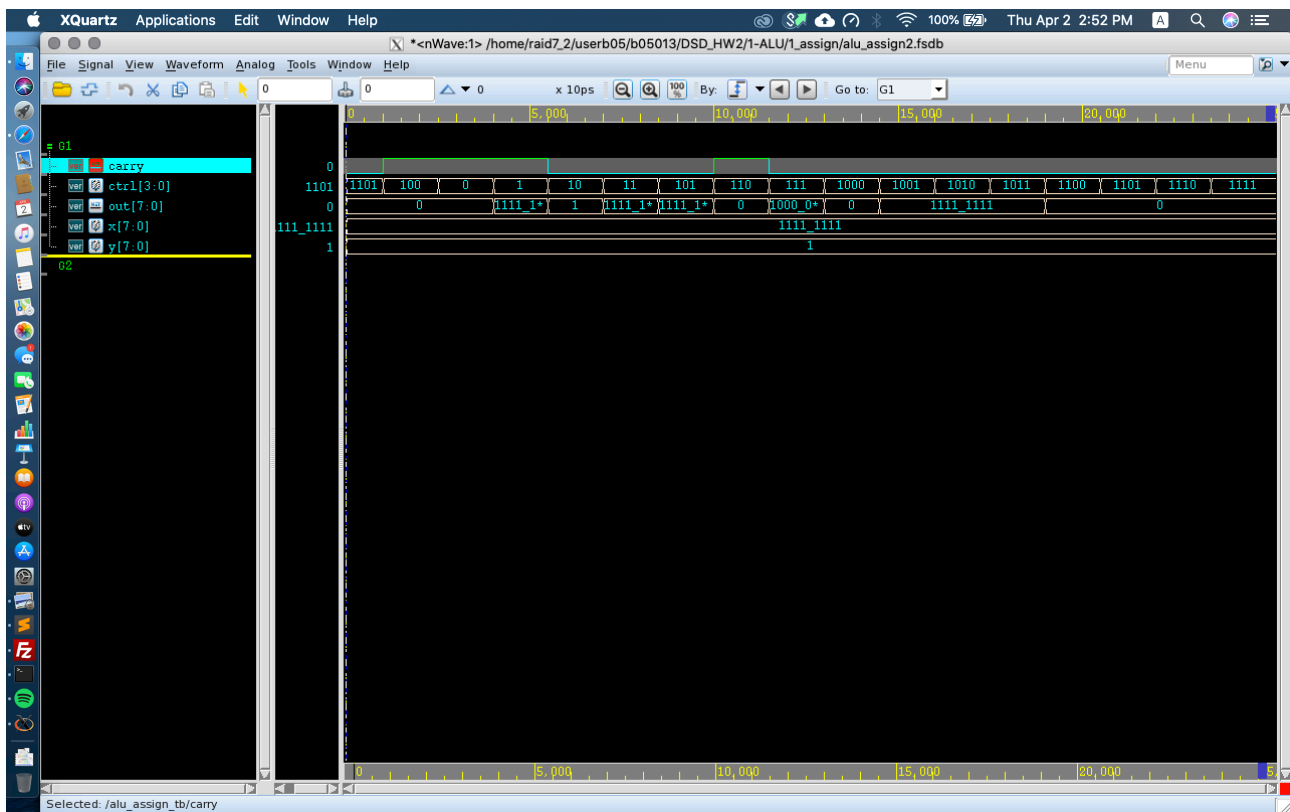
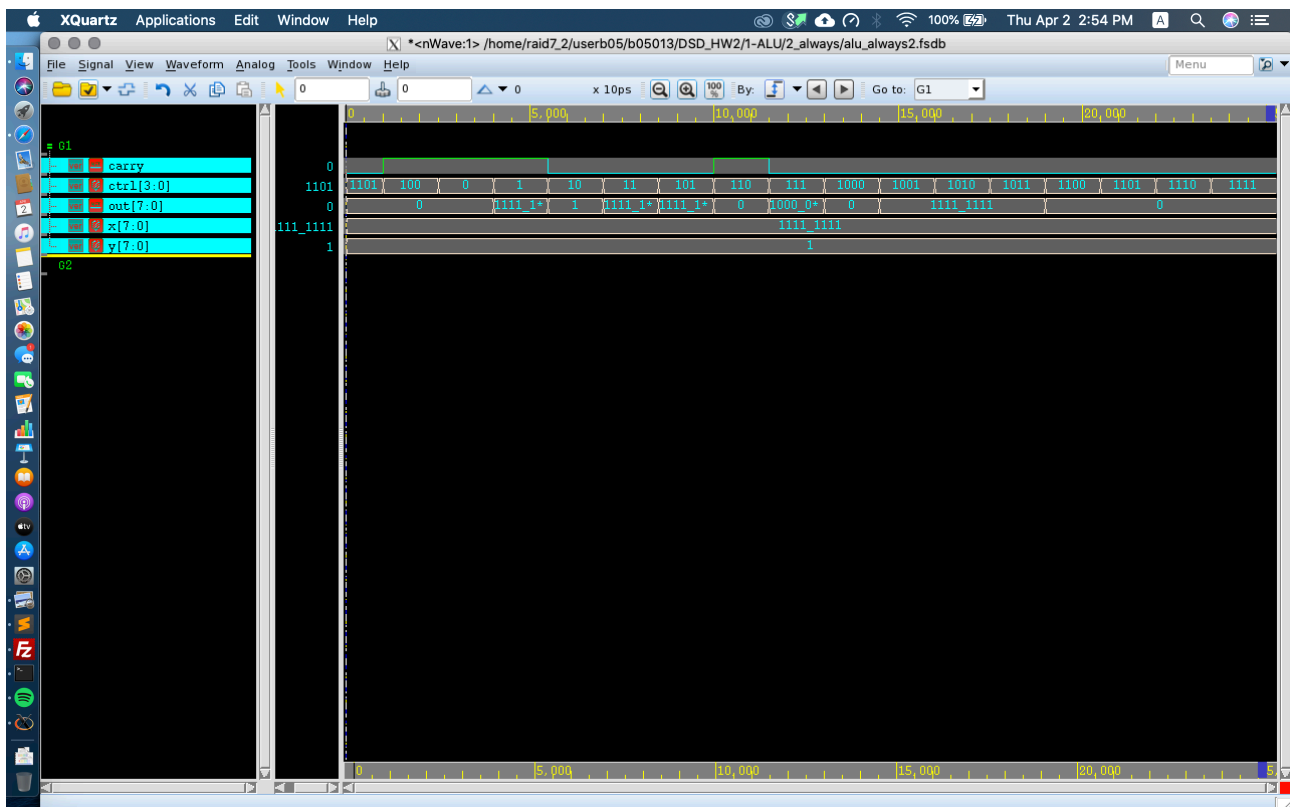


(1) ALU:
alu_assign waveform:

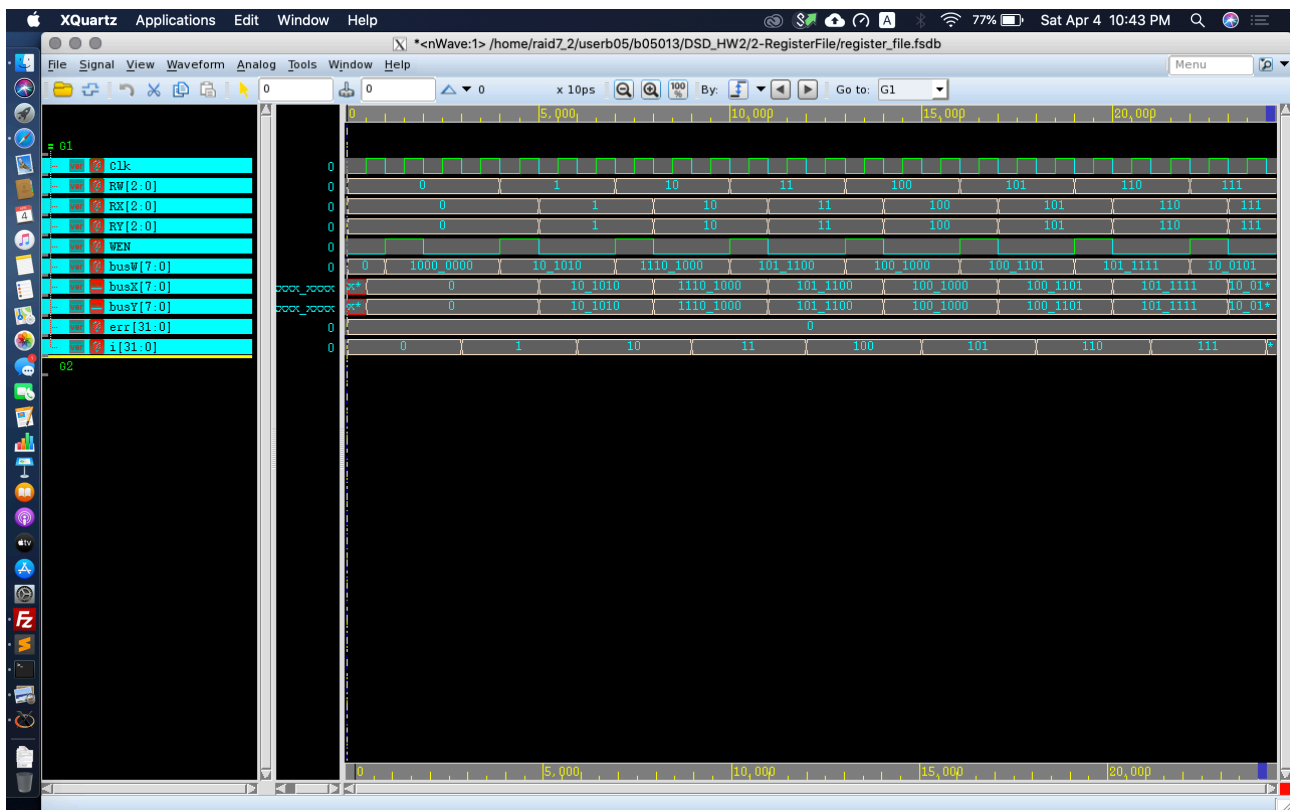


alu_always waveform:



我的testbench寫法是依時序模擬各個指令，並驗證運算結果是否正確。

(2) 8x8 register file:
register_file waveform:



我的busW輸入值用了Verilog中的\$random。用了for-loop模擬了八次讀寫，在第i次cycle時，busW會寫進register[i] (RW = i)，再來輸入RX = RY = i，讀取register[i]的值輸出至busX和busY。

(3) Describe what you've found:

在HW2我原本用的reg是用助教給的檔案原本寫好的(r0_r~r7_r, r0_w~r7_w)，但是在處理r*_w要寫入r*_r的時機總會和simple_calculator_tb.v所模擬的cycle對不太起來(會晚2個cycle進來)，所以最後乾脆讀寫都是在同個reg上面(r[0]~r[7])，於是問題也迎刃而解了。另外，我在寫register_file_tb.v的時候，用了3個@(negedge Clk)，訊號寫入DUT才不會出現8'bx，不太確定原理為何。