```
EEEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                AAAAAAAAA     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                AAAAAAAAA     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                AAAAAAAAA     TTTTTTTTTTTTTTT
EEE               MMMMMMM   MMMMMMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMMMMMM   MMMMMMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMMMMMM   MMMMMMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM MMM MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM MMM MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM MMM MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEEEEEEEEEEE      MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEEEEEEEEEEE      MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEEEEEEEEEEE      MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEE               MMM     MMM  UUU         UUU  LLL              AAA         AAA         TTT
EEEEEEEEEEEEEEEE  MMM     MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA         TTT
EEEEEEEEEEEEEEEE  MMM     MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA         TTT
EEEEEEEEEEEEEEEE  MMM     MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA         TTT
```

FILEID**VAXREGDEF

```
VV      VV    AAAAAA    XX      XX  RRRRRRRR    EEEEEEEEEE    GGGGGGGG  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
VV      VV    AAAAAA    XX      XX  RRRRRRRR    EEEEEEEEEE    GGGGGGGG  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
VV      VV  AA      AA  XX      XX  RR      RR  EE          GG          DD      DD  EE          FF
VV      VV  AA      AA  XX      XX  RR      RR  EE          GG          DD      DD  EE          FF
VV      VV  AA      AA    XX  XX    RR      RR  EE          GG          DD      DD  EE          FF
VV      VV  AA      AA    XX  XX    RR      RR  EE          GG          DD      DD  EE          FF
VV      VV  AA      AA      XX      RRRRRRRR    EEEEEEEE    GG          DD      DD  EEEEEEEE    FFFFFFFF
VV      VV  AA      AA      XX      RRRRRRRR    EEEEEEEE    GG          DD      DD  EEEEEEEE    FFFFFFFF
VV      VV  AAAAAAAAAA    XX  XX    RR  RR      EE          GG  GGGGGG  DD      DD  EE          FF
VV      VV  AAAAAAAAAA    XX  XX    RR  RR      EE          GG  GGGGGG  DD      DD  EE          FF
  VV  VV    AA      AA  XX      XX  RR    RR    EE          GG      GG  DD      DD  EE          FF
  VV  VV    AA      AA  XX      XX  RR    RR    EE          GG      GG  DD      DD  EE          FF
    VV      AA      AA  XX      XX  RR      RR  EEEEEEEEEE    GGGGGG    DDDDDDDD    EEEEEEEEEE  FF
    VV      AA      AA  XX      XX  RR      RR  EEEEEEEEEE    GGGGGG    DDDDDDDD    EEEEEEEEEE  FF


  SSSSSSSS  DDDDDDDD    LL
  SSSSSSSS  DDDDDDDD    LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
  SSSSSS    DD      DD  LL
  SSSSSS    DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
```

VAX

MOD

/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*

AGG

END

END

```
{       SRC$:VAXREGDEF.SDL
{
{       Version 'V04-000'
{
{****************************************************************************
{*                                                                          *
{*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
{*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
{*  ALL RIGHTS RESERVED.                                                    *
{*                                                                          *
{*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
{*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
{*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
{*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
{*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
{*  TRANSFERRED.                                                            *
{*                                                                          *
{*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
{*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
{*  CORPORATION.                                                            *
{*                                                                          *
{*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
{*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
{*                                                                          *
{*                                                                          *
{****************************************************************************

{++
{
{ Facility:
{
{       VAX-11 Instruction Emulator
{
{ Abstract:
{
{       This file contains a set of symbolic definitions for the placement
{       of operands into registers for use by the VAX-11 instruction emulator.
{       The use of stack space by various pieces of the emulator front end
{       are also defined.
{
{ Author:
{
{       Lawrence J. Kenah
{
{ Creation Date:
{
{       10 January 1983
{
{ Revision History:
{
{       V01-004 LJK0038         Lawrence J. Kenah       19 Jult 1984
{               Add restart context fields to MOVP to allow original setting
{               of C-bit to be preserved across access violations.
{
{       V01-003 LJK0026         Lawrence J. Kenah       19 March 1984
{               Add fields to CVTLP, CVTPL, and EDITPC that allow these
```

```
;               instructions to be restarted at other than the original entry
;               point. Add an internal FPD bit to CVTPT and CVTTP.
;
;       V01-002 LJK0007         Lawrence J. Kenah       17 October 1983
;               Add delta-PC fields to CVTPT and CVTTP definitions.
;
;       V01-001                 Lawrence J. Kenah       10 January 1983
;               Original version.
;--
```

VAX

END

END

```
MODULE movtc_def;

/* Define register usage for the MOVTC instruction
/*
/*     31               23               15               07              00
/*    +----------------+----------------+----------------+----------------+
/*    |         initial srclen          |             srclen              | : R0
/*    +----------------+----------------+----------------+----------------+
/*    |                              srcaddr                               | : R1
/*    +----------------+----------------+----------------+----------------+
/*    |    delta-PC    |      XXXX      |     flags      |      fill      | : R2
/*    +----------------+----------------+----------------+----------------+
/*    |                              tbladdr                               | : R3
/*    +----------------+----------------+----------------+----------------+
/*    |         initial dstlen          |             dstlen              | : R4
/*    +----------------+----------------+----------------+----------------+
/*    |                              dstaddr                               | : R5
/*    +----------------+----------------+----------------+----------------+

AGGREGATE movtc STRUCTURE PREFIX "movtc_";

    srclen WORD UNSIGNED;                   /* srclen.rw
    inisrclen WORD UNSIGNED;
    srcaddr ADDRESS;                        /* srcaddr.ab
    fill BYTE UNSIGNED;                     /* fill.rb
    flags STRUCTURE BYTE UNSIGNED TAG "b";
        FPD BITFIELD LENGTH 1 MASK;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    tbladdr ADDRESS;                        /* tbladdr.ab
    dstlen WORD UNSIGNED;                   /* dstlen.rw
    inidstlen WORD UNSIGNED;
    dstaddr ADDRESS;                        /* dstaddr.ab

END;

END_MODULE movtc_def;
```

```
MODULE movtuc_def;

/* Define register usage for the MOVTUC instruction
/*
/*     31               23               15               07               00
/*    +----------------+----------------+----------------+----------------+
/*    |         initial srclen          |             srclen              | : R0
/*    +----------------+----------------+----------------+----------------+
/*    |                             srcaddr                                | : R1
/*    +----------------+----------------+----------------+----------------+
/*    |    delta-PC    |      XXXX      |     flags      |      esc       | : R2
/*    +----------------+----------------+----------------+----------------+
/*    |                             tbladdr                                | : R3
/*    +----------------+----------------+----------------+----------------+
/*    |         initial dstlen          |             dstlen              | : R4
/*    +----------------+----------------+----------------+----------------+
/*    |                             dstaddr                                | : R5
/*    +----------------+----------------+----------------+----------------+

AGGREGATE movtuc STRUCTURE PREFIX "movtuc_";

    srclen WORD UNSIGNED;                   /* srclen.rw
    inisrclen WORD UNSIGNED;
    srcaddr ADDRESS;                        /* srcaddr.ab
    esc BYTE UNSIGNED;                      /* esc.rb
    flags STRUCTURE BYTE UNSIGNED TAG "b";
        FPD BITFIELD LENGTH 1 MASK;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    tbladdr ADDRESS;                        /* tbladdr.ab
    dstlen WORD UNSIGNED;                   /* dstlen.rw
    inidstlen WORD UNSIGNED;
    dstaddr ADDRESS;                        /* dstaddr.ab

END;

END_MODULE movtuc_def;
```

```
MODULE cmpc3_def;

/* Define register usage for the CMPC3 instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+----------------+
/*    |    delta-PC     |      XXXX       |                 len               | : R0
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                 src1addr                              | : R1
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                   XXXXX                               | : R2
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                 src2addr                              | : R3
/*    +-----------------+-----------------+-----------------+----------------+

AGGREGATE cmpc3 STRUCTURE PREFIX "cmpc3_";

    len WORD UNSIGNED;                  /* len.rw
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    src1addr ADDRESS;                   /* src1addr.ab
    XXXXX LONGWORD FILL;
    src2addr ADDRESS;                   /* src2addr.ab

END;

END_MODULE cmpc3_def;
```

VAX
MOD
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
AGG
END
END

```
MODULE cmpc5_def;

/* Define register usage for the CMPC5 instruction
/*
/*
/*     31                23                15                07              00
/*     +-----------------+-----------------+-----------------+---------------+
/*     |    delta-PC     |      fill       |              src1len              | : R0
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                 src1addr                              | : R1
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                XXXXX              |              src2len              | : R2
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                 src2addr                              | : R3
/*     +-----------------+-----------------+-----------------+---------------+

AGGREGATE cmpc5 STRUCTURE PREFIX "cmpc5_";

    src1len WORD UNSIGNED;              /* src1len.rw
    fill BYTE UNSIGNED;                 /* fill.rb
    delta_PC BYTE UNSIGNED;
    src1addr ADDRESS;                   /* src1addr.ab
    src2len WORD UNSIGNED;              /* src2len.rw
    XXXXX WORD FILL;
    src2addr ADDRESS;                   /* src2addr.ab

END;

END_MODULE cmpc5_def;
```

```
MODULE scanc_def;

/* Define register usage for the SCANC instruction
/*
/*     31              23              15              07              00
/*     +---------------+---------------+---------------+---------------+
/*     |    delta-PC   |      XXXX     |               len             | : R0
/*     +---------------+---------------+---------------+---------------+
/*     |                             addr                              | : R1
/*     +---------------+---------------+---------------+---------------+
/*     |               XXXXX           |      XXXX     |      mask     | : R2
/*     +---------------+---------------+---------------+---------------+
/*     |                            tbladdr                            | : R3
/*     +---------------+---------------+---------------+---------------+

AGGREGATE scanc STRUCTURE PREFIX "scanc_";

    len WORD UNSIGNED;                  /* len.rw
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    addr ADDRESS;                       /* addr.ab
    mask BYTE UNSIGNED;                 /* mask.rb
    XXXX BYTE FILL;
    XXXXX WORD FILL;
    tbladdr ADDRESS;                    /* tbladdr.ab

END;

END_MODULE scanc_def;
```

```
MODULE spanc_def;

/* Define register usage for the SPANC instruction
/*
/*      31                23                15                07                00
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |    delta-PC     |      XXXX       |                 len               | : R0
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                   addr                                | : R1
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |               XXXXX               |      XXXX       |      mask       | : R2
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 tbladdr                               | : R3
/*     +-----------------+-----------------+-----------------+-----------------+

AGGREGATE spanc STRUCTURE PREFIX "spanc_";

    len WORD UNSIGNED;                  /* len.rw
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    addr ADDRESS;                       /* addr.ab
    mask BYTE UNSIGNED;                 /* mask.rb
    XXXX BYTE FILL;
    XXXXX WORD FILL;
    tbladdr ADDRESS;                    /* tbladdr.ab

END;

END_MODULE spanc_def;
```

```
MODULE locc_def;

/* Define register usage for the LOCC instruction
/*
/*      31                  23                  15                  07                  00
/*     +------------------+------------------+------------------+------------------+
/*     |     delta-PC     |       char       |                  len                 | : R0
/*     +------------------+------------------+------------------+------------------+
/*     |                                    addr                                    | : R1
/*     +------------------+------------------+------------------+------------------+

AGGREGATE locc STRUCTURE PREFIX "locc_";

    len WORD UNSIGNED;                  /* len.rw
    char BYTE UNSIGNED;                 /* char.rb
    delta_PC BYTE UNSIGNED;
    addr ADDRESS;                       /* addr.ab

END;

END_MODULE locc_def;
```

```
MODULE skpc_def;

/* Define register usage for the SKPC instruction
/*
/*     31                23                15                07                00
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |    delta-PC     |      char       |                 len               | : R0
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                   addr                                | : R1
/*     +-----------------+-----------------+-----------------+-----------------+

AGGREGATE skpc STRUCTURE PREFIX "skpc_";

    len WORD UNSIGNED;                  /* len.rw
    char BYTE UNSIGNED;                 /* char.rb
    delta_PC BYTE UNSIGNED;
    addr ADDRESS;                       /* addr.ab

END;

END_MODULE skpc_def;
```

```
MODULE matchc_def;

/* Define register usage for the MATCHC instruction
/*
/*     31                23                15                07              00
/*     +-----------------+-----------------+-----------------+---------------+
/*     |    delta-PC     |      XXXX       |                 objlen          | : R0
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                   objaddr                           | : R1
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                XXXXX              |                 srclen          | : R2
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                   srcaddr                           | : R3
/*     +-----------------+-----------------+-----------------+---------------+

AGGREGATE matchc STRUCTURE PREFIX "matchc_";

    objlen WORD UNSIGNED;               /* objlen.rw
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    objaddr ADDRESS;                    /* objaddr.ab
    srclen WORD UNSIGNED;               /* srclen.rw
    XXXXX WORD FILL;
    srcaddr ADDRESS;                    /* srcaddr.ab

END;

END_MODULE matchc_def;
```

```
MODULE crc_def;

/* Define register usage for the CRC instruction
/*
/*     31               23               15               07              00
/*    +----------------+----------------+----------------+----------------+
/*    |                              inicrc                               | : R0
/*    +----------------+----------------+----------------+----------------+
/*    |                               tbl                                 | : R1
/*    +----------------+----------------+----------------+----------------+
/*    |    delta-PC    |      XXXX      |             strlen               | : R2
/*    +----------------+----------------+----------------+----------------+
/*    |                              stream                               | : R3
/*    +----------------+----------------+----------------+----------------+

AGGREGATE crc STRUCTURE PREFIX "crc_";

    inicrc LONGWORD UNSIGNED;           /* inicrc.rl
    tbl ADDRESS;                        /* tbl.ab
    strlen WORD UNSIGNED;               /* strlen.rw
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    stream ADDRESS;                     /* stream.ab

END;

END_MODULE crc_def;
```

```
MODULE stack_def;

/* Define exception parameters for VAX$_OPCDEC exception, the exception that
/* occurs when an unimplemented instruction executes on a microVAX processor.
/*
/*     31           23           15           07        00
/*    +------------+------------+------------+------------+
/*    |              opcode of reserved instruction        | : 00(SP)
/*    +------------+------------+------------+------------+
/*    |              PC of reserved instruction (old PC)   | : 04(SP)
/*    +------------+------------+------------+------------+
/*    |              first operand specifier               | : 08(SP)
/*    +------------+------------+------------+------------+
/*    |              second operand specifier              | : 12(SP)
/*    +------------+------------+------------+------------+
/*    |              third operand specifier               | : 16(SP)
/*    +------------+------------+------------+------------+
/*    |              fourth operand specifier              | : 20(SP)
/*    +------------+------------+------------+------------+
/*    |              fifth operand specifier               | : 24(SP)
/*    +------------+------------+------------+------------+
/*    |              sixth operand specifier               | : 28(SP)
/*    +------------+------------+------------+------------+
/*    |              seventh operand specifier             | : 32(SP)
/*    +------------+------------+------------+------------+
/*    |              eighth operand specifier              | : 36(SP)
/*    +------------+------------+------------+------------+
/*    |              PC of next instruction (new PC)       | : 40(SP)
/*    +------------+------------+------------+------------+
/*    |              PSL at time of exception              | : 44(SP)
/*    +------------+------------+------------+------------+

AGGREGATE stack STRUCTURE PREFIX '"' TAG '"';

    opcode LONGWORD TAG '"';              /* Opcode of reserved instruction
    old_PC LONGWORD TAG '"';              /* PC of reserved instruction (old PC)
    operand_1 LONGWORD TAG '"';           /* First operand specifier
    operand_2 LONGWORD TAG '"';           /* Second operand specifier
    operand_3 LONGWORD TAG '"';           /* Third operand specifier
    operand_4 LONGWORD TAG '"';           /* Fourth operand specifier
    operand_5 LONGWORD TAG '"';           /* Fifth operand specifier
    operand_6 LONGWORD TAG '"';           /* Sixth operand specifier
    operand_7 LONGWORD TAG '"';           /* Seventh operand specifier
    operand_8 LONGWORD TAG '"';           /* Eight operand specifier
    new_PC LONGWORD TAG '"';              /* PC of instruction following
                                          /*  reserved instruction (new PC)
    exception_PSL LONGWORD TAG '"';       /* PSL at time of exception

END;

END_MODULE stack_def;
```

```
MODULE pack_def;

/* Define stack usage when packing registers to back up instruction
/*
/*     31           23           15           07         00
/*     +------------+------------+------------+-------------+
/*     |                        saved R0                    | : 00(SP)
/*     +------------+------------+------------+-------------+
/*     |                        saved R1                    | : 04(SP)
/*     +------------+------------+------------+-------------+
/*     |                        saved R2                    | : 08(SP)
/*     +------------+------------+------------+-------------+
/*     |                        saved R3                    | : 12(SP)
/*     +------------+------------+------------+-------------+
/*
/* The next longword after the saved R3 depends on the context.
/*
/* In the case of software generated exceptions, a signal array sits
/* immediately underneath the saved register array.
/*
/*     +------------+------------+------------+-------------+
/*     |           first longword of signal array           | : 16(SP)
/*     +------------+------------+------------+-------------+
/*
/* In the case of an access
/* violation, there is a return PC followed by an argument list.
/*
/*     +------------+------------+------------+-------------+
/*     |                        return PC                   | : 16(SP)
/*     +------------+------------+------------+-------------+
/*     |              argument count (always 2)             | : 20(SP)
/*     +------------+------------+------------+-------------+
/*     |               pointer to signal array              | : 24(SP)
/*     +------------+------------+------------+-------------+
/*     |              pointer to mechanism array            | : 28(SP)
/*     +------------+------------+------------+-------------+
/*
/* After the stack has been modified, a new value of SP is inserted here.
/*
/*     +------------+------------+------------+-------------+
/*     |                        saved SP                    | : 16(SP)
/*     +------------+------------+------------+-------------+

AGGREGATE pack STRUCTURE PREFIX "pack_" TAG "";

    saved_R0 LONGWORD;
    saved_R1 LONGWORD;
    saved_R2 LONGWORD;
    saved_R3 LONGWORD;

    pack_overlay UNION PREFIX pack_ FILL;

        return_PC LONGWORD;
        signal_array LONGWORD;
        saved_SP LONGWORD;
```

```
        END pack_overlay;

    argument_count LONGWORD;
    signal_array_pointer LONGWORD;
    mechanism_array_pointer LONGWORD;

    R1_bits STRUCTURE LONGWORD FILL;
        delta_PC BITFIELD LENGTH 8;
        FPD BITFIELD LENGTH 1 MASK;      /* Set FPD bit in PSL
        accvio BITFIELD LENGTH 1 MASK;   /* Stack contains extra data
        END;

END;

END_MODULE pack_def;
```

```
MODULE addp4_def;

/* Define register usage for the ADDP4 instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+---------------+
/*    |    delta-PC     |      XXXX       |                 addlen          | : R0
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                   addaddr                           | : R1
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                XXXXX              |                 sumlen          | : R2
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                   sumaddr                           | : R3
/*    +-----------------+-----------------+-----------------+---------------+

AGGREGATE addp4 STRUCTURE PREFIX "addp4_";

    addlen STRUCTURE WORD UNSIGNED;     /* addlen.rw
        addlen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    addaddr ADDRESS;                    /* addaddr.ab
    sumlen STRUCTURE WORD UNSIGNED;     /* sumlen.rw
        sumlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    sumaddr ADDRESS;                    /* sumaddr.ab

END;

END_MODULE addp4_def;
```

VAX
MOD
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
/*
AGG
END
END

```
MODULE addp6_def;

/* Define register usage for the ADDP6 instruction
/*
/*      31              23              15              07              00
/*     +---------------+---------------+---------------+---------------+
/*     |    delta-PC   |      XXXX     |             add1len           | : R0
/*     +---------------+---------------+---------------+---------------+
/*     |                           add1addr                            | : R1
/*     +---------------+---------------+---------------+---------------+
/*     |               XXXXX           |             add2len           | : R2
/*     +---------------+---------------+---------------+---------------+
/*     |                           add2addr                            | : R3
/*     +---------------+---------------+---------------+---------------+
/*     |               XXXXX           |             sumlen            | : R4
/*     +---------------+---------------+---------------+---------------+
/*     |                           sumaddr                             | : R5
/*     +---------------+---------------+---------------+---------------+

AGGREGATE addp6 STRUCTURE PREFIX "addp6_";

    add1len STRUCTURE WORD UNSIGNED;    /* add1len.rw
        add1len BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    add1addr ADDRESS;                   /* add1addr.ab
    add2len STRUCTURE WORD UNSIGNED;    /* add2len.rw
        add2len BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    add2addr ADDRESS;                   /* add2addr.ab
    sumlen STRUCTURE WORD UNSIGNED;     /* sumlen.rw
        sumlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    sumaddr ADDRESS;                    /* sumaddr.ab

END;

END_MODULE addp6_def;
```

```
MODULE ashp_def;

/* Define register usage for the ASHP instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |    delta-PC     |       cnt       |                 srclen              | : R0
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |                                  srcaddr                                | : R1
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |       XXXX      |      round      |                 dstlen              | : R2
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |                                  dstaddr                                | : R3
/*    +-----------------+-----------------+-----------------+-----------------+

AGGREGATE ashp STRUCTURE PREFIX "ashp_";

    srclen STRUCTURE WORD UNSIGNED;     /* srclen.rw
        srclen BITFIELD LENGTH 5;
        END;
    cnt BYTE UNSIGNED;                  /* cnt.rb
    delta_PC BYTE UNSIGNED;
    srcaddr ADDRESS;                    /* srcaddr.ab
    dstlen STRUCTURE WORD UNSIGNED;     /* dstlen.rw
        dstlen BITFIELD LENGTH 5;
        END;
    round STRUCTURE BYTE UNSIGNED TAG "b";      /* round.rb
        round BITFIELD LENGTH 4;
        END;
    XXXX BYTE FILL;
    dstaddr ADDRESS;                    /* dstaddr.ab

END;

END_MODULE ashp_def;
```

```
MODULE cmpp3_def;

/* Define register usage for the CMPP3 instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+---------------+
/*    |    delta-PC     |      XXXX       |                 len             | : R0
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                              src1addr                               | : R1
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                               XXXXX                                 | : R2
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                              src2addr                               | : R3
/*    +-----------------+-----------------+-----------------+---------------+

AGGREGATE cmpp3 STRUCTURE PREFIX "cmpp3_";

    len STRUCTURE WORD UNSIGNED;        /* len.rw
        len BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    src1addr ADDRESS;                   /* src1addr.ab
    XXXXX LONGWORD FILL;
    src2addr ADDRESS;                   /* src2addr.ab

END;

END_MODULE cmpp3_def;
```

```
MODULE cmpp4_def;

/* Define register usage for the CMPP4 instruction
/*
/*     31                23                15                07              00
/*     +-----------------+-----------------+-----------------+---------------+
/*     |    delta-PC     |      XXXX       |                 src1len         | : R0
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                 src1addr                            | : R1
/*     +-----------------+-----------------+-----------------+---------------+
/*     |               XXXXX               |                 src2len         | : R2
/*     +-----------------+-----------------+-----------------+---------------+
/*     |                                 src2addr                            | : R3
/*     +-----------------+-----------------+-----------------+---------------+

AGGREGATE cmpp4 STRUCTURE PREFIX "cmpp4_";

    src1len STRUCTURE WORD UNSIGNED;    /* src1len.rw
        src1len BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    src1addr ADDRESS;                   /* src1addr.ab
    src2len STRUCTURE WORD UNSIGNED;    /* src2len.rw
        src2len BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    src2addr ADDRESS;                   /* src2addr.ab

END;

END_MODULE cmpp4_def;
```

```
MODULE cvtlp_def;

/* Define register usage for the CVTLP instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                   src                                 | : R0
/*    +-----------------+-----------------+-----------------+---------------+
/*    |      state      |    saved_PSW    |    saved_R5     |    saved_R4   | : R1
/*    +-----------------+-----------------+-----------------+---------------+
/*    |    delta-PC     |       XXXX      |                 dstlen            | : R2
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                 dstaddr                               | : R3
/*    +-----------------+-----------------+-----------------+---------------+

AGGREGATE cvtlp STRUCTURE PREFIX "cvtlp_";

    src LONGWORD UNSIGNED;                  /* src.rl
    saved_R4 BYTE UNSIGNED;
    saved_R5 BYTE UNSIGNED;
    saved_PSW BYTE UNSIGNED;
    state STRUCTURE BYTE UNSIGNED;
        state BITFIELD LENGTH 3 MASK;
        FPD BITFIELD LENGTH 1 MASK;
        END;
    dstlen STRUCTURE WORD UNSIGNED;         /* dstlen.rw
        dstlen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    dstaddr ADDRESS;                        /* dstaddr.ab
END;
END_MODULE cvtlp_def;
```

```
MODULE cvtpl_def;

/* Define register usage for the CVTPL instruction
/*
/*      31                23                15                07              00
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |    delta-PC     |      state      |                 srclen              | : R0
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 srcaddr                                 | : R1
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 result                                  | : R2
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                   dst                                   | : R3
/*     +-----------------+-----------------+-----------------+-----------------+

AGGREGATE cvtpl STRUCTURE PREFIX "cvtpl_";

    srclen STRUCTURE WORD UNSIGNED;      /* srclen.rw
        srclen STRUCTURE BYTE UNSIGNED;
            srclen BITFIELD LENGTH 5;
            END;
        delta_srcaddr BYTE UNSIGNED;
        END;
    state STRUCTURE BYTE UNSIGNED;
        saved_PSW BITFIELD LENGTH 4 MASK;
        state BITFIELD LENGTH 3 MASK;
        FPD BITFIELD LENGTH 1 MASK;
        END;
    delta_PC BYTE UNSIGNED;
    srcaddr ADDRESS;                     /* srcaddr.ab
    result LONGWORD UNSIGNED;
    dst ADDRESS;                         /* dst.wl

END;

END_MODULE cvtpl_def;
```

```
MODULE cvtps_def;

/* Define register usage for the CVTPS instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+---------------+
/*    |    delta-PC     |      XXXX       |                 srclen          | : R0
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                 srcaddr                             | : R1
/*    +-----------------+-----------------+-----------------+---------------+
/*    |               XXXXX               |                 dstlen          | : R2
/*    +-----------------+-----------------+-----------------+---------------+
/*    |                                 dstaddr                             | : R3
/*    +-----------------+-----------------+-----------------+---------------+

AGGREGATE cvtps STRUCTURE PREFIX "cvtps_";

    srclen STRUCTURE WORD UNSIGNED;     /* srclen.rw
        srclen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    srcaddr ADDRESS;                    /* srcaddr.ab
    dstlen STRUCTURE WORD UNSIGNED;     /* dstlen.rw
        dstlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    dstaddr ADDRESS;                    /* dstaddr.ab

END;

END_MODULE cvtps_def;
```

```
MODULE cvtpt_def;

/* Define register usage for the CVTPT instruction
/*
/*      31                23                15                07              00
/*     +-----------------+-----------------+-----------------+----------------+
/*     |              dstlen               |              srclen               | : R0
/*     +-----------------+-----------------+-----------------+----------------+
/*     |                                 srcaddr                               | : R1
/*     +-----------------+-----------------+-----------------+----------------+
/*     |                                 tbladdr                               | : R2
/*     +-----------------+-----------------+-----------------+----------------+
/*     |                                 dstaddr                               | : R3
/*     +-----------------+-----------------+-----------------+----------------+
/*
/*
/*     There is no spare byte to store the delta-PC quantity with three
/*     addresses to be stored in four registers. The delta-PC quantity is
/*     stored in the upper byte of the dstlen cell. Once the destination length
/*     has been checked for a legal range (a decimal string length larger than
/*     31 causes a reserved operand abort), that length can be easily stored in
/*     five bytes, leaving the upper byte free.
/*
/*                          31              24      20      16
/*                         +-----------------+------+-------+
/*                         |              dstlen             |
/*                         +-----------------+------+-------+
/*                        /                 /        \       \
/*                       /                 /          \       \
/*                      +-----------------+            +-------+
/*                      |    delta-PC     |            |  dstlen |
/*                      +-----------------+            +-------+
/*                       31              24            20      16

AGGREGATE cvtpt STRUCTURE PREFIX "cvtpt_";

    srclen STRUCTURE WORD UNSIGNED;     /* srclen.rw
        srclen BITFIELD LENGTH 5;
        XXXX BITFIELD LENGTH 10 FILL;
        FPD BITFIELD LENGTH 1;
        END;

    dstlen STRUCTURE WORD UNSIGNED;     /* dstlen.rw
        dstlen STRUCTURE BYTE UNSIGNED;
            dstlen BITFIELD LENGTH 5;
            END;
        delta_PC BYTE UNSIGNED;
        END;

    srcaddr ADDRESS;                    /* srcaddr.ab
    tbladdr ADDRESS;                    /* tbladdr.ab
    dstaddr ADDRESS;                    /* dstaddr.ab

END;
```

```
END_MODULE cvtpt_def;
```

```
MODULE cvtsp_def;

/* Define register usage for the CVTSP instruction
/*
/*    31          23          15          07          00
/*
/*    +-----------+-----------+-----------+-----------+
/*    | delta-PC  |   XXXX    |        srclen         | : R0
/*    +-----------+-----------+-----------+-----------+
/*    |                    srcaddr                    | : R1
/*    +-----------+-----------+-----------+-----------+
/*    |         XXXXX         |        dstlen         | : R2
/*    +-----------+-----------+-----------+-----------+
/*    |                    dstaddr                    | : R3
/*    +-----------+-----------+-----------+-----------+

AGGREGATE cvtsp STRUCTURE PREFIX "cvtsp_";

    srclen STRUCTURE WORD UNSIGNED;     /* srclen.rw
        srclen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    srcaddr ADDRESS;                    /* srcaddr.ab
    dstlen STRUCTURE WORD UNSIGNED;     /* dstlen.rw
        dstlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    dstaddr ADDRESS;                    /* dstaddr.ab

END;

END_MODULE cvtsp_def;
```

```
MODULE cvttp_def;

/* Define register usage for the CVTTP instruction
/*
/*      31                23                15                07                00
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |              dstlen               |              srclen               | : R0
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 srcaddr                                 | : R1
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 tbladdr                                 | : R2
/*     +-----------------+-----------------+-----------------+-----------------+
/*     |                                 dstaddr                                 | : R3
/*     +-----------------+-----------------+-----------------+-----------------+
/*
/*
/*     There is no spare byte to store the delta-PC quantity with three
/*     addresses to be stored in four registers. The delta-PC quantity is
/*     stored in the upper byte of the dstlen cell. Once the destination length
/*     has been checked for a legal range (a decimal string length larger than
/*     31 causes a reserved operand abort), that length can be easily stored in
/*     five bytes, leaving the upper byte free.
/*
/*                          31              24      20      16
/*                         +-----------------+-----+-----+
/*                         |                 dstlen        |
/*                         +-----------------+-----+-----+
/*                        /                 /       \       \
/*                       /                 /         \       \
/*                      +-----------------+           +-----+-----+
/*                      |    delta-PC     |           |  dstlen   |
/*                      +-----------------+           +-----+-----+
/*                       31              24            20      16

AGGREGATE cvttp STRUCTURE PREFIX "cvttp_";

    srclen STRUCTURE WORD UNSIGNED;     /* srclen.rw
        srclen BITFIELD LENGTH 5;
        XXXX BITFIELD LENGTH 10 FILL;
        FPD BITFIELD LENGTH 1;
        END;

    dstlen STRUCTURE WORD UNSIGNED;     /* dstlen.rw
        dstlen STRUCTURE BYTE UNSIGNED;
            dstlen BITFIELD LENGTH 5;
            END;
        delta_PC BYTE UNSIGNED;
        END;

    srcaddr ADDRESS;                    /* srcaddr.ab
    tbladdr ADDRESS;                    /* tbladdr.ab
    dstaddr ADDRESS;                    /* dstaddr.ab

END;
```

```
END_MODULE cvttp_def;
```

```
MODULE divp_def;

/* Define register usage for the DIVP instruction
/*
/*     31                23                15                07               00
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |    delta-PC     |      XXXX       |              divrlen              | : R0
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |                              divraddr                                 | : R1
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |               XXXXX               |              divdlen              | : R2
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |                              divdaddr                                 | : R3
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |               XXXXX               |              quolen               | : R4
/*    +-----------------+-----------------+-----------------+-----------------+
/*    |                              quoaddr                                  | : R5
/*    +-----------------+-----------------+-----------------+-----------------+

AGGREGATE divp STRUCTURE PREFIX "divp_";

    divrlen STRUCTURE WORD UNSIGNED;    /* divrlen.rw
        divrlen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    divraddr ADDRESS;                   /* divraddr.ab
    divdlen STRUCTURE WORD UNSIGNED;    /* divdlen.rw
        divdlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    divdaddr ADDRESS;                   /* divdaddr.ab
    quolen STRUCTURE WORD UNSIGNED;     /* quolen.rw
        quolen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    quoaddr ADDRESS;                    /* quoaddr.ab

END;

END_MODULE divp_def;
```

```
MODULE movp_def;

/* Define register usage for the MOVP instruction
/*
/*      31                23                15                07              00
/*     +----------------+----------------+----------------+----------------+
/*     |    delta-PC    |     state      |                len               | : R0
/*     +----------------+----------------+----------------+----------------+
/*     |                                srcaddr                             | : R1
/*     +----------------+----------------+----------------+----------------+
/*     |                                 XXXXX                              | : R2
/*     +----------------+----------------+----------------+----------------+
/*     |                                dstaddr                             | : R3
/*     +----------------+----------------+----------------+----------------+

AGGREGATE movp STRUCTURE PREFIX "movp_";

    len STRUCTURE WORD UNSIGNED;        /* len.rw
        len BITFIELD LENGTH 5;
        END;
    state STRUCTURE BYTE UNSIGNED;
        saved_PSW BITFIELD LENGTH 4 MASK;
        FPD BITFIELD LENGTH 1 MASK;
        END;
    delta_PC BYTE UNSIGNED;
    srcaddr ADDRESS;                    /* srcaddr.ab
    XXXXX LONGWORD FILL;
    dstaddr ADDRESS;                    /* dstaddr.ab

END;

END_MODULE movp_def;
```

```
MODULE mulp_def;

/* Define register usage for the MULP instruction
/*
/*     31                23                15                07              00
/*    +-----------------+-----------------+-----------------+----------------+
/*    |    delta-PC     |      XXXX       |                 mulrlen           | : R0
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                 mulraddr                             | : R1
/*    +-----------------+-----------------+-----------------+----------------+
/*    |               XXXXX               |                 muldlen           | : R2
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                 muldaddr                             | : R3
/*    +-----------------+-----------------+-----------------+----------------+
/*    |               XXXXX               |                 prodlen           | : R4
/*    +-----------------+-----------------+-----------------+----------------+
/*    |                                 prodaddr                             | : R5
/*    +-----------------+-----------------+-----------------+----------------+

AGGREGATE mulp STRUCTURE PREFIX "mulp_";

    mulrlen STRUCTURE WORD UNSIGNED;    /* mulrlen.rw
        mulrlen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    mulraddr ADDRESS;                   /* mulraddr.ab
    muldlen STRUCTURE WORD UNSIGNED;    /* muldlen.rw
        muldlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    muldaddr ADDRESS;                   /* muldaddr.ab
    prodlen STRUCTURE WORD UNSIGNED;    /* prodlen.rw
        prodlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    prodaddr ADDRESS;                   /* prodaddr.ab

END;

END_MODULE mulp_def;
```

```
MODULE subp4_def;

/* Define register usage for the SUBP4 instruction
/*
/*     31               23               15               07              00
/*     +----------------+----------------+----------------+----------------+
/*     |    delta-PC    |      XXXX      |                sublen           | : R0
/*     +----------------+----------------+----------------+----------------+
/*     |                                 subaddr                           | : R1
/*     +----------------+----------------+----------------+----------------+
/*     |               XXXXX             |                diflen           | : R2
/*     +----------------+----------------+----------------+----------------+
/*     |                                 difaddr                           | : R3
/*     +----------------+----------------+----------------+----------------+

AGGREGATE subp4 STRUCTURE PREFIX "subp4_";

    sublen STRUCTURE WORD UNSIGNED;     /* sublen.rw
        sublen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    subaddr ADDRESS;                    /* subaddr.ab
    diflen STRUCTURE WORD UNSIGNED;     /* diflen.rw
        diflen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    difaddr ADDRESS;                    /* difaddr.ab

END;

END_MODULE subp4_def;
```

```
MODULE subp6_def;

/* Define register usage for the SUBP6 instruction
/*
/*      31              23              15              07              00
/*     +---------------+---------------+---------------+---------------+
/*     |    delta-PC   |      XXXX     |             sublen            | : R0
/*     +---------------+---------------+---------------+---------------+
/*     |                           subaddr                             | : R1
/*     +---------------+---------------+---------------+---------------+
/*     |               XXXXX           |             minlen            | : R2
/*     +---------------+---------------+---------------+---------------+
/*     |                           minaddr                             | : R3
/*     +---------------+---------------+---------------+---------------+
/*     |               XXXXX           |             diflen            | : R4
/*     +---------------+---------------+---------------+---------------+
/*     |                           difaddr                             | : R5
/*     +---------------+---------------+---------------+---------------+

AGGREGATE subp6 STRUCTURE PREFIX "subp6_";

    sublen STRUCTURE WORD UNSIGNED;     /* sublen.rw
        sublen BITFIELD LENGTH 5;
        END;
    XXXX BYTE FILL;
    delta_PC BYTE UNSIGNED;
    subaddr ADDRESS;                    /* subaddr.ab
    minlen STRUCTURE WORD UNSIGNED;     /* minlen.rw
        minlen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    minaddr ADDRESS;                    /* minaddr.ab
    diflen STRUCTURE WORD UNSIGNED;     /* diflen.rw
        diflen BITFIELD LENGTH 5;
        END;
    XXXXX WORD FILL;
    difaddr ADDRESS;                    /* difaddr.ab

END;

END_MODULE subp6_def;
```

```
MODULE editpc_def;

/* Define register usage for the EDITPC instruction
/*
/*     31               23               15               07              00
/*    +----------------+----------------+----------------+----------------+
/*    |           zero count            |             srclen              | : R0
/*    +----------------+----------------+----------------+----------------+
/*    |                              srcaddr                               | : R1
/*    +----------------+----------------+----------------+----------------+
/*    |  delta-srcaddr |    delta-PC    |      sign      |      fill      | : R2
/*    +----------------+----------------+----------------+----------------+
/*    |                              pattern                               | : R3
/*    +----------------+----------------+----------------+----------------+
/*    |   loop-count   |     state      |    saved-PSW   |   inisrclen    | : R4
/*    +----------------+----------------+----------------+----------------+
/*    |                              dstaddr                               | : R5
/*    +----------------+----------------+----------------+----------------+

AGGREGATE editpc STRUCTURE PREFIX "editpc_";

    srclen STRUCTURE WORD UNSIGNED;      /* srclen.rw
        srclen STRUCTURE BYTE UNSIGNED;
            srclen BITFIELD LENGTH 5;
            END;
        eo_read_char BYTE UNSIGNED;
        END;
    zero_count WORD;
    srcaddr ADDRESS;                     /* srcaddr.ab
    fill BYTE UNSIGNED;
    sign BYTE UNSIGNED;
    delta_PC BYTE UNSIGNED;
    delta_srcaddr BYTE UNSIGNED;         /* current srcaddr minus initial srcaddr
    pattern ADDRESS;                     /* pattern.ab

    editpc_overlay UNION PREFIX editpc_ FILL;

        editpc_overlay_1 STRUCTURE FILL;
            inisrclen BYTE UNSIGNED;
            saved_PSW BYTE UNSIGNED;
            state STRUCTURE BYTE UNSIGNED;
                state BITFIELD LENGTH 4 MASK;
                FPD BITFIELD LENGTH 1 MASK;
                END;
            loop_count BYTE UNSIGNED;
            dstaddr ADDRESS;                     /* dstaddr.ab
            END editpc_overlay_1;

        editpc_overlay_2 STRUCTURE FILL;
            saved_R0 LONGWORD UNSIGNED;     /* initial srclen.rw
            saved_R1 LONGWORD UNSIGNED;     /* initial srcaddr.ab
            END editpc_overlay_2;

        END editpc_overlay;

END editpc;
```

```
END_MODULE editpc_def;
```

XLATE
LIS

VAXEMUL
MAP

WIOHNDLR
LIS

WORKIO
LIS

BOOSTRING
LIS

EMULAT

FPEMUL
MAP

BOOEMULAT
LIS

WORDWRAP
LIS

VAXMACROS
MAR

WRIEDTMSG
LIS

FPEMULATE
LIS

VAXREGDEF
SDL