EECS2021E

Computer Organization

Amir Ashouri
York University

Fall 2019

These slides are based on the slides by the authors.
The slides doesn't include all the material covered in the lecture.
The slides will be explained, modified, and sometime corrected in the lecture.

Course Staff

Instructor:

- Amir Ashouri (aashouri@eecs.yorku.ca)
- Office Hours: **Wednesdays (will be announced)**
- https://wiki.eecs.yorku.ca/course_archive/2019-20/F/2021E

- **TAs**: **(will be announced)**
  - Main point of contact for your course labs
- **Lab Hours**:
  LAB 01 (Mondays) 19:00-22:00 (YK LAS 1006)
  LAB 02 (Tuesdays) 19:00-22:00 (YK LAS 1006)

Course Textbook

Required Textbook:

“Computer Organization and Design RISC-V Edition: The Hardware Software Interface”

(The Morgan Kaufmann Series in Computer Architecture and Design)

David A. Patterson & John L. Hennessy

1st edition.

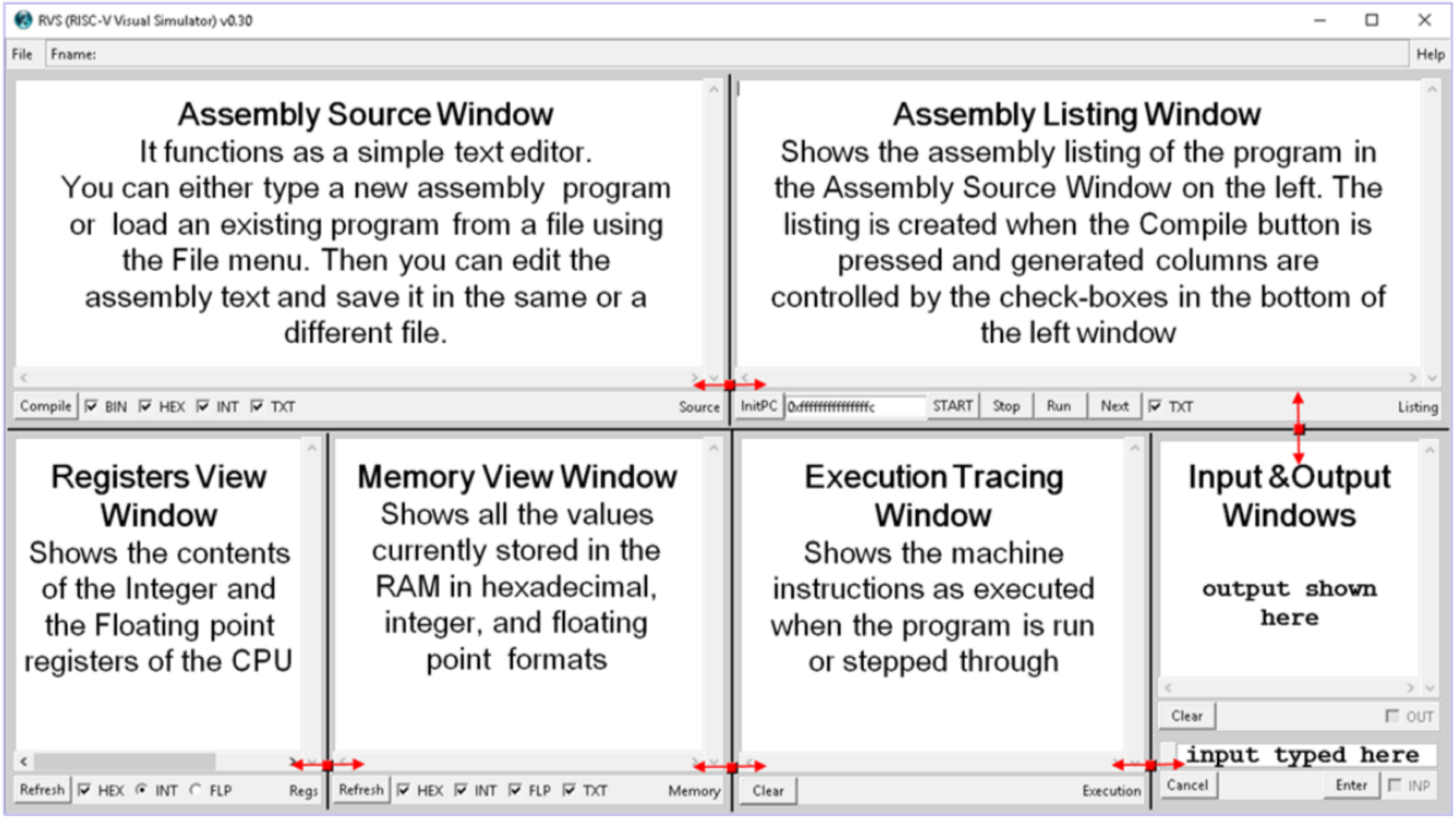
Tentative Schedule

	Date	Lecture Content	Labs	
1	Sep 9 to 11	Chapter 1, Chapter 2 (2.1 - 2.4)		
2	Sep 9 to 11	Chapter 2 (2.5 - 2.7)		
	Sep 16 to 20	Chapter 2 (2.8)	Lab 1	
3	Sep 23 to 27	Chapter 2 (2.9 - 2.11)	Lab 2	
4	Sep 30 to Oct 4	Chapter 2	Lab 3	
5	Oct 7 to Oct 11	Chapter 3	Lab 4	
6	Oct 14 to Oct 18	Fall Reading Week - NO CLASSES		
7	Oct 21 to Oct 25	Chapter 3		**MidTerm**
8	Oct 28 to Nov 1	Chapter 3	Lab 5	
9	Nov 4 to Nov 1	Chapter 3	Lab 6	
10	Oct 28 to Nov 1	Chapter 3	Lab 7	
11	Oct 28 to Nov 1	Chapter 4	Lab 8	
12	Apr 1 to Apr 5	Chapter 4		
13	Apr 8 to Apr 12	Chapter 4		

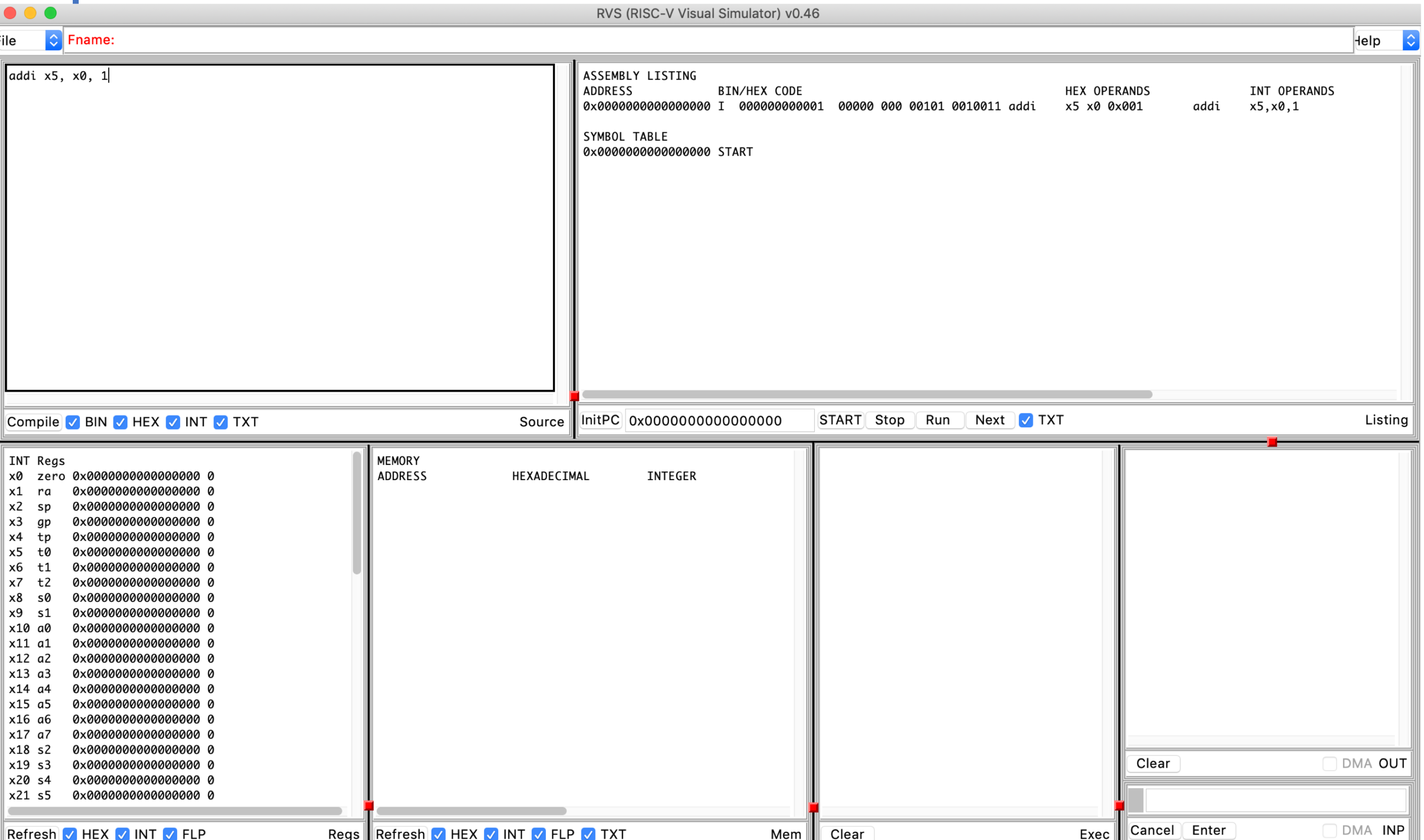
Prerequisites

- General Prerequisite
- Basic Understanding of Programing
- Labs (@ LAS 1006)
  - We will use York's inhouse RISK-V simulator for our lab assignments

RISK-V Simulator (1/2)



RISK-V Simulator (2/2)



Grade Composition

Lab	30%
Midterm	30%
Final	40%

EECS2021E Course Description

- Features RISC-V, the first such architecture designed to be used in modern computing environments, such as cloud computing, mobile devices, and other embedded systems
- Includes relevant examples, exercises, and material highlighting the emergence of mobile computing and the cloud

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

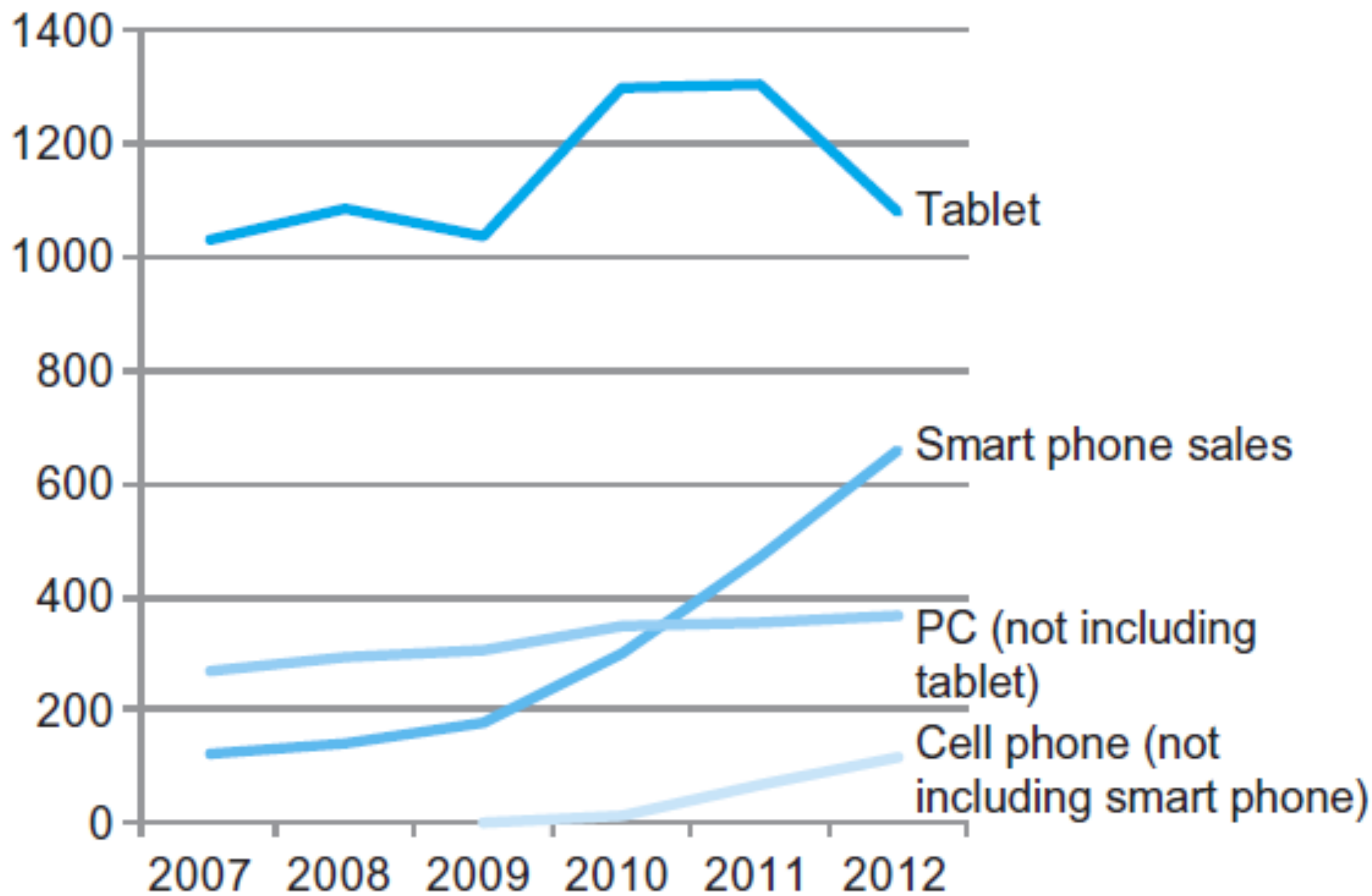
The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore's Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasive

Classes of Computers

- Supercomputers
  - High-end scientific and engineering calculations
  - Highest capability but represent a small fraction of the overall computer market
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints

The PostPC Era



The PostPC Era

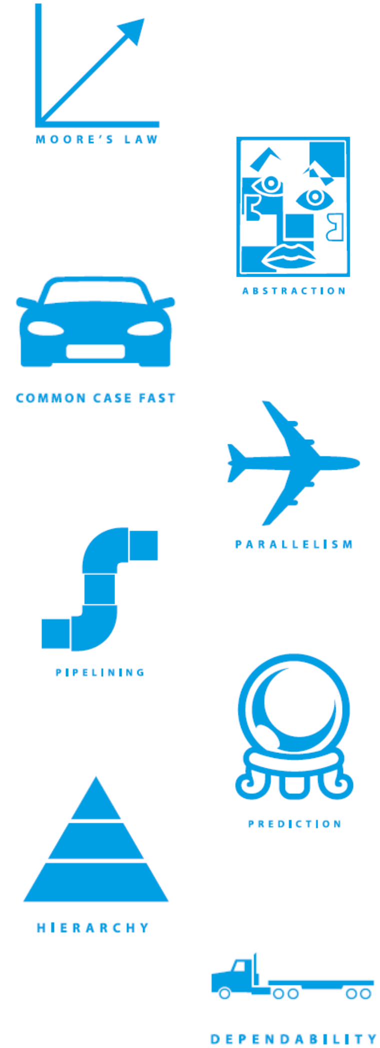
- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- Cloud computing
  - Warehouse Scale Computers (WSC)
  - Software as a Service (SaaS)
  - Portion of software run on a PMD and a portion run in the Cloud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

Understanding Performance

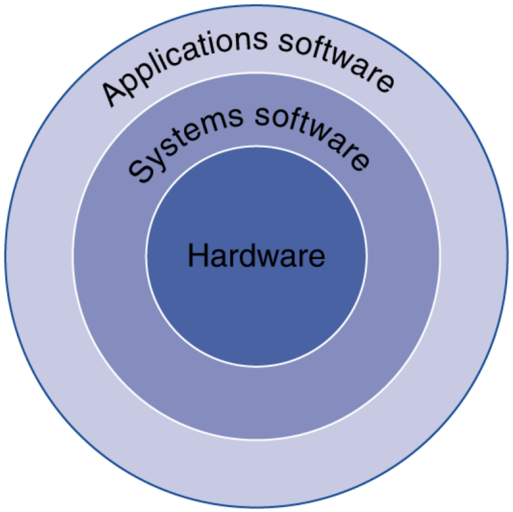
- Algorithm
  - Determines number of operations executed
- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation
- Processor and memory system
  - Determine how fast instructions are executed
- I/O system (including OS)
  - Determines how fast I/O operations are executed

Eight Great Ideas

- 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 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
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded instructions and data

High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Compiler

Assembly language program (for RISC-V)

```
swap:
      slli x6, x11, 3
      add  x6, x10, x6
      ld   x5, 0(x6)
      ld   x7, 8(x6)
      sd   x7, 0(x6)
      sd   x5, 8(x6)
      jalr x0, 0(x1)
```

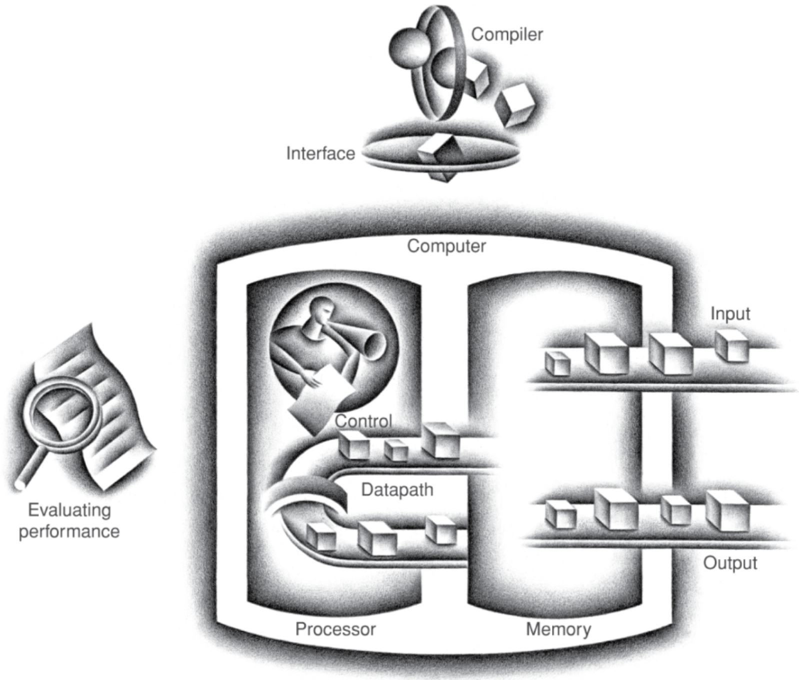
Assembler

Binary machine language program (for RISC-V)

```
00000000001101011001001100010011
00000000011001010000001100110011
00000000000000110011001010000011
00000000100000110011001110000011
00000000011100110011000000100011
00000000010100110011010000100011
00000000000000001000000001100111
```

Components of a Computer

The BIG Picture



- 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

Inside the Processor (CPU)

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

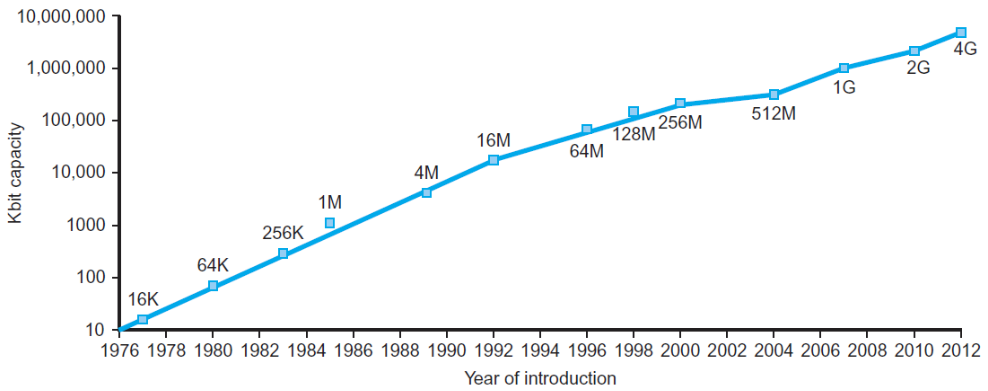
Abstractions

The BIG Picture

- Abstraction helps us deal with complexity
  - Hide lower-level detail
- Instruction set architecture (ISA)
  - The hardware/software interface
- 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



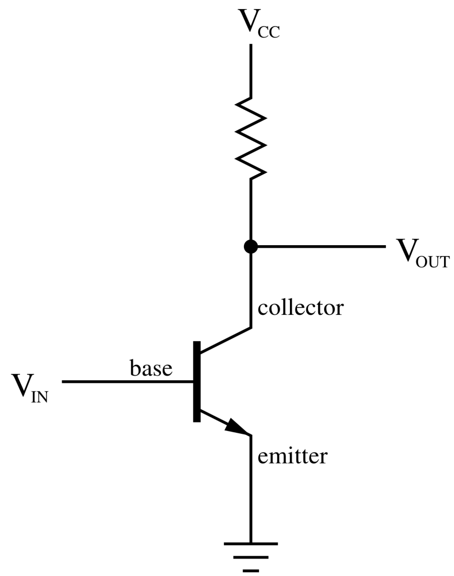
DRAM capacity

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

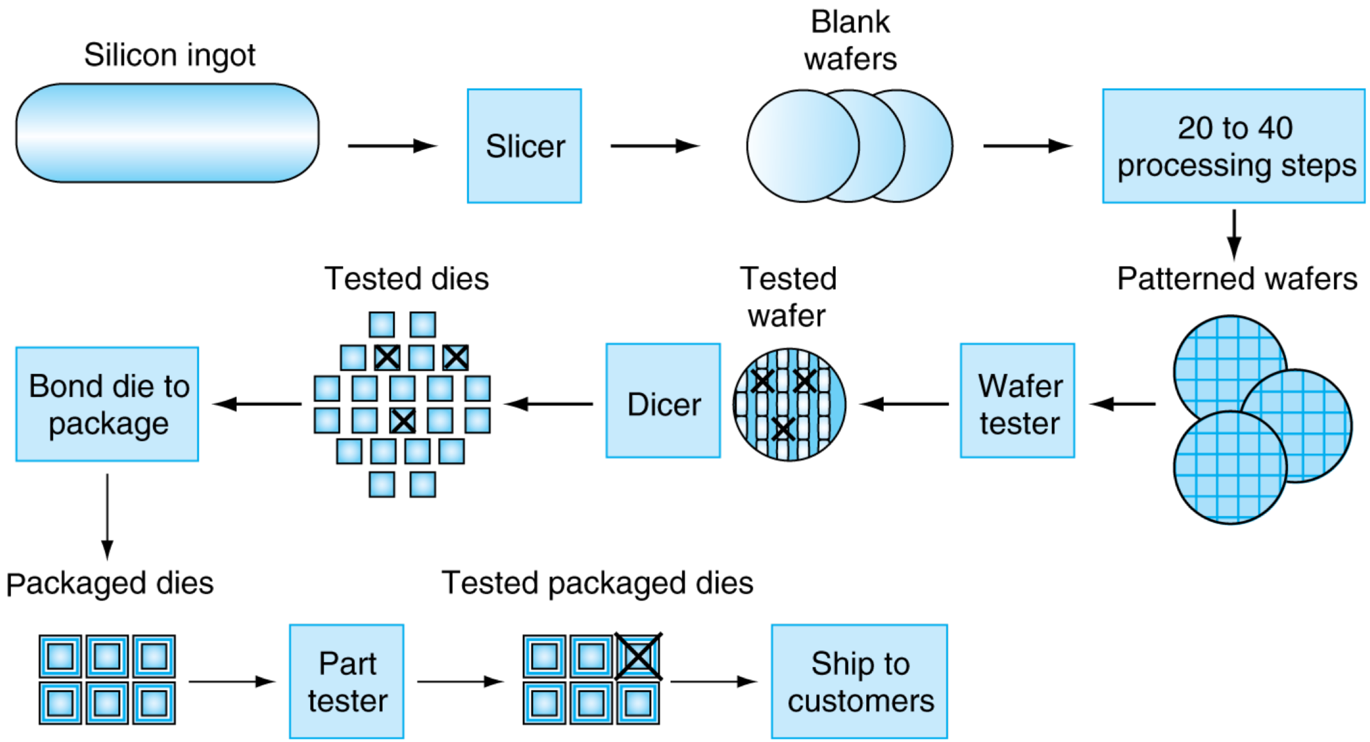
§1.5 Technologies for Building Processors and Memory

Semiconductor Technology

- Silicon atoms: semiconductor
- Add materials to transform properties:
  - Conductors N or P.
  - Switch Combine them to make switches.

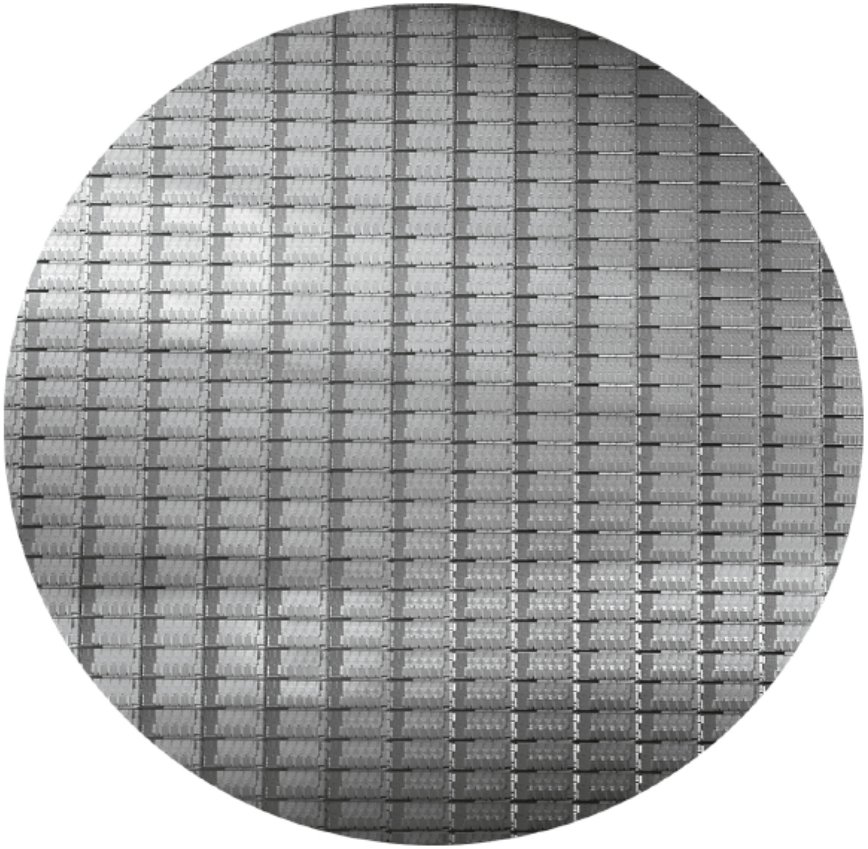


Manufacturing ICs



- Yield: proportion of working dies per wafer

Intel Core i7 Wafer



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

From Sand To Silicon

https://www.youtube.com/watch?v=Q5paWn7bFg4