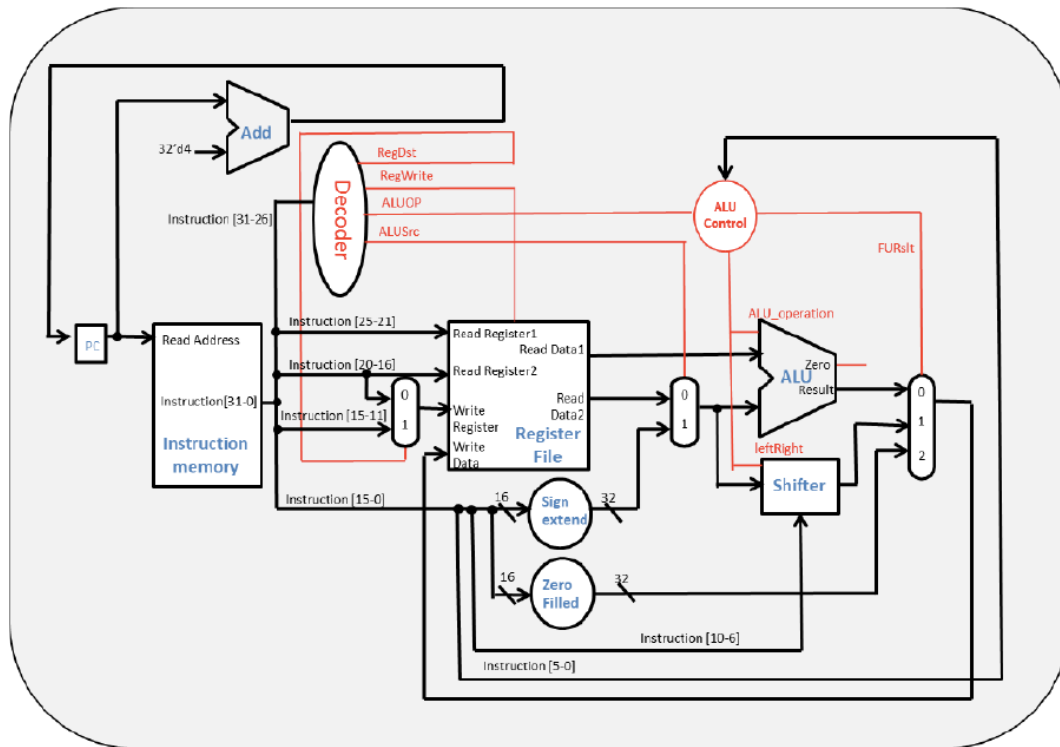


Computer Organization

Architecture diagrams:



我拉的線和 spec 完全相同

Hardware module analysis:

Adder.v: 將兩筆 32 位元的 input 相加並輸出。

ALU_Ctrl.v: 依據 instruction 的 functional field 和 ALUOp 決定 ALU、shifter、multiplexer 的動作。

Decoder.v: 依照 instruction 的 opcode 輸出正確的 control signal。

Multiplexer2to1.v and Multiplexer3to1.v: 實作數位電路 multiplexer 應有的功能。

Sign_Extended.v: 將 16 位元的 input 進行 sign extension 作為 32 位元的 output。

Simple_Single_CPU.v: 將所有 module 的線連起來。

Finished part:

完成的 module: Adder.v, ALU_Ctrl.v, Decoder.v, Multiplexer2to1.v,

Multiplexer3to1.v, Sign_Extended.v, Simple_Single_CPU.v.

通過的測資: "C0_P3_test_data1.txt ~ C0_P3_test_data3.txt"

Problems you met and solutions:

此作業 CPU 的結構與講義有明顯不同，需進行額外的比較與討論以調整 verilog 的結構。其他大多是邏輯上的謬誤，跑測資時發現並解決該問題。

Summary:

在本次的作業中，藉由實做一個簡單的 CPU，以讓我對其結構更加熟悉，順便提升我寫 verilog 的能力。過程雖然花了不少時間，但能實作我所學到的內容感覺頗為充實。