ESTGOH
Escola Superior de Tecnologia e Gestão de Oliveira do Hospital
Instituto Politécnico de Coimbra

Sistemas Operativos

01 – Introduction

Disclaimer

These slides and notes are based on the contents of the book:

N. Dale and J. Lewis, "*Computer Science Illuminated*", 7th Edition, Jones and Bartlett Publishers, 2019

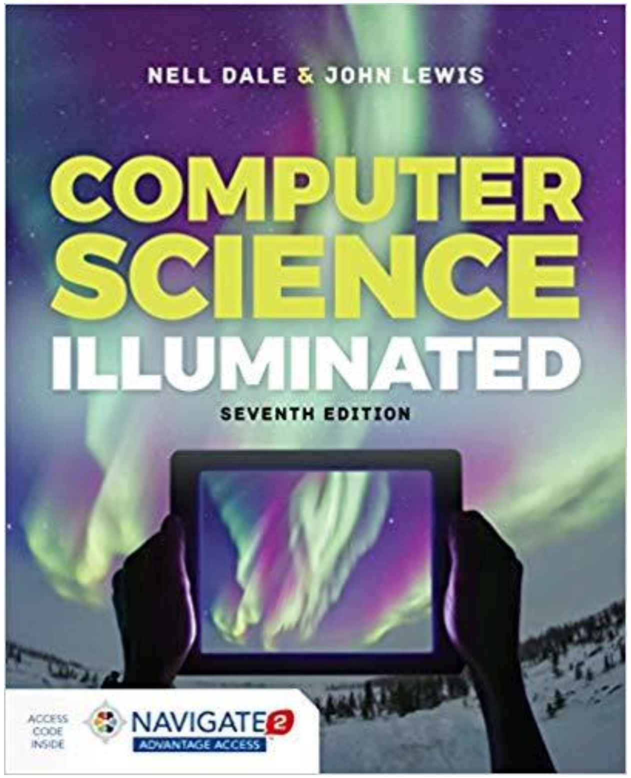


Table of Contents

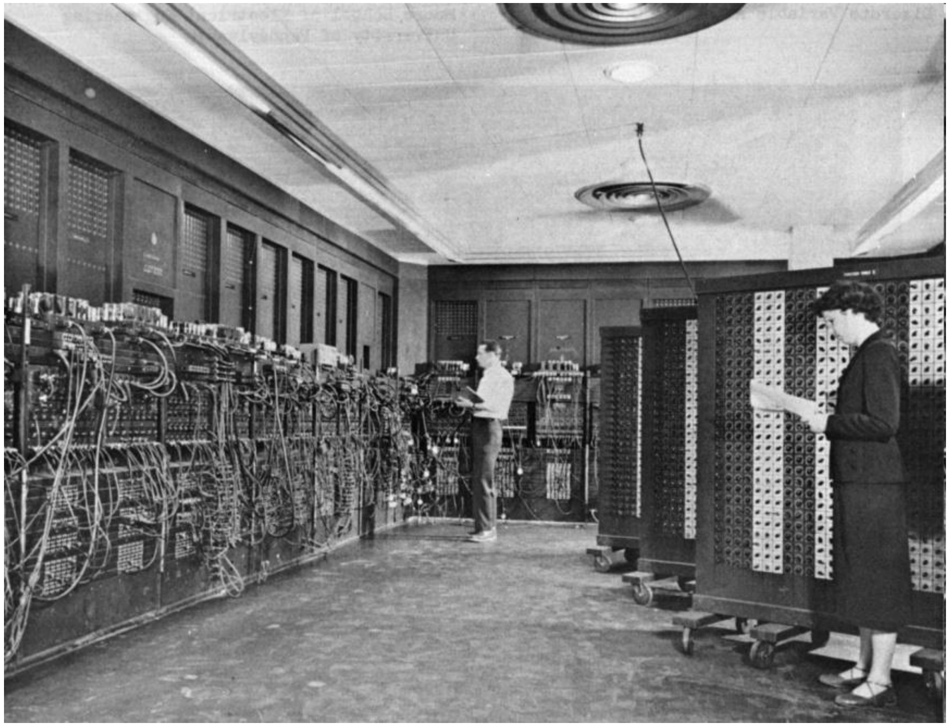
Table of Contents

- **PC Hardware**
  - **Motherboard**
  - **Central Processor Unit (CPU)**
  - **Memory (RAM, ROM, Cache)**
  - **Chipset**
    - Northbridge
    - Southbridge
  - **BIOS**
  - **Secondary Storage Devices**
    - Hard Disk
    - Laser Disks
  - **Video and Audio Standards**
  - **System Clock**
- **Non Von Neumann Architectures**
  - **Synchronous processing**
  - **Pipelining**
  - **Shared-memory configuration**

Table of Contents

Introduction

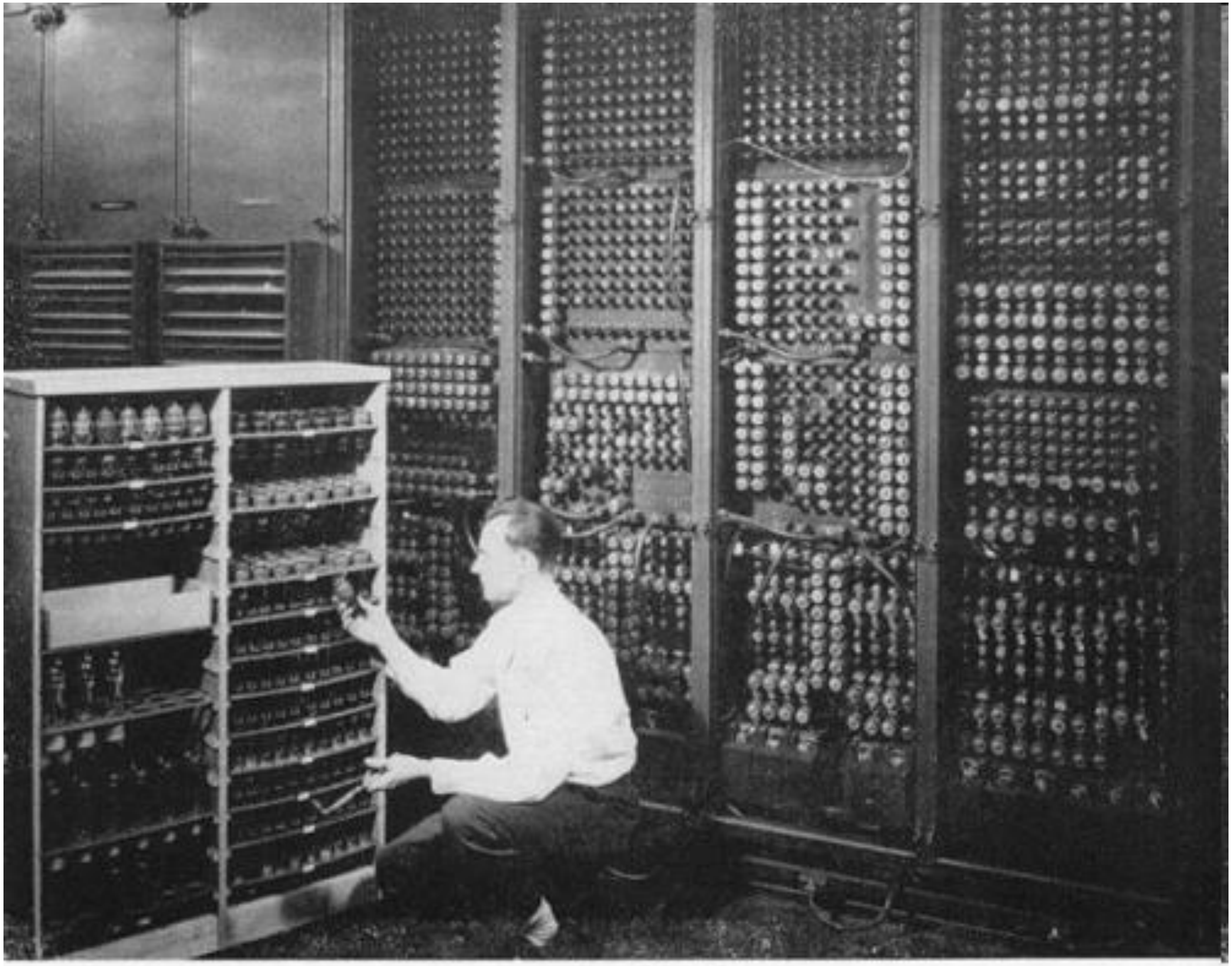
- **Background**
  - In the earliest electronic computing machines, programming was synonymous with connecting wires to plugs.
  - No layered architecture existed, so programming a computer was an exercise in algorithm design and then implementing the algorithm by rewiring the machine.
  - However, the tedium of rewiring the machine each time it had a new problem to solve or an old one to debug is annoying.



- **Background**
  - Before the work on the **ENIAC** (U.S. Army's Computer) was complete, John W. Mauchly and J. Presper Eckert conceived of an easier way to change the behavior of their calculating machine. They reckoned that memory devices, in the form of mercury delay lines, could provide a way to store program instructions. This would forever end the tedium of rewiring the machine.

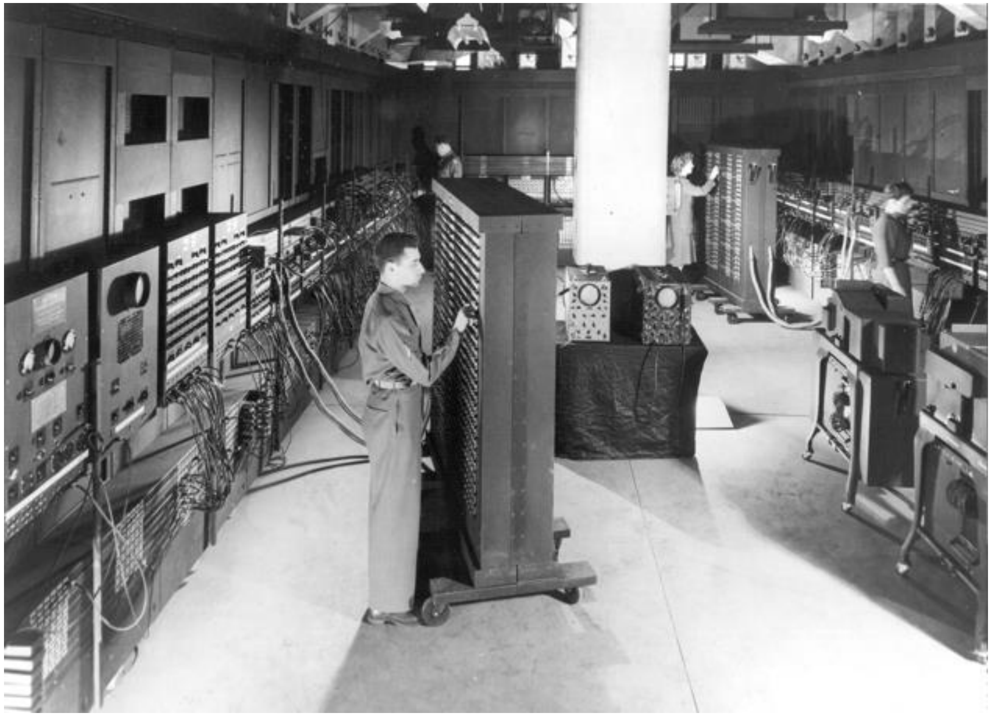
http://crosscuttingconcerns.com/Brief-Bio-John-W-Mauchly

- **Background**



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

- **Background**
  - Mauchly and Eckert documented their idea, proposing it as the foundation for their next computer.
  - Unfortunately, while they were involved in the top secret ENIAC project during World War II, <u>Mauchly and Eckert could not immediately publish their insight</u>.



https://en.wikipedia.org/wiki/ENIAC#/media/File:Classic_shot_of_the_ENIAC.jpg

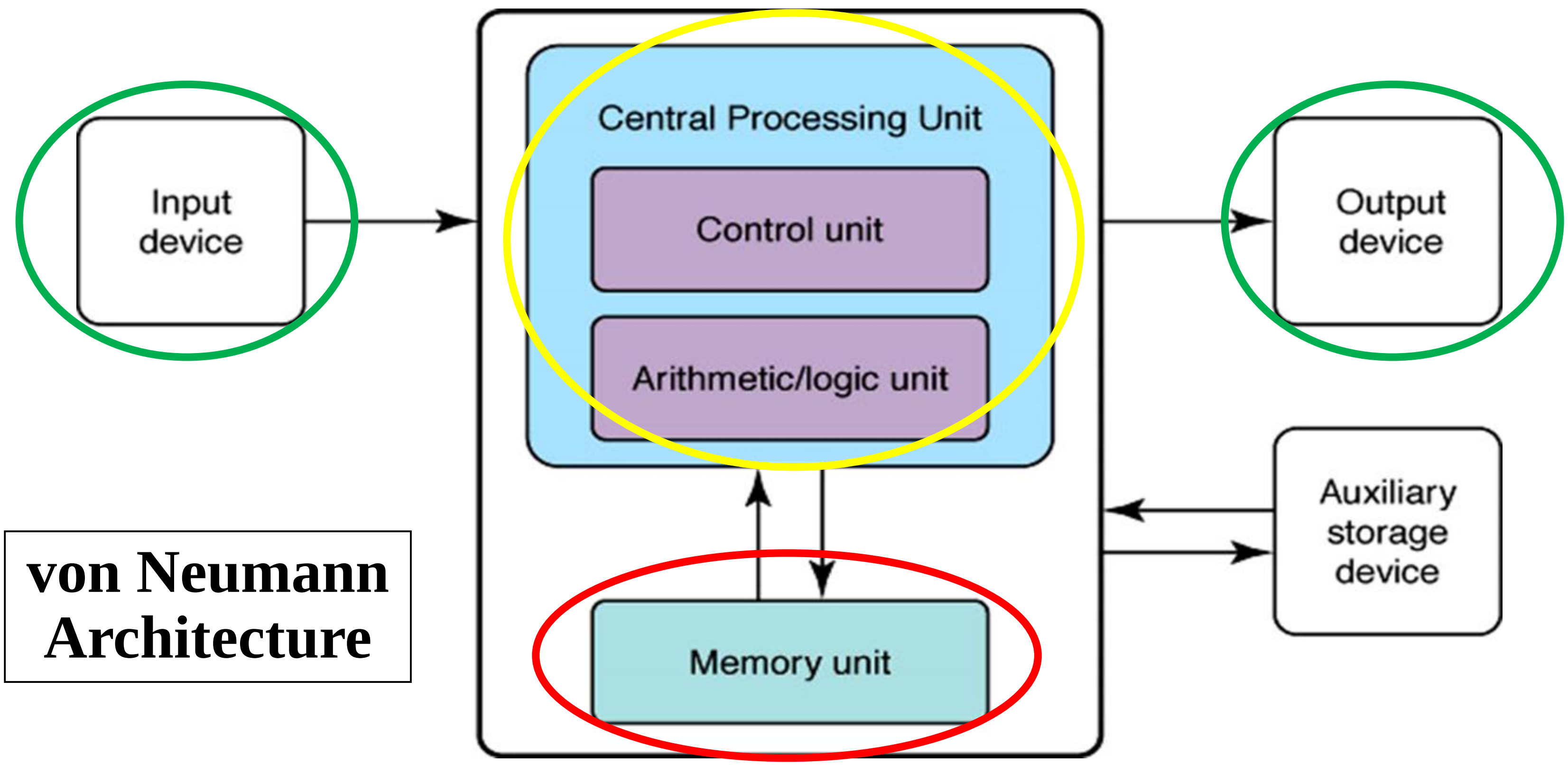
- **Background**
  - No such proscriptions, however, applied to a number of people working at the periphery of the ENIAC project. One of these people was a famous Hungarian mathematician named **John von Neumann** (pronounced von noy-man). After reading Mauchly and Eckert's proposal, von Neumann published and publicized the idea. So effective was he in the delivery of this concept that history has credited him with its invention.
  - All stored-program computers have come to be known as von Neumann systems using the **von Neumann architecture.**

´John von Neumann (1903-1957)

Table of Contents

- **Background**
- **Von Neumann Architecture**
  - **Components**
    - Memory
    - Central Processing Unit (CPU)
      - Arithmetic/Logic Unit (ALU)
      - Control Unit
    - Input/Output Units
    - BUS
  - **The Von Neumann Architecture Extended**
  - **Fetch-Execute Cycle**
    1. Fetch the Next Instruction
    2. Decode the Instruction
    3. Get Data If Nedeed
    4. Execute the Instruction

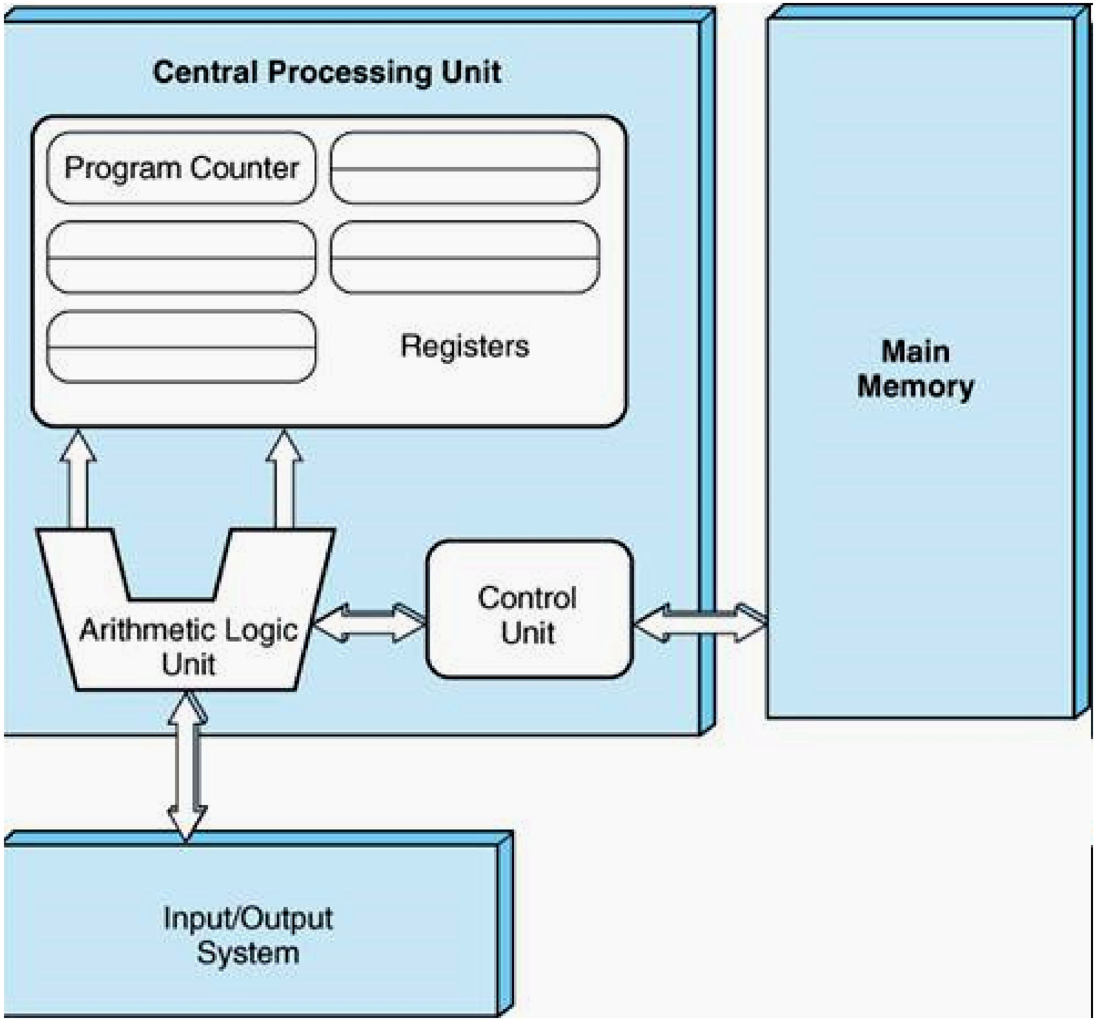
- **The von Neumann Architecture**
  - A major defining point in the history of computing was the realization in 1944-45 that data and instructions to manipulate the data are logically the same and could be stored in the same place. The computer design built around this principle is still the basis for computers today.



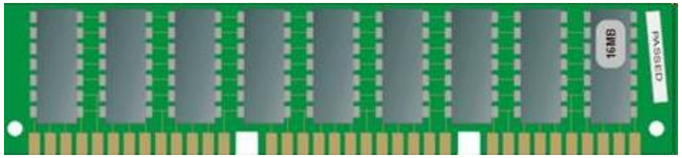
- **The von Neumann Architecture**
  - A general depiction of a von Neumann system satisfies at least the following characteristics:
    - **3 hardware systems:**
      - a central processing unit (CPU) containing:
        - a control unit;
        - an arithmetic/logic unit (ALU);
        - registers;
          - a Program Counter (PC);
          - a Instruction Register (IR)
      - a main memory system;
      - an I/O (Input/output) system.
    - **capacity to carry out sequential instruction processing.**
    - **a single data path between the CPU and main memory (von Neumann bottleneck).**

Introduction

- **The von Neumann Architecture**



- **The von Neumann Architecture**
  - **Components**

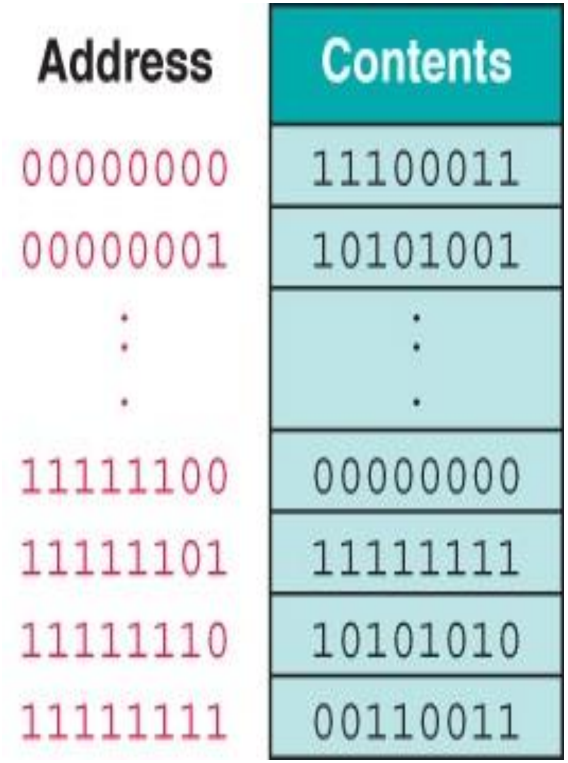


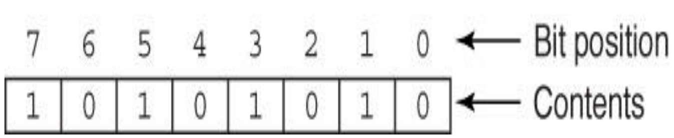
- **Memory**
- Memory is a collection of cells, each with a unique physical address.
- **memory's addressability:** the number of bits in each addressable location (cell), called the, varies from one machine to another. However, most computers are byte-addressable today.
- Relationship between the bits in the processor and the number of different addresses:

n bits can address 2^n different locations.

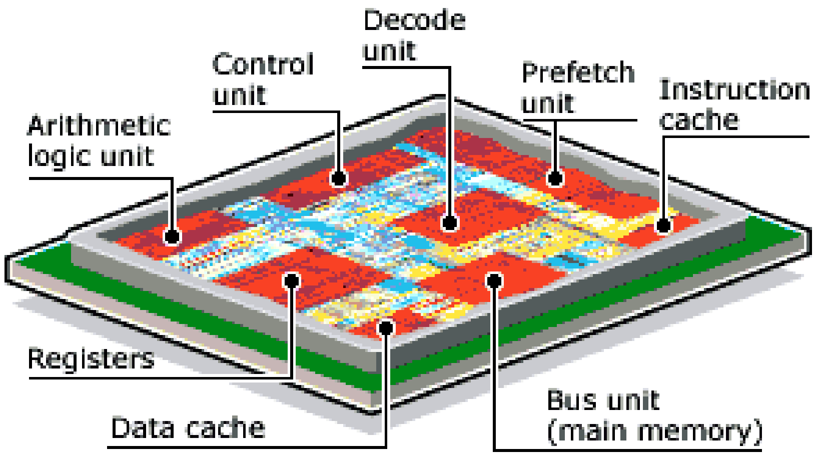
- **Example:**
  - A 32-bit machine can address 2^{32} cells of memory. In other words, it can address 4294967296 bytes of memory if the addressability of the machine is 8 bits. This means that the processor can distinguish 2^{32} different memory addresses.

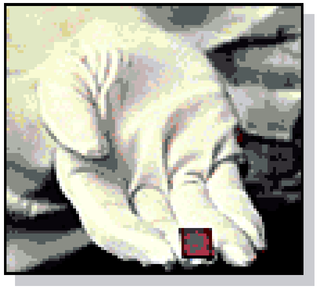
- **The von Neumann Architecture**
  - **Components**
  - **Memory**
  - The cells in memory are numbered consecutively beginning with 0. For example, if the addressability is 8, and there are 256 cells of memory, the cells would be addressed as shown in the figure.
  - What are the contents of address 11111110?
    - The bit pattern stored at that location is 10101010. What does it mean? Does location 11111110 contain an instruction? A integer with a sign? A two's complement value? Part of an image? Without knowing what the contents represent, we cannot determine what it means: It is just a bit pattern.





- **The von Neumann Architecture**
  - **Components**
  - **Central Processing Unit (CPU)**
  - **Arithmetic/logic Unit (ALU)**
  - The arithmetic/logic unit (ALU) is capable of performing basic:
    - <u>arithmetic operations such as adding, subtracting, multiplying, and dividing two numbers</u>.
    - <u>logical operations such as AND, OR, and NOT</u>.
  - The ALU operates on words; thus the <u>**word length** of a computer is the size of the quantities processed by the ALU</u>. The word length of the Pentium IV is 32 bits or 4 bytes.





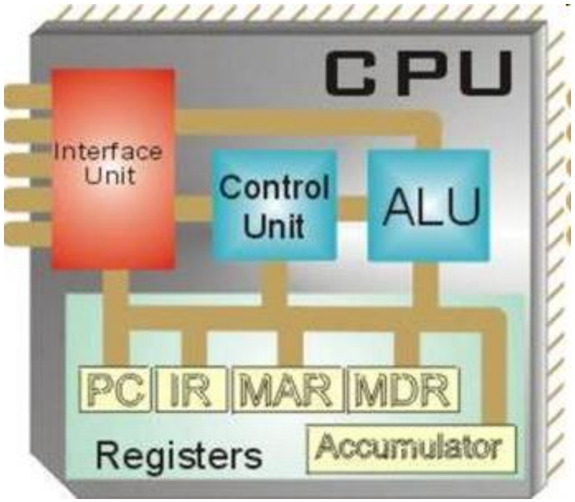
The key processing elements (above) contained in such a tiny chip (right).

- **The von Neumann Architecture**
  - **Components**
  - **Central Processing Unit (CPU)**
  - **Arithmetic/logic Unit (ALU)**
  - Most modern ALUs have a small amount of special storage units called registers. These registers contain one word and are used to store information that is needed again immediately. For example, in the calculation of

    One * (Two + Three)

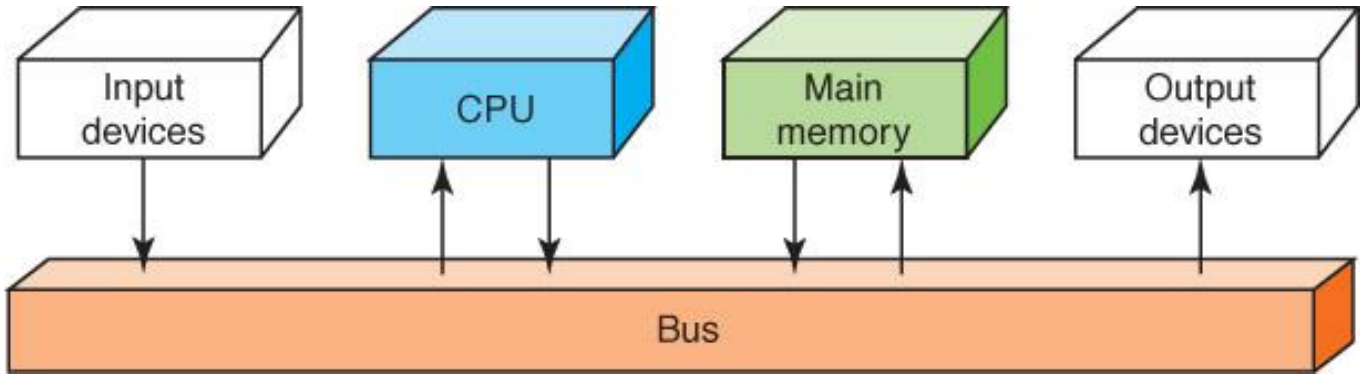
  - Two is first added to Three and the result is then multiplied by One.
  - Rather than storing the result of adding Two and Three in memory and then retrieving it to multiply it by One, the result is left in a register and the contents of the register is multiplied by One. Access to registers is much faster than access to memory locations.

- **The von Neumann Architecture**
  - **Components**
  - **Central Processing Unit (CPU)**
  - **Control Unit**
  - The control unit is the organizing force in the computer, for it is in charge of the fetch-execute cycle, discussed next.
  - There are two registers in the control unit.
    - The **instruction register (IR)** contains the instruction that is being executed
    - The **program counter (PC)** contains the address of the next instruction to be executed.
  - Because the ALU and the control unit work so closely together, they are often thought of as one unit called the *Central Processing Unit, or CPU*.



- **The von Neumann Architecture**
  - **Components**
  - **Input/Output Units**
  - Input and output units are the <u>channels through which the computer communicates with the outside world:</u>
    - **input unit** is a device through which data and programs from the outside world are entered into the computer. The first input units interpreted holes punched on paper tape or cards. Modern-day input devices include the terminal <u>keyboard, the mouse</u>, and scanning devices used at supermarkets.
    - **output unit** is a device through which results stored in the computer memory are made available to the outside world. The most common output devices are <u>printers and video display terminals.</u>

- **The von Neumann Architecture**
  - **Components**
  - **Bus**
  - The figure shows the flow of information through the parts of a von Neumann machine.



  - The parts are connected to one another by a collection of wires called a **bus** through which data travels within the computer.
  - In a personal computer, the components in a von Neumann machine reside physically in a printed circuit board called the <u>motherboard</u>. The motherboard also has connections for attaching other devices to the bus, such as a mouse, a keyboard, or additional storage device.

Table of Contents

- **Background**
- **Von Neumann Architecture**
    - **Components**
        - Memory
        - Central Processing Unit (CPU)
            - Arithmetic/Logic Unit (ALU)
            - Control Unit
        - Input/Output Units
        - BUS
    - **The Von Neumann Architecture Extended**
    - **Fetch-Execute Cycle**
        1. Fetch the Next Instruction
        2. Decode the Instruction
        3. Get Data If Nedeed
        4. Execute the Instruction

- **The von Neumann Architecture**
  - **The von Neumann Architecture Extended**
  - The ideas present in von Neumann architecture have been extended.
  - The **system bus model** improve the single data bus (von Neumann bottleneck):
    - **data bus** moves data from main memory to the CPU registers (and vice versa).
    - **address** bus holds the address of the data that the data bus is currently accessing.
    - **control bus** carries the necessary control signals that specify how the information transfer is to take place (write, read, ..).
  - **Other enhancements** to the von Neumann architecture include:
    - the use of index registers for addressing
    - adding floating point data
    - using <u>interrupts</u> and asynchronous I/O
    - adding virtual memory
    - adding <u>general registers</u>

- **The von Neumann Architecture**
  - **The von Neumann Architecture Extended**

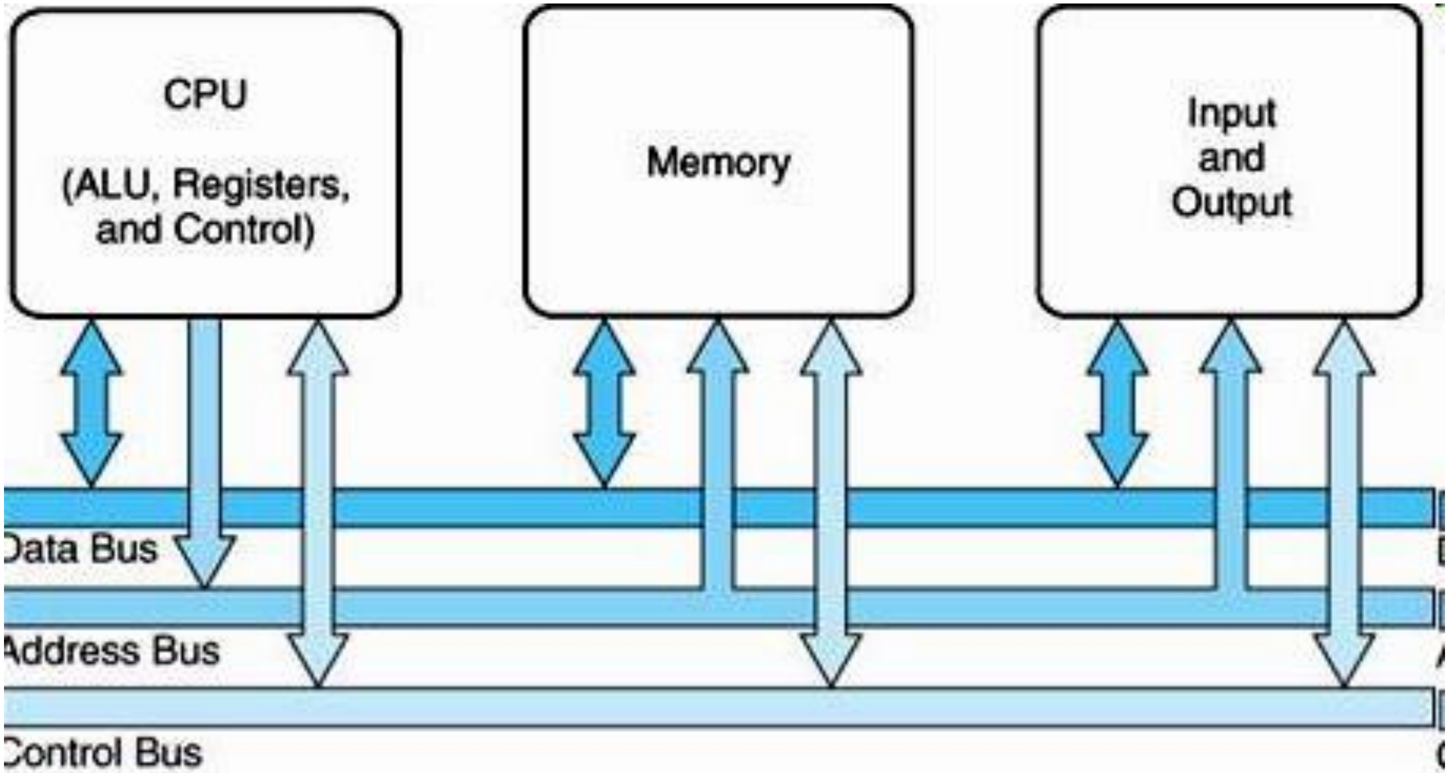
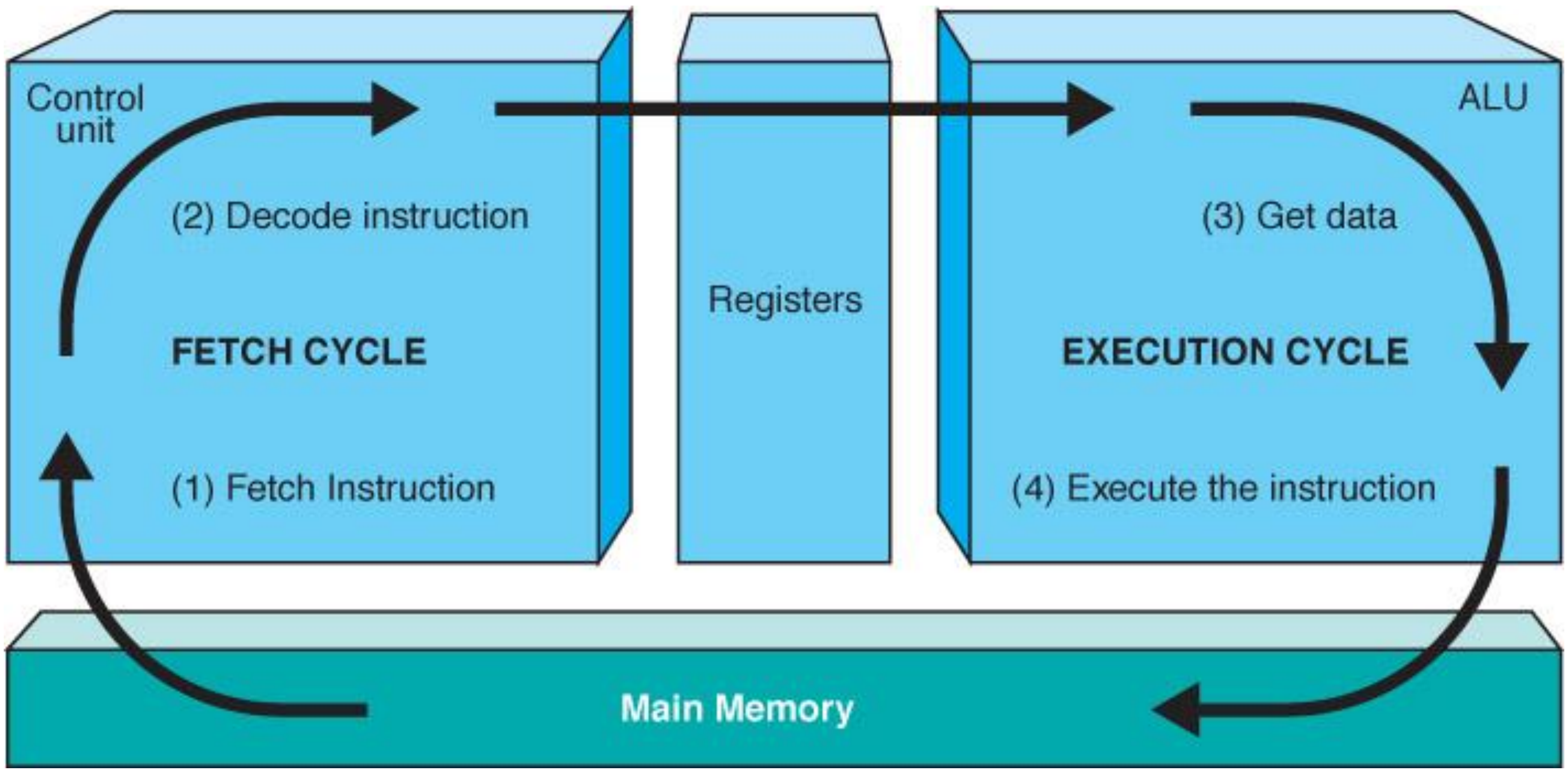


Table of Contents

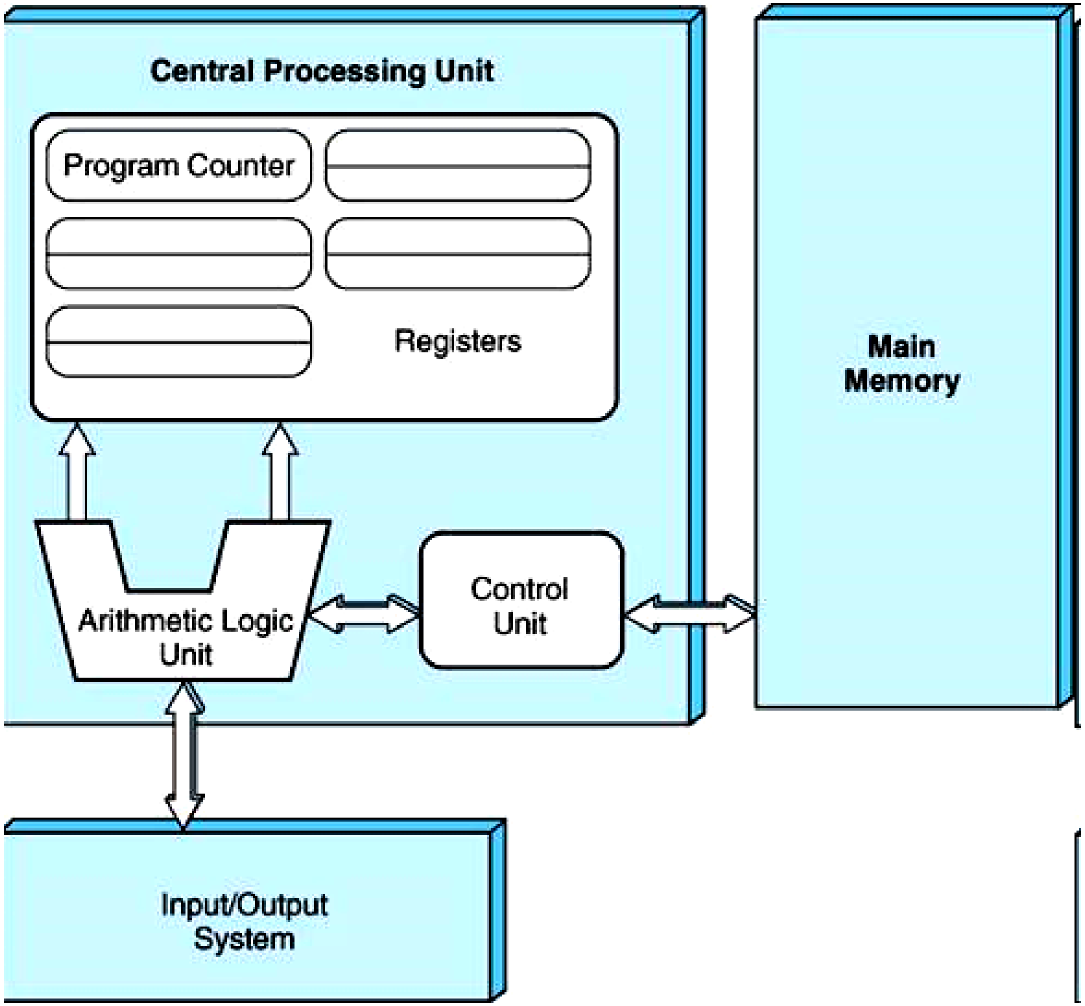
Introduction

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - Principal of the von Neumann machine:
    - Data and instructions are stored in memory and treated alike. This means that instructions and data are both addressable.
    - Instructions are stored in contiguous memory locations; data to be manipulated are stored together in a different part of memory.
  - To start the fetch-execute cycle, the address of the first instruction is loaded into the Program Counter.
  - The steps in the processing cycle are:
    - **Fetch the next instruction.**
    - **Decode the instruction.**
    - **Get data if needed.**
    - **Execute the instruction.**

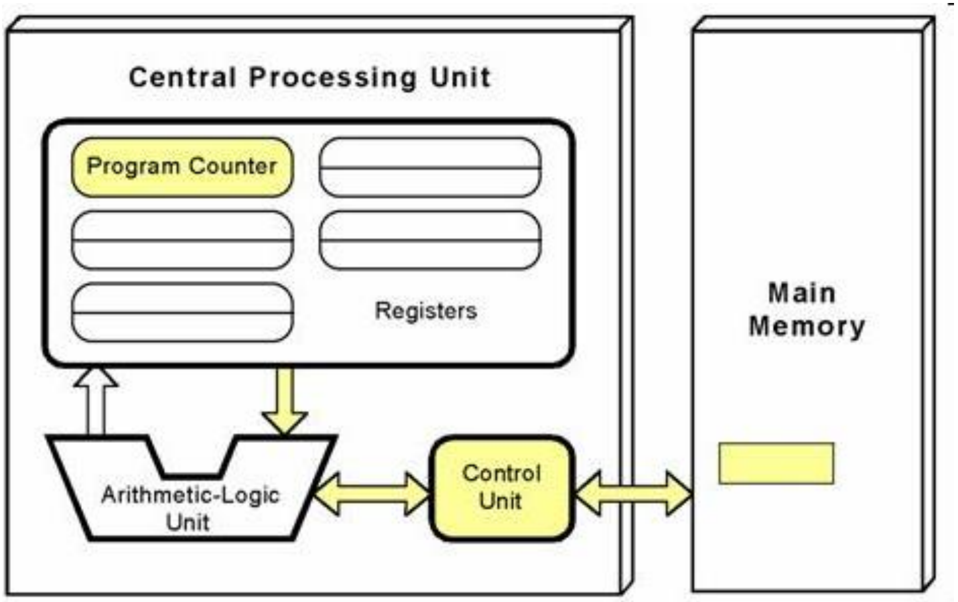
- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**



- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**



- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **1. Fetch the Next Instruction**
    - The ***Program Counter*** **(PC)** contains the address of the next instruction to be executed, so the control unit goes to the address in memory specified in the PC, makes a copy of the contents, and places the copy in the *Instruction Register* (IR). At this point the ***Instruction Register*** contains the instruction to be executed.

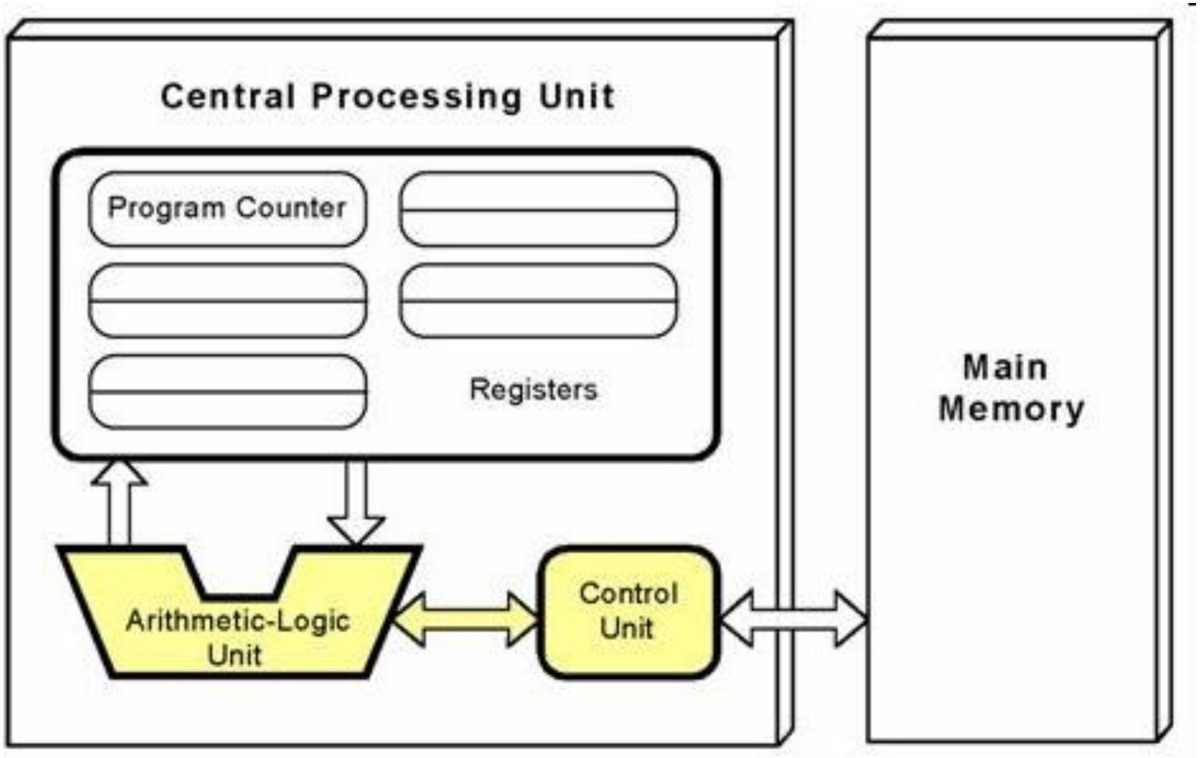


Introduction

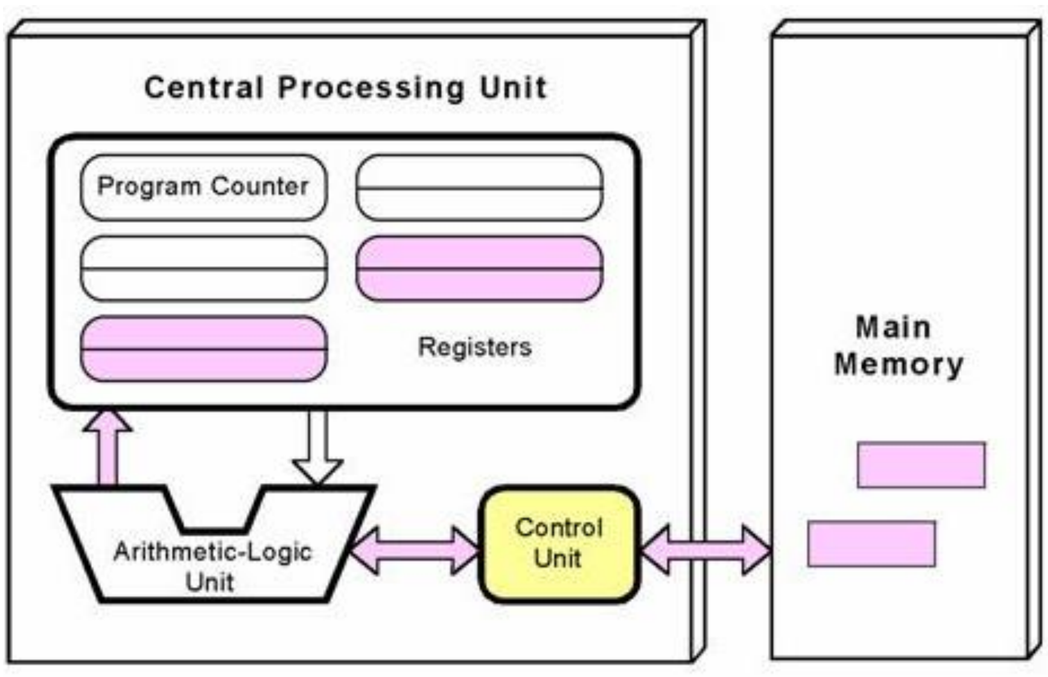
- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **1. Fetch the Next Instruction**
    - Before going on to the next step in the cycle, the *Program Counter* must be updated to hold the address of the next instruction to be executed when the current instruction has been completed. Because the instructions are stored contiguously in memory, adding 1 to the *Program Counter* should put the address of the next instruction into the PC. So the control unit increments the program counter. It is possible that the PC may be changed later by the instruction being executed.
    - Accessing memory takes one cycle.

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **2. Decode the Instruction**
    - In order to execute the instruction in the *Instruction Register*, the <u>control unit has to determine what instruction it is.</u>
      - It might be an instruction to access data from an input device, to send data to an output device, or to perform some operation on a data value.
    - At this phase, the instruction is decoded into control signals. That is, the logic of the circuitry in the CPU determines which operation is to be executed.
    - This step shows why a computer can only execute instructions that are expressed in its own machine language. The instructions themselves are literally built into the circuits.

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **2. Decode the Instruction**

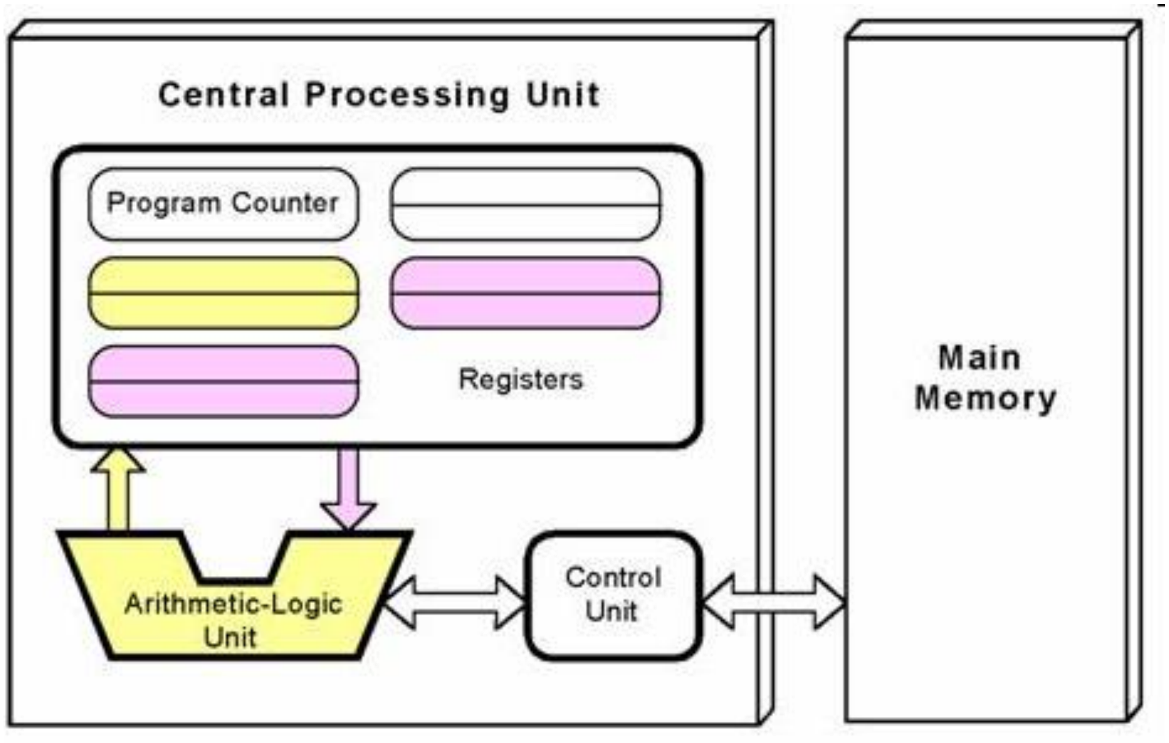


- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **3. Get Data If Needed**
  - It may be that the instruction to be executed requires additional memory accesses in order to complete its task.
    - For example, if the instruction says to add the contents of a memory location to a register, the control unit must get the contents of the memory location.



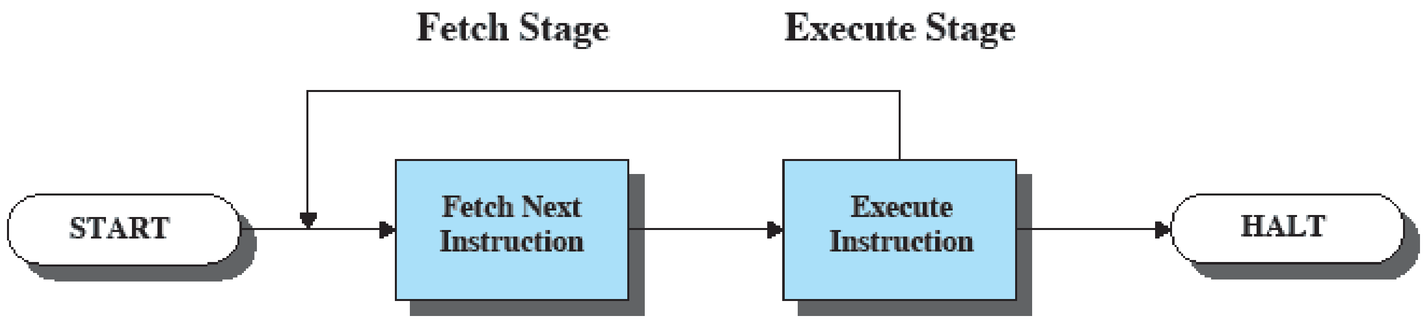
- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **4. Executes the Instruction**
    - Execution involves sending signals to the Arithmetic/Logic Unit to carry out the processing. In the case of adding a number to a register, the operand is sent to the ALU and added to the contents of the register.
    - **When the execution is complete**, the cycle begins again. If the last instruction was to add a value to the contents of a register, the next instruction probably says to store the results into a place in memory.
    - However, the next instruction might be a control instruction: an instruction that asks a question about the result of the last instruction and perhaps changes the contents of the program counter.

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **4. Executes the Instruction**



Introduction

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **Example:**
  - add two numbers and store the result at some place in the memory



(a) Instruction format

(b) Integer format

- **The von Neumann Architecture**
  - **Fetch-Execute Cycle**
  - **Example:**
  - add two numbers and store the result at some place in the memory

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

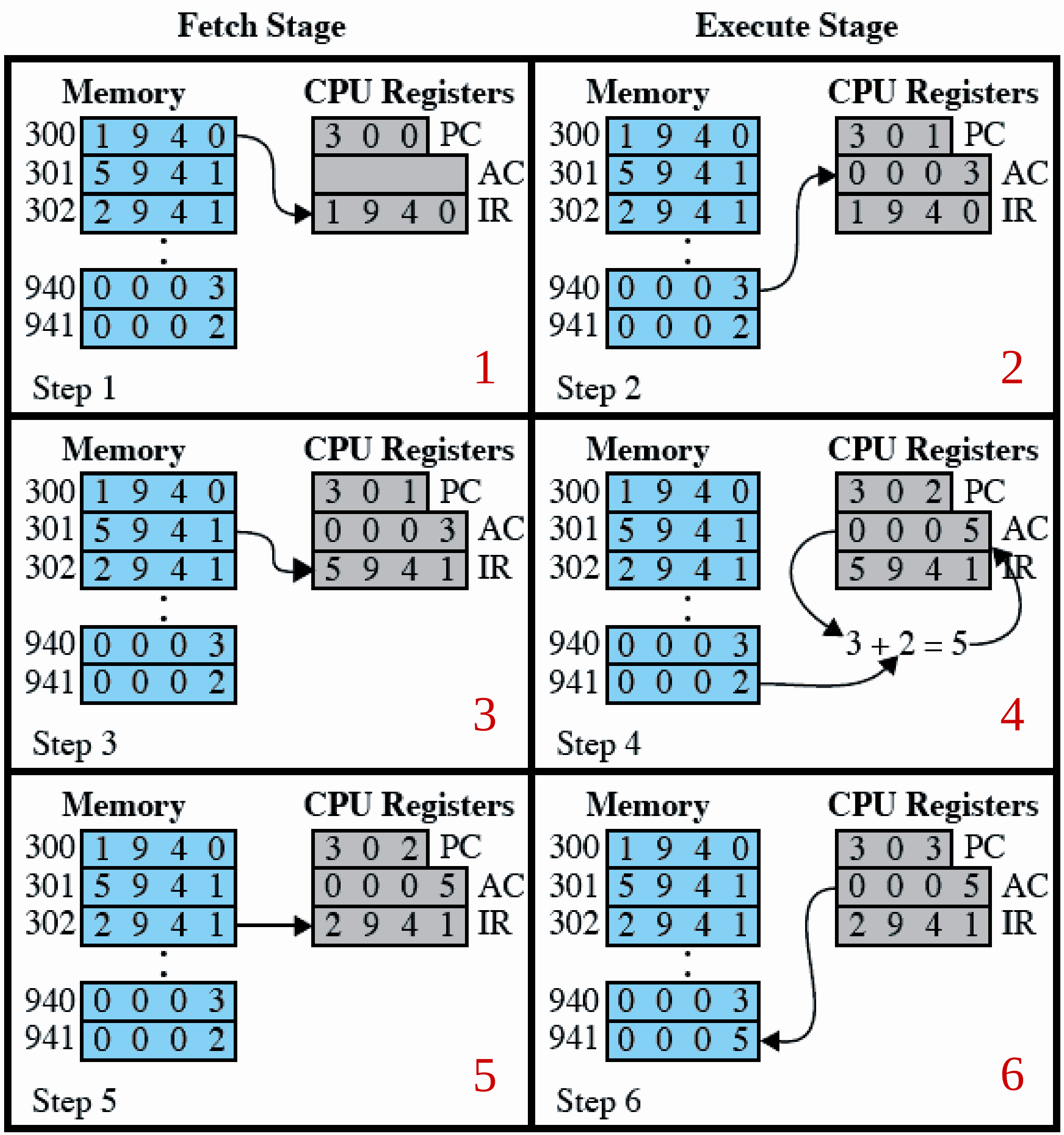
(d) Partial list of opcodes

Introduction

Next instruction is at 300 address. The contents are loaded in to the IR.

Next instruction is fetched.

Next instruction is fetched.



PC is incremented. The contents at 300 location are: first four bits are the Op-code (0001 = 1) and the remaining 12 bits indicate the address (9 4 0). The contents of the 940 location are loaded into the AC.

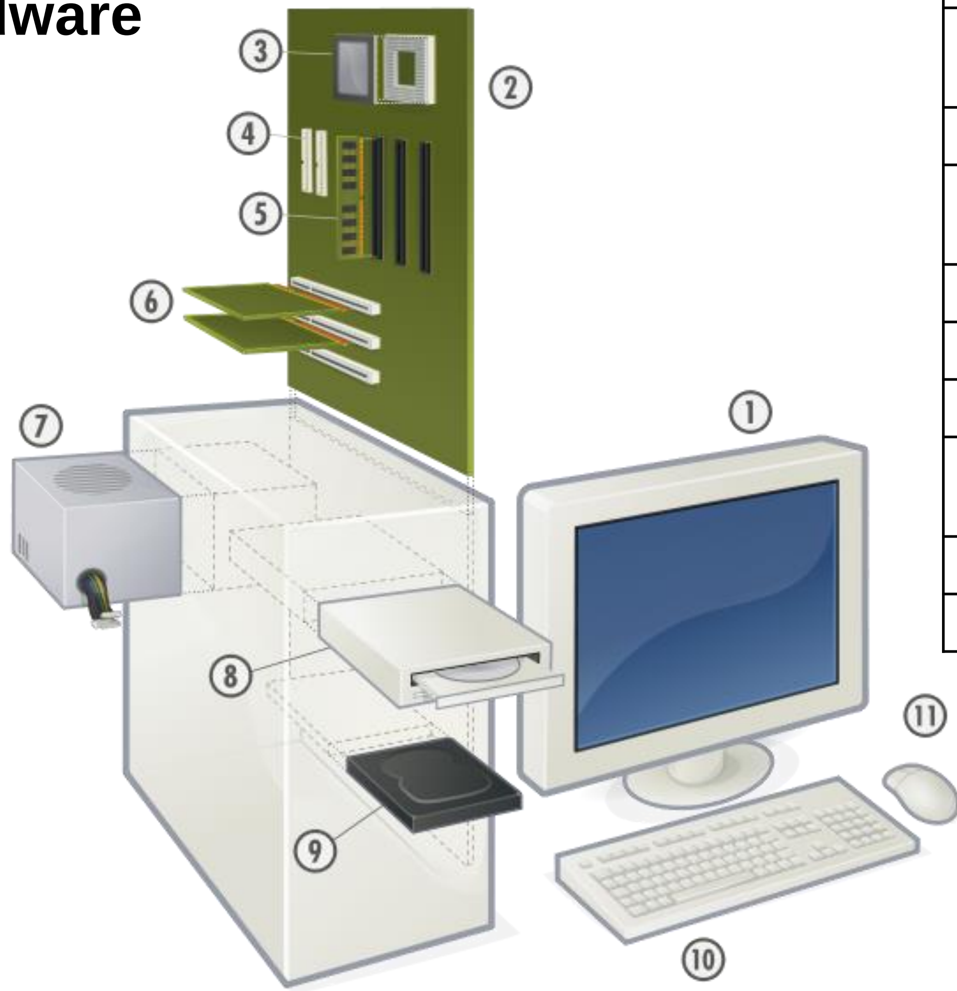
PC is incremented. Instruction according to the Op-code (0101 = 5) says to add the contents of AC (0 0 0 3) with the given address (9 4 1) contents (0 0 0 2).

PC is incremented. Next instructions Op-code (0010 = 2) says store the contents of AC to location (9 4 1).

Table of Contents

- **PC Hardware**
  - **Motherboard**
  - **Central Processor Unit (CPU)**
  - **Memory (RAM, ROM, Cache)**
  - **Chipset**
    - Northbridge
    - Southbridge
  - **BIOS**
  - **Secondary Storage Devices**
    - Hard Disk
    - Laser Disks
  - **Video and Audio Standards**
  - **System Clock**
- **Non Von Neumann Architectures**
  - **Synchronous processing**
  - **Pipelining**
  - **Shared-memory configuration**

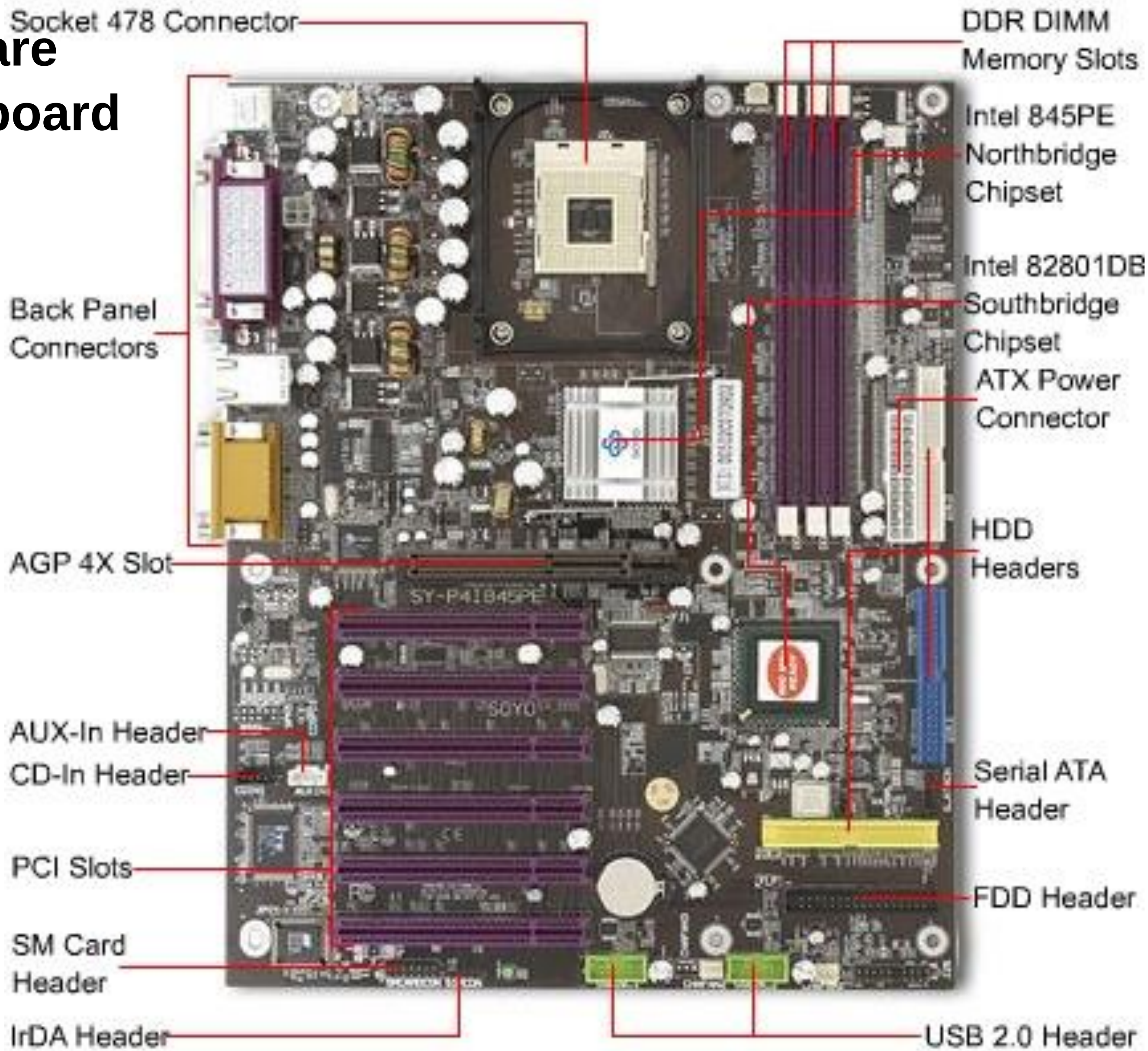
- **PC Hardware**
  - **PC Hardware**



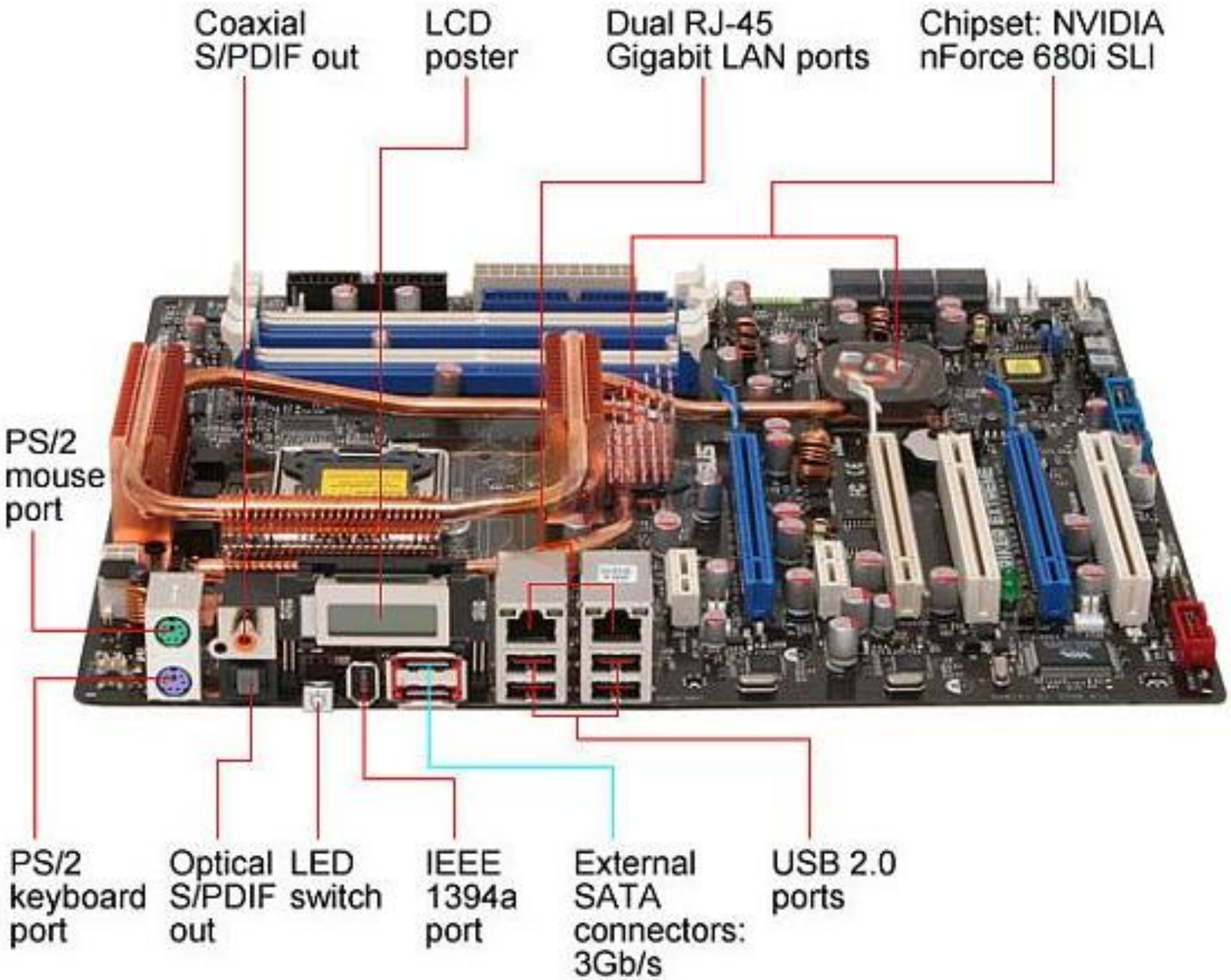
1	Monitor
2	Motherboard
3	CPU (Microprocessor)
4	ATA sockets
5	Main memory (RAM)
6	Computer Bus
7	Power supply unit
8	Optical disc drive
9	Hard disk drive (HDD)
10	Keyboard
11	Mouse

- **PC Hardware**
  - **Motherboard**
  - The motherboard is a *Printed Circuit Board* (PCB), which houses a number of the system components and hardware devices.
  - Electrical conductor lines or wires, link these to facilitate data communication between them. A bus comprises a group of these lines and the entire outline of the design is called the backplane or system bus.
  - **System components** are mounted on the system board and are connected to each other by conductor lines while **hardware devices** (like Hard Drive, CD-Rom drive) are laid on storage bays within the computer unit and are connected by wires through ports mounted on the system board and the hardware device itself.

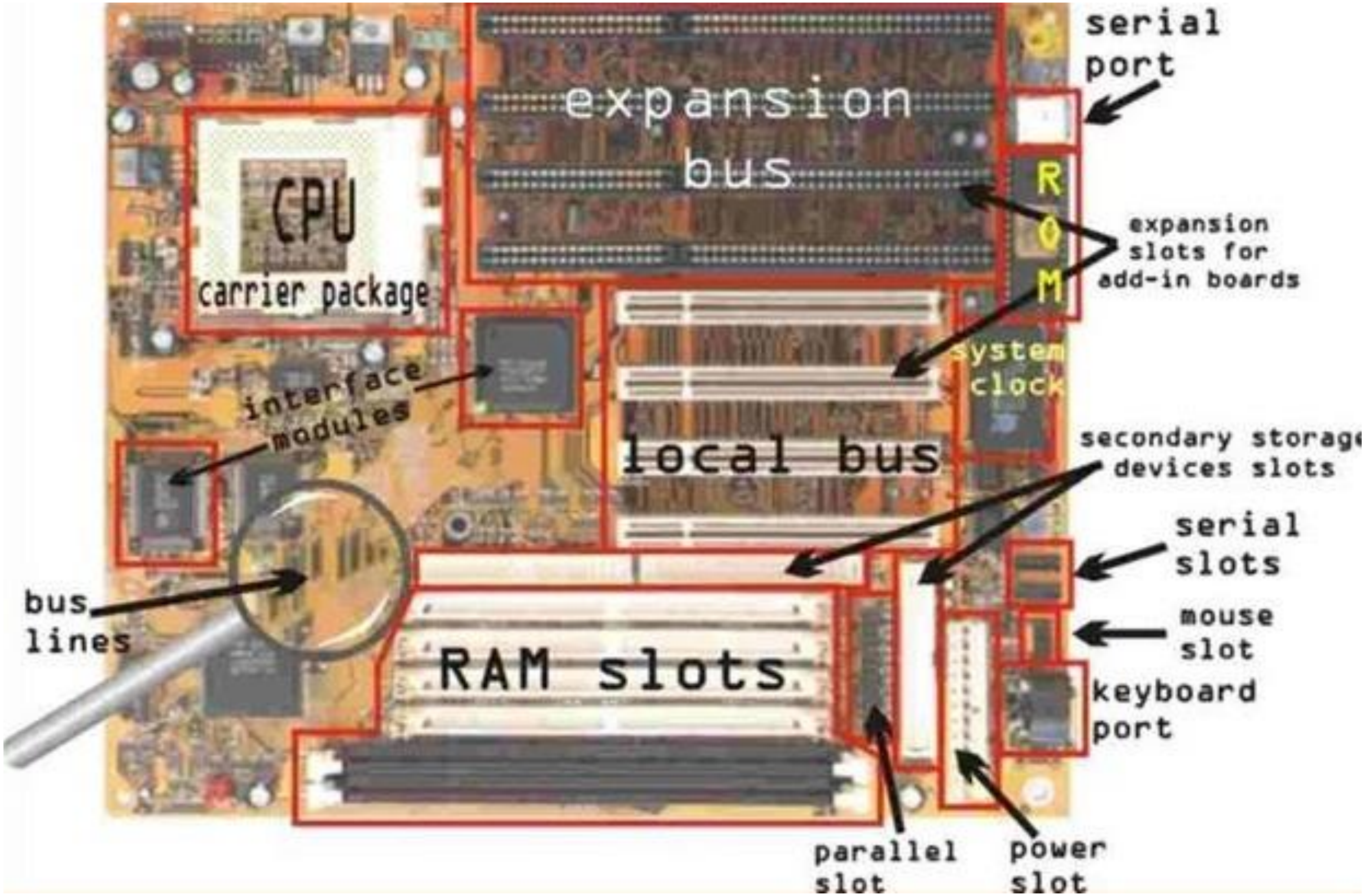
- **PC Hardware**
  - **Motherboard**



- **PC Hardware**
  - **Motherboard**

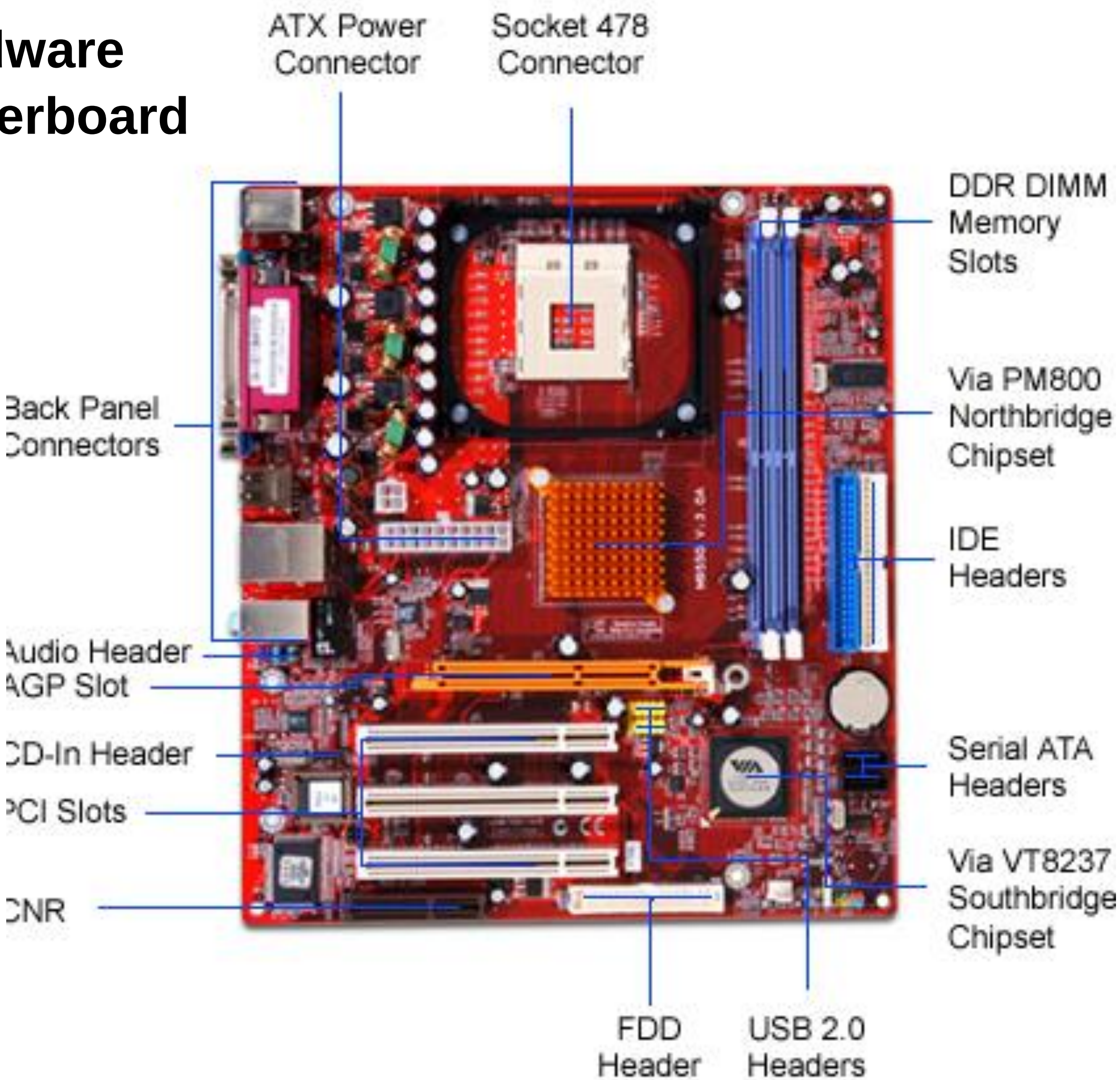


- **PC Hardware**
  - **Motherboard**



Introduction

- **PC Hardware**
  - **Motherboard**

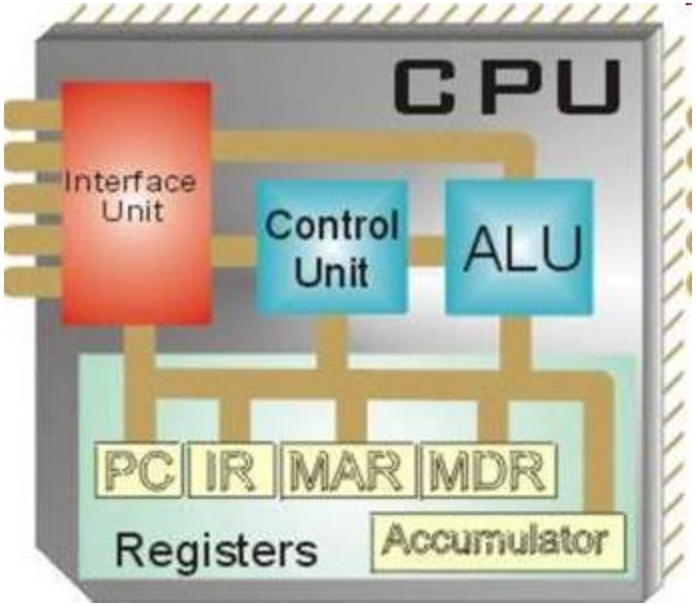


Introduction

- **PC Hardware**
  - **Motherboard**
  - Components directly attached to the motherboard include:
    - The ***Central Processing Unit*** **(CPU)** performs most of the calculations which enable a computer to work.
    - The **chipset** mediates communication between the CPU and the other components of the system, including main memory. The chipset is part of the motherboard's logic system and is usually made of two parts:
      - **- the northbridge**
      - **- the southbridge.**
    - **RAM Memory** stores all running processes (applications) and the current running OS.
    - The ***Basic Input/Output System*** **(BIOS)** is boot firmware, designed to be the first code run by a PC when powered on. The initial function of the BIOS is to identify, test, and initialize system devices such as the video display card, hard disk, floppy disk and other hardware.

- **PC Hardware**
  - **Motherboard**
    - **Internal Buses:**
      - connect the CPU to various internal components and to expansion cards for graphics. The **Northbridge** memory controller, for:
        - RAM
        - PCI Express (for expansion cards such as graphics and physics processors, and high-end network interfaces)
    - **External Buses:**
      - support ports for external peripherals. These ports may be controlled directly by the **Southbridge** I/O controller or based on expansion cards attached to the motherboard through the **PCI bus**.
        - USB
        - eSATA
        - SCSI

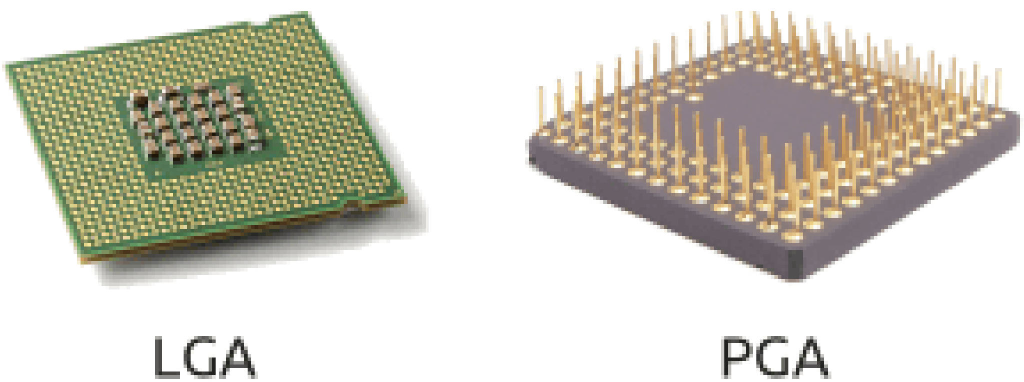
- **PC Hardware**
  - **Central Processor Unit (CPU)**
  - Each computer system has at least one CPU, which carries out instructions and performs data manipulation. It is often called a microprocessor.
  - The numerous parts within the CPU exchange data via an internal bus on a binary computational basis. All these parts are wired together and are dependant upon the CPU architecture design.



- **PC Hardware**
  - **Central Processor Unit (CPU)**
  - The CPU contains:
    - **Arithmetic Logic Unit (ALU),** which carries out arithmetical and logical functions.
    - **Control Unit (CU),** controls all operations within the CPU and arranges the data communication paths between the CPU and the components of the system unit.
    - **Interface Unit (IU)** transmits and receives program instructions and data to other computer components.
    - **Registers** temporary storage (memory cells within the CPU) for program instructions and data, however they are under the direct control of the CPU (direct addressing):
      - **Memory Address Register (MAR)**
      - **Memory Data Register (MDR)**
      - **Instruction Register (IR)**
      - **Program Counter (PC)**
      - **Accumulator**

- **PC Hardware**
  - **Central Processor Unit (CPU)**
  - Each register is designed for a special purpose:
    - **Memory Address Register (MAR):** contain the address of instructions or data being accessed in main memory **(indirect addressing)**
    - **Memory Data Register (MDR):** contain the actual instruction or data, fetched or to be stored in main memory:
      - **Instruction Register (IR):** holds the current instruction to be executed.
      - **Program Counter (PC):** a register that holds the address of the next instruction to be executed and when this instruction is fetched (by IR) it increases automatically to point (hold) the next address of the instruction to be executed
    - **Accumulator:** collects intermediate results from the ALU in order to store and manipulate these intermediate results to obtain the final desired result.

- **PC Hardware**
  - **Central Processor Unit (CPU)**
  - **CPU Sockets**
  - **Pin Grid Array (PGA):** CPUs feature a set of pins that have to perfectly fit the socket. As microprocessors advance, they also need more and more pins, both to handle new features and to provide more and more power to the chip. Socket arrangements are often named for the number of pins in the PGA.
  - **LGA (Land Grid Array):** LGA is different from PGA in that the pins are actually part of the socket, not the CPU. LGA has the pins on the motherboard, with contact pads on the processor itself, hence the name Land Grid Array



https://ezpcbuilding.com/how-to-choose-a-cpu/

- **PC Hardware**
  - **Central Processor Unit (CPU)**

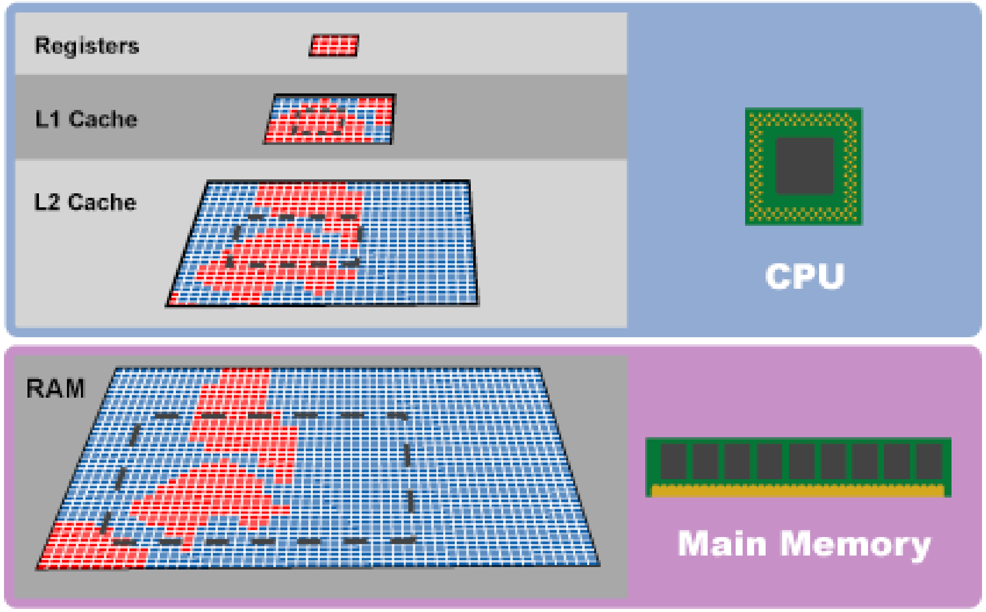
Year	Intel name	Real Name
1976	8086	8086
1982	80186	186
1982	80286	286
1985	80386	386
1989	80486	486
1993	Pentium	586
1995	P6 (Pentium Pro/II/III)	686
2000	Netburst (Pentium 4)	786
2006	Core (Core 2 Duo)	886
2008	Nehalem (Core i5/i7)	986
2011	Sandy Bridge (Core i5/i7 2nd gen)	1086
2013	Haswell (core i5/i6 4th gen)	1186
2015	Skylake (Core i5/i7 6th gen)	1286
2018?	Ice Lake	1386

https://twitter.com/mega1024kb/status/978913246022320128

- **PC Hardware**
  - **Memory - RAM**
  - RAM stands for *Random Access Memory*.
  - RAM is memory in which each cell (usually byte) can be directly accessed.
  - It is a <u>volatile memory</u>, so the data stored in the RAM will NOT remain when the power is turned off
  - Recall that <u>data and instructions reside in main memory and are treated alike</u>.

- **PC Hardware**
  - **Memory - ROM**
  - ROM stands for *Read Only Memory*.
  - can permanently or semi permanently hold data. As the name implies it can only read the data and it needs a special program to write data. It is a non-volatile memory, so the data stored in the ROM will remain even if the power is turned off.
  - ROM is an ideal place to put the computer's startup instruction that is, the software that boots the system.
  - There are 3 types of ROM:
    - **PROM:** it can be programmed once and never erased.
    - **EPROM (Erasable PROM):** it is a PROM that is erasable by the exposure of the intense UV light.
    - **EEPROM (Electrically Erasable PROM):** it can be erased from electrical signals instead of U.V. light. EEPROM is also called flash ROMs, and it is possible to erase and reprogram the ROM in a computer with removing the chip from the system.

- **PC Hardware**
  - **Memory – Cache**
  - Cache stores the most recent data the CPU has accessed from RAM and is also volatile. When the CPU seeks data from RAM, it first searches level-1 and then level-2 and if the desired data are not found the search moves to the slower RAM:
  - **Cache Level-2:** faster and more expensive than RAM, it increases the performance of data communication between CPU and RAM.
  - **Cache Level-1:** lies within the CPU and is faster but smaller than level-2.



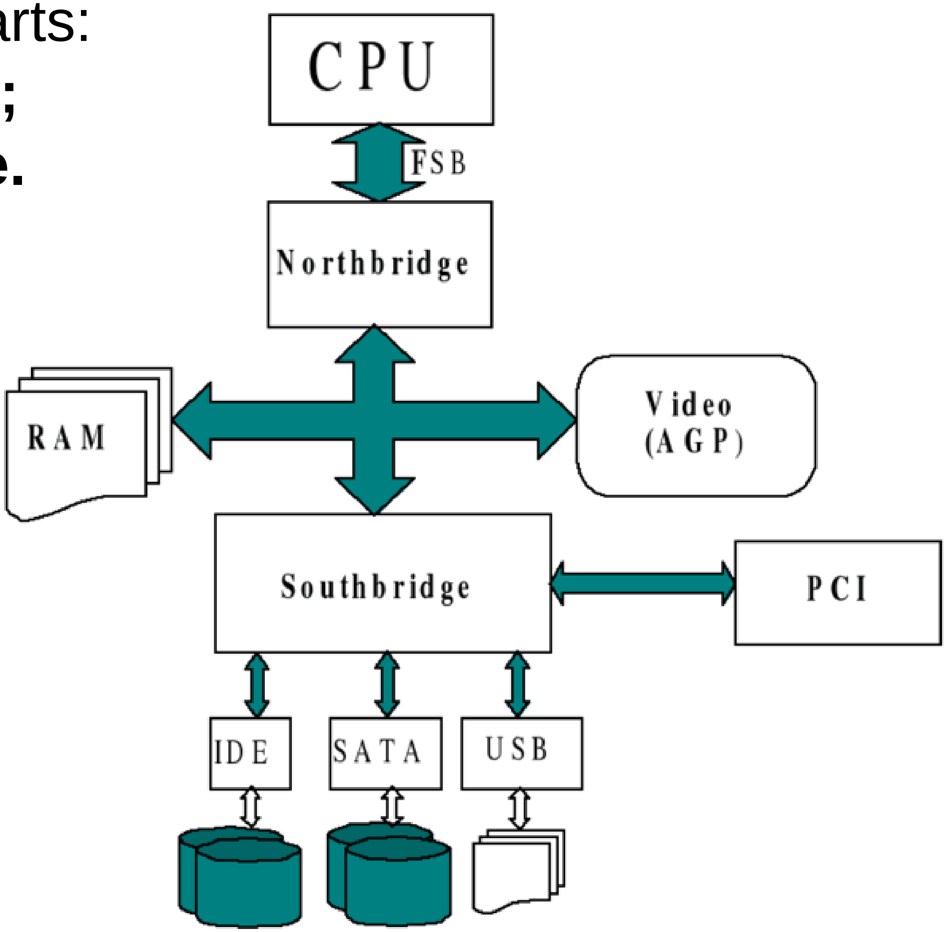
- **PC Hardware**
  - **Memory – Cache**

Level	Access Time	Typical Size	Technology	Managed By
Registers	1-3 ns	1 KB	Custom CMOS	Compiler
Level 1 Cache (on-chip)	2-8 ns	8 KB-128 KB	SRAM	Hardware
Level 2 Cache (off-chip)	5-12 ns	0.5 MB - 8 MB	SRAM	Hardware
Main Memory	10-60 ns	64 MB - 1 GB	DRAM	Operating System
Hard Disk	3,000,000 – 10,000,000 ns	20 - 100 GB	Magnetic	Operating System/ User

- **PC Hardware**
  - **Memory – Cache**

Characteristic			Scaled to Human Time	
CPU frequency	*2GHz*			
Processor Cycle Time	0.5	ns	1	s
L2 cache access	10	ns	20	s
Memory access	80	ns	160	s (2.6 mins)
Thread context switch	5000	ns (5us)	10000	s (2.7 hours)
Disk access	8000000	ns (8ms)	16000000	s (185 days)
Process quantum	100000000	ns (100ms)	200000000	s (6.3 years)
In blue ► Things improving very fast				
In orange ► Things improving to a degree				
In red ► Things not really improving				

- **PC Hardware**
  - **Chipset**
  - The chipset mediates communication between the CPU and the other components of the system, including main memory. It is usually made of two parts:
    - **the Northbridge;**
    - **the Southbridge.**



- **PC Hardware**
  - **Chipset Northbridge**
  - Northbridge is responsible for interconnecting the CPU, RAM, and all of the PC's most bandwidth-intensive I/O devices (e.g. video)
  - Segregating these high-bandwidth components from the rest of the system and giving them their own private "hub" enables system designers to keep down the cost of the bridge chip and the motherboard.
  - Thus the Northbridge can focus primarily on routing traffic among processor, memory, and bandwidth-intensive I/O components, while letting another chip (the Southbridge) focus on slower I/O traffic and miscellaneous functionalities (timing, power management, the BIOS).

- **PC Hardware**
  - **Chipset Northbridge**
  - Northbridge is helped by a series of buses (or lanes) :
    - **Memory bus** - the RAM's single lane of communication with the rest of the system. The Northbridge's memory controller manages traffic on the memory bus, and performs all that the memory accesses on behalf of the rest of the components in the system.
    - **Front-side Bus (FSB)** connects the CPU to the Northbridge; this is the only way for the CPU to communicate with the other components.
      - instructions, data, and results all travel to and from the CPU over the front-side bus. Because the FSB plays such a critical role in enabling the processor to talk to the other components on the motherboard, it is very important to implement a generous bandwidth for the FSB.

Introduction

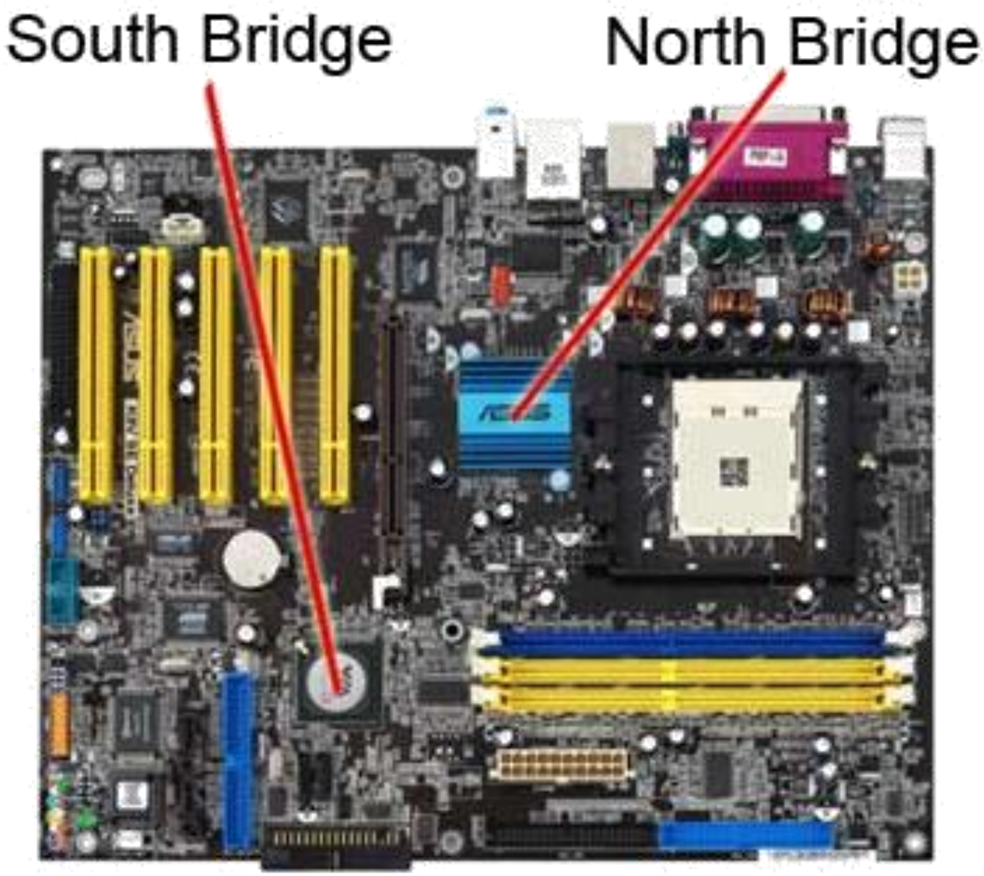
- **PC Hardware**
  - **Chipset Southbridge**
  - Present-day Southbridge's have to manage a series of peripheral controllers:
    - **PCI controller** - the Southbridge sits on the PCI bus and needs a PCI interface so that it can "talk" to the Northbridge and to the other devices on the PCI bus.
    - **IDE controller** - the integrated drive electronics (IDE) bus is the standard bus for personal computer mass storage devices (CD-ROM, DVD-ROM, hard disk).
    - **USB controller** - the universal serial bus (USB) was introduced to replace a number of aging peripheral interconnect buses and formats, like ISA, PS/2 (for mouse and keyboard), the serial port, and the parallel port.
    - **X-bus interface** - the 8-bit X-bus is a legacy bus for use by the PS/2 keyboard and mouse, and the Flash ROM that contains the BIOS code.

- **PC Hardware**
  - **Chipset Southbridge**
    - **DMA controller:** direct memory access (DMA) is a catch-term for a family of protocols that allow components on the motherboard (especially hard drives) to access main memory directly, without having to use the CPU as a go-between the manage and the transfer (as was the case in the days before DMA).
    - **System timer:** the system timer generates a clock pulse for the ISA bus and an oscillating tone for the speaker. The latter is responsible for the beeps you hear on boot-up.
    - **Interrupt controllers** (AIPIC, NMI, standard IRQs): there are times when a device needs to interrupt the processor and tell it to drop what it's doing and address some new input or development. For instance, there has to be some way of letting the processor know that the mouse has just moved, so that the code that handles mouse input can be loaded and run.

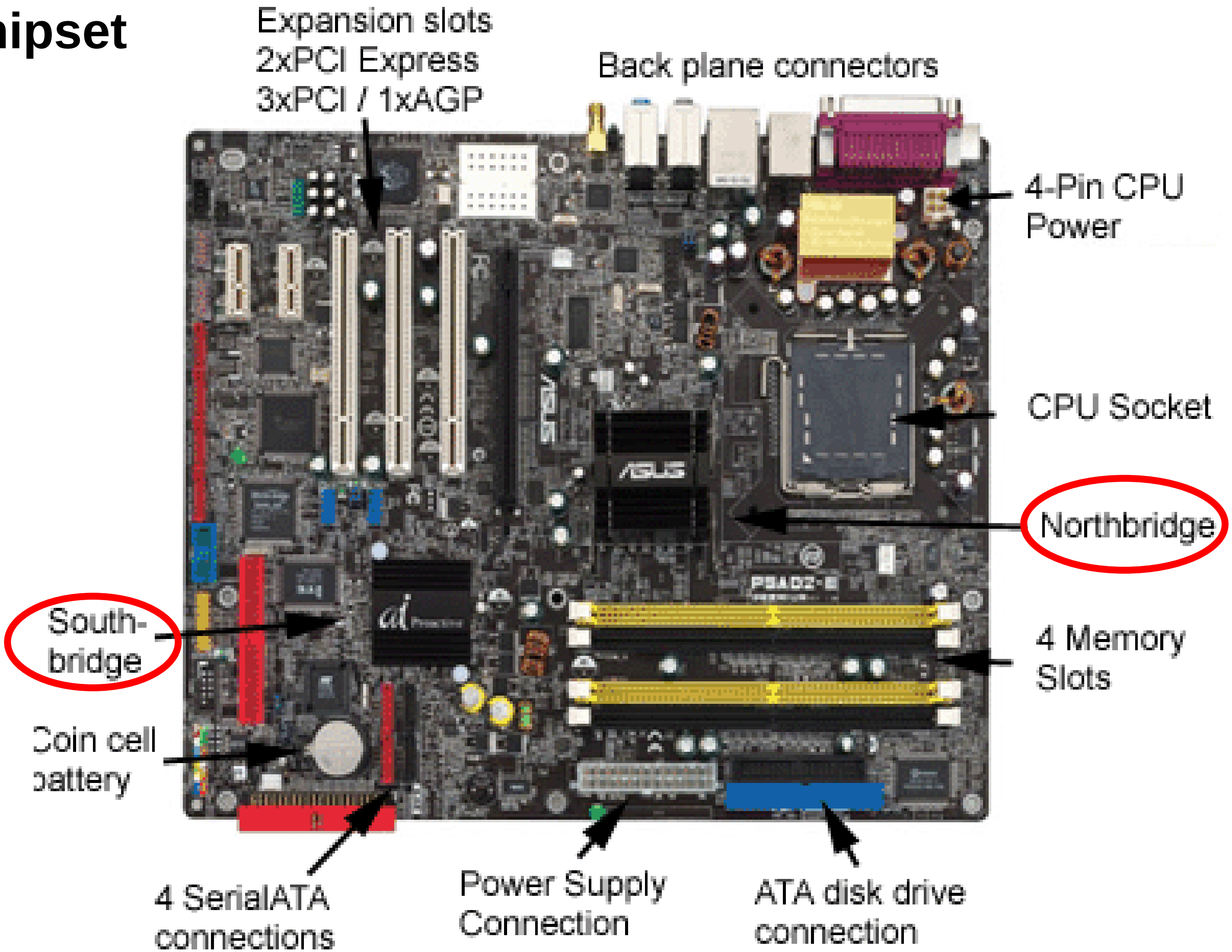
- **PC Hardware**
  - **Chipset Southbridge**
    - Last but not least, the Southbridge has to deal with the nonvolatile **BIOS memory and the APM and ACPI power management** features that keep a motherboard in optimal condition.

- **PC Hardware**
  - **Chipset Northbridge versus Southbridge**
    - The Southbridge is natively slower than the Northbridge, and information from the CPU has to go through the Northbridge before reaching the Southbridge.
      - All of the mission critical jobs are taken by the **Northbridge**
      - the system designers tend to load the Southbridge with a ton of miscellaneous, lower-priority tasks. Whenever system designers want to integrate a new component into the core logic chipset (e.g networking, RAID, sound, etc.), the **Southbridge** should be the first choice.

- **PC Hardware**
  - **Chipset Northbridge versus Southbridge**



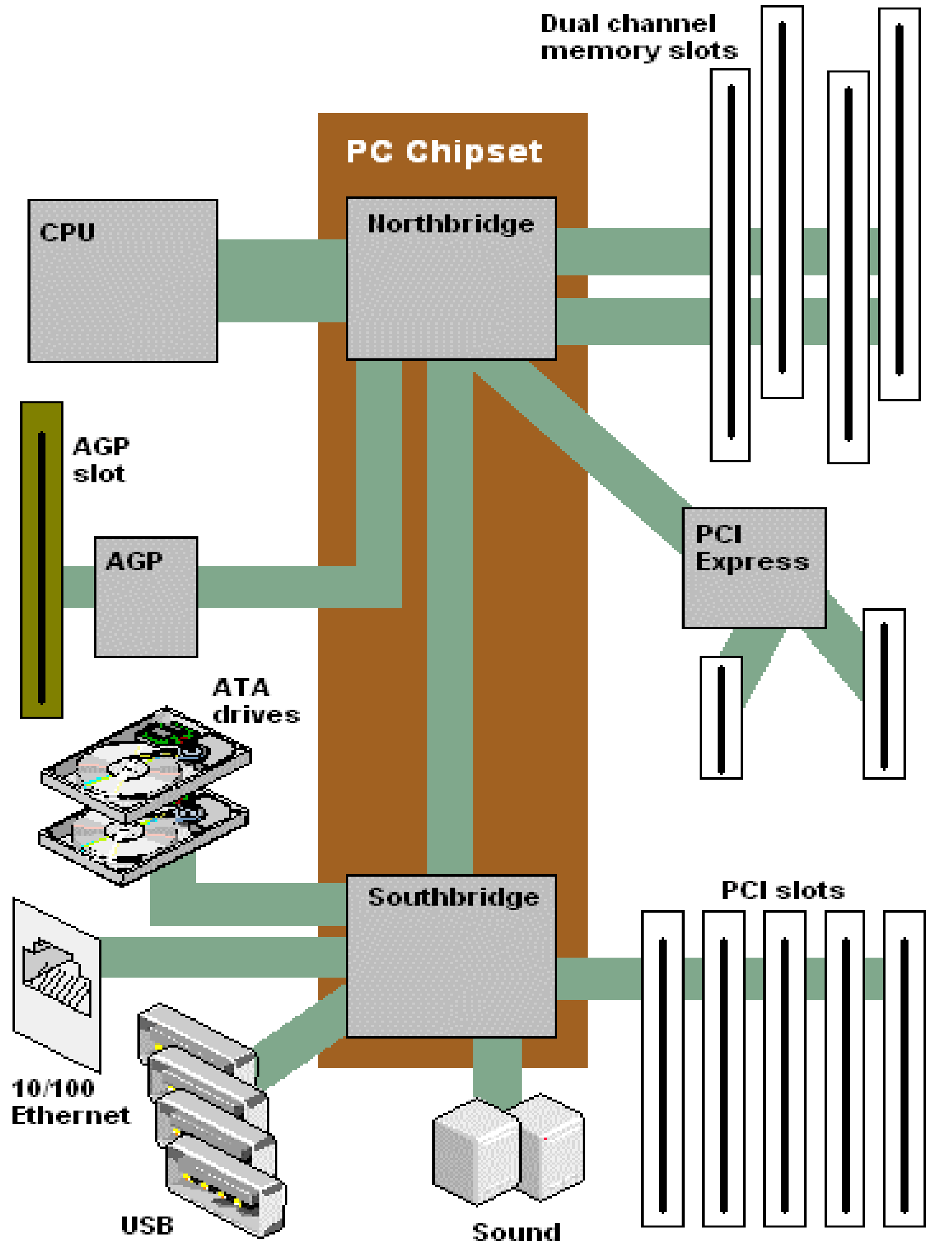
- **PC Hardware**
  - **Chipset**



ASUS P5AD2-E Premium Motherboard

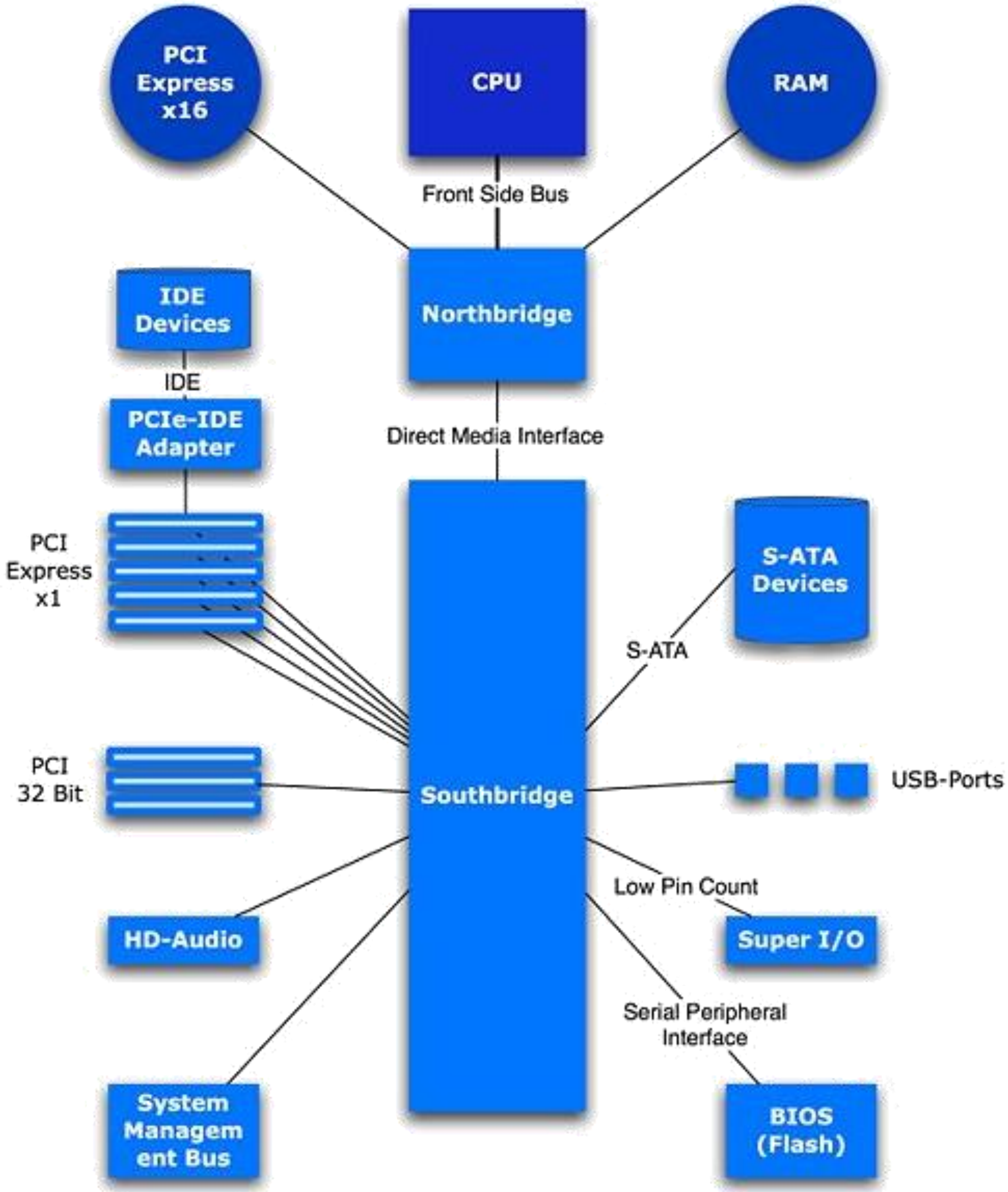
Introduction

- **PC Hardware**
  - **Chipset**

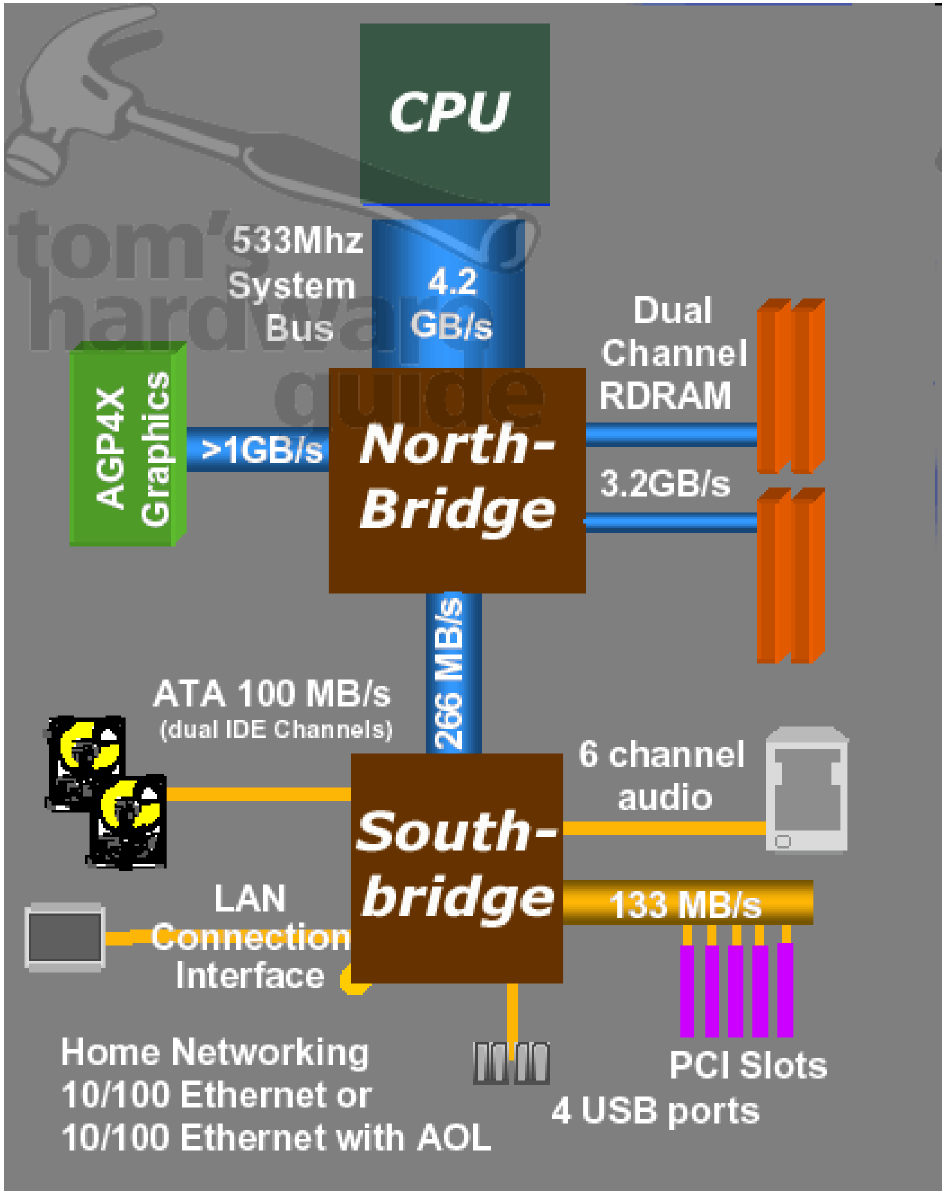


Introduction

- **PC Hardware**
  - **Chipset**

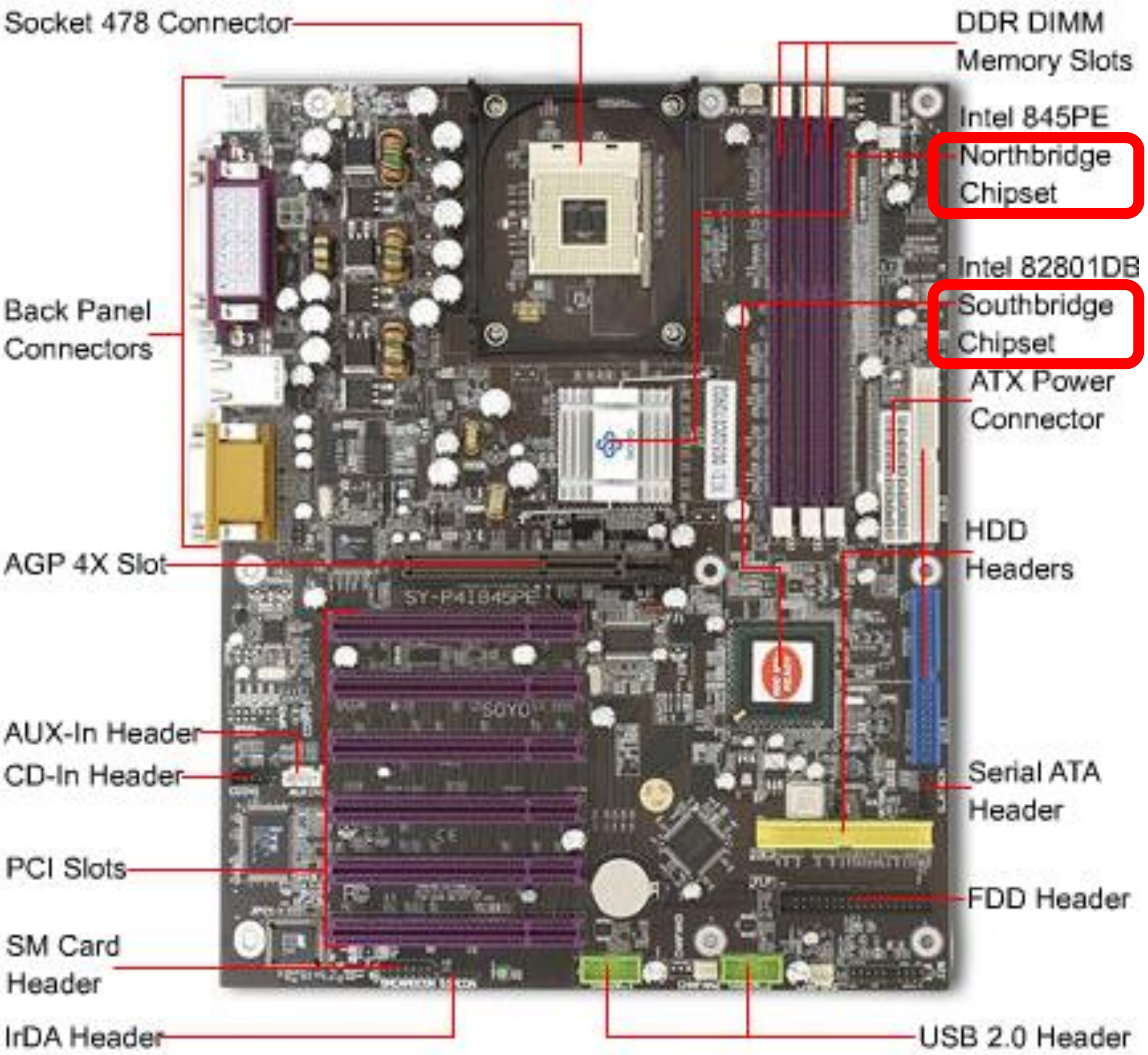


- **PC Hardware**
  - **Chipset**

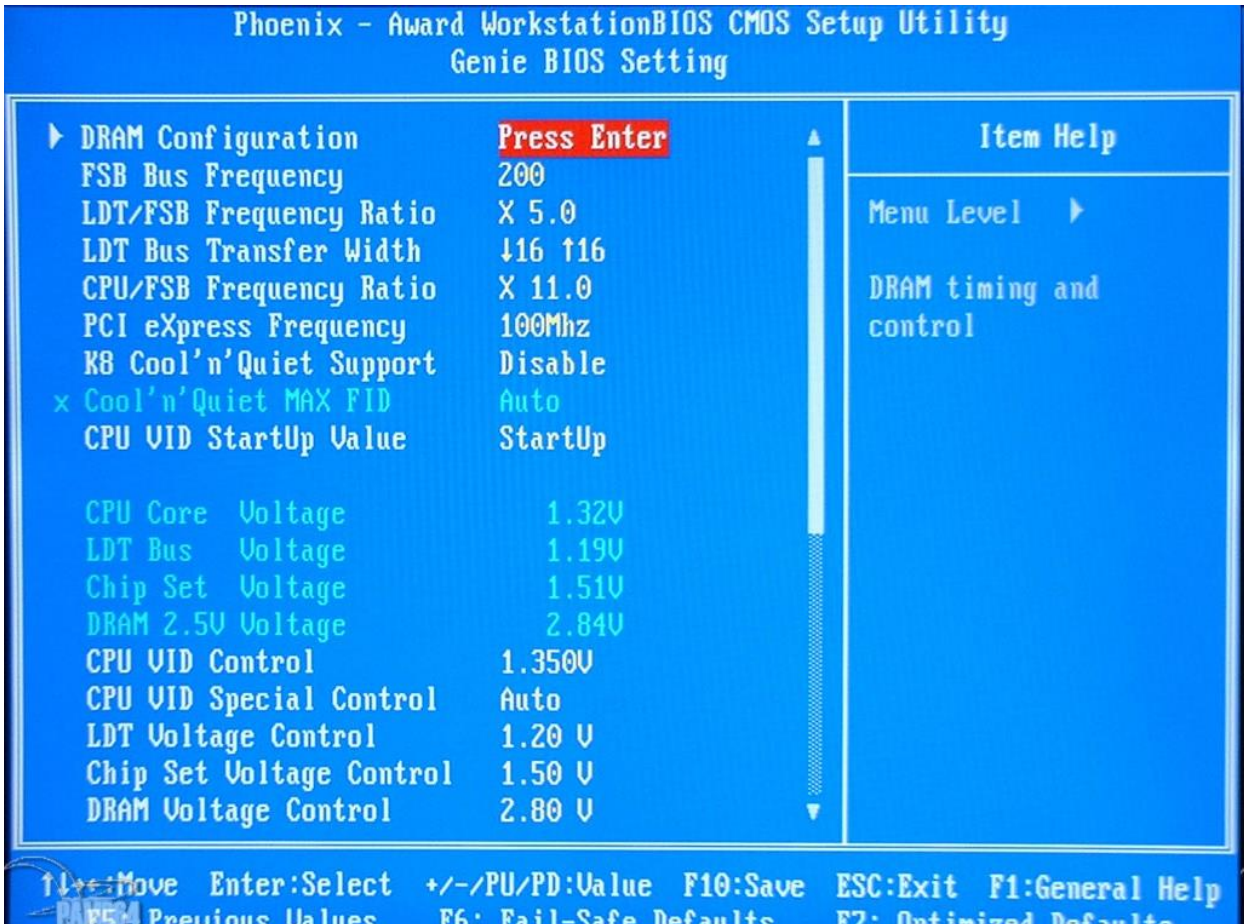


Introduction

- **PC Hardware**

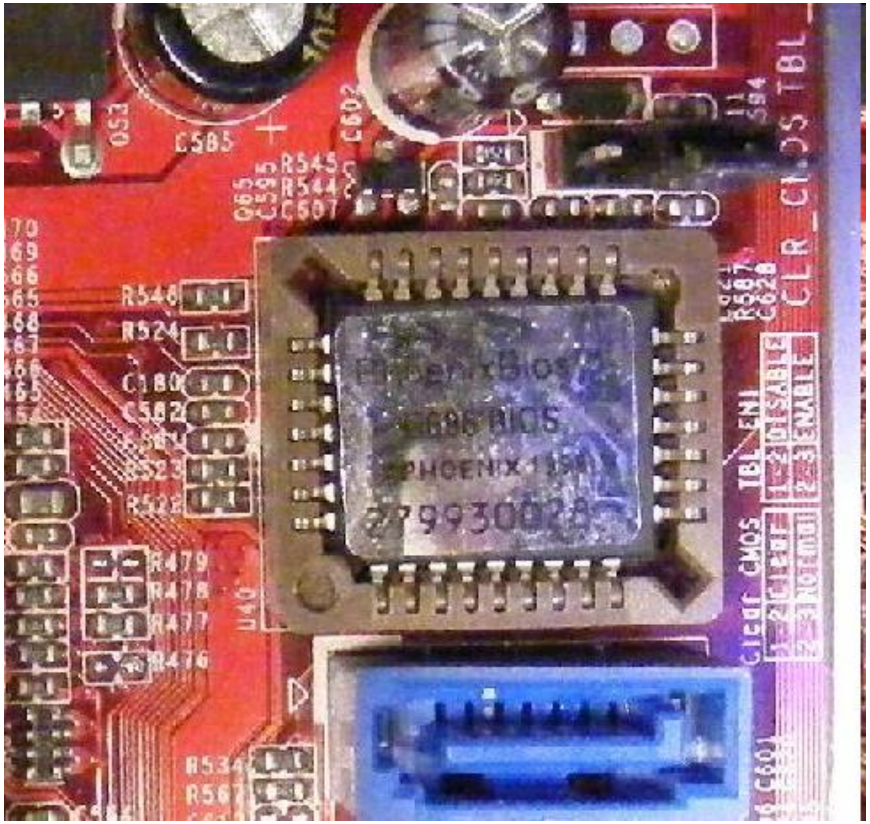


- **PC Hardware**
  - **BIOS**
  - The BIOS is boot firmware, designed to be the first code run by a PC when powered on.



- **PC Hardware**
  - **BIOS**
  - The initial function of the BIOS is to identify, test, and initialize system devices such as the video display card, hard disk and other hardware.
  - The BIOS prepares the machine for a known state, so that software stored on compatible media can be loaded, executed, and given control of the PC. This process is known as **booting**, or booting up, which is short for bootstrapping.
  - BIOS firmware is most commonly stored on **PROM, EEPROM or flash** memory devices.
  - **Flashing the BIOS**
    - flashing the BIOS is overwriting the BIOS contents with a BIOS image. This is done to update the BIOS to a newer version either to fix specific bugs, support newer hardware, or fix a damaged BIOS.

- **PC Hardware**
  - **BIOS**



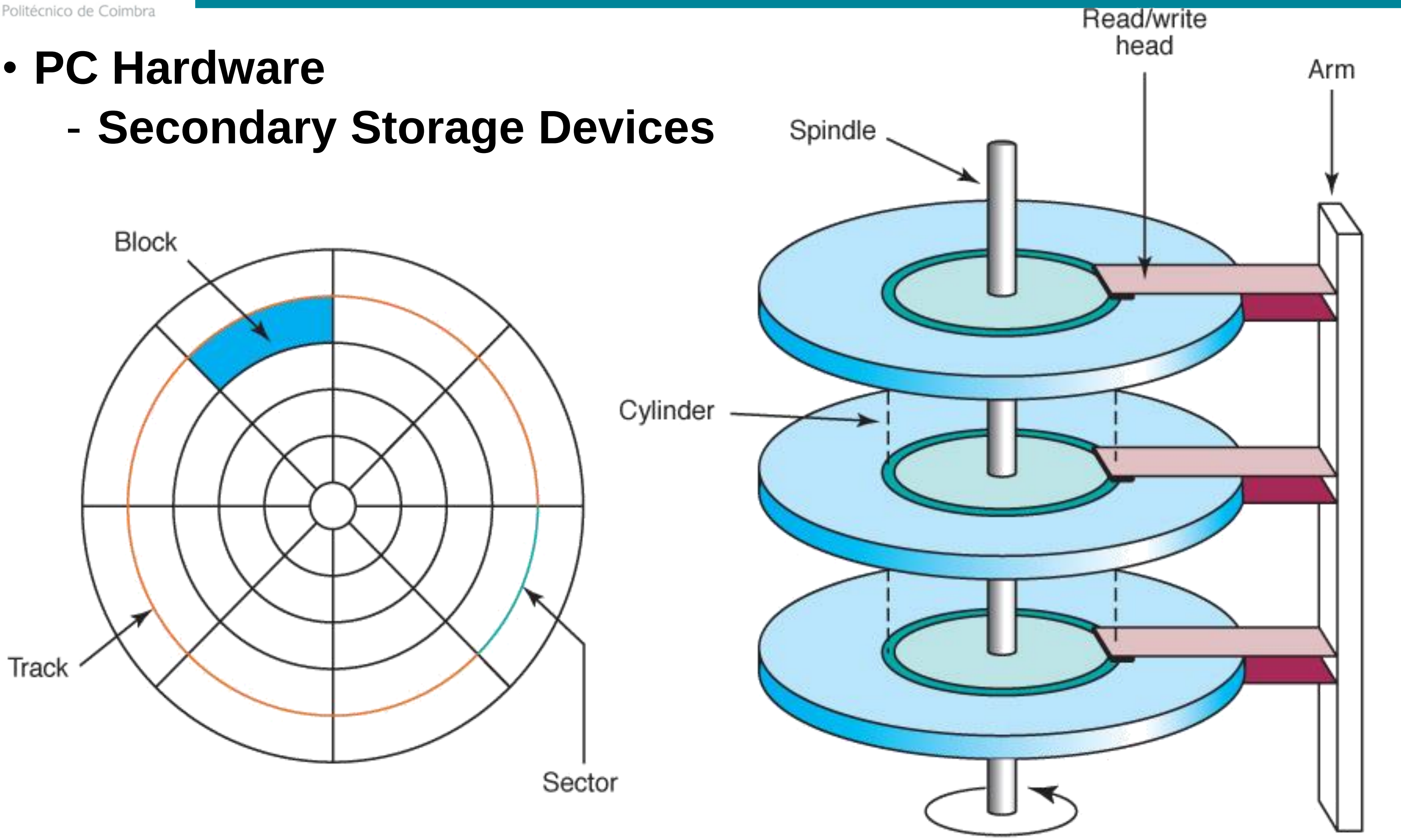
Close up of Phoenix Award BIOS, clear CMOS table, and jumper

- **PC Hardware**
  - **BIOS - Firmware**
  - Firmware is a term sometimes used to denote the fixed, usually rather small, programs that internally control various electronic devices.
    - Typical examples range from end-user products such as remote controls or calculators, through computer parts and devices like hard disks, keyboards or memory cards, all the way to scientific instrumentation and industrial robotics.
    - Also more complex consumer devices, such as mobile phones, digital cameras, etc., contain firmware to enable the device's basic operation as well as implementing higher level functions.
  - Firmware is typically involved with very basic low-level operations in a device, without which the device would be completely non-functional.
  - Simple firmware typically reside in ROM or OTP/PROM, while more complex firmware often employ flash memory to allow for updates.

- **PC Hardware**
  - **BIOS - Firmware**
  - Firmware has evolved to mean almost any programmable content of a hardware device, not only machine code for a processor, but also configurations and data for application-specific integrated circuits (ASICs), programmable logic devices, etc.
  - **Examples:**
    - Timing and control systems for washing machines
    - The BIOS found in IBM-compatible Personal Computers;
    - The platform code found on Itanium systems, Intel-based Mac OS X machines, and many Intel desktop boards is EFI compliant firmware;

- **PC Hardware**
  - **Secondary Storage Devices**
  - **Hard Disk**
  - The surface of each disk is logically organized into:
    - **tracks**: concentric circles around the surface of the disk.
    - **sectors**: each track is divided into sectors. Each sector holds a block of information as a continuous sequence of bits. Although the tracks nearer the center look smaller, each track has the same number of sectors, and each sector has the same number of bits. <u>The blocks of data nearer the center are just more densely packed</u>.
  - The actual number of tracks per surface and the number of sectors per track vary, but 512 bytes or 1024 bytes are common. <u>The location of the tracks and sectors are marked magnetically when a disk is formatted; they are not physically part of the disk</u>.
  - An **address in a hard drive** <u>is the cylinder number, the surface number, and the sector</u>.

- **PC Hardware**
  - **Secondary Storage Devices**



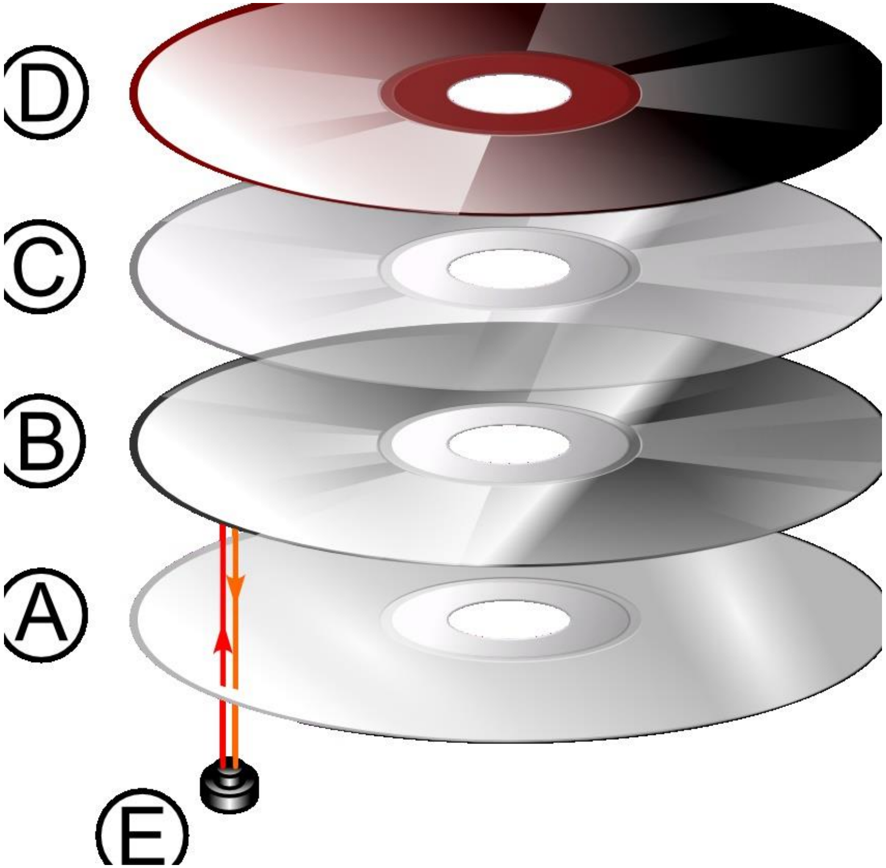
(a) A single disk (b) A hard disk drive

Hard disks, consist of several platters attached to a spindle that rotates. Each platter has its own read/write head. <u>All of the tracks that line up under one another are called a </u>**cylinder.**

- **PC Hardware**
  - **Secondary Storage Devices**
  - **Hard Disk**
  - The read/write head in a disk drive is positioned on an arm that moves from one track to another. An input/output instruction specifies the track and sector. When the read/write head is over the proper track, it waits until the appropriate sector is beneath the head and the block of information in the sector is then accessed. This process gives rise to four measures of a disk drive's efficiency:
    - **Seek time** is the time it takes for the read/write head to get positioned over the specified track.
    - **Latency** is the time it takes for the specified sector to spin to the read/write head. The average latency is one-half the time for a full rotation of the disk. For this reason, latency is also called rotation delay.
    - **Access time** is the sum of seek time and latency.
    - **Transfer rate** is the rate at which data is transferred from the disk to memory.

- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**
  - CDs, DVDs and Blu-ray have a number of physical similarities:
    - each uses a 1.2mm (4/100 inch) piece of clear <u>polycarbonate plastic with microscopic bumps</u> arranged as a single, continuous, spiral track of data.
    - optical media requires a player composed of a fast-spinning drive motor for spinning the media, a laser (infrared, red, or blue, depending on the player and media), and a tracking mechanism that moves the laser beam to follow the spiral track (with a resolution in the scale of submicrons).
    - the function of the player is to focus the laser on the track of bumps.

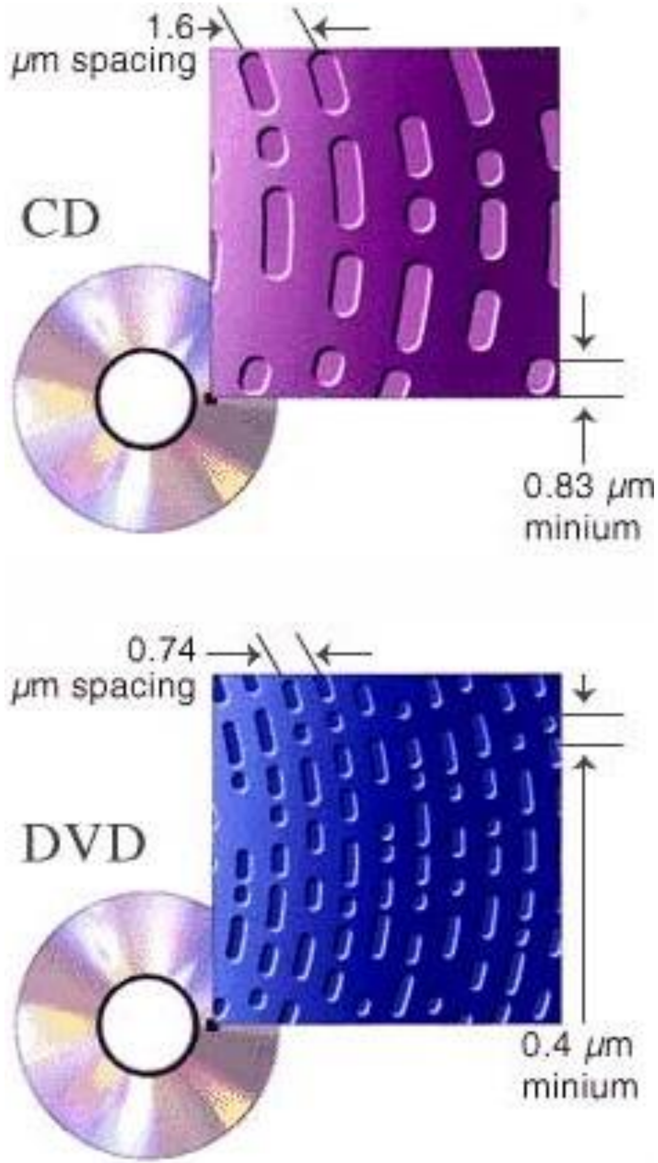
- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**



A. A polycarbonate disc layer has the data encoded by using lands and pits.
B. A reflective layer reflects the laser back.
C. A lacquer layer is used to prevent oxidation
D. Artwork is screen printed on the top of the disc.
E. A laser beam reads the polycarbonate disc, is reflected back, and read by the player.

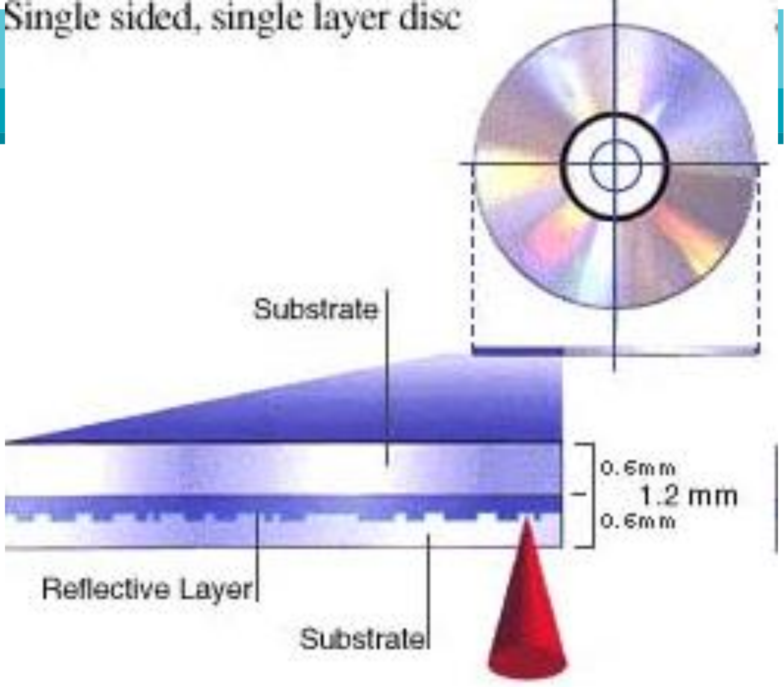
Introduction

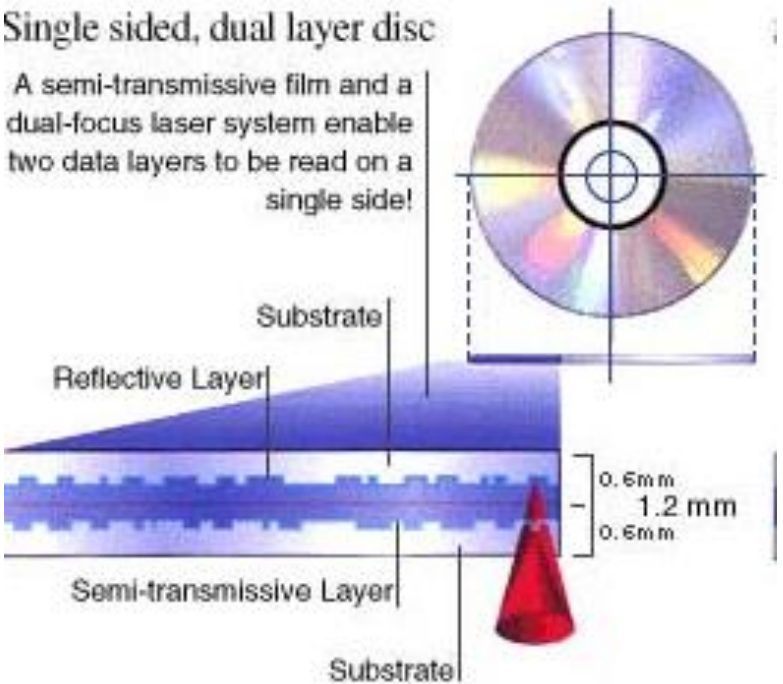
- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**
  - The main difference between the current types of optical media is the size of the data track, bumps and the wavelength of the laser used for reading the data.
    - **CD,** each track is about 1.6 microns wide and each pit has a depth of about 0.11 micron and a minimal length of about 0.834 micron.
    - **DVD** shrinks this almost by half with a 0.74 micron track and a pit length of 0.4 (interestingly, the pit depth of a DVD is a bit deeper at 0.12 micron). As we already mentioned, the 650nm wavelength (down from 780nm of a CD) allows the DVD to read this extra information.



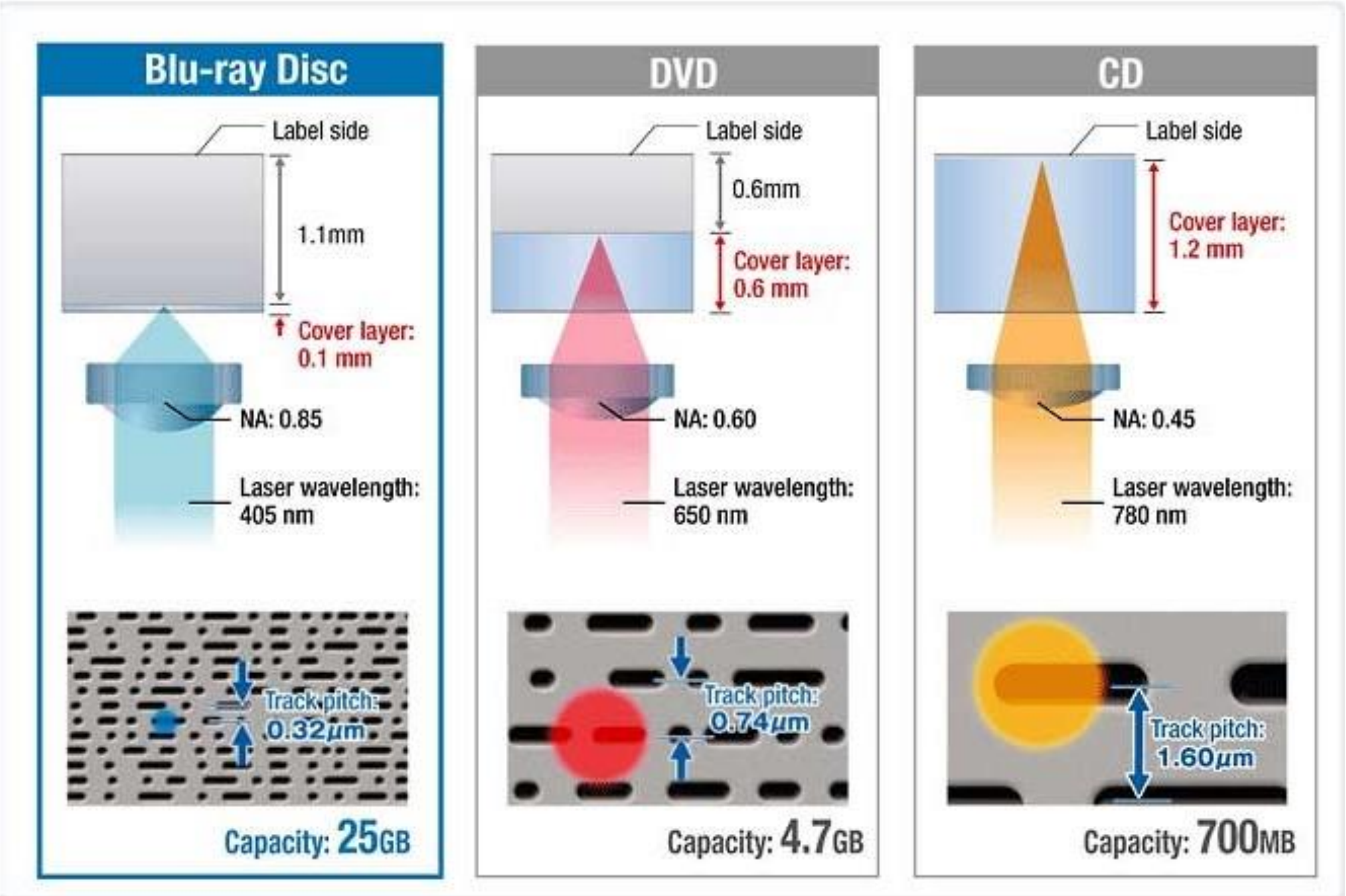
Introduction

- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**
  - <u>Another important technical change is the change in the data location inside the media.</u>
    - **CD** had only one layer located in the innermost part of the 1.2mm thick polycarbonate plastic (fairly close to the label)
    - **DVD** (and HD-DVD) the data surface is located in the middle of the media.
    - **Blu-ray** however is very different in this respect, locating its data surface on the opposite side of the label.

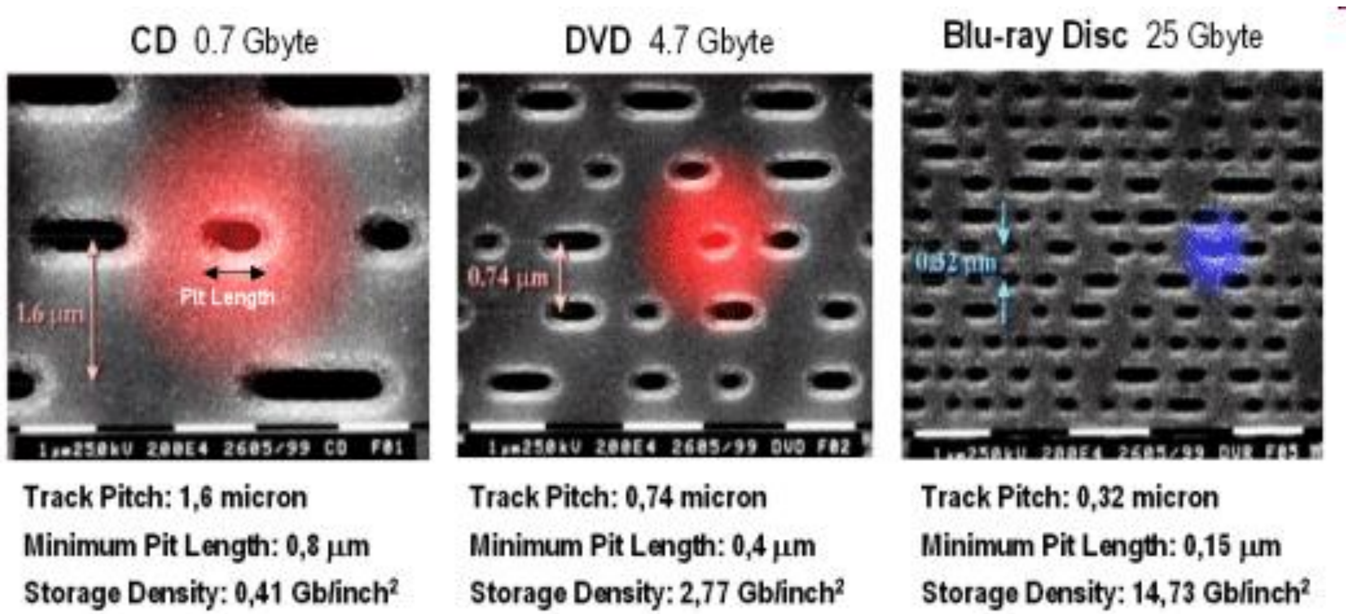




- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**



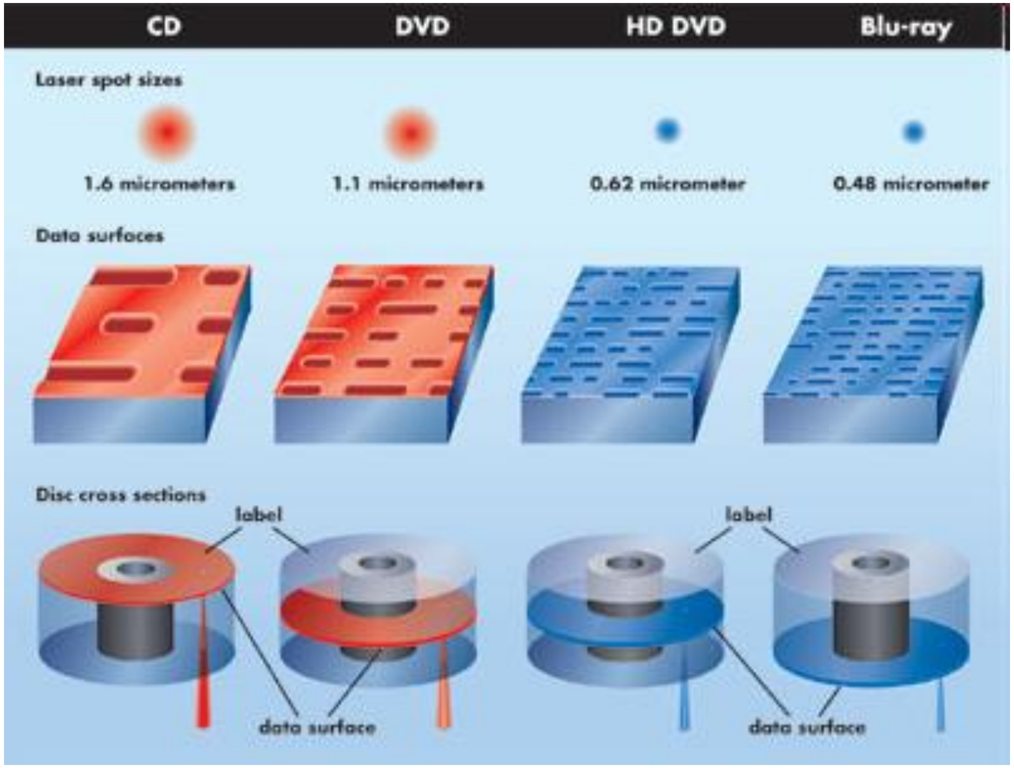
- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**

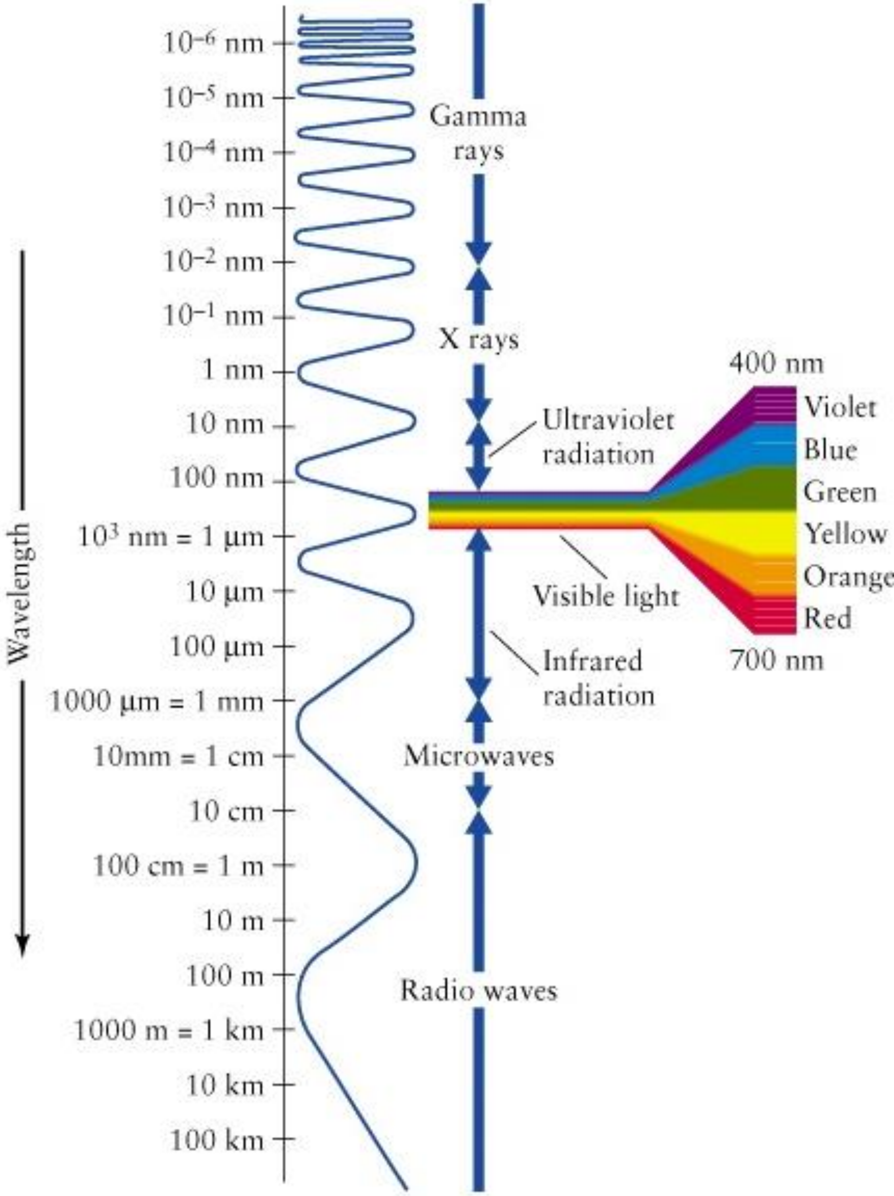


Due to the fact that the data layer on a Blu-ray Disc is placed much "closer" to the laser lens than in DVD (or even the HD-DVD proposal), there is less distortion resulting in significantly improved tolerances. Hence, more precision and ultra high storage densities are made possible.

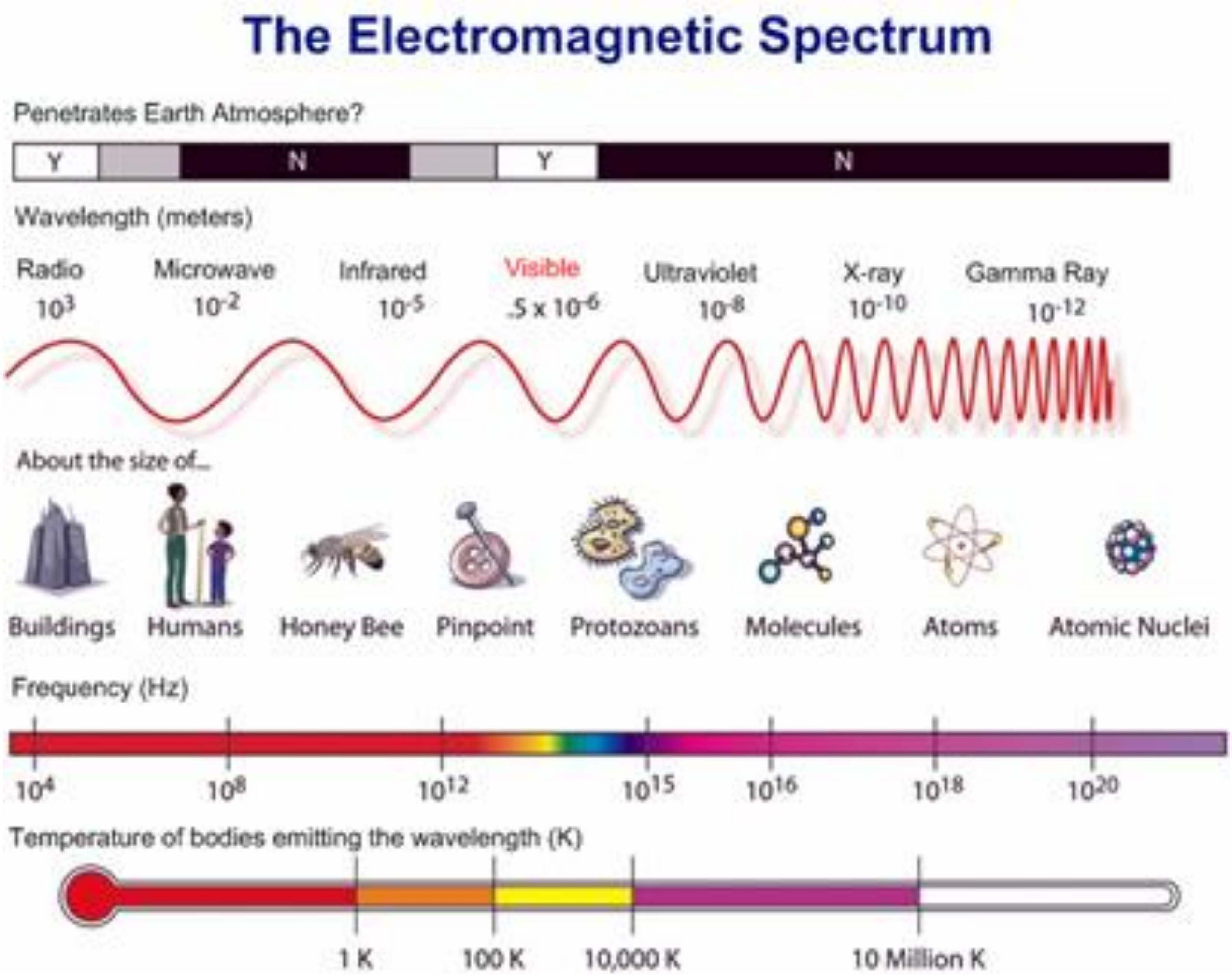
Introduction

- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**

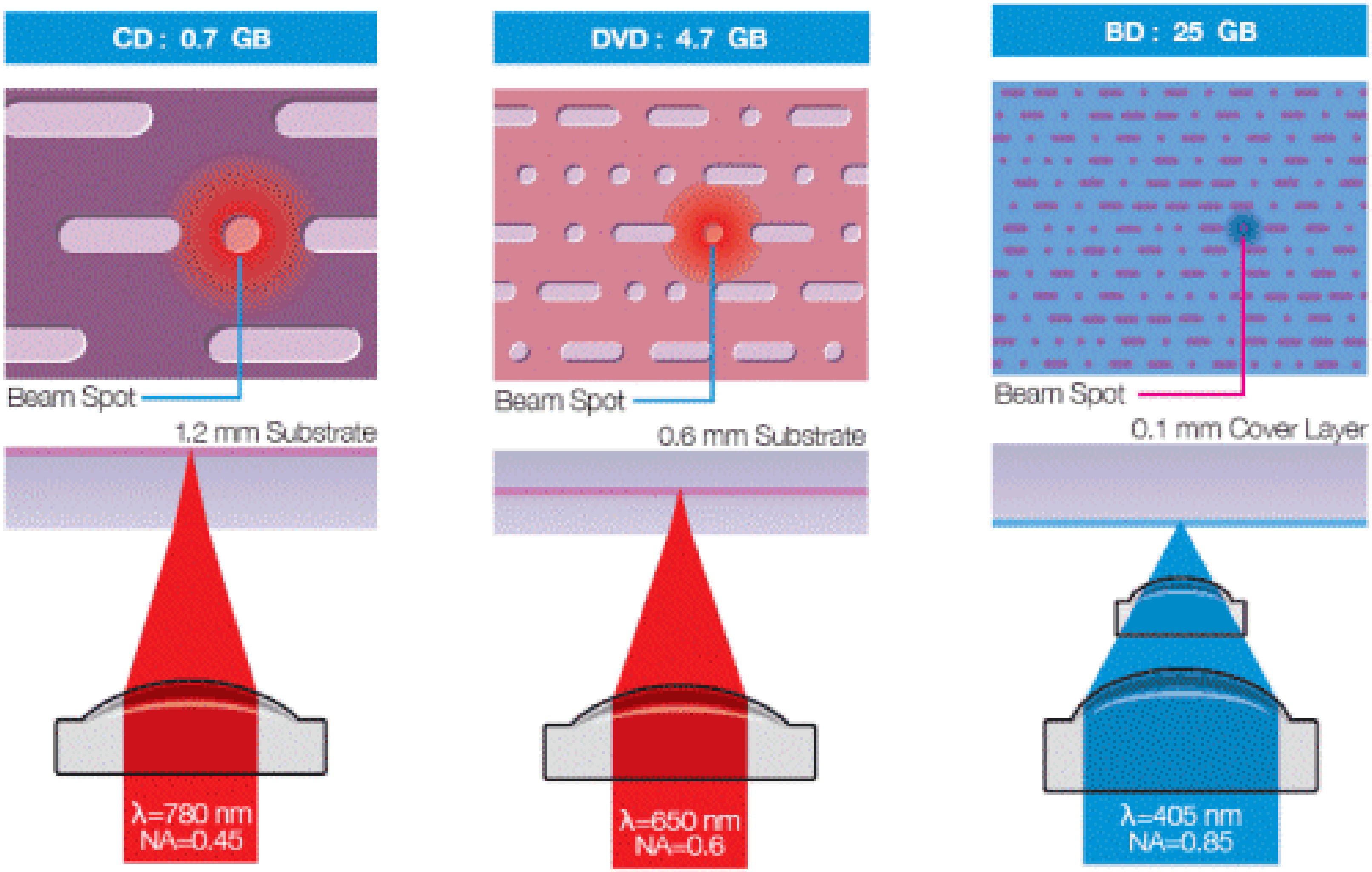




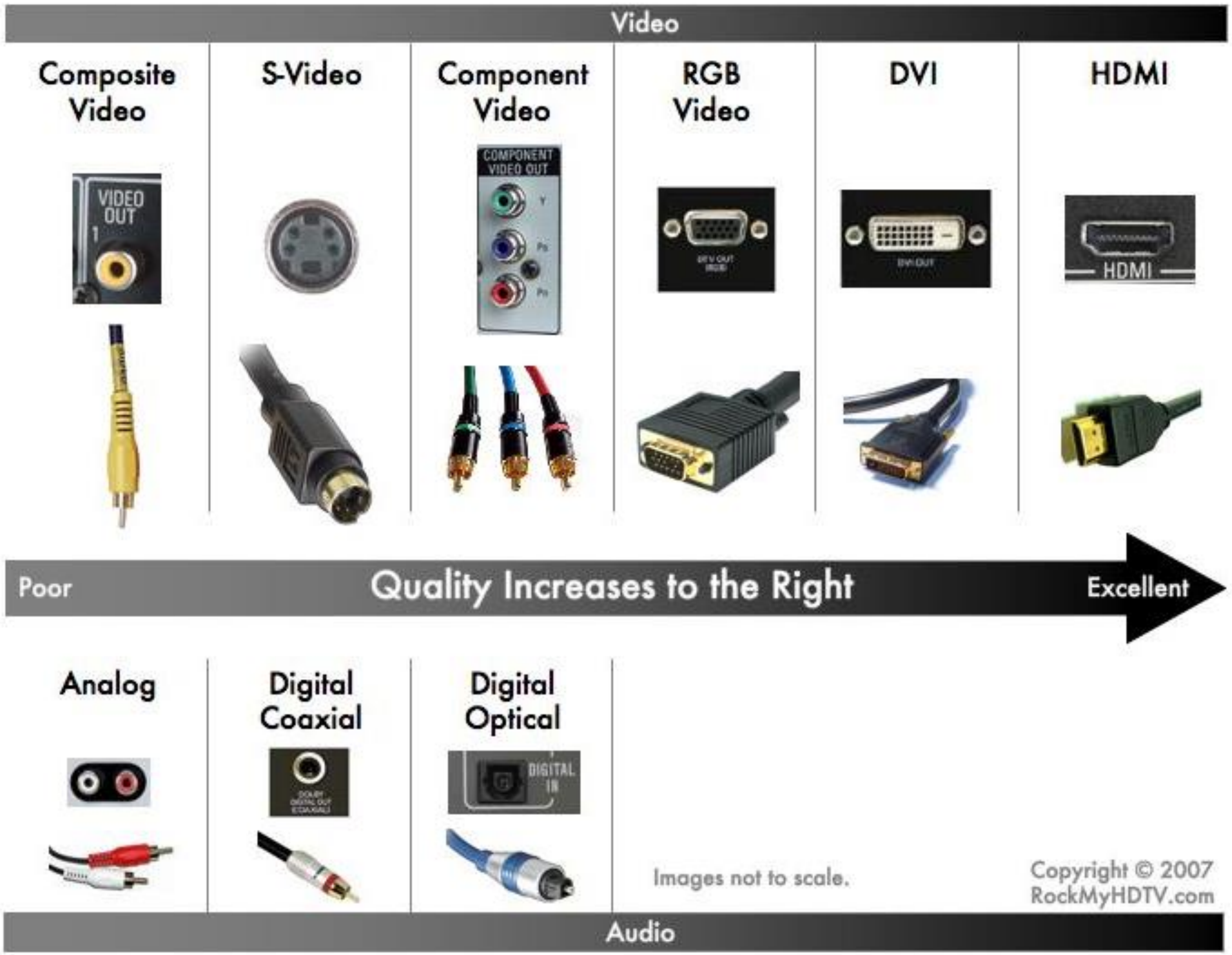
- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**



- **PC Hardware**
  - **Secondary Storage Devices**
  - **Laser Disks**



- **PC Hardware**
  - **Video and Audio Standards**



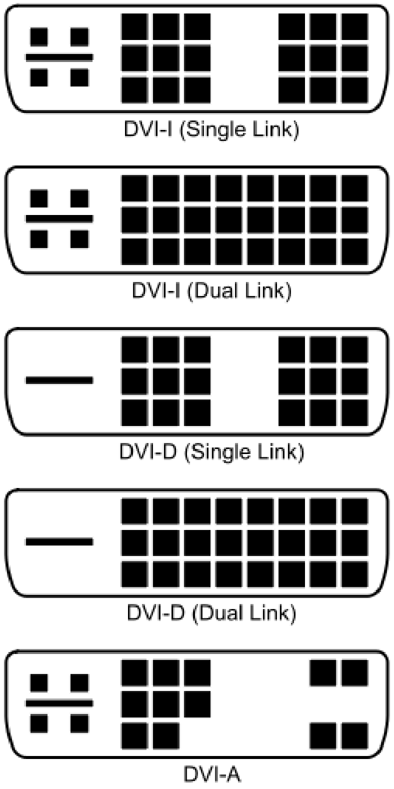
Introduction

- **PC Hardware**
  - **Video and Audio Standards**
    - **Composite Video**
      - An inferior analog video format used for low-quality analog connections between video devices, such as a VCR and an old television set. Should be avoided at all costs for connection to HDTVs due to poor image quality. Requires 1 video cable carrying the full video signal. The lowest-quality analog video format. The audio signal is carried separately.
    - **S-Video**
      - An inferior analog video format used for NTSC television recording and moderate-quality analog connections between video devices, such as a VCR and an old television set. Modestly superior to composite video, yet far inferior RGB and digital video. The audio signal is carried separately.

Introduction

- **PC Hardware**
  - **Video and Audio Standards**
    - **Component Video**
      - An analog video format typically used for connections between video devices, such as a DVD player and a HDTV.
      - Requires 3 separate video cables, or at least, 3 separate wires bundled into a single cable. Each cable carries a part, or "component," of the full video signal. These components are red ("R"), green ("G"), and blue ("B"), thus another name for this format, RGB. The highest-quality analog video format.
      - The audio signal is carried separately.

- **PC Hardware**
  - **Video and Audio Standards**
    - **Digital Visual Interface (DVI)**
      - A high-quality digital video connection format typically used between digital video devices, such as a DVD player and a HDTV.
      - The audio signal is carried separately.



- **PC Hardware**
  - **Video and Audio Standards**
    - **High-Definition Multimedia Interface (HDMI)**
      - A high-quality digital connection format typically used between digital video devices, such as a DVD player and a HDTV.
      - It includes both video and audio.

HDMI vs DVI

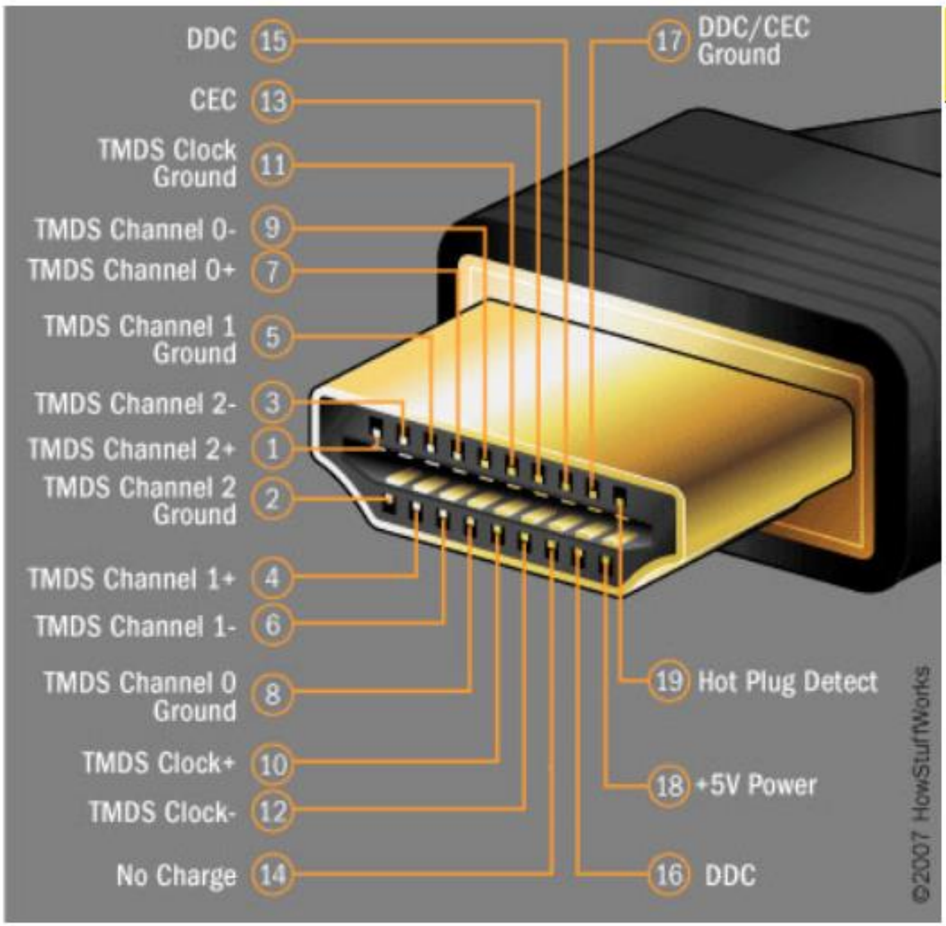
The standard Type A HDMI connector has 19 pins and a higher resolution Type B also exists but is not yet in common use.

HDMI is backwards-compatible with the single-link Digital Visual Interface (DVI) commonly found on older HD televisions and set top devices through the use of a suitable adapter or cable, but the audio and remote control features of HDMI will not be available.

Additionally, without support for HDCP, the video quality and resolution may be downgraded by the player unit.

HDMI-DVI Cable

- **PC Hardware**
  - **Video and Audio Standards**
    - **High-Definition Multimedia Interface (HDMI)**



Specifications

Hot plugable Yes

External Yes

Audio signal PCM, DVD-Audio, Super Audio CD, Dolby TrueHD, DTS-HD Master Audio

Video signal 480i, 480p, 576i, 576p, 720p, 1080i, 1080p, 1440p, etc.

Bandwidth (with HDMI v 1.3) 10.2 Gbps at 340 Mpixels/sec

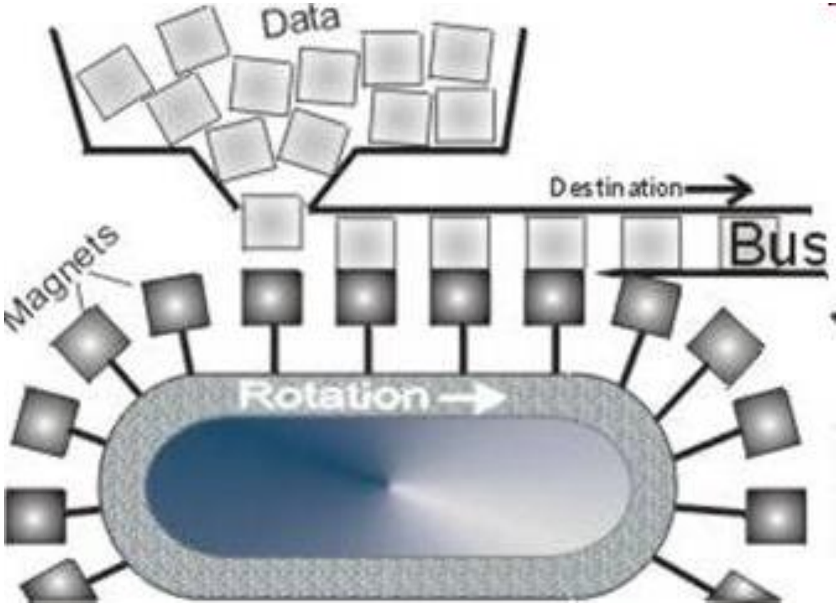
Pins 19


MALE


FEMALE

Introduction

- **PC Hardware**
  - **System Clock**
  - The System clock synchronizes component function and regulates data control by a factor of time. It is a timer operating with high frequency binary pulses (one and zero) up to millions of cycles per second. When a communication pathway has been established a data word is transmitted to other components on a particular clock pulse. The systems clock must support the internal clock speed of the CPU.

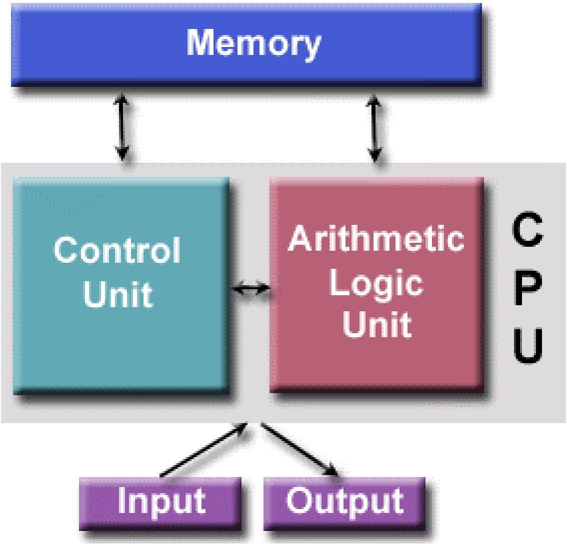


The diagram illustrates how each clock pulse acts as a magnet to which each data word is attached so that they pass through the gap in order to send the data to its destination.

ESTGOH
Escola Superior de Tecnologia e Gestão de Oliveira do Hospital
Instituto Politécnico de Coimbra

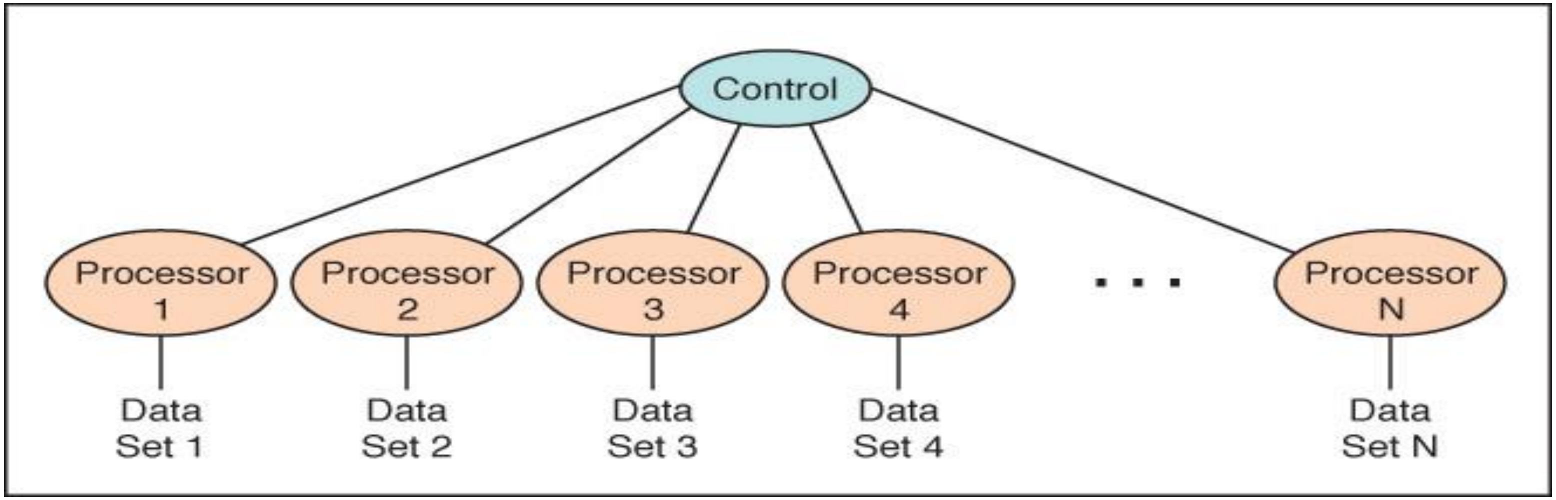
Table of Contents

- **Non-von Neumann Architectures**
  - The linear fetch-execute cycle of the von Neumann architecture still dominates the technology today.

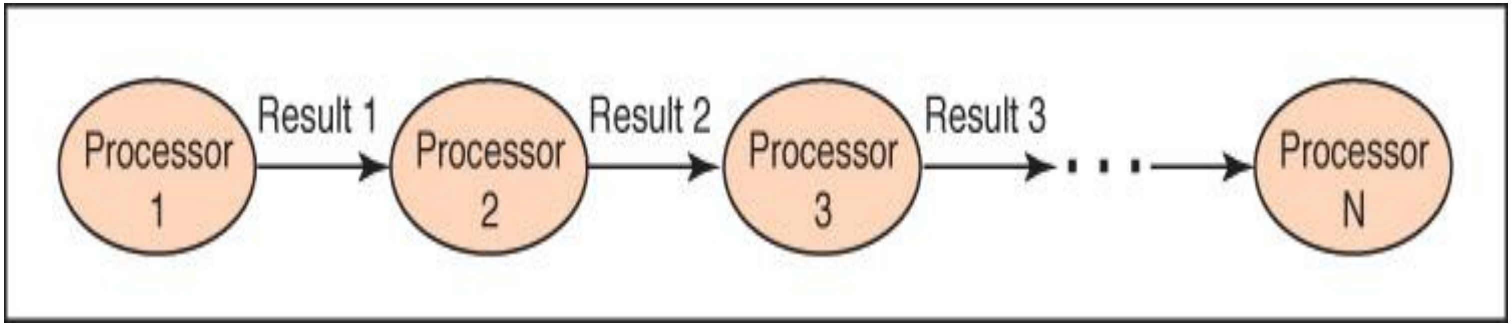


  - However, since 1990, alternative parallel-processing systems have entered the marketplace. They have the potential to process much more data at much higher speeds:
    - **synchronous processing**
    - **pipelining**
    - **shared-memory configuration**

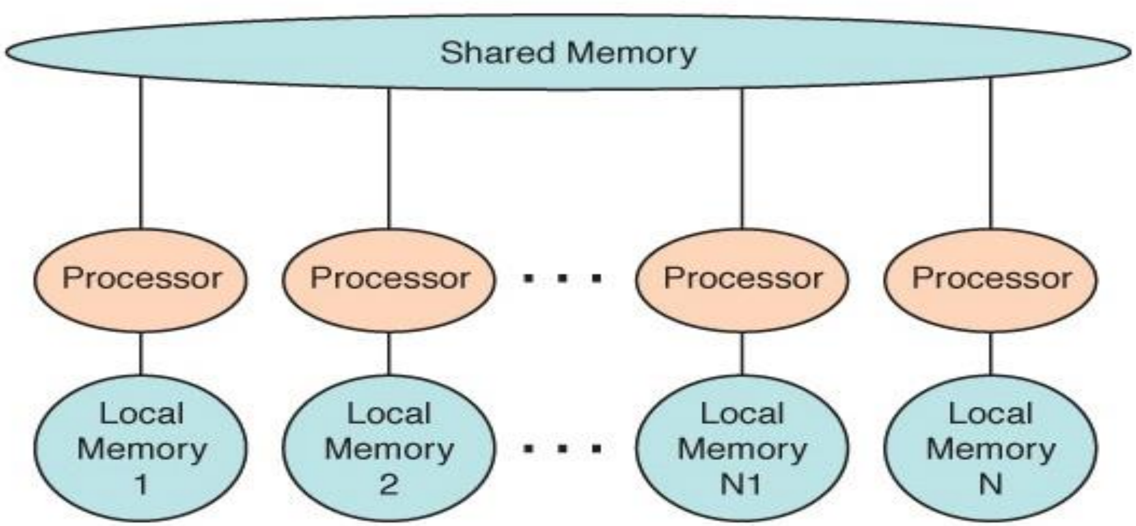
- **Non-von Neumann Architectures**
  - **synchronous processing**
    - to have multiple processors apply the same program to multiple data sets.
    - this approach is similar to that of the NASA backup system in which three computers do the same thing as a security measure. However, here there are multiple processors applying the same process to different data sets in parallel.



- **Non-von Neumann Architectures**
  - **pipelining**
    - processors are arranged in tandem, where each processor contributes one part to an overall computation.
    - reminiscent of an assembly line. When this organization is applied to data, the first processor does the first task. Then the second processor starts working on the output from the first processor, while the first processor applies its computation to the next data set. Eventually, each processor is working on one phase of the job, each getting material or data from the previous stage of processing, and each in turn handing over its work to the next stage.



- **Non-von Neumann Architectures**
  - **shared-memory configuration**
    - different processors doing different things with different data.
    - allows processors to work independently much of the time, but introduces problems of coordination among the processors.
    - leads to a configuration where the processors each have a local memory and a shared memory. The processors use the shared memory for communication, and the configuration is thus called a shared-memory configuration.



Bibliografia

- N. Dale and J. Lewis, “Computer Science Illuminated”, 7th Edition, Jones and Bartlett Publishers, 2019;
- J. Glenn Brookshea, “Computer Science: An Overview”, 13th Edition, Pearson, 2018;
- J. Delgado and C. Ribeiro, “Arquitectura de Computadores”, 5ª Edição, FCA, 2014
- Cisco Academy, "Linux Essentials", Cisco, 2018
- Rui Vasco Monteiro, Filipe Neves, João Pereira, Nuno Rodrigues e Ricardo Martinho, “Tecnologia dos Equipamentos Informáticos”, 2004, FCA