6.5930/1

Hardware Architectures for Deep Learning

Introduction and Applications

February 2, 2026

Joel Emer and Vivienne Sze

Massachusetts Institute of Technology
Electrical Engineering & Computer Science

AI Ingredients

Big Data Availability

350M images uploaded per day

300 hours of video uploaded every minute

2.5 Petabytes of customer data hourly

GPU Acceleration

New ML Techniques

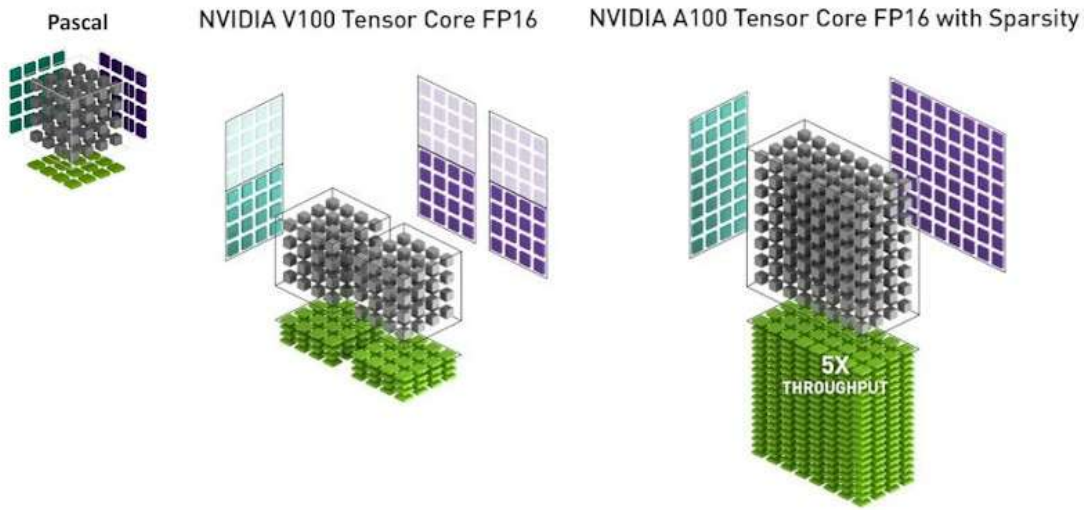
ACM's Celebration of 50 Years of the ACM Turing Award (June 2017)

"Compute has been the oxygen of deep learning"

– Ilya Sutskever, Research Director of Open AI

GPUs Targeting Deep Neural Networks

Add specialized hardware to support matrix multiplication, add support for reduced precision formats and exploit sparsity

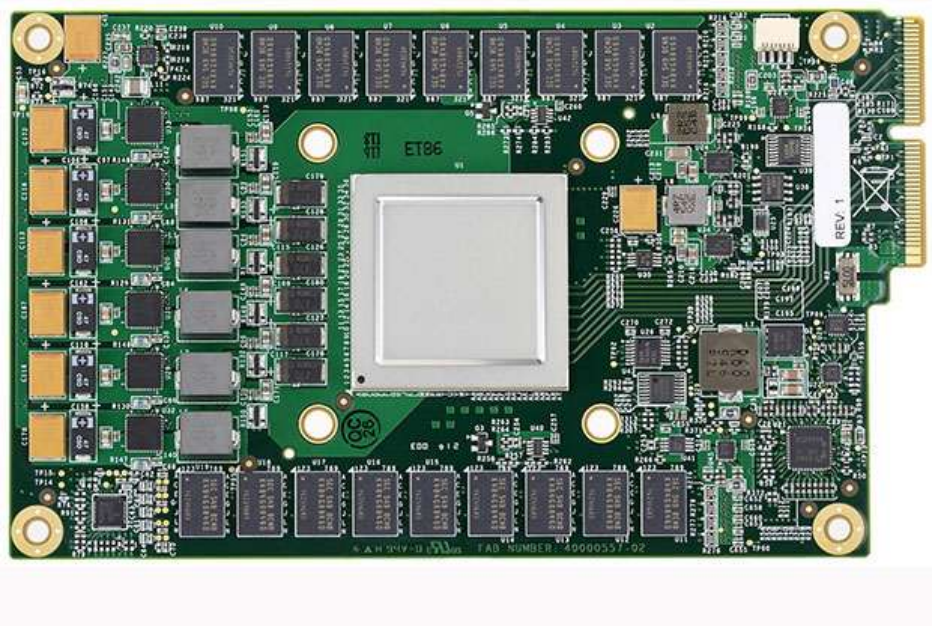


Introduced **Tensor Core in 2017**

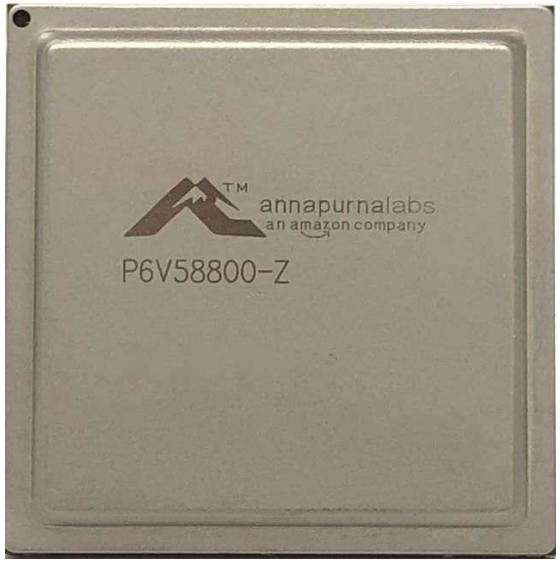
Source: Nvidia

Software Companies are Building HW

Data center operators build their own specialized hardware



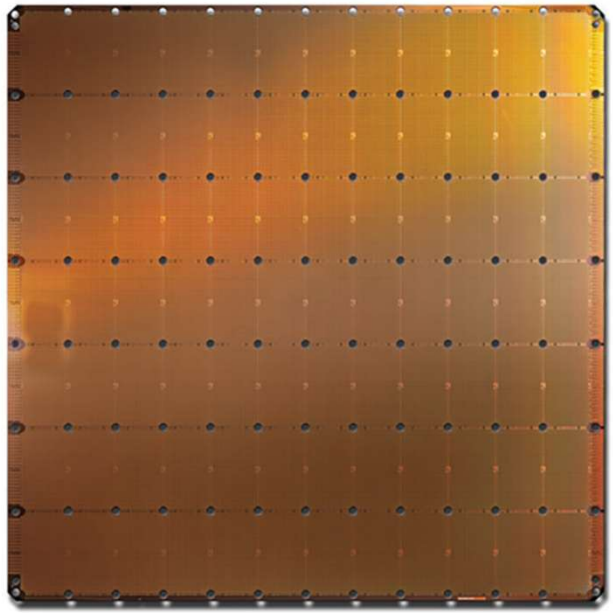
Google
TPUv1 in 2016 for inference
TPUv4 in 2021 for training and inference



Amazon
Inferentia in 2019 for inference
Trainium in 2020 for training

Specialized Hardware for Deep Neural Networks

Cerebras (WSE– Aug 2018)
400,000 cores, 18 GB SRAM, 9.6 PB/s
https://www.cerebras.net/wp-content/uploads/2019/08/Cerebras-Wafer-Scale-Engine-An-Introduction.pdf



Cerebras WSE
1.2 Trillion transistors
46,225 mm² silicon

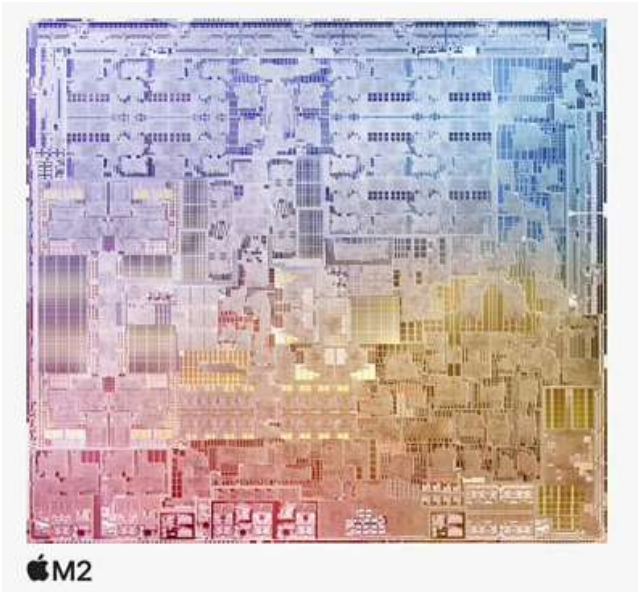
Largest GPU
21.1 Billion transistors
815 mm² silicon

Mobile SOCs for Deep Neural Networks

Phone and laptop chips have specialized hardware for DNNs in their System on Chip (SoC)

Apple A11 (Sept 2017)

Apple Neural Engine (ANE) introduced for FaceID and animated emojis (Animoji)



Apple M2 (June 2022)

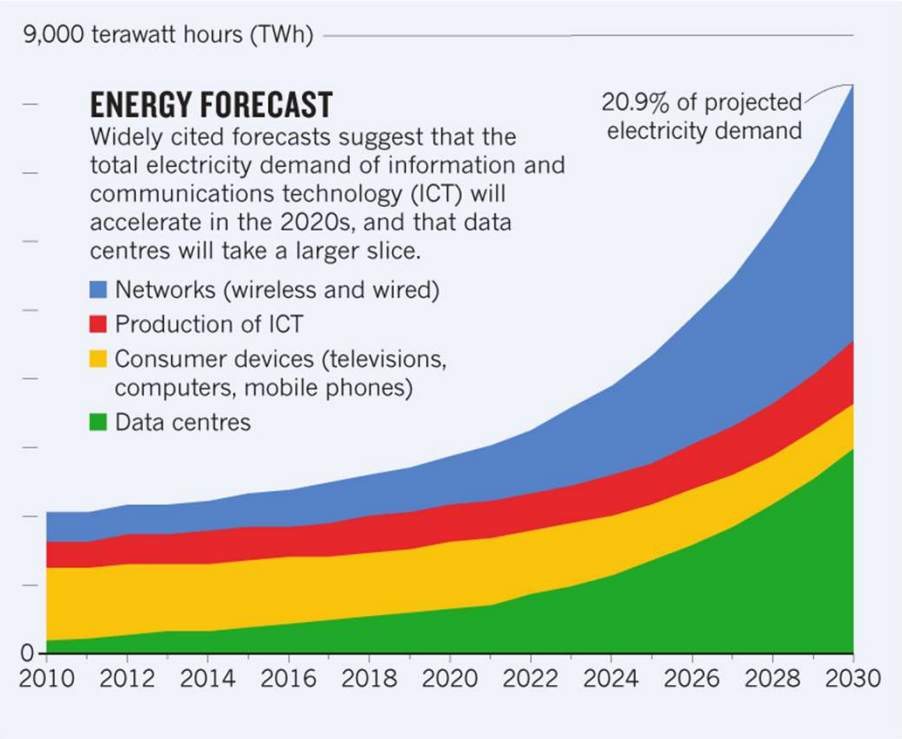
16 Neural Engine cores

A15 (Sept 2021) also has 16 neural Engine cores with a 26x speed up over A11

https://machinelearning.apple.com/research/neural-engine-transformers

Can run small, unsigned integer, pruned DNNs in 3 to 15 msec
https://machinelearning.apple.com/research/on-device-scene-analysis

Rapid Growth of Energy Consumption for Computing

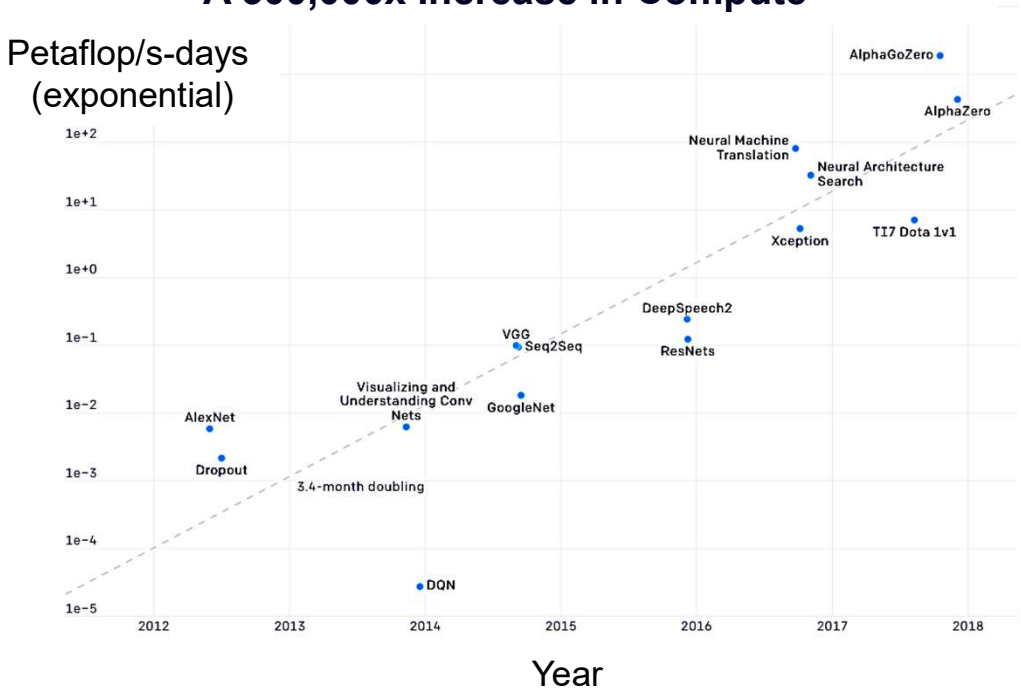


Data centers accounted for 3% of US global electricity demand in 2022
and is expected to grow to 8% by 2030
[Goldman Sachs, April 2024]

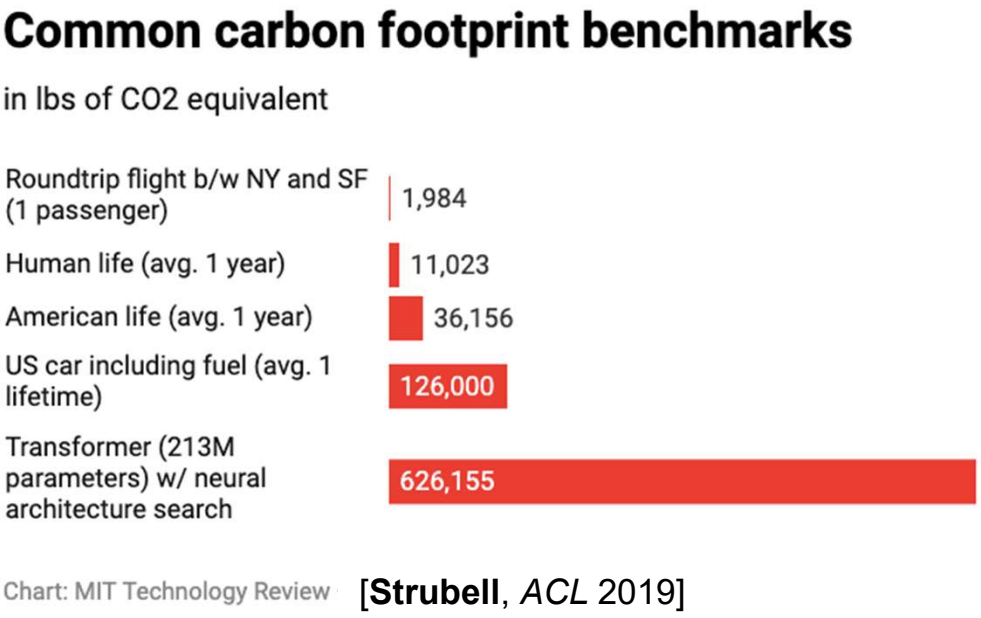
Source: Nature (https://www.nature.com/articles/d41586-018-06610-y)

Compute Demands for Deep Neural Networks

AlexNet to AlphaGo Zero:
A 300,000x Increase in Compute



Source: Open AI (https://openai.com/blog/ai-and-compute/)



[**Strubell**, *ACL* 2019]

Cloud Service Providers Investing in Power Plants

npr

NEWSLETTERS SIGN IN NPR SHOP

NEWS CULTURE MUSIC PODCASTS & SHOWS SEARCH

NATIONAL

Three Mile Island nuclear plant will reopen to power Microsoft data centers

SEPTEMBER 20, 2024 · 1:40 PM ET

By C Mandler

NuclearNewswire

Search the Nuclear Newswire

TOPICS | SOURCES | SIGN UP | BUYERS GUIDE | ADVERTISE | American Nuclear Society

Headlines For You

Supreme Court urged to uphold ruling against Texas SNF storage site
32m ago

NRC board to hear petitions on Palisades restart
16h ago

Ann Stouffer Bisconti—ANS member since 1990
19h ago

A message from PYRAGON and SOR Controls Group

The Advantage of Upgrading Power Supply Infrastructure in Nuclear Power Plants

CRANE program offers teachings on computational methods in nuclear fusion
22h ago

Studsvik

A message from Studsvik Scandpower

About Studsvik Scandpower

Learn More

INDUSTRY

Amazon buys nuclear-powered data center from Talen

Thu, Mar 7, 2024, 8:01AM | Nuclear News

Large Language Models: ChatGPT

Slide credit: Liane Bernstein

Computing Cost of ChatGPT (Training)

- ChatGPT is based on a variant of GPT-3 [**Brown,** *NeurIPS* 2020]
- GPT-3 has 96-layers, 175 billion parameters and requires $3.14x10^{23}$ FLOPS of computing for training
- It would take **355 years** to train GPT-3 on a Tesla V100 GPU
- It would cost **~$4.6 million** to train GPT-3 on using the lowest cost GPU cloud provider
- Training costs continue to rise
  - e.g., GPT-4 estimated **over $100 million**

Source: https://lambdalabs.com/blog/demystifying-gpt-3

Changing Trends?

How Chinese A.I. Start-Up DeepSeek Is Competing With Silicon Valley Giants

The company built a cheaper, competitive chatbot with fewer high-end computer chips than U.S. behemoths like Google and OpenAI, showing the limits of chip export control.

Training Costs	Pre-Training	Context Extension	Post-Training	Total
in H800 GPU Hours	2664K	119K	5K	2788K
in USD	$5.328M	$0.238M	$0.01M	$5.576M

Table 1 | Training costs of DeepSeek-V3, assuming the rental price of H800 is $2 per GPU hour.

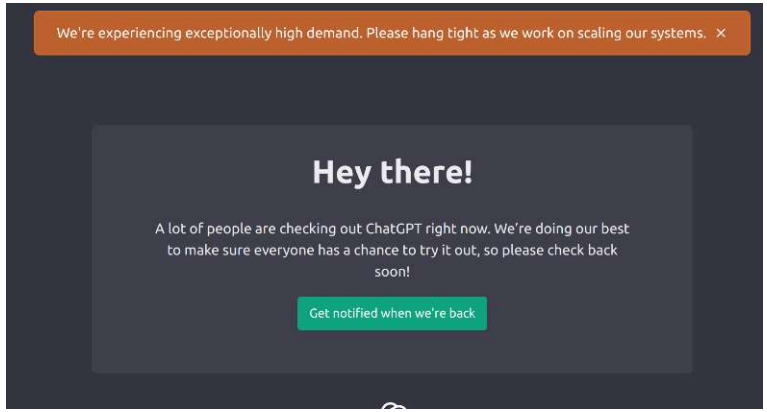
Feature/Model	DeepSeek V3	DeepSeek Coder	DeepSeek R1
Primary Purpose	General-purpose multitasking	Coding and programming-specific tasks	Logical reasoning and problem-solving
Training Focus	Coding, mathematics, multilingualism	Code datasets (87% code, 13% natural language)	Reinforcement learning for reasoning
Architecture	Mixture-of-Experts (MoE)	Traditional Transformer architecture	Reinforcement Learning (RL) optimized
Use Cases	Multilingual tools, research, AI apps	IDE integration, coding platforms	Educational platforms, research tools
Open Source	Yes	Yes	Yes
Parameter Range	671B (37B activated per token)	1.3B to 33B	1.5B to 70B

https://play.ht/blog/deepseek-v3-vs-r1-vs-coder/

Computing Cost of ChatGPT (Inference)

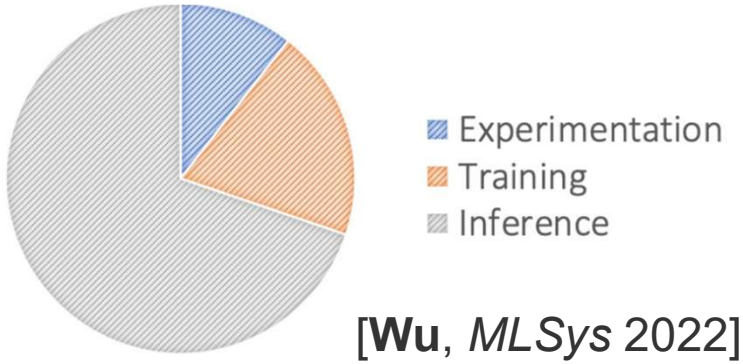
Estimated monthly cost of
$1.5 to $8 million!

Source: https://medium.com/swlh/3-questions-puzzled-me-about-openais-chatgpt-and-here-is-what-i-learned-1dda74b5f6db



Compute for Training versus Inference

- Training **high cost per iteration**, but **low frequency**
- Inference **low cost per iteration**, but **high frequency**
- Regarding computing at Google:
  - "Across all three years [2019-2021], about three-fifths of the ML energy use was for inference, and two-fifths were for training" [**Patterson**, *Computer* 2022]
- Regarding computing at Meta:
  - "…trillions of inference per day across Meta's data centers. … we observe a rough power capacity breakdown of 10:20:70 for AI infrastructures devoted to the three key phases — Experimentation, Training, and Inference"



[**Wu**, *MLSys* 2022]

Huge Financial Investment in GPUs

Home > News > Components > Graphics Cards

Zuckerberg's Meta Is Spending Billions to Buy 350,000 Nvidia H100 GPUs

In total, Meta will have the compute power equivalent to 600,000 Nvidia H100 GPUs to help it develop next-generation AI, says CEO Mark Zuckerberg.

By Michael Kan January 18, 2024

Source: https://www.pcmag.com/news/zuckerbergs-meta-is-spending-billions-to-buy-350000-nvidia-h100-gpus

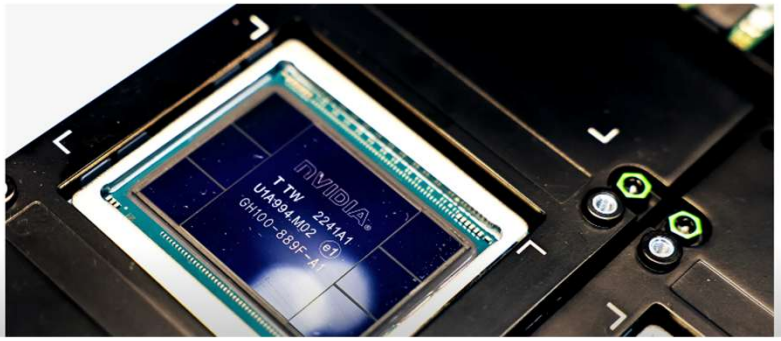
GPU Shortage

WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS POLITICS SCIENCE SECURITY MERCH SIGN IN SUBSCRI

PARESH DAVE BUSINESS AUG 24, 2023 6:00 AM

Nvidia Chip Shortages Leave AI Startups Scrambling for Computing Power

Trimming profits, delaying launches, begging friends. Companies are going to extreme lengths to make do with shortages of GPUs, the chips the heart of generative AI programs.



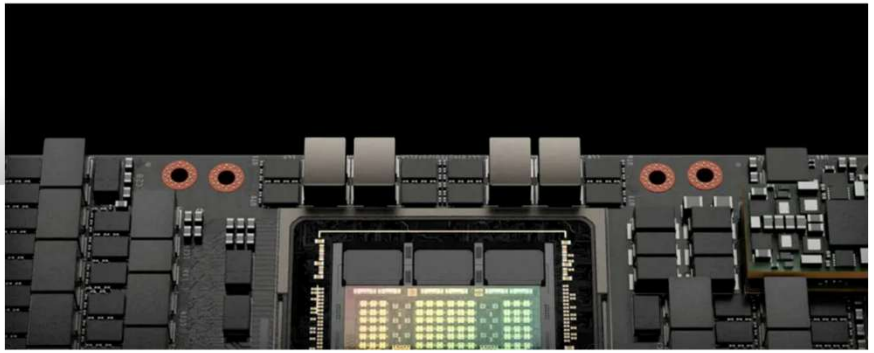
Source: https://www.wired.com/story/nvidia-chip-shortages-leave-ai-startups-scrambling-for-computing-power/

The New York Times

The Desperate Hunt for the A.I. Boom's Most Indispensable Prize

To power artificial-intelligence products, start-ups and investors are taking extraordinary measures to obtain critical chips known as graphics processing units, or GPUs.

Share full article 42



Source: https://www.nytimes.com/2023/08/16/technology/ai-gpu-chips-shortage.html

Processing “On-Device” instead of the “Cloud”

Communication

Privacy

Latency

Computing Challenge for Self-Driving Cars

JACK STEWART TRANSPORTATION 02.06.18 08:00 AM

SELF-DRIVING CARS USE CRAZY AMOUNTS OF POWER, AND IT'S BECOMING A PROBLEM

Shelley, a self-driving Audi TT developed by Stanford University, uses the brains in the trunk to speed around a racetrack autonomously.

NIKKI KAHN/THE WASHINGTON POST/GETTY IMAGES

WIRED

(Feb 2018)

Cameras and radar generate ~6 gigabytes of data every 30 seconds.

Self-driving car prototypes use approximately 2,500 Watts of computing power.

Generates wasted heat and some prototypes need water-cooling!

Self-Driving Cars: Data Center On Wheels

Data center

"[T]rillions of inference per day across Facebook's data centers"

[**Wu**, *MLSys* 2021]

Autonomous vehicles (AVs) w/ 10 deep neural network (DNN) inferences at 60 Hz on 10 cameras:

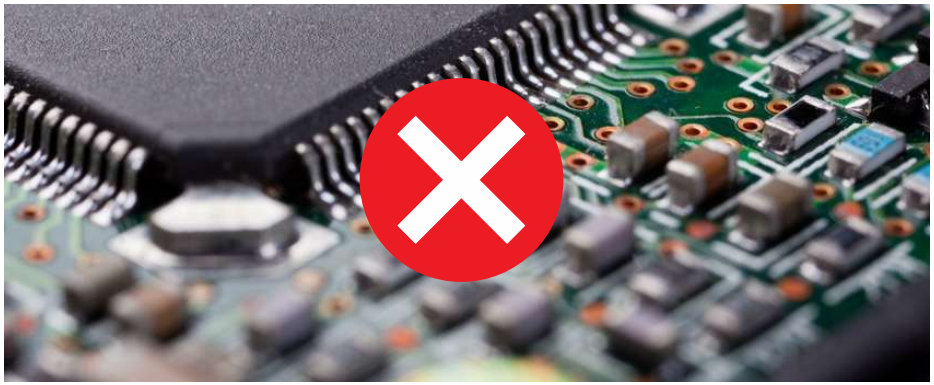
One AV: 21.6 million inferences per hour driven

One million AVs (< 0.1% of vehicles worldwide): 21.6 trillion inferences per hour driven!

[**Sudhakar**, IEEE *Micro* 2023]

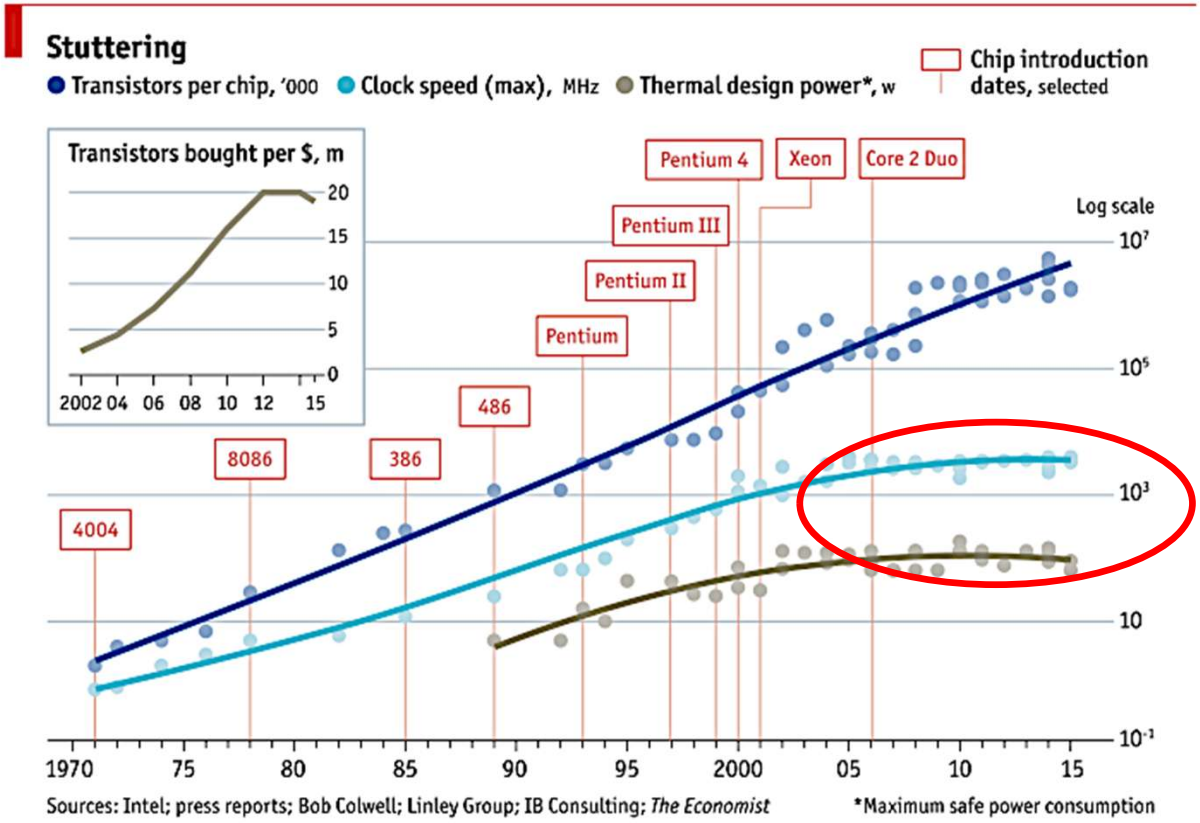
Existing Processors Consume Too Much Power

< 1 Watt



> 10 Watts

Transistors Are Not Getting More Efficient



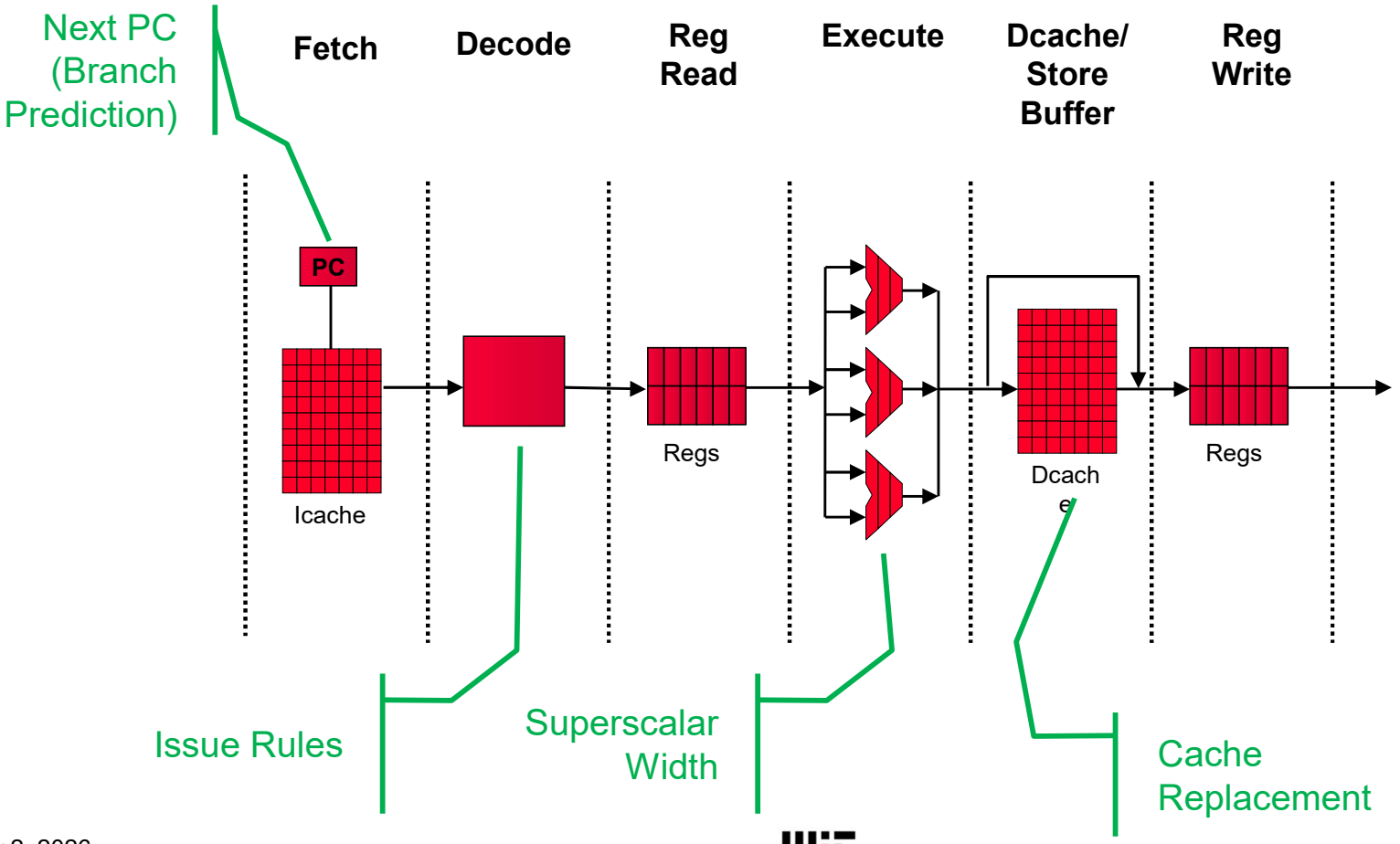
Slowdown of Moore's Law and Dennard Scaling

General purpose microprocessors (CPUs) are not getting faster or more efficient

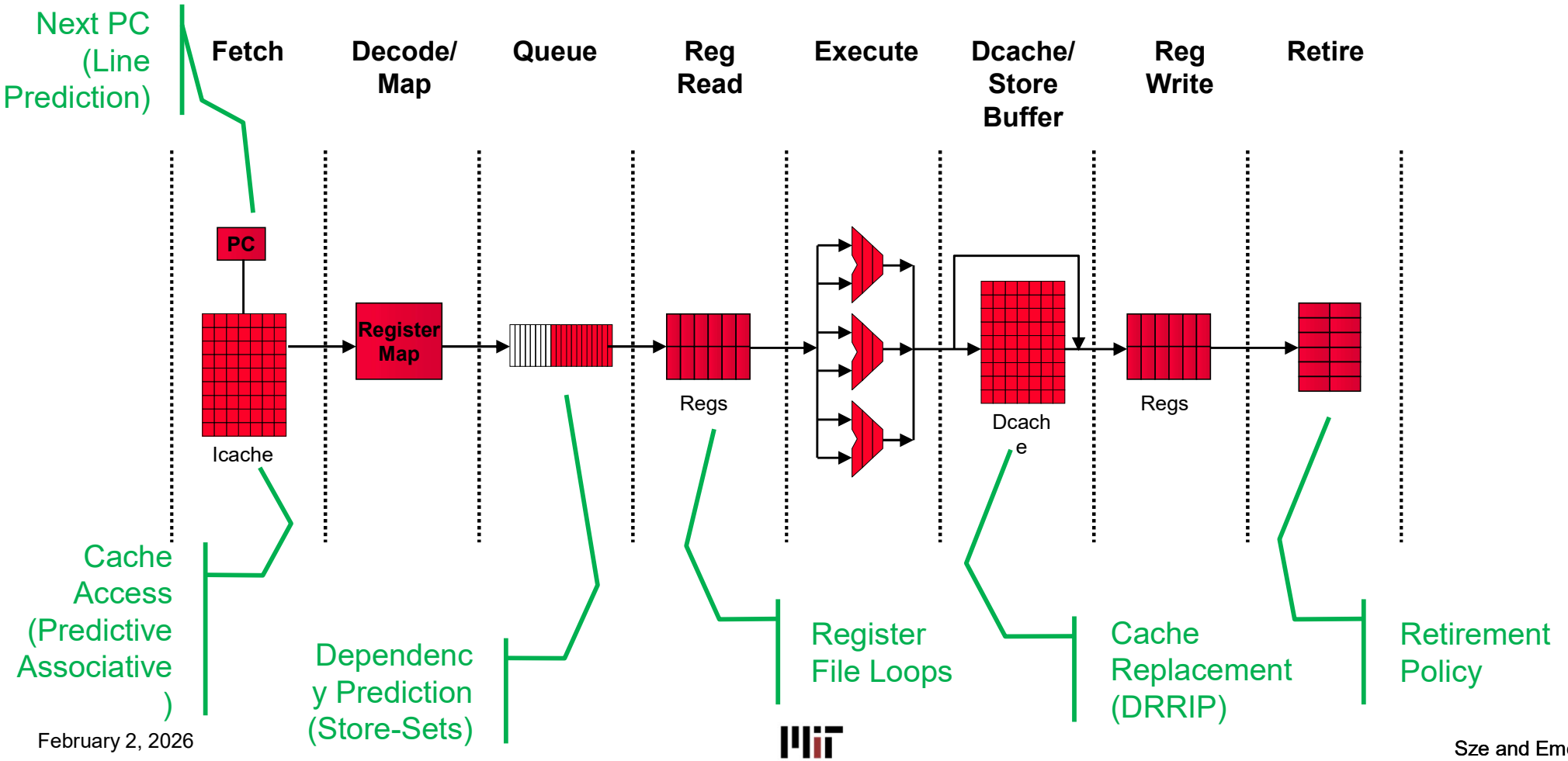
Slowdown

Need **specialized / domain-specific hardware** for significant improvements in speed and energy efficiency

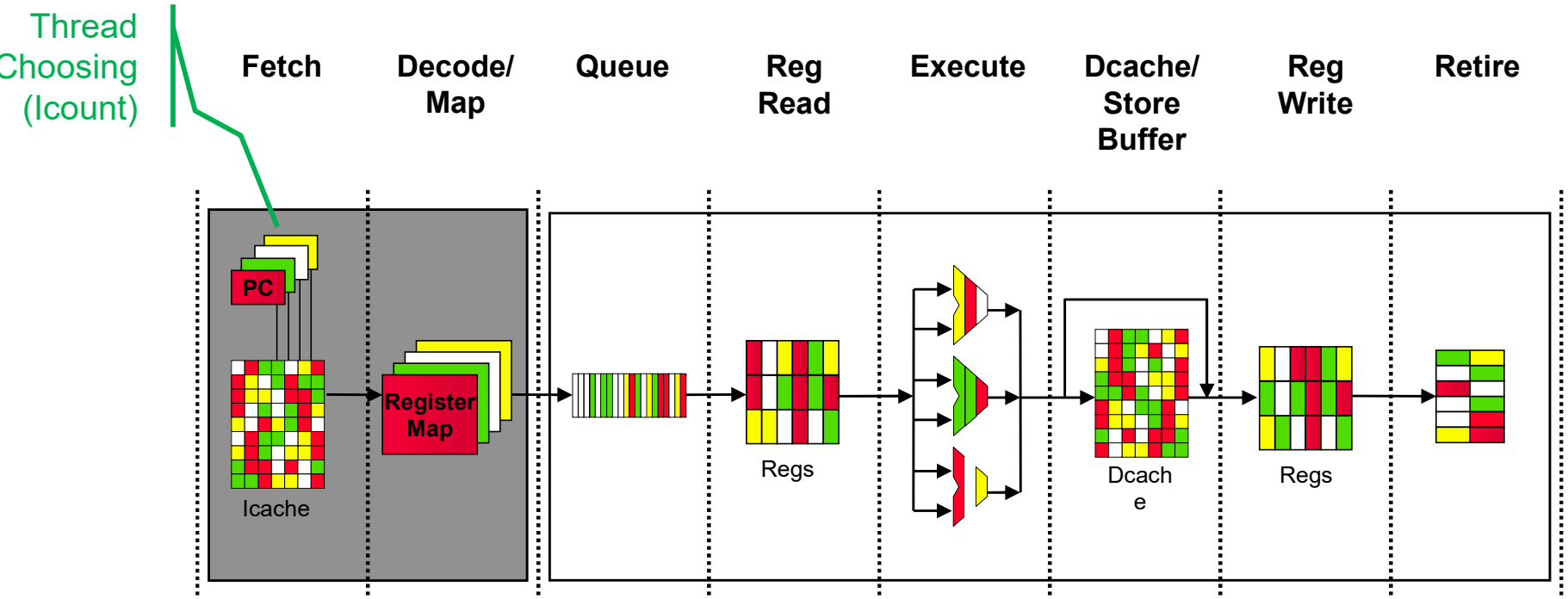
Simple In-Order Pipeline



Basic Out-of-order Pipeline

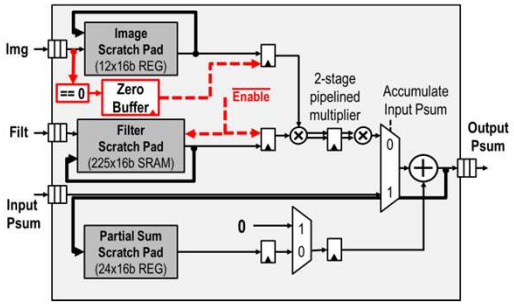


Out-of-Order SMT Pipeline

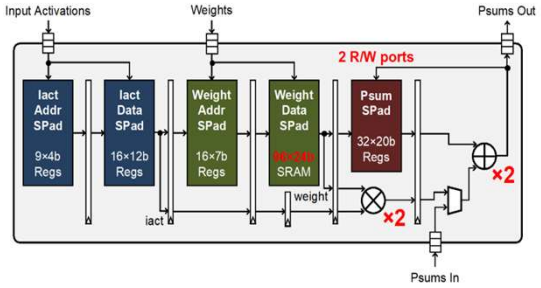


Every Accelerator is Unique

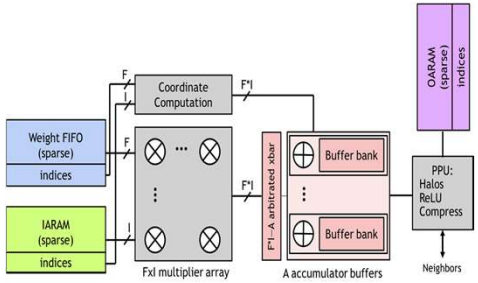
Selected tensor accelerator designs - 2017-2021



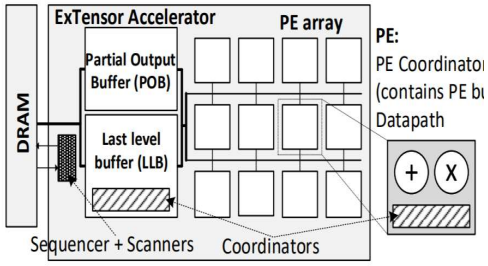
Eyeriss [JSSC2017]



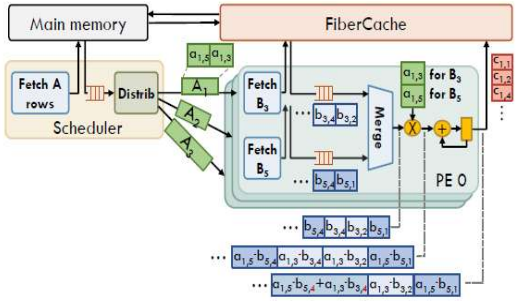
Eyeriss V2 [JETCAS2019]



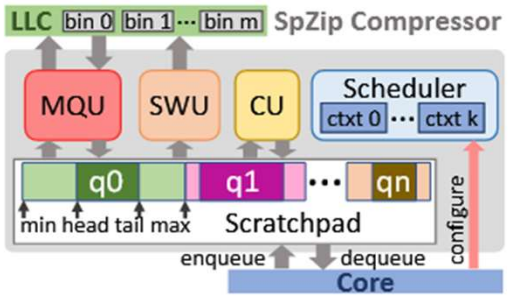
SCNN [ISCA2017]



ExTensor [MICRO2019]



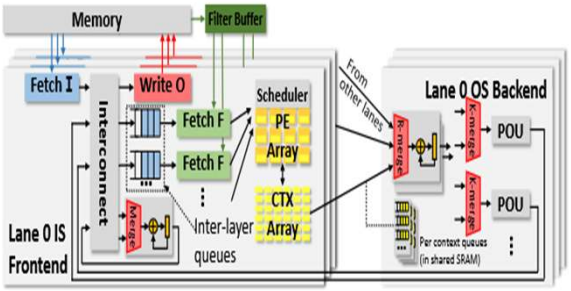
Gamma [ASPLOS2021]



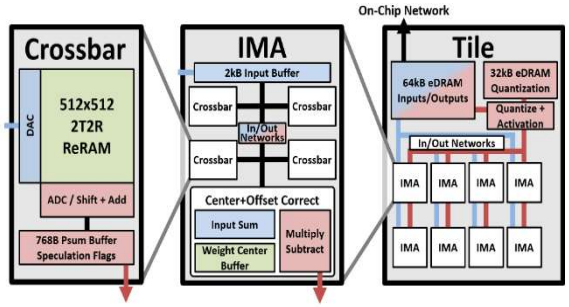
spZip [ISCA2021]

Every Accelerator is Unique

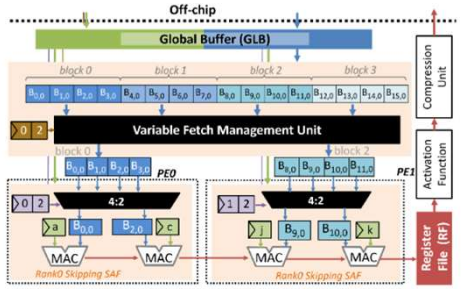
Selected tensor accelerator designs - 2023 -



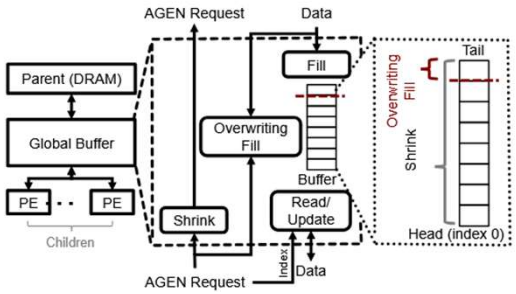
ISOSceles [HPCA2023]



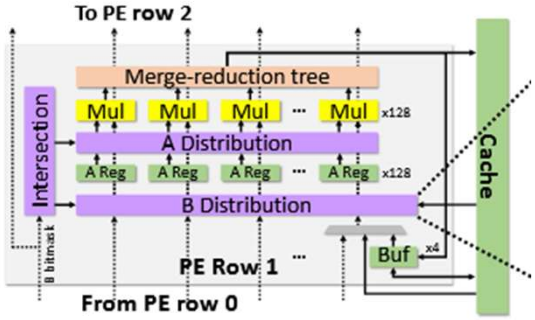
RAELLA [ISCA2023]



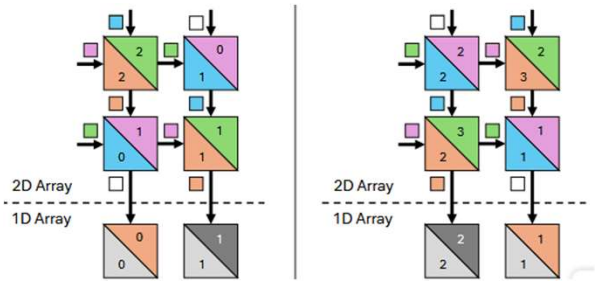
Highlight [MICRO2023]



Overbooking [MICRO2023]

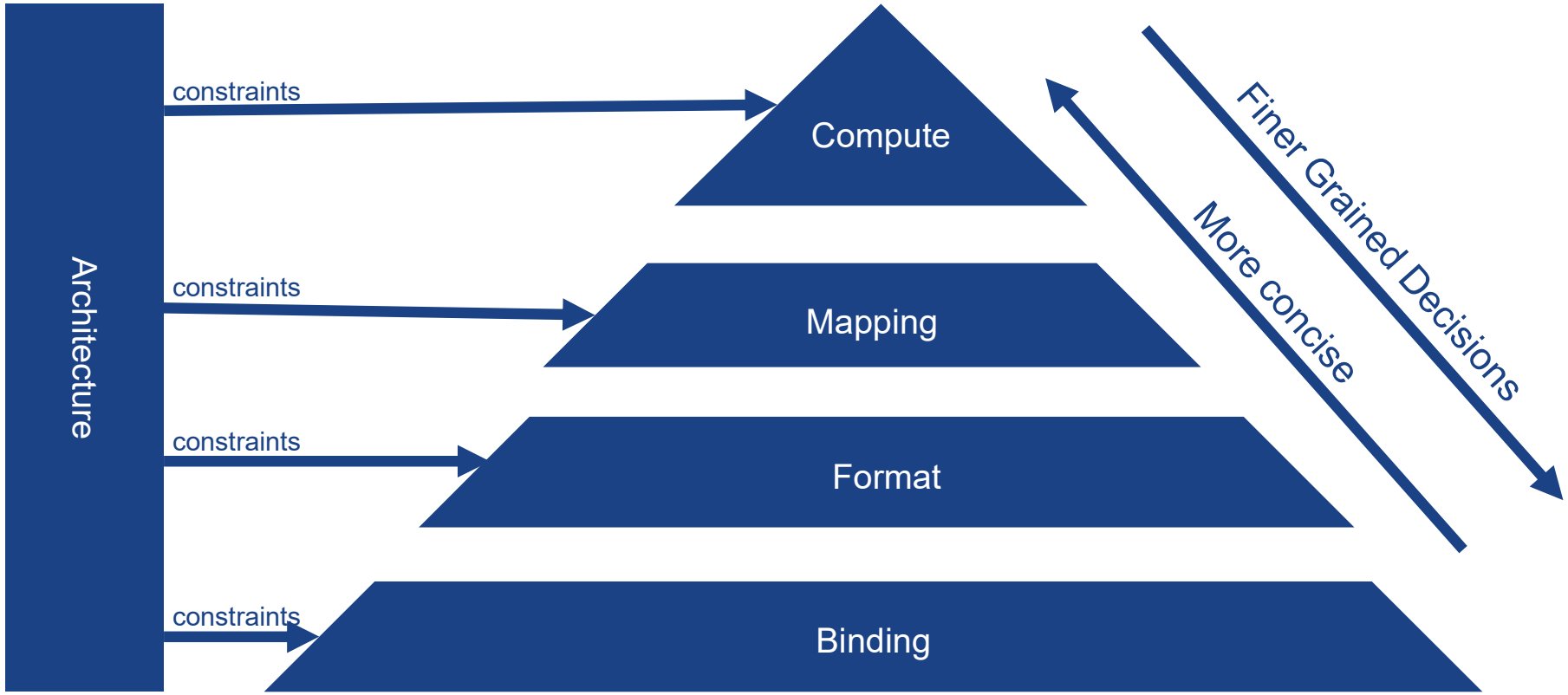


Trapezoid [ISCA 2024]



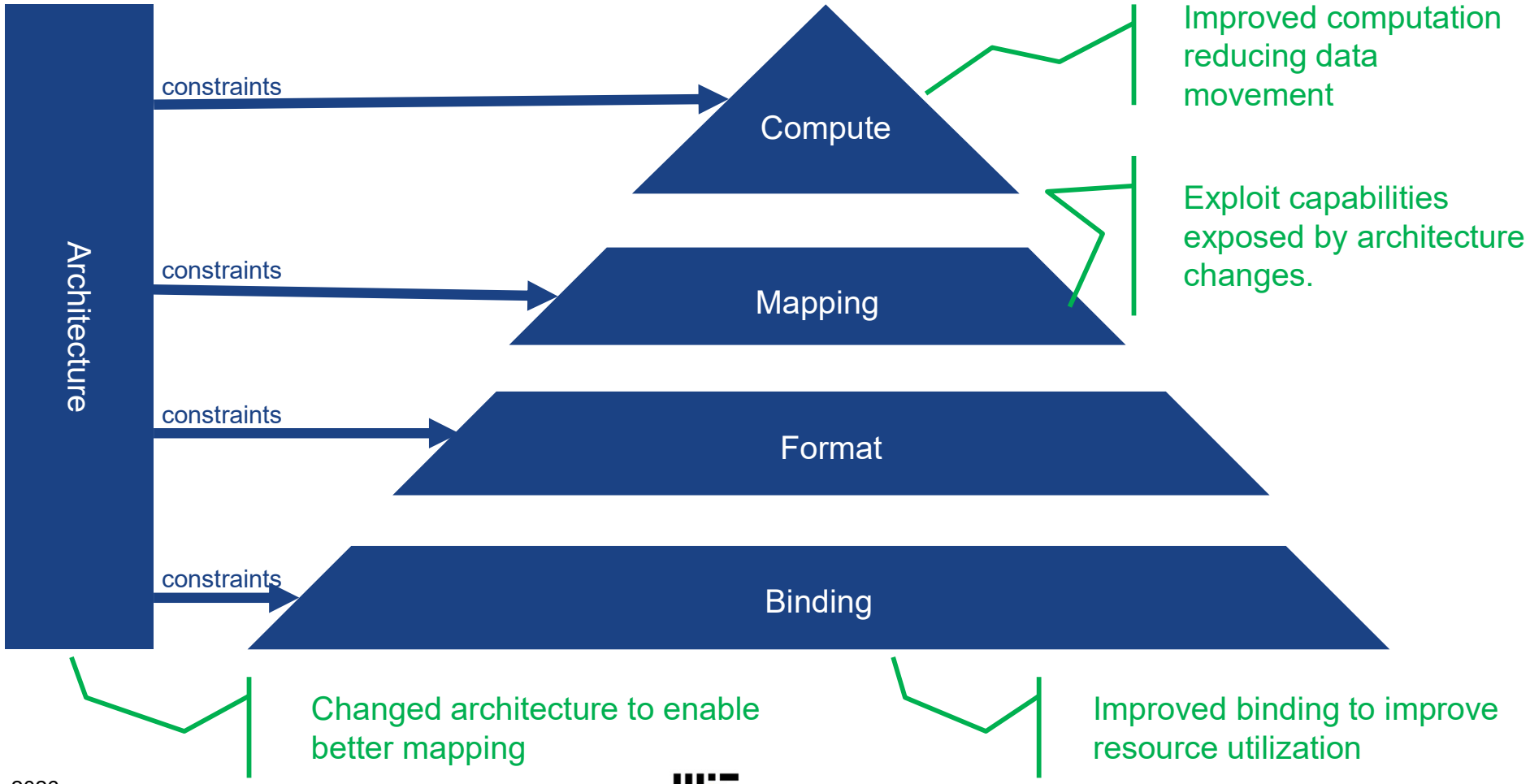
FuseMax [MICRO2024]

TeAAL Pyramid of Concerns



[TeAAL, Nayak et.a. MICRO 2023]

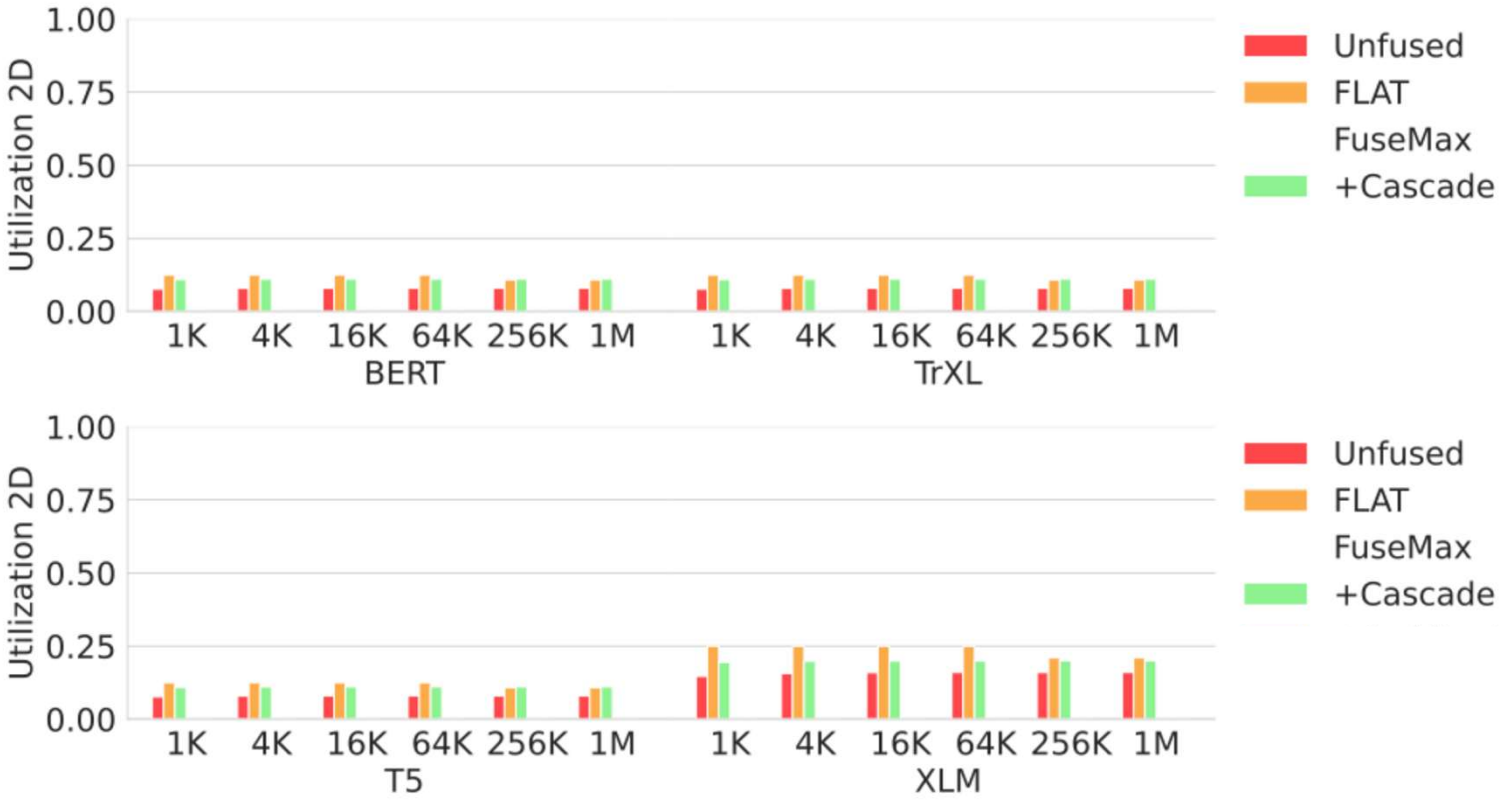
FuseMax - Enhancements



PE Utilization - Baseline

[FuseMax, Nayak et.a. MICRO 2024]

PE Utilization – Enhanced Computation

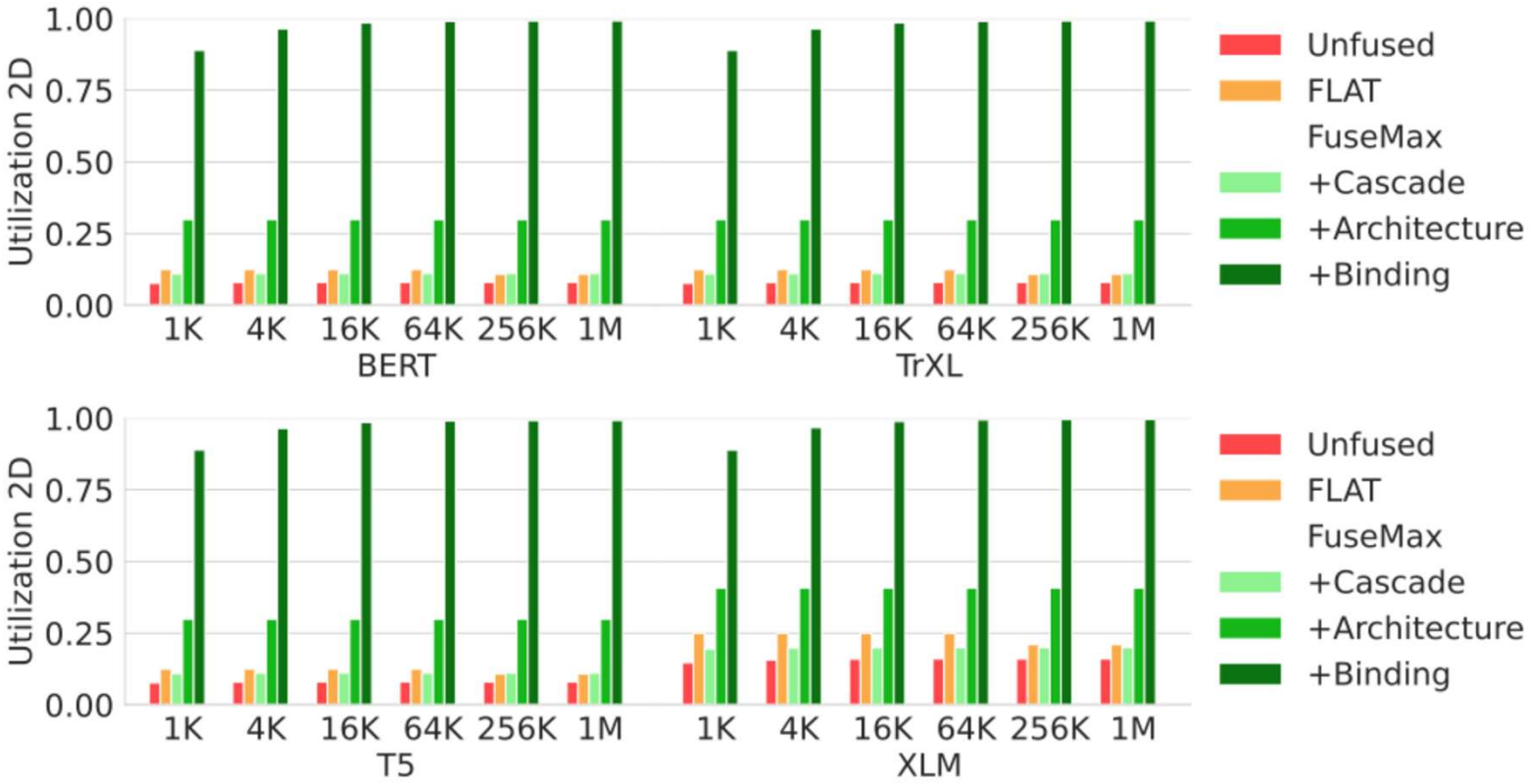


[FuseMax, Nayak et.a. MICRO 2024]

PE Utilization – Improve Architecture/Mapping

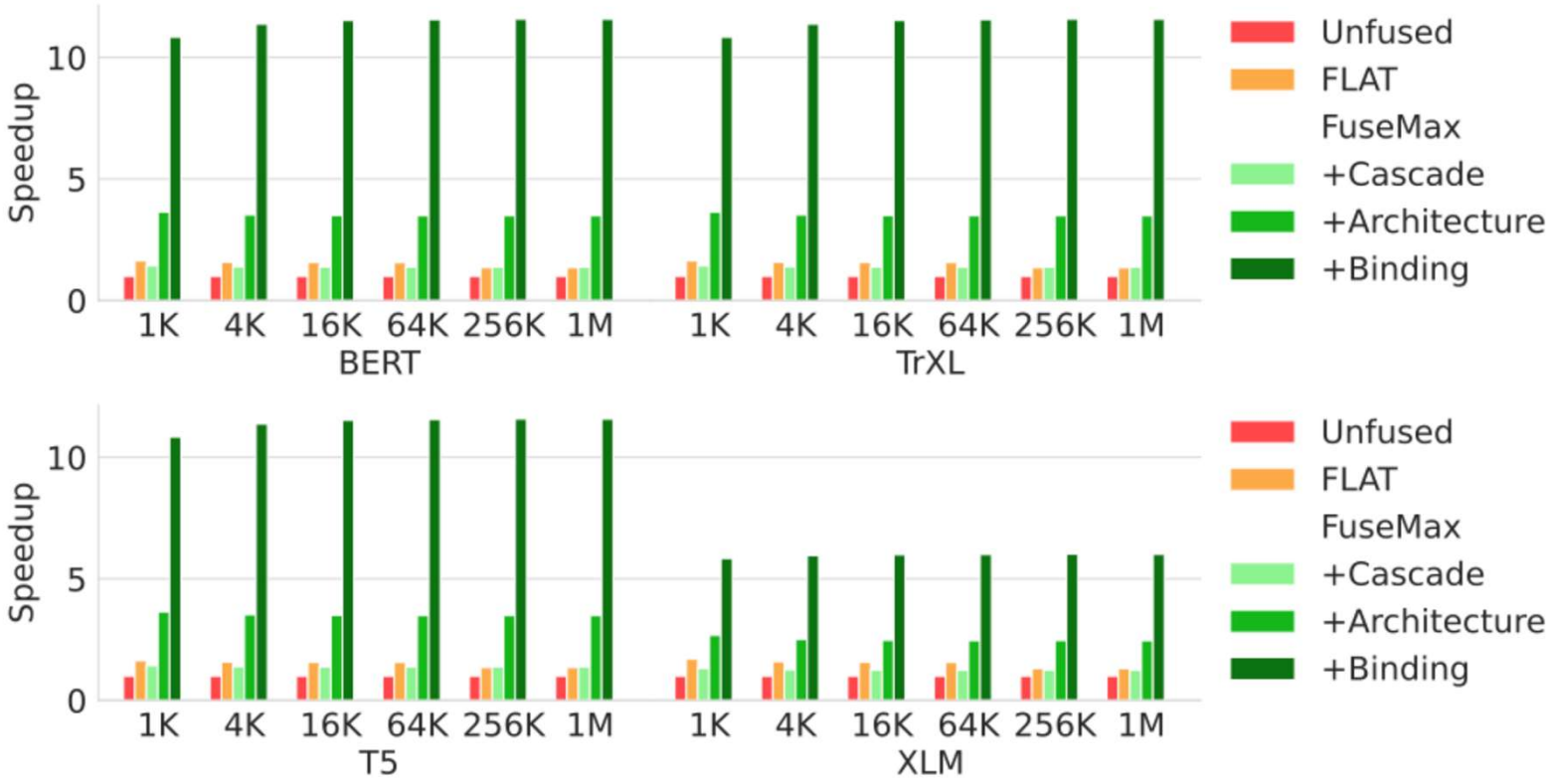
[FuseMax, Nayak et.a. MICRO 2024]

PE Utilization – Improved binding



[FuseMax, Nayak et.a. MICRO 2024]

FuseMax - Speedup on Attention



[FuseMax, Nayak et.a. MICRO 2024]

Challenges and Opportunities

- Define the domain and degree of flexibility required
- Requires understanding of domain, variants of algorithms (identify which ones are important), and resulting workloads
- Handle heterogenous computing at the system level
- Handle heterogenous devices (emerging device technology)
- May have tighter resource constraints since hardware cannot be used for other applications
- Co-design across algorithms and hardware
- Domain specific languages to program specialized hardware
- Tools for rapid evaluation and prototyping

Class Overview

Course Outline

- Overview of Deep Neural Networks (DNNs)
- DNN Development Resources
- DNNs on Programmable Hardware
- DNN Accelerator Architecture
- DNN Model and Hardware Co-Design
- Advanced Technologies for DNN

Takeaways

- Know the key computations used by DNNs
- Be familiar with how DNN computations are mapped to various hardware platforms
- Understand the tradeoffs between various architectures and platforms
- Be able to evaluate different DNN accelerator implementations with benchmarks and comparison metrics
- Have an appreciation for the utility of various optimization and approximation approaches
- Be able to distill the key attributes of recent implementation trends and opportunities

Course Objective

By the end of this course, we want you to be able understand a new design in terms of attributes like:

- Order of computation
- Partitioning of computation
- Flow of data for computation
- Data movement in the storage hierarchy
- Data attribute specific optimizations
- Exploiting algorithm/hardware co-design
- Degree of flexibility

Course Staff and Contact Info

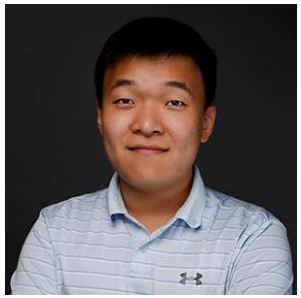
- Instructors (Office hours by request)
  - Joel Emer (jsemer@mit.edu)
  - Vivienne Sze (sze@mit.edu)
- TAs (Office hours W 5-6PM @ Location **TBD**)
  - Tanner Andrulis (andrulis@mit.edu)
  - Michael Gilbert (gilbertm@mit.edu)
  - Fisher Xue (fzyxue@mit.edu)
  - Reng Zheng (rengz@mit.edu)


Joel Emer


Vivienne Sze


Tanner Andrulis


Michael Gilbert


Fisher Xue


Reng Zheng

- Schedule
  - Lectures: MW 1PM-2:30PM – 54-100
  - Recitations/Office Hours: F 11AM-12PM – 32-155
- Course Website: http://csg.csail.mit.edu/6.5930/

Slide Contributors: ***Joel Emer****, Yu-Hsin Chen and Tien-Ju Yang (http://eyeriss.mit.edu/tutorial.html)*

Course Requirements and Materials

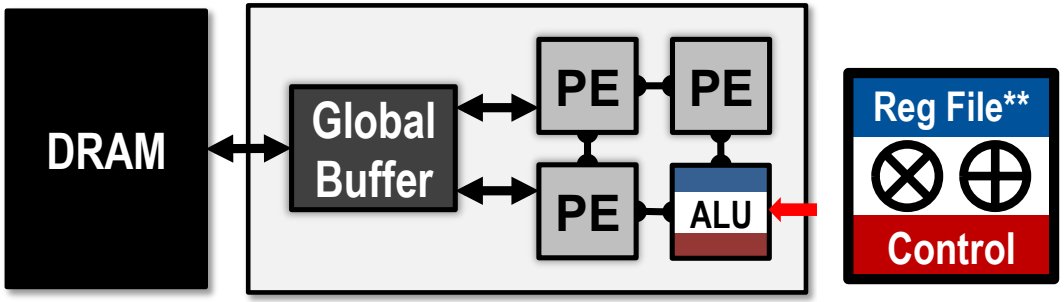
- Pre-requisites
  - $6.1910_{[6.004]}$ (Computation Structures)
  - $6.3000_{[6.003]}$ (Signal Processing) or $6.3900_{[6.036]}$ (Intro to Machine Learning)
- We will use Python and PyTorch
  - PyTorch website: https://pytorch.org/
  - Introduction to PyTorch Code Examples: https://cs230.stanford.edu/blog/pytorch/
- Course Textbook/Readings
  - Book "Efficient Processing of Deep Neural Networks"
    - https://doi.org/10.1007/978-3-031-01766-7 (download free on MIT network)
    - *We welcome feedback (including errata) on Piazza thread*
  - Selected papers published in past few years.
- Course Handouts (uploaded on http://csg.csail.mit.edu/6.5930/)

Labs

- Lab 0: Infrastructure setup (Released Today on Piazza)
  - Release: Feb 2 – Due: Feb 6
- Lab 1: Analyze and Evaluate DNN workloads (Einsums)
  - Release: Feb 4 – Due: Feb 13 (1.5 weeks)
- Lab 2: Hardware Design & Mapping
  - Release: Feb 13 – Due: Feb 23 (1.5 weeks)
- Lab 3: Advanced Mapping: Parallel Processing and Fusion
  - Release: Feb 23 – Due: March 4 (1.5 weeks)
- Lab 4: Sparsity
  - Release: March 4 – Due: March 13 (1.5 weeks)
- Lab 5: Compute In Memory (CiM)
  - Release: March 13 – Due: March 20 (1 week)

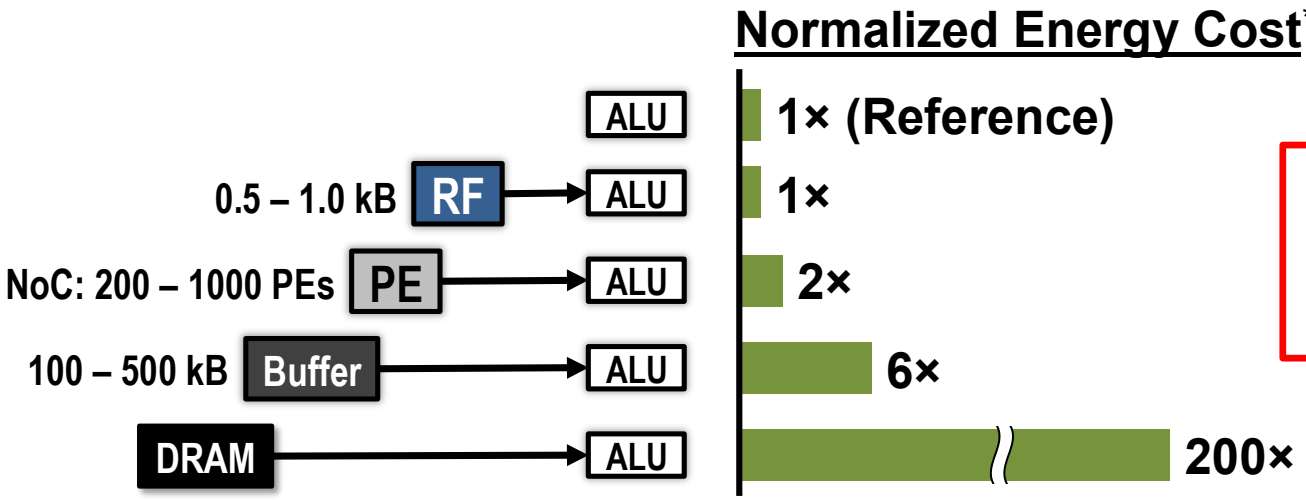
All labs due before spring break

Typical Architecture for DNN Accelerator



Spatial architecture with small (< 1kB) low-cost memory near compute

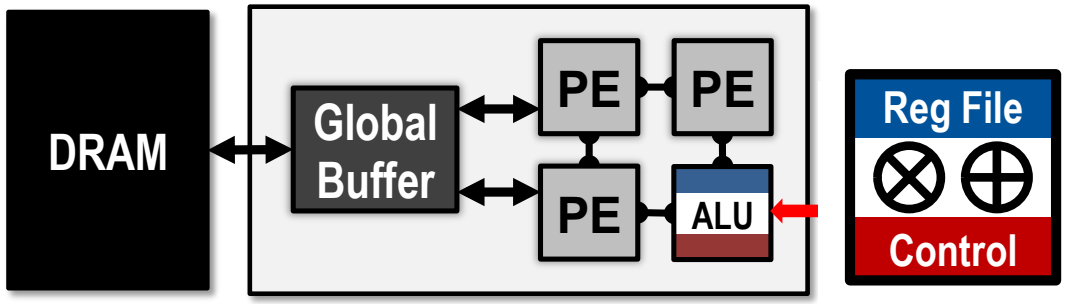
**Register File is also referred to as local buffer



Farther and **larger** memories consume more power

* measured from a commercial 65nm process

Example Design Choices for DNN Accelerator

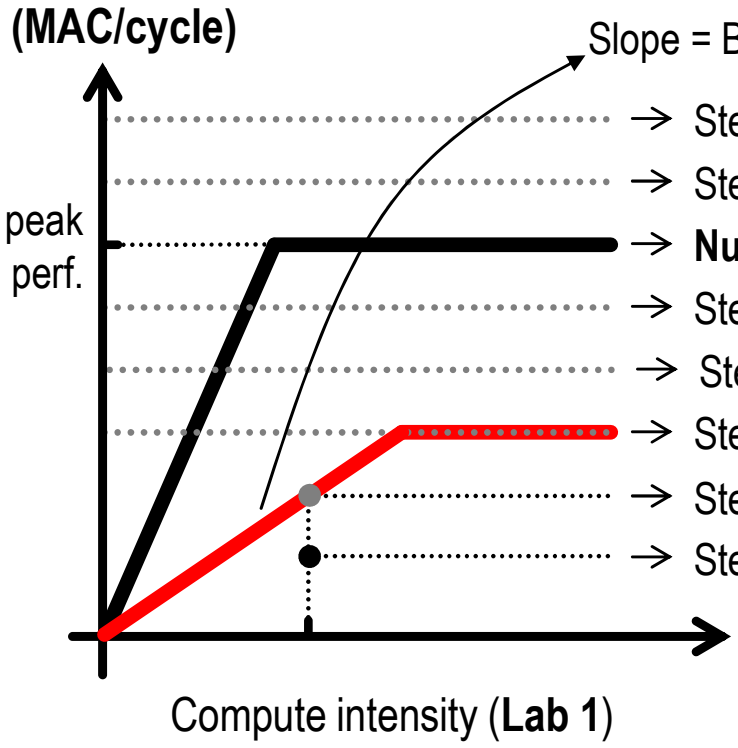


- Processing Element (PE) array (**Lab 2 & 3**)
  - number of PEs, connection between PEs (Network on chip [NoC])
- Memory hierarchy (**Lab 2 & 3**)
  - number of levels, capacity per level, data layout in memory
- Scheduling of operations to reduce data movement and increase PE utilization (**Lab 2 & 3**)
  - mapping (dataflow, tiling), parallelism, fusion
- Handling sparsity (**Lab 4**)
  - gating, skipping, representation format (compression)
- Technology used to implement components such as PE, NoC and Memory (**Lab 5**)
  - e.g., RRAM, optical, superconductors

Evaluate Inefficiencies in DNN Accelerators

Use **roofline model** as a systematic way to evaluate how each architectural decision affects performance (throughput) for a given DNN workload

Tightens the roofline model



→ Step 1: maximum workload parallelism (total iteration space of Einsum (**Lab 1**))

→ Step 2: maximum dataflow parallelism (product of loop limits of parallel_for)

→ **Number of PEs** (Theoretical Peak Performance)

→ Step 3: # of active PEs under a finite PE array size (<= min (# of PEs, Step 2))

→ Step 4: # of active PEs under fixed PE array dimension (constrained on both dimens.)

→ Step 5: # of active PEs under fixed storage capacity (space for stationary data)

→ Step 6: lower active PE util. due to insufficient average BW (tools cover up to this)

→ Step 7: lower active PE util. due to insufficient instantaneous BW (ramp up/down)

See Chapter 6 in Book

Why Architectural Modeling?

- In this class, primarily dealing with architectural design decisions
- Want to rapidly explore design space and evaluate design decisions
  - Each design would take a long time to implement and evaluate using RTL
- Perform evaluation of architectural design decisions using modeling tools (AccelForge and Accelergy)
  - Once identify desirable architecture can implement portions in RTL to increase accuracy of evaluation (not part of labs, but can be part of design project)

- Note: No GPUs programming (check out 6.S894) or FPGA implementation (check out 6.2050 [6.111])

Design Project

- Project
  - Choose from a list of suggested projects
- Use tools from labs
  - PyTorch, Accelergy, AccelForge
- Teams of 3 (may change due to course enrollment)
- Schedule
  - March 9 – List of projects released
  - March 18 – Submit project selection
  - **March 30 – April 29** – Weekly check ins + milestone report outs with mentor during lecture (attendance at weekly check ins is mandatory)
    - **April 17 - Milestone 3 proposal due**
  - May 1 – Project Report Due
    - Graduate groups (w/ at least one member enrolled in grad version) include an overview on related work
    - Poster sessions May 4, 6, and 11

Lecture time dedicated to project after spring break

Design Project

- Design project allows for deeper understanding and application of concepts covered in course
  - Explore more advanced functionality of modeling tools used in the labs
- Projects will have three key milestones
  - Milestone 1 (April 6) [15 pt]: Read relevant paper and model prior work (baseline design)
  - Milestone 2 (April 13) [10 pt]: Perform various modifications to analyze impact on design metrics. Demonstrate understanding the various tradeoffs.
  - Milestone 3 (April 27) [5 pt]: Open-ended design space exploration (Proposal due April 17)
- Note:
  - We recommend that you "**complete"** each milestone before moving on to the next

Example Design Projects

- Hardware design study of well-known DNN accelerators (e.g., TPU, NVDLA) and analyze architectural tradeoffs though modeling
- Analyze co-design approaches to gain deeper understanding of impact on accuracy and efficiency
- Evaluate impact of new technologies (e.g., RRAM, Optical, superconductors)
- Extend tools for improved design exploration and analysis capabilities

Goal: Apply concepts and tools from class!

Published Design Projects!

Architecture-Level Modeling of Photonic Deep Neural Network Accelerators

Tanner Andrulis, *MIT*, Cambridge, USA, andrulis@mit.edu
Gohar Irfan Chaudhry, *MIT*, Cambridge, USA, girfan@mit.edu
Vinith M. Suriyakumar, *MIT*, Cambridge, USA, vinithms@mit.edu
Joel S. Emer, *MIT, Nvidia*, Cambridge, USA, jsemer@mit.edu
Vivienne Sze, *MIT*, Cambridge, USA, sze@mit.edu

***Abstract*—Photonics is a promising technology to accelerate Deep Neural Networks as it can use optical interconnects to reduce data movement energy and it enables low-energy, high-throughput optical-analog computations.**

To realize these benefits in a full system (accelerator + DRAM), designers must ensure that the benefits of using the electrical, optical, analog, and digital domains exceed the costs of converting data between domains. Designers must also consider system-level energy costs such as data fetch from DRAM. Converting data and accessing DRAM can consume significant energy, so to evaluate and explore the photonic system space, there is a need for a tool that can model these full-system considerations.

In this work, we show that similarities between Compute-in-Memory (CiM) and photonics let us use CiM system modeling tools to accurately model photonics systems. Bringing modeling tools to photonics enables evaluation of photonic research in a full-system context, rapid design space exploration, co-design, and comparison between systems.

Using our open-source model, we show that cross-domain conversion and DRAM can consume a significant portion of photonic system energy. We then demonstrate optimizations that reduce conversions and DRAM accesses to improve photonic system energy efficiency by up to 3×.

***Index Terms*—photonics, optical computing, photonic computing, compute-in-memory, modeling, accelerator**

I. Introduction

Deep Neural Networks (DNNs) can be energy-intensive to compute due to the movement of large tensors and the many multiply-accumulate (MAC) operations that they require. To address these challenges, photonic systems (accelerator + DRAM) leverage the digital-electrical (DE), analog-electrical (AE), digital-optical (DO), analog-optical (AO) domains. Specifically, optical (*i.e.*, DO and AO) interconnects can

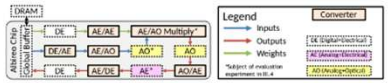


Fig. 1. Albireo architecture. As data traverse the DE, AO, and AE domains, they leverage different movement and reuse opportunities but pay energy for data converters, notated X/Y for conversion from domain X to domain Y.

optical resonators), architecture (*i.e.*, what components are used, how many components, how they connect), workload (*i.e.*, DNN layer types, tensor shapes/values), and mapping (*i.e.*, how the workload is scheduled onto the architecture).

Fortunately, these characteristics are not unique to photonics. Analog Compute-in-Memory (CiM) systems have a large full-system co-design space, leverage the advantages of multiple domains (AE and DE), and face the challenge of high cross-domain conversion energy.

In this work, we show that these similarities let us leverage the open-source CiMLoop [1]–[4] tool to accurately model photonic systems. Bringing this tool to photonics enables researchers to **(1)** accurately evaluate and compare research contributions in a full-system context (*e.g.*, see how a novel component affects a full system or compare two photonic systems across a range of DNN workloads) **(2)** perform fast design-space exploration over the large co-design space [1], and **(3)** share knowledge between the photonics and CiM research communities.

II. Photonics Modeling Tool

The tool takes as input specifications of a DNN workload, components, and architecture as defined in Section I. The tool

ISPASS 2024

Ultra-low Power Superconducting Electronics for Deep Learning Accelerator Architectures: Evaluating Energy Efficiency and Scalability

9.22

L. Camron Blackburn, Evan Golden, Tanner Andrulis, Vivienne Sze, Joel Emer, Neil Gershenfeld, Karl K. Berggren

Sponsorship: MIT Lincoln Laboratory, the MIT AI Hardware Program

Since the invention of the Josephson junction in the 1960s, superconducting electronics have shown promise for high-speed and energy-efficient computing. Since 2013, the Adiabatic Quantum Flux Parametron (AQFP) device has gained popularity for its ultra-low energy dissipation. AQFP inverters dissipate 10^{-21}J per switching event, 100× less than other superconductor logic, and 10^{6}× less energy than modern-day CMOS transistors or 10^{3}× when including the cryogenic cooling cost. As Moore's law ends and energy efficiency emerges as a limit on today's computing systems, superconducting AQFP logic is a promising technology to address these energy challenges. Although individual AQFP device performance is impressive, superconducting electronics have failed to replace CMOS systems in the past in part due to the high cost of cryogenic low-noise testing environments and the limitations of superconductor memory scaling. To realize the promise of superconducting electronics, there is a need to architect full systems that can leverage the benefits of the unique superconductor physics (e.g., low-energy logic, low-energy interconnects on zero-resistance wires) while addressing the challenges (e.g., using low-noise cryogenic environments commoditized by the quantum computing industry, constructing a memory hierarchy that addresses the lack of a scalable, high-density superconducting memory). In this work, we extend Timeloop/Accelergy accelerator modeling tools to support superconducting accelerators. This framework explores the design space of deep learning accelerator architectures with a toolbox of superconducting circuits from various logic families. We present results demonstrating the tradeoffs between superconductor vs. CMOS accelerators while running a range of deep learning workloads.

MARC 2025

Assignments and Grading

- **Grading**
  - **Labs 50%**
    - Lab 0: 1 / Lab 1: 12 / Lab 2: 25 / Lab 3: 25 / Lab 4: 25 / Lab 5: 25
  - **Final Project 50%**
    - Milestone 1: 15 / Milestone 2: 10 / Milestone 3: 5
    - Project proposal (due April 17) : 5
    - Final project: 115
- All assignments are due by **11:59PM ET on the due date** (submitted online)
- Labs are to be completed individually, although discussion of course concepts covered in the laboratories is encouraged. Please carefully review collaboration policy at http://csg.csail.mit.edu/6.5930/collaboration.html

Late Policy

- **Late Policy for Labs**
  - You should always submit your labs on time. Nonetheless, since unexpected situations, like illnesses, might occur, you have a budget of 5 late days to spend on the labs. **We will not grant any additional extensions, so please use these days carefully.**
  - The budget is spent in increments of 1 day, and you may not use more than 2 days per lab.
  - **If you submit your lab later than two days after the deadline, or if you are late and have no budget left, your submission will not count towards your grade. However, you must complete all the labs to pass the course.**
  - You do not need to inform us about your use of your budget. The course staff will keep track of the days you have spent.

- *No late days for project due to tight timeline*

Pre-requisites

- 6.191 Computation Structures **and** (6.390 Intro to Machine Learning or 6.300 Signal Processing).
- Students who don't fulfill the prerequisites will be de-registered in the second week of class.

If you believe you have equivalent prior experience (e.g., a computer architecture course taken during your undergraduate studies at another institution), you may petition for consideration. **Please submit your Petition Form by this Friday, Feb 6,11:59 PM ET**