Zceb 0.52

This document is in the Stable state. Assume anything could still change, but limited change should be expected. For more information see: https://riscv.org/spec-state

This extension reuses encodings from the *D*-extension. Therefore it is *incompatible* with *D*. It is fully compatible with *F* and also with *Zdinx*.

The instructions are all 16-bit versions of existing 32-bit load/store instructions.

RV32	RV64	RV128	Mnemonic	Instruction
✓	✓	✓	c.lbu *rd'*, uimm(*rs1'*)	Load unsigned byte, 16-bit encoding
✓	✓	✓	c.lb *rd'*, uimm(*rs1'*)	Load signed byte, 16-bit encoding
✓	✓	✓	c.lhu *rd'*, uimm(*rs1'*)	Load unsigned half, 16-bit encoding
✓	✓	✓	c.lh *rd'*, uimm(*rs1'*)	Load signed half, 16-bit encoding
✓	✓	✓	c.sb *rs2'*, uimm(*rs1'*)	Store byte, 16-bit encoding
✓	✓	✓	c.sh *rs2'*, uimm(*rs1'*)	Store byte, 16-bit encoding

c.lbu

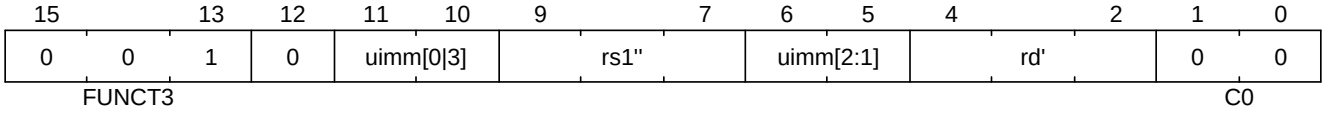
Synopsis

Load unsigned byte, 16-bit encoding

Mnemonic

c.lbu *rd'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction loads a byte from the memory address formed by adding *rs1'* to the zero extended immediate *uimm*. The resulting byte is zero extended to XLEN bits and is written to *rd'*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-lbu]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

X(rdc) = zext(mem[X(rs1c)+zext(imm)][7:0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable

c.lb

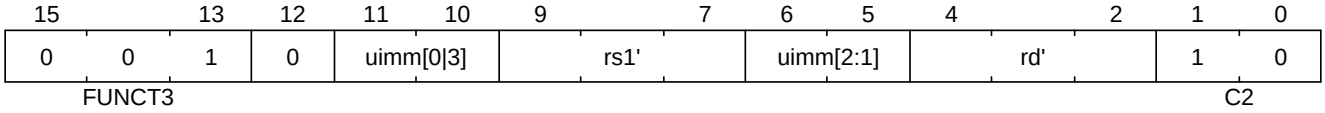
Synopsis

Load signed byte, 16-bit encoding

Mnemonic

c.lb *rd'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction loads a byte from the memory address formed by adding *rs1'* to the zero extended immediate *uimm*. The resulting byte is sign extended to XLEN bits and is written to *rd'*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-lb]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

X(rdc) = sext(mem[X(rs1c)+zext(imm)][7:0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable

c.lhu

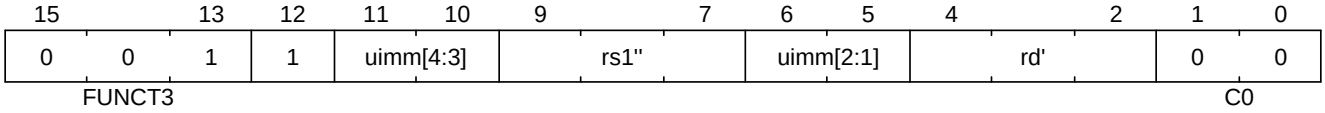
Synopsis

Load unsigned half, 16-bit encoding

Mnemonic

c.lhu *rd'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction loads a half from the memory address formed by adding *rs1'* to the zero extended immediate *uimm*. The resulting half is zero extended to XLEN bits and is written to *rd'*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-lhu]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

X(rdc) = zext(mem[X(rs1c)+zext(imm)][15:0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable

c.lh

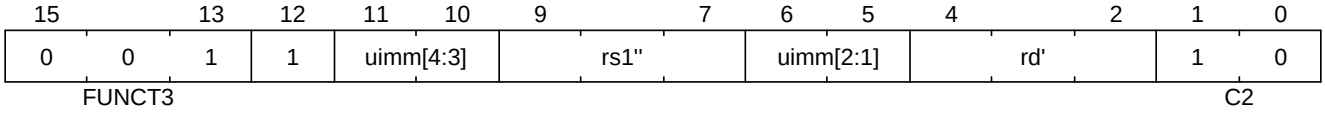
Synopsis

Load signed half, 16-bit encoding

Mnemonic

c.lh *rd'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction loads a half from the memory address formed by adding *rs1'* to the zero extended immediate *uimm*. The resulting half is sign extended to XLEN bits and is written to *rd'*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-lh]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

X(rdc) = sext(load_mem[X(rs1c)+zext(imm)][15:0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable

c.sb

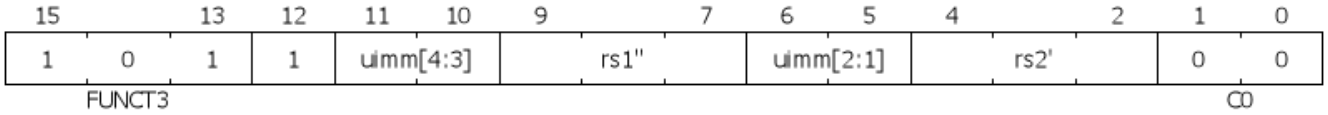
Synopsis

Store byte, 16-bit encoding

Mnemonic

c.sb *r2'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction stores the least significant half of *rs2'* to the memory address formed by adding *rs1'* to the zero extended immediate *uimm*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-sb]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

mem[X(rs1c)+zext(uimm)][7:0] = X(rs2c)
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable

c.sh

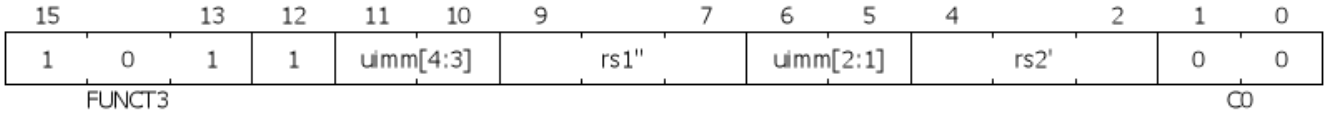
Synopsis

Store byte, 16-bit encoding

Mnemonic

c.sh *r2'*, *uimm*(*rs1'*)

Encoding (RV32, RV64, RV128)



Description

This instruction stores the least significant byte of *rs2'* to the memory address formed by adding *rs1'* to the zero extended immediate *uimm*.

NOTE *rd'* and *rs1'* are from the standard 8-register set x8-x15.

Prerequisites

The C-extension. This encoding conflicts with the D-extension, but there is no conflict with Zdinx if double-precision arithmetic is required.

32-bit equivalent

[insns-sh]

Operation

```
//This is not SAIL, it's pseudo-code. The SAIL hasn't been written yet.

mem[X(rs1c)+zext(uimm)][15:0] = X(rs2c)
```

Included in

Extension	Minimum version	Lifecycle state
Zceb ([zceb])	0.52	Stable