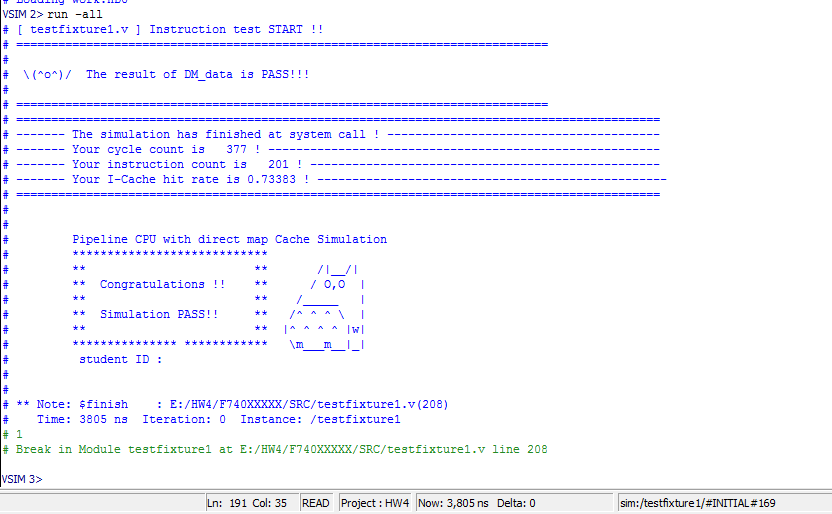
**Computer Organization 2017**

**HOMEWORK IV**

系級: 資訊二乙 學號: E94046157 姓名: 蔡宇軒

**實驗結果圖(snapshot of result)**

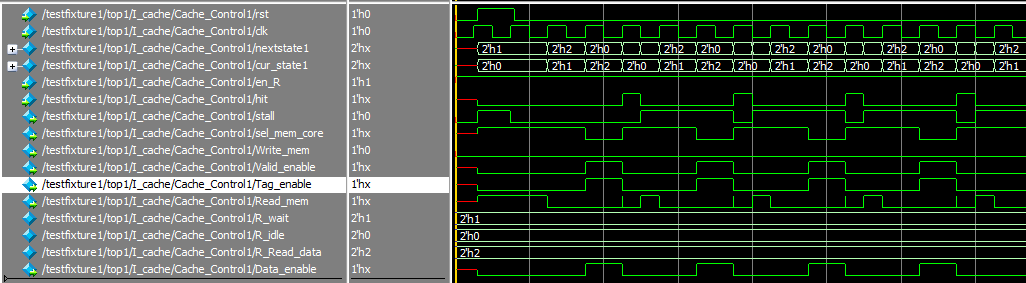
****

**The I-Cache hit rate of your simulation & your reason:**

**指令波型圖( Instruction waveform of some situations)**

(Please explain why your snapshot is correct, including the wires, signals.)

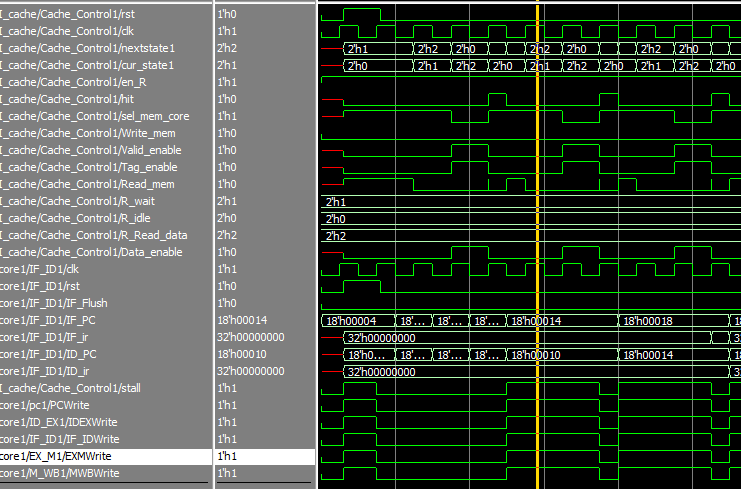
1. I-Cache Miss



Description:

I-Cache Miss時的觸發為en\_Read==1，而hit==0時，此時會觸發stall及readmemory，改變cache裡的state，進而連動IM裡面的state讀取資料。等到讀取完時，valid、tag、data\_enable會設定為1，使資料能夠寫進cache內。完成一次read instruction循環。

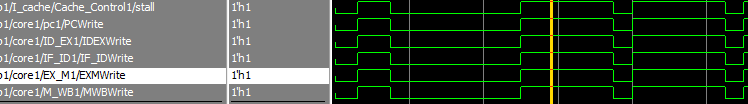
1. I-Cache Hit:



Description:

I\_cache hit時，hit訊號為1，而cache\_controll的state保持為idle，就跟前幾個作業一樣動作，直接讀取資料即可。

1. CPU stall:



Description:

如圖，當cache controll下達stall指令時，表示正在讀取資料，而此時所有五級的Write都將被啟動，使整個CPU keep住目前的值，在verilog語法上就是不做任何事。

**心得(Report)**

這次的作業其實要寫的地方很少，因為助教都把大部分的cache寫完了，我們需要做的只有寫cache controll以及加入write訊號使整個CPU能夠在miss時停止。我在這次作業遇到最大的困難其實不是在本次作業的範圍，而是前幾次作業的範圍。例如ALU上的beq判斷其實有小漏洞，但在上次的作業bug沒有顯露出來，在這次作業中debug時就看了很久才找到這個小錯誤並改正。這學期的作業，若是前面作業不清楚，其實後面就都很難跟上了。