

# REPORT

## Thiết kế và thực hiện khối tính nhân chập 2-D dùng cho CNN

Ver 2.0

13/06/2022

	Full name	Function	Date
Written by	Ngô Minh Khánh Đinh Tiến Dương		
Verified by	Nguyễn Kiêm Hùng		
Approved by	Nguyễn Kiêm Hùng		

<b>Abstract (from 5 to 10 lines)</b>
Thiết kế mức RTL, lập mô hình VHDL, mô phỏng ModelSIM và triển khai FPGA của tích chập 2D đơn giản cho CNN.

<b>Keywords</b>
SM, FSMD, ModelSim, 2D Convolution, VHDL, RTL

<b>Work context</b>
<ol style="list-style-type: none"><li>1. Tìm hiểu kiến thức cơ bản về VHDL, ModelSim</li><li>2. Phân tích yêu cầu bài toán (Tìm hiểu thuật toán nhân chập)</li><li>3. Bắt đầu viết các tệp thành phần bộ nhân chập</li><li>4. Kết hợp các tệp thành phần thành tệp phần thành khối nhân chập hoàn chỉnh</li><li>5. Viết và chạy testbench và kiểm tra kết quả đầu ra</li><li>6. Hoàn thiện báo cáo dựa trên kết quả đầu ra</li></ol>

## Document History

Version	Time	Revised by	Description
V0.1	19/04/2022	Nguyễn Kiên Hùng	Original Version
V0.2	13/06/2022	Ngô Minh Khánh Đinh Tiến Dương	

## MỤC LỤC

Document History .....	3
MỤC LỤC .....	3
1. Giới thiệu.....	5
2. Yêu cầu .....	6
2.1. Yêu cầu đối với thiết kế:.....	6
2.2. Định nghĩa giao diện vào/ra .....	7
3. Thuật toán.....	8
4. Thiết kế mức RTL .....	8
4.1. Mô hình máy FSMMD .....	8
4.2. Đơn vị xử lý dữ liệu (Datapath) .....	10
4.3. Đơn vị điều khiển (Control Unit) .....	11
4.4. Sơ đồ khối tổng thể.....	13
5. Mô hình hóa bằng VHDL .....	13
6. Mô phỏng/thực thi và đánh giá .....	14
7. Kết luận .....	14
Appendix A: VHDL Code.....	15
List of Figures .....	16
List of Tables.....	16

<b>References .....</b>	<b>17</b>
-------------------------	-----------

## 1. Giới thiệu

*(Introduction to the motivation, Objectives, and main Contents of the project)*

**Mục tiêu:** Vận dụng các kiến thức, kỹ năng đã được học để thiết kế, mô phỏng và thực thi một mô-đun phần cứng thực hiện tính tích chập  $J = 2DConV(I, K)$  giữa hình ảnh lõi vào  $I$  với một ma trận kernel  $K$  ([1]). Trong đó, mỗi pixel trong hình ảnh tích phân  $J$  đại diện cho tổng tích lũy của tích điểm-điểm giữa ma trận  $K$  với một ma trận cùng thước được trích xuất từ ma trận đầu vào  $I$ . Phép chuyển đổi hình ảnh được mô tả bằng ví dụ sau.

**Ví dụ:** nếu hình ảnh đầu vào  $I$  là ma trận có kích thước  $5 \times 5$  như sau:

$$I = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

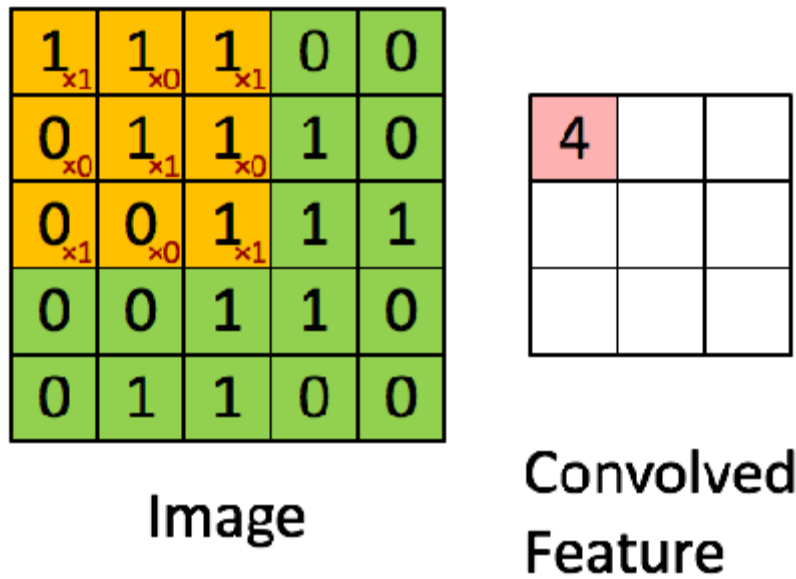
Và kernel có kích thước  $3 \times 3$  như sau:

$$K = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

thì kết quả tính toán của khối 2D-Convolution trả về hình ảnh  $J$  có kích thước  $3 \times 3$  như sau:

$$K = \begin{pmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

Quá trình tính toán được minh họa trong Hình 1.



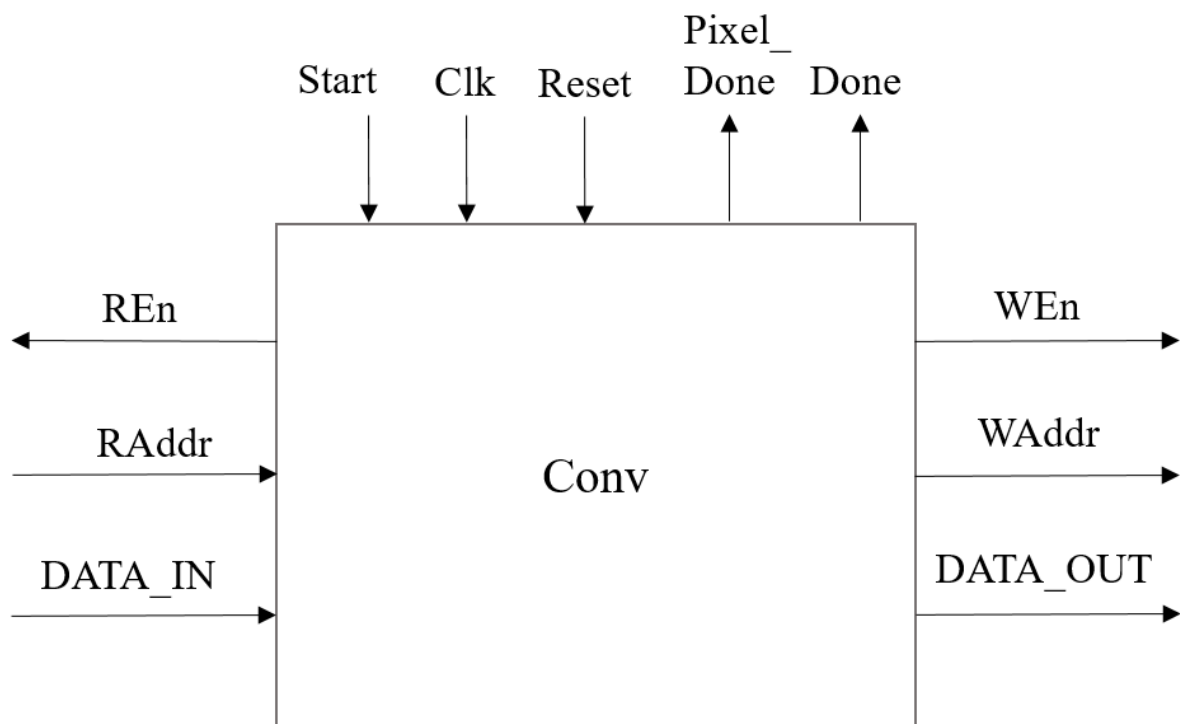
*Hình 1. Ví dụ minh họa phép tính tích chập 2D.*

Lưu ý rằng pixel có tọa độ  $(r,c)$  – (hàng, cột) – trong ảnh lõi ra được tính bằng cách nhân chập điểm – điểm giữa ma trận kernel và ma trận  $3 \times 3$  có tâm nằm ở vị trí có tọa độ  $(r + 1, c + 1)$  trong ảnh lõi vào.

## 2. Yêu cầu

### 2.1. Yêu cầu đối với thiết kế:.

- CPU có thể viết các phần tử của ma trận đầu vào tới bộ nhớ đệm bên trong bộ nhân chập
- Khối Conv có giao diện ghép nối tới CPU sao cho CPU kích hoạt quá trình tính toán của khối Conv bằng các đặt tín hiệu Start = '1'.
- Sau khi quá trình tính hình ảnh tích phân hoàn thành, khối Conv sẽ báo cho CPU biết bằng cách đặt tín hiệu Done = '1';
- Khối Conv có 1 giao diện ghép nối tới bộ nhớ để đọc hình ảnh đầu vào
- Khối Conv có 1 giao diện ghép nối tới bộ nhớ để ghi dữ liệu đầu ra



*Hình 2. Giao diện ghép nối I/O.*

## 2.2. Định nghĩa giao diện vào/ra

TT	Port	Direction	Width	Meaning
1	Start	IN	1	Tín hiệu bắt đầu nhân chập
2	Clk	IN	1	Xung clock hệ thống
3	Reset	IN	1	Xung reset hệ thống
4	Pixel_Done	OUT	1	Tín hiệu báo đã tính xong 1 pixel đầu ra
5	Done	OUT	1	Tín hiệu báo tính xong ma trận đầu ra
6	REn	OUT	1	Tín hiệu báo đọc ma trận từ bộ nhớ ngoài vào bộ nhớ đệm MA
7	RAddr	IN	N	Địa chỉ pixel vào
8	DATA_IN	IN	N	Dữ liệu vào

9	WEN	OUT	1	Tín hiệu báo ghi ma trận từ bộ nhớ đệm MB ra bộ nhớ ngoài
10	WAddr	OUT	N	Địa chỉ pixel ra
11	DATA_OUT	OUT	N	Dữ liệu ra

*Bảng 1. Định nghĩa giao diện vào ra.*

### 3. Thuật toán

Cho A là ma trận ảnh đầu vào, B là ma trận ảnh đầu ra, K là ma trận Kernel. Ta có thuật toán sau :

Begin : Wait for Start = '1'

Done = '0'

For br = 0 to (rowB - 1)

    For bc = 0 to (colB - 1)

        For kr = 0 to (rowK - 1)

            For kc = 0 to (colK - 1)

$B[br][bc] += A[br + kr][bc + kc] * K[kr][kc]$

            End for

        End for

    End for

End for

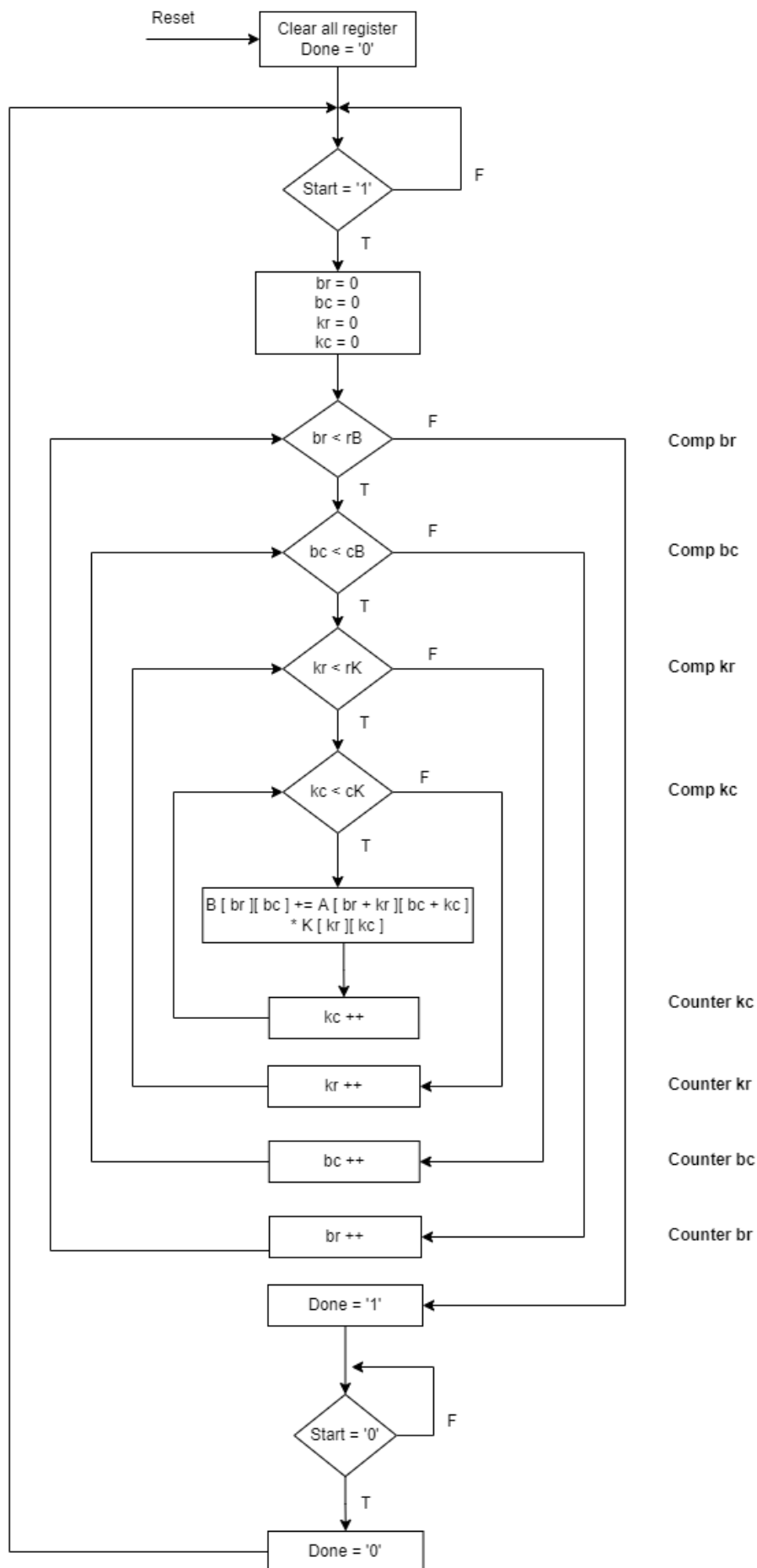
### 4. Thiết kế mức RTL

#### 4.1. Mô hình máy FSM

Để bắt đầu xây dựng mạch điện thực hiện chức năng nhân chập, trước tiên chúng ta chuyển đổi thuật toán thành sơ đồ máy trạng thái phức tạp FSM như Hình 3. Như tên gọi của nó các trạng thái của FSM có thể bao gồm các biểu thức số học của tổ hợp các đầu vào và đầu ra bên ngoài hoặc biến.

Sơ đồ trạng thái FSM như dưới đây:





*Hình 3: Mô hình máy FSMD.*

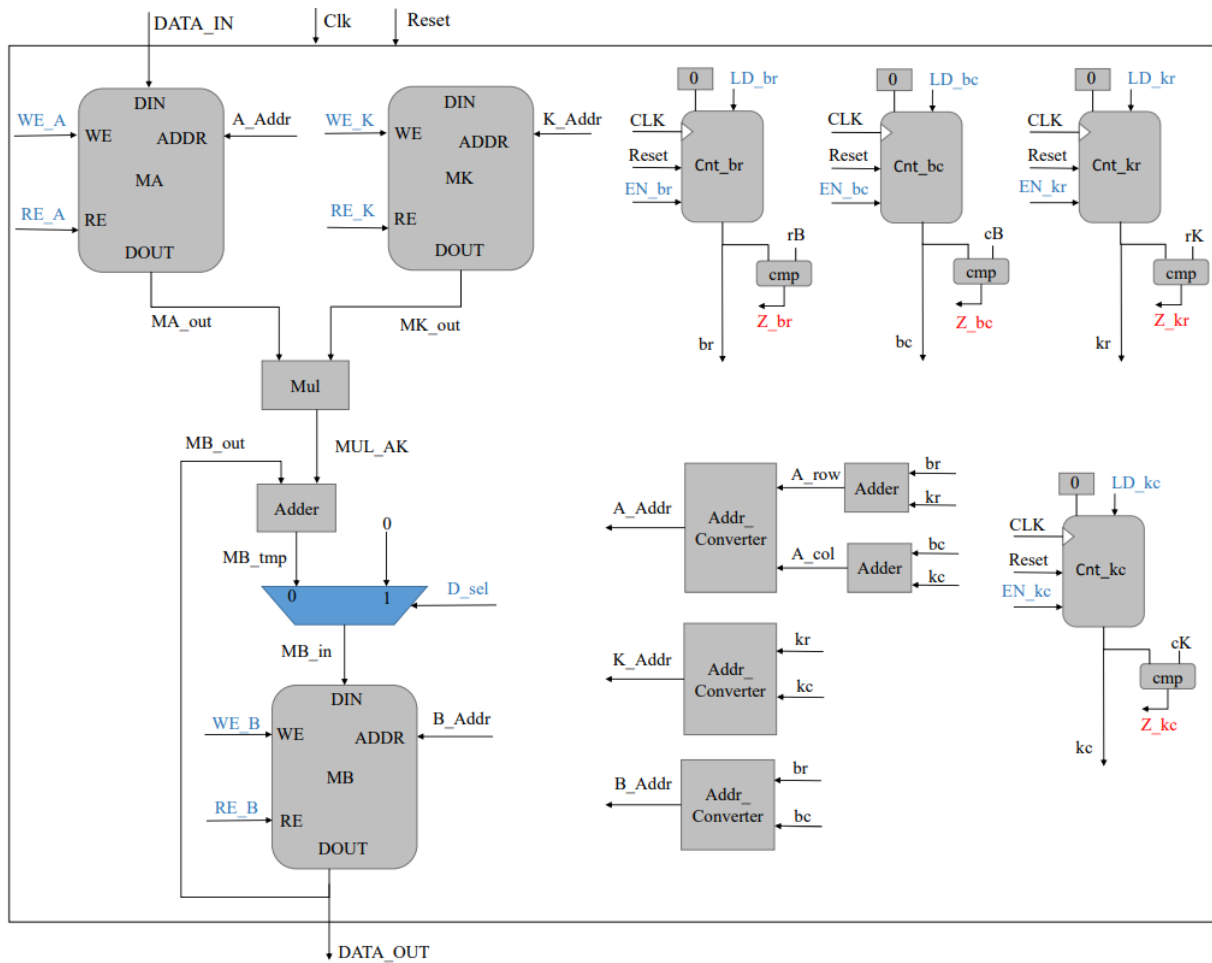
#### 4.2. Đơn vị xử lý dữ liệu (Datapath)

Bước tiếp theo của chúng ta là phân chia chức năng máy FSMD thành các thành phần cấu trúc gồm đơn vị xử lý dữ liệu datapath và bộ điều khiển. Phần datapath phải bao gồm một sự kết nối của các mạch tổ hợp và tuần tự. Phần điều khiển phải bao gồm một FSM thuần túy (tức là chỉ chứa các phép và điều kiện trên biến logic).

Datapath có các phần tử dpmemory (A\_mem, B\_mem, K\_mem) để lưu trữ các phần tử của ma trận A, B, K. Phần tử dpmemory được chia thành 3 phần tử khác nhau, về cơ bản cấu trúc chức năng đều như nhau nhưng dữ liệu lưu trữ bên trong khác nhau.

Các bộ counter để đếm các biến br, bc, kr, kc các bộ xử lý tính toán bộ nhân và bộ cộng, bộ so sánh các biến đếm.

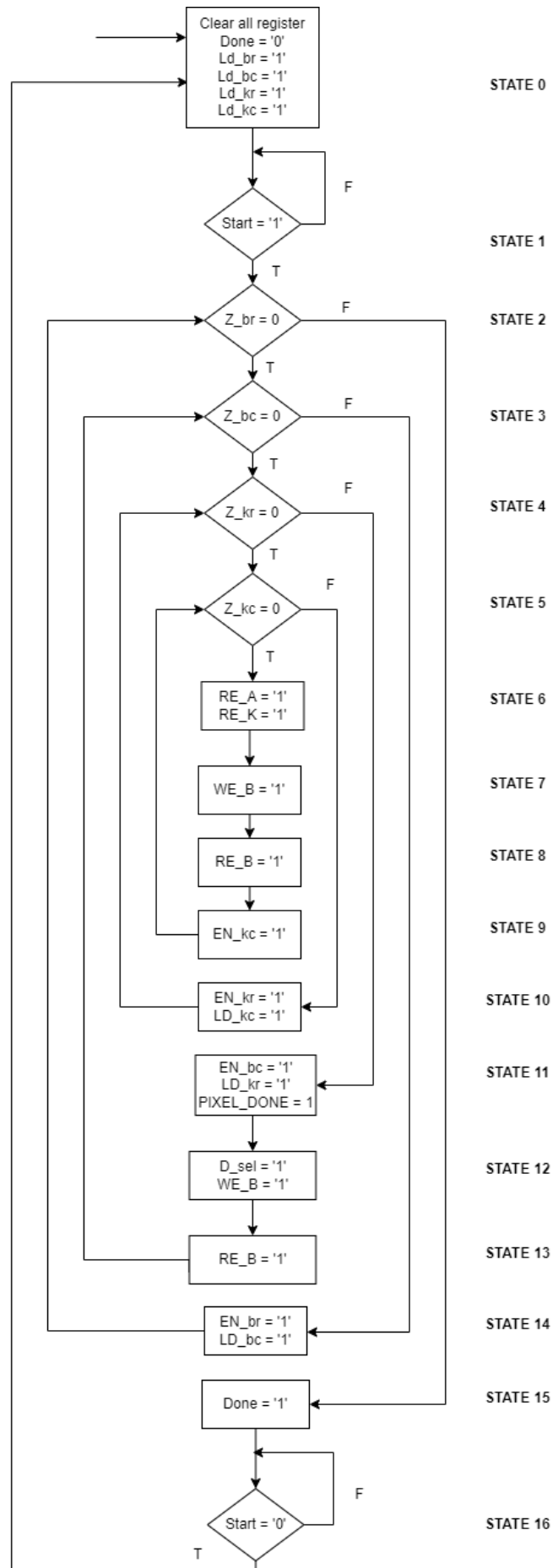
Bộ chuyển đổi (Addr\_Converter) chuyển đổi từ địa chỉ ma trận 2 chiều sang địa chỉ ma trận 1 chiều.



*Hình 4: Cấu trúc của đơn vị xử lý dữ liệu Datapath.*

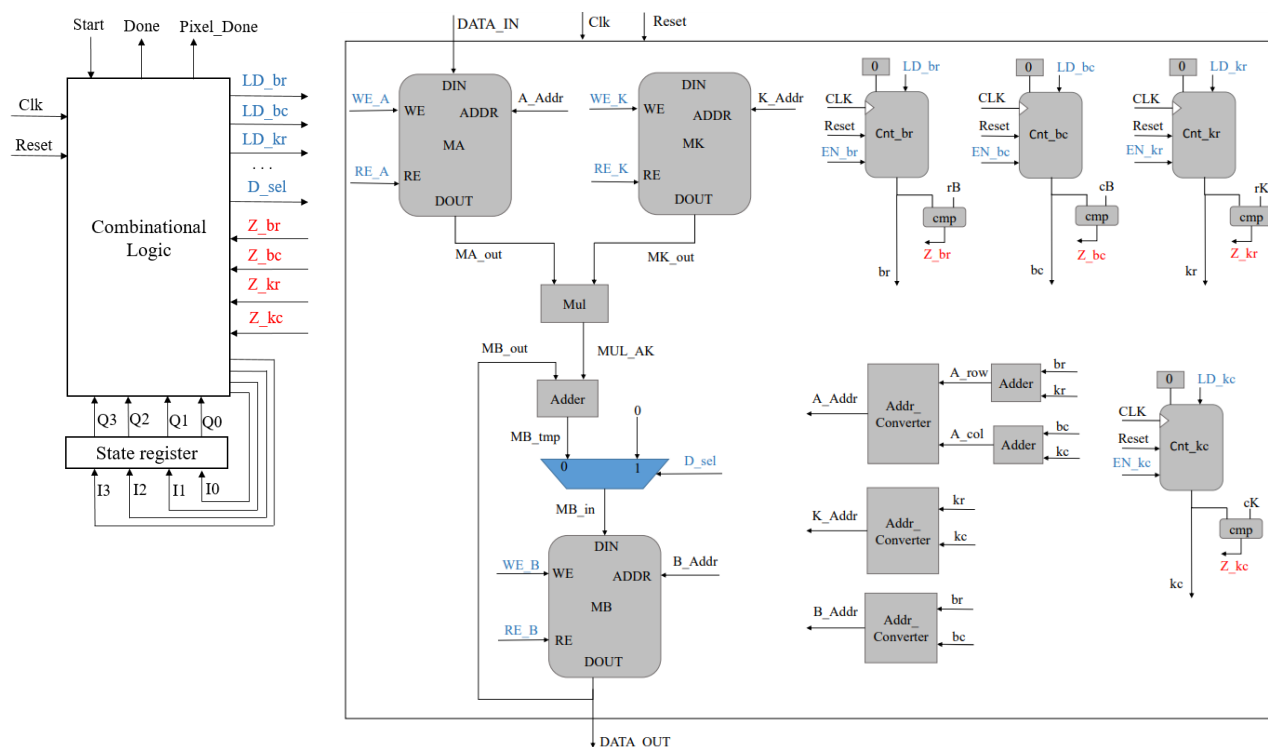
#### 4.3. Đơn vị điều khiển (Control Unit)

Sau khi thiết kế xong datapath chúng ta tiến hành chuyển đổi máy trạng thái FSMD trong thành máy trạng thái FSM mô tả hoạt động của bộ điều khiển. Máy trạng thái FSM có các trạng thái và chuyển tiếp giống như máy FSMD. Tuy nhiên, trong FSM chúng ta đã thay thế các phép tính và điều kiện phức tạp bằng các phép tính và điều kiện trên các biến logic và tạo ra các tín hiệu điều khiển hoạt động của datapath.



Hình 5: Máy FSM của đơn vị điều khiển.

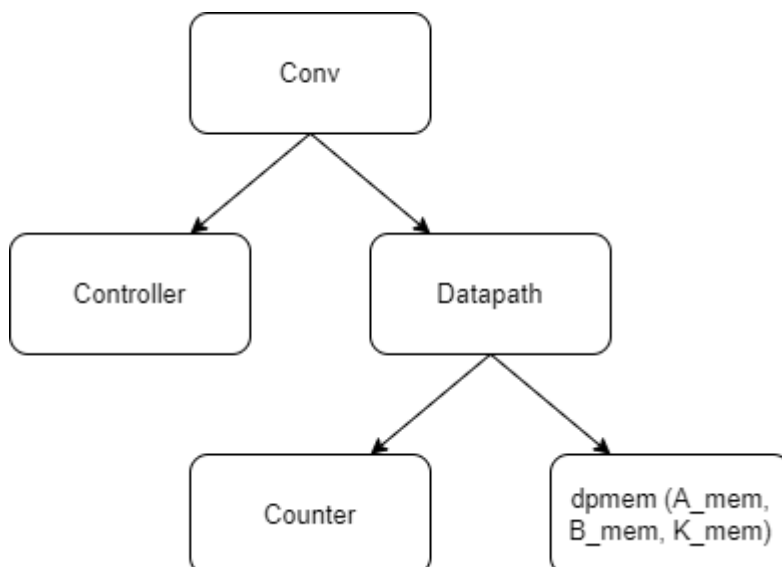
#### 4.4. Sơ đồ khối tổng thể



Hình 6: Sơ đồ khối tổng thể.

## 5. Mô hình hóa bằng VHDL

Tổ chức top-down của các tệp mã nguồn



*Hình 7: Tổ chức của file VHDL.*

## 6. Mô phỏng/thực thi và đánh giá

Kết quả mô phỏng với ma trận đầu vào kích thước 5x5 và kernel kích thước 3x3.



## 7. Kết luận

## Appendix A: VHDL Code

*(Trong file .zip đính kèm)*

## **List of Figures**

Hình 1. Ví dụ minh họa phép tính tích chập 2D. ....	6
Hình 2. Giao diện ghép nối I/O. ....	7
Hình 3: Mô hình máy FSMD. ....	10
Hình 4: Cấu trúc của đơn vị xử lý dữ liệu Datapath. ....	11
Hình 5: Máy FSM của đơn vị điều khiển. ....	13
Hình 6: Sơ đồ khối tổng thể. ....	13
Hình 7: Tổ chức của file VHDL.....	14

## **List of Tables**

Bảng 1. Định nghĩa giao diện vào ra. ....	8
---	---



## References

- [1] <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>
- [2] <https://dayhocstem.com/blog/2021/02/ma-nguon-vhdl-muc-rtl-mach-nhan-2-ma-tran.html>