⚙ [RISCV] Assemble/Disassemble v-ext instructions.

Actions

</> Needs Review Public

Authored by **HsiangKai** on Nov 7 2019, 10:43 PM.

Tags

LLVM

clang

Subscribers

cfe-commits, evandro, Joy12138 and 28 others

Details

Reviewers

- asb
- rogfer01
- rkruppe
- kito-cheng
- khchen
- evandro

SUMMARY

Assemble/disassemble RISC-V V extension instructions according to version 0.8-draft-20191004 in https://github.com/riscv/riscv-v-spec/.

I have tested this patch using GNU toolchain. The encoding is aligned to GNU assembler output. In this patch, there is a test case for each instruction at least.

The V register definition is just for assemble/disassemble. Its type is not important in this stage. I think it will be reviewed and modified as we want to do codegen for scalable vector types.

This patch does not include Zvamo, Zvlsseg, and Zvediv.

Diff Detail

Repository

rG LLVM Github Monorepo

Build Status

- **Buildable 53851**
- Build 61570: pre-merge checks

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

hanges, so older changes are hidden. Show Older Changes

evandro added a comment. Feb 4 2020, 4:50 PM

Shouldn't most instructions have the `vtype` register in `Uses`? Shouldn't most FP instructions have `fflags` in `Defs`?

The tests are extensive, but it would be important to also add negative tests.

llvm/lib/Target/RISCV/AsmParser/RISCVAsmParser.cpp

158	Please, rename to `parseVTypeI()`.
208	Please, rename to `defaultMaskRegOp()`.
755	This should be the `default` case for the `switch` above.
769	Ditto.
851	Avoid calling `logBase2()` multiple times.
852	Ditto.
1163	Please, rename to `Match_InvalidVTypeI`.
1168	Please, rename to `Match_InvalidVMaskRegister`.
1522	Please, rename to `parseVTypeI()`.
1565	Please, rename to `parseMaskReg()`.

llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp

165	Please, rename to `decodeVMaskReg()`.

llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.cpp

164	Please, rename to `printVMaskReg()`.

llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.h

46	Please, rename to `printVMaskReg()`.

llvm/lib/Target/RISCV/MCTargetDesc/RISCVMCCodeEmitter.cpp

84	Please, rename to `getVMaskReg()`.

llvm/lib/Target/RISCV/RISCV.td

152	Should it also imply `FeatureStdExtD`?

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

33	Please, rename to `VRegAsmOperand`.
40	Please, rename to `VRegOp`.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

smOperand.

p.

78	No need to set fields to the default `0`.
535	Shouldn't these have `vxsat` in `Defs`?
541	Shouldn't these have `vxrm` in `Uses`?
547	Shouldn't this hace `vxrm` in `Uses` and `vxsat` in `Defs`?
550	Aren't these part of the base V extension?
556	Shouldn't this hace `vxrm` in `Uses`?
560	Shouldn't this hace `vxrm` in `Uses` and `vxsat` in `Defs`?

HsiangKai updated this revision to **Diff 242816**. Edited · Feb 6 2020, 12:17 AM

HsiangKai marked 23 inline comments as done.

Address **@fpallares** and **@evandro**'s comments.

About `fflags` in FP instructions, it is also related to RISCVInstrInfoF.td and RISCVInstrInfoD.td. So, I think we could prepare another patch for it.

HsiangKai added inline comments. Feb 6 2020, 12:18 AM

llvm/lib/Target/RISCV/RISCV.td

152	I didn't find that V will imply D extension. Could you indicate the description in spec about this?

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

78	There is no default value for hasSideEffects, mayLoad, and mayStore.
535	It should have no impact on CodeGen. It seems no instructions to use vxsat.

HsiangKai updated this revision to **Diff 242905**. Feb 6 2020, 7:55 AM

Add the predicate for extension Zvqmac.

simoncook mentioned this in ~~**D73891: [RISCV] Support experimental/unratified extensions**~~.
Feb 7 2020, 2:44 AM

fpallares added a comment. Feb 14 2020, 8:46 AM

Thanks for addressing my comments **@HsiangKai**. After a closer look I've added some more inline comments, and some general ones too:

ove instructions `vmv<nf>r.v`.
that are missing the register overlap constraints (see inline
`smParser::validateInstruction`, as well as new tests.

- I've noticed you are using `${vm}` for describing the mask arg on most instructions, but AFAICT `$vm` is equally valid (e.g. You could write `$vd, $vs2$vm` instead of `$vd, $vs2${vm}`).
- We should probably recognize the assembler pseudo instructions defined in the ISA. Most can be implemented via `InstAlias`. I suggest to do this in another patch.
- Instruction from the `Zvqmac` extension instructions were added, should we add instructions for the other extensions as well? Otherwise we might want to restrict this patch to the base only.

llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp

154 Following **@evandro**'s naming suggestions, this should probably be renamed to `DecodeVRegisterClass`.

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

40 Whitespace missing before the colon.

55 Ditto.

107 I don't see why setting `hasSideEffects` is necessary and setting `mayStore` is not enough, could you please elaborate a bit on this?

426 Whitespace missing before the colon.

438 I'd say those instructions need the `@earlyclobber $vd` restriction too. Not because of the first source operand (since it has the same width as the destination), but because of the second (in the case of `VWOP_WV`), or the mask (for `VWOP_WX`).

Quoting section 11.2. Widening Vector Arithmetic Instructions:

> *The destination vector register group cannot overlap a source vector register group of a different element width (including the mask register if masked), otherwise an illegal instruction exception is raised.*

This has the downside that the `earlyclobber` constraint is too coarse and will impose unnecessary restrictions by not allowing the destination to overlap with the first (wide) operand, even when the instruction is used unmasked. Maybe we can use the `earlyclobber` for now, and find a better solution for this in a later patch.

460 In section 11.3. Narrowing Vector Arithmetic Instructions we find the following paragraph:

> *The destination vector register group cannot overlap the first source vector register group (specified by vs2). The destination vector register group cannot overlap the mask register if used, unless LMUL=1. If either constraint is violated, an illegal instruction exception is raised.*

From my understanding, this applies to `VNSRL` and `VNSRA`. In that case we need an `@earlyclobber` constraint here.

480 Whitespace missing before the colon.

on narrowing instructions apply to `VNCLIP` and `VNCLIPU` too.

ine comment above (for `VWADD_W` and such), there is an overlap restriction for these instructions too.

584 Whitespace missing before the colon.

661 I believe we are also missing the `@earlyclobber $vd` overlap constraints for these narrowing conversions.

From section 14.17. Narrowing Floating-Point/Integer Type-Convert Instructions:

> *These instructions have the same constraints on vector register overlap as other narrowing instructions (see Narrowing Vector Arithmetic Instructions).*

675 If I'm not mistaken, those instructions should have the `@earlyclobber $vd` constraint too to avoid overlaps.

685 Ditto.

llvm/lib/Target/RISCV/RISCVRegisterInfo.td

50 My understanding is that here you specify minimum sizes (at least we will have one element of size `ELEN`). As they stand, they are consistent with `ELEN=64`. So far it is unclear to me what sizes we would want to use if we also want to support `ELEN=32`. I imagine we could reuse `HwMode` or something similar.

As far as the offsets are concerned I'd suggest setting them to `-1` because we don't really know the offset where the `hi` part starts.

That said, you already mentioned that types are not that important in this patch.

llvm/test/MC/RISCV/rvv/macc.s

2 Looks like `+v` attribute gets passed twice. I'm also not sure we need `+a` and `+c`.

evandro added inline comments. Feb 18 2020, 9:22 AM

llvm/lib/Target/RISCV/RISCV.td

152 Indeed it doesn't:

> *If the base scalar ISA does not include floating-point, then a `fcsr` register is also added to hold mirrors of the `vxsat` and `vxrm` CSRs as explained below.*

Which means that the instructions that take FP scalars should also depend on `hasStdExt{F|D}`. Likewise when the scalar is a 64 bit integer, `hasRV64`.

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

78 Right. It would be cleaner if they were moved to the respective base instruction class then. Just a suggestion.

107 ecessary, since `mayStore` is indeed enough. The purpose of atch all when the rest of the instruction attributes miss some side

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

535 For the sake of completeness.

rogfer01 added inline comments. Feb 18 2020, 1:00 PM

llvm/lib/Target/RISCV/RISCV.td

152 I don't think we have to do anything special with 64 bit: section 11.1 states what is the behaviour when `XLEN` is different than `SEW`.

evandro added inline comments. Feb 18 2020, 1:24 PM

llvm/lib/Target/RISCV/RISCV.td

152 Good point.

HsiangKai marked 3 inline comments as done. Feb 19 2020, 12:09 AM

HsiangKai added inline comments.

llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp

154 The function name will be expected by TableGen as `"Decode" + Record->getName().str() + "RegisterClass"`. It seems related to the name of register class. Currently, the name of RegisterClass for V-extension registers is "VR".

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

107 There is no default value for hasSideEffects, mayLoad, and mayStore in `Instruction` class. So, I specify these three values for V instructions. It follows the style in RISCVInstrInfo.td. I agree that we could give these attributes some default values in some base class for V instructions. I will try to refactor it.

460 In current implementation for MC layer, there is no LMUL information in these instructions. So, if LMUL=1, there is no such restriction for them. I have a downstream patch to extend these instruction definitions for LMUL > 1. Maybe we could keep this MC implementation for LMUL=1 and extend it later. I will add comments for it.

HsiangKai updated this revision to **Diff 245339**. Feb 19 2020, 12:19 AM

Address some of @fpallares ' comments.

HsiangKai updated this revision to **Diff 245341**. Feb 19 2020, 12:37 AM

Remove Zvqmac extension. Keep this patch for base only.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

2020, 12:47 AM

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

460

Perhaps the wording in the spec is unclear here?

Looks like the following sentence does apply regardless of LMUL.

> > *The destination vector register group cannot overlap the first source vector register group (specified by vs2).*

While the sentence that follows it is clear that applies only to LMUL>1

> > *The destination vector register group cannot overlap the mask register if used, unless LMUL=1*

You also said that

> *I have a downstream patch to extend these instruction definitions for LMUL > 1*

I'm curious here: did you use a Pseudo for that or somehow avoided the issue that they'd be encoded in the same way?

Thanks!

fpallares marked an inline comment as done. Feb 19 2020, 2:23 AM

fpallares added inline comments.

llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp

154

I see, sorry for the noise.

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

107

I was only referring tot he fact that `hasSideEffects` here changed from `0` to `1` in a recent update of the patch. I don't really think a refactor is necessary here.

HsiangKai marked 2 inline comments as done. Feb 19 2020, 7:33 PM

HsiangKai added inline comments.

llvm/lib/Target/RISCV/RISCV.td

152

Yes, it is a problem about accessing fcsr if F extension is turned off. However, I found that there is vcsr in the latest version.

https://github.com/riscv/riscv-v-spec/commit/b25b643b9c29b4fde570293c64ca682a99a09f2a

So, I think we could keep V and F as separate unrelated extensions.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

ctor floating point operations.

td

460

I misunderstood the statements. Sorry for that. I will add earlyclobber for narrowing instructions. Thanks a lot.

About instruction definitions for LMUL > 1, I enable 'isCodeGenOnly' for these instructions to avoid encoding checking.

HsiangKai updated this revision to **Diff 245568**. Feb 19 2020, 7:38 PM

- Add predicates for vector floating-point instructions.
- Add earlyclobber for narrowing instructions.

HsiangKai updated this revision to **Diff 245571**. Feb 19 2020, 8:40 PM

HsiangKai updated this revision to **Diff 245577**. Feb 19 2020, 10:01 PM

Update test cases.

HsiangKai updated this revision to **Diff 245579**. Feb 19 2020, 10:18 PM

fpallares added inline comments. Mar 6 2020, 2:20 AM

llvm/test/MC/RISCV/rvv/invalid.s

6

Suggestion: If you use `CHECK-LABEL` between the regular `CHECK` you can avoid these directives to succeed by matching lines corresponding to a later instruction. Like this:

```
vsetvli a2, a0, e31
// CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]
// CHECK-ERROR-LABEL: vsetvli a2, a0, e31
```

(Note the `CHECK-LABEL` is placed *after* the `CHECK` since it appears in this order in the output)

Without the `CHECK-LABELS` the test will still fail when something is wrong, but with them FileCheck will be able to point exactly where the problem occurred.

More info on this in the corresponding section of the FileCheck manual.

HsiangKai marked an inline comment as done. Mar 10 2020, 8:43 AM

HsiangKai added inline comments.

llvm/test/MC/RISCV/rvv/invalid.s

6

Got it. Thanks.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

49399. Mar 10 2020, 8:45 AM

In RISCVRegisterInfo.td, set offset of vector sub-registers to -1.

Harbormaster failed remote builds in **B48696: Diff 249399**! Mar 10 2020, 9:46 AM

HsiangKai updated this revision to **Diff 249833**. Mar 11 2020, 10:18 PM

- Rewrite validateInstruction() in RISCVAsmParser.cpp. Use TSFlags to classify instructions' constraints.
- Update invalid.s test cases.

Harbormaster failed remote builds in **B48928: Diff 249833**! Mar 11 2020, 11:18 PM

HsiangKai updated this revision to **Diff 254501**. Thu, Apr 2, 6:27 AM

- Rebase on master branch.
- Fix validInstruction() bugs.
- Update test cases.

Harbormaster failed remote builds in **B51467: Diff 254501**! Thu, Apr 2, 7:01 AM

HsiangKai updated this revision to **Diff 254667**. Thu, Apr 2, 6:23 PM

- Fix clang-format errors.
- Rename variables according to LLVM Coding Standards.

Harbormaster failed remote builds in **B51573: Diff 254667**! Thu, Apr 2, 7:00 PM

HsiangKai updated this revision to **Diff 258526**. Sat, Apr 18, 8:48 AM

- Rebase master.
- Make "V" extension as an experimental one.
  - This follows the design as discussed on the mailing lists in the following RFC: http://lists.llvm.org/pipermail/llvm-dev/2020-January/138364.html

Herald added a project: **clang**. · View Herald Transcript Sat, Apr 18, 8:48 AM

Herald added a subscriber: **cfe-commits**. · View Herald Transcript

Harbormaster failed remote builds in **B53851: Diff 258526**! Sat, Apr 18, 9:09 AM

Revision Contents Changeset List

Files | History | Commits | Similar

Path **Packages**

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

V.cpp (5 lines)

M llvm/lib/Target/RISCV/AsmParser/RISCVAsmParser.cpp (258 lines)

M llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp (36 lines)

M llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.h (4 lines)

M llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.cpp (24 lines)

M llvm/lib/Target/RISCV/MCTargetDesc/RISCVMCCodeEmitter.cpp (20 lines)

M llvm/lib/Target/RISCV/RISCV.td (7 lines)

M llvm/lib/Target/RISCV/RISCVInstrFormats.td (19 lines)

A M llvm/lib/Target/RISCV/RISCVInstrFormatsV.td (300 lines)

M llvm/lib/Target/RISCV/RISCVInstrInfo.h (22 lines)

M llvm/lib/Target/RISCV/RISCVInstrInfo.td (1 line)

A M llvm/lib/Target/RISCV/RISCVInstrInfoV.td (820 lines)

M llvm/lib/Target/RISCV/RISCVRegisterInfo.td (104 lines)

M llvm/lib/Target/RISCV/RISCVSchedRocket32.td (1 line)

M llvm/lib/Target/RISCV/RISCVSchedRocket64.td (1 line)

M llvm/lib/Target/RISCV/RISCVSubtarget.h (2 lines)

M llvm/lib/Target/RISCV/RISCVSystemOperands.td (10 lines)

M llvm/lib/Target/RISCV/Utils/RISCVBaseInfo.h (2 lines)

A M llvm/test/MC/RISCV/rvv/add.s (315 lines)

A M llvm/test/MC/RISCV/rvv/and.s (45 lines)

A M llvm/test/MC/RISCV/rvv/clip.s (81 lines)

A M llvm/test/MC/RISCV/rvv/compare.s (249 lines)

A M llvm/test/MC/RISCV/rvv/convert.s (189 lines)

A M llvm/test/MC/RISCV/rvv/div.s (105 lines)

A M llvm/test/MC/RISCV/rvv/fadd.s (81 lines)

A M llvm/test/MC/RISCV/rvv/fcompare.s (129 lines)

A M llvm/test/MC/RISCV/rvv/fdiv.s (45 lines)

A M llvm/test/MC/RISCV/rvv/fmacc.s (297 lines)

A M llvm/test/MC/RISCV/rvv/fminmax.s (57 lines)

A M llvm/test/MC/RISCV/rvv/fmul.s (57 lines)

s)

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

ines)

A M llvm/test/MC/RISCV/rvv/freduction.s (81 lines)

A M llvm/test/MC/RISCV/rvv/fsub.s (93 lines)

A M llvm/test/MC/RISCV/rvv/invalid.s (550 lines)

A M llvm/test/MC/RISCV/rvv/load.s (339 lines)

A M llvm/test/MC/RISCV/rvv/macc.s (189 lines)

A M llvm/test/MC/RISCV/rvv/mask.s (141 lines)

A M llvm/test/MC/RISCV/rvv/minmax.s (105 lines)

A M llvm/test/MC/RISCV/rvv/mul.s (201 lines)

A M llvm/test/MC/RISCV/rvv/mv.s (63 lines)

A M llvm/test/MC/RISCV/rvv/or.s (45 lines)

A M llvm/test/MC/RISCV/rvv/others.s (141 lines)

A M llvm/test/MC/RISCV/rvv/reduction.s (129 lines)

A M llvm/test/MC/RISCV/rvv/shift.s (261 lines)

A M llvm/test/MC/RISCV/rvv/sign-injection.s (81 lines)

A M llvm/test/MC/RISCV/rvv/snippet.s (32 lines)

A M llvm/test/MC/RISCV/rvv/store.s (207 lines)

A M llvm/test/MC/RISCV/rvv/sub.s (285 lines)

A M llvm/test/MC/RISCV/rvv/vsetvl.s (21 lines)

A M llvm/test/MC/RISCV/rvv/xor.s (45 lines)

Diff 258526

clang/lib/Driver/ToolChains/Arch/RISCV.cpp

View Options

Show First 20 Lines • Show All 56 Lines • ▼ Show 20 Line(s)

```
57  57  // If the extension is supported as experimental, return the version of that
58  58  // extension that the compiler currently supports.
59  59  static Optional<RISCVExtensionVersion>
60  60  isExperimentalExtension(StringRef Ext) {
61  61    if (Ext == "b" || Ext == "zbb" || Ext == "zbc" || Ext == "zbe" ||
62  62        Ext == "zbf" || Ext == "zbm" || Ext == "zbp" || Ext == "zbr" ||
63  63        Ext == "zbs" || Ext == "zbt" || Ext == "zbproposedc")
64  64      return RISCVExtensionVersion{"0", "92"};
    65    if (Ext == "v")
    66      return RISCVExtensionVersion{"0", "8"};
65  67    return None;
66  68  }
67  69
                                                  n(StringRef Ext) {
                                                  ns which are marked as experimental.
                                                  xt))
71  73      return true;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
72  74
```

▲ Show 20 Lines • Show All 321 Lines • ▼ Show 20 Line(s)

```
394 396       HasD = true;
395 397       break;
396 398     case 'c':
397 399       Features.push_back("+c");
398 400       break;
399 401     case 'b':
400 402       Features.push_back("+experimental-b");
401 403       break;
    404     case 'v':
    405       Features.push_back("+experimental-v");
    406       break;
402 407     }
403 408
404 409     // Consume full extension name and version, including any optional '_'
405 410     // between this extension and the next
406 411     ++I;
407 412     I += Major.size();
408 413     if (Minor.size())
409 414       I += Minor.size() + 1 /*'p'*/;
```

▲ Show 20 Lines • Show All 248 Lines • Show Last 20 Lines

clang/test/Driver/riscv-arch.c View Options

Show First 20 Lines • Show All 354 Lines • ▼ Show 20 Line(s)

```
355 355 // RUN: %clang -target riscv32-unknown-elf -march=rv32izbb0p92 -menable-experimental-extensio
        ns -### %s \
356 356 // RUN: -fsyntax-only 2>&1 | FileCheck -check-prefix=RV32-EXPERIMENTAL-ZBB %s
357 357 // RV32-EXPERIMENTAL-ZBB: "-target-feature" "+experimental-zbb"
358 358
359 359 // RUN: %clang -target riscv32-unknown-elf -march=rv32izbb0p92_zbp0p92 -menable-experimental-
        extensions -### %s \
360 360 // RUN: -fsyntax-only 2>&1 | FileCheck -check-prefix=RV32-EXPERIMENTAL-ZBB-ZBP %s
361 361 // RV32-EXPERIMENTAL-ZBB-ZBP: "-target-feature" "+experimental-zbb"
362 362 // RV32-EXPERIMENTAL-ZBB-ZBP: "-target-feature" "+experimental-zbp"
    363
    364 // RUN: %clang -target riscv32-unknown-elf -march=rv32iv -### %s \
    365 // RUN: -fsyntax-only 2>&1 | FileCheck -check-prefix=RV32-EXPERIMENTAL-V-NOFLAG %s
    366 // RV32-EXPERIMENTAL-V-NOFLAG: error: invalid arch name 'rv32iv'
    367 // RV32-EXPERIMENTAL-V-NOFLAG: requires '-menable-experimental-extensions'
    368
    369 // RUN: %clang -target riscv32-unknown-elf -march=rv32iv -menable-experimental-extensions -##
        # %s \
    370 // RUN: -fsyntax-only 2>&1 | FileCheck -check-prefix=RV32-EXPERIMENTAL-V-NOVERS %s
    371 // RV32-EXPERIMENTAL-V-NOVERS: error: invalid arch name 'rv32iv'
    372 // RV32-EXPERIMENTAL-V-NOVERS: experimental extension requires explicit version number
    373
    374 // RUN: %clang -target riscv32-unknown-elf -march=rv32iv0p1 -menable-experimental-extensions
        -### %s \
    375 // RUN: -fsyntax-only 2>&1 | FileCheck -check-prefix=RV32-EXPERIMENTAL-V-BADVERS %s
    376 // RV32-EXPERIMENTAL-V-BADVERS: error: invalid arch name 'rv32iv0p1'
    377 // RV32-EXPERIMENTAL-V-BADVERS: unsupported version number 0.1 for experimental extension
    378
                                                    unknown-elf -march=rv32iv0p8 -menable-experimental-extensions
                                                    leCheck -check-prefix=RV32-EXPERIMENTAL-V-GOODVERS %s
                                                    : "-target-feature" "+experimental-v"
```

llvm/lib/Target/RISCV/AsmParser/RISCVAsmParser.cpp

View Options

```
1   1  //===-- RISCVAsmParser.cpp - Parse RISCV assembly to MCInst instructions --===//
2   2  //
3   3  // Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
4   4  // See https://llvm.org/LICENSE.txt for license information.
5   5  // SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
6   6  //
7   7  //===----------------------------------------------------------------------===//
8   8
9   9  #include "MCTargetDesc/RISCVAsmBackend.h"
10  10 #include "MCTargetDesc/RISCVMCExpr.h"
11  11 #include "MCTargetDesc/RISCVMCTargetDesc.h"
12  12 #include "MCTargetDesc/RISCVTargetStreamer.h"
    13 #include "RISCVInstrInfo.h"
13  14 #include "TargetInfo/RISCVTargetInfo.h"
14  15 #include "Utils/RISCVBaseInfo.h"
15  16 #include "Utils/RISCVMatInt.h"
16  17 #include "llvm/ADT/STLExtras.h"
    18 #include "llvm/ADT/SmallBitVector.h"
17  19 #include "llvm/ADT/SmallVector.h"
18  20 #include "llvm/ADT/Statistic.h"
19  21 #include "llvm/ADT/StringSwitch.h"
20  22 #include "llvm/CodeGen/Register.h"
21  23 #include "llvm/MC/MCAssembler.h"
22  24 #include "llvm/MC/MCContext.h"
23  25 #include "llvm/MC/MCExpr.h"
24  26 #include "llvm/MC/MCInst.h"
```

▲ Show 20 Lines • Show All 98 Lines • ▼ Show 20 Line(s)

```
123 125                            MCStreamer &Out, bool HasTmpReg);
124 126
125 127   // Checks that a PseudoAddTPRel is using x4/tp in its second input operand.
126 128   // Enforcing this using a restricted register class for the second input
127 129   // operand of PseudoAddTPRel results in a poor diagnostic due to the fact
128 130   // 'add' is an overloaded mnemonic.
129 131   bool checkPseudoAddTPRel(MCInst &Inst, OperandVector &Operands);
130 132
    133   // Check instruction constraints.
    134   bool validateInstruction(MCInst &Inst, OperandVector &Operands);
    135
131 136   /// Helper for processing MC instructions that have been successfully matched
132 137   /// by MatchAndEmitInstruction. Modifications to the emitted instructions,
133 138   /// like the expansion of pseudo instructions (e.g., "li"), can be performed
134 139   /// in this method.
135 140   bool processInstruction(MCInst &Inst, SMLoc IDLoc, OperandVector &Operands,
136 141                           MCStreamer &Out);
137 142
138 143 // Auto-generated instruction matching functions
139 144 #define GET_ASSEMBLER_HEADER
140 145 #include "RISCVGenAsmMatcher.inc"
141 146
142 147   OperandMatchResultTy parseCSRSystemRegister(OperandVector &Operands);
143 148   OperandMatchResultTy parseImmediate(OperandVector &Operands);
144 149   OperandMatchResultTy parseRegister(OperandVector &Operands,
      150                                      bool AllowParens = false);
      151   OpBaseReg(OperandVector &Operands);
      152   micMemOp(OperandVector &Operands);
      153   randWithModifier(OperandVector &Operands);
149 154   OperandMatchResultTy parseBareSymbol(OperandVector &Operands);
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
150 155   OperandMatchResultTy parseCallSymbol(OperandVector &Operands);
151 156   OperandMatchResultTy parsePseudoJumpSymbol(OperandVector &Operands);
152 157   OperandMatchResultTy parseJALOffset(OperandVector &Operands);
    158   OperandMatchResultTy parseVTypeI(OperandVector &Operands);
```

evandro — ✔ Done

Please, rename to `parseVTypeI()`.

```
    159   OperandMatchResultTy parseMaskReg(OperandVector &Operands);
153 160
154 161   bool parseOperand(OperandVector &Operands, StringRef Mnemonic);
155 162
156 163   bool parseDirectiveOption();
157 164   bool parseDirectiveAttribute();
158 165
159 166   void setFeatureBits(uint64_t Feature, StringRef FeatureString) {
160 167     if (!(getSTI().getFeatureBits()[Feature])) {
```

Show All 32 Lines

```
193 200     copySTI().setFeatureBits(FeatureBits);
194 201     setAvailableFeatures(ComputeAvailableFeatures(FeatureBits));
195 202
196 203     ParserOptions = ParserOptionsStack.pop_back_val();
197 204
198 205     return false;
199 206   }
200 207
    208   std::unique_ptr<RISCVOperand> defaultMaskRegOp() const;
```

evandro — ✔ Done

Please, rename to `defaultMaskRegOp()`.

```
    209
201 210 public:
202 211   enum RISCVMatchResultTy {
203 212     Match_Dummy = FIRST_TARGET_MATCH_RESULT_TY,
204 213 #define GET_OPERAND_DIAGNOSTIC_TYPES
205 214 #include "RISCVGenAsmMatcher.inc"
206 215 #undef GET_OPERAND_DIAGNOSTIC_TYPES
207 216   };
208 217
```

Show All 31 Lines

```
240 249 /// RISCVOperand - Instances of this class represent a parsed machine
241 250 /// instruction
242 251 struct RISCVOperand : public MCParsedAsmOperand {
243 252
244 253   enum class KindTy {
245 254     Token,
246 255     Register,
247 256     Immediate,
248       SystemRegister
    257     SystemRegister,
    258     VType,
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
253 263   struct RegOp {
```

```
254 264     Register RegNum;
255 265   };
256 266
257 267   struct ImmOp {
258 268     const MCExpr *Val;
259 269   };
260 270
261 271   struct SysRegOp {
262 272     const char *Data;
263 273     unsigned Length;
264 274     unsigned Encoding;
265 275     // FIXME: Add the Encoding parsed fields as needed for checks,
266 276     // e.g.: read/write or user/supervisor/machine privileges.
267 277   };
268 278
    279   enum class VSEW {
    280     SEW_8 = 0,
    281     SEW_16,
    282     SEW_32,
    283     SEW_64,
    284     SEW_128,
    285     SEW_256,
    286     SEW_512,
    287     SEW_1024,
    288   };
    289
    290   enum class VLMUL { LMUL_1 = 0, LMUL_2, LMUL_4, LMUL_8 };
    291
    292   struct VTypeOp {
    293     VSEW Sew;
    294     VLMUL Lmul;
    295     unsigned Encoding;
    296   };
    297
```

rogfer01 — Not Done

You will have to replace this with another one `VType` otherwise clang (at least 7.0) will reject this code.

I suggest `VTypeOp`.

```
269 298   SMLoc StartLoc, EndLoc;
270 299   union {
271 300     StringRef Tok;
272 301     RegOp Reg;
273 302     ImmOp Imm;
274 303     struct SysRegOp SysReg;
    304     struct VTypeOp VType;
```

rogfer01 — Not Done

`VTypeOp VType;` should do here

```
275 305   };
276 306
277 307   RISCVOperand(KindTy K) : MCParsedAsmOperand(), Kind(K) {}
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
                                                    nd &o) : MCParsedAsmOperand() {
282 312     IsRV64 = o.IsRV64;
```

```
283 313     StartLoc = o.StartLoc;
284 314     EndLoc = o.EndLoc;
285 315     switch (Kind) {
286 316     case KindTy::Register:
287 317       Reg = o.Reg;
288 318       break;
289 319     case KindTy::Immediate:
290 320       Imm = o.Imm;
291 321       break;
292 322     case KindTy::Token:
293 323       Tok = o.Tok;
294 324       break;
295 325     case KindTy::SystemRegister:
296 326       SysReg = o.SysReg;
297 327       break;
    328     case KindTy::VType:
    329       VType = o.VType;
    330       break;
298 331     }
299 332   }
300 333
301 334   bool isToken() const override { return Kind == KindTy::Token; }
302 335   bool isReg() const override { return Kind == KindTy::Register; }
    336   bool isV0Reg() const {
    337     return Kind == KindTy::Register && Reg.RegNum == RISCV::V0;
    338   }
303 339   bool isImm() const override { return Kind == KindTy::Immediate; }
304 340   bool isMem() const override { return false; }
305 341   bool isSystemRegister() const { return Kind == KindTy::SystemRegister; }
    342   bool isVType() const { return Kind == KindTy::VType; }
306 343
307 344   bool isGPR() const {
308 345     return Kind == KindTy::Register &&
309 346            RISCVMCRegisterClasses[RISCV::GPRRegClassID].contains(Reg.RegNum);
310 347   }
311 348
312 349   static bool evaluateConstantImm(const MCExpr *Expr, int64_t &Imm,
313 350                                   RISCVMCExpr::VariantKind &VK) {
```

▲ Show 20 Lines • Show All 67 Lines • ▼ Show 20 Line(s)

```
381 418     if (!isImm() || evaluateConstantImm(getImm(), Imm, VK))
382 419       return false;
383 420     return RISCVAsmParser::classifySymbolRef(getImm(), VK, Imm) &&
384 421            VK == RISCVMCExpr::VK_RISCV_TPREL_ADD;
385 422   }
386 423
387 424   bool isCSRSystemRegister() const { return isSystemRegister(); }
388 425
    426   bool isVTypeI() const { return isVType(); }
    427
389 428   /// Return true if the operand is a valid for the fence instruction e.g.
390 429   /// ('iorw').
391 430   bool isFenceArg() const {
392 431     if (!isImm())
393 432       return false;
394 433     const MCExpr *Val = getImm();
395 434     auto *SVal = dyn_cast<MCSymbolRefExpr>(Val);
        )  != MCSymbolRefExpr::VK_None)
```


es • Show All 91 Lines • ▼ Show 20 Line(s)

```
        = RISCVMCExpr::VK_RISCV_None;
489 528     if (!isImm())
```

```
490 529       return false;
491 530     bool IsConstantImm = evaluateConstantImm(getImm(), Imm, VK);
492 531     return IsConstantImm && isUInt<5>(Imm) && (Imm != 0) &&
493 532            VK == RISCVMCExpr::VK_RISCV_None;
494 533   }
495 534
    535   bool isSImm5() const {
    536     if (!isImm())
    537       return false;
    538     RISCVMCExpr::VariantKind VK = RISCVMCExpr::VK_RISCV_None;
    539     int64_t Imm;
    540     bool IsConstantImm = evaluateConstantImm(getImm(), Imm, VK);
    541     return IsConstantImm && isInt<5>(Imm) && VK == RISCVMCExpr::VK_RISCV_None;
    542   }
    543
496 544   bool isSImm6() const {
497 545     if (!isImm())
498 546       return false;
499 547     RISCVMCExpr::VariantKind VK = RISCVMCExpr::VK_RISCV_None;
500 548     int64_t Imm;
501 549     bool IsConstantImm = evaluateConstantImm(getImm(), Imm, VK);
502 550     return IsConstantImm && isInt<6>(Imm) &&
503 551       VK == RISCVMCExpr::VK_RISCV_None;
```

▲ Show 20 Lines • Show All 176 Lines • ▼ Show 20 Line(s)

```
680 728     return Imm.Val;
681 729   }
682 730
683 731   StringRef getToken() const {
684 732     assert(Kind == KindTy::Token && "Invalid type access!");
685 733     return Tok;
686 734   }
687 735
    736   static StringRef getSEWStr(VSEW Sew) {
    737     switch (Sew) {
    738     default:
```

rogfer01 — Not Done

This label is unneeded because this switch is covering all of them.

As per the llvm coding standards you dont't want a default in this case.

Also the formatting is unusal, make sure you run clang-format to the new code you contribute.

```
    739       llvm_unreachable("SEW must be [8|16|32|64|128|256|512|1024]");
    740     case VSEW::SEW_8:
    741       return "e8";
    742     case VSEW::SEW_16:
    743       return "e16";
    744     case VSEW::SEW_32:
    745       return "e32";
    746     case VSEW::SEW_64:
    747       return "e64";
    748     case VSEW::SEW_128:
    749       return "e128";
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
    754     case VSEW::SEW_1024:
```

```
755       return "e1024";
```

evandro ✔ Done

This should be the `default` case for the `switch` above.

```
756     }
757   }
758
759   static StringRef getLMULStr(VLMUL Lmul) {
760     switch (Lmul) {
```

rogfer01 Not Done

Ditto.

```
761     default:
762       llvm_unreachable("LMUL must be [1|2|4|8]");
763     case VLMUL::LMUL_1:
764       return "m1";
765     case VLMUL::LMUL_2:
766       return "m2";
767     case VLMUL::LMUL_4:
768       return "m4";
769     case VLMUL::LMUL_8:
```

evandro ✔ Done

Ditto.

```
770       return "m8";
771     }
772   }
```

kito-cheng Not Done

Indention seems weird for getSEWStr and getLMULStr?

```
773
774   StringRef getVType(SmallVectorImpl<char> &Out) const {
775     assert(Kind == KindTy::VType && "Invalid access!");
776     Twine VTypeStr(getSEWStr(VType.Sew));
777     VTypeStr.concat(Twine(","));
778     VTypeStr.concat(Twine(getLMULStr(VType.Lmul)));
779
780     return VTypeStr.toStringRef(Out);
781   }
782
688 783   void print(raw_ostream &OS) const override {
689 784     switch (Kind) {
690 785     case KindTy::Immediate:
691 786       OS << *getImm();
692 787       break;
693 788     case KindTy::Register:
694 789       OS << "<register x";
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
699 794       break;
```

```
700 795     case KindTy::SystemRegister:
701 796       OS << "<sysreg: " << getSysReg() << '>';
702 797       break;
    798     case KindTy::VType:
    799       SmallVector<char, 8> VTypeBuf;
    800       OS << "<vtype: " << getVType(VTypeBuf) << '>';
    801       break;
703 802     }
704 803   }
705 804
706 805   static std::unique_ptr<RISCVOperand> createToken(StringRef Str, SMLoc S,
707 806                                                    bool IsRV64) {
708 807     auto Op = std::make_unique<RISCVOperand>(KindTy::Token);
709 808     Op->Tok = Str;
710 809     Op->StartLoc = S;
```

Show All 28 Lines

```
739 838     Op->SysReg.Data = Str.data();
740 839     Op->SysReg.Length = Str.size();
741 840     Op->SysReg.Encoding = Encoding;
742 841     Op->StartLoc = S;
743 842     Op->IsRV64 = IsRV64;
744 843     return Op;
745 844   }
746 845
    846   static std::unique_ptr<RISCVOperand> createVType(APInt Sew, APInt Lmul,
    847                                                    SMLoc S, bool IsRV64) {
    848     auto Op = std::make_unique<RISCVOperand>(KindTy::VType);
    849     Sew.ashrInPlace(3);
    850     unsigned SewLog2 = Sew.logBase2();
    851     unsigned LmulLog2 = Lmul.logBase2();
```

evandro ✔ Done

Avoid calling `logBase2()` multiple times.

```
    852     Op->VType.Sew = static_cast<VSEW>(SewLog2);
```

evandro ✔ Done

Ditto.

```
    853     Op->VType.Lmul = static_cast<VLMUL>(LmulLog2);
    854     Op->VType.Encoding = (SewLog2 << 2) | LmulLog2;
    855     Op->StartLoc = S;
    856     Op->IsRV64 = IsRV64;
    857     return Op;
    858   }
    859
747 860   void addExpr(MCInst &Inst, const MCExpr *Expr) const {
748 861     assert(Expr && "Expr shouldn't be null!");
749 862     int64_t Imm = 0;
750 863     RISCVMCExpr::VariantKind VK = RISCVMCExpr::VK_RISCV_None;
751 864     bool IsConstant = evaluateConstantImm(Expr, Imm, VK);
752 865
753 866     if (IsConstant)
      ::createImm(Imm));
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

Show All 31 Lines

```
      createImm(Imm));
787 900   }
```

```
788 901
789 902   void addCSRSystemRegisterOperands(MCInst &Inst, unsigned N) const {
790 903     assert(N == 1 && "Invalid number of operands!");
791 904     Inst.addOperand(MCOperand::createImm(SysReg.Encoding));
792 905   }
793 906
    907   void addVTypeIOperands(MCInst &Inst, unsigned N) const {
    908     assert(N == 1 && "Invalid number of operands!");
    909     Inst.addOperand(MCOperand::createImm(VType.Encoding));
    910   }
    911
794 912   // Returns the rounding mode represented by this RISCVOperand. Should only
795 913   // be called after checking isFRMArg.
796 914   RISCVFPRndMode::RoundingMode getRoundingMode() const {
797 915     // isFRMArg has validated the operand, meaning this cast is safe.
798 916     auto SE = cast<MCSymbolRefExpr>(getImm());
799 917     RISCVFPRndMode::RoundingMode FRM =
800 918         RISCVFPRndMode::stringToRoundingMode(SE->getSymbol().getName());
801 919     assert(FRM != RISCVFPRndMode::Invalid && "Invalid rounding mode");
```

▲ Show 20 Lines • Show All 61 Lines • ▼ Show 20 Line(s)

```
863 981
864 982   auto Result =
865 983     MatchInstructionImpl(Operands, Inst, ErrorInfo, MissingFeatures,
866 984                          MatchingInlineAsm);
867 985   switch (Result) {
868 986   default:
869 987     break;
870 988   case Match_Success:
    989     if (validateInstruction(Inst, Operands))
    990       return true;
871 991     return processInstruction(Inst, IDLoc, Operands, Out);
872 992   case Match_MissingFeature: {
873 993     assert(MissingFeatures.any() && "Unknown missing features!");
874 994     bool FirstFeature = true;
875 995     std::string Msg = "instruction requires the following:";
876 996     for (unsigned i = 0, e = MissingFeatures.size(); i != e; ++i) {
877 997       if (MissingFeatures[i]) {
878 998         Msg += FirstFeature ? " " : ", ";
```

▲ Show 20 Lines • Show All 156 Lines • ▼ Show 20 Line(s)

```
1035 1155   case Match_InvalidCallSymbol: {
1036 1156     SMLoc ErrorLoc = ((RISCVOperand &)*Operands[ErrorInfo]).getStartLoc();
1037 1157     return Error(ErrorLoc, "operand must be a bare symbol name");
1038 1158   }
1039 1159   case Match_InvalidTPRelAddSymbol: {
1040 1160     SMLoc ErrorLoc = ((RISCVOperand &)*Operands[ErrorInfo]).getStartLoc();
1041 1161     return Error(ErrorLoc, "operand must be a symbol with %tprel_add modifier");
1042 1162   }
     1163   case Match_InvalidVTypeI: {
```

evandro ✔ Done

Please, rename to `Match_InvalidVTypeI`.

```
     1164     SMLoc ErrorLoc = ((RISCVOperand &)*Operands[ErrorInfo]).getStartLoc();
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

evandro ✔ Done

Please, rename to `Match_InvalidVMaskRegister`.

```
1169      SMLoc ErrorLoc = ((RISCVOperand &)*Operands[ErrorInfo]).getStartLoc();
1170      return Error(ErrorLoc, "operand must be v0.t");
1171    }
1043 1172  }
1044 1173
1045 1174  llvm_unreachable("Unknown match type detected!");
1046 1175}
1047 1176
1048 1177// Attempts to match Name as a register (either using the default name or
1049 1178// alternative ABI names), setting RegNo to the matching register. Upon
1050 1179// failure, returns true and sets RegNo to 0. If IsRV32E then registers
```

▲ Show 20 Lines • Show All 333 Lines • ▼ Show 20 Line(s)

```
1384 1513  // is an identifier and is followed by a comma.
1385 1514  if (getLexer().is(AsmToken::Identifier) &&
1386 1515      getLexer().peekTok().is(AsmToken::Comma))
1387 1516    return MatchOperand_NoMatch;
1388 1517
1389 1518  return parseImmediate(Operands);
1390 1519}
1391 1520
     1521OperandMatchResultTy RISCVAsmParser::parseVTypeI(OperandVector &Operands) {
     1522  SMLoc S = getLoc();
```

evandro ✔ Done

Please, rename to `parseVTypeI()`.

```
     1523  if (getLexer().getKind() != AsmToken::Identifier)
     1524    return MatchOperand_NoMatch;
     1525
     1526  // Parse "e8,m1"
     1527  StringRef Name = getLexer().getTok().getIdentifier();
     1528  if (!Name.consume_front("e"))
     1529    return MatchOperand_NoMatch;
     1530  APInt Sew(16, Name, 10);
     1531  if (Sew != 8 && Sew != 16 && Sew != 32 && Sew != 64 && Sew != 128 &&
     1532      Sew != 256 && Sew != 512 && Sew != 1024)
     1533    return MatchOperand_NoMatch;
     1534  getLexer().Lex();
     1535
     1536  if (getLexer().getKind() == AsmToken::EndOfStatement) {
     1537    Operands.push_back(
     1538        RISCVOperand::createVType(Sew, APInt(16, 1), S, isRV64()));
     1539
     1540    return MatchOperand_Success;
     1541  }
     1542
     1543  if (!getLexer().is(AsmToken::Comma))
     1544    return MatchOperand_NoMatch;
     1545  getLexer().Lex();
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
tIdentifier();

;
     1550  APInt Lmul(16, Name, 10);
```

```
1551    if (Lmul != 1 && Lmul != 2 && Lmul != 4 && Lmul != 8)
1552      return MatchOperand_NoMatch;
1553    getLexer().Lex();
1554
1555    if (getLexer().getKind() != AsmToken::EndOfStatement)
1556      return MatchOperand_NoMatch;
1557
1558    Operands.push_back(RISCVOperand::createVType(Sew, Lmul, S, isRV64()));
1559
1560    return MatchOperand_Success;
1561  }
1562
1563  OperandMatchResultTy RISCVAsmParser::parseMaskReg(OperandVector &Operands) {
1564    switch (getLexer().getKind()) {
1565    default:
```

evandro — ✔ Done

Please, rename to `parseMaskReg()`.

```
1566      return MatchOperand_NoMatch;
1567    case AsmToken::Identifier:
1568      StringRef Name = getLexer().getTok().getIdentifier();
1569      if (!Name.consume_back(".t")) {
1570        Error(getLoc(), "expected '.t' suffix");
1571        return MatchOperand_ParseFail;
1572      }
1573      Register RegNo;
1574      matchRegisterNameHelper(isRV32E(), RegNo, Name);
1575
1576      if (RegNo == RISCV::NoRegister)
```

rogfer01 — Not Done

Shouldn't we check that `RegNo == RISCV::V0` here? I tried with

```
vadd.vv  v3, v4, v5, v0.t
vadd.vv  v3, v4, v5, v4.t
```

and I got

```
vadd.vv  v3, v4, v5, v0.t        # encoding: [0xd7,0x81,0x42,0x00]
vadd.vv  v3, v4, v5, v4.t        # encoding: [0xd7,0x81,0x42,0x00]
```

This doesn't seem right, does it?

```
1577        return MatchOperand_NoMatch;
1578      if (RegNo != RISCV::V0)
1579        return MatchOperand_NoMatch;
1580      SMLoc S = getLoc();
1581      SMLoc E = SMLoc::getFromPointer(S.getPointer() - 1);
1582      getLexer().Lex();
1583      Operands.push_back(RISCVOperand::createReg(RegNo, S, E, isRV64()));
1584    }
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
1392 1589  OperandMatchResultTy
```

```
1393 1590 RISCVAsmParser::parseMemOpBaseReg(OperandVector &Operands) {
1394 1591   if (getLexer().isNot(AsmToken::LParen)) {
1395 1592     Error(getLoc(), "expected '('");
1396 1593     return MatchOperand_ParseFail;
1397 1594   }
1398 1595
1399 1596   getParser().Lex(); // Eat '('
```

Show All 28 Lines

```
1428 1625   // operand that is silently dropped.
1429 1626   //
1430 1627   // Instead, we use this custom parser. This will: allow (and discard) an
1431 1628   // offset if it is zero; require (and discard) parentheses; and add only the
1432 1629   // parsed register operand to `Operands`.
1433 1630   //
1434 1631   // These operands are printed with RISCVInstPrinter::printAtomicMemOp, which
1435 1632   // will only print the register surrounded by parentheses (which GNU as also
1436 1633   // uses as its canonical representation for these operands).
```

rogfer01 — Not Done

Could this encoding logic be in `RISCVOperand::createVType` itself?

You would have to pass `sew` and `lmul` without doing `logBase2()` here and the call in line `1395` would set `lmul = 1`.

```
1437 1634   std::unique_ptr<RISCVOperand> OptionalImmOp;
1438 1635
1439 1636   if (getLexer().isNot(AsmToken::LParen)) {
1440 1637     // Parse an Integer token. We do not accept arbitrary constant expressions
1441 1638     // in the offset field (because they may include parens, which complicates
1442 1639     // parsing a lot).
```

fpallares — Not Done

There is an issue here: we are accepting an instruction like `vadd.vv v1, v2, v3, v0` (without `.t` suffix). This happens because `RISCVAsmParser::parseOperand` falls back to parse a register if a specific parsing like this function returns `MatchOperand_NoMatch`.

I've been looking into how to fix that and I believe we can diagnose an error here doing something like this:

```
if (!Name.consume_back(".t")) {
  Error(getLoc(), "expected '.t' suffix");
  return MatchOperand_ParseFail;
}
```

In summary:

- `v0.t` is accepted by the current parser, this is already OK.
- things like `v0.` or `v0.x` will be rejected by the proposed change.
- things like `v4.t` are already rejected by the `PredicateMethod` of `VRMaskAsmOperand`. I'd suggest adding a `DiagnosticType` in `VRMaskAsmOperand` in tablegen (`RISCVInstrInfoV.td`) [illegible] se diagnostic.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

HsiangKai Author — ✔ Done

It makes sense. I will modify the patch according to your comments. Thanks a lot.

```
1443 1640     int64_t ImmVal;
1444 1641     SMLoc ImmStart = getLoc();
1445 1642     if (getParser().parseIntToken(ImmVal,
1446 1643                                   "expected '(' or optional integer offset"))
1447 1644       return MatchOperand_ParseFail;
1448 1645
1449 1646     // Create a RISCVOperand for checking later (so the error messages are
1450 1647     // nicer), but we don't add it to Operands.
```

▲ Show 20 Lines • Show All 588 Lines • ▼ Show 20 Line(s)

```
2039 2236     SMLoc ErrorLoc = ((RISCVOperand &)*Operands[3]).getStartLoc();
2040 2237     return Error(ErrorLoc, "the second input operand must be tp/x4 when using "
2041 2238                            "%tprel_add modifier");
2042 2239   }
2043 2240
2044 2241   return false;
2045 2242 }
2046 2243
     2244 std::unique_ptr<RISCVOperand> RISCVAsmParser::defaultMaskRegOp() const {
```

rogfer01 — Not Done

You plan to add tests for the erros diagnosed in this function as well, right?

HsiangKai Author — ✔ Done

Yes, I will add more tests for these invalid combinations later.

```
     2245   return RISCVOperand::createReg(RISCV::NoRegister, llvm::SMLoc(),
     2246                                  llvm::SMLoc(), isRV64());
     2247 }
     2248
     2249 bool RISCVAsmParser::validateInstruction(MCInst &Inst,
     2250                                          OperandVector &Operands) {
     2251   const MCInstrDesc &MCID = MII.get(Inst.getOpcode());
```

fpallares — Not Done

`Operands[1]` here might be an out-of-bounds access for some instructions with less than 2 operators (e.g. `RISCV::JALR`).

HsiangKai Author — ✔ Done

Operands[0] in OperandVector will be operator. Operands[1] will be the first operand.
If there is an instruction with no operand, it might be an out-of-bounds access.

I will change the default location to point to operator. Thanks.


```
onstraintOffset) & RISCV::ConstraintMask;
onstraint)
     2256
```

```
2257   unsigned DestReg = Inst.getOperand(0).getReg();
2258   // Operands[1] will be the first operand, DestReg.
2259   SMLoc Loc = Operands[1]->getStartLoc();
2260   if ((TargetFlags == RISCV::WidenV) || (TargetFlags == RISCV::WidenW) ||
2261       (TargetFlags == RISCV::SlideUp) || (TargetFlags == RISCV::Vrgather) ||
2262       (TargetFlags == RISCV::Vcompress)) {
2263     if (TargetFlags != RISCV::WidenW) {
2264       unsigned Src2Reg = Inst.getOperand(1).getReg();
2265       if (DestReg == Src2Reg)
2266         return Error(Loc, "The destination vector register group cannot overlap"
2267                           " the source vector register group.");
2268     }
2269     if (Inst.getOperand(2).isReg()) {
2270       unsigned Src1Reg = Inst.getOperand(2).getReg();
2271       if (DestReg == Src1Reg)
2272         return Error(Loc, "The destination vector register group cannot overlap"
2273                           " the source vector register group.");
2274     }
2275     if (Inst.getNumOperands() == 4) {
2276       unsigned MaskReg = Inst.getOperand(3).getReg();
2277
2278       if (DestReg == MaskReg)
2279         return Error(Loc, "The destination vector register group cannot overlap"
2280                           " the mask register.");
2281     }
2282   } else if (TargetFlags == RISCV::Narrow) {
2283     unsigned Src2Reg = Inst.getOperand(1).getReg();
2284     if (DestReg == Src2Reg)
2285       return Error(Loc, "The destination vector register group cannot overlap"
2286                         " the source vector register group.");
2287   } else if (TargetFlags == RISCV::WidenCvt || TargetFlags == RISCV::Iota) {
2288     unsigned Src2Reg = Inst.getOperand(1).getReg();
2289     if (DestReg == Src2Reg)
2290       return Error(Loc, "The destination vector register group cannot overlap"
2291                         " the source vector register group.");
2292     if (Inst.getNumOperands() == 3) {
2293       unsigned MaskReg = Inst.getOperand(2).getReg();
2294
2295       if (DestReg == MaskReg)
2296         return Error(Loc, "The destination vector register group cannot overlap"
2297                           " the mask register.");
2298     }
2299   }
2300   return false;
```

rogfer01 — Not Done

There is a macro `LLVM_FALLTHROUGH` for these cases. You can use it here and in the later occurrences of a similar comment.

```
2301 }
2302
2047 2303 bool RISCVAsmParser::processInstruction(MCInst &Inst, SMLoc IDLoc,
2048 2304                                         OperandVector &Operands,
2049 2305                                         MCStreamer &Out) {
2050 2306   Inst.setLoc(IDLoc);
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
2054 2310   break;
```

▲ Show 20 Lines • Show All 92 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/Disassembler/RISCVDisassembler.cpp

View Options

Show All 12 Lines

```
13  13  #include "MCTargetDesc/RISCVMCTargetDesc.h"
14  14  #include "TargetInfo/RISCVTargetInfo.h"
15  15  #include "Utils/RISCVBaseInfo.h"
16  16  #include "llvm/CodeGen/Register.h"
17  17  #include "llvm/MC/MCContext.h"
18  18  #include "llvm/MC/MCDisassembler/MCDisassembler.h"
19  19  #include "llvm/MC/MCFixedLenDisassembler.h"
20  20  #include "llvm/MC/MCInst.h"
    21  #include "llvm/MC/MCInstrInfo.h"
21  22  #include "llvm/MC/MCRegisterInfo.h"
22  23  #include "llvm/MC/MCSubtargetInfo.h"
23  24  #include "llvm/Support/Endian.h"
24  25  #include "llvm/Support/TargetRegistry.h"
25  26
26  27  using namespace llvm;
27  28
28  29  #define DEBUG_TYPE "riscv-disassembler"
29  30
30  31  typedef MCDisassembler::DecodeStatus DecodeStatus;
31  32
32  33  namespace {
33  34  class RISCVDisassembler : public MCDisassembler {
    35    std::unique_ptr<MCInstrInfo const> const MCII;
34  36
35  37  public:
36        RISCVDisassembler(const MCSubtargetInfo &STI, MCContext &Ctx)
37            : MCDisassembler(STI, Ctx) {}
    38    RISCVDisassembler(const MCSubtargetInfo &STI, MCContext &Ctx,
    39                      MCInstrInfo const *MCII)
    40        : MCDisassembler(STI, Ctx), MCII(MCII) {}
38  41
39  42    DecodeStatus getInstruction(MCInst &Instr, uint64_t &Size,
40  43                                ArrayRef<uint8_t> Bytes, uint64_t Address,
41  44                                raw_ostream &CStream) const override;
42  45  };
43  46  } // end anonymous namespace
44  47
45  48  static MCDisassembler *createRISCVDisassembler(const Target &T,
46  49                                                 const MCSubtargetInfo &STI,
47  50                                                 MCContext &Ctx) {
48        return new RISCVDisassembler(STI, Ctx);
    51    return new RISCVDisassembler(STI, Ctx, T.createMCInstrInfo());
49  52  }
50  53
51  54  extern "C" LLVM_EXTERNAL_VISIBILITY void LLVMInitializeRISCVDisassembler() {
52  55    // Register the disassembler for each target.
53  56    TargetRegistry::RegisterMCDisassembler(getTheRISCV32Target(),
54  57                                           createRISCVDisassembler);
55  58    TargetRegistry::RegisterMCDisassembler(getTheRISCV64Target(),
56  59                                           createRISCVDisassembler);
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

s • Show All 86 Lines • ▼ Show 20 Line(s)

```
1   1
1   1                                                                    ;
145 148
```

```
146 149   Register Reg = RISCV::X8 + RegNo;
147 150   Inst.addOperand(MCOperand::createReg(Reg));
148 151   return MCDisassembler::Success;
149 152 }
150 153
    154 static DecodeStatus DecodeVRRegisterClass(MCInst &Inst, uint64_t RegNo,
```

fpallares — Not Done

Following @evandro 's naming suggestions, this should probably be renamed to `DecodeVRegisterClass`.

HsiangKai Author — ✔ Done

The function name will be expected by TableGen as `"Decode" + Record->getName().str() + "RegisterClass"`. It seems related to the name of register class. Currently, the name of RegisterClass for V-extension registers is "VR".

fpallares — ✔ Done

I see, sorry for the noise.

```
    155                                        uint64_t Address,
    156                                        const void *Decoder) {
157 157   if (RegNo >= 32)
158 158     return MCDisassembler::Fail;
159 159
160 160   Register Reg = RISCV::V0 + RegNo;
161 161   Inst.addOperand(MCOperand::createReg(Reg));
162 162   return MCDisassembler::Success;
163 163 }
164 164
165 165 static DecodeStatus decodeVMaskReg(MCInst &Inst, uint64_t RegNo,
```

evandro — ✔ Done

Please, rename to `decodeVMaskReg()`.

```
    166                                    uint64_t Address, const void *Decoder) {
167 167   Register Reg = RISCV::NoRegister;
168 168   switch (RegNo) {
169 169   default:
170 170     return MCDisassembler::Fail;
171 171   case 0:
172 172     Reg = RISCV::V0;
173 173     break;
174 174   case 1:
175 175     break;
176 176   }
177 177   Inst.addOperand(MCOperand::createReg(Reg));
178 178   return MCDisassembler::Success;
179 179 }
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
1   183 nstructions *SP compressed instructions. The SP
1   184 oded in the instruction.
1   185 Inst, int64_t Address, const void *Decoder) {
154 186   if (Inst.getOpcode() == RISCV::C_LWSP || Inst.getOpcode() == RISCV::C_SWSP ||
```

```
155 185       Inst.getOpcode() == RISCV::C_LDSP || Inst.getOpcode() == RISCV::C_SDSP ||
156 186       Inst.getOpcode() == RISCV::C_FLWSP ||
157 187       Inst.getOpcode() == RISCV::C_FSWSP ||
158 188       Inst.getOpcode() == RISCV::C_FLDSP ||
```

▲ Show 20 Lines • Show All 214 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.h

View Options

Show All 35 Lines

```
36  36  void printCSRSystemRegister(const MCInst *MI, unsigned OpNo,
37  37                               const MCSubtargetInfo &STI, raw_ostream &O);
38  38  void printFenceArg(const MCInst *MI, unsigned OpNo,
39  39                     const MCSubtargetInfo &STI, raw_ostream &O);
40  40  void printFRMArg(const MCInst *MI, unsigned OpNo, const MCSubtargetInfo &STI,
41  41                   raw_ostream &O);
42  42  void printAtomicMemOp(const MCInst *MI, unsigned OpNo,
43  43                        const MCSubtargetInfo &STI, raw_ostream &O);
    44  void printVTypeI(const MCInst *MI, unsigned OpNo, const MCSubtargetInfo &STI,
    45                   raw_ostream &O);
    46  void printVMaskReg(const MCInst *MI, unsigned OpNo,
```

evandro ✔ Done

Please, rename to `printVMaskReg()`.

```
    47                     const MCSubtargetInfo &STI, raw_ostream &O);
44  48
45  49  // Autogenerated by tblgen.
46  50  void printInstruction(const MCInst *MI, uint64_t Address,
47  51                        const MCSubtargetInfo &STI, raw_ostream &O);
48  52  bool printAliasInstr(const MCInst *MI, uint64_t Address,
49  53                       const MCSubtargetInfo &STI, raw_ostream &O);
50  54  void printCustomAliasOperand(const MCInst *MI, uint64_t Address,
51  55                               unsigned OpIdx, unsigned PrintMethodIdx,
52  56                               const MCSubtargetInfo &STI, raw_ostream &O);
53  57  static const char *getRegisterName(unsigned RegNo);
54  58  static const char *getRegisterName(unsigned RegNo, unsigned AltIdx);
55  59  };
56  60  } // namespace llvm
57  61
58  62  #endif
```

llvm/lib/Target/RISCV/MCTargetDesc/RISCVInstPrinter.cpp

View Options

Show First 20 Lines • Show All 144 Lines • ▼ Show 20 Line(s)

```
145 145
146 146  assert(MO.isReg() && "printAtomicMemOp can only print register operands");
147 147  O << "(";
148 148  printRegName(O, MO.getReg());
149 149  O << ")";
150 150  return;
151 151  }
152 152
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
peI(const MCInst *MI, unsigned OpNo,
    const MCSubtargetInfo &STI, raw_ostream &O) {
(OpNo).getImm();
x7;
    157  unsigned Lmul = Imm & 0x3;
```

```
158
159   Lmul = 0x1 << Lmul;
160   Sew = 0x1 << (Sew + 3);
161   O << "e" << Sew << ",m" << Lmul;
162 }
163
164 void RISCVInstPrinter::printVMaskReg(const MCInst *MI, unsigned OpNo,
```

evandro ✔ Done

Please, rename to `printVMaskReg()`.

```
165                                      const MCSubtargetInfo &STI,
166                                      raw_ostream &O) {
167   const MCOperand &MO = MI->getOperand(OpNo);
168
169   assert(MO.isReg() && "printVMaskReg can only print register operands");
170   if (MO.getReg() == RISCV::NoRegister)
171     return;
172   O << ", ";
173   printRegName(O, MO.getReg());
174   O << ".t";
175 }
176
177 const char *RISCVInstPrinter::getRegisterName(unsigned RegNo) {
178   return getRegisterName(RegNo, ArchRegNames ? RISCV::NoRegAltName
179                                              : RISCV::ABIRegAltName);
180 }
```

llvm/lib/Target/RISCV/MCTargetDesc/RISCVMCCodeEmitter.cpp ☰ View Options

Show First 20 Lines • Show All 74 Lines • ▼ Show 20 Line(s)

```
75  75
76  76   unsigned getImmOpValueAsr1(const MCInst &MI, unsigned OpNo,
77  77                              SmallVectorImpl<MCFixup> &Fixups,
78  78                              const MCSubtargetInfo &STI) const;
79  79
80  80   unsigned getImmOpValue(const MCInst &MI, unsigned OpNo,
81  81                          SmallVectorImpl<MCFixup> &Fixups,
82  82                          const MCSubtargetInfo &STI) const;
83  83
84      unsigned getVMaskReg(const MCInst &MI, unsigned OpNo,
```

evandro ✔ Done

Please, rename to `getVMaskReg()`.

```
85                          SmallVectorImpl<MCFixup> &Fixups,
86                          const MCSubtargetInfo &STI) const;
87  };
88  } // end anonymous namespace
89
90  MCCodeEmitter *llvm::createRISCVMCCodeEmitter(const MCInstrInfo &MCII,
91                                                const MCRegisterInfo &MRI,
                                                  MCContext &Ctx) {
                                     (Ctx, MCII);
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

▲ Show 20 Lines • Show All 278 Lines • ▼ Show 20 Line(s)

```
369 373     MCFixup::create(0, Dummy, MCFixupKind(RISCV::fixup_riscv_relax),
370 374                     MI.getLoc()));
371 375     ++MCNumFixups;
372 376   }
373 377
374 378   return 0;
375 379 }
376 380
    381 unsigned RISCVMCCodeEmitter::getVMaskReg(const MCInst &MI, unsigned OpNo,
    382                                          SmallVectorImpl<MCFixup> &Fixups,
    383                                          const MCSubtargetInfo &STI) const {
    384   MCOperand MO = MI.getOperand(OpNo);
    385   assert(MO.isReg() && "Expected a register.");
    386
    387   switch (MO.getReg()) {
    388   default:
    389     llvm_unreachable("Invalid mask register.");
    390   case RISCV::V0:
    391     return 0;
    392   case RISCV::NoRegister:
    393     return 1;
    394   }
    395 }
    396
377 397 #include "RISCVGenMCCodeEmitter.inc"
```

llvm/lib/Target/RISCV/RISCV.td View Options

Show First 20 Lines • Show All 141 Lines • ▼ Show 20 Line(s)

```
142 142
143 143 def FeatureRVCHints
144 144     : SubtargetFeature<"rvc-hints", "EnableRVCHintInstrs", "true",
145 145                        "Enable RVC Hint Instructions.">;
146 146 def HasRVCHints : Predicate<"Subtarget->enableRVCHintInstrs()">,
147 147                            AssemblerPredicate<(all_of FeatureRVCHints),
148 148                            "RVC Hint Instructions">;
149 149
    150 def FeatureStdExtV
    151     : SubtargetFeature<"experimental-v", "HasStdExtV", "true",
    152                        "'V' (Vector Instructions)">;
```

evandro Not Done

Should it also imply `FeatureStdExtD` ?

HsiangKai Author ✔ Done

I didn't find that V will imply D extension. Could you indicate the description in spec about this?

evandro Not Done

Indeed it doesn't:

...clude floating-point, then a `fcsr` register is also added to hold ...m CSRs as explained below.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

Which means that the instructions that take FP scalars should also depend on `hasStdExt{F|D}`. Likewise when the scalar is a 64 bit integer, `hasRV64`.

rogfer01 Not Done

I don't think we have to do anything special with 64 bit: section 11.1 states what is the behaviour when `XLEN` is different than `SEW`.

evandro Not Done

Good point.

HsiangKai Author ✔ Done

Yes, it is a problem about accessing fcsr if F extension is turned off. However, I found that there is vcsr in the latest version.

https://github.com/riscv/riscv-v-spec/commit/b25b643b9c29b4fde570293c64ca682a99a09f2a

So, I think we could keep V and F as separate unrelated extensions.

I will add predicates for vector floating point operations.

```
153 def HasStdExtV : Predicate<"Subtarget->hasStdExtV()">,
154                           AssemblerPredicate<(all_of FeatureStdExtV),
```

fpallares ✔ Done

Since D69899 was commited, we should add the feature name in the `AssemblerPredicate` parameter and update the missing feature error messages in the tests. Like this:

```
def HasStdExtV : Predicate<"Subtarget->hasStdExtV()">,
                           AssemblerPredicate<"FeatureStdExtV",
                           "'V' (Vector Instructions)">;
```

```
155                           "'V' (Vector Instructions)">;
156
150 157 def Feature64Bit
151 158     : SubtargetFeature<"64bit", "HasRV64", "true", "Implements RV64">;
152 159 def IsRV64 : Predicate<"Subtarget->is64Bit()">,
153 160                       AssemblerPredicate<(all_of Feature64Bit),
154 161                       "RV64I Base Instruction Set">;
155 162 def IsRV32 : Predicate<"!Subtarget->is64Bit()">,
156 163                       AssemblerPredicate<(all_of (not Feature64Bit)),
157 164                       "RV32I Base Instruction Set">;
```

▲ Show 20 Lines • Show All 78 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/RISCVInstrFormats.td

View Options

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

...es • Show All 43 Lines • ▼ Show 20 Line(s)

```
...at<11>;
...at<12>;
46 46 def InstFormatCS      : InstFormat<13>;
```

```
def InstFormatCA     : InstFormat<14>;
def InstFormatCB     : InstFormat<15>;
def InstFormatCJ     : InstFormat<16>;
def InstFormatOther  : InstFormat<17>;

class RISCVVConstraint<bits<4> val> {
  bits<4> Value = val;
}
def NoConstraint : RISCVVConstraint<0>;
def WidenV       : RISCVVConstraint<1>;
def WidenW       : RISCVVConstraint<2>;
def WidenCvt     : RISCVVConstraint<3>;
def QuadWiden    : RISCVVConstraint<4>;
def Narrow       : RISCVVConstraint<5>;
def Iota         : RISCVVConstraint<6>;
def SlideUp      : RISCVVConstraint<7>;
def Vrgather     : RISCVVConstraint<8>;
def Vcompress    : RISCVVConstraint<9>;

// The following opcode names match those given in Table 19.1 in the
// RISC-V User-level ISA specification ("RISC-V base opcode map").
class RISCVOpcode<bits<7> val> {
  bits<7> Value = val;
}
def OPC_LOAD      : RISCVOpcode<0b0000011>;
def OPC_LOAD_FP   : RISCVOpcode<0b0000111>;
def OPC_MISC_MEM  : RISCVOpcode<0b0001111>;
def OPC_OP_IMM    : RISCVOpcode<0b0010011>;
def OPC_AUIPC     : RISCVOpcode<0b0010111>;
def OPC_OP_IMM_32 : RISCVOpcode<0b0011011>;
def OPC_STORE     : RISCVOpcode<0b0100011>;
def OPC_STORE_FP  : RISCVOpcode<0b0100111>;
def OPC_AMO       : RISCVOpcode<0b0101111>;
def OPC_OP        : RISCVOpcode<0b0110011>;
def OPC_LUI       : RISCVOpcode<0b0110111>;
def OPC_OP_32     : RISCVOpcode<0b0111011>;
def OPC_MADD      : RISCVOpcode<0b1000011>;
def OPC_MSUB      : RISCVOpcode<0b1000111>;
def OPC_NMSUB     : RISCVOpcode<0b1001011>;
def OPC_NMADD     : RISCVOpcode<0b1001111>;
def OPC_OP_FP     : RISCVOpcode<0b1010011>;
def OPC_OP_V      : RISCVOpcode<0b1010111>;
def OPC_BRANCH    : RISCVOpcode<0b1100011>;
def OPC_JALR      : RISCVOpcode<0b1100111>;
def OPC_JAL       : RISCVOpcode<0b1101111>;
def OPC_SYSTEM    : RISCVOpcode<0b1110011>;

class RVInst<dag outs, dag ins, string opcodestr, string argstr,
             list<dag> pattern, InstFormat format>
    : Instruction {
```

Show All 12 Lines

```
  let Namespace = "RISCV";

  dag OutOperandList = outs;
  dag InOperandList = ins;
  let AsmString = opcodestr # "\t" # argstr;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
ue;
```

```
118 118   // Defaults
    119   RISCVVConstraint RVVConstraint = NoConstraint;
    120   let TSFlags{8-5} = RVVConstraint.Value;
102 121 }
103 122
104 123 // Pseudo instructions
105 124 class Pseudo<dag outs, dag ins, list<dag> pattern, string opcodestr = "", string argstr = "">
106 125     : RVInst<outs, ins, opcodestr, argstr, pattern, InstFormatPseudo>,
107 126       Sched<[]> {
108 127   let isPseudo = 1;
109 128   let isCodeGenOnly = 1;
```

▲ Show 20 Lines • Show All 206 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/RISCVInstrFormatsV.td

View Options

This file was added.

```
  1 //===-- RISCVInstrFormatsV.td - RISCV V Instruction Formats --*- tablegen -*-=//
  2 //
  3 // Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
  4 // See https://llvm.org/LICENSE.txt for license information.
  5 // SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
  6 //
  7 //===----------------------------------------------------------------------===//
  8 //
  9 //  This file describes the RISC-V V extension instruction formats.
 10 //
 11 //===----------------------------------------------------------------------===//
 12
 13 class RISCVVFormat<bits<3> val> {
 14   bits<3> Value = val;
 15 }
 16 def OPIVV : RISCVVFormat<0b000>;
 17 def OPFVV : RISCVVFormat<0b001>;
 18 def OPMVV : RISCVVFormat<0b010>;
 19 def OPIVI : RISCVVFormat<0b011>;
 20 def OPIVX : RISCVVFormat<0b100>;
 21 def OPFVF : RISCVVFormat<0b101>;
 22 def OPMVX : RISCVVFormat<0b110>;
 23
 24 class RISCVMOP<bits<3> val> {
 25   bits<3> Value = val;
 26 }
 27 def MOPLDUnitStrideU : RISCVMOP<0b000>;
 28 def MOPLDStridedU    : RISCVMOP<0b010>;
 29 def MOPLDIndexedU    : RISCVMOP<0b011>;
 30 def MOPLDUnitStrideS : RISCVMOP<0b100>;
 31 def MOPLDStridedS    : RISCVMOP<0b110>;
 32 def MOPLDIndexedS    : RISCVMOP<0b111>;
 33
 34 def MOPSTUnitStride  : RISCVMOP<0b000>;
 35 def MOPSTStrided     : RISCVMOP<0b010>;
 36 def MOPSTIndexedOrder: RISCVMOP<0b011>;
 37 def MOPSTIndexedUnOrd: RISCVMOP<0b111>;
 38
                                        {
                                        MOP<0b00000>;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
43  def LUMOPUnitStrideWholeReg : RISCVLSUMOP<0b01000>;
44  def LUMOPUnitStrideFF: RISCVLSUMOP<0b10000>;
45  def SUMOPUnitStride  : RISCVLSUMOP<0b00000>;
46  def SUMOPUnitStrideWholeReg : RISCVLSUMOP<0b01000>;
47
48  class RISCVWidth<bits<3> val> {
49    bits<3> Value = val;
50  }
51  def LSWidthVByte : RISCVWidth<0b000>;
52  def LSWidthVHalf : RISCVWidth<0b101>;
53  def LSWidthVWord : RISCVWidth<0b110>;
54  def LSWidthVSEW  : RISCVWidth<0b111>;
55
56  class RVInstSetVLi<dag outs, dag ins, string opcodestr, string argstr>
57      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatI> {
58    bits<5> rs1;
59    bits<5> rd;
60    bits<11> vtypei;
61
62    let Inst{31} = 0;
63    let Inst{30-20} = vtypei;
64    let Inst{19-15} = rs1;
65    let Inst{14-12} = 0b111;
66    let Inst{11-7} = rd;
67    let Opcode = OPC_OP_V.Value;
68
69    let Defs = [VTYPE, VL];
70  }
71
72  class RVInstSetVL<dag outs, dag ins, string opcodestr, string argstr>
73      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
74    bits<5> rs2;
75    bits<5> rs1;
76    bits<5> rd;
77
78    let Inst{31} = 1;
79    let Inst{30-25} = 0b000000;
80    let Inst{24-20} = rs2;
81    let Inst{19-15} = rs1;
82    let Inst{14-12} = 0b111;
83    let Inst{11-7} = rd;
84    let Opcode = OPC_OP_V.Value;
85
86    let Defs = [VTYPE, VL];
87  }
88
89  class RVInstVV<bits<6> funct6, RISCVVFormat opv, dag outs, dag ins,
90                 string opcodestr, string argstr>
91      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
92    bits<5> vs2;
93    bits<5> vs1;
94    bits<5> vd;
95    bit vm;
96
97    let Inst{31-26} = funct6;
98    let Inst{25} = vm;
```



```
103   let Opcode = OPC_OP_V.Value;
```

```
104
105   let Uses = [VTYPE, VL];
106 }
107
108 class RVInstVX<bits<6> funct6, RISCVVFormat opv, dag outs, dag ins,
109                string opcodestr, string argstr>
110     : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
111   bits<5> vs2;
112   bits<5> rs1;
113   bits<5> vd;
114   bit vm;
115
116   let Inst{31-26} = funct6;
117   let Inst{25} = vm;
118   let Inst{24-20} = vs2;
119   let Inst{19-15} = rs1;
120   let Inst{14-12} = opv.Value;
121   let Inst{11-7} = vd;
122   let Opcode = OPC_OP_V.Value;
123
124   let Uses = [VTYPE, VL];
125 }
126
127 class RVInstV2<bits<6> funct6, bits<5> vs2, RISCVVFormat opv, dag outs, dag ins,
128                string opcodestr, string argstr>
129     : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
130   bits<5> rs1;
131   bits<5> vd;
132   bit vm;
133
134   let Inst{31-26} = funct6;
135   let Inst{25} = vm;
136   let Inst{24-20} = vs2;
137   let Inst{19-15} = rs1;
138   let Inst{14-12} = opv.Value;
139   let Inst{11-7} = vd;
140   let Opcode = OPC_OP_V.Value;
141
142   let Uses = [VTYPE, VL];
143 }
144
145 class RVInstIVI<bits<6> funct6, dag outs, dag ins, string opcodestr,
146                 string argstr>
147     : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
148   bits<5> vs2;
149   bits<5> imm;
150   bits<5> vd;
151   bit vm;
152
153   let Inst{31-26} = funct6;
154   let Inst{25} = vm;
155   let Inst{24-20} = vs2;
156   let Inst{19-15} = imm;
157   let Inst{14-12} = 0b011;
158   let Inst{11-7} = vd;
159   let Opcode = OPC_OP_V.Value;
```



```
164 class RVInstV<bits<6> funct6, bits<5> vs1, RISCVVFormat opv, dag outs,
```

```
165                dag ins, string opcodestr, string argstr>
166      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
167    bits<5> vs2;
168    bits<5> vd;
169    bit vm;
170
171    let Inst{31-26} = funct6;
172    let Inst{25} = vm;
173    let Inst{24-20} = vs2;
174    let Inst{19-15} = vs1;
175    let Inst{14-12} = opv.Value;
176    let Inst{11-7} = vd;
177    let Opcode = OPC_OP_V.Value;
178
179    let Uses = [VTYPE, VL];
180  }
181
182  class RVInstVLU<bits<3> nf, RISCVMOP mop, RISCVLSUMOP lumop,
183                  RISCVWidth width, dag outs, dag ins, string opcodestr,
184                  string argstr>
185      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
186    bits<5> rs1;
187    bits<5> vd;
188    bit vm;
189
190    let Inst{31-29} = nf;
191    let Inst{28-26} = mop.Value;
192    let Inst{25} = vm;
193    let Inst{24-20} = lumop.Value;
194    let Inst{19-15} = rs1;
195    let Inst{14-12} = width.Value;
196    let Inst{11-7} = vd;
197    let Opcode = OPC_LOAD_FP.Value;
198
199    let Uses = [VTYPE, VL];
200  }
201
202  class RVInstVLS<bits<3> nf, RISCVMOP mop, RISCVWidth width,
203                  dag outs, dag ins, string opcodestr, string argstr>
204      : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
205    bits<5> rs2;
206    bits<5> rs1;
207    bits<5> vd;
208    bit vm;
209
210    let Inst{31-29} = nf;
211    let Inst{28-26} = mop.Value;
212    let Inst{25} = vm;
213    let Inst{24-20} = rs2;
214    let Inst{19-15} = rs1;
215    let Inst{14-12} = width.Value;
216    let Inst{11-7} = vd;
217    let Opcode = OPC_LOAD_FP.Value;
218
219    let Uses = [VTYPE, VL];
220  }
                                                                    CVMOP mop, RISCVWidth width,
                                                                    ns, string opcodestr, string argstr>
                                                                    tr, argstr, [], InstFormatR> {
225    bits<5> vs2;
```

```
226   bits<5> rs1;
227   bits<5> vd;
228   bit vm;
229
230   let Inst{31-29} = nf;
231   let Inst{28-26} = mop.Value;
232   let Inst{25} = vm;
233   let Inst{24-20} = vs2;
234   let Inst{19-15} = rs1;
235   let Inst{14-12} = width.Value;
236   let Inst{11-7} = vd;
237   let Opcode = OPC_LOAD_FP.Value;
238
239   let Uses = [VTYPE, VL];
240 }
241
242 class RVInstVSU<bits<3> nf, RISCVMOP mop, RISCVLSUMOP sumop,
243                 RISCVWidth width, dag outs, dag ins, string opcodestr,
244                 string argstr>
245     : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
246   bits<5> rs1;
247   bits<5> vs3;
248   bit vm;
249
250   let Inst{31-29} = nf;
251   let Inst{28-26} = mop.Value;
252   let Inst{25} = vm;
253   let Inst{24-20} = sumop.Value;
254   let Inst{19-15} = rs1;
255   let Inst{14-12} = width.Value;
256   let Inst{11-7} = vs3;
257   let Opcode = OPC_STORE_FP.Value;
258
259   let Uses = [VTYPE, VL];
260 }
261
262 class RVInstVSS<bits<3> nf, RISCVMOP mop, RISCVWidth width,
263                 dag outs, dag ins, string opcodestr, string argstr>
264     : RVInst<outs, ins, opcodestr, argstr, [], InstFormatR> {
265   bits<5> rs2;
266   bits<5> rs1;
267   bits<5> vs3;
268   bit vm;
269
270   let Inst{31-29} = nf;
271   let Inst{28-26} = mop.Value;
272   let Inst{25} = vm;
273   let Inst{24-20} = rs2;
274   let Inst{19-15} = rs1;
275   let Inst{14-12} = width.Value;
276   let Inst{11-7} = vs3;
277   let Opcode = OPC_STORE_FP.Value;
278
279   let Uses = [VTYPE, VL];
280 }
281
                                                  CVMOP mop, RISCVWidth width,
                                                  ns, string opcodestr, string argstr>
                                                  tr, argstr, [], InstFormatR> {
286   bits<5> rs1;
```

```
287   bits<5> vs3;
288   bit vm;
289
290   let Inst{31-29} = nf;
291   let Inst{28-26} = mop.Value;
292   let Inst{25} = vm;
293   let Inst{24-20} = vs2;
294   let Inst{19-15} = rs1;
295   let Inst{14-12} = width.Value;
296   let Inst{11-7} = vs3;
297   let Opcode = OPC_STORE_FP.Value;
298
299   let Uses = [VTYPE, VL];
300 }
```

llvm/lib/Target/RISCV/RISCVInstrInfo.h

View Options

Show First 20 Lines • Show All 127 Lines • ▼ Show 20 Line(s)

```
128 128   // Insert a call to an outlined function into a given basic block.
129 129   virtual MachineBasicBlock::iterator
130 130   insertOutlinedCall(Module &M, MachineBasicBlock &MBB,
131 131                      MachineBasicBlock::iterator &It, MachineFunction &MF,
132 132                      const outliner::Candidate &C) const override;
133 133 protected:
134 134   const RISCVSubtarget &STI;
135 135 };
136       }
    136
    137 namespace RISCV {
    138 // Match with the definitions in RISCVInstrFormatsV.td
    139 enum RVVConstraintType {
    140   NoConstraint = 0,
    141   WidenV = 1,
    142   WidenW = 2,
    143   WidenCvt = 3,
    144   QuadWiden = 4,
    145   Narrow = 5,
    146   Iota = 6,
    147   SlideUp = 7,
    148   Vrgather = 8,
    149   Vcompress = 9,
    150
    151   ConstraintOffset = 5,
    152   ConstraintMask = 0b1111
    153 };
    154 } // end namespace RISCV
    155
    156 } // end namespace llvm
137 157 #endif
```

llvm/lib/Target/RISCV/RISCVInstrInfo.td

View Options

Show First 20 Lines • Show All 1159 Lines • ▼ Show 20 Line(s)

```
1160 1160 //===----------------------------------------------------------------------===//
1165 1165 include "RISCVInstrInfoD.td"
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
1166 1166 include "RISCVInstrInfoC.td"
1167 1167 include "RISCVInstrInfoB.td"
     1168 include "RISCVInstrInfoV.td"
```

llvm/lib/Target/RISCV/RISCVInstrInfoV.td

View Options

This file was added.

```
1  //===-- RISCVInstrInfoV.td - RISC-V 'V' instructions -------*- tablegen -*-===//
2  //
3  // Part of the LLVM Project, under the Apache License v2.0 with LLVM Exceptions.
4  // See https://llvm.org/LICENSE.txt for license information.
5  // SPDX-License-Identifier: Apache-2.0 WITH LLVM-exception
6  //
7  //===----------------------------------------------------------------------===//
8  ///
9  /// This file describes the RISC-V instructions from the standard 'V',
10 /// Vector instruction set extension.
11 ///
12 //===----------------------------------------------------------------------===//
13
14 include "RISCVInstrFormatsV.td"
15
16 //===----------------------------------------------------------------------===//
17 // Operand and SDNode transformation definitions.
18 //===----------------------------------------------------------------------===//
19
20 def VTypeIAsmOperand : AsmOperandClass {
21   let Name = "VTypeI";
22   let ParserMethod = "parseVTypeI";
23   let DiagnosticType = "InvalidVTypeI";
24 }
25
26 def VTypeIOp : Operand<XLenVT> {
27   let ParserMatchClass = VTypeIAsmOperand;
28   let PrintMethod = "printVTypeI";
29   let DecoderMethod = "decodeUImmOperand<11>";
30 }
31
32 def VRegAsmOperand : AsmOperandClass {
33   let Name = "RVVRegOpOperand";
```

evandro — Done

Please, rename to `VRegAsmOperand`.

```
34   let RenderMethod = "addRegOperands";
35   let PredicateMethod = "isReg";
36   let ParserMethod = "parseRegister";
37 }
38
39 def VRegOp : RegisterOperand<VR> {
40   let ParserMatchClass = VRegAsmOperand;
```

evandro — Done

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

fpallares Not Done

Whitespace missing before the colon.

```
41   let PrintMethod = "printOperand";
42 }
43
44 def VMaskAsmOperand : AsmOperandClass {
45   let Name = "RVVMaskRegOpOperand";
```

evandro ✔ Done

Please, rename to `VMaskAsmOperand`.

```
46   let RenderMethod = "addRegOperands";
47   let PredicateMethod = "isV0Reg";
48   let ParserMethod = "parseMaskReg";
49   let IsOptional = 1;
50   let DefaultMethod = "defaultMaskRegOp";
51   let DiagnosticType = "InvalidVMaskRegister";
52 }
53
54 def VMaskOp : RegisterOperand<VMV0> {
55   let ParserMatchClass = VMaskAsmOperand;
```

evandro ✔ Done

Please, rename to `VMaskOp`.

fpallares Not Done

Ditto.

```
56   let PrintMethod = "printVMaskReg";
57   let EncoderMethod = "getVMaskReg";
58   let DecoderMethod = "decodeVMaskReg";
59 }
60
61 def simm5 : Operand<XLenVT>, ImmLeaf<XLenVT, [{return isInt<5>(Imm);}]> {
62   let ParserMatchClass = SImmAsmOperand<5>;
63   let EncoderMethod = "getImmOpValue";
64   let DecoderMethod = "decodeSImmOperand<5>";
65   let MCOperandPredicate = [{
66     int64_t Imm;
67     if (MCOp.evaluateAsConstantImm(Imm))
68       return isInt<5>(Imm);
69     return MCOp.isBareSymbolRef();
70   }];
71 }
72
73 //===----------------------------------------------------------------------===//
74 // Instruction class templates
75 //===----------------------------------------------------------------------===//
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
= 1, mayStore = 0 in {
```

evandro Not Done

No need to set fields to the default `0`.

HsiangKai Author — ✔ Done

There is no default value for hasSideEffects, mayLoad, and mayStore.

evandro — Not Done

Right. It would be cleaner if they were moved to the respective base instruction class then. Just a suggestion.

```
79  class VUnitStrideLoad<RISCVMOP mop, RISCVLSUMOP lumop, RISCVWidth width,
80                        string opcodestr>
81      : RVInstVLU<0b000, mop, lumop, width, (outs VRegOp:$vd),
82                  (ins GPR:$rs1, VMaskOp:$vm), opcodestr, "$vd, (${rs1})$vm">;
83
84  // load vd, (rs1), rs2, vm
85  class VStridedLoad<RISCVMOP mop, RISCVWidth width, string opcodestr>
86      : RVInstVLS<0b000, mop, width, (outs VRegOp:$vd),
87                  (ins GPR:$rs1, GPR:$rs2, VMaskOp:$vm), opcodestr,
88                  "$vd, (${rs1}), $rs2$vm">;
89
90  // load vd, (rs1), vs2, vm
91  class VIndexedLoad<RISCVMOP mop, RISCVWidth width, string opcodestr>
92      : RVInstVLX<0b000, mop, width, (outs VRegOp:$vd),
93                  (ins GPR:$rs1, VRegOp:$vs2, VMaskOp:$vm), opcodestr,
94                  "$vd, (${rs1}), $vs2$vm">;
95
96  // vl<nf>r.v vd, (rs1)
97  class VWholeLoad<bits<3> nf, string opcodestr>
98      : RVInstVLU<nf, MOPLDUnitStrideU, LUMOPUnitStrideWholeReg,
99                  LSWidthVSEW, (outs VRegOp:$vd), (ins GPR:$rs1),
100                 opcodestr, "$vd, (${rs1})"> {
101   let vm = 1;
102   let Uses = [];
103 }
104 }
105
106 let hasSideEffects = 0, mayLoad = 0, mayStore = 1 in {
107 // store vd, vs3, (rs1), vm
```

fpallares — Not Done

I don't see why setting `hasSideEffects` is necessary and setting `mayStore` is not enough, could you please elaborate a bit on this?

evandro — Not Done

`hasSideEffects` is not necessary, since `mayStore` is indeed enough. The purpose of `hasSideEffects` is as a catch all when the rest of the instruction attributes miss some side effect.

✔ Done

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

There is no default value for hasSideEffects, mayLoad, and mayStore in `Instruction` class. So, I specify these three values for V instructions. It follows the style in RISCVInstrInfo.td. I agree that we could give these attributes some default values in some base class for V instructions. I will try to refactor it.

fpallares Not Done

I was only referring tot he fact that `hasSideEffects` here changed from `0` to `1` in a recent update of the patch. I don't really think a refactor is necessary here.

```
108 class VUnitStrideStore<RISCVMOP mop, RISCVLSUMOP sumop, RISCVWidth width,
109                        string opcodestr>
110     : RVInstVSU<0b000, mop, sumop, width, (outs),
111                 (ins VRegOp:$vs3, GPR:$rs1, VMaskOp:$vm), opcodestr,
112                 "$vs3, (${rs1})$vm">;
113
114 // store vd, vs3, (rs1), rs2, vm
115 class VStridedStore<RISCVMOP mop, RISCVWidth width, string opcodestr>
116     : RVInstVSS<0b000, mop, width, (outs),
117                 (ins VRegOp:$vs3, GPR:$rs1, GPR:$rs2, VMaskOp:$vm),
118                 opcodestr, "$vs3, (${rs1}), $rs2$vm">;
119
120 // store vd, vs3, (rs1), vs2, vm
121 class VIndexedStore<RISCVMOP mop, RISCVWidth width, string opcodestr>
122     : RVInstVSX<0b000, mop, width, (outs),
123                 (ins VRegOp:$vs3, GPR:$rs1, VRegOp:$vs2, VMaskOp:$vm),
124                 opcodestr, "$vs3, (${rs1}), $vs2$vm">;
125
126 // vs<nf>r.v vd, (rs1)
127 class VWholeStore<bits<3> nf, string opcodestr>
128     : RVInstVSU<nf, MOPSTUnitStride, SUMOPUnitStrideWholeReg,
129                 LSWidthVSEW, (outs), (ins VRegOp:$vs3, GPR:$rs1),
130                 opcodestr, "$vs3, (${rs1})"> {
131   let vm = 1;
132   let Uses = [];
133 }
134 }
135
136 let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in {
137 // op vd, vs2, vs1, vm
138 class VALUVV<bits<6> funct6, RISCVVFormat opv, string opcodestr>
139     : RVInstVV<funct6, opv, (outs VRegOp:$vd),
140                 (ins VRegOp:$vs2, VRegOp:$vs1, VMaskOp:$vm),
141                 opcodestr, "$vd, $vs2, $vs1$vm">;
142
143 // op vd, vs2, vs1, v0 (without mask, use v0 as carry input)
144 class VALUmVV<bits<6> funct6, RISCVVFormat opv, string opcodestr>
145     : RVInstVV<funct6, opv, (outs VRegOp:$vd),
146                 (ins VRegOp:$vs2, VRegOp:$vs1, VMV0:$v0),
147                 opcodestr, "$vd, $vs2, $vs1, v0"> {
148   let vm = 0;
149 }
150
                                  the order of vs1 and vs2)
                                 ISCVVFormat opv, string opcodestr>
                                 ts VRegOp:$vd),
                                 1, VRegOp:$vs2, VMaskOp:$vm),
155                 opcodestr, "$vd, $vs1, $vs2$vm">;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
156
157 // op vd, vs1, vs2
158 class VALUVVNoVm<bits<6> funct6, RISCVVFormat opv, string opcodestr>
159     : RVInstVV<funct6, opv, (outs VRegOp:$vd),
160                (ins VRegOp:$vs2, VRegOp:$vs1),
161                opcodestr, "$vd, $vs2, $vs1"> {
162   let vm = 1;
163 }
164
165 // op vd, vs2, rs1, vm
166 class VALUVX<bits<6> funct6, RISCVVFormat opv, string opcodestr>
167     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
168                (ins VRegOp:$vs2, GPR:$rs1, VMaskOp:$vm),
169                opcodestr, "$vd, $vs2, $rs1$vm">;
170
171 // op vd, vs2, rs1, v0 (without mask, use v0 as carry input)
172 class VALUmVX<bits<6> funct6, RISCVVFormat opv, string opcodestr>
173     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
174                (ins VRegOp:$vs2, GPR:$rs1, VMV0:$v0),
175                opcodestr, "$vd, $vs2, $rs1, v0"> {
176   let vm = 0;
177 }
178
179 // op vd, rs1, vs2, vm (reverse the order of rs1 and vs2)
180 class VALUrVX<bits<6> funct6, RISCVVFormat opv, string opcodestr>
181     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
182                (ins GPR:$rs1, VRegOp:$vs2, VMaskOp:$vm),
183                opcodestr, "$vd, $rs1, $vs2$vm">;
184
185 // op vd, vs1, vs2
186 class VALUVXNoVm<bits<6> funct6, RISCVVFormat opv, string opcodestr>
187     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
188                (ins VRegOp:$vs2, GPR:$rs1),
189                opcodestr, "$vd, $vs2, $rs1"> {
190   let vm = 1;
191 }
192
193 // op vd, vs2, imm, vm
194 class VALUVI<bits<6> funct6, string opcodestr, Operand optype = simm5>
195     : RVInstIVI<funct6, (outs VRegOp:$vd),
196                (ins VRegOp:$vs2, optype:$imm, VMaskOp:$vm),
197                opcodestr, "$vd, $vs2, $imm$vm">;
198
199 // op vd, vs2, imm, v0 (without mask, use v0 as carry input)
200 class VALUmVI<bits<6> funct6, string opcodestr, Operand optype = simm5>
201     : RVInstIVI<funct6, (outs VRegOp:$vd),
202                (ins VRegOp:$vs2, optype:$imm, VMV0:$v0),
203                opcodestr, "$vd, $vs2, $imm, v0"> {
204   let vm = 0;
205 }
206
207 // op vd, vs2, imm, vm
208 class VALUVINoVm<bits<6> funct6, string opcodestr, Operand optype = simm5>
209     : RVInstIVI<funct6, (outs VRegOp:$vd),
210                (ins VRegOp:$vs2, optype:$imm),
211                opcodestr, "$vd, $vs2, $imm"> {
```



```
216 class VALUVF<bits<6> funct6, RISCVVFormat opv, string opcodestr>
```

```
217     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
218                (ins VRegOp:$vs2, FPR32:$rs1, VMaskOp:$vm),
219                opcodestr, "$vd, $vs2, $rs1$vm">;
220
221 // op vd, rs1, vs2, vm (Float) (with mask, reverse the order of rs1 and vs2)
222 class VALUrVF<bits<6> funct6, RISCVVFormat opv, string opcodestr>
223     : RVInstVX<funct6, opv, (outs VRegOp:$vd),
224                (ins FPR32:$rs1, VRegOp:$vs2, VMaskOp:$vm),
225                opcodestr, "$vd, $rs1, $vs2$vm">;
226
227 // op vd, vs2, vm (use vs1 as instruction encoding)
228 class VALUVs2<bits<6> funct6, bits<5> vs1, RISCVVFormat opv, string opcodestr>
229     : RVInstV<funct6, vs1, opv, (outs VRegOp:$vd),
230               (ins VRegOp:$vs2, VMaskOp:$vm),
231               opcodestr, "$vd, $vs2$vm">;
232 }
233
234 //===----------------------------------------------------------------------===//
235 // Combination of instruction classes.
236 // Use these multiclasses to define instructions more easily.
237 //===----------------------------------------------------------------------===//
238 multiclass VALU_IV_V_X_I<string opcodestr, bits<6> funct6, Operand optype = simm5, string vw
    = "v"> {
239   def V  : VALUVV<funct6, OPIVV, opcodestr # "." # vw # "v">;
240   def X  : VALUVX<funct6, OPIVX, opcodestr # "." # vw # "x">;
241   def I  : VALUVI<funct6, opcodestr # "." # vw # "i", optype>;
242 }
243
244 multiclass VALU_IV_V_X<string opcodestr, bits<6> funct6, string vw = "v"> {
245   def V  : VALUVV<funct6, OPIVV, opcodestr # "." # vw # "v">;
246   def X  : VALUVX<funct6, OPIVX, opcodestr # "." # vw # "x">;
247 }
248
249 multiclass VALUr_IV_V_X<string opcodestr, bits<6> funct6, string vw = "v"> {
250   def V : VALUrVV<funct6, OPIVV, opcodestr # "." # vw # "v">;
251   def X : VALUrVX<funct6, OPIVX, opcodestr # "." # vw # "x">;
252 }
253
254 multiclass VALU_IV_X_I<string opcodestr, bits<6> funct6, Operand optype = simm5, string vw =
    "v"> {
255   def X  : VALUVX<funct6, OPIVX, opcodestr # "." # vw # "x">;
256   def I  : VALUVI<funct6, opcodestr # "." # vw # "i", optype>;
257 }
258
259 multiclass VALU_IV_V<string opcodestr, bits<6> funct6> {
260   def _VS  : VALUVV<funct6, OPIVV, opcodestr # ".vs">;
261 }
262
263 multiclass VALUr_IV_X<string opcodestr, bits<6> funct6, string vw = "v"> {
264   def X : VALUrVX<funct6, OPIVX, opcodestr # "." # vw # "x">;
265 }
266
267 multiclass VALU_MV_V_X<string opcodestr, bits<6> funct6, string vw = "v"> {
268   def V : VALUVV<funct6, OPMVV, opcodestr # "." # vw # "v">;
269   def X  : VALUVX<funct6, OPMVX, opcodestr # "." # vw # "x">;
270 }
271
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
odestr, bits<6> funct6> {
V, opcodestr # ".vs">;
275
```

```
276 multiclass VALU_MV_Mask<string opcodestr, bits<6> funct6, string vm = "v"> {
277   def M : VALUVVNoVm<funct6, OPMVV, opcodestr # "." # vm # "m">;
278 }
279
280 multiclass VALU_MV_X<string opcodestr, bits<6> funct6, string vw = "v"> {
281   def X  : VALUVX<funct6, OPMVX, opcodestr # "." # vw # "x">;
282 }
283
284 multiclass VALUr_MV_V_X<string opcodestr, bits<6> funct6, string vw = "v"> {
285   def V : VALUrVV<funct6, OPMVV, opcodestr # "." # vw # "v">;
286   def X : VALUrVX<funct6, OPMVX, opcodestr # "." # vw # "x">;
287 }
288
289 multiclass VALUr_MV_X<string opcodestr, bits<6> funct6, string vw = "v"> {
290   def X : VALUrVX<funct6, OPMVX, opcodestr # "." # vw # "x">;
291 }
292
293 multiclass VALU_MV_VS2<string opcodestr, bits<6> funct6, bits<5> vs1> {
294   def "" : VALUVs2<funct6, vs1, OPMVV, opcodestr>;
295 }
296
297 multiclass VALUm_IV_V_X_I<string opcodestr, bits<6> funct6> {
298   def VM : VALUmVV<funct6, OPIVV, opcodestr # ".vvm">;
299   def XM : VALUmVX<funct6, OPIVX, opcodestr # ".vxm">;
300   def IM : VALUmVI<funct6, opcodestr # ".vim">;
301 }
302
303 multiclass VALUm_IV_V_X<string opcodestr, bits<6> funct6> {
304   def VM : VALUmVV<funct6, OPIVV, opcodestr # ".vvm">;
305   def XM : VALUmVX<funct6, OPIVX, opcodestr # ".vxm">;
306 }
307
308 multiclass VALUNoVm_IV_V_X_I<string opcodestr, bits<6> funct6, Operand optype = simm5> {
309   def V : VALUVVNoVm<funct6, OPIVV, opcodestr # ".vv">;
310   def X : VALUVXNoVm<funct6, OPIVX, opcodestr # ".vx">;
311   def I : VALUVINoVm<funct6, opcodestr # ".vi", optype>;
312 }
313
314 multiclass VALUNoVm_IV_V_X<string opcodestr, bits<6> funct6> {
315   def V : VALUVVNoVm<funct6, OPIVV, opcodestr # ".vv">;
316   def X : VALUVXNoVm<funct6, OPIVX, opcodestr # ".vx">;
317 }
318
319 multiclass VALU_FV_V_F<string opcodestr, bits<6> funct6, string vw = "v"> {
320   def V : VALUVV<funct6, OPFVV, opcodestr # "." # vw # "v">;
321   def F : VALUVF<funct6, OPFVF, opcodestr # "." # vw # "f">;
322 }
323
324 multiclass VALU_FV_F<string opcodestr, bits<6> funct6, string vw = "v"> {
325   def F : VALUVF<funct6, OPFVF, opcodestr # "." # vw # "f">;
326 }
327
328 multiclass VALUr_FV_V_F<string opcodestr, bits<6> funct6, string vw = "v"> {
329   def V : VALUrVV<funct6, OPFVV, opcodestr # "." # vw # "v">;
330   def F : VALUrVF<funct6, OPFVF, opcodestr # "." # vw # "f">;
331 }
332
333 odestr, bits<6> funct6> {
334 V, opcodestr # ".vs">;
335
336
```

```
337 multiclass VALU_FV_VS2<string opcodestr, bits<6> funct6, bits<5> vs1> {
338   def "" : VALUVs2<funct6, vs1, OPFVV, opcodestr>;
339 }
340
341 //===----------------------------------------------------------------------===//
342 // Instructions
343 //===----------------------------------------------------------------------===//
344
345 let Predicates = [HasStdExtV] in {
346
347 let hasSideEffects = 1, mayLoad = 0, mayStore = 0 in {
348 def VSETVLI : RVInstSetVLi<(outs GPR:$rd), (ins GPR:$rs1, VTypeIOp:$vtypei),
349                            "vsetvli", "$rd, $rs1, $vtypei">;
350
351 def VSETVL : RVInstSetVL<(outs GPR:$rd), (ins GPR:$rs1, GPR:$rs2),
352                          "vsetvl", "$rd, $rs1, $rs2">;
353 }
354
355 // Vector Unit-Stride Instructions
356 def VLB_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStride, LSWidthVByte, "vlb.v">;
357 def VLH_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStride, LSWidthVHalf, "vlh.v">;
358 def VLW_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStride, LSWidthVWord, "vlw.v">;
359
360 def VLBU_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStride, LSWidthVByte, "vlbu.v">;
361 def VLHU_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStride, LSWidthVHalf, "vlhu.v">;
362 def VLWU_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStride, LSWidthVWord, "vlwu.v">;
363
364 def VLE_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStride, LSWidthVSEW, "vle.v">;
365
366 def VLBFF_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStrideFF, LSWidthVByte, "vlbff.v">;
367 def VLHFF_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStrideFF, LSWidthVHalf, "vlhff.v">;
368 def VLWFF_V : VUnitStrideLoad<MOPLDUnitStrideS, LUMOPUnitStrideFF, LSWidthVWord, "vlwff.v">;
369
370 def VLBUFF_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStrideFF, LSWidthVByte,
    "vlbuff.v">;
371 def VLHUFF_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStrideFF, LSWidthVHalf,
    "vlhuff.v">;
372 def VLWUFF_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStrideFF, LSWidthVWord,
    "vlwuff.v">;
373
374 def VLEFF_V : VUnitStrideLoad<MOPLDUnitStrideU, LUMOPUnitStrideFF, LSWidthVSEW, "vleff.v">;
375
376 def VSB_V : VUnitStrideStore<MOPSTUnitStride, SUMOPUnitStride, LSWidthVByte, "vsb.v">;
377 def VSH_V : VUnitStrideStore<MOPSTUnitStride, SUMOPUnitStride, LSWidthVHalf, "vsh.v">;
378 def VSW_V : VUnitStrideStore<MOPSTUnitStride, SUMOPUnitStride, LSWidthVWord, "vsw.v">;
379
380 def VSE_V : VUnitStrideStore<MOPSTUnitStride, SUMOPUnitStride, LSWidthVSEW, "vse.v">;
381
382 // Vector Strided Instructions
383 def VLSB_V : VStridedLoad<MOPLDStridedS, LSWidthVByte, "vlsb.v">;
384 def VLSH_V : VStridedLoad<MOPLDStridedS, LSWidthVHalf, "vlsh.v">;
385 def VLSW_V : VStridedLoad<MOPLDStridedS, LSWidthVWord, "vlsw.v">;
386
387 def VLSBU_V : VStridedLoad<MOPLDStridedU, LSWidthVByte, "vlsbu.v">;
388 def VLSHU_V : VStridedLoad<MOPLDStridedU, LSWidthVHalf, "vlshu.v">;
389 def VLSWU_V : VStridedLoad<MOPLDStridedU, LSWidthVWord, "vlswu.v">;
390
                                StridedU, LSWidthVSEW, "vlse.v">;
                                TStrided, LSWidthVByte, "vssb.v">;
                                TStrided, LSWidthVHalf, "vssh.v">;
395 def VSSW_V : VStridedStore<MOPSTStrided, LSWidthVWord, "vssw.v">;
```

```
396 def VSSE_V : VStridedStore<MOPSTStrided, LSWidthVSEW, "vsse.v">;
397
398 // Vector Indexed Instructions
399 def VLXB_V : VIndexedLoad<MOPLDIndexedS, LSWidthVByte, "vlxb.v">;
400 def VLXH_V : VIndexedLoad<MOPLDIndexedS, LSWidthVHalf, "vlxh.v">;
401 def VLXW_V : VIndexedLoad<MOPLDIndexedS, LSWidthVWord, "vlxw.v">;
402
403 def VLXBU_V : VIndexedLoad<MOPLDIndexedU, LSWidthVByte, "vlxbu.v">;
404 def VLXHU_V : VIndexedLoad<MOPLDIndexedU, LSWidthVHalf, "vlxhu.v">;
405 def VLXWU_V : VIndexedLoad<MOPLDIndexedU, LSWidthVWord, "vlxwu.v">;
406
407 def VLXE_V : VIndexedLoad<MOPLDIndexedU, LSWidthVSEW, "vlxe.v">;
408
409 def VSXB_V : VIndexedStore<MOPSTIndexedOrder, LSWidthVByte, "vsxb.v">;
410 def VSXH_V : VIndexedStore<MOPSTIndexedOrder, LSWidthVHalf, "vsxh.v">;
411 def VSXW_V : VIndexedStore<MOPSTIndexedOrder, LSWidthVWord, "vsxw.v">;
412 def VSXE_V : VIndexedStore<MOPSTIndexedOrder, LSWidthVSEW, "vsxe.v">;
413
414 def VSUXB_V : VIndexedStore<MOPSTIndexedUnOrd, LSWidthVByte, "vsuxb.v">;
415 def VSUXH_V : VIndexedStore<MOPSTIndexedUnOrd, LSWidthVHalf, "vsuxh.v">;
416 def VSUXW_V : VIndexedStore<MOPSTIndexedUnOrd, LSWidthVWord, "vsuxw.v">;
417 def VSUXE_V : VIndexedStore<MOPSTIndexedUnOrd, LSWidthVSEW, "vsuxe.v">;
418
419 def VL1R_V : VWholeLoad<0, "vl1r.v">;
420 def VS1R_V : VWholeStore<0, "vs1r.v">;
```

fpallares Not Done

As of the stable v0.8 release, the whole register vector loads and stores have been restricted to a single register only. You probably just forgot to change this `foreach` since `vl1r.v` and `vs1r.v` are the only instructions being tested.

After this change you may want to revisit whether the `VWholeLoad` and `VWholeStore` multiclasses are still needed.

```
421
422 // Vector Single-Width Integer Add and Subtract
423 defm VADD_V : VALU_IV_V_X_I<"vadd", 0b000000>;
424 defm VSUB_V : VALU_IV_V_X<"vsub", 0b000010>;
425 defm VRSUB_V : VALU_IV_X_I<"vrsub", 0b000011>;
426
```

fpallares Not Done

Whitespace missing before the colon.

```
427 // Vector Widening Integer Add/Subtract
428 // Refer to 11.2 Widening Vector Arithmetic Instructions
429 // The destination vector register group cannot overlap a source vector
430 // register group of a different element width (including the mask register
431 // if masked), otherwise an illegal instruction exception is raised.
432 let Constraints = "@earlyclobber $vd" in {
433 let RVVConstraint = WidenV in {
434 defm VWADDU_V : VALU_MV_V_X<"vwaddu", 0b110000>;
435 defm VWSUBU_V : VALU_MV_V_X<"vwsubu", 0b110010>;
436                                dd", 0b110001>;
437                                ub", 0b110011>;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

fpallares

Not Done

I'd say those instructions need the `@earlyclobber $vd` restriction too. Not because of the first source operand (since it has the same width as the destination), but because of the second (in the case of `VWOP_WV`), or the mask (for `VWOP_WX`).

Quoting section 11.2. Widening Vector Arithmetic Instructions:

> *The destination vector register group cannot overlap a source vector register group of a different element width (including the mask register if masked), otherwise an illegal instruction exception is raised.*

This has the downside that the `earlyclobber` constraint is too coarse and will impose unnecessary restrictions by not allowing the destination to overlap with the first (wide) operand, even when the instruction is used unmasked. Maybe we can use the `earlyclobber` for now, and find a better solution for this in a later patch.

```
439 // Set earlyclobber for following instructions for second and mask operands.
440 // This has the downside that the earlyclobber constraint is too coarse and
441 // will impose unnecessary restrictions by not allowing the destination to
442 // overlap with the first (wide) operand.
443 let RVVConstraint = WidenW in {
444 defm VWADDU_W : VALU_MV_V_X<"vwaddu", 0b110100, "w">;
445 defm VWSUBU_W : VALU_MV_V_X<"vwsubu", 0b110110, "w">;
446 defm VWADD_W : VALU_MV_V_X<"vwadd", 0b110101, "w">;
447 defm VWSUB_W : VALU_MV_V_X<"vwsub", 0b110111, "w">;
448 }
449 }
450
451 // Vector Integer Add-with-Carry / Subtract-with-Borrow Instructions
452 defm VADC_V : VALUm_IV_V_X_I<"vadc", 0b010000>;
453 defm VMADC_V : VALUm_IV_V_X_I<"vmadc", 0b010001>;
454 defm VMADC_V : VALUNoVm_IV_V_X_I<"vmadc", 0b010001>;
455 defm VSBC_V : VALUm_IV_V_X<"vsbc", 0b010010>;
456 defm VMSBC_V : VALUm_IV_V_X<"vmsbc", 0b010011>;
457 defm VMSBC_V : VALUNoVm_IV_V_X<"vmsbc", 0b010011>;
458
459 // Vector Bitwise Logical Instructions
460 defm VAND_V : VALU_IV_V_X_I<"vand", 0b001001>;
```

fpallares Not Done

In section 11.3. Narrowing Vector Arithmetic Instructions we find the following paragraph:

> *The destination vector register group cannot overlap the first source vector register group (specified by vs2). The destination vector register group cannot overlap the mask register if used, unless LMUL=1. If either constraint is violated, an illegal instruction exception is raised.*

From my understanding, this applies to `VNSRL` and `VNSRA`. In that case we need an `@earlyclobber` constraint here.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

✔ Done

In current implementation for MC layer, there is no LMUL information in these instructions. So, if LMUL=1, there is no such restriction for them. I have a downstream patch to extend these instruction definitions for LMUL > 1. Maybe we could keep this MC implementation for LMUL=1 and extend it later. I will add comments for it.

rogfer01 Not Done

Perhaps the wording in the spec is unclear here?

Looks like the following sentence does apply regardless of LMUL.

> *The destination vector register group cannot overlap the first source vector register group (specified by vs2).*

While the sentence that follows it is clear that applies only to LMUL>1

> *The destination vector register group cannot overlap the mask register if used, unless LMUL=1*

You also said that

> *I have a downstream patch to extend these instruction definitions for LMUL > 1*

I'm curious here: did you use a Pseudo for that or somehow avoided the issue that they'd be encoded in the same way?

Thanks!

HsiangKai Author ✔ Done

I misunderstood the statements. Sorry for that. I will add earlyclobber for narrowing instructions. Thanks a lot.

About instruction definitions for LMUL > 1, I enable 'isCodeGenOnly' for these instructions to avoid encoding checking.

```
461 defm VOR_V : VALU_IV_V_X_I<"vor", 0b001010>;
462 defm VXOR_V : VALU_IV_V_X_I<"vxor", 0b001011>;
463
464 // Vector Single-Width Bit Shift Instructions
465 defm VSLL_V : VALU_IV_V_X_I<"vsll", 0b100101, uimm5>;
466 defm VSRL_V : VALU_IV_V_X_I<"vsrl", 0b101000, uimm5>;
467 defm VSRA_V : VALU_IV_V_X_I<"vsra", 0b101001, uimm5>;
468
469 // Vector Narrowing Integer Right Shift Instructions
470 // Refer to 11.3. Narrowing Vector Arithmetic Instructions
471 // The destination vector register group cannot overlap the first source
                                    fied by vs2). The destination vector register
                                    k register if used, unless LMUL=1.
                                    r $vd", RVVConstraint = Narrow in {
                                    nsrl", 0b101100, uimm5, "w">;
476 defm VNSRA_W : VALU_IV_V_X_I<"vnsra", 0b101101, uimm5, "w">;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
477 }
478
479 // Vector Integer Comparison Instructions
480 defm VMSEQ_V : VALU_IV_V_X_I<"vmseq", 0b011000>;
```

> **fpallares** — Not Done
>
> Whitespace missing before the colon.

```
481 defm VMSNE_V : VALU_IV_V_X_I<"vmsne", 0b011001>;
482 defm VMSLTU_V : VALU_IV_V_X<"vmsltu", 0b011010>;
483 defm VMSLT_V : VALU_IV_V_X<"vmslt", 0b011011>;
484 defm VMSLEU_V : VALU_IV_V_X_I<"vmsleu", 0b011100>;
485 defm VMSLE_V : VALU_IV_V_X_I<"vmsle", 0b011101>;
486 defm VMSGTU_V : VALU_IV_X_I<"vmsgtu", 0b011110>;
487 defm VMSGT_V : VALU_IV_X_I<"vmsgt", 0b011111>;
488
489 // Vector Integer Min/Max Instructions
490 defm VMINU_V : VALU_IV_V_X<"vminu", 0b000100>;
491 defm VMIN_V : VALU_IV_V_X<"vmin", 0b000101>;
492 defm VMAXU_V : VALU_IV_V_X<"vmaxu", 0b000110>;
493 defm VMAX_V : VALU_IV_V_X<"vmax", 0b000111>;
494
495 // Vector Single-Width Integer Multiply Instructions
496 defm VMUL_V : VALU_MV_V_X<"vmul", 0b100101>;
497 defm VMULH_V : VALU_MV_V_X<"vmulh", 0b100111>;
498 defm VMULHU_V : VALU_MV_V_X<"vmulhu", 0b100100>;
499 defm VMULHSU_V : VALU_MV_V_X<"vmulhsu", 0b100110>;
500
501 // Vector Integer Divide Instructions
502 defm VDIVU_V : VALU_MV_V_X<"vdivu", 0b100000>;
503 defm VDIV_V : VALU_MV_V_X<"vdiv", 0b100001>;
504 defm VREMU_V : VALU_MV_V_X<"vremu", 0b100010>;
505 defm VREM_V : VALU_MV_V_X<"vrem", 0b100011>;
506
507 // Vector Widening Integer Multiply Instructions
508 let Constraints = "@earlyclobber $vd", RVVConstraint = WidenV in {
509 defm VWMUL_V : VALU_MV_V_X<"vwmul", 0b111011>;
510 defm VWMULU_V : VALU_MV_V_X<"vwmulu", 0b111000>;
511 defm VWMULSU_V : VALU_MV_V_X<"vwmulsu", 0b111010>;
```

> **fpallares** — Done
>
> I wonder if this should be `VMERGE_V`, so that the defined records are `VMERGE_VVM`, `VMERGE_VXM`, `VMERGE_VIM`.
>
> Currently these records are defined `VMERGE_VM`, `VMERGE_XM` and `VMERGE_IM`.

```
512 }
513
514 // Vector Single-Width Integer Multiply-Add Instructions
515 defm VMACC_V : VALUr_MV_V_X<"vmacc", 0b101101>;
516 defm VNMSAC_V : VALUr_MV_V_X<"vnmsac", 0b101111>;
517 defm VMADD_V : VALUr_MV_V_X<"vmadd", 0b101001>;
518 defm VNMSUB_V : VALUr_MV_V_X<"vnmsub", 0b101011>;
```



```
iply-Add Instructions
r $vd", RVVConstraint = WidenV in {
vwmaccu", 0b111100>;
523 defm VWMACC_V : VALUr_MV_V_X<"vwmacc", 0b111101>;
```

```
524 defm VWMACCSU_V : VALUr_MV_V_X<"vwmaccsu", 0b111111>;
525 defm VWMACCUS_V : VALUr_MV_X<"vwmaccus", 0b111110>;
526 }
527
528 // Vector Integer Merge Instructions
529 defm VMERGE_V : VALUm_IV_V_X_I<"vmerge", 0b010111>;
530
531 // Vector Integer Move Instructions
532 let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in {
533 // op vd, vs1
534 def VMV_V_V : RVInstVV<0b010111, OPIVV, (outs VRegOp:$vd),
535                        (ins VRegOp:$vs1), "vmv.v.v", "$vd, $vs1"> {
```

evandro — Not Done

Shouldn't these have `vxsat` in `Defs`?

HsiangKai Author — ✔ Done

It should have no impact on CodeGen. It seems no instructions to use vxsat.

evandro — Not Done

For the sake of completeness.

```
536   let vs2 = 0;
537   let vm = 1;
538 }
539 // op vd, rs1
540 def VMV_V_X : RVInstVX<0b010111, OPIVX, (outs VRegOp:$vd),
541                        (ins GPR:$rs1), "vmv.v.x", "$vd, $rs1"> {
```

evandro — Not Done

Shouldn't these have `vxrm` in `Uses`?

```
542   let vs2 = 0;
543   let vm = 1;
544 }
545 // op vd, imm
546 def VMV_V_I : RVInstIVI<0b010111, (outs VRegOp:$vd),
547                         (ins simm5:$imm), "vmv.v.i", "$vd, $imm"> {
```

evandro — Not Done

Shouldn't this hace `vxrm` in `Uses` and `vxsat` in `Defs`?

```
548   let vs2 = 0;
549   let vm = 1;
550 }
```

evandro — Not Done

…ension?

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
553 // Vector Fixed-Point Arithmetic Instructions
554 defm VSADDU_V : VALU_IV_V_X_I<"vsaddu", 0b100000>;
555 defm VSADD_V : VALU_IV_V_X_I<"vsadd", 0b100001>;
556 defm VSSUBU_V : VALU_IV_V_X<"vssubu", 0b100010>;
```

evandro Not Done

Shouldn't this hace `vxrm` in `Uses` ?

```
557 defm VSSUB_V : VALU_IV_V_X<"vssub", 0b100011>;
```

fpallares Not Done

I'd say overlap restrictions on narrowing instructions apply to `VNCLIP` and `VNCLIPU` too.

```
558
559 // Vector Single-Width Averaging Add and Subtract
560 defm VAADDU_V : VALU_MV_V_X<"vaaddu", 0b001000>;
```

evandro Not Done

Shouldn't this hace `vxrm` in `Uses` and `vxsat` in `Defs` ?

```
561 defm VAADD_V : VALU_MV_V_X<"vaadd", 0b001001>;
562 defm VASUBU_V : VALU_MV_V_X<"vasubu", 0b001010>;
563 defm VASUB_V : VALU_MV_V_X<"vasub", 0b001011>;
564
565 // Vector Single-Width Fractional Multiply with Rounding and Saturation
566 defm VSMUL_V : VALU_IV_V_X<"vsmul", 0b100111>;
567
568 // Vector Single-Width Scaling Shift Instructions
569 defm VSSRL_V : VALU_IV_V_X_I<"vssrl", 0b101010, uimm5>;
570 defm VSSRA_V : VALU_IV_V_X_I<"vssra", 0b101011, uimm5>;
571
```

fpallares Not Done

As discussed in another inline comment above (for `VWADD_W` and such), there is an overlap restriction for these instructions too.

```
572 // Vector Narrowing Fixed-Point Clip Instructions
573 let Constraints = "@earlyclobber $vd", RVVConstraint = Narrow in {
574 defm VNCLIPU_W : VALU_IV_V_X_I<"vnclipu", 0b101110, uimm5, "w">;
575 defm VNCLIP_W : VALU_IV_V_X_I<"vnclip", 0b101111, uimm5, "w">;
576 }
577
578 // Vector Single-Width Integer Reduction Instructions
579 defm VREDSUM : VALU_MV_V<"vredsum", 0b000000>;
580 defm VREDMAXU : VALU_MV_V<"vredmaxu", 0b000110>;
581 defm VREDMAX : VALU_MV_V<"vredmax", 0b000111>;
582 defm VREDMINU : VALU_MV_V<"vredminu", 0b000100>;
583 defm VREDMIN : VALU_MV_V<"vredmin", 0b000101>;
584 defm VREDAND : VALU_MV_V<"vredand", 0b000001>;
```

fpallares Not Done

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

ion.

```
585 defm VREDOR : VALU_MV_V<"vredor", 0b000010>;
```

```
586 defm VREDXOR : VALU_MV_V<"vredxor", 0b000011>;
587
588 // Vector Widening Integer Reduction Instructions
589 let Constraints = "@earlyclobber $vd" in {
590 // Set earlyclobber for following instructions for second and mask operands.
591 // This has the downside that the earlyclobber constraint is too coarse and
592 // will impose unnecessary restrictions by not allowing the destination to
593 // overlap with the first (wide) operand.
594 defm VWREDSUMU : VALU_IV_V<"vwredsumu", 0b110000>;
595 defm VWREDSUM : VALU_IV_V<"vwredsum", 0b110001>;
596 }
597
598 // Vector Mask-Register Logical Instructions
599 defm VMAND_M : VALU_MV_Mask<"vmand", 0b011001, "m">;
600 defm VMNAND_M : VALU_MV_Mask<"vmnand", 0b011101, "m">;
601 defm VMANDNOT_M : VALU_MV_Mask<"vmandnot", 0b011000, "m">;
602 defm VMXOR_M : VALU_MV_Mask<"vmxor", 0b011011, "m">;
603 defm VMOR_M : VALU_MV_Mask<"vmor", 0b011010, "m">;
604 defm VMNOR_M : VALU_MV_Mask<"vmnor", 0b011110, "m">;
605 defm VMORNOT_M : VALU_MV_Mask<"vmornot", 0b011100, "m">;
606 defm VMXNOR_M : VALU_MV_Mask<"vmxnor", 0b011111, "m">;
607
608 let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in {
609 // Vector mask population count vpopc
610 def VPOPC_M : RVInstV<0b010000, 0b10000, OPMVV, (outs GPR:$vd),
611                       (ins VRegOp:$vs2, VMaskOp:$vm),
612                       "vpopc.m", "$vd, $vs2$vm">;
613
614 // vfirst find-first-set mask bit
615 def VFIRST_M : RVInstV<0b010000, 0b10001, OPMVV, (outs GPR:$vd),
616                       (ins VRegOp:$vs2, VMaskOp:$vm),
617                       "vfirst.m", "$vd, $vs2$vm">;
618 }
619
620 // vmsbf.m set-before-first mask bit
621 defm VMSBF_M : VALU_MV_VS2<"vmsbf.m", 0b010100, 0b00001>;
622
623 // vmsif.m set-including-first mask bit
624 defm VMSIF_M : VALU_MV_VS2<"vmsif.m", 0b010100, 0b00011>;
625
626 // vmsof.m set-only-first mask bit
627 defm VMSOF_M : VALU_MV_VS2<"vmsof.m", 0b010100, 0b00010>;
628
629 // Vector Iota Instruction
630 let Constraints = "@earlyclobber $vd", RVVConstraint = Iota in {
631 defm VIOTA_M : VALU_MV_VS2<"viota.m", 0b010100, 0b10000>;
632 }
633
634 // Vector Element Index Instruction
635 let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in {
636 def VID_V : RVInstV<0b010100, 0b10001, OPMVV, (outs VRegOp:$vd),
637                     (ins VMaskOp:$vm), "vid.v", "$vd$vm"> {
638   let vs2 = 0;
639 }
640
641 // Integer Scalar Move Instructions
                                                  0b00000, OPMVV, (outs GPR:$vd),
                                                Op:$vs2), "vmv.x.s", "$vd, $vs2">;
                                                , 0b00000, OPMVX, (outs VRegOp:$vd),
646                       (ins GPR:$rs1), "vmv.s.x", "$vd, $rs1">;
```

```
647
648 }
649
650 foreach nf = [1, 2, 4, 8] in {
651   def VMV#nf#R_V  : RVInstV<0b100111, !add(nf, -1), OPIVI, (outs VRegOp:$vd),
652                             (ins VRegOp:$vs2), "vmv" # nf # "r.v",
653                             "$vd, $vs2"> {
654     let Uses = [];
655     let vm = 1;
656   }
657 }
658 }
659
660 // Vector Slide Instructions
661 let Constraints = "@earlyclobber $vd", RVVConstraint = SlideUp in {
```

fpallares — Not Done

I believe we are also missing the `@earlyclobber $vd` overlap constraints for these narrowing conversions.

From section 14.17. Narrowing Floating-Point/Integer Type-Convert Instructions:

> *These instructions have the same constraints on vector register overlap as other narrowing instructions (see Narrowing Vector Arithmetic Instructions).*

```
662 defm VSLIDEUP_V : VALU_IV_X_I<"vslideup", 0b001110, uimm5>;
663 }
664 defm VSLIDEDOWN_V : VALU_IV_X_I<"vslidedown", 0b001111, uimm5>;
665
666 let Constraints = "@earlyclobber $vd", RVVConstraint = SlideUp in {
667 defm VSLIDE1UP_V : VALU_MV_X<"vslide1up", 0b001110>;
668 }
669 defm VSLIDE1DOWN_V : VALU_MV_X<"vslide1down", 0b001111>;
670
671 // Vector Register Gather Instruction
672 let Constraints = "@earlyclobber $vd", RVVConstraint = Vrgather in {
673 defm VRGATHER_V : VALU_IV_V_X_I<"vrgather", 0b001100, uimm5>;
674 }
675
```

fpallares — Not Done

If I'm not mistaken, those instructions should have the `@earlyclobber $vd` constraint too to avoid overlaps.

```
676 // Vector Compress Instruction
677 let Constraints = "@earlyclobber $vd", RVVConstraint = Vcompress in {
678 defm VCOMPRESS_V : VALU_MV_Mask<"vcompress", 0b010111>;
679 }
680 } // Predicates = [HasStdExtV]
681
682 let Predicates = [HasStdExtF, HasStdExtV] in {
683 // Vector Single-Width Floating-Point Add/Subtract Instructions
                                dd", 0b000000>;
                                ub", 0b000010>;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

Not Done

Ditto.

```
686 defm VFRSUB_V : VALU_FV_F<"vfrsub", 0b100111>;
687
688 // Vector Widening Floating-Point Add/Subtract Instructions
689 let Constraints = "@earlyclobber $vd" in {
690 let RVVConstraint = WidenV in {
691 defm VFWADD_V : VALU_FV_V_F<"vfwadd", 0b110000>;
692 defm VFWSUB_V : VALU_FV_V_F<"vfwsub", 0b110010>;
693 }
694 // Set earlyclobber for following instructions for second and mask operands.
695 // This has the downside that the earlyclobber constraint is too coarse and
696 // will impose unnecessary restrictions by not allowing the destination to
697 // overlap with the first (wide) operand.
698 let RVVConstraint = WidenW in {
699 defm VFWADD_W : VALU_FV_V_F<"vfwadd", 0b110100, "w">;
700 defm VFWSUB_W : VALU_FV_V_F<"vfwsub", 0b110110, "w">;
701 }
702 }
703
704 // Vector Single-Width Floating-Point Multiply/Divide Instructions
705 defm VFMUL_V : VALU_FV_V_F<"vfmul", 0b100100>;
706 defm VFDIV_V : VALU_FV_V_F<"vfdiv", 0b100000>;
707 defm VFRDIV_V : VALU_FV_F<"vfrdiv", 0b100001>;
708
709 // Vector Widening Floating-Point Multiply
710 let Constraints = "@earlyclobber $vd", RVVConstraint = WidenV in {
711 defm VFWMUL_V : VALU_FV_V_F<"vfwmul", 0b111000>;
712 }
713
714 // Vector Single-Width Floating-Point Fused Multiply-Add Instructions
715 defm VFMACC_V : VALUr_FV_V_F<"vfmacc", 0b101100>;
716 defm VFNMACC_V : VALUr_FV_V_F<"vfnmacc", 0b101101>;
717 defm VFMSAC_V : VALUr_FV_V_F<"vfmsac", 0b101110>;
718 defm VFNMSAC_V : VALUr_FV_V_F<"vfnmsac", 0b101111>;
719 defm VFMADD_V : VALUr_FV_V_F<"vfmadd", 0b101000>;
720 defm VFNMADD_V : VALUr_FV_V_F<"vfnmadd", 0b101001>;
721 defm VFMSUB_V : VALUr_FV_V_F<"vfmsub", 0b101010>;
722 defm VFNMSUB_V : VALUr_FV_V_F<"vfnmsub", 0b101011>;
723
724 // Vector Widening Floating-Point Fused Multiply-Add Instructions
725 let Constraints = "@earlyclobber $vd", RVVConstraint = WidenV in {
726 defm VFWMACC_V : VALUr_FV_V_F<"vfwmacc", 0b111100>;
727 defm VFWNMACC_V : VALUr_FV_V_F<"vfwnmacc", 0b111101>;
728 defm VFWMSAC_V : VALUr_FV_V_F<"vfwmsac", 0b111110>;
729 defm VFWNMSAC_V : VALUr_FV_V_F<"vfwnmsac", 0b111111>;
730 }
731
732 // Vector Floating-Point Square-Root Instruction
733 defm VFSQRT_V : VALU_FV_VS2<"vfsqrt.v", 0b100011, 0b00000>;
734
735 // Vector Floating-Point MIN/MAX Instructions
736 defm VFMIN_V : VALU_FV_V_F<"vfmin", 0b000100>;
737 defm VFMAX_V : VALU_FV_V_F<"vfmax", 0b000110>;
738
739 // Vector Floating-Point Sign-Injection Instructions
                                    sgnj", 0b001000>;
                                    fsgnjn", 0b001001>;
                                    fsgnjx", 0b001010>;
744 // Vector Floating-Point Compare Instructions
```

```
745 defm VMFEQ_V : VALU_FV_V_F<"vmfeq", 0b011000>;
746 defm VMFNE_V : VALU_FV_V_F<"vmfne", 0b011100>;
747 defm VMFLT_V : VALU_FV_V_F<"vmflt", 0b011011>;
748 defm VMFLE_V : VALU_FV_V_F<"vmfle", 0b011001>;
749 defm VMFGT_V : VALU_FV_F<"vmfgt", 0b011101>;
750 defm VMFGE_V : VALU_FV_F<"vmfge", 0b011111>;
751
752 // Vector Floating-Point Classify Instruction
753 defm VFCLASS_V : VALU_FV_VS2<"vfclass.v", 0b100011, 0b10000>;
754
755 let hasSideEffects = 0, mayLoad = 0, mayStore = 0 in {
756 // Vector Floating-Point Merge Instruction
757 def VFMERGE_VFM : RVInstVX<0b010111, OPFVF, (outs VRegOp:$vd),
758                            (ins VRegOp:$vs2, FPR32:$rs1, VMV0:$v0),
759                            "vfmerge.vfm", "$vd, $vs2, $rs1, v0"> {
760   let vm = 0;
761 }
762
763 // Vector Floating-Point Move Instruction
764 def VFMV_V_F : RVInstVX<0b010111, OPFVF, (outs VRegOp:$vd),
765                         (ins FPR32:$rs1), "vfmv.v.f", "$vd, $rs1"> {
766   let vs2 = 0;
767   let vm = 1;
768 }
769 }
770
771 // Single-Width Floating-Point/Integer Type-Convert Instructions
772 defm VFCVT_XU_F_V : VALU_FV_VS2<"vfcvt.xu.f.v", 0b100010, 0b00000>;
773 defm VFCVT_X_F_V : VALU_FV_VS2<"vfcvt.x.f.v", 0b100010, 0b00001>;
774 defm VFCVT_F_XU_V : VALU_FV_VS2<"vfcvt.f.xu.v", 0b100010, 0b00010>;
775 defm VFCVT_F_X_V : VALU_FV_VS2<"vfcvt.f.x.v", 0b100010, 0b00011>;
776
777 // Widening Floating-Point/Integer Type-Convert Instructions
778 let Constraints = "@earlyclobber $vd", RVVConstraint = WidenCvt in {
779 defm VFWCVT_XU_F_V : VALU_FV_VS2<"vfwcvt.xu.f.v", 0b100010, 0b01000>;
780 defm VFWCVT_X_F_V : VALU_FV_VS2<"vfwcvt.x.f.v", 0b100010, 0b01001>;
781 defm VFWCVT_F_XU_V : VALU_FV_VS2<"vfwcvt.f.xu.v", 0b100010, 0b01010>;
782 defm VFWCVT_F_X_V : VALU_FV_VS2<"vfwcvt.f.x.v", 0b100010, 0b01011>;
783 defm VFWCVT_F_F_V : VALU_FV_VS2<"vfwcvt.f.f.v", 0b100010, 0b01100>;
784 }
785
786 // Narrowing Floating-Point/Integer Type-Convert Instructions
787 let Constraints = "@earlyclobber $vd", RVVConstraint = Narrow in {
788 defm VFNCVT_XU_F_W : VALU_FV_VS2<"vfncvt.xu.f.w", 0b100010, 0b10000>;
789 defm VFNCVT_X_F_W : VALU_FV_VS2<"vfncvt.x.f.w", 0b100010, 0b10001>;
790 defm VFNCVT_F_XU_W : VALU_FV_VS2<"vfncvt.f.xu.w", 0b100010, 0b10010>;
791 defm VFNCVT_F_X_W : VALU_FV_VS2<"vfncvt.f.x.w", 0b100010, 0b10011>;
792 defm VFNCVT_F_F_W : VALU_FV_VS2<"vfncvt.f.f.w", 0b100010, 0b10100>;
793 defm VFNCVT_ROD_F_F_W : VALU_FV_VS2<"vfncvt.rod.f.f.w", 0b100010, 0b10101>;
794 }
795
796 // Vector Single-Width Floating-Point Reduction Instructions
797 defm VFREDOSUM : VALU_FV_V<"vfredosum", 0b000011>;
798 defm VFREDSUM : VALU_FV_V<"vfredsum", 0b000001>;
799 defm VFREDMAX : VALU_FV_V<"vfredmax", 0b000111>;
800 defm VFREDMIN : VALU_FV_V<"vfredmin", 0b000101>;
801
802                                    nt Reduction Instructions
803                                    r $vd" in {
804                                    ng instructions for second and mask operands.
805 // This has the downside that the earlyclobber constraint is too coarse and
```

```
806 // will impose unnecessary restrictions by not allowing the destination to
807 // overlap with the first (wide) operand.
808 defm VFWREDOSUM : VALU_FV_V<"vfwredosum", 0b110011>;
809 defm VFWREDSUM : VALU_FV_V<"vfwredsum", 0b110001>;
810 }
811
812 let hasSideEffects = 0, mayLoad = 0, mayStore = 0, vm = 1 in {
813 // Floating-Point Scalar Move Instructions
814 def VFMV_F_S : RVInstV<0b010000, 0b00000, OPFVV, (outs FPR32:$vd),
815                        (ins VRegOp:$vs2), "vfmv.f.s", "$vd, $vs2">;
816 def VFMV_S_F : RVInstV2<0b010000, 0b00000, OPFVF, (outs VRegOp:$vd),
817                        (ins FPR32:$rs1), "vfmv.s.f", "$vd, $rs1">;
818
819 }
820 }  // Predicates = [HasStdExtF, HasStdExtV]
```

llvm/lib/Target/RISCV/RISCVRegisterInfo.td

View Options

Show All 27 Lines

```
28  28 class RISCVReg64<RISCVReg32 subreg> : Register<""> {
29  29   let HWEncoding{4-0} = subreg.HWEncoding{4-0};
30  30   let SubRegs = [subreg];
31  31   let SubRegIndices = [sub_32];
32  32   let AsmName = subreg.AsmName;
33  33   let AltNames = subreg.AltNames;
34  34 }
35  35
    36 class RISCVRegWithSubRegs<bits<5> Enc, string n, list<Register> subregs,
    37                           list<string> alt = []>
    38       : RegisterWithSubRegs<n, subregs> {
    39   let HWEncoding{4-0} = Enc;
    40   let AltNames = alt;
    41 }
    42
36  43 def ABIRegAltName : RegAltNameIndex;
    44
    45 def sub_vrm2    : SubRegIndex<64, -1>;
    46 def sub_vrm2_hi : SubRegIndex<64, -1>;
    47 def sub_vrm4    : SubRegIndex<128, -1>;
    48 def sub_vrm4_hi : SubRegIndex<128, -1>;
    49 def sub_vrm8    : SubRegIndex<256, -1>;
    50 def sub_vrm8_hi : SubRegIndex<256, -1>;
```

fpallares — Not Done

My understanding is that here you specify minimum sizes (at least we will have one element of size `ELEN`). As they stand, they are consistent with `ELEN=64`. So far it is unclear to me what sizes we would want to use if we also want to support `ELEN=32`. I imagine we could reuse `HwMode` or something similar.

As far as the offsets are concerned I'd suggest setting them to `-1` because we don't really know the offset where the `hi` part starts.

That said, you already mentioned that types are not that important in this patch.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
39  53 // Integer registers
```

```
40  54  // CostPerUse is set higher for registers that may not be compressible as they
41  55  // are not part of GPRC, the most restrictive register class used by the
42  56  // compressed instruction set. This will influence the greedy register
43  57  // allocator to reduce the use of registers that can't be encoded in 16 bit
44  58  // instructions. This affects register allocation even when compressed
```

▲ Show 20 Lines • Show All 183 Lines • ▼ Show 20 Line(s)

```
228 242     (sequence "F%u_D", 8, 9),
229 243     (sequence "F%u_D", 18, 27)
230 244 )>;
231 245
232 246 def FPR64C : RegisterClass<"RISCV", [f64], 64, (add
233 247   (sequence "F%u_D", 10, 15),
234 248   (sequence "F%u_D", 8, 9)
235 249 )>;
    250
    251 // Vector registers
    252 let RegAltNameIndices = [ABIRegAltName] in {
```

rogfer01 Not Done

Would it help to simplify this as

```
// Define V0-V31
let RegAltNameIndices = [ABIRegAltName] in {
   foreach Index = 0-31 in {
      def V#Index : RISCVReg<Index, "v"#Index, ["v"#Index]>,
       DwarfRegNum<[!add(#Index, 64)]>;
   }
}
```

```
    253   foreach Index = 0-31 in {
    254     def V#Index : RISCVReg<Index, "v"#Index, ["v"#Index]>, DwarfRegNum<[!add(Index, 64)]>;
    255   }
    256
    257   foreach Index = [0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22,
    258                    24, 26, 28, 30] in {
    259     def V#Index#M2 : RISCVRegWithSubRegs<Index, "v"#Index,
    260                         [!cast<Register>("V"#Index),
    261                          !cast<Register>("V"#!add(Index, 1))],
    262                         ["v"#Index]>,
    263                       DwarfRegAlias<!cast<Register>("V"#Index)> {
    264       let SubRegIndices = [sub_vrm2, sub_vrm2_hi];
    265     }
    266   }
    267
    268   foreach Index = [0, 4, 8, 12, 16, 20, 24, 28] in {
    269     def V#Index#M4 : RISCVRegWithSubRegs<Index, "v"#Index,
    270                         [!cast<Register>("V"#Index#"M2"),
    271                          !cast<Register>("V"#!add(Index, 2)#"M2")],
    272                         ["v"#Index]>,
    273                       DwarfRegAlias<!cast<Register>("V"#Index)> {
    274       let SubRegIndices = [sub_vrm4, sub_vrm4_hi];
    275     }
    276   }
    277
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
                                ] in {
                              thSubRegs<Index, "v"#Index,
                              egister>("V"#Index#"M4"),
                              egister>("V"#!add(Index, 4)#"M4")],
    282                         ["v"#Index]>,
```

```
283                     DwarfRegAlias<!cast<Register>("V"#Index)> {
284       let SubRegIndices = [sub_vrm8, sub_vrm8_hi];
285     }
286   }
287
288   def VTYPE  : RISCVReg<0, "vtype", ["vtype"]>;
289   def VL     : RISCVReg<0, "vl", ["vl"]>;
290 }
291
292 class RegisterTypes<list<ValueType> reg_types> {
293   list<ValueType> types = reg_types;
294 }
295
```

rogfer01 Not Done

What does the P in VPR stand for?

```
296 def VDataVT : RegisterTypes<[nxv8i8, nxv16i8, nxv32i8,
297                               nxv4i16, nxv8i16, nxv16i16, nxv32i16,
298                               nxv2i32, nxv4i32, nxv8i32, nxv16i32,
299                               nxv1i64, nxv2i64, nxv4i64, nxv8i64]>;
300
301 // The order of registers represents the preferred allocation sequence,
302 // meaning caller-save regs are listed before callee-save.
303 def VR : RegisterClass<"RISCV", [nxv8i8, nxv4i16, nxv2i32, nxv1i64],
304                           64, (add
305     (sequence "V%u", 25, 31),
306     (sequence "V%u", 8, 24),
307     (sequence "V%u", 0, 7)
308   )> {
309   let Size = 64;
310 }
311
312 def VRM2 : RegisterClass<"RISCV", [nxv16i8, nxv8i16, nxv4i32, nxv2i64], 64,
313                           (add V26M2, V28M2, V30M2, V8M2, V10M2, V12M2, V14M2, V16M2,
314                                V18M2, V20M2, V22M2, V24M2, V0M2, V2M2, V4M2, V6M2)> {
315   let Size = 128;
316 }
317
318 def VRM4 : RegisterClass<"RISCV", [nxv32i8, nxv16i16, nxv8i32, nxv4i64], 64,
319                           (add V28M4, V8M4, V12M4, V16M4, V20M4, V24M4, V0M4, V4M4)> {
320   let Size = 256;
321 }
322
323 def VRM8 : RegisterClass<"RISCV", [nxv32i16, nxv16i32, nxv8i64], 64,
324                           (add V8M8, V16M8, V24M8, V0M8)> {
325   let Size = 512;
326 }
327
328 def VMaskVT : RegisterTypes<[nxv1i1, nxv2i1, nxv4i1, nxv8i1, nxv16i1, nxv32i1]>;
329
330 def VM : RegisterClass<"RISCV", VMaskVT.types, 64, (add
331     (sequence "V%u", 25, 31),
332     (sequence "V%u", 8, 24),
333     (sequence "V%u", 0, 7))> {
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
                                ", VMaskVT.types, 64, (add V0)> {
338   let Size = 64;
```

```
339 }
```

llvm/lib/Target/RISCV/RISCVSchedRocket32.td

View Options

Show All 11 Lines

```
12 12
13 13 // Rocket machine model for scheduling and other instruction cost heuristics.
14 14 def Rocket32Model : SchedMachineModel {
15 15   let MicroOpBufferSize = 0; // Explicitly set to zero since Rocket is in-order.
16 16   let IssueWidth = 1;        // 1 micro-ops are dispatched per cycle.
17 17   let LoadLatency = 3;
18 18   let MispredictPenalty = 3;
19 19   let CompleteModel = 1;
   20   let UnsupportedFeatures = [HasStdExtV];
20 21 }
21 22
22 23 //===----------------------------------------------------------------------===//
23 24 // Define each kind of processor resource and number available.
24 25
25 26 // Modeling each pipeline as a ProcResource using the BufferSize = 0 since
26 27 // Rocket is in-order.
27 28
```

▲ Show 20 Lines • Show All 199 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/RISCVSchedRocket64.td

View Options

Show All 10 Lines

```
11 11 // This works with MachineScheduler. See MCSchedule.h for details.
12 12
13 13 // Rocket machine model for scheduling and other instruction cost heuristics.
14 14 def Rocket64Model : SchedMachineModel {
15 15   let MicroOpBufferSize = 0; // Explicitly set to zero since Rocket is in-order.
16 16   let IssueWidth = 1;        // 1 micro-ops are dispatched per cycle.
17 17   let LoadLatency = 3;
18 18   let MispredictPenalty = 3;
   19   let UnsupportedFeatures = [HasStdExtV];
19 20 }
20 21
21 22 //===----------------------------------------------------------------------===//
22 23 // Define each kind of processor resource and number available.
23 24
24 25 // Modeling each pipeline as a ProcResource using the BufferSize = 0 since
25 26 // Rocket is in-order.
26 27
```

▲ Show 20 Lines • Show All 201 Lines • Show Last 20 Lines

llvm/lib/Target/RISCV/RISCVSubtarget.h

View Options

Show First 20 Lines • Show All 44 Lines • ▼ Show 20 Line(s)

```
45 45   bool HasStdExtZbe = false;
46 46   bool HasStdExtZbf = false;
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
51 51   bool HasStdExtZbt = false;
```

```
52  52    bool HasStdExtZbproposedc = false;
    53    bool HasStdExtV = false;
53  54    bool HasRV64 = false;
54  55    bool IsRV32E = false;
55  56    bool EnableLinkerRelax = false;
56  57    bool EnableRVCHintInstrs = false;
57  58    bool EnableSaveRestore = false;
58  59    unsigned XLen = 32;
59  60    MVT XLenVT = MVT::i32;
60  61    RISCVABI::ABI TargetABI = RISCVABI::ABI_Unknown;
```

▲ Show 20 Lines • Show All 44 Lines • ▼ Show 20 Line(s)

```
105 106   bool hasStdExtZbe() const { return HasStdExtZbe; }
106 107   bool hasStdExtZbf() const { return HasStdExtZbf; }
107 108   bool hasStdExtZbm() const { return HasStdExtZbm; }
108 109   bool hasStdExtZbp() const { return HasStdExtZbp; }
109 110   bool hasStdExtZbr() const { return HasStdExtZbr; }
110 111   bool hasStdExtZbs() const { return HasStdExtZbs; }
111 112   bool hasStdExtZbt() const { return HasStdExtZbt; }
112 113   bool hasStdExtZbproposedc() const { return HasStdExtZbproposedc; }
    114   bool hasStdExtV() const { return HasStdExtV; }
113 115   bool is64Bit() const { return HasRV64; }
114 116   bool isRV32E() const { return IsRV32E; }
115 117   bool enableLinkerRelax() const { return EnableLinkerRelax; }
116 118   bool enableRVCHintInstrs() const { return EnableRVCHintInstrs; }
117 119   bool enableSaveRestore() const { return EnableSaveRestore; }
118 120   MVT getXLenVT() const { return XLenVT; }
119 121   unsigned getXLen() const { return XLen; }
120 122   RISCVABI::ABI getTargetABI() const { return TargetABI; }
```

Show All 21 Lines

llvm/lib/Target/RISCV/RISCVSystemOperands.td

View Options

Show First 20 Lines • Show All 341 Lines • ▼ Show 20 Line(s)

```
342 342 def : SysReg<"tdata3", 0x7A3>;
343 343
344 344 //===-----------------------------------------------
345 345 // Debug Mode Registers
346 346 //===-----------------------------------------------
347 347 def : SysReg<"dcsr", 0x7B0>;
348 348 def : SysReg<"dpc", 0x7B1>;
349 349 def : SysReg<"dscratch", 0x7B2>;
    350
    351 //===-----------------------------------------------
    352 // User Vector CSRs
    353 //===-----------------------------------------------
    354 def : SysReg<"vstart", 0x008>;
    355 def : SysReg<"vxsat", 0x009>;
    356 def : SysReg<"vxrm", 0x00A>;
    357 def : SysReg<"vl", 0xC20>;
    358 def : SysReg<"vtype", 0xC21>;
    359 def : SysReg<"vlenb", 0xC22>;
```

llvm/lib/Target/RISCV/Utils/RISCVBaseInfo.h

View Options

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

Show All 39 Lines

```
41  41    InstFormatCL = 12,
```

```
42  42    InstFormatCS = 13,
43  43    InstFormatCA = 14,
44  44    InstFormatCB = 15,
45  45    InstFormatCJ = 16,
46  46    InstFormatOther = 17,
47  47
48        InstFormatMask = 31
    48    InstFormatMask = 31,
49  49  };
50  50
```

evandro Not Done

Why not just 32 ?

```
51  51  // RISC-V Specific Machine Operand Flags
52  52  enum {
53  53    MO_None = 0,
54  54    MO_CALL = 1,
55  55    MO_PLT = 2,
56  56    MO_LO = 3,
57  57    MO_HI = 4,
58  58    MO_PCREL_LO = 5,
```

▲ Show 20 Lines • Show All 164 Lines • Show Last 20 Lines

llvm/test/MC/RISCV/rvv/add.s

View Options

This file was added.

```
1   # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2   # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3   # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4   # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5   # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6   # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7   # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8   # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9   # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11  vadd.vv v8, v4, v20, v0.t
12  # CHECK-INST: vadd.vv v8, v4, v20, v0.t
13  # CHECK-ENCODING: [0x57,0x04,0x4a,0x00]
14  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15  # CHECK-UNKNOWN: 57 04 4a 00 <unknown>
16
17  vadd.vv v8, v4, v20
18  # CHECK-INST: vadd.vv v8, v4, v20
19  # CHECK-ENCODING: [0x57,0x04,0x4a,0x02]
20  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21  # CHECK-UNKNOWN: 57 04 4a 02 <unknown>
22
23  vadd.vx v8, v4, a0, v0.t
24  # CHECK-INST: vadd.vx v8, v4, a0, v0.t
25  # CHECK-ENCODING: [0x57,0x44,0x45,0x00]
                    ires the following: 'V' (Vector Instructions)
                    nknown>
30  # CHECK-INST: vadd.vx v8, v4, a0
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
31  # CHECK-ENCODING: [0x57,0x44,0x45,0x02]
32  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33  # CHECK-UNKNOWN: 57 44 45 02 <unknown>
34
35  vadd.vi v8, v4, 15, v0.t
36  # CHECK-INST: vadd.vi v8, v4, 15, v0.t
37  # CHECK-ENCODING: [0x57,0xb4,0x47,0x00]
38  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39  # CHECK-UNKNOWN: 57 b4 47 00 <unknown>
40
41  vadd.vi v8, v4, 15
42  # CHECK-INST: vadd.vi v8, v4, 15
43  # CHECK-ENCODING: [0x57,0xb4,0x47,0x02]
44  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45  # CHECK-UNKNOWN: 57 b4 47 02 <unknown>
46
47  vwaddu.vv v8, v4, v20, v0.t
48  # CHECK-INST: vwaddu.vv v8, v4, v20, v0.t
49  # CHECK-ENCODING: [0x57,0x24,0x4a,0xc0]
50  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51  # CHECK-UNKNOWN: 57 24 4a c0 <unknown>
52
53  vwaddu.vv v8, v4, v20
54  # CHECK-INST: vwaddu.vv v8, v4, v20
55  # CHECK-ENCODING: [0x57,0x24,0x4a,0xc2]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 24 4a c2 <unknown>
58
59  vwaddu.vx v8, v4, a0, v0.t
60  # CHECK-INST: vwaddu.vx v8, v4, a0, v0.t
61  # CHECK-ENCODING: [0x57,0x64,0x45,0xc0]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 64 45 c0 <unknown>
64
65  vwaddu.vx v8, v4, a0
66  # CHECK-INST: vwaddu.vx v8, v4, a0
67  # CHECK-ENCODING: [0x57,0x64,0x45,0xc2]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 64 45 c2 <unknown>
70
71  vwadd.vv v8, v4, v20, v0.t
72  # CHECK-INST: vwadd.vv v8, v4, v20, v0.t
73  # CHECK-ENCODING: [0x57,0x24,0x4a,0xc4]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 24 4a c4 <unknown>
76
77  vwadd.vv v8, v4, v20
78  # CHECK-INST: vwadd.vv v8, v4, v20
79  # CHECK-ENCODING: [0x57,0x24,0x4a,0xc6]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 24 4a c6 <unknown>
82
83  vwadd.vx v8, v4, a0, v0.t
84  # CHECK-INST: vwadd.vx v8, v4, a0, v0.t
85  # CHECK-ENCODING: [0x57,0x64,0x45,0xc4]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    a0
91  # CHECK-ENCODING: [0x57,0x64,0x45,0xc6]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 64 45 c6 <unknown>
94
95  vwaddu.wv v8, v4, v20, v0.t
96  # CHECK-INST: vwaddu.wv v8, v4, v20, v0.t
97  # CHECK-ENCODING: [0x57,0x24,0x4a,0xd0]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 24 4a d0 <unknown>
100
101 vwaddu.wv v8, v4, v20
102 # CHECK-INST: vwaddu.wv v8, v4, v20
103 # CHECK-ENCODING: [0x57,0x24,0x4a,0xd2]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 24 4a d2 <unknown>
106
107 vwaddu.wx v8, v4, a0, v0.t
108 # CHECK-INST: vwaddu.wx v8, v4, a0, v0.t
109 # CHECK-ENCODING: [0x57,0x64,0x45,0xd0]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 64 45 d0 <unknown>
112
113 vwaddu.wx v8, v4, a0
114 # CHECK-INST: vwaddu.wx v8, v4, a0
115 # CHECK-ENCODING: [0x57,0x64,0x45,0xd2]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 64 45 d2 <unknown>
118
119 vwadd.wv v8, v4, v20, v0.t
120 # CHECK-INST: vwadd.wv v8, v4, v20, v0.t
121 # CHECK-ENCODING: [0x57,0x24,0x4a,0xd4]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 24 4a d4 <unknown>
124
125 vwadd.wv v8, v4, v20
126 # CHECK-INST: vwadd.wv v8, v4, v20
127 # CHECK-ENCODING: [0x57,0x24,0x4a,0xd6]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 24 4a d6 <unknown>
130
131 vwadd.wx v8, v4, a0, v0.t
132 # CHECK-INST: vwadd.wx v8, v4, a0, v0.t
133 # CHECK-ENCODING: [0x57,0x64,0x45,0xd4]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 64 45 d4 <unknown>
136
137 vwadd.wx v8, v4, a0
138 # CHECK-INST: vwadd.wx v8, v4, a0
139 # CHECK-ENCODING: [0x57,0x64,0x45,0xd6]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 57 64 45 d6 <unknown>
142
143 vadc.vvm v8, v4, v20, v0
144 # CHECK-INST: vadc.vvm v8, v4, v20, v0
145 # CHECK-ENCODING: [0x57,0x04,0x4a,0x40]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 04 4a 40 <unknown>

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

a0, v0
45,0x40]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
153 # CHECK-UNKNOWN: 57 44 45 40 <unknown>
154
155 vadc.vim v8, v4, 15, v0
156 # CHECK-INST: vadc.vim v8, v4, 15, v0
157 # CHECK-ENCODING: [0x57,0xb4,0x47,0x40]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 b4 47 40 <unknown>
160
161 vmadc.vvm v8, v4, v20, v0
162 # CHECK-INST: vmadc.vvm v8, v4, v20, v0
163 # CHECK-ENCODING: [0x57,0x04,0x4a,0x44]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 04 4a 44 <unknown>
166
167 vmadc.vxm v8, v4, a0, v0
168 # CHECK-INST: vmadc.vxm v8, v4, a0, v0
169 # CHECK-ENCODING: [0x57,0x44,0x45,0x44]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
171 # CHECK-UNKNOWN: 57 44 45 44 <unknown>
172
173 vmadc.vim v8, v4, 15, v0
174 # CHECK-INST: vmadc.vim v8, v4, 15, v0
175 # CHECK-ENCODING: [0x57,0xb4,0x47,0x44]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
177 # CHECK-UNKNOWN: 57 b4 47 44 <unknown>
178
179 vmadc.vv v8, v4, v20
180 # CHECK-INST: vmadc.vv v8, v4, v20
181 # CHECK-ENCODING: [0x57,0x04,0x4a,0x46]
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 04 4a 46 <unknown>
184
185 vmadc.vx v8, v4, a0
186 # CHECK-INST: vmadc.vx v8, v4, a0
187 # CHECK-ENCODING: [0x57,0x44,0x45,0x46]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 44 45 46 <unknown>
190
191 vmadc.vi v8, v4, 15
192 # CHECK-INST: vmadc.vi v8, v4, 15
193 # CHECK-ENCODING: [0x57,0xb4,0x47,0x46]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 57 b4 47 46 <unknown>
196
197 vsaddu.vv v8, v4, v20, v0.t
198 # CHECK-INST: vsaddu.vv v8, v4, v20, v0.t
199 # CHECK-ENCODING: [0x57,0x04,0x4a,0x80]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 57 04 4a 80 <unknown>
202
203 vsaddu.vv v8, v4, v20
204 # CHECK-INST: vsaddu.vv v8, v4, v20
205 # CHECK-ENCODING: [0x57,0x04,0x4a,0x82]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 57 04 4a 82 <unknown>
208

                                  a0, v0.t
                                  45,0x80]
                                  ires the following: 'V' (Vector Instructions)
213 # CHECK-UNKNOWN: 57 44 45 80 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
214
215 vsaddu.vx v8, v4, a0
216 # CHECK-INST: vsaddu.vx v8, v4, a0
217 # CHECK-ENCODING: [0x57,0x44,0x45,0x82]
218 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
219 # CHECK-UNKNOWN: 57 44 45 82 <unknown>
220
221 vsaddu.vi v8, v4, 15, v0.t
222 # CHECK-INST: vsaddu.vi v8, v4, 15, v0.t
223 # CHECK-ENCODING: [0x57,0xb4,0x47,0x80]
224 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
225 # CHECK-UNKNOWN: 57 b4 47 80 <unknown>
226
227 vsaddu.vi v8, v4, 15
228 # CHECK-INST: vsaddu.vi v8, v4, 15
229 # CHECK-ENCODING: [0x57,0xb4,0x47,0x82]
230 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
231 # CHECK-UNKNOWN: 57 b4 47 82 <unknown>
232
233 vsadd.vv v8, v4, v20, v0.t
234 # CHECK-INST: vsadd.vv v8, v4, v20, v0.t
235 # CHECK-ENCODING: [0x57,0x04,0x4a,0x84]
236 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
237 # CHECK-UNKNOWN: 57 04 4a 84 <unknown>
238
239 vsadd.vv v8, v4, v20
240 # CHECK-INST: vsadd.vv v8, v4, v20
241 # CHECK-ENCODING: [0x57,0x04,0x4a,0x86]
242 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
243 # CHECK-UNKNOWN: 57 04 4a 86 <unknown>
244
245 vsadd.vx v8, v4, a0, v0.t
246 # CHECK-INST: vsadd.vx v8, v4, a0, v0.t
247 # CHECK-ENCODING: [0x57,0x44,0x45,0x84]
248 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
249 # CHECK-UNKNOWN: 57 44 45 84 <unknown>
250
251 vsadd.vx v8, v4, a0
252 # CHECK-INST: vsadd.vx v8, v4, a0
253 # CHECK-ENCODING: [0x57,0x44,0x45,0x86]
254 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
255 # CHECK-UNKNOWN: 57 44 45 86 <unknown>
256
257 vsadd.vi v8, v4, 15, v0.t
258 # CHECK-INST: vsadd.vi v8, v4, 15, v0.t
259 # CHECK-ENCODING: [0x57,0xb4,0x47,0x84]
260 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
261 # CHECK-UNKNOWN: 57 b4 47 84 <unknown>
262
263 vsadd.vi v8, v4, 15
264 # CHECK-INST: vsadd.vi v8, v4, 15
265 # CHECK-ENCODING: [0x57,0xb4,0x47,0x86]
266 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
267 # CHECK-UNKNOWN: 57 b4 47 86 <unknown>
268
269 vaadd.vv v8, v4, v20, v0.t
                                v20, v0.t
                                4a,0x24]
                                ires the following: 'V' (Vector Instructions)
                                nknown>
274
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
275 vaadd.vv v8, v4, v20
276 # CHECK-INST: vaadd.vv v8, v4, v20
277 # CHECK-ENCODING: [0x57,0x24,0x4a,0x26]
278 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
279 # CHECK-UNKNOWN: 57 24 4a 26 <unknown>
280
281 vaadd.vx v8, v4, a0, v0.t
282 # CHECK-INST: vaadd.vx v8, v4, a0, v0.t
283 # CHECK-ENCODING: [0x57,0x64,0x45,0x24]
284 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
285 # CHECK-UNKNOWN: 57 64 45 24 <unknown>
286
287 vaadd.vx v8, v4, a0
288 # CHECK-INST: vaadd.vx v8, v4, a0
289 # CHECK-ENCODING: [0x57,0x64,0x45,0x26]
290 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
291 # CHECK-UNKNOWN: 57 64 45 26 <unknown>
292
293 vaaddu.vv v8, v4, v20, v0.t
294 # CHECK-INST: vaaddu.vv v8, v4, v20, v0.t
295 # CHECK-ENCODING: [0x57,0x24,0x4a,0x20]
296 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
297 # CHECK-UNKNOWN: 57 24 4a 20 <unknown>
298
299 vaaddu.vv v8, v4, v20
300 # CHECK-INST: vaaddu.vv v8, v4, v20
301 # CHECK-ENCODING: [0x57,0x24,0x4a,0x22]
302 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
303 # CHECK-UNKNOWN: 57 24 4a 22 <unknown>
304
305 vaaddu.vx v8, v4, a0, v0.t
306 # CHECK-INST: vaaddu.vx v8, v4, a0, v0.t
307 # CHECK-ENCODING: [0x57,0x64,0x45,0x20]
308 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
309 # CHECK-UNKNOWN: 57 64 45 20 <unknown>
310
311 vaaddu.vx v8, v4, a0
312 # CHECK-INST: vaaddu.vx v8, v4, a0
313 # CHECK-ENCODING: [0x57,0x64,0x45,0x22]
314 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
315 # CHECK-UNKNOWN: 57 64 45 22 <unknown>
```

llvm/test/MC/RISCV/rvv/and.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
20, v0.t
4a,0x24]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
15 # CHECK-UNKNOWN: 57 04 4a 24 <unknown>
16
17 vand.vv v8, v4, v20
18 # CHECK-INST: vand.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x04,0x4a,0x26]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 04 4a 26 <unknown>
22
23 vand.vx v8, v4, a0, v0.t
24 # CHECK-INST: vand.vx v8, v4, a0, v0.t
25 # CHECK-ENCODING: [0x57,0x44,0x45,0x24]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 57 44 45 24 <unknown>
28
29 vand.vx v8, v4, a0
30 # CHECK-INST: vand.vx v8, v4, a0
31 # CHECK-ENCODING: [0x57,0x44,0x45,0x26]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 44 45 26 <unknown>
34
35 vand.vi v8, v4, 15, v0.t
36 # CHECK-INST: vand.vi v8, v4, 15, v0.t
37 # CHECK-ENCODING: [0x57,0xb4,0x47,0x24]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 b4 47 24 <unknown>
40
41 vand.vi v8, v4, 15
42 # CHECK-INST: vand.vi v8, v4, 15
43 # CHECK-ENCODING: [0x57,0xb4,0x47,0x26]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 b4 47 26 <unknown>
```

llvm/test/MC/RISCV/rvv/clip.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vnclipu.wv v8, v4, v20, v0.t
12 # CHECK-INST: vnclipu.wv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x04,0x4a,0xb8]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 04 4a b8 <unknown>
16
17 vnclipu.wv v8, v4, v20
18 # CHECK-INST: vnclipu.wv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x04,0x4a,0xba]
                    ires the following: 'V' (Vector Instructions)
                    nknown>
24 # CHECK-INST: vnclipu.wx v8, v4, a0, v0.t
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
25  # CHECK-ENCODING: [0x57,0x44,0x45,0xb8]
26  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27  # CHECK-UNKNOWN: 57 44 45 b8 <unknown>
28
29  vnclipu.wx v8, v4, a0
30  # CHECK-INST: vnclipu.wx v8, v4, a0
31  # CHECK-ENCODING: [0x57,0x44,0x45,0xba]
32  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33  # CHECK-UNKNOWN: 57 44 45 ba <unknown>
34
35  vnclipu.wi v8, v4, 31, v0.t
36  # CHECK-INST: vnclipu.wi v8, v4, 31, v0.t
37  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xb8]
38  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39  # CHECK-UNKNOWN: 57 b4 4f b8 <unknown>
40
41  vnclipu.wi v8, v4, 31
42  # CHECK-INST: vnclipu.wi v8, v4, 31
43  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xba]
44  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45  # CHECK-UNKNOWN: 57 b4 4f ba <unknown>
46
47  vnclip.wv v8, v4, v20, v0.t
48  # CHECK-INST: vnclip.wv v8, v4, v20, v0.t
49  # CHECK-ENCODING: [0x57,0x04,0x4a,0xbc]
50  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51  # CHECK-UNKNOWN: 57 04 4a bc <unknown>
52
53  vnclip.wv v8, v4, v20
54  # CHECK-INST: vnclip.wv v8, v4, v20
55  # CHECK-ENCODING: [0x57,0x04,0x4a,0xbe]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 04 4a be <unknown>
58
59  vnclip.wx v8, v4, a0, v0.t
60  # CHECK-INST: vnclip.wx v8, v4, a0, v0.t
61  # CHECK-ENCODING: [0x57,0x44,0x45,0xbc]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 44 45 bc <unknown>
64
65  vnclip.wx v8, v4, a0
66  # CHECK-INST: vnclip.wx v8, v4, a0
67  # CHECK-ENCODING: [0x57,0x44,0x45,0xbe]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 44 45 be <unknown>
70
71  vnclip.wi v8, v4, 31, v0.t
72  # CHECK-INST: vnclip.wi v8, v4, 31, v0.t
73  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xbc]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 b4 4f bc <unknown>
76
77  vnclip.wi v8, v4, 31
78  # CHECK-INST: vnclip.wi v8, v4, 31
79  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xbe]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
    nknown>
```

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
# RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:         | llvm-objdump -d --mattr=+experimental-v - \
# RUN:         | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vmseq.vv v8, v4, v20, v0.t
# CHECK-INST: vmseq.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x60]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 60 <unknown>

vmseq.vv v8, v4, v20
# CHECK-INST: vmseq.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x04,0x4a,0x62]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 62 <unknown>

vmseq.vx v8, v4, a0, v0.t
# CHECK-INST: vmseq.vx v8, v4, a0, v0.t
# CHECK-ENCODING: [0x57,0x44,0x45,0x60]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 60 <unknown>

vmseq.vx v8, v4, a0
# CHECK-INST: vmseq.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x44,0x45,0x62]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 62 <unknown>

vmseq.vi v8, v4, 15, v0.t
# CHECK-INST: vmseq.vi v8, v4, 15, v0.t
# CHECK-ENCODING: [0x57,0xb4,0x47,0x60]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 b4 47 60 <unknown>

vmseq.vi v8, v4, 15
# CHECK-INST: vmseq.vi v8, v4, 15
# CHECK-ENCODING: [0x57,0xb4,0x47,0x62]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 b4 47 62 <unknown>

vmsne.vv v8, v4, v20, v0.t
# CHECK-INST: vmsne.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x64]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 64 <unknown>

vmsne.vv v8, v4, v20
# CHECK-INST: vmsne.vv v8, v4, v20
4a,0x66]
ires the following: 'V' (Vector Instructions)
nknown>

vmsne.vx v8, v4, a0, v0.t
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
60  # CHECK-INST: vmsne.vx v8, v4, a0, v0.t
61  # CHECK-ENCODING: [0x57,0x44,0x45,0x64]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 44 45 64 <unknown>
64
65  vmsne.vx v8, v4, a0
66  # CHECK-INST: vmsne.vx v8, v4, a0
67  # CHECK-ENCODING: [0x57,0x44,0x45,0x66]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 44 45 66 <unknown>
70
71  vmsne.vi v8, v4, 15, v0.t
72  # CHECK-INST: vmsne.vi v8, v4, 15, v0.t
73  # CHECK-ENCODING: [0x57,0xb4,0x47,0x64]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 b4 47 64 <unknown>
76
77  vmsne.vi v8, v4, 15
78  # CHECK-INST: vmsne.vi v8, v4, 15
79  # CHECK-ENCODING: [0x57,0xb4,0x47,0x66]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 b4 47 66 <unknown>
82
83  vmsltu.vv v8, v4, v20, v0.t
84  # CHECK-INST: vmsltu.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x04,0x4a,0x68]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 04 4a 68 <unknown>
88
89  vmsltu.vv v8, v4, v20
90  # CHECK-INST: vmsltu.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x04,0x4a,0x6a]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 04 4a 6a <unknown>
94
95  vmsltu.vx v8, v4, a0, v0.t
96  # CHECK-INST: vmsltu.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x44,0x45,0x68]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 44 45 68 <unknown>
100
101 vmsltu.vx v8, v4, a0
102 # CHECK-INST: vmsltu.vx v8, v4, a0
103 # CHECK-ENCODING: [0x57,0x44,0x45,0x6a]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 44 45 6a <unknown>
106
107 vmslt.vv v8, v4, v20, v0.t
108 # CHECK-INST: vmslt.vv v8, v4, v20, v0.t
109 # CHECK-ENCODING: [0x57,0x04,0x4a,0x6c]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 04 4a 6c <unknown>
112
113 vmslt.vv v8, v4, v20
114 # CHECK-INST: vmslt.vv v8, v4, v20
115 # CHECK-ENCODING: [0x57,0x04,0x4a,0x6e]
                  ires the following: 'V' (Vector Instructions)
                  nknown>

120 # CHECK-INST: vmslt.vx v8, v4, a0, v0.t
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
121 # CHECK-ENCODING: [0x57,0x44,0x45,0x6c]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 44 45 6c <unknown>
124
125 vmslt.vx v8, v4, a0
126 # CHECK-INST: vmslt.vx v8, v4, a0
127 # CHECK-ENCODING: [0x57,0x44,0x45,0x6e]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 44 45 6e <unknown>
130
131 vmsleu.vv v8, v4, v20, v0.t
132 # CHECK-INST: vmsleu.vv v8, v4, v20, v0.t
133 # CHECK-ENCODING: [0x57,0x04,0x4a,0x70]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 04 4a 70 <unknown>
136
137 vmsleu.vv v8, v4, v20
138 # CHECK-INST: vmsleu.vv v8, v4, v20
139 # CHECK-ENCODING: [0x57,0x04,0x4a,0x72]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 57 04 4a 72 <unknown>
142
143 vmsleu.vx v8, v4, a0, v0.t
144 # CHECK-INST: vmsleu.vx v8, v4, a0, v0.t
145 # CHECK-ENCODING: [0x57,0x44,0x45,0x70]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 44 45 70 <unknown>
148
149 vmsleu.vx v8, v4, a0
150 # CHECK-INST: vmsleu.vx v8, v4, a0
151 # CHECK-ENCODING: [0x57,0x44,0x45,0x72]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 57 44 45 72 <unknown>
154
155 vmsleu.vi v8, v4, 15, v0.t
156 # CHECK-INST: vmsleu.vi v8, v4, 15, v0.t
157 # CHECK-ENCODING: [0x57,0xb4,0x47,0x70]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 b4 47 70 <unknown>
160
161 vmsleu.vi v8, v4, 15
162 # CHECK-INST: vmsleu.vi v8, v4, 15
163 # CHECK-ENCODING: [0x57,0xb4,0x47,0x72]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 b4 47 72 <unknown>
166
167 vmsle.vv v8, v4, v20, v0.t
168 # CHECK-INST: vmsle.vv v8, v4, v20, v0.t
169 # CHECK-ENCODING: [0x57,0x04,0x4a,0x74]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
171 # CHECK-UNKNOWN: 57 04 4a 74 <unknown>
172
173 vmsle.vv v8, v4, v20
174 # CHECK-INST: vmsle.vv v8, v4, v20
175 # CHECK-ENCODING: [0x57,0x04,0x4a,0x76]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    a0, v0.t
181 # CHECK-ENCODING: [0x57,0x44,0x45,0x74]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 44 45 74 <unknown>
184
185 vmsle.vx v8, v4, a0
186 # CHECK-INST: vmsle.vx v8, v4, a0
187 # CHECK-ENCODING: [0x57,0x44,0x45,0x76]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 44 45 76 <unknown>
190
191 vmsle.vi v8, v4, 15, v0.t
192 # CHECK-INST: vmsle.vi v8, v4, 15, v0.t
193 # CHECK-ENCODING: [0x57,0xb4,0x47,0x74]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 57 b4 47 74 <unknown>
196
197 vmsle.vi v8, v4, 15
198 # CHECK-INST: vmsle.vi v8, v4, 15
199 # CHECK-ENCODING: [0x57,0xb4,0x47,0x76]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 57 b4 47 76 <unknown>
202
203 vmsgtu.vx v8, v4, a0, v0.t
204 # CHECK-INST: vmsgtu.vx v8, v4, a0, v0.t
205 # CHECK-ENCODING: [0x57,0x44,0x45,0x78]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 57 44 45 78 <unknown>
208
209 vmsgtu.vx v8, v4, a0
210 # CHECK-INST: vmsgtu.vx v8, v4, a0
211 # CHECK-ENCODING: [0x57,0x44,0x45,0x7a]
212 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
213 # CHECK-UNKNOWN: 57 44 45 7a <unknown>
214
215 vmsgtu.vi v8, v4, 15, v0.t
216 # CHECK-INST: vmsgtu.vi v8, v4, 15, v0.t
217 # CHECK-ENCODING: [0x57,0xb4,0x47,0x78]
218 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
219 # CHECK-UNKNOWN: 57 b4 47 78 <unknown>
220
221 vmsgtu.vi v8, v4, 15
222 # CHECK-INST: vmsgtu.vi v8, v4, 15
223 # CHECK-ENCODING: [0x57,0xb4,0x47,0x7a]
224 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
225 # CHECK-UNKNOWN: 57 b4 47 7a <unknown>
226
227 vmsgt.vx v8, v4, a0, v0.t
228 # CHECK-INST: vmsgt.vx v8, v4, a0, v0.t
229 # CHECK-ENCODING: [0x57,0x44,0x45,0x7c]
230 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
231 # CHECK-UNKNOWN: 57 44 45 7c <unknown>
232
233 vmsgt.vx v8, v4, a0
234 # CHECK-INST: vmsgt.vx v8, v4, a0
235 # CHECK-ENCODING: [0x57,0x44,0x45,0x7e]
236 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
237 # CHECK-UNKNOWN: 57 44 45 7e <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
15, v0.t
47,0x7c]
242 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
243 # CHECK-UNKNOWN: 57 b4 47 7c <unknown>
244
245 vmsgt.vi v8, v4, 15
246 # CHECK-INST: vmsgt.vi v8, v4, 15
247 # CHECK-ENCODING: [0x57,0xb4,0x47,0x7e]
248 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
249 # CHECK-UNKNOWN: 57 b4 47 7e <unknown>
```

llvm/test/MC/RISCV/rvv/convert.s

View Options

This file was added.

```
  1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
  2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
  3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
  4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
  5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
  6 # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
  7 # RUN:         | FileCheck %s --check-prefix=CHECK-INST
  8 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
  9 # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
 10
 11 vfcvt.xu.f.v v8, v4, v0.t
 12 # CHECK-INST: vfcvt.xu.f.v v8, v4, v0.t
 13 # CHECK-ENCODING: [0x57,0x14,0x40,0x88]
 14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 15 # CHECK-UNKNOWN: 57 14 40 88 <unknown>
 16
 17 vfcvt.xu.f.v v8, v4
 18 # CHECK-INST: vfcvt.xu.f.v v8, v4
 19 # CHECK-ENCODING: [0x57,0x14,0x40,0x8a]
 20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 21 # CHECK-UNKNOWN: 57 14 40 8a <unknown>
 22
 23 vfcvt.x.f.v v8, v4, v0.t
 24 # CHECK-INST: vfcvt.x.f.v v8, v4, v0.t
 25 # CHECK-ENCODING: [0x57,0x94,0x40,0x88]
 26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 27 # CHECK-UNKNOWN: 57 94 40 88 <unknown>
 28
 29 vfcvt.x.f.v v8, v4
 30 # CHECK-INST: vfcvt.x.f.v v8, v4
 31 # CHECK-ENCODING: [0x57,0x94,0x40,0x8a]
 32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 33 # CHECK-UNKNOWN: 57 94 40 8a <unknown>
 34
 35 vfcvt.f.xu.v v8, v4, v0.t
 36 # CHECK-INST: vfcvt.f.xu.v v8, v4, v0.t
 37 # CHECK-ENCODING: [0x57,0x14,0x41,0x88]
 38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 39 # CHECK-UNKNOWN: 57 14 41 88 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
    v4
    41,0x8a]
 44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
```

```
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 41 8a <unknown>

vfcvt.f.x.v v8, v4, v0.t
# CHECK-INST: vfcvt.f.x.v v8, v4, v0.t
# CHECK-ENCODING: [0x57,0x94,0x41,0x88]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 94 41 88 <unknown>

vfcvt.f.x.v v8, v4
# CHECK-INST: vfcvt.f.x.v v8, v4
# CHECK-ENCODING: [0x57,0x94,0x41,0x8a]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 94 41 8a <unknown>

vfwcvt.xu.f.v v8, v4, v0.t
# CHECK-INST: vfwcvt.xu.f.v v8, v4, v0.t
# CHECK-ENCODING: [0x57,0x14,0x44,0x88]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 44 88 <unknown>

vfwcvt.xu.f.v v8, v4
# CHECK-INST: vfwcvt.xu.f.v v8, v4
# CHECK-ENCODING: [0x57,0x14,0x44,0x8a]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 44 8a <unknown>

vfwcvt.x.f.v v8, v4, v0.t
# CHECK-INST: vfwcvt.x.f.v v8, v4, v0.t
# CHECK-ENCODING: [0x57,0x94,0x44,0x88]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 94 44 88 <unknown>

vfwcvt.x.f.v v8, v4
# CHECK-INST: vfwcvt.x.f.v v8, v4
# CHECK-ENCODING: [0x57,0x94,0x44,0x8a]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 94 44 8a <unknown>

vfwcvt.f.xu.v v8, v4, v0.t
# CHECK-INST: vfwcvt.f.xu.v v8, v4, v0.t
# CHECK-ENCODING: [0x57,0x14,0x45,0x88]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 45 88 <unknown>

vfwcvt.f.xu.v v8, v4
# CHECK-INST: vfwcvt.f.xu.v v8, v4
# CHECK-ENCODING: [0x57,0x14,0x45,0x8a]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
nknown>

v4, v0.t
# CHECK-ENCODING: [0x57,0x94,0x45,0x88]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
98  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
99  # CHECK-UNKNOWN: 57 94 45 88 <unknown>
100
101 vfwcvt.f.x.v v8, v4
102 # CHECK-INST: vfwcvt.f.x.v v8, v4
103 # CHECK-ENCODING: [0x57,0x94,0x45,0x8a]
104 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
105 # CHECK-UNKNOWN: 57 94 45 8a <unknown>
106
107 vfwcvt.f.f.v v8, v4, v0.t
108 # CHECK-INST: vfwcvt.f.f.v v8, v4, v0.t
109 # CHECK-ENCODING: [0x57,0x14,0x46,0x88]
110 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
111 # CHECK-UNKNOWN: 57 14 46 88 <unknown>
112
113 vfwcvt.f.f.v v8, v4
114 # CHECK-INST: vfwcvt.f.f.v v8, v4
115 # CHECK-ENCODING: [0x57,0x14,0x46,0x8a]
116 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
117 # CHECK-UNKNOWN: 57 14 46 8a <unknown>
118
119 vfncvt.xu.f.w v8, v4, v0.t
120 # CHECK-INST: vfncvt.xu.f.w v8, v4, v0.t
121 # CHECK-ENCODING: [0x57,0x14,0x48,0x88]
122 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
123 # CHECK-UNKNOWN: 57 14 48 88 <unknown>
124
125 vfncvt.xu.f.w v8, v4
126 # CHECK-INST: vfncvt.xu.f.w v8, v4
127 # CHECK-ENCODING: [0x57,0x14,0x48,0x8a]
128 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
129 # CHECK-UNKNOWN: 57 14 48 8a <unknown>
130
131 vfncvt.x.f.w v8, v4, v0.t
132 # CHECK-INST: vfncvt.x.f.w v8, v4, v0.t
133 # CHECK-ENCODING: [0x57,0x94,0x48,0x88]
134 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
135 # CHECK-UNKNOWN: 57 94 48 88 <unknown>
136
137 vfncvt.x.f.w v8, v4
138 # CHECK-INST: vfncvt.x.f.w v8, v4
139 # CHECK-ENCODING: [0x57,0x94,0x48,0x8a]
140 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
141 # CHECK-UNKNOWN: 57 94 48 8a <unknown>
142
143 vfncvt.f.xu.w v8, v4, v0.t
144 # CHECK-INST: vfncvt.f.xu.w v8, v4, v0.t
145 # CHECK-ENCODING: [0x57,0x14,0x49,0x88]
146 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
```



```
nknown>

150 # CHECK-INST: vfncvt.f.xu.w v8, v4
```

```
151 # CHECK-ENCODING: [0x57,0x14,0x49,0x8a]
152 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
153 # CHECK-UNKNOWN: 57 14 49 8a <unknown>
154
155 vfncvt.f.x.w v8, v4, v0.t
156 # CHECK-INST: vfncvt.f.x.w v8, v4, v0.t
157 # CHECK-ENCODING: [0x57,0x94,0x49,0x88]
158 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
159 # CHECK-UNKNOWN: 57 94 49 88 <unknown>
160
161 vfncvt.f.x.w v8, v4
162 # CHECK-INST: vfncvt.f.x.w v8, v4
163 # CHECK-ENCODING: [0x57,0x94,0x49,0x8a]
164 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
165 # CHECK-UNKNOWN: 57 94 49 8a <unknown>
166
167 vfncvt.f.f.w v8, v4, v0.t
168 # CHECK-INST: vfncvt.f.f.w v8, v4, v0.t
169 # CHECK-ENCODING: [0x57,0x14,0x4a,0x88]
170 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
171 # CHECK-UNKNOWN: 57 14 4a 88 <unknown>
172
173 vfncvt.f.f.w v8, v4
174 # CHECK-INST: vfncvt.f.f.w v8, v4
175 # CHECK-ENCODING: [0x57,0x14,0x4a,0x8a]
176 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
177 # CHECK-UNKNOWN: 57 14 4a 8a <unknown>
178
179 vfncvt.rod.f.f.w v8, v4, v0.t
180 # CHECK-INST: vfncvt.rod.f.f.w v8, v4, v0.t
181 # CHECK-ENCODING: [0x57,0x94,0x4a,0x88]
182 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
183 # CHECK-UNKNOWN: 57 94 4a 88 <unknown>
184
185 vfncvt.rod.f.f.w v8, v4
186 # CHECK-INST: vfncvt.rod.f.f.w v8, v4
187 # CHECK-ENCODING: [0x57,0x94,0x4a,0x8a]
188 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
189 # CHECK-UNKNOWN: 57 94 4a 8a <unknown>
```

llvm/test/MC/RISCV/rvv/div.s

View Options

This file was added.

```
 1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
 2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
 3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
 4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
 5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
                                                --mattr=+experimental-v - \
                                               check-prefix=CHECK-INST
                                               -filetype=obj --mattr=+experimental-v < %s \
                                                - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
vdivu.vv v8, v4, v20, v0.t
# CHECK-INST: vdivu.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x80]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 80 <unknown>

vdivu.vv v8, v4, v20
# CHECK-INST: vdivu.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x82]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 82 <unknown>

vdivu.vx v8, v4, a0, v0.t
# CHECK-INST: vdivu.vx v8, v4, a0, v0.t
# CHECK-ENCODING: [0x57,0x64,0x45,0x80]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 64 45 80 <unknown>

vdivu.vx v8, v4, a0
# CHECK-INST: vdivu.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x64,0x45,0x82]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 64 45 82 <unknown>

vdiv.vv v8, v4, v20, v0.t
# CHECK-INST: vdiv.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x84]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 84 <unknown>

vdiv.vv v8, v4, v20
# CHECK-INST: vdiv.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x86]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 86 <unknown>

vdiv.vx v8, v4, a0, v0.t
# CHECK-INST: vdiv.vx v8, v4, a0, v0.t
# CHECK-ENCODING: [0x57,0x64,0x45,0x84]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 64 45 84 <unknown>

vdiv.vx v8, v4, a0
# CHECK-INST: vdiv.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x64,0x45,0x86]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 64 45 86 <unknown>

vremu.vv v8, v4, v20, v0.t
# CHECK-INST: vremu.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x88]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 88 <unknown>

vremu.vv v8, v4, v20
# CHECK-INST: vremu.vv v8, v4, v20
                                4a,0x8a]
                                ires the following: 'V' (Vector Instructions)
                                nknown>

vremu.vx v8, v4, a0, v0.t
```

```
72  # CHECK-INST: vremu.vx v8, v4, a0, v0.t
73  # CHECK-ENCODING: [0x57,0x64,0x45,0x88]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 64 45 88 <unknown>
76
77  vremu.vx v8, v4, a0
78  # CHECK-INST: vremu.vx v8, v4, a0
79  # CHECK-ENCODING: [0x57,0x64,0x45,0x8a]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 64 45 8a <unknown>
82
83  vrem.vv v8, v4, v20, v0.t
84  # CHECK-INST: vrem.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x24,0x4a,0x8c]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 24 4a 8c <unknown>
88
89  vrem.vv v8, v4, v20
90  # CHECK-INST: vrem.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x24,0x4a,0x8e]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 24 4a 8e <unknown>
94
95  vrem.vx v8, v4, a0, v0.t
96  # CHECK-INST: vrem.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x64,0x45,0x8c]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 64 45 8c <unknown>
100
101 vrem.vx v8, v4, a0
102 # CHECK-INST: vrem.vx v8, v4, a0
103 # CHECK-ENCODING: [0x57,0x64,0x45,0x8e]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 64 45 8e <unknown>
```

llvm/test/MC/RISCV/rvv/fadd.s

View Options

This file was added.

```
1   # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2   # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3   # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4   # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5   # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6   # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7   # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8   # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9   # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11  vfadd.vv v8, v4, v20, v0.t
12  # CHECK-INST: vfadd.vv v8, v4, v20, v0.t
13  # CHECK-ENCODING: [0x57,0x14,0x4a,0x00]
14  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
15  # CHECK-UNKNOWN: 57 14 4a 00 <unknown>
16
17
18                                  v20
19                                  4a,0x02]
20  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 02 <unknown>

vfadd.vf v8, v4, fa0, v0.t
# CHECK-INST: vfadd.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x00]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 00 <unknown>

vfadd.vf v8, v4, fa0
# CHECK-INST: vfadd.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x02]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 02 <unknown>

vfwadd.vv v8, v4, v20, v0.t
# CHECK-INST: vfwadd.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0xc0]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a c0 <unknown>

vfwadd.vv v8, v4, v20
# CHECK-INST: vfwadd.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0xc2]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a c2 <unknown>

vfwadd.vf v8, v4, fa0, v0.t
# CHECK-INST: vfwadd.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0xc0]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 c0 <unknown>

vfwadd.vf v8, v4, fa0
# CHECK-INST: vfwadd.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0xc2]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 c2 <unknown>

vfwadd.wv v8, v4, v20, v0.t
# CHECK-INST: vfwadd.wv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0xd0]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a d0 <unknown>

vfwadd.wv v8, v4, v20
# CHECK-INST: vfwadd.wv v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0xd2]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
                                                nknown>

                                                fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0xd0]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
74 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
75 # CHECK-UNKNOWN: 57 54 45 d0 <unknown>
76
77 vfwadd.wf v8, v4, fa0
78 # CHECK-INST: vfwadd.wf v8, v4, fa0
79 # CHECK-ENCODING: [0x57,0x54,0x45,0xd2]
80 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
81 # CHECK-UNKNOWN: 57 54 45 d2 <unknown>
```

llvm/test/MC/RISCV/rvv/fcompare.s

View Options

This file was added.

```
 1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
 2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
 3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
 4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
 5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
 6 # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
 7 # RUN:         | FileCheck %s --check-prefix=CHECK-INST
 8 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
 9 # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vmfeq.vv v8, v4, v20, v0.t
12 # CHECK-INST: vmfeq.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x14,0x4a,0x60]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
15 # CHECK-UNKNOWN: 57 14 4a 60 <unknown>
16
17 vmfeq.vv v8, v4, v20
18 # CHECK-INST: vmfeq.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x14,0x4a,0x62]
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 4a 62 <unknown>
22
23 vmfeq.vf v8, v4, fa0, v0.t
24 # CHECK-INST: vmfeq.vf v8, v4, fa0, v0.t
25 # CHECK-ENCODING: [0x57,0x54,0x45,0x60]
26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
27 # CHECK-UNKNOWN: 57 54 45 60 <unknown>
28
29 vmfeq.vf v8, v4, fa0
30 # CHECK-INST: vmfeq.vf v8, v4, fa0
31 # CHECK-ENCODING: [0x57,0x54,0x45,0x62]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 54 45 62 <unknown>
34
35 vmfne.vv v8, v4, v20, v0.t
36 # CHECK-INST: vmfne.vv v8, v4, v20, v0.t
37 # CHECK-ENCODING: [0x57,0x14,0x4a,0x70]
   ires the following: 'F' (Single-Precision Floating-Point), 'V'
   nknown>
41 vmfne.vv v8, v4, v20
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
42 # CHECK-INST: vmfne.vv v8, v4, v20
43 # CHECK-ENCODING: [0x57,0x14,0x4a,0x72]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
45 # CHECK-UNKNOWN: 57 14 4a 72 <unknown>
46
47 vmfne.vf v8, v4, fa0, v0.t
48 # CHECK-INST: vmfne.vf v8, v4, fa0, v0.t
49 # CHECK-ENCODING: [0x57,0x54,0x45,0x70]
50 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
51 # CHECK-UNKNOWN: 57 54 45 70 <unknown>
52
53 vmfne.vf v8, v4, fa0
54 # CHECK-INST: vmfne.vf v8, v4, fa0
55 # CHECK-ENCODING: [0x57,0x54,0x45,0x72]
56 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
57 # CHECK-UNKNOWN: 57 54 45 72 <unknown>
58
59 vmflt.vv v8, v4, v20, v0.t
60 # CHECK-INST: vmflt.vv v8, v4, v20, v0.t
61 # CHECK-ENCODING: [0x57,0x14,0x4a,0x6c]
62 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
63 # CHECK-UNKNOWN: 57 14 4a 6c <unknown>
64
65 vmflt.vv v8, v4, v20
66 # CHECK-INST: vmflt.vv v8, v4, v20
67 # CHECK-ENCODING: [0x57,0x14,0x4a,0x6e]
68 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
69 # CHECK-UNKNOWN: 57 14 4a 6e <unknown>
70
71 vmflt.vf v8, v4, fa0, v0.t
72 # CHECK-INST: vmflt.vf v8, v4, fa0, v0.t
73 # CHECK-ENCODING: [0x57,0x54,0x45,0x6c]
74 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
75 # CHECK-UNKNOWN: 57 54 45 6c <unknown>
76
77 vmflt.vf v8, v4, fa0
78 # CHECK-INST: vmflt.vf v8, v4, fa0
79 # CHECK-ENCODING: [0x57,0x54,0x45,0x6e]
80 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
81 # CHECK-UNKNOWN: 57 54 45 6e <unknown>
82
83 vmfle.vv v8, v4, v20, v0.t
84 # CHECK-INST: vmfle.vv v8, v4, v20, v0.t
85 # CHECK-ENCODING: [0x57,0x14,0x4a,0x64]
86 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
87 # CHECK-UNKNOWN: 57 14 4a 64 <unknown>
88
89 vmfle.vv v8, v4, v20
90 # CHECK-INST: vmfle.vv v8, v4, v20
                          4a,0x66]
                          ires the following: 'F' (Single-Precision Floating-Point), 'V'

                          nknown>
94
```

```
vmfle.vf v8, v4, fa0, v0.t
# CHECK-INST: vmfle.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x64]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 64 <unknown>

vmfle.vf v8, v4, fa0
# CHECK-INST: vmfle.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x66]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 66 <unknown>

vmfgt.vf v8, v4, fa0, v0.t
# CHECK-INST: vmfgt.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x74]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 74 <unknown>

vmfgt.vf v8, v4, fa0
# CHECK-INST: vmfgt.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x76]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 76 <unknown>

vmfge.vf v8, v4, fa0, v0.t
# CHECK-INST: vmfge.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x7c]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 7c <unknown>

vmfge.vf v8, v4, fa0
# CHECK-INST: vmfge.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x7e]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 7e <unknown>
```

llvm/test/MC/RISCV/rvv/fdiv.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
# RUN:         | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
v20, v0.t
4a,0x80]
ires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
```

```
15 # CHECK-UNKNOWN: 57 14 4a 80 <unknown>
16
17 vfdiv.vv v8, v4, v20
18 # CHECK-INST: vfdiv.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x14,0x4a,0x82]
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 4a 82 <unknown>
22
23 vfdiv.vf v8, v4, fa0, v0.t
24 # CHECK-INST: vfdiv.vf v8, v4, fa0, v0.t
25 # CHECK-ENCODING: [0x57,0x54,0x45,0x80]
26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
27 # CHECK-UNKNOWN: 57 54 45 80 <unknown>
28
29 vfdiv.vf v8, v4, fa0
30 # CHECK-INST: vfdiv.vf v8, v4, fa0
31 # CHECK-ENCODING: [0x57,0x54,0x45,0x82]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 54 45 82 <unknown>
34
35 vfrdiv.vf v8, v4, fa0, v0.t
36 # CHECK-INST: vfrdiv.vf v8, v4, fa0, v0.t
37 # CHECK-ENCODING: [0x57,0x54,0x45,0x84]
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
39 # CHECK-UNKNOWN: 57 54 45 84 <unknown>
40
41 vfrdiv.vf v8, v4, fa0
42 # CHECK-INST: vfrdiv.vf v8, v4, fa0
43 # CHECK-ENCODING: [0x57,0x54,0x45,0x86]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
45 # CHECK-UNKNOWN: 57 54 45 86 <unknown>
```

llvm/test/MC/RISCV/rvv/fmacc.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vfmacc.vv v8, v20, v4, v0.t
12 # CHECK-INST: vfmacc.vv v8, v20, v4, v0.t
13 # CHECK-ENCODING: [0x57,0x14,0x4a,0xb0]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
   nknown>
   , v4
19 # CHECK-ENCODING: [0x57,0x14,0x4a,0xb2]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 4a b2 <unknown>
22
23 vfmacc.vf v8, fa0, v4, v0.t
24 # CHECK-INST: vfmacc.vf v8, fa0, v4, v0.t
25 # CHECK-ENCODING: [0x57,0x54,0x45,0xb0]
26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
27 # CHECK-UNKNOWN: 57 54 45 b0 <unknown>
28
29 vfmacc.vf v8, fa0, v4
30 # CHECK-INST: vfmacc.vf v8, fa0, v4
31 # CHECK-ENCODING: [0x57,0x54,0x45,0xb2]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 54 45 b2 <unknown>
34
35 vfnmacc.vv v8, v20, v4, v0.t
36 # CHECK-INST: vfnmacc.vv v8, v20, v4, v0.t
37 # CHECK-ENCODING: [0x57,0x14,0x4a,0xb4]
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
39 # CHECK-UNKNOWN: 57 14 4a b4 <unknown>
40
41 vfnmacc.vv v8, v20, v4
42 # CHECK-INST: vfnmacc.vv v8, v20, v4
43 # CHECK-ENCODING: [0x57,0x14,0x4a,0xb6]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
45 # CHECK-UNKNOWN: 57 14 4a b6 <unknown>
46
47 vfnmacc.vf v8, fa0, v4, v0.t
48 # CHECK-INST: vfnmacc.vf v8, fa0, v4, v0.t
49 # CHECK-ENCODING: [0x57,0x54,0x45,0xb4]
50 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
51 # CHECK-UNKNOWN: 57 54 45 b4 <unknown>
52
53 vfnmacc.vf v8, fa0, v4
54 # CHECK-INST: vfnmacc.vf v8, fa0, v4
55 # CHECK-ENCODING: [0x57,0x54,0x45,0xb6]
56 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
57 # CHECK-UNKNOWN: 57 54 45 b6 <unknown>
58
59 vfmsac.vv v8, v20, v4, v0.t
60 # CHECK-INST: vfmsac.vv v8, v20, v4, v0.t
61 # CHECK-ENCODING: [0x57,0x14,0x4a,0xb8]
62 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
63 # CHECK-UNKNOWN: 57 14 4a b8 <unknown>
64
65 vfmsac.vv v8, v20, v4
66 # CHECK-INST: vfmsac.vv v8, v20, v4
67 # CHECK-ENCODING: [0x57,0x14,0x4a,0xba]
68 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
                                                    nknown>
72 # CHECK-INST: vfmsac.vf v8, fa0, v4, v0.t
```

```
73  # CHECK-ENCODING: [0x57,0x54,0x45,0xb8]
74  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
75  # CHECK-UNKNOWN: 57 54 45 b8 <unknown>
76
77  vfmsac.vf v8, fa0, v4
78  # CHECK-INST: vfmsac.vf v8, fa0, v4
79  # CHECK-ENCODING: [0x57,0x54,0x45,0xba]
80  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
81  # CHECK-UNKNOWN: 57 54 45 ba <unknown>
82
83  vfnmsac.vv v8, v20, v4, v0.t
84  # CHECK-INST: vfnmsac.vv v8, v20, v4, v0.t
85  # CHECK-ENCODING: [0x57,0x14,0x4a,0xbc]
86  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
87  # CHECK-UNKNOWN: 57 14 4a bc <unknown>
88
89  vfnmsac.vv v8, v20, v4
90  # CHECK-INST: vfnmsac.vv v8, v20, v4
91  # CHECK-ENCODING: [0x57,0x14,0x4a,0xbe]
92  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
93  # CHECK-UNKNOWN: 57 14 4a be <unknown>
94
95  vfnmsac.vf v8, fa0, v4, v0.t
96  # CHECK-INST: vfnmsac.vf v8, fa0, v4, v0.t
97  # CHECK-ENCODING: [0x57,0x54,0x45,0xbc]
98  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
99  # CHECK-UNKNOWN: 57 54 45 bc <unknown>
100
101 vfnmsac.vf v8, fa0, v4
102 # CHECK-INST: vfnmsac.vf v8, fa0, v4
103 # CHECK-ENCODING: [0x57,0x54,0x45,0xbe]
104 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
105 # CHECK-UNKNOWN: 57 54 45 be <unknown>
106
107 vfmadd.vv v8, v20, v4, v0.t
108 # CHECK-INST: vfmadd.vv v8, v20, v4, v0.t
109 # CHECK-ENCODING: [0x57,0x14,0x4a,0xa0]
110 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
111 # CHECK-UNKNOWN: 57 14 4a a0 <unknown>
112
113 vfmadd.vv v8, v20, v4
114 # CHECK-INST: vfmadd.vv v8, v20, v4
115 # CHECK-ENCODING: [0x57,0x14,0x4a,0xa2]
116 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
117 # CHECK-UNKNOWN: 57 14 4a a2 <unknown>
118
119 vfmadd.vf v8, fa0, v4, v0.t
120 # CHECK-INST: vfmadd.vf v8, fa0, v4, v0.t
121 # CHECK-ENCODING: [0x57,0x54,0x45,0xa0]
    ires the following: 'F' (Single-Precision Floating-Point), 'V'
    nknown>
125 vfmadd.vf v8, fa0, v4
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
126 # CHECK-INST: vfmadd.vf v8, fa0, v4
127 # CHECK-ENCODING: [0x57,0x54,0x45,0xa2]
128 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
129 # CHECK-UNKNOWN: 57 54 45 a2 <unknown>
130
131 vfnmadd.vv v8, v20, v4, v0.t
132 # CHECK-INST: vfnmadd.vv v8, v20, v4, v0.t
133 # CHECK-ENCODING: [0x57,0x14,0x4a,0xa4]
134 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
135 # CHECK-UNKNOWN: 57 14 4a a4 <unknown>
136
137 vfnmadd.vv v8, v20, v4
138 # CHECK-INST: vfnmadd.vv v8, v20, v4
139 # CHECK-ENCODING: [0x57,0x14,0x4a,0xa6]
140 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
141 # CHECK-UNKNOWN: 57 14 4a a6 <unknown>
142
143 vfnmadd.vf v8, fa0, v4, v0.t
144 # CHECK-INST: vfnmadd.vf v8, fa0, v4, v0.t
145 # CHECK-ENCODING: [0x57,0x54,0x45,0xa4]
146 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
147 # CHECK-UNKNOWN: 57 54 45 a4 <unknown>
148
149 vfnmadd.vf v8, fa0, v4
150 # CHECK-INST: vfnmadd.vf v8, fa0, v4
151 # CHECK-ENCODING: [0x57,0x54,0x45,0xa6]
152 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
153 # CHECK-UNKNOWN: 57 54 45 a6 <unknown>
154
155 vfmsub.vv v8, v20, v4, v0.t
156 # CHECK-INST: vfmsub.vv v8, v20, v4, v0.t
157 # CHECK-ENCODING: [0x57,0x14,0x4a,0xa8]
158 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
159 # CHECK-UNKNOWN: 57 14 4a a8 <unknown>
160
161 vfmsub.vv v8, v20, v4
162 # CHECK-INST: vfmsub.vv v8, v20, v4
163 # CHECK-ENCODING: [0x57,0x14,0x4a,0xaa]
164 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
165 # CHECK-UNKNOWN: 57 14 4a aa <unknown>
166
167 vfmsub.vf v8, fa0, v4, v0.t
168 # CHECK-INST: vfmsub.vf v8, fa0, v4, v0.t
169 # CHECK-ENCODING: [0x57,0x54,0x45,0xa8]
170 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
171 # CHECK-UNKNOWN: 57 54 45 a8 <unknown>
172
173 vfmsub.vf v8, fa0, v4
174 # CHECK-INST: vfmsub.vf v8, fa0, v4
                                  45,0xaa]
                                  ires the following: 'F' (Single-Precision Floating-Point), 'V'

                                  nknown>
178
```

```
179 vfnmsub.vv v8, v20, v4, v0.t
180 # CHECK-INST: vfnmsub.vv v8, v20, v4, v0.t
181 # CHECK-ENCODING: [0x57,0x14,0x4a,0xac]
182 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
183 # CHECK-UNKNOWN: 57 14 4a ac <unknown>
184
185 vfnmsub.vv v8, v20, v4
186 # CHECK-INST: vfnmsub.vv v8, v20, v4
187 # CHECK-ENCODING: [0x57,0x14,0x4a,0xae]
188 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
189 # CHECK-UNKNOWN: 57 14 4a ae <unknown>
190
191 vfnmsub.vf v8, fa0, v4, v0.t
192 # CHECK-INST: vfnmsub.vf v8, fa0, v4, v0.t
193 # CHECK-ENCODING: [0x57,0x54,0x45,0xac]
194 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
195 # CHECK-UNKNOWN: 57 54 45 ac <unknown>
196
197 vfnmsub.vf v8, fa0, v4
198 # CHECK-INST: vfnmsub.vf v8, fa0, v4
199 # CHECK-ENCODING: [0x57,0x54,0x45,0xae]
200 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
201 # CHECK-UNKNOWN: 57 54 45 ae <unknown>
202
203 vfwmacc.vv v8, v20, v4, v0.t
204 # CHECK-INST: vfwmacc.vv v8, v20, v4, v0.t
205 # CHECK-ENCODING: [0x57,0x14,0x4a,0xf0]
206 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
207 # CHECK-UNKNOWN: 57 14 4a f0 <unknown>
208
209 vfwmacc.vv v8, v20, v4
210 # CHECK-INST: vfwmacc.vv v8, v20, v4
211 # CHECK-ENCODING: [0x57,0x14,0x4a,0xf2]
212 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
213 # CHECK-UNKNOWN: 57 14 4a f2 <unknown>
214
215 vfwmacc.vf v8, fa0, v4, v0.t
216 # CHECK-INST: vfwmacc.vf v8, fa0, v4, v0.t
217 # CHECK-ENCODING: [0x57,0x54,0x45,0xf0]
218 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
219 # CHECK-UNKNOWN: 57 54 45 f0 <unknown>
220
221 vfwmacc.vf v8, fa0, v4
222 # CHECK-INST: vfwmacc.vf v8, fa0, v4
223 # CHECK-ENCODING: [0x57,0x54,0x45,0xf2]
224 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
225 # CHECK-UNKNOWN: 57 54 45 f2 <unknown>
226
227 vfwnmacc.vv v8, v20, v4, v0.t
                20, v4, v0.t
                4a,0xf4]
                ires the following: 'F' (Single-Precision Floating-Point), 'V'
231 # CHECK-UNKNOWN: 57 14 4a f4 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
232
233 vfwnmacc.vv v8, v20, v4
234 # CHECK-INST: vfwnmacc.vv v8, v20, v4
235 # CHECK-ENCODING: [0x57,0x14,0x4a,0xf6]
236 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
237 # CHECK-UNKNOWN: 57 14 4a f6 <unknown>
238
239 vfwnmacc.vf v8, fa0, v4, v0.t
240 # CHECK-INST: vfwnmacc.vf v8, fa0, v4, v0.t
241 # CHECK-ENCODING: [0x57,0x54,0x45,0xf4]
242 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
243 # CHECK-UNKNOWN: 57 54 45 f4 <unknown>
244
245 vfwnmacc.vf v8, fa0, v4
246 # CHECK-INST: vfwnmacc.vf v8, fa0, v4
247 # CHECK-ENCODING: [0x57,0x54,0x45,0xf6]
248 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
249 # CHECK-UNKNOWN: 57 54 45 f6 <unknown>
250
251 vfwmsac.vv v8, v20, v4, v0.t
252 # CHECK-INST: vfwmsac.vv v8, v20, v4, v0.t
253 # CHECK-ENCODING: [0x57,0x14,0x4a,0xf8]
254 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
255 # CHECK-UNKNOWN: 57 14 4a f8 <unknown>
256
257 vfwmsac.vv v8, v20, v4
258 # CHECK-INST: vfwmsac.vv v8, v20, v4
259 # CHECK-ENCODING: [0x57,0x14,0x4a,0xfa]
260 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
261 # CHECK-UNKNOWN: 57 14 4a fa <unknown>
262
263 vfwmsac.vf v8, fa0, v4, v0.t
264 # CHECK-INST: vfwmsac.vf v8, fa0, v4, v0.t
265 # CHECK-ENCODING: [0x57,0x54,0x45,0xf8]
266 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
267 # CHECK-UNKNOWN: 57 54 45 f8 <unknown>
268
269 vfwmsac.vf v8, fa0, v4
270 # CHECK-INST: vfwmsac.vf v8, fa0, v4
271 # CHECK-ENCODING: [0x57,0x54,0x45,0xfa]
272 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
273 # CHECK-UNKNOWN: 57 54 45 fa <unknown>
274
275 vfwnmsac.vv v8, v20, v4, v0.t
276 # CHECK-INST: vfwnmsac.vv v8, v20, v4, v0.t
277 # CHECK-ENCODING: [0x57,0x14,0x4a,0xfc]
278 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
279 # CHECK-UNKNOWN: 57 14 4a fc <unknown>
280
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
20, v4
4a,0xfe]
ires the following: 'F' (Single-Precision Floating-Point), 'V'
(Vector Instructions)
```

```
285 # CHECK-UNKNOWN: 57 14 4a fe <unknown>
286
287 vfwnmsac.vf v8, fa0, v4, v0.t
288 # CHECK-INST: vfwnmsac.vf v8, fa0, v4, v0.t
289 # CHECK-ENCODING: [0x57,0x54,0x45,0xfc]
290 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
291 # CHECK-UNKNOWN: 57 54 45 fc <unknown>
292
293 vfwnmsac.vf v8, fa0, v4
294 # CHECK-INST: vfwnmsac.vf v8, fa0, v4
295 # CHECK-ENCODING: [0x57,0x54,0x45,0xfe]
296 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
297 # CHECK-UNKNOWN: 57 54 45 fe <unknown>
```

llvm/test/MC/RISCV/rvv/fminmax.s

View Options

This file was added.

```
  1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
  2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
  3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
  4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
  5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
  6 # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
  7 # RUN:         | FileCheck %s --check-prefix=CHECK-INST
  8 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
  9 # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
 10
 11 vfmin.vv v8, v4, v20, v0.t
 12 # CHECK-INST: vfmin.vv v8, v4, v20, v0.t
 13 # CHECK-ENCODING: [0x57,0x14,0x4a,0x10]
 14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 15 # CHECK-UNKNOWN: 57 14 4a 10 <unknown>
 16
 17 vfmin.vv v8, v4, v20
 18 # CHECK-INST: vfmin.vv v8, v4, v20
 19 # CHECK-ENCODING: [0x57,0x14,0x4a,0x12]
 20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 21 # CHECK-UNKNOWN: 57 14 4a 12 <unknown>
 22
 23 vfmin.vf v8, v4, fa0, v0.t
 24 # CHECK-INST: vfmin.vf v8, v4, fa0, v0.t
 25 # CHECK-ENCODING: [0x57,0x54,0x45,0x10]
 26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
 27 # CHECK-UNKNOWN: 57 54 45 10 <unknown>
 28
 29 vfmin.vf v8, v4, fa0
 30 # CHECK-INST: vfmin.vf v8, v4, fa0
 31 # CHECK-ENCODING: [0x57,0x54,0x45,0x12]
 32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
    nknown>

    v20, v0.t
 37 # CHECK-ENCODING: [0x57,0x14,0x4a,0x18]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 14 4a 18 <unknown>
40
41 vfmax.vv v8, v4, v20
42 # CHECK-INST: vfmax.vv v8, v4, v20
43 # CHECK-ENCODING: [0x57,0x14,0x4a,0x1a]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 14 4a 1a <unknown>
46
47 vfmax.vf v8, v4, fa0, v0.t
48 # CHECK-INST: vfmax.vf v8, v4, fa0, v0.t
49 # CHECK-ENCODING: [0x57,0x54,0x45,0x18]
50 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 57 54 45 18 <unknown>
52
53 vfmax.vf v8, v4, fa0
54 # CHECK-INST: vfmax.vf v8, v4, fa0
55 # CHECK-ENCODING: [0x57,0x54,0x45,0x1a]
56 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 57 54 45 1a <unknown>
```

llvm/test/MC/RISCV/rvv/fmul.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vfmul.vv v8, v4, v20, v0.t
12 # CHECK-INST: vfmul.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x14,0x4a,0x90]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 14 4a 90 <unknown>
16
17 vfmul.vv v8, v4, v20
18 # CHECK-INST: vfmul.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x14,0x4a,0x92]
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 4a 92 <unknown>
22
23 vfmul.vf v8, v4, fa0, v0.t
24 # CHECK-INST: vfmul.vf v8, v4, fa0, v0.t
25 # CHECK-ENCODING: [0x57,0x54,0x45,0x90]
                    ires the following: 'F' (Single-Precision Floating-Point), 'V'
                    nknown>
29 vfmul.vf v8, v4, fa0
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
30 # CHECK-INST: vfmul.vf v8, v4, fa0
31 # CHECK-ENCODING: [0x57,0x54,0x45,0x92]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 54 45 92 <unknown>
34
35 vfwmul.vv v8, v4, v20, v0.t
36 # CHECK-INST: vfwmul.vv v8, v4, v20, v0.t
37 # CHECK-ENCODING: [0x57,0x14,0x4a,0xe0]
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
39 # CHECK-UNKNOWN: 57 14 4a e0 <unknown>
40
41 vfwmul.vv v8, v4, v20
42 # CHECK-INST: vfwmul.vv v8, v4, v20
43 # CHECK-ENCODING: [0x57,0x14,0x4a,0xe2]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
45 # CHECK-UNKNOWN: 57 14 4a e2 <unknown>
46
47 vfwmul.vf v8, v4, fa0, v0.t
48 # CHECK-INST: vfwmul.vf v8, v4, fa0, v0.t
49 # CHECK-ENCODING: [0x57,0x54,0x45,0xe0]
50 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
51 # CHECK-UNKNOWN: 57 54 45 e0 <unknown>
52
53 vfwmul.vf v8, v4, fa0
54 # CHECK-INST: vfwmul.vf v8, v4, fa0
55 # CHECK-ENCODING: [0x57,0x54,0x45,0xe2]
56 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
57 # CHECK-UNKNOWN: 57 54 45 e2 <unknown>
```

llvm/test/MC/RISCV/rvv/fmv.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vfmv.v.f v8, fa0
12 # CHECK-INST: vfmv.v.f v8, fa0
13 # CHECK-ENCODING: [0x57,0x54,0x05,0x5e]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
15 # CHECK-UNKNOWN: 57 54 05 5e <unknown>
16
17 vfmv.f.s fa0, v4

   40,0x42]
   ires the following: 'F' (Single-Precision Floating-Point), 'V'

21 # CHECK-UNKNOWN: 57 15 40 42 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
22
23 vfmv.s.f v8, fa0
24 # CHECK-INST: vfmv.s.f v8, fa0
25 # CHECK-ENCODING: [0x57,0x54,0x05,0x42]
26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
27 # CHECK-UNKNOWN: 57 54 05 42 <unknown>
```

llvm/test/MC/RISCV/rvv/fothers.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vfsqrt.v v8, v4, v0.t
12 # CHECK-INST: vfsqrt.v v8, v4, v0.t
13 # CHECK-ENCODING: [0x57,0x14,0x40,0x8c]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
15 # CHECK-UNKNOWN: 57 14 40 8c <unknown>
16
17 vfsqrt.v v8, v4
18 # CHECK-INST: vfsqrt.v v8, v4
19 # CHECK-ENCODING: [0x57,0x14,0x40,0x8e]
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 40 8e <unknown>
22
23 vfclass.v v8, v4, v0.t
24 # CHECK-INST: vfclass.v v8, v4, v0.t
25 # CHECK-ENCODING: [0x57,0x14,0x48,0x8c]
26 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
27 # CHECK-UNKNOWN: 57 14 48 8c <unknown>
28
29 vfclass.v v8, v4
30 # CHECK-INST: vfclass.v v8, v4
31 # CHECK-ENCODING: [0x57,0x14,0x48,0x8e]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 14 48 8e <unknown>
34
35 vfmerge.vfm v8, v4, fa0, v0
36 # CHECK-INST: vfmerge.vfm v8, v4, fa0, v0
37 # CHECK-ENCODING: [0x57,0x54,0x45,0x5c]
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
39 # CHECK-UNKNOWN: 57 54 45 5c <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

n.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
# RUN:         | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vfredosum.vs v8, v4, v20, v0.t
# CHECK-INST: vfredosum.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0x0c]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 0c <unknown>

vfredosum.vs v8, v4, v20
# CHECK-INST: vfredosum.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0x0e]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 0e <unknown>

vfredsum.vs v8, v4, v20, v0.t
# CHECK-INST: vfredsum.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0x04]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 04 <unknown>

vfredsum.vs v8, v4, v20
# CHECK-INST: vfredsum.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0x06]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 06 <unknown>

vfredmax.vs v8, v4, v20, v0.t
# CHECK-INST: vfredmax.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0x1c]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 1c <unknown>

vfredmax.vs v8, v4, v20
# CHECK-INST: vfredmax.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0x1e]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 1e <unknown>

vfredmin.vs v8, v4, v20, v0.t
# CHECK-INST: vfredmin.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0x14]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
```



```
nknown>

# CHECK-INST: vfredmin.vs v8, v4, v20
```

```
# CHECK-ENCODING: [0x57,0x14,0x4a,0x16]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 16 <unknown>

vfwredosum.vs v8, v4, v20, v0.t
# CHECK-INST: vfwredosum.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0xcc]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a cc <unknown>

vfwredosum.vs v8, v4, v20
# CHECK-INST: vfwredosum.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0xce]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a ce <unknown>

vfwredsum.vs v8, v4, v20, v0.t
# CHECK-INST: vfwredsum.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0xc4]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a c4 <unknown>

vfwredsum.vs v8, v4, v20
# CHECK-INST: vfwredsum.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0xc6]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a c6 <unknown>
```

llvm/test/MC/RISCV/rvv/fsub.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
# RUN:         | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
# RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vfsub.vv v8, v4, v20, v0.t
# CHECK-INST: vfsub.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0x08]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a 08 <unknown>

vfsub.vv v8, v4, v20
# CHECK-INST: vfsub.vv v8, v4, v20
4a,0x0a]
ires the following: 'F' (Single-Precision Floating-Point), 'V'

nknown>

```

```
vfsub.vf v8, v4, fa0, v0.t
# CHECK-INST: vfsub.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x08]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 08 <unknown>

vfsub.vf v8, v4, fa0
# CHECK-INST: vfsub.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x0a]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 0a <unknown>

vfrsub.vf v8, v4, fa0, v0.t
# CHECK-INST: vfrsub.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0x9c]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 9c <unknown>

vfrsub.vf v8, v4, fa0
# CHECK-INST: vfrsub.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0x9e]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 9e <unknown>

vfwsub.vv v8, v4, v20, v0.t
# CHECK-INST: vfwsub.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x14,0x4a,0xc8]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a c8 <unknown>

vfwsub.vv v8, v4, v20
# CHECK-INST: vfwsub.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x14,0x4a,0xca]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 14 4a ca <unknown>

vfwsub.vf v8, v4, fa0, v0.t
# CHECK-INST: vfwsub.vf v8, v4, fa0, v0.t
# CHECK-ENCODING: [0x57,0x54,0x45,0xc8]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 c8 <unknown>

vfwsub.vf v8, v4, fa0
# CHECK-INST: vfwsub.vf v8, v4, fa0
# CHECK-ENCODING: [0x57,0x54,0x45,0xca]
# CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 54 45 ca <unknown>

vfwsub.wv v8, v4, v20, v0.t
                          v20, v0.t
                          4a,0xd8]
                          ires the following: 'F' (Single-Precision Floating-Point), 'V'
# CHECK-UNKNOWN: 57 14 4a d8 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
76
77  vfwsub.wv v8, v4, v20
78  # CHECK-INST: vfwsub.wv v8, v4, v20
79  # CHECK-ENCODING: [0x57,0x14,0x4a,0xda]
80  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
81  # CHECK-UNKNOWN: 57 14 4a da <unknown>
82
83  vfwsub.wf v8, v4, fa0, v0.t
84  # CHECK-INST: vfwsub.wf v8, v4, fa0, v0.t
85  # CHECK-ENCODING: [0x57,0x54,0x45,0xd8]
86  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
87  # CHECK-UNKNOWN: 57 54 45 d8 <unknown>
88
89  vfwsub.wf v8, v4, fa0
90  # CHECK-INST: vfwsub.wf v8, v4, fa0
91  # CHECK-ENCODING: [0x57,0x54,0x45,0xda]
92  # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
    (Vector Instructions)
93  # CHECK-UNKNOWN: 57 54 45 da <unknown>
```

llvm/test/MC/RISCV/rvv/invalid.s

View Options

This file was added.

```
1  # RUN: not llvm-mc -triple=riscv64 --mattr=+experimental-v --mattr=+f < %s 2>&1 \
2  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
3
4  vsetvli a2, a0, e31
5  # CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]
6
```

fpallares — Not Done

Suggestion: If you use `CHECK-LABEL` between the regular `CHECK` you can avoid these directives to succeed by matching lines corresponding to a later instruction. Like this:

```
vsetvli a2, a0, e31
// CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]
// CHECK-ERROR-LABEL: vsetvli a2, a0, e31
```

(Note the `CHECK-LABEL` is placed *after* the `CHECK` since it appears in this order in the output)

Without the `CHECK-LABELS` the test will still fail when something is wrong, but with them FileCheck will be able to point exactly where the problem occurred.

More info on this in the corresponding section of the FileCheck manual.

HsiangKai Author — ✔ Done

Got it. Thanks.

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
# CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]

vsetvli a2, a0, e32,m16
# CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]

vsetvli a2, a0, e2048,m8
# CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]

vsetvli a2, a0, e1,m8
# CHECK-ERROR: operand must be e[8|16|32|64|128|256|512|1024],m[1|2|4|8]

vadd.vv v1, v3, v2, v4.t
# CHECK-ERROR: operand must be v0.t

vadd.vv v1, v3, v2, v0
# CHECK-ERROR: expected '.t' suffix

viota.m v0, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: viota.m v0, v2, v0.t

viota.m v2, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: viota.m v2, v2

vfwcvt.xu.f.v v0, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwcvt.xu.f.v v0, v2, v0.t

vfwcvt.xu.f.v v2, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwcvt.xu.f.v v2, v2

vfwcvt.x.f.v v0, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwcvt.x.f.v v0, v2, v0.t

vfwcvt.x.f.v v2, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwcvt.x.f.v v2, v2

vfwcvt.f.xu.v v0, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwcvt.f.xu.v v0, v2, v0.t

vfwcvt.f.xu.v v2, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwcvt.f.xu.v v2, v2

vfwcvt.f.x.v v0, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwcvt.f.x.v v0, v2, v0.t

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

vector register group cannot overlap the source vector registe

.v v2, v2

```

```
68  vfwcvt.f.f.v v0, v2, v0.t
69  # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
70  # CHECK-ERROR-LABEL: vfwcvt.f.f.v v0, v2, v0.t
71
72  vfwcvt.f.f.v v2, v2
73  # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
74  # CHECK-ERROR-LABEL: vfwcvt.f.f.v v2, v2
75
76  vslideup.vx v0, v2, a0, v0.t
77  # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
78  # CHECK-ERROR-LABEL: vslideup.vx v0, v2, a0, v0.t
79
80  vslideup.vx v2, v2, a0
81  # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
82  # CHECK-ERROR-LABEL: vslideup.vx v2, v2, a0
83
84  vslideup.vi v0, v2, 31, v0.t
85  # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
86  # CHECK-ERROR-LABEL: vslideup.vi v0, v2, 31, v0.t
87
88  vslideup.vi v2, v2, 31
89  # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
90  # CHECK-ERROR-LABEL: vslideup.vi v2, v2, 31
91
92  vslide1up.vx v0, v2, a0, v0.t
93  # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
94  # CHECK-ERROR-LABEL: vslide1up.vx v0, v2, a0, v0.t
95
96  vslide1up.vx v2, v2, a0
97  # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
98  # CHECK-ERROR-LABEL: vslide1up.vx v2, v2, a0
99
100 vnsrl.wv v2, v2, v4
101 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
102 # CHECK-ERROR-LABEL: vnsrl.wv v2, v2, v4
103
104 vnsrl.wx v2, v2, a0
105 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
106 # CHECK-ERROR-LABEL: vnsrl.wx v2, v2, a0
107
108 vnsrl.wi v2, v2, 31
109 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
110 # CHECK-ERROR-LABEL: vnsrl.wi v2, v2, 31
111
112 vnsra.wv v2, v2, v4
113 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
114 # CHECK-ERROR-LABEL: vnsra.wv v2, v2, v4
115
116 vnsra.wx v2, v2, a0
                    vector register group cannot overlap the source vector registe
                  2, v2, a0
120 vnsra.wi v2, v2, 31
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
121 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
122 # CHECK-ERROR-LABEL: vnsra.wi v2, v2, 31
123
124 vnclipu.wv v2, v2, v4
125 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
126 # CHECK-ERROR-LABEL: vnclipu.wv v2, v2, v4
127
128 vnclipu.wx v2, v2, a0
129 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
130 # CHECK-ERROR-LABEL: vnclipu.wx v2, v2, a0
131
132 vnclipu.wi v2, v2, 31
133 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
134 # CHECK-ERROR-LABEL: vnclipu.wi v2, v2, 31
135
136 vnclip.wv v2, v2, v4
137 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
138 # CHECK-ERROR-LABEL: vnclip.wv v2, v2, v4
139
140 vnclip.wx v2, v2, a0
141 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
142 # CHECK-ERROR-LABEL: vnclip.wx v2, v2, a0
143
144 vnclip.wi v2, v2, 31
145 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
146 # CHECK-ERROR-LABEL: vnclip.wi v2, v2, 31
147
148 vfncvt.xu.f.w v2, v2
149 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
150 # CHECK-ERROR-LABEL: vfncvt.xu.f.w v2, v2
151
152 vfncvt.x.f.w v2, v2
153 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
154 # CHECK-ERROR-LABEL: vfncvt.x.f.w v2, v2
155
156 vfncvt.f.xu.w v2, v2
157 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
158 # CHECK-ERROR-LABEL: vfncvt.f.xu.w v2, v2
159
160 vfncvt.f.x.w v2, v2
161 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
162 # CHECK-ERROR-LABEL: vfncvt.f.x.w v2, v2
163
164 vfncvt.f.f.w v2, v2
165 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
166 # CHECK-ERROR-LABEL: vfncvt.f.f.w v2, v2
```



```
    vector register group cannot overlap the source vector registe
170 # CHECK-ERROR-LABEL: vfncvt.rod.f.f.w v2, v2
```

```
171
172 vrgather.vv v0, v2, v4, v0.t
173 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
174 # CHECK-ERROR-LABEL: vrgather.vv v0, v2, v4, v0.t
175
176 vrgather.vv v2, v2, v4
177 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
178 # CHECK-ERROR-LABEL: vrgather.vv v2, v2, v4
179
180 vrgather.vx v0, v2, a0, v0.t
181 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
182 # CHECK-ERROR-LABEL: vrgather.vx v0, v2, a0, v0.t
183
184 vrgather.vx v2, v2, a0
185 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
186 # CHECK-ERROR-LABEL: vrgather.vx v2, v2, a0
187
188 vrgather.vi v0, v2, 31, v0.t
189 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
190 # CHECK-ERROR-LABEL: vrgather.vi v0, v2, 31, v0.t
191
192 vrgather.vi v2, v2, 31
193 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
194 # CHECK-ERROR-LABEL: vrgather.vi v2, v2, 31
195
196 vwaddu.vv v0, v2, v4, v0.t
197 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
198 # CHECK-ERROR-LABEL: vwaddu.vv v0, v2, v4, v0.t
199
200 vwaddu.vv v2, v2, v4
201 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
202 # CHECK-ERROR-LABEL: vwaddu.vv v2, v2, v4
203
204 vwsubu.vv v0, v2, v4, v0.t
205 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
206 # CHECK-ERROR-LABEL: vwsubu.vv v0, v2, v4, v0.t
207
208 vwsubu.vv v2, v2, v4
209 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
210 # CHECK-ERROR-LABEL: vwsubu.vv v2, v2, v4
211
212 vwadd.vv v0, v2, v4, v0.t
213 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
214 # CHECK-ERROR-LABEL: vwadd.vv v0, v2, v4, v0.t
215
216 vwadd.vv v2, v2, v4
217 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
218 # CHECK-ERROR-LABEL: vwadd.vv v2, v2, v4
219
220 vwsub.vv v0, v2, v4, v0.t
221 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
                                                    0, v2, v4, v0.t
```



```
                                                    vector register group cannot overlap the source vector registe
    r group.
```

```
226 # CHECK-ERROR-LABEL: vwsub.vv v2, v2, v4
227
228 vwmul.vv v0, v2, v4, v0.t
229 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
230 # CHECK-ERROR-LABEL: vwmul.vv v0, v2, v4, v0.t
231
232 vwmul.vv v2, v2, v4
233 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
234 # CHECK-ERROR-LABEL: vwmul.vv v2, v2, v4
235
236 vwmulu.vv v0, v2, v4, v0.t
237 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
238 # CHECK-ERROR-LABEL: vwmulu.vv v0, v2, v4, v0.t
239
240 vwmulu.vv v2, v2, v4
241 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
242 # CHECK-ERROR-LABEL: vwmulu.vv v2, v2, v4
243
244 vwmulsu.vv v0, v2, v4, v0.t
245 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
246 # CHECK-ERROR-LABEL: vwmulsu.vv v0, v2, v4, v0.t
247
248 vwmulsu.vv v2, v2, v4
249 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
250 # CHECK-ERROR-LABEL: vwmulsu.vv v2, v2, v4
251
252 vwmaccu.vv v0, v4, v2, v0.t
253 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
254 # CHECK-ERROR-LABEL: vwmaccu.vv v0, v4, v2, v0.t
255
256 vwmaccu.vv v2, v4, v2
257 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
258 # CHECK-ERROR-LABEL: vwmaccu.vv v2, v4, v2
259
260 vwmacc.vv v0, v4, v2, v0.t
261 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
262 # CHECK-ERROR-LABEL: vwmacc.vv v0, v4, v2, v0.t
263
264 vwmacc.vv v2, v4, v2
265 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
266 # CHECK-ERROR-LABEL: vwmacc.vv v2, v4, v2
267
268 vwmaccsu.vv v0, v4, v2, v0.t
269 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
270 # CHECK-ERROR-LABEL: vwmaccsu.vv v0, v4, v2, v0.t
271
272 vwmaccsu.vv v2, v4, v2
273 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
274 # CHECK-ERROR-LABEL: vwmaccsu.vv v2, v4, v2
275
276 vfwadd.vv v0, v2, v4, v0.t
                                vector register group cannot overlap the mask register.
                                v0, v2, v4, v0.t
281 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
r group.
# CHECK-ERROR-LABEL: vfwadd.vv v2, v2, v4

vfwsub.vv v0, v2, v4, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwsub.vv v0, v2, v4, v0.t

vfwsub.vv v2, v2, v4
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwsub.vv v2, v2, v4

vfwmul.vv v0, v2, v4, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwmul.vv v0, v2, v4, v0.t

vfwmul.vv v2, v2, v4
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwmul.vv v2, v2, v4

vfwmacc.vv v0, v4, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwmacc.vv v0, v4, v2, v0.t

vfwmacc.vv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwmacc.vv v2, v4, v2

vfwnmacc.vv v0, v4, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwnmacc.vv v0, v4, v2, v0.t

vfwnmacc.vv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwnmacc.vv v2, v4, v2

vfwmsac.vv v0, v4, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwmsac.vv v0, v4, v2, v0.t

vfwmsac.vv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwmsac.vv v2, v4, v2

vfwnmsac.vv v0, v4, v2, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwnmsac.vv v0, v4, v2, v0.t

vfwnmsac.vv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwnmsac.vv v2, v4, v2

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

vector register group cannot overlap the mask register.
v0, v2, a0, v0.t

vwaddu.vx v2, v2, a0
```

```
337 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
338 # CHECK-ERROR-LABEL: vwaddu.vx v2, v2, a0
339
340 vwsubu.vx v0, v2, a0, v0.t
341 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
342 # CHECK-ERROR-LABEL: vwsubu.vx v0, v2, a0, v0.t
343
344 vwsubu.vx v2, v2, a0
345 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
346 # CHECK-ERROR-LABEL: vwsubu.vx v2, v2, a0
347
348 vwadd.vx v0, v2, a0, v0.t
349 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
350 # CHECK-ERROR-LABEL: vwadd.vx v0, v2, a0, v0.t
351
352 vwadd.vx v2, v2, a0
353 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
354 # CHECK-ERROR-LABEL: vwadd.vx v2, v2, a0
355
356 vwsub.vx v0, v2, a0, v0.t
357 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
358 # CHECK-ERROR-LABEL: vwsub.vx v0, v2, a0, v0.t
359
360 vwsub.vx v2, v2, a0
361 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
362 # CHECK-ERROR-LABEL: vwsub.vx v2, v2, a0
363
364 vwmul.vx v0, v2, a0, v0.t
365 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
366 # CHECK-ERROR-LABEL: vwmul.vx v0, v2, a0, v0.t
367
368 vwmul.vx v2, v2, a0
369 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
370 # CHECK-ERROR-LABEL: vwmul.vx v2, v2, a0
371
372 vwmulu.vx v0, v2, a0, v0.t
373 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
374 # CHECK-ERROR-LABEL: vwmulu.vx v0, v2, a0, v0.t
375
376 vwmulu.vx v2, v2, a0
377 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
378 # CHECK-ERROR-LABEL: vwmulu.vx v2, v2, a0
379
380 vwmulsu.vx v0, v2, a0, v0.t
381 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
382 # CHECK-ERROR-LABEL: vwmulsu.vx v0, v2, a0, v0.t
383
384 vwmulsu.vx v2, v2, a0
385 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
386 # CHECK-ERROR-LABEL: vwmulsu.vx v2, v2, a0
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
    vector register group cannot overlap the mask register.
    v0, a0, v2, v0.t
391
```

```
392 vwmaccu.vx v2, a0, v2
393 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
394 # CHECK-ERROR-LABEL: vwmaccu.vx v2, a0, v2
395
396 vwmacc.vx v0, a0, v2, v0.t
397 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
398 # CHECK-ERROR-LABEL: vwmacc.vx v0, a0, v2, v0.t
399
400 vwmacc.vx v2, a0, v2
401 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
402 # CHECK-ERROR-LABEL: vwmacc.vx v2, a0, v2
403
404 vwmaccsu.vx v0, a0, v2, v0.t
405 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
406 # CHECK-ERROR-LABEL: vwmaccsu.vx v0, a0, v2, v0.t
407
408 vwmaccsu.vx v2, a0, v2
409 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
410 # CHECK-ERROR-LABEL: vwmaccsu.vx v2, a0, v2
411
412 vwmaccus.vx v0, a0, v2, v0.t
413 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
414 # CHECK-ERROR-LABEL: vwmaccus.vx v0, a0, v2, v0.t
415
416 vwmaccus.vx v2, a0, v2
417 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
418 # CHECK-ERROR-LABEL: vwmaccus.vx v2, a0, v2
419
420 vfwadd.vf v0, v2, fa0, v0.t
421 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
422 # CHECK-ERROR-LABEL: vfwadd.vf v0, v2, fa0, v0.t
423
424 vfwadd.vf v2, v2, fa0
425 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
426 # CHECK-ERROR-LABEL: vfwadd.vf v2, v2, fa0
427
428 vfwsub.vf v0, v2, fa0, v0.t
429 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
430 # CHECK-ERROR-LABEL: vfwsub.vf v0, v2, fa0, v0.t
431
432 vfwsub.vf v2, v2, fa0
433 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
434 # CHECK-ERROR-LABEL: vfwsub.vf v2, v2, fa0
435
436 vfwmul.vf v0, v2, fa0, v0.t
437 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
438 # CHECK-ERROR-LABEL: vfwmul.vf v0, v2, fa0, v0.t
439
440 vfwmul.vf v2, v2, fa0
441 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
                                                    v2, v2, fa0
                                                    vector register group cannot overlap the mask register.
446 # CHECK-ERROR-LABEL: vfwmacc.vf v0, fa0, v2, v0.t
```

```
447
448 vfwmacc.vf v2, fa0, v2
449 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
450 # CHECK-ERROR-LABEL: vfwmacc.vf v2, fa0, v2
451
452 vfwnmacc.vf v0, fa0, v2, v0.t
453 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
454 # CHECK-ERROR-LABEL: vfwnmacc.vf v0, fa0, v2, v0.t
455
456 vfwnmacc.vf v2, fa0, v2
457 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
458 # CHECK-ERROR-LABEL: vfwnmacc.vf v2, fa0, v2
459
460 vfwmsac.vf v0, fa0, v2, v0.t
461 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
462 # CHECK-ERROR-LABEL: vfwmsac.vf v0, fa0, v2, v0.t
463
464 vfwmsac.vf v2, fa0, v2
465 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
466 # CHECK-ERROR-LABEL: vfwmsac.vf v2, fa0, v2
467
468 vfwnmsac.vf v0, fa0, v2, v0.t
469 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
470 # CHECK-ERROR-LABEL: vfwnmsac.vf v0, fa0, v2, v0.t
471
472 vfwnmsac.vf v2, fa0, v2
473 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
474 # CHECK-ERROR-LABEL: vfwnmsac.vf v2, fa0, v2
475
476 vcompress.vm v2, v2, v4
477 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
478 # CHECK-ERROR-LABEL: vcompress.vm v2, v2, v4
479
480 vwaddu.wv v0, v2, v4, v0.t
481 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
482 # CHECK-ERROR-LABEL: vwaddu.wv v0, v2, v4, v0.t
483
484 vwaddu.wv v2, v4, v2
485 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
486 # CHECK-ERROR-LABEL: vwaddu.wv v2, v4, v2
487
488 vwsubu.wv v0, v2, v4, v0.t
489 # CHECK-ERROR: The destination vector register group cannot overlap the mask register.
490 # CHECK-ERROR-LABEL: vwsubu.wv v0, v2, v4, v0.t
491
492 vwsubu.wv v2, v4, v2
493 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
    r group.
494 # CHECK-ERROR-LABEL: vwsubu.wv v2, v4, v2
495
496 vwadd.wv v0, v2, v4, v0.t
                          vector register group cannot overlap the mask register.
                          0, v2, v4, v0.t
501 # CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
r group.
# CHECK-ERROR-LABEL: vwadd.wv v2, v4, v2

vwsub.wv v0, v2, v4, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vwsub.wv v0, v2, v4, v0.t

vwsub.wv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vwsub.wv v2, v4, v2

vfwadd.wv v0, v2, v4, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwadd.wv v0, v2, v4, v0.t

vfwadd.wv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwadd.wv v2, v4, v2

vfwsub.wv v0, v2, v4, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwsub.wv v0, v2, v4, v0.t

vfwsub.wv v2, v4, v2
# CHECK-ERROR: The destination vector register group cannot overlap the source vector registe
r group.
# CHECK-ERROR-LABEL: vfwsub.wv v2, v4, v2

vwaddu.wx v0, v2, a0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vwaddu.wx v0, v2, a0, v0.t

vwsubu.wx v0, v2, a0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vwsubu.wx v0, v2, a0, v0.t

vwadd.wx v0, v2, a0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vwadd.wx v0, v2, a0, v0.t

vwsub.wx v0, v2, a0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vwsub.wx v0, v2, a0, v0.t

vfwadd.wf v0, v2, fa0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwadd.wf v0, v2, fa0, v0.t

vfwsub.wf v0, v2, fa0, v0.t
# CHECK-ERROR: The destination vector register group cannot overlap the mask register.
# CHECK-ERROR-LABEL: vfwsub.wf v0, v2, fa0, v0.t
```

llvm/test/MC/RISCV/rvv/load.s View Options

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
-show-encoding --mattr=+experimental-v < %s \
# RUN:          | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
```

```
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vlb.v v8, (a0), v0.t
12 # CHECK-INST: vlb.v v8, (a0), v0.t
13 # CHECK-ENCODING: [0x07,0x04,0x05,0x10]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 07 04 05 10 <unknown>
16
17 vlb.v v8, (a0)
18 # CHECK-INST: vlb.v v8, (a0)
19 # CHECK-ENCODING: [0x07,0x04,0x05,0x12]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 07 04 05 12 <unknown>
22
23 vlh.v v8, (a0), v0.t
24 # CHECK-INST: vlh.v v8, (a0), v0.t
25 # CHECK-ENCODING: [0x07,0x54,0x05,0x10]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 07 54 05 10 <unknown>
28
29 vlh.v v8, (a0)
30 # CHECK-INST: vlh.v v8, (a0)
31 # CHECK-ENCODING: [0x07,0x54,0x05,0x12]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 07 54 05 12 <unknown>
34
35 vlw.v v8, (a0), v0.t
36 # CHECK-INST: vlw.v v8, (a0), v0.t
37 # CHECK-ENCODING: [0x07,0x64,0x05,0x10]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 07 64 05 10 <unknown>
40
41 vlw.v v8, (a0)
42 # CHECK-INST: vlw.v v8, (a0)
43 # CHECK-ENCODING: [0x07,0x64,0x05,0x12]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 07 64 05 12 <unknown>
46
47 vlbu.v v8, (a0), v0.t
48 # CHECK-INST: vlbu.v v8, (a0), v0.t
49 # CHECK-ENCODING: [0x07,0x04,0x05,0x00]
50 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 07 04 05 00 <unknown>
52
53 vlbu.v v8, (a0)
54 # CHECK-INST: vlbu.v v8, (a0)
55 # CHECK-ENCODING: [0x07,0x04,0x05,0x02]
56 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 07 04 05 02 <unknown>
58
                                          v0.t
                                          05,0x00]
                                          ires the following: 'V' (Vector Instructions)
63 # CHECK-UNKNOWN: 07 54 05 00 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
64
65  vlhu.v v8, (a0)
66  # CHECK-INST: vlhu.v v8, (a0)
67  # CHECK-ENCODING: [0x07,0x54,0x05,0x02]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 07 54 05 02 <unknown>
70
71  vlwu.v v8, (a0), v0.t
72  # CHECK-INST: vlwu.v v8, (a0), v0.t
73  # CHECK-ENCODING: [0x07,0x64,0x05,0x00]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 07 64 05 00 <unknown>
76
77  vlwu.v v8, (a0)
78  # CHECK-INST: vlwu.v v8, (a0)
79  # CHECK-ENCODING: [0x07,0x64,0x05,0x02]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 07 64 05 02 <unknown>
82
83  vlbff.v v8, (a0), v0.t
84  # CHECK-INST: vlbff.v v8, (a0), v0.t
85  # CHECK-ENCODING: [0x07,0x04,0x05,0x11]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 07 04 05 11 <unknown>
88
89  vlbff.v v8, (a0)
90  # CHECK-INST: vlbff.v v8, (a0)
91  # CHECK-ENCODING: [0x07,0x04,0x05,0x13]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 07 04 05 13 <unknown>
94
95  vlhff.v v8, (a0), v0.t
96  # CHECK-INST: vlhff.v v8, (a0), v0.t
97  # CHECK-ENCODING: [0x07,0x54,0x05,0x11]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 07 54 05 11 <unknown>
100
101 vlhff.v v8, (a0)
102 # CHECK-INST: vlhff.v v8, (a0)
103 # CHECK-ENCODING: [0x07,0x54,0x05,0x13]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 07 54 05 13 <unknown>
106
107 vlwff.v v8, (a0), v0.t
108 # CHECK-INST: vlwff.v v8, (a0), v0.t
109 # CHECK-ENCODING: [0x07,0x64,0x05,0x11]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 07 64 05 11 <unknown>
112
113 vlwff.v v8, (a0)
114 # CHECK-INST: vlwff.v v8, (a0)
115 # CHECK-ENCODING: [0x07,0x64,0x05,0x13]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 07 64 05 13 <unknown>
118
119 vlbuff.v v8, (a0), v0.t
                       , v0.t
                       05,0x01]
                       ires the following: 'V' (Vector Instructions)
                       nknown>
124
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
125 vlbuff.v v8, (a0)
126 # CHECK-INST: vlbuff.v v8, (a0)
127 # CHECK-ENCODING: [0x07,0x04,0x05,0x03]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 07 04 05 03 <unknown>
130
131 vlhuff.v v8, (a0), v0.t
132 # CHECK-INST: vlhuff.v v8, (a0), v0.t
133 # CHECK-ENCODING: [0x07,0x54,0x05,0x01]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 07 54 05 01 <unknown>
136
137 vlhuff.v v8, (a0)
138 # CHECK-INST: vlhuff.v v8, (a0)
139 # CHECK-ENCODING: [0x07,0x54,0x05,0x03]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 07 54 05 03 <unknown>
142
143 vlwuff.v v8, (a0), v0.t
144 # CHECK-INST: vlwuff.v v8, (a0), v0.t
145 # CHECK-ENCODING: [0x07,0x64,0x05,0x01]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 07 64 05 01 <unknown>
148
149 vlwuff.v v8, (a0)
150 # CHECK-INST: vlwuff.v v8, (a0)
151 # CHECK-ENCODING: [0x07,0x64,0x05,0x03]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 07 64 05 03 <unknown>
154
155 vleff.v v8, (a0), v0.t
156 # CHECK-INST: vleff.v v8, (a0), v0.t
157 # CHECK-ENCODING: [0x07,0x74,0x05,0x01]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 07 74 05 01 <unknown>
160
161 vleff.v v8, (a0)
162 # CHECK-INST: vleff.v v8, (a0)
163 # CHECK-ENCODING: [0x07,0x74,0x05,0x03]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 07 74 05 03 <unknown>
166
167 vlsb.v v8, (a0), a1, v0.t
168 # CHECK-INST: vlsb.v v8, (a0), a1, v0.t
169 # CHECK-ENCODING: [0x07,0x04,0xb5,0x18]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
171 # CHECK-UNKNOWN: 07 04 b5 18 <unknown>
172
173 vlsb.v v8, (a0), a1
174 # CHECK-INST: vlsb.v v8, (a0), a1
175 # CHECK-ENCODING: [0x07,0x04,0xb5,0x1a]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
177 # CHECK-UNKNOWN: 07 04 b5 1a <unknown>
178
179 vlsh.v v8, (a0), a1, v0.t
180 # CHECK-INST: vlsh.v v8, (a0), a1, v0.t
                                b5,0x18]
                                ires the following: 'V' (Vector Instructions)
                                nknown>
185 vlsh.v v8, (a0), a1
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
186 # CHECK-INST: vlsh.v v8, (a0), a1
187 # CHECK-ENCODING: [0x07,0x54,0xb5,0x1a]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 07 54 b5 1a <unknown>
190
191 vlsw.v v8, (a0), a1, v0.t
192 # CHECK-INST: vlsw.v v8, (a0), a1, v0.t
193 # CHECK-ENCODING: [0x07,0x64,0xb5,0x18]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 07 64 b5 18 <unknown>
196
197 vlsw.v v8, (a0), a1
198 # CHECK-INST: vlsw.v v8, (a0), a1
199 # CHECK-ENCODING: [0x07,0x64,0xb5,0x1a]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 07 64 b5 1a <unknown>
202
203 vlsbu.v v8, (a0), a1, v0.t
204 # CHECK-INST: vlsbu.v v8, (a0), a1, v0.t
205 # CHECK-ENCODING: [0x07,0x04,0xb5,0x08]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 07 04 b5 08 <unknown>
208
209 vlsbu.v v8, (a0), a1
210 # CHECK-INST: vlsbu.v v8, (a0), a1
211 # CHECK-ENCODING: [0x07,0x04,0xb5,0x0a]
212 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
213 # CHECK-UNKNOWN: 07 04 b5 0a <unknown>
214
215 vlshu.v v8, (a0), a1, v0.t
216 # CHECK-INST: vlshu.v v8, (a0), a1, v0.t
217 # CHECK-ENCODING: [0x07,0x54,0xb5,0x08]
218 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
219 # CHECK-UNKNOWN: 07 54 b5 08 <unknown>
220
221 vlshu.v v8, (a0), a1
222 # CHECK-INST: vlshu.v v8, (a0), a1
223 # CHECK-ENCODING: [0x07,0x54,0xb5,0x0a]
224 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
225 # CHECK-UNKNOWN: 07 54 b5 0a <unknown>
226
227 vlswu.v v8, (a0), a1, v0.t
228 # CHECK-INST: vlswu.v v8, (a0), a1, v0.t
229 # CHECK-ENCODING: [0x07,0x64,0xb5,0x08]
230 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
231 # CHECK-UNKNOWN: 07 64 b5 08 <unknown>
232
233 vlswu.v v8, (a0), a1
234 # CHECK-INST: vlswu.v v8, (a0), a1
235 # CHECK-ENCODING: [0x07,0x64,0xb5,0x0a]
236 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
237 # CHECK-UNKNOWN: 07 64 b5 0a <unknown>
238
239 vlse.v v8, (a0), a1, v0.t
240 # CHECK-INST: vlse.v v8, (a0), a1, v0.t
241 # CHECK-ENCODING: [0x07,0x74,0xb5,0x08]
    ires the following: 'V' (Vector Instructions)
    nknown>

246 # CHECK-INST: vlse.v v8, (a0), a1
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
247 # CHECK-ENCODING: [0x07,0x74,0xb5,0x0a]
248 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
249 # CHECK-UNKNOWN: 07 74 b5 0a <unknown>
250
251 vlxb.v v8, (a0), v4, v0.t
252 # CHECK-INST: vlxb.v v8, (a0), v4, v0.t
253 # CHECK-ENCODING: [0x07,0x04,0x45,0x1c]
254 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
255 # CHECK-UNKNOWN: 07 04 45 1c <unknown>
256
257 vlxb.v v8, (a0), v4
258 # CHECK-INST: vlxb.v v8, (a0), v4
259 # CHECK-ENCODING: [0x07,0x04,0x45,0x1e]
260 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
261 # CHECK-UNKNOWN: 07 04 45 1e <unknown>
262
263 vlxh.v v8, (a0), v4, v0.t
264 # CHECK-INST: vlxh.v v8, (a0), v4, v0.t
265 # CHECK-ENCODING: [0x07,0x54,0x45,0x1c]
266 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
267 # CHECK-UNKNOWN: 07 54 45 1c <unknown>
268
269 vlxh.v v8, (a0), v4
270 # CHECK-INST: vlxh.v v8, (a0), v4
271 # CHECK-ENCODING: [0x07,0x54,0x45,0x1e]
272 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
273 # CHECK-UNKNOWN: 07 54 45 1e <unknown>
274
275 vlxw.v v8, (a0), v4, v0.t
276 # CHECK-INST: vlxw.v v8, (a0), v4, v0.t
277 # CHECK-ENCODING: [0x07,0x64,0x45,0x1c]
278 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
279 # CHECK-UNKNOWN: 07 64 45 1c <unknown>
280
281 vlxw.v v8, (a0), v4
282 # CHECK-INST: vlxw.v v8, (a0), v4
283 # CHECK-ENCODING: [0x07,0x64,0x45,0x1e]
284 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
285 # CHECK-UNKNOWN: 07 64 45 1e <unknown>
286
287 vlxbu.v v8, (a0), v4, v0.t
288 # CHECK-INST: vlxbu.v v8, (a0), v4, v0.t
289 # CHECK-ENCODING: [0x07,0x04,0x45,0x0c]
290 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
291 # CHECK-UNKNOWN: 07 04 45 0c <unknown>
292
293 vlxbu.v v8, (a0), v4
294 # CHECK-INST: vlxbu.v v8, (a0), v4
295 # CHECK-ENCODING: [0x07,0x04,0x45,0x0e]
296 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
297 # CHECK-UNKNOWN: 07 04 45 0e <unknown>
298
299 vlxhu.v v8, (a0), v4, v0.t
300 # CHECK-INST: vlxhu.v v8, (a0), v4, v0.t
301 # CHECK-ENCODING: [0x07,0x54,0x45,0x0c]
302 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    v4
307 # CHECK-ENCODING: [0x07,0x54,0x45,0x0e]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
308 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
309 # CHECK-UNKNOWN: 07 54 45 0e <unknown>
310
311 vlxwu.v v8, (a0), v4, v0.t
312 # CHECK-INST: vlxwu.v v8, (a0), v4, v0.t
313 # CHECK-ENCODING: [0x07,0x64,0x45,0x0c]
314 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
315 # CHECK-UNKNOWN: 07 64 45 0c <unknown>
316
317 vlxwu.v v8, (a0), v4
318 # CHECK-INST: vlxwu.v v8, (a0), v4
319 # CHECK-ENCODING: [0x07,0x64,0x45,0x0e]
320 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
321 # CHECK-UNKNOWN: 07 64 45 0e <unknown>
322
323 vlxe.v v8, (a0), v4, v0.t
324 # CHECK-INST: vlxe.v v8, (a0), v4, v0.t
325 # CHECK-ENCODING: [0x07,0x74,0x45,0x0c]
326 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
327 # CHECK-UNKNOWN: 07 74 45 0c <unknown>
328
329 vlxe.v v8, (a0), v4
330 # CHECK-INST: vlxe.v v8, (a0), v4
331 # CHECK-ENCODING: [0x07,0x74,0x45,0x0e]
332 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
333 # CHECK-UNKNOWN: 07 74 45 0e <unknown>
334
335 vl1r.v v8, (a0)
336 # CHECK-INST: vl1r.v v8, (a0)
337 # CHECK-ENCODING: [0x07,0x74,0x85,0x02]
338 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
339 # CHECK-UNKNOWN: 07 74 85 02 <unknown>
```

llvm/test/MC/RISCV/rvv/macc.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
```

fpallares — Not Done

Looks like `+v` attribute gets passed twice. I'm also not sure we need `+a` and `+c`.

```
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vmacc.vv v8, v20, v4, v0.t
12 # CHECK-INST: vmacc.vv v8, v20, v4, v0.t
13 # CHECK-ENCODING: [0x57,0x24,0x4a,0xb4]
                    ires the following: 'V' (Vector Instructions)
                    nknown>
18 # CHECK-INST: vmacc.vv v8, v20, v4
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
19 # CHECK-ENCODING: [0x57,0x24,0x4a,0xb6]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 24 4a b6 <unknown>
22
23 vmacc.vx v8, a0, v4, v0.t
24 # CHECK-INST: vmacc.vx v8, a0, v4, v0.t
25 # CHECK-ENCODING: [0x57,0x64,0x45,0xb4]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 57 64 45 b4 <unknown>
28
29 vmacc.vx v8, a0, v4
30 # CHECK-INST: vmacc.vx v8, a0, v4
31 # CHECK-ENCODING: [0x57,0x64,0x45,0xb6]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 64 45 b6 <unknown>
34
35 vnmsac.vv v8, v20, v4, v0.t
36 # CHECK-INST: vnmsac.vv v8, v20, v4, v0.t
37 # CHECK-ENCODING: [0x57,0x24,0x4a,0xbc]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 24 4a bc <unknown>
40
41 vnmsac.vv v8, v20, v4
42 # CHECK-INST: vnmsac.vv v8, v20, v4
43 # CHECK-ENCODING: [0x57,0x24,0x4a,0xbe]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 24 4a be <unknown>
46
47 vnmsac.vx v8, a0, v4, v0.t
48 # CHECK-INST: vnmsac.vx v8, a0, v4, v0.t
49 # CHECK-ENCODING: [0x57,0x64,0x45,0xbc]
50 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 57 64 45 bc <unknown>
52
53 vnmsac.vx v8, a0, v4
54 # CHECK-INST: vnmsac.vx v8, a0, v4
55 # CHECK-ENCODING: [0x57,0x64,0x45,0xbe]
56 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 57 64 45 be <unknown>
58
59 vmadd.vv v8, v20, v4, v0.t
60 # CHECK-INST: vmadd.vv v8, v20, v4, v0.t
61 # CHECK-ENCODING: [0x57,0x24,0x4a,0xa4]
62 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63 # CHECK-UNKNOWN: 57 24 4a a4 <unknown>
64
65 vmadd.vv v8, v20, v4
66 # CHECK-INST: vmadd.vv v8, v20, v4
67 # CHECK-ENCODING: [0x57,0x24,0x4a,0xa6]
68 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69 # CHECK-UNKNOWN: 57 24 4a a6 <unknown>
70
71 vmadd.vx v8, a0, v4, v0.t
72 # CHECK-INST: vmadd.vx v8, a0, v4, v0.t
73 # CHECK-ENCODING: [0x57,0x64,0x45,0xa4]
74 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    v4
79 # CHECK-ENCODING: [0x57,0x64,0x45,0xa6]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 64 45 a6 <unknown>
82
83  vnmsub.vv v8, v20, v4, v0.t
84  # CHECK-INST: vnmsub.vv v8, v20, v4, v0.t
85  # CHECK-ENCODING: [0x57,0x24,0x4a,0xac]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 24 4a ac <unknown>
88
89  vnmsub.vv v8, v20, v4
90  # CHECK-INST: vnmsub.vv v8, v20, v4
91  # CHECK-ENCODING: [0x57,0x24,0x4a,0xae]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 24 4a ae <unknown>
94
95  vnmsub.vx v8, a0, v4, v0.t
96  # CHECK-INST: vnmsub.vx v8, a0, v4, v0.t
97  # CHECK-ENCODING: [0x57,0x64,0x45,0xac]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 64 45 ac <unknown>
100
101 vnmsub.vx v8, a0, v4
102 # CHECK-INST: vnmsub.vx v8, a0, v4
103 # CHECK-ENCODING: [0x57,0x64,0x45,0xae]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 64 45 ae <unknown>
106
107 vwmaccu.vv v8, v20, v4, v0.t
108 # CHECK-INST: vwmaccu.vv v8, v20, v4, v0.t
109 # CHECK-ENCODING: [0x57,0x24,0x4a,0xf0]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 24 4a f0 <unknown>
112
113 vwmaccu.vv v8, v20, v4
114 # CHECK-INST: vwmaccu.vv v8, v20, v4
115 # CHECK-ENCODING: [0x57,0x24,0x4a,0xf2]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 24 4a f2 <unknown>
118
119 vwmaccu.vx v8, a0, v4, v0.t
120 # CHECK-INST: vwmaccu.vx v8, a0, v4, v0.t
121 # CHECK-ENCODING: [0x57,0x64,0x45,0xf0]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 64 45 f0 <unknown>
124
125 vwmaccu.vx v8, a0, v4
126 # CHECK-INST: vwmaccu.vx v8, a0, v4
127 # CHECK-ENCODING: [0x57,0x64,0x45,0xf2]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 64 45 f2 <unknown>
130
131 vwmacc.vv v8, v20, v4, v0.t
132 # CHECK-INST: vwmacc.vv v8, v20, v4, v0.t
133 # CHECK-ENCODING: [0x57,0x24,0x4a,0xf4]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 24 4a f4 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
, v4
4a,0xf6]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
141 # CHECK-UNKNOWN: 57 24 4a f6 <unknown>
142
143 vwmacc.vx v8, a0, v4, v0.t
144 # CHECK-INST: vwmacc.vx v8, a0, v4, v0.t
145 # CHECK-ENCODING: [0x57,0x64,0x45,0xf4]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 64 45 f4 <unknown>
148
149 vwmacc.vx v8, a0, v4
150 # CHECK-INST: vwmacc.vx v8, a0, v4
151 # CHECK-ENCODING: [0x57,0x64,0x45,0xf6]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 57 64 45 f6 <unknown>
154
155 vwmaccsu.vv v8, v20, v4, v0.t
156 # CHECK-INST: vwmaccsu.vv v8, v20, v4, v0.t
157 # CHECK-ENCODING: [0x57,0x24,0x4a,0xfc]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 24 4a fc <unknown>
160
161 vwmaccsu.vv v8, v20, v4
162 # CHECK-INST: vwmaccsu.vv v8, v20, v4
163 # CHECK-ENCODING: [0x57,0x24,0x4a,0xfe]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 24 4a fe <unknown>
166
167 vwmaccsu.vx v8, a0, v4, v0.t
168 # CHECK-INST: vwmaccsu.vx v8, a0, v4, v0.t
169 # CHECK-ENCODING: [0x57,0x64,0x45,0xfc]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
171 # CHECK-UNKNOWN: 57 64 45 fc <unknown>
172
173 vwmaccsu.vx v8, a0, v4
174 # CHECK-INST: vwmaccsu.vx v8, a0, v4
175 # CHECK-ENCODING: [0x57,0x64,0x45,0xfe]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
177 # CHECK-UNKNOWN: 57 64 45 fe <unknown>
178
179 vwmaccus.vx v8, a0, v4, v0.t
180 # CHECK-INST: vwmaccus.vx v8, a0, v4, v0.t
181 # CHECK-ENCODING: [0x57,0x64,0x45,0xf8]
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 64 45 f8 <unknown>
184
185 vwmaccus.vx v8, a0, v4
186 # CHECK-INST: vwmaccus.vx v8, a0, v4
187 # CHECK-ENCODING: [0x57,0x64,0x45,0xfa]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 64 45 fa <unknown>
```

llvm/test/MC/RISCV/rvv/mask.s

View Options

This file was added.

```
1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
                                  check-prefixes=CHECK-ENCODING,CHECK-INST
                                  v64 -show-encoding < %s 2>&1 \
                                  check-prefix=CHECK-ERROR
                                  -filetype=obj --mattr=+experimental-v < %s \
6 # RUN:          | llvm-objdump -d --mattr=+experimental-v - \
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
# RUN:        | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vmand.mm v8, v4, v20
# CHECK-INST: vmand.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x66]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 66 <unknown>

vmnand.mm v8, v4, v20
# CHECK-INST: vmnand.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x76]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 76 <unknown>

vmandnot.mm v8, v4, v20
# CHECK-INST: vmandnot.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x62]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 62 <unknown>

vmxor.mm v8, v4, v20
# CHECK-INST: vmxor.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x6e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 6e <unknown>

vmor.mm v8, v4, v20
# CHECK-INST: vmor.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x6a]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 6a <unknown>

vmnor.mm v8, v4, v20
# CHECK-INST: vmnor.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x7a]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 7a <unknown>

vmornot.mm v8, v4, v20
# CHECK-INST: vmornot.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x72]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 72 <unknown>

vmxnor.mm v8, v4, v20
# CHECK-INST: vmxnor.mm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x7e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 7e <unknown>

vpopc.m a2, v4, v0.t
# CHECK-INST: vpopc.m a2, v4, v0.t
# CHECK-ENCODING: [0x57,0x26,0x48,0x40]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                  nknown>

# CHECK-ENCODING: [0x57,0x26,0x48,0x42]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 26 48 42 <unknown>
70
71  vfirst.m a2, v4, v0.t
72  # CHECK-INST: vfirst.m a2, v4, v0.t
73  # CHECK-ENCODING: [0x57,0xa6,0x48,0x40]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 a6 48 40 <unknown>
76
77  vfirst.m a2, v4
78  # CHECK-INST: vfirst.m a2, v4
79  # CHECK-ENCODING: [0x57,0xa6,0x48,0x42]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 a6 48 42 <unknown>
82
83  vmsbf.m v8, v4, v0.t
84  # CHECK-INST: vmsbf.m v8, v4, v0.t
85  # CHECK-ENCODING: [0x57,0xa4,0x40,0x50]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 a4 40 50 <unknown>
88
89  vmsbf.m v8, v4
90  # CHECK-INST: vmsbf.m v8, v4
91  # CHECK-ENCODING: [0x57,0xa4,0x40,0x52]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 a4 40 52 <unknown>
94
95  vmsif.m v8, v4, v0.t
96  # CHECK-INST: vmsif.m v8, v4, v0.t
97  # CHECK-ENCODING: [0x57,0xa4,0x41,0x50]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 a4 41 50 <unknown>
100
101 vmsif.m v8, v4
102 # CHECK-INST: vmsif.m v8, v4
103 # CHECK-ENCODING: [0x57,0xa4,0x41,0x52]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 a4 41 52 <unknown>
106
107 vmsof.m v8, v4, v0.t
108 # CHECK-INST: vmsof.m v8, v4, v0.t
109 # CHECK-ENCODING: [0x57,0x24,0x41,0x50]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 24 41 50 <unknown>
112
113 vmsof.m v8, v4
114 # CHECK-INST: vmsof.m v8, v4
115 # CHECK-ENCODING: [0x57,0x24,0x41,0x52]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 24 41 52 <unknown>
118
119 viota.m v8, v4, v0.t
120 # CHECK-INST: viota.m v8, v4, v0.t
121 # CHECK-ENCODING: [0x57,0x24,0x48,0x50]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 24 48 50 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
                                  48,0x52]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
# CHECK-UNKNOWN: 57 24 48 52 <unknown>

vid.v v8, v0.t
# CHECK-INST: vid.v v8, v0.t
# CHECK-ENCODING: [0x57,0xa4,0x08,0x50]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 a4 08 50 <unknown>

vid.v v8
# CHECK-INST: vid.v v8
# CHECK-ENCODING: [0x57,0xa4,0x08,0x52]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 a4 08 52 <unknown>
```

llvm/test/MC/RISCV/rvv/minmax.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
# RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:         | llvm-objdump -d --mattr=+experimental-v - \
# RUN:         | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vminu.vv v8, v4, v20, v0.t
# CHECK-INST: vminu.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x10]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 10 <unknown>

vminu.vv v8, v4, v20
# CHECK-INST: vminu.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x04,0x4a,0x12]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 12 <unknown>

vminu.vx v8, v4, a0, v0.t
# CHECK-INST: vminu.vx v8, v4, a0, v0.t
# CHECK-ENCODING: [0x57,0x44,0x45,0x10]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 10 <unknown>

vminu.vx v8, v4, a0
# CHECK-INST: vminu.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x44,0x45,0x12]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 12 <unknown>

vmin.vv v8, v4, v20, v0.t
# CHECK-INST: vmin.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x14]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 14 <unknown>

# CHECK-INST: vmin.vv v8, v4, v20
```

```
43  # CHECK-ENCODING: [0x57,0x04,0x4a,0x16]
44  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45  # CHECK-UNKNOWN: 57 04 4a 16 <unknown>
46
47  vmin.vx v8, v4, a0, v0.t
48  # CHECK-INST: vmin.vx v8, v4, a0, v0.t
49  # CHECK-ENCODING: [0x57,0x44,0x45,0x14]
50  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51  # CHECK-UNKNOWN: 57 44 45 14 <unknown>
52
53  vmin.vx v8, v4, a0
54  # CHECK-INST: vmin.vx v8, v4, a0
55  # CHECK-ENCODING: [0x57,0x44,0x45,0x16]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 44 45 16 <unknown>
58
59  vmaxu.vv v8, v4, v20, v0.t
60  # CHECK-INST: vmaxu.vv v8, v4, v20, v0.t
61  # CHECK-ENCODING: [0x57,0x04,0x4a,0x18]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 04 4a 18 <unknown>
64
65  vmaxu.vv v8, v4, v20
66  # CHECK-INST: vmaxu.vv v8, v4, v20
67  # CHECK-ENCODING: [0x57,0x04,0x4a,0x1a]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 04 4a 1a <unknown>
70
71  vmaxu.vx v8, v4, a0, v0.t
72  # CHECK-INST: vmaxu.vx v8, v4, a0, v0.t
73  # CHECK-ENCODING: [0x57,0x44,0x45,0x18]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 44 45 18 <unknown>
76
77  vmaxu.vx v8, v4, a0
78  # CHECK-INST: vmaxu.vx v8, v4, a0
79  # CHECK-ENCODING: [0x57,0x44,0x45,0x1a]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 44 45 1a <unknown>
82
83  vmax.vv v8, v4, v20, v0.t
84  # CHECK-INST: vmax.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x04,0x4a,0x1c]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 04 4a 1c <unknown>
88
89  vmax.vv v8, v4, v20
90  # CHECK-INST: vmax.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x04,0x4a,0x1e]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 04 4a 1e <unknown>
94
95  vmax.vx v8, v4, a0, v0.t
96  # CHECK-INST: vmax.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x44,0x45,0x1c]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    0
103 # CHECK-ENCODING: [0x57,0x44,0x45,0x1e]
```

```
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 44 45 1e <unknown>
```

llvm/test/MC/RISCV/rvv/mul.s

View Options

This file was added.

```
 1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
 2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
 3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
 4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
 5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
 6 # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
 7 # RUN:         | FileCheck %s --check-prefix=CHECK-INST
 8 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
 9 # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vmul.vv v8, v4, v20, v0.t
12 # CHECK-INST: vmul.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x24,0x4a,0x94]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 24 4a 94 <unknown>
16
17 vmul.vv v8, v4, v20
18 # CHECK-INST: vmul.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x24,0x4a,0x96]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 24 4a 96 <unknown>
22
23 vmul.vx v8, v4, a0, v0.t
24 # CHECK-INST: vmul.vx v8, v4, a0, v0.t
25 # CHECK-ENCODING: [0x57,0x64,0x45,0x94]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 57 64 45 94 <unknown>
28
29 vmul.vx v8, v4, a0
30 # CHECK-INST: vmul.vx v8, v4, a0
31 # CHECK-ENCODING: [0x57,0x64,0x45,0x96]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 64 45 96 <unknown>
34
35 vmulh.vv v8, v4, v20, v0.t
36 # CHECK-INST: vmulh.vv v8, v4, v20, v0.t
37 # CHECK-ENCODING: [0x57,0x24,0x4a,0x9c]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 24 4a 9c <unknown>
40
41 vmulh.vv v8, v4, v20
42 # CHECK-INST: vmulh.vv v8, v4, v20
43 # CHECK-ENCODING: [0x57,0x24,0x4a,0x9e]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 24 4a 9e <unknown>
46
47 vmulh.vx v8, v4, a0, v0.t
48 # CHECK-INST: vmulh.vx v8, v4, a0, v0.t
   # 45,0x9c]
   # ires the following: 'V' (Vector Instructions)
   # nknown>
53 vmulh.vx v8, v4, a0
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
54  # CHECK-INST: vmulh.vx v8, v4, a0
55  # CHECK-ENCODING: [0x57,0x64,0x45,0x9e]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 64 45 9e <unknown>
58
59  vmulhu.vv v8, v4, v20, v0.t
60  # CHECK-INST: vmulhu.vv v8, v4, v20, v0.t
61  # CHECK-ENCODING: [0x57,0x24,0x4a,0x90]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 24 4a 90 <unknown>
64
65  vmulhu.vv v8, v4, v20
66  # CHECK-INST: vmulhu.vv v8, v4, v20
67  # CHECK-ENCODING: [0x57,0x24,0x4a,0x92]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 24 4a 92 <unknown>
70
71  vmulhu.vx v8, v4, a0, v0.t
72  # CHECK-INST: vmulhu.vx v8, v4, a0, v0.t
73  # CHECK-ENCODING: [0x57,0x64,0x45,0x90]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 64 45 90 <unknown>
76
77  vmulhu.vx v8, v4, a0
78  # CHECK-INST: vmulhu.vx v8, v4, a0
79  # CHECK-ENCODING: [0x57,0x64,0x45,0x92]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 64 45 92 <unknown>
82
83  vmulhsu.vv v8, v4, v20, v0.t
84  # CHECK-INST: vmulhsu.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x24,0x4a,0x98]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 24 4a 98 <unknown>
88
89  vmulhsu.vv v8, v4, v20
90  # CHECK-INST: vmulhsu.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x24,0x4a,0x9a]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 24 4a 9a <unknown>
94
95  vmulhsu.vx v8, v4, a0, v0.t
96  # CHECK-INST: vmulhsu.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x64,0x45,0x98]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 64 45 98 <unknown>
100
101 vmulhsu.vx v8, v4, a0
102 # CHECK-INST: vmulhsu.vx v8, v4, a0
103 # CHECK-ENCODING: [0x57,0x64,0x45,0x9a]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 64 45 9a <unknown>
106
107 vwmul.vv v8, v4, v20, v0.t
108 # CHECK-INST: vwmul.vv v8, v4, v20, v0.t
109 # CHECK-ENCODING: [0x57,0x24,0x4a,0xec]
                  ires the following: 'V' (Vector Instructions)
                  nknown>

114 # CHECK-INST: vwmul.vv v8, v4, v20
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
115 # CHECK-ENCODING: [0x57,0x24,0x4a,0xee]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 24 4a ee <unknown>
118
119 vwmul.vx v8, v4, a0, v0.t
120 # CHECK-INST: vwmul.vx v8, v4, a0, v0.t
121 # CHECK-ENCODING: [0x57,0x64,0x45,0xec]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 64 45 ec <unknown>
124
125 vwmul.vx v8, v4, a0
126 # CHECK-INST: vwmul.vx v8, v4, a0
127 # CHECK-ENCODING: [0x57,0x64,0x45,0xee]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 64 45 ee <unknown>
130
131 vwmulu.vv v8, v4, v20, v0.t
132 # CHECK-INST: vwmulu.vv v8, v4, v20, v0.t
133 # CHECK-ENCODING: [0x57,0x24,0x4a,0xe0]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 24 4a e0 <unknown>
136
137 vwmulu.vv v8, v4, v20
138 # CHECK-INST: vwmulu.vv v8, v4, v20
139 # CHECK-ENCODING: [0x57,0x24,0x4a,0xe2]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 57 24 4a e2 <unknown>
142
143 vwmulu.vx v8, v4, a0, v0.t
144 # CHECK-INST: vwmulu.vx v8, v4, a0, v0.t
145 # CHECK-ENCODING: [0x57,0x64,0x45,0xe0]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 64 45 e0 <unknown>
148
149 vwmulu.vx v8, v4, a0
150 # CHECK-INST: vwmulu.vx v8, v4, a0
151 # CHECK-ENCODING: [0x57,0x64,0x45,0xe2]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 57 64 45 e2 <unknown>
154
155 vwmulsu.vv v8, v4, v20, v0.t
156 # CHECK-INST: vwmulsu.vv v8, v4, v20, v0.t
157 # CHECK-ENCODING: [0x57,0x24,0x4a,0xe8]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 24 4a e8 <unknown>
160
161 vwmulsu.vv v8, v4, v20
162 # CHECK-INST: vwmulsu.vv v8, v4, v20
163 # CHECK-ENCODING: [0x57,0x24,0x4a,0xea]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 24 4a ea <unknown>
166
167 vwmulsu.vx v8, v4, a0, v0.t
168 # CHECK-INST: vwmulsu.vx v8, v4, a0, v0.t
169 # CHECK-ENCODING: [0x57,0x64,0x45,0xe8]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
    nknown>

    , a0
175 # CHECK-ENCODING: [0x57,0x64,0x45,0xea]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
177 # CHECK-UNKNOWN: 57 64 45 ea <unknown>
178
179 vsmul.vv v8, v4, v20, v0.t
180 # CHECK-INST: vsmul.vv v8, v4, v20, v0.t
181 # CHECK-ENCODING: [0x57,0x04,0x4a,0x9c]
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 04 4a 9c <unknown>
184
185 vsmul.vv v8, v4, v20
186 # CHECK-INST: vsmul.vv v8, v4, v20
187 # CHECK-ENCODING: [0x57,0x04,0x4a,0x9e]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 04 4a 9e <unknown>
190
191 vsmul.vx v8, v4, a0, v0.t
192 # CHECK-INST: vsmul.vx v8, v4, a0, v0.t
193 # CHECK-ENCODING: [0x57,0x44,0x45,0x9c]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 57 44 45 9c <unknown>
196
197 vsmul.vx v8, v4, a0
198 # CHECK-INST: vsmul.vx v8, v4, a0
199 # CHECK-ENCODING: [0x57,0x44,0x45,0x9e]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 57 44 45 9e <unknown>
```

llvm/test/MC/RISCV/rvv/mv.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vmv.v.v v8, v20
12 # CHECK-INST: vmv.v.v v8, v20
13 # CHECK-ENCODING: [0x57,0x04,0x0a,0x5e]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 04 0a 5e <unknown>
16
17 vmv.v.x v8, a0
18 # CHECK-INST: vmv.v.x v8, a0
19 # CHECK-ENCODING: [0x57,0x44,0x05,0x5e]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 44 05 5e <unknown>
22
23 vmv.v.i v8, 15
24 # CHECK-INST: vmv.v.i v8, 15
                                07,0x5e]
                                ires the following: 'V' (Vector Instructions)
                                nknown>
29 vmv.x.s a2, v4
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
30 # CHECK-INST: vmv.x.s a2, v4
31 # CHECK-ENCODING: [0x57,0x26,0x40,0x42]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 26 40 42 <unknown>
34
35 vmv.s.x v8, a0
36 # CHECK-INST: vmv.s.x v8, a0
37 # CHECK-ENCODING: [0x57,0x64,0x05,0x42]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 64 05 42 <unknown>
40
41 vmv1r.v v8, v4
42 # CHECK-INST: vmv1r.v v8, v4
43 # CHECK-ENCODING: [0x57,0x34,0x40,0x9e]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 34 40 9e <unknown>
46
47 vmv2r.v v8, v4
48 # CHECK-INST: vmv2r.v v8, v4
49 # CHECK-ENCODING: [0x57,0xb4,0x40,0x9e]
50 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 57 b4 40 9e <unknown>
52
53 vmv4r.v v8, v4
54 # CHECK-INST: vmv4r.v v8, v4
55 # CHECK-ENCODING: [0x57,0xb4,0x41,0x9e]
56 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 57 b4 41 9e <unknown>
58
59 vmv8r.v v8, v24
60 # CHECK-INST: vmv8r.v v8, v24
61 # CHECK-ENCODING: [0x57,0xb4,0x83,0x9f]
62 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63 # CHECK-UNKNOWN: 57 b4 83 9f <unknown>
```

llvm/test/MC/RISCV/rvv/or.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vor.vv v8, v4, v20, v0.t
12 # CHECK-INST: vor.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x04,0x4a,0x28]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 04 4a 28 <unknown>
16
17                                    0
18                                    4a,0x2a]
19                                    ires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 04 4a 2a <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
22
23 vor.vx v8, v4, a0, v0.t
24 # CHECK-INST: vor.vx v8, v4, a0, v0.t
25 # CHECK-ENCODING: [0x57,0x44,0x45,0x28]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 57 44 45 28 <unknown>
28
29 vor.vx v8, v4, a0
30 # CHECK-INST: vor.vx v8, v4, a0
31 # CHECK-ENCODING: [0x57,0x44,0x45,0x2a]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 44 45 2a <unknown>
34
35 vor.vi v8, v4, 15, v0.t
36 # CHECK-INST: vor.vi v8, v4, 15, v0.t
37 # CHECK-ENCODING: [0x57,0xb4,0x47,0x28]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 b4 47 28 <unknown>
40
41 vor.vi v8, v4, 15
42 # CHECK-INST: vor.vi v8, v4, 15
43 # CHECK-ENCODING: [0x57,0xb4,0x47,0x2a]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 b4 47 2a <unknown>
```

llvm/test/MC/RISCV/rvv/others.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vmerge.vvm v8, v4, v20, v0
12 # CHECK-INST: vmerge.vvm v8, v4, v20, v0
13 # CHECK-ENCODING: [0x57,0x04,0x4a,0x5c]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 04 4a 5c <unknown>
16
17 vmerge.vxm v8, v4, a0, v0
18 # CHECK-INST: vmerge.vxm v8, v4, a0, v0
19 # CHECK-ENCODING: [0x57,0x44,0x45,0x5c]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 44 45 5c <unknown>
22
23 vmerge.vim v8, v4, 15, v0
24 # CHECK-INST: vmerge.vim v8, v4, 15, v0
25 # CHECK-ENCODING: [0x57,0xb4,0x47,0x5c]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
   nknown>

   4, a0, v0.t
31 # CHECK-ENCODING: [0x57,0x44,0x45,0x38]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 44 45 38 <unknown>
34
35 vslideup.vx v8, v4, a0
36 # CHECK-INST: vslideup.vx v8, v4, a0
37 # CHECK-ENCODING: [0x57,0x44,0x45,0x3a]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 44 45 3a <unknown>
40
41 vslideup.vi v8, v4, 31, v0.t
42 # CHECK-INST: vslideup.vi v8, v4, 31, v0.t
43 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x38]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 b4 4f 38 <unknown>
46
47 vslideup.vi v8, v4, 31
48 # CHECK-INST: vslideup.vi v8, v4, 31
49 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x3a]
50 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 57 b4 4f 3a <unknown>
52
53 vslidedown.vx v8, v4, a0, v0.t
54 # CHECK-INST: vslidedown.vx v8, v4, a0, v0.t
55 # CHECK-ENCODING: [0x57,0x44,0x45,0x3c]
56 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 57 44 45 3c <unknown>
58
59 vslidedown.vx v8, v4, a0
60 # CHECK-INST: vslidedown.vx v8, v4, a0
61 # CHECK-ENCODING: [0x57,0x44,0x45,0x3e]
62 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63 # CHECK-UNKNOWN: 57 44 45 3e <unknown>
64
65 vslidedown.vi v8, v4, 31, v0.t
66 # CHECK-INST: vslidedown.vi v8, v4, 31, v0.t
67 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x3c]
68 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69 # CHECK-UNKNOWN: 57 b4 4f 3c <unknown>
70
71 vslidedown.vi v8, v4, 31
72 # CHECK-INST: vslidedown.vi v8, v4, 31
73 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x3e]
74 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75 # CHECK-UNKNOWN: 57 b4 4f 3e <unknown>
76
77 vslide1up.vx v8, v4, a0, v0.t
78 # CHECK-INST: vslide1up.vx v8, v4, a0, v0.t
79 # CHECK-ENCODING: [0x57,0x64,0x45,0x38]
80 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81 # CHECK-UNKNOWN: 57 64 45 38 <unknown>
82
83 vslide1up.vx v8, v4, a0
84 # CHECK-INST: vslide1up.vx v8, v4, a0
85 # CHECK-ENCODING: [0x57,0x64,0x45,0x3a]
86 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87 # CHECK-UNKNOWN: 57 64 45 3a <unknown>
```



```
, v4, a0, v0.t
45,0x3c]
92 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
# CHECK-UNKNOWN: 57 64 45 3c <unknown>

vslide1down.vx v8, v4, a0
# CHECK-INST: vslide1down.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x64,0x45,0x3e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 64 45 3e <unknown>

vrgather.vv v8, v4, v20, v0.t
# CHECK-INST: vrgather.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x30]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 30 <unknown>

vrgather.vv v8, v4, v20
# CHECK-INST: vrgather.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x04,0x4a,0x32]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 32 <unknown>

vrgather.vx v8, v4, a0, v0.t
# CHECK-INST: vrgather.vx v8, v4, a0, v0.t
# CHECK-ENCODING: [0x57,0x44,0x45,0x30]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 30 <unknown>

vrgather.vx v8, v4, a0
# CHECK-INST: vrgather.vx v8, v4, a0
# CHECK-ENCODING: [0x57,0x44,0x45,0x32]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 44 45 32 <unknown>

vrgather.vi v8, v4, 31, v0.t
# CHECK-INST: vrgather.vi v8, v4, 31, v0.t
# CHECK-ENCODING: [0x57,0xb4,0x4f,0x30]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 b4 4f 30 <unknown>

vrgather.vi v8, v4, 31
# CHECK-INST: vrgather.vi v8, v4, 31
# CHECK-ENCODING: [0x57,0xb4,0x4f,0x32]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 b4 4f 32 <unknown>

vcompress.vm v8, v4, v20
# CHECK-INST: vcompress.vm v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x5e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 5e <unknown>
```

llvm/test/MC/RISCV/rvv/reduction.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
                                              check-prefixes=CHECK-ENCODING,CHECK-INST
                                              v64 -show-encoding < %s 2>&1 \
                                              check-prefix=CHECK-ERROR
                                              -filetype=obj --mattr=+experimental-v < %s \
# RUN:          | llvm-objdump -d --mattr=+experimental-v - \
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
# RUN:        | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vredsum.vs v8, v4, v20, v0.t
# CHECK-INST: vredsum.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x00]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 00 <unknown>

vredsum.vs v8, v4, v20
# CHECK-INST: vredsum.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x02]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 02 <unknown>

vredmaxu.vs v8, v4, v20, v0.t
# CHECK-INST: vredmaxu.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x18]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 18 <unknown>

vredmaxu.vs v8, v4, v20
# CHECK-INST: vredmaxu.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x1a]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 1a <unknown>

vredmax.vs v8, v4, v20, v0.t
# CHECK-INST: vredmax.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x1c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 1c <unknown>

vredmax.vs v8, v4, v20
# CHECK-INST: vredmax.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x1e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 1e <unknown>

vredminu.vs v8, v4, v20, v0.t
# CHECK-INST: vredminu.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x10]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 10 <unknown>

vredminu.vs v8, v4, v20
# CHECK-INST: vredminu.vs v8, v4, v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x12]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 24 4a 12 <unknown>

vredmin.vs v8, v4, v20, v0.t
# CHECK-INST: vredmin.vs v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x24,0x4a,0x14]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                  nknown>

                                                  , v20
# CHECK-ENCODING: [0x57,0x24,0x4a,0x16]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 24 4a 16 <unknown>
70
71  vredand.vs v8, v4, v20, v0.t
72  # CHECK-INST: vredand.vs v8, v4, v20, v0.t
73  # CHECK-ENCODING: [0x57,0x24,0x4a,0x04]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 24 4a 04 <unknown>
76
77  vredand.vs v8, v4, v20
78  # CHECK-INST: vredand.vs v8, v4, v20
79  # CHECK-ENCODING: [0x57,0x24,0x4a,0x06]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 24 4a 06 <unknown>
82
83  vredor.vs v8, v4, v20, v0.t
84  # CHECK-INST: vredor.vs v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x24,0x4a,0x08]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 24 4a 08 <unknown>
88
89  vredor.vs v8, v4, v20
90  # CHECK-INST: vredor.vs v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x24,0x4a,0x0a]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 24 4a 0a <unknown>
94
95  vredxor.vs v8, v4, v20, v0.t
96  # CHECK-INST: vredxor.vs v8, v4, v20, v0.t
97  # CHECK-ENCODING: [0x57,0x24,0x4a,0x0c]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 24 4a 0c <unknown>
100
101 vredxor.vs v8, v4, v20
102 # CHECK-INST: vredxor.vs v8, v4, v20
103 # CHECK-ENCODING: [0x57,0x24,0x4a,0x0e]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 24 4a 0e <unknown>
106
107 vwredsumu.vs v8, v4, v20, v0.t
108 # CHECK-INST: vwredsumu.vs v8, v4, v20, v0.t
109 # CHECK-ENCODING: [0x57,0x04,0x4a,0xc0]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 04 4a c0 <unknown>
112
113 vwredsumu.vs v8, v4, v20
114 # CHECK-INST: vwredsumu.vs v8, v4, v20
115 # CHECK-ENCODING: [0x57,0x04,0x4a,0xc2]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 04 4a c2 <unknown>
118
119 vwredsum.vs v8, v4, v20, v0.t
120 # CHECK-INST: vwredsum.vs v8, v4, v20, v0.t
121 # CHECK-ENCODING: [0x57,0x04,0x4a,0xc4]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 04 4a c4 <unknown>
```



```
                                  4, v20
                                  4a,0xc6]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
129 # CHECK-UNKNOWN: 57 04 4a c6 <unknown>
```

llvm/test/MC/RISCV/rvv/shift.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vsll.vv v8, v4, v20, v0.t
12 # CHECK-INST: vsll.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x04,0x4a,0x94]
14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
15 # CHECK-UNKNOWN: 57 04 4a 94 <unknown>
16
17 vsll.vv v8, v4, v20
18 # CHECK-INST: vsll.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x04,0x4a,0x96]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 57 04 4a 96 <unknown>
22
23 vsll.vx v8, v4, a0, v0.t
24 # CHECK-INST: vsll.vx v8, v4, a0, v0.t
25 # CHECK-ENCODING: [0x57,0x44,0x45,0x94]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 57 44 45 94 <unknown>
28
29 vsll.vx v8, v4, a0
30 # CHECK-INST: vsll.vx v8, v4, a0
31 # CHECK-ENCODING: [0x57,0x44,0x45,0x96]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 44 45 96 <unknown>
34
35 vsll.vi v8, v4, 31, v0.t
36 # CHECK-INST: vsll.vi v8, v4, 31, v0.t
37 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x94]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 b4 4f 94 <unknown>
40
41 vsll.vi v8, v4, 31
42 # CHECK-INST: vsll.vi v8, v4, 31
43 # CHECK-ENCODING: [0x57,0xb4,0x4f,0x96]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 b4 4f 96 <unknown>
46
47 vsrl.vv v8, v4, v20, v0.t
48 # CHECK-INST: vsrl.vv v8, v4, v20, v0.t
49 # CHECK-ENCODING: [0x57,0x04,0x4a,0xa0]
   ires the following: 'V' (Vector Instructions)
   nknown>
54 # CHECK-INST: vsrl.vv v8, v4, v20
```

```
55  # CHECK-ENCODING: [0x57,0x04,0x4a,0xa2]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 04 4a a2 <unknown>
58
59  vsrl.vx v8, v4, a0, v0.t
60  # CHECK-INST: vsrl.vx v8, v4, a0, v0.t
61  # CHECK-ENCODING: [0x57,0x44,0x45,0xa0]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 44 45 a0 <unknown>
64
65  vsrl.vx v8, v4, a0
66  # CHECK-INST: vsrl.vx v8, v4, a0
67  # CHECK-ENCODING: [0x57,0x44,0x45,0xa2]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 44 45 a2 <unknown>
70
71  vsrl.vi v8, v4, 31, v0.t
72  # CHECK-INST: vsrl.vi v8, v4, 31, v0.t
73  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xa0]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 b4 4f a0 <unknown>
76
77  vsrl.vi v8, v4, 31
78  # CHECK-INST: vsrl.vi v8, v4, 31
79  # CHECK-ENCODING: [0x57,0xb4,0x4f,0xa2]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 b4 4f a2 <unknown>
82
83  vsra.vv v8, v4, v20, v0.t
84  # CHECK-INST: vsra.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x04,0x4a,0xa4]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 04 4a a4 <unknown>
88
89  vsra.vv v8, v4, v20
90  # CHECK-INST: vsra.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x04,0x4a,0xa6]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 04 4a a6 <unknown>
94
95  vsra.vx v8, v4, a0, v0.t
96  # CHECK-INST: vsra.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x44,0x45,0xa4]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 44 45 a4 <unknown>
100
101 vsra.vx v8, v4, a0
102 # CHECK-INST: vsra.vx v8, v4, a0
103 # CHECK-ENCODING: [0x57,0x44,0x45,0xa6]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 57 44 45 a6 <unknown>
106
107 vsra.vi v8, v4, 31, v0.t
108 # CHECK-INST: vsra.vi v8, v4, 31, v0.t
109 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xa4]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    1
115 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xa6]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 b4 4f a6 <unknown>
118
119 vnsrl.wv v8, v4, v20, v0.t
120 # CHECK-INST: vnsrl.wv v8, v4, v20, v0.t
121 # CHECK-ENCODING: [0x57,0x04,0x4a,0xb0]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 04 4a b0 <unknown>
124
125 vnsrl.wv v8, v4, v20
126 # CHECK-INST: vnsrl.wv v8, v4, v20
127 # CHECK-ENCODING: [0x57,0x04,0x4a,0xb2]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 04 4a b2 <unknown>
130
131 vnsrl.wx v8, v4, a0, v0.t
132 # CHECK-INST: vnsrl.wx v8, v4, a0, v0.t
133 # CHECK-ENCODING: [0x57,0x44,0x45,0xb0]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 44 45 b0 <unknown>
136
137 vnsrl.wx v8, v4, a0
138 # CHECK-INST: vnsrl.wx v8, v4, a0
139 # CHECK-ENCODING: [0x57,0x44,0x45,0xb2]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 57 44 45 b2 <unknown>
142
143 vnsrl.wi v8, v4, 31, v0.t
144 # CHECK-INST: vnsrl.wi v8, v4, 31, v0.t
145 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xb0]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 b4 4f b0 <unknown>
148
149 vnsrl.wi v8, v4, 31
150 # CHECK-INST: vnsrl.wi v8, v4, 31
151 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xb2]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 57 b4 4f b2 <unknown>
154
155 vnsra.wv v8, v4, v20, v0.t
156 # CHECK-INST: vnsra.wv v8, v4, v20, v0.t
157 # CHECK-ENCODING: [0x57,0x04,0x4a,0xb4]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 04 4a b4 <unknown>
160
161 vnsra.wv v8, v4, v20
162 # CHECK-INST: vnsra.wv v8, v4, v20
163 # CHECK-ENCODING: [0x57,0x04,0x4a,0xb6]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 04 4a b6 <unknown>
166
167 vnsra.wx v8, v4, a0, v0.t
168 # CHECK-INST: vnsra.wx v8, v4, a0, v0.t
169 # CHECK-ENCODING: [0x57,0x44,0x45,0xb4]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
171 # CHECK-UNKNOWN: 57 44 45 b4 <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
a0
45,0xb6]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
177 # CHECK-UNKNOWN: 57 44 45 b6 <unknown>
178
179 vnsra.wi v8, v4, 31, v0.t
180 # CHECK-INST: vnsra.wi v8, v4, 31, v0.t
181 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xb4]
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 b4 4f b4 <unknown>
184
185 vnsra.wi v8, v4, 31
186 # CHECK-INST: vnsra.wi v8, v4, 31
187 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xb6]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 b4 4f b6 <unknown>
190
191 vssrl.vv v8, v4, v20, v0.t
192 # CHECK-INST: vssrl.vv v8, v4, v20, v0.t
193 # CHECK-ENCODING: [0x57,0x04,0x4a,0xa8]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 57 04 4a a8 <unknown>
196
197 vssrl.vv v8, v4, v20
198 # CHECK-INST: vssrl.vv v8, v4, v20
199 # CHECK-ENCODING: [0x57,0x04,0x4a,0xaa]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 57 04 4a aa <unknown>
202
203 vssrl.vx v8, v4, a0, v0.t
204 # CHECK-INST: vssrl.vx v8, v4, a0, v0.t
205 # CHECK-ENCODING: [0x57,0x44,0x45,0xa8]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 57 44 45 a8 <unknown>
208
209 vssrl.vx v8, v4, a0
210 # CHECK-INST: vssrl.vx v8, v4, a0
211 # CHECK-ENCODING: [0x57,0x44,0x45,0xaa]
212 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
213 # CHECK-UNKNOWN: 57 44 45 aa <unknown>
214
215 vssrl.vi v8, v4, 31, v0.t
216 # CHECK-INST: vssrl.vi v8, v4, 31, v0.t
217 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xa8]
218 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
219 # CHECK-UNKNOWN: 57 b4 4f a8 <unknown>
220
221 vssrl.vi v8, v4, 31
222 # CHECK-INST: vssrl.vi v8, v4, 31
223 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xaa]
224 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
225 # CHECK-UNKNOWN: 57 b4 4f aa <unknown>
226
227 vssra.vv v8, v4, v20, v0.t
228 # CHECK-INST: vssra.vv v8, v4, v20, v0.t
229 # CHECK-ENCODING: [0x57,0x04,0x4a,0xac]
230 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
231 # CHECK-UNKNOWN: 57 04 4a ac <unknown>
232

                                    v20
                                    4a,0xae]
                                    ires the following: 'V' (Vector Instructions)
237 # CHECK-UNKNOWN: 57 04 4a ae <unknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
238
239 vssra.vx v8, v4, a0, v0.t
240 # CHECK-INST: vssra.vx v8, v4, a0, v0.t
241 # CHECK-ENCODING: [0x57,0x44,0x45,0xac]
242 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
243 # CHECK-UNKNOWN: 57 44 45 ac <unknown>
244
245 vssra.vx v8, v4, a0
246 # CHECK-INST: vssra.vx v8, v4, a0
247 # CHECK-ENCODING: [0x57,0x44,0x45,0xae]
248 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
249 # CHECK-UNKNOWN: 57 44 45 ae <unknown>
250
251 vssra.vi v8, v4, 31, v0.t
252 # CHECK-INST: vssra.vi v8, v4, 31, v0.t
253 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xac]
254 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
255 # CHECK-UNKNOWN: 57 b4 4f ac <unknown>
256
257 vssra.vi v8, v4, 31
258 # CHECK-INST: vssra.vi v8, v4, 31
259 # CHECK-ENCODING: [0x57,0xb4,0x4f,0xae]
260 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
261 # CHECK-UNKNOWN: 57 b4 4f ae <unknown>
```

llvm/test/MC/RISCV/rvv/sign-injection.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v --mattr=+f < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v --mattr=+f - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v --mattr=+f < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vfsgnj.vv v8, v4, v20, v0.t
12 # CHECK-INST: vfsgnj.vv v8, v4, v20, v0.t
13 # CHECK-ENCODING: [0x57,0x14,0x4a,0x20]
14 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
15 # CHECK-UNKNOWN: 57 14 4a 20 <unknown>
16
17 vfsgnj.vv v8, v4, v20
18 # CHECK-INST: vfsgnj.vv v8, v4, v20
19 # CHECK-ENCODING: [0x57,0x14,0x4a,0x22]
20 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
21 # CHECK-UNKNOWN: 57 14 4a 22 <unknown>
22
23 vfsgnj.vf v8, v4, fa0, v0.t
24 # CHECK-INST: vfsgnj.vf v8, v4, fa0, v0.t
25 # CHECK-ENCODING: [0x57,0x54,0x45,0x20]
   ires the following: 'F' (Single-Precision Floating-Point), 'V'
   nknown>
29 vfsgnj.vf v8, v4, fa0
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
30 # CHECK-INST: vfsgnj.vf v8, v4, fa0
31 # CHECK-ENCODING: [0x57,0x54,0x45,0x22]
32 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
33 # CHECK-UNKNOWN: 57 54 45 22 <unknown>
34
35 vfsgnjn.vv v8, v4, v20, v0.t
36 # CHECK-INST: vfsgnjn.vv v8, v4, v20, v0.t
37 # CHECK-ENCODING: [0x57,0x14,0x4a,0x24]
38 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
39 # CHECK-UNKNOWN: 57 14 4a 24 <unknown>
40
41 vfsgnjn.vv v8, v4, v20
42 # CHECK-INST: vfsgnjn.vv v8, v4, v20
43 # CHECK-ENCODING: [0x57,0x14,0x4a,0x26]
44 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
45 # CHECK-UNKNOWN: 57 14 4a 26 <unknown>
46
47 vfsgnjn.vf v8, v4, fa0, v0.t
48 # CHECK-INST: vfsgnjn.vf v8, v4, fa0, v0.t
49 # CHECK-ENCODING: [0x57,0x54,0x45,0x24]
50 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
51 # CHECK-UNKNOWN: 57 54 45 24 <unknown>
52
53 vfsgnjn.vf v8, v4, fa0
54 # CHECK-INST: vfsgnjn.vf v8, v4, fa0
55 # CHECK-ENCODING: [0x57,0x54,0x45,0x26]
56 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
57 # CHECK-UNKNOWN: 57 54 45 26 <unknown>
58
59 vfsgnjx.vv v8, v4, v20, v0.t
60 # CHECK-INST: vfsgnjx.vv v8, v4, v20, v0.t
61 # CHECK-ENCODING: [0x57,0x14,0x4a,0x28]
62 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
63 # CHECK-UNKNOWN: 57 14 4a 28 <unknown>
64
65 vfsgnjx.vv v8, v4, v20
66 # CHECK-INST: vfsgnjx.vv v8, v4, v20
67 # CHECK-ENCODING: [0x57,0x14,0x4a,0x2a]
68 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
69 # CHECK-UNKNOWN: 57 14 4a 2a <unknown>
70
71 vfsgnjx.vf v8, v4, fa0, v0.t
72 # CHECK-INST: vfsgnjx.vf v8, v4, fa0, v0.t
73 # CHECK-ENCODING: [0x57,0x54,0x45,0x28]
74 # CHECK-ERROR: instruction requires the following: 'F' (Single-Precision Floating-Point), 'V'
   (Vector Instructions)
75 # CHECK-UNKNOWN: 57 54 45 28 <unknown>
76
77 vfsgnjx.vf v8, v4, fa0
78 # CHECK-INST: vfsgnjx.vf v8, v4, fa0
                                45,0x2a]
                                ires the following: 'F' (Single-Precision Floating-Point), 'V'

                                nknown>
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

llvm/test/MC/RISCV/rvv/snippet.s

View Options

This file was added.

```
1  ## A snippet from https://github.com/riscv/riscv-v-spec.
2
3  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
4  # RUN: | llvm-objdump -d --mattr=+experimental-v - \
5  # RUN: | FileCheck %s --check-prefix=CHECK-INST
6
7  loop:
8      vsetvli a3, a0, e16,m4  # vtype = 16-bit integer vectors
9  # CHECK-INST: d7 76 65 00    vsetvli a3, a0, e16,m4
10     vlh.v v4, (a1)          # Get 16b vector
11 # CHECK-INST: 07 d2 05 12    vlh.v   v4, (a1)
12     slli t1, a3, 1          # Multiply length by two bytes/element
13 # CHECK-INST: 13 93 16 00    slli    t1, a3, 1
14     add a1, a1, t1          # Bump pointer
15 # CHECK-INST: b3 85 65 00    add     a1, a1, t1
16     vwmul.vx v8, v4, x10    # 32b in <v8--v15>
17 # CHECK-INST: 57 64 45 ee    vwmul.vx        v8, v4, a0
18
19     vsetvli x0, a0, e32,m8  # Operate on 32b values
20 # CHECK-INST: 57 70 b5 00    vsetvli zero, a0, e32,m8
21     vsrl.vi v8, v8, 3
22 # CHECK-INST: 57 b4 81 a2    vsrl.vi v8, v8, 3
23     vsw.v v8, (a2)          # Store vector of 32b
24 # CHECK-INST: 27 64 06 02    vsw.v   v8, (a2)
25     slli t1, a3, 2          # Multiply length by four bytes/element
26 # CHECK-INST: 13 93 26 00    slli    t1, a3, 2
27     add a2, a2, t1          # Bump pointer
28 # CHECK-INST: 33 06 66 00    add     a2, a2, t1
29     sub a0, a0, a3          # Decrement count
30 # CHECK-INST: 33 05 d5 40    sub     a0, a0, a3
31     bnez a0, loop           # Any more?
32 # CHECK-INST: e3 1a 05 fc    bnez    a0, -44
```

llvm/test/MC/RISCV/rvv/store.s

View Options

This file was added.

```
1  # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
2  # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
3  # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
4  # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
5  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
6  # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
7  # RUN:         | FileCheck %s --check-prefix=CHECK-INST
8  # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
9  # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
10
11 vsb.v v24, (a0), v0.t
12 # CHECK-INST: vsb.v v24, (a0), v0.t
                                    05,0x00]
                                    ires the following: 'V' (Vector Instructions)
                                    nknown>
17 vsb.v v24, (a0)
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
18 # CHECK-INST: vsb.v v24, (a0)
19 # CHECK-ENCODING: [0x27,0x0c,0x05,0x02]
20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
21 # CHECK-UNKNOWN: 27 0c 05 02 <unknown>
22
23 vsh.v v24, (a0), v0.t
24 # CHECK-INST: vsh.v v24, (a0), v0.t
25 # CHECK-ENCODING: [0x27,0x5c,0x05,0x00]
26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
27 # CHECK-UNKNOWN: 27 5c 05 00 <unknown>
28
29 vsh.v v24, (a0)
30 # CHECK-INST: vsh.v v24, (a0)
31 # CHECK-ENCODING: [0x27,0x5c,0x05,0x02]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 27 5c 05 02 <unknown>
34
35 vsw.v v24, (a0), v0.t
36 # CHECK-INST: vsw.v v24, (a0), v0.t
37 # CHECK-ENCODING: [0x27,0x6c,0x05,0x00]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 27 6c 05 00 <unknown>
40
41 vsw.v v24, (a0)
42 # CHECK-INST: vsw.v v24, (a0)
43 # CHECK-ENCODING: [0x27,0x6c,0x05,0x02]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 27 6c 05 02 <unknown>
46
47 vse.v v24, (a0), v0.t
48 # CHECK-INST: vse.v v24, (a0), v0.t
49 # CHECK-ENCODING: [0x27,0x7c,0x05,0x00]
50 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51 # CHECK-UNKNOWN: 27 7c 05 00 <unknown>
52
53 vse.v v24, (a0)
54 # CHECK-INST: vse.v v24, (a0)
55 # CHECK-ENCODING: [0x27,0x7c,0x05,0x02]
56 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57 # CHECK-UNKNOWN: 27 7c 05 02 <unknown>
58
59 vssb.v v24, (a0), a1, v0.t
60 # CHECK-INST: vssb.v v24, (a0), a1, v0.t
61 # CHECK-ENCODING: [0x27,0x0c,0xb5,0x08]
62 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63 # CHECK-UNKNOWN: 27 0c b5 08 <unknown>
64
65 vssb.v v24, (a0), a1
66 # CHECK-INST: vssb.v v24, (a0), a1
67 # CHECK-ENCODING: [0x27,0x0c,0xb5,0x0a]
68 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69 # CHECK-UNKNOWN: 27 0c b5 0a <unknown>
70
71 vssh.v v24, (a0), a1, v0.t
72 # CHECK-INST: vssh.v v24, (a0), a1, v0.t
73 # CHECK-ENCODING: [0x27,0x5c,0xb5,0x08]
   ires the following: 'V' (Vector Instructions)
   nknown>

78 # CHECK-INST: vssh.v v24, (a0), a1
```

```
79  # CHECK-ENCODING: [0x27,0x5c,0xb5,0x0a]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 27 5c b5 0a <unknown>
82
83  vssw.v v24, (a0), a1, v0.t
84  # CHECK-INST: vssw.v v24, (a0), a1, v0.t
85  # CHECK-ENCODING: [0x27,0x6c,0xb5,0x08]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 27 6c b5 08 <unknown>
88
89  vssw.v v24, (a0), a1
90  # CHECK-INST: vssw.v v24, (a0), a1
91  # CHECK-ENCODING: [0x27,0x6c,0xb5,0x0a]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 27 6c b5 0a <unknown>
94
95  vsse.v v24, (a0), a1, v0.t
96  # CHECK-INST: vsse.v v24, (a0), a1, v0.t
97  # CHECK-ENCODING: [0x27,0x7c,0xb5,0x08]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 27 7c b5 08 <unknown>
100
101 vsse.v v24, (a0), a1
102 # CHECK-INST: vsse.v v24, (a0), a1
103 # CHECK-ENCODING: [0x27,0x7c,0xb5,0x0a]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
105 # CHECK-UNKNOWN: 27 7c b5 0a <unknown>
106
107 vsxb.v v24, (a0), v4, v0.t
108 # CHECK-INST: vsxb.v v24, (a0), v4, v0.t
109 # CHECK-ENCODING: [0x27,0x0c,0x45,0x0c]
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 27 0c 45 0c <unknown>
112
113 vsxb.v v24, (a0), v4
114 # CHECK-INST: vsxb.v v24, (a0), v4
115 # CHECK-ENCODING: [0x27,0x0c,0x45,0x0e]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 27 0c 45 0e <unknown>
118
119 vsxh.v v24, (a0), v4, v0.t
120 # CHECK-INST: vsxh.v v24, (a0), v4, v0.t
121 # CHECK-ENCODING: [0x27,0x5c,0x45,0x0c]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 27 5c 45 0c <unknown>
124
125 vsxh.v v24, (a0), v4
126 # CHECK-INST: vsxh.v v24, (a0), v4
127 # CHECK-ENCODING: [0x27,0x5c,0x45,0x0e]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 27 5c 45 0e <unknown>
130
131 vsxw.v v24, (a0), v4, v0.t
132 # CHECK-INST: vsxw.v v24, (a0), v4, v0.t
133 # CHECK-ENCODING: [0x27,0x6c,0x45,0x0c]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    v4
139 # CHECK-ENCODING: [0x27,0x6c,0x45,0x0e]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 6c 45 0e <unknown>

vsxe.v v24, (a0), v4, v0.t
# CHECK-INST: vsxe.v v24, (a0), v4, v0.t
# CHECK-ENCODING: [0x27,0x7c,0x45,0x0c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 7c 45 0c <unknown>

vsxe.v v24, (a0), v4
# CHECK-INST: vsxe.v v24, (a0), v4
# CHECK-ENCODING: [0x27,0x7c,0x45,0x0e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 7c 45 0e <unknown>

vsuxb.v v24, (a0), v4, v0.t
# CHECK-INST: vsuxb.v v24, (a0), v4, v0.t
# CHECK-ENCODING: [0x27,0x0c,0x45,0x1c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 0c 45 1c <unknown>

vsuxb.v v24, (a0), v4
# CHECK-INST: vsuxb.v v24, (a0), v4
# CHECK-ENCODING: [0x27,0x0c,0x45,0x1e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 0c 45 1e <unknown>

vsuxh.v v24, (a0), v4, v0.t
# CHECK-INST: vsuxh.v v24, (a0), v4, v0.t
# CHECK-ENCODING: [0x27,0x5c,0x45,0x1c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 5c 45 1c <unknown>

vsuxh.v v24, (a0), v4
# CHECK-INST: vsuxh.v v24, (a0), v4
# CHECK-ENCODING: [0x27,0x5c,0x45,0x1e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 5c 45 1e <unknown>

vsuxw.v v24, (a0), v4, v0.t
# CHECK-INST: vsuxw.v v24, (a0), v4, v0.t
# CHECK-ENCODING: [0x27,0x6c,0x45,0x1c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 6c 45 1c <unknown>

vsuxw.v v24, (a0), v4
# CHECK-INST: vsuxw.v v24, (a0), v4
# CHECK-ENCODING: [0x27,0x6c,0x45,0x1e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 6c 45 1e <unknown>

vsuxe.v v24, (a0), v4, v0.t
# CHECK-INST: vsuxe.v v24, (a0), v4, v0.t
# CHECK-ENCODING: [0x27,0x7c,0x45,0x1c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 27 7c 45 1c <unknown>

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

, v4
45,0x1e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
201 # CHECK-UNKNOWN: 27 7c 45 1e <unknown>
202
203 vs1r.v v24, (a0)
204 # CHECK-INST: vs1r.v v24, (a0)
205 # CHECK-ENCODING: [0x27,0x7c,0x85,0x02]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 27 7c 85 02 <unknown>
```

llvm/test/MC/RISCV/rvv/sub.s

View Options

This file was added.

```
  1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
  2 # RUN:         | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
  3 # RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
  4 # RUN:         | FileCheck %s --check-prefix=CHECK-ERROR
  5 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
  6 # RUN:         | llvm-objdump -d --mattr=+experimental-v - \
  7 # RUN:         | FileCheck %s --check-prefix=CHECK-INST
  8 # RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
  9 # RUN:         | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN
 10
 11 vsub.vv v8, v4, v20, v0.t
 12 # CHECK-INST: vsub.vv v8, v4, v20, v0.t
 13 # CHECK-ENCODING: [0x57,0x04,0x4a,0x08]
 14 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
 15 # CHECK-UNKNOWN: 57 04 4a 08 <unknown>
 16
 17 vsub.vv v8, v4, v20
 18 # CHECK-INST: vsub.vv v8, v4, v20
 19 # CHECK-ENCODING: [0x57,0x04,0x4a,0x0a]
 20 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
 21 # CHECK-UNKNOWN: 57 04 4a 0a <unknown>
 22
 23 vsub.vx v8, v4, a0, v0.t
 24 # CHECK-INST: vsub.vx v8, v4, a0, v0.t
 25 # CHECK-ENCODING: [0x57,0x44,0x45,0x08]
 26 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
 27 # CHECK-UNKNOWN: 57 44 45 08 <unknown>
 28
 29 vsub.vx v8, v4, a0
 30 # CHECK-INST: vsub.vx v8, v4, a0
 31 # CHECK-ENCODING: [0x57,0x44,0x45,0x0a]
 32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
 33 # CHECK-UNKNOWN: 57 44 45 0a <unknown>
 34
 35 vrsub.vx v8, v4, a0, v0.t
 36 # CHECK-INST: vrsub.vx v8, v4, a0, v0.t
 37 # CHECK-ENCODING: [0x57,0x44,0x45,0x0c]
 38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
 39 # CHECK-UNKNOWN: 57 44 45 0c <unknown>
 40
 41 vrsub.vx v8, v4, a0
 42 # CHECK-INST: vrsub.vx v8, v4, a0
 43 # CHECK-ENCODING: [0x57,0x44,0x45,0x0e]
    ires the following: 'V' (Vector Instructions)
    nknown>
 48 # CHECK-INST: vrsub.vi v8, v4, 15, v0.t
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
49  # CHECK-ENCODING: [0x57,0xb4,0x47,0x0c]
50  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
51  # CHECK-UNKNOWN: 57 b4 47 0c <unknown>
52
53  vrsub.vi v8, v4, 15
54  # CHECK-INST: vrsub.vi v8, v4, 15
55  # CHECK-ENCODING: [0x57,0xb4,0x47,0x0e]
56  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
57  # CHECK-UNKNOWN: 57 b4 47 0e <unknown>
58
59  vwsubu.vv v8, v4, v20, v0.t
60  # CHECK-INST: vwsubu.vv v8, v4, v20, v0.t
61  # CHECK-ENCODING: [0x57,0x24,0x4a,0xc8]
62  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
63  # CHECK-UNKNOWN: 57 24 4a c8 <unknown>
64
65  vwsubu.vv v8, v4, v20
66  # CHECK-INST: vwsubu.vv v8, v4, v20
67  # CHECK-ENCODING: [0x57,0x24,0x4a,0xca]
68  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
69  # CHECK-UNKNOWN: 57 24 4a ca <unknown>
70
71  vwsubu.vx v8, v4, a0, v0.t
72  # CHECK-INST: vwsubu.vx v8, v4, a0, v0.t
73  # CHECK-ENCODING: [0x57,0x64,0x45,0xc8]
74  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
75  # CHECK-UNKNOWN: 57 64 45 c8 <unknown>
76
77  vwsubu.vx v8, v4, a0
78  # CHECK-INST: vwsubu.vx v8, v4, a0
79  # CHECK-ENCODING: [0x57,0x64,0x45,0xca]
80  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
81  # CHECK-UNKNOWN: 57 64 45 ca <unknown>
82
83  vwsub.vv v8, v4, v20, v0.t
84  # CHECK-INST: vwsub.vv v8, v4, v20, v0.t
85  # CHECK-ENCODING: [0x57,0x24,0x4a,0xcc]
86  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
87  # CHECK-UNKNOWN: 57 24 4a cc <unknown>
88
89  vwsub.vv v8, v4, v20
90  # CHECK-INST: vwsub.vv v8, v4, v20
91  # CHECK-ENCODING: [0x57,0x24,0x4a,0xce]
92  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
93  # CHECK-UNKNOWN: 57 24 4a ce <unknown>
94
95  vwsub.vx v8, v4, a0, v0.t
96  # CHECK-INST: vwsub.vx v8, v4, a0, v0.t
97  # CHECK-ENCODING: [0x57,0x64,0x45,0xcc]
98  # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
99  # CHECK-UNKNOWN: 57 64 45 cc <unknown>
100
101 vwsub.vx v8, v4, a0
102 # CHECK-INST: vwsub.vx v8, v4, a0
103 # CHECK-ENCODING: [0x57,0x64,0x45,0xce]
104 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
                                                    nknown>

                                                    v20, v0.t
109 # CHECK-ENCODING: [0x57,0x24,0x4a,0xd8]
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
110 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
111 # CHECK-UNKNOWN: 57 24 4a d8 <unknown>
112
113 vwsubu.wv v8, v4, v20
114 # CHECK-INST: vwsubu.wv v8, v4, v20
115 # CHECK-ENCODING: [0x57,0x24,0x4a,0xda]
116 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
117 # CHECK-UNKNOWN: 57 24 4a da <unknown>
118
119 vwsubu.wx v8, v4, a0, v0.t
120 # CHECK-INST: vwsubu.wx v8, v4, a0, v0.t
121 # CHECK-ENCODING: [0x57,0x64,0x45,0xd8]
122 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
123 # CHECK-UNKNOWN: 57 64 45 d8 <unknown>
124
125 vwsubu.wx v8, v4, a0
126 # CHECK-INST: vwsubu.wx v8, v4, a0
127 # CHECK-ENCODING: [0x57,0x64,0x45,0xda]
128 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
129 # CHECK-UNKNOWN: 57 64 45 da <unknown>
130
131 vwsub.wv v8, v4, v20, v0.t
132 # CHECK-INST: vwsub.wv v8, v4, v20, v0.t
133 # CHECK-ENCODING: [0x57,0x24,0x4a,0xdc]
134 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
135 # CHECK-UNKNOWN: 57 24 4a dc <unknown>
136
137 vwsub.wv v8, v4, v20
138 # CHECK-INST: vwsub.wv v8, v4, v20
139 # CHECK-ENCODING: [0x57,0x24,0x4a,0xde]
140 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
141 # CHECK-UNKNOWN: 57 24 4a de <unknown>
142
143 vwsub.wx v8, v4, a0, v0.t
144 # CHECK-INST: vwsub.wx v8, v4, a0, v0.t
145 # CHECK-ENCODING: [0x57,0x64,0x45,0xdc]
146 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
147 # CHECK-UNKNOWN: 57 64 45 dc <unknown>
148
149 vwsub.wx v8, v4, a0
150 # CHECK-INST: vwsub.wx v8, v4, a0
151 # CHECK-ENCODING: [0x57,0x64,0x45,0xde]
152 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
153 # CHECK-UNKNOWN: 57 64 45 de <unknown>
154
155 vsbc.vvm v8, v4, v20, v0
156 # CHECK-INST: vsbc.vvm v8, v4, v20, v0
157 # CHECK-ENCODING: [0x57,0x04,0x4a,0x48]
158 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
159 # CHECK-UNKNOWN: 57 04 4a 48 <unknown>
160
161 vsbc.vxm v8, v4, a0, v0
162 # CHECK-INST: vsbc.vxm v8, v4, a0, v0
163 # CHECK-ENCODING: [0x57,0x44,0x45,0x48]
164 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
165 # CHECK-UNKNOWN: 57 44 45 48 <unknown>
166
167
168                                  v20, v0
169                                  4a,0x4c]
170 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
171 # CHECK-UNKNOWN: 57 04 4a 4c <unknown>
172
173 vmsbc.vxm v8, v4, a0, v0
174 # CHECK-INST: vmsbc.vxm v8, v4, a0, v0
175 # CHECK-ENCODING: [0x57,0x44,0x45,0x4c]
176 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
177 # CHECK-UNKNOWN: 57 44 45 4c <unknown>
178
179 vmsbc.vv v8, v4, v20
180 # CHECK-INST: vmsbc.vv v8, v4, v20
181 # CHECK-ENCODING: [0x57,0x04,0x4a,0x4e]
182 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
183 # CHECK-UNKNOWN: 57 04 4a 4e <unknown>
184
185 vmsbc.vx v8, v4, a0
186 # CHECK-INST: vmsbc.vx v8, v4, a0
187 # CHECK-ENCODING: [0x57,0x44,0x45,0x4e]
188 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
189 # CHECK-UNKNOWN: 57 44 45 4e <unknown>
190
191 vssubu.vv v8, v4, v20, v0.t
192 # CHECK-INST: vssubu.vv v8, v4, v20, v0.t
193 # CHECK-ENCODING: [0x57,0x04,0x4a,0x88]
194 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
195 # CHECK-UNKNOWN: 57 04 4a 88 <unknown>
196
197 vssubu.vv v8, v4, v20
198 # CHECK-INST: vssubu.vv v8, v4, v20
199 # CHECK-ENCODING: [0x57,0x04,0x4a,0x8a]
200 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
201 # CHECK-UNKNOWN: 57 04 4a 8a <unknown>
202
203 vssubu.vx v8, v4, a0, v0.t
204 # CHECK-INST: vssubu.vx v8, v4, a0, v0.t
205 # CHECK-ENCODING: [0x57,0x44,0x45,0x88]
206 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
207 # CHECK-UNKNOWN: 57 44 45 88 <unknown>
208
209 vssubu.vx v8, v4, a0
210 # CHECK-INST: vssubu.vx v8, v4, a0
211 # CHECK-ENCODING: [0x57,0x44,0x45,0x8a]
212 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
213 # CHECK-UNKNOWN: 57 44 45 8a <unknown>
214
215 vssub.vv v8, v4, v20, v0.t
216 # CHECK-INST: vssub.vv v8, v4, v20, v0.t
217 # CHECK-ENCODING: [0x57,0x04,0x4a,0x8c]
218 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
219 # CHECK-UNKNOWN: 57 04 4a 8c <unknown>
220
221 vssub.vv v8, v4, v20
222 # CHECK-INST: vssub.vv v8, v4, v20
223 # CHECK-ENCODING: [0x57,0x04,0x4a,0x8e]
224 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
225 # CHECK-UNKNOWN: 57 04 4a 8e <unknown>
226
227 vssub.vx v8, v4, a0, v0.t
228 # CHECK-INST: vssub.vx v8, v4, a0, v0.t
229 # CHECK-ENCODING: [0x57,0x44,0x45,0x8c]
230 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
231 # CHECK-UNKNOWN: 57 44 45 8c <unknown>
```

```
232
233 vssub.vx v8, v4, a0
234 # CHECK-INST: vssub.vx v8, v4, a0
235 # CHECK-ENCODING: [0x57,0x44,0x45,0x8e]
236 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
237 # CHECK-UNKNOWN: 57 44 45 8e <unknown>
238
239 vasub.vv v8, v4, v20, v0.t
240 # CHECK-INST: vasub.vv v8, v4, v20, v0.t
241 # CHECK-ENCODING: [0x57,0x24,0x4a,0x2c]
242 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
243 # CHECK-UNKNOWN: 57 24 4a 2c <unknown>
244
245 vasub.vv v8, v4, v20
246 # CHECK-INST: vasub.vv v8, v4, v20
247 # CHECK-ENCODING: [0x57,0x24,0x4a,0x2e]
248 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
249 # CHECK-UNKNOWN: 57 24 4a 2e <unknown>
250
251 vasub.vx v8, v4, a0, v0.t
252 # CHECK-INST: vasub.vx v8, v4, a0, v0.t
253 # CHECK-ENCODING: [0x57,0x64,0x45,0x2c]
254 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
255 # CHECK-UNKNOWN: 57 64 45 2c <unknown>
256
257 vasub.vx v8, v4, a0
258 # CHECK-INST: vasub.vx v8, v4, a0
259 # CHECK-ENCODING: [0x57,0x64,0x45,0x2e]
260 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
261 # CHECK-UNKNOWN: 57 64 45 2e <unknown>
262
263 vasubu.vv v8, v4, v20, v0.t
264 # CHECK-INST: vasubu.vv v8, v4, v20, v0.t
265 # CHECK-ENCODING: [0x57,0x24,0x4a,0x28]
266 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
267 # CHECK-UNKNOWN: 57 24 4a 28 <unknown>
268
269 vasubu.vv v8, v4, v20
270 # CHECK-INST: vasubu.vv v8, v4, v20
271 # CHECK-ENCODING: [0x57,0x24,0x4a,0x2a]
272 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
273 # CHECK-UNKNOWN: 57 24 4a 2a <unknown>
274
275 vasubu.vx v8, v4, a0, v0.t
276 # CHECK-INST: vasubu.vx v8, v4, a0, v0.t
277 # CHECK-ENCODING: [0x57,0x64,0x45,0x28]
278 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
279 # CHECK-UNKNOWN: 57 64 45 28 <unknown>
280
281 vasubu.vx v8, v4, a0
282 # CHECK-INST: vasubu.vx v8, v4, a0
283 # CHECK-ENCODING: [0x57,0x64,0x45,0x2a]
284 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
285 # CHECK-UNKNOWN: 57 64 45 2a <unknown>
```

llvm/test/MC/RISCV/rvv/vsetvl.s

View Options

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

```
1 # RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
```

```
# RUN:        | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:        | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d --mattr=+experimental-v - \
# RUN:        | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

```

kito-cheng Not Done

Add a testcase for `vsetvli a2, a0, e32`, spec say LMUL can be omitted and default to m1[1], and need negative testcase would be better, e.g. feed `e31` and `m3`.

[1] https://riscv.github.io/documents/riscv-v-spec/#_vsetvlivsetvl_instructions

```
vsetvli a2, a0, e32,m4
# CHECK-INST: vsetvli a2, a0, e32,m4
# CHECK-ENCODING: [0x57,0x76,0xa5,0x00]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 76 a5 00 <unknown>

vsetvl a2, a0, a1
# CHECK-INST: vsetvl a2, a0, a1
# CHECK-ENCODING: [0x57,0x76,0xb5,0x80]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 76 b5 80 <unknown>
```

llvm/test/MC/RISCV/rvv/xor.s

View Options

This file was added.

```
# RUN: llvm-mc -triple=riscv64 -show-encoding --mattr=+experimental-v < %s \
# RUN:        | FileCheck %s --check-prefixes=CHECK-ENCODING,CHECK-INST
# RUN: not llvm-mc -triple=riscv64 -show-encoding < %s 2>&1 \
# RUN:        | FileCheck %s --check-prefix=CHECK-ERROR
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d --mattr=+experimental-v - \
# RUN:        | FileCheck %s --check-prefix=CHECK-INST
# RUN: llvm-mc -triple=riscv64 -filetype=obj --mattr=+experimental-v < %s \
# RUN:        | llvm-objdump -d - | FileCheck %s --check-prefix=CHECK-UNKNOWN

vxor.vv v8, v4, v20, v0.t
# CHECK-INST: vxor.vv v8, v4, v20, v0.t
# CHECK-ENCODING: [0x57,0x04,0x4a,0x2c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 2c <unknown>

vxor.vv v8, v4, v20
# CHECK-INST: vxor.vv v8, v4, v20
# CHECK-ENCODING: [0x57,0x04,0x4a,0x2e]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
# CHECK-UNKNOWN: 57 04 4a 2e <unknown>

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.

0, v0.t
45,0x2c]
# CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
```

```
27 # CHECK-UNKNOWN: 57 44 45 2c <unknown>
28
29 vxor.vx v8, v4, a0
30 # CHECK-INST: vxor.vx v8, v4, a0
31 # CHECK-ENCODING: [0x57,0x44,0x45,0x2e]
32 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
33 # CHECK-UNKNOWN: 57 44 45 2e <unknown>
34
35 vxor.vi v8, v4, 15, v0.t
36 # CHECK-INST: vxor.vi v8, v4, 15, v0.t
37 # CHECK-ENCODING: [0x57,0xb4,0x47,0x2c]
38 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
39 # CHECK-UNKNOWN: 57 b4 47 2c <unknown>
40
41 vxor.vi v8, v4, 15
42 # CHECK-INST: vxor.vi v8, v4, 15
43 # CHECK-ENCODING: [0x57,0xb4,0x47,0x2e]
44 # CHECK-ERROR: instruction requires the following: 'V' (Vector Instructions)
45 # CHECK-UNKNOWN: 57 b4 47 2e <unknown>
```

Add Action...

Submit

Your browser timezone setting differs from the timezone setting in your profile, click to reconcile.