R-type：

The R-type command format is very clear. In the actual encoding process, the arrangement of encoding positions is meaningful. For example, the encoding position of the three register indexes in different instruction formats are always the same. Index of Rd is at 11-7, Index of rs1 is at 19-15, and Index of rs2 is at 24-20. This is their fixed position. Some instructions may not be useful. The index to the partial register. For example, there is no rs2 in the second instruction type I-type, but there are rs1 and rd and their indexes are in the corresponding positions. For another example, in s-type funct3 is at bits 14-12. The opcode is available in all instruction formats, and the position remains unchanged, always bit 0-6.

Structure: *opcode | rd | funct3 | rs1 | rs2 | funct7*

Use: Perform arithmetic/logic operations requiring two source registers (rs1, rs2) and one destination register (rd).

Examples:

add x3, x1, x2 (x3 = x1 + x2)

sub x3, x1, x2 (x3 = x1 - x2)

I-TYPE:

The upper 12 bits of I-type is an immediate number. The opcode is different from other instruction formats because the corresponding specific operations are different, and other parts are very similar to R-type.

Structure: *opcode | rd | funct3 | rs1 | immediate*

Use: Operations using one register and an immediate constant (e.g., arithmetic with constants, memory access).

Examples:

addi x3, x1, 10 (x3 = x1 + 10)

lw x3, 4(x1) (Load word from address x1 + 4 into x3)