NYCU-ECE DCS-2025

HW04

Design: MAC Array for Matrix Multiplication and Convolution

Learning Objectives

1. 了解convolution與zero padding的運算方法
2. 透過resource sharing提升硬體的使用率並減少面積
3. 利用較大的輸入與輸出data rate加快運算
4. 自行決定輸入與輸出的順序以達到更流暢的運算並節省DFF的使用

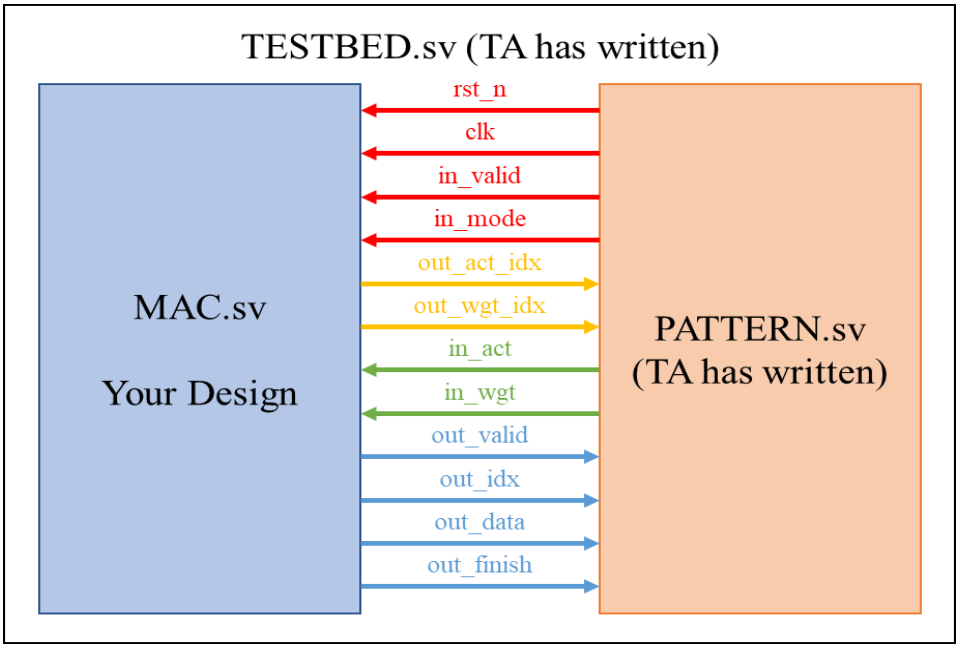
Data Preparation

1. 從 TA 目錄資料夾解壓縮:

   % tar xvf ~dcsTA01/HW04.tar

2. 解壓縮資料夾 HW01 包含以下:
   a. 00_TESTBED/
   b. 01_RTL/
   c. 02_SYN/
   d. 03_GATE/

Block Diagram



Background Knowledge

在深度學習與數學中，矩陣乘法與卷積(convolution)運算是兩個核心概念，特別是在影像處理與神經網路中應用廣泛。

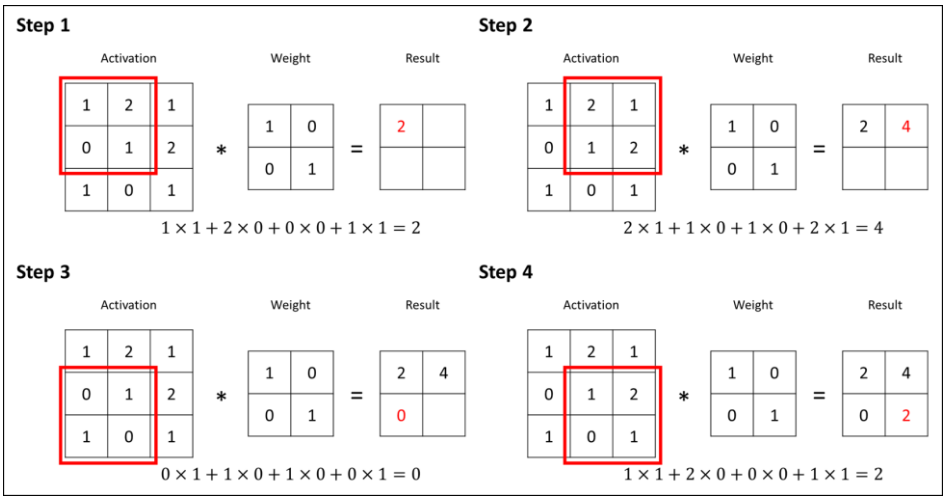
1. 矩陣乘法 (Matrix Multiplication)

矩陣乘法是一種線性代數運算，用來計算兩個矩陣的乘積。若矩陣 A 為 m×n 大小，而矩陣 B為n×p 大小，則其乘積矩陣 C 為 m×p 大小，計算方式為：

$$C_{ij} = \sum_{k=1}^{n} A_{ik} \times B_{kj}$$

2. 卷積運算 (Convolution Operation)

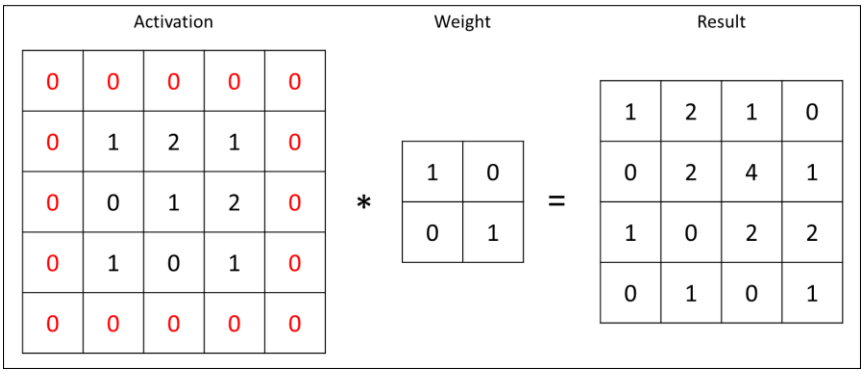
卷積 (Convolution) 是一種數學運算，通常用於影像處理與神經網路中的卷積層 (Convolutional Layer)。它的概念來自於信號處理，主要是將一個「小範圍的權重矩陣」應用到「較大的輸入矩陣」上，並透過滑動窗口進行運算。



在深度學習中，我們一般稱「較大的輸入矩陣」為activation，「小範圍的權重矩陣」為kernel或weight。運算時，我們首先將weight矩陣置於activation矩陣的左上角，並且將所有對應位置的數值相乘，最後將得到的所有乘積加總，得到輸出矩陣左上角的值。如上圖中step 1應將weight矩陣置於activation矩陣的紅框處。之後將weight矩陣逐漸向右移動，並且重複運算，直到activation矩陣的邊界，如此可以得到輸出矩陣的第一列(如上圖中step 2)。之後便重新將weight矩陣移至左方並向下移動一行，重複以上操作，能夠得到下一列的輸出。如此不斷循環，直到activation矩陣的最下列，便完成運算。若將其表示成數學式，則其輸出應為：

$$Y(i,j) = \sum_{m}\sum_{n} A(i+m, j+n) \times K(m,n)$$

其中Y為輸出矩陣，A為activation，K為weight，且K的大小為m×n。但是從上述例子中可以發現，經過卷積運算後，輸出矩陣的大小變得比原本的輸入矩陣小。若想要避免此情況發生，zero padding是一種常見的處理方式。其作法十分簡單，只需要事先在輸入矩陣周圍圍上一或數圈的0，使其增大即可。如下圖中，將3×3的輸入矩陣擴大為5×5之後與2×2的weight進行卷積運算，其輸出矩陣甚至變得比原輸入矩陣還大。

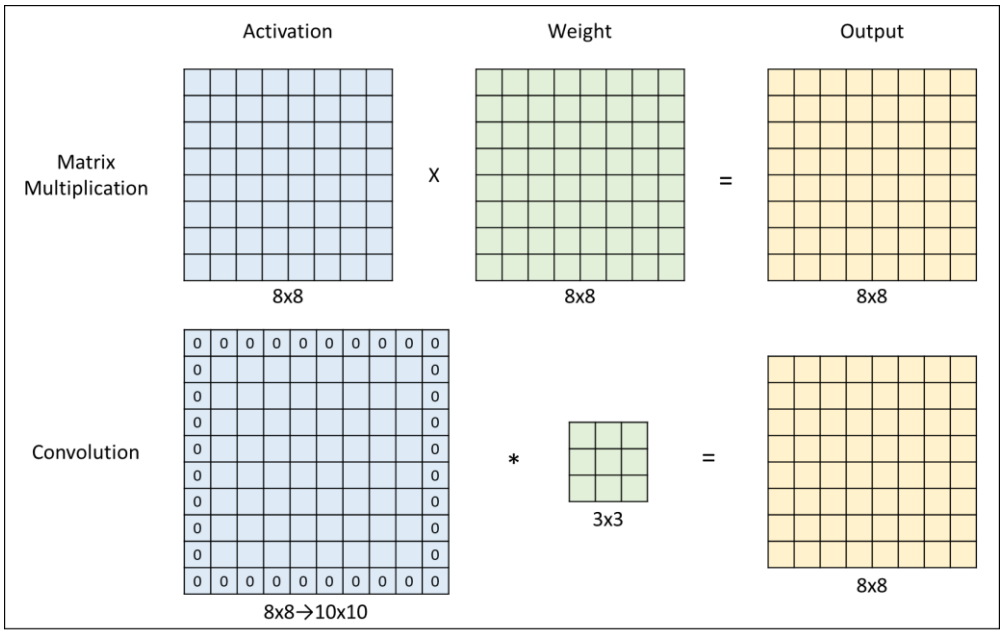


上述二種運算，其本質都是大量的乘法與加法，若是能夠妥善設計硬體，即可令其能夠同時支援上述兩種不同的運算。

Design Desciption

本次作業目標為設計具備矩陣乘法與卷積運算兩種功能的AI加速器。

進行矩陣乘法時，你需要將8x8的activation矩陣與8x8的weight矩陣相乘，得到8x8的輸出矩陣。進行卷積運算時，你需要將將8x8的activation使用zero padding使之變為10x10的大小，與3x3的weight矩陣做卷積運算，得到8x8的輸出矩陣。可參考下圖。

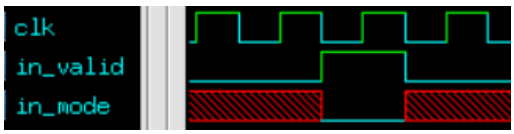


本次設計的所有輸入與輸出可以分為三組：

1. 開始訊號

當接收到in_valid訊號為high時，表示新的一筆測資開始。與此同時，你需要根據in_mode的值判斷需要進行矩陣乘法或是卷積運算。如以下波形圖的範例，由於in_mode為0，因此需要進行矩陣運算。

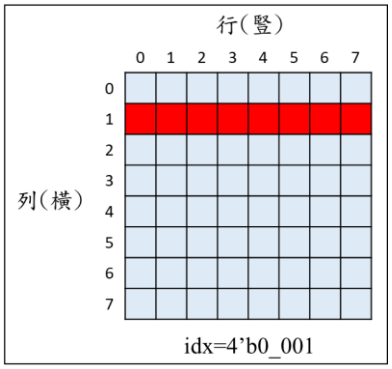
Signal Name	Type	Bit Width / Dimension	Description
in_valid	input	1	新的一筆測資開始時，會在一個clock cycle內持續為high。
in_mode	input	1	當in_valid為high時，提供關於此測資的運算模式。0表示矩陣乘法，1表示卷積運算。



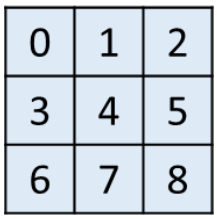
2. Activation與Weight矩陣輸入

本次的資料輸入採用較為特殊的方式。自in_valid為high開始的下一個clock cycle，可以從in_act與in_wgt獲取activation與weight矩陣的資料，但是目前提供矩陣上的哪部分資料，則需要由你的設計透過out_act_idx與out_wgt_idx兩個輸出來指定。

對於activation矩陣，以及進行矩陣運算時的weight矩陣，其大小皆為8x8(做卷積運算時雖需要將activation矩陣透過zero padding擴大到10x10，但是輸入只會提供原先的8x8矩陣)，in_act與in_wgt能夠在一個clock cycle提供矩陣中一整行或一整列的資料，你需要使用out_act_idx與out_wgt_idx指定你要求一行(豎)或者一列(橫)的資料，以及要求哪一行或列。如下圖的範例，idx[3](MSB)表示要求一行或一列，而其餘三個位元則表示是第幾行或第幾列。因此下圖中要求的是第1列，即圖中紅色區塊所示。



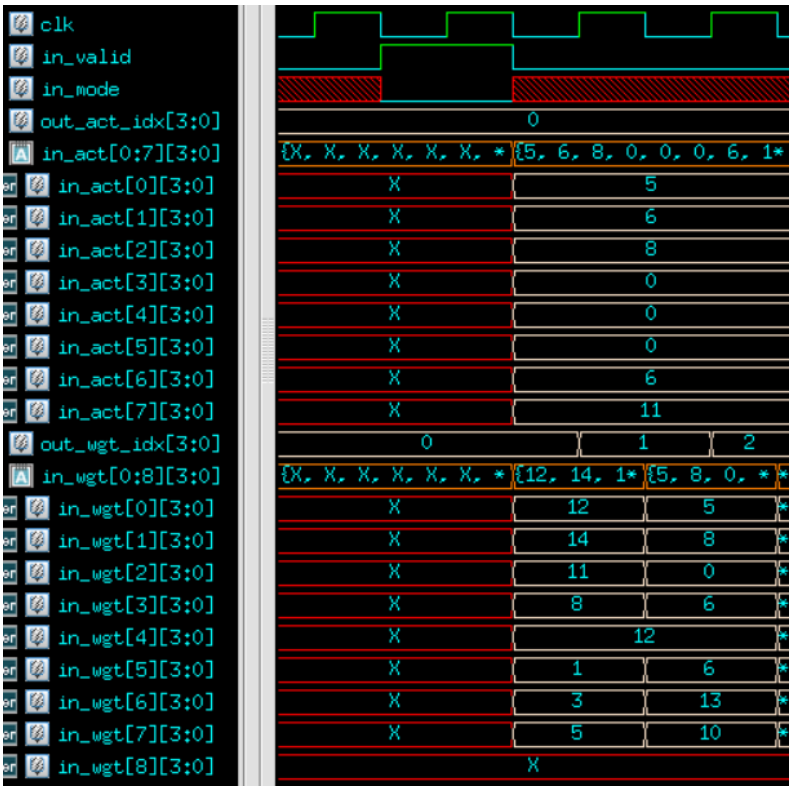
進行卷積運算時，由於weight矩陣大小僅有3x3，因此無需特別指定要求的資料位置，會固定給予整個矩陣的資料，其順序從左上角開始，先往右後往下，如下圖所示。



須注意所有輸入資料皆為4 bit的正整數，不考慮負數。另外in_wgt為了能提供整個3x3的矩陣，因此會同時輸入9個數，但是當運算為矩陣乘法時其第九個數(in_wgt[8])則會為unknown。out_act_idx與out_wgt_idx兩個輸出在本次作業中沒有任何限制(僅有此二輸出訊號允許不進行reset)，同學可以以其控制資料的提供順序，或重複讀取資料也是可以的。

Signal Name	Type	Bit Width / Dimension	Description
out_act_idx	output	[3:0]	要求activation矩陣資料時指定位置。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示要求的行或列的編號。
in_act	input	[0:7][3:0]	根據上一個clock cycle的out_act_idx給予activation矩陣的一行或一列。
out_wgt_idx	output	[3:0]	要求weight矩陣資料時指定位置。若進行卷積運算則無意義。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示要求的行或列的編號。
in_wgt	input	[0:8][3:0]	進行矩陣乘法時，根據上一個clock cycle的out_wgt_idx給予weight矩陣的一行或一列，而in_wgt[8]為unknown。進行卷積運算時，則固定輸出整個weight矩陣，順序為從左上角開始，先往右後往下。

下圖為進行矩陣乘法時的資料讀取範例。其中可以看到資料從in_valid為high的下一個clock cycle (negedge)才開始提供，並且根據out_act_idx的數值，提供activation矩陣的第0列為{5,6,8,0,0,0,6,11}；根據out_wgt_idx的數值，提供weight矩陣的第0列為{12,14,11,8,12,1,3,5}，第1列為{5,8,0,6,12,6,13,10}。須注意pattern在clock negedge時才會讀取out_act_idx與out_wgt_idx的值，並且更新in_act與in_wgt的內容。同學在設計時應提前一個clock cycle輸出out_act_idx與out_wgt_idx，才能在下一個clock cycle獲得矩陣的資料。



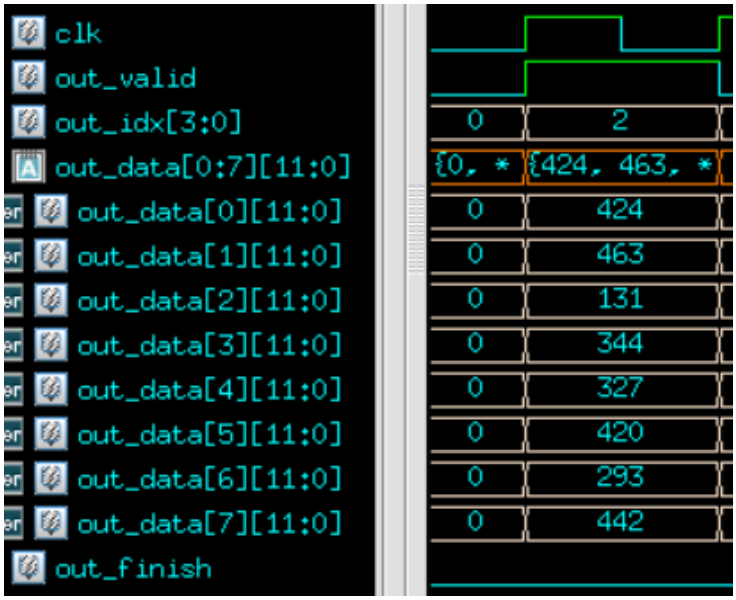
3. 輸出結果

本次的輸出方式與上述讀取activation矩陣的方式十分相似。你一次需要輸出一整行或一整列共8個數值，並且你可以指定你輸出的是行或列，以及是第幾行或第幾列。不過，與讀取矩陣資料時你需要在前一個clock給定行列編號不同，在輸出時你只需要將out_valid設為1，並且在同一個clock cycle給予out_idx(行列編號)以及out_data(輸出矩陣內容)即可。當你進行最後一次輸出時，你還需要將out_finish設為1表示輸出結束。

在輸出結束前，你可以不限次數輸出，pattern會記住你輸出的位置的值。當你對同一個位置重複輸出時，pattern會採用較晚拿到的值作為最終結果。out_idx, out_data, out_finish 三個輸出都只有在out_valid為1時才會被讀取，其餘時間其可以是任何值。另外，在輸出結束前，pattern都不會提供下一筆測資的資料(in_valid為0)。

Signal Name	Type	Bit Width / Dimension	Description
out_valid	output	1	為1時表示存在需要輸出的資料。
out_idx	output	[3:0]	表示輸出值在輸出矩陣中的位置。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示輸出的行或列的編號。
out_data	output	[0:7][11:0]	輸出值。為輸出矩陣中的一列或一行。
out_finish	output	1	為1時表示已經完成整個輸出矩陣的內容。

下圖為一次輸出的範例。其out_idx表示輸出的是第2列，且其內容為{424,463,131,344,327,420,293,442}。由於out_finish為0，表示這並不是最後一次輸出，之後還會對輸出矩陣中的其他位置進行輸出。跟輸入時不同，out_idx與out_data應同時在out_valid為high時提供，兩者不存在時間差。



Inputs

Signal Name	Bit Width / Dimension	Description
clk	1	Clock signal.
rst_n	1	Asynchronous reset.
in_valid	1	新的一筆測資開始時，會在一個clock cycle內持續為high。
in_mode	1	當in_valid為high時，提供關於此測資的運算模式。0表示矩陣乘法，1表示卷積運算。
in_act	[0:7][3:0]	根據上一個clock cycle的out_act_idx給予activation矩陣的一行或一列。
in_wgt	[0:7][3:0]	進行矩陣乘法時，根據上一個clock cycle的out_act_idx給予weight矩陣的一行或一列，而in_wgt[8]無意義。進行卷積乘法時，則固定輸出整個weight矩陣，順序為從左上角開始，先往右後往下。

Outputs

Signal Name	Bit Width / Dimension	Description
out_act_idx	[3:0]	要求activation矩陣資料時指定位置。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示要求的行或列的編號。
out_wgt_idx	[3:0]	要求weight矩陣資料時指定位置。若進行卷積運算則無意義。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示要求的行或列的編號。
out_valid	1	為1時表示存在需要輸出的資料。
out_idx	[3:0]	表示輸出值在輸出矩陣中的位置。 [3]為0時表示一列，為1時表示一行。 [2:0]則表示輸出的行或列的編號。
out_data	[0:7][11:0]	輸出值。為輸出矩陣中的一列或一行。
out_finish	1	為1時表示已經完成整個輸出矩陣的內容。

Specifications

1. Top module name: **MAC**(File name : **MAC.sv**)
2. 請用 SystemVerilog 完成你的作業。
3. 01_RTL & 03_GATE 模擬必須通過(不可以有timing violation)。
4. 02_SYN result 不能有 error 且不能有任何 latch，同時timing必須是MET。
5. 考量到本次需要對較大的矩陣做運算，可能會需要多開幾組平行的硬體，因此允許使用for迴圈加速設計。但還是請同學注意自己的coding style。
6. Clock cycle time 可以自行調整，最多取到小數點後一位，並且在繳交作業時必須寫明你的cycle time。自己測試時，你需要修改00_TESTBED/PATTERN.sv的第1行與02_SYN/syn.tcl的第23行所標明的cycle time。

```
`define CYCLE_TIME 10.0
```

00_TESTBED/PATTERN.sv第1行

```
set CYCLE 10.0
```

02_SYN/syn.tcl第23行

7. 當rst_n為0時，除去out_act_idx與out_wgt_idx以外的所有輸出訊號必須為0。
8. 經過reset後，out_valid不可為unknown。
9. 自in_valid為1並開始新的測資後，必須要在1000個clock cycle以內完成運算並

且輸出，令out_valid與out_finish在同一個clock cycle內同時為1。

10. 當out_valid為1時，out_idx與out_finish不可為unknown。
11. 當out_valid與out_finish在同一個clock cycle內同時為1時，你必須完成整個輸出矩陣的輸出。
12. 輸出矩陣的內容必須運算正確。
13. Performance = area * total latency (total latency = cycle time * number of cycles)
14. Latency的計算從in_valid為high的下一個clock cycle開始計算，到out_valid與out_finish在同一個clock cycle內同時為1時結束。
15. Demo時會使用隱藏測資，請同學確認設計完整，能應對各種不同情況。

File Upload

1. Code檔名: **MAC_[cycle time]_dcsxxx.sv**。xxx為工作站帳號。上傳至new E3。
   範例：若你的帳號為dcs180，cycle time為5.0，檔名為"MAC_5.0_dcs180.sv"。
   請同學注意，由於本次設計允許調整cycle time，命名規則與之前有所不同。
2. Report檔名: report_dcsxxx.pdf。xxx為工作站帳號。上傳至new E3。
3. 1de請在 4/24 23:59 pm 之前上傳 / 2de 請在 5/1 23:59 pm 之前上傳。

Grading policy

- Pass the RTL, Synthesis, and GATE level simulation. 60%
- Performance 30%
- Report 10%

Note

1. Template folders and reference commands:
   a. 01_RTL/ (RTL simulation) → **./01_run**
   b. 02_SYN/ (synthesis) → **./01_run_dc**
   c. 03_GATE/ (GATE simulation) → **./01_run**
2. 本次助教在pattern中撰寫了錯誤報告的功能，能夠印出activation, weight, answer(正確解答), yours(你的答案)等矩陣。其中你的答案若顯示為?表示你沒有對此位置有過輸出， X則表示雖然有輸出但是為unknown值。請善加利用。

```
PATTERN No.  0
Operation: Matrix multiplication
                Activation                          Weight
 5  6  8  0  0  0  6 11     12 14 11  8 12  1  3  5
 4  7 14  7  8  4 15 11      5  8  0  6 12  6 13 10
 4 10  9  7  7  4  9  8      0  6  1  1  0 15  1  6
 7  4  3 12 15  4 14  8     15 10  2  9  2  9  9 15
15  1  5  5 11 14 14 15     12 12  0  8  3  1  3 14
 2 15  7  7  1  6 10  8      3 13  3  1  7  5  0  3
 8  0  1  3  2  3 10  9      5  3  4  0  0  3  2  5
 1  5  0 11  0 13 12  4     10  5  2 15 12 13  5  1
                                  Answer                                  Yours
 230  239  109  249  264  322  168  174     230  239  109  249  264  322  168    X
 481  514  166  384  330  535  289  489     481  514  166  384  330  535  289    X
 424  463  131  344  327  420  293  442     424  463  131  344  327  420  293    X
 626  582  188  435  325  365  297  573     626  582  188  435  325  365  297    X
 654  729  308  503  513  459  244  471     654  729  308  503  513  459  244    X
 364  420  117  310  359  425  334  397     364  420  117  310  359  425  334    X
 314  286  162  246  237  214  123  187     314  286  162  246  237  214  123    X
 341  389  128  210  233  283  211  323       ?    ?    ?    ?    ?    ?    ?    ?
                                        YOU FAIL!!!
                          Your output isn't complete when out_finish is high.
```

3. 報告請簡單且重點撰寫，不超過兩頁A4，並包括以下內容:
   a. 你的設計方法，包含但不限於如何加速(減少critical path)或降低面積。
   b. 基於以上，畫出你的架構圖(Block diagram)
   c. 心得報告，不侷限於此次作業，對於作業或上課內容都可以寫下。