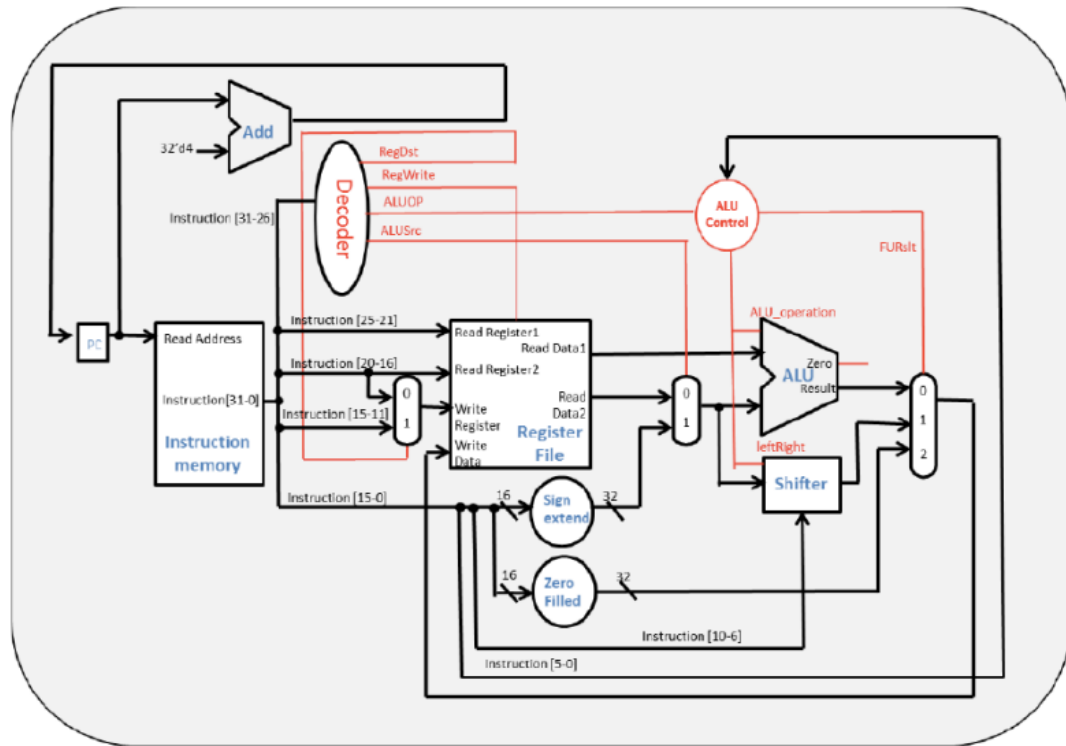


Computer Organization

Architecture diagrams:

3. Architecture Diagram



我使用的跟 Spec 給的一樣

Hardware module analysis:

- Simple_Single_CPU: 根據 Architecture diagram 把所有其他的 module 接線接起來
- Adder: 將兩個 32 bits 相加，主要是給 PC 用
- ALU, ALU 1bit, Full Adder: 使用助教提供的根據 HW2 實作出來的 ALU
- ALU_Ctrl: 透過 {ALUOp_i, funct_i} 合併成 9bit 看 ALU_operation_o 該放入甚麼指令，其中 ALUOp_i 是從 decoder 出來的，funct_i 是根據 Spec 中給的 Function field 設計。FUR_slt 在 sll/srl 中數值改成 1，其他則為 0，以便讓 FUR_slt 接到的 mux 知道現在該輸出 result of ALU 還是 result of Shifter。
- Decoder: 將 instruction[31:26] 輸入進去，如果是 000000 就是 R-type，如果是 001000 就是 I-type。我的設計是根據上課講義的圖將 R-type 的 ALUOp_o 設為 010，但是因為講義沒給 I-type 該設為多少，所以我就將 I-type 的 ALUOp_o 設為 011，如果是其他指令則是 000。至於 RegWrite 則是永遠設為 1，因為

Spec 中給的指令都會動到 Write Register，所以我就將它設成 1 一勞永逸(?)。ALUSrc_o 則是 R-type 設為 0，I-type 設為 1，RegDst_o 則是 R-type 設為 1，I-type 設為 0，這都跟講義的一樣沒有變。

- Instr_Memory: 助教提供
- Mux2to1/Mux3to1: 依 select 選擇要哪個 input，供之後運算使用。
- Program_Counter: 助教提供
- Reg_File: 助教提供
- Shifter: 使用助教提供的根據 HW2 實作出來的 Shifter
- Sign_Extend: 16bits 變成 32bits，其中第 17~32 bit 是根據第 16 bit (最高位元) 來 extend。

Finished part:

根據測試，我的 code 可以跑過所有公開測資

Problems you met and solutions:

我這次實作遇到的最大問題是 ALU 的部分，因為我是拿助教提供的 ALU，但是用的是講義的 ALU 接法，結果講義的 ALU 跟助教提供的 ALU 有些許不同，最大的差異是在 AND 跟 OR 這兩個 gate 放的 mux 不一樣，HW2 的 spec 給的是 00 是 OR，01 是 AND，但是講義的 ALU 完全相反。結果這個就害我在實作 ALU_ctrl 的時候一直錯，不知道為甚麼在跑 1 和 2 測資的時候只要遇到 AND 和 NOR 就會報錯。和同學討論後才發現是 HW2 ALU 跟講義的 ALU 接法不一樣，害我花了 3 個多小時在 debug 😞。

Summary:

這次的 project 學到了怎麼設計 single cycle CPU，雖然大多是照著講義和 Spec 設計，但對如何實作整個 CPU 的運作及架構更加了解，也更加熟悉 verilog 的語法。