Instruction		Description	Operation	Type	funct7 (31–25)		funct3 (14–12)		opcode (6–0)
lui	rd,imm	Load Upper Immediate	rd ← imm_u, pc ← pc+4	U	imm[31:12]			rd	0 1 1 0 1 1 1
auipc	rd,imm	Add Upper Immediate to PC	rd ← pc + imm_u, pc ← pc+4	U	imm[31:12]			rd	0 0 1 0 1 1 1
jal	rd,pcrel_21	Jump And Link	rd ← pc+4, pc ← pc+imm_j	J	imm[20|10:1|11|19:12]			rd	1 1 0 1 1 1 1
jalr	rd,imm(rs1)	Jump And Link Register	rd ← pc+4, pc ← (rs1+imm_i) ∧ ∼1	I	imm[11:0]		0 0 0	rd	1 1 0 0 1 1 1
beq	rs1,rs2,pcrel_13	Branch Equal	pc ← pc + ((rs1==rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	0 0 0	imm[4:1|11]	1 1 0 0 0 1 1
bne	rs1,rs2,pcrel_13	Branch Not Equal	pc ← pc + ((rs1!=rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	0 0 1	imm[4:1|11]	1 1 0 0 0 1 1
blt	rs1,rs2,pcrel_13	Branch Less Than	pc ← pc + ((rs1<rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	1 0 0	imm[4:1|11]	1 1 0 0 0 1 1
bge	rs1,rs2,pcrel_13	Branch Greater or Equal	pc ← pc + ((rs1>=rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	1 0 1	imm[4:1|11]	1 1 0 0 0 1 1
bltu	rs1,rs2,pcrel_13	Branch Less Than Unsigned	pc ← pc + ((rs1<rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	1 1 0	imm[4:1|11]	1 1 0 0 0 1 1
bgeu	rs1,rs2,pcrel_13	Branch Greater or Equal Unsigned	pc ← pc + ((rs1>=rs2) ? imm_b : 4)	B	imm[12|10:5]	rs2	1 1 1	imm[4:1|11]	1 1 0 0 0 1 1
lb	rd,imm(rs1)	Load Byte	rd ← sx(m8(rs1+imm_i)), pc ← pc+4	I	imm[11:0]		0 0 0	rd	0 0 0 0 0 1 1
lh	rd,imm(rs1)	Load Halfword	rd ← sx(m16(rs1+imm_i)), pc ← pc+4	I	imm[11:0]		0 0 1	rd	0 0 0 0 0 1 1
lw	rd,imm(rs1)	Load Word	rd ← sx(m32(rs1+imm_i)), pc ← pc+4	I	imm[11:0]		0 1 0	rd	0 0 0 0 0 1 1
lbu	rd,imm(rs1)	Load Byte Unsigned	rd ← zx(m8(rs1+imm_i)), pc ← pc+4	I	imm[11:0]		1 0 0	rd	0 0 0 0 0 1 1
lhu	rd,imm(rs1)	Load Halfword Unsigned	rd ← zx(m16(rs1+imm_i)), pc ← pc+4	I	imm[11:0]		1 0 1	rd	0 0 0 0 0 1 1
sb	rs2,imm(rs1)	Store Byte	m8(rs1+imm_s) ← rs2[7:0], pc ← pc+4	S	imm[11:5]	rs2	0 0 0	imm[4:0]	0 1 0 0 0 1 1
sh	rs2,imm(rs1)	Store Halfword	m16(rs1+imm_s) ← rs2[15:0], pc ← pc+4	S	imm[11:5]	rs2	0 0 1	imm[4:0]	0 1 0 0 0 1 1
sw	rs2,imm(rs1)	Store Word	m32(rs1+imm_s) ← rs2[31:0], pc ← pc+4	S	imm[11:5]	rs2	0 1 0	imm[4:0]	0 1 0 0 0 1 1
addi	rd,rs1,imm	Add Immediate	rd ← rs1 + imm_i, pc ← pc+4	I	imm[11:0]		0 0 0	rd	0 0 1 0 0 1 1
slti	rd,rs1,imm	Set Less Than Immediate	rd ← (rs1 < imm_i) ? 1 : 0, pc ← pc+4	I	imm[11:0]		0 1 0	rd	0 0 1 0 0 1 1
sltiu	rd,rs1,imm	Set Less Than Immediate Unsigned	rd ← (rs1 < imm_i) ? 1 : 0, pc ← pc+4	I	imm[11:0]		0 1 1	rd	0 0 1 0 0 1 1
xori	rd,rs1,imm	Exclusive Or Immediate	rd ← rs1 ⊕ imm_i, pc ← pc+4	I	imm[11:0]		1 0 0	rd	0 0 1 0 0 1 1
ori	rd,rs1,imm	Or Immediate	rd ← rs1 ∨ imm_i, pc ← pc+4	I	imm[11:0]		1 1 0	rd	0 0 1 0 0 1 1
andi	rd,rs1,imm	And Immediate	rd ← rs1 ∧ imm_i, pc ← pc+4	I	imm[11:0]		1 1 1	rd	0 0 1 0 0 1 1
slli	rd,rs1,shamt	Shift Left Logical Immediate	rd ← rs1 << shamt_i, pc ← pc+4	I	0 0 0 0 0 0 0	shamt	0 0 1	rd	0 0 1 0 0 1 1
srli	rd,rs1,shamt	Shift Right Logical Immediate	rd ← rs1 >> shamt_i, pc ← pc+4	I	0 0 0 0 0 0 0	shamt	1 0 1	rd	0 0 1 0 0 1 1
srai	rd,rs1,shamt	Shift Right Arithmetic Immediate	rd ← rs1 >> shamt_i, pc ← pc+4	I	0 1 0 0 0 0 0	shamt	1 0 1	rd	0 0 1 0 0 1 1
add	rd,rs1,rs2	Add	rd ← rs1 + rs2, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	0 0 0	rd	0 1 1 0 0 1 1
sub	rd,rs1,rs2	Subtract	rd ← rs1 - rs2, pc ← pc+4	R	0 1 0 0 0 0 0	rs2	0 0 0	rd	0 1 1 0 0 1 1
sll	rd,rs1,rs2	Shift Left Logical	rd ← rs1 << (rs2%XLEN), pc ← pc+4	R	0 0 0 0 0 0 0	rs2	0 0 1	rd	0 1 1 0 0 1 1
slt	rd,rs1,rs2	Set Less Than	rd ← (rs1 < rs2) ? 1 : 0, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	0 1 0	rd	0 1 1 0 0 1 1
sltu	rd,rs1,rs2	Set Less Than Unsigned	rd ← (rs1 < rs2) ? 1 : 0, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	0 1 1	rd	0 1 1 0 0 1 1
xor	rd,rs1,rs2	Exclusive Or	rd ← rs1 ⊕ rs2, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	1 0 0	rd	0 1 1 0 0 1 1
srl	rd,rs1,rs2	Shift Right Logical	rd ← rs1 >> (rs2%XLEN), pc ← pc+4	R	0 0 0 0 0 0 0	rs2	1 0 1	rd	0 1 1 0 0 1 1
sra	rd,rs1,rs2	Shift Right Arithmetic	rd ← rs1 >> (rs2%XLEN), pc ← pc+4	R	0 1 0 0 0 0 0	rs2	1 0 1	rd	0 1 1 0 0 1 1
or	rd,rs1,rs2	Or	rd ← rs1 ∨ rs2, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	1 1 0	rd	0 1 1 0 0 1 1
and	rd,rs1,rs2	And	rd ← rs1 ∧ rs2, pc ← pc+4	R	0 0 0 0 0 0 0	rs2	1 1 1	rd	0 1 1 0 0 1 1