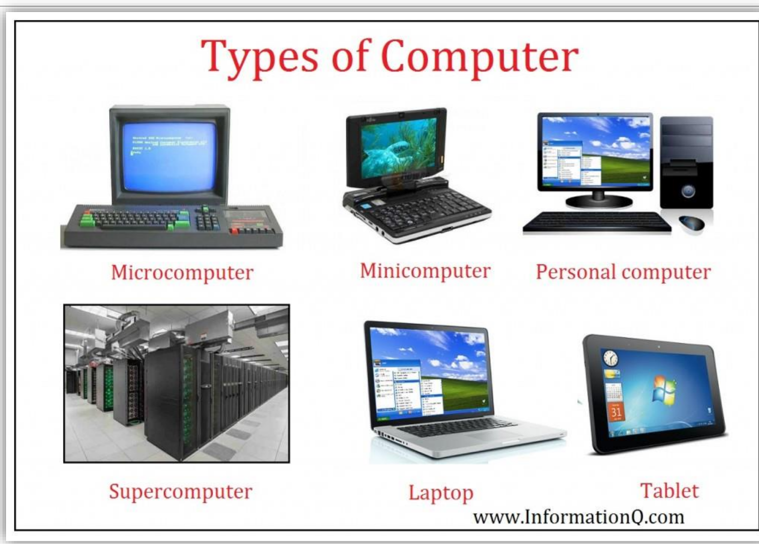
Computer Abstractions and Technology

COMPUTER ORGANIZATION AND ARCHITECTURE

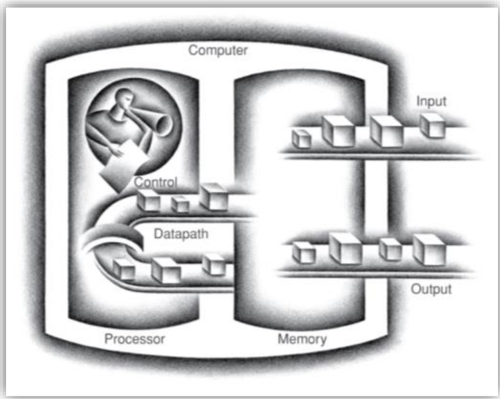
What is a Computer?



CSULB

Components of a Computer

- Classical components
  - Processor (control + data path), Memory, Input, Output



CSULB

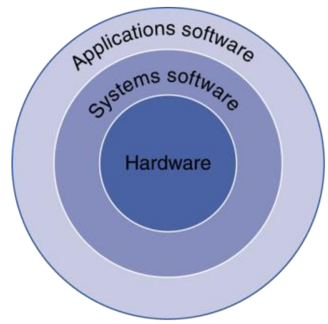
What You Will Learn

- How programs are translated into the machine language
  - And how the hardware executes them
- The hardware/software _interface_
- What determines program performance
  - And how it can be improved
- How hardware designers improve performance
- What is parallel processing

CSULB

Below Your Program

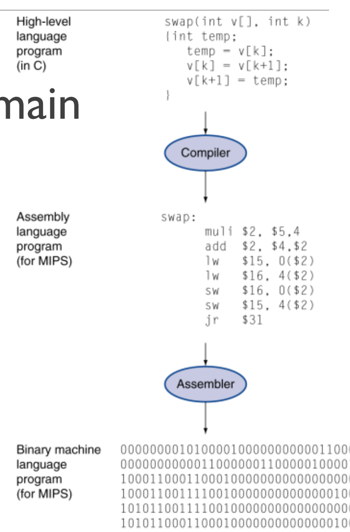
- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers



CSULB

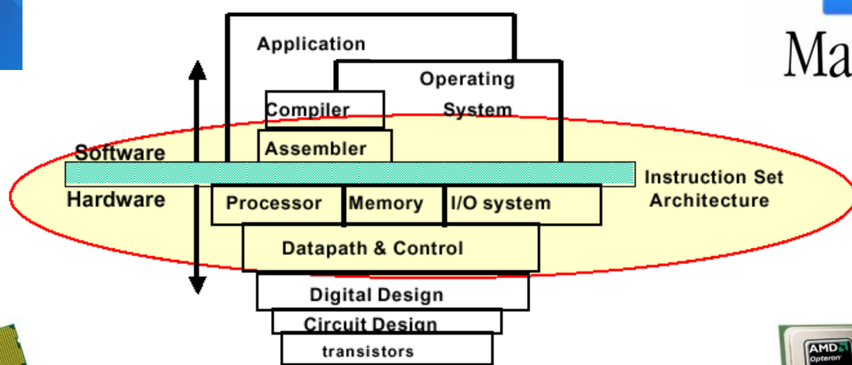
Levels of Program Code

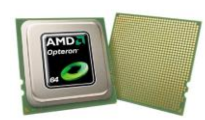
- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data



CSULB

Abstractions



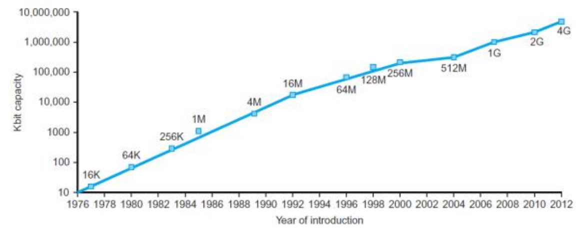


Abstractions

- Abstraction helps us deal with complexity
  - Hide lower-level detail
- The hardware/software interface
  - Instruction Set Architecture (ISA)
- Application binary interface
  - The ISA plus system software interface
- Implementation
  - The details underlying and interface

Technology Trends

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

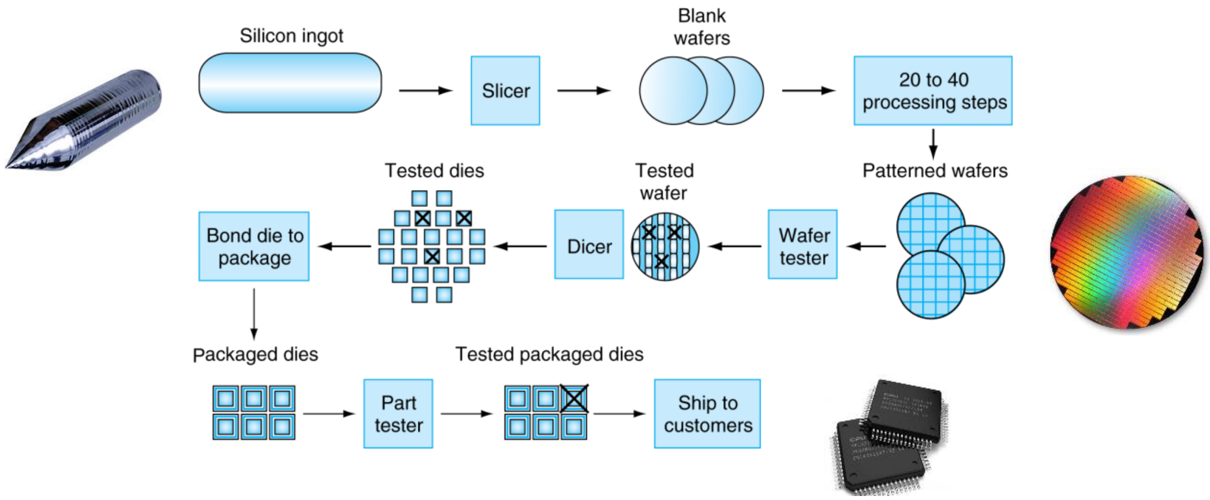


Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Semiconductor Technology

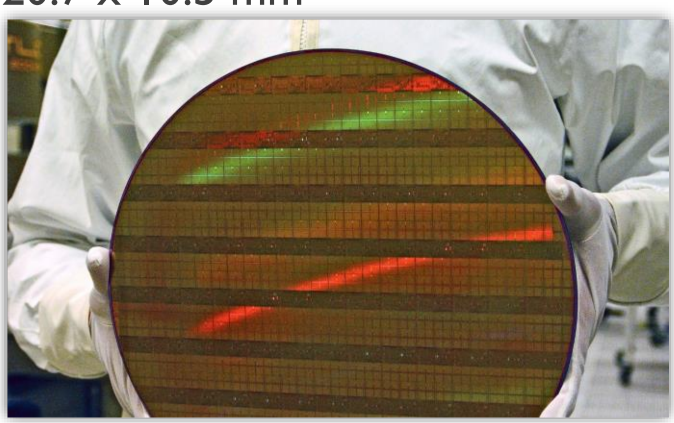
- Silicon: semiconductor
- Add materials to transform properties:
  - Conductors
  - Insulators
  - Switch

Manufacturing ICs



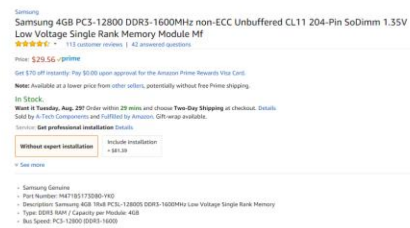
Intel Core i7 Wafer

- 300mm wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm



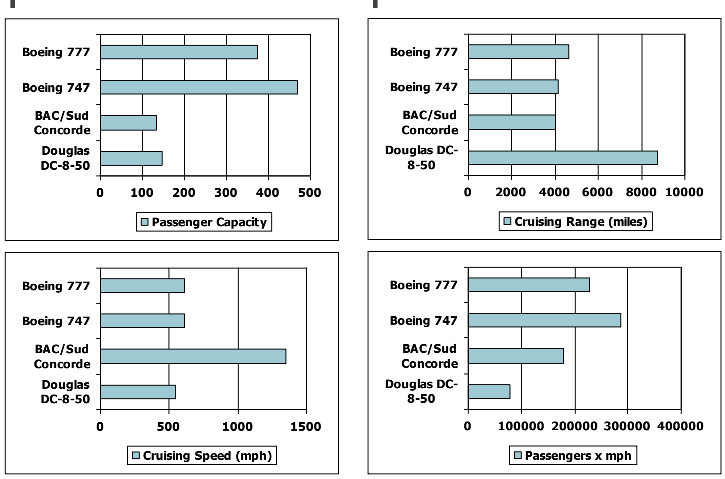
Yield Matters

- Yield: proportion of working dies per wafer



Defining Performance

- Which airplane has the best performance?



Response Time and Throughput

- Response time
  - How long it takes to do a task
- Throughput
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour
- How are response time and throughput affected by
  - Replacing the processor with a faster version?
  - Adding more processors?
- We'll focus on response time for now…

CSULB

Relative Performance

- Define Performance = $\frac{1}{Execution\ Time}$
- "X is n time faster than Y"
  - $\frac{Performance_X}{Performance_Y} = \frac{Execution\ Time_Y}{Execution\ Time_X} = n$
- Example: time taken to run a program
  - 10s on A, 15s on B
  - Execution TimeB / Execution TimeA = 15/10
  - So A is 1.5 times faster than B

CSULB

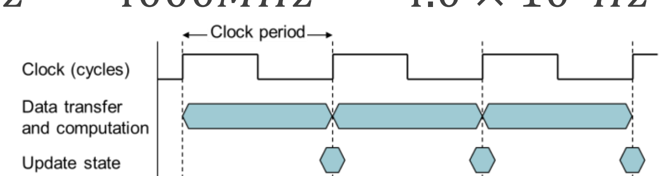
Measuring Execution Time

- Elapsed time
  - Total response time, including all aspects
    - Processing, I/O, OS overhead, idle time
  - Determines system performance
- CPU time
  - Time spent processing a given job
    - Discounts I/O time, other jobs' shares
  - Comprises user CPU time and system CPU time
  - Different programs are affected differently by CPU and system performance

CSULB

CPU Clocking

- Operation of digital hardware governed by a constant-rate clock
- Clock period: duration of a clock cycle
  - e.g., $250ps = 0.25ns = 250 \times 10^{-12}s$
- Clock frequency (rate): cycles per second
  - e.g., $4.0GHz = 4000MHz = 4.0 \times 10^{9}Hz$



CSULB

CPU Time

- $CPU\ Time = CPU\ Clock\ Cycles \times Clock\ Cycle\ Time$
$$= CPU\ Clock\ Cycles \times \frac{1}{Clock\ Rate}$$
- Performance improved by
  - Reducing number of clock cycles
  - Increasing clock rate
  - Hardware designer must often trade off clock rate against cycle count

CSULB

CPU Time Example

- Computer A: 2GHz clock, 10s CPU time
  - Designing Computer B
    - Aim for 6s CPU time
    - Can do faster clock, but causes 1.2 × clock cycles
- How fast must Computer B clock be?

$$\text{Clock rate}_B = \frac{\text{CPU Clock Cycles}}{\text{CPU Time}} = \frac{1.2 \times \text{CPU Clock Cycles}_A}{\text{CPU time}}$$

$$\text{Clock Cycles}_A = \text{CPU Time} \times \text{Clock rate}_A = 10 \times 2\,\text{GHz}$$

$$= \frac{1.2 \times 10 \times 2\,\text{GHz}}{6\,s} = 4\,\text{GHz}$$

8/29

CSULB