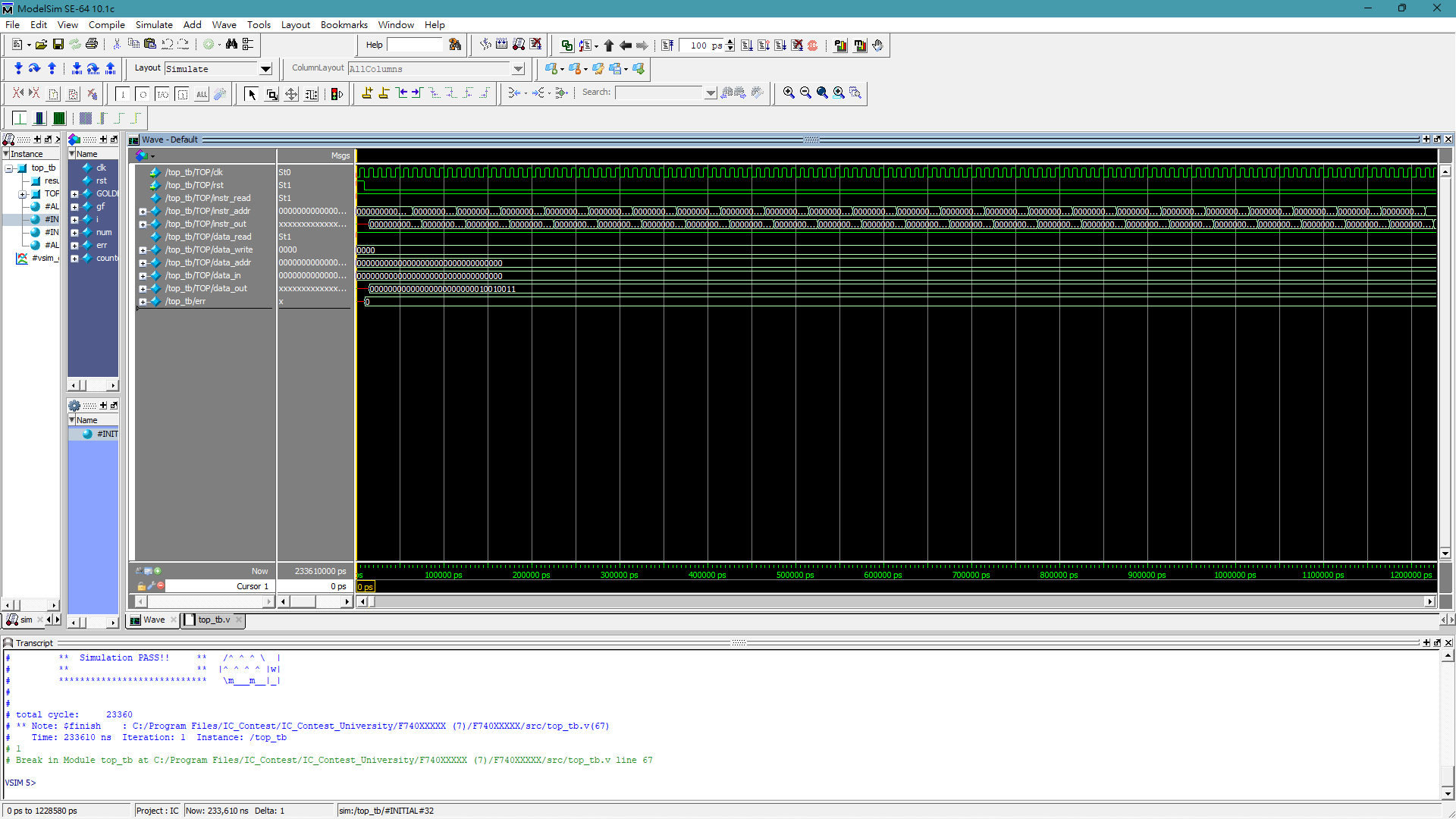
**Computer Organization 2019**

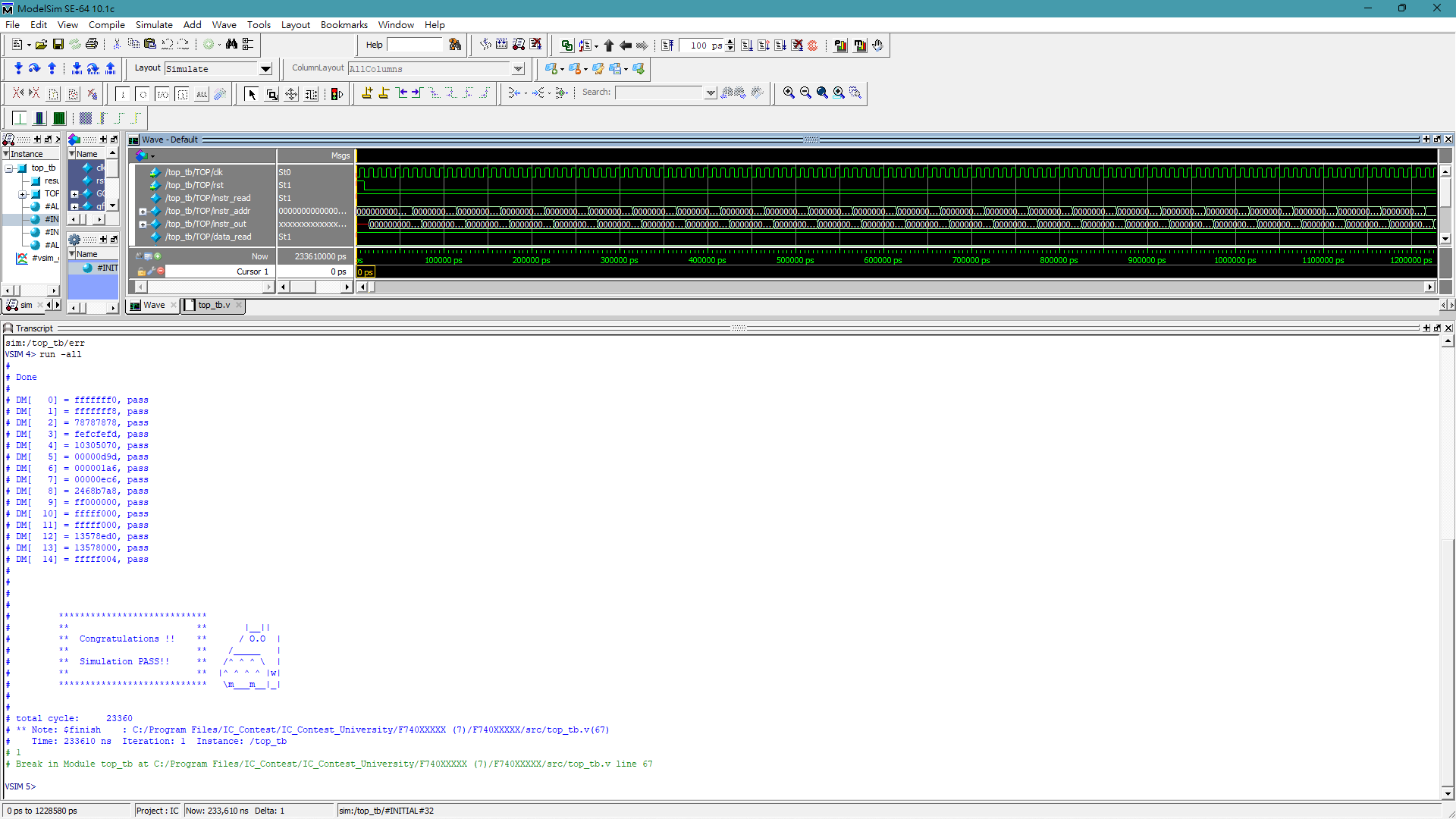
**HOMEWORK 4**

系級: 資訊113 學號: F74096069 姓名: 趙昌昱

**實驗結果圖:**

(波形圖及模擬完成截圖)

****

****

**程式運作流程:**

(簡單說明波形變化的意義)

1. Instr\_read表示instruction 是否被讀進IM
2. Instr\_addr表示 在IM中instruction 的位置，在這裡用來做PC
3. Instr\_out 表示從IM 送進來的instruction
4. Data\_read 表示資料是否讀進DM
5. Data\_write 表示資料是否被寫進DM
6. Data\_addr 表示在DM中資料的位置
7. Data\_in 表示寫入DM的資料
8. Data\_out 表示從DM讀出的資料

波型內容與前次作業相似，但多了幾個instruction，使得instr\_out的內容更為豐富。

**心得**

(請寫下完成本次作業的心得、學到哪些東西、困難點的部分。)

這次的作業多了3種type，分別是Btype, Utype, 跟Jtype。

在經過2次的single cpu作業後，我逐漸了解 RiscV 的Instruction set是如何撰寫的，同時也了解其Register file的作用與不同type之間的功能。

對於Utype, Jtype的參考表如同之前一樣，照著上面的Instruction去刻出功能，然而，這次比較要注意的點是在I type上的JALR指令需要去考慮

Register[0]，也就是$zero是否有保持在0的數值，因為在JALR上有時會動到$zero，造成之後的simulation failed，問題點可能就出在這。

我認為在實作CPU中，不僅可學到verilog的技巧，更能利用這些技能去好好發揮以及改良硬體，對於IC設計來說也是如此。