







{ 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.

{--

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
initsrclen 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

```

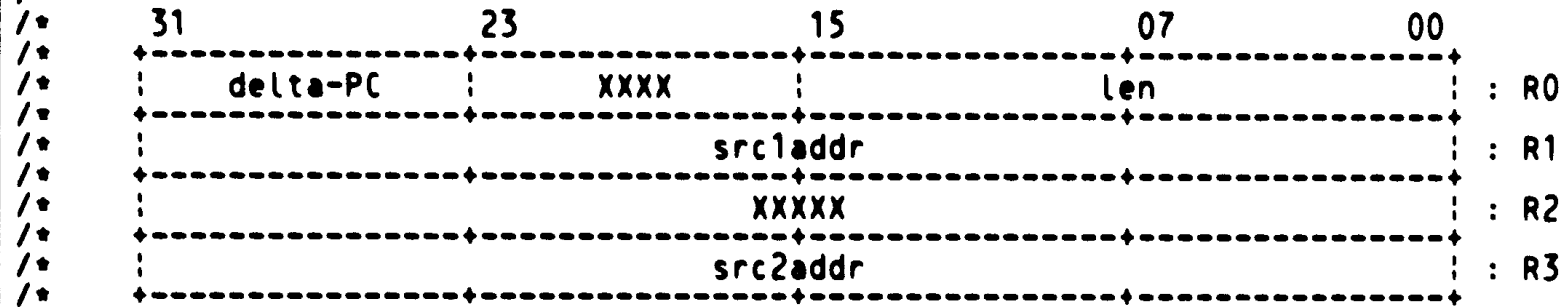
END;

END\_MODULE movtc\_def;



MODULE cmpc3\_def;

/\* Define register usage for the CMPC3 instruction



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

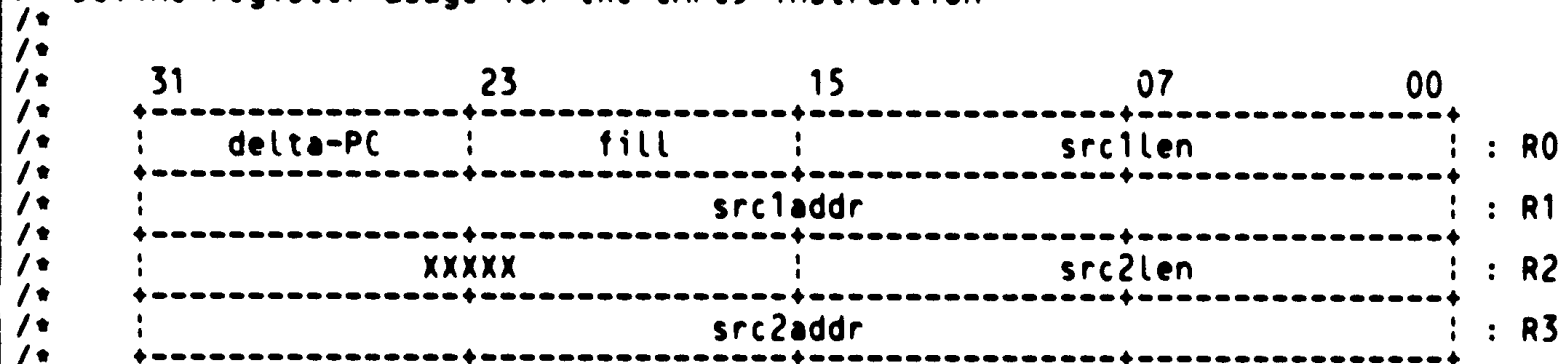
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*
/\*

AGG

END
END

MODULE cmpc5\_def;

/\* Define register usage for the CMPC5 instruction



AGGREGATE cmpc5 STRUCTURE PREFIX "cmpc5\_":

```

src1len WORD UNSIGNED; /* src1len.rw
fill BYTE UNSIGNED; /* fill.rt
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;

VAX

NOE

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

AGC

END

END





MODULE spanc\_def;

/\* Define register usage for the SPANC instruction

```

/*
/*      31          23          15          07          00
/*      +-----+-----+-----+-----+
/*      | delta-PC | XXXX |          len          | : R0
/*      +-----+-----+-----+-----+
/*      |          |          |          addr          | : R1
/*      +-----+-----+-----+-----+
/*      |          |          |          XXXX          | : R2
/*      |          |          |          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;

VA)

MOE

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

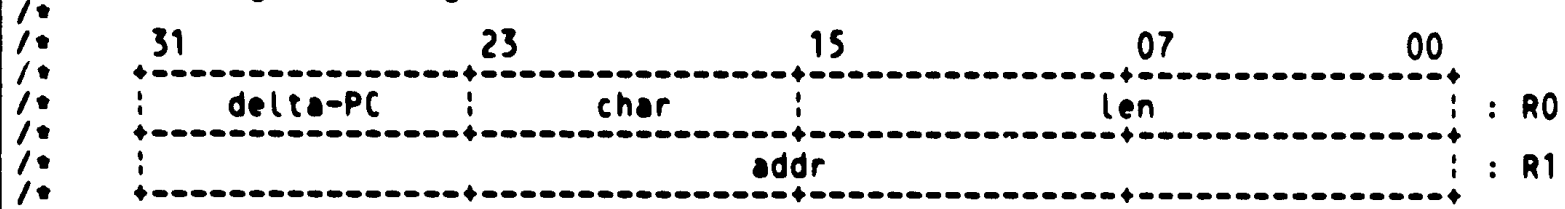
AGC

ENC

ENC

MODULE locc\_def;

/\* Define register usage for the LOCC instruction



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;

VAX

MOD

```

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

```

AGG

END

END



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;

VAX

PROC

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

/\*

AGG

ENC

```
MODULE crc_def;
```

```
/* Define register usage for the CRC instruction
```

```
/*
/*      31          23          15          07          00
/*      +-----+-----+-----+-----+
/*      |                               | : R0
/*      |                               | : R1
/*      |                               | : R2
/*      | delta-PC | XXXX |           | : R2
/*      |-----+-----+-----+-----+
/*      |                               | : R3
/*      |                               | : R3
/*      +-----+-----+-----+-----+
```

```
AGGREGATE crc STRUCTURE PREFIX "crc_";
```

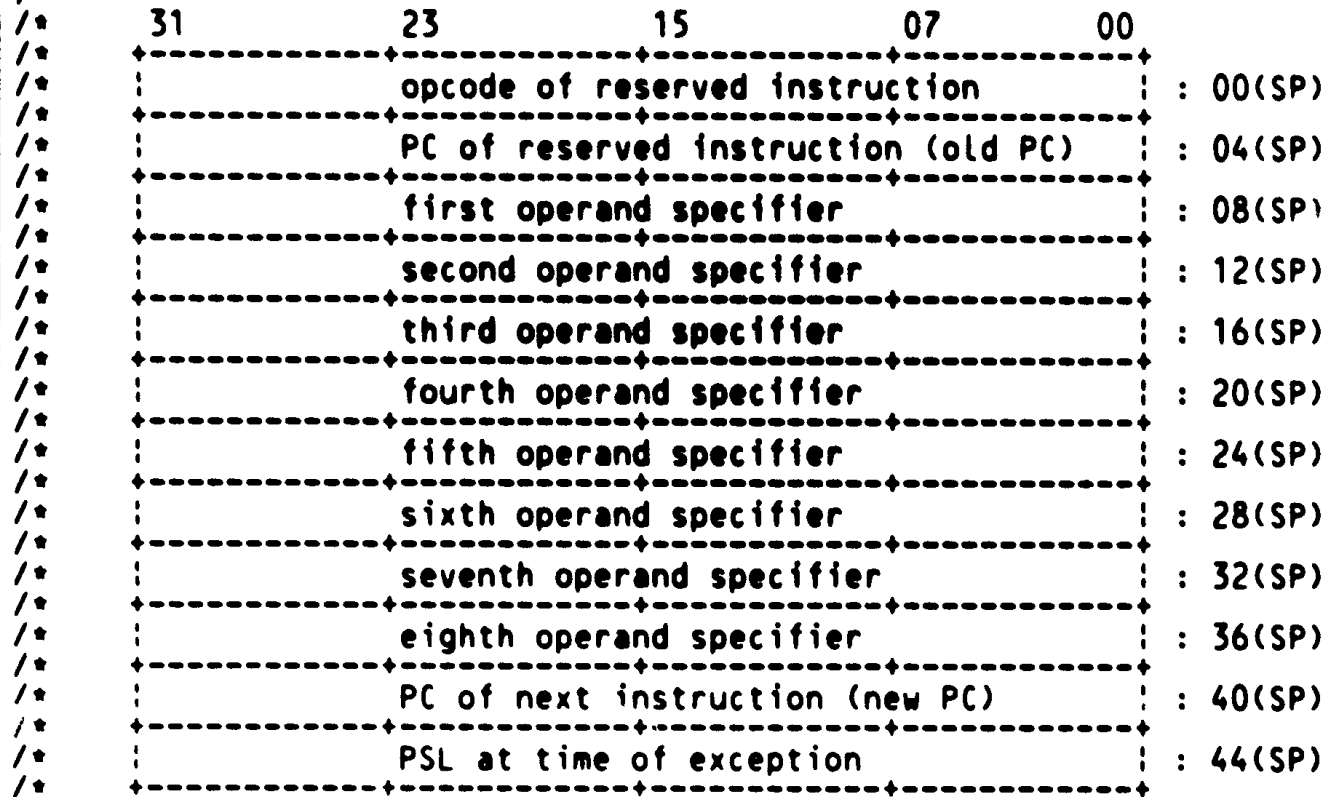
```
inicrc LONGWORD UNSIGNED;      /* inicrc.rl
tbl ADDRESS;                   /* tbl.ab
strlen WORD UNSIGNED;         /* strlen.rw
XXXX BYTE FILL;
delta_PC BYTE UNSIGNED;
stream ADDRESS;               /* stream.ab
```

```
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.



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 ""; /* Eighth operand specifier
new_PC LONGWORD TAG ""; /* PC of instruction following
reserved LONGWORD TAG ""; /* reserved instruction (new PC)
exception_PSL LONGWORD TAG ""; /* PSL at time of exception
  
```

LND;

END\_MODULE stack\_def;

MODULE pack\_def;

/\* Define stack usage when packing registers to back up instruction

```

/*
/*      31          23          15          07          00
/*      +-----+-----+-----+-----+
/*      |                                     | : 00(SP)
/*      |          saved R0                   | :
/*      +-----+-----+-----+-----+
/*      |                                     | : 04(SP)
/*      |          saved R1                   | :
/*      +-----+-----+-----+-----+
/*      |                                     | : 08(SP)
/*      |          saved R2                   | :
/*      +-----+-----+-----+-----+
/*      |                                     | : 12(SP)
/*      |          saved R3                   | :
/*      +-----+-----+-----+-----+

```

/\* 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;

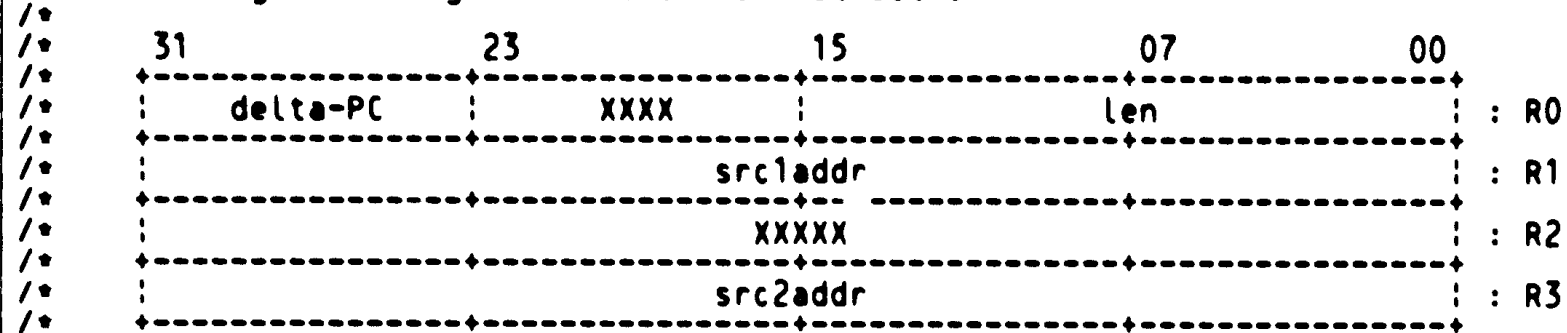






MODULE cmpp3\_def;

/\* Define register usage for the CMPP3 instruction



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 cvtpl_def;
```

END

```
/* Define register usage for the CVTPL instruction
```

```
/*
/*      31          23          15          07          00
/*      +-----+-----+-----+-----+
/*      | delta-PC | state | srcaddr | srcaddr | : R0
/*      +-----+-----+-----+-----+
/*      |          |          |          |          | : R1
/*      |          |          |          |          | : R2
/*      |          |          |          |          | : 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;

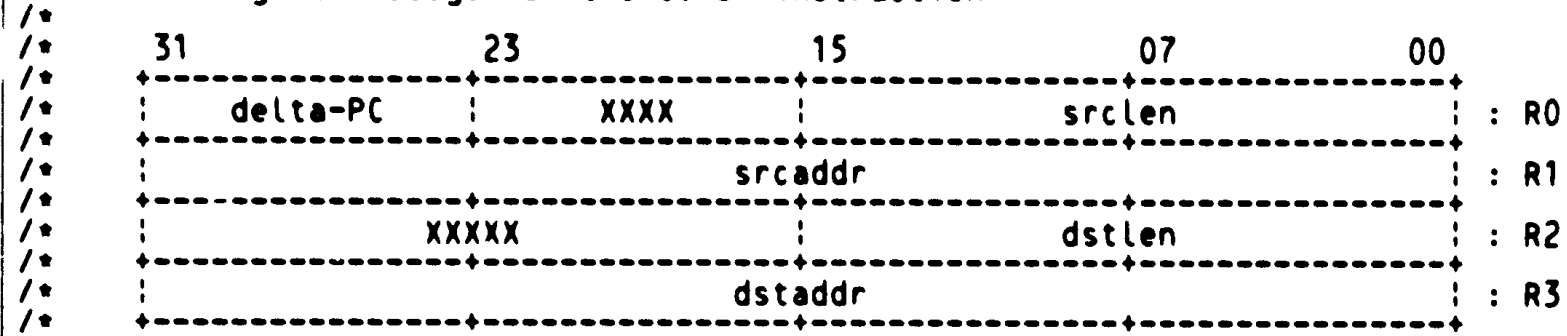
END\_MODULE cvtps\_def;





MODULE cvtsp\_def;

/\* Define register usage for the CVTSP instruction



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;

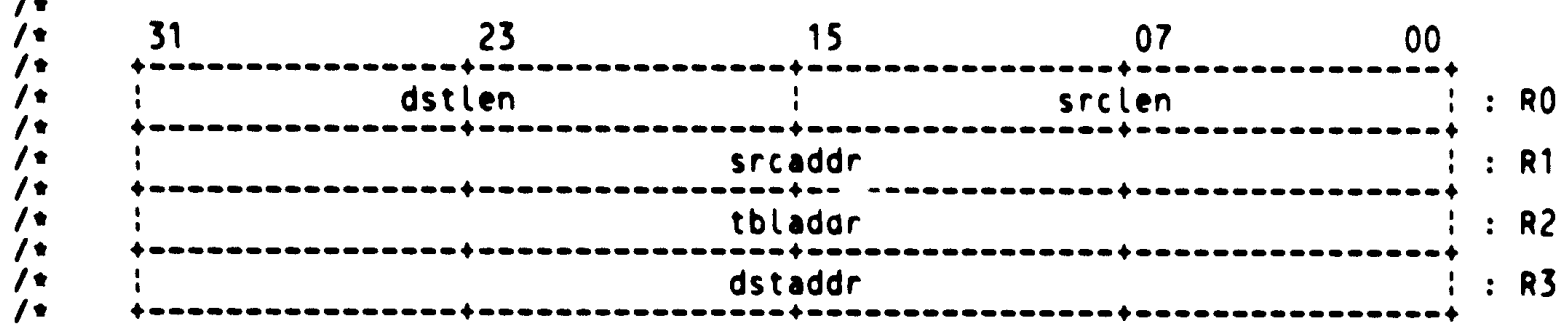
PC\_

HAN

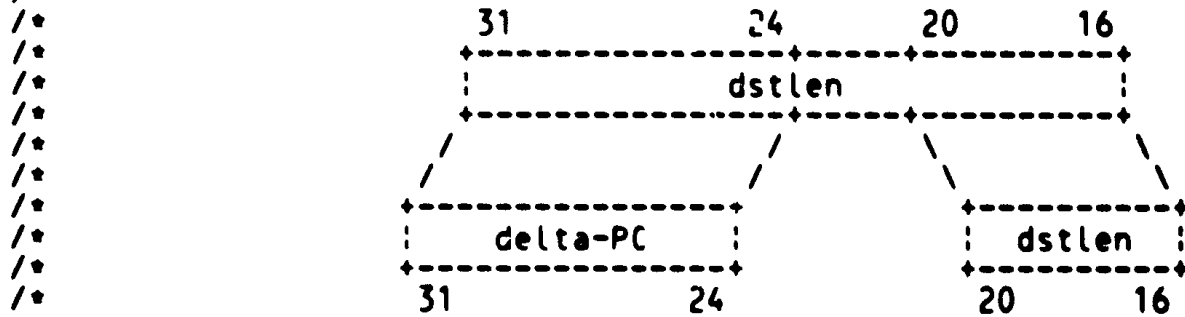
RES

MODULE cvttp\_def;

/\* Define register usage for the CVTTP instruction



/\* 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.



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;

VAXREGDEF.SDL;1

END\_MODULE cvttp\_def;

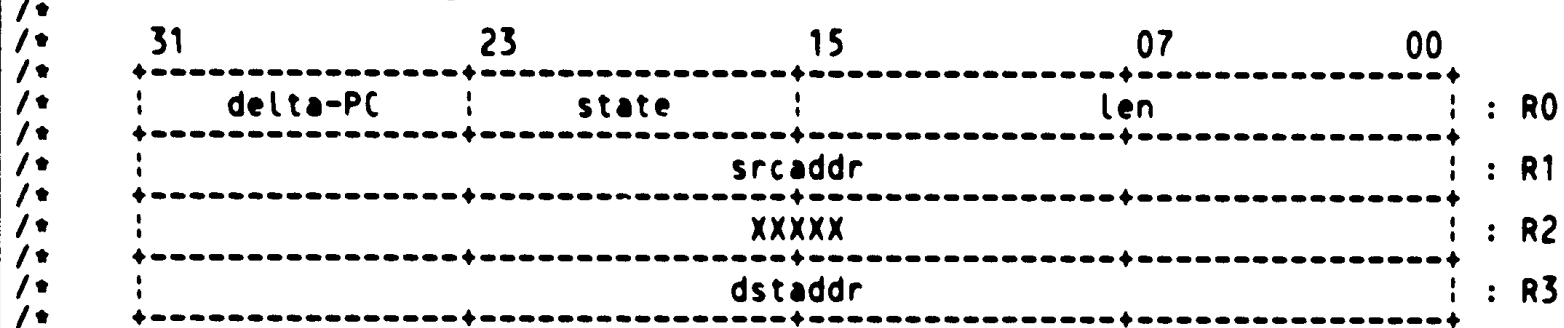
VAXI

↑  
.....  
↓



MODULE movp\_def;

/\* Define register usage for the MOVVP instruction



AGGREGATE movp STRUCTURE PREFIX 'movp\_';

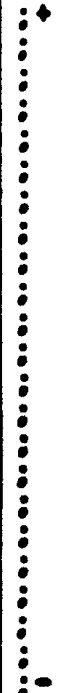
```

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;



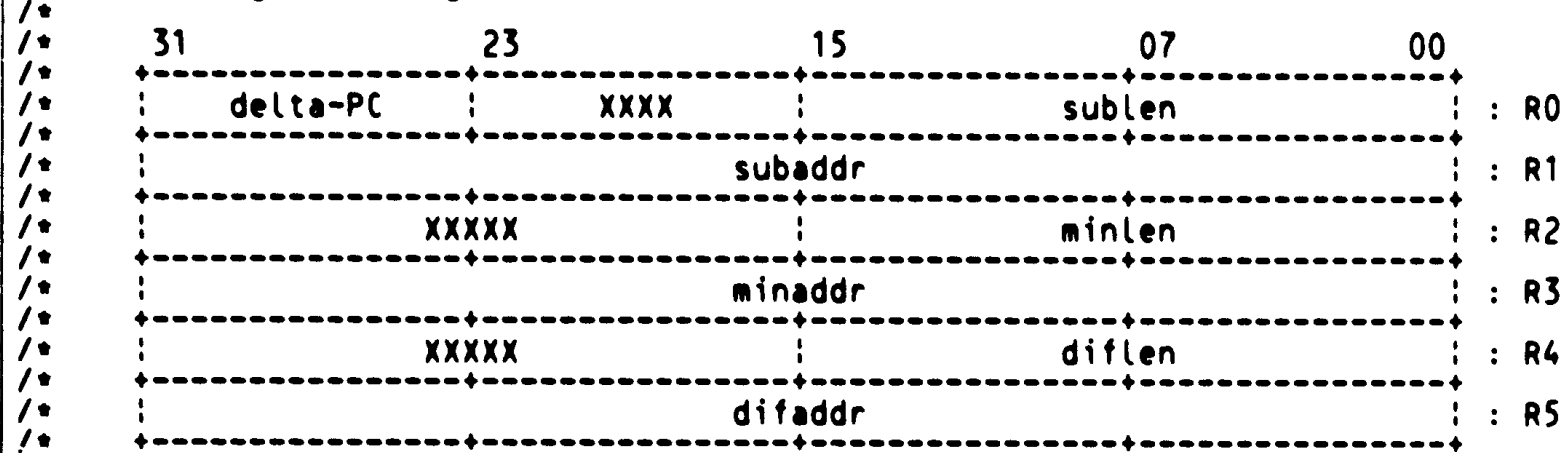






MODULE subp6\_def;

/\* Define register usage for the SUBP6 instruction



BAS

MAX

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;
```

VAXREGDEF.SDL;1

16-SEP-1984 16:40:37.61<sup>9</sup> Page 35

800'  
Tab

END\_MODULE editpc\_def;



0142 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a dense grid of approximately 100 small, overlapping windows or document pages. Each window contains text, likely technical specifications or code, and is arranged in a regular, repeating pattern. The text is mostly illegible due to the small size and low contrast, but several larger, more prominent windows are visible. These include:

- XLATE LIS** and **VAXEMUL MAP** (top center)
- WLOHNDLR LIS** and **WORKIO LIS** (middle left)
- BOOSTRING LIS** (middle right)
- EMULAT** (center)
- FPREMUL MAP** (center)
- BOGEMULAT LIS** (center)
- WORDWRAP LIS** (bottom left)
- VAXMACROS MAR** (bottom center)
- WRTEDTMSG LIS** (bottom left)
- VAXREGDEF SDL** (bottom center)