

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

01

■ 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

数据的表示形式



两种表示方法：真值法和机器数（机器码）



真值法：一般书写形式表示的数，通常用“+”、“-”表示正负



机器数：正负符号数码化后的数据



机器数实际上是数据在机器中的表示形式，是由数据的符号位和数值部分一起编码而成



符号位的表示



$$\begin{cases} X_n = 0, & X \geq 0 \\ X_n = 1, & X \leq 0 \end{cases}$$

■ 常用机器码



机器码

原码表示法

又称之为符号数值表示法，是一种比较直观的编码表示法

表示了数据的正或负

原 码

符号位



数码 0 表示正号，数码 1 表示负号



数值部分保留了真值的特征，为真值的绝对值



简言之：符号位+真值绝对值



原码表示法

整数原码的定义

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X < 2^n \\ 2^n - X & -2^n < X \leq 0 \end{cases}$$

小数原码的定义

$$[X]_{\text{原}} = \begin{cases} X & 0 \leq X < 1 \\ 1 - X & -1 < X \leq 0 \end{cases}$$

整数原码举例

例1



$x = +1101$

$[x]_{\text{原}} = 01101$

■ 整数原码举例

例2



$$x = -1101$$

$$\begin{aligned}[x]_{\text{原}} &= 2^4 - (-1101) \\ &= 10000 + 1101 \\ &= 11101\end{aligned}$$

■ 小数原码表示举例

例3



$$x=0.1011$$

$$[x]_{\text{原}}=0.1011$$

■ 小数原码表示举例

例4



$$x = -0.1011$$

$$\begin{aligned}[x]_{\text{原}} &= 1 - x \\ &= 1 - (-0.1011) \\ &= 1 + 0.1011 \\ &= 1.1011\end{aligned}$$

十进制数的原码求法举例

例5



求 $x = 25$ 的原码

$$x = 25$$

$$= 16 + 8 + 1$$

$$= (10000)_2 + (1000)_2 + (1)_2$$

$$= (11001)_2$$

$$[x]_{\text{原}} = 011001$$

带符号数的原码表示

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

原码表示法的特点



零的表示有 “+0” 和 “-0” 之分



$$[+0.00 \dots 0]_{\text{原}} = 0.00 \dots 0$$



$$[-0.00 \dots 0]_{\text{原}} = 1.00 \dots 0$$



正数的原码是其本身，负数的原码的符号位为1，数值位不变

原码的加法

假设字长为8bits



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



二进制运算

$$\begin{aligned} & (1)_{10} + (1)_{10} \\ &= (00000001)_{\text{原码}} + (00000001)_{\text{原码}} \\ &= (00000010)_{\text{原码}} \\ &= (2)_{10} \\ & \text{正确} \end{aligned}$$

原码的加法

假设字长为8bits



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



二进制运算

$$\begin{aligned} & (-1)_{10} + (-1)_{10} \\ &= (10000001)_{\text{原码}} + (10000001)_{\text{原码}} \\ &= (\textcolor{red}{1}00000010)_{\text{原码}} \\ &= ? \end{aligned}$$

数值位：2，符号位如何确定呢？



原码的减法



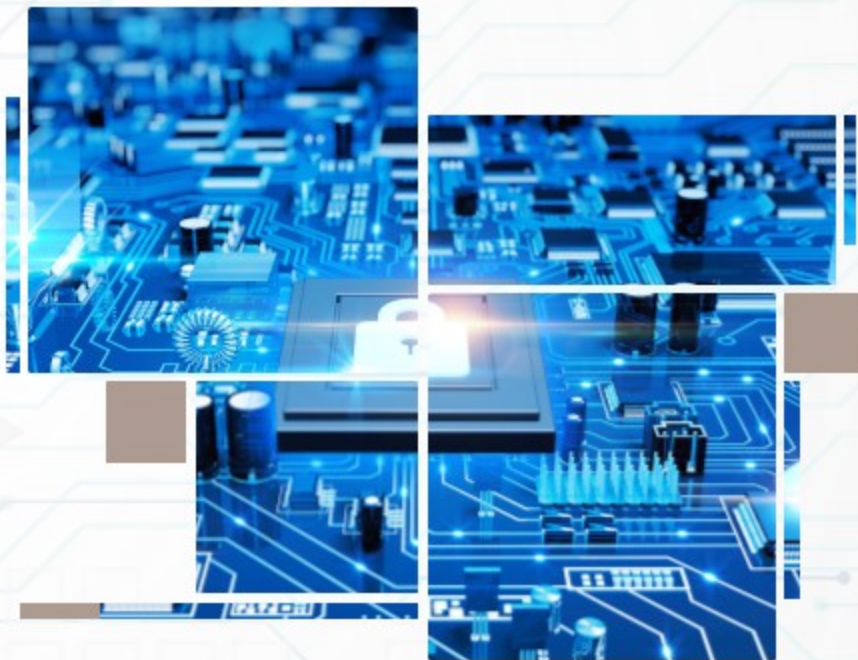
十进制运算： $(1)_{10} - (1)_{10} = (0)_{10}$



二进制运算

$$\begin{aligned} & (1)_{10} - (1)_{10} \\ &= (1)_{10} + (-1)_{10} \\ &= (00000001)_{\text{原码}} + (10000001)_{\text{原码}} \\ &= (10000010)_{\text{原码}} \\ &= (-2)_{10} \end{aligned}$$

正确吗？



原码的减法



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



二进制运算

$$\begin{aligned} & (1)_{10} - (2)_{10} \\ &= (1)_{10} + (-2)_{10} \\ &= (00000001)_{\text{原码}} + [10000010]_{\text{原码}} \\ &= (10000011)_{\text{原码}} \\ &= (-3)_{10} \end{aligned}$$

正确吗？



结论



原码不能直接进行减法运算

正确做法



当对两个数求和时，如果符号相异，则需要先比较两个数的绝对值的大小，然后做减法



绝对值大的符号是结果的符号



绝对值的差值是结果的数值位

缺点



利用它进行加减法运算较为麻烦



问题



如何简化问题呢？

解决办法



减法变加法，符号位直接参与运算



原码不行，有其他编码吗？

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

01