# Lecture 1: CS/ECE 3810 Introduction

- Today's topics:
  - Why computer organization is important
  - Logistics
  - Modern trends

# Why Computer Organization

- Yes, I know, required class…

# Why Computer Organization

- Embarrassing if you are a BS in CS/CE and can't make sense of the following terms: DRAM, pipelining, cache hierarchies, I/O, virtual memory, …

- Embarrassing if you are a BS in CS/CE and can't decide which processor to buy: 3 GHz P4  or 2.5 GHz Athlon (helps us reason about performance/power), …

- Obvious first step for chip designers, compiler/OS writers

- Will knowledge of the hardware help you write better programs?

# Must a Programmer Care About Hardware?

- Must know how to reason about program performance and energy

- Memory management: if we understand how/where data is placed, we can help ensure that relevant data is nearby

- Thread management: if we understand how threads interact, we can write smarter multi-threaded programs

→ Why do we care about multi-threaded programs?

# Example

200x speedup for matrix vector multiplication

- Data level parallelism: 3.8x
- Loop unrolling and out-of-order execution: 2.3x
- Cache blocking: 2.5x
- Thread level parallelism: 14x

# Key Topics

- Moore’s Law, power wall
- Use of abstractions
- Assembly language
- Computer arithmetic
- Pipelining
- Using predictions
- Memory hierarchies
- Reliability

# Logistics

- See class web-page
  http://www.cs.utah.edu/~rajeev/cs3810

- Sign up for the csece3810 mailing list !!

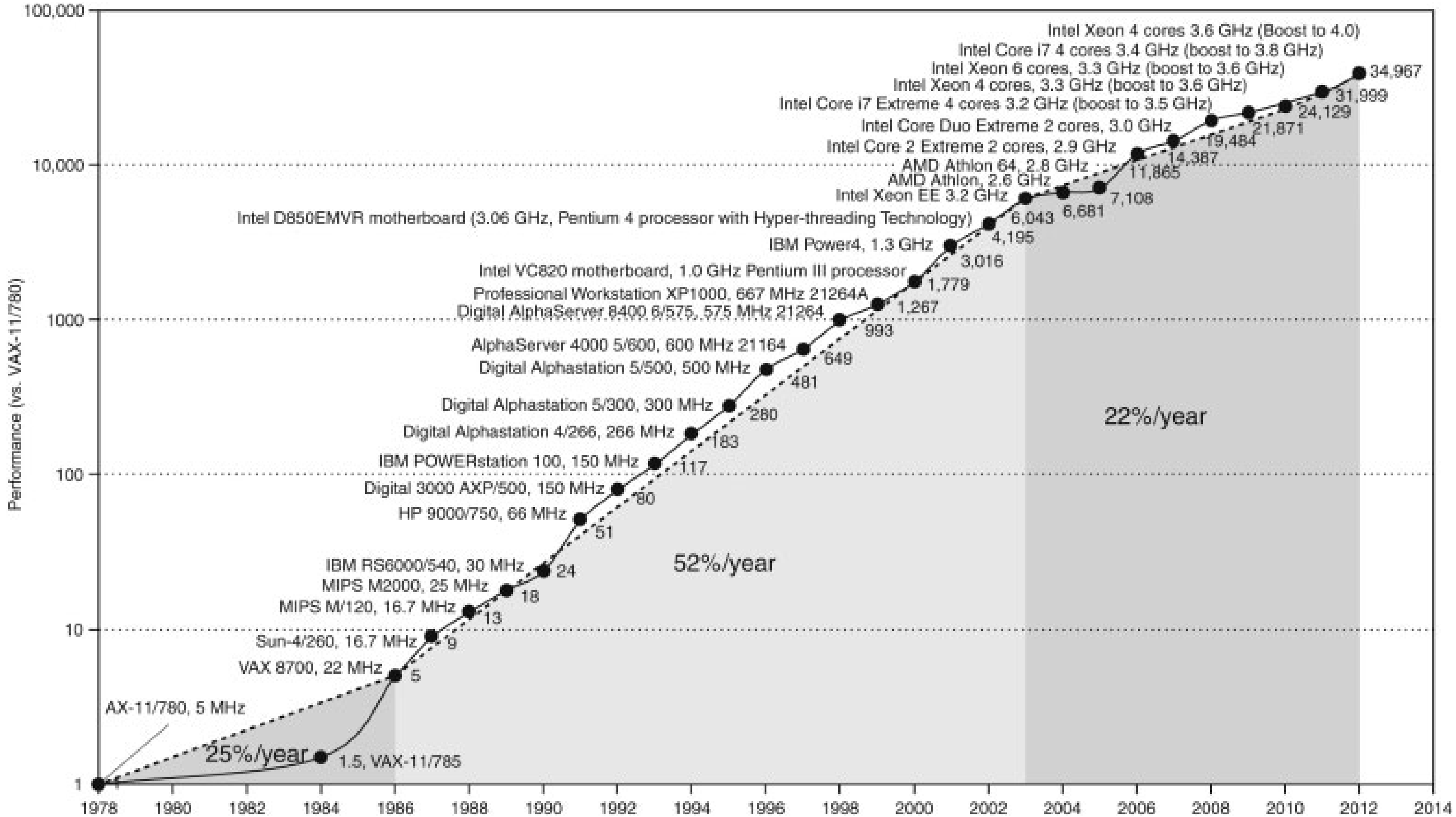
- TAs: Ben, Chandrasekhar, Marsh, Padmashree
  Office hours: TBA

- Textbook: Computer Organization – HW/SW Interface,
  Patterson and Hennessy, 5$^{th}$ edition

# Course Organization

- 30% midterm, 40% final, 30% assignments
- ~10 assignments – you may skip one; assignments due at the start of class (either paper or electronic)
- Co-operation policy: you may discuss – you may not see someone else's written matter when writing your solution
- Print slides just before class
- Screencast videos will be made available in many cases

# Microprocessor Performance




Source: H&P Textbook

50% improvement every year!!

What contributes to this improvement?

# Important Trends

- Running out of ideas to improve single thread performance
- Power wall makes it harder to add complex features
- Power wall makes it harder to increase frequency

# Important Trends

- Historical contributions to performance:
    1. Better processes (faster devices) ~20%
    2. Better circuits/pipelines ~15%
    3. Better organization/architecture ~15%

In the future, bullet-2 will help little and bullet-1 will eventually disappear!

| | Pentium | P-Pro | P-II | P-III | P-4 | Itanium | Montecito |
|---|---|---|---|---|---|---|---|
| Year | 1993 | 95 | 97 | 99 | 2000 | 2002 | 2005 |
| Transistors | 3.1M | 5.5M | 7.5M | 9.5M | 42M | 300M | 1720M |
| Clock Speed | 60M | 200M | 300M | 500M | 1500M | 800M | 1800M |

Moore's Law in action

At this point, adding transistors to a core yields little benefit

# What Does This Mean to a Programmer?

- Today, one can expect only a 20% improvement; the improvement is even lower if the program is not multi-threaded
  - A program needs many threads
  - The threads need efficient synchronization and communication
  - Data placement in the memory hierarchy is important

# Challenges for Hardware Designers

- Find efficient ways to
  - boost single-thread performance
  - improve data sharing
  - boost programmer productivity
  - manage the memory system
  - build accelerators for important kernels
  - reduce system energy per instruction

# The HW/SW Interface

Application software

Systems software
(OS, compiler)

Hardware

a[i] = b[i] + c;

Compiler

```
lw    $15, 0($2)
add   $16, $15, $14
add   $17, $15, $13
lw    $18, 0($12)
lw    $19, 0($17)
add   $20, $18, $19
sw    $20, 0($16)
```

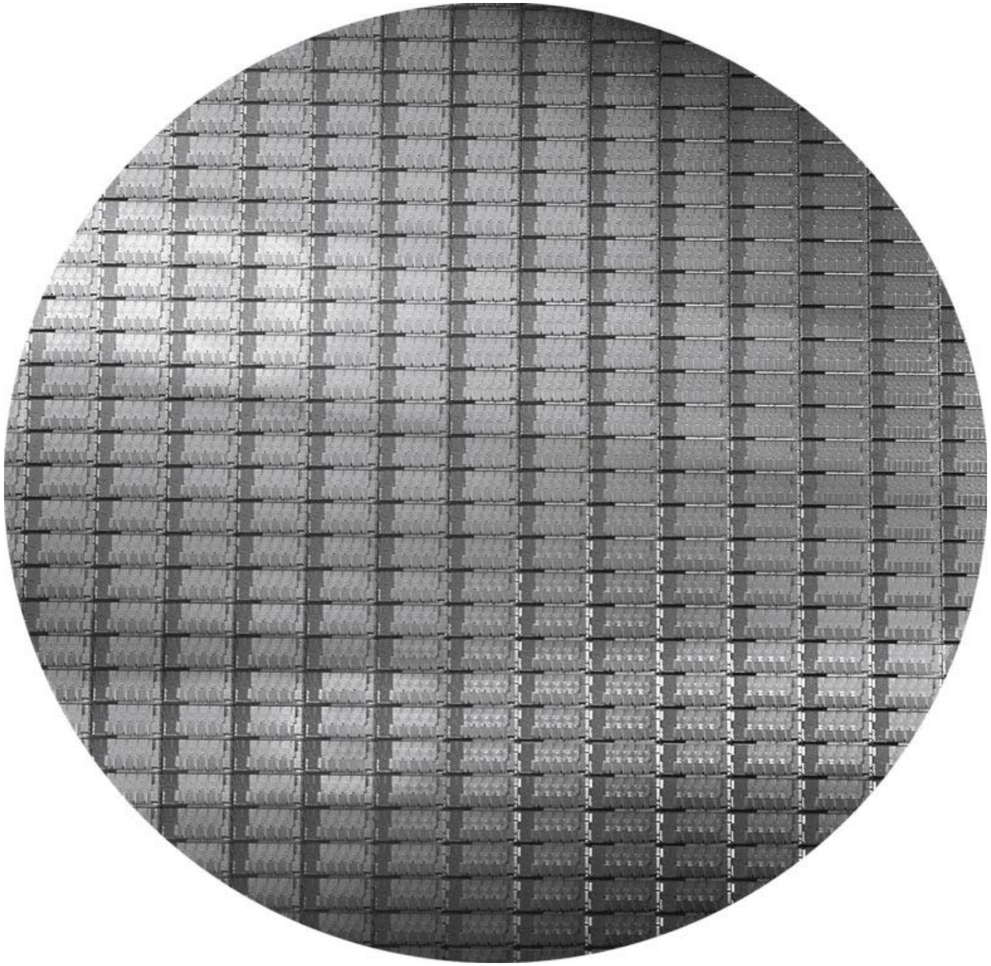
Assembler

```
000000101100000
110100000100010
…
```

# Computer Components

- Input/output devices
- Secondary storage: non-volatile, slower, cheaper
- Primary storage: volatile, faster, costlier
- CPU/processor (datapath and control)

# Wafers and Dies



Source: H&P Textbook

# Manufacturing Process

- Silicon wafers undergo many processing steps so that different parts of the wafer behave as insulators, conductors, and transistors (switches)

- Multiple metal layers on the silicon enable connections between transistors

- The wafer is chopped into many dies – the size of the die determines yield and cost

# Processor Technology Trends

- Shrinking of transistor sizes: 250nm (1997) → 130nm (2002) → 70nm (2008) → 35nm (2014)
- Transistor density increases by 35% per year and die size increases by 10-20% per year… functionality improvements!
- Transistor speed improves linearly with size (complex equation involving voltages, resistances, capacitances)
- Wire delays do not scale down at the same rate as transistor delays

# Memory and I/O Technology Trends

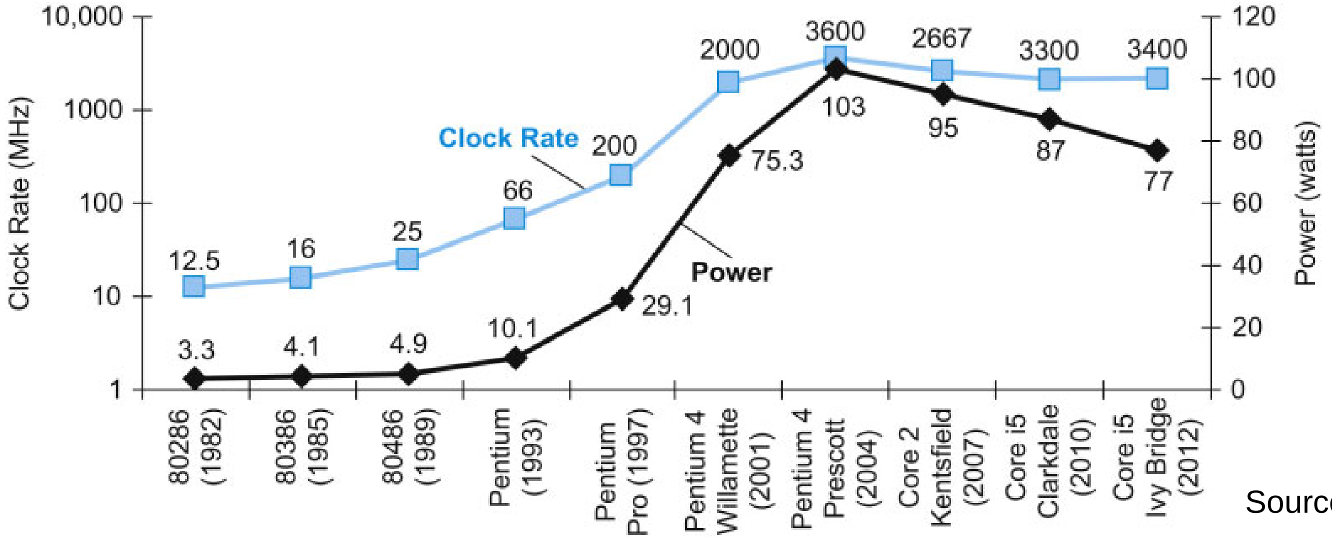
- DRAM density increases by 40-60% per year, latency has reduced by 33% in 10 years (the memory wall!), bandwidth improves twice as fast as latency decreases
- Disk density improves by 100% every year, latency improvement similar to DRAM
- Networks: primary focus on bandwidth; 10Mb → 100Mb in 10 years; 100Mb → 1Gb in 5 years

# Power Consumption Trends

- Dyn power $\alpha$ activity x capacitance x voltage$^2$ x frequency

- Voltage and frequency are somewhat constant now, while capacitance per transistor is decreasing and number of transistors (activity) is increasing

- Leakage power is also rising (function of #trans and voltage)




Source: H&P Textbook

# Next Class

- Topics: Performance, MIPS instruction set
  architecture (Chapter 2)

- Visit the class web-page
  http://www.cs.utah.edu/~rajeev/cs3810

- Sign up for the mailing list

# Title

- Bullet