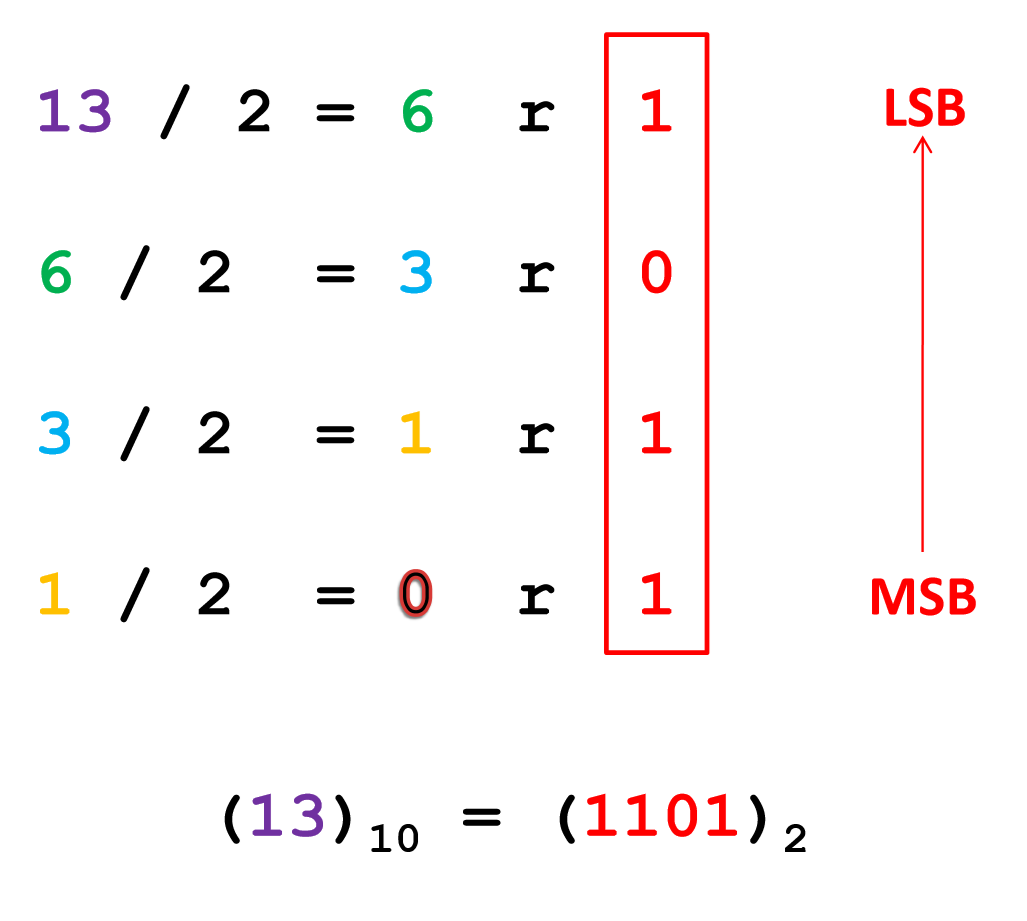
Addition note for 0.1

****

Logic gate

图片包含 文字

描述已自动生成

图片包含 文字

描述已自动生成图片包含 文字

描述已自动生成

AP CSP 0.1 note

# Unit0 Intro

# How computer work (hardware/software/Program language)

### Hardware and software work together in computer systems to provide computerized functionality.

### Hardware

# CPU: central processing unit. The CPU performs basic arithmetic, logic, controlling, and input/output (I/O) operations specified by the instructions.

# Two main type of CPU ISA (Instruction set architecture): complex instruction set computer (CISC), reduced instruction set computer (RISC).

# Two main ISA: ARM(Advanced RISC Machine),X86-64(base on  [Intel 8086](https://en.wikipedia.org/wiki/Intel_8086) and AMD64)(CISC)

# ARM is main using in phone/table/smart tv, with main manufacturer Qualcomm, MediaTek, Apple, Samsung and Huawei.

# X86-64 is mainly using in computer and server, with main manufacturer Intel, AMD.

# Memory: Random-access memory (RAM)

# A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, DVD and the magnetic tapes, the time required to read and write data items varies significantly depending on their physical locations on the recording medium.

# Two main type of memory: Dynamic random-access memory (DRAM)(volatile memory), Static random-access memory (SRAM)( non-volatile memory)

# DRAM is volatile memory (vs. non-volatile memory), that requires power to maintain the stored information; it retains its contents while powered on but when the power is interrupted, the stored data is quickly lost.

# DIMM (dual in-line memory module) and SODIMM (small outline dual in-line memory module) is often use in computer with DIMM in desktop and SODIMM in laptop.

# DDR (double data rate) is the speed rate use for memory, today most device using DDR4, with Graphics card using DDR5/DDR6.

# Bit and Bytes:

# The bit(b) is a [basic unit](https://en.wikipedia.org/wiki/Units_of_information) of [information](https://en.wikipedia.org/wiki/Information) in [information theory](https://en.wikipedia.org/wiki/Information_theory), [computing](https://en.wikipedia.org/wiki/Computing), and digital [communications](https://en.wikipedia.org/wiki/Communication). The name is a [portmanteau](https://en.wikipedia.org/wiki/Portmanteau) of binary digit.

The **byte(B)** is a unit of digital information that most commonly consists of eight bits. Historically, the byte was the number of bits used to encode **a single character of text** in a computer and for this reason it is the **smallest addressable unit of memory** in many computer architectures.

# 

# Storage device:

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | capacity | Cost (USD) | latency | Volatile? | Random access? |
| RAM | Gigabytes(8Gb typically for home use,16-128 for professional, up to 2TB for Server) | 7-30/GB | 20-50ns | yes | yes |
| SSD(Solid-state drive) | typical 256GB-2TB | 100/TB | 20ns | no | yes |
| HDD (Hard drive) | Terabyte (1-20TB/HDD) | 25/TB | 100ns | no | no |
| Disk | typically9GB for DVD25/50GB for Bur-ray | 1/DVD2-5/BD | Too long | no | no |
| Tape | typically, 1-20TB | 5/TB | Too long | no | no |
| Music | 1MB-20MB |  |  |  |  |
| Video | 100MB/h-10GB/h |  |  |  |  |
| Video Game | 100MB-100GB |  |  |  |  |

# I/O: input/output device

# Input: mouse, keyboard, controller, camera, microphone, scanner, etc.

# Output: monitor, printer, speaker, etc.

# I/O port: hardware interface for computer.

# Digital or analog:

# Connectors: physical interface. Ex: USB-A, micro USB, USB-C, Lighting, RJ45.

# Protocol: underlay communication rules. Ex:USB,DP,thunderbolt.

# USB(Universal Serial Bus): both connector and protocol.

# 

# Eternet(RJ45): for networking, speed 10/100/1000Mb.

# In refence, WiFi with 802.11n is 150Mb/s/antenna, while 802.11ac is 433mb/s/ antenna

### Software

# Operating System: operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

# UNIX is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, development starting in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.

# The Berkeley Software Distribution (BSD) was an operating system based on Research Unix, developed and distributed by the Computer Systems Research Group (CSRG) at the University of California, Berkeley. Today, "BSD" often refers to its descendants, such as FreeBSD, OpenBSD, NetBSD, or DragonFly BSD.

# Linux is a family of open source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds.

# Windows NT is a family of [operating systems](https://en.wikipedia.org/wiki/Operating_system) produced by [Microsoft](https://en.wikipedia.org/wiki/Microsoft), the first version of which was released on July 27, 1993. It is a processor-independent, [multiprocessing](https://en.wikipedia.org/wiki/Multiprocessing) and [multi-user](https://en.wikipedia.org/wiki/Multi-user) operating system.

# macOS is a series of [non-free](https://en.wikipedia.org/wiki/Proprietary_software) [graphical](https://en.wikipedia.org/wiki/Graphical_user_interface) [operating systems](https://en.wikipedia.org/wiki/Operating_system) developed and marketed by [Apple Inc.](https://en.wikipedia.org/wiki/Apple_Inc.) since 2001.

# An application programming interface (API) is an interface or communication protocol between different parts of a computer program intended to simplify the implementation and maintenance of software.

# An API may be for a web-based system, operating system, database system, computer hardware, or software library.

# Middleware

# Middleware is computer software that provides services to software applications beyond those available from the operating system. It can be described as "software glue".

### Application

Application software (app for short) is a program or group of programs designed for end users. Examples of an application include a word processor, a spreadsheet, an accounting application, a web browser, an email client, a media player, a file viewer, an aeronautical flight simulator, a console game or a photo editor.

Most programmer make application.

### Programming language

# Formal Language vs Informal Language (Natural Language)

# In mathematics, computer science, and linguistics, a formal language consists of words whose letters are taken from an alphabet and are well-formed according to a specific set of rules.

# The alphabet of a formal language consists of symbols, letters, or tokens that concatenate into strings of the language. Each string concatenated from symbols of this alphabet is called a word, and the words that belong to a particular formal language are sometimes called well-formed words or well-formed formulas. A formal language is often defined by means of a formal grammar such as a regular grammar or context-free grammar, which consists of its formation rules.

# All programing language is formal language

# In [neuropsychology](https://en.wikipedia.org/wiki/Neuropsychology), [linguistics](https://en.wikipedia.org/wiki/Linguistics), and the [philosophy of language](https://en.wikipedia.org/wiki/Philosophy_of_language), a natural language or ordinary language is any [language](https://en.wikipedia.org/wiki/Language) that has [evolved](https://en.wikipedia.org/wiki/Linguistic_evolution) naturally in [humans](https://en.wikipedia.org/wiki/Human) through use and repetition without conscious planning or premeditation. Natural languages can take different forms, such as [speech](https://en.wikipedia.org/wiki/Vocal_language) or [signing](https://en.wikipedia.org/wiki/Language_interpretation). They are distinguished from [constructed](https://en.wikipedia.org/wiki/Constructed_language) and [formal languages](https://en.wikipedia.org/wiki/Formal_language) such as [those used to program computers](https://en.wikipedia.org/wiki/Programming_language) or to study [logic](https://en.wikipedia.org/wiki/Logic).

# Level of language

# 图片包含 文字 描述已自动生成

# Programing language at top is hard for human to understand but more efficient.

# Programing language at button is easy for human to understand but more costly.

# Types

# In [computer science](https://en.wikipedia.org/wiki/Computer_science), a high-level programming language is a [programming language](https://en.wikipedia.org/wiki/Programming_language) with strong [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) from the details of the [computer](https://en.wikipedia.org/wiki/Computer). Ex: C, java, python

# A low-level programming language is a [programming language](https://en.wikipedia.org/wiki/Programming_language) that provides little or no [abstraction](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) from a computer's [instruction set architecture](https://en.wikipedia.org/wiki/Instruction_set_architecture)—commands or functions in the language map closely to processor instructions. Ex: Assembly

# In [computer software](https://en.wikipedia.org/wiki/Computer_software), a general-purpose programming language is a [programming language](https://en.wikipedia.org/wiki/Programming_language) designed to be used for writing [software](https://en.wikipedia.org/wiki/Software) in the widest variety of [application domains](https://en.wikipedia.org/wiki/Application_domain) (a [general-purpose language](https://en.wikipedia.org/wiki/General-purpose_language)).

# An interpreted language is a type of programming language for which most of its implementations execute instructions directly and freely, without previously compiling a program into machine-language instructions. The interpreter executes the program directly, translating each statement into a sequence of one or more subroutines, and then into another language (often machine code).

# A compiled language is a programming language whose implementations are typically compilers (translators that generate machine code from source code), and not interpreters (step-by-step executors of source code, where no pre-runtime translation takes place).

# Syntax (form) and Semantics (meaning)

# In computer science, the syntax of a computer language is the set of rules that defines the combinations of symbols that are considered to be a correctly structured document or fragment in that language. This applies both to programming languages, where the document represents source code, and to markup languages, where the document represents data.

# In [programming language theory](https://en.wikipedia.org/wiki/Programming_language_theory), semantics is the field concerned with the rigorous mathematical study of the meaning of [programming languages](https://en.wikipedia.org/wiki/Programming_language). It does so by evaluating the meaning of [syntactically](https://en.wikipedia.org/wiki/Programming_language_syntax) valid [strings](https://en.wikipedia.org/wiki/String_(computer_science)) defined by a specific programming language, showing the computation involved.