



NORTH SOUTH UNIVERSITY

Department of Electrical & Computer Engineering (ECE)

Course Code - CSE 332

Section: 05

Faculty: Tanjila Farah (Tnf)

20 Bit CPU

Group : 16

Name : Md Abu Bokar

ID : 2222977642

→ Number of Operands

- ◆ **R-type instruction:** 3 operands (e.g, add, sub, and, or etc)
- ◆ **I-type instruction:** 2 operands (e.g, beq, sw, lw, addi etc)
- ◆ **J-type instruction:** 1 operand (target address for jump)

→ Types of Operands

- ◆ **Register-based:** Operands are registers like \$r0, \$r1, ..., \$r15
- ◆ **Memory-based:** For lw (load word) and sw (store word), memory access is performed indirectly through a base register and immediate offset.
- ◆ **Immediate-based:** Instructions like addi, beq, and jmp use immediate values as operands.

→ Number of Operations

- ◆ Total number of operations are 15. Here is the list of operations in serial order:

R-Type

| | | | | | | | | | |
|--------|----|----|----|----|---|----|---|----------|---|
| 19 | 16 | 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| opcode | | rs | | rt | | rd | | function | |

I-Type

| | | | | | | | |
|--------|----|----|----|----|---|-----------|---|
| 19 | 16 | 16 | 12 | 10 | 8 | 7 | 0 |
| opcode | | rs | | rd | | immediate | |

J-Type

| | | | |
|--------|----|---------|---|
| 19 | 17 | 16 | 0 |
| opcode | | address | |

Table

| Inst. | opcode | rs | rt | rd | func. | Operation |
|-------|--------|---------|------|-----------|-------|------------------------------------|
| ADD | 0000 | used | used | used | 0000 | $\$r1 = \$r1 + \$r2$ |
| SLL | 0000 | x | used | used | 0001 | $\$r1 = \$r2 \ll a$ |
| OR | 0000 | used | used | used | 0010 | $\$r1 = \$r2 \mid \$r3$ |
| SUB | 0000 | used | used | used | 0011 | $\$r1 = \$r1 - \$r2$ |
| NOR | 0000 | used | used | used | 0100 | $\$r1 = \sim(\$r2 \mid \$r3)$ |
| SLT | 0000 | used | used | used | 0101 | $\$r1 = (\$r2 < \$r3)$ |
| AND | 0000 | used | used | used | 0110 | $\$r1 = \$r1 \& \$r2$ |
| NOP | 0000 | x | x | x | 0111 | $\$r1 = \$r2 \ggg a$ |
| SRL | 0000 | x | used | used | 1000 | Does nothing |
| Inst. | opcode | rs | rt | immediate | | |
| BEQ | 0001 | used | used | used | | if ($\$r1 == \$r2$) pc += i << 2 |
| SW | 0010 | used | used | used | | MEM [$\$r2 + i$]:4 = $\$r1$ |
| BNE | 0011 | used | used | used | | if ($\$r1 != \$r2$) pc += i << 2 |
| ADDi | 0100 | used | used | used | | $\$r1 = \$r0 + a$ |
| LW | 0110 | used | used | used | | $\$r1 = \text{MEM } [\$r2 + i]:4$ |
| Inst. | opcode | address | | | | |
| JMP | 0101 | used | | | | pc += i << 2 |

Control Unit

| Inst. | op | func | ALU op | Reg write | Alu src | Mem write | Mem read | Mem to reg | branch | Reg dst | jump |
|-------|------|------|--------|-----------|---------|-----------|----------|------------|--------|---------|------|
| ADD | 0000 | 0000 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| SLL | 0000 | 0001 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| OR | 0000 | 0010 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| BEQ | 0001 | x | 01 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| SW | 0010 | x | 01 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| SUB | 0000 | 0011 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| NOR | 0000 | 0100 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| SLT | 0000 | 0101 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| BNE | 0011 | x | 01 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| ADDi | 0100 | x | 01 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| JMP | 0101 | x | 11 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| LW | 0110 | x | 01 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| AND | 0000 | 0110 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| NOP | 0000 | 0111 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SRL | 0000 | 1000 | 10 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Alu Control

| No | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|-----|-----|----|-----|-----|-----|-----|-----|-----|
| Inst. | ADD | SLL | OR | SUB | NOR | SLT | AND | NOP | SRL |

→ Types of Operations

◆ Arithmetic Operations:

| Operation | Opcode |
|-----------|--------|
| add | 0000 |
| sub | 0000 |
| addi | 0100 |

◆ Logical Operations:

| Operation | Opcode |
|-----------|--------|
| sll | 0000 |
| or | 0000 |
| nor | 0000 |
| slt | 0000 |
| and | 0000 |
| srl | 0000 |

◆ Branch Operations:

| Operation | Opcode |
|-----------|--------|
| beq | 0001 |
| bne | 0011 |
| jmp | 0101 |

◆ Memory Operations:

| Operation | Opcode |
|-----------|--------|
| sw | 0010 |
| lw | 0110 |

◆ Special Operation:

| Operation | Opcode |
|-----------|--------|
| nop | 0000 |

→ Number of Instruction Formats

- ◆ There are 3 different instruction formats:
 - R-type
 - I-type
 - J-type

→ Description of the Formats

- ◆ R-type:

| Field | Bit | Description |
|----------|-------|-------------------------------|
| opcode | 4 bit | Operation code (0 for R-type) |
| rs | 4 bit | Source register |
| rt | 4 bit | Target register |
| rd | 4 bit | Destination register |
| function | 4 bit | Function code |

- ◆ I-type:

| Field | Bit | Description |
|-----------|-------|-----------------|
| opcode | 4 bit | Operation code |
| rs | 4 bit | Source register |
| rt | 4 bit | Target register |
| immediate | 8 bit | Immediate value |

- ◆ J-type:

| Field | Bit | Description |
|---------|--------|-----------------|
| opcode | 4 bit | Operation code |
| address | 16 bit | Address to jump |

→ Conclusion

The implementation of the 20-bit CPU was a challenging yet rewarding project that provided valuable insights into the design and functioning of a processor. By designing a custom Instruction Set Architecture (ISA) capable of supporting various instructions like arithmetic, logical, branch, memory and special operations, this CPU demonstrates efficiency in handling computational tasks. The project involved the careful integration of multiple components, including:

- Register File
- Alu
- Control Unit
- Alu Control

In conclusion, this project enhanced our knowledge of CPU Architecture and also showcased the importance of planning, testing and iteration in creating a functional hardware system.