## R-Type

|  |  |  |  |
| --- | --- | --- | --- |
| Op-Code  3 bit | rs  2 bit | rt  2 bit | rd  1 bit |

**I-Type**

|  |  |  |  |
| --- | --- | --- | --- |
| Op-Code  3 bit | rs  2 bit | rt  2 bit | Immediate  1 bit |

## J-Type

|  |  |
| --- | --- |
| Op-Code  3 bit | Target Address  5 bit |

**Operations**

## R-Type Table

|  |  |  |
| --- | --- | --- |
| **Instruction Type** | **Instruction** | **Op-Code** |
| Arithmetic | add | 0000 |
| sub | 0001 |
| Logical | and | 0010 |
| or | 0011 |
| nor | 0100 |
| xor | 1110 |

**I-Type Table**

|  |  |  |
| --- | --- | --- |
| **Instruction Type** | **Instruction** | **Op-Code** |
| Data Transfer | lw | 0101 |
| sw | 0110 |
| Conditional Branch | beq | 0111 |
| mul | 1000 |
| Arithmetic | addi | 1001 |
| Logical | andi | 1010 |
| ori | 1011 |
| sll | 1100 |
| srl | 1101 |

**J-Type Table**

|  |  |  |
| --- | --- | --- |
| **Instruction Type** | **Instruction** | **Op-Code** |
| Unconditional Jump | J | 1111 |

# Operations’ Instructions

## add:

It adds the content of the source register 1 to the contents of the source register 2 and saves it in the destination register.

Operation: $s0 = $s0 + $t0 Syntax: add $s0, $s0, $t0

## sub:

It subtracts the content of the source register 2 from the contents of the source register 1 and saves it in the destination register.

Operation: $s0 = $s0 - $t0 Syntax: sub $s0, $s0, $t0

## and:

It does a bit by bit logical and operation between two source registers contents. Operation: $s0 = $s0 & $t0

Syntax: and $s0, $s0, $t0

## or:

It does a bit by bit logical or operation between two source registers contents.

Operation: $s0=$s0||$t0

Syntax: or $s0, $s0, $t0