Computer Organization 104-2

Lab 2: The Arithmetic Logic Unit

Due: 2016/04/25 23:59:59

1. Goal

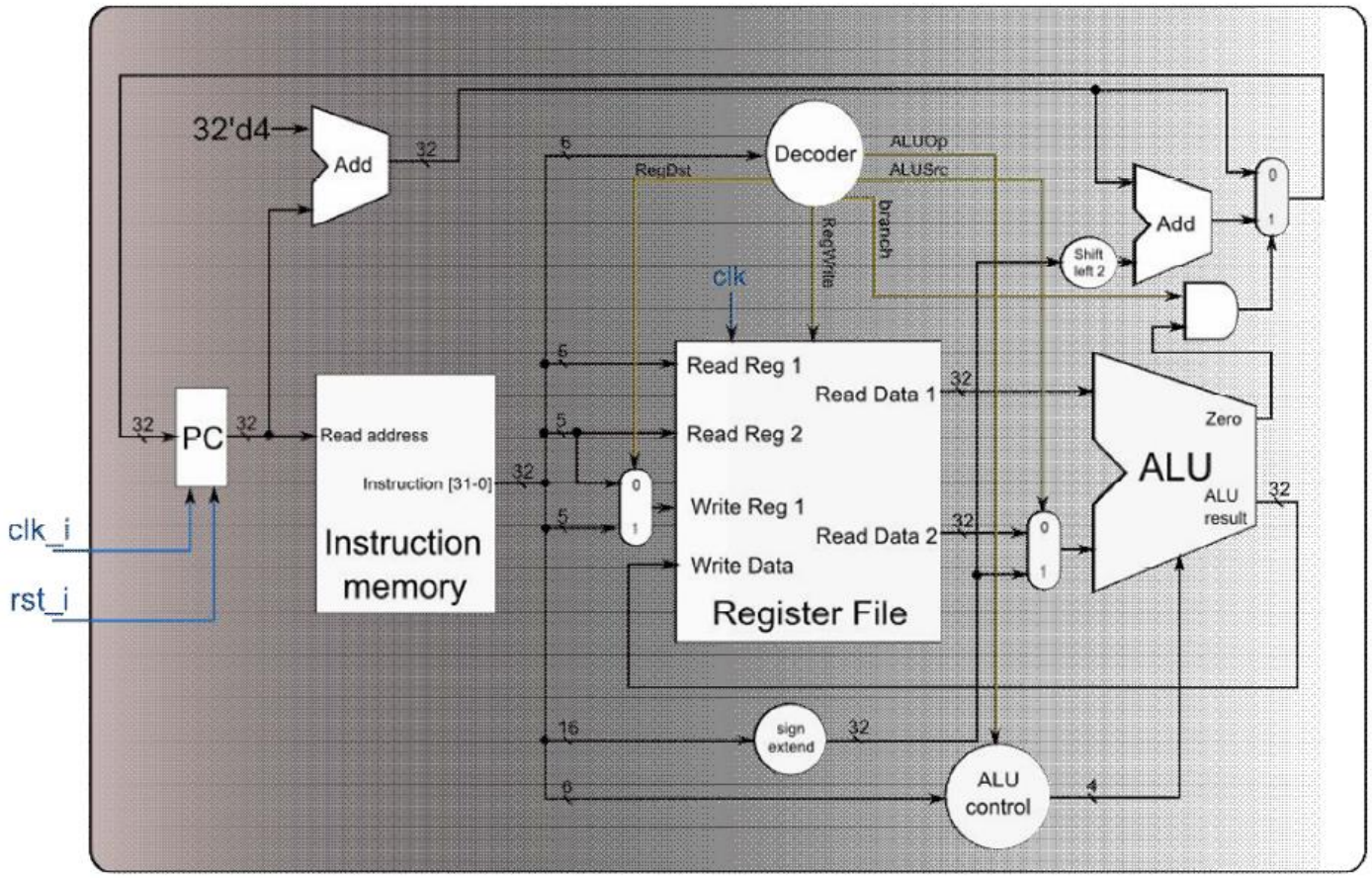
- Utilizing the ALU in Lab1 to implement a simple single cycle CPU. CPU is the most important unit in computer system. Reading the document carefully and do the Lab, you will have elementary knowledge of CPU.

2. Requirement (75%)

- Please use ModleSim or Xilinx as your HDL simulator.
- One people form a group. Just hand in one assignment for one group.
  - Please attach your names and student IDs as comments in the top of each file.
  - The assignment you upload on E3 must have the form of "student ID_LAB2.rar ".
- Program Counter, Instruction Memory, Register File and Test Bench are supplied.
- Instruction set: the following instructions have to running in your designed CPU (75 pts.).

Instruction	Example	Meaning	Op field	Function field
ADD (Addition)	add r1, r2, r3	r1 = r2 + r3	0	32(0x20)
ADDI (Add Immediate)	addi r1, r2,100	r1 = r2 + 100	8	0
SUB (Subtraction)	sub r1, r2, r3	r1 = r2 - r3	0	34(0x22)
AND (Logic And)	and r1, r2, r3	r1 = r2 & r3	0	36(0x24)
OR (Logic Or)	or r1, r2, r3	r1 = r2 | r3	0	37(0x25)
SLT (Set on Less Than)	slt r1, r2, r3	if (r2 < r3), r1 = 1, else r1 = 0	0	42(0x2a)
SLTI (Set on Less Than Immed.)	slti r1, r2, 10	if (r2 < 10), r1 = 1, else r1 = 0	10(0xa)	0
BEQ (Branch On Equal)	beq r1, r2, 25	if (r1 == r2), go to PC+4+100	4	0

3. Architecture Diagram



Top module: Simple_Single_CPU

4. Advanced Instruction

Should modify the architecture of basic design above:

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

- Original 2bits ALUOp from textbook：00->000, 01->001, 10->010

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

- Ex. ALU_ctrl=0 is AND, 1 is OR…, 0 1 2 6 7 &12 are used by basic instructions

Instruction	Example	Meaning	Op field	Function field
SLL (Shift Left Logic)	sll r1, r2, 10	r1 = r2 << 10	0	0
SRLV (Shift Right Logic Variable)	srlv r1, r2, r3	r1 = r2 >> r3	0	6
LUI (Load Upper Immediate)	lui r1, 10	r1 = 10 * 2^16	15(0xf)	0
ORI (Add Immediate)	ori r1, r2, 100	r1 = 2 |100	13(0xd)	0
BNE (Branch On Not Equal)	bne r1,r2,30	If(r1!=r2), go to PC+4+120	5	0

To implement those advanced instructions, please note about the following formats

- SLL Rd, Rt, shamt

- LUI Rt, Imm

- ORI Rt, Rs, Imm

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

5. Test

There are 3 test patterns, CO_P2_test_data1.txt ~ CO_P2_test_data3.txt. The default pattern is the first one. Please change the column 39 in the file “Instr_Memory.v” if you want to test the other cases.

$readmemb("CO_P2_test_data1.txt", Instr_Mem)

The following are the assembly code for the test patterns.

1	2	3

addi r1, r0, 10
addi r2, r0, 4
slt r3, r1, r2
beq r3, r0, 1
add r4, r1, r2.
sub r5, r1, r2
	addi r6, r0, 2
	addi r7, r0, 14
	and r8, r6, r7
	or r9, r6, r7
	addi r6, r6, -1
	slti r1, r6, 1
	beq r1, r0, -5
		ori r10, r0, 1
		lui r11, 1
		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

The file ”CO_P2_Result.txt” will be generated after executing the Testbench. Check your answer with it.

6. Grading Policy

- Total source: 110pts
  - Basic score: 75 pts, Advance instructions: 25 pts, Report: 10 pts.
  - ※ **Any Plagiarism will be punished with a null score!**
- **Delay: 10%off/day**

7. Hand in your assignment

Please upload the assignment to the E3.

Put all of .v source files and report into same compressed file. (Use your student ID to be the name of your compressed file)