Instruction Encoding Table(RV32I)

rv32i_cpu by 王靖尧，吴长轩，王泽锴

Instruction	Type	Opcode (bin)	Funct3	Funct7	Verilog Operation / Description
LUI	U	`0110111`	-	-	`rd = imm << 12`
AUIPC	U	`0010111`	-	-	`rd = PC + (imm << 12)`
JAL	J	`1101111`	-	-	`rd = PC+4; PC += imm`
JALR	I	`1100111`	`000`	-	`rd = PC+4; PC = (rs1+imm)&~1`
BEQ	B	`1100011`	`000`	-	`if(rs1==rs2) PC += imm`
BNE	B	`1100011`	`001`	-	`if(rs1!=rs2) PC += imm`
BLT	B	`1100011`	`100`	-	`if(rs1<rs2) PC += imm` (signed)
BGE	B	`1100011`	`101`	-	`if(rs1>=rs2) PC += imm` (signed)
LB	I	`0000011`	`000`	-	Load Byte
LH	I	`0000011`	`001`	-	Load Half
LW	I	`0000011`	`010`	-	Load Word
SB	S	`0100011`	`000`	-	Store Byte
SH	S	`0100011`	`001`	-	Store Half
SW	S	`0100011`	`010`	-	Store Word
ADDI	I	`0010011`	`000`	-	`rd = rs1 + imm`
SLTI	I	`0010011`	`010`	-	Set Less Than Imm
ANDI	I	`0010011`	`111`	-	`rd = rs1 & imm`

Instruction	Type	Opcode (bin)	Funct3	Funct7	Verilog Operation / Description
ORI	I	`0010011`	`110`	-	`rd = rs1 | imm`
XORI	I	`0010011`	`100`	-	`rd = rs1 ^ imm`
SLLI	I	`0010011`	`001`	`0000000`	`rd = rs1 << shamt`
SRLI	I	`0010011`	`101`	`0000000`	`rd = rs1 >> shamt` (logical)
SRAI	I	`0010011`	`101`	`0100000`	`rd = rs1 >>> shamt` (arithmetic)
ADD	R	`0110011`	`000`	`0000000`	`rd = rs1 + rs2`
SUB	R	`0110011`	`000`	`0100000`	`rd = rs1 - rs2`
SLL	R	`0110011`	`001`	`0000000`	`rd = rs1 << rs2`
SLT	R	`0110011`	`010`	`0000000`	Set Less Than
XOR	R	`0110011`	`100`	`0000000`	`rd = rs1 ^ rs2`
SRL	R	`0110011`	`101`	`0000000`	`rd = rs1 >> rs2`
SRA	R	`0110011`	`101`	`0100000`	`rd = rs1 >>> rs2`
OR	R	`0110011`	`110`	`0000000`	`rd = rs1 | rs2`
AND	R	`0110011`	`111`	`0000000`	`rd = rs1 & rs2`