



南通大學
NANTONG UNIVERSITY

南通大学电气工程学院

数字逻辑电路

主讲老师：王亚芳



二进制的算术运算

本节主要内容

- ▶ 无符号二进制的算术运算
- ▶ 有符号二进制数的表示（原码、反码和补码）
- ▶ 补码的加减运算
- ▶ 溢出（判别及解决方法）

学习完本节， 你能

- ▶ 了解无符号二进制数的加、减、乘、除四种运算规则
- ▶ 掌握有符号二进制数原码、反码和补码的表示
- ▶ 掌握补码的加减运算规则
- ▶ 掌握溢出的判别及解决方法



无符号二进制的算术运算

算术运算

- 当0和1表示数量大小时，两个二进制数可以进行算术运算；
- 二进制数的加、减、乘、除四种运算的运算规则与十进制类似，区别在于进位或借位规则不同。

逢二进一、借一当二



无符号二进制的算术运算

1、二进制加法

➤ 无符号二进制的加法规则：

$$0+0=0, 0+1=1, 1+1=\boxed{1}0$$

进位位

例

计算两个二进制数1010和0101的和。

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

所以

$$1010+0101=1111$$



无符号二进制的算术运算

2、二进制减法

➤ 无符号二进制的减法规则：

$$0-0=0, 1-1=0, 1-0=1, 0-1=$$

借位位

1 1

例

计算两个二进制数1010和0101的差。

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

所以

$$1010-0101=0101$$



无符号二进制的算术运算

3、二进制乘法

➤ 无符号二进制的乘法规则：

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



无符号二进制的算术运算

例

计算两个二进制数1010和0101的积。

$$\begin{array}{r} 1010 \\ \times 0101 \\ \hline 1010 \\ 0000 \\ 0000 \\ 0000 \\ \hline 110010 \end{array}$$

所以 $1010 \times 0101 = 110010$



无符号二进制的算术运算

4、二进制除法

➤ 无符号二进制的除法规则：

$$0 \div 1 = 0, 1 \div 1 = 1$$

➤ 注：除数不能为0，否则无意义。



无符号二进制的算术运算

例

计算两个二进制数1010和111之商。

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

所以

$$1010 \div 111 = 1.011 \quad \text{余} 11$$



有符号二进制数的表示



数字系统只能识别和处理用0、1表示的二进制形式的数据，
在计算机内部如何表示正、负数？

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。

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

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





有符号二进制数的表示

◆ 原码的表示：

- 正数的符号位用0表示，负数的符号位用1表示；
- 数值用其绝对值的二进制数形式表示。

◆ 反码的表示：

- 正数的反码和原码相同；
- 负数的反码：符号位不变，数值位取反。

◆ 补码的表示：

- 正数的补码和原码相同；
- 负数的补码：符号位不变，数值位取反后加1（反码加1）。



有符号二进制数的表示

例

分别计算出 $A=+5$ 和 $B=-5$ 的4位二进制的原码、反码和补码。

解：A和B的绝对值均为5。除最高位为符号位外，还有3位为数值位。

$$A_{\text{原}} = 0\ 101$$

$$B_{\text{原}} = 1\ 101$$

$$A_{\text{反}} = 0\ 101$$

$$B_{\text{反}} = 1\ 010$$

$$A_{\text{补}} = 0\ 101$$

$$B_{\text{补}} = 1\ 011$$



有符号二进制的表示

4位二进制原码、反码、补码对照表

十进制数	二进制数		
	原码	反码	补码
-8	—	—	1 0 0 0
-7	1 1 1 1	1 0 0 0	1 0 0 1
-6	1 1 1 0	1 0 0 1	1 0 1 0
-5	1 1 0 1	1 0 1 0	1 0 1 1
-4	1 1 0 0	1 0 1 1	1 1 0 0
-3	1 0 1 1	1 1 0 0	1 1 0 1
-2	1 0 1 0	1 1 0 1	1 1 1 0
-1	1 0 0 1	1 1 1 0	1 1 1 1
-0	1 0 0 0	1 1 1 1	0 0 0 0

十进制数	二进制数		
	原码	反码	补码
+0	0 0 0 0	0 0 0 0	0 0 0 0
+1	0 0 0 1	0 0 0 1	0 0 0 1
+2	0 0 1 0	0 0 1 0	0 0 1 0
+3	0 0 1 1	0 0 1 1	0 0 1 1
+4	0 1 0 0	0 1 0 0	0 1 0 0
+5	0 1 0 1	0 1 0 1	0 1 0 1
+6	0 1 1 0	0 1 1 0	0 1 1 0
+7	0 1 1 1	0 1 1 1	0 1 1 1

原码

$$-(2^{n-1}-1) \sim +(2^{n-1}-1)$$

反码

$$-(2^{n-1}-1) \sim +(2^{n-1}-1)$$

补码

$$-2^{n-1} \sim +(2^{n-1}-1)$$



有符号二进制数的算术运算

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

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 ←



补码的加减运算

例

试用4位二进制补码计算5+7。

解：因为 $(5+7)_{\text{补}} = (5)_{\text{补}} + (7)_{\text{补}}$
 $= 0101 + 0111$
 $= 1100$

计算错误！

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

4位有符号数所能表示的补码数的最大值为+7

溢出



溢出

如何判断是否产生溢出？

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



$$\begin{array}{r} - 5 \\ +) - 3 \\ \hline - 8 \end{array}$$

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



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



$$\begin{array}{r} - 3 \\ +) - 6 \\ \hline - 9 \end{array}$$

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



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。



溢出

解决溢出的办法:进行位扩展

$$\begin{array}{r} + 5 \\ +) + 7 \\ \hline + 12 \end{array} \quad \begin{array}{r} 0 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \ 1 \\ \hline 1 \ 1 \ 0 \ 0 \end{array}$$



$$\begin{array}{r} + 5 \\ +) + 7 \\ \hline + 12 \end{array} \quad \begin{array}{r} 0 \ 0 \ 1 \ 0 \ 1 \\ + 0 \ 0 \ 1 \ 1 \ 1 \\ \hline 0 \ 1 \ 1 \ 0 \ 0 \end{array}$$



小结：

- 与十进制数类似，二进制数也有加、减、乘、除四种运算；
- 有符号的二进制数的原码、反码、补码的表示方法；
- 常用补码来进行有符号数的加减运算；
- “溢出”仅发生在两个同符号的数相加的情况。解决方法：位的扩展。

