Zceb

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.lb

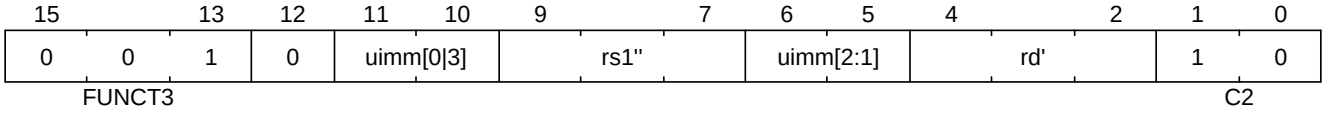
Synopsis

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' is from the standard 8-register set x8-x15. *rs1''* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rd    = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
X(rd) = sext(mem[X(rs1)+zext(imm)][7:0]);
```

Included in

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

c.lbu

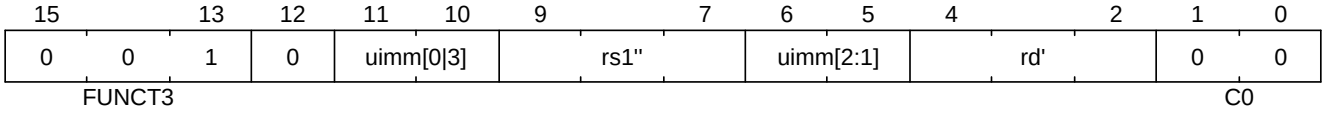
Synopsis

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' is from the standard 8-register set x8-x15. *rs1''* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rd    = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
X(rd) = zext(mem[X(rs1)+zext(imm)][7:0]);
```

Included in

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

c.lh

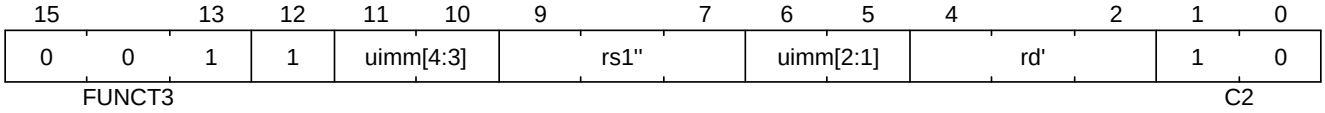
Synopsis

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'* is from the standard 8-register set x8-x15. *rs1''* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rd    = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
X(rd) = sext(load_mem[X(rs1)+zext(imm)][15:0]);
```

Included in

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

c.lhu

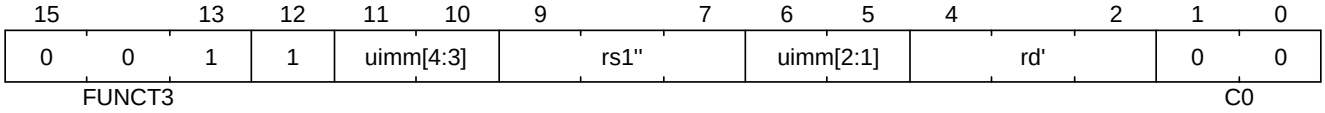
Synopsis

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'* is from the standard 8-register set x8-x15. *rs1''* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rd    = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
X(rd) = zext(mem[X(rs1)+zext(imm)][15:0]);
```

Included in

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

c.sb

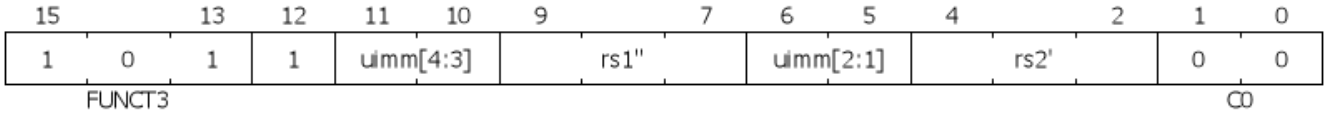
Synopsis

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 *rs2′* is from the standard 8-register set x8-x15. *rs1′′* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rs2   = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
mem[X(rs1)+zext(uimm)][7:0] = X(rs2)
```

Included in

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

c.sh

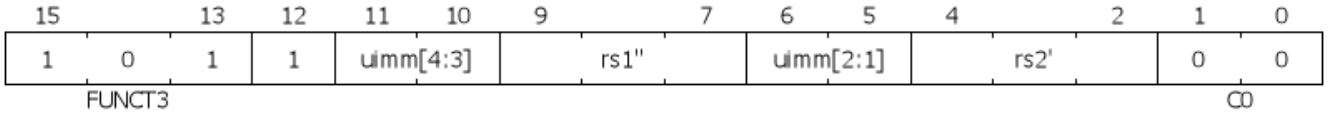
Synopsis

Store byte, 16-bit encoding

Mnemonic

c.sh *rs2′*, *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 *rs2′* is from the standard 8-register set x8-x15. *rs1′′* replaces x12 (a2) with x2 (sp) for additional code size saving. Therefore the 8-register set is x8-x11,x2,x13-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.

Operation

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

rs2   = encoding[9:7]+8;
rs1   = encoding[4:2]==4 ? 2 : 8+encoding[4:2];
mem[X(rs1)+zext(uimm)][15:0] = X(rs2)
```

Included in

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