

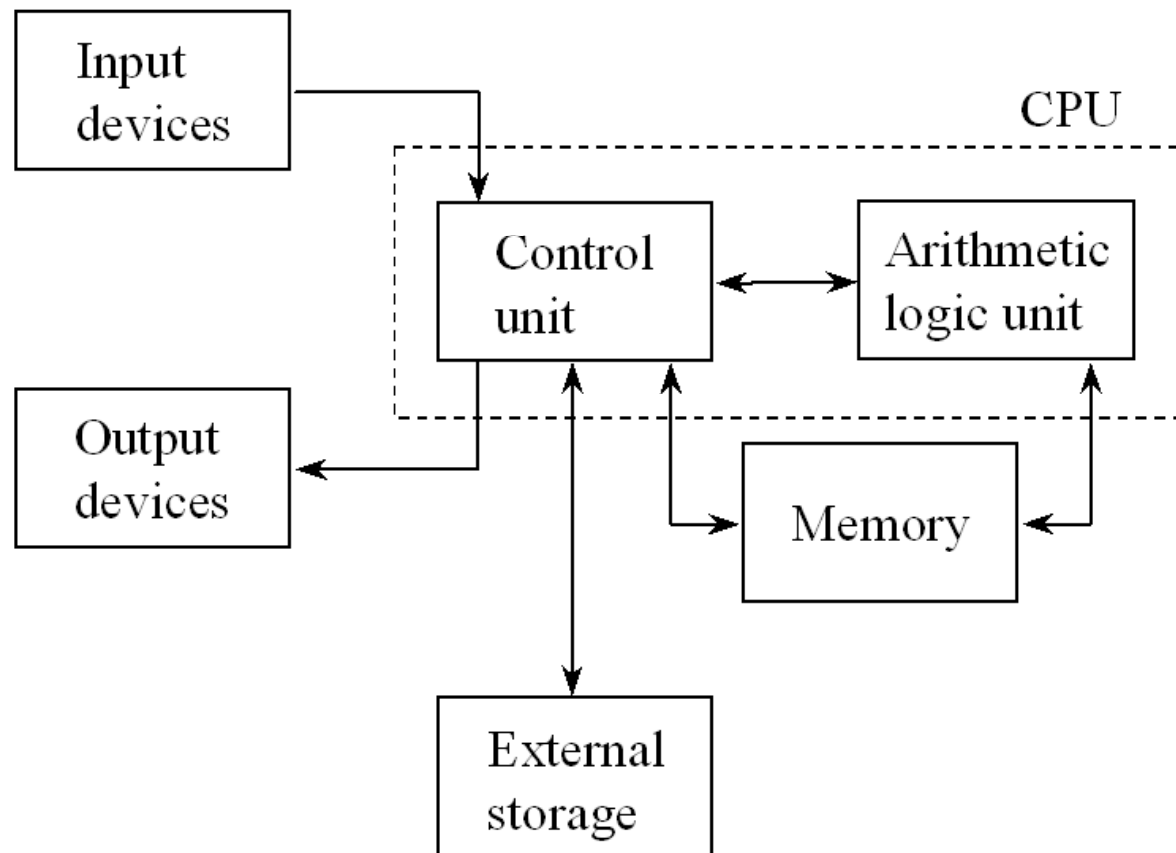
What is a Computer? ...

- Computer
 - Device capable of performing computations and making logical decisions.
 - Computers process data under the control of sets of instructions called computer programs
- Hardware
 - Various devices comprising a computer
 - Keyboard, screen, mouse, disks, memory, CD-ROM, and processing units
- Software
 - Programs that run on a computer

Computer Organization

- Six units in every computer:
 - Input unit
 - Output unit
 - Memory unit
 - Secondary storage unit
 - Arithmetic Logic Unit (ALU)
 - Control Unit

Computer Organization ...



Computer Organization ...

- **The memory unit - or random access memory (RAM)**
 - stores instructions and/or data
 - Memory is divided into an array of "boxes" each containing a *byte* of information.
 - A *byte* consists of 8 *bits*.
 - A *bit* (binary digit) is either 0 (OFF) or 1 (ON).
 - The memory unit also serves as a storage for intermediate and final results of arithmetic operations.
- **Secondary storage unit**
 - Cheap and high-capacity storageStores inactive programs

Computer Organization ...

- Bit 1 Bit (0 or 1)
- Byte 8 Bits (2^8)
- Word 16 Bits (2^{16})
- Double 32 Bits (2^{32})
- Long Double 64 Bits (2^{64})

Computer Organization ...

- **Input unit**
 - Obtains information from input devices (keyboard, mouse)
- **Output unit**
 - Outputs information (to screen, to printer, to control other devices)
- *input* (e.g. keyboard, mouse, microphone, disk drive, etc.) and *output* (e.g. monitor, status indicator lights, speakers, disk drive, etc.) units are used to transmit data into and out of the computer.
- Today there are generally 2 ways of describing data transfer speeds: in *bits per second*, or in *bytes per second*. Network engineers still describe network speeds in bits per second, while your internet browser would usually measure a file download rate in bytes per second. A lowercase "b" usually means a bit, while an uppercase "B" represents a byte.

Computer Organization ...

- a **central processing unit (CPU)** consists of
 - an *arithmetic/logic unit (ALU)* where **math and logic operations** are performed,
 - a *control unit* which directs most operations by providing **timing and control signals**,
 - and *registers* that provide **short-term data storage** and management facilities.
- **an arithmetic/logic unit (ALU)**
 - The type of operation that the ALU needs to perform is determined by **signals** from the *control unit*.
 - The **data** can come either from the **input unit** or from the **memory unit**.
 - **Results of the operation** can either be transferred back to the **memory unit** or directly to the **output unit**.

Computer Organization ...

- **control unit**
 - contains **logic and timing circuits** that generate the **appropriate signals** necessary to execute each instruction in a program
 - It **fetches** an instruction from memory by sending **an address** and a **read command** to the **memory unit**.
 - After decoding this instruction, the control unit transmits the appropriate signals to the other units in order to **execute** the specified operation.
 - This sequence of **fetch** and **execute** is repeated by the **control unit** until the computer is either **powered** off or **reset**.

Programming Language

- A *programming language* is an **artificial language** designed to express **computations** that can be performed by a machine, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine, to express algorithms precisely, or as a mode of human communication.
- Many programming languages have some form of written specification of their **syntax** (**form**) and **semantics** (**meaning**). Some languages are defined by a specification document. For example, the **C** programming language is specified by an **ISO** Standard. Other languages, such as **Perl**, have a **dominant implementation** that is used as a reference.

Programming Language ...

- Evolution of Programming Languages:
 - First Generation: *Machine languages*
 - **Strings of numbers** giving machine specific instructions
 - Example:
1300042774
1400593419
1200274027
 - *Computer only understands machine language instructions.*

Programming Language ...

- Second Generation: Assembly languages
 - **English-like abbreviations** representing elementary computer operations (translated via assemblers)

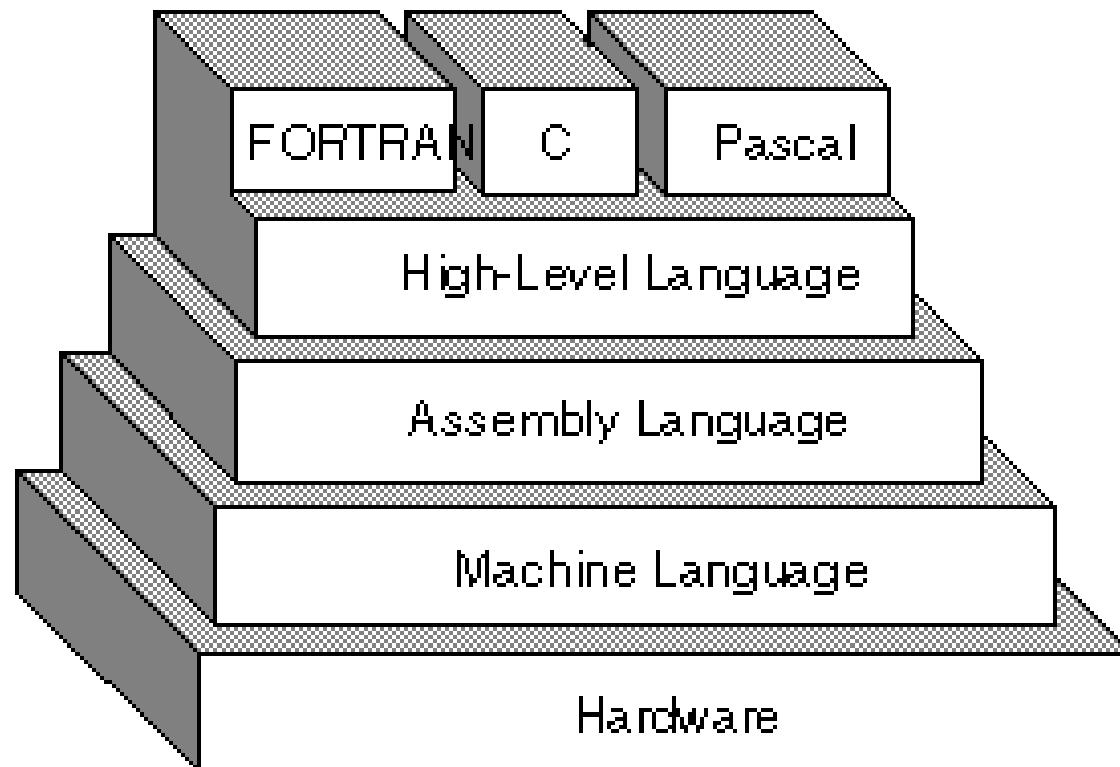
- Example:

LOAD	BASEPAY
ADD	OVERPAY
STORE	GROSSPAY

Programming Language ...

- Third Generation : High-level languages
 - Codes similar to **everyday English**
 - Use **mathematical notations** (translated via compilers)
 - Example: `grossPay = basePay + overPay`

Programming Language ...



History of C

- 1967 BCPL by Martin Richards as a language for writing operating-systems software and compilers
- 1970 B by Ken Thompson to create early versions of the UNIX operating system at Bell Laboratories
- Both BCPL and B were “typeless” languages
- 1972 C by Dennis Ritchie at Bell Laboratories and was widely known as the development language of the UNIX operating system.

History of C

- 1978 *traditional C* by Kernighan and Ritchie's book, *The C Programming Language*
- 1989 *the C standard*
- 1999 C99 : revised standard for the C

Common Software

- Operating System
- Assemblers
- Compilers
- Interpreters