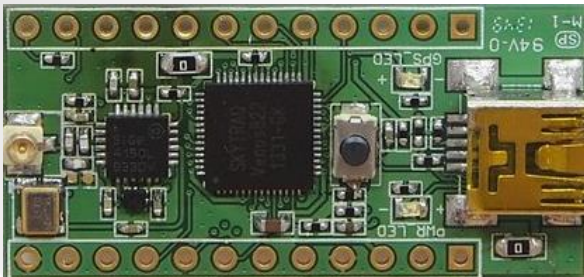


计算机组成原理

第三章 运算方法与运算器

3.6 定点数除法



1

手工除法运算方法

$$\begin{array}{r} 0.1101 \\ 0.1011 \overline{) 0.10010} \\ \underline{- 0.01011} \\ 0.001110 \\ \underline{- 0.001011} \\ 0.0000110 \\ \underline{0.0001011} \\ 0.00001100 \\ \underline{- 0.00001011} \\ 0.00000001 \end{array}$$

不够减，商上零，
除数右移1位，够减，减除数，商上1

除数右移2位，够减，减除数，商上1

除数右移3位，不够减，不减除数，商上零

除数右移4位，够减，减除数，商上1

启示：除法可通过减法实现

问题：除数移位次数不固定且多

需要长度为 $2n$ 位的余数寄存器

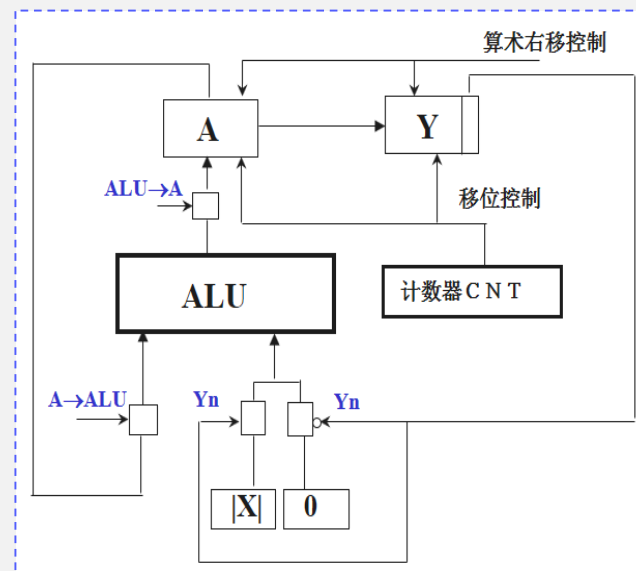
如何判断每步是否够减

原码恢复余数除法

■ 如何判断是否够减

◆ 利用**减法**，通过余数符号判断

$$\begin{array}{r}
 00.10010 \\
 - 00.01011 \\
 \hline
 00.00111
 \end{array}
 \qquad
 \begin{array}{r}
 00.10010 \\
 -00.11011 \\
 \hline
 11.10111 \\
 +00.11011 \\
 \hline
 00.10010
 \end{array}$$

■ 余数为正数时，够减，商上1，将余数**左移**一位，再与除数做减法比较■ 余数为负数时，不够减，商上0，**?**◆ 加除数恢复成原来的值，将余数**左移**一位，再与除数做减法比较

■ 重复上述过程直到商达到所需要的位数为止。

2

原码恢复余数除法

已知 $X = 0.1001$, $Y = - 0.1011$, 用原码一位除法求 X/Y

解 : $[X]_{\text{原}} = 0.1001$ $[|X|]_{\text{补}} = 0.1001$

$[Y]_{\text{原}} = 1.1011$ $[|Y|]_{\text{补}} = 0.1011$ $[-|Y|]_{\text{补}} = 1.0101$

2

原码除法运算方法

被除数/余数	商	上商位	说明
00.1001			减Y比较
+[-Y] _补 11.0101			余数<0, 商=0
11.1110		0	加Y恢复余数
+ 00.1011			
00.1001			左移一位
01.0010	0		减Y比较
+[-Y] _补 11.0101			余数>0, 商上1
00.0111		1	左移一位
00.1110	0.1		减Y比较
+[-Y] _补 11.0101			余数>0, 商上1
00.0011		1	左移一位
00.0110	0.11		减Y比较
+[-Y] _补 11.0101			余数<0, 商上0
11.1011		0	加Y恢复余数
+ 00.1011			
00.0110			左移一位
00.1100	0.110		减Y比较
+[-Y] _补 11.0101			余数>0, 商上1, 移商
00.0001	0.1101	1	

最后结果：
商Q = (X₀ ⊕ Y₀) .1101=1.1101
余数 R = 0.0001 * 2⁻⁴

该方法存在的不足：
运算步数不确定

3

原码加/减交替除法运算方法（不恢复余数法）

- 设某次余数为 R_i ，将 R_i 左移一位减除数进行比较并上商，即：

$$2R_i - Y$$

- 当上述结果小于0时，商上0，恢复余数，然后左移一位，减除数比较，即：

$$(2R_i - Y) + Y = 2R_i$$

$$2 * 2R_i - Y = 4R_i - Y$$

- 若当结果小于0时，商上0，不恢复余数而直接将余数左移一位，加Y:

$$2(2R_i - Y) + Y$$

$$= 2 * 2R_i - 2Y + Y = 4R_i - Y$$

3

原码加/减交替除法运算方法（不恢复余数法）

已知 $X = 0.1001$, $Y = - 0.1011$, 用原码一位除法求 X/Y

被除数/余数	商	上商位	说明
$+[-Y]_{补}$ 00.1001 11.0101			减Y比较
0 11.1110		0	余数 <0 商上零
11.1100	0		左移一位
$+[Y]_{补}$ 00.1011			加Y比较
1 00.0111		1	余数>0, 商上1
00.1110	0.1		左移一位
$+[-Y]_{补}$ 11.0101			减Y比较
1 00.0011		1	余数>0, 商上1
00.0110	0.11		左移一位
$+[-Y]_{补}$ 11.0101			减Y比较
0 11.1011		0	余数<0 商上零
11.0110	0.110		左移一位
$+[Y]_{补}$ 00.1011			加Y比较
1 00.0001	0.1101	1	余数>0, 商上1,移商

最后结果：

商 $Q = X_0 \oplus Y_0.1101=1.1101$

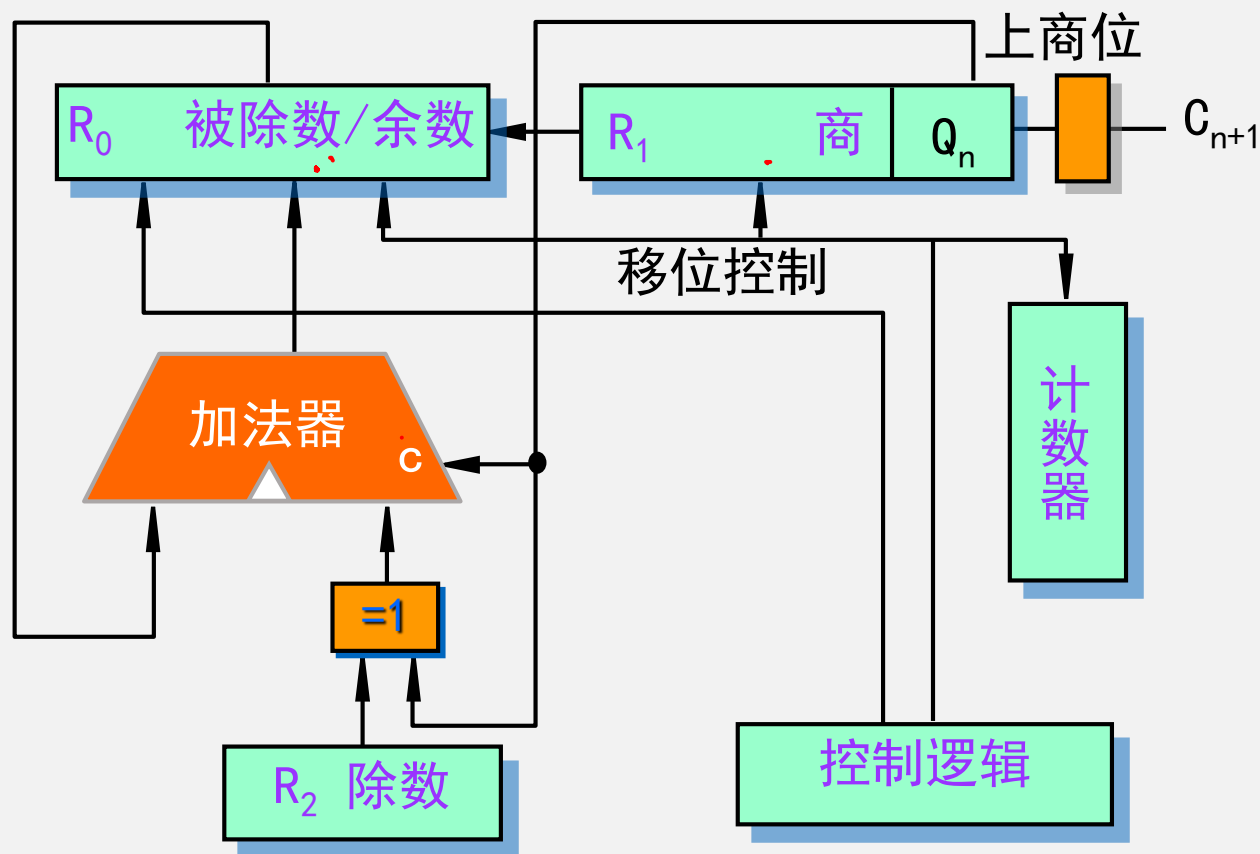
余数 $R = 0.0001 * 2^{-4}$

被除数/余数	商	上商位	说明
$+[-Y]_{补}$ 00.1001 11.0101			减Y比较
11.1110		0	余数<0, 商=0
+ 00.1011			加Y恢复余数
00.1001			左移一位
01.0010	0		减Y比较
$+[-Y]_{补}$ 11.0101			减Y比较
00.0111		1	余数>0, 商上1
00.1110	0.1		左移一位
$+[-Y]_{补}$ 11.0101			减Y比较
00.0011		1	余数>0, 商上1
00.0110	0.11		左移一位
$+[-Y]_{补}$ 11.0101			减Y比较
11.1011		0	余数<0, 商上0
+ 00.1011			加Y恢复余数
00.0110			左移一位
00.1100	0.110		减Y比较
$+[-Y]_{补}$ 11.0101			减Y比较
00.0001	0.1101	1	余数>0, 商上1, 移商

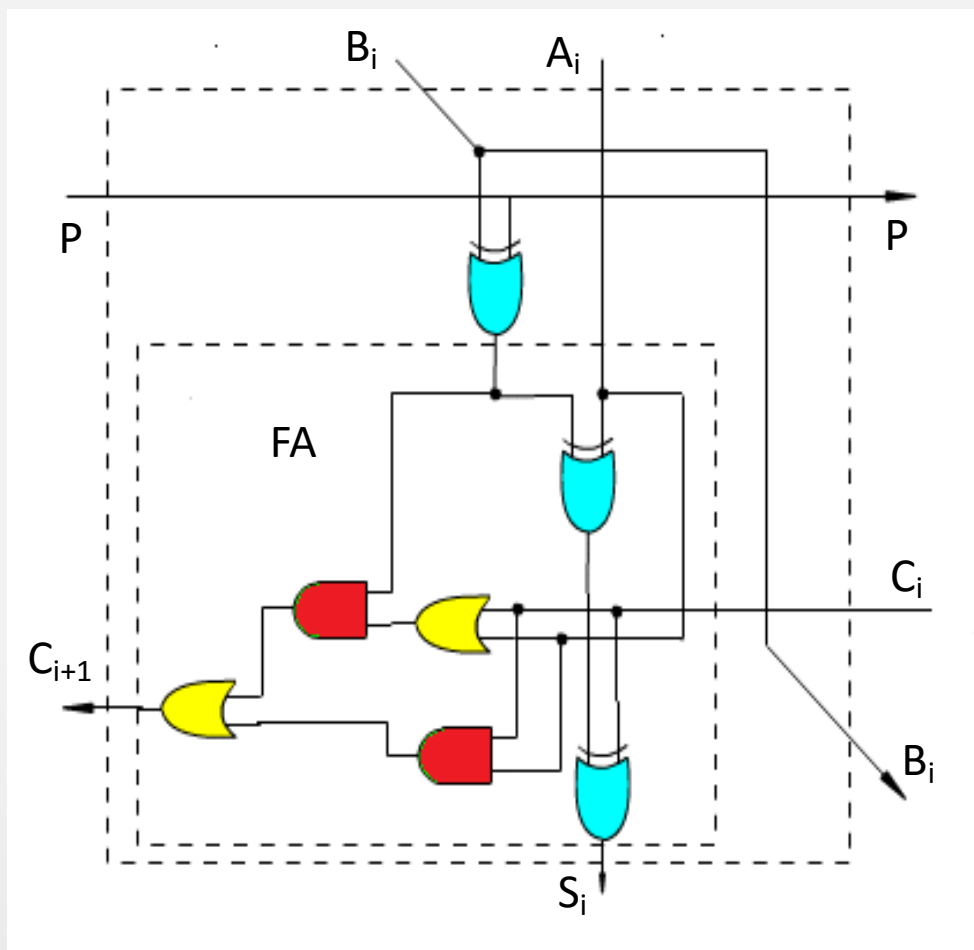
4

原码加/减交替除法实现逻辑

被除数/余数	商	上商位	说明
$+[-Y]_{\text{补}}$ 00.1001 11.0101			减Y比较
0 11.1110		0	余数<0 商上零
11.1100	0		左移一位
$+ [Y]_{\text{补}}$ 00.1011			加Y比较
1 00.0111		1	余数>0, 商上1
00.1110	0.1		左移一位
$+ [-Y]_{\text{补}}$ 11.0101			减Y比较
1 00.0011		1	余数>0, 商上1
00.0110	0.11		左移一位
$+ [-Y]_{\text{补}}$ 11.0101			减Y比较
0 11.1011		0	余数<0 商上零
11.0110	0.110		左移一位
$+ [Y]_{\text{补}}$ 00.1011			加Y比较
1 00.0001	0.1101	1	余数>0, 商上1, 移商



1) 可控制加/减法(CAS)单元



逻辑功能为：

$$S_i = A_i \oplus (B_i \oplus P) \oplus C_i$$

$$C_{i+1} = (A_i + C_i) (B_i \oplus P) + A_i C_i$$

P=0时实现加法功能

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_{i+1} = (A_i + C_i) B_i + A_i C_i$$

P=1时实现减法功能(全减)

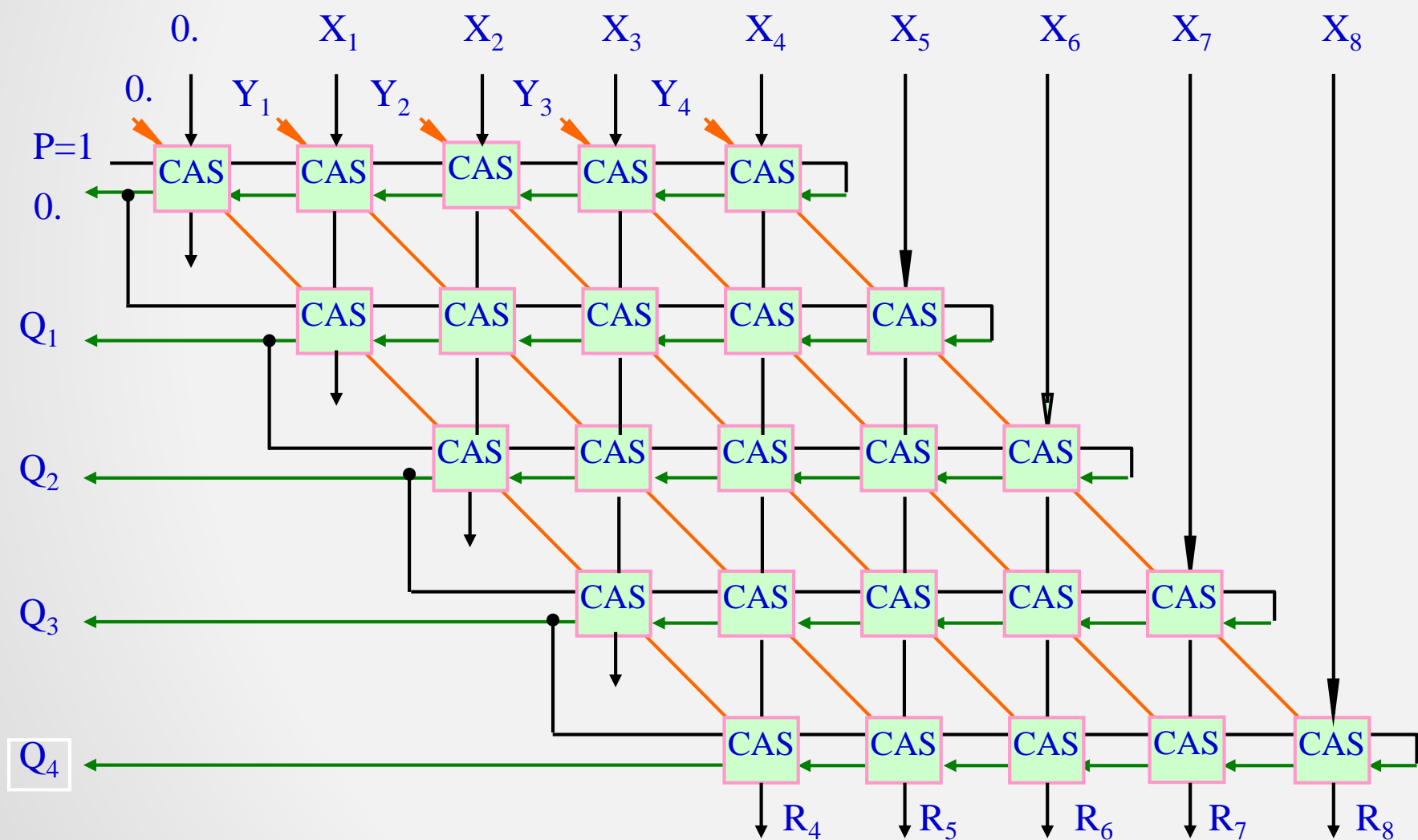
$$S_i = A_i \oplus \bar{B}_i \oplus C_i$$

$$C_{i+1} = (A_i + C_i) \bar{B}_i + A_i C_i$$

5

阵列除法

2) 基于CAS的阵列除法



- 注意连接、输入输出关系
- 使用原码不恢复余数法。第一步一定是减法，故 $P=1$ ，以后各步做加还是减取决于前一步的商
- 最左边CAS的进位输出是商，且本位商决定下一步是执行加操作还是减操作
- 每执行完一步除法，就将除数右移一位(同手工除法)