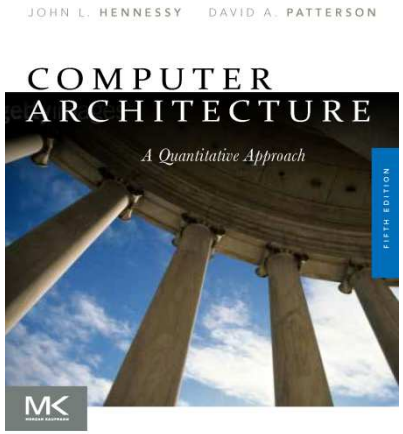


Fundamentals of Quantitative Design and Analysis

Computer Architecture

A Quantitative Approach,

Hennessy, Patterson

With other sources of information

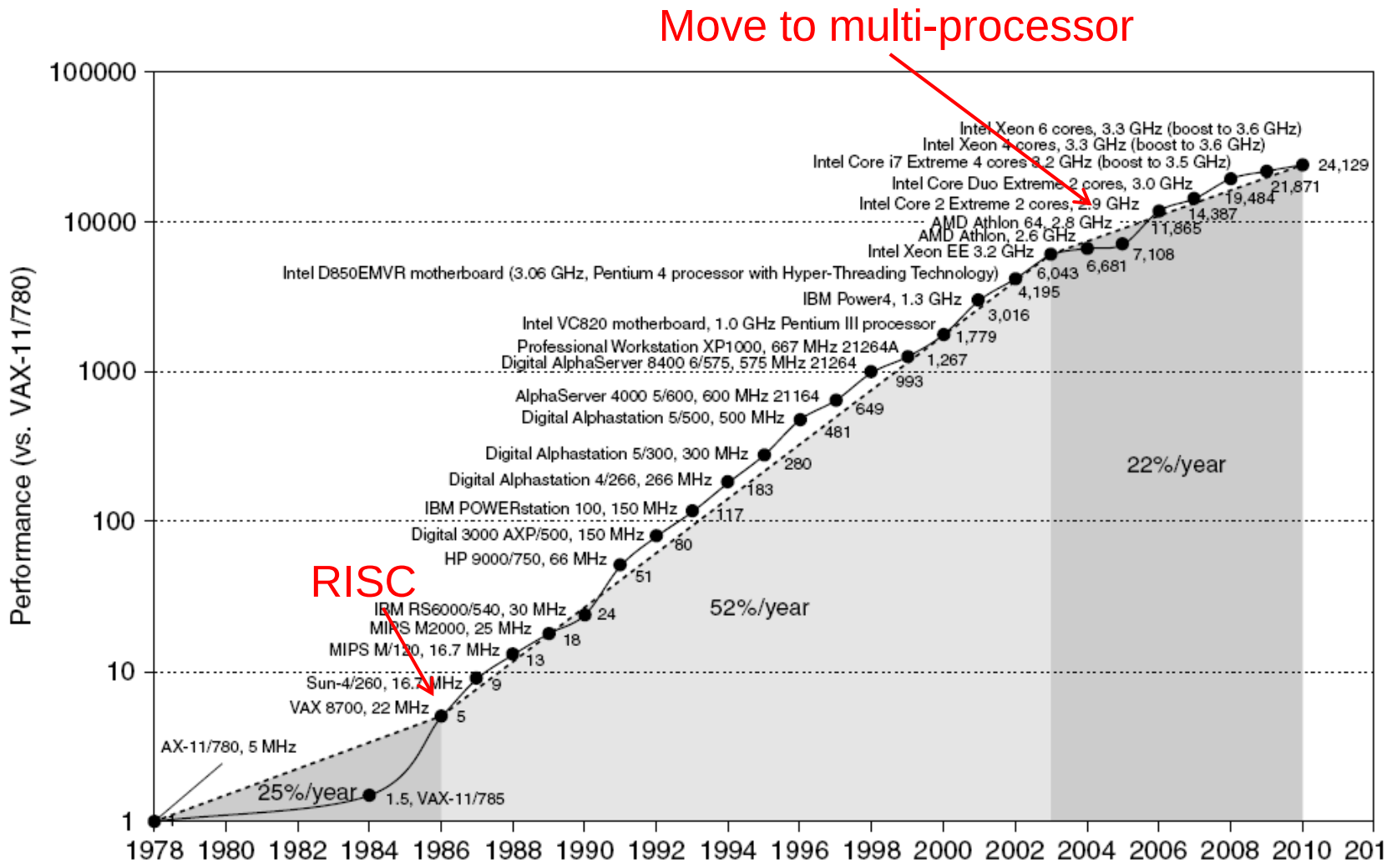
“Exponential Growth”

- Grows by a factor of (1+x) per year.
- By a factor of $(1+x)^n$ for n years.
- Example: An investment of $1000
  - 100% return in one year (i.e. doubles)
  - When will it become a million dollars?
  - Answer: $2^y=1000$, y = ?

Computer Technology

- Performance improvements:
  - Improvements in semiconductor technology
    - Feature size, clock speed
  - Improvements in computer architectures
    - Enabled by HLL compilers, UNIX
    - Lead to RISC architectures
  - Together have enabled:
    - Lightweight computers
    - Productivity-based managed/interpreted programming languages

Single Processor Performance



Defining Computer Architecture

- "Classical" view of computer architecture:
  - Instruction Set Architecture (ISA) design
  - i.e. decisions regarding:
    - registers, memory addressing, addressing modes, instruction operands, available operations, control flow instructions, instruction encoding
- "New" computer architecture:
  - Specific requirements of the target machine
  - Design to maximize performance within constraints: cost, power, and availability
  - Includes ISA, microarchitecture, hardware

Trends in Technology

- Integrated circuit technology
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year (slowing)
- Flash capacity: 50-60%/year
  - 15-20X cheaper/bit than DRAM
- Magnetic disk technology: 40%/year
  - 15-25X cheaper/bit then Flash
  - 300-500X cheaper/bit than DRAM

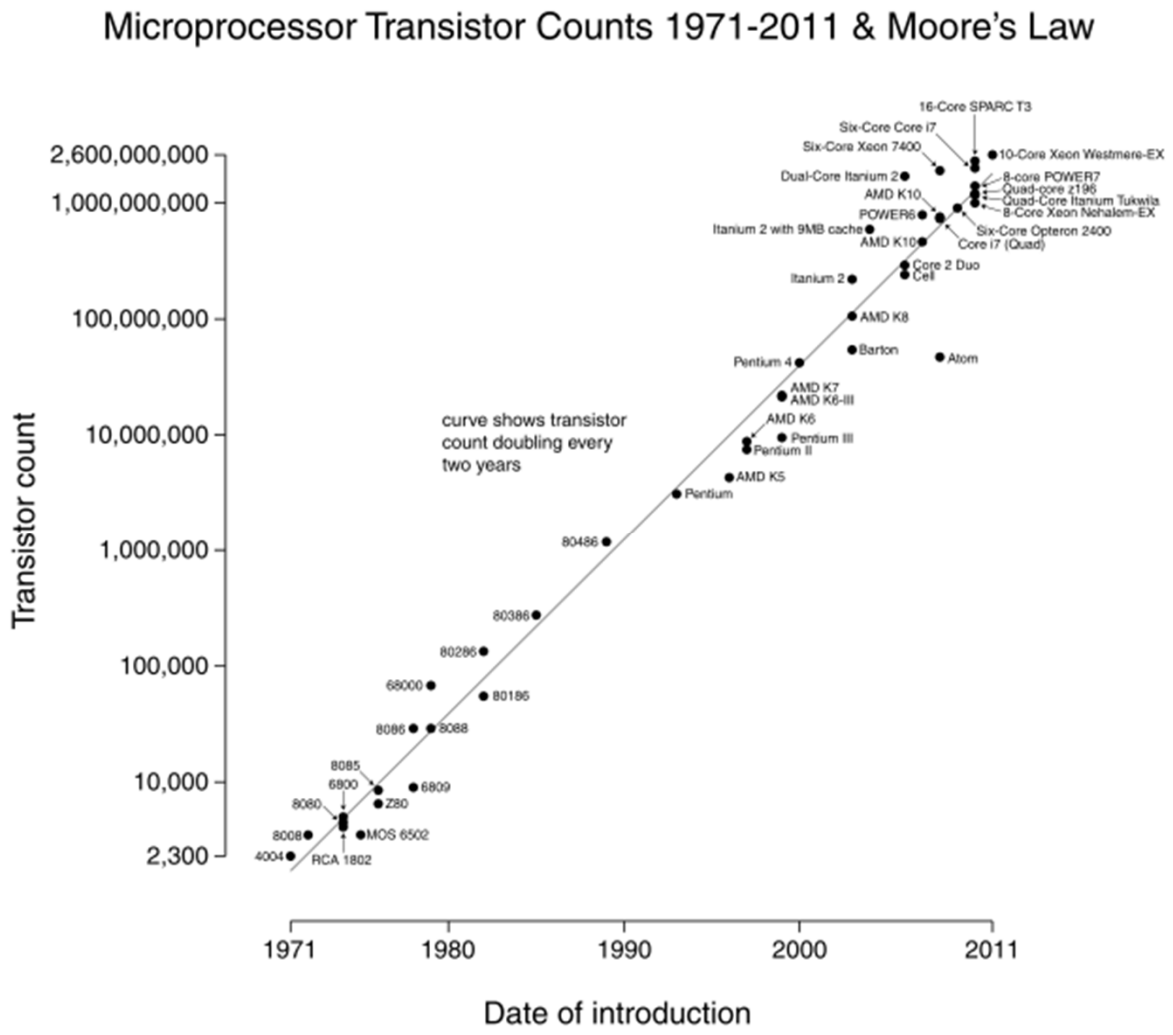
Bandwidth and Latency

- Bandwidth or throughput
  - Total work done in a given time
  - 10,000-25,000X improvement for processors
  - 300-1200X improvement for memory and disks
- Latency or response time
  - Time between start and completion of an event
  - 30-80X improvement for processors
  - 6-8X improvement for memory and disks

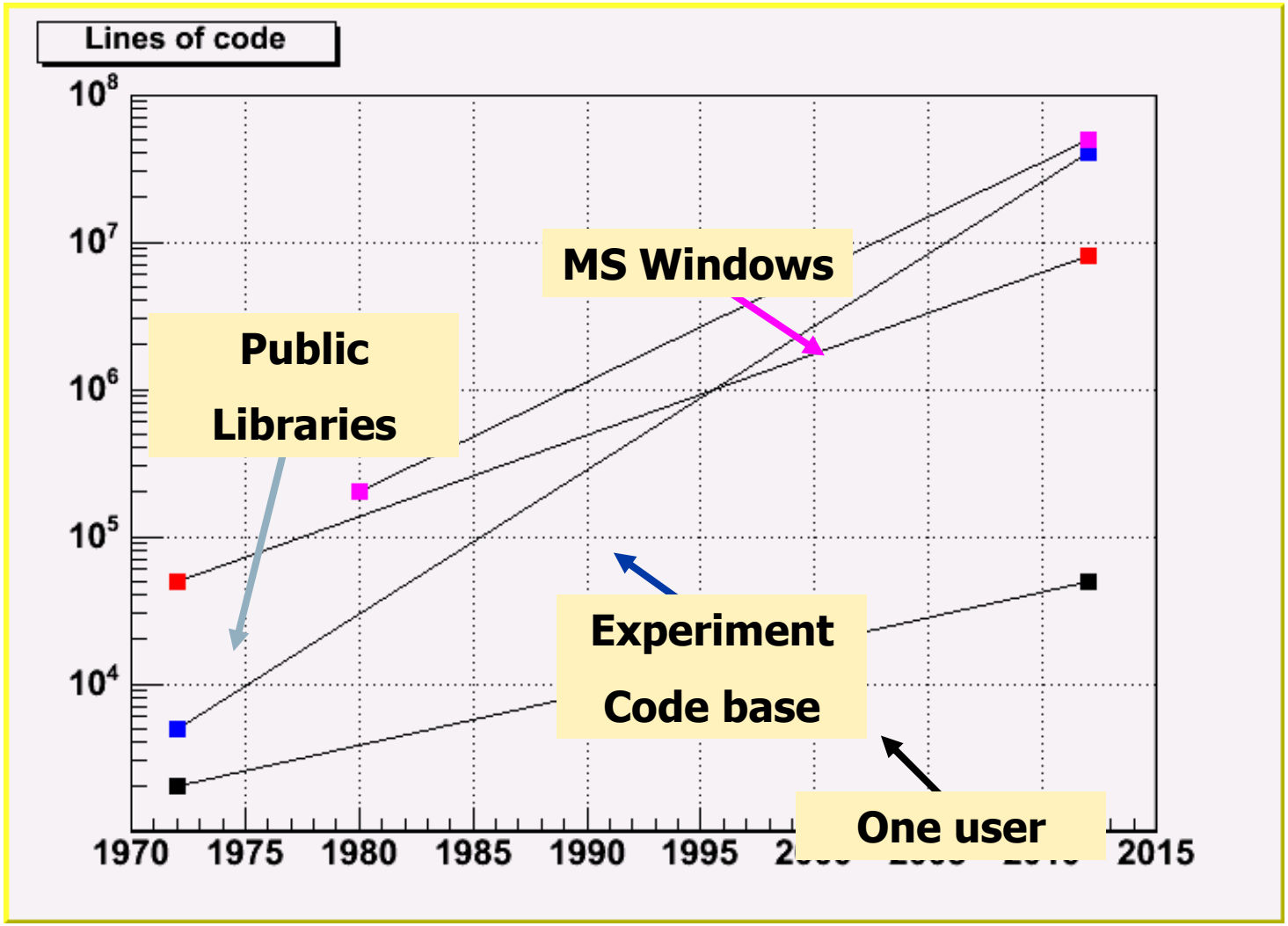
The 3 technology laws

- **Moore's Law:** formulated by Gordon Moore of Intel in the early 70's - **the number of transistors on a chip doubles every 18 months**; corollary, computers become faster and the price of a given level of computing power halves every 18 months.
- **Gilder's Law:** proposed by George Gilder, prolific author and prophet of the new technology age - **the total bandwidth of communication systems triples every twelve months**. New developments seem to confirm that bandwidth availability will continue to expand at a rate that supports Gilder's Law.
- But no laws about Software (well ! Murphy's law)

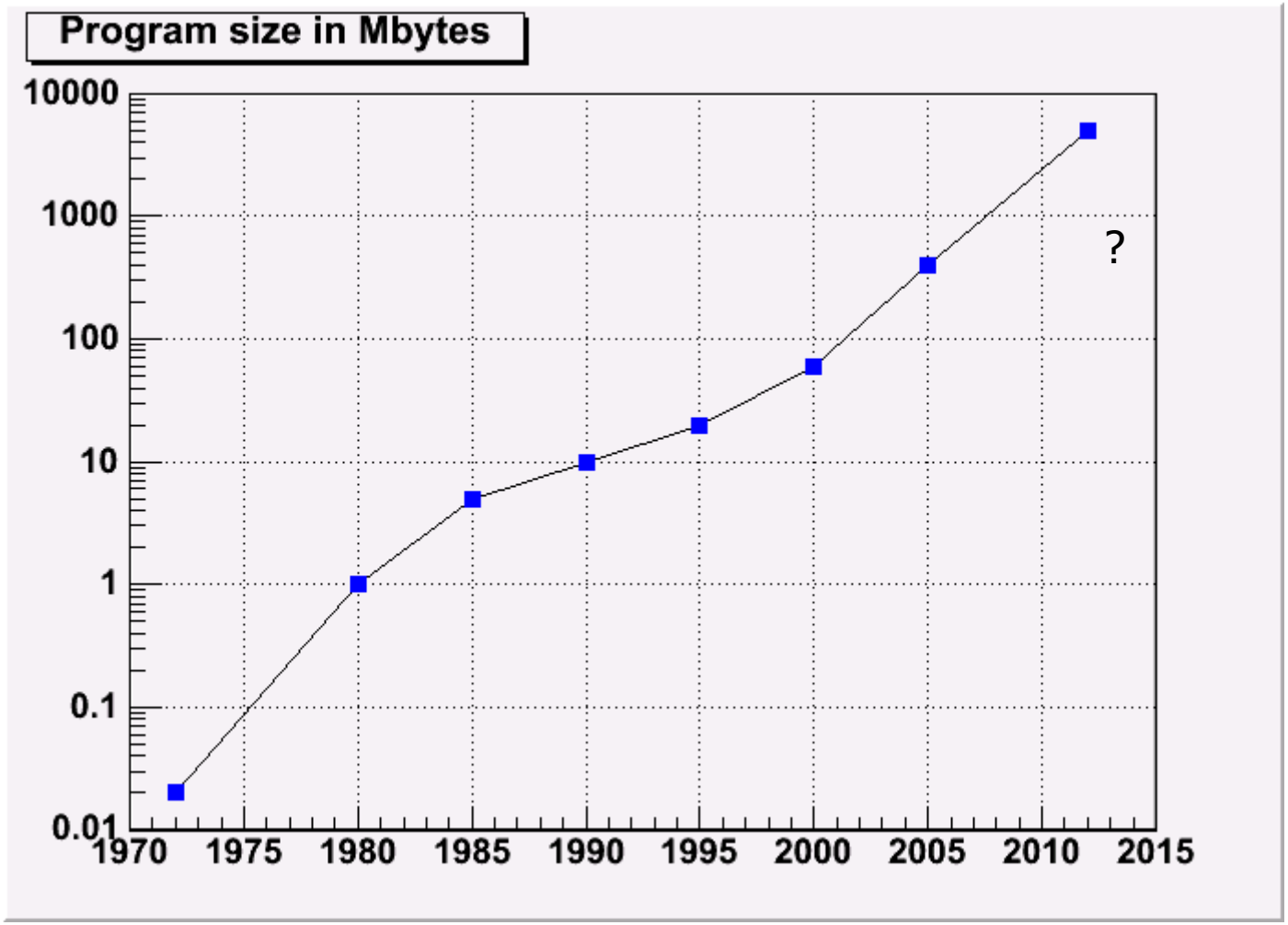
Moore’s law



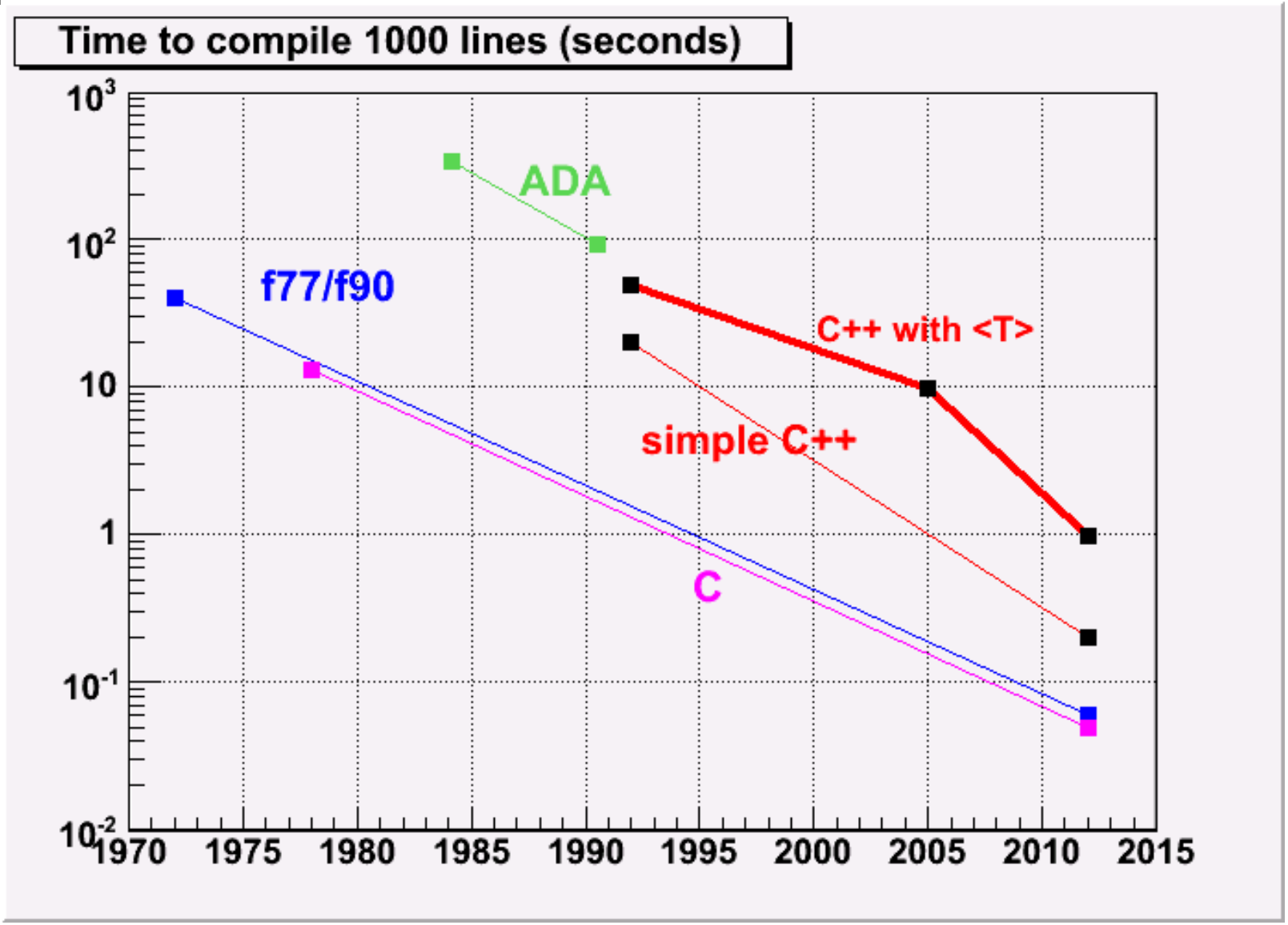
Program Size (lines of code)



Program Size (RAM)



Time to compile



Power

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air