Computer Organization

1. The input fields of each pipeline register:

All contain clock and reset.

IF/ID : PC, instruction

ID/EX : PC, register data 1&2, signed extend immediate, all control signals, address for rt & rd

EX/MEM : PC, ALU result, zero, register data 2, control signals for MEM&WB, address for write back

MEM/WB : data read from data memory, ALU result, address for write back, control signals for WB

2. Compared with lab4, the extra modules:

I modified the name of Simple_Single_CPU.v to Pipeline_CPU.v.

I added a module pipeline reg to implement the pipeline register. The inputs will be clock, reset, and the data needed to store, and the outputs will be the stored data. Every data has its input and output port.

Same as Lab 4, I contain ALU1bit and full adder, because I modified from my old code.

3. Explain your control signals in sixth cycle (both test patterns CO_P5_test_data1 and CO_P5_test_data2 are needed):

Picture:

CO_P5_test_data1	CO_P5_test_data2
ALUSrc:0 ALUop:010 RegDst:01 Branch:0, PCSrc:0 MemWrite:0 MemRead:0 RegWrite:1 MemtoReg:00	ALUSrc:1 ALUop:011 RegDst:00 Branch:0, PCSrc:0 MemWrite:0 MemRead:0 RegWrite:1 MemtoReg:00

My code:

Modified at testbench to get the current control signals.

```
$display("ALUSrc:%b",cpu.ALUSrc_ID_EX_o);
$display("ALUop:%b",cpu.ALUop_ID_EX_o);
$display("RegDst:%b",cpu.RegDst_ID_EX_o);
$display("Branch:%b, PCSrc:%b",cpu.Branch_EX_ME_o,cpu.PC_srcs);
$display("MemWrite:%b",cpu.MemWrite_EX_ME_o);
$display("MemRead:%b",cpu.MemRead_EX_ME_o);
$display("RegWrite:%b",cpu.RegWrite_ME_WB_o);
$display("MemtoReg:%b",cpu.MemtoReg_ME_WB_o);
```

4. Problems you met and solutions:

(1) I met the problem that the given graph is different from Lab4, so that when connecting the wire, I get unexpected results.

Sol : I display the data in the wire and find out this problem, then I swap the wire.

(2) For the second clock, I always have 0 at inst_IF_ID_o, and then it will keep calculating and get the wrong answer.

Sol : I let RegWrite and inst_IF_ID_o do logical and (&&) to make sure that inst_IF_ID_o will not be 0.

5. Summary:

This is a fun lab and I learned a lot.