Trends in computer systems

(slides from several sources)

Frans Kaashoek

6.033 Spring 2012

Outline

- All systems are similar
  - But computer systems are different
- Unbounded composability
  - Hardware and software
  - Easy to build too complex systems
- dtech / dt large for computer systems
- dcost / dt drives qualitative change

HW composibility via static discipline

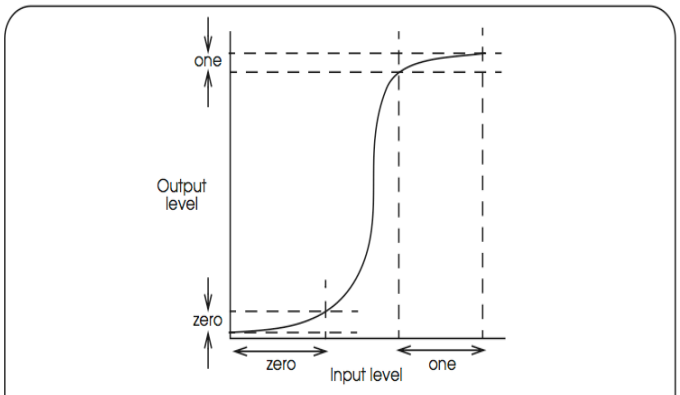


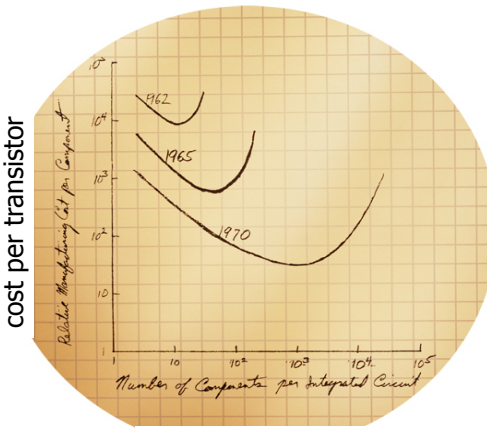
Figure 1–3: How gain and non-linearity of a digital component restore levels. The range of accepted inputs is much wider than the range of generated outputs.

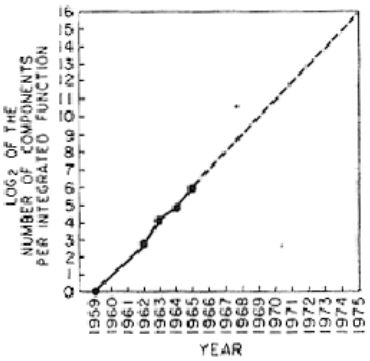
- Regenerate 0/1 at every gate
- Be tolerant of inputs and strict on outputs

Static discipline

- Noise does not accumulate
  - Unlike analog circuits
  - Can chain together arbitrary #s of gates
- Other limits to size
  - Size, cost, reliability, power
- Rapid progress over many decades
  - Integrated cuircits a vast business
  - Lots of money for R&D -> rapid improvement
- Moore observed pattern for early ICs

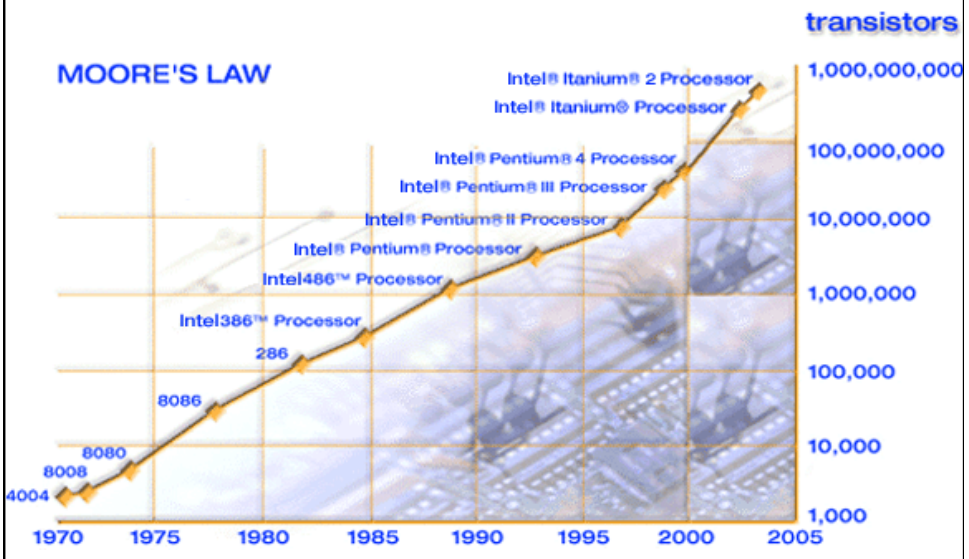
Moore's law





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

Transistors/die doubles every ~18 months

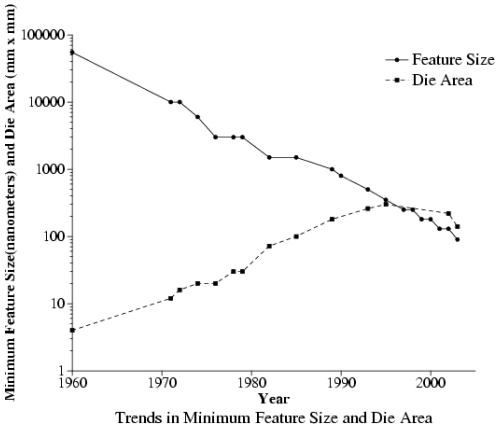


Moore's law sets a clear goal

- Tremendous investment in technology
- Technology improvement is proportional to technology
- Example: processors
  - Better processors ⇒
  - Better layout tools ⇒
  - Better processors
- Mathematically: $d(technology)/dt \approx technology$
  - $technology \approx e^t$

8

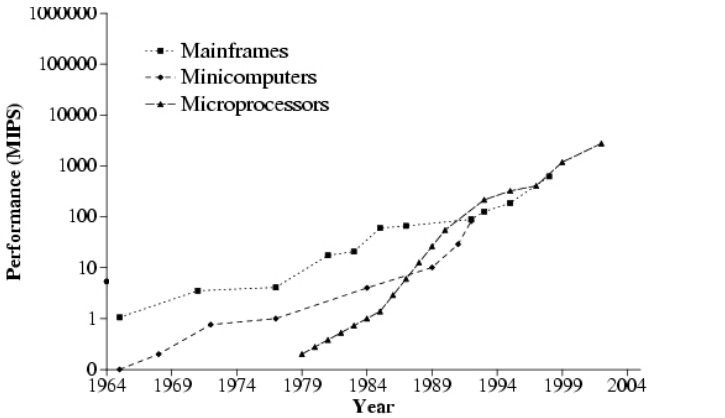
Lithography: the driver behind transistor count



Trends in Minimum Feature Size and Die Area

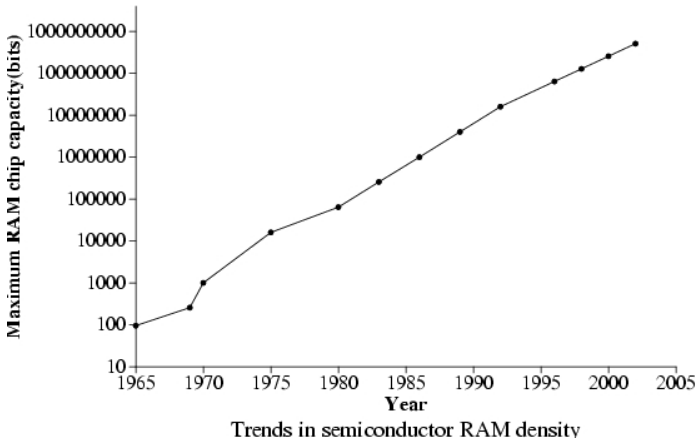
- Components/area O (x^2) with feature size
- Total components O (a) with die area
- Switching rate O(x) with feature size

CPU performance



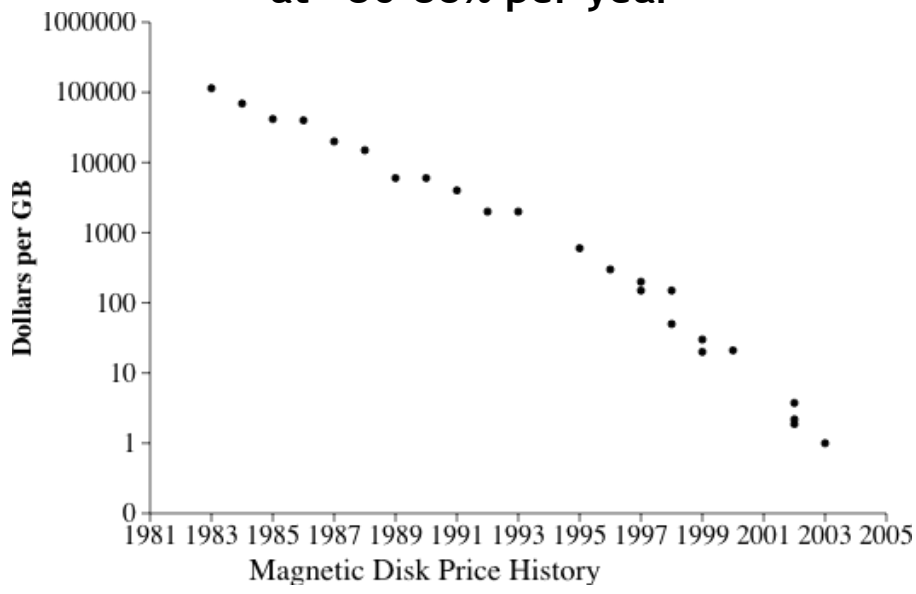
Trends in CPU performance growth, from microprocessors to supercomputers

DRAM density



Trends in semiconductor RAM density

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



ENIAC

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

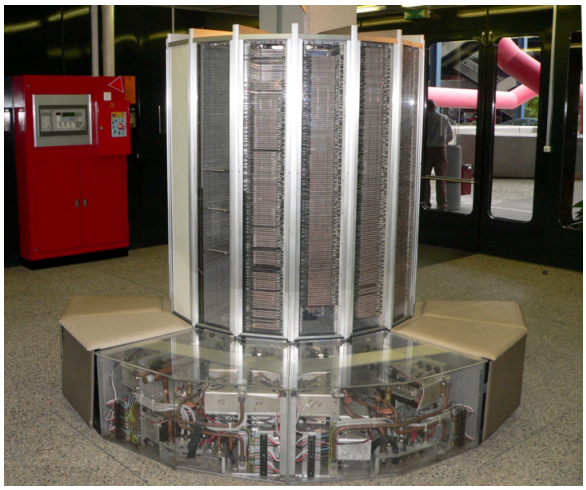
UNIVAC (Universal Automatic Computer)

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

IBM System/360-40

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

Cray 1: supercomputer



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

DEC PDP-8 (1965)

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

Apple II

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

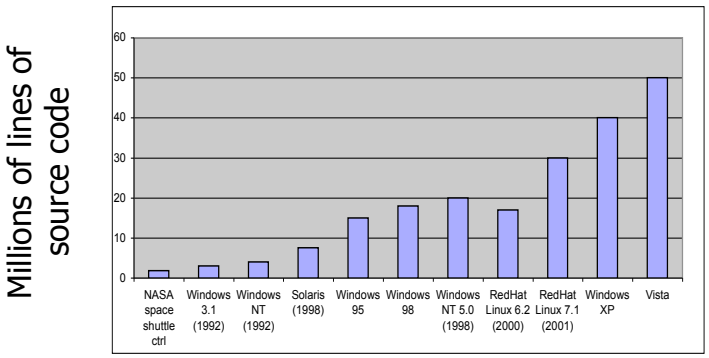
IBM's wrist watch

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

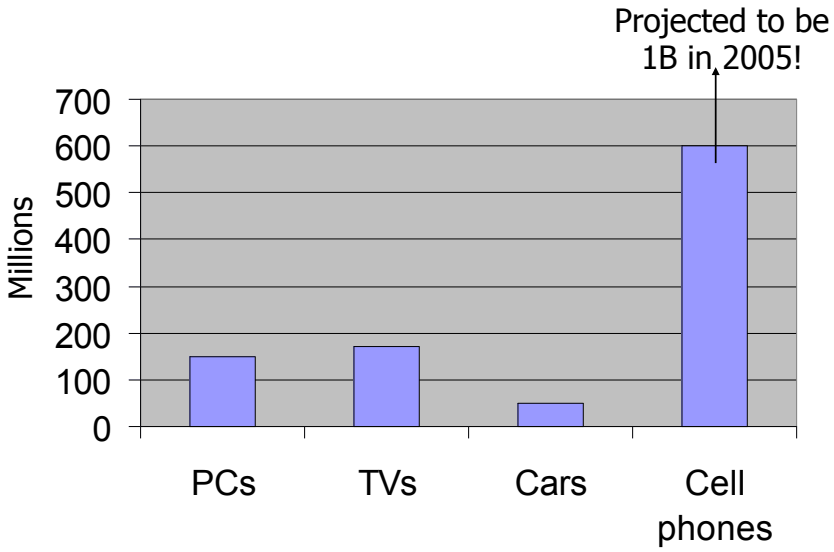
Software

- 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!

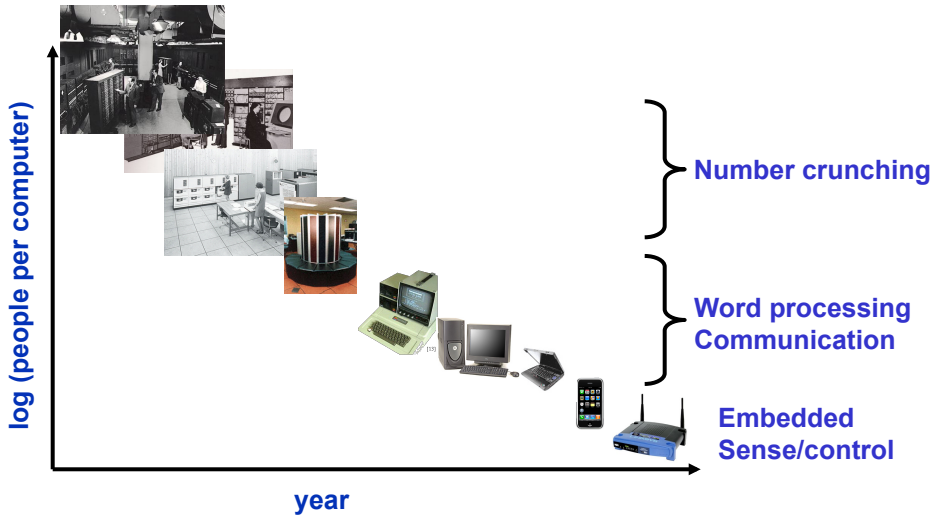
Software keeps up with hardware



Computing is everywhere!



Pervasive → qualitative change

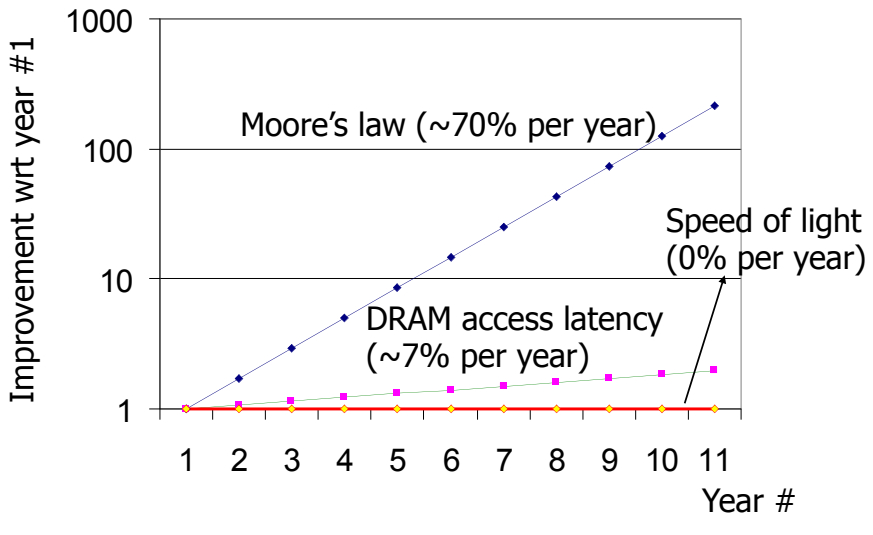


Slide from David Culler, UC Berkeley

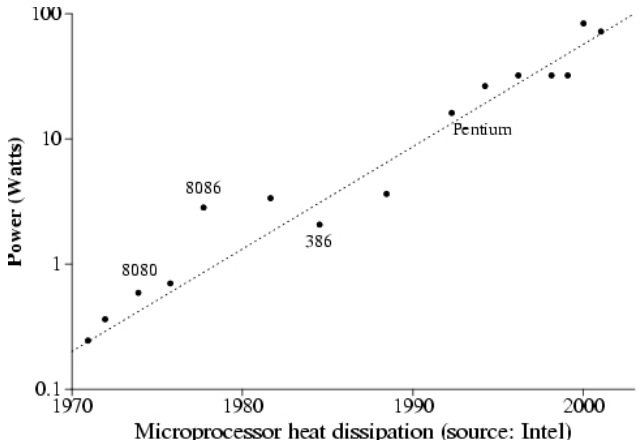
Storm clouds on horizon hidden

- Complexity
- Society and the law
- Scaling problems

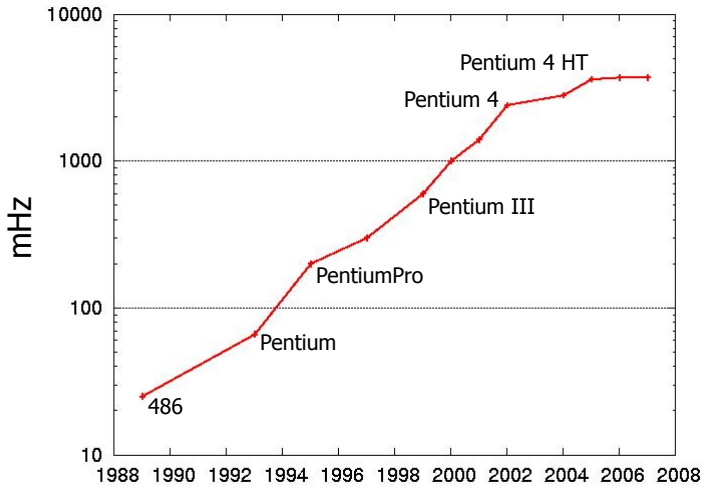
Latency improves slowly



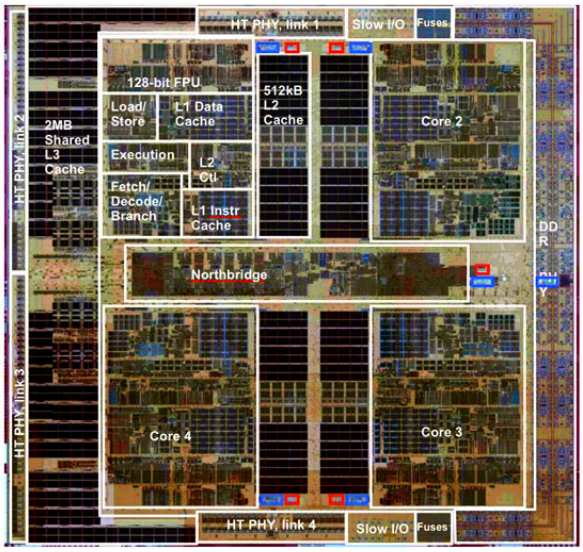
Heat is a problem



Recent Intel CPU Clock Rates



The Future: will it be painful?



AMD Barcelona Quad-core chip

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
- ➢ What you can build limited by your imagination
- ➢ Seldom design the same system twice
  - ➢ Every system is a new design problem