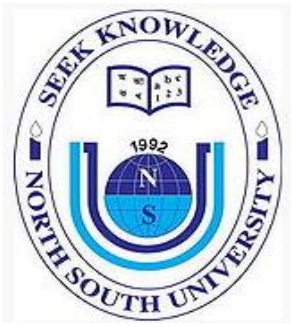


North South University

Department of Electrical & Computer Engineering

CSE332 PROJECT REPORT

To: Sheikh Shadab Towqir

Course Name: CSE332.L

Report Submission Date: 30/04/2018

Section: 02

By:

Tasfia Tahsin Anika 153 0939 642

Sheikh Shakib Zugveri 153 0392 042

Project Objective:

Our objective is to implement a 20-bit single cycle CPU which can perform the following type of instructions:

R-type
I-type
J-type

Instruction Set Architecture:

Rs = First Source Register

Rt = Second Source Register

Rd = Destination Register

Shmt = Shift Distance

R-Type

3 bits		3 bits		3 bits		3bits		5bits		3bits	
19	17	16	14	13	11	10	8	7	3	2	0
Opcode		Rs		Rt		Rd		Shmt		Function	

I-Type

3 bits		3 bits		3 bits		11 bits	
19	17	16	14	13	11	10	0
Opcode		Rs		Rt		Immediate	

J-Type

3 bits		17 bits	
19	17	16	0
Opcode		Address	

TABLES:

Instruction	RegDest	ALUSrc	Mem-toReg	Reg-Write	Mem-Read	Mem-Write	Branch	Jump	ALUOp 1	ALUOp 0
R-format	1	0	0	1	0	0	0	0	1	0
LW	0	1	1	1	1	0	0	0	0	0
SW	0	1	0	0	0	1	0	0	0	0
Beq	0	0	0	0	0	0	1	0	0	1
Bne	0	0	0	0	0	0	1	0	0	1
Jump	0	0	0	0	0	0	0	1	X	X

Fig 1: Control unit signals

Instruction	Control Unit Opcode	ALUOp	Instruction Operation	Funct Field	Desired ALU Action	ALU Control Input
R-type	000	10	Add	000	Add	000
R-type	000	10	Sub	001	Sub	001
R-type	000	10	And	010	And	010
R-type	000	10	Or	011	Or	011
R-type	000	10	Nor	100	Nor	100
R-type	000	10	SLL	101	Shift Left	101
R-type	000	10	SRL	110	Shift Right	110
R-type	000	10	SLT	111	Set on less Than	111
LW	001	00	Load	XXX	Add	000
SW	010	00	Store	XXX	Add	000
Beq	011	01	Branch Equal	XXX	Sub	001
Bne	100	01	Branch not Equal	XXX	Sub	001
Jump	101	XX	Jump unconditional	XXX	XX	XXX

Fig2:Instruction table for ALU control

ALUOp		Function bits			Operation
ALUOp1	ALUOp0	F2	F1	F0	
0	0	X	X	X	000
0	1	X	X	X	001
1	0	0	0	0	000
1	0	0	0	1	001
1	0	0	1	0	010
1	0	0	1	1	011
1	0	1	0	0	100
1	0	1	0	1	101
1	0	1	1	0	110
1	0	1	1	1	111

Fig3:ALU Control Table for ALU Operation

ALU Control Circuit: Operation bits

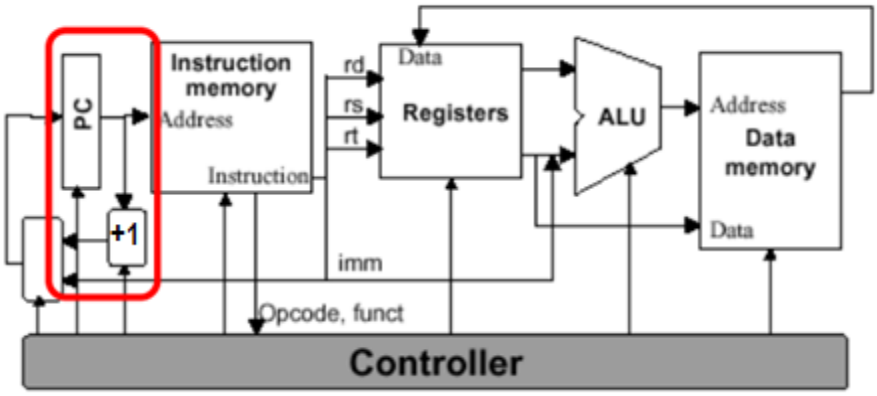
- Op0: F0.ALUOp1 + ALUOp0
- Op1: F1.ALUOp1 + ALUOp1
- Op2: F2.ALUOp1 + ALUOp1

MIPS:

For reading/writing the data segment
Base address plus displacement
Memory address computed as base and offset:
• base is obtained from a register
• offset is given directly as an integer

Program Counter(PC):

Program: a sequence of machine instructions in a text segment



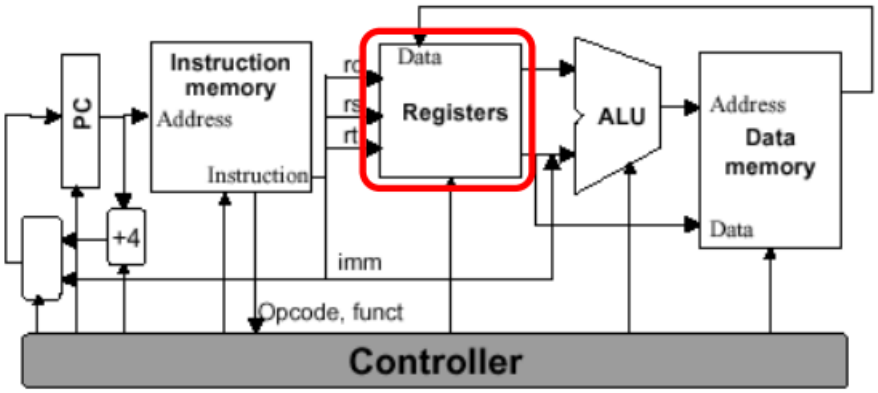
Registers that stores the address of the next instruction to fetch.

Instruction Register(IR):

Register that holds the instruction which is being decoded

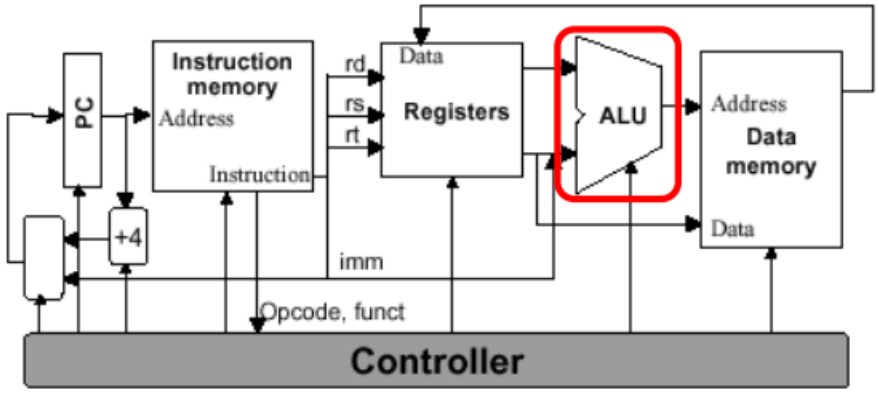
Register:

Component which stores the required bit value



Arithmetic and Logic unit (ALU):

Implements binary arithmetic and binary operations



Control Unit:

Inputs: Condition signals
from IR- decode operation, arguments, result location
from ALU- overflow, divide by zero

Outputs: Control signals
To multiplexers- buses to select
To each register- load new value
To ALU- operation to perform
To all- clock signal

Clock Signal:

Each component is implemented as an electrical circuit
- when inputs change, outputs change
- clock signals ensures that we don't use output until ready
- synchronizes the components in the data path
- the faster the clock, the faster the program will execute

Points:

- In our circuit the program counter is incremented by 1. Hence no shifting is needed in case of jump and branch operations.

- For both bne and beq, the control unit output is branch, which goes to the AND gate; where the other input of the AND gate is from the OR gate. The inputs of the OR gate are “zero flag” (incase of beq) and NOT “zero flag” (incase of bne).

- Both the instruction memory and data memory have 20 bits address and 20 bits data.