

# Homework 5 – Convolution Accelerator for VGG-16

Handout: 2021/12/07

Due 2021/12/21

1. 使用 Verilog HDL 來設計一個 DNN 硬體加速器，架構圖如 Fig 1.。

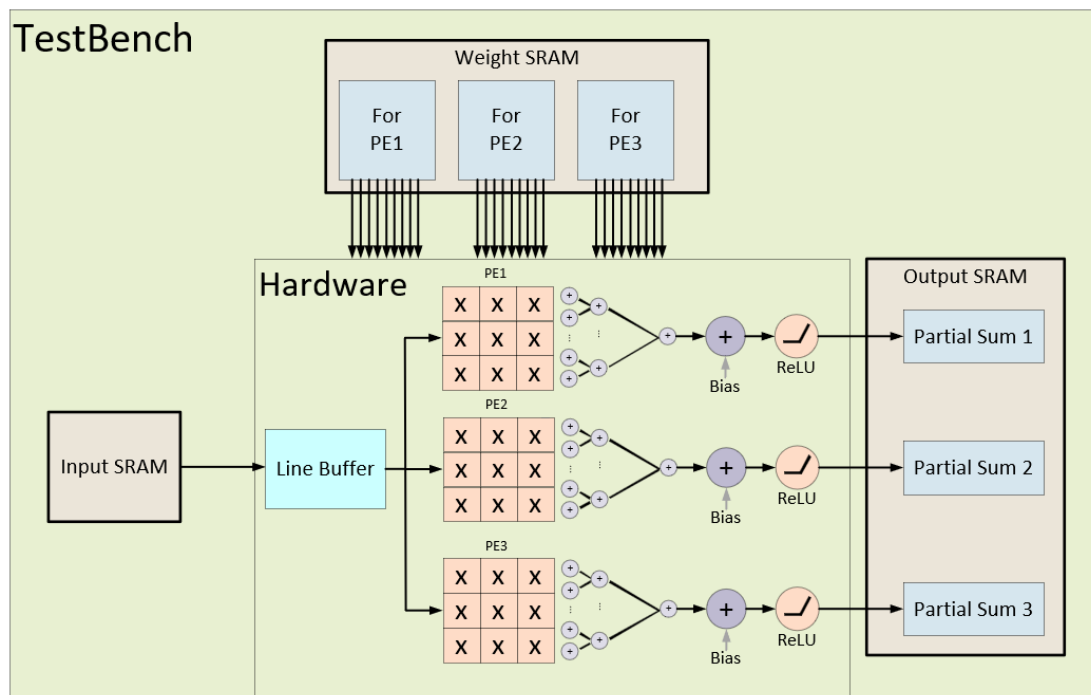


Fig 1. 整體硬體架構示意圖(可自由調整 PE 個數)

2. 使用設計的 DNN 硬體加速器，運算 Fig. 2 的 VGG 前兩層 Convolution Layer，並將 Input data 進行 Fig. 3 的 Padding 後，使 Convolution 運算後的 Output 長和寬保持在 224x224，透過 Fig. 4 的 LineBuffer 降低 Input SRAM 的讀取次數減少功耗，透過 PE 內的乘法器將 Input 和 Weight 進行相乘後，經過 Adder Tree 相加，並經由 Fig. 5 的 ReLU 輸出 Partial Sum 到 Output SRAM，當輸出為最終輸出的時候，加上相對應的 Bias 後，將輸出的每層 Channel 存成.bmp 圖片檔，第一層 Convolution 輸出尺寸為 224x224x64，可輸出 64 張.bmp 圖片檔，第二層 Convolution 輸出尺寸為 224x224x64，也可輸出 64 張圖.bmp 圖片檔，共可輸出 128 張.bmp 圖片檔。

3. 提供檔案內容如下，資料擺放順序如 Fig. 6，以下資料皆為 16 Bit 並為十六進制。

1. Input (Input\_data\_hex.txt)
2. Weight (conv1\_kernel\_hex.txt 、 conv2\_kernel\_hex.txt)
3. Bias (conv1\_bias\_hex.txt 、 conv2\_bias\_hex.txt)

Tabel 1. VGG-16 前兩層 CONV Layer

#Input Ch.	#Output Ch.	Input Size	Kernel Size	Stride
3	64	226x226(Padding)	3x3	1
64	64	226x226(Padding)	3x3	1

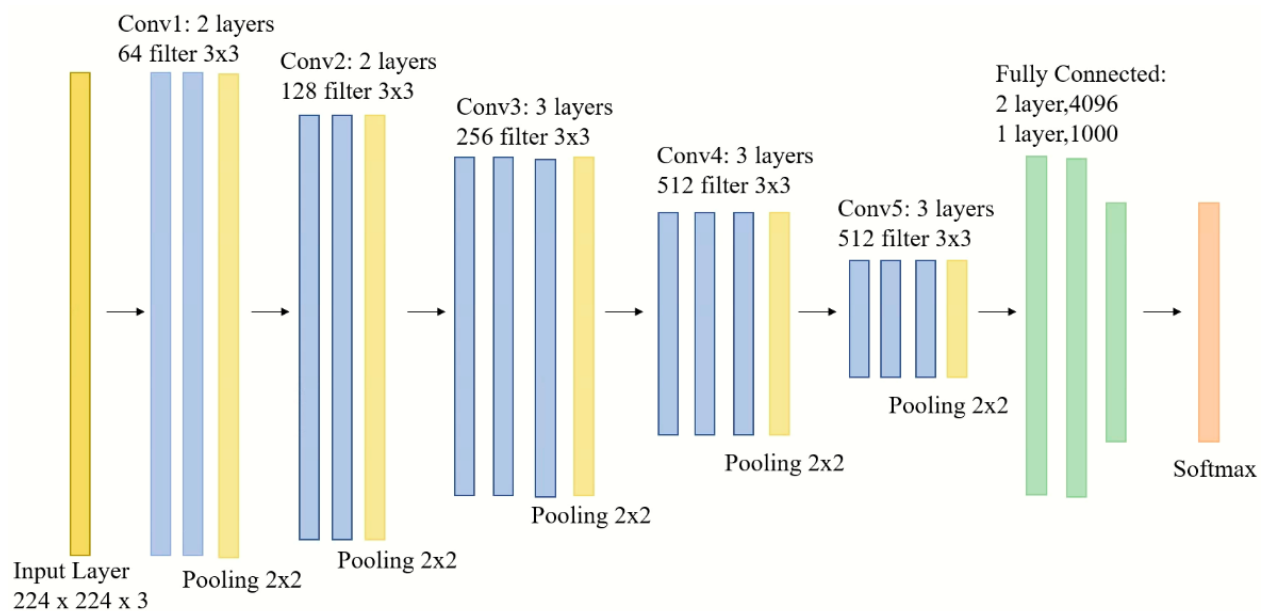


Fig 2. VGG-16

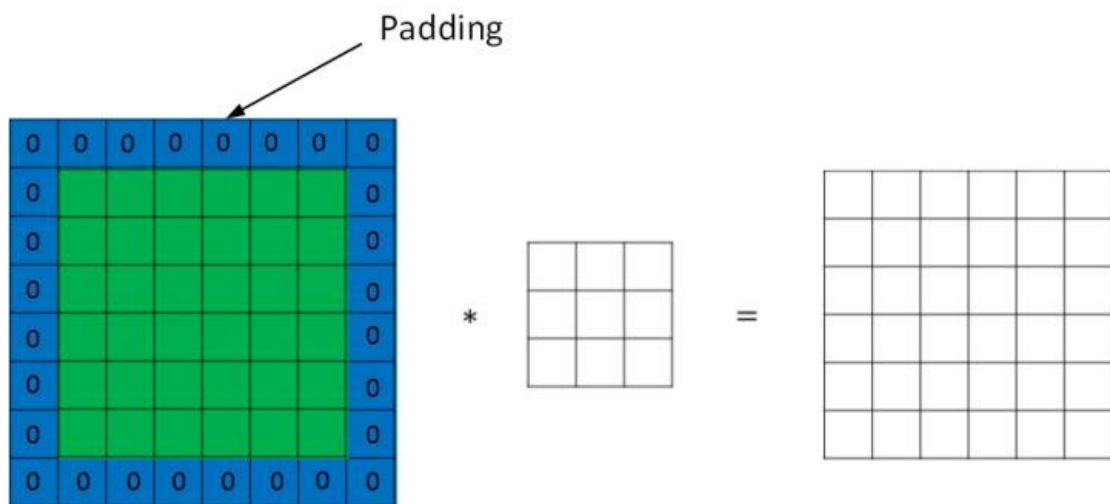


Fig 3. Padding

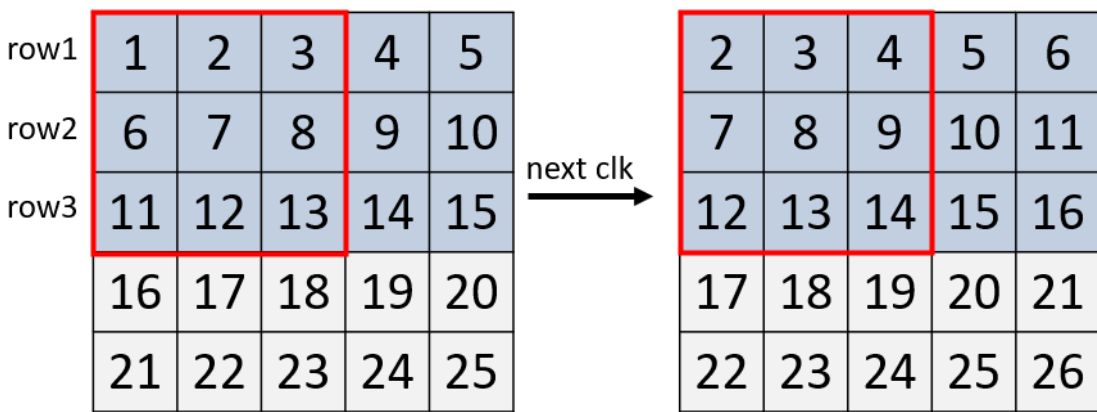
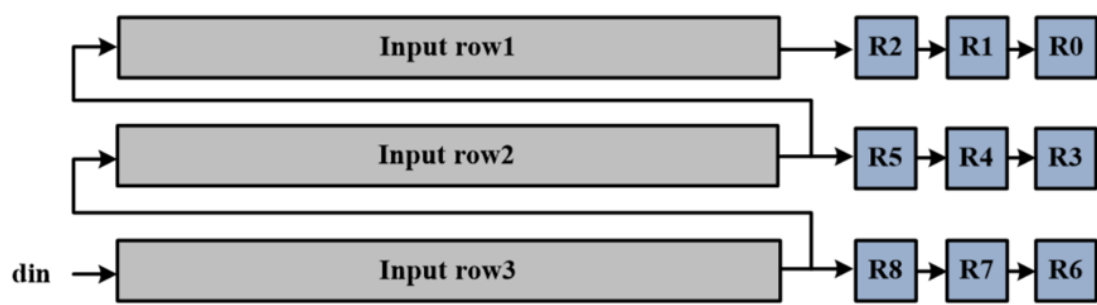


Fig 4. LineBuffer

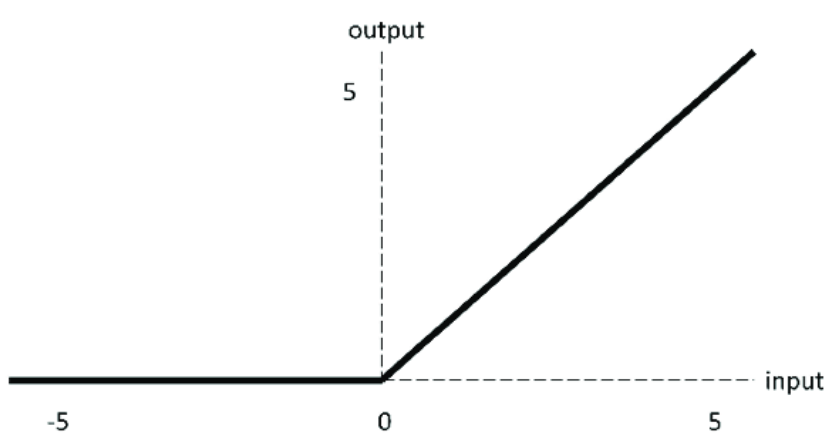


Fig 5. ReLU：輸入小於等於 0 時，輸出則為 0；輸入大於 0 時，輸出等於輸入。

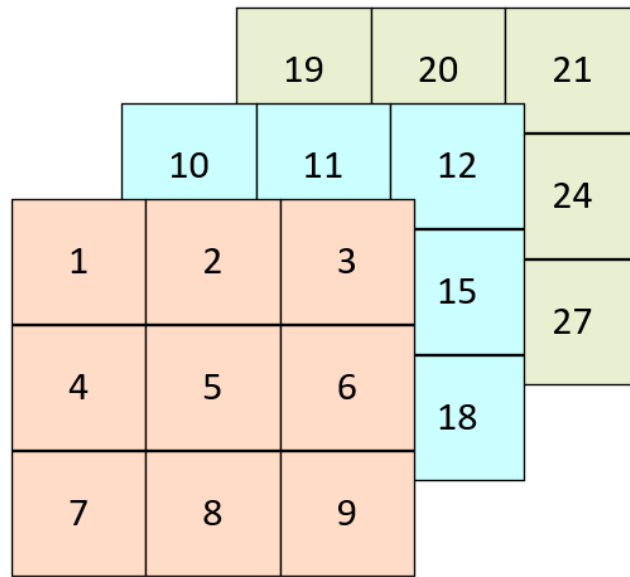


Fig 6. txt 檔內 Input、Weight 擺放順序

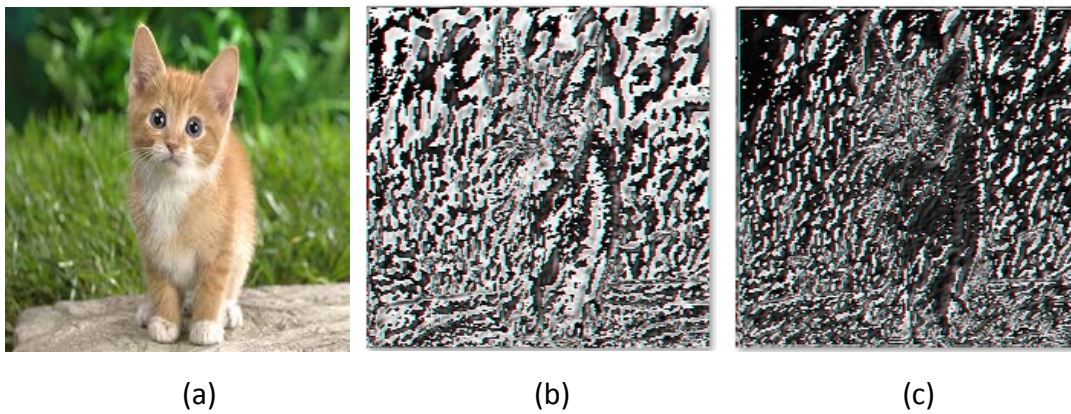


Fig 7. (a)輸入圖片 和(b)Conv1、(c)Conv2 Output Channel 1 的輸出圖片



Fig 8. Convolution 第一層輸出圖片





Fig 9. Convolution 第二層輸出圖片

## Report Requirement

### 1. Design Compiler(80%)

檔案類須含有:

- I. Testbench(10%)
- II. Verilog RTL code & Gate-level code(20%)
  - LineBuffer (5%)
  - PE (5%)
  - Adder Tree(5%)
  - ReLU(5%)
- III. Image
  - Conv1 image \*64(10%)
  - Conv2 image \*64(10%)

word 報告類須含有:

- IV. 架構圖解釋(10%)
- V. Area 資訊和 critical path 資訊(10%)
- VI. 心得(10%)

### 2. Xilinx Vivado(20%)

Project name 需為: HDL\_HW5\_MXXXXXXXXXX

檔案類須含有:

- I. HDL\_HW5\_MXXXXXXXXXX.xpr.zip(5%)
- II. xdc(5%)
- III. wcfg(5%)

word 報告類須含有:

- IV. Project Summary-Overview 截圖(5%)

以上打包成 HDL\_HW5\_MXXXXXXXXXX.7z 壓縮檔並繳交