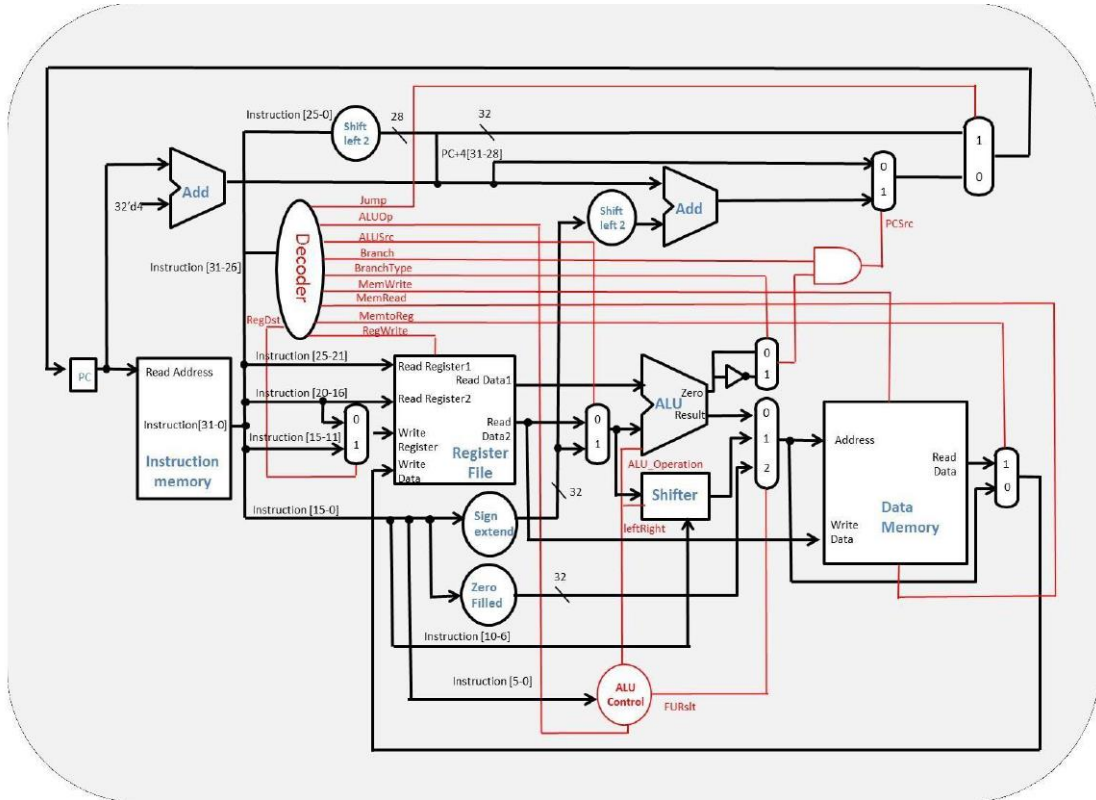


Computer Organization Lab4

1. Architecture diagrams:



因為照著助教給的 lab 3 reference 跟 spec 來做，所以直接使用 spec

中的 architecture diagram

2. Hardware module analysis:

整體上與上次 Lab3 的架構差不多，更動的部分主要在 Decoder 與

ALUCtrl 兩個 module，還有一開始要先把 Reg_File [29] 初始化成 128，跟改

動最後 Simple_Single_CPU 的接線，成功讓這個 CPU 可以成功跑 I-type 跟

branch 的指令。

- Decoder：增加了所需的 control signals，像是 MemWrite、Branch 等，能夠應對不同的指令。
- ALU_Ctrl：對應這次新增的指令，會輸出不同的 ALU_operation 與 FURslt。
- Simple_Single_CPU：這是更動最多的 module，為了成功執行 lw、sw，而增加了 Data_Memory，還需要增加數個 Mux 來處理 branch 與 jump 等指令，最終才能正確執行所有指令

3. Finished part:

成功完成了 basic instruction 跟 advanced set 1 以及 advanced set 2 的要求，可以運行以下指令，並且通過助教所提供的測資

- lw、sw
- beq、bne
- Jump、Jal、Jr
- Blt、bnez、bgez

4. Problems you met and solutions:

雖然完成了所有 set 的要求，但是對於 Simple_Single_CPU 的改動仍是十分困難，而且 decoder 要非常小心寫。像是在實作 advanced set 2 時我一直只拿到 2 分，但是我把可能出現 bug 的地方都仔細看過了，感覺應該沒有接錯線，結果 GTKwave 看了半天才發現自己 decoder 把 bnez 跟 bgez

的 Opcode 打成一樣的，才終於拿到 10 分。

5. Summary:

這次 lab 主要還是延續上次的 single cycle CPU，只是讓整個 CPU 更加完整，最後成品也幾乎跟我們在上課第 4 章所學一樣。整體來說最困難的部分就是要把上次的 Simple_Single_CPU 改成符合這次的 diagram，正好可以結合上課所學，我認為是很不錯的作業，只是要很小心做，一不注意可能會出錯。