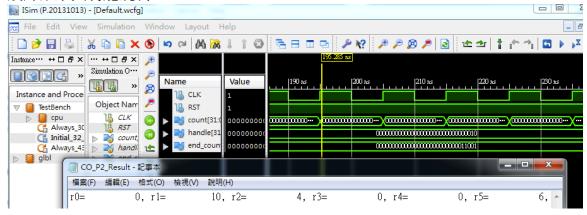
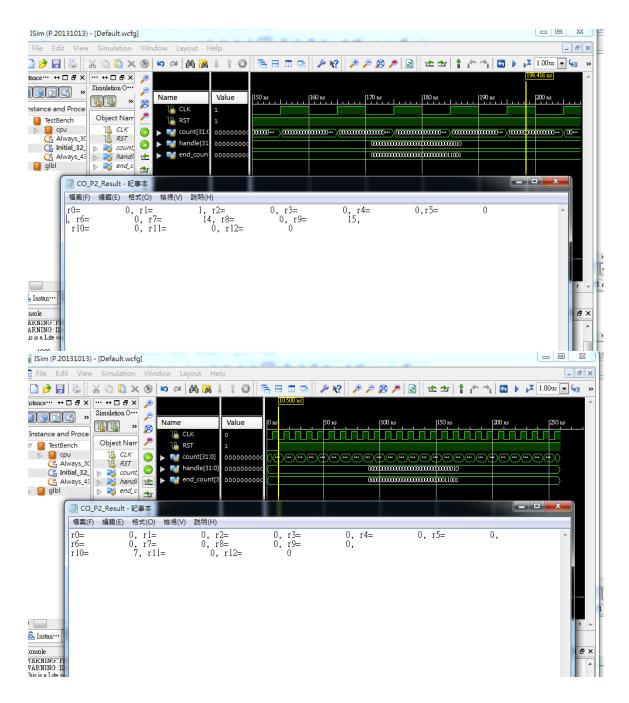
Lab2:simple_cycle_cpu

系統架構:這個cpu.v包含了pc,adder1,instr_memory,reg_file,decoder,ALU_ctrl,sign_extend,mux_ALUSrc,ALU,adder2,shift_left_two,mux_pc_source這幾個子module,而cpu.v裡面接的線是依照作業投影諞髓給的方式來接的。

設計模組分析:pc是依照每一個clk去讀取pc_in,而sign_extend則是把16bits擴大成32bits以防signed值從負號變正號之類的,而decoder比較複雜,則是依照instruct跟functionfield給予的訊號去decode,而ALU_ctrl則是依照decoder所給予的ALUop與instruct的functionfield去判斷要給ALU什麼值,而我是用if的條件判斷式來做的。

設計結果與功能說明:





遭遇困難與解決方法:一開始不知道要按runall才能跑出後面的值,後來問了同學才知道,還有我的sample_cpu.v裡面的branch前後宣告的名稱不一,導致後來的pc_in,pc_out都是xx,debug後就有值了,還有我的instruction沒有值,最後發現是我的sample_cpu.v接錯位子,導致instruction沒有值,而後來我的shamt因為沒有接進去導致我第三筆測資也沒有值,在接上去後就有值了。

作業心得討論:此次的作業雖然花了我不少時間,但是讓我對於cpu的了解又更深了一步,而助教用分module的方式也讓我在debug的時候,不會很難找。