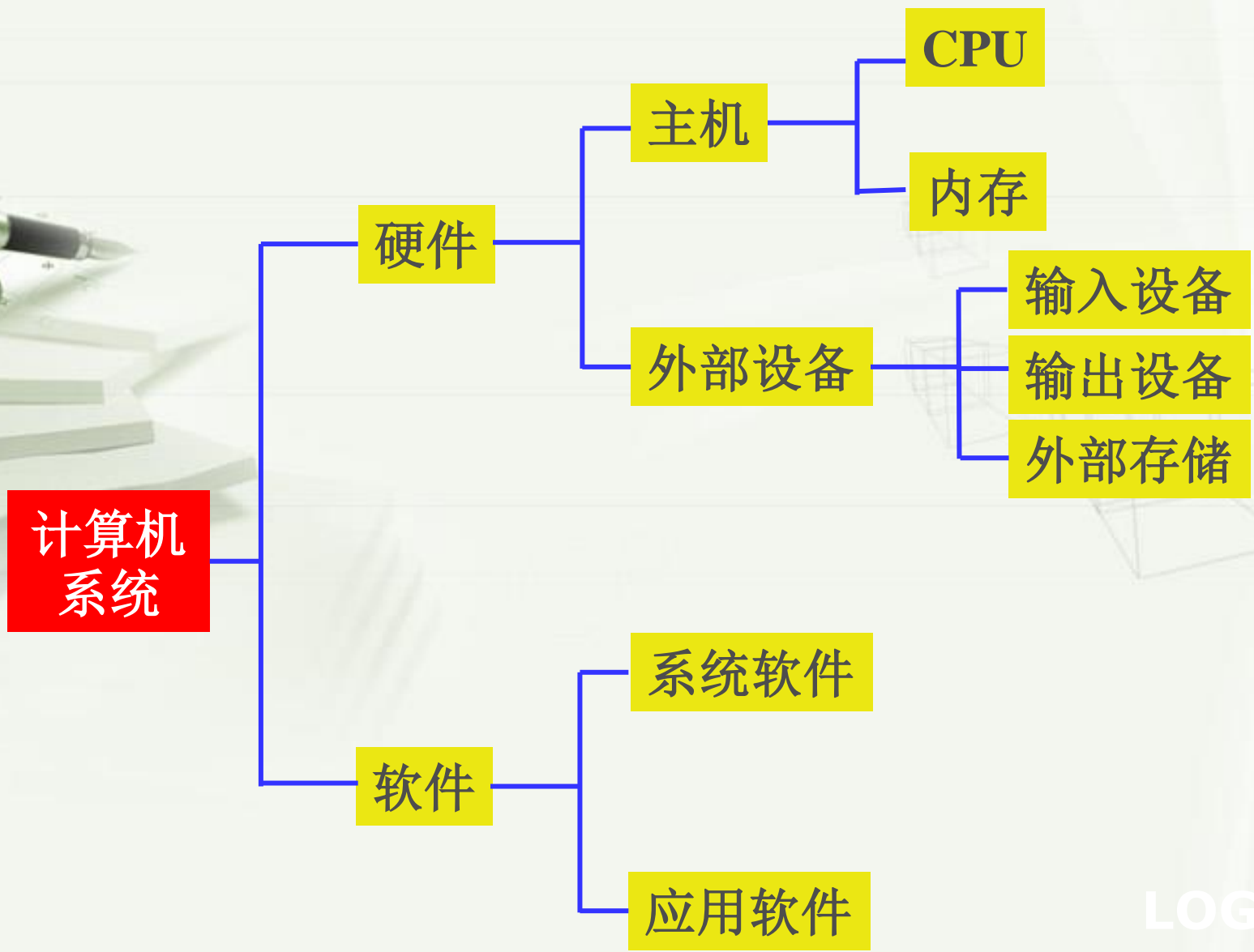
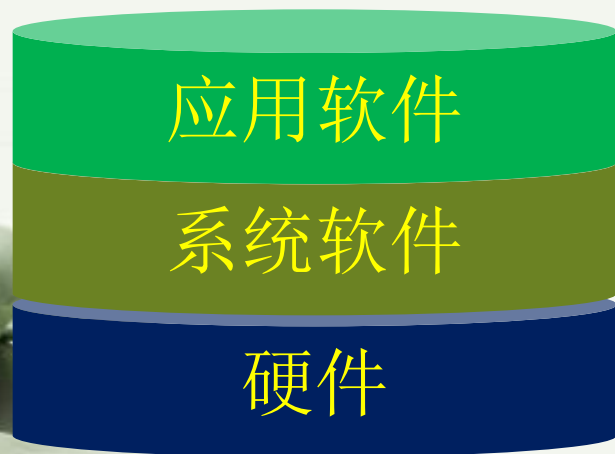


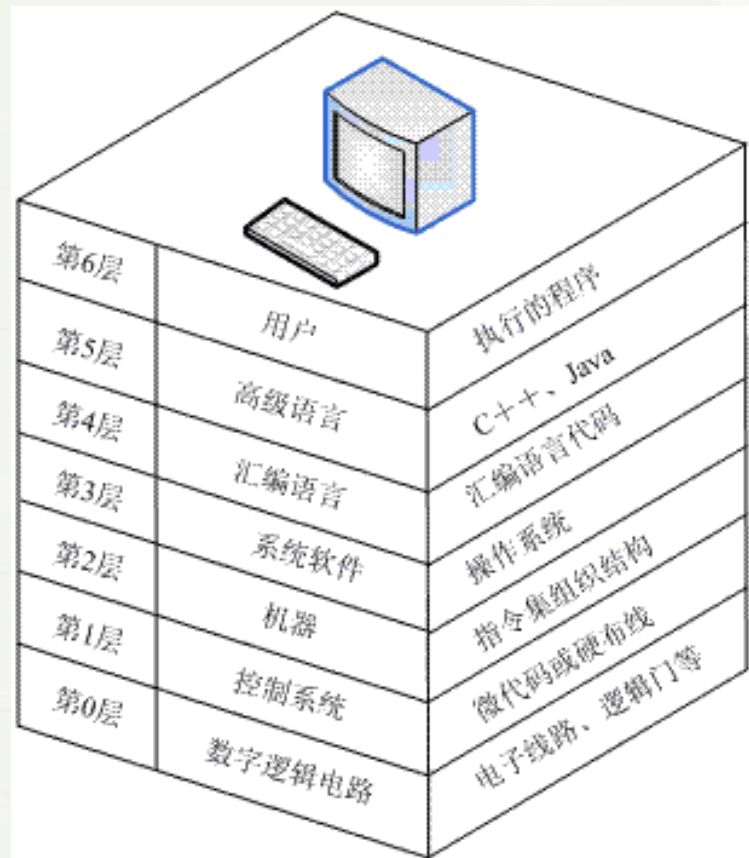
# ■ 计算机系统的组成



# ■ 计算机系统的分层结构



从一般使用者的角度来看：  
计算机系统自下而上由三个层次构成，下层为上层功能的实现提供支持。



从计算机设计者的角度看：  
计算机系统可以划分为不同的层次来实现其功能。每一个层次都实现某项特定功能，并有一个特定的**假想机器**与之对应。这种假想机器称之为**虚拟机**。每一层的虚拟机都执行自己特有的指令集，必要时还可以调用较低层次的虚拟机来完成各种任务。

## ■ 计算机体系结构

- 计算机体系结构（**Computer Architecture**）是程序员所看到的计算机的属性，即概念性结构与功能特性。
- 按照计算机系统的多级层次结构，不同级程序员所看到的计算机具有不同的属性。一般来说，低级机器的属性对于高层机器程序员基本是透明的，
- 通常所说的计算机体系结构主要指**机器语言级机器的系统结构**。
- 广义的计算机体系结构还包括计算机组成和计算机实现：
  - （1）**计算机组成**：在体系结构的基础上设计计算机各个部件的组成及连接关系。
  - （2）**计算机实现**：对计算机组成进行物理（具体）实现。

# ■ 计算机体系结构

## 几个基本概念间的关系

- 体系结构、计算机组成、计算机实现



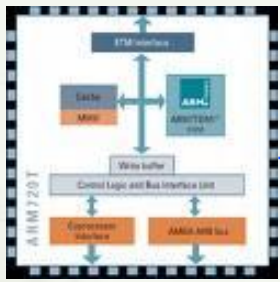
操作系统



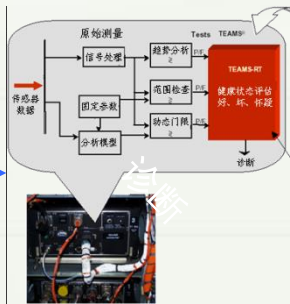
指令系统



应用程序



计算机体系结构



计算机组成



计算机实现

由图示可见，**体系结构**在整个计算机系统中占据核心地位

## ■ 与体系结构相关概念

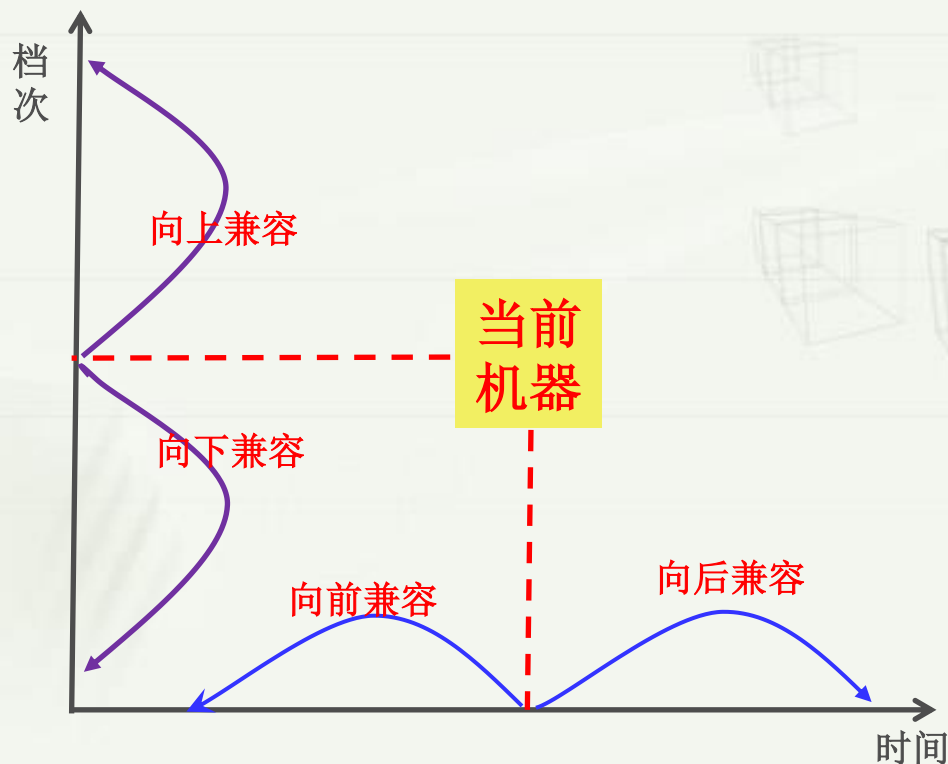
- **系列机**：同一个体系结构，可以有不同的计算机组成和实现。在一个厂家内生产的具有相同的体系结构，但具有不同组成和实现的一系列不同型号的机器。**性能和价格一般不同！**

**IBM 370系列计算机**



## ■ 与体系结构相关概念

- **软件兼容：**系列机由于体系结构相同，从程序员的角度看它们具有机器属性，软件都能够通用于这一系列机，称为各机器的软件兼容。



向前兼容和向下兼容一般不需，但向上兼容和向后兼容是需要的，特别是向后兼容。



## ■ 与体系结构相关概念

- 兼容机，不同厂家生产的具有相同体系结构的计算机称为兼容机。

### 一些典型的兼容机



# 第1章 计算机硬件结构

1

冯·诺依曼  
结构和哈  
佛结构

2

中央处理单  
元

3

存储设备

4

输入/输出  
设备

5

计算机总  
线与接口



# 第一节 冯·诺依曼结构和哈佛结构

## 1. 冯·诺依曼理论

冯·诺依曼理论的思想是：应该把程序和数据一起存储在存储器里，让计算机自己负责从存储器里提取指令，执行指令，循环式地执行这两个动作。这样，计算机在执行程序的过程中，就可以完全摆脱外界的影响，以自己可能的速度自动地运行。按照这种原理构造出来的计算机就是“存储程序控制计算机”，也被称做“冯·诺依曼计算机”——存储程序工作原理。



# 第一节 冯·诺依曼结构和哈佛结构

## 2. 冯·诺依曼理论的要点

- 指令像数据那样存放在存储器中，并可以像数据那样进行处理；
- 指令格式使用二进制机器码表示；

十进制  
→ 计算  
机指令



冯·诺依曼提出二  
进制机器码表示  
计算机指令格式



- 使用程序存储控制方式工作。

# 第一节 冯·诺依曼结构和哈佛结构

## 3. 冯·诺依曼结构

冯·诺依曼结构又称为普林斯顿体系结构（**Princeton Architecture**）。这是现代通用计算机使用的最主要结构

### ➤ 冯·诺依曼结构的主要特点：

- 指令与数据均是用二进制代码形式表现，电子线路采用二进制
- 存储器中的指令与数据形式一致，机器对它们同等对待，不加区分
- 指令在存储器中按执行顺序存储，并使用一个指令计数器来控制指令执行的方向，实现顺序执行或转移
- 存储器的结构是按地址访问的顺序线性编址的一维结构
- 计算机由五大部分组成：运算器、控制器、存储器、输入设备、输出设备
- 一个字长的各位同时进行处理，即在运算器中是并行的字处理
- 运算器的基础是加法器
- 指令由操作码和地址码两部分组成；操作码确定操作的类型，地址码指明操作数据存储的地址

# 第一节 冯·诺依曼结构和哈佛结构

## 4. 冯·诺依曼计算机

冯·诺依曼计算机— Testability Engineering System:

运算器 (ALU, Arithmetic Logic Unit)

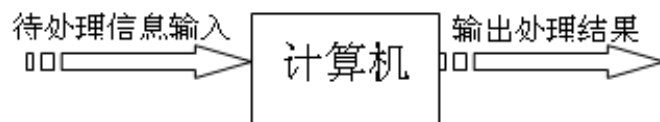
控制器 (Controler)

存储器 (Memory)

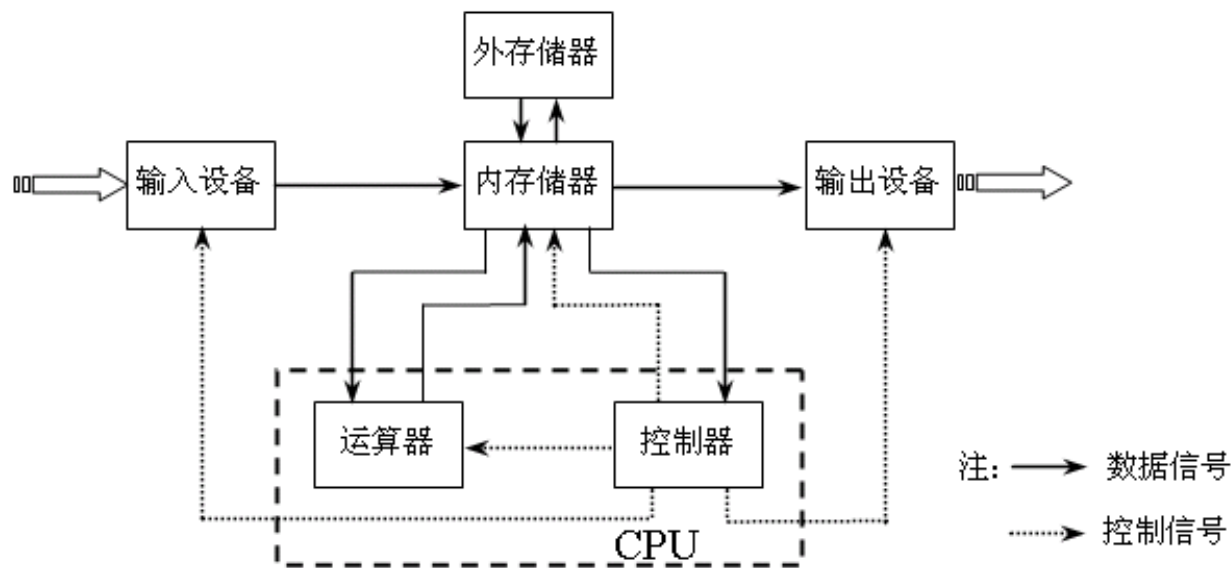
输入设备 (Input Unit)

输出设备 (Output Unit)

# 第一节 冯·诺依曼结构和哈佛结构



(a) 计算机的功能



(b) 计算机的构成

典型的冯·诺依曼计算机

# 第一节 冯·诺依曼结构和哈佛结构

## 5. 冯·诺依曼结构的缺陷

- 处理器要频繁访问存储器，指令串行执行效率低
- 高级语言与机器语言之间转换不便
- 采用的是按地址访问的顺序存储空间，对于复杂的数据结构，必须经过地址映像存放才能解决问题，因此也带来了。

## 6. 冯·诺依曼结构的演变

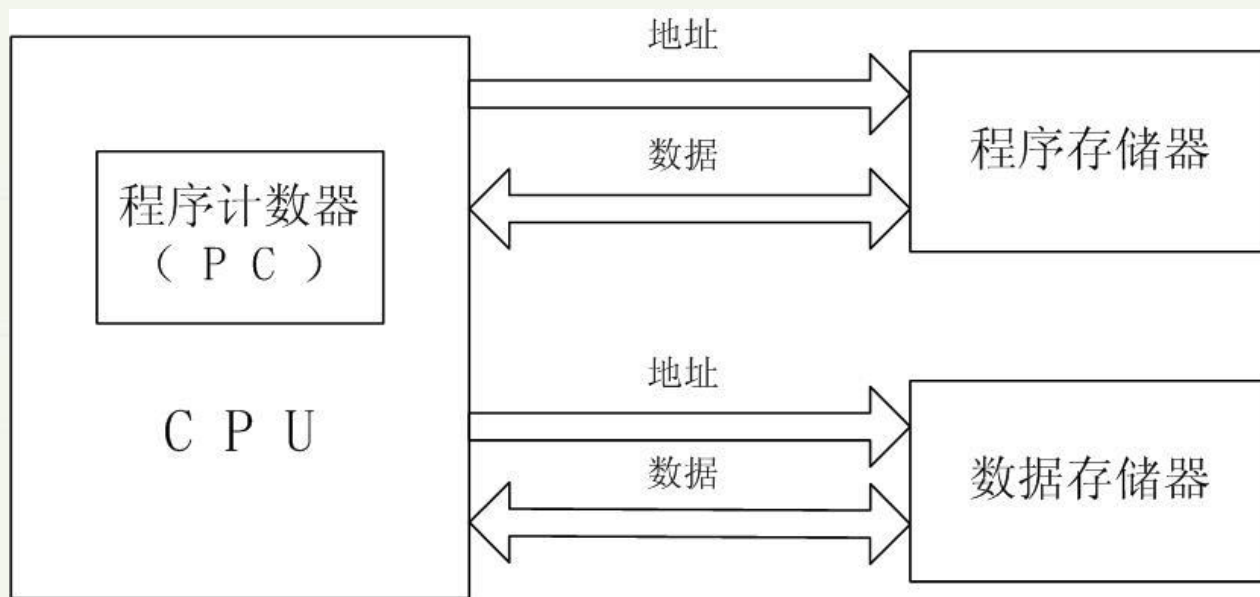
- 将运算器与控制器集中于一块芯片，作为中央处理器件（**CPU**）
- 采用先行控制、流水线等方法，开发并行性
- 采用多体交叉存储器，增加存储带宽
- 采用总线结构，将计算机各个部件连接起来，并实现各部件之间正确的数据传输
- 以存储器为核心，使**I/O**设备和处理器可并行工作



# 第一节 冯·诺依曼结构和哈佛结构

## 7. 哈佛结构

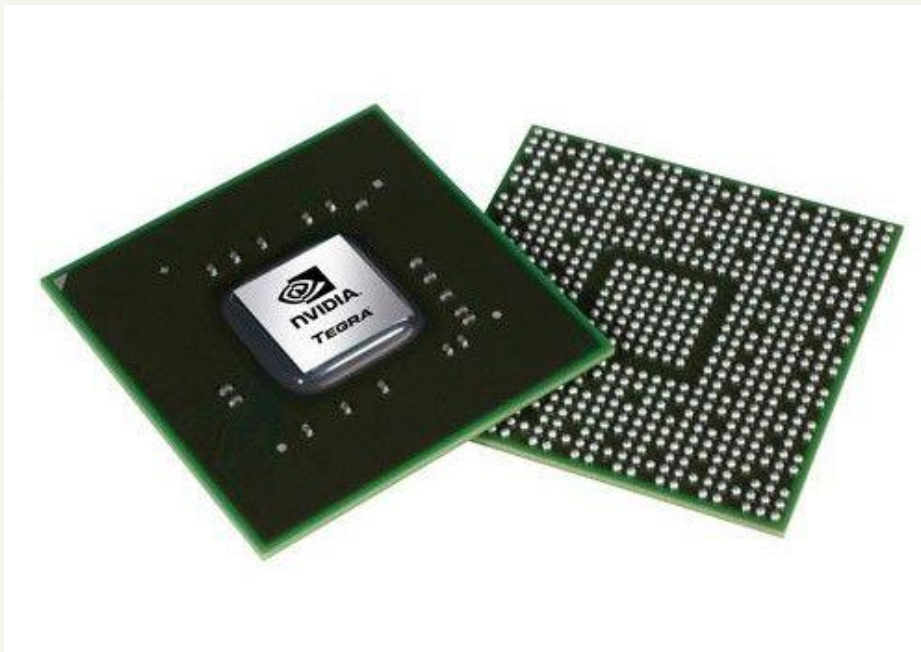
Harvard architecture, 缩写为HARC, 由哈佛学者提出。这是一种将程序指令存储和数据存储分开的存储器结构。



它的主要特点是将程序和数据存储在不同的存储空间中，即程序存储器和数据存储器是两个独立的存储器，每个存储器独立编址、独立访问，目的是为了减轻程序运行时的访存瓶颈。

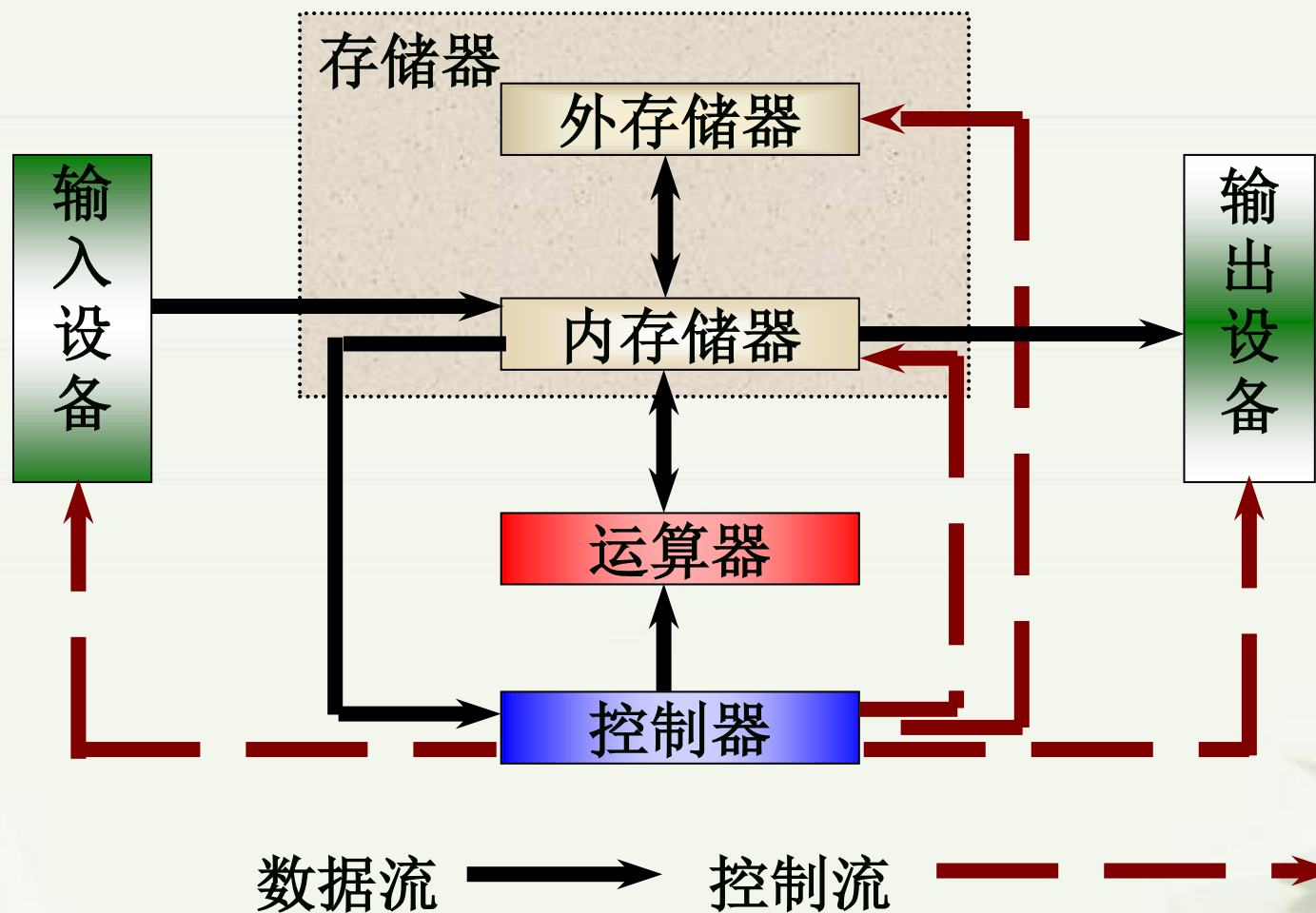
# 第一节 冯·诺依曼结构和哈佛结构

目前使用哈佛结构的中央处理器和微控制器有很多，如**Microchip**公司的**PIC**系列芯片、摩托罗拉公司的**MC68**系列、**Zilog**公司的**Z8**系列、**ATMEL**公司的**AVR**系列和**ARM**公司的**ARM9**、**ARM10**和**ARM11**等。



# 第一节 冯·诺依曼结构和哈佛结构

计算机硬件系统组成:



# 第一节 冯·诺依曼结构和哈佛结构

- **存储器：**存储器是用来存放数据和程序的部件。
- **运算器：**对信息进行运算处理的部件。主要功能是对二进制数码进行 算术（加减乘除）和逻辑（与或非）运算。
- **控制器：**整个计算机的控制核心。主要功能是读取指令、翻译指令、并向计算机各部分发出控制信号，以便执行指令。
- **输入设备：**将数据和程序转换成计算机能够识别和接受的信息，并顺序地把它们送入存储器中。输入设备有许多种，例如键盘、鼠标、扫描仪、光电输入机等。
- **输出设备：**输出设备将计算机处理的结果以人们能接受的或其它机器能接受的形式送出。输出设备有许多种类，例如显示器、打印机、绘图仪等。

各个部分通过总线和接口连接在一起。

# 第一节 冯·诺依曼结构和哈佛结构

◆ **总线（Bus）**：是计算机中用于传送信息的公用通道，是为多个部件服务的一组信息传送连接线。

◆ 总线的分类：

（1）按传送信息的属性分：

数据总线（**DB**）——在各部件之间传送数据信息

地址总线（**AB**）——传送各部件相互访问的地址信息

控制总线（**CB**）——传送控制各部件操作的控制信息

（2）按信息传送的方向分：

单向总线

双向总线

（3）按连接对象和完成的功能分：

**I/O BUS** ——用于外设与主机的信息交换。

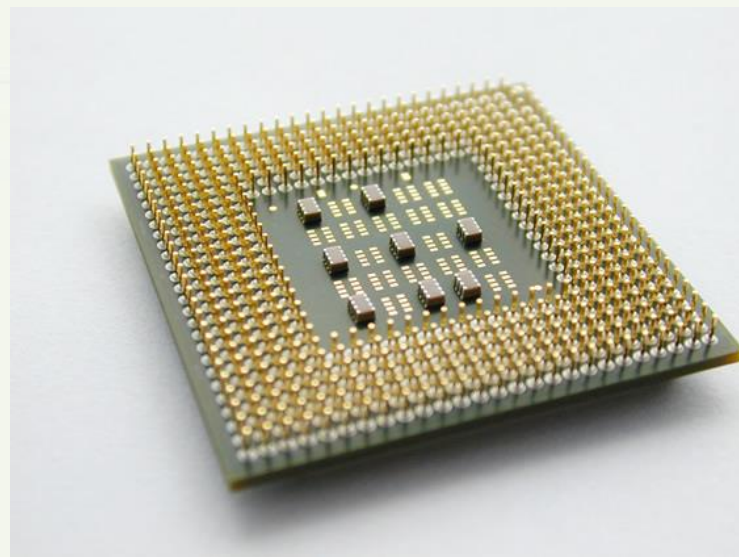
**MEMORY BUS** ——用于**CPU**与主存储器的信息交换



## 第二节 中央处理器（CPU）

### ◆ 中央处理器（CPU）

中央处理器也叫中央处理单元（**Central Processing Unit**，简称**CPU**），是计算机的核心部件。从体系结构上看，**CPU**包含了**运算器**和**控制器**，以及为保证它们高速运行所需的**寄存器**（**Register**）。





## 第二节 中央处理器（CPU）

### 1. 运算器

又称作算术逻辑单元（**Arithmetic Logic Unit, ALU**），是执行各种算术和逻辑运算操作的部件。是中央处理机的重要组成部分。

- 运算器的功能

运算器的主要功能是实现数据的算术和逻辑运算。主要包括对数值数据的算术运算，如执行加、减、乘、除运算，变更数据的符号等。也包括对各种数据的逻辑运算，例如进行与、或、求反等运算。

## 第二节 中央处理器（CPU）

### 1. 运算器

- 运算器的组成

运算器的基本结构应包括以下几个部分：

- （1）能实现算术和逻辑运算功能的部件**ALU**；
- （2）存放待加工的信息或结果信息的通用寄存器组；
- （3）按操作要求控制数据输入部件：多路开关或数据锁存器
- （4）按操作要求控制数据输出部件：输出移位和多路开关；
- （5）计算器与其它部件进行信息传送的总线以及 总线接收器与发送器；总线接收器与发送器通常是由三态门构成的。

- 运算器工作原理：计算机组成原理等其他课程会深入学习。

## 第二节 中央处理器（CPU）

### 1. 运算器

- 运算器与计算机其他部件的关系

运算器与其它部件有着紧密的联系。这种联系影响信息的传送和加工的速度。

- 与控制器的关系

运算器接收到控制器发来的各种运算控制命令，进行运算，运算过程中产生的各种信息，包括运算结果特征标志和状态信息，再反馈给控制器

- 与存储器的关系

存储器可以把参加运算的数据传送给运算器；运算器也可把运算结果传送给存储器，同时运算器提供存储器的地址。

## 第二节 中央处理器（CPU）

### 2. 控制器

**控制器**是中央处理器的重要组成部分，是整个系统的指挥中心，在控制器的控制之下，运算器、存储器和输入输出设备等部件构成了一个有机的整体。

- **控制器的功能：**

- 控制器的功能（基本功能）

控制器的基本功能就是负责指令的读出，进行识别和解释，并指挥协调各功能部件执行指令。

## 第二节 中央处理器（CPU）

### 2. 控制器

-控制器的具体功能

包括以下几个部分：

#### （1）程序控制

保证机器按一定顺序执行程序是控制器的首要任务；

#### （2）操作控制

一条指令的功能往往是由若干个操作信号的组合来实现的；

#### （3）时间控制

对各种操作实施时间上的控制称为时间控制。各种指令的操作信号均受到时间的严格控制；一条指令的整个执行过程也受到时间的严格控制；

#### （4）数据加工

## 第二节 中央处理器（CPU）

### 2. 控制器

- 控制器的组成

- （1）指令部件

- 程序计数器(**PC**)、指令寄存器(**IR**)、指令译码器 (**ID**)、程序状态寄存器 (**PSW**) 和地址形成部件等。

- （2）时序部件

- 时序部件就是用来产生各部件所需要的定时控制信号的部件。时序信号一般由工作周期，工作节拍及工作时标脉冲三级时序信号构成。

- （3）微操作控制线路

- 微操作是指计算机中最基本的操作；微操作控制逻辑，用来产生机器所需的全部的微操作信号。微操作控制逻辑的作用是把操作码译码器输出的控制电位，时序信号以及各种控制条件进行组合，按一定时间顺序产生并发出一系列微操作控制信号，以完成指令规定的全部操作。

- （4）中断控制逻辑

- 中断控制逻辑用来控制中断处理的硬件逻辑。

- 控制器的工作原理：在计算机组成原理等其他课程中学习



## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 指令系统 — Computer Master

**指令**是指控制计算机执行某种操作的命令，也称为**机器指令**。指令的作用是协调各硬件部件之间的工作关系，它反映了计算机所拥有的基本功能，是计算机运行的最小功能单位。一台计算机中所有机器指令的集合，称为这台计算机的**指令集**，或者**指令系统**。

#### ■ 指令系统的特征：

- 指令格式
- 指令长度
- 指令操作码在处理器内的存储形式
- 指令周期
- 指令类型
- 指令系统支持的寻址方式

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

根据指令系统功能结构的不同，处理器体系结构发展趋势呈现两种截然不同的方向：

- ◆ 复杂指令集计算技术（**CISC**）

- **Complex Instruction Set Computing**

- ◆ 精简指令集计算技术（**RISC**）

- **Reduced Instruction Set Computing**

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

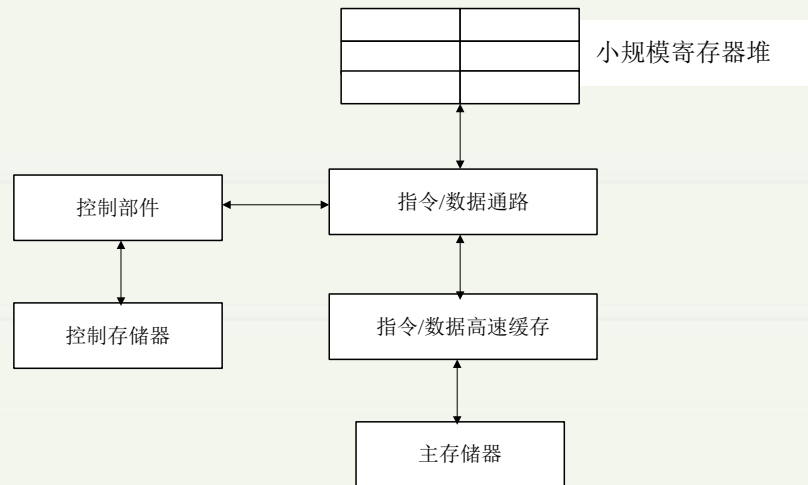
#### （1）复杂指令集计算技术的特点

**A.** 指令愈丰富功能愈强，编译程序愈好写，指令效率愈高；

**B.** 指令系统愈丰富，愈可减轻软件危机；

**C.** 指令系统丰富，尤其是存储器操作指令的增多，可以改善系统结构的质量；

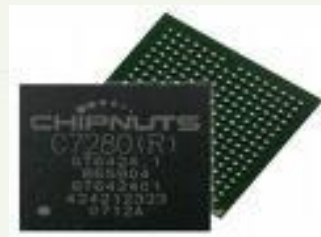
**D.** 以微程序控制器为核心，指令存储器与数据存储器共享同一个物理存储空间。



典型CISC处理器结构

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构



一些典型的**CISC**结构处理器

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

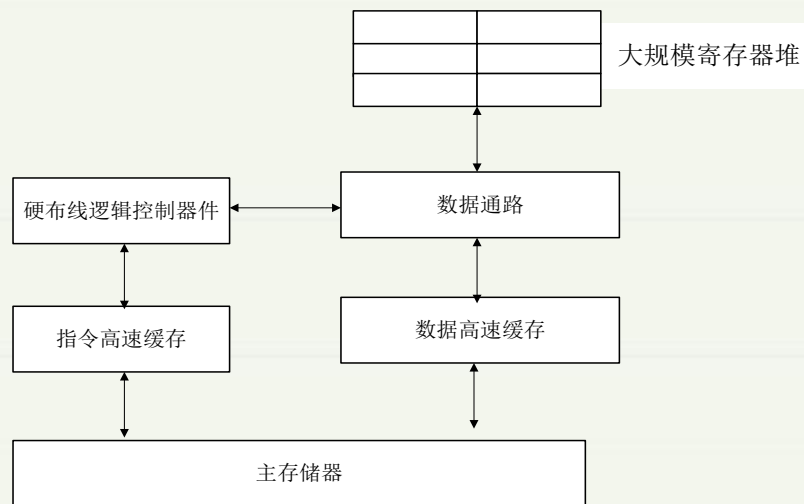
#### (2) 精简指令集计算技术的特点

**A. 精简指令系统。**可以通过对过去大量的机器语言程序进行指令使用频度的统计，来选取其中常用的基本指令；

**B. 减少指令系统可采用的寻址方式种类，**一般限制在**2或3种**；

**C. 在指令的功能、格式和编码设计上**尽可能地简化和规整，让所有指令尽可能等长；

**D. 单机器周期指令，**即大多数的指令都可以在一个机器周期内完成，并且允许处理器在同一时间内执行一系列的指令。



典型**RISC**处理器结构

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构



一些典型的**RISC**结构处理器



## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 流水线技术 — Pipeline Tecniche

**流水线技术**允许计算机在一个机器周期内的各步骤操作重叠进行。特别是，当执行一条指令时，可以读取下一条指令，也就意味着，在任何一个时刻可以有不止一条指令在“流水线”上，每条指令处在不同的执行阶段。这样，即便读取和执行每条指令的时间保持不变，计算机的总的吞吐量却提高了。

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 流水技术特点：

- 流水线可分成若干个互有联系的子过程（功能段）；
- 实现子过程的功能段所需时间尽可能相等，避免因不等而产生处理的瓶颈，形成流水线的断流；
- 形成流水处理，需要一段准备时间，称“通过时间”。只有在此之后流水过程才稳定；
- 指令流发生不能顺序执行时，会使流水过程中断，再形成流水过程，则需要“通过时间”，所以流水过程不应常断流，否则效率就不会很高。

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 流水线技术工作原理：

##### -流水线的原理

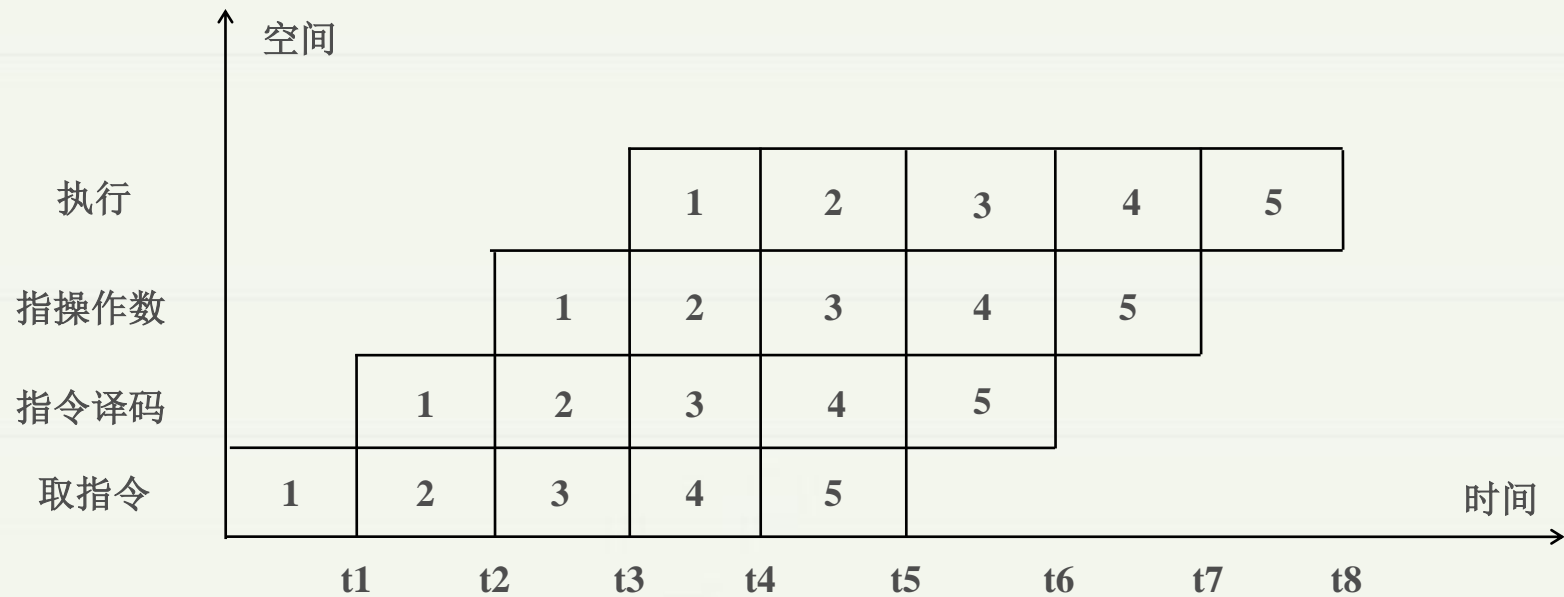
实现流水操作的基本思想是：流水线技术将一个重复的时序过程分解成若干个子过程，每一个子过程都可有效地在其专用功能段上与其它子过程同时执行。

##### -流水线工作过程描述

流水线工作过程的描述：采用时(间)-空(间)图的方法。描述的是指令执行流水线的时-空图。

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构



流水线技术原理

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 并行处理结构：

自**80**年代来由于计算机电路的操作速度最终取决于光速，而现在的许多电路已运行在纳秒级上，因此单处理器体系结构的发展正在接近极限。为了超越单处理器的性能，一种合乎逻辑的方法就是把多个微处理器联结起来，形成并行计算机。

并行计算机是由一组处理单元组成的。这组处理单元通过相互之间的通信与协作，以更快的速度共同完成一项大规模的计算任务。

并行计算机两个主要的组成部分：

**(1)** 计算节点

**(2)** 节点间的通信与协作机制

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 并行处理的概念：

在同一时刻或是在同一时间间隔内完成两种或两种以上性质相同或不同的工作，它们在时间上能互相重叠，就称为**并行处理**。

-**并行处理**有两个不同特征：

- 同时性（**Simultaneity**）：两个或多个事件在同一时刻发生。
- 并发性（**Concurrency**）：两个或多个事件在同一时间间隔内发生。



## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### ■ 并行处理计算机：

- 流水线计算机
- 阵列处理机
- 多处理器系统

#### ■ 多处理器系统：

按照著名的弗林(Flynn) 计算机分类模型，根据计算机的指令和数据流的并行性，把所有的计算机分为四类：

- (1) SISD (Single Instruction Stream Single Data Stream)  
——单指令流单数据流
- (2) SIMD (Single Instruction Stream Multiple Data Stream)  
——单指令流多数据流
- (3) MISD (Multiple Instruction Stream Single Data Stream)  
——多指令流单数据流
- (4) MIMD (Multiple Instruction Stream Multiple Data Stream)  
——多指令流多数据流

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### （1）**SISD** -单指令流单数据流

计算机的指令部件每次只对一条指令进行译码和处理，并只对一个操作部分分配数据，是按照排序的方式进行顺序处理，也就是说通常由一个处理器和一个存贮器组成，它通过执行单一的指令流对单一的数据流进行操作。

传统的冯·诺依曼机均属此类。弱点是单片处理器处理能力有限，同时，这种结构也没有发挥数据处理中的并行性潜力，在实时系统或高速系统中，很少采用**SISD**结构。

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### （2）SIMD -单指令流多数据流

属于并行运算计算机，计算机有多个处理单元，由单一的指令部件控制，按照同一指令流的要求为它们分配各不相同的数据并进行处理。系统结构为由一个控制器、多个处理器、多个存贮模块和一个互连总线（网络）组成。所有“活动的”处理器在同一时刻执行同一条指令，但每个处理器执行这条指令时所用的数据是从它本身的存储模块中读取的。

早期的多处理器计算机多属于**SIMD**结构

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### （3）MISD -多指令流单数据流

**流水线处理计算机：**计算机具有多个处理单元，按照多条不同的指令要求同时对同一数据流及其处理输出的结果进行不同的处理，是把一个单元的输出作为另一个单元的输入。

**MISD**结构计算机只出现于早期自动控制系统中，目前已很少应用。

## 第二节 中央处理器（CPU）

### 3. 处理器的体系结构

#### （4）MIMD -多指令流多数据流

典型的**MIMD**系统由多台处理机、多个存储模块和一个互连网络组成，每台处理机执行自己的指令，操作数也是各取各的。**MIMD**结构中每个处理器都可以单独编程，因而这种结构的可编程能力是最强的。但由于要用大量的硬件资源解决可编程问题，硬件利用率不高。

目前，**MIMD**开始作为一种通用多处理器体系结构出现

## 第二节 中央处理器（CPU）

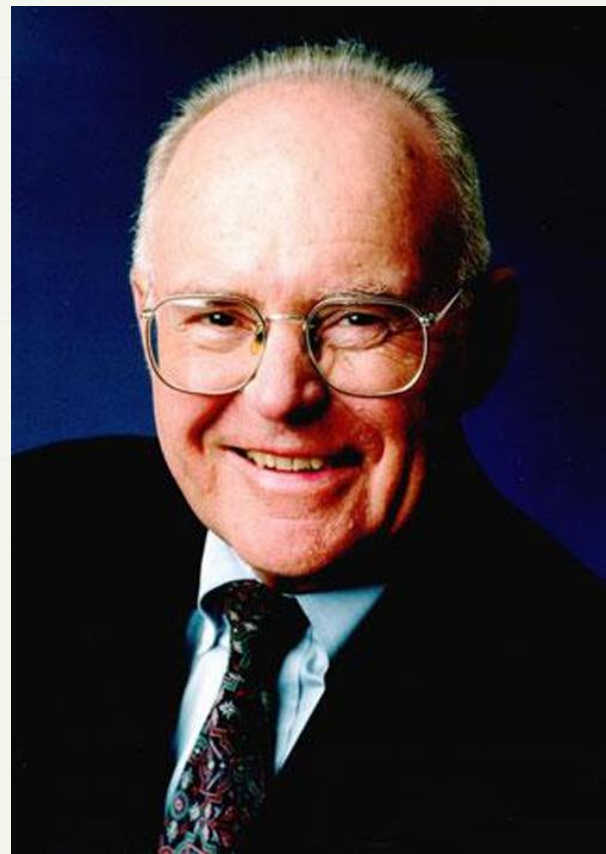
### 4. 摩尔定律与处理器的发展

#### ◆ 摩尔定律（**Moore law**）

该定律源于**1965**年戈登·摩尔（**Gordon Moore**，时任**Intel**公司名誉董事长）的一份关于计算机存储器发展趋势的报告。

摩尔定律归纳起来包括以下几点：

- （1）芯片技术的发展具有周期性，每个周期是**18-24**个月；
- （2）集成电路芯片上所集成的电路的数目，每隔一个周期就翻一番；
- （3）处理器的性能每隔一个周期提高一倍，并且价格同比下降一倍。





## 第二节 中央处理器（CPU）

### 5. CPU的主要性能指标

- **主频（外频，倍频）**：**CPU**内部的时钟频率，是**CPU**进行运算时的工作频率。一般来说，主频越高，一个时钟周期里完成的指令数也越多，**CPU**的运算速度也就越快。但由于内部结构不同，并非所有时钟频率相同的**CPU**性能一样。单位是**Hz**。
  - **CPU**的外频是其外部时钟频率，是指**CPU**到芯片组之间的总线速度。单位是**Hz**。
  - **CPU**的倍频是主频为外频的倍数。原先并没有倍频概念，**CPU**的主频和系统总线的速度是一样的，但**CPU**的速度越来越快，倍频技术也就应允而生。它可使系统总线工作在相对较低的频率上，而**CPU**速度可以通过倍频来无限提升。故也叫倍频系数，没有单位。
  - **CPU**的主频=外频×倍频
  - 目前**CPU**主频达到了**GHz**，例如**Intel 酷睿i7** 主频为 **4GHz**

## 第二节 中央处理器（CPU）

### 5. CPU的主要性能指标

- **字长**：对**CPU**在单位时间内(同一时间)能一次处理的二进制数的位数叫字长。所以能处理字长为**8**位数据的**CPU**通常就叫**8**位的**CPU**。同理**32**位的**CPU**就能在单位时间内处理字长为**32**位的二进制数据。字长的长度是不固定的，对于不同的**CPU**、字长的长度也不一样。
  - 目前**CPU**的字长多为**32**位、**64**位，例如**Intel 酷睿i7** 的字长位**64**位。
- **核心数**：核心数就是**CPU**的核心数量，单核就是**1**，双核就是**2**，四核就是**4**。但是核心数绝对不是**CPU**数量。双核的**CPU**核心数量是**2**，但是**CPU**数量是**1**。双处理器的电脑的话**CPU**数量是**2**，但是核心数量不见得是**2**，而是每个**CPU**核心数量之和。
  - 目前**CPU**多为多核心，例如**Intel 酷睿i7** 是**4**核。

## 第二节 中央处理器（CPU）

### 5. CPU的主要性能指标

- **制作工艺：**在生产**CPU**过程中，要加工各种电路和电子元件，制造导线连接各个元器件。通常其生产的精度以微米、纳米来表示。精度越高，生产工艺越先进。在同样的材料中可以制造更多的电子元件，连接线也越细，**CPU**的集成度越高，功耗也越小。
  - 目前**CPU**的制作工艺已达到纳米级，例如**Intel 酷睿i7** 的制作工艺是**14**纳米。
- **针脚数：**就是接口类型。接口方式有引脚式、卡式、触点式、针脚式等。目前**CPU**的接口多是针脚式接口，对应到主板上就有相应的插槽类型。**CPU**接口类型不同，在插孔数、体积、形状都有变化，所以不能互相接插。**CPU**接口类型的命名，习惯用针脚数来表示
  - 例如**Intel 酷睿i7** 的针脚类型是**LGA 1151** 。

## 第二节 中央处理器（CPU）

### 5. CPU的主要性能指标

- **缓存（Cache）**：**CPU**进行处理的数据信息多是从内存中调取的，但**CPU**的运算速度要比内存快得多，为此在此传输过程中放置一存储器，存储**CPU**经常使用的数据和指令。这样可以提高数据传输速度。随着需求和技术的发展，缓存形成和建立了一个层次存储结构，一般分一级、二级、三级（高端**CPU**），以达到最高性价比。多级组织还可以提高**cache**的命中率，提高执行效能。每一级缓存中所储存的全部数据都是下一级缓存的一部分，这三种缓存的技术难度和制造成本是相对递减的，所以其容量也是相对递增的。当**CPU**要读取一个数据时，首先从一级缓存中查找，如果没有找到再从二级缓存中查找，如果还是没有就从三级缓存或内存中查找。一般来说，每级缓存的命中率大概都在**80%**左右，也就是说全部数据量的**80%**都可以在一级缓存中找到，只剩下**20%**的总数据量才需要从二级缓存、三级缓存或内存中读取，由此可见一级缓存是整个**CPU**缓存架构中最为重要的部分。
  - 目前**CPU**很多已采用三级缓存，例如**Intel 酷睿i7** 就是三级缓存。

## 第二节 中央处理器（CPU）

### 5. CPU的主要性能指标

- **热设计功耗：**（**Thermal Design Power, TDP**），指当处理器达到负荷最大的时候，释放出的热量，单位为瓦（**W**）。它是反应一颗处理器热量释放的指标。**TDP**功耗并不是**CPU**的真正功耗。功耗是**CPU**的重要物理参数，它等于流经处理器核心的电流值与该处理器上的核心电压值的乘积。而**TDP**是指**CPU**电流热效应以及其他形式产生的热能，它们均以热的形式释放。显然**CPU**的**TDP**小于**CPU**功耗。换句话说，**CPU**的功耗很大程度上是对主板提出的要求，要求主板能够提供相应的电压和电流；而**TDP**是对散热系统提出要求，要求散热系统能够把**CPU**发出的热量散掉，也就是说**TDP**功耗是要求**CPU**的散热系统必须能够驱散的最大总热量。现在**CPU**厂商越来越重视**CPU**的功耗，**TDP**功耗可以大致反映出**CPU**的发热情况，因此人们希望**TDP**功耗越小越好，越小说明**CPU**发热量小，散热也越容易，对于笔记本来说，电池的使用时间也越长。

- **Intel 酷睿i7的热设计功耗(TDP)是91W**



## 第二节 中央处理器（CPU）

### 6. CPU品牌及生产厂家

- **Intel公司**，创建于1968年，是生产CPU的老大哥，它占有**80%**多的市场份额，**Intel**生产的**CPU**就成了事实上的**x86CPU**技术规范 and 标准。最近的主要有**celeron** 速龙系列，**pentium**奔腾 系列，**core 2** 酷睿系列，和**i7 i5 i3** 系列**CPU**。



- **AMD公司**，创办于**1969**年，是唯一能与**Intel**竞争的**CPU**生产厂家，**AMD**公司的产品现在已经形成了以**Athlon XP**、**Duron**、**Sempron**、**Athlon 64**等为核心的一系列产品。



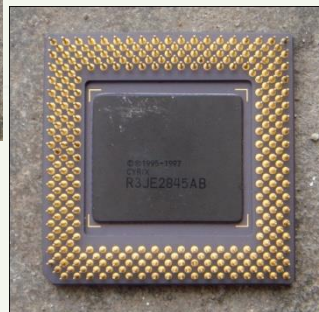


## 第二节 中央处理器（CPU）

### 6. CPU品牌及生产厂家

• **Cyrix公司**，1988年成立。它也曾与**AMD**和**Intel**在入门级处理器领域展开过激烈的争夺，一时间风头无限。主要生产廉价处理器。**1997年**，**Cyrix**先是被想要在处理器领域大干一场的国家半导体斥巨资收购，但是两年之后，灰头丧脸的国家半导体又将**Cyrix**连同**x86**架构授权许可在内的所有资产转手给了威盛。**Cyrix**算得上处理器领域最有名的失败者之一。

• **VIA公司（威胜）**，台湾地区的集成电路设计公司，主要生产主机板的晶片组、中央处理器（**CPU**）、以及记忆体。它是世界上最大的独立主机板晶片组设计公司。



## 第二节 中央处理器（CPU）

### 6. CPU品牌及生产厂家

• **国产龙芯**，2001年开始由中科院计算所研发，2008年开始产业化探索。

龙芯1A，32位，266MHz，2002年

龙芯1B，32位，200MHz

龙芯2A，64位，300MHz

龙芯2E，64位，800MHz，2006年

龙芯2F，64位，800MHz

龙芯2G，64位，1GHz

龙芯2H，64位，1GHz，2012年

龙芯3A，64位，1GHz，4核

龙芯3B，64位，1GHz，8核

龙芯3B 1500，64位，1GHz，8核，2015年



初步掌握了当代**CPU**设计的关键技术，为改变中国信息产业“无芯”的局面迈出了重要的步伐，对中国形成有自主知识产权的计算机产业有重要的推动作用，对中国的**CPU**核心技术、国家安全、经济发展都有举足轻重的作用。我们有信心：“龙芯”对**Intel**说“不”！

## 第三节 存储设备

### ◆ 存储器（Memory）

-**存储器**，是计算机系统中的记忆设备，用来存放程序和数据，是冯·诺依曼结构计算机的重要组成部分。计算机中的全部信息，包括输入的数据、运行的程序、运算的中间结果和最终结果都保存在存储器中。

-**存储器的特性：**

- （1）存储器是计算机中信息存储的核心。程序存储功能由存储器来承担；
- （2）内存是**CPU**与外界进行数据交换的窗口，**CPU**所执行的程序和所涉及的数据都由内存直接提供。**CPU**可以对内存进行直接读操作和写操作；
- （3）外存可以保存大量的程序和数据。

# 第三节 存储设备

## ◆ 存储单位

- 1 bit
- 1B(Byte) = 8b(bit)
- $1\text{KB} = 2^{10} \text{ Byte} = 1024 \text{ Byte} \approx 1000 \text{ B}$   $2^{10} \text{ Byte}$
- $1\text{MB} = 2^{10} \text{ KByte} = 1024 \text{ KB} \approx 1000 \text{ KB}$   $2^{20} \text{ Byte}$
- $1\text{GB} = 2^{10} \text{ MByte} = 1024 \text{ MB} \approx 1000 \text{ MB}$   $2^{30} \text{ Byte}$
- $1\text{TB} = 2^{10} \text{ GByte} = 1024 \text{ GB} \approx 1000 \text{ GB}$   $2^{40} \text{ Byte}$
- $1\text{PB} = 2^{10} \text{ TByte} = 1024 \text{ TB} \approx 1000 \text{ TB}$   $2^{50} \text{ Byte}$
- $1\text{EB} = 2^{10} \text{ PByte} = 1024 \text{ PB} \approx 1000 \text{ PB}$   $2^{60} \text{ Byte}$
- $1\text{ZB} = 2^{10} \text{ EByte} = 1024 \text{ EB} \approx 1000 \text{ EB}$   $2^{70} \text{ Byte}$
- $1\text{YB} = 2^{10} \text{ ZByte} = 1024 \text{ ZB} \approx 1000 \text{ ZB}$   $2^{80} \text{ Byte}$
- $1\text{DB} = 2^{10} \text{ YByte} = 1024 \text{ YB} \approx 1000 \text{ YB}$   $2^{90} \text{ Byte}$
- $1\text{NB} = 2^{10} \text{ DByte} = 1024 \text{ DB} \approx 1000 \text{ DB}$   $2^{100} \text{ Byte}$

# 第三节 存储设备

## 1. 多级存储系统

- 选用生产与运行成本不同的、存储容量不同的、读写速度不同的多种存储介质，组成一个统一的存储器系统，称为**多级存储系统**。
- **多级存储体系**是当前解决存储器性能与需求矛盾的最理想的技术途径
- **多级存储系统的特性：**
  - （1）在存储器体系结构中，各层之间的信息调度由辅助硬件或软件直接完成；
  - （2）存储体系结构能发挥整个存储系统的最大效能，有最佳的性能价格比；

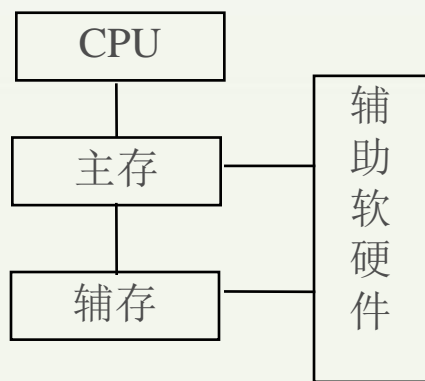


# 第三节 存储设备

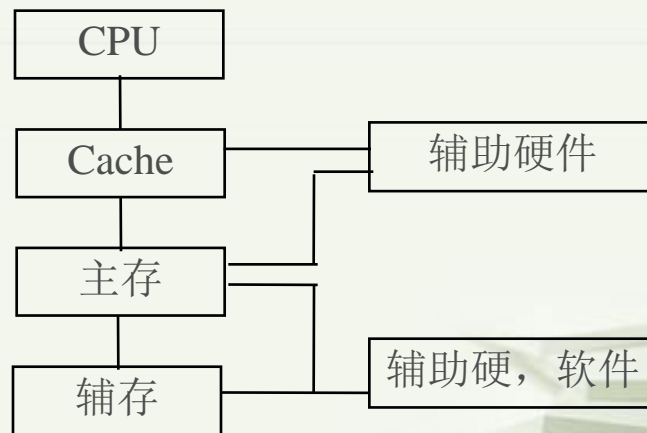
## 1. 多级存储系统

- 工作原理：

**CPU**首先访问高速缓存（**Cache**），如果命中没有，则存储系统通过辅助硬件，到主存储器中去找；如果主存没有**CPU**要访问的内容，则存储系统通过辅助硬件或软件，到辅存中去找。然后把找到的数据逐级上调。



(a) 两级存储器层次结构



(b) 三级存储器层次结构



# 第三节 存储设备

## 1. 多级存储系统

- 多级存储系统的优势：

（1）能够合理解决速度与成本的矛盾，以得到较高的性能价格比；

（2）使用磁盘、磁带等作为外存，不仅价格便宜，可以把存储容量做得很大，而且在断电时它所存放的信息也不丢失，可以长久保存，且复制、携带都很方便。

- 多级存储系统层次结构有如下规律：

- 价格依次降低；

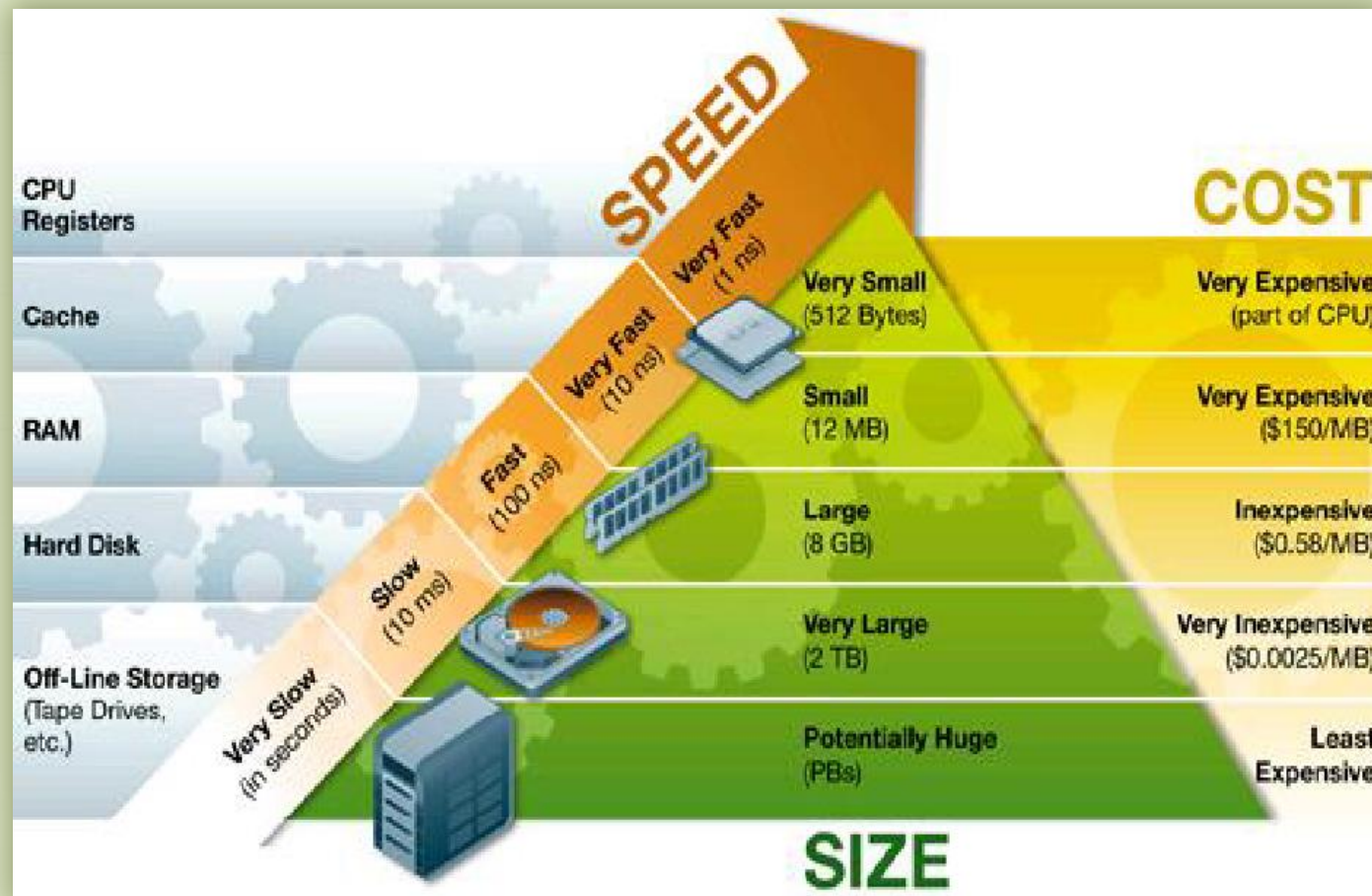
- 容量依次增加

- 访问时间依次增长

- CPU**访问频度依次减小

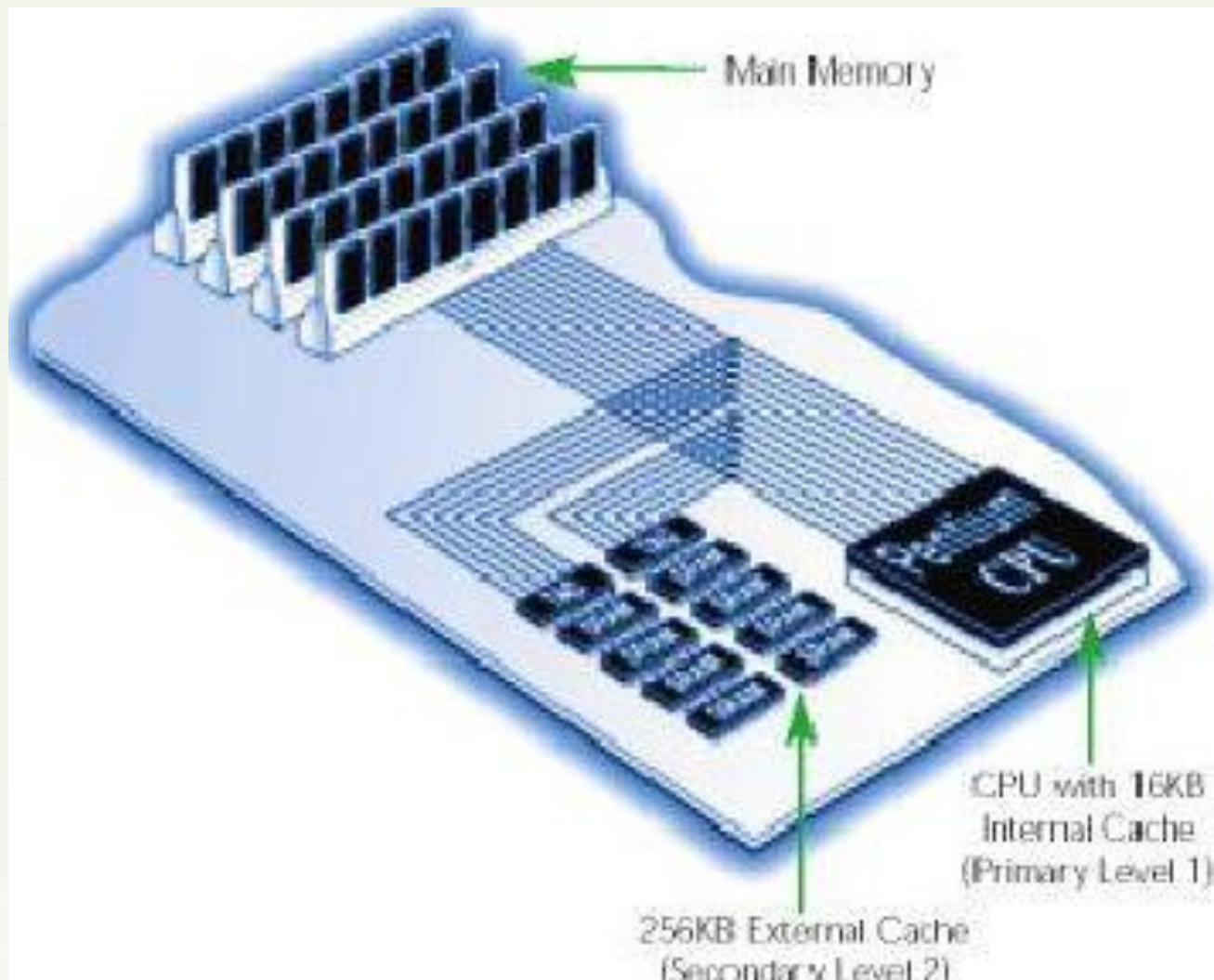
# 第三节 存储设备

## 1. 多级存储系统



# 第三节 存储设备

## 1. 多级存储系统



# 第三节 存储设备

## 2. 主存

**主存储器**指的是计算机中存储正处在运行中的程序和数据(或一部分)的部件, 通过地址、数据、控制三类总线与**CPU**等其他部件连通。

- **主存储器的组成:**

- (1) **贮存信息的存储体**

- 程序计数器(**PC**)、指令寄存器(**IR**)、指令译码器(**ID**)、程序状态寄存器(**PSW**)和地址形成部件等。

- (2) **信息的寻址机构**

- 即读出和写入信息的地址选择机构。包括: 地址寄存器(**MAR**)和地址译码器。

- (3) 写入信息所需的能源, 即写入线路、写驱动器等

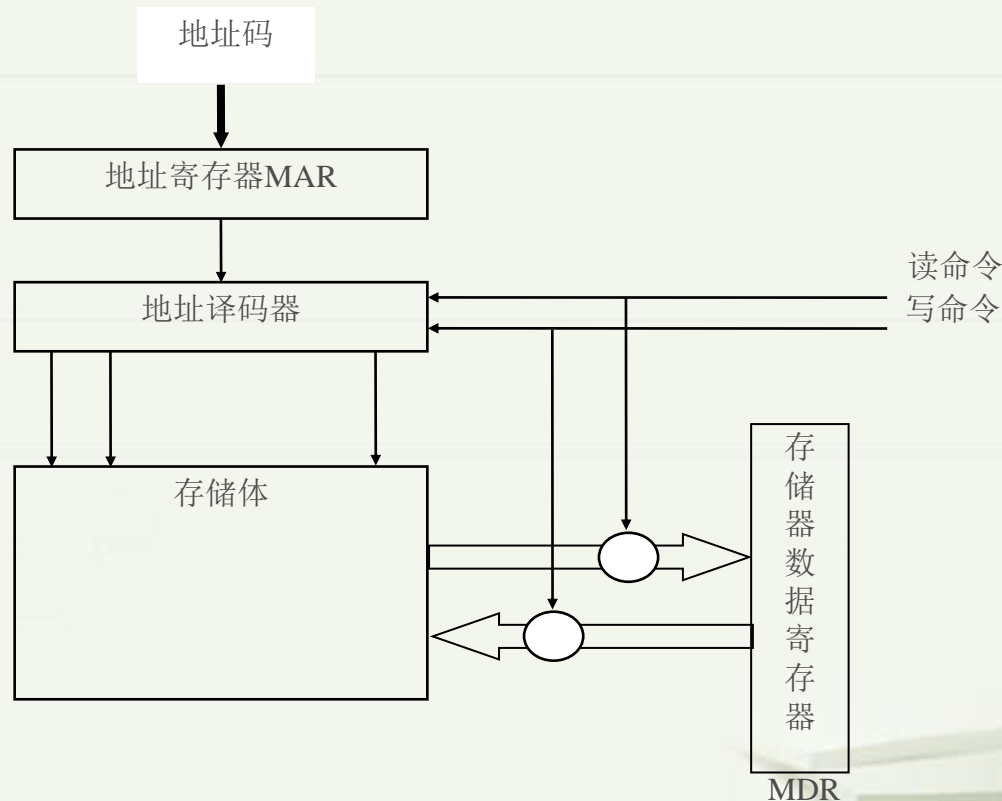
- (4) 读出所需的能源和读出放大器, 即读出线路、读驱动器和读出放大器。

# 第三节 存储设备

## 2. 主存

### (5) 存储器控制部件

无论是读或写操作，都需要由一系列明确规定的连续操作步骤来完成，这就需要主存时序线路、时钟脉冲线路、读逻辑控制线路，写或重写逻辑控制线路以及动态存储器的定时刷新线路等，这些线路总称为存储器控制部件。





# 第三节 存储设备

## 2. 主存

- 主存储器的两种主要形式：

- 随机存取存储器（**Random-Access Memory**，简称**RAM**）
- 只读存储器（**Read-Only Memory**，简称**ROM**）

### （1）随机存取存储器——**RAM**

利用双稳态触发器的开关特性进行记忆，只要保持供电，它总能保持两个稳定的状态中的一个状态。

**RAM**能随机地对存储器中的任何单元进行存取，且与存取的时间和该单元的物理位置无关。



# 第三节 存储设备

## 2. 主存

半导体**RAM**元件可以分为静态（**SRAM**）和动态（**DRAM**）两种。

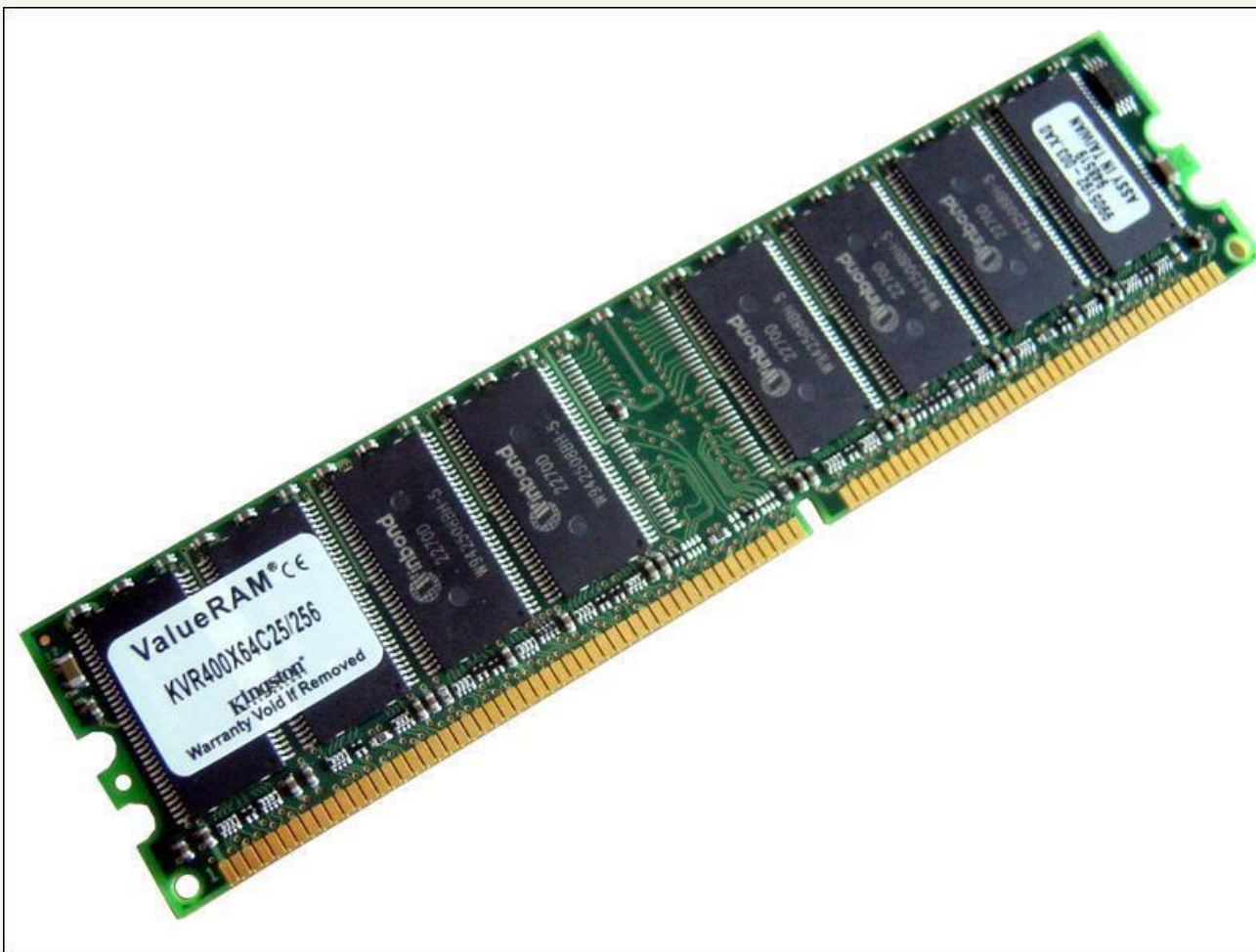
**SRAM**是利用双稳态触发器的开关特性进行记忆，只要保持供电，它总能保持两个稳定的状态中的一个状态。

**DRAM**靠**MOS**电路中的栅极电容来记忆信息，由于电容的电荷会泄漏除要保持电源外，还必须设置刷新电路，动态地每隔一定的时间间隔对它进行一次刷新，否则信息就会丢失。

**SRAM**读写速度高而成本也高，**DRAM**比**SRAM**集成度高、功耗低，从而成本也低，适于作大容量存储器。所以主内存通常采用**DRAM**，而高速缓冲存储器（**Cache**）则使用**SRAM**。

# 第三节 存储设备

## 2. 主存



# 第三节 存储设备

## 2. 主存

### (2) 只读存储器 (**Read-Only Memory**, 简称**ROM**)

是一种在机器运行过程中只能读出、不能写入信息的存储器，采用非易失性器件制造，在没有电源供电的情况下，其存储的信息也能长期保存。它所存储的信息是用特殊方式写入的。主要用于存储器经常要用的一些固定信息。

主要用于存储器经常要用的一些固定信息。**ROM**在计算机中主要用来保存出厂的一些固定设置和系统硬件引导程序，即通常所谓**BIOS**，**ROM**技术也用于一些固件产品中。

# 第三节 存储设备

## 2. 主存

### (2) 只读存储器 (**Read-Only Memory**, 简称**ROM**)

根据物理特性可将**ROM**分为如下几类:

-**MROM**——采用二次光刻掩膜工艺一次制成, 制成后内容不可改变。

-**PROM**——可以根据需要写入数据的器件, 一旦写入数据就固定下来, 不能再改变。

-**EPROM**——一种可改写的**ROM**, 可以对其内容进行多次改写, 所以叫可擦除可编程**ROM**

- 存储器的组织和管理: 在组成原理、操作系统等其他课程中学习。

# 第三节 存储设备

## 3. 高速缓冲存储器（Cache）

### ◆ Cache概念：

- ① **CPU**与主存储器之间的一种高速缓冲装置
- ② **Cache-主存**层次结构：由硬件变换地址和控制调度。

### ◆ Cache的特点：

- ① 位于**CPU**与主存之间，是存储器层次结构中级别最高的一级；
- ② 容量比主存小，目前一般有数**KB**到数**MB**；
- ③ 速度比主存快**5—10**倍，通常由存储速度高的双极型三极管或**SRAM**组成；
- ④ 其容量是主存的部分副本；
- ⑤ 其用途可用来存放指令，也可用来存放数据；
- ⑥ 快存的功能全部由硬件实现，并对程序员透明。

### ◆ Cache的基本组成

- **Cache**存储器
- **Cache**控制器

# 第三节 存储设备

## 3. 高速缓冲存储器（Cache）

### ◆ Cache的工作原理：

1、Cache以块为单位进行操作

2、当CPU发出访内操作请求后，首先由Cache控制器判断当前请求的字是否在Cache中，若在，叫命中，否则，不命中

3、若命中：

- 若是“读”请求，则直接对Cache读，与主存无关
- 若是“写”请求：
  - Cache单元与主存单元同时写（Write through写）
  - 只更新Cache单元并加标记，移出时修改主存（写回Copy back）
  - 只写入主存，并在Cache中加标记，下次从MM读出，保证正确。

4、未命中时：

- 若是“读”请求，则从主存读出所需字送CPU，且把含该字的一块送Cache，称“装入通过”，若Cache已满，置换算法；
- 若是“写”请求，直接写入主存。



# 第三节 存储设备

## 4. 辅助存储器

- ◆ **辅助存储器**是主存储器的后援存储设备，用以存放当前暂时不用的程序或数据。
- ◆ 对辅助存储器的基本要求是：
  - 存储密度高，记录容量大，每位价格低；
  - 记录介质可以重复使用；
  - 记录信息可长时间保存而不致丢失；
  - 非破坏性读出，读出时不需再生信息；
  - 存取速度较低，机械结构复杂，对工作环境要求较严。
- ◆ **辅助存储器的主要形式**：
  - ① 磁盘存储器；
  - ② 光盘存储器；
  - ③ 磁带存储器；
  - ④ 光磁盘存储器；
  - ⑤ 闪存（**Flash**）；

# 第三节 存储设备

## 4. 辅助存储器



光盘



软盘



闪存



硬盘



固态硬盘

一些典型的辅助存储器

# 第三节 存储设备

## 4. 辅助存储器

### ■ 磁盘存储器

#### 磁盘存储器原理：

采用磁记录方式进行信息记录，磁记录方式是一种编码方法，将数字信息转换成磁化单元进行存储。

#### ◆ 磁盘存储器的基本特点是：

- 存储密度高，记录容量大，每位价格低；
- 记录介质可以重复使用；
- 记录信息可长时间保存而不致丢失；
- 非破坏性读出，读出时不需再生信息；
- 存取速度较低，机械结构复杂，对工作环境要求较严。

#### ◆ 磁盘存储器的主要形式：

- 软磁盘；
- 硬磁盘；

# 第三节 存储设备

## 4. 辅助存储器



软磁盘



硬磁盘



磁盘阵列

一些典型的磁盘存储器

# 第三节 存储设备

## 4. 辅助存储器

### ■ 光盘存储器

光盘（**Optical Disk**）是利用光学原理读写信息的存储器。

光盘存储器的基本特点是：

- 存储密度高，记录容量大；
- 存取速度快；
- 介质寿命长；
- 记录的信息不易受干扰；
- 能够进行非接触读写和高速随机存取；

分不可擦写光盘，如**CD-ROM**、**DVD-ROM**等；和可擦写光盘，如**CD-RW**、**DVD-RAM**等。

# 第三节 存储设备

## 4. 辅助存储器



光盘存储器



# 第三节 存储设备

## 4. 辅助存储器

### ■ 闪存

闪存（Flash Memory）是一种长寿命的非易失性（在断电情况下仍能保持所存储的数据信息）的电子器件存储器。其数据删除不是以单个的字节为单位而是以一定大小的区块为单位，区块大小一般为256KB到20MB不等。闪存是电子可擦除只读存储器（EEPROM）的发展，与EEPROM不同的是，闪存在进行删除或重写操作时不需要整个芯片擦写，这样闪存比EEPROM更新速度快。

闪存的基本特点是：

- 存储密度高，记录容量一般；
- 存取速度快于光盘和磁盘；
- 介质寿命长；
- 抗震动、体积小、功耗低；
- 单位存储单元成本高，价格昂贵。

# 第三节 存储设备

## 4. 辅助存储器



**SD卡**



**SM卡**



**CF卡**



**记忆棒**



**XD卡**



**U盘**

一些典型的闪存产品

# 第三节 存储设备

## 4. 辅助存储器

### ■ 辅助存储外设接口

辅助存储器在功能上属于存储器，在管理上属于外部设备，其可以配置在主机内部，也可以外接，需要通过外设接口与主机相连，其接口主要有**USB**、**IDE**、**SATA**、**SCSI**等形式，有关外设接口的知识将在后面详细介绍。

# 第三节 存储设备

## 5. 存储器的主要性能指标

### ■ 内存

- 存储容量：在一个存储器中容纳的存储单元总数通常称为该存储器的存储容量。存储容量用字数或字节数(**B**)来表示。
- 存取时间：又称存储器访问时间，是指从启动一次存储器操作到完成该操作所经历的时间。具体讲，从一次读操作命令发出到该操作完成，将数据读入数据缓冲寄存器为止所经历的时间，即为存储器存取时间。
- 存储周期：是指连续启动两次独立的存储器操作(如连续两次读操作)所需间隔的最小时间。通常，存储周期略大于存取时间，其时间单位为**ns**

# 第三节 存储设备

## 5. 存储器的主要性能指标

### ■ 外存

#### ➤ 硬盘

- 容量
- 单碟容量
- 转速
- 最高内部传输速率
- 平均寻道时间

#### ➤ U盘

- 存储容量
- 数据传输率
- 写入数据传输率
- 支持接口类型
- 支持的操作系统
- 是否支持分区
- 加密功能
- 数据保存时间
- 工作环境温度
- 存放温度
- 运行相对湿度

# 第四节 外部设备

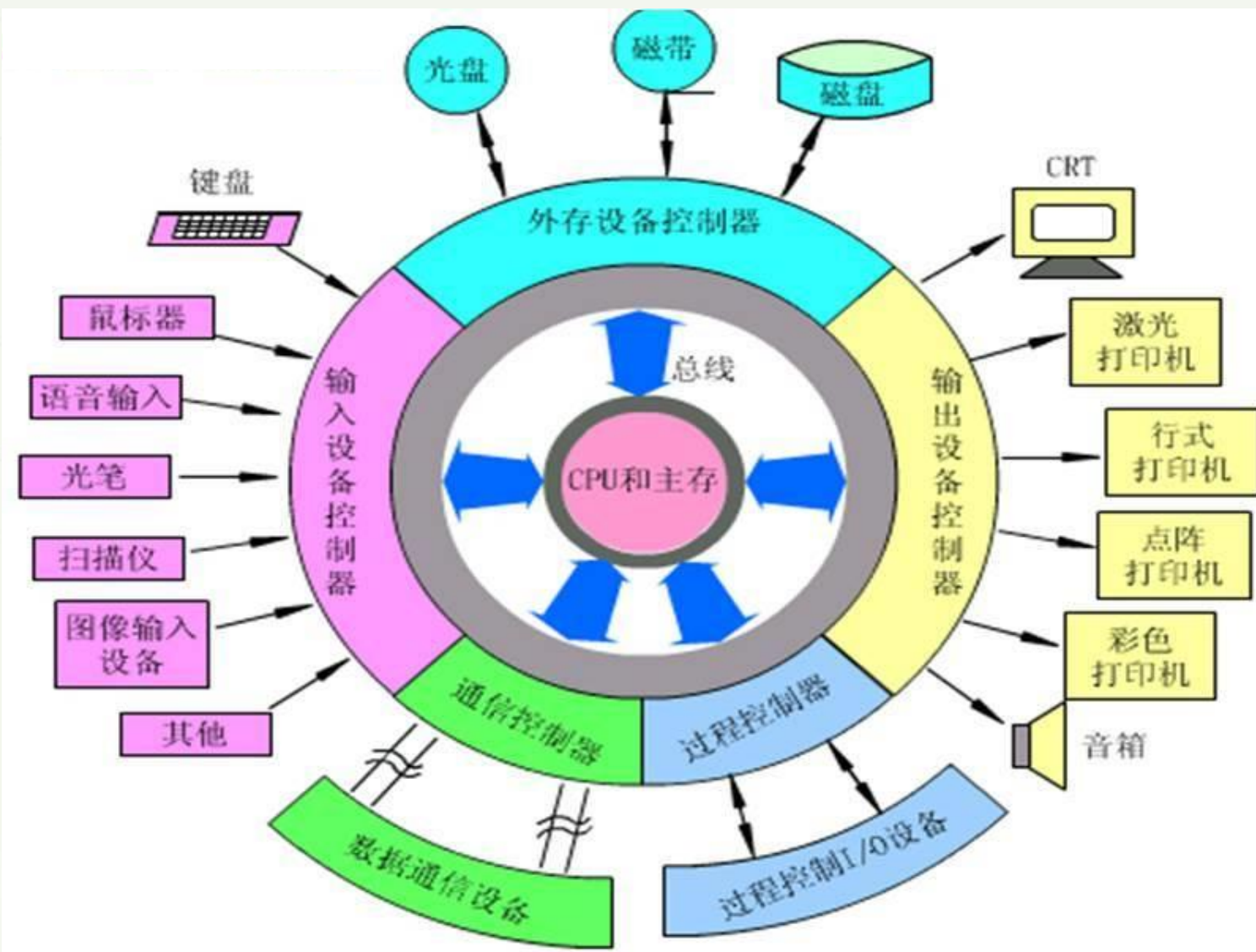
## ■ 外部设备

- **定义：** 指连在计算机主机以外的设备，包括输入和输出设备、外存储器、模数转换器、数模转换器、外围处理机等。是计算机与外界进行通信的工具，又称作外围设备。是计算机与人、计算机与其他机器间建立关系（交换信息）的设备。随着计算机技术的发展和计算机应用的深入，外围设备的种类也越来越多。
- **功能：** 在计算机和其他机器之间，以及计算机与用户之间提供联系。将外界的信息输入计算机；取出计算机要输出的信息；存储需要保存的信息和编辑整理外界信息以便输入计算机
- **分类：**
  - 机-机通信设备
  - 辅助存储（外存）设备
  - 人-机交互设备—输入/输出设备即，可分为输入设备、输出设备、外存设备、数据通信设备和过程控制设备几大类。



## 第四节 外部设备

### ■ 外部设备



## 第四节 外部设备

### ■ 输入设备

- **输入设备（InputDevice）**是人或外部与计算机进行交互的一种装置，用于把原始数据和处理这些数的程序输入到计算机中。计算机能够接收各种各样的数据，既可以是数值型的数据，也可以是各种非数值型的数据，如图形、图像、声音等都可以通过不同类型的输入设备输入到计算机中，进行存储、处理和输出。

- **常见的输入设备：**

- 字符输入设备：键盘；



# 第四节 外部设备

## ■ 输入设备

- 光学阅读设备：光学标记阅读机，光-学字符阅读机；



- 图形输入设备：鼠标器、操纵杆、光笔；



## 第四节 外部设备

### ■ 输入设备

- 图像输入设备：摄像机、数码相机、扫描仪、传真机；



- 模拟输入设备：语言模数转换识别系统。



## 第四节 外部设备

### ■ 输出设备

- **输出设备（Output Device）**是人与计算机交互的一种部件，用于数据的输出。它把各种计算结果数据或信息以数字、字符、图像、声音等形式表示出来。

- **常见的输出设备：**

- **显示器（Display）**是计算机必备的输出设备，常用的有阴极射线管显示器、液晶显示器和等离子显示器等。通过显示接口与主机相连。

**显示器的主要性能指标：**

- ① 显示屏尺寸
- ② 像素
- ③ 点距。
- ④ 分辨率





## 第四节 外部设备

### ■ 输出设备

- 打印机（**Printer**）是计算机的主要输出设备，它将计算机内部信息转换为人们需要的形式打印在纸上，便于存档保留。



打印机的主要性能指标：

- ① 质量
- ② 速度
- ③ 幅面
- ④ 耗材





# 第四节 外部设备

## ■ 输出设备



## 第五节 总线和接口

### ◆ 计算机系统总线

- 计算机系统中存储器、**CPU**等功能部件之间必须互联，才能组成计算机系统。
- 部件之间的互联方式
  - **分散连接**：各部件之间通过单独的连线互联
  - **总线连接**：将各个部件连接到一组公共信息传输线上。
  - **总线结构优点**：
    - 灵活：新加部件可以很容易地加到总线上，并且部件可以在使用相同总线的计算机系统之间互换；
    - 低成本；
- 现代计算机普遍使用的是总线互联结构。

## 第五节 总线和接口

◆ **总线 (Bus)** 是计算机各种功能部件之间传送信息的公共通信干线，它是由导线组成的传输线束。一个部件发出的信号可以被连接到总线上的其他所有部件所接收。总线通常由许多传输线或通路构成，在并行传输条件下，每条线可传输一位二进制信息，若干条线可同时传输多位二进制信息。

### ◆ 总线的分类：

计算机系统中含有多种总线，根据所连接部件的不同，总线通常被分成三种类型：

- **内部总线：**指芯片内部连接各元件的总线。
- **系统总线：**指连接CPU和其他高速部件的连接，如存储器、I/O控制器等。一般都在主板上。
- **I/O总线：**这类总线用于主机和I/O设备之间或计算机系统之间的通信。

## 第五节 总线和接口

### ➤ 系统总线:

- **数据总线:** 用于传输数据信息，双向，总线宽度一般与处理器字长相同。
- **地址总线:** 用于传输地址的，单向，地址总线的宽度决定了寻址空间的大小。
- **控制总线:** 用于传输控制信号和时序信号的，双向，总线宽度根据CPU的实际需要确定。

### ➤ 常见的总线:

- ISA总线
- PCI总线
- USB总线（VESA总线）
- PCI Express总线
- AGP总线
- SCSI总线

## 第五节 总线和接口

### ■ 接口

➤ 接口 (**Interface**)：连接两个设备之间的端口，由两侧特性所定义的共享边界。可以有物理级、软件级和逻辑级。与接口相对应的控制软件称为驱动程序。

### ➤ 常用的接口

- 硬盘接口：**IDE/ATA,SATA,SCSI**,光纤等
- 数据传输接口：串口 (**COM-232**)，并口 (**LPT**)
- **USB**：目前最通用的
- **IEEE 1394**:苹果公司的串行标准

## 第六节 数据的表示与编码

计算机最基本的功能就是对数据进行**存储**和**处理**。但到目前为止，计算机仍不能自动识别和处理人类的语言、文字、图像等形式的信息。因此，我们必须把原始的信息进行某种转换，转换为计算机能**接受和认识的数据**，然后才能存储和处理。

现代计算机采用二进制，主要原因是二进制在技术实现上简单、运算规则也简单。因此计算机中的所有的信息都是以二进制数的形式表示和存储的。



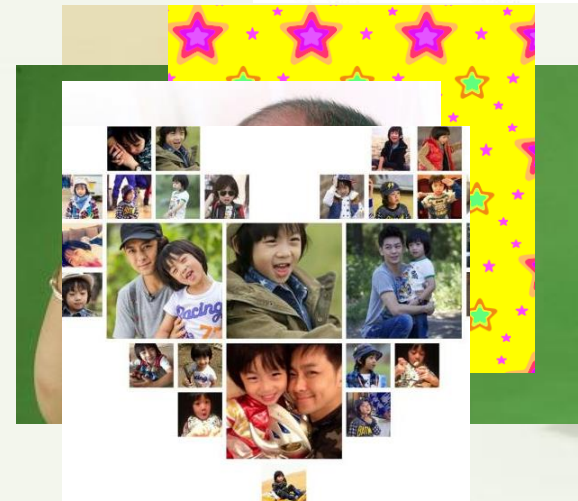
# 第六节 数据的表示与编码



以 何 於

Ce n'est pas un troisième choc pétrolier<sup>3</sup>, c'est le deuxième choc géostratégique qu'est en train de provoquer, sous nos yeux, la «drôle de guerre»<sup>3</sup> du Golfe. Le premier avait eu lieu il y a moins d'un an, au moment du effondrement simultané du mur de Berlin et des régimes communistes de l'Est. Il a entraîné la mise en place d'un nouvel ordre de sécurité en Europe qui n'a pas plus fondé, comme c'était le cas depuis quarante ans, sur la coupure en deux du continent, l'antagonisme géopolitique entre l'Est et l'Ouest et la militarisation successive qui en découlait dans l'un et l'autre camp.

La bénédiction donnée à la réunification allemande cette semaine, à Moscou, par les quatre alliés<sup>4</sup> de



## 第六节 数据的表示与编码

[illegible]

# 第六节 数据的表示与编码

## ■ 计算机中的数和数制

### ➤ 数字系统

或称记数法，数制，是使用一组数字或符号来表示数的体系。

- 一组固定的符号
- 统一的规则

### ➤ 两类表示系统

- 位置化的数字系统
- 非位置化的数字系统

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 非位置化的数字系统—罗马数制

■ 符号：共有7个，即：

**I(1)、V(5)、X(10)、L(50)、  
C(100)、D(500)、M(1000)**

■ 规则

- 重复数次表累加
- 右加左减
- 数码限制，最多出现三次
- 加线成千

# 第六节 数据的表示与编码

## ■ 计算机中的数和数制

### ➤ 非位置化的数字系统—罗马数制

I	1
II	2
III	3
IV	4
V	5
VI	6
VII	7
VIII	8
IX	9
X	10
XI	11
XII	12
XIII	13
XIV	14
XV	15
XVI	16
XVII	17

XVIII	18
XIX	19
XX	20
XXX	30
XL	40
L	50
LX	60
LXX	70
LXXX	80
XC	90
XCIX	99
C	100
CI	101
CII	102
CXCIX	199
CC	200
CCC	300

M	1000
MCD	1400
MCDXXXVII	1437
MD	1500
MDCCC	1800
MDCCCLXXX	1880
MCM	1900
MM	2000
MMM	3000
MMMCCCXXXIII	3333
M $\bar{V}$	4000
$\bar{V}$	5000
$\bar{X}$	10000
$\bar{L}$	50000
$\bar{C}$	100000
$\bar{D}$	500000
$\bar{M}$	1000000

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 位置化数字系统-进位计数制

- **基数**：可使用的数字符号的数目称为该进制的基数。若进制的基数为**n**，称之为**n**进制。“逢**n**进一”

- 有**n**个数字符号

- **规则**：每个数字符号表示的数值为该数字符号与其位置权的乘积。“位权”

- **表示形式**

$$\pm(S_{k-1} \cdots S_2 S_1 S_0 \cdot S_{-1} S_{-2} \cdots S_{-l})_b$$

- **数值**

$$\pm(S_{k-1} \cdots S_2 S_1 S_0 \cdot S_{-1} S_{-2} \cdots S_{-l})_b$$

$$n = \pm S_{k-1} \times b^{k-1} + \cdots + S_2 \times b^2 + S_1 \times b^1 + S_0 \times b^0 + S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \cdots + S_{-l} \times b^{-l}$$



## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 常用的各种进制数的表示

进位制	二进制	八进制	十进制	十六进制
规则	逢二进一	逢八进一	逢十进一	逢十六进一
基数	$R=2$	$R=8$	$R=10$	$R=16$
基本符号	0,1	0,1,2,...,7	0,1,2,...,9	0,1,...,9,A,...,F
权	$2^i$	$8^i$	$10^i$	$16^i$
形式表示	B或 $( )_2$	O或 $( )_8$	D或 $( )_{10}$	H或 $( )_{16}$

# 第六节 数据的表示与编码

## ■ 计算机中的数和数制

### ➤ 数制之间的相互转换

(1) 二、八、十六进制数转换为十进制数

(2) 十进制数转换为二、八、十六进制数

- 十进制数转换为二进制

- 十进制数转换为八进制

- 十进制数十六进制数

(3) 二进制数和八进制数、十六进制数的转换

- 二进制数转换为八、十六进制数

- 八、十六进制数转换为二进制数

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

##### (1) 二、八、十六进制数转换为十进制数

**转换规则：**

$$\begin{aligned} & \pm (S_{k-1} \cdots S_2 S_1 S_0 \cdot S_{-1} S_{-2} \cdots S_{-l})_b \\ &= \pm S_{k-1} \times b^{k-1} + \cdots + S_2 \times b^2 + S_1 \times b^1 + S_0 \times b^0 + S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \cdots + S_{-l} \times b^{-l} \end{aligned}$$

**例1-1** 将 **$(10010.11)_2$** 转换为十进制数

解：

$$\begin{aligned} & (10010.11)_2 \\ &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \\ &= (18.75)_{10} \end{aligned}$$

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

**例1-2** 将 $(547.6)_8$ 转换为十进制数  
解：

$$\begin{aligned} & (547.6)_8 \\ &= 5 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 6 \times 8^{-1} \\ &= (359.75)_{10} \end{aligned}$$

**例1-3** 将 $(C6E.A2)_{16}$ 转换为十进制数  
解：

$$\begin{aligned} & (C6E.A2)_{16} \\ &= 12 \times 16^2 + 6 \times 16^1 + 14 \times 16^0 + 10 \times 16^{-1} + 2 \times 16^{-2} \\ &= (3182.6328125)_{10} \end{aligned}$$

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### (2) 十进制数转换为二、八、十六进制数

任一十进制数 $X$ ，可以表示为 $X = X_{\text{整数}} + X_{\text{小数}}$ 。将这两部分分开转换。

##### ■ 整数部分的转换：采用“除2求余法”

- 具体方法为：将 $X_{\text{整数}}$ 连续用2除，求得余数（1或0）分别为 $K_0$ 、 $K_1$ 、 $K_2$ 、...，直到商为0，所有余数排列 $K_{n-1}K_{n-2}\dots K_2K_1K_0$ 即为所转换的二进制整数部分。

##### ■ 小数部分的转换：采用“乘2取整法”

- 具体方法为：连续用2乘 $X_{\text{小数}}$ ，依次求得各整数位（0或1） $K_{-1}$ 、 $K_{-2}$ 、...、 $K_{-m}$ ，直到乘积的小数部分 $F_i$ 为0。在小数转换过程中，出现 $F_i$ 恒不为0时，可按精度要求确定二进制小数的位数。

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

##### ● 十进制转换为二进制:

**例1-4** 求 $(65)_{10}$ 的二进制表示

解:

除以2

65/2

32/2

16/2

8/2

4/2

2/2

1/2

商 $Q_i$

32

16

8

4

2

1

0

余数 $K_i$

$K_0=1$

$K_1=0$

$K_2=0$

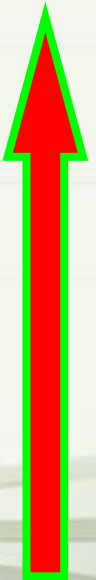
$K_3=0$

$K_4=0$

$K_5=0$

$K_5=1$

$(65)_{10}=(1000001)_2$





## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### ● 十进制转二进制

例**1-5** 求 $(0.6875)_{10}$ 的二进制值

解： 乘以2

$$0.6875 \times 2$$

$$0.3750 \times 2$$

$$0.7500 \times 2$$

$$0.5000 \times 2$$

小数 $F_i$  整数 $K_i$

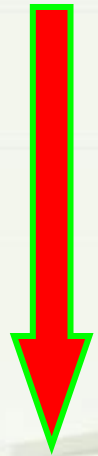
$$0.3750 \quad K_{-1} = 1$$

$$0.7500 \quad K_{-2} = 0$$

$$0.5000 \quad K_{-3} = 1$$

$$0.0000 \quad K_{-4} = 1$$

$$(0.6875)_{10} = (0.1011)_2$$



## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

- 将十进制数转换为八进制数、十六进制数时，使用的方法与十进制数转换成二进制数的方法基本相同，只是求整数部分时是用商除以**8**或**16**，取其余数；小数部分改用乘以**8**或**16**，取其整数即可。

#### 练习：

1. 将下列二，八，十六进制数转换成十进制数

**$(1101011.1011)_2$  ,  $(73.5)_8$  ,  $(FAE.BD)_{16}$**

2. 将下面十进制转换成二、八、十六进制

**$(1024.625)_{10}$  ,  $(555.32)_{10}$**

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### (3) 二进制数转换为八、十六进制数

假设二进制数表示为 ( $N_{\text{整数}} + M_{\text{小数}}$ )

例如,  $(11001.01)_2$

- **整数部分N:** 从右向左按三(四)位分组, 不足补零, 然后按三(四)位转换为相应八(十六)进制数。
- **小数部分M:** 从左向右按三(四)位分组, 不足补零, 然后按三(四)位转换为相应八(十六)进制数。

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### (3) 二进制数转换为八、十六进制数

##### ■ 二进制转八进制

例1-6 二进制数  $(10011001011.10110101)_2$

$$\begin{array}{ccccccc} \underline{(010)} & \underline{011} & \underline{001} & \underline{011} & \underline{101} & \underline{101} & \underline{010} \\ 2 & 3 & 1 & 3 & 5 & 5 & 2 \end{array} \quad {}_2 = (2313.552)_8$$

##### ■ 二进制转十六进制

例1-7 二进制数  $(1001011.110101)_2$

$$\begin{array}{cccc} \underline{(0100)} & \underline{1011} & \underline{1101} & \underline{0100} \\ 4 & B & D & 5 \end{array} \quad {}_2 = (4B.D5)_{16}$$

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### (4) 八、十六进制数转换为二进制数

##### ■ 转换规则：

- **整数部分：**从右向左按一位八(十六)进制数转为三(四)位二进制数。去掉左边第一个**1**之前的**0**；
- **小数部分：**从左向右按一位八(十六)进制数转为三(四)位二进制数。去掉右边最后一个**1**之后的**0**；

## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

#### • 八、十六进制转二进制

例1-8 八进制数(1365.24)<sub>8</sub> 转换为二进制数

$$\begin{aligned}(1365.24)_8 &= (\underline{001} \ \underline{011} \ \underline{110} \ \underline{101} . \underline{010} \ \underline{100})_2 \\ &= (\underline{1011110101.0101})_2\end{aligned}$$

例1-9 十六进制数(FB4.5C)<sub>16</sub> 转换为二进制数

$$\begin{aligned}(\text{FB4.5C})_{16} &= (\underline{1111} \ \underline{1011} \ \underline{0100} . \underline{0101} \ \underline{1100})_2 \\ &= (\underline{111110110100.010111})_2\end{aligned}$$



## 第六节 数据的表示与编码

### ■ 计算机中的数和数制

#### ➤ 数制之间的相互转换

##### ■ 练习

**1.** 将下面二进制数转换成八进制，十六进制。

$(10011011.10101)_2$

**2.** 将下面八、十六进制数转换成转二进制数。

$(175.6)_8$

$(9D.B2)_{16}$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

在日常生活中，信息的表示形式多种多样。我们常用十进制数表示数量，而用文字和图片甚至多媒体手段表达自己的思想以及所见所闻。

计算机可以处理数字、文本、声音、图片等多种信息数据。但在计算机内部，**信息的表示和存储均采用二进制形式**，即任何形式的信息在计算机中都必须转化为二进制数的形式进行表示和存储。

**人们会想：为什么如此简单的二进制系统能够表示出客观世界中那么多丰富多彩的信息呢？**

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

这是因为，尽管**0-1**简单，但它们可以有千变万化的组合，这就是编码。从而使得任何形式的信息都将经过编码转化为二进制数的形式在计算机中进行表示、处理和储存。

➤ **编码：**就是用按一定规则组合而成的若干位二进制码，用来表示信息（数字、符号等）。

➤ **编码的要素：**

- **长度**，一般编码越长，表示的信息量就越大，但是存储和传输时重要的资源就多。

- **代表的信息含义唯一**，一个信息被唯一的编码，反过来，一个编码有唯一的信息含义，否则就无法处理。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

信息要用多少个比特位来表示，这就是计算机中数据的单位（又称为计量单位）。常用的单位有位（比特）、字节和字。

无论什么信息，在计算机中表示（编码）时，都要规定一个位长（多少位，或用几个计量单位），位长决定了表示信息的范围（大小）。位长一般是字节的整数倍。

#### ➤ 编码的两种方式：

- 定长编码
- 不定长编码

下面简单介绍一下常见的信息在计算机中的表示（编码）方法。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

数是最常见的一类信息。在计算机内部，所有的数都是用二进制代码来表示的。前面我们已经学习了数如何转化成二进制数，但是要在计算机中表示为计算机中的数（机器数），还有两个问题还没解决：

➤ 数的正、负号在计算机中怎么表示？

➤ 小数点在计算机中怎么表示出来？

- **机器数**：一个数在计算机中的二进制表示形式，称为机器数。
- **真值**：带符号位的机器数对应的真正数值，称为真值。

**正负数的表示**：在计算机内，通常把**1**个二进制数的**最高**位定义为**符号位**，用“**0**”表示正数，“**1**”表示负数。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

一个数一般由整数部分、小数点、小数部分三部分组成，计算机如何表示小数点呢？

在计算机中小数点一般有两种表示方法：

- 定点数：规定小数点位置固定不变；
- 浮点数：小数点的位置不固定，可以浮动；

在计算机中，通常是用定点数来表示整数和纯小数，分别称为定点整数和定点小数。对于既有整数部分、又有小数部分的数，一般用浮点数表示。



## 第六节 数据的表示与编码

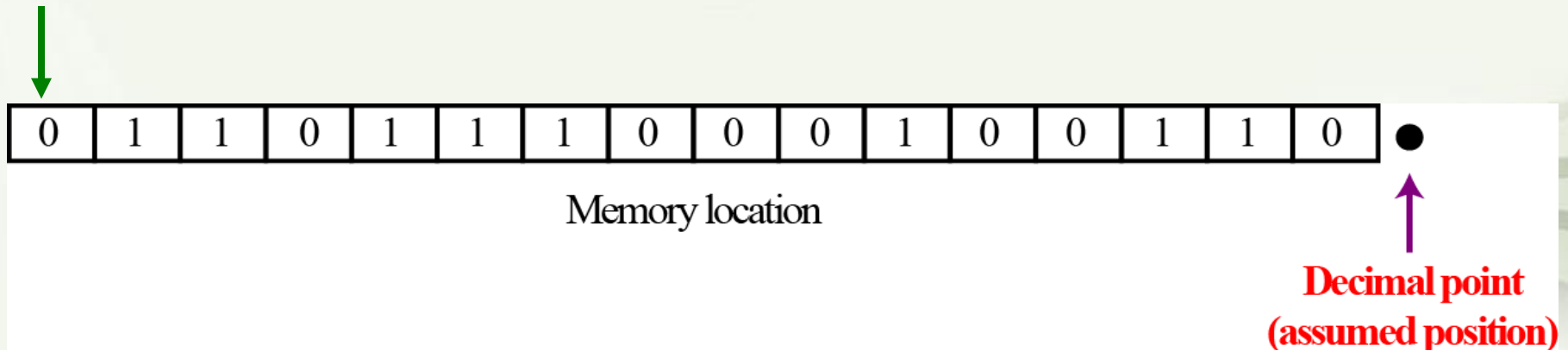
### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

- **定点整数**：在定点数中，当小数点的位置固定在数值位最低位的右边时，就表示一个整数。请注意：小数点并不单独占**1**个二进制位，而是默认在最低位的右边。定点整数又分为**有符号数**和**无符号数**两类。

是否是表示符号



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

例如，假设一个整数用**8**位二进制表示，则  
**N=11011010** 是十进制多少呢？

无符号整数：

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

$$1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (218)_{10}$$

有符号整数：

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

$$1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (-90)_{10}$$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

表示一个整数的位数，决定了表示整数的大小（范围）。  
假设用  $n$  位二进制表示一个整数，则：

- 如果表示无符号整数，其表示的整数范围是  $0 \sim 2^n - 1$
- 如果表示有符号整数，其表示整数的范围是  ~~$-2^{n-1} - 1$~~

~~$\sim 2^{n-1} - 1$~~   $-2^{n-1} \sim 2^{n-1} - 1$

例如，用 **8** 位二进制表示整数，则：如果是无符号整数，其表示的范围是  $0 \sim 255$ ；如果是有符号整数，则范围是

$-128 \sim +127$ ，而不是  $-127 \sim +127$

Why?

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

计算机中表示整数时，并不是用这种直接的编码，而是用的补码，因为直接用前面的这种编码进行运算会有问题。

例如：

$$1-1=?$$

$$(1)_{10} + (-1)_{10} = (0)_{10}$$

$$(00000001)_2 + (10000001)_2 \\ = (10000010)_2 = (-2)_{10}$$

定点整数的三种编码形式：原码、反码、补码

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 定点表示法

#### (1) 整数的原码

- 编码原则：最高位存放符号(0为正,1为负)，其他位与数的真值相同。

例：用8位表示整数，写出下列数的原码

$$X=1101001, Y=-1101001$$

解：

$$[X]_{\text{原}} = 01101001$$
$$[Y]_{\text{原}} = 11101001$$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 定点表示法

### (2) 整数的反码

- 编码原则：正数的反码与原码相同。负数的反码相对其原码，符号位不变，其余各位取反。

例：用**8**位表示整数，写出下列数的反码

**$X=1101001$ ,  $Y=-1101001$**

解：  $[X]_{\text{原}} = \mathbf{0}1101001$      $[X]_{\text{反}} = \mathbf{0}1101001$   
 $[Y]_{\text{原}} = \mathbf{1}1101001$      $[Y]_{\text{反}} = \mathbf{1}0010110$



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 定点表示法

#### (3) 整数的补码

- 编码原则：正数的补码与原码相同；负数的补码相对其原码，符号位不变，其余各位取反，末位加**1**。

例：用**8**位表示整数，写出下列数的补码

**X=1101001, Y=-1101001**

解：  $[X]_{\text{原}} = 01101001$      $[X]_{\text{反}} = 01101001$

$[X]_{\text{补}} = 01101001$

$[Y]_{\text{原}} = 11101001$      $[Y]_{\text{反}} = 10010110$

$[Y]_{\text{补}} = 10010111$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

#### ➤ 补码编码的特点：

- 符号位与有效值部分一起参加运算，简化运算规则；
- 使减法运算转换为加法运算，简化计算机硬件线路。
- 补码运算规则：

$$[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$$

$$[X-Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

➤ 在用程序设计语言写程序中使用的是原码；

➤ 在计算机中，数据是以补码的形式存在的；

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 定点表示法

例：求整数**86**，**-115**的原码、反码、补码。

（用**8**位二进制表示整数）

解： 第一步，转换为二进制	<b>1010110</b>	<b>-1110011</b>
第二步，原码	<b>01010110</b>	<b>11110011</b>
第三步，反码	<b>01010110</b>	<b>10001100</b>
第四步，补码	<b>01010110</b>	<b>10001101</b>

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

例：一整数用**8**位二进制表示，补码编码为：**11100000**  
求其真值。

解：第一步，确定数的符号    负数    **-1100000**

第二步，减一    **-1011111**

第三步，取反    **-0100000**

第四步，得到真值     **$(-0100000)_2 = (-32)_{10}$**

● 对补码再取其补码，就是原码

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

#### ➤ 溢出(overflow)

编码表示（存储）整数的范围是受编码长度限制的。给定编码位数，表示（存储）整数的范围就确定了。当一个整数超出表示（存储）的数的范围时，就会产生错误的结果，称之为**溢出**。

在现代计算机中，都有对溢出的保护设置，当发生溢出时，机器将自动停止当前运算，并报警。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 定点表示法

#### 练习：

- (1) 假设编码长度为**16**，写出**28**，**-28**的原码、反码、补码。
- (2) 某数的**8**位二进制补码为**00001101**，写出其真值。
- (3) 某数的**8**位二进制补码为**11100110**，写出其真值。



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

- **定点小数**：当小数点的位置固定在**符号位与最高数值位**之间时，就表示一个纯小数。

符号位



Memory location

约定的小数点位置

## 第六节 数据的表示与编码

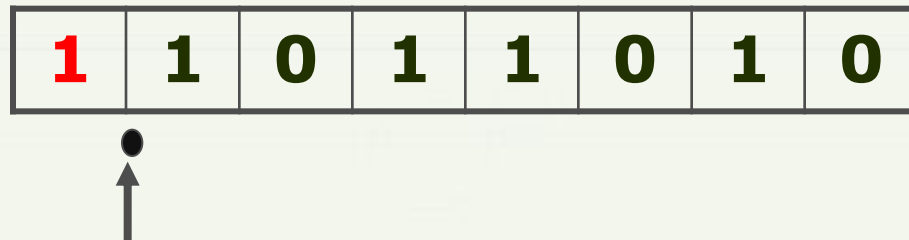
### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 定点表示法

例如，假设一个小数用**8**位二进制表示，则

$N=11011010$  是十进制多少呢？



$$1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} + 0 \times 2^{-7} \\ = (-0.703125)_{10}$$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 浮点表示法

在浮点数表示法中，小数点的位置是可以浮动的。浮点表示法类似科学计数法。例如23.45可以写成 $10 \times 2.345$ 、 $10^2 \times 0.2345$ 、 $10^3 \times 0.02345 \dots$ 等各种不同形式。因此，一个浮点数可以表示为：

$$N = M \times R^E$$

其中：

-M为该数的尾数。尾数为纯小数，它决定了所表示数的精度。二进制定点纯小数。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

##### ■ 浮点表示法

-E为该数的阶码。阶码总为整数，表示数的大小范围，用于确定小数点的浮动位数，若阶码为正数，则小数点向左移动；若阶符为负数，则小数点向右移动。**二进制定点整数**

-R为阶码的基数，二进制，基数 $R=2$

浮点数表示包括阶符、阶码、数符、尾数四个部分：

阶符	阶 码	数符	尾 数
----	-----	----	-----

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 浮点表示法

例如，假设用**2**个字节（**16**位）表示一个浮点数，规定前**6**位表示阶码（含阶符），后**10**位表示尾数（含数符）。

表示出的浮点数如下：

0	1	2	3	4	5	6	7	6	9	0	1	2	3	4	5
0	0	0	1	0	1	1	1	1	0	1	0	1	0	0	0

其表示的十进制数值是多少？

$$-0.110101\text{B} \times 2^{101\text{B}} = -11010.1\text{B} = -26.5\text{D}$$

为了使所表示的浮点数既精度高、又范围大，就必须合理规定浮点数的格式。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ■ 浮点表示法

- 实数的一种浮点表示法——IEEE 标准 754

- 实数的规范化表示

- 约定：在小数点左边使用一位非零数码，称为规范化

- 实数的三部分信息：

- 符号，指数，尾数（小数点右边的位）

注：不存储小数点和固定部分（小数点左边的1）



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

例如，有二进制实数1000111.0101B，规范化后：

$$2^{110B} \times 1.0001110101B$$

数学中表示+	$2^{110}$	×	1.0001110101
计算机中表示+	110		0001110101
	↑		↑
	符号		尾数
	↑		
	指数		

**符号**—0（正），1（负）

**指数**—小数点移动的位数。可正可负的，采用**余码表示法**。

**尾数**—小数点右边的二进制数。它定义了该数的精度。

以**无符号整数形式**存储。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ◆ 余码系统

为了让正的和负的整数都可以作为无符号整数存储，计算机通常采用余码系统。在余码系统中，使用一个正整数（称为一个偏移量）加到每个数字中，用于把它们同一移到非负的一边。

假设余码系统是基于 $m$ 位的模式，则偏移量为 $k=2^{m-1}-1$

一个数 $n$ 的余码 $=n+k$

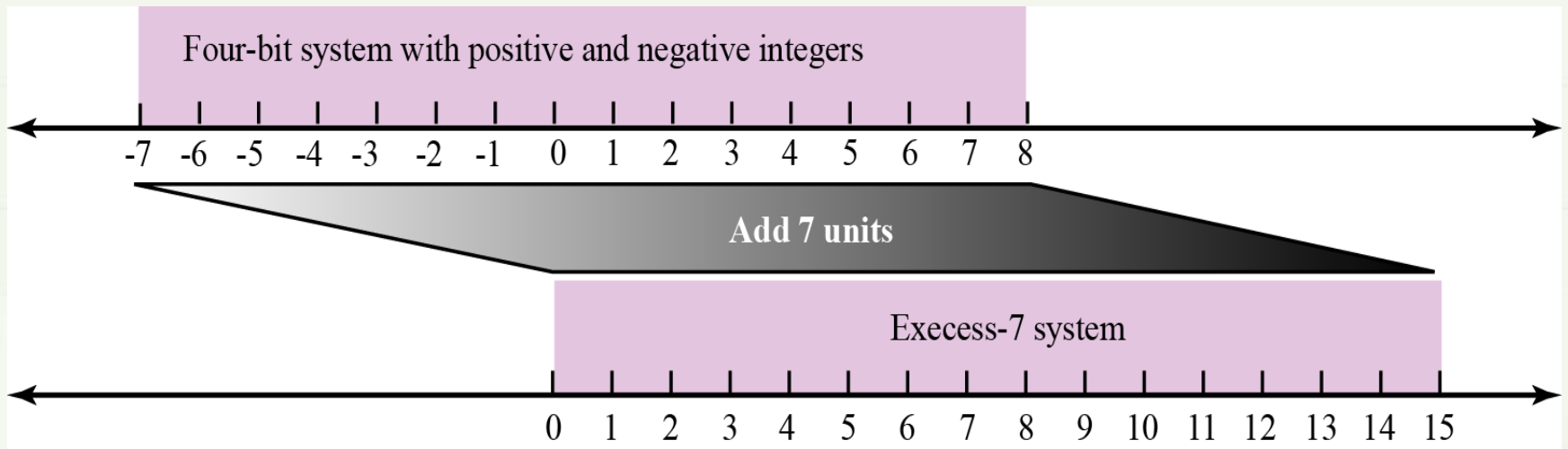
例如，存储整数的位数为 $m=4$ ，则偏移量为 $2^{4-1}-1=7$ ，于是，整数7的余码为 $7+7=14$ ，整数-7的余码为 $(-7)+7=0$ ，整数8的余码为 $8+7=15$

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

#### ◆ 余码系统



新系统称为**余7码**，或者**偏移量为7的偏移表示法**。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 计算机中数的表示

#### ◆ 余码系统

值 $n$	余码 $y=n+2^{m-1}-1$	余码二进制
4	7	111
3	6	110
2	5	101
1	4	100
0	3	011
-1	2	010
-2	1	001
-3	0	000

$m=3$ 的位模式余码系统

值 $n$	余码 $y=n+2^{m-1}$	余码二进制
8	15	1111
7	14	1110
6	13	1101
5	12	1100
4	11	1011
3	10	1010
2	9	1001
1	8	1000
0	7	0111
-1	6	0110
-2	5	0101
-3	4	0100
-4	3	0011
-5	2	0010
-6	1	0001
-7	0	0000

$m=4$ 的位模式余码系统

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中数的表示

例如，要求按IEEE 标准 754的方式表示实数  $(6.75)_{10}$  一个字节，其中符号1位，指数3位，尾数4位。请给出存储格式。

第一步，转化为二进制： $(6.75)_{10} = (110.11)_2$

第二步，规格化： $(110.11)_2 = 2^{10} \text{ B} \times 1.1011 \text{ B}$

第三步，指数取余码： $(10)_2 + 2^{3-1} - 1 = (2)_{10} + (3)_{10} = (5)_{10}$   
 $= (101)_2$

第四步，存储：符号位(0)，阶码(110)，尾数(1011)

0	1	0	1	1	0	1	1
---	---	---	---	---	---	---	---

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中西文字符的表示

- 通常把数据分为数值型和非数值型
- 字符是非数值型数据的基础
  - 目前，字符与字符串是计算机中用得最多的非数值型数据。
- 为了使计算机系统能够识别和处理字符，必须对字符按一定规则用二进制进行编码
  - 使得系统里的每一个字母有唯一的编码。



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中西文字符的表示

#### ◆ ASCII码

- 美国标准信息交换代码 (American Standard Code for Information Interchange)
- 每个字符由7位二进制编码，共可表示 $2^7=128$ 个字符。
- ASCII码表中：
  - 编码值0~31不对应任何可印刷（或称有字形）字符，通常称它们为控制字符，用于通信中的通信控制或对计算机设备的功能控制。
  - 编码值为32的是空格（或间隔）字符SP。
  - 编码值为127的是删除控制DEL码。
  - 其余的94个字符称为可印刷字符。

标准ASCII码表 见p35

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 计算机中西文字符的表示

高3位 低4位	000	001	010	011	100	101	110	111
0000	NUL	DEL	SP	0	@	P	、	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	”	2	B	R	b	r
0011	EXT	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	,	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	.	<	L	\	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	。	>	N	↑	n	~
1111	SI	US	/	?	O	↓	o	DEL

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中西文字符的表示

由于标准**ASCII** 字符集字符数目有限，在实际应用中往往无法满足要求。为此，人们将**7位ASCII** 码扩充为**8位**，即原来的最高位**0**变**1**。这样可表示的字符增加了**1**倍，即在原来**128**个标准字符的基础上，可以再增加**128**个字符，增加的字符集可根据需要确定。所以，**扩展的ASCII码字符集**包括**256**个字符，其中**128**个是标准的，另外**128**个可以根据不同地区使用的符号不同进行扩充（例如法语、俄语等等）。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

汉字是一种特殊的字符，要让计算机能够处理汉字信息，也必须解决汉字的编码问题。但是汉字的处理比西文字符要复杂很多。首先，**汉字的数量巨大**，常用的汉字就有**3000~5000**个，显然汉字的输入就是一个问题，像现在西文键盘一样的汉字键盘显然是不可能的！其次，**汉字是象形文字，同音字多，显示和输出也比西文麻烦很多**。所以，汉字尽管是一类字符，但由于计算机开始是为西文信息处理而生和汉字字符的特殊性，汉字字符的编码要比西文字符复杂的多！

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

为了让计算机能够识别、处理汉字，也需要对汉字进行编码，**即将汉字信息转化为二进制编码**。但是汉字的处理过程要复杂的多，譬如如何直接使用西文标准键盘输入汉字，如何输出各种字形和字体的汉字等，

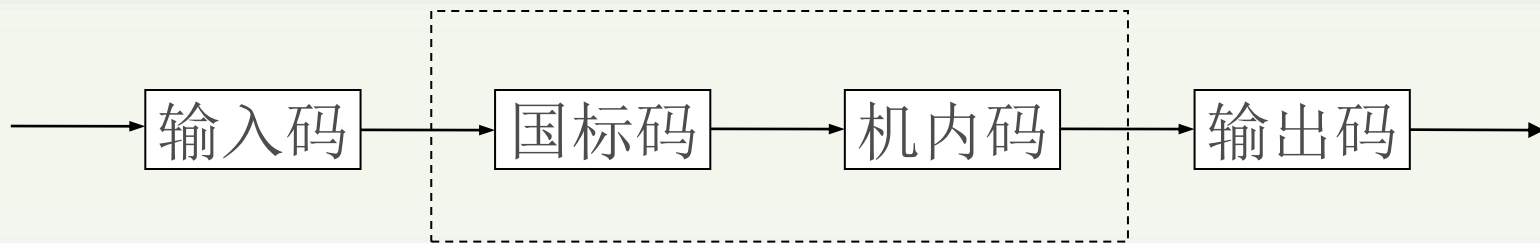
在不同的汉字处理阶段会有不同的要求，为此，人们设计了适用于汉字不同处理阶段的多种编码。主要有四类：

- 如何通过西文键盘输入汉字—**输入码**
- 汉字如何在计算机中存储—**机内码、国标码**
- 计算机中存储的汉字如何输出—**输出码**

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示



#### (1) 区位码、国标码、机内码

和西文字符**ASCII**码的产生一样，为了适应计算机处理汉字的需要，**1981**年我国国家标准局颁布了《信息交换用汉字编码字符集—基本集》(**GB2312-80**)，因此称为“国标码”（也称交换码）。该标准收集汉字**6763**个（其中一级汉字**3755**个，按汉语拼音排列；二级汉字**3008**个，按偏旁部首排列），英、日、俄字母及其他符号**682**个，共**7445**个汉字等字符。



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

- 国标编码的规则：

按规定（实际上是约定）的顺序把所有的汉字编码字符集排列成一个**94×94**的二维表（矩阵），共有**8836**个位置。这样，字符集中的每个汉字字符就有一个“行号”和“列号”，分别称为“区号”和“位号”。一个区号和位号可以唯一确定字符集中的汉字字符。由十进制**2**位区号和**2**位位号组成一个编码，这个编码称为区位码。**GB2312-80**汉字编码字符集共有**7445**个字符，已占用了**7445**个区位，还有**8836-7445=1391**个区位是空的，留作备用。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ► 计算机中汉字信息表示

1 2 3 ..... 94

94

1 2  ... 9	01-09 区：符号和数字
...	
16 ... 55	16-55区：一级汉字
56 ... 87	56-87区：二级汉字
88 ... 94	

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

例如，“国”字处在二维表的第**25**行、第**90**列，它的区位编码为**25 90**，表示为**16**进制为**195A H**(注意：区码和位码单独转换)。

为了避免与通信用的控制码（**0~31**，即**16**进制**00H~1FH**）冲突，使区位编码的区号和位号都增加**32**，这样得到的编码就是国标码。例如，“国”字的国标码为**57 122**，表示为**16**进制为**397A H**。

国标码从理论上来说就可以在计算机内表示和存储了，区号和位号分别用**1**个字节表示，即存储一个汉字就是存储其国标码，占用**2**个字节。例如“国”字的国标码用**2**个字节表示为：**0011100101111010B**。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

但是由于其每个字节的最高位为**0**（因为最大的区码和位码为**94+32**，小于**128**），而前面字符的**ASCII**用一个字节表示，其最高位也是**0**，那么，**计算机中连续的两个字节表示的是两个ASCII码字符还是一个汉字字符呢？**所以为必须加以区分！

- 处理的方法是：

把一个汉字看成是两个扩展**ASCII**码，即把表示一个汉字字符的两个字节的最高位置位“**1**”，这种高位为**1**的双字节编码就是汉字字符在计算机中表示（存储）时的编码，称为**机内码**。例如，“国”字的机内码为：

**101110011111010B**。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ► 计算机中汉字信息表示

- 特别要注意区位码、国标码、机内码三者之间的联系：

汉字基本字符集按**94×94**区位分布，按区号和位号进行编码就是“**区位码**”，区位码由于和通信用控制码有冲突，需要进行改造，统一使区号和位号都增加**32**，这种编码就是**国标码**；国标码在计算机中表示时又与**ASCII**码有冲突，也需要进行改造，把存储汉字的两个字节的最高位都置为**1**，这种双字节编码就是汉字的**机内码**。**汉字在计算机内部表示（存储）用的编码是机内码。**

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

##### (2) 输入码

汉字的输入码又称外码。英文字符输入时，只要按键盘上的字符就可以，输入码和机内码是一致的。而汉字则不同，没有专门的汉字键盘，因此必须使用预先设计的方法，将汉字音、形、义等有关要素变成英文键盘上相关按键的组合，从而实现汉字的输入。这种汉字和键盘字符组合的对应方式就是汉字的输入编码。

一种汉字输入法就是定义和实现了一种汉字输入码，通过键盘输入一个汉字的外码，计算机就能识别该汉字（实际上是映射出该汉字的国标码或机内码—汉字在计算机内的二进制表示）。



# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 计算机中汉字信息表示

#### (2) 输入码

根据汉字特征信息的不同，汉字输入码主要分为：

- ✓ 拼音编码：全拼、双拼、微软拼音、智能**ABC**、搜狗拼音等
- ✓ 数字编码：区位码
- ✓ 字形编码：五笔字型

一种汉字输入法实质上是一个转换程序，接收汉字的输入码，快速确定要输入的汉字，自动转换成该汉字的机内码。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ► 计算机中汉字信息表示

#### (3) 输出码

- 在计算机内部，汉字表示（存储）位二进制数；
- 直接输出？ ×
- 直观输出（和人们日常用的汉字一样） ✓

#### • 怎么解决呢？

由于汉字是一种象形文字，表现为一种图形形式，因此，就需要把机内码和该汉字的形状描述对应起来，从而实现输出。

- 描述汉字形状的数据就是汉字字形编码，又称**输出码**。  
常用的有两种编码：**点阵编码**，**矢量编码**

## 第六节 数据的表示与编码

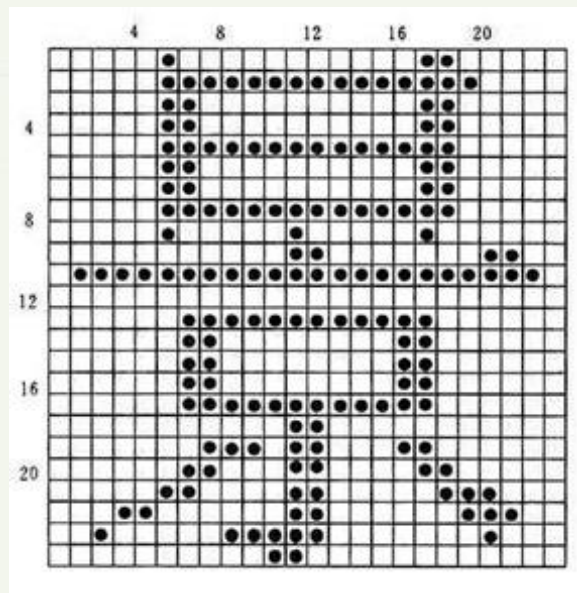
### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

#### (3) 输出码

- 点阵字型编码：也叫位图字体编码，每个字型都以一组二维像素信息（点阵）表示。每个像素点用一个二进制位表示，黑点表示为“1”，空白表示为“0”。

- ✓ 根据汉字的输出精度要求，汉字点阵有不同的密度。汉字字型点阵有 **16×16**点阵、**24×24**点阵、**32×32**点阵等。



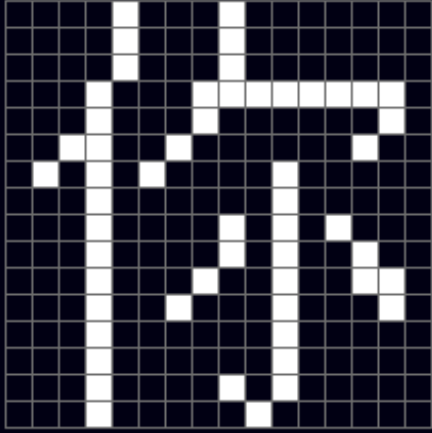
## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

##### (3) 输出码

例如，**16\*16**点阵的汉字。每个汉字需要  
**16×2=32Bytes**

中文字模	位代码	字模信息
	0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0	0x08, 0x80
	0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0	0x08, 0x80
	0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0	0x08, 0x80
	0 0 0 1 0 0 0 1 1 1 1 1 1 1 1 0	0x11, 0xfe
	0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0	0x11, 0x02
	0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0	0x32, 0x04
	0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0	0x54, 0x20
	0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0	0x10, 0x20
	0 0 0 1 0 0 0 0 1 0 1 0 1 0 0 0	0x10, 0xa8
	0 0 0 1 0 0 0 0 1 0 1 0 0 1 0 0	0x10, 0xa4
	0 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0	0x11, 0x26
	0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0	0x12, 0x22
	0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0	0x10, 0x20
	0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0	0x10, 0x20
	0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0	0x10, 0xa0
	0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0	0x10, 0x40

将计算机可以表示的汉字字形信息设计后，存放在一个文件中-字库文件。例如**HZK12**、**HZK24**，**HZK32**和**HZK48**字库及黑体、楷体和隶书字库等。字库中汉字的组织通常也都是按照区位码组织的，这样很容易实现机内码和输出码的映射。那么，一个汉字的点阵编码（输出码）是什么呢？

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ► 计算机中汉字信息表示

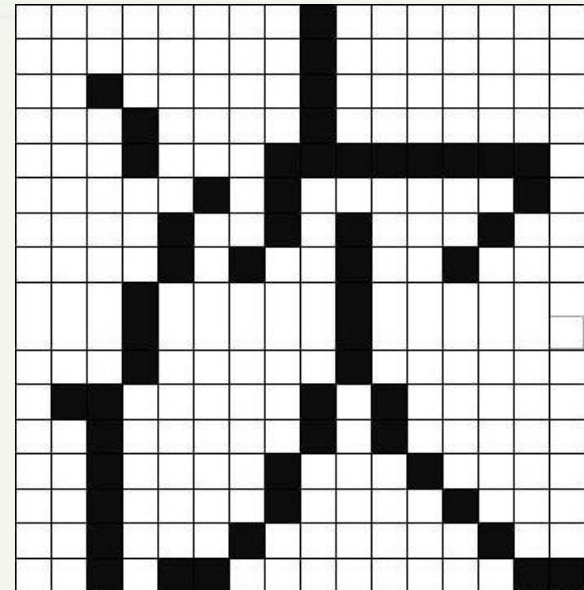
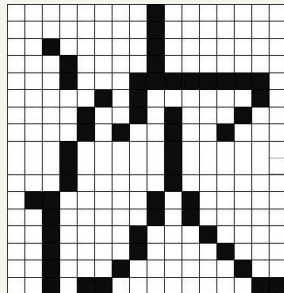
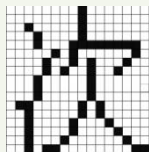
#### (3) 输出码

**问：**如果汉字采用**24×24**点阵表示字型，那么**100**个汉字的字型信息所占用的字节数是多少？

#### ✓ 特点：

- 显示速度快
- 字型清晰度与点阵规模有关
- 不易放大（放大失真）

次



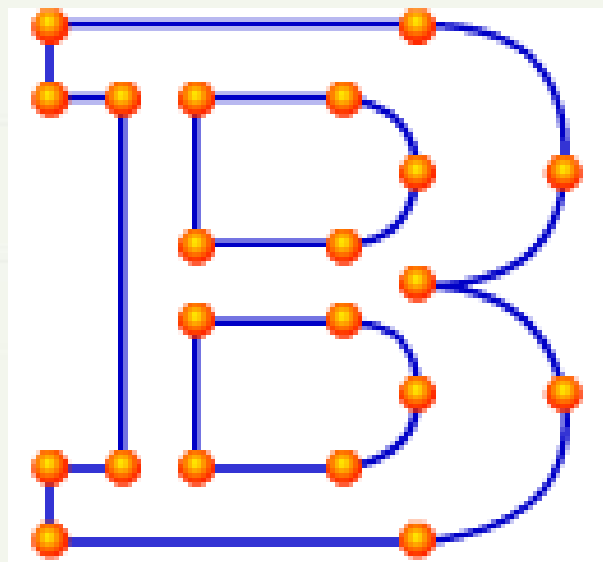
## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ► 计算机中汉字信息表示

##### (3) 输出码

- 矢量字型编码：对汉字的字形轮廓特征进行刻画，然后把这些轮廓特征信息用编码表示出来，比如一个笔划的起始、终止坐标，半径、弧度等等。这样的编码称为矢量字体或轮廓字体，在输出时不像点阵编码直接输出，而是要进行一系列的运算，生成最终输出的汉字点阵。





## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 计算机中汉字信息表示

#### (3) 输出码

- ✓ **Windows**的字体文件在**FONT**S文件夹中，矢量字体文件扩展名**ttf**。点阵字体的扩展名是**fon**。
- ✓ 点阵编码和矢量编码的区别：
  - 点阵编码：存储方式简单，无需转换直接输出，但字形放大后产生的效果差，而且同一种字体不同的点阵需要不同的字库。
  - 矢量编码表示形式精度高，字的大小在变化时能保持字形不变，但输出是需要进行许多计算。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### Unicode码

- ◆ **Unicode**即统一码，又称万国码，是一种以满足跨语言、跨平台进行文本转换、处理的要求为目的设计的计算机上字符编码。它为每种语言中的每个字符设定了统一并且唯一的二进制编码。**Unicode** 的编码方式与 **ISO 10646** 的通用字元集（亦称通用字符集）概念相对应，使用**16位**的编码空间。也就是每个字符占用**2个字节**。
- ◆ 对于中文而言，**Unicode 16**编码里面已经**包含了GB18030**里面的所有汉字（**27484**个字）
- ◆ **Unicode**扩展自**ASCII**字元集。其使用**16位元**编码，并可**扩展到32位**，这使得**Unicode**能够表示世界上所有的书写语言中可能用於电脑通讯的字元、象形文字和其他符号，这使其有可能成为**ASCII**的替代者。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 静态图像

静态图像是目前信息的一种重要表现形式。

## ■ 计算机表示静态图像的两种方式：

- 位图图像
- 矢量图

### (1) 位图图像

位图表示图像的方法中，图像被分成**像素矩阵**，也称**点阵**，每个像素是一个小点。像素的大小取决于**分辨率**。

把图像分成像素之后，每一个像素被赋值为一个**位模式**。模式的尺寸和值取决于图像。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 静态图像

- **黑白图像**：1位模式（2种颜色）
- **多级灰度图像**：2位模式（4种灰色级），位模式越大，表示的明暗变化越细；一般用8位模式（1个字节）
- **彩色图像**：每个彩色像素被分解为红绿蓝（RGB）三种主色，每种主色用有一个位模式（通常用1个字节），表示该种颜色的强度。因此一个彩色像素需要用3个字节（24位）表示。

24位模式又称为真色彩模式，可以提供 $256 \times 256 \times 256 = 16777216$ 种色彩，能表示自然的所有色彩。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 静态图像

<i>Color</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>	<i>Color</i>	<i>Red</i>	<i>Green</i>	<i>Blue</i>
Black	0	0	0	Yellow	255	255	0
Red	255	0	0	Cyan	0	255	255
Green	0	255	0	Magenta	255	0	255
Blue	0	0	255	White	255	255	255

一张照片由模拟数据组成。数据密度（色彩）因空间变化，要在计算机中表示，就数据需要采样。采样通常被称作扫描。采样的样本称为像素。在图像扫描中，要决定对每英寸的方块或线条需要记录多少像素。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 静态图像

#### ■ 图像的基本属性

- 图像分辨率，扫描分辨率
- 像素深度或位深度

#### ■ 图像的数字化过程

- 采样
- 灰度级的量化

#### ■ 位图图像分类

- 单色图像
- 灰度图像
- 彩色图像



## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 静态图像

➤ 常见的位图图像编码格式有：

✓ 位图文件格式-**BMP**（**bitmap**的缩写）格式。它是一种与硬件设备无关的图像文件格式。

✓ 标签图像格式- **TIFF**（**Tagged Image File Format**）格式，是位图文件格式的一种。在图像专业领域使用较广泛的一种编码形式，主要用来存储照片和艺术图等对图像质量要求较高的平面图像。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 静态图像

#### ➤ 图形压缩编码

- 位图图像编码，体积太大，存储和传输都产生很大压力
- 静态图像信息具有一定的规律，在保证其基本信息正确的前提下，可以适当通过一定的算法缩小其数据量。
- 数据压缩方案有两类：
  - **无损压缩**：压缩后信息表达的质量没有下降，数据量减小。
  - **有损压缩**：指在影响信息表达质量的情况下，尽可能减小数据量。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 静态图像

- 当前最主流的图像压缩方式是**JPEG**（Joint Photographic Experts Group，联合图像专家组）
  - 编码格式，文件后缀名为“**.jpg**”或“**.jpeg**”
  - **JPEG**压缩技术即能支持无损压缩，也支持大压缩比的有损压缩。

**JPEG**格式压缩的主要是高频信息，对色彩的信息保留较好，适合应用于互联网，可减少图像的传输时间。

# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 静态图像

#### (2) 矢量图像

位图图像编码数据量尺寸太大、重新调整图像大小麻烦，放大位图图像意味着扩大像素，放大后看上去很粗糙。

矢量图图像编码方法并不存储每个像素的位模式。将图像分解成一些曲线和直线的组合，其中每一曲线或直线由数学公式表示。

当图像要显示或打印时，将图像的尺寸作为输入传给系统。系统重新设计图像的大小并用相同的公式画出图像。

每次调整矢量图时，计算机将绘图公式重新估算一次，并根据新公式画出图像，由于重新估算公式地计算量远小于调整像素，因此可以有效避免屏幕抖动现象。

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 动态图像-视频

视频是图像在时间上的表示（称为帧frames）。电影是一系列帧一张接一张播放而形成运动的图像。视频是随空间（单个图像）和时间（一系列图像）变化的信息表现。

- 静态图像编码是视频编码的基础。
- 每一幅图像或帧转化成一系列位模式表示，这些图像组合起来就可以表示视频。

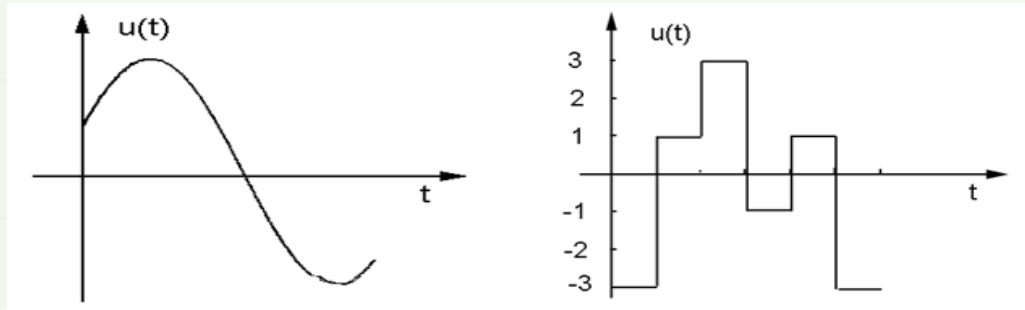
- 最有影响的视频编码技术是**MPEG**（**Moving Pictures Experts Group**，动态图像专家组）
- **MPEG**标准的视频压缩编码技术主要利用了具有**运动补偿的帧间压缩编码技术**以减小**时间冗余度**

# 第六节 数据的表示与编码

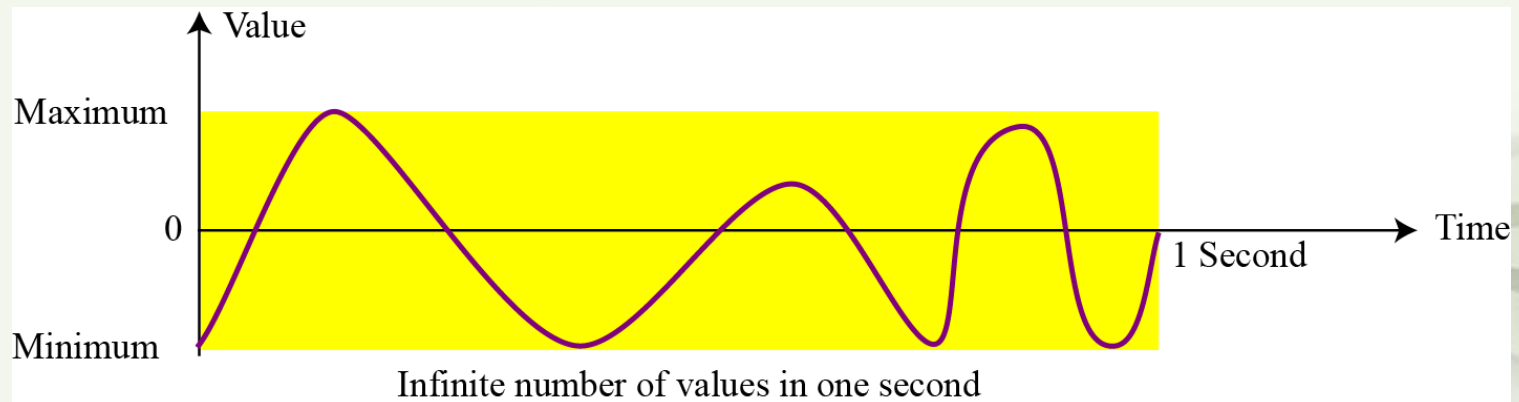
## ■ 计算机中的信息的表示

### ➤ 音频

音频表示声音或音乐，音频是不可数的，是模拟信号。计算机要处理，必须转化成二进制的数字信号。



一个  
音频  
信号

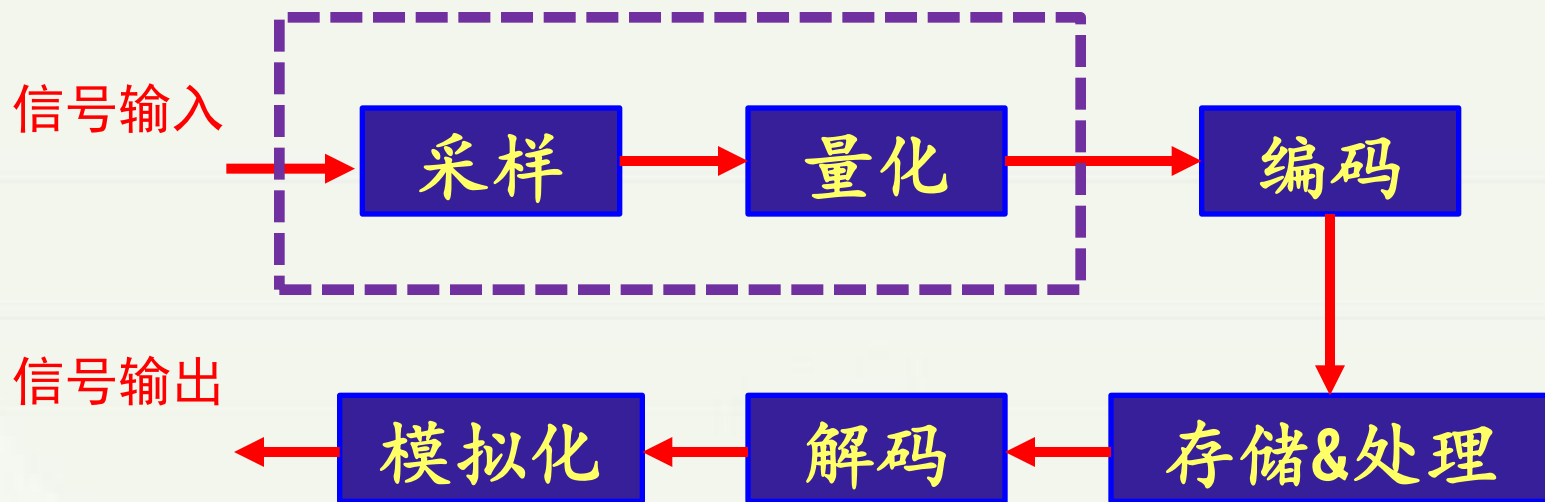




## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 音频



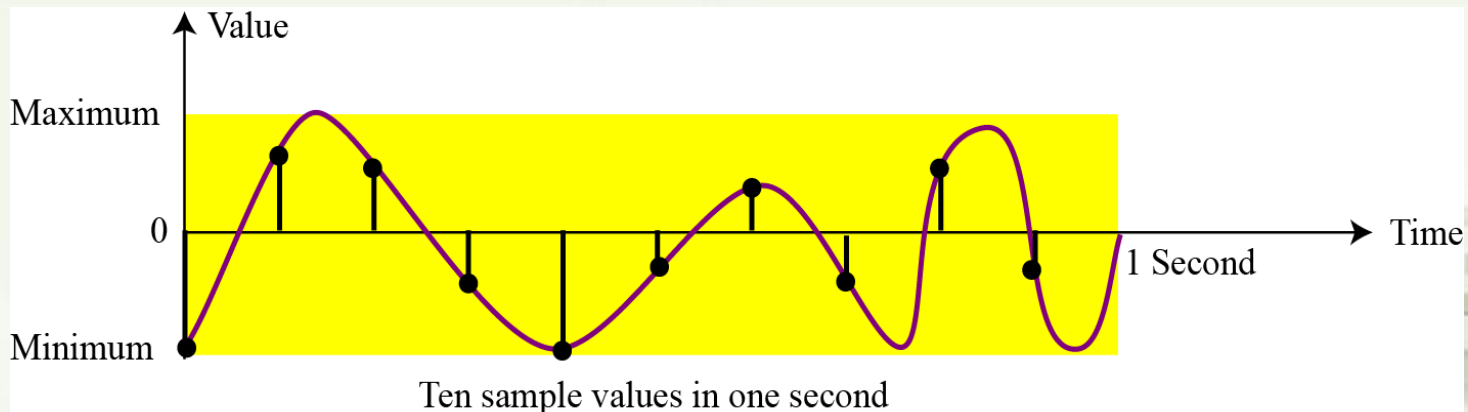
# 第六节 数据的表示与编码

## ■ 计算机中的信息的表示

### ➤ 音频

### ◆ 采样：离散时间

- 离散时间信号：采样得到的信号
- 采样频率：单位时间内的采样次数。
- 均匀采样：采样时间间隔相等
- 奈奎斯特理论：采样频率不低于信号最高频率的两倍，即可将以数字表达的信号还原成原来的信号



一个音频信号的采样

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 音频

#### ◆ 量化：离散幅度

从每个样本测量得到的值是真实值，量化是将样本值截取为最接近整数值的過程。

#### ◆ 编码：

量化的样本值需要被编码成位模式。一些系统为样本赋正值或负值，另一些仅移动曲线到正的区间从而只赋正值。

称每样本位的数量(位深度 **bit depth**)为 **B**，每秒样本数为 **S**，则需要为每秒的音频存储  **$S \times B$**  位，该乘积称为 **位率 R(bit rate)**。一般质量 **16** 位，立体声 **32** 位。

## 第六节 数据的表示与编码

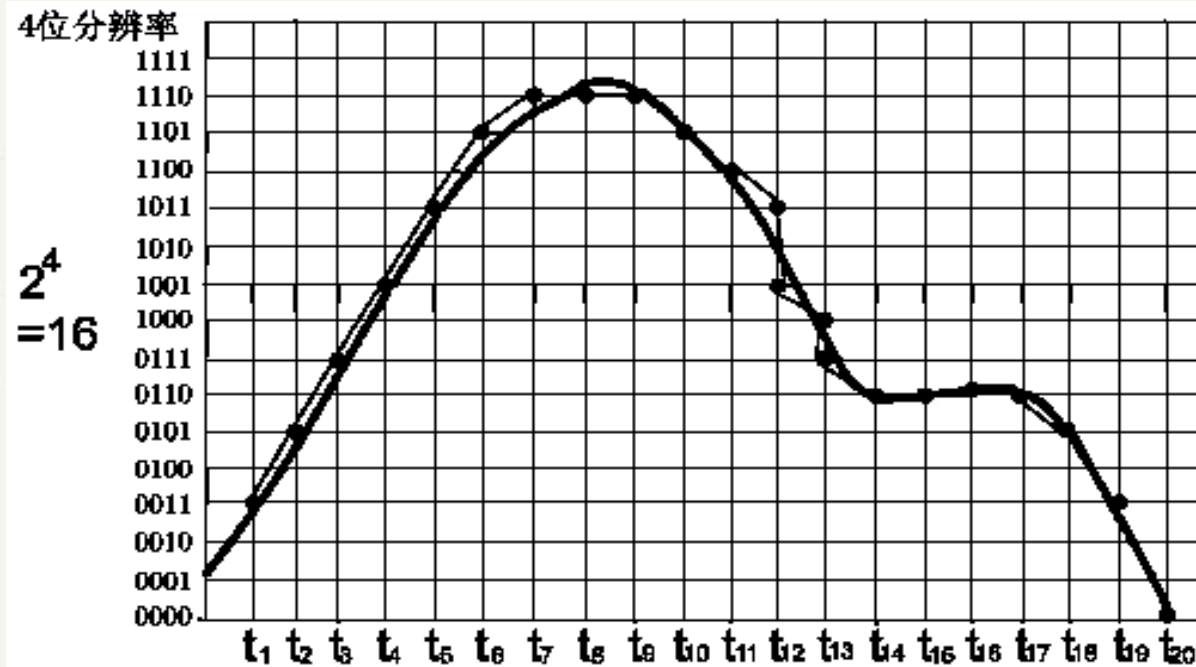
### ■ 计算机中的信息的表示

#### ➤ 音频

#### ■ 基本的音频编码-PCM (Pulse Code Modulation, 脉冲编码调制)

#### ✓ PCM的优缺点：

- 优点：音质好；
- 缺点：数据量大；

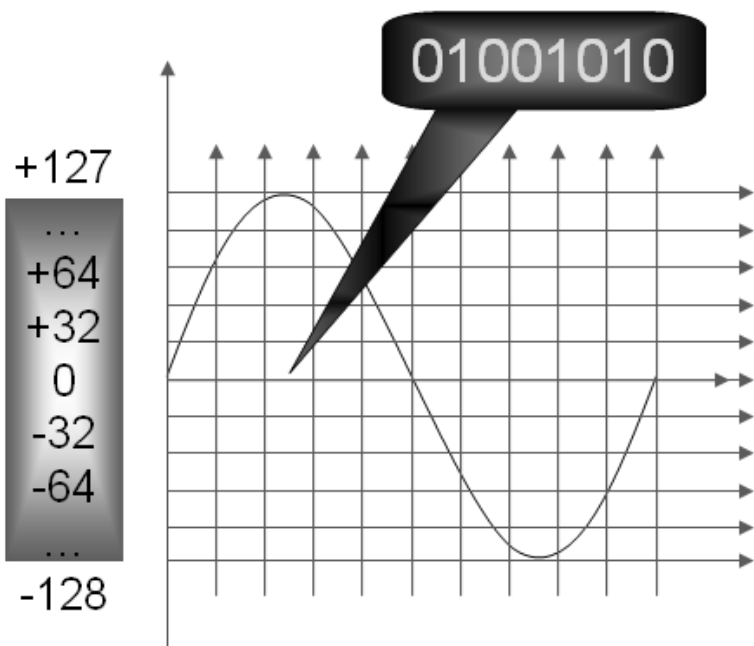


采样点	t1	t2	t3	t4	t5	t6	t7	...
幅值	0011	0101	0111	1001	1011	1101	1110	...

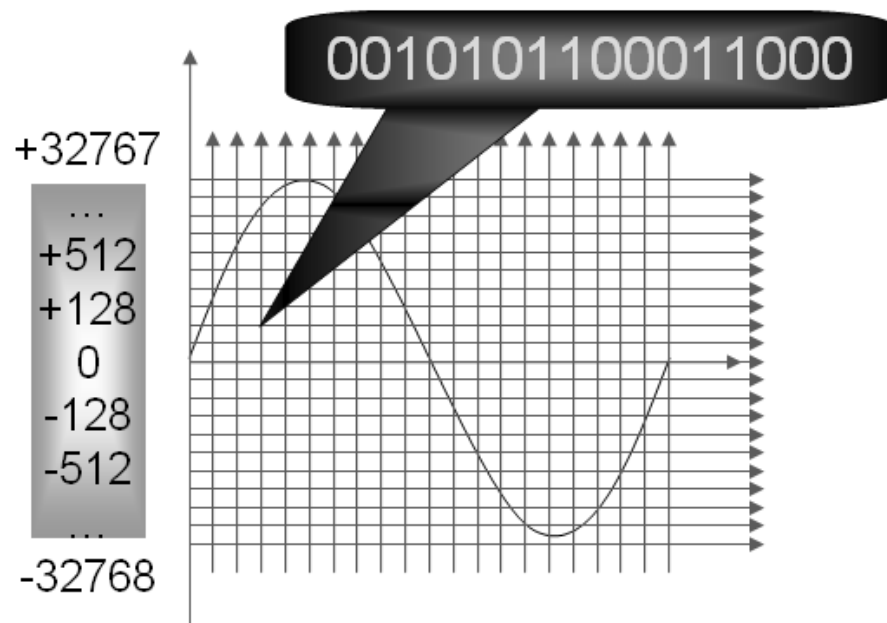
## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 音频



采样频率**11KHz**  
8位量化



采样频率**22KHz**  
16位量化

## 第六节 数据的表示与编码

### ■ 计算机中的信息的表示

#### ➤ 音频

- WAV波形音频格式

Windows系统的音频格式，文件.WAV，也是事实上的通用音频格式。通常使用WAV格式保存未压缩的音频，即PCM编码后得到的音频，故常称WAV文件为波形文件。

- mp3 (MPEG Layer 3) 音频格式

当今音频编码的主流格式MPEG 标准的一个修改版。它采用每秒44100个样本以及每样本16位。位率达到 $44100 \times 16 = 705,600 \text{ b/s}$ ，并使用人耳无法识别的数据压缩方法进行有损压缩。MP3可以做到12:1的压缩比并保持音质基本可接受。



# 本章小结

- 计算机系统的分层结构
- 冯·诺依曼理论的基本内容
- 计算机系统的组成
- 计算机硬件系统的组成
- 计算机硬件系统各个部件的主要功能、性能指标
- 信息的表示与编码

# 操作技能要求

## ■ **WINDOWS**操作系统的基本使用

文件管理，（磁盘管理），控制面板，附件

## ■ **OFFICE** 套件的基本使用

- **WORD**（文本录入、基本排版、图文混排、表格等）

- **EXCEL**（工作簿、工作表、单元格的概念  
各种数据的录入、格式化工作表、基本数据处理）

- **PPT**（各种元素的插入和编辑、动画等）

# 练习题

**1.**假设用**2**个字节（**16**位）表示一个浮点数，规定前**6**位表示阶码（含阶符），后**10**位表示尾数（含数符），求下面编码的真值。

**(1) 1001100010101000**

**(2) 0000111011101000**

**2.**假设余码系统是**6**位模式，分别计算 **$(12)_{10}$** ， **$(-30)_{10}$** 的余码。

**3.**假设编码长度为**2**个字节（约定符号位**1**位，指数**5**位，尾数**10**位），按**IEEE** 标准 **754**的方式写出实数 **$(-26.5)_{10}$** 的编码

**4.**区位码、国标码、机内码三者之间是什么关系？