Zcee 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

The instructions in Zcee are all very simple and are 16-bit encodings of existing instructions, either from the I/E ISA or from Zbb.

All proposed encodings are currently reserved for all architectures, and have no conflicts with any existing extensions.

All of the sign and zero extension instructions use the same format. These typically benefit code-size because if the compiler is working with fewer than XLEN bits it must sign/zero extend the value to XLEN bits before calling a function, or passing a return value.

c.sext.*, c.zext.* encoding group

15					10	9 7	6	5	4 2	1	0
1	0	0	0	0	0	rsd'	0	0	opcode	0	0
FUNCT6						SRCDST	FUNCT2			OP=C0	

The *c.mul* encoding has the register sources/destination in the same fields as other 16-bit encodings such as *c.sub*, *c.xor* etc.

c.mul has a high benefit for audio processing algorithms and for a variety of other benchmarks.

c.mul encoding

15					10	9 7	6	5	4 2	1	0
1	0	0	1	1	1	rsd'	1	0	rs2'	0	1
FUNCT6						SRCDST	FUNCT2		SRC2	OP=C1	

NOTE `c.sext.w` will be a pseudo-instruction for `c.addiw rd, 0` (RV64/RV128)

RV 32	RV 64	RV 128	Mnemonic	Instruction
✓	✓	✓	c.sext.b *rsd'*	Sign extend byte, 16-bit encoding
✓	✓	✓	c.sext.h *rsd'*	Sign extend half, 16-bit encoding
✓	✓	✓	c.zext.b *rsd'*	Zero extend byte, 16-bit encoding
✓	✓	✓	c.zext.h *rsd'*	Zero extend halfword, 16-bit encoding
	✓	✓	c.zext.w *rsd'*	Zero extend word, 16-bit encoding
✓	✓	✓	c.mul *rsd'*, *rs2'*	Multiply, 16-bit encoding

c.sext.b

Synopsis

Sign extend byte, 16-bit encoding

Mnemonic

c.sext.b *rsd'*

Encoding (RV32, RV64, RV128)

15					10	9 7	6	5	4		2	1	0
1	0	0	0	0	0	rsd'	0	0	0	0	1	0	0
FUNCT6						SRCDST	FUNCT2		C.SEXT.B			OP=C0	

Description

This instruction takes a single source/destination operand, from the 8-register set x8-x15. It sign-extends the least-significant byte in the operand to XLEN by copying the most-significant bit in the byte (i.e., bit 7) to all of the more-significant bits.

Prerequisites

The C-extension must also be configured.

32-bit equivalent

[insns-sext_b] from Zbb

Operation

```
X(rsdc) = EXTS(X(rsdc)[7..0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable

c.sext.h

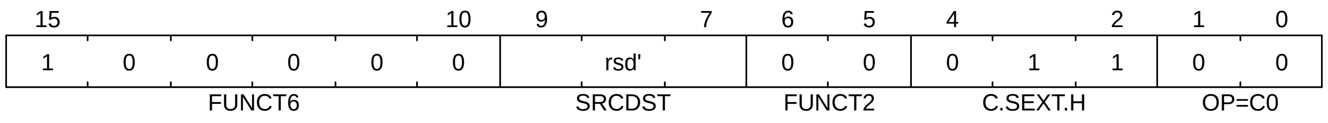
Synopsis

Sign extend half, 16-bit encoding

Mnemonic

c.sext.h *rsd'*

Encoding (RV32, RV64, RV128)



Description

This instruction takes a single source/destination operand, from the 8-register set x8-x15. It sign-extends the least-significant halfword in the operand to XLEN by copying the most-significant bit in the halfword (i.e., bit 15) to all of the more-significant bits.

Prerequisites

The C-extension must also be configured.

32-bit equivalent

[insns-sext_h] from Zbb

Operation

```
X(rsdc) = EXTS(X(rsdc)[15..0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable

c.zext.b

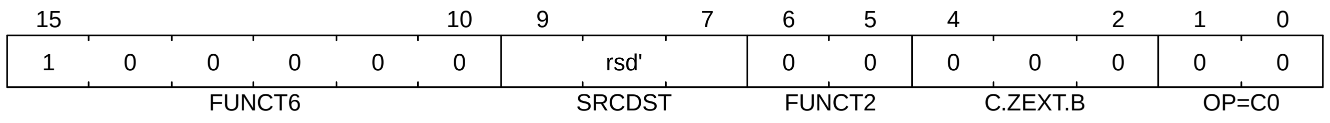
Synopsis

Zero extend byte, 16-bit encoding

Mnemonic

c.zext.b *rsd'*

Encoding (RV32, RV64, RV128)



Description

This instruction takes a single source/destination operand, from the 8-register set x8-x15. It zero-extends the least-significant byte of the operand to XLEN by inserting 0's into all of the bits more significant than 7.

Prerequisites

The C-extension must also be configured.

32-bit equivalent

```
andi rd, rs1, 0xff
```

Operation

```
X(rsdc) = EXTZ(X(rsdc)[7..0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable

c.zext.h

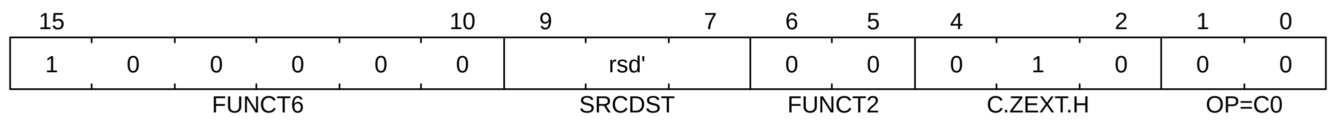
Synopsis

Zero extend halfword, 16-bit encoding

Mnemonic

c.zext.h *rsd'*

Encoding (RV32, RV64, RV128)



Description

This instruction takes a single source/destination operand, from the 8-register set x8-x15. It zero-extends the least-significant halfword of the operand to XLEN by inserting 0's into all of the bits more significant than 15.

Prerequisites

The C-extension must also be configured.

32-bit equivalent

[insns-zext_h] from Zbb

Operation

```
X(rsdc) = EXTZ(X(rsdc)[15..0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable

c.zext.w

Synopsis

Zero extend word, 16-bit encoding

Mnemonic

c.zext.w *rsd'*

Encoding (RV64, RV128)

15					10	9 — 7	6	5	4		2	1	0
1	0	0	0	0	0	rsd'	0	0	1	0	0	0	0
FUNCT6						SRCDST	FUNCT2		C.ZEXT.W			OP=C0	

Description

This instruction takes a single source/destination operand, from the 8-register set x8-x15. It zero-extends the least-significant word of the operand to XLEN by inserting 0's into all of the bits more significant than 31.

Prerequisites

The C-extension must also be configured.

32-bit equivalent

```
add.uw rd, rs1, zero
```

Operation

```
X(rsdc) = EXTZ(X(rsdc)[31..0]);
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable

c.mul

Synopsis

Multiply, 16-bit encoding

Mnemonic

c.mul *rsd′*, *rs2′*

Encoding (RV32, RV64, RV128)

15					10	9		7	6	5	4		2	1	0
1	0	0	1	1	1	rsd'			1	0	rs2'			0	1
FUNCT6						SRCDST			FUNCT2		SRC2			OP=C1	

Description

This instruction multiplies XLEN bits of the source operands from *rsd′* and *rs2′* and writes the lowest XLEN bits of the result to *rsd′*. Both operands are from the 8-register set x8-x15.

Prerequisites

The C-extension and either M or Zmmul must also be configured.

32-bit equivalent

[insns-mul]

Operation

```
let result_wide = to_bits(2 * sizeof(xlen), signed(X(rsdc)) * signed(X(rs2c)));
X(rsdc) = result_wide[(sizeof(xlen) - 1) .. 0];
```

Included in

Extension	Minimum version	Lifecycle state
Zcee ([zcee])	0.52	Stable