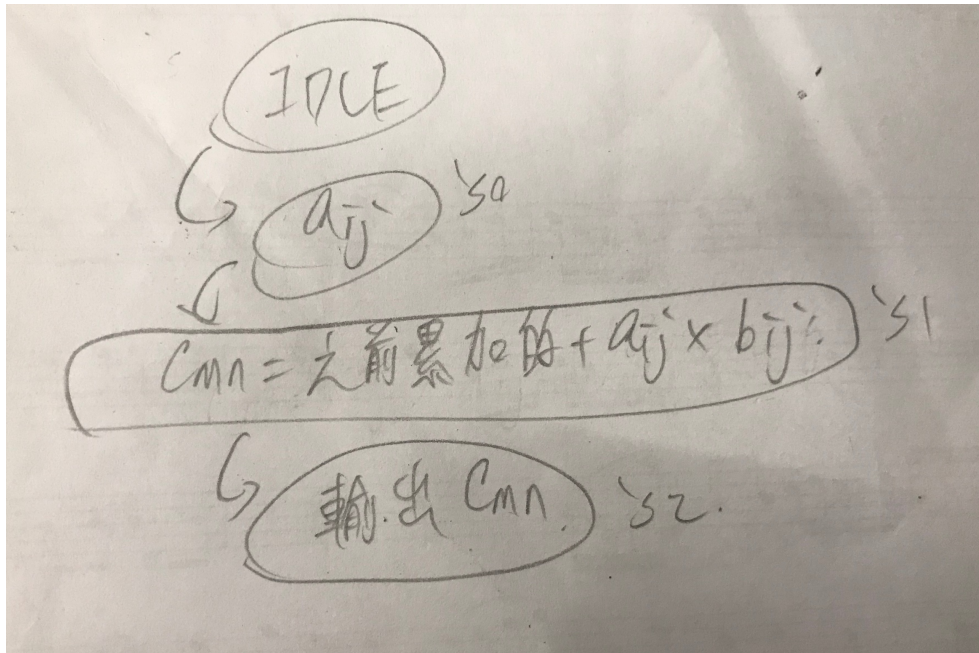


Lab4 Report

106072237 黃采瑩 經濟 21

State Transition Graph



解釋

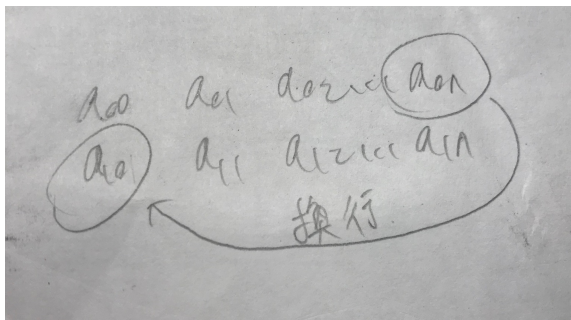
前情提要：相乘的兩個矩陣分別為 a 與 b ，數值座標由 i 與 j 控制。乘出來的矩陣為 c ，數值座標由 m 與 n 控制。我的寫法是在矩陣各取一個值，相乘並累加到完成就輸出。

``IDLE`: 當 `reset` 不是 `1'd0` 的時候開始 ``s0`。

``s0`: 取得矩陣 a 的一個值，並將 `next_i` 和 `next_j` 設定為取得矩陣 b 對應數值所需要的值。

``s1`: 取得矩陣 b 的一個值，並將之與在 ``s0` 取到的值相乘，累加到 `next_write_data`。我還在這裡寫了一個 `if else` 迴圈，表達若完成一個 c 矩陣的值 (`i == b - 20'd1`)，便跳到 ``s2` 並將其輸出。

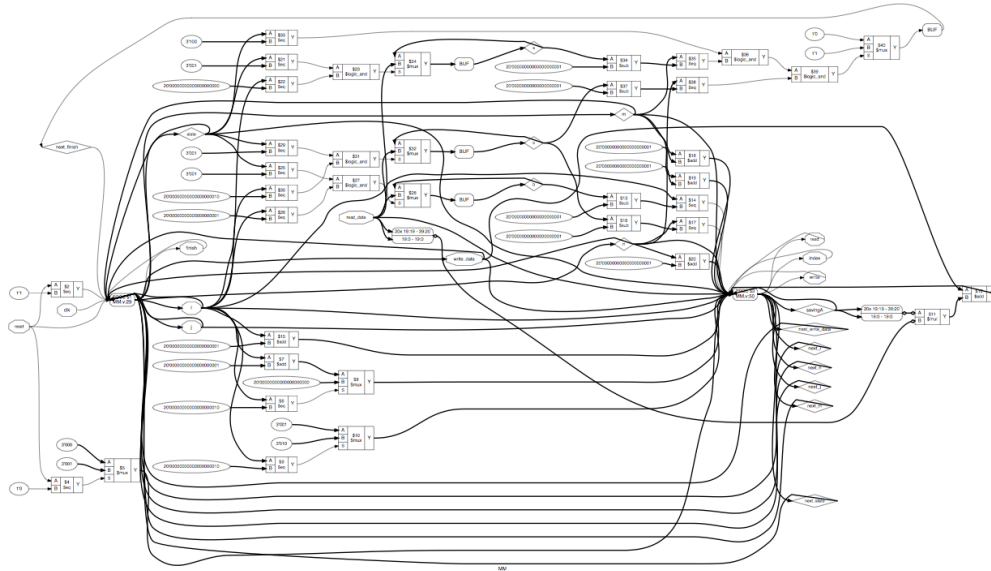
``s2`: 輸出 ``s1` 累加完的值，並寫 `if else` 迴圈表示兩個狀況，分別是要換行與不換行



換行的話 $m + 20'd1$ 且 n 變為 $20'd0$ ，不換行的話 m 維持不變， $n + 20'd1$ 。

最後我 assign next_finish 在 C 輸出最後一個數後為 1'b1，這樣 finish 就會在答案維持一個 cycle 後才為 1'b1，輸出的答案就會是正確的。

Block Diagram



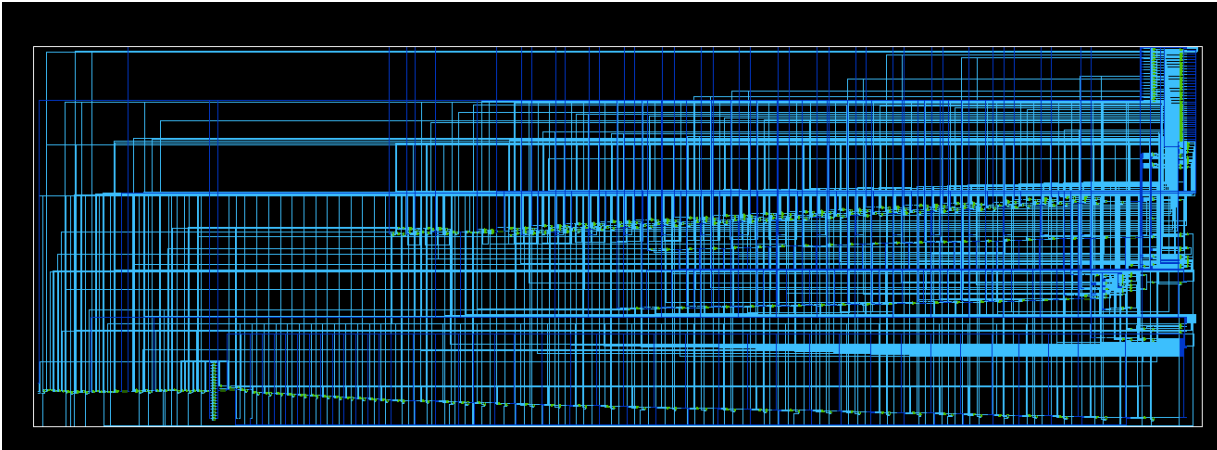
Ncverilog

```
#####
Row:      0 Column:      0 is correct.
#####
Row:      0 Column:      1 is correct.
#####
Row:      0 Column:      2 is correct.
#####
Row:      0 Column:      3 is correct.
#####
Row:      1 Column:      0 is correct.
#####
Row:      1 Column:      1 is correct.
#####
Row:      1 Column:      2 is correct.
#####
Row:      1 Column:      3 is correct.
#####
#Congratulation!!!#
#####
Simulation complete via $finish(1) at time 4920 NS + 3
./MM_tb.v:118      $finish;
ncsim> exit
[dld0106@ic52 ~/lab4]$
```

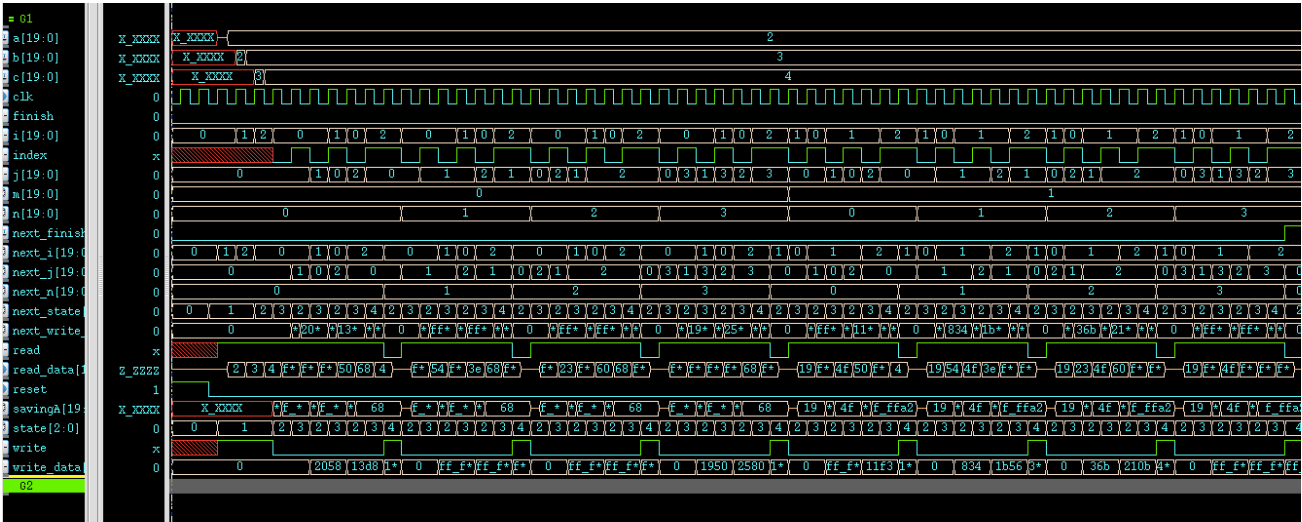
Sim



Netlist



nWave



遇到的問題與討論

- 1.最後一個 row 和 column 怎麼樣都對不了，問了助教和看了 nWave 才知道原來是因為沒有讓答案維持一個 cycle，我後來把 finish 加上一個 next_finish 後就沒事了。
- 2.我對於 reg 和 wire 的宣告很有障礙，常常會因為沒有宣告好就 error，後來自己得到了一個很早以前就應該得到的結論，放在 always block 裡面的要用 reg，放在外面的要用 wire，想要什麼電路就自己去宣告就行了，不過宣告的時候不能忘記加 bits 數量，不然也會壞掉。
- 3.Block Diagram 很複雜，我沒有想法要怎麼畫，後來有好心人教我用奇特的軟體畫，這真的省時又省力多了，謝天謝地我還有朋友。
- 4.我本來沒有 finite state machine 的概念，我沒有搞懂一個東西和 next 那個東西個別存在的必要性，以及他們本身具備的功能，直到寫這次的 lab 我才徹底了解到他們存在的必要性，其實他是一個很好用的東西，只要把那個東西的 next 設定好，下一個週期那個東西就會變成設定好的值，在寫這個 lab 的時候我因為沒有搞懂有限狀態機所以一開始障礙很多，不過在搞懂之後很多問題都迎刃而解。
- 5.本來以為要把 next_finish 宣告成 reg 寫在 always block 裡面，本來還為此困擾很久，因為我另外寫了一個 if 來放 next_finish = 1'd1，可是那個迴圈一直壞掉。後來我得知 next_finish 可以是 wire 然後直接用 assign 的，事情就因此好辦多了。