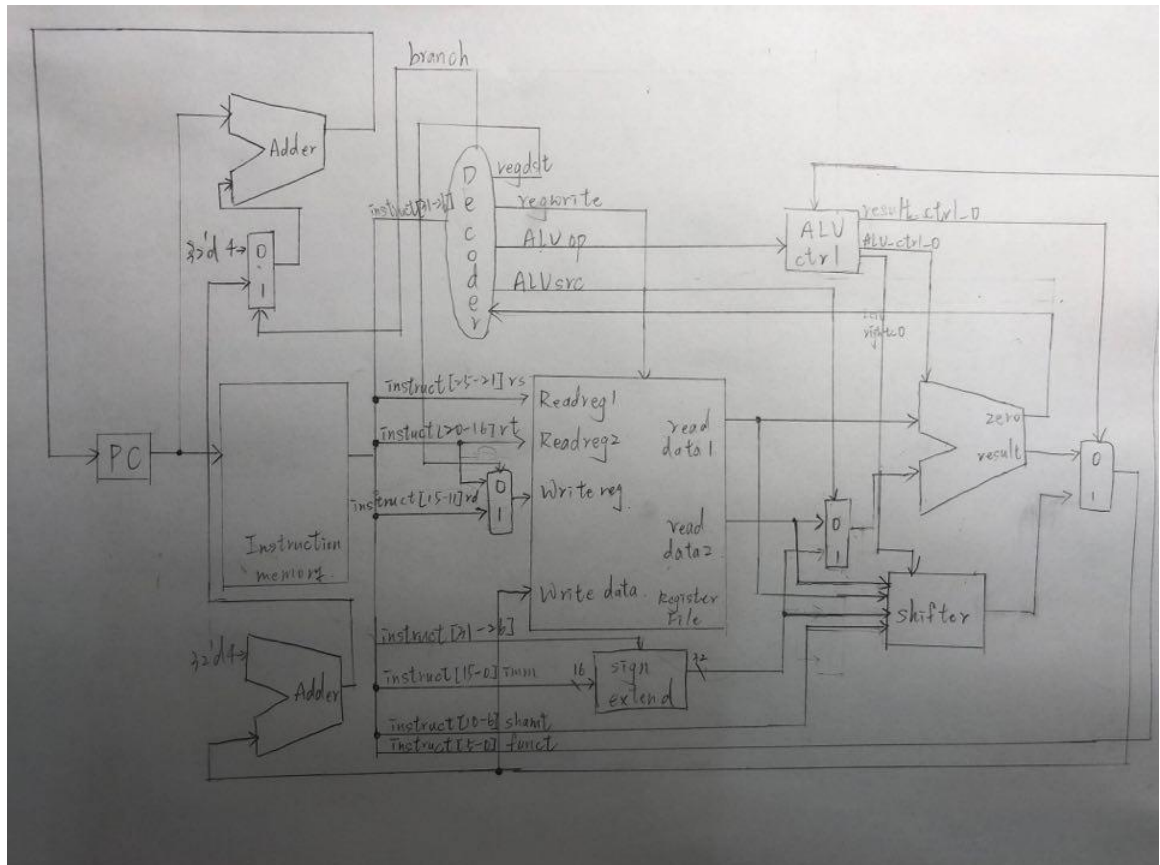


Computer Organization Lab 2 Report

Simplified Single-cycle CPU

- Architecture diagram:



- Module description:

PC: 指出下一個 instruction 的地址給 instruction memory

Instruction memory: 根據地址給出 instruction

Register file: 根據 instruction 的 rs rt 地址輸出 data，將結果 data 寫入 rd

Adder: 將兩個輸入相加然後輸出

Decoder: 依據 Opcode，給出相關資訊讓我們得到正確的 ALU 結果正確的，以及依據 zero 給出 branch 的結果

ALU_ctrl: 依據 decoder 的 ALUop 以及 R-type 的 funct 知道要對 ALU 下何種指令，還有對 shifter 輸入什麼值

ALU: 功能有加減 AND OR STL，STL 對 signed 跟 unsigned 都可以，輸出有兩個一個就是上面方法的結果，另一個是 zero 檢查結果的每一個 bit 是不是都是零，用在 beq bne

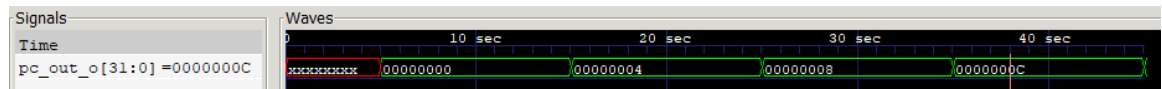
Shifter: 輸入有五個:要如何 shift 的方法，sign extend，read data1，read data2，shamt，可以因應 bne beq sra srav lui 所有情況

Sign extend: 裡面有分成 zero extend 跟 sign extend 前者是不管怎樣都將 16bit 前面加 0，後者是根據第 16bit 去決定要加 0 還是 1

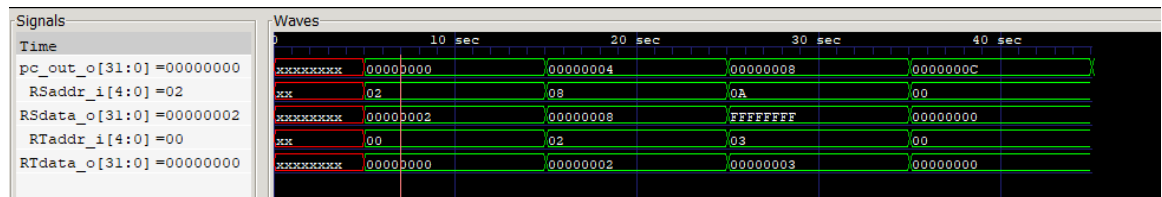
MUX_2to1:將兩個輸入根據有一個決定值去決定給下一個 module 要用什麼值

- Waveform

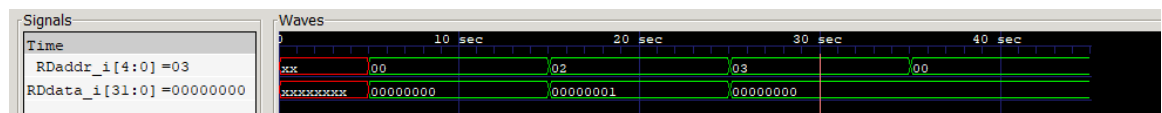
The output of program counter module



rs and rt register selection input and data output



rd register selection input and data input



- Questions

- Input [15:0] input_0 是把 input_0 的 MSB 設定成 input_0[15] LSB 設定成 input_0[0]，而 input [0:15]則是把 input_0 的 MSB 設定成 input_0[0] LSB 設定成 input_0[15]
- always block 的意思是一直監視著 program 裡面的值一有發生變化就會立即改變 always 裡面值的 reg
- by order: 優點是很直覺，根據 module 的輸入輸出順序去填就可以缺點是如果 module 很多的話要記得很多連接的順序是很難的
By name: 優點是不用考慮順序，根據 module 的名稱去填就可以，缺點是比較不直覺沒有按照順序

- Contribution

我負責 sign_extend ALU MUX，另一位負責 decoder alu_ctrl shifter adder 至於 bug fixing 跟 case generation 都是由我們兩個人一起找出的，但應該也沒有說一個 module 是誰去完成的，我們兩個人都會對對方的 module 去做提問以及覺得怎麼樣寫可以更好，最後最終的 Architecture diagram 是由我們兩

個一起想過，通過 **test** 找出哪裡設計錯誤，最後成功的結果，前面的寫各個 **module** 只是最剛開始的分工。

- **Discussions, problem encountered and miscellaneous**

這次的 **LAB** 是第一有接這麼多不同的 **module**，剛開始覺得很難要寫很多 **module**，後來才覺得原來每一個 **module** 所做出的工作其實沒有很難，但全不接再一起卻能有如此強大的功能，這也是第一次程式作業是兩個人一起，感覺到了分工合作的強大吧，自己不用全部完成一件事，反倒是了解別人幫自己做的事，自己也幫別人做的事，最後還蠻快的就完成。