Thinking about performance

- Let’s think some more about performance
  - How can we increase it? There are two ways, with one big caveat...

1. New processor hardware
   - Just wait and buy next year’s faster machine
   - Moore’s Law is an empirical law that fits this increase well
   - It is still true even though clock speed has leveled off since 2004
2. Program optimization
   - Use a faster (but perhaps more complex) algorithm
   - But be careful: “Premature optimization is the root of all evil”
     (Donald Knuth, “The Art of Computer Programming”)
3. Intractable problems and the P=NP question
   - Some problems seem to be inherently time consuming
   - But is it really true? The P=NP question is still unanswered!

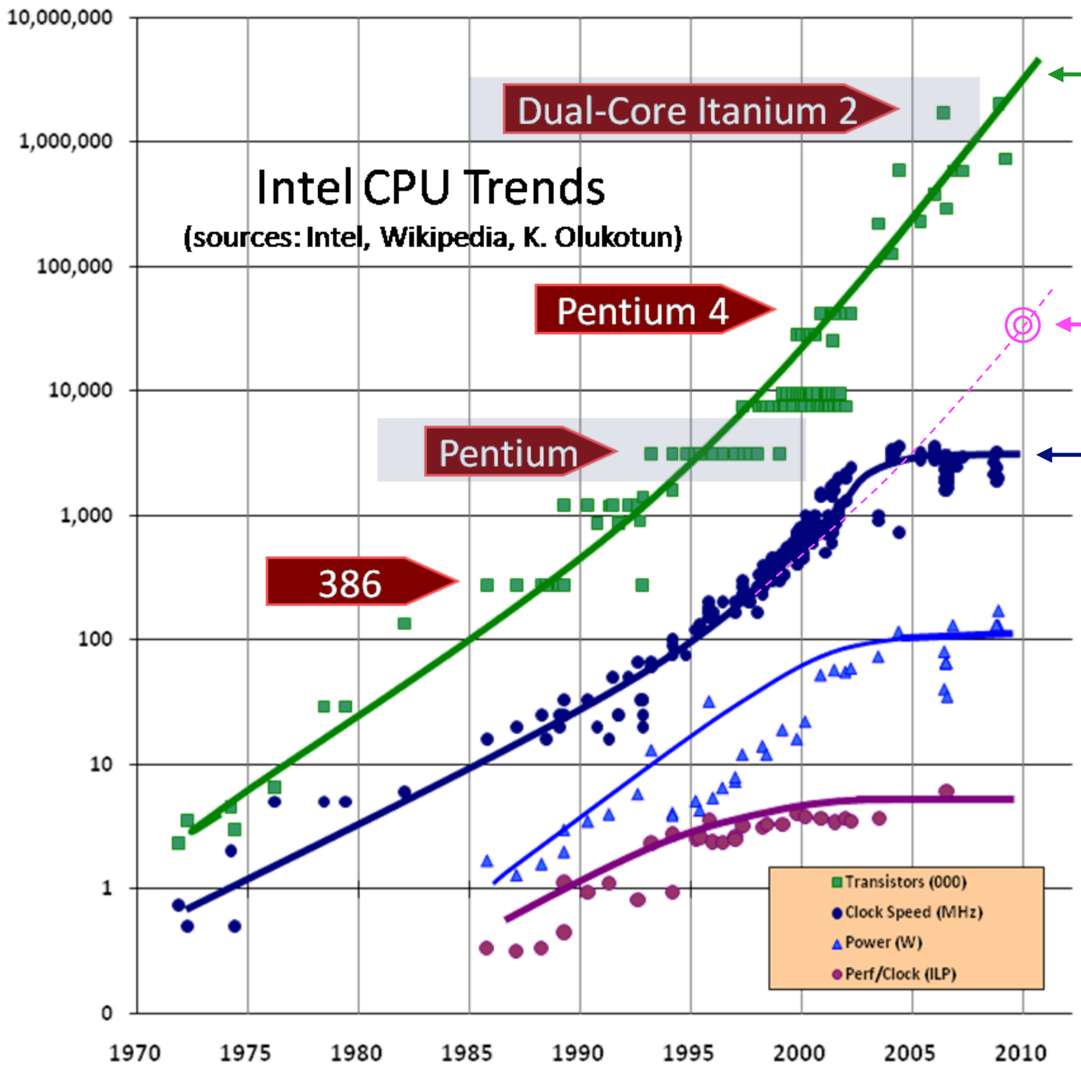
Moore’s Law (1)

- The density of integrated circuits doubles around every two years
  - First observed by Gordon E. Moore in 1965
  - This behavior continues to hold now (and will for a few years more)!
  - The origins of this law are both economic and technological

- “Performance doubles around every 18 months”
  - A false but widespread interpretation of Moore’s Law
  - This seems to be true as well

- On the other hand, clock speed does not improve in the same way!
  - We are currently at a “plateau” with a clock speed of around 3 GHz which has not increased since 2004 (with current silicon technology)
  - But circuit density is still increasing ⇒ multicore processors
  - How can we program a multicore processor? We will see later on!

Moore’s Law (2)



Moore’s Law
(still valid in 2014)

Clock ≈ 30 GHz in 2010 if it would have increased in the same way as the transistor density

Clock speed
(plateau at 3 GHz since 2004)

Should we optimize programs?

- Sometimes the performance of an algorithm is insufficient
- We can use several techniques to improve performance
  - Develop a better algorithm (requires thinking!)
  - Use a general technique such as memoization: keep the results of previous computations to avoid recomputing them
    - Memoization can convert an exponential version of the Fibonacci algorithm into a linear time version
- In general, it is possible to improve performance up to a certain point, after which the improvements get smaller and smaller with more and more complex algorithms
- "Premature optimization is the root of all evil" (D. E. Knuth)
  - Never do optimization before the need is manifest. Always start with a simple algorithm.