CDPEAC Reference Card

Mode Setting Operations

```
vmmode(mode)
vmmode_s(mode)
```
available modes: cond, always, condalu, condmem.
```
vmold
vmnew
vmnop
vmcurrent
vmrotate
vminvert
vmtrue
align
noalign
vmcount(dreg)
vmcountv(dreg)
vmcounts(dreg)
exchange
noexchange
pad(n)
nopad
set_vmmode(vmmode)
set_mem_stride(stride)
set_rs1_stride(rs1_stride)
set_vector_length(vlen)
set_vector_length_and_vmmode(vlen,vmmode)
set_vector_length_and_rs1_stride(vlen,rs1_stride)
set_vector_length_and_rs1_stride_and_vmmode(vlen,
rs1_stride,vmmode)
```

Join Operations

```
join(instruction1,instruction2)
join2(instruction1,instruction2)
join3(instruction1,...,instruction3)
join4(instruction1,...,instruction4)
join5(instruction1,...,instruction5)
```

Accessors

```
dpwrt[_sync,_nosync](type,selector,src,dreg)
dprd[_sync,_nosync](type,selector,dreg,dst)
dpset[_supervisor](type,selector,src,creg)
dpget[_supervisor](type,selector,creg,dst)
dpld(type,src,dst)
dpst(type,src,dst)
dpchgsp(src,dst)
dpchgbk(src,selector,dst)
```

Population Count

```
epc{s,v}(type,src,dest)
```

Control Operators

```
dpsync()
ldvm(dreg)
stvm(dreg)
dpsetup()
dpcleanup()
```

Register Expression Operators

```
dreg_x(dreg,index)
dreg_i(dreg,ireg)
dreg_u(dreg,stride)
dreg_s(dreg,setstride)
dreg_u_s(dreg,stride,setstride)
scalar(dreg)
SCALAR(dreg)
```

Memory Operations

```
load{s,v}(type,memop,dreg)
load{s,v}_i(type,memop,ireg,dreg)
load{s,v}_{v,vs,vh,vhs}(type,vlen,memop,dreg)
load{s,v}_u(type,memop,ustride,dreg)
load{s,v}_s(type,memop,sstride,dreg)
load{s,v}_u_s(type,memop,ustride,sstride,dreg)
load{s,v}_v_u(type,vlen,memop,ustride,dreg)
store{s,v}(type,memop,dreg)
store{s,v}_i(type,memop,ireg,dreg)
store{s,v}_{v,vs,vh,vhs}(type,vlen,memop,dreg)
store{s,v}_u(type,memop,ustride,dreg)
store{s,v}_s(type,memop,sstride,dreg)
store{s,v}_u_s(type,memop,ustride,sstride,dreg)
store{s,v}_v_u(type,vlen,memop,ustride,dreg)
```

Shift Operators

```
shl{s,v}[i](type,src1,src2,dest)
shlr{s,v}[i](type,src1,src2,dest)
shr{s,v}[i](type,src1,src2,dest)
shrr{s,v}[i](type,src1,src2,dest)
shl{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
shlr{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
shr{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
shrr{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
```

Monadic Arithmetic Operators

```
move{s,v}[i](type,src,dest)
test{s,v}[i](type,src,dest)
neg{s,v}[i](type,src,dest)
abs{s,v}[i](type,src,dest)
not {s,v}[i](type,src,dest)
tos{s,v}[i](type,src,dest)
sqrt{s,v}[i](type,src,dest)
ffb{s,v}[i](type,src,dest)
inv{s,v}[i](type,src,dest)
move{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
test{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
neg{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
abs{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
not {s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
tos{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
sqrt{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
ffb{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
inv{s,v}_{v,vs,vh,vhs}(type,vlen,src,dest)
```

Dyadic Arithmetic Operators

```
add{s,v}[i](type,src1,src2,dest)
addc{s,v}[i](type,src1,src2,dest)
sub{s,v}[i](type,src1,src2,dest)
subc{s,v}[i](type,src1,src2,dest)
subr{s,v}[i](type,src1,src2,dest)
sbrc{s,v}[i](type,src1,src2,dest)
mul{s,v}[i](type,src1,src2,dest)
mulh{s,v}[i](type,src1,src2,dest)
div{s,v}[i](type,src1,src2,dest)
enc{s,v}[i](type,src1,src2,dest)
add{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
addc{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
sub{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
subc{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
subr{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
sbrc{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
mul{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
mulh{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
div{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
enc{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
```

Bitwise Logical Operators

```
and{s,v}[i](type,src1,s2reg,dest)
nand{s,v}[i](type,src1,s2reg,dest)
andc{s,v}[i](type,src1,s2reg,dest)
or{s,v}[i](type,src1,s2reg,dest)
nor{s,v}[i](type,src1,s2reg,dest)
xor{s,v}[i](type,src1,s2reg,dest)
and{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
nand{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
andc{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
or{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
nor{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
xor{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
```

Logical Comparisons

```
gt{s,v}[i](type,src1,src2)
ge{s,v}[i](type,src1,src2)
lt{s,v}[i](type,src1,src2)
le{s,v}[i](type,src1,src2)
eq{s,v}[i](type,src1,src2)
ne{s,v}[i](type,src1,src2)
lg{s,v}[i](type,src1,src2)
un{s,v}[i](type,src1,src2)
cmp{s,v}[i](type,src1,src2)
gt{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
ge{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
lt{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
le{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
eq{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
ne{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
lg{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
un{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
cmp{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2)
mrg{s,v}[i](type,src1,src2,dest)
mrg{s,v}_{v,vs,vh,vhs}(type,vlen,src1 src2,dest)
```

Dyadic Two-operation Operators

```
mada{s,v}[i](type,src1,src2,dest)
msba{s,v}[i](type,src1,src2,dest)
msra{s,v}[i](type,src1,src2,dest)
nmaa{s,v}[i](type,src1,src2,dest)
madi{s,v}[i](type,src1,src2,dest)
msri{s,v}[i](type,src1,src2,dest)
nmai{s,v}[i](type,src1,src2,dest)
mada{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
msba{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
msra{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
nmaa{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
madi{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
msri{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
nmai{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,dest)
```

Triadic Two-operation Operators

```
madt{s,v}[i](type,src1,src2,src3,dest)
msbt{s,v}[i](type,src1,src2,src3,dest)
msrt{s,v}[i](type,src1,src2,src3,dest)
nmat{s,v}[i](type,src1,src2,src3,dest)
madt{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,src3,dest)
msbt{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,src3,dest)
msrt{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,src3,dest)
nmat{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,src3,dest)
```

Multiply with Logical Instructions

```
m[h]{s,m,o,x}{a,i}{s,v}[i](type,src1,src2,dest)
m[h]{s,m,o,x}{a,i}{s,v}_{v,vs,vh,vhs}(type,vlen,src1,
src2,dest)
m[h]{s,m,o,x}t{s,v}[i](type,src1,src2,src3,dest)
m[h]{s,m,o,x}t{s,v}_{v,vs,vh,vhs}(type,vlen,src1,src2,
src3,dest)
```