RVV和C910中的指令对比

结论：

1 **C910新增矢量**

VAMO指令（5.7.10 - 5.7.36），riscv-v-spec 0.7.1中Table 13定义了amoop，但是具体的指令格式都没有定义，vamo\*所用的Inst(6-0)= 0101111，拿原子指令AMO没用完的部分定义的。

5.7.346 VWSMACC.VV——矢量有符号带饱和扩位乘累加指令

5.7.347 VWSMACC.VX——矢量标量有符号带饱和扩位乘累加指令

5.7.348 VWSMACCSU.VV——矢量带饱和取反有符号无符号扩位乘累减指令

5.7.349 VWSMACCSU.VX——矢量标量带饱和取反有符号无符号扩位乘累减指令

5.7.350 VWSMACCU.VV——矢量无符号带饱和扩位乘累加指令

5.7.351 VWSMACCU.VX——矢量标量无符号带饱和扩位乘累加指令

5.7.352 VWSMACCUS.VX——矢量标量带饱和取反有符号无符号扩位乘累减指令

**2 C910有但没有具体的指令格式**

vfncvt.f.f.v vd, vs2, vm # Convert double-width float to single-width float.

vfrsub.vf vd, vs2, rs1, vm # Scalar-vector vd[i] = f[rs1] - vs2[i]

**3 C910没有RVV有**

vmseq.vi vd, vs2, imm, vm # vector-immediate

vmsgeu和vmsge（Set if greater than or equal）指令

vdotu.vv vd, vs2, vs1, vm # Vector-vector Unsigned dot-product

vdot.vv vd, vs2, vs1, vm # Vector-vector Signed dot-product

vfdot.v vd, vs2, vs1, vm # vd[i][31:0] += vs2[i][31:16] \* vs1[i][31:16]+vs2[i][15:0] \* vs1[i][15:0]

vfdot.vv vd, vs2, vs1, vm # Vector-vector

一 指令对比

当配置矢量执行单元时，C910 拥有 32 个独立的 128 位矢量寄存器，可在普通用户模式、超级用户模式和机器模式下被访问。矢量寄存器通过矢量传送指令实现与通用寄存器/浮点寄存器的数据交换。

* 1. 向量整数计算指令

------**Vector Single-Width Integer Add and Subtract**

vadd\*，vsub\*，vrsub\*都一样

------**Vector Widening Integer Add/Subtract**

vwaddu\*，vwsubu\*，vwadd\*，vwsub\*都一样

C910同时继承了**扩位**（vwaddu\*，vwsubu\*，vwadd\*，vwsub\*）指令.

例如：VWSUB.VV——矢量整型有符号扩位减法指令，操作是：vd[i] <- sign\_extend(vs2[i]) - sign\_extend(vs1[i])

VWSUB.WV——扩位矢量整型有符号扩位减法指令，操作是： vd[i] <- vs2[i] - sign\_extend(vs1[i])

------**Vector Integer Add-with-Carry / Subtract-with-Borrow Instructions**

vadc\*，vmadc\*，vsbc\*，vmsbc\*都一样

------ **Vector Bitwise Logical Instructions**

vand\*，vor\*，vxor\*都一样

------**Vector Single-Width Bit Shift Instructions**

vsll\*，vsrl\*，vsra\*都一样

------Vector Narrowing Integer Right Shift Instructions

vnsrl\*，vnsra\*都一样

------Vector Integer Comparison Instructions

vmseq\*，vmsne\*，vmsltu\*，vmslt\*，vmsleu\*，vmsle\*，vmsgtu\*，vmsgt\*都一样

C910没有定义 vmseq.vi （矢量立即数整型比较相等指令）指令的具体编码，没有定义vmsgeu和vmsge（Set if greater than or equal）指令

------Vector Integer Min/Max Instructions

vminu\*，vmin\*，vmaxu\*，vmax\*都一样

------Vector Single-Width Integer Multiply Instructions

vmul\*，vmulh\*，vmulhu\*，vmulhsu\*都一样

------ Vector Integer Divide Instructions

vdivu\*，vdiv\*，vremu\*，vrem\*都一样

------ Vector Widening Integer Multiply Instructions

vwmul\*，vwmulu\*，vwmulsu\*都一样

------Vector Single-Width Integer Multiply-Add Instructions

vmacc\*，vmadd\*，vnmsac\*，vnmsub\*都一样

------Vector Widening Integer Multiply-Add Instructions

vwmaccu\*，vwmaccus\*，vwmacc\*，vwmaccsu\*都一样

------ Vector Integer Merge and Move Instructions

vmerge\*，vmv.v\*都一样

1.2 Vector Fixed-Point Arithmetic Instructions

------ Vector Single-Width Saturating Add and Subtract

vsaddu\*，vsadd\*，vssub\*，vssubu\*都一样

------ Vector Single-Width Averaging Add and Subtract

vaadd\*，vasub\*都一样

------Vector Single-Width Fractional Multiply with Rounding and Saturation

vsmul\*都一样

------Vector Widening Saturating Scaled Multiply-Add

Vwsmaccu\*，vwsmaccus\*，vwsmaccsu\*，vwsmacc\*都一样

------ Vector Single-Width Scaling Shift Instructions

vssrl\*，vssra\*都一样

------Vector Narrowing Fixed-Point Clip Instructions

vnclip\*，vnclipu\*都一样

1.3 Vector Floating-Point Instructions

------Vector Single-Width Floating-Point Add/Subtract Instructions

vfadd\*，vfsub\*都一样

C910没有定义vfrsub，vfrsub.vf vd, vs2, rs1, vm # Scalar-vector vd[i] = f[rs1] - vs2[i]

------Vector Widening Floating-Point Add/Subtract Instructions

vfwadd\*，vfwsub\*都一样

------Vector Single-Width Floating-Point Multiply/Divide Instructions

vfmul\*， vfdiv\*，vfrdiv都一样

------Vector Widening Floating-Point Multiply

vfwmul\*都一样

------Vector Single-Width Floating-Point Fused Multiply-Add Instructions

vfmacc\*，vfnmacc\*，vfmsac\*，vfnmsac\*，vfmadd\*，vfnmadd\*，vfmsub\*，vfnmsub\*都一样

------Vector Widening Floating-Point Fused Multiply-Add Instructions

Vfwmacc\*，vfwnmacc\*，vfwmsac\*，vfwnmsac\*都一样

------Vector Floating-Point Square-Root Instruction

Vfsqrt（矢量浮点开方指令 ）都一样

------Vector Floating-Point MIN/MAX Instructions

vfmin\*，vfmax\*都一样

------Vector Floating-Point Sign-Injection Instructions

vfsgnj\*，vfsgnjn\*，vfsgnjx\*都一样

------Vector Floating-Point Compare Instructions

vmfeq\*，vmfne\*，vmflt\*，vmfle\*，vmfgt.vf，vmfge.vf， vmford\*都一样

------Vector Floating-Point Classify Instruction

Vfclass（矢量浮点分类指令）都一样

------Vector Floating-Point Merge Instruction

vfmerge.vfm （矢量浮点元素选择指令），vfmv.v.f（矢量浮点标量传送指令 ）都一样

------Single-Width Floating-Point/Integer Type-Convert Instructions

vfcvt.xu.f.v，vfcvt.x.f.v，vfcvt.f.xu.v，vfcvt.f.x.v 都一样

------ Widening Floating-Point/Integer Type-Convert Instructions

vfwcvt.xu.f.v，vfwcvt.x.f.v ，vfwcvt.f.xu.v ，vfwcvt.f.x.v，vfwcvt.f.f.v 都一样

------Narrowing Floating-Point/Integer Type-Convert Instructions

vfncvt.xu.f.v ，vfncvt.x.f.v ，vfncvt.f.xu.v，vfncvt.f.x.v ，vfncvt.f.f.v 都一样

1.4 Vector Reduction Operations

------Vector Single-Width Integer Reduction Instructions

vredsum.vs ， vredmaxu.vs ， vredmax.vs ， vredminu.vs ， vredmin.vs ，vredand.vs ， vredor.vs ，vredxor.vs 都一样

------Vector Widening Integer Reduction Instructions

Vwredsum，vwredsumu都一样

------ Vector Single-Width Floating-Point Reduction Instructions

vfredosum.vs，vfredsum.vs，vfredmax.vs，vfredmin.vs 都一样

------Vector Widening Floating-Point Reduction Instructions

vfwredosum.vs， vfwredsum.vs 都一样

1.5 Vector Mask Instructions

------Vector Mask-Register Logical Instructions

vmand.mm ，vmnand.mm，vmandnot.mm，vmxor.mm，vmor.mm ，vmnor.mm，vmornot.mm，vmxnor.mm 都一样

------Vector mask population count vmpopc

vmpopc.m 都一样

------vmfirst nd-rst-set mask bit

vmfirst.m 都一样

------ vmsbf.m set-before-rst mask bit

vmsbf.m 都一样

------ vmsif.m set-including-rst mask bit

vmsif.m 都一样

------ vmsof.m set-only-rst mask bit

vmsof.m 都一样

------ Vector Iota Instruction

viota.m 都一样

------ Vector Element Index Instruction

vid.v 都一样

1.6 Vector Permutation Instructions

------Integer Extract Instruction

vext.x.v 都一样

------Integer Scalar Move Instruction

vmv.s.x 都一样

------Floating-Point Scalar Move Instructions

vfmv.f.s ，vfmv.s.f 都一样

------Vector Slide Instructions

Vslideup\*， vslidedown\*， vslide1up.vx， vslide1down.vx都一样

------Vector Register Gather Instruction

vrgather\*都一样

------Vector Compress Instruction

vcompress.vm 都一样

1.6 Divided Element Extension ('Zvediv')

------ Instructions Affected by EDIV

Vsetvli，vwredsum.vs，vredmax.vs 都一样

------ Vector Integer Dot-Product Instruction

vdotu.vv，vdot.vv 是riscv-v-spec官方定义（可以查阅rvv-llvm看其源代码）

------ Vector Floating-Point Dot Product Instruction

vfdot.vv，vfdot.v是riscv-v-spec官方定义（可以查阅rvv-llvm看其源代码）

二 C910新增矢量

VAMO指令（5.7.10 - 5.7.36），riscv-v-spec 0.7.1中Table 13定义了amoop，但是具体的指令格式都没有定义，vamo\*所用的Inst(6-0)= 0101111，拿原子指令AMO没用完的部分定义的。

5.7.346 VWSMACC.VV（矢量有符号带饱和扩位乘累加指令）

5.7.347 VWSMACC.VX——矢量标量有符号带饱和扩位乘累加指令

5.7.348 VWSMACCSU.VV——矢量带饱和取反有符号无符号扩位乘累减指令

5.7.349 VWSMACCSU.VX——矢量标量带饱和取反有符号无符号扩位乘累减指令

5.7.350 VWSMACCU.VV——矢量无符号带饱和扩位乘累加指令

5.7.351 VWSMACCU.VX——矢量标量无符号带饱和扩位乘累加指令

5.7.352 VWSMACCUS.VX——矢量标量带饱和取反有符号无符号扩位乘累减指令

三 C910新增矢量中有但没有具体的指令格式

vfncvt.f.f.v vd, vs2, vm # Convert double-width float to single-width float.

四 C910没有RVV有

vmseq.vi vd, vs2, imm, vm # vector-immediate

表1：两个指令集都支持的指令

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vaadd.vv | vadd.vv | vamoaddw.v |  |  |  |  |  |  |  |
| vaadd.vx | vadd.vx |  |  |  |  |  |  |  |  |
| vadc.vim | vadd.vi |  |  |  |  |  |  |  |  |
| vadc.vvm | vadc.vxm |  |  |  |  |  |  |  |  |

表2：仅C910支持的指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| VAMOADDD.V |  |  |  |  |  |  |
| VAMOADDQ.V |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

表3：仅RVV支持的指令

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

二 vor.vv指令在RVV和C910中的对比

2.1 RISC-V "V" Vector Extension中的vor.vv指令

在RISC-V "V" Vector Extension指令集手册的第12章Vector Integer Arithmetic Instructions的12.4. Vector Bitwise Logical Instructions中定义了vor.vv指令。

vor.vv vd, vs2, vs1, vm # Vector-vector

https://github.com/riscv/riscv-v-spec/blob/master/inst-table.adoc

所示的编码表中，vor指令的指令编码是001010，支持操作数是Vector-vector，vector-scalar和 vector-immediate这三种形式。

在rvv-llvm/RISCVInstrInfoV.td中对vor.vv的定义：

let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in

multiclass VALU\_OPIVV<bits<6> funct6, string opcodestr> {

def \_m : RVInstVA<funct6, 0b000, RVV\_Masked, OPC\_OP\_V,

(outs VR:$vd), (ins VR:$vs2, VR:$vs1, VLR:$vl, VMR:$vm),

opcodestr, "$vd, $vs2, $vs1, $vm"> {

bits<5> vs1;

let Inst{19-15} = vs1;

}

... ...

defm VOR\_VV : VALU\_OPIVV<0b001010, "vor.vv">;

在RISCVInstrFormatsV.td中

// Formats for Vector Arithmetic Instructions under OP-V major opcode

class RVInstVA<bits<6> funct6, bits<3> funct3, RVVMaskCond m, RISCVOpcode opcode,

dag outs, dag ins, string opcodestr, string argstr>

: RVInst<outs, ins, opcodestr, argstr, [], InstFormatOther> {

bits<5> vs2;

bits<5> vd;

let Inst{31-26} = funct6;

let Inst{25} = m.Value;

let Inst{24-20} = vs2;

let Inst{14-12} = funct3;

let Inst{11-7} = vd;

let Opcode = opcode.Value;

}

在rvv-mask-valid.s中

# CHECK-ASM-AND-OBJ: vor.vv v0, v1, v0, v0.t

# CHECK-ASM: encoding: [0x57,0x00,0x10,0x28]

vor.vv v0, v1, v0, v0.t

在rvv-valid.s中

# CHECK-ASM-AND-OBJ: vor.vv v0, v1, v0

# CHECK-ASM: encoding: [0x57,0x00,0x10,0x2a]

vor.vv v0, v1, v0

2.2 玄铁C910中的vor.vv指令

5.7.248 VOR.VV——矢量按位或指令

语法: vor.vv vd, vs2, vs1, vm

操作: vd[i]  vs2[i] | vs1[i]

执行权限: M mode/S mode/U mode

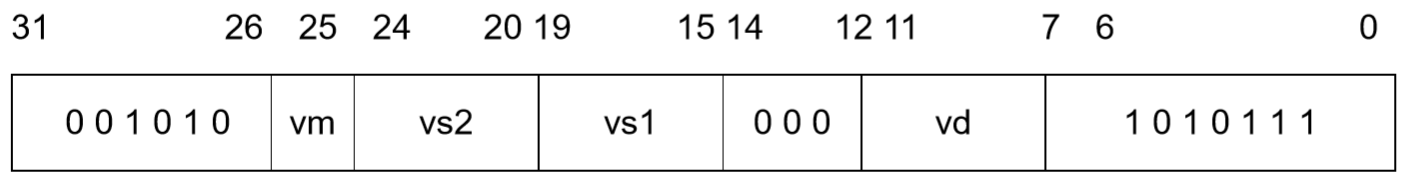
异常: 非法指令异常

影响标志位: 无

说明： 当 vm=1 时，指令未被 mask，对应汇编指令 vor.vv vd, vs2, vs1。

当 vm=0 时，指令被 mask，对应汇编指令 vor.vv vd, vs2, vs1, v0.t。

指令格式:



v0.t # enabled where v0[i].LSB=1, m=0

2.3 vor.vv指令对比结果

RVV中的vor.vv指令与玄铁C910的矢量指令集中的vor.vv（矢量按位或）指令进行对比发现，没区别。