ece.uah.edu/~gaede/cpe437
comparch15f
chargernatron

CPE 431/531

Chapter 1 - Computer Abstractions and Technology

Dr. Rhonda Kay Gaede

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

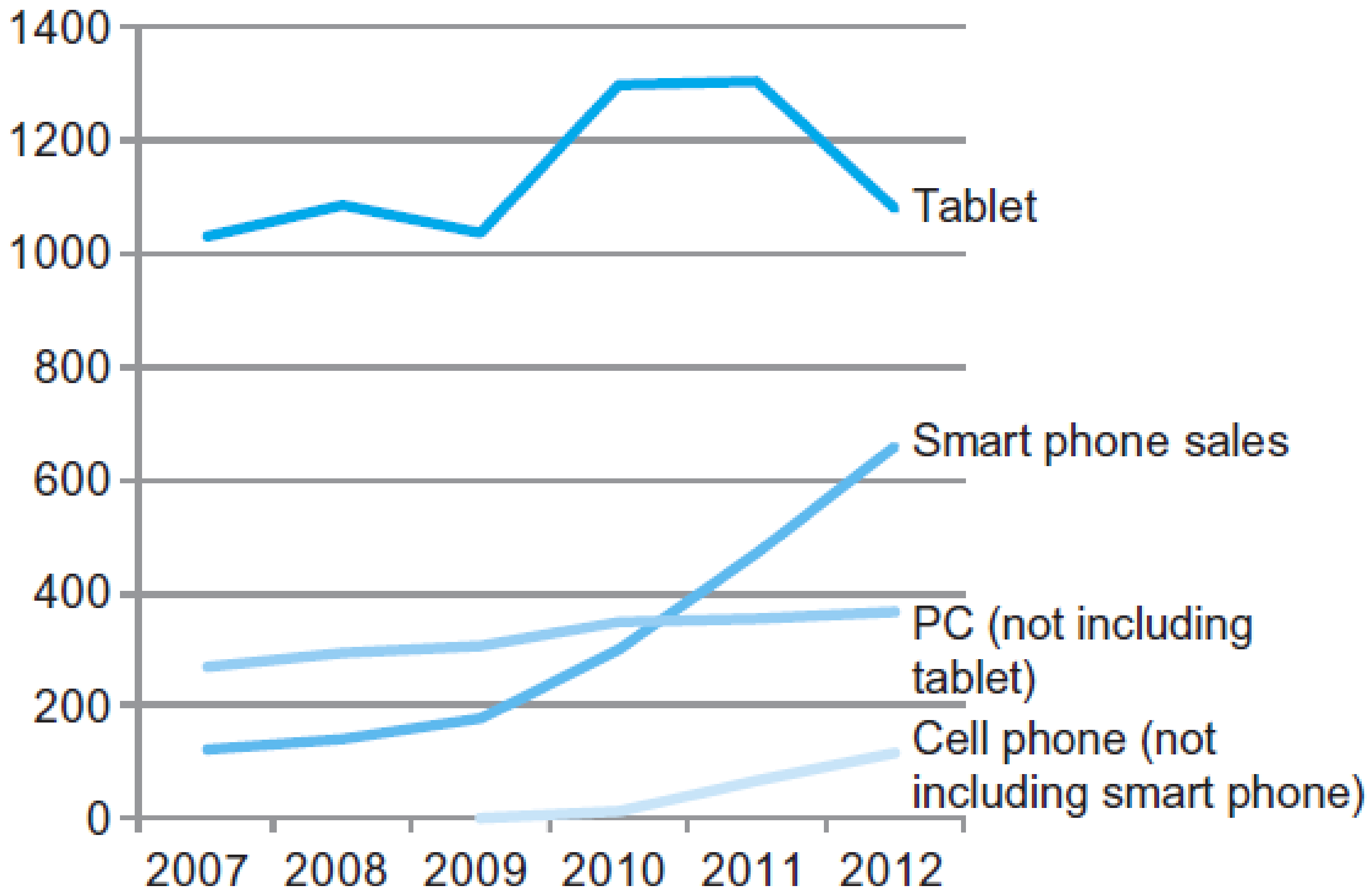
1.1 Introduction

- The computing industry embraces ___innovation___ at a breathtaking rate.
- If transportation had kept pace with computing, today we could travel coast to coast in about ___a second___ for ___a penny___
- Revolutions
  - ___agricultural___, ___industrial___, ___information___
- Recent innovations enabled by computing
  - ___computers in automobiles___
  - ___cell phones___
  - ___human genome project___
- Tommorrow's Killer Apps
  - ___augmented reality___ Google glass
  - ___cashless society___
  - ___driverless cars___

1.1 Classes of Computing Applications

- Personal Computers
  - good performance for a single user
  - execute third party software
- Servers
  - oriented to carry large workloads
  - network based
  - expandability, dependability
- Supercomputers
  - highest capability
  - small market segment
- Embedded Computers
  - hidden as components of systems
  - stringent power/performance/cost constraints

1.1 The PostPC Era



1.1 What You Can Learn in This Book

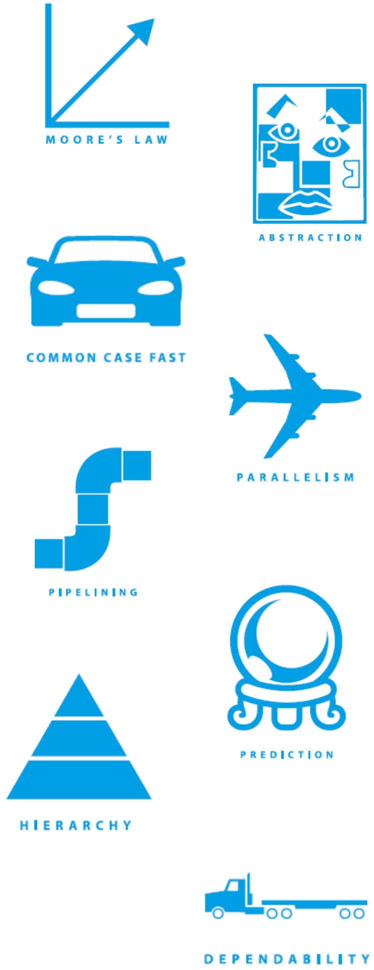
- How are programs written in a _high-level_ language translated into _language of the hardware_ and how does the _hardware_ execute the resulting program?
- What is the _interface_ between the _hardware_ and the _software_?
- What determines the _performance_ of a _program_?
- What techniques can be used by _hardware designers_ to improve performance?
- What are the _reasons_ for and the _consequence_ of the switch from _sequential_ processing to _parallel_ processing?

1.1 Elements Contributing to Program Performance

- The algorithm chosen
- The programming language and compiler
- The processor and memory system
- The I/O system (hardware and operating system)

1.2 Eight Great Computer Architecture Ideas

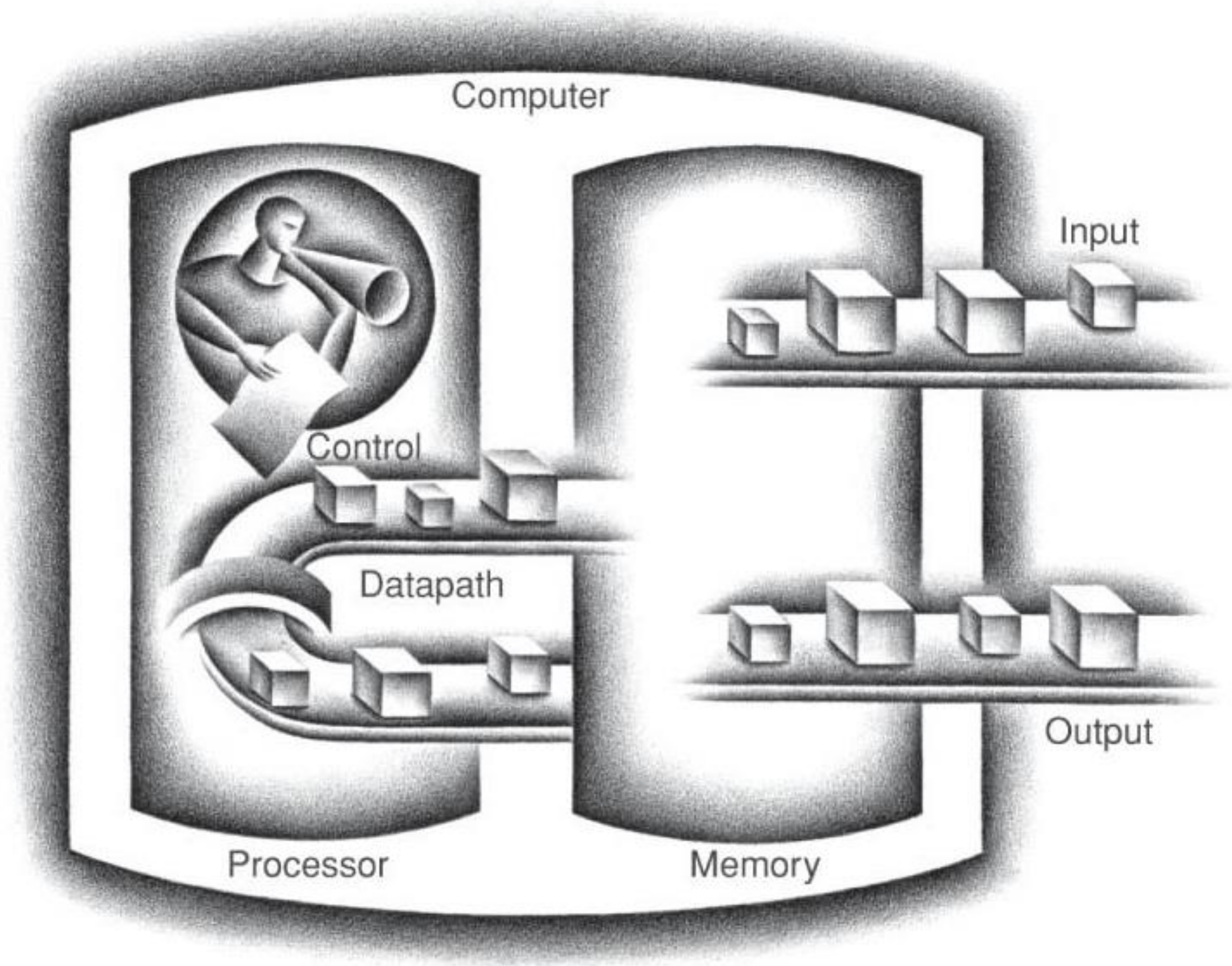
- Design for Moore’s Law
- Use abstractions to simplify design
- Make the common case fast
- Performance via Parallelism
- Performance via Pipelining
- Performance via Prediction
- Hierarchy of Memories
- Dependability via Redundancy



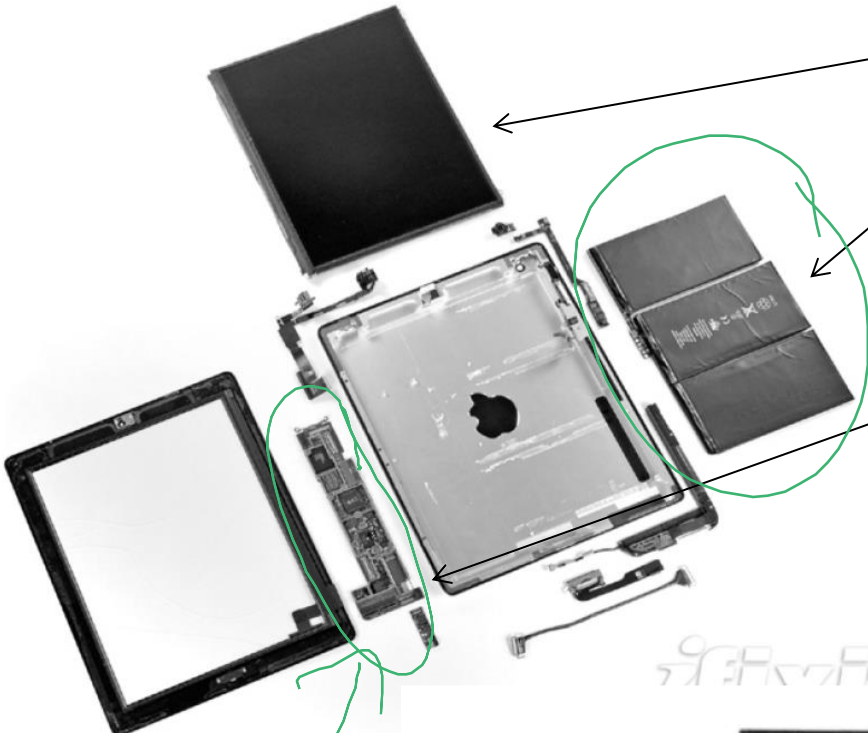
1.3 Below Your Program

- Operating system
  - Interface between user program and the hardware
    - Handles basic I/O
    - Allocates storage and memory
    - Provides for protected sharing
- Compiler
  - Translator ~~to~~ from high level language to machine language

1.4 Under the Covers – The BIG Picture



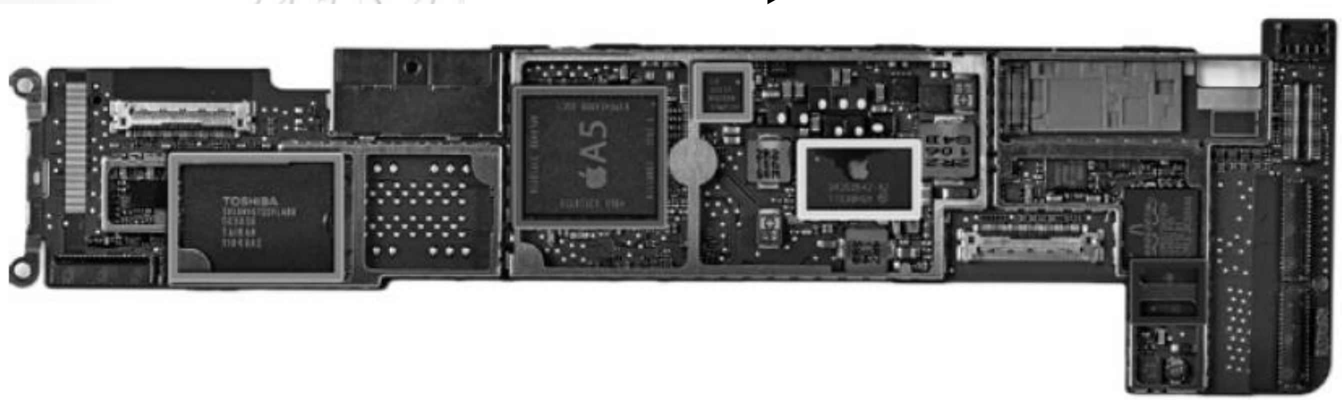
1.4 Under the Covers – Apple iPad 2



Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

Computer board

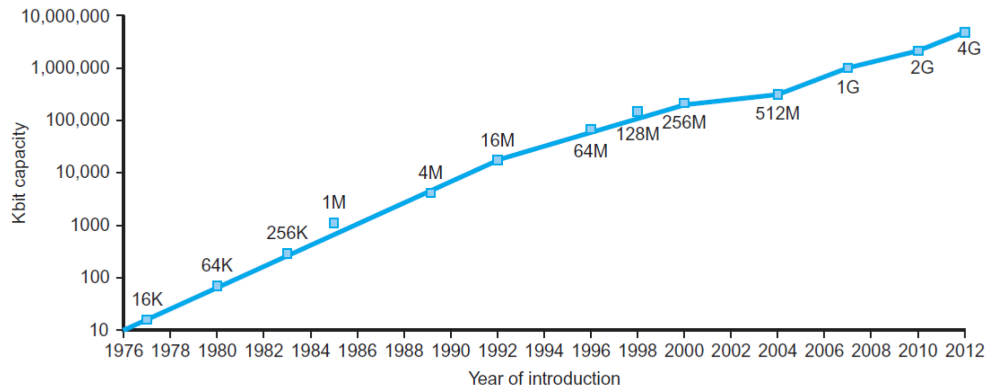


1.4 Communicating with Other Computers

- Network Services
  - Communication
  - Resource Sharing
  - Nonlocal access
- Types of Networks
  - Local area network (Ethernet)
  - Wide area network (Fiber optic cables)
  - Wireless

1.5 Technologies Enabling Processors and Memory

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost

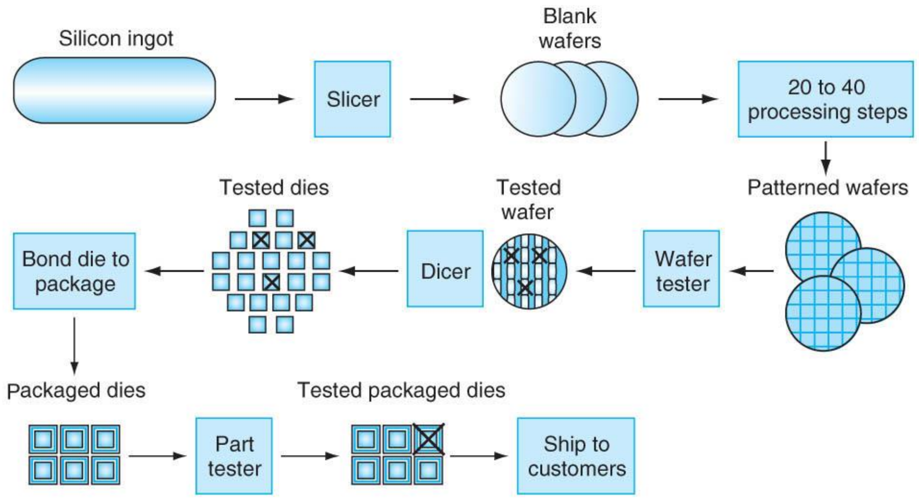


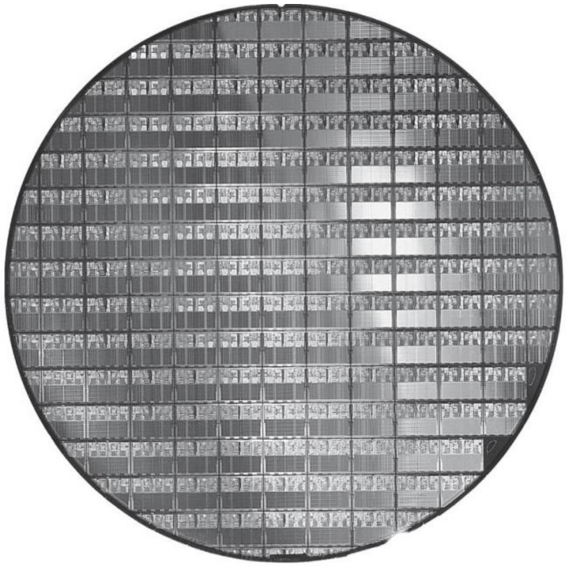
DRAM capacity

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
  - Conductor
  - Insulator
  - Semiconductor



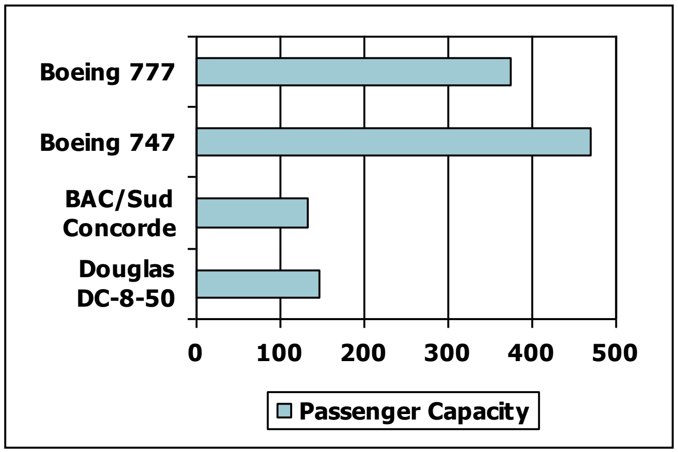


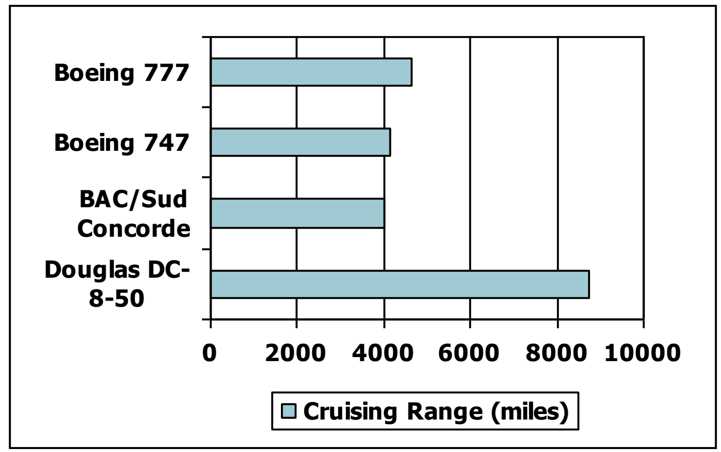
1.6 Performance - Motivation

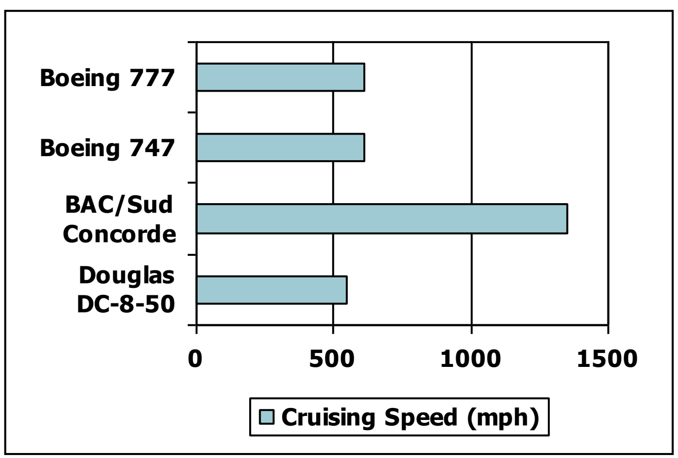
- Accurately measuring and comparing different computers is critical to __purchasers___, and therefore, to __designers____.
  - We need to understand what determines the ___performance_____ of a computer
  - __Hardware_ performance and _software_ performance are linked _symbiotically___

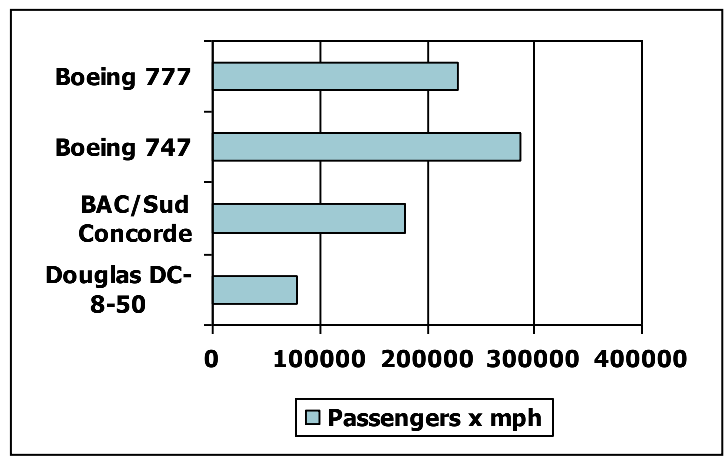
1.6 Defining Performance

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 response time and the system manager wants throughput.

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.

    Both

  - Adding additional processors to a system that uses multiple processors for separate tasks, i.e., net surfing

    throughput

1.4 Performance Related to Execution Time

- Decreasing execution time increases performance.

$ET \quad P \qquad ET = \frac{1}{P} \qquad P = \frac{1}{ET}$

$P_1 > P_2$

$\frac{1}{ET_1} > \frac{1}{ET_2} \qquad ET_2 > ET_1$

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

$\frac{P_A}{P_B} = \frac{\frac{1}{ETA}}{\frac{1}{ETB}} = \frac{ETB}{ETA} = \frac{15s}{10s} = 1.5 \qquad \frac{3}{2}$

$\frac{P_B}{P_A} = \frac{\frac{1}{ETB}}{\frac{1}{ETA}} = \frac{ETA}{ETB} = \frac{10s}{15s} = 0.6\overline{6} \qquad \frac{2}{3}$

Use ratios not percentages!!!

1.6 Measuring Performance

- __Time_____ is the measure of computer performance
  - __Elapsed time__________
  - __CPU time____________
- Users think in __seconds___, designers think in _clock cycles__.
  - ET = CC * CT   clock cycles   cycle time
  - ET = CC/CR

clock rate

$CC = \frac{1}{CR}$

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?

$ET_A = 10s$ $ET_B = 6s$ $CC_B = 1.2 CC_A$

$CR_A = 2 GHz$ $CR_B = ?$

$ET = CC * CT$

$ET_A = CC_A * CT_A$

$10s = CC_A * 0.5 ns/cycle$

$CC_A = 2 \times 10^{10}$ cycles

$$6s = \frac{1.2 \times (2 \times 10^{10} \text{ cycles})}{CR_B}$$

$$CR_B = \frac{2.4 \times 10^{10} \text{ cycles}}{6 \text{ s}}$$

$= 4$ GHz

1.6 Instruction Performance

How does the number of instructions factor in?

$ET = CC * CT$

CPI = cycles per instruction

$ET = (IC * CPI) * CT$

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?

$IC_A = IC_B$

$CT_A = 250 ps$

$CPI_A = 2.0$

$CT_B = 500 ps$

$CPI_B = 1.2$

$$\frac{P_A}{P_B} = \frac{\frac{1}{ET_A}}{\frac{1}{ET_B}} = \frac{ET_B}{ET_A} = \frac{\cancel{IC_B} * CPI_B * CT_B}{\cancel{IC_A} * CPIA * CTA}$$

$$= \frac{1.2 * \cancel{500} \cancel{ps}}{2.\cancel{0} \times \cancel{250} \cancel{ps}}$$

$$= 1.2$$

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)	IC
1	2	1	2	5
2	4	1	1	6

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

$CC_1 = 2*1 + 1*2 + 2*3 = 10$ $\quad CPI = 2$

$CC_2 = 4*1 + 1*2 + 1*3 = 9$ $\quad$ faster $\quad CPI = 1.5$

1.6 Understanding Program Performance

IC, CPI, CR

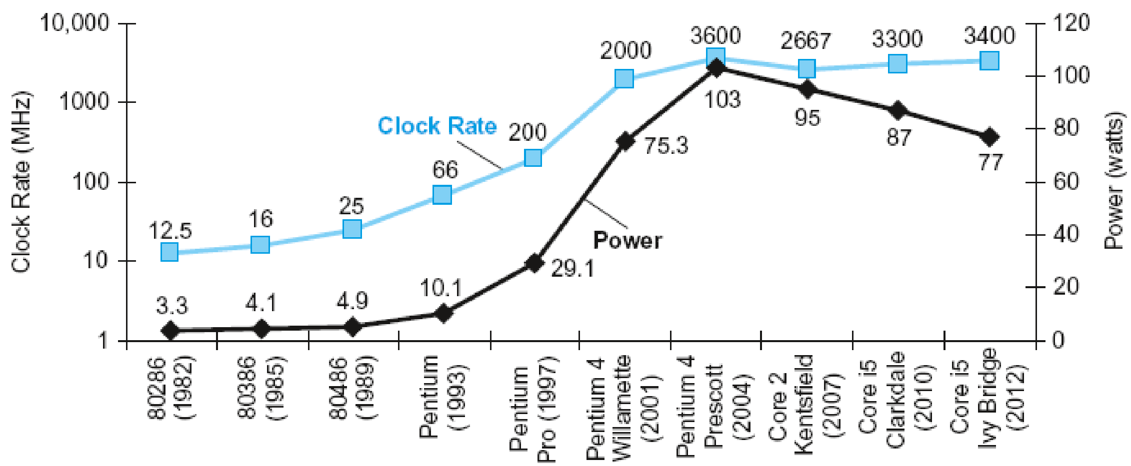
Component	Affects What
Algorithm	IC, maybe CPI
Programming Language	IC, CPI
Compiler	IC, CPI
ISA	CPI, CR, IC

1.7 The Power Wall

- We have run into the practical power limit for cooling commodity processors
- For CMOS, the primary source of power dissipation is dynamic power.
  - Power = Capacitive load x Voltage^2 x Frequency switched

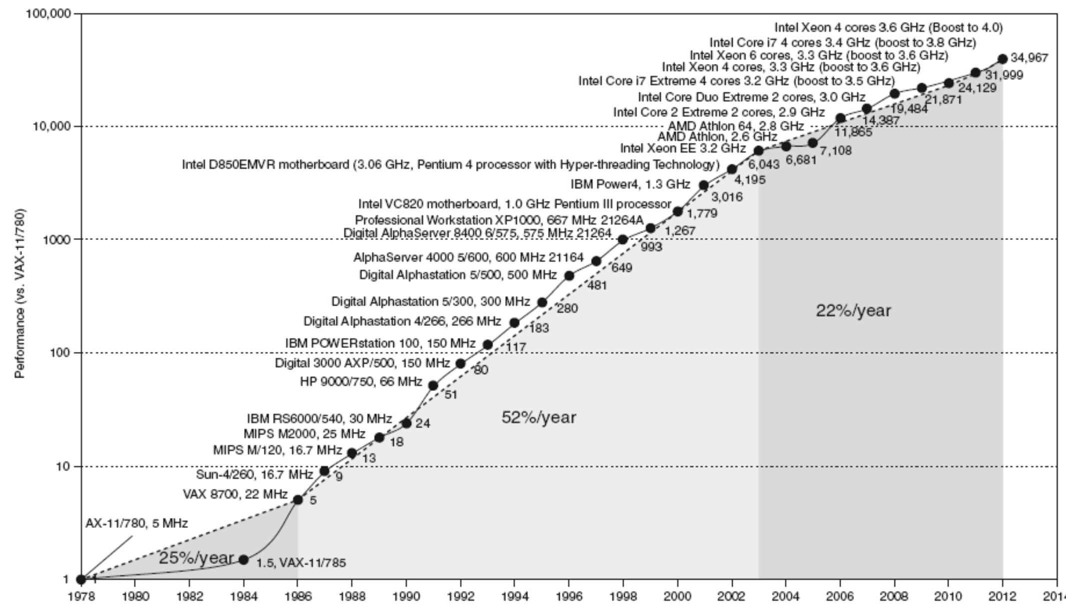
128

- We can lower the power by lowering the voltage
- The voltage has gone about as low as it can go, any further and there is too much leakage current



1.8 The Switch from Uni- to Multi-processors

- As of 2006, all desktop and server companies started shipping __multicore__ processors
- The benefit is often more of __throughput__ than __response time__
- In the past, __application__ 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?
  - The programmer divide an application so that each processor has roughly the same enough work
  - The overhead of coordinating and communicating must be small.

1.9 RealStuff: Benchmarking the Intel Core i7

- The tasks executed by a computer form a workload.
- A typical workload specifies both the programs run and their frequencies.
- Benchmarks form a workload that the user hopes will predict 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 standard sets of benchmarks 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
Σssj_ops/Σpower =		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.