

### Technology Advances

# SET THE CANADA TO THE CANADA T

### **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?

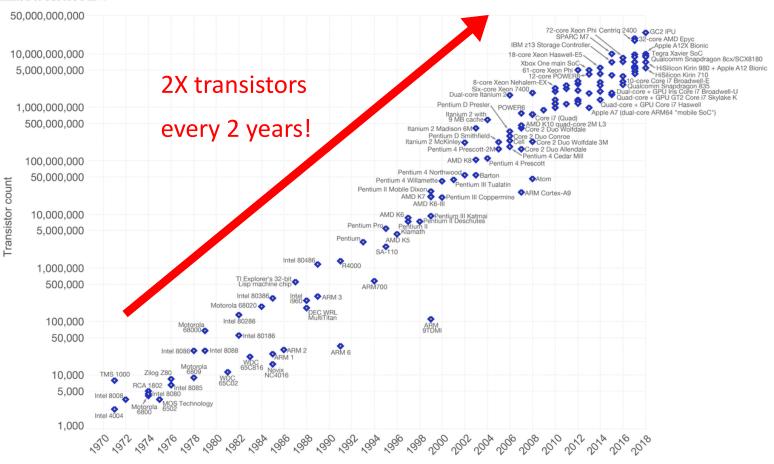




#### Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

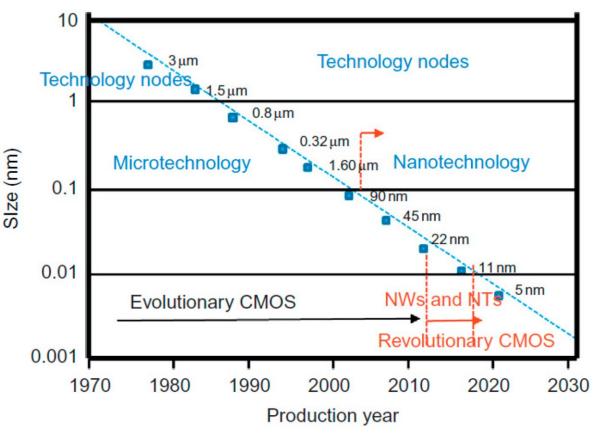


Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



#### 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)

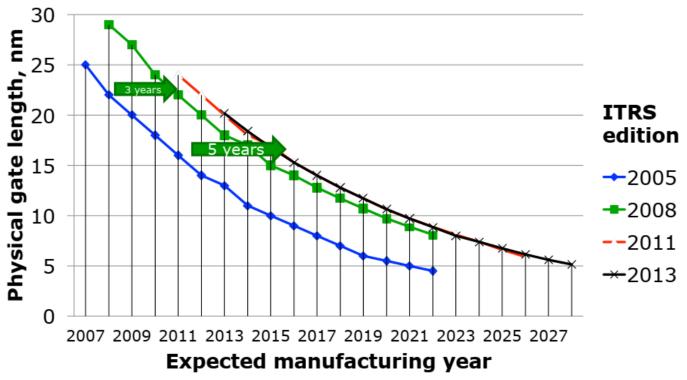
Year of Production	2018	2020	2022	2025	2028	2031	2034
Technology Node (nm)	7	5	3	2.1	1.5	1.0	0.7

- 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

# THE THE PARTY OF T

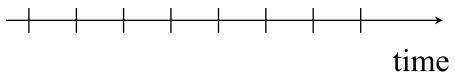
#### Moore's Law and Performance

- Did Moore's Law result in higher performance CPUs?
- When you decide on a CPU for your desktop, what number(s) do you look at to see how fast it is?
- Best way: try running your favorite apps on it.
- Everyone has different favorite apps so instead publish
  - Results from running a suite of benchmark apps (e.g. SPECCPU)
  - Several important components of performance

# SECTION OF THE PROPERTY OF THE

#### 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 =** 
$$\frac{\text{seconds}}{\text{program}}$$

$$\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \quad X \quad \frac{\text{seconds}}{\text{cycle}}$$

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

#### How to improve Execution Time

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

- Reduce seconds / cycle :
   Clock frequency = cycles / second = reverse of seconds / cycle
   Higher clock frequency (GHz) leads to shorter exec time
- **Reduce**  $\frac{\text{cycles}}{\text{instruction}}$ :
  - 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
- Reduce instructions program:
  - Less instructions leads to shorter execution time
  - ISAs that do a lot of work with one instruction shortens time



#### Moore's Law impacts two layers

- Did Moore's Law result in higher performance CPUs?
- Law impacts both architecture and physical layers

Instruction Set Architecture Computer
Processor Organization Architecture

**Transistor Implementation** 

Physical Layer

- 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

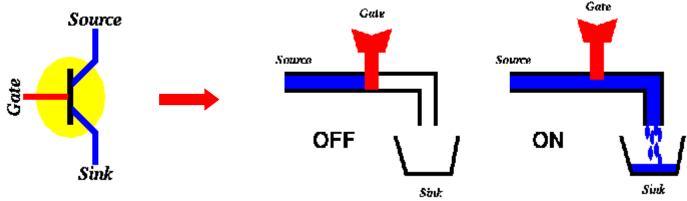
### Moore's Law Impact on Physical Layer

- 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**



Transistors are like faucets!

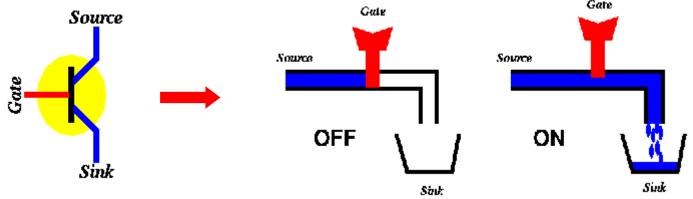


- To make a transistor go fast, do one of the following:
  - Increase water pressure (supply voltage)
  - Change faucet (transistor) design
- Transistor design can be changed in two ways:
  - Increase pipe thickness (reduce channel resistance R)
  - Reduce bucket size (reduce capacitance C)
  - $T_{switch} \propto RC$  (transistor switch delay is proportional to RC)

#### **Smaller Transistors are Faster!**



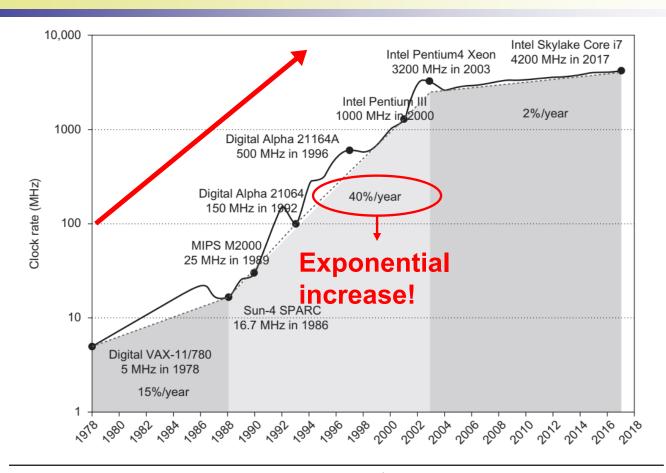
Transistors are like faucets!



- $\blacksquare$  For transistor dimension d (node size)
  - Capacitance  $C \propto d$
  - Channel resistance  $R \propto$  Channel length  $L \propto d$
  - $T_{switch} \propto RC \propto d^2$
- Given the same supply voltage, smaller is faster!
- Did Moore's Law enjoy faster and faster frequencies?

## SEVERSITE OF THE PROPERTY OF T

#### Yes, for a while ...

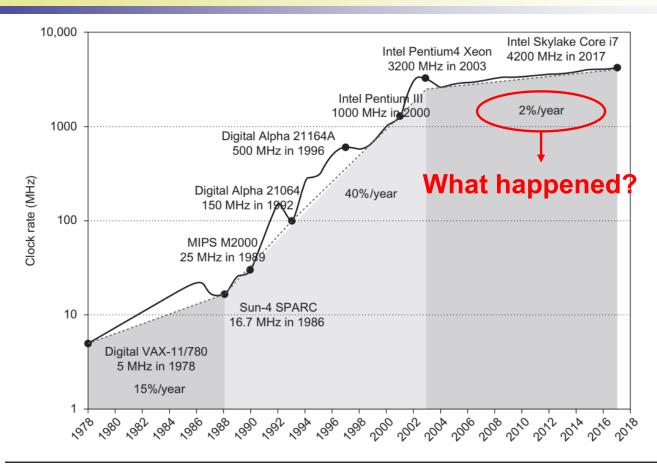


Source: Computer Architecture, A Quantitative Approach (6th ed.) by John Hennessy and David Patterson, 2017

- Due in large part to improvement in transistor speed
  - CPU design (pipelining) contributed but we'll discuss later

# SEVERSITE OF THE PROPERTY OF T

### But not so much lately



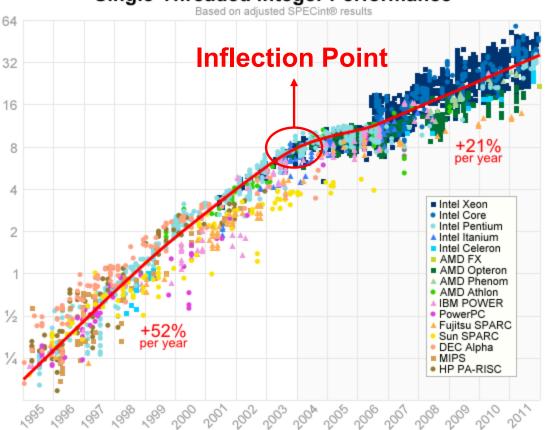
Source: Computer Architecture, A Quantitative Approach (6th ed.) by John Hennessy and David Patterson, 2017

- Suddenly around 2003, frequency scaling stops. Why?
  - Improvements in transistor speed stopped.
  - CPU design (pipelining) has met its limits.

#### Dent in CPU Performance



#### Single-Threaded Integer Performance



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

- This caused a big dent in CPU performance circa 2003
- Improvements henceforth mostly came from IPC

#### Why did frequency scaling stop? 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 \* CFV² must be < TDP</p>
  - 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?

### CPU Power (with Fixed Voltage)

- Change in CPU Power ∝ A \* N \* CFV², with fixed V:
  - A = Activity factor
  - N = Number of transistors  $\propto 1/d^2$  ☆ ☆

  - $F \propto 1/T_{\text{switch}} \propto 1/d^2 \Omega \Omega \Omega \Omega$  (if supply voltage is kept constant)
  - V = Supply Voltage (water pressure)
  - $\rightarrow$  CPU Power  $\propto 1/d^3 \Omega \Omega \Omega$
- Recipe used until 1990's until power started to matter
  - That meant F could not keep increasing while meeting TDP
- Q) So how did CPU frequency keep increasing up to 2003?

# THE CANAL PROPERTY OF THE PARTY OF THE PARTY

### **CPU Power (with Dennard Scaling)**

- Goal: to scale frequency while keeping power constant
- By reducing V proportional to d, then change in CPU Power ∝ A \* N \* CFV² is:
  - A = Activity factor
  - N = Number of transistors  $\propto 1/d^2$  ☆ ☆

  - $F \propto 1/d$  1 (Can scale by 1/d while keeping power constant!)
  - V = Supply Voltage ( $\propto$  d)  $\checkmark$   $\rightarrow$  V<sup>2</sup>  $\checkmark$   $\checkmark$
- Dennard Scaling: Above recipe for scaling up frequency, while reducing supply voltage to keep power constant

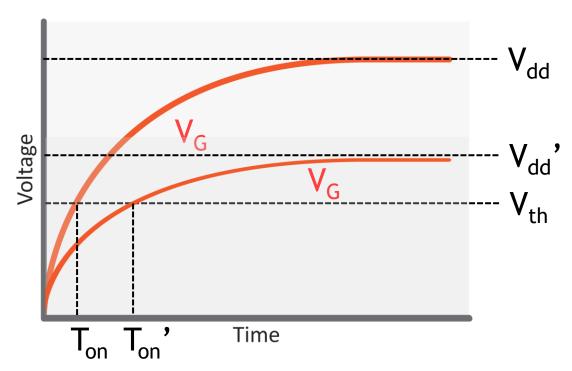
# TO SERVICE OF THE SER

### Dennard Scaling and V<sub>th</sub>

- So, it's that easy? Just reduce V until you meet TDP?
- No, it's not that simple ⊗.
- Reducing V<sub>dd</sub> (supply voltage) affects CPU operation
  - As V<sub>dd</sub> is reduced, transistor becomes slower (lower pressure)
  - Eventually, CPU stops working altogether
- Transistors need redesigning to work at lower voltage
  - V<sub>th</sub> is the threshold voltage when switching happens
  - V<sub>th</sub> needs to be reduced along with V<sub>dd</sub> to maintain speed

#### Speed is determined by V<sub>dd</sub> if V<sub>th</sub> is fixed

RC Charging Curve of V<sub>G</sub> (Gate Voltage)

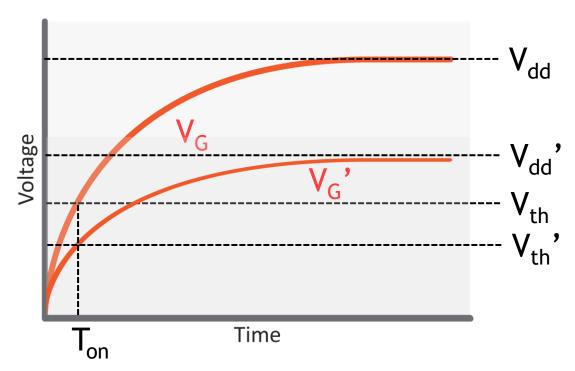


- $V_{dd}$  is high  $\rightarrow V_G$  reaches  $V_{th}$  slowly at  $T_{on}$  (high pressure)
- $V_{dd}$ ' is high  $\rightarrow V_G$  reaches  $V_{th}$  slowly at  $T_{on}$ '(low pressure)

# THI ON THE PROPERTY OF THE PRO

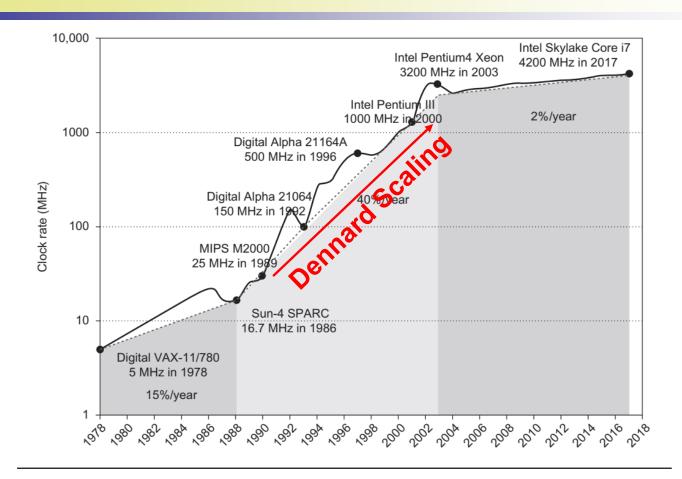
### Speed is maintained with lower V<sub>th</sub>

RC Charging Curve of V<sub>G</sub>



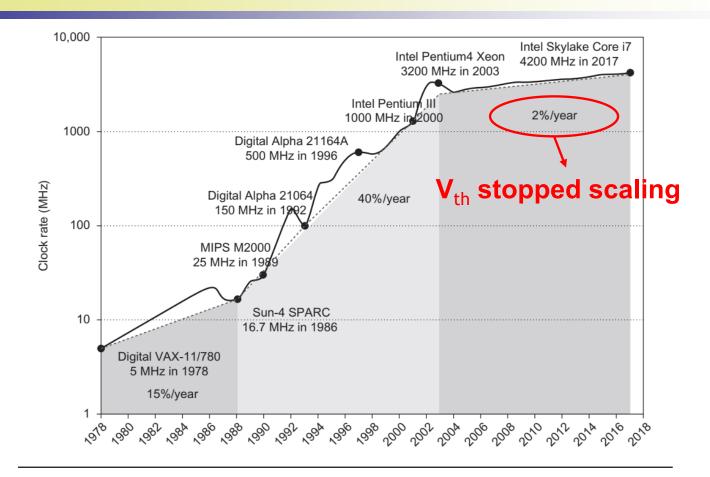
■ Speed  $(T_{on})$  is maintained while reducing  $V_{dd}$  to  $V_{dd}$ , but only if  $V_{th}$  is also reduced to  $V_{th}$ .

#### Dennard Scaling paid dividends for a while



Aided by repeated reductions of V<sub>dd</sub> and V<sub>th</sub>

### Then Dennard Scaling ended at 2003

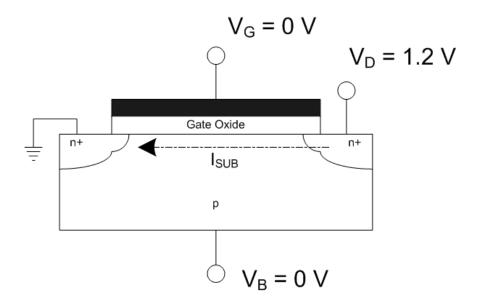


Main reason was that V<sub>th</sub> could no longer be reduced

# SELECTION OF THE PARTY OF THE P

## Limits to Dropping V<sub>th</sub>

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

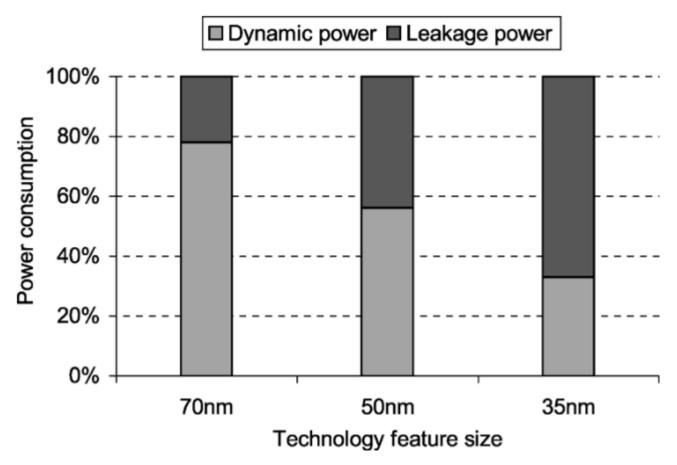


- This leakage current translates to leakage power
- Leakage worsens when V<sub>th</sub> is dropped



### 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

### CPU Power must now include leakage

- Previous power calculation was incomplete
  - CPU power is the sum of both dynamic and leakage power
- Power<sub>CPU</sub> ∝ Power<sub>dynamic</sub> + Power<sub>leakage</sub>
  - Power<sub>dynamic</sub>  $\propto$  A \* N \* CFV<sub>dd</sub><sup>2</sup>
  - Power<sub>leakage</sub>  $\propto$  N \* V<sub>dd</sub> \* e<sup>-Vth</sup>
  - Leakage worsens exponentially when V<sub>th</sub> is dropped
  - Catch-22: when dropping V<sub>th</sub>, Power<sub>dynamic</sub> ↓ but Power<sub>leakage</sub> ☆☆
- $ightharpoonup V_{th}$  can't be reduced further, so  $V_{dd}$  can't be reduced
- Dennard Scaling relies on reducing V<sub>dd</sub>, so it's the end

### **CPU Power (End of Dennard Scaling)**

- What happens to frequency without Dennard Scaling?
- Power<sub>dynamic</sub> ( $\propto$  A \* N \* CFV<sup>2</sup>) + Power<sub>leakage</sub> ( $\propto$  N \* V \* e<sup>-Vth</sup>)
  - A = Activity factor
  - N = Number of transistors ( $\propto 1/d^2$ ) ① ①
  - C = Capacitance (∝ d) ↓
  - $V = Supply Voltage \Leftrightarrow (Due to fixed V_{th})$
  - F = 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

# THE TOTAL PROPERTY OF THE PARTY OF THE PARTY

#### 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 (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