University of Pittsburgh

CS 1541 Introduction

Technology Advances

Wonsun Ahn
Department of Computer Science
School of Computing and Information

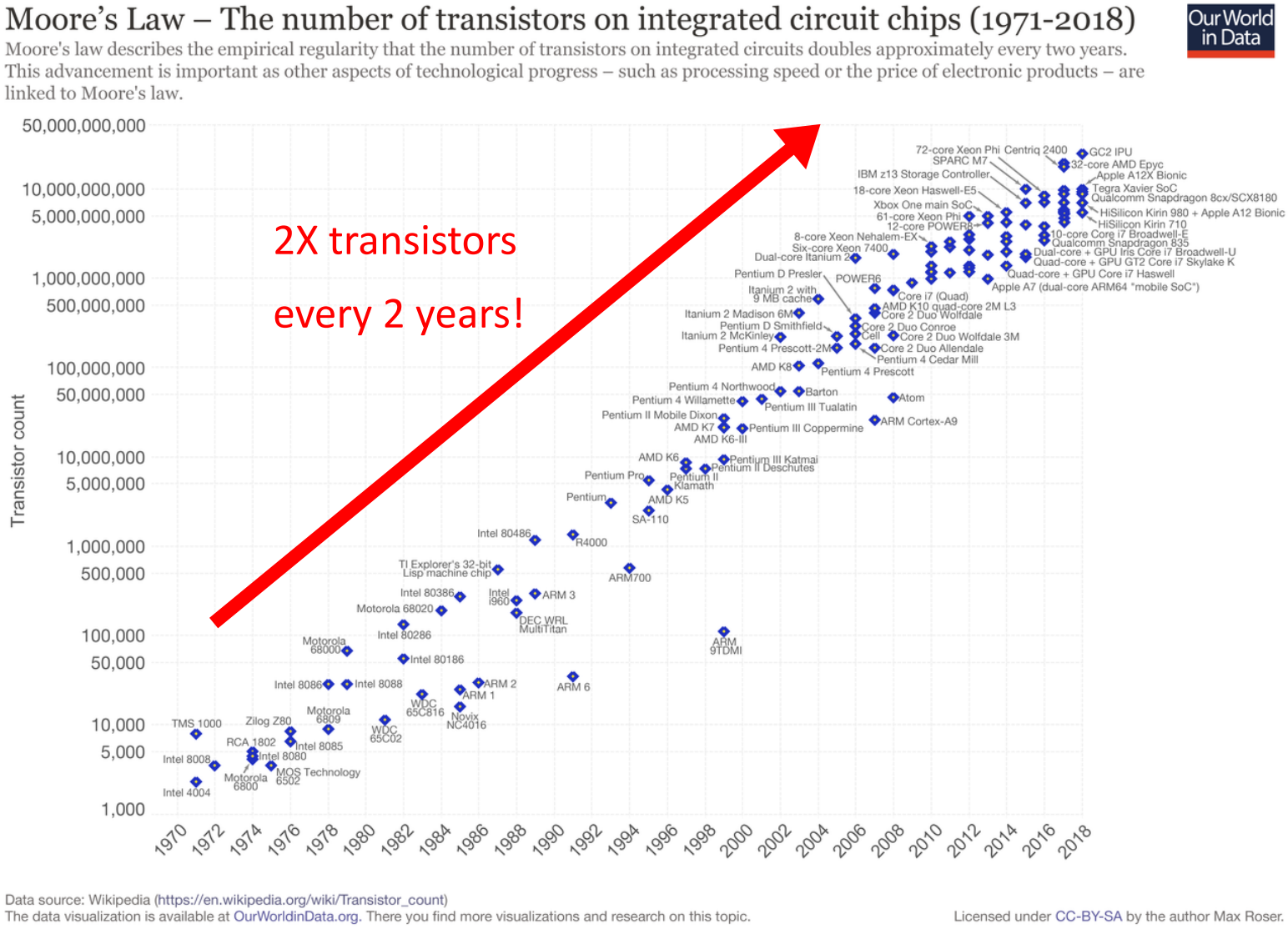
Technology Advances

Advances in Technology

- Technology has been advancing at lightning speed
- Architecture and IT as a whole were beneficiaries

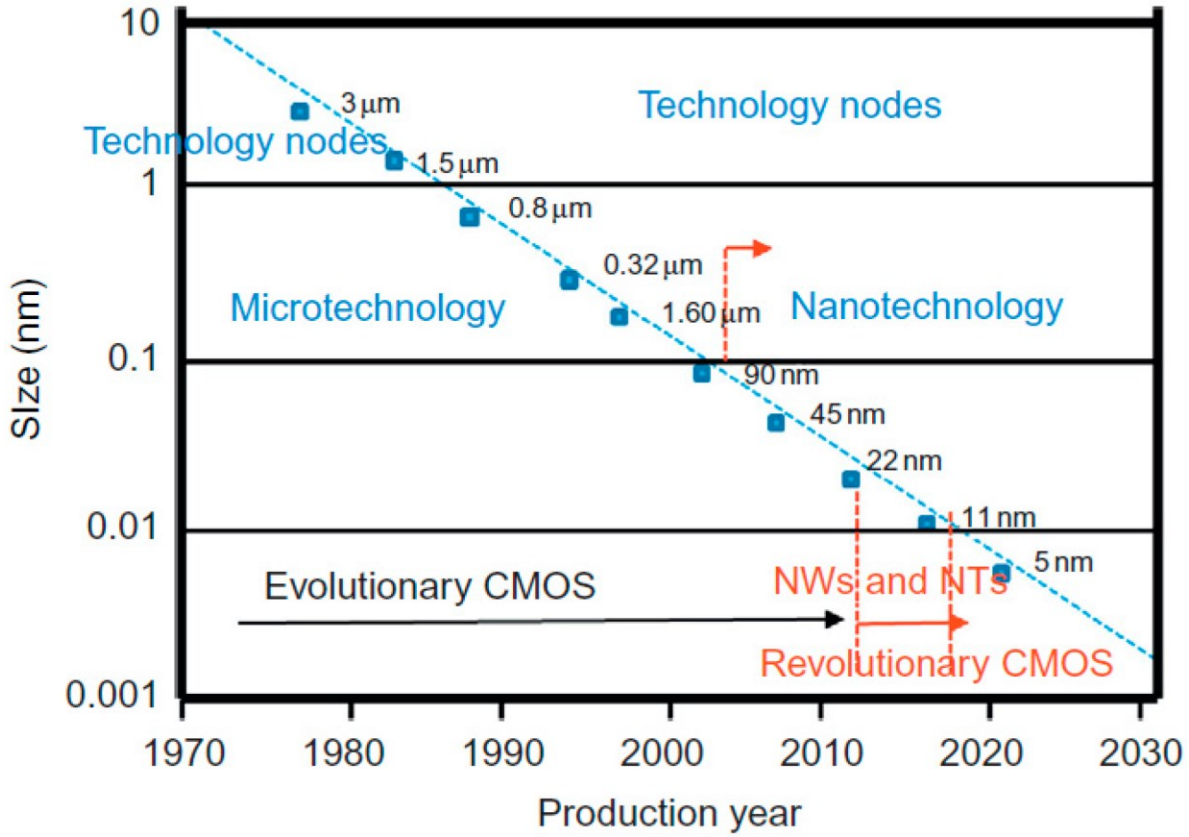
- Technology advance is summarized by *Moore's Law*
  - You probably heard of it at some point. Something about …
  - "X doubles every 18-24 months at constant cost"
- Is X:
  - CPU performance?
  - CPU clock frequency?
  - Transistors per CPU chip?
  - Area of CPU chip?

Moore's Law



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.

Miniaturization of Transistors



Data source: Radamson, H.H.; He, X.; Zhang, Q.; Liu, J.; Cui, H.; Xiang, J.; Kong, Z.; Xiong, W.; Li, J.; Gao, J.; Yang, H.; Gu, S.; Zhao, X.; Du, Y.; Yu, J.; Wang, G. Miniaturization of CMOS. *Micromachines* **2019**, *10*, 293.

- Moore's Law has been driven by transistor miniaturization
  - CPU chip area hasn't changed much

Future of Moore's Law

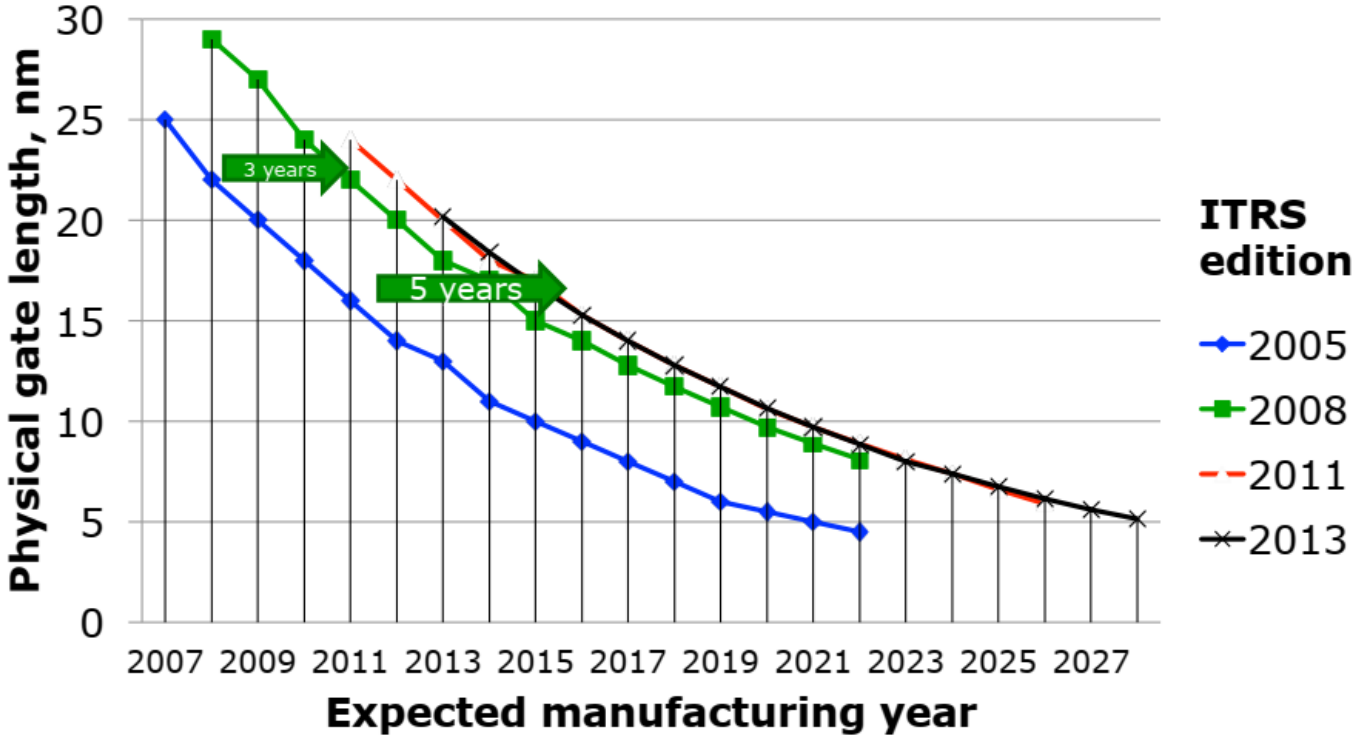
- The semiconductor industry has produced roadmaps
  - Semiconductor Industry Association (SIA): 1977~1997
  - International Technology Roadmap for Semiconductors (ITRS): 1998~2016
  - International Roadmap for Devices and Systems (IRDS): 2017~Present
- IRDS Lithography Projection (2020)

Year of Production	2018	2020	2022	2025	2028	2031	2034
Technology Node (nm)	7	5	3	2.1	1.5	1.0	0.7

  - Looks like Moore's Law will continue into foreseeable future
  - IRDS does not project significant increase in CPU chip size
  - Increases in transistors will come from *transistor density*

IRDS isn't Perfect

- ITRS (predecessor of IRDS) has made corrections before



  - After all, you are trying to predict the future
  - But architects rely on the roadmap to design future processors

Moore's Law and Performance

- Million-dollar question:
  Did Moore's Law result in higher performance CPUs?

- We will do a Zoom breakout room session

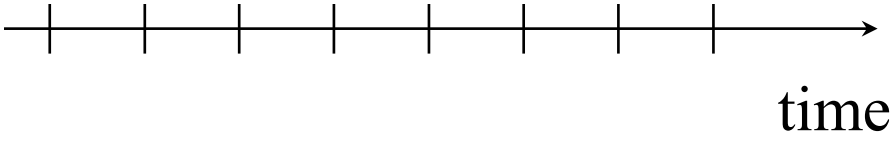
1. Get to know each other (5 mins):
   - Introduce yourself and say one fun thing you did over winter
2. And then answer the following questions (5 mins):
   - When you decide on a CPU for your laptop, what number(s) do you look at to measure how fast the CPU is?
   - Are CPUs getting faster using that measure?
3. After 10 minutes, we will share discussions with class

Are CPUs getting Faster?

Measure of Performance	Is it Getting Better?
Response time (for an app)	Depends (generally increasing)
Number of cores	Generally upwards trajectory
Power draw / Thermals	Getting lower
Clock speed	Same
Cache size	Increasing

Components of Execution Time

- Processor activity happens on clock "ticks" or cycles



  - On each tick, bits flow through logic gates and are latched

- Execution time = $\frac{\text{seconds}}{\text{program}}$

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{seconds}}{\text{cycle}}$$

$$= \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

Improving Execution Time

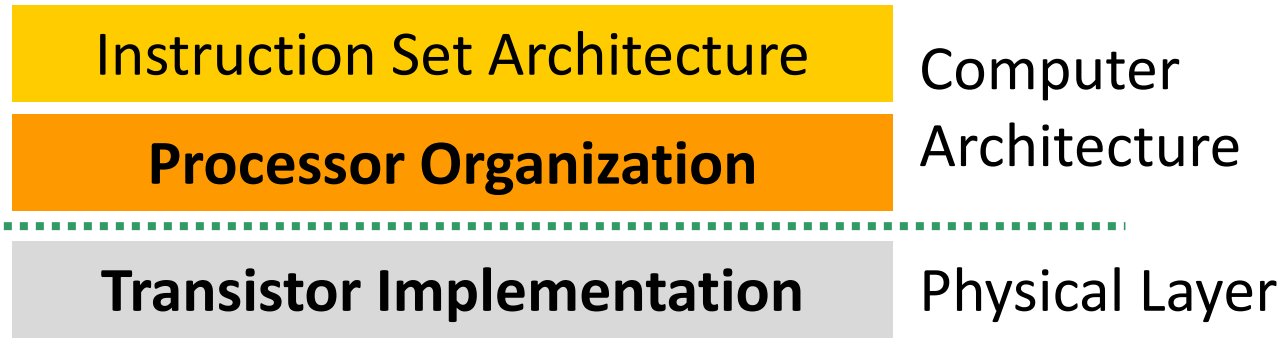
$$\frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

- Improving $\frac{\text{seconds}}{\text{cycle}}$:
  - Clock frequency = $\frac{\text{cycles}}{\text{second}}$ = reverse of $\frac{\text{seconds}}{\text{cycle}}$
  - Higher clock frequency (GHz) leads to shorter exec time
- Improving $\frac{\text{cycles}}{\text{instruction}}$:
  - Also known as CPI (Cycles Per Instruction)
  - IPC (Instructions Per Cycle) = $\frac{\text{instructions}}{\text{cycles}}$ = reverse of $\frac{\text{cycles}}{\text{instructions}}$
  - Higher IPC leads to shorter execution time
- Improving $\frac{\text{instructions}}{\text{program}}$:
  - Less instructions leads to shorter execution time
  - ISAs that do a lot of work with one instruction shortens time

Moore's Law and Performance

- Million-dollar question:
  Did Moore's Law result in higher performance CPUs?

- Law impacts both architecture and physical layers



  - Processor Organization: many more transistors to use in design
  - Transistor Implementation: smaller, more efficient transistors

Moore's Law Impact on Architecture

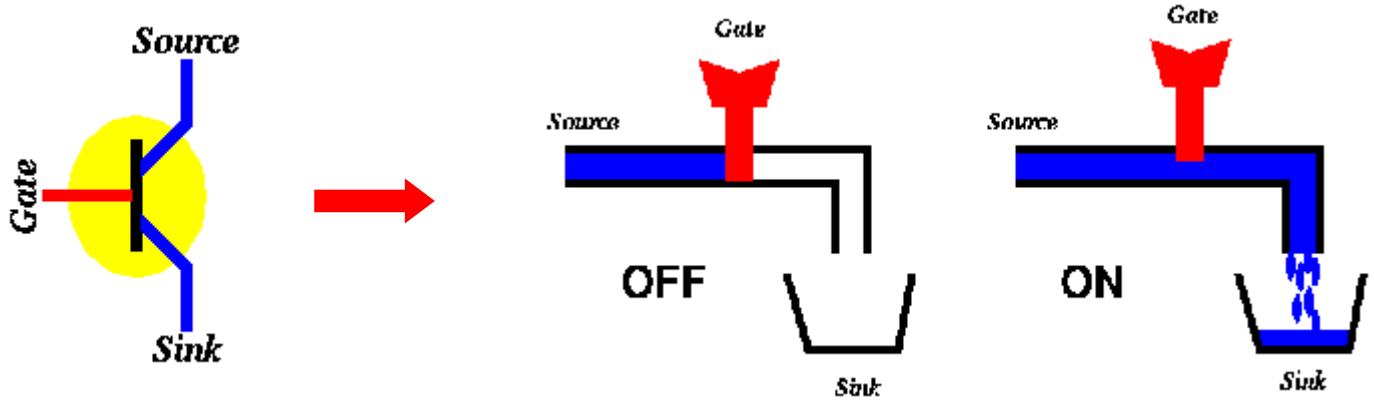
- So where did architects use all those transistors?
- Well, we will learn this throughout the semester ☺
  - Pipelining
  - Parallel execution
  - Prediction of values
  - Speculative execution
  - Memory caching
  - In short, they were used to improve frequency or IPC
- Let's go on to impact on the physical layer for now

Moore's Law Impact on Physical Layer

- CPU frequency is also impacted by transistor speed
  - As well as how many transistors are in between clock ticks (which is determined by processor organization)
- So did Moore's Law result in faster transistors?
  - In other words, are smaller transistors faster?

Speed of Transistors

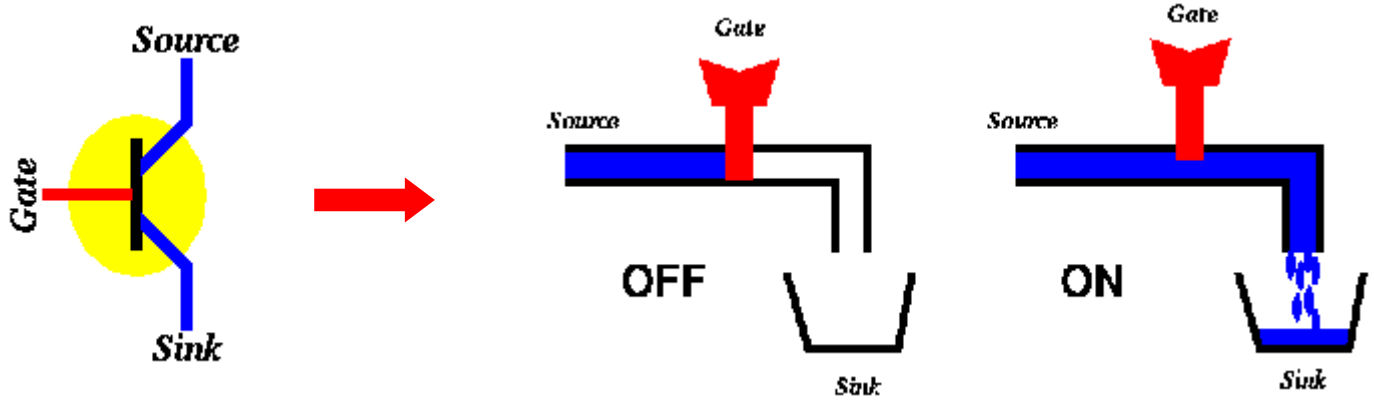
- Transistor 101: Transistors are like faucets!



- To make a transistor go fast, do one of the following:
  - Reduce distance from source to sink (*channel length*) ⇩
  - Reduce bucket size (*capacitance*) ⇩
  - Increase water pressure (*supply voltage*) ⇧

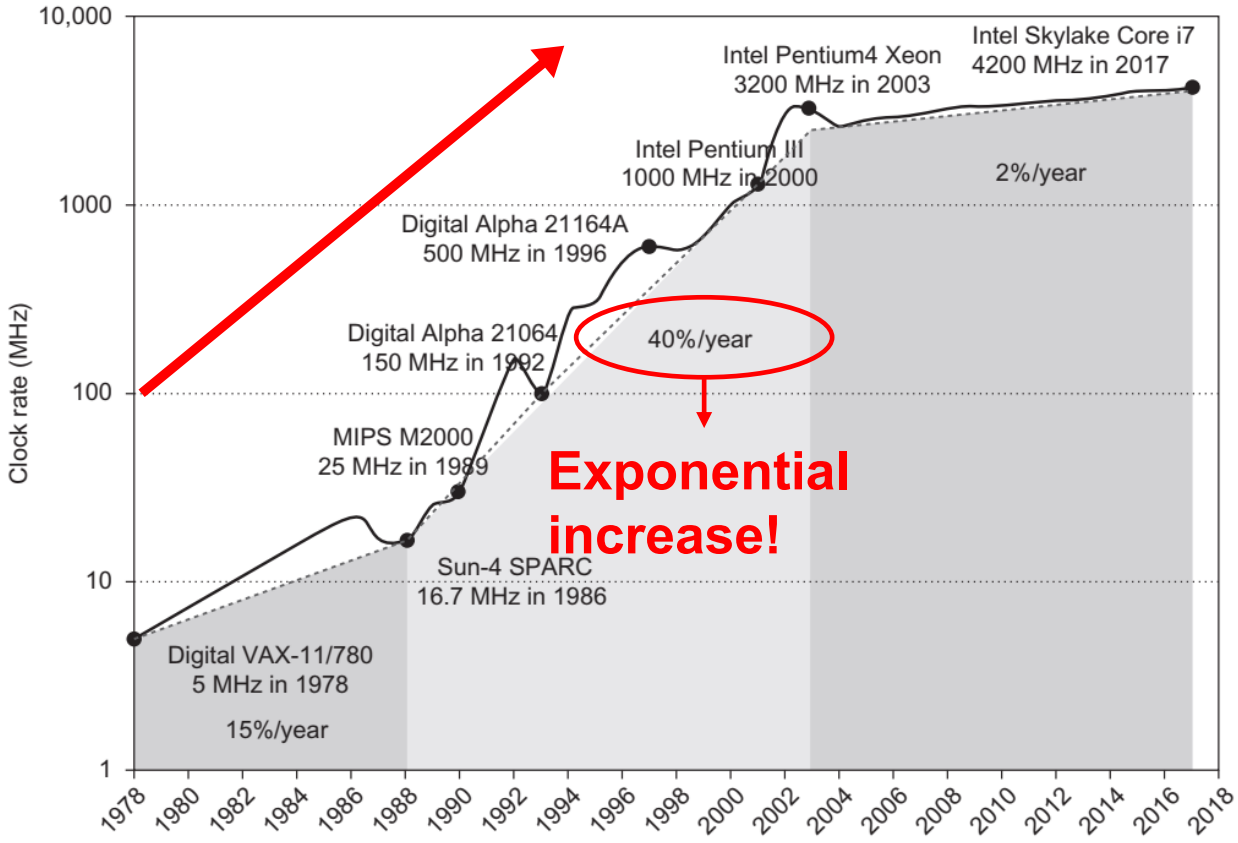
Smaller Transistors are Faster!

- Transistor 101: Transistors are like faucets!



- When a transistor gets smaller:
  - *Channel length* (channel resistance) is reduced ⇩
  - *Capacitance* is reduced ⇩
- So, given the same *supply voltage*, smaller is faster!
- So, did Moore's Law enjoy faster and faster frequencies?

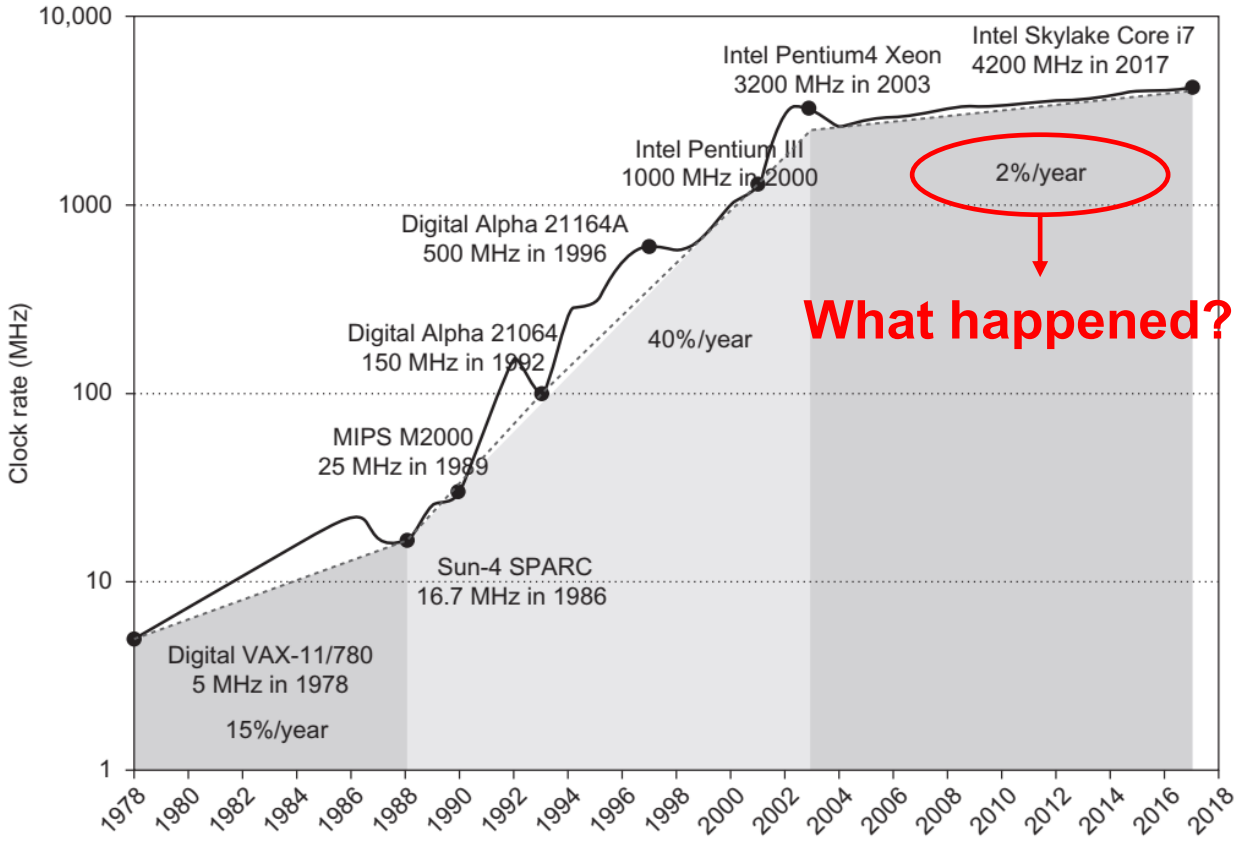
Yes, for a while …



Source: Computer Architecture, A Quantitative Approach (6th ed.) by John Hennessy and David Patterson, 2017

- Improvements in large part due to transistors
  - Processor design also contributed but we'll discuss later

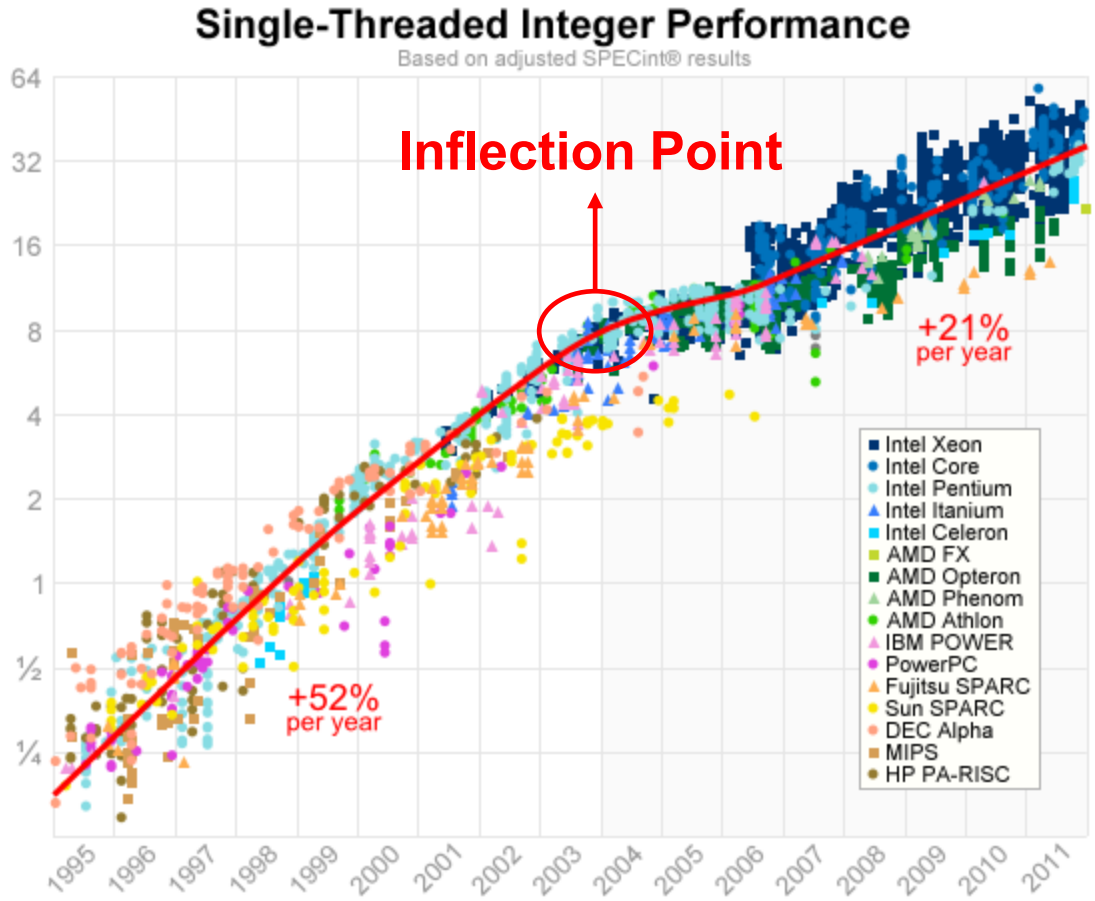
But not so much lately



Source: Computer Architecture, A Quantitative Approach (6th ed.) by John Hennessy and David Patterson, 2017

- Suddenly around 2003, frequency scaling stops

Dent in CPU Performance



Source: https://preshing.com/20120208/a-look-back-at-single-threaded-cpu-performance/

- This caused a big dent in CPU performance at 2003
- Improvements henceforth only came from architecture
  - From improvements to IPC (instructions per cycle)

So What Happened? TDP.

- *TDP* (*Thermal Design Power*):
  - Maximum power (heat) that CPU is designed to generate
  - Capped by the amount of heat cooling system can handle
  - Cooling system hasn't improved much over generations

- CPU Power = A * N * CFV^2 must be < TDP
  - A = Activity factor (% of transistors with activity)
  - N = Number of transistors
  - C = Capacitance
  - F = Frequency
  - V = Supply Voltage

- What happens to each factor with Moore's Law?

TDP and Moore's Law

- CPU Power $\propto$ A * N * CFV^2 with Moore's Law
  - A = Activity factor
  - N = Number of transistors ($\propto$ 1/transistor size^2) ⇧ ⇧
  - C = Capacitance ($\propto$ transistor size) ⇩
  - F = CPU frequency ($\propto$ 1/transistor size) ⇧
  - V = CPU Supply Voltage

- Decrease in C cannot offset increases in N and F
  - Power increases quadratically with reductions in transistor size
  - That means F (frequency) needs to be decreased to meet TDP

Q) So how did CPU frequency keep increasing up to 2003?

A) By maintaining power through reductions in Voltage ⇩

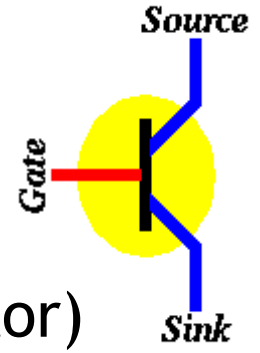
Dennard Scaling

- By reducing CPU Supply Voltage ∝ transistor size
- CPU Power ∝ A * N * CFV^2 with Moore's Law
  - A = Activity factor
  - N = Number of transistors (∝ 1/transistor size^2) ⇧ ⇧
  - C = Capacitance (∝ transistor size) ⇩
  - F = CPU frequency (∝ 1/transistor size) ⇧
  - V = CPU Supply Voltage (∝ transistor size) ⇩ ⇩
- Factors balance each other out to keep power constant
  - Note that reducing V (∝ transistor size) has a quadratic effect
- Dennard Scaling: Above recipe for scaling up frequency, while reducing supply voltage to keep power constant

Dennard Scaling and V_{th}

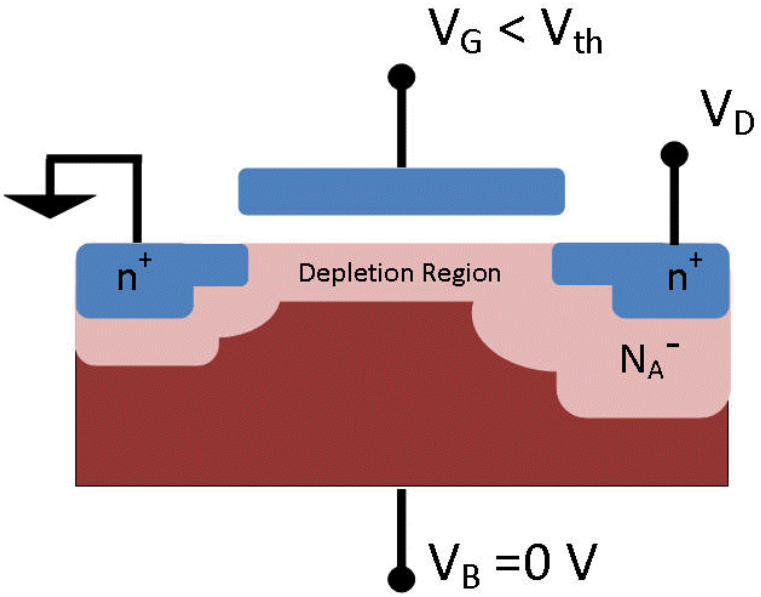
- So, it's that easy? Just reduce V until you meet TDP?
- No, it's not that simple ☹.
- Reducing V_{dd} (supply voltage) affects CPU operation
  - As V_{dd} is reduced, CPU becomes slower and slower
  - Eventually, CPU stops working altogether
- CPU (specifically transistors) needs redesigning
  - V_{th} (threshold voltage) needs to be reduced along with V_{dd}
  - To understand this, we need a 101 on MOSFETs

MOSFET 101

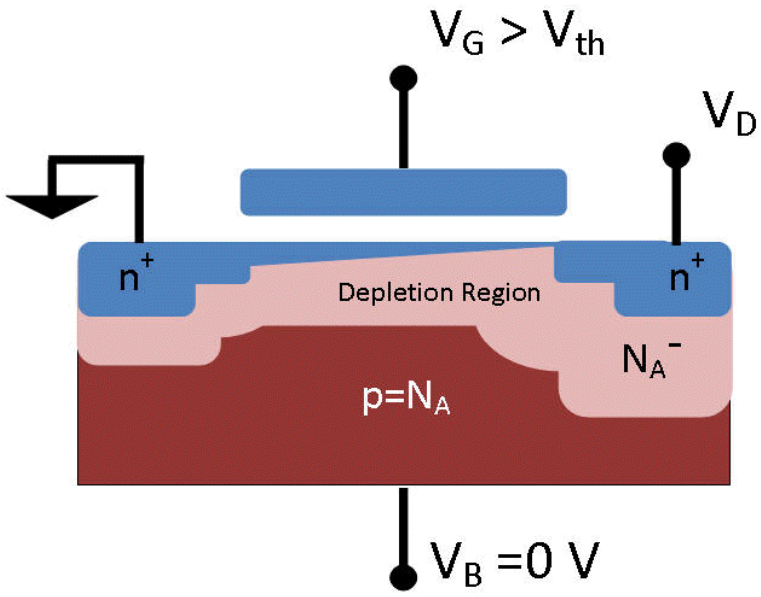


- MOSFET (Metal Oxide Silicon Field Effect Transistor)

[A MOSFET transistor switched off]



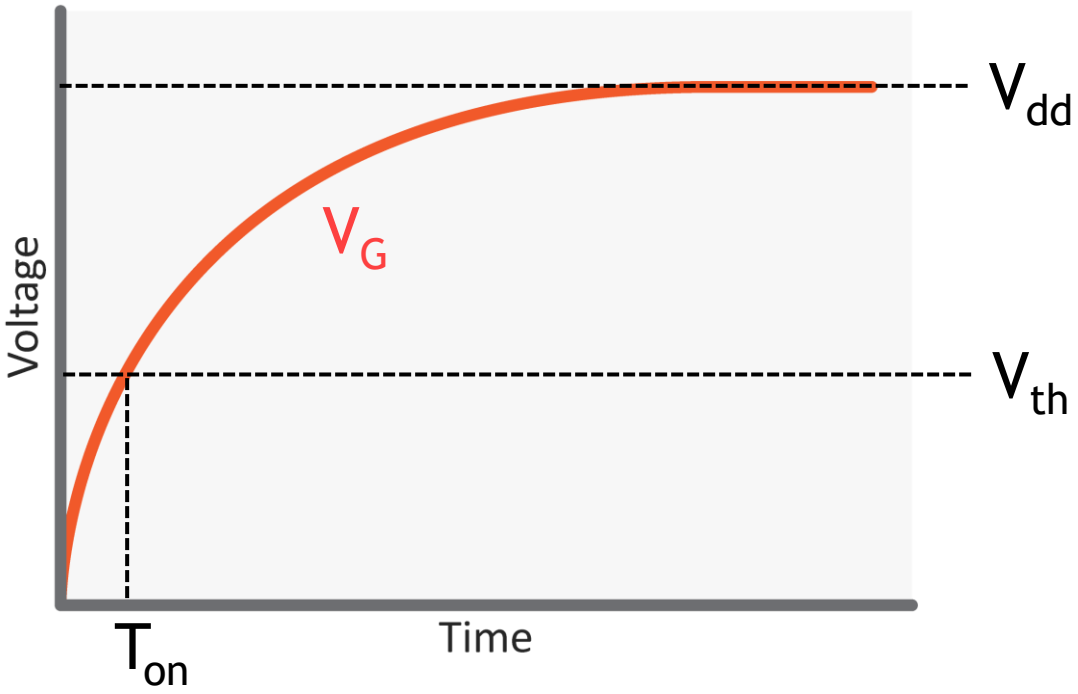
[A MOSFET transistor switched on]



- Gate is switched on when V_G reaches a threshold V_{th}
  - By creating a channel in depletion region through field effect
  - V_{th}: threshold voltage (minimum voltage to create channel)

MOSFET 101

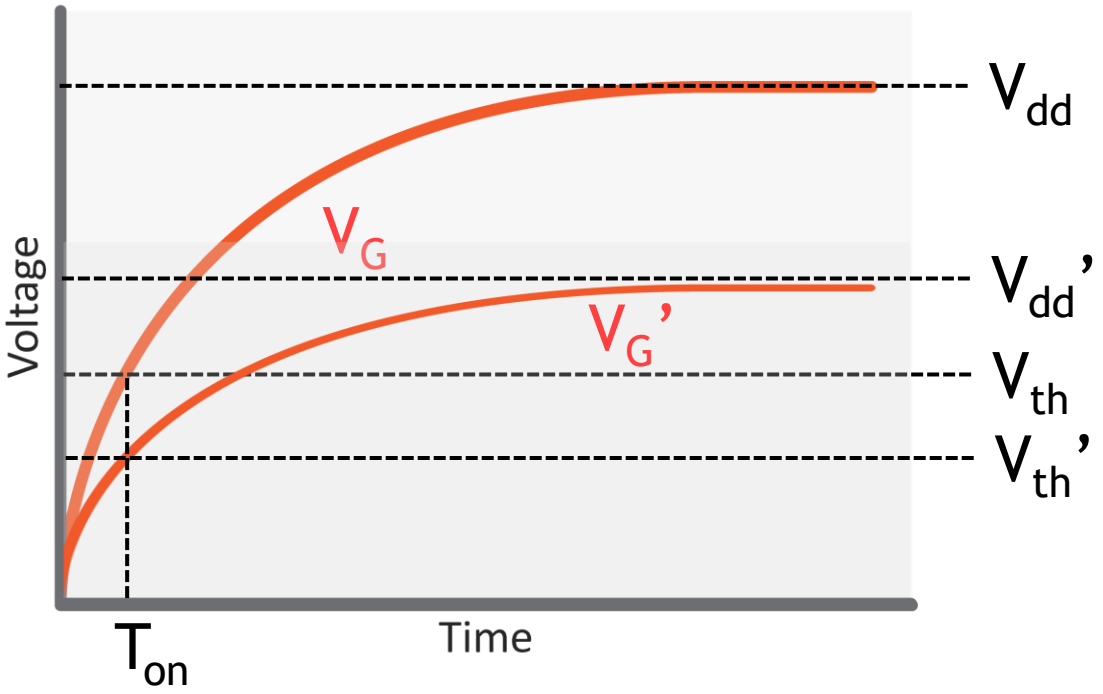
- RC charging curve of V_G



- Speed (T_{on}) is determined by V_{dd} if V_{th} is fixed
  - V_{dd} is the CPU supply voltage (the water pressure)
  - If V_{dd} is lower, V_G will reach V_{th} more slowly (low pressure)

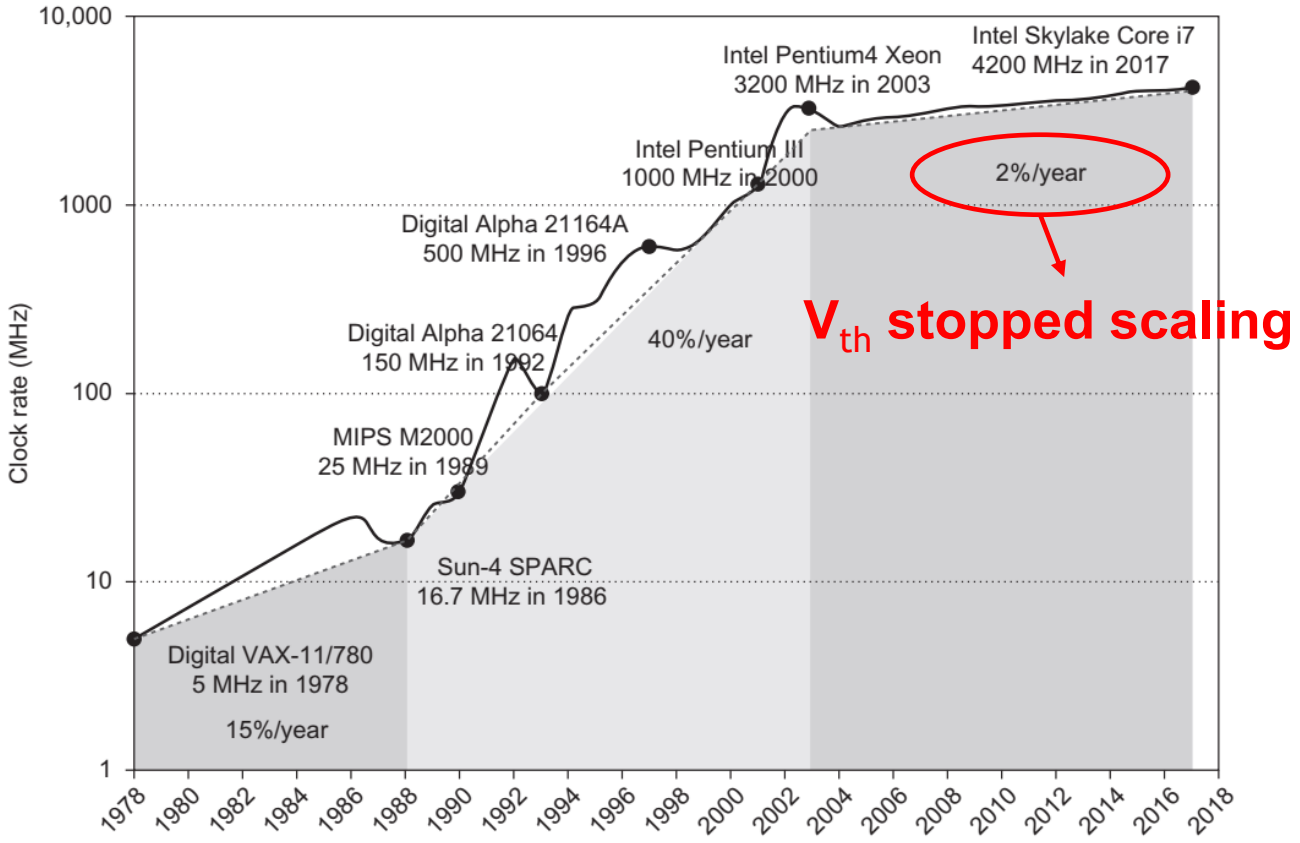
MOSFET 101

- RC Charging Curve of V_G



- Speed (T_{on}) is maintained while reducing V_{dd} to V_{dd}', only if V_{th} is also reduced to V_{th}'

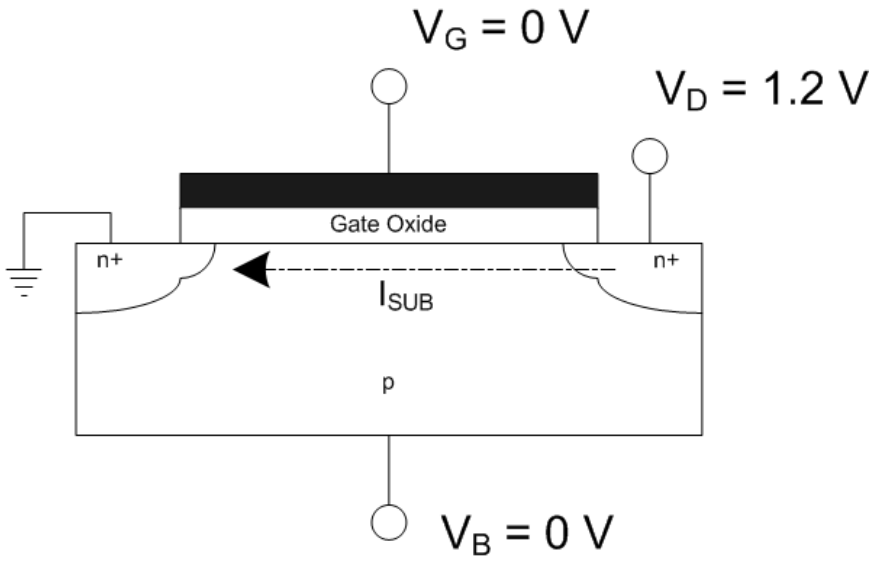
End of Dennard Scaling



- And around 2003 is when Dennard Scaling ended

Limits to Dropping V_{th}

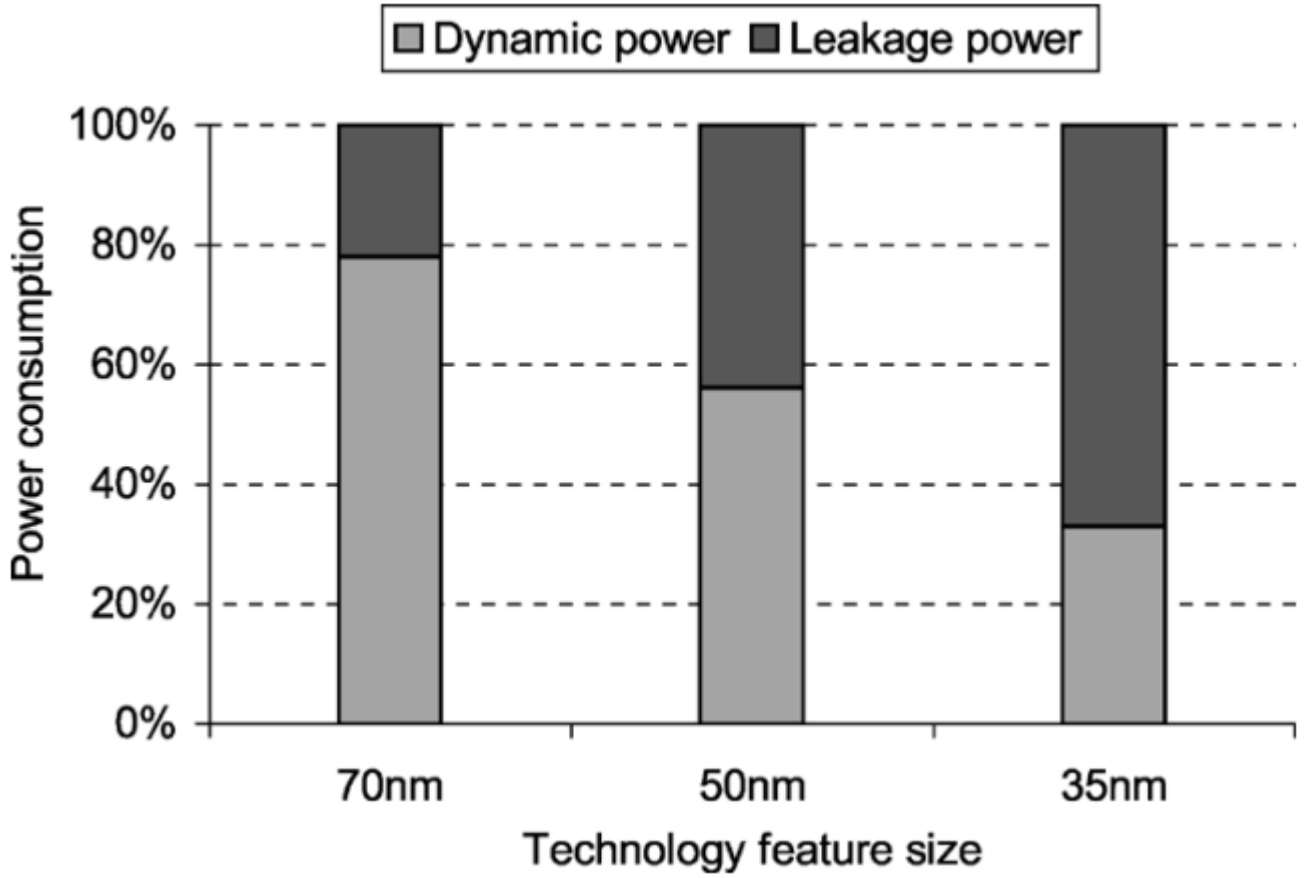
- Subthreshold leakage
  - Transistor leaks current even when gate is off ($V_G = 0$)



  - This leakage current translates to leakage power
  - Leakage worsens when V_{th} is dropped (related to oxide thickness)

Leakage Power across Generations

- Leakage power has increased across technology nodes



Source: L. Yan, Jiong Luo and N. K. Jha, "Joint dynamic voltage scaling and adaptive body biasing for heterogeneous distributed real-time embedded systems," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1030-1041, July 2005

End of Dennard Scaling

- Previous power calculation was incomplete
  - CPU power is the sum of both dynamic and leakage power
- $Power_{CPU} \propto Power_{dynamic} + Power_{leakage}$
  - $Power_{dynamic} \propto A * N * CFV_{dd}^2$
  - $Power_{leakage} \propto f(N, V_{dd}, V_{th}) \propto N * V_{dd} * e^{-Vth}$
  - Leakage worsens *exponentially* when V_{th} is dropped
  - Catch-22: when dropping V_{th}, $Power_{dynamic}$ ⇩ but $Power_{leakage}$ ⇧⇧
- V_{th} can't be reduced further, so V_{dd} can't be reduced
- Dennard Scaling relies on reducing V_{dd}, so it's the end

“Dark Silicon” Rears its Head

- What happens to frequency without Dennard Scaling?
- $Power_{dynamic}$ ($\propto$ A * N * CFV^2) + $Power_{leakage}$ ($\propto$ N * V * e^{-Vth})
  - A = Activity factor
  - N = Number of transistors ($\propto$ 1/transistor $size^2$) ⇧ ⇧
  - C = Capacitance ($\propto$ transistor size) ⇩
  - V = CPU Supply Voltage ⇔ (Due to fixed V_{th})
  - F = CPU frequency **???**
- To offset N, you actually have to *decrease* F
- Otherwise, if you want to maintain F, must decrease N
  - That is, you cannot power on all the transistors at any given point
  - Dark silicon: situation where chip is only partially powered

Free Ride is Over

- "Free" speed improvements from transistors is over
- Now it's up to architects to improve performance
  - Moore's Law is still alive and well (although slowing down)
  - Architects are flooded with extra transistors each generation
  - But it's hard to even keep them powered without reducing F!
- Now is a good time to discuss technology constraints
  - Since we already mentioned a big one: TDP