RISC-V Vector Extension Intrinsic API Reference Manual

1. Preface

These builtins targets on rvv 0.9-draft-20200221 and trying to document rvv_intrinsics programming model.

2. Design Decisions and philosophy

Please see rvv-intrinsic-rfc.md

3. Known Issues

3.1 `ELEN` should be known in compiler time.

For example, the supported widening instructions are depend on `ELEN`, so if `ELEN` is 32, then the widening instruction with `SEW` 32 (result is 64bit) is not supported.

TBD: issue

3.2 There is no implementation and error handling when `XLEN` < `SEW`

1. vslide1up: If `XLEN` < `SEW`, the value is `sign-extended` to `SEW` bits. If `XLEN` > `SEW`, the least-significant bits are copied over and the high SEW-XLEN bits are ignored.

in RISCV32, `XLEN`=32, `SEW`=64 config, intrinsic interface are

```
vint64m1_t vslide1up_int64m1(vint64m1_t src, long scalar);
vuint64m1_t vslide1up_uint64m1(vuint64m1_t src, unsigned long scalar); // It's illegal.
// If users assign the value = 0xFFFF_FFFF, it will store 0xFFFF_FFFF_FFFF_FFFF into vd[vl-1
```

in RISCV64, `XLEN`=64, `SEW`=64 config, the above intrinsic functions are legal.

2. vmv.s.x The `vmv.s.x` instruction copies the scalar integer register to element 0 of the destination vector register. If `XLEN` < `SEW`, the value is `sign-extended` to SEW bits. If `XLEN` > `SEW`, the least-significant bits are copied and the upper XLEN-SEW bits are ignored.

in RISCV32, `XLEN`=32, `SEW`=64 config, intrinsic functions are

```
vint64m1_t vsplat_i64m1(long a);
vint64m1_t vsplat_i64m1(unsigned long b); // It's illegal
```

The main prolbem is rvv only provides the sign-extended for scalar, but if the input scalar is unsigned, we can not support it. (XLEN is 32, but SEW is 64 so 32-bit unsigned scalar need to sign-extended)

TBD: issue

4. None

Keep this chapter none to aligned to riscv-v-spec chapters

5. General Naming Rules

Please see rvv-intrinsic-rfc.md

6. Configuration-Setting and Utility Functions

Instructions

- vsetvli
- vsetvl

Set vl and vtype Functions

Intrinsics function list

Set vl to VLMAX with specific vtype

Intrinsic functions list

Read vl value

Intrinsic functions list

Reinterpret Cast Conversion Functions

Reinterpret the contents of a data as a different type, without changing any bits and generating any RVV instructions.

Intrinsic functions list

Vector Initialization Functions

Intrinsic functions list

Read/Write URW vector CSRs

```
enum RVV_CSR {
  RVV_VSTART = 0,
  RVV_VXSAT,
  RVV_VXRM,
  RVV_VCSR,
};

unsigned long vread_csr(enum RVV_CSR csr);
void vwrite_csr(enum RVV_CSR csr, unsigned long value);
```

7. Vector Loads and Stores

7.4. Vector Unit-Stride Operations

Instructions

- vlw.v
- vsw.v
- vlwu.v
- vlh.v
- vlhu.v
- vsh.v
- vlb.v
- vlbu.v
- vsb.v
- vle.v
- vse.v

Intrinsic functions list

Notes

- Fixed-sized vector loads can optionally sign or zero-extend their memory element into the destination register element if the register element is wider than the memory element.
- A fixed-size vector load raises an illegal instruction exception if the destination register element is narrower than the memory element.

7.5. Vector Strided Load/Store Operations

Instructions

- vlsw.v
- vssw.v
- vlswu.v
- vlsh.v
- vlshu.v
- vssh.v
- vlsb.v
- vlsbu.v
- vssb.v
- vlse.v
- vsse.v

Intrinsic functions list

7.6. Vector Indexed Load/Store Operations

Instructions

- vlxw.v
- vsxw.v
- vlxwu.v
- vlxh.v
- vlxhu.v
- vsxh.v
- vsuxh.v
- vlxb.v
- vlxbu.v
- vsxb.v
- vsuxb.v
- vlxe.v
- vsxe.v
- vsuxe.v

Intrinsic functions list

7.7. Unit-stride Fault-Only-First Loads Operations

Instructions

- vlbff.v

- vlhff.v
- vlwff.v
- vlbuff.v
- vlhuff.v
- vlwuff.v
- vleff.v

Intrinsic functions list

Notes

- Execute a regular load and stop load operation if there is 0 value of element, and then set vl equals the index of last non-zero value.

7.8. Vector Load/Store Segment Operations (Zvlsseg)

7.8.1. Vector Unit-Stride Segment Loads and Stores

Intrinsic functions list

TODO

7.8.2. Vector Strided Segment Loads and Stores

Intrinsic functions list

TODO

7.8.3. Vector Indexed Segment Loads and Stores

Intrinsic functions list

TODO

8. Vector AMO Operations (Zvamo)

- Vector AMOs
- If `SEW` is greater than `XLEN`, an illegal instruction exception is raised.

Instructions

- vamoswap{w,e}.v
- vamoadd{w,e}.v
- vamoxor{w,e}.v
- vamoand{w,e}.v
- vamoor{w,e}.v
- vamomin{w,e}.v
- vamomax{w,e}.v
- vamominu{w,e}.v
- vamomaxu{w,e}.v
- vamomaxu{w,e}.v

Intrinsic functions list

9. None

Keep this chapter none to aligned to riscv-v-spec chapters

10. None

Keep this chapter none to aligned to riscv-v-spec chapters

11. None

Keep this chapter none to aligned to riscv-v-spec chapters

12. Vector Integer Arithmetic Operations

12.1. Vector Single-Width Integer Add and Subtract

Instructions

- vadd.{vv,vx,vi}
- vsub.{vv,vx}
- vrsub.{vx,vi}

Intrinsic functions list

12.2. Vector Widening Integer Add/Subtract Operations

Instructions

- vwaddu.{vv,vx,wv,wx}
- vwsubu.{vv,vx,wv,wx}
- vwadd.{vv,vx,wv,wx}
- vwsub.{vv,vx,wv,wx}

Intrinsic functions list

12.3. Vector Integer Add-with-Carry / Subtract-with-Borrow Operations

Instructions

- vadc.{vvm,vxm,vim}
- vmadc.{vvm,vxm,vim}
- vsbc.{vvm,vxm}
- vmsbc.{vvm,vxm}

Intrinsic functions list

12.4. Vector Bitwise Logical Operations

Instructions

- vand.{vv,vx,vi}
- vxor.{vv,vx,vi}
- vor.{vv,vx,vi}

Intrinsic functions list

Note

- With an immediate of -1, scalar-immediate forms of the vxor instruction provide a bitwise NOT operation. This can be provided as an assembler pseudoinstruction vnot.v.

12.5. Vector Single-Width Bit Shift Operations

Instructions

- vsll.{vv,vx,vi}
- vsrl.{vv,vx,vi}
- vsra.{vv,vx,vi}

Intrinsic functions list

Notes

- A full complement of vector shift instructions are provided, including logical shift left, and logical (zero-extending) and arithmetic (sign-extending) shift right.

12.6. Vector Narrowing Integer Right Shift Operations

Instructions

- vnsra.{vv,vx,vi}
- vnsrl.{vv,vx,vi}

Intrinsic functions list

12.7. Vector Integer Comparison Operations

Instructions

- vmseq.{vv,vx,vi}
- vmsne.{vv,vx,vi}
- vmsltu.{vv,vx}
- vmslt.{vv,vx}
- vmsleu.{vv,vx,vi}
- vmsle.{vv,vx,vi}
- vmsgtu.{vx,vi}
- vmsgt.{vx,vi}

Intrinsic functions list

12.8. Vector Integer Min/Max Operations

Instructions

- vminu_{vv,vx}
- vmin_{vv,vx}
- vmaxu_{vv,vx}
- vmax_{vv,vx}

Intrinsic functions list

12.9. Vector Single-Width Integer Multiply Operations

Instructions

- vmul_{vv,vx}
- vmulh_{vv,vx}
- vmulhu_{vv,vx}
- vmulhsu_{vv,vx}

Intrinsic functions list

12.10. Vector Integer Divide Operations

Instructions

- vdivu.{vv,vx}
- vdiv.{vv,vx}
- vremu.{vv,vx}
- vrem.{vv,vx}

Intrinsic functions list

12.11. Vector Widening Integer Multiply Operations

Instructions

- vwmul.{vv,vx}
- vwmulu.{vv,vx}
- vwmulsu.{vv,vx}

Intrinsic functions list

12.12. Vector Single-Width Integer Multiply-Add Operations

Instructions

- vmacc_{vv,vx}
- vnmsac_{vv,vx}
- vmadd_{vv,vx}
- vnmsub_{vv,vx}

Intrinsic functions list

12.13. Vector Widening Integer Multiply-Add Operations

Instructions

- vwmaccu.{vv,vx}
- vwmacc.{vv,vx}
- vwmaccsu.{vv,vx}
- vwmaccus.{vv,vx}

Intrinsic functions list

12.14. Vector Quad-Widening Integer Multiply-Add Operations (Extension Zvqmac)

Instuctions

- vqmaccu.{vv,vx}
- vqmacc.{vv,vx}
- vqmaccsu.{vv,vx}
- vqmaccus.vx

Intrinsic functions list

12.15. Vector Integer Merge Operations

Instructions

- vmerge.{vvm,vxm,vim}

Intrinsic functions list

12.16. Vector Integer Move Operations

Instructions

- vmv.v.v
- vmv.v.x
- vmv.v.i

Intrinsic functions list

13. Vector Fixed-Point Arithmetic Operations

13.1. Vector Single-Width Saturating Add and Subtract

Instructions

- vsaddu.{vv,vx,vi}
- vsadd.{vv,vx,vi}
- vssubu.{vv,vx}
- vssub.{vv,vx}

Intrinsic functions list

13.2. Vector Single-Width Averaging Add and Subtract

Instructions

- vaadd.{vv,vx,vi}
- vasub.{vv,vx}

Intrinsic functions list

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

Instructions

- vsmul.{vv,vx}

Intrinsic functions list

13.4. Vector Single-Width Scaling Shift Operations

Instructions

- vssrl.{vv,vx,vi}
- vssra.{vv,vx,vi}

Intrinsic functions list

13.5. Vector Narrowing Fixed-Point Clip Operations

Instructions

- vnclipu.{wx,wv,wi}
- vnclip.{wx,wv,wi}

Intrinsic functions list

14. Vector Floating-Point Operations

14.2. Vector Single-Width Floating-Point Add/Subtract Operations

Instructions

- vfadd.{vv,vf}
- vfsub.{vv,vf}
- vfrsub.vf

Intrinsic functions list

14.3. Vector Widening Floating-Point Add/Subtract Operations

Instructions

- vfwadd.{vv,vf,wv,wf}
- vfwsub.{vv,vf,wv,wf}

Intrinsic functions list

14.4. Vector Single-Width Floating-Point Multiply/Divide Operations

Instructions

- vfmul.{vv,vf}
- vfdiv.{vv,vf}
- vfrdiv.{vv,vf}

Intrinsic functions list

14.5. Vector Widening Floating-Point Multiply Operations

Instructions

- vfwmul.{vv,vf}

Intrinsic functions list

14.6. Vector Single-Width Floating-Point Fused Multiply-Add Operations

Instructions

- vfmacc.{vv,vf}
- vfnmacc.{vv,vf}
- vfmsac.{vv,vf}
- vfnmsac.{vv,vf}
- vfmadd.{vv,vf}
- vfnmadd.{vv,vf}
- vfmsub.{vv,vf}
- vfnmsub.{vv,vf}

Intrinsic functions list

14.7. Vector Widening Floating-Point Fused Multiply-Add Operations

Instructions

- vfwmacc.{vv,vf}
- vfwnmacc.{vv,vf}
- vfwmsac.{vv,vf}
- vfwnmsac.{vv,vf}

Intrinsic functions list

14.8. Vector Floating-Point Square-Root Operations

Instructions

- vfsqrt.v

Intrinsic functions list

14.9. Vector Floating-Point MIN/MAX Operations

- vfmin.{vv,vf}
- vfmax.{vv,vf}

Intrinsic functions list

14.10. Vector Floating-Point Sign-Injection Operations

Instructions

- vfsgnj.{vv,vf}
- vfsgnjn.{vv,vf}
- vfsgnjx.{vv,vf}

Intrinsic functions list

14.11. Vector Floating-Point Compare Operations

Instructions

- vmfeq.{vv,vf}
- vmfne.{vv,vf}
- vmflt.{vv,vf}
- vmfle.{vv,vf}
- vmfgt.vf
- vmfge.vf

Intrinsic functions list

14.12. Vector Floating-Point Classify Operations

Instructions

- vfclass.v

Intrinsic functions list

14.13. Vector Floating-Point Merge Operations

Instructions

- vfmerge.vfm

Intrinsic functions list

14.14. Vector Floating-Point Move Operations

Instructions

- vfmv.v.f

Intrinsic functions list

14.15. Single-Width Floating-Point/Integer Type-Convert Operations

Instructions

- vfcvt.xu.f.v
- vfcvt.x.f.v
- vfcvt.f.xu.v
- vfcvt.f.x.v

Intrinsic functions list

14.16. Widening Floating-Point/Integer Type-Convert Operations

Instructions

- vfwcvt.xu.f.v
- vfwcvt.x.f.v
- vfwcvt.f.xu.v

- vfwcvt.f.x.v
- vfwcvt.f.f.v

Intrinsic functions list

14.17. Narrowing Floating-Point/Integer Type-Convert Operations

Instructions

- vfncvt.xu.f.w
- vfncvt.x.f.w
- vfncvt.f.xu.w
- vfncvt.f.x.w
- vfncvt.f.f.w
- vfncvt.rod.f.f.w

Intrinsic functions list

15. Vector Reduction Operations

15.1. Vector Single-Width Integer Reduction Operations

Instructions

- vredsum.vs
- vredmaxu.vs
- vredmax.vs
- vredminu.vs
- vredmin.vs
- vredand.vs
- vredor.vs
- vredxor.vs

Intrinsic functions list

15.2. Vector Widening Integer Reduction Operations

Instructions

- vwredsumu.vs
- vwredsum.vs

Intrinsic functions list

15.3. Vector Single-Width Floating-Point Reduction Operations

Instructions

- vfredosum.vs
- vfredsum.vs
- vfredmax.vs
- vfredmin.vs

Intrinsic functions list

15.4. Vector Widening Floating-Point Reduction Operations

Instructions

- vfwredosum.vs
- vfwredsum.vs

Intrinsic functions list

16. Vector Mask Instructions

16.1. Vector Mask-Register Logical Operations

Instructions

- vmand.mm
- vmnand.mm
- vmandnot.mm
- vmxor.mm
- vmor.mm
- vmnor.mm
- vmornot.mm
- vmxnor.mm

Intrinsic functions list

16.2. Vector mask population count vpopc

Instructions

- vpopc.m

Intrinsic functions list

16.3. vfirst find-first-set mask bit

Instructions

- vfirst.m

Intrinsic functions list

16.4. vmsbf.m set-before-first mask bit

Instructions

- vmsbf.m

Intrinsic functions list

16.5. vmsif.m set-including-first mask bit

Instructions

- vmsif.m

Intrinsic functions list

16.6. vmsof.m set-only-first mask bit

Instructions

- vmsof.m

Intrinsic functions list

16.8. Vector Iota Operations

Instructions

- viota.m

Intrinsic functions list

16.9. Vector Element Index Operations

Instructions

- vid.v

Intrinsic functions list

17. Vector Permutation Operations

17.1. Integer Scalar Move Operations

Instructions

- vmv.s.x
- vmv.x.s

Intrinsic functions list

17.2. Floating-Point Scalar Move Operations

Instructions

- vfmv.f.s
- vfmv.s.f

Intrinsic functions list

17.3. Vector Slide Operations

Instructions

- vslideup.{vx,vi}
- vslidedown.{vx,vi}

- vslide1up.vx
- vslide1down.vx

Intrinsic functions list

17.4. Vector Register Gather Operations

Instructions

- vrgather.{vx,vi}

Intrinsic functions list

17.5. Vector Compress Operations

Instructions

- vcompress.vm

Intrinsic functions list

18. None

Keep this chapter none to aligned to riscv-v-spec chapters

19. Divided Element Extension ('Zvediv')

19.3. Vector Integer Dot-Product Operations

Instructions

- vdotu.vv
- vdot.vv

Intrinsic functions list

TODO

19.4. Vector Floating-Point Dot Product Operations

Instructions

- vfdotu.vv

Intrinsic functions list

TODO

20. RVV Intrinsic Examples

- sgemm
- saxpy