

# Computer Organization

1. The input fields of each pipeline register:

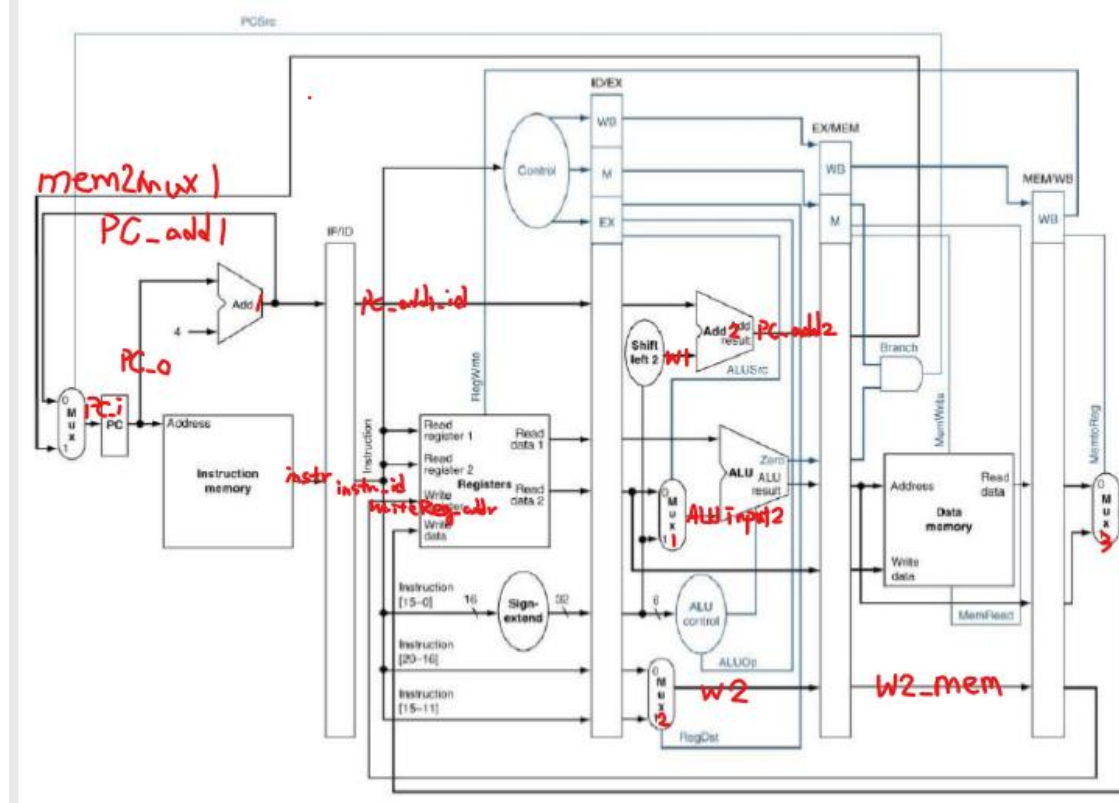
IF/ID: {instr, PC\_add1}

ID/EX: {ReadData1, ReadData2, signextend, PC\_add1\_id, ALUOP, RegDst, MemtoReg, RegWrite, ALUSrc, Branch, MemWrite, MemRead, instr\_id[20:0]}

EX/MEM: {PC\_add2, ALUResult, ReadData2\_ex, zero, w2, MemtoReg\_ex, RegWrite\_ex, Branch\_ex, MemRead\_ex, MemWrite\_ex}

MEM/WB: {ALUResult\_mem, DM\_ReadData, w2\_mem, MemtoReg\_mem, RegWrite\_mem}

## A. Architecture Diagram



2. Compared with lab4, the extra modules:

Pipe\_Reg: the register for storing each stage

Pipe\_CPU: combine all modules together

3. Explain your control signals in **sixth cycle** (both test patterns

C0\_P5\_test\_data1 and C0\_P5\_test\_data2 are needed):

Picture:

C0_P5_test_data1		C0_P5_test_data2	
Name	Value	Name	Value
> handle[31:0]	XXXXXXXX	> handle[31:0]	XXXXXXXX
> count[31:0]	00000006	> count[31:0]	00000006
> instr_op_i[5:0]	00	> instr_op_i[5:0]	23
> ALUOp_o[2:0]	2	> ALUOp_o[2:0]	0
ALUSrc_o	0	ALUSrc_o	1
RegWrite_o	1	RegWrite_o	0
> RegDst_o[1:0]	1	> RegDst_o[1:0]	0
> MemtoReg_o[1:0]	0	> MemtoReg_o[1:0]	0
Jump_o	0	Jump_o	0
Branch_o	0	Branch_o	0
BranchType_o	0	BranchType_o	0
MemWrite_o	0	MemWrite_o	1
MemRead_o	0	MemRead_o	0
RegWrite_o1	1	RegWrite_o1	0
> ALUOp_o1[2:0]	2	> ALUOp_o1[2:0]	0
ALUSrc_o1	0	ALUSrc_o1	1
> RegDst_o1[1:0]	1	> RegDst_o1[1:0]	0
> MemtoReg_o1[1:0]	0	> MemtoReg_o1[1:0]	0
Jump_o1	0	Jump_o1	0
Branch_o1	0	Branch_o1	0
BranchType_o1	0	BranchType_o1	0
MemWrite_o1	0	MemWrite_o1	1

#### 4. Problems you met and solutions:

I should remember the wire name that I created. I have debugged for the names for several hours and Vivado is a terrible tool that it won't tell me where I made a mistake on a wrong name. Next time, I should write on VScode and simulate on Vivado.

## 5. Summary:

This lab is a nightmare for me. Although this is the last homework for this class, I feel that I 'm unfamiliar with Vivado. But in the better part, the coding homeworks let me learn more about how a computer works and be more organized.