RocketCPU - RV64GC Instruction Wall Chart (Mnemonic-level quick reference)

v1.0 (spec-aligned)

Scope: RV64I + M + A + F + D + C. This is a mnemonic grouping (not full encoding tables).

Use official spec for exact funct3/funct7 encodings.

RV64I Base Integer

Upper/PC: lui, auipc
Jumps/Calls: jal, jalr
Branches: beq, bne, blt, bge, bltu, bgeu
Loads: lb, lh, lw, lbu, lhu, lwu, ld
Stores: sb, sh, sw, sd
Arithmetic/Logic (reg): add, sub, sll, slt, sltu, xor, srl, sra, or, and
Arithmetic/Logic (imm): addi, slti, sltiu, xori, ori, andi, slli, srli, srai
64-bit ops: addw, subw, sllw, srlw, sraw; addiw, slliw, srliw, sraiw
Fences & system: fence, fence.i, ecall, ebreak
CSRs: csrrw, csrrs, csrrc, csrrwi, csrrsi, csrrci

M Extension (Integer Mul/Div)

Mul/div (64): mul, mulh, mulhsu, mulhu, div, divu, rem, remu
Mul/div (32W): mulw, divw, divuw, remw, remuw

A Extension (Atomics)

Atomic (LR/SC): lr.w, sc.w, lr.d, sc.d
AMOs (W/D): amoadd, amoswap, amoxor, amoand, amoor, amomin
amomax, amominu, amomaxu
Notes: AMOs are read-modify-write; use fences for ordering where required

F Extension (Single-Precision FP)

Moves/convert: fmv.x.w, fmv.w.x, fcvt.*.s / fcvt.s.* (int <-> single)
Arithmetic: fadd.s, fsub.s, fmul.s, fdiv.s, fsqrt.s
FMA: fmadd.s, fmsub.s, fnmadd.s, fnmsub.s
Compare/minmax: fmin.s, fmax.s, feq.s, flt.s, fle.s
Sign/class: fsgnj.s, fsgnjn.s, fsgnjx.s, fclass.s
CSR: fcsr, frm, fflags (via CSRs)

D Extension (Double-Precision FP)

Moves/convert: fmv.x.d, fmv.d.x, fcvt.*.d / fcvt.d.* (int <-> double)
Arithmetic: fadd.d, fsub.d, fmul.d, fdiv.d, fsqrt.d
FMA: fmadd.d, fmsub.d, fnmadd.d, fnmsub.d
Compare/minmax: fmin.d, fmax.d, feq.d, flt.d, fle.d
Sign/class: fsgnj.d, fsgnjn.d, fsgnjx.d, fclass.d

C Extension (Compressed)

Compressed 16-bit encodings map to common ops:
Loads/stores: c.lw, c.sw, c.ld, c.sd, c.lwsp, c.swsp, c.ldsp, c.sdsp
ALU/imm: c.addi, c.addiw, c.li, c.lui, c.andi, c.srli, c.srai, c.slli
Reg-reg: c.add, c.mv, c.and, c.or, c.xor, c.sub, c.addw, c.subw
Control: c.j, c.jal (RV32), c.jr, c.jalr, c.beqz, c.bnez
Misc: c.nop, c.ebreak

Notes: (1) 'w' suffix operates on low 32 bits and sign-extends to 64. (2) RV64GC = RV64I + M + A + F + D + C.

RocketCPU: generator-driven cores; see docs/GENERATION.md for Chisel→Verilog flow.