# North South University

Department of Electrical & Computer Engineering

## Project Report

Course ID & Name: CSE332 Computer Architecture & Organization

Section: 2

Semester: Spring 2025

Project Name: 18-bit custom MIPS Instruction Set Architecture

Submitted to: Tnf

Report Submission Date: 22/3/25

Group Number: 12

Group Members ID & Name:

| | |
|---|---|
| 1. 2132239642 | Farhan Faiyaz |
| 2. 2311657042 | Saud |
| 3. | |
| 4. | |
| 5. | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Remarks:

Score

## Number of Operands:

The MIPS architecture can have up to 3 operands at a time. We have shown how and where they are used in our custom 18-bit architecture.

3 Operands: ADD, NAND, NOR, SLT (all registers); BEQ (two registers + offset); ORi, SUBi (two registers + immediate).

2 Operands: LW, SW (register + memory address).

1 Operand: JMP (memory address).

## Types of Operands

In our 18-bit MIPS architecture, operands are classified based on their source and usage.

**Register-Based Operands** involve direct register interactions, mainly in R-type and I-type instructions. R-type operations like ADD, NAND, NOR, and SLT use only registers, while I-type instructions such as BEQ, ORi, and SUBi involve at least one register, with the latter two also using an immediate value.

**Memory-Based Operands** appear in load and store instructions, where a base register and an offset define the memory address. LW (Load Word) transfers data from memory to a register, while SW (Store Word) moves data from a register to memory. Their formats are LW rt, offset(rs) and SW rt, offset(rs).

## Number of Operations

Our 18-bit MIPS design includes the following 10 operations:

1. **ADD** (Addition)
2. **JMP** (Jump)
3. **LW** (Load Word)
4. **BEQ** (Branch if Equal)
5. **SW** (Store Word)
6. **NAND** (Bitwise NAND)
7. **NOR** (Bitwise NOR)
8. **ORi** (Bitwise OR with Immediate)
9. **SLT** (Set Less Than)
10. **SUBi** (Subtract Immediate)

## Types of Operations and Opcodes

| Category | Operation | Opcode (4-bit) | Binary Value |
|---|---|---|---|
| **R-type** | ADD | 0000 | 0000 + rs(4) + rt(4) + rd(4) + 00 |
| | NAND | 0000 | 0000 + rs(4) + rt(4) + rd(4) + 01 |
| | NOR | 0000 | 0000 + rs(4) + rt(4) + rd(4) + 10 |
| | SLT | 0000 | 0000 + rs(4) + rt(4) + rd(4) + 11 |
| **I-type** | SUBi | 0001 | 0001 + rs(4) + rt(4) + imm(6) |
| | ORI | 0010 | 0010 + rs(4) + rt(4) + imm(6) |
| | BEQ | 0011 | 0011 + rs(4) + rt(4) + offset(6) |
| | LW | 0100 | 0100 + rs(4) + rt(4) + offset(6) |
| | SW | 0101 | 0101 + rs(4) + rt (4)+ offset |
| **J-type** | JMP | 1000 | 0110 + address(14) |

## Number of Instruction Formats

Three formats: **R-type**, **I-type**, and **J-type**. We have for our ISA four R-types, five I-types, and a single J-type instruction.

## Instruction Formats

**R-type (Register Operations)**

Opcode: 0000 (4 bits, common for all R-type).

Used for ADD, NAND, NOR, SLT:

| Opcode | Rs | Rt | Rd | function |
|---|---|---|---|---|
| 4bits | 4bits | 4bits | 4bits | 2bits |

Total: 4 + 4 + 4 + 4 + 2 = 18 bits.

**I-type (Immediate/Memory/Branch)**

Opcode: 4 bits (unique for each I-type operation).

Used for SUBi, ORI, BEQ, LW, SW:

| Opcode | Rs | Rt | Immediate/Ofset |
|---|---|---|---|
| 4bits | 4bits | 4bits | 6bits |

Total: 4 + 4 + 4 + 6 = 18 bits.

**J-type (Jump)**

Opcode: 0110 (4 bits).

Used for JMP:

| Opcode | Address |
|---|---|
| 4bits | 14bits |

Total: 4 + 12 = 18 bits.

## Table with assigned function/operation list and assigned opcode/ function bit.

| **Category** | **Operation** | **Opcode (4-bit)** | **funct (2-bit)** |
|---|---|---|---|
| **R-type** | ADD | 0000 | 00 |
| | NAND | 0000 | 01 |
| | NOR | 0000 | 10 |
| | SLT | 0000 | 11 |

| | | | |
|---|---|---|---|
| **I-type** | SUBi | 0001 | XX |
| | ORI | 0010 | XX |
| | BEQ | 0011 | XX |
| | LW | 0100 | XX |
| | SW | 0101 | XX |
| **J-type** | JMP | 1000 | XX |

## Table with the number of registers.

| **Register** | **4-bit Binary** | **Use** |
|---|---|---|
| R0 | 0000 | For R type (Always 0) |
| R1–R15 | 0001-1111 | For other types/purpose |