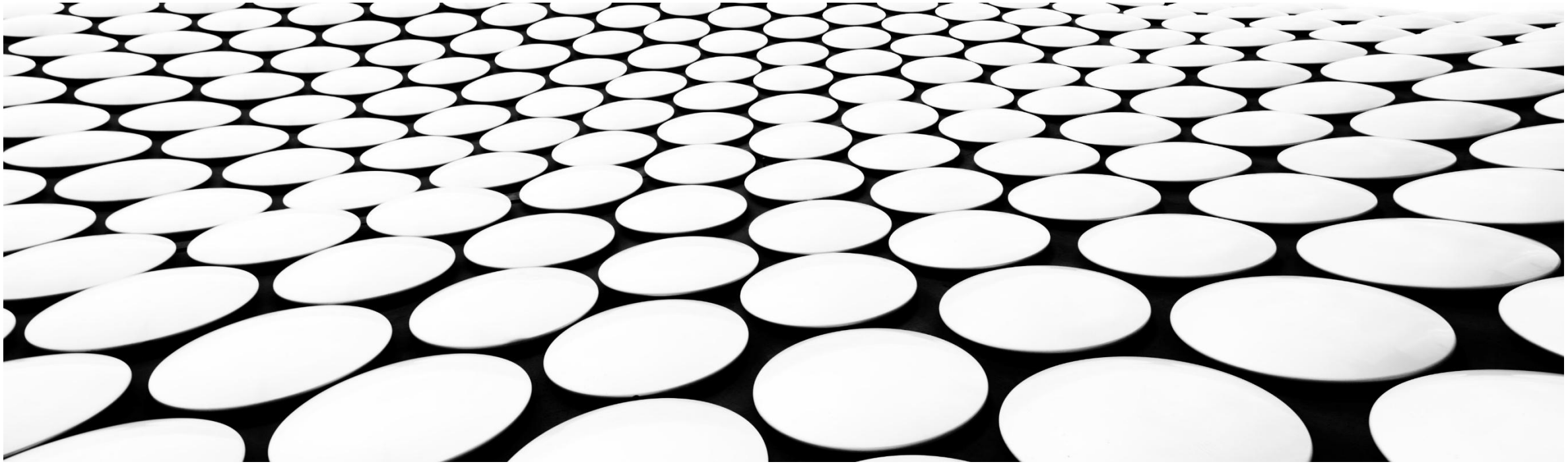
CS5102: FOUNDATIONS OF COMPUTER SYSTEMS

TOPIC-1: INTRODUCTION

DR. ARIJIT ROY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY PATNA



These slides are based on the book: David A. Patterson, John L. Hennessy, Computer Organization and Design and Operating Systems Concepts by Galvin

DETAILED SYLLABUS

Computer architecture:

- Performance measures
- Memory and Addressing Modes
- Instruction Set, A Simple Machine,
- Instruction Mnemonics and Syntax,
- Machine Language Program,
- Assembly Language Program with examples
- Processing Unit Design
- Control Unit Design
- Pipelining and parallel processing

Operating systems:

- Process and Threads,
- Process scheduling
- Concurrent processes
- Process
- Semaphore
- Memory Management
- Deadlock
- Resource Allocation Graph
- Deadlock prevention and Avoidance

WHAT'S INSIDE THE BOX?

WHY LEARN THIS COURSE

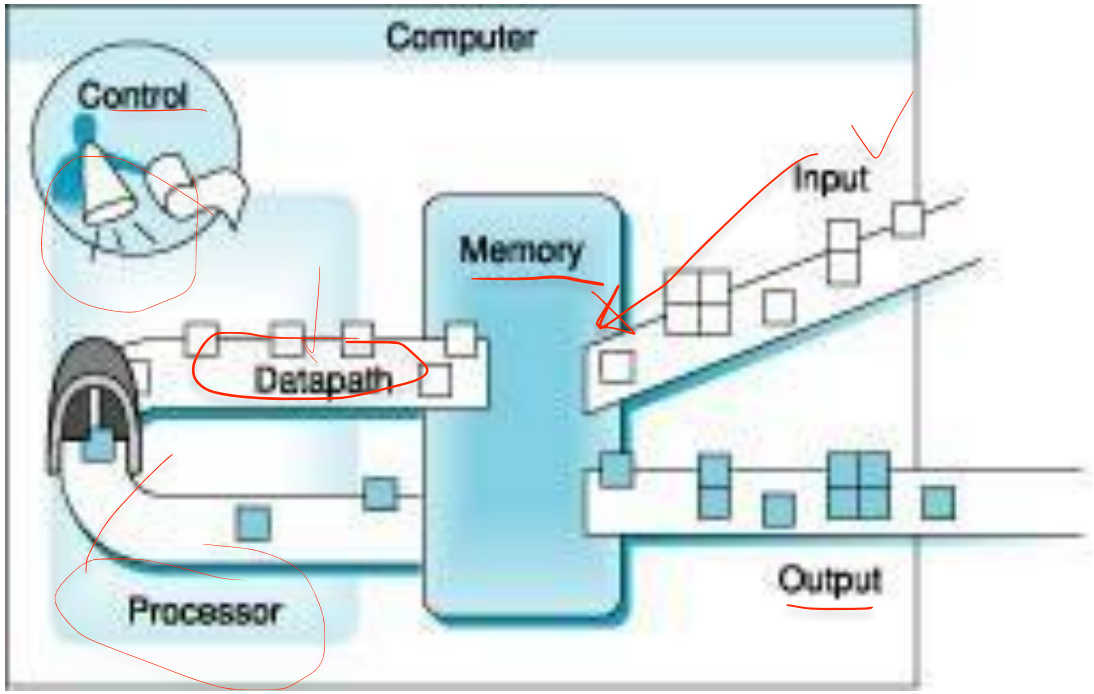
- You want to call yourself a "**computer engineer**"
- You want to build software people use (need **performance**)
- You need to make a purchasing decision or offer "**expert**" advice

- Both Hardware and Software affect performance:
  - Algorithm determines number of **source-level statements**
  - Language/Compiler/Architecture determine **machine instructions**
  - Processor/Memory determine how **fast** instructions are executed
  - I/O and Number_of_Cores determine **overall system performance**

ORGANIZATION OF A COMPUTER

- Five classic components of a computer – input, output, memory, datapath, and control

datapath + control = processor (CPU)

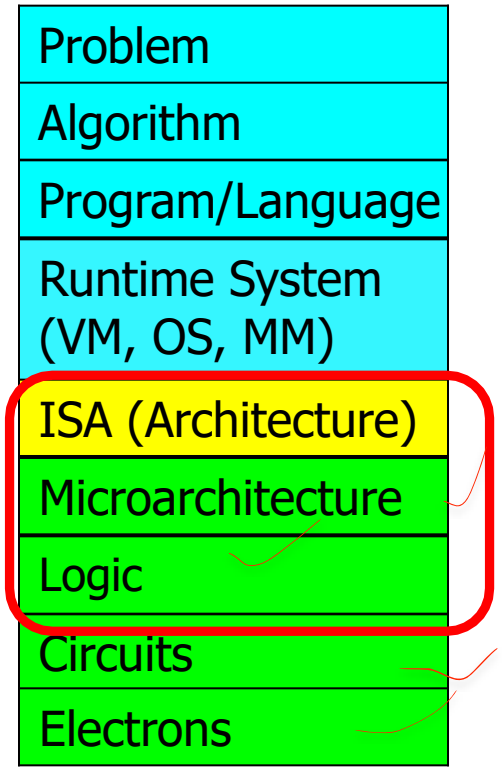


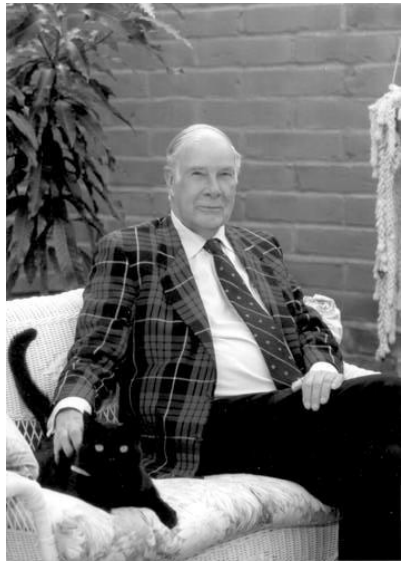
LEVELS OF TRANSFORMATION

"The purpose of computing is insight" (*Richard Hamming*)

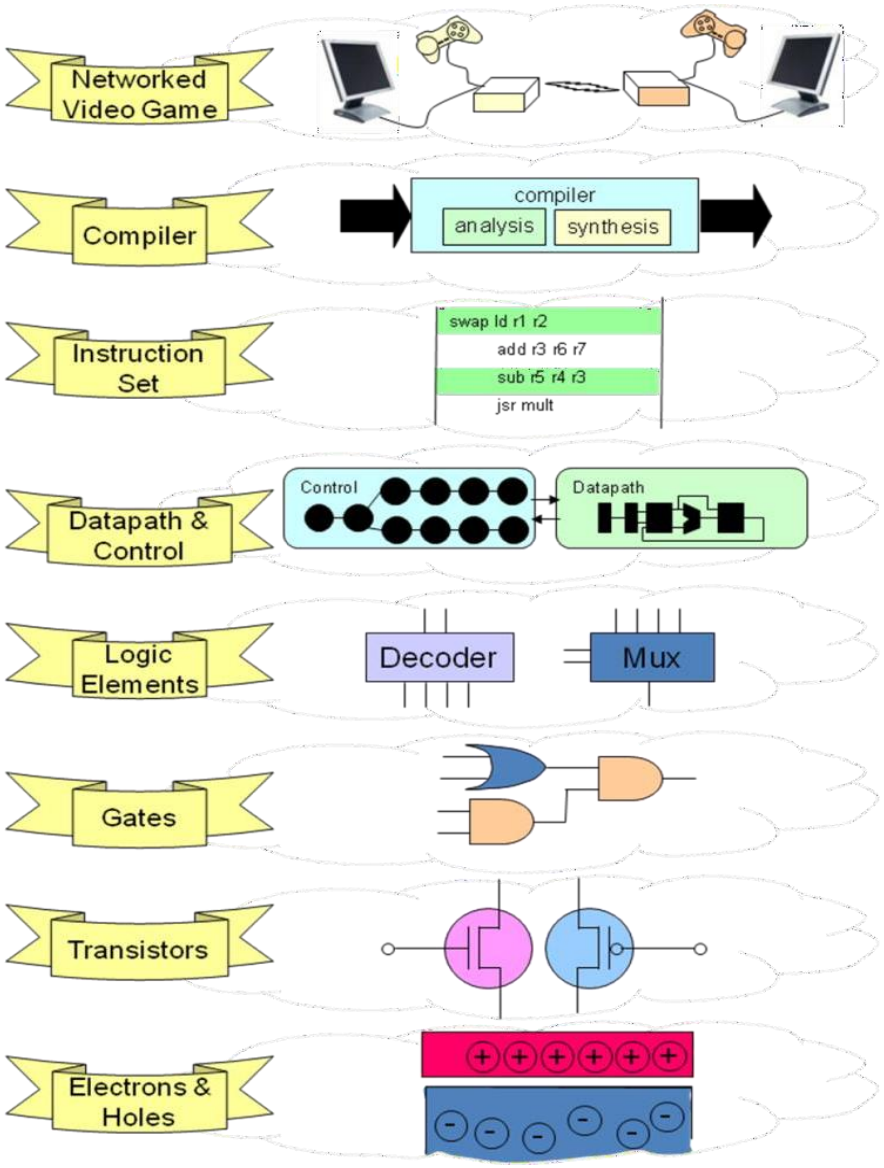
We gain and generate insight by solving problems

How do we ensure problems are solved by electrons?





FROM ELECTRONS & HOLES TO A MULTIPLAYER VIDEO GAME



In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language).

In computer science, an instruction set architecture (ISA), also called computer architecture, is an abstract model of a computer.

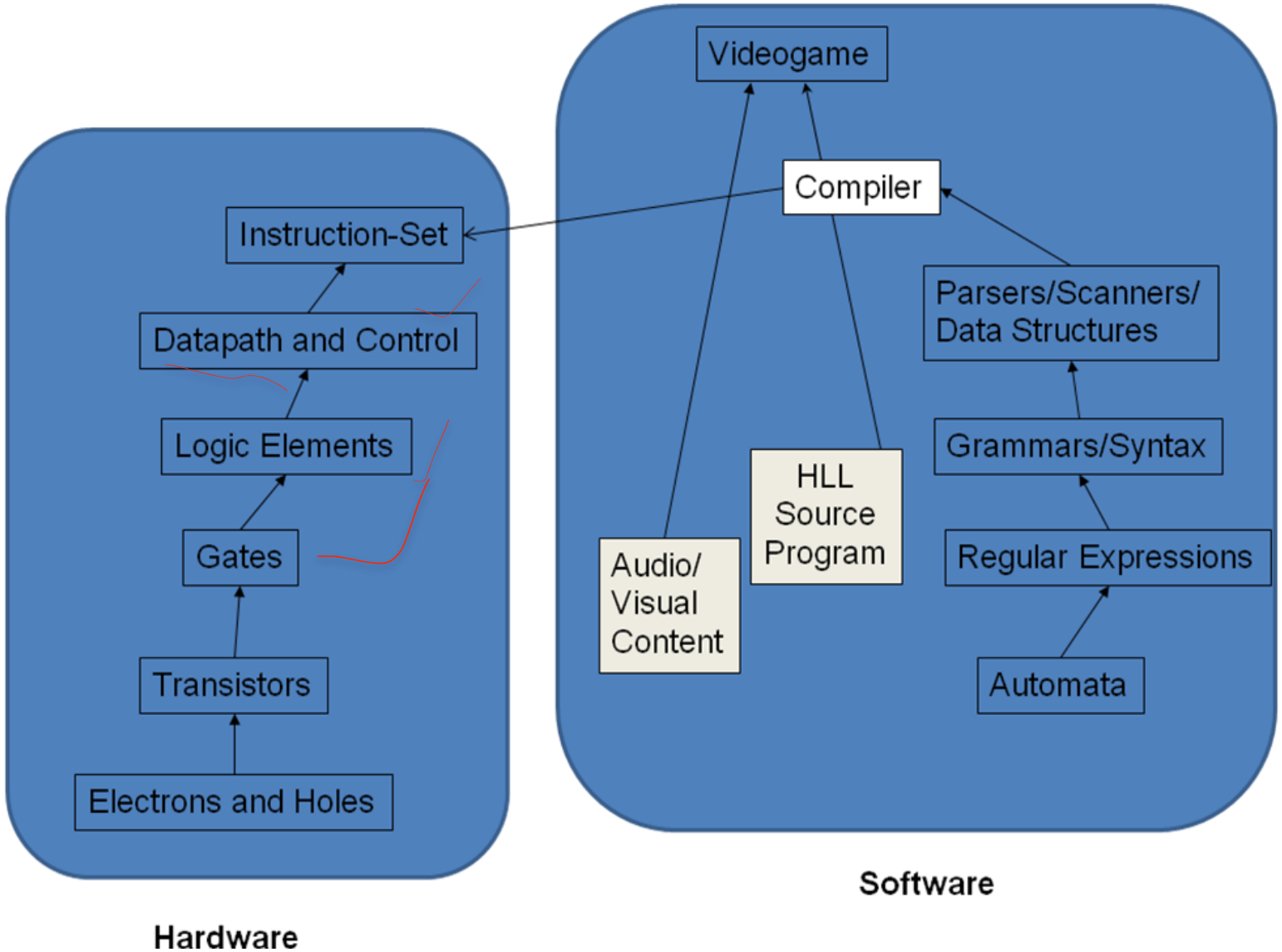
A data path is a collection of functional units such as **arithmetic** logic units (ALUs) or multipliers that perform data processing operations, registers, and buses.

A **logic** gate is an idealized or physical device that performs a **Boolean** function, a **logical** operation performed on one or more binary inputs that produces a single binary output.

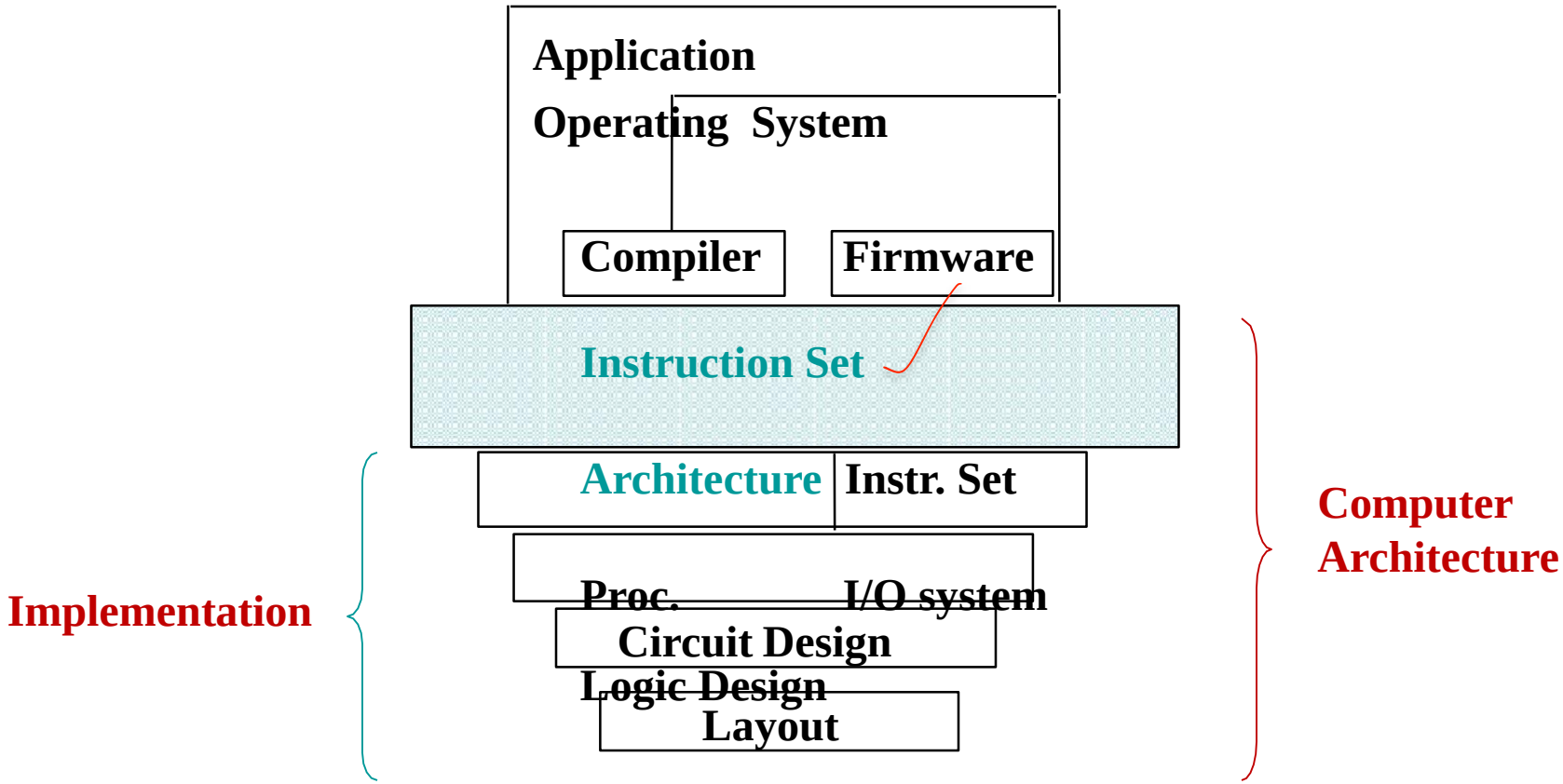
A transistor is **a semiconductor device used to amplify or switch electrical signals and power**.

A semiconductor is a material which has **an electrical conductivity** value falling between that of a **conductor**, such as copper**, and an insulator**, such as glass**.**

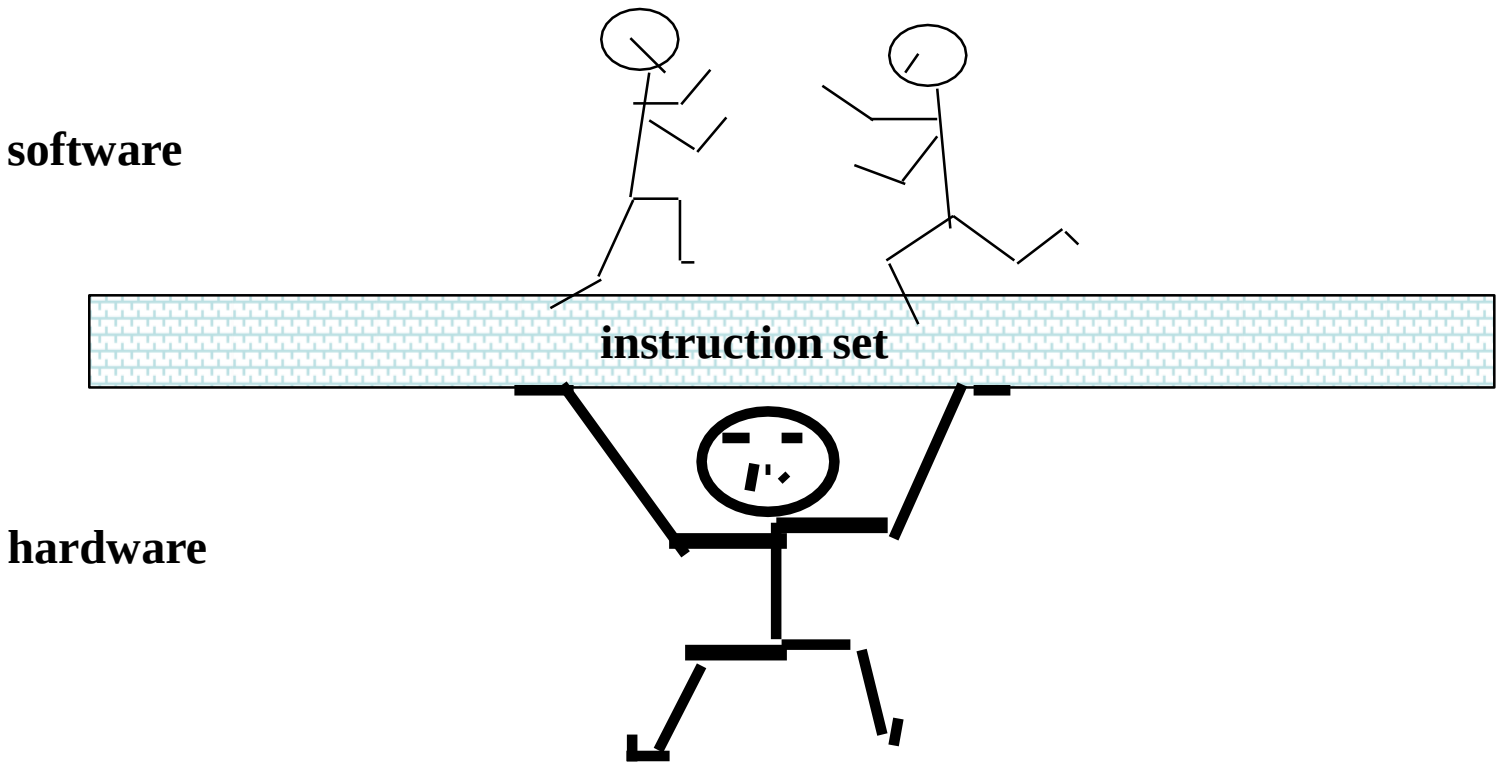
HARDWARE SOFTWARE INTERFACE



COMPUTER ARCHITECTURE



THE INSTRUCTION SET: A CRITICAL INTERFACE



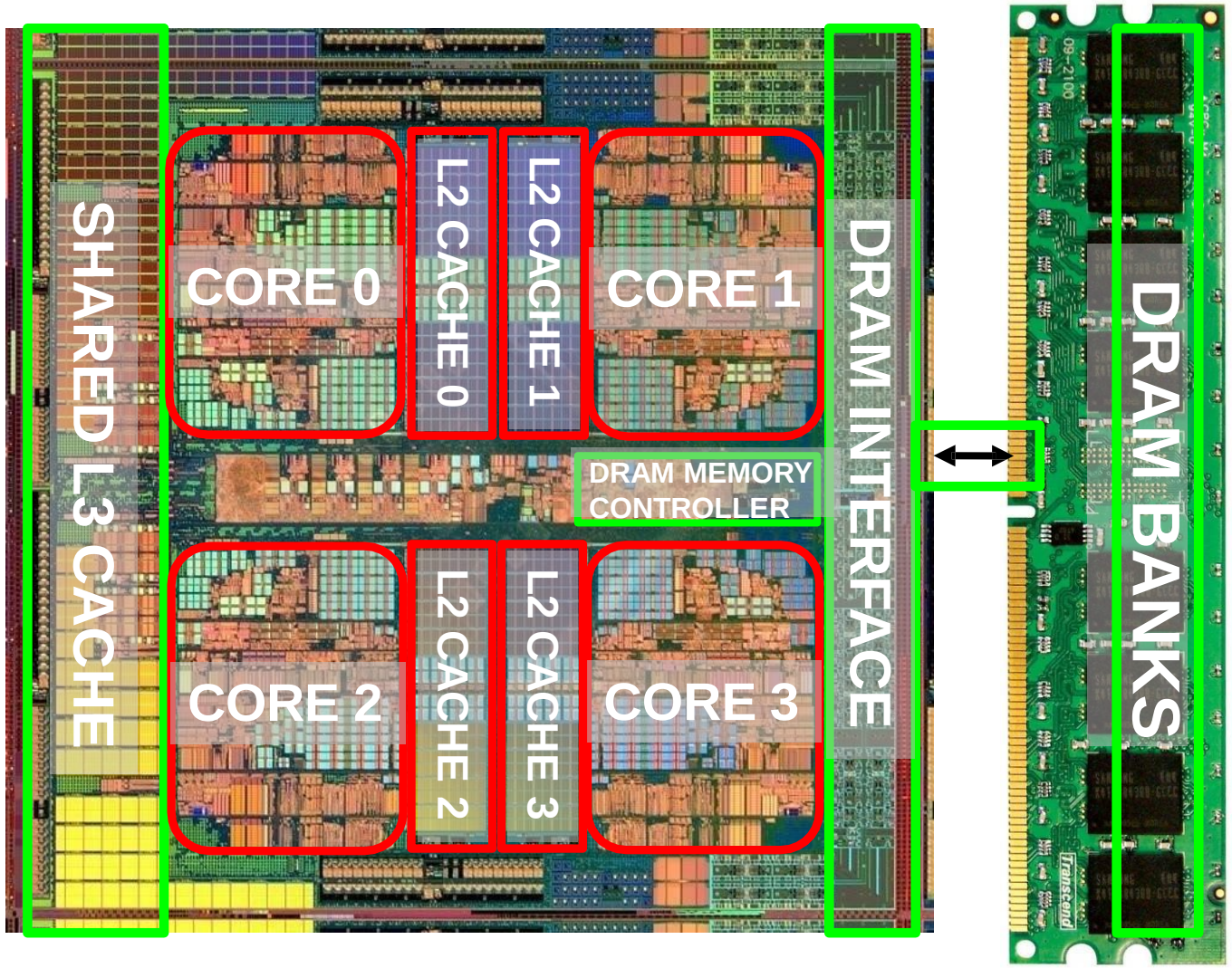
INSTRUCTION SET ARCHITECTURE

- A very important abstraction
  - interface between hardware and low-level software
  - standardizes instructions, machine language bit patterns, etc.
  - advantage: *different implementations (cost, performance, power) of the same architecture*
  - disadvantage: *sometimes prevents using new innovations*

- Common instruction set architectures:
  - IA-32, PowerPC, MIPS, SPARC, ARM, and others

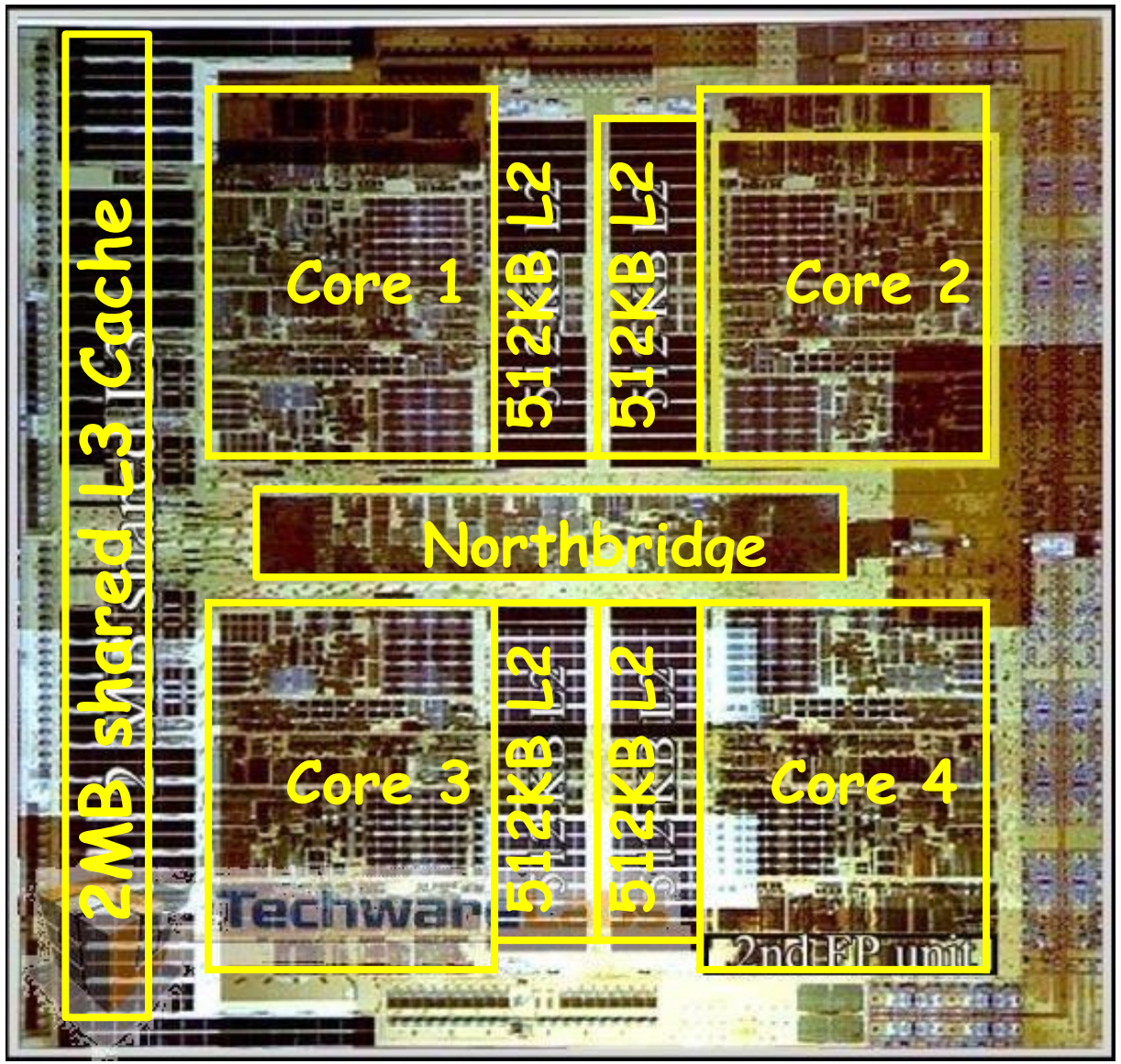
AN EXAMPLE: MULTI-CORE SYSTEMS

Multi-Core Chip



*Die photo credit: AMD Barcelona

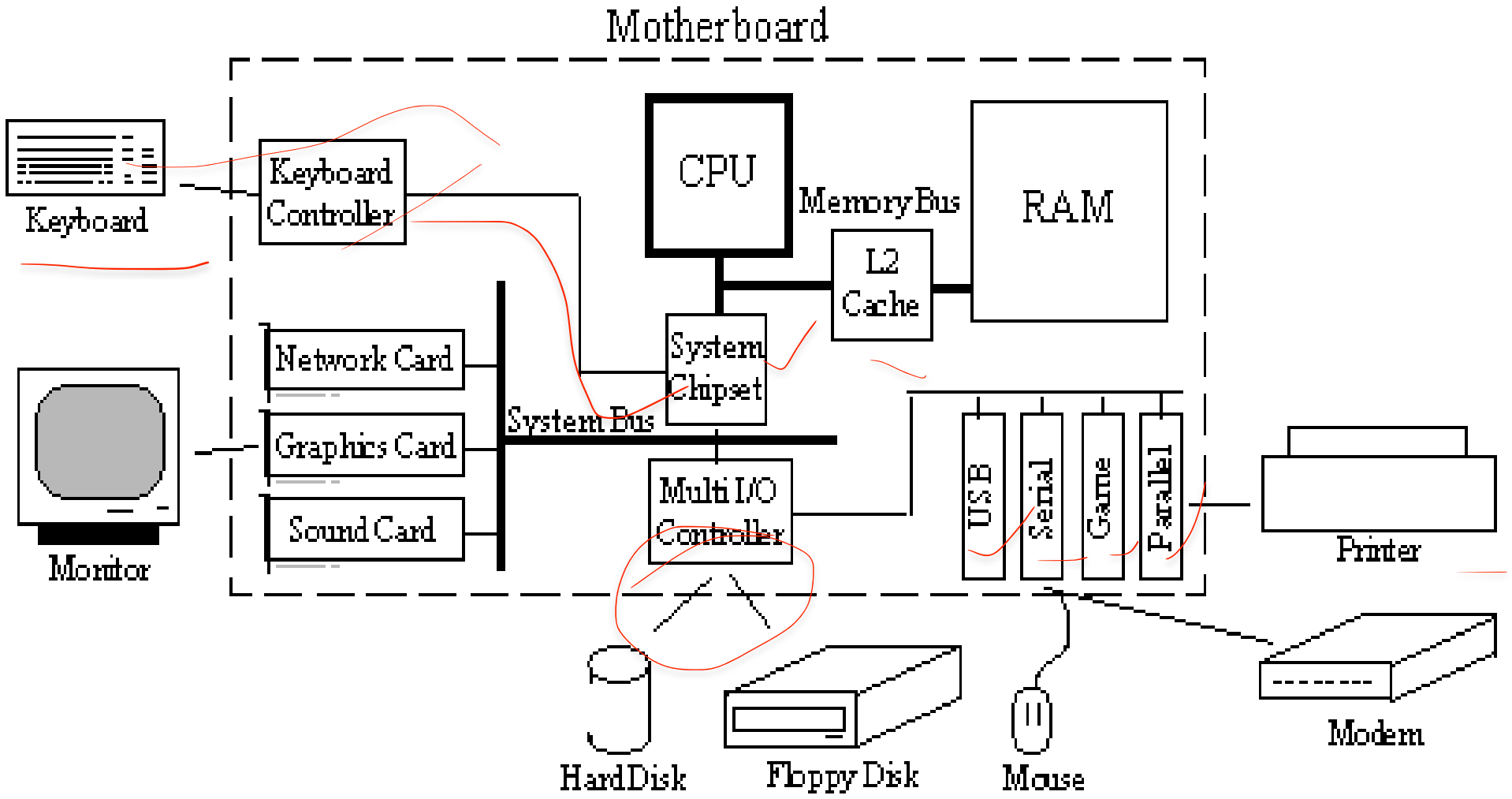
AMD'S BARCELONA MULTICORE CHIP



- Four out-of- order cores on one chip
- 1.9 GHz clock rate
- 65nm technology
- Three levels of caches (L1, L2, L3) on chip
- Integrated Northbridge

http://www.techwarelabs.com/reviews/processors/barcelona/

FUNCTION UNITS IN A COMPUTER



MULTICORES-EXAMPLES

- The power challenge has forced a change in the design of microprocessors
- Since 2002 the rate of improvement in the response time of programs on desktop computers has slowed from a factor of 1.5 per year to less than a factor of 1.2 per year
- Since 2006, all desktop and server companies are shipping microprocessors with multiple processors – cores – per chip

Product	AMD Barcelona	Intel Nehalem	IBM Power 6	Sun Niagara 2
Cores per chip	4	4	2	8
Clock rate	2.5 GHz	~2.5 GHz?	4.7 GHz	1.4 GHz
Power	120 W	~100 W?	~100 W?	94 W

The plan is to double the number of cores per chip per-generation (about every two years)

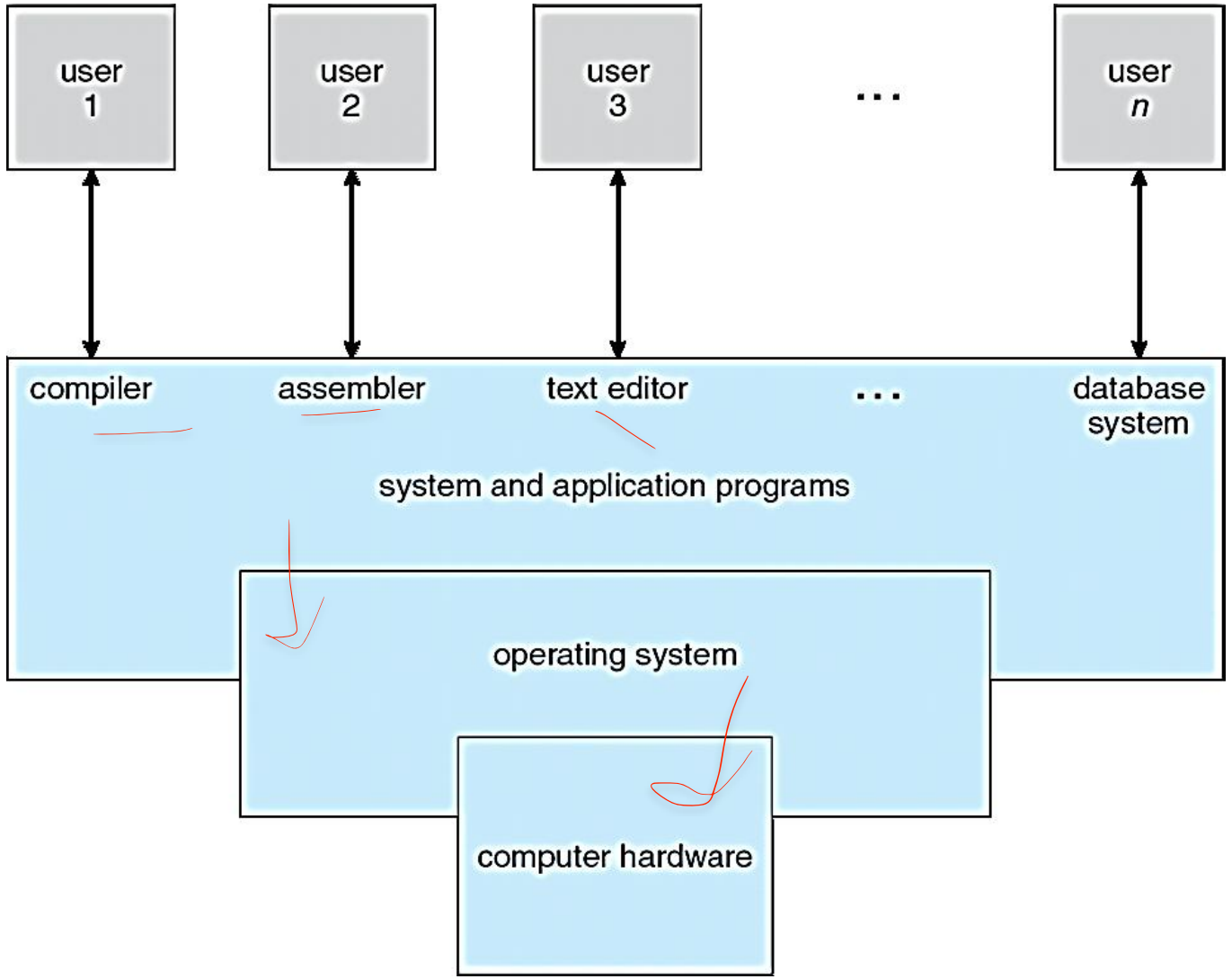
WHAT IS AN OPERATING SYSTEM?

- A program that acts as an intermediary between a user of a computer and the computer hardware
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

COMPUTER SYSTEM STRUCTURE

- Computer system can be divided into four components:
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

FOUR COMPONENTS OF A COMPUTER SYSTEM



WHAT OPERATING SYSTEMS DO

- Depends on the point of view
- Users want convenience, **ease of use** and **good performance**
  - Don't care about **resource utilization**
- But shared computer such as **mainframe** or **minicomputer** must keep all users happy
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- Handheld computers are resource poor, optimized for usability and battery life
- Some computers have little or no user interface, such as embedded computers in devices and automobiles

OPERATING SYSTEM DEFINITION

- OS is a **resource allocator**
  - Manages all resources
  - Decides between conflicting requests for efficient and fair resource use
- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the computer

OPERATING SYSTEM DEFINITION (CONT.)

- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is a good approximation
  - But varies wildly
- “The one program running at all times on the computer” is the **kernel**.
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.

COMPUTER STARTUP

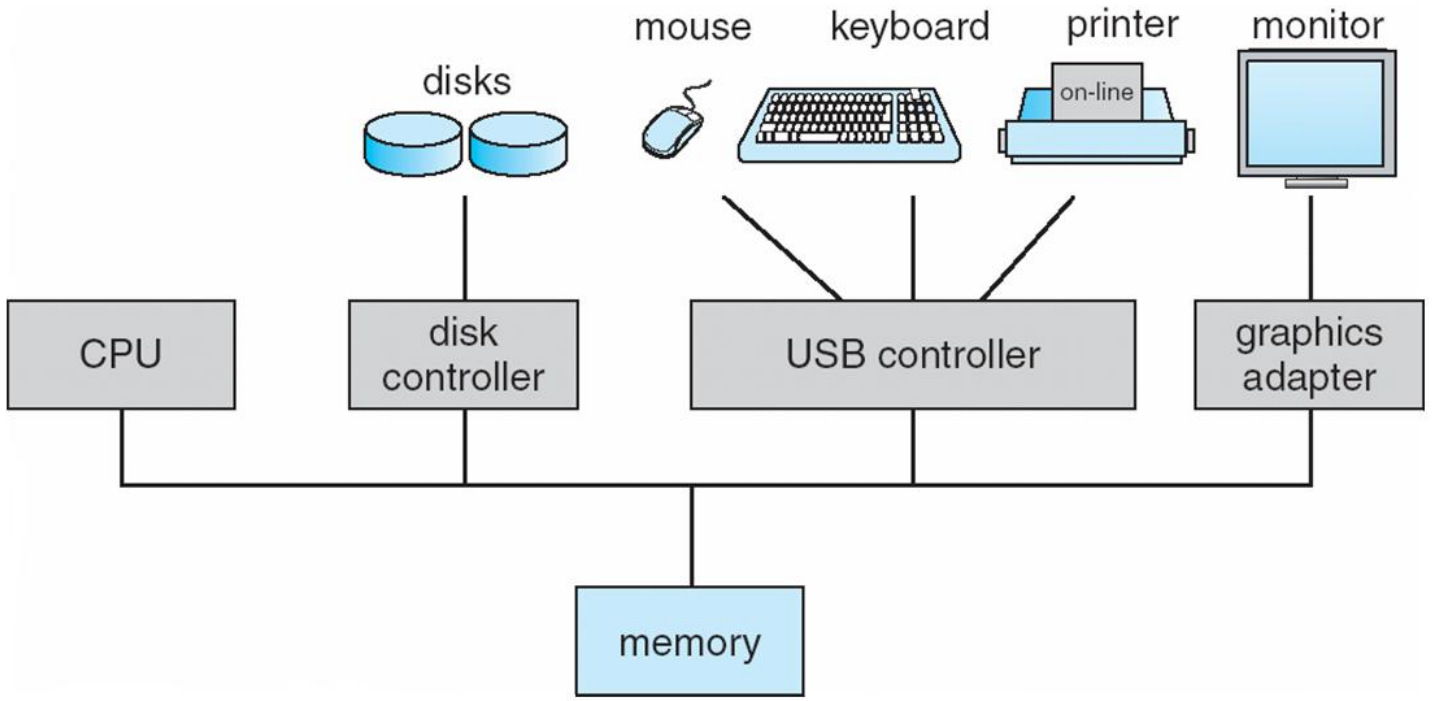
- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution

SYSTEM BOOT

- When power initialized on system, execution starts at a fixed memory location
  - Firmware ROM used to hold initial boot code
- Operating system must be made available to hardware so hardware can start it
  - Small piece of code – **bootstrap loader**, stored in **ROM** or **EEPROM** locates the kernel, loads it into memory, and starts it
  - Sometimes two-step process where **boot block** at fixed location loaded by ROM code, which loads bootstrap loader from disk
- Common bootstrap loader, **GRUB**, allows selection of kernel from multiple disks, versions, kernel options
- Kernel loads and system is then **running**

COMPUTER SYSTEM ORGANIZATION

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



COMPUTER-SYSTEM OPERATION

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

COMMON FUNCTIONS OF INTERRUPTS

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**

Examples of trap or exception -- breakpoint, division by zero, invalid memory access

INTERRUPT HANDLING

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt

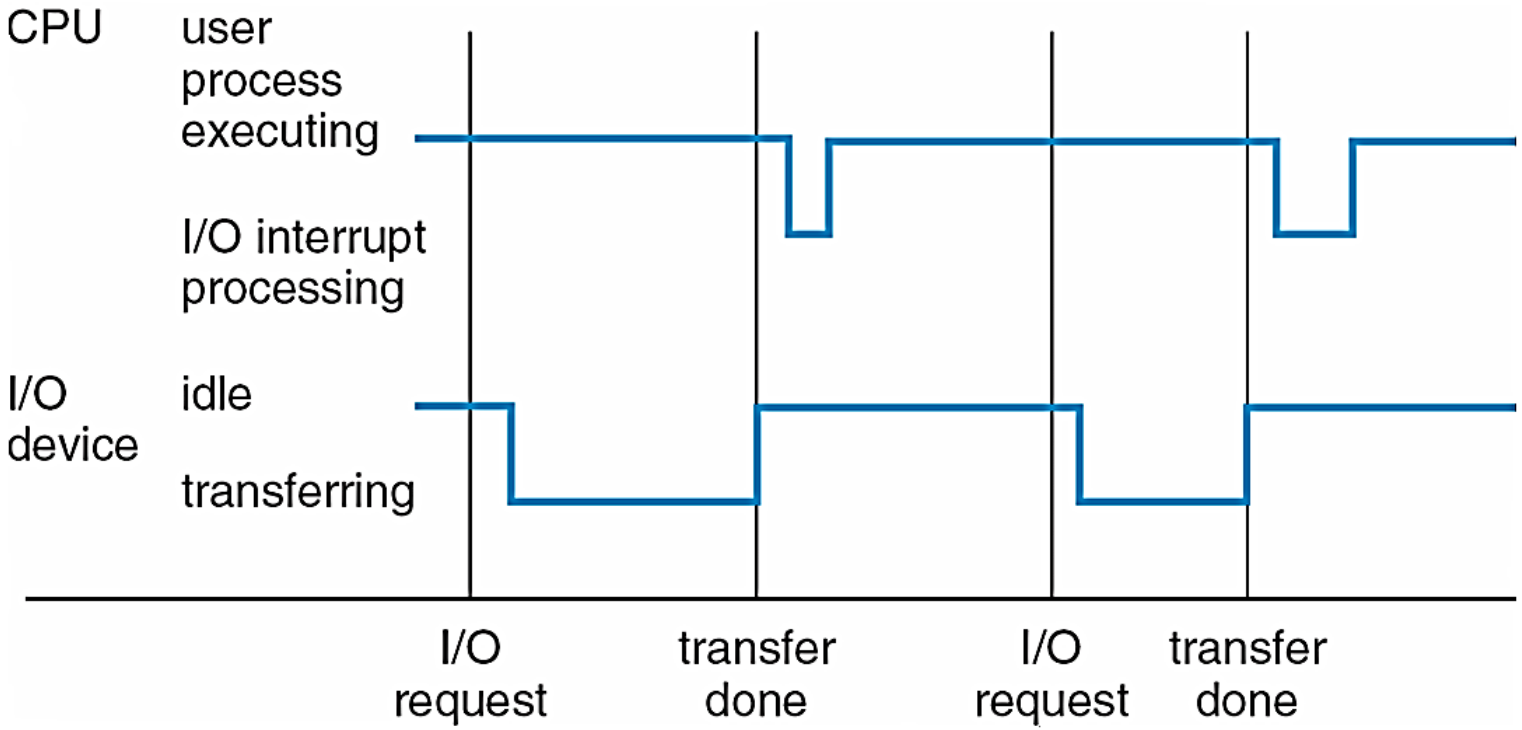
Polled interrupt

- A specific type of I/O interrupt
- Indicates that a device is ready to be read
- Does not indicate the exact device
- Interrupt controller send a signal (**Poll**) to each device to determine which one made the request

Vector interrupt

- The interrupting device directs the processor to the appropriate interrupt service routine.

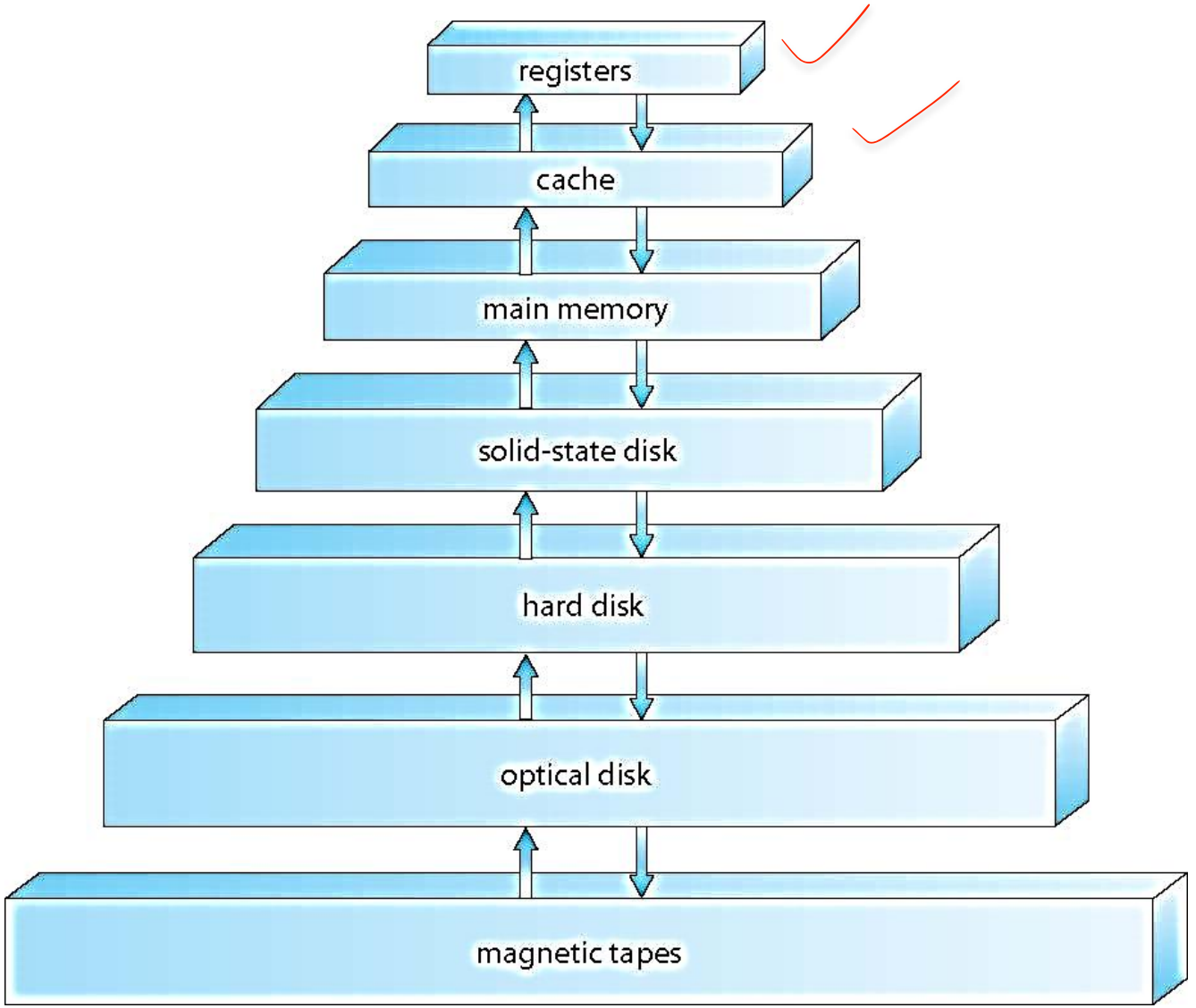
INTERRUPT TIMELINE



STORAGE STRUCTURE

- Main memory – only large storage media that the CPU can access directly
  - **Random access**
  - Typically **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular

STORAGE-DEVICE HIERARCHY



THANK YOU!