

**时间频率/主频：1s完成几次基本动作 CPU决定**

**时钟周期（T）：完成1次基本动作需要几s，最小的时间单位**

**一个任务的时钟周期数：完成这个任务需要几次基本动作**

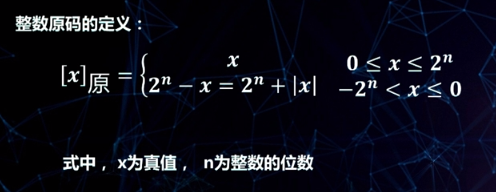
**CPI：（1个程序有很多指令）每条指令周期数，即执行一条指令所需的平均时钟周期数（一个指令平均几次基本动作）**

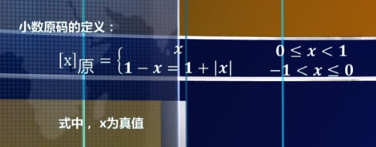
不同程序1个指令需要的基本动作数（CPI）不同，不同程序指令数不同

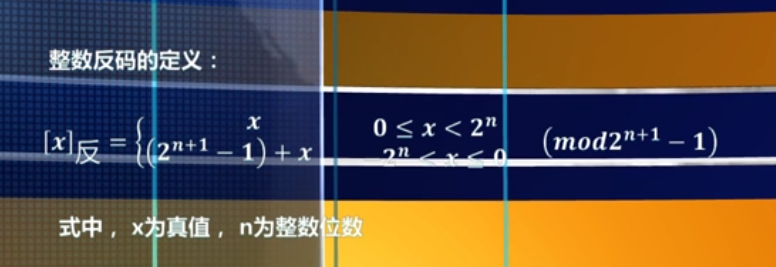
**计算机系统结构主要研究软硬件功能分配和对软硬件界面的确定（**机器语言程序员所看到的传统机器级所具有的属性。**）**

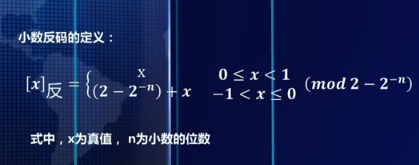
计算机组成：计算机系统结构的逻辑实现

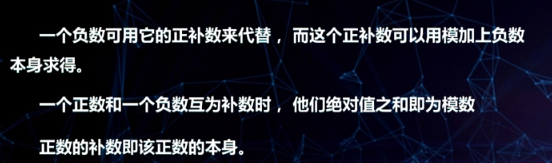
**计算机实现是指计算机组成的物理实现**

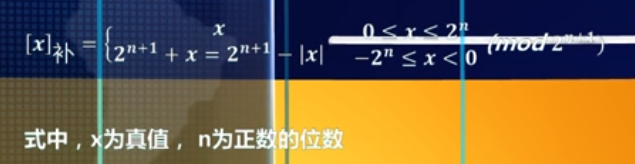












补码的表示方法是：

正数的补码就是其本身；

负数的补码是在其原码的基础上，符号位不变，其余各位取反，最后+1。(也即在反码的基础上+1)

已知一个数的补码，求原码的操作其实就是对该补码再求补码。

同一个数字在不同的补码表示形式中是不同的。比如-15的补码，在8位二[进制](https://baike.baidu.com/item/%E8%BF%9B%E5%88%B6)中是11110001，然而在16位二进制补码表示中，就是1111111111110001。

先看一下常见资料给出的定义：（**8位整型**）

原码：－127～127  
一个二进制数左边加上符号位后所得的码。最高位为符号位，"0"表示正，"1"表示负。  
反码：－127～127  
正数的反码——与原码相同。  
负数的反码——符号位为“1”，数值部分按位取反。  
补码：－128～127  
正数的补码——与原码相同。  
负数的补码——将其原码除符号位之外的所有位取反后加1。即“反码＋1”。

-128的补码=1000 0000，并没有原码和反码表示形式

在[IEEE 754](https://zh.wikipedia.org/wiki/IEEE_754)浮点数表示中移码是非标准的，它的偏移值为2k-1，也就是说对于[单精度浮点数](https://zh.wikipedia.org/wiki/%E5%8D%95%E7%B2%BE%E7%A1%AE%E6%B5%AE%E7%82%B9%E6%95%B0)的偏移值为127。

**原码一位乘法：阵列乘法器**

**补码一位乘法：Booth**