F 10

```
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMMMMM   MMMMMM  PPP         PPP
MMMMMM   MMMMMM  PPP         PPP
MMMMMM   MMMMMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM   MMM   MMM  PPP         PPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPPPPPPPPPPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
MMM         MMM  PPP
```

```
**FILE**ID**XDELTA                                                    M  5

XX      XX  DDDDDDDD    EEEEEEEEEE  LL          TTTTTTTTTT    AAAAAA
XX      XX  DDDDDDDD    EEEEEEEEEE  LL          TTTTTTTTTT    AAAAAA
XX      XX  DD      DD  EE          LL              TT      AA      AA
XX      XX  DD      DD  EE          LL              TT      AA      AA
  XX  XX    DD      DD  EE          LL              TT      AA      AA
  XX  XX    DD      DD  EE          LL              TT      AA      AA
    XX      DD      DD  EEEEEEEE    LL              TT      AA      AA
    XX      DD      DD  EEEEEEEE    LL              TT      AA      AA
  XX  XX    DD      DD  EE          LL              TT      AAAAAAAAAA
  XX  XX    DD      DD  EE          LL              TT      AAAAAAAAAA
XX      XX  DD      DD  EE          LL              TT      AA      AA       ....
XX      XX  DD      DD  EE          LL              TT      AA      AA       ....
XX      XX  DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL      TT      AA      AA       ....
XX      XX  DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL      TT      AA      AA       ....

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
```

XD
VO

Table of contents

```
0000    ;
0000    ; Version:      'V04-000'
0000    ;
0000
0000            .MCALL  MFPR
0000            .IF     DF,SW_PROCESS                     ;
0000            .TITLE  DELTA - MULTIMODE PROCESS DEBUGGER
0000            .IFF                                      ;
0000            .TITLE  XDELTA - EXECUTIVE DEBUGGER
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: EXECUTIVE, DEBUGGING TOOLS
0000    ;
0000    ; ABSTRACT:
0000    ;       THIS MODULE PRODUCES TWO DIFFERENT DEBUGGERS DEPENDING ON THE SETTING
0000    ;       OF THE ASSEMBLY SWITCH, SW_PROCESS.  DELTA IS A MULTIMODE PROCESS
0000    ;       DEBUGGER USING SYSTEM SERVICES WHILE XDELTA IS A STANDALONE EXEC
0000    ;       DEBUGGING TOOL.
0000    ;
0000    ;       COMMAND SYNTAX IS IDENTICAL FOR BOTH VERSIONS EXCEPT FOR ENVIRONMENTAL
0000    ;       DIFFERENCES.  THE SYNTAX IS QUITE TERSE AND SOMEWHAT CRYPTIC AND
0000    ;       IS DOCUMENTED IN THE ''GUIDE TO WRITING AN I/O DRIVER''.
0000    ;
0000    ; ENVIRONMENT:
0000    ;       DELTA - NORMAL PROCESS ENVIRONMENT, VARIOUS ACCESS MODES.
0000    ;       XDELTA - STANDALONE, RESIDENT, KERNEL MODE, IPL=31
0000    ;       BOTH VERSIONS MUST BE POSITION INDEPENDENT - BEWARE!
0000    ;--
```

```
                  0000          .SBTTL  HISTORY                 ; DETAILED
                  0000  ;
                  0000  ; AUTHOR:       R. HUSTVEDT     CREATION DATE: 15-NOV-76
                  0000  ;
                  0000  ; REVISION HISTORY:
                  0000  ;
                  0000  ;       V02-009 LJK0030         Lawrence J. Kenah       31-Jul-1981
                  0000  ;               Make changes necessary to support large physical memory
                  0000  ;               configurations. Change names of PFN listhead cells. Add
                  0000  ;               labels to cells for XE and XF stored strings to allow
                  0000  ;               access from INIT.
                  0000  ;
                  0000  ;       V02-008 TCM0001         Trudy C. Matthews       29-Jul-1981
                  0000  ;               Change all ''7ZZ''s to ''730''s.
                  0000  ;
                  0000  ;       V02-007 KDM0003         Kathleen D. Morse       15-Sep-1980
                  0000  ;               Make changes to run with multi-processor privileged program.
                  0000  ;
```

```
                0000                .SBTTL  DECLARATIONS
                0000
                0000        ;
                0000        ; INCLUDE FILES:
                0000        ;
                0000                $ACBDEF                               ; DEFINE AST CONTROL BLOCK
                0000                $CADEF                                ; DEFINE ASSEMBLY SWITCHES
                0000                $CLIDEF                               ; DEFINE CLI VALUES
                0000                $IPLDEF                               ; DEFINE IPL VALUES
                0000                $IRPDEF                               ; DEFINE IRP VALUES
                0000                $PCBDEF                               ; DEFINE PROCESS CONTROL BLOCK
                0000                $PRDEF                                ; DEFINE PROCESSOR REGISTERS
                0000                $PRIDEF                               ; DEFINE PRIORITY INCREMENT CLASSES
                0000                $PRTDEF                               ; DEFINE PROTECTION VALUES
                0000                $PSLDEF                               ; DEFINE PSL FIELDS
                0000                $SSDEF                                ; DEFINE SYSTEM SERVICE STATUS CODES
                0000
                0000        ;
                0000        ; MACROS:
                0000        ;
                0000
                0000        ;
                0000        ; CPU TYPE DISPATCH MACRO:
                0000        ;
                0000        ;       THE ADDRESSES IN THE ADDRESS LIST ARE:
                0000        ;               -ADDRESS OF CODE FOR CPU TYPE=1 (11/780)
                0000        ;               -ADDRESS OF CODE FOR CPU TYPE=2 (11/750)
                0000        ;               -ADDRESS OF CODE FOR CPU TYPE=3 (11/730)
                0000        ;               -ADDRESS OF CODE FOR CPU TYPE=4 (?)
                0000        ;               -ETC.
                0000        ;
                0000        ;       CPUDISP IN INVOKED TO HANDLE CPU DIFFERENCES IN LINE.  WHEN THE
                0000        ;       NEXT CPU IS ADDED, ALL OCCURRENCES OF CPUDISP MUST BE EXPANDED
                0000        ;       TO HANDLE FOUR CPU SPECIFIC PATHS.
                0000        ;
                0000                .MACRO  CPUDISP,ADDRLIST
                0000                CASE    G^EXE$GB_CPUTYPE,<ADDRLIST>,LIMIT=#PR$_SID_TYP780,TYPE=B
                0000                .ENDM   CPUDISP
                0000
                0000        ;
                0000        ; EQUATED SYMBOLS:
                0000        ;
00000008        0000        V_F1=8                                        ; FIELD 1 PRESENT FLAG
00000009        0000        V_F2=9                                        ; FIELD 2 PRESENT FLAG
0000000A        0000        V_F3=10                                       ; FIELD 3 PRESENT FLAG
0000000B        0000        V_F4=11                                       ; FIELD 4 PRESENT FLAG
0000000C        0000        V_F5=12                                       ; FIELD 5 PRESENT FLAG
                0000
00000000        0000        V_OPEN=0                                      ; OPEN CELL FLAG
00000001        0000        V_ASCII=1                                     ; ASCII
00000002        0000        V_INFIELD=2                                   ; FIELD IN PROGRESS
00000003        0000        V_TBIT=3                                      ; ENABLE TBIT
00000004        0000        V_ATBRK=4                                     ; AT BREAKPOINT
00000005        0000        V_TBITOK=5                                    ; TBIT EXPECTED
00000006        0000        V_RUB=6                                       ; RUBOUT IN PROGRESS
00000007        0000        V_NEGATE=7                                    ; NEGATE BIT
0000000F        0000        V_PRMODE=15                                   ; PROCESSOR REGISTER MODE
```

```
0000001F  0000  V_PREG=31                                 ; PROCESSOR REGISTER FLAG
          0000
00000000  0000  RDCR=0                                    ; READ CSR
00000002  0000  RDBUF=2                                   ; READ BUFFER
00000004  0000  OUTCR=4                                   ; OUTPUT CSR
00000006  0000  OUTB=6                                    ; OUTPUT BUFFER
          0000
0000005C  0000  BSLSH=92                                  ; BACK SLASH CODE
0000000D  0000  CR=13                                     ; CARRIAGE RETURN
0000000A  0000  LF=10                                     ; LINE FEED
00000027  0000  QUOT=39                                   ; QUOTE
0000007F  0000  RUBOUT=127                                ; RUBOUT CODE
0000002F  0000  SLSH=47                                   ; SLASH CODE
          0000
          0000
          0000
          0000  ;
          0000  ;         OWN STORAGE:
          0000  ;
          0000            .LIST   MEB                     ; DISPLAY MACRO EXPANSIONS
          0000            .IF     DF,SW_PROCESS
          0000            .PSECT  _DELTA,LONG
          0000  DELBASE:.LONG     DELBASE-DELBASE         ; RELATIVE PAGE NUMBER OF WRITABLE
          0000            .LONG   <511+DELEND-DELBASE>&^C511; REL PAGE NUMBER OF END OF WRITABLE
          0000            .LONG   DELTA_START-DELBASE     ; START ADDRESS
          0000            .IFF
    00000000  0000        .PSECT  Z$DEBUGXDELTA,LONG
          0000            .ENDC
          0000  CONTEXT:                                  ;
00000000  0000            .LONG   0                       ; BUFFER PADDING
00000034  0004  INBUF:    .BLKB   48                      ; INPUT BUFFER
00000000  0034  STATUS:   .LONG   0                       ; STATUS FLAGS
00000000  0038  F1:       .LONG   0                       ; FIELDS
00000000  003C  F2:       .LONG   0                       ; 0-7
00000000  0040  F3:       .LONG   0                       ;
00000000  0044  F4:       .LONG   0                       ;
00000000  0048  F5:       .LONG   0                       ;
          004C
00000000  004C  MFYFLG:   .LONG   0                       ; MODIFY ENABLE FLAG FOR OTHER PROCESS
          0050                                            ; ADDRESS SPACES
00000000  0050  PID:      .LONG   0                       ; PID FOR ADDRESS SPACE 0=>SELF
          0054
      00  0054  FCTR:     .BYTE   0                       ; FIELD COUNTER
          0055
      02  0055  DTYPE:    .BYTE   2                       ; DATA TYPE
      02  0056  CURTYPE:.BYTE     2                       ; CURRENT TYPE
          0057
      00  0057  OPER:     .BYTE   0                       ; OPERATOR
          0058  B:                                        ; BASE OF DATA AREA(CENTER)
00000000  0058  CURDOT:   .LONG   0                       ; CURRENT LOCATION
00000000  005C  QUAN:     .LONG   0                       ; QUANTITY (;Q)
00000070  0060  OUTBUF:   .BLKL   4                       ; OUTPUT BUFFER
          0070  ;
          0070  ;         REGISTER SAVE AREA
          0070  ;
          0070  SAVREG:                                   ; REGISTER SAVE AREA
00000074  0070            .BLKL   1                       ; R0
```

```
00000078  0074            .BLKL   1                            ; R1
0000007C  0078  SAVR2:    .BLKL   1                            ; R2
00000080  007C            .BLKL   1                            ; R3
00000084  0080            .BLKL   1                            ; R4
00000088  0084            .BLKL   1                            ; R5
0000008C  0088            .BLKL   1                            ; R6
00000090  008C            .BLKL   1                            ; R7
00000094  0090            .BLKL   1                            ; R8
00000098  0094            .BLKL   1                            ; R9
0000009C  0098            .BLKL   1                            ; R10
000000A0  009C            .BLKL   1                            ; R11
000000A4  00A0  SAVAP:    .BLKL   1                            ; AP
000000A8  00A4            .BLKL   1                            ; (FP)
000000AC  00A8  SAVSP:    .BLKL   1                            ; SP
000000B0  00AC  SAVPC:    .BLKL   1                            ; PC
000000B4  00B0  SAVPSL:   .BLKL   1                            ; PSL
000000B6  00B4  SAVOCR:   .BLKW   1                            ; OUTPUT CSR SAVE
000000B8  00B6  SAVRCR:   .BLKW   1                            ; INPUT CSR SAVE
          00B8  ASTEN:                                         ; AST ENABLE SAVE LOCATION
000000BC  00B8  SAVRXCS:.BLKL     1                            ; CONSOLE RECEIVER STATUS
          00BC
000000BC  00BC  CONTEXTSZ=.-CONTEXT                            ; SIZE OF PER MODE CONTEXT AREA
          00BC  ;
          00BC  ;         RESERVE SPACE FOR MULTIPLE MODE CONTEXT AREA
          00BC  ;
          00BC            .IF     DF,SW_PROCESS                ;
          00BC            .REPT   3
          00BC            .BLKB   CONTEXTSZ                    ; FOR EXEC,SUPER AND USER
          00BC  SAV...= .
          00BC            .=.-CONTEXTSZ+<DTYPE-CONTEXT>        ; POINT AT DTYPE,CURTYP
          00BC            .BYTE   2,2                          ; SET TYPE TO LONGWORD
          00BC            .=SAV...                             ; RESTORE LOCATION COUNTER
          00BC            .ENDR                                ;
          00BC            .ENDC                                ;
          00BC
          00BC
          00BC  ;
          00BC  ;         BREAK POINT DATA
          00BC  ;
000000B8  00BC  BRKADR=.-4
          00BC            .IF     NDF,SW_PROCESS   ;
          00BC  XDELIBRK::                         ;
00000000' 00BC            .LONG   INI$BRK                      ; ADDRESS OF INITIAL BREAKPOINT
          00C0            .IFF                                 ; FOR PROCESS VERSION
          00C0  INIBRKA:.LONG     0                            ; INITIAL BREAKPOINT
          00C0            .ENDC                                ;
000000DC  00C0            .BLKL   7                            ; OTHER BREAK POINT ADDRESSES
00000008  00DC  NBRK=<.-4-BRKADR>/4                            ; NUMBER OF BREAKPOINTS
000000DB  00DC  BRKOP=.-1                                      ; SAVED OPCODE
      01  00DC            NOP                                  ; INITIAL OPCODE
000000E4  00DD            .BLKB   7                            ; REMAINING OPCODES
          00E4
          00E4
000000E0  00E4  BRKDSP=.-4
00000104  00E4            .BLKL   8                            ; DISPLAY LOCATION START
00000100  0104  BRKCOM=.-4
00000124  0104            .BLKL   8                            ; COMMAND START
```

```
                0124
     00000130   0124   XREGV:  .BLKL   3                             ; X REGISTER VECTOR
                0130   XDEL_LOADBASE::                               ; BASE OF LOADABLE
     00000000   0130           .LONG   0                             ;  CPU DEPENDENT CODE
     00000000'  0134           .LONG   SCH$GL_CURPCB                 ; X4 = CURRENT PCB ADDRESS
     00000000'  0138           .LONG   SCH$GL_PCBVEC                 ; X5 = BASE OF PCB VECTOR
                013C           .IF     NDF,SW_PROCESS                ;
     00000000'  013C           .LONG   PFN$AW_SWPVBN                 ; X6 = SWAP VBN
     00000000'  0140           .LONG   PFN$AL_PTE                    ; X7 = PTE BACK POINTER
     00000000'  0144           .LONG   PFN$AL_BAK                    ; X8 = BACKUP ADDRESS
     00000000'  0148           .LONG   PFN$AW_REFCNT                 ; X9 = REFERENCE COUNT
     00000000'  014C           .LONG   PFN$Ax_FLINK                  ; XA = FORWARD LINK
     00000000'  0150           .LONG   PFN$Ax_BLINK                  ; XB = BACK LINK
     00000000'  0154           .LONG   PFN$AB_STATE                  ; XC = STATE
     00000000'  0158           .LONG   PFN$AB_TYPE                   ; XD = TYPE
                015C   XDS$GL_XESTRING::
      0000000'  015C           .LONG   XDS$GT_WORD_PFN               ; XE;E WITH X0 = PFN , DEFAULT TO WORD ARRAY
                0160   XDS$GL_XFSTRING::
     00000000'  0160           .LONG   XDS$GT_WORD_PFN               ; XF;E WITH R0 = PFN , DEFAULT TO WORD ARRAY
     00000168   0164   MCHKSAV:.BLKL   1                             ; SAVED CONTENT OF MACHINE CHECK VECTOR
                0168           .IFF                                  ; FOR PROCESS VERSION
                0168           .BLKL   10
                0168   TTIOSB: .BLKL   2                             ; IO STATUS BLOCK FOR TERMINAL READ
                0168   TTCHAN: .BLKL   1                             ; CHANNEL NUMBER
                0168   TTNAMD: .LONG   2,TTSTR                       ; ACTUAL ADDRESS FOR DESCR SET BY INIT
                0168   TTSTR:  .ASCII  /TT/
                0168   DBGACTIVE:                                    ; ACTIVE FLAGS BY ACCESS MODE
                0168           .LONG   0
                0168   EXITBLK:                                      ; EXIT HANDLER BLOCK
                0168           .LONG   0
                0168   EXIHADR:.LONG   EXIHANDLE                     ; EXIT HANDLER
                0168           .LONG   1                             ; ARGUMENT COUNT
                0168   EXCODA: .LONG   EXITCODE                      ; ADDRESS TO STORE EXIT CODE
                0168   EXITCODE:
                0168           .LONG   1                             ; RECEIVER FOR EXIT CODE
                0168   KCOND:  .LONG   0                             ; PREVIOUS KERNEL HANDLER
                0168   ECOND:  .LONG   0                             ; PREVIOUS EXEC HANDLER
                0168   SCOND:  .LONG   0                             ; PREVIOUS SUPER HANDLER
                0168   TERMASKD:                                     ; TERMINATOR MASK DESCRIPTOR
                0168           .LONG   16                            ; MASK LENGTH
                0168           .LONG   TERMASK                       ; MASK ADDRESS
                0168   TERMASK:.LONG   <1@9>!<1@10>!<1@13>!<1@27>          ; TAB,LF,CR,ESC
                0168           .LONG   <1@2>!<1@15>!<1@29>           ; DOUBLE QUOTE,SLASH,EQUALS
                0168           .LONG   <1@19>                        ; 'S'
                0168           .LONG   0                             ;
                0168                                                 ;
                0168           .ENDC                                 ;
```

```
                                 0168       .SBTTL  PRIMARY COMMAND CHARACTER SWITCH
                                 0168
                                 0168  ;
                                 0168  ;          PRIMARY CHARACTER LIST
                                 0168  ;
                                 0168  PRIMARY:                                    ;
42 41 39 38 37 36 35 34 33 32 31 30 0168    .ASCII  /0123456789ABCDEF/   ; DECIMAL AND HEX CHARS
                       46 45 44 43 0174
                                2E 0178     .ASCII  /./                  ; DOT - CURRENT LOCATION
                                2C 0179     .ASCII  /,/                  ; COMMA - FIELD SEPARATOR
                          00000012 017A  OPERBAS=.-PRIMARY                ; OPERATORS
                                2B 017A     .ASCII  /+/                  ; PLUS - ADD
                                20 017B     .ASCII  / /                  ; BLANK - SAME AS PLUS
                                40 017C     .ASCII  /@/                  ; SHIFT OPERATOR
                                2A 017D     .ASCII  /*/                  ; MULTIPLY OPERATOR
                                25 017E     .ASCII  /%/                  ; DIVIDE OPERATOR
                                2D 017F     .ASCII  /-/                  ; MINUS - SUBTRACT OPERATOR
                                5B 0180     .ASCII  /[/                  ; LBRACKET - LEFT BRACKET
                                   0181  TERM:                            ; BASE OF TERMINATOR LIST
                                09 0181     .ASCII  <9>                  ; TAB - INDIRECT
                                0A 0182     .ASCII  <10>                 ; LINEFEED -
                                0D 0183     .ASCII  <CR>                 ; RETURN -
                                2F 0184     .ASCII  '/'                  ; SLASH - OPEN FOR DISPLAY
                                22 0185     .ASCII  '"'                  ; DOUBLE QUOTE - OPEN FOR ASCII DISPLAY
                                3D 0186     .ASCII  /=/                  ; EQUALS - DISPLAY
                                1B 0187     .ASCII  <27>                 ; ESCAPE - PREVIOUS LOCATION
                                53 0188     .ASCII  /S/                  ; STEP
                          00000008 0189  NTERM=.-TERM                     ; NUMBER OF TERMINATORS
                                3B 0189     .ASCII  <59>                 ; SEMI - INITIATE SECONDARY
                                3A 018A     .ASCII  /:/                  ; COLON - SEPARATE PID FORM ADDRESS
                                50 018B     .ASCII  /P/                  ; P - PROCESSOR REGISTER PREFIX
                                51 018C     .ASCII  /Q/                  ; Q - LAST QUANTITY
                                27 018D     .ASCII  /'/                  ; QUOTE - BEGIN CHAR STRING
                                52 018E     .ASCII  /R/                  ; REGISTER PREFIX
                                47 018F     .ASCII  /G/                  ; G - GLOBAL PREFIX
                                48 0190     .ASCII  /H/                  ; H - HIGH, P1 SPACE PREFIX
                                58 0191     .ASCII  /X/                  ; X REGISTER PREFIX
                          0000002A 0192  NPRIM=.-PRIMARY                  ; NUMBER OF PRIMARY COMMANDS
                                   0192
```

```
                               0192              .SBTTL  PRIMARY COMMAND SCANNER
                               0192
                               0192      ;
                               0192      ;       PRIMARY COMMAND SCANNER
                               0192      ;
                               0192
                               0192
      00 0D 0A 3F 48 45 0D 0A  0192 OUTER:   .ASCIZ  <LF><CR>/EH?/<LF><CR>
                               019A
                         0000  019A DCOM:    .WORD                               ; CALL ENTRY POINT
                               019C          .IF     DF,SW_PROCESS               ; FOR PROCESS VERSION ONLY
                               019C          MOVAB   W^DBGEXCEP,(FP)             ; SET CONDITION HANDLER ADDRESS
                               019C          .ENDC
                      13   11  019C          BRB     SCANP                       ; ENTER SCANP
                54    F1 AF 9E 019E ERROR:   MOVAB   OUTER,R4                    ; SET ADDR OF CONTROL STRING
                      01D9 30  01A2          BSBW    OUTZSTRING                  ; OUTPUT ASCIZ STRING
                  5E  5D   D0  01A5 SUPERST:MOVL     FP,SP                       ; RESET STACK
              59   AC AB   9E  01A8          MOVAB   INBUF-B(R11),R9             ; RESET STRING ADDRESS
                      69   94  01AC          CLRB    (R9)                        ; AND FORCE READ
                    02C6   30  01AE          BSBW    RESET                       ; RESET SCANNER
                      02   10  01B1 SCANP:   BSBB    NEXTP                       ; SCAN INPUT
                      FC   11  01B3          BRB     SCANP                       ; SCAN IT ALL
                               01B5 NEXTP:                                       ; PROCESS NEXT PRIMARY CHAR
                    0206   30  01B5          BSBW    GETCHAR                     ; GET CHARACTER
          AB AF   2A  58   3A  01B8          LOCC    R8,#NPRIM,PRIMARY           ; CHECK IT
                      DF   13  01BD          BEQL    ERROR                       ; NOT FOUND, ERROR
              50   2A  50  C3  01BF          SUBL3   R0,#NPRIM,R0                ; RATIONALIZE INDEX
                               01C3          CASE    R0,LIMIT=#16,<-
                               01C3                  DOT,-                       ; DOT - CURRENT LOCATION
                               01C3                  COMMA,-                     ; COMMA - FIELD SEPARATOR
                               01C3                  OPERATOR,-                  ; PLUS - ADD OPERATOR
                               01C3                  OPERATOR,-                  ; BLANK - ADD OPERATOR
                               01C3                  OPERATOR,-                  ; @ - SHIFT OPERATOR
                               01C3                  OPERATOR,-                  ; * - MULTIPLY OPERATOR
                               01C3                  OPERATOR,-                  ; % - DIVIDE OPERATOR
                               01C3                  NEGATE,-                    ; MINUS - SUBTRACT/NEGATE
                               01C3                  LBRACKET,-                  ; LEFT BRACKET - MODE SELECT
                               01C3                  TAB,-                       ; TAB - INDIRECT
                               01C3                  LINEFEED,-                  ; LINE FEED - NEXT LOCATION
                               01C3                  RETURN,-                    ; RETURN - CLOSE OPEN CELL
                               01C3                  SLASH,-                     ; SLASH - OPEN FOR DISPLAY
                               01C3                  DQUOTE,-                    ; DOUBLE QUOTE - OPEN FOR ASCII DISPLAY
                               01C3                  EQUALS,-                    ; EQUALS - DISPLAY VALUE
                               01C3                  ESCAP,-                     ; ESCAPE - PREVIOUS LOCATION
                               01C3                  STEP,-                      ; 'S' - SINGLE STEP
                               01C3                  SEMI,-                      ; SEMI COLON - SECONDARY COMMAND
                               01C3                  COLON,-                     ; COLON - SEPARATE PID FROM ADDRESS
                               01C3                  PREG,-                      ; 'P' - PROCESSOR REGISTER
                               01C3                  QUANT,-                     ; 'Q' - LAST QUANTITY
                               01C3                  QUOTE,-                     ; QUOTE - BEGIN ASCII STRING
                               01C3                  REGISTER,-                  ;
                               01C3                  GLOBL,-                     ; G - GLOBAL PREFIX
                               01C3                  HIGH,-                      ; H - P1 SPACE PREFIX
                               01C3                  XREG,-                      ; X REGISTER
                               01C3                  >
          19'  10  50      AF  01C3          CASEW   R0,#16,S^#<<30001$-30000$>/2>-1
                               01C7 30000$:
```

```
                  03C8' 01C7          .SIGNED_WORD    DOT-30000$
                  00D4' 01C9          .SIGNED_WORD    COMMA-30000$
                  026A' 01CB          .SIGNED_WORD    OPERATOR-30000$
                  026A' 01CD          .SIGNED_WORD    OPERATOR-30000$
                  026A' 01CF          .SIGNED_WORD    OPERATOR-30000$
                  026A' 01D1          .SIGNED_WORD    OPERATOR-30000$
                  026A' 01D3          .SIGNED_WORD    OPERATOR-30000$
                  0273' 01D5          .SIGNED_WORD    NEGATE-30000$
                  02EB' 01D7          .SIGNED_WORD    LBRACKET-30000$
                  027F' 01D9          .SIGNED_WORD    TAB-30000$
                  012B' 01DB          .SIGNED_WORD    LINEFEED-30000$
                  00BD' 01DD          .SIGNED_WORD    RETURN-30000$
                  008C' 01DF          .SIGNED_WORD    SLASH-30000$
                  0087' 01E1          .SIGNED_WORD    DQUOTE-30000$
                  02A1' 01E3          .SIGNED_WORD    EQUALS-30000$
                  028F' 01E5          .SIGNED_WORD    ESCAP-30000$
                  0306' 01E7          .SIGNED_WORD    STEP-30000$
                  02C4' 01E9          .SIGNED_WORD    SEMI-30000$
                  03B8' 01EB          .SIGNED_WORD    COLON-30000$
                  065E' 01ED          .SIGNED_WORD    PREG-30000$
                  03D5' 01EF          .SIGNED_WORD    QUANT-30000$
                  0616' 01F1          .SIGNED_WORD    QUOTE-30000$
                  03ED' 01F3          .SIGNED_WORD    REGISTER-30000$
                  0043' 01F5          .SIGNED_WORD    GLOBL-30000$
                  0049' 01F7          .SIGNED_WORD    HIGH-30000$
                  041B' 01F9          .SIGNED_WORD    XREG-30000$
                        01FB  30001$:
        10   50   B1    01FB          CMPW    R0,#16                  ; IS NUMBER > RADIX
             9E   18    01FE          BGEQ    ERROR                   ; YES
        56   10   C4    0200          MULL    #16,R6                  ; SCALE BY RADIX
        56   50   C0    0203          ADDL    R0,R6                   ; AND ADD NEW DIGIT
        6A   04   C8    0206  INFLD:  BISL    #<1@V_INFIELD>,(R10)    ; NOTE FIELD INPUT
                  05    0209          RSB                             ; NEXT PRIMARY CHARACTER
                        020A
                        020A
   54   01   1F   9C    020A  GLOBL:  ROTL    #31,#1,R4               ; GENERATE SYSTEM SPACE PREFIX
             07   11    020E          BRB     PRE1                    ; MERGE WITH COMMON
54   7FFE0000 8F  D0    0210  HIGH:   MOVL    #^X7FFE0000,R4          ; P1 SPACE BASE ADDRESS
             06   10    0217  PRE1:   BSBB    ENDEXPR                 ; END EXPRESSION
        56   54   D0    0219          MOVL    R4,R6                   ; SET INTO ACCUM
        E7   AF   9F    021C          PUSHAB  INFLD                   ; RETURN THROUGH INFLD
                        021F  ;       BRB     ENDEXPR
                        021F
```

```
                                021F         .SBTTL  ENDEXPR - END EXPRESSION
                                021F
                                021F     ;
                                021F     ;
                                021F     ;
                                021F     ENDEXPR:
          03 6A    07     E5    021F         BBCC    #V_NEGATE,(R10),5$       ; SKIP IF NOT NEGATE
             56    56     CE    0223         MNEGL   R6,R6                    ; NEGATE ACCUMULATOR
                   06     10    0226     5$: BSBB    10$                      ; PERFORM OPERATION
                   56     D4    0228         CLRL    R6                       ; CLEAR ACCUMULATOR
                FF AB     94    022A         CLRB    OPER-B(R11)              ; INIT OPERATOR
                          05    022D         RSB                              ; AND RETURN
                                022E     10$: CASE   OPER-B(R11),TYPE=B,<-    ; DO OPERATION
                                022E                 ADD,-                    ; ADD, PLUS
                                022E                 ADD,-                    ; BLANK, PLUS
                                022E                 SHFT,-                   ; SHIFT, @
                                022E                 MUL,-                    ; MULTIPLY, *
                                022E                 DIV,-                    ; DIVIDE, %
                                022E                 >                        ;
   04'    00    FF AB     8F    022E         CASEB   OPER-B(R11),#0,S^#<<30003$-30002$>/2>-1
                                0233     30002$:
                       0017'    0233         .SIGNED_WORD    ADD-30002$
                       0017'    0235         .SIGNED_WORD    ADD-30002$
                       000A'    0237         .SIGNED_WORD    SHFT-30002$
                       000F'    0239         .SIGNED_WORD    MUL-30002$
                       0013'    023B         .SIGNED_WORD    DIV-30002$
                                023D     30003$:
      57     57    56     78    023D     SHFT:   ASHL    R6,R7,R7                 ; SHIFT
                          05    0241         RSB                              ; AND EXIT
             57    56     C4    0242     MUL:    MULL    R6,R7   ; MULTIPLY
                          05    0245         RSB                              ; AND EXIT
             57    56     C6    0246     DIV:    DIVL    R6,R7   ; DIVIDE
                          05    0249         RSB                              ; AND EXIT
             57    56     C0    024A     ADD:    ADDL    R6,R7   ; ADD
                          05    024D         RSB                              ; AND EXIT
                                024E
                          024E          .SBTTL  SLASH - OPEN CELL
                          024E
                          024E  ;
                          024E  ;       OPEN SPECIFIED CELL
                          024E  ;
                          024E  DQUOTE:                                       ; DISPLAY IN ASCII
           6A    02   88  024E          BISB    #<1@V_ASCII>,(R10)            ; SET ASCII FLAG
                 03   11  0251          BRB     OPEN                          ;
                          0253                                                ;
                          0253  SLASH:                                        ;
           6A    02   8A  0253          BICB    #<1@V_ASCII>,(R10)            ; CLEAR ASCII DISPLAY MODE
                 46   10  0256  OPEN:   BSBB    ENDFIELD                      ; TERMINATE FIELD
        06 6A    08   E0  0258          BBS     #V_F1,(R10),5$                ; ADDR SPECIFIED?
        6B    04 AB   D0  025C          MOVL    QUAN-B(R11),CURDOT-B(R11)     ;       ; NO, GO INDIRECT
                 04   11  0260          BRB     10$                           ; AND DISPLAY CONTENT
        6B    E0 AB   D0  0262  5$:     MOVL    F1-B(R11),CURDOT-B(R11)       ; SET NEW DOT
  50    6A  01   0F   EF  0266  10$:    EXTZV   #V_PRMODE,#1,(R10),R0         ; GET PROCESSOR REGISTER MODE FLAG
  6A    01  1F   50   F0  026B          INSV    R0,#V_PREG,#1,(R10)           ; AND MOVE TO SEMI-PERMANENT COPY
               0086   30  0270          BSBW    LOCOUT                        ; OUTPUT AND OPEN
        1A 6A    09   E1  0273          BBC     #V_F2,(R10),RSET              ; RANGE SPECIFIED?
        6B    E4 AB   D1  0277  15$:    CMPL    F2-B(R11),CURDOT-B(R11)       ; CHECK FOR END
                 14   15  027B          BLEQ    RSET                          ; YES
                          027D          .IF     NDF,SW_PROCESS
                 76   10  027D          BSBB    NEXTLOC                       ; INCREMENT TO NEXT DOT
                          027F          .IFF
                          027F          BSBW    NEXTLOC                       ; INCREMENT TO NEXT DOT
                          027F          .ENDC
                 F6   11  027F          BRB     15$                           ; AND CONTINUE
               FF1A   31  0281  ERR4:   BRW     ERROR                         ; DECLARE ERROR
                          0284
```

```
                0284  460          .SBTTL  RETURN - CLOSE CURRENT OPEN CELL
                0284  461
                0284  462  ;
                0284  463  ;       RETURN - CLOSE CURRENT OPEN CELL
                0284  464  ;
                0284  465
                0284  466  RETURN:
       1B   10  0284  467          BSBB    ENDFIELD                ; TERMINATE CURRENT FIELD
                0286  468          .ENABL  LSB
 0A 6A  00  E5  0286  469          BBCC    #V_OPEN,(R10),10$       ; SKIP IF NONE OPEN
 03 6A  08  E1  028A  470          BBC     #V_F1,(R10),RSET        ; SKIP IF NOTHING TO STORE
     0560   30  028E  471          BSBW    DEPOSIT                 ; DEPOSIT
     01E3   31  0291  472  RSET:   BRW     RESET                   ; RESET SCANNER
 F9 6A  08  E1  0294  473  10$:    BBC     #V_F1,(R10),RSET        ; DONE IF NO INPUT
     01D4   31  0298  474          BRW     EQ[1                    ; OTHERWISE OUTPUT
                029B  475          .DSABL  LSB                     ;
```

```
                        029B            .SBTTL  ENDFIELD - TERMINATE CURRENT FIELD
                        029B
                        029B   ;
                        029B   ;        COMMA TERMINATE CURRENT FIELD
                        029B   ;
             FF68  30   029B   COMMA:   BSBW    INFLD                   ; ZERO IF NULL FIELD
                        029E
                        029E   ;
                        029E   ;        TERMINATE CURRENT FIELD
                        029E   ;
                        029E   ENDFIELD:                                ;
    16 6A     02   E5   029E            BBCC    #V_INFIELD,(R10),10$    ; CLEAR PENDING FIELD
             FF7A  30   02A2            BSBW    ENDEXPR                 ; END EXPRESSION
      50    FC AB  9A   02A5            MOVZBL  FCTR-B(R11),R0          ; GET FIELD POINTER
  D3 01 AA    50   E2   02A9            BBSS    R0,1(R10),ERR4          ; ERROR IF TOO MANY FIELDS
   E0 AB40    57   D0   02AE            MOVL    R7,F1-B(R11)[R0]        ; STORE FIELD VALUE
            FC AB  96   02B3            INCB    FCTR-B(R11)             ; INCREMENT FIELD COUNTER
              56   7C   02B6            CLRQ    R6                      ; CLEAR ACCUMULATORS
                   05   02B8   10$:     RSB                             ; RETURN
                        02B9
```

```
                          02B9          .SBTTL  FETCH - OBTAIN DATA SPECIFIED
                          02B9
                          02B9     ;
                          02B9     ;            FETCH SPECIFIED DATA
                          02B9     ;
                          02B9     FETCH:                                          ;
           1D 6A    1F  E0 02B9          BBS     #V_PREG,(R10),40$               ; BR IF PROCESSOR REGISTER
                          02BD          .IF     DF,SW_PROCESS                   ;
                          02BD          TSTL    PID-B(R11)                      ; CHECK FOR PROCESS GET
                          02BD          BNEQ    50$                             ; BR IF YES
                          02BD          .ENDC                                   ;
                          02BD          CASE    CURTYPE-B(R11),TYPE=B,<-              ; OPERATE ON TYPE
                          02BD                  10$,-                           ; BYTE
                          02BD                  20$,-                           ; WORD
                          02BD                  30$,-                           ; LONG
                          02BD                  >
    02'  00    FE AB    8F 02BD          CASEB   CURTYPE-B(R11),#0,S^#<<30005$-30004$>/2>-1
                          02C2     30004$:
                    0006' 02C2          .SIGNED_WORD    10$-30004$
                    000C' 02C4          .SIGNED_WORD    20$-30004$
                    0012' 02C6          .SIGNED_WORD    30$-30004$
                          02C8     30005$:
      04 AB    00 BB    9A 02C8     10$:   MOVZBL  @CURDOT-B(R11),QUAN-B(R11)            ; GET BYTE
                       05 02CD          RSB                                     ; RETURN
      04 AB    00 BB    3C 02CE     20$:   MOVZWL  @CURDOT-B(R11),QUAN-B(R11)            ; GET WORD
                       05 02D3          RSB                                     ; RETURN
      04 AB    00 BB    D0 02D4     30$:   MOVL    @CURDOT-B(R11),QUAN-B(R11)            ; GET LONGWORD
                       05 02D9          RSB                                     ; RETURN
                          02DA          .IF     NDF,SW_PROCESS                  ;
                          02DA     40$:   MFPR    CURDOT-B(R11),QUAN-B(R11)             ; GET PROCESSOR REGISTER
           04 AB    6B  DB 02DA                  MFPR    CURDOT-B(R11),QUAN-B(R11)
                       05 02DE          RSB                                     ;
                          02DF          .IFF                                    ; FALSE IF PROCESS VERSION
                          02DF     40$:
                          02DF          $CMKRNL_S       B^FTCHPREG,(AP) ; CALL IN KERNEL MODE TO FETCH
                          02DF          RSB                                     ;
                          02DF     50$:   BRW     FETCHP                          ; FETCH FROM FOREIGN PROCESS
                          02DF          .ENDC
                          02DF
                          02DF          .IF     DF,SW_PROCESS                   ;
                          02DF     FTCHPREG:                                       ;
                          02DF          .WORD   0                               ; ENTRY MASK
                          02DF          MOVAB   W^PREXC,(FP)                    ; SET EXCEPTION HANDLER
                          02DF          MFPR    CURDOT-B(R11),QUAN-B(R11)             ; GET PROCESSOR REGISTER
                          02DF          MOVL    #1,R0                           ; RETURN SUCCESS
                          02DF          RET                                     ;
                          02DF
                          02DF          .ENDC                                   ;
                          02DF          .SBTTL  NEXTDOT - INCREMENT CURRENT LOCATION
                          02DF
                          02DF     ;
                          02DF     ;        INCREMENT TO NEXT LOCATION
                          02DF     ;
                          02DF     NEXTDOT:                                             ;
             51    01  D0 02DF              MOVL    #1,R1                               ; ASSUME UNIT INCREMENT
                   6A  D5 02E2              TSTL    (R10)                               ; CHECK FOR PREG
                   05  19 02E4              BLSS    10$                                 ; YES, USE UNIT INCREMENT
    51    51   FE  AB  9C 02E6              ROTL    CURTYPE-B(R11),R1,R1                ; FORM INCREMENT
             6B    51  C0 02EB     10$:     ADDL    R1,CURDOT-B(R11)                    ; AND ADD TO DOT
                       05 02EE              RSB                                         ; RETURN
                          02EF
```

```
                        02EF          .SBTTL  OUTPUT - DISPLAY CONTENT
                        02EF     ;
                        02EF     ;        OUTPUT CONTENT
                        02EF     ;
                        02EF     OUTBB:
             1C 0C 04   02EF          .BYTE   4,12,28                    ; STARTING DIGIT LIST
                        02F2          .SBTTL  LINE FEED - DISPLAY NEXT
                        02F2     ;
                        02F2     ;
                        02F2     ;
                        02F2     LINEFEED:                               ;
            FF8F   30   02F2          BSBW    RETURN                     ; CLOSE OPEN CELL
                        02F5     NEXTLOC:                                ; PROMPT WITH NEXT LOCATION
              E8   10   02F5          BSBB    NEXTDOT                    ; INCREMENT LOCATION
                        02F7     LOCPROMPT:                              ; DISPLAY ADDR/CONTENT
              2B   10   02F7          BSBB    OUTPUTA                    ; OUTPUT ADDRESS
              BE   10   02F9     LOCOUT: BSBB FETCH                      ; FETCH CONTENT
        6A    01   C8   02FB          BISL    #<1@V_OPEN>,(R10)          ; INDICATE OPEN CELL
                        02FE
                        02FE     OUTPUT:                                 ;
     51    FE AB   9A   02FE          MOVZBL  CURTYPE-B(R11),R1          ; GET TYPE
     52  E9 AF41   9A   0302          MOVZBL  OUTBB[R1],R2               ; INIT DIGIT SELECTOR
     53    04 AB   D0   0307          MOVL    QUAN-B(R11),R3             ; GET QUANTITY TO DISPLAY
     04 6A    01   E0   030B          BBS     #V_ASCII,(R10),10$         ; CHECK FOR ASCII OUT
              53   10   030F          BSBB    OUTCOM                     ; OUTPUT NUMBER IN HEX
              0E   11   0311          BRB     20$                        ; AND EXIT THROUGH OUTSPACE
     08 AB    53   D0   0313     10$:    MOVL R3,OUTBUF-B(R11)           ; PUT STRING IN BUFFER
     52  01   51   78   0317          ASHL    R1,#1,R2                   ; GET COUNT
        08 AB42    94   031B          CLRB    OUTBUF-B(R11)[R2]          ; MARK END OF STRING
              59   10   031F          BSBB    OUTZBUF                    ; OUTBUT ASCIIZ BUFFER
            008B   31   0321     20$:    BRW  OUTSPACE                   ; FOLLOW WITH SPACE
                        0324
```

```
                            0324          .SBTTL  OUTPUTA - OUTPUT ADDRESS
                            0324  ;
                            0324  ;       OUTPUT ADDRESS
                            0324  ;
                            0324  OUTPUTA:                                          ; OUTPUT ADDRESS
             008D      30   0324          BSBW    CRLF                              ; OUTPUT CR/LF
          53    18 AB  9E   0327          MOVAB   SAVREG-B(R11),R3                  ; BASE OF REGISTER AREA
                            032B          .IF     DF,SW_PROCESS                     ; ONLY FOR PROCESS VERSION
                            032B          TSTL    PID-B(R11)                        ; CHECK FOR OTHER PROCESS ADDRESS
                            032B          BNEQ    3$                                ; BR IF YES
                            032B          .ENDC                                     ;
     53    6B    53    C3   032B          SUBL3   R3,CURDOT-B(R11),R3               ; COMPUTE OFFSET INTO REGISTER AREA
                 12    19   032F          BLSS    5$                                ; NOT GENERAL REGISTER
           53    04    C6   0331          DIVL    #4,R3                             ; SCALE TO LONGWORD NUMBER
           0F    53    D1   0334          CMPL    R3,#15                            ; CHECK FOR MAX REG NUMBER
                 0A    14   0337          BGTR    5$                                ; GTR, NOT A REGISTER
        50    52 8F    9A   0339          MOVZBL  #^A'R',R0                         ; OUTPUT PREFIX
                 52    10   033D          BSBB    OUTCHAR                           ; OF 'R'
                 52    D4   033F          CLRL    R2                                ; AND SET FOR ONE DIGIT OF OUTPUT
                 13    11   0341          BRB     10$                               ;
                            0343          .IF     DF,SW_PROCESS                     ; FOR PROCESS VERSION ONLY
                            0343  3$:     TSTL    (R10)                             ; CHECK FOR PROCESSOR REGISTER
                            0343          BLSS    5$                                ; BR IF YES
                            0343          MOVL    #28,R2                            ; SET FOR LONGWORD OUTPUT
                            0343          MOVL    PID-B(R11),R3                     ; GET PID OF TARGET
                            0343          BSBB    OUTCOM                            ; OUTPUT PID AS LONGWORD
                            0343          MOVZBL  #^A':',R0                         ; SEPARATE WITH ':'
                            0343          BSBB    OUTCHAR                           ; OUTPUT COLON
                            0343          .ENDC                                     ;
           53    6B    D0   0343  5$:     MOVL    CURDOT-B(R11),R3                  ; GET ADDRESS
           52    1C    D0   0346          MOVL    #28,R2                            ; ASSUME LONGWORD OUTPUT
                 6A    D5   0349          TSTL    (R10)                             ; CHECK FOR PROCESSOR REGISTER
                 09    18   034B          BGEQ    10$                               ; NO, JUST A LONGWORD
        50    50 8F    9A   034D          MOVZBL  #^A'P',R0                         ; PRECEDE WITH A 'P'
                 3E    10   0351          BSBB    OUTCHAR                           ; OUTPUT P
           52    04    D0   0353          MOVL    #4,R2                             ; SET FIELD TO 2 DIGITS
                 0C    10   0356  10$:    BSBB    OUTCOM                            ; COMMON OUTPUT
           50    2F    9A   0358          MOVZBL  #SLSH,R0                          ; OUTPUT SLASH
                 34    11   035B          BRB     OUTCHAR                           ; RETURN THROUGH OUTCHAR
                            035D  OUTDIGIT:                                         ; OUTPUT ONE DIGIT
                 52    D4   035D          CLRL    R2                                ; ZAP DIGIT SELECTOR
                 03    11   035F          BRB     OUTCOM                            ; AND MERGE WITH COMMON
                            0361
                            0361  OUTLONG:                                          ; OUTPUT LONGWORD
           52    1C    D0   0361          MOVL    #28,R2                            ; SET DIGIT SELECTOR
                            0364  OUTCOM:                                           ; FORMAT IT
        54    08 AB    9E   0364          MOVAB   OUTBUF-B(R11),R4                  ; GET ADDRESS OF OUTPUT BUFFER
  51    53    04 52    EF   0368  10$:    EXTZV   R2,#4,R3,R1                       ; GET DIGIT
     84    FDF6 CF41   90   036D          MOVB    PRIMARY[R1],(R4)+                 ; BUFFER IT
           52    04    C2   0373          SUBL    #4,R2                             ; NEXT DIGIT
                 F0    18   0376          BGEQ    10$                               ; DO ALL REQUESTED
                 64    94   0378          CLRB    (R4)                              ; MARK END OF BUFFER
        54    08 AB    9E   037A  OUTZBUF:MOVAB   OUTBUF-B(R11),R4                  ; GET START OF BUFFER
                            037E
                            037E  OUTZSTRING:                                       ; OUTPUT ASCIZ STRING
           50    84    9A   037E          MOVZBL  (R4)+,R0                          ; GET A CHAR
                 04    13   0381          BEQL    10$                               ; BR IF DONE
          0C   10  0383          BSBB    OUTCHAR              ; OUTPUT CHAR
          F7   11  0385          BRB     OUTZSTRING           ; CONTINUE
               05  0387  10$:    RSB                          ; RETURN IF DONE
                   0388
                   0388
                   0388  OUTBSLSH:                            ; OUTPUT BACK SLASH
  50    5C 8F  9A  0388          MOVZBL  #BSLSH,R0            ; SET CHARACTER CODE
          03   11  038C          BRB     OUTCHAR              ; AND OUTPUT IT
     50   58   9A  038E  OUTR8:  MOVZBL  R8,R0                ; GET CHAR TO OUTPUT
                   0391  OUTCHAR:                             ; OUTPUT CHAR IN R0
                   0391          .IF     NDF,SW_PROCESS
          5C   D5  0391          TSTL    AP                   ; CHECK FOR CONSOLE
          05   12  0393          BNEQ    10$                  ; NO, USE DEVICE DIRECTLY
                   0395          MFPR    #PR$_TXCS,R1         ; GET CONSOLE TRANSMIT STATUS
     51   22   DB  0395          MFPR       #PR$_TXCS,R1
          04   11  0398          BRB     20$                  ; MERGE WITH COMMON CODE
  51    04 AC  B0  039A  10$:    MOVW    OUTCR(AP),R1         ; GET STATUS
  EF 51   07   E1  039E  20$:    BBC     #7,R1,OUTCHAR        ; WAIT FOR READY
          5C   D5  03A2          TSTL    AP                   ; CHECK FOR CONSOLE
          04   12  03A4          BNEQ    30$                  ; YES
     23   50   DA  03A6          MTPR    R0,#PR$_TXDB         ; SEND CHARACTER TO CONSOLE
               05  03A9          RSB                          ; RETURN
  06 AC   50   90  03AA  30$:    MOVB    R0,OUTB(AP)          ; OUTPUT CHAR
                   03AE          .IFF                         ; FALSE FOR PROCESS VERSION
                   03AE          PUSHL   R0                   ; BUFFER CHARACTER ON STACK
                   03AE          MOVL    SP,R0                ; SAVE POINTER TO IT
                   03AE          $QIO_S  EFN=#30,-            ;
                   03AE                  CHAN=TTCHAN,-        ;
                   03AE                  FUNC=#IO$_WRITEVBLK,-  ;
                   03AE                  P1=(R0),-            ; BUFFER ADDRESS
                   03AE                  P2=#1                ; ONE CHARACTER
                   03AE          POPR    #^M<R0>              ; RESTORE CHARACTER
                   03AE          .ENDC                        ;
               05  03AE          RSB                          ; AND RETURN
                   03AF  OUTSPACE:                            ;
     50   20   9A  03AF          MOVZBL  #32,R0               ; SET CODE FOR SPACE
          DD   11  03B2          BRB     OUTCHAR              ; AND SEND IT
     50   0D   9A  03B4  CRLF:   MOVZBL  #CR,R0               ; RETURN
          D8   10  03B7          BSBB    OUTCHAR              ; SEND IT
     50   0A   9A  03B9          MOVZBL  #LF,R0               ; LINE FEED
          D3   11  03BC          BRB     OUTCHAR              ; SEND IT
                   03BE
                   03BE
```

```
                      03BE           .SBTTL  GETCHAR - GET INPUT CHARACTER ROUTINE
                      03BE
                      03BE      ;
                      03BE      ;       GETCHAR - GET INPUT CHARACTER
                      03BE      ;
                      03BE      ; OUTPUT:
                      03BE      ;       R8 - INPUT CHARACTER
                      03BE      ;       R9 - BUFFER POINTER UPDATED (BUFFER IN ASCIZ FORMAT)
                      03BE      ;
                      03BE
                      03BE      GETCHAR:
   58   89   9A       03BE              MOVZBL  (R9)+,R8                ; GET NEXT CHARACTER
        01   13       03C1              BEQL    10$                     ; READ IF NONE AVAIL
             05       03C3              RSB
  59  AC AB  9E       03C4      10$:    MOVAB   INBUF-B(R11),R9         ; SET ADDRESS OF INPUT BUFFER
                      03C8              .IF     NDF,SW_PROCESS          ;
        5C   D5       03C8      20$:    TSTL    AP                      ; CHECK FOR CONSOLE
        05   13       03CA              BEQL    30$                     ; YES
   50   6C   B0       03CC              MOVW    RDCR(AP),R0             ; GET STATUS
        03   11       03CF              BRB     40$                     ; CHECK STATUS
                      03D1      30$:    MFPR    #PR$_RXCS,R0            ; GET CONSOLE STATUS
   50   20   DB       03D1                      MFPR    #PR$_RXCS,R0
F0 50   07   E1       03D4      40$:    BBC     #7,R0,20$               ; WAIT FOR READY
        5C   D5       03D8              TSTL    AP                      ; CHECK FOR CONSOLE
        06   13       03DA              BEQL    50$                     ; YES
  58  02 AC  90       03DC              MOVB    RDBUF(AP),R8            ; GET CHARACTER
        03   11       03E0              BRB     60$                     ; MERGE WITH COMMON
                      03E2      50$:    MFPR    #PR$_RXDB,R8            ; GET CONSOLE CHARACTER
   58   21   DB       03E2                      MFPR    #PR$_RXDB,R8
                      03E5              .IFF                            ; FALSE IF PROCESS VERSION
                      03E5      15$:    MOVAB   TERMASKD,R1             ; ADDRESS OF TERMINATOR MASK DESCR
                      03E5              $QIOW_S EFN=#31,-               ;
                      03E5                      CHAN=TTCHAN,-           ; INPUT DEVICE CHANNEL
                      03E5                      IOSB=TTIOSB,-           ; IO STATUS BLOCK
                      03E5                      FUNC=#<IO$_READVBLK>,-  ;
                      03E5                      P1=(R9),-               ; BUFFER ADDRESS
                      03E5                      P2=#80,-                ; READ SIZE
                      03E5              P4=R1                           ;
                      03E5              MOVZWL  TTIOSB+2,R0             ; GET SIZE READ
                      03E5              MOVB    TTIOSB+4,(R0)+[R9]      ; BUFFER TERMINATOR
                      03E5              CLRB    (R9)[R0]                ; MARK END OF BUFFER
                      03E5              MOVL    R9,R2                   ; POINT TO START OF STRING
                      03E5      20$:    MOVZBL  (R2)+,R8                ; GET A CHARACTER
                      03E5              BEQL    15$                     ; EMPTY, READ SOME MORE
                      03E5              .ENDC
58  80 8F    8A       03E5      60$:    BICB    #^X80,R8                ; STRIP PARITY
7F 8F   58   91       03E9              CMPB    R8,#RUBOUT              ; CHECK FOR RUBOUT
        15   12       03ED              BNEQ    90$                     ; NO
03 6A   06   E2       03EF              BBSS    #V_RUB,(R10),70$        ; SET START OF RUBOUT SEQUENCE
      FF92   30       03F3              BSBW    OUTBSLSH                ; OUTPUT BACK SLASH
   58   79   9A       03F6      70$:    MOVZBL  -(R9),R8                ; GET RUBBED OUT CHAR
        04   12       03F9              BNEQ    80$                     ; SKIP INC
        59   D6       03FB              INCL    R9                      ; POINT AT START OF BUFFER
        C9   11       03FD              BRB     20$                     ; AND GET ANOTHER
      FF8C   30       03FF      80$:    BSBW    OUTR8                   ; OUTPUT RUBBED OUT CHAR
        C4   11       0402              BRB     20$                     ; AND GET ANOTHER
03 6A   06   E5       0404      90$:    BBCC    #V_RUB,(R10),100$       ; TERMINATE RUBOUT SEQUENCE
```

```
              FF7D   30  0408          BSBW    OUTBSLSH                ; OUTPUT BACK SLASH
        03 58    06   E1  040B  100$:   BBC     #6,R8,110$              ; BR IF NOT ALPHA
           58    20   8A  040F          BICB    #32,R8                  ; SET TO UPPER CASE
                          0412  110$:                                   ;
                          0412          .IF     NDF,SW_PROCESS          ;
              FF79   30  0412          BSBW    OUTR8                   ; ECHO CHARACTER
                          0415          .ENDC
           89    58   90  0415          MOVB    R8,(R9)+                ; BUFFER NEW CHAR
  FD63 CF  08    58   3A  0418          LOCC    R8,#NTERM,TERM          ; CHECK FOR TERMINATOR
                 A8   13  041E          BEQL    20$                     ; NOT A TERMINATOR
           58    0D   91  0420          CMPB    #CR,R8                  ; IS CHAR = RETURN
                 03   12  0423          BNEQ    120$                    ; NO,
               FF8C   30  0425          BSBW    CRLF                    ; YES, SEND CR/LF
                 69   94  0428  120$:   CLRB    (R9)                    ; MARK END OF BUFFER
        59    AC AB   9E  042A          MOVAB   INBUF-B(R11),R9         ; RESTORE BUFFER BASE
               FF8D   31  042E          BRW     GETCHAR                 ; AND TRY AGAIN
```

```
                              0431      .SBTTL  PLUS/MINUS OPERATORS
                              0431 ;
                              0431 ;        PLUS/MINUS OPERATORS
                              0431 ;
                              0431 BLANK:                                   ; SAME AS PLUS
                              0431 OPERATOR:
                  FDEB   30   0431      BSBW    ENDEXPR                      ; END EXPR
       FF AB   50   12   83   0434      SUBB3   #OPERBAS,R0,OPER-B(R11)      ; SET OPERATOR
                        05    0439      RSB                                  ; RETURN
                              043A ;
                              043A ;        MONADIC MINUS - NEGATE
                              043A ;
                    56   D5   043A NEGATE: TSTL R6                           ; TEST ACCUMULATOR
                    03   13   043C      BEQL    5$                           ; EMPTY
                  FDDE   30   043E      BSBW    ENDEXPR                      ; OTHERWISE PERFORM OPERATION
            6A    80 8F  8C   0441 5$:  XORB    #<1@V_NEGATE>,(R10)          ; TOGGLE NEGATE FLAG
                        05    0445 10$: RSB                                  ; AND RETURN
                              0446
                              0446
                              0446       .SBTTL  TAB - INDIRECT DISPLAY
                              0446   ;
                              0446   ;       TAB
                              0446   ;
             6B    04 AB  D0  0446   TAB:    MOVL    QUAN-B(R11),CURDOT-B(R11)       ; GO INDIRECT
    50    6A    01    0F  EF  044A           EXTZV   #V_PRMODE,#1,(R10),R0   ; GET PROCESSOR REGISTER MODE
    6A    01    1F    50  F0  044F           INSV    R0,#V_PREG,#1,(R10)     ; AND COPY TO SEMI-PERMANENT FLAG
                      0F  11  0454           BRB     LOCP                    ; AND DISPLAY IT
                              0456
                              0456   ;
                              0456   ;       ESCAPE - DISPLAY PREVIOUS LOCATION
                              0456   ;
                              0456
                              0456   ESCAP:                                  ;
                51    01  D0  0456           MOVL    #1,R1                   ; ASSUME UNIT INCREMENT
                      6A  D5  0459           TSTL    (R10)                   ; CHECK FOR PROCESSOR REGISTER
                      05  19  045B           BLSS    10$                     ; YES, USE UNIT INCREMENT
       51    51    FE AB  9C  045D           ROTL    CURTYPE-B(R11),R1,R1    ; FORM INCREMENT
                6B    51  C2  0462   10$:    SUBL    R1,CURDOT-B(R11)        ; AND SUBTRACT FROM DOT
                    FE8F  31  0465   LOCP:   BRW     LOCPROMPT               ; PROMPT WITH CONTENT
```

```
                            0468       .SBTTL  EQUALS - DISPLAY VALUE
                            0468  ;
                            0468  ;        EQUALS - VALUE DISPLAY
                            0468  ;
                            0468  EQUALS:                                 ;
                            0468       .ENABL  LSB                        ;
                   FE33  30 0468       BSBW    ENDFIELD                   ; TERMINATE FIELD
             05 6A   08  E1 046B       BBC     #V_F1,(R10),10$            ; IGNORE IF FIELD BLANK
      04 AB    E0 AB     D0 046F  EQL1: MOVL   F1=B(R11),QUAN-B(R11)      ; SET QUANTITY
                   FE87  30 0474  10$:  BSBW   OUTPUT                     ; OUTPUT IT
                            0477  ;     BRB    RESET                      ; RESET SCANNER
                            0477       .DSABL  LSB                        ;
                            0477
                            0477  ;
                            0477  ;        RESET
                            0477  ;
                            0477
  6A     00FFFF80 8F     CA 0477  RESET: BICL  #^X0FFFF80,(R10)                 ; CLEAR FIELD AND NEGATE FLAGS
                  FC AB  94 047E       CLRB    FCTR-B(R11)                ; CLEAR FIELD COUNTER
                     56  7C 0481       CLRQ    R6                         ; RESET ACCUMULATORS
                         05 0483       RSB                                ; RETURN
```

```
                           0484          .SBTTL  SEMI - SECONDARY COMMAND SET
                           0484      ;
                           0484      ;       SEMI
                           0484      ;
                           0484
                           0484      SECOND:                                         ;
                       58  0484              .ASCII  /X/                             ; X REGISTER SET/DISPLAY
                       50  0485              .ASCII  /P/                             ; P - PROCEED
                       4D  0486              .ASCII  /M/                             ; M - SET MODIFY FLAG
                       49  0487              .ASCII  /I/                             ; I - PROGRAM COUNTER
                       47  0488              .ASCII  /G/                             ; G - GO, START
                       45  0489              .ASCII  /E/                             ; E - EXECUTE STRING
                       42  048A              .ASCII  /B/                             ; B - SET/CLR BREAKPOINT
                 00000007  048B      NSEC=.-SECOND                                   ; NUMBER OF SECONDARY COMMANDS
                           048B
                           048B      SEMI:                                           ;
            6A   01    8A  048B              BICB    #<1@V_OPEN>,(R10)               ; CLEAR OPEN FLAG
                FE0D   30  048E              BSBW    ENDFIELD                        ; TERMINATE FIELD
                FF2A   30  0491              BSBW    GETCHAR                         ; GET SECONDARY COMMAND CHAR
   EB AF   07    58    3A  0494              LOCC    R8,#NSEC,SECOND                 ; LOCATE SECONDARY COMMAND
                           0499      10$:    CASE    R0,LIMIT=#1,<-                  ; SWITCH ON TYPE
                           0499                      BRKPOINT,-                      ; SET BREAKPOINT
                           0499                      EXECUTE,-                       ; EXECUTE STRING
                           0499                      GO,-                            ; SEMI-G, GO
                           0499                      PROGCTR,-                       ; SEMI-I, INSTRUCTION CONTER
                           0499                      MFYFLGS,-                       ; SEMI-M, MODIFY FLAG
                           0499                      PROCED,-                        ; SEMI-P, PROCEED
                           0499                      XSET,-                          ; SET XREGISTER
                           0499                      >                               ;
      06'  01    50    AF  0499              CASEW   R0,#1,S^#<<30011$-30010$>/2>-1
                           049D      30010$:
                    003A'  049D              .SIGNED_WORD    BRKPOINT-30010$
                    037A'  049F              .SIGNED_WORD    EXECUTE-30010$
                    00D8'  04A1              .SIGNED_WORD    GO-30010$
                    0105'  04A3              .SIGNED_WORD    PROGCTR-30010$
                    00EC'  04A5              .SIGNED_WORD    MFYFLGS-30010$
                    00E1'  04A7              .SIGNED_WORD    PROCED-30010$
                    0133'  04A9              .SIGNED_WORD    XSET-30010$
                           04AB      30011$:
                FCF0   31  04AB      ERR2:   BRW     ERROR                           ; ERROR
```

```
                              04AE        .SBTTL  LEFT BRACKET - MODE SELECTION
                              04AE  ;
                              04AE  ;
                              04AE  ;     LEFT BRACKET
                              04AE  ;
                              04AE  MODES:                                           ; MODE CHARACTER LIST
                          43  04AE        .ASCII  /C/                                ; CHARACTER
                          4C  04AF        .ASCII  /L/                                ; LONG, HEX
                          57  04B0        .ASCII  /W/                                ; WORD, HEX
                          42  04B1        .ASCII  /B/                                ; BYTE, HEX
                    00000004  04B2  NMODES=.-MODES                                   ; NUMBER OF MODE CHARACTERS
                              04B2
                              04B2
                              04B2  LBRACKET:                                        ; MODE SELECTION
                    FF09  30  04B2        BSBW    GETCHAR                            ; GET MODE CHAR
         F4 AF   04   58  3A  04B5        LOCC    R8,#NMODES,MODES                   ; CONVERT TO INDEX
                      EF  13  04BA        BEQL    ERR2                               ; NOT FOUND, ERROR
              09 50   02  E0  04BC        BBS     #2,R0,10$                          ; CHECK FOR 'C'
         FE AB   50   01  83  04C0        SUBB3   #1,R0,CURTYPE-B(R11)               ; SET MODE
                 6A   02  8A  04C5        BICB    #<1@V_ASCII>,(R10)                 ; CLEAR CHAR MODE
                          05  04C8        RSB                                        ; RETURN
                 6A   02  88  04C9  10$:  BISB    #<1@V_ASCII>,(R10)                 ; SET CHARACTER MODE
                          05  04CC        RSB
```

```
                              04CD   877          .SBTTL  SINGLE STEP
                              04CD   878 ;
                              04CD   879 ;        STEP
                              04CD   880 ;
   6A   02   03   01   F0     04CD   881 STEP:    INSV    #1,#V_TBIT,#2,(R10)        ; CLR V_ATBRK, SET V_TBIT
           00 6A  0F   E5     04D2   882          BBCC    #V_PRMODE,(R10),20$        ; CLEAR PROCESSOR REGISTER DISPLAY MODE
                       04     04D6   883 20$:     RET                                ; AND RETURN
```

```
                          04D7          .SBTTL  BRKPOINT - SET/CLEAR BREAKPOINTS
                          04D7      ;
                          04D7      ;       BRKPOINT
                          04D7      ;
                          04D7  BRKPOINT:                                       ;
          58 6A   08  E1  04D7          BBC     #V_F1,(R10),SHOBRK              ; DISPLAY BREAKPOINTS
          12 6A   09  E0  04DB          BBS     #V_F2,(R10),20$                 ; YES, IT WAS SPECIFIED
             52   01  D0  04DF          MOVL    #1,R2                           ; INIT INDEX
          FBD1 CF42   D5  04E2  10$:    TSTL    BRKADR[R2]                      ; FIND FREE SLOT
                  13  13  04E7          BEQL    30$                             ; YES, GOT ONE
     FFF3 52   01  08  F1  04E9          ACBL    #NBRK,#1,R2,10$                 ; CHECK THEM ALL
                  BA  11  04EF          BRB     ERR2                            ; ERROR
          52   E4 AB  D0  04F1  20$:    MOVL    F2-B(R11),R2                    ; GET BRKPOINT NUMBER
                  EB  13  04F5          BEQL    10$                             ; NULL FIELD, SCAN FOR SLOT
             52   08  D1  04F7          CMPL    #NBRK,R2                        ; CHECK FOR LEGAL
                  AF  19  04FA          BLSS    ERR2                            ; OUT OF RANGE
          FBDF CF42   D4  04FC  30$:    CLRL    BRKDSP[R2]                      ; CLEAR DISPLAY
          FBFA CF42   D4  0501          CLRL    BRKCOM[R2]                      ; CLEAR COMMAND ADDRESS
          50   E0 AB  D0  0506          MOVL    F1-B(R11),R0                    ; GET BREAKPOINT ADDRESS
                  03  13  050A          BEQL    35$                             ; ALLOW CLEAR OF BREAKPOINT
                          050C          .IF     DF,SW_PROCESS
                          050C          PUSHR   #^M<R0,R1,R2,R3,R4,R5,R6>               ; SAVE REGISTERS FOR PROTECTION CHAN
                          050C          MOVL    R0,R5                           ; SET START ADDRESS
                          050C          MOVL    R0,R6                           ; AND END ADDRESS
                          050C          BSBW    SETWRT                          ; SET PAGE WRITABLE
                          050C          MOVL    (SP),R0                         ; RESTORE BPT ADDRESS
                          050C          .ENDC
          60   60     90  050C          MOVB    (R0),(R0)                       ; TEST WRITABILITY OF ADDRESS
                          050F          .IF     DF,SW_PROCESS
                          050F          BSBW    REPROT                          ; RESTORE PROTECTION
                          050F          POPR    #^M<R0,R1,R2,R3,R4,R5,R6>       ;       ; AND REGISTERS
                          050F          .ENDC
          0C 6A   0A  E1  050F  35$:    BBC     #V_F3,(R10),40$                 ; DISPLAY SPECIFIED?
  FBC6 CF42   E8 AB  D0  0513          MOVL    F3-B(R11),BRKDSP[R2]            ; SET DISPLAY START
                  03  13  051A          BEQL    40$                             ; SKIP TEST IF NULL
               E8 BB  D5  051C          TSTL    @F3-B(R11)                      ; CHECK READABILITY
          07 6A   0B  E1  051F  40$:    BBC     #V_F4,(R10),45$                 ; SKIP IF NO COMMAND ADDRESS
  FBD6 CF42   EC AB  D0  0523          MOVL    F4-B(R11),BRKCOM[R2]            ; SET COMMAND STRING
     FB88 CF42     50  D0  052A  45$:    MOVL    R0,BRKADR[R2]                   ; SAVE BREAKPOINT ADDRESS
                FF44  31  0530          BRW     RESET                           ; RESET SCANNER AND RETURN
                          0533      ;
                          0533      ;       SHOBRK
                          0533      ;
                          0533  SHOBRK:                                         ;
             55   01  D0  0533          MOVL    #1,R5                           ; INIT INDEX FOR LOOP
     58   FB7D CF45   D0  0536  10$:    MOVL    BRKADR[R5],R8                   ; GET BREAKPOINT ADDRESS
                  2E  13  053C          BEQL    20$                             ; SKIP IF NULL
             53   55  D0  053E          MOVL    R5,R3                           ; BREAKPOINT NUMBER
                FE70  30  0541          BSBW    CRLF                            ; NEW LINE
                FE16  30  0544          BSBW    OUTDIGIT                        ; BPT NUMBER
                FE65  30  0547          BSBW    OUTSPACE                        ; SPACE
             53   58  D0  054A          MOVL    R8,R3                           ; ADDRESS OF BPT
                FE11  30  054D          BSBW    OUTLONG                         ; OUTPUT ADDRESS
                FE5C  30  0550          BSBW    OUTSPACE                        ; SPACE OVER
     53   FB88 CF45   D0  0553          MOVL    BRKDSP[R5],R3                   ; GET DISPLAY START
                  03  13  0559          BEQL    15$                             ; NONE
                FE03  30  055B          BSBW    OUTLONG                         ; OUTPUT DISPLAY START
```

```
            53   FB9D CF45   D0  055E    942 15$:     MOVL    BRKCOM[R5],R3              ; GET COMMAND STRING ADDRESS
                       06    13  0564    943          BEQL    20$                        ; NONE
                     FE46    30  0566    944          BSBW    OUTSPACE                   ; SPACE ANOTHER
                     FDF5    30  0569    945          BSBW    OUTLONG                    ; AND OUTPUT A LONGWORD
     FFC4 55    01     08    F1  056C    946 20$:     ACBL    #NBRK,#1,R5,10$            ; DO THEM ALL
                     FE3F    31  0572    947          BRW     CRLF                       ; AND EXIT THROUGH CRLF
```

```
                         0575       .SBTTL  GO - START EXECUTION AT SPECIFIED LOCATION
                         0575   ;
                         0575   ;       GO
                         0575   ;
   05 6A    08     E1    0575   GO:     BBC     #V_F1,(R10),PROCED      ; JUST PROCEED IF NO VALUE
54 AB    E0 AB     D0    0579           MOVL    F1=B(R11),SAVPC-B(R11)  ; SET NEW PC
                         057E   ;       BRW     PROCED                  ; FALL INTO PROCEED
                         057E   ;
                         057E   ;       PROCEED
                         057E   ;
                         057E   PROCED:                                 ;
                   04    057E           RET                             ; RETURN
```

```
                                057F          .SBTTL  SEMI-I, PC VALUE
                                057F     ;
                                057F     ;        SEMI-I
                                057F     ;
                      FC9D  30  057F     COLON:   BSBW    ENDEXPR                  ; TERMINATE EXPRESSION
              F8 AB   57    D0  0582              MOVL    R7,PID-B(R11)            ; SET PID FOR PROCESS
                        56  7C  0586              CLRQ    R6                       ; RESET ACCUMULATORS
                            05  0588              RSB                              ;
                                0589
          51   F4 AB        DE  0589     MFYFLGS:MOVAL    MFYFLG-B(R11),R1         ; SET MODIFY FLAG ADDRESS
                        17  11  058D              BRB     VALUE                    ; SET/GET VALUE
                   51   6B  DE  058F     DOT:     MOVAL   CURDOT-B(R11),R1         ; SET ADDRESS OF DOT
                18 6A   1F  E1  0592              BBC     #V_PREG,(R10),VALR       ; WAS IT PROCESSOR REGISTER?
                14 6A   0F  E2  0596              BBSS    #V_PRMODE,(R10),VALR     ; YES, SET PROCESSOR REGISTER MODE
                        12  11  059A              BRB     VALR                     ; READ VALUE
          51   04 AB        DE  059C     QUANT:   MOVAL   QUAN-B(R11),R1           ; SET QUANTITY ADDRESS
                        0C  11  05A0              BRB     VALR                     ; READ VALUE
                                05A2     PROGCTR:                                  ;
          51   54 AB        DE  05A2              MOVAL   SAVPC-B(R11),R1          ; SET PC ADDRESS
          04 6A        08   E1  05A6     VALUE:   BBC     #V_F1,(R10),VALR         ; SKIP IF NO VALUE
          61   E0 AB        D0  05AA              MOVL    F1-B(R11),(R1)           ; SET NEW VALUE FOR PC
                   56   61  D0  05AE     VALR:    MOVL    (R1),R6 ; AND GET VALUE
                      FC52  31  05B1     VALI:    BRW     INFLD                    ; SET FIELD IN PROGRESS
                                05B4     REGISTER:
          55   18 AB        DE  05B4              MOVAL   SAVREG-B(R11),R5         ; SET BASE OF REGISTER AREA
                        02  10  05B8              BSBB    REGCOM                   ; FETCH ADDRESS
                        F5  11  05BA              BRB     VALI                     ; AND USE IT
                      FDFF  30  05BC     REGCOM:  BSBW    GETCHAR                  ; GET SECOND CHAR
        FBA3 CF   10    58  3A  05BF              LOCC    R8,#16,PRIMARY           ; TRANSLATE TO HEX
                                05C5              .IF     DF,SW_PROCESS            ; FOR PROCESS VERSION
                                05C5              BNEQ    10$                      ; LEGAL HEX DIGIT
                                05C5              CMPW    #^A/XI/,-2(R9)           ; CHECK FOR EXIT COMMAND
                                05C5              BNEQ    ERR3                     ; NO, ERROR
                                05C5              $EXIT_S EXITCODE                 ; YES EXIT
                                05C5              .IFF                             ;
                        43  13  05C5              BEQL    ERR3                     ; ERROR, NOT HEX
                                05C7              .ENDC                            ;
                                05C7     10$:                                      ;
          50   10       50  C3  05C7              SUBL3   R0,#16,R0                ; INVERT
             56       6540  DE  05CB              MOVAL   (R5)[R0],R6              ; ACCUMULATE
                            05  05CF              RSB                              ; RETURN
                                05D0
          36 6A        09   E1  05D0     XSET:    BBC     #V_F2,(R10),ERR3         ; ERROR IF NOT TWO FIELDS
   51    E4 AB    04        EF  05D4              EXTZV   #0,#4,F2-B(R11),R1       ; GET REGISTER NUMBER
        51    FB45 CF41     DE  05DA              MOVAL   XREGV[R1],R1             ; AND COMPUTE REGISTER ADDRESS
                        C4  11  05E0              BRB     VALUE                    ; PROCESS VALUE
                                05E2     XREG:                                     ; X-REGISTER VALUE
          55   FB3E CF      DE  05E2              MOVAL   XREGV,R5                 ; SET ADDRESS OF REGISTER VECTOR
                        D3  10  05E7              BSBB    REGCOM                   ; ADDRESS TO R6
                   56   66  D0  05E9              MOVL    (R6),R6                  ; GET VALUE
                        C3  11  05EC              BRB     VALI                     ; AND NOTE INPUT IN FIELD
                                05EE              .ALIGN  LONG                     ; LONGWORD ALIGN EXCEPTION ROUTINES
                                05F0     XDELACV:                                  ; ACCESS VIOLATION HANDLER
                                05F0     MCHK:                                     ; MACHINE CHECK
                                05F0              .IF     NDF,SW_PROCESS           ;
                        5C  D5  05F0              TSTL    AP                       ; CHECK FOR SIMULATOR
                        16  12  05F2              BNEQ    ERR3                     ; YES, SKIP RESET
```

```
                                  05F4                                              ;
                                  05F4                                              ;
                                  05F4              CPUDISP <CLR_780,-              ; *DISPATCH ON CPU TYPE*
                                  05F4                       CLR_750,-              ;
                                  05F4                       CLR_730>
02'  01   00000000'GF    8F       05F4              CASEB   G^EXE$GB_CPUTYPE,#PR$_SID_TYP780,S^#<<30013$-30012$>/2>-1
                                  05FC     30012$:
                         0006'    05FC              .SIGNED_WORD    CLR_780-30012$
                         000B'    05FE              .SIGNED_WORD    CLR_750-30012$
                         000B'    0600              .SIGNED_WORD    CLR_730-30012$
                                  0602     30013$:
                                  0602                                              ;
                                  0602     CLR_780:                                 ; FOR 11/780:
          30   00   DA            0602              MTPR    #0,#PR$_SBIFS           ; CLEAR SBI FAULT
               03   11            0605              BRB     CLR_END                 ; ERROR CLEARED
                                  0607
                                  0607     CLR_730:                                 ; FOR 11/730:
                                  0607     CLR_750:                                 ; FOR 11/750:
          26   0F   DA            0607              MTPR    #^XF,#PR$_MCESR         ; SET 1 TO CLEAR MCHECK ERROR SUMMARY
                                  060A                                              ;
                                  060A     CLR_END:                                 ; *END OF CPU-DEPENDENT CODE*
                                  060A                                              ;
                                  060A                                              ;
                                  060A              .ENDC                           ;
                                  060A     10$:                                     ;
          FB91      31            060A     ERR3:    BRW     ERROR                   ; AND DECLARE ERROR
                                  060D
```

```
                           060D        .SBTTL  REGISTER SAVE AND RESTORE
                           060D
                           060D  ;
                           060D  ;     SAVE - SAVE TARGET REGISTERS, PC, PSL
                           060D  ;
                           060D  SAVE:                                        ;
                           060D        .IF     NDF,SW_PROCESS                 ;
                           060D        SETIPL  #31                            ; DISABLE
          12   1F   DA     060D        MTPR    #31,S^#PR$_IPL
                           0610  ;     JSB     INI$WRITABLE                   ; MAKE THE SYSTEM WRITABLE
  FA5B CF      50   7D     0610        MOVQ    R0,SAVREG                      ; SAVE R0,R1
  51   FA5F CF      9E     0615        MOVAB   SAVR2,R1                       ; SETUP BASE FOR REMAINING REGS
                           061A        .IFF                                   ; FALSE IF PROCESS VERSION
                           061A        $SETAST_S       #0                     ; DISABLE ASTS
                           061A        PUSHAB  -(R0)                          ; SAVE ENABLE VALUE-1
                           061A        MOVPSL  R1                             ; GET CURRENT PSL
                           061A        EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R1,R1       ; ISOLATE CURRENT MODE
                           061A        MULW    #CONTEXTSZ,R1                  ; COMPUTE OFFSET TO PROPER CONTEXT AREA
                           061A        MOVAB   SAVREG[R1],R1                  ; FORM ADDRESS OF REGISTER SAVE
                           061A        MOVL    8(AP),R0                       ; GET POINTER TO MECHANISM
                           061A        MOVQ    12(R0),(R1)+                   ; SAVE R0,R1
                           061A        .ENDC                                  ;
          81   52   7D     061A        MOVQ    R2,(R1)+                       ; SAVE R2,R3
          81   54   7D     061D        MOVQ    R4,(R1)+                       ; SAVE R4,R5
          81   56   7D     0620        MOVQ    R6,(R1)+                       ; SAVE R6,R7
          81   58   7D     0623        MOVQ    R8,(R1)+                       ; SAVE R8,R9
          81   5A   7D     0626        MOVQ    R10,(R1)+                      ; SAVE R10,R11
                           0629        .IF     NDF,SW_PROCESS                 ;
          81   5C   7D     0629        MOVQ    AP,(R1)+                       ; SAVE AP,FP
     81   0C AE     9E     062C        MOVAB   12(SP),(R1)+                   ; ASSUME KERNEL STACK
     81   04 AE     7D     0630        MOVQ    4(SP),(R1)+                    ; SAVE PC,PSL
                           0634        .IFF                                   ;
                           0634        MOVQ    8(FP),(R1)+                    ; SAVE AP,FP
                           0634        SUBL3   #1,@4(AP),R0                   ; GET NUMBER OF ARGS IN SIGNAL
                           0634        MOVAL   @4(AP)[R0],R0                  ; POINT TO PC,PSL
                           0634        MOVAL   8(R0),(R1)+                    ; COMPUTE SP
                           0634        MOVQ    (R0),(R1)+                     ; SAVE PC,PSL
                           0634        .ENDC                                  ;
                           0634        .IF     NDF,SW_PROCESS                 ;
                           0634        MFPR    #PR$_TXCS,(R1)+                ; SAVE CONSOLE TRANSMIT STATUS
          81   22   DB     0634        MFPR    #PR$_TXCS,(R1)+
                           0637        MFPR    #PR$_RXCS,(R1)+                ; SAVE CONSOLE RECVR STATUS
          81   20   DB     0637        MFPR    #PR$_RXCS,(R1)+
               5C   D4     063A        CLRL    AP                             ; ZAP DEVICE ADDRESS BASE
                           063C        .ENDC                                  ;
                           063C        .IF     NDF,SW_PROCESS                 ;
          22   00   DA     063C        MTPR    #0,#PR$_TXCS                   ; CLEAR INTERRUPT ENABLE
          20   00   DA     063F        MTPR    #0,#PR$_RXCS                   ; FOR BOTH TRANSMIT AND RECEIVE
                           0642        .ENDC                                  ;
                           0642        .IF     NDF,SW_PROCESS                 ;
  5B   FA12 CF      9E     0642  20$:  MOVAB   B,R11                          ; AND DATA BASE ADDRESS
                           0647        .IFF                                   ; FALSE FOR PROCESS VERSION
                           0647        MOVAB   W^<B-<SAVPSL+4>>(R1),R11
                                                                              ; SET BASE OF CONTEXT AREA
                           0647        MOVL    (SP)+,ASTEN-B(R11)             ; SAVE AST ENABLE
                           0647        .ENDC                                  ;
     5A   DC AB     9E     0647        MOVAB   STATUS-B(R11),R10              ; SET STATUS BASE
     59   AC AB     9E     064B        MOVAB   INBUF-B(R11),R9                ; POINT TO INPUT BUFFER
```

```
                  69   94  064F             CLRB    (R9)                           ; MAKE BUFFER EMPTY
                           0651             .IF     NDF,SW_PROCESS                 ;
                0084   30  0651             BSBW    GETSCB                         ; GET BASE OF SCB
        FB0A CF 04 A0  D0  0654             MOVL    4(R0),MCHKSAV                  ; SAVE ORIGINAL MCHK VECTOR
          04 A0 93 AF  9E  065A             MOVAB   MCHK,4(R0)                     ; SET TO XDELTA VECTOR
          20 A0 8E AF  9E  065F             MOVAB   XDELACV,^X20(R0)               ; SET ACCESS VIOLATION VECTOR
          24 A0 89 AF  9E  0664             MOVAB   XDELACV,^X24(R0)               ; SET PG FAULT VECTOR
          18 A0 84 AF  9E  0669             MOVAB   XDELACV,^X18(R0)               ; SET RESERVED OPERAND HANDLER
   50  08 AE   02  18  EF  066E             EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,8(SP),R0    ; GET MODE
                   07  13  0674             BEQL    30$                            ; CORRECT ALREADY IF KERNEL
             50    00  C0  0676             ADDL    #PR$_KSP,R0                    ; COMPUTE PROCESSOR REGISTER
                           0679             MFPR    R0,SAVSP-B(R11)                ; AND SAVE CORRECT SP
          50 AB    50  DB  0679                     MFPR     R0,SAVSP-B(R11)
                           067D             .ENDC                                  ;
                FDF7   31  067D     30$:    BRW     RESET                          ; RESET SCANNER
                           0680
                           0680     ;
                           0680     ;       RESTORE - RESTORE TARGET REGISTERS
                           0680     ;
                           0680     RESTORE:                                       ; RESTORE EVERYTHING
                           0680             .IF     NDF,SW_PROCESS                 ;
          04 AE    54 AB   7D  0680         MOVQ    SAVPC-B(R11),4(SP)             ; SET PC,PSL
                           0685             .IFF                                   ; FALSE IF PROCESS
                           0685             SUBL3   #1,@4(AP),R0                   ; GET SIGNAL ARG COUNT
                           0685             MOVAL   @4(AP)[R0],R0                  ; COMPUTE ADDRESS OF PC,PSL
                           0685             MOVQ    SAVPC-B(R11),(R0)              ; STORE UPDATED PC,PSL
                           0685             .ENDC                                  ;
                           0685     RESTORR:                                       ; RESTORE REGISTERS ONLY
                           0685             .IF     NDF,SW_PROCESS                 ;
                   51  10  0685             BSBB    GETSCB                         ; GET BASE OF SCB
  20 A0    00000000'EF  9E  0687            MOVAB   EXE$ACVIOLAT,^X20(R0)          ; RESTORE ACCESS VECTOR
  24 A0    00000000'EF  9E  068F            MOVAB   MMG$PAGEFAULT,^X24(R0)         ; AND PAGE FAULT VECTOR
       04 A0    FAC9 CF  D0  0697           MOVL    MCHKSAV,4(R0)                  ; RESTORE MACHINE CHECK VECTOR
  18 A0    00000000'EF  9E  069D            MOVAB   EXE$ROPRAND,^X18(R0)           ; RESTORE RESERVED OPERAND VECTOR
                   5C  B5  06A5             TSTW    AP                             ; CHECK FOR CONSOLE
                   0A  12  06A7             BNEQ    10$                            ; NO, OTHER DEVICE
             22 5C AB  DA  06A9             MTPR    SAVOCR-B(R11),#PR$_TXCS        ; RESTORE INITIAL TX STATUS
             20 60 AB  DA  06AD             MTPR    SAVRXCS-B(R11),#PR$_RXCS;      AND INITIAL RECEIVER STATE
                   09  11  06B1             BRB     20$                            ; MERGE WITH COMMON CODE
          04 AC 5C AB  B0  06B3     10$:    MOVW    SAVOCR-B(R11),OUTCR(AP)        ; RESTORE OUTPUT CSR
             6C 5E AB  B0  06B8             MOVW    SAVRCR-B(R11),RDCR(AP)         ; AND INPUT CSR CONTENT
                           06BC             .IFF                                   ;
                           06BC             PUSHL   ASTEN-B(R11)                   ; SAVE AST ENABLE
                           06BC             .ENDC                                  ;
             51 20 AB  9E  06BC     20$:    MOVAB   SAVR2-B(R11),R1                ; SET BASE FOR RESTORE
                52 81  7D  06C0             MOVQ    (R1)+,R2                       ; RESTORE R2,R3
                54 81  7D  06C3             MOVQ    (R1)+,R4                       ; RESTORE R4,R5
                56 81  7D  06C6             MOVQ    (R1)+,R6                       ; RESTORE R6,R7
                58 81  7D  06C9             MOVQ    (R1)+,R8                       ; RESTORE R8,R9
                5A 81  7D  06CC             MOVQ    (R1)+,R10                      ; RESTORE R10,R11
                           06CF             .IF     NDF,SW_PROCESS                 ;
                5C 81  7D  06CF             MOVQ    (R1)+,AP                       ; RESTORE AP,FP
          50  F99A CF  7D  06D2             MOVQ    SAVREG,R0                      ; RESTORE R0,R1
                           06D7             .IFF                                   ; FALSE IF PROCESS VERSION
                           06D7             MOVQ    (R1)+,8(FP)                    ; SET NEW VALUES FOR AP,FP
                           06D7             MOVL    8(AP),R0                       ; GET MECHANISM POINTER
                           06D7             MOVQ    <SAVREG-SAVSP>(R1),12(R0)              ; STORE UPDATED R0,R1
          06D7              MOVPSL  R1                              ; GET CURRENT PSL
          06D7              EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R1,R1      ; GET CURRENT MODE
          06D7              BBCC    R1,DBGACTIVE,30$                ; CLEAR ACTIVE BIT FOR MODE
          06D7         30$:                                         ;
          06D7              TSTL    (SP)+                           ; CHECK FOR AST ENABLE
          06D7              BEQL    35$                             ; NO
          06D7              $SETAST_S        #1                     ; RE- ENABLE AST RECOGNITION
          06D7         35$:                                         ;
          06D7              .ENDC                                   ;
          06D7              .IF     NDF,SW_PROCESS                  ;
          06D7         ;    JSB     INI$RDONLY                      ; REPROTECT THE SYSTEM CODE
          06D7              .ENDC                                   ;
      05  06D7              RSB                                     ; AND RETURN
```

```
                       06D8          .SBTTL  GET SCB ADDRESS
                       06D8
                       06D8  ;
                       06D8  ; SUBROUTINE GETSCB IS CALLED TO GET THE PHYSICAL OR VIRTUAL
                       06D8  ; ADDRESS OF THE CURRENT SCB.
                       06D8  ;
                       06D8  ; INPUTS:          NONE
                       06D8  ;
                       06D8  ; OUTPUTS:         R0 = SCB ADDRESS
                       06D8  ;                  OTHER REGISTERS PRESERVED
                       06D8  ;
                       06D8
                       06D8          .IF     NDF,SW_PROCESS          ; NOT FOR PROCESS VERSION
                       06D8  GETSCB: MFPR    #PR$_MAPEN,R0           ; GET MAPPING STATUS
         50    38   DB 06D8          MFPR    #PR$_MAPEN,R0
               05   12 06DB          BNEQ    10$                     ; BRANCH IF MAPPING ENABLED
                       06DD          MFPR    #PR$_SCBB,R0            ; ELSE GET PHY ADDR OF SCB
         50    11   DB 06DD          MFPR    #PR$_SCBB,R0
               07   11 06E0          BRB     20$                     ; JOIN COMMON RETURN
  50    00000000'EF DE 06E2  10$:    MOVAL   SCB$AL_BASE,R0          ; IF MAPPING ENABLED, GET SCB VA
                    05 06E9  20$:    RSB                             ; RETURN
                       06EA          .ENDC                           ;
```

```
                               06EA                .SBTTL  BPT TRAP HANDLER
                               06EA        ;
                               06EA        ;       HANDLE BREAKPOINT TRAPS
                               06EA        ;
 00 20 54 41 20 4B 52 42 20    06EA  BMSG:         .ASCIZ  / BRK AT /                    ; BREAK POINT MESSAGE
                               06F3                .ALIGN  LONG                          ; LONGWORD ALIGNMENT
                               06F4                .IF     NDF,SW_PROCESS                ; EXEC VERSION
                               06F4  XDELBPT::                                           ; XDELTA BPT ENTRY
                               06F4                .IFF                                  ;
                               06F4  XDELBPT:                                            ; DELTA BPT ENTRY
                               06F4                .ENDC                                 ;
                  FF16    30   06F4                BSBW    SAVE                          ; SAVE REGS AND DISABLE
                  00D3    30   06F7                BSBW    GETBPTX                       ; GET INDEX OF BPT
                    53    D5   06FA                TSTL    R3                            ; CHECK FOR MATCH
                    10    12   06FC                BNEQ    10$                           ; YES, FOUND IT
                  FF84    30   06FE                BSBW    RESTORR                       ; RESTORE REGISTERS ONLY
                               0701                .IF     NDF,SW_PROCESS                ;
            7E    06 AE   9A   0701                MOVZBL  6(SP),-(SP)                   ; GET IPL
                               0705                ENBINT                                ; ENABLE
               12    8E   DA   0705                        MTPR    (SP)+,S^#PR$_IPL
          00000000'EF     17   0708                JMP     EXE$BREAK                     ; AND HANDLE NORMALLY
                               070E                .IFF                                  ; FALSE IF PROCESS VERSION
                               070E        ;
                               070E        ;       ***** UNEXPECTED BREAKPOINT *****
                               070E                CLRL    R0                            ; RETURN FALSE
                               070E                RET                                   ;
                               070E                .ENDC                                 ;
               6A    18   88   070E  10$:          BISB    #<<1@V_TBIT>!<1@V_ATBRK>>,(R10) ; SET STATUS
                               0711  30$:                                                ;
                  0081    30   0711                BSBW    UNBRK                         ; RESTORE OPCODES
         38 58 AB    04   E0   0714                BBS     #PSL$V_TBIT,SAVPSL-B(R11),PROCEED       ; PROCEED IF BPT AND TBIT
               55    53   D0   0719                MOVL    R3,R5                         ; SAVE BPT NUMBER
                  FC95    30   071C                BSBW    CRLF                          ; OUTPUT CR/LF PAIR
                  FC3B    30   071F                BSBW    OUTDIGIT                      ; OUTPUT BPT NUMBER
            54    C5 AF   9E   0722                MOVAB   BMSG,R4                       ; MSG ADDRESS
                  FC55    30   0726                BSBW    OUTZSTRING                    ; OUTPUT ASCIIZ
            53    54 AB   D0   0729                MOVL    SAVPC-B(R11),R3               ; OUTPUT PC
                  FC31    30   072D                BSBW    OUTLONG                       ; OUTPUT HEX LONGWORD
                  FC7C    30   0730                BSBW    OUTSPACE                      ; SEND SPACE
       51    F9A8 CF45    D0   0733                MOVL    BRKDSP[R5],R1                 ; GET ADDRESS TO DISPLAY
                    06    13   0739                BEQL    40$                           ; NONE
               6B    51   D0   073B                MOVL    R1,CURDOT-B(R11)              ; SET AS CURRENT DOT
                  FBB6    30   073E                BSBW    LOCPROMPT                     ; AND DISPLAY
       51    F9BA CF45    D0   0741  40$:          MOVL    BRKCOM[R5],R1                 ; GET COMMAND STRING ADDRESS
                    03    13   0747                BEQL    GETCMD                        ; NONE GET COMMAND
               59    51   D0   0749                MOVL    R1,R9                         ; SET TO SCAN STORED COMMAND
                               074C  GETCMD:                                             ; GET COMMANDS
                               074C
          FA49 CF    6C   FA   074C                CALLG   (AP),DCOM                     ; PERFORM DEBUG COMMANDS
                               0751  PROCEED:                                            ; PROCEED
                    57    10   0751                BSBB    SETBRK                        ; SET BREAKPOINTS
            09 6A    03   E5   0753                BBCC    #V_TBIT,(R10),50$             ; TEST AND CLR TRACE FLAG
         00 58 AB    04   E2   0757  30$:          BBSS    #PSL$V_TBIT,SAVPSL-B(R11),40$    ; SET TBIT
                               075C  40$:                                                ;
                               075C                .IF     DF,SW_PROCESS                 ; FOR PROCESS VERSION
                               075C                CMPB    #2,@SAVPC-B(R11)              ; CHECK FOR REI OPCODE
                               075C                BNEQ    45$                           ; NO, NOTHING SPECIAL
```

```
                      075C            EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,SAVPSL-B(R11),R0      ; GET NEW MODE
                      075C            MULW    #CONTEXTSZ,R0                  ; SCALE BY PER MODE CONTEXT AREA SIZE
                      075C            MOVAB   STATUS-B(R0),R10               ; POINT TO NEW FLAGS
                      075C            .ENDC                                  ;
    00 6A   05    E2  075C  45$:      BBSS    #V_TBITOK,(R10),50$            ; SET TBIT EXPECTED
           FF1D   30  0760  50$:      BSBW    RESTORE                        ; RESTORE EVERYTHING
                      0763            .IF     NDF,SW_PROCESS                 ;
                  02  0763            REI                                    ; AND RETURN
                      0764            .IFF                                   ; FALSE IF PROCESS VERSION
                      0764            MOVL    #1,R0                          ; RETURN TRUE
                      0764            RET                                    ;
                      0764            .ENDC                                  ;
                      0764
```

```
                           0764          .SBTTL  TBIT EXCEPTION HANDLER
                           0764  ;
                           0764  ;       HANDLER FOR TBIT EXCEPTION
                           0764  ;
                           0764          .ALIGN  LONG                    ; LONGWORD ALIGNED
                           0764          .IF     NDF,SW_PROCESS          ;
                           0764  XDELTBIT::                              ; XDELTA TBIT HANDLER
                           0764          .IFF                            ;
                           0764  XDELTBIT:                               ;
                           0764          .ENDC                           ;
            FEA6   30      0764          BSBW    SAVE                    ; SAVE AND DISABLE
      10 6A   05   E4      0767          BBSC    #V_TBITOK,(R10),XDELDBG ; BR IF TBIT EXPECTED
            FF17   30      076B          BSBW    RESTORR                 ; RESTORE REGISTERS
                           076E          .IF     NDF,SW_PROCESS          ;
      7E   06 AE   9A      076E          MOVZBL  6(SP),-(SP)             ; GET IPL FOR ENABLE
                           0772          ENBINT                          ; ENABLE
        12   8E    DA      0772          MTPR    (SP)+,S^#PR$_IPL
    00000000'EF    17      0775          JMP     EXE$TBIT                ; OTHERWISE LET EXEC HANDLE
                           077B          .IFF                            ; FALSE IF PROCESS VERSION
                           077B          CLRL    R0                      ; RESIGNAL
                           077B          RET                             ; UNEXPECTED TBIT EXCEPTION
                           077B          .ENDC                           ;
                           077B  XDELDBG:                                ; COMMON WITH DEBUG EXCEPTION
      58 AB   10   CA      077B          BICL    #<1@PSL$V_TBIT>,SAVPSL-B(R11)   ; CLEAR TBIT IN PSL
              14   10      077F          BSBB    UNBRK                   ; REPLACE OPCODES
      CC 6A   04   E4      0781          BBSC    #V_ATBRK,(R10),PROCEED  ; CHECK FOR PROCEED
                           0785  ;
                           0785  ;       OUTPUT STEP MESSAGE
                           0785  ;
      6B   54 AB   D0      0785          MOVL    SAVPC-B(R11),CURDOT-B(R11)      ; SET ADDRESS
                           0789          IFNORD  #4,@CURDOT-B(R11),GETCMD        ; SKIP DISPLAY IF NOT READABLE
 00 BB   04   00   0C      0789          PROBER  #0,#4,@CURDOT-B(R11)
              BC   13      078E          BEQL    GETCMD
            FB64   30      0790          BSBW    LOCPROMPT               ; PROMPT WITH ADDRESS/CONTENT
              B7   11      0793          BRB     GETCMD                  ; GO GET COMMANDS
                           0795
```

```
                          0795         .SBTTL  UNBRK - RESTORE OPCODES FOR BREAKPOINTS
                          0795   ;
                          0795   ;       UNBRK
                          0795   ;
                          0795   UNBRK:                                  ;
           51    08   D0  0795          MOVL    #NBRK,R1                ; INIT LOOP
   50  F91B CF41   D0  0798   10$:   MOVL    BRKADR[R1],R0           ; GET BREAKPOINT ADDRESS
                 06   13  079E          BEQL    20$                     ; SKIP IF NOT ENABLED
                          07A0          .IF     DF,SW_PROCESS           ;
                          07A0          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; SAVE REGS FOR PROTECTION CHANGE
                          07A0          MOVL    R0,R4                   ; FORM INADR RANGE FOR SET PROTECTION
                          07A0          MOVL    R0,R5                   ;
                          07A0          BSBW    SETWRT                  ; SET PAGE WRITABLE
                          07A0          MOVQ    (SP),R0                 ; RESTORE R0,R1
                          07A0          .ENDC                           ;
   60  F936 CF41   90  07A0          MOVB    BRKOP[R1],(R0)          ; RESTORE OPCODE
                          07A6          .IF     DF,SW_PROCESS           ;
                          07A6          BSBW    REPROT                  ; RESTORE PROTECTION
                          07A6          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; RESTORE REGISTERS
                          07A6          .ENDC                           ;
              EF 51   F5  07A6   20$:   SOBGTR  R1,10$                  ; DO THEM ALL
                      05  07A9          RSB                             ; AND RETURN
                          07AA
```

```
                        07AA           .SBTTL  SETBRK - SET BREAK POINT INSTRUCTIONS
                        07AA      ;
                        07AA      ;            SETBRK
                        07AA      ;
       51   08     D0   07AA      SETBRK: MOVL    #NBRK,R1                        ; INIT LOOP
 50  F906 CF41     D0   07AD      10$:    MOVL    BRKADR[R1],R0                   ; GET ADDRESS
            14     13   07B3              BEQL    20$                             ; SKIP IF NOT ENABLED
 F920 CF41  60     90   07B5              MOVB    (R0),BRKOP[R1]                  ; SAVE OPCODE
       6A   18     93   07BB              BITB    #<<1@V_TBIT>!<1@V_ATBRK>>,(R10) ; CHECK FOR TRACE
            06     13   07BE              BEQL    15$                             ; NO TRACE, SET ANYWAY
    54 AB   50     D1   07C0              CMPL    R0,SAVPC-B(R11)                 ; CHECK FOR AT BPT
            03     13   07C4              BEQL    20$                             ; YES, DONT SET IT
                        07C6      15$:                                            ;
                        07C6              .IF     DF,SW_PROCESS                   ;
                        07C6              PUSHR   #^M<R0,R1,R2,R3,R4,R5>          ; SAVE REGISTERS FOR PROTECTION CHANGE
                        07C6              MOVL    R0,R4                           ; SET START ADDRESS OF RANGE
                        07C6              MOVL    R0,R5                           ; AND END ADDRESS
                        07C6              BSBW    SETWRT                          ;SET PAGE WRITABLE
                        07C6              MOVL    (SP),R0                         ; RESTORE BPT ADDRESS
                        07C6              .ENDC                                   ;
       60   03     90   07C6              MOVB    #3,(R0)                         ; SET BREAKPOINT OPCODE
                        07C9              .IF     DF,SW_PROCESS                   ;
                        07C9              BSBW    REPROT                          ; RESTORE ORIGINAL PROTECTION VALUE
                        07C9              POPR    #^M<R0,R1,R2,R3,R4,R5>          ; AND REGISTERS
                        07C9              .ENDC                                   ;
       E1 51       F5   07C9      20$:    SOBGTR  R1,10$                          ; DO THEM ALL
                   05   07CC              RSB                                     ; AND RETURN
                        07CD
```

```
                    07CD            .SBTTL  GETBPTX - GET INDEX FOR BREAKPOINT
                    07CD       ;
                    07CD       ;        GETBPTX
                    07CD       ;
                    07CD       GETBPTX:                                          ;
           53  08  D0  07CD            MOVL    #NBRK,R3                          ; INIT LOOP
F8E1 CF43  54  AB  D1  07D0  10$:      CMPL    SAVPC-B(R11),BRKADR[R3]           ; IS THIS A BPT?
               03  13  07D7            BEQL    20$                               ; YES
           F4  53  F5  07D9            SOBGTR  R3,10$                            ; NO, CONTINUE
                   05  07DC  20$:      RSB                                       ; RETURN
```

```
XDE
Sym

ADD
AST
B
BLA
BMS
BR
BR
BR
BR
BR
BSL
CLF
CLF
CLF
CLF
COL
COM
COM
COM
CR
CRL
CUF
CUF
DC(
DEF
DIV
DOT
DQL
DTY
END
END
EQL
EQL
ERF
ERF
ERF
ERF
ESC
EXI
EXI
EXI
EXI
EXI
EXI
F1
F2
F3
F4
F5
FC
FE
GE
GE
GE
GE
GL
GO
```

```
                   07DD       .SBTTL  QUOTE - INPUT CHARACTER STRING
                   07DD  ;
                   07DD  ;        QUOTE - START CHARACTER STRING INPUT
                   07DD  ;
                   07DD  QUOTE:                                       ;
 55    6B    D0    07DD          MOVL    CURDOT-B(R11),R5              ; POINT TO STRING BUFFER
     FBDB    30    07E0  5$:     BSBW    GETCHAR                       ; GET CHARACTER
 58    27    91    07E3          CMPB    #QUOT,R8                      ; CHECK FOR QUOTE
       05    13    07E6          BEQL    10$                           ; YES, END OF STRING
 85    58    90    07E8          MOVB    R8,(R5)+                      ; INSERT IN BUFFER
       F3    11    07EB          BRB     5$                            ; AND CONTINUE
 6B    55    D0    07ED  10$:    MOVL    R5,CURDOT-B(R11)              ; SAVE NEW DOT
             05    07F0          RSB                                   ; RETURN
```

```
                           07F1              .SBTTL  DEPOSIT
                           07F1        ;
                           07F1        ;         DEPOSIT DATA
                           07F1        ;
                           07F1        DEPOSIT:                                  ;
         1D 6A   1F   E0   07F1              BBS     #V_PREG,(R10),40$           ; BR IF PROCESSOR REGISTER
                           07F5              .IF     DF,SW_PROCESS
                           07F5              MOVL    CURDOT-B(R11),R4            ; GET CURRENT DOT
                           07F5              TSTL    PID-B(R11)                  ; CHECK FOR ARBITRARY PROCESS DEPOSIT
                           07F5              BNEQ    50$                         ; BR IF YES
                           07F5              .ENDC                               ;
                           07F5              CASE    CURTYPE-B(R11),TYPE=B,<-         ; SWITCH ON TYPE
                           07F5                      10$,-                       ; BYTE
                           07F5                      20$,-                       ; WORD
                           07F5                      30$,-                       ; LONG
                           07F5                      >                           ;
 02'   00    FE AB   8F    07F5              CASEB   CURTYPE-B(R11),#0,S^#<<30020$-30019$>/2>-1
                           07FA        30019$:
                    0006'  07FA              .SIGNED_WORD    10$-30019$
                    000C'  07FC              .SIGNED_WORD    20$-30019$
                    0012'  07FE              .SIGNED_WORD    30$-30019$
                           0800        30020$:
                           0800              .IF     NDF,SW_PROCESS              ;
    00 BB    E0 AB   90    0800        10$:  MOVB    F1-B(RT1),@CURDOT-B(R11);   STORE BYTE
                     05    0805              RSB                                 ; RETURN
    00 BB    E0 AB   B0    0806        20$:  MOVW    F1-B(R11),@CURDOT-B(R11);   STORE WORD
                     05    080B              RSB                                 ; RETURN
    00 BB    E0 AB   D0    080C        30$:  MOVL    F1-B(R11),@CURDOT-B(R11);   STORE LONG
                     05    0811              RSB                                 ; RETURN
       6B    E0 AB   DA    0812        40$:  MTPR    F1-B(R11),CURDOT-B(R11)     ; SET VALUE IN PROCESSOR REGISTER
                     05    0816              RSB                                 ;
                           0817              .IFF                                ; FALSE IF PROCESS VERSION
                           0817        10$:                                      ; BYTE DEPOSIT
                           0817              MOVL    R4,R5                       ; START AND END ADDRESSES EQUAL
                           0817              BSBW    SETWRT                      ; SET WRITABLE, OLD PROT TO R2
                           0817              MOVB    F1-B(R11),(R4)              ; STORE BYTE
                           0817              BSBW    REPROT                      ; RESTORE PROTECTION
                           0817              RSB                                 ;
                           0817                                                  ;
                           0817        20$:  ADDL3   #1,R4,R5                    ; WORD DEPOSIT, FORM END ADDRESS
                           0817              BSBW    SETWRT                      ; SET WRITABLE
                           0817              MOVW    F1-B(R11),(R4)              ; STORE WORD
                           0817              BSBW    REPROT                      ; RESTORE PROTECTION
                           0817              RSB                                 ;
                           0817                                                  ;
                           0817        30$:  ADDL3   #3,R4,R5                    ; LONGWORD DEPOSIT, FORM END ADDRESS
                           0817              BSBW    SETWRT                      ; SET WRITABLE
                           0817              MOVL    F1-B(R11),(R4)              ; STORE LONG WORD
                           0817              BSBW    REPROT                      ; RESTORE PROTECTION
                           0817              RSB                                 ;
                           0817                                                  ;
                           0817        40$:                                      ; PROCESSOR REGISTER
                           0817              $CMKRNL_S       B^DEPPREG,(AP)      ; DEPOSIT IN PROCESSOR REGISTER
                           0817              RSB                                 ;
                           0817        50$:                                      ; DEPOSIT IN ARBITRARY PROCESS
                           0817              CASE    CURTYPE-B(R11),TYPE=B,<-         ; SWITCH ON TYPE
                           0817                      60$,-                       ; BYTE
```

```
XDE
Pse

PSE
---
.
$AE
Z$D

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

The
872
The
184
24

Mac
---
_$2
_$2
_$2
TOT

139

The

MAC
```

```
0817                          70$,-                            ; WORD
0817                          80$>                             ; LONGWORD
0817               RSB                                         ;
0817    60$:       PUSHAB     W^DPBYTE                         ; SET ADDRESS OF BYTE ROUTINE
0817               BRB        90$                              ;
0817    70$:       PUSHAB     W^DPWORD                         ; SET ADDRESS OF WORD ROUTINE
0817               BRB        90$                              ;
0817    80$:       PUSHAB     W^DPLONG                         ; SET ADDRESS OF LONG ROUTINE
0817    90$:       PUSHL      PID-B(R11)                       ; SET PID OF TARGET
0817               PUSHL      CURDOT-B(R11)                    ; ADDRESS FOR STORE
0817               PUSHL      F1-B(R11)                        ; VALUE TO STORE
0817               PUSHL      #4                               ; ARGUMENT COUNT
0817               MOVL       SP,R0                            ; POINTER TO ARGUMENT LIST
0817               TSTL       MFYFLG-B(R11)                    ; CHECK FOR STORE ENABLED
0817               BEQL       100$                             ; BR IF NOT
0817               $CMKRNL_S       W^QGET,(R0)                 ; CALL TO QUEUE REQUEST
0817    100$:      ADDL       #20,SP                           ; CLEAN STACK
0817               RSB                                         ; AND RETURN
0817
0817    DEPPREG:.WORD         0                                ; DEPOSIT INTO PROCESSOR REGISTER
0817               MOVAB      W^PREXC,(FP)                     ; SET EXCEPTION HANDLER
0817               MTPR       F1-B(R11),CURDOT-B(R11)          ; PLACE FIELD VALUE IN REG
0817               MOVL       #1,R0                            ; RETURN SUCESS
0817               RET                                         ;
0817
0817    PREXC:     .WORD      0                                ; PROCESSOR REGISTER EXCEPTION HANDLER
0817               ADDL3      #4,8(AP),R1                      ; POINT TO EXCEPTION FP
0817               MOVL       (R1),12(FP)                      ; SET AS RETURN FP
0817               MOVAB      B^10$,16(FP)                     ; SET RETURN ADDRESS
0817    10$:       MOVZWL     #1,R0                ;SET NORMAL STATUS
0817               RET                                         ; AND RETURN
0817
0817               .ENDC
```

```
                         0817         .SBTTL  EXECUTE - PERFORM COMMAND STRING
                         0817    ;
                         0817    ;       EXECUTE
                         0817    ;
                         0817    EXECUTE:                                        ;
09 6A    08        E1    0817            BBC     #V_F1,(R10),10$                 ; EXIT IF NO ADDRESS
59    E0 AB        D0    081B            MOVL    F1=B(R11),R9                    ; SET CHAR STRING
         03        12    081F            BNEQ    10$                             ; NOT NULL
       F981        31    0821            BRW     SUPERST                         ; SUPER RESET
                   05    0824    10$:    RSB                                     ; RETURN
                         0825
                         0825  1466        .SBTTL  P - PROCESSOR REGISTER PREFIX
                         0825  1467 ;
                         0825  1468 ;      SET PROCESSOR REGISTER MODE
                         0825  1469 ;
                         0825  1470 PREG:                                          ; PROCESSOR REGISTER MODE
        00 6A    0F   E2 0825  1471        BBSS    #V_PRMODE,(R10),10$             ; SET PROCESSOR REG FLAG
                      05 0829  1472 10$:   RSB                                     ; RETURN
```

```
082A            .SBTTL  PROCESS DEBUGGER INITIALIZATION
082A
082A            .IF     DF,SW_PROCESS                   ;
082A    SALUTE: .ASCIZ  <CR><LF>/DELTA Version X2.1/<CR><LF>      ;
082A
082A    TEST:                                           ; START ADDRESS OF IMAGE ENTRY
082A    XDT$START::                                     ; GLOBAL START ADDRESS FOR CLI DEBUG
082A            .WORD   0
082A    DELTA_START:                                    ; START ADDRESS FOR DEBUGGER ENTRY
082A            $WAKE_S                                 ; NULL WAKE AND
082A            $HIBER_S                                ; HIBERNATE TO GET SYNCHRONIZED
082A            MOVAB   TERMASK,TERMASKD+4              ; RELOCATE TERMINATOR MASK DESCR
082A            MOVAB   TTSTR,TTNAMD+4                  ; RELOCATE DESCRIPTOR
082A            MOVAB   EXIHANDLE,EXIHADR               ;
082A            MOVAB   EXITCODE,EXCODA                 ; RELOCATE EXIT HANDLER ARGS
082A            CALLG   (AP),B^INITCALL                 ; GENERATE CALL FRAME
082A            RET                                     ;
082A
082A    NOBRK:  MOVL    4(AP),AP                        ; GET EXCEPTION ARGUMENT LIST
082A            BRW     EXCEPT+2                        ; AND GOTO EXCEPTION HANDLER
082A
082A    INITCALL:
082A            .WORD   0                               ; ENTRY MASK
082A            MOVAB   W^CATCHALL,(FP)                 ; SET CATCHALL EXCEPTION HANDLER
082A            $DCLEXH_S       EXITBLK                 ; DECLARE USER MODE EXIT HANDLER
082A            $CMKRNL_S       W^SETEXC,(AP)           ; SET EXCEPTION VECTORS
082A            $SETEXV_S       ADDRES=W^EXCEPT,-       ;
082A                            ACMODE=#3,-
082A                            VECTOR=#0               ; SET PRIMARY FOR USER
082A            $SETEXV_S       ADDRES=W^CATCHALL,-     ; SET LAST CHANCE HANDLER
082A                            ACMODE=#3,-             ; FOR USER MODE
082A                            VECTOR=#2               ; SPECIFY LAST CHANCE HANDLER
082A            $ASSIGN_S       TTNAMD,TTCHAN           ; ASSIGN DEVICE
082A            BLBS    R0,10$                          ; CONTINUE IF SUCCESS
082A            RET                                     ; ELSE EXIT WITH ERROR CODE IN R0
082A    10$:    MOVAB   SALUTE,R4                       ; SET ADDRESS OF SALUTATION
082A            BSBW    OUTZSTRING                      ; OUTPUT IT
082A            BBS     #CLI$V_DBGEXCP,24(AP),NOBRK     ; BR IF LATER INVOCATION
082A                                                    ; VIA $DEBUG COMMAND
082A            CALLG   (AP),B^20$                      ; CREATE TOP CALL FRAME
082A            RET                                     ;
082A    20$:    .WORD   0                               ; NULL ENTRY MASK
082A            ADDL    #4,4(AP)                        ; ADVANCE STARTING ADDRESS POINTER
082A            MOVPSL  -(SP)                           ; SAVE PSL
082A            ADDL3   #2,@4(AP),-(SP)                 ; FETCH CURRENT STARTING ADDRESS
082A            MOVZWL  #SS$_DEBUG,-(SP)                ; SET EXCEPTION CODE
082A            PUSHL   #3                              ; SIGNAL ARG COUNT
082A            MOVL    SP,R0                           ; SAVE POINTER
082A            MOVQ    R0,-(SP)                        ; SAVE PHONY R0,R1
082A            PUSHL   #0                              ; DEPTH
082A            PUSHL   FP                              ; FP
082A            PUSHL   #4                              ; ARG COUNT
082A            PUSHL   SP                              ; POINTER TO MECH
082A            PUSHL   R0                              ; POINTER TO SIGNAL
082A            CALLS   #2,W^EXCEPT                     ; SIGNAL PHONY EXCEPTION
082A            ADDL    #12,SP                          ; CLEAN BACK TO R0,R1
082A            MOVQ    (SP)+,R0                        ; RESTORE R0,R1
```

```
082A                    ADDL    #8,SP                           ; CLEAN BACK TO PC,PSL
082A                    REI                                     ; RETURN TO TARGET PROGRAM
082A
082A
082A    SETEXC: .WORD   0                                       ; ENTRY MASK
082A                    $SETEXV_S       ADDRES=B^EXCEPT,-       ;
082A                                    PRVHND=KCOND,-          ;
082A                                    ACMODE=#0               ; SET KERNEL
082A                    $SETEXV_S       ADDRES=W^CATCHALL,-     ;
082A                                    ACMODE=#0,-             ; SET KERNEL MODE LAST CHANCE HANDLER
082A                                    VECTOR=#2               ; SPECIFY LAST CHANCE VECTOR
082A    ;------------------------------------------------------------------------------------------
082A                    $SETEXV_S       ADDRES=B^EXCEPT,-       ;
082A                                    PRVHND=ECOND,-          ;
082A                                    ACMODE=#1               ; SET EXEC MODE EXCEPTION HANDLER
082A                    $SETEXV_S       ADDRES=W^CATCHALL,-     ;
082A                                    ACMODE=#1,-             ; SET EXEC MODE LAST CHANCE HANDLER
082A                                    VECTOR=#2               ; SPECIFY LAST CHANCE VECTOR
082A    ;------------------------------------------------------------------------------------------
082A                    $SETEXV_S       ADDRES=B^EXCEPT,-       ;
082A                                    PRVHND=SCOND,-          ;
082A                                    ACMODE=#2               ; SET SUPERVISOR MODE EXCEPTION HANDLER
082A                    $SETEXV_S       ADDRES=W^CATCHALL,-     ;
082A                                    ACMODE=#2,-             ; SET SUPERVISOR LAST CHANCE HANDLER
082A                                    VECTOR=#2               ; SPECIFY LAST CHANCE VECTOR
082A                    RET                                     ;
082A
082A    EXCEPT: .WORD   0                                       ; EXCEPTION HANDLER ENTRY MASK
082A                    $SETEXV_S       ADDRES=B^EXCEPT,-       ;
082A                                    ACMODE=#3,-             ;
082A                                    VECTOR=#0               ; RE-ESTABLISH USER PRIMARY VECTOR
082A                    ADDL3   #4,4(AP),R0                     ; GET POINTER TO SIGNAL
082A                    MOVPSL  R1                              ; GET CURRENT PSL
082A                    EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R1,R1       ;
082A                    BBSS    R1,DBGACTIVE,40$                ; BR IF ALREADY ACTIVE
082A                    CMPL    #SS$_TBIT,(R0)                  ; IS IT TBIT?
082A                    BNEQ    10$                             ; NO,
082A    5$:             BRW     XDELTBIT                        ; YES, A TBIT
082A    10$:            CMPL    #SS$_BREAK,(R0)                 ; IS IT BREAKPOINT?
082A                    BNEQ    20$                             ; NO,
082A    15$:            BRW     XDELBPT                         ; YES, A BREAKPOINT
082A    20$:                                                    ; SOME OTHER EXCEPTION
082A                    CMPL    #SS$_UNWINDING,(R0)             ; IS IT UNWINDING
082A                    BEQL    60$                             ; YES
082A                    CMPL    #SS$_COMPAT,(R0)+               ; IS IT COMPATIBILITY MODE EXCEPT?
082A                    BNEQ    30$                             ; NO
082A                    CMPL    #1,(R0)                         ; IS IT COMPATIBILITY BPT?
082A                    BEQL    15$                             ; YES
082A                    CMPL    #7,(R0)                         ; IS IT COMPATIBILITY TBIT?
082A                    BEQL    5$                              ; YES
082A    30$:            CMPL    #SS$_DEBUG,-(R0)                ; IS IT DEBUG EXCEPTION?
082A                    BNEQ    40$                             ; NO,
082A                    BSBW    SAVE                            ; SAVE EVERYTHING
082A                    BRW     XDELDBG                         ; AND TREAT AS FUNNY BPT
082A    40$:                                                    ; UNEXPECTED EXCEPTION
082A                    BBCC    R1,DBGACTIVE,50$                ; CLEAR DEBUG ACTIVE
082A    50$:            CLRL    R0                              ; RETURN FALSE FOR RESIGNAL
```

```
082A             RET                                      ;
082A       60$:  MOVL    #1,R0                            ; IGNORE AND RESIGNAL
082A             RET                                      ;
082A             .PAGE
082A             .SBTTL  HANDLER FOR DEBUG EXCEPTIONS
082A
082A       DBGEXCEP:                                      ;
082A             .WORD   0                                ;
082A             ADDL3   #4,8(AP),R1                      ; POINT TO EXCEPTION FP
082A             MOVL    FP,R0                            ; INIT LINK FOR CALL FRAMES
082A       10$:  CMPL    12(R0),(R1)                      ; IS THIS THE LAST ONE?
082A             BEQL    20$                              ; YES
082A             MOVAB   B^30$,16(R0)                     ; SET FOR RETURN
082A             MOVL    12(R0),R0                        ;
082A             BRB     10$                              ; CONTINUE
082A       20$:  MOVAB   XDELACV,16(R0)                   ; SET RETURN FOR ERROR
082A       30$:  RET                                      ;
082A
082A       CATCHALL:                                      ; CATCHALL EXCEPTION HANDLER
082A             .WORD   0                                ; ENTRY MASK
082A             MOVPSL  R1                               ; GET CURMOD
082A             EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R1,R1        ; ISOLATE CURRENT MODE
082A             BBCS    R1,DBGACTIVE,10$                 ; MUST NOT BE DEBUGGER EXCEPTION
082A             CLRL    R0                               ; RESIGNAL
082A             RET                                      ;
082A       10$:  BSBW    SAVE                             ; SAVE EVERYTHING
082A             ADDL3   #4,4(AP),R0                      ; POINT TO EXCEPTION CODE
082A             MOVL    (R0),R3                          ; GET IT
082A             BSBW    CRLF                             ; OUTPUT CR/LF
082A             BSBW    OUTLONG                          ; OUTPUT EXCEPTION CODE
082A             MOVAB   B^EXCMSG,R4                      ; OUTPUT MESSAGE
082A             BSBW    OUTZSTRING                       ; TEXT FOR EXCEPTION
082A             BRW     XDELDBG                          ; AND DISPLAY INSTRUCTION
082A       EXCMSG: .ASCIZ  / EXCEPTION /                  ;
082A
082A       EXIHANDLE:                                     ; EXIT HANDLER
082A             .WORD   0                                ; ENTRY MASK
082A             BITB    #15,DBGACTIVE                    ; TEST FOR DEBUG ACTIVE IN ANY MODE
082A             BEQL    10$                              ; NO, REPORT EXIT
082A             $CMKRNL_S       CLREXV,(AP)              ; RESET EXCEPTION VECTORS
082A             MOVL    @4(AP),R0                        ; RESTORE
082A             RET                                      ; RETURN
082A       10$:                                           ; PROGRAM EXIT
082A             MOVPSL  -(SP)                            ; BUILD EXCEPTION FRAME
082A             PUSHL   16(FP)                           ;
082A             PUSHL   @4(AP)                           ; EXIT CODE FOR EXCEPTION CODE
082A             PUSHL   #3                               ; ARG COUNT
082A             PUSHR   #^M<R0,R1>                       ;
082A             MOVQ    AP,-(SP)                         ;
082A             PUSHL   #4                               ; MECHANISM COUNT
082A             PUSHL   SP                               ; POINTER TO MECHANISM
082A             PUSHAL  24(SP)                           ; POINTER TO SIGNAL
082A             PUSHL   #2                               ;
082A             MOVL    SP,AP                            ; SET AP FOR EXCEPTION
082A             BSBW    SAVE                             ; SAVE EVERYTHING
082A             MOVAB   B^EXIMSG,R4                      ; DISPLAY EXIT MESSAGE
082A             BSBW    OUTZSTRING                       ; OUTPUT TEXT
```

```
_$2

Sym
---
BUG
CLU
EXE
EXE
EXE
EXE
EXE
IOC
MMG
PMS
PMS
PMS
PMS
SCS
SCS
SCS
SCS
SCS
SCS
SCS
SCS
SCS
SCS
SYS
```

```
082A                MOVL    SAVAP-B(R11),R3          ; GET POINTER TO EXCEPTION ARGLIST
082A                MOVL    4(R3),R3                 ; GET EXIT CODE ADDRESS
082A                BSBW    OUTLONG                  ; DISPLAY IT
082A                $DCLEXH_S       EXITBLK          ; RE-ESTABLISH EXIT HANDLER
082A                MOVPSL  R1                       ; GET CURRENT PSL
082A                EXTZV   #PSL$V_CURMOD,#PSL$S_CURMOD,R1,R1 ; GET CURRENT MODE
082A                BBSS    R1,DBGACTIVE,20$         ; SET DELTA ACTIVE FOR MODE
082A      20$:      BRW     XDELDBG                  ;
082A
082A      EXIMSG: .ASCIZ  <CR><LF>/ EXIT /         ;
082A
082A      CLREXV:                                    ; CLR EXCEPTION VECTORS
082A                .WORD   0                        ; ENTRY MASK
082A                $SETEXV_S       ADDRES=@KCOND,- ;
082A                                ACMODE=#0        ;
082A                $SETEXV_S       ADDRES=@ECOND,- ;
082A                                ACMODE=#1        ;
082A                $SETEXV_S       ADDRES=@SCOND,- ;
082A                                ACMODE=#2        ;
082A                RET                              ;
082A
082A                .PAGE
082A                .SBTTL  SETWRT - SET PAGES WRITABLE
082A
082A      SETWRT:                                    ;
082A                MOVAL   -(SP),R2                 ; ADDRESS FOR RETURN OF PROT
082A                $CMKRNL_S       B^SETPRTK,(R2)   ;
082A                BLBS    R0,10$                   ; CONTINUE IF NO ERROR
082A                CALLG   (R2),B^SETPRTK
082A      10$:      POPR    #^M<R2>                  ; RESTORE PROTECTION VALUE
082A                RSB                              ; RETURN
082A
082A      SETPRTK:.WORD   0                        ;
082A                MOVQ    R5,-(SP)                 ; INADR, START AND END ADDRESSES
082A                MOVL    SP,R1                    ; ADDRESS OF INADR
082A                $SETPRT_S       INADR=(R1),-     ;
082A                                PROT=#PRT$C_UW,-; WRITABLE BY ALL
082A                                ACMODE=#0,-      ;
082A                                PRVPRT=(AP)      ; ADDRESS AT WHICH TO RETURN PROT
082A                MOVL    #1,R0                    ; ALWAYS SUCCESS
082A                RET                              ;
082A
082A      REPROT:                                    ; RESTORE PROTECTION
082A                RSB                              ;
082A                .PAGE
082A                .SBTTL  FETCHP - FETCH DATA FROM ANOTHER PROCESS
082A      FETCHP: CASE    CURTYPE-B(R11),TYPE=B,<-         ;
082A                        10$,-                    ; 0 => BYTE
082A                        20$,-                    ; 1 => WORD
082A                        30$>                     ; 2 => LONG
082A                RSB                              ; UNKNOWN
082A      10$:      PUSHAB  W^FPBYTE                 ; SET FOR BYTE FETCH
082A                BRB     40$
082A      20$:      PUSHAB  W^FPWORD                 ; SET FOR WORD FETCH
082A                BRB     40$
082A      30$:      PUSHAB  W^FPLONG                 ; SET FOR LONGWORD FETCH
082A      40$:      PUSHL   PID-B(R11)               ; PID OF TARGET PROCESS
```

```
_$2
Val
---
000
485
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
800
```

```
082A            PUSHAB  QUAN-B(R11)              ; SET ADDRESS TO RETURN VALUE
082A            PUSHL   CURDOT-B(R11)            ; AND ADDRESS OF VALUE
082A            PUSHL   #4                       ; ARGUMENT COUNT
082A            MOVL    SP,R0                    ; SAVE POINTER TO ARG LIST
082A            $CMKRNL_S       W^QGET,(R0)      ; Q AST FOR DATA FETCH
082A            BLBC    R0,50$                   ; BR IF FAILED
082A            $HIBER_S                         ; WAIT FOR DATA TO RETURN
082A    50$:    ADDL    #20,SP                   ; CLEAN STACK
082A            RSB                              ; AND RETURN DATA
082A            .PAGE
082A            .SBTTL  QGET - QUEUE AST TO GET DATA FROM ANOTHER PROCESS
082A    ;
082A    ;       INPUTS: 04(AP) - LOCATION OF DATA
082A    ;               08(AP) - RETURN LOCATION
082A    ;               12(AP) - PID OF TARGET PROCESS
082A    ;               16(AP) - CODE SEGMENT POINTER
082A    ;
082A            FP_ORIGPID=ACB$L_AST
082A            FP_ADDR=ACB$L_ASTPRM
082A            FP_VALUE=ACB$[_ASTPRM
082A            FP_RETLOC=ACB$[_KAST+4
082A    QGET:   .WORD   ^M<R2,R3,R4,R5>          ; ENTRY MASK
082A            MOVZWL  #SS$_NONEXPR,R0          ; ASSUME BAD PIX
082A            CMPW    12(AP),@#SCH$GL_MAXPIX   ; CHECK PIX FOR LEGAL PROCESS
082A            BGTR    10$                      ; BR IF NOT
082A            MOVZWL  @16(AP),R1               ; GET SIZE OF CODE SEGMENT
082A            MOVAB   IRP$C_LENGTH(R1),R1      ; ADD SIZE OF PACKET DATA
082A            JSB     @#EXE$ALLOCBUF           ; ALLOCATE BUFFER TO CONTAIN CODE
082A            BLBC    R0,10$                   ; BRANCH IF NONE
082A            MOVL    R2,R5                    ; SAVE ADDRESS OF PACKET
082A            MOVL    PCB$L_PID(R4),FP_ORIGPID(R5)     ; SET PID FOR RETURN
082A            MOVB    #^X80,ACB$B_RMOD(R5)     ; SET FOR SPECIAL KERNEL AST
082A            MOVAB   ACB$L_KAST+8(R5),ACB$L_KAST(R5) ; SET ADDRESS FOR AST
082A            MOVL    4(AP),FP_ADDR(R5)        ; SET ADDRESS FOR FETCH
082A            MOVL    8(AP),FP_RETLOC(R5)      ; AND ADDRESS OF RETURN LOCATION
082A            MOVL    16(AP),R0                ; GET ADDRESS OF CODE SEGMENT
082A            MOVL    12(AP),ACB$L_PID(R5)     ; SET TARGET PID
082A            PUSHR   #^M<R0,R1,R2,R3,R4,R5>   ; SAVE REGS FOR MOVC
082A            MOVC3   (R0)+,(R0),ACB$L_KAST+8(R5); COPY CODE SEGMENT TO BUFFER
082A            POPR    #^M<R0,R1,R2,R3,R4,R5>   ; RESTORE REGISTERS
082A            MOVZBL  #PRI$_TICOM,R2           ; SET PRIORITY INCREMENT CLASS
082A            JSB     @#SCH$QAST               ; QUEUE AST FOR TARGET
082A    10$:    RET                              ; RETURN TO ORIGINAL MODE
082A
082A            .SBTTL  FPBYTE - FETCH BYTE FROM PROCESS
082A    FPBYTE: .WORD   90$-.-2                  ; SIZE OF CODE SEGMENT
082A            IFNORD  #1,@FP_ADDR(R5),10$      ; BRANCH IF NOT READABLE
082A            MOVB    @FP_ADDR(R5),FP_VALUE(R5) ; GET VALUE
082A    10$:    MOVL    FP_ORIGPID(R5),ACB$L_PID(R5) ; SET PID FOR RETURN AST
082A            MOVB    #^X80,ACB$B_RMOD(R5)     ; SET FOR KAST AGAIN
082A            MOVAB   B^20$,ACB$L_KAST(R5)     ; SET NEW AST ADDRESS
082A            MOVZBL  #PRI$_TICOM,R2           ; SET PRIORITY INCREMENT CLASS
082A            JMP     @#SCH$QAST               ; QUEUE RETURN AST
082A    20$:    IFNOWRT #1,@FP_RETLOC(R5),30$    ; IF NOT WRITABLE THEN SKIP IT
082A            MOVB    FP_VALUE(R5),@FP_RETLOC(R5)      ; RETURN VALUE
082A    30$:    MOVL    ACB$L_PID(R5),R1         ; GET PID FOR WAKE
082A            SETIPL  #IPL$_SYNCH              ; RAISE TO SYNCH
```

```
_$2

Vir
Sta
Ima
Ima
Ima
Num
Num
Num
Num
Num
Num
Ima
Map
Est

Per
---

Tot

Usi

Tot

Num

23

A t

LIN
CTI
BAS
```

```
082A             JSB     @#SCH$WAKE                  ; WAKE PROCESS
082A             SETIPL  #IPL$_ASTDEL                ; LOWER IPL
082A             MOVL    R5,R0                       ; SET ADDRESS FOR RELEASE
082A             JMP     @#EXE$DEANONPAGED           ; FREE BLOCK AND EXIT
082A    90$:                                         ; END OF CODE SEGMENT
082A
082A             .PAGE
082A             .SBTTL  DPBYTE - DEPOSIT BYTE TO PROCESS
082A    DPBYTE: .WORD   90$-.-2                     ; SIZE OF CODE SEGMENT
082A    20$:     IFNOWRT #1,@FP_RETLOC(R5),30$       ; IF NOT WRITABLE THEN SKIP IT
082A             MOVB    FP_VALUE(R5),@FP_RETLOC(R5)       ; RETURN VALUE
082A    30$:     MOVL    R5,R0                       ; SET ADDRESS FOR RELEASE
082A             JMP     @#EXE$DEANONPAGED           ; FREE BLOCK AND EXIT
082A    90$:                                         ; END OF CODE SEGMENT
082A
082A             .PAGE
082A             .SBTTL  FPWORD - FETCH WORD FROM PROCESS
082A    FPWORD: .WORD   90$-.-2                     ; SIZE OF CODE SEGMENT
082A             IFNORD  #2,@FP_ADDR(R5),10$         ; BRANCH IF NOT READABLE
082A             MOVW    @FP_ADDR(R5),FP_VALUE(R5)   ; GET VALUE
082A    10$:     MOVL    FP_ORIGPID(R5),ACB$L_PID(R5) ; SET PID FOR RETURN AST
082A             MOVB    #^X80,ACB$B_RMOD(R5)        ; SET FOR KAST AGAIN
082A             MOVAB   B^20$,ACB$L_KAST(R5)        ; SET FOR NEW AST ADDRESS
082A             MOVZBL  #PRI$_TICOM,R2              ; SET PRIORITY INCREMENT CLASS
082A             JMP     @#SCH$QAST                  ; QUEUE RETURN AST
082A    20$:     IFNOWRT #2,@FP_RETLOC(R5),30$       ; IF NOT WRITABLE THEN SKIP IT
082A             MOVW    FP_VALUE(R5),@FP_RETLOC(R5)       ; RETURN VALUE
082A    30$:     MOVL    ACB$L_PID(R5),R1            ; GET PID FOR WAKE
082A             SETIPL  #IPL$_SYNCH                 ; RAISE TO SYNCH
082A             JSB     @#SCH$WAKE                  ; WAKE PROCESS
082A             SETIPL  #IPL$_ASTDEL                ; LOWER IPL
082A             MOVL    R5,R0                       ; SET ADDRESS FOR RELEASE
082A             JMP     @#EXE$DEANONPAGED           ; FREE BLOCK AND EXIT
082A    90$:                                         ; END OF CODE SEGMENT
082A
082A             .PAGE
082A             .SBTTL  DPWORD - DEPOSIT WORD TO PROCESS
082A    DPWORD: .WORD   90$-.-2                     ; SIZE OF CODE SEGMENT
082A    20$:     IFNOWRT #2,@FP_RETLOC(R5),30$       ; IF NOT WRITABLE THEN SKIP IT
082A             MOVW    FP_VALUE(R5),@FP_RETLOC(R5)       ; RETURN VALUE
082A    30$:     MOVL    R5,R0                       ; SET ADDRESS FOR RELEASE
082A             JMP     @#EXE$DEANONPAGED           ; FREE BLOCK AND EXIT
082A    90$:                                         ; END OF CODE SEGMENT
082A
082A             .PAGE
082A             .SBTTL  FPLONG - FETCH LONG FROM PROCESS
082A    FPLONG: .WORD   90$-.-2                     ; SIZE OF CODE SEGMENT
082A             IFNORD  #4,@FP_ADDR(R5),10$         ; BRANCH IF NOT READABLE
082A             MOVL    @FP_ADDR(R5),FP_VALUE(R5)   ; GET VALUE
082A    10$:     MOVL    FP_ORIGPID(R5),ACB$L_PID(R5) ; SET PID FOR RETURN AST
082A             MOVB    #^X80,ACB$B_RMOD(R5)        ; SET FOR KAST AGAIN
082A             MOVAB   B^20$,ACB$L_KAST(R5)        ; SET NEW KAST ADDRESS
082A             CLRL    R2                          ; NULL PRIO INCR
082A             JMP     @#SCH$QAST                  ; QUEUE RETURN AST
082A    20$:     IFNOWRT #4,@FP_RETLOC(R5),30$       ; IF NOT WRITABLE THEN SKIP IT
082A             MOVL    FP_VALUE(R5),@FP_RETLOC(R5)       ; RETURN VALUE
082A    30$:     MOVL    ACB$L_PID(R5),R1            ; GET PID FOR WAKE
```

```
082A            SETIPL  #IPL$_SYNCH                     ; RAISE TO SYNCH
082A            JSB     @#SCH$WAKE                      ; WAKE PROCESS
082A            SETIPL  #IPL$_ASTDEL                    ; LOWER IPL
082A            MOVL    R5,R0                           ; SET ADDRESS FOR RELEASE
082A            JMP     @#EXE$DEANONPAGED               ; FREE BLOCK AND EXIT
082A  90$:                                              ; END OF CODE SEGMENT
082A
082A            .PAGE
082A            .SBTTL  DPLONG - DEPOSIT LONGWORD TO PROCESS
082A  DPLONG:   .WORD   90$-.-2                         ; SIZE OF CODE SEGMENT
082A  20$:      IFNOWRT #4,@FP_RETLOC(R5),30$           ; IF NOT WRITABLE THEN SKIP IT
082A            MOVL    FP_VALUE(R5),@FP_RETLOC(R5)     ; RETURN VALUE
082A  30$:      MOVL    R5,R0                           ; SET ADDRESS FOR RELEASE
082A            JMP     @#EXE$DEANONPAGED               ; FREE BLOCK AND EXIT
082A  90$:                                              ; END OF CODE SEGMENT
082A  DELEND:                                           ;
082A            .ENDC                                   ;
082A  ;
082A  ;         NORMAL END STATEMENT WITHOUT START ADDRESS
082A  ;         USED TO ASSEMBLE XDELTA FOR EXEC DEBUGGING.
082A  ;
082A            .END
```

```
                                                            C 10
XDELTA                            - EXECUTIVE DEBUGGER                   16-SEP-1984 02:02:16  VAX/VMS Macro V04-00               Page  54
Symbol table                                                              3-SEP-1980 13:47:15  DISK$VMSMASTER:[MP.SRC]END.MAR;1   (1)

ADD                     0000024A R       02    HIGH                     00000210 R       02
ASTEN                   000000B8 R       02    INBUF                    00000004 R       02
B                       00000058 R       02    INFLD                    00000206 R       02
BLANK                   00000431 R       02    INI$BRK                  ********     X   02
BMSG                    000006EA R       02    LBRACKET                 000004B2 R       02
BRKADR                = 000000B8 R       02    LF                     = 0000000A
BRKCOM                = 00000100 R       02    LINEFEED                 000002F2 R       02
BRKDSP                = 000000E0 R       02    LOCOUT                   000002F9 R       02
BRKOP                 = 000000DB R       02    LOCP                     00000465 R       02
BRKPOINT                000004D7 R       02    LOCPROMPT                000002F7 R       02
BSLSH                 = 0000005C               MCHK                     000005F0 R       02
CLR_730                 00000607 R       02    MCHKSAV                  00000164 R       02
CLR_750                 00000607 R       02    MFYFLG                   0000004C R       02
CLR_780                 00000602 R       02    MFYFLGS                  00000589 R       02
CLR_END                 0000060A R       02    MMG$PAGEFAULT            ********     X   02
COLON                   0000057F R       02    MODES                    000004AE R       02
COMMA                   0000029B R       02    MUL                      00000242 R       02
CONTEXT                 00000000 R       02    NBRK                   = 00000008
CONTEXTSZ             = 000000BC               NEGATE                   0000043A R       02
CR                    = 0000000D               NEXTDOT                  000002DF R       02
CRLF                    000003B4 R       02    NEXTLOC                  000002F5 R       02
CURDOT                  00000058 R       02    NEXTP                    000001B5 R       02
CURTYPE                 00000056 R       02    NMODES                 = 00000004
DCOM                    0000019A R       02    NPRIM                  = 0000002A
DEPOSIT                 000007F1 R       02    NSEC                   = 00000007
DIV                     00000246 R       02    NTERM                  = 00000008
DOT                     0000058F R       02    OPEN                     00000256 R       02
DQUOTE                  0000024E R       02    OPER                     00000057 R       02
DTYPE                   00000055 R       02    OPERATOR                 00000431 R       02
ENDEXPR                 0000021F R       02    OPERBAS                = 00000012
ENDFIELD                0000029E R       02    OUTB                   = 00000006
EQL1                    0000046F R       02    OUTBB                    000002EF R       02
EQUALS                  00000468 R       02    OUTBSLSH                 00000388 R       02
ERR2                    000004AB R       02    OUTBUF                   00000060 R       02
ERR3                    0000060A R       02    OUTCHAR                  00000391 R       02
ERR4                    00000281 R       02    OUTCOM                   00000364 R       02
ERROR                   0000019E R       02    OUTCR                  = 00000004
ESCAP                   00000456 R       02    OUTDIGIT                 0000035D R       02
EXE$ACVIOLAT            ********     X   02    OUTER                    00000192 R       02
EXE$BREAK               ********     X   02    OUTLONG                  00000361 R       02
EXE$GB_CPUTYPE          ********     X   02    OUTPUT                   000002FE R       02
EXE$ROPRAND             ********     X   02    OUTPUTA                  00000324 R       02
EXE$TBIT                ********     X   02    OUTR8                    0000038E R       02
EXECUTE                 00000817 R       02    OUTSPACE                 000003AF R       02
F1                      00000038 R       02    OUTZBUF                  0000037A R       02
F2                      0000003C R       02    OUTZSTRING               0000037E R       02
F3                      00000040 R       02    PFN$AB_STATE             ********     X   02
F4                      00000044 R       02    PFN$AB_TYPE              ********     X   02
F5                      00000048 R       02    PFN$AL_BAK               ********     X   02
FCTR                    00000054 R       02    PFN$AL_PTE               ********     X   02
FETCH                   000002B9 R       02    PFN$AW_REFCNT            ********     X   02
GETBPTX                 000007CD R       02    PFN$AW_SWPVBN            ********     X   02
GETCHAR                 000003BE R       02    PFN$AX_BLINK             ********     X   02
GETCMD                  0000074C R       02    PFN$AX_FLINK             ********     X   02
GETSCB                  000006D8 R       02    PID                      00000050 R       02
GLOBL                   0000020A R       02    PR$_IPL                = 00000012
GO                      00000575 R       02    PR$_KSP                = 00000000
```

```
PR$_MAPEN               = 00000038                    TERM                      00000181 R     02
PR$_MCESR               = 00000026                    UNBRK                     00000795 R     02
PR$_RXCS                = 00000020                    VALI                      000005B1 R     02
PR$_RXDB                = 00000021                    VALR                      000005AE R     02
PR$_SBIFS               = 00000030                    VALUE                     000005A6 R     02
PR$_SCBB                = 00000011                    V_ASCII                 = 00000001
PR$_SID_TYP780          = 00000001                    V_ATBRK                 = 00000004
PR$_TXCS                = 00000022                    V_F1                    = 00000008
PR$_TXDB                = 00000023                    V_F2                    = 00000009
PRET                      00000217 R     02           V_F3                    = 0000000A
PREG                      00000825 R     02           V_F4                    = 0000000B
PRIMARY                   00000168 R     02           V_F5                    = 0000000C
PROCED                    0000057E R     02           V_INFIELD               = 00000002
PROCEED                   00000751 R     02           V_NEGATE                = 00000007
PROGCTR                   000005A2 R     02           V_OPEN                  = 00000000
PSL$S_CURMOD            = 00000002                    V_PREG                  = 0000001F
PSL$V_CURMOD            = 00000018                    V_PRMODE                = 0000000F
PSL$V_TBIT              = 00000004                    V_RUB                   = 00000006
QUAN                      0000005C R     02           V_TBIT                  = 00000003
QUANT                     0000059C R     02           V_TBITOK                = 00000005
QUOT                    = 00000027                    XDELACV                   000005F0 R     02
QUOTE                     000007DD R     02           XDELBPT                   000006F4 RG    02
RDBUF                   = 00000002                    XDELDBG                   0000077B R     02
RDCR                    = 00000000                    XDELIBRK                  000000BC RG    02
REGCOM                    000005BC R     02           XDELTBIT                  00000764 RG    02
REGISTER                  000005B4 R     02           XDEL_LOADBASE             00000130 RG    02
RESET                     00000477 R     02           XDS$GL_XESTRING           0000015C RG    02
RESTORE                   00000680 R     02           XDS$GL_XFSTRING           00000160 RG    02
RESTORR                   00000685 R     02           XDS$GT_WORD_PFN           ********   X   02
RETURN                    00000284 R     02           XREG                      000005E2 R     02
RSET                      00000291 R     02           XREGV                     00000124 R     02
RUBOUT                  = 0000007F                    XSET                      000005D0 R     02
SAVAP                     000000A0 R     02
SAVE                      0000060D R     02
SAVOCR                    000000B4 R     02
SAVPC                     000000AC R     02
SAVPSL                    000000B0 R     02
SAVR2                     00000078 R     02
SAVRCR                    000000B6 R     02
SAVREG                    00000070 R     02
SAVRXCS                   000000B8 R     02
SAVSP                     000000A8 R     02
SCANP                     000001B1 R     02
SCB$AL_BASE               ********   X   02
SCH$GL_CURPCB             ********   X   02
SCH$GL_PCBVEC             ********   X   02
SECOND                    00000484 R     02
SEMI                      0000048B R     02
SETBRK                    000007AA R     02
SHFT                      0000023D R     02
SHOBRK                    00000533 R     02
SLASH                     00000253 R     02
SLSH                    = 0000002F
STATUS                    00000034 R     02
STEP                      000004CD R     02
SUPERST                   000001A5 R     02
TAB                       00000446 R     02
```

```
                                             +----------------+
                                             ! 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
Z$DEBUGXDELTA                   0000082A  (  2090.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        38    00:00:00.11    00:00:00.65
Command processing                   124    00:00:01.02    00:00:04.74
Pass 1                               405    00:00:15.16    00:00:44.35
Symbol table sort                      0    00:00:02.05    00:00:03.57
Pass 2                               338    00:00:05.33    00:00:12.81
Symbol table output                   24    00:00:00.23    00:00:00.68
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                 933    00:00:23.92    00:01:06.82

The working set limit was 1950 pages.
87249 bytes (171 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1237 non-local and 91 local symbols.
1842 source lines were read in Pass 1, producing 18 object records in Pass 2.
24 pages of virtual memory were used to define 23 macros.

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

Macro library name                         Macros defined
------------------                         --------------
_$255$DUA28:[MP.OBJ]MP.MLB;1                           10
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       8
TOTALS (all libraries)                                 28

1396 GETS were required to define 28 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:XDELTA/OBJ=OBJ$:XDELTA MSRC$:MPPREFIX/UPDATE=(ENH$:MPPREFIX)+MSRC$:XDELTA/UPDATE=(ENH$:XDELTA)+MSRC$:END/UPDATE=(ENH$
```

0249 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MSCP

MSCP
MAP

ADDUNIT
LIS

MSCP
LIS

MPWAIT
LIS

MPTIMER
LIS

XDELTA
LIS

MSCPDEF
MAR