SFU

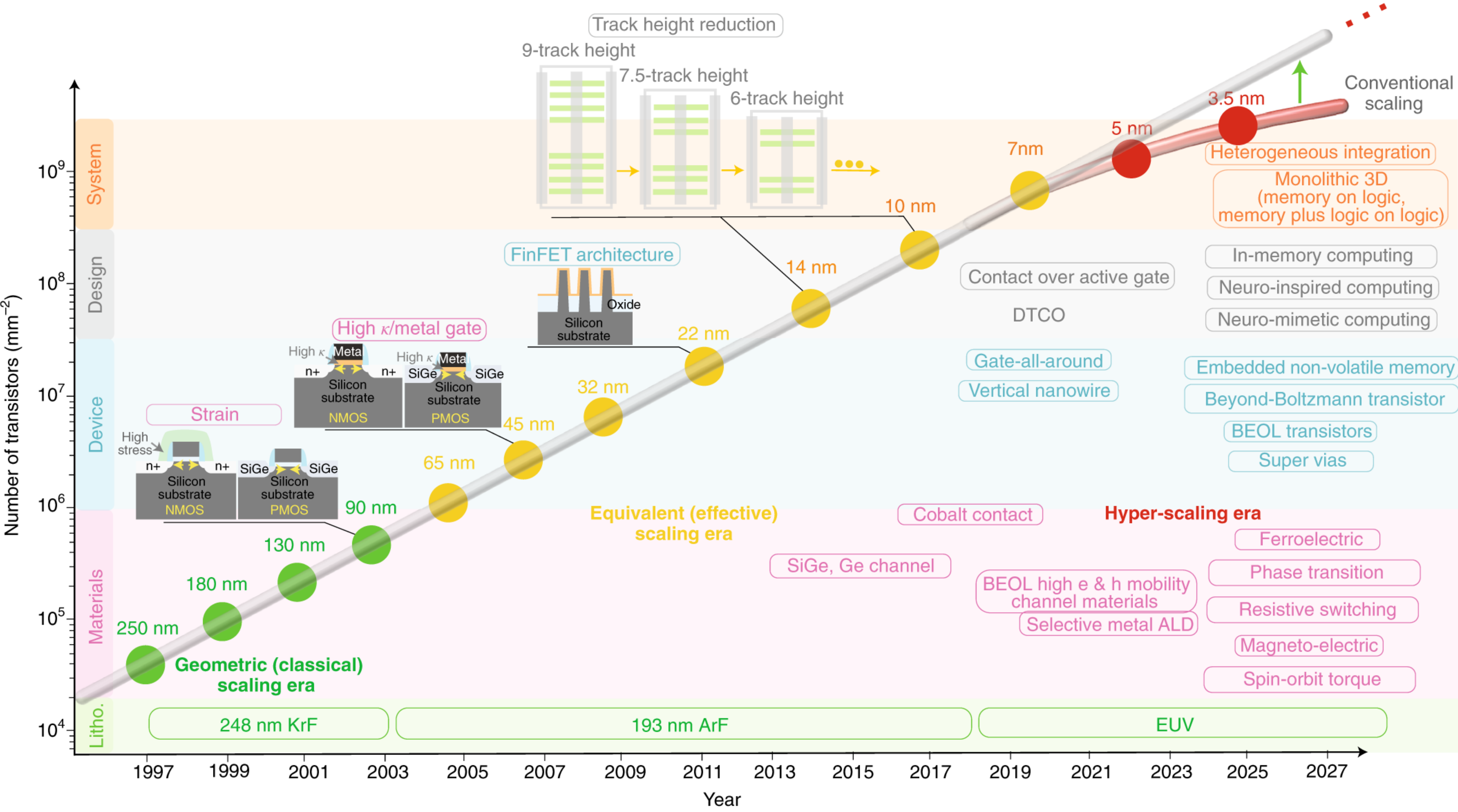
CMPT 450/750: Computer Architecture
Fall 2023

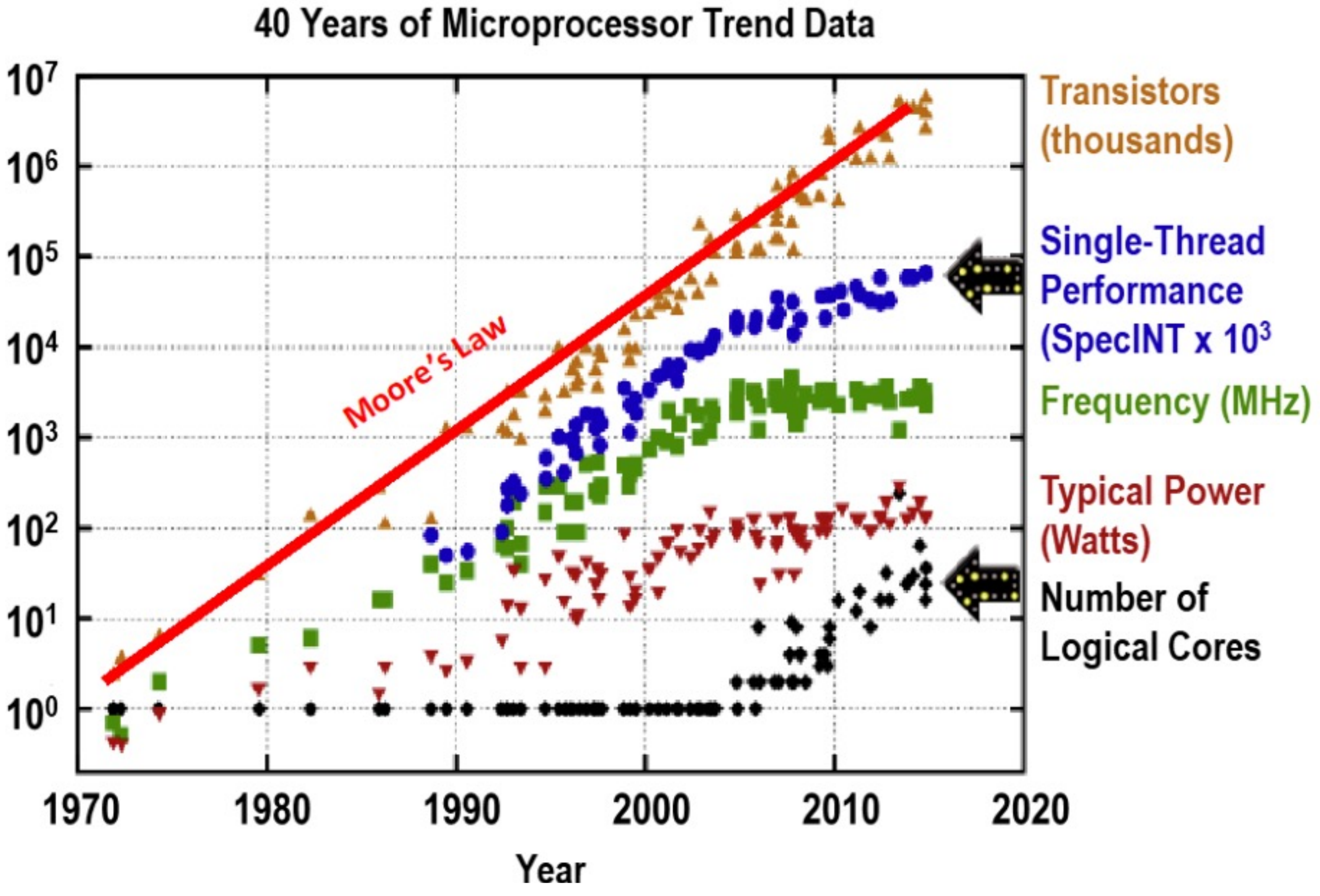
Impact of Technology on Computer Architecture

Alaa Alameldeen & Arrvindh Shriraman

Technology Trends: Logic Technology

- **Moore's Law: #transistors/IC die increasing exponentially**
- **Process technologies are labeled by "feature size", i.e. minimum size of a transistor or a wire in either the x or y dimension**
  - Feature sizes have decrease from 10 micrometers (μm) in 1971 to 0.007 μm in 2021
    - Now we refer to feature sizes in nanometers. Current technology node is 7 nm
  - Transistor density (#transistors/unit area) increases quadratically with a linear decrease in feature size
- **Historical size scaling trends:**
  - Transistor density has increased by 35% per year
    - Almost quadrupling every 4 years
  - Die (chip) size has increased between 10% and 20% per year
  - Combined effect: #transitors per chip increased at a rate of 40%-55% per year
    - Doubling every 18-24 months
- **Moore's Law has slowed down recently, so the doubling rate isn't quite as high**
- **Increases in transistor speeds have been slowing down for a longer time due to power limitations**


Track height reduction
9-track height
7.5-track height
6-track height
Conventional scaling
3.5 nm
5 nm
7nm
10 nm
14 nm
22 nm
32 nm
45 nm
65 nm
90 nm
130 nm
180 nm
250 nm
System
Design
Device
Materials
Litho.
Heterogeneous integration
Monolithic 3D (memory on logic, memory plus logic on logic)
In-memory computing
Neuro-inspired computing
Neuro-mimetic computing
Contact over active gate
DTCO
FinFET architecture
Oxide
Silicon substrate
High κ/metal gate
High κ
Metal
n+
SiGe
NMOS
PMOS
Strain
High stress
Gate-all-around
Vertical nanowire
Embedded non-volatile memory
Beyond-Boltzmann transistor
BEOL transistors
Super vias
Equivalent (effective) scaling era
Cobalt contact
Hyper-scaling era
Ferroelectric
Phase transition
Resistive switching
Magneto-electric
Spin-orbit torque
SiGe, Ge channel
BEOL high e & h mobility channel materials
Selective metal ALD
Geometric (classical) scaling era
248 nm KrF
193 nm ArF
EUV
Number of transistors (mm^-2)
10^4
10^5
10^6
10^7
10^8
10^9
Year
1997
1999
2001
2003
2005
2007
2009
2011
2013
2015
2017
2019
2021
2023
2025
2027



40 Years of Microprocessor Trend Data
Moore's Law
Transistors (thousands)
Single-Thread Performance (SpecINT x 10³)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
10⁷
10⁶
10⁵
10⁴
10³
10²
10¹
10⁰
1970
1980
1990
2000
2010
2020
Year

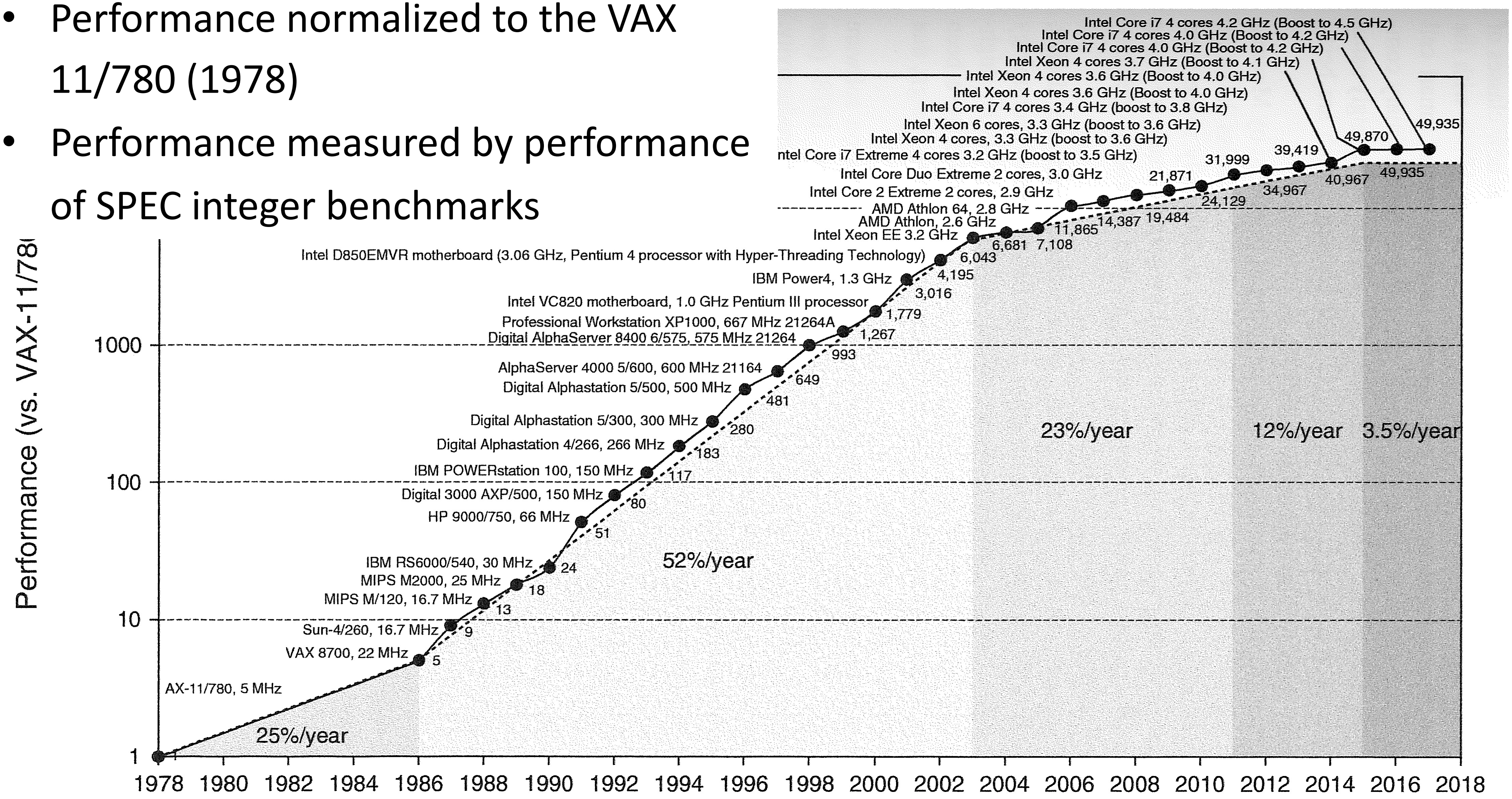
Technology Trends: Transistor Performance

- **Devices (i.e., transistors) shrink quadratically in area, both horizontally and vertically**
- **Reduction in transistor size led to a reduction in operating voltage**
- **In the past (before power wall), transistor performance improved linearly with decreasing feature size**
- **Improvement in both transistor count and performance led to dramatic improvements in microarchitecture**
  - Increasing operand width from 4-bits in 1971 to 64 bits today. We now have microprocessors with 64-bit addresses and 64-bit data
  - More aggressive superscalar processors with wider pipelines (power-limited)
  - Deeper pipelines to push for higher frequencies (power-limited)
    - Less work done per pipeline stage $\Rightarrow$ shorter cycle time and higher frequency
- **Power wall led to different architectural tradeoffs**
  - Wider SIMD units (e.g., vector processing units)
  - More cores per processor (i.e., multi-core processors)
  - Domain specific accelerators (covered next week)

Trends in Processor Performance over Time

ARCH Figure 1.1

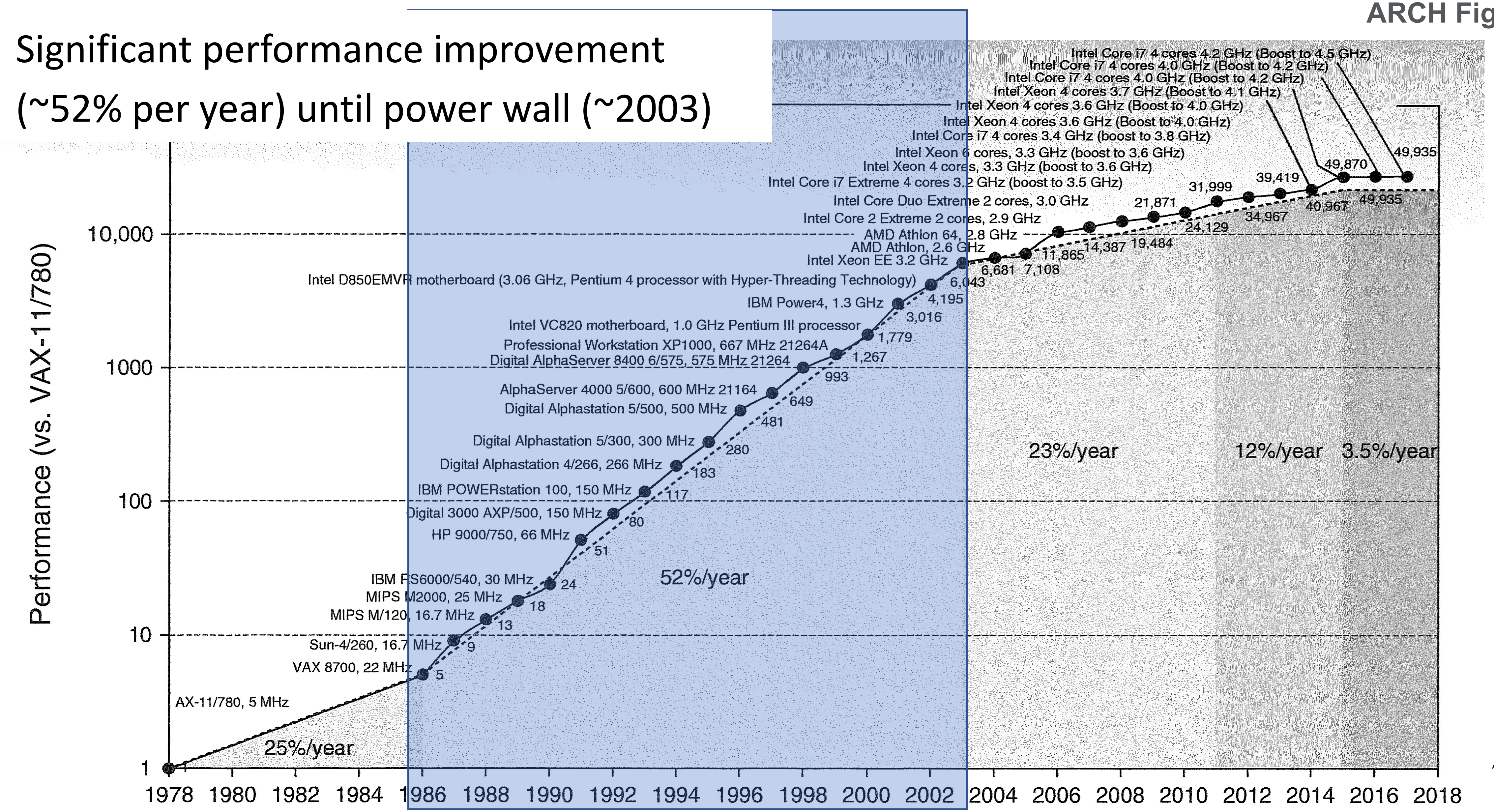
- Performance normalized to the VAX 11/780 (1978)
- Performance measured by performance of SPEC integer benchmarks



Trends in Processor Performance over Time

ARCH Figure 1.1

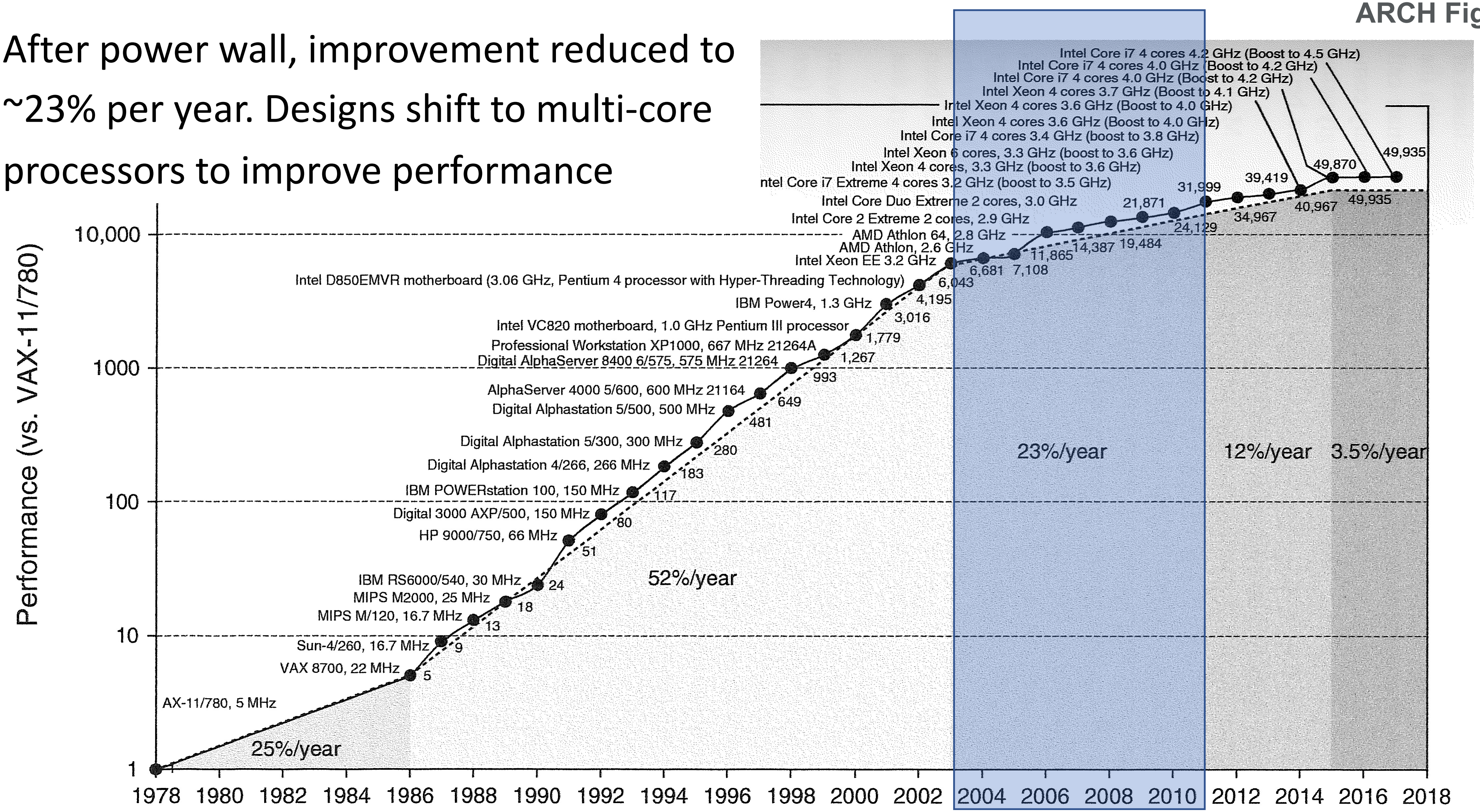
Significant performance improvement (~52% per year) until power wall (~2003)



Trends in Processor Performance over Time

ARCH Figure 1.1

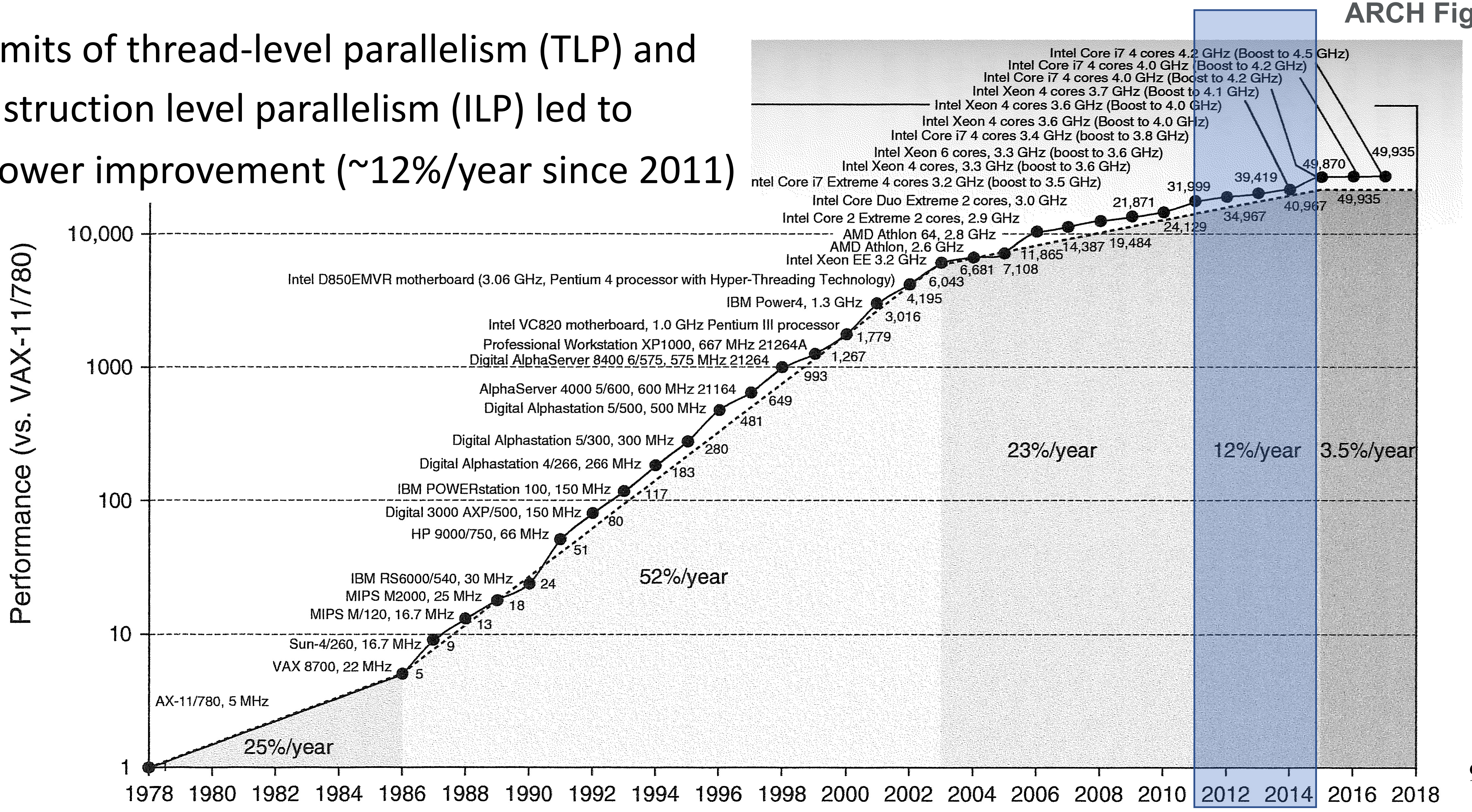
After power wall, improvement reduced to ~23% per year. Designs shift to multi-core processors to improve performance



Trends in Processor Performance over Time

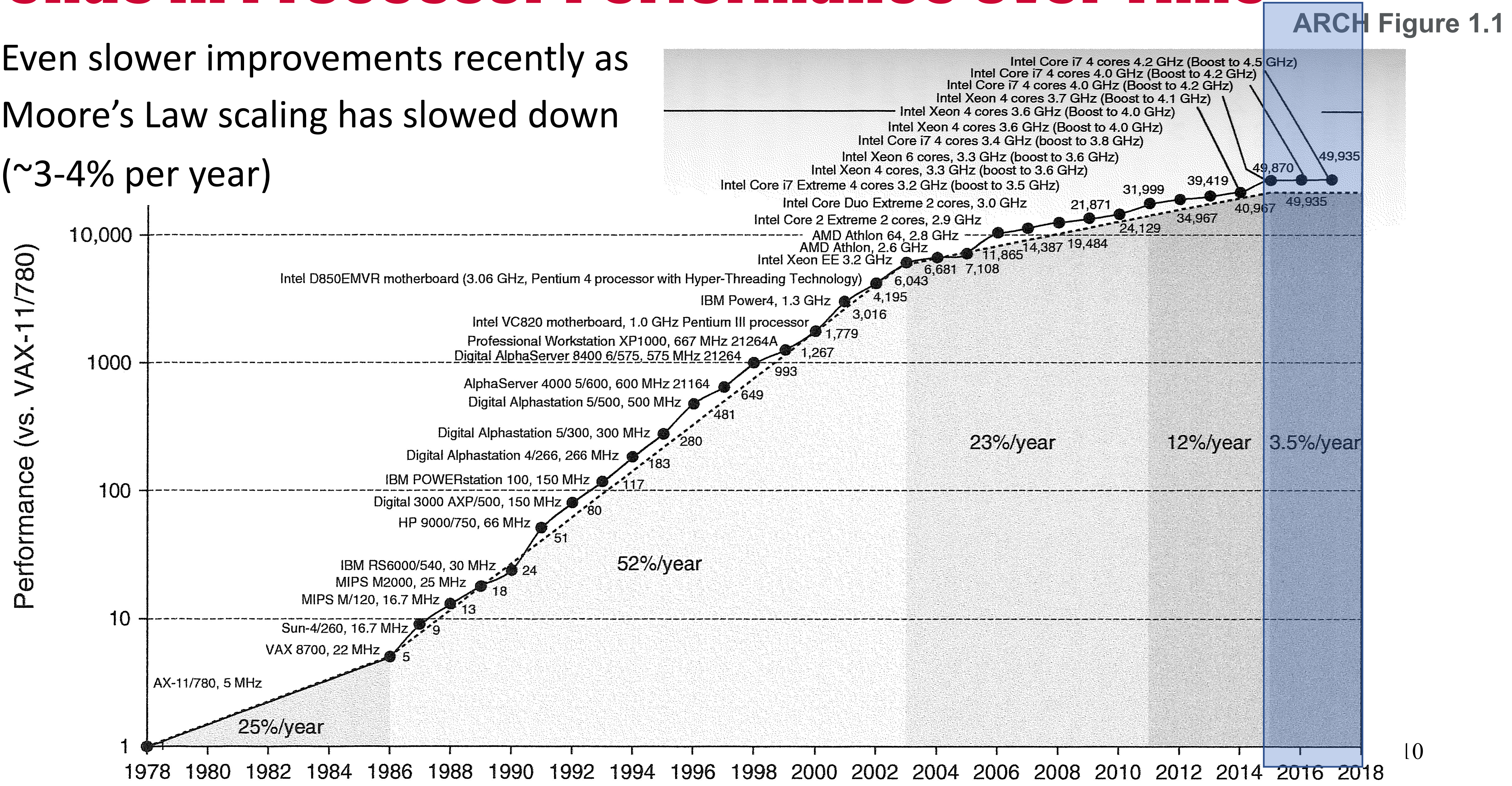
ARCH Figure 1.1

Limits of thread-level parallelism (TLP) and instruction level parallelism (ILP) led to slower improvement (~12%/year since 2011)

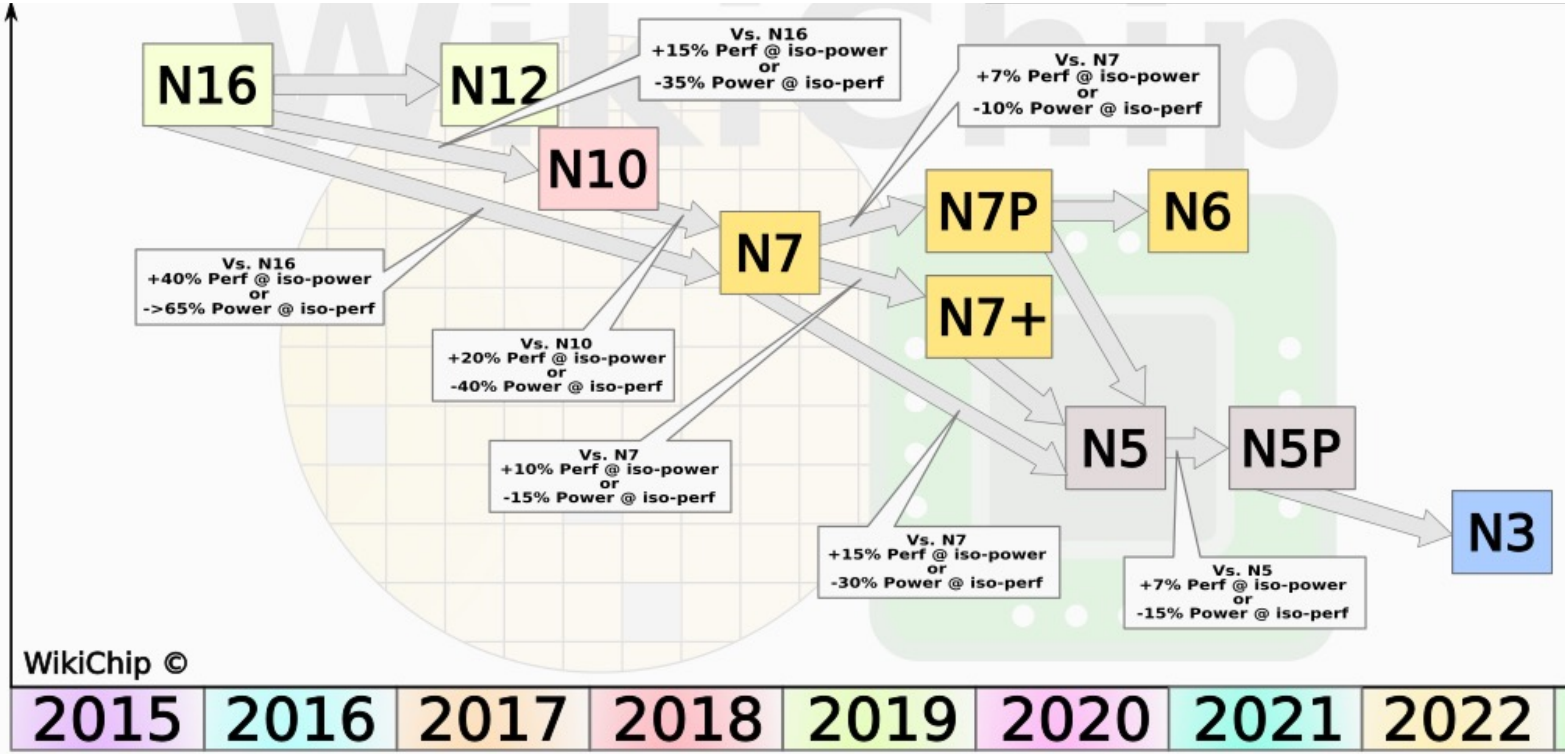


Trends in Processor Performance over Time

Even slower improvements recently as Moore's Law scaling has slowed down (~3-4% per year)



Current Status

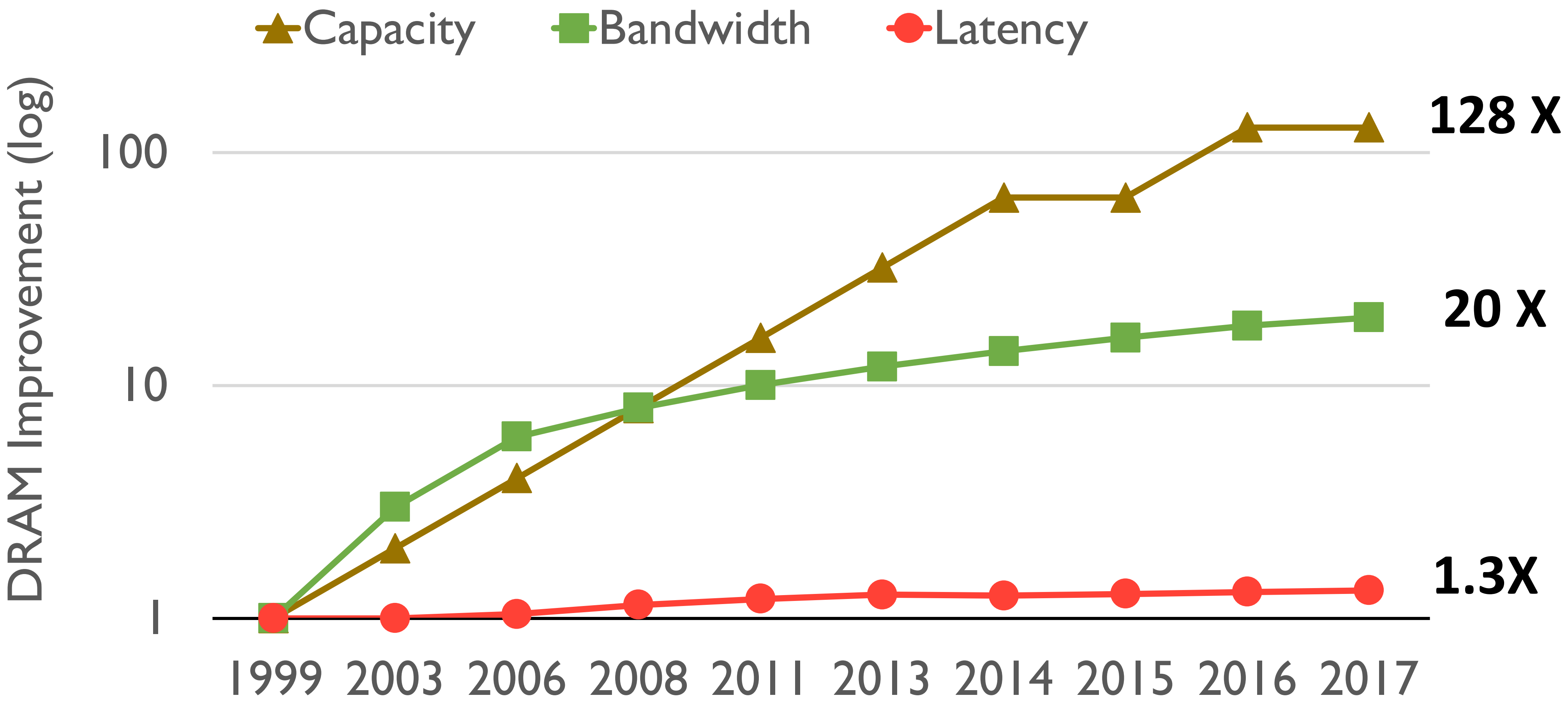


	Latency	Bandwidth / Channel	Max Capacity*	Significance	Programmers View
Reg	**0.2ns**		KB	In CPU	L1 - dereference pointer
Cache	**40ns**		KB		
DDR (Main)	**80-140ns**	32-51.2 GB/s (DDR5)	Up to 4TB		L2 - dereference pointer high perf memcpy
DDR (NUMA)	**170-250ns**	32-51.2 GB/s (DDR5)	Up to 8TB		
DDR (CXL)	**170-250ns**	32-51.2 GB/s (DDR5)	2-4 TB	CPU independent but local	L3 - dereference pointer high perf memcpy, swap
DDR (CXL Switched)	**300-400ns**	32-51.2 GB/s (DDR5)	64TB		
Far Memory	**2-4us**	100 GB/s (800g ethernet)	infinite	Network attached	L4 - memcpy, swap
SSD	**50-100us**				L5 - memcpy, swap

https://www.semianalysis.com/p/the-memory-wall

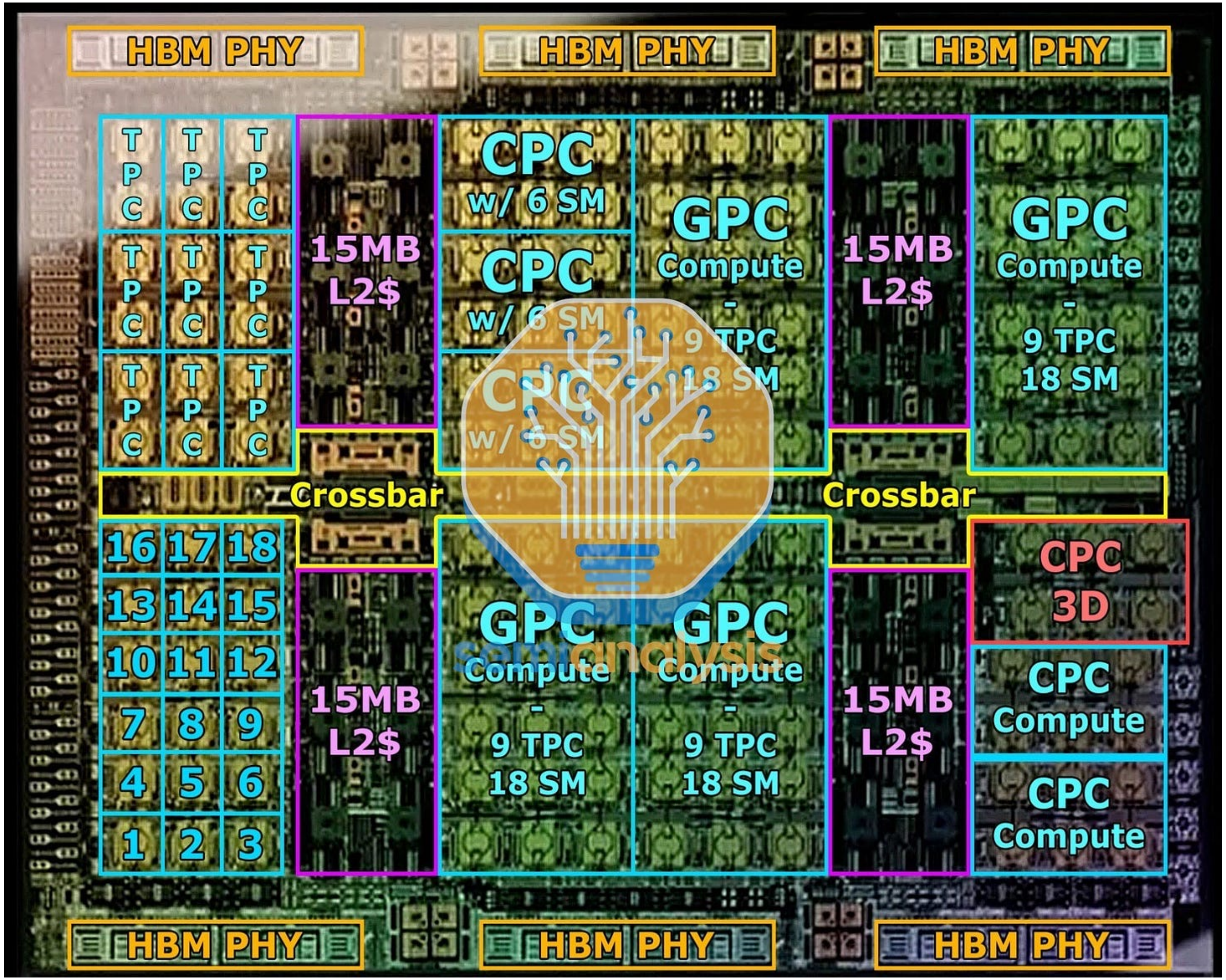
Technology Trends: Memory Technology

- **DRAM (Dynamic Random Access Memory)**
  - In the past, DRAM density was quadrupling every 3 years but has slowed down significantly



Source: “Memory Systems and Memory-Centric Computing Systems Tutorial” by Prof. Onur Mutlu, September 2019 - https://safari.ethz.ch/memory_systems/Perugia2019/lib/exe/fetch.php?media=onur-perugia-ss-2019-part1-memoryimportancetrendsfundamentals-september-3-2019-beforelecture.pdf

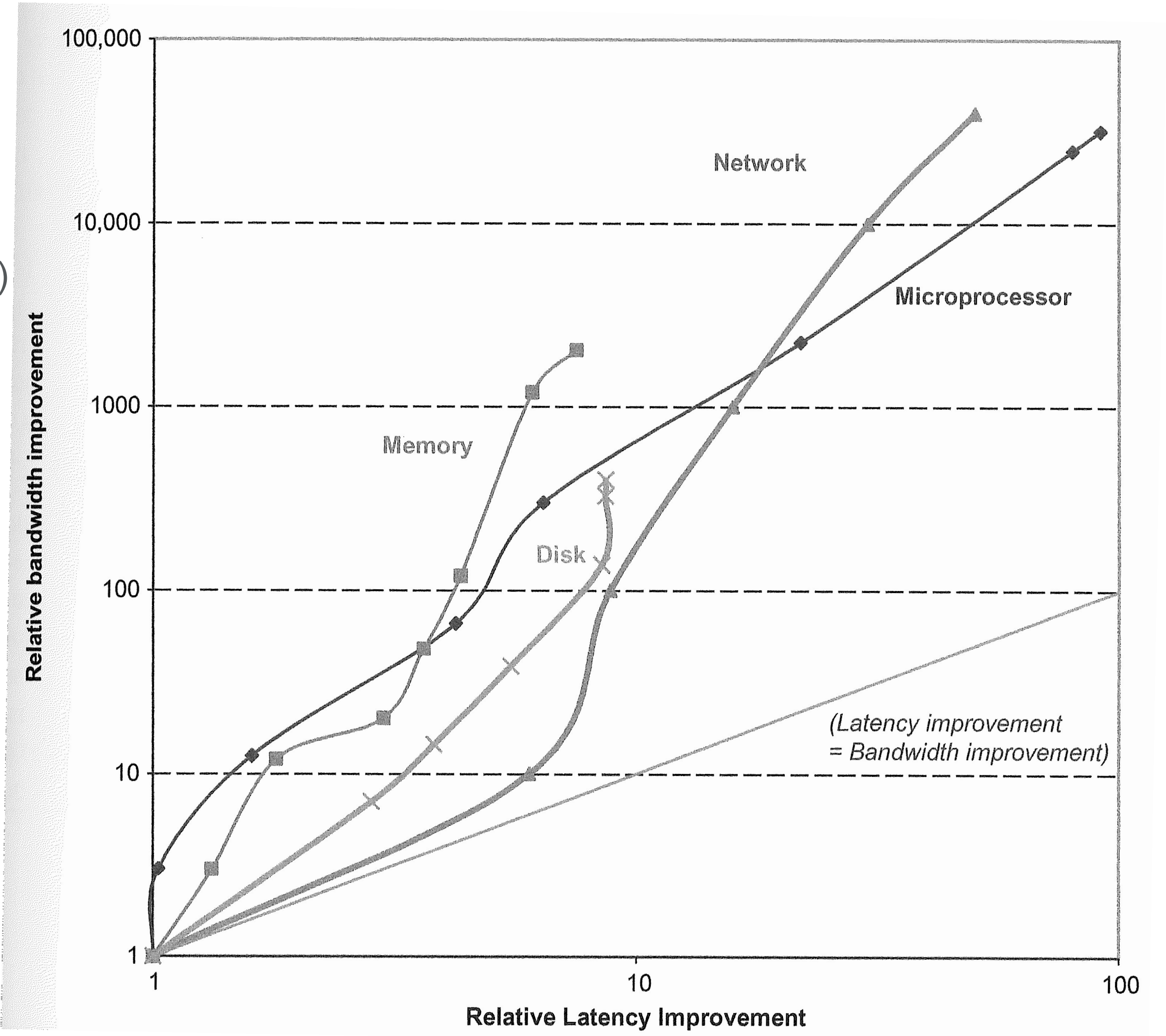
Memory bandwidth competes for die space



Technology Trends: Bandwidth vs. Latency

ARCH Figure 1.9

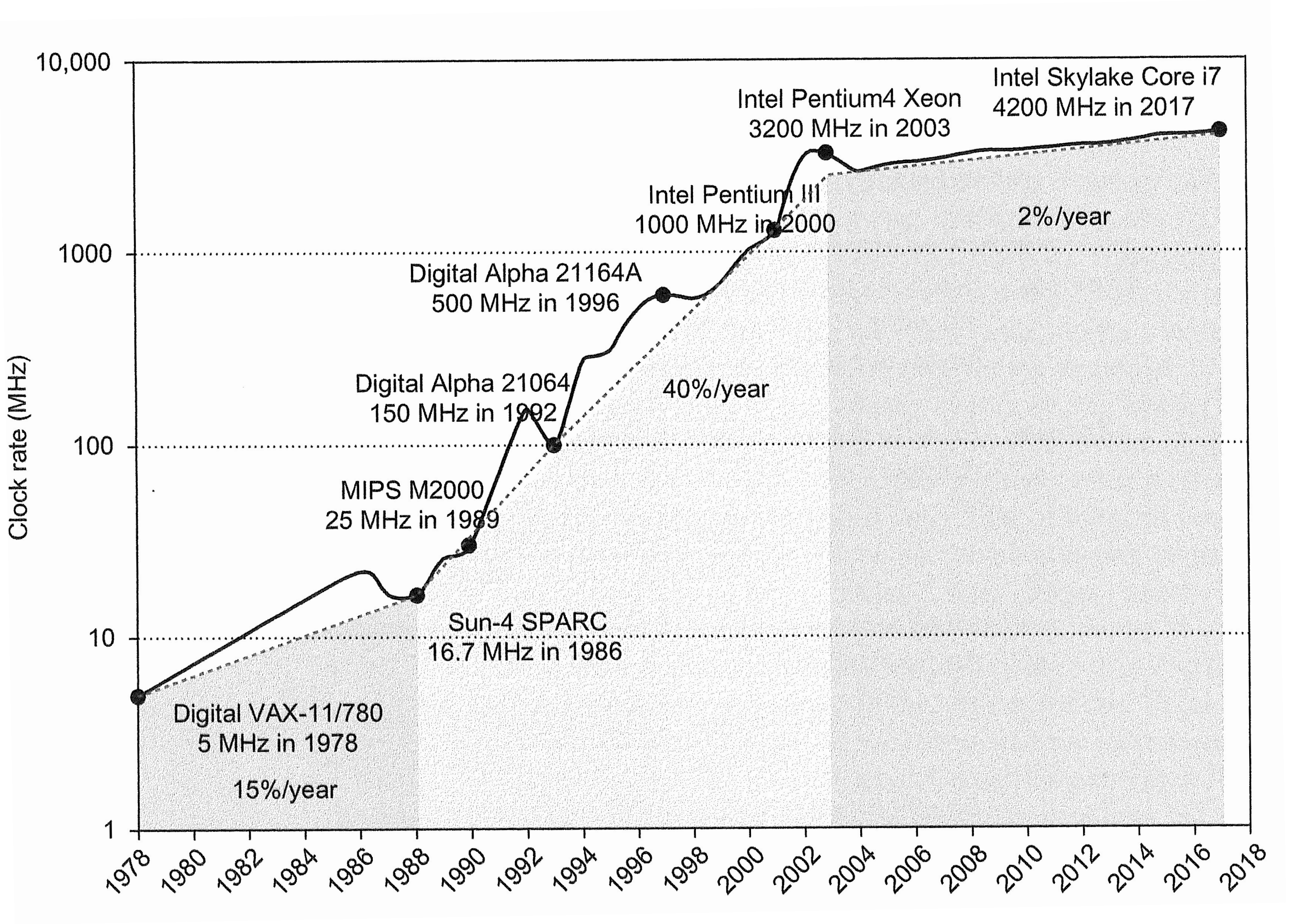
- **Design Points:**
  1. Intel 20286: 16-bit CPU (1982)
  2. Intel 20386: 32-bit CPU (1985)
  3. Intel 80486: Pipelineing, caches, FPU (1989)
  4. Intel Pentium: 64-bit, 2-way superscalar (1993)
  5. Intel Pentium Pro: OoO, 3-way SS (1997)
  6. Intel Pentium 4: wider SS, L2 on chip (2001)
  7. Intel core i7: Multicore, L3 on chip (2015)
- **Latency improved 8-91X for different system components**
- **Bandwidth improved 400-32,000X**
- **Both improvement trends slowed down recently, but latency is much slower**



Technology Trends: Frequency

ARCH Figure 1.11

- **Data from 1978 to 2017**
- **Before power wall, frequency improved ~40% per year**
  - ➢ Combined with architectural improvements, this led to ~52%/year improvement in processor performance.
- **Since power wall, frequency has been mostly flat (~2% increase per year)**
- **What is the correlation between frequency and power?**



Power and Energy

What is Power?

- **Electric power is the rate (per unit time) at which electrical energy is transferred by an electric circuit**
- **Power Equation:**

$$Power = \frac{Total\ Energy}{Time} \qquad or \qquad P = \frac{E}{T}$$

- **Power is measured in Watts; Energy is measure in Joules**
  - ➢Watt = Joules/sec; Joule = Watt x sec
- **Power and Energy fall into two main classes:**
  - ➢Dynamic Power/Energy: Used to switch transistors (from logic 0 to 1 and vice versa)
  - ➢Static Power/Energy: Caused by leakage current which flows even when transistors are turned off (Power = Voltage x Current)

What is the Maximum Power of a Processor?

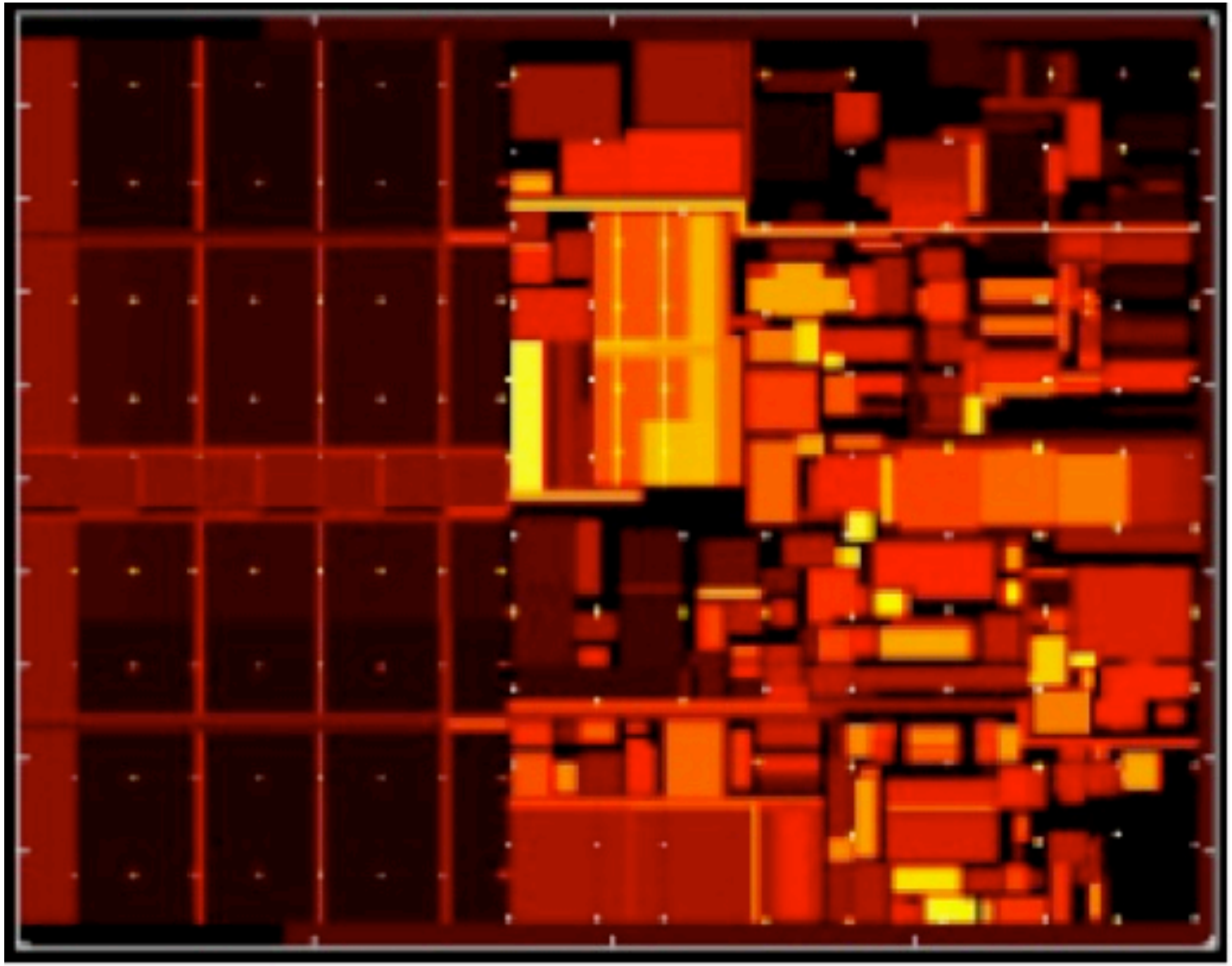
- **System power is provided from a power supply source (e.g., electric outlet, battery)**
- **Devices operate in a voltage range between Vmin and Vmax:**
  - Vmin is the minimum operating voltage below which devices will malfunction (i.e., not switch properly)
  - Vmax is the maximum operating voltage to safely operate a device.
- **If processor attempts to draw more power than available supply, i.e., draw more current, then its voltage would drop ($P = V \times I$)**
  - Lowering voltage causes device switching to slow down, which slows down performance
- **Processors have varying power consumption**
  - Processors don’t always run at peak current
  - To save power, Voltage can be regulated and processors can slow down when performance is not critical

Thermal Design Power (TDP)

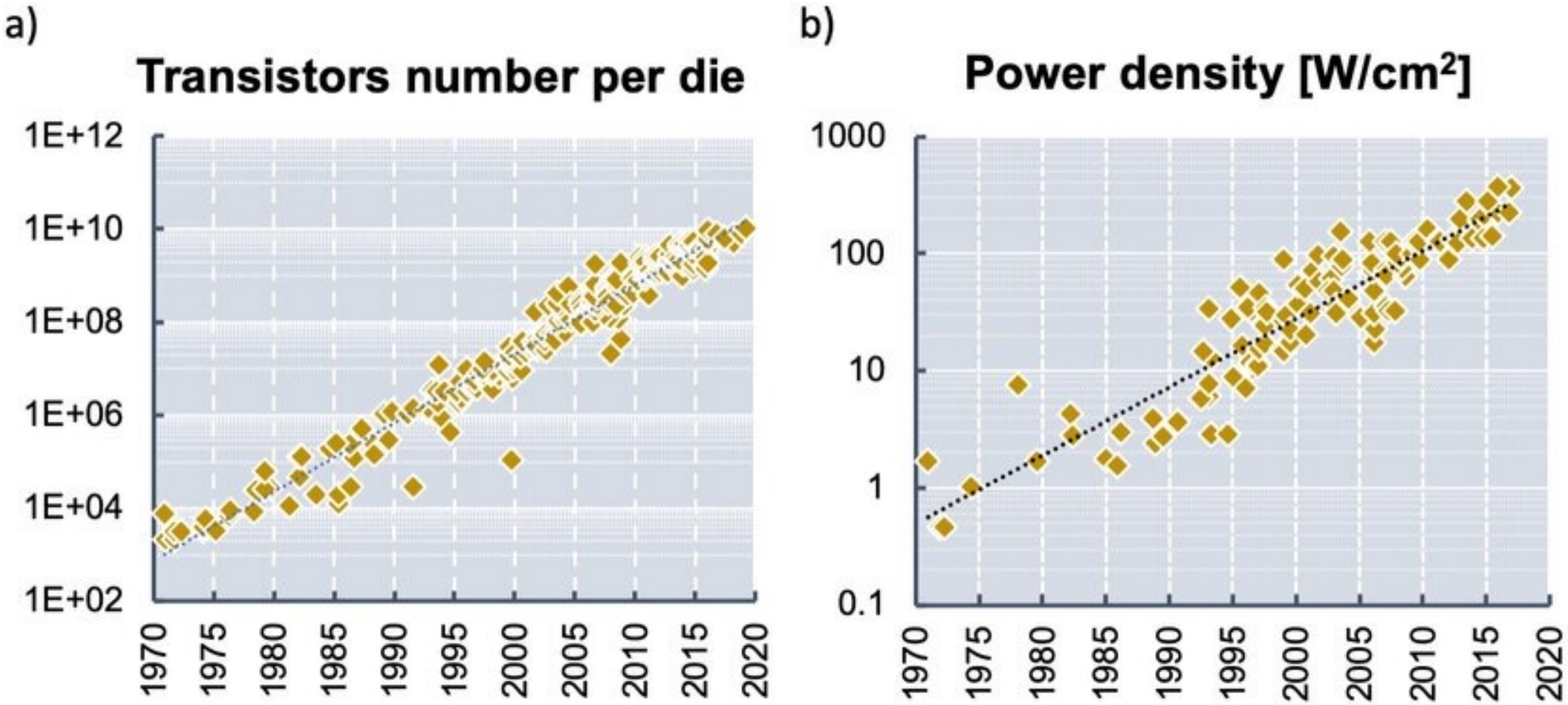
- **Sustained power consumption for a processor/system**
  - ➢Used to determine the cooling requirements of a system
- **TDP is usually lower than peak power (~1.5x higher); but is higher than average power**
  - ➢System power supply is designed to exceed TDP
- **Cooling Systems need to match or exceed TDP**
  - ➢Failure to cool circuits properly can lead to overheating which causes device failure and potentially permanent damage
- **To manage overheating, processors can**
  - ➢Reduce power by lowering frequency
  - ➢Power down the chip

Power Density

- Power Density = Power per unit area (Watts/mm^2)
- Problem: Denser power is harder to cool, leading to overheating
- Power density increases with shrinking technology nodes (since transistor density is increasing)



Simulated Power Density Map for Intel Pentium M Processor
Source: Genossar & Shamir, Intel Technology Journal, 2003



a)
Transistors number per die
1E+12
1E+10
1E+08
1E+06
1E+04
1E+02
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020
b)
Power density [W/cm²]
1000
100
10
1
0.1
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020

Energy Efficiency in a Processor

- Energy required to execute a program is the product of average power multiplied by execution time

$$Energy\ (Program\ P) = Average\ Power \times Execution\ Time(P)$$

- Energy is a more relevant metric than power since it measures power over a period of time for a specific task
- **Remember that for energy, lower is better!**
- Energy-efficient processors consume lower energy to execute the same task
- Sometimes we care about both energy and performance, so use metrics like Energy Delay product (ED) or Energy x Delay^2 (ED^2)
    - ➢Again, lower is better

Energy Efficiency Example

Processor A executes program P in 10 seconds and consumes 10 Watts on average during that execution. Processor B executes the same program P in 6 seconds and consumes 15 Watts on average during that execution. Which Processor is more energy-efficient?

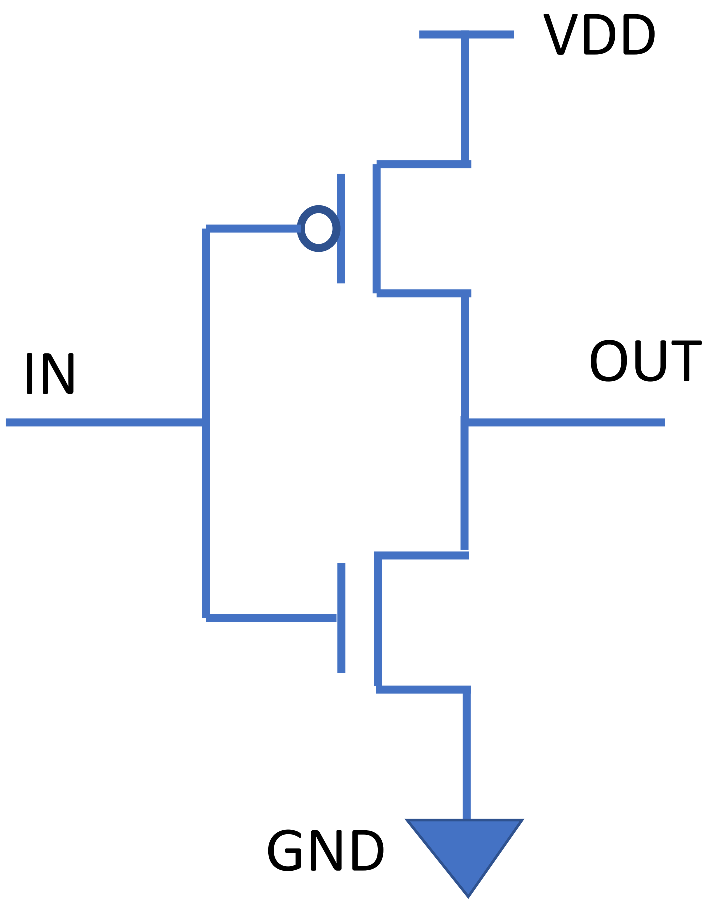
$$Energy\ (A) = Average\ Power\ (A) \times Execution\ Time(A) = 10 \times 10 = 100\ Joules$$

$$Energy\ (B) = Average\ Power\ (B) \times Execution\ Time(B) = 15 \times 6 = 90\ Joules$$

- **So B is more energy-efficient (even though it consumes more average power than A)**

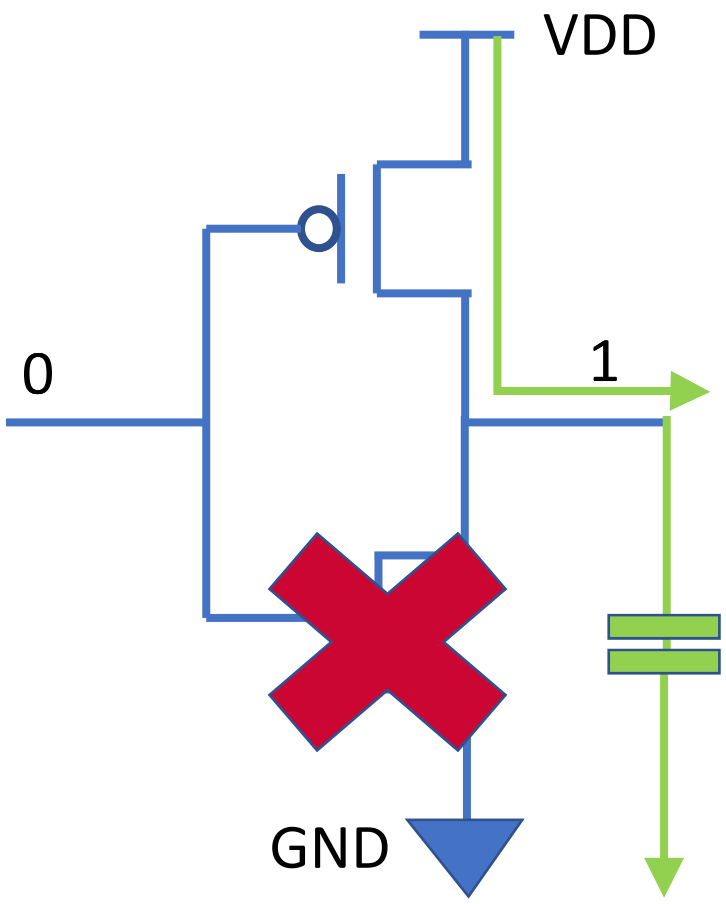
Dynamic Energy

- Energy consumed when switching transistors
- Also called “Active Energy”
- Example: Inverter



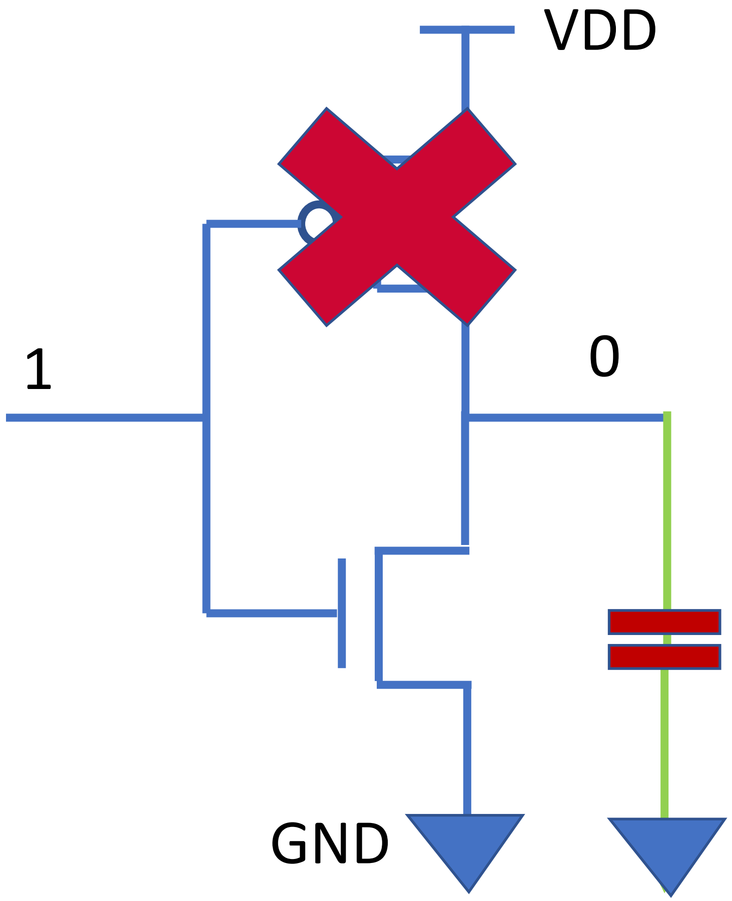
Dynamic Energy

- Example: Inverter
- “0” Input turns on top transistor, turns off bottom transistor, allowing VDD to flow to output, charging capacitor



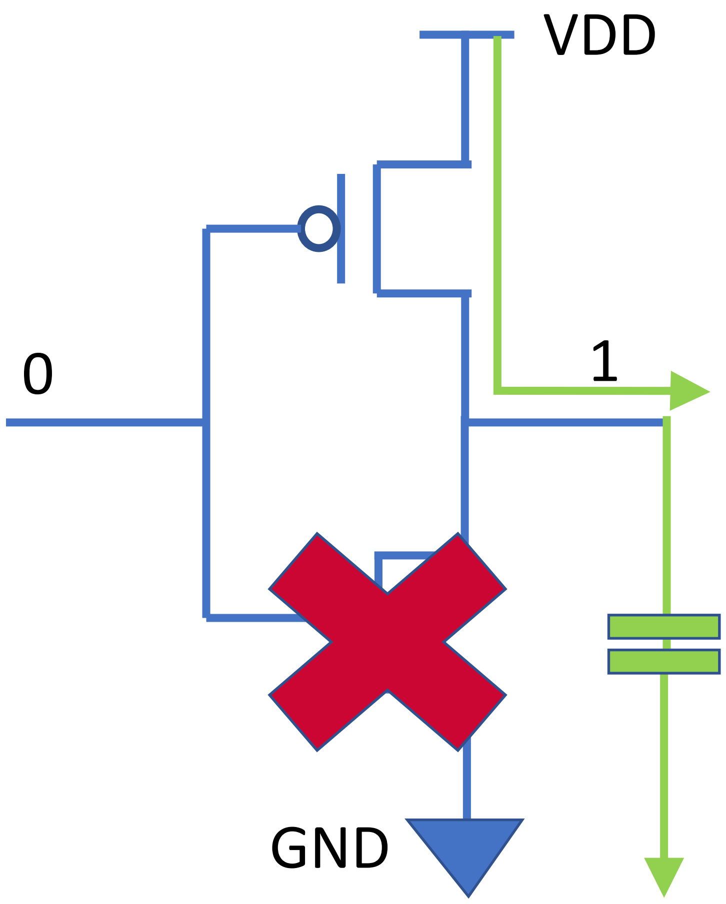
Dynamic Energy

- **Example: Inverter**
- **"1" Input turns off top transistor, turns on bottom transistor, discharging capacitor flow to output, discharging capacitor**



Dynamic Energy

- **Example: Inverter**
- **Switching back to "0" turns on top transistor, turns off bottom transistor, so capacitor needs to charge again**
- **Note that switching capacitors can be:**
  - Gates of other transistors; OR
  - Wires for busses and interconnects



Dynamic Energy

- **Dynamic energy is proportional to the capacitive load and the square of the voltage**
  - Capacitive load is a function of #transistors connected to an output, as well as the capacitance of wires and transistors determined by the process technology

$$Energy_{dynamic} \propto Capacitive\ Load \times Voltage^2$$

(energy of the pulse of the logic transition 0→1 →0 or 1 →0 →1)

- **For a single transition (0 →1 or 1 →0):**

$$Energy_{dynamic} \propto {}^1/_2 \times Capacitive\ Load \times Voltage^2$$

- **Since Power is energy divided by switching time, and switching time is the reciprocal of frequency:**

$$Power_{dynamic} \propto {}^1/_2 \times Capacitive\ Load \times Voltage^2 \times f$$

Reducing Dynamic Energy/Power

- Equations:

$$Energy_{dynamic} \propto \ ^1/_2 \times Capacitive\ Load \times Voltage^2$$

$$Power_{dynamic} \propto \ ^1/_2 \times Capacitive\ Load \times Voltage^2 \times f$$

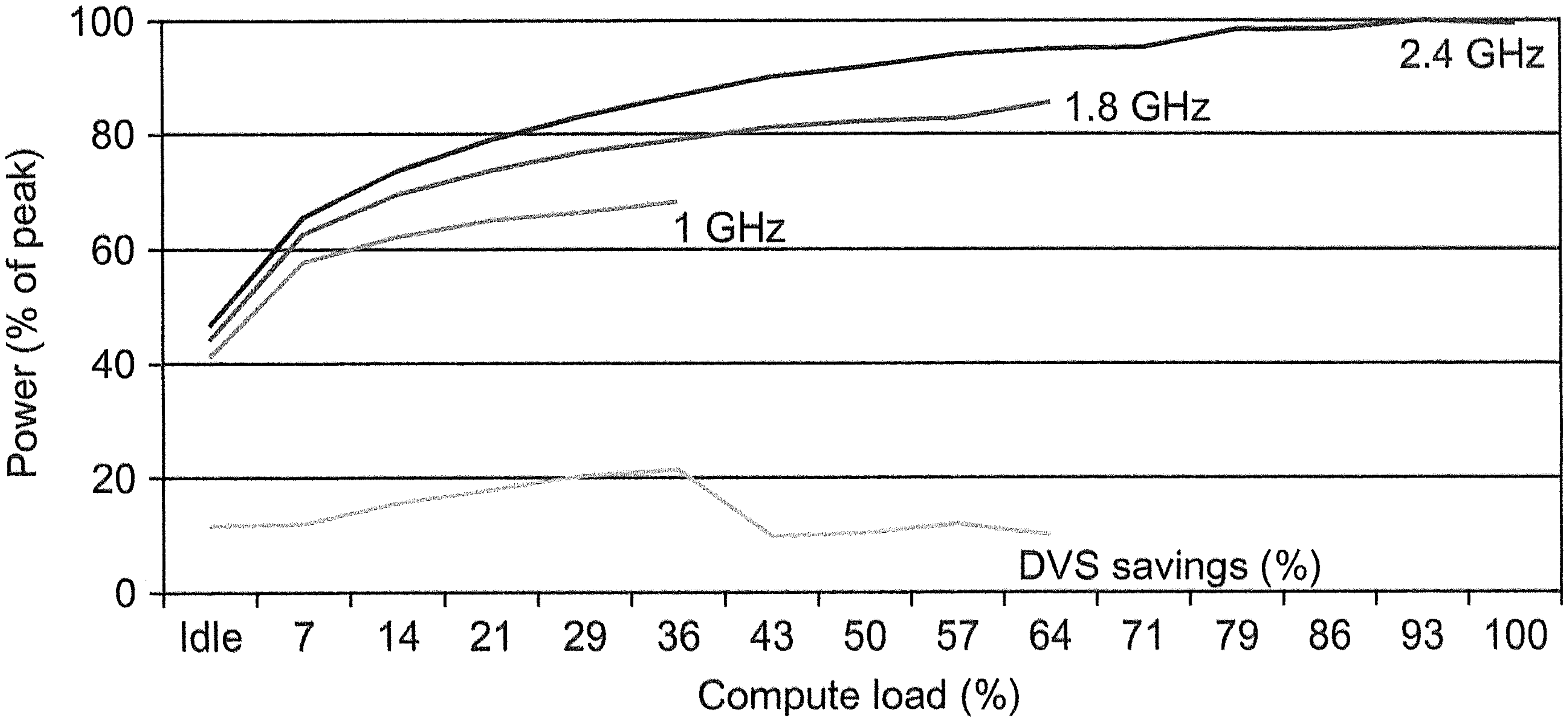
- Energy can be greatly reduced by lowering voltage. Power can be reduced by lowering voltage and frequency
- Note that frequency depends on voltage: Higher frequency requires fast switching time which requires higher voltage.
- This led to the "Cube Law": $Power_{dynamic} \propto \ Voltage^3$
- Implication: In the limit, a 1% change in voltage leads to a 3% change in power
- So processors can save power (and therefore energy) by lowering voltage and frequency when performance isn't critical

Techniques to Reduce Power and Energy

- **Power and energy can be reduced by:**
  - Turning off clock (or powering off) inactive structures
  - **Dynamic Voltage-Frequency Scaling (DVFS):** When there is low activity, or when performance is not critical, the processor can reduce operating frequency and operating voltage. Typically a processor has a few operating points (voltage, frequency)

Dynamic Voltage-Frequency Scaling (DVFS) Example

- AMD Opteron processor with 8GB of DRAM and three operating modes: 1/1.8/2.4GHz
- At lower operating modes, the processor can only handle a fraction of the compute load



ARCH Figure 1.12

Techniques to Reduce Power and Energy

- **Power and energy can be reduced by:**
  - Turning off clock (or powering off) inactive structures
  - **Dynamic Voltage-Frequency Scaling (DVFS):** When there is low activity, or when performance is not critical, the processor can reduce operating frequency and operating voltage. Typically a processor has a few operating points (voltage, frequency)
  - Designing for the common case: Since mobile devices are often idle, memory and storage have low power modes to save energy
    - Example: Standby mode where processor is powered off while DRAM remains on self-refresh for fast wakeup
    - Example: Hibernate where processor and DRAM are powered off. Slower wakeup.
  - Overclocking: Run at a lower clock in the common case, run at a faster clock when performance is needed.
    - In a multi-core processor, all processors except one can be turned off, and one processor is overclocked to improve single-thread performance

Dynamic Power/Energy Example

Processor A runs at a frequency of 4GHz with an operating voltage of 1.3V. How would dynamic energy and power change if the processor reduces its frequency to 3GHz and its operating voltage to 0.975V?

Energy is proportional to V^2, power is proportional to $V^2 F$

$$\frac{Energy_{new}}{Energy_{old}} = \frac{V_{new}^2}{V_{old}^2} = \frac{0.975^2}{1.3^2} = 0.5625$$

$$\frac{Power_{new}}{Power_{old}} = \frac{V_{new}^2 \, F_{new}}{V_{old}^2 \, F_{old}} = \frac{0.975^2 . \times 3}{1.3^2 \times 4} = 0.422$$

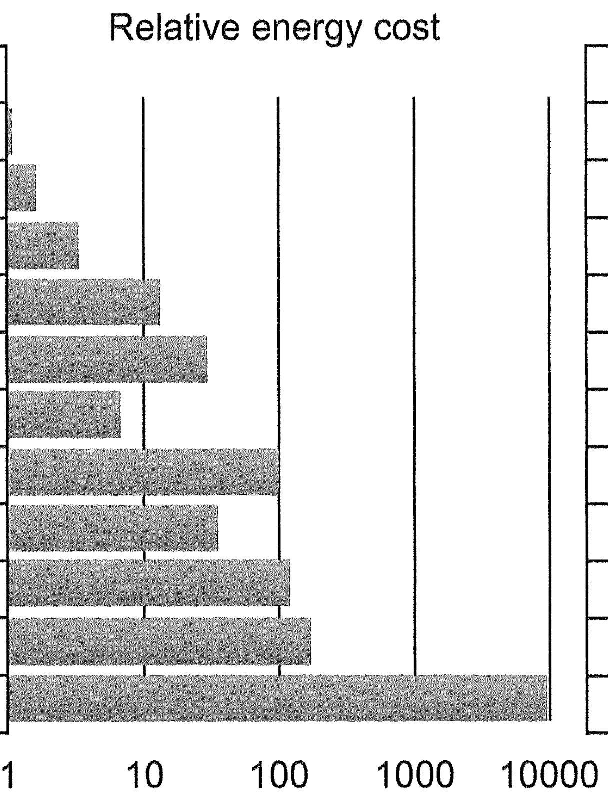
- So the dynamic energy reduces to 56.25% of its original value, while dynamic power reduces to 42.2% of its original value

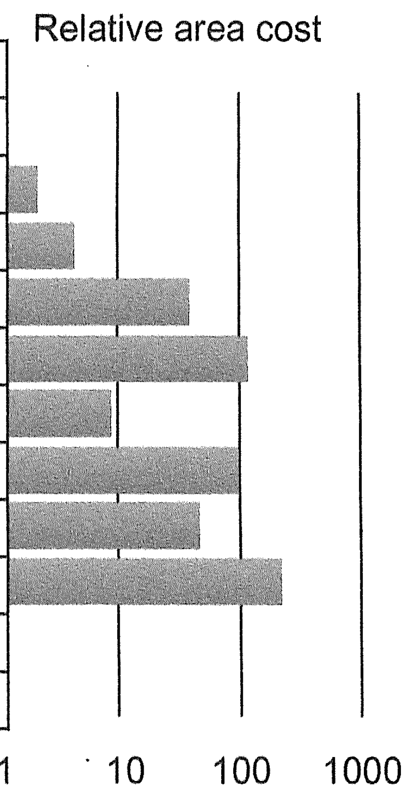
Comparing Dynamic Energy for Different Operations

- **Dynamic energy increases with the complexity of operations**

ARCH Figure 1.13

Operation:	Energy (pJ)	Area (μm^2)
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FB Add	0.4	1360
32b FB Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FB Mult	1.1	1640
32b FB Mult	3.7	7700
32b SRAM Read (8KB)	5	N/A
32b DRAM Read	640	N/A

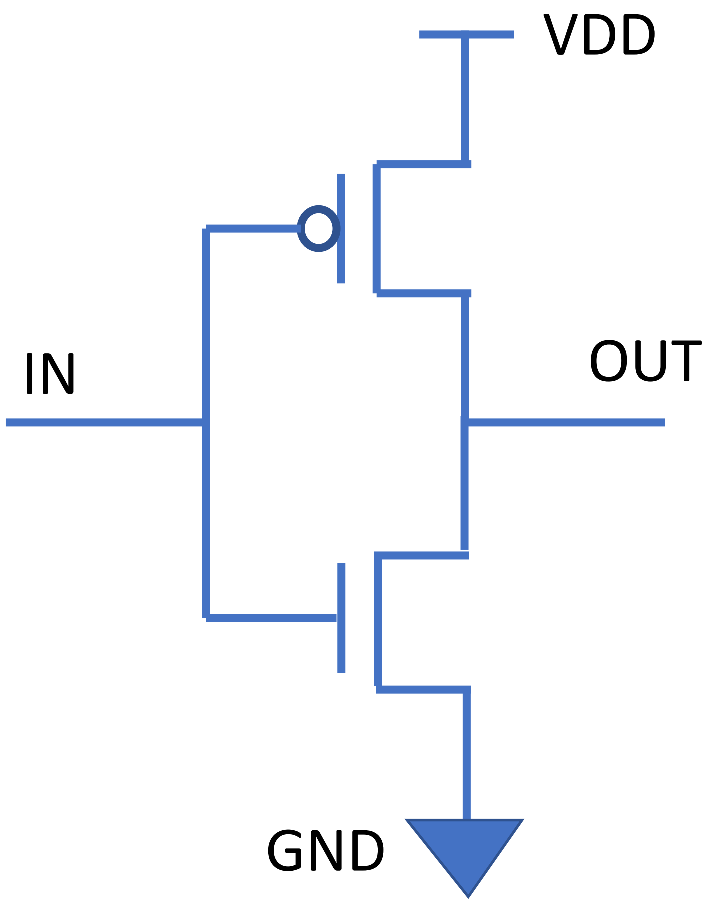




Energy numbers are from Mark Horowitz *Computing's Energy problem (and what we can do about it)*. ISSCC 2014
Area numbers are from synthesized result using Design compiler under TSMC 45nm tech node. FP units used DesignWare Library.

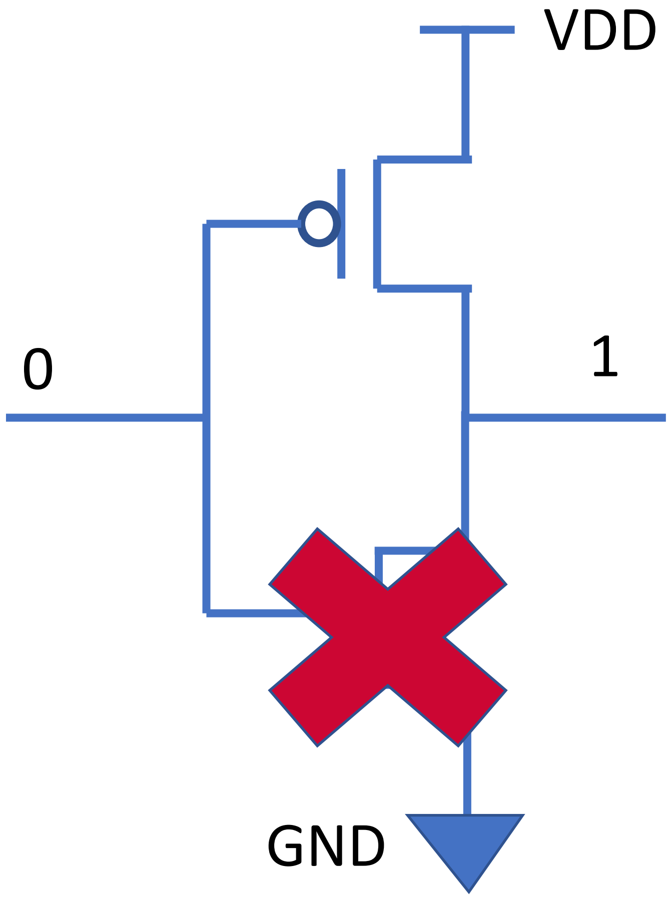
Static Energy

- Also called “Idle” or “Leakage” energy
- Energy consumed due to leakage current even when device is off
- Example: Inverter



Static Energy

- Example: Inverter
- Even the lower transistor that is turned off has some "leakage" current that flows through it



Static Power/Energy

- **Static power is proportional to the static (leakage) current and the voltage**

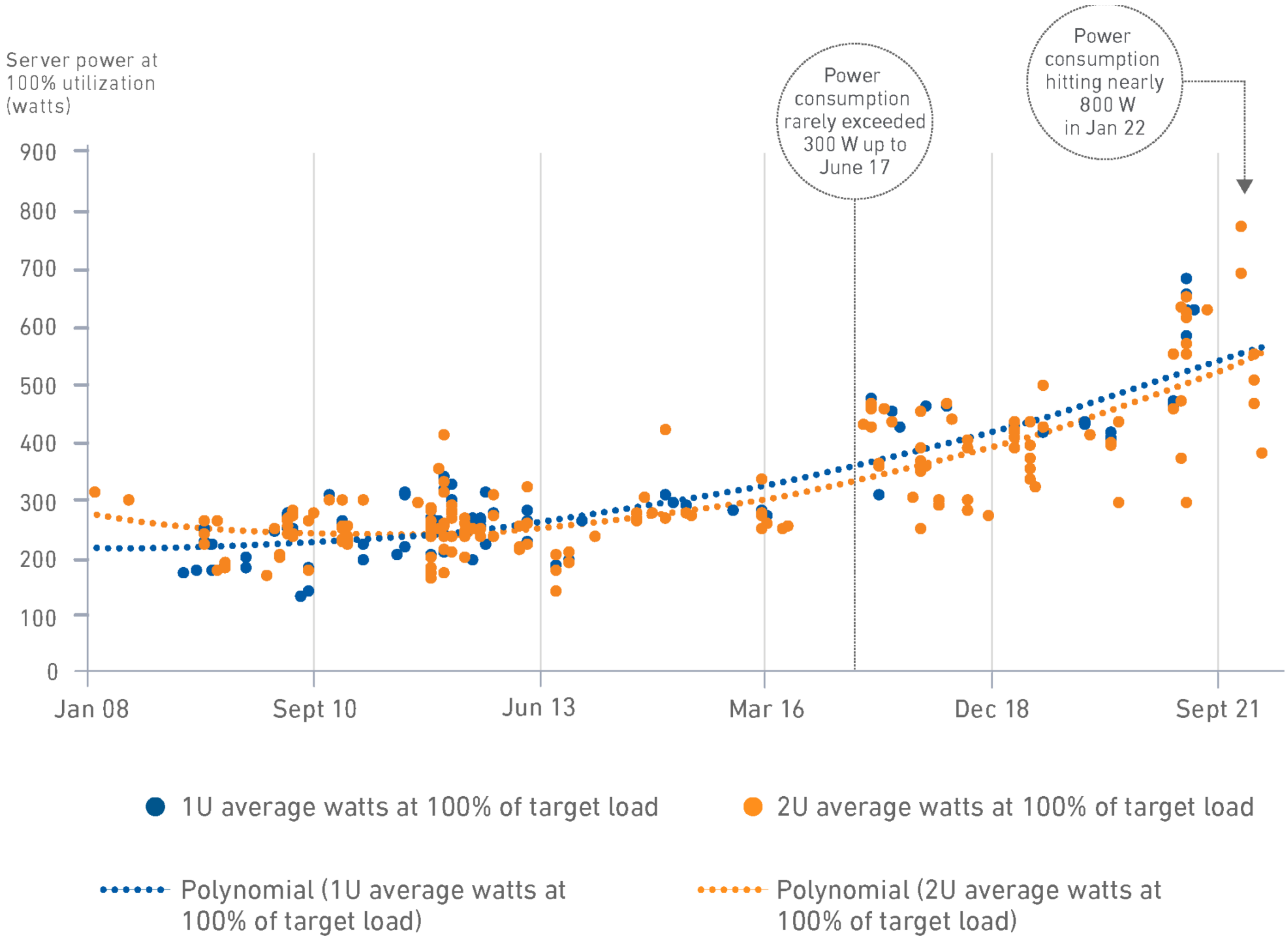
$$Power_{static} \propto Current_{static} \times Voltage$$

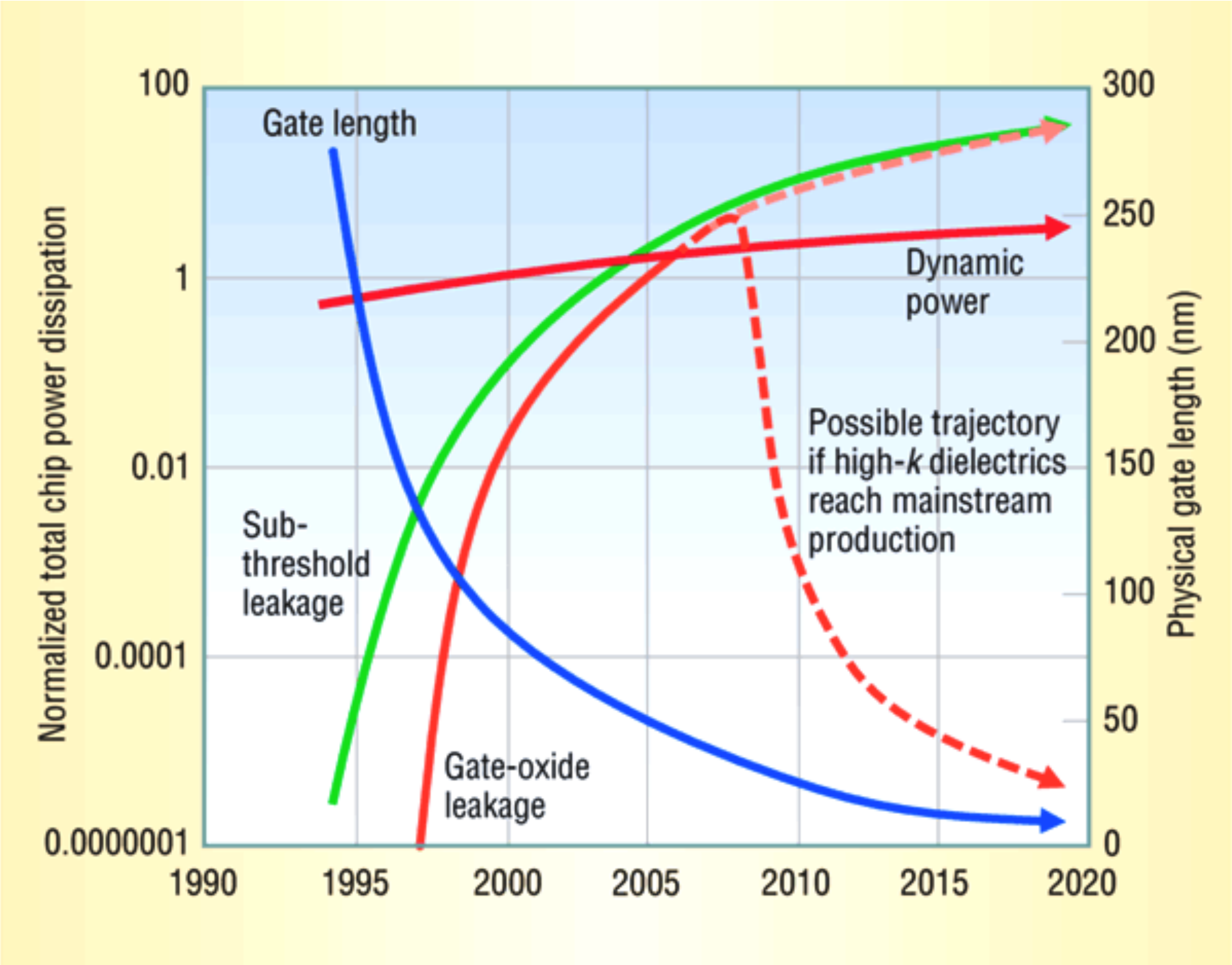
- **Since current increases with the number of devices, static power also proportionally increases with number of devices (and area)**
- **Static power has been increasing over time (as a fraction of total power) due to increasing transistor counts**
  - Could be even 50% or higher of total power if large parts of the chip aren't used
- **Some structures are dominated by static power since they are mostly idle**
  - Example: Large SRAM caches that need to be powered on to preserve stored values
- **Static energy is proportional to static power and time**

Reducing Static Power/Energy

$$Power_{static} \propto Current_{static} \times Voltage$$

- **Static power/energy can also be reduced by lowering operating voltage**
- **Power gating can be used to turn off power from unused components. However, that results in loss of hardware state**
  - Power-gating SRAM caches will lose all the values stored there (backed up in main memory)
  - For volatile memories (e.g., SRAM, DRAM), powering off loses all stored data
  - Non-volatile memories can retain data even when losing power
    - However, they typically are much slower and have lower bandwidth compared to volatile memories

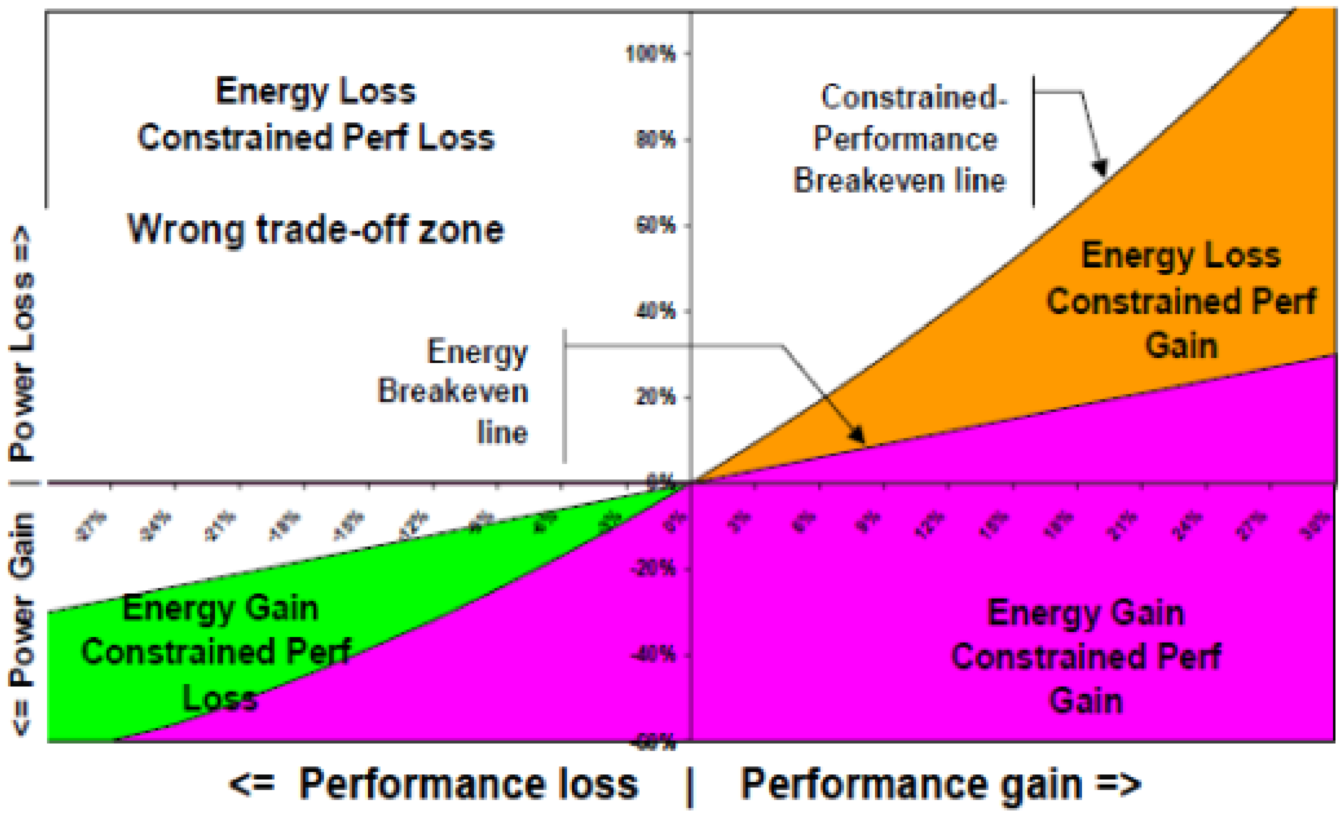

Server power at 100% utilization (watts)
900
800
700
600
500
400
300
200
100
0
Jan 08
Sept 10
Jun 13
Mar 16
Dec 18
Sept 21
Power consumption rarely exceeded 300 W up to June 17
Power consumption hitting nearly 800 W in Jan 22
1U average watts at 100% of target load
2U average watts at 100% of target load
Polynomial (1U average watts at 100% of target load)
Polynomial (2U average watts at 100% of target load)


Gate length
Normalized total chip power dissipation
Physical gate length (nm)
100
1
0.01
0.0001
0.0000001
300
250
200
150
100
50
0
1990
1995
2000
2005
2010
2015
2020
Dynamic power
Possible trajectory if high-*k* dielectrics reach mainstream production
Sub-threshold leakage
Gate-oxide leakage

Optimizing for Performance vs. Energy

- **Optimizing for Performance:**
  - ➢An architectural mechanism is good for performance/energy saving if it is better than DVFS
  - ➢Cube Law: 1% performance for 3% power
- **Optimizing for Energy:**
  - ➢An architectural mechanism is good for energy if it increases performance more than it increases power
  - ➢Energy = Power x Time

    = Power / Performance



Gochman et al. Figure 1

Designing for Energy Efficiency: Principles

- **Execute fewer instructions per program. Examples:**
  - Better branch predictors reduce extra instructions on the wrong path
  - Reduce updates to stack pointer: Avoid SP updates for corresponding PUSH and POP operations
  - Reduce updates to program counter: Only update for taken branches and control transfer instructions
- **Reduce transistor switching activity**
  - Use structures with lower complexity, e.g., RAM instead of CAM
- **Only turning on necessary components**
  - Domain-Specific Accelerators: Next week's topic.

Reading Assignments

- ARCH Chapter 1.1, 1.4, 1.5 (Read)
- ARCH Chapter 1.6 (Skim)
- Gochman, et al., "The Intel Pentium M Processor: Microarchitecture and Performance," Intel Technology Journal, 2003 (Skim)