数字电路与逻辑设计

Digital circuit and logic design

第一章　基本知识

主讲教师 | 于俊清

提纲

- 数字信号与系统
- 数制及其转换
- 带符号二进制数的代码表示
- 几种常用的编码

问题

还有更合理的表示方法吗？

常用机器码

模的概念

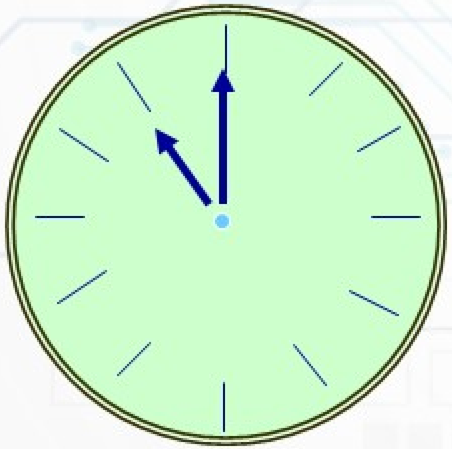
- “模”是指一个计量系统的计数范围
  - 如时钟的计量范围是0～11，模=12
- 计算机也可以看成一个计量系统，它也有一个计量范围，即都存在一个“模”
  - 表示n位的计算机计量范围是0～2^n-1，模= 2^n
- “模”实质上是计量系统产生“溢出”的量，它的值在计量系统上表示不出来，计量系统上只能表示出模的余数
- 任何有模的计量系统，均可化减法为加法运算

补码的特征

标准时间3点整，有一只表的当前时间为11点，如何校准时间？

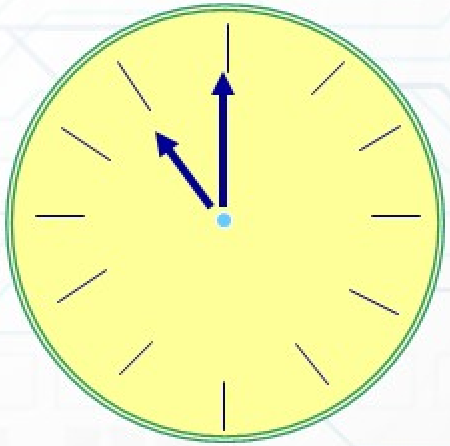
方法一：

顺时针转4个小时



方法二：

逆时针转8个小时



补码的特征

- 结论：两种方法是等效的，这种关系记作：

  -8=4（ mod 12 ）

- 含义：-8与4对模12是互补的，或者说以12为模时-8的补码为4

- 模（或称模数）：一个计量系统的范围，记作mod或M

- 同理，模为12时，-2的补码是10，-5的补码是7

补码表示法

整数补码的定义

$$[X]_{补} = \begin{cases} X & 0 \le X < 2^n \\ 2^{n+1} + X & -2^n < X \le 0 \end{cases}$$

小数补码的定义

$$[X]_{补} = \begin{cases} X & 0 \le X < 1 \\ 2 + X & -1 < X \le 0 \end{cases}$$

补码表示法

- 符号位与原码相同
- 负数的补码的符号位为1，数值位为其反码的末位加1
- 例如：$[-1010]_{补}=[-1010]_{反}+1=10101+1=10110$
- 对于0，在补码中的定义下只有一种表示形式：

$$[+0.00\ldots0]_{补}=[-0.00\ldots0]_{补}=0.00\ldots0$$

补码的意义

计算机中的数据受字长的限制，数据的运算属于有模运算

计算结果直接丢掉进位

可以将减法转为加法运算

计算机中可只设置加法器，从而可简化设计、降低成本

求补码举例

例

- 若$X = 0.1010$，则$[X]_{补} = 0.1010$
- 若$X = -0.1010$，则$[X]_{补} = 2 - 0.1010 = 1.0110$
- 若$X = 1010$，则$[X]_{补} = 01010$
- 若$X = -1010$，则$[X]_{补} = 2^5 - 1010$

$$= 10000 - 1010 = 10110$$

带符号数的原码、反码和补码表示

	原码	反码	补码		原码	反码	补码
+0	0000	0000	0000	-0	1000	1111	0000
+1	0001	0001	0001	-1	1001	1110	1111
+2	0010	0010	0010	-2	1010	1101	1110
+3	0011	0011	0011	-3	1011	1100	1101
+4	0100	0100	0100	-4	1100	1011	1100
+5	0101	0101	0101	-5	1101	1010	1011
+6	0110	0110	0110	-6	1110	1001	1010
+7	0111	0111	0111	-7	1111	1000	1001

求补码？

- X =0.11111111, $[X]_{补}$ = ?

  $[X]_{补}$ =0.11111111

- X = - 0.11111111, $[X]_{补}$=?

  $[X]_{补}$=1.00000000+0.00000001=1.00000001

- X =-0.10101001, $[X]_{补}$=?

  $[X]_{补}$=1.01010110+0.00000001=1.01010111

负数的补码加法

假设字长为8bits

- 十进制运算：$(-1)_{10}+(-1)_{10}=(-2)_{10}$
- 二进制运算

$$(-1)_{10}+(-1)_{10}$$
$$=(10000001)_{原码}+(10000001)_{原码}$$
$$=(11111111)_{补码}+(11111111)_{补码}$$
$$=(\color{red}{1}11111110)_{补码}$$
$$=(11111110)_{补码}=(-2)_{10}$$

符号位进位直接丢掉

补码的减法

- 十进制运算：$(1)_{10}-(2)_{10}=(-1)_{10}$

- 二进制运算

$(1)_{10}-(2)_{10}$

$=(1)_{10}+(-2)_{10}$

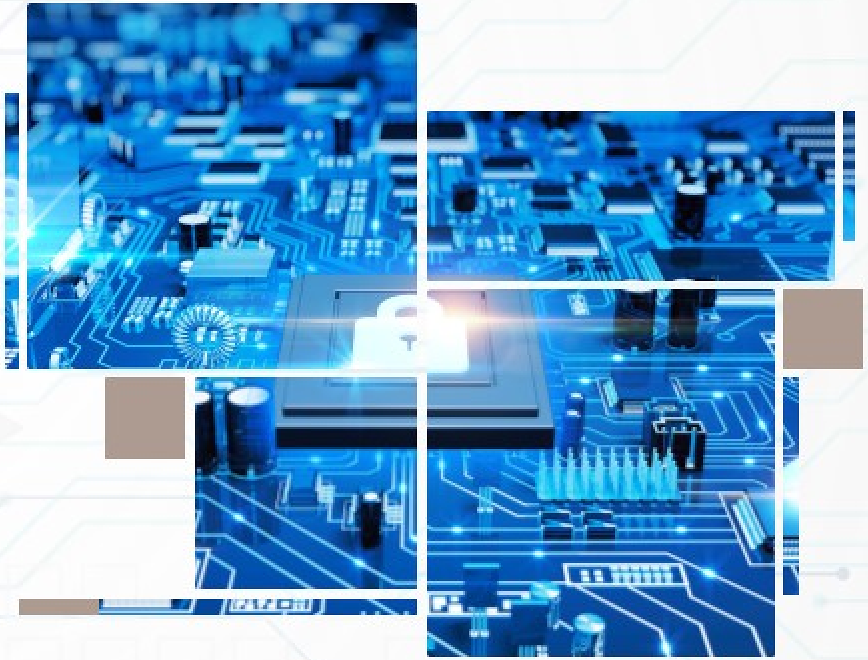
$=(00000001)_{原码}+(10000010)_{原码}$

$=(00000001)_{补码}+(11111110)_{补码}$

$=(11111111)_{补码}$

$=(-1)_{10}$

正确



补码的减法

- 十进制运算：$(1)_{10} - (1)_{10} = (0)_{10}$

- 二进制运算

$$
\begin{aligned}
& (1)_{10} - (1)_{10} \\
&= (1)_{10} + (-1)_{10} \\
&= (00000001)_{\text{原码}} + (10000001)_{\text{原码}} \\
&= (00000001)_{\text{补码}} + (11111111)_{\text{补码}} \\
&= (100000000)_{\text{补码}} \\
&= (00000000)_{\text{补码}} = (0)_{10}
\end{aligned}
$$

符号位进位丢弃

合理

常用机器码

机器数的应用

机器数的应用

- 补码的加减法比较方便，得到了广泛应用
- 目前计算机中广泛采用补码表示
- 少数机器采用原码进行存储和传输，计算时用补码表示

机器码的求法对比

机器码	真值为正数	真值为负数
原　码	符号位为0，等于真值	符号为1，等于真值
反　码	符号位为0，等于真值	符号为1，逐位取反
补　码	符号位为0，等于真值	符号为1，逐位取反，末位加1

机器码的求法对比

真　值	+10001111	-10001111	+0.10011111	-0.10001111
原　码	010011111	110011111	0.10011111	1.10011111
反　码	010011111	101100000	0.10011111	1.01100000
补　码	010011111	101100001	0.10011111	1.01100001

三种表示方法均有符号位和数值位两部分，符号位都是用1表示“负”，用0表示“正”，而数值位各不相同

数字电路与逻辑设计

Digital circuit and logic design

谢谢，祝学习快乐！

主讲教师 | 于俊清