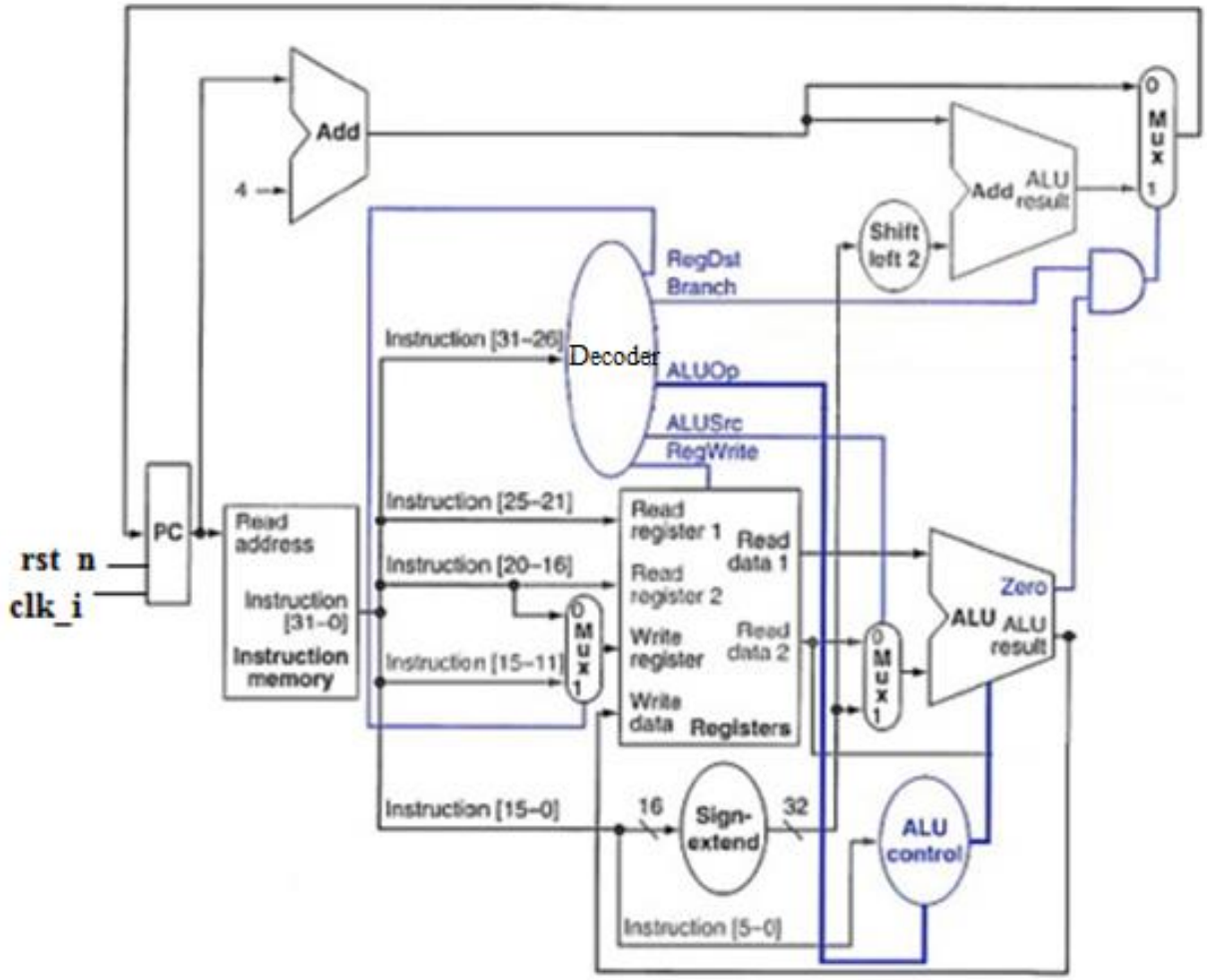
1. Goal

Use the ALU in Lab1 to implement a simple single cycle CPU. CPU is the most important unit in computer system. Please refer to this directions carefully and do the Lab, by the end of the assignment you will have a basic knowledge of CPU.



2. Requirements

a. Please use ModleSim or Xilinx as your HDL simulator.

b. It's a no team assignment. Please attach your names and student IDs as comments in the top of each file. Upload the assignment to E3 with the same format as in lab1 "student ID.rar ". Remember, if you write the report in English you will get some extra points.

c. Program Counter, Instruction Memory, Register File and Test Bench are provided.

The following set of instructions have to be included in your design (75 pts.).

Instr. Type	Instr.	Example	Meaning	Op Field	Function Field
R-type	ADD	add r1, r2,r3	r1 = r2 + 3	000000	100000
I-type	ADDI	addi r1, r2, 100	r1 = r2 + 100	001000	immediate value
R-type	SUB	sub r1, r2, r3	r1 = r2 - r3	000000	100010
R-type	AND	and r1, r2, r3	r1 = r2 & r3	000000	100100
R-type	OR	or r1, r2, r3	r1 = r2 | r3	000000	100101
R-type	SLT	slt r1, r2, r3	r1 = r2 < r3 ? 1 : 0	000000	101010
I-type	SLTI	slti r1, r2, 10	r1 = r2 < 10 ? 1 : 0	001010	immediate value
branch	BEQ	beq r1, r2, 25	if (r1 == r2) go to PC+4+100	000100	address

3. Advance Instructions (25 pts.)

You will have to modify the architecture of the basic design above

ALUOp should be extended to 3 bits to implement I-type instructions.

Originally the ALUop contains 2 bits as shown in the textbook : 00->000, 01->001, 10->010

II. Encode shift L/R and LUI instruction by using unused ALU_ctrl.

Put the logical OR of register Rs and the zero-extended immediate into register Rt

Instr. Type	Inst	Example	Meaning	Op Field	Function Field
R-type	SLL	sll r1, r2, 10	r1 = r2 << 10	000000	000000
R-type	SRLV	srlv r1, r2, r3	r1 = r2 >> r3	000000	000110
R-type	LUI	lui r1, 10	r1 = 10 * 2^16	001111	immediate value
I-type	ORI	ori r1, r2, 100	r1 = 2 |100	001101	immediate value
Branch	BNE	bne r1,r2,30	If(r1!=r2) go to PC+4+120	000101	address

5. Test Cases

There are three test patterns. Namely, CO_P2_testdata1.txt ~ CO_P2_testdata3.txt. For the simple test cases you may use the first file. Please change the line in the file "Instr_Memory" whenever you want to use a different test pattern file.

```
$readmemb("CO_P2_test_data1.txt", Instr_Mem)
```

1	2	3
addi r1, r0, 10 addi r2, r0, 4	addi r6, r0, 2 addi r7, r0, 14	ori r10, r0, 1 lui r11, 1

slt r3, r1, r2
beq r3, r0, 1
add r4, r1, r2.
sub r5, r1, r2
	and r8, r6, r7
	or r9, r6, r7
	addi r6, r6, -1
	slti r1, r6, 1
	beq r1, r0, -5
		sll r11,r11,3
		srlv r11,r11,r10
		addi r10,r10,1
		bne r11,r0,-3

final result	final result	final result
r1 = 10, r2 = 4, r3 = 0 r4 = 0, r5 = 6.	r6 = 0, r7 = 14, r8 = 0 r9 = 15	r10 =7 , r11 = 0

These are the test cases included in the files

6. Grading Policy

a. Total score: 110 pts. Cheating will be punished with a null score!

b. Basic score: 75 pts. Advance instructions: 25 pts.

c. Report: 10pts – format as in previous assignments.

d. **Delay: 10%off/day**

7. Deadline

Please upload the assignment to the E3 before 2015/4/12 23:59:59

Put all of .v source files and report into same compressed file.