计算机系统知识

# 计算机系统基础知识

## 1.1.1 计算机系统硬件基本组成

计算机的基本硬件系统由运算器、控制器、储存器、输入设备、输出设备5大部件组成。

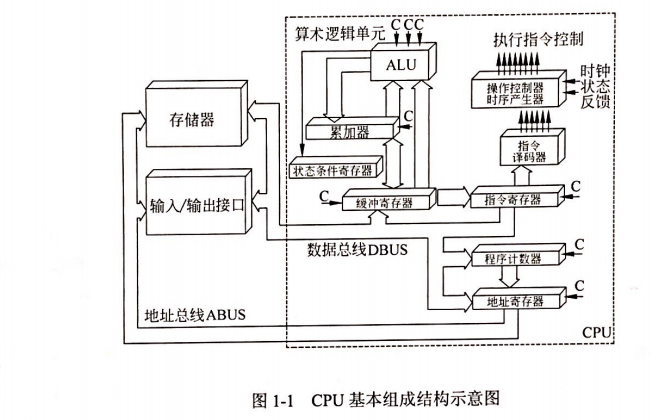
## 1.1.2 中央处理单元

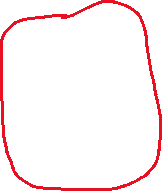
中央处理单元（CPU）负责获取程序指令、对指令进行译码并加以执行。

### CPU 的功能

1. 程序控制。CPU通过执行指令来控制程序的执行顺序。
2. 操作控制。一条指令的实现要若干操作信号来配合完成，CPU产生每条指令的操作信号并将操作信号送往对应的部件，控制相应的部件按指令的功能要求进行操作。
3. 时间控制。CPU对各种操作进行时间上的控制，即指令执行过程中操作信号的出现时间、持续时间及出现的时间顺序都需要进行严格控制。
4. 数据处理。CPU通过对数据进行算数运算及逻辑运算等方式进行加工处理，数据加工处理的结果被人们所利用。所以，对数据的加工处理也是CPU 最根本的任务。

### CPU的组成





**CPU**主要由运算器、控制器、寄存器和内部总线组成

1. 运算器

运算器由算术逻辑单元、累加寄存器、数据缓冲寄存器和条件状态寄存器组成，它用于完成计算机的各种算数运算和逻辑运算。

运算器的主要功能：

1. 执行所有算术运算，加减乘除等
2. 执行所有的逻辑运算并进行逻辑测试。与或非零测试或两值得比较。

各组成部件得功能：

1. 算数逻辑单元（ALU）。ALU负责处理数据，实现对数据的算数运算和逻辑运算
2. 累加寄存器（AC）。其功能为当运算器的算数逻辑单元执行算术或逻辑运算时，为ALU提供一个工作区。例如，在执行一个减法运算前，先将被减数去除暂存在AC中，再从内存器中取出减数，然后同AC的内容相减，将所得的结果送回AC中。运算的结果是放在累加器中的，运算器中至少要有一个累加器。
3. 数据缓冲寄存器（DR）。在对内存储器进行读写操作的时，用DR暂时存放由内存储器读/写的一条指令和一个数据字，将不同时间段内读/写的数据隔离开来。DR的主要作用：1、作为CPU和内存、外部设备之间数据传送的中转站2、作为CPU和内存、外围设备之间在操作速度上的缓冲3、在单累加器结构的运算器中，数据缓冲器还可兼作为操作数寄存器。
4. 状态条件寄存器（PSW）。PSW保存由算术指令和逻辑指令运行或测试的结果建立的各种条件码内容，主要分为状态标志和控制标志，例如结果进位标志（C）、运算结果溢出标志（V）、运算结果为0标志（Z）、中断标志（I）、方向标志（D）。这些标志通常分别由一位触发器保存，保存了当前指令执行完成之后的状态。通常，一个算术操作产生一个运算结果，而一个逻辑操作产生一个判决。
5. 控制器

控制器用于整个CPU的工作，它决定了计算机运行的自动化。控制器一般包括指令控制逻辑、时序控制逻辑、总线控制逻辑和中断控制逻辑等几个部分。

指令控制逻辑要完成取指令、分析指令和执行指令的操作。其过程为取指令、指令译码、按指令操作码执行、形成下一条指令地址等

1. 指令寄存器（IR）。当CPU 执行一条指令时，先把它从内存器中取到缓冲寄存器中，再送入IR暂存，指令译码器根据IR 的内容产生各种微操作指令，控制其他组成部件的工作。
2. 程序计数器（PC）。PC有寄存信息和计数两种功能。程序的执行分为两种情况，一是顺序执行，二是转移执行。在程序开始执行之前，程序将起始地址送入PC，该地址在程序加载到内存时确定，因此PC的内容就是程序第一条指令的地址。执行指令时，CPU 自动修改PC的内容，以便其保持的总是将要执行的下一条指令的地址。大多数指令按顺序来执行，所以修改的过程只是简单地相对PC加1。当遇到转移指令时，就根据当前地址加上向前或向后的位移量，或者直接给出转移的地址。
3. 地址寄存器（AR）。AR保存当前CPU 所访问的内存单元的地址。
4. 指令译码器(ID)。指令包含操作码和地址码两部分，执行给定的指令，必须对操作码进行分析。指令译码器就是对指令中的操作码字段进行分析解释，再向操作控制器发出具体的控制信号。

时序控制逻辑要为每条指令按时间顺序提供应有的控制信号。总线逻辑是为多个功能部件服务的信息通路的控制电路。中断控制逻辑用于控制各种中断请求，并根据优先级的高低对中断请求进行排队，逐个交给CPU处理。

1. 寄存器组

寄存器组可分为专用寄存器和通用寄存器。运算器和控制器中的寄存器是专用寄存器，其作用是固定的。通用寄存器用途广泛并可由程序员规定其用途，其数目因处理器不同有所差异。

### 3.多核CPU

核心又称内核，是CPU最重要的组成部分。

多核即在一个单芯片上面集成两个甚至更多个处理器内核，其中每个内核都有自己的逻辑单元、控制单元、中断处理器、运算单元、一级Cache、二级Cache共享或独有。

多核CPU系统最大的优点是可满足用户同时进行多个任务处理的要求。

## 1.1.3 数据表示

各种数值在计算机中表示的形式称为机器数。机器数对应的实际数值称为数的真值。

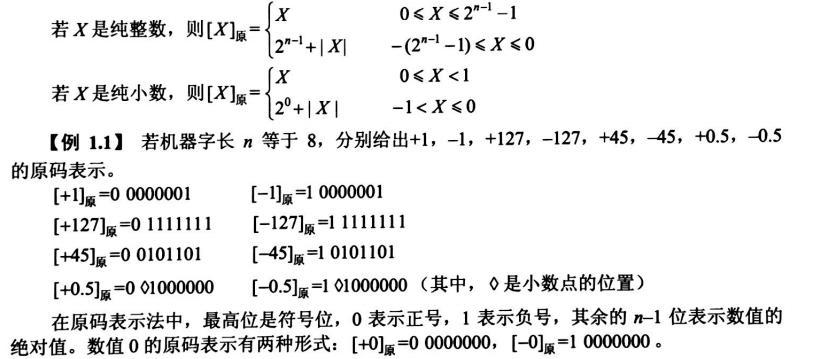
机器数有无符号数和带符号数之分。

对于无符号数，若约定小数点的卫视在机器数的最低位之后，则为纯整数；若约定的小数点位置在机器数的最高位前，则是纯小数。

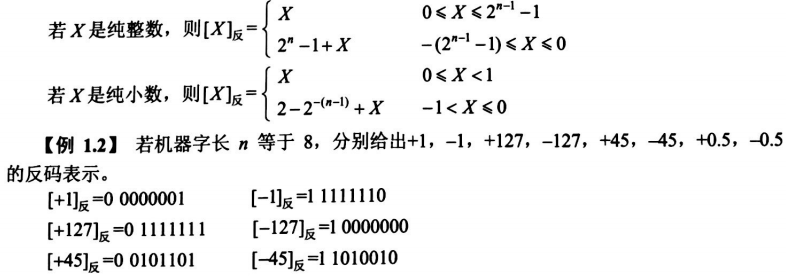
对于带符号数，机器数的最高位表示正、负的符号位，其余位则表示数值。

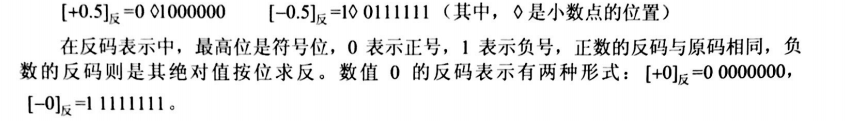
1. 原码、反码补码和移码

（1）原码表示法。

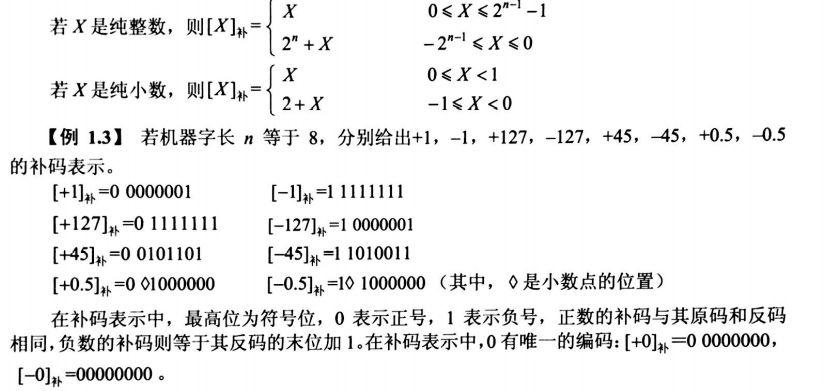


1. 反码表示法

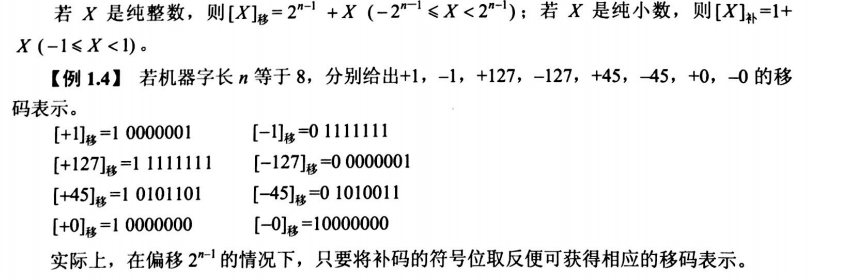




1. 补码表示方法。



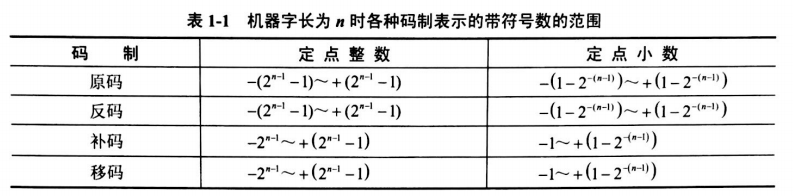
1. 移码表示法



1. 定点数和浮点数

（1）定点数。就是小数点的位置固定不变的数，小数点的位置通常由两种约定方式：定点整数（纯整数，小数点在最低有效数值位之后）和定点小数（纯小数，小数点在最高有效数值为之前）。

设机器字长位n，各种码制下带符号数的范围如表所示



（2）浮点数。

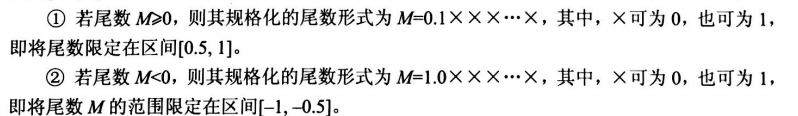
当机器字长为n时，定点数的补码和移码可表示2^n各数，而其原码和反码只能表示2^n-1个数（0的表示占用了两个编码），因此定点数所能表示的数值范围比较小，在运算种很容易因结果超出范围而溢出。浮点数时小数点位置不固定的数，他能表示更大范围的数。

一个二进制数N可以表示为更一般的形式N=2^e\*f，其中E为阶码，f称为尾数，用阶码和尾数表示的数称为浮点数，这种表示数的方法称为浮点表示法。

在浮点表示法种，阶码为带符号纯整数，尾数为带符号的纯小数，浮点数的表示格式



浮点数所能表示的数值范围主要由阶码决定，所表示的精度则有尾数决定。为了充分利用尾数来表示更多的有效的数字，通常采用规格化浮点数。规格化就是将尾数的绝对值限定在区间[0.5，1].档位数用补码表示时，需要注意如下问题：



如果浮点数的阶码用R位的移码表示，尾数用M位的补码表示，则这种浮点数所能表示的数值范围如下：



（3）工业标准IEEE754。

该标准的表示形式如下



其中，（-1）^s为该浮点数的数符，当S为0时表示正数，s为1时表示负数；E为指数（阶码），用移码表示；最后部分则为尾数，其长度为P位，用原码表示；

根据IEEE754标准，被编码的值分为3种不同的请况：规格化的值、非规格化的值和特殊的值，规格化值为最普遍的情形。

**规格化的值：**

当阶码部分的二进制不全为0也不全为1时，所表示的是规格化的值。例如在单精度浮点格式下，阶码为10110011时，偏移量为+127（01111111），则其表示的真值为10110011-01111111=00110100，转换为十进制为52；

**浮点数的运算**

设有浮点数X=M\*2^j，Y=N\*2^j，求他们的运算过程要经过对阶、求尾数和（差）、结果规格化并判溢出，舍入处理和溢出判别等步骤。

1. 对阶：使两个的阶码相同。
2. 求尾数和
3. 结果规格化并判溢出：若运算结果所得的尾数不是规格化的数，则需要进行规格化处理。当尾数溢出时，需要调整阶码。
4. 舍入处理
5. 溢出判别：以阶码为准，若阶码溢出，则运算结果溢出；若阶码下溢（小于最小值 ），

则结果为0；否则结果正确，无溢出。

## 1.1.4 校验码

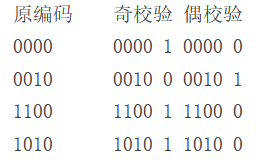
数据校验码是一种常用的用带有发现某些错误或自动改错能力的数据编码的方法。它的原理是加进一些冗余码，使合法数据编码出现某些错误时，就成为非法编码。

### 奇偶校验码

实现的具体方法是为了一个字节补充一个二进制位，称为校验位，使字节的8位和该校验位含有1值得个数为奇数或者偶数。

在使奇数个1的方案进行校验时，称为奇校验；反之，称为偶校验。

这种方法只能发现一位错，或奇数位错，但不能确定是哪一位错，也不能发现偶数个位错。

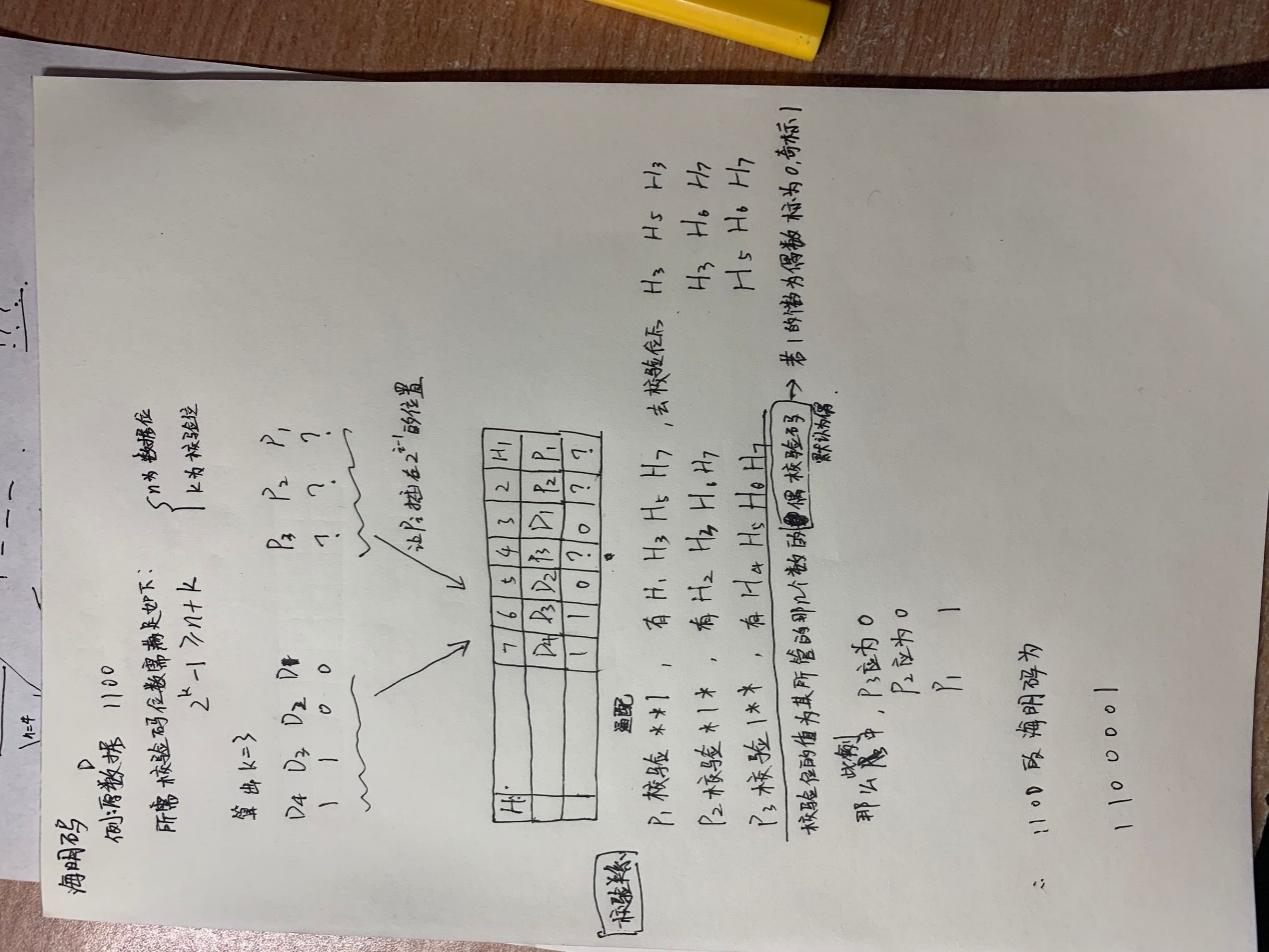


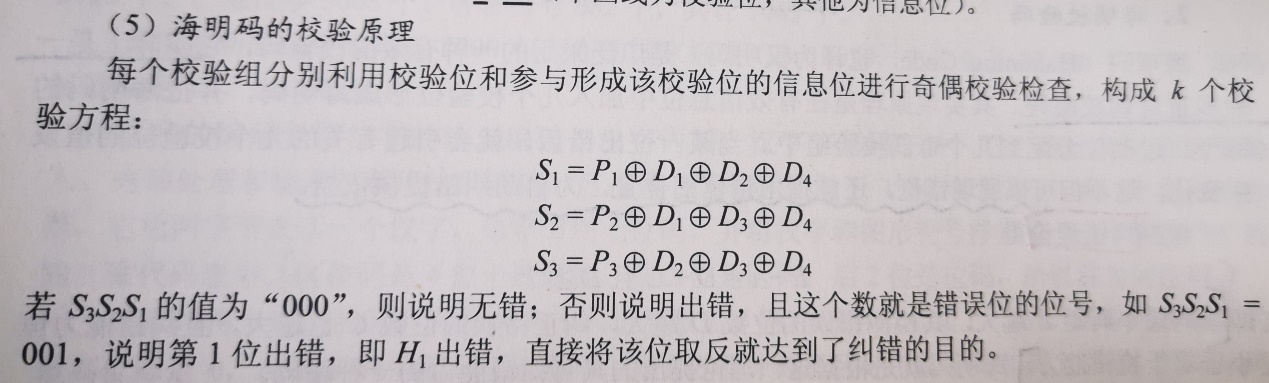
### 海明码校验

海明码（Hamming Code）是由贝尔实验室的Richard Hamming 设计的，是一种利用奇偶性来检错和纠错的校验方法。海明码的构成方法是在数据位之间的特定位置上插入k个校验位，通过扩大码距来实现检错和纠错。

设数据位是n位，校验位是k位，则n和k必须满足以下关系:







### 3.循环冗余校验码

