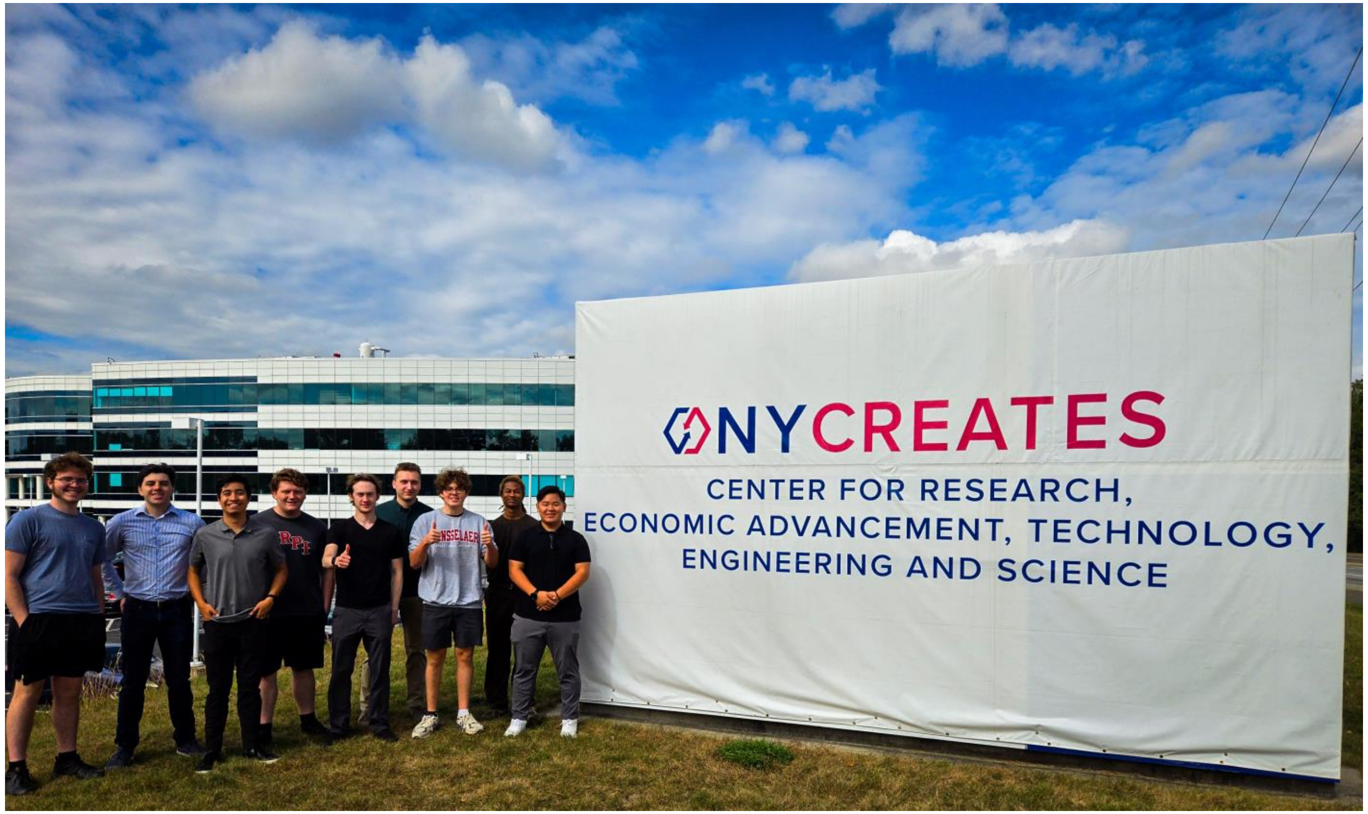
Breaking the Bottleneck

Tackling The Rising Costs of Moving Data

IEEE EPS @ RPI
David King

IEEE EPS at RPI General Updates



CMP for Hybrid Bonding Seminar

- Oct. 29th at Albany Nano
- Free / Lunch included

2nd Albany Nanotech Complex Tour

- Nov. 13th / Sign up QR at the end

ITherm Challenge

Mini colloquium

- Date: TBD / Conf. with Industry (IBM)

What are you learning today?

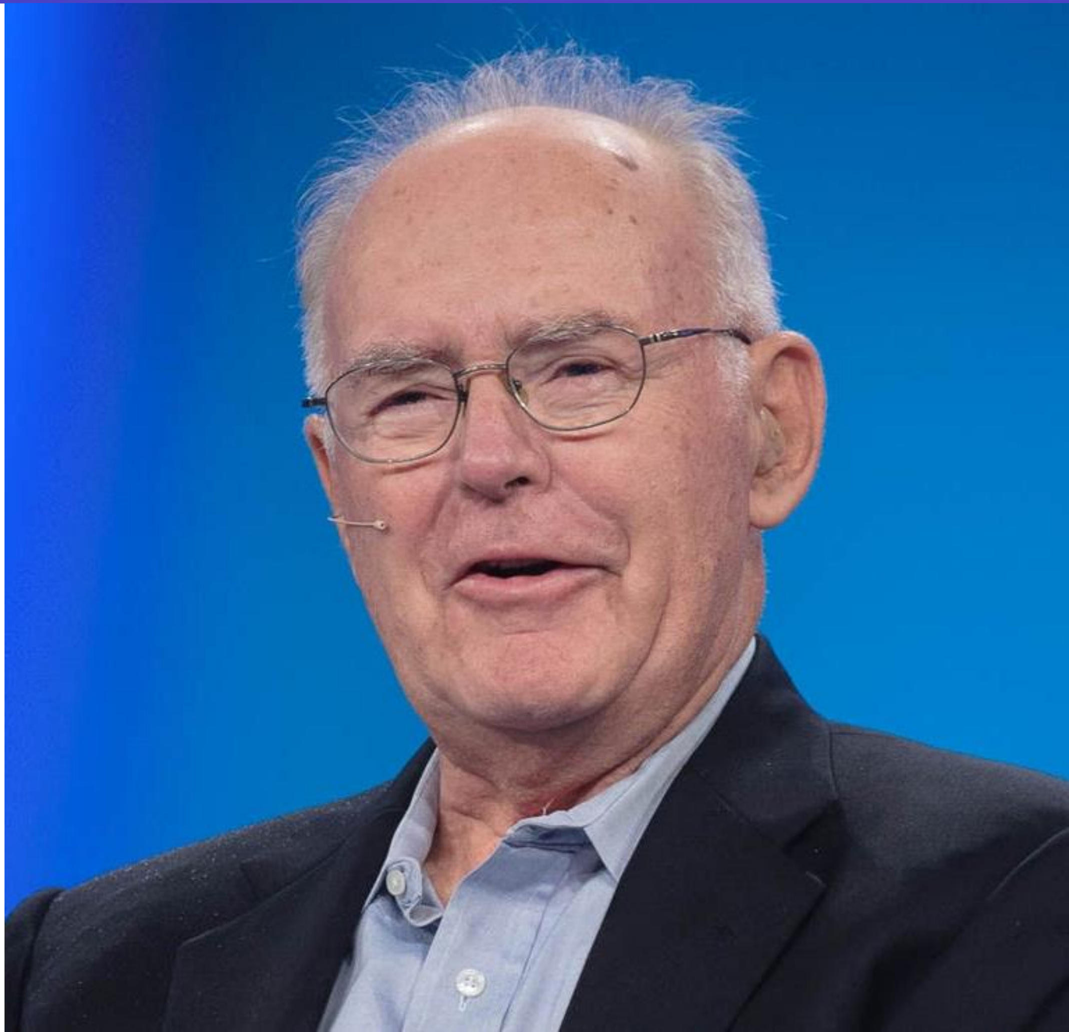
1. The Memory Wall
2. Why is it forming?
3. Workload Nuclear Bomb
4. Solutions

The beginning: Moore’s Law

- In 1965, Gordon Moore predicted the exponential growth of the number of transistors on an IC

- Transistor count would double every two years since creation of law.

- Many say the law is self-fulfilling



Moore's Law is Great

- 1.4x Annual Performance Improvement for 40+ Years ≈ 10,000x
  - More transistors -> More complexity
  - Less Capacitance and Lower $\boldsymbol{V_{dd}}$ -> Less Power Used
  - Less Capacitance and Higher Saturation Current -> Higher Clock Speed

Time to charge capacitor:

$$T = RC$$

Capacitance defined as:

$$C = \frac{\varepsilon A}{d}$$

Energy Per Cycle:

$$P = CV^2$$

Scaling:

$$L = \frac{1}{S} \quad W = \frac{1}{S} \quad D = \frac{1}{S}$$

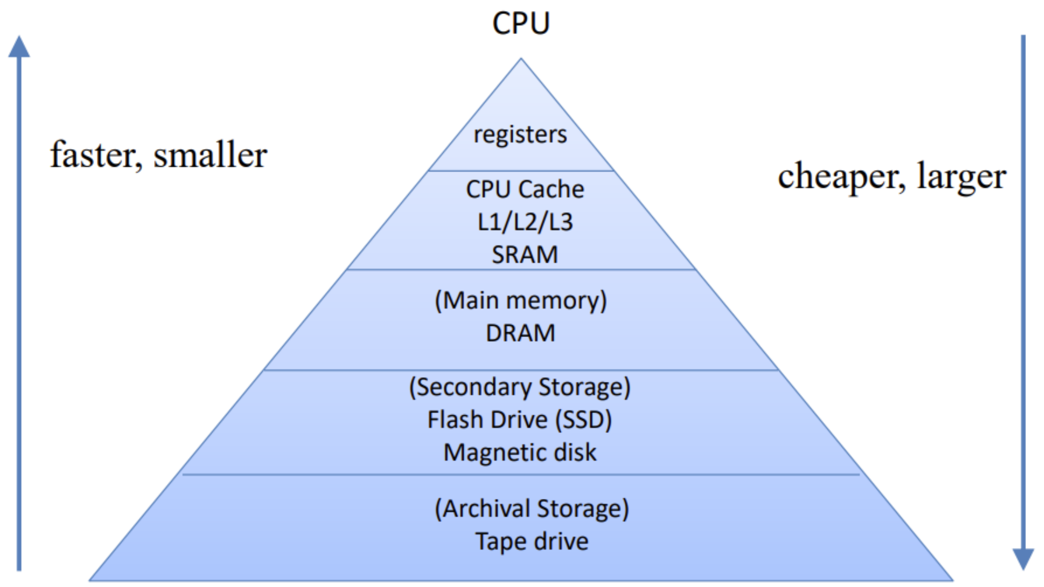
$$C = \frac{\frac{1}{S} * \frac{1}{S}}{\frac{1}{S}} = \frac{1}{S}$$

$$T = \frac{1}{S}$$

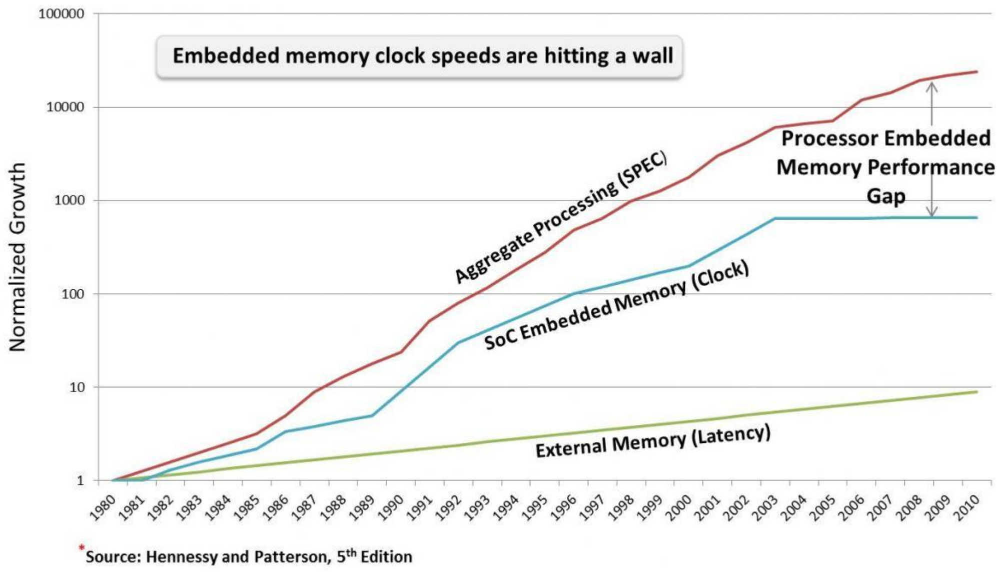
$$P = \left(\frac{1}{S}\right)\left(\frac{1}{S}\right)^2 = \frac{1}{S^3}$$

But Only For The CPU

- Moore's law does not apply across the entire computer
- Different components have different scaling requirements



A growing disparity



*Source: Hennessy and Patterson, 5th Edition

Figure 1: Embedded Memory Performance Gap is Getting Worse

Historically, DRAM capacity doubles every 3.5 years

The difference between the speeds will scale exponentially.

“Although the disparity between processor and memory speed is already an issue, downstream someplace it will be a much bigger one” (Wulf Et Al)

How big and how soon?

$$t_{avg} = p \cdot t_c + (1-p) \cdot t_m$$

Let's use the expected value formula, to calculate the expected access time as cache misses increase

Definitions:

- p: The probability that the memory is in CPU cache
- t_c: Time in clock steps to access memory in CPU
- t_m: Time in clock steps to access memory off the CPU

Assumptions:

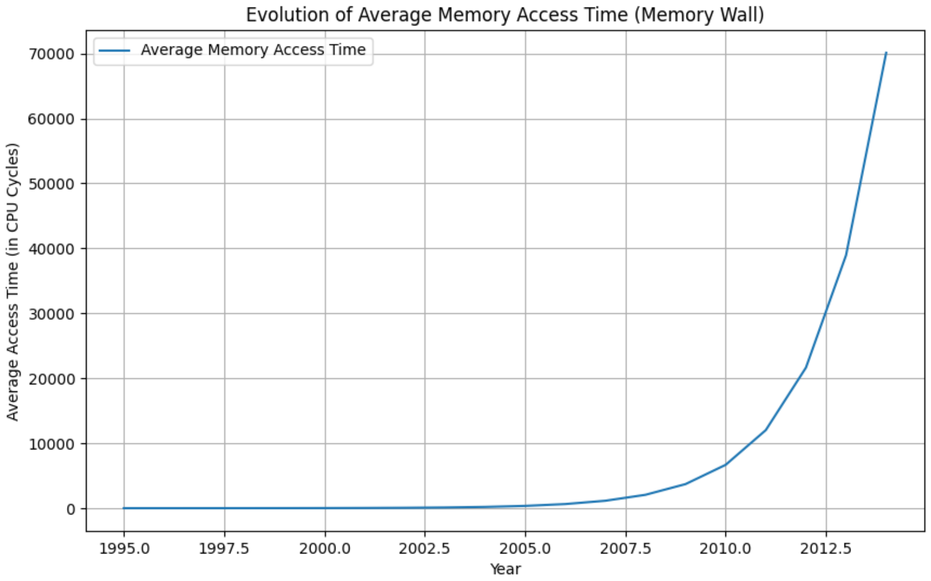
1. 20% of instructions reference memory (in reality its 20-40%)
2. The cache never has a conflict or capacity miss
3. $t_c = 1$ it just makes calculation easier

The Results

Assuming 20% or 1/5 instructions references memory

The moment we get to $t_{avg} = 5$ the performance will be completely bottlenecked

This happens in 1998 according to our sim



Why is this happening?

Economics

The memory industry would rather scale capacity over speed

- **Marketability:** Harder to market latency drops compared to size increases
- **Cost-Per-Bit Reduction**: Scaling capacity makes memory solutions more cost-effective for:
  - Enterprise
  - Consumer
- Speed issues can be masked with cache systems and prefetching to an extent.

Heat Sensitivity and Device Level Limitations

DRAM cells are packed extremely tightly

Heat Issues -> Transistor Leakage Current

The capacitor can discharge faster then expected, and data can be corrupted, if not refreshed in time.

This is why lower latencies are difficult to achieve

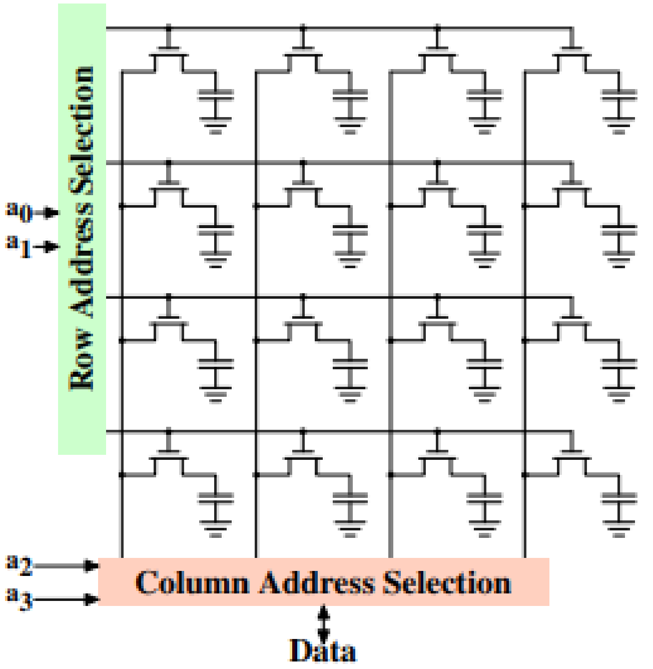


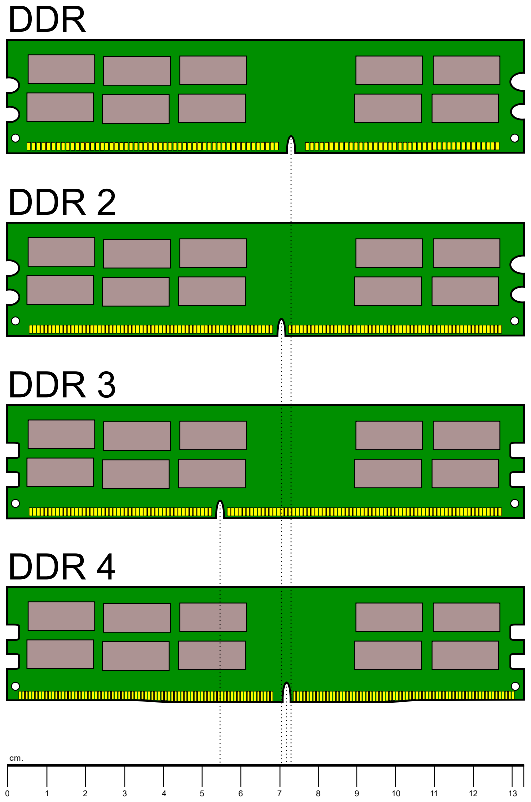
Figure 2.7: Dynamic RAM Schematic

Standard Limitations

If you can’t decrease latency, increase bandwidth, but you can't increase bandwidth alone.

JEDEC DDR Standard Adherence is Required:

1. Limitations on allowed frequency range
2. Limitations on bandwidth



Something is coming

AI Explosion

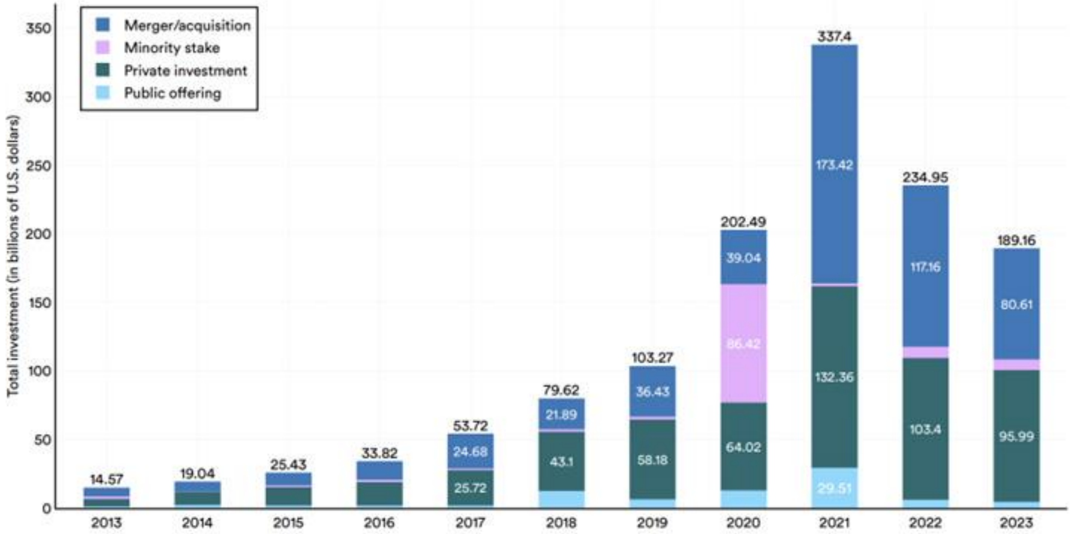
Expected CAGR of 37.3%

According to Grand View Research

- AI is expected to contribute more than the current output of India and China combined, to the world economy by 2030.

According to Forbes

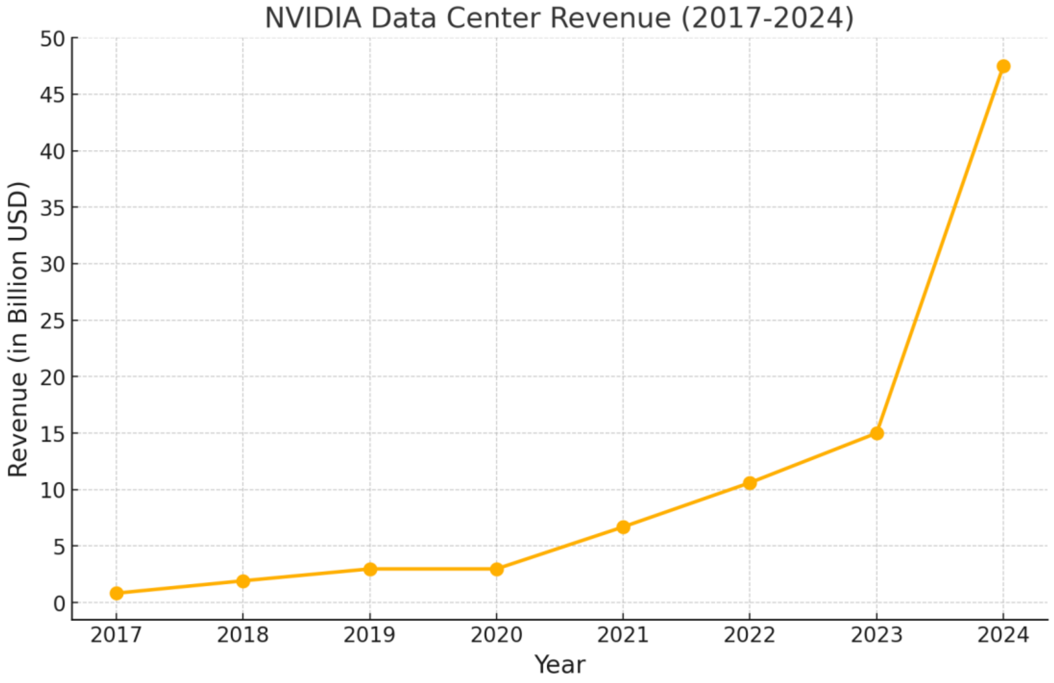
- AI is expected to contribute a significant 21% net increase in the US GDP by 2030.



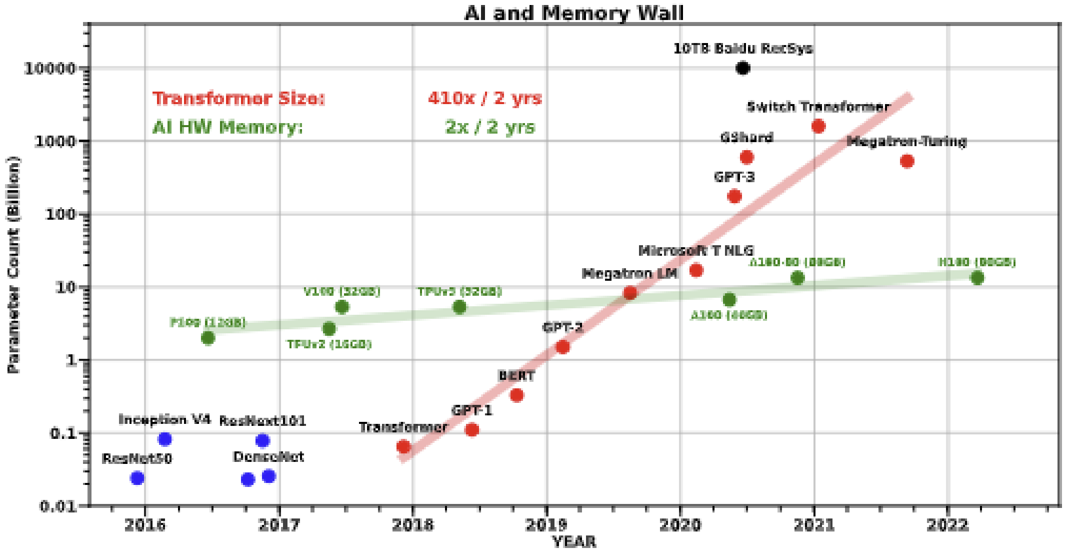
Source: Nestor Maslej, Loredana Fattorini, Raymond Perrault, Vanessa Parli, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald and Jack Clark, AI Index Steering Committee, Institute for Human-Centered AI, Stanford University, Stanford, CA, April 2024. ("The AI Index 2024 Annual Report" hereafter)

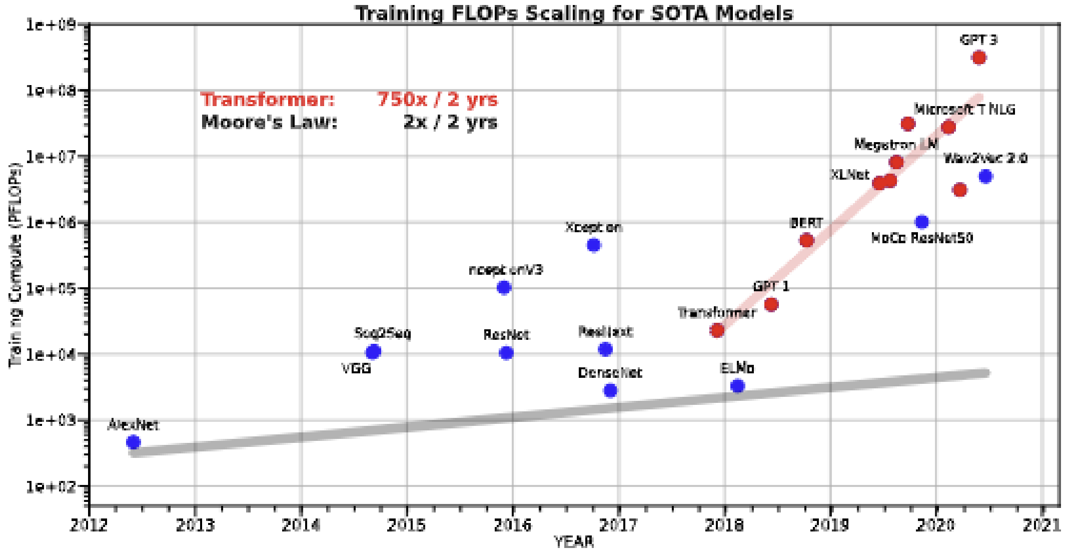
Impact on Chip Industry

- Global AI chip market size is set to reach $82.25 billion by 2027.
- Expected to grow at a CAGR of 35% during 2019-2027
- We need more AI chips!



AI Workload Scaling



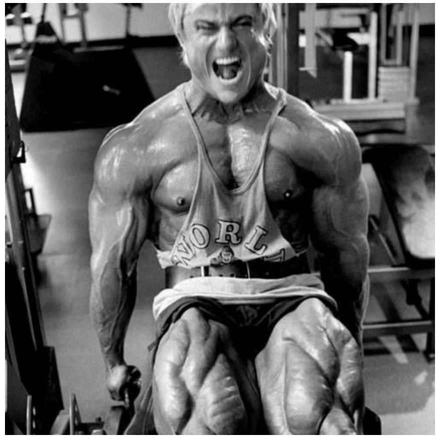


Architectural Issues

Scaling:

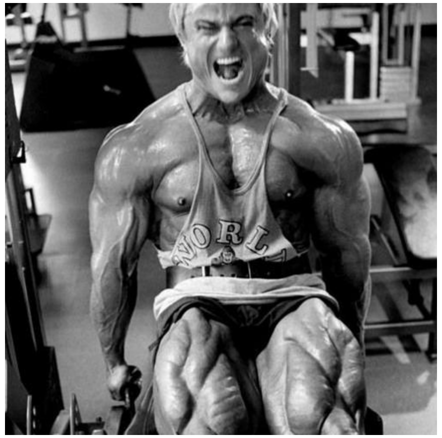
- FLOPS required for training: 750x / 2yrs
- LLM sizes: 410 x / 2yrs

Not a new problem. Just worse in the ways that count.



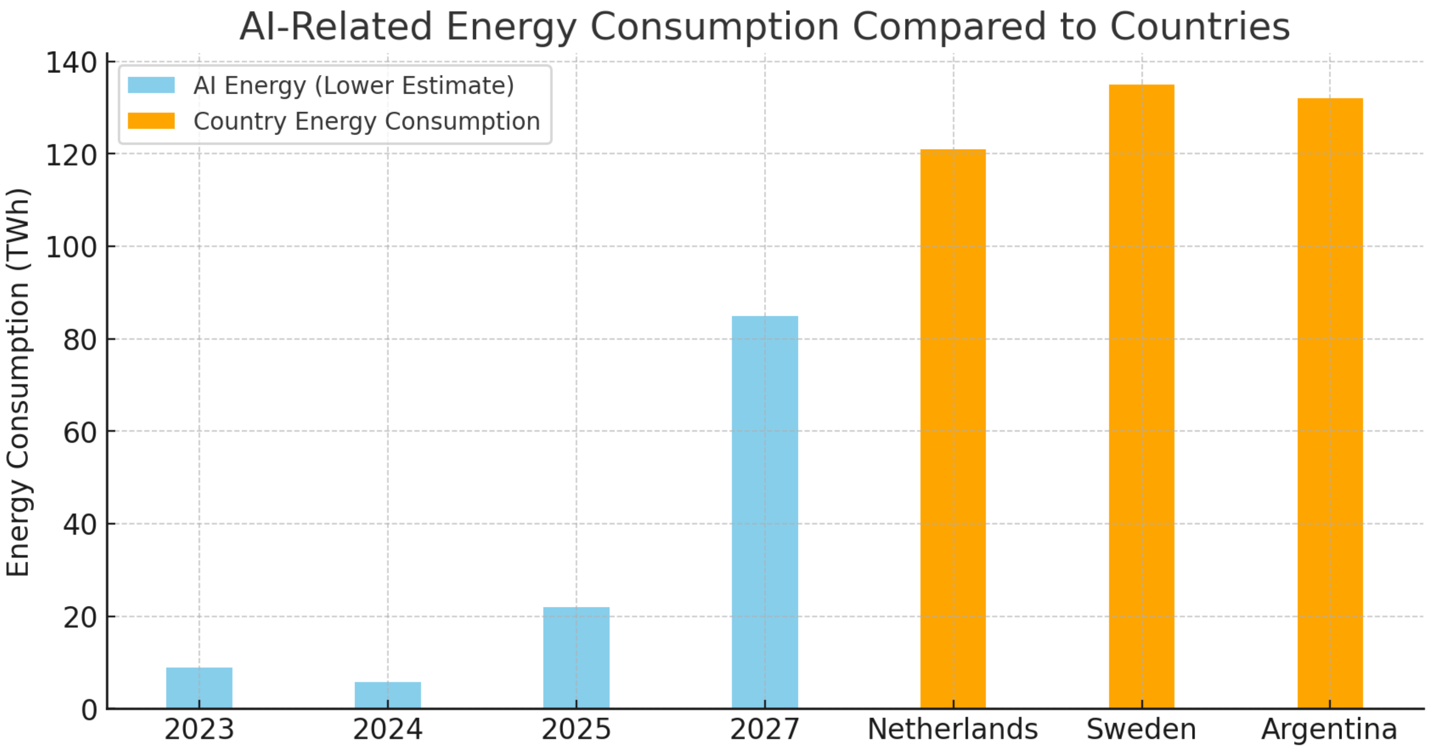
Main Memory

Interconnect

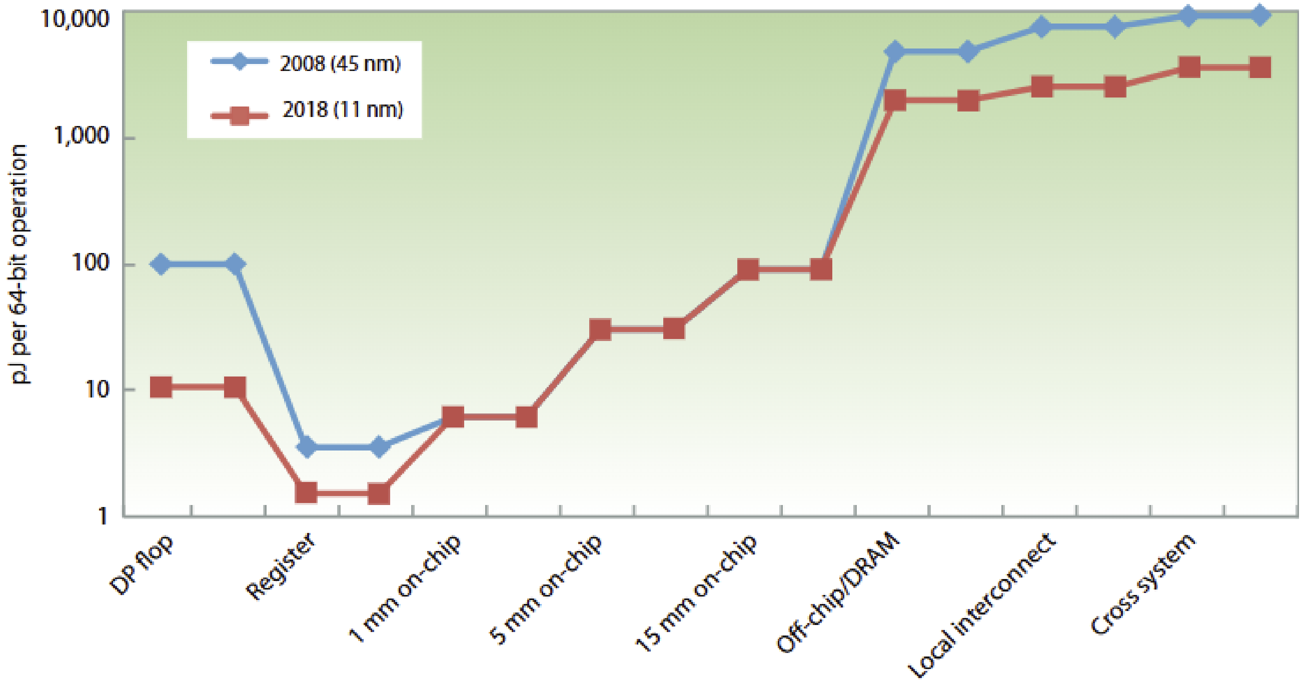


Compute

Energy may be a small problem



Where is this energy spent?



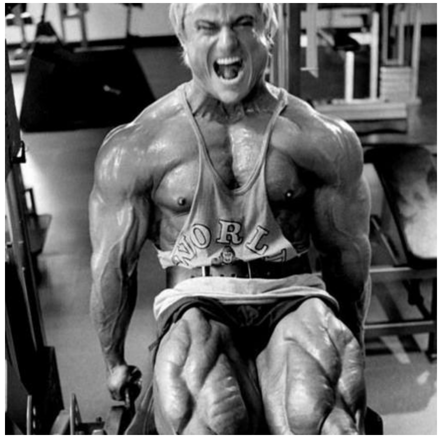
Unsurprisingly, energy increases with distance

Higher voltage from amplifiers/drivers, uses more power

Copper interconnects have higher resistance

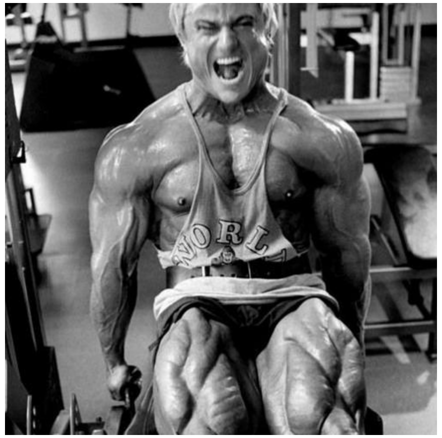
Problem Overview

- The fundamental problem revolves around rising cost of moving data:
  - Energy domain
  - Time domain
- Future gains in performance and efficiency are undermined



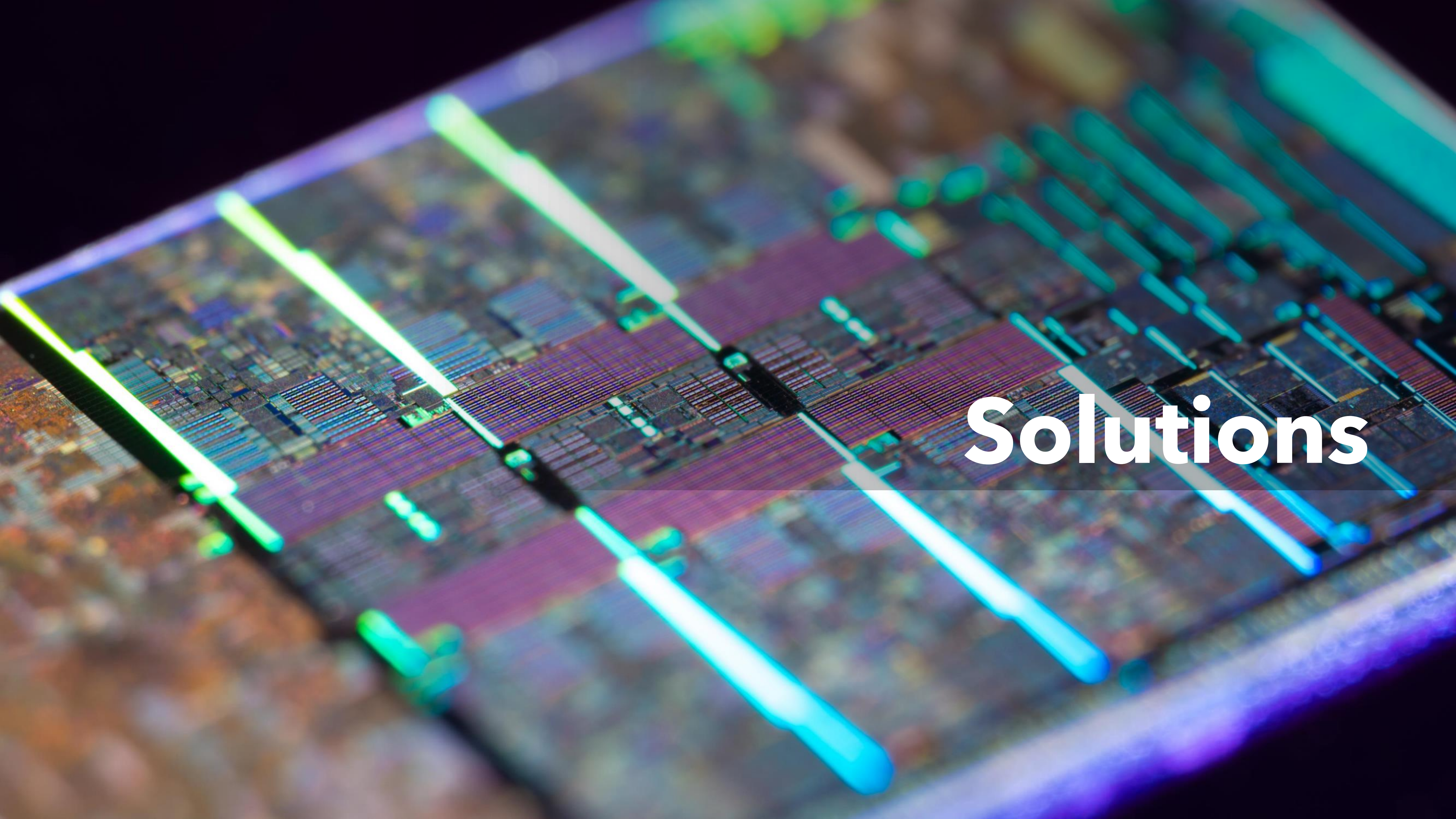
Main Memory

Interconnect



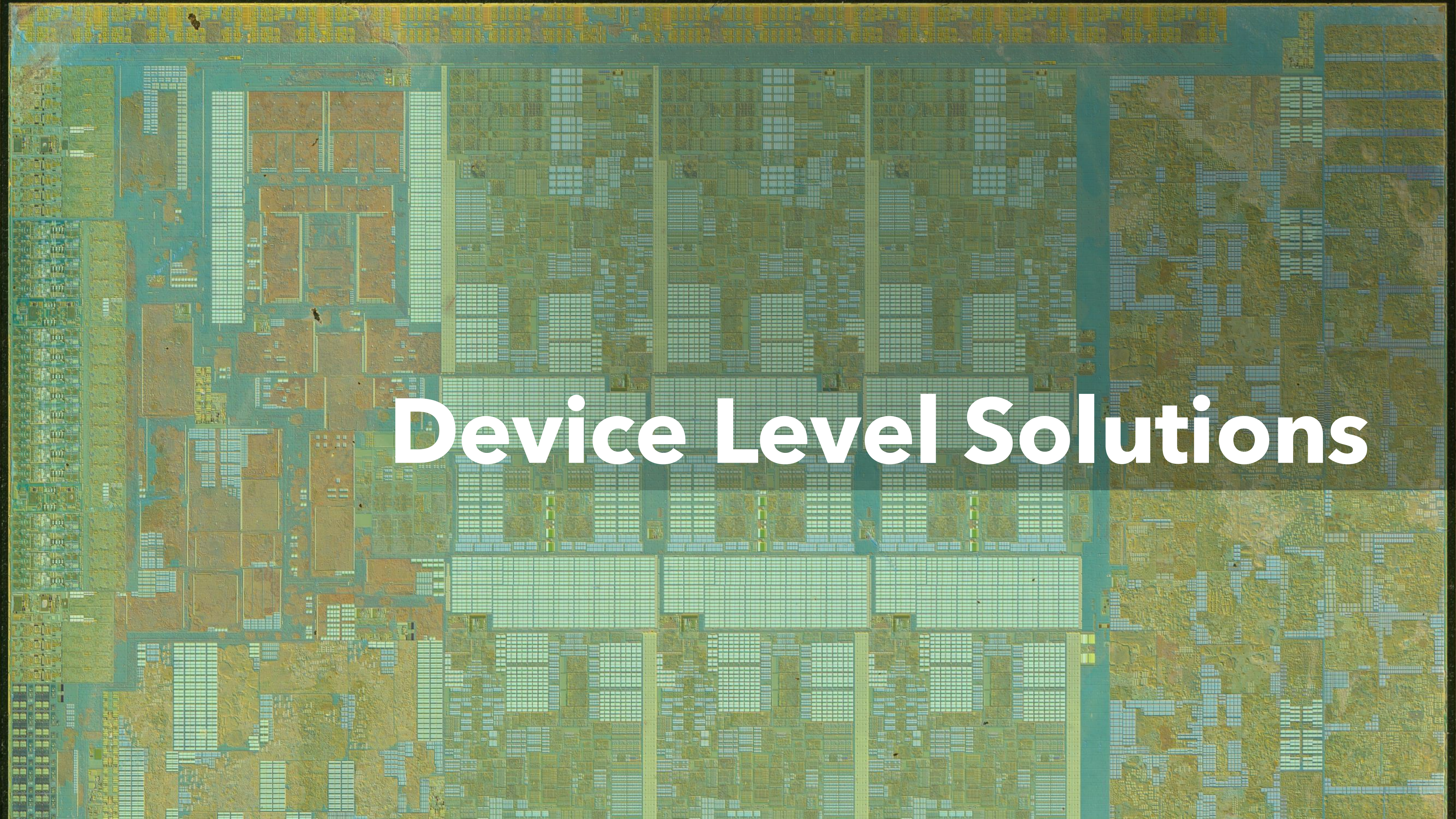
Compute

Solutions



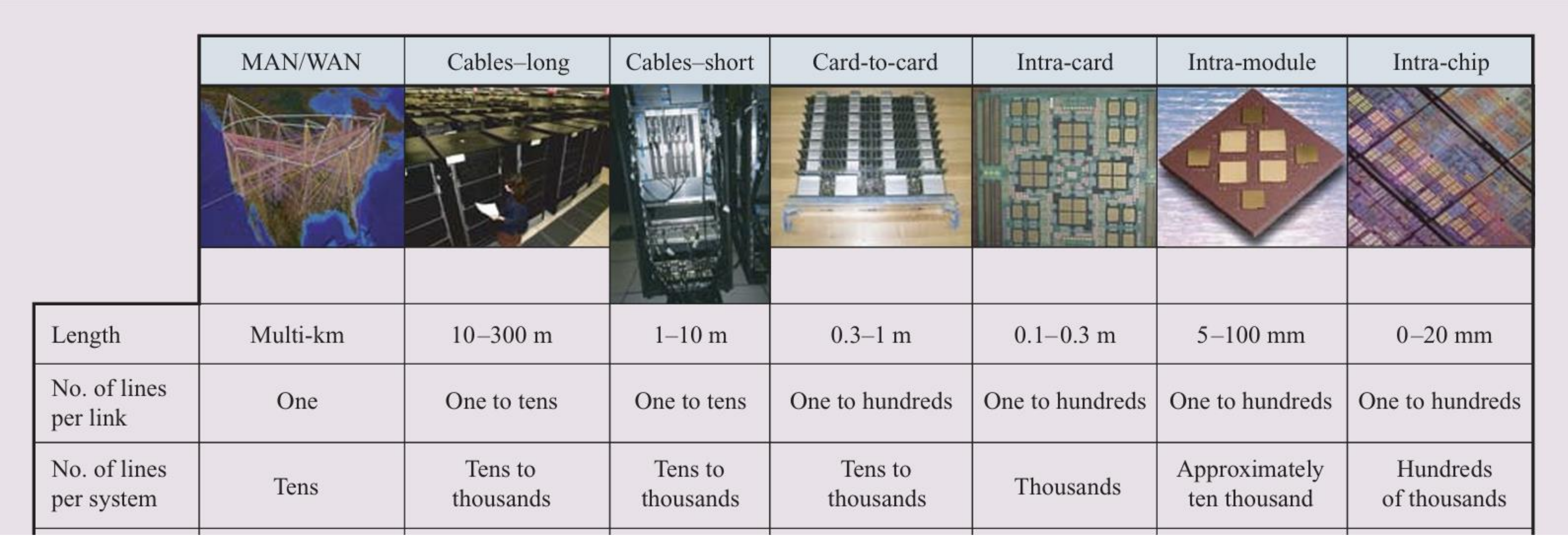
Solution Domains

Application	Software-Hardware Awareness Decrease Memory Accesses
Architectural	Data Locality Scalability Bandwidth
Device	Power Efficiency Bandwidth Latency



Device Level Solutions

Photonics



	MAN/WAN	Cables–long	Cables–short	Card-to-card	Intra-card	Intra-module	Intra-chip
Length	Multi-km	10–300 m	1–10 m	0.3–1 m	0.1–0.3 m	5–100 mm	0–20 mm
No. of lines per link	One	One to tens	One to tens	One to hundreds	One to hundreds	One to hundreds	One to hundreds
No. of lines per system	Tens	Tens to thousands	Tens to thousands	Tens to thousands	Thousands	Approximately ten thousand	Hundreds of thousands

Off-Chip DRAM Interconnects

Pushing photonics further up the memory hierarchy

Decrease power consumption and increase throughput

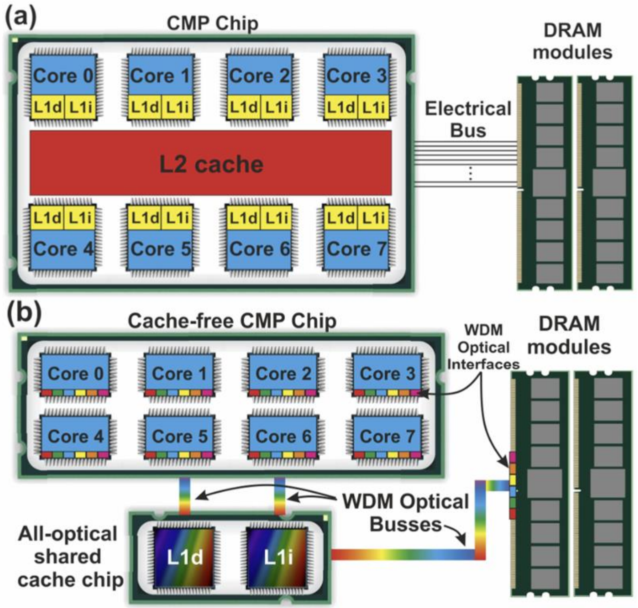


Fig. 10. (a) Conventional CMP architecture with on-chip Cache Memories and Electrical Bus for CPU-MM communication (b) The proposed CMP architecture with off-chip optical Cache Memories between CPU-MM and Optical Busses between them.

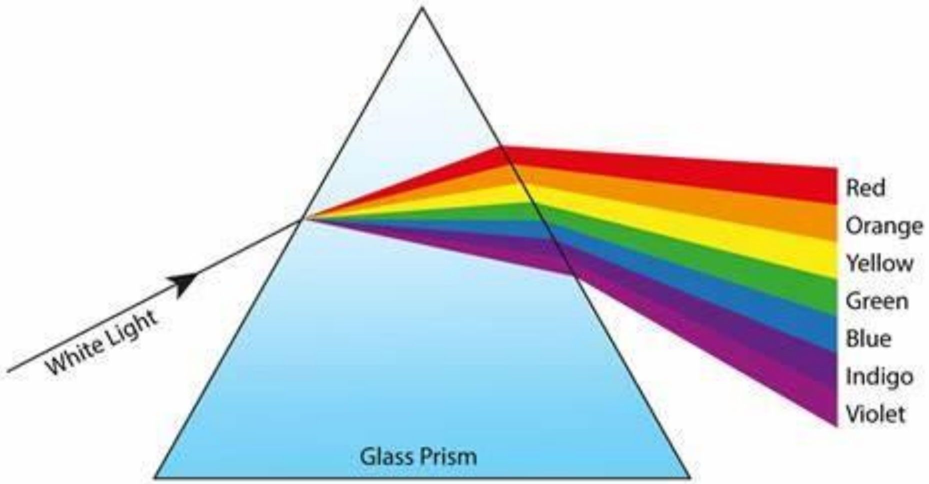
Wavelength Division Multiplexing

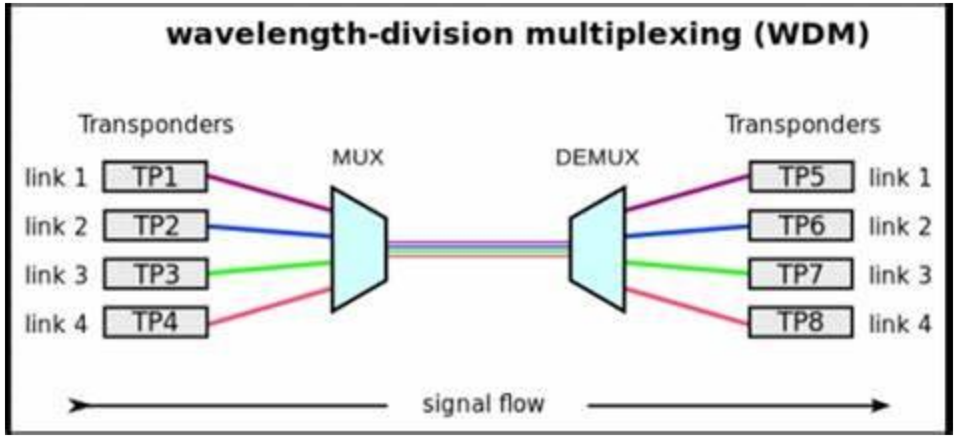
Uses non-interfering wavelengths of light to create channels.

Multiple channels can be passed in a single beam of light

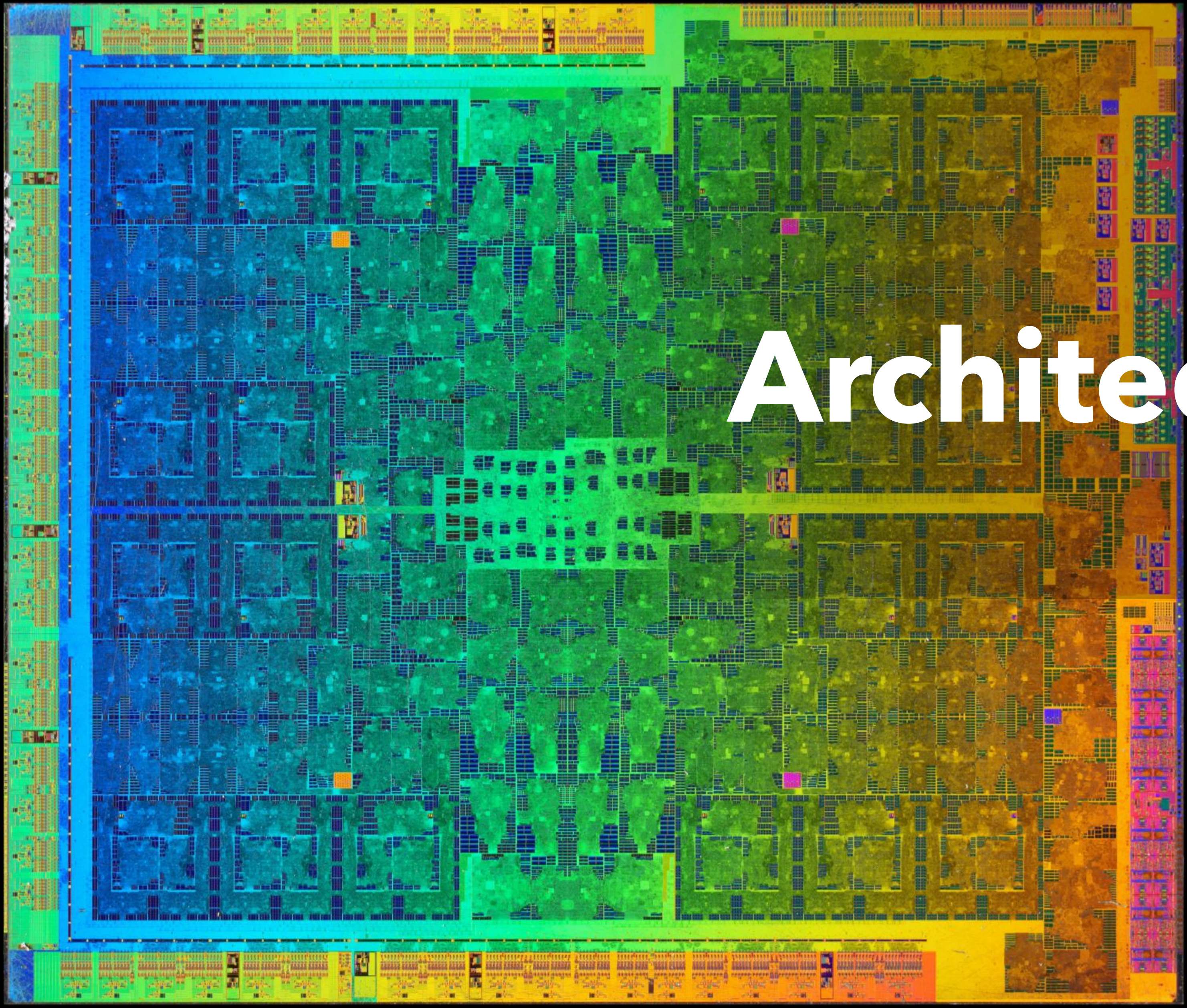
Increases bandwidth of single strand of fiber

Many form of modulation can be used, AM is most common

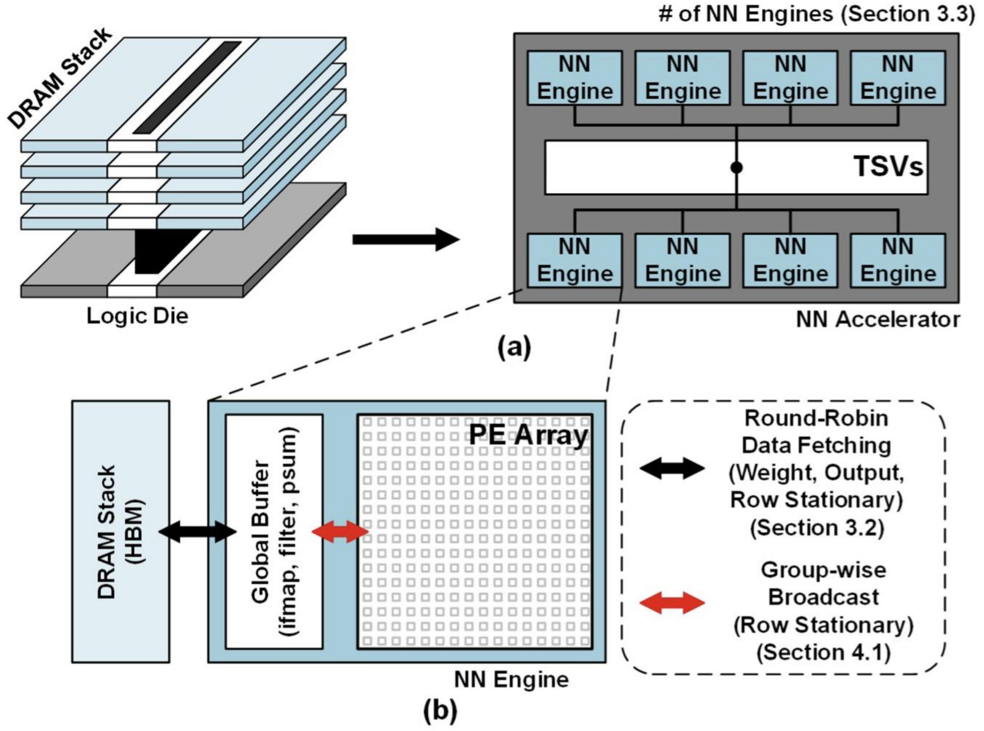




Architectural Level Solutions



Near-Memory Computing



DRAM Stack
Logic Die
# of NN Engines (Section 3.3)
NN Engine
NN Engine
NN Engine
NN Engine
TSVs
NN Engine
NN Engine
NN Engine
NN Engine
NN Accelerator
(a)
DRAM Stack (HBM)
Global Buffer (ifmap, filter, psum)
PE Array
NN Engine
(b)
Round-Robin Data Fetching (Weight, Output, Row Stationary) (Section 3.2)
Group-wise Broadcast (Row Stationary) (Section 4.1)

The name of the game is moving data closer to processing elements

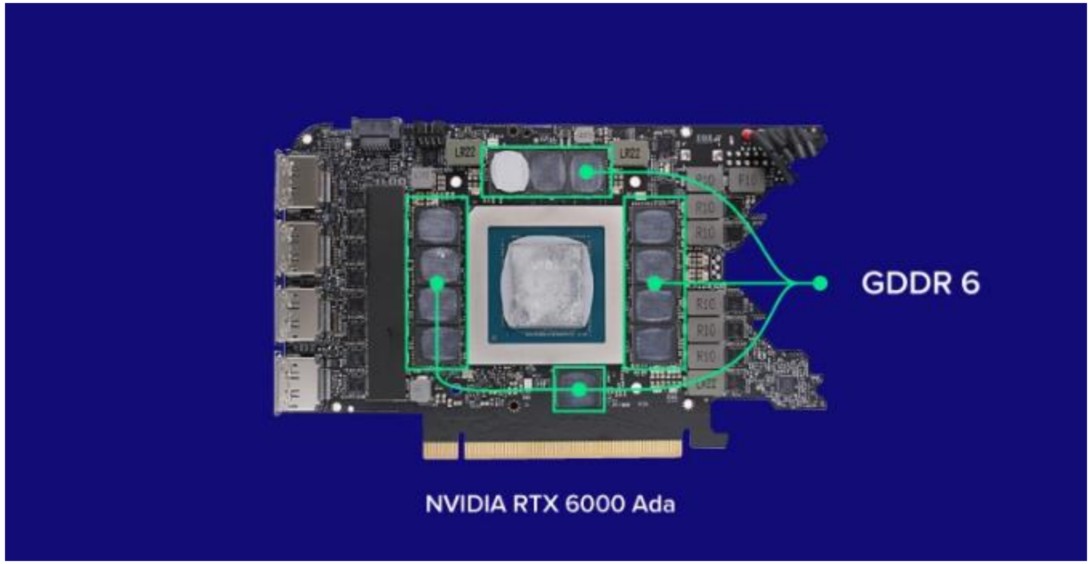
Graphics Double Data Rate

Main advantages:

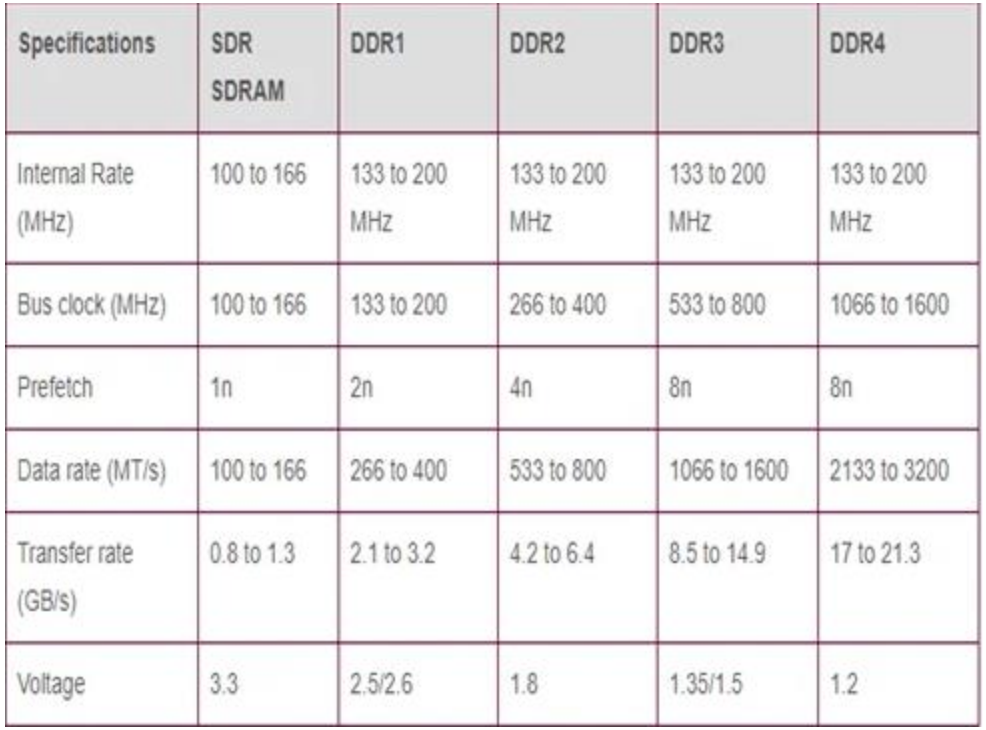
- Greater data locality (Ideally next to PE)
- Uses 170 BGA package directly instead of going through DIMM
- Goes through substrate, requires SerDes

Main Disadvantages:

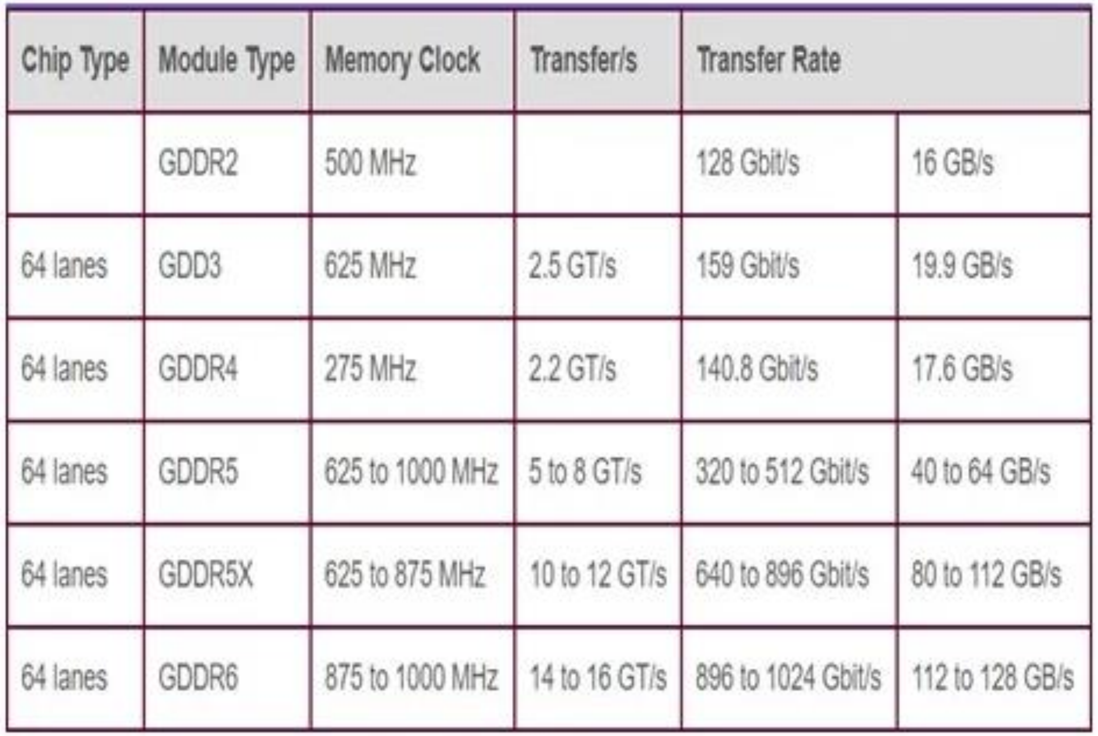
- Slower
- Less power efficient



Graphics Double Data Rate



Specifications	SDR SDRAM	DDR1	DDR2	DDR3	DDR4
Internal Rate (MHz)	100 to 166	133 to 200 MHz	133 to 200 MHz	133 to 200 MHz	133 to 200 MHz
Bus clock (MHz)	100 to 166	133 to 200	266 to 400	533 to 800	1066 to 1600
Prefetch	1n	2n	4n	8n	8n
Data rate (MT/s)	100 to 166	266 to 400	533 to 800	1066 to 1600	2133 to 3200
Transfer rate (GB/s)	0.8 to 1.3	2.1 to 3.2	4.2 to 6.4	8.5 to 14.9	17 to 21.3
Voltage	3.3	2.5/2.6	1.8	1.35/1.5	1.2



Chip Type	Module Type	Memory Clock	Transfer/s	Transfer Rate	
	GDDR2	500 MHz		128 Gbit/s	16 GB/s
64 lanes	GDD3	625 MHz	2.5 GT/s	159 Gbit/s	19.9 GB/s
64 lanes	GDDR4	275 MHz	2.2 GT/s	140.8 Gbit/s	17.6 GB/s
64 lanes	GDDR5	625 to 1000 MHz	5 to 8 GT/s	320 to 512 Gbit/s	40 to 64 GB/s
64 lanes	GDDR5X	625 to 875 MHz	10 to 12 GT/s	640 to 896 Gbit/s	80 to 112 GB/s
64 lanes	GDDR6	875 to 1000 MHz	14 to 16 GT/s	896 to 1024 Gbit/s	112 to 128 GB/s

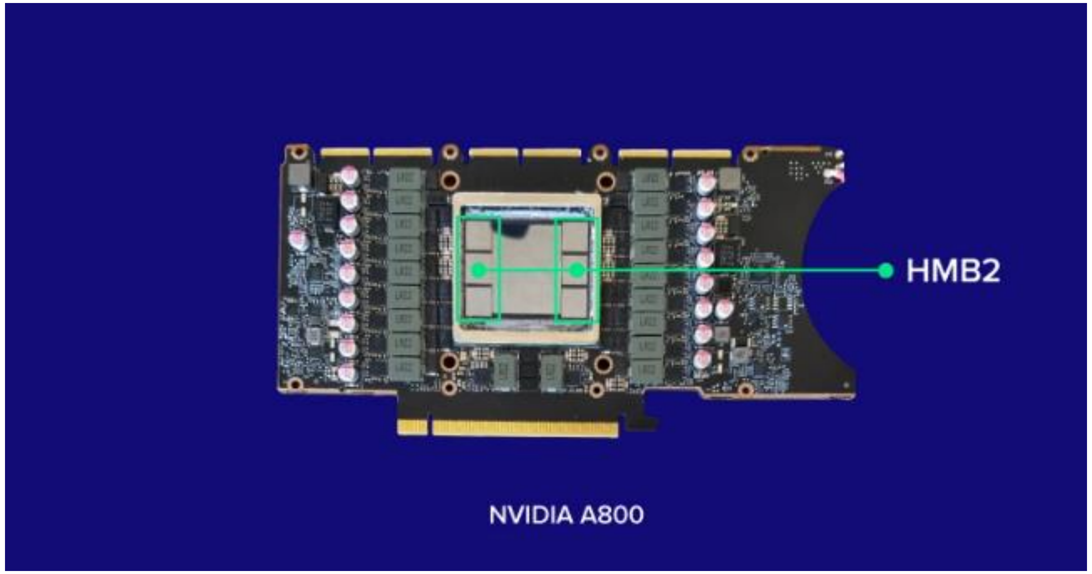
High Bandwidth Memory

Main advantages:

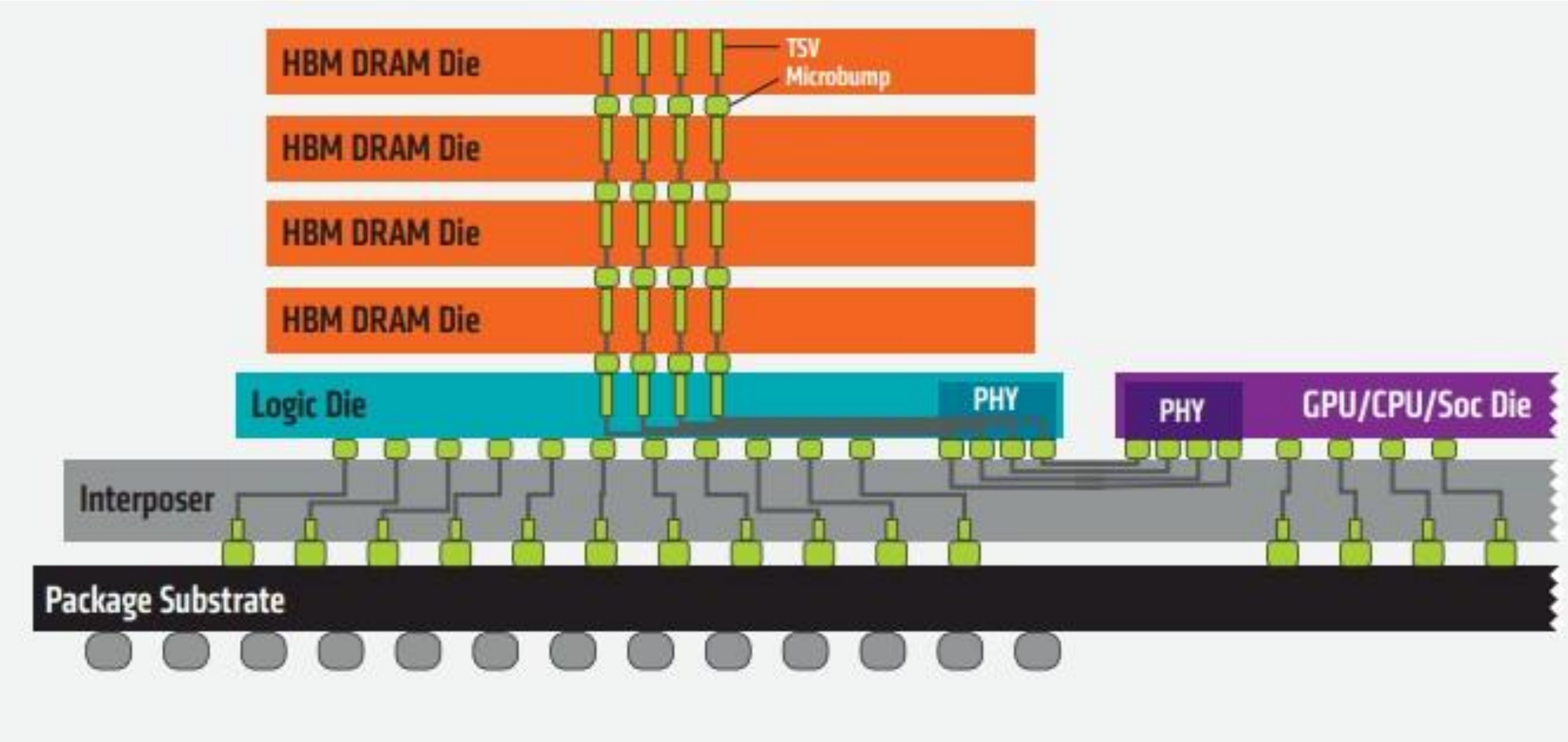
- Ideally in package with the PE
- On an interposer made for the HBM

Main Disadvantages:

- Thermal
- Cost
- Technical Overhead



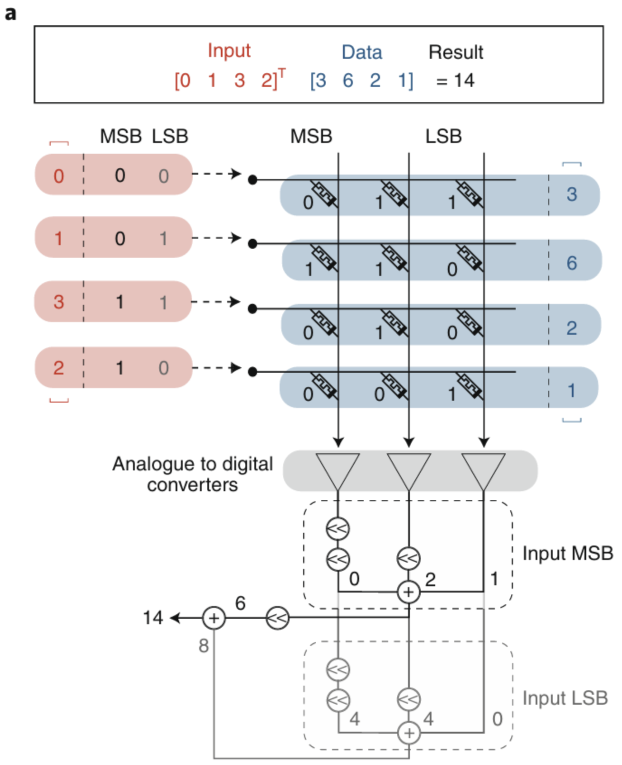
HBM Architecture



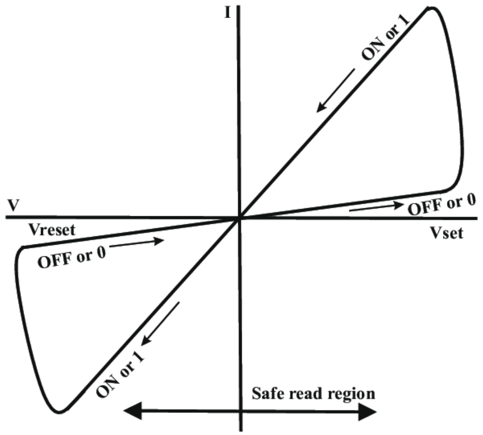
HBM vs GDDR Performance

GPU	Memory Type	Memory Bus Width	Memory Bandwidth
RTX 6000 Ada	GDDR6	384-bits	960 GB/s
GeForce RTX 4090	GDDR6X	384-bits	1008 GB/s (1 TB/s)
NVIDIA L40S	GDDR6	384-bits	864 GB/s
NVIDIA A800 40GB Active	HBM2	5120-bits	1555 GB/s (1.5 TB/s)
NVIDIA H100 80GB PCIe	HBM2e	5120-bits	2039 GB/s (2 TB/s)
NVIDIA H100 80G SXM5	HBM3	5120-bits	3350 GB/s (3.35 TB/s)

In-Memory Computing Solutions



What if we made the memory and compute, the same thing?

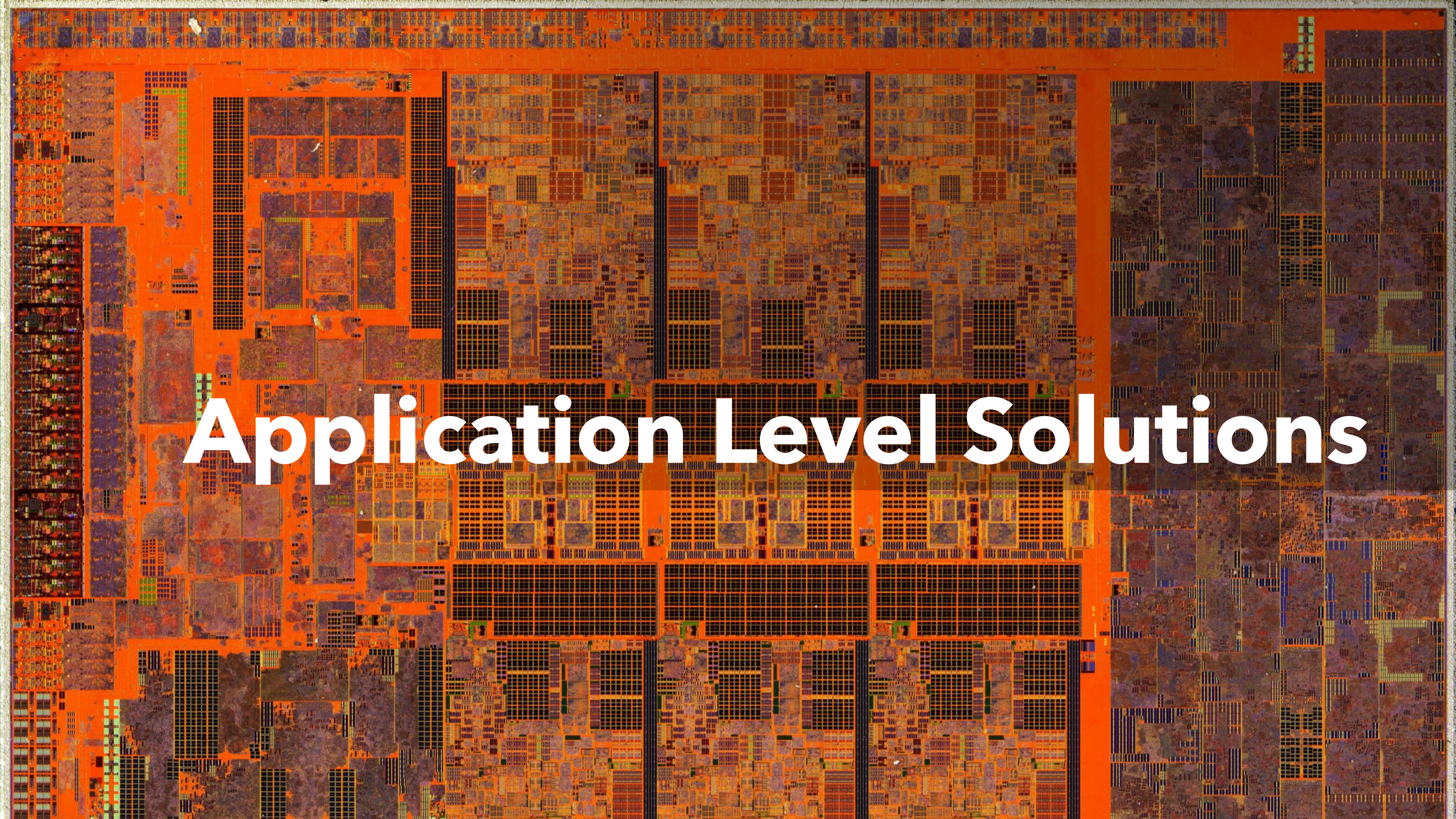


Challenges

- Limited Compatibility
- Variability in Fabrication
- Limited Endurance
- Sneak Path Current
- Noise Issues

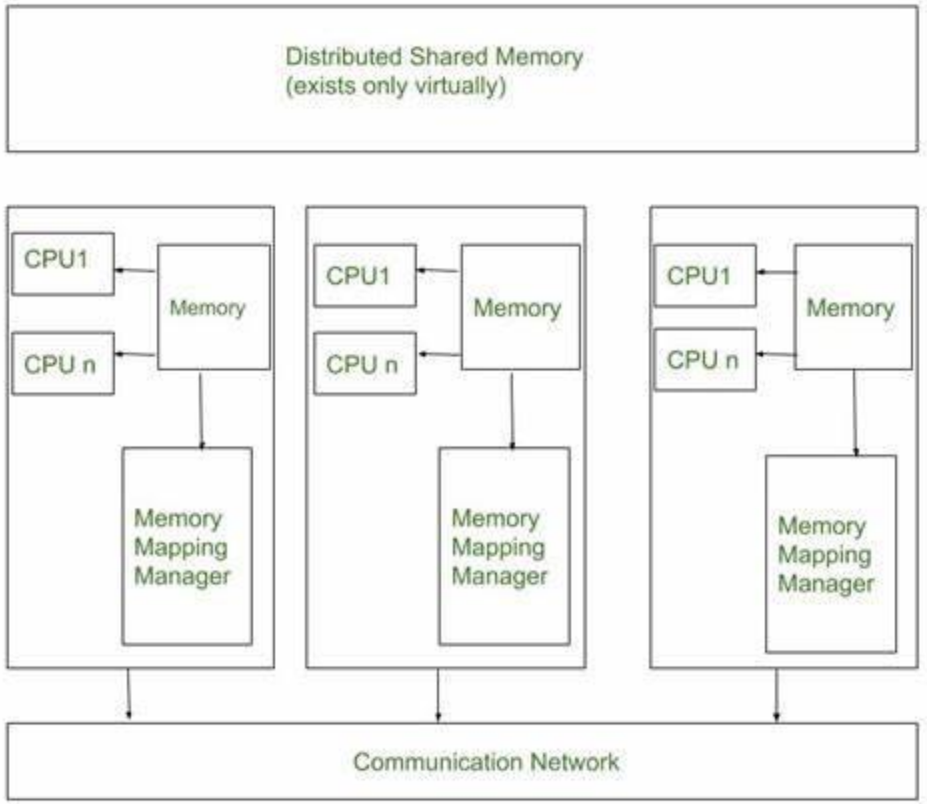
Too many unsolved challenges to make this a viable solution.

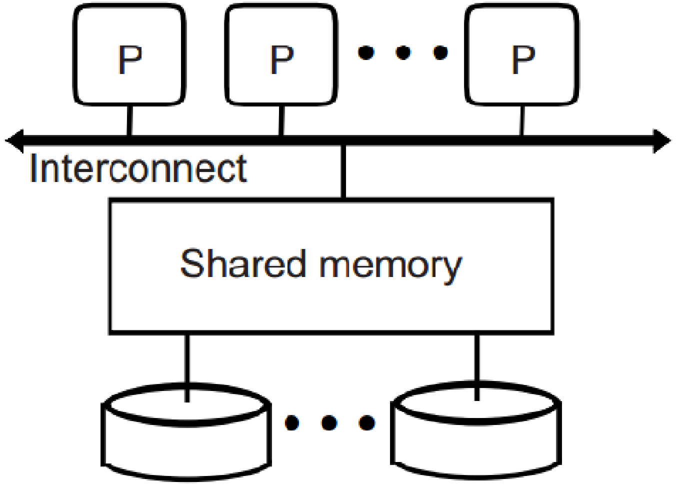
But it’s cool



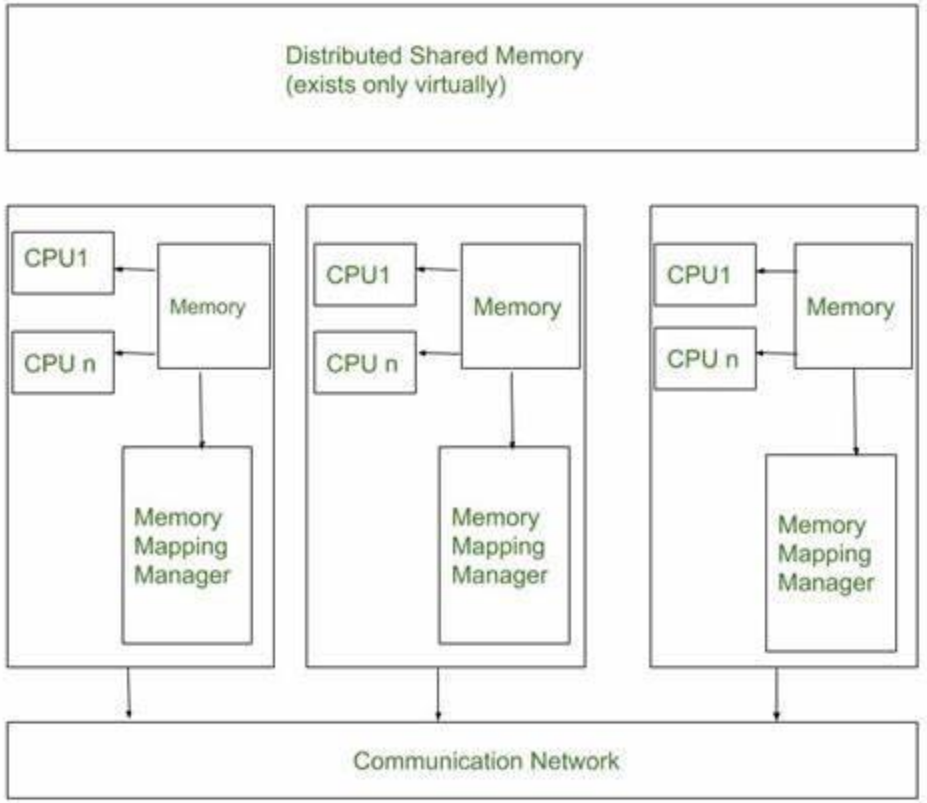
Application Level Solutions

Memory Architectures at HPC





Main Issue with Distributed Memory



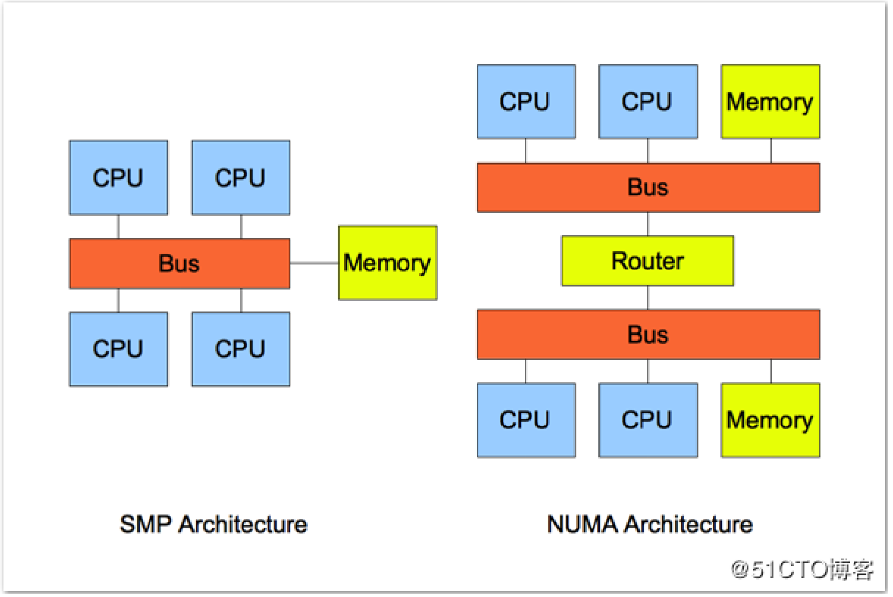
More complexity in application development

Memory has to be explicitly sent between processors

NUMA Aware Data Locality

Shared address space

Data Locality is now considered



That is all! Join Us!

We will now do some literature review if time allows