# 第4章 数值表示及转换

何宾

2018.03

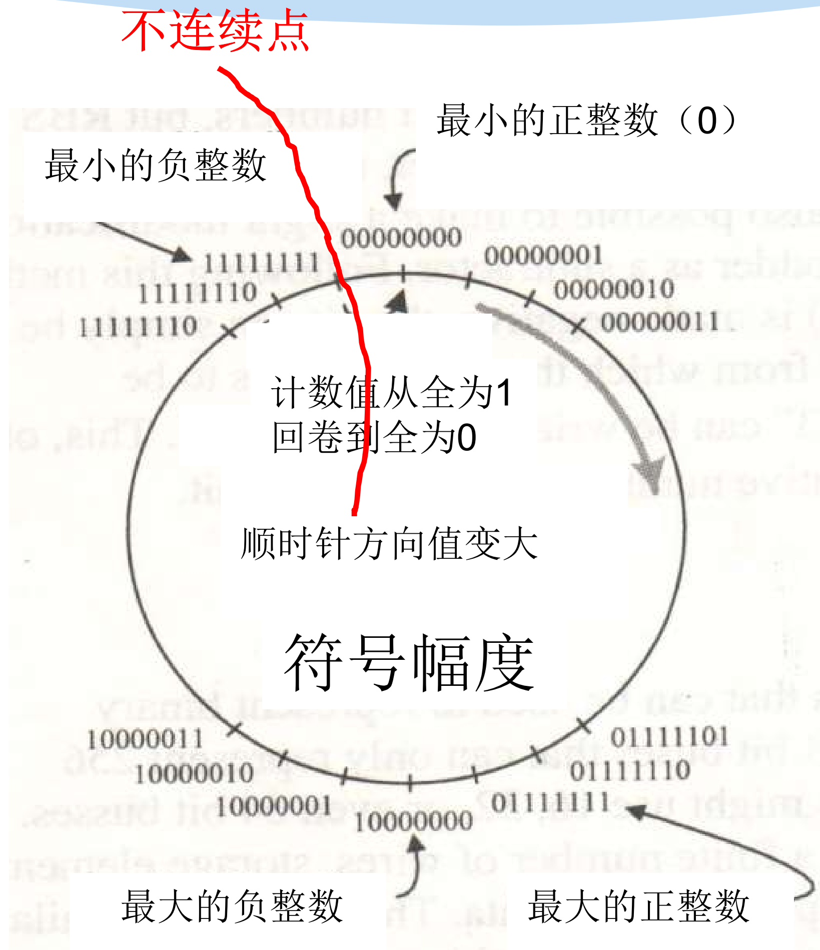
# 负数表示方法

一个N位的系统总共可以表示$2^N$个数，因此一个有用的编码就是使用一半可用的编码（$2^N/2$）表示正数，另一半表示负数。

- 可以将一个比特位设计成一个符号位，用于区分正数和负数。在这种表示方法中，最高有效位（Most Significant Bit，MSB）可以作为符号位。如果：
  - 符号位为1，所表示的数为负数。
  - 符号位为0，所表示的数为正数。
- 在所有可能的负数编码方案中，经常使用两种：
  - 符号幅度
  - 二进制补码

# 负数表示方法
# --符号幅度表示法

- **就是用MSB表示符号位，剩下的位表示幅度，如右图所示。在一个8位的符号幅度系统中：**
  - **十进制数16表示为$(00010000)_2$**
  - **而十进制数－16表示为$(10010000)_2$**



符号幅度表示法

# 负数表示方法
# --符号幅度表示法

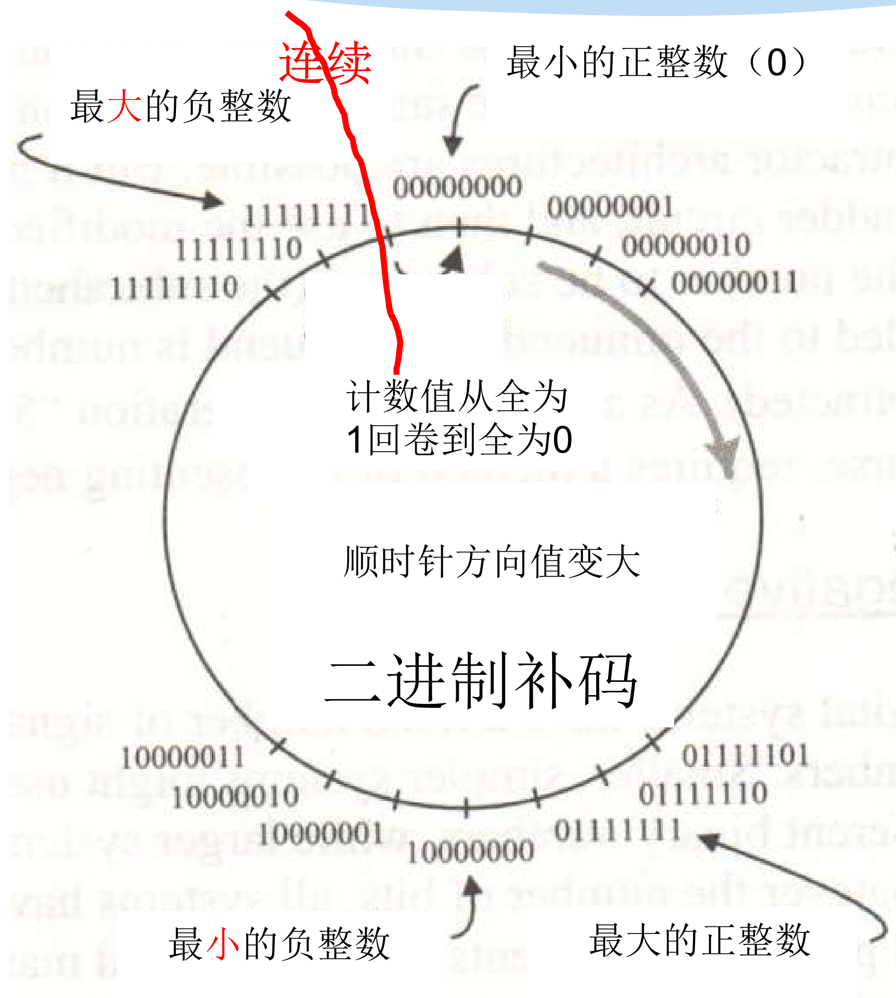
## 符号幅度表示法最不利的方面表现在：

- 如果0到$2^N$的计数范围从最小到最大，则最大的正数将出现在所表示范围的一半的地方，然后，跟随**最大的负数**。
- 最小的负数出现在所表示范围的末尾，更大的计数将回卷到0。这是由于不能表示$2^N+1$。
- 因此，在计数范围内，$2^N$后面跟着0，这样**最小的负数**就立即调整到最小的正数。

# 负数表示方法
# --补码表示法

## 补码表示法

- 由于上面的原因，引入了二进制补码的概念，如右图所示。




二进制补码表示法

# 负数表示方法
# --补码表示法

- 在二进制补码编码中，MSB仍是符号位，1表示负数，0表示正数。
- 在二进制补码中，0由一个包含所有0的比特模式所定义。其余的$2^N$-1个数表示非零的正数和负数。
  - 由于$2^N$ -1是奇数， ($2^N$ -1)/2个编码表示负数，[($2^N$ -1)/2]-1个编码表示正数。
  - 换句话说，可以表示的负数比正数要多一个。
  - 最大负数的幅度要比最大正数的幅度个数要多一个。

# 负数表示方法
## --补码表示法

**二进制补码编码的不利的地方是，不容易理解负数。**

- **对于一个N位字长的二进制补码来说，其可以表示的有符号数（包括正数、负数和0）的范围是：**

$$- 2^{N-1} \sim 2^{N-1} - 1$$

# 负数补码的计算
## --负整数补码的计算

- 原码转补码
  - 将该负数所对应的正数按位全部取反。
  - 将取反后的结果加1。

【例】将+17转换-17的二进制补码

- 对应的正整数17的二进制原码为（00010001）$_2$
- 按位取反后得到二进制反码（11101110）$_2$
- 结果加1，得到二进制补码（11101111）$_2$

# 负数补码的计算
## --负整数补码的计算

【例】将-35转换为+35的二进制补码

- 对应的负整数-35的二进制补码为（11011101）$_2$
- 按位取反后得到二进制反码（00100010）$_2$
- 结果加1，得到+35的二进制补码（00100011）$_2$

【例】将-127转换为+127的二进制补码

- 对应的负整数-127的二进制补码为（10000001）$_2$
- 按位取反后得到二进制反码（01111110）$_2$
- 结果加1，得到+127的二进制补码为（01111111）$_2$

# 负数补码的计算
# --负整数补码的计算

## 比较法

- 得到需要转换负数的最小权值，该权值为负数，以$-2^i$表示，使得其满足：

$$-2^i < \text{需要转换的负数};$$

- 取比该权值绝对值$2^i$小的权值，以$2^{i-1}$，$2^{i-2}$，…，$2^0$的幂次方表示；
- 需要转换的负数$+2^i$，得到了正数，以后的权值$2^{i-1}$，$2^{i-2}$，…，$2^0$与这个正数进行比较。

# 负数补码的计算
## --负整数补码的计算

【例】使用比较法得到负整数-97所对应的二进制补码

对于负的十进制整数-97来说，假设使用8位二进制数进行表示，则其所对应的二进制补码为10011111B。

| 转换的数 | -97 | 31 | 31 | 31 | 15 | 7 | 3 | 1 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 权值 | $-2^7$ (-128) | $2^6$ (64) | $2^5$ (32) | $2^4$ (16) | $2^3$ (8) | $2^2$ (4) | $2^1$ (2) | $2^0$ (1) |
| 二进制数 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 余数 | 31 | 31 | 31 | 15 | 7 | 3 | 1 | 0 |

# 负数补码的计算
# --负小数补码的计算

## 比较法

- 得到需要转换负小数的最小权值，该权值为负数，以$-2^0$表示
- 取比该权值绝对值$2^i$小的权值，以$2^{-1}$，$2^{-2}$，…，$2^{-N}$的幂次方表示。
- 需要转换的负数+1，得到了正数，以后的权值$2^{-1}$，$2^{-2}$，…，$2^{-N}$与这个正小数数进行比较。

# 负数补码的计算
## --负小数补码的计算

【例】使用比较法得到负小数-0.03125所对应的二进制补码。

通过比较法，得到十进制负小数-0.03125所对应的二进制小数为1.11111。

| 转换的数 | -0.03125 | 0.96875 | 0.46875 | 0.21875 | 0.09375 | 0.03125 |
|---|---|---|---|---|---|---|
| 权值 | $-2^{0}$ (-1) | $2^{-1}$ (0.5) | $2^{-2}$ (0.25) | $2^{-3}$ (0.125) | $2^{-4}$ (0.0625) | $2^{-5}$ (0.03125) |
| 二进制数 | 1 | 1 | 1 | 1 | 1 | 1 |
| 余数 | 0.96875 | 0.46875 | 0.21875 | 0.09375 | 0.03125 | 0 |