**1. R-type (Register)**

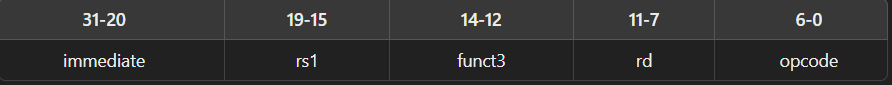
Used for arithmetic and logical operations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 31-25 | 24-20 | 19-15 | 14-12 | 11-7 | 6-0 |
| Funct7 | Rs2 | Rs1 | Funct3 | Rd | opcode |

* **opcode**: Basic operation of the instruction.
* **rd**: Destination register.
* **Func t3**: Additional opcode extension.
* **rs1**: First source register.
* **rs2**: Second source register.
* **funct7**: Further distinguishes between different instructions.

**2. I-type (Immediate)**

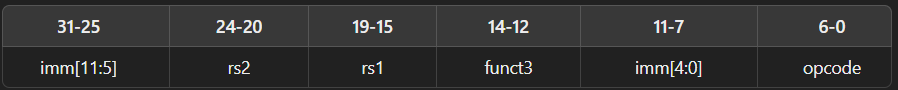
Used for immediate arithmetic, loads, and other operations that need a single immediate value.



* **opcode**: Basic operation of the instruction.
* **rd**: Destination register.
* **funct3**: Additional opcode extension.
* **rs1**: Source register.
* **immediate**: Immediate value (12 bits).

**3. S-type (Store)**

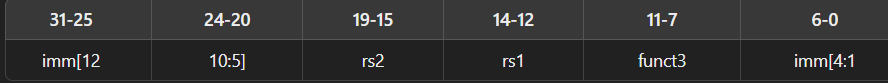
Used for store instructions, where data is stored from a register to memory.



* **opcode**: Basic operation of the instruction.
* **imm[11:5]**: Part of the immediate value.
* **rs1**: Base register.
* **rs2**: Source register.
* **funct3**: Additional opcode extension.
* **imm[4:0]**: Part of the immediate value.

**4. B-type (Branch)**

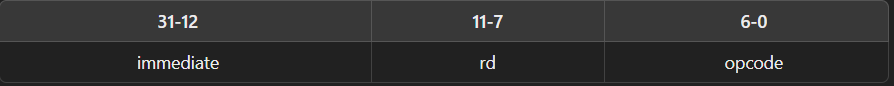
Used for conditional branch instructions.



* **opcode**: Basic operation of the instruction.
* **imm[12|10:5]**: Part of the immediate value.
* **rs1**: First source register.
* **rs2**: Second source register.
* **funct3**: Additional opcode extension.
* **imm[4:1|11]**: Part of the immediate value.

**5. U-type (Upper Immediate)**

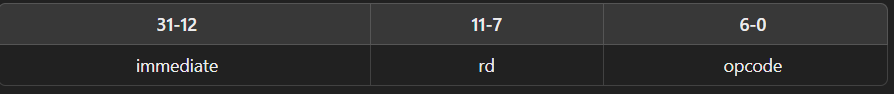
Used for instructions that operate with a 20-bit upper immediate value.

****

* **opcode:** Basic operation of the instruction.
* **rd:** Destination register.
* **immediate:** Immediate value (20 bits).

**6. J-type (Jump)**

Used for jump instructions with a 20-bit immediate value.

****

* ***opcode:*** Basic operation of the instruction.
* ***rd***: Destination register.
* ***immediate:*** Immediate value (20 bits).

These formats are designed to make decoding simple and efficient, ensuring that the RISC-V architecture remains true to its principles of simplicity and scalability**.**