# The Beauty and Joy of Computing

UC Berkeley EECS
Sr Lecturer SOE
Dan Garcia

## Lecture #8
## Concurrency

**Quest** (first exam) in **in 2 days!!**
In this room!

## "KOOMEY'S LAW" – EFFICIENCY 2X EVERY 18 MO

Prof Jonathan Koomey looked at 6 decades of data and found that energy efficiency of computers doubles roughly every 18 months. This is even more relevant as battery-powered devices become more popular. Restated, it says that for a fixed computing load, the amount of battery you need drops by half every 18 months. This was true before transistors!

www.technologyreview.com/computing/38548/

# Concurrency & Parallelism, 10 mi up…

**Intra-computer**

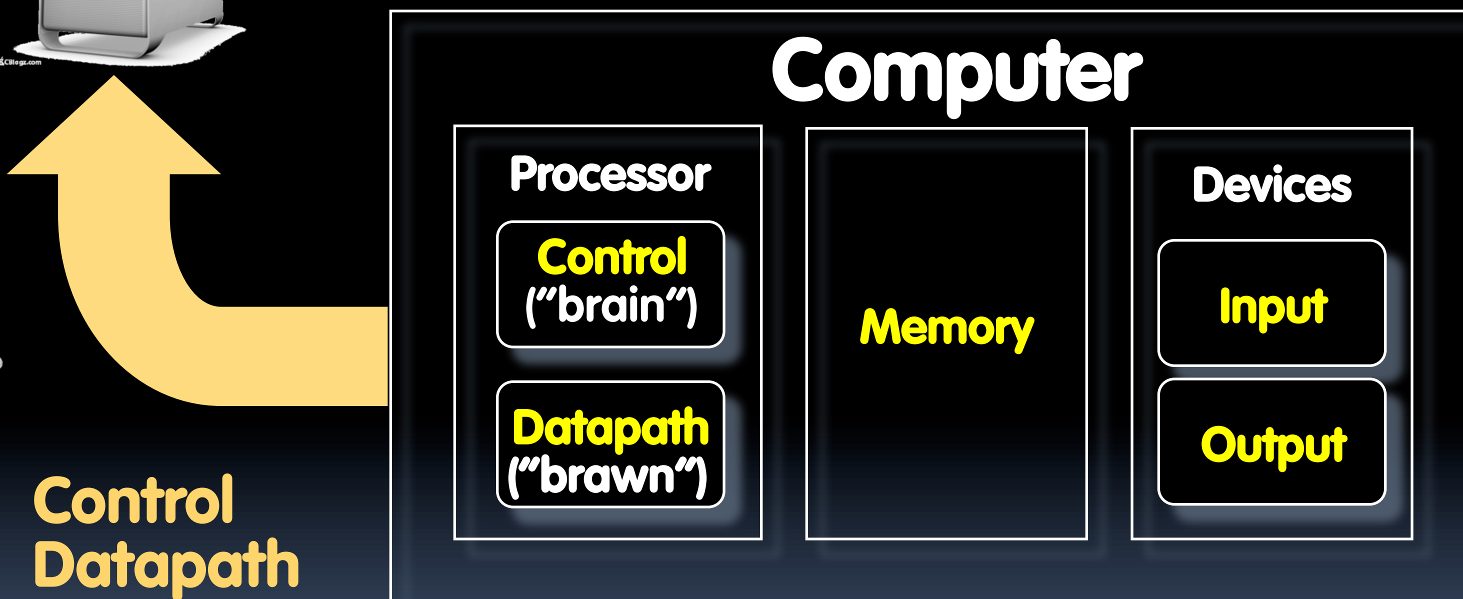
- Today's lecture
- Multiple computing "helpers" are cores <u>within one machine</u>
- Aka "multi-core"
  - Although GPU parallism is also "intra-computer"

**Inter-computer**

- Week 12's lectures
- Multiple computing "helpers" are <u>different machines</u>
- Aka "distributed computing"
  - Grid & cluster computing

# Anatomy: 5 components of any Computer

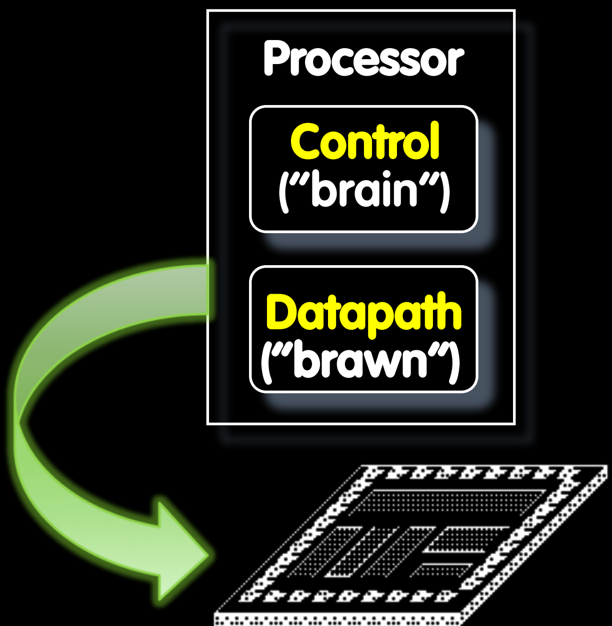
John von Neumann invented this architecture



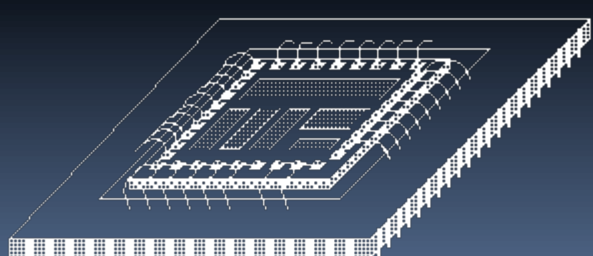

a) Control
b) Datapath
c) Memory
d) Input
e) Output

**What causes the most headaches for SW and HW designers with multi-core computing?**

# But what is INSIDE a Processor?




Bare Processor Die



Chip in Package

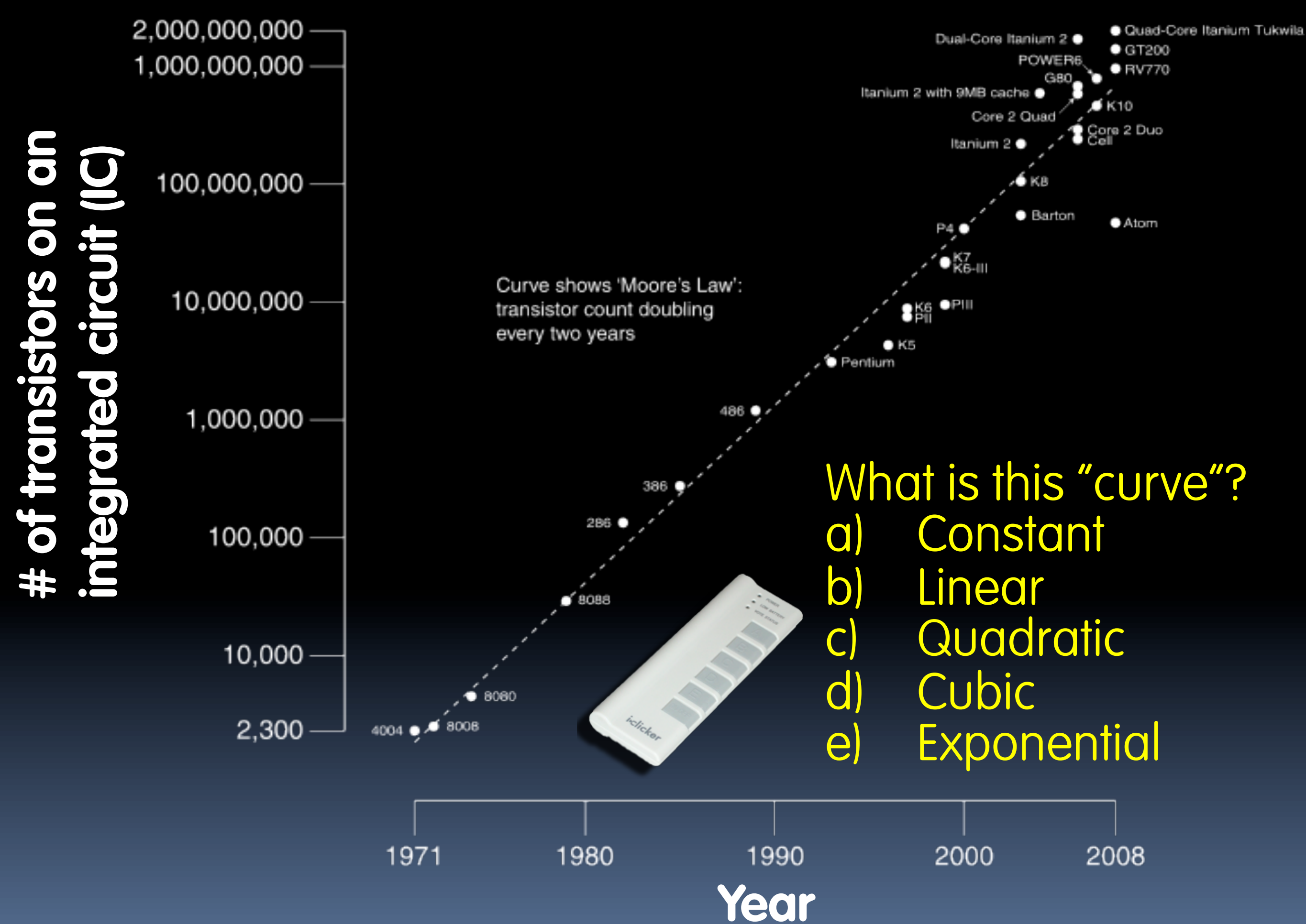
- Primarily Crystalline Silicon
- 1 mm – 25 mm on a side
- 2009 "feature size" (aka process) ~ 45 nm = 45 x $10^{-9}$ m (then 32, 22, and 16 [by yr 2013])
- 100 - 1000M transistors
- 3 - 10 conductive layers
- "CMOS" (complementary metal oxide semiconductor) - most common

- Package provides:
  - spreading of chip-level signal paths to board-level
  - heat dissipation.
- Ceramic or plastic with gold wires.

Garcia

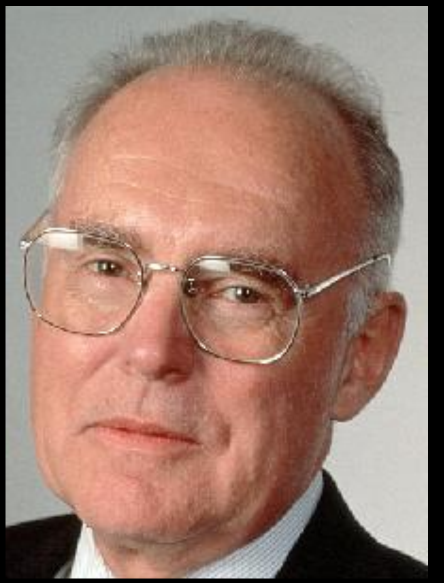
# Moore's Law

## Predicts: 2X Transistors / chip every 2 years



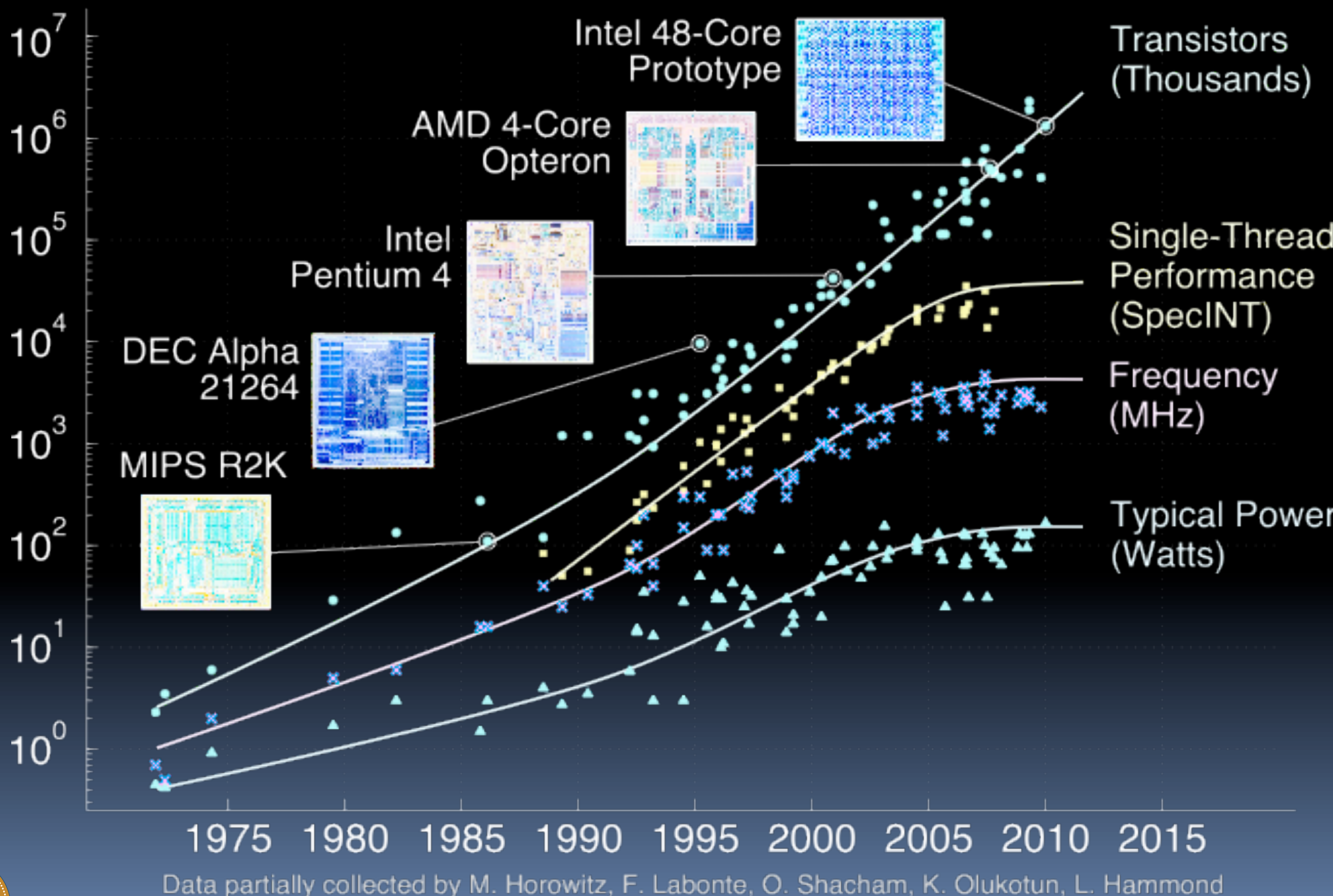

What is this "curve"?

a) Constant
b) Linear
c) Quadratic
d) Cubic
e) Exponential



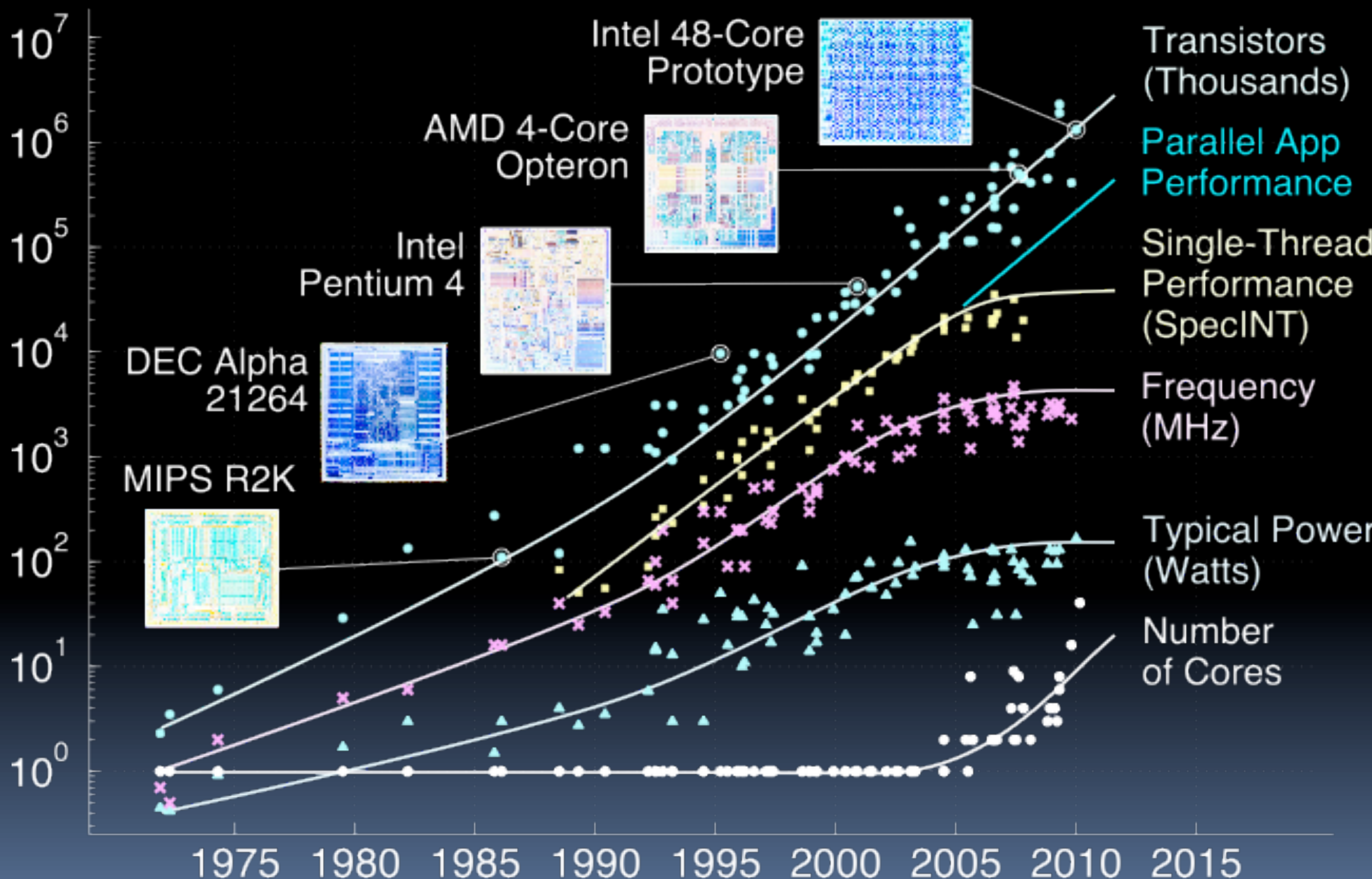
Gordon Moore
Intel Cofounder
B.S. Cal 1950!

# Moore's Law and related curves



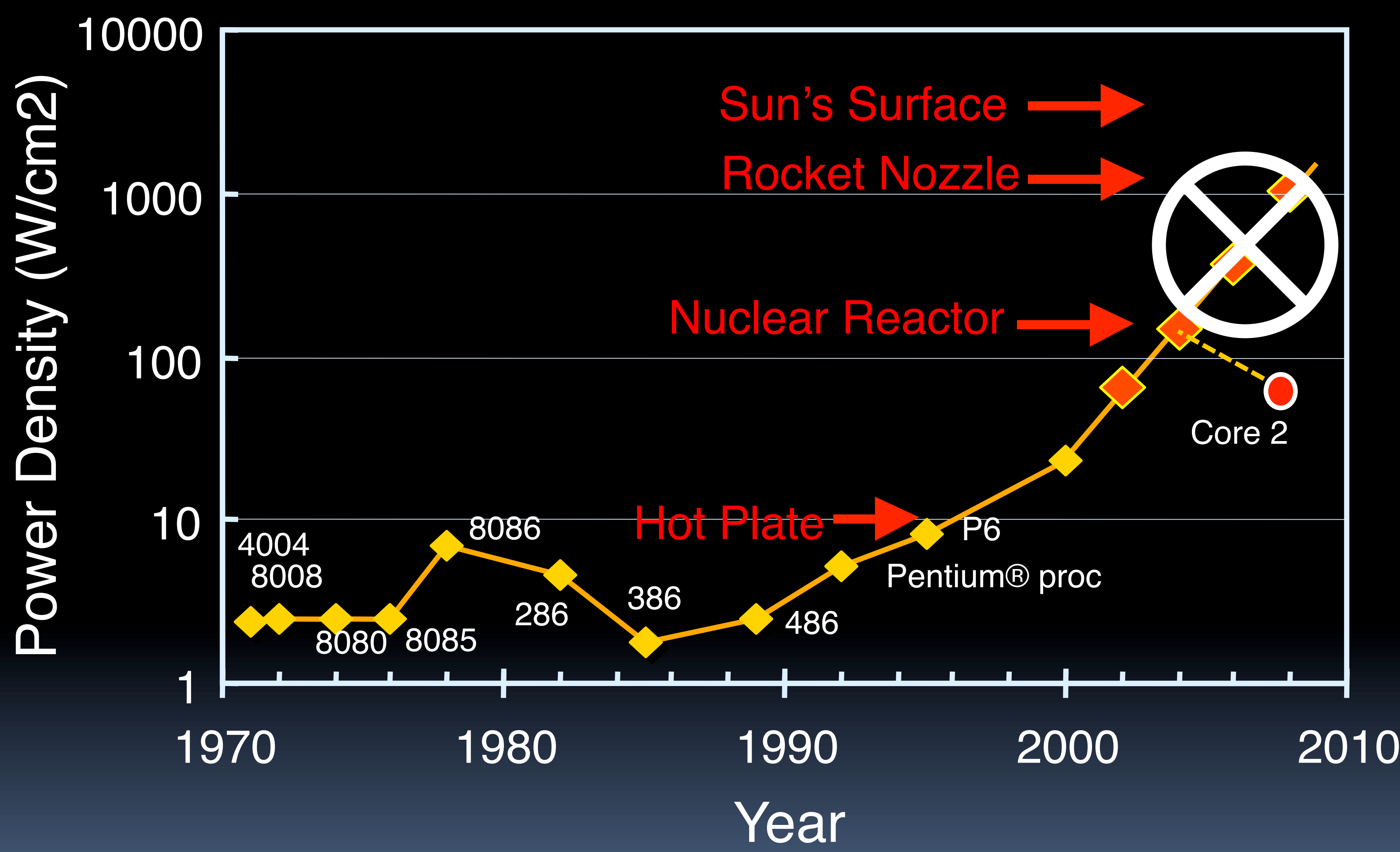

Garcia

# Moore's Law and related curves




Data partially collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond

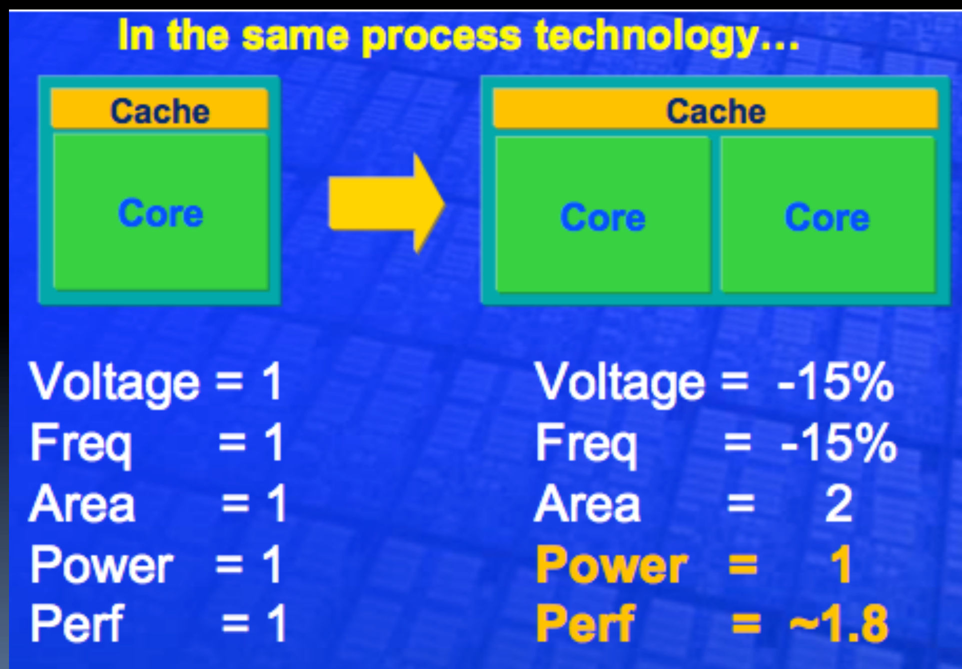
# Power Density Prediction circa 2000




Source: S. Borkar (Intel)

# Going Multi-core Helps Energy Efficiency

- **Power of typical integrated circuit ~ C $V^2$ *f***
  - C = Capacitance, how well it "stores" a charge
  - V = Voltage
  - *f* = frequency. I.e., how fast clock is (e.g., 3 GHz)



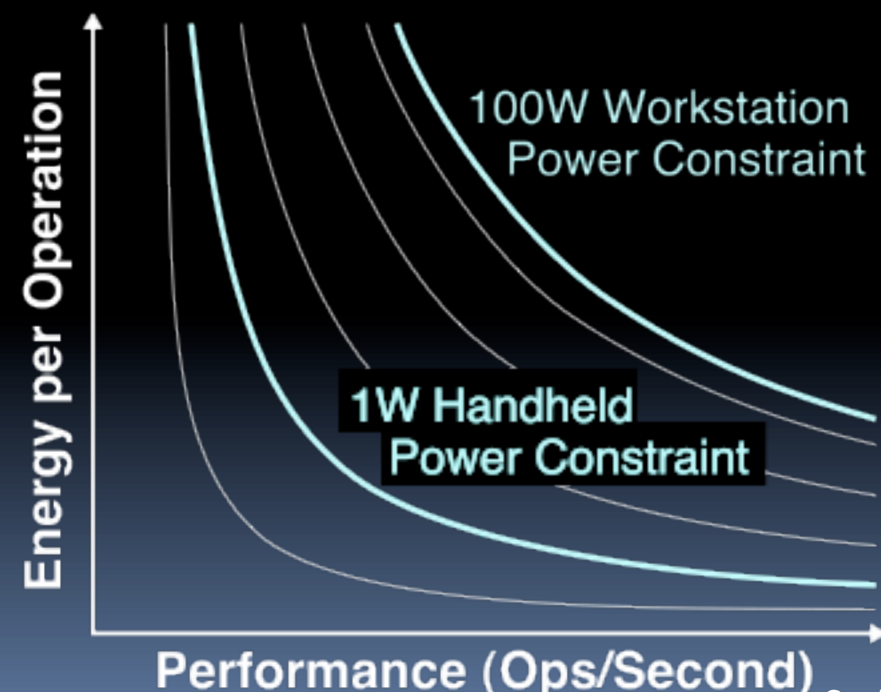

William Holt, HOT Chips 2005

Activity Monitor (on the lab Macs) shows how active your cores are

# Energy & Power Considerations

$$\text{Power} = \frac{\text{Energy}}{\text{Second}} = \frac{\text{Energy}}{\text{Op}} \times \frac{\text{Ops}}{\text{Second}}$$

| Power | Energy |
| --- | --- |
| Chip Packaging | Battery Life |
| Chip Cooling | Electricity Bill |
| System Noise | Mobile Device Weight |
| Case Temperature | |
| Data-Center Air Conditioning | |




Courtesy: Chris Batten

# Parallelism again? What's different this time?

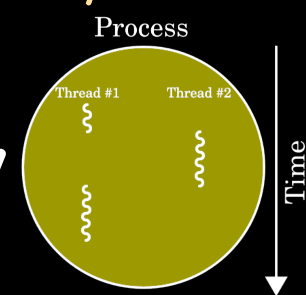
*"This shift toward increasing parallelism is not a triumphant stride forward based on breakthroughs in novel software and architectures for parallelism; instead, this plunge into parallelism is actually a retreat from even greater challenges that thwart efficient silicon implementation of traditional uniprocessor architectures."*

**– Berkeley View, December 2006**

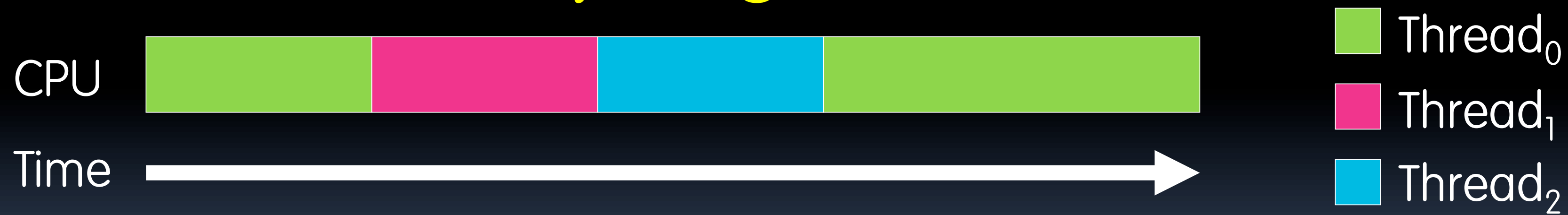
- HW/SW Industry bet its future that breakthroughs will appear before it's too late

# Background: Threads

- A ***Thread*** **stands for "thread of execution", is a single stream of instructions**
  - A program / process can split, or fork itself into separate threads, which can (in theory) execute simultaneously.
  - An easy way to describe/think about parallelism



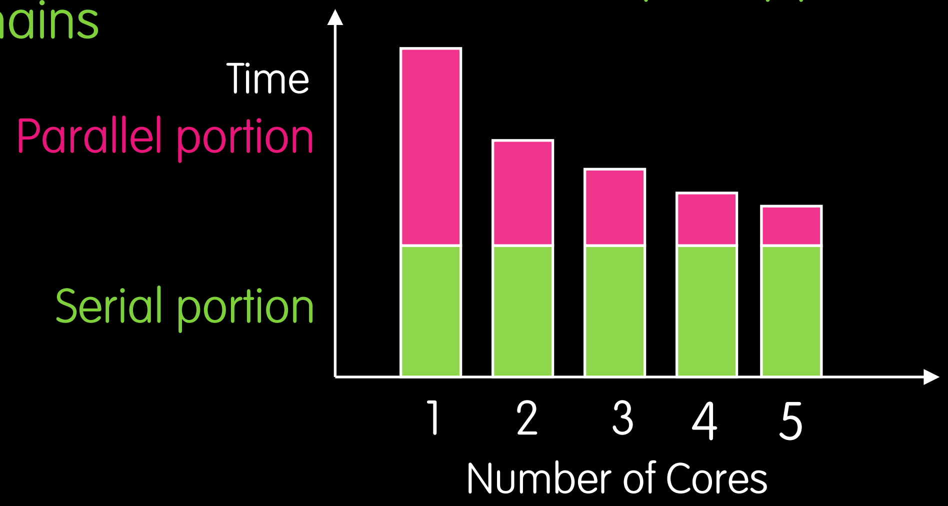

- **A single CPU can execute many threads by** ***Time Division Multipexing***




- ***Multithreading*** **is running multiple threads through the same hardware** en.wikipedia.org/wiki/Amdahl's_law

# Speedup Issues : Amdahl's Law

- Applications can almost never be completely parallelized; some serial code remains




- s is serial fraction of program, P is # of cores (was processors)
- **Amdahl's law:**

Speedup(P) = Time(1) / Time(P)

≤ 1 / ( s + [ (1-s) / P) ], and as P → ∞

≤ 1 / s

- Even if the parallel portion of your application speeds up perfectly, your performance may be limited by the sequential portion

# Speedup Issues : Overhead

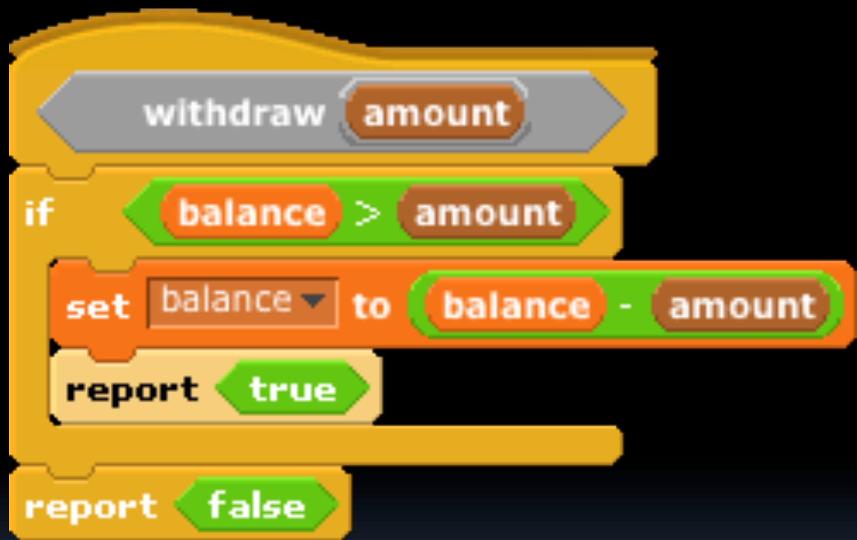
- **Even assuming no sequential portion, there's…**
  - Time to think how to divide the problem up
  - Time to hand out small "work units" to workers
  - All workers may not work equally fast
  - Some workers may fail
  - There may be contention for shared resources
  - Workers could overwriting each others' answers
  - You may have to wait until the last worker returns to proceed (the slowest / weakest link problem)
  - There's time to put the data back together in a way that looks as if it were done by one

# Life in a multi-core world…

- **This "sea change" to multi-core parallelism means that the computing community has to rethink:**
  a) Languages
  b) Architectures
  c) Algorithms
  d) Data Structures
  e) All of the above


# But parallel programming is hard!

- **What if two people were calling withdraw at the same time?**
  - E.g., balance=100 and two withdraw 75 each
  - Can anyone see what the problem *could* be?
  - This is a race condition
- **In most languages, this is a problem.**
  - In Scratch, the system doesn't let two of these run at once.



# Another concurrency problem … deadlock!

- Two people need to draw a graph but there is only one pencil and one ruler.
  - One grabs the pencil
  - One grabs the ruler
  - Neither release what they hold, waiting for the other to release
- Livelock also possible
  - Movement, no progress
  - Dan and Luke demo

# Summary

- "Sea change" of computing because of inability to cool CPUs means we're now in multi-core world
- This brave new world offers lots of potential for innovation by computing professionals, but challenges persist