

## Zcee 0.53.2

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.

The *c.mul* encoding uses the CR register format along with other instructions such as *c.sub*, *c.xor* etc.

### NOTE

*c.sext.w* is a pseudo-instruction for *c.addiw rd, 0* (RV64)

RV32	RV64	Mnemonic	Instruction
✓	✓	<i>c.zext.b rsd'</i>	<a href="#">Zero extend byte, 16-bit encoding</a>
✓	✓	<i>c.sext.b rsd'</i>	<a href="#">Sign extend byte, 16-bit encoding</a>
✓	✓	<i>c.zext.h rsd'</i>	<a href="#">Zero extend halfword, 16-bit encoding</a>
✓	✓	<i>c.sext.h rsd'</i>	<a href="#">Sign extend halfword, 16-bit encoding</a>
	✓	<i>c.zext.w rsd'</i>	<a href="#">Zero extend word, 16-bit encoding</a>
✓	✓	<i>c.mul rsd', rs2'</i>	<a href="#">Multiply, 16-bit encoding</a>

# c.zext.b

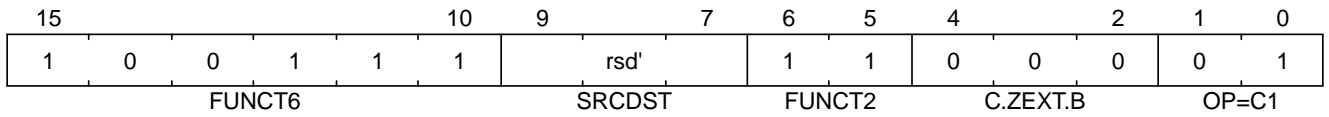
## Synopsis

Zero extend byte, 16-bit encoding

## Mnemonic

c.zext.b *rsd'*

## Encoding (RV32, RV64)



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

## Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable

# c.sext.b

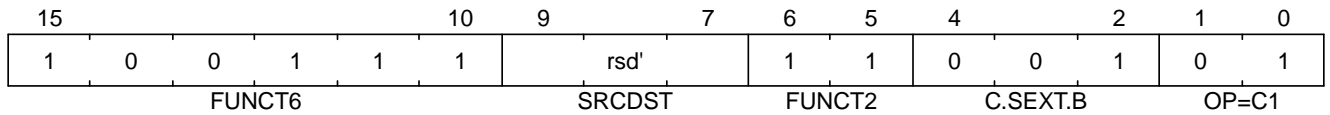
## Synopsis

Sign extend byte, 16-bit encoding

## Mnemonic

c.sext.b *rsd'*

## Encoding (RV32, RV64)



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

## Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable

# c.zext.h

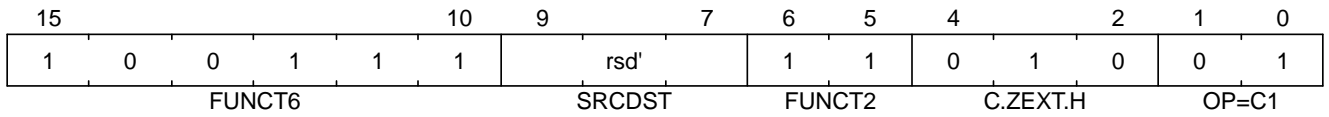
## Synopsis

Zero extend halfword, 16-bit encoding

## Mnemonic

c.zext.h *rsd'*

## Encoding (RV32, RV64)



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

## Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable

# c.sext.h

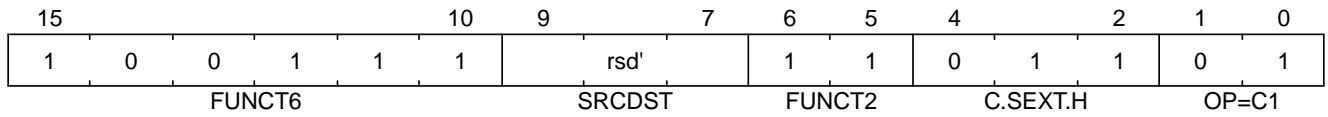
## Synopsis

Sign extend halfword, 16-bit encoding

## Mnemonic

c.sext.h *rsd'*

## Encoding (RV32, RV64)



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

## Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable

## c.zext.w

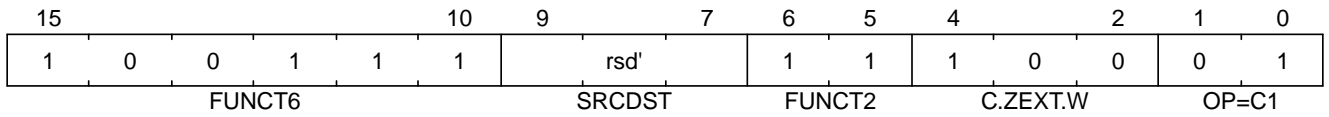
### Synopsis

Zero extend word, 16-bit encoding

### Mnemonic

c.zext.w *rsd'*

### Encoding (RV64)



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

### Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable

# c.mul

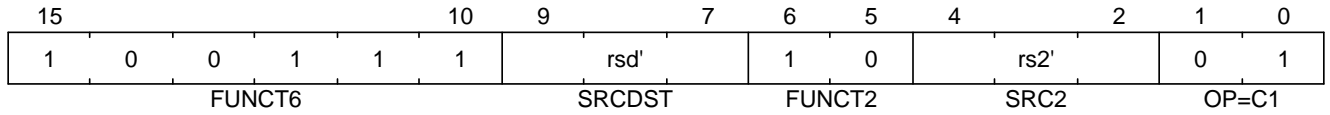
## Synopsis

Multiply, 16-bit encoding

## Mnemonic

c.mul *rsd'*, *rs2'*

## Encoding (RV32, RV64)



## 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(rsd)) * signed(X(rs2)));
X(rsd) = result_wide[(sizeof(xlen) - 1) .. 0];
```

## Included in

Extension	Minimum version	Lifecycle state
Zcee ( <a href="#">Zcee 0.53.2</a> )	0.53.2	Stable