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.

1. Requirement
2. Please use Modelsim or Xilinx as your simulation platform
3. 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 ".
4. Reg\_File[29] represents stack point. Please give an initial value to Reg\_File[29] as 128, others 0.
5. Decoder may add the following control signal:

* BranchType\_o
* Jump\_o
* MemRead\_o
* MemWrite\_o
* MemtoReg\_o

1. Please name the Data\_Memory module DM in Simple\_Single\_CPU.
2. 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] | | | | |

1. lw

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

Reg[rt] **←** Mem[rs+imm]

1. sw

MemWrite to be 1, MemRead to be 0

Mem[rs+imm] **←** Reg[rt]

1. mul

Reg[rd] **←** Reg[rs]\*Reg[rt]

1. jump

Jump to be 1

PC={PC[31:28]，address<<2}

1. 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] |

1. bgt (branch greater than)

Branch to be 1

if (rs > rt) then PC = PC + 4 + (sign\_imm<<2)

1. bnez (branch non equal zero)

Branch to be 1

if (rs != 0) then PC = PC + 4 + (sign\_imm<<2)

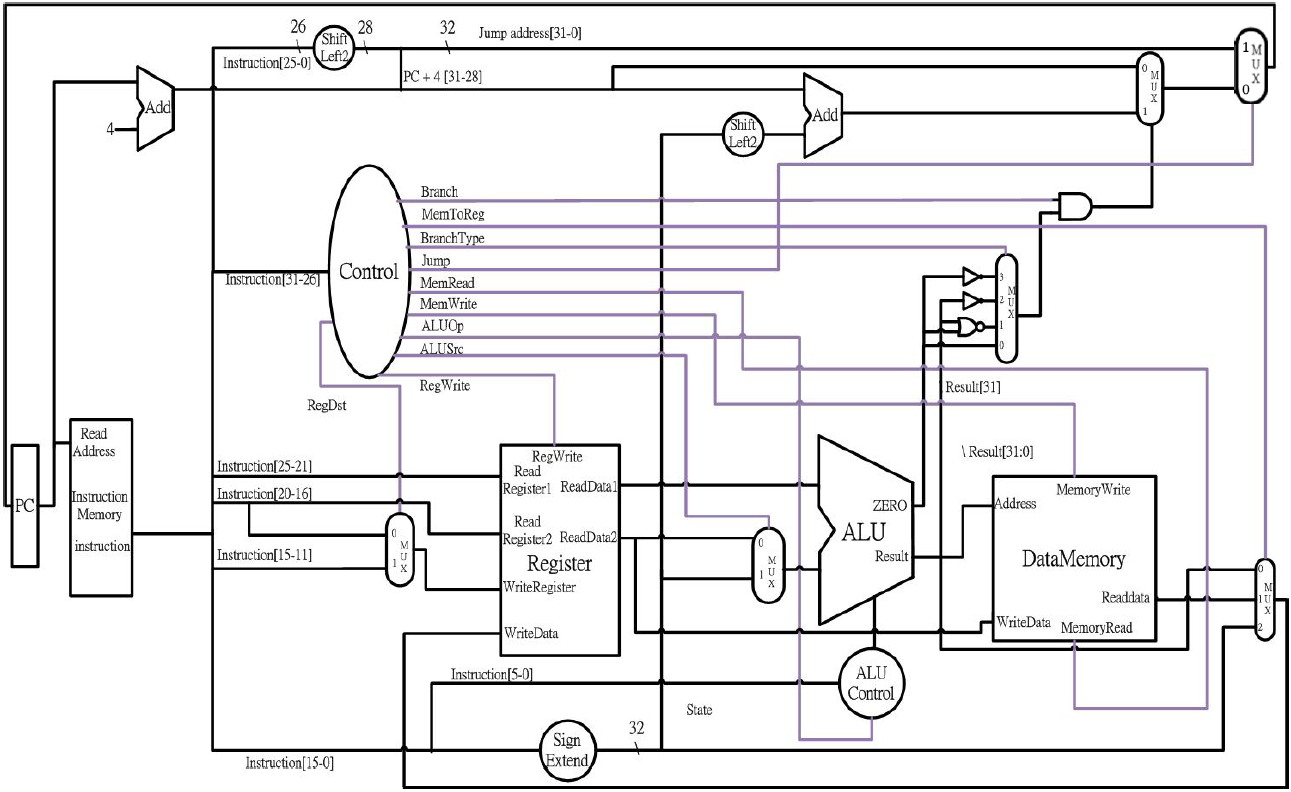
1. bgez (branch greater equal zero)

Branch to be 1

if (rs >= 0) then PC = PC + 4 + (sign\_imm<<2)

1. 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.



1. 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 |

1. Test
2. 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.

1. 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.

1. 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.

1. 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);

1. Grade
2. Total score: 110 pts. Cheating will be punished with a null score!
3. Basic instruction (60%)
4. Advanced instruction (30%)
5. Bonus (10%)
6. Report (10%): format as in previous assignments.
7. Delay: 10% off/day
8. This Lab needs demo, demo time will be announced.
9. 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