Development and Applications of Artificial Intelligence Hardware

Kun-Chih (Jimmy) Chen
Associate Professor/ Electric Junior Chair Professor

Dep. Electronics and Electrical Engineering/ Institute of Electronics,
National Yang Ming Chiao Tung University (NYCU)
Email: kcchen@nycu.edu.tw
URL: https://sites.google.com/site/cereslaben

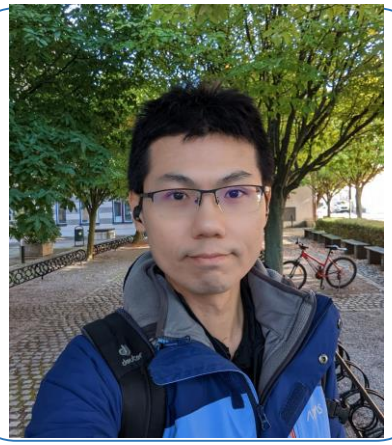
個人簡介

基本資料

姓名：陳坤志

單位：國立陽明交通大學電子研究所

職稱：副教授/ 電子青年講座教授



經歷

陳教授專長於超大型積體電路系統架構與演算法開發領域，其中包含了多核心系統晶片設計、類神經網路學習演算法設計、可靠度系統設計、超大型積體電路架構與電腦輔助系統設計等。陳教授曾擔任IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS) 國際期刊的客座編輯(Guest Editor)。此外，陳教授亦曾擔任國際研討會International Workshop on Network-on-Chip Architectures (NoCArc)大會主席(General Chair)，以及國際研討會NoCArc和IEEE MCSoC的議程主席(Technical Program Chair)。陳教授獲得多項國內外獎項的肯定，其中包含了「國科會吳大猷先生紀念獎」、「中國電機工程學會優秀青年電機工程師獎」、「臺灣積體電路設計學會(TICD)傑出年輕學者獎」、「潘文淵文教基金會考察研究獎」、「IEEE Tainan Section最佳年輕專業會員獎」等，陳教授亦是美國電機電子工程師學會(IEEE)資深會員。

演講摘要

Title

Development and Applications of Artificial Intelligence Hardware

Abstract

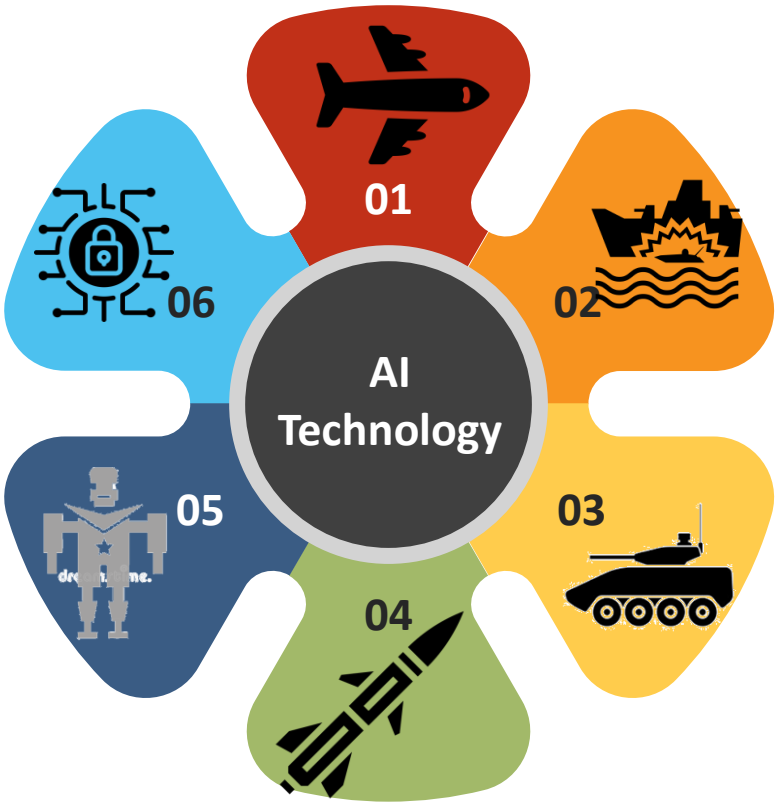
The development of artificial intelligence (AI) hardware is pivotal in advancing AI technologies across various sectors, including military applications. Innovations in specialized processors and neural network chips have greatly enhanced computational performance, enabling real-time data processing and complex algorithm execution. In defense, AI hardware is increasingly integrated into systems for surveillance, target recognition, and autonomous weaponry, where speed and efficiency are crucial for mission success. Moving forward, the integration of cutting-edge hardware with AI algorithms will be essential in unlocking new opportunities, driving technological advancements, and shaping the future landscape of artificial intelligence across both civilian and military domains. Despite these advancements, challenges such as cost, scalability, and sustainability remain significant hurdles in AI hardware development. In this talk, I will address these design challenges and present a method for creating scalable and reconfigurable AI hardware. I will conclude with two demonstrations to showcase advancements in AI technology.

AI-assisted War is the Future

Cyber Security - 06

Army Robot - 05

Missile - 04



01 - Drone

02 - Sea Drone

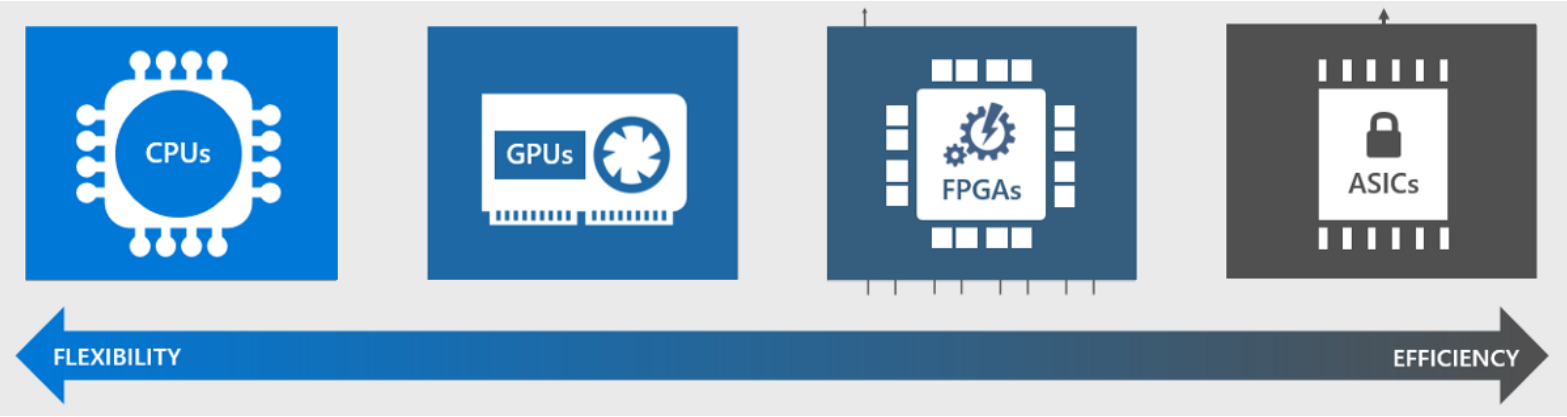
03 - Drone Tank

AI Hardware Requirements for Military Applications

- **Computational Power**
  - Real-time processing
  - Deep-learning support
- **Robustness and Durability**
  - Shock and vibration resistance
  - Temperature tolerance
  - EMI shielding
- **Power Efficiency**
  - Low-power consumption
  - Thermal management
- **Compact Size and Lightweight**
  - Portability
- **Reliability and Security**
  - Fault tolerance
  - security
- **Adaptability and Flexibility**
  - Reconfigurability
  - Scalability

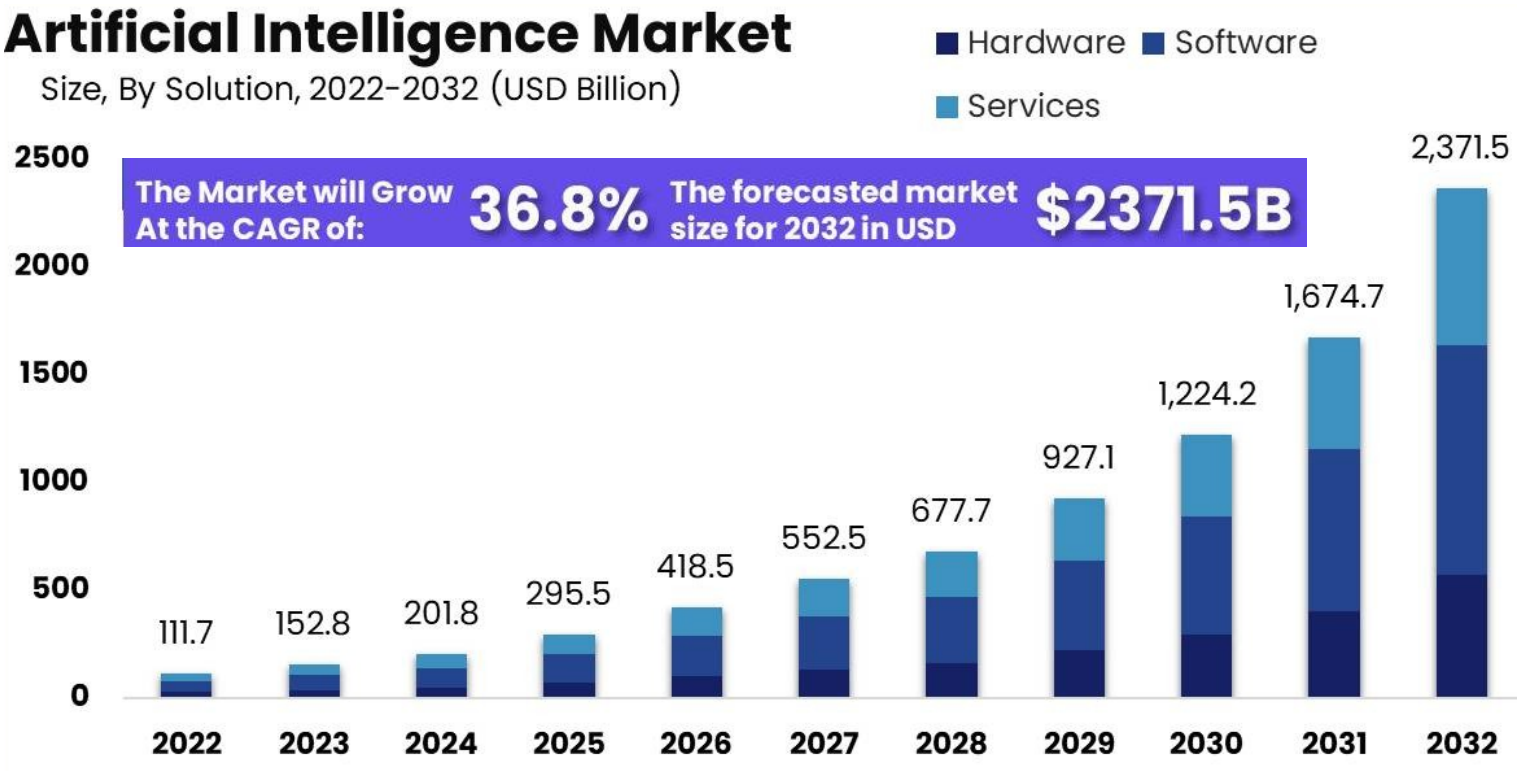
AI Hardware Technologies for Military Applications

- Hardware efficiency and flexibility
  - CPUs: A general-purpose processing unit.
  - GPUs : A massively parallel processing unit.
  - FPGAs : Reconfigurable interconnections of basic components.
  - ASICs : Customization and high performance.



On-device AI is the future! Now is the future!

- The AI service has moved toward the local devices, such as IoTs, vehicles *etc.*, which tends to be personal usage habits and experience.

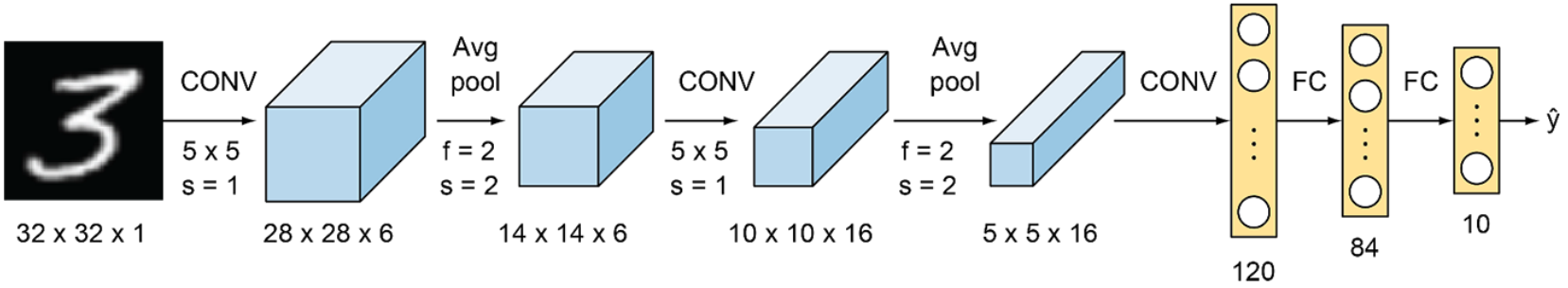


Source: MarketResearch

Fundamental of DNN Hardware Design

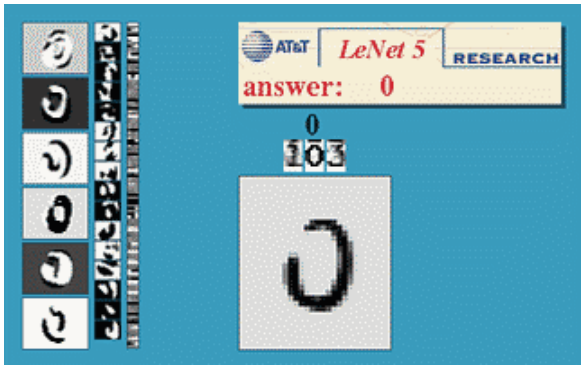
Convolutional Neural Network (CNN)

- Convolutional neural network (CNN) often used for image recognition
- When an image is input to CNN, the output result will be the recognition result of this image



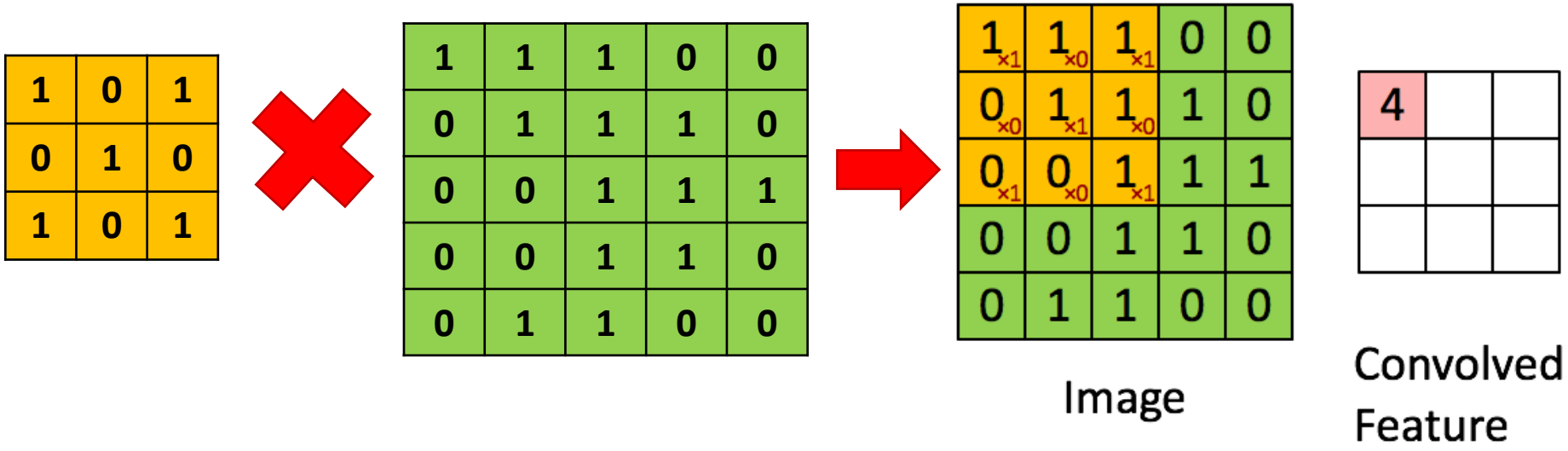
Required Operations in CNN

- Convolution
- Pooling
- Multiply-accumulation



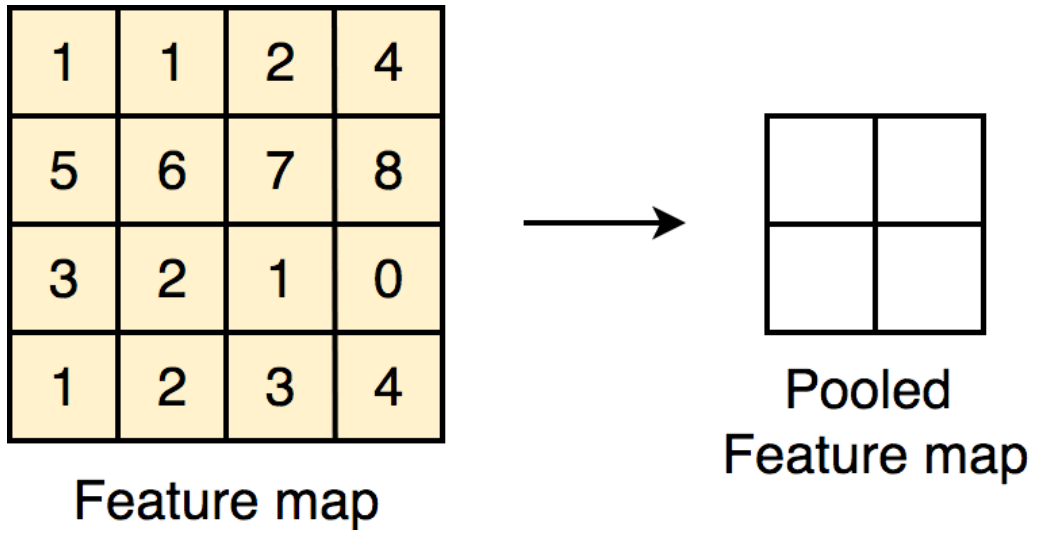
Convolution Operation

- Convolution layer is used to extract the proper feature of the target image
- Convolutional operation is to multiply input data with kernel
- The kernel is the weight of convolution layer



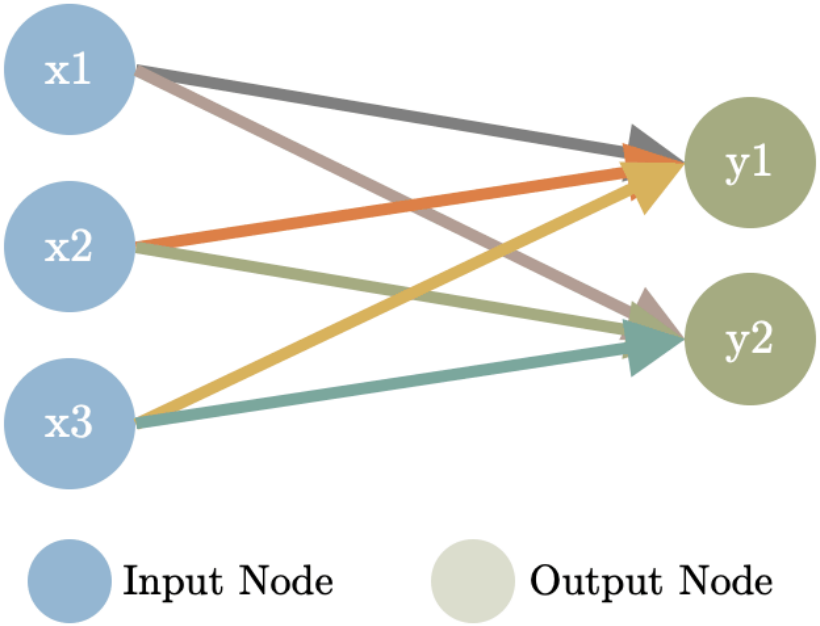
Pooling Operation

- The main goal of the pooling operation is to extract the most representative features of the sentence using a function that aggregates the output of each filter
- Categories of pooling operation
  - Max-pooling
  - Average-pooling



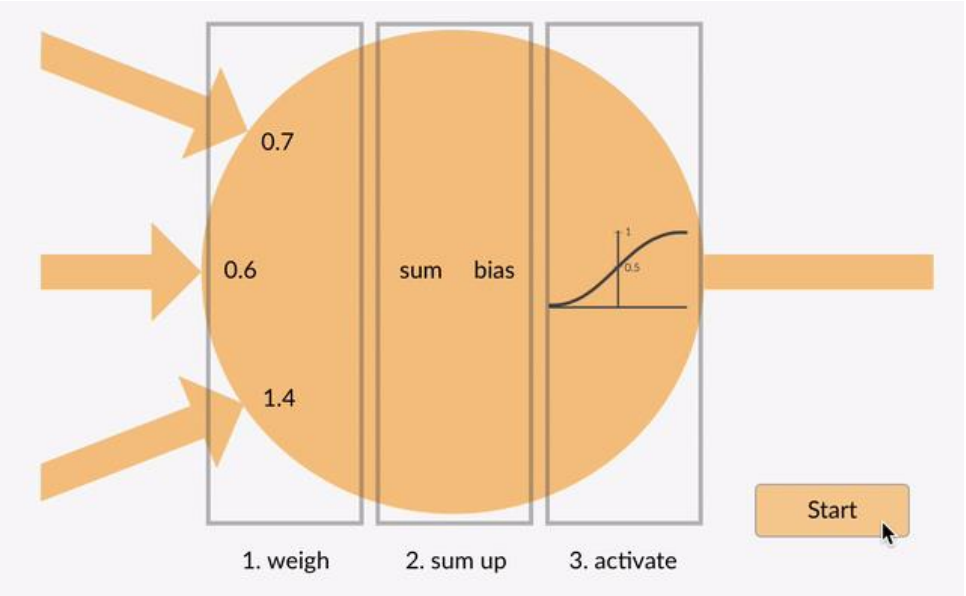
Multiply-accumulation Operation

- The type of operation is really like the convolution operation.
- The only different is that the involved weights can not be reused.
- The dense layer (or fully connected layer) is used to consider the extracted feature interpretively and further perform the classification.

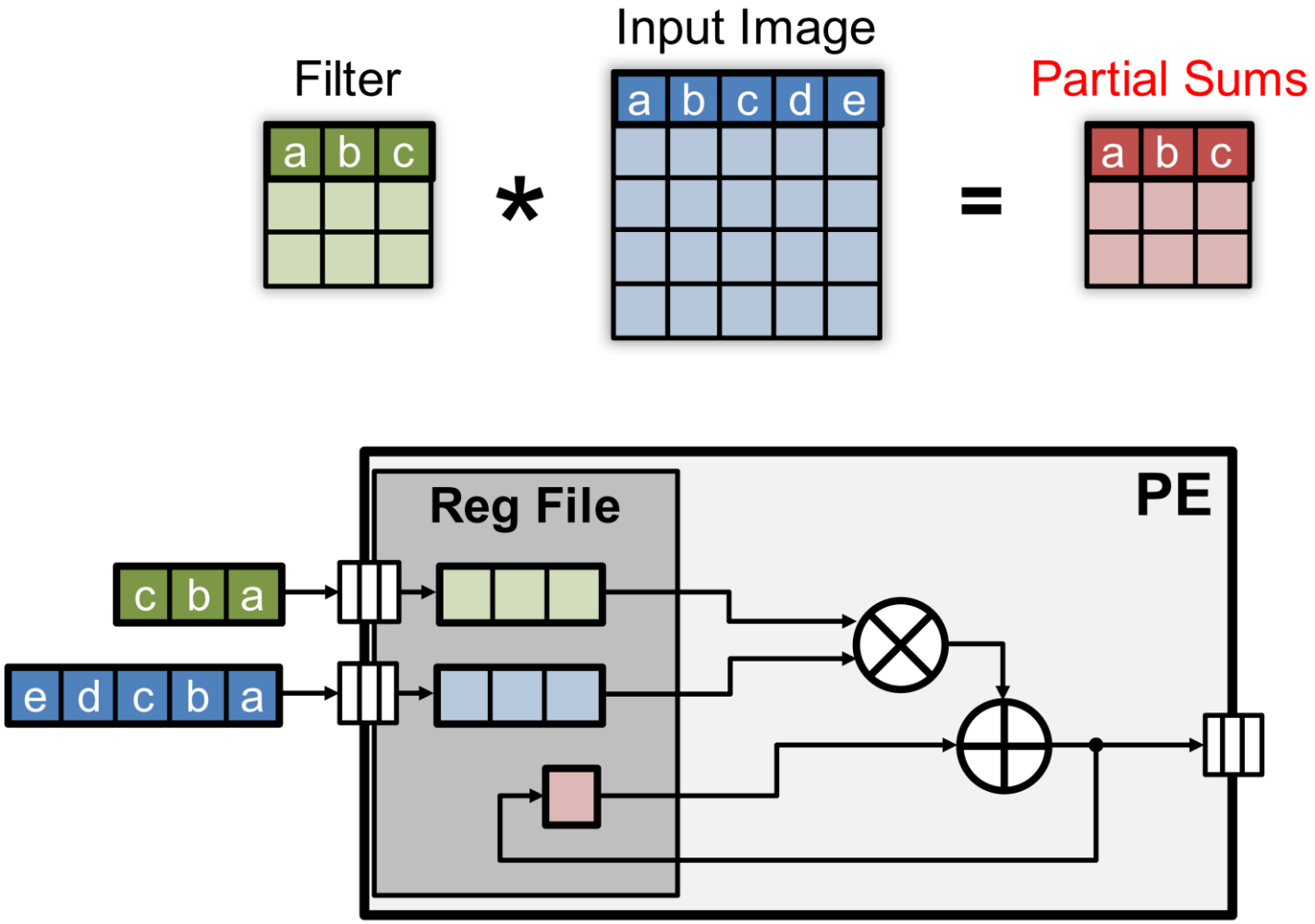


Non-linear Activation Function

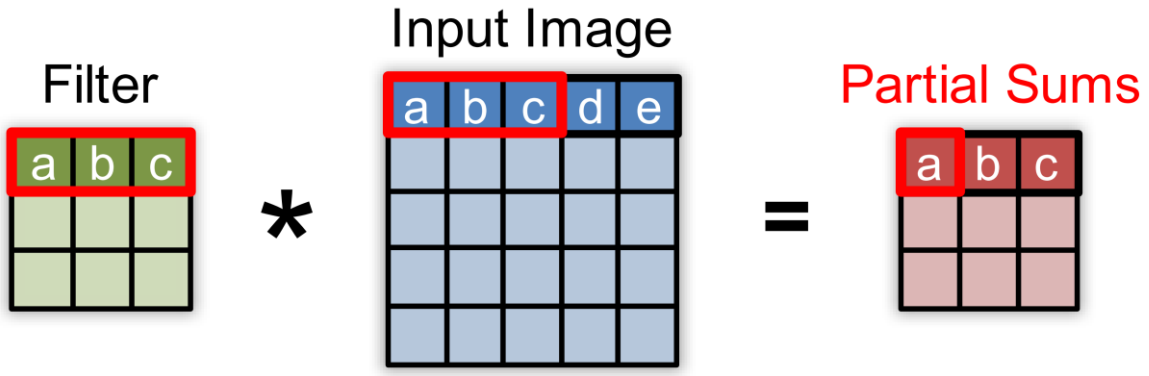
- The problem in real world is usually not a deterministic problem (i.e., linear problem).
- After the multiply-accumulate operation, a non-linear activation function is usually involved.
  - One of the design challenges in AI hardware design.

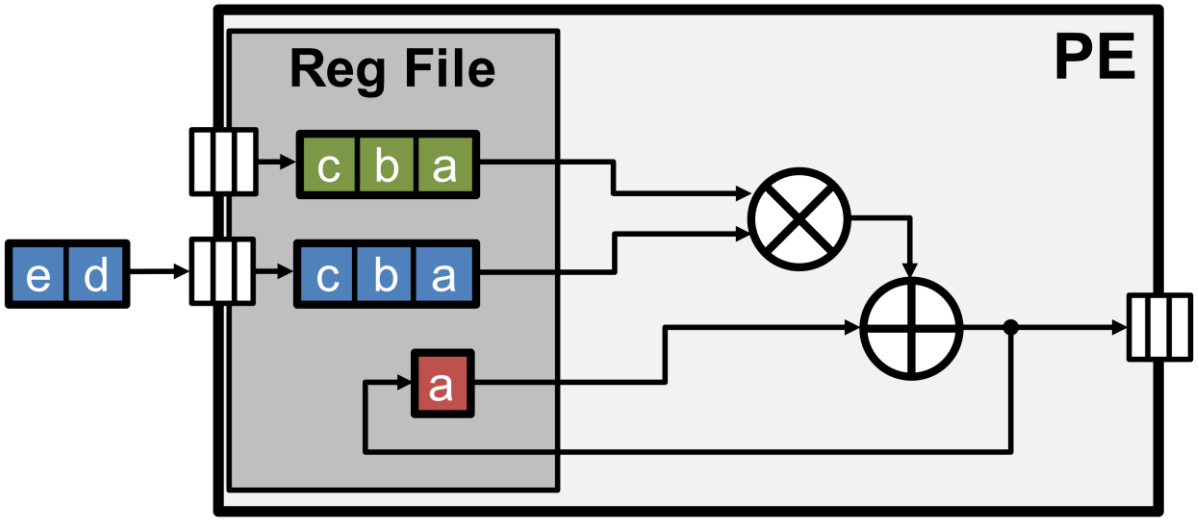


1D Row Convolution in PE (1/5)

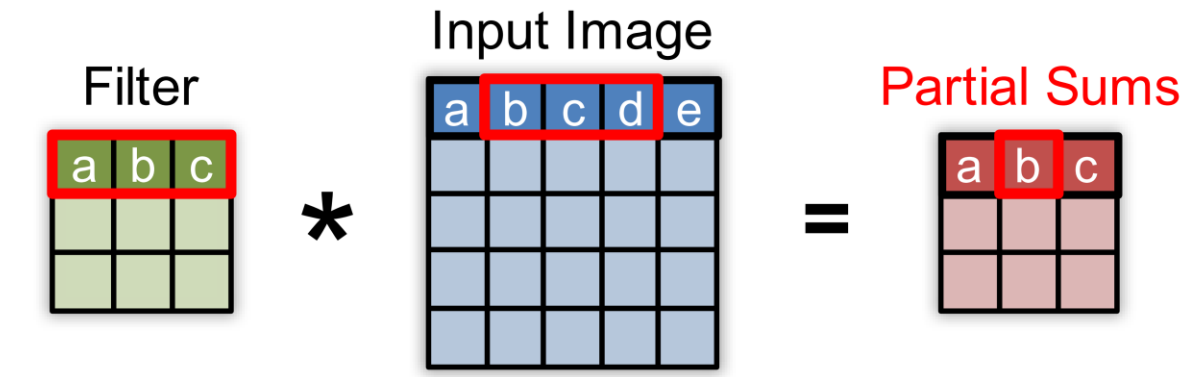


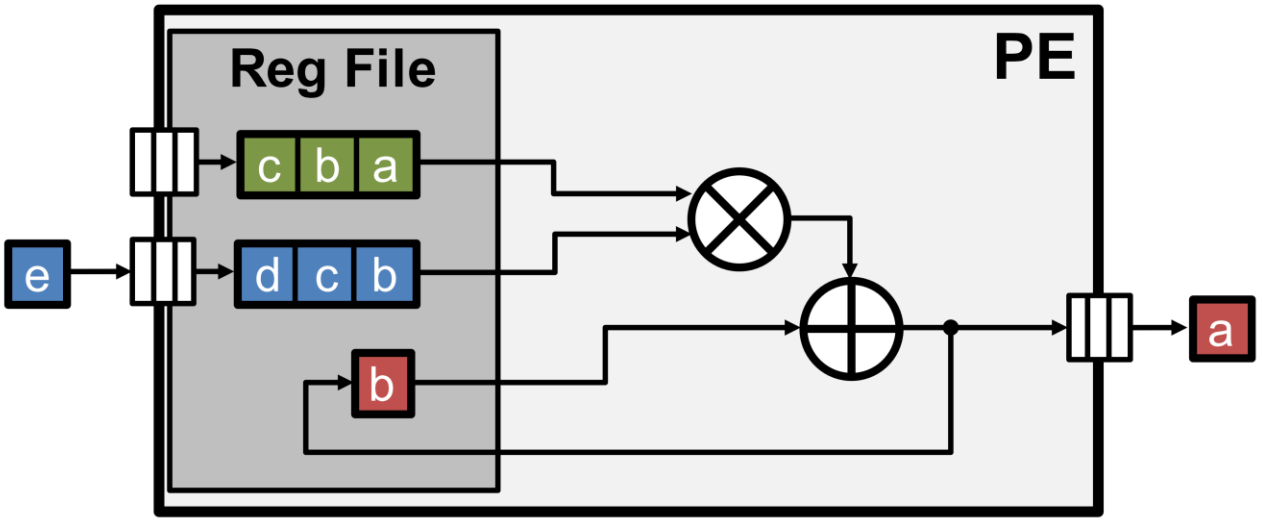
1D Row Convolution in PE (2/5)



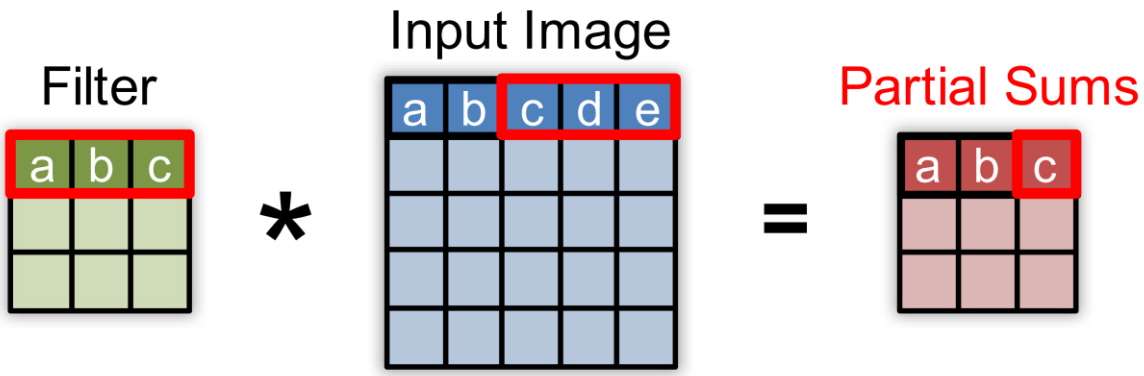


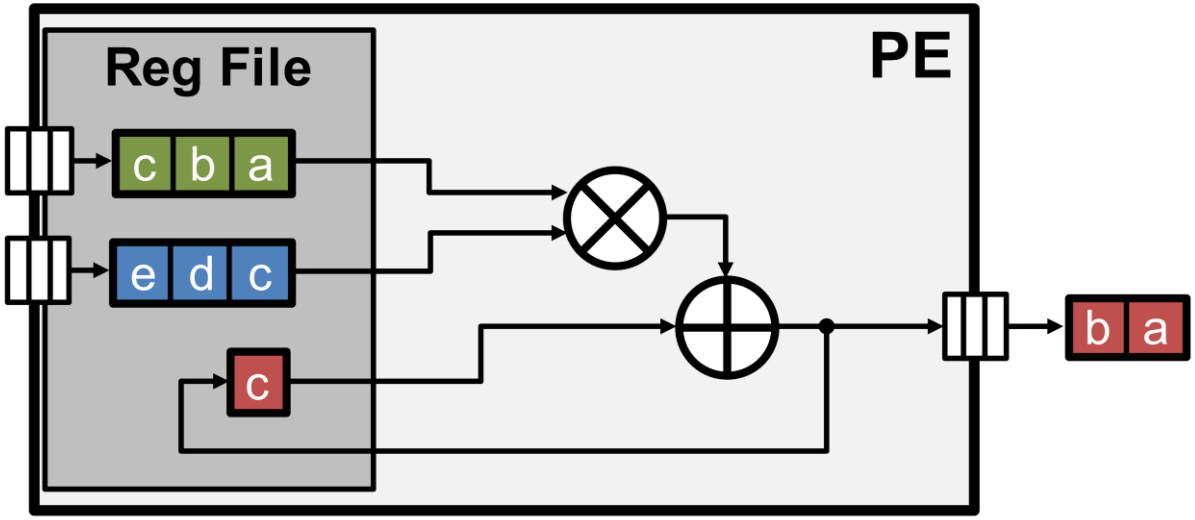
1D Row Convolution in PE (3/5)





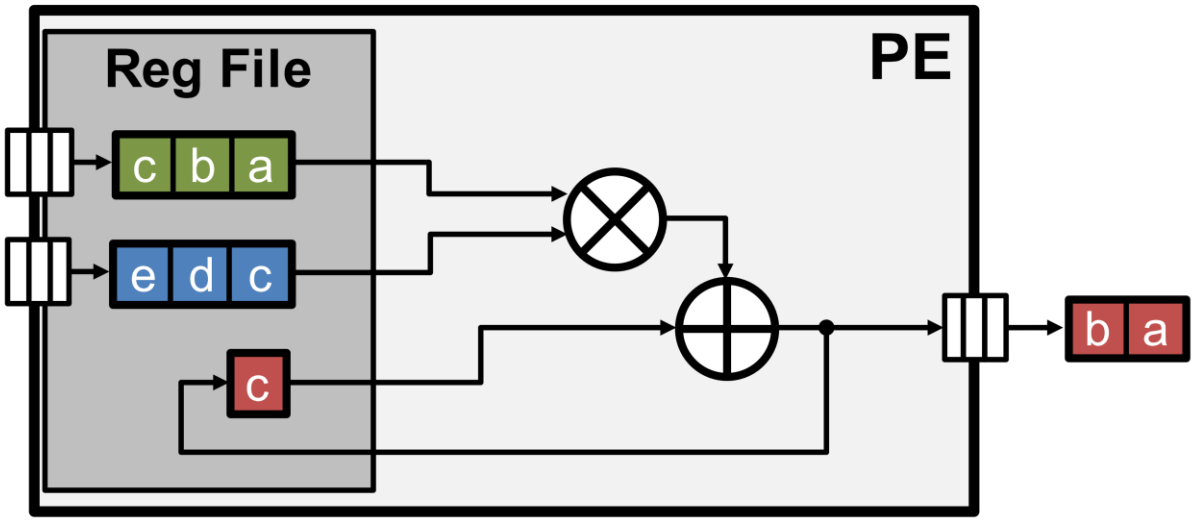
1D Row Convolution in PE (4/5)



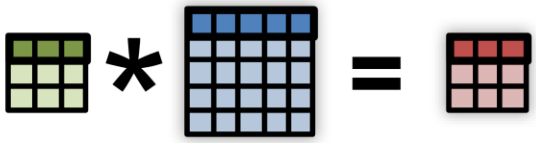


1D Row Convolution in PE (5/5)

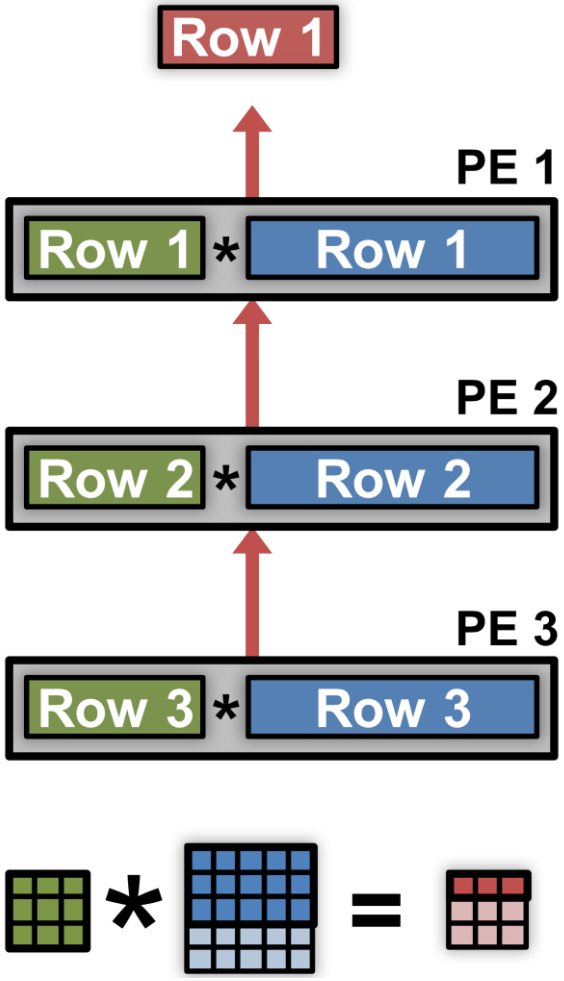
- Maximize row **convolutional reuse** in RF
  - Keep a **filter** row and **image** sliding window in RF
- Maximize row **psum accumulation** in RF



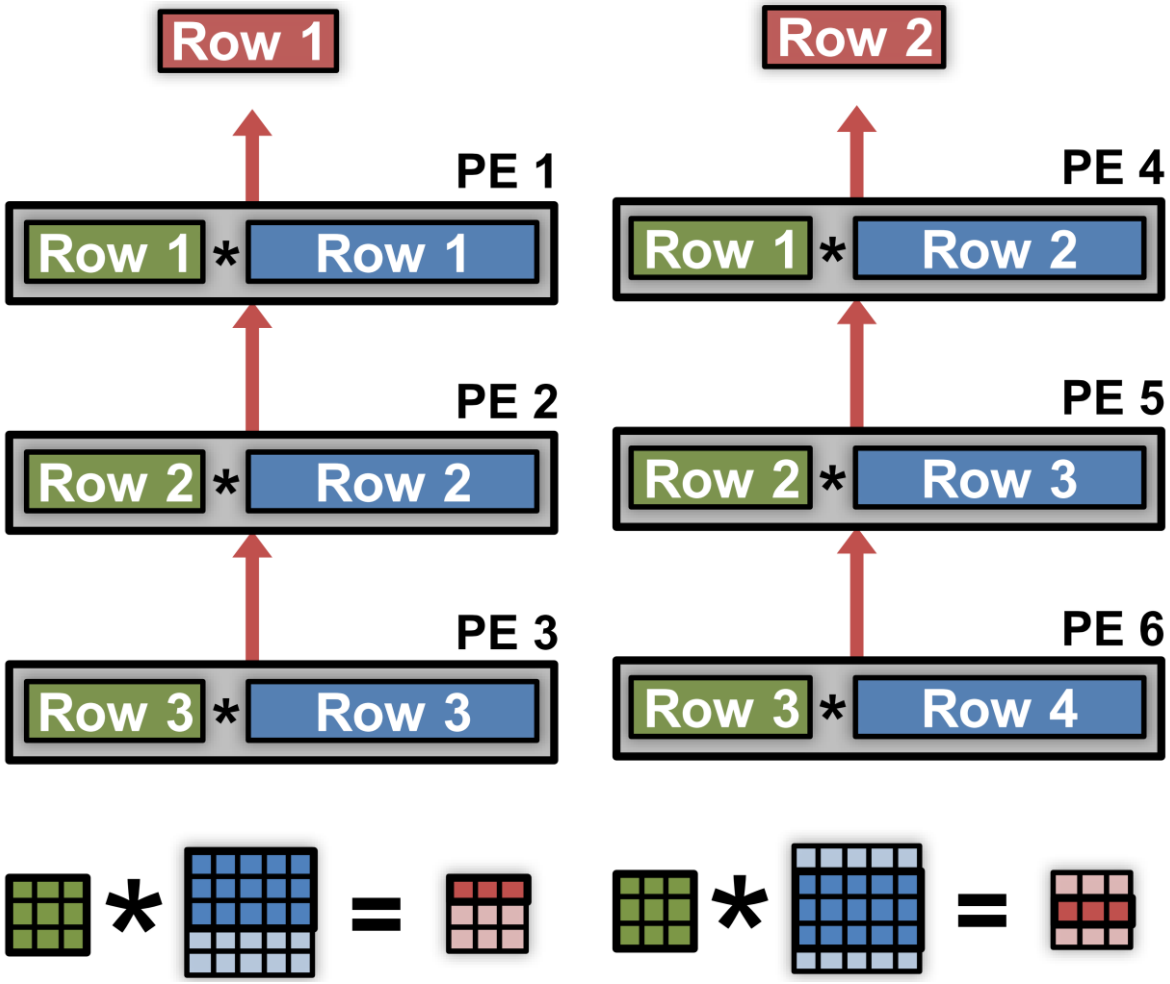
2D Row Convolution in PE (1/4)



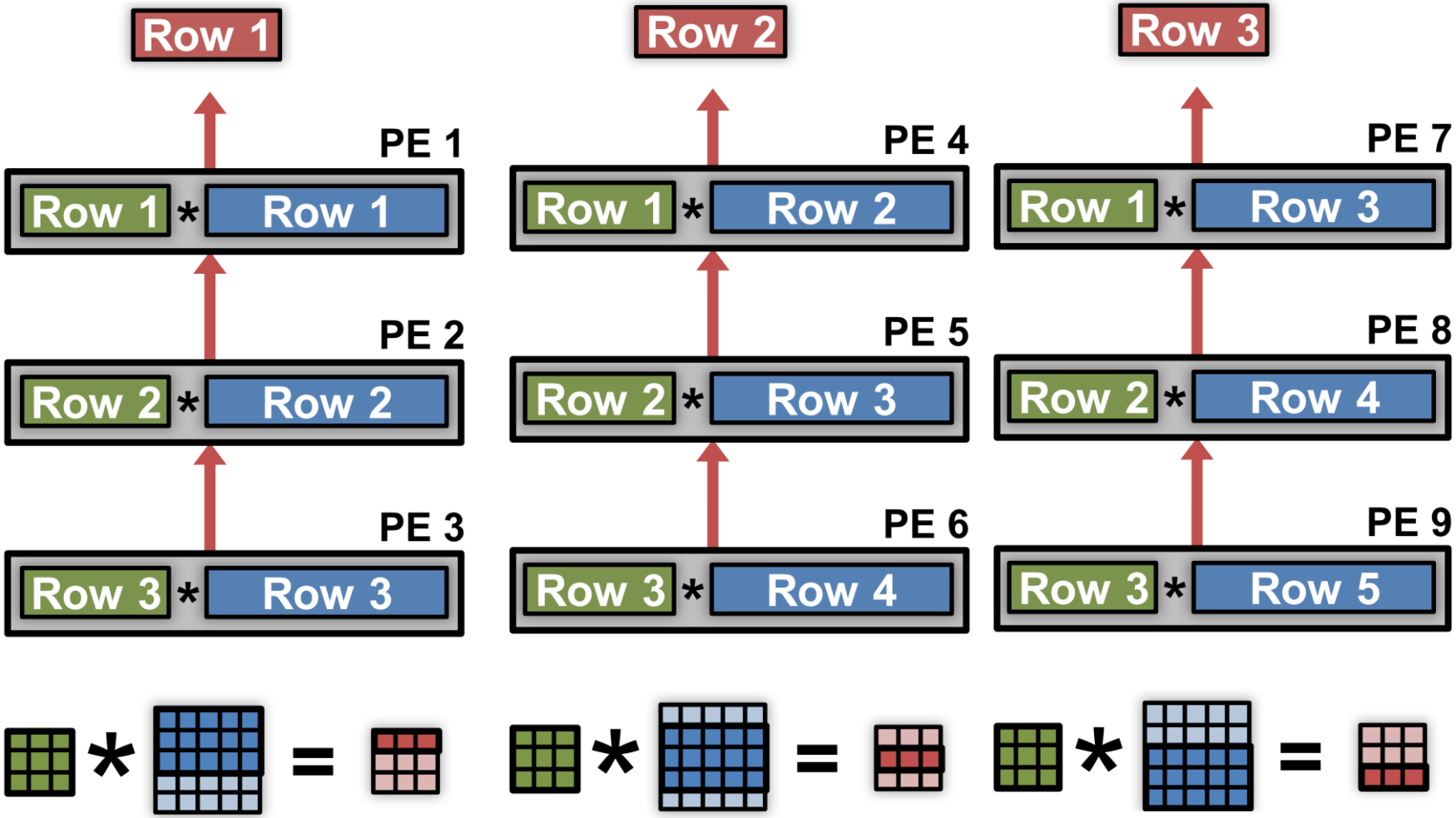
2D Row Convolution in PE (2/4)



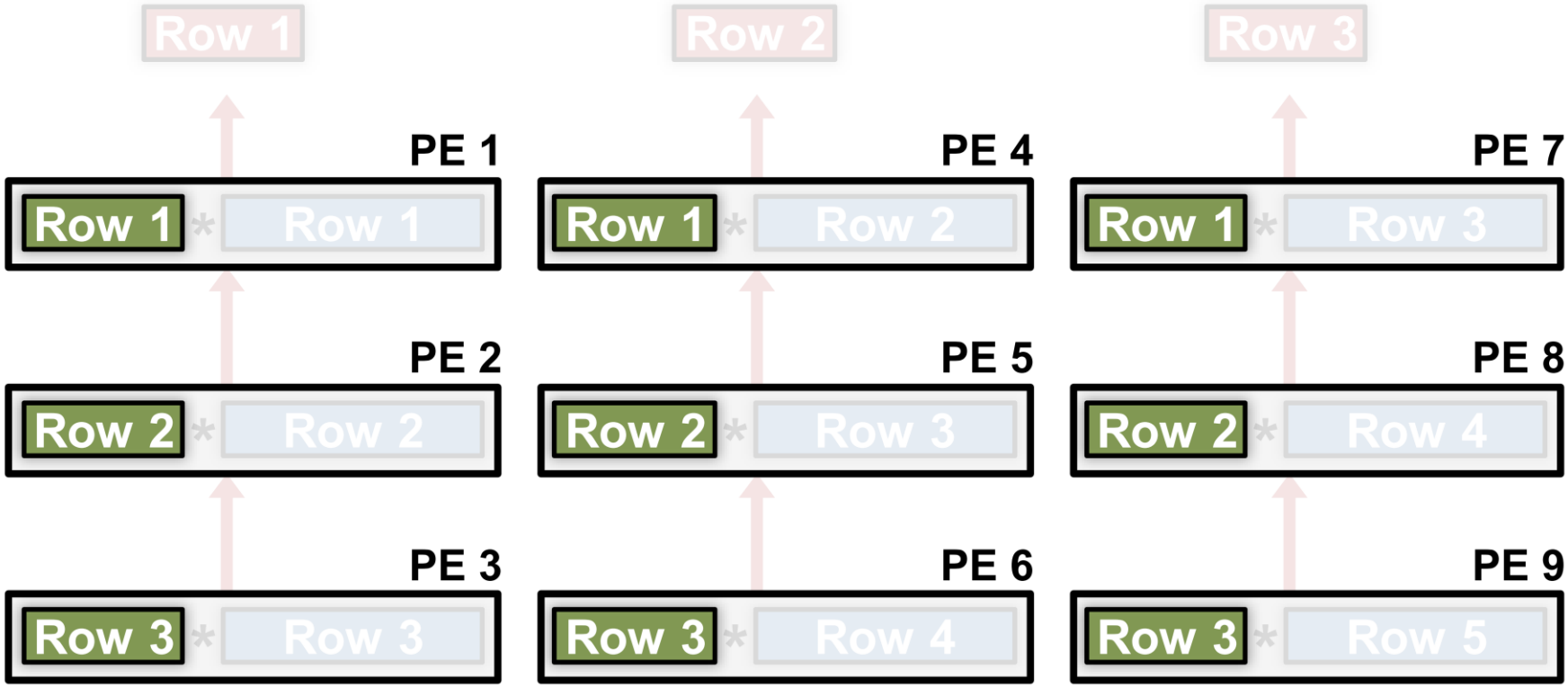
2D Row Convolution in PE (3/4)



2D Row Convolution in PE (4/4)



Convolutional Reuse Maximized



Filter rows are reused across PEs **horizontally**

Convolutional Reuse Maximized

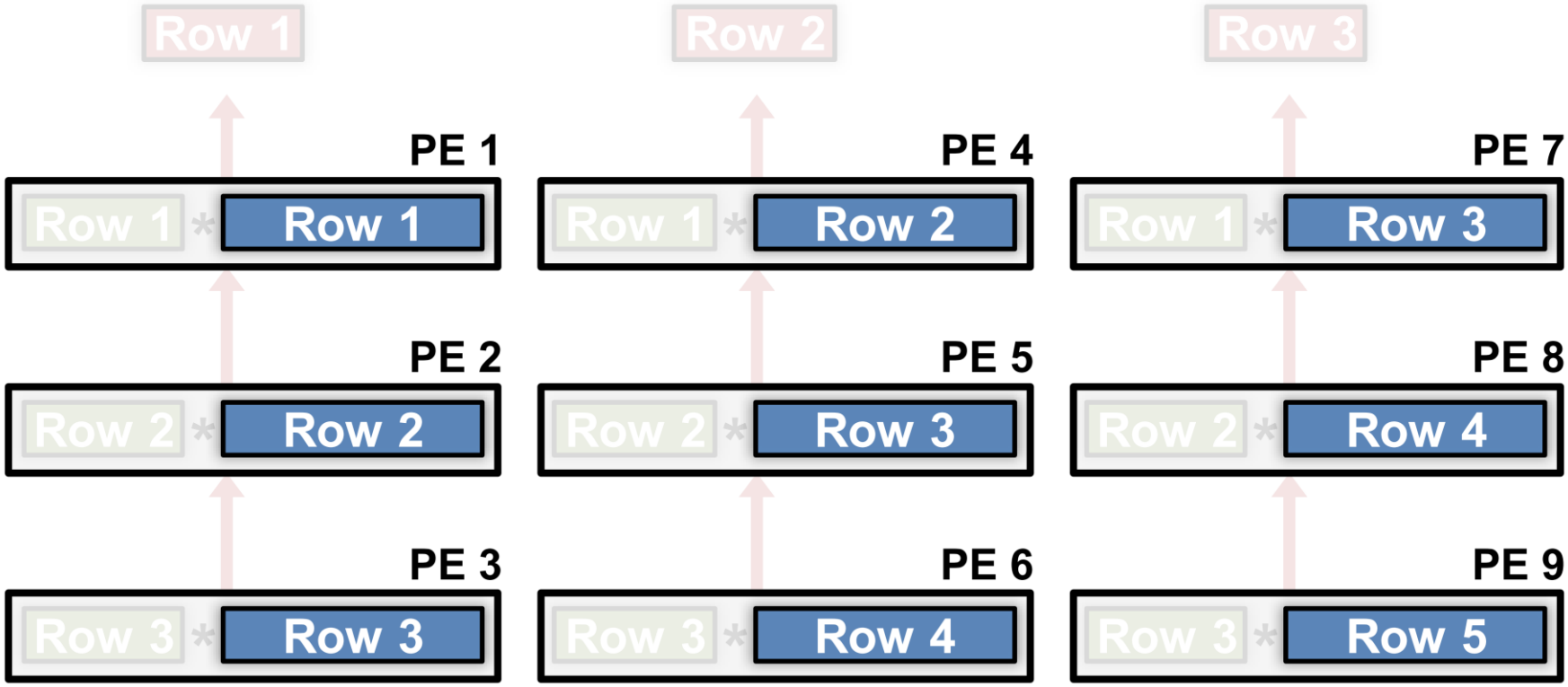
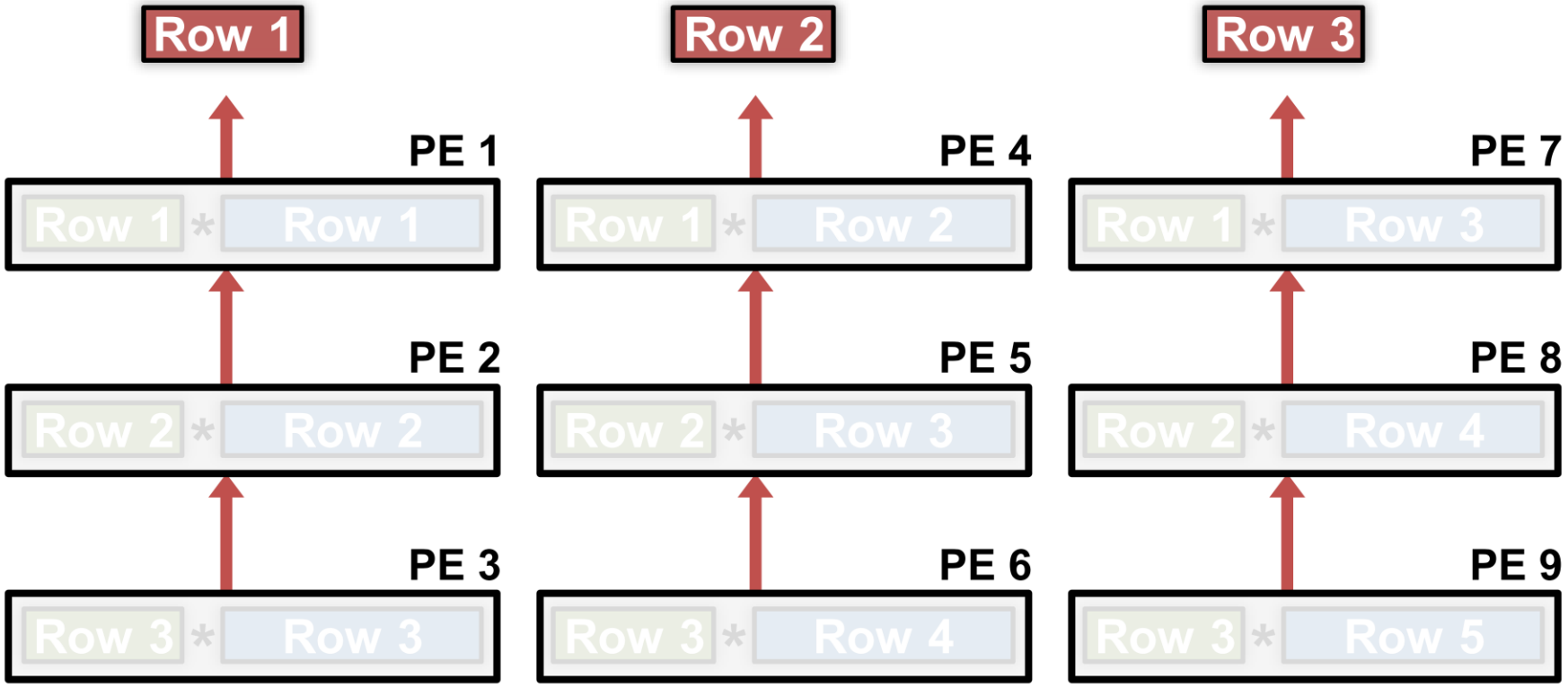


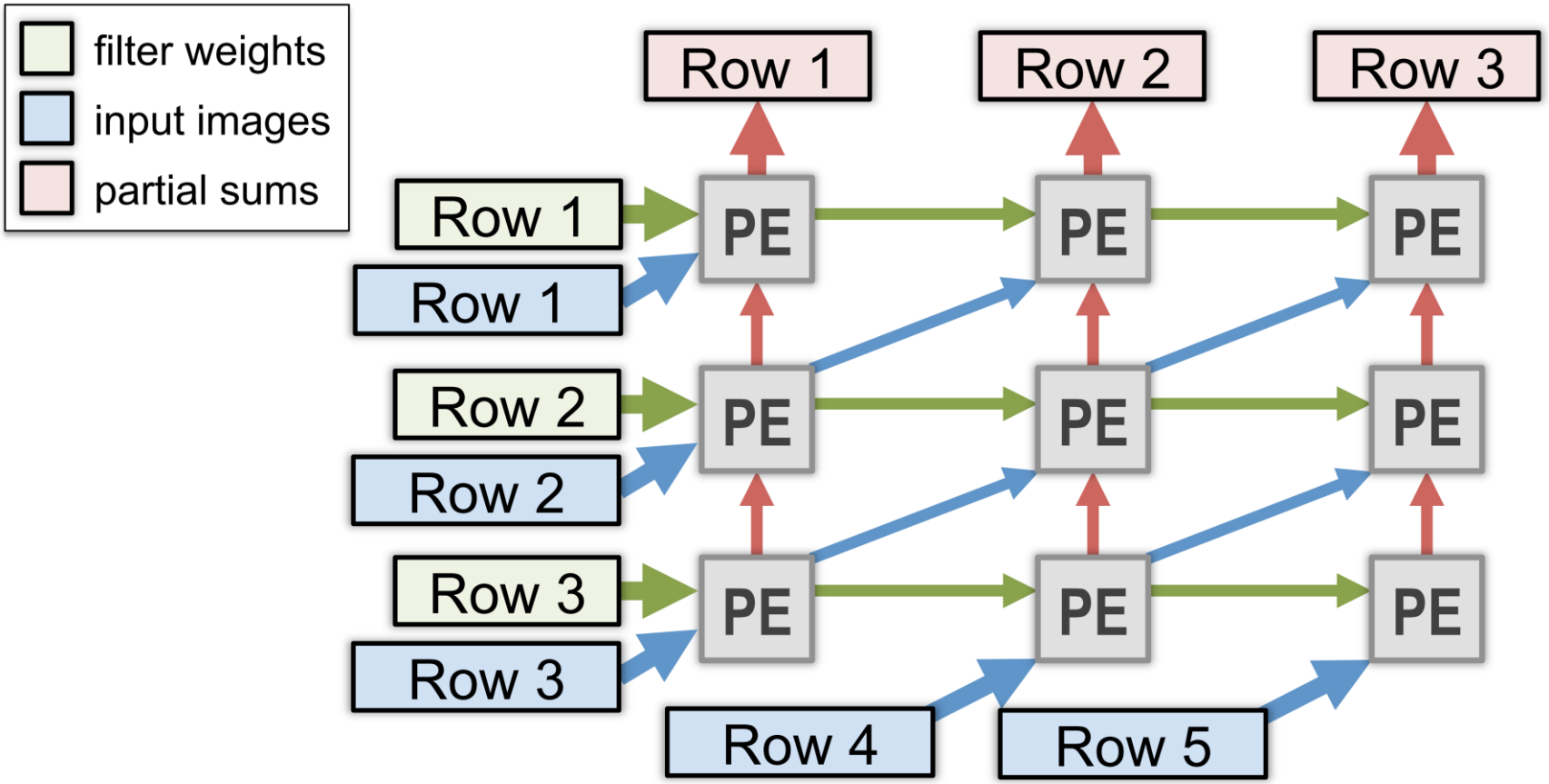
Image rows are reused across PEs **diagonally**

Maximize 2D Accumulation in PE Array



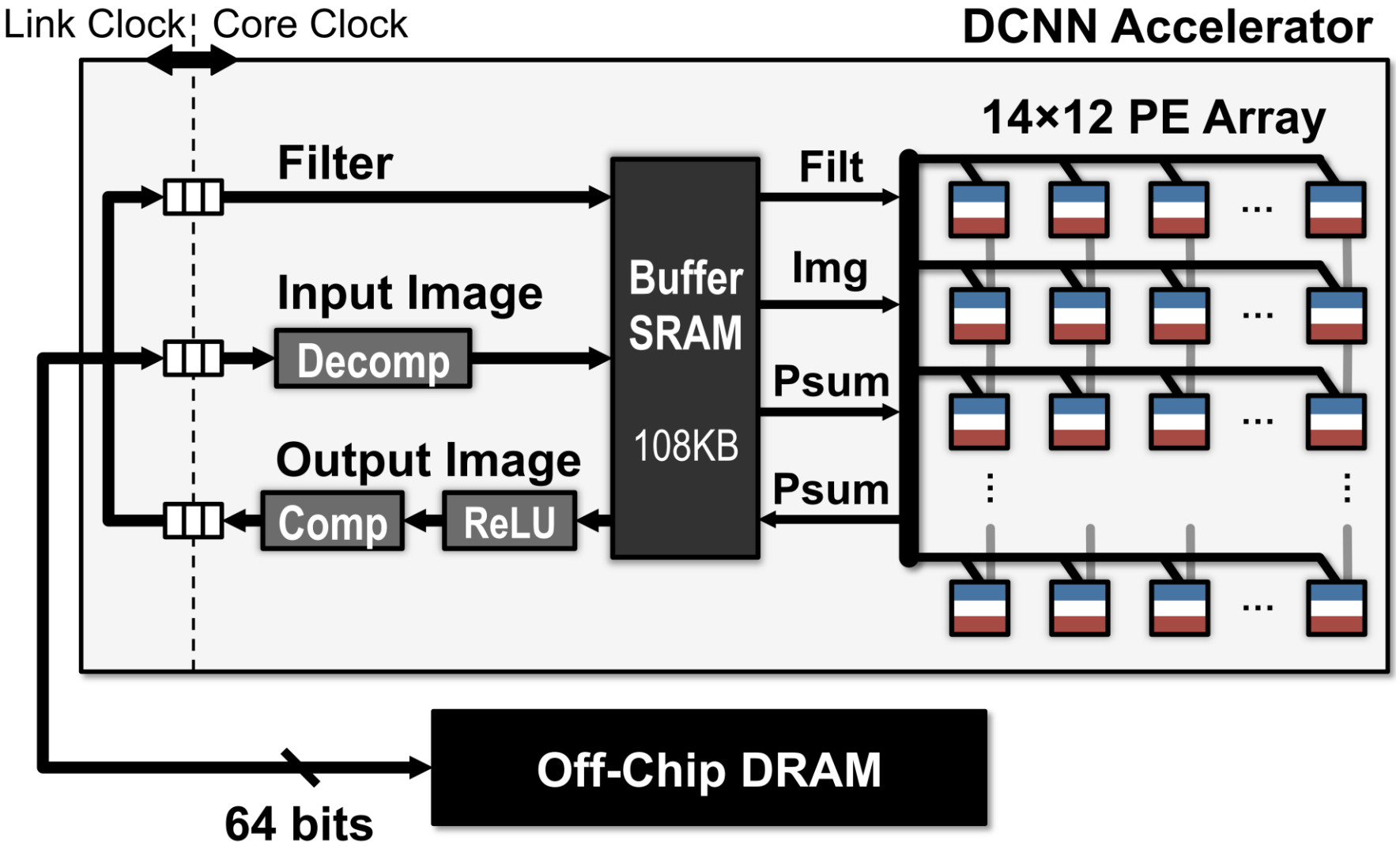
Partial sums accumulate across PEs **vertically**

Convolutional Reuse within PE Array

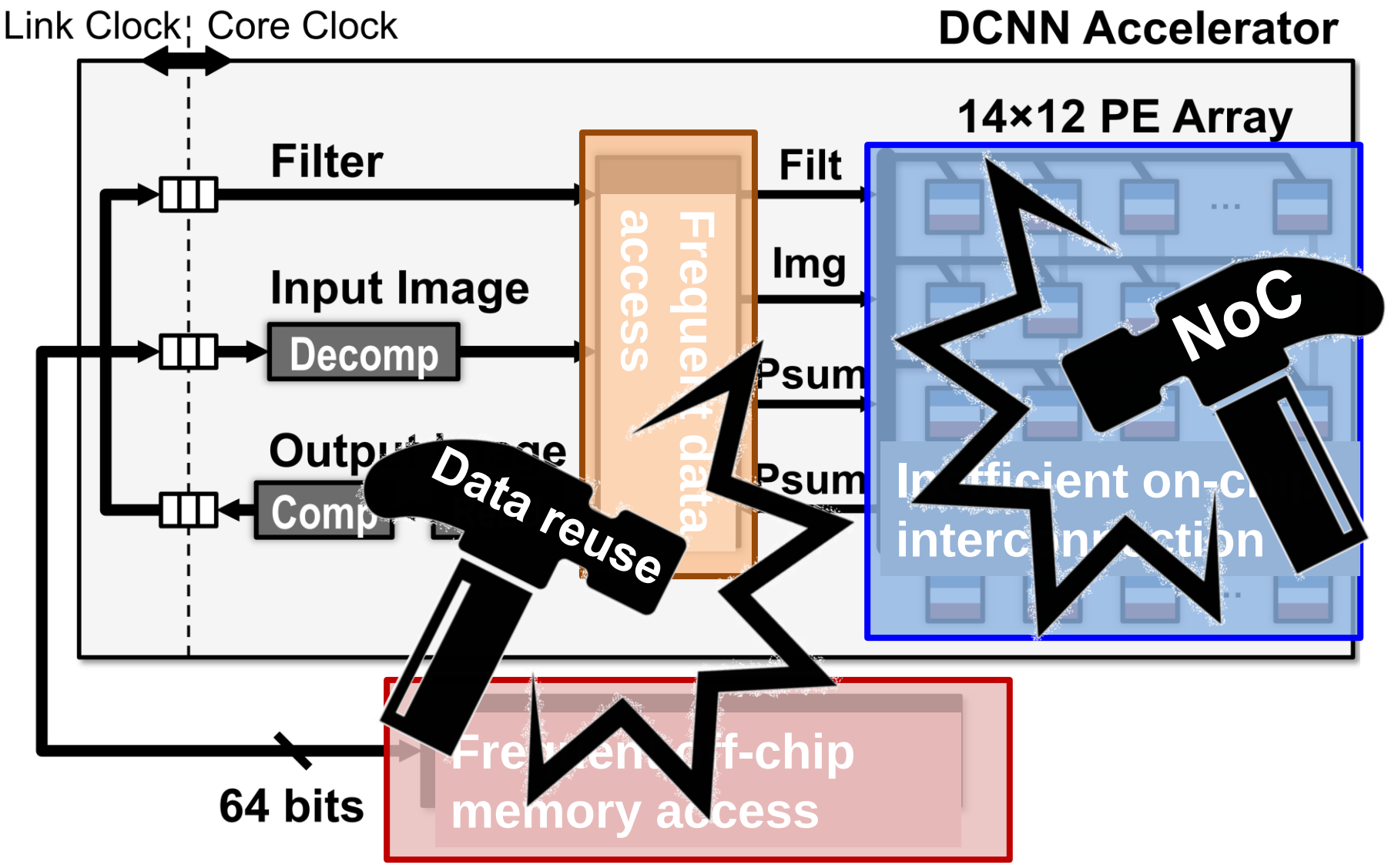


Mapping rows from **multiple channels** and/or **multiple filter/images** to each PE results in even more **reuse**

The DCNN Accelerator Architecture

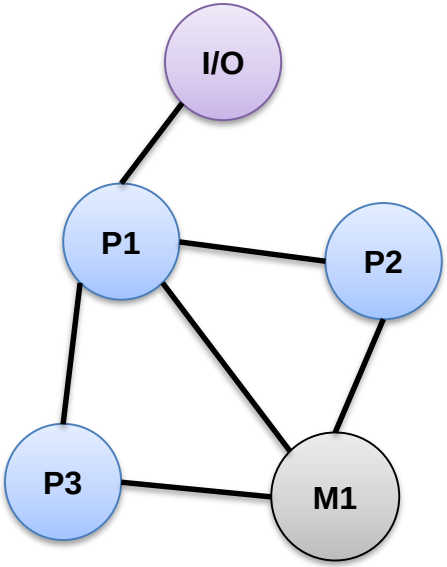


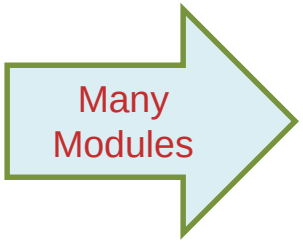
Design Challenges

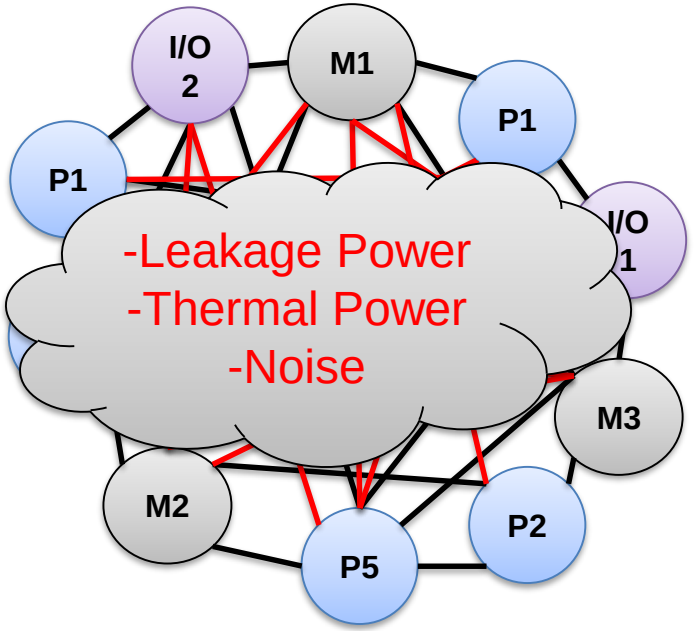


Scalable and Flexible On-chip Interconnection – NoC Interconnection

On-chip Interconnection Types

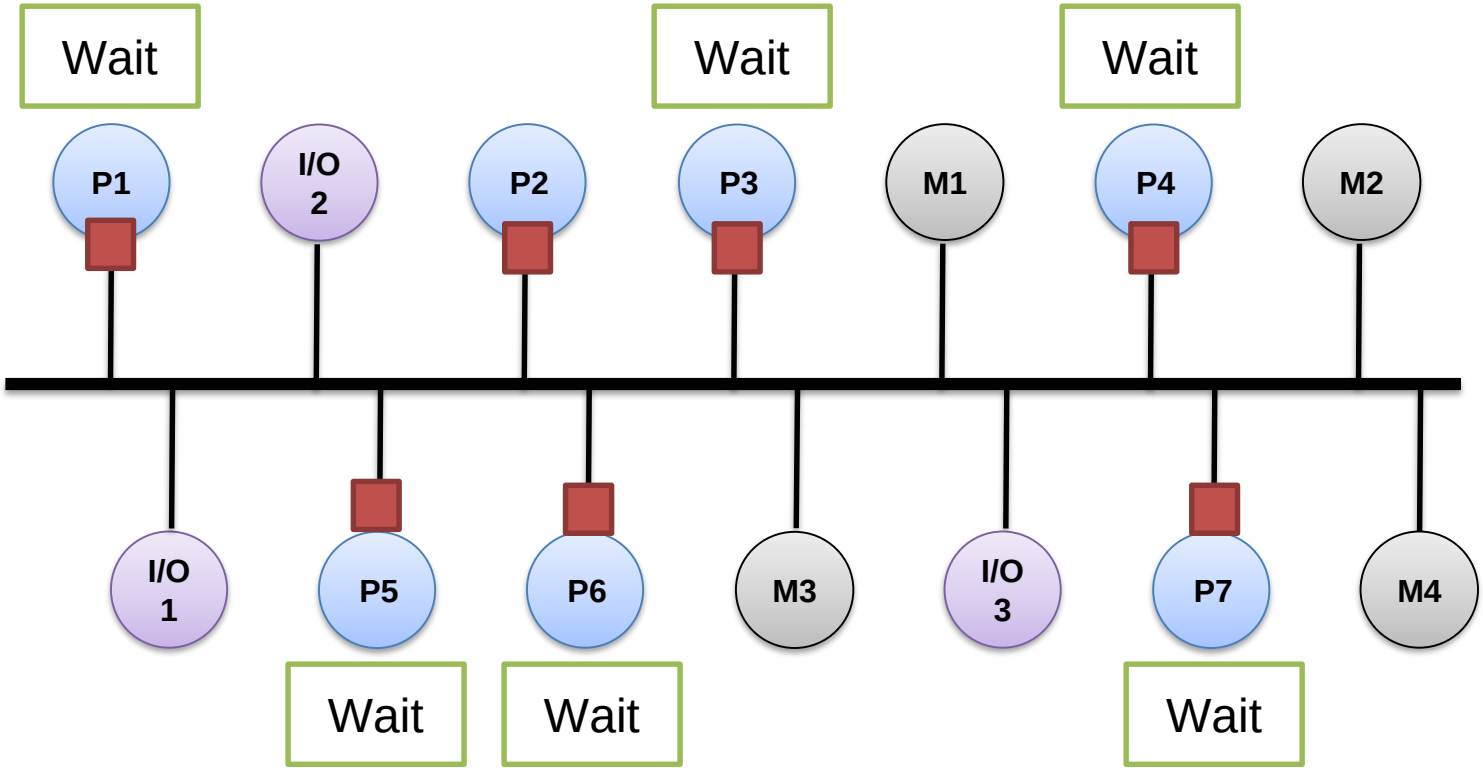






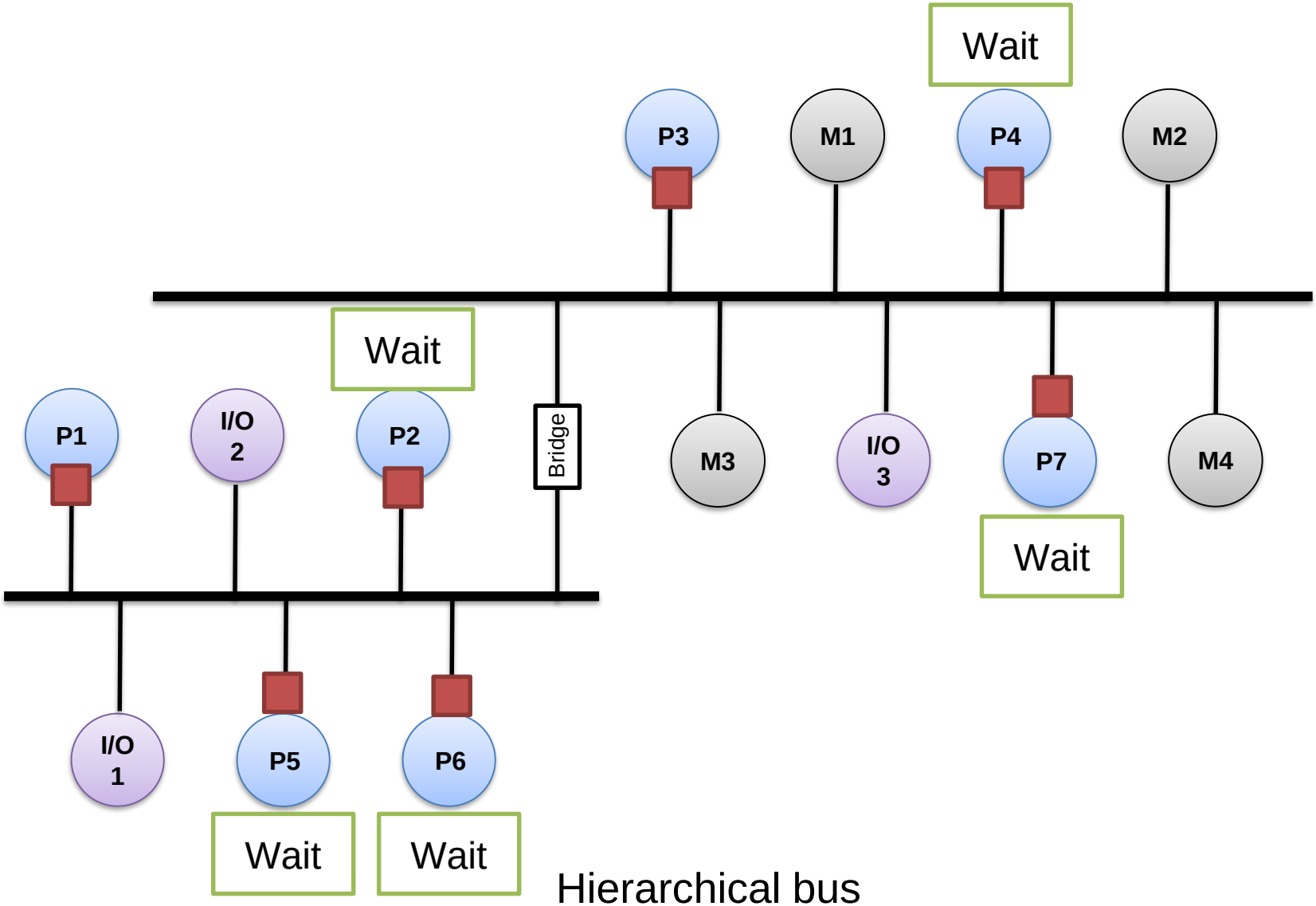
Point-to-Point

On-chip Interconnection Types



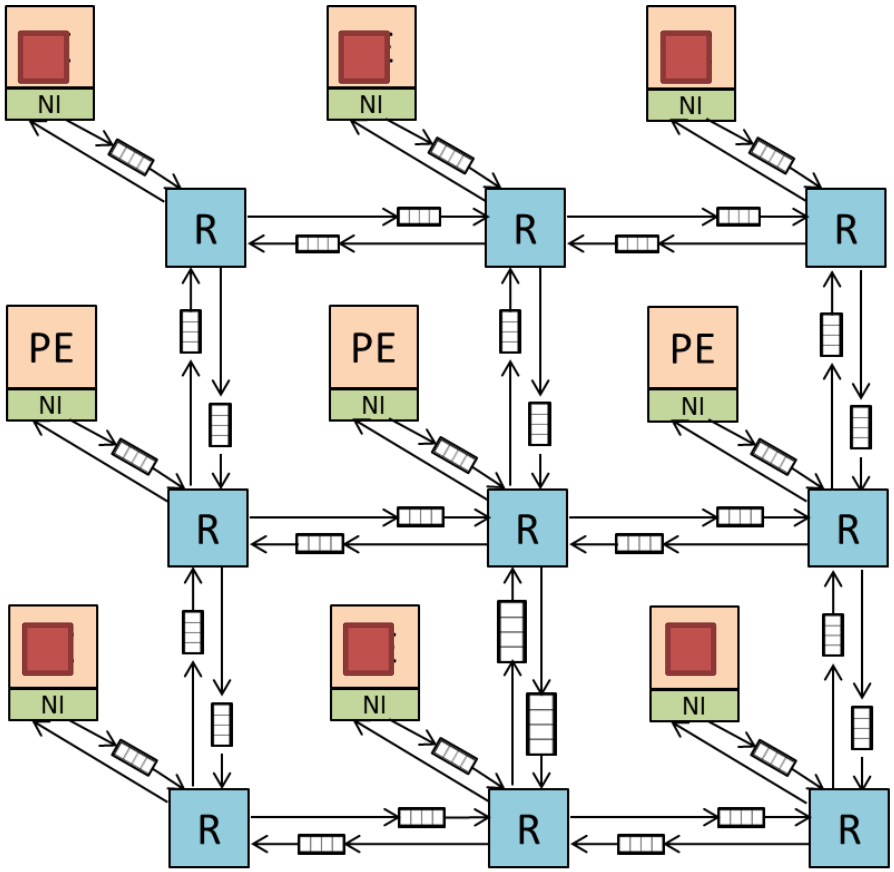
Shared bus

On-chip Interconnection Types



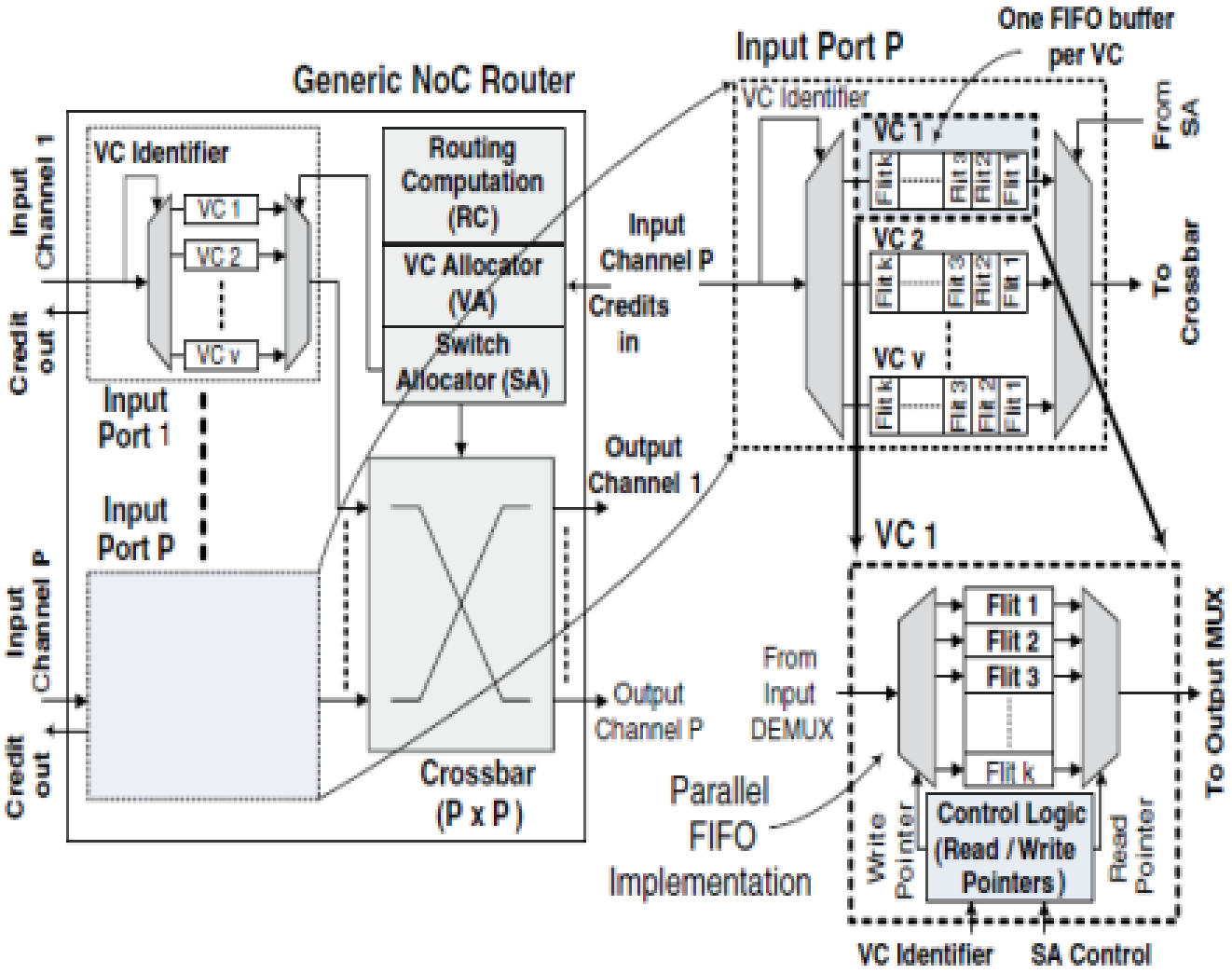
Hierarchical bus

On-chip Interconnection Types



Network-on-Chip -> our main topic in this talk

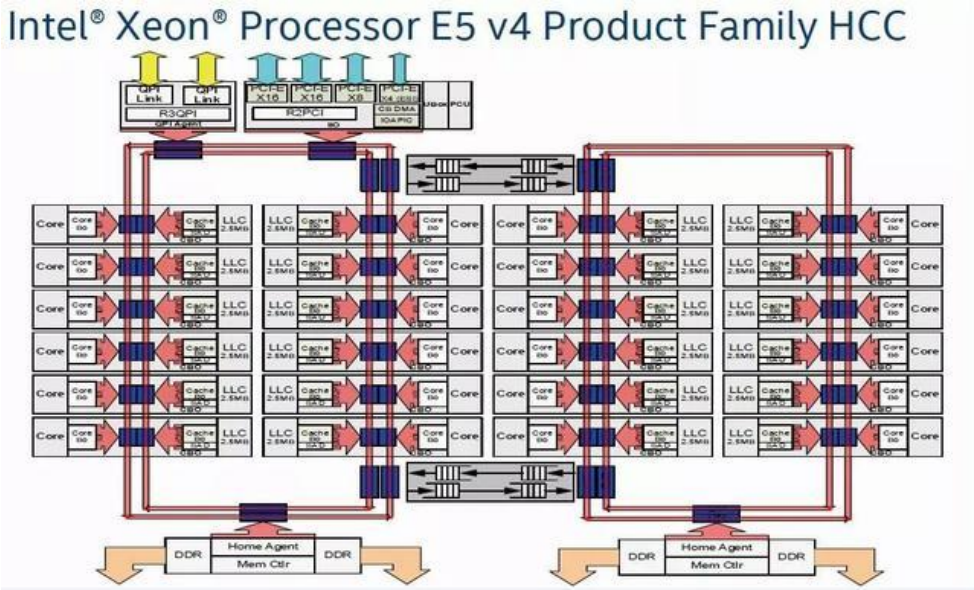
Generic NoC Router Architecture

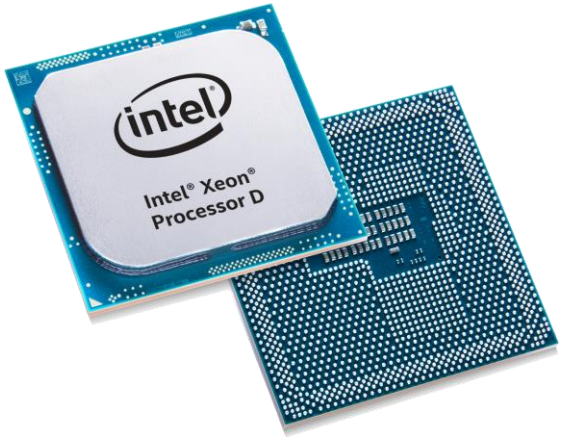


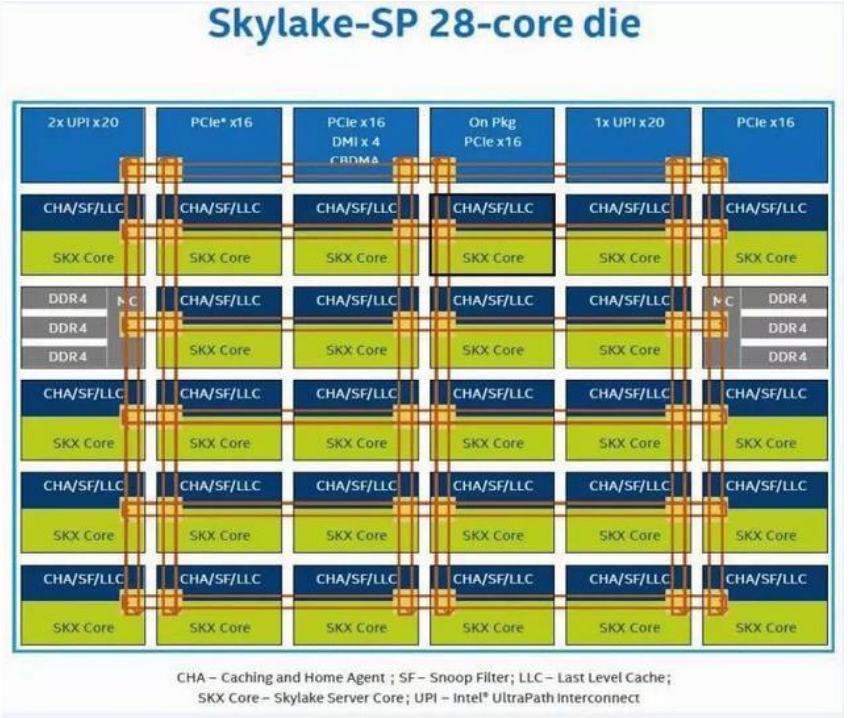
Commercial Product 1:
Xilinx Versal Series FPGA, 2018

- Adaptive Compute Acceleration Platform (ACAP)
  - New name of “programmable logic”
  - Connect each SoC component via NoC interconnection
  - Target the 5G and AI application

Commercial Product 2:
Intel Skylake (server) Architecture, 2018

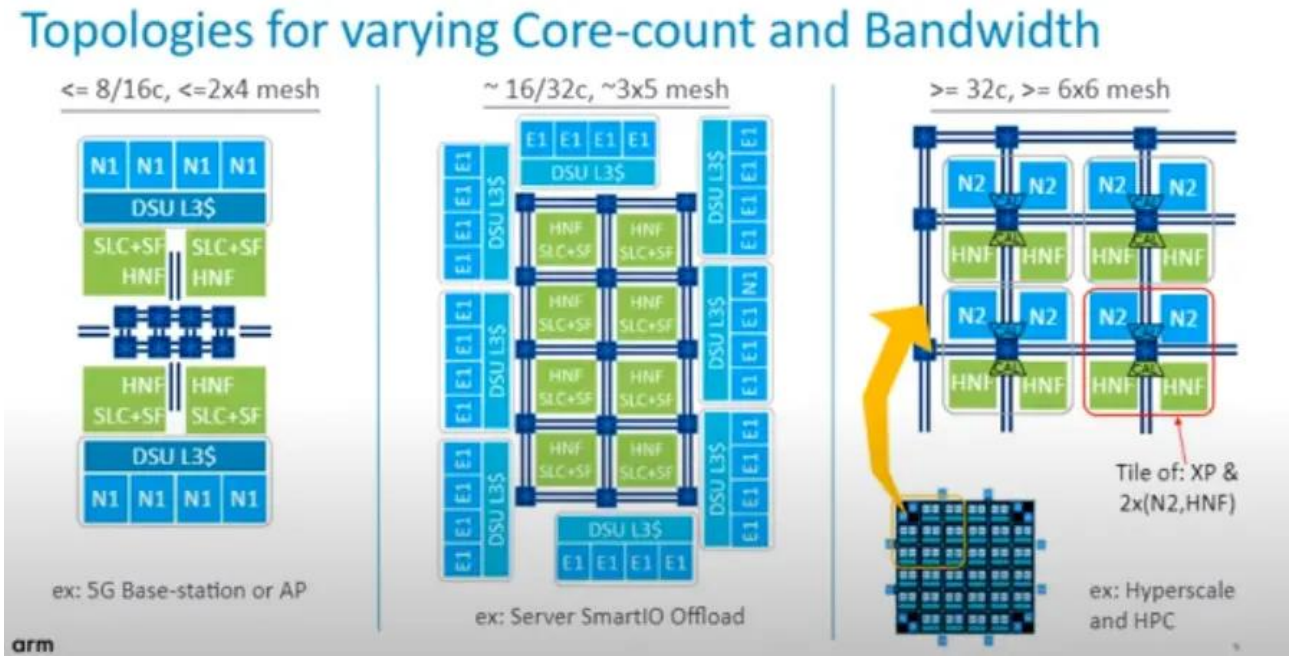






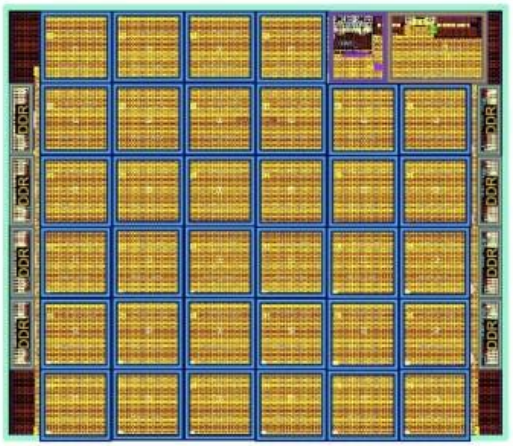
Commercial Product 3: ARM CMN-700 Mesh Network, 2021

- ARM CoreLink NI-700,
  - A new flexible packetized network-on-chip interconnect for both high-bandwidth accelerators
  - Scalable and highly configurable network-on-chip (NoC)
  - all the AMBA transactions are converted to a packetized format
    - Reduce the wire count by 30% on average

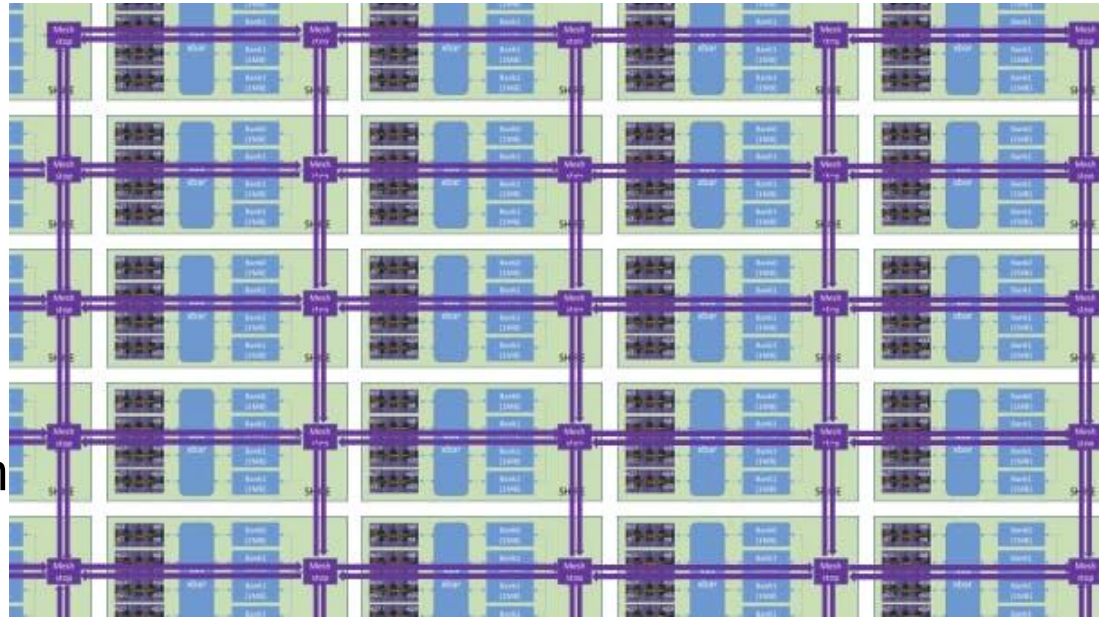


Commercial Product 4: Esperanto ET-SoC-1, 2021

- Esperanto ET-SoC-1
  - 1093 RISC-V AI Accelerator
  - The first AI processor in Esperanto Inc.

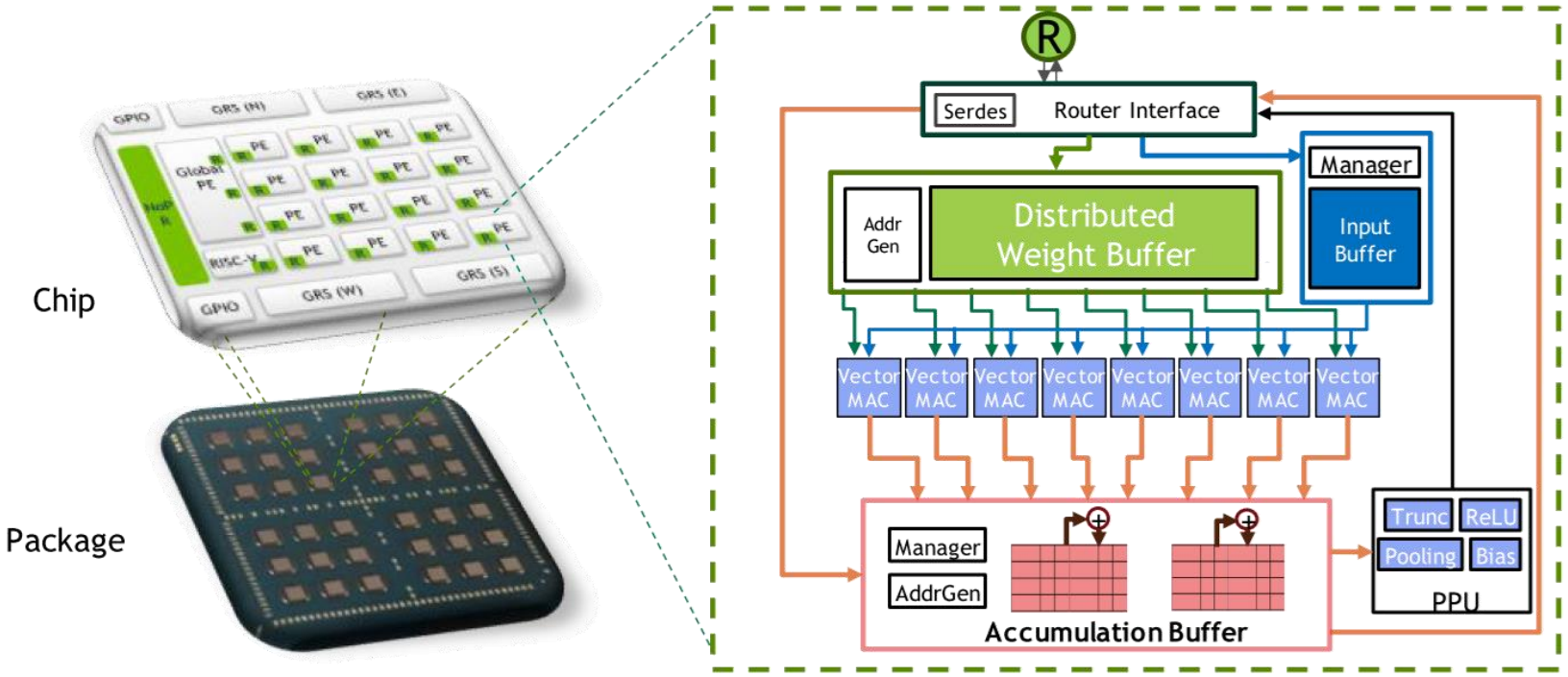


- TSMC 7nm
- 4 ET-Maxion (superscalar out-of-order core)
- 1089 ET-Minion (in-order multithreaded core)
- 32GB DRAM
- 137GB/sec memory bandwidth
- 256-bit wide interface



Commercial Product 5: NVIDIA Muti-Chip-Module (MCM), 2019

- Multi-Chip-Module (MCM)
  - Hierarchical NoC interconnection
    - 4x5 mesh topology connects 16 PEs, 1 global PE, and 1 RISC-V
    - 6x6 mesh topology connects 36 chips in package

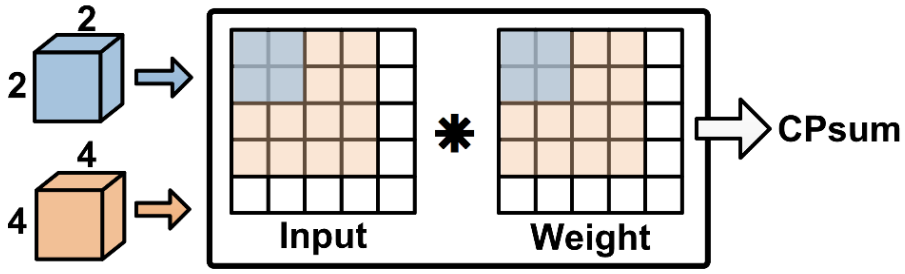


Reconfigurable Compact NN Processing – Kernel-size applicable DNNoC design

Design Challenge of DNN Accelerator: Various kernel size

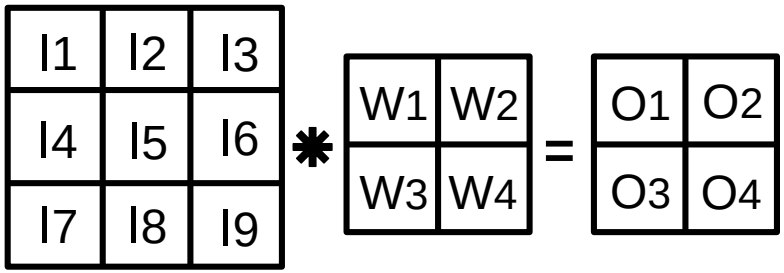
- The convolutional kernel sizes are usually not fixed in the DNN model.
  - Worst-case design consideration
- The register size of processing element (PE) is usually based on the largest kernel size in the target model.
  - Low utilization of PE computational capability.
  - Cannot process the operation.

DNN model	Kernel size/shape
AlexNet	3x3, 5x5, 11x11
GoogLeNet	1x1, 3x3, 5x5, 7x7
DeepSpeech2	21x11, 41x11



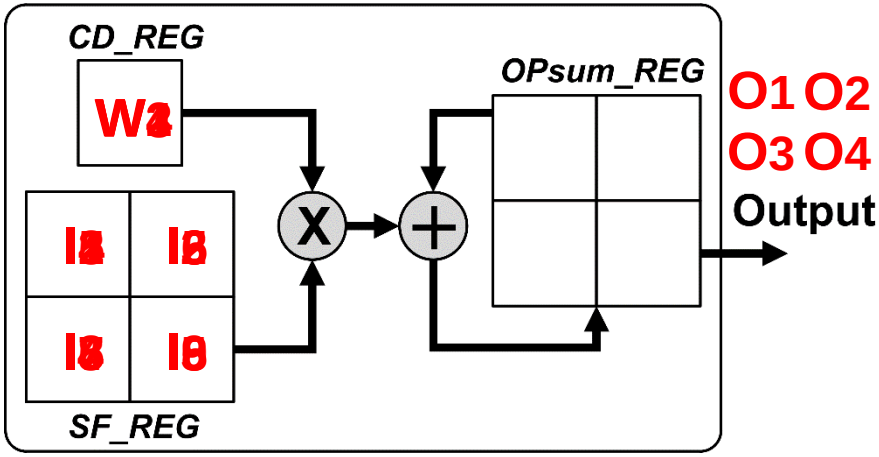
Weight-wise NN Processing Mechanism

- Computing data register (*CD_REG*).
- Scaling factor register (*SF_REG*).
  - SF register size will not be restricted.
- Reduce memory access.



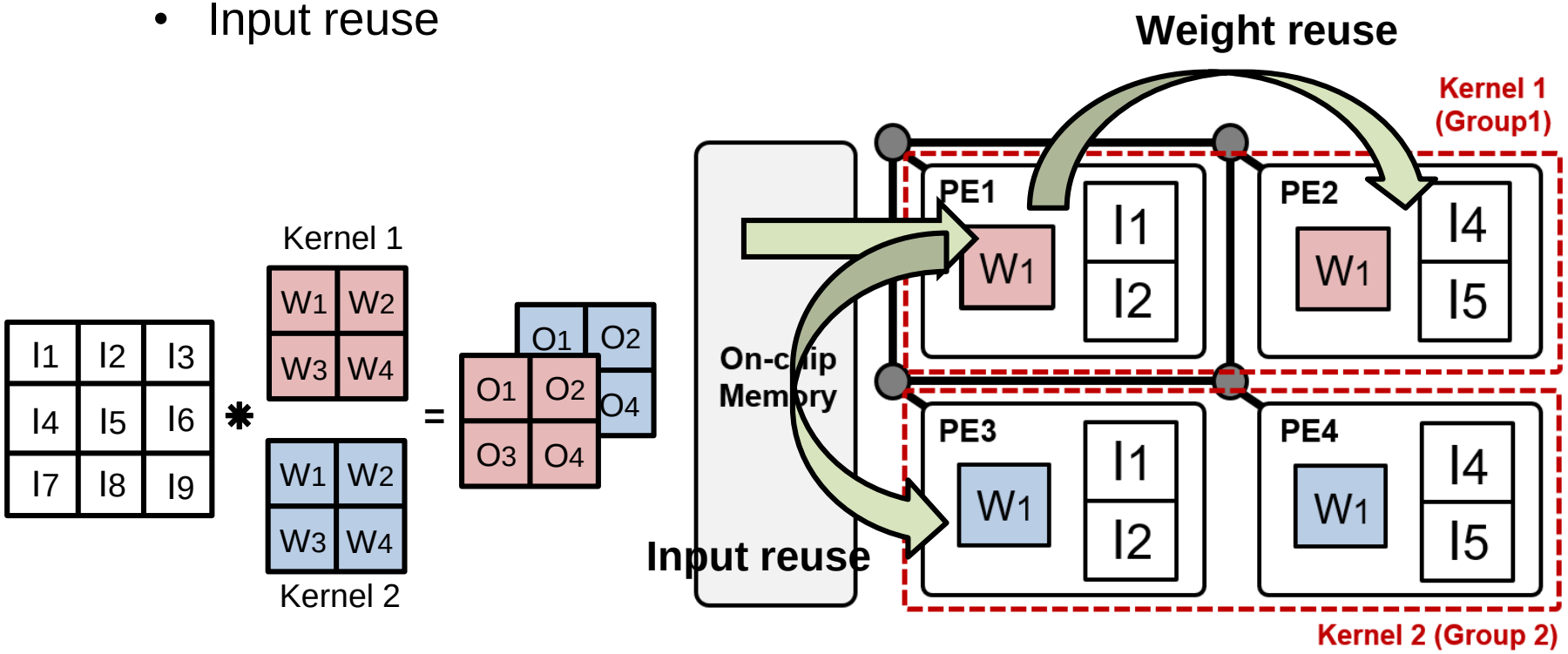
$$O_1 = (I_1 \times W_1) + (I_2 \times W_2) + (I_4 \times W_3) + (I_5 \times W_4)$$
$$O_2 = (I_2 \times W_1) + (I_3 \times W_2) + (I_5 \times W_3) + (I_6 \times W_4)$$
$$O_3 = (I_4 \times W_1) + (I_5 \times W_2) + (I_7 \times W_3) + (I_8 \times W_4)$$
$$O_4 = (I_5 \times W_1) + (I_6 \times W_2) + (I_8 \times W_3) + (I_9 \times W_4)$$

OPsum



Hybrid Data Reuse Method by Using NoC

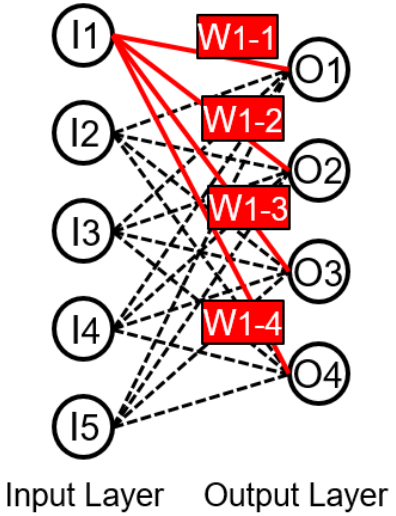
- After accessing the data from on-chip memory once, PE will share duplicated data through packet transmission.
  - Does not need to design complicated dataflow.
  - Weight reuse
  - Input reuse

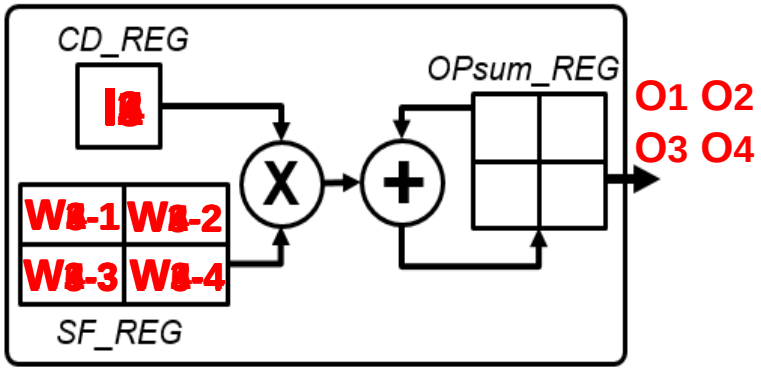


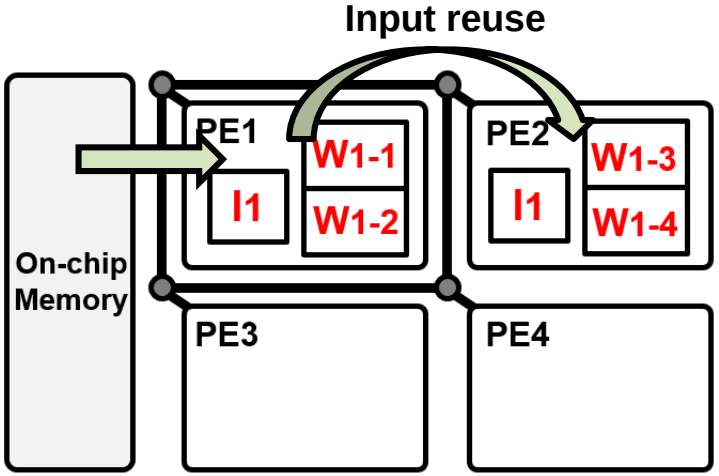
The Processing of Fully-connected Layer

- The proposed processing mechanism can also be applied in the fully-connected layer.
  - Store the input data to the *CD_REG*, and the corresponding weights to the *SF_REG*, respectively.
  - Input reuse

$$O_1 = (I_1 \times W_{1-1}) + (I_2 \times W_{2-1}) + (I_3 \times W_{3-1}) + (I_4 \times W_{4-1}) + (I_5 \times W_{5-1})$$
$$O_2 = (I_1 \times W_{1-2}) + (I_2 \times W_{2-2}) + (I_3 \times W_{3-2}) + (I_4 \times W_{4-2}) + (I_5 \times W_{5-2})$$
$$O_3 = (I_1 \times W_{1-3}) + (I_2 \times W_{2-3}) + (I_3 \times W_{3-3}) + (I_4 \times W_{4-3}) + (I_5 \times W_{5-3})$$
$$O_4 = (I_1 \times W_{1-4}) + (I_2 \times W_{2-4}) + (I_3 \times W_{3-4}) + (I_4 \times W_{4-4}) + (I_5 \times W_{5-4})$$

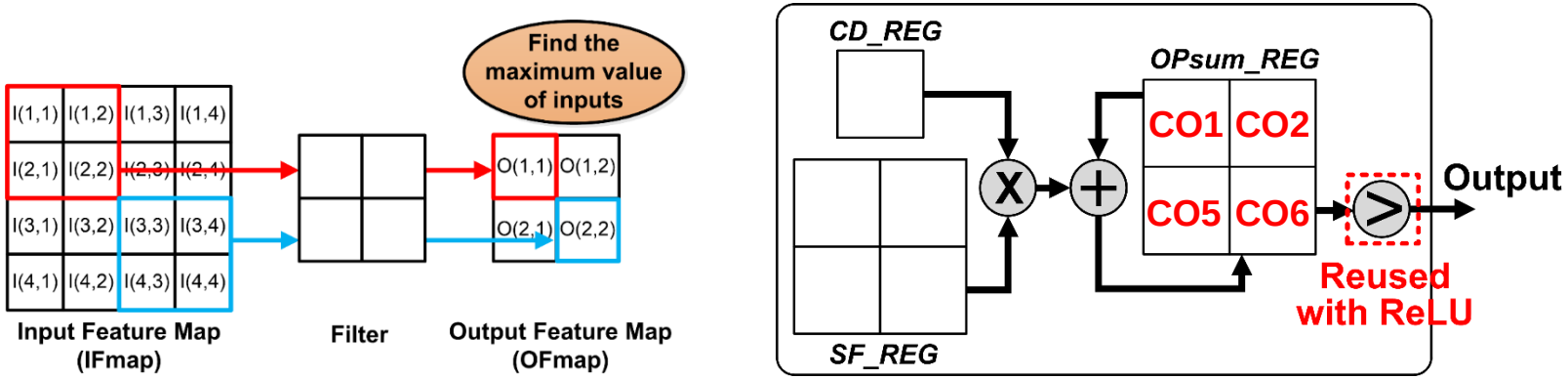


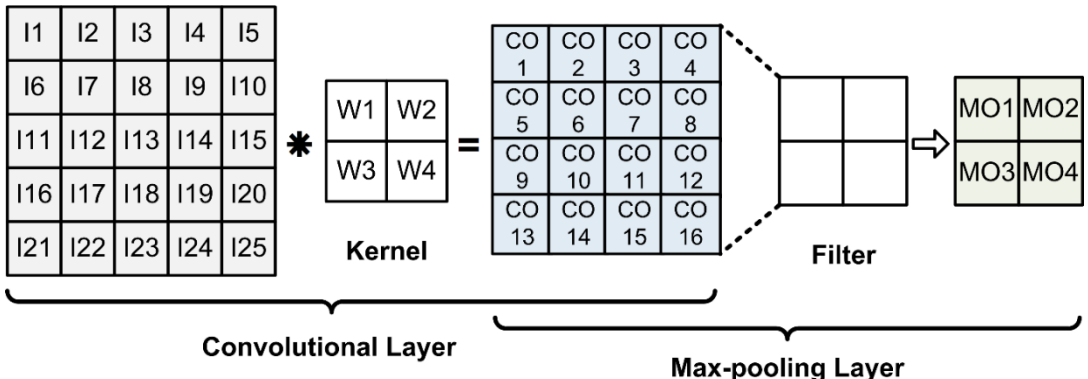




The Applicability of Max-Pooing Layer

- The proposed mechanism can be performed in the max-pooling layer.
- The *SF_REG* size will be designed as a multiple of the filter size.
- The comparator can be reused by ReLU.

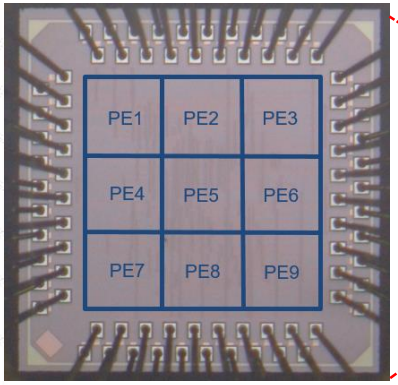


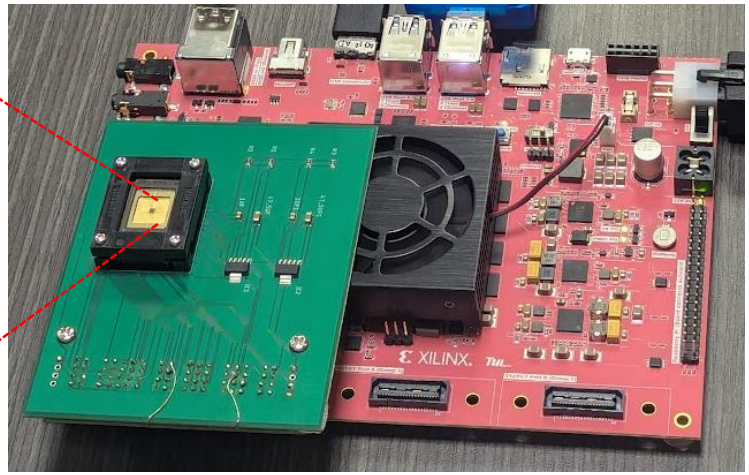


The first NoC-based Reconfigurable DNN Accelerator with AXI communication protocol in Taiwan

- Support arbitrary kernel size and shape to compute the convolution operations
- Adopting flexible Network on Chip (NoC) interconnection to reduce interconnection complexity and reduce time-to-market
- Adopt AXI4-stream communication protocol

Technology		TSMC 40nm
Area (mm^2)	Chip	1.4 x 1.4
	Core	0.84 x 0.84
	Gate count	6,871k
IO/Core VDD (v)		2.5/0.9
Clock freq (MHz)		105
Power (mW)		10.3672
Throughput (GOPS)		143.5





Demonstration Videos

Low-complex geological analysis system with spectral AI-sonar

Motor Faults Diagnosis by Using NoC-based DNN Chip

Take Away

- The most neural network operations are matrix operation
  - Proper dataflow is important to design an efficient neural network chip
- Memory access is the bottleneck in a neural network system
  - Data reuse is important to improve the performance
- Data communication become complex in large-scale neural network model
  - Network-on-Chip (NoC) interconnection become emerging in current many-core system design

Conclusion

- Fundamental of DNN accelerator design
  - DNN hardware ensures information privacy, improves performance, and realizes edge AI applications.
  - The data delivery and sharing on chips becomes the performance bottleneck.

- Fundamental of Network-on-Chip (NoC) interconnection
  - NoC-based interconnection provide a flexible and efficient interconnection way to build a DNN accelerator
  - It becomes a popular way in the industry to build a multi-core system

- Two demonstration videos
  - AI sonar
  - Anomaly detection


國立陽明交通大學 National Yang Ming Chiao Tung University
Thank you