CSE 332L

Section 02

22 Bit Single Cycle CPU

Project Report

Submitted To:

Faculty: Tanjila Farha (TnF)

Instructor: Kamrun Naher

Submitted By:

Group 03	
Ekhfa Hossain	172 1625 0 42
Sudipta Mandal	172 1489 0 42

Table of Contents

Project Name: **MIPS Architecture (A single CPU cycle)**

Project Bit: **22**

Objective:

Designing and simulating a 22 bit single cycle CPU which can perform R-Type, I-Type, J-Type instruction with 22 bit binary instruction is the main objective of our project.

The circuit will cover the following instructions which are given by the instructor,

TYPE	Instruction	Covered (YES/NO)
R-Type	**ADD**	YES
	SUB	YES
	NAND	YES
	NOR	YES
	AND	YES
	OR	YES
	SLT	YES
	SLL	YES
	SRL	YES
I-Type	**LW**	YES
	SW	YES
	BEQ	YES
	BNE	YES
	ADDi	YES
	SUBi	YES
J Type	**JUMP**	YES

*Mandatory Operations are in BOLD form

Designing:

We were given 22 bit instructions and we divided it in the following format

R-TYPE					
4 bits	3 bits	3 bits	3 bits	5 bits	4 bits
21 18	17 15	14 12	11 09	08 04	03 00
OPP Code	RS	RT	RD	SHIFT	Operation

I-TYPE			
4 bits	3 bits	3 bits	12 bits
21 18	17 15	14 12	11 00
OPP Code	RS	RD	Immediate Value

J-TYPE	
4 bits	18 bits
21 18	17 0
OPP Code	Address

Instruction Set:

OPP Code: Operation Code

RS: Source Register 1st

RT: Source Register 2nd

RD: Destination Register

SHIFT: Shift distance for shifting the value

Operation: Operation value

IV: Immediate Value for I type instruction

Address: Address value for J type instruction to jump in certain address

Tables:

Table 1: Control Unit Control Operations

Instructions	OPP Code	RegDst	ALUSrc	MemToReg	RegWrite	Mem Read	Mem Write	Brunch	Jump	ALU op2	ALU op1	ALU op0
R-Format	0000	1	0	0	1	0	0	0	0	0	1	0
LW	0011	0	1	1	1	1	0	0	0	0	0	0
SW	0100	0	1	0	0	0	1	0	0	0	0	0
ADDi	0101	0	1	0	1	0	0	0	0	0	1	1
SUBi	0110	0	1	0	1	0	0	0	0	1	1	1
BEQ	0111	0	0	0	0	0	0	1	0	0	0	1
BNE	1000	0	0	0	0	0	0	1	0	0	0	1
Jump	1111	0	0	0	0	0	0	0	1	X	X	X

Table 2: ALU Control Operations

Instruction	CU Opp	ALUOpp	Instruction Operation	Operation	ALU Controls	ALU control Input
R-Type	0 0 0 0	0 1 0	AND	0 0 0 0	AND	0 0 0 0
			OR	0 0 0 1	OR	0 0 0 1
			NAND	0 0 1 0	NAND	0 0 1 0
			NOR	0 0 1 1	NOR	0 0 1 1
			ADD	0 1 0 0	ADD	0 1 0 0
			SUB	1 1 0 0	SUB	1 1 0 0
			SLL	0 1 0 1	SLL	0 1 0 1
			SLR	0 1 1 0	SLR	0 1 1 0
			SLT	0 1 1 1	SLT	0 1 1 1
LW	0 0 1 1	0 0 0	Load	xxxx	Add	0 1 0 0
SW	0 1 0 0		Store	xxxx	Add	0 1 0 0
ADDi	0 1 0 1	0 1 1	Add Immediate	xxxx	Add	0 1 0 0
SUBi	0 1 1 0	1 1 1	Subtract Immediate	xxxx	Sub	1 1 0 0
BEQ	0 1 1 1	0 0 1	Branch Equal	xxxx	Sub	1 1 0 0
BNE	1 0 0 0		Branch Not Equal	xxxx	Sub	1 1 0 0
Jump	1 1 1 1	xxx	Jump	xxxx	xxxx	xxxx

Table 3: ALU Result MUX Control Operations

Operation Circuit	Operation (0-3)	Operations	Invert B	ALUOpp	MUXControl
ALU	0000	**AND**	0	000	00
	0001	**OR**	0	001	
	0010	**NAND**	0	010	
	0011	**NOR**	0	011	
	0100	**ADD**	0	100	
	1100	**SUB**	1	100	
SLL Operation	1101	**SLL**	0	101	01
SRL Operation	1110	**SRL**	0	110	10
SLT Operation	1111	**SLT**	0	111	11

Table 4: Control Operations for ALU Control Unit

Instruction	ALU Op(Opp) 2 1 0	Operation(0-3)(O) 3 2 1 0	ALUOperation(ALUOp) 3 2 1 0	Actions
I-Type	0 0 0	xxxx	0 0 1 1	xxx
BEQ/BNE	0 0 1	xxxx	1 0 1 1	xxx
R-Type	0 1 0	0 0 0 0	0 0 0 0	**AND**
	0 1 0	0 0 0 1	0 0 0 1	**OR**
	0 1 0	0 0 1 0	0 0 1 0	**NAND**
	0 1 0	0 0 1 1	0 0 1 1	**NOR**
	0 1 0	1 0 1 1	0 1 0 0	**ADD**
	0 1 0	0 1 0 0	1 1 0 0	**SUB**
	0 1 0	0 1 0 1	0 1 0 1	**SLL**
	0 1 0	0 1 1 0	0 1 1 0	**SRL**
	0 1 0	0 1 1 1	0 1 1 1	**SLT**
ADDi	0 1 1	xxxx	0 1 0 0	**ADD**
SUBi	1 1 1	xxxx	1 1 0 0	**SUB**

Instruction Examples:

R-TYPE

Instructions	Actions	22 Bits Instruction	HexaCode
ADD	R3=R2+R1	0000 010 001 011 00000 0100	011604
SUB	R3=R2-R1	0000 010 001 011 00000 1100	01160c
NAND	R3=(R2^R1)'	0000 010 001 011 00000 0010	011602
NOR	R3=(R2||R1)'	0000 010 001 011 00000 0011	011603
AND	R3=R2^R1	0000 010 001 011 00000 0000	011600
OR	R3=R2||R1	0000 010 001 011 00000 0001	011601
SLT	R3=R1<R2	0000 001 010 011 00000 0111	00a607
SLL	R3=R1<<01	0000 001 000 011 00001 0101	008615
SRL	R3=R1>>01	0000 001 000 011 00001 0110	008616

I-TYPE

Instructions	Actions	22 Bits Instruction	HexaCode
LW	M[010] to R2	0011 000 010 0000 0000 0010	0c2002
SW	R1 to M[02]	0100 000 001 0000 0000 0010	101002
BEQ	If(R2==R1)	0111 010 001 0000 0000 0010	1d1002
BNE	If(R2!=R1)	1000 010 001 0000 0000 0010	211002
Addi	R1=R1+3	0101 001 001 0000 0000 0011	149003
Subi	R1=R1-1	0110 001 001 0000 0000 0001	189001

J-TYPE

Instructions	Actions	22 Bits Instruction	HexaCode
JUMP	Jump to Address 000	1111 000 000 0000 0001 1000	3c0000

Operation Map: For Simulation

ROM	Instruction	Actions	22 Bit instructions	HexaCodes
1	ADDi	R1=R0+6	0101 000 001 0000 0000 0110	141006
2	SW	R1 to M[02]	0100 000 001 0000 0000 0010	101002
3	LW	M[010] to R2	0011 000 010 0000 0000 0010	0c2002
4	SUBi	R1=R2-3	0110 010 001 0000 0000 0011	191003
5	SW	R1 to M[01]	0100 000 001 0000 0000 0001	101001
6	AND	R3=R2^R1	0000 010 001 011 00000 0000	011600
7	SW	R3 to M[03]	0100 000 011 0000 0000 0011	103003
8	OR	R3=R2||R1	0000 010 001 011 00000 0001	011601
9	SW	R3 to M[04]	0100 000 011 0000 0000 0100	103004
A	NAND	R3=(R2^R1)’	0000 010 001 011 00000 0010	011602
B	SW	R3 to M[05]	0100 000 011 0000 0000 0101	103005
C	NOR	R3=(R2||R1)’	0000 010 001 011 00000 0011	011603
D	SW	R3 to M[06]	0100 000 011 0000 0000 0110	103006
E	ADD	R3=R2+R1	0000 010 001 011 00000 0100	011604
F	SW	R3 to M[07]	0100 000 011 0000 0000 0111	103007
10	SUB	R3=R2-R1	0000 010 001 011 00000 1100	01160c
11	SW	R3 to M[08]	0100 000 011 0000 0000 1000	103008
12	SLL	R3=R1<<01	0000 001 000 011 00001 0101	008615
13	SW	R3 to M[09]	0100 000 011 0000 0000 1001	103009
14	SRL	R3=R1>>01	0000 001 000 011 00001 0110	008616
15	SW	R3 to M[0a]	0100 000 011 0000 0000 1010	10300a
16	SLT	R3=R1<R2	0000 001 010 011 00000 0111	00a607
17	SW	R3 to M[0b]	0100 000 011 0000 0000 1011	10300b
18	BEQ	If(R2==R1)	0111 010 001 0000 0000 0010	1d1002
19	Addi	R1=R1+3	0101 001 001 0000 0000 0011	149003
1a	JUMP	CurrentAdd-02	1111 000 000 0000 0001 1000	3c0018

1b	BNE	If(R2!=R1)	1000 010 001 0000 0000 0010	211002
1c	Subi	R1=R1-1	0110 001 001 0000 0000 0001	189001
1d	JUMP		1111 000 000 0000 0001 1011	3c001b
1e	JUMP		1111 000 000 0000 0001 0000	3c0000

Expected RAM Values: Stored Values after the simulation

RAM Address	Stored Values	Actions
000		
001	3	R2
002	6	R1
003	2	AND
004	7	OR
005	3fffd	NAND
006	3fff8	NOR
007	9	ADD
008	3	SUB
009	6	SLL
00a	1	SRL
00b	1	SLT