CS 250B: Modern Computer Systems

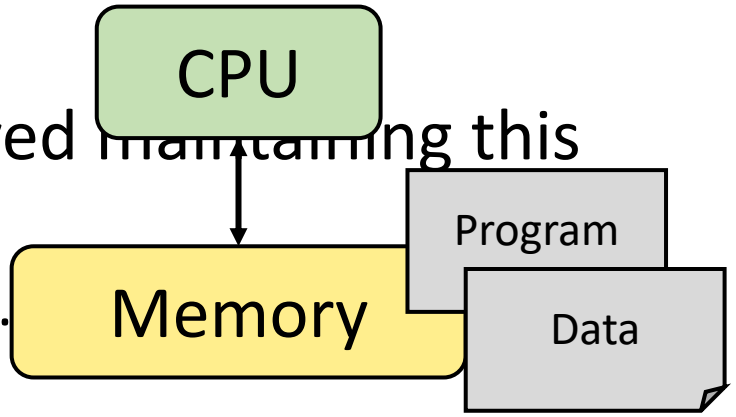
The End of Conventional Performance Scaling

Sang-Woo Jun

UCI

Conventional Performance Scaling

- Traditional model of a computer is simple
  - Single, in-order flow of instructions on a processor
  - Simple, in-order memory model
- Large part of computer architecture research involved maintaining this abstraction while improving performance
  - Transparent caches, Transparent superscalar scheduling, ...
  - Same software runs faster tomorrow
  - (Slow software becomes acceptable tomorrow)
- Driven largely by continuing march of Moore's law



Moore’s Law

- What exactly does it mean?
- What is it that is scaling?

Moore's Law

- Typically cast as:
  "Performance doubles every X months"

- Actually closer to:
  "Number of transistors per unit cost doubles every two years"

Moore's Law

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year.

[…]

Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years.

-- Gordon Moore, Electronics, 1965

Why is Moore's Law conflated with processor performance?

Dennard Scaling: Moore's Law to Performance

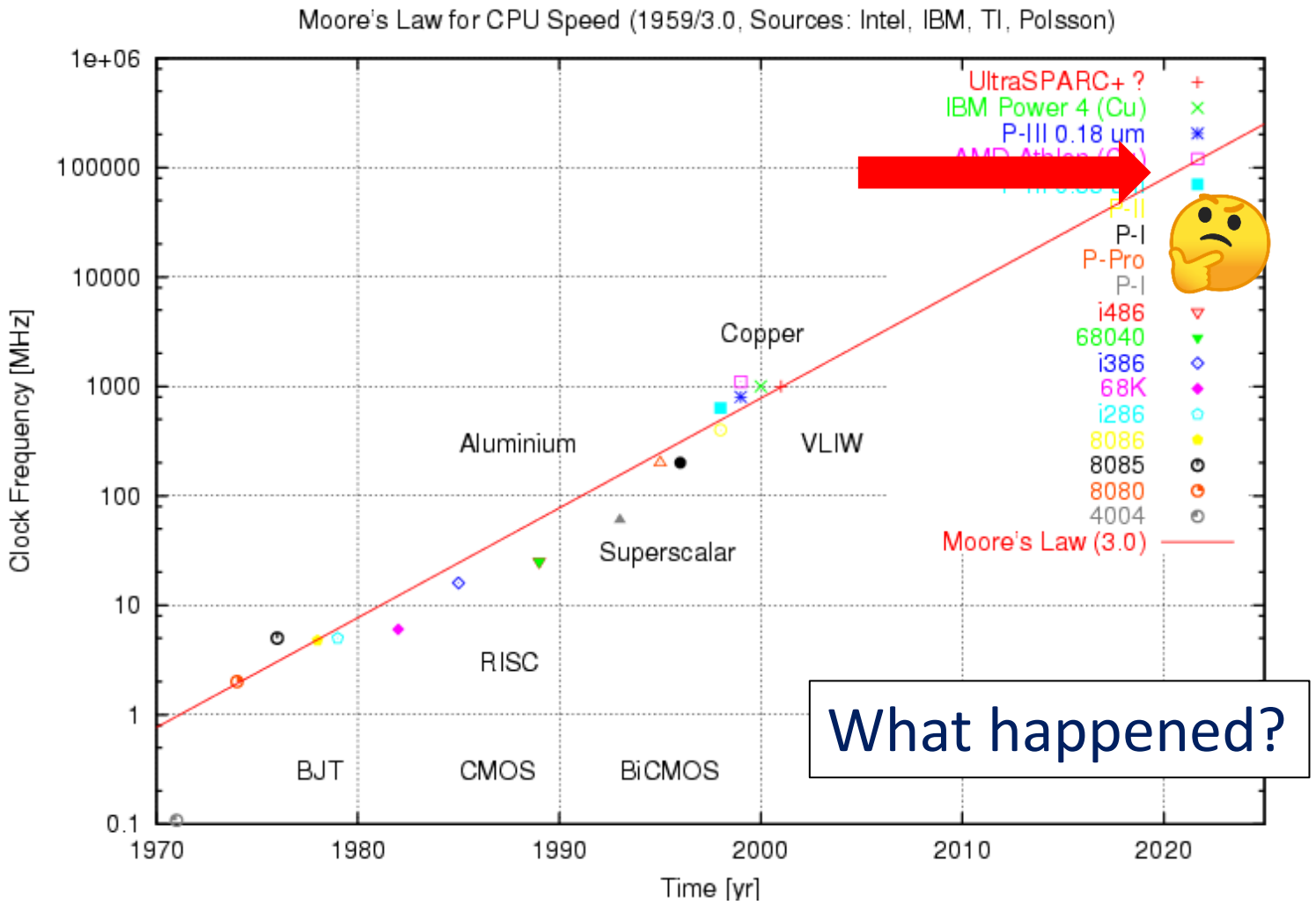
- "Power density stays constant as transistors get smaller"
  - Robert H. Dennard, 1974

- Intuitively:
  - Smaller transistors → shorter propagation delay → faster frequency
  - Smaller transistors → smaller capacitance → lower voltage

  - $Power \propto Capacitance \times Voltage^2 \times Frequency$

Moore's law → Faster performance @ Constant power!

Single-Core Performance Scaling Projection



(Slightly) More Accurate Processor Power Consumption

Gate-oxide stopped scaling

Stopped scaling due to leakage

$$Power = \underbrace{(ActiveTransistors \times Capacitance \times Voltage^2 \times Frequency)}_{\text{Dynamic power}}$$

$$\underbrace{+ (Voltage \times Leakage)}_{\text{Static power}}$$

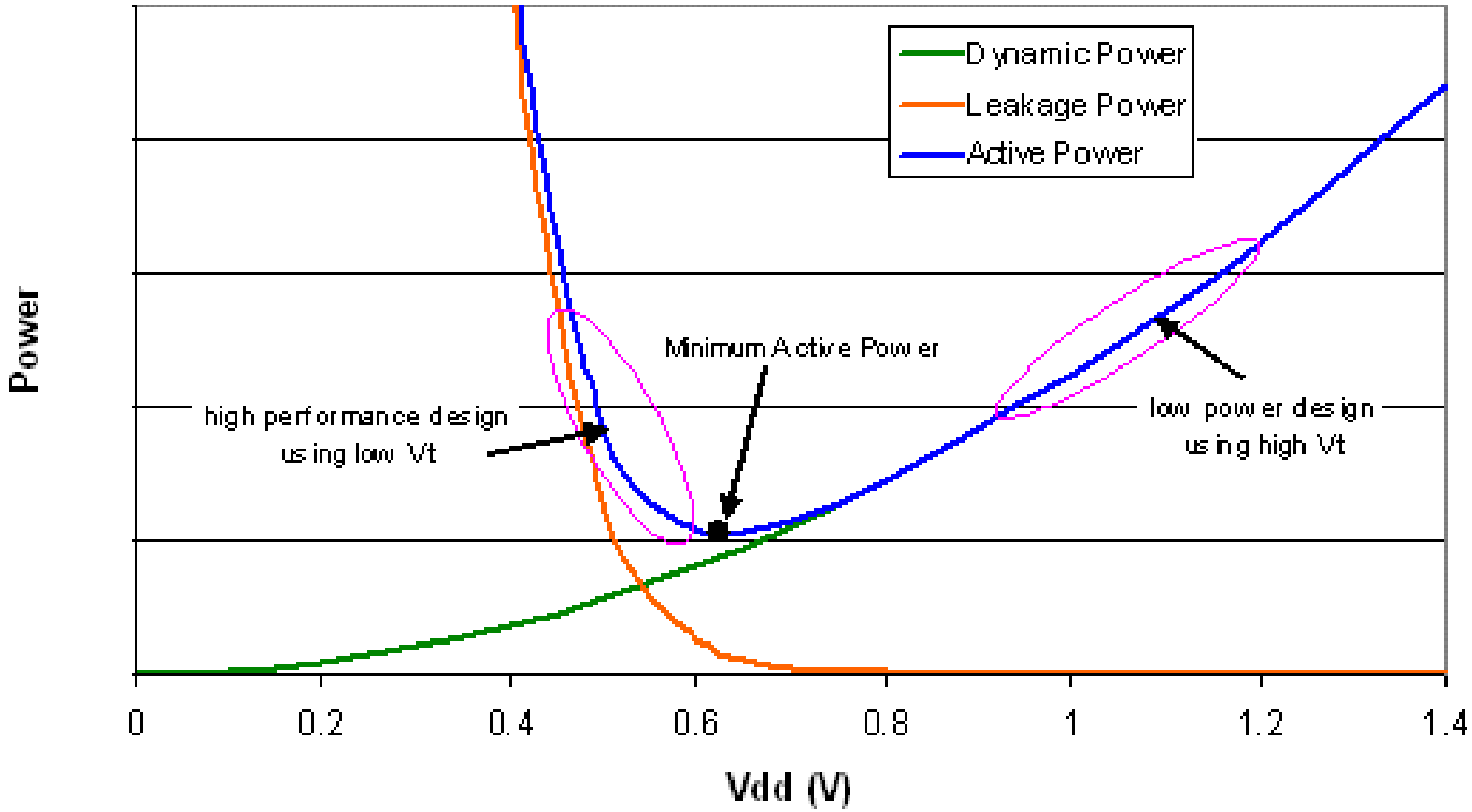
Unfortunately…

$$Leakage \propto \frac{1}{e^{Voltage}}$$

EXTREMELY simplified model!

Power Consumption of High-Density Circuits

- Total power consumption with constant frequency



https://www.design-reuse.com/articles/20296/power-management-leakage-control-process-compensation.html

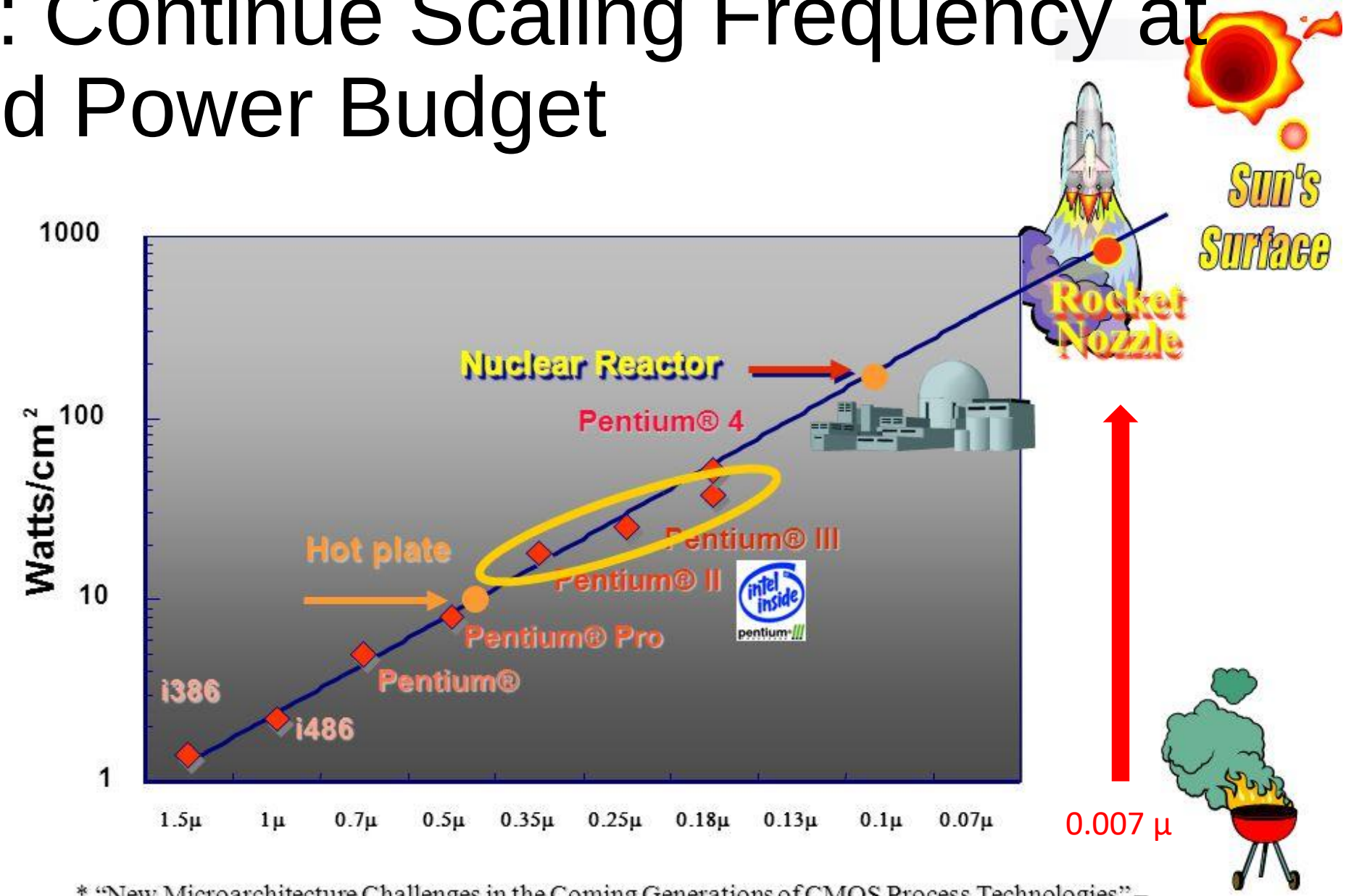
End of Dennard Scaling

- Even with smaller transistors, we cannot continue reducing power
  - What do we do now?

- Option 1: Continue scaling frequency at increased power budget
  - Chip quickly become too hot to cool!
  - Thermal runaway:

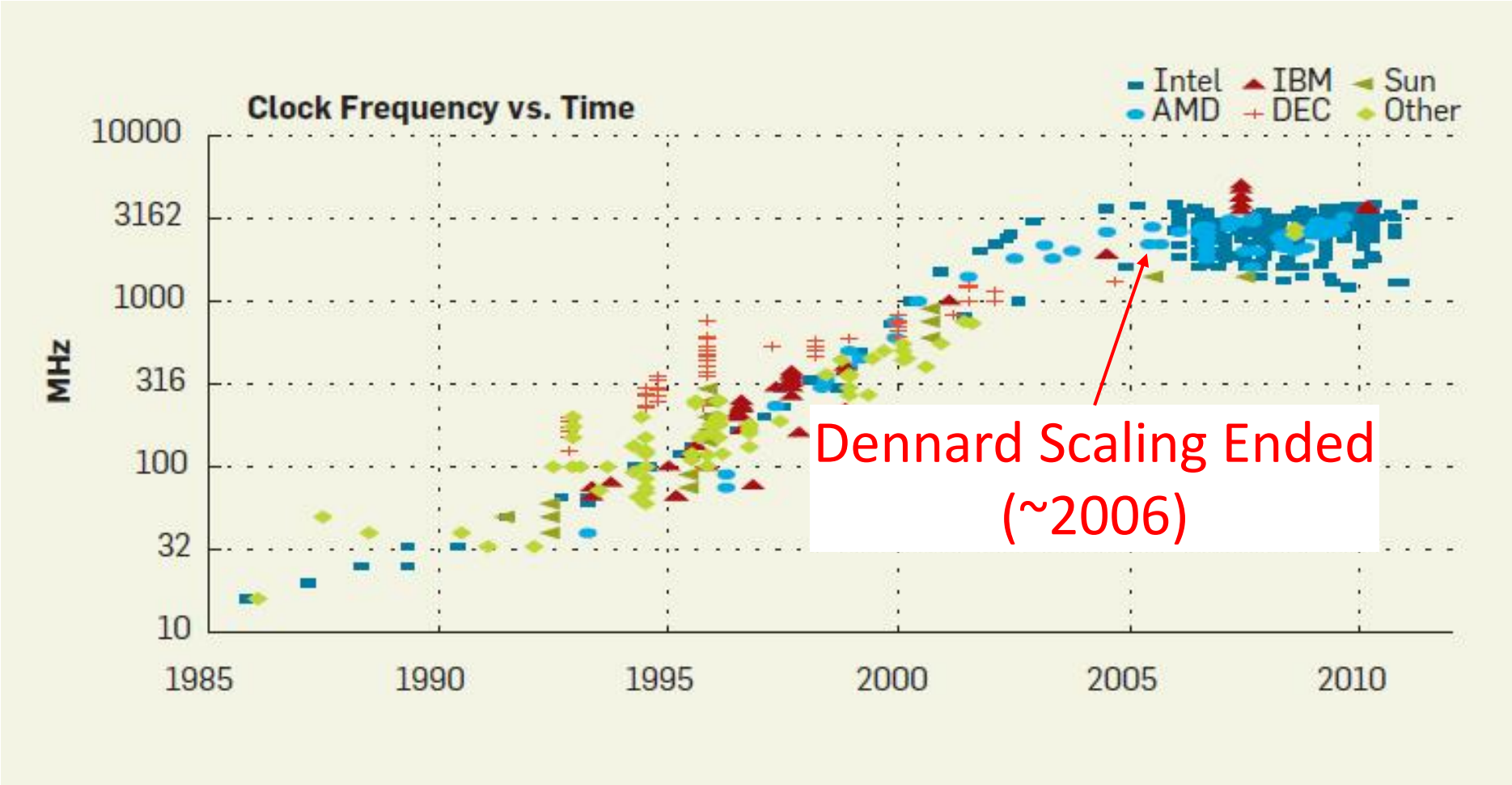
    Hotter chip → increased resistance → hotter chip → …

Option 1: Continue Scaling Frequency at Increased Power Budget



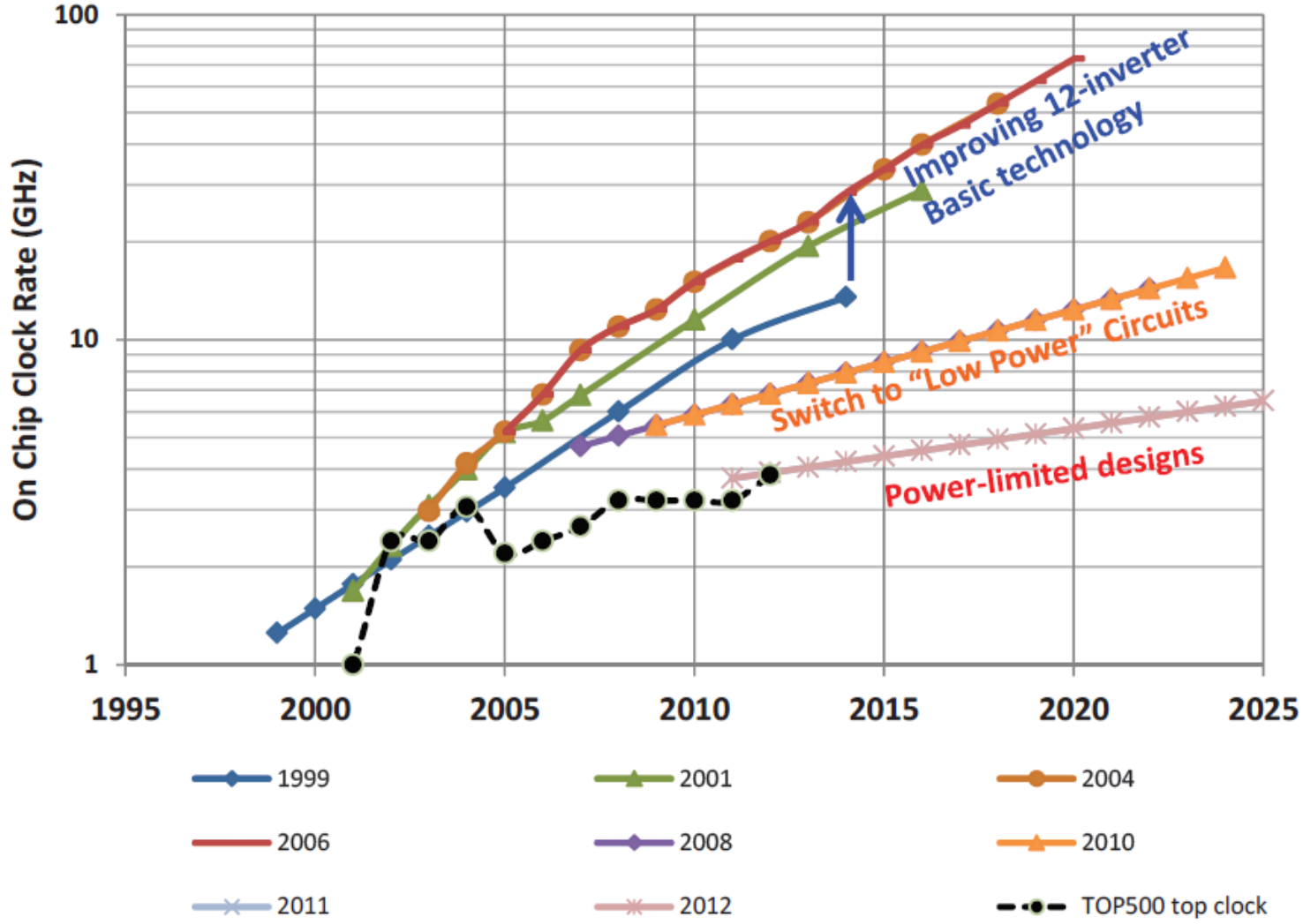
* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Option 2: Stop Frequency Scaling



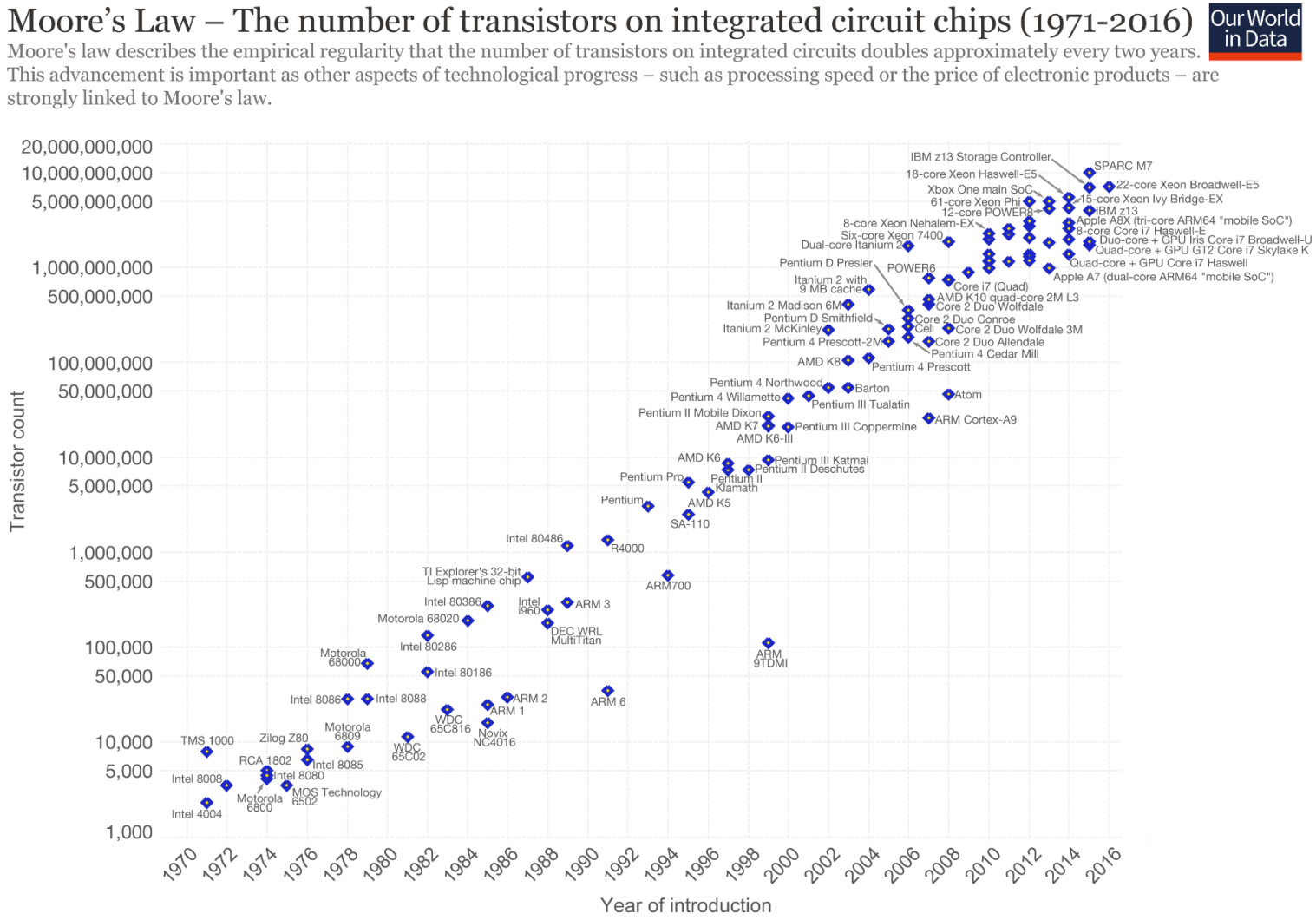
Danowitz et.al., “CPU DB: Recording Microprocessor History,” Communications of the ACM, 2012

Looking Back: Change of Predictions



Kogge et. al., "Yearly update : exascale projections for 2013,"Sandia National Laboratoris, 2013

But Moore’s Law Continues Beyond 2006

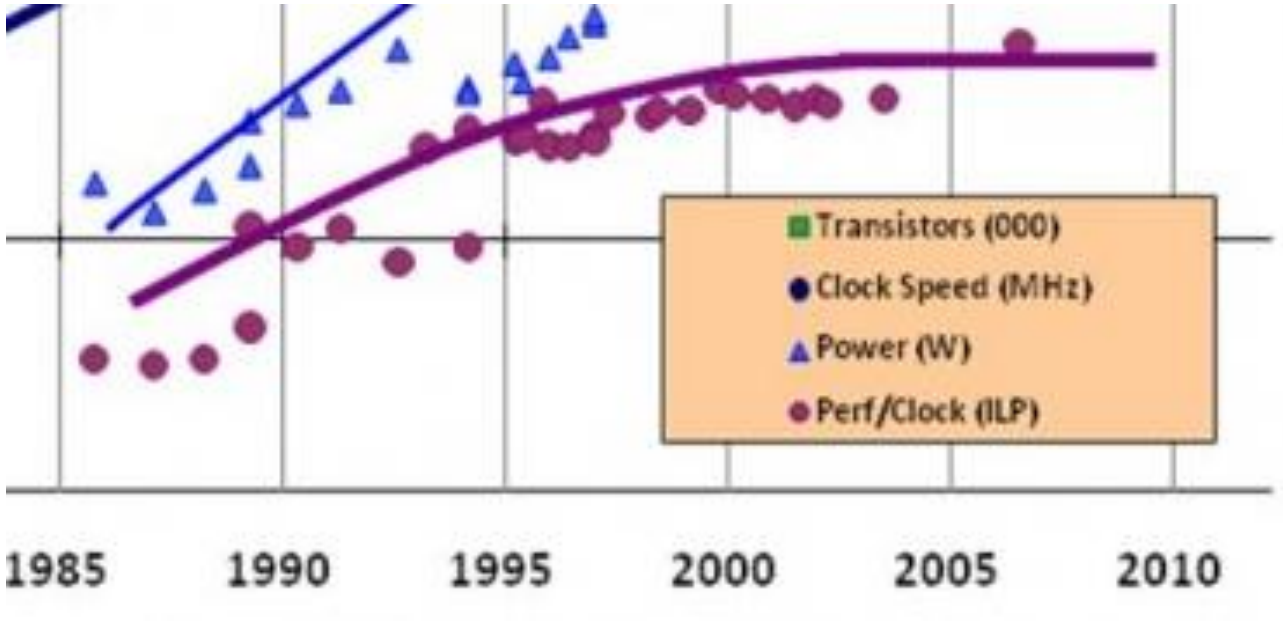


Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

State of Things at This Point (2006)

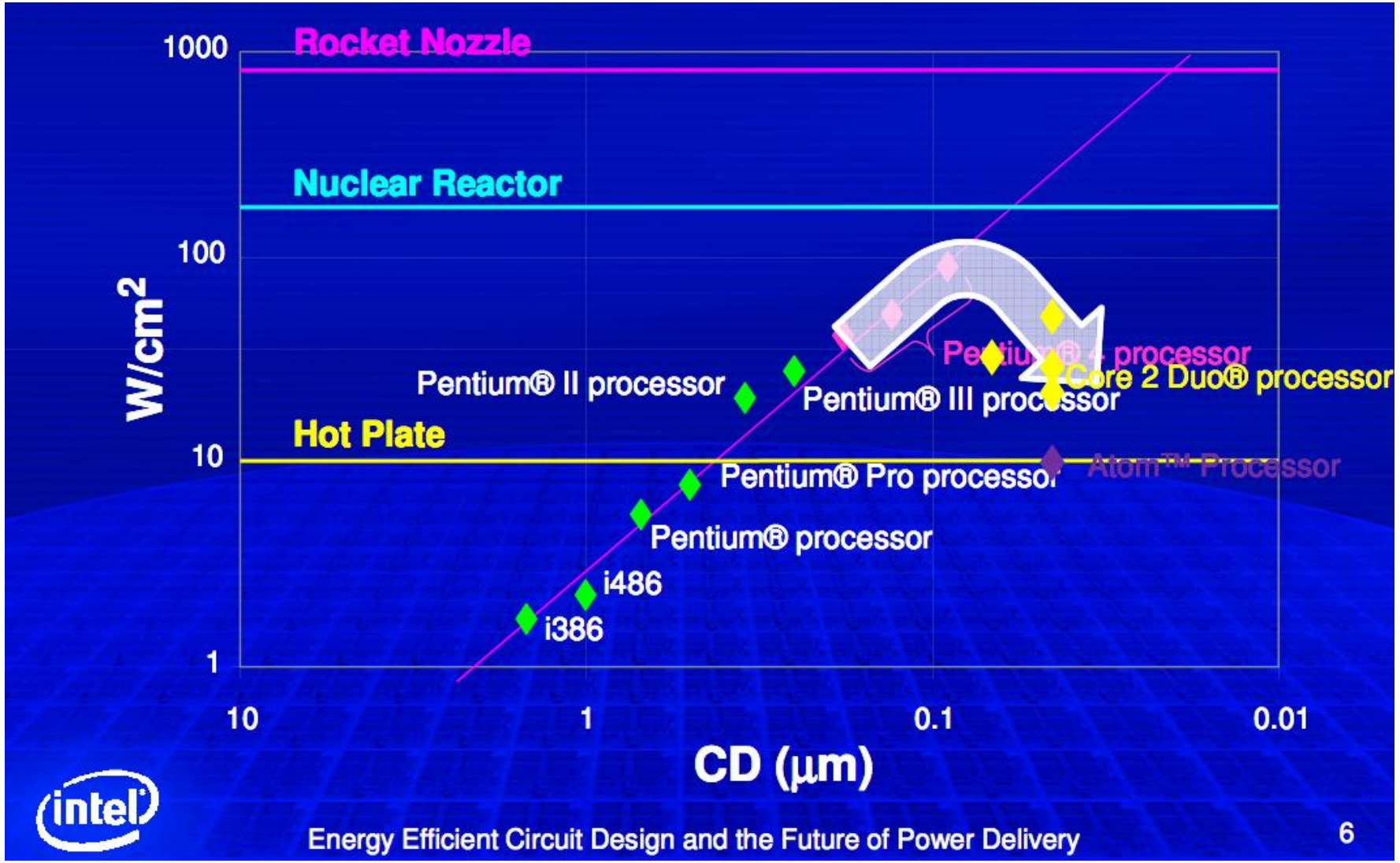
- Single-thread performance scaling ended
  - Frequency scaling ended (Dennard Scaling)
  - Instruction-level parallelism scaling stalled … also around 2005
- Moore's law continues
  - Double transistors every two years
  - What do we do with them?

Instruction Level Parallelism

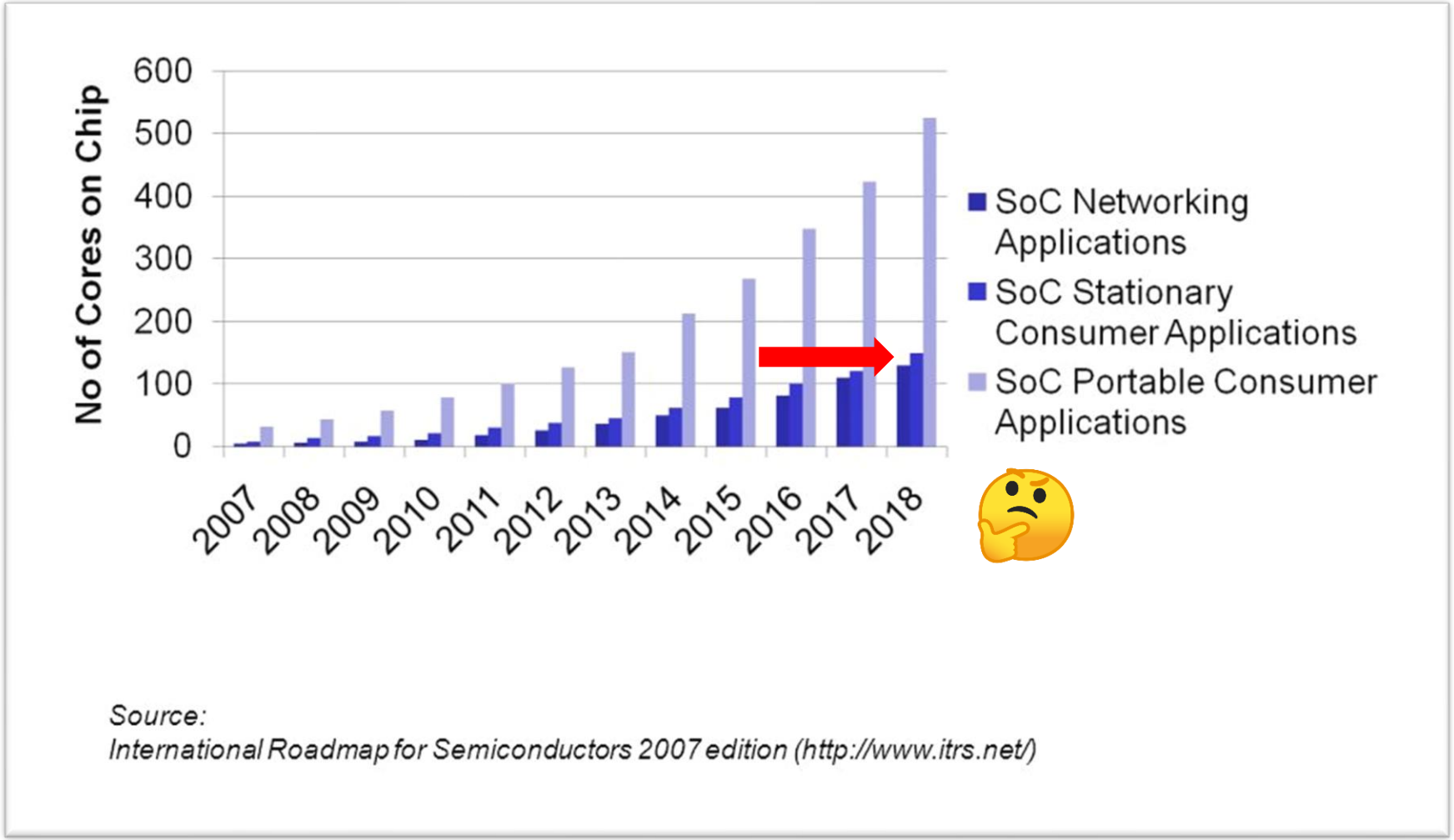


K. Olukotun, "Intel CPU Trends"

Crisis Averted With Manycores?



Crisis Averted With Manycores?



What Happened?

Can't keep going up

Gate-oxide stopped scaling

Stopped scaling due to leakage

$$Power = (ActiveTransistors \times Capacitance \times Voltage^2 \times Frequency)$$

Dynamic power

Stopped scaling due to leakage

$$+ (Voltage \times LeakageCurrent)$$

Static power

"Utilization Wall"

Regardless of Moore's Law, a limited amount of gates can be active at a given time

Where To, From Here?

- The number of active transistors at a given time is limited
  - Left unchecked, we won't get much performance improvements even with Moore's law continuing
  - We need to make the best use of those active transistors!

Also, Scaling Size is Becoming More Difficult!

- Processor fabrication technology has always reduced in size
  - As of 2022, 5 nm is cutting edge, working towards 3 nm

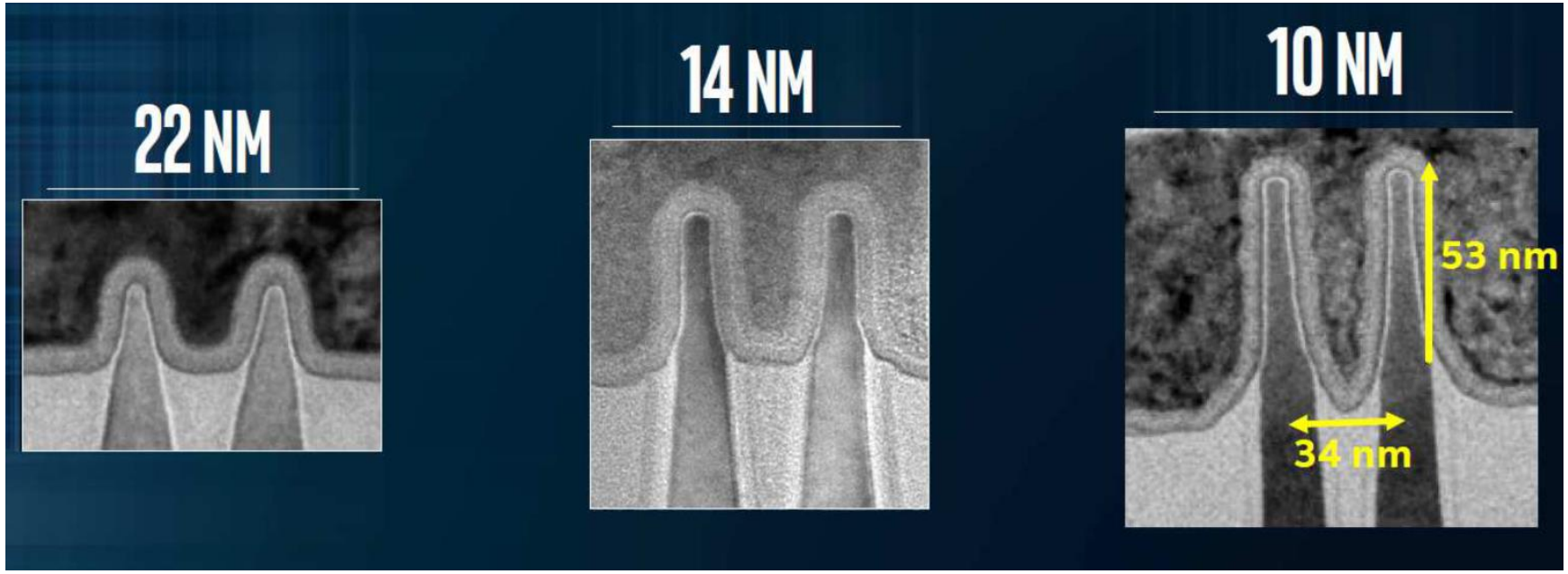
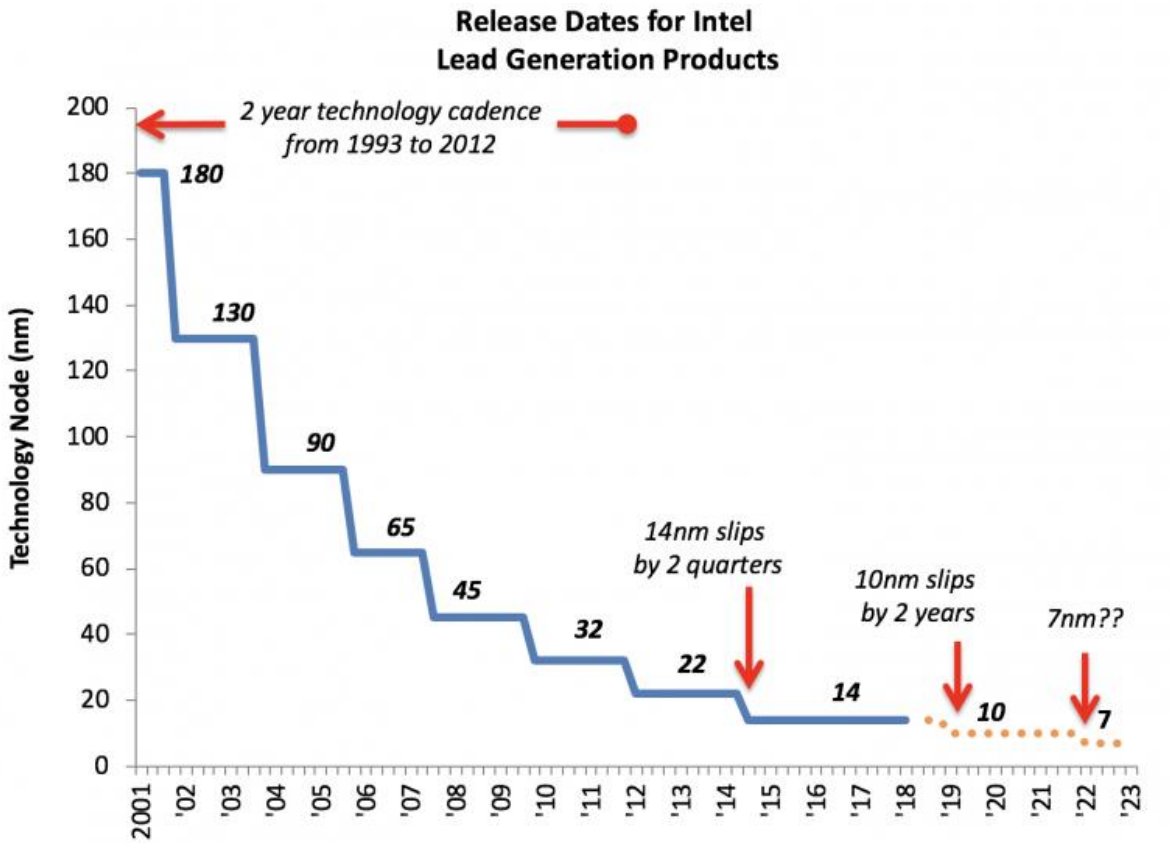
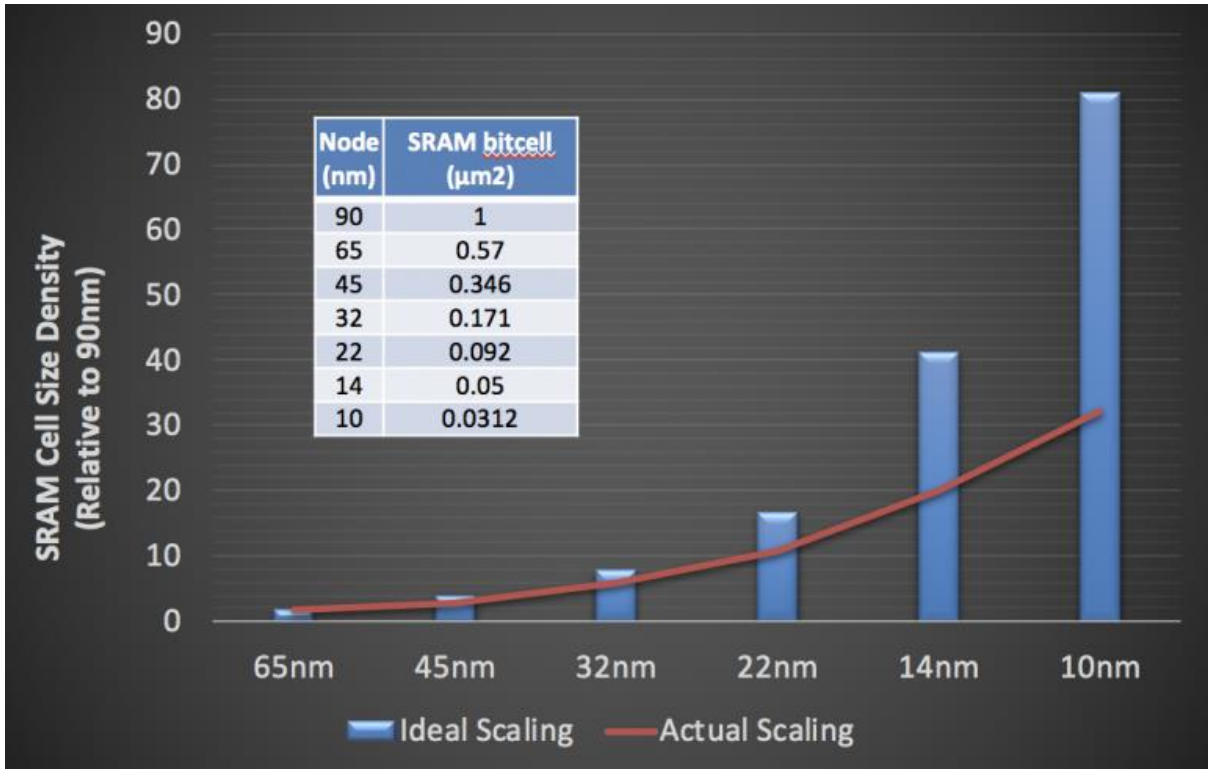


Image source: Intel

Forecast Not Good For Scaling…





Node (nm)	SRAM bitcell (µm2)
90	1
65	0.57
45	0.346
32	0.171
22	0.092
14	0.05
10	0.0312

Less transistors for processors, less bits for memory

David Brooks, “What’s the future of technology scaling?,” Computer Architecture Today

Year 2000

Year 2008

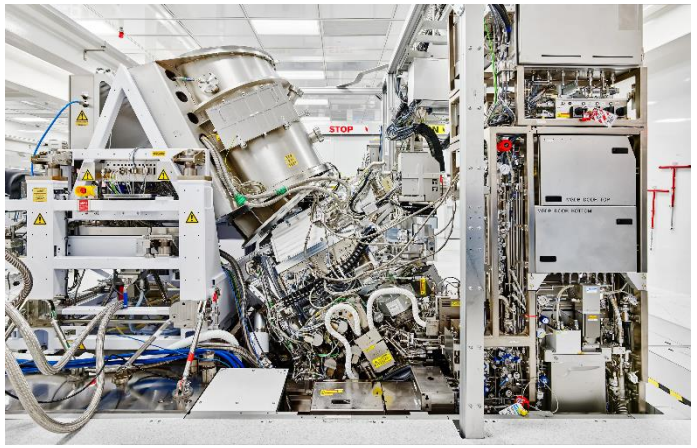
Year 2023

Number of Semiconductor Manufacturers with a Cutting Edge Logic Fab

SilTerra										
X-FAB										
Dongbu HiTek										
ADI	ADI									
Atmel	Atmel									
Rohm	Rohm									
Sanyo	Sanyo									
Mitsubishi	Mitsubishi									
ON	ON									
Hitachi	Hitachi									
Cypress	Cypress	Cypress								
Sony	Sony	Sony								
Infineon	Infineon	Infineon								
Sharp	Sharp	Sharp								
Freescale	Freescale	Freescale								
Renesas (NEC)	Renesas	Renesas	Renesas	Renesas						
Toshiba	Toshiba	Toshiba	Toshiba	Toshiba						
Fujitsu	Fujitsu	Fujitsu	Fujitsu	Fujitsu						
TI	TI	TI	TI	TI						
Panasonic	Panasonic	Panasonic	Panasonic	Panasonic	Panasonic					
STMicroelectronics	STM	STM	STM	STM	STM					
HLMC	HLMC		HLMC	HLMC	HLMC					
UMC	UMC	UMC	UMC	UMC	UMC		UMC			
IBM	IBM	IBM	IBM	IBM	IBM	IBM				
SMIC	SMIC	SMIC	SMIC	SMIC	SMIC		SMIC			
AMD	AMD	AMD	GlobalFoundries	GF	GF	GF	GF			
Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung	Samsung
TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC	TSMC
Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel	Intel
180 nm	130 nm	90 nm	65 nm	45 nm/40 nm	32 nm/28 nm	22 nm/20 nm	16 nm/14 nm	10 nm	7 nm	5 nm

Image source: WikiChip

Not going into details:
EUV lithography (@ ASML) at the cutting edge



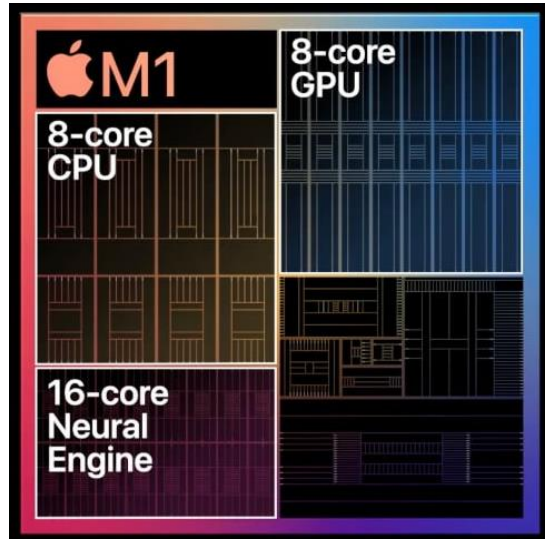
https://www.technologyreview.com/2021/10/27/1037118/moores-law-computer-chips/

Only three players left?!

We Can’t Keep Doing What we Used to

- Limited number of transistors, limited clock speed
  - How to make the ABSOLUTE BEST of these resources?

- Timely example: Apple M1 Processor
  - Claims to outperform everyone (per Apple)
  - How?
    - “8-wide decoder” […] “16 execution units (per core)”
    - “(Estimated) 630-deep out-of-order”
    - “Unified memory architecture”
    - Hardware/software optimized for each other



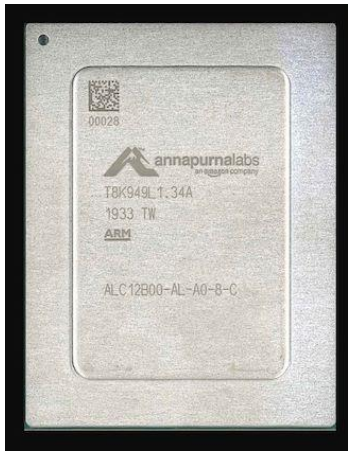
What do these mean?

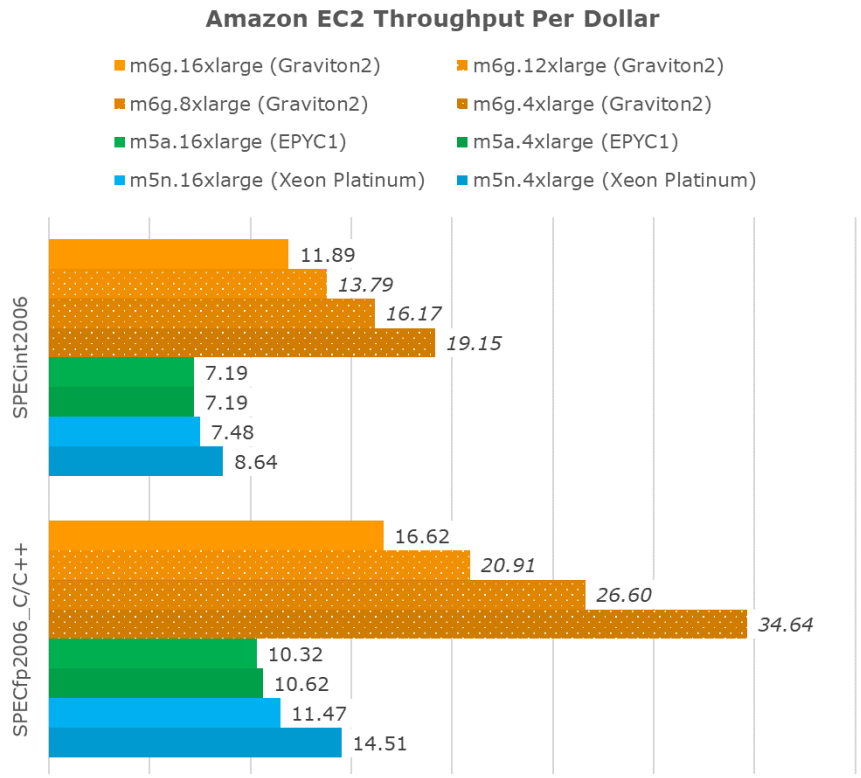
Not just apple! (Amazon, Microsoft, EU, …)

Image source: Apple

We can’t keep doing what we used to

AWS Graviton 2:
64-Core ARM





European Processor Accelerator (EPAC):
4-Core RISC-V +
Variable Precision Accelerator +
Stencil and Tensor Accelerator

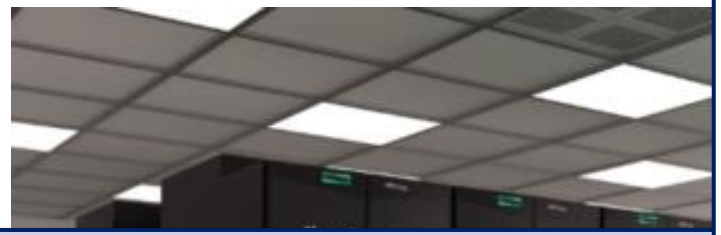
Sunway TaihuLight
Manycore custom RISC +
SIMD, Vector
Non-coherent scratchpad

Fujitsu A64FX (Fugaku)
ARM Variant
SIMD, Vector

Image source: Anandtech, “Amazon's Arm-based Graviton2 Against AMD and Intel: Comparing Cloud Compute”
Image source: TheNextPlatform, “Europe Inches Closer to Native RISC-V Reality”

The State of C

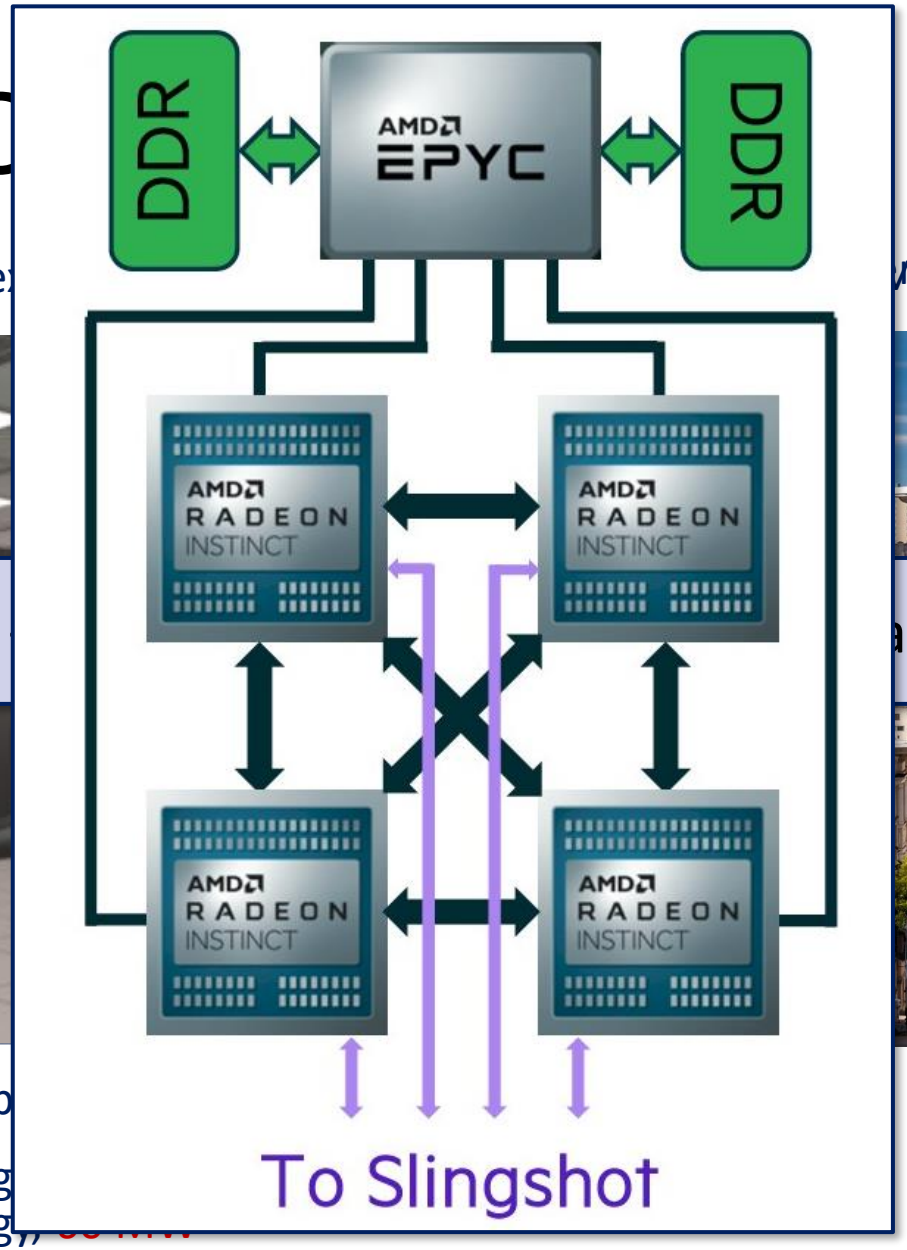
Department of Energy requested e



Heavy use of GPUs

1,000,000,000,000,000,000 floating p

Using 2016 technolog
Using 2019 technolog
Using 2022 technology, 20 MW



ntial power consumption of San Francisco

nTI Research nuclear reactor

nded programming

168 MW

Image: TheNextPlatform
(Calculated from "Electricity Consumption by County", California energy commission)

Lynn Freeny, Department of Energy

Where To, From Here?

- Potential Solution 1: The software solution
  - Write efficient software to make the efficient use of hardware resources
  - No longer depend entirely on hardware performance scaling
  - “Performance engineering” software, using hardware knowledge

Impact of Software Performance Engineering

- Multiplying two 2048 x 2048 matrices
  - 16 MiB, doesn't fit in smaller caches
- Machine: Intel i5-7400 @ 3.00GHz

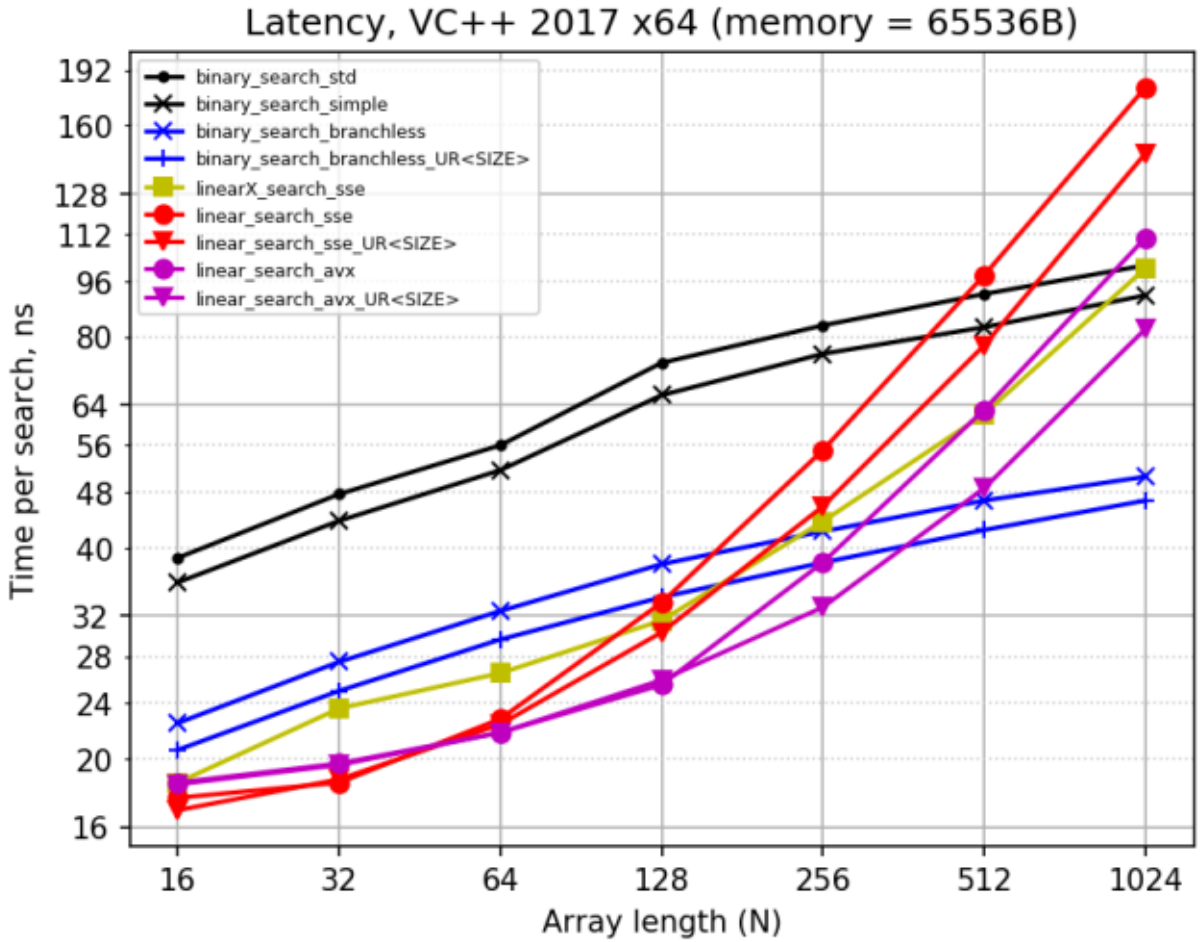
63.19 seconds

10.39 seconds
(6x performance!)

Last year, we measured 42.13x performance improvement just by writing better software

Computer architecture effects example 2

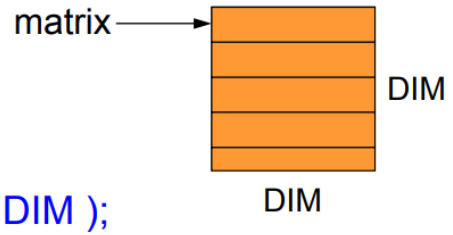
- Binary search vs. branchless binary search vs. linear search
  - Where does this difference come from, and how do I exploit this?
  - Architecture, assembly knowledge!

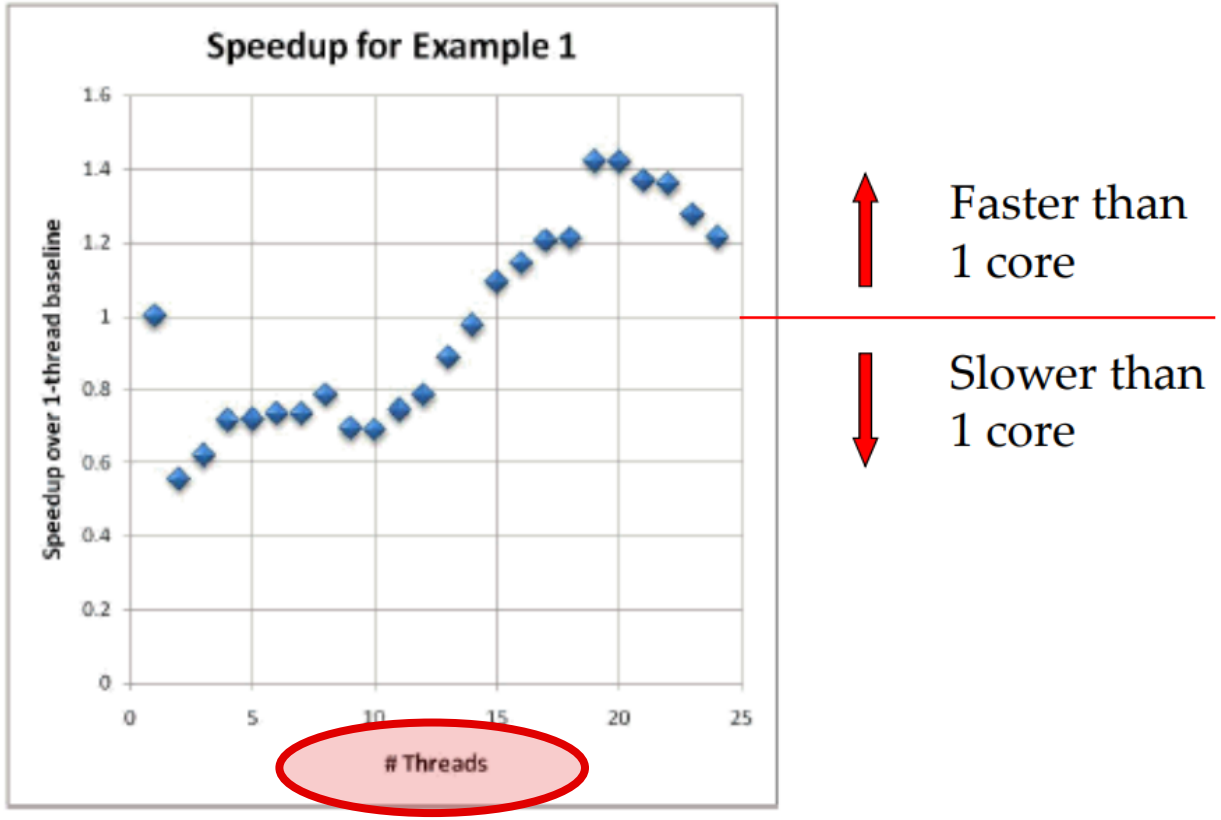


Source: “Performance comparison: linear search vs binary search.” dirtyhandscoding.wordpress.com

Computer architecture effects example 3

```
int result[P];
// Each of P parallel workers processes 1/P-th of the data;
// the p-th worker records its partial count in result[p]
for (int p = 0; p < P; ++p )
   pool.run( [&,p] {
      result[p] = 0;
      int chunkSize = DIM/P + 1;
      int myStart = p * chunkSize;
      int myEnd = min( myStart+chunkSize, DIM );
      for( int i = myStart; i < myEnd; ++i )
         for( int j = 0; j < DIM; ++j )
            if( matrix[i*DIM+j] % 2 != 0 )
               ++result[p]; } );
pool.join();                          // Wait for all tasks to complete
odds = 0;                             // combine the results
for( int p = 0; p < P; ++p )
   odds += result[p];
```





REALLY BAD scalability! Why?

Source: Scott Meyers, “CPU Caches and Why You care”

Computer architecture effects example 4

```
for (target in stream):
    entities[target].string.append(char);
```

When entities.size < (1<<16): 1 GB/s

When entities.size > (1<<20): 200 MB/s

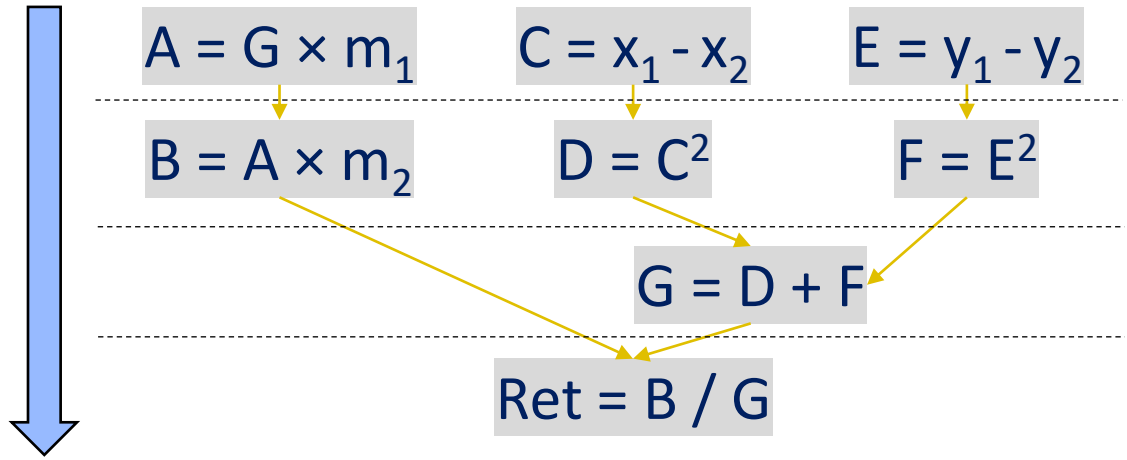
Why??

Where To, From Here?

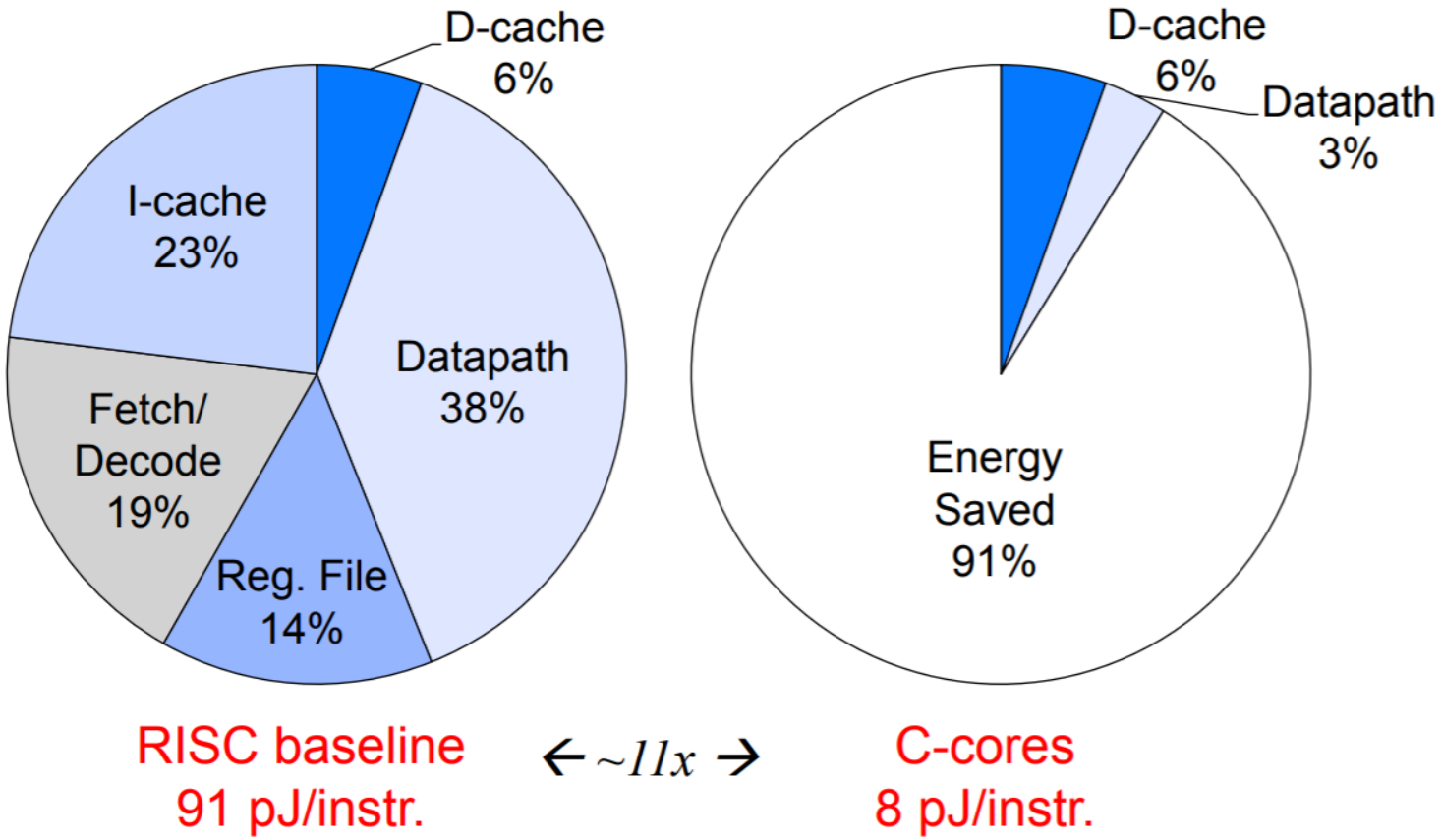
- Solution 2: The specialized architectural solution
  - Chip space is now cheap, but power is expensive
  - Stop depending on more complex general-purpose cores
  - Use space to build heterogeneous systems, with compute engines well-suited for each application

Fine-Grained Parallelism of Special-Purpose Circuits

- Example -- Calculating gravitational force: $\frac{G \times m_1 \times m_2}{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
- 8 instructions on a CPU, 16 instructions for two calculations, …
- Specialized datapath can be extremely efficient
  - Pipelined implementation can emit one result per cycle
  - Also, no need for general-purpose overhead such as instruction decoding
    - Much more cores can fit on chip
    - Much lower power consumption per unit

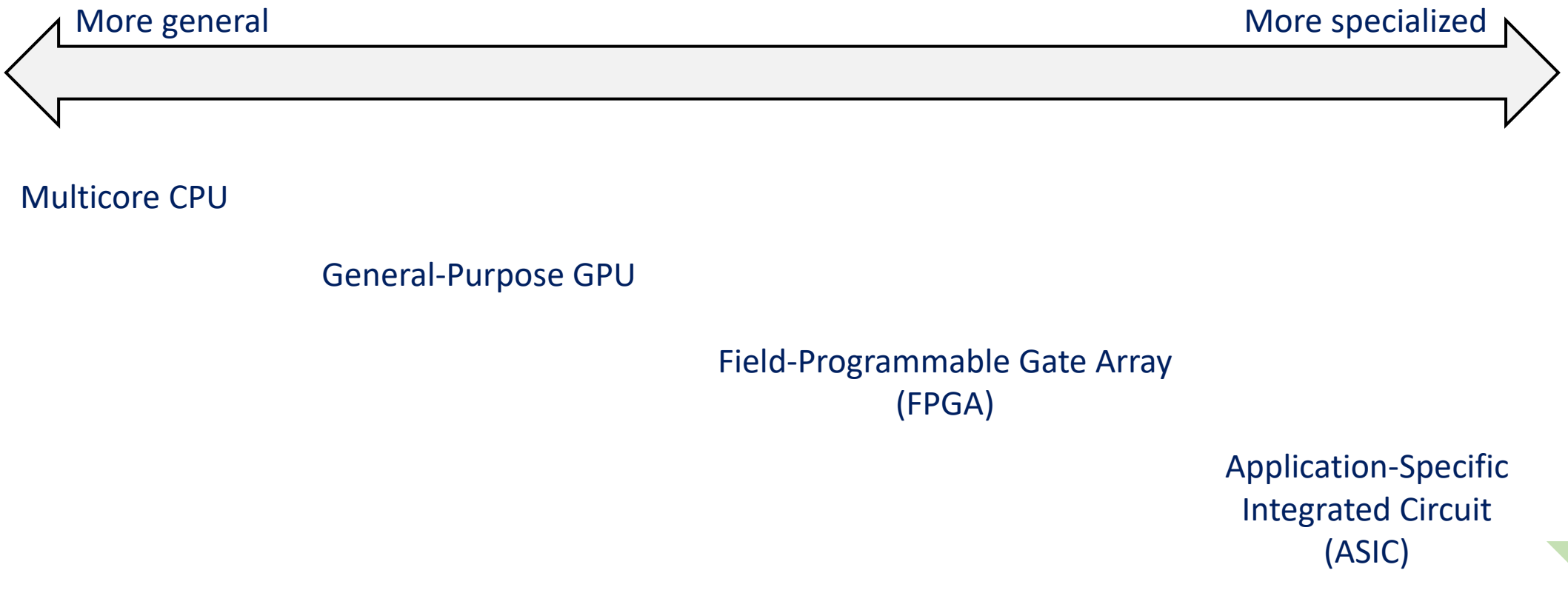


Typical Energy Efficiency Benefits of Optimized Hardware



Michael Taylor, “Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse,” 2012

Spectrum of Specialized Hardware



The Bottom Line: Architecture is No Longer Transparent

- Optimized software requires architecture knowledge
- Special-purpose “accelerators” (GPU, FPGA, …) programmed explicitly
- Even general-purpose processors implement specialized instructions
  - Single-Instruction Multiple Data (SIMD) instructions such as AVX
  - Special-purpose instructions sets such as AES-NI

Coming Up

- Before we go into newer technologies, let's first make sure we make good use of what we have
  - SIMD (SSE, AVX), Cache-optimized code, etc
  - "Performance engineering"
- "Our implementation delivers 9.2X the performance (RPS) and 2.8X the system energy efficiency (RPS/watt) of the best-published FPGA-based claims."
  - Li et. al., Intel, "Architecting to Achieve a Billion Requests Per Second Throughput on a Single Key-Value Store Server Platform," ISCA 2015
  - Intel software implementation of memcached