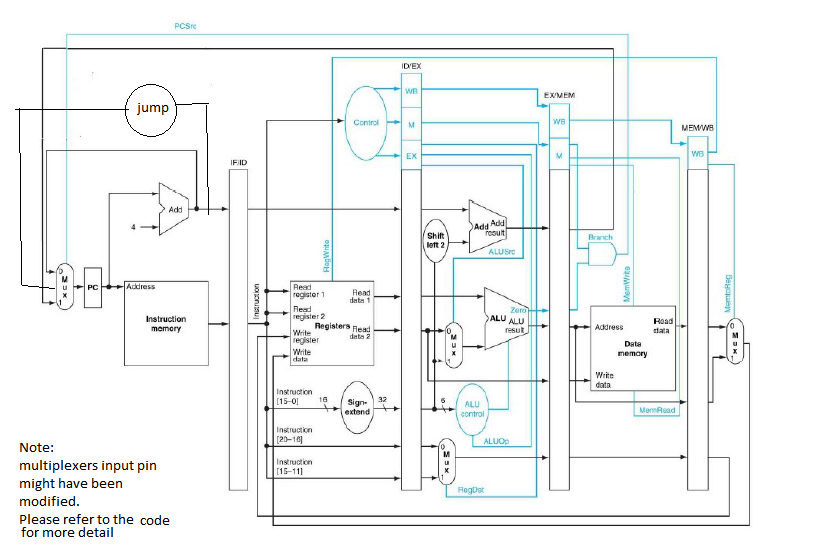
**Computer Organization**

**Architecture diagrams:**



**Hardware module analysis:**

**The modules used in this lab (LAB 04) are identical with the previous LAB.**

**Nonetheless, we add a lot registers functioning as the pipeline registers and also a lot of wires are added to connect the pipeline registers to the modules.**

**In extra to this LAB requirement, we did implement the jump instruction as shown in the architecture diagram.**

**Experiment result:**

**We successfully implemented the basic part and the advanced parts.**

**Problems you met and solutions:**

**The most problematic hitch was tracing the flow in the pipeline. It’s not easy to trace the data by observing the waves and therefore we had a hard time debugging.**

**Moreover, there was little tricky problem when the instruction needs to write to register and read the register at the same clock. It might read the old data (before write instruction).**

**Solution: write to register done in positive edge while read register done in negative edge.**

**Summary:**

**0016328江振皓: Through this lab, I had a better understanding of pipelining and learnt various skills of debugging with the aid of the waves. I also learnt how to co-work with other people in a project.**

**0016110 洪茂榮:**

**This lab brought me a comprehensive practical understanding of pipelining concept as well as data dependency problem. One thing that I learn from this lab is that pipelining requires more registers at each intermediate phase.**

**Job distribution:**

**Basic – 0016110 洪茂榮、江振皓**

**Advanced 1 – 0016110 洪茂榮**

**Advanced 2 – 0016328 江振皓**

**Additional Reports:**

**Advanced Part 1**

Dependentless Assembly code

I1: addi $1, $0, 16

I10: addi $9, $0, 100

I3: addi $3, $0, 8

I4: sw $1, 4($0)

I8: addi $7, $1, 10

I2: addi $2, $1, 4

I5: lw $4, 4($0)

I7: add $6, $3, $1

I9: and $8, $7, $3

I6: sub $5, $4, $3