                          03C9    817 90$:
                13        03C9    818          BEQL    100$                ; SKIP IF NONE
                          03CB    819          GETIND  0                   ; SELECT OUTPUT SILO
06 A0    53     90        03DA    820          MOVB    R3,6(R0)            ; OUTPUT CHARACTER
                          03DE    821 100$:
                05        03DE    822          RSB
                          03DF    823          .DISABLE        LSB
                          03DF    824
```

```
                                                I  1
                                        .SBTTL  PORT DMA ROUTINES
                                ;++
                                ; DMA_START     -       INITIATE DMA OUTPUT
                                ;
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ; THESE ROUTINES ARE CALLED BY THE PORT INPUT INTERRUPT, OUTPUT
                                ; INTERRUPT, AND STARTIO TO INITIATE NEW DAM OUTPUT.
                                ; THEY HANDLE ALLOCATION AND LOADING OF MAP REGISTERS
                                ; TO HANDLE DMA OUTPUT. MAP REGISTERS ARE ALLOCATED IN PAIRS
                                ; TO ALLOW OUTPUT BURSTS UP TO 512 BYTES. TRANSFERS LARGER THAN
                                ; THAT ARE DONE IN SEGMENTS. IF INSUFFICIENT MAPS ARE AVAILABLE,
                                ; THE TRANSFER IS DONE SILO MODE.
                                ;
                                ; INPUTS:
                                ;
                                ;       R5 = UCB ADDRESS
                                ;
                                ; OUTPUTS:
                                ;
                                ;       R5 = UCB ADDRESS
                                ;
                                ;       R0 PRESERVED
                                ;       R1,R2,R3,R4 DESTROYED
                                ;--
                                DMA_START:
1000 8F   A8                            BISW    #TTY$M_TANK_DMA,-       ; SHOW DMA MODE ACTIVE
0108 C5                                 UCB$W_TT_HOLD(R5)
     01   8A                            BICB    #TTY$M_TP_ABORT,-       ; RESET ANY OLD ABORT REQUESTS
0130 C5                                 UCB$B_TP_STAT(R5)
                                ;
                                ;       CHECK IF UNIT HAS PERMANENT MAP REGISTERS
                                ;
          03   E1                    BBC     #TTY$V_PC_PRMMAP,-      ; SKIP IF NOT AUTHORIZED FOR PERM MAPS
  06 0122 C5                            UCB$W_TT_PRTCTL(R5),5$
          04   E0                    BBS     #TTY$V_PC_MAPAVL,-      ; SKIP FORK IF MAPS ALLOCATED ALREADY
  49 0122 C5                            UCB$W_TT_PRTCTL(R5),DMA_CONTINUE
                                5$:
          02   88                    BISB    #TTY$M_TP_ALLOC,-       ; SHOW ALLOC FORK ACTIVE
    0130 C5                             UCB$B_TP_STAT(R5)
51    0114 C5   D0                    MOVL    UCB$L_TT_CLASS(R5),R1   ; GET CLASS VECTOR ADDRESS
      1C B1   16                      JSB     @CLASS_FORK(R1)         ; FORK TO FIPL FOR MAP
                                                                      ; REGISTER ALLOCATION
              05                      RSB                             ; RETURN TO CALLER WITH
                                                                      ; "INT" LEFT ON TO INTERLOCK
                                                                      ; OUTPUT. FORK ROUTINE WILL
                                                                      ; RESUME AT DMA_ALLOC.
                                DMA_ALLOC:
                                ;
                                ;
                                5$:
    53   02   9A                    MOVZBL  #2,R3                   ; REQUEST 2 MAP REGISTERS
00000000'GF   16                    JSB     G^IOC$ALOUBAMAPN        ;
                                ;
                                ;       SETIPL  UCB$B_DIPL(R5)          ; INTERLOCK TO DEVICE IPL
          10   A8                    BISW    #TTY$M_PC_MAPAVL,-      ; SHOW MAP ALLOCATED
```

```
                0122 C5      0414   883                 UCB$W_TT_PRTCTL(R5)
                     02  8A  0417   884         BICB    #TTY$M_TP_ALLOC,-       ; SHOW ALLOC FORK DONE
                0130 C5      0419   885                 UCB$B_TP_STAT(R5)
                  17 50  E8  041C   886         BLBS    R0,20$                  ; SUCCESS
                           041F   887
           50     24 A5  D0  041F   888         MOVL    UCB$L_CRB(R5),R0        ; GET CRB OF UNIT
           50     2C B0  D0  0423   889         MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0; GET CSR
                1000 8F  AA  0427   890         BICW    #TTY$M_TANK_DMA,-       ; RESET DMA MODE
                0108 C5      042B   891                 UCB$W_TT_HOLD(R5)
        52      0120 C5  3C  042E   892         MOVZWL  UCB$W_TT_OUTLEN(R5),R2  ; RESTORE OUTPUT LENGTH
                  FF49   31  0433   893         BRW     SILO_OUTPUT             ; USE SILO FOR OUTPUT
                           0436   894
                           0436   895 20$:
           50     24 A5  D0  0436   896         MOVL    UCB$L_CRB(R5),R0        ; GET CRB ADDRESS
                  34 A0  D0  043A   897         MOVL    CRB$L_INTD+VEC$W_MAPREG(R0),-
                0120 C5      043D   898                 UCB$L_TP_MAP(R5)        ; SAVE  MAP FIELD IN UCB
                           0440   899
                           0440   900
                           0440   901 DMA_CONTINUE:
                     00  E0  0440   902         BBS     #TTY$V_TP_ABORT,-       ; BRANCH IF DMA TO BE ABORTED
           0C 0130 C5      0442   903                 UCB$B_TP_STAT(R5),2$
                           0446   904
        53      011C C5  D0  0446   905         MOVL    UCB$L_TT_OUTADR(R5),R3  ; GET ADDRESS OF NEXT STRING
        52      0120 C5  3C  044B   906         MOVZWL  UCB$W_TT_OUTLEN(R5),R2  ; LENGTH OF OUTPUT
                     03  12  0450   907         BNEQ    4$                      ; SKIP IF MORE TO DO
                  00B2   31  0452   908 2$:      BRW     DMA_DONE                ; BRANCH IF TRANSFER IS DONE
                           0455   909 4$:
                     50  DD  0455   910         PUSHL   R0                      ; SAVE INPUT VOLITAL REGISTER "CSR"
                           0457   911
                     52  D1  0457   912         CMPL    R2,#512                 ; NEXT BURST TOO LONG FOR MAPS?
              00000200 8F        0459
                     05  15  045E   913         BLEQ    5$                      ; NO
           52      0200 8F  3C  0460   914         MOVZWL  #512,R2
                           0465   915
                           0465   916
        011C C5      52  C0  0465   917 5$:      ADDL    R2,UCB$L_TT_OUTADR(R5)  ; UPDATE CHARACTER POINTER FOR NEXT TIME
        0120 C5      52  A2  046A   918         SUBW    R2,UCB$W_TT_OUTLEN(R5)  ; UPDATE COUNT FOR NEXT TIME
                           046F   919
                           046F   920         TIMSET  R2,R1,LOCKOUTPUT        ;RECOMPUTE TIMEOUT VALUE FOR THIS
                           0490   921                                         ;PORTION OF THE DMA BURST
                           0490   922
                           0490   923
                           0490   924 ;               R3 - STRING ADDRESS
                           0490   925 ;               R2 - LENGTH
                           0490   926 ;               R5 - UCB
                           0490   927
                     24  BB  0490   928         PUSHR   #^M<R2,R5>
           50     24 A5  D0  0492   929         MOVL    UCB$L_CRB(R5),R0        ; GET CRB ADDRESS
           51     38 B0  D0  0496   930         MOVL    @CRB$L_INTD+VEC$L_ADP(R0),R1    ; CONFIG REGISTER
                  0F 00  EF  049A   931         EXTZV   #VEC$V_MAPREG,#VEC$S_MAPREG,-
           50     012C C5      049D   932                 UCB$L_TP_MAP(R5),R0     ; GET STARTING MAP REGISTER
        51      0800 C140  DE  04A1   933         MOVAL   UBA$L_MAP(R1)[R0],R1    ; GET 1ST MAP REGISTER ADDRESS
                           04A7   934
              80000000 8F  CB  04A7   935         BICL3   #^X80000000,R3,R4       ; CALC SVAPTE OF BUFFER
                     54        04AD
                54     F7 8F  78  04AF   936         ASHL    #-9,R4,R4               ; ISOLATE PAGE
                           54        04B3
```

```
00000000'GF   D0  04B4   937          MOVL    G^MMG$GL_SPTBASE,R5      ; GETS SVAPTE OF BUFFER
          55      04BA
    54  6544   DE  04BB   938          MOVAL   (R5)[R4],R4              ; INTO R4
FFFFFE00 8F   CA  04BF   939          BICL    #^C^X1FF,R3              ; COMPUTE BYTE OFFSET IN PAGE
          53      04C5
                  04C6   940
                  04C6   941 ;        LOAD MAP REGISTERS
                  04C6   942 ;              R0 - MAP REGISTER NUMBER
                  04C6   943 ;              R1 - ADDRESS OF FIRST MAP REGISTER
                  04C6   944 ;              R2 - BUFFER LENGTH
                  04C6   945 ;              R3 - BYTE OFFSET IN PAGE
                  04C6   946 ;              R4 - SVAPTE OF BUFFER
                  04C6   947
    52  02   9A  04C6   948          MOVZBL  #2,R2
    55  84   D0  04C9   949 10$:     MOVL    (R4)+,R5                 ; GET CONTENTS OF NEXT PTE
                  04CC   950
                  04CC   951 ;        THIS CODE ASSUMES THAT DMA IS FROM NONPAGED POOL
                  04CC   952
 00000400 8F   F0  04CC   953          INSV    #^X400,#21,#11,R5        ; SET VALID BIT, DATA PATH 0
55    0B   15      04D2
    81   55   D0  04D5   954          MOVL    R5,(R1)+                 ; LOAD INTO MAP REGISTER
      EE 52   F5  04D8   955          SOBGTR  R2,10$
                  04DB   956
          24  BA  04DB   957          POPR    #^M<R2,R5>               ; RESTORE LENGTH,WRITE BUFFER, UCB
                  04DD   958
09    09   50   F0  04DD   959          INSV    R0,#9,#9,R3              ; COMPUTE UNIBUS ADDRESS
          53      04E1
53    02   10   EF  04E2   960          EXTZV   #16,#2,R3,R0             ; GET HIGH 2 UNIBUS ADDRESS BITS
          50      04E6
02    0E   50   F0  04E7   961          INSV    R0,#14,#2,R2             ; MERGE WITH BYTE COUNT
          52      04EB
                  04EC   962
                  04EC   963          GETIND  2                        ; SELECT DMA INDIRECT REGISTERS
  06 A0   53   B0  04FB   964          MOVW    R3,6(R0)                 ; LOAD ADDRESS (INDIRECT REGISTER
                  04FF   965                                           ; AUTO INCREMTS TO COUNT)
  06 A0   52   B0  04FF   966          MOVW    R2,6(R0)                 ; LOAD COUNT/ INIT TRANSFER
                  0503   967
          50 8ED0  0503   968          POPL    R0                       ; RESTORE CSR ADDRESS
             05   0506   969          RSB                              ; RETURN TO CALLER
                  0507   970                                           ; FORK DISPATCHER, ISR, OR STARTIO
                  0507   971
                  0507   972 DMA_DONE:                                 ; DMA COMPLETION
                  0507   973
                  0507   974
          03   E0  0507   975          BBS     #TTY$V_PC_PRMMAP,-       ; SKIP FORK IF MAPS PERMANENT
  2E 0122 C5       0509   976                  UCB$W_TT_PRTCTL(R5),DMA_POST
          04   88  050D   977          BISB    #TTY$M_TP_DLLOC,-        ; SHOW DEALLOC FORK ACTIVE
    0130 C5       050F   978                  UCB$B_TP_STAT(R5)
51  0114 C5   D0  0512   979          MOVL    UCB$L_TT_CLASS(R5),R1    ; GET CLASS VECTOR ADDRESS
    1C B1   16  0517   980          JSB     @CLASS_FORK(R1)          ; SCHEDULE FORK TO FIPL FOR MAP
                  051A   981                                           ; REGISTER DEALLOCATION
             05   051A   982          RSB                              ; RETURN TO CALLER, FORK WILL RESUME
                  051B   983                                           ; AT DMA_DEALLOC
                  051B   984
                  051B   985 DMA_DEALLOC:
50    24 A5   D0  051B   986          MOVL    UCB$L_CRB(R5),R0         ; GET CRB ADDRESS
    012C C5   D0  051F   987          MOVL    UCB$L_TP_MAP(R5),-
```

```
                        0523   988                 CRB$L_INTD+VEC$W_MAPREG(R0); RESTORE MAP FIELD IN CRB
          34 A0         0525   989         BEQL    5$                          ; SKIP IF NONE
          06 13         0527   990         JSB     G^IOC$RELMAPREG             ; RELEASE MAP REGISTERS
00000000'GF 16          052D   991
                        052D   992 5$:      SETIPL  UCB$B_DIPL(R5)              ; INTERLOCK TO DEVICE IPL
          10 AA         0531   993         BICW    #TTY$M_PC_MAPAVL,-          ; SHOW MAP ALLOCATED
        0122 C5         0533   994                 UCB$W_TT_PRTCTL(R5)
          04 8A         0536   995         BICB    #TTY$M_TP_DLLOC,-           ; SHOW DEALLOC FORK DONE
        0130 C5         0538   996                 UCB$B_TP_STAT(R5)
                        053B   997
                        053B   998 DMA_POST:
          01 8A         053B   999         BICB    #TTY$M_TP_ABORT,-           ; RESET ABORT REQUEST
        0130 C5         053D  1000                 UCB$B_TP_STAT(R5)
        1000 8F AA      0540  1001         BICW    #TTY$M_TANK_DMA,-           ; RESET DMA MODE
        0108 C5         0544  1002                 UCB$W_TT_HOLD(R5)
                        0547  1003
                        0547  1004 ;        CALL GETNEXT TO CONTINUE PROCESSING
                        0547  1005 ;
          03 8A         0547  1006         BICB    #UCB$M_TIM!UCB$M_INT,-  ;CLEAR TIMEOUT AND INT EXPECTED
          64 A5         0549  1007                 UCB$W_STS(R5)
        010C D5 16      054B  1008         JSB     @UCB$L_TT_GETNXT(R5)        ; GET NEXT BURST
          FE0F 31       054F  1009         BRW     YC$STARTIO                  ; AND PROCEED
                        0552  1010
                        0552  1011 YC$FORK:                                    ;
                        0552  1012         SAVIPL                              ; SAVE CURRENT IPL ON THE STACK
0000056D'EF DF          0555  1013         PUSHAL  20$                         ; BUILD RETURN ADDRESS ON STACK
          01 E1         055B  1014         BBC     #TTY$V_TP_ALLOC,-           ; SKIP IF NOT ALLOCATE FORK
       03 0130 C5       055D  1015                 UCB$B_TP_STAT(R5),10$
          FEA1 31       0561  1016         BRW     DMA_ALLOC                   ; RESUME AT ALLOCATE CODE THREAD
                        0564  1017 10$:
          02 E1         0564  1018         BBC     #TTY$V_TP_DLLOC,-           ; CHECK FOR DEALLOCATE
       03 0130 C5       0566  1019                 UCB$B_TP_STAT(R5),20$
          FFAE 31       056A  1020         BRW     DMA_DEALLOC
                        056D  1021 20$:
                        056D  1022         ENBINT                              ; RESTORE SAVED FORK IPL FROM STACK
                        0570  1023
                05      0570  1024         RSB
                        0571  1025
                        0571  1026
```

```
                     0571  1028          .SBTTL  PORT ROUTINES  STOP,RESUME,XON,XOFF
                     0571  1029 ;++
                     0571  1030 ; YC$XOFF -      SEND XOFF
                     0571  1031 ; YC$XON -       SEND XON
                     0571  1032 ; YC$STOP -      STOP OUTPUT
                     0571  1033 ; YC$ABORT -     ABORT CURRENT OUTPUT
                     0571  1034 ; YC$RESUME -    RESUME STOPPED OUTPUT
                     0571  1035 ;
                     0571  1036 ; FUNCTIONAL DESCRIPTION:
                     0571  1037 ;
                     0571  1038 ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                     0571  1039 ; CONTROL OUTPUT ON THE PORT
                     0571  1040 ;
                     0571  1041 ; INPUTS:
                     0571  1042 ;
                     0571  1043 ;       R5 = UCB ADDRESS
                     0571  1044 ;
                     0571  1045 ; OUTPUTS:
                     0571  1046 ;
                     0571  1047 ;       R5 = UCB ADDRESS
                     0571  1048 ;--
                     0571  1049 ;
                     0571  1050 ; SCHEDULE XOFF OR XON TO BE SEND
                     0571  1051 ;
                     0571  1052 ; INPUTS:
                     0571  1053 ;
                     0571  1054 ;       R3 - CONTAINS THE CHARACTER TO SEND AS FLOW CONTROL.
                     0571  1055 ;
                     0571  1056 YC$XOFF:
                     0571  1057 YC$XON:
          03   BB    0571  1058          PUSHR   #^M<R0,R1>
                     0573  1059          GETIND  1,R1
51   8000 8F   A9    0586  1060          BISW3   #DMFLCT$M_PREMPT,R1,6(R0)        ; PREMPT ANY CURRENT OUTPUT
       06 A0         058B
     4040 8F   A9    058D  1061          BISW3   #^X4040,UCB$W_UNIT(R5),(R0)      ; SELECT TRANSMIT SILO
60     54 A5         0591
06 A0  53   90    0594  1062          MOVB    R3,6(R0)                        ; OUTPUT CHARACTER
          03   BA    0598  1063          POPR    #^M<R0,R1>
               05    059A  1064          RSB
                     059B  1065 ;
                     059B  1066 ; STOP PORT OUTPUT
                     059B  1067 ;
                     059B  1068 YC$STOP:
          03   BB    059B  1069          PUSHR   #^M<R0,R1>
          03   8A    059D  1070          BICB    #UCB$M_INT!UCB$M_TIM,-          ; RESET TIMER AND OUTPUT ACTIVE
       64 A5         059F  1071                  UCB$W_STS(R5)
                     05A1  1072          GETIND  1                               ;
                     05B0  1073
0122 C5    05   E1    05B0  1074          BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),10$; AUTOXON XOFF AVAILABLE ON
               06    05B5
                     05B6  1075                                                  ; YES THEN IS
0122 C5    06   E0    05B6  1076          BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),20$; AUTOXON XOFF ENABLED
               04    05BB
06 A0      01   AA    05BC  1077 10$:     BICW2   #DMFLCT$M_XMIT,6(R0)            ; RESET TRANSMIT ENABLE
                     05C0  1078 20$:
          03   BA    05C0  1079          POPR    #^M<R0,R1>
               05    05C2  1080          RSB
```

```
                     05C3  1081 ;
                     05C3  1082 ; ABORT ANY CURRENT PORT OUTPUT ACTIVITY
                     05C3  1083 ;
                     05C3  1084 YC$ABORT:
              03 BB  05C3  1085         PUSHR   #^M<R0,R1>
0108 C5       0B E5  05C5  1086         BBCC    #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),-   ; RESET BURST ACTIVE
              00     05CA  1087                 10$                                     ;
                     05CB  1088 10$:
                     05CB  1089         GETIND  1
   06 A0      01 A8  05DA  1090         BISW2   #<DMFLCT$M_XMIT>,6(R0)                  ; SET XMIT
09 64 A5      01 E1  05DE  1091         BBC     #UCB$V_INT,UCB$W_STS(R5),15$            ; SKIP IF NOT BUSY.
   06 A0      10 A8  05E3  1092         BISW2   #<DMFLCT$M_FLUSH>,6(R0)                 ; FLUSH OUTPUT
0130 C5       01 88  05E7  1093         BISB    #TTY$M_TP_ABORT,UCB$B_TP_STAT(R5)       ; REQUEST DMA ABORT
                     05EC  1094 15$:
              03 BA  05EC  1095         POPR    #^M<R0,R1>
                 05 05EE  1096         RSB
                     05EF  1097
                     05EF  1098 ;
                     05EF  1099 ; RESUME PREVIOUSLY STOPPED PORT OUTPUT
                     05EF  1100 ;
                     05EF  1101 YC$RESUME:
              0F BB  05EF  1102         PUSHR   #^M<R0,R1,R2,R3>
                     05F1  1103         GETIND  1
                     0600  1104
0122 C5       05 E1  0600  1105         BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),10$; AUTOXON XOFF AVAILABLE ON
                 06 0605
                     0606  1106                                                 ; NO THEN RESUME TRANSMITION
0122 C5       06 E0  0606  1107         BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),20$; AUTOXON XOFF ENABLED
                 04 060B
                     060C  1108                                                 ; YES THEN DON'T ENABLE THE
   06 A0      01 A8  060C  1109 10$:    BISW2   #DMFLCT$M_XMIT,6(R0)                  ; ENABLE TRANSMIT
38 64 A5      01 E0  0610  1110 20$:    BBS     #UCB$V_INT,UCB$W_STS(R5),40$            ; SKIP IF OUTPUT ON
              0C E0  0615  1111         BBS     #TTY$V_TANK_DMA,-                       ; SPECIAL IF DMA ACTIVE
   35 0108 C5         0617  1112                 UCB$W_TT_HOLD(R5),50$
0108 C5       0B E1  061B  1113         BBC     #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),40$ ; NO BURST IN PROGRESS
                 2C 0620
                     0621  1114                                                 ; (RESET ANYWAY. WILL BE
                     0621  1115                                                 ; SET IF NEEDED BY BURST_OUT
   4040 8F      A9 0621  1116         BISW3   #^X4040,UCB$W_UNIT(R5),(R0)             ; SELECT TRANSMIT SILO
60   54 A5         0625
51   06 A0      9A 0628  1117         MOVZBL  6(R0),R1                                ; GET SILO DEPTH
                     062C  1118         TIMSET  R1,R1,LOCKOUTPUT                        ; COMPUTE TIMEOUT AND
                     064D  1119                                                 ; SET INTERUPT EXPECTED
                     064D  1120
                     064D  1121 40$:
              0F BA  064D  1122         POPR    #^M<R0,R1,R2,R3>
                 05 064F  1123         RSB
                     0650  1124
                     0650  1125 50$:
   4058 8F      A9 0650  1126         BISW3   #^X4058,UCB$W_UNIT(R5),(R0)             ; GET CURRENT BYTE COUNT
60   54 A5         0654
51   06 A0      3C 0657  1127         MOVZWL  6(R0),R1
51   C000 8F    AA 065B  1128         BICW    #^XC000,R1                              ; MASK OFF ADDRESS BITS
                     0660  1129         TIMSET  R1,R1,LOCKOUTPUT                        ; COMPUTE TIMEOUT AND
                     0681  1130                                                 ; SET INTERUPT EXPTECTEDD
              CA 11  0681  1131         BRB     40$
                     0683  1132
```

```
                     0683  1134          .SBTTL  OUTPUT INTERRUPT SERVICE
                     0683  1135 ;++
                     0683  1136 ; YC$INTOUT - DMF OUTPUT INTERRUPT SERVICE
                     0683  1137 ;
                     0683  1138 ; FUNCTIONAL DESCRIPTION:
                     0683  1139 ;
                     0683  1140 ; THIS ROUTINE IS ENTERED WHEN THE DMF FINDS A LINE ENABLED
                     0683  1141 ; AND AN EMPTY UART. THE CORRESPONDING UCB IS FOUND AND
                     0683  1142 ; ANY OUTSTANDING PORT OUTPUT IS DONE. WHEN ALL OUTSTANDING PORT
                     0683  1143 ; OUTPUT IS COMPLETED, THE CLASS DRIVER IS CALLED TO RETURN THE NEXT
                     0683  1144 ; CHARACTER OR STRING TO BE OUTPUT. IF NO MORE OUTPUT IS FOUND, THEN
                     0683  1145 ; THE LINE IS DISBALED.
                     0683  1146 ;
                     0683  1147 ; INPUTS:
                     0683  1148 ;
                     0683  1149 ;       SP(00) = ADDRESS OF THE IDB
                     0683  1150 ;
                     0683  1151 ; IMPLICIT INPUTS:
                     0683  1152 ;
                     0683  1153 ;       R0,R1,R2,R3,R4,R5 SAVED ON THE STACK.
                     0683  1154 ;
                     0683  1155 ; OUTPUTS:
                     0683  1156 ;
                     0683  1157 ;       THE INTERRUPT IS DISMISSED.
                     0683  1158 ;
                     0683  1159 ;--
                     0683  1160 YC_OUT_EXIT:                            ; EXIT OUTPUT INTERRUPT
5E   04   C0         0683  1161          ADDL    #4,SP                  ; REMOVE IDB ADDRESS
50   8E   7D         0686  1162          MOVQ    (SP)+,R0               ; RESTORE REGISTERS
52   8E   7D         0689  1163          MOVQ    (SP)+,R2               ;
54   8E   7D         068C  1164          MOVQ    (SP)+,R4               ;
               02    068F  1165          REI                            ; DISMISS INTERRUPT
                     0690  1166
                     0690  1167 YC$INTOUT::                             ; DMF OUTPUT INTERRUPT SERVICE
                     0690  1168
                     0690  1169 YC_OUT_LOOP:
54     00 BE   D0    0690  1170          MOVL    @(SP),R4               ; GET THE IDB ADDRESS
   50    64   D0    0694  1171          MOVL    (R4),R0                ; GET THE CSR ADDRESS
                     0697  1172 ;
                     0697  1173 ; GET THE LINE INFO FROM THE CSR
                     0697  1174 ;
                     0697  1175
   52    60   B0    0697  1176          MOVW    (R0),R2                ; GET THE CSR VALUE
         E7   18    069A  1177          BGEQ    YC_OUT_EXIT            ; NO MORE LINES
52     F8 8F   78    069C  1178          ASHL    #-8,R2,R1              ; GET THE LINE NUMBER
            51       06A0
FFFFFFF8 8F   CA    06A1  1179          BICL    #^C<7>,R1              ;
            51       06A7
55   18 A441 D0    06A8  1180          MOVL    IDB$L_UCBLST(R4)[R1],R5 ; GET THE UCB ADDRESS
         E1   13    06AD  1181          BEQL    YC_OUT_LOOP            ; IF EQL THEN DISMISS
                     06AF  1182 ;
                     06AF  1183 ; CHECK TO MAKE SURE NO DATA IS PENDING BEFORE ASKING FOR MORE
                     06AF  1184 ;
   4040 8F   A9      06AF  1185          BISW3   #^X4040,UCB$W_UNIT(R5),(R0); GET THE SILO INDIRECT REGISTER
60   54 A5           06B3
      06 A0   95     06B6  1186          TSTB    6(R0)                  ; ANY DATA STILL IN THE SILO
         27   12     06B9  1187          BNEQ    40$                    ; YES THEN LET IT COMPLETE
```

```
                        06BB  1188 ;
                        06BB  1189 ;       CHECK FOR BURST OR DMA ACTIVE ON LINE
                        06BB  1190 ;
                        06BB  1191
          08    91      06BB  1192         CMPB    #TTY$M_TANK_BURST@-8,-  ; ONLY BURST ACTIVE?
    0109  C5            06BD  1193                 UCB$W_TT_HOLD+1(R5)
          2C    13      06C0  1194         BEQL    YC_SILO                 ; YES, CONTINUE SILO OUTPUT
                        06C2  1195
          10    93      06C2  1196         BITB    #TTY$M_TANK_DMA@-8,-    ; DMA ACTIVE?
    01J9  C5            06C4  1197                 UCB$W_TT_HOLD+1(R5)
          7A    12      06C7  1198         BNEQ    YC_DMA_INTERRUPT        ; YES, PROCESS IT.
                        06C9  1199 ;
                        06C9  1200 ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
                        06C9  1201 ;
    64 A5 03    8A      06C9  1202 10$:    BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
                        06CD  1203 ;
                        06CD  1204 ; CALL CLASS DRIVER FOR MORE OUTPUT
                        06CD  1205 ;
    010C  D5    16      06CD  1206         JSB     @UCB$L_TT_GETNXT(R5)    ; GET THE NEXT CHARACTER
          15    19      06D1  1207         BLSS    YC_START_BURST          ; BURST SPECIFIED
          BB    13      06D3  1208         BEQL    YC_OUT_LOOP             ; NONE
                        06D5  1209 ;
                        06D5  1210 ; OUTPUT A CHARACTER TO THE DMF
                        06D5  1211 ;
                        06D5  1212 20$:
    4040  8F    A9      06D5  1213         BISW3   #^X4040,UCB$W_UNIT(R5),(R0)     ; SELECT OUTPUT SILO INDIRECT REGIST
    60  54    A5        06D9
    06 A0 53    90      06DC  1214         MOVB    R3,6(R0)                ; OUTPUT CHARACTER
          AE    11      06E0  1215         BRB     YC_OUT_LOOP
                        06E2  1216
    F952  CF    D6      06E2  1217 40$:    INCL    YC$L_SIL_ERROR
          A8    11      06E6  1218         BRB     YC_OUT_LOOP
```

```
                     06E8  1220
                     06E8  1221 YC_START_BURST:
          FC7F CF    16 06E8  1222          JSB     BURST_OUTPUT            ; START OUTPUT SILO OR DMA
               A2    11 06EC  1223          BRB     YC_OUT_LOOP
                     06EE  1224
                     06EE  1225 ;
                     06EE  1226 ;  CONTINUE SILO OUTPUT
                     06EE  1227 ;
                     06EE  1228 YC_SILO:
          4040 8F    A9 06EE  1229          BISW3   #^X4040,UCB$W_UNIT(R5),(R0)     ; SELECT OUTPUT SILO INDIRECT REGIST
    60      54 A5         06F2
    51      06 A0    9A 06F5  1230          MOVZBL  6(R0),R1                ; GET SILO DEPTH
51  20          51    C3 06F9  1231          SUBL3   R1,#32,R1                       ; CONVERT TO NUMBER SLOTS AVAILABLE
                     06FD  1232
52  0120 C5        3C 06FD  1233          MOVZWL  UCB$W_TT_OUTLEN(R5),R2  ; GET CURRENT LENGTH
53  011C C5        D0 0702  1234          MOVL    UCB$L_TT_OUTADR(R5),R3  ; GET CURRENT ADDRESS
          51    52   B1 0707  1235          CMPW    R2,R1                   ; BURST LARGER THAN SILO?
                03   1B 070A  1236          BLEQU   50$                     ; NO
          52    51   9A 070C  1237          MOVZBL  R1,R2                   ; MAXIMUM
                     070F  1238 50$:
011C C5         52   C0 070F  1239          ADDL    R2,UCB$L_TT_OUTADR(R5)  ; UPDATE POINTER
0120 C5         52   A2 0714  1240          SUBW    R2,UCB$W_TT_OUTLEN(R5)  ; AND COUNT
                07   12 0719  1241          BNEQ    60$                     ; NOT DONE
          0800 8F    AA 071B  1242          BICW    #TTY$M_TANK_BURST,-      ; RESET BURST ACTIVE
               0108 C5      071F  1243                  UCB$W_TT_HOLD(R5)
                52   D5 0722  1244 60$:     TSTL    R2                      ; ANY ROOM AT ALL
                1A   13 0724  1245          BEQL    80$                     ; NO THEN EXIT
                     0726  1246
             08 52    E9 0726  1247          BLBC    R2,70$                  ; EVEN TRANSFER
    06 A0       83    90 0729  1248          MOVB    (R3)+,6(R0)             ; OUTPUT ODD BYTE
                52   D7 072D  1249          DECL    R2                      ; UPDATE COUNT
                0F   13 072F  1250          BEQL    80$                     ; DONE
                     0731  1251 70$:
52     FF 8F     78 0731  1252          ASHL    #-1,R2,R2               ; CONVERT TO WORD COUNT
                52        0735
                     0736  1253 75$:
    06 A0       83    B0 0736  1254          MOVW    (R3)+,6(R0)
                     073A  1255          DELAY                           ; TO RELEASE THE UNIBUS
          F6 52      F5 073D  1256          SOBGTR  R2,75$                  ; LOOP TILL DONE
                     0740  1257
            FF4D     31 0740  1258 80$:     BRW     YC_OUT_LOOP
                     0743  1259
                     0743  1260
                     0743  1261 YC_DMA_INTERRUPT:
          FF49 CF    DF 0743  1262          PUSHAL  YC_OUT_LOOP             ; BUILD RETURN ADDRESS ON STACK
    04 52       0C   E1 0747  1263          BBC     #DMFCSR$V_DMAERR,R2,30$ ; CHECK FOR A DMA ERROR
          F8F1 CF    D6 074B  1264          INCL    YC$L_DMAXMT_ERROR       ; ERROR OCCURED INCREMENT COUNTS
                     074F  1265 30$:
                06   93 074F  1266          BITB    #TTY$M_TP_ALLOC!TTY$M_TP_DLLOC,- ;CHECK FOR FORKS ACTIVE
               0130 C5      0751  1267                  UCB$B_TP_STAT(R5)       ; AND IGNORE IF SO
                21   12 0754  1268          BNEQ    10$
0130 C5         00   E0 0756  1269          BBS     #TTY$V_TP_ABORT,UCB$B_TP_STAT(R5),-; ABORT ACTIVE DMA
                18        075B  1270                  20$
          4058 8F    A9 075C  1271          BISW3   #^X4058,UCB$W_UNIT(R5),(R0); GET CURRENT BYTE COUNT
    60      54 A5         0760
    51      06 A0    3C 0763  1272          MOVZWL  6(R0),R1
51  C000 8F    AA   0767  1273          BICW    #^XOC000,R1             ; MASK OFF ADDRESS BITS
```

```
                06   13  076C  1274                BEQL    20$                     ; DMA BYTE COUNT DONE.
            FBCA C1   D6  076E  1275                INCL    YC$L_ERROR
                03   11  0772  1276                BRB     10$                     ; NO THEN CONTINUE
                         0774  1277
                         0774  1278 20$:
                FCC9    31  0774  1279                BRW     DMA_CONTINUE            ; OTHERWISE, CONTINUE THE DMA
                         0777  1280
                         0777  1281                                                ; IF THIS INTERRUPT WAS THE RESULT
                         0777  1282                                                ; OF AN ABORT, THIS WILL BE HANDLED
                         0777  1283                                                ; BY DMA_CONTINUE
                         0777  1284 10$:
                05       0777  1285                RSB
                         0778  1286
```

```
                                0778  1288          .SBTTL  SET SPEED, PARITY PARAMETERS
                                0778  1289
                                0778  1290 ;++
                                0778  1291 ; YC$SET_LINE - RESET SPEED, PARITY
                                0778  1292 ;
                                0778  1293 ; FUNCTIONAL DESCRIPTION:
                                0778  1294 ;
                                0778  1295 ; INPUTS:
                                0778  1296 ;
                                0778  1297 ;       R5 - UCB ADDRESS
                                0778  1298 ;
                                0778  1299 ; OUTPUTS:
                                0778  1300 ;
                                0778  1301 ;       R4 USED
                                0778  1302 ;--
                                0778  1303
                                0778  1304 YC$SET_LINE:
                          53    DD    0778  1305          PUSHL   R3
                    54    24    A5    D0    077A  1306          MOVL    UCB$L_CRB(R5),R4        ; ADDRESS CRB
                    54    2C    B4    D0    077E  1307          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4    ; GET THE CSR ADDRESS VIA CRB
                          4048  8F    A9    0782  1308          BISW3   #^X4048,UCB$W_UNIT(R5),(R4)     ; GET THE RIGHT INDIRECT REGISTER
                    64    54    A5          0786
                0122  C5    05    E1    0789  1309          BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),4$; AUTOXON XOFF AVAILABLE ON T
                                    10    078E
                                          078F  1310                                              ; YES THEN IS
                0122  C5    06    E0    078F  1311          BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),2$; AUTOXON XOFF ENABLED
                                    06    0794
                  06  A4    02    AA    0795  1312          BICW    #DMFLCT$M_AUTOX,6(R4)           ; NO THEN CLEAR THE AUTOXOFF ENABLE
                                    04    11    0799  1313          BRB     4$                      ; AND CONTINUE
                  06  A4    02    A8    079B  1314  2$:     BISW    #DMFLCT$M_AUTOX,6(R4)           ; ENABLED THEN SET AUTOXOFF
                                          079F  1315  4$:
                          7E    D4    079F  1316          CLRL    -(SP)                           ; RESET A TEMPORARY LOCATION
                                          07A1  1317 ;
                                          07A1  1318 ;       SET UP LINE SPEED AND PARITY
                                          07A1  1319 ;
                                          07A1  1320
                    00F5  C5    95    07A1  1321          TSTB    UCB$W_TT_SPEED+1(R5)    ; RECEIVE SPEED SPECIFIED?
                                    07    12    07A5  1322          BNEQ    5$                      ; YES
                    00F4  C5    90    07A7  1323          MOVB    UCB$W_TT_SPEED(R5),-    ; NO, SO USE TRANSMITTER SPEED
                    00F5  C5          07AB  1324                  UCB$W_TT_SPEED+1(R5)
                00F4  C5    01    83    07AE  1325  5$:     SUBB3   #1,UCB$W_TT_SPEED(R5),R3        ; ADJUST TRANSMIT SPEED
                                    53    07B3
                04    0C    53    F0    07B4  1326          INSV    R3,#DMFLPR$V_TSPEED,#4,(SP)     ; SET TRANSMIT SPEED
                                    6E    07B8
                00F5  C5    01    83    07B9  1327          SUBB3   #1,UCB$W_TT_SPEED+1(R5),R3      ; ADJUST RECEIVER SPEED
                                    53    07BE
                04    08    53    F0    07BF  1328          INSV    R3,#DMFLPR$V_RSPEED,#4,(SP)     ; SET RECEIVE SPEED
                                    6E    07C3
                                          07C4  1329
                                          07C4  1330 ;       NONSTANDARD PARITY/SIOP LAYOUT
                                          07C4  1331
                          02    03    EE    07C4  1332          EXTV    #UCB$V_TT_LEN,#2,UCB$B_TT_PARITY(R5),R3         ; GET CHAR SIZE
                53    00F8  C5          07C7
                02    03    53    F0    07CB  1333          INSV    R3,#DMFLPR$V_SIZE,#2,(SP)               ; SET IT
                                    6E    07CF
                          02    06    EE    07D0  1334          EXTV    #UCB$V_TT_PARTY,#2,UCB$B_TT_PARITY(R5),R3       ; GET PARITY/ODD
                53    00F8  C5          07D3
```

```
                02   05   53   F0  07D7  1335          INSV    R3,#DMFLPR$V_PARITY,#2,(SP)
                          6E       07DB
                  6E   40 8F   8C  07DC  1336          XORB    #DMFLPR$M_ODD,(SP)                      ; INVERT PARITY BIT
                       01 05   EE  07E0  1337          EXTV    #UCB$V_TT_STOP,#1,UCB$B_TT_PARITY(R5),R3  ; GET STOP
                53     00F8 C5       07E3
                01     07   53   F0  07E7  1338          INSV    R3,#DMFLPR$V_STOP,#1,(SP)
                          6E       07EB
                                   07EC  1339
                       6E   07   AA  07EC  1340          BICW    #^X0007,(SP)                ; CLEAR SPECIAL FIELDS
                  6E   54   A5   A8  07EF  1341          BISW    UCB$W_UNIT(R5),(SP)         ; SET LINE NUMBER
                  02   A4   8E   F7  07F3  1342          CVTLW   (SP)+,2(R4)                 ; INSERT AS LINE PARAMETER
                            53 8ED0  07F7  1343          POPL    R3
                               05    07FA  1344          RSB
                                   07FB  1345
                                   07FB  1346
                                   07FB  1347
                                   07FB  1348  YC$END:                                       ; End of driver
                                   07FB  1349
                                   07FB  1350          .END
```

```
$$$                     = 00000020 R      02
$$OP                    = 00000002
AT$_UBA                 = 00000001
BIT...                  = 00000010
BURST_OUTPUT              0000036B R      03
CLASS_DDT               = 00000010
CLASS_DS_TRAN           = 0000000C
CLASS_FORK              = 0000001C
CLASS_GETNXT            = 00000000
CLASS_POWERFAIL         = 00000020
CLASS_PUTNXT            = 00000004
CLASS_READERROR         = 00000014
CLASS_SETUP_UCB         = 00000008
CRB$B_TT_TYPE           = 0000000B
CRB$L_INTD              = 00000024
CRB$L_INTD2             = 00000048
DC$_TERM                = 00000042
DDB$L_DDT               = 0000000C
DEV$M_AVL               = 00040000
DEV$M_CCL               = 00000002
DEV$M_IDV               = 04000000
DEV$M_NNM               = 00000200
DEV$M_ODV               = 08000000
DEV$M_REC               = 00000001
DEV$M_TRM               = 00000004
DMA_ALLOC                 00000405 R      03
DMA_CONTINUE              00000440 R      03
DMA_DEALLOC               0000051B R      03
DMA_DONE                  00000507 R      03
DMA_POST                  0000053B R      03
DMA_START                 000003DF R      03
DMFCSR$M_CLEAR          = 00000020
DMFCSR$M_DMAERR         = 00001000
DMFCSR$M_IADDR          = 0000001F
DMFCSR$M_LINE           = 00000700
DMFCSR$M_RCVINT         = 00000040
DMFCSR$M_SNDINT         = 00004000
DMFCSR$M_SNDRDY         = 00008000
DMFCSR$S_CLEAR          = 00000001
DMFCSR$S_DMAERR         = 00000001
DMFCSR$S_IADDR          = 00000005
DMFCSR$S_LINE           = 00000003
DMFCSR$S_RCVINT         = 00000001
DMFCSR$S_SNDINT         = 00000001
DMFCSR$S_SNDRDY         = 00000001
DMFCSR$V_CLEAR          = 00000005
DMFCSR$V_DMAERR         = 0000000C
DMFCSR$V_IADDR          = 00000000
DMFCSR$V_LINE           = 00000008
DMFCSR$V_RCVINT         = 00000006
DMFCSR$V_SNDINT         = 0000000E
DMFCSR$V_SNDRDY         = 0000000F
DMFLCT$M_AUTOX          = 00000002
DMFLCT$M_BREAK          = 00000008
DMFLCT$M_DS_ENABLE      = 00000020
DMFLCT$M_FLOSH          = 00000010
DMFLCT$M_MAINT          = 000000C0
DMFLCT$M_PREMPT         = 00008000
DMFLCT$M_RCV            = 00000004
DMFLCT$M_XMIT           = 00000001
DMFLCT$S_AUTOX          = 00000001
DMFLCT$S_BREAK          = 00000001
DMFLCT$S_DS_ENABLE      = 00000001
DMFLCT$S_FLOSH          = 00000001
DMFLCT$S_MAINT          = 00000002
DMFLCT$S_PREMPT         = 00000001
DMFLCT$S_RCV            = 00000001
DMFLCT$S_XMIT           = 00000001
DMFLCT$V_AUTOX          = 00000001
DMFLCT$V_BREAK          = 00000003
DMFLCT$V_DS_ENABLE      = 00000005
DMFLCT$V_FLOSH          = 00000004
DMFLCT$V_MAINT          = 00000006
DMFLCT$V_PREMPT         = 0000000F
DMFLCT$V_RCV            = 00000002
DMFLCT$V_XMIT           = 00000000
DMFLPR$M_LINE           = 00000007
DMFLPR$M_ODD            = 00000040
DMFLPR$M_PARITY         = 00000020
DMFLPR$M_RSPEED         = 00000F00
DMFLPR$M_SIZE           = 00000018
DMFLPR$M_STOP           = 00000080
DMFLPR$M_TSPEED         = 0000F000
DMFLPR$S_LINE           = 00000003
DMFLPR$S_ODD            = 00000001
DMFLPR$S_PARITY         = 00000001
DMFLPR$S_RSPEED         = 00000004
DMFLPR$S_SIZE           = 00000002
DMFLPR$S_STOP           = 00000001
DMFLPR$S_TSPEED         = 00000004
DMFLPR$V_LINE           = 00000000
DMFLPR$V_ODD            = 00000006
DMFLPR$V_PARITY         = 00000005
DMFLPR$V_RSPEED         = 00000008
DMFLPR$V_SIZE           = 00000003
DMFLPR$V_STOP           = 00000007
DMFLPR$V_TSPEED         = 0000000C
DMFRCV$M_BUF            = 000000FF
DMFRCV$M_DS_CHG         = 00000800
DMFRCV$M_FRAMER         = 00002000
DMFRCV$M_LINE           = 00000700
DMFRCV$M_OVERRUN        = 00004000
DMFRCV$M_PARERR         = 00001000
DMFRCV$M_VALID          = 00008000
DMFRCV$S_BUF            = 00000008
DMFRCV$S_DS_CHG         = 00000001
DMFRCV$S_FRAMER         = 00000001
DMFRCV$S_LINE           = 00000003
DMFRCV$S_OVERRUN        = 00000001
DMFRCV$S_PARERR         = 00000001
DMFRCV$S_VALID          = 00000001
DMFRCV$V_BUF            = 00000000
DMFRCV$V_DS_CHG         = 0000000B
DMFRCV$V_FRAMER         = 0000000D
```

```
DMFRCV$V_LINE                   =  00000008
DMFRCV$V_OVERRUN                =  0000000E
DMFRCV$V_PARERR                 =  0000000C
DMFRCV$V_VALID                  =  0000000F
DPT$C_LENGTH                    =  00000038
DPT$C_VERSION                   =  00000004
DPT$INITAB                        00000038 R      02
DPT$M_NOUNLOAD                  =  00000004
DPT$REINITAB                      000000D3 R      02
DPT$TAB                           00000000 R      02
DPT$W_VECTOR                    =  0000001E
DT$_DMF32                       =  0000000A
DT$_DMZ32                       =  00000045
DYN$C_CRB                       =  00000005
DYN$C_DDB                       =  00000006
DYN$C_DPT                       =  0000001E
DYN$C_ORB                       =  00000049
DYN$C_UCB                       =  00000010
EXE$GL_ABSTIM                     ********    X   03
EXE$GL_TENUSEC                    ********    X   03
EXE$GL_UBDELAY                    ********    X   03
FUNCTAB_LEN                     =  00000000
IDB$B_COMBO_CSR_OFFSET          =  0000000F
IDB$B_COMBO_VECTOR_OFFSET       =  00000010
IDB$B_VECTOR                    =  0000000B
IDB$L_UCBLST                    =  00000018
IO$M_AUTXOF_DIS                 =  00004000
IO$M_AUTXOF_ENA                 =  00002000
IO$M_LINE_OFF                   =  00000200
IO$M_LINE_ON                    =  00000800
IO$M_LOOP                       =  00000080
IO$M_LOOP_EXT                   =  00001000
IO$M_UNLOOP                     =  00000100
IOC$ALOUBAMAPN                    ********    X   03
IOC$MNTVER                        ********    X   03
IOC$RELMAPREG                     ********    X   03
IOC$RETURN                        ********    X   03
MMG$GL_SPTBASE                    ********    X   03
MODEM$C_DATASET                 =  00000003
MODEM$C_INIT                    =  00000000
ORB$B_FLAGS                     =  0000000B
ORB$L_OWNER                     =  00000000
ORB$M_PROT_16                   =  00000001
ORB$W_PROT                      =  00000018
PORT_ABORT                      =  00000020
PORT_DS_SET                     =  0000000C
PORT_FORKRET                    =  00000034
PORT_LENGTH                     =  00000038
PORT_MAINT                      =  00000030
PORT_RESUME                     =  00000024
PORT_SET_LINE                   =  00000008
PORT_STARTIO                    =  00000000
PORT_STOP                       =  00000018
PORT_VECTOR                       00000044 R      03
PORT_XOFF                       =  00000014
PORT_XON                        =  00000010
PR$_IPL                           ********    X   03
SILO_OUTPUT                       0000037F R      03
SIZ...                          =  00000001
SS$_NORMAL                      =  00000001
TT$M_DS_DTR                     =  00000002
TT$M_DS_RTS                     =  00000010
TT$_UNKNOWN                     =  00000000
TTY$GB_DEFSPEED                   ********    X   02
TTY$GB_PARITY                     ********    X   02
TTY$GB_RSPEED                     ********    X   02
TTY$GB_SILOTIME                   ********    X   03
TTY$GL_DEFCHAR                    ********    X   02
TTY$GL_DEFCHAR2                   ********    X   02
TTY$GL_DEFPORT                    ********    X   03
TTY$GL_DPT                        ********    X   03
TTY$GL_OWNUIC                     ********    X   02
TTY$GW_DEFBUF                     ********    X   02
TTY$GW_DMASIZE                    ********    X   03
TTY$GW_PROT                       ********    X   02
TTY$M_PC_DMAAVL                 =  00000004
TTY$M_PC_MAPAVL                 =  00000010
TTY$M_PC_XOFAVL                 =  00000020
TTY$M_TANK_BURST                =  00000800
TTY$M_TANK_DMA                  =  00001000
TTY$M_TP_ABORT                  =  00000001
TTY$M_TP_ALLOC                  =  00000002
TTY$M_TP_DLLOC                  =  00000004
TTY$V_PC_DMAENA                 =  00000001
TTY$V_PC_MAPAVL                 =  00000004
TTY$V_PC_NOTIME                 =  00000000
TTY$V_PC_PRMMAP                 =  00000003
TTY$V_PC_XOFAVL                 =  00000005
TTY$V_PC_XOFENA                 =  00000006
TTY$V_TANK_BURST                =  0000000B
TTY$V_TANK_DMA                  =  0000000C
TTY$V_TP_ABORT                  =  00000000
TTY$V_TP_ALLOC                  =  00000001
TTY$V_TP_DLLOC                  =  00000002
UBA$L_MAP                       =  00000800
UCB$B_DEVCLASS                  =  00000040
UCB$B_DEVTYPE                   =  00000041
UCB$B_DIPL                      =  0000005E
UCB$B_FIPL                      =  0000000B
UCB$B_TP_STAT                   =  00000130
UCB$B_TT_DEPARI                 =  000000EC
UCB$B_TT_DETYPE                 =  000000F0
UCB$B_TT_DS_RCV                 =  00000124
UCB$B_TT_DS_TX                  =  00000125
UCB$B_TT_MAINT                  =  0000012A
UCB$B_TT_PARITY                 =  000000F8
UCB$C_TT_LENGTH                 =  00000134
UCB$L_CRB                       =  00000024
UCB$L_DDB                       =  00000028
UCB$L_DDT                       =  00000088
UCB$L_DEVCHAR                   =  00000038
UCB$L_DEVCHAR2                  =  0000003C
UCB$L_DEVDEPEND                 =  00000044
UCB$L_DEVDEPND2                 =  00000048
```

```
UCB$L_DUETIM                    = 0000006C
UCB$L_TP_MAP                    = 0000012C
UCB$L_TT_CLASS                  = 00000114
UCB$L_TT_DECHA1                 = 000000C8
UCB$L_TT_DECHAR                 = 000000C4
UCB$L_TT_GETNXT                 = 0000010C
UCB$L_TT_OUTADR                 = 0000011C
UCB$L_TT_PORT                   = 00000118
UCB$L_TT_PUTNXT                 = 00000110
UCB$L_TT_RTIMOU                 = 000000B4
UCB$L_TT_WBLINK                 = 000000D0
UCB$L_TT_WFLINK                 = 000000CC
UCB$M_INT                       = 00000002
UCB$M_ONLINE                    = 00000010
UCB$M_TIM                       = 00000001
UCB$V_INT                       = 00000001
UCB$V_POWER                     = 00000005
UCB$V_TT_LEN                    = 00000003
UCB$V_TT_PARTY                  = 00000006
UCB$V_TT_STOP                   = 00000005
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_STS                       = 00000064
UCB$W_TT_DESIZE                 = 000000F1
UCB$W_TT_DESPEE                 = 000000E8
UCB$W_TT_HOLD                   = 00000108
UCB$W_TT_OUTLEN                 = 00000120
UCB$W_TT_PRTCTL                 = 00000122
UCB$W_TT_SPEED                  = 000000F4
UCB$W_TT_UNITBIT                = 00000106
UCB$W_UNIT                      = 00000054
VEC$L_ADP                       = 00000014
VEC$L_IDB                       = 00000008
VEC$L_INITIAL                   = 0000000C
VEC$L_UNITINIT                  = 00000018
VEC$S_MAPREG                    = 0000000F
VEC$V_MAPREG                    = 00000000
VEC$W_MAPREG                    = 00000010
YC$ABORT                          000005C3  R      03
YC$CTRL_ERROR                     0000010D  R      03
YC$DDT                            00000000  RG     03
YC$DPT                            00000000  RG     02
YC$DS_SET                         00000295  R      03
YC$END                            000007FB  R      03
YC$FORK                           00000552  R      03
YC$INITIAL                        00000081  RG     03
YC$INITLINE                       0000010E  RG     03
YC$INTINP                         000002C4  RG     03
YC$INTOUT                         00000690  RG     03
YC$L_DMAXMT_ERROR                 00000040  R      03
YC$L_ERROR                        0000003C  R      03
YC$L_SIL_ERROR                    00000038  R      03
YC$MAINT                          000001C0  R      03
YC$NULL                           0000080D  R      03
YC$RESUME                         000005EF  R      03
YC$SET_LINE                       00000778  R      03
YC$STARTIO                        00000361  RG     03
YC$STOP                           0000059B  R      03
YC$UNIT_ERROR                     000001BB  R      03
YC$VEC                            00000044  R      03
YC$VECEND                         0000007C  R      03
YC$XOFF                           00000571  R      03
YC$XON                            00000571  R      03
YC_DMA_INTERRUPT                  00000743  R      03
YC_OUT_EXIT                       00000683  R      03
YC_OUT_LOOP                       00000690  R      03
YC_SILO                           000006EE  R      03
YC_START_BURST                    000006E8  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	000007FB (2043.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	105	00:00:00.24	00:00:01.14
Command processing	112	00:00:00.57	00:00:02.20
Pass 1	805	00:00:37.95	00:01:04.18
Symbol table sort	0	00:00:04.87	00:00:07.07
Pass 2	415	00:00:08.30	00:00:14.86
Symbol table output	2	00:00:00.26	00:00:00.44
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1443	00:00:52.26	00:01:29.95

The working set limit was 2550 pages.
214887 bytes (420 pages) of virtual memory were used to buffer the intermediate code.
There were 160 pages of symbol table space allocated to hold 2949 non-local and 79 local symbols.
1363 source lines were read in Pass 1, producing 22 object records in Pass 2.
72 pages of virtual memory were used to define 67 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA18:[SYS.OBJ]LIB.MLB;1	32
_255DUA18:[SYSLIB]STARLET.MLB;3	11
TOTALS (all libraries)	43

3430 GETS were required to define 43 macros.

There were no errors, warnings or information messages.

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

0449 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

YCDRIVER
LIS

TTYSTRSTP
LIS

TTYSUB
LIS

0450 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

UAFMAIN
LIS

UAF

AUTHORIZE
MAP