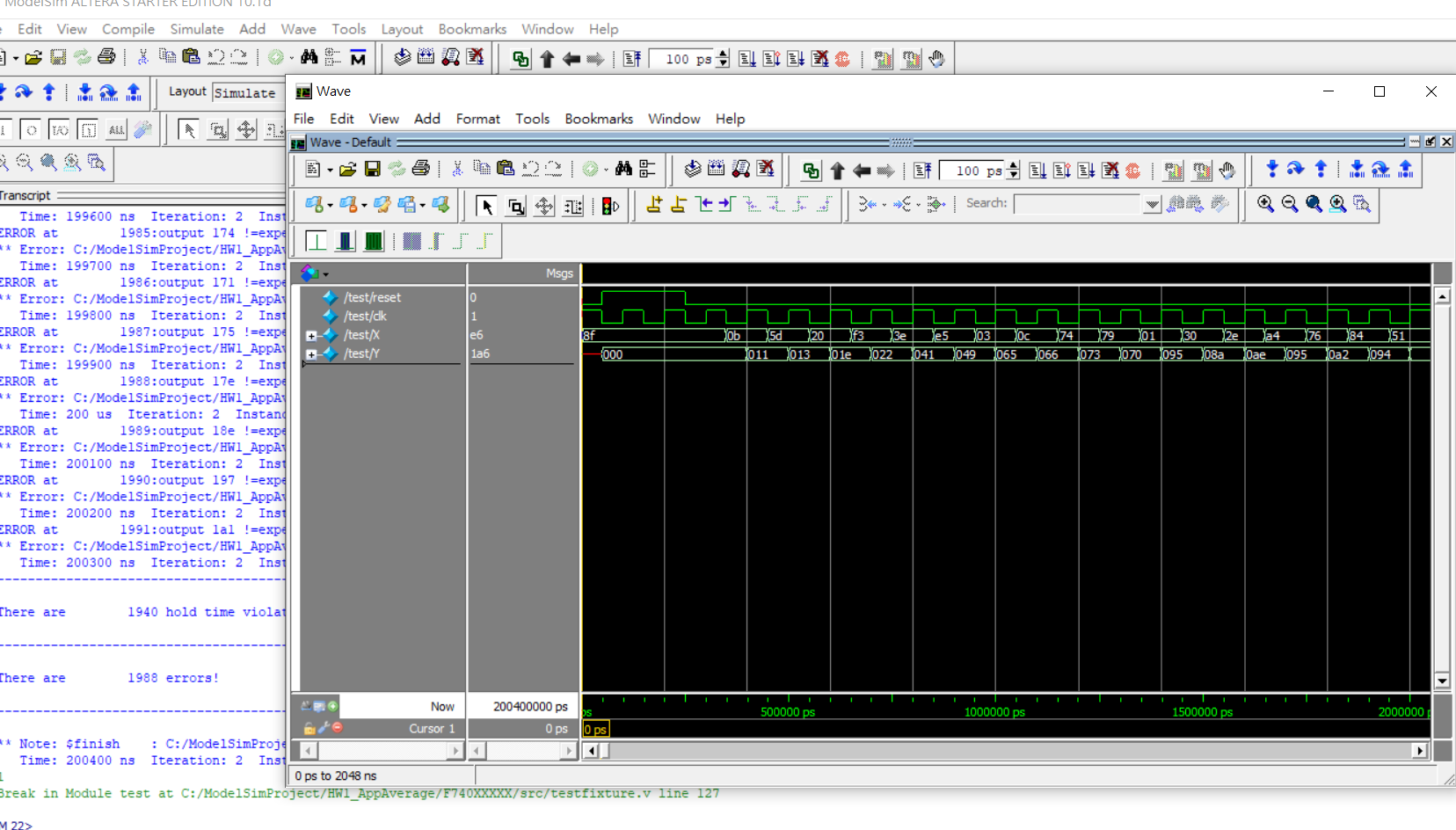
**Computer Organization 2019**

**HOMEWORK 1**

系級: 資訊112 學號: F74086381 姓名: 蘇恩質

**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)

首先，當reset時，會把所有值清空，其他當clk觸發時，會把資料push進data裡面，新的X的值會放在data[7:0]，其他之前的值會往後推8bit。

再來會先把data[7:0]、data[15:8]…等等的值全部加起來，丟到X\_sum裡面，再來把他除以9得到X\_avg。

之後做的事情是去檢查data[7:0]、data[15:8]…等等的值是否有等於X\_avg，如果等於的話那他就會是X\_app，如果沒有的話後面再去作檢查，檢查每一個data[7:0]、data[15:8]…等等的值，如果小於X\_avg，那就去看他有沒有比現在的X\_app大，如果有的話那他就會是新的X\_app，用這個方法去找到小於X\_avg但最接近X\_avg的X\_app。最後把X\_sum加上X\_app \* 9，之後再除以9-1 = 8，得到output Y。

波形圖代表的就是，在X一個一個push進來後，Y的值經過計算會每次都產生變化，而且會比X push進來時晚一點改變，因為Y要經過計算會需要一點時間。

**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。)

1. 會有點難分辨到底是要用reg還是要用wire(後來知道應該是reg要在always裡用，wire應該是在always外要assign時用)，還有因為雖然他是C like的語言，但陣列上的使用卻也不是這麼相近，還蠻不一樣的，有多花蠻多時間在熟悉語法上面。
2. 一開始在做的時候以為是一次輸入九個進來，後來才發現是每過一個clk進來一個數字，然後要重新去計算結果，因此在輸入輸出的方式上有做調整，原本是像陣列輸入一樣一個一個assign，像data[7:0] = X[0], data[15:8] = X[1]這樣，後來改成data[15:8] = data[7:0], data[23:16] = data[15:8]...這樣。
3. 作業給的n是9，在算平均數的時候就沒辦法直接把sum直接右移來做除法，也不是很確定Verilog能不能直接用/這個運算子。
4. 因為心裡提醒自己盡量不要用迴圈，但又不太熟悉Verilog的陣列操作，所以有蠻多地方都是要一個一個刻上去，寫這次作業還可以，因為一次也只有9個input data，但要是換到更大的case可能就會寫得很辛苦。
5. 在Debug時有時候光看wave有點難除錯，只能自己慢慢debug。