Accelerators-I

Tushar Krishna
Associate Professor @ Georgia Tech
Visiting Professor @ MIT EECS and CSAIL

Outline

- Why do we need accelerators?
- Why now?
- How to design accelerators

Outline

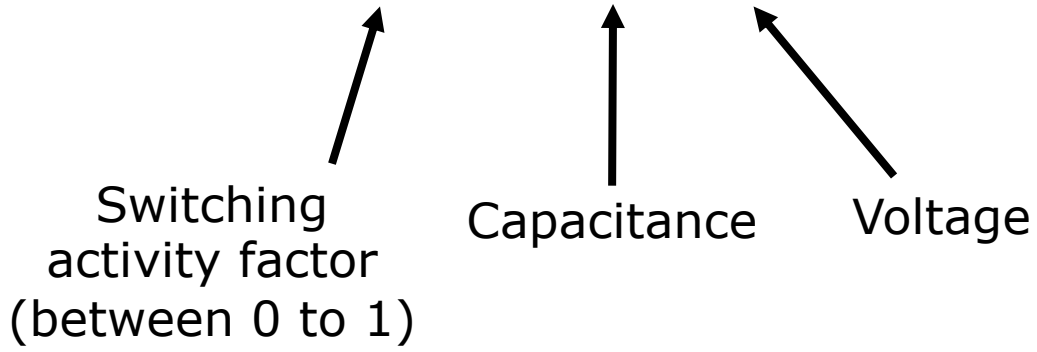
- Why do we need accelerators?
- Why now?
- How to design accelerators

Power Constraints in Modern Computers

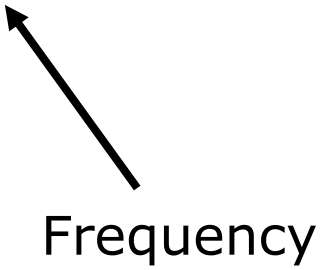
<< 1W **~ 1W** **~ 15W** **~ 50W** **~ 100W** **~ 100W**

Energy and Power Consumption

- Energy Consumption = $\alpha \times C \times V^2$



- Power Consumption = $\alpha \times C \times V^2 \times f$



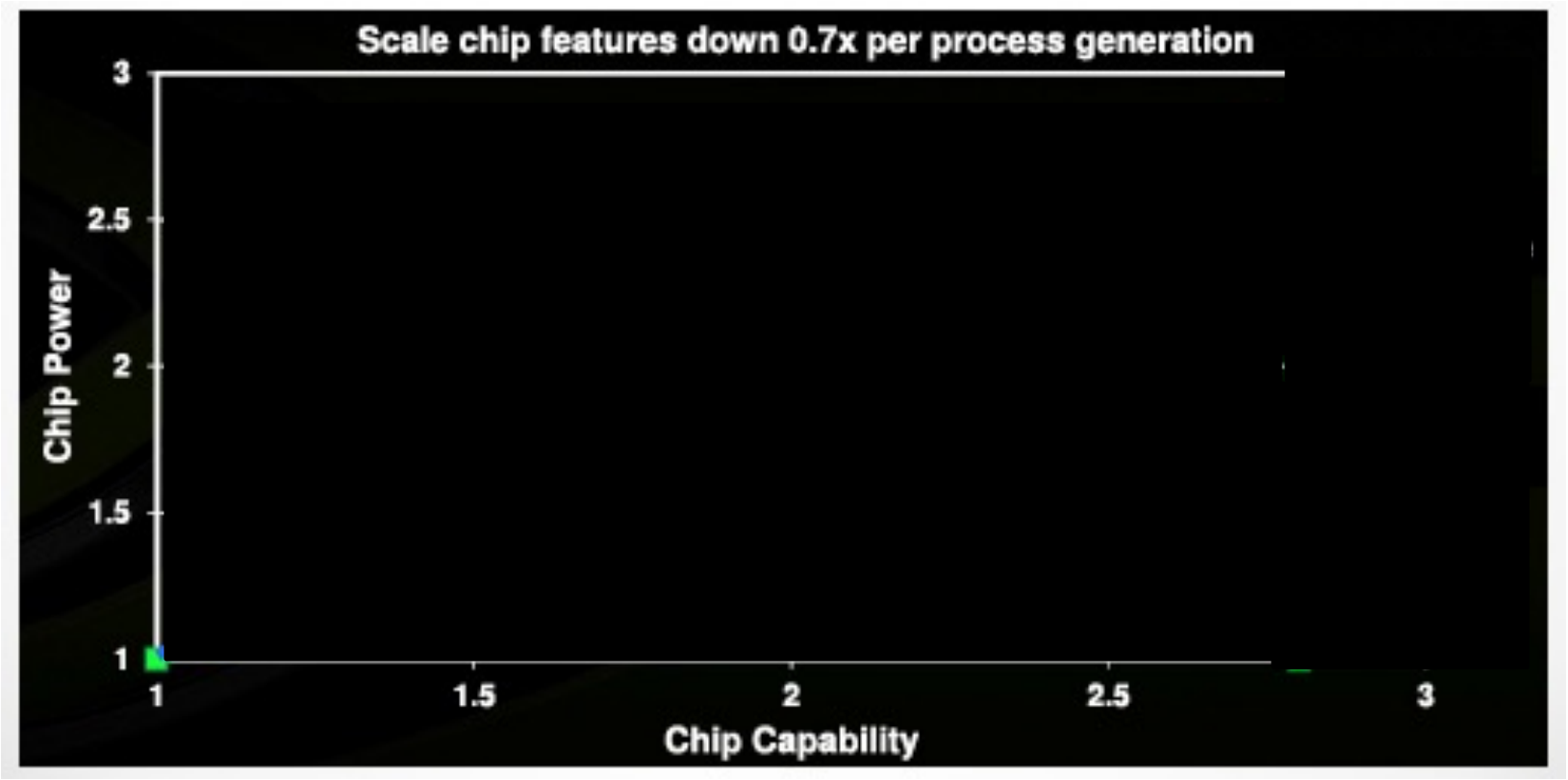
CMOS Scaling (idealized)

	Gen X	Gen X+1
Gate Width (Moore's Law)	 1.0	 0.7
Device Area/ Capacitance	 1.0 1.0	 0.5 0.7  0.5 0.7
Voltage (Dennard's)	1.0	0.7
Energy	~ 1.0 x 1.0^2 = 1.0	~ 2 x 0.7 x 0.7^2 = 0.65
Delay	1.0	0.7
Frequency	1/1.0 = 1.0	1/0.7 = 1.4
Power	~ 1.0 x 1.0^2 x 1.0 = 1.0	~ 2 x 0.7 x 0.7^2 x 1.4 = 1.0

[Dennard et al., "Design of ion-implanted MOSFET's with very small physical dimensions", JSSC 1974]

CMOS Scaling (idealized)

- Moore's Law (transistors) + Dennard's Scaling (voltage)
  - 2.8X in chip capability per generation at constant power
  - ~5000x performance improvement in 20 years



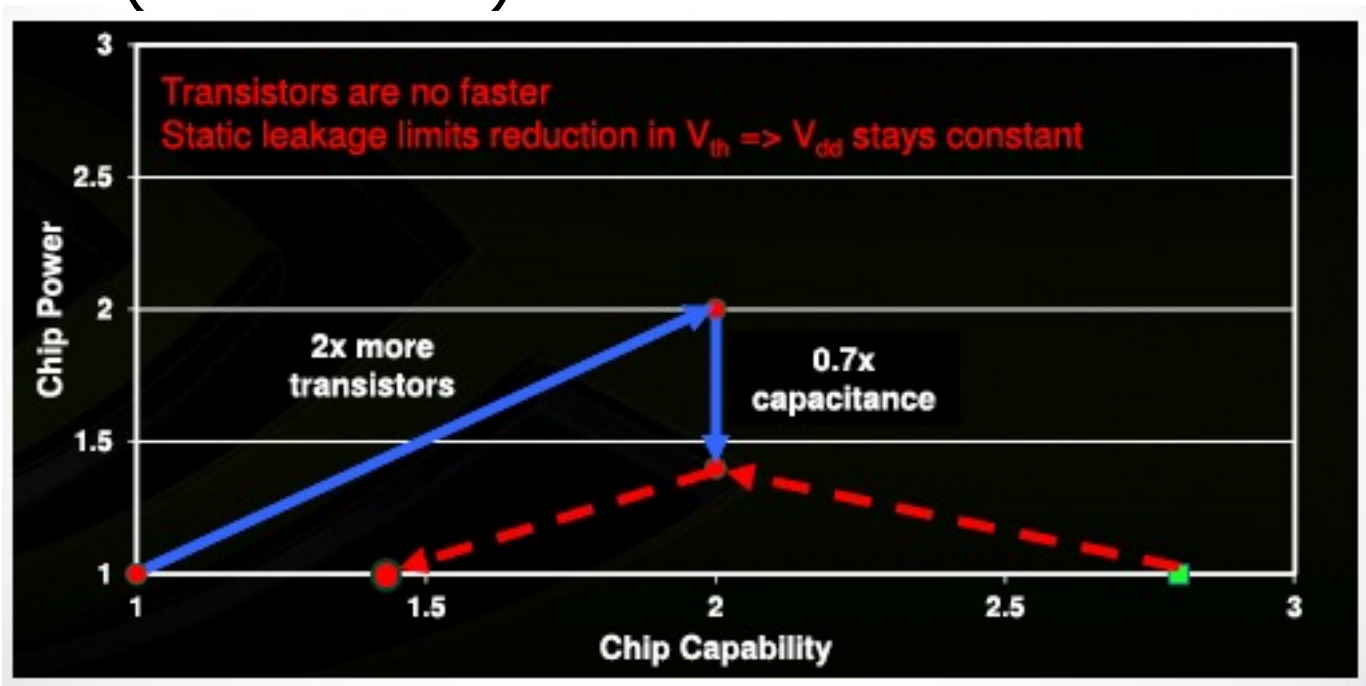
Source: "Advancing Computer Systems Without Technology Progress, ISAT Outbrief, 2012

"Power Wall"

- Dennard's Scaling has stopped
  - Why? Already operating close to $V_{threshold}$

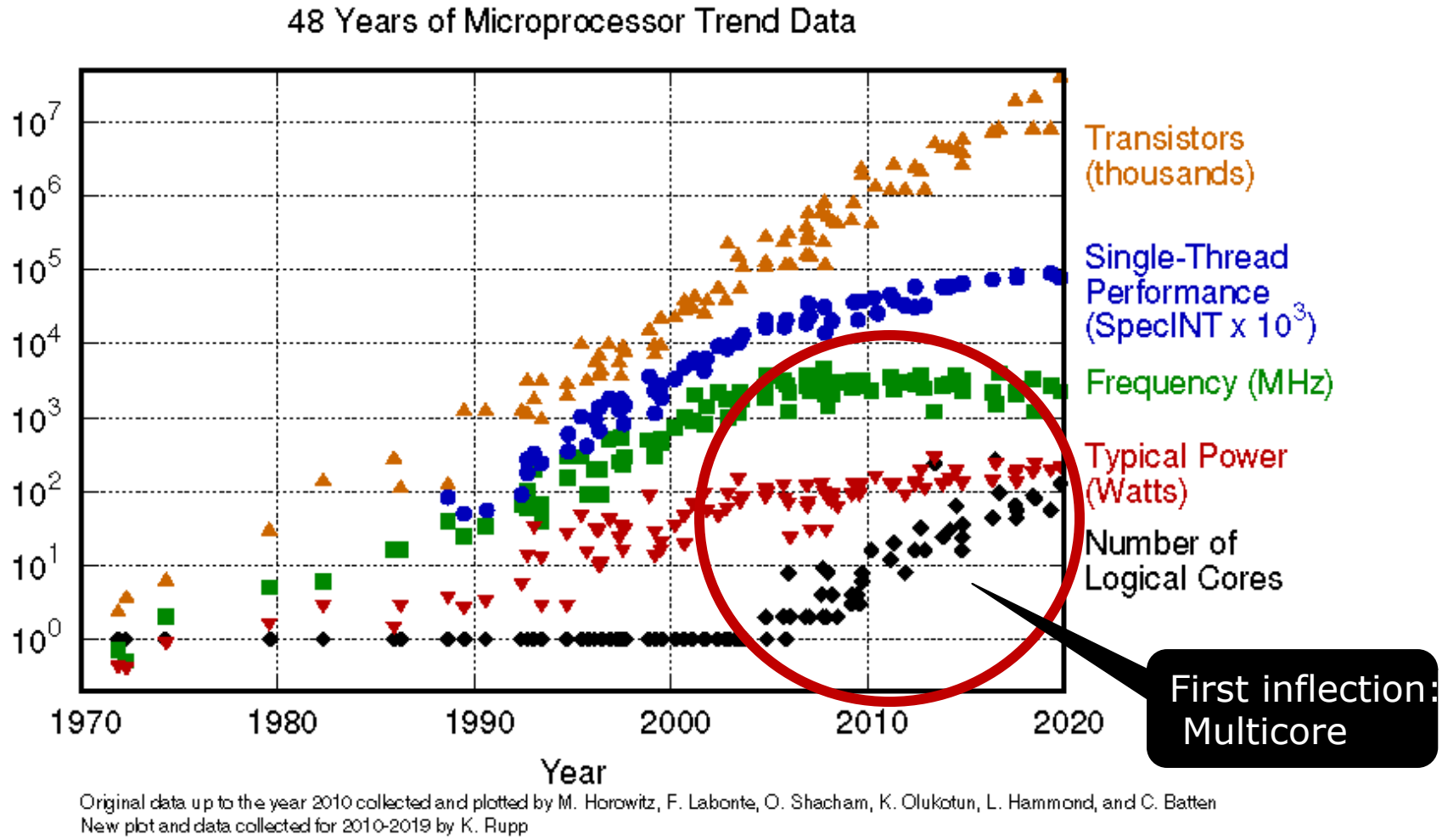
→ Cannot increase operating frequency

($P = \frac{1}{2} CV^2$)

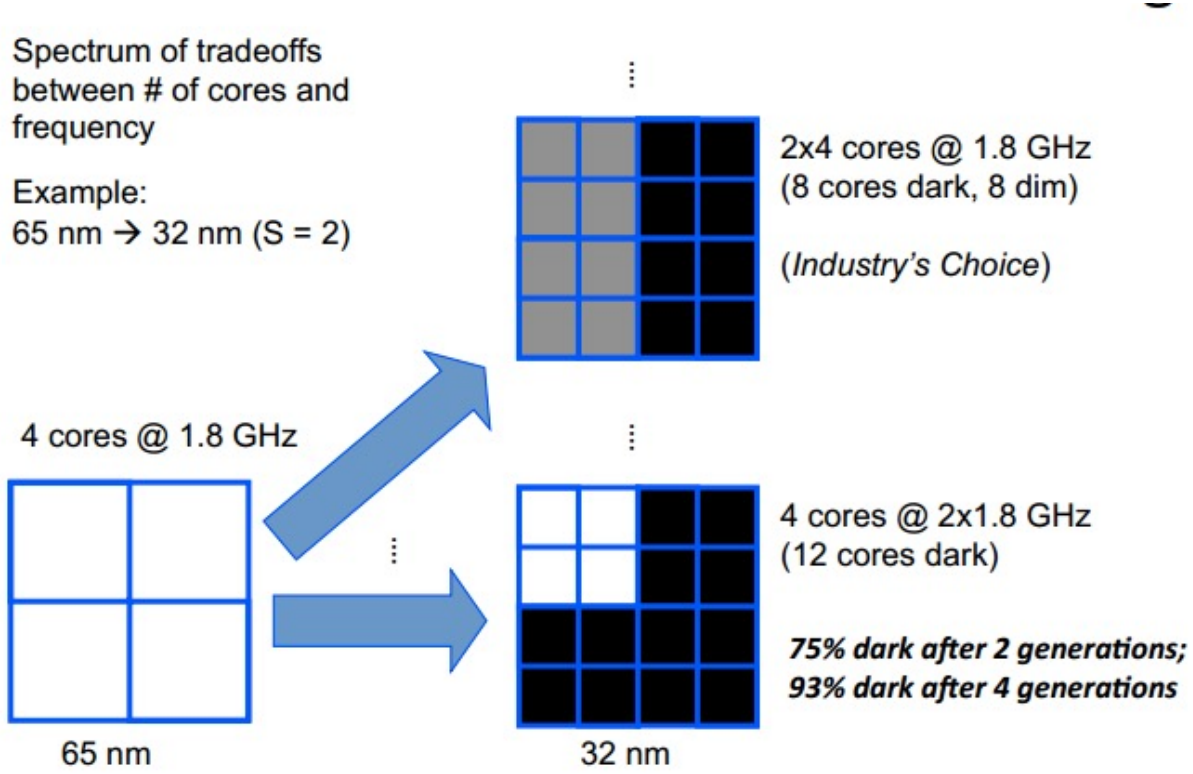


Source: "Advancing Computer Systems Without Technology Progress, ISAT Outbrief, 2012

Technology Trends



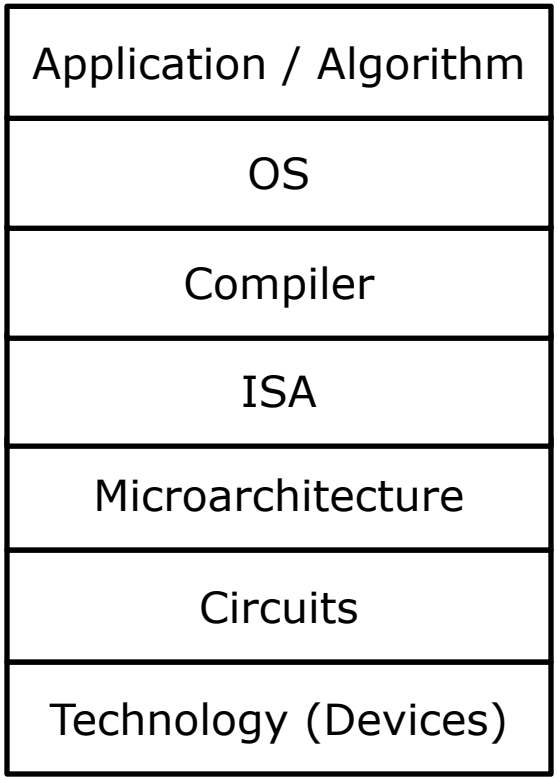
Utilization Wall ("Dark Silicon")



Source: M. Taylor, "Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse", DAC 2012

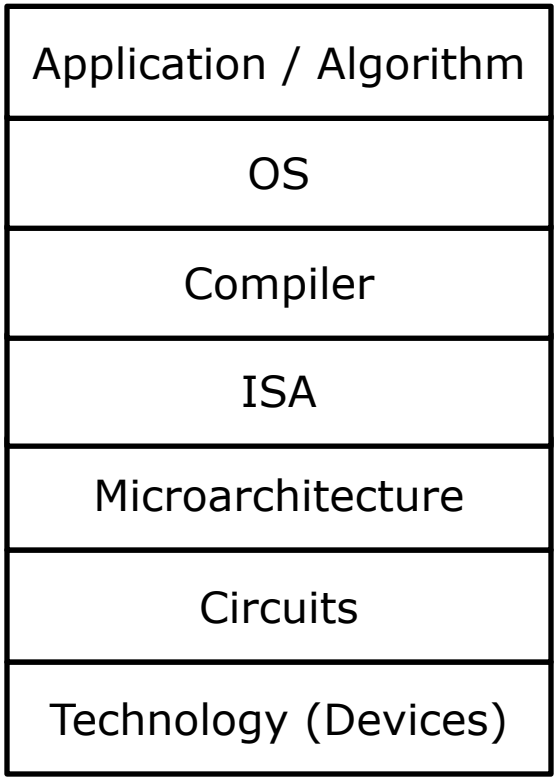
Power Management options?

- Tackle power across all levels of the computing stack
  - **Technology**
    - Cost of switching
  - **Circuits**
    - High-speed vs low-power implementation
    - Clock gating and power gating support
    - DVFS
  - **Microarchitecture and ISA**
    - Simplify design
    - Parallelism
    - Heterogeneity and Specialization
  - **Compiler**
    - Instruction footprint and Cache behavior
  - **OS**
    - Tune DVFS and power states
  - **Algorithm:**
    - Switching activity



Power Management options?

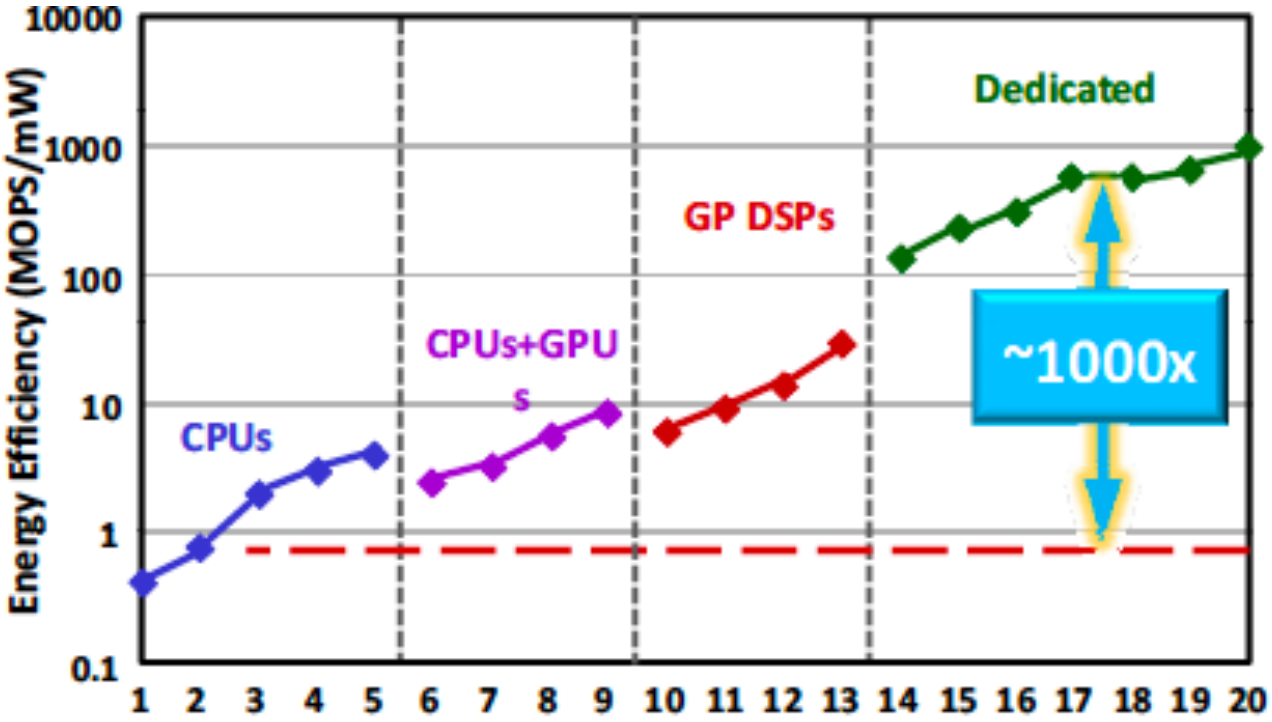
- Tackle power across all levels of the computing stack
  - **Technology**
    - Cost of switching
  - **Circuits**
    - High-speed vs low-power implementation
    - Clock gating and power gating support
    - DVFS
  - **Microarchitecture and ISA**
    - Simplify design
    - Parallelism
    - Heterogeneity and Specialization
  - **Compiler**
    - Instruction footprint and Cache behavior
  - **OS**
    - Tune DVFS and power states
  - **Algorithm:**
    - Switching activity



Heterogeneity and Specialization

Chip type:

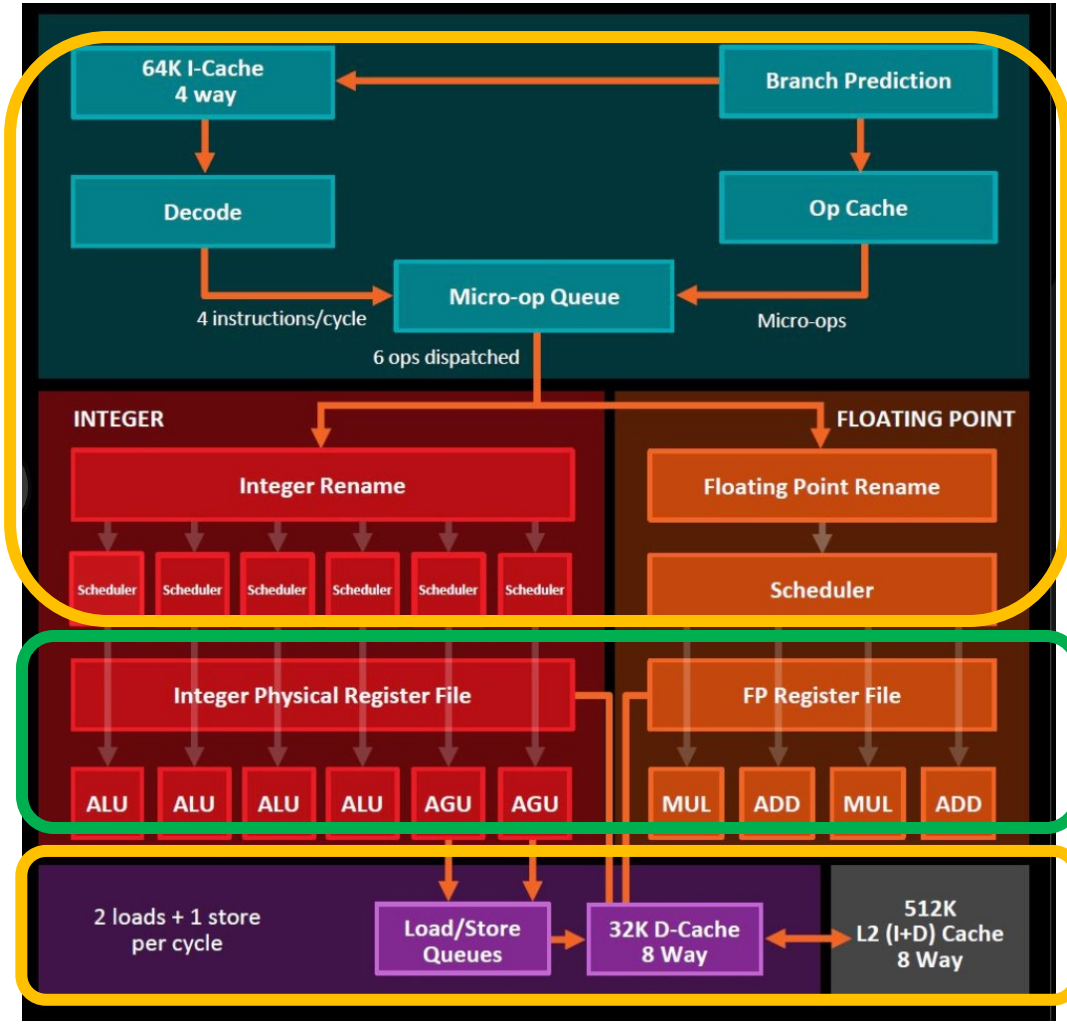
- Microprocessor
- Microprocessor + GPU
- General purpose DSP
- Dedicated design



Improve Energy Efficiency via Customization!

Why?

A modern CPU



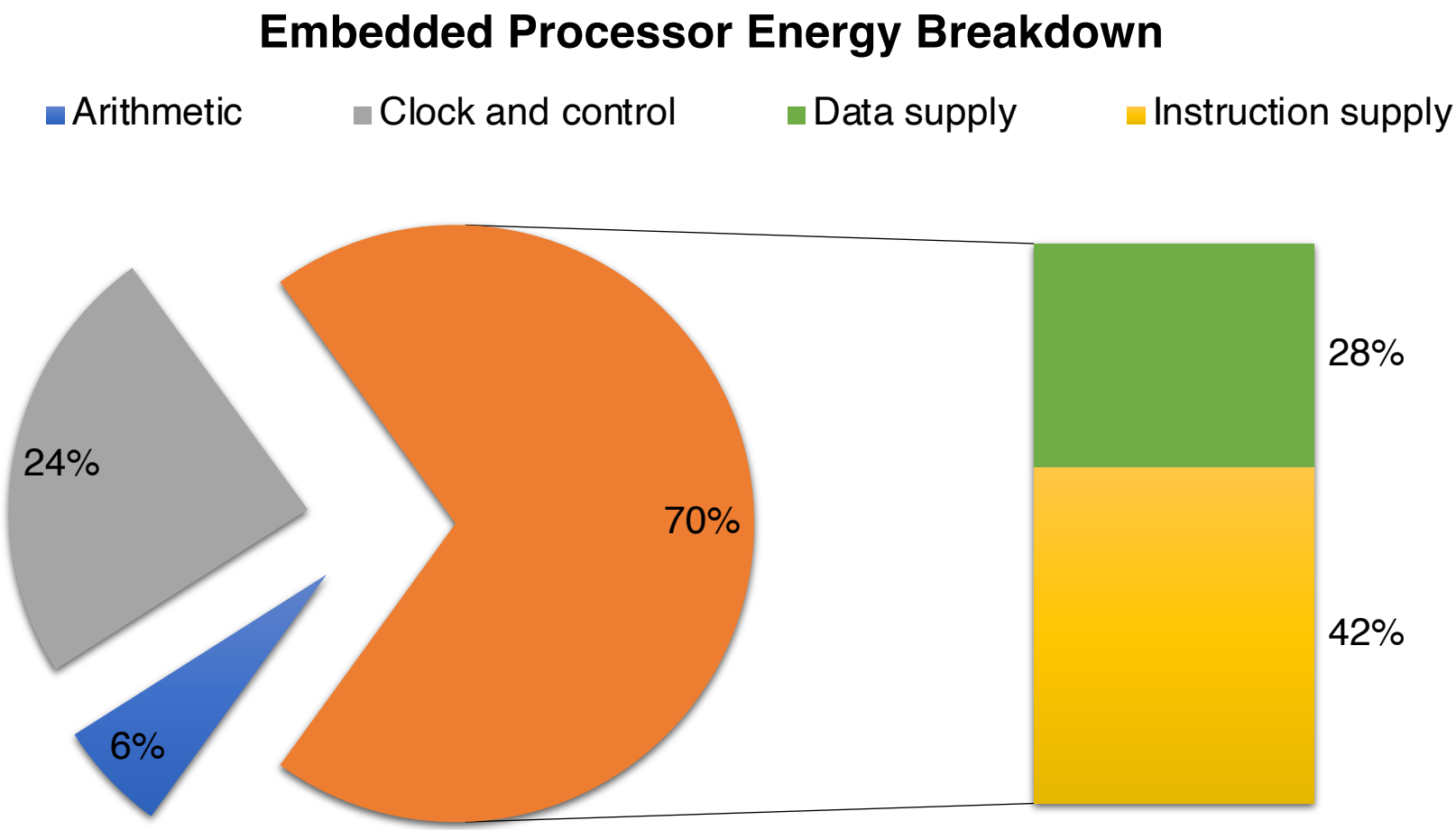
AMD Zen (2016)

Instruction Fetch, Decode, Scheduling and Speculation for "programmability"

Actual Computation

Implicit data management via caches

Quantifying this overhead



Source: Dally et al. Efficient Embedded Computing, IEEE'08

Performance/Area Benefits

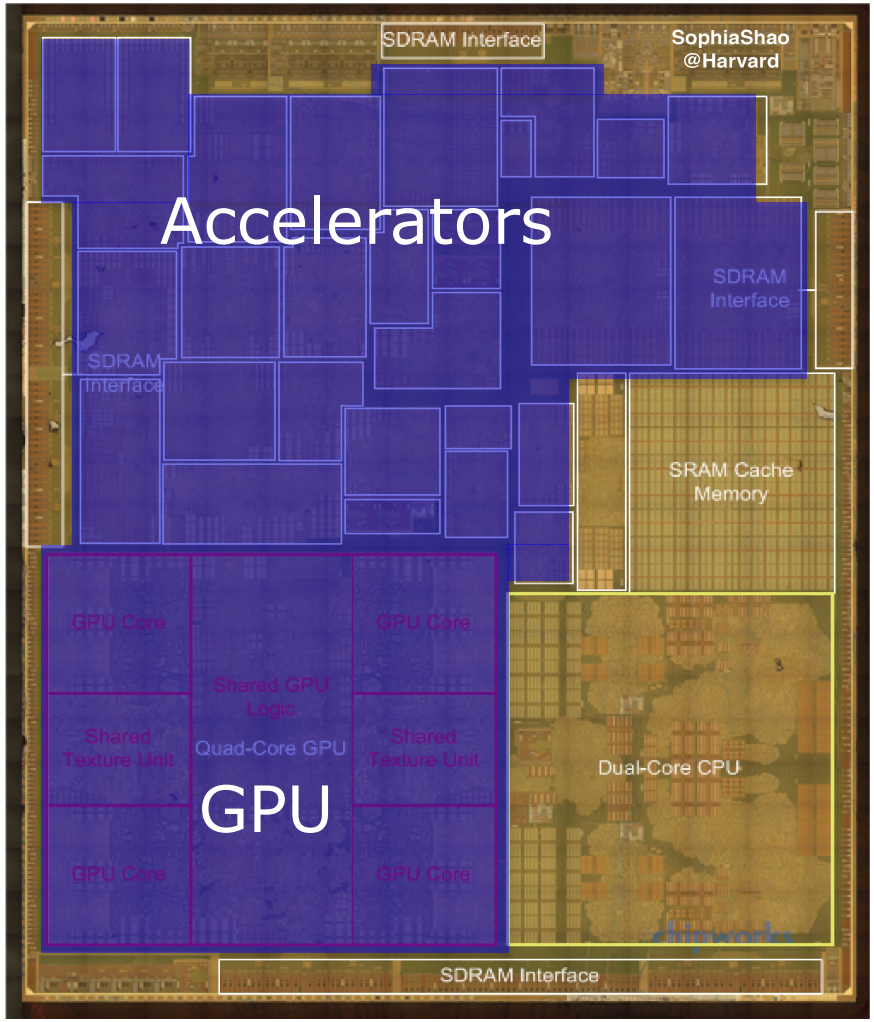
		GFLOP/s	(GFLOP/s)/mm^2	GFLOP/J
FFT-2^{10}	Intel Core i7 (45nm)	67	0.35	0.71
	Nvidia GTX285 (55nm)	250	1.41	4.2
	Nvidia GTX480 (40nm)	453	1.08	4.3
	ATI R5870 (40nm)	-	-	-
	Xilinx V6-LX760 (40nm)	380	0.99	6.5
	same RTL std cell (65nm)	952	239	90
		Mopt/s	**(Mopts/s)/mm^2**	**Mopts/J**
Black-Scholes	Intel Core i7 (45nm)	487	2.52	4.88
	Nvidia GTX285 (55nm)	10756	60.72	189
	Nvidia GTX480 (40nm)	-	-	-
	ATI R5870 (40nm)	-	-	-
	Xilinx V6-LX760 (40nm)	7800	20.26	138
	same RTL std cell (65nm)	25532	1719	642.5

Source:Chung et al., "Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGPUs?", MICRO 2010

Outline

- Why do we need accelerators?
- Why now?
- How to design accelerators

Domain-specific Accelerators in SoCs

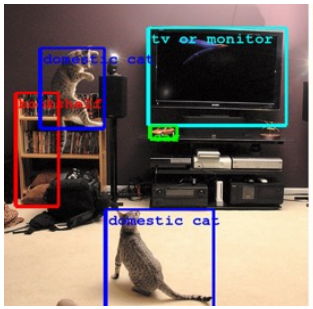


Apple A8 SoC

The rise of AI

"AI is the new electricity" – Andrew Ng

Object Detection



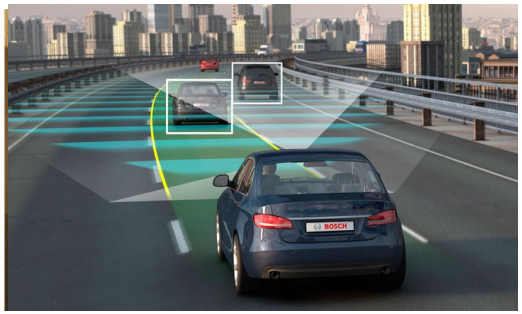
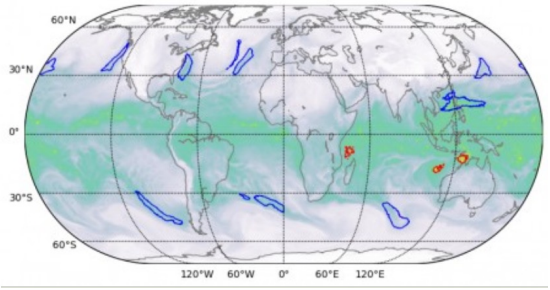
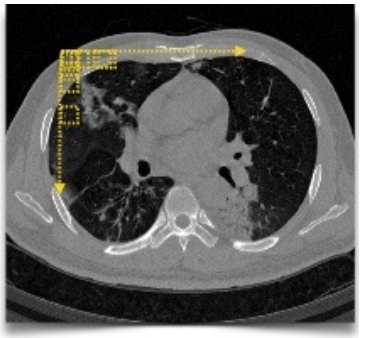


Image Segmentation

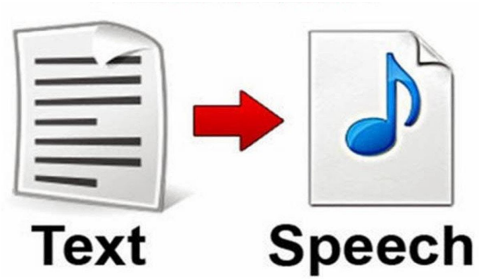


Medical Imaging



Speech Recognition

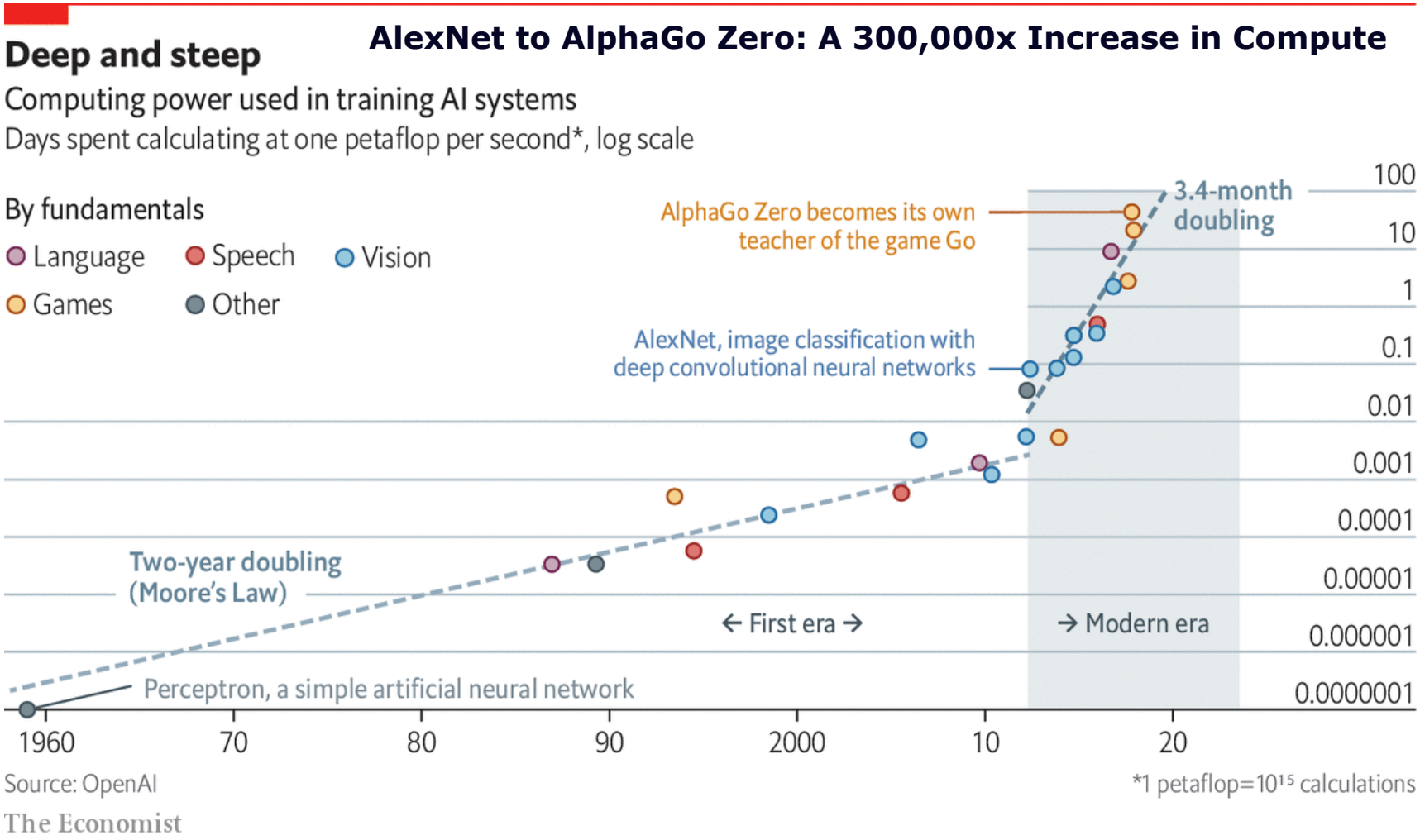
Text to Speech



Recommendations

Games

AI Compute Demands Growing Exponentially



Source: https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem

Cloud SW Companies Building HW

Google's dedicated TensorFlow processor, or TPU, crushes Intel, Nvidia in inference workloads

By Joel Hruska on April 6, 2017 at 9:48 am | 25 Comments

How Amazon is racing to catch Microsoft and Google in generative A.I. with custom AWS chips

PUBLISHED SAT, AUG 12 2023•9:00 AM EDT | UPDATED MON, AUG 21 2023•7:40 PM EDT

Meta announces AI training and inference chip project

By **Katie Paul** and **Stephen Nellis**

May 18, 2023 4:34 PM EDT · Updated 7 months ago

Microsoft announces custom AI chip that could compete with Nvidia

PUBLISHED WED, NOV 15 2023•11:00 AM EST | UPDATED WED, NOV 15 2023•3:05 PM EST

HW Beyond Cloud Computing

WIRED

lusk Says Tesla Is Building Its Own Chip for Autopilot

TOM SIMONITE BUSINESS 12.08.17 01:09 PM

MUSK SAYS TESLA IS BUILDING ITS OWN CHIP FOR AUTOPILOT

Elon Musk disclosed plans for Tesla to design its own chip to power its self-driving function.

NASA/ALAMY

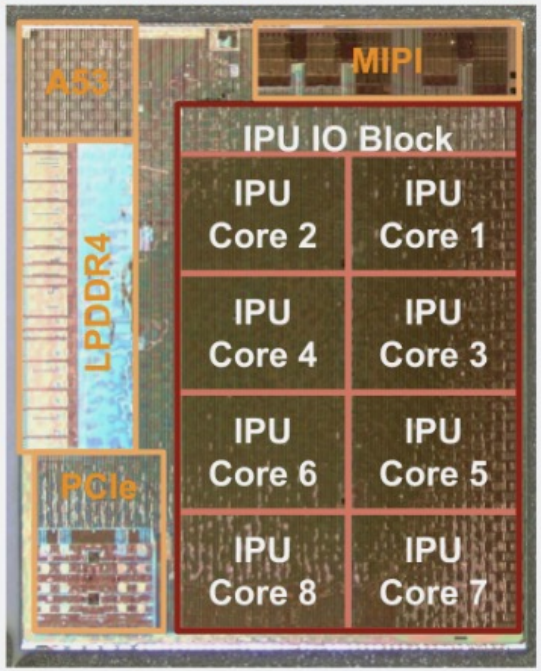
ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

TWO SOCS IS BETTER THAN ONE —

Surprise! The Pixel 2 is hiding a custom Google SoC for image processing

Google's 8-core Image Processing Unit will be enabled with Android 8.1.

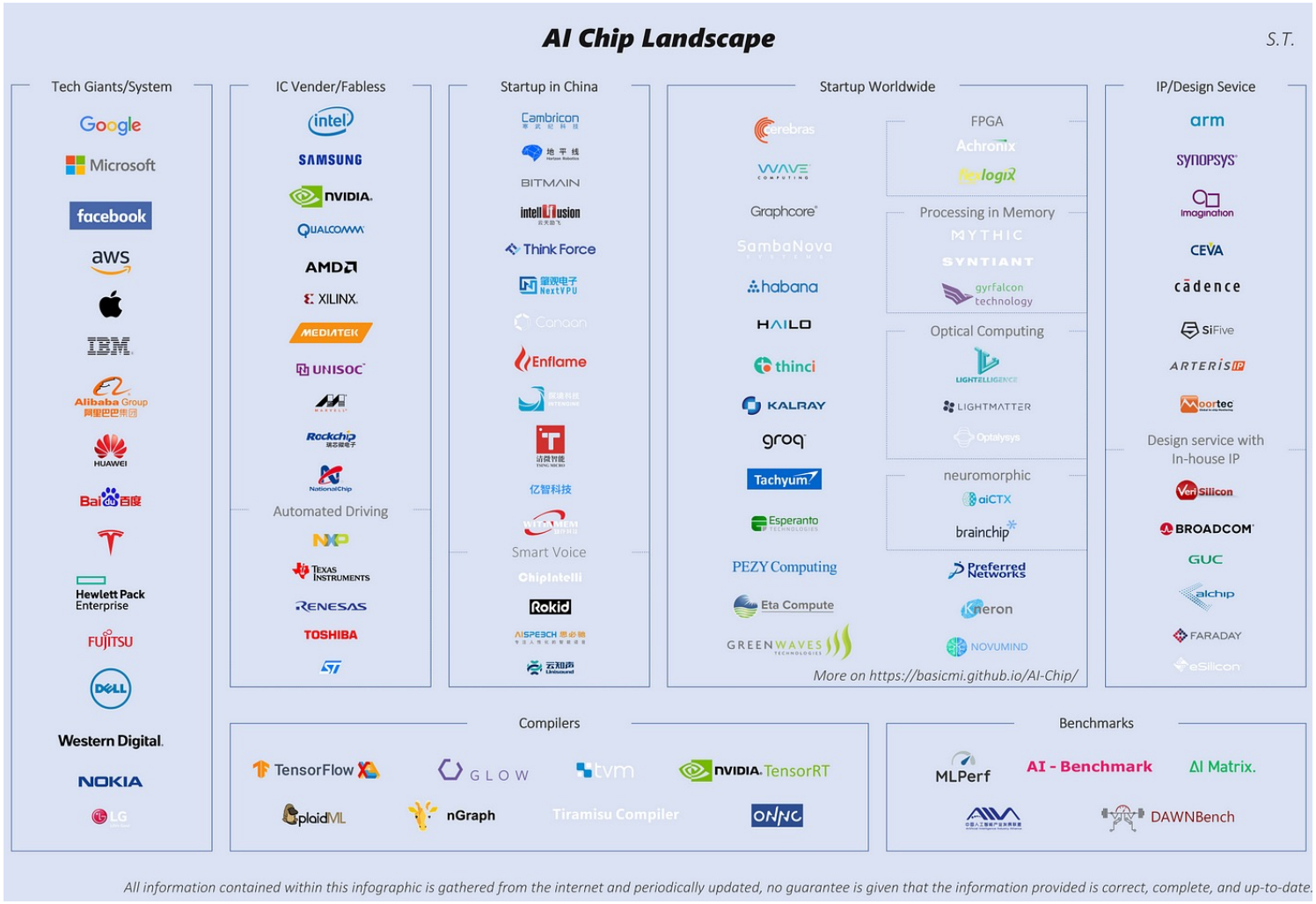
RON AMADEO - 10/17/2017, 9:00 AM



Enlarge / Google's Pixel Visual Core, an SoC designed for image processing and machine learning.

[Also Nvidia, Intel, Qualcomm…]

AI Chips ecosystem



Opportunities

From EE Times – September 27, 2016

"Today the job of training machine learning models is limited by compute, if we had faster processors we'd run bigger models…in practice we train on a reasonable subset of data that can finish in a matter of months. We could use improvements of several orders of magnitude – 100x or greater."

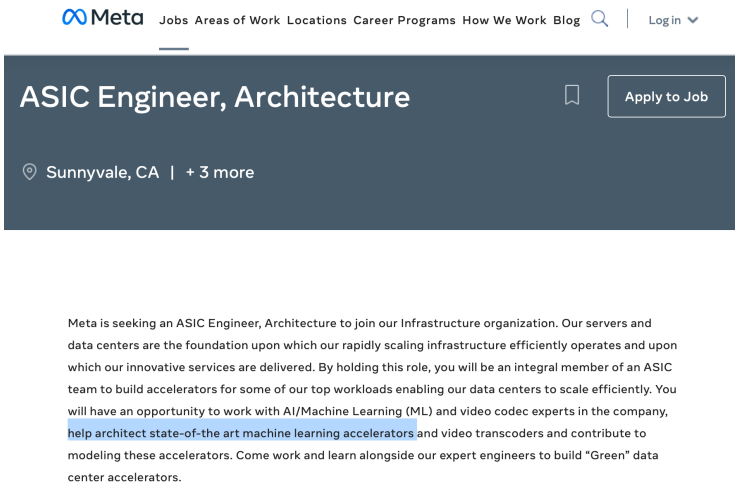
– Greg Diamos, Senior Researcher, SVAIL, Baidu

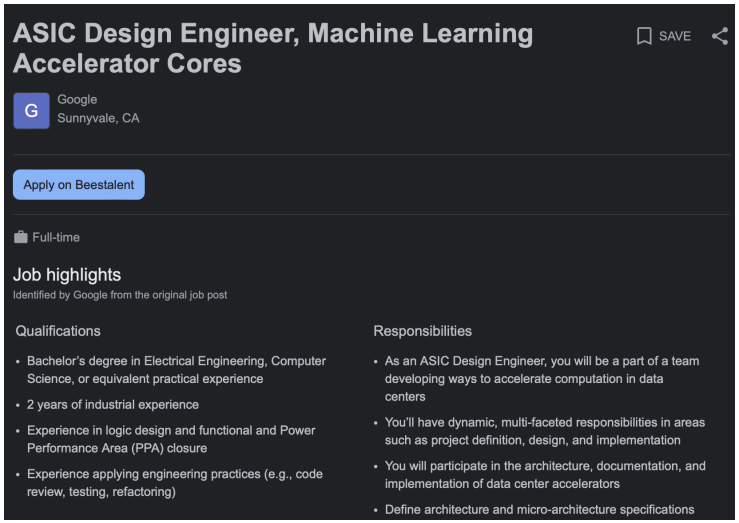
ACM's Celebration of 50 Years of the ACM Turing Award (June 2017)

"***Compute has been the oxygen of deep learning***"

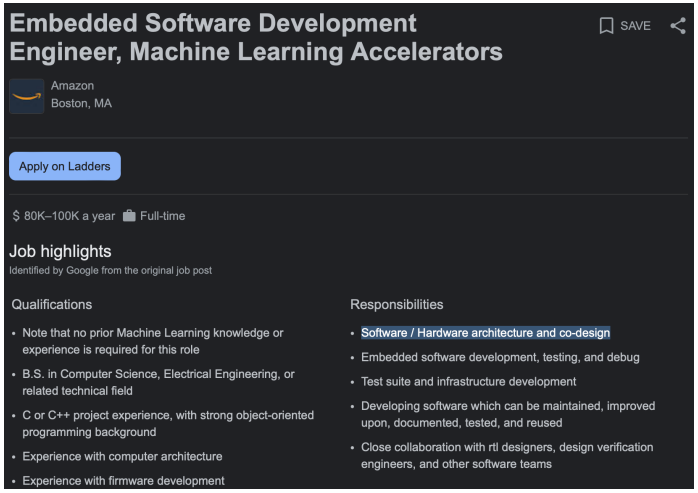
– Ilya Sutskever, Research Director of Open AI

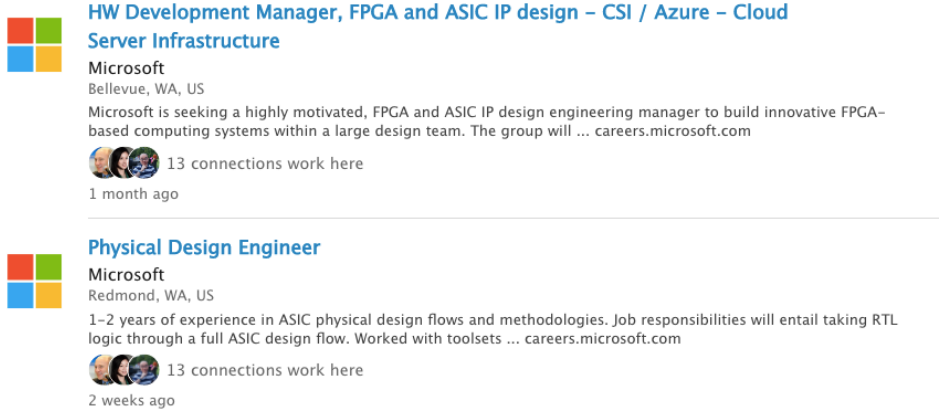
Demand for Computer Architects





...





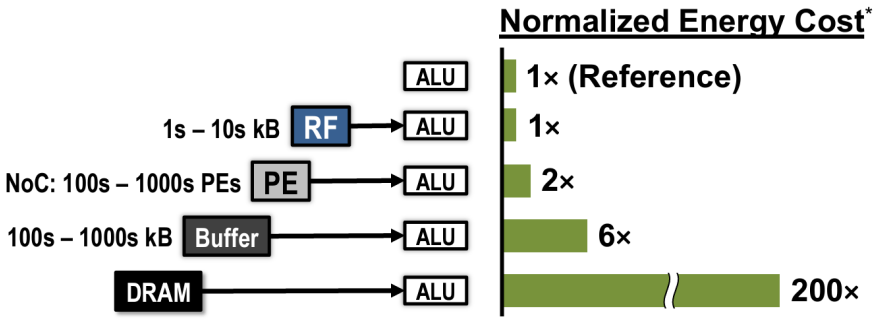
Why do we need DNN accelerators?

- **Millions of Parameters (i.e., weights)**
  - Billions of computations ➡ **Need lots of parallel compute**

DNN Topology	Number of Weights
AlexNet (2012)	3.98M
VGGnet-16 (2014)	28.25M
GoogleNet (2015)	6.77M
Resnet-50 (2016)	23M
DLRM (2019)	540M
Megatron (2019)	8.3B

This makes CPUs inefficient

  - Heavy data movement ➡ **Need to reduce energy**

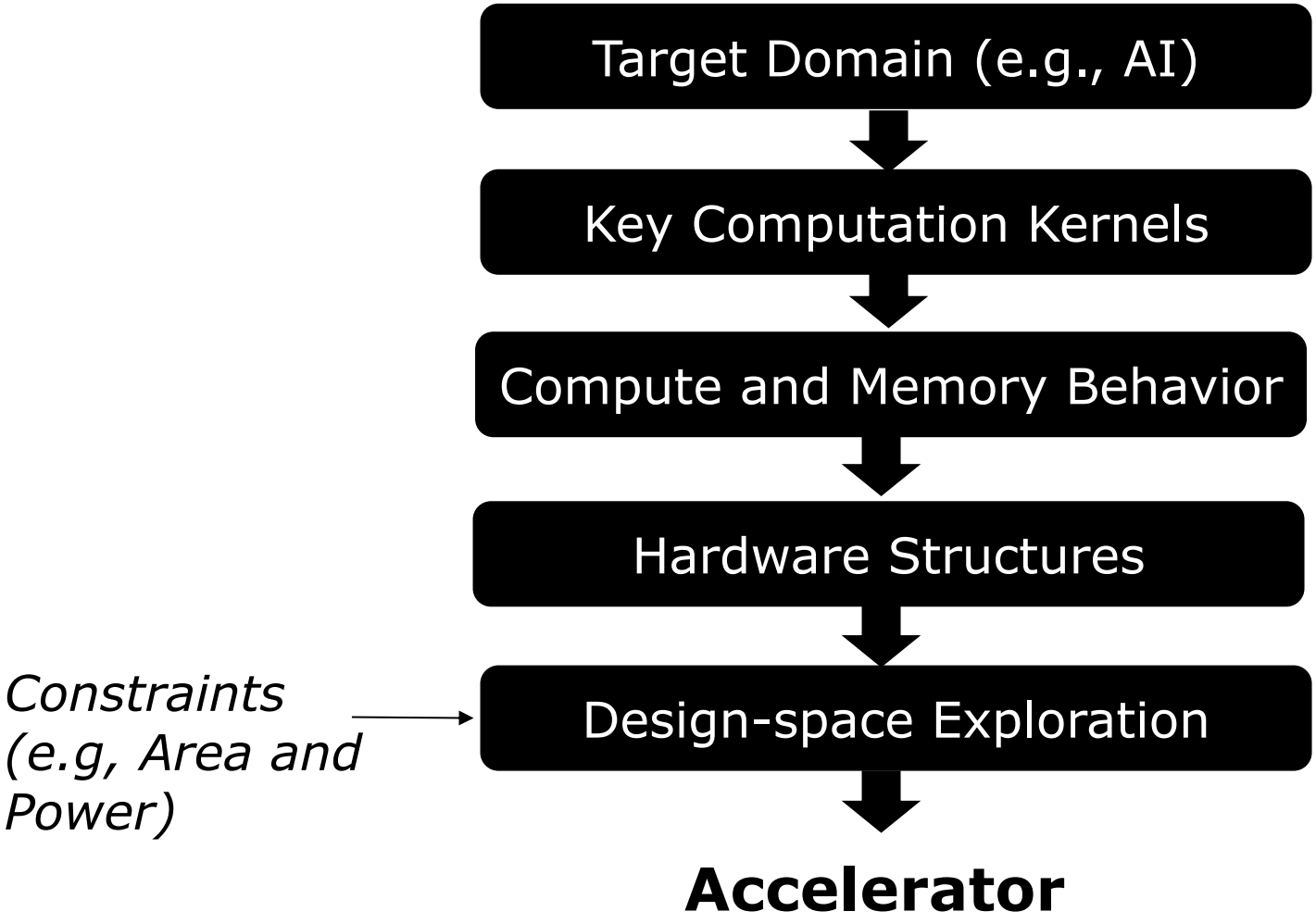


This makes GPUs inefficient

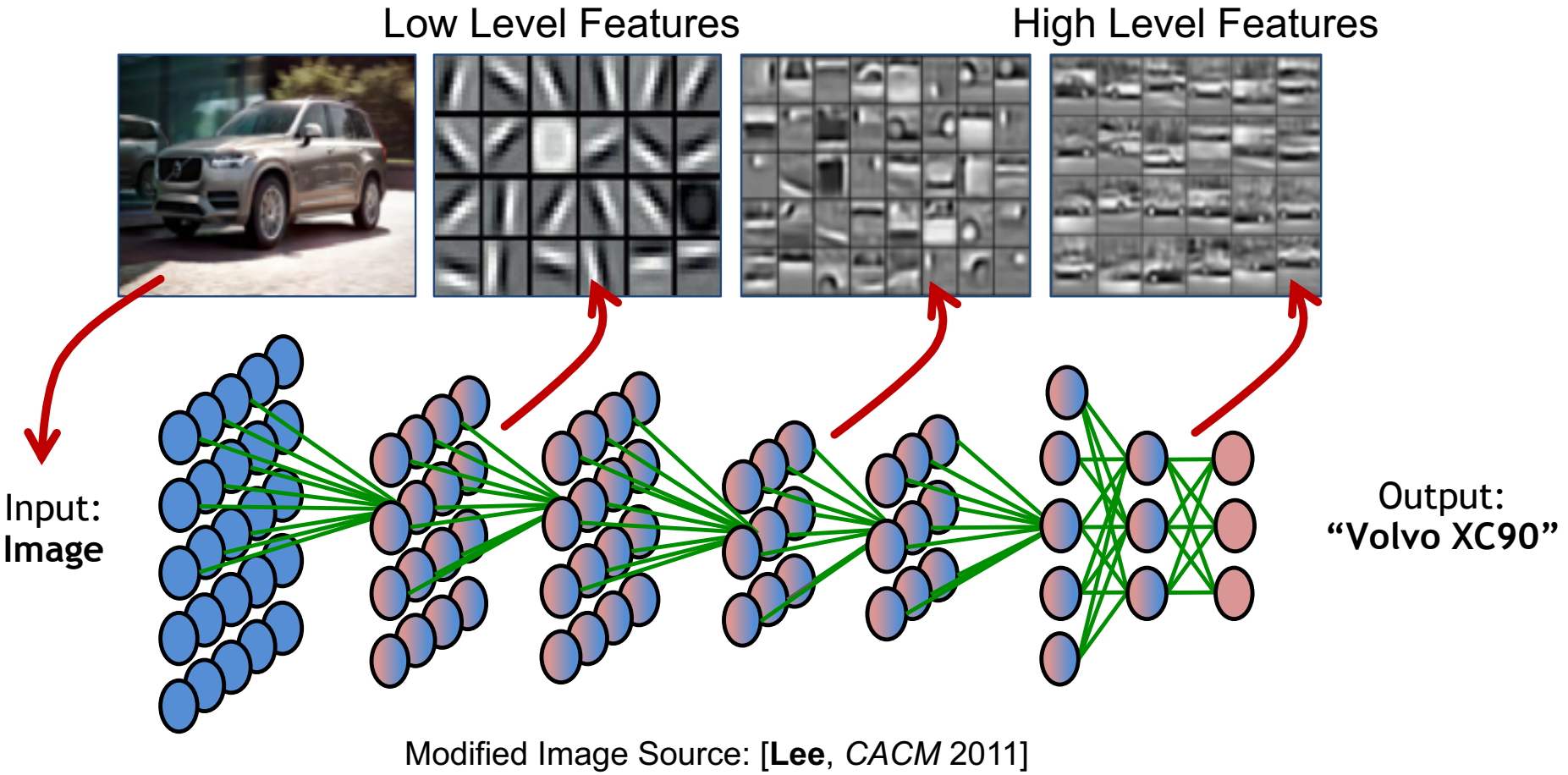
Outline

- Why do we need accelerators?
- Why now?
- How to design accelerators

HW-SW Co-Design

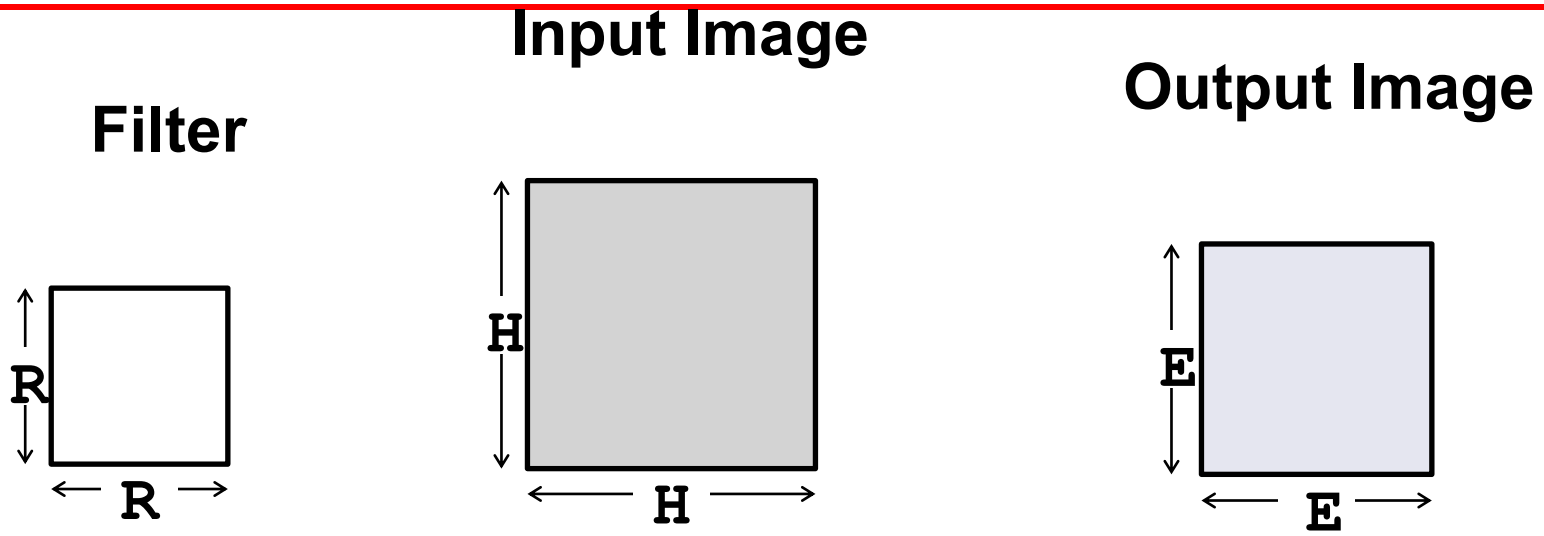


Understanding Deep Neural Networks

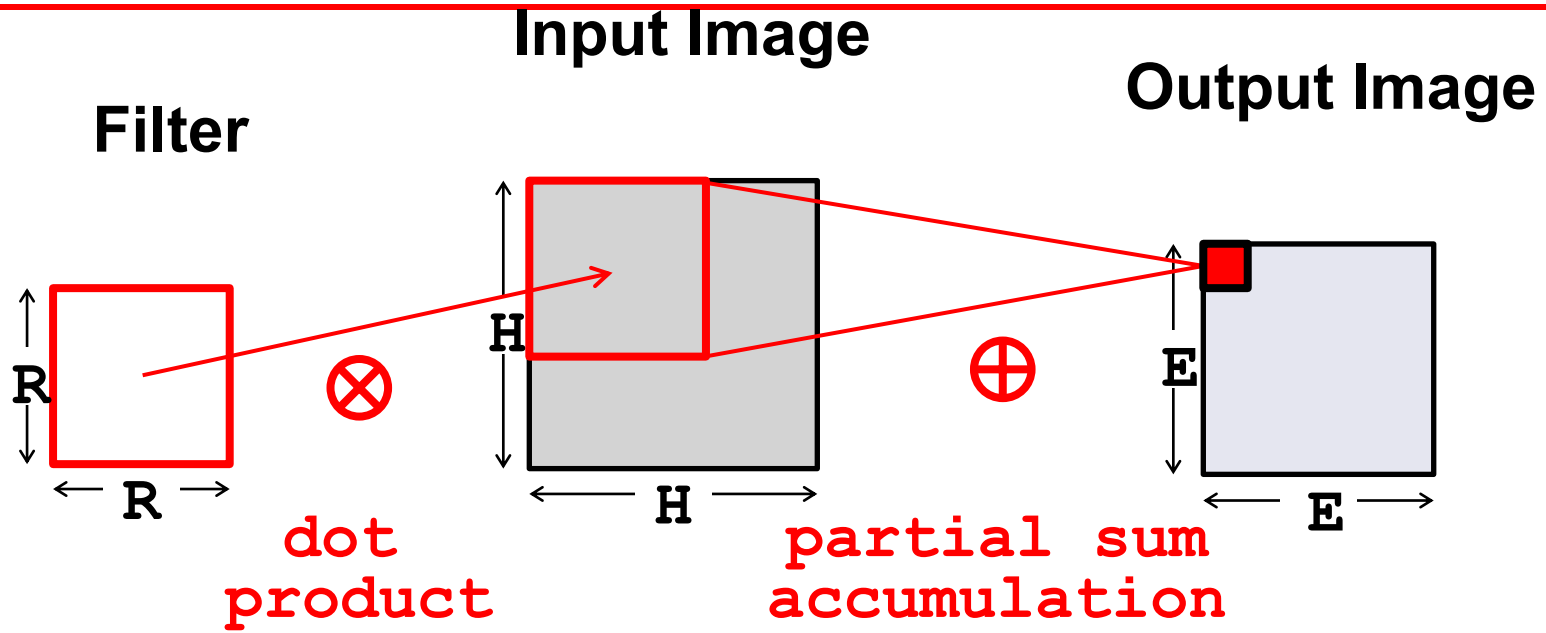


Modified Image Source: [**Lee**, *CACM* 2011]

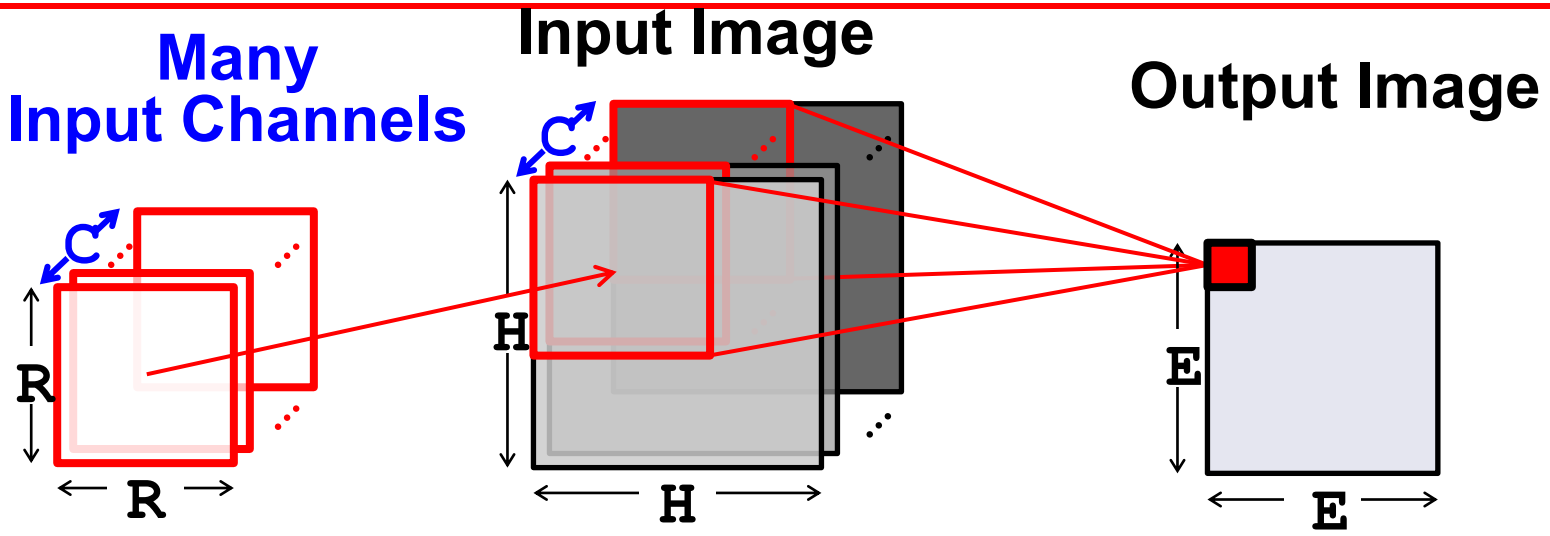
Convolution Operation



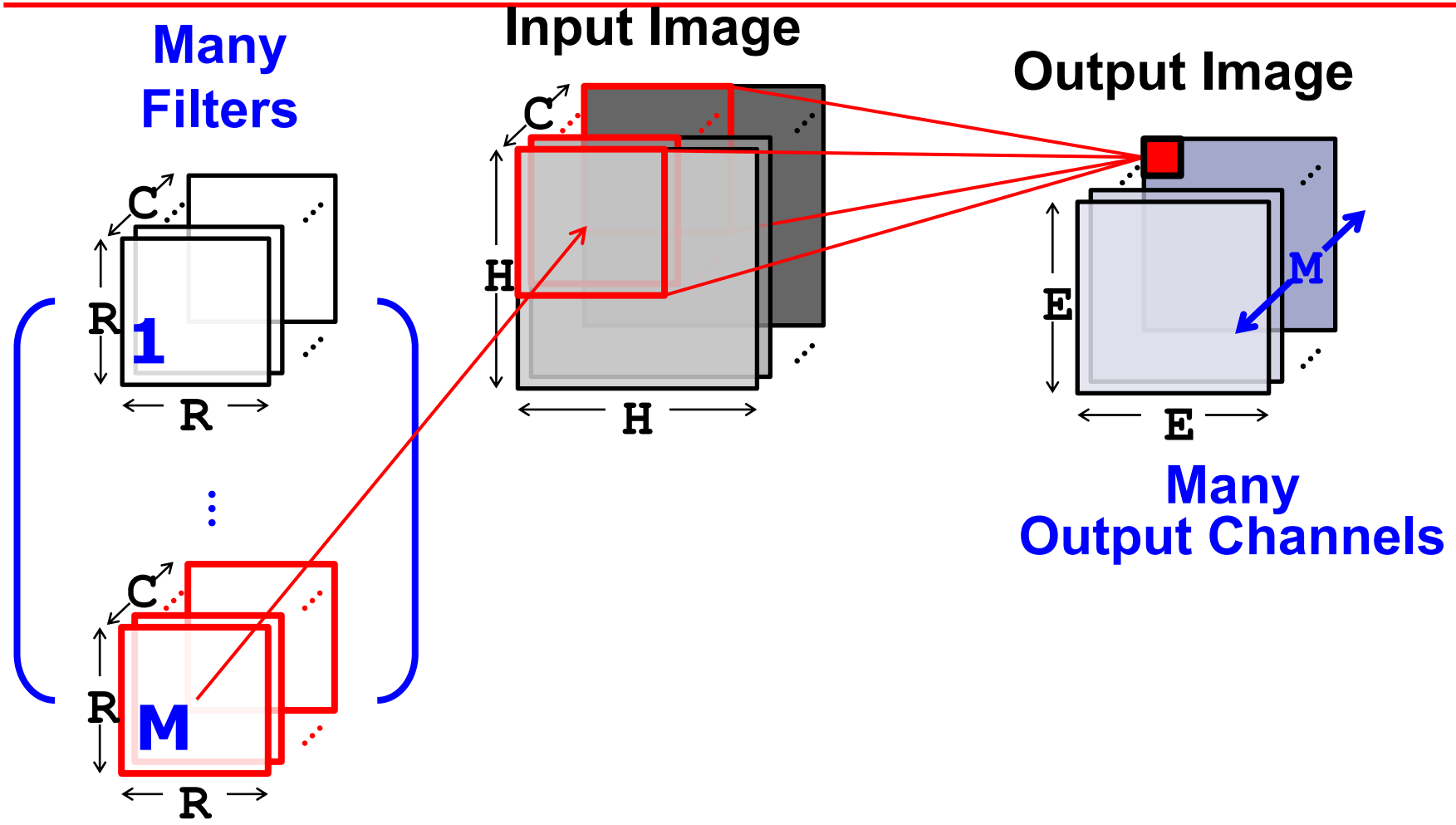
Convolution Operation



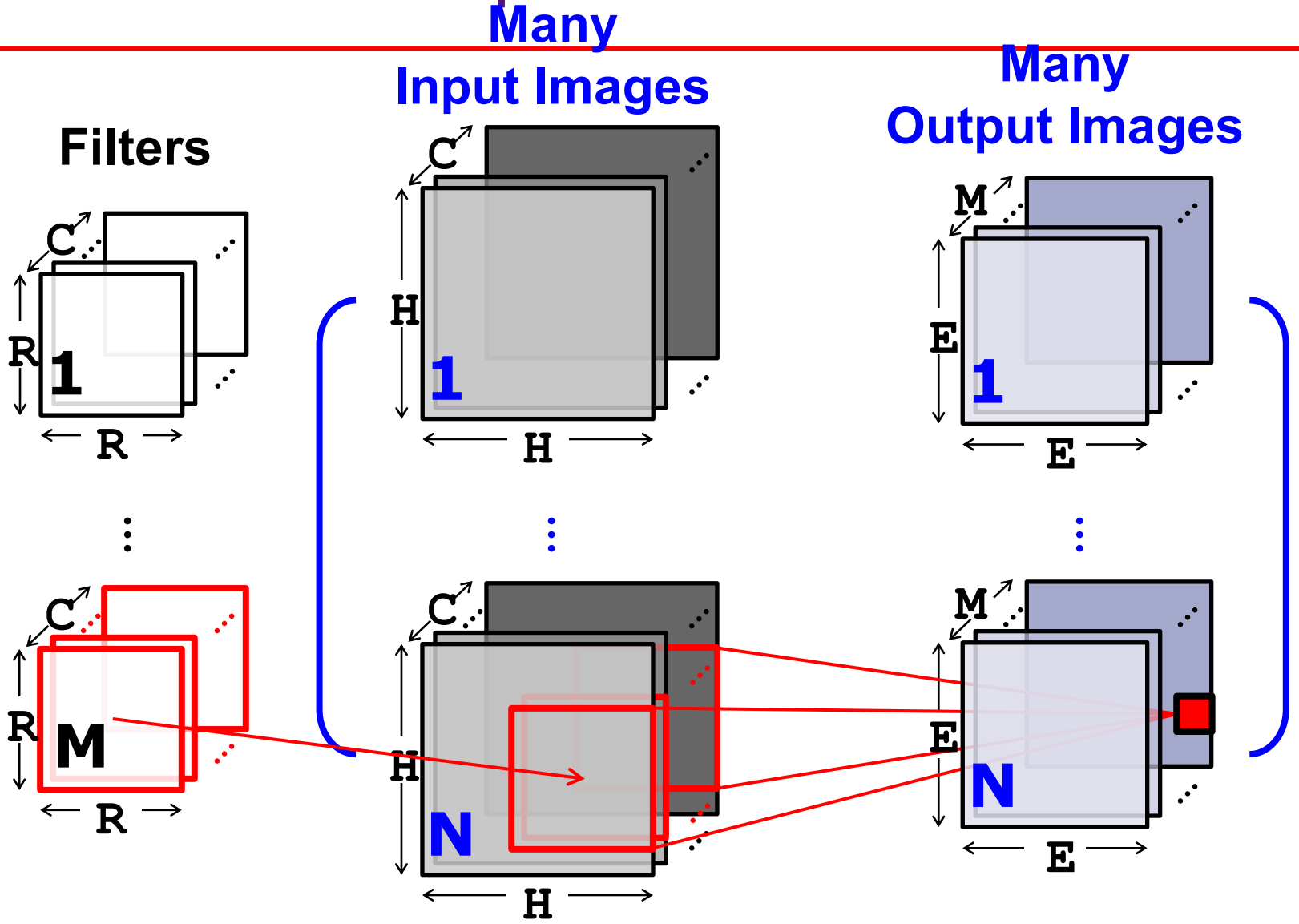
Convolution Operation



Convolution Operation

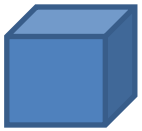


Convolution Operation



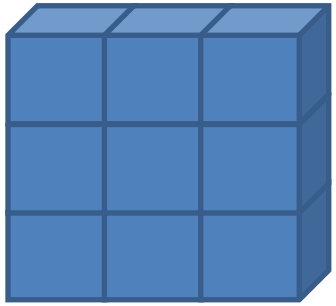
Representation: Tensors

Rank-0 - Scalar

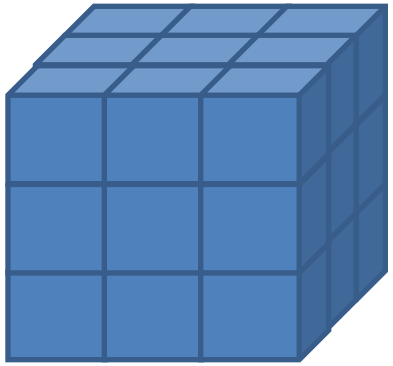


Rank-1 - Vector

Rank-2 - Matrix

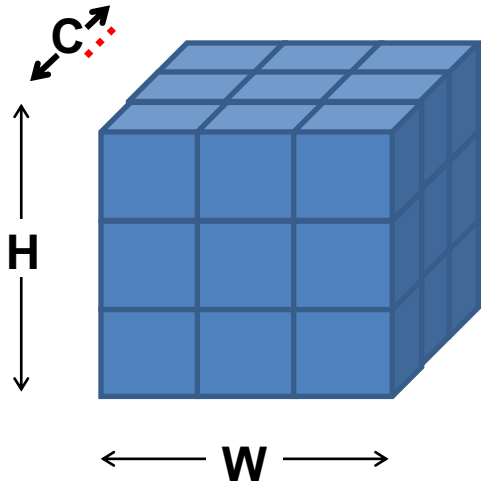


Rank-3 - Cube



Example: Input Act/Fmap Tensor

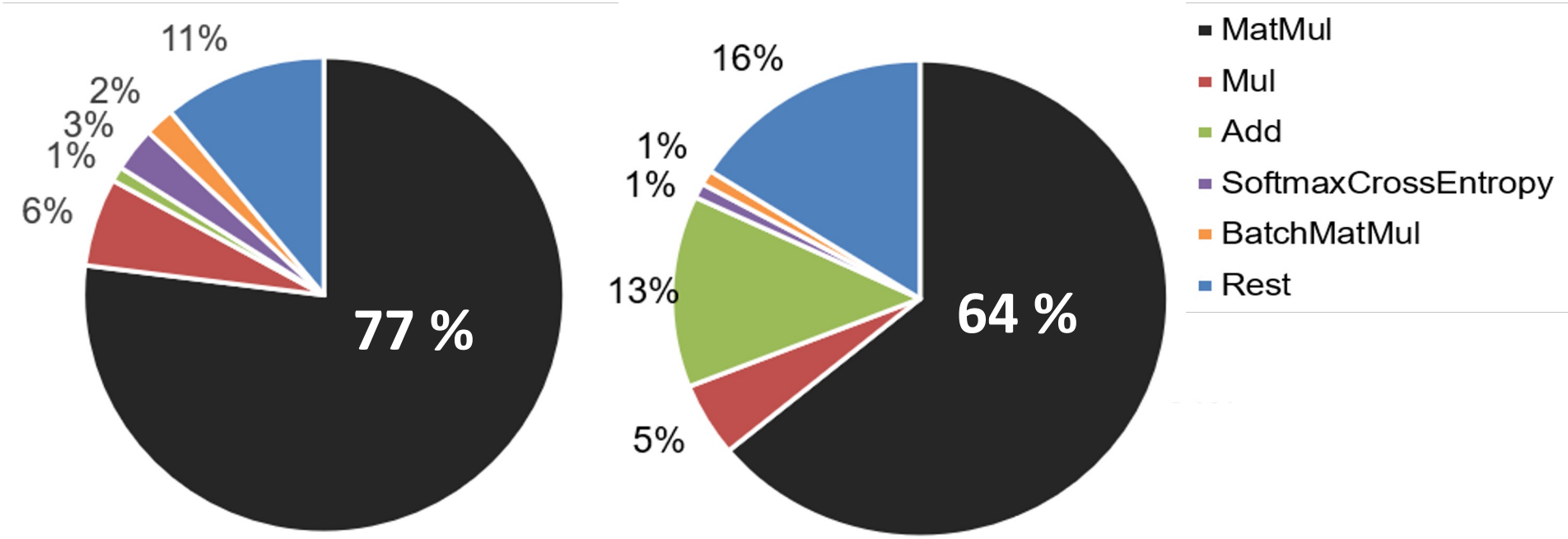
input fmap



The compiler "lowers" high-rank tensors into appropriate HW structures

I[C][H][W]

What to accelerate?



Transformer (Language Understanding)

GNMT (Machine Translation)

Runtime breakdown on V100 GPU

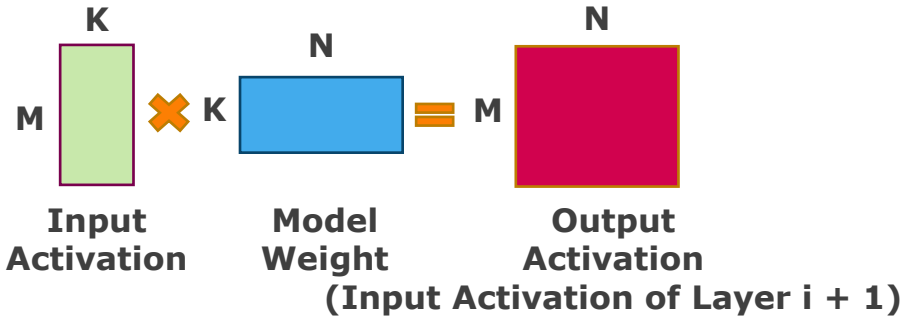
Matrix multiplications (GEMMs) consume around **70%** of the total runtime on modern deep learning workloads.

Prime candidate for acceleration

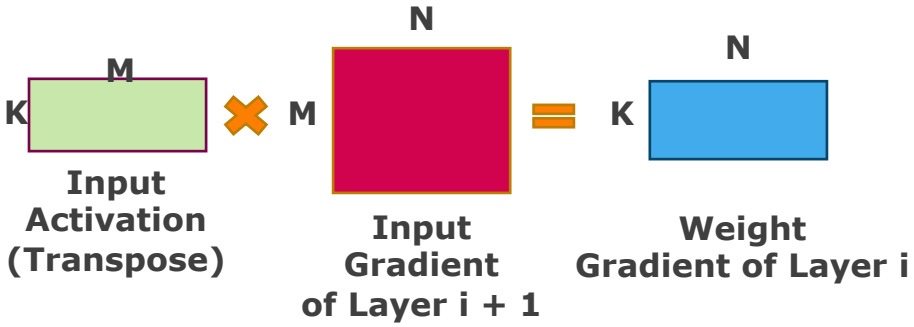
Possible Pitfall?

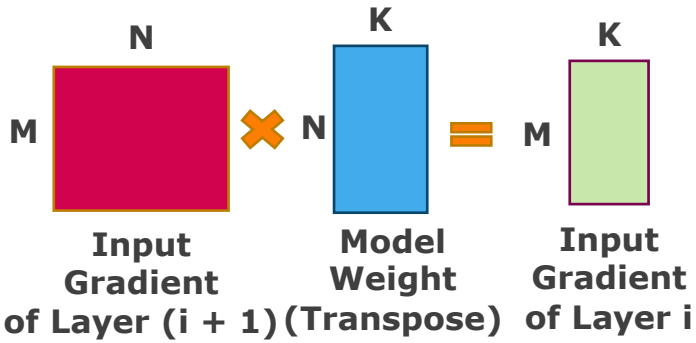
Beware of Amdahl's Law

GEMMs in Deep Learning



Forward Pass



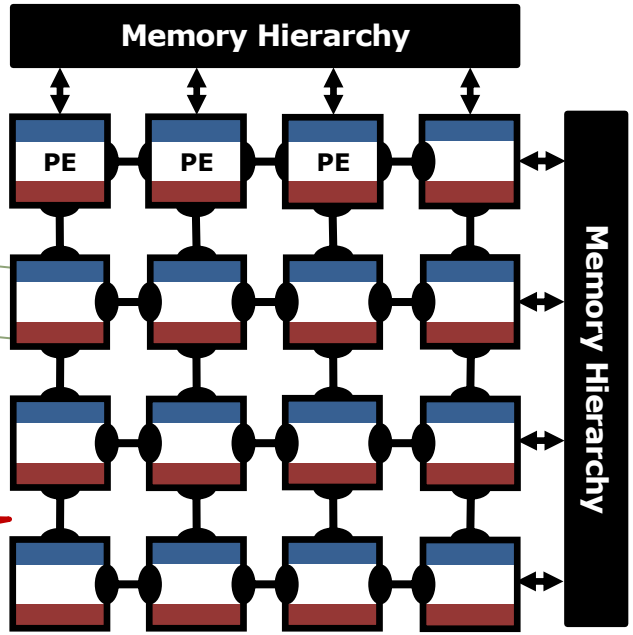


Backward Pass

Spatial (or Dataflow) Accelerators

- **Millions of Parameters (i.e., weights)**
  - Billions of computations

Spread computations across thousands of PEs (i.e., **parallelism**)



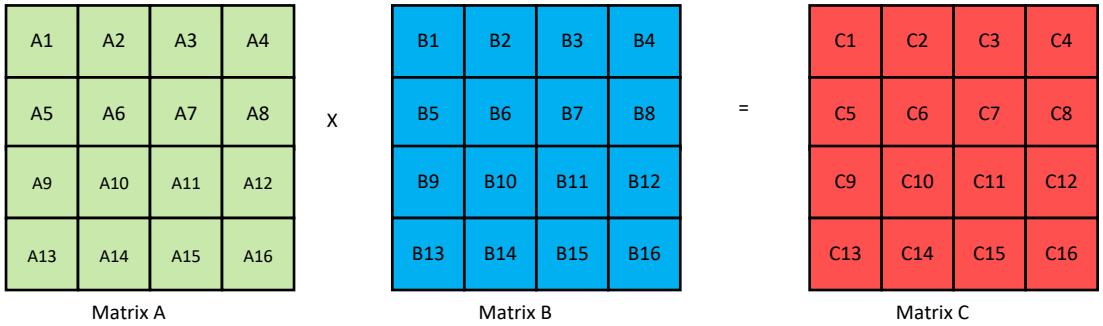
- Heavy data movement

Reuse data within the array via dataflow.

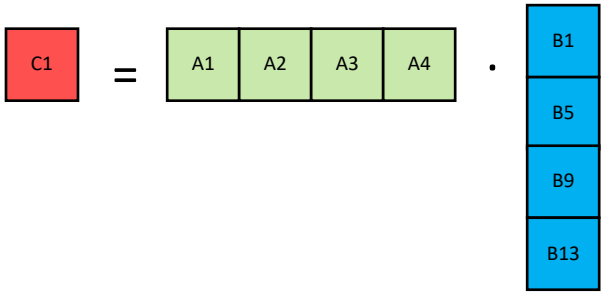
Features

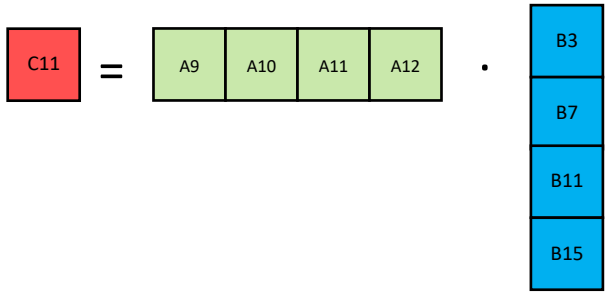
- Thousands of Processing Elements (PEs)
- Custom Memory Hierarchy (typically scratchpads, no caches)
- Custom NoCs

What is Reuse?



Example Operations





C1 = A1***B1** + A2***B5** + A3***B9** + A4***B13**

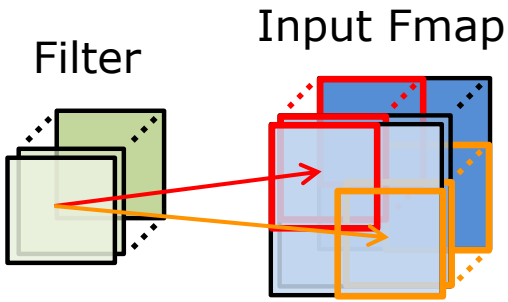
C11 = A9*B3 + A10*B7 + A11*B11 + A12*B15

C5 = A5***B1** + A6***B5** + A7***B9** + A8***B13**

Examples of Data Reuse in DNN

Convolutional Reuse

CONV layers only
(sliding window)

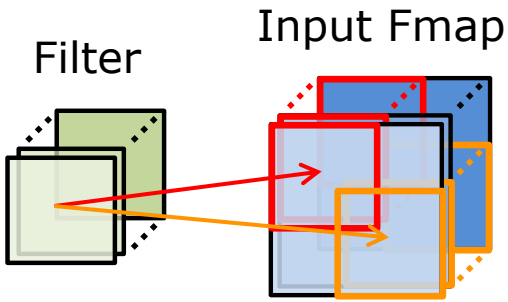


Reuse: Activations
Filter weights

Examples of Data Reuse in DNN

Convolutional Reuse

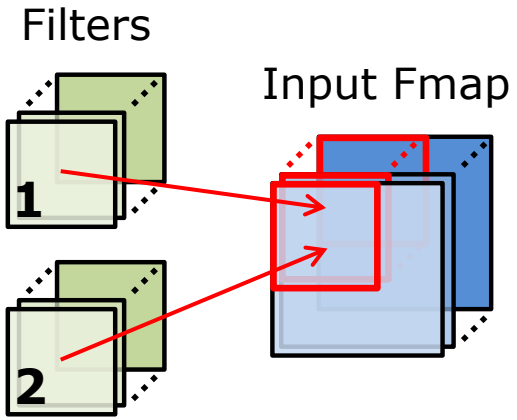
CONV layers only
(sliding window)



Reuse: Activations
Filter weights

Fmap Reuse

CONV and FC layers

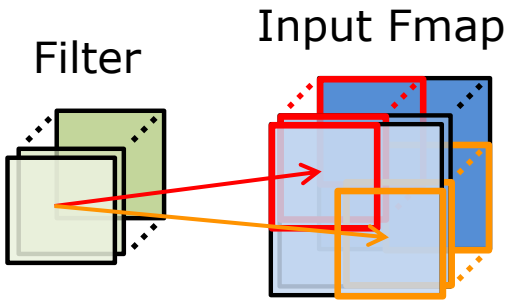


Reuse: Activations

Examples of Data Reuse in DNN

Convolutional Reuse

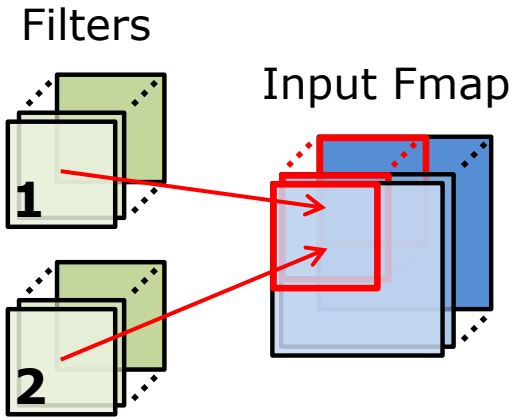
CONV layers only (sliding window)



Reuse: Activations, Filter weights

Fmap Reuse

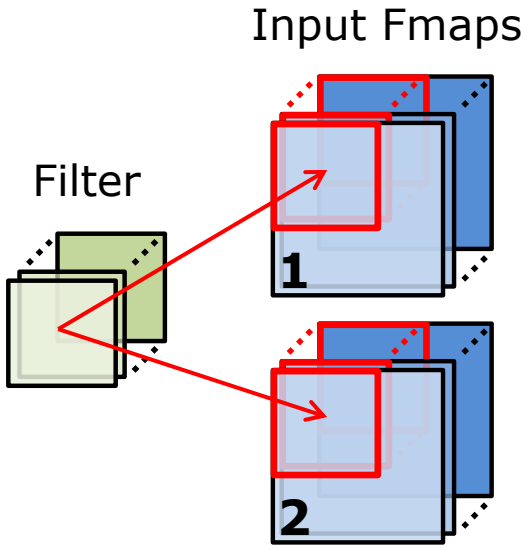
CONV and FC layers



Reuse: Activations

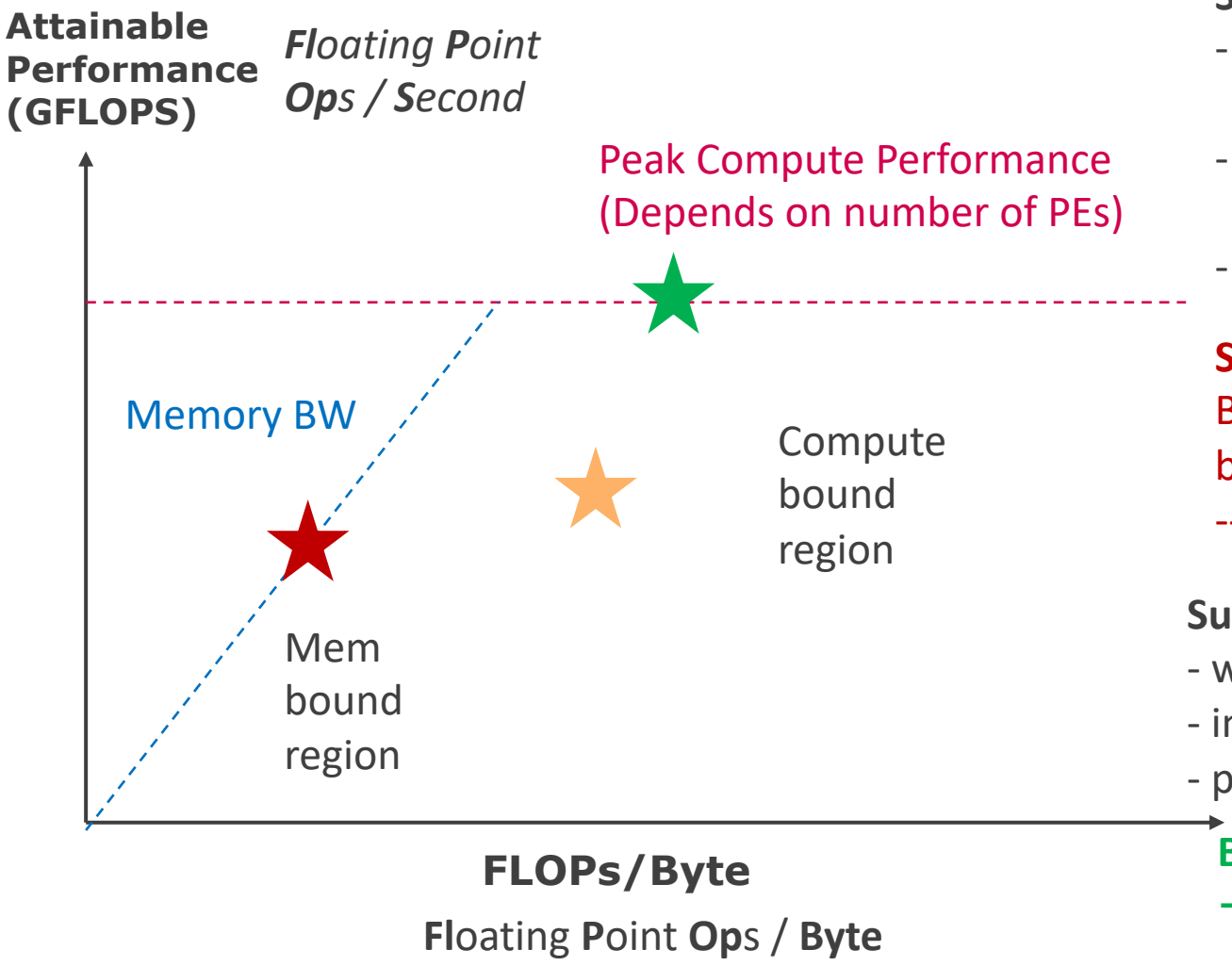
Filter Reuse

CONV and FC layers (batch size > 1)



Reuse: Filter weights

Why does reuse help?



Suppose

- 100 PEs operating at 1 GHz => 100 GFLOPs/sec
- Each PE needs 2 bytes of read and 1 byte of write.
- DRAM BW ~25 GBps

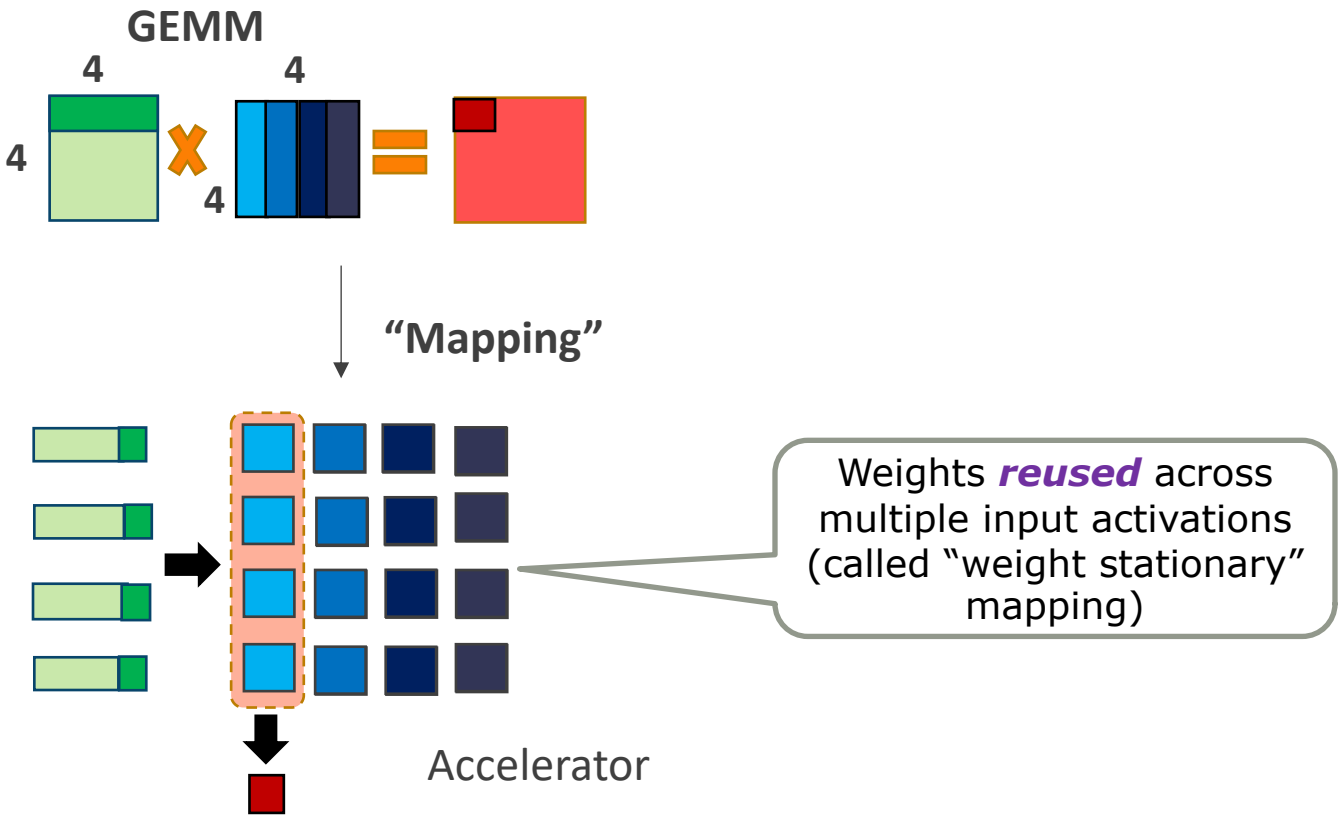
Simple Accelerator:
BW requirement: 3 bytes/cycle => 300 GBps
--> Mem Bound!

Suppose we have data reuse
- weight reused completely
- input reused for 10 cycles
- psums reused for 10 cycles

BW requirement: 20 GBps
→ Compute Bound

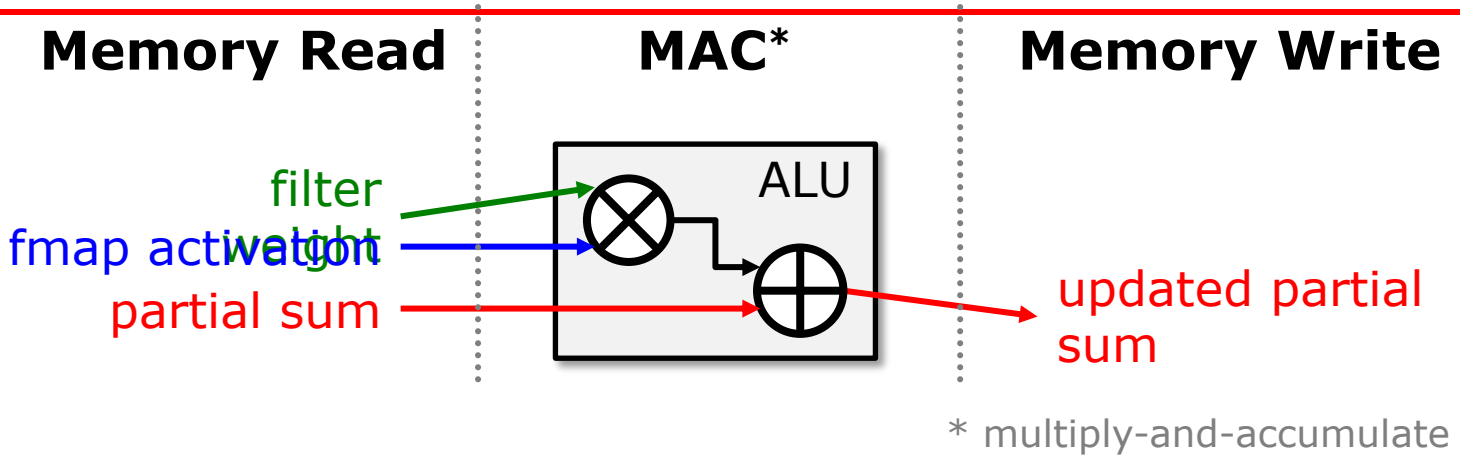
Realistic?

How to exploit Reuse?

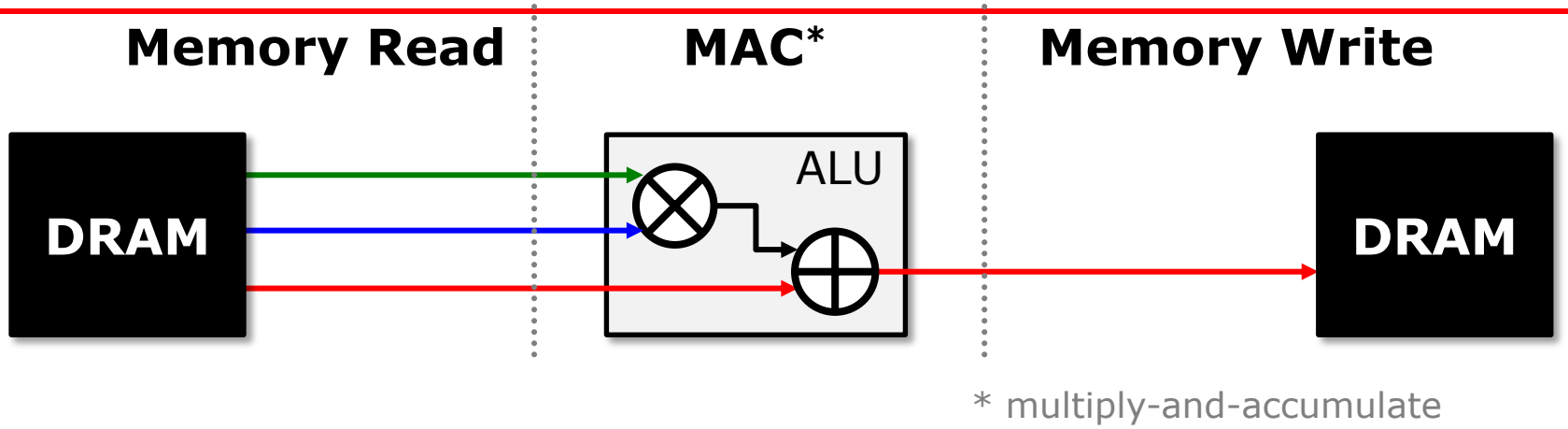


What HW structures would you need?

Building a DNN Accelerator



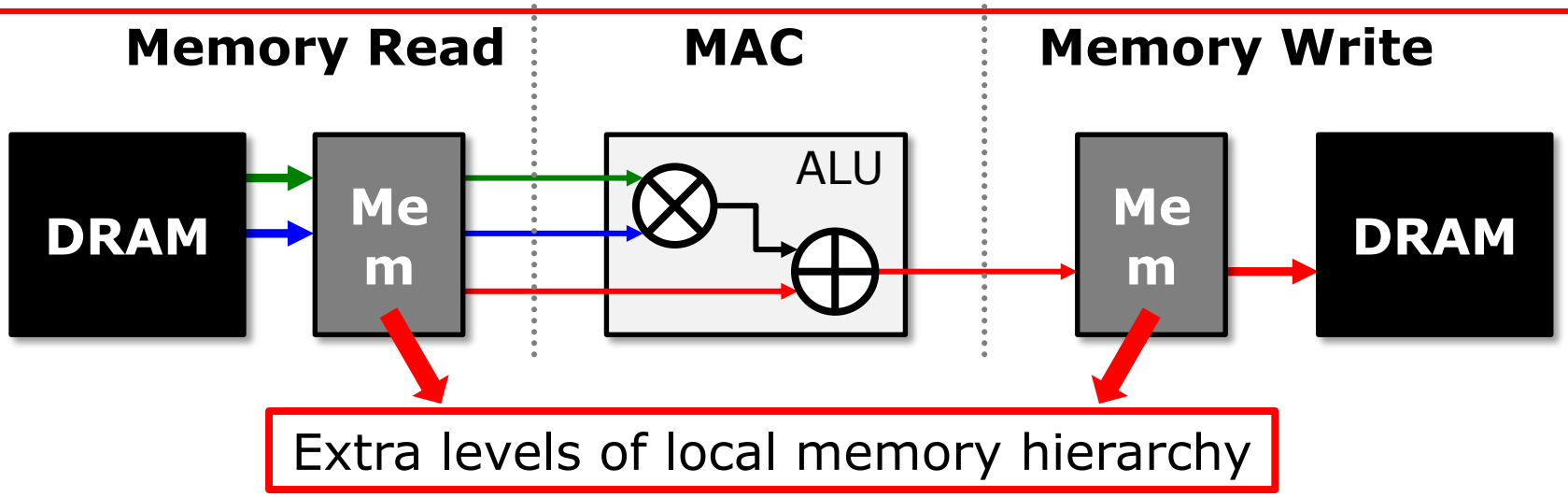
Building a DNN Accelerator



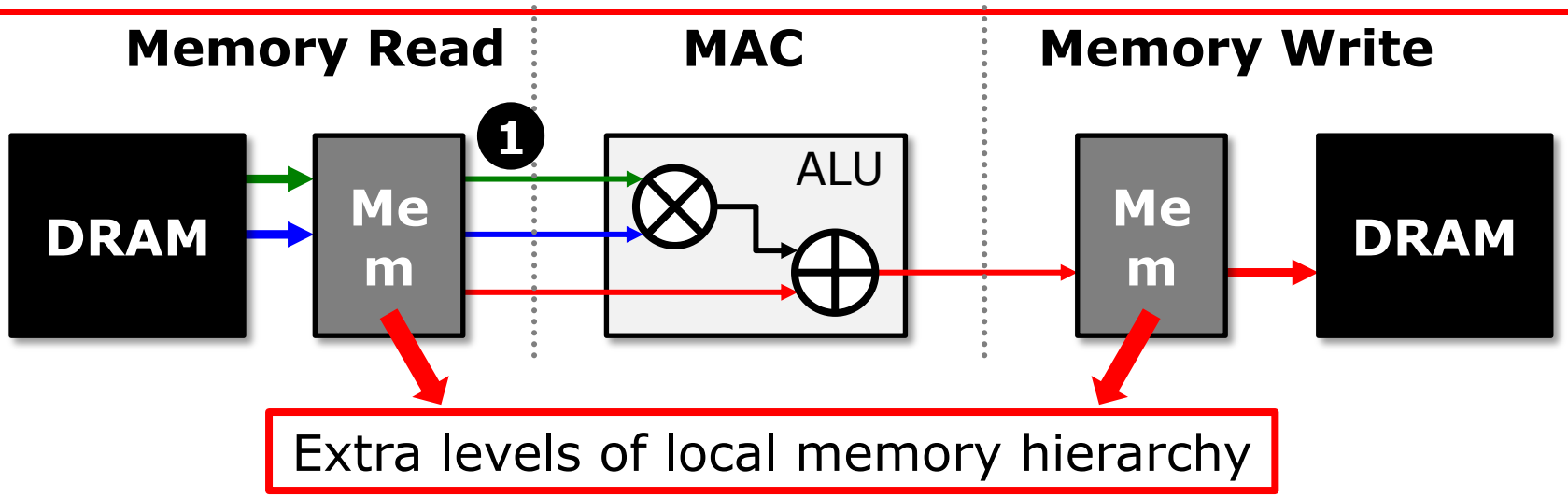
Worst Case: all memory R/W are **DRAM** accesses

- Example: AlexNet [NeurIPS 2012] has **724M** MACs
  → **2896M** DRAM accesses required

Building a DNN Accelerator

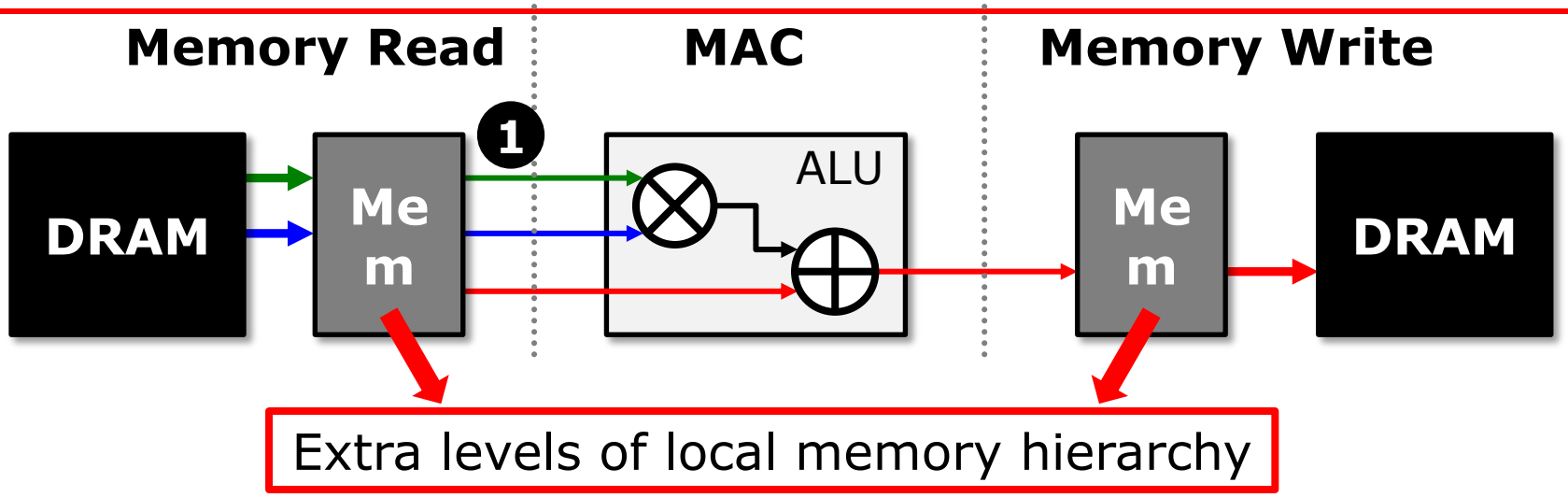


Building a DNN Accelerator



Opportunities: 1 **data reuse**

Building a DNN Accelerator

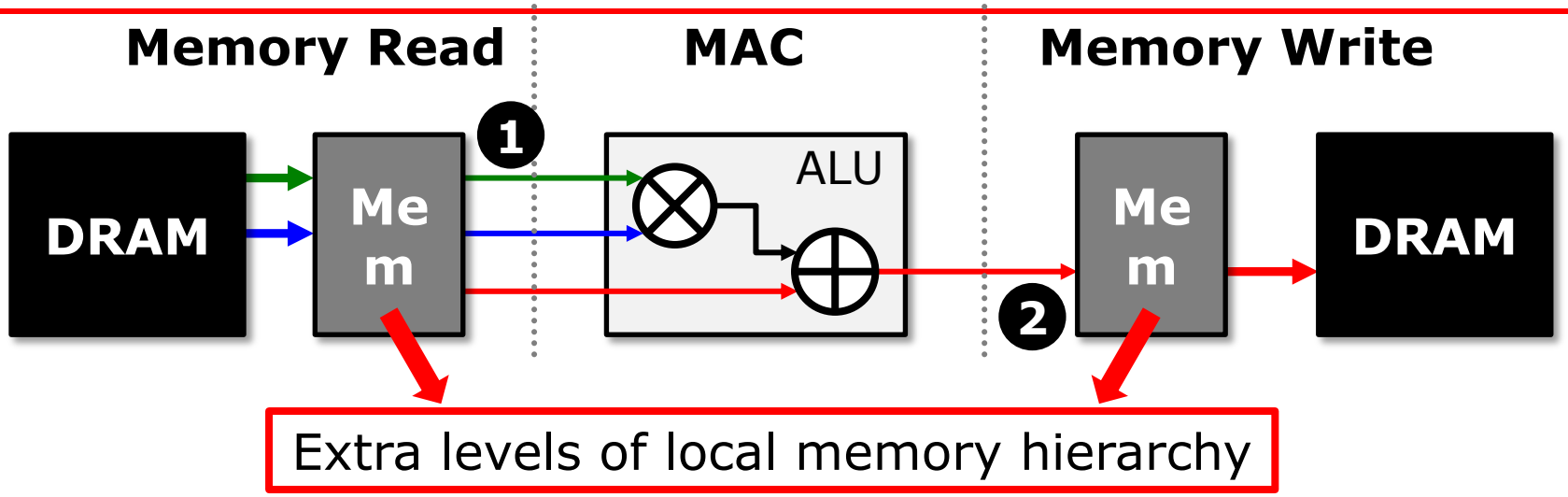


Opportunities: 1 **data reuse**

1 Can reduce DRAM reads of filter/fmap by up to **500×****

** AlexNet CONV layers

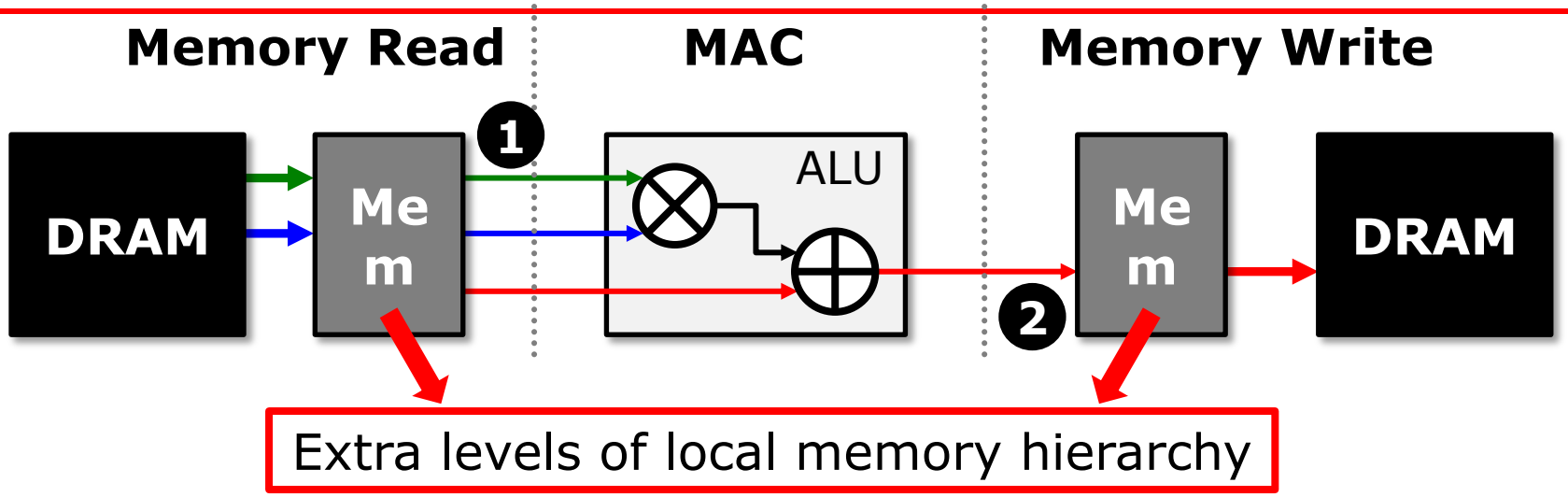
Building a DNN Accelerator



Opportunities: 1 **data reuse** 2 **local accumulation**

1. Can reduce DRAM reads of filter/fmap by up to **500×**
2. Partial sum accumulation does **NOT** have to access DRAM

Building a DNN Accelerator

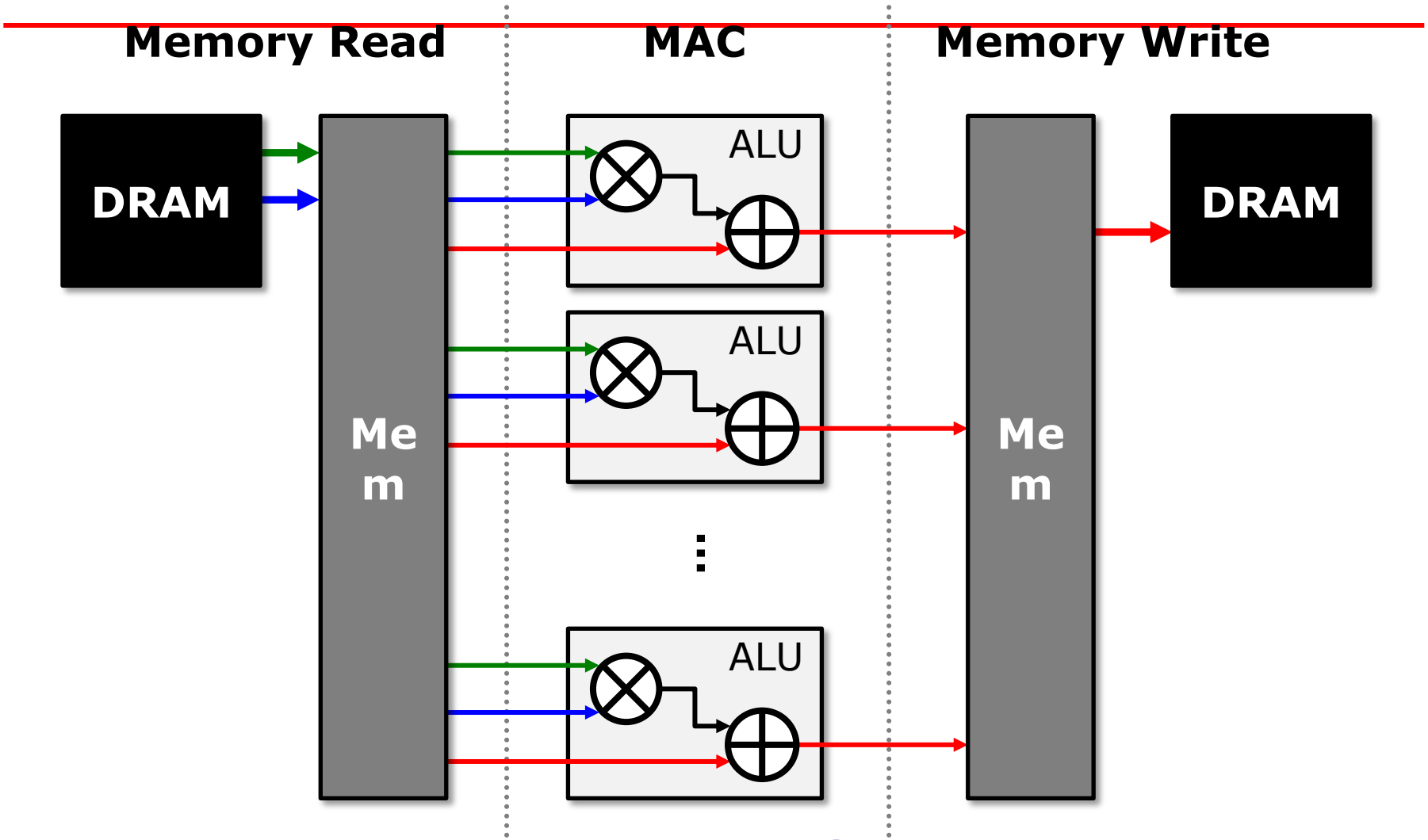


Opportunities: 1 **data reuse** 2 **local accumulation**

1. Can reduce DRAM reads of filter/fmap by up to **500×**
2. Partial sum accumulation does **NOT** have to access DRAM

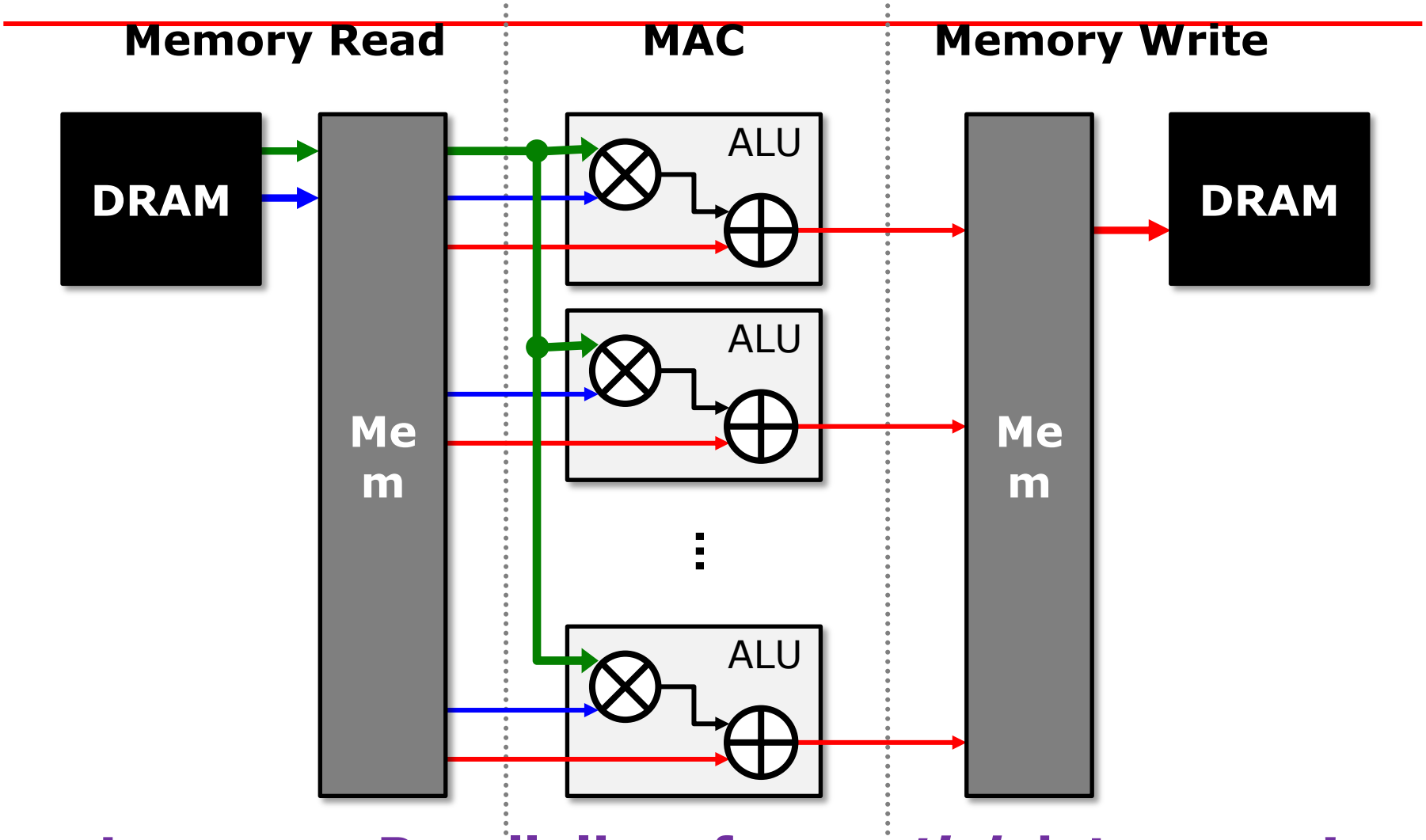
- Example: DRAM access in AlexNet can be reduced from **2896M** to **61M** (best case)

Building a DNN Accelerator



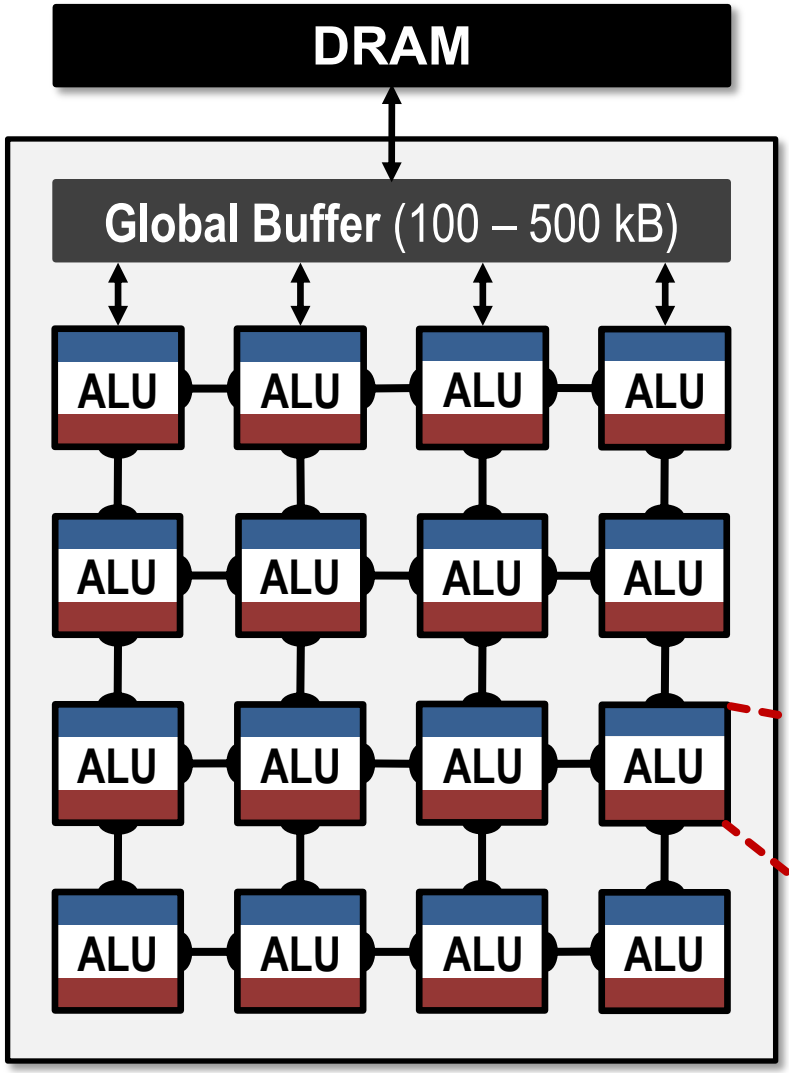
Leverage Parallelism for Throughput!

Building a DNN Accelerator



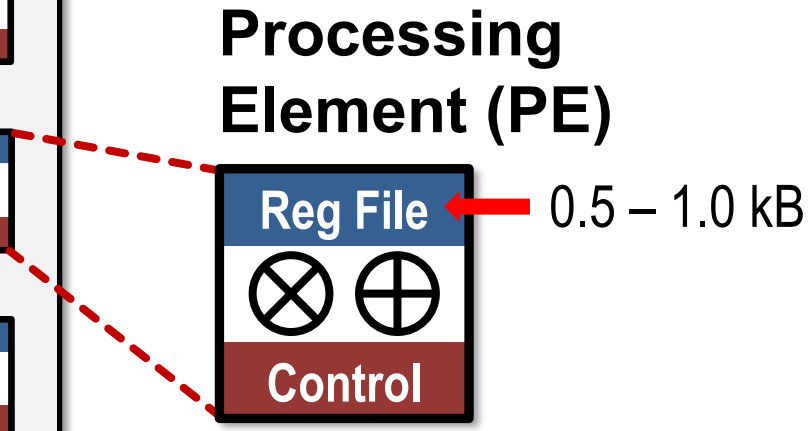
Leverage Parallelism for *spatial* data reuse!

Spatial DNN Accelerator

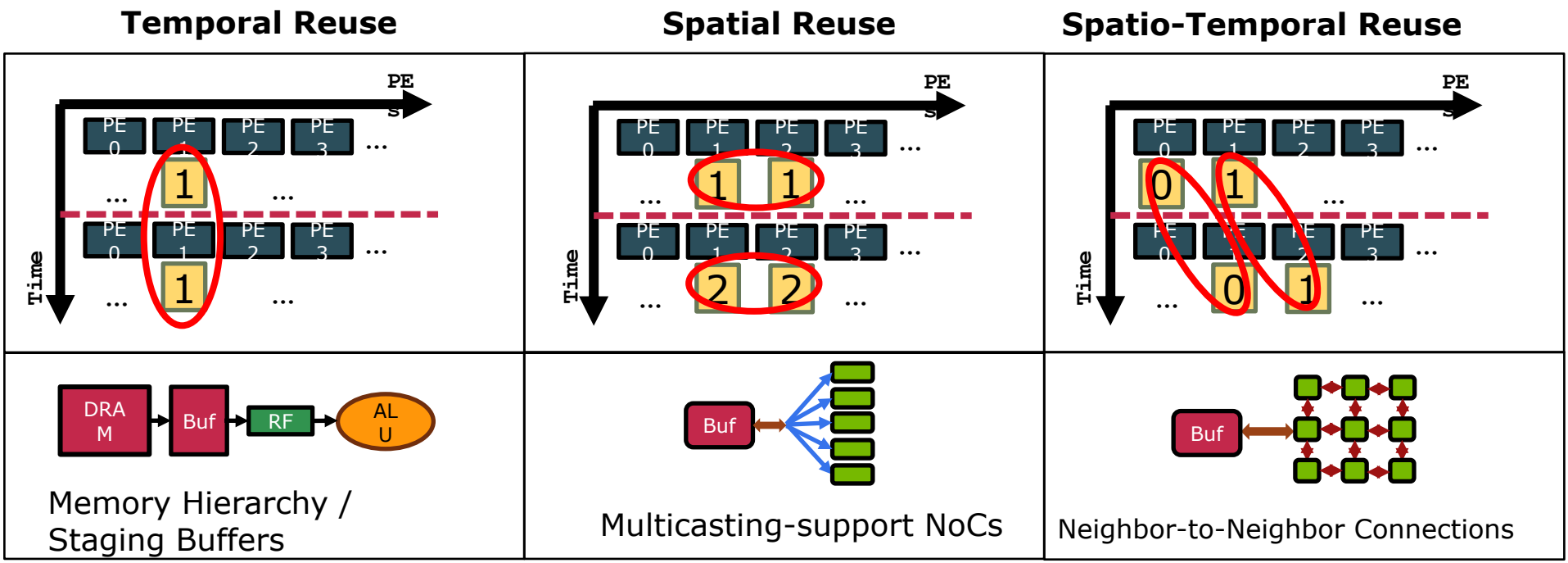


Local Memory Hierarchy

- Global Buffer
- Direct inter-PE network
- PE-local memory (RF)



Hardware structures to exploit reuse



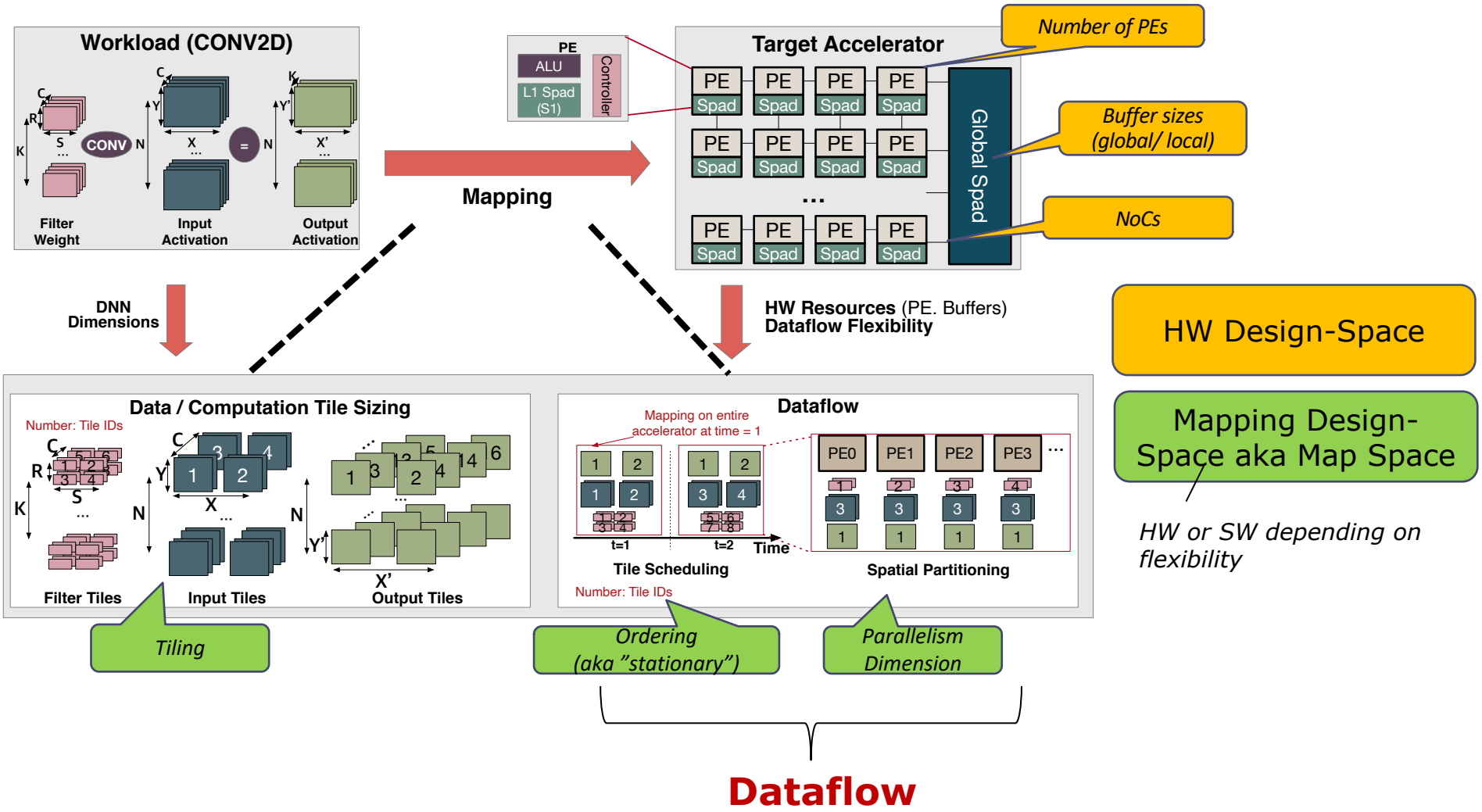
E.g., Custom memory hierarchies in accelerators.

E.g., Hierarchical Bus in Eyeriss (ISCA 2016), Tree in MAERI (ASPLOS 2018)

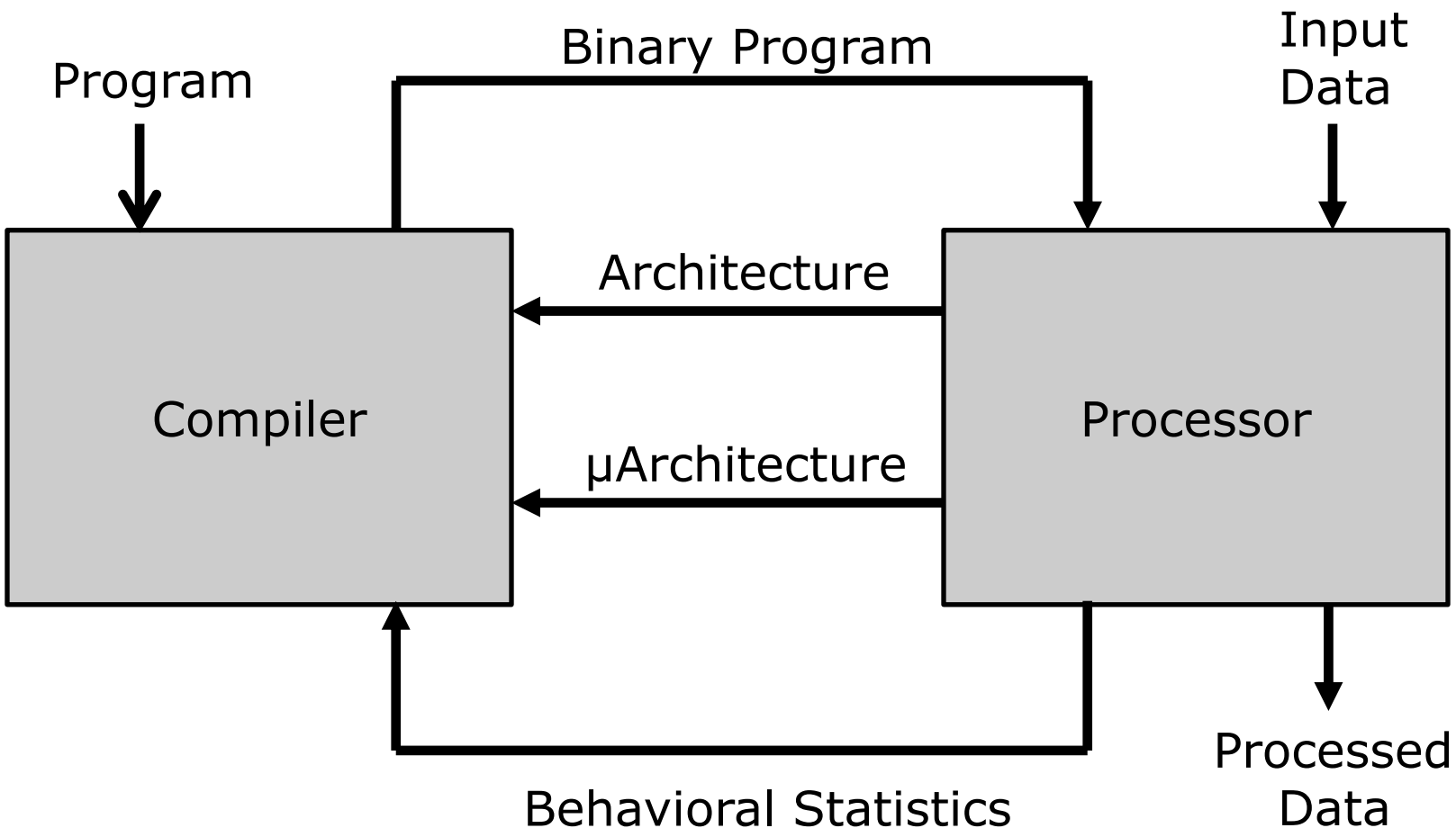
E.g., TPU (ISCA 2017), local network in Eyeriss (ISCA 2016)

The availability of the hardware structure limits the "mapping-space"

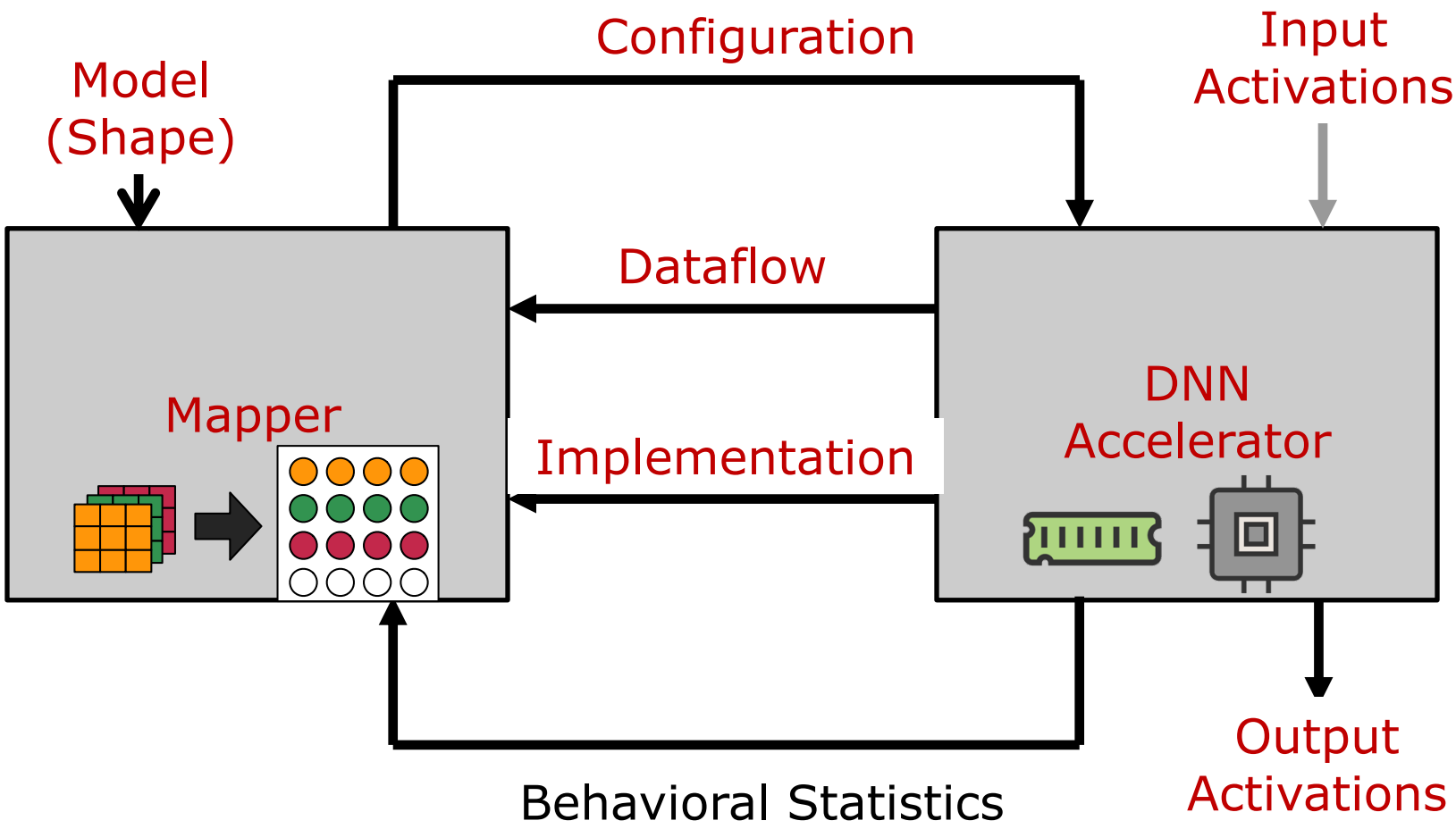
Design-space of a DNN Accelerator



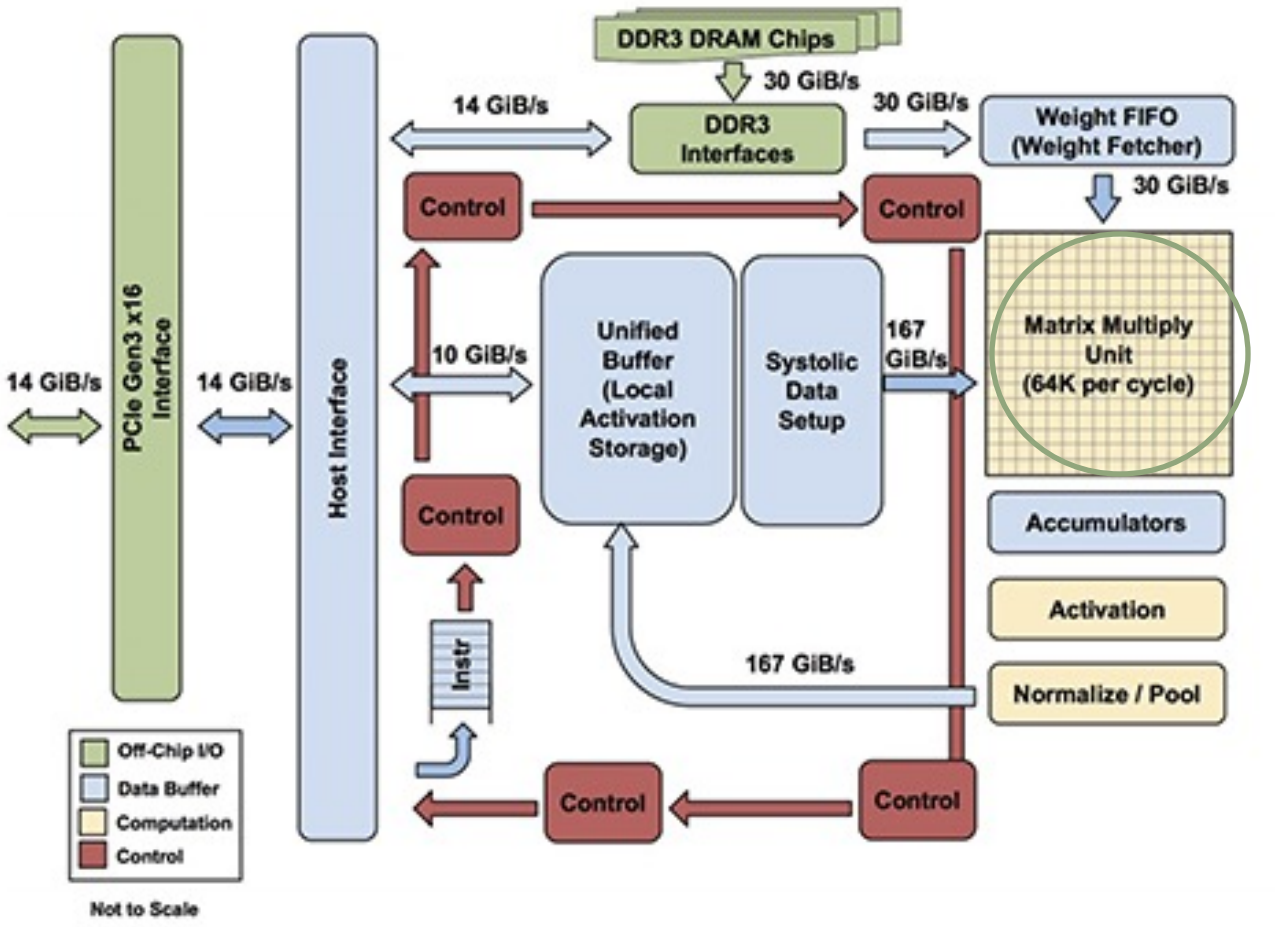
CPU Compute Model



DNN Compute Model

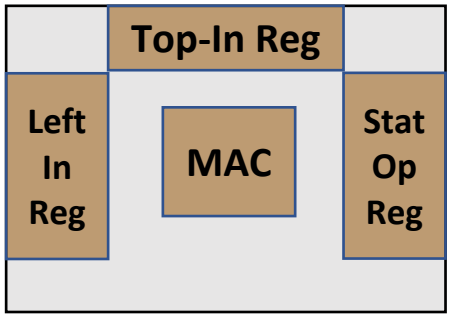


Putting it all together: Case Study of Google TPUv1 (2016)

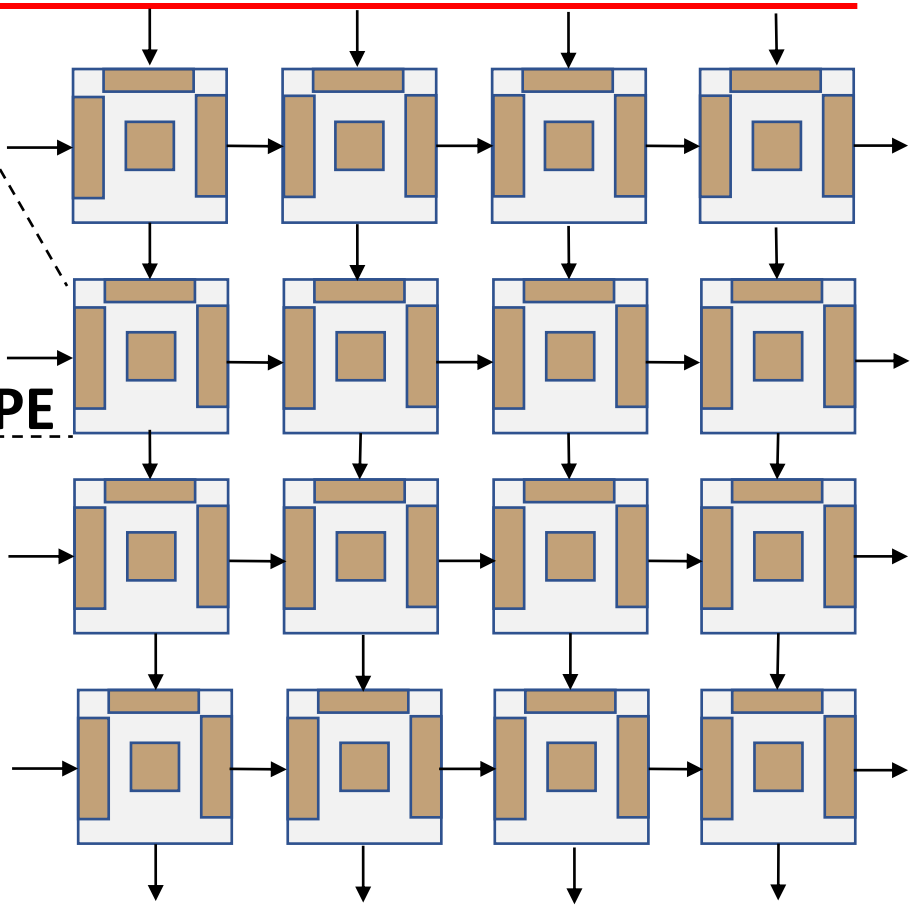


"Systolic Array"

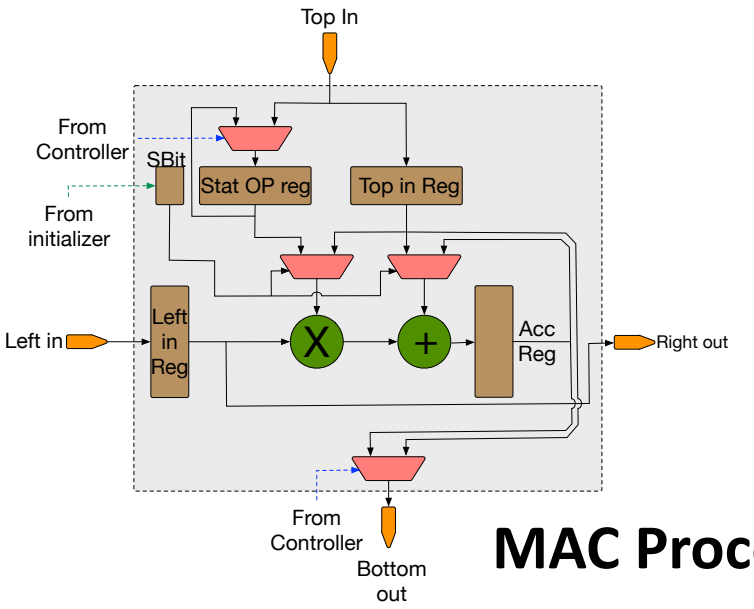
Systolic Array Structure



Schematic of MAC PE



Systolic Array

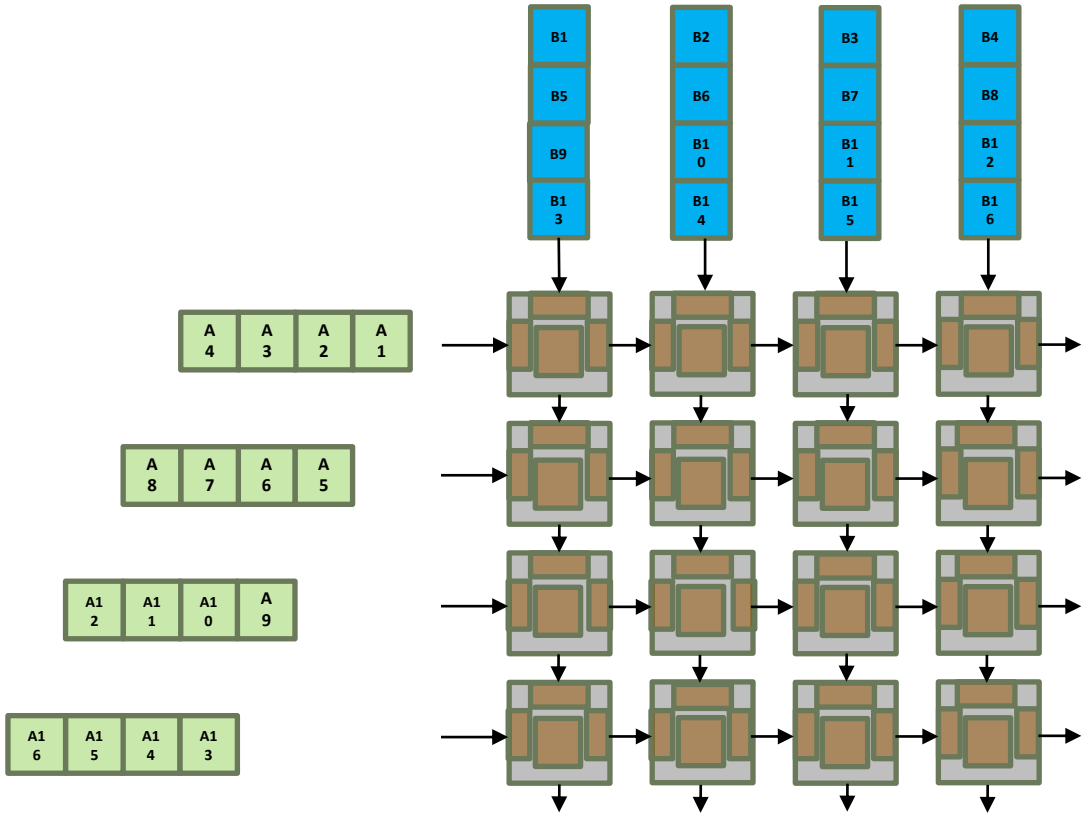


MAC Processing Element (PE)

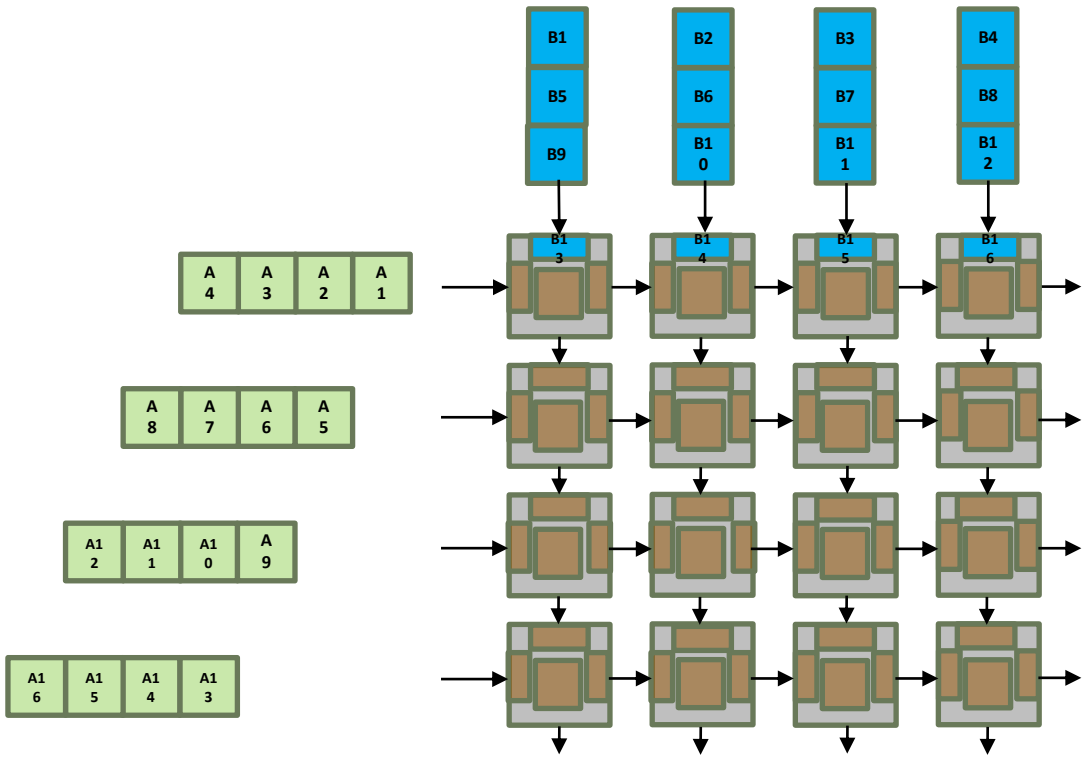
Difference from CPU Architecture

- Registers distributed across PEs
  - One per operand per PE
    - Can me more than one as well
- Operands "forwarded" from one register to the other
  - More energy-efficient than reading large register file
- Stage data through array in deterministic manner
  - No need for hazard checks etc

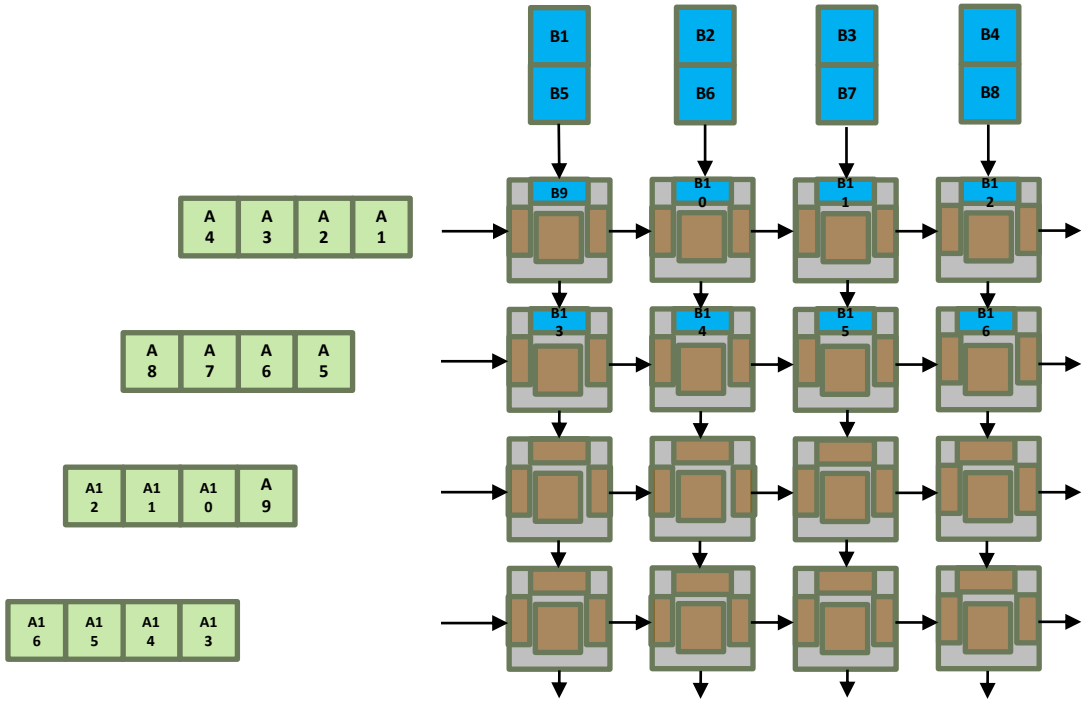
Walkthrough Example



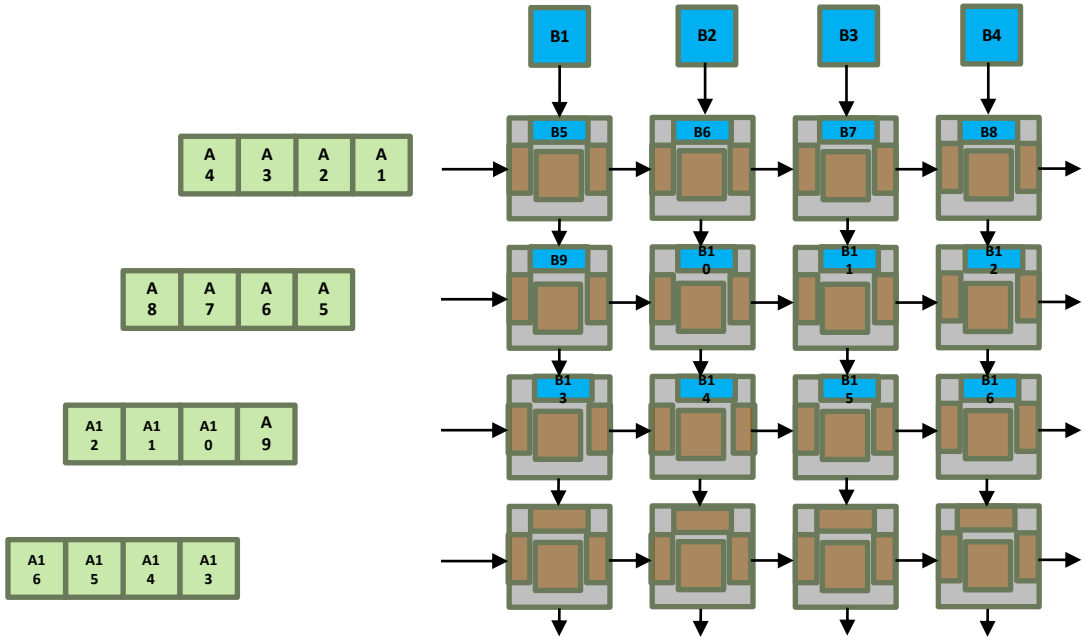
Walkthrough Example



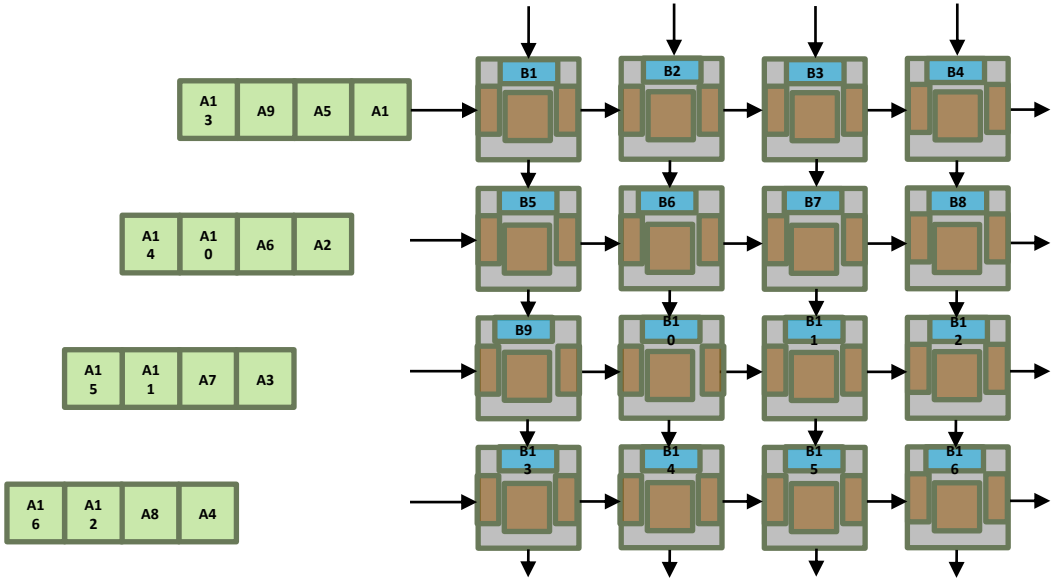
Walkthrough Example



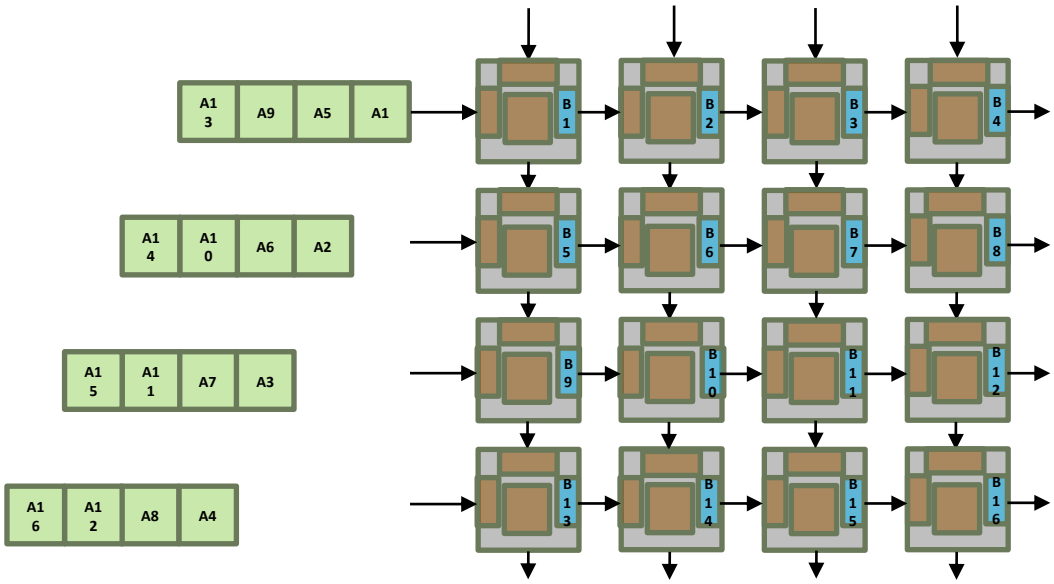
Walkthrough Example



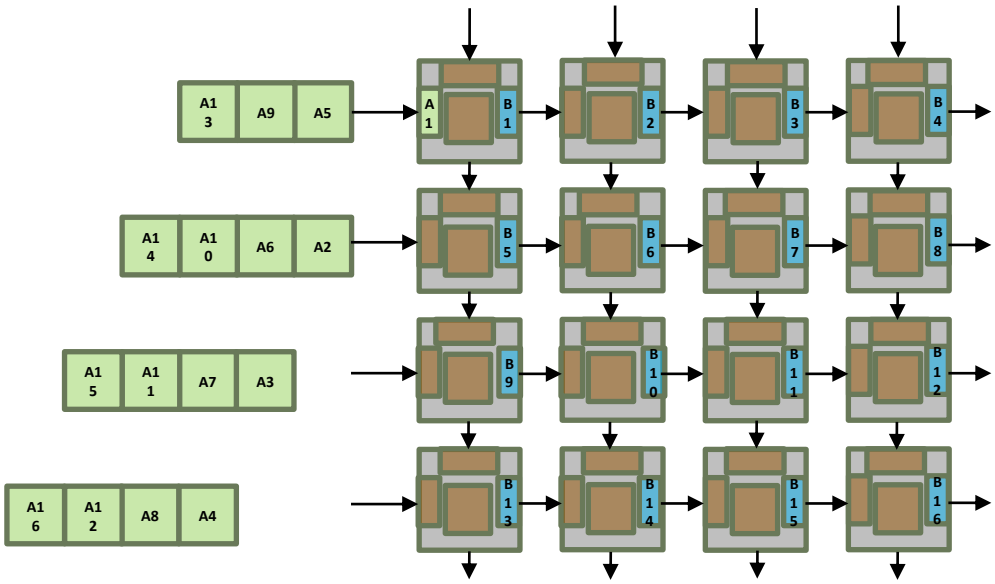
Walkthrough Example



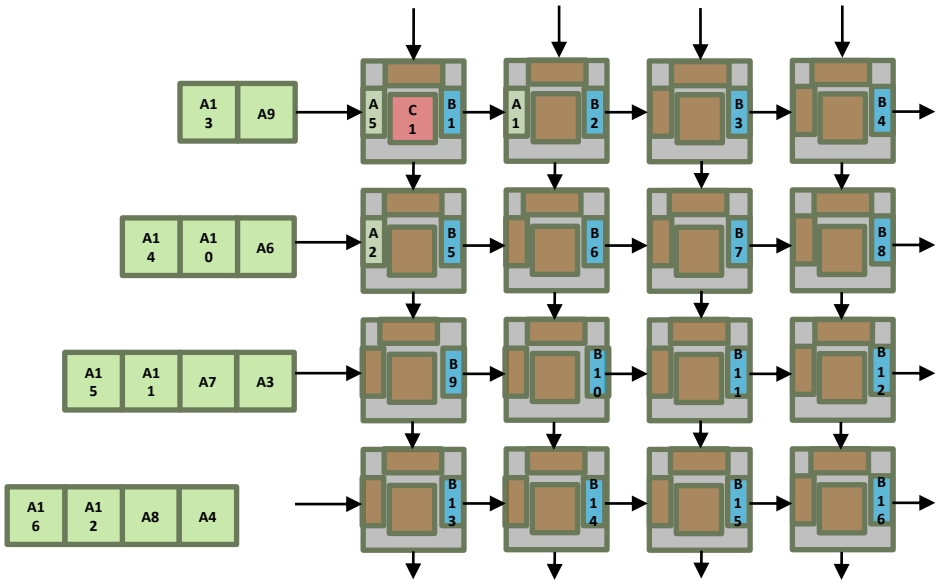
Walkthrough Example



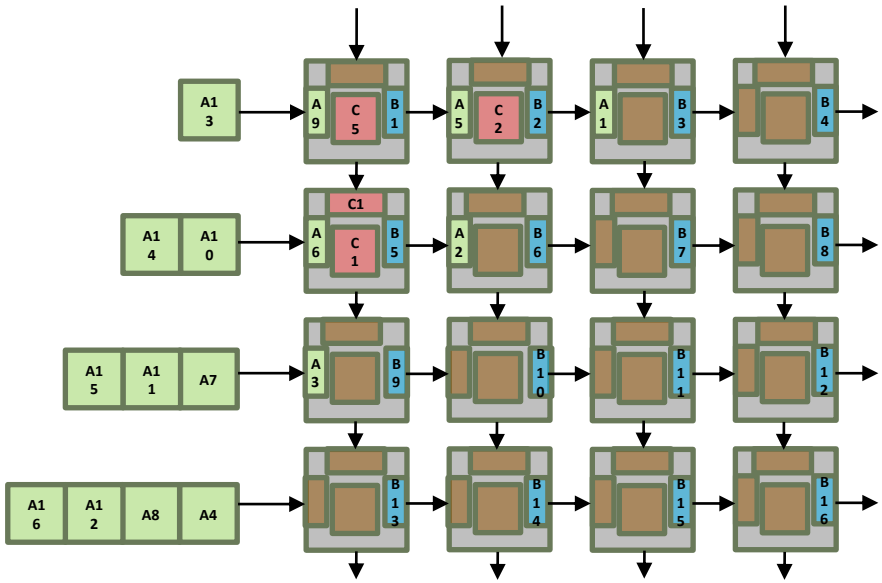
Walkthrough Example



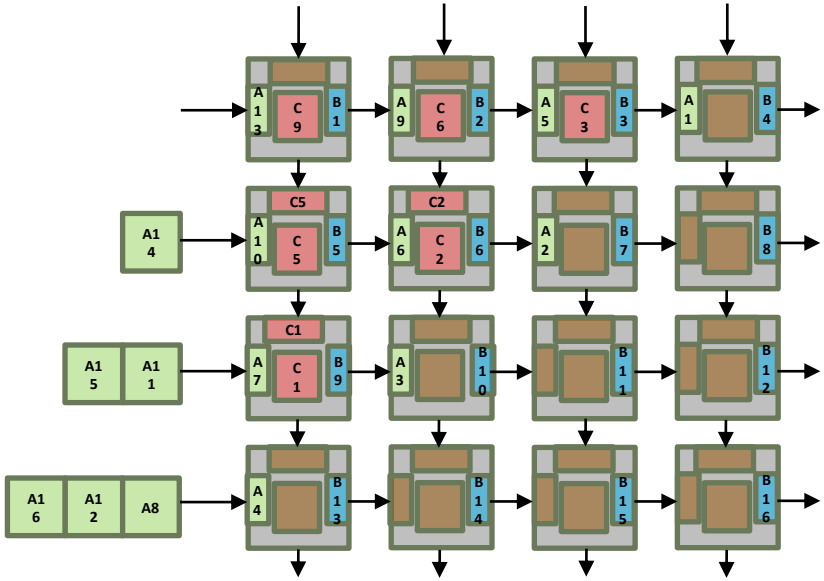
Walkthrough Example



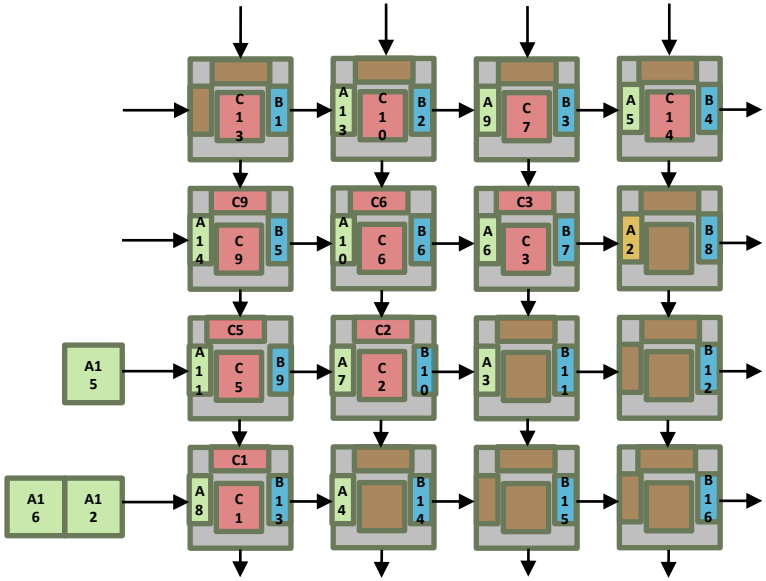
Walkthrough Example



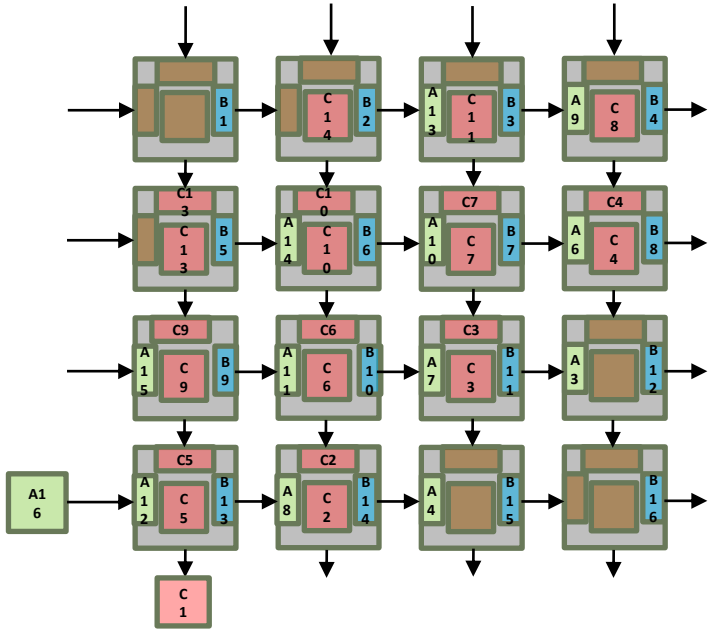
Walkthrough Example



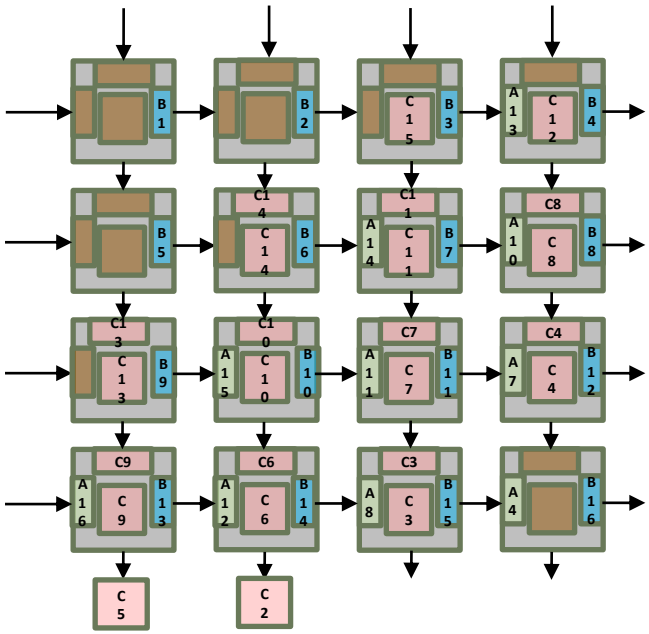
Walkthrough Example



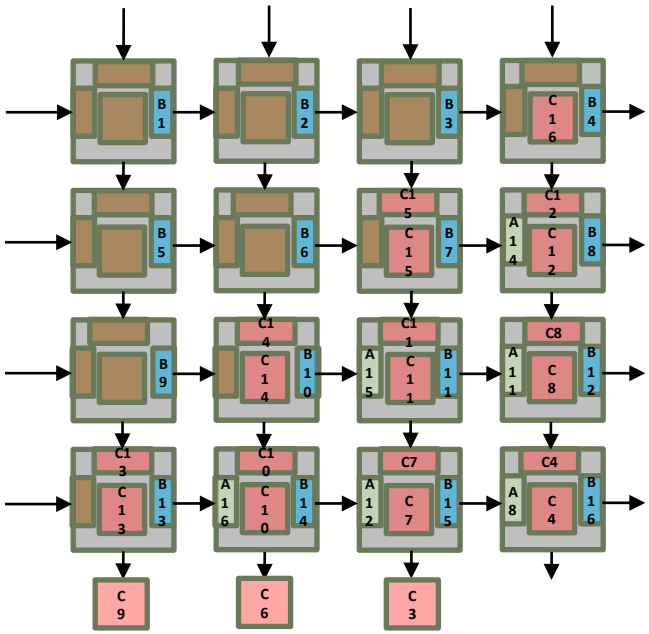
Walkthrough Example



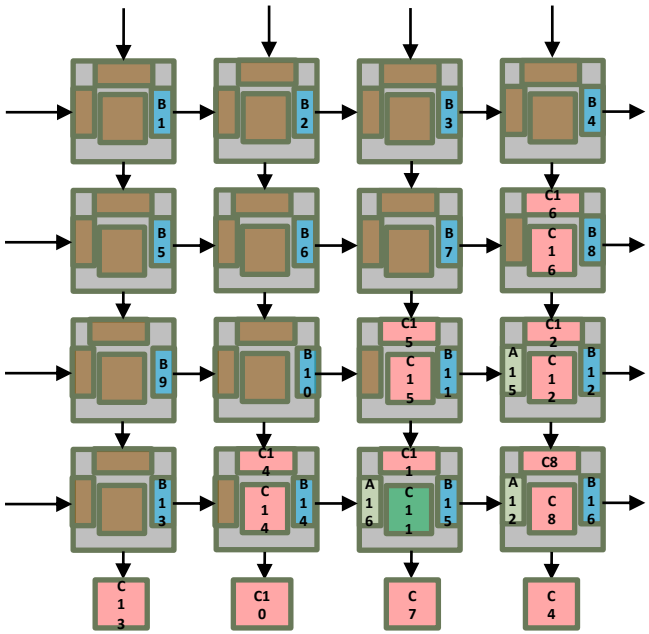
Walkthrough Example



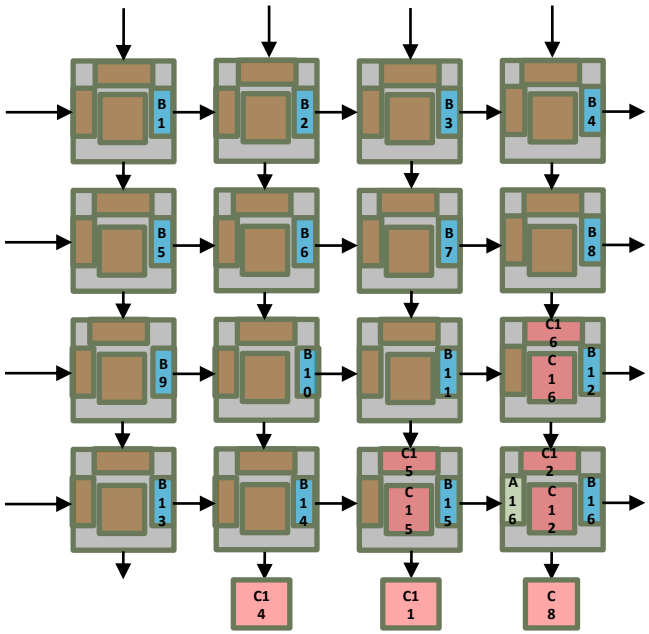
Walkthrough Example



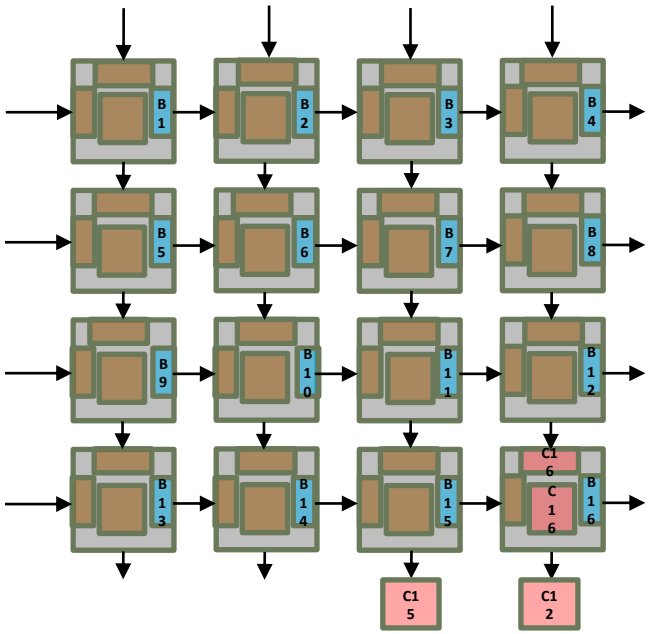
Walkthrough Example



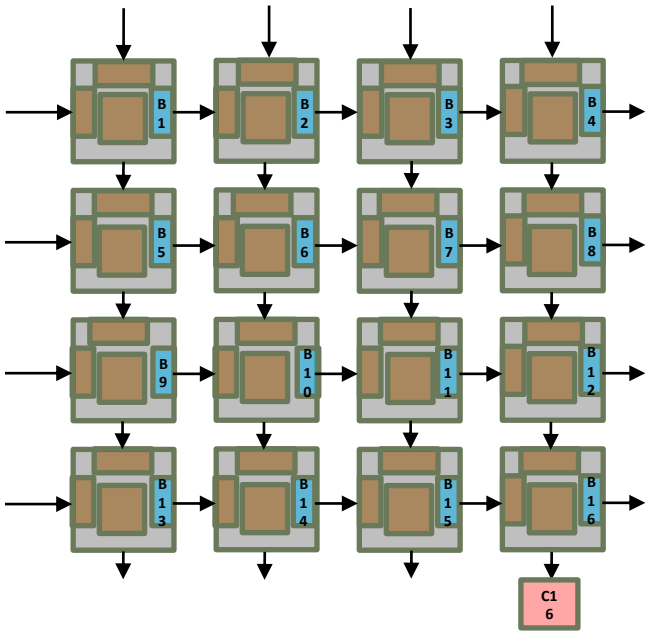
Walkthrough Example



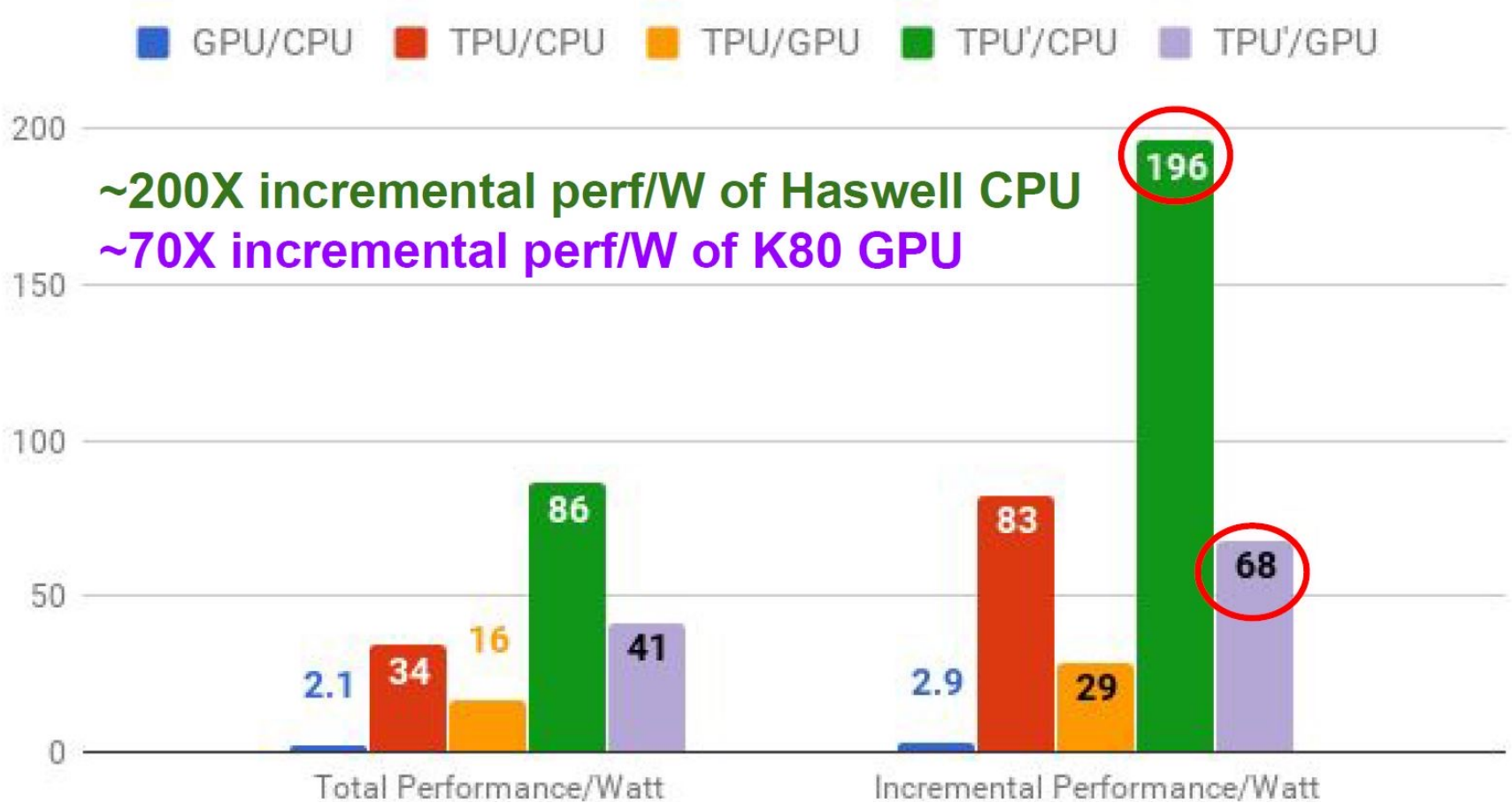
Walkthrough Example



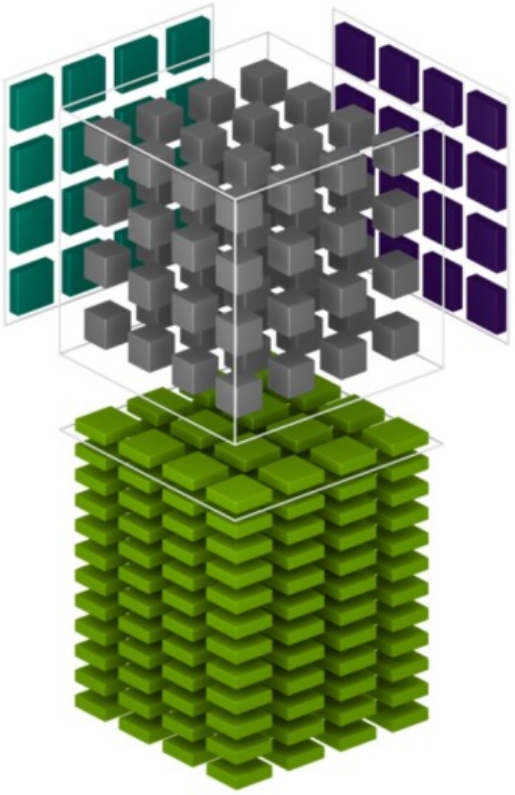
Walkthrough Example

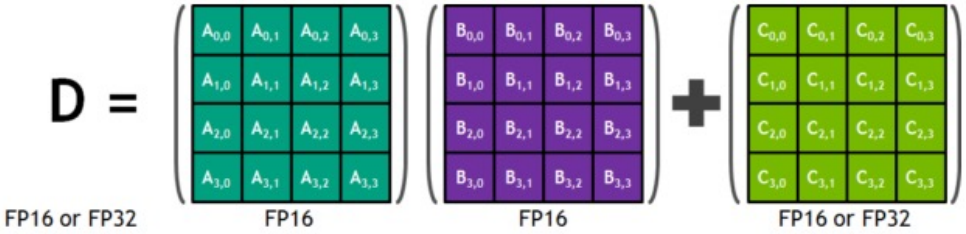


Performance/watt



NVIDIA Response: Tensor Cores





Google TPU – large tensor engine vs NVIDIA: multiple tiny tensor engines.
Trade-off?

Summary

- High Throughput requirements and Energy costs of Data Movement are key drivers towards the trend towards custom accelerators
- Heavy HW-SW Co-Design is used in practice to design accelerators
- Open questions: how to avoid “over”-specialization

Thank you!

Next Lecture: Accelerators-II