CS311 Computer Organization

Lecture 1:Computer Abstractions and Technology

Fall 2018

Soontae Kim

School of Computing

KAIST

The Computer Revolution

- **Progress in computer technology**
  - Underpinned by Moore's Law
    - **The number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years**
- **Makes novel applications feasible**
  - Computers in automobiles (Electronic Control Unit, ECU)
  - Smartphones
  - Human genome project
  - World Wide Web
  - Search Engines
  - Google AlphaGo, IBM Watson
- **Computers are pervasive**
  - Embedded systems, Internet Of Things (IOT)

Classes of Computers

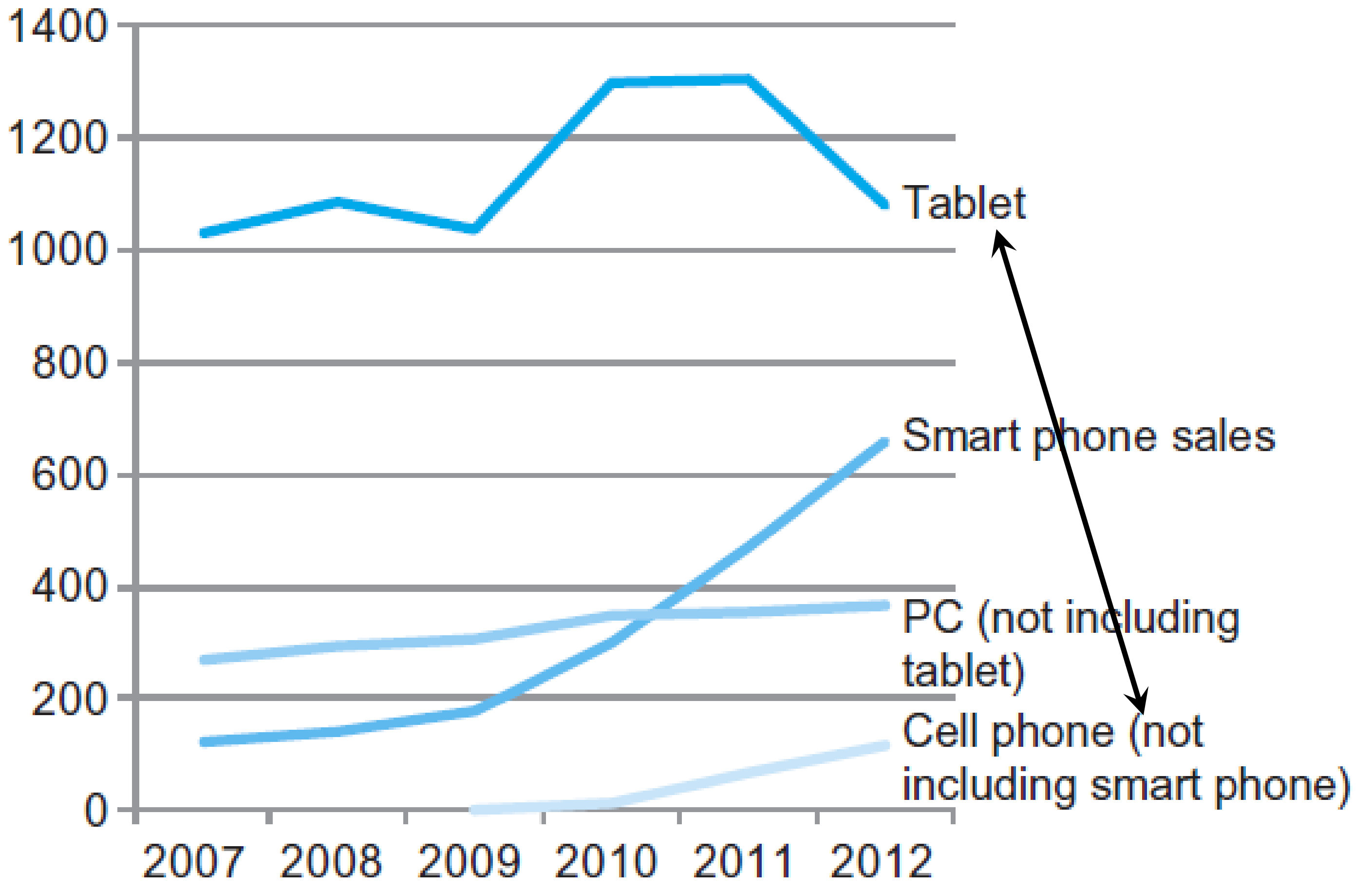
- **Personal computers**
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- **Server computers**
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
  - Developed into cloud computing

Classes of Computers II

- **Supercomputers**
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
    - **Cray supercomputers**

- **Embedded computers**
  - Hidden as components of systems
  - Stringent power/performance/cost constraints
  - E.g. Cell phone, ECUs in cars, games, TVs
  - Developed into smartphones and IOT

The PostPC Era



The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- Cloud computing
  - Warehouse Scale Computers (WSC)
    - Giant datacenters
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion ru n in the Cloud
  - Amazon and Google

What You Will Learn

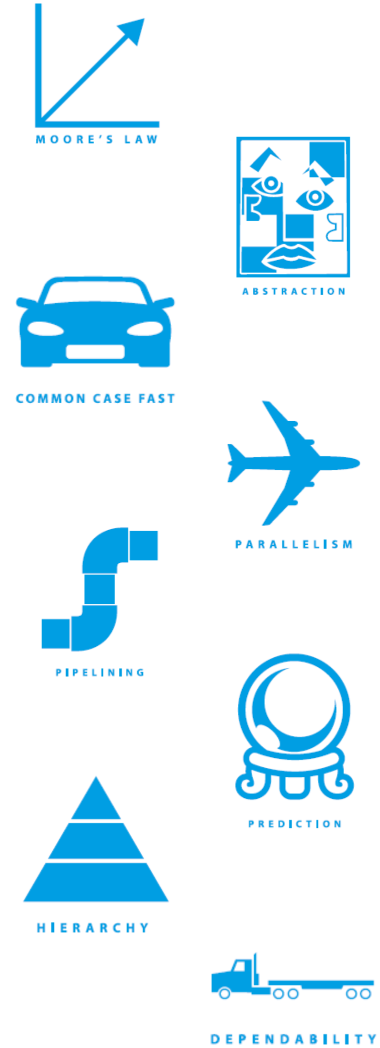
- **How programs are translated into the machine language**
  - And how the hardware executes them
- **The hardware/software interface**
  - Instruction set architecture
- **What determines program performance**
  - And how it can be improved
- **How hardware designers improve performance**
- **What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

Performance of program

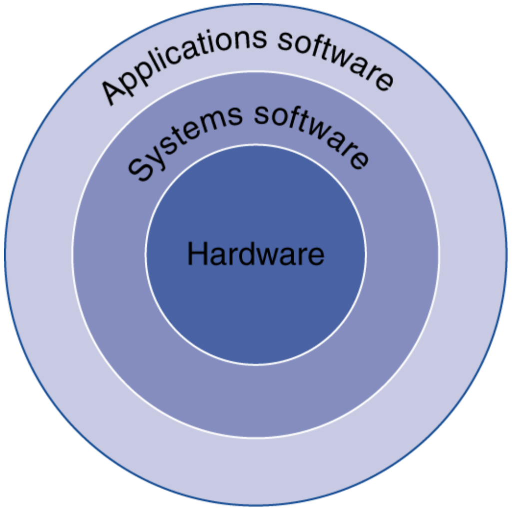
- **Algorithm**
  - Determines number of source-level statements and I/O operations executed- algorithms, data structure
- **Programming language, compiler, architecture**
  - Determine number of machine instructions executed per statement – ch2, 3
- **Processor and memory system**
  - Determine how fast instructions are executed – ch 4,5,6
- **I/O system (including OS)**
  - Determines how fast I/O operations are executed – ch 4,5,6

Eight Great Ideas invented

- **Design for *Moore's Law***
- **Use *abstraction* to simplify design**
- **Make the *common case fast***
- **Performance *via parallelism***
- **Performance *via pipelining***
- **Performance *via prediction***
- ***Hierarchy* of memories**
- ***Dependability via* redundancy**



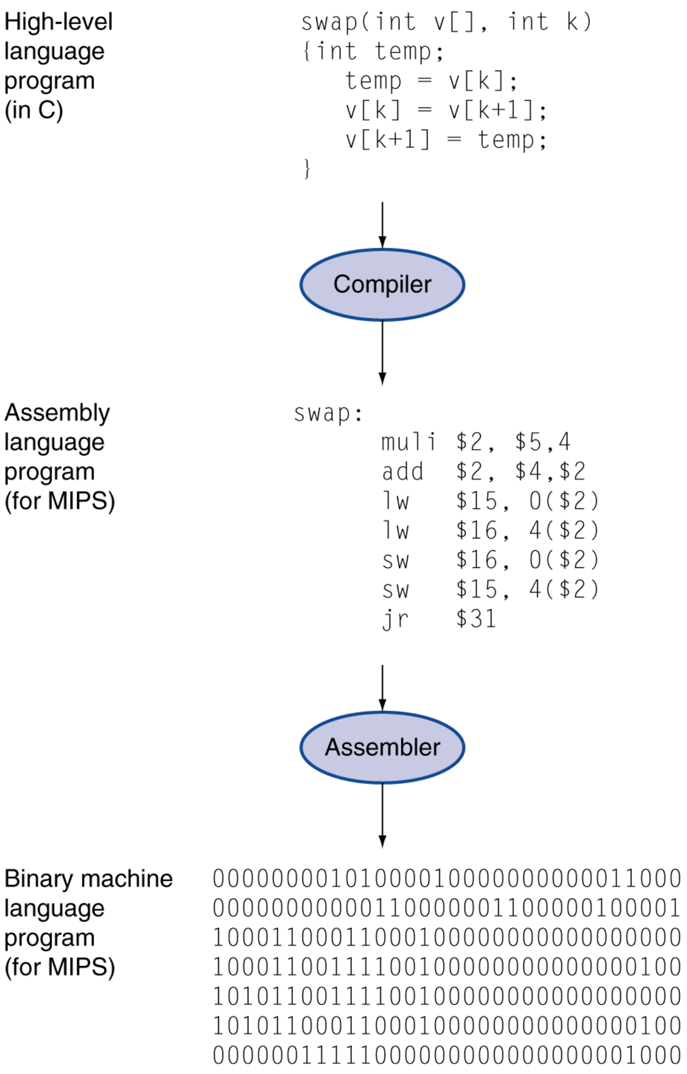
Below Your Program



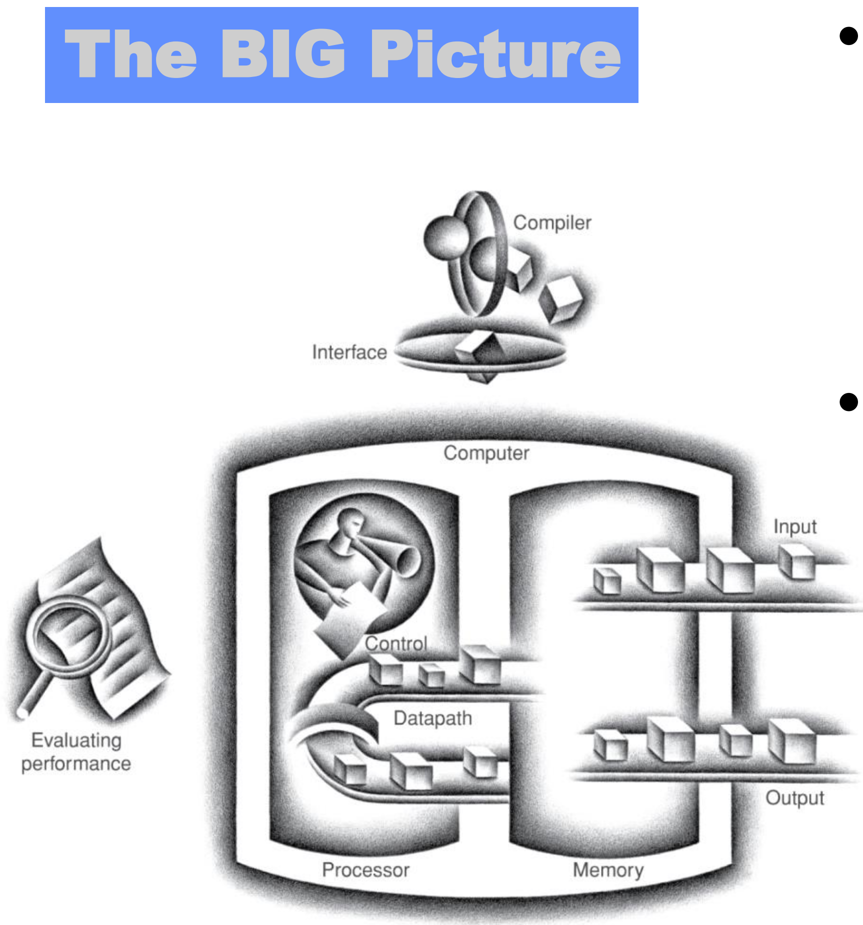
- **Application software**
  - Written in high-level language
- **System software**
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - **Handling input/output**
    - **Managing memory and storage**
    - **Scheduling tasks & sharing of resources**
- **Hardware**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

Levels of Program Code

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



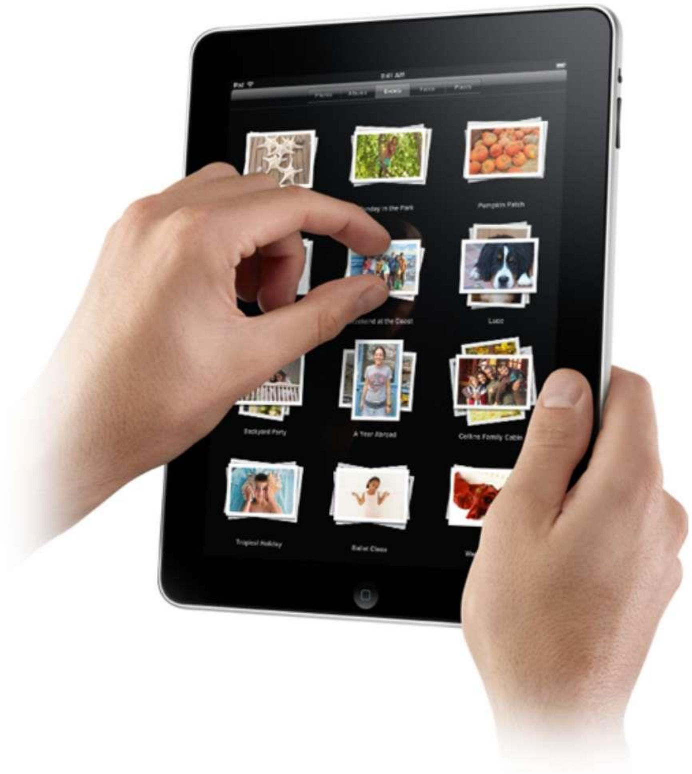
Components of a Computer



- **Same components for all kinds of computer**
  - Desktop, server, embedded
- **Input/output includes**
  - User-interface devices
    - **Display, keyboard, mouse**
  - Storage devices
    - **Hard disk, CD/DVD, flash**
  - Network adapters
    - **For communicating with other computers**

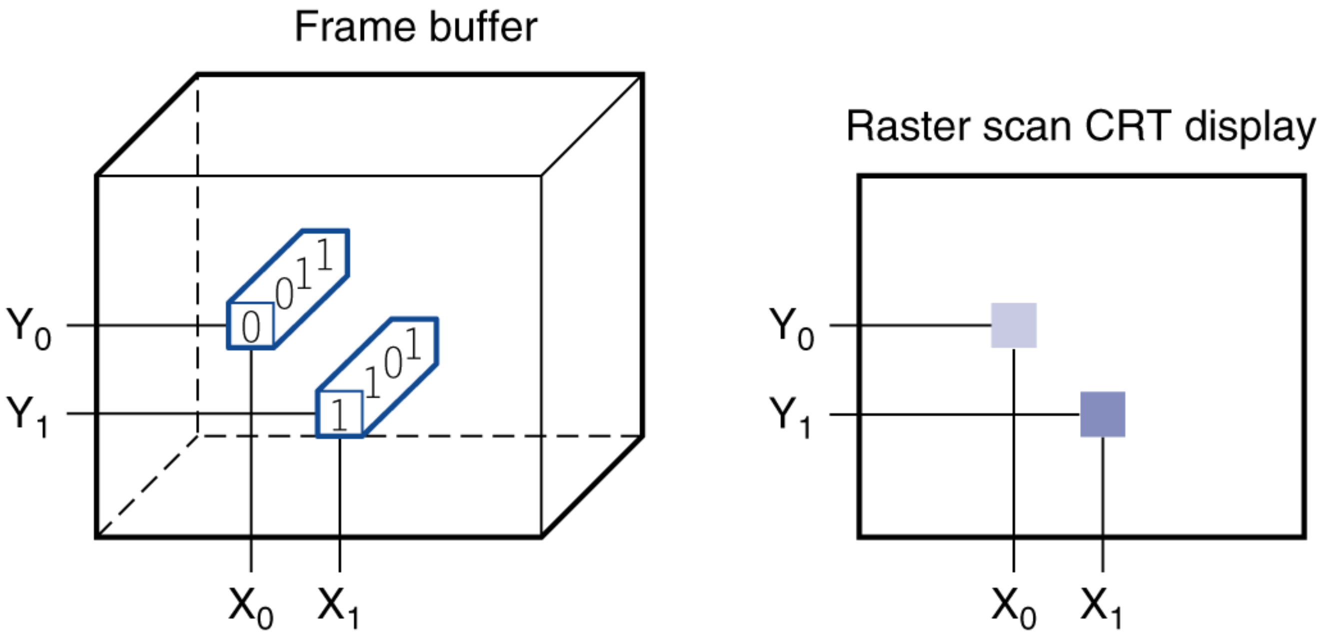
Touchscreen

- **PostPC device**
- **Supersedes keyboard and mouse**
- **Resistive and Capacitive types**
  - Most tablets, smart phones use capacitive
  - Capacitive allows multiple touches simultaneously



Through the Looking Glass

- **LCD screen: picture elements (pixels)**
  - Mirrors content of frame buffer memory

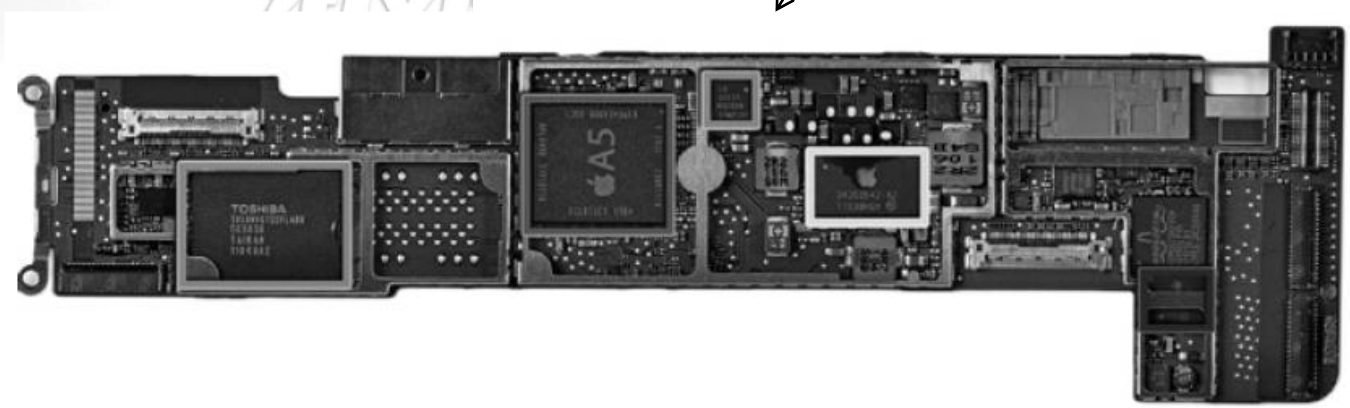


Opening the Box

Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

Computer board



Components of the Apple iPad 2 A1395

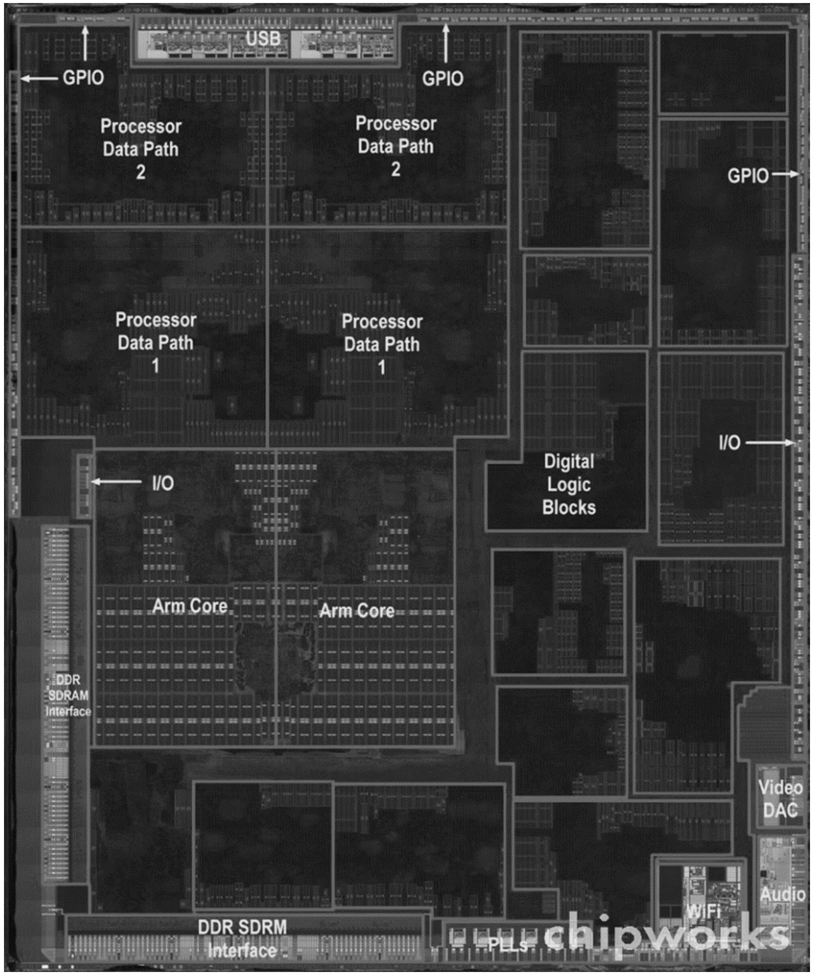
Inside the Processor (CPU)

- **Datapath: performs operations on data**
- **Control: sequences datapath, memory, ...**
- **Cache memory**
  - Small fast SRAM memory for immediate access to data

Inside the Processor

- **Apple A5**

GPU



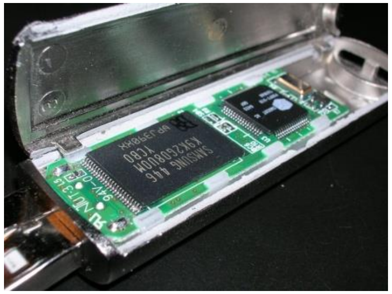
Abstractions

The BIG Picture

- **Abstraction helps us deal with complexity**
  - Hide lower-level detail
- **Instruction set architecture (ISA)**
  - The hardware/software interface including instructions, registers, memory access, I/O …
- **Application binary interface (ABI)**
  - The ISA + OS interface provided to app. programmers
- **Implementation**
  - Hardware that obeys the architecture abstraction

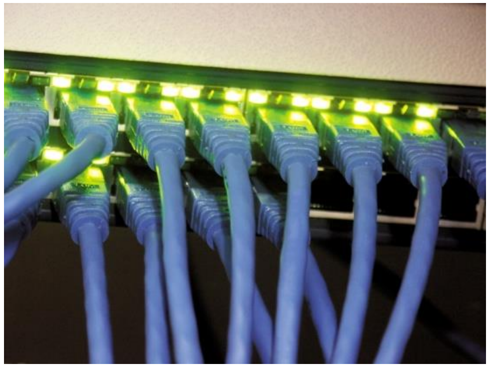
A Safe Place for Data

- **Volatile main memory**
  - Loses instructions and data when power off
  - Mainly use DRAM (Dynamic RAM)
- **Non-volatile secondary memory**
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)



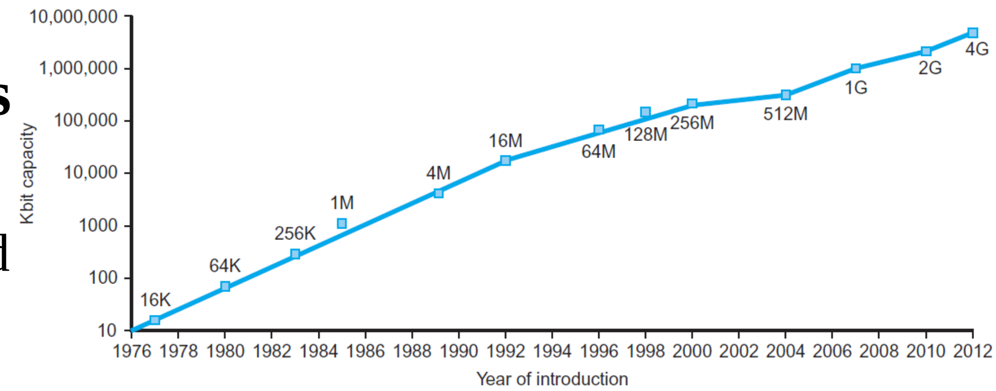
Networks

- **Networked computers have advantages of communication, resource sharing, nonlocal access**
- **Local area network (LAN): Ethernet**
- **Wide area network (WAN): the Internet**
- **Wireless network: WiFi, Bluetooth**



Technology Trends

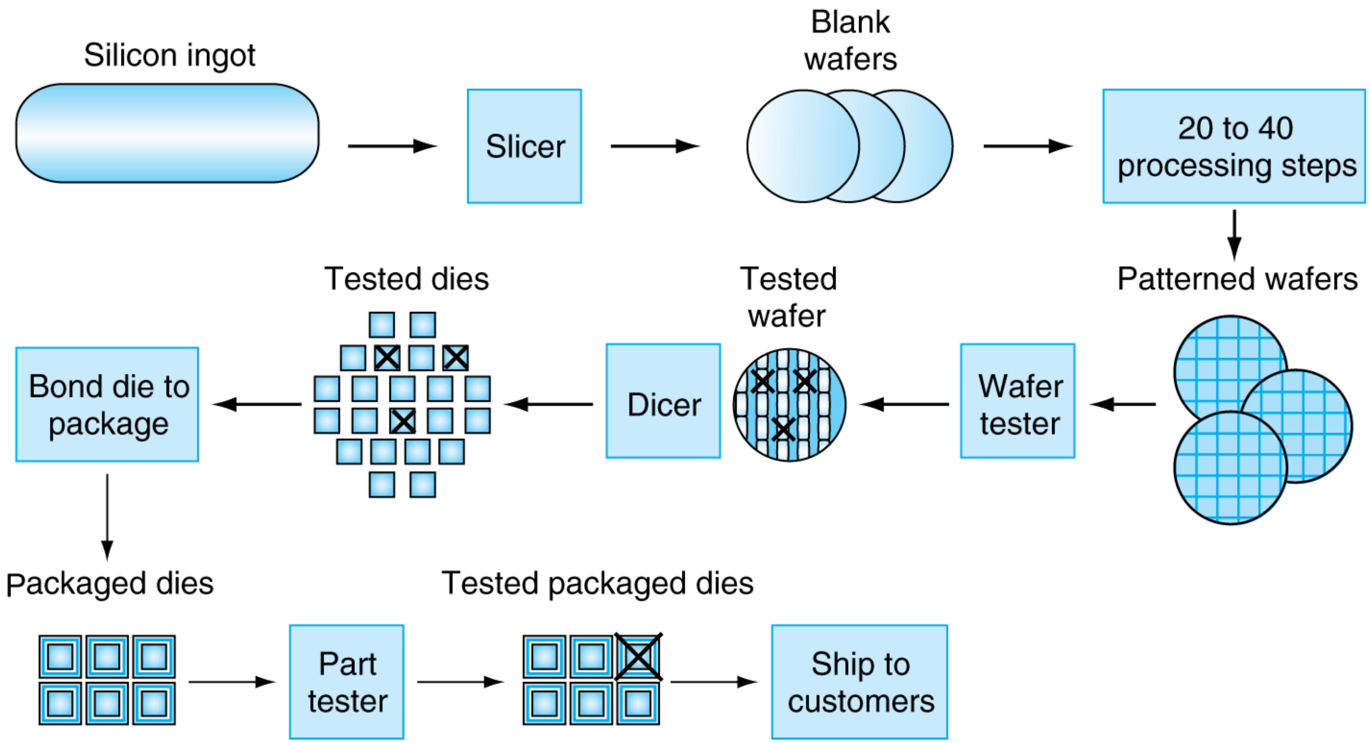
- **Electronics technology continues to evolve**
  - Increased capacity and performance
  - Reduced cost



DRAM capacity

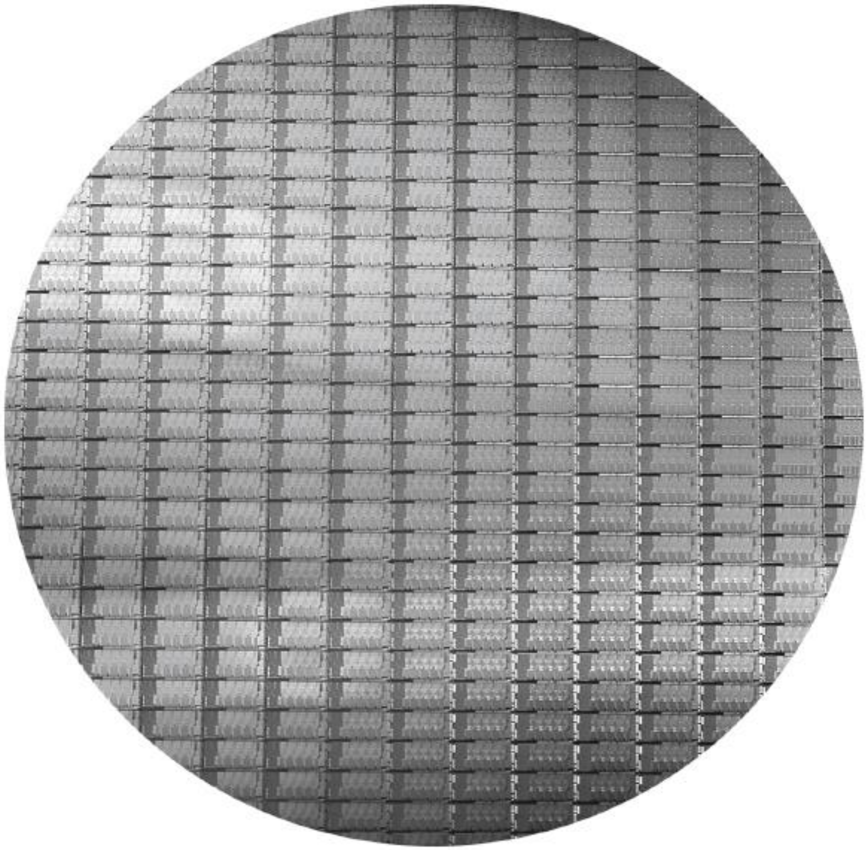
Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Manufacturing ICs



- **Yield: proportion of working dies per wafer**

Intel Core i7 Wafer



- **300mm wafer, 280 dies, 32nm technology**
- **Each die is 20.7 x 10.5 mm**

Integrated Circuit Cost

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area} / \text{Die area}$$

$$\text{Yield} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area}/2))^2}$$

- **Nonlinear relation to area and defect rate**
  - Wafer cost and area are fixed
  - Defect rate determined by manufacturing process
  - Die area determined by architecture and circuit design