Lecture 19
Case Study: Eyeriss (Part 2)

Xuan ‘Silvia’ Zhang
Washington University in St. Louis

http://classes.engineering.wustl.edu/ese566/

Class Project Mentoring

- Yunfei (Team 2, 5, 6, 7)
- Dengxue (Team 1, 3, 4)
- Team 1
  - Andrew Ellison, Shixuan Zhang
- Team 2
  - Brett Gilpin, Matthew Wedrewer, Nestor Gonzalez
- Team 3
  - Weidong Cao, Liu Ke, Xinyao Li
- Team 4
  - Meizhi Wang, Longzhen Zhang, An Zou
- Team 5
  - Yuyang Li, Yu Liu, Qilan Ding
- Team 6
  - Chenxi Yin, Yuyao Hu
- Team 7
  - Wenmei Bo, Jizhou Huang, Bojun Li

Class Project Proposal

- Due on Monday 4/10 by noon
- Submit initial project proposal
  - system block diagrams
  - details of functional unit and interfaces
  - targeted metric for design optimization (e.g. power, area, latency, throughput)
  - proposed techniques to implement
  - details of testbench (e.g. how to feed data, simulation)
  - timeline and deliverables
  - division of labor, individual contribution
- Commit/upload to Git repository as README.md

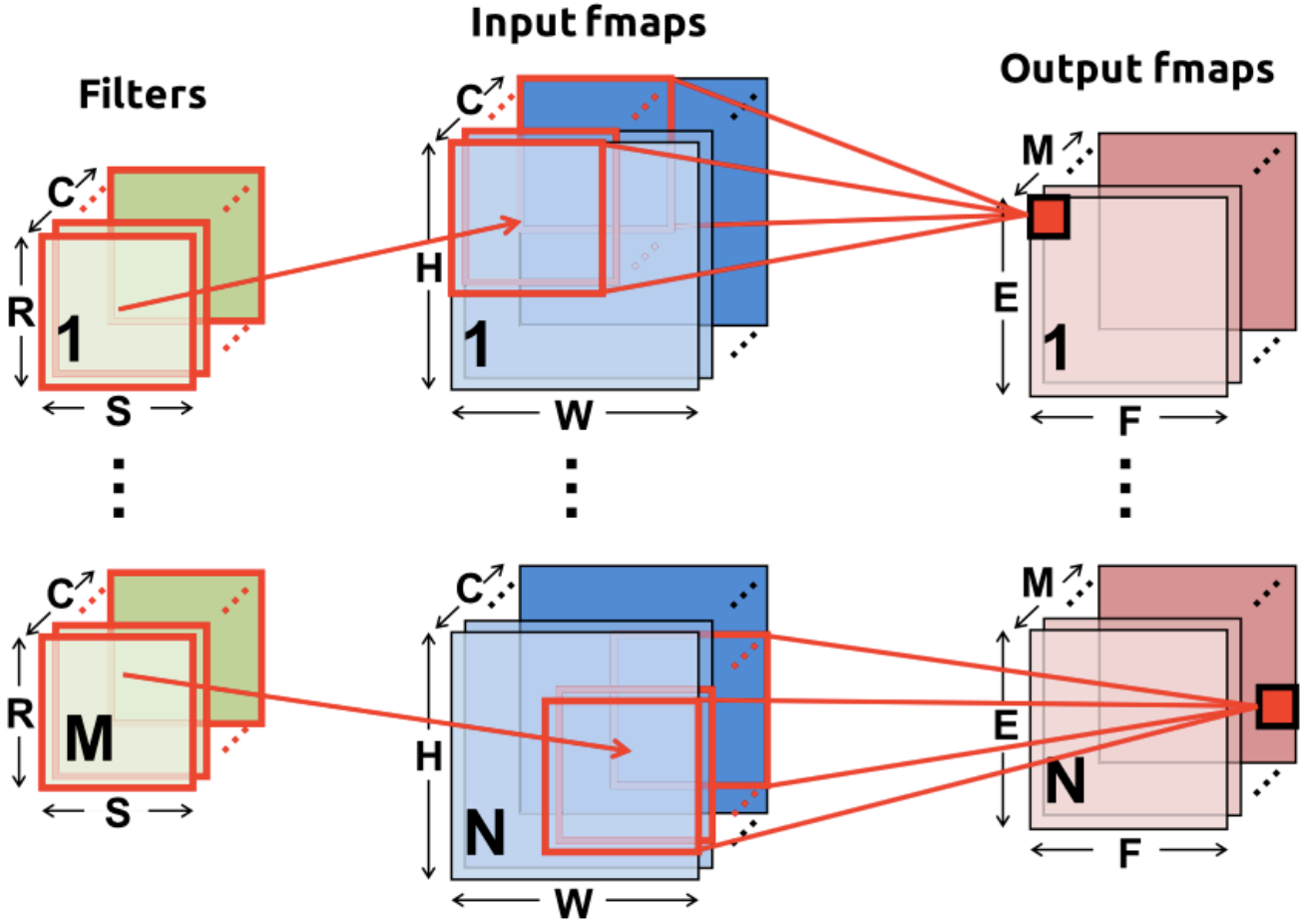
Project Discussion and Presentation (Week 13-15)

- Meet in the lecture room
- Schedule
  - Team 1: Mon 2:40-3pm
  - Team 2: Mon 3-3:20pm
  - Team 3: Mon 3:20-3:40pm
  - Team 4: Mon 3:40-4pm
  - Team 5: Wed 2:40-3pm
  - Team 6: Wed 3-3:20pm
  - Team 7: Wed 3:20-3:40pm
  - All: Wed 3:40-4pm

Eyeriss (JSSC 2017)

- JSSC
  - IEEE Journal of Solid-State Circuits
  - top journal on integrated circuits design and prototyping
- ISSCC
  - International Solid-State Circuits Conference
  - top-tier conference on IC design
  - where Intel, IBM, AMD, Nvidia, Samsung debut their newest processors

- Hardware implementation of CNN accelerators
  - experimental results following Eyeriss ISCA paper
  - data compression, clock gating, and NoC



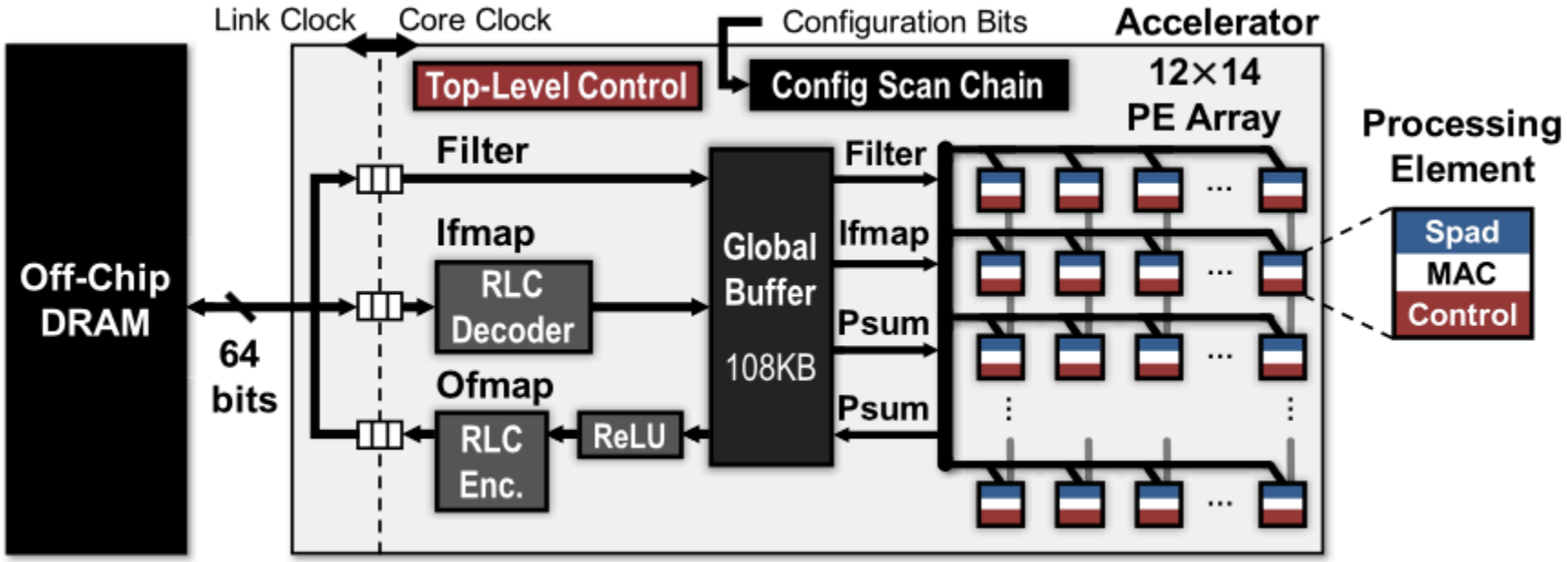
Shape Parameter	Description
N	batch size of 3D fmaps
M	# of 3D filters / # of ofmap channels
C	# of ifmap/filter channels
H/W	ifmap plane height/width
R/S	filter plane height/width
E/F	ofmap plane height/width

Eyeriss System Diagram

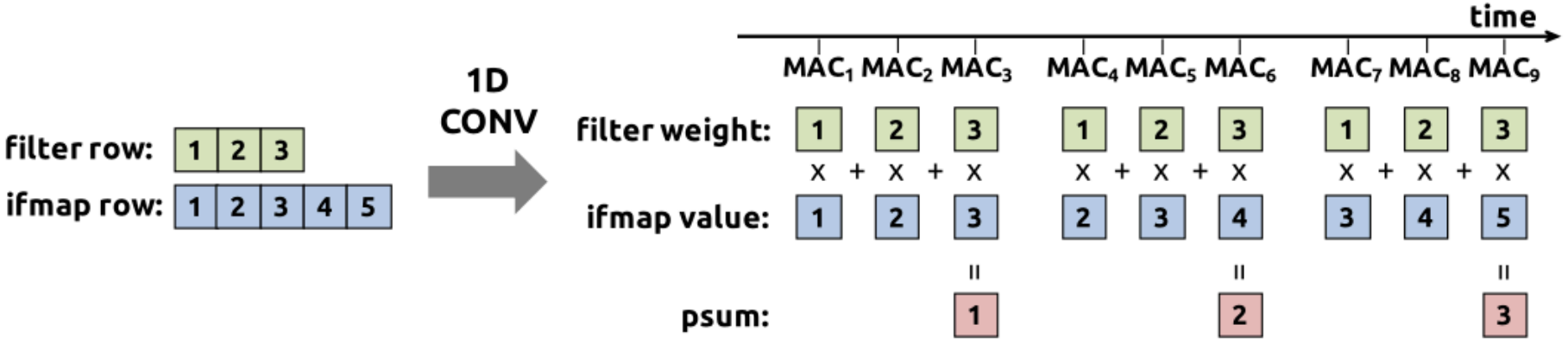
$$\mathbf{O}[z][u][x][y] = \mathrm{ReLU}\left(\mathbf{B}[u] + \sum_{k=0}^{C-1}\sum_{i=0}^{R-1}\sum_{j=0}^{S-1}\mathbf{I}[z][k][Ux+i][Uy+j] \times \mathbf{W}[u][k][i][j]\right),$$

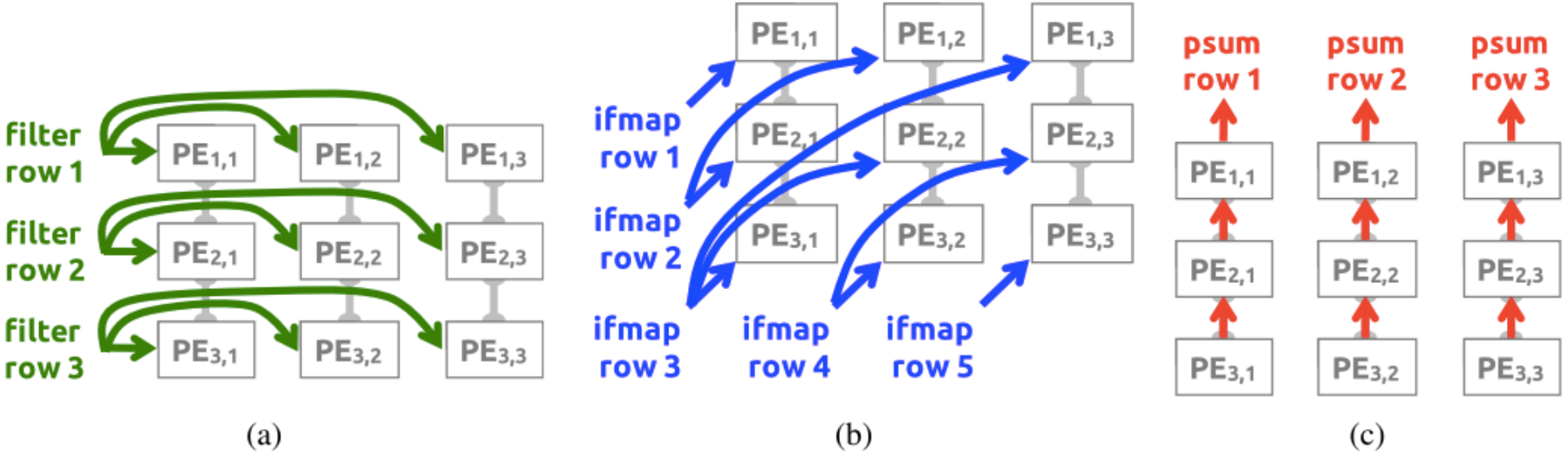
$$0 \le z < N, \quad 0 \le u < M, \quad 0 \le y < E, \quad 0 \le x < F$$

$$E = (H - R + U)/U, \quad F = (W - S + U)/U \qquad (1)$$



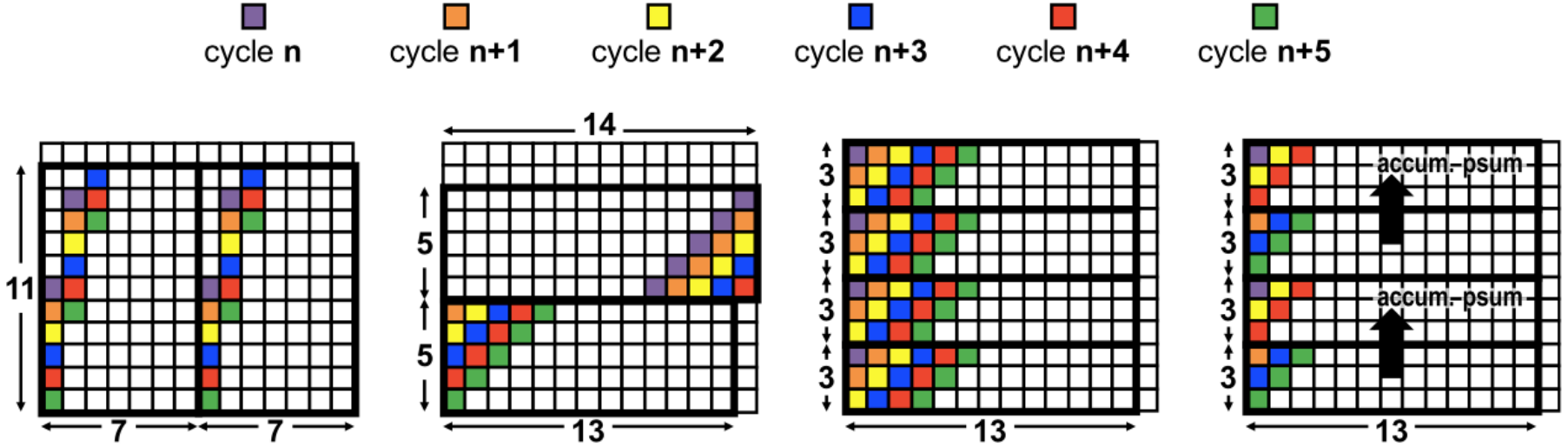
Row Stationary Dataflow





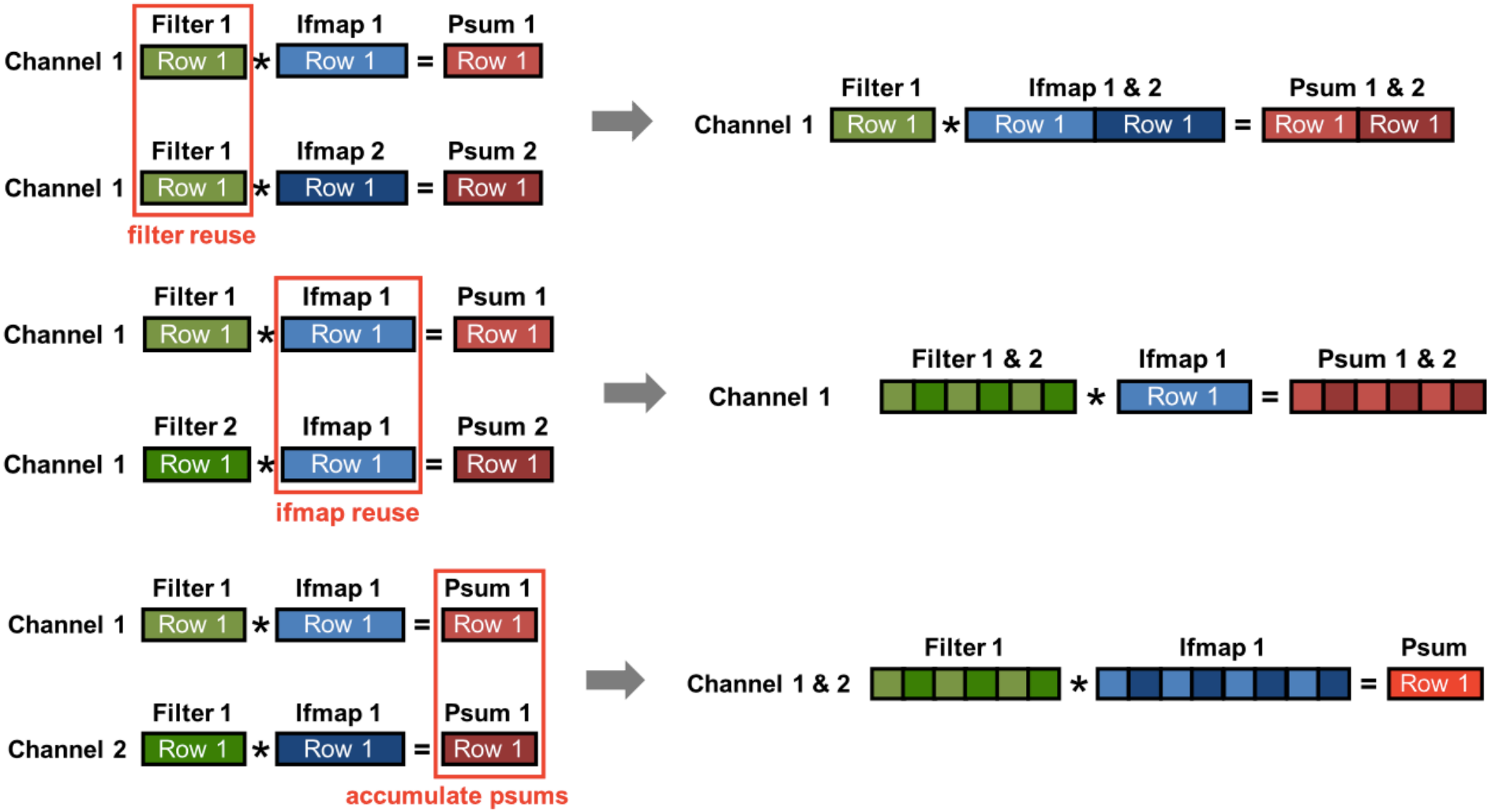
(a) (b) (c)

2-D Convolution PE Set



Layer	CNN Shape Parameters						RS Dataflow Mapping Parameters							Global Buffer Allocation	
	H/W[1]	R/S	E/F	C	M	U	m	n	e	p	q	r	t	ifmap	psum
CONV1	227	11	55	3	96	4	96	1	7	16	1	1	2	15.5KB	72.2KB
CONV2	31	5	27	48	256	1	64	1	27	16	2	1	1	3.8KB	91.1KB
CONV3	15	3	13	256	384	1	64	4	13	16	4	1	4	7.0KB	84.5KB
CONV4	15	3	13	192	384	1	64	4	13	16	3	2	2	10.5KB	84.5KB
CONV5	15	3	13	192	256	1	64	4	13	16	3	2	2	10.5KB	84.5KB

Beyond 2-D in PE Array



Processing Pass Scheduling

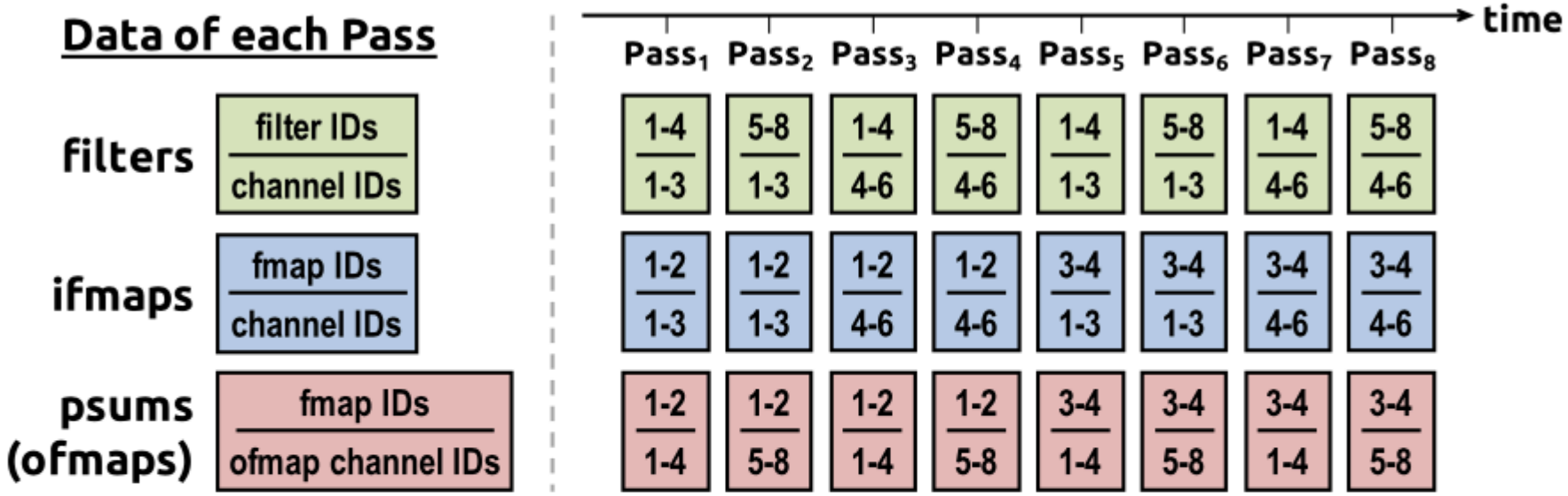


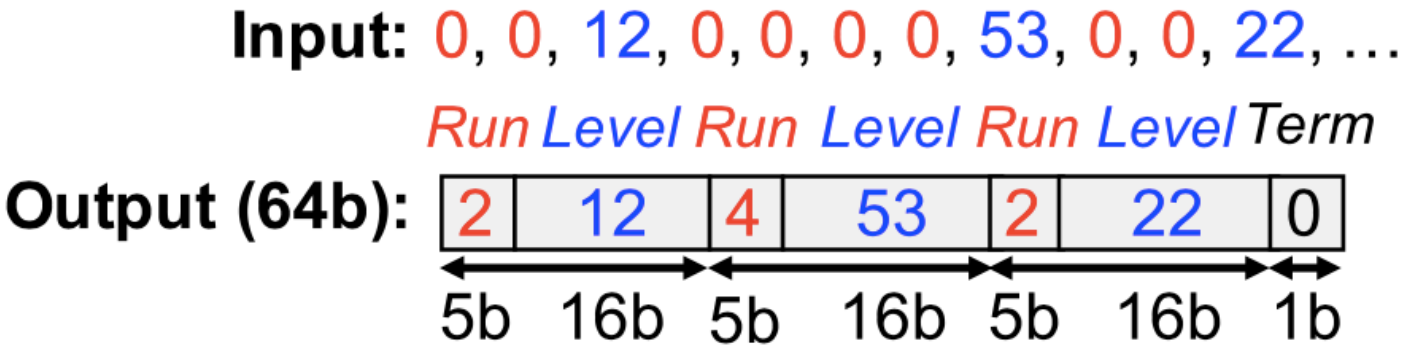
Fig. 7. Scheduling of processing passes. Each block of filters, ifmaps, or psums is a group of 2-D data from the specified dimensions used by a processing pass. The number of channels (C), filters (M), and ifmaps (N) used in this example layer created for demonstration purpose are 6, 8, and 4, respectively, and the RS dataflow uses eight passes to process the layer.

Power/Energy Saving Techniques

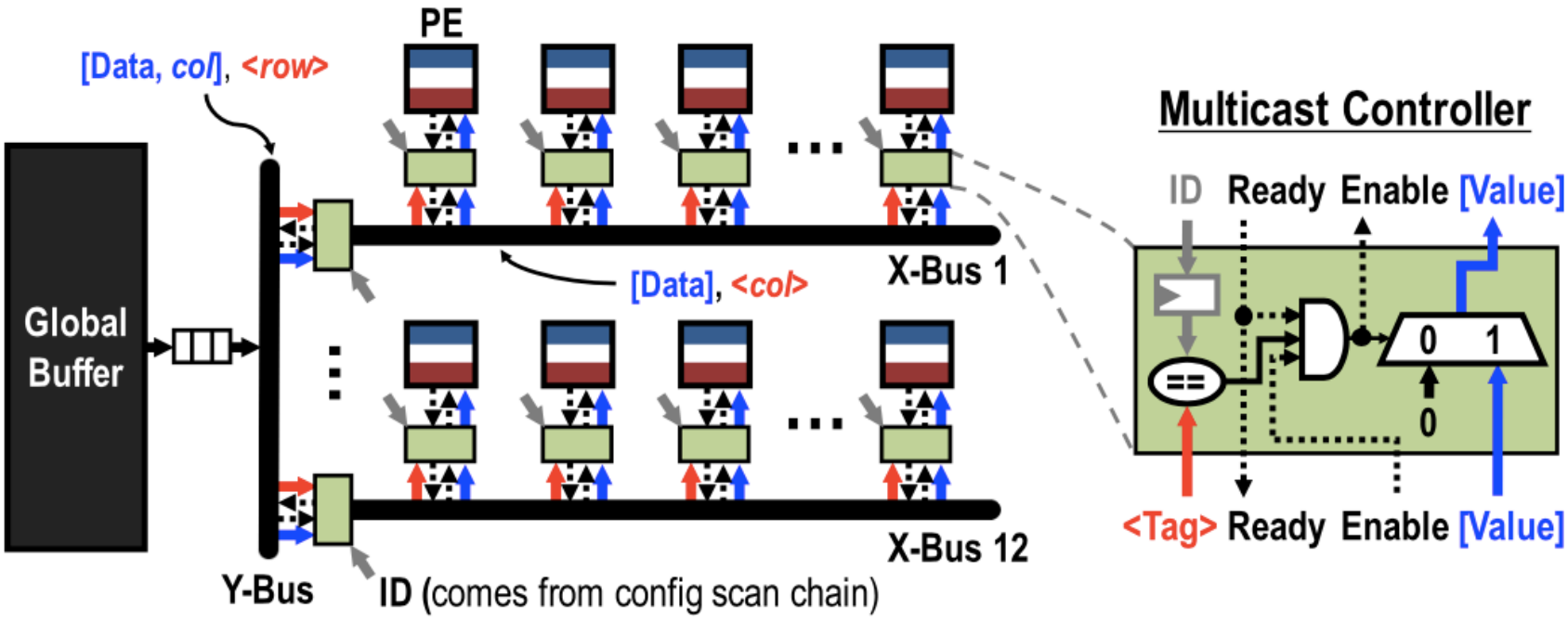
- Row stationary dataflow

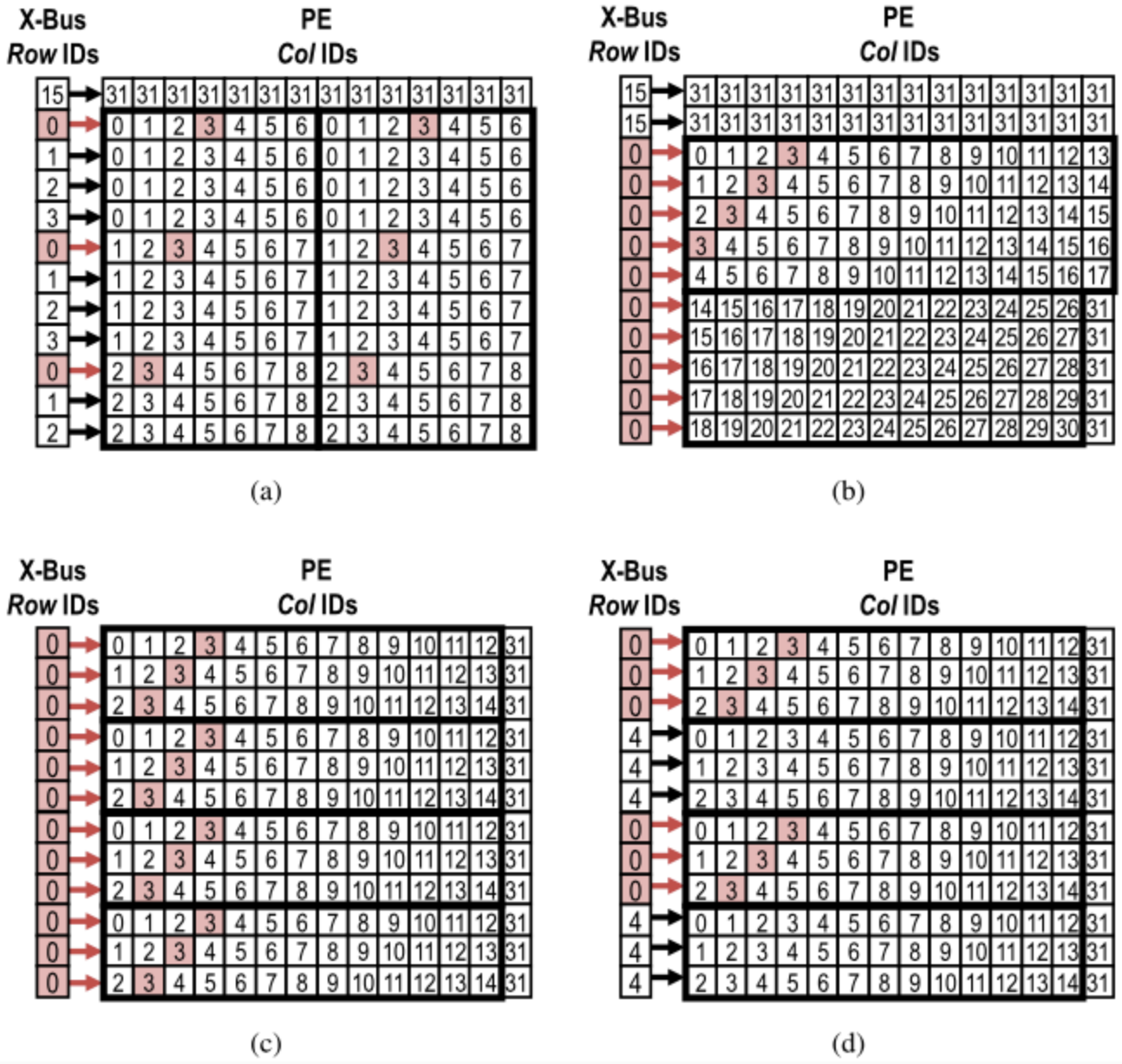
Parameter	Description
m	number of ofmap channels stored in the global buffer
n	number of ifmaps used in a processing pass
e	width of the PE set (strip-mined if necessary)
p	number of filters processed by a PE set
q	number of channels processed by a PE set
r	number of PE sets that process different channels in the PE array
t	number of PE sets that process different filters in the PE array

- Data compression
  - run-length compression (RLC)



GIN Architecture for Network-on-Chip (NoC)




X-Bus
Row IDs
PE
Col IDs
(a)
X-Bus
Row IDs
PE
Col IDs
(b)
X-Bus
Row IDs
PE
Col IDs
(c)
X-Bus
Row IDs
PE
Col IDs
(d)

PE Architecture

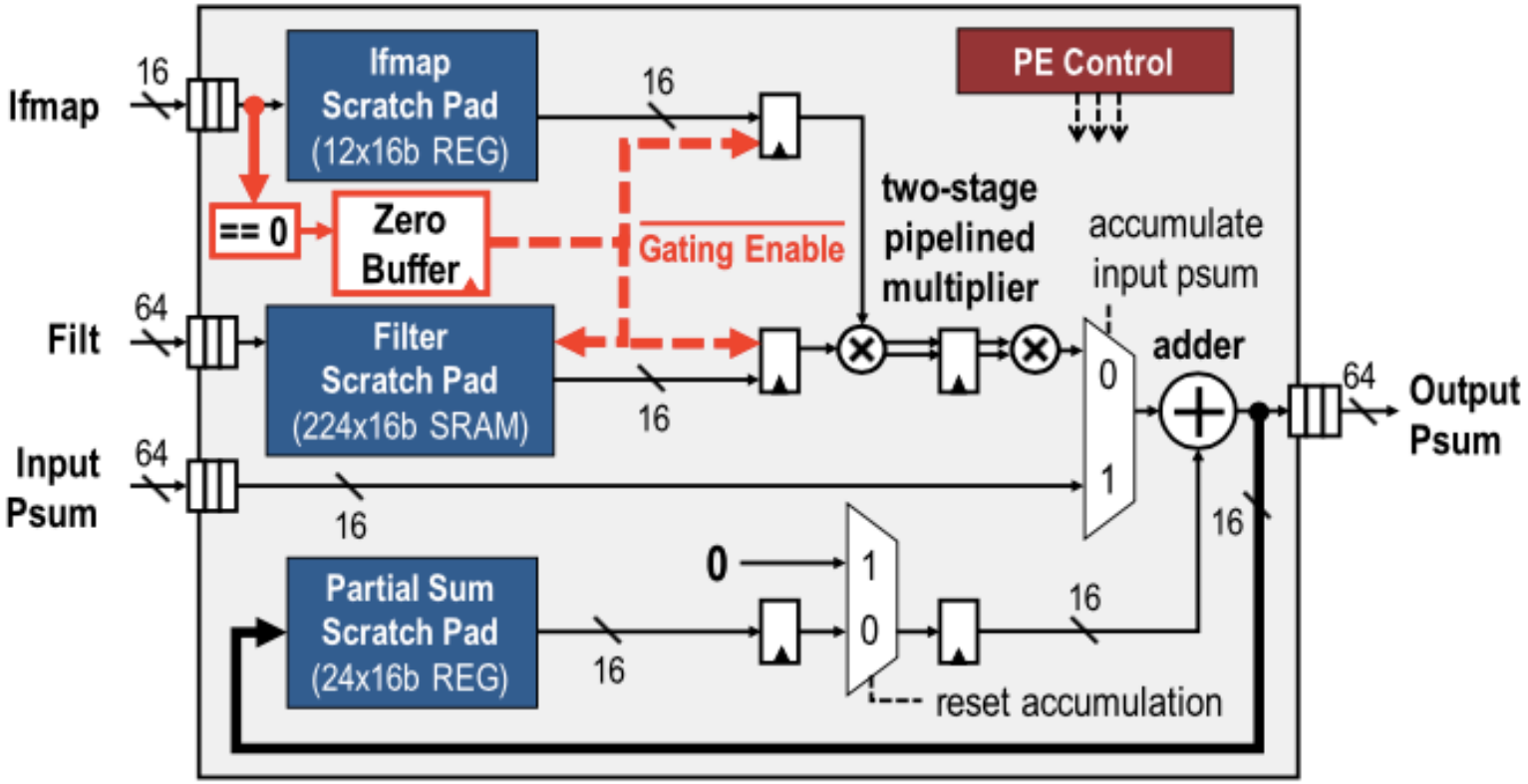


Fig. 12. PE architecture. The datapaths in red show the data gating logic to skip the processing of zero ifmap data.

Experimental Setup

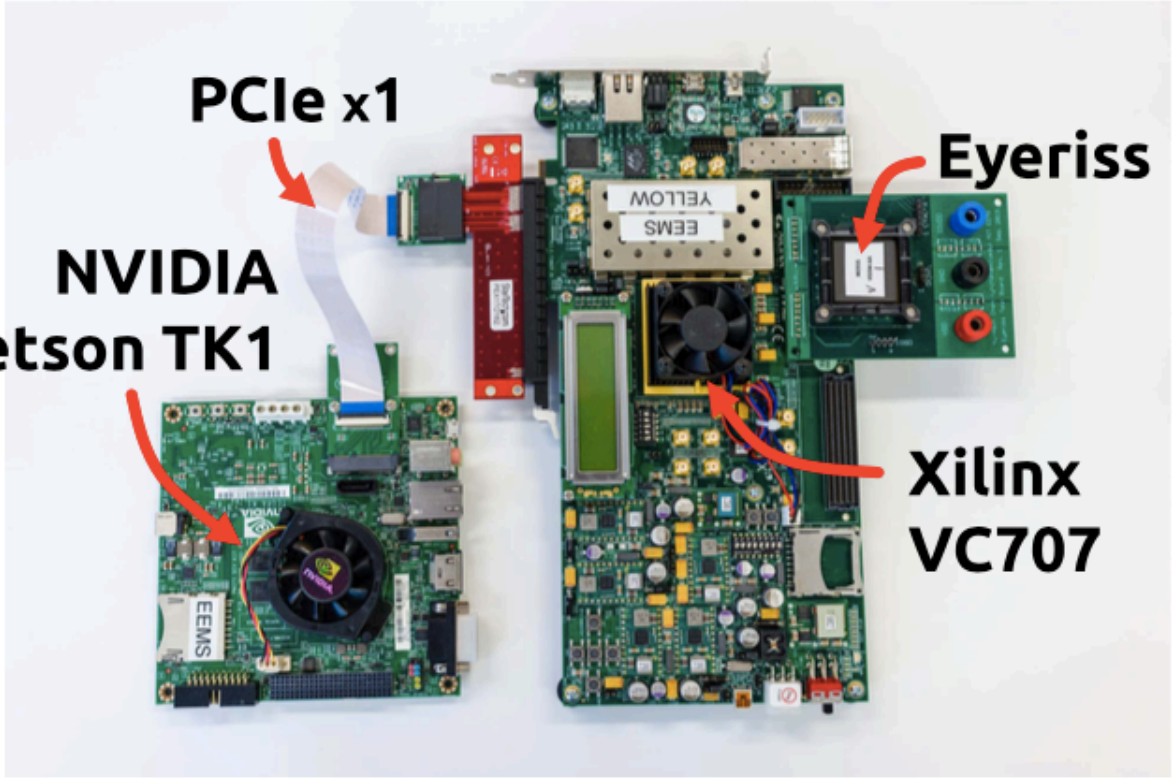
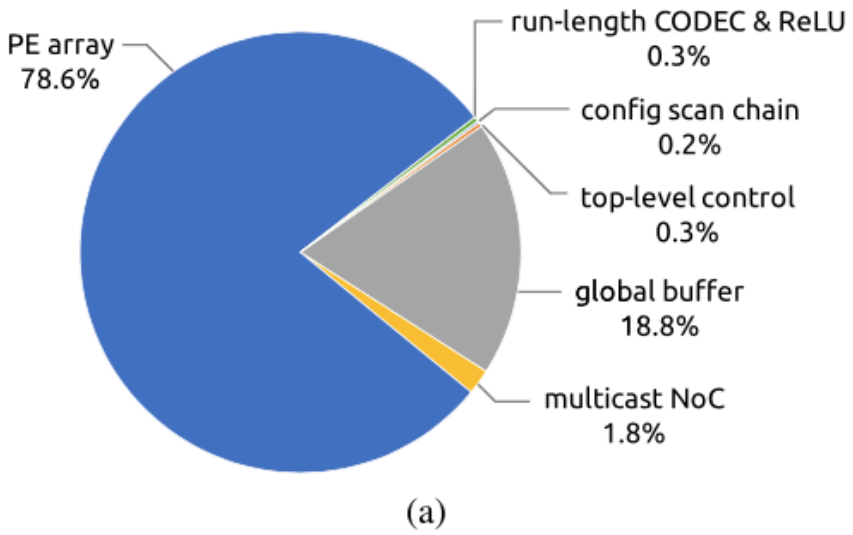
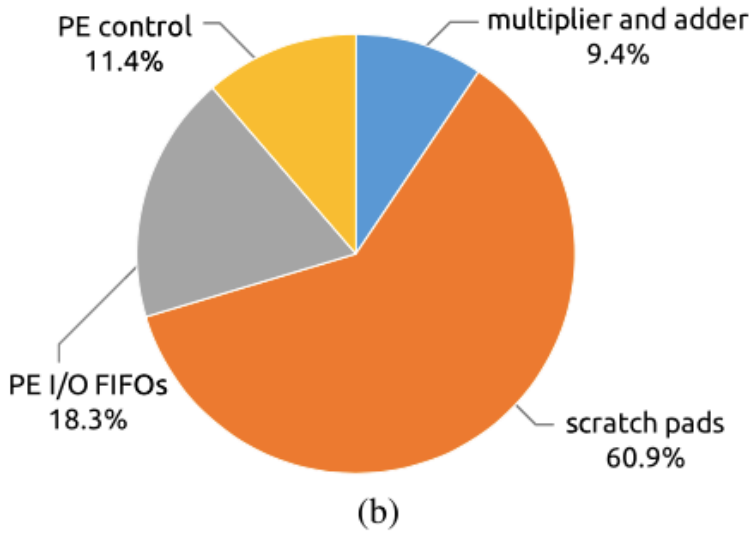


Fig. 14. Eyeriss-integrated deep-learning system that runs Caffe [28], which is one of the most popular deep-learning frameworks. The customized Caffe runs on the NVIDIA Jetson TK1 development board, and offloads the processing of a CNN layer to Eyeriss through the PCIe interface. The Xilinx VC707 serves as the PCIe controller and does not perform any processing. We have demonstrated an 1000-class image classification task [27] using this system, and a live demo can be found in [29].

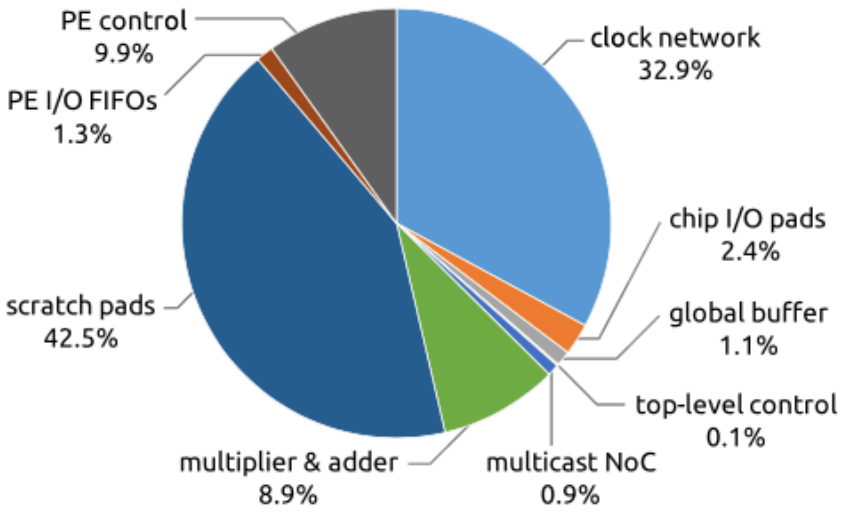
Area and Power Breakdown



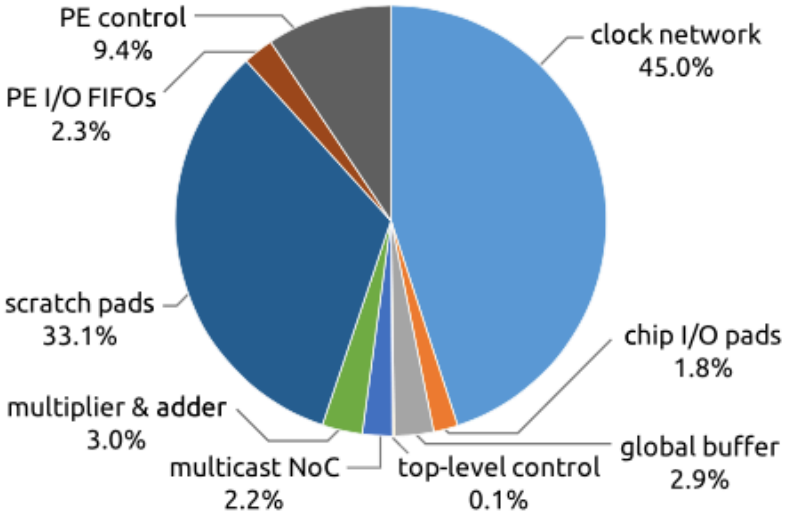
(a)



(b)



(a)



(b)

Questions?

Comments?

Discussion?