North South University

Department of Electrical & Computer Engineering

Course Code: CSE332

Course Title: Computer Organization and Architecture

Submitted to: Dr. Mainul Hossain (MHo1)

Design a 10-bit Custom RISC-V Microprocessor

ISA DESIGN

Section: 10

Group Number: 4

Submitted by:

Serial No	Student Name	ID
1	Foyez Ahmed	2013122042
2	Nusrat Zahan	2013942642
3	Reem Gazi Mahmud Hossain	2014234042
4	Syed Niamul Kazbe Rayian	2021931642

Title: Design a 10-bit Custom RISC-V Microprocessor

Objectives:

Our objective was to design a 10-bit Custom RISC-V Microprocessor which can solve a particular problem i.e. arithmetic addition, logical operation, etc.

Number of Operands:

- There are three operands, which are represented as **rd**, **rs**, and **rt**.

Types of Operands:

- Register based
- Memory based

Operations:

We allocated 3 bits for opcode; therefore, the number of instructions can be executed is 2^3 or 8. Because 000 is for addition, 001 for subtraction, 010 for load, 011 for store, 100 for beq, and 101 for jump.

Types of Operations:

There will be three different types of operations performed.

1. Arithmetic
2. Logical
3. Data Transfer

Category	Operation	Name	Opcode	Type	Syntax	Comments
Arithmetic	Addition	add	000	R	add $rd, $rs, $rt	Three registers
Arithmetic	Subtraction	sub	001	R	sub $rd, $rs, $rt	Three registers
Data Transfer	Load Word	lw	010	I	lw $rd, $rs, imm	Load data from memory to register
Data Transfer	Store Word	sw	011	I	sw $rd, $rs, imm	Store data from register to memory
Logical	Shift Logical	sll	100	I	sll $rd, $rs, offset	Shift by constant

Types of Instruction:

We are using 2 types of instruction for our ISA

1. Register Type – (R-Type)
2. Immediate Type – (I-Type)

(R-Type) ISA Format

opcode	**rd**	**rs**	**rt**
3 bits	**4 bits**	**1 bit**	**2bits**

(I-Type) ISA Format

opcode	**rd**	**rs**	**Imm**
3 bits	**4 bits**	**1 bit**	**2bits**

List of Registers:

We allocated 2 bits for $rt registers, 1 bit for $rs registers and 4 bits for $rd registers.

Register Table

Register Name	Type	Register Number	Binary Value
$t0	Saves values	0	0000
$t1	Saves values	1	0001
$t2	Saves values	2	0010
$t3	Saves values	3	0011
$t4	Saves values	4	0100
$t5	Saves values	5	0101
$t6	Saves values	6	0110
$t7	Saves values	7	0111
$t8	Saves values	8	1000
$t9	Saves values	9	1001

<u>Detailed Instructions and their Operations Instruction Description:</u>

Addition: It adds values of two temporary registers and stores the result in a destination register.

ADD

- Operation: d = s + t
- Syntax: add $rd, $rs, $rt

SUB

- Operation: d = s - t
- Syntax: sub $rd, $rs, $rt

Load Word: It loads required value from the memory to the register for calculation.

lw

- Operation: d = M[s + t]
- Syntax: lw $rd, $rs, imm

Store Word: It stores specific value from register to memory.

sw

- Operation: M[s + t] = d
- Syntax: sw $rd, $rs, imm

Translating Some High Level Language codes using our designed 10-bit ISA

Assigned register for individual variable: $t0 - $t9

1. y = x + a

```
add $rd, $rs, $rt      #values of $rs and $rt will be added and stored
                        in $rd
```

2. y = x - a

```
sub $rd, $rs, $rt      #values of $rs and $rt will be subtracted and
                        stored in $rd
```

3. y = x [a]   #we have fetched the value that was stored in x[a]

```
lw $rd, $rs, imm
```

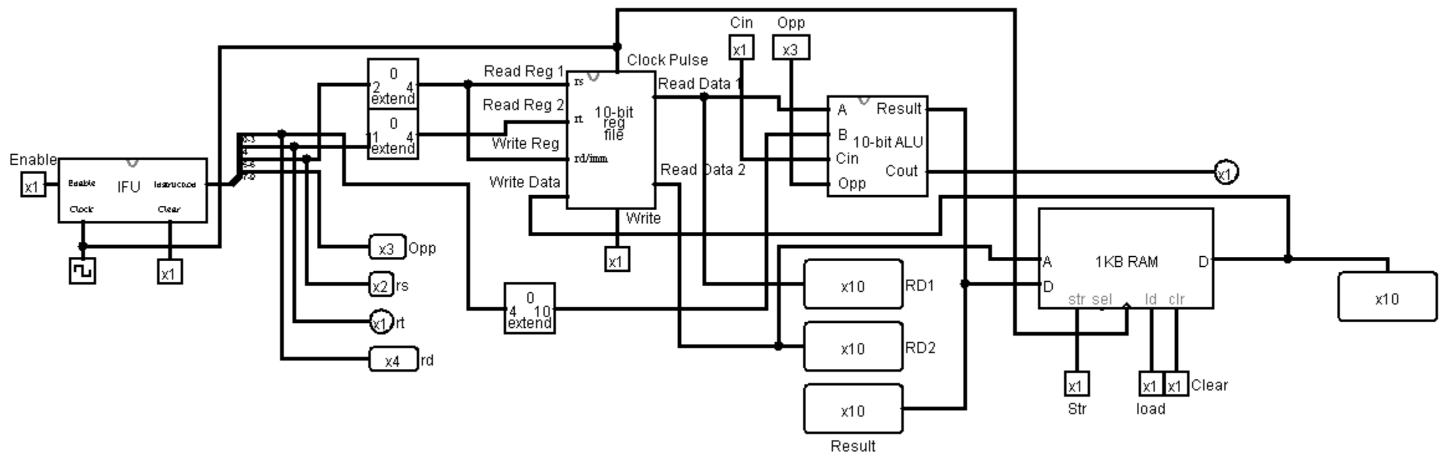
4. x [a] = y   #we have stored the value that was fetched from x[a]

```
sw $rd, $rs, imm
```

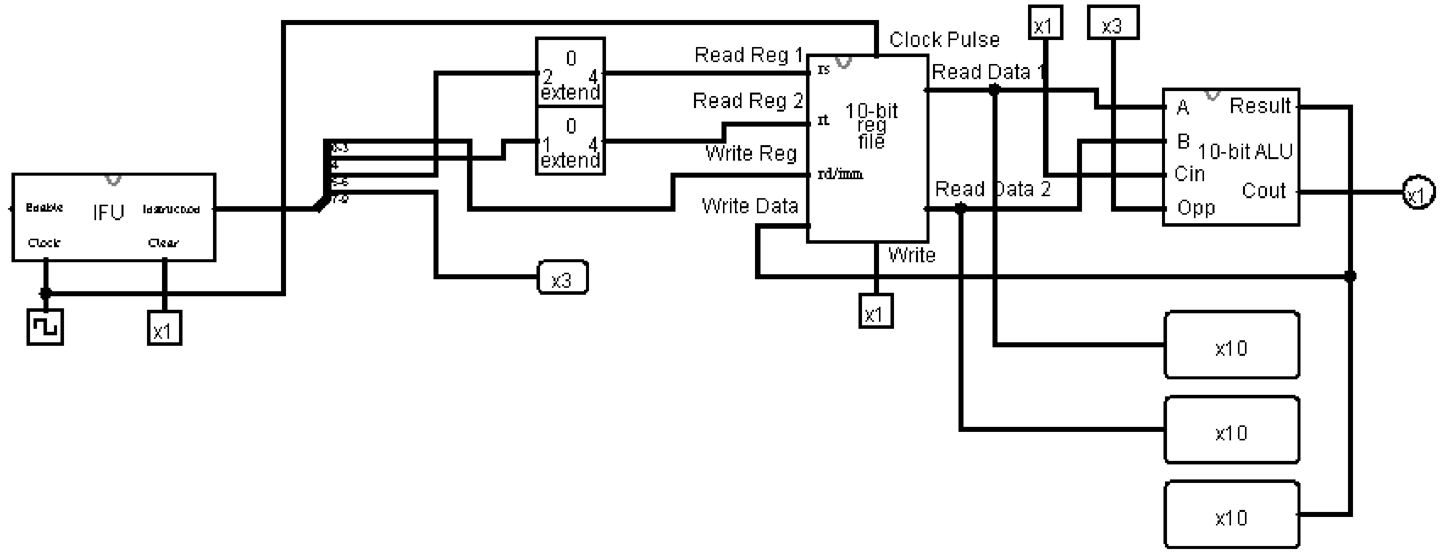
Analysis and Limitation:

Limitations:

1. Our ISA does not have J-type instruction.
2. Since our ISA is of 10 bits, we cannot use anymore registers greater than 10 bits.
3. We cannot use more than 8 bits for our opcode.
4. We cannot use more than 1 bit for $rs registers.



Enable
IFU
Instruction
Clock
Clear
extend
Read Reg 1
Read Reg 2
Write Reg
Write Data
10-bit reg file
Clock Pulse
Read Data 1
Read Data 2
Write
Cin
Opp
A
B
Result
10-bit ALU
Cout
1KB RAM
str sel
ld clr
Str
load
Clear
RD1
RD2
Result
rs
rt
rd



IFU
Enable
Instruction
Clock
Clear
extend
Read Reg 1
Read Reg 2
Write Reg
Write Data
10-bit reg file
Clock Pulse
Read Data 1
Read Data 2
Write
Cin
Opp
A
B
Result
10-bit ALU
Cout