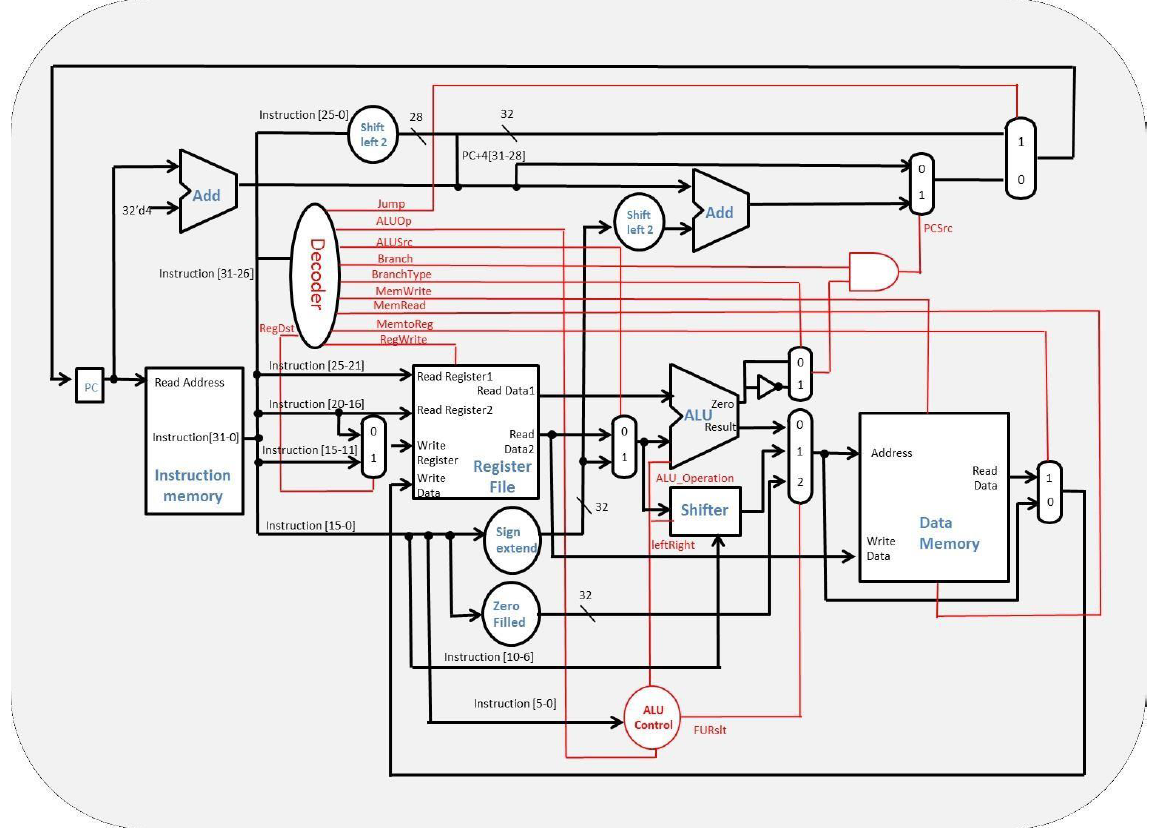
**Computer Organization Lab4**

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

1. **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

1. **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

1. **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。

1. **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，一不注意可能就會出錯。