

# NYCU-ECE DCS-2023

## Final Project

### Design: Simple CNN

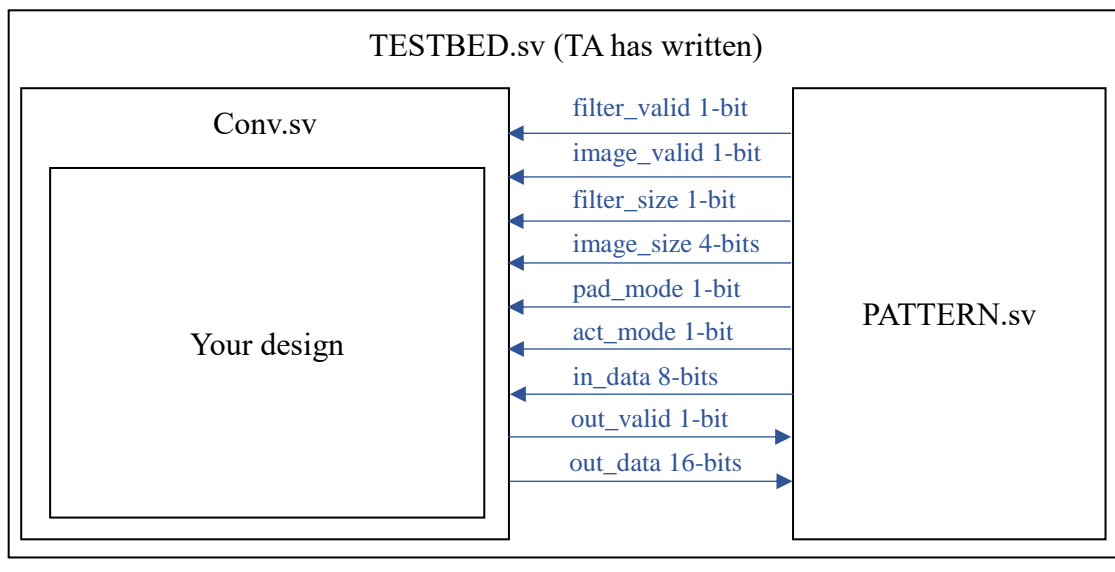
#### 資料準備

---

1. 從 TA 目錄資料夾解壓縮:  
`% tar -xvf ~dcsta01/Final.tar`
2. 解壓縮資料夾 Final 包含以下:
  - a. 00\_TESTBED/
  - b. 01\_RTL/
  - c. 02\_SYN/
  - d. 03\_GATE/

#### Block Diagram

---



這次Final Project可以自己調整clock period ! 詳情請看Specification。

CNN在AI、Deep Learning 中是常見的架構，這次作業主要做Convolution的計算。Convolution分為filter跟image 兩部分，主要利用filter在image上移動計算，而會得出一張新的特徵圖片(feature map)，以下為本次作業之介紹。

- Input

有兩種不同類別的輸入模式，分別是輸入Filter模式和輸入image模式。

- 輸入filter模式

filter\_valid拉起時代表pattern向design輸入filter的數值和convolution的configuration，包括filter size, image size, padding mode, activation function，細節會在後面解釋。

filter\_valid拉起的第一cycle，上述convolution的configuration會給值。filter\_valid會根據filter\_size為0或1分別拉起9或25 cycles，in\_data會在filter\_valid拉起時以raster scan order連續輸入filter數值，根據filter\_size為0或1持續9或25 cycles。

- 輸入image模式

image\_valid拉起時代表pattern向design輸入image的數值，此時in\_data會以raster scan order連續輸入image數值，持續image\_size<sup>2</sup> cycles。

- Pattern和Design的互動模式

首先會先進入一次輸入filter模式，代表初始化/更新filter數值和convolution的configuration，輸入filter模式結束後不需要輸出。在間隔2~6個negedge clk後，pattern會輸入image模式，與此同時design需要運算convolution的結果並輸出。輸出結束2~6個negedge clk後，pattern可能進入輸入image模式或輸入filter模式，前者代表繼續沿用原本的filter數值和configuration並使用新輸入的image運算convolution的結果後輸出，後者代表要更新filter數值和configuration並進入新一輪循環。

filter\_valid和image\_valid不會同時拉起，out\_valid可以和image\_valid重疊，而本次project推薦使用pipeline架構，如此out\_valid和image\_valid便會重疊。

總而言之，pattern和design會反覆執行以下流程：

輸入filter→in/out歸零→輸入image(design同時輸出)→in/out歸零

(藍色部分重複多次)

- Filter

根據filter\_size的值決定filter大小，0代表3x3 filter，1代表5x5 filter，數值範圍-128~127。當filter\_valid拉起時，由in\_data一cycle給一個pixel，給值的順序依照raster scan order，從左到右，從上往下。

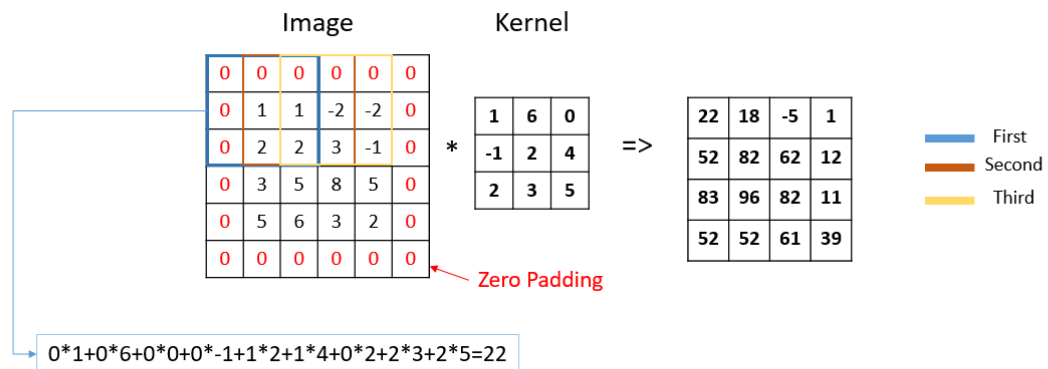
- Image

邊長為image\_size(邊長範圍3~8)的正方形image，數值範圍-128~127。當image\_valid拉起時，由in\_data一cycle給一個pixel，共image\_size<sup>2</sup> cycles，給值的順序依照raster scan order，從左到右，從上往下。

- Convolution

Convolution利用filter在image上移動計算，將image pixel和filter相對應位置做內積，結果即為filter中心在image對應位置的結果，公式和範例圖片如下：

$$R(x, y) = \sum_{x', y'} (Kernel(x', y') \times Image(x + x', y + y'))$$



以上圖為例，黑色數字部分為4x4的input image，先將filter中心對準image最左上角pixel，可以發現filter超出image範圍，因此超出的範圍需要padding(補值，細節見後面段落)，本範例為zero padding，也就是image外圍紅色的0。filter和image相對應的位置數值相乘的總和為22，也就是左上角位置的convolution結果。接著移動filter一格，重複上述步驟即可得出完整convolution結果。

- Padding

本作業有zero padding和replication padding兩種padding模式。Zero padding就是將image周圍filter會超出的範圍補上0，如上一段的範例；Replication padding就是image周圍補上最接近的image邊緣pixel的數值，i.e. image邊緣外在邊上的pixel等同與之相鄰的image邊緣內側pixel，在角落的pixel等同image同角落的pixel，如下圖所示。請留意，當filter size為5x5時會需要padding兩層。

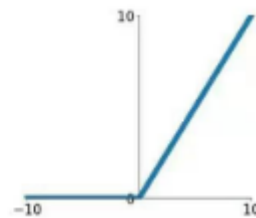
Replication Padding

1	1	1	-2	-2	-2
1	1	1	-2	-2	-2
2	2	2	3	-1	-1
3	3	5	8	5	5
5	5	6	3	2	2
5	5	6	3	2	2

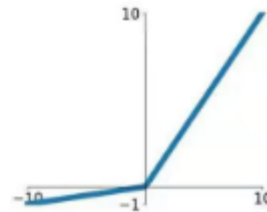
- Activation Function

實務上，在Convolution運算結束後會讓結果再經過一個函數，最後的結果才構成feature map。這個函數稱做activation function，本project所使用的ReLU和Leaky ReLU為Deep learning的常見的activation function，公式如下圖所示：

**ReLU**  
 $\max(0, x)$



**Leaky ReLU**  
 $\max(0.1x, x)$



請注意0.1x的結果請無條件捨去，也就是 $\lfloor 0.1x \rfloor$ 。(因為用到0.1x的條件是 $x < 0$ )

- Output

將convolution完並經過activation function的結果以raster scan order，從左到右，從上往下，一次輸出一個pixel，總共輸出 $\text{image\_size}^2$ 。

請注意，由於out\_data只有16bit，如果運算結果數值超過 $[-2^{15}, 2^{15}-1]$ 請分別以 $-2^{15}$ 和 $2^{15}-1$ 輸出。

- Pattern

請注意，這次final project不會提供pattern，助教會在兩到三周後提供一個pattern範本(pattern數較少且不會涵蓋所有configuration組合)。同學可利用前兩周時間練習寫pattern。

## Inputs

Signal name	Number of bit	Description
clk	1-bit	Clock
rst_n	1-bit	Asynchronous active-low reset
filter_valid	1-bit	拉起時代表初始化/更新 filter 數值和 convolution 屬性，根據 filter_size 為 0 或 1 分別拉起 9 或 25 cycles。 filter_size, image_size, pad_mode, act_mode 會在 filter_valid 拉起的第一 cycle 給值，in_data 會隨著 filter_valid 拉起連續給值 9 或 25 cycles
image_valid	1-bit	拉起時代表輸入 image 數值，連續拉起 image_size <sup>2</sup> cycles
filter_size	1-bit	0 代表 3x3 filter，1 代表 5x5 filter，在 filter_valid 拉起的第一 cycle 給值
image_size	4-bit	image 的邊長，範圍 3~8，在 filter_valid 拉起的第一 cycle 給值
pad_mode	1-bit	Padding mode，0 代表 zero padding，1 代表 replication padding，在 filter_valid 拉起的第一 cycle 給值
act_mode	1-bit	Activation function，0 代表 ReLU，1 代表 Leaky ReLU，在 filter_valid 拉起的第一 cycle 給值
in_data	8-bit	有號數，在 filter_valid 拉起時以 raster scan order 連續輸入 filter 數值，根據 filter_size 為 0 或 1 持續 9 或 25 cycles；在 image_valid 拉起時以 raster scan order 連續輸入 image 數值，持續 image_size <sup>2</sup> cycles

## Outputs

Signal name	Number of bit	Description
out_valid	1-bit	拉起時輸出結果，持續 image_size <sup>2</sup> cycles。
out_data	16-bit	有號數，在 out_valid 拉起時以 raster scan order 連續輸出結果，持續 image_size <sup>2</sup> cycles。

## Specifications

1. Top module name: **Conv**(File name : **Conv.sv**)
2. 在非同步負準位 reset 後，所有的 output 訊號必須全部歸零。
3. 在 image input 和 output 可重疊，output 要在 input 結束 image\_valid 歸零後的 300 cycles 內輸出。
4. Output 要連續輸出 image\_size<sup>2</sup> cycles，不能多不能少，且輸出的答案要正確。
5. 所有 Output 訊號要在輸出結束後全部歸零。
6. Clock period 最大 8.0ns，以 0.1ns 為單位，例如 5.1ns，不要有 5.17ns。要更改 clock period 要修改兩個地方再去跑 02\_SYN 和 03\_GATE，一個是 PATTERN.sv 第 1 行，另一個是 syn.tcl 第 25 行，如下圖所示。

```
1  define CYCLE_TIME 6.0
2  module PATTERN(
3  // Input signals
```

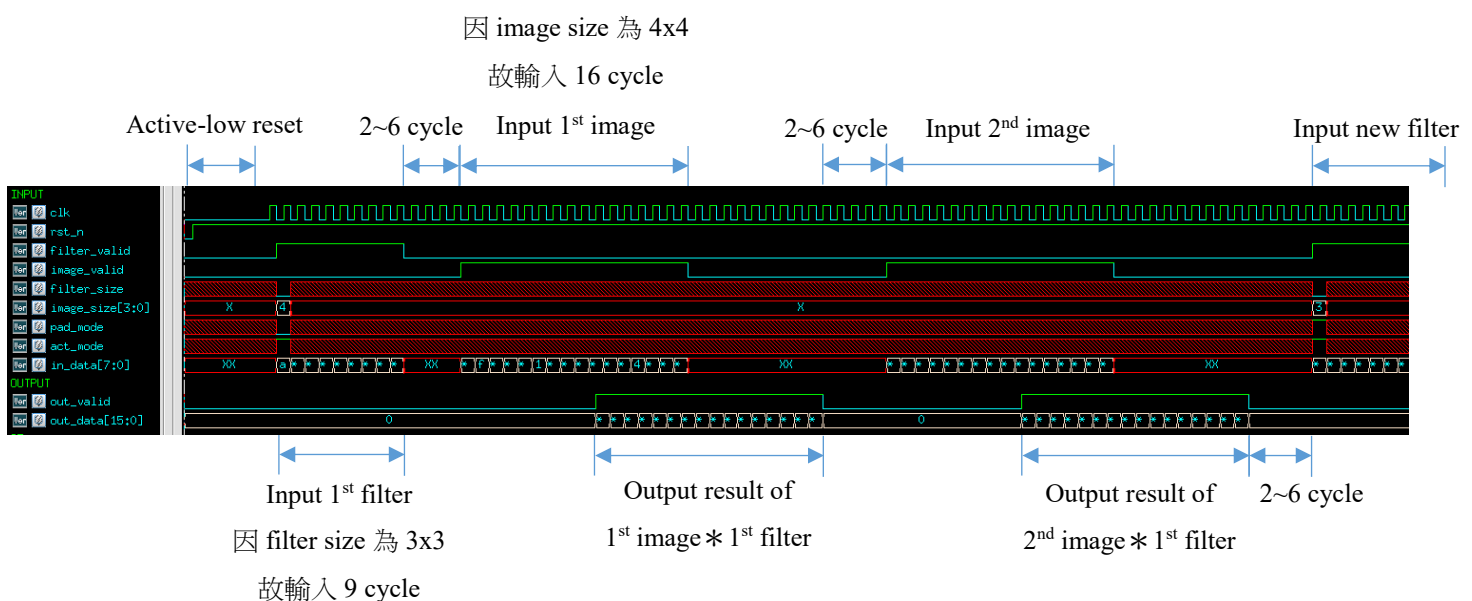
Final/00\_TESTBED/PATTERN.sv

```
22  # Global Parameters
23  #=====
24  set DESIGN "Conv"
25  set CYCLE 6.0
26
```

Final/02\_SYN/syn.tcl

7. Input delay = 0.5 \* clock period; Output delay = 0.5 \* clock period
8. 02\_SYN result 不行有 error、不能有任何 latch、不可以 timing violation。
9. 03\_GATE 不能有 任何 timing violation。
10. 03\_GATE 的 Latency 要與 01\_RTL 相同。
11. Coding style 和 for loop 的限制解除。

## Example waveform



## 上傳檔案

---

1. 請將Final/01\_RTL裡的Conv.sv依以下命名規則重新命名後上傳至E3。  
命名規則：Conv\_{clock cycle time}\_dcsxxx.sv，xxx為工作站帳號號碼, clock cycle time請取到小數第一位。  
例如：李小任的工作站帳號為dcs230，clock cycle time為6ns，他的檔名應為Conv\_6.0\_dcs230.sv。命名錯誤扣5分
2. report\_dcsxxx.pdf, xxx is your server account. 上傳至E3。
3. Deadline:  
1 demo: 6/21 (Wed) 23:59  
2 demo: 6/27 (Tue) 23:59 分數7折

## Grading policy

---

1. Pass the RTL& Synthesis simulation. 60%
2. Performance 30%
  - Performance = Area \* Computation time
  - Computation time  
= (image\_valid開始為1到out\_valid結束回到0所需cycle數) \* clock cycle time
3. Report 10% : 6/21 (Wed) 23:59前上傳(與demo1 deadline相同)
4. Performance 排名達到修課人數前10%的同學，依排名可額外加1~5分

## Note

---

Template folders and reference commands:

1. 01\_RTL/ (RTL simulation) → **./01\_run**
2. 02\_SYN/ (synthesis) → **./01\_run\_dc**
3. 03\_GATE/ (gate-level simulation) → **./01\_run**

報告請簡單且重點撰寫，不超過兩頁A4，並包括以下內容

1. 描述你的設計方法，包含但不限於如何加速(減少critical path)或降低面積。
2. 基於以上，畫出你的架構圖(Block diagram)
3. 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。
4. 遇到的困難與如何解決。