6.20

(1) x=0.110111,y=-0.101110

原码一位乘

[x]原=0.110111 [y]原=1.101110 x*=0.110111 y*=0. 101110

部分积	乘数	操作
0.000000	.101110	->1
0.000000	0.10111	+x*
+0.110111		
0.110111		->1
0.011011	10.1011	+x*
+0.110111		
1.010010		->1
0.101001	010.101	+x*
+0.110111		
1.100000		->1
0.110000	0010.10	->1
0.011000	00010.1	+x*
+0.110111		
1.001111		->1
0.100111	100010.	

x*×y* = 0.100111 100010   x0⊕y0 = 1

[x×y]原=1.100111 100010   x×y=-0.100111 100010

原码两位乘

x*=0.110111   [-x*]补=111.001001   2x*=001.101110 y*=00. 101110

部分积	乘数	Cj	操作
000.000000	00. 101110	0	+2x*
+001.101110			
001.101110		0	->2
000.011011	1000.1011	0	+[-x*]补
+111.001001			
111.100100		1	->2
111.111001	001000.10	1	+[-x*]补
+111.001001			
111.000010		1	->2
111.110000	10001000.	1	+x*
+000.110111			

000.100111	100010	0	最后一步不移位

x*×y* = 0.100111 100010　x0⊕y0 = 1

[x×y]原=1.100111 100010　x×y=-0.100111 100010

补码一位乘（Booth 算法）

[x]补=0.110111 [y]补=1.010010 [-x]补=1.001001

部分积	乘数	yi+1	操作
00.000000	1.010010	0	->1
00.000000	01.01001	0	+[-x]补
+11.001001			
11.001001			->1
11.100100	101.0100	1	+[x]补
+00.110111			
00.011011			->1
00.001101	1101.010	0	->1
00.000110	11101.01	0	+[-x]补
+11.001001			
11.001111			->1
11.100111	111101.0	1	+[x]补
+00.110111			
00.011110			->1
00.001111	0111101.	0	+[-x]补
+11.001001			
11.011000	001111		最后一步不移位

[x×y]补=1.011000 001111　x×y=-0.100111 100010

补码两位乘

[x]补=000.110111 2[x]补=001.101110　[-x]补=111.001001 2[-x]补=110.010010

[y]补=11.010010

部分积	乘数	yi+1	操作
000.000000	11.010010	0	+2[-x]补
+110.010010			
110.010010			->2
111.100100	1011.0100	1	+[x]补
+000.110111			
000.011011			->2

000.000110	111011.01	0	+[x]补
+000.110111			
000.111101			->2
000.001111	01111011.	0	+[-x]补
+111.001001			
111.011000	011110		最后一步不移位

[x×y]补=1.011000 001111　x×y=-0.100111 100010

(2) x=-0.010111,y=-0.010101

原码一位乘

[x]原=1.010111 [y]原=1.010101 x*=0.010111 y*=0.010101

部分积	乘数	操作
0.000000	.010101	+x*
+0.010111		
0.010111		->1
0.001011	1.01010	->1
0.000101	11.0101	+x*
+0.010111		
0.011100		->1
0.001110	011.010	->1
0.000111	0011.01	+x*
+0.010111		
0.011110		->1
0.001111	00011.0	->1
0.000111	100011.	

x*×y* =0.000111 100011 x0⊕y0=0

[x×y]原=0.000111 100011　x×y==0.000111 100011

原码两位乘

x*=000.010111 [-x*]补=111.101001 2x*=000.101110 y*=00.010101

部分积	乘数	Cj	操作
000.000000	00.010101	0	+x*
+000.010111			
000.010111			->2
000.000101	1100.0101	0	+x*
+000.010111			
000.011100			->2

000.000111	001100.01	0	+x*
+000.010111			
000.011110			->2
000.000111	10001100.	0	
000.000111	100011		最后一步不移位

x*×y* =0.000111 100011 x0⊕y0=0

[x×y]原=0.000111 100011　 x×y=0.000111 100011

补码一位乘

[x]补=11.101001 [-x]补=00 .010111 [y]补=1.101011

部分积	乘数	yi+1	操作
00.000000	1.101011	0	+ [-x]补
+00.010111			
00.010111			->1
00.001011	11.10101	1	->1
00.000101	111.1010	1	+ [x]补
+11.101001			
11.101110			->1
11.110111	0111.101	0	+ [-x]补
+00.010111			
00.001110			->1
00.000111	00111.10	1	+ [x]补
+11.101001			
11.110000			->1
11.111000	000111.1	0	+ [-x]补
+00.010111			
00.001111			->1
00.000111	1000111.	1	最后一步不移位
00.000111	100011		

[x×y]补=00.000111 100011　 x×y=0. 000111 100011

补码两位乘

[x]补=111.101001 [-x]补=000.010111 2[x]补=111.110100 2[-x]补=000.001011

[y]补=11.101011

部分积	乘数	yi+1	操作
000.000000	11.101011	0	+[-x]补
+000.010111			

000.010111			->2
000.00 0101	1111.1010	1	+[-x]补
+000.010111			
000.011100			->2
000.000111	001111.10	1	+[-x]补
+000.010111			
000.011110			->2
000.000111	10001111.	1	最后一步不移位
000.000111	100011		

[x×y]补=000.000111 100011　 x×y=0. 000111 100011

(3) x=19,y=35

原码一位乘

[x]原=0,010011 [y]原=0,100011 x*=010011 y*=100011

部分积	乘数	操作
000000	,100011	+x*
+010011		
010011		->1
001001	1,10001	+x*
+010011		
011100		->1
001110	01,1000	->1
000111	001,100	->1
000011	1001,10	->1
000001	11001,1	+x*
+010011		
010100		->1
001010	011001,	

x*×y* =001010 011001 x0⊕y0=0

[x×y]原=0,001010 011001　 x×y=1 010 011 001 =665

原码两位乘

x*=000,010011 2x*=000,100110 [-x*]补=111,101101 y*=00,100011

部分积	乘数	Cj	操作
000,000000	00,100011	0	+[-x*]补
+111,101101		1	
111,101101			->2

111,111011	0100,1000	1	+x*
+000,010011		0	
000,001110			->2
000,000011	100100,10	0	+2x*
+000,100110			
000,101001			->2
000,001010	01100100,	0	最后一步不移位
000,001010	011001		

x*×y* =000,001010 011001 $x_0 \oplus y_0=0$

[x×y]原=0,001010 011001　　x×y=1 010 011 001 =665

补码一位乘

[x]补=00,010011 [-x]补=11,101101 [y]补=0,100011

部分积	乘数	yi+1	操作
00,000000	0,100011	0	+[-x]补
+11,101101			
11,101101			->1
11,110110	10,10001	1	->1
11,111011	010,1000	1	+[x]补
+00,010011			
00,001110			->1
00,000111	0010,100	0	->1
00,000011	10010,10	0	->1
00,000001	110010,1	0	+[-x]补
+11,101101			
11,101110			->1
11,110111	0110010,	1	+[x]补
+00,010011			
00,001010	011001		最后一步不移位

[x×y]补=00,001010 011001　　x×y=0,001010 011001

补码两位乘

[x]补=000,010011 [-x]补=111,101101 2[x]补=000,100110 2[-x]补=111,011010 [y]补=00,100011

部分积	乘数	yi+1	操作
000,000000	00,100011	0	+[-x]补
+111,101101			

111,101101			->2
111,111011	0100,1000	1	+[x]补
+000,010011			
000,001110			->2
000,000011	100100,10	0	+2[-x]补
+111,011010			
111,011101			->2
111,110111	01100100,	1	+[x]补
+000,010011			
000,001010	011001		最后一步不移位

[x×y]补=000,001010 011001　x×y=0,001010 011001

(4) x=0.11011,y=-0.11101

原码一位乘

[x]原=0.11011 [y]原=1.11101 x*=0.11011 y*=0.11101

部分积	乘数	操作
0.00000	.11101	+x*
+0.11011		
0.11011		->1
0.01101	1.1110	->1
0.00110	11.111	+x*
+0.11011		
1.00001		->1
0.10000	111.11	+x*
+0.11011		
1.01011		->1
0.10101	1111.1	+x*
+0.11011		
1.10000		->1
0.11000	01111	

x*×y* =0.11000 01111 x0⊕y0=1

[x×y]原=1.11000 01111　x×y=-0.11000 01111

原码两位乘

x*=000.110110 2x*=001.101100 [-x*]补=111.001010 y*=00.111010 （需要将小数点后位数补成偶数）

部分积	乘数	Cj	操作

000.000000 +001.101100	00.111010	0	+2x*
001.101100 000.011011 +001.101100	 0000.1110	 0	->2 +2x*
010.000111 000.100001 +111.001010	 110000.11	 0 1	->2 +[-x*]补
111.101011 111.111010 +000.110110	 11110000.	 1	->2 +x*
000.110000	111100		最后一步不移位

x*×y* =000.11000 01111 x0⊕y0=1

[x×y]原=1.11000 01111　x×y=-0.11000 01111

补码一位乘

[x]补=00.11011 [-x]补=11.00101 [y]补=1.00011

部分积	乘数	yi+1	操作
00.00000 +11.00101	1.00011	0	+[-x]补
11.00101 11.10010 11.11001 +00.11011	 11.0001 011.000	 1 1	->1 ->1 +[x]补
00.10100 00.01010 00.00101 00.00010 +11.00101	 0011.00 00011.0 100011.	 0 0 0	->1 ->1 ->1 +[-x]补
11.00111	10001		最后一步不移位

[x×y]补=11.00111 10001　x×y=-0.11000 01111

补码两位乘

[x]补=000.110110 [-x]补=111.001010 2[x]补=001.101100 2[-x]补=110.010100 [y]补=11.000110

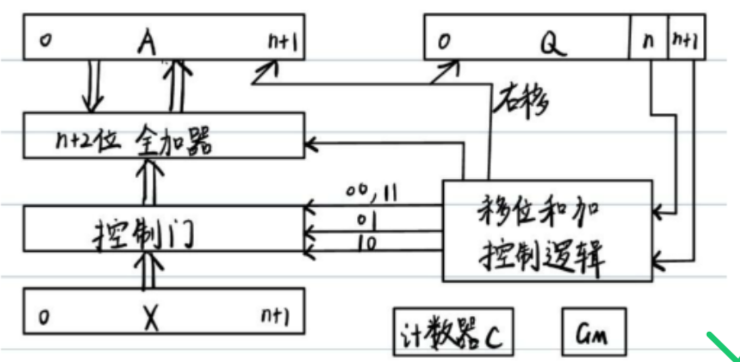
部分积	乘数	yi+1	操作
000.000000 +110.010100	11.000110	0	+ 2[-x]补
110.010100			->2

111.100101 +001.101100	0011.0001	1	+ 2[x]补
001.010001			->2
000.010100	010011.00	0	->2
000.000101 +111.001010	00010011.	0	+ [-x]补
111.001111	000100		最后一步不移位

[x×y]补=111.00111 1000100　x×y=-0.11000 01111

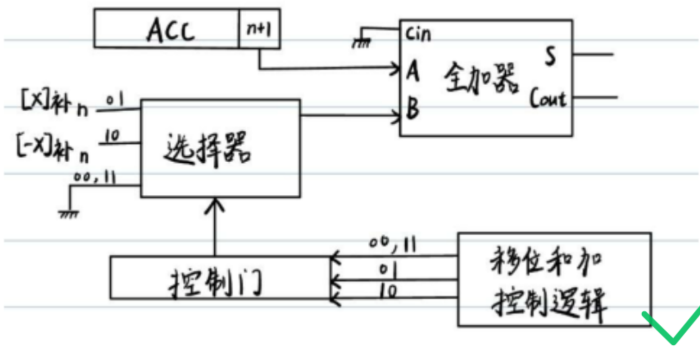
6.23

（1）寄存器和全加器均为 n+2 位



（2）共做 n 次移位，最多做 n+1 次加法（最后一步以及每次移位前都可以做一次）

（3）



（4）加的过程受 Q 寄存器末两位控制，当末两位为 00 或 11 时，部分积不变；当末两位为 01 时，部分积加上被乘数；末两位为 10 时，部分积减去被乘数，即与求补后的被乘数相加。

移位时 A、Q 两个寄存器串接（A//Q），一起右移一位（算术移位）。

6.21

（1）x=0.100111, y=0.101011

原码加减交替法

x*=0.100111 y*=0.101011 [-y*]补=1.010101

被除数（余数）	商	操作
0.100111 +1.010101	0.000000	第一步试减 +[-y*]补
1.111100 1.111000 +0.101011	0.	上商 0 1<- +y*
0.100011 1.000110 +1.010101	0.1	上商 1 1<- +[-y*]补
0.011011 0.110110 +1.010101	0.11	上商 1 1<- +[-y*]补
0.001011 0.010110 +1.010101	0.111	上商 1 1<- +[-y*]补
1.101011 1.010110 +0.101011	0.1110	上商 0 1<- +y*
0.000001 0.000010 +1.010101	0.11101	上商 1 1<- +[-y*]补
1.010111 +0.101011	0.111010	上商 0 +y*,恢复余数
0.000010		

[x*÷y*]原=0.111010　x0⊕y0=0

[x÷y]原=0.111010　x÷y=0.111010

补码加减交替法

[x]补=0.100111 [y]补=0.101011 [-y]补=1.010101

z 被除数（余数）	商	操作
00.100111	0.000000	z、y 同号

+11.010101		+[-y]补
11.111100 11.111000 +00.101011	0.	异号，上商 0 1<- +[y]补
00.100011 01.000110 +11.010101	0.1	同号，上商 1 1<- +[-y]补
00.011011 00.110110 +11.010101	0.11	同号，上商 1 1<- +[-y]补
00.001011 00.010110 +11.010101	0.111	同号，上商 1 1<- +[-y]补
11.101011 11.010110 +00.101011	0.1110	异号，上商 0 1<- +[y]补
00.000001 00.000010 +11.010101	0.11101	同号，上商 1 1<- +[-y]补
11.010111 +00.101011	0.111011	末位恒置 1 异号，+[y]补
00.000010		

[x÷y]补=0.111011　x÷y=0.111011

（2）x=-0.10101, y=0.11011

原码加减交替法

x*=0.10101 y*=0.11011 [-y*]补=1.00101

被除数（余数）	商	操作
0.10101 +1.00101	0.000000	第一步试减 +[-y*]补
1.11010 1.10100 +0.11011	0.	上商 0 1<- +y*
0.01111 0.11110 +1.00101	0.1	上商 1 1<- +[-y*]补
0.00011 0.00110 +1.00101	0.11	上商 1 1<- +[-y*]补
1.01011	0.110	上商 0

0.10110		1<-
+0.11011		+y*
1.10001	0.1100	上商 0
1.00010		1<-
+0.11011		+y*
1.11101	0.11000	上商 0
+0.11011		+y*，恢复余数
0.11000		

[x*÷y*]原=0.11000　x0⊕y0=1

[x÷y]原=1. 11000　x÷y=-0. 11000

补码加减交替法

[x]补=11.01011 [y]补=00.11011 [-y]补=11.00101

被除数（余数）	商	操作
11.01011	0.000000	异号
+00.11011		+[y]补
00.00110	1.	同号，上商 1
00.01100		1<-
+11.00101		+[-y]补
11.10001	1.0	异号，上商 0
11.00010		1<-
+00.11011		+[y]补
11.11101	1.00	异号，上商 0
11.11010		1<-
+00.11011		+[y]补
00.10101	1.001	同号，上商 1
01.01010		1<-
+11.00101		+[-y]补
00.01111	1.0011	同号，上商 1
00.11110		1<-
+11.00101		+[-y]补
00.00011	1.00111	末位恒置 1
+11.00101		同号，+[-y]补
11.01000		恢复余数

[x÷y]补=1.00111　x÷y=-0.11001

（3）x = 0.10100, y= -0.10001

原码加减交替法

x*=0.10100 y*=0.10001 [-y*]补=1.01111

被除数（余数）	商	操作
0.10100	0.00000	第一步试减
+1.01111		+[-y*]补
0.00011	1.	上商 1
0.00110		1<-
+1.01111		+[-y*]补
1.10101	1.0	上商 0
1.01010		1<-
+0.10001		+y*
1.11011	1.00	上商 0
1.10110		1<-
+0.10001		+y*
0.00111	1.001	上商 1
0.01110		1<-
+1.01111		+[-y*]补
1.11101	1.0010	上商 0
1.11010		1<-
+0.10001		+y*
0.01011	1.00101	上商 1
+1.01111		+[-y*]补，恢复余数
1.11010		

仅在余数为负数/余数与被除数不同号时
才需要恢复余数

[x*÷y*]原=1.00101　x0⊕y0=1

结果溢出！

补码加减交替法

[x]补=00.10100 [y]补=11.01111 [-y]补=00.10001

被除数（余数）	商	操作
00.10100	0.00000	异号，+[y]补
+11.01111		
00.00011	0.	异号，上商 0
00.00110		1<-
+11.01111		+[y]补
11.10101	0.1	同号，上商 1
11.01010		1<-
+00.10001		+[-y]补
11.11011	0.11	同号，上商 1
11.10110		1<-
+00.10001		+[-y]补
00.00111	0.110	异号，上商 0

00.01110 +11.01111		1<- +[y]补
11.11101 11.11010 +00.10001	0.1101	同号，上商 1 1<- +[-y]补
00.01011 +11.01111	0.11011	末位恒置 1 异号，+[y]补
11.11010		恢复余数

理论上结果会溢出，但实际计算出的结果无法看出异常，所以应当在计算开始前就判断是否有 x*>y*，若有，直接进行溢出处理；若没有，再进行计算。

(4) x=13/32, y=-27/32

原码加减交替法

[x]原=0.01101 [y]原=1.11011 x*=0.01101 y*=0.11011 [-y*]补=1.00101

被除数（余数）	商	操作
0.01101 +1.00101	0.00000	第一步试减 +[-y*]补
1.10010 1.00100 +0.11011	0.	上商 0 1<- +y*
1.11111 1.11110 +0.11011	0.0	上商 0 1<- +y*
0.11001 1.10010 +1.00101	0.01	上商 1 1<- +[-y*]补
0.10111 1.01110 +1.00101	0.011	上商 1 1<- +[-y*]补
0.10011 1.00110 +1.00101	0.0111	上商 1 1<- +[-y*]补
0.01011	0.01111	上商 1

[x*÷y*]原=0.01111　x0⊕y0=1

[x÷y]原=1.01111　x÷y=-0.01111

补码加减交替法

[x]补=00.01101 [y]补=11.00101 [-y]补=00.11011

被除数（余数）	商	操作
00.01101	0.00000	异号
+11.00101		+[y]补
11.10010	1.	同号，上商 1
11.00100		1<-
+00.11011		+[-y]补
11.11111	1.1	同号，上商 1
11.11110		1<-
+00.11011		+[-y]补
00.11001	1.10	异号，上商 0
01.10010		1<-
+11.00101		+[y]补
00.10111	1.100	异号，上商 0
01.01110		1<-
+11.00101		+[y]补
00.10011	1.1000	异号，上商 0
01.00110		1<-
+11.00101		+[y]补
00.01011	1.10001	末位恒置 1

[x÷y]补=1.10001　x÷y=-0.01111=-15/32