

Lab4 Matrix multiplication

105031212 吳紹齊

一、目標

實作兩個矩陣的相乘(非點乘)。

$$[\mathbf{AB}]_{i,j} = A_{i,1}B_{1,j} + A_{i,2}B_{2,j} + \cdots + A_{i,n}B_{n,j} = \sum_{r=1}^n A_{i,r}B_{r,j}$$

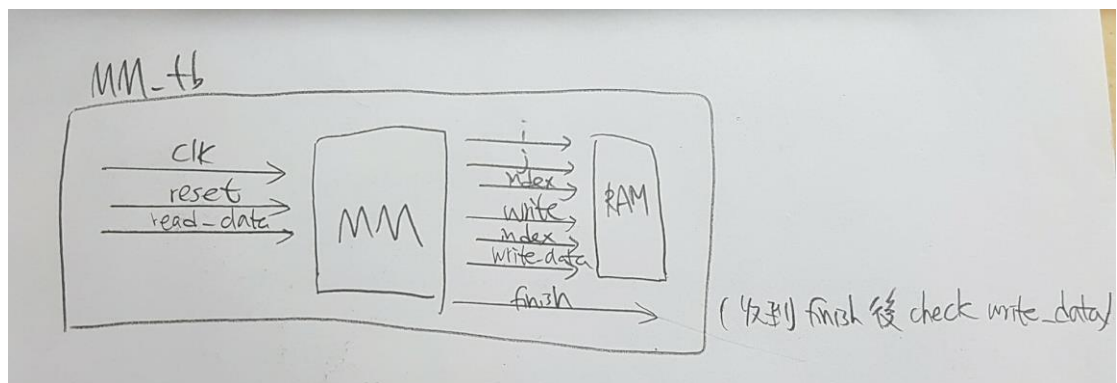
e.g.

$$\begin{pmatrix} 3 & 5 \\ 7 & 1 \end{pmatrix} \begin{pmatrix} 2 & 4 \\ 6 & 3 \end{pmatrix} = \begin{pmatrix} 36 & 27 \\ 20 & 31 \end{pmatrix}$$

(圖源：Lab4 教學平台)

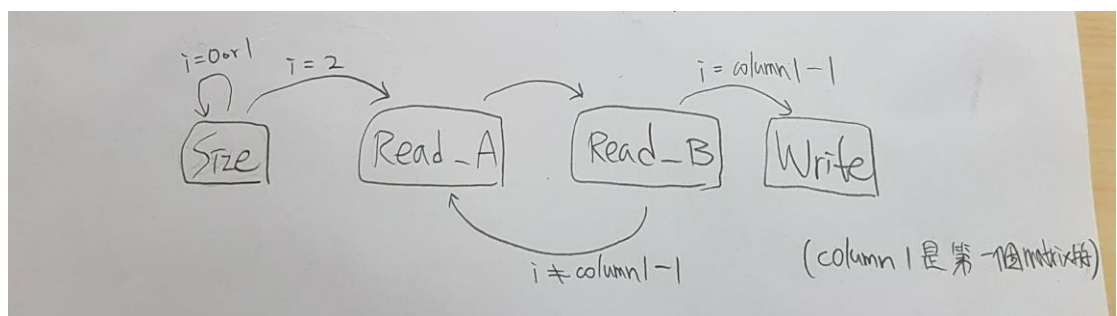
二、設計

整體概念：



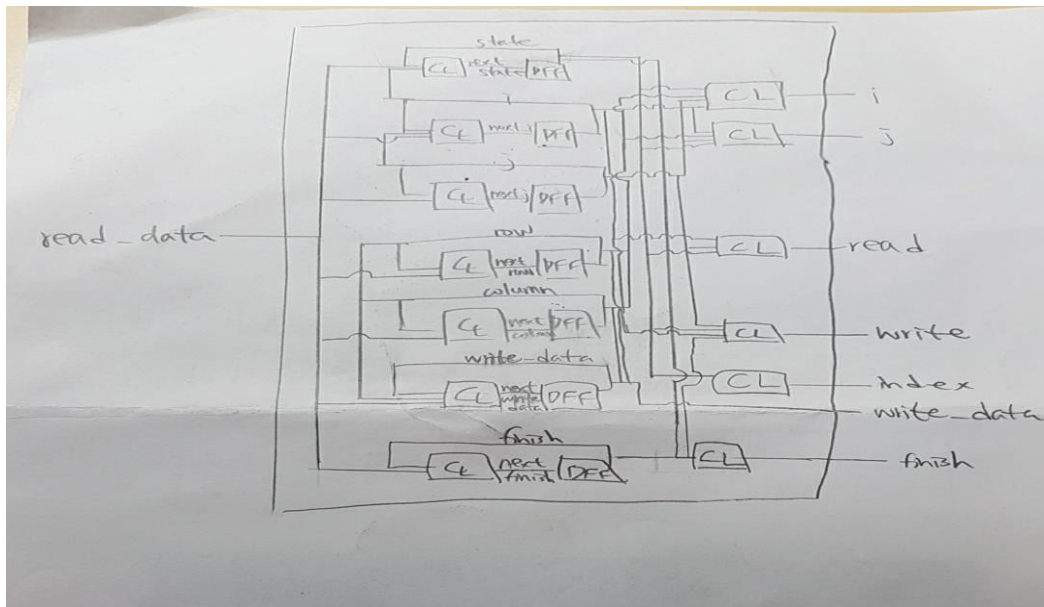
主要是 MM 與 RAM 的互動，MM 給參數讓 RAM 回傳資料，MM 內部運算完畢後，再把值寫回 RAM。(詳細見 CODE 與 Tb)

State Transition Graph



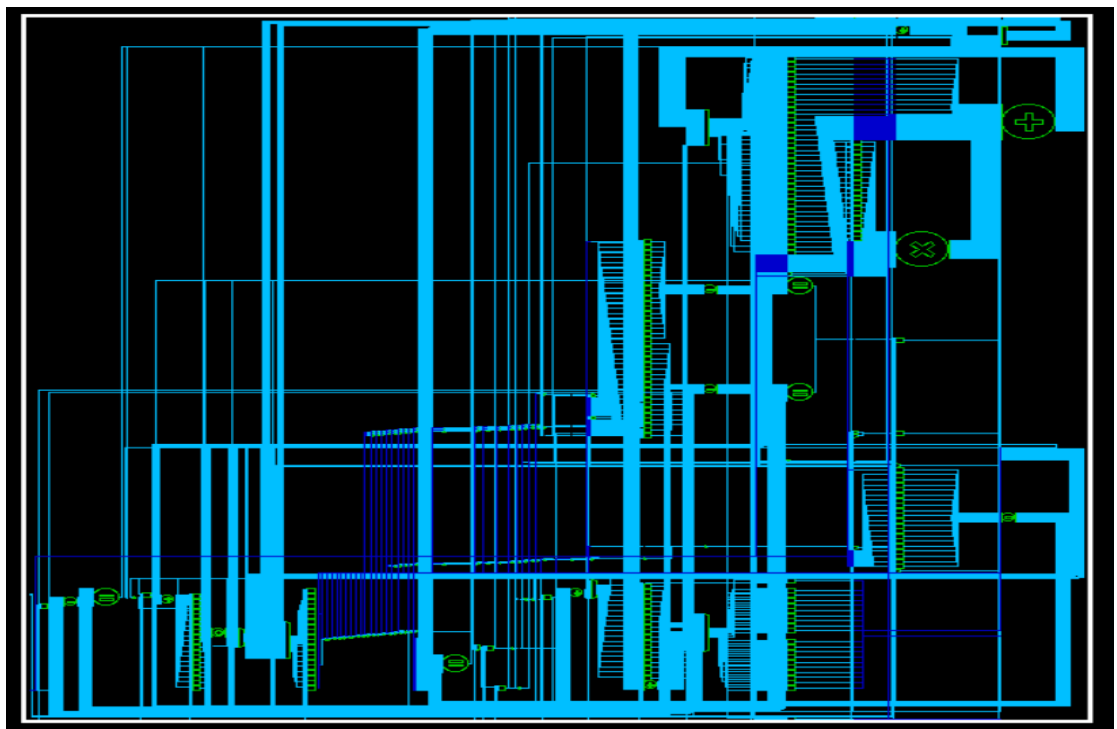
原本是設計先讀取一個矩陣，接著在一個矩陣，然後再取值作運算，如此方式感覺非常的緩慢，經過助教提醒後，決定改成現在這個樣子，讀取值後馬上運算，算好後馬上丟進 write_data 讓它累加，最後在進入 WRITE State 後舉起 finish，讓 tb 檢查。(詳細見 CODE)

Block Diagram



每個 DFF 都要接 CLK 與 RESET，圖中省略，以維持整潔。

由 DV 我們還可以看到更接近實際的電路圖長什麼樣子



圖中最顯眼的就是加法器和乘法器了，第一次放大仔細看，感覺又更貼近現實。

NCVERILOG 結果

MAKE SIM

```
Loading snapshot worklib.test:v ..... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_J-2014.12-SP3, Linux, 07/05/2015
(C) 1996 - 2015 by Synopsys, Inc.
*Verdi3* : Create FSDB file 'MM.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.
      2      3      3      4
      2      3      3      4
#####
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 4840 NS + 3
```

MAKE SYN

```
Loading snapshot worklib.test:v ..... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
FSDB Dumper for IUS, Release Verdi3_J-2014.12-SP3, Linux, 07/05/2015
(C) 1996 - 2015 by Synopsys, Inc.
*Verdi3* : Create FSDB file 'MM_syn.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.

Warning! Timing violation
$setuphold<setup>( negedge G &&& (SandR == 1):1731 PS, negedge D:1629 PS, 0.190 : 190 PS, 0.026 : 26 PS );
File: /theda21_2/CBDK_IC_Constest/cur/Verilog/tsmc13.v, line = 26241
Scope: test.mmmmm.index_reg
Time: 1731 PS

      2      3      3      4
      2      3      3      4
#####
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 4840369 PS + 0
./MM_tb.v:118      $finish;
ncsim> exit
[dld0026@ic26 ~/Lab_4]$
```

問題與心得：

這次遇到比較大的問題在於不理解 RAM 與 MM 之間的互動關係，一開始看到助教的說明，對於在給 I、J、INDEX 後，RAM 會有什麼反應不太理解，後來看了 tb 以後就比較懂了。

這次一開始會出現 time violation 的情形，似乎是因為電路中有不少 latch 產生，後來想了很久，發現是“自己接自己”的問題，以後在這環節應多加注意。