# 國立清華大學 電機工程學系

# 實作專題研究成果報告

# Implementation of Low-Power Convolutional Neural Network Accelerator

# 低功耗卷積神經網路加速器實作

專題領域： 系統領域

組　　別： A290

指導教授： 張孟凡 教授

組員姓名： 邵緯達

研究期間： 2022 年 7 月 1 日至 2023 年 5 月 1 日止，共 10 個月

# Department of Electrical Engineering, National Tsing Hua University
# Special Topic on Implementation Research Report

# Implementation of Low-Power Convolutional Neural Network Accelerator

# 低功耗卷積神經網路加速器實作

Major Category: System

Group Number: A290

Advisor: Chang, Meng-Fan

Members: Shao, Wei-Da

Research Period: From (2022/07/01) to (2023/05/01).

# Abstract

Artificial intelligence has gained significant attention and advancements in recent years, with machine learning being a major part of it. Deep neural networks (DNN) are the most widely used and applied branch of machine learning. As computational complexity and data size increase, accelerators are needed to improve model training and performance.

Convolution operation is a fundamental mathematical operation in AI, commonly used in signal processing tasks such as image and audio processing. Its main purpose is to extract local features of input signals and transform them into output features. Convolution operations usually involve parameters such as convolution kernel, stride, and padding, which can be adjusted according to task requirements. The output of convolution operations can be fed into the next layer of convolution or fully connected layers for further feature extraction and tasks such as classification or regression.

Convolution neural network (CNN) accelerator is a specialized hardware device for accelerating CNN model inference. It is usually used in conjunction with CPU or GPU to improve the speed and efficiency of model inference. The design of CNN accelerator is based on the characteristics of CNN models, so it can efficiently execute convolution operations and other common CNN operations such as pooling and activation functions. These hardware devices typically use low-power and high-efficiency architecture and optimize memory and computing resource utilization to provide higher performance.

The goal of this project is to implement a CNN accelerator with a circuit design based on the architecture of the paper [1], with additional self-design circuit (Asynchronous FIFO, Clock Gating, and Psum Pointer), aiming to achieve energy savings. According to simulation results, the energy consumption of the circuit in the designed architecture is improved by 19.43% compared to the architecture without these design.

# Content

## 1. Motivation

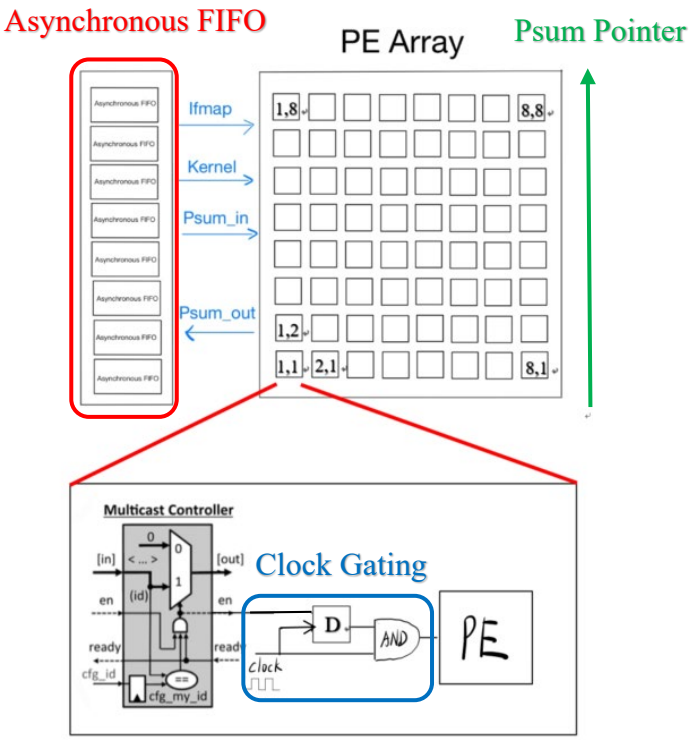
In recent years, convolutional neural networks (CNNs) have shown impressive performance in various fields, including computer vision, speech recognition, and natural language processing. However, the computational complexity of CNNs is quite high, which leads to significant energy consumption and limits their deployment on resource-constrained devices. Therefore, low power CNN accelerator design has become a crucial research topic, and there is a need for efficient CNN accelerators that can deliver high performance while consuming low power.

## 2. Purpose

The purpose of this project is to design a low-power convolutional neural network (CNN) accelerator providing high-performance processing capabilities with minimal energy consumption. Then, a comparison was made on the self-designed architectures to understand the impact of different circuit designs on the energy consumption, area, and timing of the new design accelerator. Finally compare the new design structure with the structure demonstrated on the paper.

## 3. Methodology

### 3-1. Structure Overview



## 3-2. Low Power Design

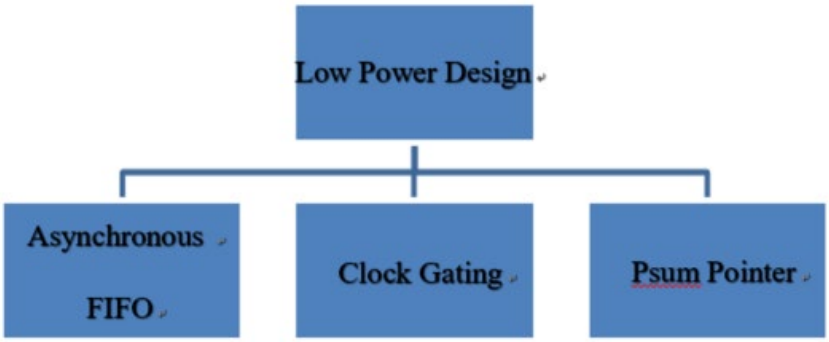



Fig. 3-1 Structure of Project

## 3-3. Detail

### 3.3.1 Asynchronous FIFO

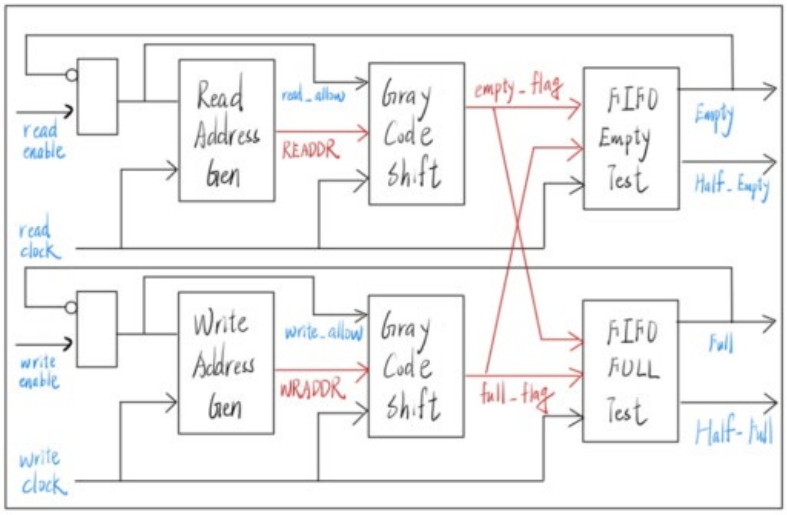



Fig. 3-2 Structure of Asynchronous FIFO

Asynchronous FIFOs [2] use separate read and write pointers to keep track of the data transfers. The write pointer indicates the location in the FIFO where the next data should be written, while the read pointer indicates the location from where the next data should be read.

Asynchronous FIFOs use a handshaking mechanism to control the flow of data between the read and write operations. The write operation sends a request to the read operation when data is available to be read. The read operation then acknowledges the request and reads the data from the FIFO

The empty and full detection conditions under the Gray code pointer are:

- When the most significant bit and the second most significant bit are the same, and the rest of the bits are the same: FIFO is empty.
- When the most significant bit and the second most significant bit are opposite, and

the rest of the bits are the same: FIFO is full.

Therefore, the final empty and full detection method is to convert the binary pointer to Gray code, use it to be received in another clock domain, and then perform detection according to the detection conditions.

### 3.3.2 Clock Gating

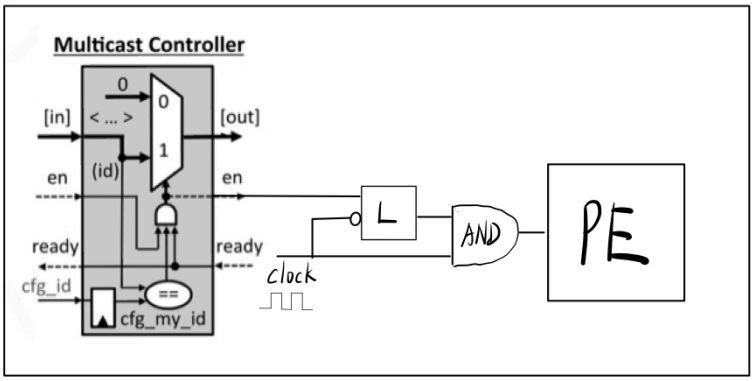



Fig. 3-3 Structure of Clock Gating

The clock gating signals are controlled by a power management unit that monitors the activity in the circuit and determines when each part can be turned off. This technique allows the work to achieve high energy efficiency while maintaining high performance for convolution calculation

### 3.3.3 Row Stationary and Psum Pointer

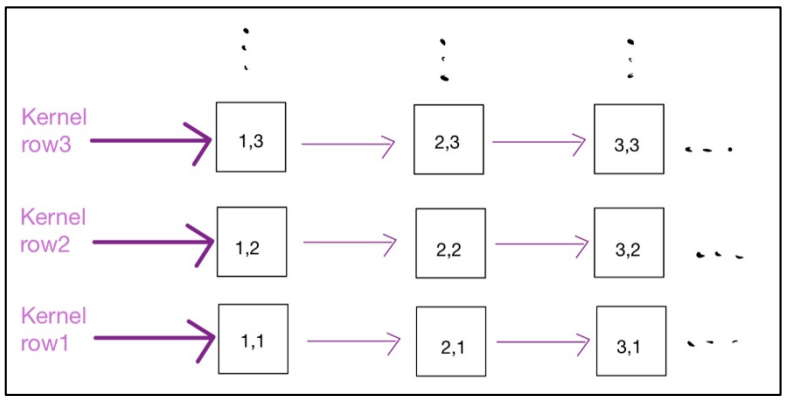



Fig. 3-4 Kernel Data Flow

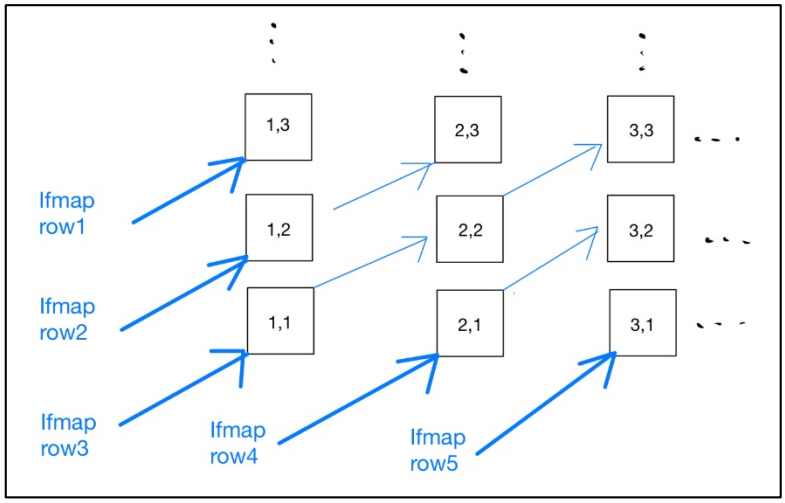



Fig. 3-5 Ifmap Data Flow

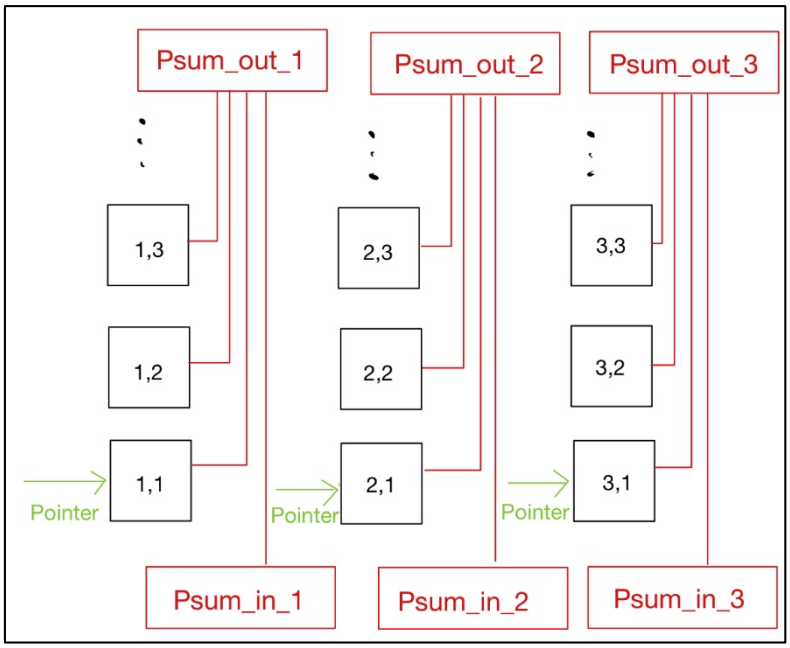



Fig. 3-6 Psum Data Flow

A pointer-based method was used to transmit Psum directly to the output. Then, at the appropriate time, the values obtained from the PE multiplication are accumulated into the output through the pointer. This approach reduces the number of times that each PE is accessed by Psum and eliminates the need for Psum registers in the PEs, resulting in reduced energy consumption.

# 4. Experimental Results

The following measurements were obtained through RTL simulation using Verilog to construct the above architecture. The correctness of the circuit was verified using 1000 sets of 8x8 ifmaps and 3x3 kernels. (Note that the difference in power

consumption and timing between implementing clock gating using Latches or Flip-Flops is not significant. Therefore, in the results presented below, only the differences in the overall architecture are shown in the area section. )

## 4.1. Simulation Result of Power Consumption

| | Power Consumption (W) | Percentage |
|---|---|---|
| w/o Low Power Design | 3.406e-03 | 100% |
| Asynchronous FIFO | 3.156e-03 | 92.65% |
| Clock Gating (Latch) | 3.264e-03 | 95.84% |
| Clock Gating (Flip-Flop) | 3.254e-03 | 95.53% |
| Psum Pointer | 3.068e-03 | 90.08% |
| Overall (with Flip-flop) | 2.744e-03 | 80.57% |

The data indicate that Asynchronous FIFO and Psum Pointer save the most energy. And the overall circuit result a 19.43% decrease in power consumption.

## 4.2. Simulation Result of Area

| | Area ($\mu m^2$) | Percentage |
|---|---|---|
| w/o Low Power Design | 32468.381337 | 100% |
| Asynchronous FIFO | 33481.394823 | 103.12% |
| Clock Gating (Latch) | 30724.829269 | 94.37% |
| Clock Gating (Flip-Flop) | 33348.274472 | 102.71% |
| Psum Pointer | 33098.267931 | 101.94% |
| Overall (with Flip-flop) | 34666.490757 | 106.77% |
| Overall (with Latch) | 31688.880764 | 97.63% |

The use of asynchronous FIFO results in the largest increase in area. And the overall circuit result a 6.77% increase in area with implementing clock gating by Flip-Flop, and 2.37% decrease in area with implementing clock gating by Latch.

## 4.3. Simulation Result of Timing

| | Timing(ns) | Percentage |
|---|---|---|
| w/o Low Power Design | 9.7122 | 100% |
| Asynchronous FIFO | 9.6582 | 99.44% |
| Clock Gating (Latch) | 9.7179 | 100.05% |

| | | |
|---|---|---|
| Clock Gating (Flip-Flop) | 9.7182 | 100.06% |
| Psum Pointer | 9.9133 | 102.07% |
| Overall (with Flip-flop) | 9.7461 | 100.03% |

For the clock gating architecture, whether using latch or flip-flop, the timing variation is minimal, indicating that the critical path remains almost unchanged. For Asynchronous FIFO, there is a significant reduction in timing. However, for the Psum Pointer architecture, the timing significantly increases, which means a longer critical path is introduced. Therefore, in the overall architecture, timing only has a slighty increase.

# 5. Conclusion

This project employs the techniques of asynchronous FIFO, clock gating, and Psum pointer to achieve low power design. From the simulation results in Part 3, it can be seen that the designed architecture meets my original goal of low power consumption. These architectures improve the utilization of PE and reduce the dynamic power consumption (switching loss) of the circuit. The overall architecture reduces power consumption by 19.43%, with only a 0.03% increase in timing. The area, on the other hand, increases by 6.77% (Flip-Flop) or decreases by 2.37% (Latch) depending on the clock gating approach used.

However, there are trade-offs between each part of the design. To achieve maximum energy savings, the Psum Pointer architecture should be used to reduce the switching losses of each register. At the same time, the increase in overall circuit area caused by this architecture is small, but it will lead to a longer critical path.

Next, the asynchronous FIFO saves the second most energy, but it also causes a significant increase in circuit area, and the effect on timing is shorten due to read and write can be occurred simultaneously.

Finally, for clock gating, since the architecture[1] already has clock gating built in, moving the clock gating structure out of the PE only results in limited energy savings. At the same time, using flip-flop will also slightly increase the area, while using latch can significantly reduce the circuit area, but it is also more prone to make the circuit unstable, situations such as toggle might happen.

# 6. Reference

[1] Y. -H. Chen, T. Krishna, J. S. Emer and V. Sze, "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," in *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017

[2] Xin Wang, T. Ahonen and J. Nurmi, "A synthesizable RTL design of asynchronous FIFO," *2004 International Symposium on System-on-Chip, 2004. Proceedings.*, Tampere, Finland, 2004

[3] J. Shinde and S. S. Salankar, "Clock gating — A power optimizing technique for VLSI circuits," *2011 Annual IEEE India Conference*, Hyderabad, India, 2011

# Review and reflections

A very solid training plan was carried out during this year's project process. During the summer break, I first learned some software that might be used in the workstation. In the first semester, after the field exploration at the beginning of the semester, the decision was made on the topic for the project. I started by reading papers in related fields and oral presentation was conducted to present the solution proposed in the paper. Then, during the winter break, I learned the digital IC design process and related practical applications. Finally, in the second semester, I selected a paper to simulate, analyzed the simulation results, and proposed a new architecture for improvement.

In the implementation process, it took about two months to simulate the selected paper in the implementation process, and then spent more than a week on each of the three new architectures that I had decided on to gradually implement them.

Throughout the entire project, I am very grateful for the assistance of the mentors. At first, I did not have a very good understanding of the digital IC design process, but the mentors provided many resources and exercises to help me gradually understand how to build circuits with hardware languages. During the process of simulating the paper and implementing the new architecture, they also gave me many different ideas and solutions to problems I encountered.

Finally, I am very appreciate to the professor for allowing me to conduct research in the laboratory. Although the research process this year was not very smooth, it was very solid. Whether it was guidance in the professional field or advice on oral presentations and results presentation, I have gained a lot from it.