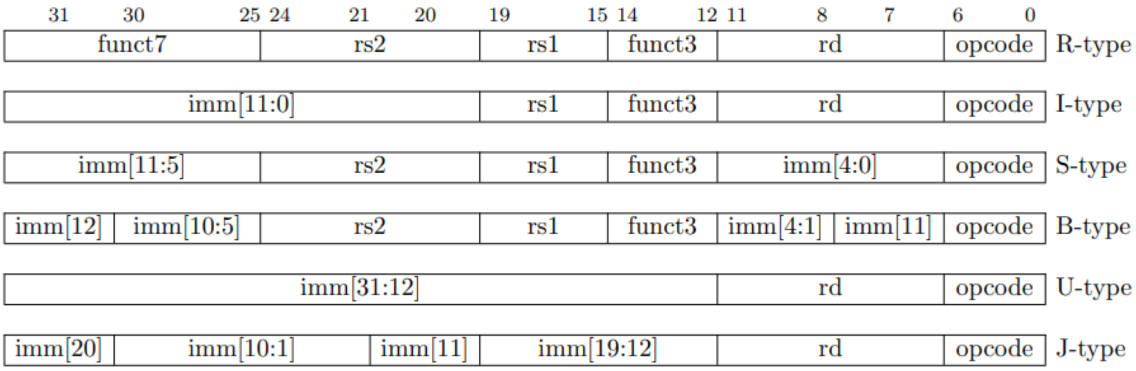
RV32I ISA



31	30 25	24 21	20	19 15	14 12	11 8	7	6 0	
funct7		rs2		rs1	funct3	rd		opcode	R-type
imm[11:0]				rs1	funct3	rd		opcode	I-type
imm[11:5]		rs2		rs1	funct3	imm[4:0]		opcode	S-type
imm[12]	imm[10:5]	rs2		rs1	funct3	imm[4:1]	imm[11]	opcode	B-type
imm[31:12]						rd		opcode	U-type
imm[20]	imm[10:1]		imm[11]	imm[19:12]		rd		opcode	J-type

Opcode is of 7 bits, and it is used by the instruction decoder to find out the exact instruction.

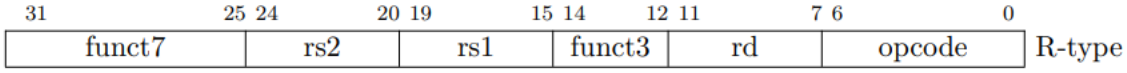
Rs1, Rs2 and Rd are 5bits each as the core has a total of 32 registers with bit width of 32.

Func3 is 3 bits and func7 is 7bits also used along with the opcode to decode the instruction type.

Imm bit width varies between different instruction formats as shown in the above figure.

Supported Instructions

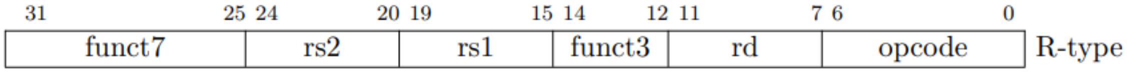
1. **ADD instruction:**
   This is a R-Type instruction, and its instruction format is as below.



31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

   Opcode = 0110011
   Func3 = 000
   Func7 = 0000000
   Rd = Rs1 + Rs2

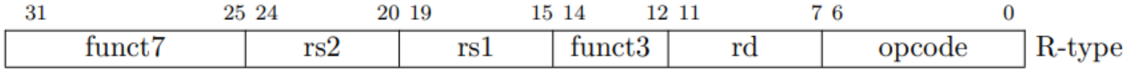
2. **SUB instruction:**
   This is a R-Type instruction, and its instruction format is as below.



31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

   Opcode = 0110011
   Func3 = 000
   Func7 = 0100000
   Rd = Rs1 - Rs2

3. **XOR instruction:**
   This is a R-Type instruction, and its instruction format is as below.



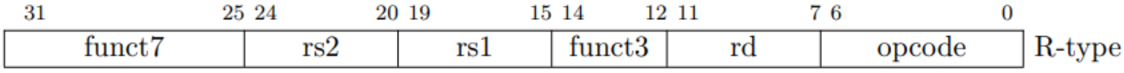
31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

   Opcode = 0110011
   Func3 = 100
   Func7 = 0000000

Rd = Rs1 ^ Rs2

4. **OR instruction:**

This is a R-Type instruction, and its instruction format is as below.

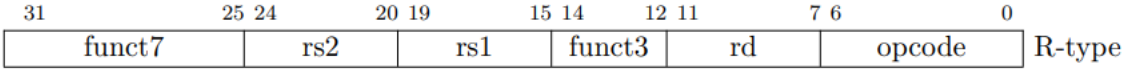


31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 110
Func7 = 0000000
Rd = Rs1 | Rs2

5. **AND instruction:**

This is a R-Type instruction, and its instruction format is as below.

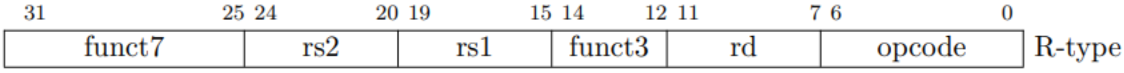


31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 111
Func7 = 0000000
Rd = Rs1 & Rs2

6. **SLL (Shift Left Logical) instruction:**

This is a R-Type instruction, and its instruction format is as below.

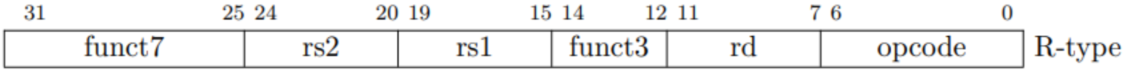


31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 001
Func7 = 0000000
Rd = Rs1 << Rs2

7. **SRL (Shift Right Logical) instruction:**

This is a R-Type instruction, and its instruction format is as below.

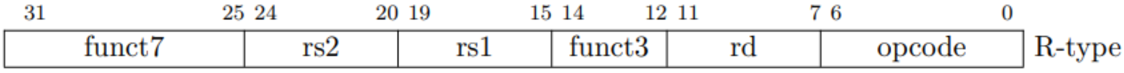


31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 101
Func7 = 0000000
Rd = Rs1 >> Rs2

8. **SRA (Shift Right Arith) instruction:**

This is a R-Type instruction, and its instruction format is as below.



31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 101
Func7 = 0100000
Rd = Rs1 >> Rs2 (MSB Extended)

9. **SLT (Set Less Than) instruction:**

This is a R-Type instruction, and its instruction format is as below.

31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 010
Func7 = 0000000
Rd = (Rs1 < Rs2) ? 1 : 0

10. SLTU (Set Less Than (U)) instruction:

This is a R-Type instruction, and its instruction format is as below.

31 25	24 20	19 15	14 12	11 7	6 0	
funct7	rs2	rs1	funct3	rd	opcode	R-type

Opcode = 0110011
Func3 = 011
Func7 = 0000000
Rd = (Rs1 < Rs2) ? 1 : 0 (Zero Extended)

11. ADDI instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 000
Imm [11:0] = immediate value to be added
Rd = Rs1 + imm

12. XORI instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 100
Imm [11:0] = immediate value to be xored
Rd = Rs1 ^ imm

13. ORI instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 110
Imm [11:0] = immediate value to be ored
Rd = Rs1 | imm

14. ANDI instruction:

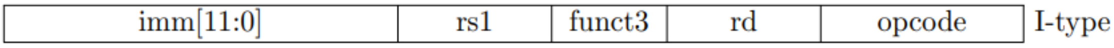
This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 111
Imm [11:0] = immediate value to be Anded
Rd = Rs1 & imm

15. SLLI (Shift Left Logical Immediate) instruction:

This is a I-Type instruction, and its instruction format is as below.

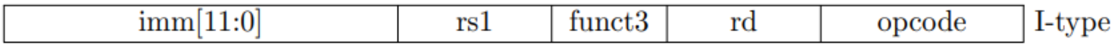


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 001
Imm [11:5] = 0000000
Rd = Rs1 << imm[4:0]

16. SRLI (Shift Right Logical Immediate) instruction:

This is a I-Type instruction, and its instruction format is as below.

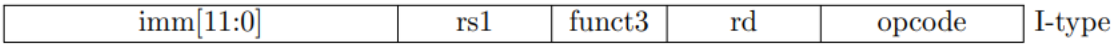


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 101
Imm [11:5] = 0000000
Rd = Rs1 >> imm[4:0]

17. SRAI (Shift Right Arith Immediate) instruction:

This is a I-Type instruction, and its instruction format is as below.

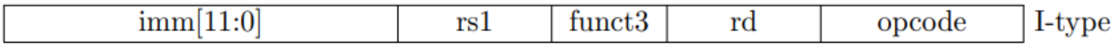


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 101
Imm [11:5] = 0100000
Rd = Rs1 >> imm[4:0] (MSB Extended)

18. SLTI (Set less Than Immediate) instruction:

This is a I-Type instruction, and its instruction format is as below.

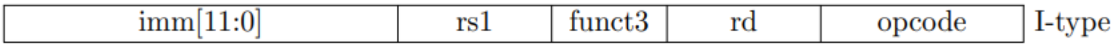


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 010
Imm [11:0] = Immediate value to be compared
Rd = (Rs1 < imm) ? 1 : 0

19. SLTIU (Set less Than Immediate (U)) instruction:

This is a I-Type instruction, and its instruction format is as below.

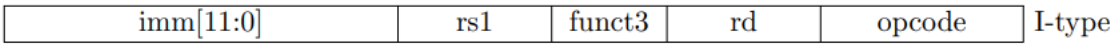


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0010011
Func3 = 011
Imm [11:0] = Immediate value to be compared
Rd = (Rs1 < imm) ? 1 : 0 (Zero Extended)

20. LB (Load Byte) instruction:

This is a I-Type instruction, and its instruction format is as below.

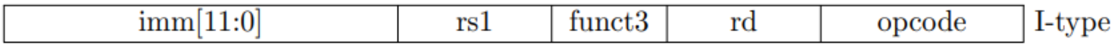


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0000011
Func3 = 000
Rd = M[Rs1+imm][7:0]

21. LH (Load Half) instruction:

This is a I-Type instruction, and its instruction format is as below.



imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0000011

Func3 = 001
Rd = M[Rs1+imm][15:0]

22. LB (Load Word) instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0000011
Func3 = 010
Rd = M[Rs1+imm][31:0]

23. LBU (Load Byte (U)) instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0000011
Func3 = 100
Rd = M[Rs1+imm][7:0] (Zero Extended)

24. LHU (Load Half (U)) instruction:

This is a I-Type instruction, and its instruction format is as below.

imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 0000011
Func3 = 101
Rd = M[Rs1+imm][15:0]

25. SB (Store Byte) instruction:

This is a S-Type instruction, and its instruction format is as below.

imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type

Opcode = 0100011
Func3 = 000
M[Rs1+imm][7:0] = Rs2[7:0]

26. SH (Store half) instruction:

This is a S-Type instruction, and its instruction format is as below.

imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type

Opcode = 0100011
Func3 = 001
M[Rs1+imm][15:0] = Rs2[15:0]

27. SW (Store Word) instruction:

This is a S-Type instruction, and its instruction format is as below.

imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode	S-type

Opcode = 0100011
Func3 = 010
M[Rs1+imm][31:0] = Rs2[31:0]

28. BEQ (Branch ==) instruction:

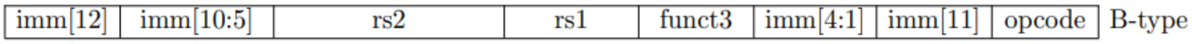
This is a B-Type instruction, and its instruction format is as below.

imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 000
If (Rs1 == Rs2) PC = PC + {imm[12:1], 1’b0};

29. BNE (Branch !=) instruction:

This is a B-Type instruction, and its instruction format is as below.

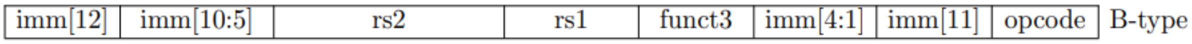


imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 001
If (Rs1 != Rs2) PC = PC + {imm[12:1], 1’b0};

30. BLT (Branch <) instruction:

This is a B-Type instruction, and its instruction format is as below.

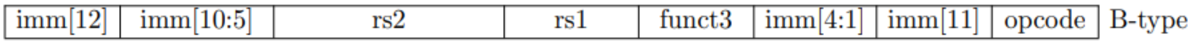


imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 100
If (Rs1 < Rs2) PC = PC + {imm[12:1], 1’b0};

31. BGE (Branch >=) instruction:

This is a B-Type instruction, and its instruction format is as below.

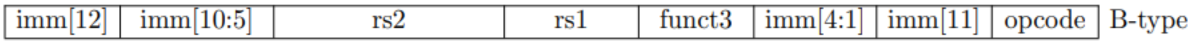


imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 101
If (Rs1 >= Rs2) PC = PC + {imm[12:1], 1’b0};

32. BLTU (Branch <(U)) instruction:

This is a B-Type instruction, and its instruction format is as below.

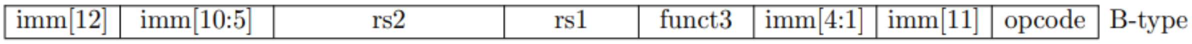


imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 110
If (Rs1 < Rs2) PC = PC + {imm[12:1], 1’b0}; (Zero Extended)

33. BGEU (Branch >=(U)) instruction:

This is a B-Type instruction, and its instruction format is as below.

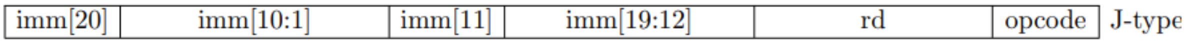


imm[12]	imm[10:5]	rs2	rs1	funct3	imm[4:1]	imm[11]	opcode	B-type

Opcode = 1100011
Func3 = 111
If (Rs1 >= Rs2) PC = PC + {imm[12:1], 1’b0}; (Zero Extended)

34. JAL (Jump and Link) instruction:

This is a J-Type instruction, and its instruction format is as below.

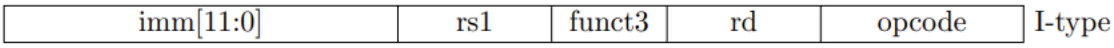


imm[20]	imm[10:1]	imm[11]	imm[19:12]	rd	opcode	J-type

Opcode = 1101111
PC = PC + {imm[20:1], 1’b0};
Rd = PC + 4

35. JALR (Jump and Link Reg) instruction:

This is a I-Type instruction, and its instruction format is as below.

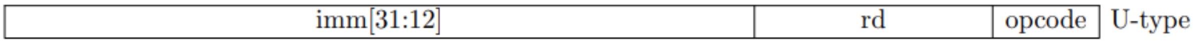


imm[11:0]	rs1	funct3	rd	opcode	I-type

Opcode = 1100111
Func3 = 000
PC = Rs1 + {imm[12:1], 1’b0};
Rd = PC + 4

36. LUI (Load Upper Immediate) instruction:

This is a U-Type instruction, and its instruction format is as below.

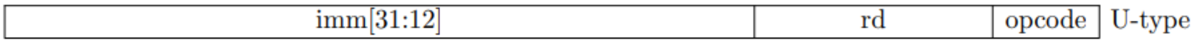


imm[31:12]	rd	opcode	U-type

Opcode = 0110111
Rd = {imm[31:12]}, 12’b0}

37. AUIPC (Add Upper Immediate to PC) instruction:

This is a U-Type instruction, and its instruction format is as below.



imm[31:12]	rd	opcode	U-type

Opcode = 0010111
Rd = PC + {imm[31:12]}, 12’b0}