Computer Organization 2015

Lab 3: Single Cycle CPU Ⅱ

Due: 2015/04/26 23:59:59

1. Purpose

Based on lab2, adding a memory unit(named the module DM in Simple_Single_CPU). Implement a complete single cycle CPU that can run R-type, I-type, branch, and jump instructions.

2. Requirement

- A. Please use Modelsim or Xilinx as your simulation platform
- B. It's a no team assignment. 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.rar".
- C. Reg_File[29] represents stack point. Please give an initial value to Reg_File[29] as 128, others 0.
- D. Decoder may add the following control signal:
 - BranchType_o
 - Jump_o
 - MemRead_o
 - MemWrite_o
 - MemtoReg_o

- E. Please name the Data_Memory module DM in Simple_Single_CPU.
- F. Basic instruction (60%)

Lab2 instruction + lw \ sw \ mul \ jump

Instruction	op[31:26]	rs	rt	rd	shamt	func
lw	6'b100011	rs[25:21]	rt[20:16]	imm[15:0]		
sw	6'b101011	rs[25:21]	rt[20:16]	imm[15:0]		
mul	6'b000000	rs[25:21]	rt[20:16]	rd[15:11]	5'b00000	6'b011000
jump	6'b000010	Address[25:0]				

i. lw

MemWrite to be 0, MemRead to be 1, RegWrite to be 1

$$Reg[rt] \leftarrow Mem[rs+imm]$$

ii. sw

MemWrite to be 1, MemRead to be 0

$$Mem[rs+imm] \leftarrow Reg[rt]$$

iii. mul

$$Reg[rd] \leftarrow Reg[rs]*Reg[rt]$$

iv. jump

Jump to be 1

G. Advanced instruction (30%)

Instruction	op[31:26]	rs	rt	imm
bgt	6'b000111	rs[25:21]	rt[20:16]	imm[15:0]
bnez	6'b000101	rs[25:21]	5'b00000	imm[15:0]
bgez	6'b000001	rs[25:21]	5'b00001	imm[15:0]

i. bgt (branch greater than)

Branch to be 1

if
$$(rs > rt)$$
 then $PC = PC + 4 + (sign_imm << 2)$

ii. bnez (branch non equal zero)

Branch to be 1

if (rs
$$!= 0$$
) then PC = PC + 4 + (sign_imm << 2)

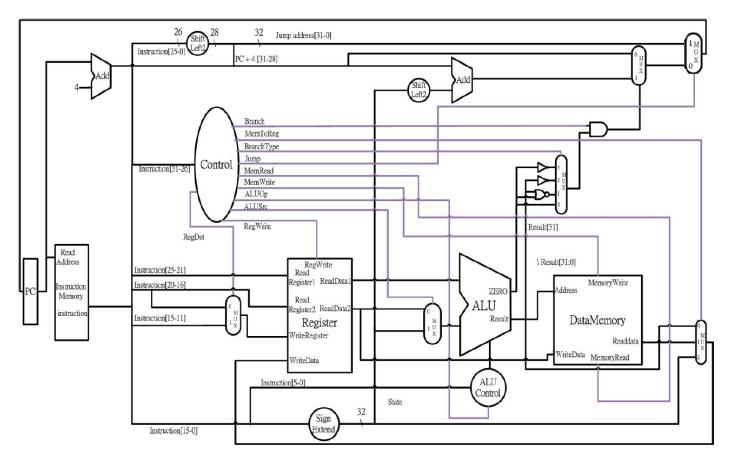
iii. bgez (branch greater equal zero)

Branch to be 1

if
$$(rs \ge 0)$$
 then $PC = PC + 4 + (sign_imm << 2)$

3. Basic Architecture

If you need to use extra control signal, please draw your design to the architecture and descript the implement flow on the report.



4. Bonus (10%)

Instruction	op[31:26]	rs	rt	rd	shamt	func		
jal	6'b000011	Address[25:0]						
jr	6'b000000	rs	0	0	0	6'b001000		

5. Test

A. Basic instruction

TA offers CO_LAB3_test_data1.txt to let you test your design. Refer to test_data1_result.txt to check your result.

B. Advanced instruction

CO_LAB3_test_data2.txt extend the previous part and add more instruction to let your test your design. Refer to test_data2_result to check your result.

C. Bonus

CO_LAB3_test_data3.txt, it is a Fibonacci function. When it done r2 is the answer we want. Refer to test_data3_result to check your result.

D. Please change the line in the file "Instr_Memory" whenever you want to use a different test pattern file

\$readmemb("CA_LAB3_bonus2.txt", Instr_Mem);

6. Grade

- A. Total score: 110 pts. Cheating will be punished with a null score!
- B. Basic instruction (60%)
- C. Advanced instruction (30%)
- D. Bonus (10%)
- E. Report (10%): format as in previous assignments.
- F. Delay: 10% off/day
- G. This Lab needs demo, demo time will be announced.

7. Hand in assignment

- Please submit your file to e3.
- Please compress your source file (not the whole Modelsim project), and report into one single file.
- The file should be named: student_ID.rar