CS152: Computer Systems Architecture
Moore's Law

Gordon Moore (Photo 1978)
1929 – 2023

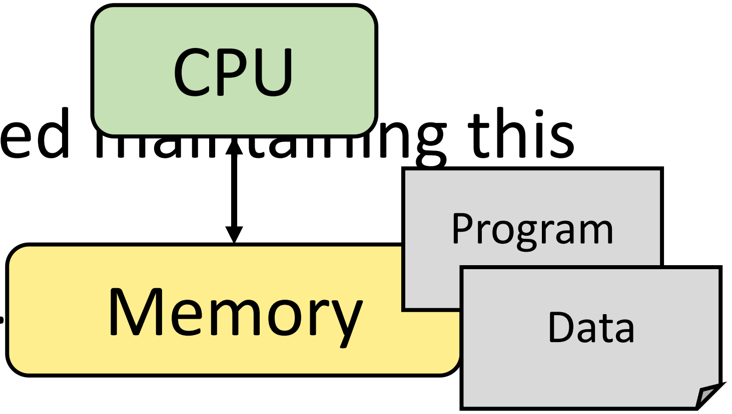
Sang-Woo Jun

Spring 2023

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 X months"

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?

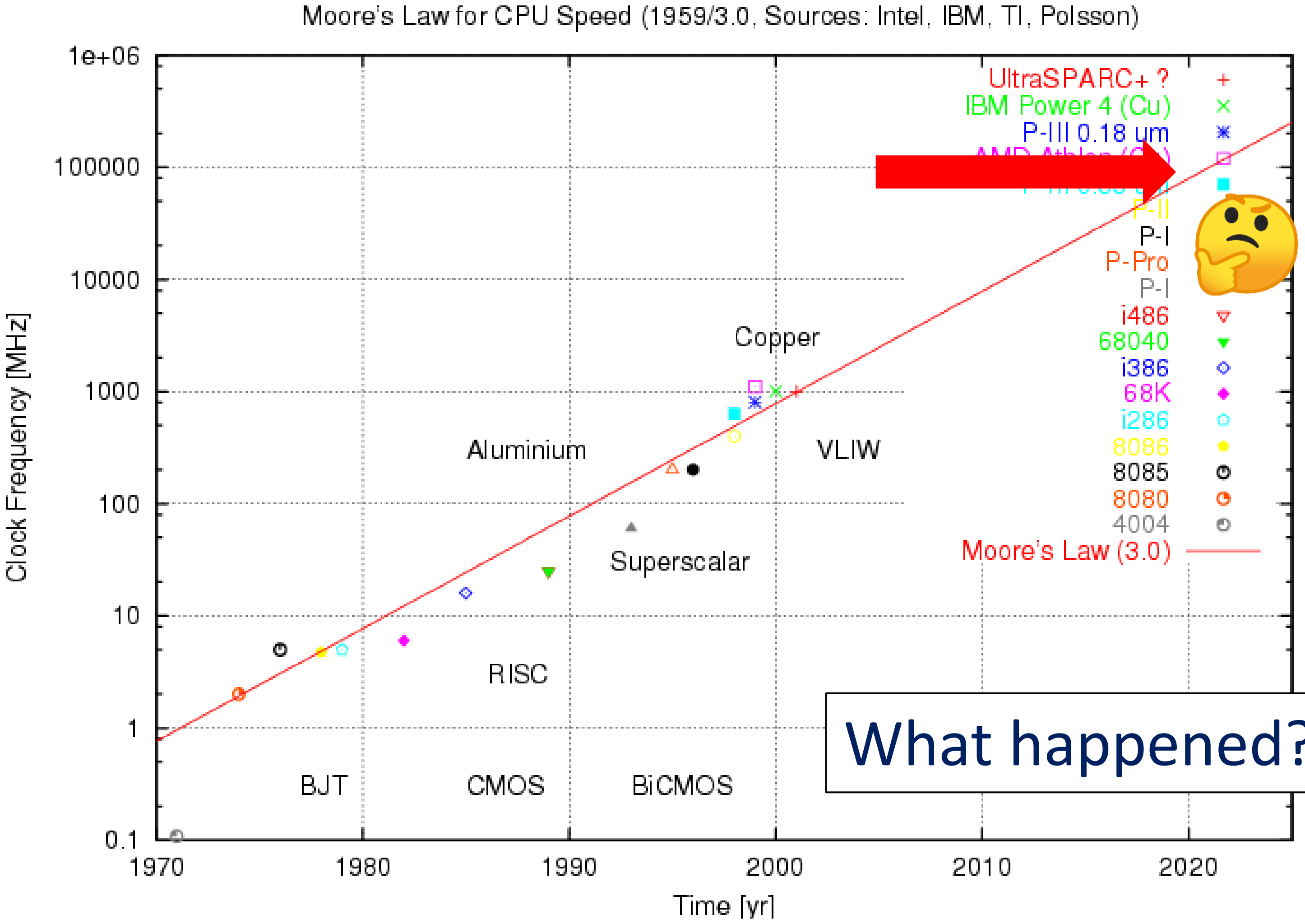
Dennard Scaling: From Moore's Law to performance

- "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



What happened?

(Slightly) more accurate processor power consumption

Gate-oxide stopped scaling

Stopped scaling due to leakage

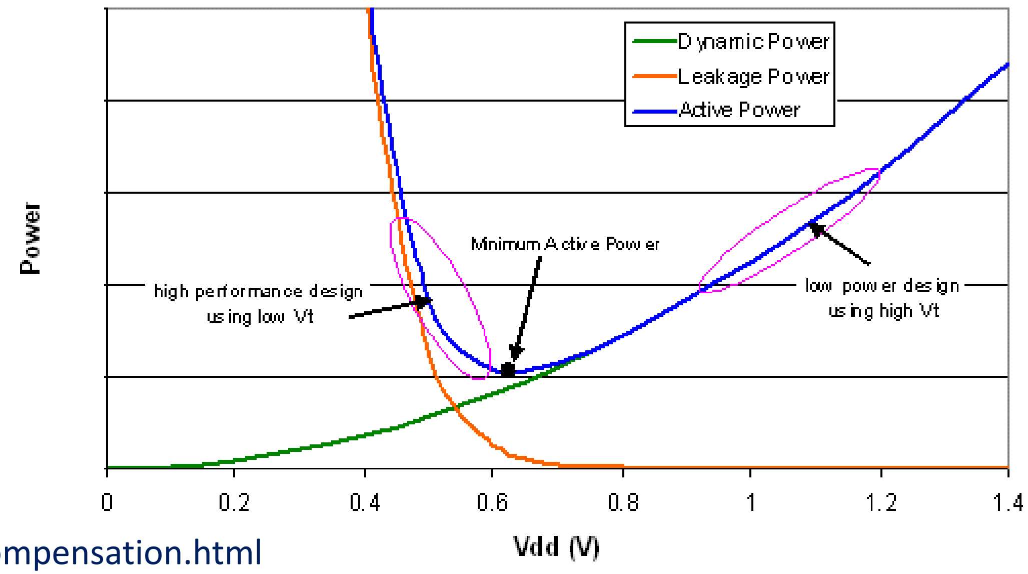
$$Power = \ (ActiveTransistors \times Capacitance \times Voltage^2 \times Frequency)$$

Dynamic power

$$+\ (Voltage \times Leakage)$$

Static power

Total power consumption with constant frequency



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

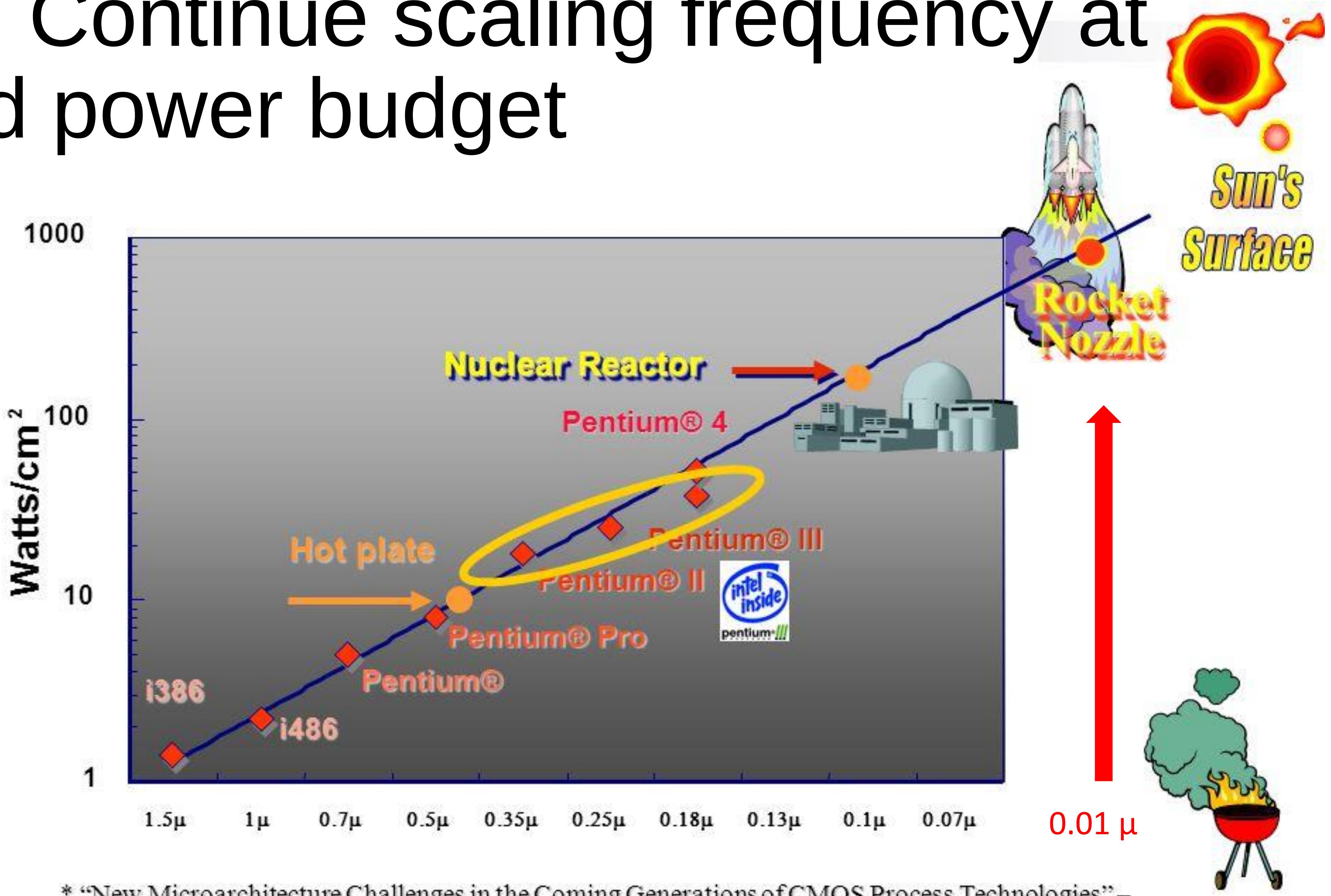
End of Dennard Scaling

- 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:

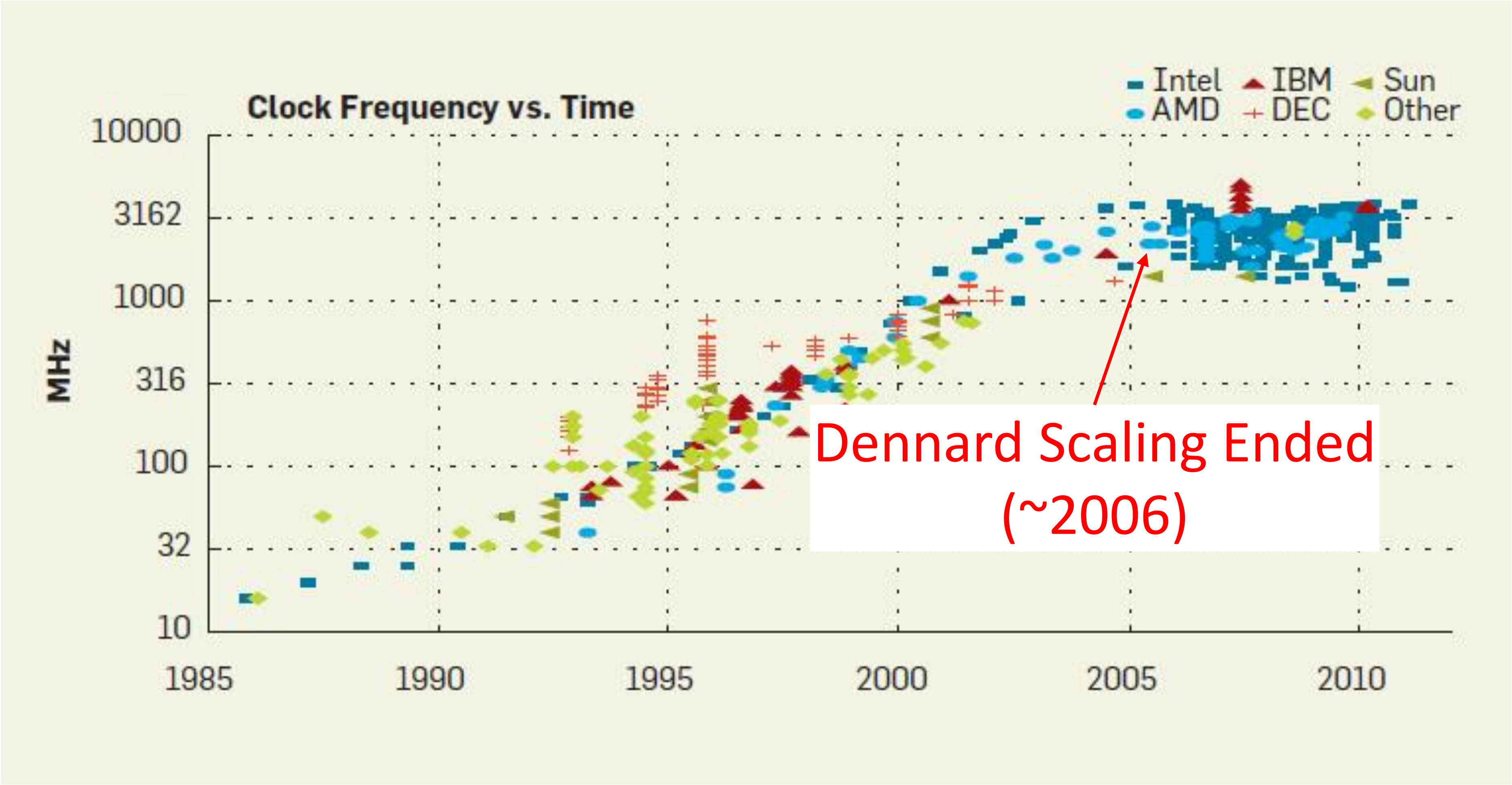
    Hotter chip → increased resistance → hotter chip → …

Option 1: Continue scaling frequency at increased power budget



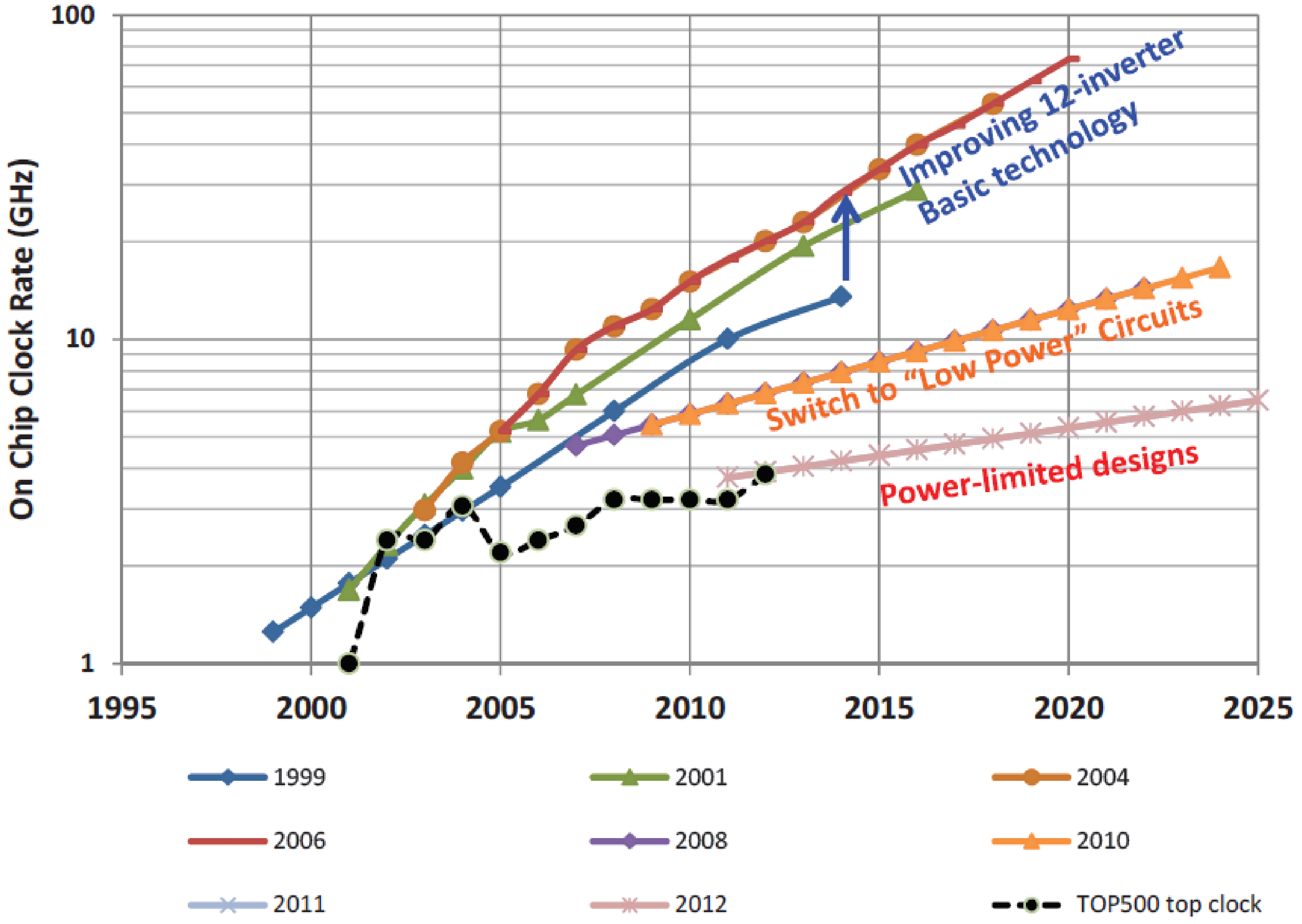
* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Option 2: Stop frequency scaling



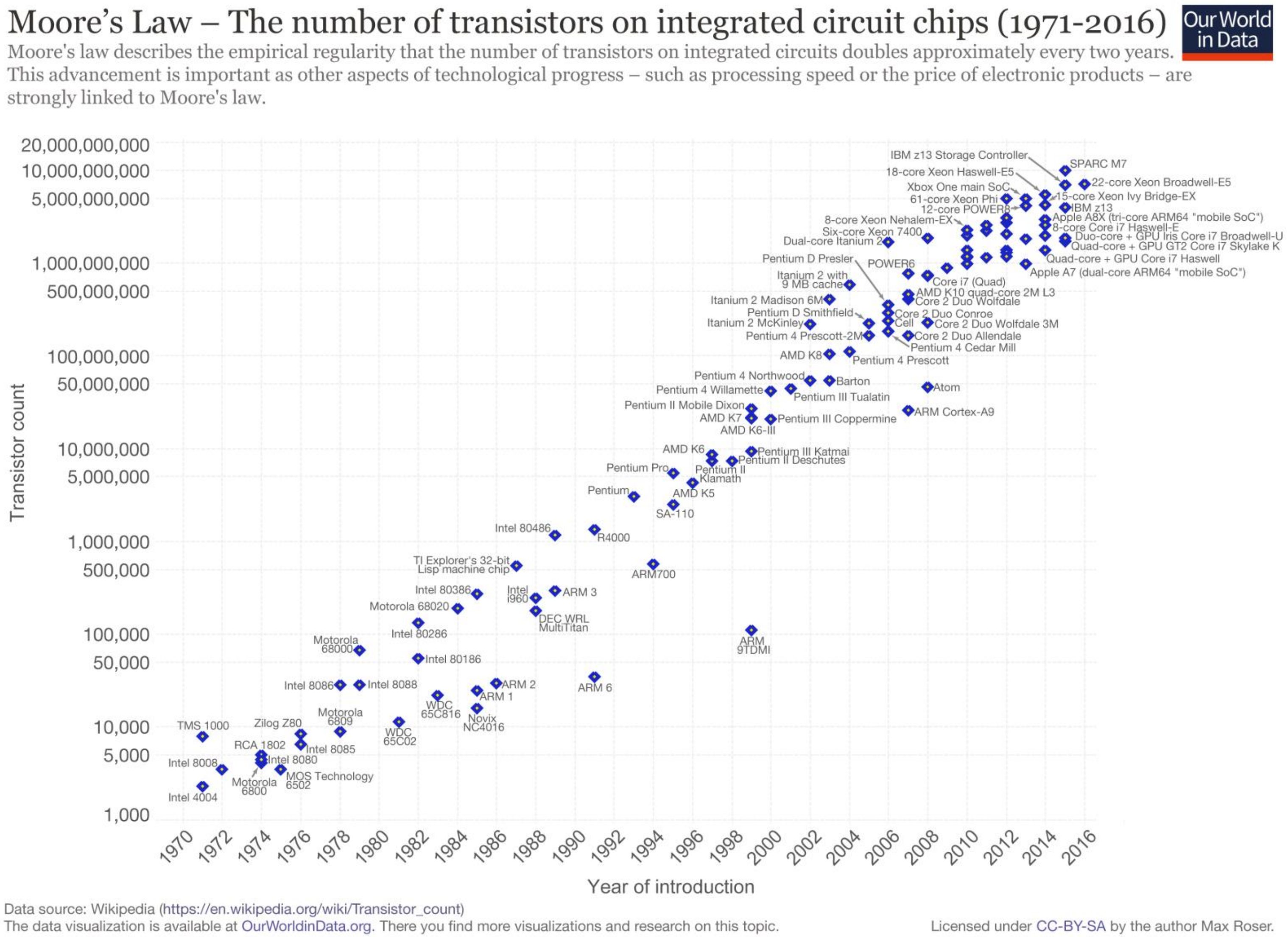
Danowitz et.al., “CPU DB: Recording Microprocessor History,” Communications of the ACM, 2012

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

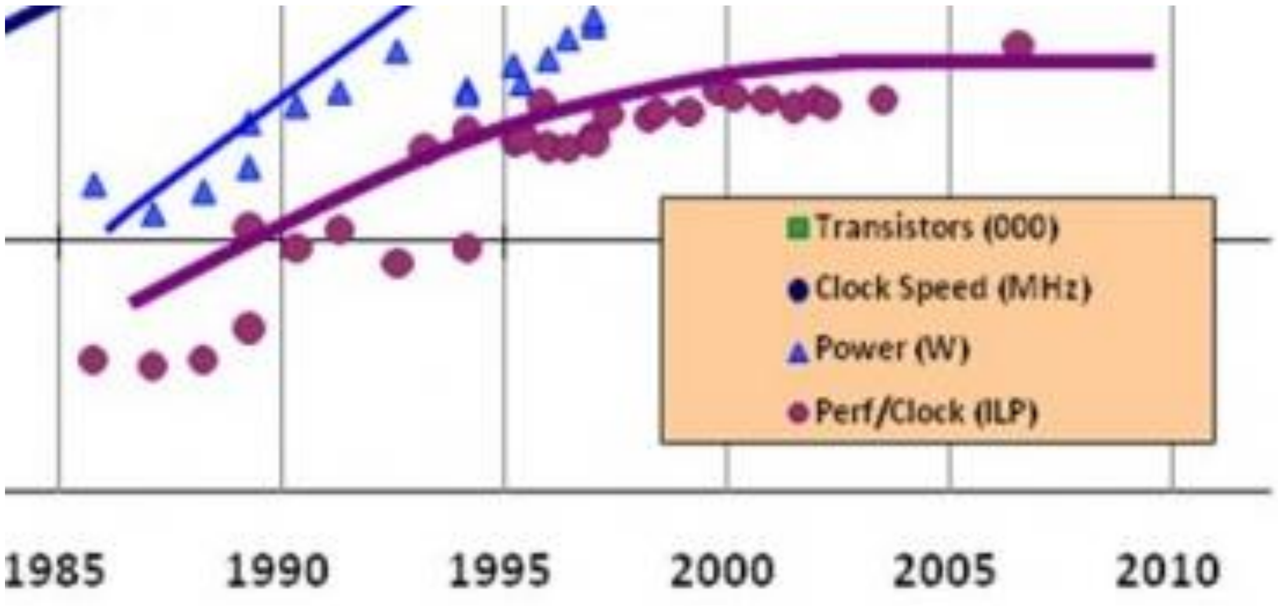


State of things at this point (2006)

- Single-thread performance scaling ended
  - Frequency scaling ended (Dennard Scaling)
  - Instruction-level parallelism scaling stalled … also around 2005

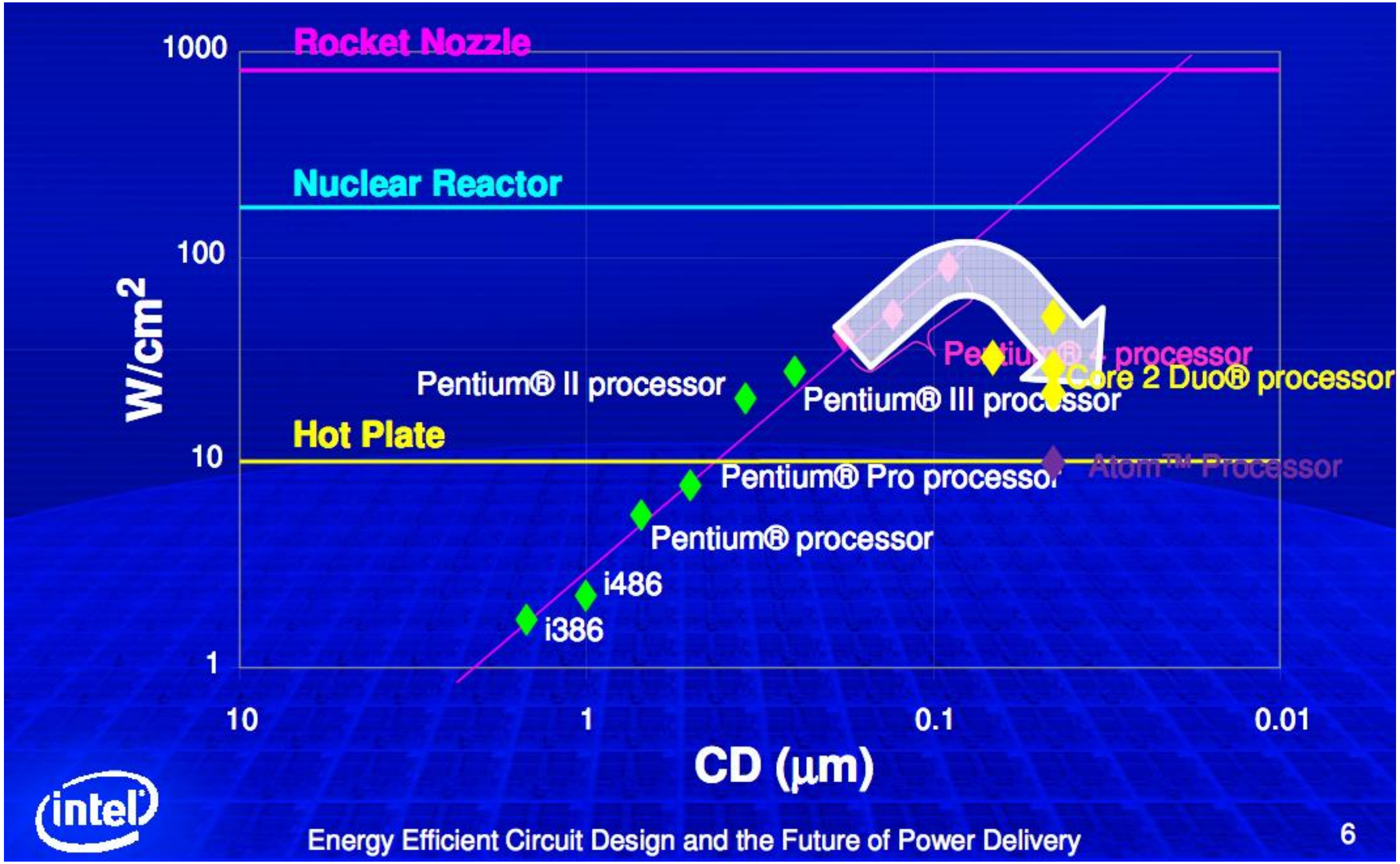
Instruction Level Parallelism

- Moore's law continues
  - Double transistors every two years
  - What do we do with them?

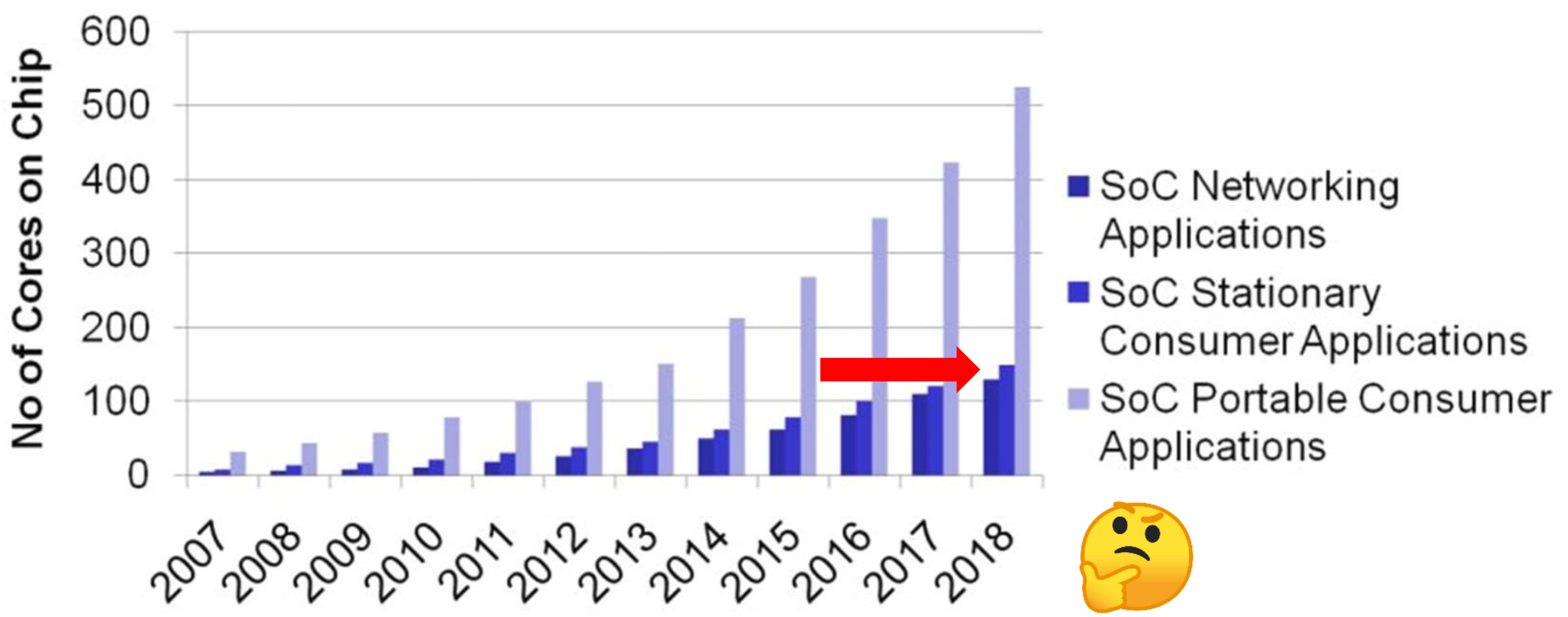


K. Olukotun, "Intel CPU Trends"

Crisis averted with manycores?



Crisis averted with manycores?



Source:
International Roadmap for Semiconductors 2007 edition (http://www.itrs.net/)

What happened?

Can't keep going up

Gate-oxide stopped scaling

Stopped scaling due to leakage

Stopped scaling due to thermal

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

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
  - We won’t get much performance improvements even if Moore’s law continues
  - We need to make the best use of those active transistors!

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

- 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

Taylor, Michael B. "Is dark silicon useful? Harnessing the four horsemen of the coming dark silicon apocalypse." DAC Design Automation Conference 2012. IEEE, 2012.

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