Computer Systems are Different!

Robert Morris and Frans Kaashoek

6.033 Spring 2009

MASSACHUSETTS
INSTITUTE OF
TECHNOLOGY

Outline
hidden

- All systems are similar
  - But computer systems are different
- Unbounded composability
  - Easy to achieve complexity
- dtech / dt large for computer systems
- dcost / dt drives qualitative change

Composibility via static discipline

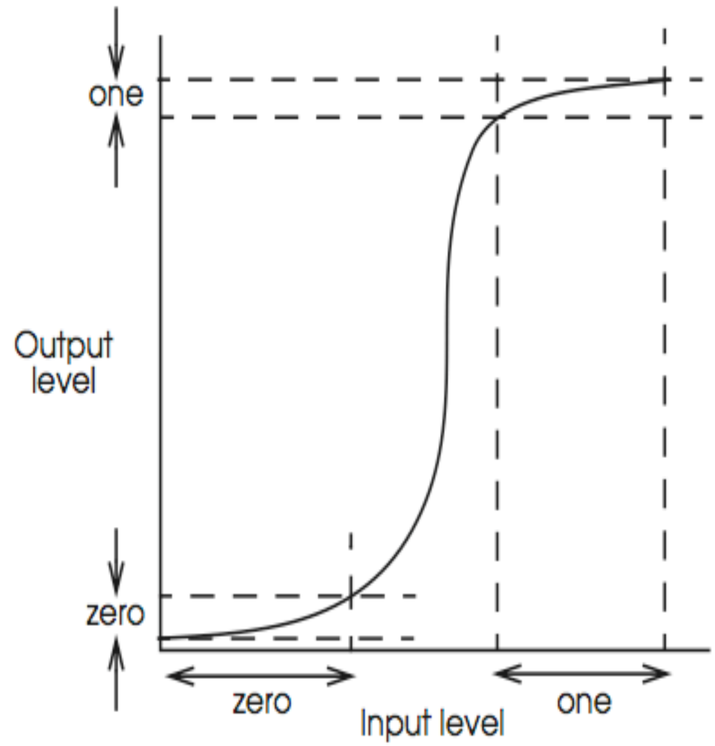


Figure 1–3: How gain and non-linearity of a digital component restore levels. The range of accepted inputs is much wider than the range of generated outputs.

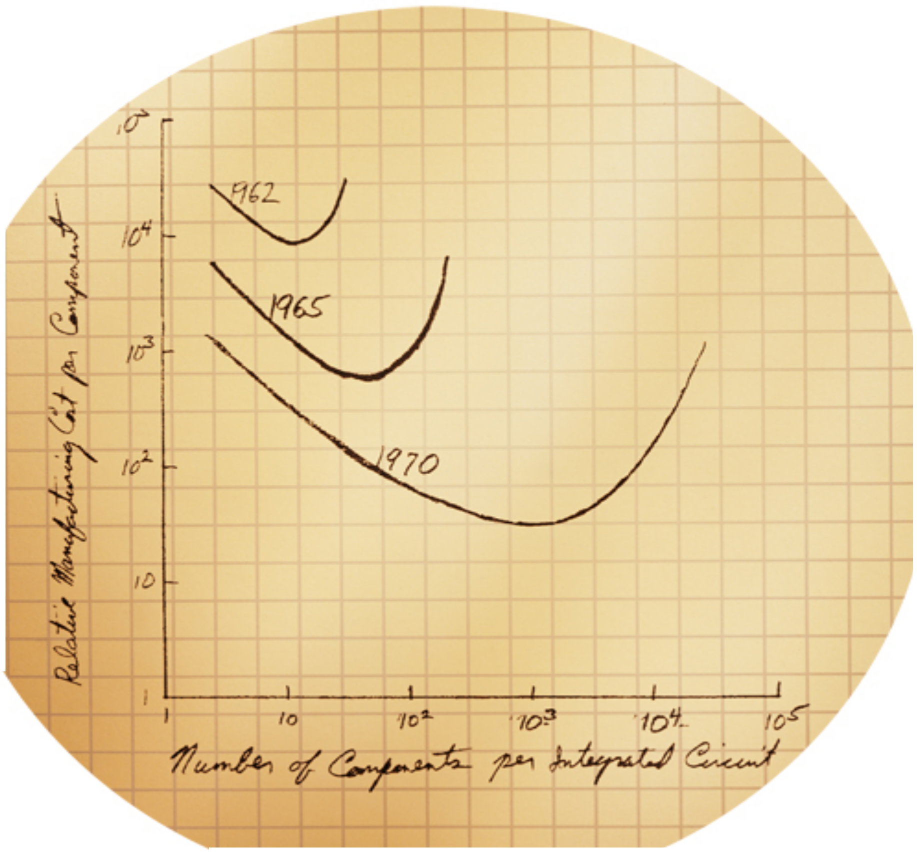
- <u>Be tolerant of inputs and strict on outputs</u>

Digital H/W hidden

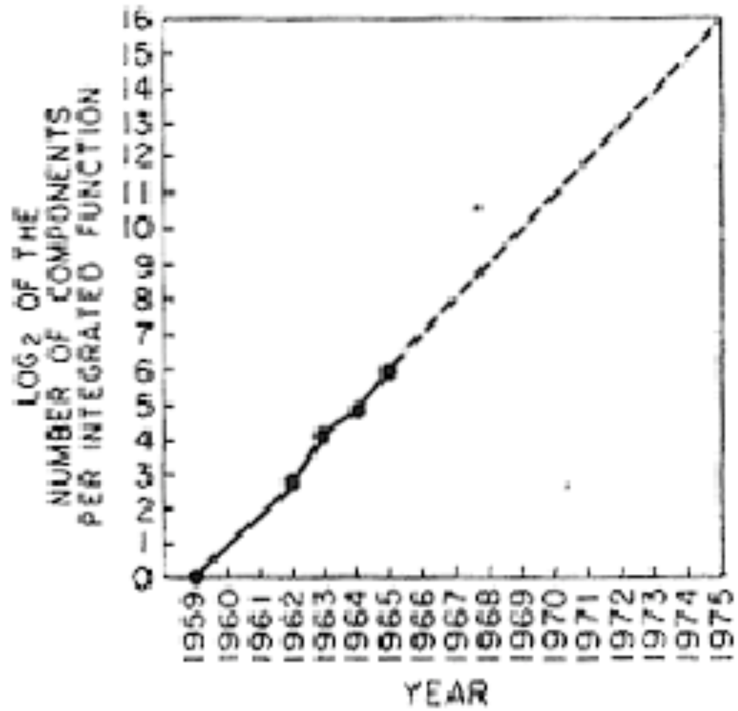
- Static discipline
  - Regenerate 0/1 at every gate
  - Noise does not accumulate (analog...)
  - Can chain together arbitrary #s of gates
- Other limits to size
  - Size, cost, reliability, power
- Rapid progress over many decades
  - Digital electronics a vast business
  - Lots of money for R&D -> rapid improvement
- Moore observed pattern for early ICs

Moore's law

cost per transistor



transistors per die

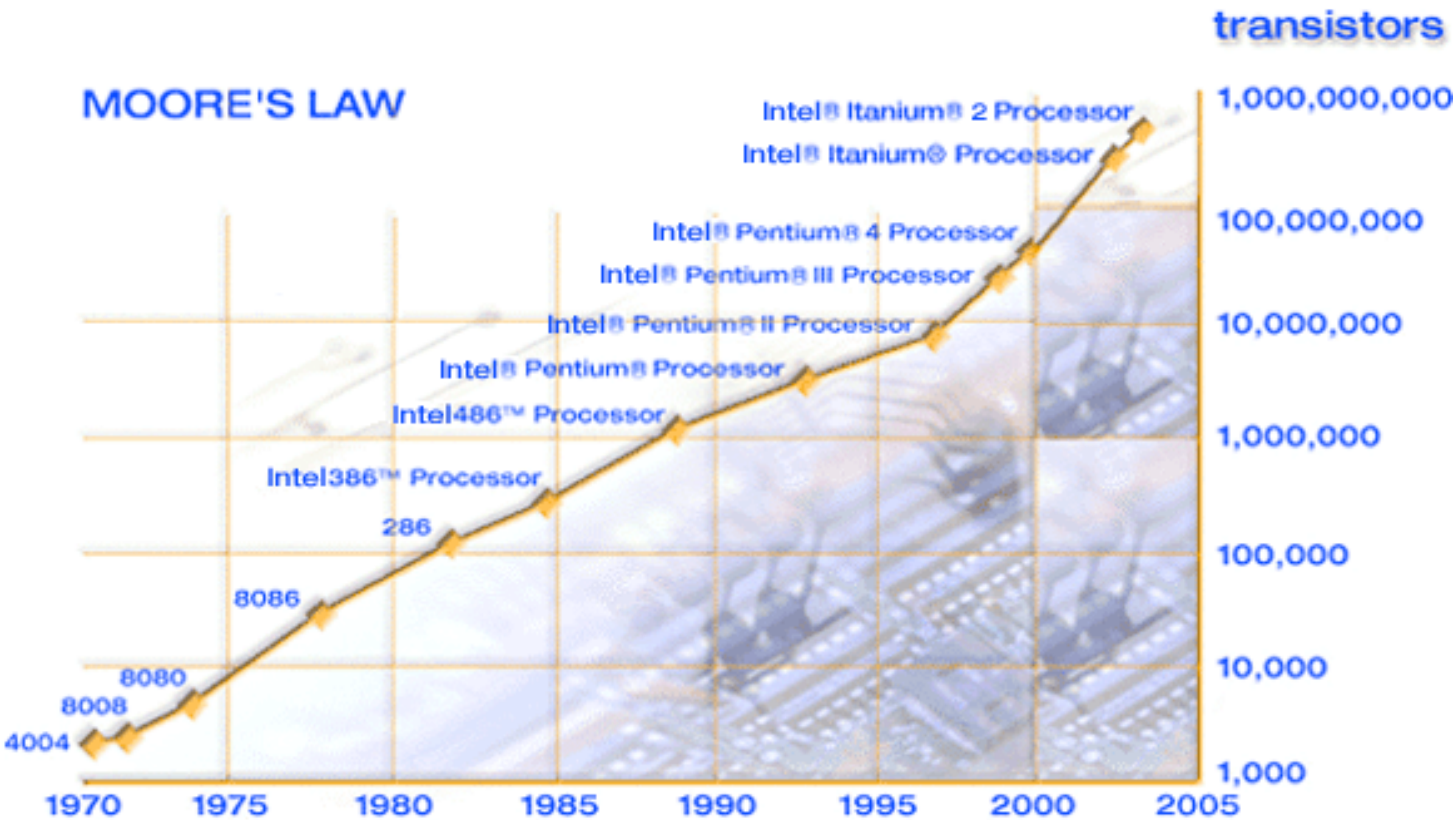


"Cramming More Components Onto Integrated Circuits", *Electronics*, April 1965

Moore’s Law
hidden

- argument to abandon flexibility of discrete devices
  - cheapness would dominate other considerations
- x-axis is transistors)per die
- y-axis is cost per transistor
- down: marginal cost basically zero
- up: yield, defects
- min is optimum die size: about 10 in 1962!
  - more AND CHEAPER every year
- right graph: plot of minima for a few years
- predicted 2^{16} by 1975: single-chip microprocessor!
- how did that prediction work out?

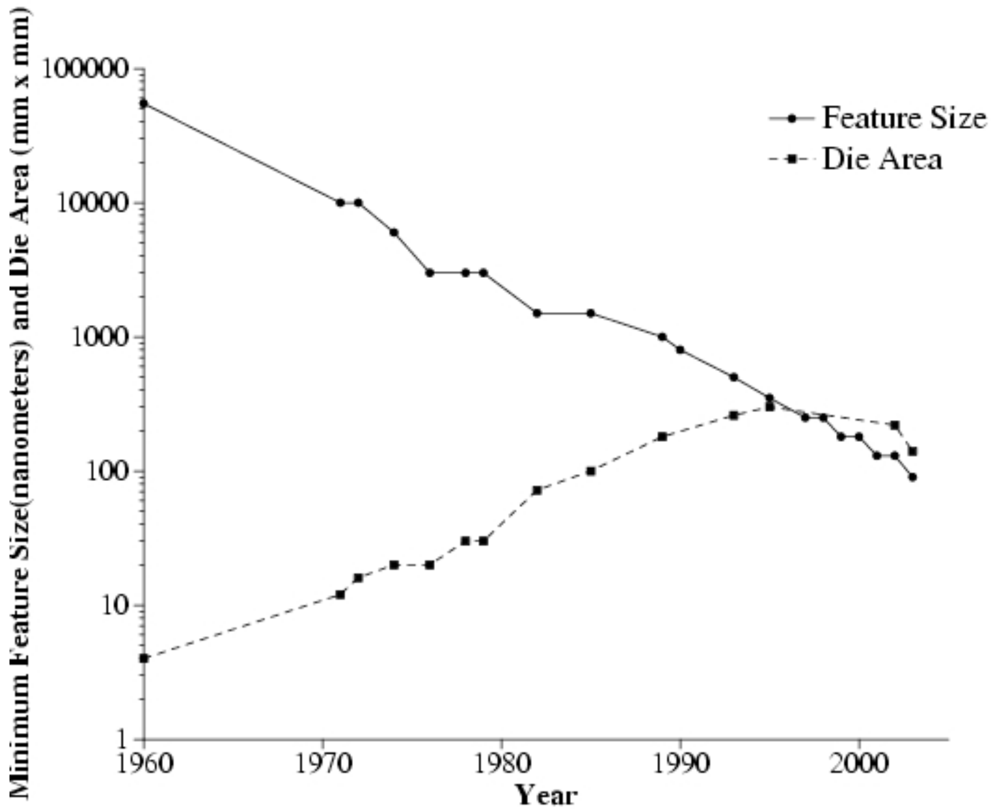
Transistors/die doubles every ~18 months



2x transistors / 18 months hidden

- Moore was right!
- 1974: 8080 (first serious uproc), 4,500, 2 mhz
- my laptop chip has 400 million
- latest server chips have 2 billion
- improvement AND EXPECTATION has had huge effect
- what drives consistent exponential increase?

Lithography: the driver behind transistor count



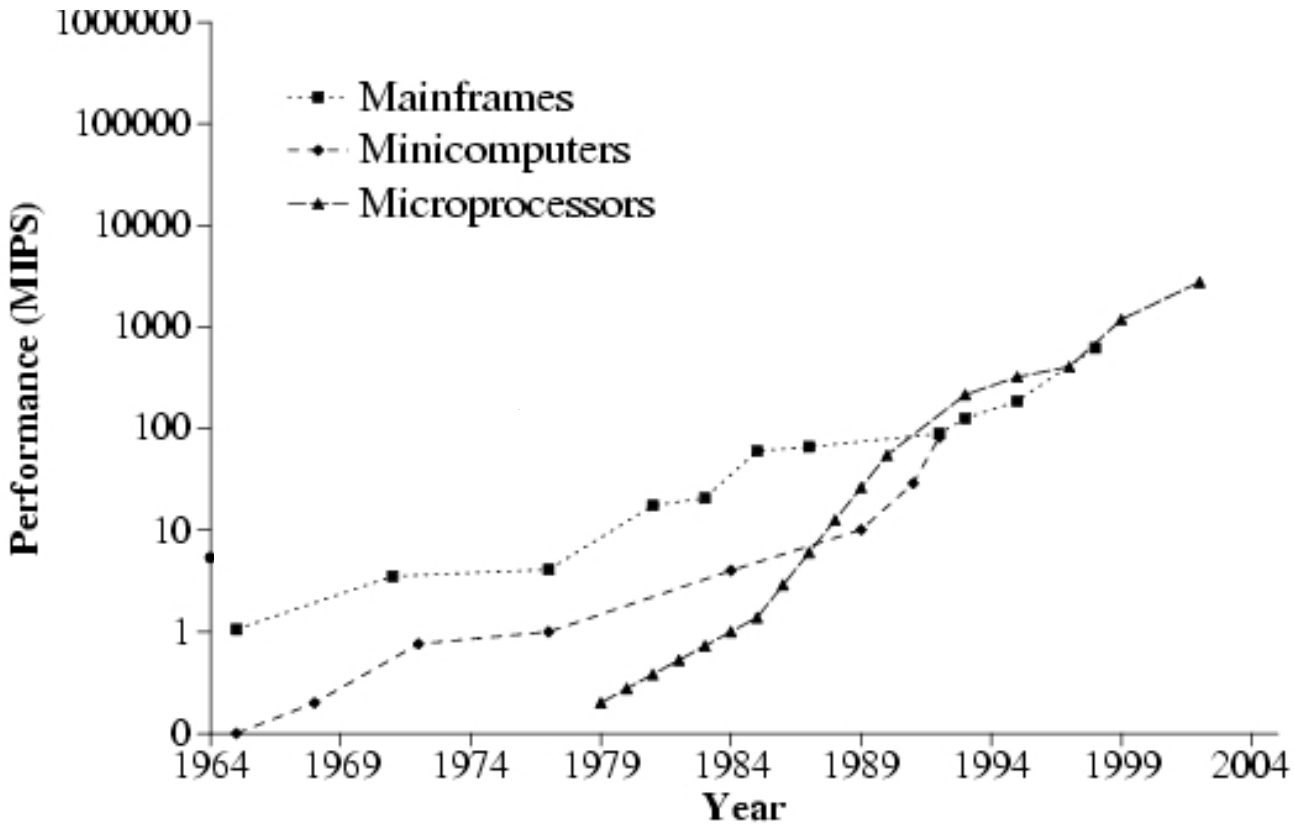
Trends in Minimum Feature Size and Die Area

- Components/area $O(x^2)$ with feature size
- Total components $O(a)$ with die area
- Switching rate $O(x)$ with feature size

Lithography
hidden

- exponential increase due to progress in lithography
  - masks, photosensitive chem, etch
- y-axis: feature size
  - feature: wire or transistor
  - smaller wavelength (ultraviolet)
  - currently 45 nm
- y-axis: die area
  - limited by defects: constant defects / unit area
- we get to multiply area and feature area! for n^3
  - claim 18 months is combination of the two

CPU performance



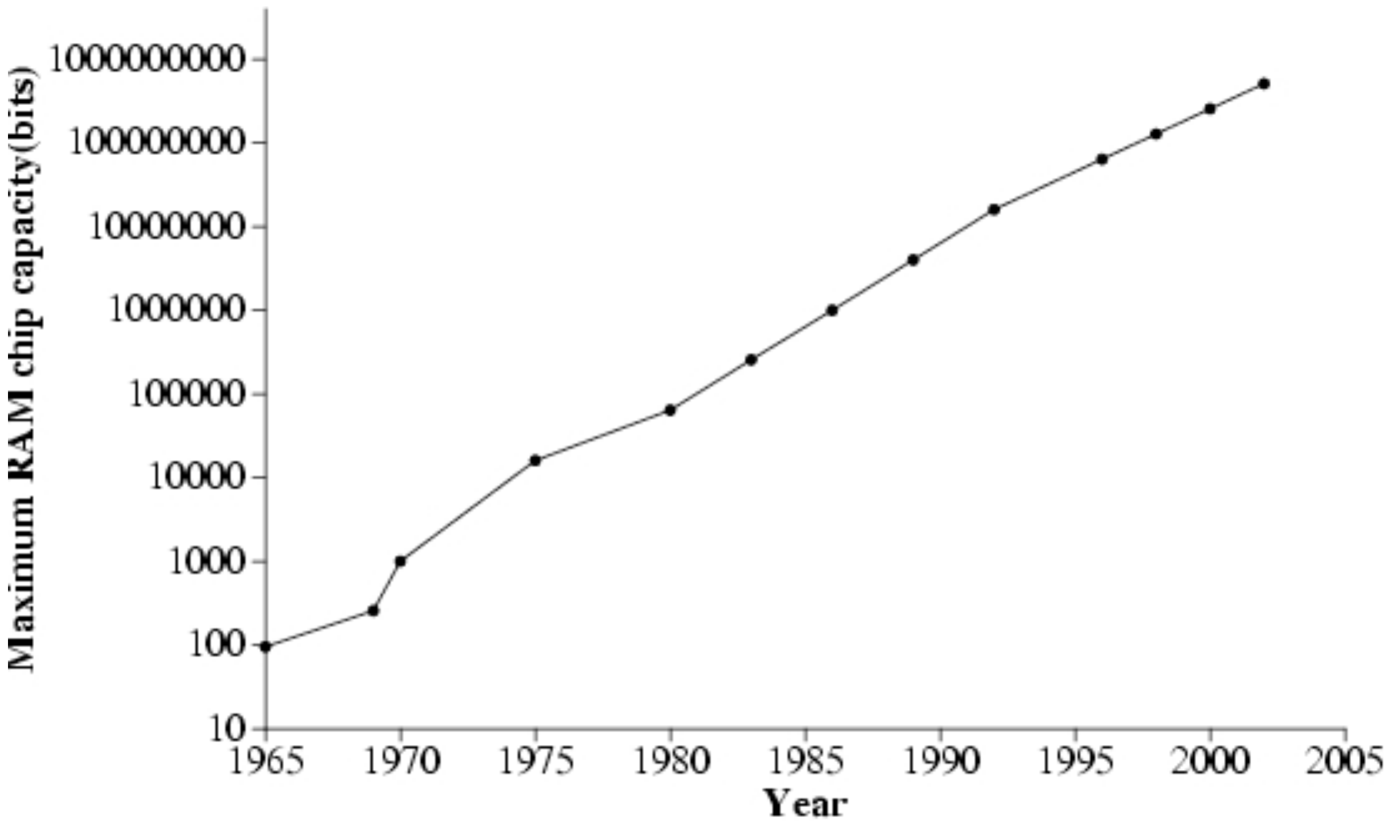
Trends in CPU performance growth, from microprocessors to supercomputers

CPU performance hidden

- the low end ate the high end
- until 1990 expensive much faster than cheap
  - made very differently, lower integration
- now expensive use same chips as cheap
- there is only one economic technology now

- other tech improved similarly: DRAM and disk

DRAM density

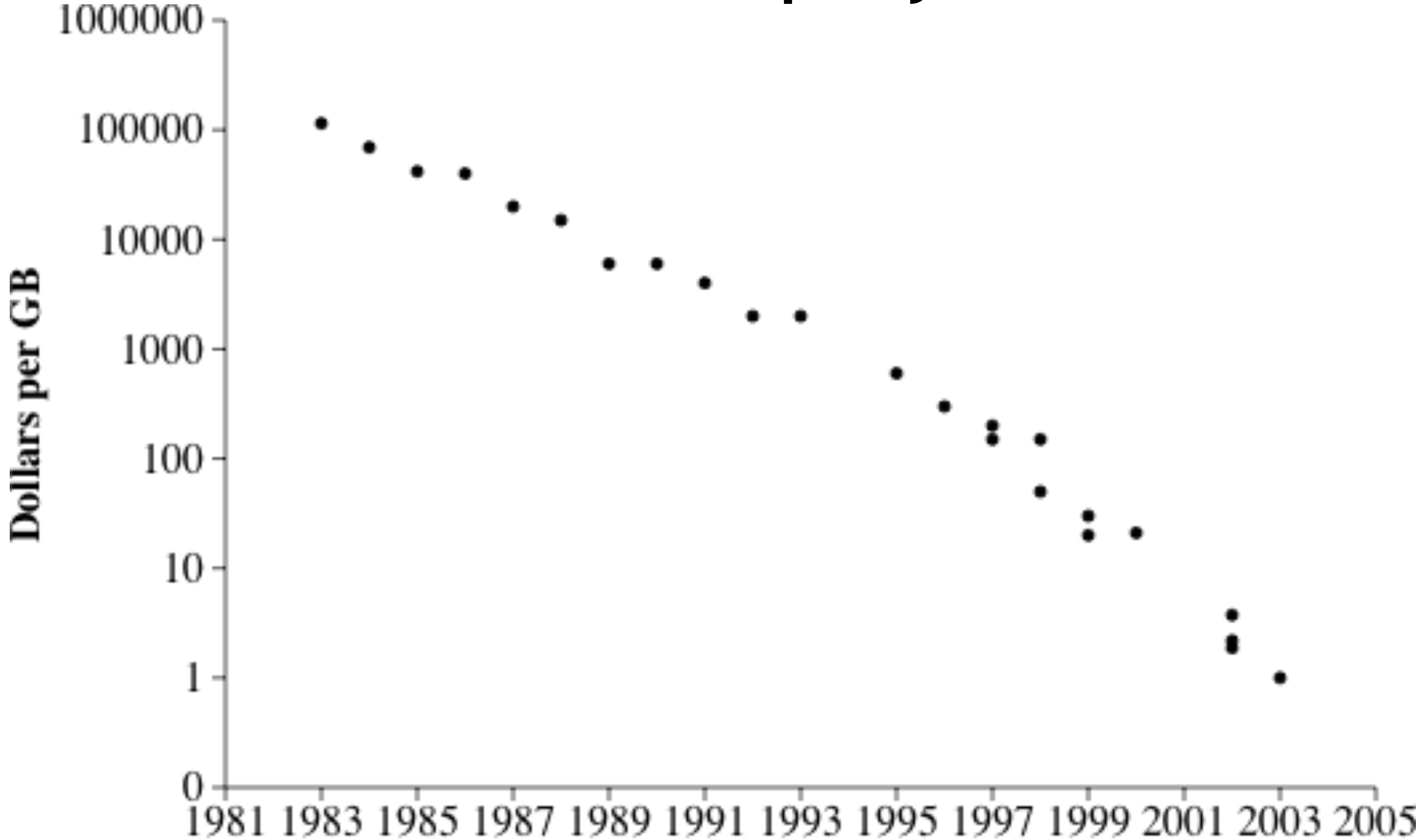


Trends in semiconductor RAM density

DRAM density hidden

- IC improvements have also driven DRAM
- memory has gotten much cheaper and denser
  - kilo or a few megabytes in 1980, 1000x in 2009
- hasn’t gotten a lot faster, maybe 3x
  - memory access time used to be about same as CPU cycle time
  - now CPU 300 times faster!
  - DRAM access is a serious bottleneck

Disk: Price per GByte drops at ~30-35% per year



Magnetic Disk Price History

Disk
hidden

- price drop due to increase in density
  - bits per inch on magnetic surface
  - density doubled every year
  - smaller heads, better electronics, surfaces
- early 1980s: 400 MB (!) disks, huge, $10,000
- 2009: 1 TB for about $120, or 12 cents / GB
- what about performance?
  - density helps: transfer rate
  - but seek times decreased only 3x since 1980
  - because mechanical
  - disk seek time is a serious bottleneck!

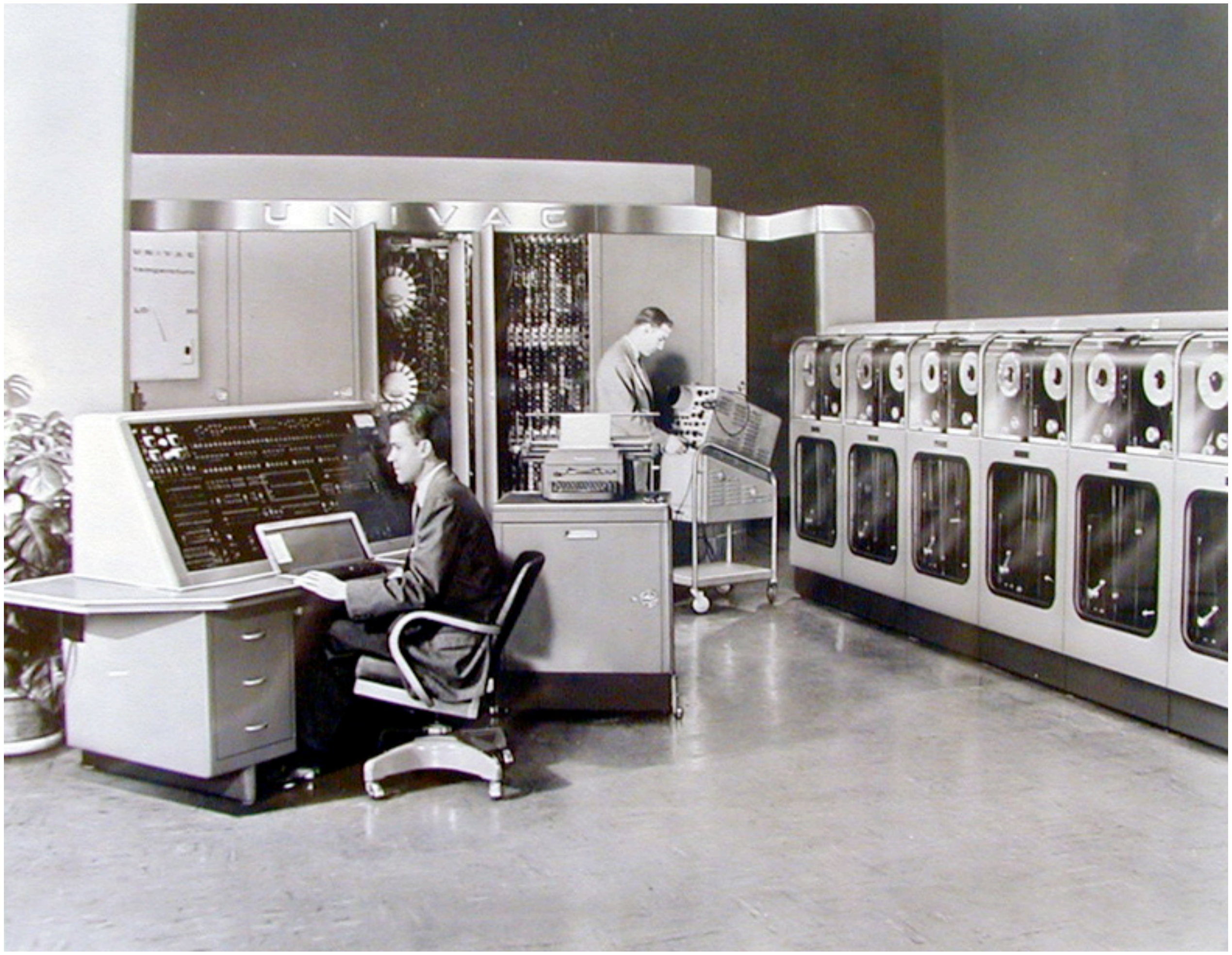
ENIAC

- 1946
- Only one
- 5000 adds/sec
- 20 10-digit registers
- 18,000 vacuum tubes
- 124,500 watts
- Not really stored program

ENIAC
hidden

- Illustrate trends w/ selected computers from history
- ENIAC: first GP electronic programmable computer
  - Electronic Numerical Integrator And Computer
- Army 1946, artillery firing tables (12 hr-> 30 min)
- no memory, just registers and constant tables
  - 20 10-digit registers
- programmable w/ switches/plugs
  - NOT stored program!
- 5000 ops/second
- 18,000 vacuum tubes: failure per day!!!
- only one, not commercial

UNIVAC (Universal Automatic Computer)



- 1951
- 46 sold
- 2000 ops/sec
- 1,000 12-digit words (mercury)
- 5000 tubes
- $1.5 million

UNIVAC
hidden

- first american commercial computer
  - 1951, 46 sold to big companies / government
- stored program! i.e. program in memory.
- designed by ENIAC designers, in a start-up
- fewer tubes than ENIAC (5000), and slower (2000 ops/sec)
- had memory: 1000 12-digit mercury delay lines
  - encode bits in acoustic waves, recycle
- expensive, huge, required a big staff

IBM System/360-40

- 1964
- 1.6 MHz
- 16-256 KB core
- $225,000
- Family of six
- 32-bit
- Time-sharing

System/360
hidden

- first modern computer system: 1964
- familiar to us (unlike previous examples)
  - 8-bit bytes, 32-bit addresses, time-sharing OS
  - programming languages, compilers
  - some had virtual memory
- a range of compatible models
  - separated architecture from implementation
  - 8K to 512K mem, 1 mhz to 5 mhz
  - $100,000 to $5,000,000 (but mostly leased?)
  - upgrade path: customer can start cheap and grow
  - preserves s/w investment

Cray 1: supercomputer

- 1976
- 80 sold
- 80 MHz
- 8 Mbyte SRAM
- 230,000 gates
- $5 million

Cray-1
hidden

- most famous and almost first super-computer: 1976
  - designed only for speed, not economy
  - you could get more speed for more money
  - simulate nuclear explosions, oil exploration, &c
- 80 mHz: very fast
  - a few mHz typical for the time
  - 130 kilowatts dissipated, due to 80 mHz
  - refrigerated w/ freon, integrated into frame
  - short wires, thus C shape, backplane in center
- 230,000 gates (only a few per chip)
- faster than any microprocessor until early 1990s!

DEC PDP-8 (1964)

- 60,000 sold
- 330,000 adds/sec
- 4096 12-bit words
- $18,000

DEC PDP-8
hidden

- first successful minicomputer
  - 1965, cheap, small, flexible
  - lab of a few people could afford one
- very widely used
  - i have owned two, learned machine lang
- built from chips with a few gates on them (like cray)
- 12 bits: cheap, but guarantees limited family life
  - crummy timesharing and compilers
  - too few address bits a problem even now
- contrast to ibm 360’s 32 bits
- great for a lab, but big/expensive/complex for personal computer

Apple II

- 1977
- 1 MHz
- 6502 microprocessor
- 4 to 48 Kilobytes RAM
- $1300
- Basic, Visicalc

Apple II
hidden

- one of first very successful personal computers
  - cheap/small enough that a family could buy one
  - single-chip microprocessor (6502)
- my high school had these
- games, educational, visicalc (first spread sheet)
- built-in basic interpreter
- pretty low end
  - but this was the winning line of development

IBM's wrist watch

- 2001
- Linux and X11
- 74 Mhz CPU
- 8 Megabyte flash
- 8 Megabyte DRAM
- Wireless

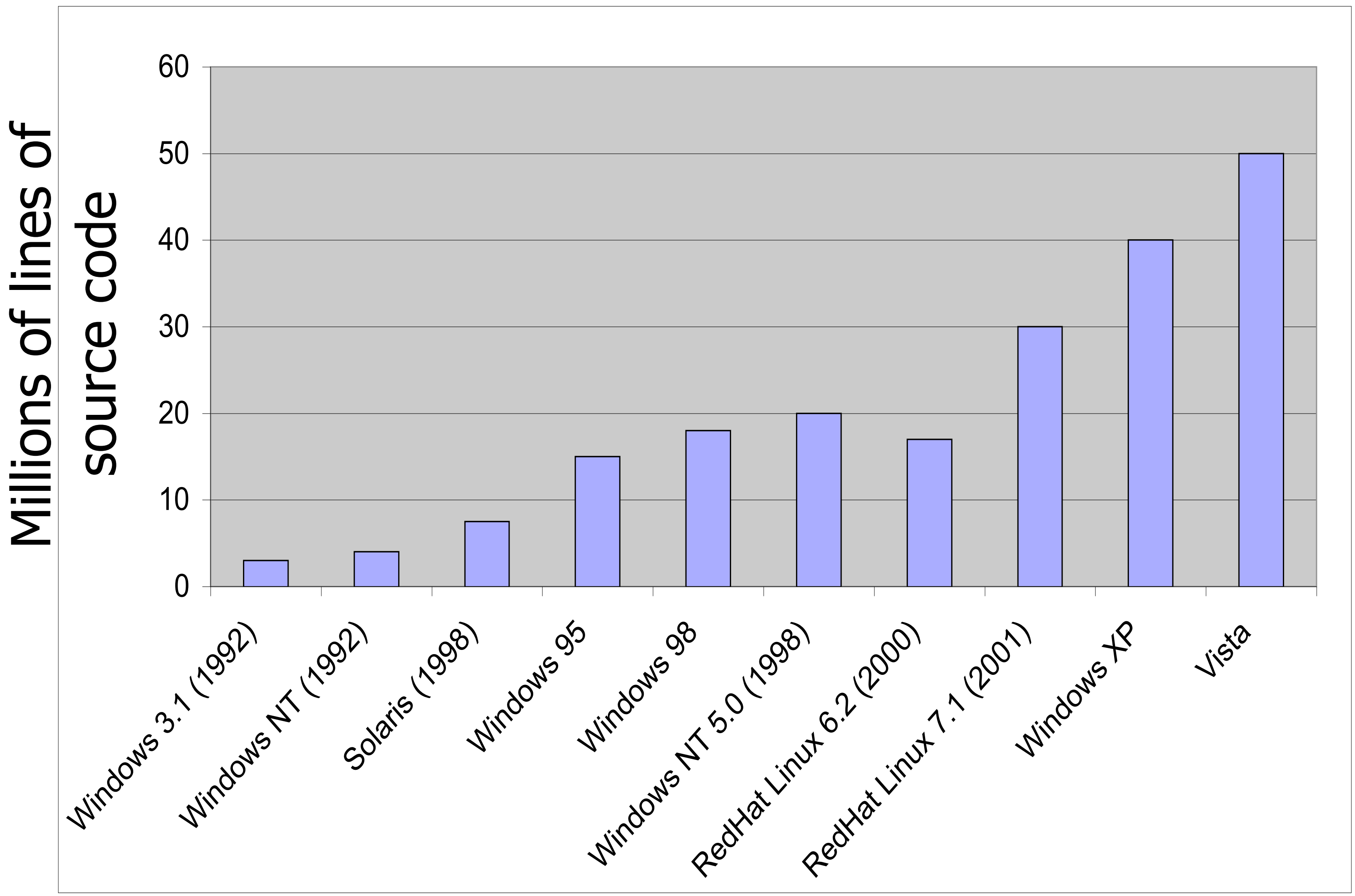
IBM Linux Wrist-Watch hidden

- from IBM Tokyo research lab
- about as powerful as Cray-1 (74 mhz, 8MB RAM)
- same size display as early IBM PCs (640x480)
- used to be a joke: impossible and pointless
- but now possible – maybe pointless, but iPhone isn’t

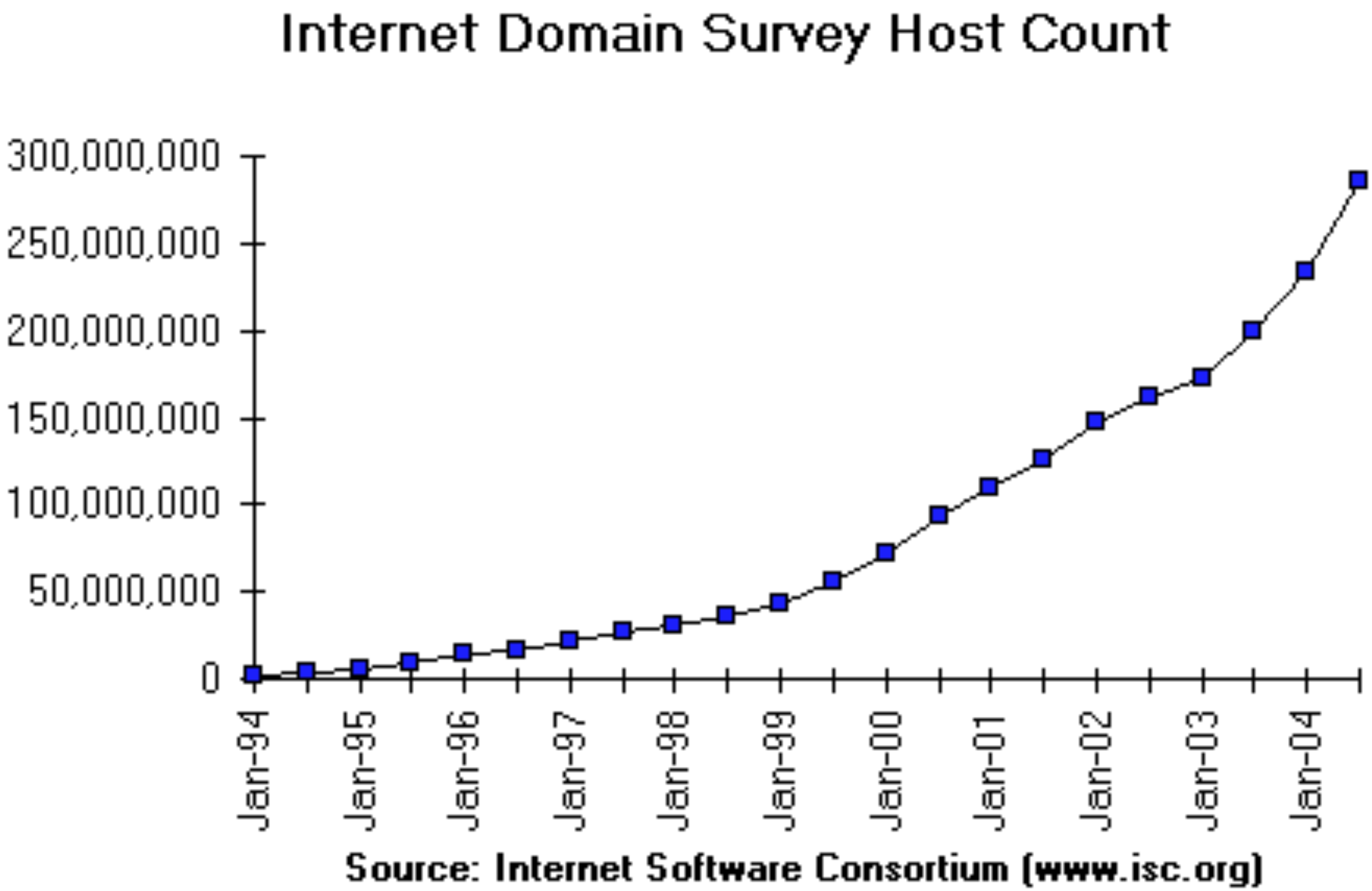
Software hidden

- No h/w limits to composition
  - Big CPU, DRAM, disk, networks
  - CHEAP
- Limiting factor is designers' understanding
- Tools have improved over the years
  - compilers, type checkers
  - high-level languages
  - language support for modularity
  - many ready-made libraries (modules)
  - version control / build / bug tracking systems
- Programmers are keeping up with hardware!

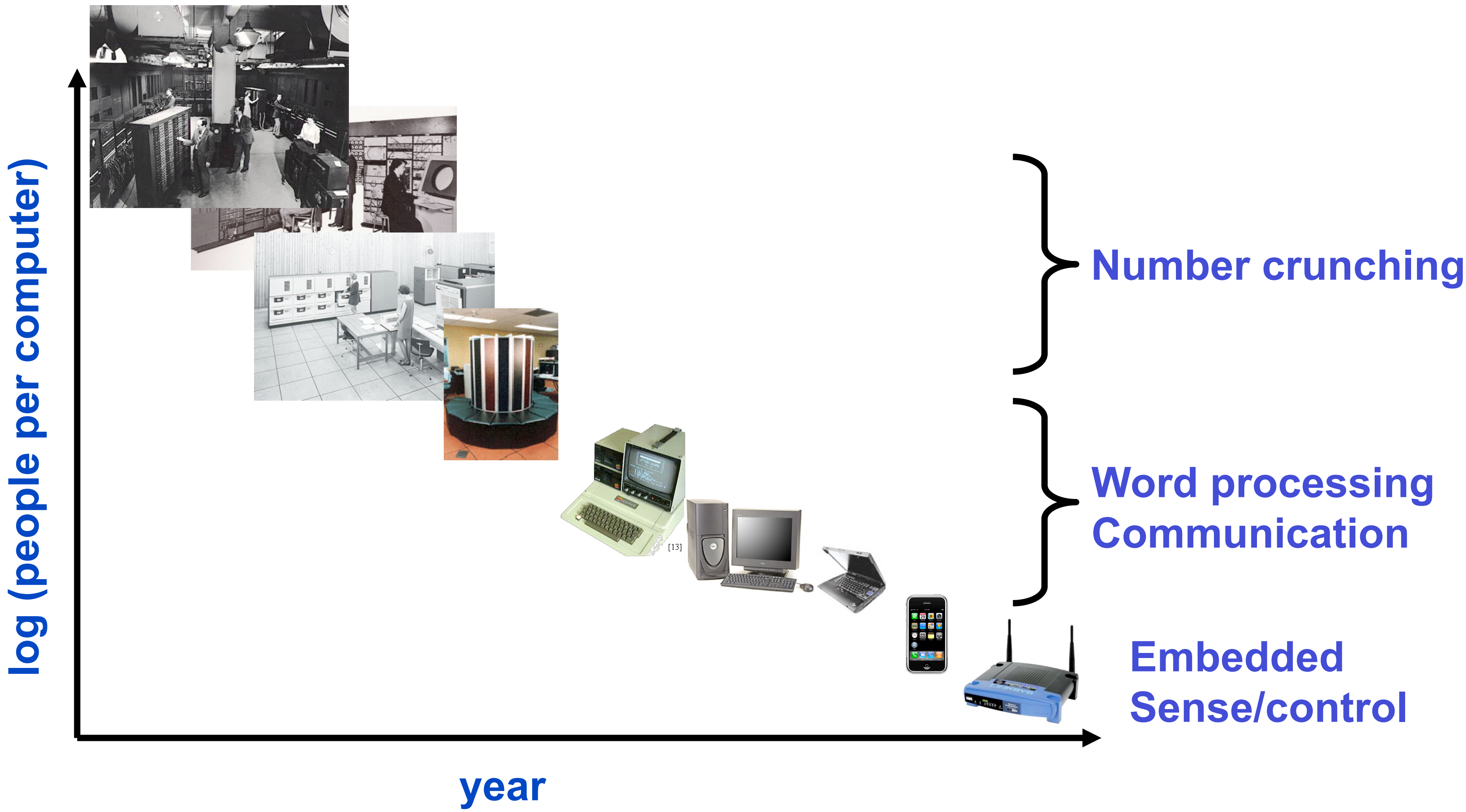
Software follows hardware



Cheap → Pervasive



Pervasive → qualitative change

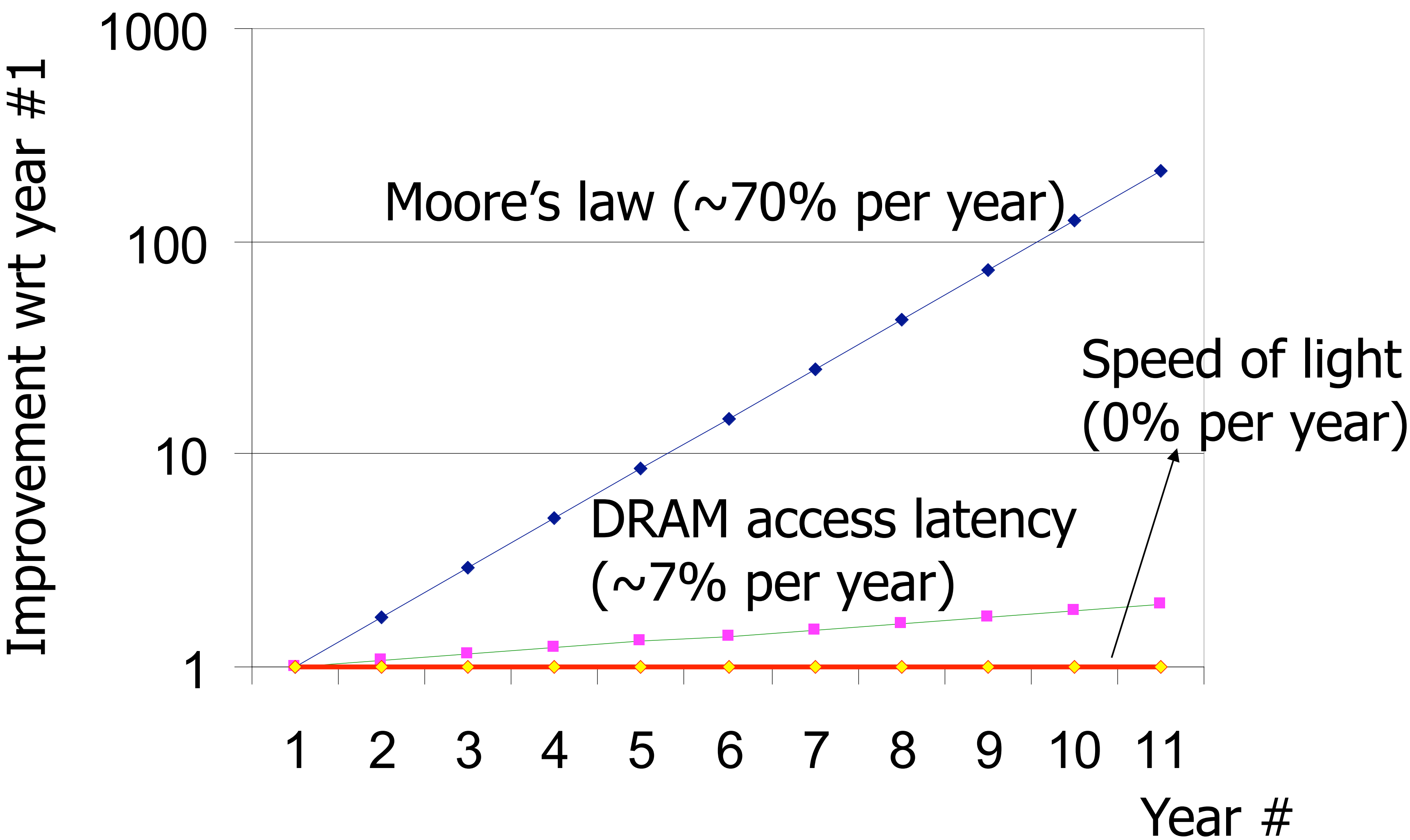


Slide from David Culler, UC Berkeley

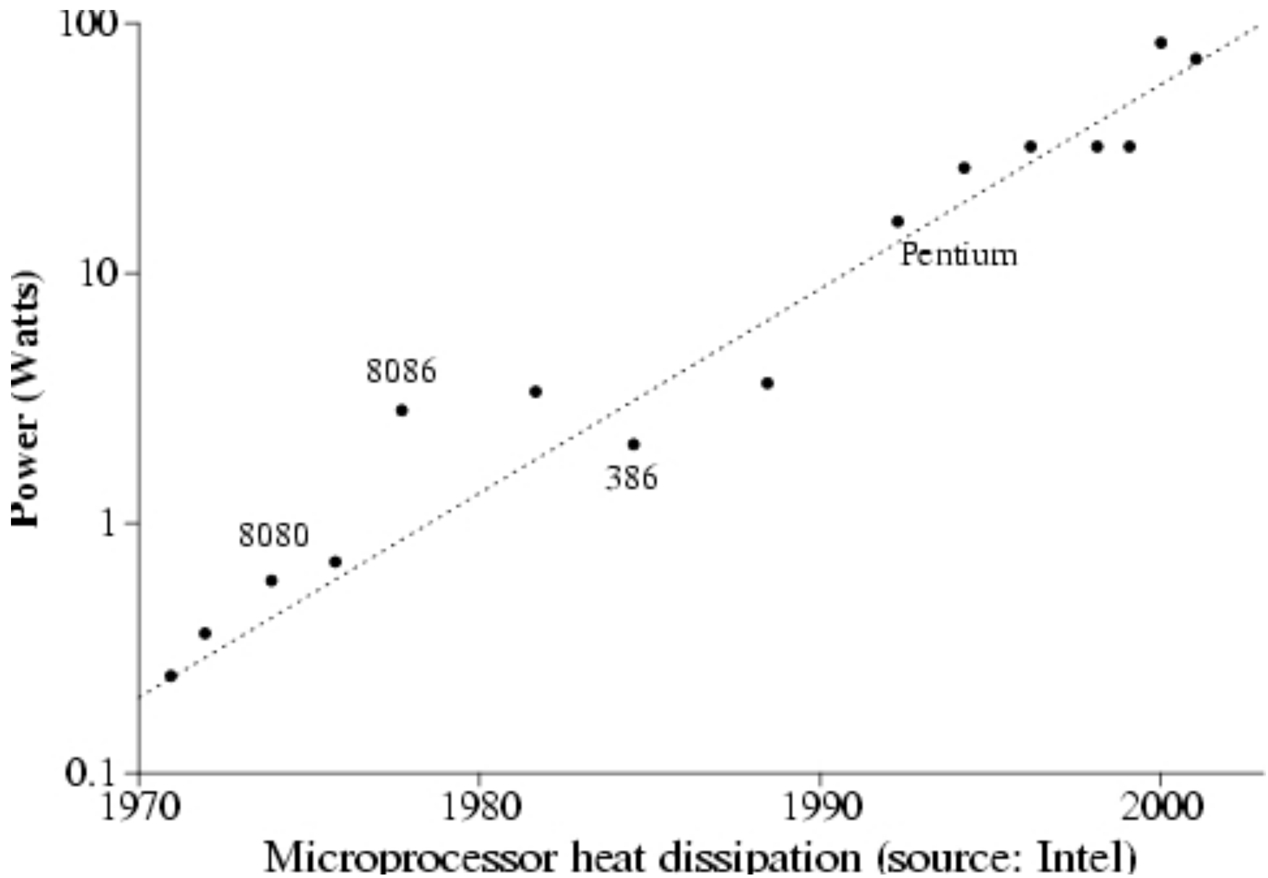
Storm clouds on horizon hidden

- Complexity
- Robustness increasingly important
- Society and the law
- Scaling problems

Latency improves slowly



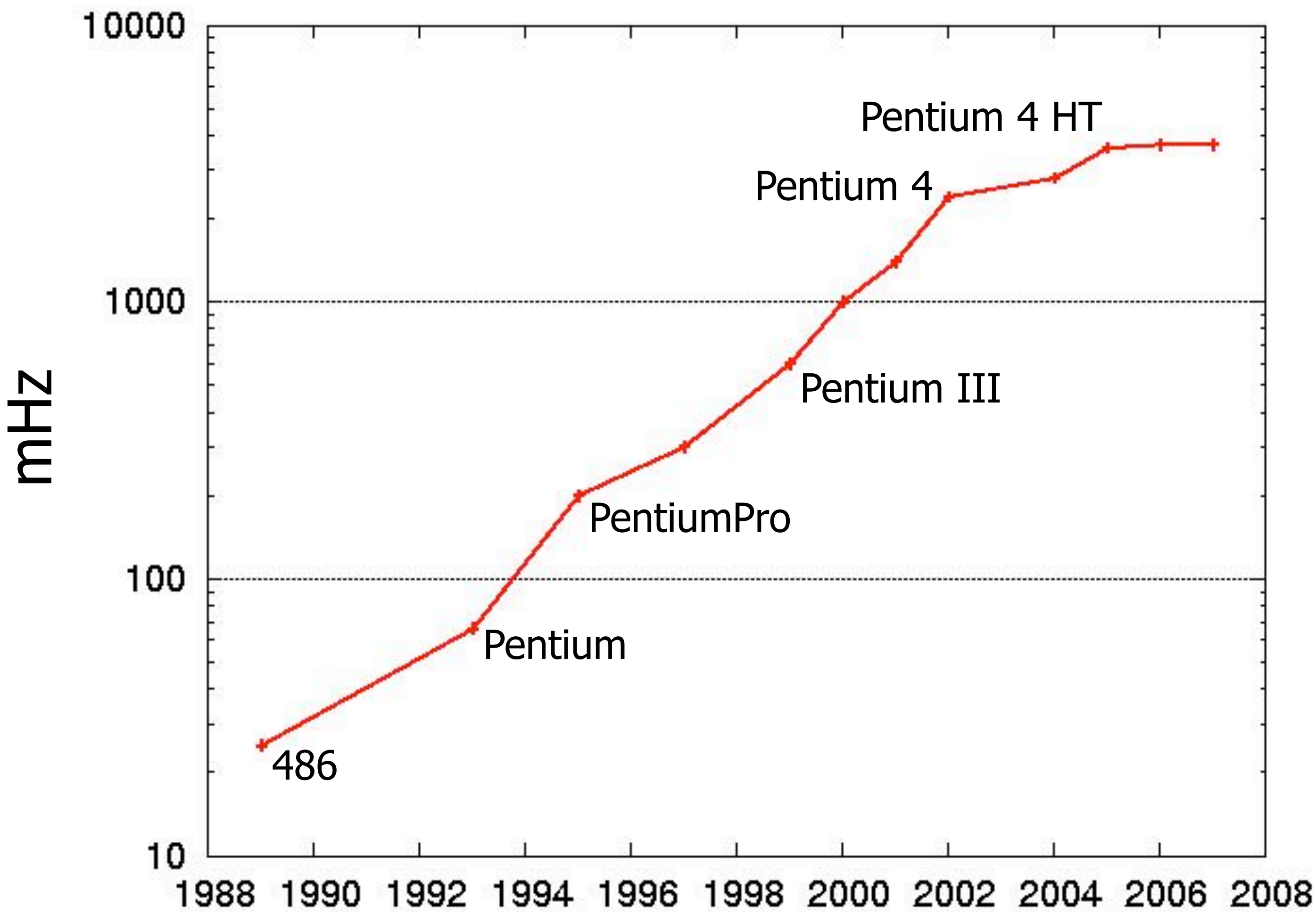
Heat is a problem



Heat is a problem hidden

- higher clock -> more switching -> heat
- modern CPUs are hot!
- 100w or 200w limit of air/fan cooling
- could go higher w/ liquid, but expensive

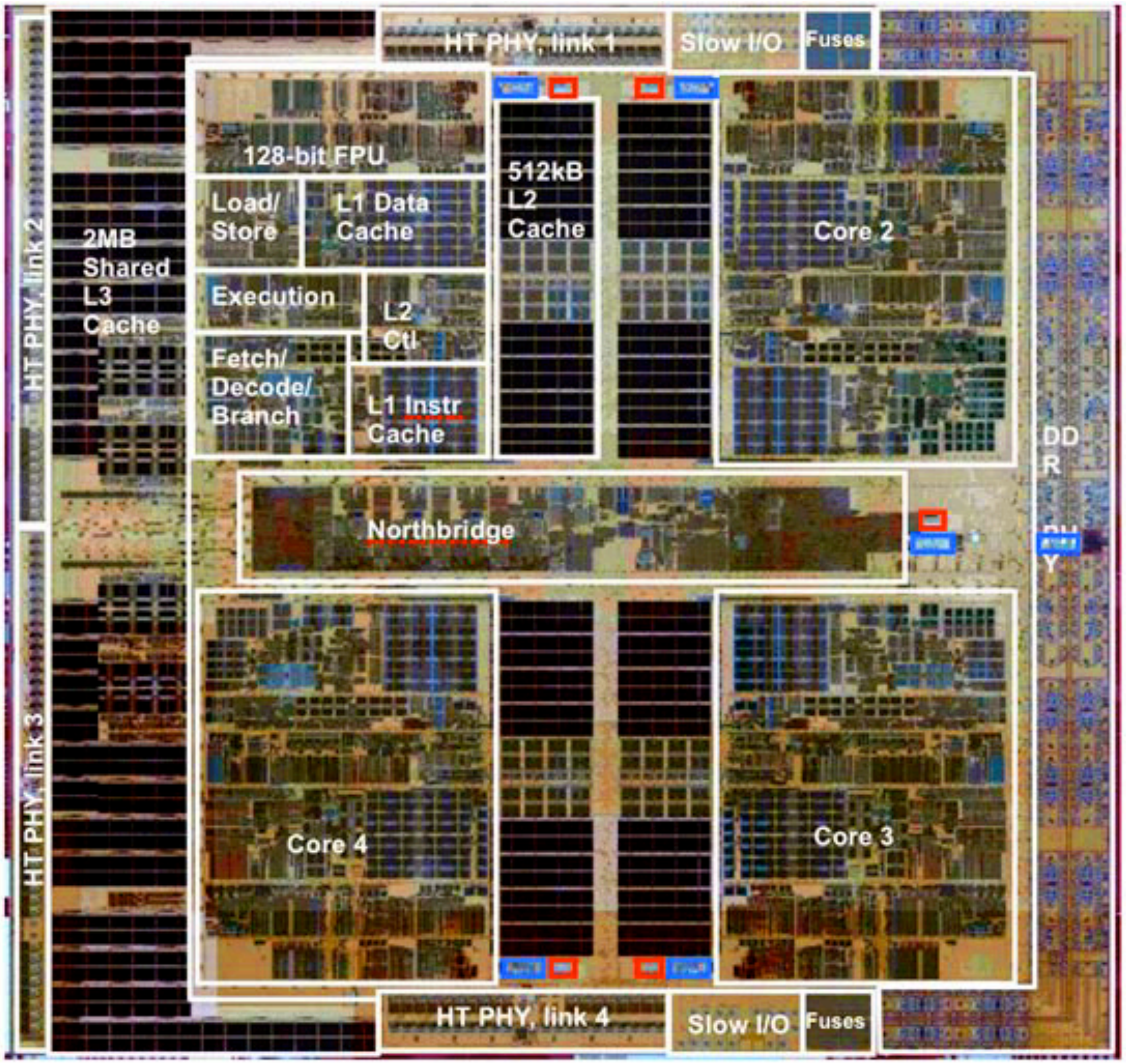
Recent Intel CPU Clock Rates



Clock rates
hidden

- up and up for many years
  - smaller features, less capacitance
  - also pipelining
- why stopped in 2005?
  - power / heat
  - small wires and gates: resistance &c
- what now?
  - still more transistors every year
  - can use them to get more performance
  - bigger caches
  - better architecture e.g. better branch prediction
  - more cores

The Future: will it be painful?



AMD Barcelona Quad-core chip

Multicore hidden

- 4x 2 GHz cores rather than one 8 GHz CPU
  - cannot build the latter
  - but 4x is "same performance"
- BUT much harder to program
  - split work into four balanced pieces
  - avoid stepping on toes when using shared data
  - not mainstream, tools (languages) not so good
- So: good news and bad news

What went right?

- Unbounded composibility
- General-purpose computers
  - Only need to make one thing fast
- Separate arch from implementation
  - S/W can exploit new H/W
- Cumulative R&D investment over years