

REPORT

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

Ver 1.0

19/4/2022

	Full name	Function	Date
Written by	Phạm Thế Bảo Nguyễn Việt Anh		14/06/2022
Verified by			
Approved by			

Abstract (from 5 to 10 lines)
Xây dựng thuật toán từ ví dụ đã cho Thiết kế mức RTL được đề xuất, mô hình VHDL, mô phỏng ModelSIM và

Keywords
FSM, FSMD, ModelSim, VHDL, RTL

Work context
<ol style="list-style-type: none">1. Tìm hiểu kiến thức cơ bản về VHDL, ModelSim, mô phỏng verify2. Phân tích yêu cầu vấn đề (Tìm hiểu, phân tích thuật toán tích chập)3. Bắt đầu viết từng các khối cấu thành4. Kết hợp các khối thành phần thành khối hoàn chỉnh5. Viết và chạy testbench và kiểm tra kết quả đầu ra6. Hoàn thiện báo cáo dựa trên kết quả đầu ra

Document History

Version	Time	Revised by	Description
V0.1	19/04/2022	Nguyễn Kiên Hùng	Original Version
V0.2	14/06/2022	Phạm Thế Bảo Nguyễn Việt Anh	1. Version

MỤC LỤC

Document History	3
Table of Contents	Error! Bookmark not defined.
1. Giới thiệu.....	5
2. Yêu cầu	6
2.1. <i>Yêu cầu đối với thiết kế:</i>	6
2.2. <i>Định nghĩa giao diện vào/ra</i>	7
3. Thuật toán.....	8
4. Thiết kế mức RTL	9
4.1. <i>Mô hình máy FSM</i>	9
4.2. <i>Đơn vị xử lý dữ liệu (Datapath)</i>	11
4.3. <i>Đơn vị điều khiển (Control Unit)</i>	11
4.4. <i>Sơ đồ khối tổng thể</i>	13
5. Mô hình hóa bằng VHDL	14
6. Mô phỏng và đánh giá.....	14
7. Kết luận	Error! Bookmark not defined.
Appendix A: Schematic	16
Appendix B: VHDL Code.....	17
Appendix C:	18
List of Figures	19
List of Tables.....	20
References	21

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 5×5 như sau:

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

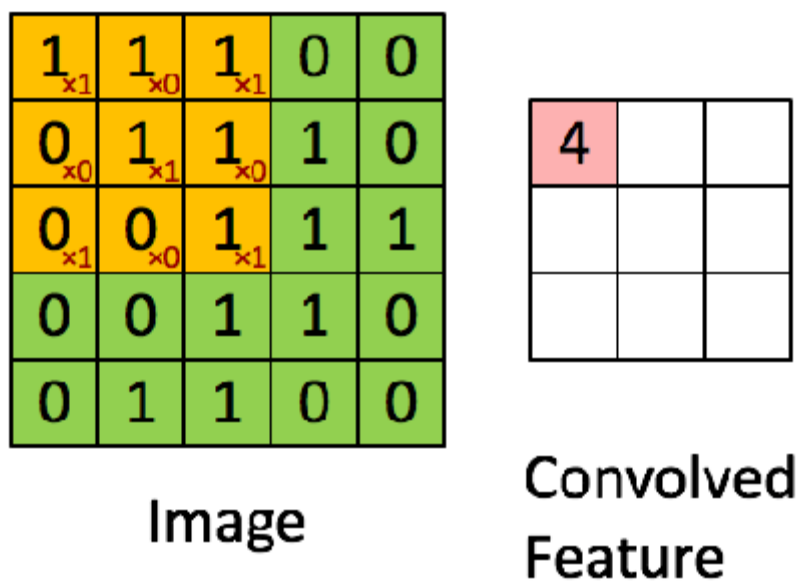
Và kernel có kích thước 3×3 như sau:

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

thì kết quả tính toán của khối 2D-Convolution trả về hình ảnh J có kích thước 3×3 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.



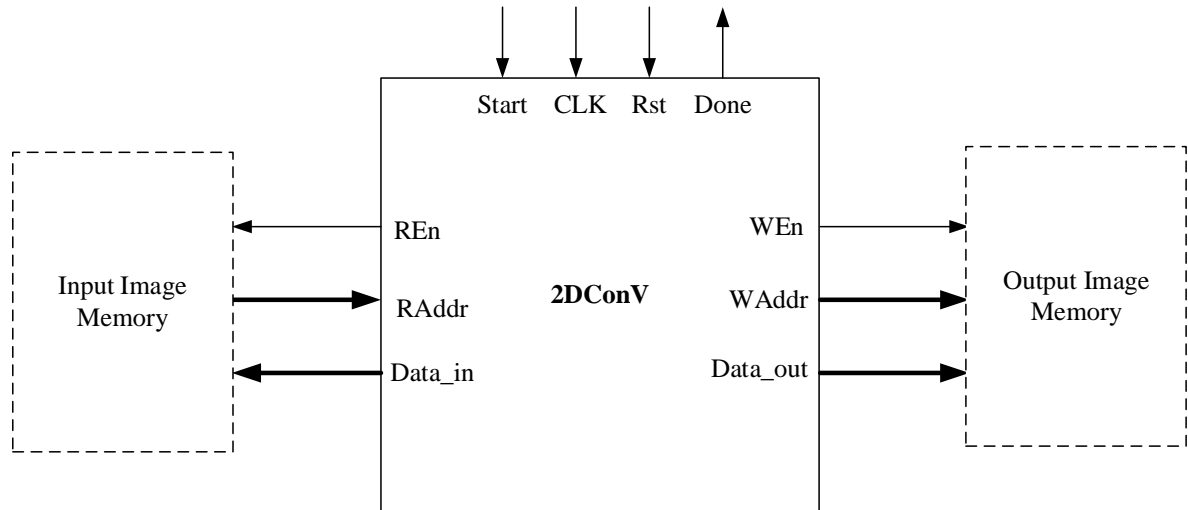
Hình 1. Ví dụ minh họa phép tính tích chập 2D.

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

2. Yêu cầu

2.1. Yêu cầu đối với thiết kế:

- Khối 2DConV có giao diện ghép nối tới CPU sao cho CPU kích hoạt quá trình tính toán của khối 2DConV bằng các đặt tín hiệu Start = ‘1’.
- Sau khi quá trình tính hình ảnh tích phân hoàn thành, khối 2DConV sẽ báo cho CPU biết bằng cách đặt tín hiệu Done = ‘1’;
- Khối 2DConV có 1 giao diện ghép nối tới bộ nhớ để đọc hình ảnh đầu vào
- Khối 2DConV 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

Bảng 1: Mô tả các tín hiệu vào ra.

TT	Port	Direction	Width	Meaning
1	Start	IN	1	Tín hiệu bắt đầu quá trình tính toán
2	CLK	IN	1	Tín hiệu xung clock hệ thống
3	Rst	IN	1	Xung reset hệ thống
4	RAddr	IN	N	Địa chỉ bộ nhớ Ma trận sẵn
5	REn	OUT	1	Tín hiệu đọc dữ liệu từ bộ nhớ
6	WEn	OUT	1	Tín hiệu ghi dữ liệu vào bộ nhớ
7	WAddr	OUT	N	Địa chỉ bộ nhớ ghi ma trận
8	Data_in	IN	N	Dữ liệu ma trận lấy từ bộ nhớ
9	Data_out	OUT	N	Dữ liệu ma trận ghi vào bộ nhớ

3. Thuật toán

Do thuật toán nhân ma trận 2 chiều sẽ làm phức tạp trong quá trình thiết kế mạch RTL. Vì vậy để tốt ưu, thuật toán nhân ma trận sẽ được xử lý dạng mảng 1 chiều. Dưới đây là lưu đồ thuật toán :

Mã giả (Pseudo code) mô tả thuật toán.

[initialize input data]

Beginning: wait for start = '1'

Done ='0'

For m=0 to rowA - rowK do

For n= 0 to colA - rowK do

For i=0 to rowK do

For j = 0 to rowK do

$C[m*(rowA-rowK) + n] += A[(i+m)*rowA + (j+n)] * B[i*rowK + j]$

End for

End for

End for

End for

Done='1'

[write output data]

Wait for start ='0'

Go to beginning

End.

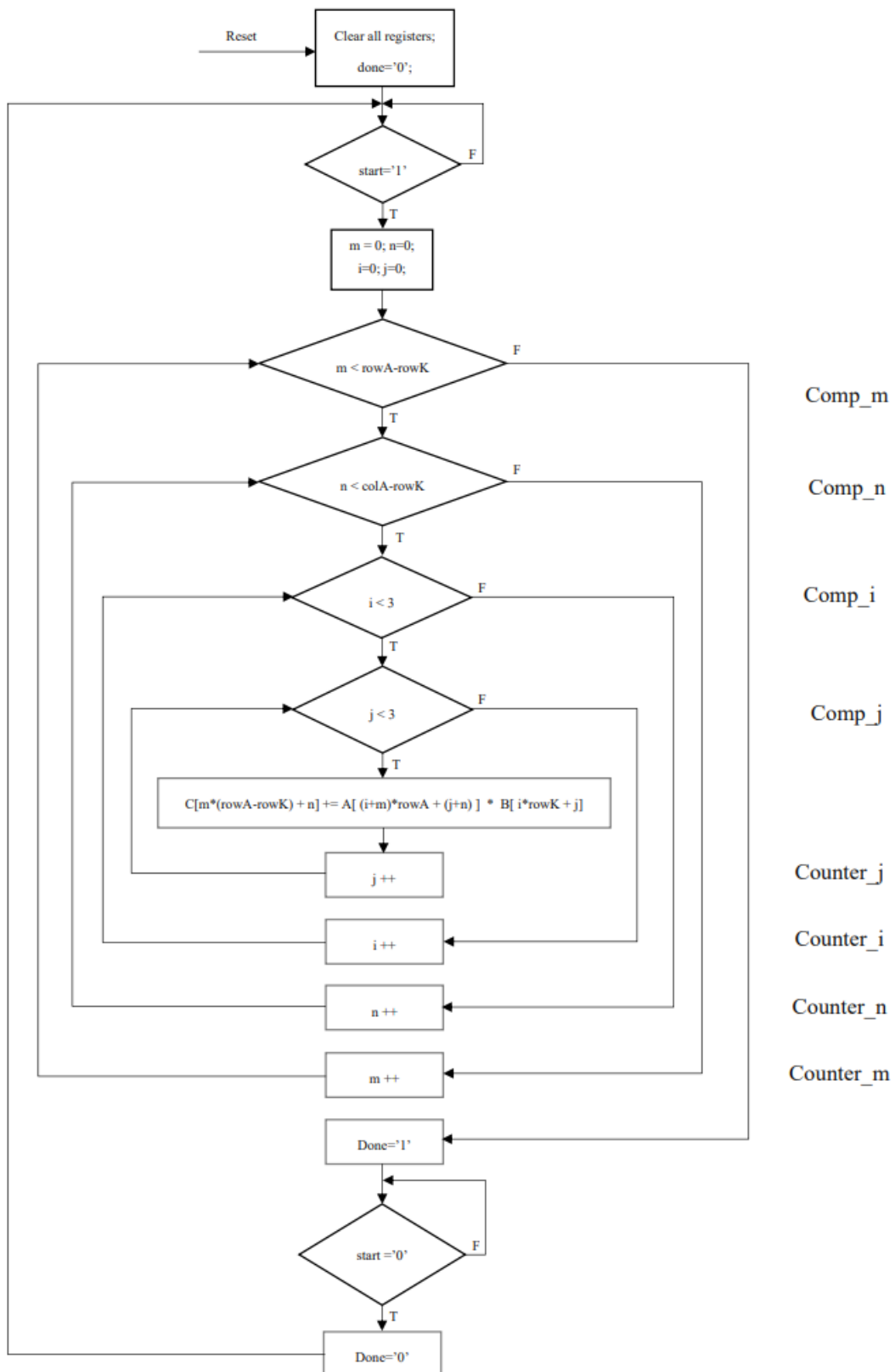
Các phần tử của ma trận A và B sẽ được trải trên mảng 1 chiều A và B, từ đó cùng 4 vòng lặp for với 2 vòng đầu mục đích để dịch chuyển ma trận kernel trên ma trận A ,còn 2 vòng sau là để nhân tích chập của ma trận con trong A với ma trận Kernel.

4. Thiết kế mức RTL

Gợi ý : Tham khảo Lecture 3

4.1. Mô hình máy FSMD

Để bắt đầu xây dựng mạch điện thực hiện chức năng nhân ma trận, trước tiên chúng ta chuyển đổi chương trình thành sơ đồ máy trạng thái phức tạp FSMD như Hình 2. Như tên gọi của nó các trạng thái của FSMD có thể bao gồm các biểu thức số học của tổ hợp các đầu vào và đầu ra bên ngoài hoặc biến. FSMD tương phản với sơ đồ trạng thái FSM thuần túy trong đó chỉ bao gồm các biểu thức boolean của chỉ các đầu vào và đầu ra bên ngoài, không có các biến. Do đó, các sơ đồ trạng thái phức tạp này trông giống như một chương trình tuần tự, trong đó các câu lệnh đã được lập lịch để thực hiện trong các trạng thái. Sơ đồ trạng thái như dưới đây:



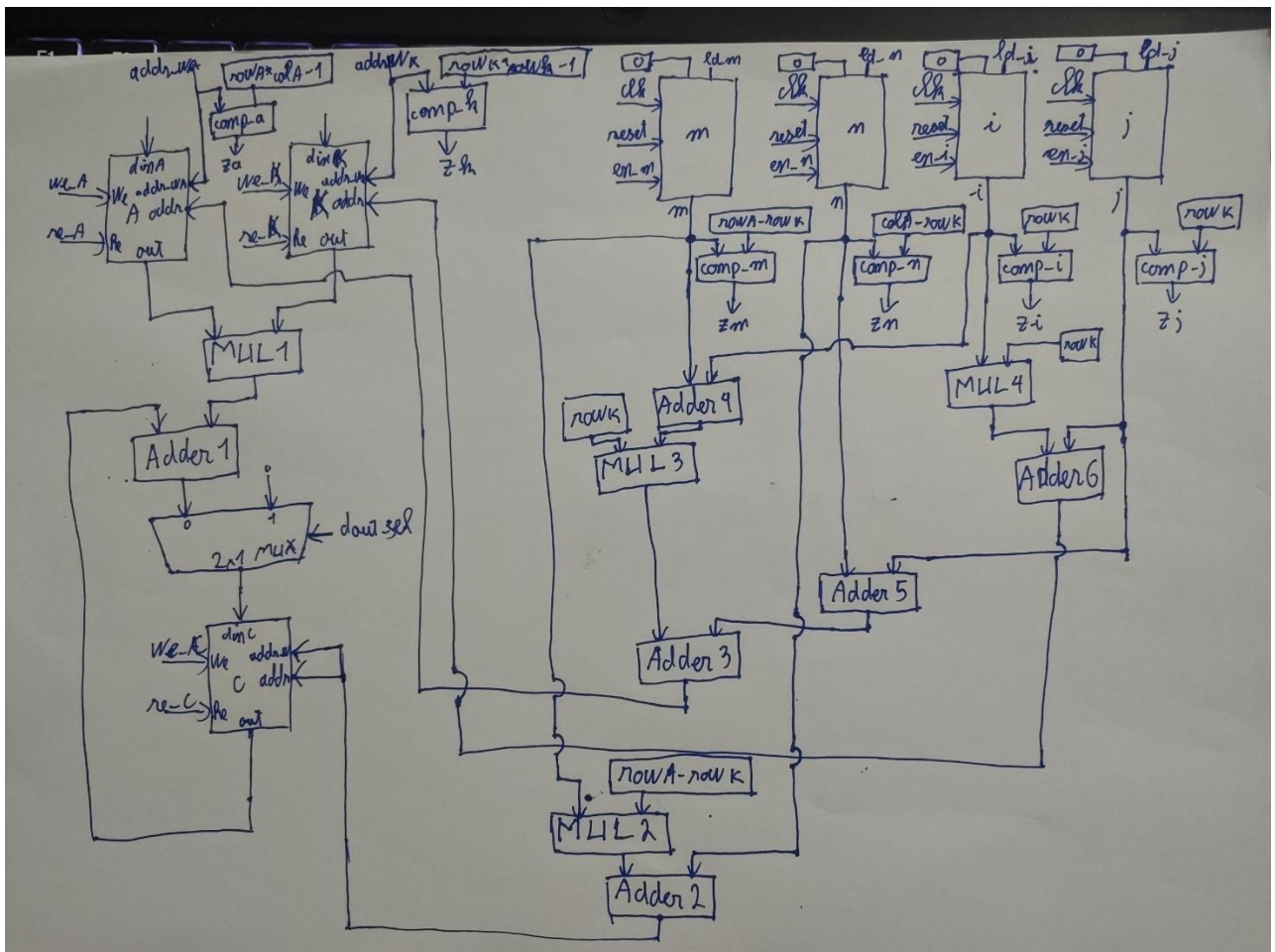
Hình 3: Mô hình máy FSMD.

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 như trong Hình 3. 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).

Ở đây sử dụng các bộ nhớ dpmemory để lưu trữ các phần tử của ma trận A,K,C, các bộ counter để đếm các biến m,n,i,j các bộ xử lý tính toán bộ nhân,cộng so sánh

Sinh viên/Học viên chỉ ra sơ đồ cấu trúc của datapath của thiết kế ở đây!



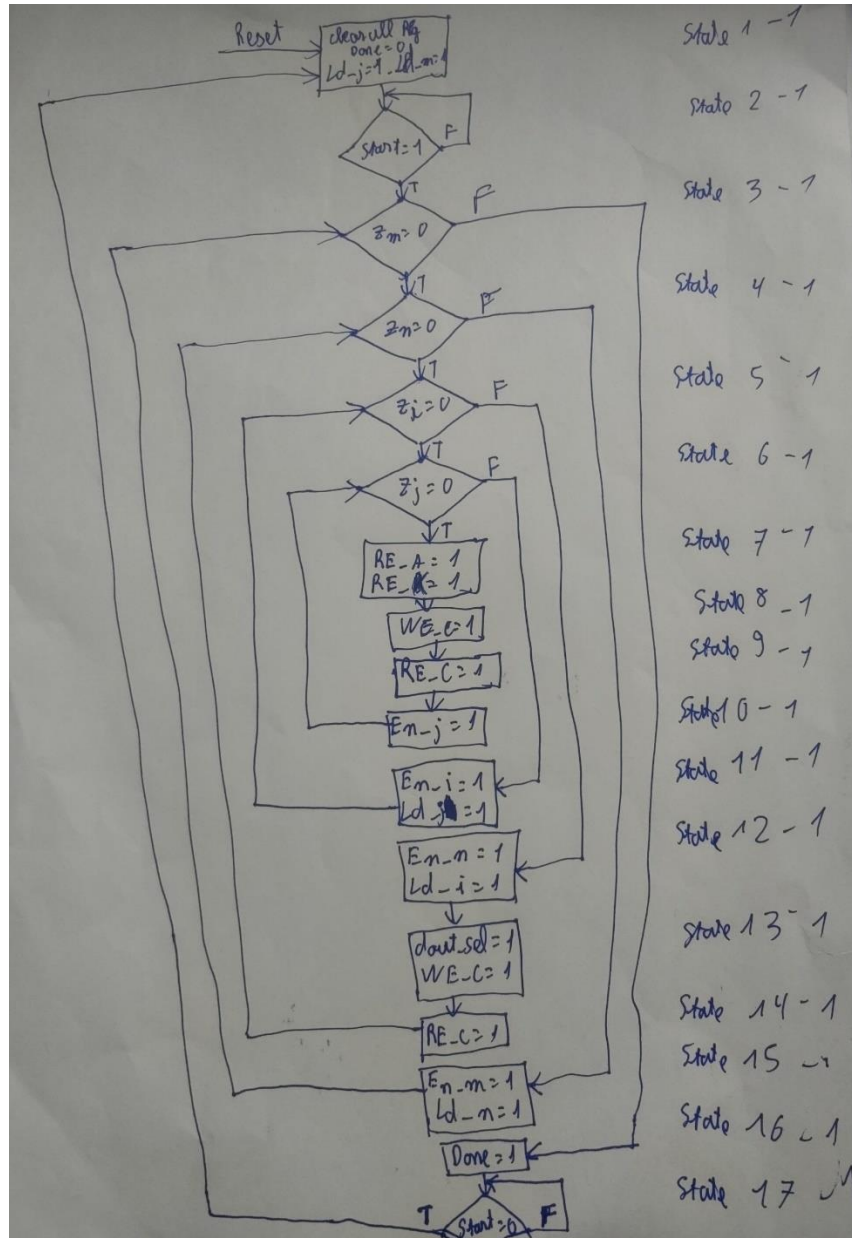
Hình 4: Cấu trúc của đơn vị xử lý dữ liệu Datapath.

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 Hình 2 thành máy trạng thái FSM trong Hình 5 mô tả hoạt động của bộ điều khiển. Máy

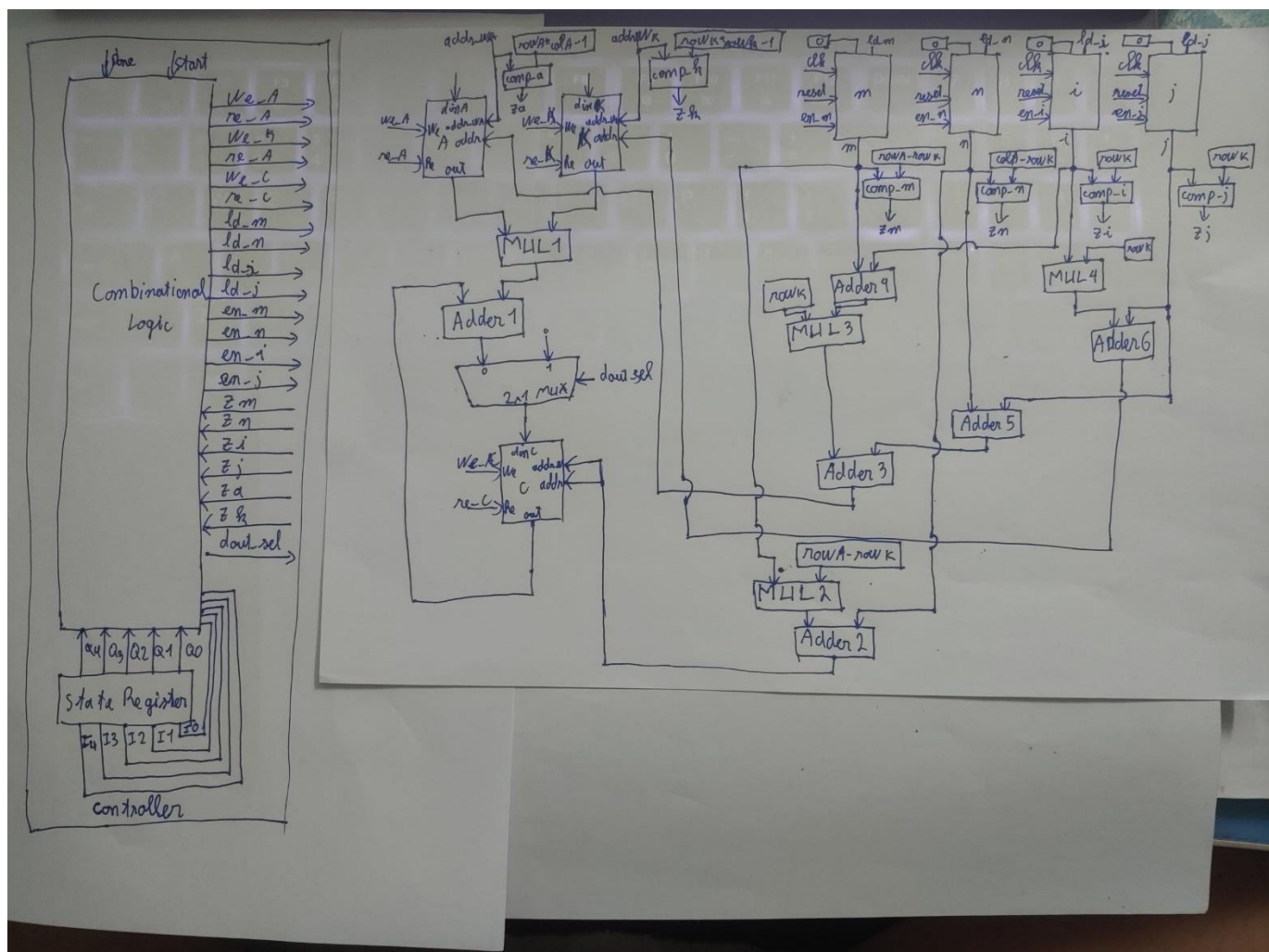
trạng thái FSM có các trạng thái và chuyển tiếp giống như máy FSM. Tuy nhiên, trong FSM chúng ta đã thay thế các phép tính và điều kiện phức tạp bằng các phép tính và điều kiện trên các biến logic và tạo ra các tín hiệu điều khiển hoạt động của datapath. Giao diện các tín hiệu vào/ra của bộ điều khiển được mô tả trong Hình 6

Sinh viên/Học viên chỉ ra sơ đồ máy trạng thái FSM của bộ điều khiển controller của thiết kế ở đây!



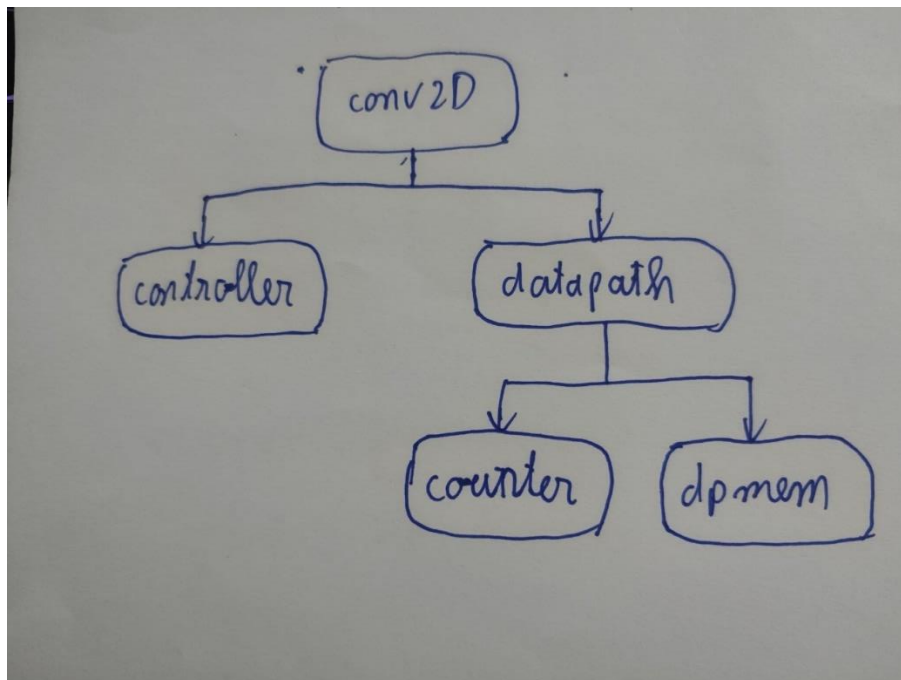
Hình 5: Máy FSM của đơn vị điều khiển.

4.4. Sơ đồ khối tổng thể



Hình 6: Sơ đồ khối tổng thể của thiết kế.

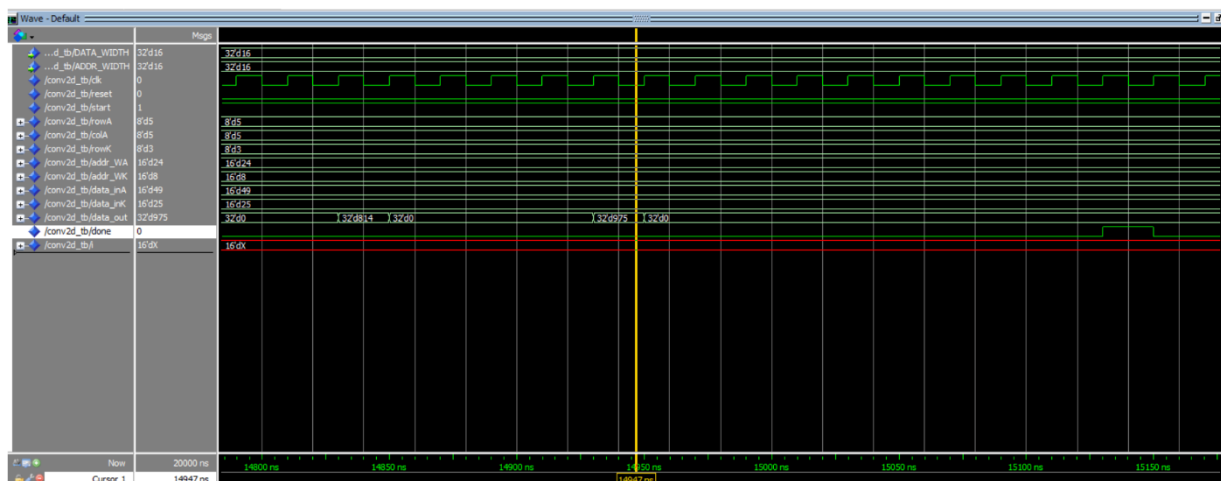
5. Mô hình hóa bằng VHDL



Hình 6: Tổ chức của các tệp mã nguồn.

6. Mô phỏng/thực thi và đánh giá

Kết quả:



Ma trận ảnh A:

00000000	0001	0003	0005	0007	0009
00000005	000B	000D	000F	0011	0013
0000000a	0015	0017	0019	001B	001D
0000000f	001F	0021	0023	0025	0027
00000014	0029	002B	002D	002F	0031

Ma trận kernel K:

00000000	0001	0004	0007
00000003	000A	000D	0010
00000006	0013	0016	0019

Ma trận kết quả:

00000000	000001A9	000001DB	0000023F
00000003	00000271	000002D5	00000307
00000006	0000036B	0000039D	000003CF

7. Mã nguồn

Chúng em up code lên drive thầy xem ạ:

<https://drive.google.com/file/d/1vZC7lcHoKn8gbURZH2Rz2R6Tasllsine/view?usp=sharing>

Appendix A:

Appendix B: VHDL Code

(đóng gói thành tệp nén và gửi kèm báo cáo)

Appendix C:

Compress and email to hungnvnu@gmail.com

List of Figures

Hình 1. Ví dụ minh họa quá trình sắp xếp một dãy gồm 4 phần tử theo trật tự tăng dần.	6
Hình 2. Giao diện ghép nối I/O của đơn vị Sorting Unit.	7
Hình 3: FSMD.	10
Hình 4: Datapath.	11
Hình 5: FSM of controller.	12
Hình 6: Block diagram of whole sorting unit.	13

List of Tables

Bảng 1: Mô tả các tín hiệu vào ra.....	7
--	---

References

- [1] <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>