

AI Infrastructure Design for Error-Prone Hardware

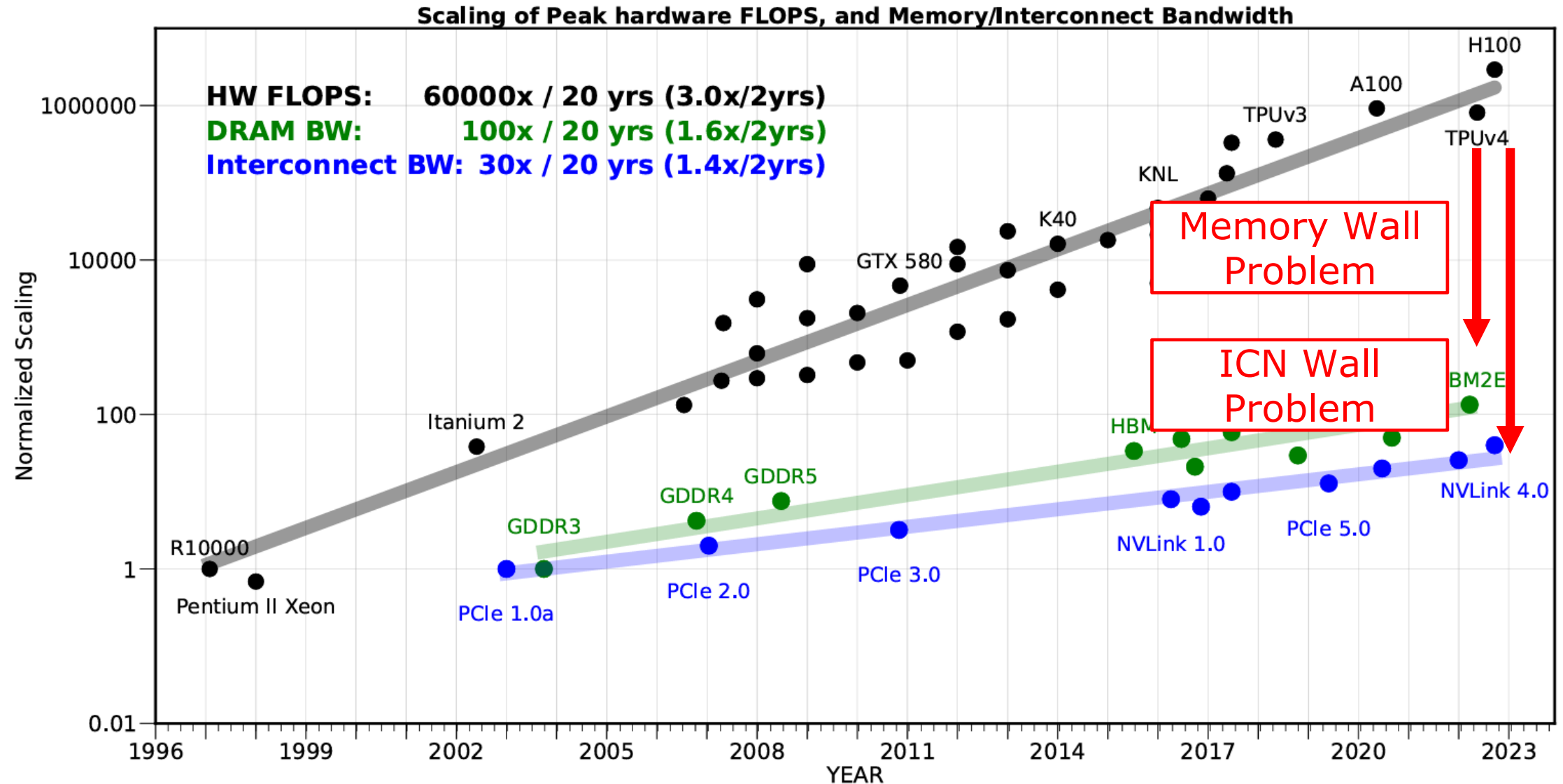
Hunjun Lee

E-Mail: hunjunlee@hanyang.ac.kr

Web: <https://hunjunlee.github.io>

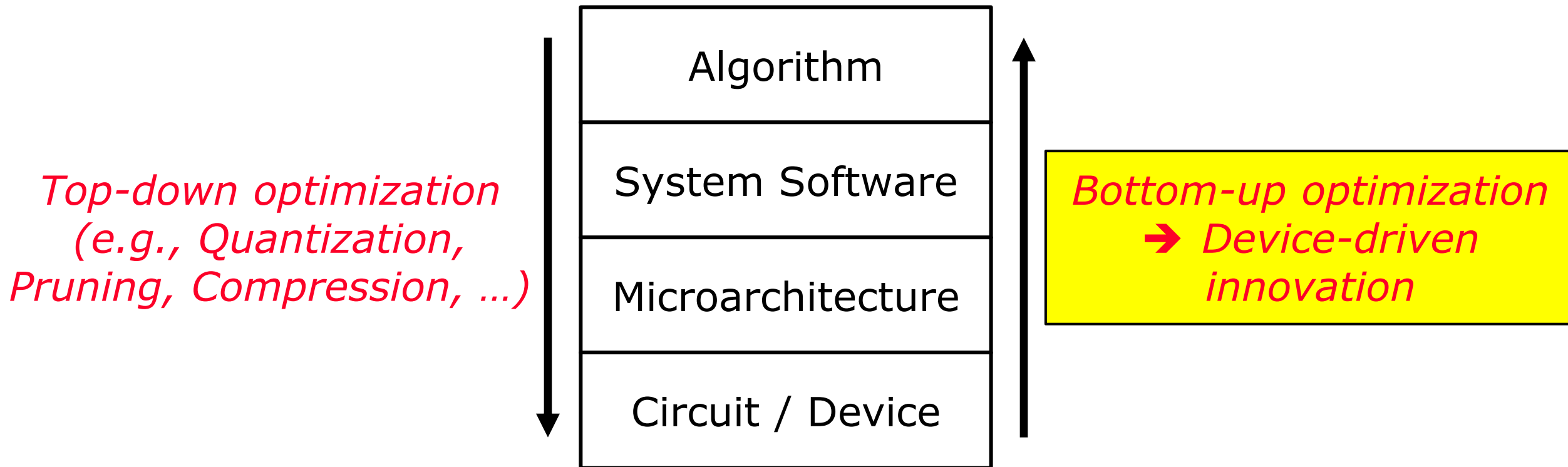
Assistant Professor @**Hanyang University**

Memory & Interconnection Wall Problem



Cross-Optimization of Computing Problem

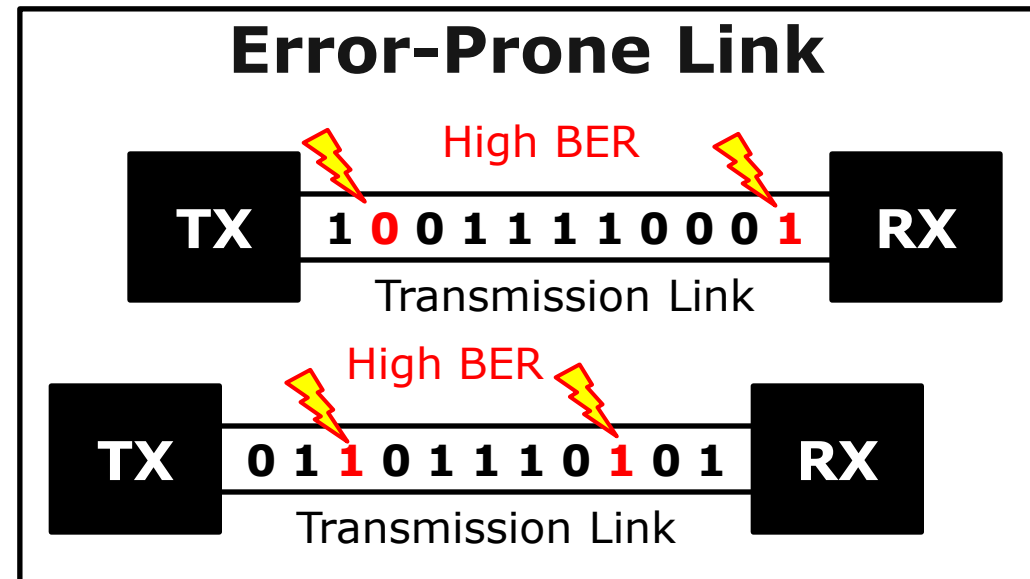
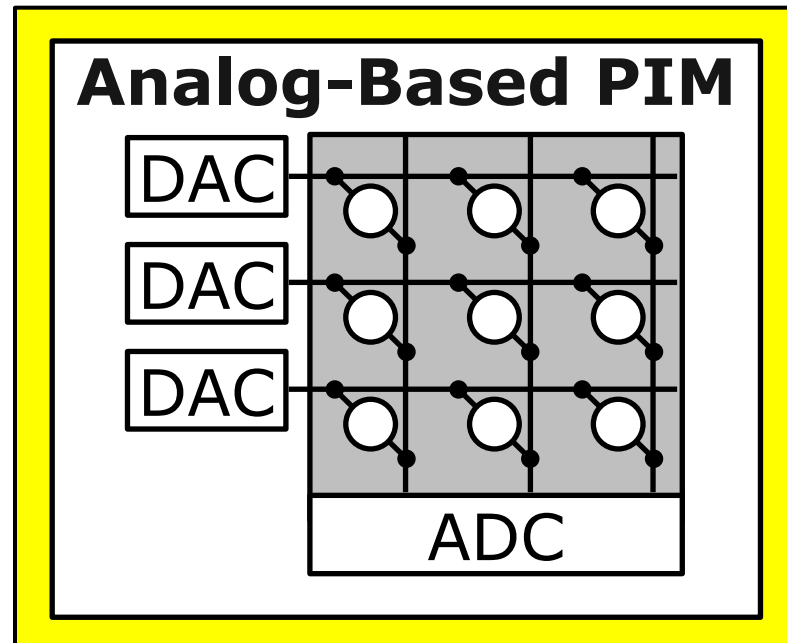
- We need to explore both top-down and **bottom-up approach** towards an **efficient AI system**



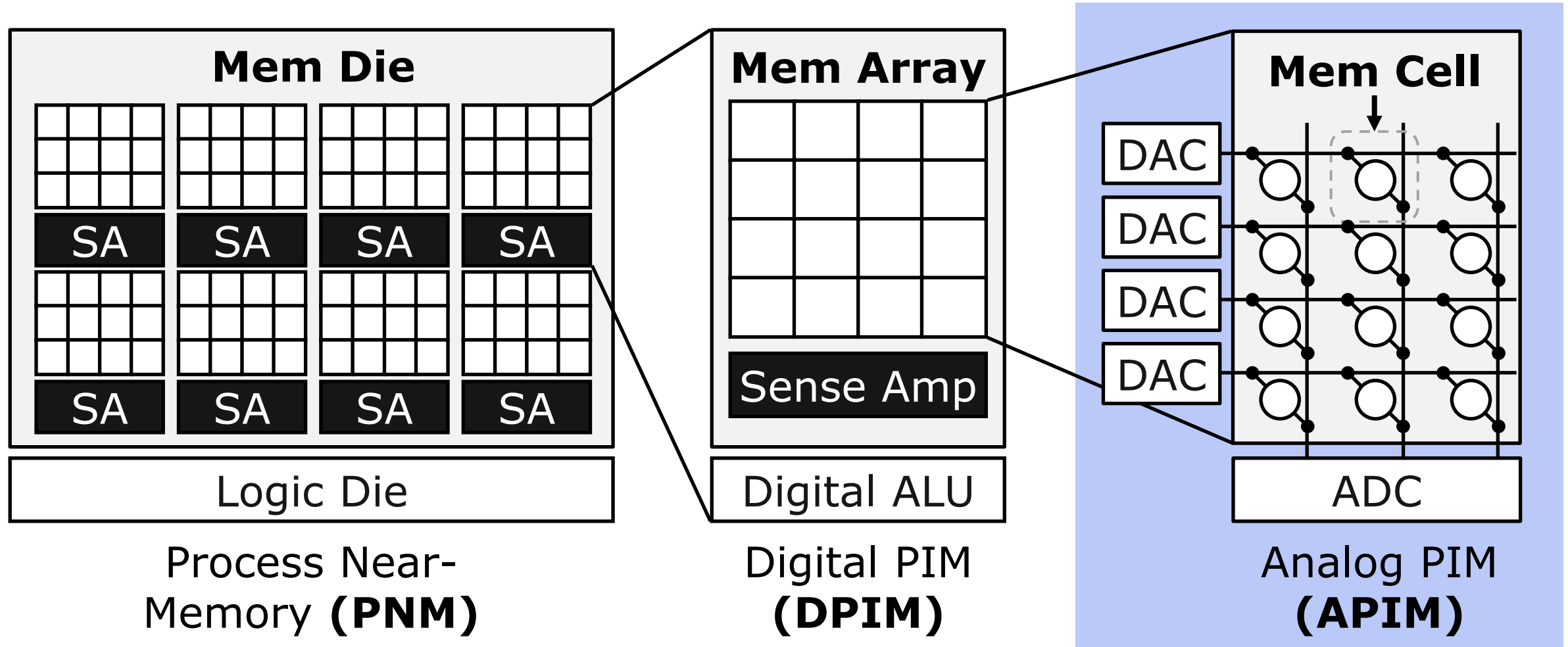
Systematically explore the bottom-up approach!

Bottom-Up Optimization Opportunities

- **Memory-Wall:** Adopt analog-based process-in-memory unit (neuromorphic processor) maximizes the memory efficiency
- **Interconnection-Wall:** Adopt an error-prone interconnection (High BER) to mitigate the excessive communication overhead



Analog-Based PIM (APIM) Architecture

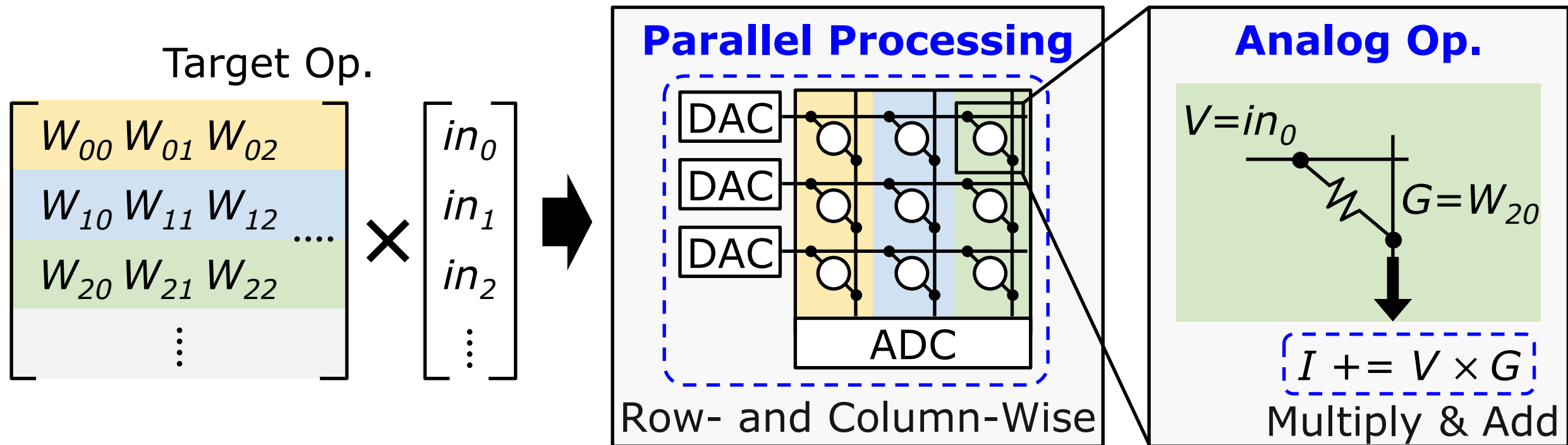


Higher Parallelism & Energy Efficiency →

Limited Applicability & Lower Reliability →

Matrix-Vector Multiply Operations in APIM

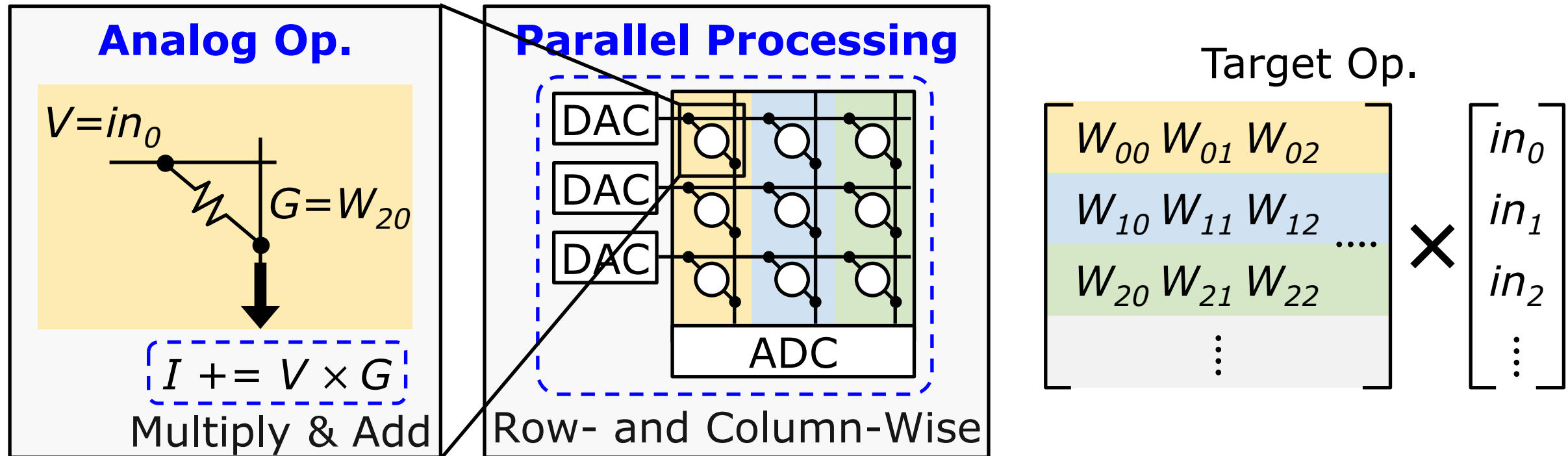
- Uses a memory array for a **parallel and energy-efficient operation** (e.g., matrix-vector multiplication)



Offloading MVM operations to the analog PIM potentially breaks the memory wall w/ high efficiency

Matrix-Vector Multiply Operations in APIM

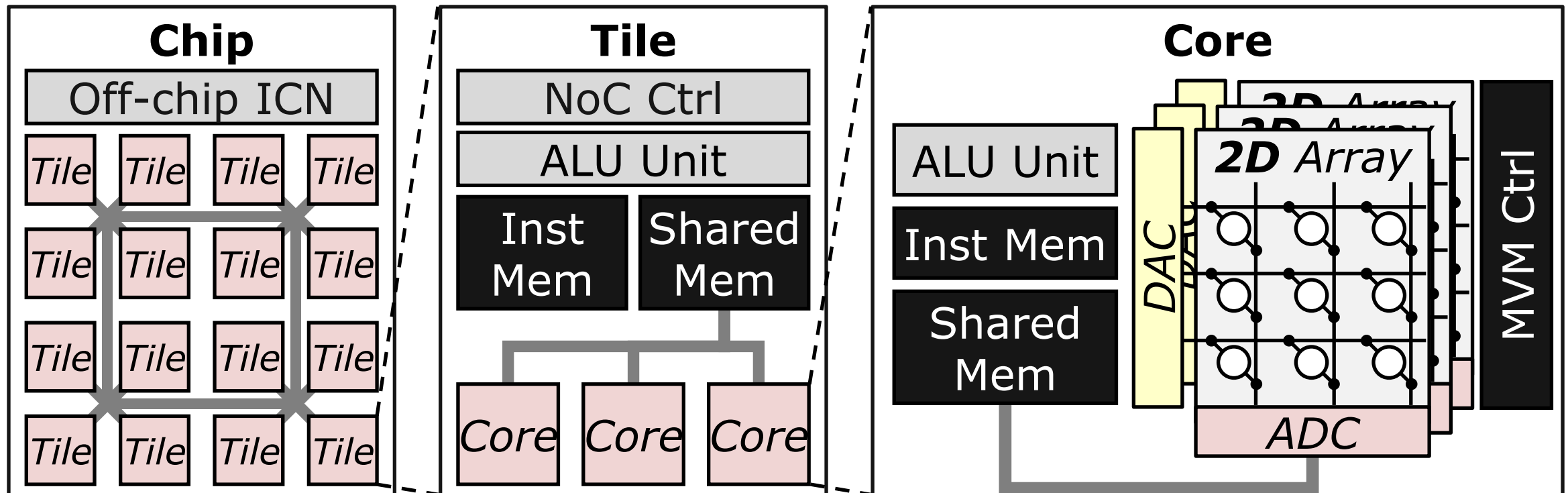
- Uses a memory array for a **parallel and energy-efficient operation** (e.g., matrix-vector multiplication)



Offloading MVM operations to the analog PIM potentially breaks the memory wall w/ high efficiency

APIM-based Architecture Design

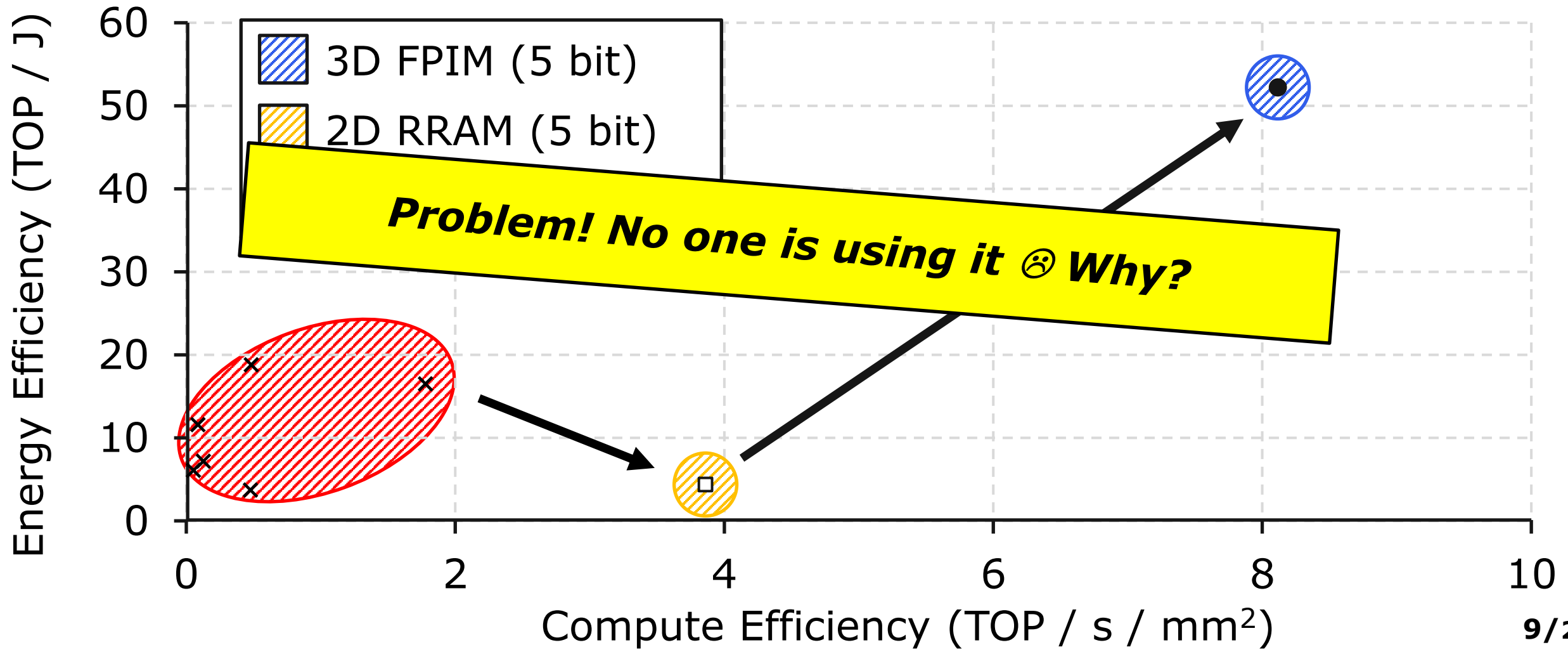
- There are various efforts to design a **highly efficient architecture for APIM-based compute units**



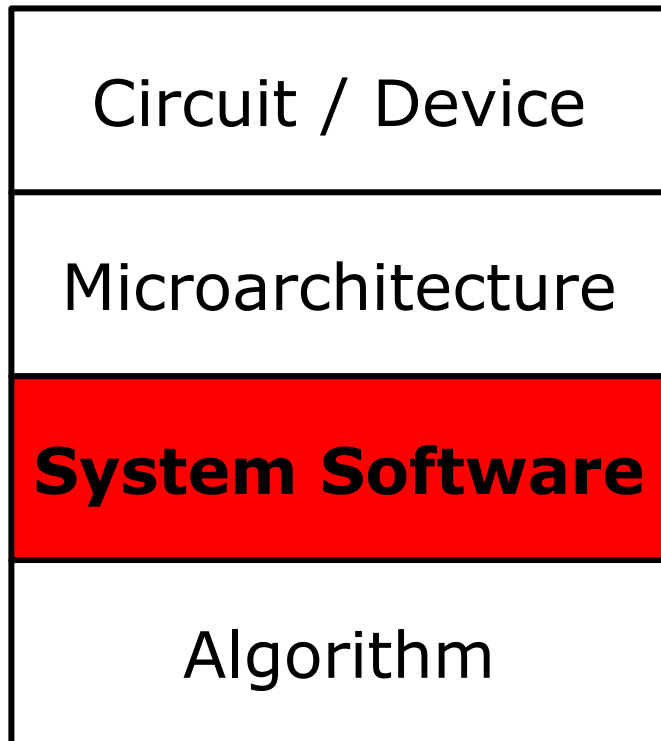
ISCA16 ISAAC
ISCA ... PRIME, ...

APIM-based Architecture Design

- There are various efforts to design a **highly efficient architecture for APIM-based compute units**



Why? We Need a Full-Stack System



- **Device- & Circuit-Level Optimization**

- Use a NAND or NOR Flash device
- 3D/4D integrated architecture

- **Architecture-Level Optimization**

- Inter-layer Pipelined architecture
- Analog summation to reduce ADC overhead

- **System-Level Optimization**

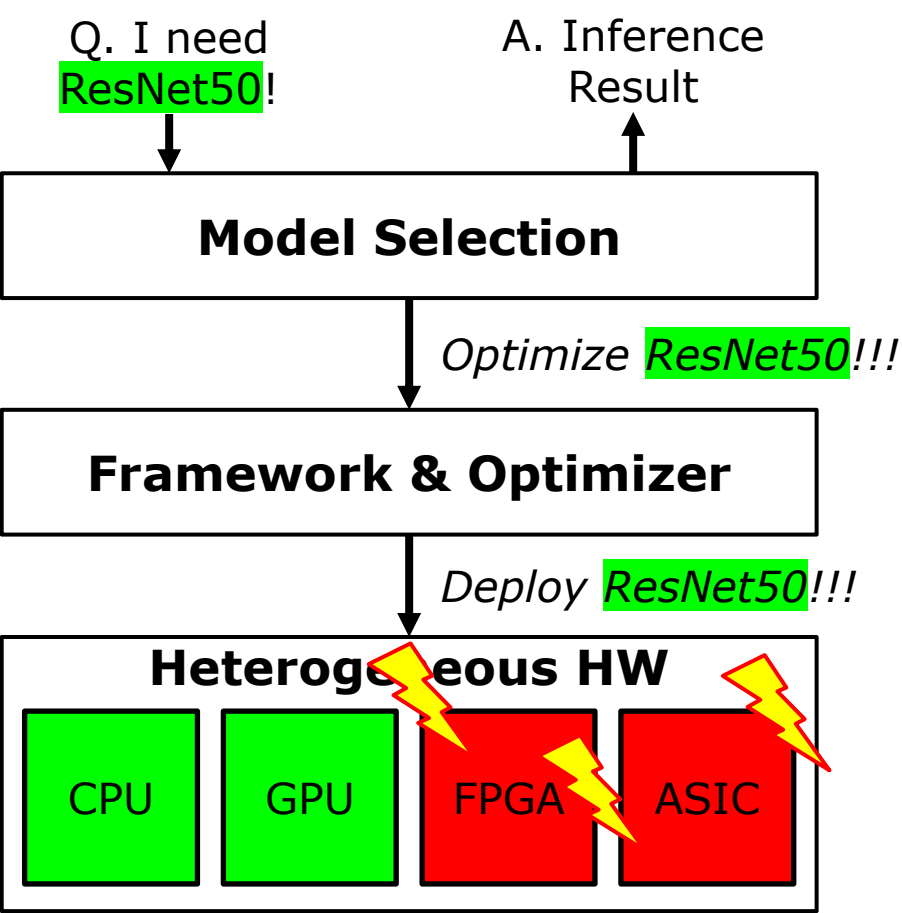
- How to design a system for an erroneous hardware?

- **Algorithm-Level Optimization**

- Variation & fault aware mapping
- Noise-aware retraining

Background #1: Modeless Serving Framework

- Instead of deploying a “specific model”, we select from **a set of models to achieve a target accuracy**

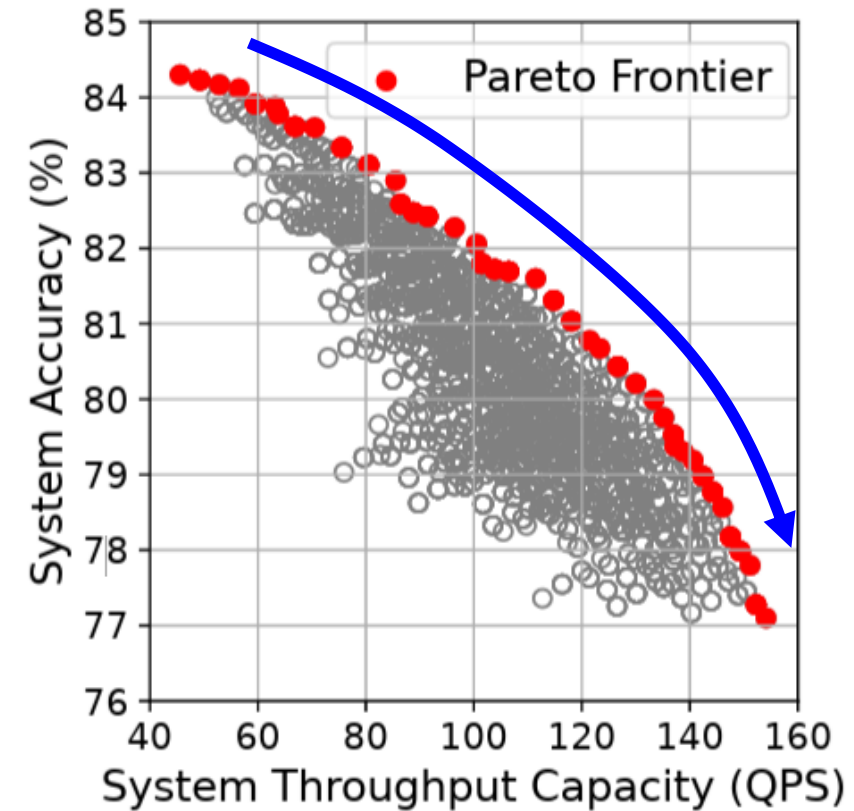
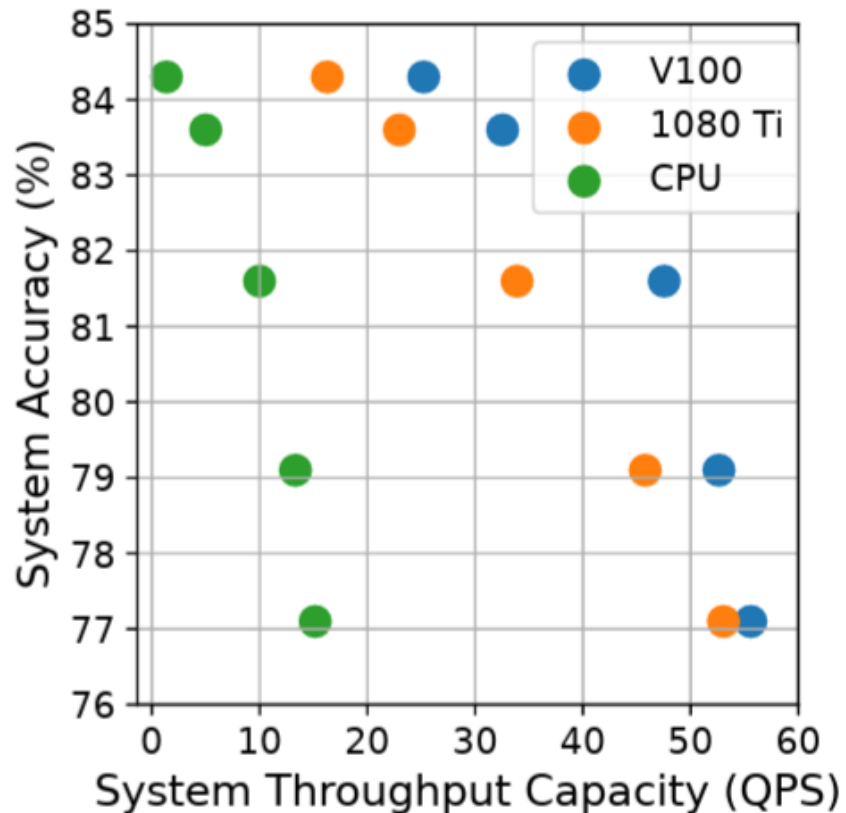


Q. I need 70% accuracy for Image Classification A. Inference Result

Model Family	Model Variants
ResNet (classification) [20]	18, 34, 50, 101, 152
DenseNet (classification) [22]	121, 161, 169, 201
ResNeSt (classification) [47]	14, 26, 50, 269
EfficientNet (classification) [40]	b0-b7
MobileNet (classification) [21]	1.0, 0.75, 0.5, 0.25
YOLOv5 (object detection) [25]	n, s, m, l, x
BERT (sentiment analysis) [11]	RoBERTa-base, large; [29], ALBERT-base, large, xlarge, xxlarge [26]; BERT-base, tiny, mini, small, medium, large [41]
T5 (translation) [34]	small, base, large, 3b, 11b
GPT-2 (question answering) [33]	base, medium, large, xl

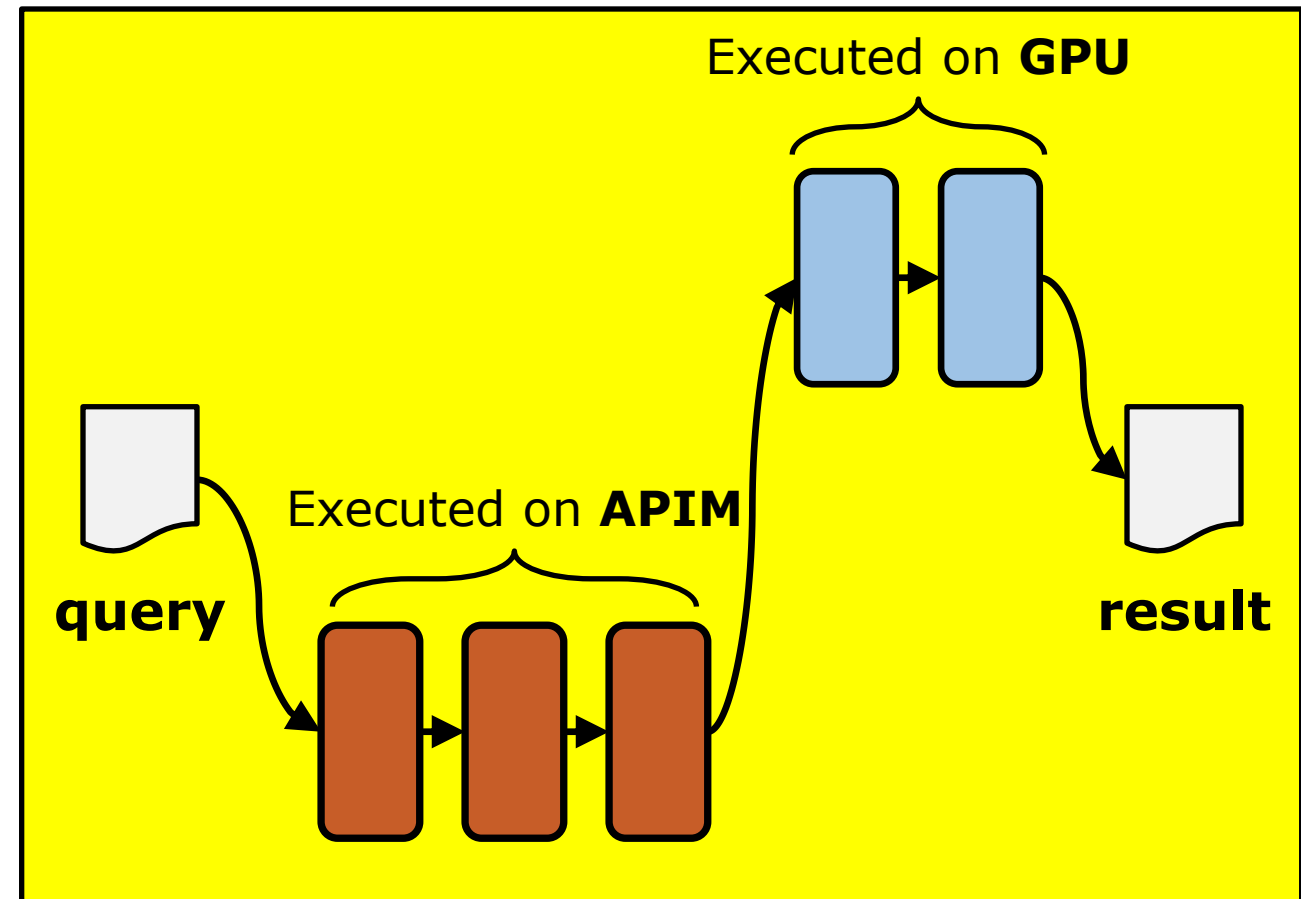
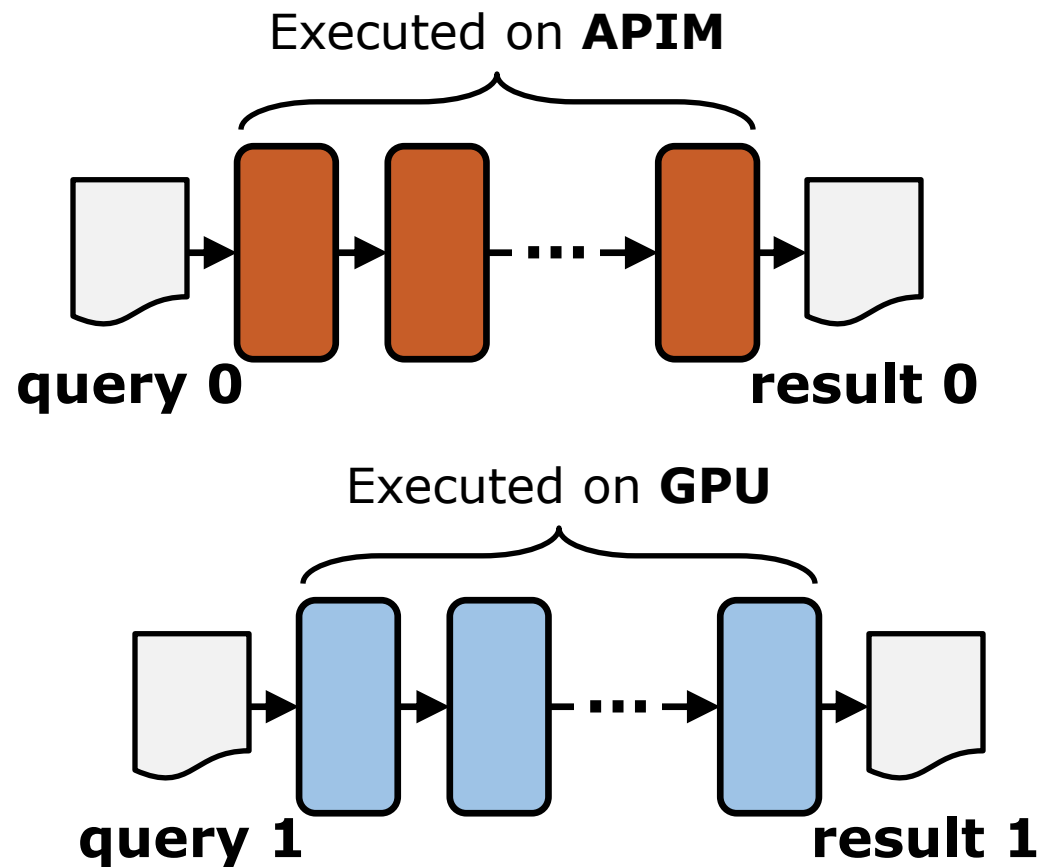
Background #2: Accuracy Scaling Framework

- For a given computing system, we can **scale the performance at the cost of the accuracy degradation**

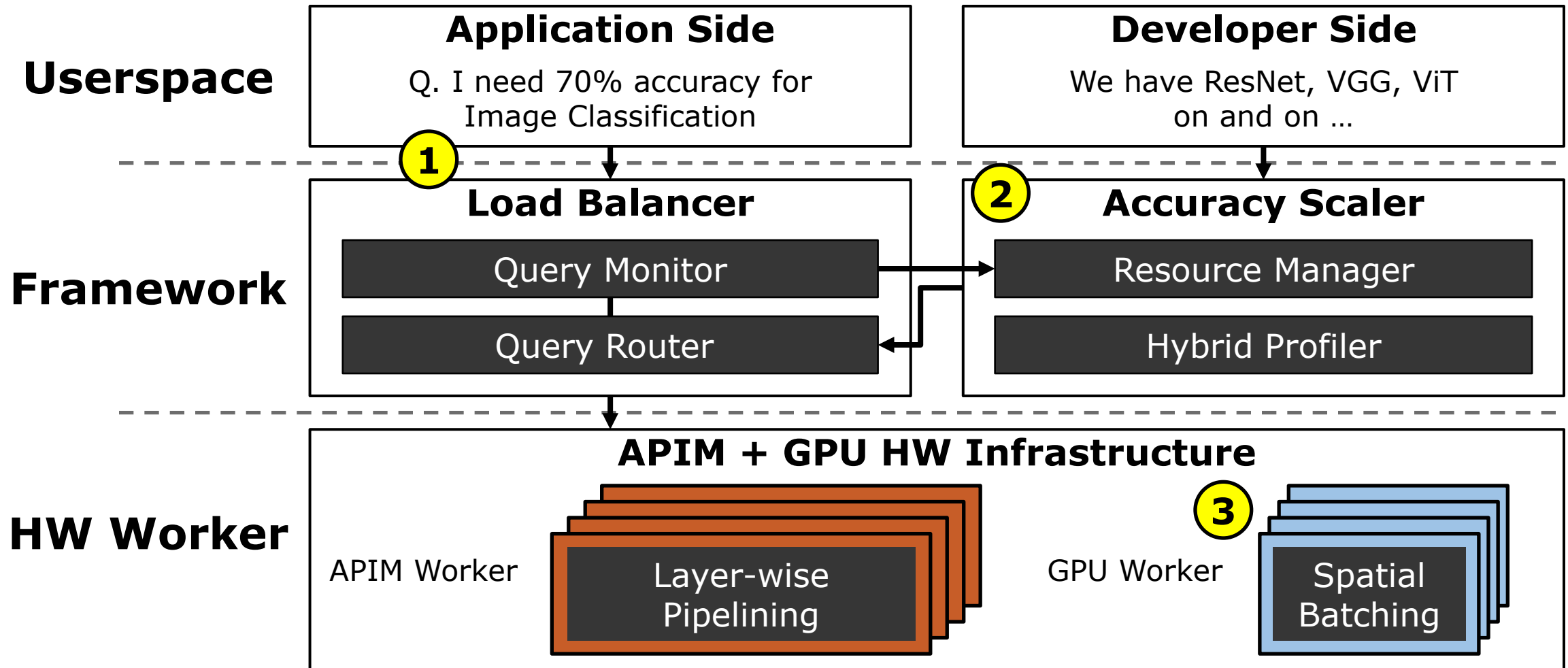


Motivation: Hybrid APIM and GPU System

- Propose both network-wise and layer-wise scheduling between **the analog and digital compute units**

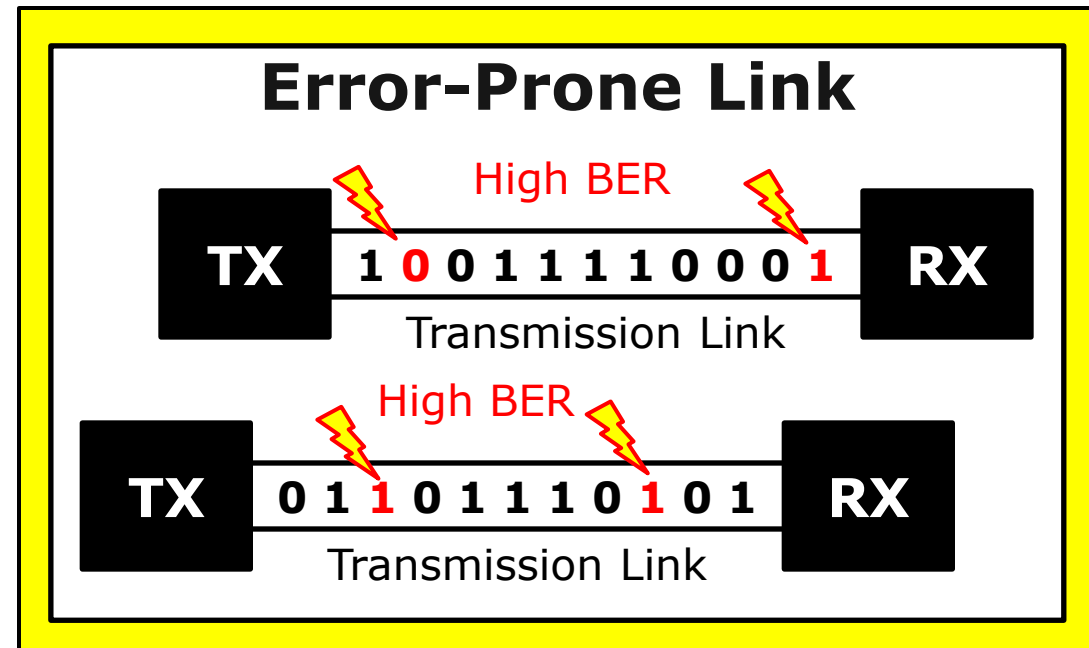
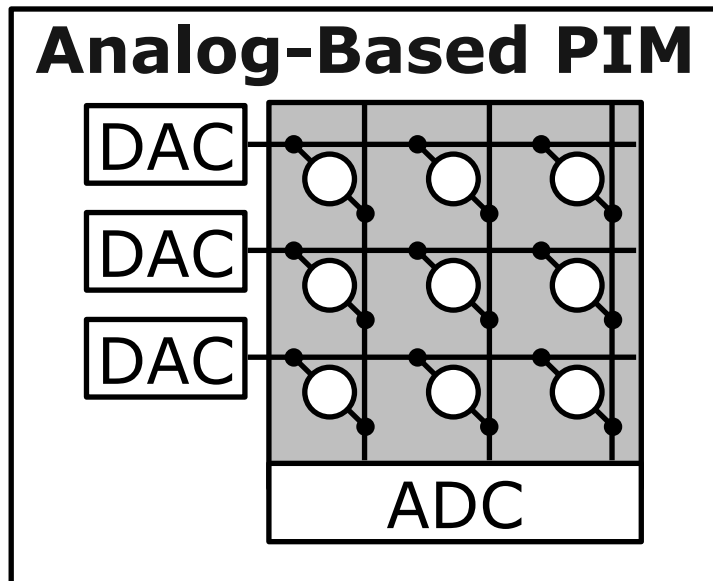


Key Idea: Software Framework for Deploying NNs on APIM-based System



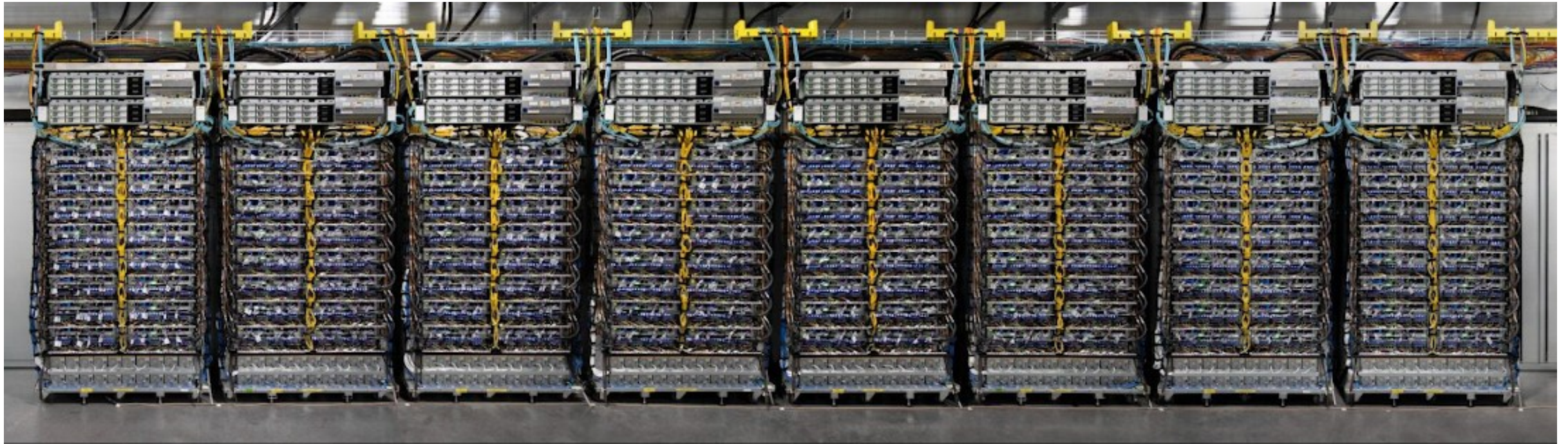
Bottom-Up Optimization Opportunities

- **Memory-Wall:** Adopt analog-based process-in-memory unit (neuromorphic processor) maximizes the memory efficiency
- **Interconnection-Wall:** Adopt an error-prone interconnection (High BER) to mitigate the excessive communication overhead



Modern AI System (e.g., TPU V4)

- Recently, AI systems are scaling up at an extremely fast rate that demands intra-server and inter-server communication



There is an excessive amount of communication system for deploying large-scale neural networks!!!

Modern AI System (e.g., TPU V4)

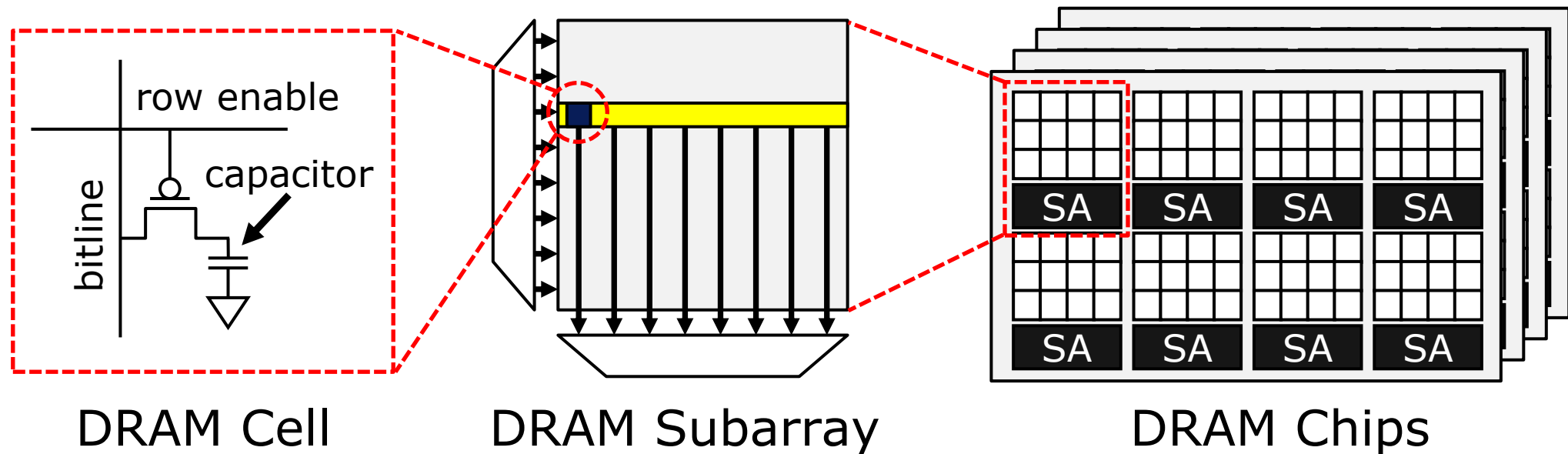
- Recently, AI systems are scaling up at an extremely fast rate that demands intra-server and inter-server communication



How to solve this problem?
→ Memory-Side Case Study

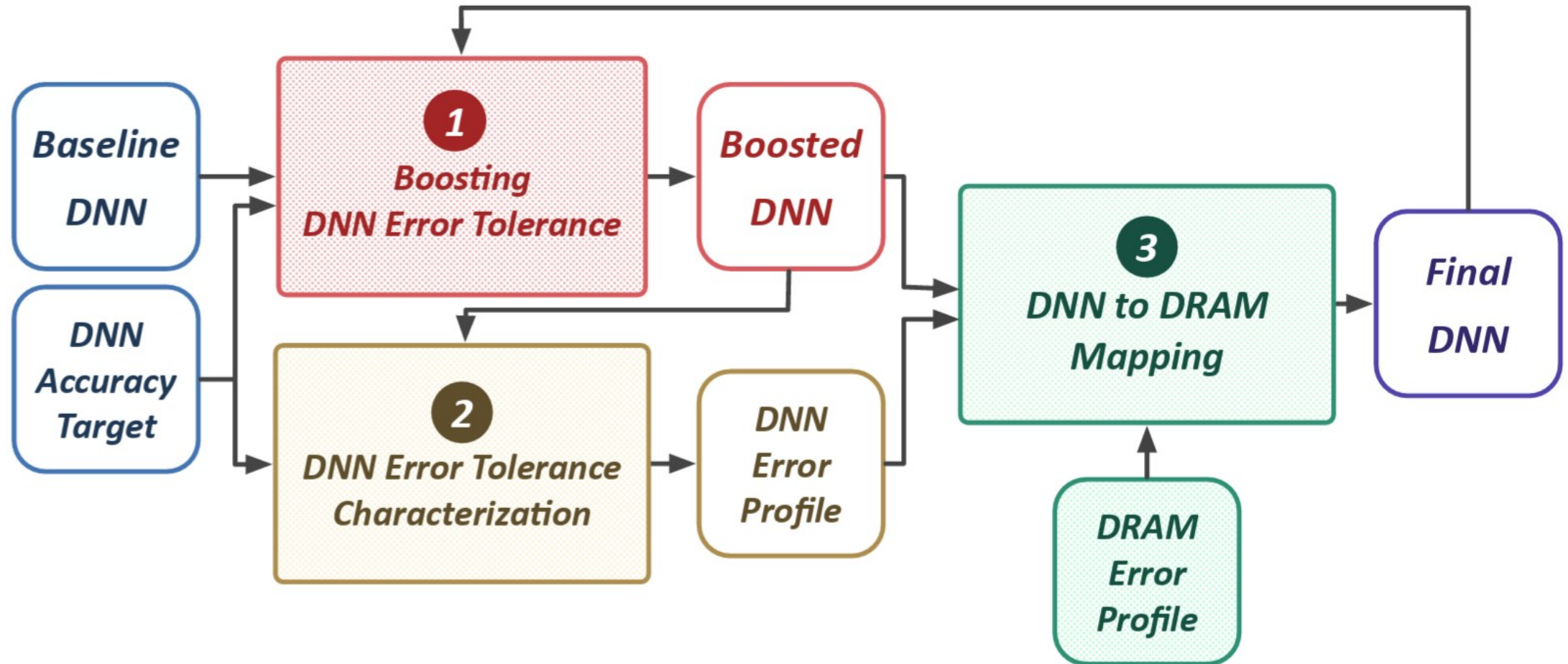
Background #1: Approximate DRAM

- There are various **approximation methods to reduce the power consumption** of the DRAM
 - Option #1. Reduce DRAM voltage → May not meet the timing margin
 - Option #2. Reduce DRAM refresh rate → May corrupt the stored data



***Sacrifice DRAM reliability for
higher performance and energy efficiency***

EDEN: DNN for Approximate DRAM



We need a full-stack approach to use the approximate DRAM in the DNN domain

What about Interconnection?

Approximation in ICN is Coming ...

- *This is a near future problem!!!*

The PCI Express® (PCIe®) 6.0 specification will feature two primary mechanisms to correct errors: Forward Error Correction (FEC) and Cyclic Redundancy Check (CRC). Pulse Amplitude Modulation 4-level (PAM4)—PAM4 is a four-level modulation format. There are four distinct amplitude levels within an electrical or optical data lane or channel such that each amplitude level represents two bits

of data. As a result, PAM-4 modulation can transmit twice the speed of optical components. However, the use of PAM4 results in a lower signal-to-noise ratio (SNR). SNR is reduced as the distance increases compared to binary NRZ modulation. This results in a theoretical SNR penalty of 3 dB. Because of this difference in SNR that both DSP and FEC are required to maintain the same error rate.

Forward error correction (FEC) is a channel coding technique that adds redundancies. It is designed such that information needed to correct errors can be received at the receiving end of the link. FEC is enabled by default on all PCIe 6.0 links.

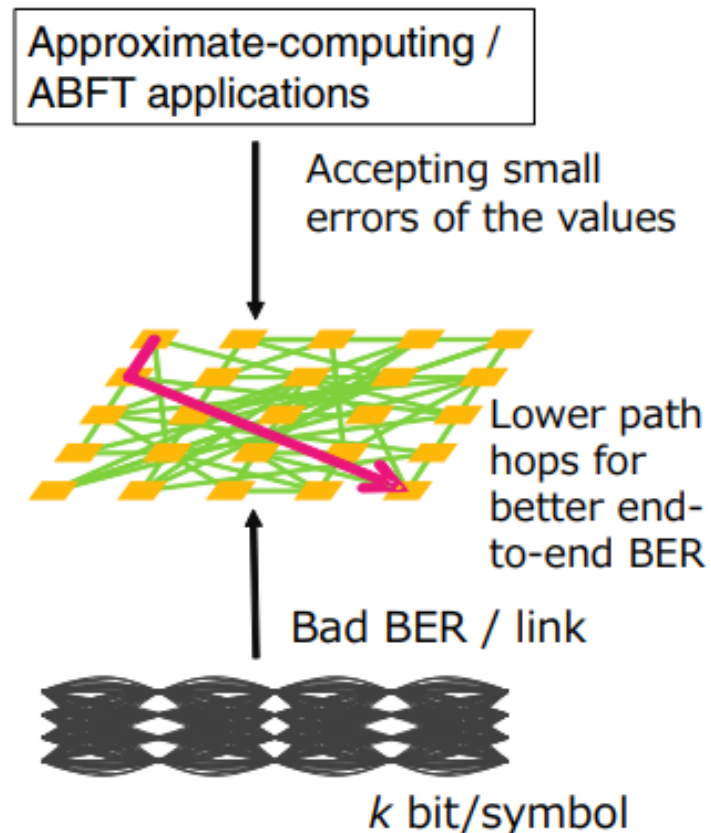
1 Evolving standards impact the ecosystem.

Standards for 800G PCS and adaptation of 400G FEC for 800G continue to evolve, bringing small changes but also some major leaps. While 800G implementations using 100G lanes rely on PAM4 signal encoding from prior generations, these rollouts will need to account for a doubling of PAM4 speeds, from 50Gb/s to 100Gb/s. Future strategies anticipate getting to 800G via four 200G lanes rather than eight 100G lanes. This would be a major change that would ultimately introduce additional technical challenges.

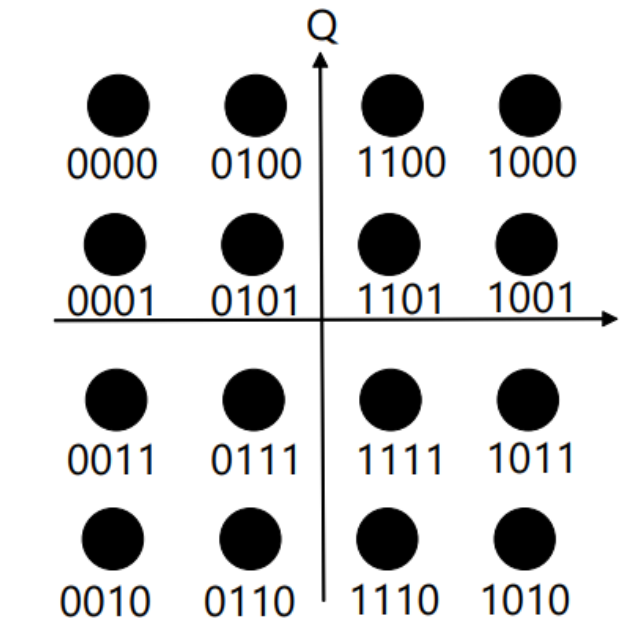
Everyone is now seeing this problem!!!

Exploiting Approximate Computing in Link

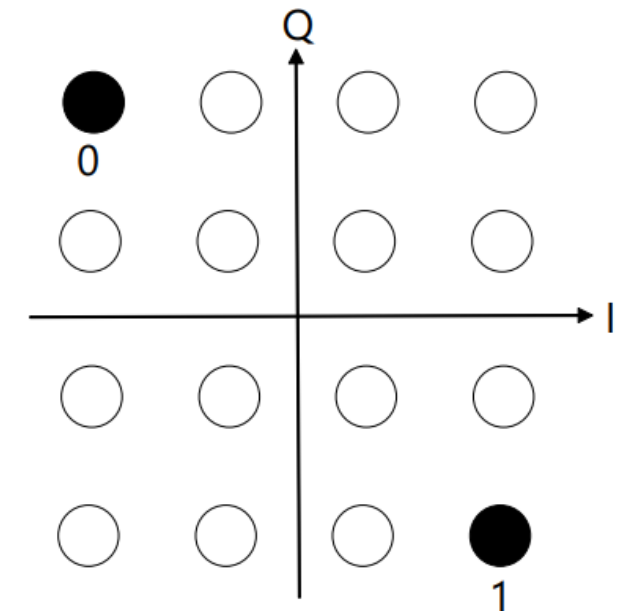
- We can design a hybrid system that supports **both conservative networks and approximate networks**



Approximate Network



(a) 4-bit Mapping
(Low reliability but high bandwidth)

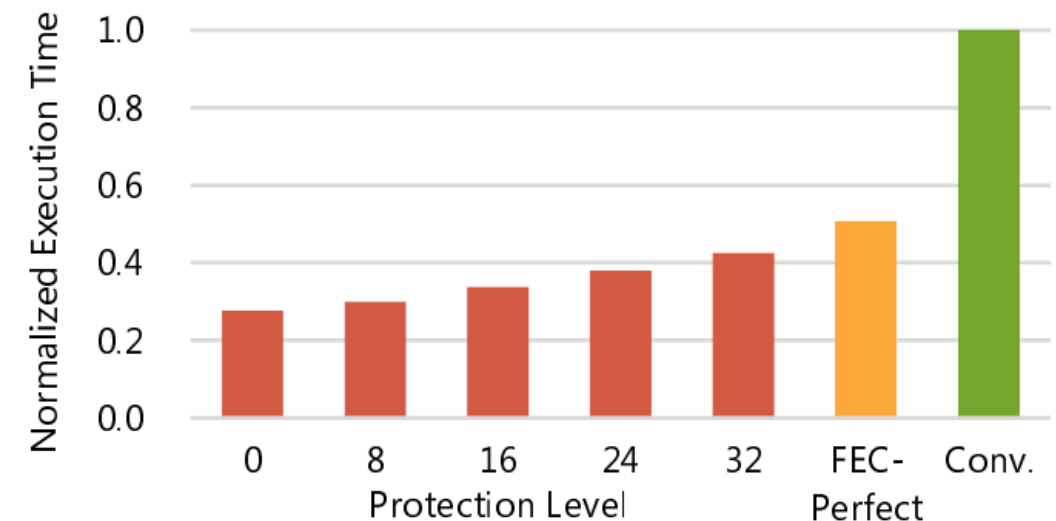
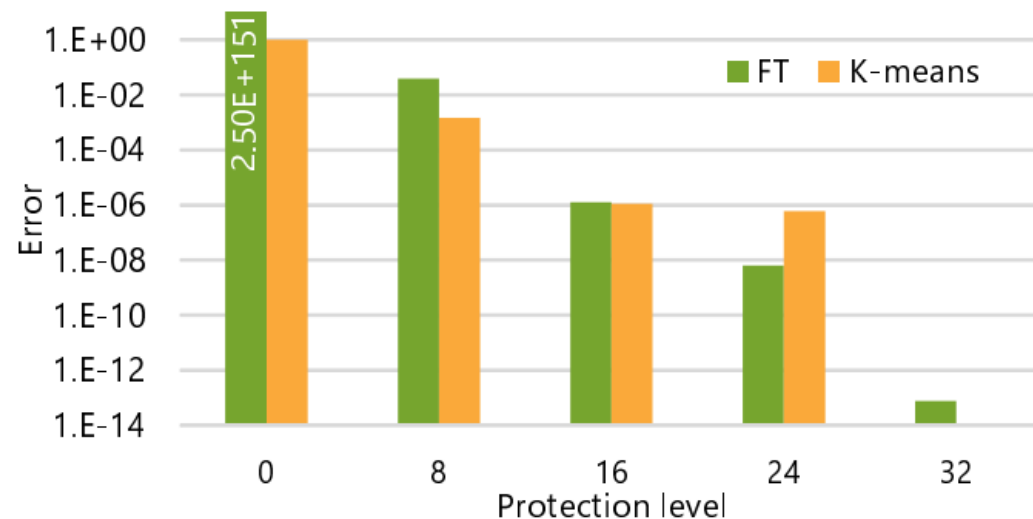


(b) 1-bit Mapping
(High reliability but low bandwidth)

Flexible Encoding Method

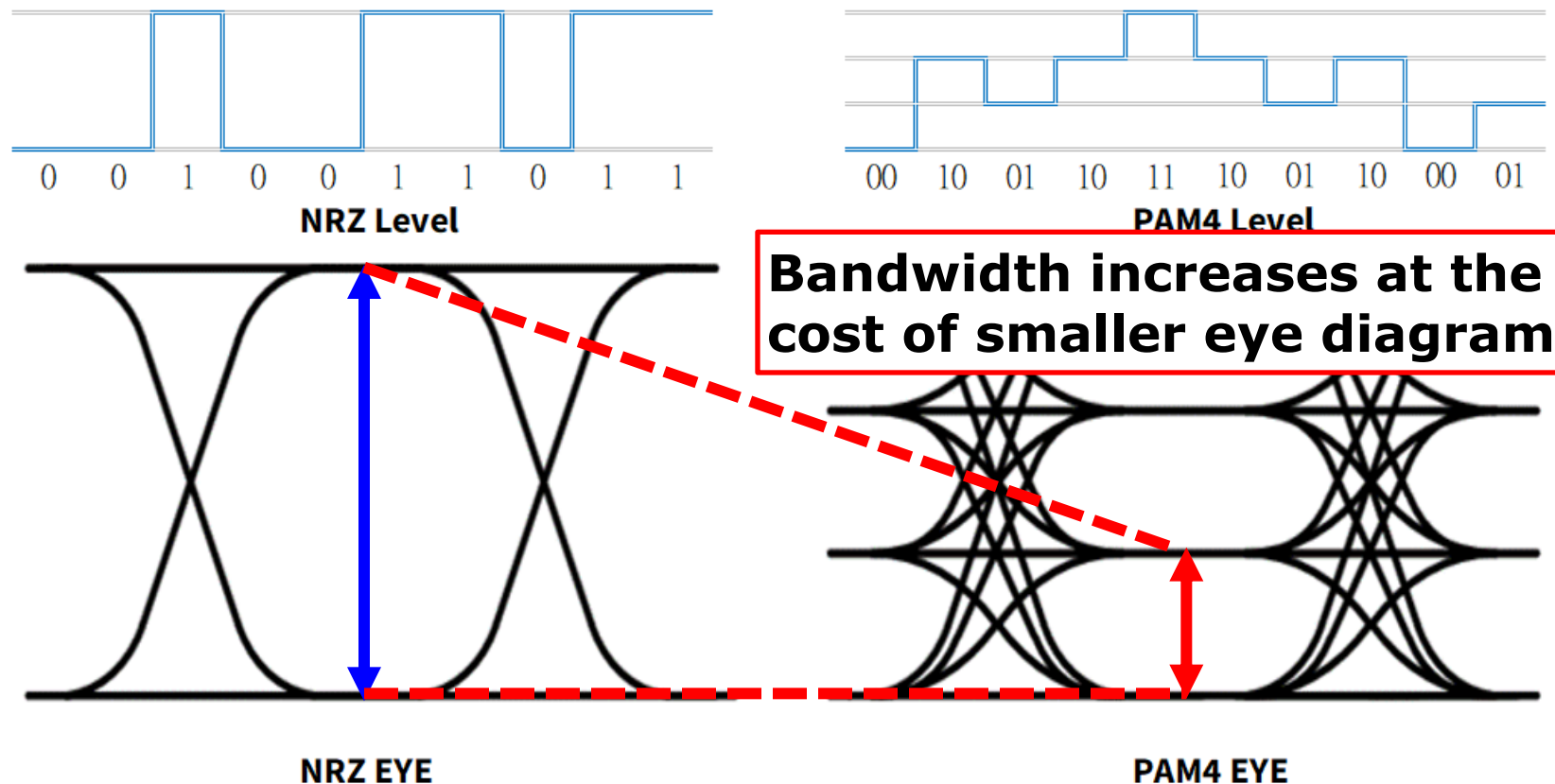
Key Insight: Value Error vs. Bit Error

- We should **differentiate value errors vs. bit errors** when using an error prone system
 - A single bit flip **may or may not critically affect the value** depending on the position of the flip
 - Bit **flips at MSBs affect the value by a greater amount**



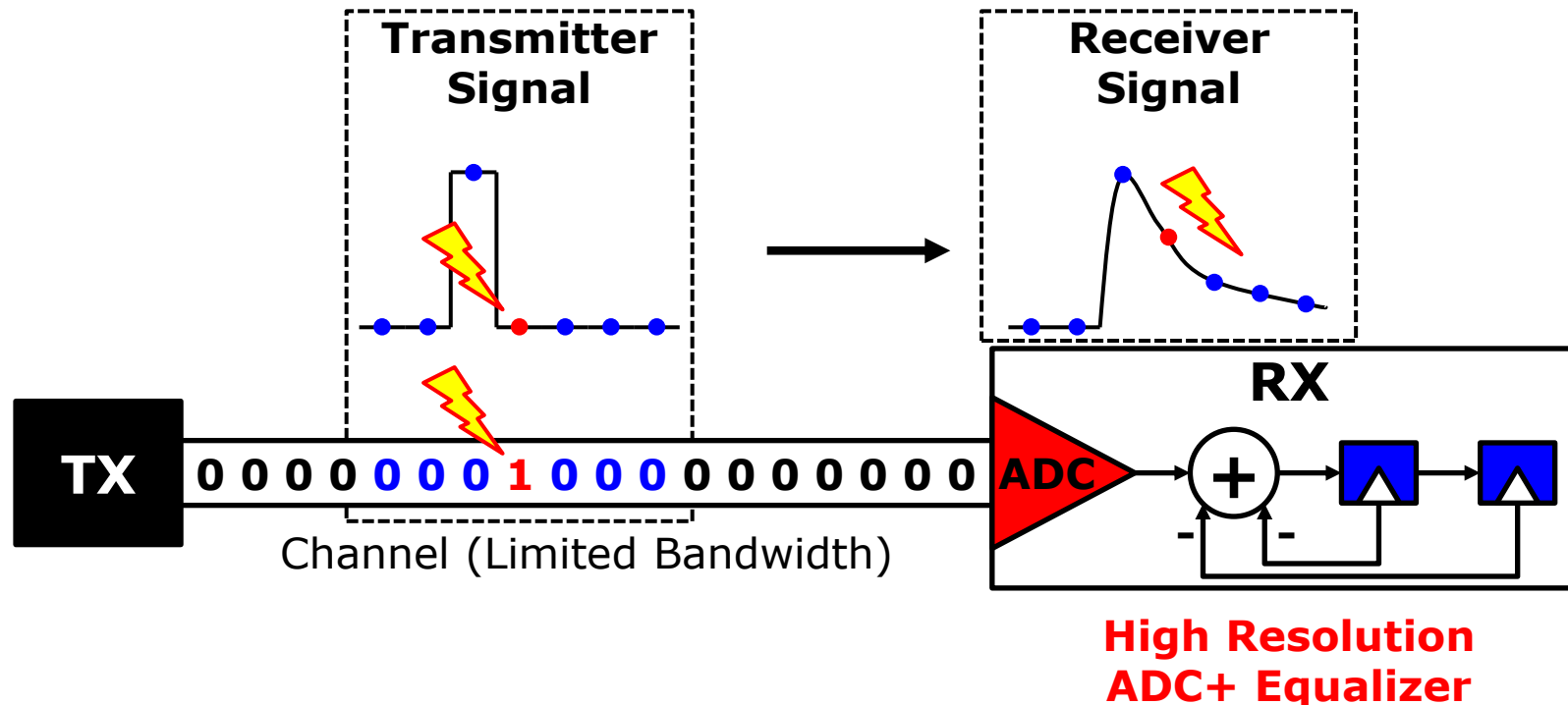
Configurable Parameters in ICN - 1

- The ICN is increasing the number of levels to increase the BW
 - From NRZ -> PAM4 -> PAM8
 - Why not **increase the data rate at the cost of smaller eye diagram?**



Configurable Parameters in ICN - 2

- We can decrease the **ADC + Equalizer** resolution to minimize the power consumption
 - Can we **reduce the ADC precision at the cost of higher BER?**

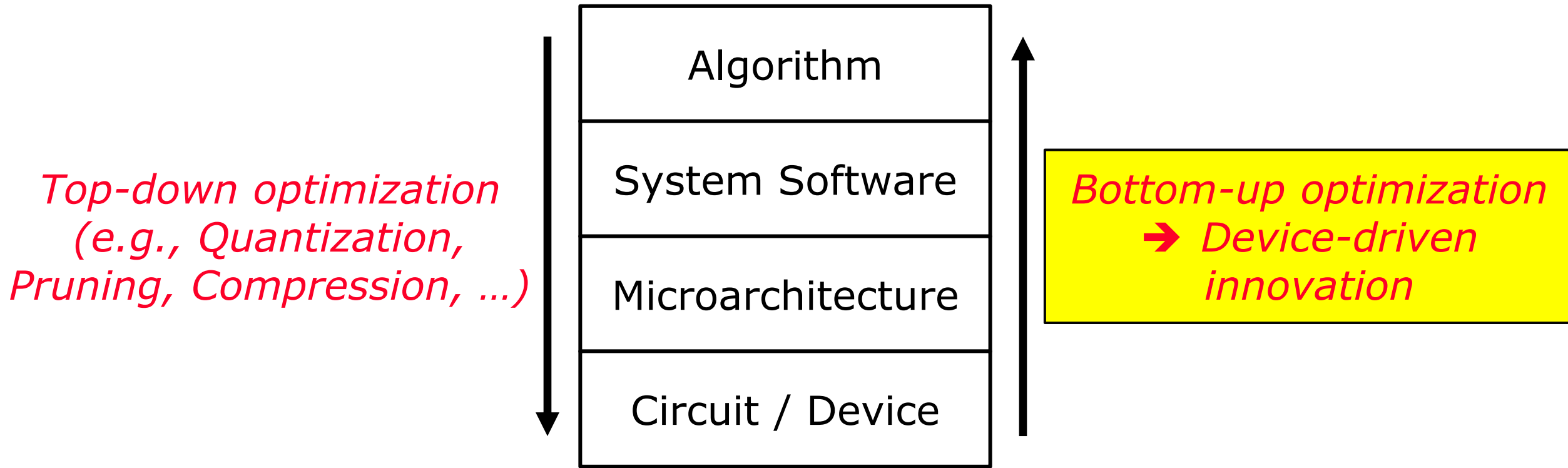


What else?
Feedforward Error Correct?
Operating Voltage?
Data Transmission Pattern?

We Need COLLABORATION

Again, and Again ... Full-Stack System

- We need a **full-stack system redesign** across algorithm, system software, microarchitecture, and circuit/device



Thank You!
Any Questions?

Hunjun Lee

Computer Architecture Lab (CArch)
Department of Computer Science
Hanyang University

E-mail: hunjunlee@hanyang.ac.kr