## RISC-V RV32I Base Instruction Set

[1] RISC-V RV32I Instruction Set Listing, Table 25.2, page 130 (pdf reader page 146/236)

RISC-V Instruction Set Manual, Vol I, Unprivileged ISA Document, Version 20190608-Base-Ratified, Editors: Andrew Waterman, Krste Asanovic, Jun 8, 2019

vicilogic RV32I Base Instruction Set (this document)

Interactive RISC-V RV32I Base Instruction Set Table

| ID (0x) | ID (0d) | instruction mnemonic | description | instruction(31:0) format | Type | instruction syntax |
|---|---|---|---|---|---|---|
| | | | | 31 12 11 7 6 opcode 0 | | |
| 1 | 1 | LUI | load upper immediate | imm[31:12] rd 0110111 | U | lui rd, imm |
| 2 | 2 | AUIPC | add upper immed to PC | imm[31:12] rd 0010111 | U | auipc rd, imm |
| | | **Jump and link** | | | | |
| 3 | 3 | JAL | jump and link | imm[20¦10:1¦11¦19:12] rd 1101111 | J | jal rd, imm |
| | | | | f3 | | |
| | | **Branch instructions** | | 31 25 24 20 19 15 14 12 11 7 6 0 | | |
| 4 | 4 | BEQ | branch if equal | imm[12¦10:5] rs2 rs1 000 imm[4:1¦11] 1100011 | B | beq rs1, rs2, imm |
| 5 | 5 | BNE | branch if not equal | imm[12¦10:5] rs2 rs1 001 imm[4:1¦11] 1100011 | B | bne rs1, rs2, imm |
| 6 | 6 | BLT | branch if less than (lt) | imm[12¦10:5] rs2 rs1 100 imm[4:1¦11] 1100011 | B | blt rs1, rs2, imm |
| 7 | 7 | BGE | branch if greater than (gt) | imm[12¦10:5] rs2 rs1 101 imm[4:1¦11] 1100011 | B | bge rs1, rs2, imm |
| 8 | 8 | BLTU | branch if lt than or eq unsgn | imm[12¦10:5] rs2 rs1 110 imm[4:1¦11] 1100011 | B | bltu rs1, rs2, imm |
| 9 | 9 | BGEU | branch if gt than unsgn | imm[12¦10:5] rs2 rs1 111 imm[4:1¦11] 1100011 | B | bgeu rs1, rs2, imm |
| | | **Jump and link (registered)** | | 31 20 19 15 14 12 11 7 6 0 | | |
| 0xA | 10 | JALR | jump & link register | imm[11:0] rs1 000 rd 1100111 | I | jalr rd, imm(rs1) |
| | | **Memory Load Instructions** | | | | |
| 0xB | 11 | LB | load byte | imm[11:0] rs1 000 rd 0000011 | I | lb rd, imm(rs1) |
| 0xC | 12 | LH | load halfword | imm[11:0] rs1 001 rd 0000011 | I | lw rd, imm(rs1) |
| 0xD | 13 | LW | load word | imm[11:0] rs1 010 rd 0000011 | I | lw rd, imm(rs1) |
| 0xE | 14 | LBU | load byte unsigned | imm[11:0] rs1 100 rd 0000011 | I | lbu rd, imm(rs1) |
| 0xF | 15 | LHU | load halfword unsigned | imm[11:0] rs1 101 rd 0000011 | I | lhu rd, imm(rs1) |
| | | **Integer register-immediate instructions** | | | | |
| 0x10 | 16 | ADDI | add immediate | imm[11:0] rs1 000 rd 0010011 | I | addi rd, rs1, imm |
| 0x11 | 17 | SLTI | set less than immediate | imm[11:0] rs1 010 rd 0010011 | I | slti rd, rs1, imm |
| 0x12 | 18 | SLTIU | set less than immed unsgn | imm[11:0] rs1 011 rd 0010011 | I | sltiu rd, rs1, imm |
| 0x13 | 19 | XORI | xor immediate | imm[11:0] rs1 100 rd 0010011 | I | xori rd, rs1, imm |
| 0x14 | 20 | ORI | or immediate | imm[11:0] rs1 110 rd 0010011 | I | ori rd, rs1, imm |
| 0x15 | 21 | ANDI | and immediate | imm[11:0] rs1 111 rd 0010011 | I | andi rd, rs1, imm |
| | | **Memory Store Instructions** | | 31 25 24 20 19 15 14 12 11 7 6 0 | | |
| 0x16 | 22 | SB | store byte | imm[11:5] rs2 rs1 000 imm[4:0] 0100011 | S | sb rs2, imm(rs1) |
| 0x17 | 23 | SH | store halfword | imm[11:5] rs2 rs1 001 imm[4:0] 0100011 | S | sh rs2, imm(rs1) |
| 0x18 | 24 | SW | store word | imm[11:5] rs2 rs1 010 imm[4:0] 0100011 | S | sw rs2, imm(rs1) |
| | | | | f7 | | |
| | | **Constant shift Instructions** | | | | |
| 0x19 | 25 | SLLI | constant-shift left | 0000000 shamt rs1 001 rd 0010011 | I | slli rd, rs1, shamt |
| 0x1A | 26 | SRLI | constant-shift right | 0000000 shamt rs1 101 rd 0010011 | I | srli rd, rs1, shamt |
| 0x1B | 27 | SRAI | constant-shift right arithmetic | 0100000 shamt rs1 101 rd 0010011 | I | srai rd, rs1, shamt |
| | | **Integer register-register instructions** | | | | |
| 0x1C | 28 | ADD | add | 0000000 rs2 rs1 000 rd 0110011 | R | add rd, rs1, rs2 |
| 0x1D | 29 | SUB | subtract | 0100000 rs2 rs1 000 rd 0110011 | R | sub rd, rs1, rs2 |
| 0x1E | 30 | SLL | register-shift left | 0000000 rs2 rs1 001 rd 0110011 | R | sll rd, rs1, rs2 |
| 0x1F | 31 | SLT | set less than | 0000000 rs2 rs1 010 rd 0110011 | R | slt rd, rs1, rs2 |
| 0x20 | 32 | SLTU | set less than unsigned | 0000000 rs2 rs1 011 rd 0110011 | R | sltu rd, rs1, rs2 |
| 0x21 | 33 | XOR | xor | 0000000 rs2 rs1 100 rd 0110011 | R | xor rd, rs1, rs2 |
| 0x22 | 34 | SRL | register-shift right (logical) | 0000000 rs2 rs1 101 rd 0110011 | R | srl rd, rs1, rs2 |
| 0x23 | 35 | SRA | register-shift right (arithmetic) | 0100000 rs2 rs1 101 rd 0110011 | R | sra rd, rs1, rs2 |
| 0x24 | 36 | OR | or | 0000000 rs2 rs1 110 rd 0110011 | R | or rd, rs1, rs2 |
| 0x25 | 37 | AND | and | 0000000 rs2 rs1 111 rd 0110011 | R | and rd, rs1, rs2 |
| | | | | 31 20 19 15 14 12 11 7 6 0 | | |
| 0x26 | 38 | FENCE | | fm pred succ rs1 000 rd 0001111 | | |
| 0x27 | 39 | ECALL | | 000000000000 00000 000 00000 1110011 | | |
| 0x28 | 40 | EBREAK | | 000000000001 00000 000 00000 1110011 | | |