

NYCU-ECE DCS-2022

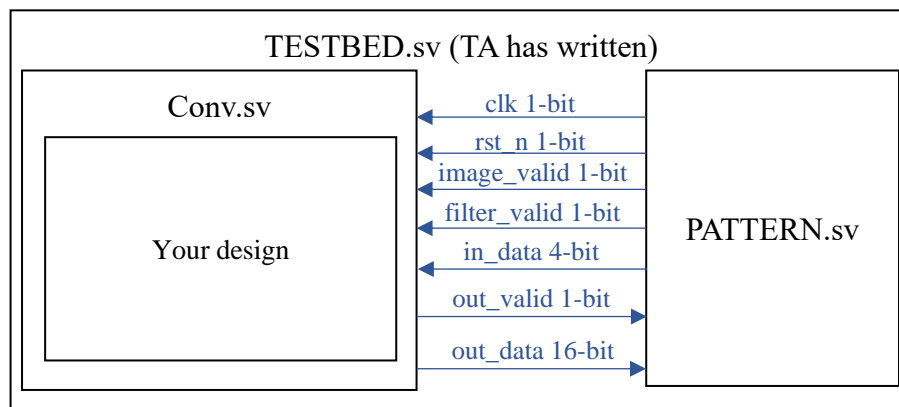
HW05

Design: Convolution

資料準備

1. 從 TA 目錄資料夾解壓縮:
% tar -xvf ~dcsta01/ HW05.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/
 - e. 09_UPLOAD/

Block Diagram



設計描述

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

CNN在AI、Machine Learning中是常見的架構，這次作業主要做兩次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)，以下以這次作業為例分為五個部分介紹，分別為filter、image、convolution、output和pattern。

- Filter

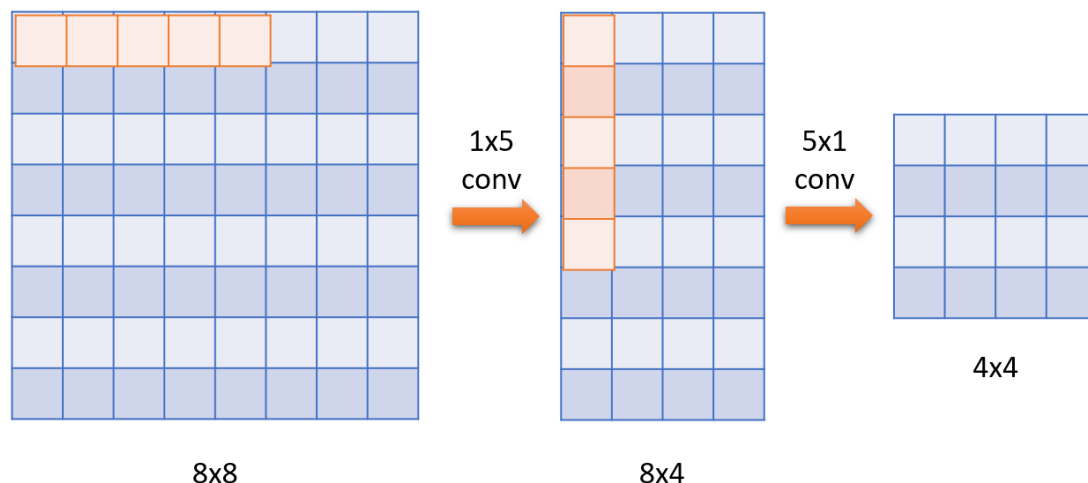
有1x5和5x1兩個filter，各5個數字，數值範圍-8~7。1x5的filter和5x1的filter都由in_data給值，先給1x5的filter再給5x1的filter，給值的順序依照raster scan order，從左到右，從上往下。

- Image

輸入一張8x8的image，總共64個pixel，數值範圍-8~7。由in_data一次給一個pixel，filter都給完才給image，給值的順序依照raster scan order，從左到右，從上往下。

- Convolution

這次計算做兩次convolution，輸入8x8的image先和1x5的filter做convolution，得到8x4的圖像，再把8x4的圖像和5x1的filter做convolution，得到4x4的圖像。



詳細的convolution步驟請參考DCS_HW05_conv.pdf

- Output

將第二次convolution完的4x4圖像依照raster scan order，從左到右，從上往下一次輸出一個pixel，總共輸出16 cycles。

● Pattern

注意，這次的pattern也要由你們寫，助教僅提供10組測資，請想辦法讀檔讀進去測(之前上課有教過)，或是直接在pattern生成測資，並產生更多測資去測試。

input.txt 範例

```

1 10
2 5 6 -6 6 2
3 -7
4 -4
5 0
6 7
7 7
8 -6 4 3 -7 4 -4 1 0
9 7 7 -2 5 4 2 -5 -6
10 7 2 2 3 -6 2 4 -6
11 -1 -8 -6 -3 -1 -6 -4 -4
12 4 5 3 7 -1 -7 0 5
13 -6 6 -8 -8 2 -1 3 -4
14 -2 2 -4 -1 3 7 6 5
15 6 4 -8 -2 4 -3 7 -5
16 6 -3 -5 -4 1
17 -1
18 -3
19 5
20 1
21 0
22 6 0 -6 3 0 5 -6 5
23 -4 4 4 3 7 5 -4 1
24 4 6 -4 -1 -7 -7 -6 -3
25 4 -6 0 -7 -1 -2 -6 0
26 -2 1 -6 -5 -7 -4 5 -2
27 1 -1 1 6 7 4 1 -7
28 -7 -8 -4 -6 -8 -2 0 -5
29 -8 -3 2 5 4 6 -6 -7

```

第1行是測資數量 Ex: 10筆

第2行是1x5的filter

Ex:

5	6	-6	6	2
---	---	----	---	---

第3~7行是5x1的filter

Ex:

-7
-4
0
7
7

第8~15行是8x8的image

第2~15行是第一筆測資，第16~29行是第二筆測資，以此類推

output.txt 範例

```

1 143 -1421 64 335
2 -479 112 -875 602
3 113 948 -874 250
4 839 790 28 362
5
6 313 441 411 361
7 365 -118 111 -238
8 -103 225 -7 -127
9 -200 -511 -401 -49

```

第1~4行是4x4的輸出

第1~4行是第一筆測資，第6~9行是第二筆測資，以此類推

Inputs

Signal name	Number of bit	Description
clk	1-bit	Clock
rst_n	1-bit	Asynchronous active-low reset
image_valid	1-bit	為 1 時代表 in_data 給 image 資料，連續給 64 cycles
filter_valid	1-bit	為 1 時代表 in_data 給 filter 資料，連續給 10 cycles，先給 1x5 再給 5x1 的 filter
in_data	4-bit	連續給 74 cycles (10+64)，根據 image_valid 和 filter_valid 判斷是哪種資料，是 signed 數值範圍-8~7

Outputs

Signal name	Number of bit	Description
out_valid	1-bit	必須在 image_valid 變 0 後 300 cycles 內拉起，並持續 16 cycles
out_data	16-bit	依序輸出計算完後的結果，共 16 cycles，是 signed

PS: 輸入和輸出都是依照 raster scan order，從左到右，從上往下。

1st cycle 11th cycle

↓ ↓

in_data order: ABCDEFGHIJ0123456789...

The diagram illustrates three different data structures:

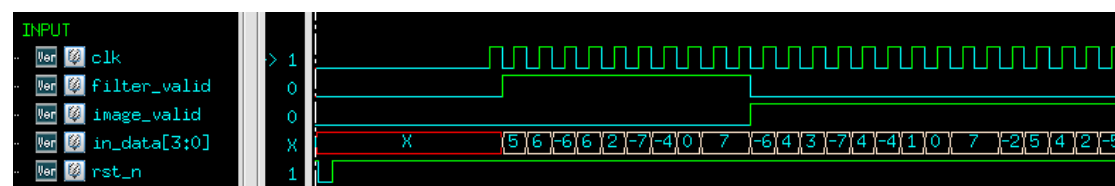
- 1D Array:** A horizontal row of five cells labeled A, B, C, D, and E.
- 1D Linked List:** A vertical column of five cells labeled F, G, H, I, and J.
- 2D Array:** A grid with 8 rows and 8 columns. The first row contains indices 0 through 7. The first column contains indices 8 through 15. The remaining cells are empty, representing a 2D array structure.

Specifications

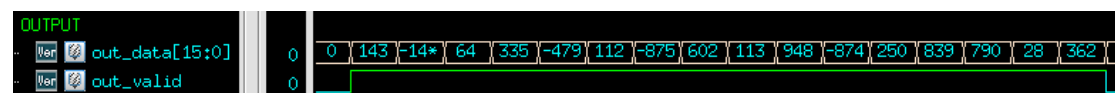
1. Top module name: **Conv**(File name : **Conv.sv**)
2. 在非同步負準位 reset 後，所有的 output 訊號必須全部歸零
3. Input 時就可 output，output 最晚要在 input 結束後的 300 cycles 內開始輸出
4. Output 要輸出連續 16 cycles，不能多不能少，且輸出的答案要正確
5. 所有 Output 訊號要在輸出結束後全部歸零
6. 每筆測資之間間隔 2 個 negedge clk
7. 02_SYN result 不行有 **error** 且不能有任何 **latch**
8. Input delay = 0.5 * clock period. Output delay = 0.5 * clock period
9. Clock period 最大 6.0ns，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 02，一個是 PATTERN.sv 第 1 行，另一個是 syn.tcl 第 25 行
10. **Separate your combination and sequential blocks**
11. 這次作業要自己寫 PATTERN 去測試 Design，寫法可以參考之前助教給的 PATTERN，PATTERN 必須要測試 Design 是否符合 Specification 2~5

Example waveform

Input



Output



上傳檔案

1. Code和clock period使用09_upload上傳，方法請參考下方Note
2. report_dcsxx.pdf, xx is your server account. 上傳至new E3
3. 1 demo請在 5/12 15:30 上課之前上傳
4. 2 demo請在 5/19 15:30 上課之前上傳

Grading policy

1. Pass the RTL & Synthesis & Gate-level simulation (60%)
2. Performance = Area × Total cycles × Clock period (30%)
3. Report (10%)
4. Combinational、sequential Logic沒有分開寫 (-5%)

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`
4. 09_UPLOAD/ (upload) → `./09_upload`

※01_upload上傳檔案與 clock period 使用教學:

- (1) `./01_upload [your clock period]`

範例:

```
linux01 [HW05/09_UPLOAD]% ./01_upload 4.5
module Conv(
    // Input signals
    clk,
    rst_n,
    image_valid,
    filter_valid,
    in_data,
    // Output signals
    out_valid,
    out_data
);

//-----
//  INPUT AND OUTPUT DECLARATION
//-----
input clk, rst_n, image_valid, filter_valid;
input [3:0] in_data;
output logic [15:0] out_data;
output logic out_valid;

endmodule

-----
Check your code and clock period.
-----
Your clock period is 4.5 ns.
-----
The 1st demo deadline is Thu May 12 15:30:00 CST 2022 , a
7 15:30:00 CST 2022
It is Sat Apr 23 16:01:52 CST 2022 now!
It will upload to demo1.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):y
Upload done!
```

如果clock period輸入錯誤會無法上傳。。

```
linux01 [HW05/09_UPLOAD]% ./01_upload
Please input clock period. -> Ex: ./01_upload 6.0
linux01 [HW05/09_UPLOAD]% ./01_upload -2
Clock period must be positive.
linux01 [HW05/09_UPLOAD]% ./01_upload 7.1
Clock period has a limit of 6.0 ns.
linux01 [HW05/09_UPLOAD]% ./01_upload JohnCena
JohnCena is not a number.
```

(2) ./02_download demoX (X=1 or 2)

用02_download確認自己是否上傳正確，下載下來的檔名會寫你上傳的 clock period。

```
linux01 [HW05/09_UPLOAD]% ./02_download demo1
Download done!
linux01 [HW05/09_UPLOAD]% ll
total 12
-rwxr-xr-- 1 dcsta02 dcs 2653 Apr 23 15:55 01_upload
-rwxr-xr-x 1 dcsta02 dcs 676 Apr 23 15:59 02_download
lrwxrwxrwx 1 dcsta02 dcs 17 Apr 21 23:59 Conv_sv -> /01 RTL
-rw-r--r-- 1 dcsta02 dcs 495 Apr 23 16:03 Conv_4.5_dcsta02.sv
```

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。