H 6

```
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**DZVDRIVER


```
DDDDDDDD    ZZZZZZZZZZ  VV      VV  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DDDDDDDD    ZZZZZZZZZZ  VV      VV  DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DD      DD          ZZ  VV      VV  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD          ZZ  VV      VV  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD        ZZ    VV      VV  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD        ZZ    VV      VV  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD      ZZ      VV      VV  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
DD      DD      ZZ      VV      VV  DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
DD      DD    ZZ        VV      VV  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD    ZZ        VV      VV  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD  ZZ            VV  VV    DD      DD  RR    RR        II        VV  VV    EE          RR    RR      ....
DD      DD  ZZ            VV  VV    DD      DD  RR    RR        II        VV  VV    EE          RR    RR      ....
DDDDDDDD    ZZZZZZZZZZ      VV      DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR    ....
DDDDDDDD    ZZZZZZZZZZ      VV      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
```

DZV
V04

Table of contents

```
00000001  0000  DZV = 1              ; CREATE DZVDRIVER (Q-BUS VERSION)
          0000  .if df DZV
          0000          .TITLE  DZVDRIVER - Port Driver for DZV-11 support
          0000  .iff
          0000          .TITLE  DZDRIVER - Port Driver for DZ-11 support
          0000  .endc
          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  ;       DZ PORT DRIVER
          0000  ;       This module functions as a port driver for DZ11 and DZ32 terminal
          0000  ;       controllers. It contains hardware specific port level service
          0000  ;       routines.
          0000  ;
          0000  ; AUTHOR:
          0000  ;
          0000  ;       RICK SPITZ
          0000  ;
          0000  ; Revision history:
          0000  ;
          0000  ;       V03-028 MIR0480         Michael I.Rosenblum     8-Aug-1984
          0000  ;               Fix bugs found in testing the DZ-32 and reported in
          0000  ;               QAR 1126 (FT1).
          0000  ;
          0000  ;       V03-027 LMP0275         L. Mark Pilant,         12-Jul-1984  21:01
          0000  ;               Initialize the ACL info in the ORB to be a null descriptor
          0000  ;               list rather than an empty queue.  This avoids the overhead
```

```
;               of locking and unlocking the ACL mutex, only to find out
;               that the ACL was empty.
;
;       V03-026 EMD0090         Ellen M. Dusseault      30-Apr-1984
;               Add DEV$M_NNM characteristic to DEVCHAR2 so that these
;               devices will have the "node$" prefix.
;
;       V03-025 LMP0221         L. Mark Pilant,          7-Apr-1984  13:37
;               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;               ORB$W_PROT.
;
;       V03-024 JLV0321         Jake VanNoy              5-JAN-1984
;               Minor enhancements to DZV11 support. Comment out
;               DPT_STORE for parity, must fix a restriction.
;
;       V03-023 MIR0055         Michael I. Rosenblum            30-June-1983
;               Remove code from unit and controler-init routines
;               and make insert calls to class driver macros
;               Add DZV11 support.
;
;       V03-022 RKS0022         RICK SPITZ                      14-MAR-1983
;               ADD ENHANCEMENTS TO SUPPORT LOGICAL UCB.
;
;       V03-021 MIR0022         Michael I. Rosenblum            19-Jan-1982
;               Change references to UCB$B_ERTCNT to use UCB$W_TT_UNITBIT
;               to be more maintainable.
;               Replace old vector table with new vector table macro.
;               Remove references to UCB$L_DEVDEPEND and UCB$Q_TT_STATE
;               move these references into the class driver jacket routines
;
;       V03-020 MIR0021         Michael I. Rosenblum            17-Jan-1983
;               Fix DZ32 DZ_SET table entry.
;
;       V03-019 RKS0019         RICK SPITZ                      13-JAN-1983
;               Repair problem with port vector macro
;
;       V03-018 MIR0019         Michael I. Rosenblum            11-Jan-1982
;               Fix undefined symbol created by MIR0018.
;
;       V03-017 MIR0018         Michael I. Rosenblum            07-Jan-1983
;               Change the port vector table to use the $VEC macros
;
;       V03-016 MIR0017         Michael I. Rosenblum            05-Jan-1983
;               Add powerfail check in the Unit init routine to allow the
;               terminal class drier to take positive action on powerfail.
;               Change code to accept a byte value as returns from the
;               GETNXT and PUTNXT class services, this will removes this
;               information from the condition codes.
;
;       V03-015 MIR0016         Michael I. Rosenblum            29-Dec-1982
;               Replace time calculation code with TIMSET macro call
;
;       V03-014 MIR0015         Michael I. Rosenblum            20-Dec-1982
;               Remove entry point to reflect the redefinition of the PORT_DISCONNEC
;               entry point.
;
;       V03-013 MIR0014         Michael I. Rosenblum            17-Dec-1982
```

```
0000  ;                     Remove code to calculate flow control characters from
0000  ;                     port XON and XOFF routines and move that code into
0000  ;                     the class driver.
0000  ;
0000  ;             V03-012 RKS0012         RICK SPITZ              16-SEP-1982
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 for modem lines.
0000  ;
0000  ;             V03-011 JLV0211         Jake VanNoy             2-JUL-1982
0000  ;                     Remove check for powerfail in unit init that prevents
0000  ;                     SETUP_UCB from being called. This insures that UCB fields
0000  ;                     are initialized correctly when Unit init is called for
0000  ;                     use with CSS unibus switch.
0000  ;
0000  ;             V03-010 KDM0002         Kathleen D. Morse       28-Jun-1982
0000  ;                     Added $DEVDEF, $IPLDEF, $PRDEF, and $SSDEF.
0000  ;
0000  ;
0000  ;--
```

```
    0000          .SBTTL  DECLARATIONS
    0000
    0000  ;
    0000  ; EXTERNAL DEFINITIONS:
    0000  ;
    0000          $ACBDEF                                 ; DEFINE ACB
    0000          $CRBDEF                                 ; DEFINE CRB
    0000          $DCDEF                                  ; DEVICE DEFINITIONS
    0000          $DDBDEF                                 ; DEFINE DDB
    0000          $DEVDEF                                 ; DEFINE DEVICE TYPES
    0000          $DYNDEF                                 ; DYNAMIC STRUCTURE DEFINITONS
    0000          $IDBDEF                                 ; DEFINE IDB OFFSETS
    0000          $IODEF                                  ; DEFINE I/O FUNCTION CODES
    0000          $IPLDEF                                 ; DEFINE INTERRUPT PRIORITY LEVELS
    0000          $IRPDEF                                 ; IRP DEFINITIONS
    0000          $ORBDEF                                 ; DEFINE OBJECT'S RIGHTS BLOCK OFFSETS
    0000          $PRDEF                                  ; DEFINE PROCESSOR REGISTERS
    0000          $SSDEF                                  ; DEFINE SYSTEM STATUS CODES
    0000          $TTYDEF                                 ; DEFINE TERMINAL DRIVER SYMBOLS
    0000          $TTDEF                                  ; DEFINE TERMINAL TYPES
    0000          $TT2DEF                                 ; DEFINE EXTENDED DEFINITIONS
    0000          $TQEDEF                                 ; DEFINE TIMER QUEUE OFFSETS
    0000          $UCBDEF                                 ; DEFINE UCB
    0000          $UBADEF                                 ; DEFINE UBA
    0000          $VECDEF                                 ; DEFINE VECTOR FOR CRB
    0000          $TTYMACS                                ; DEFINE TERMINAL DRIVER MACROS
    0000          $TTYDEFS                                ; DEFINE TERMINAL DRIVER SYMBOLS
    0000          $TTYMODEM                               ; DEFINE MODEM DEFINITIONS
    0000
    0000
    0000  ;
    0000  ; LOCAL STORAGE
    0000  ;
00000000          .PSECT  $$$105_PROLOGUE
    0000
    0000  ;
    0000  ; Driver prologue table:
    0000  ;
    0000
    0000  DZ$DPT::                                        ; DRIVER START
    0000  .IF DF DZV
    0000          DPTAB   -                               ; DRIVER PROLOGUE TABLE
    0000                  END=DZ$END,-                    ; End and offset to INIT's vectors
    0000                  UCBSIZE=UCB$C_TT_LENGTH,-       ; SIZE OF UCB
    0000                  FLAGS=DPT$M_NOUNLOAD,-          ; DO NOT ALLOW UNLOAD
    0000                  ADAPTER=UBA,-                   ; ADAPTER TYPE
    0000                  DEFUNITS=4,-                    ; DZV has 4 units
    0000                  NAME=DZDRIVER,-                 ; NAME OF DRIVER
    0000                  VECTOR=PORT_VECTOR              ; PORT DRIVER VECTOR TABLE
    0038  .IFF
    0038          DPTAB   -                               ; DRIVER PROLOGUE TABLE
    0038                  END=DZ$END,-                    ; End and offset to INIT's vectors
    0038                  UCBSIZE=UCB$C_TT_LENGTH,-       ; SIZE OF UCB
    0038                  FLAGS=DPT$M_NOUNLOAD,-          ; DO NOT ALLOW UNLOAD
    0038                  ADAPTER=UBA,-                   ; ADAPTER TYPE
    0038                  DEFUNITS=8,-                    ; Number of units to create
    0038                  NAME=DZDRIVER,-                 ; NAME OF DRIVER
```

```
    0038                     VECTOR=PORT_VECTOR        ; PORT DRIVER VECTOR TABLE
    0038        .ENDC
    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$GB_RSPEED; DEFAULT SPEED
    0075                DPT_STORE UCB,UCB$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$GB_RSPEED   ; DEFAULT SPEED
    00A3                DPT_STORE UCB,UCB$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,DZ$DDT
    00D3
    00D3                DPT_STORE REINIT
    00D3                DPT_STORE CRB,CRB$L_INTD+VEC$L_INITIAL,D,DZ$INITIAL   ; CONTROLLER INIT
    00D8                DPT_STORE CRB,CRB$L_INTD+VEC$L_UNITINIT,D,DZ$INITLINE; UNIT INIT
    00DD                DPT_STORE END
    0000
    0000                DDTAB   DEVNAM = DZ,-   ; DUMMY DZ PORT DRIVER DISPATCH TABLE
    0000                        START  = 0,-
    0000                        FUNCTB = 0
    0038
00000038                .PSECT  $$$115_DRIVER
    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
    0038        PORT_VECTOR:
    0038
```

```
        0038  ;
        0038  ; DZ-11 SPECIFIC DISPATCH TABLE
        0038  ;
        0038          $VECINI DZ11,DZ$NULL
        0070          $VEC    STARTIO,DZ11$STARTIO
        003C          $VEC    SET_LINE,DZ$SET_LINE
        0044          $VEC    DS_SET,DZ11$DS_SET
        0048          $VEC    XON,DZ11$XON
        004C          $VEC    XOFF,DZ11$XOFF
        0050          $VEC    STOP,DZ$STOP
        0054          $VEC    ABORT,DZ$ABORT
        005C          $VEC    RESUME,DZ11$RESUME
        0060          $VEC    SET_MODEM,DZ11$SET_MODEM
        0064          $VEC    MAINT,DZ11$MAINT
        006C  .IF NDF DZV
        006C          $VECEND END=NO
        006C  ;
        006C  ; DZ-32 SPECIFIC DISPATCH TABLE
        006C  ;
        006C
        006C          $VECINI DZ32,DZ$NULL
        006C          $VEC    STARTIO,DZ32$STARTIO          ; START NEW OUTPUT
        006C          $VEC    SET_LINE,DZ$SET_LINE          ; SET NEW PARITY/SPEED
        006C          $VEC    DS_SET,DZ32$DS_SET            ; SET NEW OUTPUT MODEM SIGNALS
        006C          $VEC    XON,DZ32$XON                  ; SEND XON
        006C          $VEC    XOFF,DZ32$XOFF                ; SEND XOFF
        006C          $VEC    STOP,DZ$STOP                  ; STOP CURRENT OUTPUT
        006C          $VEC    ABORT,DZ$ABORT                ; ABORT CURRENT OUTPUT
        006C          $VEC    RESUME,DZ32$RESUME            ; RESUME STOPPED OUTPUT
        006C          $VEC    MAINT,DZ32$MAINT              ; INVOKE MAINTENANCE FUNCTIONS
        006C  .ENDC
        006C          $VECEND
        0074  DZ$NULL:                                      ; NULL PORT ROUTINE
    05  0074          RSB
        0075
        0075
```

```
0075                  .SBTTL  REGISTER DEFINITIONS
0075
0075          ;
0075          ; CSR BIT DEFINITIONS ( CSR ) ( READ/WRITE )
0075          ;
0075                  $VIELD  DZCSR,0,<-
0075                          <MODE,1,M>,-       ; DZ32 - MODE/ DZ11 - UNUSED
0075                          <DS_ENAB,1,M>,-    ; DZ32 - DATA SET INTERRUPT ENABLE
0075                          <,1>,-             ; UNUSED
0075                          <MAINT,1,M>,-      ; LINE TURNAROUND
0075                          <CLEAR,1,M>,-      ; MASTER RESET
0075                          <MASTENAB,1,M>,-   ; MASTER SCAN ENABLE
0075                          <RCVINT,1,M>,-     ; RECEIVER INTERRUPT ENABLE
0075                          <RCVRDY,1,M>,-     ; RECEIVER READY
0075                          <LINE,3,M>,-       ; LINE NUMBAE (0 - 7)
0075                          <DS_CHG,1,M>,-     ; DZ32 - DATA SET INTERRUPT
0075                          <,2>,-             ; UNUSED
0075                          <SNDINT,1,M>,-     ; TRANSMIT INTERRUPT ENABLE
0075                          <SNDRDY,1,M>-      ; TRANSMITTER READY
0075                          >
0075          ;
0075          ; RECEIVER BUFFER ( CSR+2 ) ( READ ONLY )
0075          ;
0075                  $VIELD  DZRCV,0,<-
0075                          <BUF,8,M>,-        ; RECEIVER DATA
0075                          <LINE,3,M>,-       ; LINE NUMBER (0 - 7)
0075                          <,1,>,-
0075                          <PARERR,1,M>,-     ; PARITY ERROR
0075                          <FRAMER,1,M>,-     ; FRAME ERROR
0075                          <OVERRUN,1,M>,-    ; OVERRUN ERROR
0075                          <VALID,1,M>-       ; DATA VALID
0075                          >
0075          ;
0075          ; LINE PARAMETER REGISTER ( CSR+2 ) ( WRITE ONLY )
0075          ;
0075
0075                  $VIELD  DZLPR,0,<-
0075                          <LINE,3,M>,-       ; LINE NUMBER (0-7)
0075                          <SIZE,2,M>,-       ; CHARACTER SIZE
0075                          <STOP,1,M>,-       ; NUMBER STOP BITS
0075                          <PARITY,1,M>,-     ; PARITY ENABLE
0075                          <ODD,1,M>,-        ; ODD PARITY
0075                          <SPEED,4,M>,-      ; LINE SPEED
0075                          <CLOCK,1,M>,-      ; RECEIVER CLOCK
0075                          <SPLIT,1,M>,-      ; DZ32 - SPLIT SPEED
0075                          >
0075
0075          ;
0075          ; DZ-32 SPECIFIC MODEM CONTROL
0075          ;
0075
0075                  $VIELD  DZLCS1,8,<-
0075                          <,7,>,-
0075                          <ACK,1,M>,-        ; READY FOR COMMAND/ UPDATE OUTPUT MODEM
0075                          >
0075
0075
```

```
      0075   344
```

```
                        0075          .SBTTL  CONTROLLER INITIALIZATION
                        0075
                        0075
                        0075
                        0075    ;++
                        0075    ; DZ$INITIAL - INITIALIZE INTERFACE
                        0075    ;
                        0075    ; FUNCTIONAL DESCRIPTION:
                        0075    ;
                        0075    ; THIS ROUTINE IS ENTERED AT SYSTEM STARTUP AND POWER RECOVERY.
                        0075    ;
                        0075    ; INPUTS:
                        0075    ;
                        0075    ;         R4 = ADDRESS OF THE UNIT CSR
                        0075    ;         R5 = IDB OF UNIT
                        0075    ;         R8 = ADDRESS OF THE UNIT CRB
                        0075    ;
                        0075    ; OUTPUTS:
                        0075    ;
                        0075    ;         R2 is destroyed.
                        0075    ;
                        0075    ; IMPLICIT INPUTS:
                        0075    ;
                        0075    ;         IPL = IPL$_POWER
                        0075    ;
                        0075    ;--
                        0075    DZ$INITIAL::                                   ; INITIALIZE DZ UNIT
                        0075
                        0075    ;
                        0075    ; SET UP CONTROLLER
                        0075    ;
                        0075            class_ctrl_init dz$dpt,port_vector
                        00A1
        64    10    B0  00A1    25$:    MOVW    #DZCSR$M_CLEAR,(R4)    ; INIT CONTROLLER RESET
                        00A4
                        00A4    ;
                        00A4    ;       WAIT TILL CONTROLLER INIIALIZATION IS COMPLETE
                        00A4    ;
                        00A4            TIMEWAIT        #500,#DZCSR$M_CLEAR,(R4),W,.FALSE.
                        00CB
  64    4063 8F     B0  00CB            MOVW    #<<DZCSR$M_MASTENAB>!-  ;
                        00D0                    <DZCSR$M_RCVINT>!-      ; ENABLE RECEIVER INTERRUPTS
                        00D0                    <DZCSR$M_SNDINT>!-      ; ENABLE TRANSMITTER INTERRUPTS
                        00D0                    <DZCSR$M_DS_ENAB>!-     ; ENABLE DZ-32 DATA SET INTERRUPTS
                        00D0                    <DZCSR$M_MODE>>,(R4)    ; ENABLE ENHANCED MODE ON DZ-32
         21 50      E9  00D0            BLBC    R0,DZ$CTRL_ERROR
                        00D3
         52   64    B0  00D3            MOVW    (R4),R2                 ; GET NEW STATUS
      19 52   00    E0  00D6            BBS     #DZCSR$V_MODE,R2,110$   ; BRANCH IF DZ-32 CONTROLLER
                        00DA
                        00DA    100$:
  0B A8    42 8F    90  00DA            MOVB    #DT$_DZ11,CRB$B_TT_TYPE(R8); CONTROLLER IS DZ11
                        00DF    ;
                        00DF    ; INIT DZ-11 INTERRUPT VECTORS
                        00DF    ;       THIS IS DONE HERE TO ALLOW THE DRIVER TO SERVICE INTERUPTS
                        00DF    ; FOR BOTH THE DZ-11 AND DZ-32 BETWEEN CONTROLER AN UNIT INIT.
                        00DF    ;
```

```
          28 A8   0000028E'EF   DE  00DF          MOVAL   DZ11$INTINP,CRB$L_INTD+4(R8)      ; INIT RECEIVER VECTOR
          4C A8   000003FA'EF   DE  00E7          MOVAL   DZ11$INTOUT,CRB$L_INTD2+4(R8)     ; INIT TRANSMITTER VECTOR
                                    00EF
                          1C A8 B4  00EF          CLRW    CRB$B_DZ_RING(R8)          ; RESET CURRENT DZ-11 MODEM STATE
                                05  00F2          RSB
                                    00F3
                                    00F3 110$:    ;DZ-32 CONTROLLER SPECIFIC INIT
                                    00F3 .if ndf DZV
                                    00F3
                                    00F3          MOVB    #DT$_DZ32,CRB$B_TT_TYPE(R8)       ; CONTROLLER IS DZ-32
                                    00F3          CLRB    IDB$B_TT_ENABLE(R5)               ; RESET DZ-32 LINE ENABLE
                                    00F3          MOVB    IDB$B_TT_ENABLE(R5),7(R4)         ; RESET TRANSMIT LINE ENABLES
                                    00F3 ;
                                    00F3 ; INIT DZ-32 ALTERNATE INTERRUPT VECTORS
                                    00F3 ;
                                    00F3          MOVAL   DZ32$INTINP,CRB$L_INTD+4(R8)      ; INIT RECEIVER VECTOR
                                    00F3          MOVAL   DZ32$INTOUT,CRB$L_INTD2+4(R8)     ; INIT TRANSMITTER VECTOR
                                    00F3 .endc
                                05  00F3          RSB
                                    00F4
                                    00F4 DZ$CTRL_ERROR:
                                05  00F4          RSB
                                    00F5
```

```
                         00F5            .SBTTL  UNIT INITIALIZATION
                         00F5     ;++
                         00F5     ; DZ$INITLINE - UNIT INITIALIZATION
                         00F5     ;
                         00F5     ; FUNCTIONAL DESCRIPTION:
                         00F5     ;
                         00F5     ; THIS ROUTINE PERFORMS A SIMPLE UNIT INITIALIZATION.
                         00F5     ;
                         00F5     ; INPUTS:
                         00F5     ;
                         00F5     ;       R5 = UCB ADDRESS
                         00F5     ;
                         00F5     ; OUTPUTS:
                         00F5     ;
                         00F5     ;       R2,R5 ARE PRESERVED.
                         00F5     ;--
                         00F5
                         00F5     DZ$INITLINE::
                         00F5
      54    24 A5    D0  00F5            MOVL    UCB$L_CRB(R5),R4        ; GET CRB ADDRESS
                         00F9
                         00F9     .IF NDF DZV     ; IF NOT DZV
                         00F9
                         00F9            MOVAL   DZ32$VEC,R0             ; SET DZ-32 PORT VECTOR TABLE
                         00F9            CMPB    #DT$_DZ32,CRB$B_TT_TYPE(R4)     ; IS IT DZ-32 ?
                         00F9            BEQL    5$                      ; YES
                         00F9
                         00F9     .ENDC ; END OF DZ32 CODE
                         00F9
    50    FF3B CF    DE  00F9            MOVAL   DZ11$VEC,R0             ; SET DZ-11 PORT VECTOR TABLE
                         00FE     5$:    CLASS_UNIT_INIT
      64 A5    10    A8  0147            BISW    #UCB$M_ONLINE,UCB$W_STS(R5); SET ONLINE
  53   01    54 A5   78  014B     10$:   ASHL    UCB$W_UNIT(R5),#1,R3    ; BUILD UNIT'S BIT MASK
   0106 C5     53    B0  0150            MOVW    R3,UCB$W_TT_UNITBIT(R5) ; SAVE IT
   51   0114 C5      D0  0155            MOVL    UCB$L_TT_CLASS(R5),R1   ; ADDRESS CLASS VECTOR TABLE
           08 B1     16  015A            JSB     @CLASS_SETUP_UCB(R1)    ; INIT UCB FIELDS
                         015D     20$:
    000004A7'EF      16  015D            JSB     DZ$SET_LINE             ; INIT SPEED/PARITY
                         0163
      54    24 A5    D0  0163            MOVL    UCB$L_CRB(R5),R4        ; GET CRB ADDRESS
                         0167
                         0167     .IF NDF DZV     ; IF NOT DZV
                         0167
                         0167            CMPB    #DT$_DZ11,CRB$B_TT_TYPE(R4)     ; CONTROLLER DZ11?
                         0167            BEQL    25$                     ; YES
                         0167
                         0167
                         0167     ;
                         0167     ; INIT RECEIVER MODEM STATUS FOR DZ-32
                         0167     ;
                         0167            MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4    ; GET CSR ADDRESS
                         0167
                         0167     ; WAIT TILL MODEM CONTROL READY FOR COMMNAD
                         0167
                         0167            TIMEWAIT #500,#DZLCS1$M_ACK,4(R4),W,.TRUE.
                         0167            BLBC    R0,DZ$UNIT_ERROR
                         0167            MOVW    UCB$W_UNIT(R5),4(R4)            ; REQUEST STATUS ON LINE
```

```
                          0167
                          0167     ; WAIT FOR COMPLETION
                          0167
                          0167              TIMEWAIT #500,#DZLCS1$M_ACK,4(R4),W,.TRUE.
                          0167              BLBC     R0,DZ$UNIT_ERROR
                          0167              MOVB     4(R4),UCB$B_TT_DS_RCV(R5)        ; UPDATE RECEIVER MODEM STATUS
                          0167
                          0167     25$:
                          0167     .ENDC    ; END OF DZ32 CODE
                          0167
               52    DD   0167              PUSHL    R2
          51   00    9A   0169              MOVZBL   #MODEM$C_INIT,R1         ; ASSUME INIT MODEM PROTOCOL
    50   0114  C5    D0   016C              MOVL     UCB$L_TT_CLASS(R5),R0    ; ADDRESS CLASS VECTOR TABLE
          0C   B0    16   0171              JSB      @CLASS_DS_TRAN(R0)       ; INVOKE TO INIT MODEM PROTOCOL
               52  8ED0   0174              POPL     R2
                          0177     30$:
  08 64   A5   05    E1   0177              BBC      #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
    50   0114  C5    D0   017C              MOVL     UCB$L_TT_CLASS(R5),R0    ; GET THE CLASS VECTOR TABLE ADDRESS
          20   B0    17   0181              JMP      @CLASS_POWERFAIL(R0)     ; AND GOTO THE POWERFAIL CODE
                     05   0184     40$:     RSB
                          0185     ;
                          0185     ; ERROR DETECTED DURING INITIALIZATION
                          0185     ;
                          0185
                          0185     DZ$UNIT_ERROR:
       64 A5   10    AA   0185              BICW     #UCB$M_ONLINE,UCB$W_STS(R5)      ; UNIT NOT ON LINE
                     05   0189              RSB
```

```
                        018A          .SBTTL  MAINTENANCE ROUTINES
                        018A    ;++
                        018A    ; DZ$MAINT - MAINTENANCE FUNCTIONS
                        018A    ;
                        018A    ; FUNCTIONAL DESCRIPTION:
                        018A    ;
                        018A    ; THIS ROUTINE PERFORMS MAINTENANCE FUNCTIONS FOR THE DZ .
                        018A    ;
                        018A    ;
                        018A    ; INPUTS:
                        018A    ;
                        018A    ;         R5 = UBC ADDRESS
                        018A    ;         UCB$B_TT_MAINT = FUNCTION TO BE PERFORMED
                        018A    ;
                        018A    ; OUTPUTS:
                        018A    ;
                        018A    ;--
                        018A    .IF NDF DZV
                        018A    DZ32$MAINT:
                        018A            BITB    #IO$M_LOOP@-7,-                 ; LOOPBACK FUNCTION
                        018A                    UCB$B_TT_MAINT(R5)
                        018A            BEQL    5$                              ; NO
                        018A            MOVZBL  #^X40,R2                        ; SPECIFY LOOPBACK CODE
                        018A            BRB     10$
                        018A    5$:
                        018A            BITB    #IO$M_UNLOOP@-7,-               ; RESET LOOPBACK FUNCTION
                        018A                    UCB$B_TT_MAINT(R5)
                        018A            BEQL    15$                             ; NO
                        018A            MOVZWL  #^X7200,R2                      ; SPECIFY UNLOOP CODE (BOTH)
                        018A    10$:
                        018A            JSB     DZ32$DS_SET                     ; UPDATE CONTROLLER
                        018A            MOVZBL  #1,R0                           ; INDICATE SUCCESS
                        018A            RSB
                        018A    15$:
                        018A            BITB    #IO$M_LOOP_EXT@-7,-             ; LOOPBACK FUNCTION
                        018A                    UCB$B_TT_MAINT(R5)
                        018A            BEQL    20$                             ; NO
                        018A            MOVZBL  #^X72,R2                        ; SPECIFY LOOPBACK CODE
                        018A            BRB     10$
                        018A
                        018A    20$:                                            ; CHECK OTHER FUNCTIONS
                        018A    .ENDC
                        018A    DZ11$MAINT:
        04      93      018A            BITB    #IO$M_LINE_OFF@-7,-             ; LINE OFF
      012A C5           018C                    UCB$B_TT_MAINT(R5)
        08      13      018F            BEQL    10$                             ; NO
     80 8F      88      0191            BISB    #UCB$M_TT_DSBL,-                ; DISABLE LINE
      012A C5           0194                    UCB$B_TT_MAINT(R5)
        0D      11      0197            BRB     20$
                        0199    10$:
        10      93      0199            BITB    #IO$M_LINE_ON@-7,-                      ; LINE ON
      012A C5           019B                    UCB$B_TT_MAINT(R5)
        10      13      019E            BEQL    30$                             ; NO
     80 8F      8A      01A0            BICB    #UCB$M_TT_DSBL,-                ; REENABLE LINE
      012A C5           01A3                    UCB$B_TT_MAINT(R5)
                        01A6    20$:
 000004A7'EF    16      01A6            JSB     DZ$SET_LINE                     ; IMPLEMENT FUNCTION
```

```
         50   01   9A  01AC   569          MOVZBL  #1,R0
                   05  01AF   570          RSB
                       01B0   571  30$:
              50   D4  01B0   572          CLRL    R0
                   05  01B2   573          RSB
                       01B3   574
                       01B3   575
```

```
                              01B3          .SBTTL  OUTPUT MODEM CONTROL
                              01B3    ;++
                              01B3    ; DZ$DS_SET - SET OUTPUT MODEM SIGNALS
                              01B3    ;
                              01B3    ; FUNCTIONAL DESCRIPTION:
                              01B3    ;
                              01B3    ; THIS ROUTINE OUTPUTS THE OUTPUT MODEM SIGNALS FOR THE SPECIFIED UNIT
                              01B3    ;
                              01B3    ; INPUTS:
                              01B3    ;
                              01B3    ;       R2 = LOW BYTE - SIGNALS TO ACTIVATE
                              01B3    ;            HIGH BYTE- SIGNALS TO DEACTIVATE
                              01B3    ;
                              01B3    ;       R5 = UBC ADDRESS
                              01B3    ;
                              01B3    ; OUTPUTS:
                              01B3    ;
                              01B3    ;       R0-R3 ARE USED.
                              01B3    ;--
                              01B3
                              01B3    DZ11$DS_SET:
                      54   DD 01B3            PUSHL   R4                      ; SAVE
            0125 C5   52   88 01B5            BISB    R2,UCB$B_TT_DS_TX(R5)   ; SET NEW OUTPUT SIGNALS
       52     52  F8 8F    78 01BA            ASHL    #-8,R2,R2               ; ACCESS SIGNALS TO RESET
            0125 C5   52   8A 01BF            BICB    R2,UCB$B_TT_DS_TX(R5)   ; RESET THEM
              54   24 A5   D0 01C4            MOVL    UCB$L_CRB(R5),R4        ; GET CRB ADDRESS
              53   2C B4   D0 01C8            MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R3
                              01CC                                            ; GET CSR ADDRESS
  51   0125 C5   01   01   EF 01CC            EXTZV   #TT$V_DS_DTR,#1,UCB$B_TT_DS_TX(R5),R1
                              01D3                                            ; GET CURRENT DTR FOR LINE
       51     51   54 A5   78 01D3            ASHL    UCB$W_UNIT(R5),R1,R1    ; SHIFT TO RELATIVE LINE POSITION
        1E A4   0106 C5    8A 01D8            BICB    UCB$W_TT_UNITBIT(R5),CRB$B_DZ_DTR(R4)
                              01DE                                            ; RESET CURRENT DTR FOR THAT LINE
           1E A4      51   88 01DE            BISB    R1,CRB$B_DZ_DTR(R4)     ; SET IT IF NEED BE
       05 A3     1E A4     90 01E2            MOVB    CRB$B_DZ_DTR(R4),5(R3)  ; UPDATE DTR STATUS FOR LINES
                   54 8ED0    01E7            POPL    R4
                           05 01EA            RSB
                              01EB    .IF NDF DZV
                              01EB    DZ32$DS_SET:
                              01EB            PUSHL   R4                      ; SAVE
                              01EB            BISB    R2,UCB$B_TT_DS_TX(R5)   ; SET NEW OUTPUT SIGNALS
                              01EB            ASHL    #-8,R2,R2               ; ACCESS SIGNALS TO RESET
                              01EB            BICB    R2,UCB$B_TT_DS_TX(R5)   ; RESET THEM
                              01EB            MOVL    UCB$L_CRB(R5),R4        ; GET CRB ADDRESS
                              01EB            MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R3
                              01EB            TIMEWAIT #100,#DZLCS1$M_ACK,4(R3),W,.TRUE.       ; WAIT FOR READY
                              01EB            MOVZWL  UCB$B_TT_DS_TX-T(R5),-(SP)               ; CREATE TEMP LOCATION
                              01EB            MOVB    UCB$W_UNIT(R5),(SP)     ; SET UNIT NUMBER
                              01EB            BISW    #DZLCS1$M_ACK,(SP)      ; ENABLE NEW OUTPUT SIGNALS
                              01EB            CVTLW   (SP)+,4(R3)             ; SET NEW OUPUT MODEM SIGNALS
                              01EB            POPL    R4
                              01EB            RSB
                              01EB    .ENDC
```

```
                              01EB   631          .SBTTL  DZ-11 MODEM POLLER
                              01EB   632  ;++
                              01EB   633  ; DZ$TIMER - POLL FOR DZ-11 MODEM TRANSITIONS
                              01EB   634  ;
                              01EB   635  ; FUNCTIONAL DESCRIPTION:
                              01EB   636  ;
                              01EB   637  ; THIS ROUTINE CHECKS FOR DZ-11 CONTROLLER MODEM
                              01EB   638  ; TRANSITION. IT UPDATES THE INPUT MODEM STATUS FOR EACH
                              01EB   639  ; LINE AND CALLS THE CLASS TRANSITION ROUTINE FOR EACH LINE WITH
                              01EB   640  ; A CHANGE.
                              01EB   641  ;
                              01EB   642  ; INPUTS:
                              01EB   643  ;
                              01EB   644  ;       R5 - TQE ADDRESS
                              01EB   645  ;
                              01EB   646  ; OUTPUTS:
                              01EB   647  ;
                              01EB   648  ;       R0 - R4 DESTROYED
                              01EB   649  ;
                              01EB   650  ;--
                              01EB   651
                              01EB   652  DZ$TIMER:
           0060 8F       BB   01EB   653          PUSHR   #^M<R5,R6>
    54  00000000'EF      DE   01EF   654          MOVAL   DZ$L_DIALUP,R4              ; GET DZ TIMER LIST HEAD
                              01F6   655  5$:
             54 64       D0   01F6   656          MOVL    (R4),R4                     ; GET NEXT CRB ADDRESS
                05       12   01F9   657          BNEQ    15$                         ; PROCESS LINES FOR THIS CRB
           0060 8F       BA   01FB   658          POPR    #^M<R5,R6>                  ;  RESTORE REGISTERS
                         05   01FF   659          RSB                                 ; RETURN FROM TIMER IMTERRUPT
                              0200   660  ;
                              0200   661  ;       TEST LINES ON THIS CONTROLLER FOR A TRANSITION
                              0200   662  ;
                              0200   663
                              0200   664  15$:
                54       DD   0200   665          PUSHL   R4                          ; SAVE TIMER THREAD
             54 18       C2   0202   666          SUBL    #CRB$L_DZ_MODEM,R4          ; GET ACTUAL CRB ADDRESS
          53 2C B4       D0   0205   667          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R3 ; GET CSR ADDRESS
    06 A3    1C A4       B1   0209   668          CMPW    CRB$B_DZ_RING(R4),6(R3)     ; ANY TRANSITIONS
                78       13   020E   669          BEQL    60$                         ; NONE
                              0210   670  ;
                              0210   671  ;       FIND WHICH SIGNALS CHANGED AND UPDATE THEM
                              0210   672  ;
       52    06 A3       90   0210   673          MOVB    6(R3),R2                    ; GET NEW RING
    50 1C A4    52       8D   0214   674          XORB3   R2,CRB$B_DZ_RING(R4),R0     ; FIND TRANSITIONED LINES
       1C A4    52       90   0219   675          MOVB    R2,CRB$B_DZ_RING(R4)        ; UPDATE CURRENT RING
       52    07 A3       90   021D   676          MOVB    7(R3),R2                    ; GET NEW CARRIER
    56 1D A4    52       8D   0221   677          XORB3   R2,CRB$B_DZ_CARRIER(R4),R6
          50    56       88   0226   678          BISB    R6,R0                       ; FLAG LINES WITH TRANSITIONED CARRIER
       1D A4    52       90   0229   679          MOVB    R2,CRB$B_DZ_CARRIER(R4)     ; UPDATE CURRENT CARRIER
                              022D   680  ;
                              022D   681  ;       PROCESS TRANSITIONED LINES
                              022D   682  ;
                              022D   683
    51  50  08  00       EA   022D   684  50$:    FFS     #0,#8,R0,R1                 ; FIND NEXT LINE NEEDING SERVICE
                54       13   0232   685          BEQL    60$                         ; DONE
       00 50    51       E5   0234   686          BBCC    R1,R0,55$                   ; RESET ATTENTION BIT FOR THIS LINE
                              0238   687  55$:
```

```
           56   2C A4   D0 0238            MOVL    CRB$L_INTD+VEC$L_IDB(R4),R6
          55   18 A641  D0 023C            MOVL    IDB$L_UCBLST(R6)[R1],R5 ; GET UCB FOR THAT LINE
                    EA  13 0241            BEQL    50$                    ; NONE
         E5 44 A5   15  E1 0243            BBC     #TT$V_MODEM,UCB$L_DEVDEPEND(R5),50$
                           0248                                           ; SKIP IF NOT MODEM LINE
                           0248            DSBINT  UCB$B_DIPL(R5)                  ; RAISE TO DEVICE IPL
  56    1C A4   01   51 EF 024F            EXTZV   R1,#1,CRB$B_DZ_RING(R4),R6      ; GET RING FOR THAT LINE
        01      06   56 F0 0255            INSV    R6,#TT$V_DS_RING,#1,-           ; UPDATE IT IN UCB
              0124   C5    0259                    UCB$B_TT_DS_RCV(R5)
  56    1D A4   01   51 EF 025C            EXTZV   R1,#1,CRB$B_DZ_CARRIER(R4),R6   ; GET CD FOR THAT LINE
        01      05   56 F0 0262            INSV    R6,#TT$V_DS_CARRIER,#1,-        ; UPDATE IT IN UCB
              0124   C5    0266                    UCB$B_TT_DS_RCV(R5)
                90   8F 88 0269            BISB    #<TT$M_DS_DSR!TT$M_DS_CTS>,-    ; ASSUME CTS AND DSR ALWAYS SET
              0124   C5    026C                    UCB$B_TT_DS_RCV(R5)
        52    0124   C5 90 026F            MOVB    UCB$B_TT_DS_RCV(R5),R2  ; GET CURRENT RECV MODEM STATUS
                51   03 9A 0274            MOVZBL  #MODEM$C_DATASET,R1     ; SIGNAL DATASET TRANSITION
        56    0114   C5 D0 0277            MOVL    UCB$L_TT_CLASS(R5),R6   ; GET CLASS VECTOR TABLE
                     1F BB 027C            PUSHR   #^M<R0,R1,R2,R3,R4>     ; SAVE VOLITAL REGISTERS
              0C     B6 16 027E            JSB     @CLASS_DS_TRAN(R6)      ; SIGNAL TRANSITION
                     1F BA 0281            POPR    #^M<R0,R1,R2,R3,R4>     ; RESTORE REGISTERS
                           0283            ENBINT                          ; RESTORE IPL
                     A5 11 0286            BRB     50$
                           0288
                           0288    60$:
                  54 8ED0  0288            POPL    R4                      ; RESTORE TIMER THREAD
                FF68   31  028B            BRW     5$
                           028E
                           028E
```

```
                            028E        .SBTTL  RECEIVER INTERRUPT SERIVCE
                            028E  ;++
                            028E  ; DZ$INTINP - DZ RECEIVER READY INTERRUPTS
                            028E  ;
                            028E  ; FUNCTIONAL DESCRIPTION:
                            028E  ;
                            028E  ; THIS ROUTINE IS ENTERED WHEN A CHARACTER IS AVAILABLE IN THE UNIT'S
                            028E  ; SILO. THE CHARACTER IS EXTRACTED AND IS PASSED TO THE ASSOCIATED
                            028E  ; CLASS DRIVER. IF THE CLASS DRIVER RETURNS CHARACTERS(S) THEN NEW
                            028E  ; OUTPUT IT INITIATED (NORMALLY ECHO).
                            028E  ;
                            028E  ; INPUTS:
                            028E  ;
                            028E  ;         00(SP) = ADDRESS OF IDB
                            028E  ;
                            028E  ; IMPLICIT INPUTS:
                            028E  ;
                            028E  ;         R0,R1,R2,R3,R4,R5 ARE SAVED ON STACK.
                            028E  ;
                            028E  ; OUTPUTS:
                            028E  ;
                            028E  ;         THE INTERRUPT IS DISMISSED WHEN THE SILO IS EMPTY.
                            028E  ;
                            028E  ;--
                            028E  DZ11$INTINP::                                    ; DZ-11 INPUT INTERRUPTS
                            028E  ;
                            028E  ; GET THE CSR ADDRESS
                            028E  ;
        54      9E    D0    028E            MOVL    @(SP)+,R4                  ; GET THE IDB ADDRESS
                54    DD    0291            PUSHL   R4                         ; SAVE IDB ADDRESS
        50      64    D0    0293            MOVL    (R4),R0                    ; GET THE CSR ADDRESS
                            0296  ;
                            0296  ; GET THE CHARACTER FROM THE INTERFACE
                            0296  ;
      53     02 A0    B0    0296  25$:      MOVW    2(R0),R3                   ; GET THE CHARACTER,ERRORS AND LINE NUMBER
                73    18    029A            BGEQ    100$                       ;SILO EMPTY
    53     7000 8F    B3    029C            BITW    #<DZRCV$M_PARERR>!-
                            02A1                    <DZRCV$M_OVERRUN>!-
                            02A1                    <DZRCV$M_FRAMER>,R3        ;ERRORS?
                48    12    02A1            BNEQ    50$                        ;YES,PROCESS THEM
  52     53    F8 8F    78    02A3  27$:      ASHL    #-8,R3,R2                  ; GET THE LINE NUMBER
 52    FFFFFFF8 8F    CA    02A8            BICL    #^C<7>,R2                  ;
          53      53    9A    02AF            MOVZBL  R3,R3                      ; CLEAR THE HIGH BYTES OF CHARACTER
    55     18 A442    D0    02B2            MOVL    IDB$L_UCBLST(R4)[R2],R5    ; GET THE UCB FOR THAT LINE
                DD    13    02B7            BEQL    25$                        ; IF EQL THEN NOT THERE
         0110 D5    16    02B9            JSB     @UCB$L_TT_PUTNXT(R5)       ; BUFFER THE CHARACTER
         010B C5    95    02BD            TSTB    UCB$B_TT_OUTYPE(R5)        ; DID HE RETURN ANYTHING TO OUTPUT
                17    15    02C1            BLEQ    40$                        ; NONE OR STRING OUTPUT
    0108 C5      53    90    02C3            MOVB    R3,UCB$W_TT_HOLD(R5)       ; SAVE THE CHARACTER IN TANK
         0400 8F    A8    02C8            BISW    #TTY$M_TANK_HOLD,-         ; SIGNAL CHARACTER IN TANK
         0108 C5          02CC                    UCB$W_TT_HOLD(R5)
 04 A0    0106 C5    A8    02CF            BISW    UCB$W_TT_UNITBIT(R5),4(R0)         ;  ENABLE LINE
        54      6E    D0    02D5  30$:      MOVL    (SP),R4                    ; GET IDB ADDRESS
                BC    11    02D8            BRB     25$                        ; CONTINUE
                            02DA  40$:
                F9    13    02DA            BEQL    30$                        ; NO CHARACTER
         0800 8F    A8    02DC            BISW    #TTY$M_TANK_BURST,-        ; SIGNAL BURST
```

```
             0108 C5      02E0  774                  UCB$W_TT_HOLD(R5)
      04 A0  0106 C5   A8 02E3  775         BISW     UCB$W_TT_UNITBIT(R5),4(R0)      ;  ENABLE LINE
                  EA   11 02E9  776         BRB      30$
                          02EB  777 ;
                          02EB  778 ; SILO EMPTY OR CHARACTER IN ERROR
                          02EB  779 ;
                          02EB  780 50$:
                          02EB  781 ;
                          02EB  782 ; PROCESS PARITY, FRAME OR OVERRUN ERROR
                          02EB  783 ;
   52    53    F8 8F   78 02EB  784         ASHL     #-8,R3,R2                     ; GET LINE NUMBER
  52   FFFFFFF8 8F     CA 02F0  785         BICL     #^C<7>,R2                     ;
     55    18 A442     D0 02F7  786         MOVL     IDB$L_UCBLST(R4)[R2],R5;  GET UCB ADDRESS
                  0A   13 02FC  787         BEQL     70$                           ; IF EQL THEN NO UCB
     52    0114 C5     D0 02FE  788         MOVL     UCB$L_TT_CLASS(R5),R2         ; GET CLASS DISPATCH
                          0303  789
               14 B2   16 0303  790 60$:    JSB      @CLASS_READERROR(R2)          ; SIGNAL ERROR
                  9B   12 0306  791         BNEQ     27$                           ; BRANCH WITH CHARACTER TO MAIN PATH
     60    0080 8F     B3 0308  792 70$:    BITW     #^X080,(R0)                   ; VALID CHARACTER IN SILO NOW?
                  C6   12 030D  793         BNEQ     30$                           ; IF NEQ THEN YES
            5E    04   C0 030F  794 100$:   ADDL     #4,SP                         ; REMOVE IDB ADDRESS
            50    8E   7D 0312  795         MOVQ     (SP)+,R0                      ; RESTORE REGISTERS
            52    8E   7D 0315  796         MOVQ     (SP)+,R2                      ;
            54    8E   7D 0318  797         MOVQ     (SP)+,R4                      ;
                       02 031B  798         REI
                          031C  799
```

```
031C      .IF NDF DZV
031C      ;
031C      ; DZ-32 INPUT INTERRUPT SERVICE
031C      ;
031C
031C      DZ32$INTINP::                                  ; DZ-32 INPUT INTERRUPTS
031C      ;
031C      ; GET THE CSR ADDRESS
031C      ;
031C              MOVL    @(SP)+,R4                      ; GET THE IDB ADDRESS
031C              PUSHL   R4                             ; SAVE IDB ADDRESS
031C              MOVL    (R4),R0                        ; GET THE CSR ADDRESS
031C              BICW    #<<DZCSR$M_RCVINT>!-           ; DISABLE RECEIVER INTERRUPTS
031C                      <DZCSR$M_DS_ENAB>>,-           ; DISABLE DZ-32 DATA SET INTERRUPTS
031C                      (R0)                           ; DZ-32 during the interrupt service routine
031C      ;
031C      ; GET THE CHARACTER FROM THE INTERFACE
031C      ;
031C      25$:    MOVW    2(R0),R3                       ; GET THE CHARACTER,ERRORS AND LINE NUMBER
031C              BGEQ    100$                           ;SILO EMPTY
031C              BITW    #<DZRCV$M_PARERR>!-
031C                      <DZRCV$M_OVERRUN>!-            ;ERRORS?
031C                      <DZRCV$M_FRAMER>,R3            ;ERRORS?
031C              BNEQ    50$                            ;YES,PROCESS THEM
031C      27$:    ASHL    #-8,R3,R2                      ; GET THE LINE NUMBER
031C              BICL    #^C<7>,R2                      ;
031C              MOVZBL  R3,R3                          ; CLEAR THE HIGH BYTES OF CHARACTER
031C              MOVL    IDB$L_UCBLST(R4)[R2],R5        ; GET THE UCB FOR THAT LINE
031C              BEQL    25$                            ; IF EQL THEN NOT THERE
031C              JSB     @UCB$L_TT_PUTNXT(R5)           ; BUFFER THE CHARACTER
031C              TSTB    UCB$B_TT_OUTYPE(R5)            ; DID HE RETURN ANYTHING TO OUTPUT
031C              BLEQ    40$                            ; NONE OR STRING OUTPUT
031C              MOVB    R3,UCB$W_TT_HOLD(R5)           ; SAVE THE CHARACTER IN TANK
031C              BISW    #TTY$M_TANK_HOLD,-             ; SIGNAL CHARACTER IN TANK
031C                      UCB$W_TT_HOLD(R5)
031C      28$:
031C              MOVL    (SP),R4                        ; RESTORE IDB ADDRESS
031C              BISB    UCB$W_TT_UNITBIT(R5),-         ;  ENABLE LINE
031C                      IDB$B_TT_ENABLE(R4)
031C              MOVB    IDB$B_TT_ENABLE(R4),7(R0)
031C
031C      30$:    MOVL    (SP),R4                        ; GET IDB ADDRESS
031C              BRB     25$                            ; CONTINUE
031C      40$:
031C              BEQL    30$                            ; NO CHARACTER
031C              BISW    #TTY$M_TANK_BURST,-            ; SIGNAL BURST
031C                      UCB$W_TT_HOLD(R5)
031C              BRB     28$
031C      ;
031C      ; SILO EMPTY OR CHARACTER IN ERROR
031C      ;
031C      50$:
031C      ;
031C      ; PROCESS PARITY, FRAME OR OVERRUN ERROR
031C      ;
031C              ASHL    #-8,R3,R2                      ; GET LINE NUMBER
031C              BICL    #^C<7>,R2                      ;
```

```
031C              MOVL    IDB$L_UCBLST(R4)[R2],R5 ; GET UCB ADDRESS
031C              BEQL    70$                     ; IF EQL THEN NO UCB
031C              MOVL    UCB$L_TT_CLASS(R5),R2   ; GET CLASS DISPATCH
031C
031C     60$:     JSB     @CLASS_READERROR(R2)    ; SIGNAL ERROR
031C              BNEQ    27$                     ; BRANCH WITH CHARACTER TO MAIN PATH
031C     70$:     BITW    #^X080,(R0)             ; VALID CHARACTER IN SILO NOW?
031C              BNEQ    30$                     ; IF NEQ THEN YES
031C     100$:
031C              MOVW    (R0),R3                 ; TEST FOR MODEM TRANSITION
031C              BBS     #DZCSR$V_DS_CHG,R3,200$ ; BRANCH IF MODEM TRANSITION
031C              BISW    #<<DZCSR$M_RCVINT>!-    ;  ENABLE RECEIVER INTERRUPTS
031C                      <DZCSR$M_DS_ENAB>>,-    ;  ENABLE DZ-32 DATA SET INTERRUPTS
031C                      (R0)                    ; DZ-32 BEFORE EXITING
031C              ADDL    #4,SP                   ; REMOVE IDB ADDRESS
031C              MOVQ    (SP)+,R0                ; RESTORE REGISTERS
031C              MOVQ    (SP)+,R2                ;
031C              MOVQ    (SP)+,R4                ;
031C              REI
031C     200$:    PUSHL   R0                      ; SAVE R0
031C              MOVB    6(R0),R3                ; GET NEW RECEIVE MODEM SIGNALS
031C              BICL3   #^C<7>,R3,R2            ; ISOLATE UNIT NUMBER
031C              MOVL    IDB$L_UCBLST(R4)[R2],R5 ; GET ASSOCAITED UCB
031C              BEQL    110$                    ; NONE
031C              MOVB    R3,UCB$B_TT_DS_RCV(R5)  ; UPDATE INPUT MODEM SIGNALS
031C              MOVZBL  R3,R2                   ; LOAD ARGUMENT
031C              MOVZBL  #MODEM$C_DATASET,R1     ; MODEM TRANSITION
031C              MOVL    UCB$L_TT_CLASS(R5),R3   ; ACCESS CLASS VECTORS
031C              JSB     @CLASS_DS_TRAN(R3)      ; SIGNAL TRANSITION
031C     110$:    POPL    R0                      ; RESTORE R0
031C              BRW     30$                     ; DISMISS INTERRUPT
031C     .ENDC
```

```
                              031C        .SBTTL  START I/O ROUTINE
                              031C  ;++
                              031C  ; DZ$STARTIO - START I/O OERATION ON DZ
                              031C  ;
                              031C  ; FUNCTIONAL DESCRIPTION:
                              031C  ;
                              031C  ; THIS ROUTINE IS ENTERED FROM THE DEVICE INDEPENDENT TERMINAL STARTIO
                              031C  ; ROUTINE TO ENABLE OUTPUT INTERRUPTS ON AN IDLE DZ UNIT.
                              031C  ;
                              031C  ; INPUTS:
                              031C  ;
                              031C  ;       R3 =    CHARACTER       AND     CC = PLUS
                              031C  ;               ADDRESS         AND     CC = NEGATIVE
                              031C  ;
                              031C  ;       R5 = UCB ADDRESS
                              031C  ;
                              031C  ; OUTPUTS:
                              031C  ;
                              031C  ;       R5 = UCB ADDRESS
                              031C  ;--
                              031C  DZ11$STARTIO::                                  ; START I/O ON UNIT
                  16     18   031C          BGEQ    20$                             ; SINGLE CHARACTER
             0800 8F     A8   031E          BISW    #TTY$M_TANK_BURST,-             ; SIGNAL BURST ACTIVE
             0108 C5          0322                  UCB$W_TT_HOLD(R5)
         51    24 A5     D0   0325  10$:    MOVL    UCB$L_CRB(R5),R1                ; GET CRB OF UNIT
         51    2C B1     D0   0329          MOVL    @CRB$L_INTD+VEC$L_IDB(R1),R1; GET CSR
   04 A1     0106 C5     A8   032D          BISW    UCB$W_TT_UNITBIT(R5),4(R1)      ;  ENABLE LINE
                  05          0333          RSB                                     ; RETURN TO CALLER
                              0334  20$:
        0108 C5   53     90   0334          MOVB    R3,UCB$W_TT_HOLD(R5)            ; SAVE OUTPUT CHARACTER
             0400 8F     A8   0339          BISW    #TTY$M_TANK_HOLD,-              ; SIGNAL CHARACTER IN TANK
             0108 C5          033D                  UCB$W_TT_HOLD(R5)
                  E3     11   0340          BRB     10$
                              0342
                              0342  .IF NDF DZV
                              0342  DZ32$STARTIO::                                  ; START I/O ON UNIT
                              0342          BGEQ    20$                             ; SINGLE CHARACTER SPECIFIED
                              0342          BISW    #TTY$M_TANK_BURST,-             ; SIGNAL BURST ACTIVE
                              0342                  UCB$W_TT_HOLD(R5)
                              0342  10$:    MOVL    UCB$L_CRB(R5),R1                ; GET CRB OF UNIT
                              0342          MOVL    CRB$L_INTD+VEC$L_IDB(R1),R4     ; GET IDB ADDRESS
                              0342          MOVL    (R4),R1                         ; GET CSR ADDRESS
                              0342          BISB    UCB$W_TT_UNITBIT(R5),IDB$B_TT_ENABLE(R4)
                              0342          MOVB    IDB$B_TT_ENABLE(R4),7(R1)
                              0342          RSB                                     ; RETURN TO CALLER
                              0342  20$:
                              0342          MOVB    R3,UCB$W_TT_HOLD(R5)            ; SAVE OUTPUT CHARACTER
                              0342          BISW    #TTY$M_TANK_HOLD,-              ; SIGNAL CHARACTER IN TANK
                              0342                  UCB$W_TT_HOLD(R5)
                              0342          BRB     10$
                              0342  .ENDC
```

```
                              0342          .SBTTL  PORT ROUTINES  STOP,RESUME,XON,XOFF
                              0342  ;++
                              0342  ; DZ$XOFF -      SEND XOFF
                              0342  ; DZ$XON -       SEND XON
                              0342  ; DZ$STOP -      STOP OUTPUT
                              0342  ; DZ$ABORT -     ABORT CURRENT OUTPUT
                              0342  ; DZ$RESUME -    RESUME STOPPED OUTPUT
                              0342  ;
                              0342  ; FUNCTIONAL DESCRIPTION:
                              0342  ;
                              0342  ; THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
                              0342  ; CONTROL OUTPUT ON THE PORT
                              0342  ;
                              0342  ; INPUTS:
                              0342  ;
                              0342  ;         R5 = UCB ADDRESS
                              0342  ;
                              0342  ; OUTPUTS:
                              0342  ;
                              0342  ;         R5 = UCB ADDRESS
                              0342  ;--
                              0342          .ENABLE LSB
                              0342  ;
                              0342  ; SCHEDULE XOFF TO BE SEND
                              0342  ;
                              0342  ; INPUTS:
                              0342  ;         R3 - CHARACTER TO BE SENT AS FLOW CONTROL
                              0342  ;
                              0342  DZ11$XOFF:
                              0342  ;
                              0342  ; SCHEDULE XON TO BE SENT
                              0342  ;
                              0342  DZ11$XON:
0108 C5   0100 8F      A8     0342          BISW    #TTY$M_TANK_PREMPT,UCB$W_TT_HOLD(R5)    ; SCHEDULE XON
     010A C5   53      90     0349          MOVB    R3,UCB$B_TT_PREMPT(R5)                  ; SAVE THE CHARACTER
                              034E
   18 64 A5    01      E0     034E          BBS     #UCB$V_INT,UCB$W_STS(R5),10$            ; IF OUTPUT ACTIVE,
                              0353                                                          ; FINISHED
               51      DD     0353          PUSHL   R1                                      ; SAVE A REGISTER
      51    24 A5      D0     0355          MOVL    UCB$L_CRB(R5),R1                        ; ACCESS CRB ADDRESS
      51    2C B1      D0     0359          MOVL    @CRB$L_INTD+VEC$L_IDB(R1),R1            ; GET CSR ADDRESS
  04 A1   0106 C5      A8     035D          BISW    UCB$W_TT_UNITBIT(R5),4(R1)                      ; ENABLE LINE
               51    8ED0     0363          POPL    R1
   00 64 A5    01      E2     0366          BBSS    #UCB$V_INT,UCB$W_STS(R5),10$            ; SHOW OUTPUT ACTIVE
                              036B  10$:
                       05     036B          RSB
                              036C          .DISABLE        LSB
                              036C  ;
                              036C  ; STOP PORT OUTPUT
                              036C  ;
                              036C  DZ$STOP:
          0200 8F      A8     036C          BISW    #TTY$M_TANK_STOP,-                      ; SCHEDULE STOP
          0108 C5             0370                  UCB$W_TT_HOLD(R5)
                       05     0373          RSB
                              0374  ;
                              0374  ; ABORT ANY CURRENT PORT OUTPUT ACTIVITY
                              0374  ;
```

```
                              0374  DZ$ABORT:
     0108 C5   0B   E5   0374           BBCC    #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),-   ; RESET BURST ACTIVE
               00        0379                   10$                                     ;
                         037A  10$:
                         037A           TIMSET  1                                       ; SET A TIMEOUT
                         038D                                                           ; IN CASE OUTPUT ACTIVE
                    05   038D           RSB
                         038E
                         038E  ;
                         038E  ; RESUME PREVIOUSLY STOPPED PORT OUTPUT
                         038E  ;
                         038E  DZ11$RESUME:
               51   DD   038E           PUSHL   R1                                      ; SAVE A REGISTER
 0108 C5  0200 8F   AA   0390  5$:      BICW    #TTY$M_TANK_STOP-
                         0397                   ,UCB$W_TT_HOLD(R5)                      ; RESET STOP CONDITIONS
  15 0108 C5   0B   E0   0397           BBS     #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),20$ ; BURST IN PROGRESS
                         039D  10$:                                                     ; CHAR IN TANK OR OTHER
                         039D           TIMSET  1
               1F   11   03B0           BRB     30$
                         03B2  20$:
     51   0120 C5   3C   03B2           MOVZWL  UCB$W_TT_OUTLEN(R5),R1                  ; GET NUMBER CHARACTERS
                         03B7           TIMSET  R1,R1                                   ;
                         03D1  30$:
    13 64 A5   01   E0   03D1           BBS     #UCB$V_INT,UCB$W_STS(R5),40$            ; SKIP IF OUTPUT ON
       51   24 A5   D0   03D6           MOVL    UCB$L_CRB(R5),R1                        ; ACCESS CRB ADDRESS
       51   2C B1   D0   03DA           MOVL    @CRB$L_INTD+VEC$L_IDB(R1),R1            ; GET CSR ADDRESS
   04 A1  0106 C5   A8   03DE           BISW    UCB$W_TT_UNITBIT(R5),4(R1)                    ; ENABLE LINE
    00 64 A5   01   E2   03E4           BBSS    #UCB$V_INT,UCB$W_STS(R5),40$            ; SHOW OUTPUT ACTIVE
                         03E9  40$:
               51 8ED0   03E9           POPL    R1
                    05   03EC           RSB
                         03ED  .IF NDF DZV
                         03ED           .ENABLE LSB
                         03ED  ;
                         03ED  ; SCHEDULE XOFF TO BE SENT
                         03ED  ;
                         03ED  DZ32$XOFF:
                         03ED  ;
                         03ED  ; SCHEDULE XON TO BE SENT
                         03ED  ;
                         03ED  DZ32$XON:
                         03ED           BISW    #TTY$M_TANK_PREMPT,UCB$W_TT_HOLD(R5)    ; SCHEDULE XON
                         03ED           MOVB    R3,UCB$B_TT_PREMPT(R5)                  ; SAVE THE CHARACTER
                         03ED                                                           ; IN THE PREMPT SLOT
                         03ED           BBS     #UCB$V_INT,UCB$W_STS(R5),10$            ; IF OUTPUT ACTIVE,
                         03ED                                                           ; FINISHED
                         03ED           PUSHR   #^M<R1,R4>                              ; SAVE  REGISTERS
                         03ED           MOVL    UCB$L_CRB(R5),R1                        ; ACCESS CRB ADDRESS
                         03ED           MOVL    CRB$L_INTD+VEC$L_IDB(R1),R4             ; GET IDB ADDRESS
                         03ED           MOVL    (R4),R1                                 ; GET CSR ADDRESS
                         03ED           BISB    UCB$W_TT_UNITBIT(R5),IDB$B_TT_ENABLE(R4)      ; ENABLE LINE
                         03ED           MOVB    IDB$B_TT_ENABLE(R4),7(R1)
                         03ED           POPR    #^M<R1,R4>
                         03ED           BBSS    #UCB$V_INT,UCB$W_STS(R5),10$            ; SHOW OUTPUT ACTIVE
                         03ED  10$:
                         03ED           RSB
                         03ED           .DISABLE        LSB
```

```
03ED
03ED      ;
03ED      ; RESUME STOPPED OUTPUT
03ED      ;
03ED      DZ32$RESUME:
03ED              PUSHR   #^M<R1,R4>                                  ; SAVE REGISTERS
03ED      5$:     BICW    #TTY$M_TANK_STOP-
03ED                      ,UCB$W_TT_HOLD(R5)                          ; RESET STOP CONDITIONS
03ED              BBS     #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),20$     ; BURST IN PROGRESS
03ED      10$:                                                        ; CHAR IN TANK OR OTHER
03ED              TIMSET  1                                           ; TIME OUT
03ED              BRB     30$
03ED      20$:
03ED              MOVZWL  UCB$W_TT_OUTLEN(R5),R1
03ED              TIMSET  R1,R1                                       ; SET THE TIMER
03ED      30$:
03ED              BBS     #UCB$V_INT,UCB$W_STS(R5),40$                ; SKIP IF OUTPUT ON
03ED              MOVL    UCB$L_CRB(R5),R1                            ; ACCESS CRB ADDRESS
03ED              MOVL    CRB$L_INTD+VEC$L_IDB(R1),R4                 ; GET IDB ADDRESS
03ED              MOVL    (R4),R1                                     ; GET CSR ADDRESS
03ED              BISB    UCB$W_TT_UNITBIT(R5),IDB$B_TT_ENABLE(R4)            ; ENABLE LINE
03ED              MOVB    IDB$B_TT_ENABLE(R4),7(R1)
03ED              BBSS    #UCB$V_INT,UCB$W_STS(R5),40$                ; SHOW OUTPUT ACTIVE
03ED      40$:
03ED              POPR    #^M<R1,R4>
03ED              RSB
03ED      .ENDC
```

```
                         03ED         .SBTTL  OUTPUT INTERRUPT SERVICE
                         03ED    ;++
                         03ED    ; DZ$INTOUT - DZ-11 OUTPUT INTERRUPT SERVICE
                         03ED    ;
                         03ED    ; FUNCTIONAL DESCRIPTION:
                         03ED    ;
                         03ED    ; THIS ROUTINE IS ENTERED WHEN THE DZ-11 FINDS A LINE ENABLED
                         03ED    ; AND AN EMPTY UART. THE CORRESPONDING UCB IS FOUND AND
                         03ED    ; ANY OUTSTANDING PORT OUTPUT IS DONE. WHEN ALL OUTSTANDING PORT
                         03ED    ; OUTPUT IS COMPLETED, THE CLASS DRIVER IS CALLED TO RETURN THE NEXT
                         03ED    ; CHARACTER OR STRING TO BE OUTPUT. IF NO MORE OUTPUT IS FOUND, THEN
                         03ED    ; THE LINE IS DISBALED.
                         03ED    ;
                         03ED    ; INPUTS:
                         03ED    ;
                         03ED    ;       SP(00) = ADDRESS OF THE IDB
                         03ED    ;
                         03ED    ; IMPLICIT INPUTS:
                         03ED    ;
                         03ED    ;       R0,R1,R2,R3,R4,R5 SAVED ON THE STACK.
                         03ED    ;
                         03ED    ; OUTPUTS:
                         03ED    ;
                         03ED    ;       THE INTERRUPT IS DISMISSED.
                         03ED    ;
                         03ED    ;--
                         03ED    DZ11_OUT_EXIT:                          ; EXIT OUTPUT INTERRUPT
          5E    04   C0  03ED            ADDL    #4,SP                   ; REMOVE IDB ADDRESS
          50    8E   7D  03F0            MOVQ    (SP)+,R0                ; RESTORE REGISTERS
          52    8E   7D  03F3            MOVQ    (SP)+,R2                ;
          54    8E   7D  03F6            MOVQ    (SP)+,R4                ;
                     02  03F9            REI                             ; DISMISS INTERRUPT
                         03FA
                         03FA    DZ11$INTOUT::                           ; DZ-11 OUTPUT INTERRUPT SERVICE
                         03FA
                         03FA    DZ11_OUT_LOOP:
       54    00 BE   D0  03FA            MOVL    @(SP),R4                ; GET THE IDB ADDRESS
          50    64   D0  03FE            MOVL    (R4),R0                 ; GET THE CSR ADDRESS
                         0401    ;
                         0401    ; GET THE LINE INFO FROM THE CSR
                         0401    ;
                         0401
          52    60   B0  0401            MOVW    (R0),R2                 ; GET THE CSR VALUE
                E7   18  0404            BGEQ    DZ11_OUT_EXIT           ; NO MORE LINES
     52   52   F8 8F   78  0406          ASHL    #-8,R2,R2               ; GET THE LINE NUMBER
   52   FFFFFFF8 8F   CA  040B           BICL    #^C<7>,R2               ;
      55    18 A442   D0  0412           MOVL    IDB$L_UCBLST(R4)[R2],R5; GET THE UCB ADDRESS
                E1   13  0417            BEQL    DZ11_OUT_LOOP           ; IF EQL THEN DISMISS
                         0419    ;
                         0419    ;       CHECK FOR BURST ACTIVE ON LINE
                         0419    ;
                08   91  0419            CMPB    #TTY$M_TANK_BURST@-8,-  ; ONLY BURST ACTIVE?
           0109 C5       041B                    UCB$W_TT_HOLD+1(R5)
                40   13  041E            BEQL    DZ11_BURST              ; YES, CONTINUE BURST
                         0420    ;
                         0420    ;       LOOK FOR NEXT OUTPUT STATE IN TANK
                         0420    ;
```

```
                               0420
 53  0109 C5   06   00   EA    0420            FFS     #0,#6,UCB$W_TT_HOLD+1(R5),R3
                               0427            CASE    R3,TYPE=B,<=                    ; DISPATCH
                               0427                    DZ11_PREMPT,-                   ; SEND PREMPT CHARACTER
                               0427                    DZ11_STOP,-                     ; STOP OUTPUT
                               0427                    DZ11_CHAR,-                     ; CHAR IN TANK
                               0427                    DZ11_BURST,-                    ; BURST IN PROGRESS
                               0427                    >
                               0433    ;
                               0433    ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
                               0433    ;
           64 A5   03   8A     0433            BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
                               0437    ;
                               0437    ; CALL CLASS DRIVER FOR MORE OUTPUT
                               0437    ;
              010C D5   16     0437            JSB     @UCB$L_TT_GETNXT(R5)            ; GET THE NEXT CHARACTER
   01    FF 8F  010B C5   8F   043B            CASEB   UCB$B_TT_OUTYPE(R5),#-1,#1; OPTOMIZE FOR THE SINGLE
                               0442                                                    ; CHARACTER CASE BY SETTING THE
                               0442                                                    ; LIMIT TO 1
                    0017'      0442    1$:     .WORD   DZ11_START_BURST-1$             ; BURST SPECIFIED
                    000A'      0444            .WORD   50$-1$                          ; NONE
                               0446    ;
                               0446    ; OUTPUT A CHARACTER TO THE DZ-11
                               0446    ;
           06 A0   53   9B     0446    20$:    MOVZBW  R3,6(R0)                        ; OUTPUT CHARACTER
                   AE   11     044A            BRB     DZ11_OUT_LOOP
                               044C    ;
                               044C    ; DISABLE OUTPUT ON THIS LINE
                               044C    ;
                               044C    50$:
                   01   E0     044C            BBS     #UCB$V_INT,-                    ; IF INT EXP, THEN DON'T RESET,
              A9 64 A5         044E                    UCB$W_STS(R5),DZ11_OUT_LOOP
                               0451                                                    ; COULD HAVE BEEN SET DURING CALLBACK
                               0451
      04 A0  0106 C5   AA      0451            BICW    UCB$W_TT_UNITBIT(R5),4(R0)      ; RESET THE OUTPUT ENABLE
                   A1   11     0457            BRB     DZ11_OUT_LOOP
                               0459
                               0459
                               0459    DZ11_START_BURST:
              0800 8F   A8     0459            BISW    #TTY$M_TANK_BURST,-             ; SIGNAL BURST ACTIVE
              0108 C5          045D                    UCB$W_TT_HOLD(R5)
                               0460    ;
                               0460    ;  CONTINUE BURST OUTPUT
                               0460    ;
                               0460    DZ11_BURST:
              011C D5   90     0460            MOVB    @UCB$L_TT_OUTADR(R5),-          ; OUTPUT NEXT BYTE
                06 A0          0464                    6(R0)
              011C C5   D6     0466            INCL    UCB$L_TT_OUTADR(R5)             ; UPDATE POINTER
              0120 C5   B7     046A            DECW    UCB$W_TT_OUTLEN(R5)             ; UPDATE COUNT
                   8A   12     046E            BNEQ    DZ11_OUT_LOOP                   ; NOT LAST CHARACTER
              0800 8F   AA     0470            BICW    #TTY$M_TANK_BURST,-             ; RESET BURST ACTIVE
              0108 C5          0474                    UCB$W_TT_HOLD(R5)
                 FF80   31     0477            BRW     DZ11_OUT_LOOP
                               047A    ;
                               047A    ; OUTPUT SINGLE CHARACTER
                               047A    ;
                               047A    DZ11_CHAR:
```

```
06 A0   0108 C5   90  047A          MOVB    UCB$W_TT_HOLD(R5),6(R0) ; OUTPUT CHAR IN TANK
        0400 8F   AA  0480          BICW    #TTY$M_TANK_HOLD,-      ; SHOW TANK EMPTY
        0108 C5       0484                  UCB$W_TT_HOLD(R5)
           FF70   31  0487          BRW     DZ11_OUT_LOOP
                      048A  ;
                      048A  ; STOP THE OUTPUT
                      048A  ;
                      048A  DZ11_STOP:
             03   8A  048A          BICB    #UCB$M_INT!UCB$M_TIM,-
          64 A5       048C                  UCB$W_STS(R5)           ; RESET OUTPUT ACTIVE
04 A0   0106 C5   AA  048E          BICW    UCB$W_TT_UNITBIT(R5),4(R0)      ; RESET THE OUTPUT ENABLE
           FF63   31  0494          BRW     DZ11_OUT_LOOP
                      0497
                      0497          .ENABLE LSB
                      0497  ;
                      0497  ; SEND XON OR XOFF CHARACTER
                      0497  ;
                      0497
                      0497  DZ11_PREMPT:
        0100 8F   AA  0497          BICW    #TTY$M_TANK_PREMPT,-    ; RESET XOFF STATE
        0108 C5       049B                  UCB$W_TT_HOLD(R5)
06 A0   010A C5   90  049E          MOVB    UCB$B_TT_PREMPT(R5),6(R0); OUTPUT CHARACTER
           FF53   31  04A4          BRW     DZ11_OUT_LOOP
                      04A7          .DISABLE        LSB
                      04A7
```

```
04A7  .IF NDF DZV
04A7  ;
04A7  ; DZ-32 OUTPUT INTERRUPT SERVICE CODE
04A7  ;
04A7
04A7  DZ32_OUT_EXIT:                                     ; EXIT OUTPUT INTERRUPT
04A7          ADDL    #4,SP                              ; REMOVE IDB ADDRESS
04A7          MOVQ    (SP)+,R0                           ; RESTORE REGISTERS
04A7          MOVQ    (SP)+,R2                           ;
04A7          MOVQ    (SP)+,R4                           ;
04A7          REI                                        ; DISMISS INTERRUPT
04A7
04A7  DZ32$INTOUT::                                      ; DZ-32 OUTPUT INTERRUPT SERVICE
04A7  DZ32_OUT_LOOP:
04A7          MOVL    @(SP),R4                           ; GET THE IDB ADDRESS
04A7          MOVL    (R4),R0                            ; GET THE CSR ADDRESS
04A7  ;
04A7  ; GET THE LINE INFO FROM THE CSR
04A7  ;
04A7
04A7          MOVW    (R0),R2                            ; GET THE CSR VALUE
04A7          BGEQ    DZ32_OUT_EXIT
04A7          ASHL    #-8,R2,R2                          ; GET THE LINE NUMBER
04A7          BICL    #^C<7>,R2                          ;
04A7          MOVL    IDB$L_UCBLST(R4)[R2],R5; GET THE UCB ADDRESS
04A7          BEQL    DZ32_OUT_LOOP                      ; IF EQL THEN DISMISS
04A7  ;
04A7  ;       CHECK FOR BURST ACTIVE ON LINE
04A7  ;
04A7          CMPB    #TTY$M_TANK_BURST@-8,-             ; ONLY BURST ACTIVE?
04A7                  UCB$W_TT_HOLD+1(R5)
04A7          BEQL    DZ32_BURST                         ; YES, CONTINUE BURST
04A7  ;
04A7  ;       LOOK FOR NEXT OUTPUT STATE IN TANK
04A7  ;
04A7
04A7          FFS     #0,#6,UCB$W_TT_HOLD+1(R5),R3
04A7          CASE    R3,TYPE=B,<=                               ; DISPATCH
04A7                  DZ32_PREMPT,-                              ; SEND PREMPT CHARACTERS
04A7                  DZ32_STOP,-                                ; STOP OUTPUT
04A7                  DZ32_CHAR,-                                ; CHAR IN TANK
04A7                  DZ32_BURST,-                               ; BURST IN PROGRESS
04A7                  >
04A7  ;
04A7  ; NO PENDING DATA - LOOK FOR NEXT CHARACTER
04A7  ;
04A7          BICB    #UCB$M_TIM!UCB$M_INT,UCB$W_STS(R5); CLEAR TIMEOUT AND EXPECTED
04A7  ;
04A7  ; CALL CLASS DRIVER FOR MORE OUTPUT
04A7  ;
04A7          JSB     @UCB$L_TT_GETNXT(R5)               ; GET THE NEXT CHARACTER
04A7          CASEB   UCB$B_TT_OUTYPE(R5),#-1,#1; OPTOMIZE FOR THE SINGLE
04A7                                                    ; CHARACTER CASE BY SETTING THE
04A7                                                    ; LIMIT TO 1
04A7  1$:     .WORD   DZ32_START_BURST-1$               ; BURST SPECIFIED
04A7          .WORD   50$-1$                            ; NONE
04A7  ;
```

```
04A7  ; OUTPUT A CHARACTER TO THE DZ-32
04A7  ;
04A7  20$:    MOVB    R3,6(R0)                ; OUTPUT CHARACTER
04A7          BRW     DZ32_OUT_LOOP
04A7
04A7  ;
04A7  ; DISABLE OUTPUT ON THIS LINE
04A7  ;
04A7  50$:
04A7          BBS     #UCB$V_INT,-            ; IF INT EXP, THEN DON'T RESET,
04A7                  UCB$W_STS(R5),DZ32_OUT_LOOP
04A7                                          ; COULD HAVE BEEN SET DURING CALLBACK
04A7
04A7          MOVL    @(SP),R4                ; GET IDB ADDRESS
04A7          BICB    UCB$W_TT_UNITBIT(R5),-  ; RESET THE OUTPUT ENABLE
04A7                  IDB$B_TT_ENABLE(R4)
04A7          MOVB    IDB$B_TT_ENABLE(R4),7(R0)
04A7          BRW     DZ32_OUT_LOOP
04A7
04A7
04A7  DZ32_START_BURST:
04A7          BISW    #TTY$M_TANK_BURST,-     ; SIGNAL BURST ACTIVE
04A7                  UCB$W_TT_HOLD(R5)
04A7  DZ32_BURST:
04A7          MOVB    @UCB$L_TT_OUTADR(R5),-  ; OUTPUT NEXT BYTE
04A7                  6(R0)
04A7          INCL    UCB$L_TT_OUTADR(R5)     ; UPDATE POINTER
04A7          DECW    UCB$W_TT_OUTLEN(R5)     ; UPDATE COUNT
04A7          BNEQ    60$                     ; NOT LAST CHARACTER
04A7          BICW    #TTY$M_TANK_BURST,-     ; RESET BURST ACTIVE
04A7                  UCB$W_TT_HOLD(R5)
04A7  60$:    BRW     DZ32_OUT_LOOP
04A7  ;
04A7  ; OUTPUT SINGLE CHARACTER
04A7  ;
04A7  DZ32_CHAR:
04A7          MOVB    UCB$W_TT_HOLD(R5),6(R0) ; OUTPUT CHAR IN TANK
04A7          BICW    #TTY$M_TANK_HOLD,-      ; SHOW TANK EMPTY
04A7                  UCB$W_TT_HOLD(R5)
04A7          BRW     DZ32_OUT_LOOP
04A7  ;
04A7  ; STOP OUTPUT
04A7  ;
04A7  DZ32_STOP:
04A7          BICB    #UCB$M_INT!UCB$M_TIM,-
04A7                  UCB$W_STS(R5)           ; RESET OUTPUT ACTIVE
04A7          MOVL    @(SP),R4                ; GET IDB ADDRESS
04A7          BICB    UCB$W_TT_UNITBIT(R5),-  ; RESET THE OUTPUT ENABLE
04A7                  IDB$B_TT_ENABLE(R4)
04A7          MOVB    IDB$B_TT_ENABLE(R4),7(R0)
04A7          BRW     DZ32_OUT_LOOP
04A7
04A7          .ENABLE LSB
04A7  ;
04A7  ; SEND XON OR XOFF
04A7  ;
04A7  DZ32_PREMPT:
```

```
04A7          BICW    #TTY$M_TANK_PREMPT,-      ; RESET XOFF STATE
                      UCB$W_TT_HOLD(R5)
04A7          MOVB    UCB$B_TT_PREMPT(R5),6(R0); OUTPUT CHARACTER
04A7          BRW     DZ32_OUT_LOOP
04A7
04A7
04A7          .DISABLE        LSB
04A7    .ENDC
```

```
                              04A7          .SBTTL  SET SPEED, PARITY PARAMETERS
                              04A7
                              04A7  ;++
                              04A7  ; DZ$SET_LINE - RESET SPEED, PARITY
                              04A7  ;
                              04A7  ; FUNCTIONAL DESCRIPTION:
                              04A7  ;
                              04A7  ; INPUTS:
                              04A7  ;
                              04A7  ;       R5 - UCB ADDRESS
                              04A7  ;
                              04A7  ; OUTPUTS:
                              04A7  ;
                              04A7  ;       R4 USED
                              04A7  ;--
                              04A7
                              04A7  DZ$SET_LINE:
        54    24 A5     D0    04A7          MOVL    UCB$L_CRB(R5),R4          ; ADDRESS CRB
                              04AB  ;
                              04AB  ;       SET UP LINE SPEED AND PARITY
                              04AB  ;
        54    2C B4     D0    04AB          MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4    ; GET THE CSR ADDRESS VIA CRB
                 7E     D4    04AF          CLRL    -(SP)                           ; RESET A TEMPORARY LOCATION
  01 AE     00F4 C5     01    83    04B1    SUBB3   #1,UCB$W_TT_SPEED(R5),1(SP)     ; ADJUST DATA BASE SPEED
        6E  00F8 C5     90    04B8          MOVB    UCB$B_TT_PARITY(R5),(SP); SET PARITY,STOP, CHARACTER SIZE
        6E  F007 8F     AA    04BD          BICW    #^XF007,(SP)              ; CLEAR SPECIAL FIELDS
                 07     E0    04C2          BBS     #UCB$V_TT_DSBL,-          ; SKIP CLOCK ENABLE IF LINE DISABLED
     05  012A C5              04C4                  UCB$B_TT_MAINT(R5),3$
        6E  1000 8F     A8    04C8          BISW    #<DZLPR$M_CLOCK>,(SP)
                              04CD  3$:
        6E    54 A5     A8    04CD          BISW    UCB$W_UNIT(R5),(SP)       ; SET LINE NUMBER
        64       01     B3    04D1          BITW    #DZCSR$M_MODE,(R4)        ; DZ32 CONTROLLER?
                 05     12    04D4          BNEQ    10$                       ; YES
                              04D6  5$:
     02 A4       8E     F7    04D6          CVTLW   (SP)+,2(R4)               ; INSERT AS LINE PARAMETER
                        05    04DA          RSB
                              04DB  ;
                              04DB  ;        HANDLE DZ-32 SPECIFIC FUNCTIONS
                              04DB  10$:
            00F4 C5     91    04DB          CMPB    UCB$W_TT_SPEED(R5),-      ; TRANSMIT/RECEIVE THE SAME
            00F5 C5           04DF                  UCB$W_TT_SPEED+1(R5)
                 F2     13    04E2          BEQL    5$                        ; YES, NO SPLIT SPEED
            00F5 C5     95    04E4          TSTB    UCB$W_TT_SPEED+1(R5)      ; RECEIVE SPEED SPECIFIED?
                 EC     13    04E8          BEQL    5$                        ; NO, NO SPLIT SPEED
                              04EA
                              04EA  ; SET SPLIT SPEED
                              04EA
        6E  2000 8F     A8    04EA          BISW    #DZLPR$M_SPLIT,(SP)       ; SET SPLIT SPEED BIT
                 E5     11    04EF          BRB     5$                        ; COMPLETE SETUP
                              04F1
```

```
                           04F1
                           04F1          .SBTTL  INITIALIZE DZ-11 MODEM POLLING
                           04F1
                           04F1  ;++
                           04F1  ; DZ$SET_MODEM - INIT MODEM POLLING
                           04F1  ;
                           04F1  ; FUNCTIONAL DESCRIPTION:
                           04F1  ;
                           04F1  ; INIT DZ-11 MODEM TRANSITION POLLING IF NOT ALREADY ACTIVE. LINK CRB
                           04F1  ; FOR CURRENT LINE INTO MODEM TRANSITION POLLING LIST
                           04F1  ;
                           04F1  ; INPUTS:
                           04F1  ;
                           04F1          R5 - UCB ADDRESS
                           04F1  ;
                           04F1  ; OUTPUTS:
                           04F1  ;
                           04F1          R0-R4 USED
                           04F1  ;--
                           04F1
                           04F1  DZ11$SET_MODEM:
         54    24 A5   D0  04F1          MOVL    UCB$L_CRB(R5),R4            ; ADDRESS CRB
          00000000'EF  D5  04F5          TSTL    DZ$L_DIALUP                 ; DZ-11 POLLING ALREADY ACTIVE?
                   18  12  04FB          BNEQ    5$                          ; YES, SKIP STARTUP
                   38  BB  04FD          PUSHR   #^M<R3,R4,R5>
     55   00000004'EF  DE  04FF          MOVAL   DZ$TIMQUENT,R5              ; ADDRESS OF TIMER ENTRY
         0B A5     06  90  0506          MOVB    #IPL$_QUEUEAST,TQE$B_RQTYPE(R5) ; SET FORK IPL
          00000513'EF  9F  050A          PUSHAB  4$                          ; RETURN ADDRESS
                 0021  31  0510          BRW     30$                         ; QUEUE FORK
                   38  BA  0513  4$:     POPR    #^M<R3,R4,R5>
                           0515  5$:
         53    18 A4   DE  0515          MOVAL   CRB$L_DZ_MODEM(R4),R3       ; ADDRESS OF DZ CRB THREAD
     51   00000000'EF  DE  0519          MOVAL   DZ$L_DIALUP,R1              ; ADDRESS OF DZ TIMER LIST HEAD
              52   51  D0  0520          MOVL    R1,R2
                           0523  ;
                           0523  ; LINK CRB INTO DZ-11 MODEM POLLER LIST IF NEEDED
                           0523  ;
                           0523  10$:
              53   62  D1  0523          CMPL    (R2),R3                     ; IS CRB ON LIST
                   0B  13  0526          BEQL    20$                         ; YES, DONE
              52   62  D0  0528          MOVL    (R2),R2                     ; POINT TO NEXT CRB
                   F6  12  052B          BNEQ    10$                         ; LOOK FOR NEXT
              63   61  D0  052D          MOVL    (R1),(R3)                   ; LINK CRB AT LIST HEAD
              61   53  D0  0530          MOVL    R3,(R1)                     ;
                           0533  20$:
                       05  0533          RSB
                           0534  30$:
          00000000'GF  16  0534          JSB     G^EXE$FORK                  ; FORK TO QUEUE TIMER ENTRY
                           053A          DSBINT  #IPL$_SYNCH
   0C A5      FCA7 CF  9E  0540          MOVAB   W^DZ$TIMER,TQE$L_FPC(R5); ADDRESS OF TIMER SERVICE ROUTINE
  20 A5   00000000'GF  D0  0546          MOVL    G^TTY$GL_DELTA,TQE$Q_DELTA(R5)
                           054E                                              ; INTERVAL IS SYSGEN PARAMETER
         0B A5     05  90  054E          MOVB    #TQE$C_SSREPT,TQE$B_RQTYPE(R5)
     50   00000000'GF  7D  0552          MOVQ    G^EXE$GQ_SYSTIME,R0
     50   00000000'GF  C0  0559          ADDL    G^TTY$GL_DELTA,R0
              51   00  D8  0560          ADWC    #0,R1
          00000000'GF  16  0563          JSB     G^EXE$INSTIMQ               ; INSERT INTO TIMER QUEUE
```

```
    0569  1456          ENBINT                          ; RESTORE IPL
05  056C  1457          RSB
    056D  1458
```

```
                    056D
            00000000 0000         .PSECT  $$$117_DATA,QUAD
                     0000 ;
                     0000 ;       DZ-11 TIMER DATA STRUCTURES FOR DETECTION OF DZ-11 MODEM TRANSITIONS
                     0000 ;
                     0000
                     0000 DZ$L_DIALUP:
   00000000          0000                 .LONG   0                       ; LINKED LIST OF DZ-11 CRB'S
                     0004                                                 ;  USING MODEM CONTROL
                     0004 DZ$TIMQUENT:                                    ; TIMER QUE ENTRY USED TO
                     0004                                                 ; SAMPLE DZ-11 MODEM SIGNALS
                     0004                                                 ; ON PERIODIC BASIS
                     0004         .ALIGN  QUAD
   00000038          0008         .BLKB   TQE$C_LENGTH
                     0038         STO_TQE TQE$W_SIZE,WORD,TQE$C_LENGTH,DZ$TIMQUENT
                     0038         STO_TQE TQE$B_TYPE,BYTE,DYN$C_TQE,DZ$TIMQUENT
                     0038         STO_TQE TQE$B_RQTYPE,BYTE,TQE$C_SSREPT,DZ$TIMQUENT
                     0038
                     0038
                     0038 DZ$END:                               ; End of driver
                     0038
                     0038         .END
```

```
$$$                      = 00000020 R     02     DZ11$INTINP                  0000028E RG    03
$$$$$$                   = 00000038 R     04     DZ11$INTOUT                  000003FA RG    03
$$OP                     = 00000002              DZ11$MAINT                   0000018A R     03
AT$_UBA                  = 00000001              DZ11$RESUME                  0000038E R     03
BIT...                   = 00000010              DZ11$SET_MODEM               000004F1 R     03
CLASS_DDT                = 00000010              DZ11$STARTIO                 0000031C RG    03
CLASS_DS_TRAN            = 0000000C              DZ11$VEC                     00000038 R     03
CLASS_GETNXT             = 00000000              DZ11$VECEND                  00000070 R     03
CLASS_POWERFAIL          = 00000020              DZ11$XOFF                    00000342 R     03
CLASS_PUTNXT             = 00000004              DZ11$XON                     00000342 R     03
CLASS_READERROR          = 00000014              DZ11_BURST                   00000460 R     03
CLASS_SETUP_UCB          = 00000008              DZ11_CHAR                    0000047A R     03
CRB$B_DZ_CARRIER         = 0000001D              DZ11_OUT_EXIT                000003ED R     03
CRB$B_DZ_DTR             = 0000001E              DZ11_OUT_LOOP                000003FA R     03
CRB$B_DZ_RING            = 0000001C              DZ11_PREMPT                  00000497 R     03
CRB$B_TT_TYPE            = 0000000B              DZ11_START_BURST             00000459 R     03
CRB$L_DZ_MODEM           = 00000018              DZ11_STOP                    0000048A R     03
CRB$L_INTD               = 00000024              DZCSR$M_CLEAR              = 00000010
CRB$L_INTD2              = 00000048              DZCSR$M_DS_CHG             = 00000800
DC$_TERM                 = 00000042              DZCSR$M_DS_ENAB            = 00000002
DDB$L_DDT                = 0000000C              DZCSR$M_LINE               = 00000700
DEV$M_AVL                = 00040000              DZCSR$M_MAINT              = 00000008
DEV$M_CCL                = 00000002              DZCSR$M_MASTENAB           = 00000020
DEV$M_IDV                = 04000000              DZCSR$M_MODE               = 00000001
DEV$M_NNM                = 00000200              DZCSR$M_RCVINT             = 00000040
DEV$M_ODV                = 08000000              DZCSR$M_RCVRDY             = 00000080
DEV$M_REC                = 00000001              DZCSR$M_SNDINT             = 00004000
DEV$M_TRM                = 00000004              DZCSR$M_SNDRDY             = 00008000
DPT$C_LENGTH             = 00000038              DZCSR$S_CLEAR              = 00000001
DPT$C_VERSION            = 00000004              DZCSR$S_DS_CHG             = 00000001
DPT$INITAB                 00000038 R     02     DZCSR$S_DS_ENAB            = 00000001
DPT$M_NOUNLOAD           = 00000004              DZCSR$S_LINE               = 00000003
DPT$REINITAB               000000D3 R     02     DZCSR$S_MAINT              = 00000001
DPT$TAB                    00000000 R     02     DZCSR$S_MASTENAB           = 00000001
DPT$W_VECTOR             = 0000001E              DZCSR$S_MODE               = 00000001
DT$_DZ11                 = 00000042              DZCSR$S_RCVINT             = 00000001
DYN$C_CRB                = 00000005              DZCSR$S_RCVRDY             = 00000001
DYN$C_DDB                = 00000006              DZCSR$S_SNDINT             = 00000001
DYN$C_DPT                = 0000001E              DZCSR$S_SNDRDY             = 00000001
DYN$C_ORB                = 00000049              DZCSR$V_CLEAR              = 00000004
DYN$C_TQE                = 0000000F              DZCSR$V_DS_CHG             = 0000000B
DYN$C_UCB                = 00000010              DZCSR$V_DS_ENAB            = 00000001
DZ$ABORT                   00000374 R     03     DZCSR$V_LINE               = 00000008
DZ$CTRL_ERROR              000000F4 R     03     DZCSR$V_MAINT              = 00000003
DZ$DDT                     00000000 RG    03     DZCSR$V_MASTENAB           = 00000005
DZ$DPT                     00000000 RG    02     DZCSR$V_MODE               = 00000000
DZ$END                     00000038 R     04     DZCSR$V_RCVINT             = 00000006
DZ$INITIAL                 00000075 RG    03     DZCSR$V_RCVRDY             = 00000007
DZ$INITLINE                000000F5 RG    03     DZCSR$V_SNDINT             = 0000000E
DZ$L_DIALUP                00000000 R     04     DZCSR$V_SNDRDY             = 0000000F
DZ$NULL                    00000074 R     03     DZLCS1$M_ACK               = 00008000
DZ$SET_LINE                000004A7 R     03     DZLCS1$S_ACK               = 00000001
DZ$STOP                    0000036C R     03     DZLCS1$V_ACK               = 0000000F
DZ$TIMER                   000001EB R     03     DZLPR$M_CLOCK              = 00001000
DZ$TIMQUENT                00000004 R     04     DZLPR$M_LINE               = 00000007
DZ$UNIT_ERROR              00000185 R     03     DZLPR$M_ODD                = 00000080
DZ11$DS_SET                000001B3 R     03     DZLPR$M_PARITY             = 00000040
```

```
DZLPR$M_SIZE            = 00000018                ORB$M_PROT_16           = 00000001
DZLPR$M_SPEED           = 00000F00                ORB$W_PROT              = 00000018
DZLPR$M_SPLIT           = 00002000                PORT_ABORT              = 00000020
DZLPR$M_STOP            = 00000020                PORT_DS_SET             = 0000000C
DZLPR$S_CLOCK           = 00000001                PORT_LENGTH             = 00000038
DZLPR$S_LINE            = 00000003                PORT_MAINT              = 00000030
DZLPR$S_ODD             = 00000001                PORT_RESUME             = 00000024
DZLPR$S_PARITY          = 00000001                PORT_SET_LINE           = 00000008
DZLPR$S_SIZE            = 00000002                PORT_SET_MODEM          = 00000028
DZLPR$S_SPEED           = 00000004                PORT_STARTIO            = 00000000
DZLPR$S_SPLIT           = 00000001                PORT_STOP               = 00000018
DZLPR$S_STOP            = 00000001                PORT_VECTOR               00000038 R      03
DZLPR$V_CLOCK           = 0000000C                PORT_XOFF               = 00000014
DZLPR$V_LINE            = 00000000                PORT_XON                = 00000010
DZLPR$V_ODD             = 00000007                PR$_IPL                 = 00000012
DZLPR$V_PARITY          = 00000006                SIZ...                  = 00000001
DZLPR$V_SIZE            = 00000003                SS$_NORMAL              = 00000001
DZLPR$V_SPEED           = 00000008                TQE$B_RQTYPE            = 0000000B
DZLPR$V_SPLIT           = 0000000D                TQE$B_TYPE              = 0000000A
DZLPR$V_STOP            = 00000005                TQE$C_LENGTH            = 00000030
DZRCV$M_BUF             = 000000FF                TQE$C_SSREPT            = 00000005
DZRCV$M_FRAMER          = 00002000                TQE$L_FPC               = 0000000C
DZRCV$M_LINE            = 00000700                TQE$Q_DELTA             = 00000020
DZRCV$M_OVERRUN         = 00004000                TQE$W_SIZE              = 00000008
DZRCV$M_PARERR          = 00001000                TT$M_DS_CTS             = 00000010
DZRCV$M_VALID           = 00008000                TT$M_DS_DSR             = 00000080
DZRCV$S_BUF             = 00000008                TT$V_DS_CARRIER         = 00000005
DZRCV$S_FRAMER          = 00000001                TT$V_DS_DTR             = 00000001
DZRCV$S_LINE            = 00000003                TT$V_DS_RING            = 00000006
DZRCV$S_OVERRUN         = 00000001                TT$V_MODEM              = 00000015
DZRCV$S_PARERR          = 00000001                TT$_UNKNOWN             = 00000000
DZRCV$S_VALID           = 00000001                TTY$GB_DEFSPEED           ********   X  02
DZRCV$V_BUF             = 00000000                TTY$GB_PARITY             ********   X  02
DZRCV$V_FRAMER          = 0000000D                TTY$GB_RSPEED             ********   X  02
DZRCV$V_LINE            = 00000008                TTY$GL_DEFCHAR            ********   X  02
DZRCV$V_OVERRUN         = 0000000E                TTY$GL_DEFCHAR2           ********   X  02
DZRCV$V_PARERR          = 0000000C                TTY$GL_DELTA              ********   X  03
DZRCV$V_VALID           = 0000000F                TTY$GL_DPT                ********   X  03
DZV                     = 00000001                TTY$GL_OWNUIC             ********   X  02
EXE$FORK                  ********   X  03        TTY$GW_DEFBUF             ********   X  02
EXE$GL_ABSTIM             ********   X  03        TTY$GW_PROT               ********   X  02
EXE$GL_TENUSEC            ********   X  03        TTY$M_TANK_BURST        = 00000800
EXE$GL_UBDELAY            ********   X  03        TTY$M_TANK_HOLD         = 00000400
EXE$GQ_SYSTIME            ********   X  03        TTY$M_TANK_PREMPT       = 00000100
EXE$INSTIMQ               ********   X  03        TTY$M_TANK_STOP         = 00000200
FUNCTAB_LEN             = 00000000                TTY$V_PC_NOTIME         = 00000000
IDB$L_UCBLST            = 00000018                TTY$V_TANK_BURST        = 0000000B
IO$M_LINE_OFF           = 00000200                UCB$B_DEVCLASS          = 00000040
IO$M_LINE_ON            = 00000800                UCB$B_DEVTYPE           = 00000041
IOC$MNTVER                ********   X  03        UCB$B_DIPL              = 0000005E
IOC$RETURN                ********   X  03        UCB$B_FIPL              = 0000000B
IPL$_QUEUEAST           = 00000006                UCB$B_TT_DEPARI         = 000000EC
IPL$_SYNCH              = 00000008                UCB$B_TT_DETYPE         = 000000F0
MODEM$C_DATASET         = 00000003                UCB$B_TT_DS_RCV         = 00000124
MODEM$C_INIT            = 00000000                UCB$B_TT_DS_TX          = 00000125
ORB$B_FLAGS             = 0000000B                UCB$B_TT_MAINT          = 0000012A
ORB$L_OWNER             = 00000000                UCB$B_TT_OUTYPE         = 0000010B
```

```
UCB$B_TT_PARITY                 = 000000F8
UCB$B_TT_PREMPT                 = 0000010A
UCB$C_TT_LENGTH                 = 00000134
UCB$L_CRB                       = 00000024
UCB$L_DDB                       = 00000028
UCB$L_DDT                       = 00000088
UCB$L_DEVCHAR                   = 00000038
UCB$L_DEVCHAR2                  = 0000003C
UCB$L_DEVDEPEND                 = 00000044
UCB$L_DEVDEPND2                 = 00000048
UCB$L_DUETIM                    = 0000006C
UCB$L_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$M_TT_DSBL                   = 00000080
UCB$V_INT                       = 00000001
UCB$V_POWER                     = 00000005
UCB$V_TT_DSBL                   = 00000007
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$L_IDB                       = 00000008
VEC$L_INITIAL                   = 0000000C
VEC$L_UNITINIT                  = 00000018

                                     +----------------+
                                     ! 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
$$$105_PROLOGUE                 000000DE  (  222.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$115_DRIVER                   0000056D  ( 1389.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG
$$$117_DATA                     00000038  (   56.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC QUAD
```

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

Phase                        Page faults   CPU Time       Elapsed Time
-----                        -----------   --------       ------------
Initialization                        35   00:00:00.05    00:00:00.24
Command processing                   116   00:00:00.46    00:00:01.39
Pass 1                               696   00:00:22.09    00:00:44.74
Symbol table sort                      0   00:00:03.25    00:00:07.03
Pass 2                               249   00:00:04.45    00:00:09.34
Symbol table output                   31   00:00:00.21    00:00:00.43
Psect synopsis output                  2   00:00:00.02    00:00:00.02
Cross-reference output                 0   00:00:00.00    00:00:00.00
Assembler run totals                1131   00:00:30.53    00:01:03.19

The working set limit was 2250 pages.
178134 bytes (348 pages) of virtual memory were used to buffer the intermediate code.
There were 160 pages of symbol table space allocated to hold 3025 non-local and 60 local symbols.
1482 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                                 33
_$255$DUA28:[SYSLIB]STARLET.MLB;2                              12
TOTALS (all libraries)                                         45

3540 GETS were required to define 45 macros.

There were no errors, warnings or information messages.

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

0402 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DZVDRIVER
MAP

DZDRIVER
LIS

TBKSYM
LIS

YFDRIVER
MAP

DZVDRIVER
LIS

TBKSSV
LIS

TTDRVR

DZDRIVER
MAP

TTDRIVER
MAP

YCDRIVER
MAP

TBKSTO
LIS

TTYDEF
SDL

TTYMACS
MAR

TBKSTART
LIS

0403 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TTYFDT
LIS

TTYCHARO
LIS

TTYCHARI
LIS

TTYDRVDAT
LIS