This document demonstrates the impact of using unspecified versus signed extensions during the BitInt ABI. BitInt is a C23 feature that allows developers to designate the precise bit requirement for a variable.

For example:

```cx
_BitInt(17) test_bitint_add(_BitInt(17) a, _BitInt(17) b) {
    return a + b;
}
```

The variable a can store 17 bits and perform computations on it. The problem is how to define the bits that are not being used.

This document compares four different styles: unspecified extend, sign extend, zero extend, and mixed extend style. This document will show the final RISC-V assembly code for each style.

Addition

```c
#include <stdbool.h>


unsigned _BitInt(17) test_bitint_17_add_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a + b;
}

signed _BitInt(17) test_bitint_17_add_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a + b;
}
```

Unspecific

```c
test_bitint_17_add_unsigned:          # @test_bitint_17_add_unsigned
        add     a0, a0, a1
        ret
test_bitint_17_add_signed:            # @test_bitint_17_add_signed
        add     a0, a0, a1
```

```
        ret

```

Signed extend

```c
test_bitint_17_add_unsigned:          # @test_bitint_17_add_unsigned
        add     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_add_signed:            # @test_bitint_17_add_signed
        add     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret

```

Zero extend

```c
test_bitint_17_add_unsigned:          # @test_bitint_17_add_unsigned
        add     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
test_bitint_17_add_signed:            # @test_bitint_17_add_signed
        add     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```

Mix extend

```c
test_bitint_17_add_unsigned:          # @test_bitint_17_add_unsigned
        add     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
```

```
test_bitint_17_add_signed:              # @test_bitint_17_add_signed
        add     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```
```

Subtraction

```c

#include <stdbool.h>

unsigned _BitInt(17) test_bitint_17_sub_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17)
b) {
    return a - b;
}

signed _BitInt(17) test_bitint_17_sub_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a - b;
}

```

Unspecific

```c
test_bitint_17_sub_unsigned:            # @test_bitint_17_sub_unsigned
        sub     a0, a0, a1
        ret
test_bitint_17_sub_signed:              # @test_bitint_17_sub_signed
        sub     a0, a0, a1
        ret

```

Signed extend

```c
```
```

```
test_bitint_17_sub_unsigned:          # @test_bitint_17_sub_unsigned
        subw    a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_sub_signed:            # @test_bitint_17_sub_signed
        subw    a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
```

Zero extend

```c
test_bitint_17_sub_unsigned:          # @test_bitint_17_sub_unsigned
        subw    a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
test_bitint_17_sub_signed:            # @test_bitint_17_sub_signed
        subw    a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
```

Mix extend

```c
test_bitint_17_sub_unsigned:          # @test_bitint_17_sub_unsigned
        subw    a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
test_bitint_17_sub_signed:            # @test_bitint_17_sub_signed
        subw    a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
```

Multiplication

```c
#include <stdbool.h>

unsigned _BitInt(17) test_bitint_17_mul_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17)
b) {
    return a * b;
}

signed _BitInt(17) test_bitint_17_mul_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a * b;
}

```

Unspecific

```c
test_bitint_17_mul_unsigned:          # @test_bitint_17_mul_unsigned
        mul     a0, a1, a0
        ret
test_bitint_17_mul_signed:            # @test_bitint_17_mul_signed
        mul     a0, a1, a0
        ret

```

Signed extend

```c
test_bitint_17_mul_unsigned:          # @test_bitint_17_mul_unsigned
        mul     a0, a1, a0
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_mul_signed:            # @test_bitint_17_mul_signed
        mul     a0, a1, a0
        slli    a0, a0, 47
```

```
        srai    a0, a0, 47
        ret

```

Zero extend

```c
test_bitint_17_mul_unsigned:          # @test_bitint_17_mul_unsigned
        mul     a0, a1, a0
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
test_bitint_17_mul_signed:            # @test_bitint_17_mul_signed
        mul     a0, a1, a0
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```

Mix extend

```c
test_bitint_17_mul_unsigned:          # @test_bitint_17_mul_unsigned
        mul     a0, a1, a0
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret
test_bitint_17_mul_signed:            # @test_bitint_17_mul_signed
        mul     a0, a1, a0
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```

Division

```c
```

```c
#include <stdbool.h>

unsigned _BitInt(17) test_bitint_17_div_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b)
{
    return a / b;
}

signed _BitInt(17) test_bitint_17_div_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a / b;
}
```

Unspecific

```c
test_bitint_17_div_unsigned:          # @test_bitint_17_div_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        divu    a0, a0, a1
        ret
test_bitint_17_div_signed:            # @test_bitint_17_div_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        div     a0, a0, a1
        ret
```

Signed extend

```c
test_bitint_17_div_unsigned:          # @test_bitint_17_div_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        divu    a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
```

```
        ret
test_bitint_17_div_signed:          # @test_bitint_17_div_signed
        div     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret

```

Zero extend

```c
test_bitint_17_div_unsigned:          # @test_bitint_17_div_unsigned
        divu    a0, a0, a1
        ret
test_bitint_17_div_signed:          # @test_bitint_17_div_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        div     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```

Mix extend

```c
test_bitint_17_div_unsigned:          # @test_bitint_17_div_unsigned
        divu    a0, a0, a1
        ret
test_bitint_17_div_signed:          # @test_bitint_17_div_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        div     a0, a0, a1
        slli    a0, a0, 47
        srli    a0, a0, 47
        ret

```

Equal

```c
#include <stdbool.h>

bool test_bitint_17_eq_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a == b;
}

bool test_bitint_17_eq_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a == b;
}

```

Unspecific

```c
test_bitint_17_eq_unsigned:             # @test_bitint_17_eq_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        xor     a0, a0, a1
        seqz    a0, a0
        ret
test_bitint_17_eq_signed:               # @test_bitint_17_eq_signed
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        xor     a0, a0, a1
        seqz    a0, a0
        ret

```

Signed extend

```c
test_bitint_17_eq_unsigned:          # @test_bitint_17_eq_unsigned
        xor     a0, a0, a1
        seqz    a0, a0
        ret
test_bitint_17_eq_signed:            # @test_bitint_17_eq_signed
        xor     a0, a0, a1
        seqz    a0, a0
        ret

```

Zero extend

```c
test_bitint_17_eq_unsigned:          # @test_bitint_17_eq_unsigned
        xor     a0, a0, a1
        seqz    a0, a0
        ret
test_bitint_17_eq_signed:            # @test_bitint_17_eq_signed
        xor     a0, a0, a1
        seqz    a0, a0
        ret

```

Mix extend

```c
test_bitint_17_eq_unsigned:          # @test_bitint_17_eq_unsigned
        xor     a0, a0, a1
        seqz    a0, a0
        ret
test_bitint_17_eq_signed:            # @test_bitint_17_eq_signed
        xor     a0, a0, a1
        seqz    a0, a0
        ret

```

Greater than or equal to

```c
#include <stdbool.h>

bool test_bitint_17_ge_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a >= b;
}

bool test_bitint_17_ge_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a >= b;
}

```

Unspecific

```c
test_bitint_17_ge_unsigned:             # @test_bitint_17_ge_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        sltu    a0, a0, a1
        xori    a0, a0, 1
        ret
test_bitint_17_ge_signed:               # @test_bitint_17_ge_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        xori    a0, a0, 1
        ret

```

Signed extend

```c
test_bitint_17_ge_unsigned:             # @test_bitint_17_ge_unsigned
        sltu    a0, a0, a1
        xori    a0, a0, 1
        ret
```

```
test_bitint_17_ge_signed:               # @test_bitint_17_ge_signed
        slt     a0, a0, a1
        xori    a0, a0, 1
        ret

```

Zero extend

```c
test_bitint_17_ge_unsigned:             # @test_bitint_17_ge_unsigned
        sltu    a0, a0, a1
        xori    a0, a0, 1
        ret
test_bitint_17_ge_signed:               # @test_bitint_17_ge_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        xori    a0, a0, 1
        ret

```

Mix extend

```c
test_bitint_17_ge_unsigned:             # @test_bitint_17_ge_unsigned
        sltu    a0, a0, a1
        xori    a0, a0, 1
        ret
test_bitint_17_ge_signed:               # @test_bitint_17_ge_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        xori    a0, a0, 1
        ret

```

Greater than

```c
#include <stdbool.h>

bool test_bitint_17_gt_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a > b;
}

bool test_bitint_17_gt_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a > b;
}

```

Unspecific

```c
test_bitint_17_gt_unsigned:             # @test_bitint_17_gt_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a0, a0, a2
        and     a1, a1, a2
        sltu    a0, a1, a0
        ret
test_bitint_17_gt_signed:               # @test_bitint_17_gt_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        ret

```

Signed extend

```c
test_bitint_17_gt_unsigned:             # @test_bitint_17_gt_unsigned
        sltu    a0, a1, a0
        ret
```

```
test_bitint_17_gt_signed:               # @test_bitint_17_gt_signed
        slt     a0, a1, a0
        ret

```

Zero extend

```c
test_bitint_17_gt_unsigned:             # @test_bitint_17_gt_unsigned
        sltu    a0, a1, a0
        ret
test_bitint_17_gt_signed:               # @test_bitint_17_gt_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        ret

```

Mix extend

```c
test_bitint_17_gt_unsigned:             # @test_bitint_17_gt_unsigned
        sltu    a0, a1, a0
        ret
test_bitint_17_gt_signed:               # @test_bitint_17_gt_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        ret

```

Less than or equal to

```c
#include <stdbool.h>


bool test_bitint_17_le_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a <= b;
}

bool test_bitint_17_le_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a <= b;
}
```

Unspecific

```c
test_bitint_17_le_unsigned:             # @test_bitint_17_le_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a0, a0, a2
        and     a1, a1, a2
        sltu    a0, a1, a0
        xori    a0, a0, 1
        ret
test_bitint_17_le_signed:               # @test_bitint_17_le_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        xori    a0, a0, 1
        ret

```

Signed extend

```c
test_bitint_17_le_unsigned:             # @test_bitint_17_le_unsigned
        sltu    a0, a1, a0
        xori    a0, a0, 1
        ret
test_bitint_17_le_signed:               # @test_bitint_17_le_signed
        slt     a0, a1, a0
```

```
        xori    a0, a0, 1
        ret

```

Zero extend

```c
test_bitint_17_le_unsigned:           # @test_bitint_17_le_unsigned
        sltu    a0, a1, a0
        xori    a0, a0, 1
        ret
test_bitint_17_le_signed:             # @test_bitint_17_le_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        xori    a0, a0, 1
        ret

```

Mix extend

```c
test_bitint_17_le_unsigned:           # @test_bitint_17_le_unsigned
        sltu    a0, a1, a0
        xori    a0, a0, 1
        ret
test_bitint_17_le_signed:             # @test_bitint_17_le_signed
        slli    a0, a0, 47
        slli    a1, a1, 47
        srai    a0, a0, 47
        srai    a1, a1, 47
        slt     a0, a1, a0
        xori    a0, a0, 1
        ret

```

Less than

```c
#include <stdbool.h>

bool test_bitint_17_lt_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17) b) {
    return a < b;
}

bool test_bitint_17_lt_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a < b;
}
```

Unspecific

```c
test_bitint_17_lt_unsigned:             # @test_bitint_17_lt_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        sltu    a0, a0, a1
        ret
test_bitint_17_lt_signed:               # @test_bitint_17_lt_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        ret

```

Signed extend

```c
test_bitint_17_lt_unsigned:             # @test_bitint_17_lt_unsigned
        sltu    a0, a0, a1
        ret
test_bitint_17_lt_signed:               # @test_bitint_17_lt_signed
        slt     a0, a0, a1
        ret
```

```

## Zero extend

```c
test_bitint_17_lt_unsigned:             # @test_bitint_17_lt_unsigned
        sltu    a0, a0, a1
        ret
test_bitint_17_lt_signed:               # @test_bitint_17_lt_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        ret

```

Mix extend

```c
test_bitint_17_lt_unsigned:             # @test_bitint_17_lt_unsigned
        sltu    a0, a0, a1
        ret
test_bitint_17_lt_signed:               # @test_bitint_17_lt_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        slt     a0, a0, a1
        ret

```

Modulo

```c
#include <stdbool.h>
```
```

```
unsigned _BitInt(17) test_bitint_17_mod_unsigned(unsigned _BitInt(17) a, unsigned _BitInt(17)
b) {
    return a % b;
}

signed _BitInt(17) test_bitint_17_mod_signed(signed _BitInt(17) a, signed _BitInt(17) b) {
    return a % b;
}
```

Unspecific

```c
test_bitint_17_mod_unsigned:          # @test_bitint_17_mod_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        remu    a0, a0, a1
        ret
test_bitint_17_mod_signed:            # @test_bitint_17_mod_signed
        slli    a1, a1, 47
        slli    a0, a0, 47
        srai    a1, a1, 47
        srai    a0, a0, 47
        remw    a0, a0, a1
        ret

```

Signed extend

```c
test_bitint_17_mod_unsigned:          # @test_bitint_17_mod_unsigned
        lui     a2, 32
        addi    a2, a2, -1
        and     a1, a1, a2
        and     a0, a0, a2
        remu    a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_mod_signed:            # @test_bitint_17_mod_signed
```

```
        remw   a0, a0, a1
        ret

```

Zero extend

```c
test_bitint_17_mod_unsigned:          # @test_bitint_17_mod_unsigned
        remu   a0, a0, a1
        ret
test_bitint_17_mod_signed:            # @test_bitint_17_mod_signed
        slli       a1, a1, 47
        slli       a0, a0, 47
        srai       a1, a1, 47
        srai       a0, a0, 47
        remw   a0, a0, a1
        slli       a0, a0, 47
        srli       a0, a0, 47
        ret

```

Mix extend

```c
test_bitint_17_mod_unsigned:          # @test_bitint_17_mod_unsigned
        remu   a0, a0, a1
        ret
test_bitint_17_mod_signed:            # @test_bitint_17_mod_signed
        slli       a1, a1, 47
        slli       a0, a0, 47
        srai       a1, a1, 47
        srai       a0, a0, 47
        remw   a0, a0, a1
        slli       a0, a0, 47
        srli       a0, a0, 47
        ret

```

Left shift

```c
#include <stdbool.h>

unsigned _BitInt(17) test_bitint_17_shift_left_unsigned(unsigned _BitInt(17) a, int n) {
    return a << n;
}

signed _BitInt(17) test_bitint_17_shift_left_signed(signed _BitInt(17) a, int n) {
    return a << n;
}

```

Unspecific

```c
test_bitint_17_shift_left_unsigned:     # @test_bitint_17_shift_left_unsigned
        sll     a0, a0, a1
        ret
test_bitint_17_shift_left_signed:       # @test_bitint_17_shift_left_signed
        sll     a0, a0, a1
        ret

```

Signed extend

```c
test_bitint_17_shift_left_unsigned:     # @test_bitint_17_shift_left_unsigned
        sll     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_shift_left_signed:       # @test_bitint_17_shift_left_signed
        sll     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret

```

Zero extend

```c
test_bitint_17_shift_left_unsigned:     # @test_bitint_17_shift_left_unsigned
        sll     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret
test_bitint_17_shift_left_signed:       # @test_bitint_17_shift_left_signed
        sll     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret

```

Mix extend

```c
test_bitint_17_shift_left_unsigned:     # @test_bitint_17_shift_left_unsigned
        sll     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret
test_bitint_17_shift_left_signed:       # @test_bitint_17_shift_left_signed
        sll     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret

```

Right shift

```c
#include <stdbool.h>
```

```
unsigned _BitInt(17) test_bitint_17_shift_right_unsigned(unsigned _BitInt(17) a, int n) {
    return a >> n;
}

signed _BitInt(17) test_bitint_17_shift_right_signed(signed _BitInt(17) a, int n) {
    return a >> n;
}
```

Unspecific

```c
test_bitint_17_shift_right_unsigned:    # @test_bitint_17_shift_right_unsigned
        slli    a0, a0, 47
        srli    a0, a0, 47
        srl     a0, a0, a1
        ret
test_bitint_17_shift_right_signed:      # @test_bitint_17_shift_right_signed
        slli    a0, a0, 47
        srai    a0, a0, 47
        sra     a0, a0, a1
        ret
```

Signed extend

```c
test_bitint_17_shift_right_unsigned:    # @test_bitint_17_shift_right_unsigned
        slli    a0, a0, 47
        srli    a0, a0, 47
        srl     a0, a0, a1
        slli    a0, a0, 47
        srai    a0, a0, 47
        ret
test_bitint_17_shift_right_signed:      # @test_bitint_17_shift_right_signed
        sra     a0, a0, a1
        ret
```

Zero extend

```c
test_bitint_17_shift_right_unsigned:    # @test_bitint_17_shift_right_unsigned
        srl     a0, a0, a1
        ret
test_bitint_17_shift_right_signed:      # @test_bitint_17_shift_right_signed
        slli    a0, a0, 47
        srai    a0, a0, 47
        sra     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret

```

Mix extend

```c
test_bitint_17_shift_right_unsigned:    # @test_bitint_17_shift_right_unsigned
        srl     a0, a0, a1
        ret
test_bitint_17_shift_right_signed:      # @test_bitint_17_shift_right_signed
        slli    a0, a0, 47
        srai    a0, a0, 47
        sra     a0, a0, a1
        lui     a1, 32
        addi    a1, a1, -1
        and     a0, a0, a1
        ret

```