North South University

Final Project Report

Title: 32bit MIPS processor.

Course Code: CSE332

Semester: Summer 2019

Submitted By:	Submitted To:
Ridwanul Haque	**Nur Yan**
ID: 172 1144 042	**Initial: NYH**
Section: 11	

Submission Date: 11-August-2019

Objective:

Our objective is to implement a 32-bit single cycle CPU which can perform the following type instructions:

- R-type:
- I-type:
- J-type:

Instruction SET Architecture Design:

- **R-type:**

Opcode	**RS**	**RT**	**RD**	**Shamt**	**Function**
6 bit	**5 bit**	**5 bit**	**5 bit**	**5 bit**	**6 bit**

- **I-type:**

Opcode	**RS**	**RT**	**Immediate**
6 bit	**5 bit**	**5 bit**	**16 bit**

- **J-type:**

Opcode	**Address**
6 bit	**26 bit**

Here,

RS = First Source Register

RT = Second Source Register

RD = Destination Register

Data_bits: 32 bits throughout the project

Instruction with Opcode List:

Instructions	**Opcode**
R-Type	000000
ORI	000001
LW	000010
BNE	000011
BEQ	000100
J	000101

Given Instructions List:

R-Type			
Instruction	**Name**	**Action**	**Function**
ADD rd, rs, rt	Addition	rd = rs + rt	**00001**
OR rd, rs, rt	OR	rd = rs || rt	**00010**
AND rd, rs, rt	AND	rd = rs & rt	**00011**

I-Type			
Instruction	**Name**	**Action**	**Function**
ORI rt, rs, imm	OR immediate	rt = rs || imm	XXXXXX
LW rt, offset(rs)	Load Word	rt = M[offset + rs]	XXXXXX
BNE rs, rt, offset	Branch Not Equal	if(rs!=rt) than pc = pc + offset	XXXXXX
BEQ rs, rt, offset	Branch On Equal	if(rs==rt) than pc = pc +offset	XXXXXX

J-Type			
Instruction	**Name**	**Action**	**Opcode**
J target	Jump	pc [0-19] = target; pc [2023] = (pc+1) [20-23];	000101

Control unit signals:

Instruction	**RegDest**	**ALUSrc**	**Mem toReg**	**Reg Write**	**Mem Read**	**Mem Write**	**BEQ**	**BNE**	**Jump**	**ALUOp 1**	**ALUOp 0**
R-format	1	0	0	1	0	0	0	0	0	1	0
LW	0	1	1	1	1	0	0	0	0	0	0
ORI	0	1	0	1	0	0	0	0	0	1	1
BNE	0	0	0	0	0	0	0	1	0	0	1
BEQ	0	0	0	0	0	0	1	0	0	0	1
JUMP	0	0	0	0	0	0	0	0	1	X	X

Instruction table for ALU control:

Instruction	Control Unit Opcode	ALU Op	Instruction Operation	Function Field	Desired ALU Action	ALU Control Input
R-type	000000	10	Add	000001	And	000
R-type	000000	10	Or	000010	Or	010
R-type	000000	10	And	000011	Add	001
ORI	000001	11	Or imidiate	XXXXXX	Or	010
LW	000010	00	Load	XXXXXX	Add	000
BNE	000011	01	Branch not Equal	XXXXXX	Sub	100
BEQ	000100	01	Branch Equal	XXXXXX	Sub	100
Jump	000101	XX	Jump unconditional	XXXXXX	XX	XXX

ALU Control Table for ALU Operation:

ALU OP				Function					OPERATION Bin S1 S0
Op2	**Op1**		**F5**	**F4**	**F3**	**F2**	**F1**	**F0**	
0	0	LW	X	X	X	X	X	X	000
1	0	BNE, BEQ	X	X	X	X	X	X	100
1	1	Ori	X	X	X	X	X	X	010
0	1	Add	0	0	0	0	0	1	000
0	1	Or	0	0	0	0	1	0	010
0	1	And	0	0	0	0	1	1	001

Equations for ALU Operations:

Bin = OP1’. OP2

S1 = Op1. F0’

S0 = F0. F1