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

FILEID**YCDRIVER

M 15

```
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

(1)	187	DECLARATIONS
(1)	319	REGISTER DEFINITIONS
(1)	402	CONTROLLER INITIALIZATION
(1)	467	UNIT INITIALIZATION
(1)	537	MAINTENANCE ROUTINES
(1)	632	OUTPUT MODEM CONTROL
(1)	664	RECEIVER INTERRUPT SERIVCE
(1)	755	START I/O ROUTINE
(1)	826	PORT DMA ROUTINES
(1)	1028	PORT ROUTINES STOP,RESUME,XON,XOFF
(1)	1134	OUTPUT INTERRUPT SERVICE
(1)	1288	SET SPEED, PARITY PARAMETERS

```
0000          .TITLE  YCDRIVER - Port Driver for DMF  Async
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ; FACILITY:
0000    ;
0000    ;       VAX/VMS TERMINAL DRIVER
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       DMF  ASYNC PORT DRIVER
0000    ;
0000    ; AUTHOR:
0000    ;
0000    ;       RICK SPITZ
0000    ;
0000    ; Revision history:
0000    ;
0000    ;       V03-032 WHM0003         Bill Matthews           24-Jul-1984
0000    ;               Set the device type field in the CRB.
0000    ;
0000    ;       V03-231 LMP0275         L. Mark Pilant,         12-Jul-1984  21:04
0000    ;               Initialize the ACL info in the ORB to be a null descriptor
0000    ;               list rather than an empty queue.  This avoids the overhead
0000    ;               of locking and unlocking the ACL mutex, only to find out
0000    ;               that the ACL was empty.
0000    ;
0000    ;       V03-030 EMD0089         Ellen M. Dusseault      30-Apr-1984
0000    ;               Add DEV$M_NNM characteristic to DEVCHAR2 so that these
0000    ;               devices will have the prefix "node$".
0000    ;
0000    ;       V03-029 LMP0221         L. Mark Pilant,         31-Mar-1984  10:23
0000    ;               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
```

```
0000 ;          ORB$W_PROT.
0000 ;
0000 ;  V03-028 MIR0360         Michael I. Rosenblum    20-Mar-1984
0000 ;          Add code to check for DMF honoring XON and XOFF.
0000 ;          Remove Callback to BURST_OUTPUT when resume is called
0000 ;          This illiminates unnecessry interrupts when the silo
0000 ;          Goes empty and we get a control-Q.
0000 ;
0000 ;  V03-027 WHM0002         Bill Matthews   29-Feb-1984
0000 ;          Second part of change to control_init to support
0000 ;          multiple comm devices on one board.
0000 ;
0000 ;  V03-026 WHM0001         Bill Matthews           28-Dec-1983
0000 ;          Use IDB$B_COMBO_VECTOR and IDB$B_COMBO_CSR_OFFSET to find
0000 ;          the start of the combo device's CSR and load the soft
0000 ;          interrupt vector.
0000 ;
0000 ;  V03-025 JLV0318         Jake VanNoy             9-DEC-1983
0000 ;          comment out DPT_STORE for parity to fix build problem.
0000 ;
0000 ;  V03-024 MIR0083         Michael I. Rosenblum            02-Aug-1983
0000 ;          Add Field test DMZ-32 support.  Remove local def of
0000 ;          IO$M_AUTXOF_ENA and IO$M_AUTXOF_DIS and move them into STARDEF.
0000 ;
0000 ;  V03-023 MIR0062         Michael I. Rosenblum            30-Jun-1983
0000 ;          Change controler and unit init to use newly defined
0000 ;          macros.  Remove code that looked at the logical UCB
0000 ;          (code moved to the class drivers setup ucb and transition
0000 ;          routine) Remove into the class driver on resume with nothing
0000 ;          to resume.
0000 ;
0000 ;  V03-022 MIR0060         Michael I. Rosenblum            1-Jun-1983
0000 ;          Add code to combat against silo full conditions and spurious
0000 ;          DMA completeion interupts.
0000 ;
0000 ;  V03-021 MIR0050         Michael I. Rosenblum            11-May-1983
0000 ;          Remove STOP2 entery point (no longer necessary)
0000 ;          add code in set_line to allow the use of the auto xon xoff
0000 ;          feature of the dmf.  Fix bug that would cause setting of the
0000 ;          XON bit not to restart a write.
0000 ;
0000 ;  V03-020 RKS0020         RICK SPITZ                      14-MAR-1983
0000 ;          ADD ENHANCEMENTS TO SUPPORT LOGICAL UCB
0000 ;
0000 ;  V03-019 RKS0019         RICK SPITZ                      1-FEB-1983
0000 ;          DMA CODE HAS BEEN MOVED INTO THE DMF PORT DRIVER. THIS
0000 ;          CODE HAS BEEN REMOVED FROM THE CLASS LEVEL. ALL OUTPUT
0000 ;          FROM CLASS TO PORT IS NOW HANDLED USING THE SAME INTERFACE.
0000 ;          RESUME LOGIC HAS BEEN CHANGED TO INSURE INT IS ON DURING
0000 ;          DMA BURST AND BYTE COUNT = 0 TO INSURE NO NEW OUTPUT IS
0000 ;          STARTED DURING THAT WINDOW.
0000 ;
0000 ;  V03-018 MIR0022         Michael I. Rosenblum            18-Jan-1983
0000 ;          Replace old vector table with new vector table macro.
0000 ;          Remove references to UCB$L_DEVDEPEND and UCB$Q_TT_STATE
0000 ;          move these references into the class driver jacket routines
0000 ;
```

```
0000 ;	V03-017 MIR0017		Michael I. Rosenblum	04-Jan-1983
0000 ;		Add powerfail check in the Unit init routine to allow the
0000 ;		terminal class drier to take positive action on powerfail.
0000 ;
0000 ;	V03-016 MIR0016		Michael I. Rosenblum	29-Dec-1982
0000 ;		Replace time calculation code with TIMSET macro call.
0000 ;
0000 ;	V03-015 MIR0015		Michael I. Rosenblum	22-Dec-1982
0000 ;		Change code in YC$DISCONNECT to reflect the change
0000 ;		in functionality.  Remove code that referenced
0000 ;		DMAABO and DMA  state bits.
0000 ;
0000 ;	V03-014 MIR0014		Michael I. Rosenblum	17-Dec-1982
0000 ;		move code to figure out what flow control character
0000 ;		to send to TTYSUB.
0000 ;
0000 ;	V03-013 RKS0013		RICK SPITZ		18-OCT-1982
0000 ;		CORRECT SENSE OF PARITY BIT
0000 ;
0000 ;	V03-010 RKS0012		RICK SPITZ		16-SEP-1982
0000 ;		Remove check for powerfail in unit init which prevents
0000 ;		SETUP_UCB from being called. This is needed to allow
0000 ;		Unibus switches to connect a line and then switch the
0000 ;		device on the bus, followed by a psuedo powerfail recovery.
0000 ;
0000 ;		Check reference count in unit init to determine if
0000 ;		modem control should be initialized or hungup. This
0000 ;		is needed to insure that a hangup ^Y is posted on
0000 ;		powerfail on modem lines.
0000 ;
0000 ;		Refrain from clearing INT in the port abort logic.
0000 ;		This prevents output data following a flush from being
0000 ;		lost during a small time window.
0000 ;		The DMF will interrupt on silo flush complete, at
0000 ;		which point INT will be handled.
0000 ;
0000 ;	V03-009 KDM0002		Kathleen D. Morse	28-Jun-1982
0000 ;		Added $DEVDEF and $SSDEF.
0000 ;
0000 ;	V03-008 RKS008		RICK SPITZ		23-MAR-1982
0000 ;		USE SYSGEN VALUE FOR COMBO INPUT SILO TIMER.
0000 ;		CHANGE OUTPUT DELAY.
0000 ;
0000 ;	V03-007 RKS007		RICK SPITZ		13-FEB-1982
0000 ;		ADD DELAY LOGIC WHEN OUTPUTING TO TRANSMIT SILO
0000 ;
0000 ;	V03-006 RKS0006		RICK SPITZ		8-FEB-1982
0000 ;		DO NOT ALLOW UNLOAD
0000 ;		FIX CHARACTER TIMER
0000 ;		DO UNSIGNED COMPARISON IN SILO OUTPUT.
0000 ;
0000 ;	V03-005 RKS0005		RICK SPITZ		11-JAN-1982
0000 ;		REPAIR MODEM INTERRUPT LOGIC
0000 ;		ADD INPUT SILO TIMER.
0000 ;
0000 ;	V03-004 RKS0004		RICK SPITZ		15-DEC-1981
0000 ;		FIX MAINTENANCE BIT OFFSETS, ADD EXTERNAL LOOP SUPPORT.
```

```
0000  ;              ENHANCE RESUME LOGIC IN DMA MODE TO AVOID WRITING 0 BYTE
0000  ;              COUNT.
0000  ;              USE SYSPARAM VALUES FOR CLASS DRIVER LOCATION,PARITY,RSPEED
0000  ;              CHANGE TYPE CODE TO DMF-32.
0000  ;
0000  ;      V03-003 JLV0104         Jake VanNoy                 27-Oct-1981
0000  ;              Changed TTYDEFS to $TTYDEFS.
0000  ;
0000  ;      V03-002 RKS002          RICK SPITZ                       8-OCT-1981
0000  ;              USE SUBTRACT TO COMPUTE VECTOR ADDRESS TO BE LOADED INTO
0000  ;              DMF-32 VECTOR REGISTER.
0000  ;
0000  ;
0000  ;--
```

```
0000          .SBTTL  DECLARATIONS
0000
0000  ;
0000  ; EXTERNAL DEFINITIONS:
0000  ;
0000          $ACBDEF                              ; DEFINE ACB
0000          $CRBDEF                              ; DEFINE CRB
0000          $DCDEF                               ; DEFINE ADAPTER TYPES
0000          $DDBDEF                              ; DEFINE DDB
0000          $DEVDEF                              ; DEFINE DEVICE TYPES
0000          $DYNDEF                              ; DEFINE DYNAMIC STRUCTURE TYPES
0000          $IDBDEF                              ; DEFINE IDB OFFSETS
0000          $IODEF                               ; DEFINE I/O FUNCTION CODES
0000          $IRPDEF                              ; DEFINE IRP
0000          $ORBDEF                              ; DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
0000          $SSDEF                               ; DEFINE SYSTEM STATUS CODES
0000          $TTYDEF                              ; DEFINE TERMINAL DRIVER SYMBOLS
0000          $TTDEF                               ; DEFINE TERMINAL TYPES
0000          $TT2DEF                              ; DEFINE EXTENDED CHARACTERISTICS
0000          $TQEDEF                              ; DEFINE TIMER QUEUE OFFSETS
0000          $UCBDEF                              ; DEFINE UCB
0000          $UBADEF                              ; DEFINE UBA
0000          $VECDEF                              ; DEFINE VECTOR FOR CRB
0000          $TTYMACS                             ; DEFINE TERMINAL DRIVER MACROS
0000  .show me
0000          $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
          0C00      $EQU    UCB$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_O  260
00000104  0000             UCB$B_TT_ESC_O=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    $EQU     PORT$_PORT_DEF  56
        00000038 0000             PORT$_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
```

```
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
```

```
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
```

```
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
                 0000                  $DEFEND TTYDEFS,,DEF
                 0000                  .MACRO  $TTYDEFSDEF A
                 0000                  .ENDM   $TTYDEFSDEF
                 0000                  .IIF    DIF <> <GLOBAL>,.DISABLE        SUPPRESSION
                 0000                  .CROSS
                 0000                  .RESTORE
                 0000
                 0000
                 0000      .noshow me
                 0000              $TTYMODEM                       ; DEFINE MODEM DEFINITIONS
                 0000
                 0000
                 0000      ;
                 0000      ; LOCAL STORAGE
                 0000      ;
```

```
00000000          .PSECT  $$$105_PROLOGUE
    0000
    0000  ;
    0000  ; Driver prologue table:
    0000  ;
    0000
    0000  YC$DPT::                                          ; DRIVER START
    0000          DPTAB   -                                 ; DRIVER PROLOGUE TABLE
    0000                  END=YC$END,-                      ; End and offset to INIT's vectors
    0000                  UCBSIZE=UCB$C_TT_LENGTH,-         ; SIZE OF UCB
    0000                  FLAGS=DPT$M_NOUNLOAD,-            ; NO UNLOAD ALLOWED
    0000                  ADAPTER=UBA,-                     ; ADAPTER TYPE
    0000                  DEFUNITS=8,-                      ; Number of units to create
    0000                  NAME=YCDRIVER,-                   ; NAME OF DRIVER
    0000                  VECTOR=PORT_VECTOR                ; PORT DRIVER VECTOR TABLE
    0038          DPT_STORE INIT
    0038          DPT_STORE UCB,UCB$B_FIPL,B,8              ; FORK IPL
    003C          DPT_STORE UCB,UCB$L_DEVCHAR,L,<-          ; CHARACTERISTICS
    003C                          DEV$M_REC!-               ;
    003C                          DEV$M_AVL!-               ;
    003C                          DEV$M_IDV!-               ;
    003C                          DEV$M_ODV!-               ;
    003C                          DEV$M_TRM!-               ;
    003C                          DEV$M_CCL>
    0043          DPT_STORE UCB,UCB$L_DEVCHAR2,L,-          ; DEVICE CHARACTERISTICS
    0043                          <DEV$M_NNM>               ; PREFIX WITH "NODE$"
    004A          DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_TERM;
    004E          DPT_STORE UCB,UCB$B_TT_DETYPE,B,TT$_UNKNOWN          ; TYPE
    0052          DPT_STORE UCB,UCB$W_TT_DESIZE,@W,TTY$GW_DEFBUF       ; BUFFER SIZE
    0059          DPT_STORE UCB,UCB$L_TT_DECHAR,@L,TTY$GL_DEFCHAR      ; DEFAULT CHARACTERS
    0060          DPT_STORE UCB,UCB$L_TT_DECHA1,@L,TTY$GL_DEFCHAR2;    DEFAULT CHARACTERS
    0067          DPT_STORE UCB,UCB$W_TT_DESPEE,@B,TTY$GB_DEFSPEED;    DEFAULT SPEED
    006E          DPT_STORE UCB,UCB$W_TT_DESPEE+1,@B,TTY$GB_RSPEED;    DEFAULT RSPEED
    0075          DPT_STORE UCB,UCB$B_TT_DEPARI,@B,TTY$GB_PARITY       ; DEFAULT PARITY
    007C  ;**     DPT_STORE UCB,UCB$B_TT_PARITY,@B,TTY$GB_PARITY       ; DEFAULT PARITY
    007C          DPT_STORE UCB,UCB$B_DEVTYPE,B,TT$_UNKNOWN            ; TYPE
    0080          DPT_STORE UCB,UCB$W_DEVBUFSIZ,@W,TTY$GW_DEFBUF       ; BUFFER SIZE
    0087          DPT_STORE UCB,UCB$L_DEVDEPEND,@L,TTY$GL_DEFCHAR      ; DEFAULT CHARACTERS
    008E          DPT_STORE UCB,UCB$L_DEVDEPND2,@L,TTY$GL_DEFCHAR2;    DEFAULT CHARACTERS
    0095          DPT_STORE UCB,UCB$W_TT_SPEED,@B,TTY$GB_DEFSPEED      ; DEFAULT SPEED
    009C          DPT_STORE UCB,UCB$W_TT_SPEED+1,@B,TTY$GB_RSPEED      ; DEFAULT RSPEED
    00A3          DPT_STORE UCB,UCB$B_DIPL,B,21                        ; DEVICE IPL
    00A7          DPT_STORE UCB,UCB$L_TT_WFLINK,L,0                    ; Zero write queue.
    00AE          DPT_STORE UCB,UCB$L_TT_WBLINK,L,0                    ; Zero write queue.
    00B5          DPT_STORE UCB,UCB$L_TT_RTIMOU,L,0                    ; Zero read timed out disp.
    00BC          DPT_STORE ORB,ORB$B_FLAGS,B,-             ; Protection block flags
    00BC                          <ORB$M_PROT_16>           ; SOGW protection word
    00C0          DPT_STORE ORB,ORB$W_PROT,@W,TTY$GW_PROT   ; Default allocation protection
    00C7          DPT_STORE ORB,ORB$L_OWNER,@L,TTY$GL_OWNUIC ; Default owner UIC
    00CE          DPT_STORE DDB,DDB$L_DDT,D,YC$DDT
    00D3
    00D3          DPT_STORE REINIT
    00D3          DPT_STORE CRB,CRB$L_INTD+4,D,YC$INTINP    ; RECEIVER INTERRUPT
    00D8          DPT_STORE CRB,CRB$L_INTD2+4,D,YC$INTOUT   ; TRANSMITTER INTERRUPT
    00DD          DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,D,YC$INITIAL   ; CONTROLLER INIT
    00E2          DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,YC$INITLINE; UNIT INIT
    00E7          DPT_STORE END
```

```
               0000
               0000          DDTAB   DEVNAM = YC,-    ; DUMMY DMF PORT DRIVER DISPATCH TABLE
               0000                  START  = 0,-
               0000                  FUNCTB = 0
               0038
           00000038          .PSECT  $$$115_DRIVER,LONG
               0038
               0038   ;
               0038   ; THE ASSOCIATED CLASS DRIVER USES THIS TABLE TO COMMAND THE PORT DRIVER.
               0038   ; THE ADDRESS OF THIS TABLE IS CONTAINED IN THE TERMINAL UCB EXTENSION AREA.
               0038   ; THE OFFSET DEFINITONS ARE DEFINED BY TTYDEFS.
               0038   ;
               0038   YC$L_SIL_ERROR:
   00000000    0038          .LONG   0
               003C   YC$L_ERROR:
   00000000    003C          .LONG   0
               0040   YC$L_DMAXMT_ERROR:
   00000000    0040          .LONG   0
               0044   PORT_VECTOR:
               0044
               0044   ;
               0044   ; DMF SPECIFIC DISPATCH TABLE
               0044   ;
               0044          $VECINI YC,YC$NULL
               007C          $VEC    STARTIO,YC$STARTIO          ; START NEW OUTPUT
               0048          $VEC    SET_LINE,YC$SET_LINE        ; SET NEW SPEED/PARITY
               0050          $VEC    DS_SET,YC$DS_SET            ; SET OUTPUT MODEM SIGNALS
               0054          $VEC    XON,YC$XON                  ; SEND XON SEQUENCE
               0058          $VEC    XOFF,YC$XOFF                ; SEND XOFF SEQUENCE
               005C          $VEC    STOP,YC$STOP                ; STOP OUTPUT
               0060          $VEC    ABORT,YC$ABORT              ; ABORT OUTPUT IN PROGRESS
               0068          $VEC    RESUME,YC$RESUME            ; RESUME STOPPED OUTPUT
               006C          $VEC    MAINT,YC$MAINT              ; INVOKE MAINTENANCE FUNCTION
               0078          $VEC    FORKRET,YC$FORK             ; PORT FORK CALLBACK
               007C          $VECEND
               0080
               0080
               0080   ;
               0080   YC$NULL:                                   ; NULL PORT ROUTINE
         05    0080          RSB
               0081
```

```
0081          .SBTTL  REGISTER DEFINITIONS
0081
0081   ;
0081   ; CSR BIT DEFINITIONS ( CSR ) ( READ/WRITE )
0081   ; (NOTE: THIS REGISTER MUST ONLY BE READ IN RESPONSE TO A TRANSMIT INTERRUPT.
0081   ;        TO LOAD AN INDIRECT REGISTER, ONLY WRITE OPERATIONS MAY BE USED)
0081
0081          $VIELD  DMFCSR,0,<-
0081                  <IADDR,5,M>,-     ; INDERECT REGISTER ADDRESS
0081                  <CLEAR,1,M>,-     ; MASTER RESET
0081                  <RCVINT,1,M>,-    ; RECEIVER INTERRUPT ENABLE
0081                  <,1,>,-           ; DON'T USE THIS BIT
0081                  <LINE,3,M>,-      ; LINE NUMBAE (0 - 7)
0081                  <,1,>,-
0081                  <DMAERR,1,M>,-    ; DMA TRANSMIT ERROR
0081                  <,1,>,-
0081                  <SNDINT,1,M>,-    ; TRANSMIT INTERRUPT ENABLE
0081                  <SNDRDY,1,M>-     ; TRANSMITTER READY
0081                  >
0081   ;
0081   ; RECEIVER BUFFER ( CSR+4 ) ( READ ONLY )
0081   ;
0081          $VIELD  DMFRCV,0,<-
0081                  <BUF,8,M>,-       ; RECEIVER DATA
0081                  <LINE,3,M>,-      ; LINE NUMBER (0 - 7)
0081                  <DS_CHG,1,M>,-    ; DATA SET CHANGE INTERRUPT
0081                  <PARERR,1,M>,-    ; PARITY ERROR
0081                  <FRAMER,1,M>,-    ; FRAME ERROR
0081                  <OVERRUN,1,M>,-   ; OVERRUN ERROR
0081                  <VALID,1,M>-      ; DATA VALID
0081                  >
0081   ;
0081   ; LINE PARAMETER REGISTER ( CSR+2 ) ( WRITE ONLY )
0081   ;
0081
0081          $VIELD  DMFLPR,0,<-
0081                  <LINE,3,M>,-      ; LINE NUMBER (0-7)
0081                  <SIZE,2,M>,-      ; CHARACTER SIZE
0081                  <PARITY,1,M>,-    ; PARITY ENABLE
0081                  <ODD,1,M>,-       ; ODD PARITY
0081                  <STOP,1,M>,-      ; NUMBER STOP BITS
0081                  <RSPEED,4,M>,-    ; RECEIVER LINE SPEED
0081                  <TSPEED,4,M>,-    ; TRANSMITTER LINE SPEED (BOTH RVC/TX FOR LINES 2-7)
0081                  >
0081   ;
0081   ; LINE CONTROL INDIRECT REGISTER (INDIRECT REGISTER #2)
0081   ;
0081
0081          $VIELD  DMFLCT,0,<-
0081                  <XMIT,1,M>,-      ; TRANSMIT ENABLE
0081                  <AUTOX,1,M>,-     ; AUTO XON/OFF
0081                  <RCV,1,M>,-       ; RECEIVER ENABLE
0081                  <BREAK,1,M>,-     ; SEND BREAK
0081                  <FLUSH,1,M>,-     ; FLUSH OUTPUT SILO/ABORT TRANSMIT
0081                  <DS_ENABLE,1,M>,- ; ENABLE MODEM TRANSITION INTERRUPTS
0081                  <MAINT,2,M>,-     ;
0081                  <,7,>,-
```

```
0081                   <PREMPT,1,M>,-  ; PREMPT CURRENT OUTPUT
0081                   >
0081
0081      ;
0081      ; MACRO USED TO ACCESS INDIRECT REGISTERS
0081      ;
0081
0081             .MACRO  GETIND  OFFSET,RX
0081             MOVL    UCB$L_CRB(R5),R0                    ; GET CRB ADDRESS
0081             MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0        ; GET CSR ADDRESS
0081             BISW3   #<<^X4040>+<OFFSET*8>>,UCB$W_UNIT(R5),(R0); SELECT INDIRECT FIELD
0081             .IF     NB      RX
0081             MOVW    6(R0),RX                            ; GET CONTENTS OF INDIRECT F
0081             .ENDC
0081             .ENDM   GETIND
0081
0081      ;      OUTPUT DELAY MACRO
0081      ;      TO RELEASE THE UNIBUS
0081
0081             .MACRO  DELAY
0081             .REPEAT 3
0081             NOP
0081             .ENDR
0081             .ENDM
0081
```

```
                              0081          .SBTTL  CONTROLLER INITIALIZATION
                              0081
                              0081
                              0081
                              0081     ;++
                              0081     ; YC$INITIAL - INITIALIZE INTERFACE
                              0081     ;
                              0081     ; FUNCTIONAL DESCRIPTION:
                              0081     ;
                              0081     ; THIS ROUTINE IS ENTERED AT SYSTEM STARTUP AND POWER RECOVERY.
                              0081     ;
                              0081     ; INPUTS:
                              0081     ;
                              0081     ;        R4 = ADDRESS OF THE UNIT CSR
                              0081     ;        R5 = IDB OF UNIT
                              0081     ;        R8 = ADDRESS OF THE UNIT CRB
                              0081     ;
                              0081     ; OUTPUTS:
                              0081     ;
                              0081     ;        R2 is destroyed.
                              0081     ;
                              0081     ; IMPLICIT INPUTS:
                              0081     ;
                              0081     ;        IPL = IPL$_POWER
                              0081     ;
                              0081     ;--
                              0081     YC$INITIAL::                                     ; INITIALIZE DMF UNIT
                              0081     ;
                              0081     ; SET UP CONTROLLER
                              0081     ;
                              0081          CLASS_CTRL_INIT YC$DPT,PORT_VECTOR; RELOCATE THE NECESSARY TABLES
                              00AD     25$:
            0B A8      95     00AD          TSTB    CRB$B_TT_TYPE(R8)               ; CONTROLLER TYPE INITIALIZED?
               0E      12     00B0          BNEQ    30$                             ; IF NEQ YES
        0B A8  0A      90     00B2          MOVB    #DT$_DMF32,CRB$B_TT_TYPE(R8)    ; ASSUME DMF32
            10 A5      95     00B6          TSTB    IDB$B_COMBO_VECTOR_OFFSET(R5)   ; DMF32 HAS A VECTOR OFFSET OF 16
               05      12     00B9          BNEQ    30$                             ; IF NEQ NO, MUST BE A DMZ32
     0B A8  45 8F      90     00BB          MOVB    #DT$_DMZ32,CRB$B_TT_TYPE(R8)    ; SET CONTROLLER TYPE DMZ32
                              00C0     30$:
        50  0F A5      98     00C0          CVTBL   IDB$B_COMBO_CSR_OFFSET(R5),R0   ; GET THE OFFSET TO THE MAIN CSR
            10 A5      83     00C4          SUBB3   IDB$B_COMBO_VECTOR_OFFSET(R5),- ; LOAD THE SOFT VECTOR
      6440  0B A5             00C7                  IDB$B_VECTOR(R5),(R4)[R0]       ;
        64     20      B0     00CB          MOVW    #DMFCSR$M_CLEAR,(R4)            ; CONTROLLER RESET
                              00CE
                              00CE     ;
                              00CE     ;        WAIT TILL CONTROLLER INITIALIZATION IS COMPLETE
                              00CE     ;
                              00CE          TIMEWAIT        #100,#DMFCSR$M_CLEAR,(R4),W,.FALSE.
                              00F5
      64   4040 8F     B0     00F5          MOVW    #<<DMFCSR$M_RCVINT>!-   ; ENABLE RECEIVER INTERRUPTS
                              00FA                  <DMFCSR$M_SNDINT>>,(R4) ; ENABLE TRANSMITTER INTERRUPTS
            09 50      E9     00FA          BLBC    R0,YC$CTRL_ERROR
                              00FD
                              00FD
                              00FD     100$:
04 A4   00000000'GF    90     00FD          MOVB    G^TTY$GB_SILOTIME,4(R4) ; INIT INPUT SILO TIMEOUT VALUE
                              0105
```

```
         0105
         0105
   05    0105              RSB
         0106
         0106    YC$CTRL_ERROR:
   05    0106              RSB
         0107
```

```
                            0107          .SBTTL  UNIT INITIALIZATION
                            0107
                            0107  ; YC$INITLINE - UNIT INITIALIZATION
                            0107  ;
                            0107  ; FUNCTIONAL DESCRIPTION:
                            0107  ;
                            0107  ; THIS ROUTINE PERFORMS A SIMPLE UNIT INITIALIZATION.
                            0107  ;
                            0107  ; INPUTS:
                            0107  ;
                            0107          R5 = UCB ADDRESS
                            0107  ;
                            0107  ; OUTPUTS:
                            0107  ;
                            0107          R4,R5 ARE PRESERVED.
                            0107  ;--
                            0107
                            0107  YC$INITLINE::
        50    FF39 CF   DE  0107          MOVAL   YC$VEC,R0                ; GET THE DISPATCH TABLE ADDRESS
                            010C          CLASS_UNIT_INIT
          64 A5    10   A8  0155          BISW    #UCB$M_ONLINE,UCB$W_STS(R5); SET ONLINE
                            0159
    53     01 54 A5     78  0159  10$:    ASHL    UCB$W_UNIT(R5),#1,R3     ; BUILD UNIT'S BIT MASK
       0106 C5     53   B0  015E          MOVW    R3,UCB$W_TT_UNITBIT(R5)  ; SAVE IT
                            0163
                   24   A8  0163          BISW    #TTY$M_PC_DMAAVL!TTY$M_PC_XOFAVL,-; SHOW DMA FEATURE AVAILABLE FOR U
              0122 C5       0165                  UCB$W_TT_PRTCTL(R5)      ; IN PORT LEVEL
                            0168
        51    0114 C5   D0  0168          MOVL    UCB$L_TT_CLASS(R5),R1    ; ADDRESS CLASS VECTOR TABLE
                08 B1   16  016D          JSB     @CLASS_SETUP_UCB(R1)     ; INIT UCB FIELDS
                            0170
                            0170  ;SET MODE CODE NEEDS TO TOGGLE THESE BITS
                            0170
          00000764'EF   16  0170          JSB     YC$SET_LINE              ; INIT SPEED/PARITY
                            0176
                            0176
                            0176  ;
                            0176  ; ENABLE LINE RECEIVER , TRANSMITTER AND MODEM INTERRUPTS
                            0176  ;
                            0176
  64   54 A5   4048 8F  A9  0176          BISW3   #^X4048,UCB$W_UNIT(R5),(R4)       ; SELECT LCT
          06 A4    25   B0  017D          MOVW    #<<DMFLCT$M_XMIT>!-               ; ENABLE TRANSMIT
                            0181                  <DMFLCT$M_RCV>!-                  ; ENABLE RECEIVE
                            0181                  <DMFLCT$M_DS_ENABLE>>,6(R4)       ; AND MODEM INTERRUPTS
                            0181  ;
                            0181  ; INIT RECEIVER MODEM STATUS FOR DMF
                            0181  ;
                            0181
  64   54 A5   4040 8F  A9  0181          BISW3   #^X4040,UCB$W_UNIT(R5),(R4)       ; SELECT RECV MODEM INDIRECT REG
       0124 C5  07 A4   90  0188          MOVB    7(R4),UCB$B_TT_DS_RCV(R5)         ; UPDATE RECEIVER MODEM STATUS
                            018E
                            018E  25$:
             51    00   9A  018E          MOVZBL  #MODEM$C_INIT,R1         ; ASSUME INIT MODEM PROTOCOL
        50    0114 C5   D0  0191          MOVL    UCB$L_TT_CLASS(R5),R0    ; ADDRESS CLASS VECTOR TABLE
                0C B0   16  0196          JSB     @CLASS_DS_TRAN(R0)       ; INVOKE TO INIT MODEM PROTOCOL
                            0199  30$:
     08 64 A5      05   E1  0199          BBC     #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
```

```
  50      0114 C5   D0 019E            MOVL    UCB$L_TT_CLASS(R5),R0    ; GET THE CLASS VECTOR TABLE ADDRESS
             20 B0  17 01A3            JMP     @CLASS_POWERFAIL(R0)     ; AND GOTO THE POWERFAIL CODE
                       01A6
                    05 01A6   40$:     RSB
                       01A7   ;
                       01A7   ; ERROR DETECTED DURING INITIALIZATION
                       01A7   ;
                       01A7
                       01A7   YC$UNIT_ERROR:
      64 A5     10  AA 01A7            BICW    #UCB$M_ONLINE,UCB$W_STS(R5)      ; UNIT NOT ON LINE
                    05 01AB            RSB
                       01AC
```

```
                        01AC            .SBTTL  MAINTENANCE ROUTINES
                        01AC     ;++
                        01AC     ; YC$MAINT - MAINTENANCE FUNCTIONS
                        01AC     ;
                        01AC     ; FUNCTIONAL DESCRIPTION:
                        01AC     ; THIS ROUTINE PERFORMS MAINTENANCE FUNCTIONS FOR THE DMF
                        01AC     ; (LOOPBACK IS ONLY ALLOWED ON LINES 0 AND 1)
                        01AC     ;
                        01AC     ; INPUTS:
                        01AC     ;
                        01AC     ;       R5 = UBC ADDRESS
                        01AC     ;       UCB$B_TT_MAINT = FUNCTION TO BE PERFORMED
                        01AC     ;
                        01AC     ; OUTPUTS:
                        01AC     ;       R0-R4 SCRATCH
                        01AC     ;--
                        01AC
                        01AC     YC$MAINT:
             01    93   01AC            BITB    #IO$M_LOOP@-7,-                 ; LOOPBACK FUNCTION
        012A C5         01AE                    UCB$B_TT_MAINT(R5)
             0E    13   01B1            BEQL    5$                              ; NO
             12    B0   01B3            MOVW    #TT$M_DS_DTR!TT$M_DS_RTS,-      ; SET REQUIRED MODEM LINES
             52         01B5                    R2
    00000281'EF    16   01B6            JSB     YC$DS_SET                       ; SET THEM TO ALLOW LOOPBACK
          52 02    3C   01BC            MOVZWL  #^X02,R2                        ; SPECIFY LOOPBACK CODE
             15    11   01BF            BRB     10$
                        01C1     5$:
             02    93   01C1            BITB    #IO$M_UNLOOP@-7,-               ; RESET LOOPBACK FUNCTION
        012A C5         01C3                    UCB$B_TT_MAINT(R5)
             42    13   01C6            BEQL    15$                             ; NO
        1300 8F    B0   01C8            MOVW    #256*<TT$M_DS_DTR!TT$M_DS_RTS!^X01>,-; RESET REQUIRED MODEM LINES
             52         01CC                    R2
    00000281'EF    16   01CD            JSB     YC$DS_SET                       ; SET THEM TO ALLOW LOOPBACK
          52 00    3C   01D3            MOVZWL  #^X00,R2                        ; SPECIFY UNLOOP CODE
                        01D6     10$:
       51  24 A5   D0   01D6            MOVL    UCB$L_CRB(R5),R1                ; GET THE CRB ADDRESS
    0B A1  45 8F   91   01DA            CMPB    #DT$_DMZ32,CRB$B_TT_TYPE(R1)    ; IS THIS A DMZ
             06    13   01DF            BEQL    12$                             ; YES THEN MODEM CONTROL ON ALL UNIT
       54 A5  02   B1   01E1            CMPW    #2,UCB$W_UNIT(R5)               ; ALLOWED ON THIS UNIT?
             20    15   01E5            BLEQ    50$                             ; NO
                        01E7     12$:   GETIND  1,R1                            ; GET INDIRECT FIELD
   51   02  06  52 F0   01FA            INSV    R2,#DMFLCT$V_MAINT,#2,R1        ; SET MAINT FIELD
       06 A0  51   B0   01FF            MOVW    R1,6(R0)                        ; UPDATE
          50  01   9A   0203            MOVZBL  #1,R0                           ; INDICATE SUCCESS
             05         0206            RSB
                        0207     50$:
          50       D4   0207            CLRL    R0
             05         0209            RSB
                        020A
                        020A     15$:
        40 8F      93   020A            BITB    #IO$M_AUTXOF_ENA@-7,-
        012A C5         020D                    UCB$B_TT_MAINT(R5)              ;AUTOXON ENABLED
             05    13   0210            BEQL    17$                             ; NO THEN MAYBE DISABLE
             20    A8   0212            BISW    #TTY$M_PC_XOFAVL,-
        0122 C5         0214                    UCB$W_TT_PRTCTL(R5)             ; SET THE BIT AVAILABLE
                        0217
        80 8F      93   0217     17$:   BITB    #IO$M_AUTXOF_DIS@-7,-
```

```
     012A C5       021A                  UCB$B_TT_MAINT(R5)               ;AUTOXON disabled
       05    13    021D           BEQL   19$                              ; no then don't disable it
       20    AA    021F           BICW   #TTY$M_PC_XOFAVL,-
     0122 C5       0221                  UCB$W_TT_PRTCTL(R5)
                   0224
                   0224    19$:
                   0224
       20    93    0224           BITB   #IO$M_LOOP_EXT@-7,-              ; LOOPBACK FUNCTION
     012A C5       0226                  UCB$B_TT_MAINT(R5)
       0E    13    0229           BEQL   20$                              ; NO
       13    B0    022B           MOVW   #<TT$M_DS_DTR!TT$M_DS_RTS!^X01>,-; SET REQUIRED MODEM LINES
       52          022D                  R2
00000281'EF  16    022E           JSB    YC$DS_SET                        ; SET THEM TO ALLOW LOOPBACK
    52    00 3C    0234           MOVZWL #^X00,R2                         ; SPECIFY  NO LOOPBACK CODE
       9D    11    0237           BRB    10$
                   0239    20$:                                           ; CHECK OTHER FUNCTIONS
       04    93    0239           BITB   #IO$M_LINE_OFF@-7,-              ; LINE OFF
     012A C5       023B                  UCB$B_TT_MAINT(R5)
       1C    13    023E           BEQL   30$                              ; NO
                   0240           GETIND 1,R1                             ; GET FIELD
  51   05    AA    0253           BICW   #<<DMFLCT$M_XMIT>!-              ; DISABLE TRANSMIT AND RECEIVE
                   0256                  <DMFLCT$M_RCV>>,R1
06 A0  51    B0    0256           MOVW   R1,6(R0)                         ; UPDATE
       21    11    025A           BRB    40$
                   025C    30$:
       10    93    025C           BITB   #IO$M_LINE_ON@-7,-               ; LINE ON
     012A C5       025E                  UCB$B_TT_MAINT(R5)
       A4    13    0261           BEQL   50$                              ; NO
                   0263           GETIND 1,R1                             ; GET FIELD
  51   05    A8    0276           BISW   #<<DMFLCT$M_XMIT>!-              ; ENABLE TRANSMIT AND RECEIVE
                   0279                  <DMFLCT$M_RCV>>,R1
06 A0  51    B0    0279           MOVW   R1,6(R0)                         ; UPDATE
                   027D
  50   01    9A    027D    40$:   MOVZBL #1,R0
             05    0280           RSB
                   0281
                   0281
```

```
                              0281           .SBTTL  OUTPUT MODEM CONTROL
                              0281      ;++
                              0281      ; YC$DS_SET - SET OUTPUT MODEM SIGNALS
                              0281      ;
                              0281      ; FUNCTIONAL DESCRIPTION:
                              0281      ;
                              0281      ; THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT
                              0281      ;
                              0281      ; INPUTS:
                              0281      ;
                              0281      ;        R2 = LOW BYTE - SIGNALS TO ACTIVATE
                              0281      ;             HIGH BYTE- SIGNALS TO DEACTIVATE
                              0281      ;
                              0281      ;        R5 = UCB ADDRESS
                              0281      ;
                              0281      ; OUTPUTS:
                              0281      ;
                              0281      ;        R0-R3 ARE USED.
                              0281      ;--
                              0281
                              0281      YC$DS_SET:
    0125 C5     52      88    0281             BISB    R2,UCB$B_TT_DS_TX(R5)     ; SET NEW OUTPUT SIGNALS
52    52     F8 8F      78    0286             ASHL    #-8,R2,R2                 ; ACCESS SIGNALS TO RESET
    0125 C5     52      8A    028B             BICB    R2,UCB$B_TT_DS_TX(R5)     ; RESET THEM
                              0290             GETIND  1,-(SP)
0125 C5      E0 8F      8B    02A3             BICB3   #^X0E0,UCB$B_TT_DS_TX(R5),- ; OR IN OUTPUT MODEM SIGNALS
                01 AE         02A9                     1(SP)
       06 A0    8E      B0    02AB             MOVW    (SP)+,6(R0)               ; OUTPUT UPDATED VALUE
                        05    02AF             RSB
                              02B0
                              02B0
```

```
                              02B0         .SBTTL  RECEIVER INTERRUPT SERIVCE
                              02B0   ;++
                              02B0   ; YC$INTINP - DMF RECEIVER READY INTERRUPTS
                              02B0   ;
                              02B0   ; FUNCTIONAL DESCRIPTION:
                              02B0   ;
                              02B0   ; THIS ROUTINE IS ENTERED WHEN A CHARACTER IS AVAILABLE IN THE UNIT'S
                              02B0   ; SILO. THE CHARACTER IS EXTRACTED AND IS PASSED TO THE ASSOCIATED
                              02B0   ; CLASS DRIVER. IF THE CLASS DRIVER RETURNS CHARACTERS(S) THEN NEW
                              02B0   ; OUTPUT IT INITIATED (NORMALLY ECHO).
                              02B0   ;
                              02B0   ; INPUTS:
                              02B0   ;
                              02B0   ;       00(SP) = ADDRESS OF IDB
                              02B0   ;
                              02B0   ; IMPLICIT INPUTS:
                              02B0   ;
                              02B0   ;       R0,R1,R2,R3,R4,R5 ARE SAVED ON STACK.
                              02B0   ;
                              02B0   ; OUTPUTS:
                              02B0   ;
                              02B0   ;       THE INTERRUPT IS DISMISSED WHEN THE SILO IS EMPTY.
                              02B0   ;
                              02B0   ;--
                              02B0   YC$INTINP::                                  ; DMF INPUT INTERRUPTS
                              02B0   ;
                              02B0   ; GET THE CSR ADDRESS
                              02B0   ;
            54    9E     D0   02B0         MOVL    @(SP)+,R4                    ; GET THE IDB ADDRESS
                  54     DD   02B3         PUSHL   R4                           ; SAVE IDB ADDRESS
            50    64     D0   02B5         MOVL    (R4),R0                      ; GET THE CSR ADDRESS
                              02B8   ;
                              02B8   ; GET THE CHARACTER FROM THE INTERFACE
                              02B8   ;
       53    04 A0       B0   02B8   25$:  MOVW    4(R0),R3                     ; GET THE CHARACTER,ERRORS AND LINE NUMBER
                  60     18   02BC         BGEQ    100$                         ;SILO EMPTY
       53  7800 8F       B3   02BE         BITW    #<DMFRCV$M_PARERR>!-
                              02C3                 <DMFRCV$M_OVERRUN>!-
                              02C3                 <DMFRCV$M_FRAMER>!-
                              02C3                 <DMFRCV$M_DS_CHG>,R3         ;ERRORS OR MODEM TRANSITION ?
                              02C3
                  36     12   02C3         BNEQ    50$                          ;YES,PROCESS THEM
    52    53    F8 8F    78   02C5   27$:  ASHL    #-8,R3,R2                    ; GET THE LINE NUMBER
    52  FFFFFFF8 8F      CA   02CA         BICL    #^C<7>,R2                    ;
            53    53     9A   02D1         MOVZBL  R3,R3                        ; CLEAR THE HIGH BYTES OF CHARACTER
       55    18 A442     D0   02D4         MOVL    IDB$L_UCBLST(R4)[R2],R5      ; GET THE UCB FOR THAT LINE
                  DD     13   02D9         BEQL    25$                          ; IF EQL THEN NOT THERE
            0110 D5      16   02DB         JSB     @UCB$L_TT_PUTNXT(R5)         ; BUFFER THE CHARACTER
                  10     15   02DF         BLEQ    40$                          ; NONE OR STRING OUTPUT
 60    54 A5    4040 8F  A9   02E1         BISW3   #^X4040,UCB$W_UNIT(R5),(R0)      ; SELECT OUTPUT SILO INDIRECT REGIST
         06 A0    53     90   02E8         MOVB    R3,6(R0)                     ; OUTPUT CHARACTER
            54    6E     D0   02EC   30$:  MOVL    (SP),R4                      ; GET IDB ADDRESS
                  C7     11   02EF         BRB     25$                          ; CONTINUE
                              02F1   40$:
                  F9     13   02F1         BEQL    30$                          ; NO CHARACTER
       00000357'EF       16   02F3         JSB     BURST_OUTPUT                 ; START BURST
                  F1     11   02F9         BRB     30$
```

```
                              02FB  ;
                              02FB  50$:
                              02FB  ;
                              02FB  ; PROCESS PARITY, FRAME OVERRUN ERROR OR MODEM TRANSITION
                              02FB  ;
   52    53    F8 8F     78   02FB           ASHL    #-8,R3,R2                    ; GET LINE NUMBER
   52   FFFFFFF8 8F      CA   0300           BICL    #^C<7>,R2                    ;
       55    18 A442     D0   0307           MOVL    IDB$L_UCBLST(R4)[R2],R5      ; GET UCB ADDRESS
                  0E     13   030C           BEQL    70$                          ; IF EQL THEN NO UCB
         19 53    0B     E0   030E           BBS     #DMFRCV$V_DS_CHG,R3,200$;    ; MODEM TRANSITION
       52   0114 C5      D0   0312           MOVL    UCB$L_TT_CLASS(R5),R2        ; GET CLASS DISPATCH
               14 B2     16   0317           JSB     @CLASS_READERROR(R2)         ; SIGNAL ERROR
                  A9     12   031A           BNEQ    27$                          ; CHRRACTER TO ECHO
                              031C  70$:
                  CE     11   031C           BRB     30$
                              031E
             5E   04     C0   031E  100$:    ADDL    #4,SP                        ; REMOVE IDB ADDRESS
             50   8E     7D   0321           MOVQ    (SP)+,R0                     ; RESTORE REGISTERS
             52   8E     7D   0324           MOVQ    (SP)+,R2                     ;
             54   8E     7D   0327           MOVQ    (SP)+,R4                     ;
                  02          032A           REI
                              032B  200$:
60    52   4040 8F       A9   032B           BISW3   #^X4040,R2,(R0)              ; SELECT MODEM INDIRECT REGISTER
        52    07 A0      9A   0331           MOVZBL  7(R0),R2                     ; GET CURRENT RECEIVE MODEM SIGNALS
      0124 C5    52      90   0335           MOVB    R2,UCB$B_TT_DS_RCV(R5)       ; UPDATE CURRENT INPUT MODEM SIGNALS
             51    03    9A   033A           MOVZBL  #MODEM$C_DATASET,R1          ; TRANSITION TYPE IS DATASET
                   50    DD   033D           PUSHL   R0                           ; SAVE CSR ADDRESS
       54   0114 C5      D0   033F           MOVL    UCB$L_TT_CLASS(R5),R4        ; GET CLASS DISPATCH
               0C B4     16   0344           JSB     @CLASS_DS_TRAN(R4)           ; INVOKE TRANSITION ROUTINE
               50 8ED0        0347           POPL    R0                           ; RESTORE CSR ADDRESS
                FF9F     31   034A           BRW     30$
                              034D
                              034D
```

```
                              034D          .SBTTL  START I/O ROUTINE
                              034D   ;++
                              034D   ; YC$STARTIO - START I/O OERATION ON DMF
                              034D   ;
                              034D   ; FUNCTIONAL DESCRIPTION:
                              034D   ;
                              034D   ; THIS ROUTINE IS ENTERED FROM THE DEVICE INDEPENDENT TERMINAL STARTIO
                              034D   ; ROUTINE TO ENABLE OUTPUT INTERRUPTS ON AN IDLE DMF UNIT.
                              034D   ;
                              034D   ; INPUTS:
                              034D   ;
                              034D   ;          R3 =     CHARACTER        AND      CC = PLUS
                              034D   ;                   ADDRESS          AND      CC = NEGATIVE
                              034D   ;
                              034D   ;          R5 = UCB ADDRESS
                              034D   ;
                              034D   ; OUTPUTS:
                              034D   ;
                              034D   ;          R5 = UCB ADDRESS
                              034D   ;--
                              034D          .ENABLE LSB
                              034D   YC$STARTIO::                                     ; START I/O ON UNIT
                66      18    034D          BGEQ    90$                               ; SINGLE CHARACTER
     50     24 A5       D0    034F          MOVL    UCB$L_CRB(R5),R0                  ; GET CRB OF UNIT
     50     2C B0       D0    0353          MOVL    @CRB$L_INTD+VEC$L_IDB(R0),R0; GET CSR
                              0357
                              0357   BURST_OUTPUT:
  52     0120 C5        3C    0357          MOVZWL  UCB$W_TT_OUTLEN(R5),R2            ; GET LENGTH
                01      E1    035C          BBC     #TTY$V_PC_DMAENA,-                ; USE SILO IF DMA NOT ENABLED ON THIS LINE
       09 0122 C5             035E                  UCB$W_TT_PRTCTL(R5),SILO_OUTPUT
  00000000'GF   52      B1    0362          CMPW    R2,G^TTY$GW_DMASIZE               ; LARGE ENOUGH FOR DMA
                60      18    0369          BGEQ    DMA_START                         ; YES SO DO DMA
                              036B
                              036B   SILO_OUTPUT:
60    54 A5    4040 8F  A9    036B          BISW3   #^X4040,UCB$W_UNIT(R5),(R0); SELECT TRANSMIT SILO
     51     06 A0       9A    0372          MOVZBL  6(R0),R1                          ; GET SILO DEPTH
     51     20    51    C3    0376          SUBL3   R1,#32,R1                              ; CONVERT TO NUMBER SLOTS AVAILABLE
                              037A
          51      52    B1    037A          CMPW    R2,R1                             ; BURST LARGER THAN SILO?
                  03    1B    037D          BLEQU   50$                               ; NO
          52      51    9A    037F          MOVZBL  R1,R2                             ; SLOTS AVAILABLE IS MAXIMUM
                              0382   50$:
  53     011C C5  D0          0382          MOVL    UCB$L_TT_OUTADR(R5),R3            ; GET ADDRESS
  011C C5     52  C0          0387          ADDL    R2,UCB$L_TT_OUTADR(R5)            ; UPDATE POINTER
  0120 C5     52  A2          038C          SUBW    R2,UCB$W_TT_OUTLEN(R5)            ; AND COUNT
              07  13          0391          BEQL    60$                               ; ALL DONE, NO NEED FOR BURST
       0800 8F    A8          0393          BISW    #TTY$M_TANK_BURST,-               ; SIGNAL BURST ACTIVE
       0108 C5                0397                  UCB$W_TT_HOLD(R5)
                              039A   60$:
          08  52  E9          039A          BLBC    R2,70$                            ; EVEN TRANSFER
     06 A0    83  90          039D          MOVB    (R3)+,6(R0)                       ; OUTPUT ODD BYTE
              52  D7          03A1          DECL    R2                                ; UPDATE COUNT
              0F  13          03A3          BEQL    80$                               ; DONE
                              03A5   70$:
  52     52   FF 8F   78      03A5          ASHL    #-1,R2,R2                         ; CONVERT TO WORD COUNT
                              03AA   75$:
     06 A0    83  B0          03AA          MOVW    (R3)+,6(R0)
```

```
                        03AE   812          DELAY                    ; TO RELEASE THE UNIBUS
          F6 52   F5    03B1   813          SOBGTR  R2,75$           ; LOOP TILL DONE
                        03B4   814
                  05    03B4   815  80$:    RSB                      ; RETURN TO CALLER
                        03B5   816
                        03B5   817  90$:
             13   13    03B5   818          BEQL    100$             ; SKIP IF NONE
                        03B7   819          GETIND  0                ; SELECT OUTPUT SILO
       06 A0   53   90  03C6   820          MOVB    R3,6(R0)         ; OUTPUT CHARACTER
                        03CA   821  100$:
                  05    03CA   822          RSB
                        03CB   823          .DISABLE        LSB
                        03CB   824
```

```
                    03CB           .SBTTL  PORT DMA ROUTINES
                    03CB  ;++
                    03CB  ; DMA_START      -       INITIATE DMA OUTPUT
                    03CB  ;
                    03CB  ; FUNCTIONAL DESCRIPTION:
                    03CB  ;
                    03CB  ; THESE ROUTINES ARE CALLED BY THE PORT INPUT INTERRUPT, OUTPUT
                    03CB  ; INTERRUPT, AND STARTIO TO INITIATE NEW DAM OUTPUT.
                    03CB  ; THEY HANDLE ALLOCATION AND LOADING OF MAP REGISTERS
                    03CB  ; TO HANDLE DMA OUTPUT. MAP REGISTERS ARE ALLOCATED IN PAIRS
                    03CB  ; TO ALLOW OUTPUT BURSTS UP TO 512 BYTES. TRANSFERS LARGER THAN
                    03CB  ; THAT ARE DONE IN SEGMENTS. IF INSUFFICIENT MAPS ARE AVAILABLE,
                    03CB  ; THE TRANSFER IS DONE SILO MODE.
                    03CB  ;
                    03CB  ;
                    03CB  ; INPUTS:
                    03CB  ;
                    03CB  ;        R5 = UCB ADDRESS
                    03CB  ;
                    03CB  ; OUTPUTS:
                    03CB  ;
                    03CB  ;        R5 = UCB ADDRESS
                    03CB  ;
                    03CB  ;        R0 PRESERVED
                    03CB  ;        R1,R2,R3,R4 DESTROYED
                    03CB  ;--
                    03CB  DMA_START:
     1000 8F    A8  03CB           BISW    #TTY$M_TANK_DMA,-         ; SHOW DMA MODE ACTIVE
     0108 C5        03CF                   UCB$W_TT_HOLD(R5)
       01       8A  03D2           BICB    #TTY$M_TP_ABORT,-         ; RESET ANY OLD ABORT REQUESTS
     0130 C5        03D4                   UCB$B_TP_STAT(R5)
                    03D7
                    03D7  ;        CHECK IF UNIT HAS PERMANENT MAP REGISTERS
                    03D7
       03       E1  03D7           BBC     #TTY$V_PC_PRMMAP,-        ; SKIP IF NOT AUTHORIZED FOR PERM MAPS
  06 0122 C5        03D9                   UCB$W_TT_PRTCTL(R5),5$
       04       E0  03DD           BBS     #TTY$V_PC_MAPAVL,-        ; SKIP FORK IF MAPS ALLOCATED ALREADY
  49 0122 C5        03DF                   UCB$W_TT_PRTCTL(R5),DMA_CONTINUE
                    03E3  5$:
       02       88  03E3           BISB    #TTY$M_TP_ALLOC,-         ; SHOW ALLOC FORK ACTIVE
     0130 C5        03E5                   UCB$B_TP_STAT(R5)
51   0114 C5    D0  03E8           MOVL    UCB$L_TT_CLASS(R5),R1     ; GET CLASS VECTOR ADDRESS
       1C B1    16  03ED           JSB     @CLASS_FORK(R1)           ; FORK TO FIPL FOR MAP
                    03F0                                             ; REGISTER ALLOCATION
                05  03F0           RSB                               ; RETURN TO CALLER WITH
                    03F1                                             ; "INT" LEFT ON TO INTERLOCK
                    03F1                                             ; OUTPUT. FORK ROUTINE WILL
                    03F1                                             ; RESUME AT DMA_ALLOC.
                    03F1  DMA_ALLOC:
                    03F1
                    03F1
                    03F1  5$:
     53   02    9A  03F1           MOVZBL  #2,R3                     ; REQUEST 2 MAP REGISTERS
  00000000'GF   16  03F4           JSB     G^IOC$ALOUBAMAPN          ;
                    03FA
                    03FA           SETIPL  UCB$B_DIPL(R5)            ; INTERLOCK TO DEVICE IPL
       10       A8  03FE           BISW    #TTY$M_PC_MAPAVL,-        ; SHOW MAP ALLOCATED
```

```
             0122 C5        0400                          UCB$W_TT_PRTCTL(R5)
                02    8A    0403                 BICB     #TTY$M_TP_ALLOC,-               ; SHOW ALLOC FORK DONE
             0130 C5        0405                          UCB$B_TP_STAT(R5)
             17   50  E8    0408                 BLBS     R0,20$                          ; SUCCESS
                            040B
        50    24 A5   D0    040B                 MOVL     UCB$L_CRB(R5),R0                ; GET CRB OF UNIT
        50    2C B0   D0    040F                 MOVL     @CRB$L_INTD+VEC$L_IDB(R0),R0; GET CSR
           1000 8F    AA    0413                 BICW     #TTY$M_TANK_DMA,-               ; RESET DMA MODE
           0108 C5          0417                          UCB$W_TT_HOLD(R5)
      52   0120 C5    3C    041A                 MOVZWL   UCB$W_TT_OUTLEN(R5),R2          ; RESTORE OUTPUT LENGTH
              FF49    31    041F                 BRW      SILO_OUTPUT                     ; USE SILO FOR OUTPUT
                            0422
                            0422        20$:
        50    24 A5   D0    0422                 MOVL     UCB$L_CRB(R5),R0                ; GET CRB ADDRESS
              34 A0   D0    0426                 MOVL     CRB$L_INTD+VEC$W_MAPREG(R0),-
           012C C5          0429                          UCB$L_TP_MAP(R5)                ; SAVE  MAP FIELD IN UCB
                            042C
                            042C
                            042C        DMA_CONTINUE:
                00    E0    042C                 BBS      #TTY$V_TP_ABORT,-               ; BRANCH IF DMA TO BE ABORTED
         0C 0130 C5         042E                          UCB$B_TP_STAT(R5),2$
                            0432
      53   011C C5    D0    0432                 MOVL     UCB$L_TT_OUTADR(R5),R3          ; GET ADDRESS OF NEXT STRING
      52   0120 C5    3C    0437                 MOVZWL   UCB$W_TT_OUTLEN(R5),R2          ; LENGTH OF OUTPUT
                03    12    043C                 BNEQ     4$                              ; SKIP IF MORE TO DO
              00B2    31    043E        2$:      BRW      DMA_DONE                        ; BRANCH IF TRANSFER IS DONE
                            0441        4$:
                50    DD    0441                 PUSHL    R0                              ; SAVE INPUT VOLITAL REGISTER "CSR"
                            0443
   00000200 8F    52  D1    0443                 CMPL     R2,#512                         ; NEXT BURST TOO LONG FOR MAPS?
                05    15    044A                 BLEQ     5$                              ; NO
      52   0200 8F    3C    044C                 MOVZWL   #512,R2
                            0451
                            0451
      011C C5     52  C0    0451        5$:      ADDL     R2,UCB$L_TT_OUTADR(R5)          ; UPDATE CHARACTER POINTER FOR NEXT TIME
      0120 C5     52  A2    0456                 SUBW     R2,UCB$W_TT_OUTLEN(R5)          ; UPDATE COUNT FOR NEXT TIME
                            045B
                            045B                 TIMSET   R2,R1,LOCKOUTPUT                ;RECOMPUTE TIMEOUT VALUE FOR THIS
                            047C                                                          ;PORTION OF THE DMA BURST
                            047C
                            047C
                            047C        ;                 R3 - STRING ADDRESS
                            047C        ;                 R2 - LENGTH
                            047C        ;                 R5 - UCB
                            047C
                24    BB    047C                 PUSHR    #^M<R2,R5>
        50    24 A5   D0    047E                 MOVL     UCB$L_CRB(R5),R0                ; GET CRB ADDRESS
        51    38 B0   D0    0482                 MOVL     @CRB$L_INTD+VEC$L_ADP(R0),R1        ; CONFIG REGISTER
           0F     00  EF    0486                 EXTZV    #VEC$V_MAPREG,#VEC$S_MAPREG,-
        50   012C C5        0489                          UCB$L_TP_MAP(R5),R0             ; GET STARTING MAP REGISTER
   51     0800 C140   DE    048D                 MOVAL    UBA$L_MAP(R1)[R0],R1            ; GET 1ST MAP REGISTER ADDRESS
                            0493
54    53   80000000 8F  CB  0493                 BICL3    #^X80000000,R3,R4               ; CALC SVAPTE OF BUFFER
      54    54   F7 8F  78  049B                 ASHL     #-9,R4,R4                       ; ISOLATE PAGE
      55    00000000'GF D0  04A0                 MOVL     G^MMG$GL_SPTBASE,R5             ; GETS SVAPTE OF BUFFER
           54    6544   DE  04A7                 MOVAL    (R5)[R4],R4                     ; INTO R4
      53   FFFFFE00 8F  CA  04AB                 BICL     #^C^X1FF,R3                     ; COMPUTE BYTE OFFSET IN PAGE
```

```
                          04B2
                          04B2     ;       LOAD MAP REGISTERS
                          04B2     ;               R0 - MAP REGISTER NUMBER
                          04B2     ;               R1 - ADDRESS OF FIRST MAP REGISTER
                          04B2     ;               R2 - BUFFER LENGTH
                          04B2     ;               R3 - BYTE OFFSET IN PAGE
                          04B2     ;               R4 - SVAPTE OF BUFFER
                          04B2
            52    02   9A 04B2             MOVZBL  #2,R2
            55    84   D0 04B5     10$:    MOVL    (R4)+,R5                        ; GET CONTENTS OF NEXT PTE
                          04B8
                          04B8     ;       THIS CODE ASSUMES THAT DMA IS FROM NONPAGED POOL
                          04B8
 55   0B   15 00000400 8F F0 04B8          INSV    #^X400,#21,#11,R5               ; SET VALID BIT, DATA PATH 0
            81    55   D0 04C1             MOVL    R5,(R1)+                        ; LOAD INTO MAP REGISTER
               EE 52   F5 04C4             SOBGTR  R2,10$
                          04C7
                  24   BA 04C7             POPR    #^M<R2,R5>                      ; RESTORE LENGTH,WRITE BUFFER, UCB
                          04C9
      53   09   09   50   F0 04C9          INSV    R0,#9,#9,R3                     ; COMPUTE UNIBUS ADDRESS
      50   53   02   10   EF 04CE          EXTZV   #16,#2,R3,R0                    ; GET HIGH 2 UNIBUS ADDRESS BITS
      52   02   0E   50   F0 04D3          INSV    R0,#14,#2,R2                    ; MERGE WITH BYTE COUNT
                          04D8
                          04D8             GETIND  2                               ; SELECT DMA INDIRECT REGISTERS
         06 A0    53   B0 04E7             MOVW    R3,6(R0)                        ; LOAD ADDRESS (INDIRECT REGISTER
                          04EB                                                     ; AUTO INCREMTS TO COUNT)
         06 A0    52   B0 04EB             MOVW    R2,6(R0)                        ; LOAD COUNT/ INIT TRANSFER
                          04EF
               50 8ED0    04EF             POPL    R0                              ; RESTORE CSR ADDRESS
                    05    04F2             RSB                                     ; RETURN TO CALLER
                          04F3                                                     ; FORK DISPATCHER, ISR, OR STARTIO
                          04F3
                          04F3     DMA_DONE:                                       ; DMA COMPLETION
                          04F3
                          04F3
               03      E0 04F3             BBS     #TTY$V_PC_PRMMAP,-              ; SKIP FORK IF MAPS PERMANENT
         2E 0122 C5       04F5                     UCB$W_TT_PRTCTL(R5),DMA_POST
               04      88 04F9             BISB    #TTY$M_TP_DLLOC,-               ; SHOW DEALLOC FORK ACTIVE
            0130 C5       04FB                     UCB$B_TP_STAT(R5)
      51    0114 C5    D0 04FE             MOVL    UCB$L_TT_CLASS(R5),R1           ; GET CLASS VECTOR ADDRESS
              1C B1    16 0503             JSB     @CLASS_FORK(R1)                 ; SCHEDULE FORK TO FIPL FOR MAP
                          0506                                                     ; REGISTER DEALLOCATION
                       05 0506             RSB                                     ; RETURN TO CALLER, FORK WILL RESUME
                          0507                                                     ; AT DMA_DEALLOC
                          0507
                          0507     DMA_DEALLOC:
      50   24 A5       D0 0507             MOVL    UCB$L_CRB(R5),R0                ; GET CRB ADDRESS
            012C C5    D0 050B             MOVL    UCB$L_TP_MAP(R5),-
              34 A0       050F                     CRB$L_INTD+VEC$W_MAPREG(R0); RESTORE MAP FIELD IN CRB
                06     13 0511             BEQL    5$                              ; SKIP IF NONE
      00000000'GF      16 0513             JSB     G^IOC$RELMAPREG                 ; RELEASE MAP REGISTERS
                          0519
                          0519     5$:     SETIPL  UCB$B_DIPL(R5)                  ; INTERLOCK TO DEVICE IPL
               10      AA 051D             BICW    #TTY$M_PC_MAPAVL,-              ; SHOW MAP ALLOCATED
            0122 C5       051F                     UCB$W_TT_PRTCTL(R5)
               04      8A 0522             BICB    #TTY$M_TP_DLLOC,-               ; SHOW DEALLOC FORK DONE
            0130 C5       0524                     UCB$B_TP_STAT(R5)
```

```
                         0527
                         0527  DMA_POST:
             01    8A    0527          BICB    #TTY$M_TP_ABORT,-        ; RESET ABORT REQUEST
        0130 C5          0529                  UCB$B_TP_STAT(R5)
        1000 8F    AA    052C          BICW    #TTY$M_TANK_DMA,-        ; RESET DMA MODE
        0108 C5          0530                  UCB$W_TT_HOLD(R5)
                         0533
                         0533  ;               CALL GETNEXT TO CONTINUE PROCESSING
                         0533  ;
             03    8A    0533          BICB    #UCB$M_TIM!UCB$M_INT,-   ;CLEAR TIMEOUT AND INT EXPECTED
          64 A5          0535                  UCB$W_STS(R5)
        010C D5    16    0537          JSB     @UCB$L_TT_GETNXT(R5)     ; GET NEXT BURST
           FEOF    31    053B          BRW     YC$STARTIO               ; AND PROCEED
                         053E
                         053E  YC$FORK:                                 ;
                         053E          SAVIPL                           ; SAVE CURRENT IPL ON THE STACK
    00000559'EF    DF    0541          PUSHAL  20$                      ; BUILD RETURN ADDRESS ON STACK
             01    E1    0547          BBC     #TTY$V_TP_ALLOC,-        ; SKIP IF NOT ALLOCATE FORK
     03 0130 C5          0549                  UCB$B_TP_STAT(R5),10$
           FEA1    31    054D          BRW     DMA_ALLOC                ; RESUME AT ALLOCATE CODE THREAD
                         0550  10$:
             02    E1    0550          BBC     #TTY$V_TP_DLLOC,-        ; CHECK FOR DEALLOCATE
     03 0130 C5          0552                  UCB$B_TP_STAT(R5),20$
           FFAE    31    0556          BRW     DMA_DEALLOC
                         0559  20$:
                         0559          ENBINT                           ; RESTORE SAVED FORK IPL FROM STACK
                         055C
             05          055C          RSB
                         055D
                         055D
```

```
                              055D          .SBTTL  PORT ROUTINES  STOP,RESUME,XON,XOFF
                              055D  ;++
                              055D  ; YC$XOFF -       SEND XOFF
                              055D  ; YC$XON -        SEND XON
                              055D  ; YC$STOP -       STOP OUTPUT
                              055D  ; YC$ABORT -      ABORT CURRENT OUTPUT
                              055D  ; YC$RESUME -     RESUME STOPPED OUTPUT
                              055D  ;
                              055D  ; FUNCTIONAL DESCRIPTION:
                              055D  ;
                              055D  ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                              055D  ; CONTROL OUTPUT ON THE PORT
                              055D  ;
                              055D  ; INPUTS:
                              055D  ;
                              055D  ;         R5 = UCB ADDRESS
                              055D  ;
                              055D  ; OUTPUTS:
                              055D  ;
                              055D  ;         R5 = UCB ADDRESS
                              055D  ;--
                              055D  ;
                              055D  ; SCHEDULE XOFF OR XON TO BE SEND
                              055D  ;
                              055D  ; INPUTS:
                              055D  ;
                              055D  ;         R3 - CONTAINS THE CHARACTER TO SEND AS FLOW CONTROL.
                              055D  ;
                              055D  YC$XOFF:
                              055D  YC$XON:
                   03     BB  055D          PUSHR   #^M<R0,R1>
                              055F          GETIND  1,R1
06 A0    51    8000 8F    A9  0572          BISW3   #DMFLCT$M_PREMPT,R1,6(R0)       ; PREMPT ANY CURRENT OUTPUT
60    54 A5    4040 8F    A9  0579          BISW3   #^X4040,UCB$W_UNIT(R5),(R0)     ; SELECT TRANSMIT SILO
         06 A0     53     90  0580          MOVB    R3,6(R0)                        ; OUTPUT CHARACTER
                   03     BA  0584          POPR    #^M<R0,R1>
                   05         0586          RSB
                              0587  ;
                              0587  ; STOP PORT OUTPUT
                              0587  ;
                              0587  YC$STOP:
                   03     BB  0587          PUSHR   #^M<R0,R1>
                   03     8A  0589          BICB    #UCB$M_INT!UCB$M_TIM,-          ; RESET TIMER AND OUTPUT ACTIVE
              64 A5           058B                  UCB$W_STS(R5)
                              058D          GETIND  1                               ;
   06 0122 C5      05     E1  059C          BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),10$; AUTOXON XOFF AVAILABLE ON
                              05A2                                                  ; YES THEN IS
   04 0122 C5      06     E0  05A2          BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),20$; AUTOXON XOFF ENABLED
         06 A0     01     AA  05A8  10$:    BICW2   #DMFLCT$M_XMIT,6(R0)            ; RESET TRANSMIT ENABLE
                              05AC  20$:
                   03     BA  05AC          POPR    #^M<R0,R1>
                   05         05AE          RSB
                              05AF  ;
                              05AF  ; ABORT ANY CURRENT PORT OUTPUT ACTIVITY
                              05AF  ;
                              05AF  YC$ABORT:
```

```
                03   BB  05AF               PUSHR   #^M<R0,R1>
       0108 C5  0B   E5  05B1               BBCC    #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),-    ; RESET BURST ACTIVE
                00       05B6                       10$                                      ;
                         05B7      10$:
                         05B7               GETIND  1
          06 A0 01   A8  05C6               BISW2   #<DMFLCT$M_XMIT>,6(R0)                   ; SET XMIT
       09 64 A5 01   E1  05CA               BBC     #UCB$V_INT,UCB$W_STS(R5),15$             ; SKIP IF NOT BUSY.
          06 A0 10   A8  05CF               BISW2   #<DMFLCT$M_FLUSH>,6(R0)                  ; FLUSH OUTPUT
        0130 C5 01   88  05D3               BISB    #TTY$M_TP_ABORT,UCB$B_TP_STAT(R5)        ; REQUEST DMA ABORT
                         05D8      15$:
                03   BA  05D8               POPR    #^M<R0,R1>
                     05  05DA               RSB
                         05DB
                         05DB      ;
                         05DB      ; RESUME PREVIOUSLY STOPPED PORT OUTPUT
                         05DB      ;
                         05DB      YC$RESUME:
                0F   BB  05DB               PUSHR   #^M<R0,R1,R2,R3>
                         05DD               GETIND  1
                         05EC
  06 0122 C5    05   E1  05EC               BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),10$; AUTOXON XOFF AVAILABLE ON
                         05F2                                                       ; NO THEN RESUME TRANSMITION
  04 0122 C5    06   E0  05F2               BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),20$; AUTOXON XOFF ENABLED
                         05F8                                                            ; YES THEN DON'T ENABLE THE
       06 A0    01   A8  05F8      10$:     BISW2   #DMFLCT$M_XMIT,6(R0)                     ; ENABLE TRANSMIT
    38 64 A5    01   E0  05FC      20$:     BBS     #UCB$V_INT,UCB$W_STS(R5),40$             ; SKIP IF OUTPUT ON
                0C   E0  0601               BBS     #TTY$V_TANK_DMA,-                        ; SPECIAL IF DMA ACTIVE
     35 0108    C5       0603                       UCB$W_TT_HOLD(R5),50$
  2C 0108 C5    0B   E1  0607               BBC     #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),40$ ; NO BURST IN PROGRESS
                         060D                                                            ; (RESET ANYWAY. WILL BE
                         060D                                                            ; SET IF NEEDED BY BURST_OUT
60    54 A5  4040 8F A9  060D               BISW3   #^X4040,UCB$W_UNIT(R5),(R0)              ; SELECT TRANSMIT SILO
         51    06 A0 9A  0614               MOVZBL  6(R0),R1                                 ; GET SILO DEPTH
                         0618               TIMSET  R1,R1,LOCKOUTPUT                         ; COMPUTE TIMEOUT AND
                         0639                                                            ; SET INTERUPT EXPECTED
                         0639
                         0639      40$:
                0F   BA  0639               POPR    #^M<R0,R1,R2,R3>
                     05  063B               RSB
                         063C
                         063C      50$:
60    54 A5  4058 8F A9  063C               BISW3   #^X4058,UCB$W_UNIT(R5),(R0)              ; GET CURRENT BYTE COUNT
         51    06 A0 3C  0643               MOVZWL  6(R0),R1
      51     C000 8F AA  0647               BICW    #^X0C000,R1                              ; MASK OFF ADDRESS BITS
                         064C               TIMSET  R1,R1,LOCKOUTPUT                         ; COMPUTE TIMEOUT AND
                         066D                                                            ; SET INTERUPT EXPTECTEDD
                CA   11  066D               BRB     40$
                         066F
```

```
                          066F          .SBTTL  OUTPUT INTERRUPT SERVICE
                          066F  ;++
                          066F  ; YC$INTOUT - DMF OUTPUT INTERRUPT SERVICE
                          066F  ;
                          066F  ; FUNCTIONAL DESCRIPTION:
                          066F  ;
                          066F  ; THIS ROUTINE IS ENTERED WHEN THE DMF FINDS A LINE ENABLED
                          066F  ; AND AN EMPTY UART. THE CORRESPONDING UCB IS FOUND AND
                          066F  ; ANY OUTSTANDING PORT OUTPUT IS DONE. WHEN ALL OUTSTANDING PORT
                          066F  ; OUTPUT IS COMPLETED, THE CLASS DRIVER IS CALLED TO RETURN THE NEXT
                          066F  ; CHARACTER OR STRING TO BE OUTPUT. IF NO MORE OUTPUT IS FOUND, THEN
                          066F  ; THE LINE IS DISBALED.
                          066F  ;
                          066F  ; INPUTS:
                          066F  ;
                          066F  ;       SP(00) = ADDRESS OF THE IDB
                          066F  ;
                          066F  ; IMPLICIT INPUTS:
                          066F  ;
                          066F  ;       R0,R1,R2,R3,R4,R5 SAVED ON THE STACK.
                          066F  ;
                          066F  ; OUTPUTS:
                          066F  ;
                          066F  ;       THE INTERRUPT IS DISMISSED.
                          066F  ;
                          066F  ;--
                          066F  YC_OUT_EXIT:                                  ; EXIT OUTPUT INTERRUPT
           5E    04   C0  066F          ADDL    #4,SP                         ; REMOVE IDB ADDRESS
           50    8E   7D  0672          MOVQ    (SP)+,R0                      ; RESTORE REGISTERS
           52    8E   7D  0675          MOVQ    (SP)+,R2                      ;
           54    8E   7D  0678          MOVQ    (SP)+,R4                      ;
                      02  067B          REI                                   ; DISMISS INTERRUPT
                          067C
                          067C  YC$INTOUT::                                   ; DMF OUTPUT INTERRUPT SERVICE
                          067C
                          067C  YC_OUT_LOOP:
           54  00 BE  D0  067C          MOVL    @(SP),R4                      ; GET THE IDB ADDRESS
           50    64   D0  0680          MOVL    (R4),R0                       ; GET THE CSR ADDRESS
                          0683  ;
                          0683  ; GET THE LINE INFO FROM THE CSR
                          0683  ;
                          0683
           52    60   B0  0683          MOVW    (R0),R2                       ; GET THE CSR VALUE
                 E7   18  0686          BGEQ    YC_OUT_EXIT                   ; NO MORE LINES
      51    52   F8 8F  78  0688        ASHL    #-8,R2,R1                     ; GET THE LINE NUMBER
    51   FFFFFFF8 8F  CA  068D          BICL    #^C<7>,R1                     ;
         55   18 A441  D0  0694         MOVL    IDB$L_UCBLST(R4)[R1],R5       ; GET THE UCB ADDRESS
                 E1   13  0699          BEQL    YC_OUT_LOOP                   ; IF EQL THEN DISMISS
                          069B  ;
                          069B  ; CHECK TO MAKE SURE NO DATA IS PENDING BEFORE ASKING FOR MORE
                          069B  ;
  60    54 A5    4040 8F  A9  069B      BISW3   #^X4040,UCB$W_UNIT(R5),(R0); GET THE SILO INDIRECT REGISTER
              06 A0   95  06A2          TSTB    6(R0)                         ; ANY DATA STILL IN THE SILO
                 27   12  06A5          BNEQ    40$                           ; YES THEN LET IT COMPLETE
                          06A7  ;
                          06A7  ;       CHECK FOR BURST OR DMA ACTIVE ON LINE
                          06A7  ;
```

```
                              06A7
             08    91         06A7          CMPB    #TTY$M_TANK_BURST@-8,-   ; ONLY BURST ACTIVE?
       0109  C5               06A9                  UCB$W_TT_HOLD+1(R5)
             2C    13         06AC          BEQL    YC_SILO                  ; YES, CONTINUE SILO OUTPUT
                              06AE
             10    93         06AE          BITB    #TTY$M_TANK_DMA@-8,-     ; DMA ACTIVE?
       0109  C5               06B0                  UCB$W_TT_HOLD+1(R5)
             7A    12         06B3          BNEQ    YC_DMA_INTERRUPT         ; YES, PROCESS IT.
                              06B5  ;
                              06B5  ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
                              06B5  ;
      64 A5  03    8A         06B5  10$:    BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
                              06B9  ;
                              06B9  ; CALL CLASS DRIVER FOR MORE OUTPUT
                              06B9  ;
       010C  D5    16         06B9          JSB     @UCB$L_TT_GETNXT(R5)     ; GET THE NEXT CHARACTER
             15    19         06BD          BLSS    YC_START_BURST           ; BURST SPECIFIED
             BB    13         06BF          BEQL    YC_OUT_LOOP              ; NONE
                              06C1  ;
                              06C1  ; OUTPUT A CHARACTER TO THE DMF
                              06C1  ;
                              06C1  20$:
60    54 A5  4040 8F   A9     06C1          BISW3   #^X4040,UCB$W_UNIT(R5),(R0)      ; SELECT OUTPUT SILO INDIRECT REGIST
      06 A0  53    90         06C8          MOVB    R3,6(R0)                 ; OUTPUT CHARACTER
             AE    11         06CC          BRB     YC_OUT_LOOP
                              06CE
       F966  CF    D6         06CE  40$:    INCL    YC$L_SIL_ERROR
             A8    11         06D2          BRB     YC_OUT_LOOP
```

```
                                 06D4
                                 06D4  YC_START_BURST:
                FC7F CF     16   06D4            JSB     BURST_OUTPUT                  ; START OUTPUT SILO OR DMA
                     A2     11   06D8            BRB     YC_OUT_LOOP
                                 06DA
                                 06DA  ;
                                 06DA  ;  CONTINUE SILO OUTPUT
                                 06DA  ;
                                 06DA  YC_SILO:
  60    54 A5   4040 8F     A9   06DA            BISW3   #^X4040,UCB$W_UNIT(R5),(R0)      ; SELECT OUTPUT SILO INDIRECT REGIST
           51     06 A0     9A   06E1            MOVZBL  6(R0),R1                      ; GET SILO DEPTH
        51     20    51     C3   06E5            SUBL3   R1,#32,R1                          ; CONVERT TO NUMBER SLOTS AVAILABLE
                                 06E9
        52     0120 C5      3C   06E9            MOVZWL  UCB$W_TT_OUTLEN(R5),R2        ; GET CURRENT LENGTH
        53     011C C5      D0   06EE            MOVL    UCB$L_TT_OUTADR(R5),R3        ; GET CURRENT ADDRESS
        51          52      B1   06F3            CMPW    R2,R1                         ; BURST LARGER THAN SILO?
                    03      1B   06F6            BLEQU   50$                           ; NO
        52          51      9A   06F8            MOVZBL  R1,R2                         ; MAXIMUM
                                 06FB  50$:
           011C C5  52      C0   06FB            ADDL    R2,UCB$L_TT_OUTADR(R5)        ; UPDATE POINTER
           0120 C5  52      A2   0700            SUBW    R2,UCB$W_TT_OUTLEN(R5)        ; AND COUNT
                    07      12   0705            BNEQ    60$                           ; NOT DONE
                0800 8F     AA   0707            BICW    #TTY$M_TANK_BURST,-           ; RESET BURST ACTIVE
                0108 C5          070B                    UCB$W_TT_HOLD(R5)
                    52      D5   070E  60$:      TSTL    R2                            ; ANY ROOM AT ALL
                    1A      13   0710            BEQL    80$                           ; NO THEN EXIT
                                 0712
                 08 52      E9   0712            BLBC    R2,70$                        ; EVEN TRANSFER
              06 A0 83      90   0715            MOVB    (R3)+,6(R0)                   ; OUTPUT ODD BYTE
                    52      D7   0719            DECL    R2                            ; UPDATE COUNT
                    0F      13   071B            BEQL    80$                           ; DONE
                                 071D  70$:
  52       52    FF 8F      78   071D            ASHL    #-1,R2,R2                     ; CONVERT TO WORD COUNT
                                 0722  75$:
           06 A0    83      B0   0722            MOVW    (R3)+,6(R0)
                                 0726            DELAY                                 ; TO RELEASE THE UNIBUS
                 F6 52      F5   0729            SOBGTR  R2,75$                        ; LOOP TILL DONE
                                 072C
                  FF4D      31   072C  80$:      BRW     YC_OUT_LOOP
                                 072F
                                 072F
                                 072F  YC_DMA_INTERRUPT:
                FF49 CF     DF   072F            PUSHAL  YC_OUT_LOOP                   ; BUILD RETURN ADDRESS ON STACK
           04 52   0C       E1   0733            BBC     #DMFCSR$V_DMAERR,R2,30$       ; CHECK FOR A DMA ERROR
                F905 CF     D6   0737            INCL    YC$L_DMAXMT_ERROR             ; ERROR OCCURED INCREMENT COUNTS
                                 073B  30$:
                    06      93   073B            BITB    #TTY$M_TP_ALLOC!TTY$M_TP_DLLOC,-  ;CHECK FOR FORKS ACTIVE
                0130 C5          073D                    UCB$B_TP_STAT(R5)             ; AND IGNORE IF SO
                    21      12   0740            BNEQ    10$
        0130 C5     00      E0   0742            BBS     #TTY$V_TP_ABORT,UCB$B_TP_STAT(R5),-; ABORT ACTIVE DMA
                    18           0747                    20$
  60    54 A5   4058 8F     A9   0748            BISW3   #^X4058,UCB$W_UNIT(R5),(R0); GET CURRENT BYTE COUNT
           51     06 A0     3C   074F            MOVZWL  6(R0),R1
        51      C000 8F     AA   0753            BICW    #^XC000,R1                    ; MASK OFF ADDRESS BITS
                    06      13   0758            BEQL    20$                           ; DMA BYTE COUNT DONE.
                F8DE CF     D6   075A            INCL    YC$L_ERROR
                    03      11   075E            BRB     10$                           ; NO THEN CONTINUE
```

```
               0760  1277
               0760  1278 20$:
      FCC9  31 0760  1279         BRW     DMA_CONTINUE              ; OTHERWISE, CONTINUE THE DMA
               0763  1280
               0763  1281                                           ; IF THIS INTERRUPT WAS THE RESULT
               0763  1282                                           ; OF AN ABORT, THIS WILL BE HANDLED
               0763  1283                                           ; BY DMA_CONTINUE
               0763  1284 10$:
            05 0763  1285         RSB
               0764  1286
```

```
                              0764              .SBTTL  SET SPEED, PARITY PARAMETERS
                              0764
                              0764    ;++
                              0764    ; YC$SET_LINE - RESET SPEED, PARITY
                              0764    ;
                              0764    ; FUNCTIONAL DESCRIPTION:
                              0764    ;
                              0764    ; INPUTS:
                              0764    ;
                              0764    ;         R5 - UCB ADDRESS
                              0764    ;
                              0764    ; OUTPUTS:
                              0764    ;
                              0764    ;         R4 USED
                              0764    ;--
                              0764
                              0764    YC$SET_LINE:
                   53    DD   0764            PUSHL   R3
           54   24 A5    D0   0766            MOVL    UCB$L_CRB(R5),R4               ; ADDRESS CRB
           54   2C B4    D0   076A            MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4   ; GET THE CSR ADDRESS VIA CRB
   64    54 A5   4048 8F    A9   076E            BISW3   #^X4048,UCB$W_UNIT(R5),(R4)    ; GET THE RIGHT INDIRECT REGISTER
      10 0122 C5   05    E1   0775            BBC     #TTY$V_PC_XOFAVL,UCB$W_TT_PRTCTL(R5),4$; AUTOXON XOFF AVAILABLE ON T
                              077B                                                   ; YES THEN IS
      06 0122 C5   06    E0   077B            BBS     #TTY$V_PC_XOFENA,UCB$W_TT_PRTCTL(R5),2$; AUTOXON XOFF ENABLED
           06 A4   02    AA   0781            BICW    #DMFLCT$M_AUTOX,6(R4)          ; NO THEN CLEAR THE AUTOXOFF ENABLE
                   04    11   0785            BRB     4$                             ; AND CONTINUE
           06 A4   02    A8   0787    2$:     BISW    #DMFLCT$M_AUTOX,6(R4)          ; ENABLED THEN SET AUTOXOFF
                              078B    4$:
                   7E    D4   078B            CLRL    -(SP)                          ; RESET A TEMPORARY LOCATION
                              078D    ;
                              078D    ;         SET UP LINE SPEED AND PARITY
                              078D    ;
                              078D
           00F5 C5   95       078D            TSTB    UCB$W_TT_SPEED+1(R5)   ; RECEIVE SPEED SPECIFIED?
                07   12       0791            BNEQ    5$                     ; YES
           00F4 C5   90       0793            MOVB    UCB$W_TT_SPEED(R5),-   ; NO, SO USE TRANSMITTER SPEED
           00F5 C5            0797                    UCB$W_TT_SPEED+1(R5)
   53    00F4 C5   01    83   079A    5$:     SUBB3   #1,UCB$W_TT_SPEED(R5),R3        ; ADJUST TRANSMIT SPEED
   6E    04   0C   53    F0   07A0            INSV    R3,#DMFLPR$V_TSPEED,#4,(SP)     ; SET TRANSMIT SPEED
   53    00F5 C5   01    83   07A5            SUBB3   #1,UCB$W_TT_SPEED+1(R5),R3      ; ADJUST RECEIVER SPEED
   6E    04   08   53    F0   07AB            INSV    R3,#DMFLPR$V_RSPEED,#4,(SP)     ; SET RECEIVE SPEED
                              07B0
                              07B0    ;         NONSTANDARD PARITY/STOP LAYOUT
                              07B0
 53  00F8 C5   02   03   EE   07B0            EXTV    #UCB$V_TT_LEN,#2,UCB$B_TT_PARITY(R5),R3         ; GET CHAR SIZE
     6E   02   03   53   F0   07B7            INSV    R3,#DMFLPR$V_SIZE,#2,(SP)                       ; SET IT
 53  00F8 C5   02   06   EE   07BC            EXTV    #UCB$V_TT_PARTY,#2,UCB$B_TT_PARITY(R5),R3       ; GET PARITY/ODD
     6E   02   05   53   F0   07C3            INSV    R3,#DMFLPR$V_PARITY,#2,(SP)
          6E   40 8F     8C   07C8            XORB    #DMFLPR$M_ODD,(SP)                              ; INVERT PARITY BIT
 53  00F8 C5   01   05   EE   07CC            EXTV    #UCB$V_TT_STOP,#1,UCB$B_TT_PARITY(R5),R3        ; GET STOP
     6E   01   07   53   F0   07D3            INSV    R3,#DMFLPR$V_STOP,#1,(SP)
                              07D8
              6E   07    AA   07D8            BICW    #^X0007,(SP)           ; CLEAR SPECIAL FIELDS
           6E   54 A5    A8   07DB            BISW    UCB$W_UNIT(R5),(SP)    ; SET LINE NUMBER
           02 A4   8E    F7   07DF            CVTLW   (SP)+,2(R4)            ; INSERT AS LINE PARAMETER
                53 8ED0       07E3            POPL    R3
                     05       07E6            RSB
```

```
07E7
07E7
07E7
07E7       YC$END:                          ; End of driver
07E7
07E7               .END
```

```
$$$                     = 00000020 R     02     DMFLCT$M_PREMPT          = 00008000
$$OP                    = 00000002              DMFLCT$M_RCV             = 00000004
AT$_UBA                 = 00000001              DMFLCT$M_XMIT            = 00000001
BIT...                  = 00000010              DMFLCT$S_AUTOX           = 00000001
BURST_OUTPUT              00000357 R     03     DMFLCT$S_BREAK           = 00000001
CLASS_DDT               = 00000010              DMFLCT$S_DS_ENABLE       = 00000001
CLASS_DS_TRAN           = 0000000C              DMFLCT$S_FLUSH           = 00000001
CLASS_FORK              = 0000001C              DMFLCT$S_MAINT           = 00000002
CLASS_GETNXT            = 00000000              DMFLCT$S_PREMPT          = 00000001
CLASS_POWERFAIL         = 00000020              DMFLCT$S_RCV             = 00000001
CLASS_PUTNXT            = 00000004              DMFLCT$S_XMIT            = 00000001
CLASS_READERROR         = 00000014              DMFLCT$V_AUTOX           = 00000001
CLASS_SETUP_UCB         = 00000008              DMFLCT$V_BREAK           = 00000003
CRB$B_TT_TYPE           = 0000000B              DMFLCT$V_DS_ENABLE       = 00000005
CRB$L_INTD              = 00000024              DMFLCT$V_FLUSH           = 00000004
CRB$L_INTD2             = 00000048              DMFLCT$V_MAINT           = 00000006
DC$_TERM                = 00000042              DMFLCT$V_PREMPT          = 0000000F
DDB$L_DDT               = 0000000C              DMFLCT$V_RCV             = 00000002
DEV$M_AVL               = 00040000              DMFLCT$V_XMIT            = 00000000
DEV$M_CCL               = 00000002              DMFLPR$M_LINE            = 00000007
DEV$M_IDV               = 04000000              DMFLPR$M_ODD             = 00000040
DEV$M_NNM               = 00000200              DMFLPR$M_PARITY          = 00000020
DEV$M_ODV               = 08000000              DMFLPR$M_RSPEED          = 00000F00
DEV$M_REC               = 00000001              DMFLPR$M_SIZE            = 00000018
DEV$M_TRM               = 00000004              DMFLPR$M_STOP            = 00000080
DMA_ALLOC                 000003F1 R     03     DMFLPR$M_TSPEED          = 0000F000
DMA_CONTINUE              0000042C R     03     DMFLPR$S_LINE            = 00000003
DMA_DEALLOC               00000507 R     03     DMFLPR$S_ODD             = 00000001
DMA_DONE                  000004F3 R     03     DMFLPR$S_PARITY          = 00000001
DMA_POST                  00000527 R     03     DMFLPR$S_RSPEED          = 00000004
DMA_START                 000003CB R     03     DMFLPR$S_SIZE            = 00000002
DMFCSR$M_CLEAR          = 00000020              DMFLPR$S_STOP            = 00000001
DMFCSR$M_DMAERR         = 00001000              DMFLPR$S_TSPEED          = 00000004
DMFCSR$M_IADDR          = 0000001F              DMFLPR$V_LINE            = 00000000
DMFCSR$M_LINE           = 00000700              DMFLPR$V_ODD             = 00000006
DMFCSR$M_RCVINT         = 00000040              DMFLPR$V_PARITY          = 00000005
DMFCSR$M_SNDINT         = 00004000              DMFLPR$V_RSPEED          = 00000008
DMFCSR$M_SNDRDY         = 00008000              DMFLPR$V_SIZE            = 00000003
DMFCSR$S_CLEAR          = 00000001              DMFLPR$V_STOP            = 00000007
DMFCSR$S_DMAERR         = 00000001              DMFLPR$V_TSPEED          = 0000000C
DMFCSR$S_IADDR          = 00000005              DMFRCV$M_BUF             = 000000FF
DMFCSR$S_LINE           = 00000003              DMFRCV$M_DS_CHG          = 00000800
DMFCSR$S_RCVINT         = 00000001              DMFRCV$M_FRAMER          = 00002000
DMFCSR$S_SNDINT         = 00000001              DMFRCV$M_LINE            = 00000700
DMFCSR$S_SNDRDY         = 00000001              DMFRCV$M_OVERRUN         = 00004000
DMFCSR$V_CLEAR          = 00000005              DMFRCV$M_PARERR          = 00001000
DMFCSR$V_DMAERR         = 0000000C              DMFRCV$M_VALID           = 00008000
DMFCSR$V_IADDR          = 00000000              DMFRCV$S_BUF             = 00000008
DMFCSR$V_LINE           = 00000008              DMFRCV$S_DS_CHG          = 00000001
DMFCSR$V_RCVINT         = 00000006              DMFRCV$S_FRAMER          = 00000001
DMFCSR$V_SNDINT         = 0000000E              DMFRCV$S_LINE            = 00000003
DMFCSR$V_SNDRDY         = 0000000F              DMFRCV$S_OVERRUN         = 00000001
DMFLCT$M_AUTOX          = 00000002              DMFRCV$S_PARERR          = 00000001
DMFLCT$M_BREAK          = 00000008              DMFRCV$S_VALID           = 00000001
DMFLCT$M_DS_ENABLE      = 00000020              DMFRCV$V_BUF             = 00000000
DMFLCT$M_FLUSH          = 00000010              DMFRCV$V_DS_CHG          = 0000000B
DMFLCT$M_MAINT          = 000000C0              DMFRCV$V_FRAMER          = 0000000D
```

```
DMFRCV$V_LINE                   = 00000008                  SILO_OUTPUT                    0000036B R     03
DMFRCV$V_OVERRUN                = 0000000E                  SIZ...                       = 00000001
DMFRCV$V_PARERR                 = 0000000C                  SS$_NORMAL                   = 00000001
DMFRCV$V_VALID                  = 0000000F                  TT$M_DS_DTR                  = 00000002
DPT$C_LENGTH                    = 00000038                  TT$M_DS_RTS                  = 00000010
DPT$C_VERSION                   = 00000004                  TT$_UNKNOWN                  = 00000000
DPT$INITAB                        00000038 R     02         TTY$GB_DEFSPEED                ********   X  02
DPT$M_NOUNLOAD                  = 00000004                  TTY$GB_PARITY                  ********   X  02
DPT$REINITAB                      000000D3 R     02         TTY$GB_RSPEED                  ********   X  02
DPT$TAB                           00000000 R     02         TTY$GB_SILOTIME                ********   X  03
DPT$W_VECTOR                    = 0000001E                  TTY$GL_DEFCHAR                 ********   X  02
DT$_DMF32                       = 0000000A                  TTY$GL_DEFCHAR2                ********   X  02
DT$_DMZ32                       = 00000045                  TTY$GL_DPT                     ********   X  03
DYN$C_CRB                       = 00000005                  TTY$GL_OWNUIC                  ********   X  02
DYN$C_DDB                       = 00000006                  TTY$GW_DEFBUF                  ********   X  02
DYN$C_DPT                       = 0000001E                  TTY$GW_DMASIZE                 ********   X  03
DYN$C_ORB                       = 00000049                  TTY$GW_PROT                    ********   X  02
DYN$C_UCB                       = 00000010                  TTY$M_PC_DMAAVL              = 00000004
EXE$GL_ABSTIM                     ********   X   03         TTY$M_PC_MAPAVL              = 00000010
EXE$GL_TENUSEC                    ********   X   03         TTY$M_PC_XOFAVL              = 00000020
EXE$GL_UBDELAY                    ********   X   03         TTY$M_TANK_BURST             = 00000800
FUNCTAB_LEN                     = 00000000                  TTY$M_TANK_DMA               = 00001000
IDB$B_COMBO_CSR_OFFSET          = 0000000F                  TTY$M_TP_ABORT               = 00000001
IDB$B_COMBO_VECTOR_OFFSET       = 00000010                  TTY$M_TP_ALLOC               = 00000002
IDB$B_VECTOR                    = 0000000B                  TTY$M_TP_DLLOC               = 00000004
IDB$L_UCBLST                    = 00000018                  TTY$V_PC_DMAENA              = 00000001
IO$M_AUTXOF_DIS                 = 00004000                  TTY$V_PC_MAPAVL              = 00000004
IO$M_AUTXOF_ENA                 = 00002000                  TTY$V_PC_NOTIME              = 00000000
IO$M_LINE_OFF                   = 00000200                  TTY$V_PC_PRMMAP              = 00000003
IO$M_LINE_ON                    = 00000800                  TTY$V_PC_XOFAVL              = 00000005
IO$M_LOOP                       = 00000080                  TTY$V_PC_XOFENA              = 00000006
IO$M_LOOP_EXT                   = 00001000                  TTY$V_TANK_BURST             = 0000000B
IO$M_UNLOOP                     = 00000100                  TTY$V_TANK_DMA               = 0000000C
IOC$ALOUBAMAPN                    ********   X   03         TTY$V_TP_ABORT               = 00000000
IOC$MNTVER                        ********   X   03         TTY$V_TP_ALLOC               = 00000001
IOC$RELMAPREG                     ********   X   03         TTY$V_TP_DLLOC               = 00000002
IOC$RETURN                        ********   X   03         UBA$L_MAP                    = 00000800
MMG$GL_SPTBASE                    ********   X   03         UCB$B_DEVCLASS               = 00000040
MODEM$C_DATASET                 = 00000003                  UCB$B_DEVTYPE                = 00000041
MODEM$C_INIT                    = 00000000                  UCB$B_DIPL                   = 0000005E
ORB$B_FLAGS                     = 0000000B                  UCB$B_FIPL                   = 0000000B
ORB$L_OWNER                     = 00000000                  UCB$B_TP_STAT                = 00000130
ORB$M_PROT_16                   = 00000001                  UCB$B_TT_DEPARI              = 000000EC
ORB$W_PROT                      = 00000018                  UCB$B_TT_DETYPE              = 000000F0
PORT_ABORT                      = 00000020                  UCB$B_TT_DS_RCV              = 00000124
PORT_DS_SET                     = 0000000C                  UCB$B_TT_DS_TX               = 00000125
PORT_FORKRET                    = 00000034                  UCB$B_TT_MAINT               = 0000012A
PORT_LENGTH                     = 00000038                  UCB$B_TT_PARITY              = 000000F8
PORT_MAINT                      = 00000030                  UCB$C_TT_LENGTH              = 00000134
PORT_RESUME                     = 00000024                  UCB$L_CRB                    = 00000024
PORT_SET_LINE                   = 00000008                  UCB$L_DDB                    = 00000028
PORT_STARTIO                    = 00000000                  UCB$L_DDT                    = 00000088
PORT_STOP                       = 00000018                  UCB$L_DEVCHAR                = 00000038
PORT_VECTOR                       00000044 R     03         UCB$L_DEVCHAR2               = 0000003C
PORT_XOFF                       = 00000014                  UCB$L_DEVDEPEND              = 00000044
PORT_XON                        = 00000010                  UCB$L_DEVDEPND2              = 00000048
PR$_IPL                           ********   X   03         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$ABORT                  000005AF R     03
YC$CTRL_ERROR             00000106 R     03
YC$DDT                    00000000 RG    03
YC$DPT                    00000000 RG    02
YC$DS_SET                 00000281 R     03
YC$END                    000007E7 R     03
YC$FORK                   0000053E R     03
YC$INITIAL                00000081 RG    03
YC$INITLINE               00000107 RG    03
YC$INTINP                 000002B0 RG    03
YC$INTOUT                 0000067C RG    03
YC$L_DMAXMT_ERROR         00000040 R     03
YC$L_ERROR                0000003C R     03
YC$L_SIL_ERROR            00000038 R     03
YC$MAINT                  000001AC R     03
YC$NULL                   00000080 R     03
YC$RESUME                 000005DB R     03
YC$SET_LINE               00000764 R     03
YC$STARTIO                0000034D RG    03
YC$STOP                   00000587 R     03
YC$UNIT_ERROR             000001A7 R     03
YC$VEC                    00000044 R     03
YC$VECEND                 0000007C R     03
YC$XOFF                   0000055D R     03
YC$XON                    0000055D R     03
YC_DMA_INTERRUPT          0000072F R     03
YC_OUT_EXIT               0000066F R     03
YC_OUT_LOOP               0000067C R     03
YC_SILO                   000006DA R     03
YC_START_BURST            000006D4 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                   000007E7  (  2023.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

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

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     37    00:00:00.06    00:00:01.07
Command processing                116    00:00:00.36    00:00:01.78
Pass 1                            770    00:00:22.21    00:00:35.16
Symbol table sort                   0    00:00:03.21    00:00:03.25
Pass 2                            407    00:00:05.14    00:00:06.31
Symbol table output                 1    00:00:00.18    00:00:00.18
Psect synopsis output               0    00:00:00.01    00:00:00.02
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals             1333    00:00:31.18    00:00:47.77

The working set limit was 2400 pages.
214653 bytes (420 pages) of virtual memory were used to buffer the intermediate code.
There were 160 pages of symbol table space allocated to hold 2948 non-local and 78 local symbols.
1350 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
------------------                            --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          32
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       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$:YCDRIVER/UPDATE=(ENH$:YCDRIVER)+EXECML$/LIB
```

0404 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 CONFIDENTIAL AND PROPRIETARY

TTYSYNCH
LIS

TTYSTRSTP
LIS

TTYSUB
LIS

YCDRIVER
LIS

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

AUTHORIZE
CLD

ONFDEF
SDL

UAF

AUTHORIZE
MAP

CVTUAF
MAP

UAFREQ
SDL

YFDRIVER
LIS

CVTNAF
LIS

OAFDEF
SDL

CVTNAF
MAP

CVTUAF
LIS