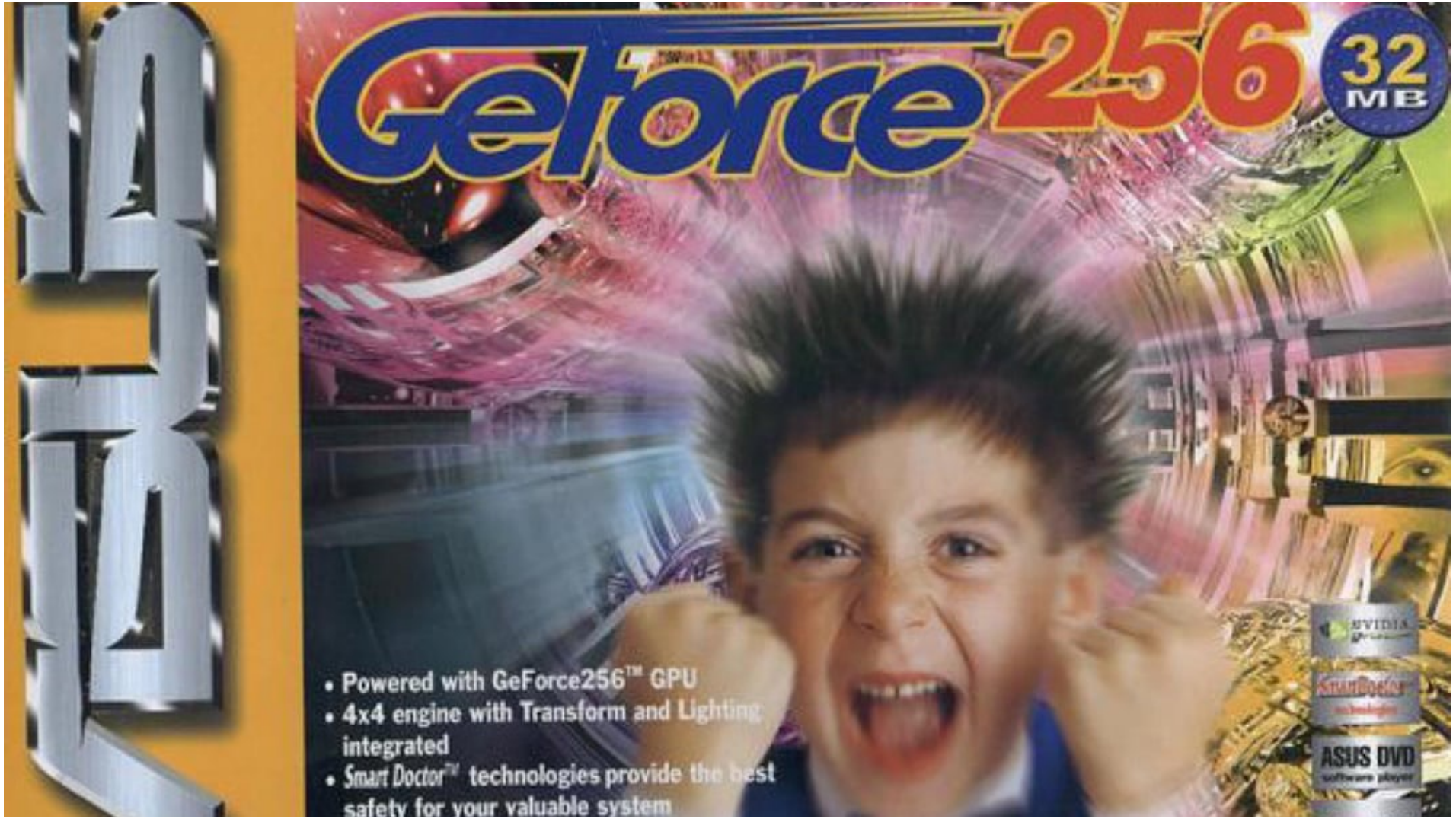
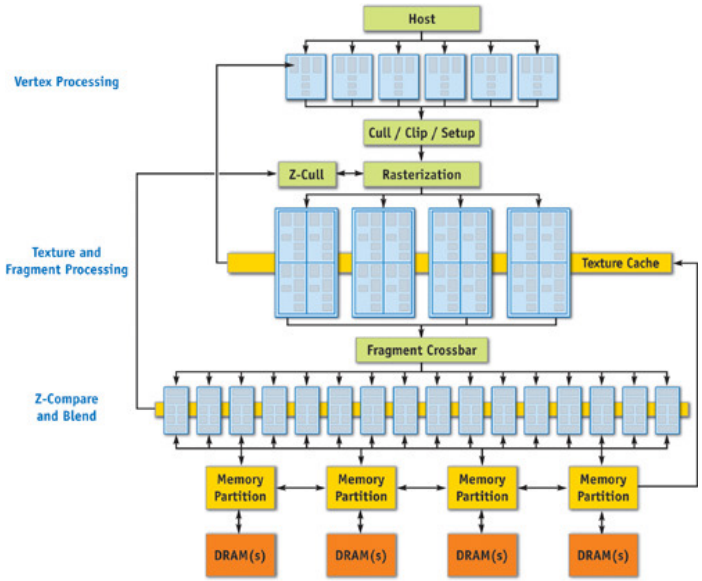
The Reconfigurable Future of Accelerators (or Wishful Thinking?)

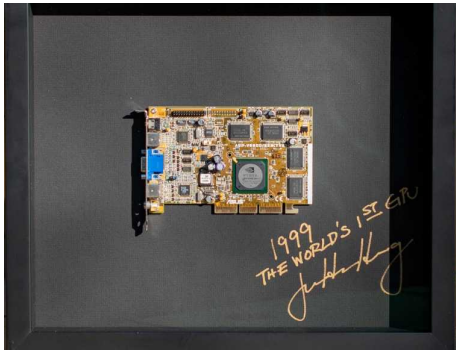
Tulika Mitra
National University of Singapore

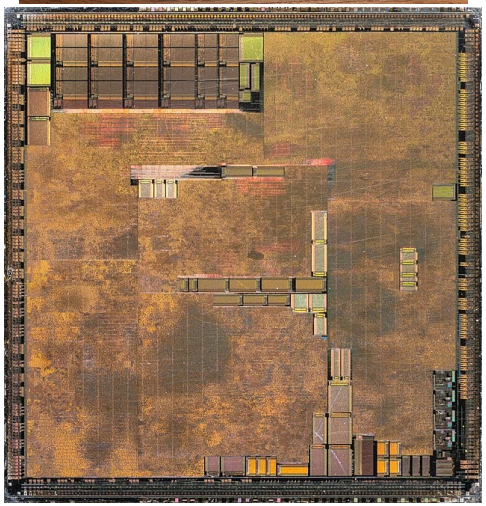
Keynote at 35th International Conference on Field-Programmable Logic and Applications (FPL) 2025


GeForce256
32 MB
• Powered with GeForce256™ GPU
• 4x4 engine with Transform and Lighting integrated
• Smart Doctor™ technologies provide the best safety for your valuable system
NVIDIA
SmartDoctor
ASUS DVD

World's First GPU Accelerator (1999)

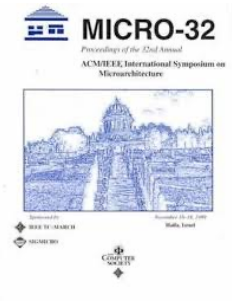






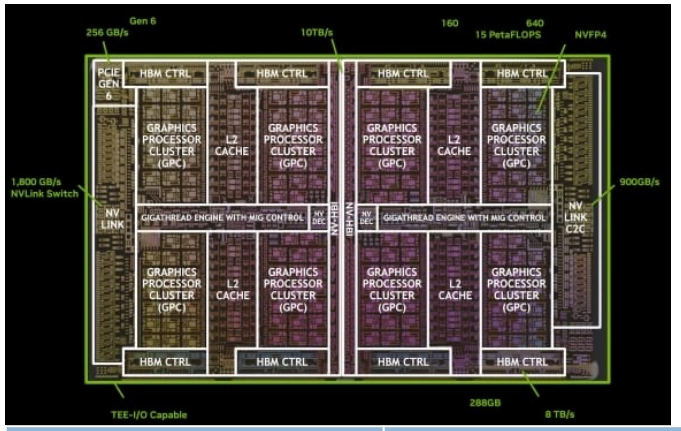
Dynamic 3D Graphics Workload Characterization and the Architectural Implications

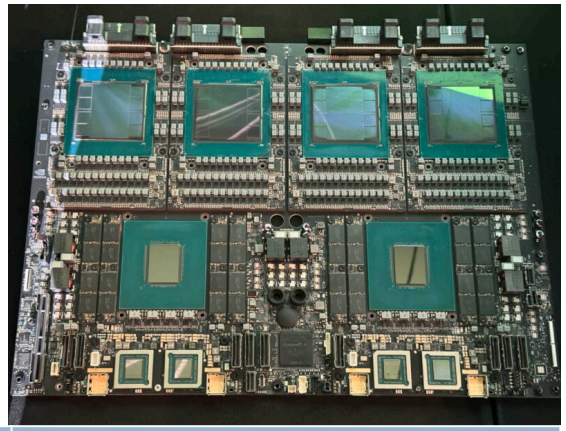
Tulika Mitra Tzi-cker Chiueh
Computer Science Department
State University of New York at Stony Brook
Stony Brook, NY 11794-4400
{mitra, chiueh}@cs.sunysb.edu



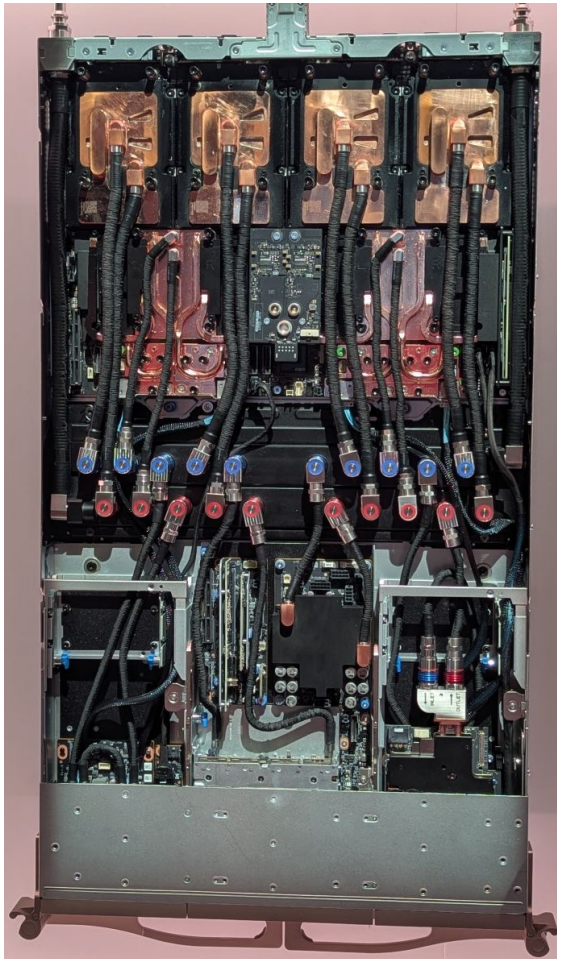
TSMC 220 nm
17 million transistors
32-64 MB memory
4.8 GB/s memory bandwidth
13 W power

GPU in 2025:
Blackwell Ultra B300 AI Accelerator



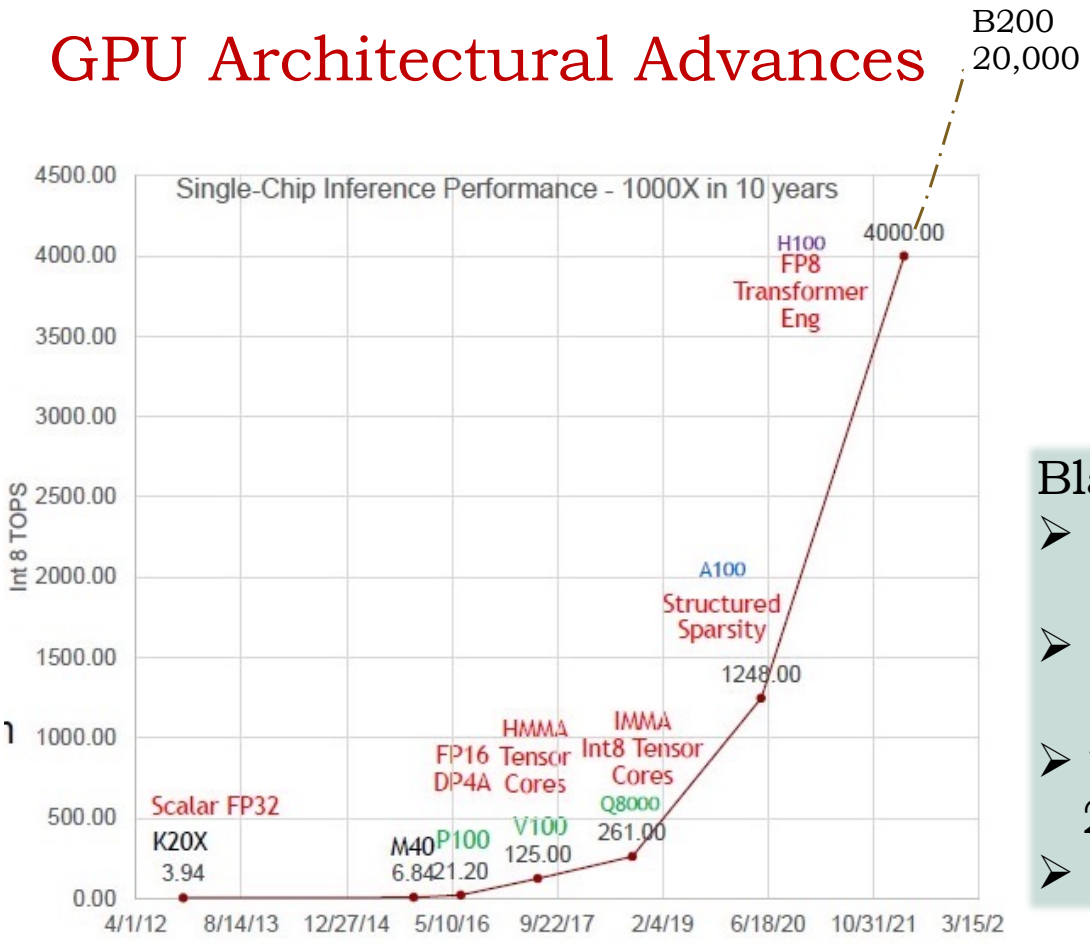


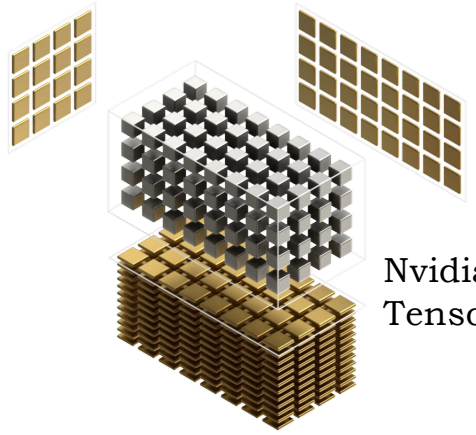
	1999	2025
Technology Node	TSMC 220nm	TSMC Custom 4NP (5nm)
Transistors	17 million	208 billion
Memory per GPU	32-64 MB	288 GB
Memory BW	4.8 GB/s	8 TB/s
Power per GPU	13 W	1400 W



NUS National University of Singapore

GPU Architectural Advances





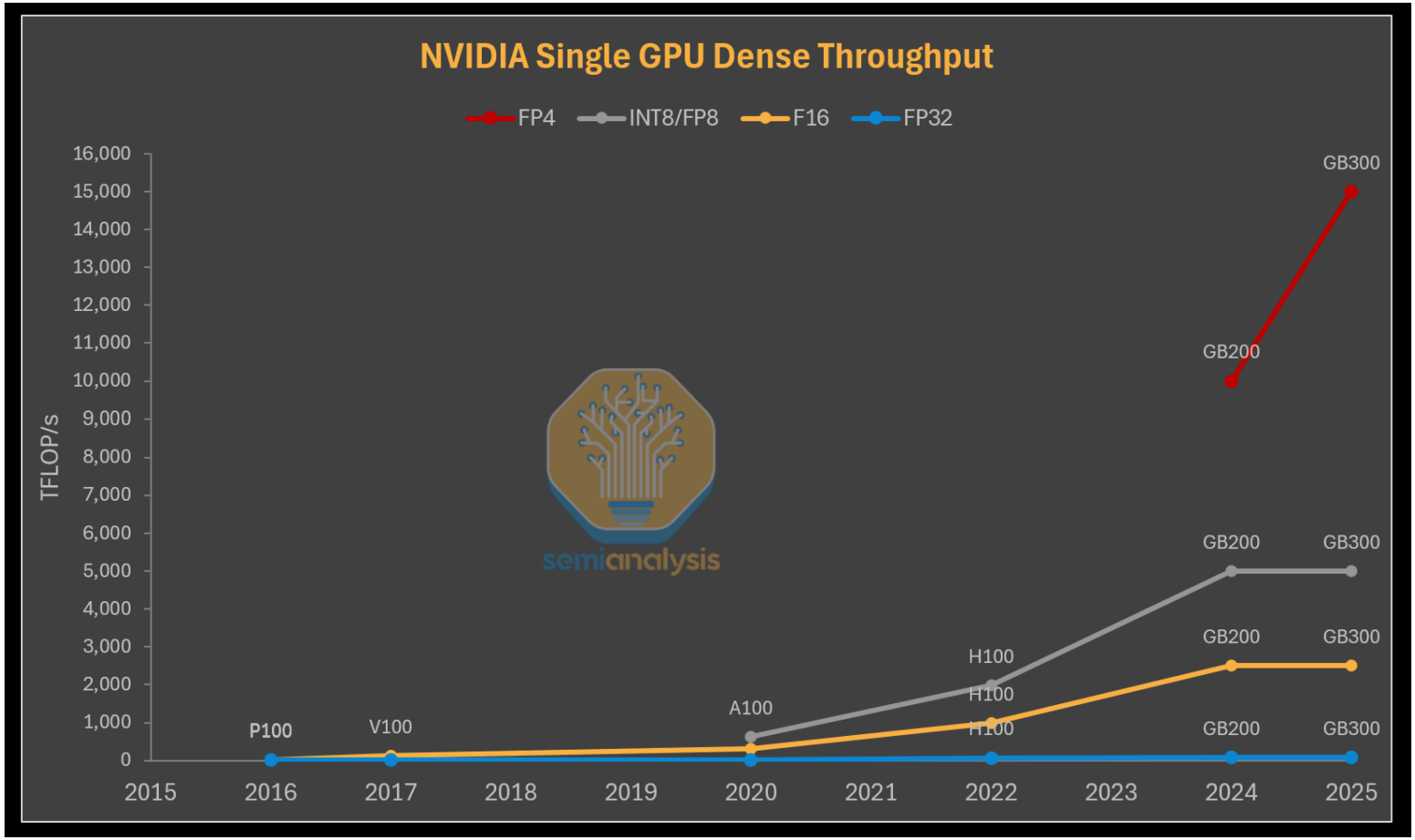
Nvidia GPU
Tensor Cores

Blackwell gains w.r.t. K20X

- Number Representation (~32x)
  FP32, FP16, Int8, FP4
- Complex Instructions (~12.5×)
  DP4, HMMA, IMMA
- Technology node (~3x)
  28m, 16nm, 7nm, 5nm, 4nm
- Die Size 2x

NUS National University of Singapore

Source: Nvidia; Bill Dally


NVIDIA Single GPU Dense Throughput
FP4
INT8/FP8
F16
FP32
TFLOP/s
16,000
15,000
14,000
13,000
12,000
11,000
10,000
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
P100
V100
A100
H100
H100
H100
GB200
GB300
GB200
GB300
GB200
GB300
GB200
GB300
GB200
GB300
semianalysis

The Hardware Lottery

Sara Hooker

Google Brain
shooker@google.com

Transformer Architecture is ideally matched with GPUs

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly. This historical treatment is odd given that hardware and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and *not* because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which makes it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are likely to become even more uneven, with certain research directions moving into the fast-lane while progress on others is further obstructed.

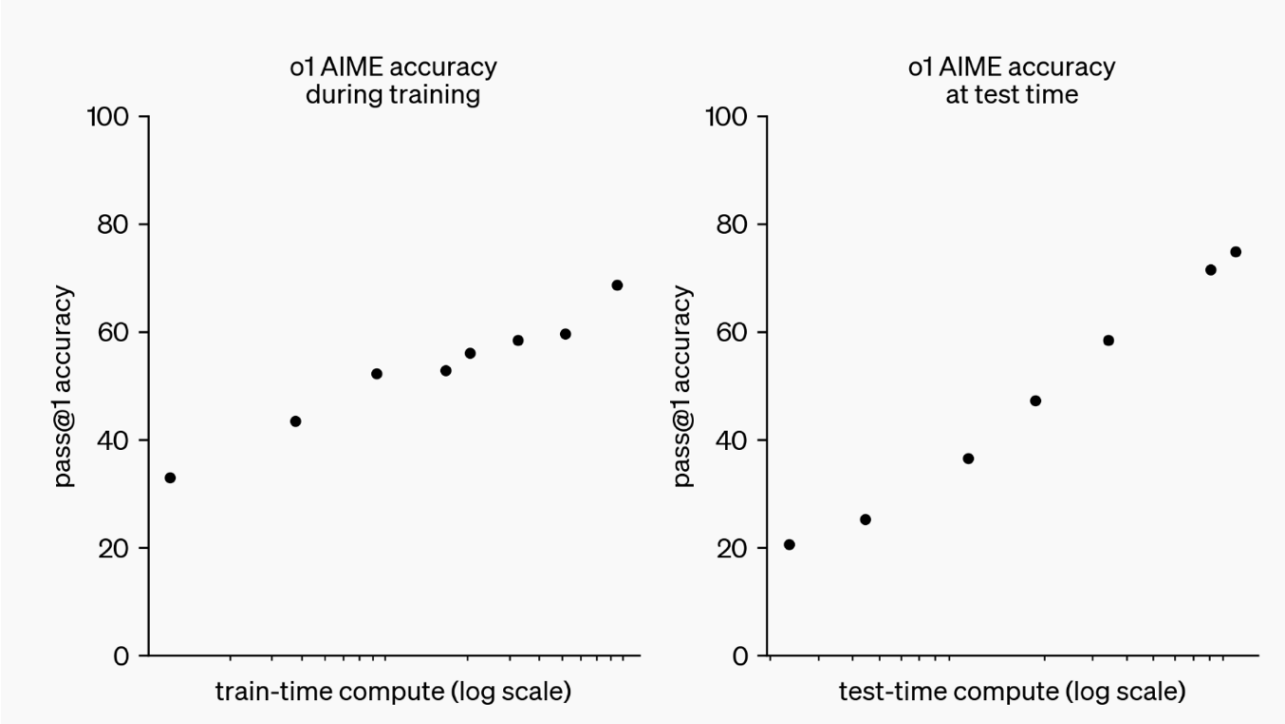
GPU dominance among AI data center accelerators

- In AI data center chips, GPUs dominate by revenue (>80%)
- In broader data-center accelerator market (all workloads, not just AI), GPU's share is less but still significant
- AI is the dominant workload today and likely in the future
- Reconfigurable future of accelerators seems wishful thinking
- But….

NUS National University of Singapore

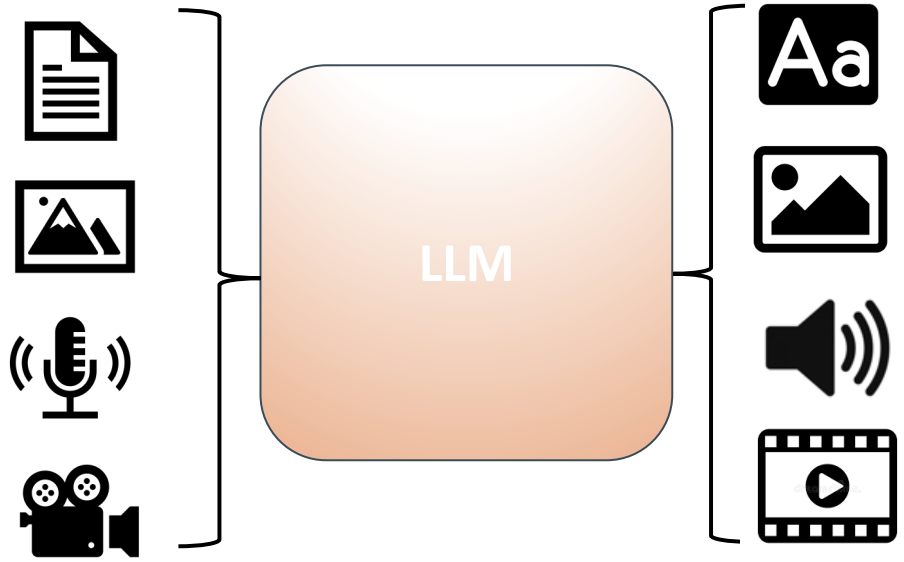
Inference has become the key workload
Test-time compute scaling: "Reasoning" LLM

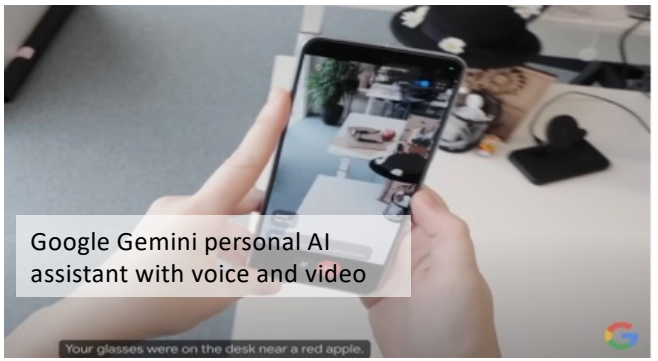
Test-time compute allows the model to engage in extensive chain-of-thought reasoning during inference, enhancing accuracy



Source: OpenAI

Emergence of Native Multimodal LLMs
AI inference is moving to the edge

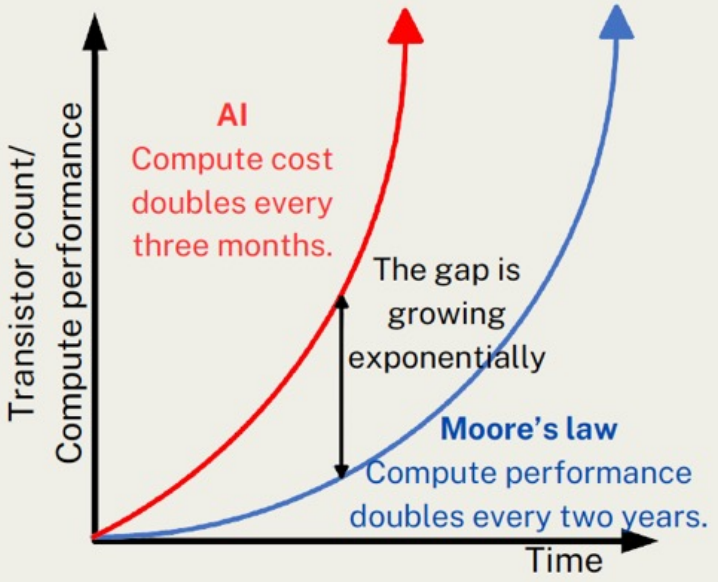




- Seamless interaction with physical world via integration of multimodal data
- Perfect foundation for embodied intelligence

AI Compute Challenge

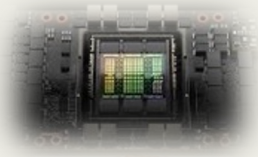
AI compute gap



Short battery life

Power-hungry applications drain battery life very quickly.

AI power gap



GPU
700W

Brain
20W

Watch
1W

Need faster, more energy-efficient, and cheaper AI inference accelerators

Era of AI Inference Accelerators

DOI:10.1145/3361682

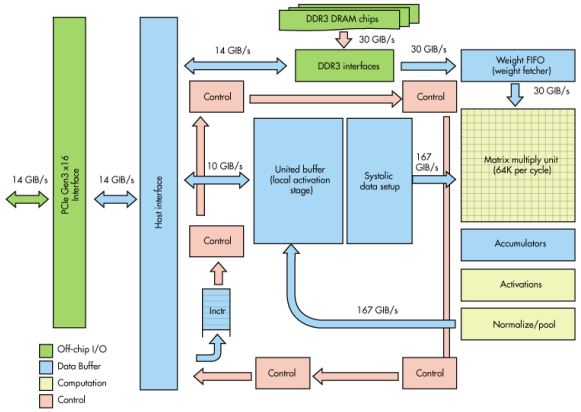
DSAs gain efficiency from specialization and performance from parallelism.

BY WILLIAM J. DALLY, YATISH TURAKHIA, AND SONG HAN

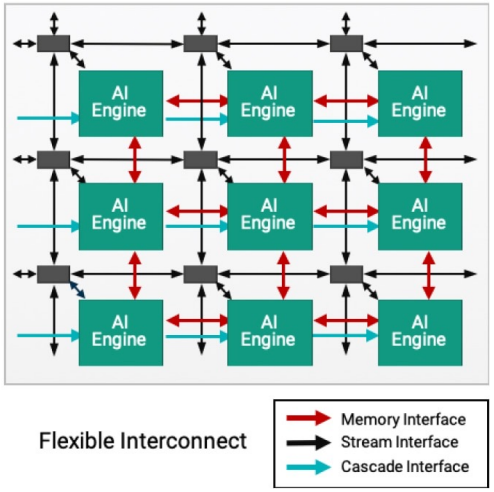
Domain-Specific Hardware Accelerators

Specialization:

- Designed for a domain
- High performance due to massive parallelism
- Low power due to design simplicity
- Low programmability



Google Tensor Processing Unit (TPU) AI accelerator



AMD AI Engine

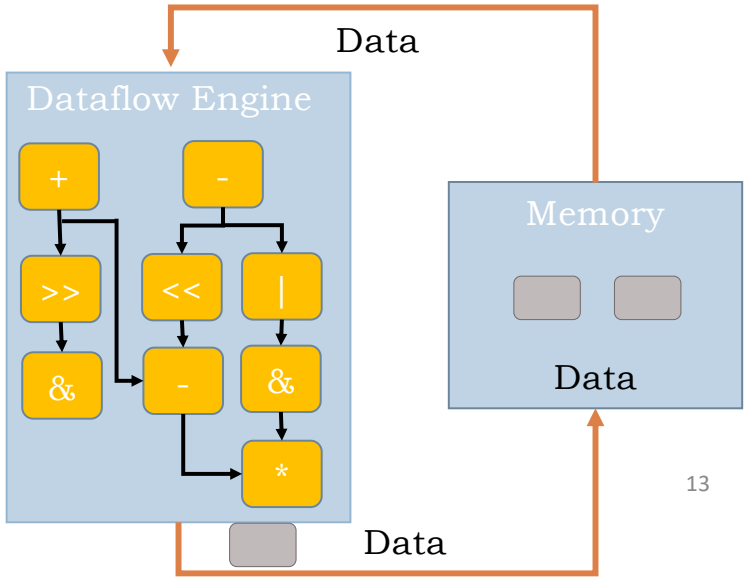
Dataflow Processor

A Preliminary Architecture for a Basic Data-Flow Processor*

Jack B. Dennis and David P. Misunas
Project MAC
Massachusetts Institute of Technology

ISCA 1975

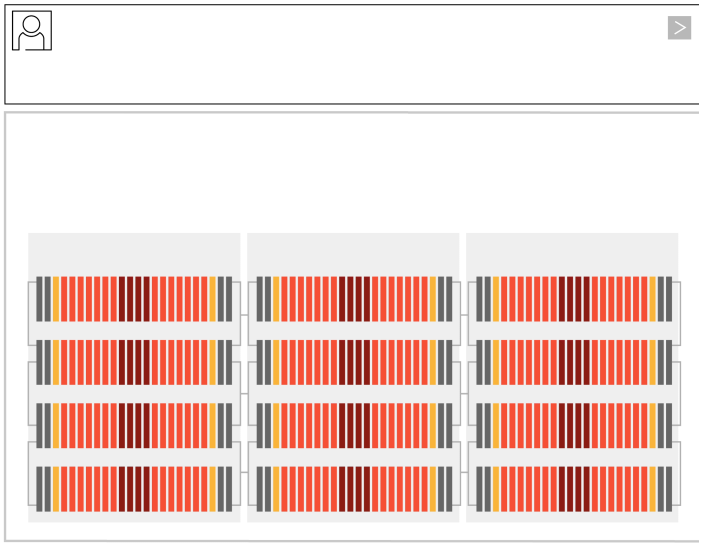
- Expose instruction-level parallelism in dataflow graph at compile time
- Dataflow graph is directly mapped onto hardware
- Operation is triggered when input data is ready



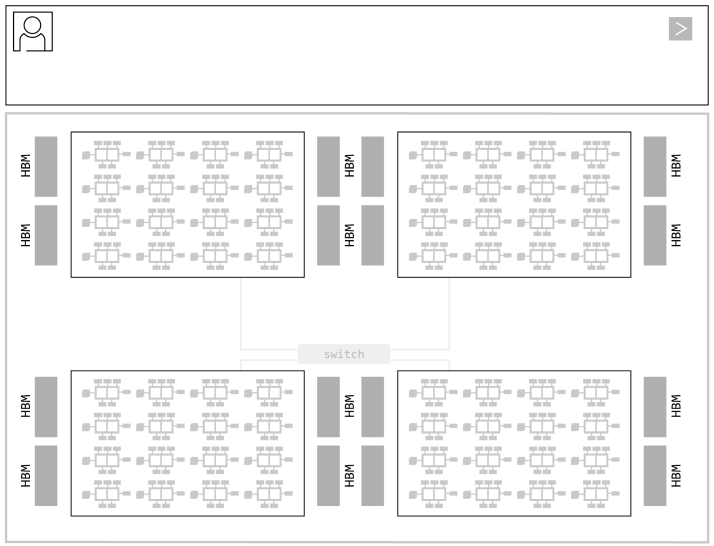
13

Groq Language Processing Unit (LPU)

- Fully software controlled
- Programmable Assembly Line Architecture
- Deterministic Compute and Networking
- Limited on-chip Memory (230MB)

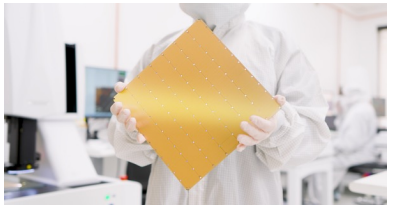


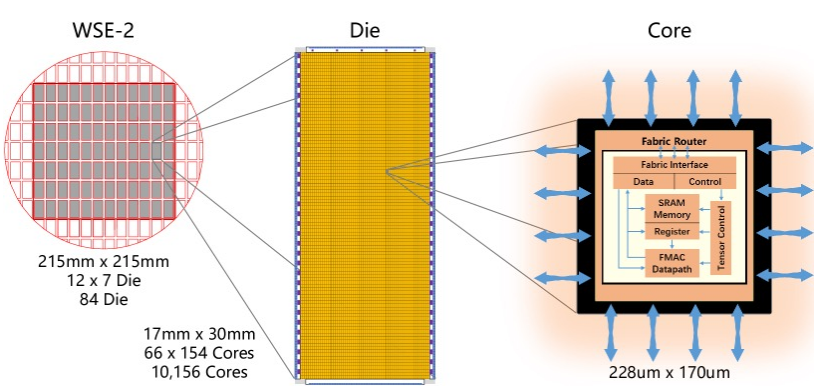
Groq LPU LLM Execution

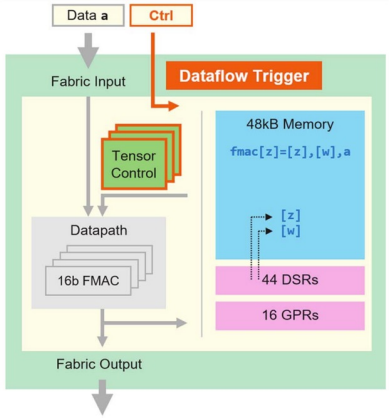


GPU LLM Execution

Cerebras Wafer Scale Engine 3 (WSE-3): Massive on-chip SRAM (44GB)

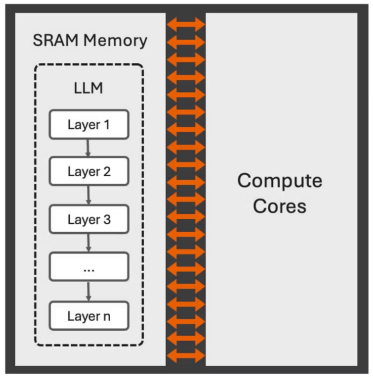






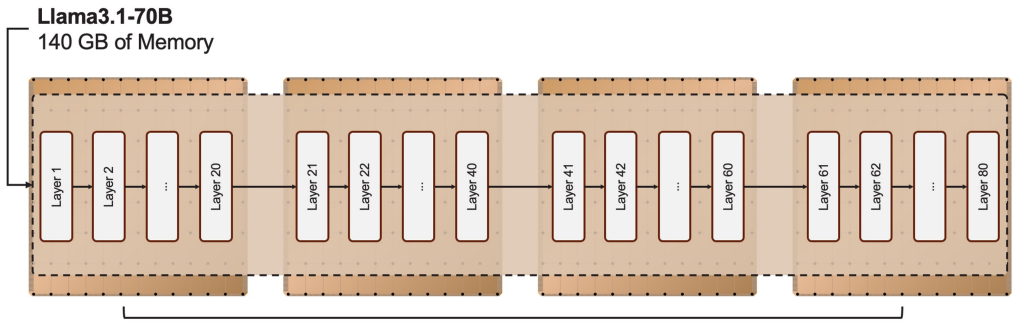
Transistors	4 trillion
Square millimeters of silicon	46,225
AI cores	900,000
AI compute	125 petaflops
On chip memory	44 gigabytes
Memory bandwidth	21 petabytes
Network fabric bandwidth	214 petabits

Cerebras Wafer Scale Engine 3



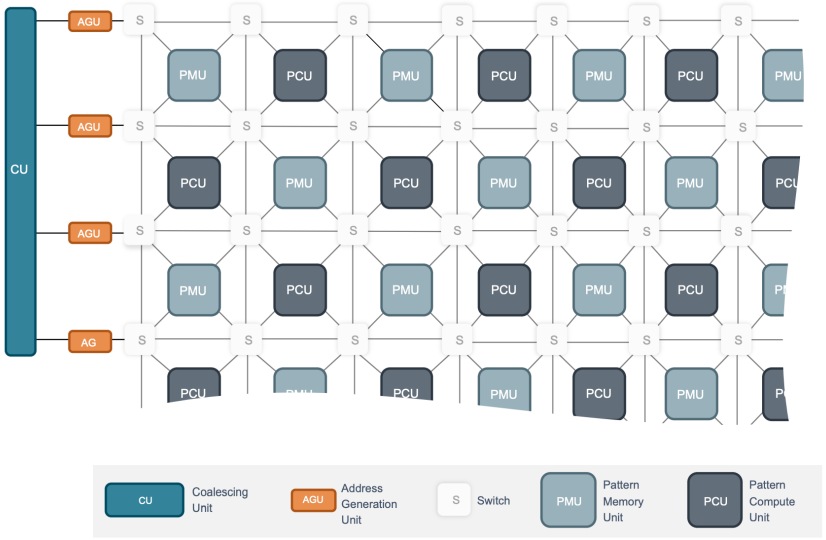
21 Petabytes/s

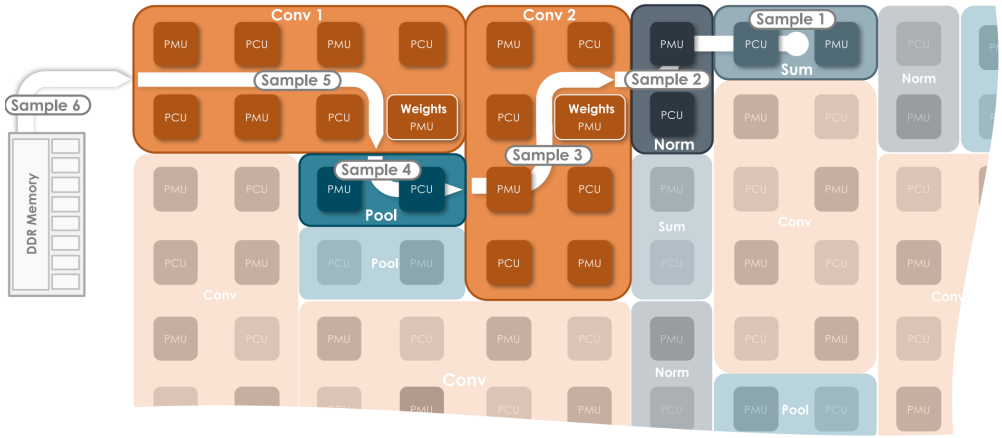
Large models easily scale across multiple Wafer Scale Engines



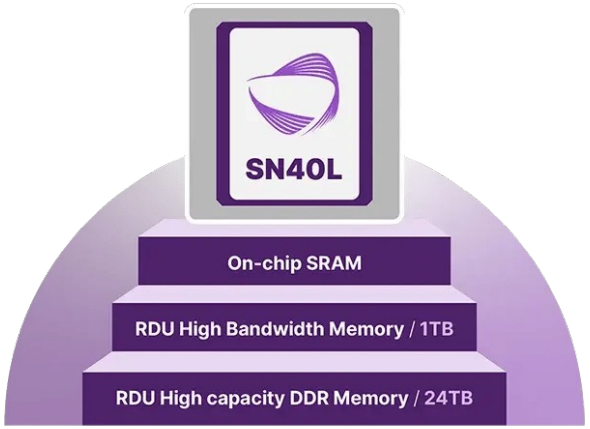
4x Wafer Scale Engines
176 GB of Memory

SambaNova Reconfigurable Dataflow Unit (RDU)

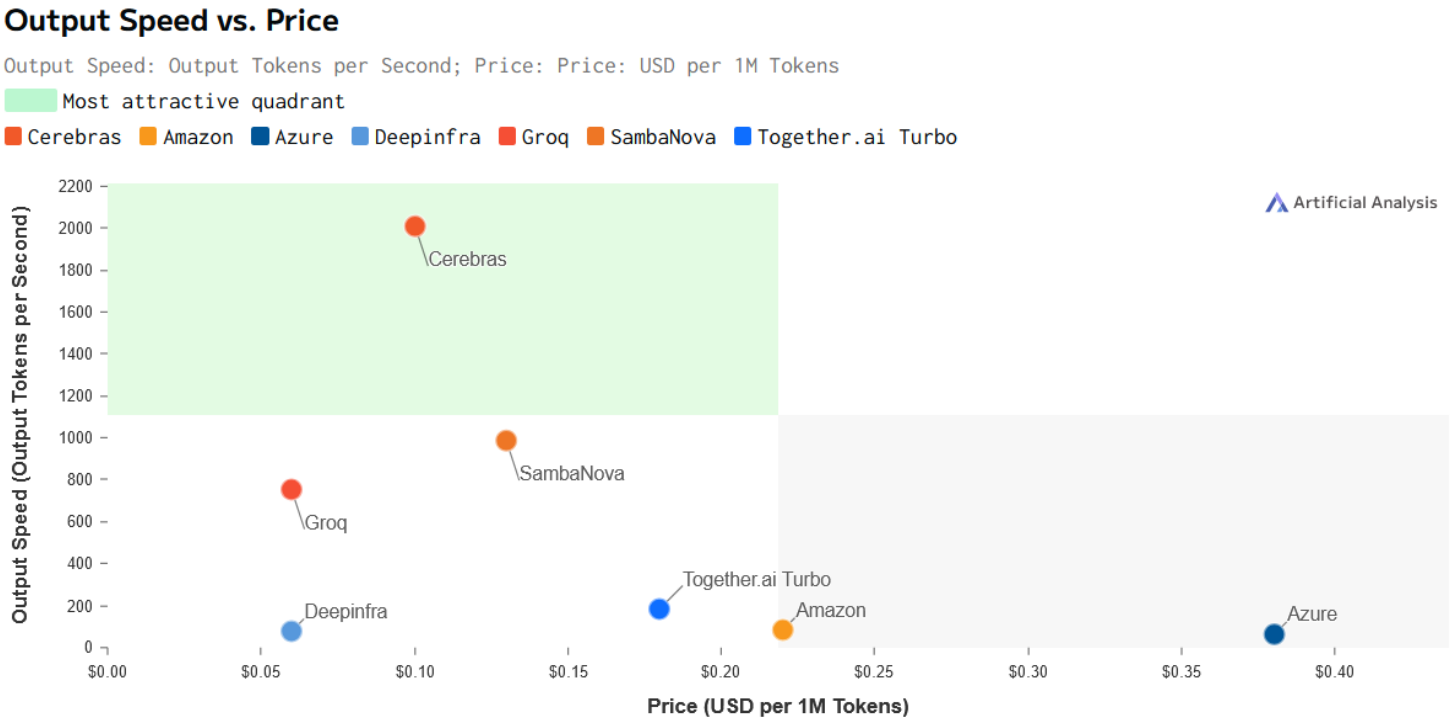




Links between tiles can be reconfigured on the fly to facilitate quick data movement



AI accelerators offer faster inference for smaller models



Price vs Speed for Llama-3.1–70B. source: Artificial Analysis

Domain-Specific AI Accelerators Tradeoffs

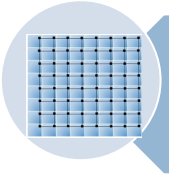
- Application characteristics to benefit from specialization:
  - Parallelism: Substantial numbers of calculations can be parallelized
  - Regularity: Computations are stable and arrive at regular intervals
  - Locality: Relatively few memory accesses are needed per computation

- Drawbacks of specialization:
  - Can only run a limited range of applications
  - Designing and creating specialized hardware can be expensive
    - Specialized design cost must be amortized over number of units sold
  - AI models/optimizations that do not match the specialization suffer

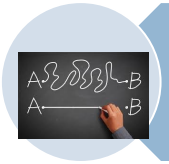
Dream Reconfigurable Accelerator: General yet Special

- Software (compiler) instantiates the dataflow corresponding to different applications onto the same hardware
- Should appear as easily programmable to software engineers
- Performance, energy-efficiency closer to domain-specific accelerators

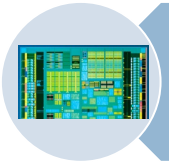
CGRA: A promising realization of the dream reconfigurable accelerator



Embrace Parallelism



Keep Hardware Simple

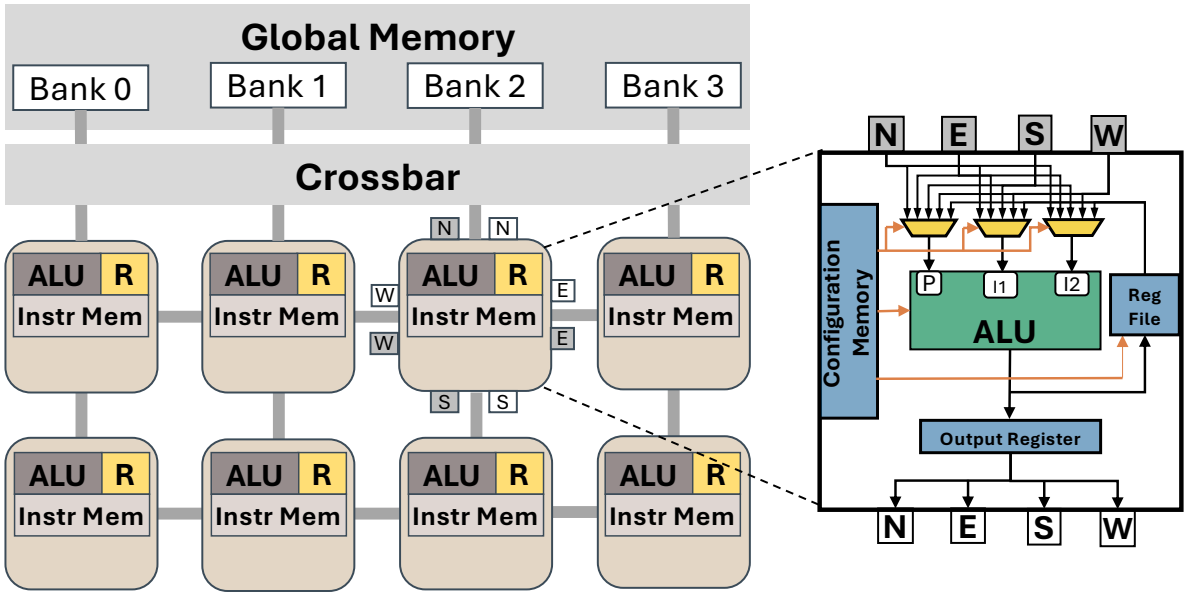


Expose to Software

NUS National University of Singapore

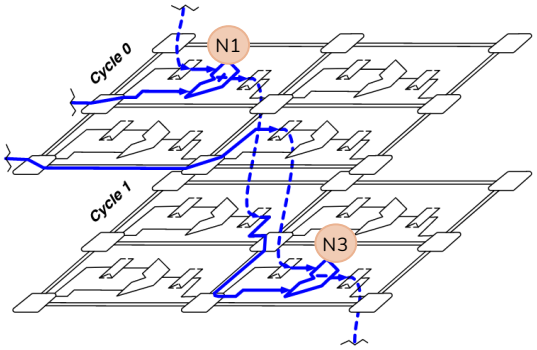
CGRA: Efficiency with Programmability

- Parallelism: Array of simple processing elements (PE)
- PE operations and routing can be configured per cycle
- Exposed to software: Compiler sets the configuration according to dataflow
- Simple hardware and hence low power: During execution, PE array and switches simply follow the pre-defined configurations

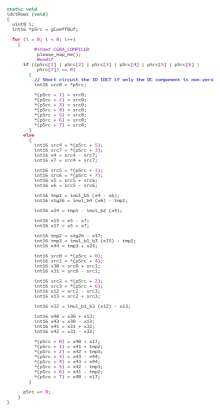


Dataflow synthesis on CGRA

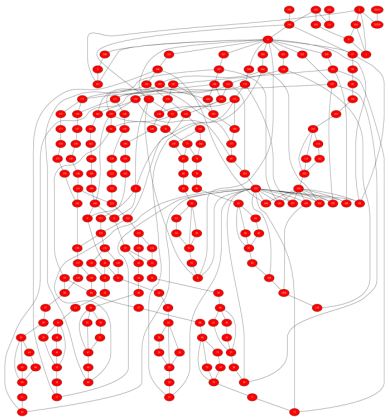
- Target: An application-level loop kernel
- Mapping the dataflow graph (DFG) of loop body on to CGRA
  - Placement: assigning DFG operations to ALUs
  - Routing: mapping data signals using wires and registers



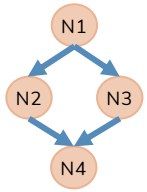
Placement and routing



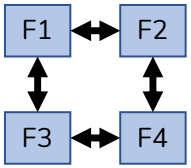
Loop kernel c code



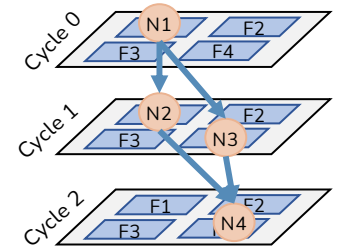
Dataflow graph (DFG) of the loop body



DFG

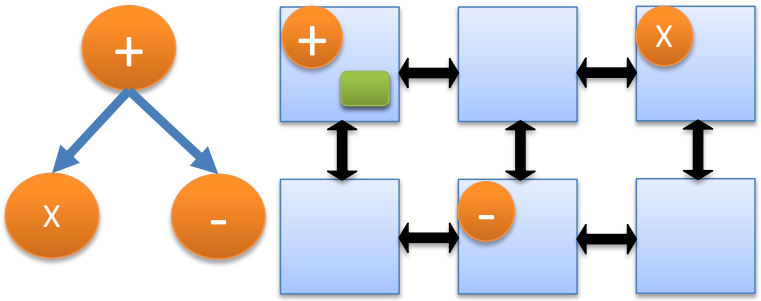


2x2 CGRA

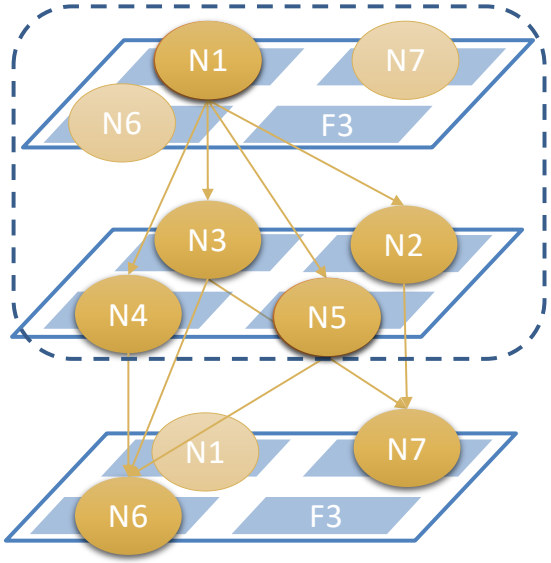


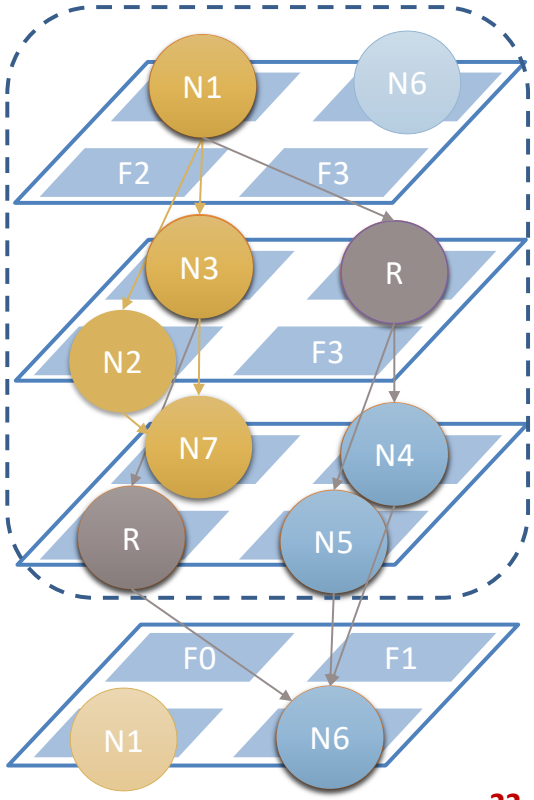
Spatio-temporal mapping

HyCUBE CGRA @ NUS



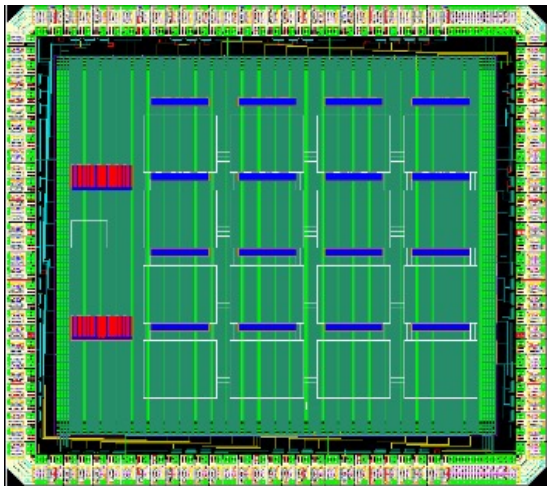
- Reconfigurable single-cycle multi-hop interconnect
- Creates large dynamic neighborhood reachable within single-cycle through asynchronous bypassing of intermediate PEs

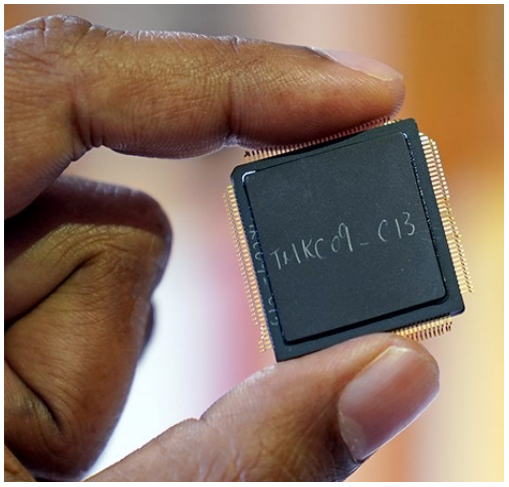


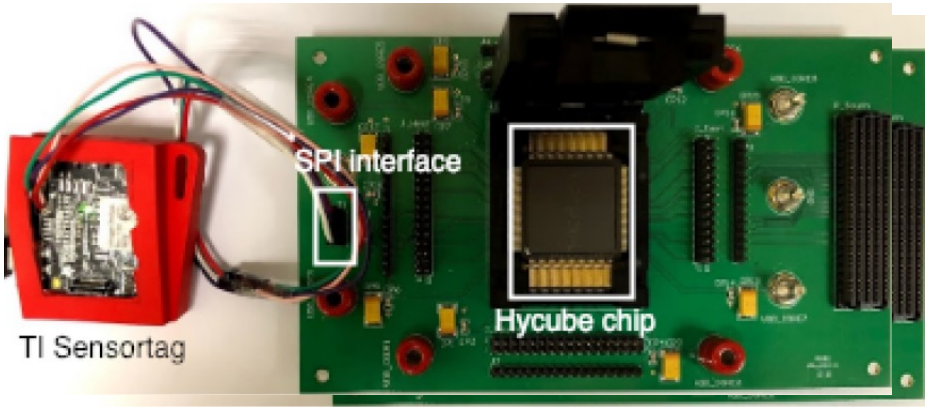


HyCUBE: a 0.9V 26.4 MOPS/mW, 290 pJ/cycle, Power Efficient Accelerator for IoT Applications ASSCC'19
HyCUBE : A CGRA with Reconfigurable Single-cycle Multi-hop Interconnect DAC 2017

HyCUBE CGRA @ NUS 2019



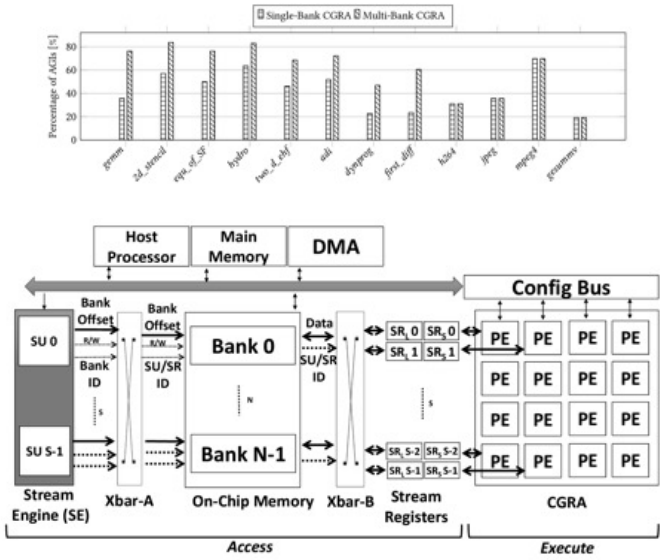


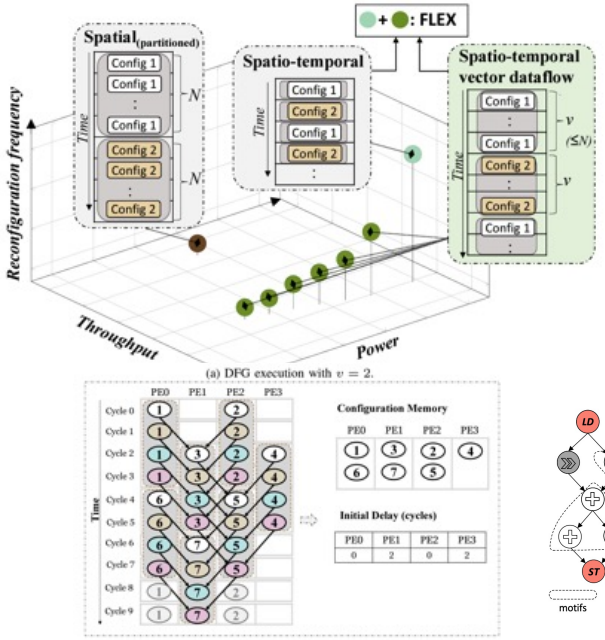


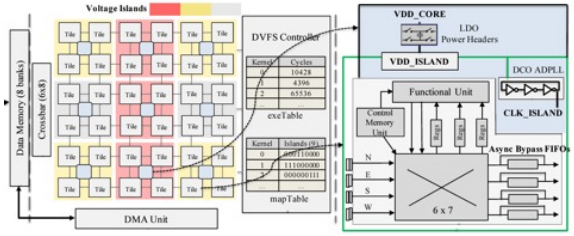
4x4 CGRA, TSMC 40nm

NUS HyCUBE: 90 MOPS/mW
Samsung SRP: 22 MOPS/mW
ARM CPU: 2.6 MOPS/mW
Xilinx FPGA: 25 MOPS/mW

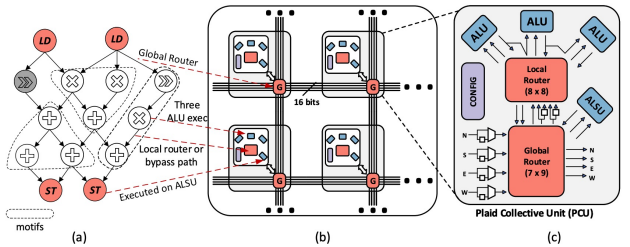
CGRA Architectural Innovations







ICED:
DVFS-aware CGRA
1.32x energy-efficiency over CGRA at same performance



CASCADE
Decoupled Access-Execute CGRA
3x speedup, 2.2x performance/watt over iso-area CGRA

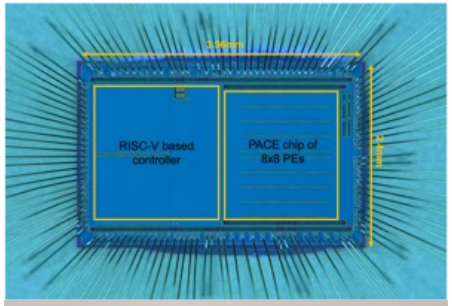
FLEX:
Spatio-temporal vector dataflow
45% less energy. 1.9× power efficiency at same performance over CGRA

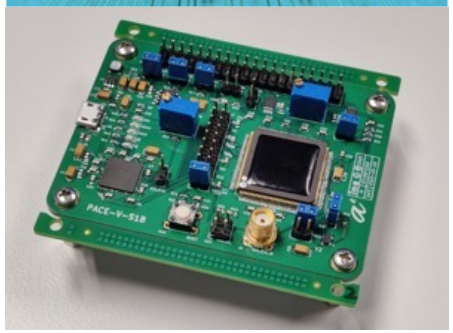
Plaid:
Reduced routing provision
43% less power
46% less area

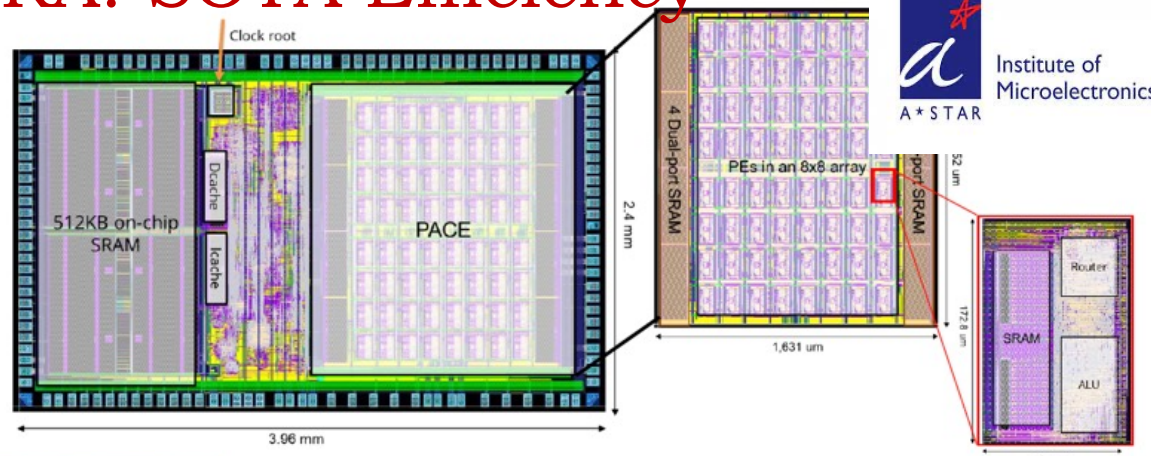
CASCADE: High Throughput Data Streaming via Decoupled Access/Execute CGRA. CASES 2019
FLEX : Introducing FLEXible Execution on CGRA with Spatio-Temporal Vector Dataflow. ICCAD 2023
ICED: An Integrated CGRA Framework Enabling DVFS-Aware Acceleration. MICRO 2024
Plaid: Enhancing CGRA Efficiency Through Aligned Compute and Communication Provisioning. ASPLOS 2025

NUS-IME PACE CGRA: SOTA Efficiency

8x8 CGRA: 1.1 mW at 10MHz
582 GOPS/W at 0.45V, 40nm ULP
Estimated 1 TOPS/W at 22nm

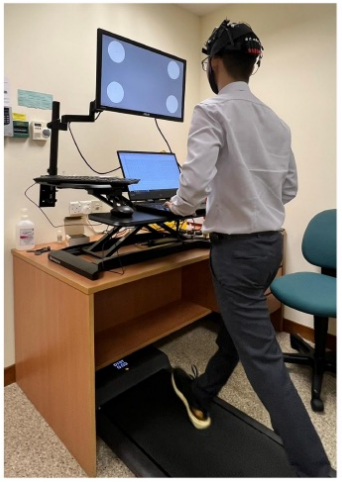


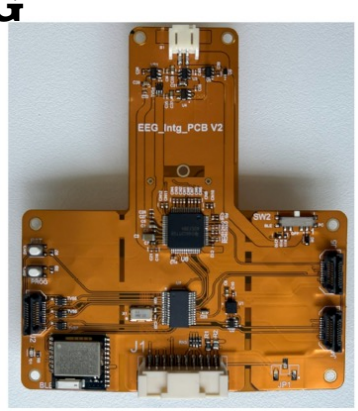




Institute of Microelectronics
A*STAR

Wearable EEG





NUS National University of Singapore

HOT CHIPS

CGRA Comparison

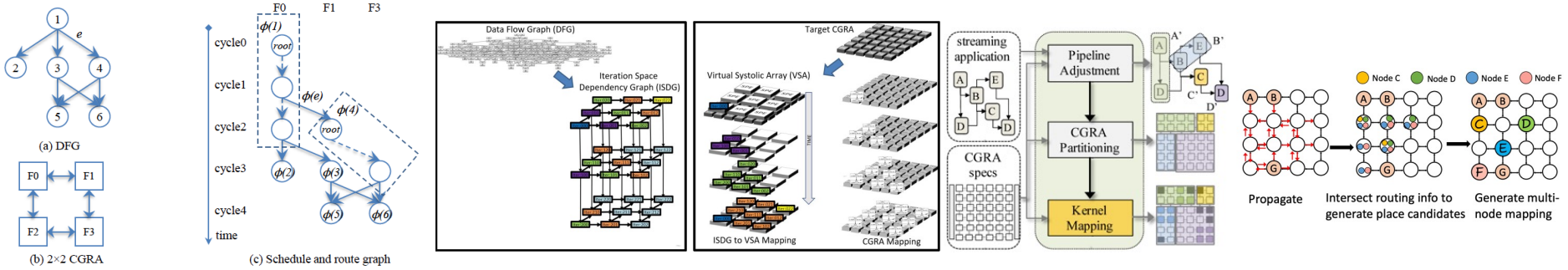
PACE has highest power efficiency even at 40nm

	Academic						Commercial		
	Post synthesis simulation/Post P&R				Silicon		Post synthesis simulation/Post P&R	Silicon	
CGRA	**UE_CGRA** HPCA'21	**SNAFU** ISCA'21	**TRANSPIRE** DATE'20	**RIPTIDE** MICRO'22	**AMBER** VLSI'22	**PACE**	**ULP-SRP** FPT'12	**Renesas DRP** VLSI'18	**Cardinal SN10 (Improved version of Plasticine for ML training)** ISSCC'22
PE array	8x8	6x6		6x6	384 PEs	**8x8**	3x3	96 PEs	640 PMU 640 PCU
Node	28nm TSMC	Intel 22FFL	28nm FDSOI	Intel 22FFL	16nm FINFET	**40nm UMC**	40nm TSMC	28 nm	7nm TSMC
Voltage (V)	-	-	0.6	-	1.29	**0.45**	1.1	-	-
Frequency (MHz)	750	50	50	50	955	**10**	100	333 MHz	-
Area (mm^2)	0.25[1]	0.27[1]	0.27	0.25[1]	20.1	**3.94**	-	-	-
Performance (MOPS)	625	71	-	62	367000	**640**	467[2]	960 gigaflops	2.6 petaflops
Power (mW)	14	0.54	-	0.24	682	**1.1**	11.3	-	-
Power efficiency (GOPS/W)	45	134	224	254	538	**582**	23	143[3]	-

[1] Only the PE array included
[2] Estimated from published 467 MIPS
[3]Estimated value from experiments

Dataflow Synthesis on CGRA

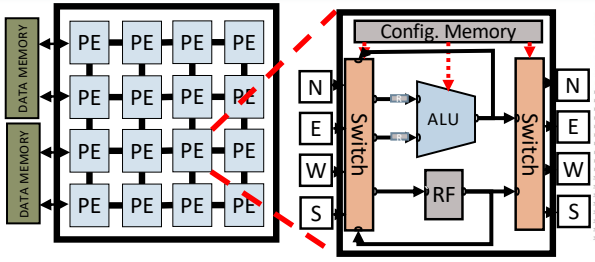
- A fertile ground for research
- Graph Minor (2012): Theoretically optimal based on graph minor modeling
- ChordMap (2020): Streaming dataflow application synthesis
- HiMap (2021), Panorama (2022): Scalable mapping through hierarchical abstractions
- LISA (2022): Automated compiler synthesis using GNN

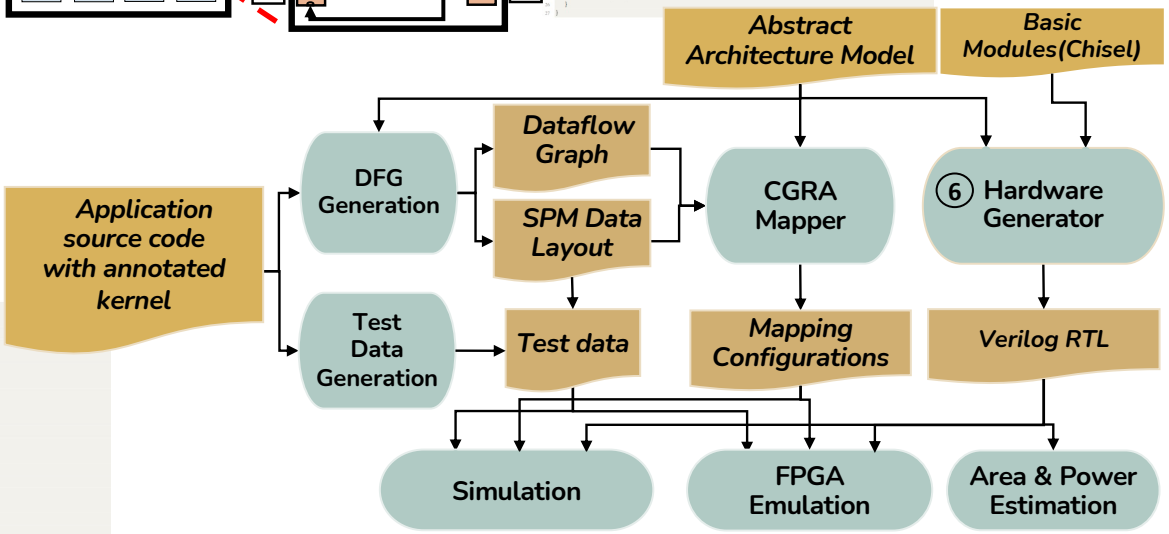


Graph Minor Approach for Application Mapping on CGRAs. **Best paper award** FPT 2012 TRETS 2014
ChordMap: Automated Mapping of Streaming Applications onto CGRA. TCAD 2021
HiMap: Fast and Scalable High-Quality Mapping on CGRA via Hierarchical Abstraction. DATE 2021 TCAD 2022
Panorama: Divide-and-Conquer Approach for Mapping Complex Loop Kernels on CGRA. DAC 2022
LISA: Graph Neural Network based Portable Mapping on Spatial Accelerators. HPCA 2022
Rewire: Advancing CGRA Mapping Through a Consolidated Routing Paradigm. DAC 2025

NUS Morpher Open-Source CGRA Toolchain

Our position paper at ICCAD 2025:
Building an Open CGRA Ecosystem for Agile Innovation



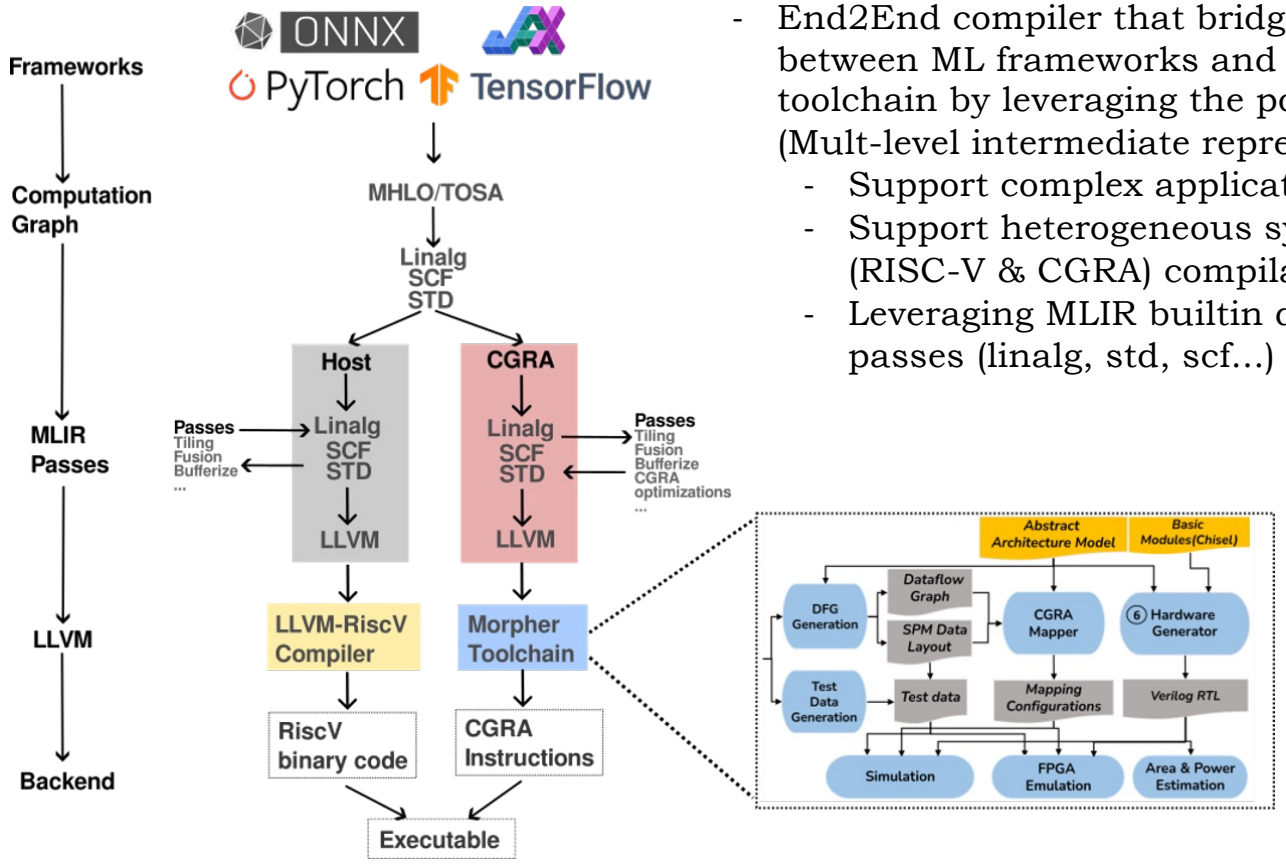



```c
#define SIZE  20
int A[SIZE], B[SIZE], C[SIZE];

__attribute__((noinline))
void array_add(){
    for (int i=0;i<SIZE; i++){
        #ifdef CGRA_COMPILER
        please_map_me();
        #endif
        C[i] = A[i]+B[i];
    }
}

int main(){
    for (int i=0;i<SIZE; i++){
        A[i] = i * 2 + 5;
        B[i] = i * 3;
        C[i] = 0;
    }
    array_add();
    for (int i=0;i<SIZE; i++) printf("%d\n", C[i]);
    return 0;
}
```

SCAN ME

https://github.com/ecolab-nus/morpher

NUS MLIR-Based End-to-End Compiler



- End2End compiler that bridges the gap between ML frameworks and the Morpher toolchain by leveraging the power of MLIR (Mult-level intermediate representation).
  - Support complex applications
  - Support heterogeneous system (RISC-V & CGRA) compilation
  - Leveraging MLIR builtin dialects and passes (linalg, std, scf…)

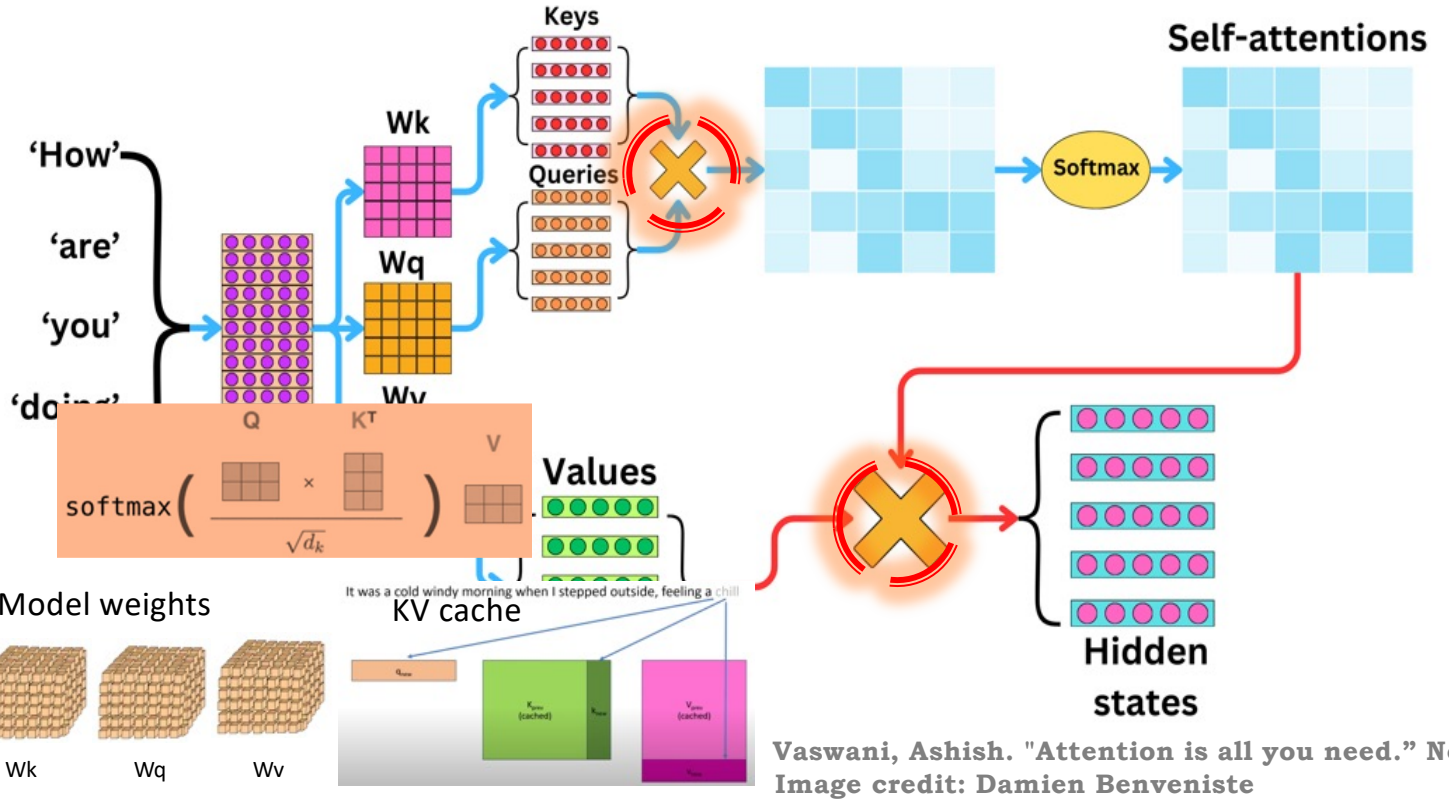
What happens when workload is not dense and regular anymore?

Programmability is solved;
Bottleneck moves to data

NUS National University of Singapore

Attention is all you need

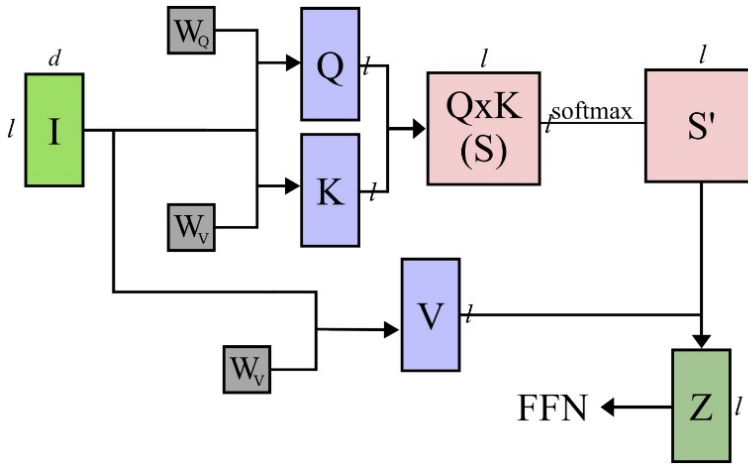
Attention is the cornerstone of LLMs enabling them to intelligently focus on the most relevant parts of any input

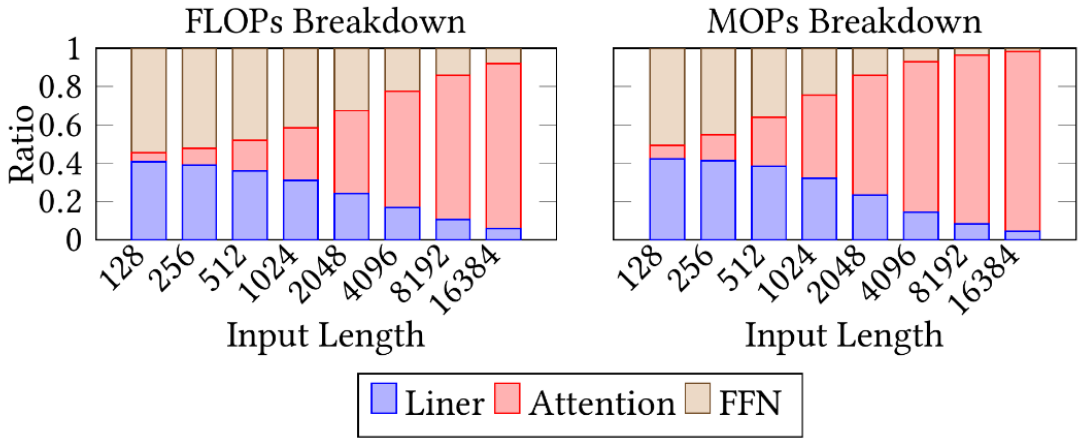


Vaswani, Ashish. "Attention is all you need." NeurIPS 2017
Image credit: Damien Benveniste

Self-Attention: Poor Scalability with Context Length

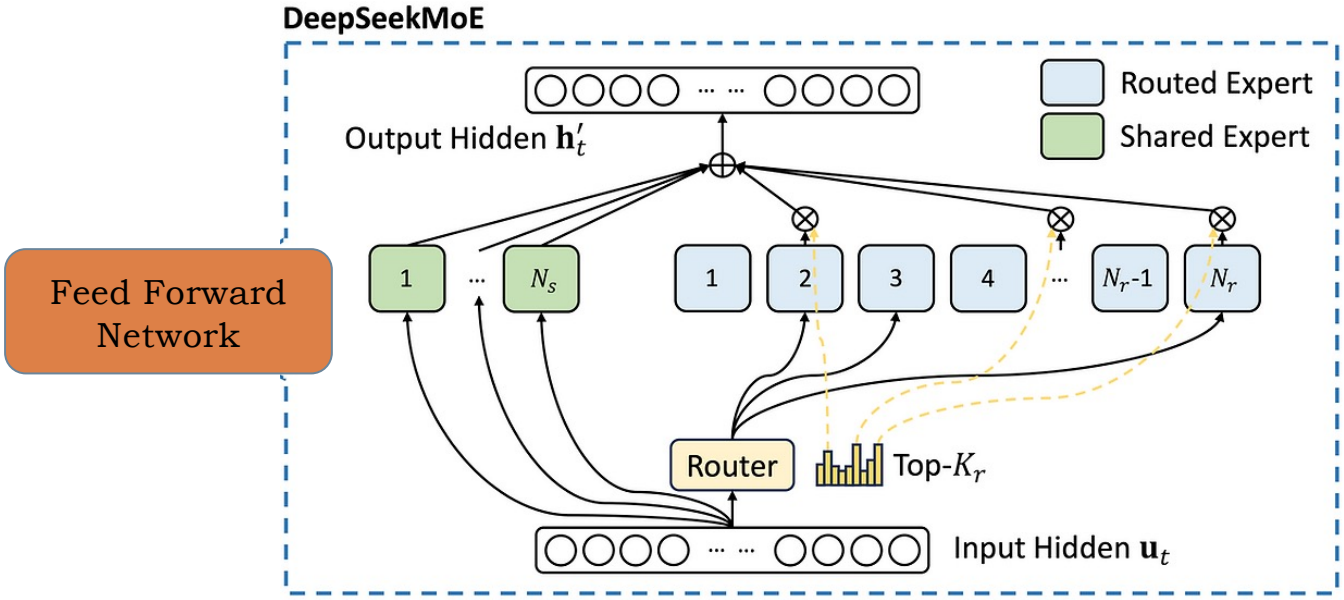
Attention scales poorly with context length, requiring quadratic operations and memory for n tokens





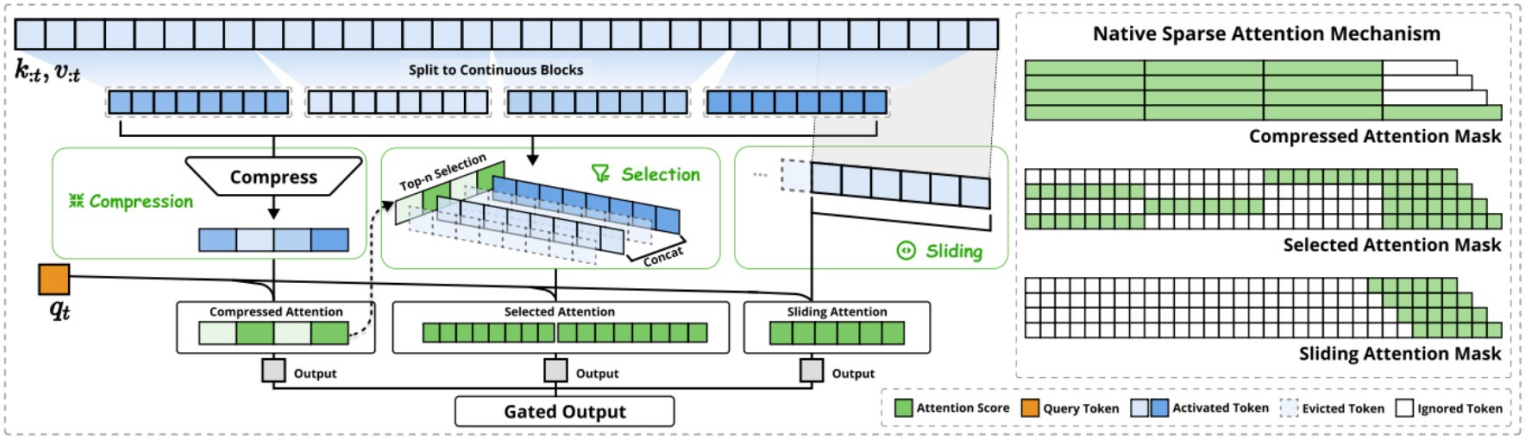
Sparsity in LLM Workloads

- **Mixture of Experts (MoE):** Only a subset of expert modules are activated per token: ~5.5% active parameters in Deepseek-v3 671B



Sparsity in LLM Workloads

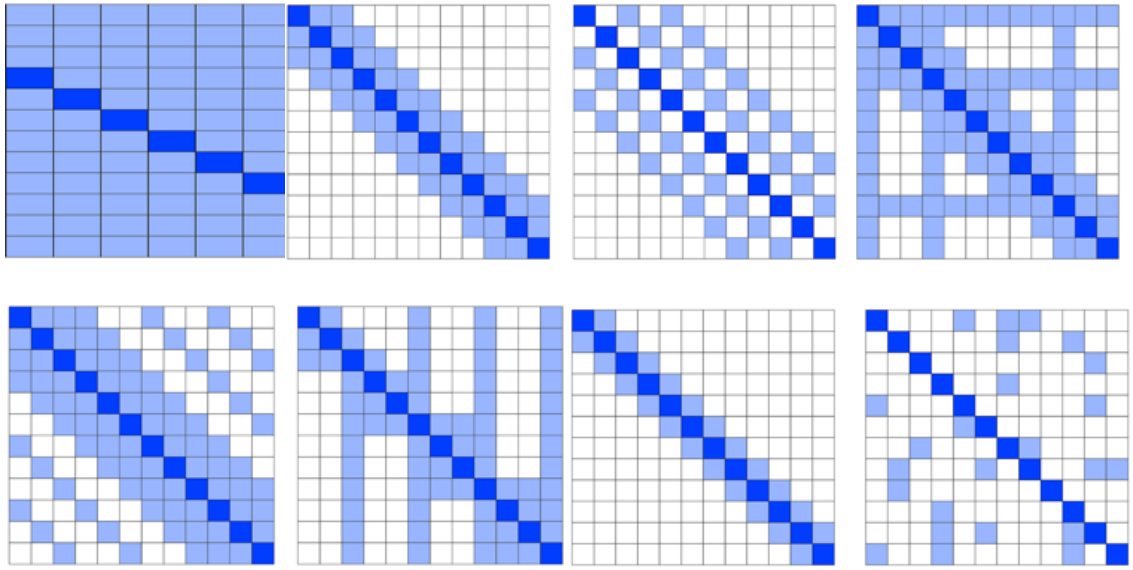
- **Mixture of Experts (MoE):** Only a subset of expert modules are activated per token: ~5.5% active parameters in Deepseek-v3 671B
- **Structured sparsity in attention:** Sparse attention patterns (e.g., Mistral's sliding window, DeepSeek native sparse attention)



Deepseek Native Sparsity

Structured Sparsity in Attentions

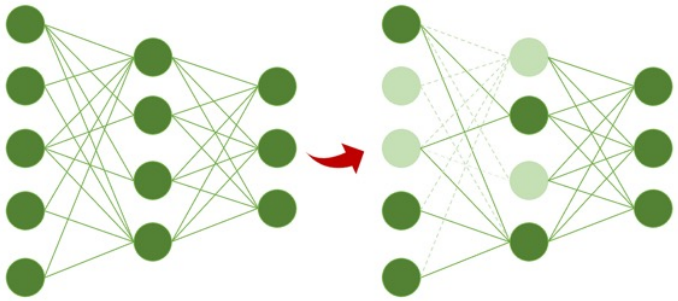
- Advanced attention mechanisms leverage diverse **sparse matrix computations** with specialized patterns for enhanced efficiency
- Each type of attention mechanism requires **unique dataflow**

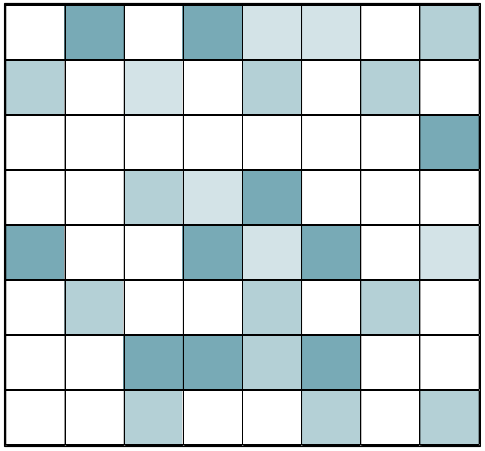


NUS National University of Singapore

Sparsity in LLM Workloads

- **Mixture of Experts (MoE):** Only a subset of expert modules are activated per token: ~5.5% active parameters in Deepseek-v3 671B
- **Structured sparsity in attention:** Sparse attention patterns (e.g., Mistral's sliding window, DeepSeek native sparse attention)
- **Unstructured Sparsity:** Individual parameters becoming zero via pruning, activation, or training-time sparsity constraints

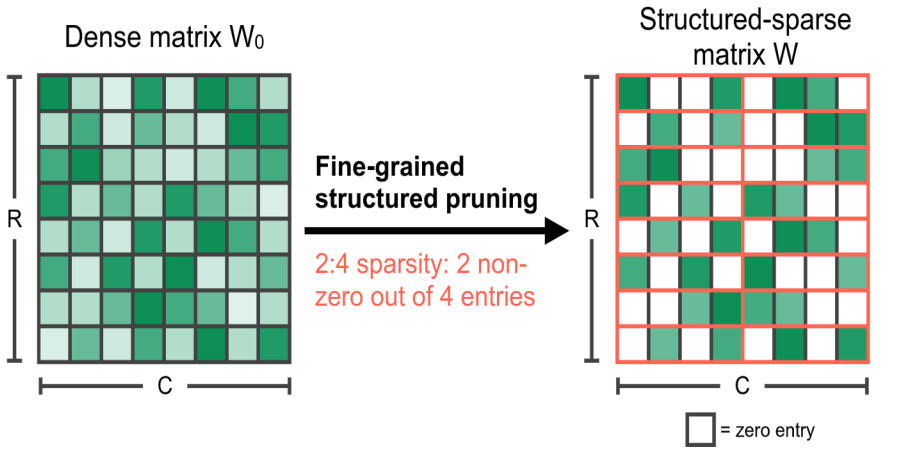




We need sparsity for efficiency…
But is the hardware ready?

NUS National University of Singapore

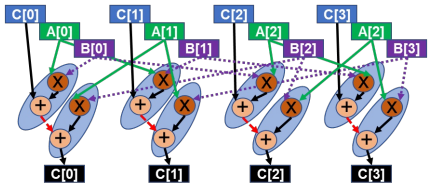
GPUs Adopted Restricted Structured Sparsity



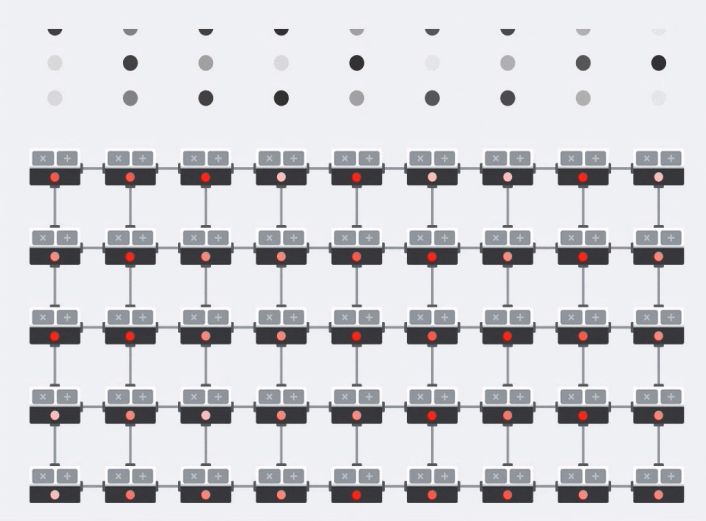
GPUs cannot deal with unstructured sparsity

Hardware-Algorithm Gap on TPU and CGRA

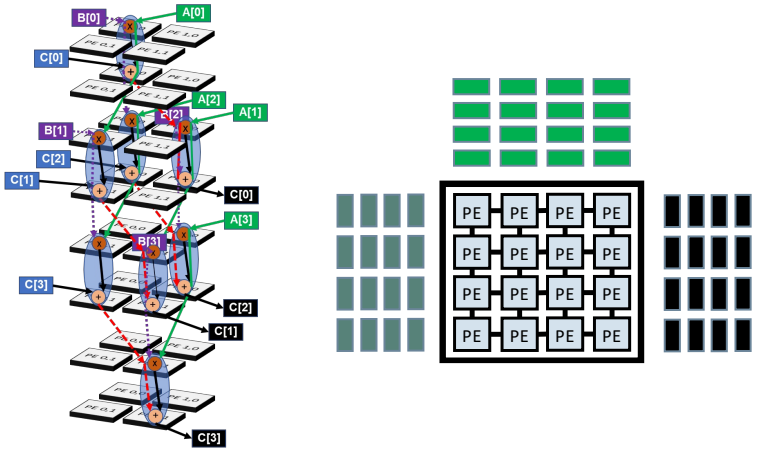
- Hardware needs structured dataflow, but AI creates unstructured sparsity
- Scattered non-zeros cannot match fixed hardware dataflow



GEMM DFG

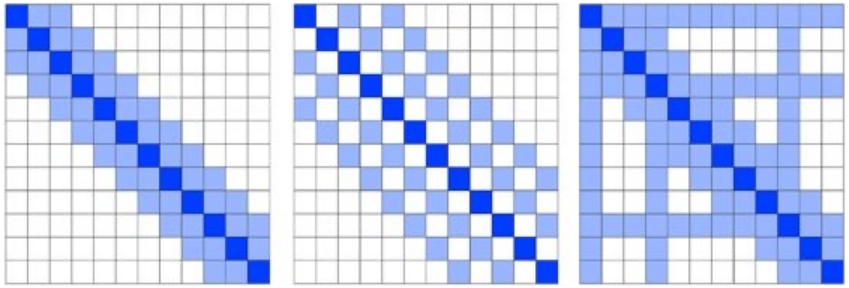


Google TPU Matrix Multiply Unit (MXU) Dataflow



HiMap GEMM Schedule & Dataflow on CGRA

SWAT: Architecture following Algorithm



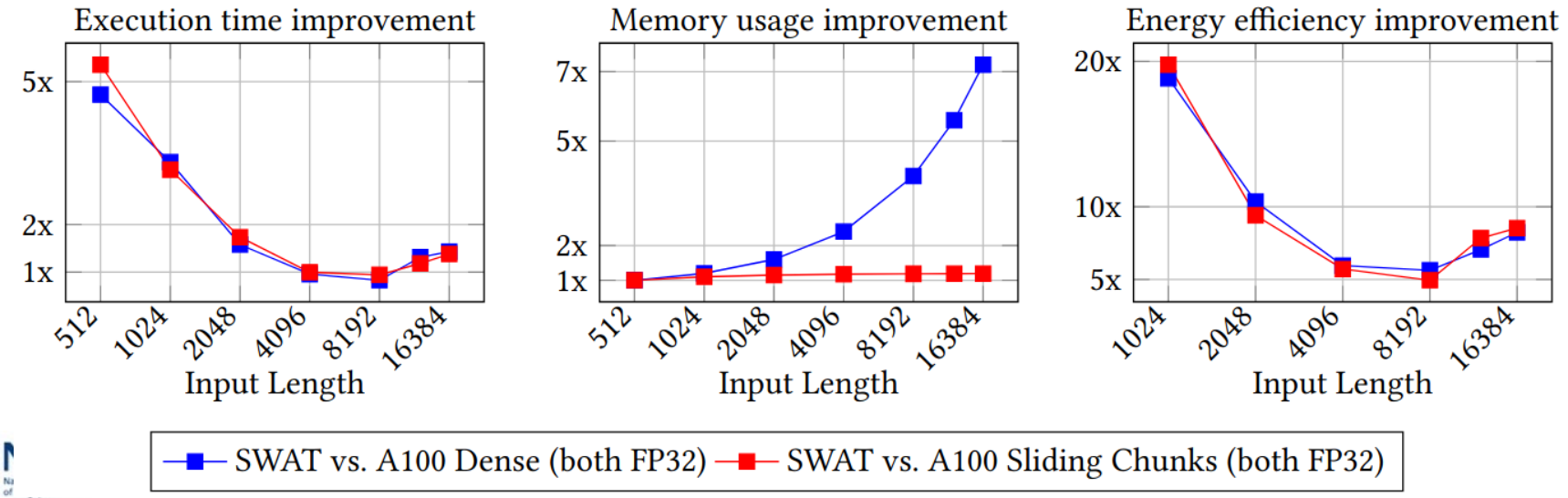
- Handcrafted dataflow engine on FPGA for sliding-window attention in LLM
- Each token attend to fixed number of predecessors/successor tokens
  - Reduces the complexity from quadratic to linear
- Used in many models: Mistral-7B, DeepSeek MLA, Swin (Vision)
- But….efficient implementation on hardware is hard
- Naïve GPU implementation is worse than dense due to masking overhead

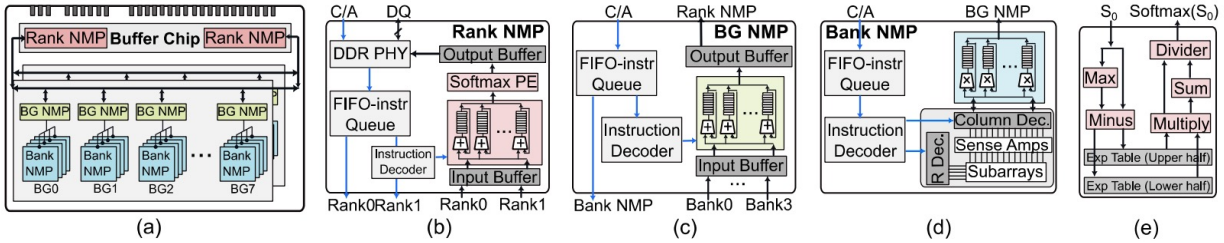
SWAT vs. GPU implementations

Single-batch:
Xilinx U55C (16nm) vs. Nvidia A100 (7nm)

SWAT FPGA Sliding-Window Attention achieves 15x energy-efficiency over GPU

- Improved execution time on worse technology node
- High energy efficiency gain





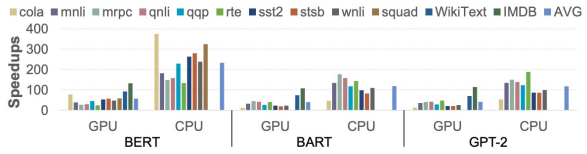
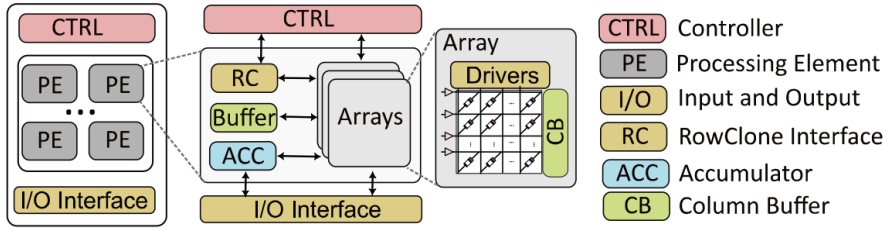


Fig. 8. The speedup improvement of SADIMM over GPU and CPU on BERT, BART, and GPT-2 benchmarks

SADIMM: Sparse Attention Accelerator with DIMM-based Near-memory Processing



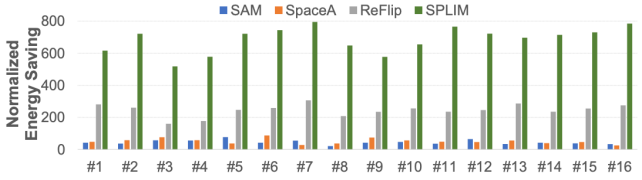
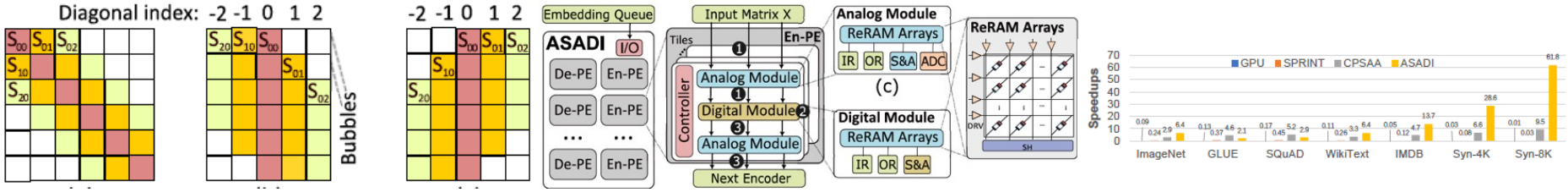


Fig. 15. Energy saving comparison between GPU baseline, SAM, SpaceA, ReFlip, and SPLIM (normalized to GPU)

SPLIM: Unstructured sparse matrix operations on structured in-situ computing

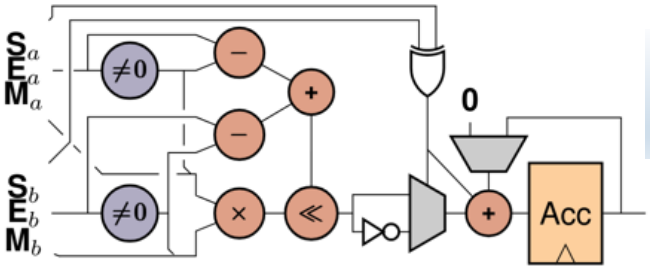


ASADI: In-memory computing for sparse attention

NUS National University of Singapore

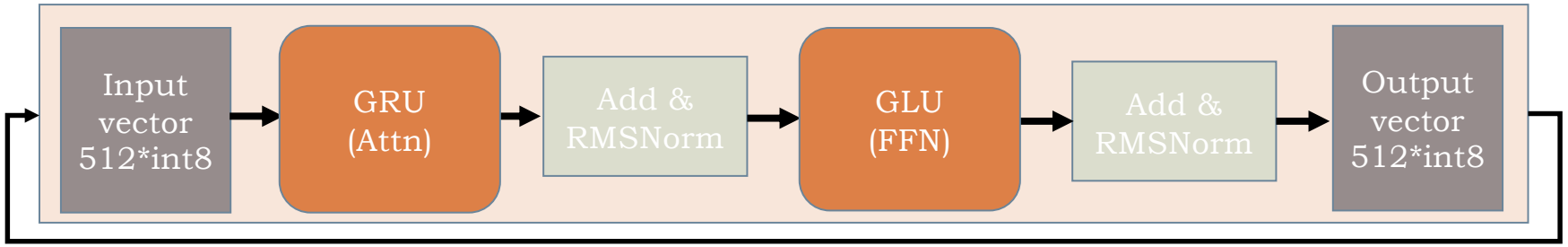
SADIMM: Accelerating Sparse Attention using DIMM-based Near-memory Processing. IEEE Transactions on Computers 2025
SPLIM: Bridging the Gap Between Unstructured SpGEMM and Structured In-situ Computing. IEEE TCAD 2025
ASADI: Accelerating Sparse Attention using Diagonal-based In-situ Computing. HPCA 2024

Quantized Attention: Reconfigurability Wins



(b) Minifloat MAC.

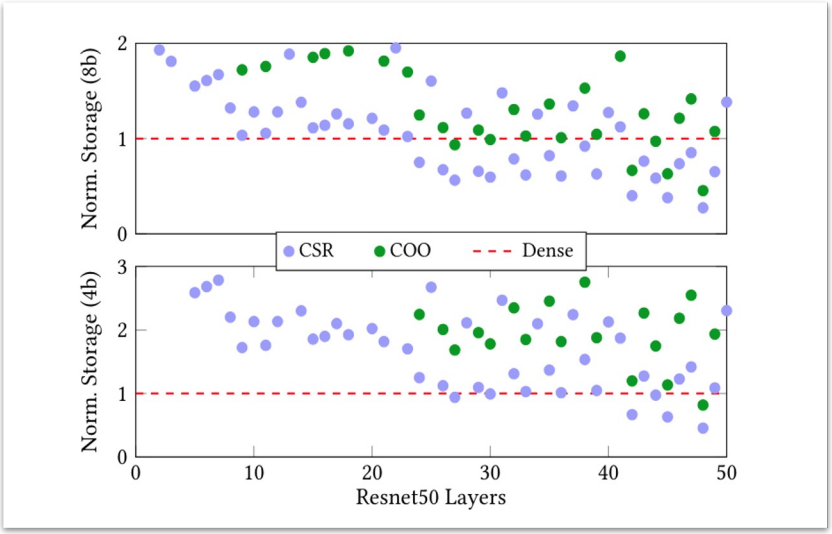
- 4-bit Mini-Floats provide comparable accuracy to 32-bit FP in transformers

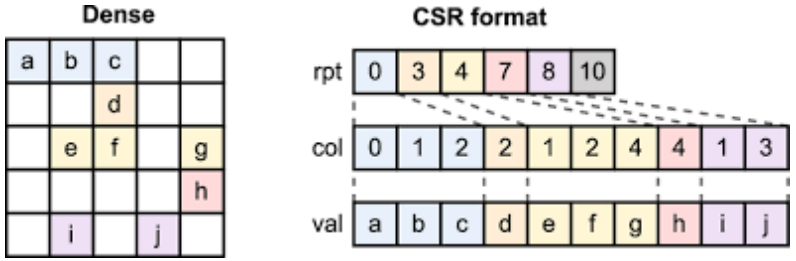


Complete on-chip storage (BRAM + URAM) of weights on U280 FPGA
MatMult-Free Ternary LLM (1.58bits):
370M parameters: ~5K tokens/sec @ 21W vs. ~100 tokens/sec @ 200W

Unstructured Sparsity is hard

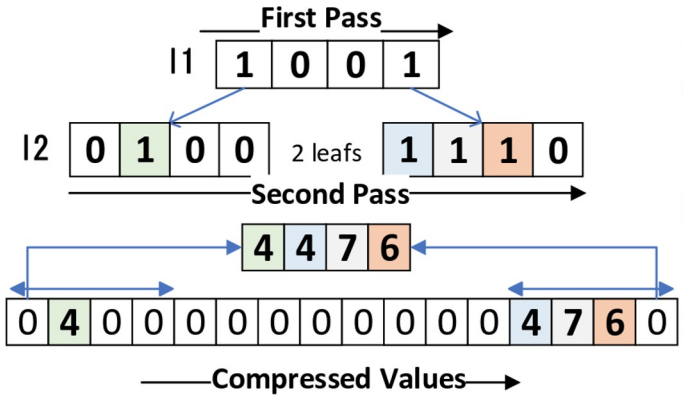
At 4-bit or 8-bit quantization, sparse matrix representation formats used by hardware accelerators (CSR, COO) require more storage than Dense format due to meta-data overhead!!

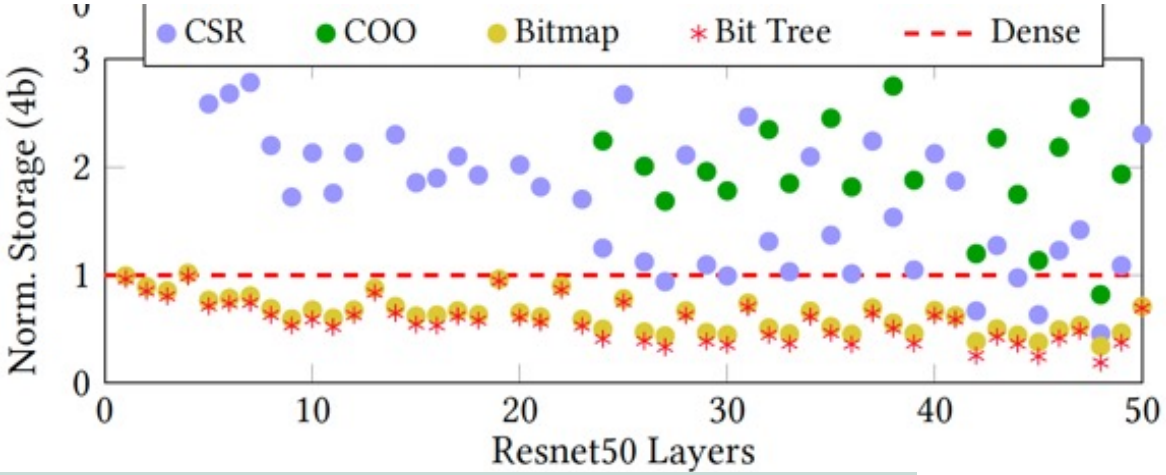




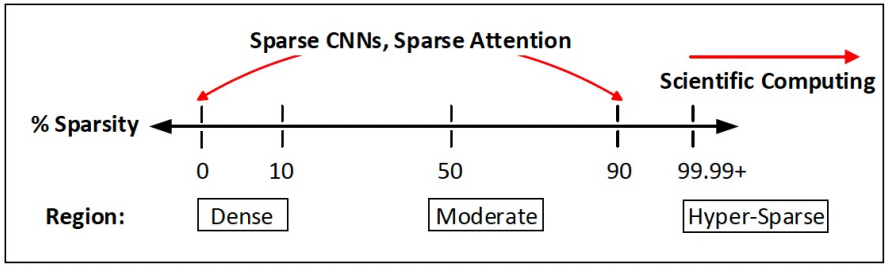
Sparse storage formats lead to inflation not compression!
Bitmap decoding is costly due to long runlength of zeros

Taming Unstructured Sparsity: Bit-Tree Format

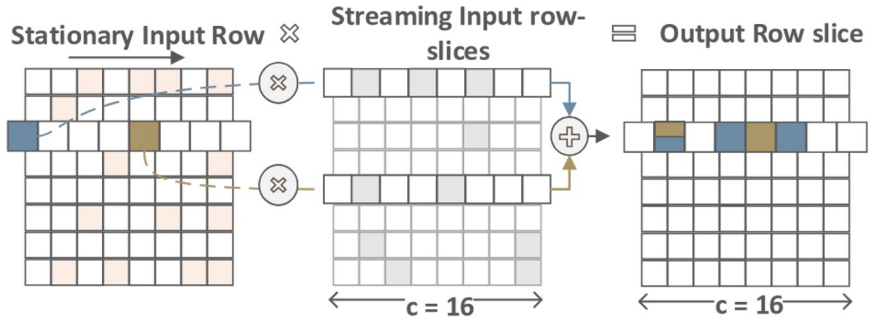




Storage: Same as ideal bitmap
Fast Decoding: N-passes over N-level Bit-Tree
Handles all sparsity level

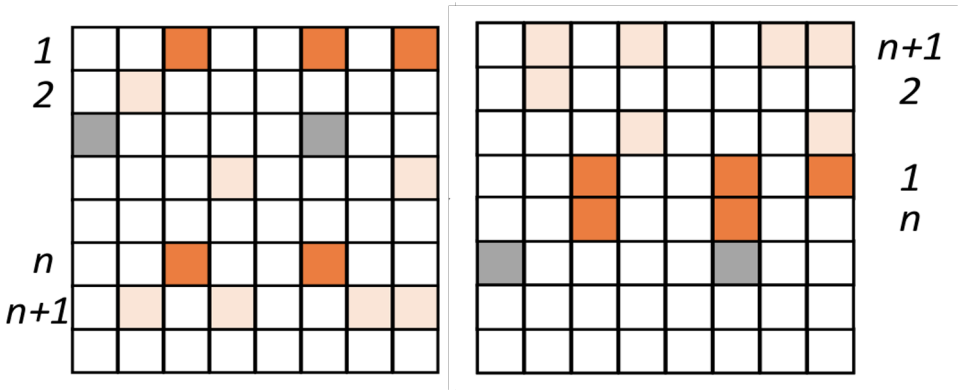


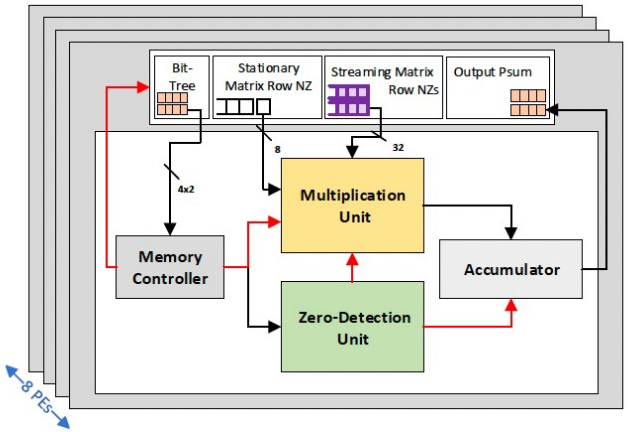
Taming Unstructured Sparsity: Irregular memory access



ZeD ASIC accelerator achieves 3.2x performance-per-area improvement over SOTA for sparse ML workloads

Gustavson's Dataflow: Row-wise product





Reorder stationary matrix rows to maximally reuse streaming rows

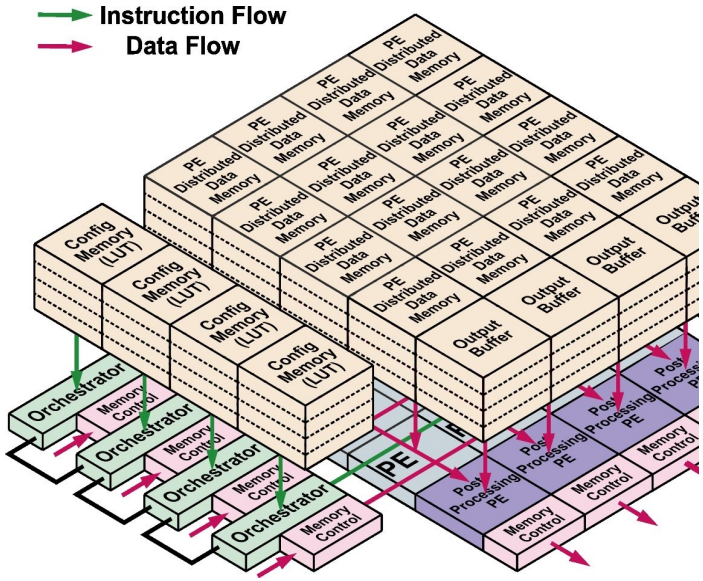

Can one reconfigurable accelerator rule them all?

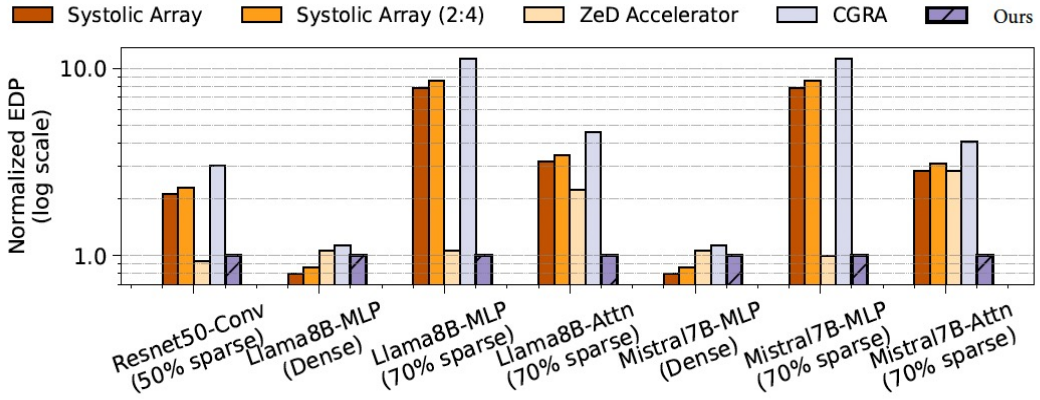
Toward Efficient Intelligence: Rethinking AI Accelerators for Sparsity

- LLMs are growing faster than compute can scale. Sparsity is not optional, it's essential.
- Today's hardware is fundamentally mismatched with emerging sparse workloads.
- Specialized architectures show what's possible; but we need a universal, reconfigurable accelerator.
- Winning the hardware lottery requires co-design: architecture must evolve with algorithms, not after them.

NUS National University of Singapore

Canon: Data-driven Reconfigurable Accelerator

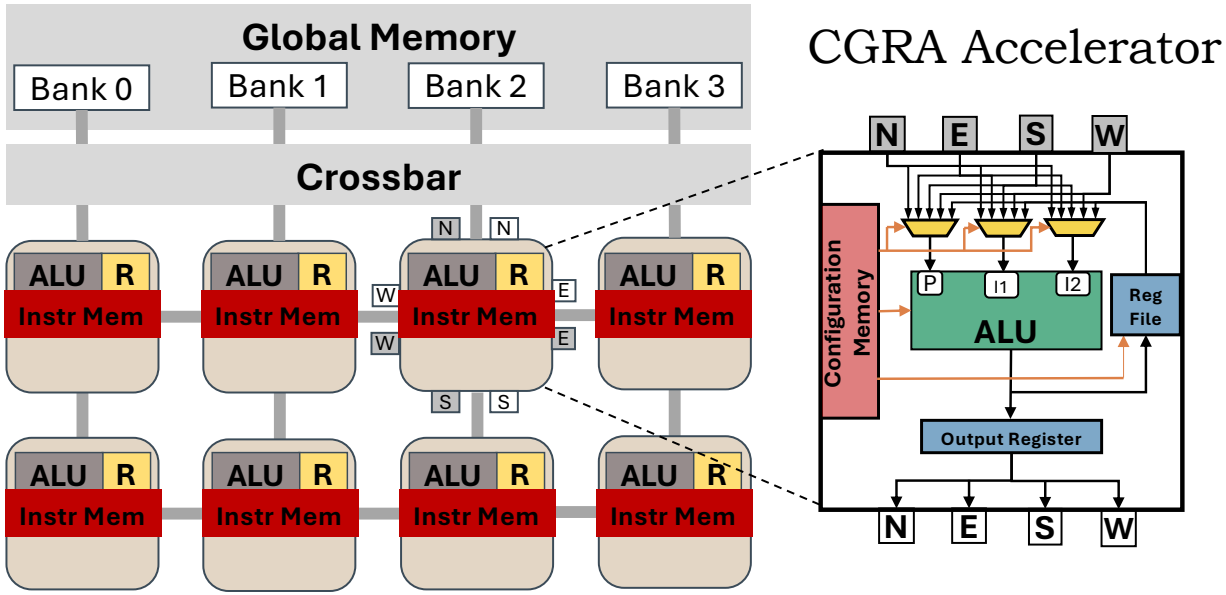




- Minimalist reconfigurable processing array that adapts to diverse sparsity with minimal reconfiguration cost
- Smart algorithm that maps arbitrary dataflows on the fabric
- High parallel processing while handling complex dataflow dependencies

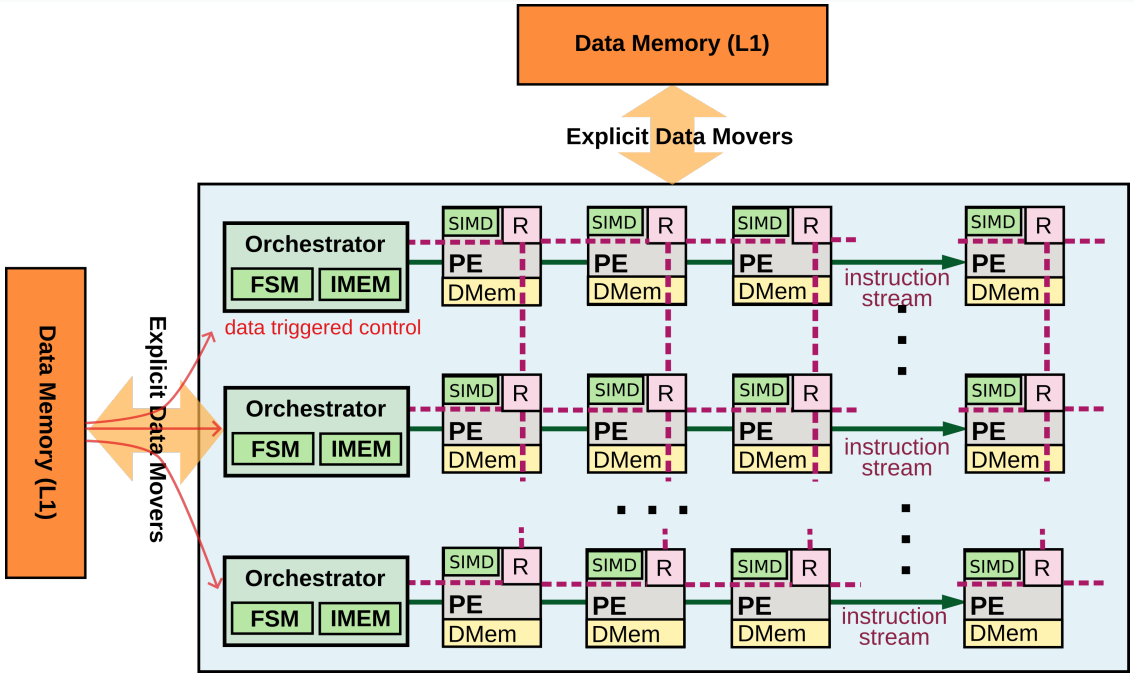
Cost of Reconfigurability

- As data becomes smaller (FP4), instruction cost is very high!
- Domain-specific accelerator has minimal config storage and access cost
- CGRA accelerator introduces config storage and access cost
- Canon achieves efficiency by adapting to workload through a compact data-driven execution orchestrator



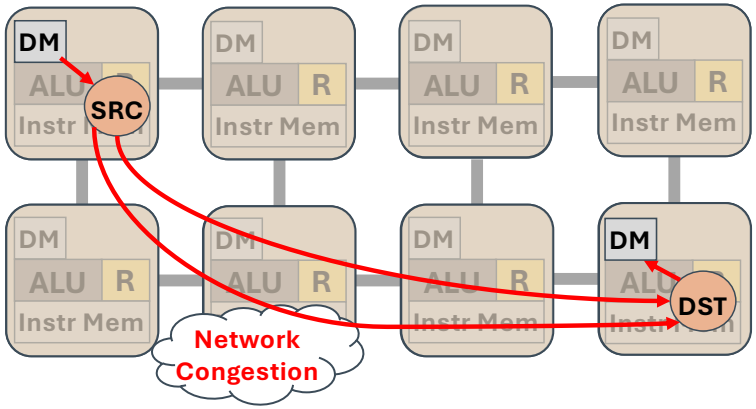
Data-driven Dynamic Execution Orchestration

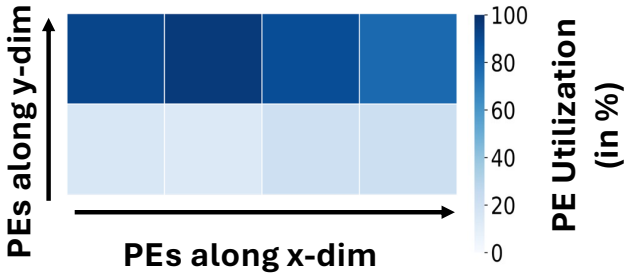
- Canon achieves efficiency by adapting to workload through a compact data-driven execution orchestrator
- Compiler generates FSM for each kernel
- FSM generates config that travels through a row (Time-lapsed SIMD)
- Sparse Metadata triggers dynamic changes to dataflow



Nexus Machine: Adaptive Message Driven Fabric On-route compute on idle PE

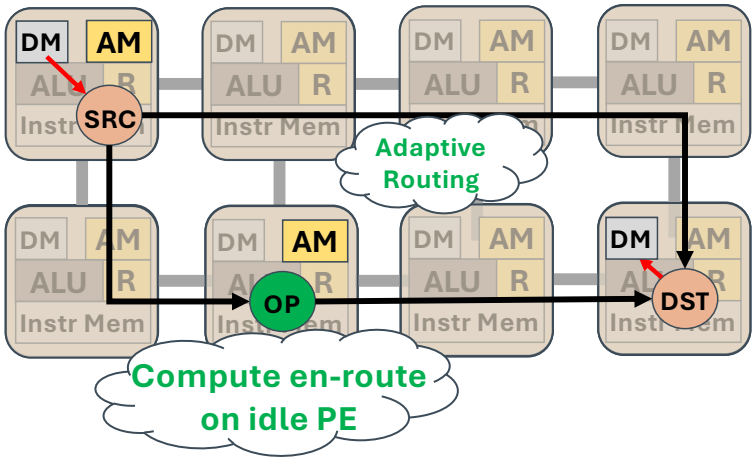
- Data-driven execution of irregular workload causes network congestion and load imbalance

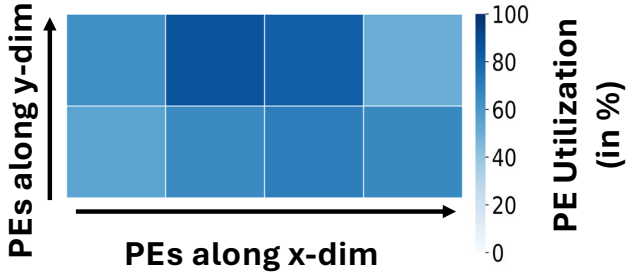




Nexus Machine: Adaptive message-driven fabric On-route compute on idle PE

- Active Messages carrying both instruction and data trigger compute en-route on idle PE improving load balancing

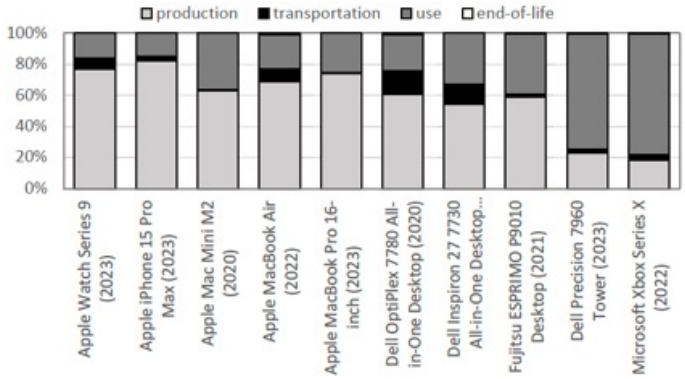




Reconfiguration offers reliability, sustainability benefits!

Embodied footprint: Raw material extraction, manufacturing, assembly, transportation, end-of-life processing

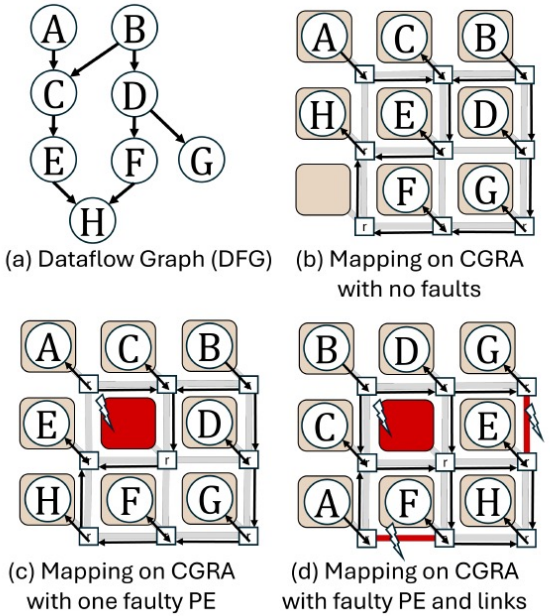
Operational footprint: Device use during its entire lifetime



Embodied footprint dominated most consumer computing devices

Concurrency (n)	Carbon Footprint Improvement	
	Avg Util ($n' < n$)	100% Util ($n' = n$)
1	-	7.60×
2	6.10×	3.84×
3	4.12×	2.59×
4	3.12×	1.97×
5	2.53×	1.59×

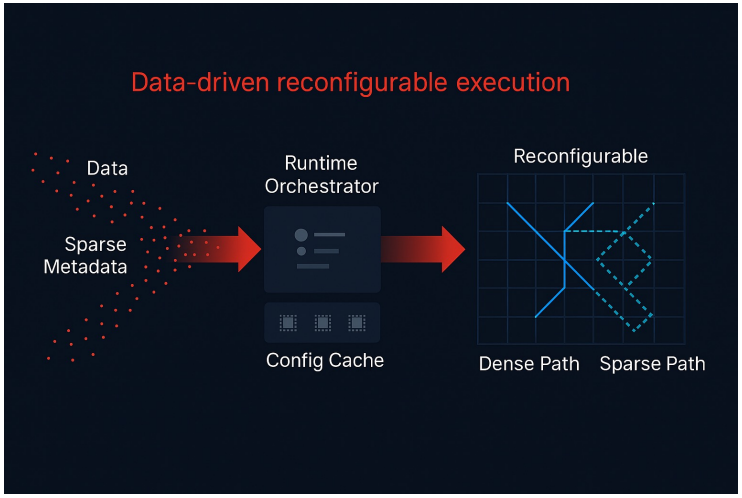
Carbon footprint saving by replacing SoC IP blocks with CGRA



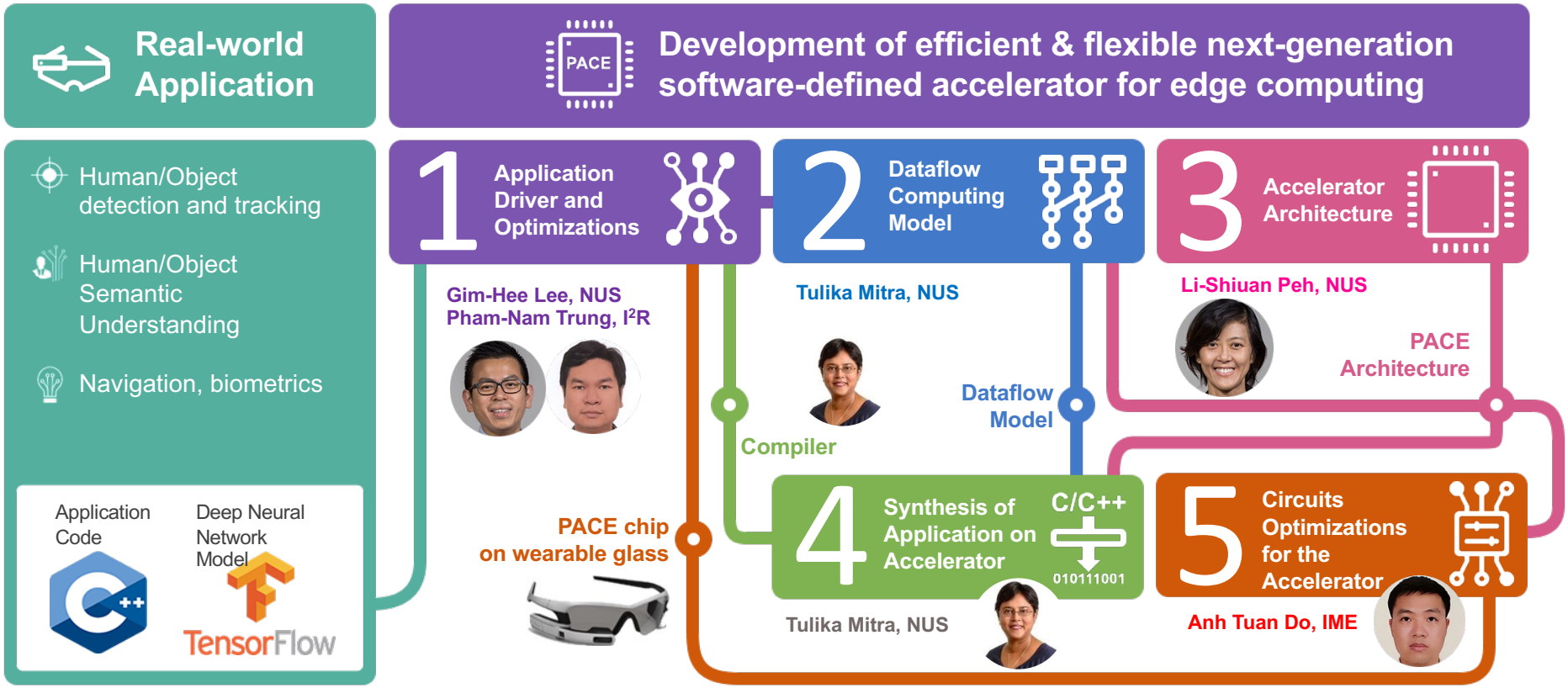
CGRA accelerators circumvents faults

Data-Driven Reconfigurable Accelerator

- Then: GPU won by becoming general-purpose from a graphics chip (architecture + CUDA + tools)
- Now: GPU Tensor Cores → Dense AI focus
- Next: Data-driven reconfigurable execution as the new general-purpose
- Payoff: Default fabric of the 2030s with near-ASIC perf/W



PACE Programme: A synergistic Cross-Layer Approach



National Research Foundation Singapore Competitive Research Programme (NRF-CRP) ~$6M for 5 years
PACE: Next-Generation IoT Edge Computing through Efficient Software Programmable Accelerators

GAIA: Rethinking data centre computation and communication design for Green AI

Achieve sustainable AI with energy-efficient sparse architecture
$9.2M Funding from Ministry of Education Singapore: July 2025-2030

Lead PI **Tulika Mitra**, Provost's Chair Professor, NUS Computing
Team PI **Trevor Carlson**, Assistant Professor, NUS Computing
Bingsheng He, Professor, NUS Computing
Jialin Li, Assistant Professor, NUS Computing
Li-Shiuan Peh, Provost's Chair Professor, NUS Computing

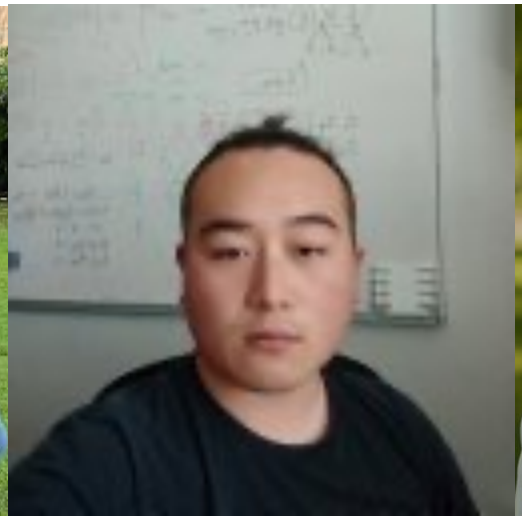
Collaborator:
AMD Research & Advanced Development: Singapore, USA, Ireland

Prof. Gustavo Alonso, ETH Zurich
Prof. Lieven Eeckhout, Ghent Univ.

NUS National University of Singapore

Thank You






Li-Shiuan Peh

Pranav Dangi
dangi@nus.edu.sg

Zhenyu Bai
zhenyu.bai@nus.edu.sg

Rohan Juneja
rohan@comp.nus.edu.sg

We are Hiring!

NUS National University of Singapore

Acknowledgments

- Anh Tuan Do
- Chen Liang
- Tan Cheng
- Manupa Karunaratne
- Wang Bo
- Aditi Kulkarni
- Dhananjaya Wijerathne
- Zhaoying Li
- Thilini Kaushalya Bandara
- Dan Wu
- Vishnu Paramasivam
- Chong Yi Sheng

National Research Foundation
Prime Minister's Office
Singapore

A*STAR Institute of Microelectronics

Ministry of Education
Singapore

AMD

Black Sesame Technologies

XILINX

NUS National University of Singapore