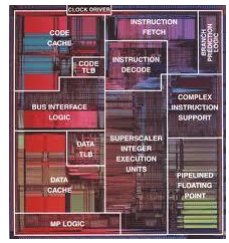
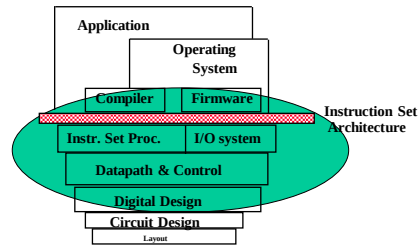
UTD

# CS/SE 3340
# Computer Architecture



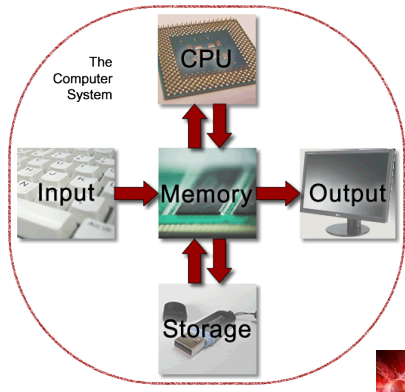



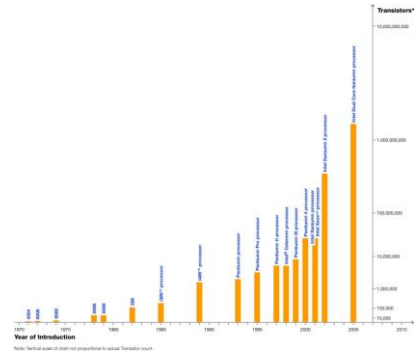

## Introduction to Computer Organization

*Adapted from "Computer Organization and Design, 4th Ed." by D. Patterson and J. Hennessy*

# Questions

- What are the five main components of a computer system?
- What is the Moore's law?
- What is abstraction and how it is used for computer systems?
- What is the interface between hardware and software?
- What is a system/application (user) program?

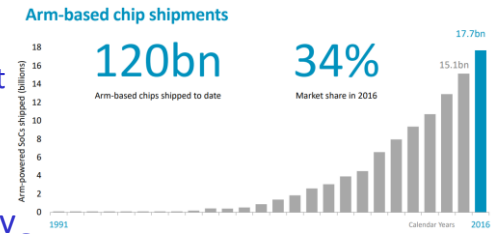



3

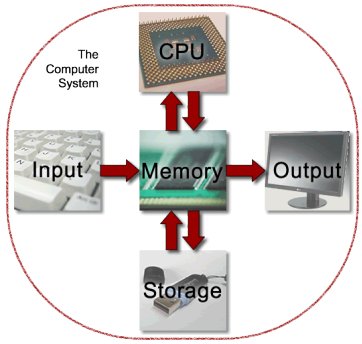
# The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore's Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones, tablets
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasiv
  - **17.7B** ARM chips sold in 2016 (ARM Holdings)


4

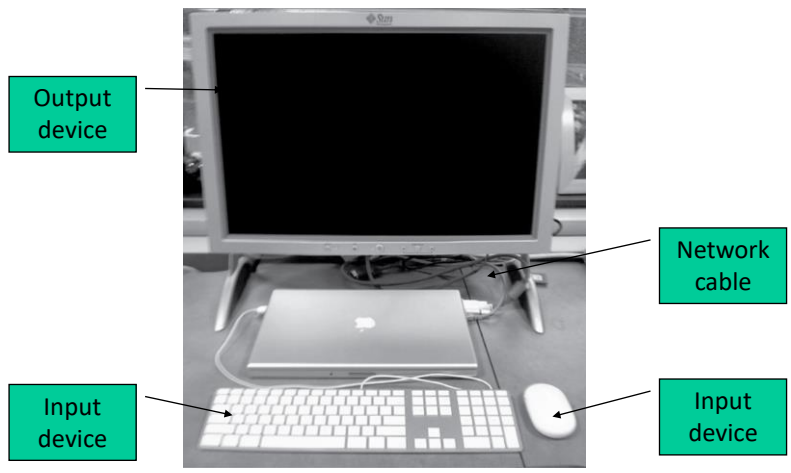
# Hardware Components




- *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
  - Something missing?
    - What about network adapters?

5

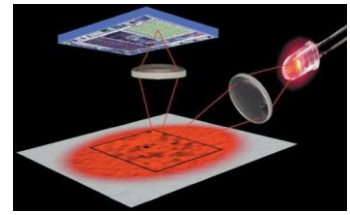
# Anatomy of a Computer




6

# Anatomy of a Mouse

- Optical mouse
  - LED illuminates desktop
  - Small low-res camera
  - Basic image processor
    - Looks for x, y movement
  - Buttons & wheel
- Supersedes roller-ball mechanical mouse

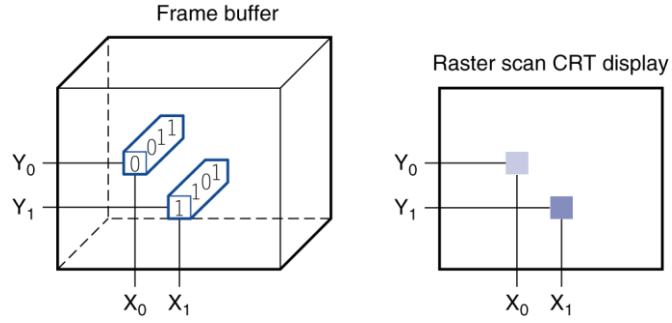


7

# Through the Looking Glass

LCD screen: picture elements (pixels)

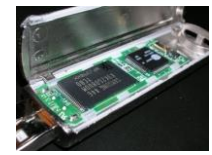
Mirrors content of frame buffer memory




8

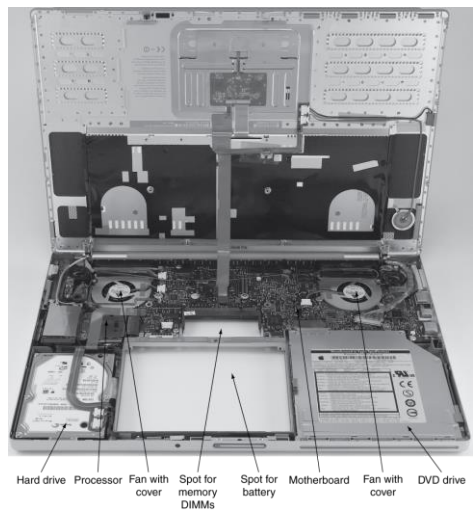
# A Safe Place for Data

- Volatile main memory
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)

9

# Opening the Box




10

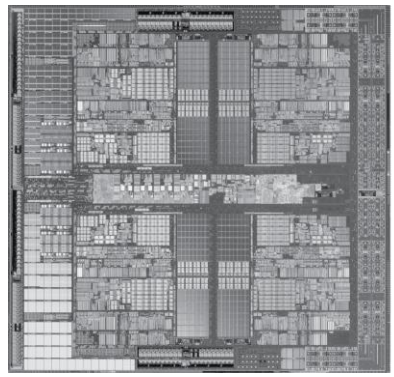
# Inside the Processor (CPU)

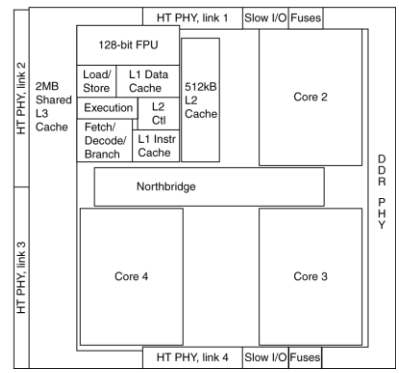
- *Datapath:* performs operations on data
- *Control:* sequences datapath, memory, ...
- Cache memory
  - Small fast SRAM memory for immediate access to data



# Inside the Processor

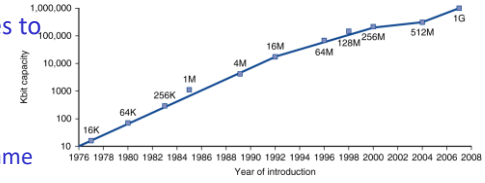
## AMD Barcelona: 4 processor cores





# Technology Trends

- Electronics technology continues to evolve
  - Increased capacity and performance, reduced cost
  - If car prices had fallen at the same rate as the price of a single transistor has since 1968, a new car today would cost about 0.5 cent!!!




DRAM capacity

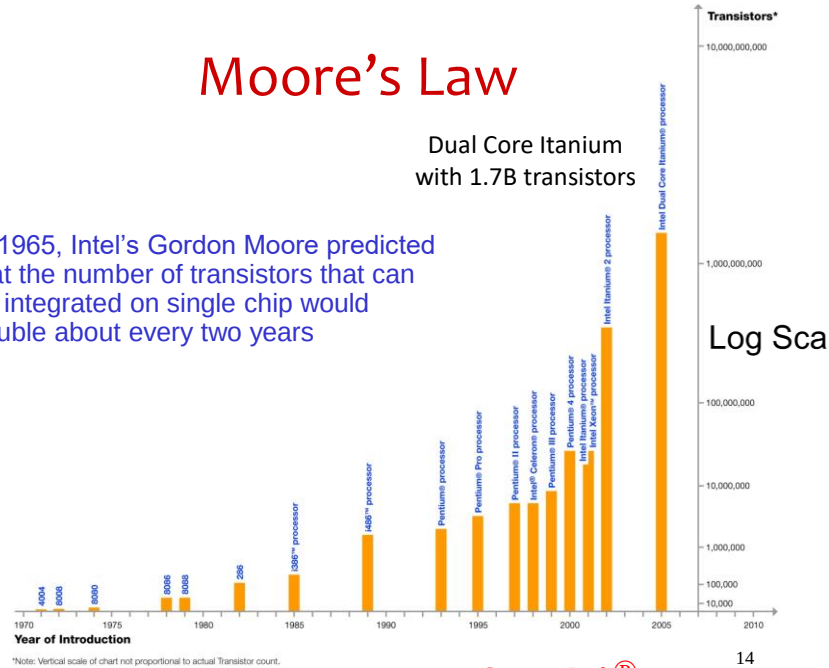
| Year | Technology | Relative performance/cost |
|---|---|---|
| 1951 | Vacuum tube | 1 |
| 1965 | Transistor | 35 |
| 1975 | Integrated circuit (IC) | 900 |
| 1995 | Very large scale IC (VLSI) | 2,400,000 |
| 2005 | Ultra large scale IC | 6,200,000,000 |



# Moore's Law

Dual Core Itanium
with 1.7B transistors

- In 1965, Intel's Gordon Moore predicted that the number of transistors that can be integrated on single chip would double about every two years




Courtesy, Intel ®

# Technology Scaling Road Map

| Year | 2004 | 2006 | 2008 | 2010 | 2012 |
|---|---|---|---|---|---|
| Feature size (nm) | 90 | 65 | 45 | 32 | 22 |
| Capacity (Bills. of Transistor) | 2 | 4 | 6 | 16 | 32 |

- Fun facts about 22 nm transistors
  - 60 million can fit on the head of a pin
  - You could fit more than 4,000 across the width of a human hair
- 7 nm transistors have been used in VLSI chips since 2018 ( Apple's A12 Bionic)

15

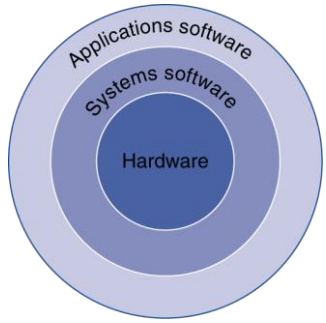
*International Technology Roadmap for Semiconductors*

# Some High-level Questions

- How are *data* and *programs* represented in a computer?
  - And how the **hardware** processes data?
- What is the interface between **hardware** an **software**?
  - *The Instruction Set Architecture (ISA)*
- ***What*** determines computer ***performance?*** and ***how*** it can be improved?
- How hardware designers improve ***performance***?
  - Faster clock, pipelining, cache, etc...

16

# Below Your Program

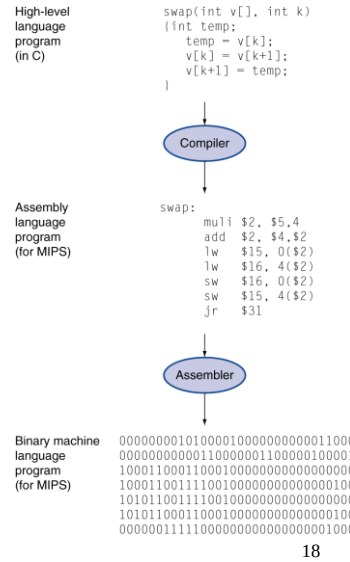



- *Application software*
  - Written in high-level language
- *System software*
  - Compiler: translates HLL code to assembly language or machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- *Hardware*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

17

# 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)
  - Instructions (machine code) and data are encoded in *binary*




18

## 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

19

## Abstractions

- *Abstraction* helps us deal with complexity
  - Hide lower-level details
- ***Instruction set architecture (ISA)***
  - The hardware/software interface
- ***Application binary interface***
  - The ISA plus system software interface
- ***Implementation***
  - The details underlying an interface

20

# Review: Some Basic Definitions

- Kilobyte – $2^{10}$ or 1,024 bytes
- Megabyte– $2^{20}$ or 1,048,576 bytes
  - sometimes "rounded" to $10^6$ or 1,000,000 bytes
- Gigabyte – $2^{30}$ or 1,073,741,824 bytes
  - sometimes rounded to $10^9$ or 1,000,000,000 bytes
- Terabyte – $2^{40}$ or 1,099,511,627,776 bytes
  - sometimes rounded to $10^{12}$ or 1,000,000,000,000 bytes
- Petabyte – $2^{50}$ or 1024 terabytes
  - sometimes rounded to $10^{15}$ or 1,000,000,000,000,000 bytes
- Exabyte – $2^{60}$ or 1024 petabytes
  - Sometimes rounded to $10^{18}$ or 1,000,000,000,000,000,000 bytes

"All mobile data traffic generated worldwide > 107 exabytes in 2017"

*ABI Research – July 2012*