Computer Organization, Spring 2017

Lab 2: Single Cycle CPU-Simple Version

Due: 2017/4/27

1. Goal

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

2. HW Requirement

(1) Please use Xilinx ISE as your HDL simulator.

(2) Please attach your names and student IDs as comment at the top of each file.

(3) Please use the Testbench we provide you.

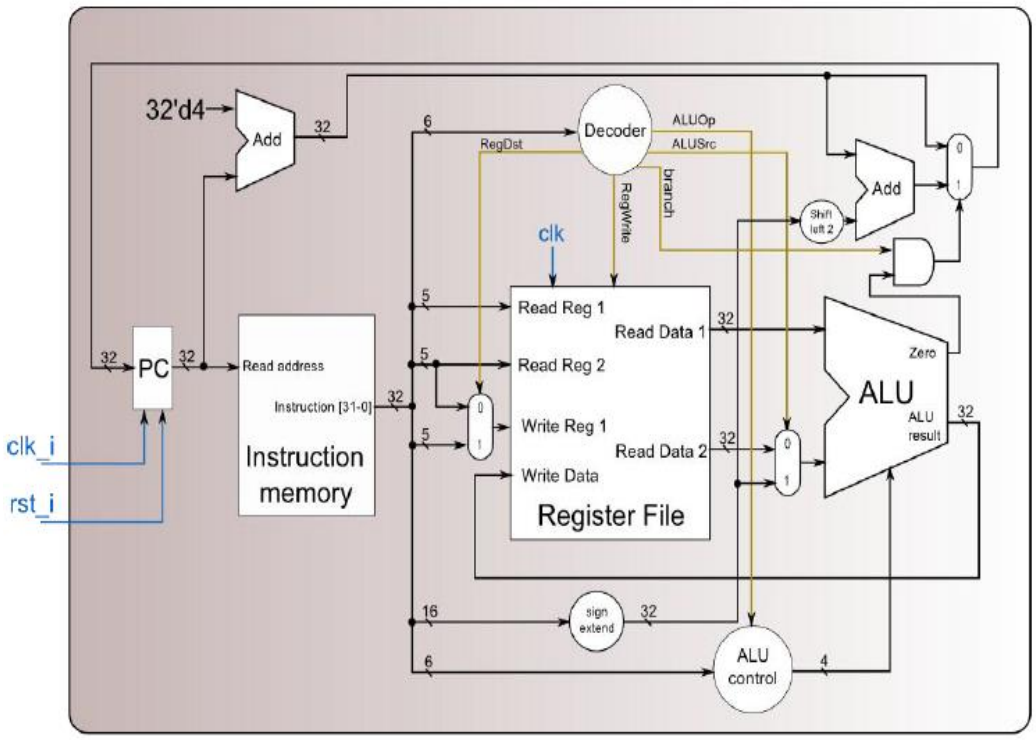
(4) PLEASE FOLLOW THE FOLLOWING RULE!

1. Zip your folder and submit *.zip file.
2. Name the *.zip file with your student IDs (e.g., 0416001_0416002.zip). Other filenames and formats such as *.rar and *.7z are NOT accepted!
3. A team's submissions must be uploaded by the same person.
4. If one violates the rules above, score will be deducted.

(5) Basic instruction set (60%)

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)
SLTU (Set on Less Than Unsigned)	sltu r1,r2,r3	if(r2 < r3) r1 = 1 else r1 = 0	0	43 (0x2b)
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 Instructions (20 pts)

Modify the architecture of the basic design

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

00->000, 01->001, 10->010

(2) Encode shift left and LUI by unused ALU_ctrl

AND(0), OR(1), and etc. 0, 1, 2, 6, 7, and 12 are used.

Instruction	Example	Meaning	Op field	Function field
SLL (Shift Left Logical)	sll r1,r2,10	r1 = r2 << 10	0	0
SLLV (Shift Left Logical Variable)	sllv r1,r2,r3	r1 = r2 << r3	0	4
LUI (Load Upper Immediate)	lui r1,10	r1 = 10 * 2^16	15 (0xf)	0
ORI (Or 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 advanced instructions, please note the following formats.

(1) SLL Rd, Rt, shamt

0	-	Rt	Rd	shamt	0

Shift register Rt left logically by the distance indicated by immediate shamt. Rs is ignored.

(2) SLLV Rd, Rt, Rs

0	Rs	Rt	Rd	shamt	4

Shift register Rt left logically by the distance indicated by register Rs.

(3) LUI Rt, Imm

0xf	0	Rt	Immediate

The immediate value is shifted left 16 bits and stored in Rt

(4) ORI Rt, Rs, Imm

0xd	Rs	Rt	Immediate

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

5. Grade

(1) Total: 100 points (plagiarism will get 0 point)
(2) Document: 20 points
(3) Late submission: 10 points off per day

6. Hand in

Please follow the rules! Zip your folder and name it as "ID1_ID2.zip" (e.g., 0416001_0416002.zip) before uploading to e3. Multiple submissions are accepted, and the version with the latest time stamp will be graded.

7. How to test

The function of testbench is to start tests automatically and output erroneous data. Please put all the .txt files in the project folder. After simulation finishes, you will get some information.

There are 3 test patterns, CO_P2_test_data1.txt ~ CO_P2_test_data3.txt. The

default pattern is the first one. Please change column 39 in the file "Instr_Memory.v" if you want to test the other cases.
Ln39: $readmemb("CO_P2_test_data1.txt", Instr_Mem)

The following are the assembly codes for the test patterns:

1	2	3

addi r1,r0,7
addi r2,r0,-1
sltu r3,r1,r2
beq r3,r0,1
and r4,r1,r3
sub r4,r4,r2
	addi r6,r0,-2
	addi r7,r0,5
	or r8,r6,r7
	addi r9,r0,-5
	addi r9,r9,2
	slt r1,r6,r9
	beq r1,r0,-3
		ori r10,r0,6
		lui r11,2
		sll r11,r11,1
		sllv r11,r11,r10
		addi r10,r10,1
		bne r10,r0,-3

final result	final result	final result
r1 = 7, r2 = -1, r3 = 1 r4 = 2	r1 = 1, r6 = -2, r7 = 5, r8 = -1, r9 = -1	r10 = 8, r11 = 0

8. Q&A

For any questions regarding Lab 2, please contact 林淯晨 (miz1205@gmail.com) and 曾天鴻 (eric830303@gmail.com)