

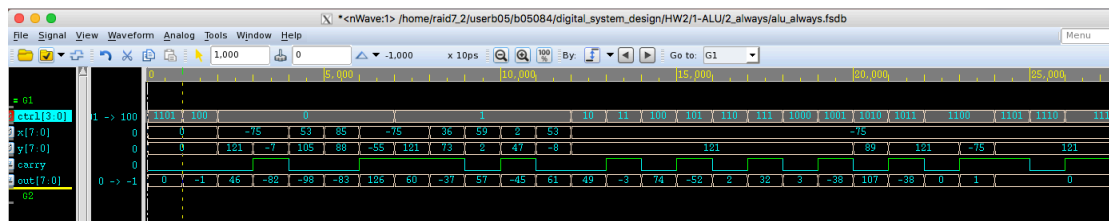
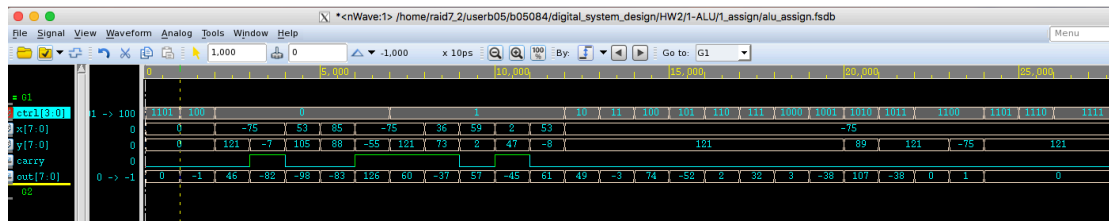
Report

B05901084 電機三 劉容均

1. ALU

testbench:

除了 0000 和 0001 有比較複雜的有號數加減法，其他的指令都是直觀的移位、邏輯運算，因此不需要太多測資驗證，只需一到兩項測資，而在測試加減法時，正負溢位和不會溢位的運算都須考慮，才能驗證 carry 的正確性。



what I found:

大致上沒有太大的問題，只有在 add 和 sub 的 carry 遇到了一些問題，要用邏輯運算算出 carry 其實沒有那麼直觀，所以我多設了 out1[8:0]，直接擴充一個 bit 進行加減，再把 out1[8]給 carry。

2. register

testbench:

因為題目要求隨機產生測資，所以使用\$random()，在第一輪時隨機產生數字寫入 register，並讀取一遍測試結果，第二輪時則先讀取，並將讀取到的值 invert 後寫入，最後再讀取測試結果。

What I found:

我在測試時，發現結果和答案差了一個 cycle，因為 register 和 tb 都是自己寫的，所以會不知道哪個才是對的，要看過 nWave 後，確認 register 符合作業需求，才知道是 tb 不對。

另一個問題是 read 的讀取時間和條件，雖然在這裡看不出影響，但卻會造成下一題的錯誤。題目中的 synchronously 並不是指和 CLK 同步，而是輸入指令時，可同步輸出結果。

3.simple calculator

what I found:

從 tb 觀察，從輸入到輸出的時間並沒有給足夠的 cycle，而且寫入只能在 posedge 進行，所以每個 module 如何連接非常重要。Data_in 經過 mux 再經過 ALU 都是 combinational logic，在 posedge 時寫入，同時可以讀出存入的值。我遇到的第一個問題是我誤以為 read 必須要在 posedge 或 negedge 讀取，後來才知道不用，另一個問題則是在接線時要用 wire 還是 reg，mux 到 ALU 用 reg，用 assign 選擇接收 bus_x 或 data_in，而 ALU 連到 register 的線則使用 wire 即可。