# CPE 431/531

# Chapter 1 - Computer Abstractions and Technology

## Dr. Rhonda Kay Gaede

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

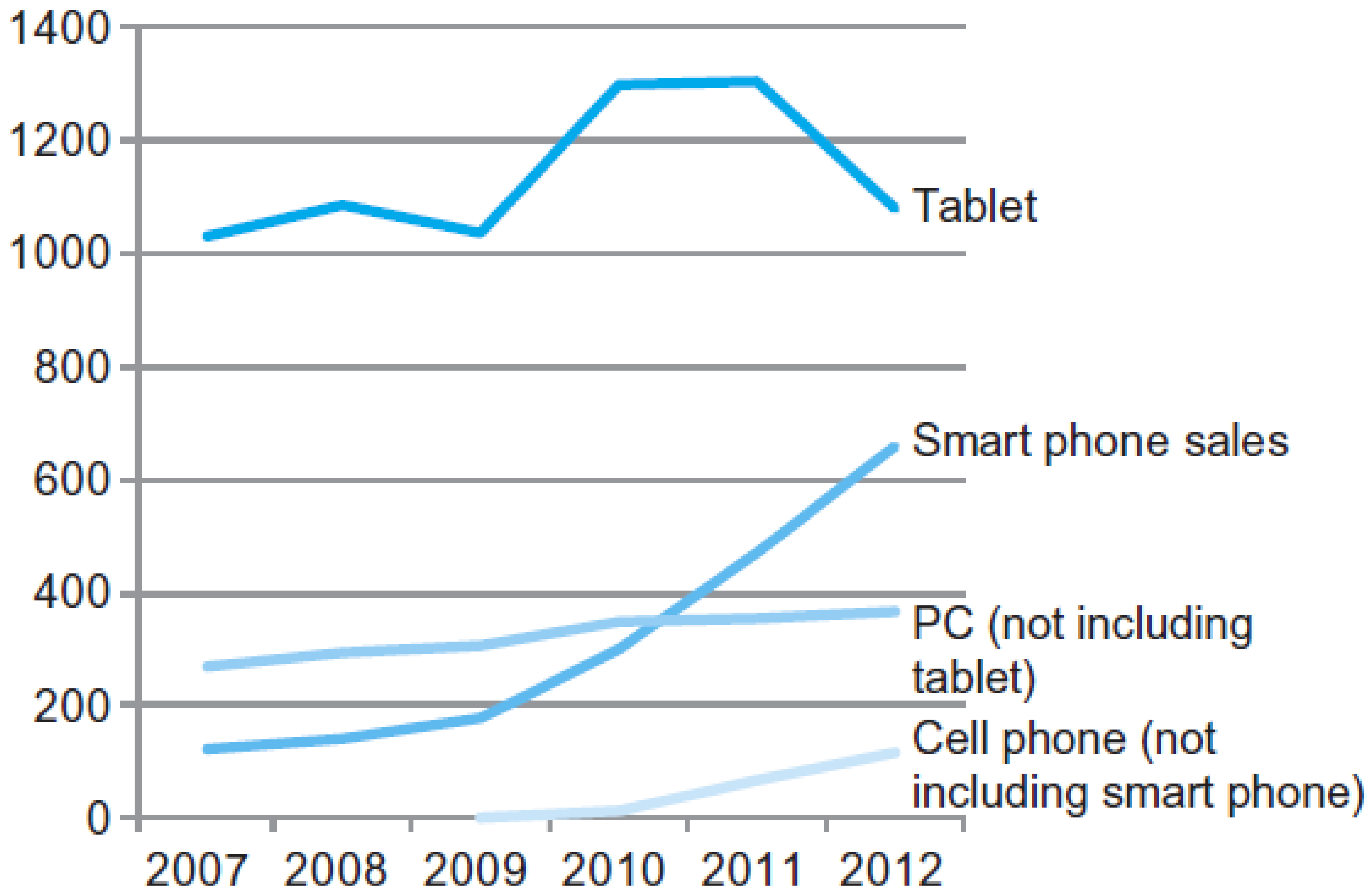
# 1.1 Introduction

- The computing industry embraces ____________ at a breathtaking rate.
- If transportation had kept pace with computing, today we could travel coast to coast in about __________for ___________
- Revolutions
    - ______________, ____________, ___________
- Recent innovations enabled by computing
    - ________________________
    - ____________
    - ____________________
- Tommorrow's Killer Apps
    - __________________
    - _________________
    - _________________

# 1.1 Classes of Computing Applications

- Personal Computers
  - ________________________________
  - ___________________________
- Servers
  - _____________________________
  - _________________
  - _________________________
- Supercomputers
  - _________________
  - ____________________
- Embedded Computers
  - _______________________________
  - ___________________________________________

# 1.1 The PostPC Era



# 1.1 What You Can Learn in This Book

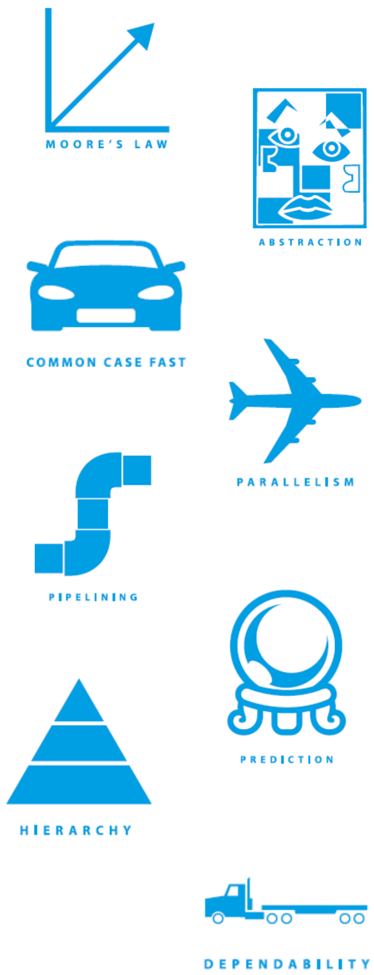
- How are programs written in a __________ language translated into ______________________________ and how does the __________execute the resulting program?
- What is the __________between the __________ and the ____________?
- What determines the _______________of a ____________?
- What techniques can be used by ________________________ to improve performance?
- What are the _________ for and the _______________ of the switch from ______________ processing to __________ processing?

# 1.1 Elements Contributing to Program Performance

- ______________________________
- ____________________________________________
- _________________________________________
- _______________________________________________________

# 1.2 Eight Great Computer Architecture Ideas

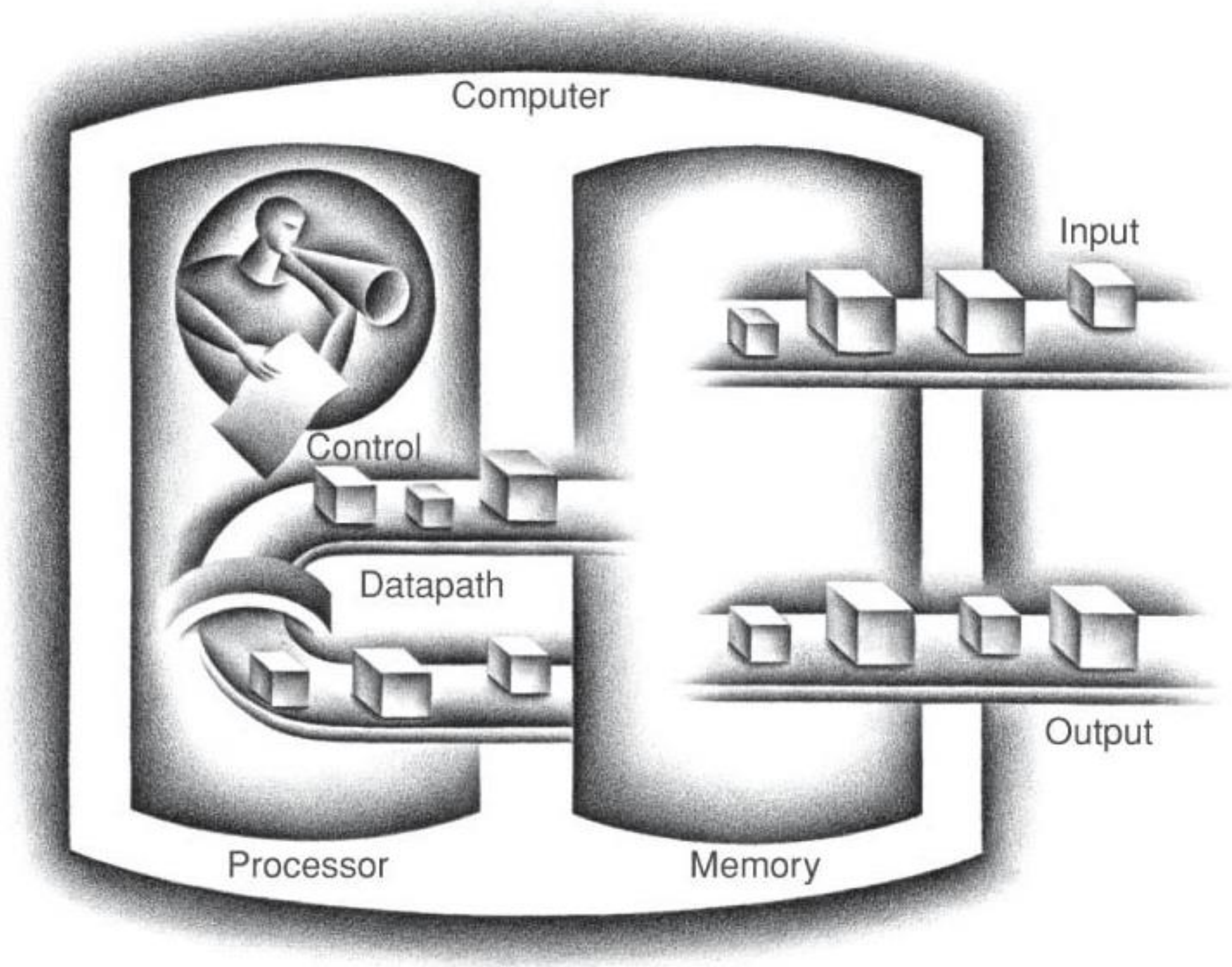
- Design for Moore’s Law
- Use abstractions to simplify design
- Make the common case fast
- Performance via Parallelism
- Performance via Pipelining
- Performance via Prediction
- Hierarchy of Memories
- Dependability via Redundancy



# 1.3 Below Your Program

- Operating system
  - Interface between ______________ and the __________
    - Handles ________ ___
    - Allocates ________ and __________
    - Provides for __________ ________
- Compiler
  - ________________________________ ________________________

# 1.4 Under the Covers – The BIG Picture

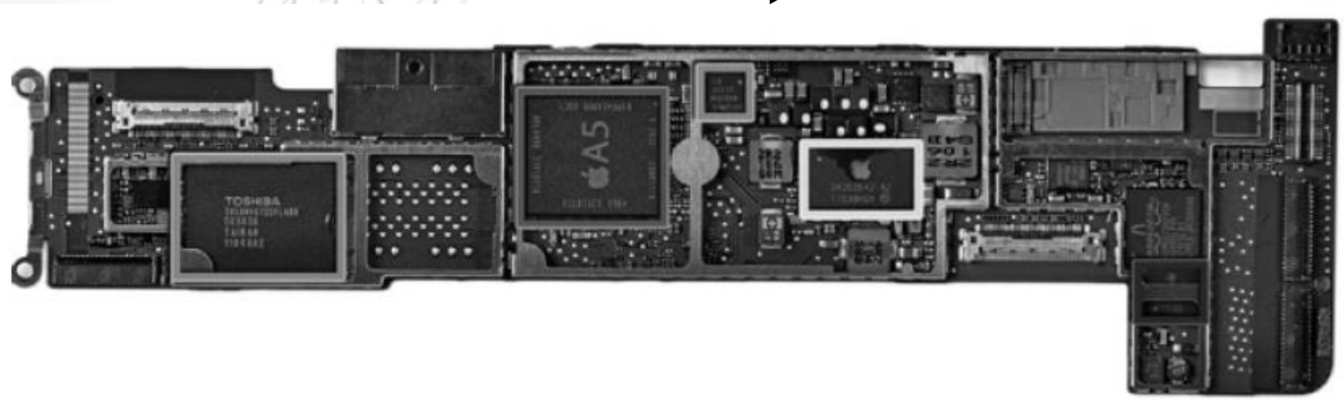


# 1.4 Under the Covers – Apple iPad 2

Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

Computer board

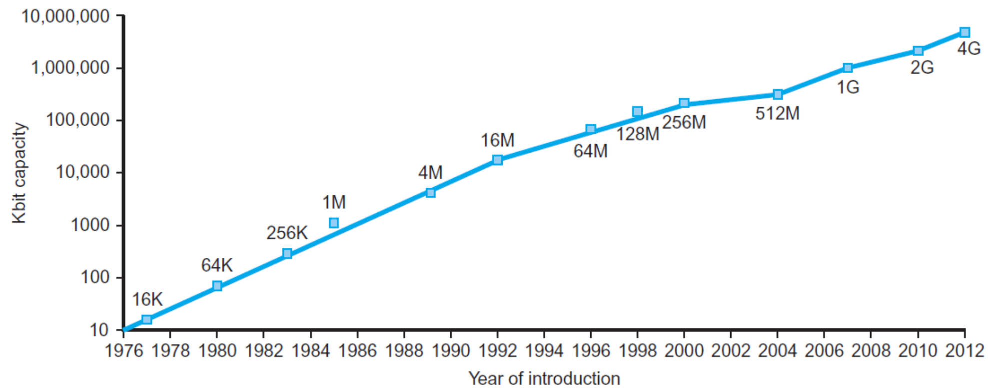


# 1.4 Communicating with Other Computers

- Network Services
  - ___________________
  - ________________
  - ___________________
- Types of Networks
  - _______ ____ network (Ethernet)
  - _______ _______ network (Fiber optic cables)
  - __________

# 1.5 Technologies Enabling Processors and Memory

- Electronics technology continues to evolve
  - Increased _________ and _______________
  - Reduced ______

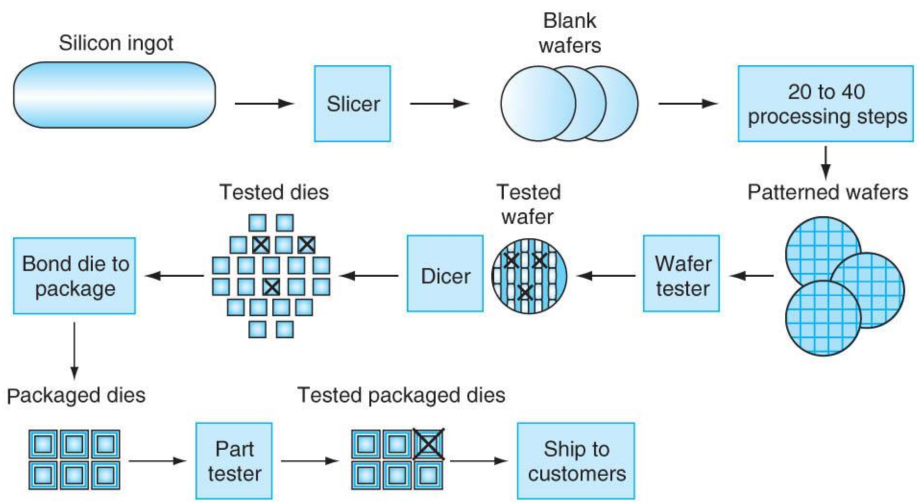


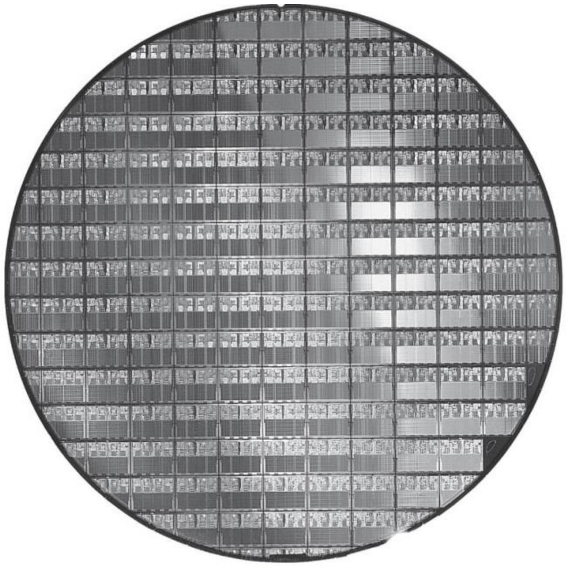

DRAM capacity

| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit (IC) | 900 |
| 1995 | Very large scale IC (VLSI) | 2,400,000 |
| 2013 | Ultra large scale IC | 250,000,000,000 |

# 1.5 Semiconductor Manufacturing

- Need different materials
  - ____________________
  - ________________
  - _____________________________



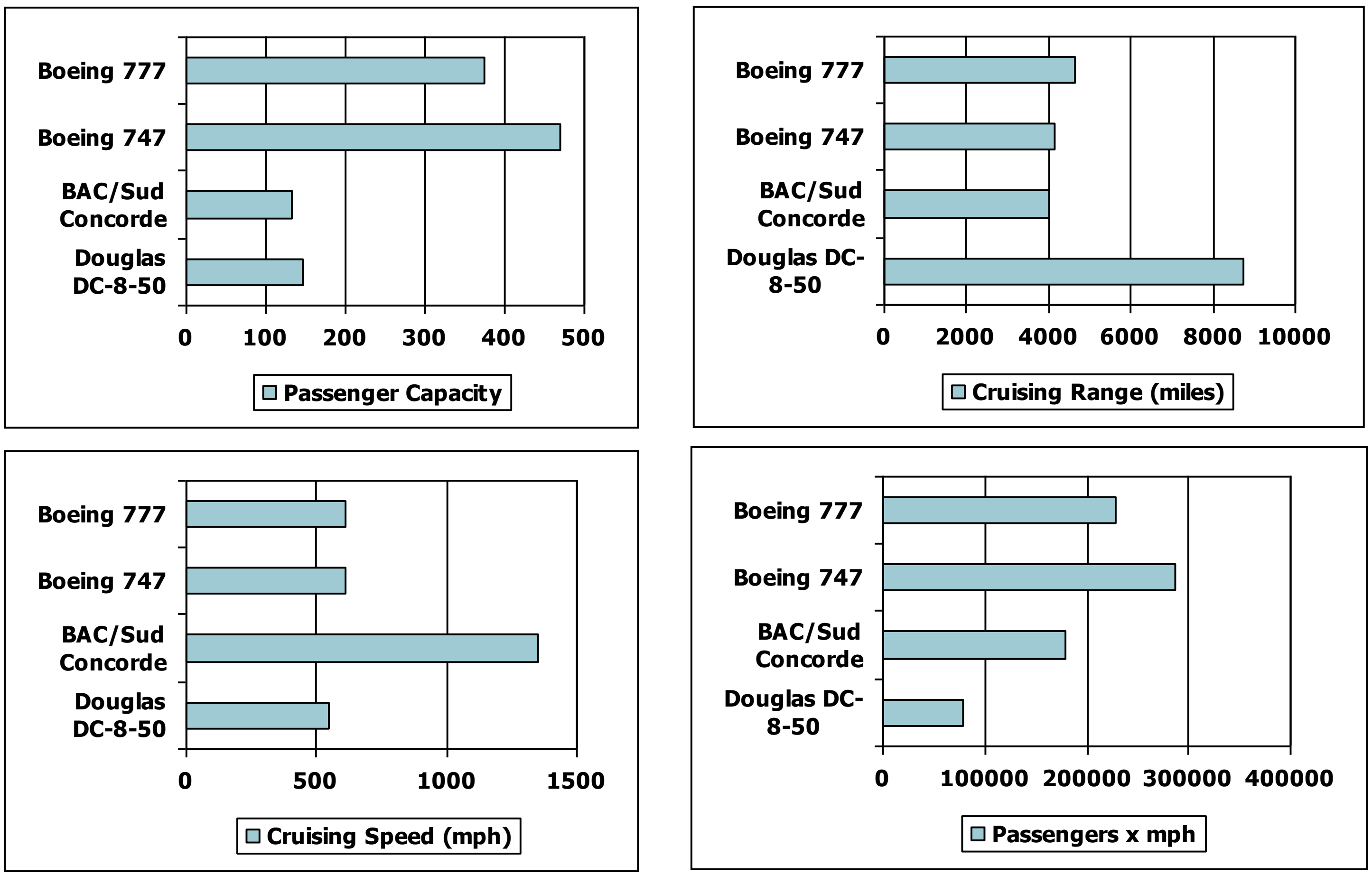



# 1.6 Performance - Motivation

- Accurately measuring and comparing different computers is critical to ____________, and therefore, to _____________.
  - We need to understand what determines the _________________ of a computer
  - _________ performance and ________ performance are linked ______________

# 1.6 Defining Performance

Which airplane has the best performance?



# 1.6 Performance Metrics

- Similarly, ambiguity exists when discussing computer performance.
  - Single program – run as quickly as possible
  - Many users – run as many programs as possible
- The user wants ___________ and the system manager wants _____________.

# 1.6 Throughput and Response Time

- Do the following changes to a computer system increase throughput, decrease response time, or both?
  - Replacing the processor in a computer with a faster version.
  - Adding additional processors to a system that uses multiple processors for separate tasks, i.e., net surfing

# 1.4 Performance Related to Execution Time

- Decreasing execution time increases performance.

- Relating performance of two computers
  - $ET_A$= 10 s, $ET_B$ = 15 s, how much faster is A than B?

# 1.6 Measuring Performance

- __________ is the measure of computer performance
  - ______________________
  - ______________________
- Users think in __________, designers think in ____________.
  - ET = CC * CT
  - ET = CC/CR

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

# 1.6 CPU Performance and its Factors

Example: Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, that will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing machine B to require 1.2 times as many clock cycles as machine A for this program. What clock rate should we tell the designer to target?

# 1.6 Instruction Performance

How does the number of instructions factor in?

Example: Suppose we have two implementations of the same instruction set architecture. Machine A has a clock cycle time of 250 ps and a CPI of 2.0 for some program, and machine B has a clock cycle of 500 ps and a CPI of 1.2 for the same program. Which machine is fastest for this program, and by how much?

# 1.6 Comparing Code Segments

Example: A compiler designer is trying to decide between two code sequences for a particular machine. The hardware designers have supplied the following CPI information and the compiler designer specifies the two segments as follows:

| Instruction class | CPI | Instruction class | CPI | Instruction class | CPI |
|---|---|---|---|---|---|
| A | 1 | B | 2 | C | 3 |

| Code Segment | IC(A) | IC(B) | IC(C) |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| 2 | 4 | 1 | 1 |

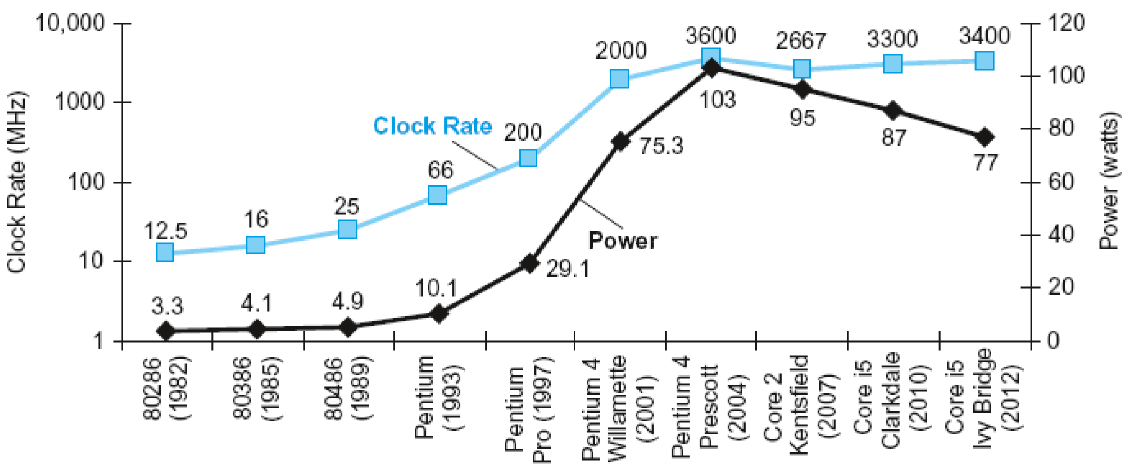
Which code segment executes the most instructions? Which will be faster? What is the CPI for each sequence?

# 1.6 Understanding Program Performance

| Component | Affects What |
|---|---|
| Algorithm | |
| Programming Language | |
| Compiler | |
| ISA | |

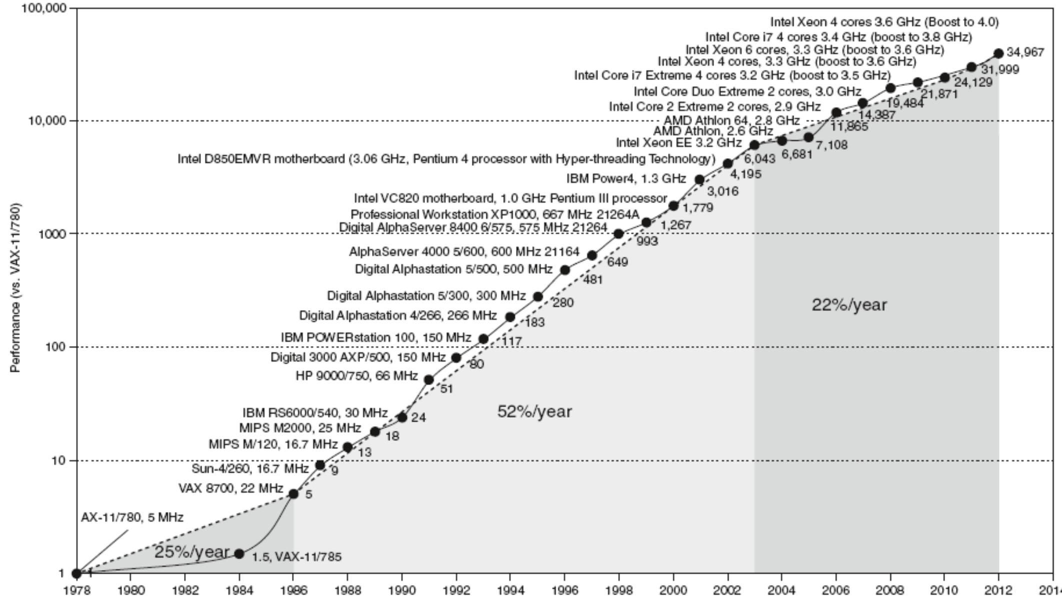
# 1.7 The Power Wall

- We have run into the practical power limit for ________ commodity processors
- For CMOS, the primary source of power dissipation is ________ power.
  - Power = Capacitive load x Voltage$^2$ x Frequency switched
- We can lower the _______ by lowering the ________
- The ________ has gone about as low as it can go, any further and there is too much ________ ________



# 1.8 The Switch from Uni- to Multi-processors

- As of 2006, all desktop and server companies started shipping ___________ processors
- The benefit is often more of ____________ than ____________ ______
- In the past, ___________ software didn't change to achieve performance gains, the underlying hardware and attendant compiler did.



# 1.8 The Switch from Uni- to Multi-processors

- Today, applications software must be rewritten to achieve performance gains.
- What's so hard about writing explicitly parallel programs?
  - ______________________________________________
    ____________________________________________
  - __________________________________________

# 1.9 RealStuff: Benchmarking the Intel Core i7

- The ______ executed by a computer form a ____________.
- A typical ______________ specifies both the ___________ run and their ________________.
- ________________ form a ____________ that the user hopes will __________ the performance of the actual computer
- SPEC (System Performance Evaluation Cooperative) is an effort funded and supported by a number of computer vendors to create ____________ sets of _______________ for modern computer systems.
- The first CPU performance benchmark appeared in 1989.
- Today, SPEC offers a dozen different benchmarks designed to test a wide variety of computing environments, the newest is SPECpower.

# 1.9 RealStuff: Benchmarking the Intel Core i7

| Description | Name | Instruction Count x $10^9$ | CPI | Clock cycle time (seconds x $10^{-9}$) | Execution Time (seconds) | Reference Time (seconds) | SPECratio |
|---|---|---|---|---|---|---|---|
| Interpreted string processing | perl | 2252 | 0.60 | 0.376 | 508 | 9770 | 19.2 |
| Block-sorting compression | bzip2 | 2390 | 0.70 | 0.376 | 629 | 9650 | 15.4 |
| GNU C compiler | gcc | 794 | 1.20 | 0.376 | 358 | 8050 | 22.5 |
| Combinatorial optimization | mcf | 221 | 2.66 | 0.376 | 221 | 9120 | 41.2 |
| Go game (AI) | go | 1274 | 1.10 | 0.376 | 527 | 10490 | 19.9 |
| Search gene sequence | hmmer | 2616 | 0.60 | 0.376 | 590 | 9330 | 15.8 |
| Chess game (AI) | sjeng | 1948 | 0.80 | 0.376 | 586 | 12100 | 20.7 |
| Quantum computer simulation | libquantum | 659 | 0.44 | 0.376 | 109 | 20720 | 190.0 |
| Video compression | h264avc | 3793 | 0.50 | 0.376 | 713 | 22130 | 31.0 |
| Discrete event simulation library | omnetpp | 367 | 2.10 | 0.376 | 290 | 6250 | 21.5 |
| Games/path finding | astar | 1250 | 1.00 | 0.376 | 470 | 7020 | 14.9 |
| XML parsing | xalancbmk | 1045 | 0.70 | 0.376 | 275 | 6900 | 25.1 |
| Geometric mean | – | – | – | – | – | – | 25.7 |

# 1.9 RealStuff: Benchmarking the Intel Xeon

| Target Load % | Performance (ssj_ops) | Average Power (Watts) |
|---|---|---|
| 100% | 865,618 | 258 |
| 90% | 786,688 | 242 |
| 80% | 698,051 | 224 |
| 70% | 607,826 | 204 |
| 60% | 521,391 | 185 |
| 50% | 436,757 | 170 |
| 40% | 345,919 | 157 |
| 30% | 262,071 | 146 |
| 20% | 176,061 | 135 |
| 10% | 86,784 | 121 |
| 0% | 0 | 80 |
| Overall Sum | 4,787,166 | 1,922 |
| $\Sigma$ssj_ops/$\Sigma$power = | | 2,490 |

# 1.10 Fallacies and Pitfalls

- Pitfall: Expecting the improvement of one aspect of a computer to increase overall performance by an amount proportional to the size of the improvement.
- Fallacy: Computers at low utilization use little power.
- Fallacy: Designing for performance and designing for energy efficiency are unrelated goals.
- Pitfall: Using a subset of the performance equation as a performance metric.

# 1.11 Concluding Remarks

- Computers continue to improve in ____ and _____________
- Both hardware and software designers use _____________
- An ______________________________ may have multiple implementations.

$$\frac{Seconds}{Program} = \frac{Instructions}{Program} \times \frac{Clock\ cycles}{Instruction} \times \frac{Seconds}{Clock\ cycle}$$

- A key technology for modern processors is ________
- Key ideas are exploiting ______________ in a program using __________ processors and exploiting __________ __ ________ with a _______ _____________ using ______.
- _______ ____________ has replaced ___ _____ as the most critical resource of processor design.