North South University

Course: CSE 332L
Faculty: TnF
Sec: 02

Report on:

20-bit single cycle CPU

Submitted by:

Name: Md. Sadiqul Islam Khan

ID: 1430334642

Submitted to:

Sheikh Shadab Towqir

(Lab Instructor)

Design of a 20-bit single cycle CPU that can perform R-Type, I-Type and J-Type instruction.

Instruction	Opcode
R-Type	0000
LW	1000
SW	1100
BEQ	1110
BNE	1111
J-Type	1010

Instruction Operation		Function Field
R-Type	Add	1000
	Subtract	1001
	AND	1010
	OR	1011
	NOR	1100
	Shift left	1101
	Shift right	1110
	Set on less than	1111
I-Type	Load word	XXXX
	Store word	XXXX
	Branch if equal	XXXX
	Branch if not equal	XXXX
J-Type	Jump	XXXX

Instruction	RegDst	AluSrc	Mem to Reg	Reg Write	Mem Read	Mem Write	Alu op0	Alu op1	Branch	Jump
R-Type	1	0	0	1	0	0	0	1	0	0
LW	0	1	1	1	1	0	1	0	0	0
SW	0	1	0	0	0	1	1	0	0	0
BEQ/BNE	0	0	0	0	0	0	0	0	1	0
J-Type	0	0	0	0	0	0	0	0	0	1

Instruction opcode	ALU control operation	ALU action	ALU input
R-type	01	And	000
R-type	01	Subtract	001
R-type	01	And	010
R-type	01	Or	011
R-type	01	Nor	100
R-type	01	Shift left	101
R-type	01	Shift right	110
R-type	01	Set on less than	111
Load word	10	And	000
Store word	10	And	000
Branch equal	00	Don’t care	XXX
Branch not equal	00	Don’t care	XXX
Jump	00	Don’t care	XXX

Here every memory block address is 1bit. That’s why going to next address add 1 to program counter.

For R-Type instruction

op(4bit) 19-16	rs(3bit) 15-13	rt(3bit) 12-10	rd(3bit) 9-7	shamt(3bit) 6-4	funct(4bit) 3-0

Example

Add instruction: rd = rs + rt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1000	00508

Subtract instruction: rd = rt - rs

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1001	00509

AND instruction: rd = rs AND rt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1010	0050A

OR instruction: rd = rs OR rt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1011	0050B

NOR instruction: rd = rs NOR rt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1100	0050C

Shift left instruction: rd = rt << shamt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	001	1101	0050D

Shift right instruction: rd = rt >> shamt

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	001	1110	0050E

Set on less than instruction: if (rs < rt) rd = 1 else rd = 0

op(4bit)	rs(3bit)	rt(3bit)	rd(3bit)	shamt(3bit)	funct(4bit)	Hex-Code
0000	000	001	010	000	1111	0050F

For I-Type instruction

op(4bit) 19-16	rs(3bit) 15-13	rt(3bit) 12-10	address/immediate(10bit) 9-0

Example

Load word instruction: rt = mem(rs + address/immediate)

op(4bit)	rs(3bit)	rt(3bit)	address/immediate(10bit)	Hex-Code
1000	000	001	0000 0000 11	80403

Store word instruction: mem(rs + address/immediate) = rt

op(4bit)	rs(3bit)	rt(3bit)	address/immediate(10bit)	Hex-Code
1100	000	001	0000 0000 11	C0403

Branch if equal instruction: if (rs = rt) branch

op(4bit)	rs(3bit)	rt(3bit)	address/immediate(10bit)	Hex-Code
1110	000	001	0000 0000 11	E0403

Branch if equal instruction: if (rs != rt) branch

op(4bit)	rs(3bit)	rt(3bit)	address/immediate(10bit)	Hex-Code
1111	000	001	0000 0000 11	F0403

For J-Type instruction

op(4bit) 19-16	address(16bit) 15-0

Example

Jump instruction:

op(4bit)	address(16bit)	Hex-Code
1010	0000 0000 0000 0000	A0000