Computer Organization, Spring 2023

Lab 4: Single Cycle CPU II

Due : 2023/05/29

1. Goal

Based on Lab 3 (simple single-cycle CPU), add a memory unit to implement a complete single-cycle CPU which can run R-type, I-type and jump instructions.

2. Demands

A. Please use Vivado as your simulator.

B. “Data_Memory.v”, and “TestBench.v” are supplied. Please add these modules to accomplish the design of your CPU.

C. We have also provided template to assist you in completing Lab4.

D. You can create additional modules (.v files) for your design. However, it is essential to include an explanation in your report for the inclusion of these modules. Failure to mention any additional modules in your report may result in a deduction of 10 points from your grade.

E. If you are using the template provided by the TAs, you may need to modify the following files: 'ALU_Ctrl.v', 'ALU.v', 'Decoder.v', and 'Simple_Single_CPU.v'. Of course, you can modify any other files that haven't been mentioned above to complete your work.

F. Submit all *.v source files and report(pdf) on E3. Other form of file will get -10%.

G. Refer to Lab 3 for top module’s name and IO ports.
Initialize the stack pointer (i.e., Reg_File[29]) to 128, and other registers to 0
Decoder may add control signals:
-Branch_o
-Jump_o
-MemRead_o
-MemWrite_o
-MemtoReg_o

3. Requirement description

A. Basic instruction:

Lab 3 instruction + lw、sw、beq、bne、j

Format:

R-type

Op[31:26]	Rs[25:21]	Rt[20:16]	Rd[15:11]	Shamt[10:6]	Func[5:0]

I-type

Op[31:26]	Rs[25:21]	Rt[20:16]	Immediate[15:0]

Jump

Op[31:26]	Address[25:0]

Definition:

lw instruction :

memwrite is 0，memread is 1，regwrite is 1
Reg[rt] ← Mem[rs+imm]

sw instruction :

memwrite is 1，memread is 0
Mem[rs+imm] ← Reg[rt]

branch instruction :

branch is 1，and decide branch or not by do AND with the zero signal from ALU
beq:

if (rs==rt) then PC=PC+4+ (sign_Imm<<2)

bne:

if (rs!=rt) then PC=PC+4+ (sign_Imm<<2)

Jump instruction :

jump is 1
PC={PC[31:28], address<<2}

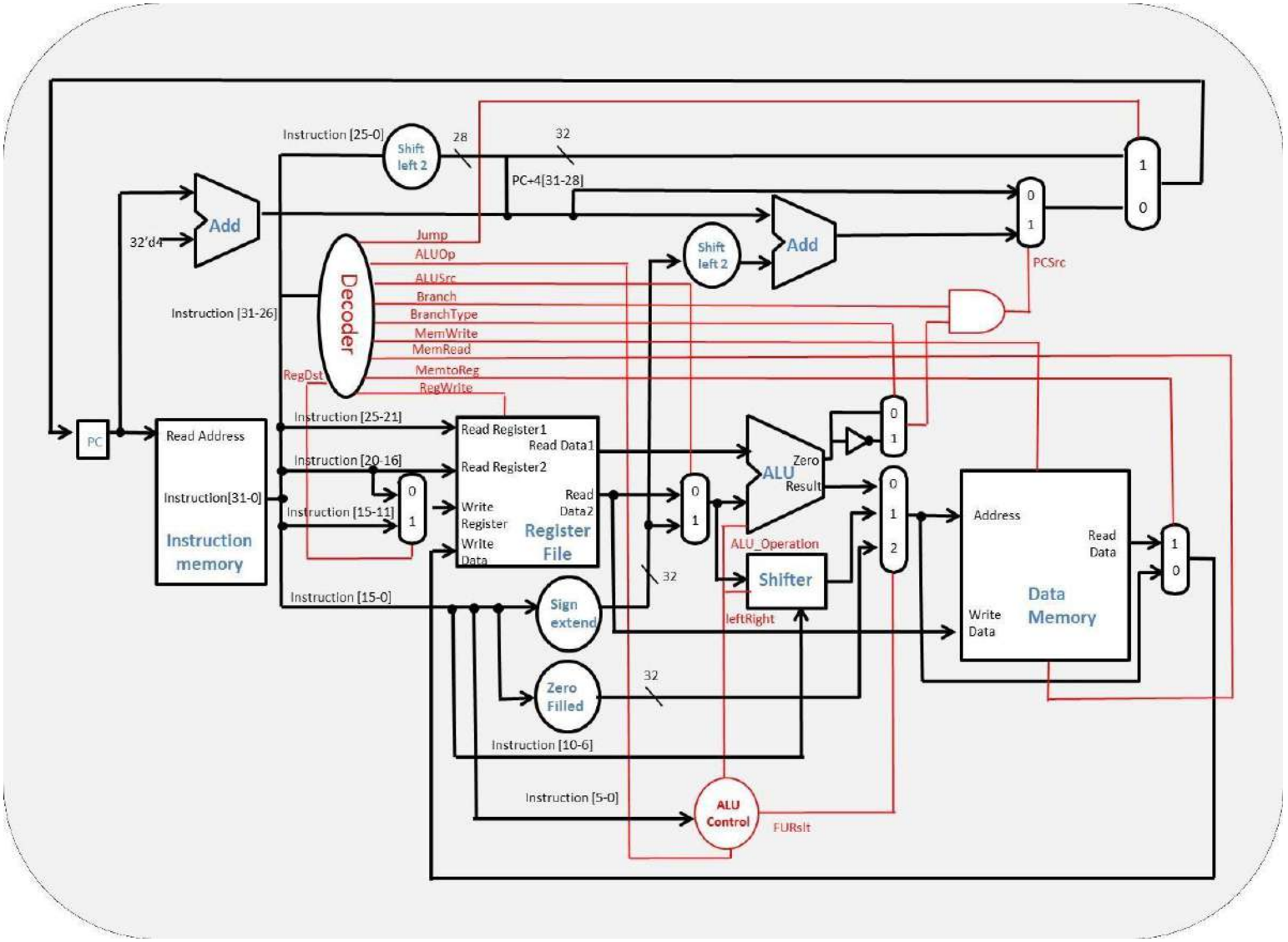
Op field:

instruction	Op[31:26]
lw	6'b100001
sw	6'b100011
beq	6'b111011
bne	6'b100101
jump	6'b100010

Extend ALUOp from 2-bit to 3-bit: (You can modify this if necessary)

instruction	ALUOp
R-type	010
addi	011
lui	101
lw、sw	000
beq	001
bne	110
jump	x

4. Architecture Diagram



5. Test

CO_P4_test_data1.txt tests the basic instructions.

6. Grade

a. Total score: 100pts. COPY WILL GET A 0 POINT!

b. Instruction score: Total 75 pts

c. Report: 25 pts – format is in CO_document. (up to 2 pages)

7. Q&A

If you have any question, just send email to TAs.