



NORTH SOUTH UNIVERSITY

Department of Electrical & Computer Engineering

Semester: Summer 2022

Course Initial: CSE332L

Course Title: Computer Organization and Architecture

Faculty: Dr. Mainul Hossain

Project on:

Design a 10-bit Custom RISC-V Microprocessor

Section: 10

Group: 05

Group Members:

Name	Student ID
Md. Afjal Hossain Dali	2013652042
Samin Yasar	2011279042
Azfar Sadat Khan	2012395042
Sheikh Nazibur Rahman	2021572642

Part 1.1: ISA Design (Document)

We have designed a 10-bit custom RISC-V Microprocessor

R-Type Format

6-9	4-5	2-3	0-1
4 bits	2 bits	2 bits	2 bits
opcode	rs	rt	rd

I-Type Format

6-9	4-5	2-3	0-1
4 bits	2 bits	2 bits	2 bits
opcode	rs	rt	immediate

Category	Operation	Format	Example	Meaning	Opcode
Logical	AND	R-Type	and \$r1, \$r2	$\$r2 = \$r1 \& \$r2$	0000
	OR	R-Type	or \$r2, \$r3	$\$r2 = \$r1 \vee \$r3$	0001
Arithmetic	Addition	R-Type	Add \$r1, \$r2	$\$r2 = \$r1 + \$r2$	0010
	Subtraction	R-Type	Sub \$r1, \$r2	$\$r2 = \$r1 - \$r2$	0011
	Addition	I-Type	addi \$rd, Constant	$\$rd = \$rd + const$	0010
	Subtraction	I-Type	subi \$rd, Constant	$\$rd = \$rd - const$	0011

And: It does **AND** logical operation between two register and store the value in a destination register.

- Operation: $d = d \& s$
- Syntax: AND \$rd, \$rs ($\$r2 = \$r1 \& \$r2$)

OR: It does **OR** logical operation between two register and store the value in a destination register.

- Operation: $d = d \vee s$
- Syntax: OR \$rd, \$rs ($\$r2 = \$r1 \vee \$r2$)

ADD.

- Operation: $d = d + s$
- Syntax: add \$rd, \$rs ($\$rd = \$rd + \$rs$)

Sub

- Operation: $d = d - s$

- Syntax: **sub \$rd, \$rs** ($\$rd = \$rd - \$rs$)

Addi: It adds a value from register with an integer value and stores the result in destination register.

- Operation: **d = s + constant**
- Syntax: **addi \$rd, Constant** ($\$rd = \$rd + \text{const.}$)

Subi: It subtract value form register with an integer value and stores the result in destination register.

- Operation: **d = s - constant**
- Syntax: **subi \$rd, Constant** ($\$rd = \$rd - \text{const.}$)

List of Registers:

For 2-bits we assigned 4 registers.

Register Name	Value
\$S0	00
\$S1	01
\$S2	10
\$S3	11

Types of operands:

2-types of operands. They are: -

- Register-operand
- immediate-operand

Types of operations: 2 types

- Arithmetic: 4 (ADD, SUB, addi, subi)
- Logical: 2 (AND, OR)

Table for control unit:

Operation	Opcode	Reg dst	ALU source	ALU OP	Memory read	Memory write	Reg write
AND	0001	0	0	00	0	0	1
OR	0010	0	0	01	0	0	1
ADD	0011	0	0	10	0	0	1
SUB	0100	0	0	11	0	0	1
ADDi	0011	1	1	10	0	0	1
SUBi	0100	1	1	11	0	0	1

Analysis: We used 4 bits for opcode so we can use extra 2 bit for shift amount. 10 bits constrains limited our options so we were needed to be careful allocating space for opcode, source and destination. For this similar reason we allocated 4 bits for immediate

