第三章 运算方法与运算器

- 运算器用于数值运算及加工处理数据
- 它是由CPU中的算术逻辑单元(ALU)、通用寄存器(GR)等部件构成
- 运算器的结构取决于指令系统、数据的表示方法、运算方法及所选用的硬件。

3.1 定点数运算

3.1.1 加减运算

1. 加减运算方法

- 补码加法

  补码加法的运算法则为：

  $$[X+Y]_{补}=[X]_{补}+[Y]_{补}$$

- 补码减法

  补码减法的运算法则为：

  $$[X-Y]_{补}=[X]_{补}+[-Y]_{补}=[X]_{补}+[[Y]_{补}]_{求补}$$

定点数加减运算

例3.1 若两个定点整数63和35，利用补码加法求 $63 + 35 = ?$

解：根据题意，用8位二进制补码表示63和35:

$$[63]_{补} = 00111111$$
$$[35]_{补} = 00100011$$

则 $[63 + 35]_{补} = 01100010$

定点数加减运算

例3.2 若两个定点整数-63和-35，利用补码加法求-63 + (-35) = ?

解：根据题意，用8位二进制补码表示-63和-35:

$$[-63]_{补} = 11000001$$

$$[-35]_{补} = 11011101$$

则 $[-63 + (-35)]_{补} = 10011110$

定点数加减运算

例3.3 若两个定点整数63和35，利用补码减法求63 − 35 = ?

解：根据题意，用8位二进制补码表示63和35:

$[63]_{补} = 00111111$

$[35]_{补} = 00100011$

而$[63 - 35]_{补} = [63]_{补} + [-35]_{补}$；

同时，$[-35]_{补} = [[35]_{补}]_{求补} = 11011101$，从而求出：

$$\begin{array}{r} 00111111 \\ +\ 11011101 \\ \hline 100011100 \end{array}$$

得到$[63 - 35]_{补} = 00011100$。

- **补码加减运算规则**

① 参加运算的操作数用补码表示；

② 符号位参加运算；

③ 若进行相加，则两个数的补码直接相加；

若进行相减运算，则对减数求补（连同符号位一起变反加1）后与被减数相加；

④ 运算结果用补码表示。

2. 溢出判断

- 溢出

例3.4 若两个定点整数63和85，利用补码加法求63 + 85 = ?

解：根据题意，若用8位二进制补码表示63和85:

$[63]_{补} = 00111111$

$[85]_{补} = 01010101$

$$\begin{array}{r} 00111111 \\ +01010101 \\ \hline 10010100 \end{array}$$

- 两个正数（63和85）相加的结果变成一个负数（符号位为1）。出现这种错误结果是由于在相加的过程中产生了溢出。
- 原因就在于运算的结果超出了所规定的数值范围。

例3.6 设负整数X= -1111000，Y= -10010，若用8位补码表示，则$[X]_{补}$=10001000，$[Y]_{补}$=11101110，求$[X+Y]_{补}$。

解：计算$[X]_{补}+[Y]_{补}$

$$
\begin{array}{r}
1\ 0001000 \\
+\quad 1\ 1101110 \\
\hline
0\ 1110110
\end{array}
$$

两个负数相加，结果为一个正数，显然也是错误的。

- 只有当两个同符号的数相加（或者是相异符号数相减）时，运算结果才有可能溢出。而在异符号的数相加（或者是同符号数相减）时，永远不会产生溢出。

- 只要运算结果超出所能表示的数据范围，就会发生溢出。发生溢出时，运算结果肯定是错误的。只要发现运算结果产生溢出，就必须采取措施防止溢出发生。最简单有效的方法就是增加补码的二进制编码长度。

● 溢出的判定

（1） 双符号位判决法

- ✓ 若补码采用两位表示符号，即00表示正号、11表示负号，一旦发生溢出，则两个符号位就一定不一致，利用判别两个符号位是否一致便可以判定是否发生了溢出。

- ✓ 若运算结果两符号分别用S_2S_1表示，则判别溢出的逻辑表示式为：

$$VF = S_2 \oplus S_1$$

例 设两正整数X=+1000001，Y= +1000011，
若用双符号位的8位补码表示，
则 $[X]_{补}$= 00 1000001，$[Y]_{补}$= 00 1000011，
求 $[X+Y]_{补}$。

解：计算$[X]_{补}+[Y]_{补}$

$$
\begin{array}{r}
00\ 1000001 \\
+\ 00\ 1000011 \\
\hline
01\ 0000100
\end{array}
$$

式中，由于结果的S2和S1 不一致，

VF＝S2 ⊕ S1＝1，溢出发生。

例 $x = +0.1100$, $y = +0.1000$, 求 $x + y$。

解:

$[x]_{补} = 00.1100$, $[y]_{补} = 00.1000$

$$
\begin{array}{lr}
 & [x]_{补} \quad 00.1100 \\
+ & [y]_{补} \quad 00.1000 \\
\hline
 & 01.0100
\end{array}
$$

两个符号位出现“01”，表示已溢出，即结果大于 +1。

例 $x = -0.1100$, $y = -0.1000$, 求 $x + y$。

解:

$$[x]_{补} = 11.0100, \quad [y]_{补} = 11.1000$$

$$\begin{array}{lr} [x]_{补} & 11.0100 \\ +\ [y]_{补} & 11.1000 \\ \hline & \color{red}{10}.1100 \end{array}$$

两个符号位出现“10”，表示已溢出，即结果小于−1。

- **溢出的判定**

（2）进位判决法

若C_{n-1}为最高数值位向最高位（符号位）的进位，C_n表示符号位的进位，则判别溢出的逻辑表示式为：

$$VF = C_{n-1} \oplus C_n$$

（3）根据运算结果的符号位和进位标志判别

该方法适用于两同号数求和或异号数求差时判别溢出。溢出的逻辑表达式为：

$$VF = SF \oplus CF$$

例 设两正整数X=+1000001，Y= +1000011，采用8位补码表示，则 $[X]_{补}$=0 1000001，$[Y]_{补}$=0 1000011，求$[X+Y]_{补}$。

解：计算$[X]_{补}+[Y]_{补}$

$$
\begin{array}{r}
0\ 1000001 \\
+\ 0\ 1000011 \\
\hline
1\ 0000100
\end{array}
$$

式中，由于$C_{n-1}=1$，$C_n=0$，

$VF=C_{n-1}\oplus C_n=1$，溢出发生。

- 溢出的判定

（4）根据运算前后的符号位进行判别

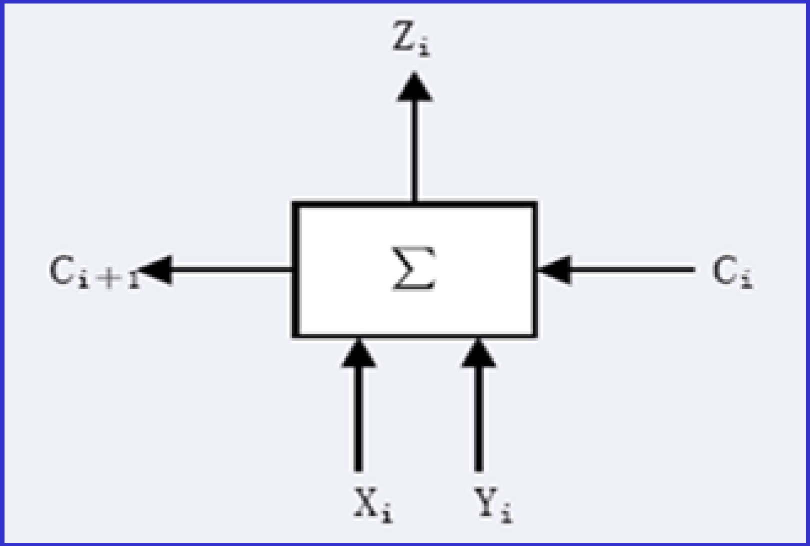
若用Xs、Ys、Zs分别表示两个操作数及运算结果的符号位，当两同号数求和或异号数求差时，就有可能发生溢出。溢出是否发生可根据运算前后的符号位进行判别，其逻辑表达式为：

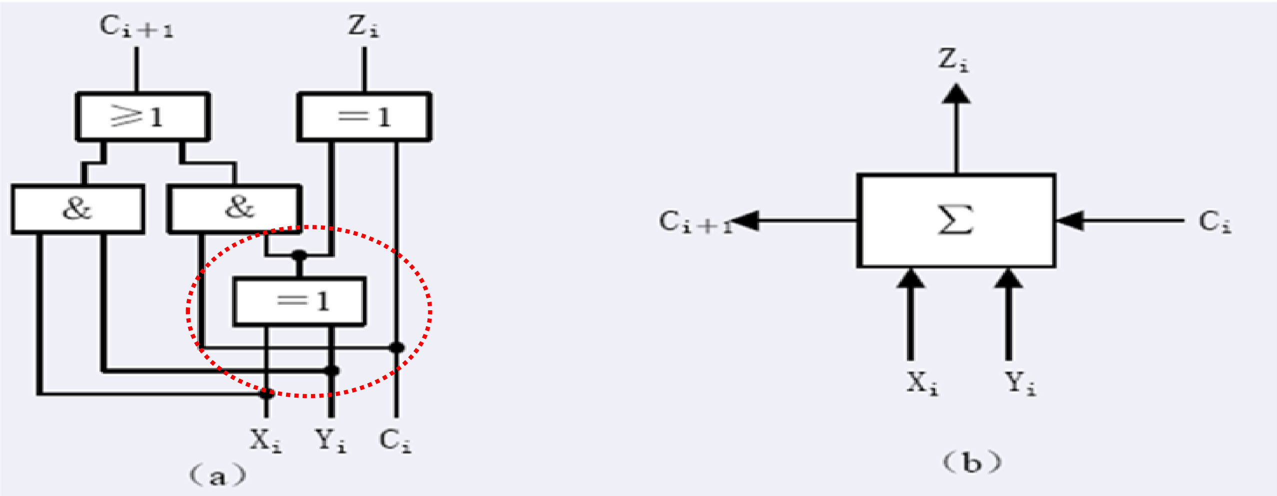
$$VF = Xs \cdot Ys \cdot \overline{Zs} + \overline{Xs} \cdot \overline{Ys} \cdot Zs$$

3. 一位全加器

设：一位全加器的输入为X_i 和 Y_i
低一位对该位的进位为 C_i
全加器的结果和进位用 Z_i 和 C_{i+1} 表示

一位全加器逻辑框图：





一位全加器逻辑及其框图

一位全加器逻辑表达式：

$$Z_i = X_i \oplus Y_i \oplus C_i$$

$$C_{i+1} = (X_i \cdot Y_i) + (X_i + Y_i) \cdot C_i$$

若令$G_i = X_i \cdot Y_i$， $P_i = X_i + Y_i$ ，则可写为：

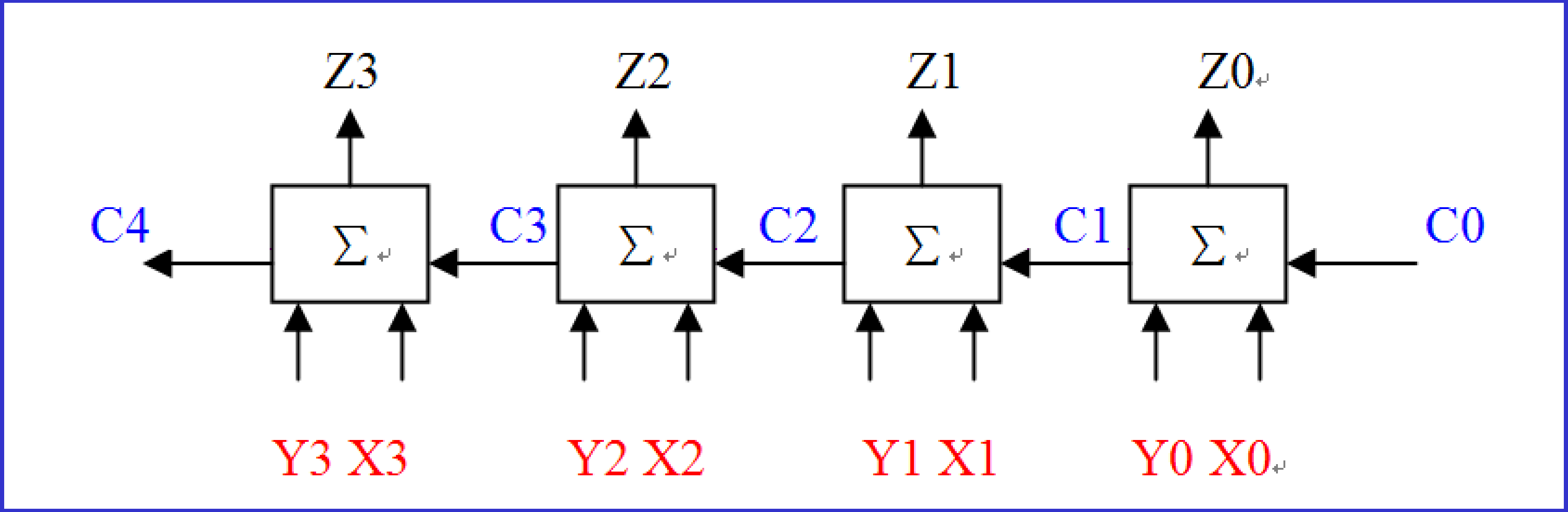
$$C_{i+1} = G_i + P_i \cdot C_i$$

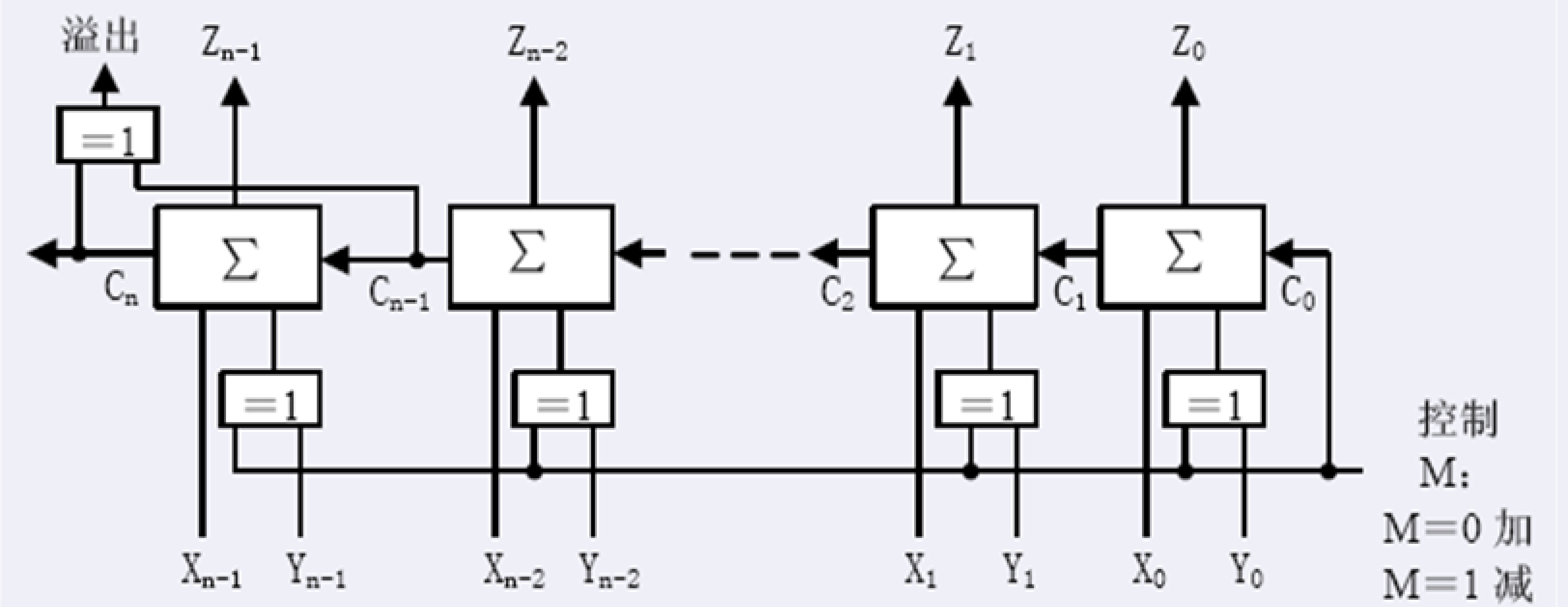
本位进位函数　　进位传递函数

Generation　　Propagate

4. n位加法器

（1）行波进位（串行进位）加法器





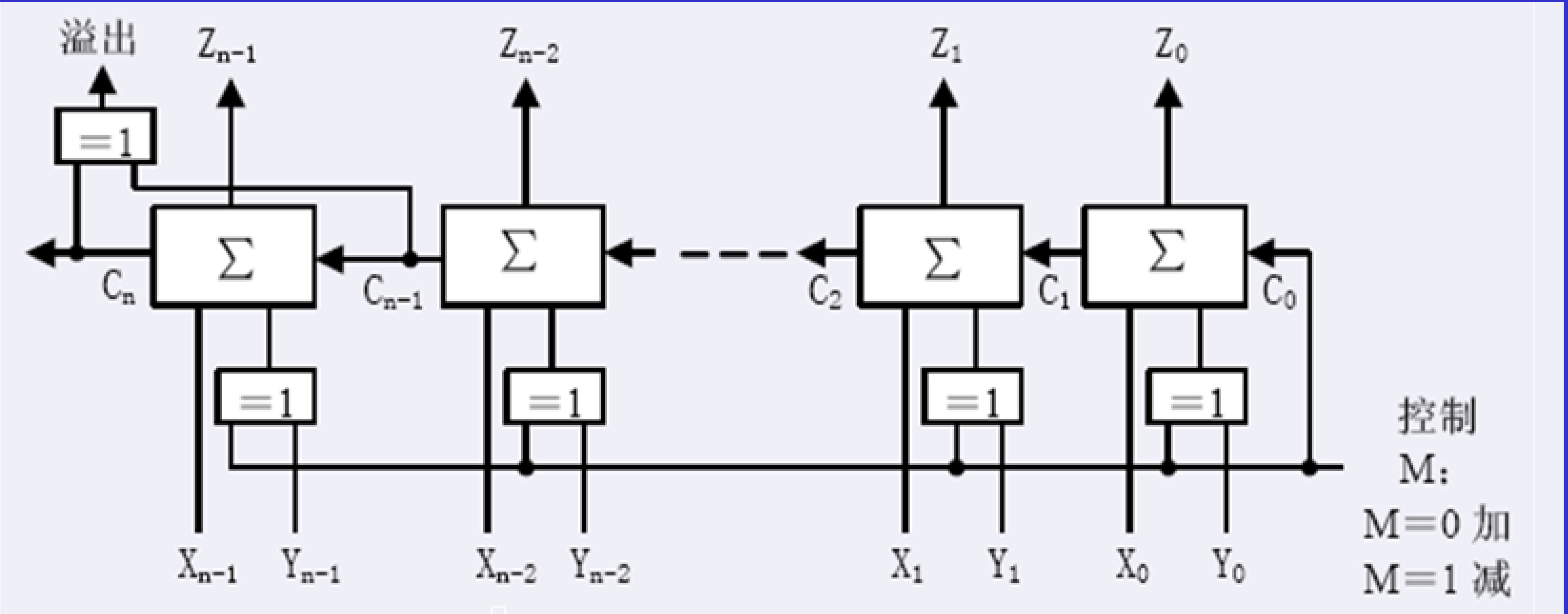
行波进位的n位加法/减法器

分析： 当 M=0 时，$Z = X + Y$

当 M=1 时，$Z = X + (\overline{Y} + 1)$

$= X - Y$

功能： 加/减法器



行波进位的n位加法/减法器

- 采用行波进位的n位加法器，各位和的产生时间关系如图：

(2) 先行进位加法器

$C_{i+1}=G_i+P_i\cdot C_i$

从式中可知，只要有输入Xi和Yi就能求出G_i和P_i，在已知输入C_i的情况下，便可以获得C_{i+1}

$$C_{i+1}=G_i+P_iC_i$$

$$C_{i+2}=G_{i+1}+P_{i+1}C_{i+1}\quad=G_{i+1}+P_{i+1}G_i+P_{i+1}P_iC_i$$

$$C_{i+3}=G_{i+2}+P_{i+2}C_{i+2}\quad=G_{i+2}+P_{i+2}G_{i+1}+P_{i+2}P_{i+1}G_i+P_{i+2}P_{i+1}P_iC_i$$

$$C_{i+4}=G_{i+3}+P_{i+3}C_{i+3}$$

$$=\underline{G_{i+3}+P_{i+3}G_{i+2}+P_{i+3}P_{i+2}G_{i+1}+P_{i+3}P_{i+2}P_{i+1}G_i}+\underline{P_{i+3}P_{i+2}P_{i+1}P_iC_i}$$

$$=G^*_{i+3}+P^*_{i+3}C_i$$

$$C_{i+1} = G_i + P_iC_i$$

$$C_{i+2} = G_{i+1} + P_{i+1}C_{i+1} = G_{i+1} + P_{i+1}G_i + P_{i+1}P_iC_i$$

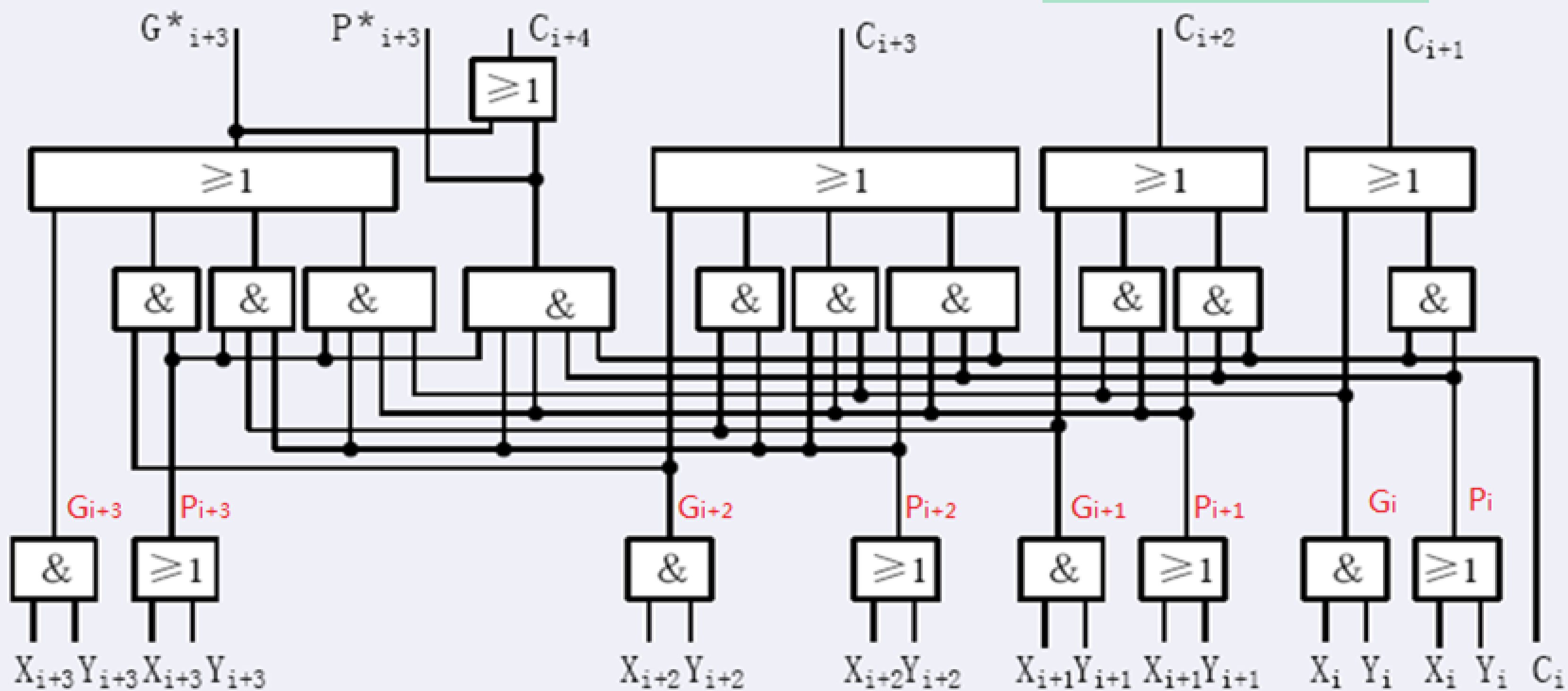
$$C_{i+3} = G_{i+2} + P_{i+2}C_{i+2} = G_{i+2} + P_{i+2}G_{i+1} + P_{i+2}P_{i+1}G_i + P_{i+2}P_{i+1}P_iC_i$$

$$C_{i+4} = G_{i+3} + P_{i+3}C_{i+3}$$

$$= G_{i+3} + P_{i+3}G_{i+2} + P_{i+3}P_{i+2}G_{i+1} + P_{i+3}P_{i+2}P_{i+1}G_i + P_{i+3}P_{i+2}P_{i+1}P_iC_i$$

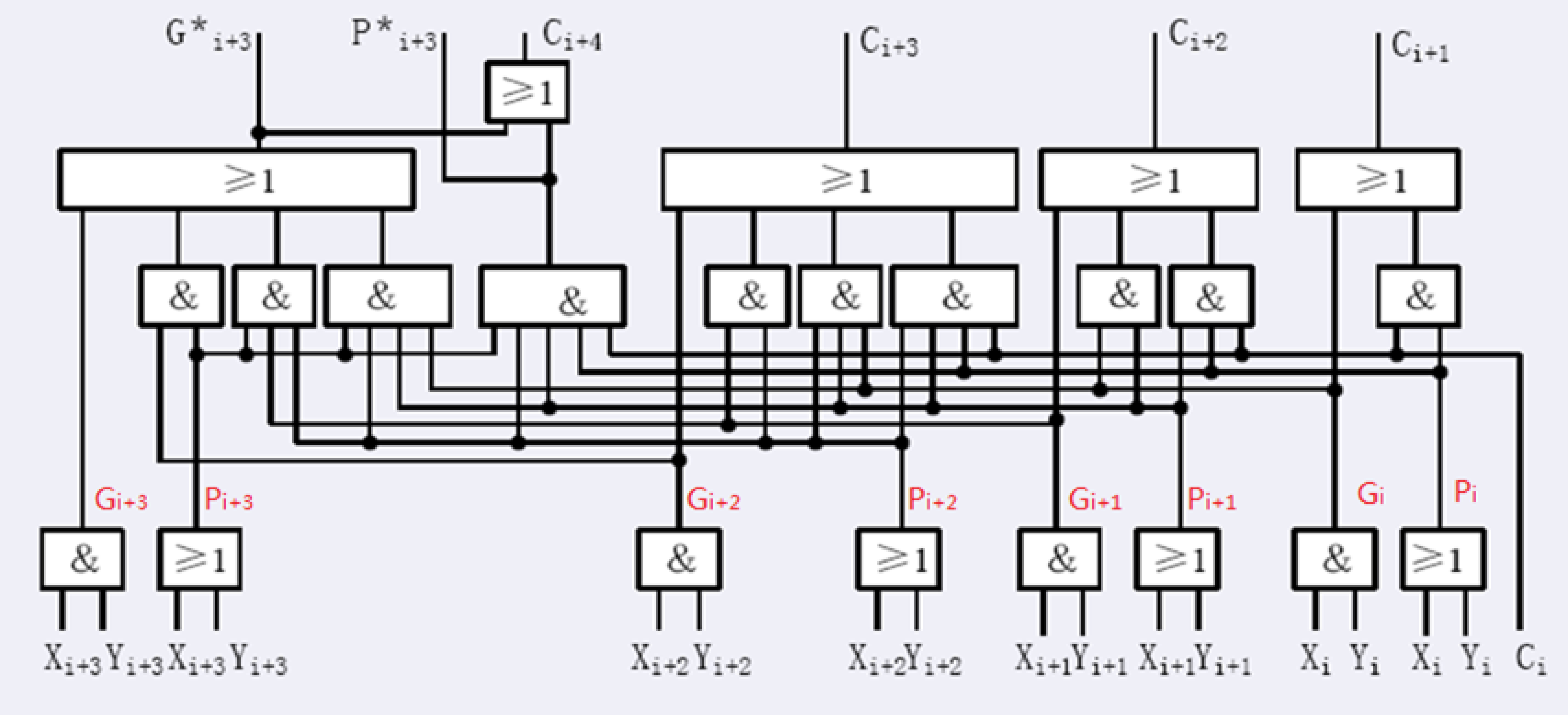
$$= G^*_{i+3} + P^*_{i+3}\ C_i$$

三级门的延时



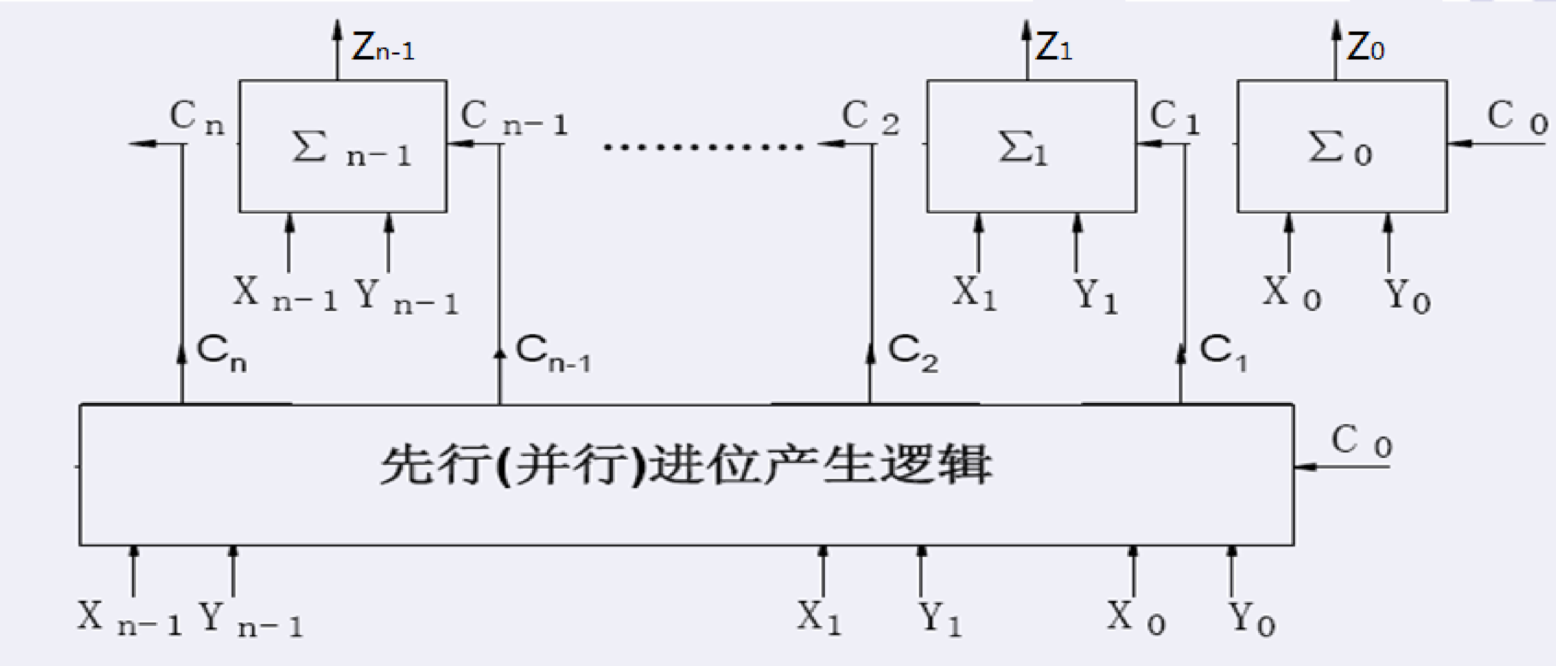
四位先行进位链电路

三级门的延时



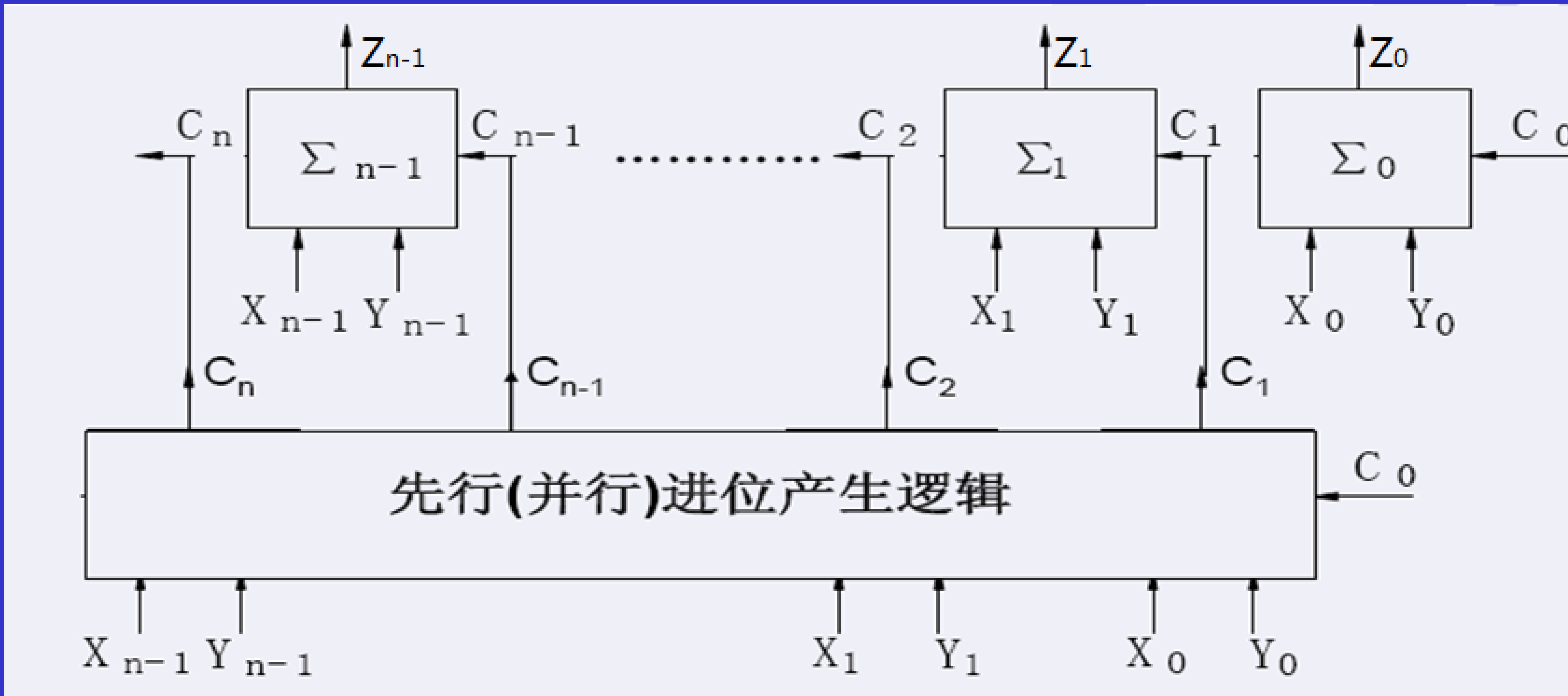
四位先行进位链电路

- 利用输入信号X_i、X_{i+1}、X_{i+2}、X_{i+3}和Y_i、Y_{i+1}、Y_{i+2}、Y_{i+3}以及C_i，通过与或逻辑电路的组合就可以**同时**将上面四个进位信号产生出来。



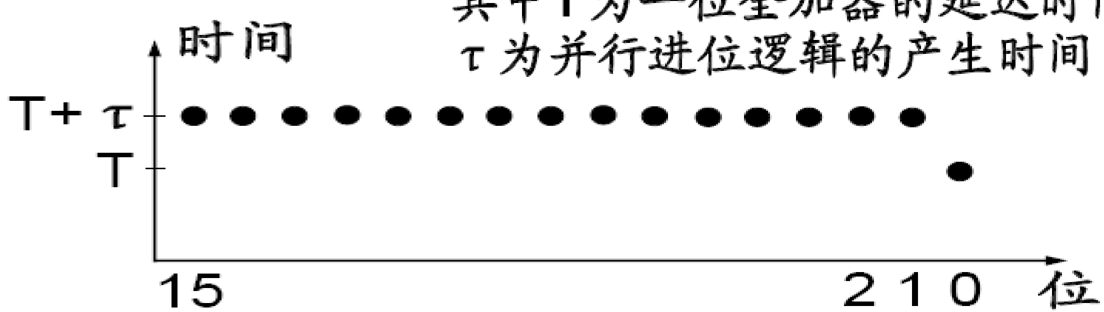
n位并行进位加法器及先行进位

- 将这些先行形成的**进位信号并行地加到加法器**上，则加法器就不必等待进位产生，从而大大地提高了加法器的速度。也就是说在进行加法运算之前，各位加法器所需要的进位已经产生出来。这将加快加法运算的速度，这就是先行进位的来由。

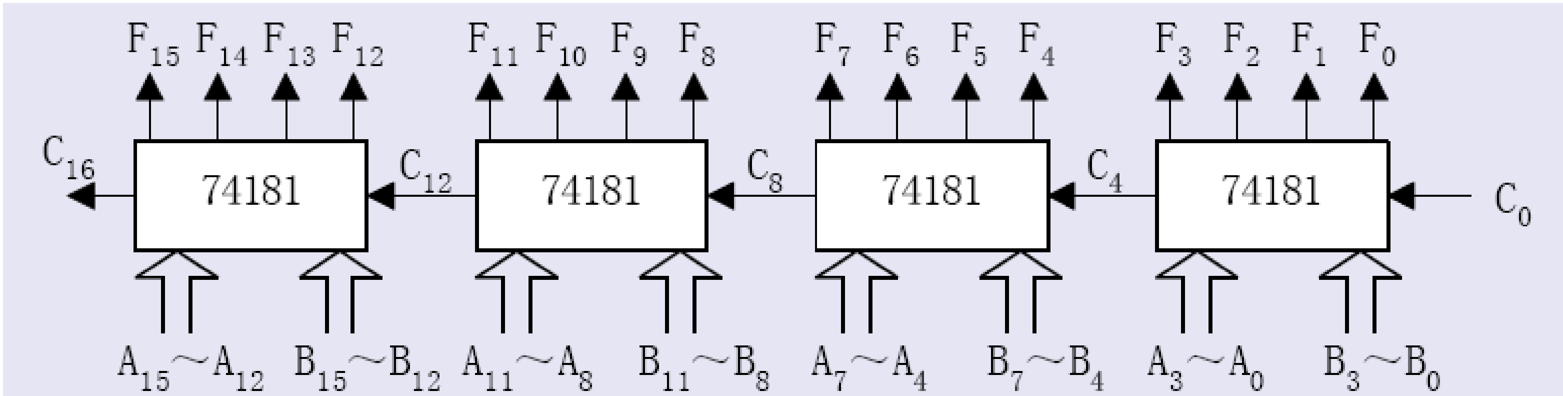


n位并行进位加法器及先行进位

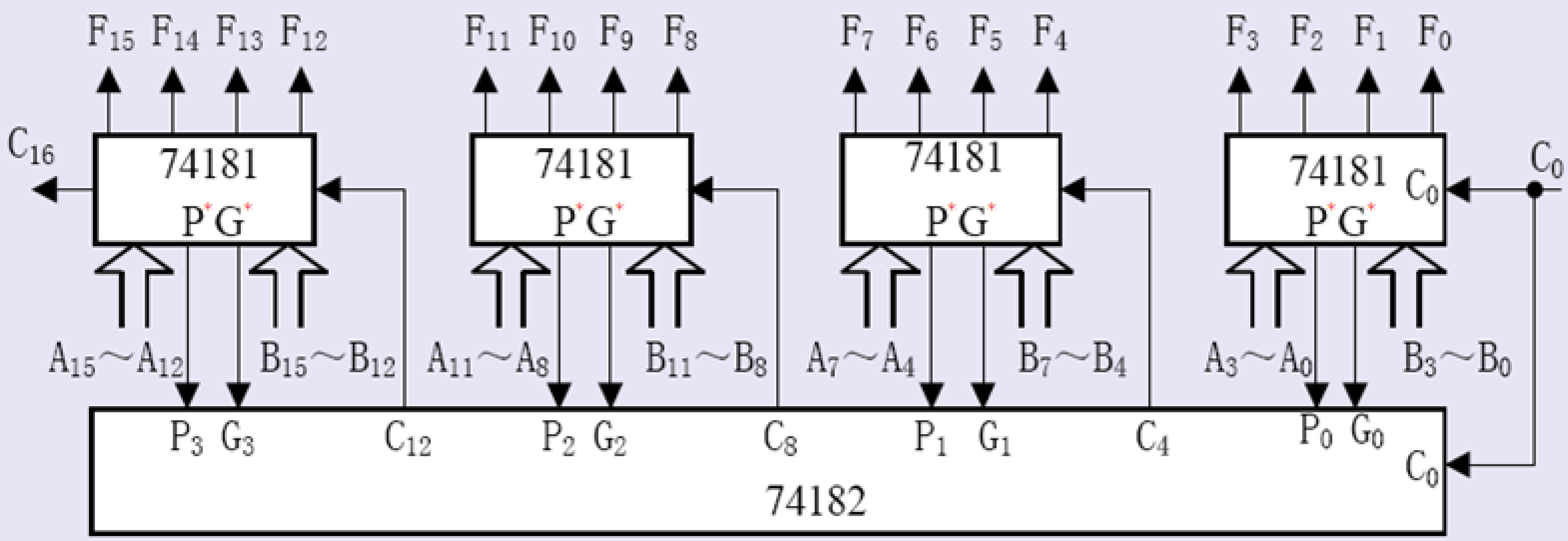
16位并行加法器各位和的产生时间如图
其中**T**为一位全加器的延迟时间，
τ为并行进位逻辑的产生时间



(3) 先行进位加法器的**级联**



由74181构成组间串行进位的16位ALU



由74181和74182构成组内组间均并行进位的16位ALU

5. 移码加减运算

$$[X]_{移} \pm [Y]_{移} = (2^{n-1}+X) \pm (2^{n-1}+Y) = 2^n + (X \pm Y)$$

$$= [X \pm Y]_{补} \longleftrightarrow [X \pm Y]_{移} \quad MOD\ 2^n$$

符号位取反

定点**整数移码**的**加减运算法则**:

①对两移码求和差时，首先对该两移码求和差;

②然后，对结果进行修正——将结果的符号取反。这样，就可以得到正确的结果。

6. BCD数加法器（课后学习）

- **8421 BCD码**

  8421 BCD码只利用了四位二进制编码的0000到1001这十种来表示十进制数的0到9。剩余的六种：1010，1011，1100，1101，1110，1111对用于表示十进制数来说是非法的。一旦在定义的BCD运算中出现这六种编码，结果一定是错误的。

- 8421 BCD码加法运算也可以用多位全加器实现，在运算过程中有可能产生上面提到的错误结果。

a:

```
  0110
+ 0010
------
  1000
```

b:

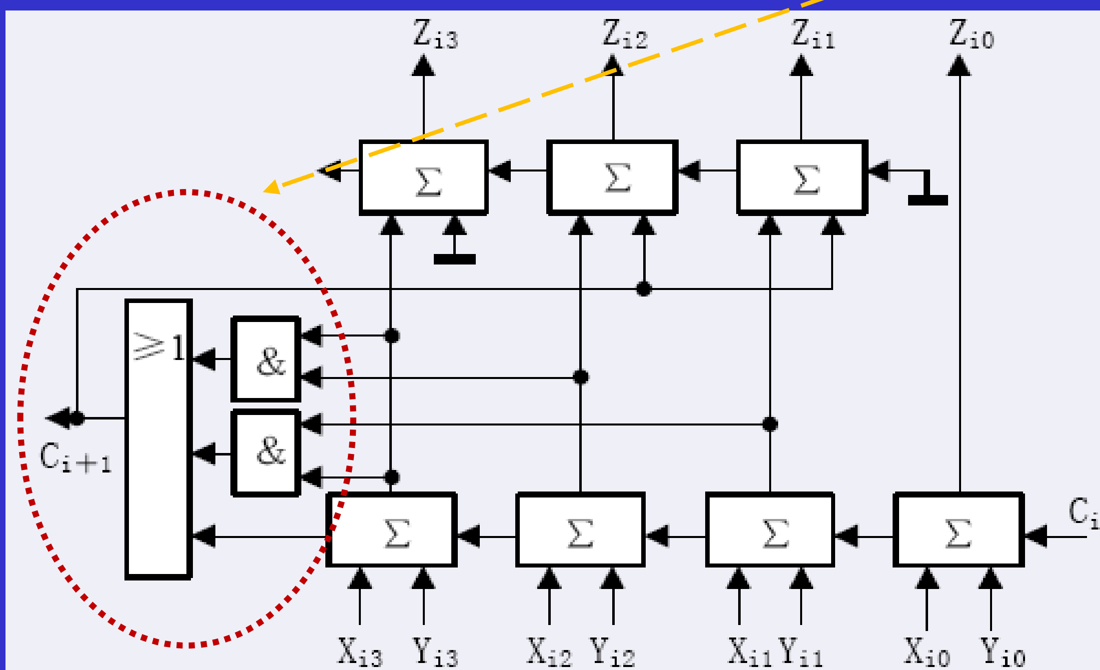
```
  0110
+ 0111
------
  1101
```

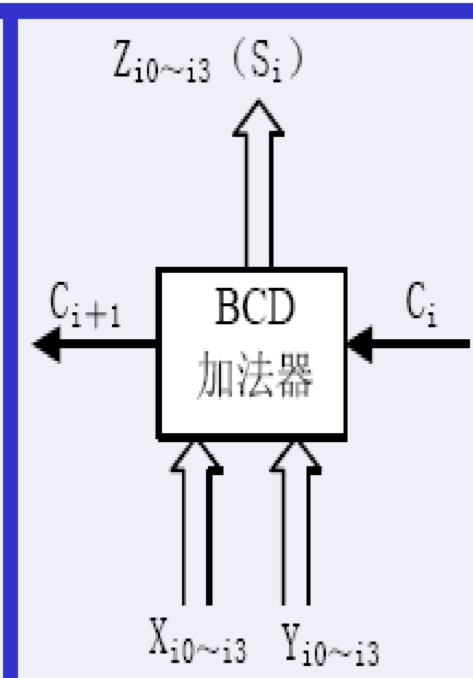
- 8421 BCD码加法运算也可以用多位全加器实现，在运算过程中有可能产生上面提到的错误结果。

a:		b:	
	0110		0110
	+ 0010		+ 0111
	1000		1101

- **校正**

运算中低四位相加的结果＞9或有由bit3向bit4的进位，则结果加06H；





本章作业3-1

第17（1）（2）、18（2）、19（2）题

注：本次作业与下次作业3-2一起交

3.1.2 乘法运算

1. 原码一位乘法运算

- 原码一位乘法的法则:

①乘积的符号为被乘数的符号位与乘数的符号位相异或;

②乘积的绝对值为被乘数的绝对值与乘数的绝对值之积。即

$$[X]_{原} \times [Y]_{原} = (X_0 \oplus Y_0)(|X| \times |Y|)$$

- 手工乘法运算

例 若$[X]_原=0.1101$，$[Y]_原=1.1011$，求两者之积。

解：乘积的符号为$0 \oplus 1 = 1$

手算过程如下：

$$
\begin{array}{r l}
1101 & \\
\times\ \ 1011 & \\
\hline
1101 & \quad \text{部分积} \\
1101 & \quad \text{部分积} \\
0000 & \quad \text{部分积} \\
1101 & \quad \text{部分积} \\
\hline
.10001111 &
\end{array}
$$

综上，乘积的原码为：1.10001111

- 机器实现思路：部分积，右移，4位累加

例 $[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

例　$[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, $+X$

例 $[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					

例 $[X]_{原}=0.1101$，$[Y]_{原}=1.1011$，求两者之积。

解：$|X|=1101$，$|Y|=1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$，+X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次

例 $[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	

例 $[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次

例 $[X]_{原} = 0.1101$，$[Y]_{原} = 1.1011$，求两者之积。

解：$|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次
0	0	0	0	0					$A_0=0$, +0
0	1	0	0	1	1	1	1	0	

例 $[X]_原 = 0.1101$，$[Y]_原 = 1.1011$，求两者之积。

解： $|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次
0	0	0	0	0					$A_0=0$, +0
0	1	0	0	1	1	1	1	0	
0	0	1	0	0	1	1	1	1	→右移一次

例　$[X]_{原}=0.1101$，$[Y]_{原}=1.1011$，求两者之积。

解：$|X|=1101$，$|Y|=1011$，$X_S \oplus Y_S=0 \oplus 1=1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次
0	0	0	0	0					$A_0=0$, +0
0	1	0	0	1	1	1	1	0	
0	0	1	0	0	1	1	1	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	0	1	1	1	1	1	

例 $[X]_{原}=0.1101$，$[Y]_{原}=1.1011$，求两者之积。

解：$|X|=1101$，$|Y|=1011$，$X_S \oplus Y_S=0 \oplus 1=1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	$A_0=1$, +X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次
0	0	0	0	0					$A_0=0$, +0
0	1	0	0	1	1	1	1	0	
0	0	1	0	0	1	1	1	1	→右移一次
+0	1	1	0	1					$A_0=1$, +X
1	0	0	0	1	1	1	1	1	
0	1	0	0	0	1	1	1	1	→右移一次

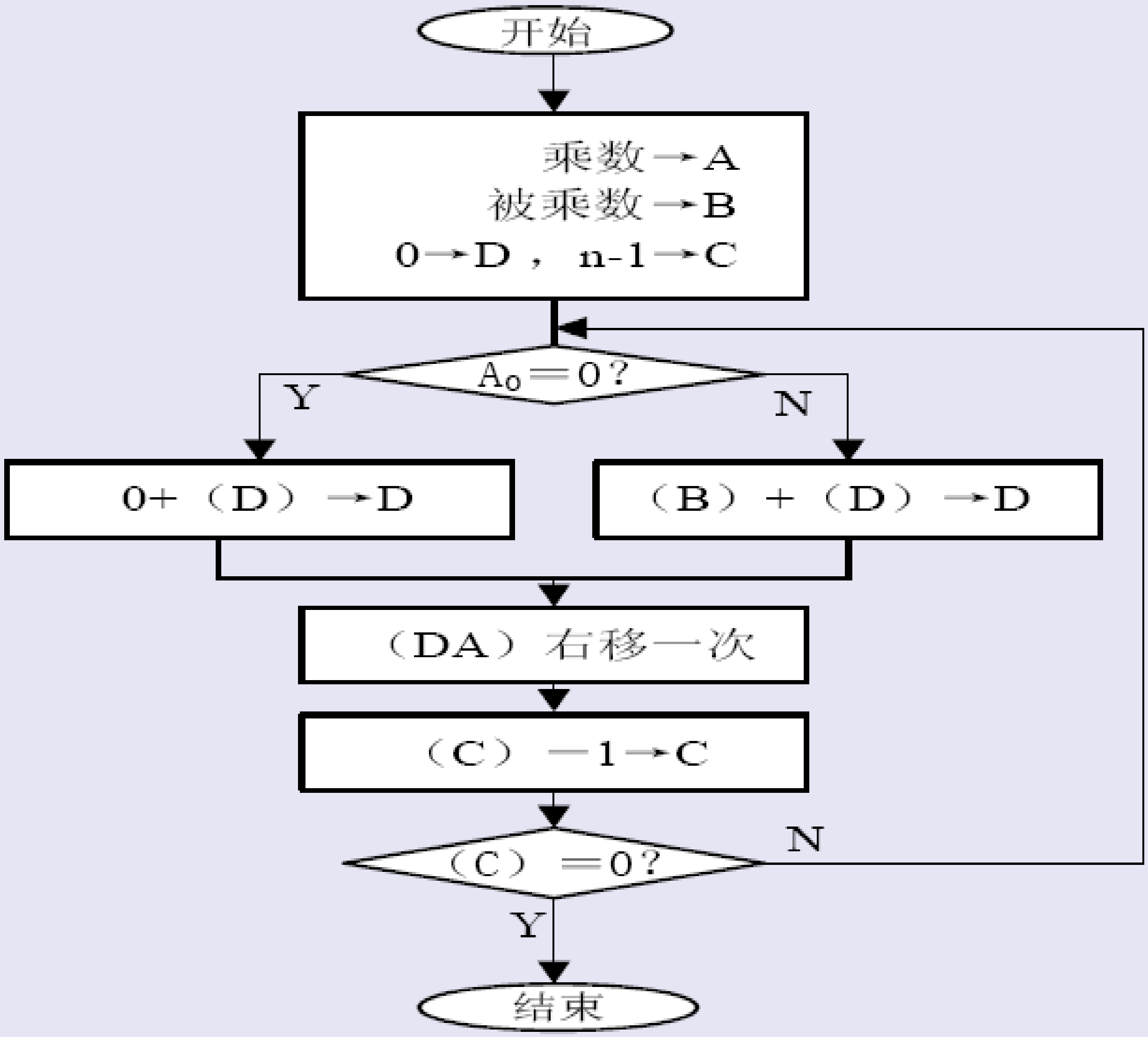
例 $[X]_原 = 0.1101$，$[Y]_原 = 1.1011$，求两者之积。

解： $|X| = 1101$，$|Y| = 1011$，$X_S \oplus Y_S = 0 \oplus 1 = 1$

	D				A			A_0	操作
0	0	0	0	0	1	0	1	1	A_0=1，+X
+0	1	1	0	1					
0	1	1	0	1					
0	0	1	1	0	1	1	0	1	→右移一次
+0	1	1	0	1					A_0=1，+X
1	0	0	1	1	1	1	0	1	
0	1	0	0	1	1	1	1	0	→右移一次
0	0	0	0	0					A_0=0，+0
0	1	0	0	1	1	1	1	0	
0	0	1	0	0	1	1	1	1	→右移一次
+0	1	1	0	1					A_0=1，+X
1	0	0	0	1	1	1	1	1	
0	1	0	0	0	1	1	1	1	→右移一次

拼接符号后积为：1. 1 0 0 0 1 1 1 1

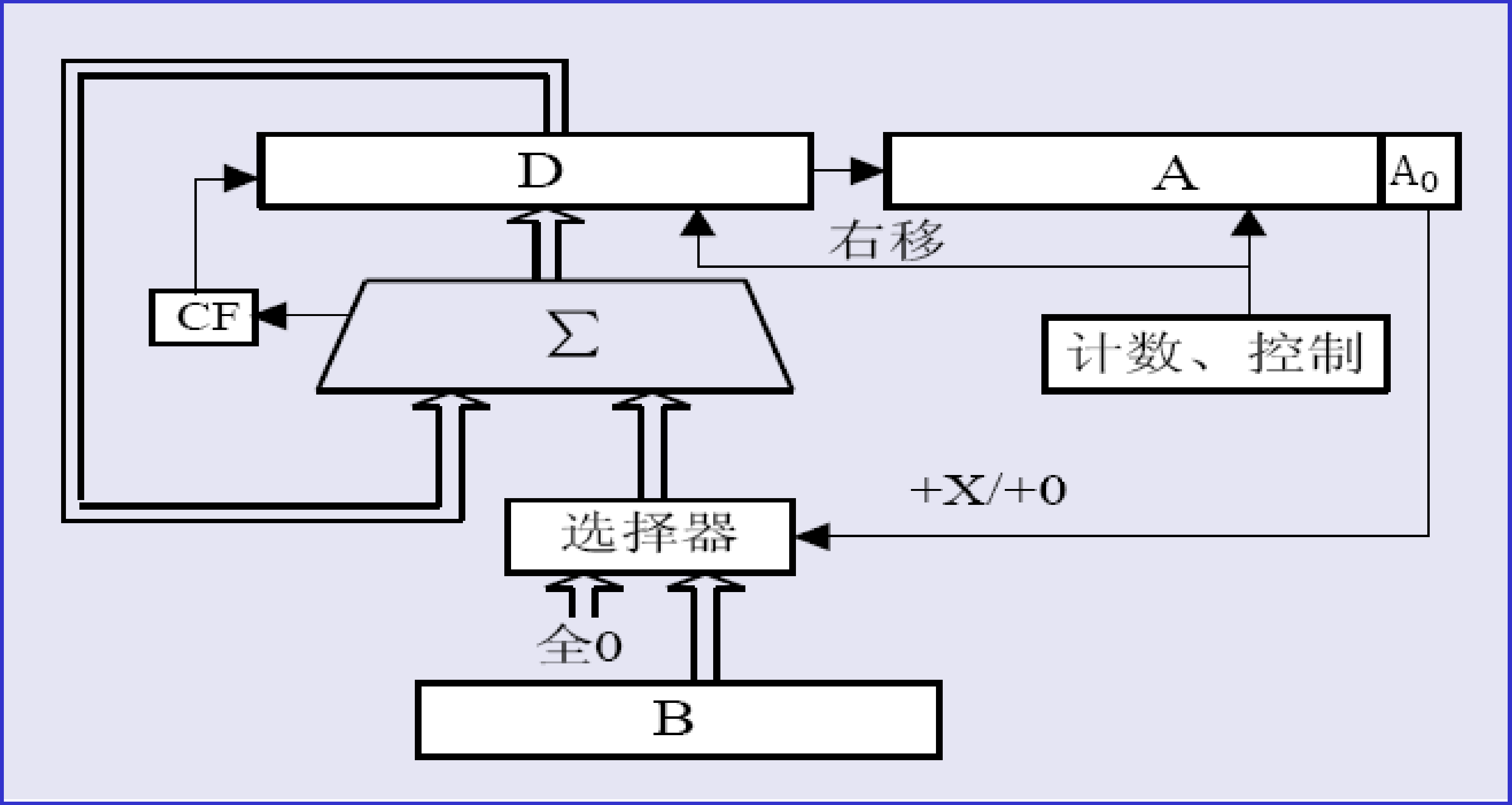
- 实现流程



绝对值乘法思路框图

```
      1101   B
  ×   1011   A
  --------
      1101
     1101
    0000
   1101
-----------
 .10001111
```

• 原码一位乘法器框图



D 部分积（初始清0），A 乘数，B 被乘数

D, A 乘积

• 原码二位乘法运算（了解）：

Y_{i+1}	Y_i	C	操　作
0	0	0	+0，　右移2次，C = 0
0	0	1	+｜X｜，右移2次，C = 0
0	1	0	+｜X｜，右移2次，C = 0
0	1	1	+2｜X｜，右移2次，C = 0
1	0	0	+2｜X｜，右移2次，C = 0
1	0	1	−｜X｜，右移2次，C = 1
1	1	0	−｜X｜，右移2次，C = 1
1	1	1	+0，　右移2次，C = 1

原码二位乘法的法则表

2. 补码一位乘法运算

- **布斯（Booth）法**

假定被乘数X和乘数Y均为用补码表示的纯小数，分别为：

$$[X]_{补} = X_0 . X_{-1}X_{-2}\ldots X_{-(n-1)}$$

$$[Y]_{补} = Y_0 . Y_{-1}Y_{-2}\ldots Y_{-(n-1)} \quad 0 \quad \text{附加位}$$

其中X_0、Y_0 是它们的符号位。

则布斯法补码一位乘法的算法公式为：

$$[X \cdot Y]_{补} = [X]_{补} \cdot [(Y_{-1} - Y_0)2^0 + (Y_{-2} - Y_{-1})2^{-1} + (Y_{-3} - Y_{-2})2^{-2} + \ldots\ldots + (Y_{-(n-1)} - Y_{-(n-2)})2^{-(n-2)} + (0 - Y_{-(n-1)})2^{-(n-1)}]$$

$$Y' = (Y_{-1} - Y_0).\ (Y_{-2} - Y_{-1})\ (Y_{-3} - Y_{-2})\ \ldots\ldots\ (Y_{-(n-1)} - Y_{-(n-2)})\ (0 - Y_{-(n-1)})$$

则： $[X \cdot Y]_{补} = [X]_{补} \cdot Y'$

Y_i Y_{i-1}	$Y_{i-1}-Y_i$	操　　作
0　0	0	+0，右移一次
0　1	1	+$[X]_{补}$，右移一次
1　0	−1	+$[-X]_{补}$，右移一次
1　1	0	+0，右移一次

乘数相邻两位的操作规律

- 布斯法的**运算法则**：

①乘数与被乘数均用补码表示，连同符号位一起参加运算。

②乘数最低位后增加一个附加位（可用A_{-1}表示），初始设定为0。

③从附加位开始，依据上表所示的操作规律，完成运算。

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解： $[X]_{补} = 00.1010$，$[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解： $[X]_补 = 00.1010$， $[Y]_补 = 11.0011$，

而 $[-X]_补 = 11.0110$。

符号		D				A					A_{-1}	操 作
0	0	0	0	0	0	1	0	0	1	1	0	

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解： $[X]_{补} = 00.1010$， $[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

符号		D				A					A_{-1}	操 作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	0							

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补}$ = 00.1010，$[Y]_{补}$ = 11.0011，

而$[-X]_{补}$ = 11.0110。

符号		D				A					A_{-1}	操 作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位

例　已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补}$ = 00.1010，$[Y]_{补}$ = 11.0011，

而$[-X]_{补}$ = 11.0110。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							

例　已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补} = 00.1010$，　$[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解： $[X]_{补} = 00.1010$， $[Y]_{补} = 11.0011$，

而 $[-X]_{补} = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							$+[X]_{补}$
0	0	0	1	1	1							

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补} = 00.1010$，$[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							$+[X]_{补}$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解: $[X]_补 = 00.1010$，$[Y]_补 = 11.0011$，

而$[-X]_补 = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_补$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							$+[X]_补$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位
0	0	0	0	0	0							+0
0	0	0	0	1	1							

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解: $[X]_补 = 00.1010$, $[Y]_补 = 11.0011$,

而$[-X]_补 = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_补$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							$+[X]_补$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位
0	0	0	0	0	0							+0
0	0	0	0	1	1							
0	0	0	0	0	1	1	1	1	0	1	0	右移一位

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补} = 00.1010$，$[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

符号	D	A	A_{-1}	操作
0 0	0 0 0 0	1 0 0 1 1	0	
1 1	0 1 1 0			$+[-X]_{补}$
1 1	0 1 1 0			
1 1	1 0 1 1	0 1 0 0 1	1	右移一位
0 0	0 0 0 0			+0
1 1	1 0 1 1			
1 1	1 1 0 1	1 0 1 0 0	1	右移一位
0 0	1 0 1 0			$+[X]_{补}$
0 0	0 1 1 1			
0 0	0 0 1 1	1 1 0 1 0	0	右移一位
0 0	0 0 0 0			+0
0 0	0 0 1 1			
0 0	0 0 0 1	1 1 1 0 1	0	右移一位
1 1	0 1 1 0			$+[-X]_{补}$
1 1	0 1 1 1			

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解： $[X]_{补}$ = 00.1010， $[Y]_{补}$ = 11.0011，

而 $[-X]_{补}$ = 11.0110。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							+ $[X]_{补}$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位
0	0	0	0	0	0							+0
0	0	0	0	1	1							
0	0	0	0	0	1	1	1	1	0	1	0	右移一位
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	1							
1	1	1	0	1	1	1	1	1	1	0		右移一位

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

解：　$[X]_{补} = 00.1010$，$[Y]_{补} = 11.0011$，

而$[-X]_{补} = 11.0110$。

符号		D				A					A_{-1}	操作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							$+[X]_{补}$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位
0	0	0	0	0	0							+0
0	0	0	0	1	1							
0	0	0	0	0	1	1	1	1	0	1	0	右移一位
1	1	0	1	1	0							$+[-X]_{补}$
1	1	0	1	1	1							
1	1	1	0	1	1	1	1	1	1	0		右移一位

例 已知X = 0.1010, Y = -0.1101。利用布斯法求积。

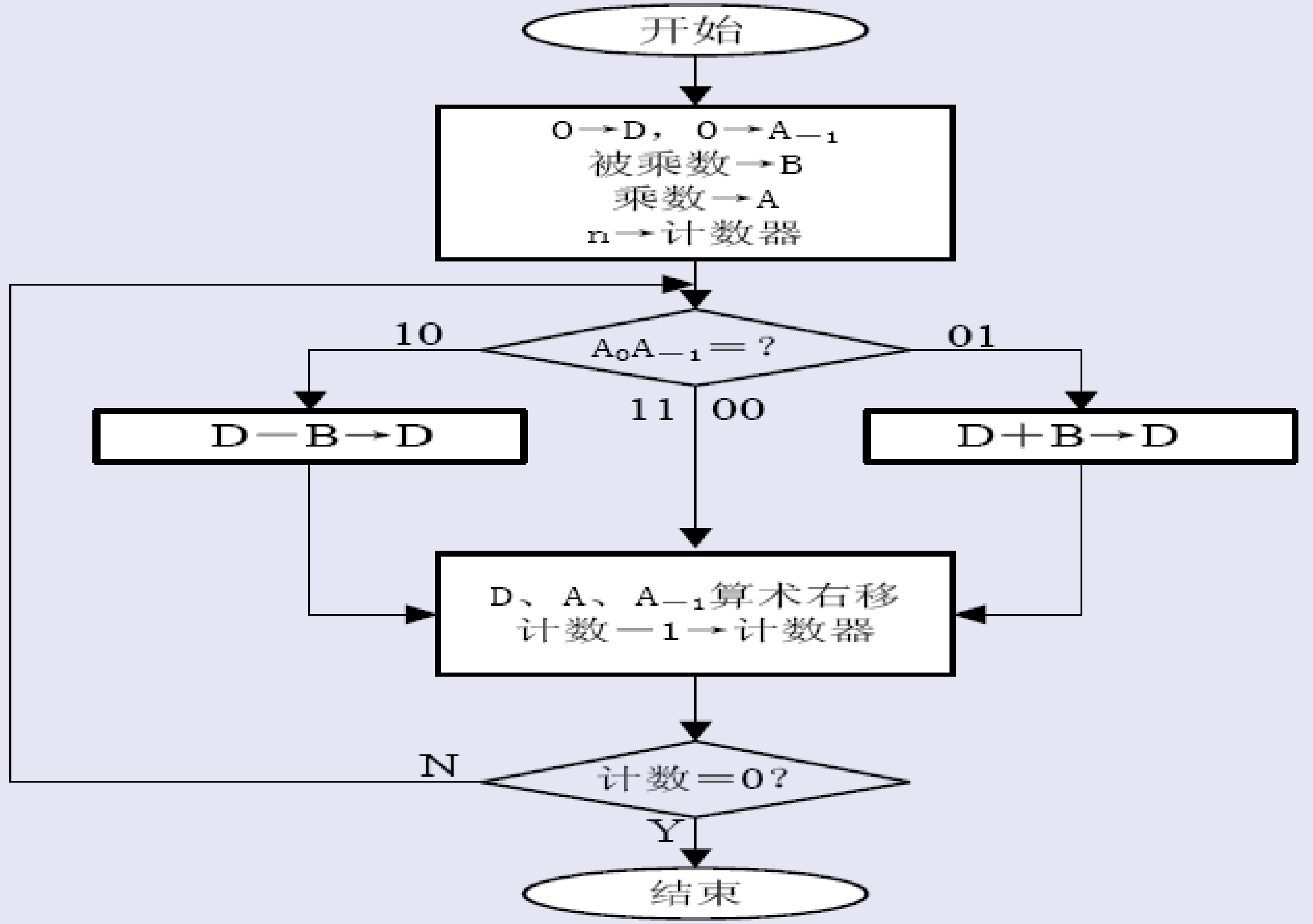
解：　$[X]_{补}$ = 00.1010，$[Y]_{补}$ = 11.0011，

而$[-X]_{补}$ = 11.0110。

符号		D				A					A_{-1}	操 作
0	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	0							
1	1	1	0	1	1	0	1	0	0	1	1	右移一位
0	0	0	0	0	0							+0
1	1	1	0	1	1							
1	1	1	1	0	1	1	0	1	0	0	1	右移一位
0	0	1	0	1	0							+ $[X]_{补}$
0	0	0	1	1	1							
0	0	0	0	1	1	1	1	0	1	0	0	右移一位
0	0	0	0	0	0							+0
0	0	0	0	1	1							
0	0	0	0	0	1	1	1	1	0	1	0	右移一位
1	1	0	1	1	0							+ $[-X]_{补}$
1	1	0	1	1	1							
1	1	1	0	1	1	1	1	1	1	0		右移一位

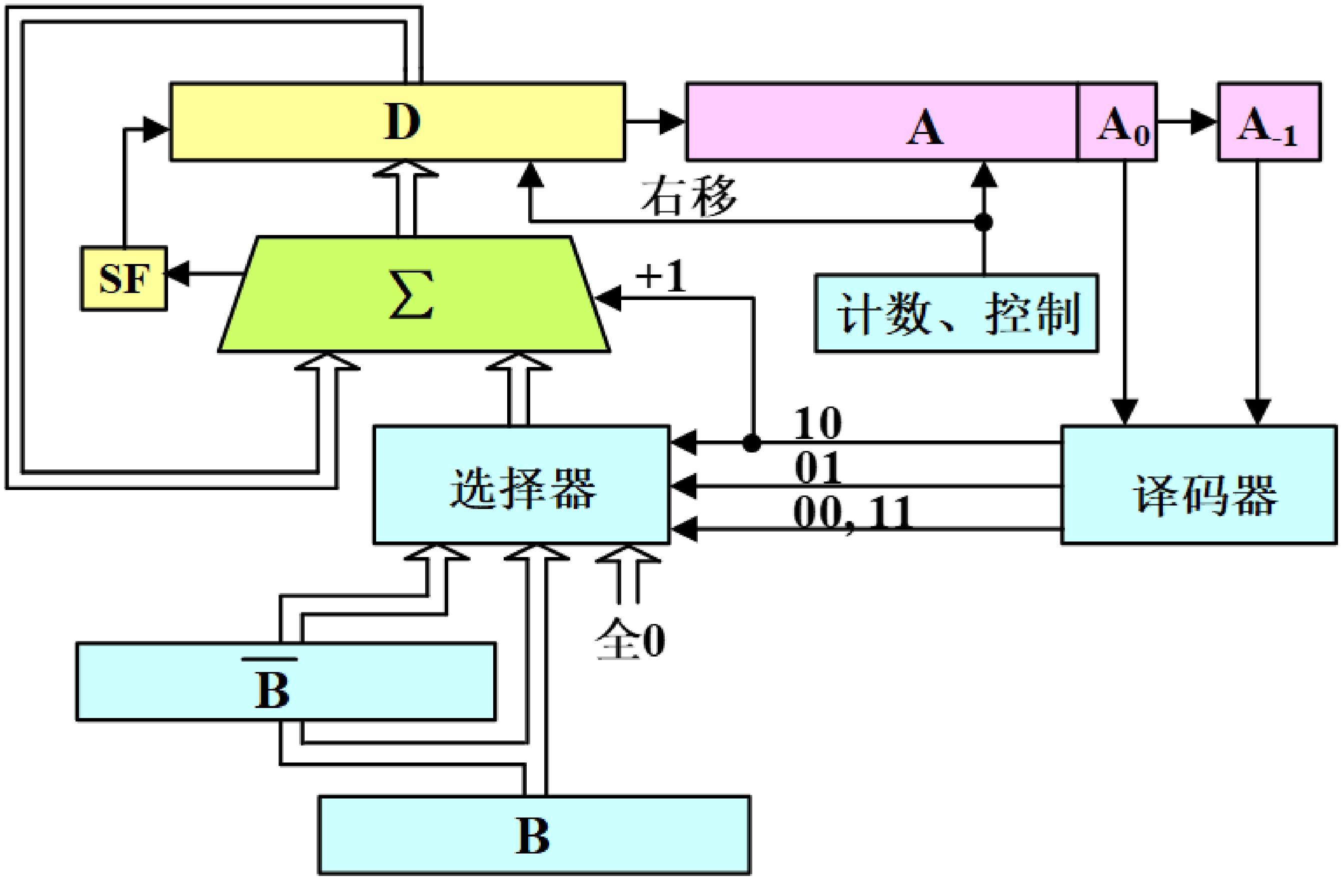
$[X \cdot Y]_{补}$ = 1.01111110

• 运算过程



布斯补码一位乘法流程图

- 补码乘法器框图（课后学习）



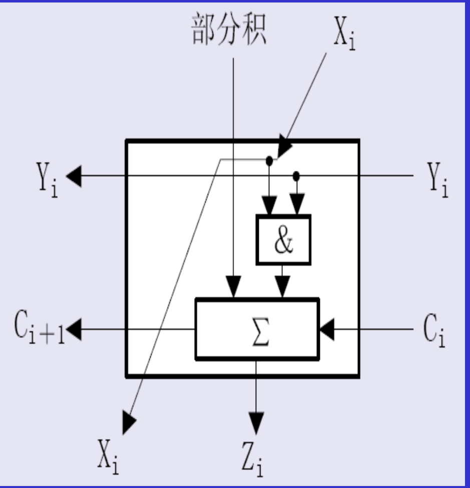
补码一位乘法器(Booth算法)框图

阵列乘法器（课后学习）

- 阵列乘法器（原码一位乘）

设$X = X_3X_2X_1X_0$，$Y = Y_3Y_2Y_1Y_0$，计算$X \cdot Y = ?$

			X_3	X_2	X_1	X_0
		$\times$	Y_3	Y_2	Y_1	Y_0
			X_3Y_0	X_2Y_0	X_1Y_0	X_0Y_0
		X_3Y_1	X_2Y_1	X_1Y_1	X_0Y_1	
	X_3Y_2	X_2Y_2	X_1Y_2	X_0Y_2		
X_3Y3	X_2Y_3	X_1Y_3	X_0Y_3			
Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0



3.1.3 除法运算

1. 原码除法运算

- 原码除法的法则

①除数≠0；定点纯小数时，｜被除数｜＜｜除数｜；定点纯整数时，｜被除数｜＞｜除数｜。

②与原码乘法类似的是原码除法商的符号和商的值也是分别处理的。

③商的符号等于被除数的符号与除数的符号相异或,商的值就等于被除数的绝对值除以除数的绝对值。

④最后将商的符号与商的值拼接在一起就得到原码除法的商。

- 手工除法过程

例 设X = 0.1011，Y = 0.1101，求X ÷ Y = ?

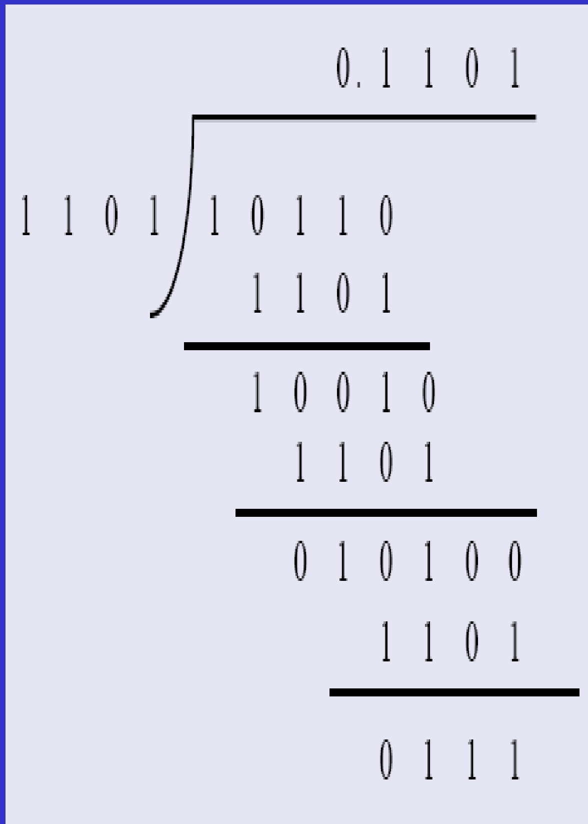
解：由题意可知，假定被除数X和除数Y均为正数。未来商的符号也为正。

```
                 0. 1  1  0  1
          ____________________
1 1 0 1 ) 1  0  1  1  0
             1  1  0  1
          ------------
             1  0  0  1  0
                1  1  0  1
             ----------------
                0  1  0  1  0  0
                      1  1  0  1
                   ---------------
                      0  1  1  1
```

X/Y＝0.1101

余数＝0.0111×2^{-4}

商的符号＝$0\oplus0=0$



- **恢复余数法**
  - 符号位单独处理。
  - 对于定点纯小数，被除数左移一位，减除数，
    若够减，　上商为1；
    若不够减，上商为0，同时加除数（恢复余数）。
  - 余数左移一位，减除数，
    若够减，　上商为1；
    若不够减，上商为0，同时加除数（恢复余数）。
  - 重复上述过程直到除尽或精度达到要求。
  - 拼接商符得到商，即可获得除法的结果。

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且 $|Y| \neq 0$

已知$[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 = $1 \oplus 0 = 1$;

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且 $|Y| \neq 0$

已知 $[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 $= 1 \oplus 0 = 1$;

绝对值相除见表。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且 $|Y| \neq 0$

已知$[X]_原 = 1.10001011$，

$[Y]_原 = 0.1110$。

商符 $= 1 \oplus 0 = 1$；

绝对值相除见表。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\vert Y\vert$
0	0	0	0	1	1	0	1	1	0	1	够减，商为1

例

若被除数$X = -0.10001011$，

除数$Y = 0.1110$，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且$|Y| \neq 0$

已知$[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 $= 1 \oplus 0 = 1$;

绝对值相除见表。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\|Y\|$
0	0	0	0	1	1	0	1	1	0	1	够减，商为1
0	0	0	1	1	0	1	1	0	1		左移一位

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且 $|Y| \neq 0$

已知 $[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 $= 1 \oplus 0 = 1$;

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

|X| < |Y|，

且|Y|≠0

已知$[X]_{原}$ = 1.10001011，

$[Y]_{原}$ = 0.1110。

商符 = 1 ⊕ 0 = 1；

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		＋|Y|

例

若被除数X = -0.10001011，
除数Y = 0.1110，
试利用原码恢复余数法求商及余数。

解：该例满足
$|X| < |Y|$，
且 $|Y| \neq 0$

已知 $[X]_{原} = 1.10001011$，
$[Y]_{原} = 0.1110$。

商符 $= 1 \oplus 0 = 1$;
绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		$+\lvert Y\rvert$
0 0	0 1 1 0 1 1 0 1	0	恢复余数

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且 $|Y| \neq 0$

已知 $[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 = $1 \oplus 0 = 1$；

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		$-\|Y\|$
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		$-\|Y\|$
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		$+\|Y\|$
0 0	0 1 1 0 1 1 0 1	0	恢复余数
0 0	1 1 0 1 1 0 1 0		左移一位

例

若被除数$X = -0.10001011$，

除数$Y = 0.1110$，

试利用原码恢复余数法求商及余数。

解：该例满足

$|X| < |Y|$，

且$|Y| \neq 0$

已知$[X]_{原} = 1.10001011$，

$[Y]_{原} = 0.1110$。

商符 $= 1 \oplus 0 = 1$;

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		−|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		−|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		+|Y|
0 0	0 1 1 0 1 1 0 1	0	恢复余数
0 0	1 1 0 1 1 0 1 0		左移一位
1 1	0 0 1 0		−|Y|
1 1	1 1 1 1 1 0 1 0	0	不够减，商为0

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

|X| < |Y|，

且|Y|≠0

已知$[X]_{原}$ = 1.10001011，

$[Y]_{原}$ = 0.1110。

商符 = 1 ⊕ 0 = 1；

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		+|Y|
0 0	0 1 1 0 1 1 0 1	0	恢复余数
0 0	1 1 0 1 1 0 1 0		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 1 1 1 1 0 1 0	0	不够减，商为0
0 0	1 1 1 0		+|Y|
0 0	1 1 0 1 1 0 1 0	0	恢复余数

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

|X| < |Y|，

且 |Y| ≠0

已知$[X]_{原}$ = 1.10001011，

$[Y]_{原}$ = 0.1110。

商符 = 1 ⊕ 0 = 1;

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		+|Y|
0 0	0 1 1 0 1 1 0 1	0	恢复余数
0 0	1 1 0 1 1 0 1 0		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 1 1 1 1 0 1 0	0	不够减，商为0
0 0	1 1 1 0		+|Y|
0 0	1 1 0 1 1 0 1 0	0	恢复余数
0 1	1 0 1 1 0 1 0 0		左移一位

例

若被除数X = -0.10001011，

除数Y = 0.1110，

试利用原码恢复余数法求商及余数。

解：该例满足

|X| < |Y|，

且 |Y| ≠0

已知$[X]_{原}$ = 1.10001011，

$[Y]_{原}$ = 0.1110。

商符 = 1 ⊕ 0 = 1;

绝对值相除见表。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	0 0 1 1 0 1 1 0	1	够减，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 0 0 0 1 1 0 1	0	不够减，商为0
0 0	1 1 1 0		＋|Y|
0 0	0 1 1 0 1 1 0 1	0	恢复余数
0 0	1 1 0 1 1 0 1 0		左移一位
1 1	0 0 1 0		－|Y|
1 1	1 1 1 1 1 0 1 0	0	不够减，商为0
0 0	1 1 1 0		＋|Y|
0 0	1 1 0 1 1 0 1 0	0	恢复余数
0 1	1 0 1 1 0 1 0 0		左移一位
1 1	0 0 1 0		－|Y|
0 0	1 1 0 1 0 1 0 0	1	够减，商为1

例

若被除数X = -0.10001011，除数Y = 0.1110，试利用原码恢复余数法求商及余数。

解：该例满足

| X | < | Y |，

且 | Y | ≠0

已知$[X]_{原}$ = 1.10001011，

$[Y]_{原}$ = 0.1110。

商符 = 1 ⊕ 0 = 1；

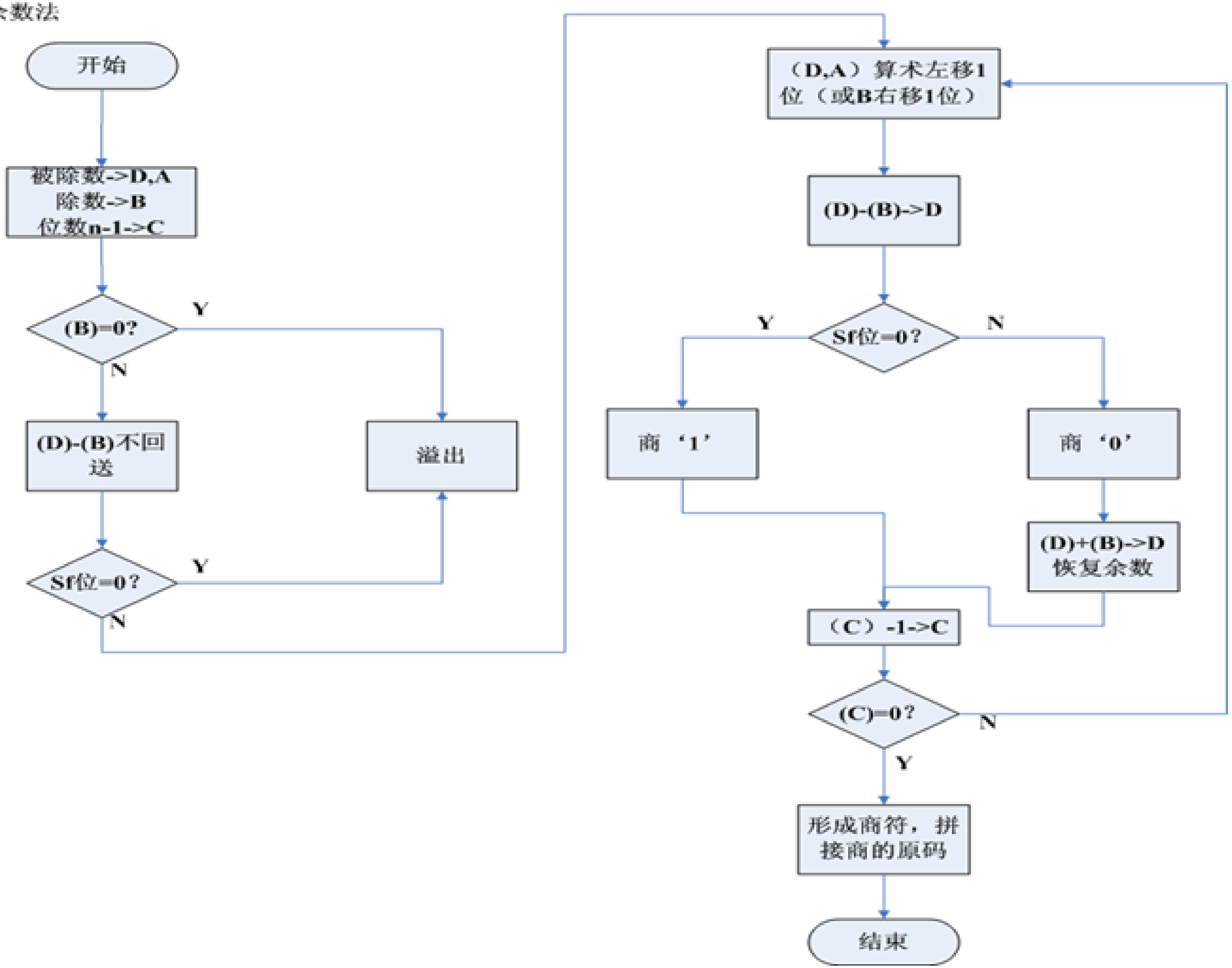
绝对值相除见表。

商 = 1.1001

余数 = 1.1101 × 2^{-4}

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						－｜Y｜
0	0	0	0	1	1	0	1	1	0	1	够减，商为1
0	0	0	1	1	0	1	1	0	1		左移一位
1	1	0	0	1	0						－｜Y｜
1	1	1	0	0	0	1	1	0	1	0	不够减，商为0
0	0	1	1	1	0						+｜Y｜
0	0	0	1	1	0	1	1	0	1	0	恢复余数
0	0	1	1	0	1	1	0	1	0		左移一位
1	1	0	0	1	0						－｜Y｜
1	1	1	1	1	1	1	0	1	0	0	不够减，商为0
0	0	1	1	1	0						+｜Y｜
0	0	1	1	0	1	1	0	1	0	0	恢复余数
0	1	1	0	1	1	0	1	0	0		左移一位
1	1	0	0	1	0						－｜Y｜
0	0	1	1	0	1	0	1	0	0	1	够减，商为1

恢复余数法


开始
被除数->D,A
除数->B
位数n-1->C
(B)=0?
Y
N
(D)-(B)不回送
溢出
Sf位=0？
Y
N
（D,A）算术左移1位（或B右移1位）
(D)-(B)->D
Sf位=0？
Y
N
商 ‘1’
商 ‘0’
(D)+(B)->D
恢复余数
（C）-1->C
(C)=0？
N
Y
形成商符，拼接商的原码
结束

加减交替法

- 假定第i次余数减除数（用B表示）得的余数为R，

当R⩾0时，商1，应左移一位，即 2R，

然后（第i+1次）减去除数，就是 2R －B

当R＜0时，商0，应恢复余数，即 R+B，

再左移一位，即 2(R+B)，

然后（第i+1次）减去除数，就是 2(R+B)－B = 2R +B

- 即当第i次余数减除数的余数R＜0时，不用加除数来恢复余数，而只是将其左移一次，变为2R，到第i+1次运算时将其加除数，也就是2R+B。

- 加减交替法的运算法则

①若余数R≥0，则商上1，左移一次，减除数；

②若余数R＜0，则商上0，左移一次，加除数。

例　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

例　　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$， 商符 $= 1 \oplus 0 = 1$。

例　　若$X = -0.10001011$，$Y = 0.1110$ 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$，商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	

例　　若$X = -0.10001011$，$Y = 0.1110$　试利用原码加减交替法求商及余数。

解：　$[X]_原 = 1.10001011$，$[Y]_原 = 0.1110$，　商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位

例　　若$X = -0.10001011$，$Y = 0.1110$ 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$， 商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\lvert Y\rvert$

例　　若$X = -0.10001011$，$Y = 0.1110$ 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$， 商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\|Y\|$
0	0	0	0	1	1	0	1	1	0	1	R≥0，商为1

例 若$X=-0.10001011$，$Y=0.1110$ 试利用原码加减交替法求商及余数。

解：$[X]_{原}=1.10001011$，$[Y]_{原}=0.1110$， 商符 $=1\oplus 0=1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\lvert Y\rvert$
0	0	0	0	1	1	0	1	1	0	1	$R\geqslant 0$，商为1
0	0	0	1	1	0	1	1	0	1		左移一位

例　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：$[X]_{原}$ = 1.10001011，$[Y]_{原}$ = 0.1110， 商符 = 1 ⊕ 0 = 1。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		$-\|Y\|$
0 0	0 0 1 1 0 1 1 0	1	R≥0，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		$-\|Y\|$

例　　　若$X=-0.10001011$，$Y=0.1110$ 试利用原码加减交替法求商及余数。

解：　$[X]_{原}=1.10001011$，$[Y]_{原}=0.1110$，　商符$=1\oplus 0=1$。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
0 0	0 0 1 1 0 1 1 0	1	R≥0，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
1 1	1 0 0 0 1 1 0 1	0	R<0，商为0

例　　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：$[X]_原 = 1.10001011$，$[Y]_原 = 0.1110$，商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						$-\lvert Y \rvert$
0	0	0	0	1	1	0	1	1	0	1	R≥0，商为1
0	0	0	1	1	0	1	1	0	1		左移一位
1	1	0	0	1	0						$-\lvert Y \rvert$
1	1	1	0	0	0	1	1	0	1	0	R<0，商为0
1	1	0	0	0	1	1	0	1	0		左移一位

例　　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$，商符 $= 1 \oplus 0 = 1$。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
0 0	0 0 1 1 0 1 1 0	1	R≥0，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		$-\lvert Y\rvert$
1 1	1 0 0 0 1 1 0 1	0	R<0，商为0
1 1	0 0 0 1 1 0 1 0		左移一位
0 0	1 1 1 0		$+\lvert Y\rvert$

例　　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$，商符 = 1 ⊕ 0 = 1。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						−|Y|
0	0	0	0	1	1	0	1	1	0	1	R≥0，商为1
0	0	0	1	1	0	1	1	0	1		左移一位
1	1	0	0	1	0						−|Y|
1	1	1	0	0	0	1	1	0	1	0	R<0，商为0
1	1	0	0	0	1	1	0	1	0		左移一位
0	0	1	1	1	0						+|Y|
1	1	1	1	1	1	1	0	1	0	0	R<0，商为0

例　　若$X = -0.10001011$，$Y = 0.1110$ 试利用原码加减交替法求商及余数。

解：$[X]_原 = 1.10001011$，$[Y]_原 = 0.1110$，商符 $= 1 \oplus 0 = 1$。

符号		被除数（余数）								商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						－|Y|
0	0	0	0	1	1	0	1	1	0	1	R≥0，商为1
0	0	0	1	1	0	1	1	0	1		左移一位
1	1	0	0	1	0						－|Y|
1	1	1	0	0	0	1	1	0	1	0	R＜0，商为0
1	1	0	0	0	1	1	0	1	0		左移一位
0	0	1	1	1	0						＋|Y|
1	1	1	1	1	1	1	0	1	0	0	R＜0，商为0
1	1	1	1	1	1	0	1	0	0		左移一位

例　　若X = -0.10001011，Y = 0.1110 试利用原码加减交替法求商及余数。

解：　$[X]_{原}$ = 1.10001011，$[Y]_{原}$ = 0.1110，　商符 = 1 ⊕ 0 = 1。

符号	被除数（余数）	商	操作
0 0	1 0 0 0 1 0 1 1	0	
0 1	0 0 0 1 0 1 1 0		左移一位
1 1	0 0 1 0		－｜Y｜
0 0	0 0 1 1 0 1 1 0	1	R≥0，商为1
0 0	0 1 1 0 1 1 0 1		左移一位
1 1	0 0 1 0		－｜Y｜
1 1	1 0 0 0 1 1 0 1	0	R＜0，商为0
1 1	0 0 0 1 1 0 1 0		左移一位
0 0	1 1 1 0		＋｜Y｜
1 1	1 1 1 1 1 0 1 0	0	R＜0，商为0
1 1	1 1 1 1 0 1 0 0		左移一位
0 0	1 1 1 0		＋｜Y｜

例　　若X = -0.10001011，Y = 0.1110　试利用原码加减交替法求商及余数。

解：　$[X]_{原} = 1.10001011$，$[Y]_{原} = 0.1110$，　商符 = $1 \oplus 0 = 1$。

符号	被除数（余数）	商	操作
0 0 0 1 1 1	1 0 0 0 1 0 1 1 0 0 0 1 0 1 1 0 0 0 1 0	0	 左移一位 －|Y|
0 0 0 0 1 1	0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 0 0 1 0	1	$R \geq 0$，商为1 左移一位 －|Y|
1 1 1 1 0 0	1 0 0 0 1 1 0 1 0 0 0 1 1 0 1 0 1 1 1 0	0	$R < 0$，商为0 左移一位 ＋|Y|
1 1 1 1 0 0	1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 1 1 1 0	0	$R < 0$，商为0 左移一位 ＋|Y|
0 0	1 1 0 1 0 1 0 0	1	$R \geq 0$，商为1

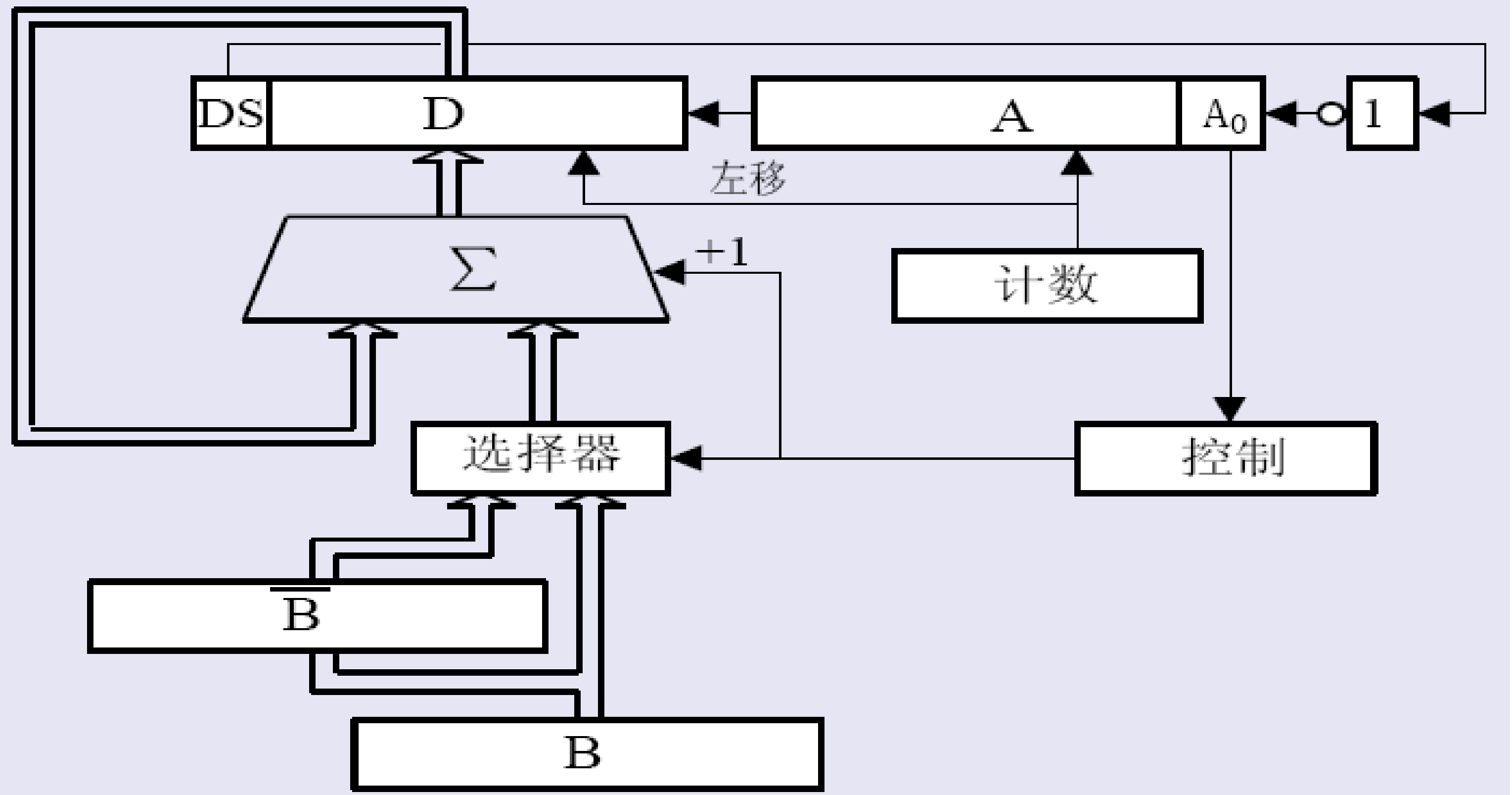
商 = 1.1001　余数 = ?

问题：

当加减交替法运算结束时，如果末位商为0，这时的余数是错误的（负数）。

如何获得正确的余数？

● 加减交替法除法器



加减交替法除法器框图

2. 补码除法运算（课后学习）

补码加减交替除法运算规则：（推导过程略）

(1) 被除数$[X]_{补}$与除数$[Y]_{补}$同号，商为正，做减运算，若余数$[R_1]_{补}$与$[Y]_{补}$同号，则溢出；

被除数$[X]_{补}$与除数$[Y]_{补}$异号，商为负，做加运算，若余数$[R_1]_{补}$与$[Y]_{补}$异号，则溢出。

(2) 若余数$[R_i]_{补}$与除数$[Y]_{补}$同号，上商‘1’，左移一位，减除数；

若余数$[R_i]_{补}$与除数$[Y]_{补}$异号，上商‘0’，左移一位，加除数。

(3) 重复步骤(2)，连同符号位在内，共做n-1次(n 为字长)，末位采用恒置‘1’法。

(4) 商的符号位与数值位均在运算中产生。

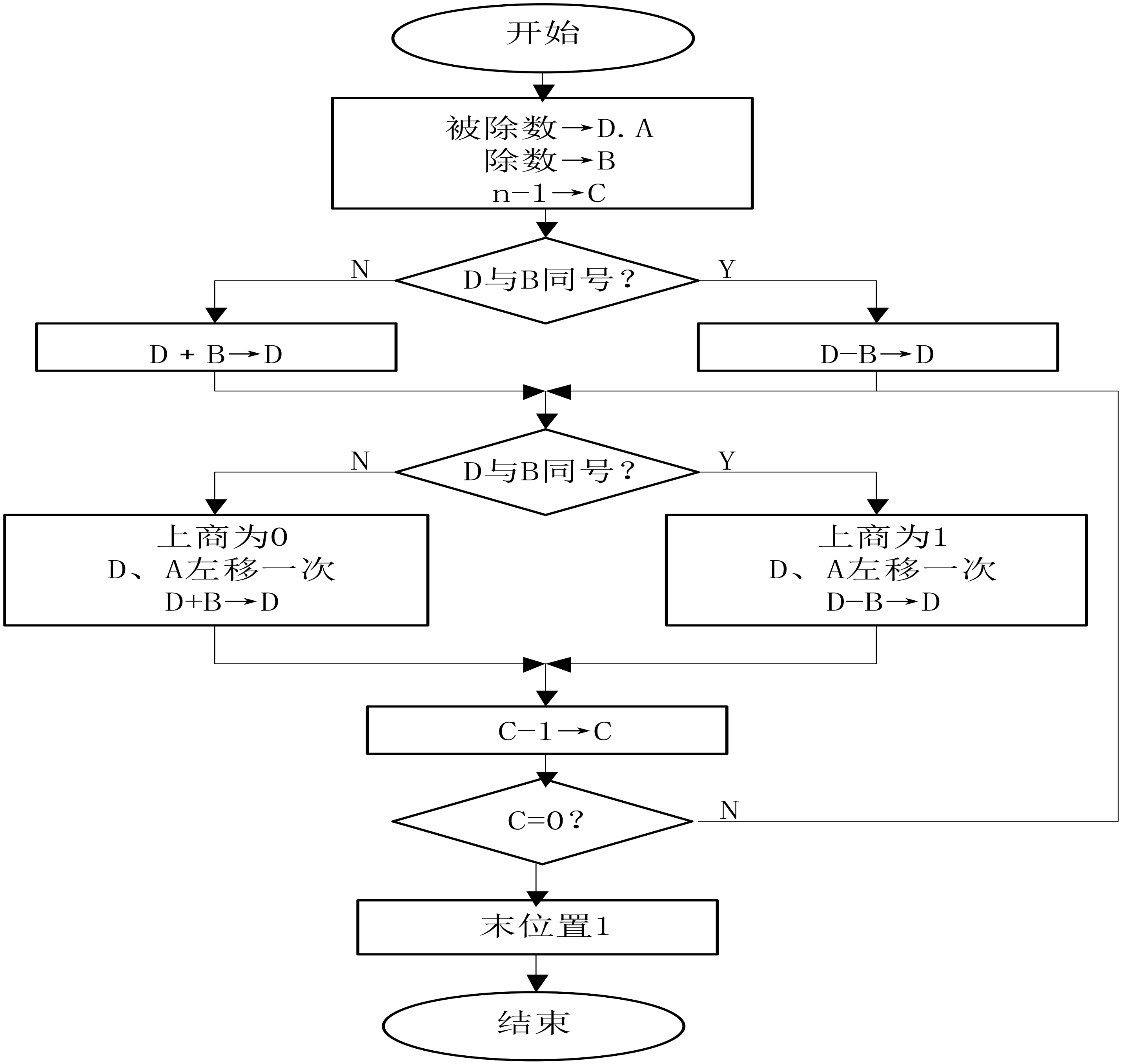
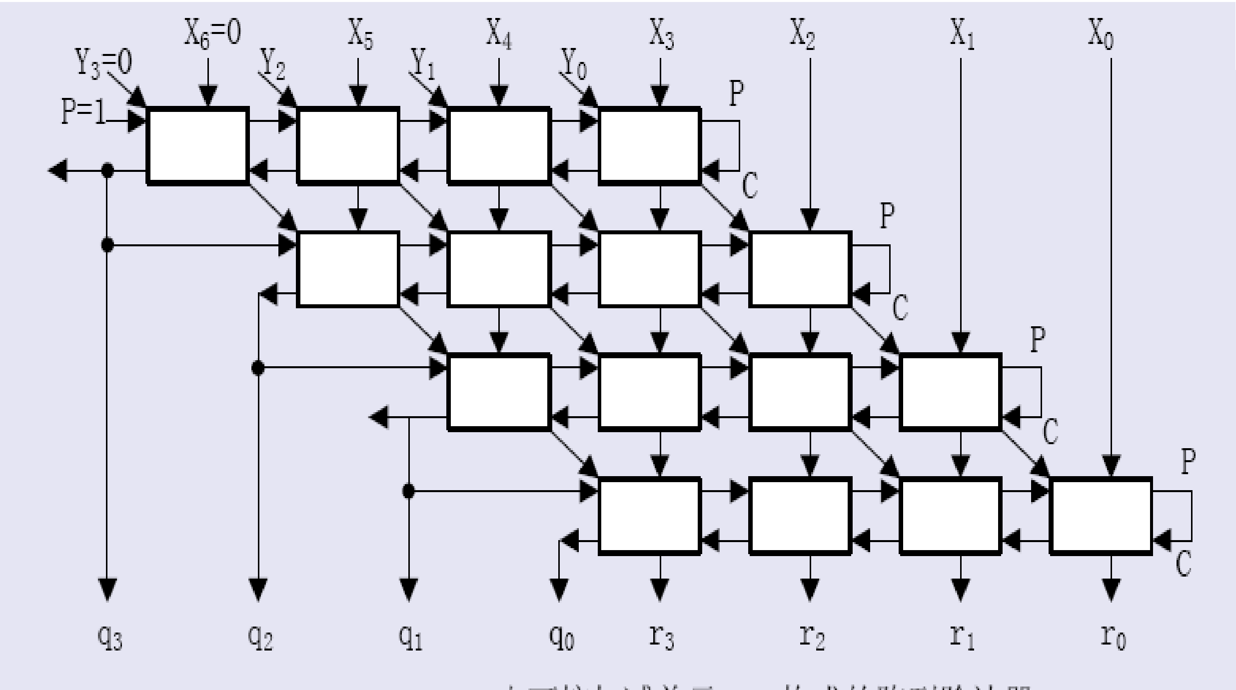
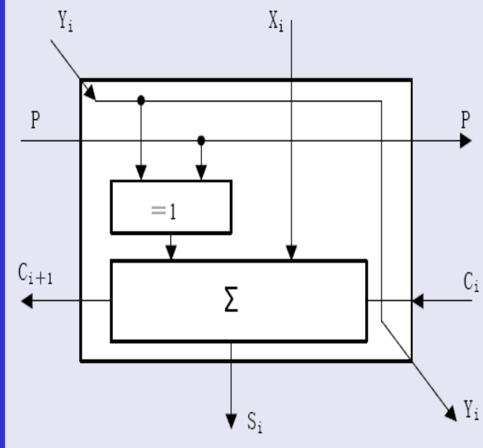


图 3-20　补码除法流程框图

阵列除法器（课后学习）



由可控加减单元CAS构成的阵列除法器



可控加减单元CAS

本章作业3-2（第4次作业）

第 20（1）题

第 22（1）题 （只用原码加减交替法）

注：与上次作业3-1一起交

本章作业3-1

第 17（1）（2）、18（2）、19（2）题

注：本次作业与下次作业3-2一起交