

## Registers

| name | othername | description         | save? | name | othername | description | save? |
|------|-----------|---------------------|-------|------|-----------|-------------|-------|
| x0   | zero      | always 0            | yes?  | x16  | a6        | argument 6  | -     |
| x1   | ra        | return address      | -     | x17  | a7        | argument 7  | -     |
| x2   | sp        | stack pointer       | yes   | x18  | s2        | saved       | yes   |
| x3   | gp        | global pointer      | -     | x19  | s3        | saved       | yes   |
| x4   | tp        | thread pointer      | -     | x20  | s4        | saved       | yes   |
| x5   | t0        | temp                | -     | x21  | s5        | saved       | yes   |
| x6   | t1        | temp                | -     | x22  | s6        | saved       | yes   |
| x7   | t2        | temp                | -     | x23  | s7        | saved       | yes   |
| x8   | s0, fp    | saved/frame pointer | yes   | x24  | s8        | saved       | yes   |
| x9   | s1        | saved               | yes   | x25  | s9        | saved       | yes   |
| x10  | a0        | argument 0          | -     | x26  | s10       | saved       | yes   |
| x11  | a1        | argument 1          | -     | x27  | s11       | saved       | yes   |
| x12  | a2        | argument 2          | -     | x28  | t3        | temp        | -     |
| x13  | a3        | argument 3          | -     | x29  | t4        | temp        | -     |
| x14  | a4        | argument 4          | -     | x30  | t5        | temp        | -     |
| x15  | a5        | argument 5          | -     | x31  | t6        | temp        | -     |

## Arithmetic

### Addition and subtraction

| name          | format           | example          | definition      |
|---------------|------------------|------------------|-----------------|
| add immediate | addi rd, r1, imm | addi x4, x9, 123 | $x4 = x9 + 123$ |
| add           | add rd, r1, r2   | add x4, x9, x13  | $x4 = x9 + x13$ |
| subtract      | sub rd, r1, r2   | sub x4, x9, x13  | $x4 = x9 - x13$ |
| *negate       | neg rd, r2       | neg x4, x9       | $x4 = -x9$      |

### Multiplication and division

| name                                     | format            | example            | definition                       |
|--|-------------------|--------------------|----------------------------------|
| multiply                                 | mul rd, r1, r2    | mul x4, x9, x13    | $x4 = x9 * x13$                  |
| multiply high bits (signed)              | mulh rd, r1, r2   | mulh x4, x9, x13   | $x4 = \text{highbits}(x9 * x13)$ |
| multiply high bits (unsigned)            | mulhu rd, r1, r2  | mulhu x4, x9, x13  | $x4 = \text{highbits}(x9 * x13)$ |
| multiply high bits (signed and unsigned) | mulhsu rd, r1, r2 | mulhsu x4, x9, x13 | $x4 = \text{highbits}(x9 * x13)$ |
| divide (signed)                          | div rd, r1, r2    | div x4, x9, x13    | $x4 = x9 \text{ div } x13$       |
| divide (unsigned)                        | divu rd, r1, r2   | divu x4, x9, x13   | $x4 = x9 \text{ div } x13$       |
| remainder (signed)                       | rem rd, r1, r2    | rem x4, x9, x13    | $x4 = x9 \text{ rem } x13$       |
| remainder (unsigned)                     | remu rd, r1, r2   | remu x4, x9, x13   | $x4 = x9 \text{ rem } x13$       |

## Loads

| name                     | format          | example          | definition                    |
|--------------------------|-----------------|------------------|-------------------------------|
| load byte (signed)       | lb rd, imm(r1)  | lb x4, 1234(x9)  | $x4 = \text{mem}[x9 + 1234]$  |
| load byte (unsigned)     | lbu rd, imm(r1) | lbu x4, 1234(x9) | $x4 = \text{mem}[x9 + 1234]$  |
| *load byte               | lb rd, imm      | lb x4, var       | $x4 = \text{mem}[\text{var}]$ |
| load halfword (signed)   | lh rd, imm(r1)  | lh x4, 1234(x9)  | $x4 = \text{Mem}[x9+1234]$    |
| load halfword (unsigned) | lhu rd, imm(r1) | lhu x4, 1234(x9) | $x4 = \text{Mem}[x9+1234]$    |
| *load halfword           | lh rd, imm      | lh x4, var       | $x4 = \text{mem}[\text{var}]$ |
| load word (signed)       | lw rd, imm(r1)  | lw x4, 1234(x9)  | $x4 = \text{Mem}[x9+1234]$    |
| load word (unsigned)     | lwu rd, imm(r1) | lwu x4, 1234(x9) | $x4 = \text{Mem}[x9+1234]$    |
| *load word               | lw rd, imm      | lw x4, var       | $x4 = \text{mem}[\text{var}]$ |

## Stores

| name            | format         | example         | definition  |
|-----------------|----------------|-----------------|---|
| store byte      | sb rd, imm(r1) | sb x4, 1234(x9) | $\text{mem}[x9 + 1234] = x4$                        |
| *store byte     | sb rd, imm, r1 | sb x4, var, a0  | $\text{mem}[\text{var}] = x4, \text{a0 is temp}$    |
| store halfword  | sh rd, imm(r1) | sh x4, 1234(x9) | $\text{mem}[x9 + 1234] = x4$                        |
| *store halfword | sh rd, imm, r1 | sh x4, var, a0  | $7 \text{ mem}[\text{var}] = x4, \text{a0 is temp}$ |
| store word      | sw rd, imm(r1) | sw x4, 1234(x9) | $\text{mem}[x9 + 1234] = x4$                        |
| *store word     | sw rd, imm, r1 | sw x4, var, a0  | $\text{mem}[\text{var}] = x4, \text{a0 is temp}$    |

### misc

| name                             | format             | example           | definition                |
|----------------------------------|--------------------|-------------------|---------------------------|
| *nop                             | nop                | nop               | does nothing              |
| *move register to register       | mv rd, r1          | mv x4, x9         | x4 = x9                   |
| load upper immediate             | lui rd, immed      | lui x4 0x12345    | x4 = 0x12345000           |
| *load immediate                  | li rd, immed       | li x4, 123        | x4 = 123                  |
| add upper immediate to pc        | auipc rd, immed    | auipc x4, 0x12345 | x4 = pc + (0x12345 << 12) |
| *load address                    | la rd, address     | la x4, loop       | x4 = loop                 |
| set if less then (signed)        | slt rd, r1, r2     | slt x4, x9, x13   | x4 = (x9 < x13) ? 1 : 0   |
| set less then immediate (signed) | slti rd, r1, immed | slti x4, x9, 123  | x4 = (x9 < 123) ? 1 : 0   |

### Jumps

| name                    | format             | example           | definition              |
|-------------------------|--------------------|-------------------|-------------------------|
| *jump and link          | jal rd, immed      | jal x1, loop      | goto loop, x1 = pc      |
| *jump                   | j immed            | jal loop          | goto loop               |
| *jump and link register | jalr rd, r1, immed | jalr x1, x4, loop | goto loop + x4, x1 = pc |
| *jump register          | jr r1              | jr x4             | goto r1                 |
| *return                 | ret                | ret               | goto x1                 |
| *call far               | call immed         | call func         | goto func, x1 = pc      |
| *tail call far          | tail immed         | tail func         | goto func, discard pc   |

jalr reg is the same as jalr x0, reg, 0.

### Branches

| name                                       | format             | example           | definition            |
|--|--------------------|-------------------|-----------------------|
| branch if equal                            | beq r1, r2, immed  | beq x4, x9, loop  | if x4 == x9 goto loop |
| branch if not equal                        | bne r1, r2, immed  | bne x4, x9, loop  | if x4 != x9 goto loop |
| branch if less than (signed)               | blt r1, r2, immed  | blt x4, x9, loop  | if x4 < x9 goto loop  |
| *branch if less than or equal (signed)     | ble r1, r2, immed  | ble x4, x9, loop  | if x4 ≤ x9 goto loop  |
| *branch if greater then (signed)           | bgt r1, r2, immed  | bgt x4, x9, loop  | if x4 > x9 goto loop  |
| branch if greater than or equal (signed)   | bge r1, r2, immed  | bge x4, x9, loop  | if x4 ≥ x9 goto loop  |
| branch if less than (unsigned)             | bltu r1, r2, immed | bltu x4, x9, loop | if x4 < x9 goto loop  |
| *branch if less than or equal (unsigned)   | bleu r1, r2, immed | bleu x4, x9, loop | if x4 ≤ x9 goto loop  |
| *branch if greater then (unsigned)         | bgtu r1, r2, immed | bgtu x4, x9, loop | if x4 > x9 goto loop  |
| branch if greater than or equal (unsigned) | bgeu r1, r2, immed | bgeu x4, x9, loop | if x4 ≥ x9 goto loop  |

If r2 is omitted, it is assumed to be x0.

### Bitwise

| name          | format             | example          | definition      |
|---------------|--------------------|------------------|-----------------|
| and immediate | andi rd, r1, immed | andi x4, x9, 123 | x4 = x9 & 123   |
| and           | and rd, r1, r2     | and x4, x9, x13  | x4 = x9 & x13   |
| or immediate  | ori rd, r1, immed  | ori x4, x9, 123  | x4 = x9   123   |
| or            | or rd, r1, r2      | or x4, x9, x13   | x4 = x9   x13   |
| xor immediate | xori rd, r1, immed | xori x4, x9, 123 | x4 = x9 xor 123 |
| xor           | xor rd, r1, r2     | xor x4, x9, x13  | x4 = x9 xor x13 |
| *not          | not rd, r1         | not x4, x9       | x4 = !x9        |

### Shifts

| name                             | format             | example          | definition                   |
|----------------------------------|--------------------|------------------|------------------------------|
| shift left logical immediate     | slli rd, r1, immed | slli x4, x9, 5   | x4 = x9 << 5                 |
| shift left logical               | sll rd, r1, r2     | sll x4, x9, x13  | x4 = x9 << x13               |
| shift right logical immediate    | srli rd, r1, immed | srli x4, x9, 5   | x4 = x9 >> 5                 |
| shift right logical              | srl rd, r1, r2     | srl x4, x9, x13  | x4 = x9 >> x13               |
| shift right Arithmetic immediate | srai rd, r1, immed | srai x4, x9, 5   | x4 = x9 >> 5 sign extended   |
| shift right Arithmetic           | sra rd, r1, r2     | srai x4, x9, x13 | x4 = x9 >> x13 sign extended |

### Set conditions

| name                               | format              | example           | definition                |
|------------------------------------|---------------------|-------------------|---------------------------|
| set if less then (signed)          | slt rd, r1, r2      | slt x4, x9, x13   | $x4 = (x9 < x13) ? 1 : 0$ |
| set less then immediate (signed)   | slti rd, r1, immed  | slti x4, x9, 123  | $x4 = (x9 < 123) ? 1 : 0$ |
| set if greater than (signed)       | sgt rd, r1, r2      | sgt x4, x9, x13   | $x4 = (x9 > x13) ? 1 : 0$ |
| set if less then (unsigned)        | sltu rd, r1, r2     | slty x4, x9, x13  | $x4 = (x9 < x13) ? 1 : 0$ |
| set less then immediate (unsigned) | sltiu rd, r1, immed | sltiu x4, x9, 123 | $x4 = (x9 < 123) ? 1 : 0$ |
| set if greater than (unsigned)     | sgtu rd, r1, r2     | sgtu x4, x9, x13  | $x4 = (x9 > x13) ? 1 : 0$ |
| *set if equal to zero              | seqz rd, r1         | seqz x4, x9       | $x4 = (x9 == 0) ? 1 : 0$  |
| *set if not equal to zero          | snez rd, r1         | snez x4, x9       | $x4 = (x9 != 0) ? 1 : 0$  |
| *set if less then zero             | sltz rd, r1         | sltz x4, x9       | $x4 = (x9 < 0) ? 1 : 0$   |
| *set if greater then zero          | sgtz rd, r1         | sgtz x4, x9       | $x4 = (x9 > 0) ? 1 : 0$   |