定点数乘法的实现原理

本节内容

- 乘法器的实现原理
  - 原码乘法
  - 补码乘法

一位乘法电路实现

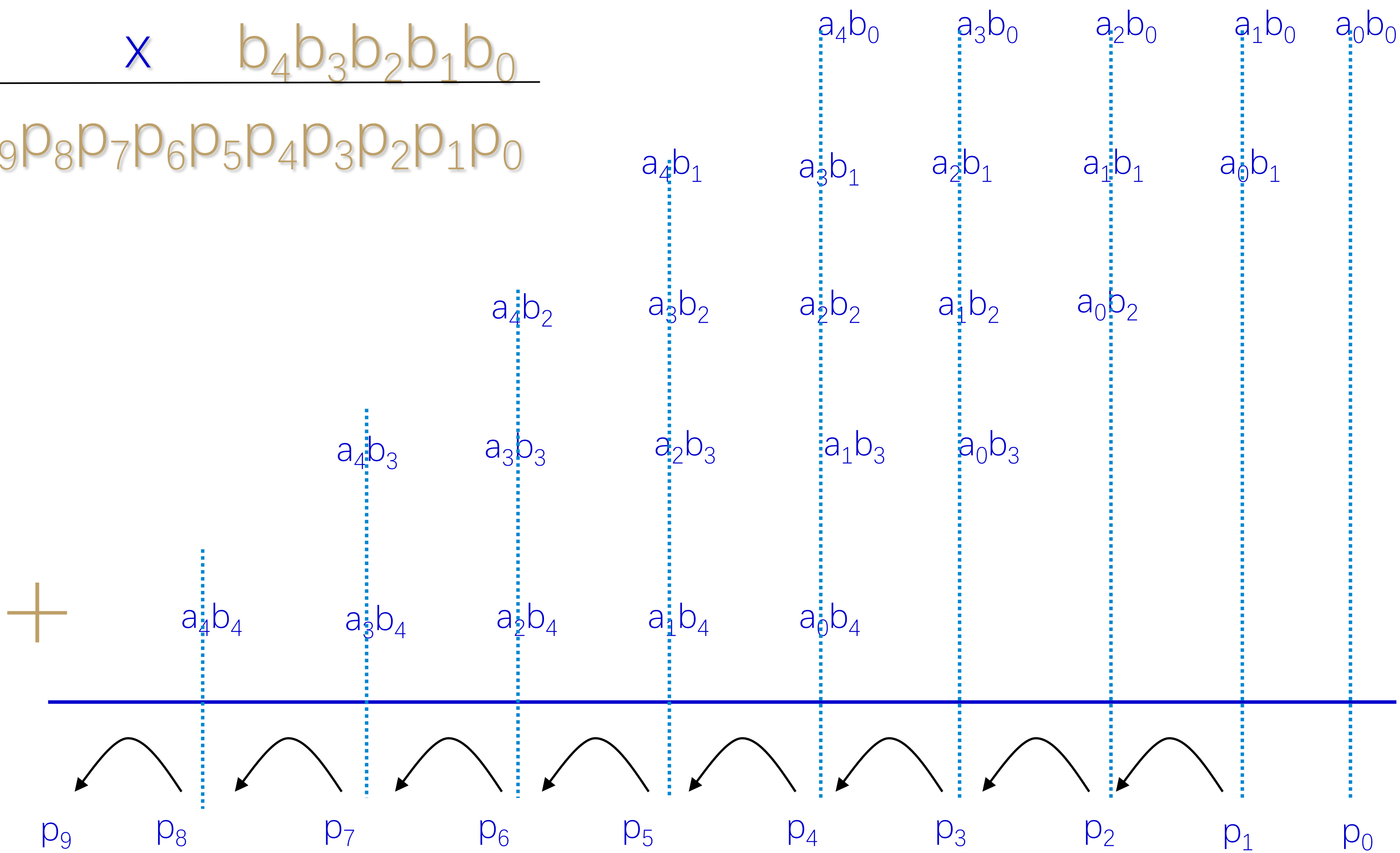
X_i	Y_i	Output
0	0	0
0	1	0
1	0	0
1	1	1

$$Output = X_i \bullet Y_i$$

一个与门
就可以实现一位乘法

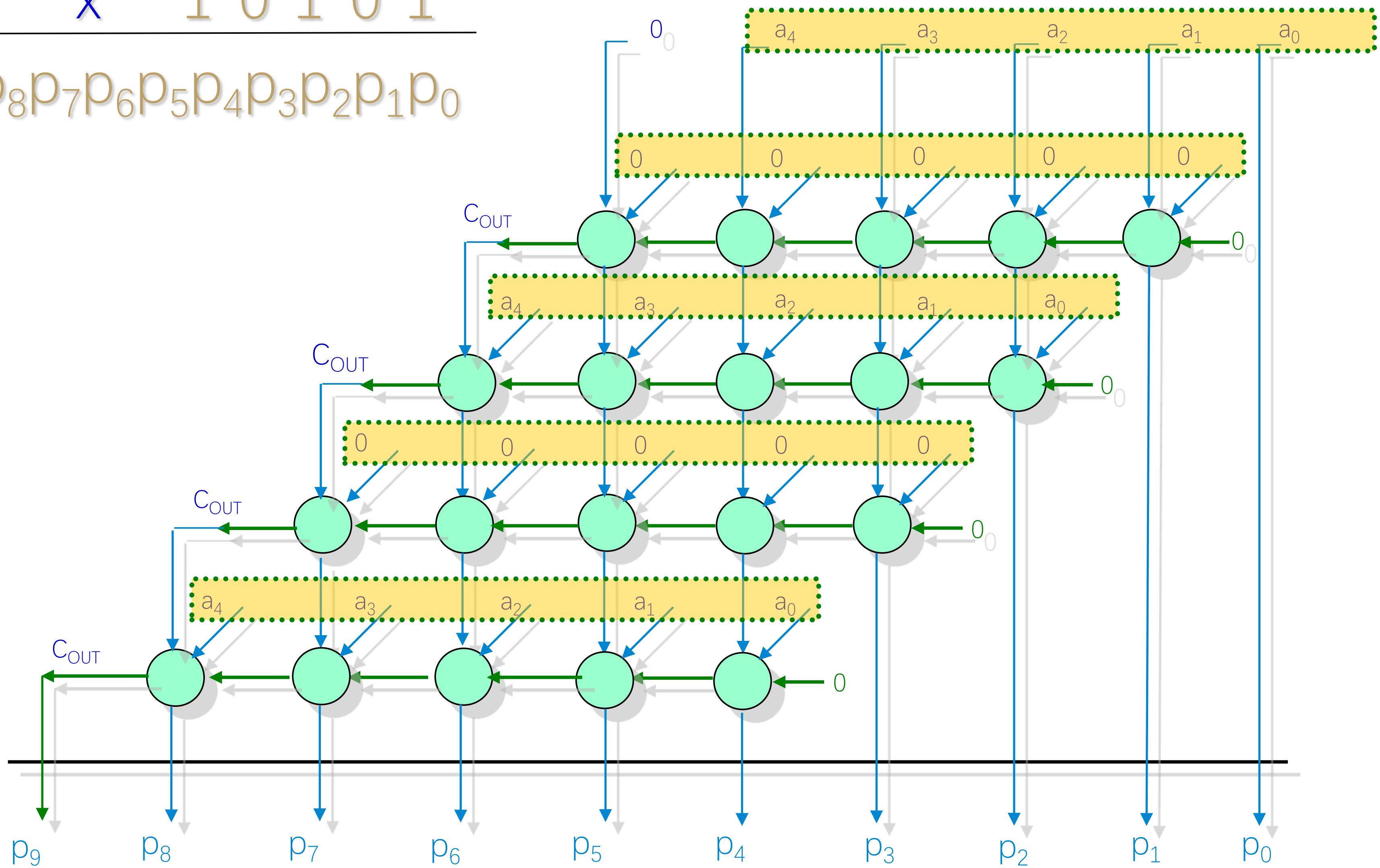
多位乘法

$$
\begin{array}{r}
a_4a_3a_2a_1a_0 \\
\times \quad b_4b_3b_2b_1b_0 \\
\hline
p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0
\end{array}
$$

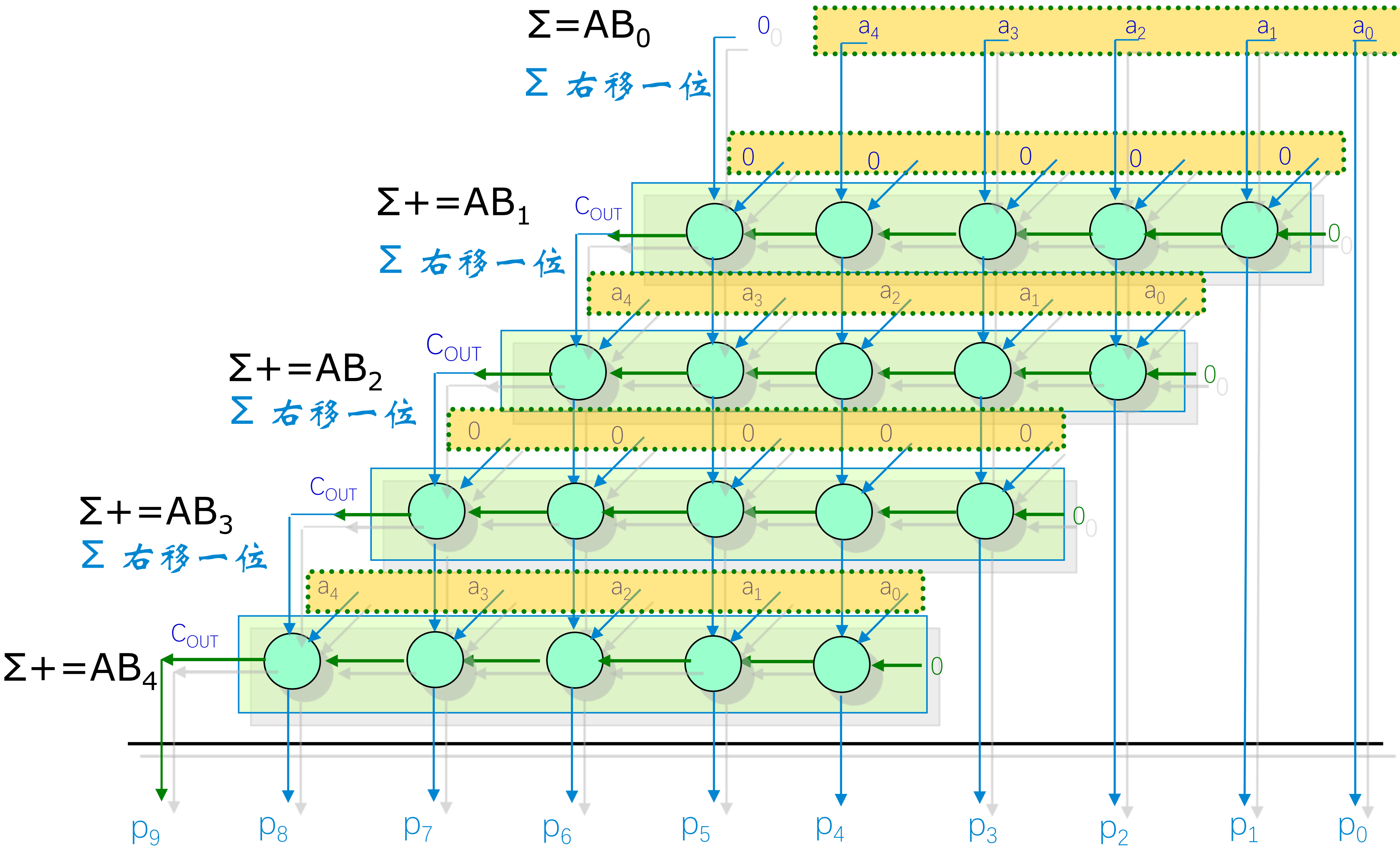


多位乘法必须先计算一位乘法，然后累加求和

$$\begin{array}{r} a_4a_3a_2a_1a_0 \\ \times \quad 1\ 0\ 1\ 0\ 1 \\ \hline p_9p_8p_7p_6p_5p_4p_3p_2p_1p_0 \end{array}$$

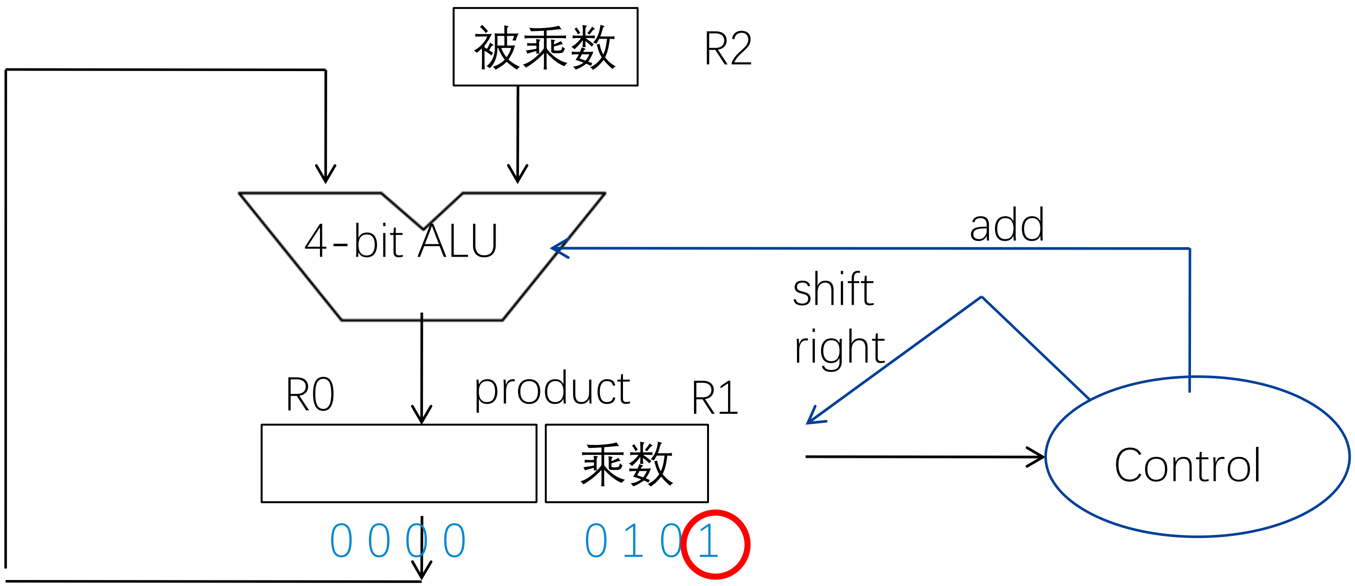


5位无符号数阵列乘法器电路



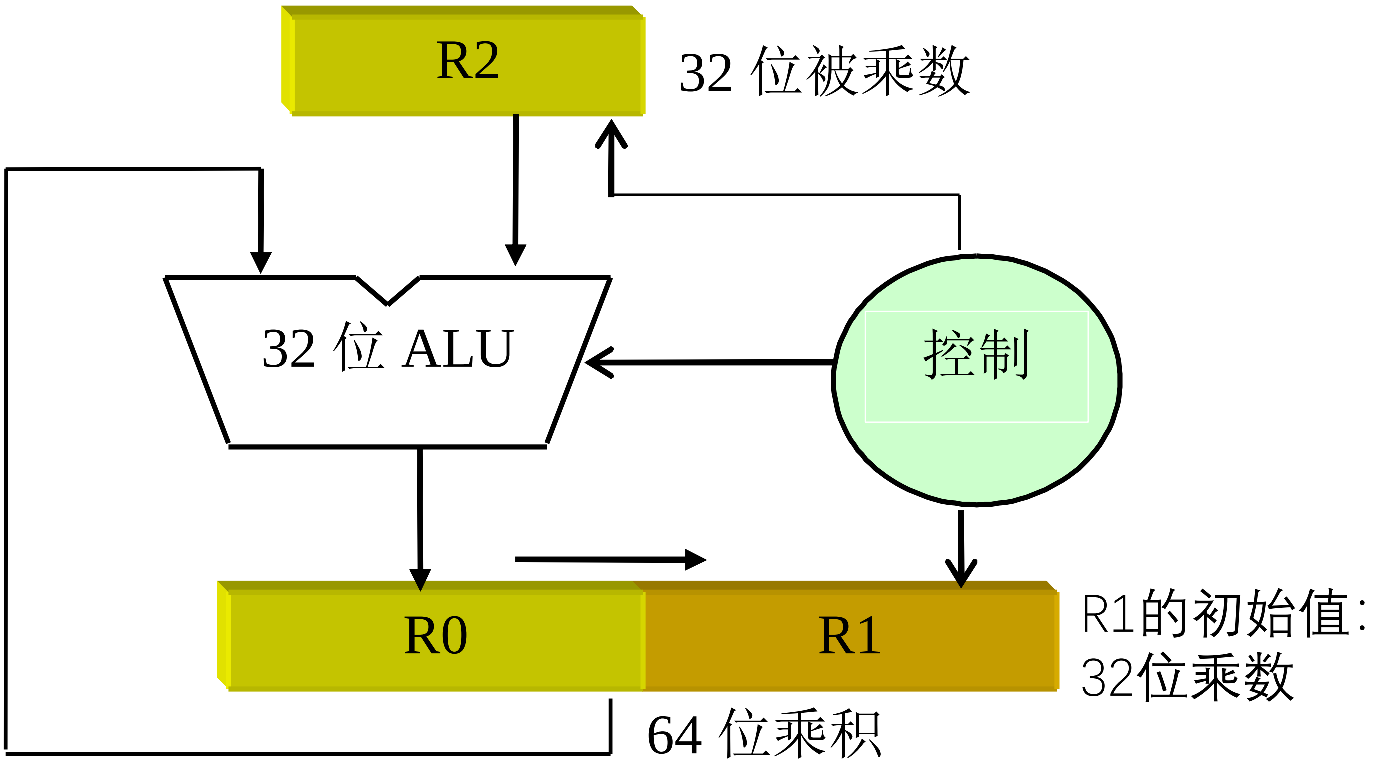
5位无符号数阵列乘法器电路

0 1 1 0 = 6



	0 0 0 0	0 1 0 1
add	0 1 1 0	
加法结果	0 1 1 0	0 1 0 1
右移后	0 0 1 1	0 0 1 0
add	0 0 0 0	
加法结果	0 0 1 1	0 0 1 0
右移后	0 0 0 1	1 0 0 1
add	0 1 1 0	
加法结果	0 1 1 1	1 0 0 1
右移后	0 0 1 1	1 1 0 0
add	0 0 0 0	
加法结果	0 0 1 1	1 1 0 0
右移后	0 0 0 1	1 1 1 0 = 30

用加法器实现乘法电路



32 位原码（或：无符号数）一位乘法的方案

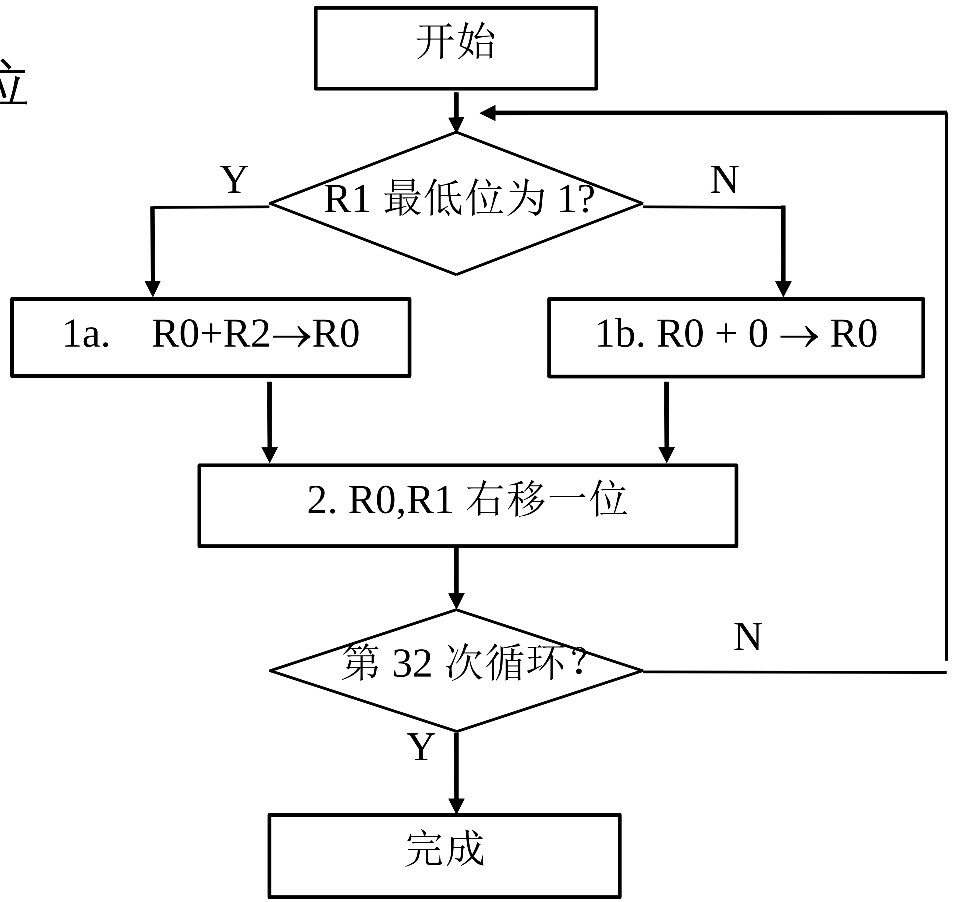
乘法流程

寄存器长度：32位

R0： 初始值为0

R1: 乘数

R2： 被乘数



实现 32 位定点原码一位乘法的流程图

用加法器实现乘法

- 模拟手算乘法的过程
- 每次循环结束前，对加法结果的右移，等价于手算过程的一次按位乘法的结果与已有结果的向左错一位的加法
- 特点：
  - 简单
  - 速度慢： 32次加法实现一次乘法

补码乘法

- 不存在补码乘法公式
- 例: (-2)×3

```
            1 1 1 0
      x   0 0 1 1
     ----------------
          1 1 1 0
        1 1 1 0
      0 0 0 0
    0 0 0 0          0
   ------------------
   0 0 1 0 1 0 1 0
```

补码乘法

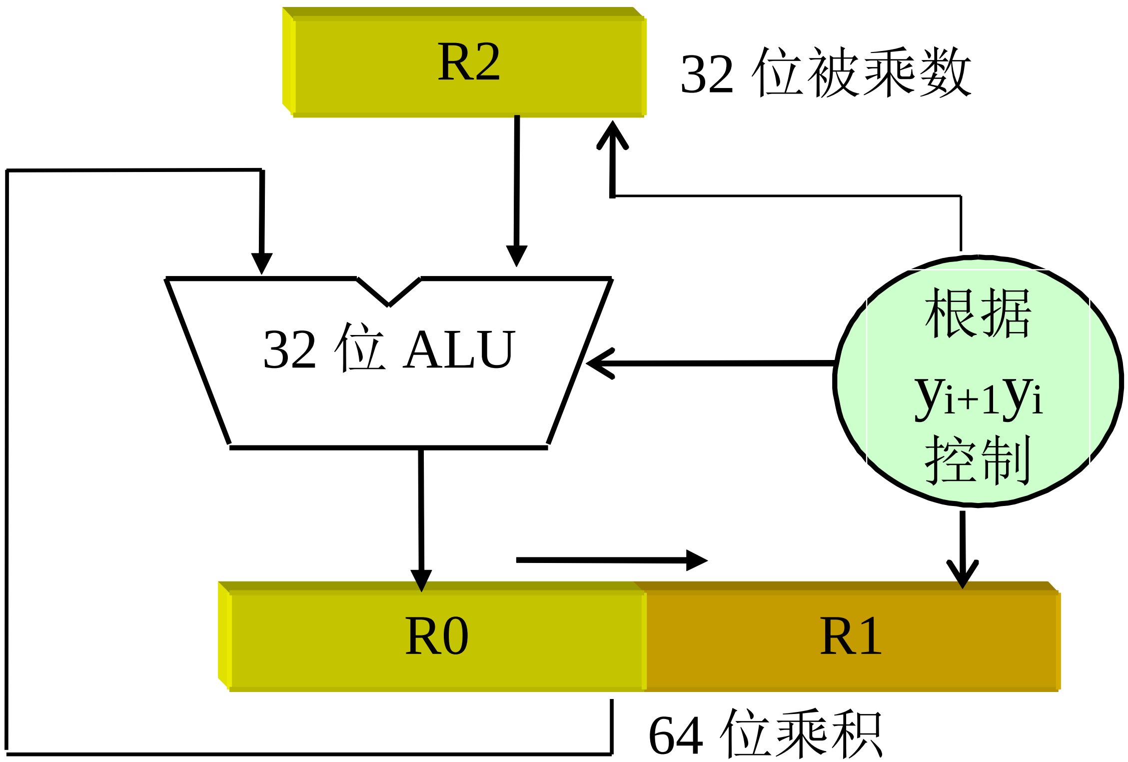
- 用原码乘法器计算补码乘法
  - 补码转换为原码
  - 运算
  - 运算结果转换为补码
- 用Booth算法计算补码乘法
  - 原理:将乘法转换为对2^n的加法
  - 如:6x=8x-2x
  - 即:0110x = 1000x–0010x

Booth 算法

- 假设x为被乘数，y为被乘数
- $[x]_{补}=x_n\,x_{n-1}\,x_{n-2}\cdots\cdots x_0$， $[y]_{补}=y_n\,y_{n-1}\,y_{n-2}\cdots\cdots y_0$
- 其中， $[x]_{补}$的符号位是x_n，$[y]_{补}$的符号位是y_n

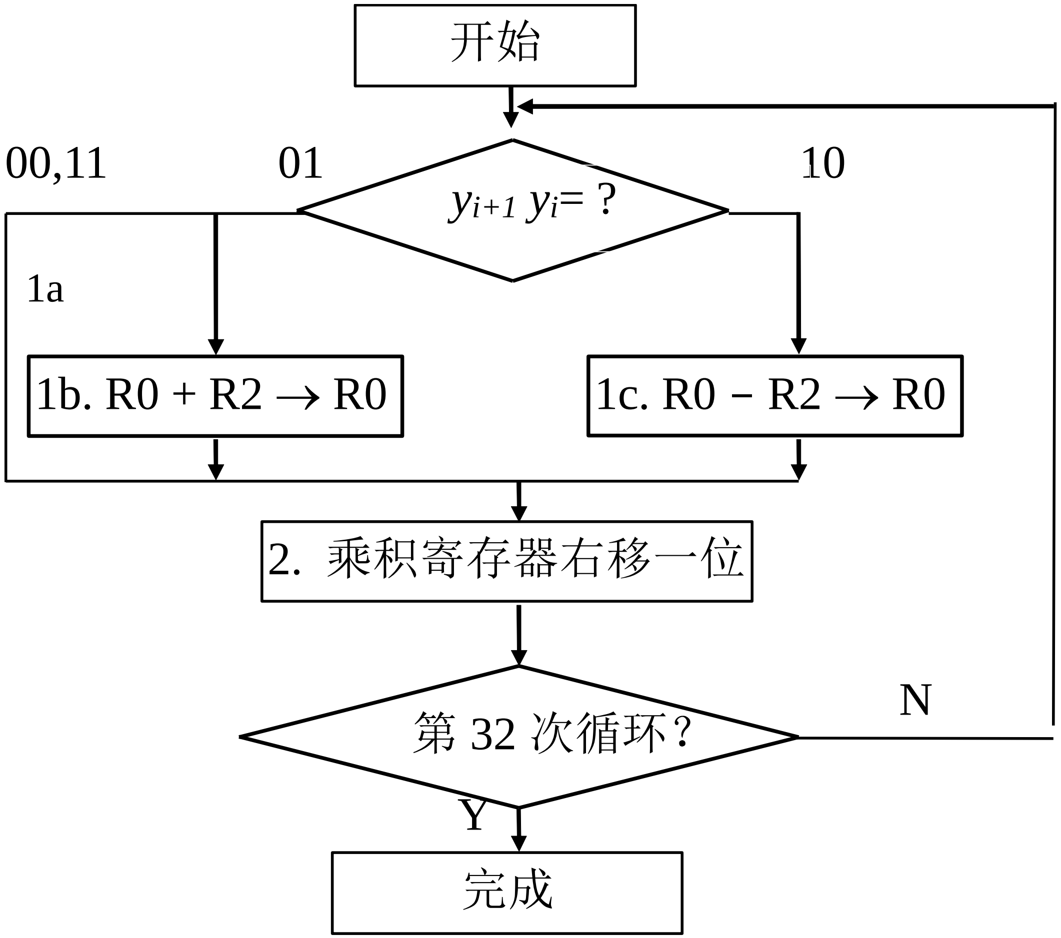
y_{i+1}	y_i	操作
0	0	无
0	1	加x
1	0	减x
1	1	无

Booth算法补码乘法逻辑结构



根据 R1 最后两位 $y_{i+1}y_i$ 的值，进行不同操作

Booth算法 32位乘法流程



Booth 算法的原理

- 假设x为被乘数，y为被乘数
- $[x]_补 = x_n\, x_{n-1}\, x_{n-2} \cdots\cdots x_0$， $[y]_补 = y_n\, y_{n-1}\, y_{n-2} \cdots\cdots y_0$
- 其中， $[x]_补$的符号位是x_n , $[y]_补$的符号位是y_n , n=31
- Booth 算法所计算的结果可表示为：

$$x \times (0-y_0)\times 2^0 + x \times (y_0-y_1)\times 2^1 + x \times (y_1-y_2)\times 2^2 + \cdots + x \times (y_{30}-y_{31})\times 2^{31}$$

$$= x \times (-y_{31}\times 2^{31} + y_{30}\times 2^{30} + y_{29}\times 2^{29} + \cdots + y_1\times 2^1 + y_0\times 2^0)$$

- y_{31}为0， y是正数； y_{31}为1， y是负数，
- （ $-y_{31}\times 2^{31} + y_{30}\times 2^{30} + y_{29}\times 2^{29} + \cdots + y_1\times 2^1 + y_0\times 2^0$ ） 是y的补码所对应的y的真值

用Booth补码一位乘法计算 2×(-3)的过程

循环	步骤	乘积(R0R1)
0	初始值	0000 1101 0
1	减0010	1110 1101 0
	右移1位	1111 0110 1
2	加0010	0001 0110 1
	右移1位	0000 1011 0
3	减0010	1110 1011 0
	右移1位	1111 0101 1
4	无操作	1111 0101 1
	右移1位	1111 1010 1

小结

- 乘法器的实现原理
- 无符号数乘法
- 有符号数乘法