Computer Architecture & Assembly Language

Chapter 1

Computer System Overview

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY
UNIVERSITY OF SCIENCE

Thái Hùng Văn
thvan@fit.hcmus.edu.vn

Outline

- Definitions & basic concepts
- Etymology
- Generations of Computer
- Types
- Hardware
- Software

Definitions & basic concepts

4.0

Definitions & basic concepts

- A **computer** is a machine/device that performs processes, calculations and operations based on instructions provided by a **program**.
- Computers are designed to execute apps and provide solutions by combining integrated **hardware** & **software** components. They are used as control systems for a wide range of industrial and consumer devices.
- A "complete" computer including the hardware, the OS (main software), and necessary peripheral equipment can be called a **computer system**.
- Today, the **IoT** allows connecting computer devices to the Internet and to other connected devices. Many computers can work together to share information, operate as design.

Definitions & basic concepts

- Computers have greatly affected our daily lives - helping us complete an extremely wide range of tasks by **software programs**
- **Program** is a set of step-by-step instructions that tells or directs the computer what to do. It sequences the tasks a user wants to be done and produces the results or output needed
- Computer programs are designed specifically for each task. They are created with programming languages
- A **programming language** is a formal language, comprises a set of instructions that produce various kinds of output, used to implement **algorithms** to perform specific tasks.
- A **programmer** is the person who designs a program, converts problem solutions into instructions for the computer.

Etymology of “Computer”

- "Computer" comes from the Latin "**putare**" which means both to **think** and to prune**.** **Computare** (**com** - means "**together**") also meant **calculate**. [https://www.bbc.com/news/blogs-magazine-monitor-35428300]
- The *Online Etymology Dictionary* gives the first attested use of "computer" word in the 1640s, meaning "**one who calculates**“.
- Since 1897, the use of the term to mean "**calculating machine**“. It has been a “**programmable digital electronic computer**“ from 1945
- Now, a computer is a **machine or device** that **performs processes by a program.**

Generations of Computer

Generations of Computer

- The development of electronic computers can be divided into 5 generations depending upon the technologies used:
    - ❑ **1st gen: vacuum tube** (~1940-1956)
    - ❑ **2nd gen: transistor** (~1956-1963)
    - ❑ **3rd gen: integrated circuits (IC)** (~1964-1971)
    - ❑ **4th gen: microprocessor** (~1971-2010)
    - ❑ **5th gen: artificial intelligence (AI)** (~2010 to present)
- Future generations might be **quantum computers**. They are probably 10^{14} times faster than the fastest supercomputers [https://www.livescience.com/china-quantum-supremacy.html]
- [There are several new computer types that can be used in the future: Quantum, Chemical, DNA, Optical, Spintronics-based, Wetware/Organic computer, etc]

Generations of Computer – 1st gen

- The 1st-gen computers used vacuum tubes technology.
- ***Advantages:*** Vacuum tubes were the only electronic component available during those days; these computers could calculate in ms.
- ***Disadvantages*:** The computers were very large in size. weight was about 30 tones; very costly; very slow speed; very less data storage; very faulty and not versatile; used machine language only & limited programming capabilities; consumed a large amount of energy; limited commercial use; …
- Example: The ENIAC consisted of nearly 20,000 vacuum tubes, as well as 10,000 capacitors and 70,000 resistors. It weighed over 30 tons and took up a lot of space, requiring a large room to house it.

Generations of Computer – 2nd gen

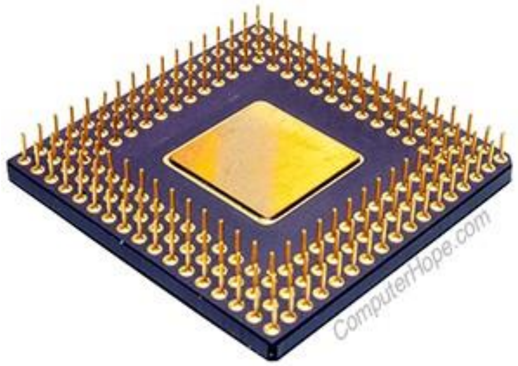
- The 2nd-gen were based on transistor instead of vacuum tubes
- ***Advantages:*** the size of electron component decreased and reduced the computer size; less energy and not produce as much heat as the first genration; Assembly language were used; low cost, better speed, better portability, more reliable as compared to 1st-gen; wider commercial use
- ***Disadvantages*****:** cooling system, constant maintenance was required; only used for specific purposes; commercial production was difficult
- The first computer to use transistors was the TX-0 (1956); some others: IBM 7070, Philco Transac S-1000, and RCA 501..

Generations of Computer – 3rd gen

- These computers were based on IC (IC was a single component containing number of transistors), use SSI or MSI technology
- ***Advantages:*** cheaper, faster, more reliable, smaller, IC not only reduce the computer size but it also improves the performance, has big storage capacity, mouse and keyboard are used for input instead of punch cards, caculate in ns.
- ***Disadvantages*:** IC chips are difficult to maintain, air conditioning is required, highly sophisticated technology required for the manufacturing of IC chips.

Generations of Computer – 4th gen

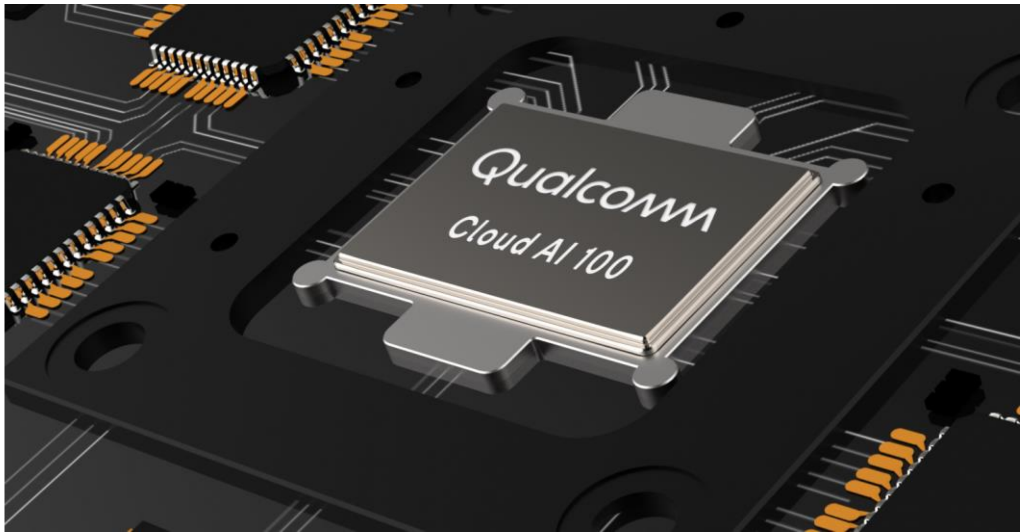
- These computers took advantage of the invention of the micro-processors, more commonly known as a CPU. ICs, chips and CPU are built by LSI or VLSI technology that helped make it possible for desktop PC and laptop
- ***Advantages:*** Fastest in computation and size get reduced, less maintenance as compared to the previous generation of computer, all types of high-level programming language can be used.
- ***Disadvantages*:** The microprocessor design and fabrication are very complex, air conditioning is required in many cases due to the presence of Ics, advance technology is required.



Generations of Computer – 5th gen

- The 5th-gen computers were based on AI, leaps have been made in AI technology and computers, AI devices can respond to natural language input and are capable of learning and self-organization.
- Today, ULSI (Ultra Large Scale Integration) technology resulting in the production of microprocessor chips having ten million electronic component.

  Ex: The 7nm "Qualcomm Cloud AI 100" chip can hit "far greater" than 100 TOPs (the Snapdragon 855 maxes out at around 7 TOPs), it is a power-efficient edge and cloud computing chip purpose-built for machine learning and big data workloads.

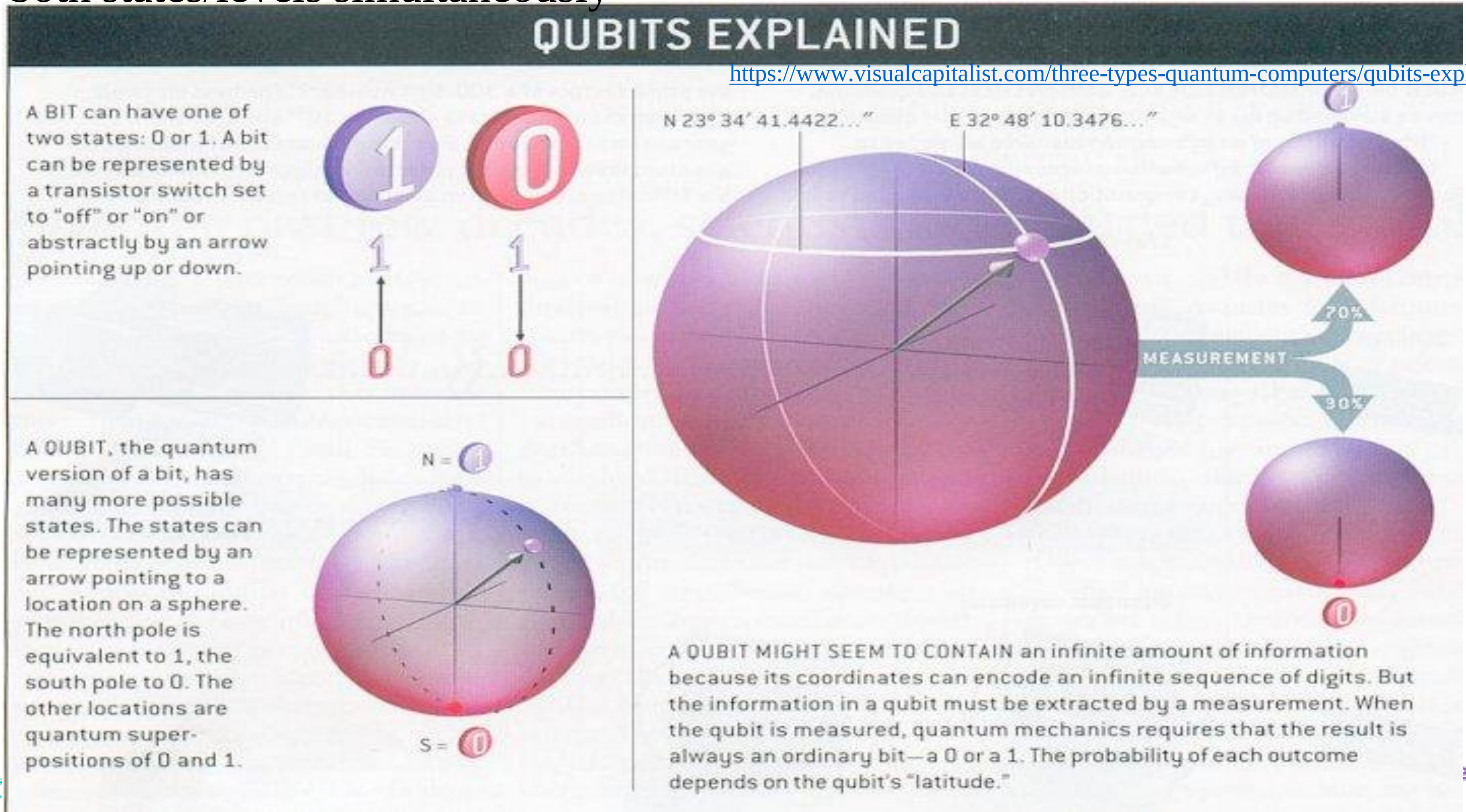


Generations of Computer – 5th gen

- Today, many key players in the IT industry have focused on developing AI chips and applications. Furthermore, emergence of quantum computing and increase in implementation of AI chips in robotics drive the growth of the global AI chip market.
- The AI chip market was valued at $6,638 million in 2018, and is projected to reach $91,185 million by 2025. [https://www.alliedmarketresearch.com/artificial-intelligence-chip-market]
- ***Advantages:*** more reliable and works faster, available in different sizes and unique features, provides computers with more user-friendly interfaces with multimedia features.
- ***Disadvantages:*** need very low-level languages, may make the human brains dull and doomed.
- *Ques: Do you know the quantum computer? Is it 6th generation?*

Generations of Computer – Quantum Computer

- In theory, Quantum computer with 64-qubit is faster than traditional electronic computer with 64-bit CPU very very much: up to 2^{64} = 18 billion billion times!
- Reason: quantum mechanics allows the qubit to be in a coherent superposition of both states/levels simultaneously



QUBITS EXPLAINED

https://www.visualcapitalist.com/three-types-quantum-computers/qubits-explained/

A BIT can have one of two states: 0 or 1. A bit can be represented by a transistor switch set to "off" or "on" or abstractly by an arrow pointing up or down.

A QUBIT, the quantum version of a bit, has many more possible states. The states can be represented by an arrow pointing to a location on a sphere. The north pole is equivalent to 1, the south pole to 0. The other locations are quantum superpositions of 0 and 1.

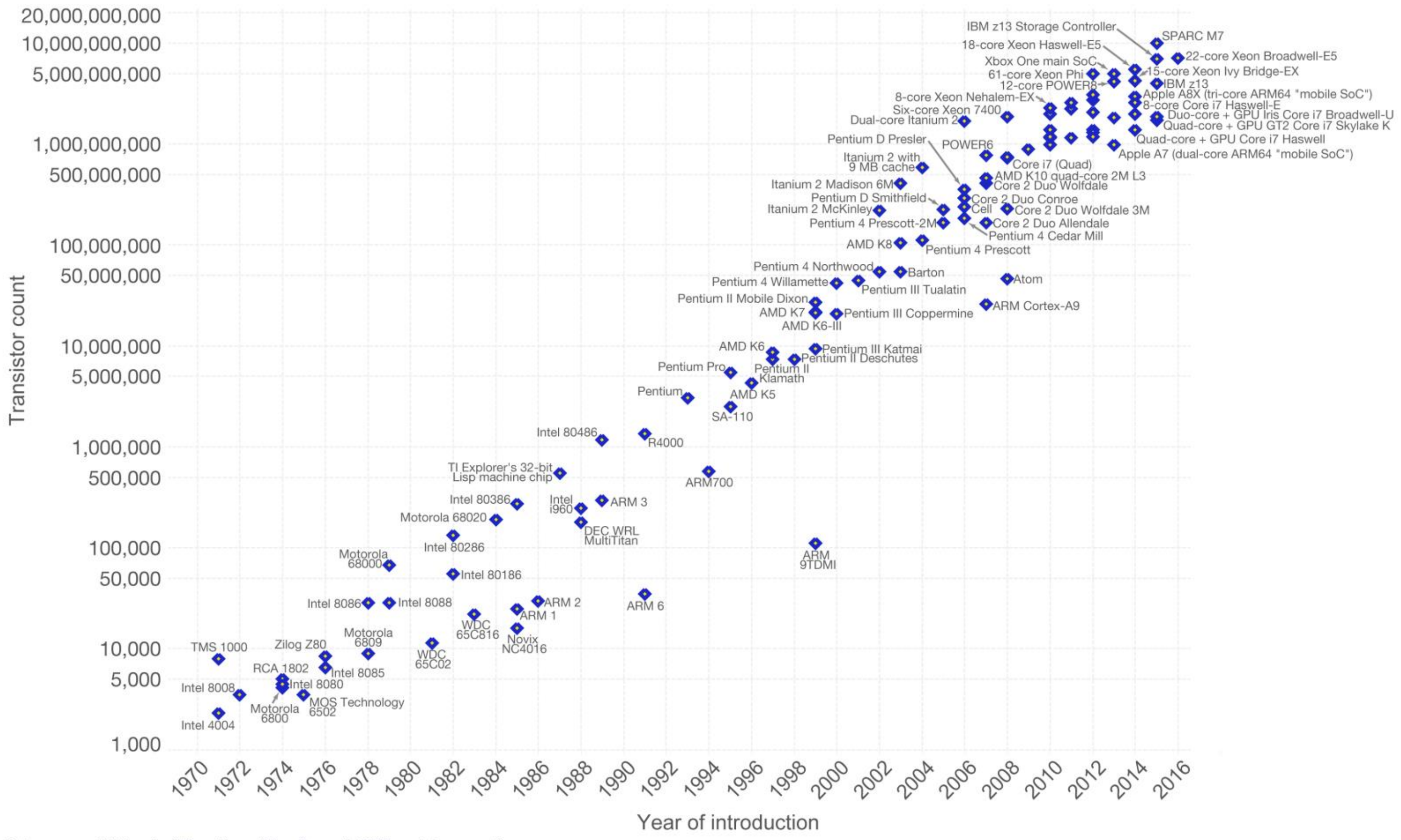
A QUBIT MIGHT SEEM TO CONTAIN an infinite amount of information because its coordinates can encode an infinite sequence of digits. But the information in a qubit must be extracted by a measurement. When the qubit is measured, quantum mechanics requires that the result is always an ordinary bit—a 0 or a 1. The probability of each outcome depends on the qubit's "latitude."

Moore's Law

Moore's Law – The number of transistors on integrated circuit chips (1971-2016)

Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are strongly linked to Moore's law.



Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)

The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Computer Generations - Quiz

1. What is Moore's law? Why is it important?
2. What is nanometer technology in processor?
3. Is Moore's law still alive for pm technology? Why?
4. What do “5nm” and “10nm” mean for CPUs? Why do they matter?
5. Is 7nm better than 14nm technology? Why?
6. Which is the best laptop /phone processor in 2019? Specifications?
7. Describe your laptop /phone processor information

[Homework for group (2-4)]

Types

Types - by size and form-factor

- Mainframe computer
- Supercomputer
- Minicomputer
- Microcomputer
- Workstation
- Personal computer
- Laptop
- Tablet computer
- Smartphone
- Single-board computer

Types - by architecture

- Analog computer
- Digital computer
- Hybrid computer
- Harvard architecture
- Von Neumann architecture
- Reduced instruction set computer

Computer Types - Quiz

1. What type of computer is the strongest?
2. What type of computer is most popular?
3. What type of computer has the smallest size?
4. What type of computer is the cheapest?
5. Your favorite computer type? Why?
6. Do you know Raspberry Pi 4 computer? Find out about it and compare to PC.

[Homework for group]

Hardware

Hardware

- The term ***hardware*** covers all of those parts of a computer that are (tangible) physical objects. *(Circuits, chips, graphic cards, sound cards, RAM, mobo, displays, PSUs, cables, printers, keyboards, and "mice" input devices are all hardware)*
- A **general purpose computer** has 4 main components: **ALU, CU, memory, I/O devices**. These parts are interconnected by **buses** (wires or build-in board)
- Inside these parts are thousands to trillions of **electrical circuits** that can be turned on / off (means bit 1 /bit 0) by electronic **switches**.
- The circuits are arranged in **logic gates** so that one or more circuits can control the state of other circuits.

Hardware - Input devices

- When unprocessed data is sent to the computer with the help of input devices, the data is processed and sent to output devices.
- The input devices may be hand-operated or automated. The act of processing is mainly regulated by the CPU.
- Some examples of input devices are: Keyboard, Mouse /Trackball /TouchScreen, Microphone, Digital camera /video recorder /webcam, Scanner, Joystick, Air Mouse (with Keyboard and Mic)

Hardware - Output devices

- An output device is any peripheral that receives data from a computer.
- Output /input devices are all peripheral hardware, and they are usually connected to a computer by cables, or wireless networks *(most of them use the 2.4 GHz band as the preferred frequency range for communication)*
- Some examples of input devices are: Monitor / TV /Video card, Speaker /Headphone /Sound card, Projector, Printer,...

Hardware – I/O Devices Quiz

1. How to use multiple monitors on a computer (PC) ?
2. How to use one monitor on multiple computers ?
3. How to use one speaker on multiple computers ?
4. How to use one printer on multiple computers ?
5. How to use one mouse and keyboard on multiple computers ?
6. How to remote control other computers ?
7. How to use the mouse and keyboard on smartphone ?
8. How to convert a wired I/O device to wireless device?

[Homework for group]

Hardware - ALU

- The set of arithmetic operations that a particular ALU supports may be addition, subtraction, multiplication, division, sine, cosine, square roots, etc. Some can only operate on integers.
- However, any computer can be programmed to perform any arithmetic operation *(but it will take more time to do so if its ALU does not directly support)*.
- An ALU may also compare numbers or perform other logic operations (And /Or /Xor /Not) and return boolean truth values.
- Computers may contain multiple ALUs, allowing them to process several instructions simultaneously.
- Graphics processors and computers with SIMD and MIMD features often contain ALUs that can perform arithmetic on vectors and matrices.

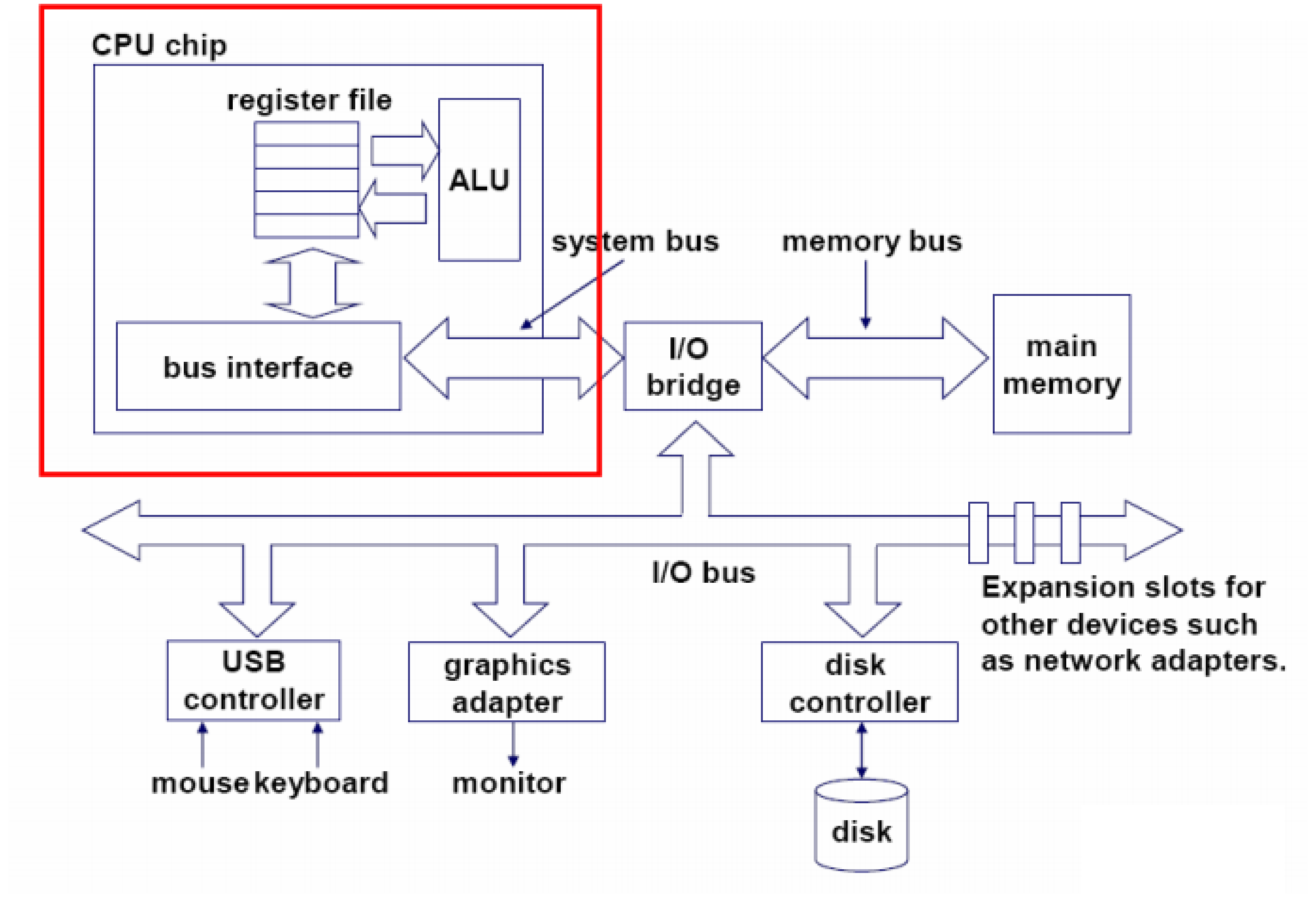
Hardware - CU

- The CU *(control unit /central controller)* reads and interprets (decodes) the program instructions, transforming them into control signals that activate other computer parts
- A key component is PC *(program counter)*, a special register that keeps track of which location in memory the next instruction is to be read from. The control system's function is as follows:
    - ❑ **Read the code for the next instruction by the PC.**
    - ❑ **Decode the numerical code for the instruction into a set of commands**
    - ❑ **Increment the PC so it points to the next instruction.**
    - ❑ **Read whatever data the instruction requires (the location stored within code)**
    - ❑ **Provide the necessary data to an ALU or register.**
    - ❑ **If the instruction requires an ALU or specialized hardware to complete, instruct the hardware to perform the requested operation.**
    - ❑ **Write the result from ALU back to memory location /register /output device.**
    - ❑ **Jump back to step1.**

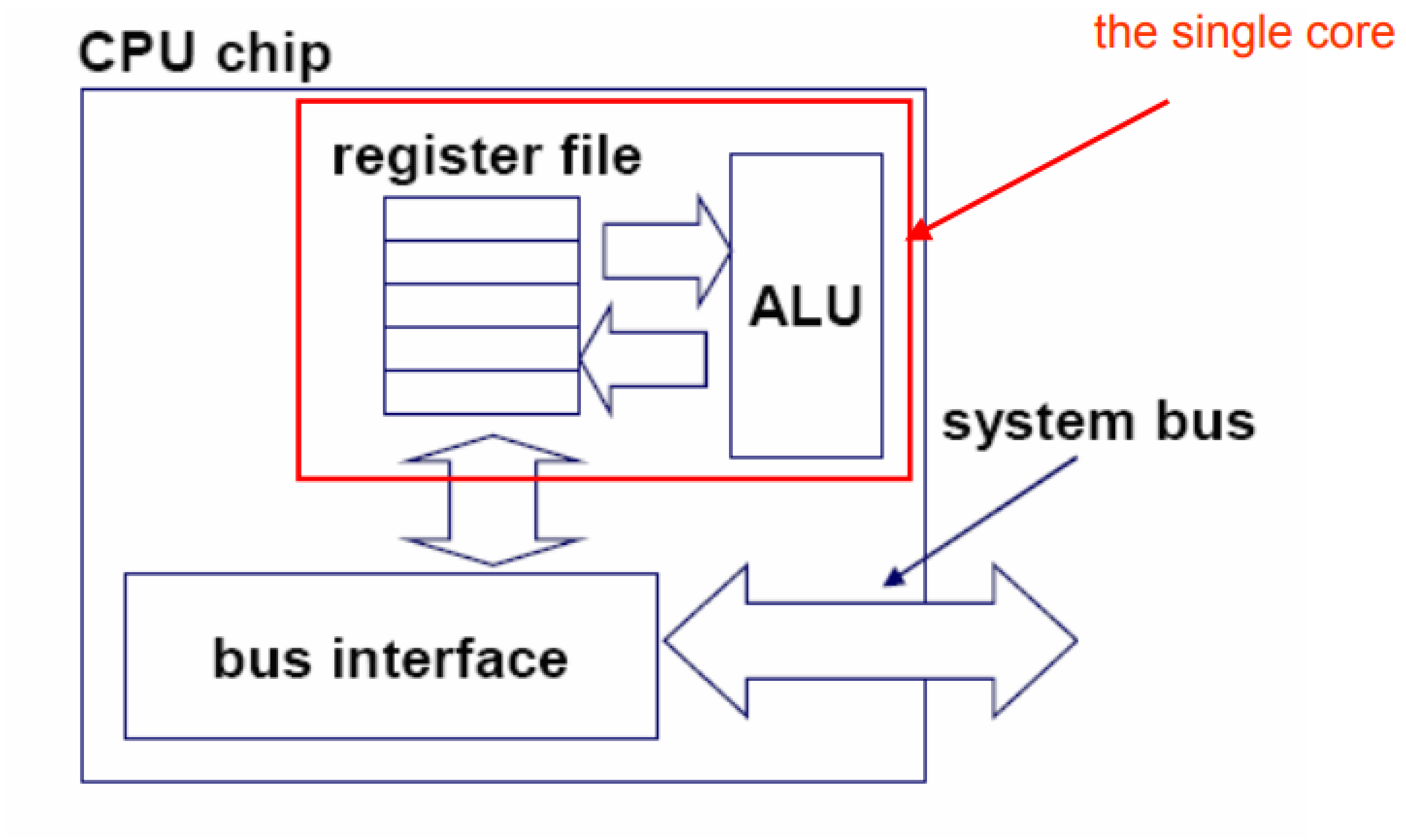
Hardware - CPU

- The CU, ALU, and registers are collectively known as a central processing unit (CPU).
- Early CPUs were composed of many separate components but since the mid-1970s CPUs have been constructed on a single integrated circuit called a **microprocessor**.
- Nowadays, general purpose computers often use **multi-core processors.** A multicore processor is a single integrated circuit that contains **two or more separate processing units**, called **cores**, each core can read and execute program instructions, as if the computer had several processors.

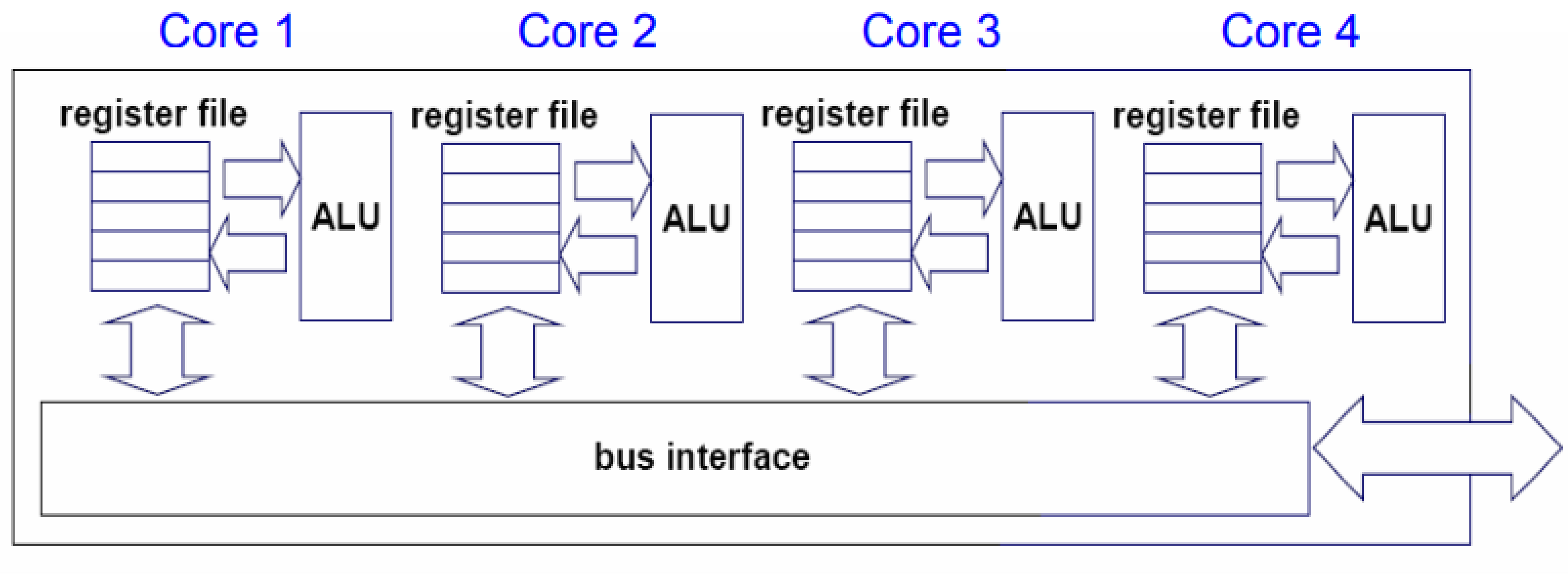
Hardware – Single-core computer



Hardware – Single-core CPU



Hardware – Multi-core CPU



Hardware – MIMD /SIMD

- Multi-core processors are **MIMD**: Different cores execute different threads (**M**ultiple **I**nstructions), operating on different parts of memory (**M**ultiple **D**ata).
- Multi-core is a shared memory multiprocessor: All cores share the same memory
- **Multithreading** is the ability of a CPU. So each core can run separate threads at the same time.
- **SIMD** (Single Instruction Multiple Data) is an instruction set available mostly on all current processors. A single instruction is executed in parallel on multiple data points as opposed to executing multiple instructions.

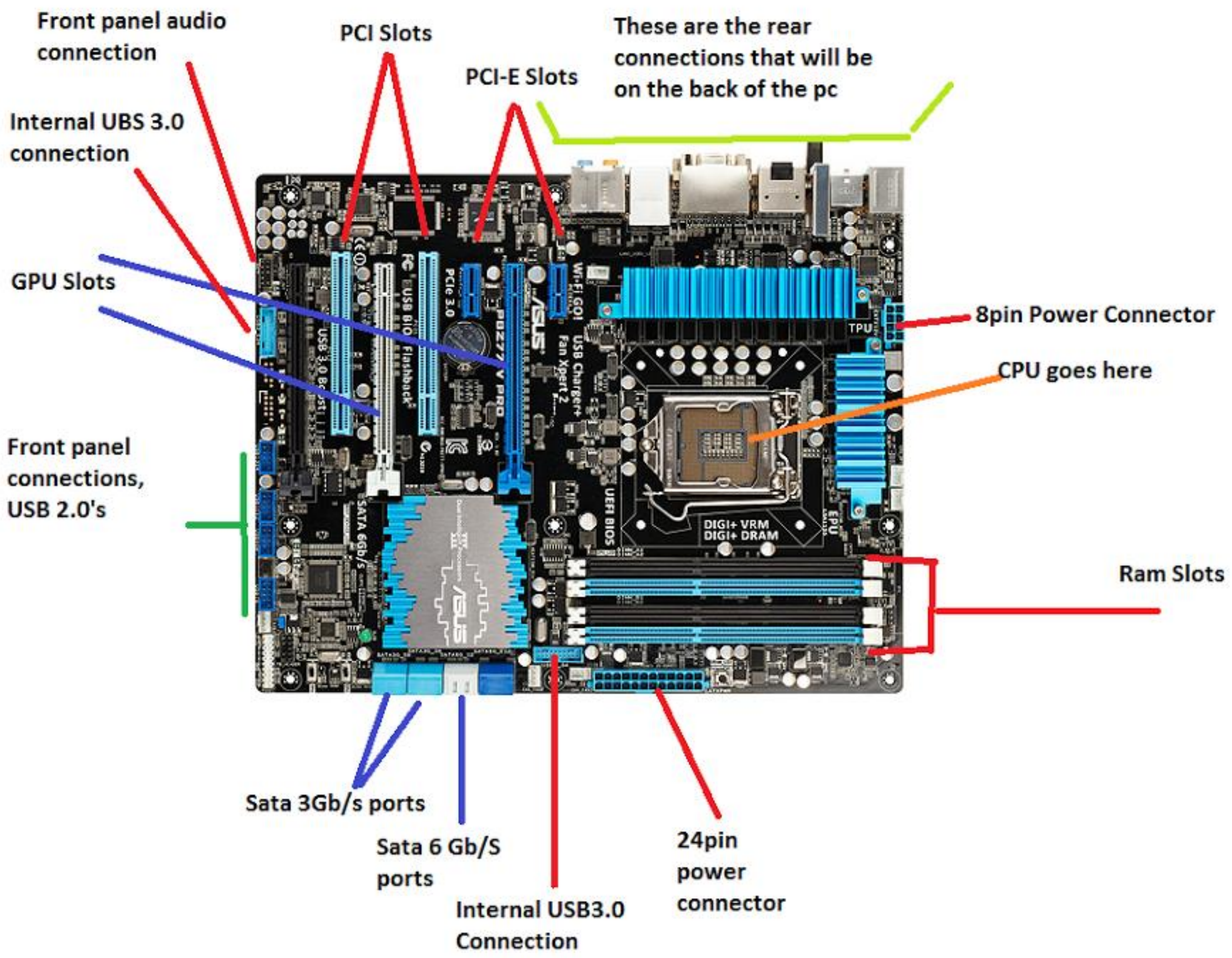
Hardware - Memory

- The computer's memory can be viewed as a list of cells where numbers can be placed or read.
- Each cell has a numbered "address" and can store a single number.
- In almost computers, each memory cell is set up to store binary numbers in groups of 8 bits (called a **byte**).
- To store large numbers, several consecutive bytes may be used (typically, 2, 4 or 8).
- The computer can store any kind of information in memory (if they can be represented numerically)
- Modern computers have billions or even trillions of bytes of memory.

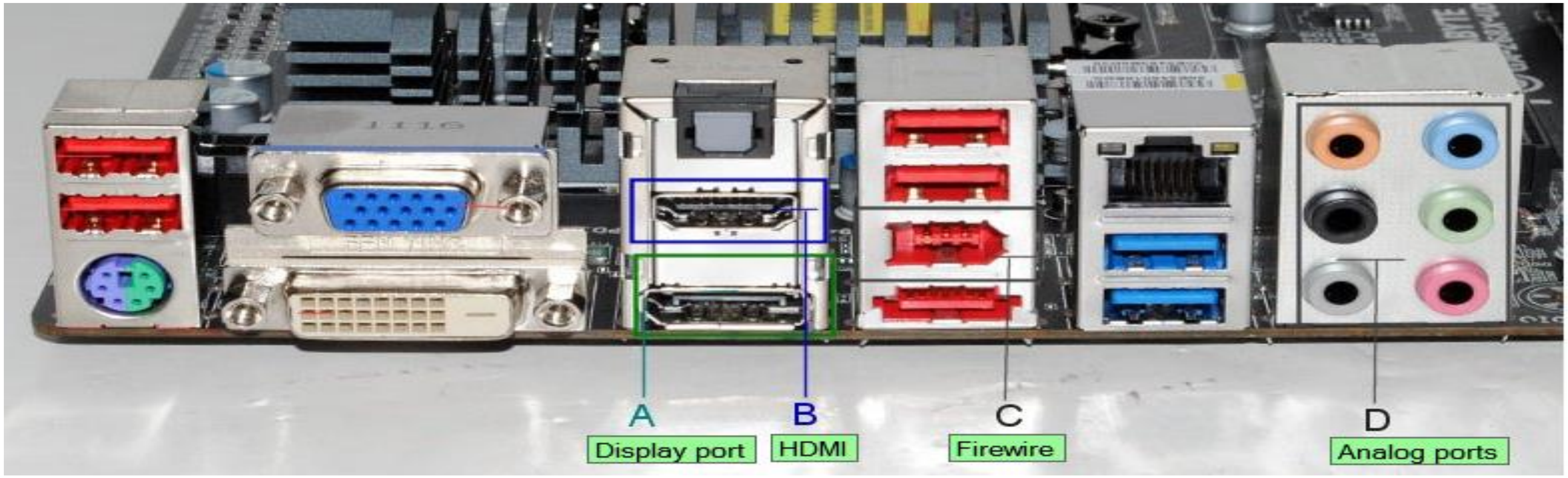
Hardware – Memory - types

- **RAM** (Random-Access Memory) is the main memory with big size (usually, in GB). It can be read and written anytime by CPU. The data in RAM is lost when the computer is shut off or restart.
- **ROM** (Read-Only Memory) is typically used to store the computer's initial start-up instructions.
- **Registers** are the smallest and the fastest storage. They are organized inside the CPU and provide the fastest way to access data.
- (CPU) **cache** is a type of cache memory that a processor uses to access data and programs much more quickly than through host RAM. It enables storing and providing access to frequently used programs and data.

Hardware – Physical view

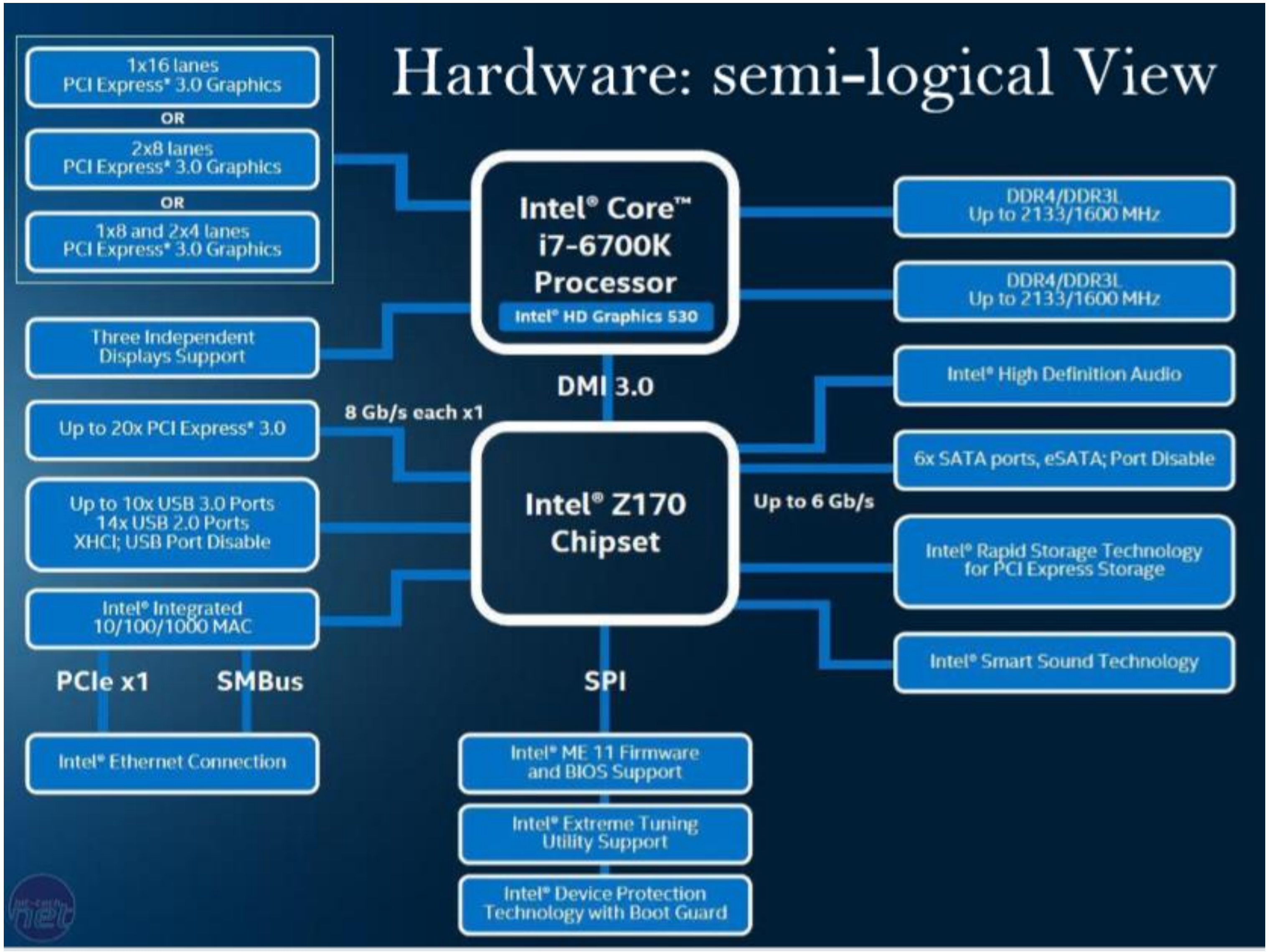

Front panel audio connection
PCI Slots
PCI-E Slots
These are the rear connections that will be on the back of the pc
Internal UBS 3.0 connection
GPU Slots
8pin Power Connector
CPU goes here
Front panel connections, USB 2.0's
Ram Slots
Sata 3Gb/s ports
Sata 6 Gb/S ports
Internal USB3.0 Connection
24pin power connector

Hardware – Physical View

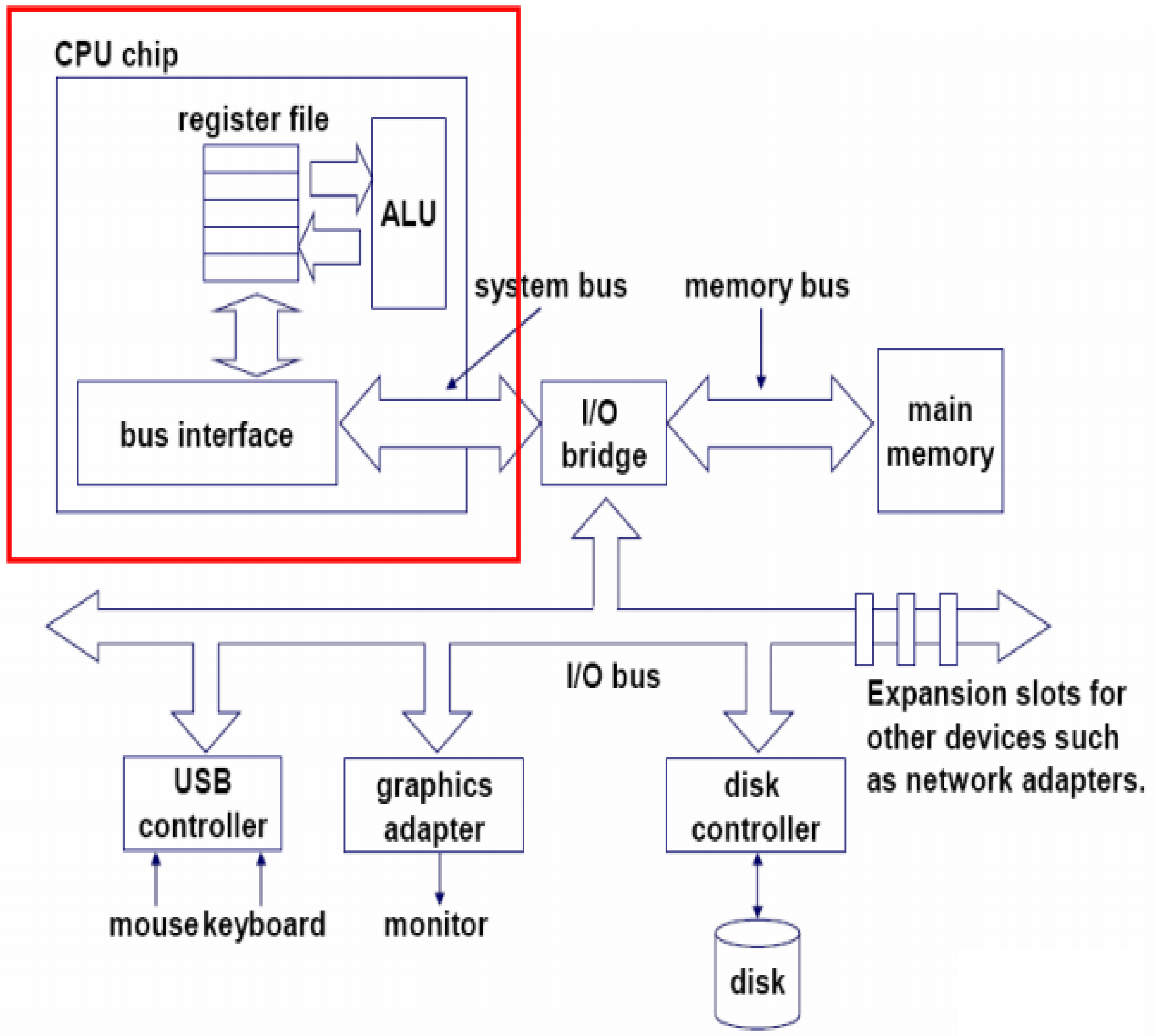


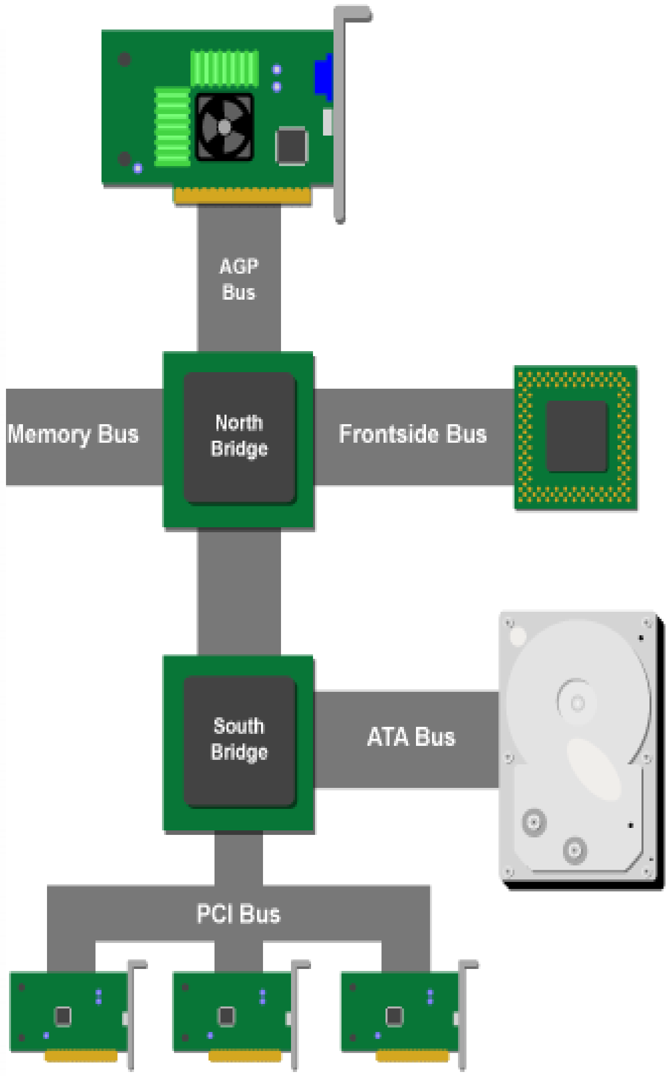
- The Motherboard (system board /main board /PCB) is an important component of any PC. It connects all the other components.
- Basic Components of MotherBoard: *Form factors, Chipsets, Processor sockets, Memory slots, Expansion slots, Disk Connectors, Power connectors, BIOS/firmware, CMOS and CMOS battery, Back-panel connectors, Front-panel connectors.*

Hardware – Semi-logical View


Hardware: semi-logical View
1x16 lanes PCI Express* 3.0 Graphics
OR
2x8 lanes PCI Express* 3.0 Graphics
OR
1x8 and 2x4 lanes PCI Express* 3.0 Graphics
Intel® Core™ i7-6700K Processor
Intel® HD Graphics 530
DDR4/DDR3L Up to 2133/1600 MHz
DDR4/DDR3L Up to 2133/1600 MHz
Three Independent Displays Support
DMI 3.0
8 Gb/s each x1
Up to 20x PCI Express* 3.0
Intel® High Definition Audio
6x SATA ports, eSATA; Port Disable
Up to 6 Gb/s
Intel® Z170 Chipset
Up to 10x USB 3.0 Ports 14x USB 2.0 Ports XHCI; USB Port Disable
Intel® Rapid Storage Technology for PCI Express Storage
Intel® Integrated 10/100/1000 MAC
Intel® Smart Sound Technology
PCIe x1
SMBus
SPI
Intel® Ethernet Connection
Intel® ME 11 Firmware and BIOS Support
Intel® Extreme Tuning Utility Support
Intel® Device Protection Technology with Boot Guard

Hardware – Logical View





Hardware – Quiz

1. Multi-core processor is multiprocessor ?
2. Why Multi-core?
3. How many cores are on your CPU?
4. How many cores does Intel Core-I9 CPU have?
5. How a control unit works inside a CPU?
6. What is ROM and RAM in computer?
7. What is Cache and Register in CPU?
8. What is stored in PC ROM and smartphone ROM?
9. What is different between ROM and RAM?
10. Describe briefly the components of PC Motherboard

Software

4.0

Software

- Software consists of programs written to perform specific tasks.
- Two types of programs:
  - ❑**System programs**
  - ❑**Application programs**
- System Programs
  - ❑**System programs control the computer.**
  - ❑**When computer turn on, the operating system (OS) is load and manages all of the other application programs.**
- Application programs
  - ❑**The application programs make use of the OS by making requests for services through a defined application program interface (API).**

System software

- **Operating system (OS)**: Harnesses communication between hardware, system programs, and other applications.
- **Device driver**: Enables device communication with the OS and other programs.
- **Firmware**: Enables device control and identification.
- **Translator**: Translates high-level languages to machine codes.
- **Utility**: Ensures optimum functionality of devices and applications.

Computer Language

- Machine language is the most basic language of a computer.
- Machine instructions are sequences of 0s and 1s.
- Every computer directly understands its own machine language.

➔ Very difficult for programmers

Assembly Language

- Early computers programmed in machine language (ML) ➔ so difficult to programming
- Assembly languages (AL) were developed to make programmer's job easier.
- In AL, an instruction is an easy-to-remember form called a mnemonic.
- Assembler: Translates AL instructions into ML

High-level Language

- Programming in AL is still difficult and inconvenient
- High-level languages (HLLs) make programming easier. Closer to spoken languages.
- There are many HLLs in use:
    - ❑ **C**
    - ❑ **C++**
    - ❑ **C#**
    - ❑ **Java**
    - ❑ **Java Script**
    - ❑ **Python**
    - ❑ **Golang**

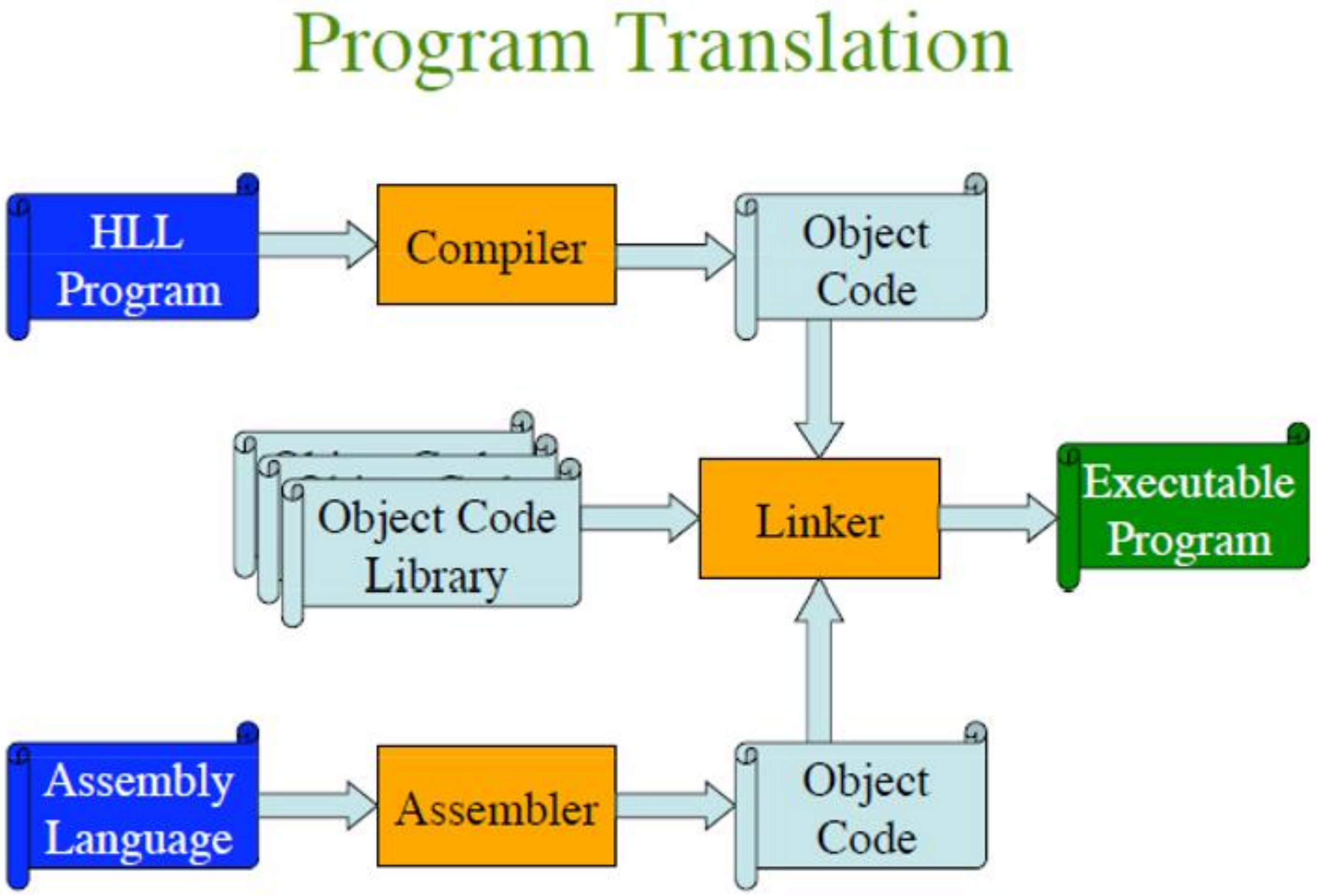
Evolution of Programming Languages

1. ML or First Generation Programming Language (GPL) – lowest level of programming.
2. AL or Second GPL – considered as low level language uses Mnemonic codes (abbreviations that easy to remember).
3. HLL or Third GPL – language is written in English like manner.
4. Very HLL or Fourth GPL
5. Natural Language – Fifth GPL resemblance to English language.

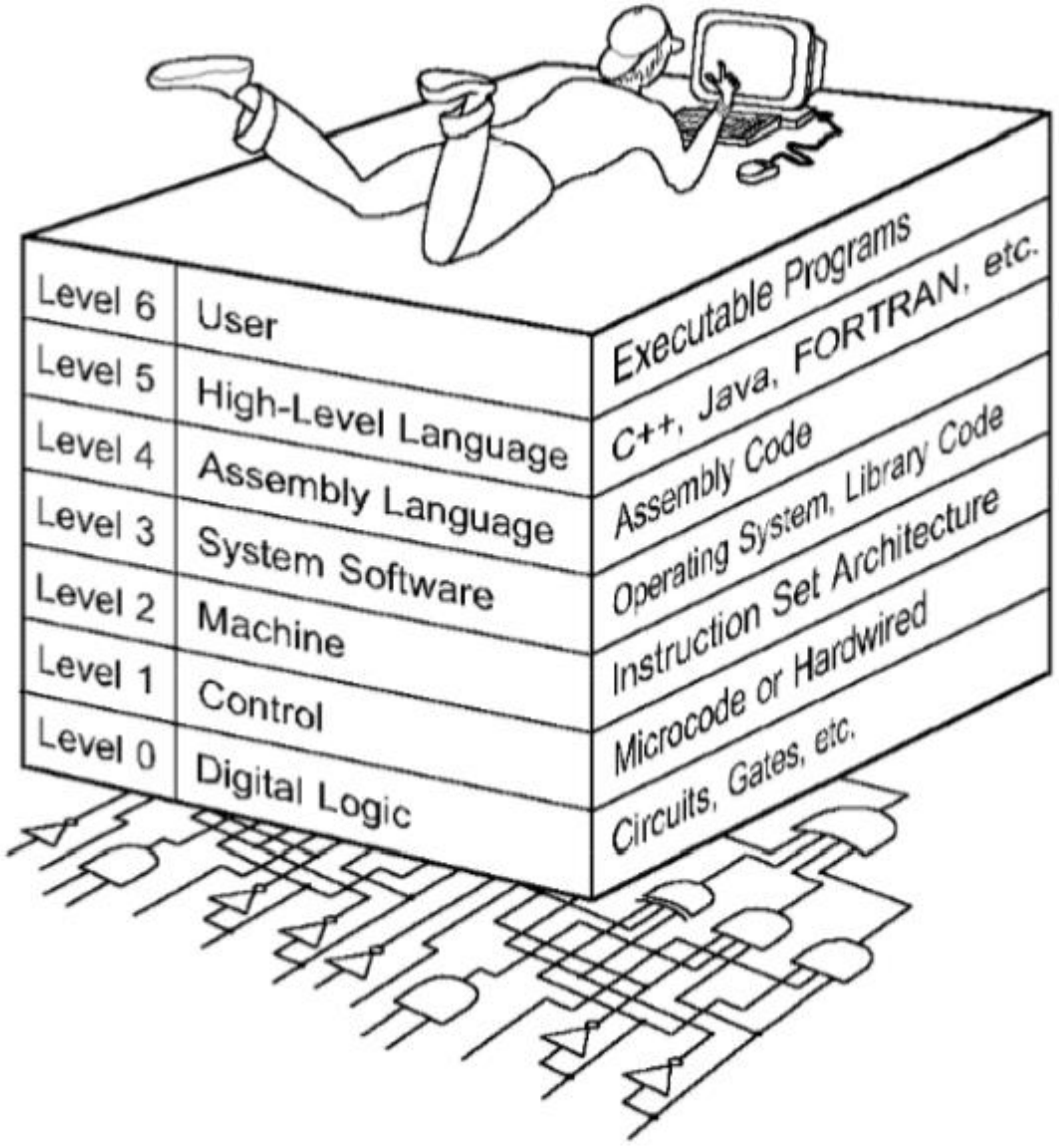
Compiler – Interpreter - Assembler

- Computer only understands ML instructions. Instructions written in AL or HLL must be translated to ML instructions
- Assembler: One AL instruction is mapped to one ML instruction
- Compiler /Interpreter: translates a program written in a HLL into the equivalent ML, generally one HLL instruction is mapped to many ML instruction.
- Compiler translates all HLL program, Interpreter translates one by one

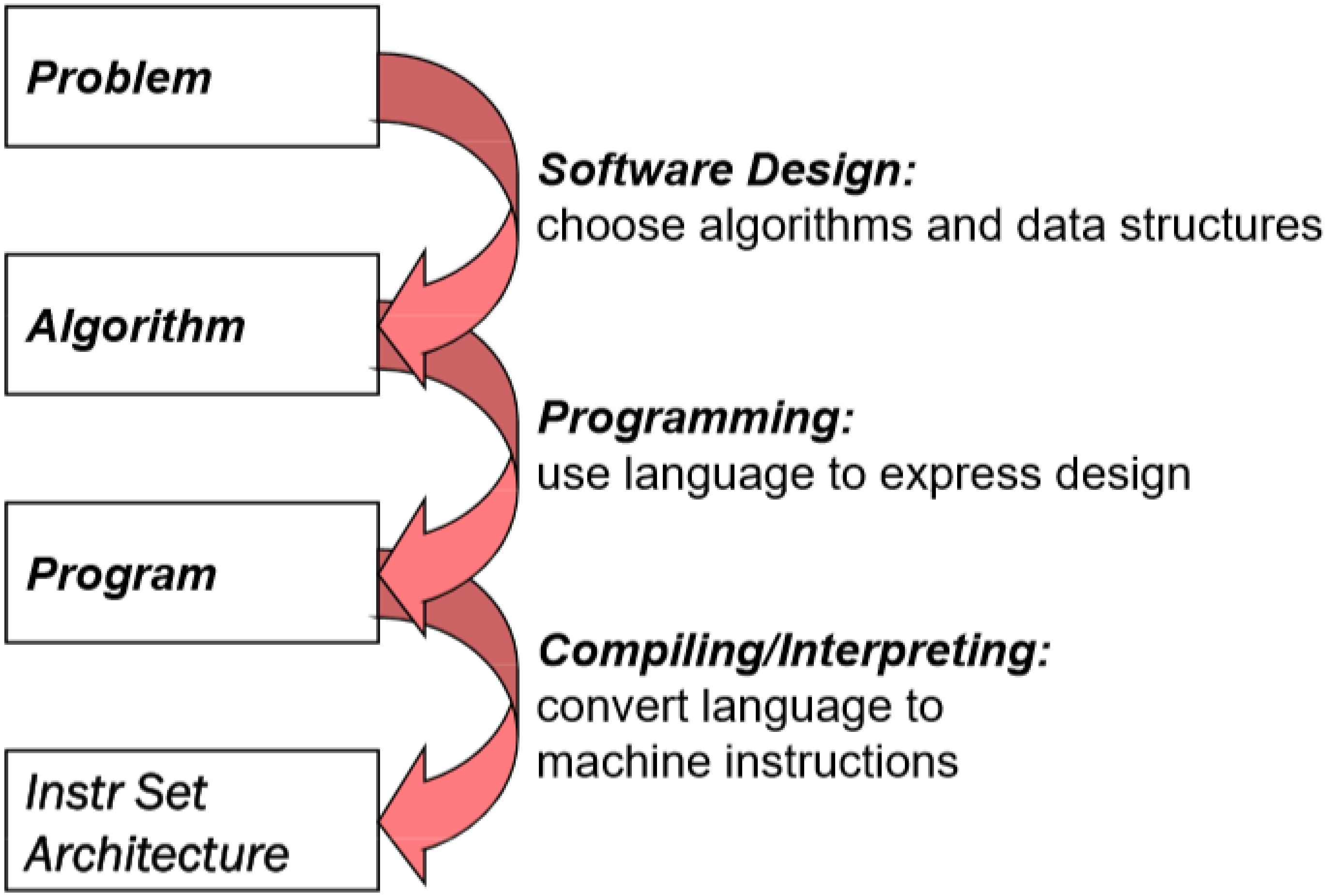
Program Translation Diagram



Computer Level Hierarchy



Solving problem using computer



Software Design:
choose algorithms and data structures

Programming:
use language to express design

Compiling/Interpreting:
convert language to machine instructions

Program Development Life Cycle

- When we want to develop a program using any programming language, we follow a sequence of steps.
- The PDLC is a set of steps or phases that are used to develop a program. Generally, they are as follow:

1. ***Defining*** *the Problem*
2. ***Analyzing*** *the Problem*
3. ***Designing*** *the Program*
4. ***Coding*** *the Program*
5. ***Testing & Debugging*** *the Program*
6. ***Documenting*** *the Program*
7. ***Deploying*** *and* ***Maintaining*** *the Program*

PDLC # *Defining the Problem*

- The first step is to define the problem. In this phase we need to understand the problem statement, what is our requirement, what should be the output of the problem solution.
- In major software projects, this is a job for system analyst, who provides the results of their work to programmers in the form of a program specification.

PDLC # *Analyzing the Problem*

- Read the case thoroughly.
- Define the central issue.
- Define the firm's goals.
- Identify the constraints to the problem.
- Identify all the relevant alternatives.
- Select the best alternative.
- Develop an implementation plan.

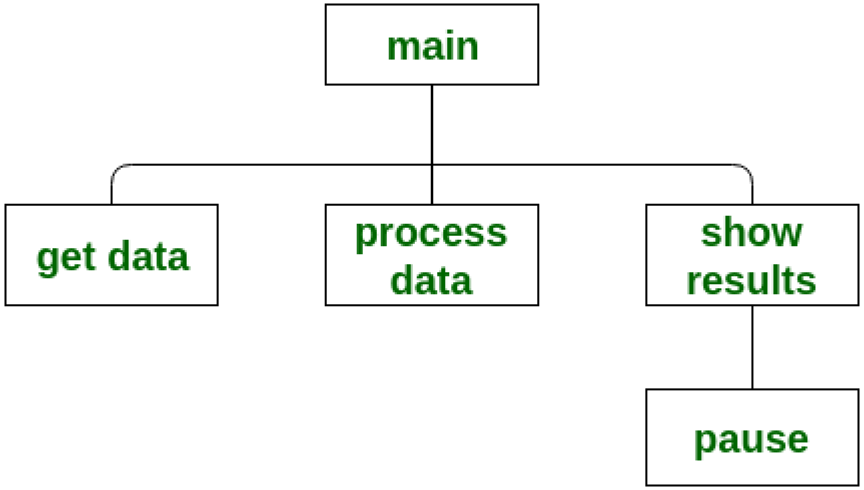
PDLC # *Designing the Program*

- Starts by focusing on the main goal and then breaking the program into manageable components, each of which contributes to this goal. This approach is called ***top-bottom program design*** or ***modular programming***.
- Identify ***main routine***, which is the one of program's major activity.
- Try to divide the various components of the main routine into smaller parts called ***modules***.
- For each module, programmer draws a conceptual plan using an appropriate **program design tool** to visualize how the module will do its assign job

PDLC # *Designing the Program* - Program Design Tools

1. **Structure Charts**, also called *Hierarchy chart*

- Show top-down design of program.
- Each box in the structure chart indicates a task that program must accomplish.
- The Top module, called the *Main module* or *Control module*.



PDLC # *Designing the Program* - Program Design Tools

2. Algorithms

- An ***algorithm*** is a step-by-step description of how to arrive at a solution in the most easiest way.
- Algorithms are not restricted to computer world only. In fact, we use them in everyday life.

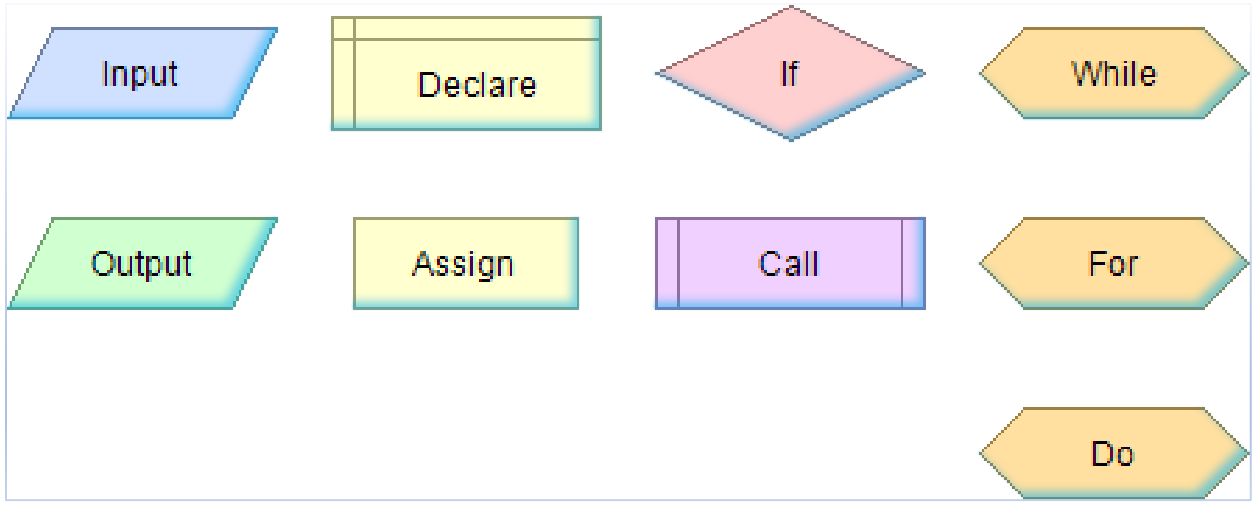
Example: ***Algorithm to find roots of a quadratic equation aX^2+bX+c =0.***

```
Step 1: Input a, b, c
Step 2: If a = 0: Solving linear  equation bX+c =0 and return
Step 3: Calculate discriminant (=b^2-4ac)
Step 4: If discriminant < 0
                return  No solutions
        Else
                r1 ← (-b+√D)/2a
                r2 ← (-b-√D)/2a
Step 5: Stop
```

PDLC # *Designing the Program* - Program Design Tools

3. Flowcharts

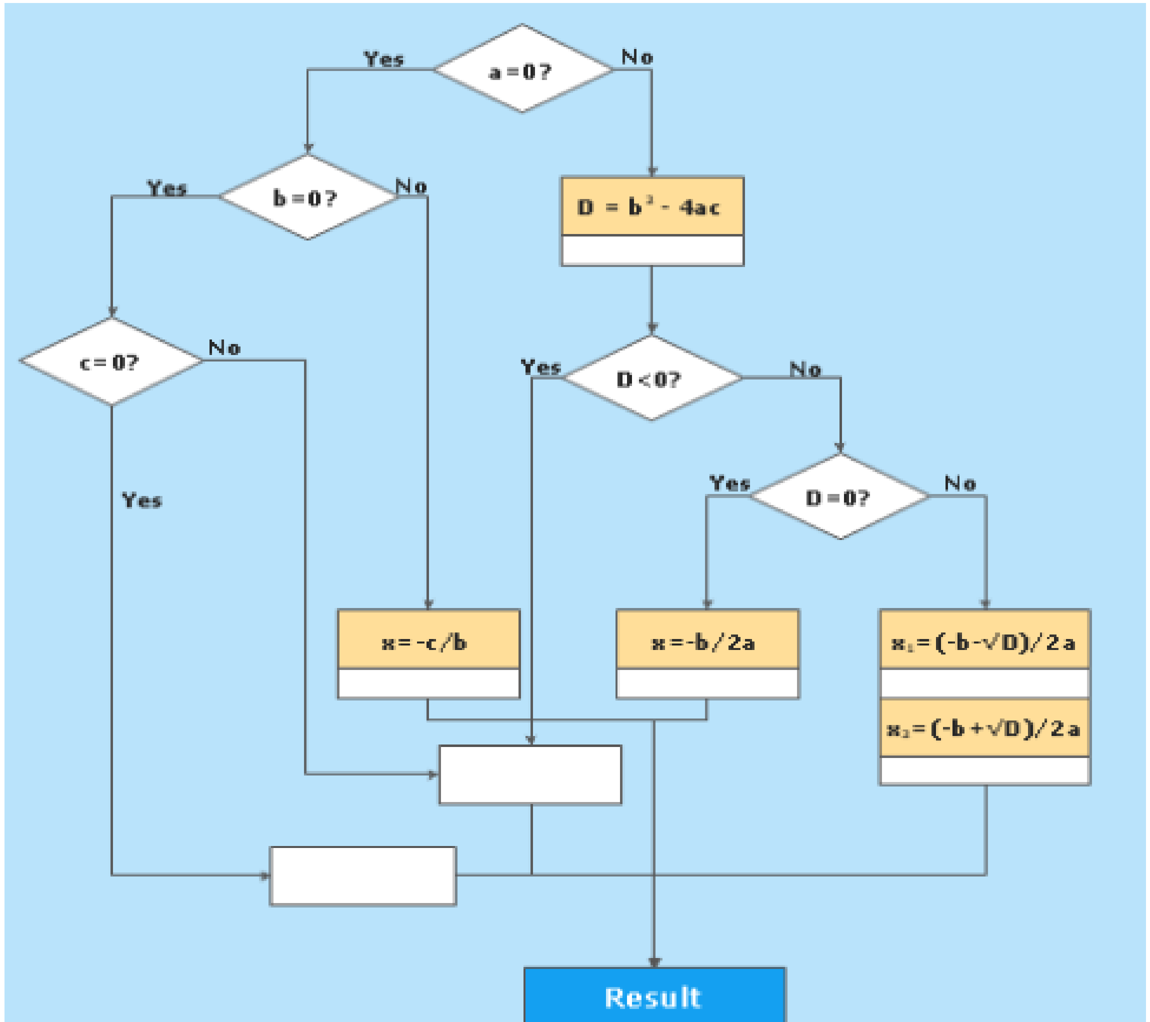
- It is a visual diagram that shows the logic of the program.
- It will help you break down the problem, and very important for teamwork.
- Common symbols used in flowcharts:



PDLC # *Designing the Program* - Program Design Tools

3. Flowcharts -

Example: ***Flowchart to find roots of the equation aX^2+bX+c =0.***



PDLC # *Designing the Program* - Program Design Tools

4. Pseudocode

- It is another tool to describe the way to arrive at a solution.
- It is expressed based on the program language..

Example: ***Pseudocode to find roots of equation aX^2+bX+c =0.***

```
    Input a, b, c
    If (a = 0)
        call fuction_SolvingLinearEquation (b , c)
    Else
        Calculate discriminant D ← b2-4ac
        If D < 0
                return  No solutions
        Else
                r1←(-b+√D)/2a
                r2←(-b-√D)/2a
        EndIf
    EndIf
```

PDLC # *Coding the Program*

- Coding the program means translating an algorithm into specific programming language.
- That means we write the actual program to solve the given problem!
- The technique of programming using only well defined control structures is known as *Structured programming*.
- Programmer must follow the language rules, violation of any rule causes *error*. These errors must be eliminated before going to the next step.

PDLC # *Testing & Debugging the Program*

- After removal of syntax errors, the program will execute. However, the output of the program may not be correct. This is because of logical error in the program.
- A logical error is a mistake that the programmer made while designing the solution to a problem. So the programmer must find and correct logical errors by carefully examining the program output using *Test data*.
- Syntax error and Logical error are collectively known as ***Bugs***. The process of identifying errors and eliminating them is known as ***Debugging***.

PDLC # *Documenting the Program*

- After testing, the software project is almost complete. The *structure charts*, *pseudocodes*, *flowcharts*, *decision tables* developed during the design phase become documentation for others who are associated with the software project.
- This phase ends by writing a manual that provides an overview of the program’s functionality, tutorials for the user, explanations of major program features, reference documentation of all program commands and a description of the error messages generated by the program.

PDLC # *Deploying and Maintaining the Program*

- In the final phase, the program is deployed (installed) at the user’s site. Here also, the program is kept under watch till the user gives a green signal to it.
- Even after the software is completed, it needs to be maintained and evaluated regularly. In software maintenance, the programming team fixes program errors and updates the software.

Evolution of Programming Languages

1. Machine Language or First Generation Programming Language (GPL) – lowest level of programming.
2. Assemble Language or Second GPL – considered as low level language uses mnemonic codes (*abbreviations that easy to remember*)
3. High Level Laguage or Third GPL – language is written in English like manner.
4. Very High Level Language or Fourth GPL
5. Natural Language – Fifth GPL resemblance to English language.

Software – Quiz

1. What is the difference between application software and system software?
2. What is the difference between machine language and assembly language?
3. What is the difference between high-level language and assembly language?
4. What is the difference between assembler and compiler?
5. What are the advantages and disadvantages of assembly language programming?
6. Why is learning assembly language still important?

!

4.0