



1.3 二进制的算术运算

1.3.1 无符号二进制的数算术运算

1.3.2 有符号二进制的数算术运算



1.3 二进制的算术运算

1.3.1 无符号数算术运算

1、二进制加法

无符号二进制的加法规则：

$$0+0=0, 0+1=1, 1+1=\textcolor{red}{1}0。$$

例1.3.1 计算两个二进制数1010和0101的和。

解：

$$\begin{array}{r} 1 \quad 0 \quad 1 \quad 0 \\ + \quad 0 \quad 1 \quad 0 \quad 1 \\ \hline 1 \quad 1 \quad 1 \quad 1 \end{array}$$

所以， $1010+0101=1111$



2. 二进制减法

无符号二进制数的减法规则：

$$0-0=0, \quad 1-1=0, \quad 1-0=1, \quad 0-1=1$$

例1.3.2 计算两个二进制数1010和0101的差。

解：

$$\begin{array}{r} 1010 \\ - 0101 \\ \hline 0101 \end{array}$$

所以， $1010-0101=0101$

注： 由于无符号二进制数中无法表示负数，则要求被减数一定大于减数。

3、二进制乘法

无符号二进制数的乘法规则:

$$0 \times 0 = 0, \quad 0 \times 1 = 0, \quad 1 \times 0 = 0 \quad 1 \times 1 = 1$$

例1.3.3 计算两个二进制数1010和0101的积。

解：

			1	0	1	0
		×	0	1	0	1
			1	0	1	0
		0	0	0	0	
	1	0	1	0		
0	0	0	0			
	1	1	0	0	1	0

所以, $1010 \times 0101 = 110010$



4、二进制除法

无符号二进制数的除法规则：

$$0 \div 1 = 0, \quad 1 \div 1 = 1 \quad (\text{除数不能为} 0)$$

例1.3.4 计算两个二进制数1010和111之商。

解：

$$\begin{array}{r} 1.011 \\ 111 \overline{) 1010} \\ \underline{111} \\ 1100 \\ \underline{111} \\ 1010 \\ \underline{111} \\ 11\ldots\text{余数} \end{array}$$

所以， $1010 \div 111 = 1.011$ 余11

1.3.2 带符号二进制的减法运算

有符号的二进制数表示：

二进制数的最高位表示符号位，且用0表示正数，用1表示负数。其余部分用原码的形式表示数值位。

$$(+11)_D = (0\ 1011)_B$$

$$(-11)_D = (1\ 1011)_B$$

补码：为简化运算电路，数字电路中，两数相减的运算是用它们的补码相加来完成的。二进制数的补码是这样定义的：最高位为符号位，正数为0，负数为1；

补码：符号位 + 尾数部分

正数的反码、补码和它的原码相同

负数的补码可通过将它的原码数值位逐位求反，最后在最低位加1得到



例1.3.6 分别计算出A=+6和B=-6的4位二进制数的原码、反码和补码。

$$A_{\text{原}} = 0110$$

$$B_{\text{原}} = 1110$$

$$A_{\text{反}} = 0110$$

$$B_{\text{反}} = 1001$$

$$A_{\text{补}} = 0110$$

$$B_{\text{补}} = 1010$$



二进制补码的减法运算:

减法运算的原理:减去一个正数相当于加上一个负数

$A-B=A+(-B)$, 对 $(-B)$ 求补码, 然后进行加法运算。

例1.3.7 试用4位二进制补码计算 $5-2$ 。

解: 因为 $(5-2)_{\text{补}}=(5)_{\text{补}}+(-2)_{\text{补}}$

$=0101+1110$

$=0011$

所以 $5-2=3$

$$\begin{array}{r} 0101 \\ + 1110 \\ \hline [1]0011 \end{array}$$

自动丢弃



运算举例

例：

$$(1001)_2 - (0101)_2 = ?$$



溢出

例1.3.8 试用4位二进制补码计算 $5+7$ 。

解：因为 $(5+7)_{\text{补}} = (5)_{\text{补}} + (7)_{\text{补}}$

$$= 0101 + 0111$$

$$= 1100$$

	0	1	0	1
+	0	1	1	1
<hr/>				
	[1]	1	0	0

解决溢出的办法：进行位扩展。



溢出的判别

两个符号相反的数相加不会产生溢出，但两个符号相同的数相加有可能产生溢出。

如何判断是否产生溢出？💡

$$\begin{array}{r}
 + 4 \\
 +) + 3 \\
 \hline
 + 7
 \end{array}
 \quad
 \begin{array}{r}
 0 \ 1 \ 0 \ 0 \\
 + 0 \ 0 \ 1 \ 1 \\
 \hline
 [0] \ 0 \ 1 \ 1 \ 1
 \end{array}$$

$$\begin{array}{r}
 + 2 \\
 +) + 6 \\
 \hline
 + 8
 \end{array}
 \quad
 \begin{array}{r}
 0 \ 0 \ 1 \ 0 \\
 + 0 \ 1 \ 1 \ 0 \\
 \hline
 [0] \ 1 \ 0 \ 0 \ 0
 \end{array}$$

-8~7

$$\begin{array}{r}
 - 5 \\
 +) - 3 \\
 \hline
 - 8
 \end{array}
 \quad
 \begin{array}{r}
 1 \ 0 \ 1 \ 1 \\
 + 1 \ 1 \ 0 \ 1 \\
 \hline
 [1] \ 1 \ 0 \ 0 \ 0
 \end{array}$$

$$\begin{array}{r}
 - 3 \\
 +) - 6 \\
 \hline
 - 9
 \end{array}
 \quad
 \begin{array}{r}
 1 \ 1 \ 0 \ 1 \\
 + 1 \ 0 \ 1 \ 0 \\
 \hline
 [1] \ 0 \ 1 \ 1 \ 1
 \end{array}$$

当方框中的进位位与和数的符号位（即 b_3 位）相反时，则运算结果是错误的，产生溢出。



1.4 二进制代码

1.4.1 二-十进制码

1.4.2 格雷码

1.4.3 ASCII码

1.4 二进制代码

码制:编制代码所要遵循的规则

二进制代码的位数(n),与需要编码的事件（或信息）的个数(N)之间应满足以下关系:

$$2^{n-1} \leq N \leq 2^n$$

1. 二—十进制码 (数值编码)

(BCD码----- Binary Coded Decimal)

用4位二进制数来表示一位十进制数中的0~9十个数码。

从4 位二进制数16种代码中,选择10种来表示0~9个数码的方案有很多种。每种方案产生一种BCD码。



1.4.1 二-十进制码

(1) 几种常用的BCD代码

BCD码十进制数码	8421码	2421 码	5421 码	余3码	余3循环码
0	0000	0000	0000	0011 3	0010
1	0001	0001	0001	0100 4	0110
2	0010	0010	0010	0101 5	0111
3	0011	0011	0011	0110 6	0101
4	0100	0100	0100	0111 7	0100
5	0101	1011	1000	1000 8	1100
6	0110	1100	1001	1001	1101
7	0111	1101	1010	1010	1111
8	1000	1110	1011	1011	1110
9	1001	1111	1100	1100	1010



求BCD代码表示的十进制数

对于有权BCD码，可以根据位权展开求得所代表的十进制数。例如：

$$[0111]_{8421\text{BCD}} = 0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1 = (7)_D$$

$$[1101]_{2421\text{BCD}} = 1 \times 2 + 1 \times 4 + 0 \times 2 + 1 \times 1 = (7)_D$$

$$[1010]_{5421\text{BCD}} = 1 \times 5 + 0 \times 4 + 1 \times 2 + 0 \times 1 = (7)_D$$



用BCD代码表示十进制数

对于一个多位的十进制数，需要有与十进制位数相同的几组BCD代码来表示。例如：

$$(463.5)_{10} = \left[\begin{array}{c} \underline{0100} \quad \underline{0110} \quad \underline{0011} \cdot \underline{0101} \\ 4 \quad 6 \quad 3 \quad 5 \end{array} \right]_{8421\text{BCD}}$$

不能省略！

$$(863.2)_{10} = \left[\begin{array}{c} \underline{1110} \quad \underline{1100} \quad \underline{0011} \cdot \underline{0010} \\ 8 \quad 6 \quad 3 \quad 2 \end{array} \right]_{2421\text{BCD}}$$

不能省略！

1.4.2 格雷码

格雷码是一种无权码。

编码特点是：任何两个相邻代码之间仅有一位不同。

该特点常用于模拟量的转换。当模拟量发生微小变化，格雷码仅仅改变一位，这与其它码同时改变2位或更多的情况相比，更加可靠,且容易检错。

二进制码 $b_3b_2b_1b_0$	格雷码 $G_3G_2G_1G_0$
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

二进制码到格雷码的转换

- (1) 格雷码的最高位（最左边）与二进制码的最高位相同。
- (2) 从左到右，逐一将二进制码相邻的两位相加（舍去进位），作为格雷码的下一位。

1 — + —→ 0 — + —→ 1 — + —→ 1				二进制码
↓	↓	↓	↓	
1	1	1	0	格雷码

格雷码到二进制码的转换

- (1) 二进制码的最高位（最左边）与格雷码的最高位相同。
- (2) 将产生的每一位二进制码，与下一位相邻的格雷码相加（舍去进位），作为二进制码的下一位。

1	1	0	1	格雷码
↓	↓ + ↗	↓ + ↗	↓ + ↗	
1	0	0	1	二进制码



1.4.3 ASCII 码(字符编码)

ASCII码即美国标准信息交换码。

(American Standard Code for Information Interchange, ASCII)

它共有128个代码，可以表示大、小写英文字母、十进制数、标点符号、运算符号、控制符号等，普遍用于计算机的键盘指令输入和数据等。

P28 表1.4.3A: ASCII码; 表1.4.3B: ASCII码中各字符的含义