

MIPS Instruction Set

R-Type Instructions

- `add rd, rs, rt` : Add
- `addu rd, rs, rt` : Add Unsigned
- `sub rd, rs, rt` : Subtract
- `subu rd, rs, rt` : Subtract Unsigned
- `and rd, rs, rt` : Bitwise AND
- `or rd, rs, rt` : Bitwise OR
- `xor rd, rs, rt` : Bitwise XOR
- `nor rd, rs, rt` : Bitwise NOR
- `slt rd, rs, rt` : Set Less Than
- `sltu rd, rs, rt` : Set Less Than Unsigned
- `sll rd, rt, shamt` : Shift Left Logical
- `srl rd, rt, shamt` : Shift Right Logical
- `sra rd, rt, shamt` : Shift Right Arithmetic
- `jr rs` : Jump Register

I-Type Instructions

- `addi rt, rs, immediate` : Add Immediate
- `addiu rt, rs, immediate` : Add Immediate Unsigned
- `andi rt, rs, immediate` : Bitwise AND Immediate
- `ori rt, rs, immediate` : Bitwise OR Immediate
- `xori rt, rs, immediate` : Bitwise XOR Immediate
- `slti rt, rs, immediate` : Set Less Than Immediate
- `sltiu rt, rs, immediate` : Set Less Than Immediate Unsigned
- `lui rt, immediate` : Load Upper Immediate
- `beq rs, rt, label` : Branch if Equal
- `bne rs, rt, label` : Branch if Not Equal
- `lb rt, offset(rs)` : Load Byte
- `lh rt, offset(rs)` : Load Halfword
- `lw rt, offset(rs)` : Load Word
- `lbu rt, offset(rs)` : Load Byte Unsigned
- `lhu rt, offset(rs)` : Load Halfword Unsigned
- `sb rt, offset(rs)` : Store Byte
- `sh rt, offset(rs)` : Store Halfword
- `sw rt, offset(rs)` : Store Word

J-Type Instructions

- `j target` : Jump
- `jal target` : Jump and Link

Floating-Point Instructions (if the FPU is present)

- `add.s rd, rs, rt` : Add Single Precision
- `sub.s rd, rs, rt` : Subtract Single Precision
- `mul.s rd, rs, rt` : Multiply Single Precision
- `div.s rd, rs, rt` : Divide Single Precision
- `lwc1 rt, offset(rs)` : Load Word to Floating Point
- `swc1 rt, offset(rs)` : Store Word from Floating Point

Pseudoinstructions

- `move rd, rs` : Move
- `li rd, immediate` : Load Immediate
- `b label` : Branch
- `nop` : No Operation