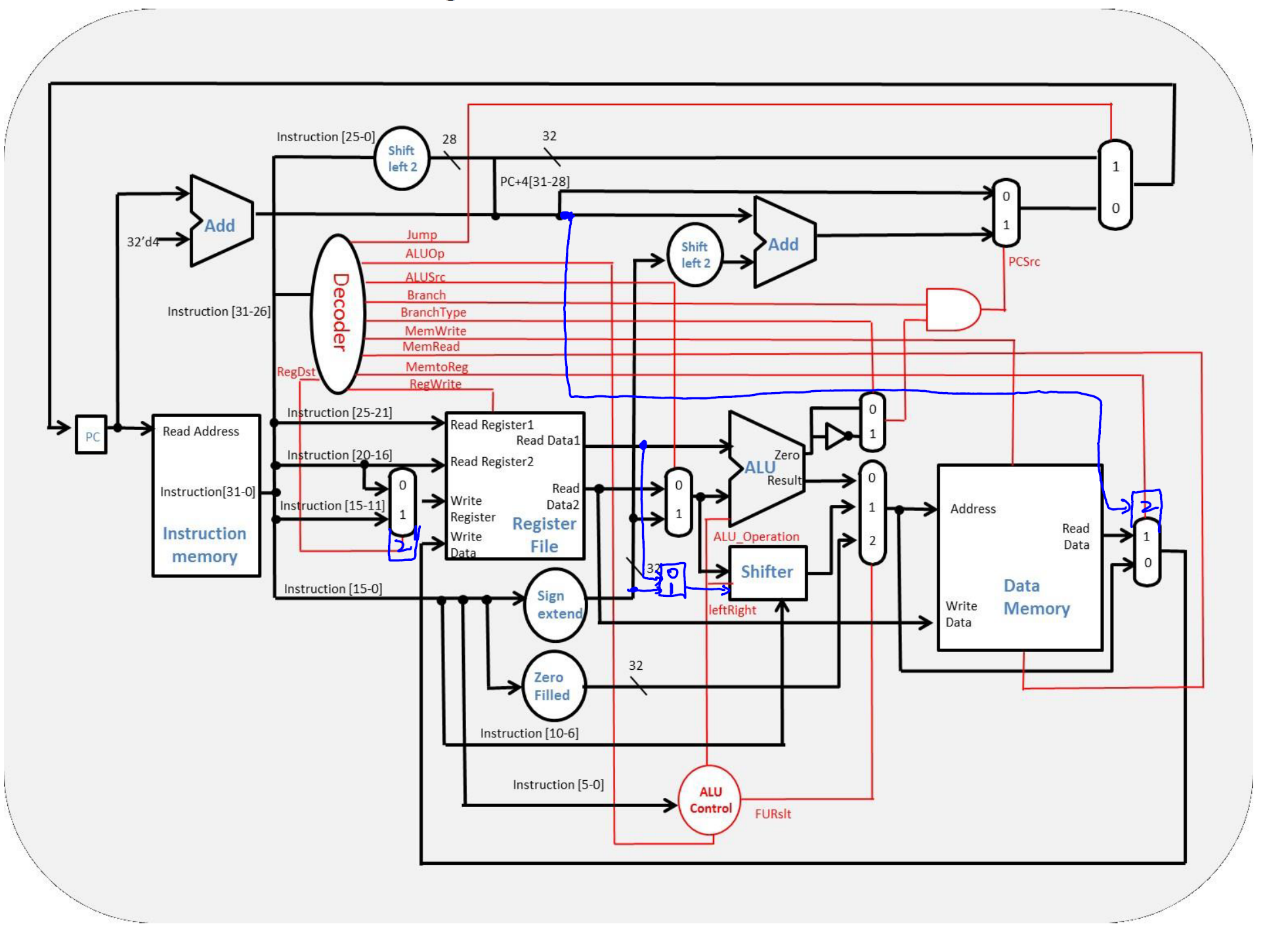
**Computer Organization**

**Architecture diagrams:**

****

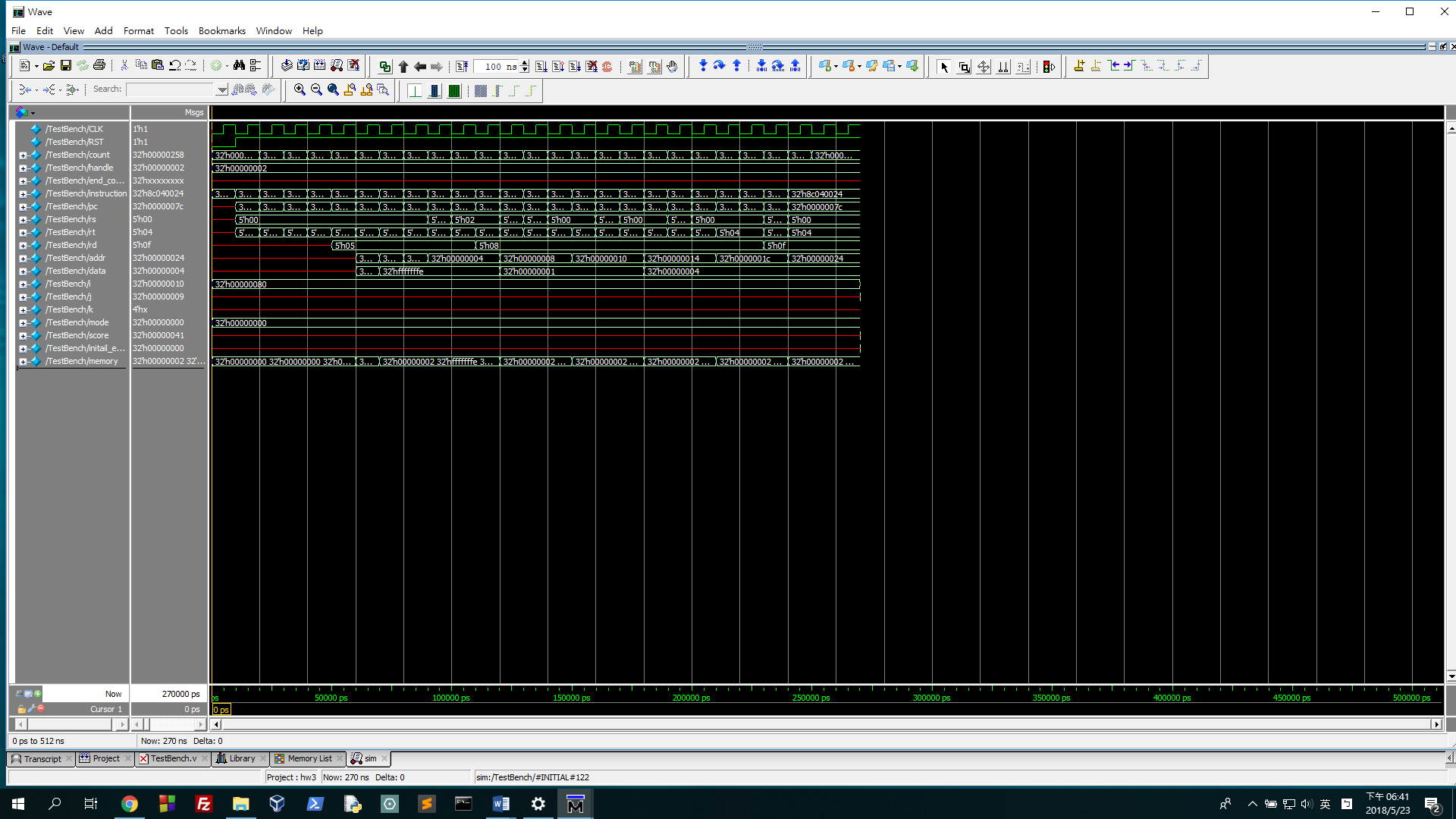
**Hardware module analysis:**

**在Lab4在Decoder, ALU\_ctrl 必須要增加branch,jump,lw,sw等指令，還有要修改一些ALUop的編碼。**

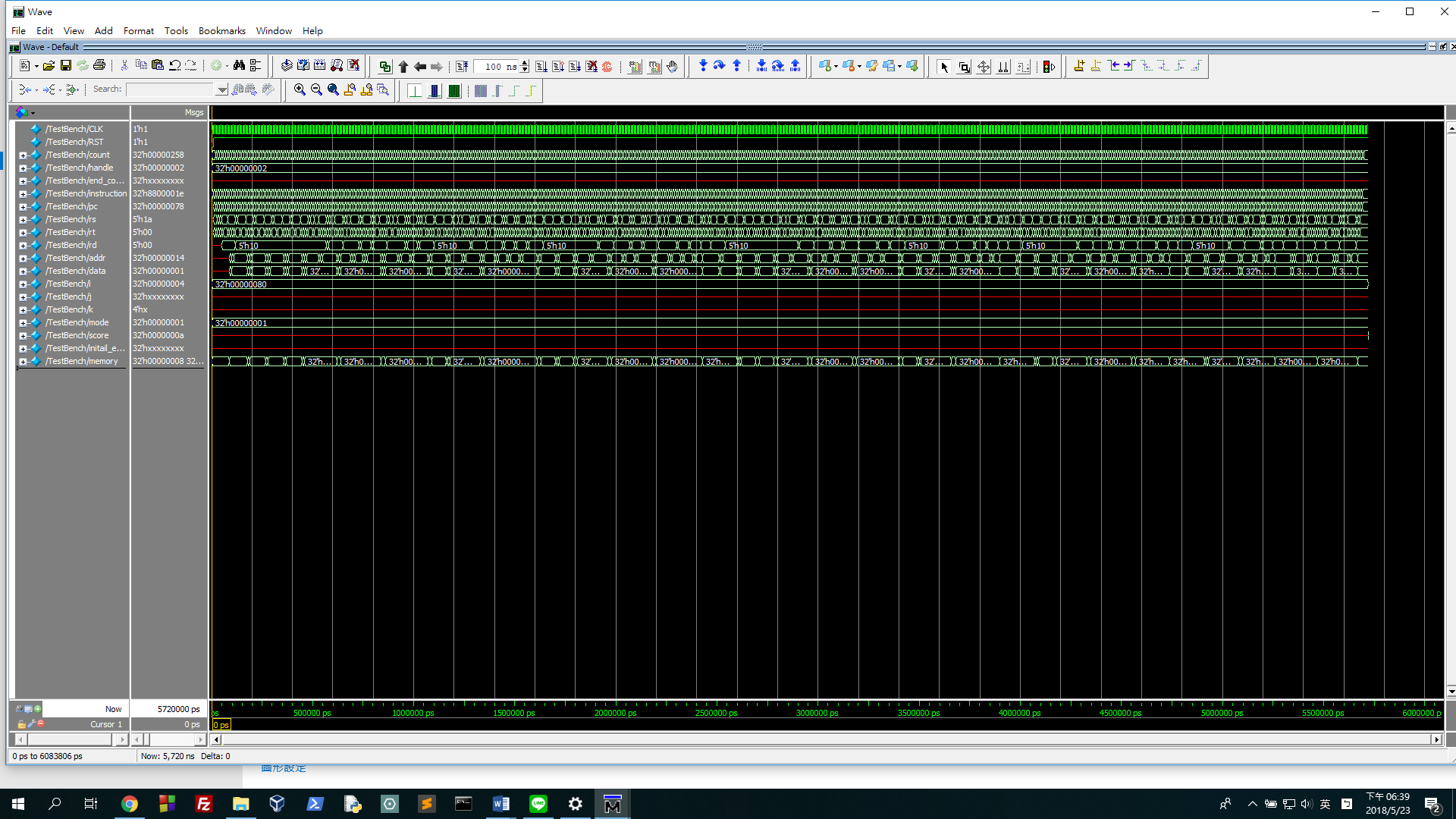
**然後在Simple\_cycle\_cpu 中必須加入兩個3to1的Mux 和一個2to1的Mux來完成這整個cpu，多加這些Mux可以再決定Branch,Jump的時候更加的方便，也可以確定shamt的來源。**

**Finished part:**

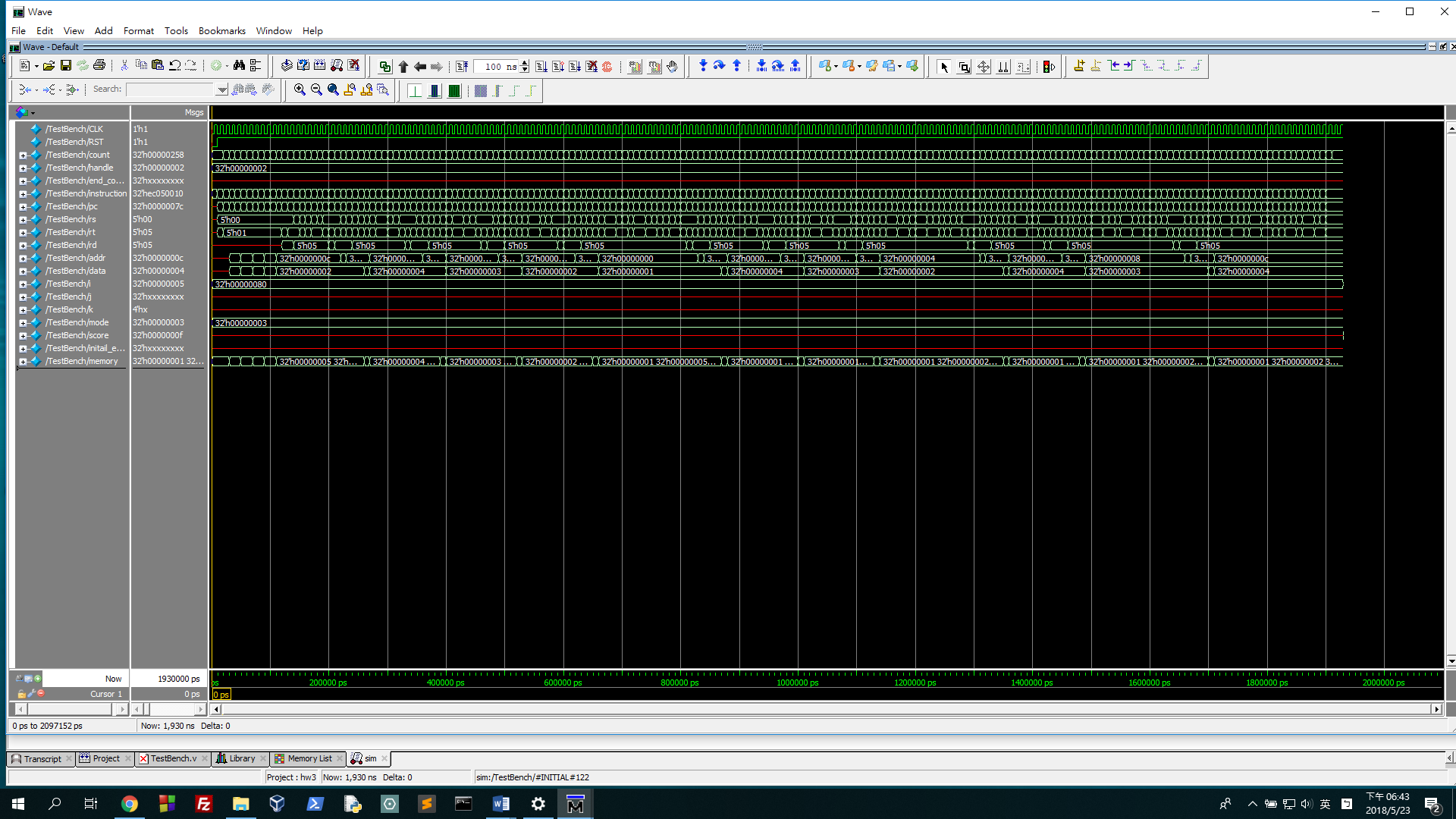
**Testdata1:**



**Testdata2:**



**Bonus:**



**Problems you met and solutions:**

**首先要解決的問題是要知道各種type的Decoder 和 ALU control的路線要怎麼跑，從先理解到整個電路跑的流程，再去用code去把每個gate串起來，我覺得再區分並且熟悉不同type指令的跑法讓我花了一點時間，並且這次用到branch和lw sw，必須對整個電路圖、Mux的路線有更加的了解才有辦法接線的。**

**而大部分電路接好之後比較難的是debug，因為以前比較少用verilog debug所以比較不知道要用甚麼方式看自己的錯誤在哪 code，之後再simulate有問人知道原來也可以用看wave的方式去判斷，在甚麼值的情況下會有問題，讓我可以去改code的邏輯，看wave是一個還不錯的debug方式。**

**Summary:**

**透過這次的lab讓我了解到了更完整的single cycle cpu的運作，以及各個線路的接法，還有不同type的指令(加入新的lw,sw,branch)，應該要跑到哪條對應的電路，先理解整個大架構的電路，再去看細節要如如何去要去如何連接，並且透過這次的lab也多練習到了一些以前比較不會用的verilog語法，希望透過這次的lab可以幫助我在之後寫更複雜cpu時更容易上手。**