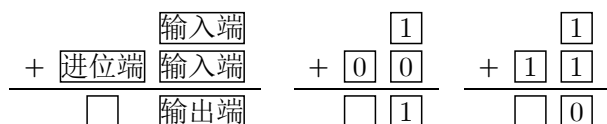


1、**异或门**:①带非门的或门;②与门控制非门。

2、**半加器**:两个一位二进制数加法运算。

异或门的输出为输出端,两个输入端组成与门输出为进位端。



3、**全加器**:两个二进制数相加运算,两个输入端,一个进位输入端,一个本位输出端,一个进位输出端。

第一个半加器的输出端连接到第二个半加器的输入端,两个半加器的进位端组成或门输出为进位输出端,第二个半加器的输出端为全加器的本位输出端,第二个半加器的另一个输入端为进位输入端。



4、**加法器**:将前一个半加器的进位输出端与后一个半加器的进位输入端相连接。

5、**二进制的减法**:

负数的二进制:将最高位作为符号位,正数为0,负数为1,例如:10010表示-2。

$$\begin{array}{r} 2 \\ + - 2 \\ \hline 0 \end{array} \Rightarrow \begin{array}{r} 00010 \\ + 10010 \\ \hline 10100 \end{array} \quad \text{结果为}-4, \text{显然不对。}$$

求一下2加上什么数等于0。

$$\begin{array}{r} 2 \\ + ? \\ \hline 0 \end{array} \Rightarrow \begin{array}{r} 0010 \\ + ???? \\ \hline 0000 \end{array} \Rightarrow \begin{array}{r} 0010 \\ + 1110 \\ \hline 0000 \end{array} \quad 1110 + 0010 = 10000$$

同理求-3。

$$\begin{array}{r} 3 \\ + ? \\ \hline 0 \end{array} \Rightarrow \begin{array}{r} 0011 \\ + ???? \\ \hline 0000 \end{array} \Rightarrow \begin{array}{r} 0011 \\ + 1101 \\ \hline 0000 \end{array} \quad 1101 + 0011 = 10000$$

上面的1110和1101分别为2(0010)和3(0011)的补码,他们的和为16(10000),称为一组互补数,16为他们的模。

在计算机中,数据以二进制形式存储,且位数固定(如8位、16位、32位),这个固定位数的二进制数的取值范围被称为模。对于n位二进制数,模为 2^n ,例如8位二进制数的模为 $2^8 = 256$ 。补码的本质是利用“模运算”的特性:一个数减去另一个数,等价于加上这个数的补数(模减去该数的绝对值)。



将时钟从10点拨到8点,可以向前拨2小时,也可以向后拨10小时。对于时钟,10-2等价于10+10,12就是模。

$$7-2=5$$

$$7+8=15 \text{ (个位相同)}$$

$$-2 \Leftrightarrow -2+10=8 \quad 2 \text{ 和 } 8 \text{ 对于模 } 10 \text{ 互补。}$$

$$37-25=12$$

$$37+75=112 \text{ (个十位相同)}$$

$$-25 \Leftrightarrow -25+100=75 \quad 25 \text{ 和 } 75 \text{ 关于 } 100 \text{ 互补。}$$

$$1010-0111=0011$$

$$1010+1001=10011 \text{ (后四位相同)}$$

$$-0111 \Leftrightarrow -0111+10000=1001 \quad 0111 \text{ 和 } 1001 \text{ 关于 } 10000 \text{ 互补。}$$

对于四位二进制数，他们与他们的互补数的和为 10000，因此，求 X 的补码，就是 $10000-X$ ，但是我们无法利用四位二进制数加法器去计算。

由于 $10000-X=1111+1-X$ ，而 $1111-X$ 就是 X 的反码，可以利用非门取反，然后+1 得到补码。

对于任意二进制数，原码+补码=模，补码=反码+1。将补码-1 取反，就能求得原码。

下面是四位二进制数-16 到 15 的补码表：

0	0	0000	-1	1	1111
1	0	0001	-2	1	1110
2	0	0010	-3	1	1101
3	0	0011	-4	1	1100
4	0	0100	-5	1	1011
5	0	0101	-6	1	1010
6	0	0110	-7	1	1001
7	0	0111	-8	1	1000
8	0	1000	-9	1	0111
9	0	1001	-10	1	0110
10	0	1010	-11	1	0101
11	0	1011	-12	1	0100
12	0	1100	-13	1	0011
13	0	1101	-14	1	0010
14	0	1110	-15	1	0001
15	0	1111	-16	1	0000

n 位二进制补码的取值范围为 $[-2^{n-1}, 2^{n-1}-1]$ ，例如：8 位二进制补码的取值范围是 $[-128, 127]$ 。

注意，上表中是将符号位和数据位分开的，有-16 到 15 共 32 个数。而上面的取值范围公式包括了符号位，所以上表应该是五位二进制补码，即 $[-16, 15]$ 。包含符号位的四位二进制补码如下 $[-8, 7]$ ：

0	0000	-1	1111
1	0001	-2	1110
2	0010	-3	1101
3	0011	-4	1100
4	0100	-5	1011
5	0101	-6	1010
6	0110	-7	1001
7	0111	-8	1000

补码和原码：

对于正数，补码和原码相同；对于负数，例如：1010 原码表示 10，补码表示-6；1011 原码表示 11，补码表示-5。

溢出现象（127+1=-128；-128-1=127）：

正溢出：对于 8 位二进制补码表示有符号整数，最大的正数是+127，如果对+127 进行加 1 的操作，会导致正溢出，因为 01111111+1=10000000，最高位为 1，表示-128。

负溢出：最小负数是-128，如果对-128 进行减 1 操作，会导致负溢出，因为 10000000-1=01111111，最高位为 0，表示+127。

符号位的运算：

如果将符号位单独拿出来，该怎么参与运算。

1001-0010（9-2）

		1	0	0	1
+	1	1	1	1	0
		0	1	1	1

0111-1101（7-13）

		0	1	1	1
+	0	0	0	1	1
		1	0	1	0

对于任意四位二进制数 X,Y(均为原码,且为正数),求 X-Y:

Y 取反+1 为 10000-Y

若 X>Y,X+10000-Y>10000,发生进位,结果为正;

若 X<Y,X+10000-Y<10000,不发生进位,结果为负。

通过最高位全加器的进位输出端判断正负,输出 1 为正,输出 0 为负,通过一个非门连接到-1 计算和取反的开关处,以及符号显示位。

对于任意四位二进制数 X,Y(均为原码,且为正数),进行以下运算:

	X+Y	-X-Y	X-Y (X>Y)	X-Y (X<Y)	-X+Y (X>Y)	-X+Y (X<Y)
X 符号	0	1	0	0	1	1
Y 符号	0	1	1	1	0	0
最高位全加器进位输出	0	1	1	0	0	1
结果正负	0	1	0	1	1	0

四位二进制加法器输入端的取值范围:X,Y 异号,因为 X、Y<10000,所以 X-Y 或 Y-X<10000。输入任意四位二进制数都能正确输出结果。X,Y 同号,±(X+Y)时,由于最大只能输出四位二进制数,因此要求输入 X+Y<10000。

上表中,X+Y 时,由于 X+Y<10000,因此不会进位,最高位全加器进位输出 0;-X-Y 时,用加法器计算前要先取反+1,即 10000-X+10000-Y=100000-(X+Y),由于 X+Y<10000,因此 100000-(X+Y)>100000-10000=10000,会发生进位,最高位全加器进位输出 1。

从上表中可以看出,X、Y 符号和最高位全加器进位输出值接入到全加器中,本位输出值刚好满足结果的符号值。

符号位完全可以参与到加法器的计算中,需要注意,如果这样做,左侧最高位为符号位,需要将其直接连接到加法器的最高位,不需要接入取反器和+1 半加器组,同理,拆掉左侧最高位的-1 加法器和取反器,将加法器最高位本位输出端连接到-1 和取反器的控制线上,以及最左侧的符号位。

6.乘法器

11×13=143

1011×1101=10001111

$\times$	1 0 1 1		$1011 \times 1 = 1011$
	1 1 0 1		
	1 0 1 1		$1011 \times 0 = 0000$
	0 0 0 0		
	1 0 1 1		$1011 \times 1 = 1011$
	1 0 1 1		
	1 0 0 0 1 1 1 1		$1011 \times 1 = 1011$

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1 \text{ (符合与门的特征。)}$$

将乘数的每一位与被乘数作与门运算,并将结果移位后相加。

7、除法器

$$154 \div 10 = 15 \cdots 4$$

$$10011010 \div 1010 = 1111 \cdots 0100$$

$$\begin{array}{r}
 0 \\
 1010 \overline{) 10011010} \\
 \underline{1010} \\
 0001
 \end{array}$$

注意:试减结果为负,商取0,除数向低位移动1位。

$$\begin{array}{r}
 01111 \\
 1010 \overline{) 10011010} \\
 \underline{1010} \\
 10010 \\
 \underline{1010} \\
 10001 \\
 \underline{1010} \\
 01110 \\
 \underline{1010} \\
 0100
 \end{array}$$

试减结果为负,商取0,除数向低位移动1位。试减结果为正,商取1,除数向低位移动1位继续下一位减法运算,当余数小于除数时,停止运算。

注意,对于八位除以四位除法器:

- 1、对于每个加法器下方输入端空出来的位数,将其设置为1;每个加法器最右侧进位输入端设置为1。
- 2、当除数为二、三、四位数时,需要锁住第一、第一二、第一二三个试减选择模块的控制线,以确保除数位移到正确位置后才开始运算。
- 3、前四个加法器分别为除数为一、二、三、四位数时的首次试减运算,若试减结果为负,除数向低位移动1位,参与试减运算的被除数将增加1位,因此需要从第五个加法器起,每个加法器最左侧添加1位全加器,即第五至八个加法器为五位加法器。
- 4、为保证除数不能为零,使用一个或非门连接除数输入端,输出商和余数的阻断信号。

设两个四位二进制数 X, Y , 则 $X, Y \leq 15$ 。 $X < Y$ 时:

$$2X + 1 - Y = X + 1 + (X - Y)$$

$$X - Y < 0, X + 1 \leq 16$$

$$2X + 1 - Y \leq 15 \text{ (四位二进制数)}$$

首次试减运算:

一位除数：

$$\begin{array}{r} 1 \\ 1 \overline{) 10011010} \\ \underline{1} \end{array}$$

二位除数：

$$\begin{array}{r} 1 \\ 10 \overline{) 10011010} \\ \underline{10} \end{array}$$

三位除数：

$$\begin{array}{r} 0 \\ 101 \overline{) 10011010} \\ \underline{101} \end{array}$$

四位除数：

$$\begin{array}{r} 0 \\ 1010 \overline{) 10011010} \\ \underline{1010} \end{array}$$

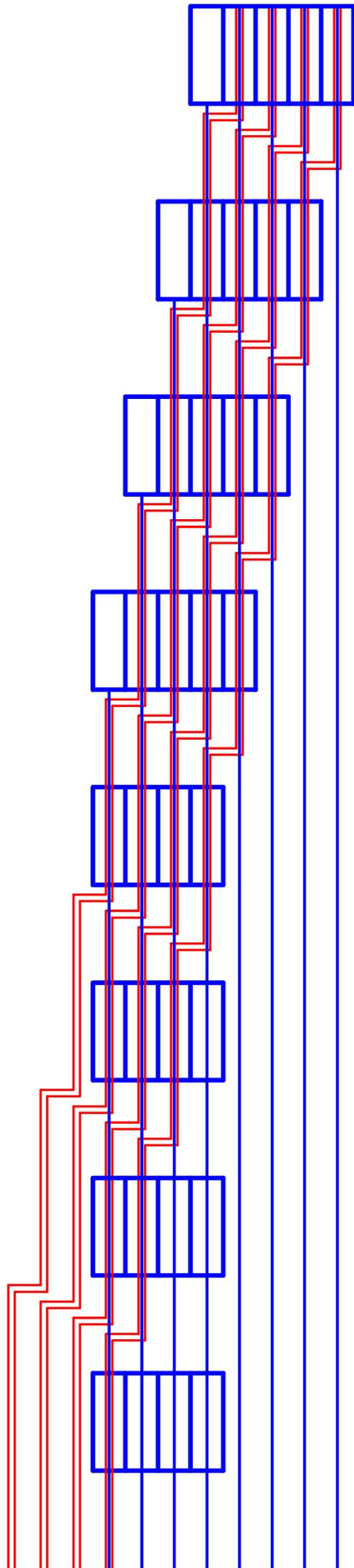
剩余试减运算：

$$\begin{array}{r} 01111 \\ 1010 \overline{) 10011010} \\ \underline{1010} \\ 10010 \\ \underline{1010} \\ 10001 \\ \underline{1010} \\ 01110 \\ \underline{1010} \\ 0100 \end{array}$$

除数右移一次后仍然不够减的情况：

$$\begin{array}{r} 10001 \\ 1001 \overline{) 10011010} \\ \underline{1001} \\ 00001 \\ \underline{1001} \\ 00010 \\ \underline{1001} \\ 00101 \\ \underline{1001} \\ 01010 \\ \underline{1001} \\ 0001 \end{array}$$

除法器平面图：



除法余数和小数的运算：

二进制的位权

a. 整数部分(小数点左侧,从右往左)

位数从0开始:

第0位: $2^0=1$

第1位: $2^1=2$

第2位: $2^2=4$

第3位: $2^3=8$

第4位: $2^4=16$

规律:往左一位,位权 $\times 2$

b. 小数部分(小数点右侧,从左往右)

位数为负:

第-1位: $2^{-1}=0.5$

第-2位: $2^{-2}=0.25$

第-3位: $2^{-3}=0.125$

第-4位: $2^{-4}=0.0625$

规律:往右一位,位权 $\div 2$

示例: $1010.0101_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4}$
 $= 8 + 0 + 2 + 0 + 0 + 0.25 + 0 + 0.0625$
 $= 10.3125_{10}$

计算小数部分:

$$\begin{array}{r}
 01111.011001100110 \\
 1010 \overline{) 10011010.000000000000} \\
 \underline{1010} \\
 10010 \\
 \underline{1010} \\
 10001 \\
 \underline{1010} \\
 01110 \\
 \underline{1010} \\
 010000 \\
 \underline{1010} \\
 01100 \\
 \underline{1010} \\
 0010000 \\
 \underline{1010} \\
 01100 \\
 \underline{1010} \\
 0010000 \\
 \underline{1010} \\
 01100 \\
 \underline{1010} \\
 00100
 \end{array}$$

- ① 除数右移1位,余数右侧添加一位0($\times 2$);
- ② 余数和除数比较:
 - $\geq$ 除数:商取1,减去除数得新余数;
 - $<$ 除数:商取0,余数不变;
- ③ 重复上面步骤,直到余数为0或者需要的精度。

误差分析：

$$1111.01100110_2$$

$$=1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 0 \times 2^{-5} + 1 \times 2^{-6} + 1 \times 2^{-7} + 0 \times 2^{-8}$$

$$=8+4+2+1+0+0.25+0.125+0+0+0.015625+0.0078125+0$$

$$=15.3984375_{10}$$

$$154 \div 10 = 15.4 \quad \text{存在误差}$$

十进制除法计算小数：余数 $\times 10$

二进制除法计算小数：余数 $\times 2$

(1)十进制

十进制基数是 10

质因数： $10=2 \times 5$

只要除数的质因数**只有 2 或 5**，余数不断 $\times 10$ 之后，总能除尽，余数最后为 0

→ 有限小数，余数精准无误差

(2)二进制

二进制基数是 2

质因数只有 2

只要除数的质因数**不是只有 2**，无论余数怎么 $\times 2$ ，不断做减法，最终的余数都不可能为 0

只会无限循环下去 → 无限循环二进制小数

$$\frac{1}{2} = 0.5_{10} = 0.1 \rightarrow \text{有限无误差}$$

$$\frac{1}{4} = 0.25_{10} = 0.01 \rightarrow \text{有限无误差}$$

$$\frac{1}{5} = 0.2_{10} \rightarrow \text{二进制无限循环}$$

$$\frac{1}{10} = 0.1_{10} \rightarrow \text{二进制无限循环}$$

$$\frac{1}{3} = 0.3333..._{10} \rightarrow \text{二进制无限循环}$$

关键点：误差不是**算法错误**，而是**被迫截断**

无论是十进制还是二进制：

能除尽 → 有限位 → 余数归 0 → 无误差

不能除尽 → 无限循环 → 必须截断位数 → 产生误差