| Instruction type/opcode | Instruction meaning |
|---|---|
| *Data transfers* | *Move data between registers and memory, or between the integer and FP or special registers; only memory address mode is 12-bit displacement+contents of a GPR* |
| lb, lbu, sb | Load byte, load byte unsigned, store byte (to/from integer registers) |
| lh, lhu, sh | Load half word, load half word unsigned, store half word (to/from integer registers) |
| lw, lwu, sw | Load word, load word unsigned, store word (to/from integer registers) |
| ld, sd | Load double word, store double word (to/from integer registers) |
| flw, fld, fsw, fsd | Load SP float, load DP float, store SP float, store DP float |
| fmv._.x, fmv.x._ | Copy from/to integer register to/from floating-point register; "_"=S for single-precision, D for double-precision |
| csrrw, csrrwi, csrrs, csrrsi, csrrc, csrrci | Read counters and write status registers, which include counters: clock cycles, time, instructions retired |
| *Arithmetic/logical* | *Operations on integer or logical data in GPRs* |
| add, addi, addw, addiw | Add, add immediate (all immediates are 12 bits), add 32-bits only & sign-extend to 64 bits, add immediate 32-bits only |
| sub, subw | Subtract, subtract 32-bits only |
| mul, mulw, mulh, mulhsu, mulhu | Multiply, multiply 32-bits only, multiply upper half, multiply upper half signed-unsigned, multiply upper half unsigned |
| div, divu, rem, remu | Divide, divide unsigned, remainder, remainder unsigned |
| divw, divuw, remw, remuw | Divide and remainder: as previously, but divide only lower 32-bits, producing 32-bit sign-extended result |
| and, andi | And, and immediate |
| or, ori, xor, xori | Or, or immediate, exclusive or, exclusive or immediate |
| lui | Load upper immediate; loads bits 31-12 of register with immediate, then sign-extends |
| auipc | Adds immediate in bits 31–12 with zeros in lower bits to PC; used with JALR to transfer control to any 32-bit address |
| sll, slli, srl, srli, sra, srai | Shifts: shift left logical, right logical, right arithmetic; both variable and immediate forms |
| sllw, slliw, srlw, srliw, sraw, sraiw | Shifts: as previously, but shift lower 32-bits, producing 32-bit sign-extended result |
| slt, slti, sltu, sltiu | Set less than, set less than immediate, signed and unsigned |
| *Control* | *Conditional branches and jumps; PC-relative or through register* |
| beq, bne, blt, bge, bltu, bgeu | Branch GPR equal/not equal; less than; greater than or equal, signed and unsigned |
| jal, jalr | Jump and link: save PC+4, target is PC-relative (JAL) or a register (JALR); if specify x0 as destination register, then acts as a simple jump |
| ecall | Make a request to the supporting execution environment, which is usually an OS |
| ebreak | Debuggers used to cause control to be transferred back to a debugging environment |
| fence, fence.i | Synchronize threads to guarantee ordering of memory accesses; synchronize instructions and data for stores to instruction memory |

**Figure 1.5 Subset of the instructions in RISC-V.** RISC-V has a base set of instructions (R64I) and offers optional extensions: multiply-divide (RVM), single-precision floating point (RVF), double-precision floating point (RVD). This figure includes RVM and the next one shows RVF and RVD. Appendix A gives much more detail on RISC-V.