Machine code written in: opcode- rs- rt- rd- shm- (r-format), opcode-rs-rt-immediate (i-format)

Underlined registers belong to that format i.e: are fixed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction** | **Type** | **Opcode** | **Functionality** | **Assembly opcode** | **Format || Example** |
| Load word | Data transfer | 0000 | Copy 16 bits from memory to register | lw | lw $rt im($rs) || lw $3 0($2) |
| Store word | Data transfer | 0001 | Copy 16 bits from register to memory | sw | sb $rs im($rt) || sw $5 6($1) |
| Load address | Data transfer | 0010 | Loads the decimal address(not its contents) into register | la | la $rt $zero im ||lui $3 $zero 10 |
| Add | Arithmetic | 0011 | Adds the contents of 2 registers | add | add $rd $rs $rt || add $5 $4 $3 |
| Subtract | Arithmetic | 0100 | Subtracts the content of one register from other | sub | sub $rd $rs $rt || sub $5 $4 $3 |
| AND | Logical | 0101 | Bitwise AND | and | and $rd $rs $rt || and $5 $6 $7 |
| OR | Logical | 0110 | Bitwise OR | or | or $rd $rs $rt || or $5 $6 $7 |
| NOT | Logical | 0111 | Bitwise NOT | not | not $rd $rs $zero ||nor $1 $2 $zero |
| Shift left | Logical | 1000 | Shift left by constant number of bits | sl | sl $rd $rs $zero shm || sl $3 $4 $zero 2 |
| Shift right | Logical | 1001 | Shift right by constant number of bits | sr | sr $rd $rs $zero shm || sr $3 $4 $zero 2 |
| Or immediate | Data transfer | 1010 | Performs Or operation between a register and decimal | ori | ori $rt $rs im || ori $3 $3 14 |
| Add immediate | Arithmetic | 1011 | Adds an immediate value to a register’s content | addi | addi $rt $rs im || addi $3 $zero 10 |
| Branch on equal | Branching | 1100 | Test if registers are equal | beq | beq $rt $rs label || beq $3 $4 15 |
| set if less than | Arithmetic | 1101 | Sets a register value to 1 if 1st reg < 2nd reg | slt | slt $rd $rs $rt |
|  |  |  |  |  |  |