AOC 2024 Spring - Lab 4 Processing Element

施宇庭 NN6124030

1 Implmenetation of Eyeriss PE

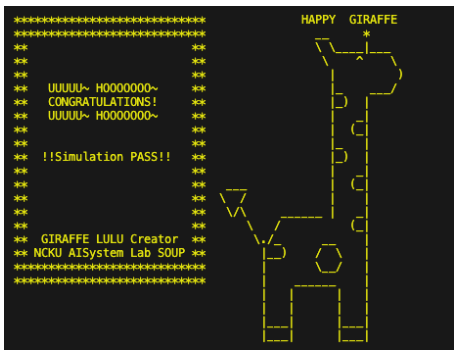

Figure 1: Testbench 0 Passed

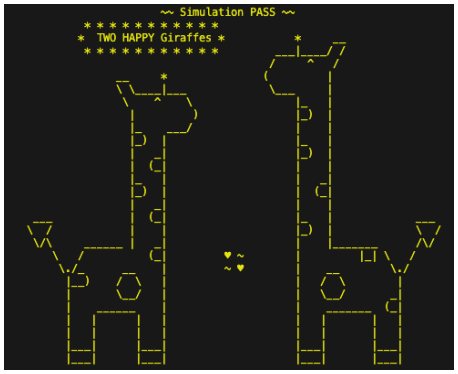

Figure 2: Testbench 1 Passed

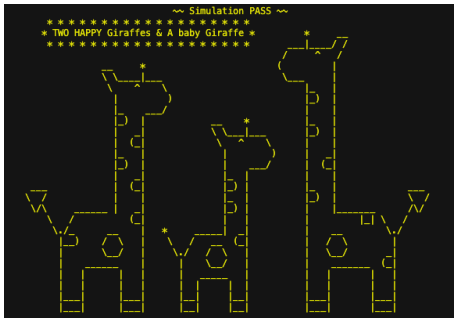

Figure 3: Testbench 2 Passed

2 PE Design

Architecture

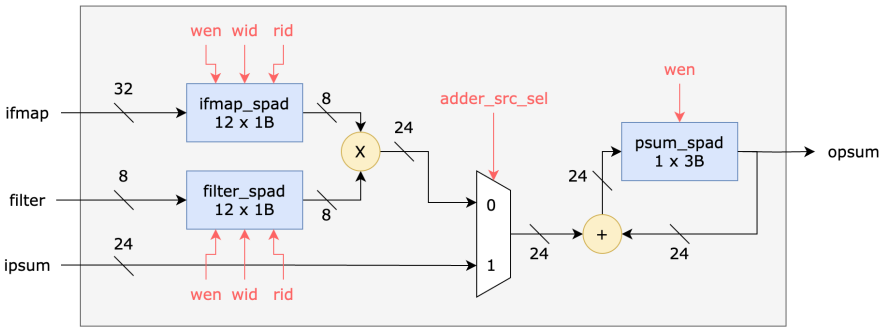

Figure 4: Hardware architecture of Eyeriss PE.

State Transition

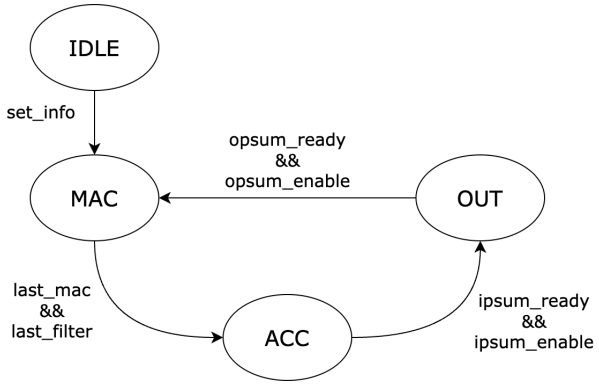

Figure 5: Finite state machine of PE controller.

Control Signals

	ifmap			filter			adder	psum
	wen	wid	rid	wen	wid	rid	src_sel	wen
IDLE	0	0	0	0	0	0	DC	0
MAC	1	<oldval>+ 4	<old_val>+1	1	<oldval>+ 1	<oldval>+1	0	1
ACC	0	DC	DC	0	DC	DC	1	1
OUT	1	0 if clear	0	1	0 if clear	0	DC	0

Table 1: Control signals of PE controller.

3 Row Stationary Dataflow with Ifmap Reuse

	batch size	channels	height	width
filter	2	2	3	3
ifmap	2	2	5	5
ofmap	2	2	3	3

Table 2: Convolution 2D shape parameters.

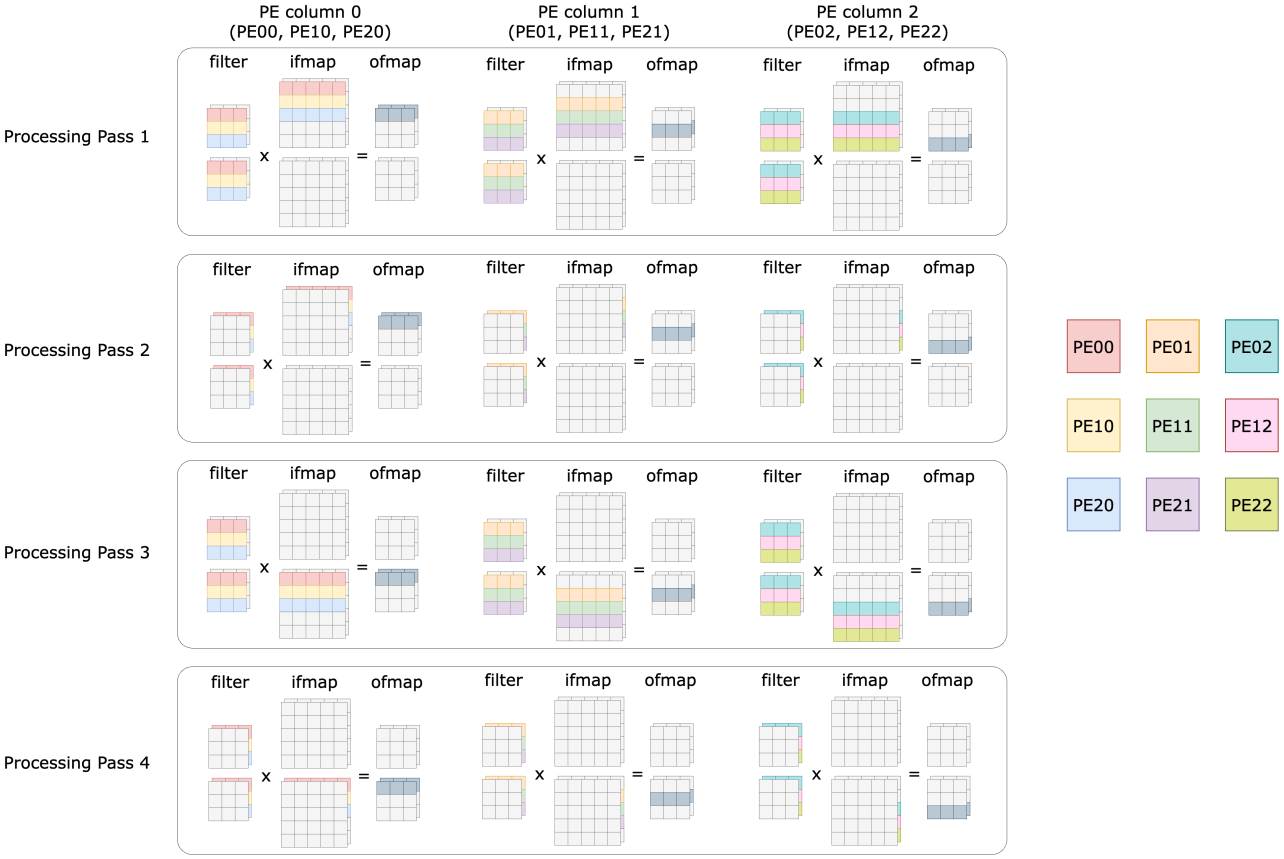

Figure 6: Processing passes and workload mapping of row stationary dataflow with ifmap reuse.

4 Comparison of Row Stationary Dataflow in Different Scenarios

	filter reuse (A)	ifmap reuse (B)	psum reuse (C)
n	2	1	1
p	1	2	1
q	1	1	2

Table 3: RS dataflow mapping prameters of filter reuse (scenario A), ifmap reuse (scenario B), and psum reuse (scenario C).

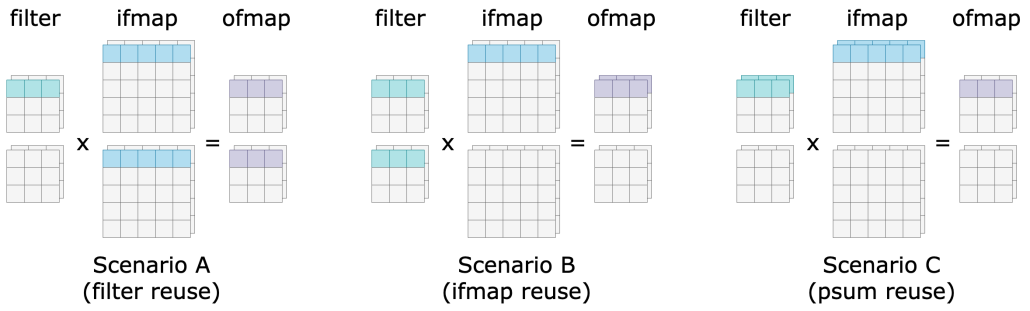

Figure 7: Schematic diagram of row stationary dataflow in different scenarios.

	filter reuse (A)	ifmap reuse (B)	psum reuse (C)
reuse distance (filter/ifmap/psum)	1/4/2	6/1/2	6/4/1
proper spad size (filter/ifmap/psum)	3/6/2	6/3/2	6/6/1

Table 4: Comparison of reuse distance, proper spad size, number of memory read/write and energy consumption between 3 scenarios.

5 Thoughts and Advices

- lab 的講解可以簡潔一點，講義和影片都太過冗長不易抓到重點
- testbench 有些行為並沒有在規格中說明清楚，例如 overflow handling、timing spec 等等
- 講義中可以不用放不相關的內容

6 Improvement of PE Design

這次的 PE 設計當中我採用了以下兩種方式來改進 PE 的運算效率：

1. zero-skipping：PE 可以支援 channel = {1, 2, 3, 4}，當 channel size 較小時，可以跳過部分 MAC 運算
2. asynchronous data fetching：不需要等到 ifmap 和 filter 全部拿完才開始進行運算，MAC computation 和 data fetching 由不同的 FSM 來控制，因此可以獨立進行