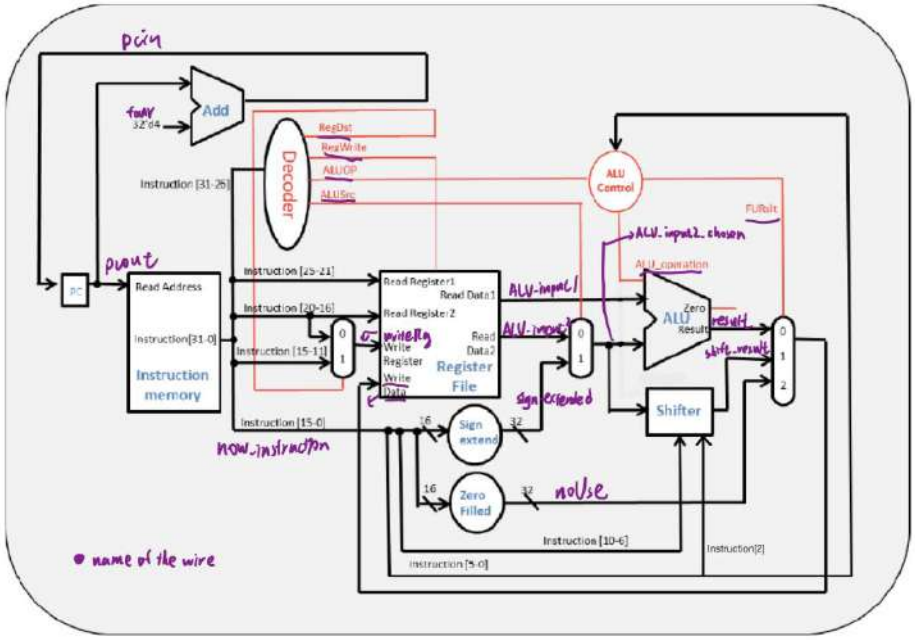
Computer Organization

Architecture diagrams:

Only a few modifications on the diagram given by Tas (output of ALU_ctrl and input of shifter)



Hardware module analysis:

Every instruction I implemented can be divided into 4 stages (IF, ID, EX, WB). The instruction fetched is decided by the program counter and the instruction memory. Instrution is decoded by the decoder and sent into the register to get the values in ID stage. Zero extension is also done in ID stage by assigning 1s or 0s according to the most significant bit of Instruction [15:0]. ALU and shifter are used to compute the values in EX stage. The result of the data is sent back to the register in WB stage.

Since we did not implement load word and save word in this lab, there is no instruction that needs to use data memory, so the architecture diagram is different from the CPU diagram we saw in class. There are also several architecture reduced since we can implement all the funciton we want in this lab.

Just to mention, the zero filled is not used in the lab, but it is still in my code, so I still put it in the architecture diagram.

Finished part:

I finished the design of ADD, SUB, AND, OR, NOR, SLT, SLL, SRL, ADDI intructions by designing the adder to add 4, the decoder, the ALU control, the 2-1 MUX, the 4-1 MUX (only 3 input), the sign extension module, and wire all the module mentioned above in “Simple_Single_CPU.v”.

Problems you met and solutions:

I used the ALU given by TAs and I set the operation AND to be 00 and operation OR to be 01, which is not the same as the design in ALU_1bit.v, it took me some time to find the design mismatch and modify the ALU_1bit.v.

Summary:

I utilizied the ALU in Lab2 to implement a simple single cycle CPU (without a few functions). After finishing this Lab, I now have some elementary knowledges of CPU.