Instruction types:

Not including system instructions for multicore processors

Branch Instructions: if(reg[rs1] OP reg[rs2]) then pc = pc + offset

Beq, if(reg[rs1] == reg[rs2]) then pc = pc + sext(offset)

Bne, if(reg[rs1] != reg[rs2]) then pc = pc + sext(offset)

Blt, if(reg[rs1] < signed reg[rs2]) then pc = pc +sext(offset)

Bge, if(reg[rs1] >= signed reg[rs2]) then pc = pc + sext(offset)

Bltu, if(reg[rs1] < unsigned reg[rs2]) then pc = pc + sext(offset)

Bgeu, if(reg[rs1] >= unsigned reg[rs2]) then pc = pc + sext(offset)

ALU register instructions: reg[rd] = reg[rs1] OP reg[rs2]

Add, reg[rd] = reg[rs1] + reg[rs2]

Sub, reg[rd] = reg[rs1] – reg[rs2]

Sll, reg[rd] = reg[rs1] << reg[rs2][4:0]

Slt, reg[rd] = reg[rs1] < signed reg[rs2]

Sltu, reg[rd] = reg[rs1] < unsigned reg[rs2]

Xor, reg[rd] = reg[rs1] ^ reg[rs2]

Srl, reg[rd] = reg[rs1] >> reg[rs2][4:0]

Sra, reg[rd] = reg[rs1] >>> reg[rs2][4:0]

Or, reg[rd] = reg[rs1] | reg[rs2]

And, reg[rd] = reg[rs1] & reg[rs2]

ALU immediate instructions: reg[rd] = reg[rs1] OP immediate

Addi, reg[rd] = reg[rs1] + sext(immediate)

Slti, reg[rd] = reg[rs1] < signed sext(immediate)

Sltiu, reg[rd] = reg[rs1] < unsigned sext(immediate)

Xori, reg[rd] = reg[rs1] ^ sext(immediate)

Ori, reg[rd] = reg[rs1] | sext(immediate)

Andi, reg[rd] = reg[rs1] & sext(immediate)

Slli, reg[rd] = reg[rs1] << immediate

Srli, reg[rd] = reg[rs1] >> immediate

Srai, reg[rd] = reg[rs1] >>> immediate

Load instructions: reg[rd] = memory[reg[rs1] + immediate]

Lb, reg[rd] = sext( memory[ reg[rs1] + sext(offset) ][7:0] )

Lh, reg[rd] = sext( memory[ reg[rs1] + sext(offset) ][15:0] )

Lw, reg[rd] = sext( memory[ reg[rs1] + sext(offset) ][31:0] )

Lbu, reg[rd] = memory[ reg[rs1] + sext(offset) ]

Lhu, reg[rd] = memory[ reg[rs1] + sext(offset) ][15:0]

Store instructions: memory[reg[rs1] + immediate] = reg[rs2]

Sb, memory[ reg[rs1] + sext(offset) ] = x[rs2][7:0]

Sh, memory[ reg[rs1] + sext(offset) ] = x[rs2][15:0]

Sw, memory[ reg[rs1] + sext(offset) ] = x[rs2][31:0]

Other Instructions:

LUI, reg[rd] = sext(immediate[31:12] << 12)

AUIPC, reg[rd] = pc + sext(immediate[31:12] << 12),

JAL, reg[rd] = pc + 4; pc = pc + sext(offset)

JALR, reg[rd] = pc + 4; pc = (reg[rs1] + sext(offset))