**REPORT**

**Thiết kế và thực hiện mạch tính Pooling 2-D dùng cho mạng nơ-ron nhân chập CNN**

Ver 1.0

20/10/2022

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Full name** | **Function** | **Date** |
| Written by |  |  |  |
| Verified by |  |  |  |
| Approved by |  |  |  |

|  |
| --- |
| **Abstract (from 5 to 10 lines)** |
| Bản báo cáo này thiết kế mô-đun phần cứng mạch tích hợp chuyên dụng. Đề xuất thiết kế mức RTL, mô hình hóa bằng VHDL, mô phỏng bằng ModelSim (QuestaSim) và thực thi trên FPGA một mô-đun phần cứng thực hiện tính phép tính average pooling. Ứng dụng phần cứng sử dụng trong mạng nơ-ron nhân chập CNN. |

|  |
| --- |
| **Keywords** |
| FSM, FSMD, datapath, controller, average pooling, mạng nơ-ron nhân chập CNN, VHDL |

|  |
| --- |
| **Work context** |
|  |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Time** | **Revised by** | **Description** |
| V0.1 | 20/10/2022 | Nguyễn Kiêm Hùng | Problem definition |
| V0.2 |  |  |  |
|  |  |  |  |

MỤC LỤC

[Document History 3](#_Toc117778198)

[MỤC LỤC 4](#_Toc117778199)

[1. Giới thiệu 5](#_Toc117778200)

[2. Yêu cầu 5](#_Toc117778201)

[2.1. Yêu cầu đối với thiết kế: 5](#_Toc117778202)

[2.2. Định nghĩa giao diện vào/ra 6](#_Toc117778203)

[3. Thuật toán 7](#_Toc117778204)

[4. Thiết kế mức RTL 8](#_Toc117778205)

[4.1. Mô hình máy FSMD 8](#_Toc117778206)

[4.2. Đơn vị xử lý dữ liệu (Datapath) 9](#_Toc117778207)

[4.3. Đơn vị điều khiển (Control Unit) 10](#_Toc117778208)

[4.4. Sơ đồ khối tổng thể 11](#_Toc117778209)

[5. Mô hình hóa bằng VHDL 11](#_Toc117778210)

[6. Mô phỏng/thực thi và đánh giá 12](#_Toc117778211)

[7. Kết luận 15](#_Toc117778212)

[Appendix A: 16](#_Toc117778213)

[Appendix B: VHDL Code 17](#_Toc117778214)

[(đóng gói thành tệp nén và gửi kèm báo cáo) 17](#_Toc117778215)

[Appendix C: 18](#_Toc117778216)

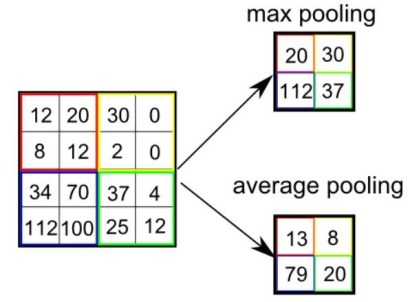
[List of Figures 19](#_Toc117778217)

[List of Tables 20](#_Toc117778218)

[References 21](#_Toc117778219)

# Giới thiệu

**Mục tiêu:** Vận dụng các kiến thức, kỹ năng đã được học đểthiết kế, mô phỏng và thực thi một mô-đun phần cứng thực hiện tính Pooling J = AvgPool2D(I, size, stride) cho hình ảnh lối vào I với kích thước pooling size = (2,2), và bước nhảy stride = 2, padding = 0. Trong đó, mỗi pixel trong hình ảnh đầu ra J đại diện cho giá trị trung bình (average) của một ma trận có kích thước size được trích xuất từ ma trận đầu vào I. Phép tính average pooling được minh họa bằng ví dụ trong Hình 1. Trong đó hình ảnh lối vào I có kích thước 4×4, size = (2,2), stride = 2, padding = 0.

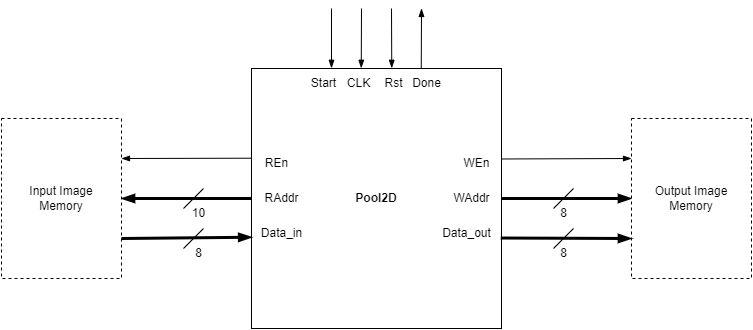


Hình 1. Ví dụ phép tính pooling 2D.

# Yêu cầu

## Yêu cầu đối với thiết kế:

* Hình ảnh đầu vào I có kích thước 32×32, size = (2,2), stride = 2 (padding = 0).
* Khối Pool2D có giao diện ghép nối tới CPU sao cho CPU kích hoạt quá trình tính toán của khối Pool2D bằng các đặt tín hiệu Start = ‘1’.
* Sau khi quá trình tính hình ảnh tích phân hoàn thành, khối Pool2D sẽ báo cho CPU biết bằng cách đặt tín hiệu Done = ‘1’;
* Khối Pool2D có 1 giao diện ghép nối tới tới bộ nhớ để đọc hình ảnh đầu vào
* Khối Pool2D có 1 giao diện ghép nối tới tới bộ nhớ để ghi dữ liệu đầu ra



Hình 2. Giao diện ghép nối I/O.

## Định nghĩa giao diện vào/ra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TT** | **Port** | **Direction** | **Width** | **Meaning** |
| 1 | Start | In | 1 | Lựa chọn chế độ hoạt động:  - 0 : khởi tạo  - 1 : thực hiện pooling |
| 2 | CLK | In | 1 | Tín hiệu xung nhịp Clock |
| 3 | Rst | In | 1 | Tín hiệu Reset tích cực mức cao |
| 4 | Done | Out | 1 | Tín hiệu xác nhận hoàn thành quá trình pooling |
| 5 | Data\_in | In | 8 | Bus dữ liệu đầu vào |
| 6 | RAddr | Out | 10 | Địa chỉ đọc dữ liệu |
| 7 | REn | Out | 1 | Tín hiệu cho phép đọc |
| 8 | Data\_out | Out | 8 | Bus dữ liệu đầu ra |
| 9 | WAddr | Out | 8 | Địa chỉ ghi dữ liệu |
| 10 | WEn | Out | 1 | Tín hiệu cho phép ghi |

Bảng 1: Mô tả các tín hiệu vào ra.

# Thuật toán

**Thuật toán cho AvgPool2D**

0: int i, j, k, sum, RAddr\_root, offset[4] = {0,1,32,33};

1: if (Rst = ‘1’) {Done := ‘0’;}

else {

2: while (!Start);

3: i := 0;

4: while (i < 16) {

5: j := 0;

6: while (j < 16) {

7: sum := 0;

8: k := 0;

9: while (k < 4) {

10: RAddr := RAddr\_root + offset[k];

11: data\_in := Mem\_in[RAddr];

12: sum := sum + data\_in;

13: k++;

}

14: Mem\_out[WAddr] := sum / 4;

15: RAddr\_root := RAddr\_root + 2;

16: WAddr++;

17: j++;

}

18: RAddr\_root := RAddr\_root + 32;

19: i++;

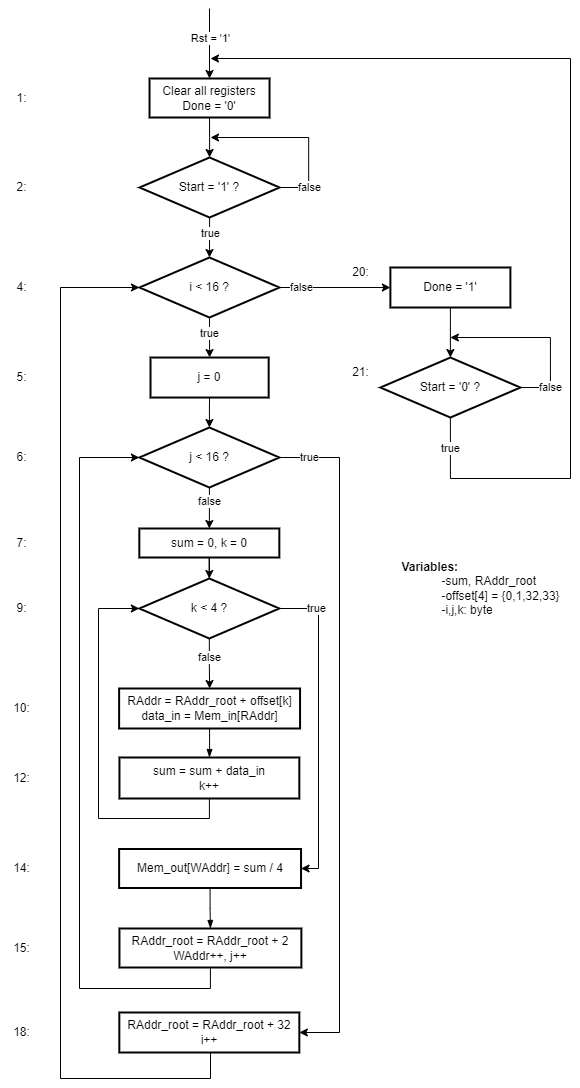
}

20: Done\_o = '1';

}

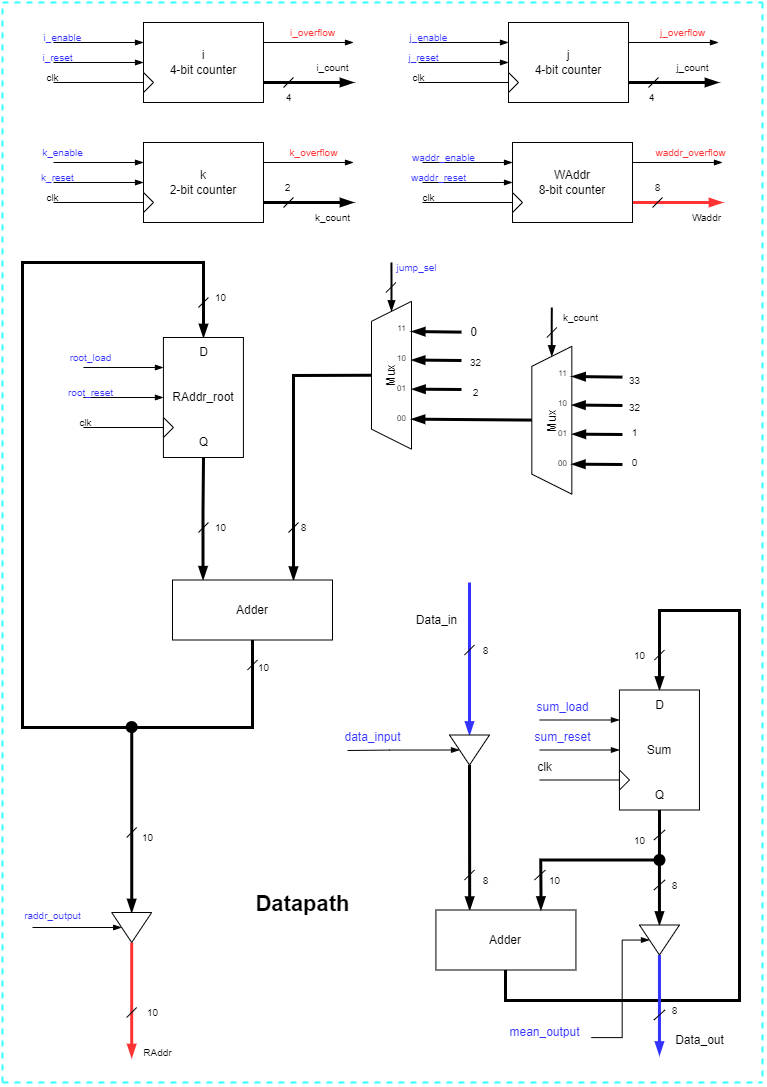
# Thiết kế mức RTL

## Mô hình máy FSMD



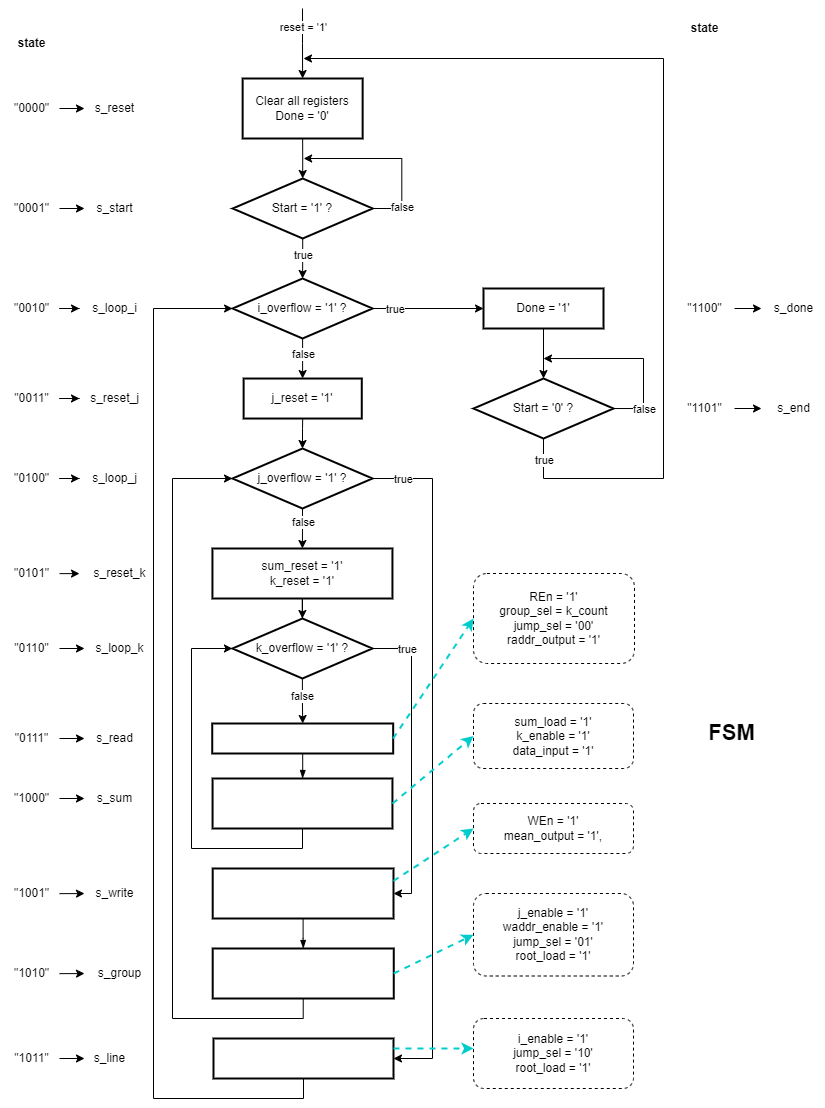
Hình 3: Mô hình máy FSMD.

## Đơn vị xử lý dữ liệu (Datapath)



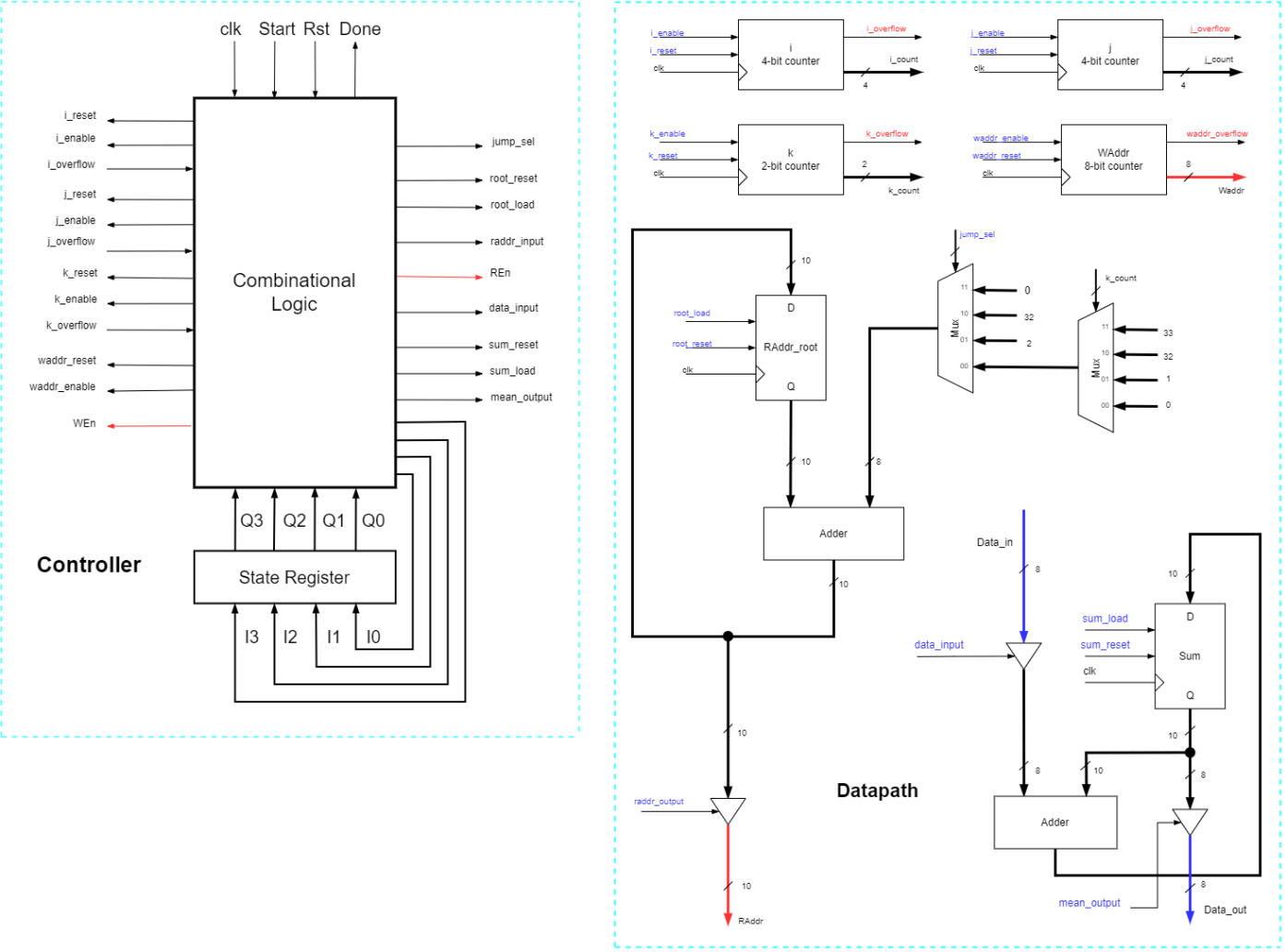
Hình 4: Cấu trúc của đơn vị xử lý dữ liệu Datapath.

## Đơn vị điều khiển (Control Unit)



Hình 5: Máy FSM của đơn vị điều khiển.

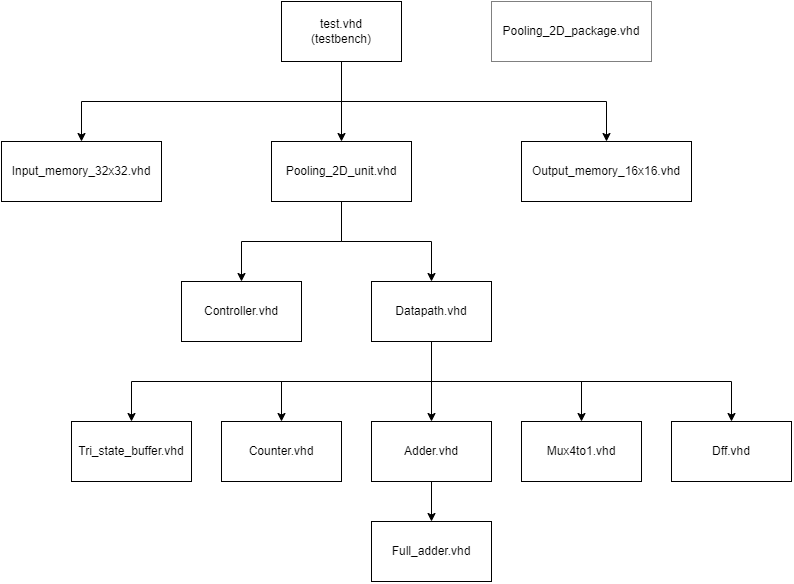
## Sơ đồ khối tổng thể



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

# Mô hình hóa bằng VHDL

**Mã nguồn VHDL:** Appendix B

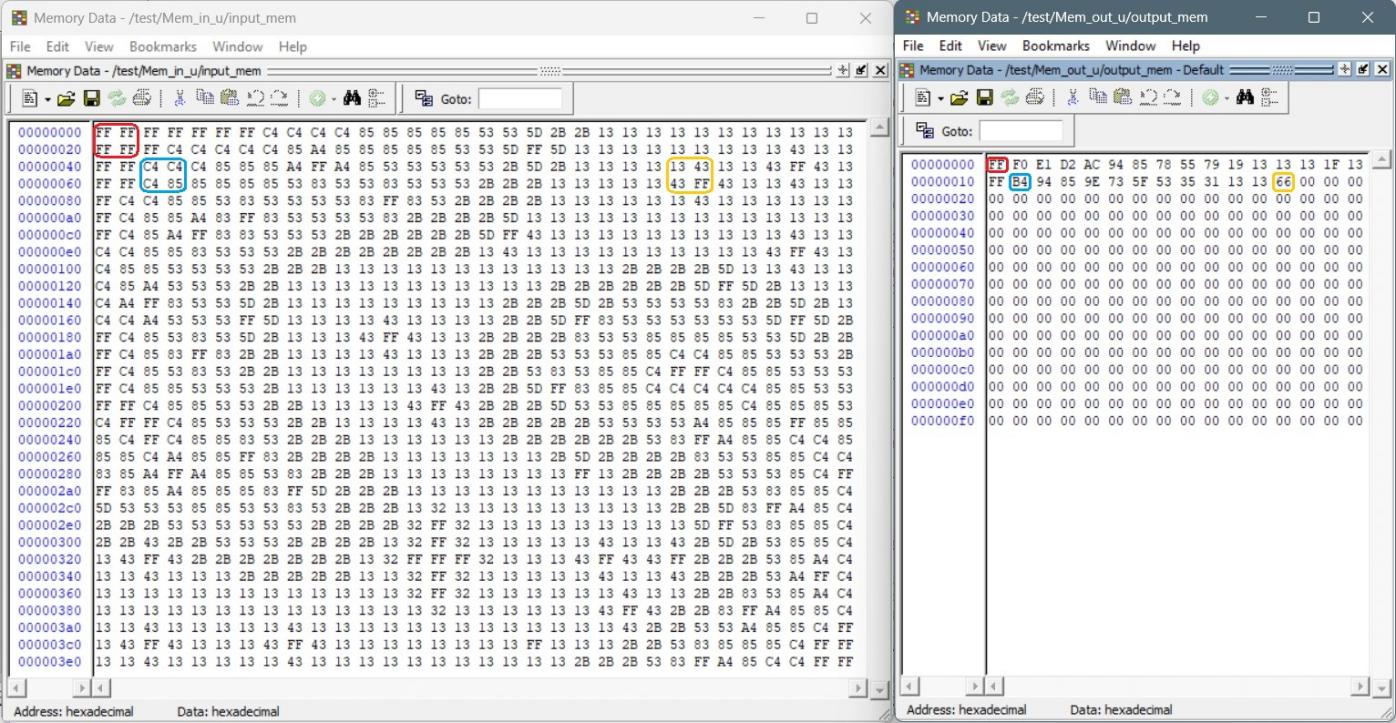


Hình 7:Tổ chức dạng cây của các file mã nguồn.

# Mô phỏng/thực thi và đánh giá

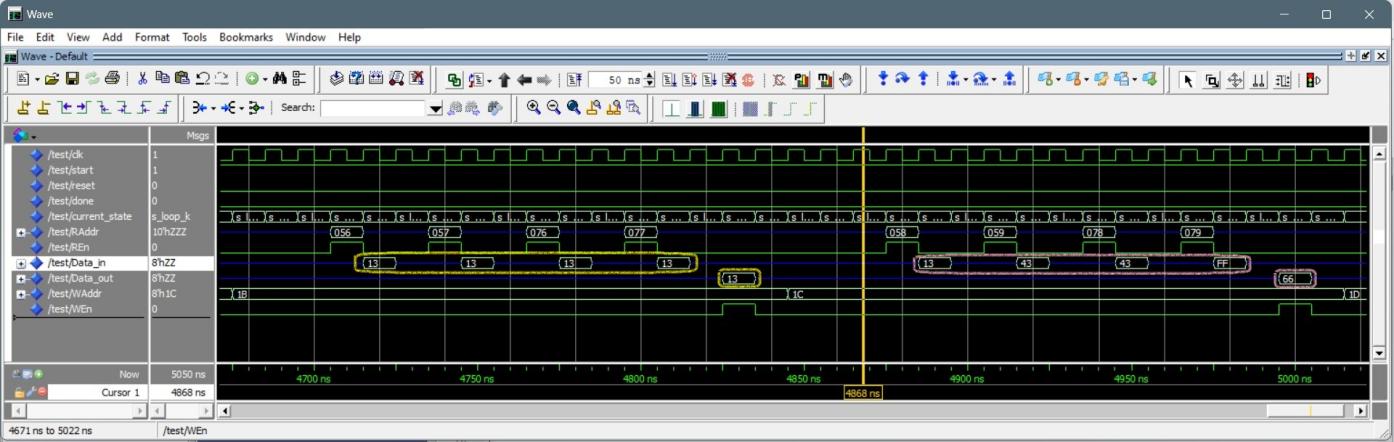
Mô phỏng thiết kế được thể hiện qua các hình phía dưới tại các thời điểm 5020ns (Hình 8) và 44210ns (Hình 10) (chu kỳ xung nhịp 10ns). Ta lấy một bức ảnh grayscale kích thước 32x32 (Hình 11) làm mẫu thử, thực hiện quá trình Average Pooling 2D, kết quả trả về là một ảnh 16x16 (Hình 12). Các phần được khoanh cùng màu thể hiện quá trình average pooling cho từng điểm ảnh.

Kết quả mô phỏng có khác đôi chút so với kết quả khi chạy trên phần mềm. Một vài điểm ảnh sai lệch 1 đơn vị do kết quả của phần mềm được làm tròn còn trên mô phỏng phần cứng thì không. Nhưng sự sai lệch này không gây ảnh hưởng rõ rệt đến kết quả ảnh đầu ra.

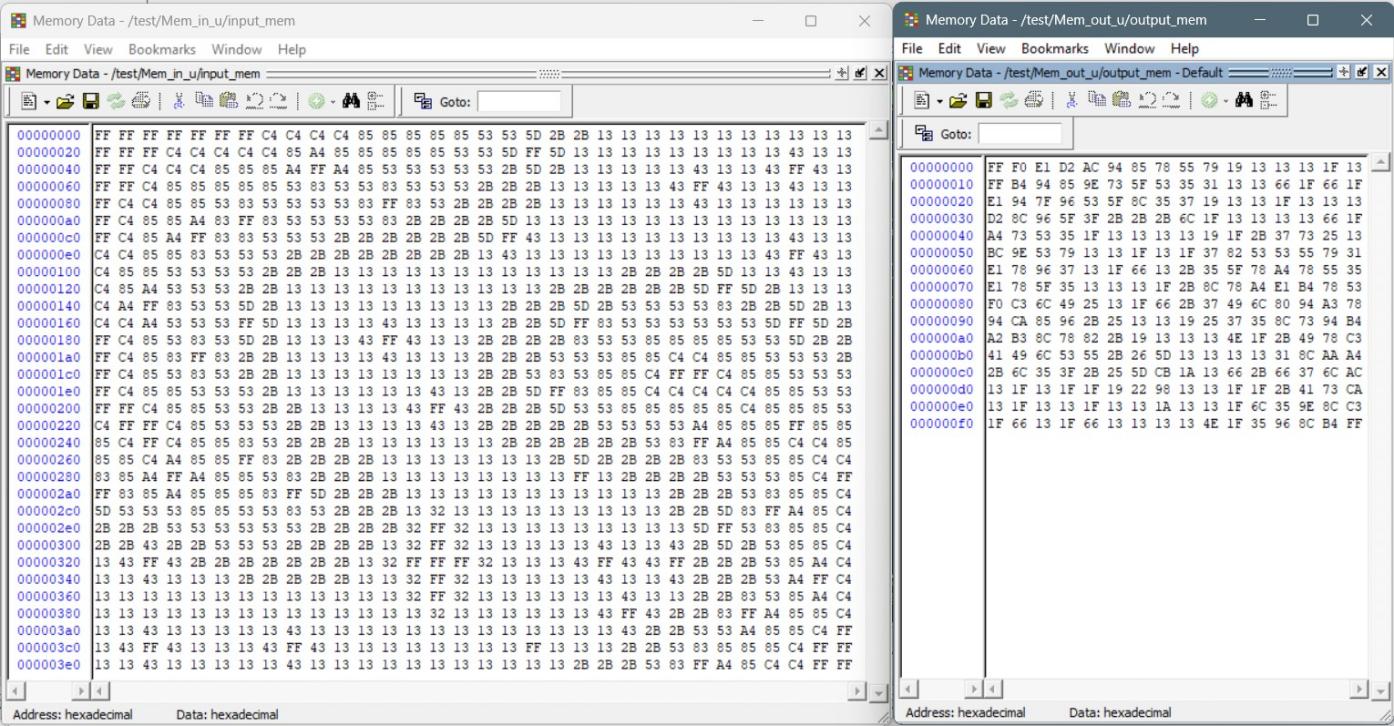


Hình 8:Bộ nhớ vào và bộ nhớ ra tại thời điểm 5020ns,

average pooling cho điểm ảnh tại địa chỉ 29.



Hình 9:Các tín hiệu quá trình average pooling cho điểm ảnh 28 và 29.



Hình 10:Bộ nhớ vào và bộ nhớ ra tại thời điểm 44210ns,

quá trình average pooling hoàn thành.

|  |  |
| --- | --- |
| Grayscale_Galaxy | Grayscale_Galaxy_Pooled |
| Hình 11:Tổ chức dạng cây của các file mã nguồn. | Hình 12:Tổ chức dạng cây của các file mã nguồn. |

# Kết luận

Kết quả mô phỏng chứng tỏ thiết kế của bộ Pool2D hoạt động đúng yêu cầu.

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 phép tính pooling 2D. 5](#_Toc6202)

[Hình 2. Giao diện ghép nối I/O. 6](#_Toc6447)

[Hình 3: Mô hình máy FSMD. 8](#_Toc17618)

[Hình 4: Cấu trúc của đơn vị xử lý dữ liệu Datapath. 9](#_Toc1238)

[Hình 5: Máy FSM của đơn vị điều khiển. 10](#_Toc20658)

[Hình 6: Sơ đồ khối tổng thể của thiết kế. 11](#_Toc2903)

[Hình 7:Tổ chức dạng cây của các file mã nguồn. 12](#_Toc31468)

[Hình 8:Bộ nhớ vào và bộ nhớ ra tại thời điểm 5020ns, 13](#_Toc7949)

[average pooling cho điểm ảnh tại địa chỉ 29. 13](#_Toc9662)

[Hình 9:Các tín hiệu quá trình average pooling cho điểm ảnh 28 và 29. 13](#_Toc5474)

[Hình 10:Bộ nhớ vào và bộ nhớ ra tại thời điểm 44210ns, 14](#_Toc25725)

[quá trình average pooling hoàn thành. 14](#_Toc24968)

[Hình 11:Tổ chức dạng cây của các file mã nguồn. 14](#_Toc7680)

[Hình 12:Tổ chức dạng cây của các file mã nguồn. 14](#_Toc13758)

List of Tables

[Bảng 1: Mô tả các tín hiệu vào ra. 6](#_Toc22927)

References

1. https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1