Implementation of CNN Hardware Accelerator based on Eyeriss structure

KU KONKUK UNIVERSITY

System-on-a-Chip Design LAB

최동훈, 박현주

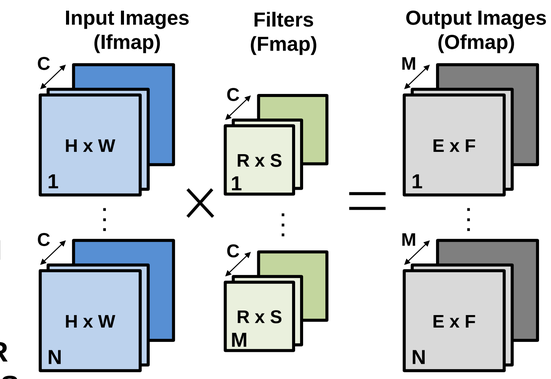
Supervised by **Chester Sungchung Park**

1 Abstract

CNN operations require convolution operations, data loads, and storage. Accessing DRAM for all data significantly increases execution time and consumes energy. In order to reduce access to the DRAM where the data is stored, we used Processing Element (PE) proposed by Eyeriss. We also **designed the entire data flow**, such as network on chip (NoC), address generator unit (AGU) and buffer controller.

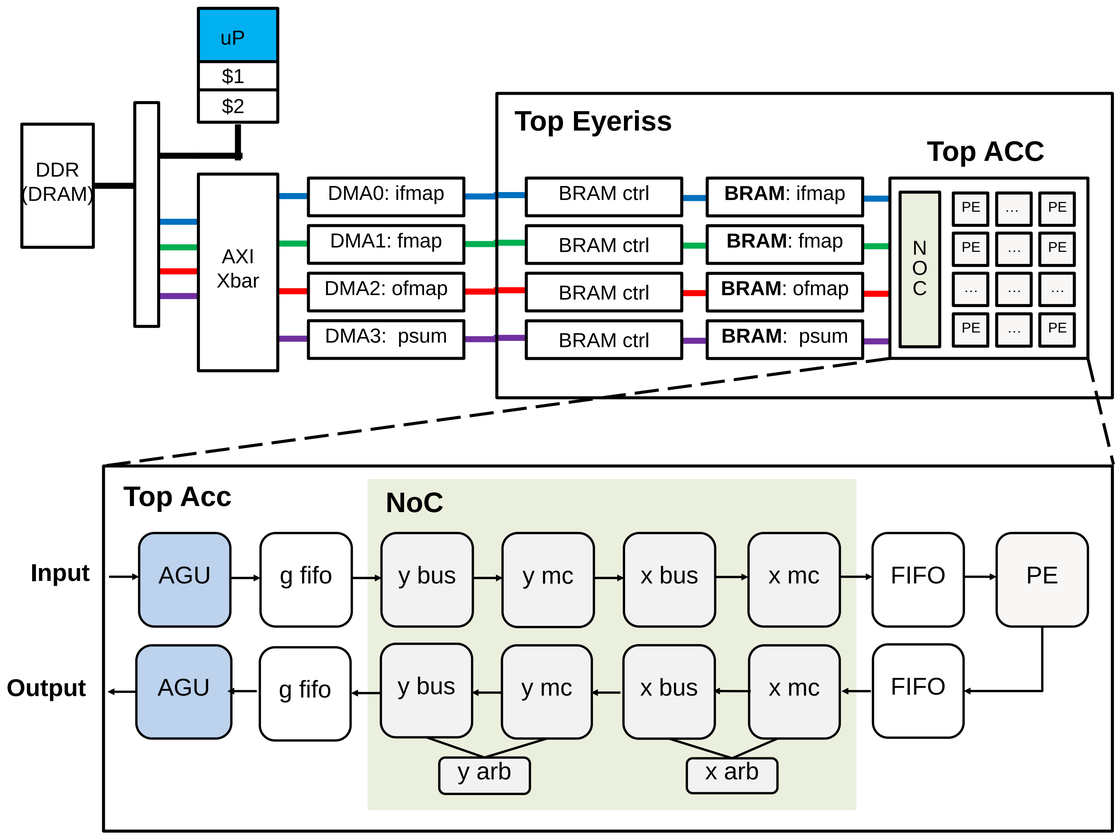
2 CNN

CNN consists of multidimensional convolutions that require time and energy to calculate, load and store data. The worst case is accessing DRAM every time.



for **n** in **N**
for **c** *in* **C**
for **m** *in* **M**
for **e** *in* **E**
for **f** *in* **F**
for **r** *in* **R**
for **s** *in* **S**
O [n][m][e][f] += **I** [n][c][e + r][f + s] * **W** [m][c][r][s]

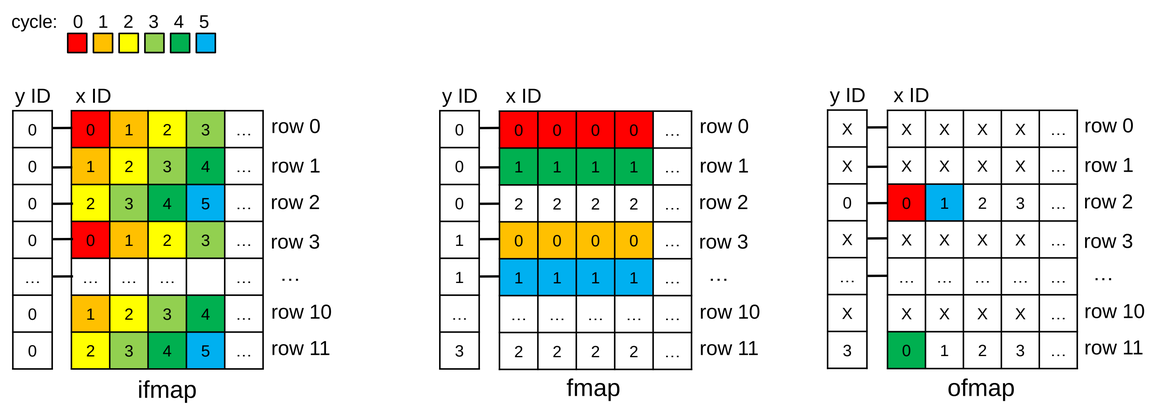
3 Structure



- It reduces DRAM access by storing required data in **BRAM**.
- The X-Y bus uses round robin arbitration to send calculation results from multiple mcs to the next element.

4 Using Data in Parallel

Each y mc(multicast controller) and x mc have an id. In AGU, a packet containing the y and x tags is attached to the data to be sent. The mc interprets the sent packet and gets the data if the tag matches its id.



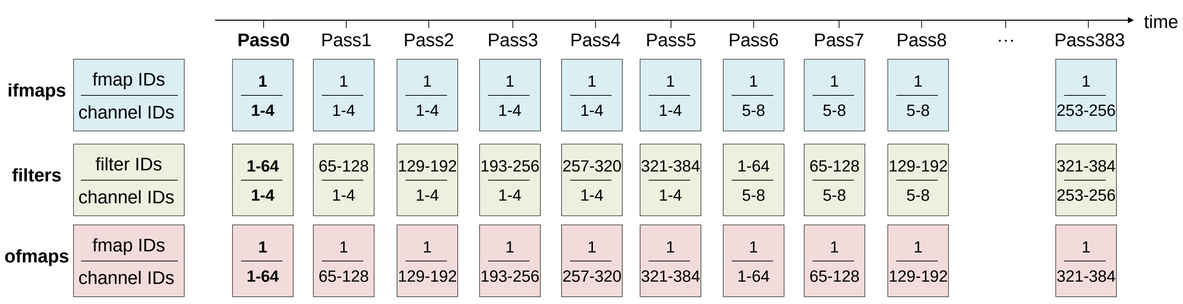
Example: mapping ids, conv. layer 3.

5 Processing Pass

The processing pass is a unit in which the accelerator performs an operation after loading data and stores the results back. In the following example, the input image data is loaded once per six processing passes.

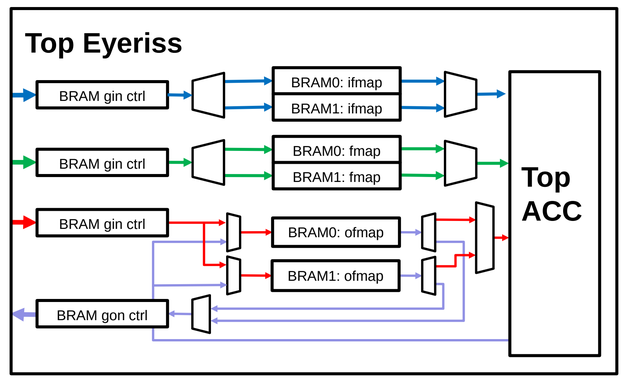
Example: Processing passes, conv. layer 3.

Layer	CNN Shape Parameters					
	H/W^1	R/S	E/F	C	M	U
CONV3	15	3	13	256	384	1



6 Double Buffer

As the example in the previous 5-Processing Pass, the same ifmap is used in six passes, and ofmap is repeated every six passes. Considering that ofmap is used repeatedly, we used a double buffer to reduce the number of ofmap loads.



7 Experiment Environment

Xilinx Zynq 706
Alexnet Convolution Layer 3
- Size of ifmap: 15x15, fmap: 3x3, ofmap: 13x13
- Size of input channel C: 256,
  output channel W: 384

Eyeriss parameters of Layer 3
- p: 16, q: 4, r: 1, t: 4

8 Result

Model	Cycle	Time Required (s)	Comparison
SW	149,520,384	2.99	x1
HW(SB_t1)	15,604,224	0.312	x9.58
HW(SB_t4)	14,662,656	0.293	x10.12
HW(DB_t4)	5,408,167	0.108	x27.65

* SW : The plain convolution loop.
* HW(SB_t1) : Proposed accelerator with single buffer and parameter t=1.
* HW(SB_t4) : Proposed accelerator with single buffer and parameter t=4.
* HW(DB_t4) : Proposed accelerator with double buffer and parameter t=4.

Comparison of execution speed of HW(SB_t4) and SW on Zynq board. HW(SB_t4) is 13.87 times faster than SW.

```
DMA - HW run time : 4098.216309 [us]
SW run time : 56879.121094 [us]
Difference of HW and SW result : 0
```

9 Conclusion

The proposed structure decreases DRAM access and reuses data to shorten execution time. The result of the Xilinx Zynq board and the simulation shows that the execution time is reduced by up to 27 times.

* Reference

Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks – ISSCC 2016