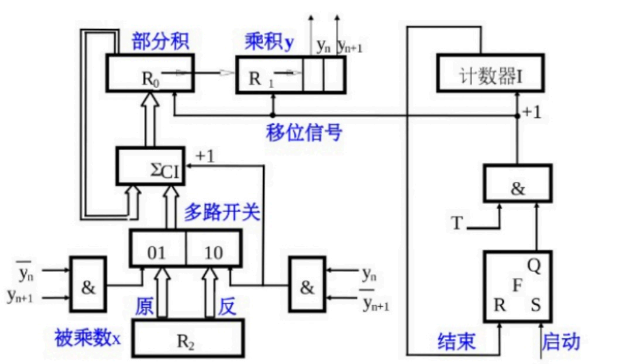
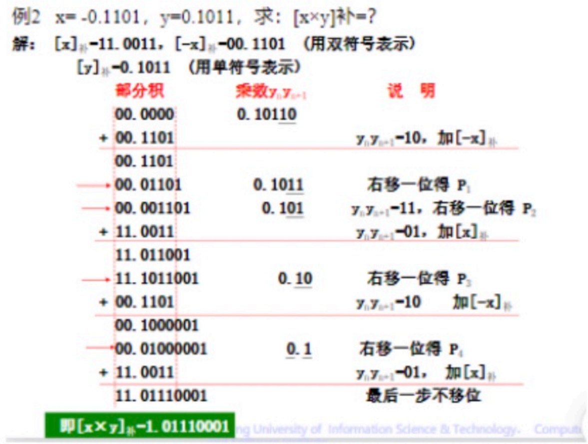
一、如下图所示，用例 2 解释图 2.8 的工作原理，特别叙述 R0、R1、R2 最初装的是什么、最后装的是什么，计数器计算了多少次，移了多少次?





例2 x= -0.1101，y=0.1011，求：[x×y]补=?

解：$[x]_补=11.0011$，$[-x]_补=00.1101$ （用双符号表示）

$[y]_补=0.1011$ （用单符号表示）

部分积	乘数$y_n y_{n+1}$	说 明
00. 0000	0. 10110	
+ 00. 1101		$y_n y_{n+1}=10$，加$[-x]_补$
00. 1101		
→ 00. 01101	0. 1011	右移一位得 P_1
→ 00. 001101	0. 101	$y_n y_{n+1}=11$，右移一位得 P_2
+ 11. 0011		$y_n y_{n+1}=01$，加$[x]_补$
11. 011001		
→ 11. 1011001	0. 10	右移一位得 P_3
+ 00. 1101		$y_n y_{n+1}=10$ 加$[-x]_补$
00. 1000001		
→ 00. 01000001	0. 1	右移一位得 P_4
+ 11. 0011		$y_n y_{n+1}=01$，加$[x]_补$
11. 01110001		最后一步不移位

即$[x\times y]_补=1.01110001$

工作原理：

① 乘数寄存器末端增设附加位 y_{n+1}，初值为0

② y_n与y_{n+1}的形式决定R_0与R_2的操作

y_n	y_{n+1}	
0	0	部分积右移一位
0	1	部分积加$[x]_补$，右移一位
1	0	部分积加$[-x]_补$，右移一位
1	1	部分积右移一位

③ R_2根据上述条件，从相应形式到加法器与部分积求和

④ 通过移位获取新的 $y_n y_{n+1}$，以此循环n次

初值	结束
$R_0=00.0000$	$R_0=11.0111$
$R_1=0.10110$	$R_1=00101$
$R_2=11.0011$	$R_2=11.0011$

计算了5次，移动了4次

二. 为什么要对阶？

使两数阶码相同来保证运算的正确

三. 对阶的方法？

① 求两数的阶码之差 得出阶码小的一方

② 使小的一方尾数右移，从而使阶码一致

四. 为什么要规格化？是什么时候左规、右规？如何实现？

① 为了保证运算精度

② 当符号位和最高有效为 11.1x…x 或 00.0 需要左规

当符号位为 01 或 10 时，需要右规

③ 左规，尾数连同符号位左移一位，阶码减1

右规，尾数连同符号位右移一位，阶码加1 | 直到变为 $\begin{cases} 11.0 \\ 00.1 \end{cases}$

五. 例：设浮点数的阶码为 4b （含阶符），尾数为 6b （含尾数），x、y 中的指数项，小数项均为二进制真值。

(1) $x=2^{01}\times0.1101$ ，$y=2^{11}\times(-0.1010)$，求 $x+y=?$

(2) $x=-2^{-010}\times0.1111$ ，$y=2^{-100}\times0.1110$，求 $x-y=?$

(1) x 阶码：0001 尾数：0.11010

y 阶码：0011 尾数：1.01100

① 对阶 $\Delta E = [m]_{补} + [-n]_{补} = 0001 + 1101 = 1110$

原 = 1010 = −2

x 尾数右移2位，阶码加2 ⇒ $[X]_{补} = 0011，0.00111$ (0舍1入)

② 尾数相加 (双符号) $[X]_{尾} + [Y]_{尾} = 11.10011$

③ 规格化 $[X+Y]_{补} = 0010，1.00110 \Rightarrow x+y = 2^{010} \times (-0.11010)$

(2) X 阶码：1110 尾数 0.11110

Y 阶码：1100 尾数 0.11100

① 对阶 $\Delta E = [X]_{补} + [-Y]_{补} = 1110 + 0100 = 0010 = 2$

-Y尾数右移2位 $[-Y]_{补} = 1110, 1.11001$

② 尾数相减 $[X_{尾}]_{补} + [Y_{尾}]_{补} = 110.11011$ 舍

③ 规格化 $[X-Y]_{补} = 1111, 11.01110$ (0舍1入)

$X-Y = 2^{-001} \times (-0.10010)$