(5) 能否用快速乘法器实现除法运算？如何实现？

3. 考虑以下C语言程序代码：

```
int func1(unsigned word)
{
    return (int)((word <<24) >> 24);
}
int func2(unsigned word)
{
    return ((int)word <<24) >> 24;
}
```

func 1：先将 unsigned 数 word 逻辑左移24位，再逻辑右移24位，最后再类型转化为int.

func 2：先将 word 转化为int型，再算术左移 24位，再算术右移24位

假设在一个32位机器上执行这些函数，该机器使用二进制补码表示带符号整数。无符号数采用逻辑移位，带符号整数采用算术移位。请填写表6.2，并说明函数 func1 和 func2 的功能。

表6.2 习题3用表

w		func1(w)		func2(w)	
机器数	值	机器数	值	机器数	值
0000 ··· 0111 1111	127	0000 ··· 0111 1111	127	0000 ··· 0111 1111	127
0000 ··· 1000 0000	128	0000 ··· 1000 0000	128	1111 ··· 1000 0000	-128
0000 ··· 1111 1111	255	0000 ··· 1111 1111	255	1111 ··· 1111 1111	-1
0000 ··· 0001 0000 0000	256	0000 ··· 0001 0000 0000	256	0000 ··· 0000 0000	0

4. 填写表6.3，注意对比无符号整数相乘和带符号整数相乘时，在截断操作前（取6位乘积）、截断操作后（取低3位乘积）的结果

4. 填写表 6.3，注意对比无符号整数相乘和带符号整数相乘的[illegible]截断操作后（取低 3 位乘积）的结果。

表 6.3　习题 4 用表

模式	x		y		x×y（截断前）		x×y（截断后）	
	机器数	值	机器数	值	机器数	值	机器数	值
无符号数	110	6	010	2	001100	12	100	4
二进制补码	110	-2	010	2	111100	-4	100	-4
无符号数	001	1	111	7	000111	7	111	7
二进制补码	001	1	111	-1	111111	-1	111	-1
无符号数	111	7	111	7	110001	49	001	1
二进制补码	111	-1	111	-1	110001	-15	001	1

以下是两段 C 语言代码，函数 arith() 是直接用 C 语言写的，而 optarith() 是对函数 arith()

6.
$$\begin{cases} C_1 = A_1B_1 + A_1C_0 + B_1C_0 \\ C_2 = A_2B_2 + A_2C_1 + B_2C_1 \\ C_3 = A_3B_3 + A_3C_2 + B_3C_2 \\ C_4 = A_4B_4 + A_4C_3 + B_4C_3 \end{cases}$$

$$\begin{cases} C_1 = A_1B_1 + (A_1+B_1)C_0 \\ C_2 = A_2B_2 + (A_2+B_2)A_1B_1 + (A_2+B_2)(A_1+B_1)C_0 \\ C_3 = A_3B_3 + (A_3+B_3)A_2B_2 + (A_3+B_3)(A_2+B_2)A_1B_1 + (A_3+B_3)(A_2+B_2)(A_1+B_1)C_0 \\ C_4 = A_4B_4 + (A_4+B_4)A_3B_3 + (A_4+B_4)(A_3+B_3)A_2B_2 + (A_4+B_4)(A_3+B_3)(A_2+B_2)A_1B_1 + (A_4+B_4)(A_3+B_3)(A_2+B_2)(A_1+B_1)C_0 \end{cases}$$

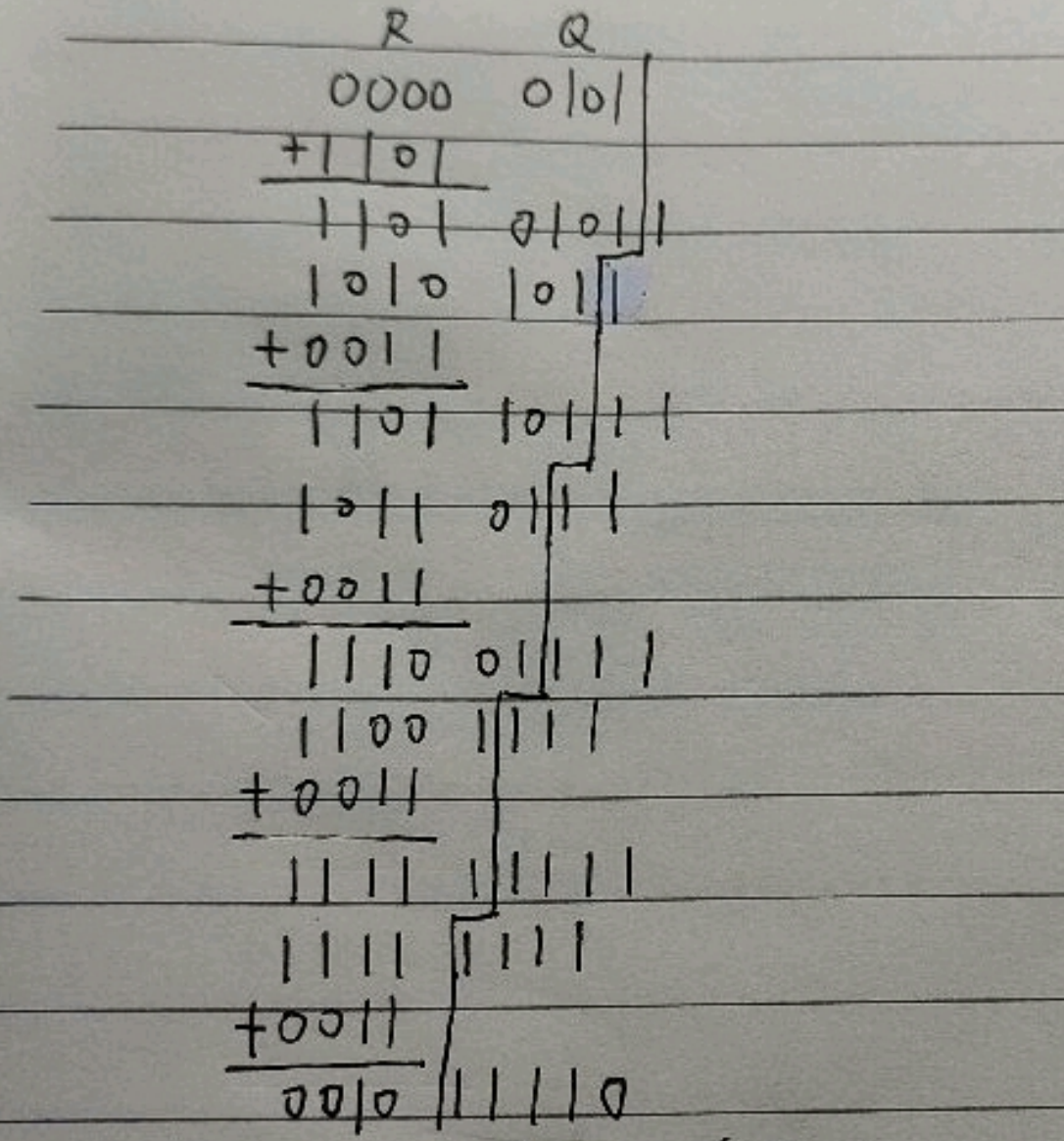
低位来的进位。当加法器分别采用串行进位

达式。

∴ 商: 0001 = 1

余数: 0000 = 0

(5) $[x]_{补}=0101$ $[Y]_{补}=1101$ $[-Y]_{补}=0011$



余数: 0010 真值: 2

商: 1110+1 = 1111 真值: -1

DATE

(4) $[X]_{补}=0.101$ $[|Y|]_{补}=0.101$ $[-|Y|]_{补}=1.011$

	R	Q
	0000	101
+	1011	
	1011	1010
	0111	010
+	0101	
	1100	0100
	1000	100
+	0101	
	1101	1000
	1011	000
+	0101	
	0000	0001

∴ 商: 0001 = 1

余数: 0000 = 0

$[Y]_{补}=1.101$ $[-Y]_{补}=0.011$

[x×y]补= 01

(3) $[x]_{补}=0101$ $[y]_{补}=1101$ $[-x]_{补}=1011$

P	Y	Y_{-1}	
0000	1101	0	
+0101			
0101	1101	0	>>2
0010	1110	1	
+1011			
1101	1110	1	>>2
1110	1111	0	

$[x\times y]_{补}=11101111$

$[x\times y]_{T}=-17$

$[x-y]_{补} = -9$

(2) $[x]_{原} = 0.101$ 、$[y]_{原} = 1.101$

C	P	Y	
0	0000	1101	
	+1101		
0	1101	1101	>>
0	0110	1.110	>>
0	0011	0.111	
	+0111		
0	1010	0.111	>>
0	0101	0011	
	+0011		
0	1000	001.1	>>
0	0100	0001	

$[x \times y]_{原} = 0.100000$

$[x \times y]_{补} = 65$

7. (1) $[x]_{补} = 0101$, $[y]_{补} = 1101$

$[x+y]_{补} = [x]_{补} + [y]_{补} \pmod{2^{补}}$

$= 0010$

$[x+y]_T = 2$

$[x-y]_{补} = [x]_{补} + [-y]_{补}$

$= 0101 + 0011 = 1000$

$[x-y]_T = -4$