# RISC-V ISA Overview and Instruction Formats

## RISC-V ISA Overview

The RISC-V Instruction Set Architecture (ISA) consists of a base integer ISA, which must be present in all implementations, and optional extensions to expand functionality. The base integer ISA is minimalist, similar to early RISC processors, but it lacks branch delay slots and supports optional variable-length instruction encodings.

## RISC-V Instruction Formats

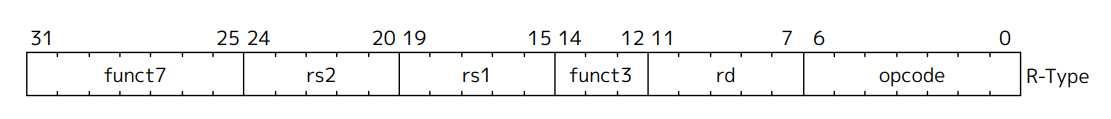
RISC-V instructions are organized into five main types based on how operands and data are accessed and processed:

### 1. R-Type (Register-Register)

Used for operations involving two source registers and one destination register, mainly for arithmetic and logical operations.

Fields:

• opcode (7 bits): Defines the operation (e.g., integer or floating-point arithmetic).  
• funct3 (3 bits): Adds specificity (e.g., add vs. sub).  
• funct7 (7 bits): Provides additional operation control.  
• rs1 and rs2 (5 bits each): Source registers.  
• rd (5 bits): Destination register.

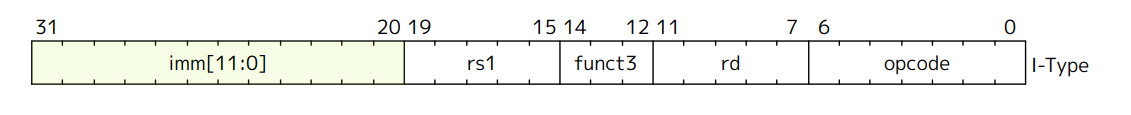


### 2. I-Type (Immediate)

Used for operations with one register and an immediate (constant) value, commonly for load instructions and arithmetic with a constant.

Fields:

• opcode (7 bits): Specifies the operation (e.g., load or add-immediate).  
• funct3 (3 bits): Specifies the operation within the immediate category.  
• rs1 (5 bits): Source register.  
• rd (5 bits): Destination register.  
• Immediate (12 bits): Signed or zero-extended constant value.

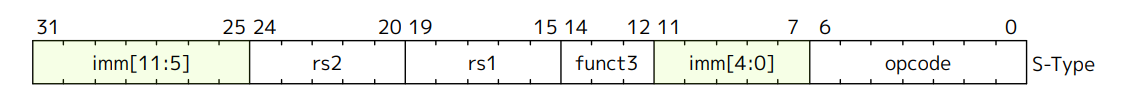


### 3. S-Type (Store)

Used for store operations, where data from a register is written to memory. The memory address is derived by adding an immediate offset to a base address.

Fields:

• opcode (7 bits): Specifies the store operation (e.g., store word).  
• funct3 (3 bits): Defines store operation type (e.g., word, byte).  
• rs1 (5 bits): Base address register.  
• rs2 (5 bits): Source register with data to store.  
• Immediate (12 bits): Offset for memory address calculation.

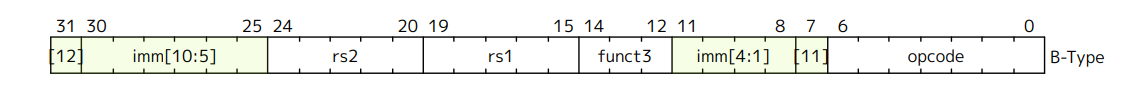


### 4. B-Type (Branch)

Enables conditional branching based on comparisons between two registers.

Fields:

• opcode (7 bits): Indicates branch operation.  
• funct3 (3 bits): Defines comparison type (e.g., equal, less than).  
• rs1 and rs2 (5 bits each): Registers for comparison.  
• Immediate (12 bits): Offset for branching, split across the instruction.

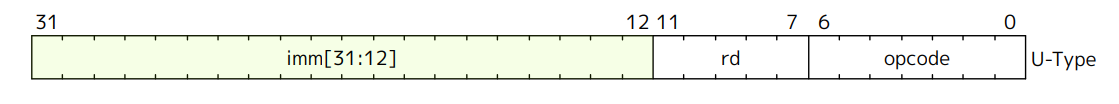


### 5. U-Type (Upper Immediate)

Loads a 20-bit immediate value into the upper 20 bits of a register, used for large constants or addresses.

Fields:

• opcode (7 bits): Indicates an upper immediate operation (e.g., Load Upper Immediate).  
• rd (5 bits): Destination register.  
• Immediate (20 bits): High-order immediate value.



### 6. J-Type (Jump)

Supports unconditional jumps with an address calculated by adding an immediate offset to the current program counter, often used for function calls.

Fields:

• opcode (7 bits): Indicates jump operation.  
• rd (5 bits): Stores return address (PC + 4).  
• Immediate (20 bits): Offset for the jump target, split within the instruction.

## 

## RISC-V Instruction Examples

|  |  |  |
| --- | --- | --- |
| Assembly Instruction | Instruction Type | Encoding (Hexadecimal) |
| lui a0, 0x21 | U-Type | 0x02100537 |
| addi sp, sp, -16 | I-Type | 0xff110113 |
| li a2, 15 | I-Type | 0x00f00113 |
| sd ra, 8(sp) | S-Type | 0x00a13023 |
| ld ra, 8(sp) | I-Type | 0x00a13003 |
| auipc a5, 0xffff0 | U-Type | 0xffff0537 |
| add a3, a4, a2 | R-Type | 0x00c202b3 |
| sub a1, a0, a2 | R-Type | 0x40a20333 |
| jal x1, 12 | J-Type | 0x0000006f |
| jalr x0, x1, 0 | I-Type | 0x000080e7 |
| slli a5, a5, 2 | I-Type | 0x0022b193 |
| bne a2, a3, -4 | B-Type | 0xfff2dfe3 |
| and a4, a3, a2 | R-Type | 0x00c2e2b3 |
| or a5, a4, a3 | R-Type | 0x00e2f2b3 |
| xor a6, a4, a3 | R-Type | 0x00e2f333 |