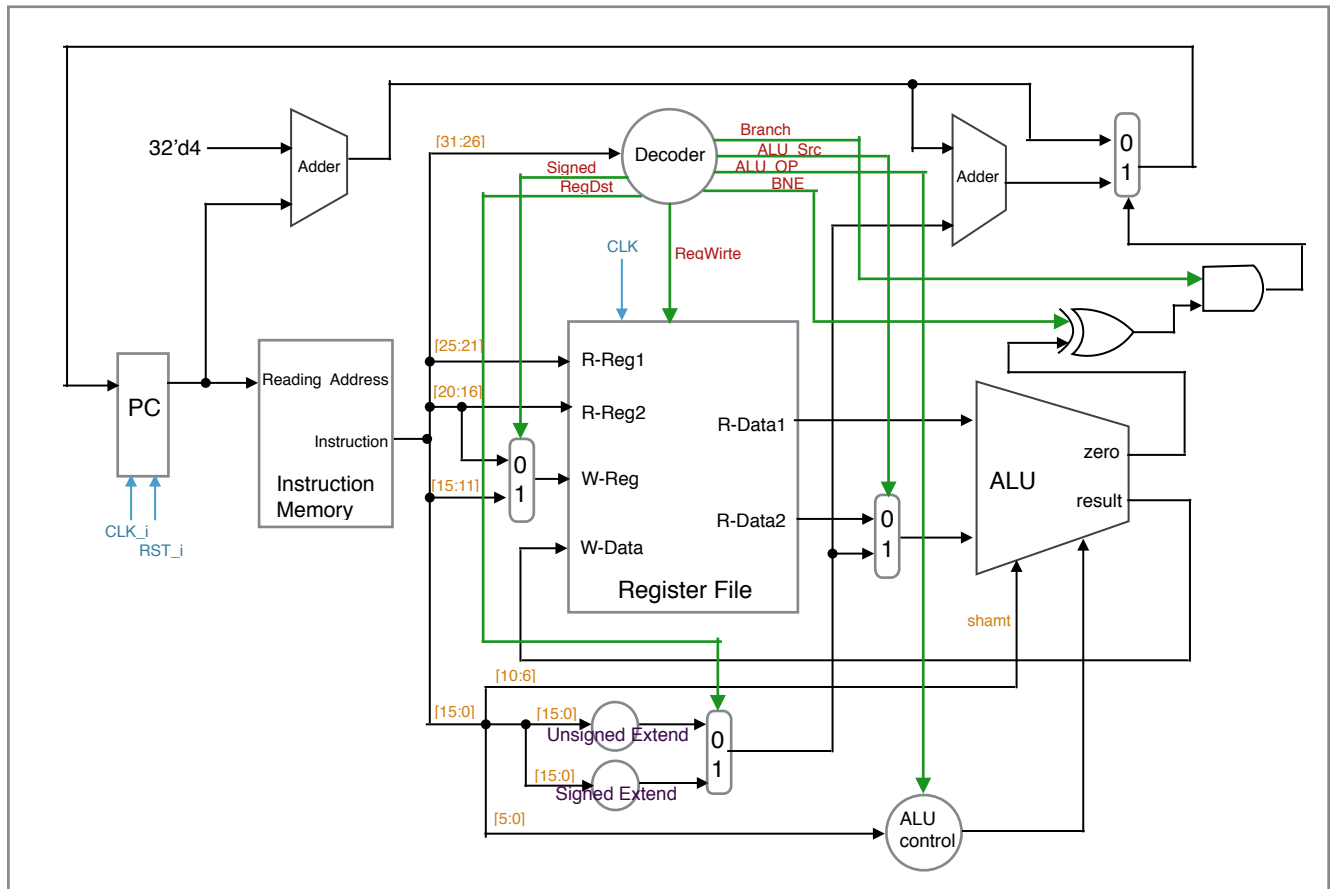


Computer Organization

Architecture diagrams:



Hardware module analysis :

從 Program Counter 開始送出 Address 給 Instruction Memory 讀取程式的指令。之後指令在依據 MIPS 的指令集，各部份的指令分別傳進 Decoder、Extender、ALU controller 或者 Register File。Decoder 負責控制訊號包含：Branch、Signed Extend、Unsigned Extend、一部分 ALU 的控制訊號、暫存器的存取。還有 ALU controller 控制 ALU 的所有動作。

Finished part:

基本指令 ADD, ADDI, SUB, AND, OR, SLT, SLTI, BEQ
進階指令 SLL, SLLV, LUI, ORI, BNE

Problems you met and solutions:

Decoder 和 ALU controller 之間的溝通常常有漏掉的地方。並且要在有限的訊號裡包含所有指令，在寫進階指令的時候發現訊號不夠了，之後整理發現其中幾個指令是可以利用同樣的訊號，才解決這個問題。

Shamt 的處理也碰到一些問題，在 SLL 的時候都是 RT 去做 Shift 所以在 ALU 的時候都是讀到 SRC2，因此不知道該怎麼把 Shamt 的資料送進 ALU，之後有想到要在 ALU 外部去控制 Shift 但是發現要多加一個 controller，所以利用另一個方法去實作，把 ALU 多新增一個 SRC 把 shamt 送進 ALU 來運算。

困難點最主要是進階指令沒有架構圖，有些細節不知道該怎麼實作，或者實作出來不曉得是否符合常理、經濟效益。

Summary:

最大的收穫主要是看訊號在一步一步的拆解運算時候，從各各小 module 完成自己的工作，合作完成一個指令。還有 DEBUG 的過程也滿有趣的，雖然遇到很多困難，但是最後弄懂者整個 CPU 運作的流程後，修正每個 module 的 BUG 才跑出第一個指令，然後看一個指令一個指令的運算，找出有 BUG 的指令，然後再修正，最後算出正確結果有很大的成就感。