2.5 二进制的乘除

2.5 二进制乘除

01 定点数一位乘法

02 补码乘法（Booth算法）

03 恢复余数除法

04 不恢复余数除法

2.5.1 无符号数乘法

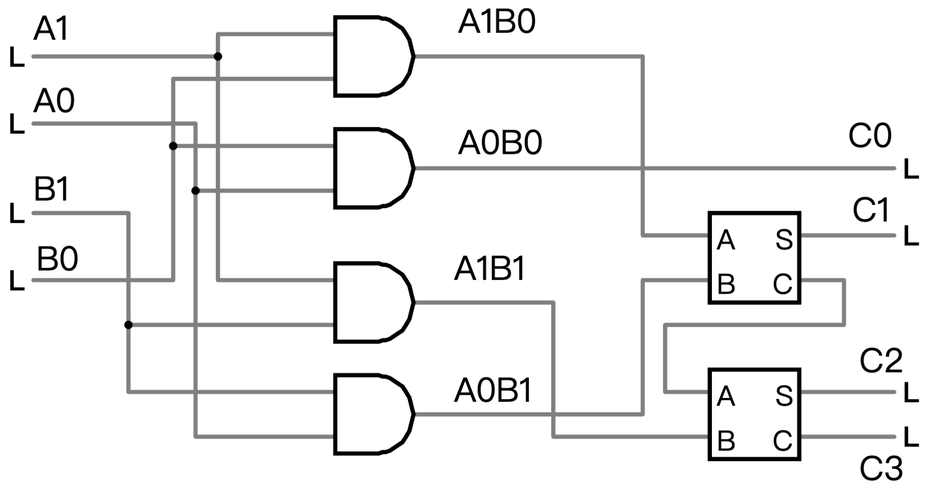
1、类似十进制乘法

例：3x3

2.5.1 无符号数乘法实现

1、类似十进制乘法

例：3x3

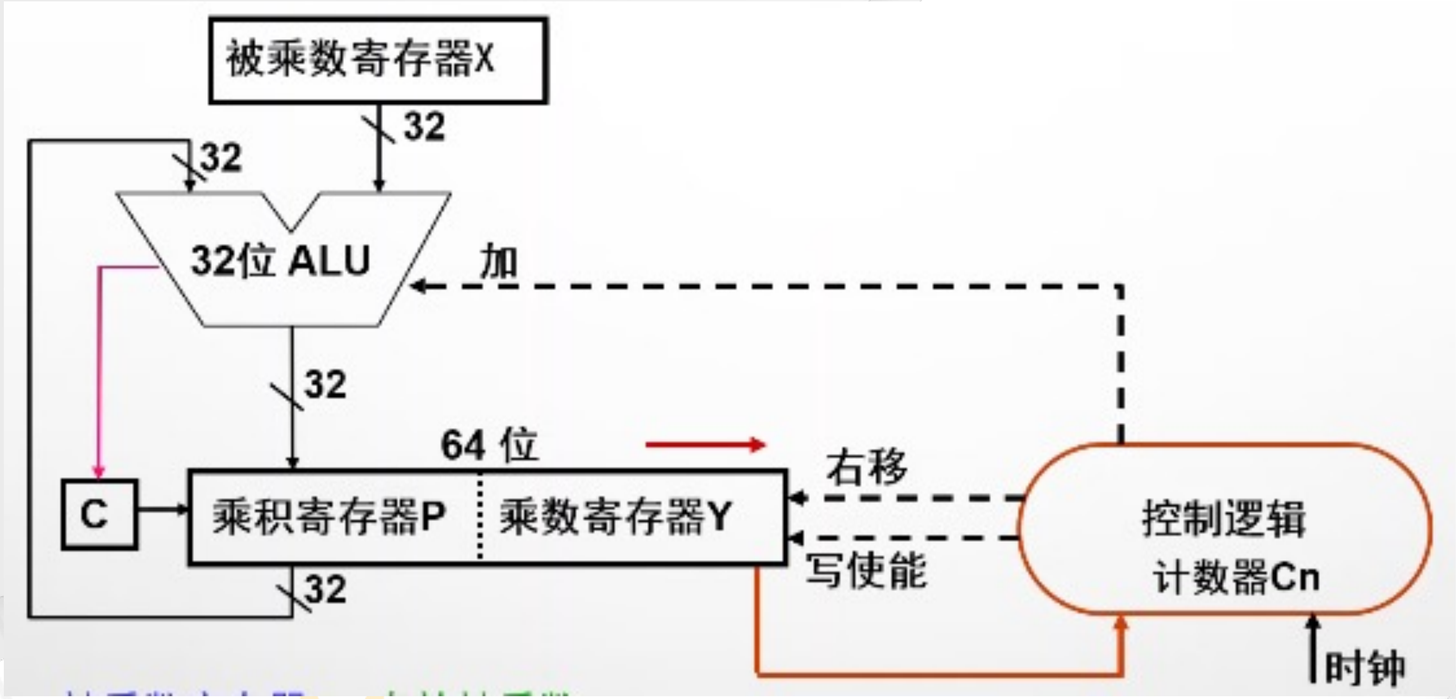


2.5.1 无符号数乘法实现

例：11010 × 1010

2.5.1 无符号数乘法：原码一位乘法运算单元

1、用加法运算和定长寄存器做乘法：
看乘数最后一位，如果是0，乘右移，如果是1，加上被乘数后右移。计算过程的移位次数为乘数的位数。



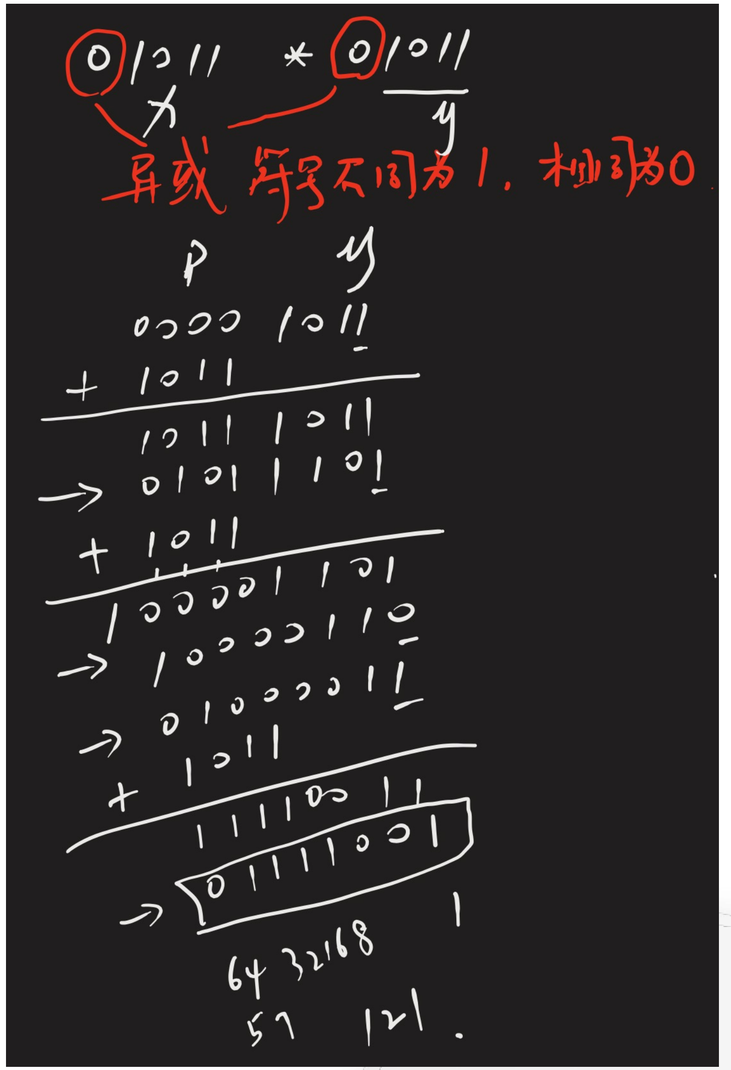
2.5.1 有符号原码数乘法

1、符号单独异或，数值相乘

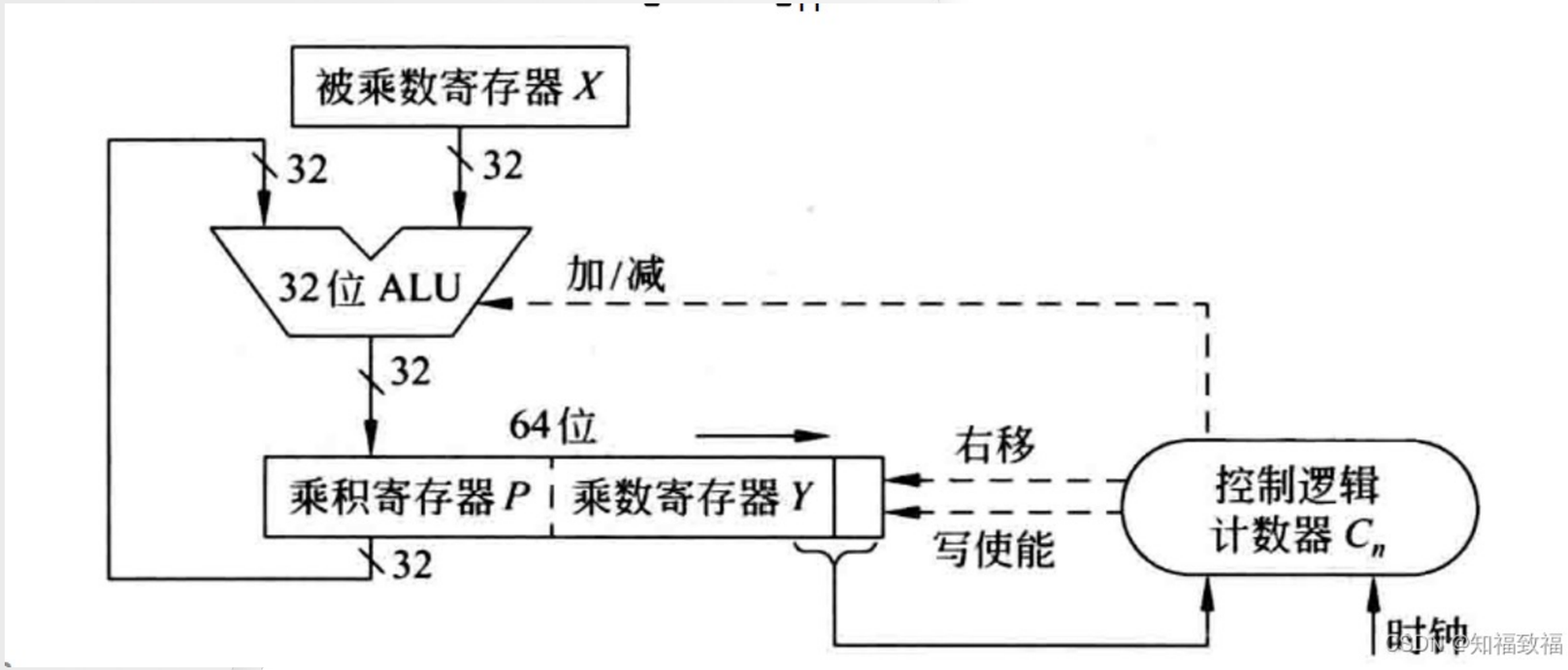
例：11x11

小结
1.将被乘数和乘数都化为原码
2.被乘数和乘数符号位之间异或
3.用原码一位相乘
4.将结果添加对应符号位

问题：符号要单独处理异或



2.5.2 补码一位乘法（booth算法逻辑框图）



例：x = 1101, y = -1010，求【x*y】$_{补}$及【x*y】

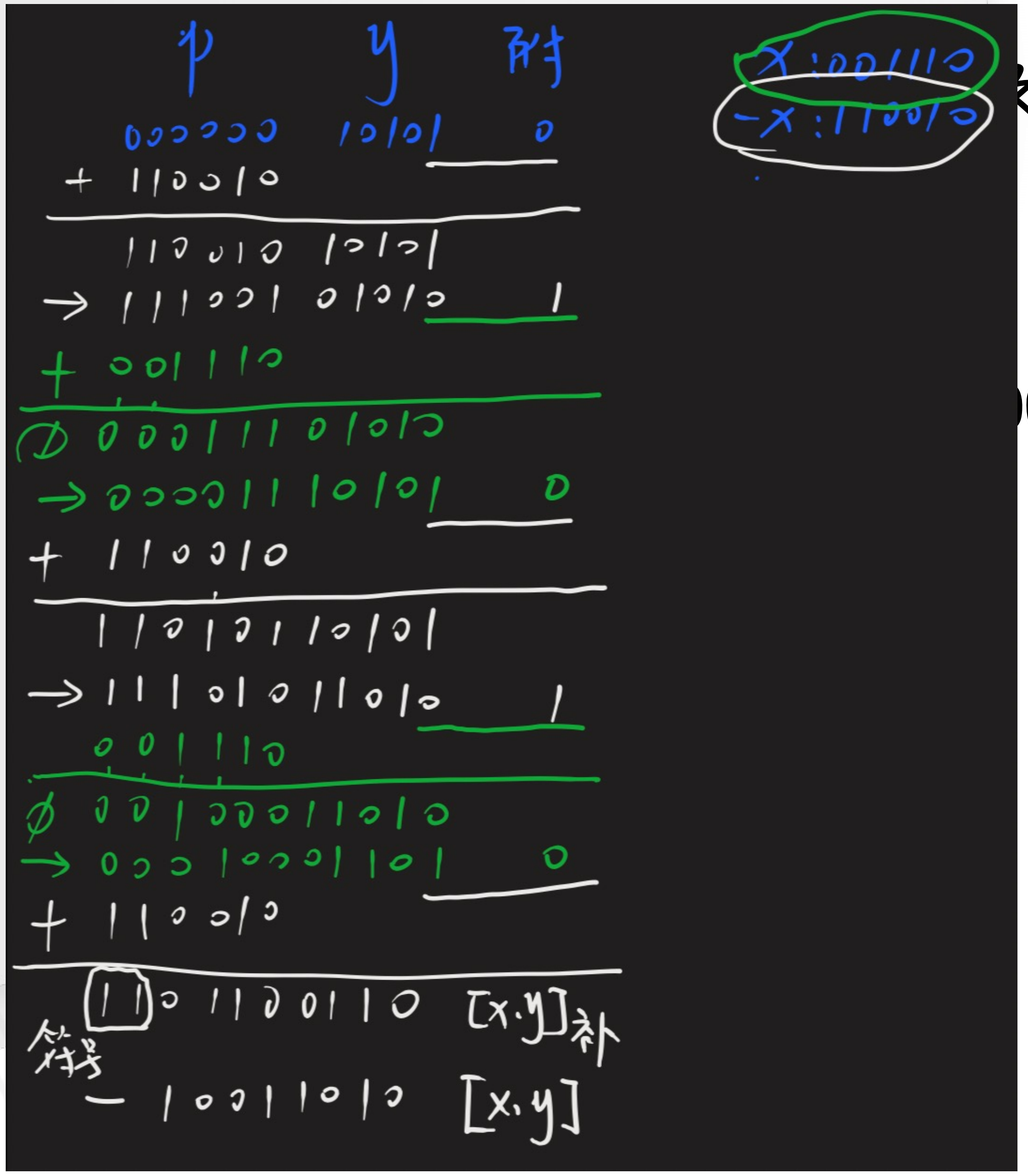
2.5.2 补码一位乘法（Booth算法）

1.符号位参与计算
2.采用补码进行计算
3.被乘数X 一般取**双符号位**参与计算，并且让部分积P初始值为0，长度与被乘数X相同，乘数Y 可取**单符号位**
4.开始计算时，乘数Y 末尾增设附加位（Yn+1），值为0
5.移位规则（移位看乘数后两位（包括附加位），部分积右移时补位看最高位，超出的进位丢弃，移位n次，移出位n个）

移位规则：
•00/11 部分积右移一位（即加0再右移一位）
•01 部分积加[X]补，且右移一位
•10 部分积加[-X]补，且右移一位

以4位有效bit为例：乘数移出4bit，处理过4次规则。

2.5.2 举例



求[x*y]，写出计算过程。

$[x]_{补}$ = 01110, $[-x]_{补}$ = 10010, $[y]_{补}$ = 10101

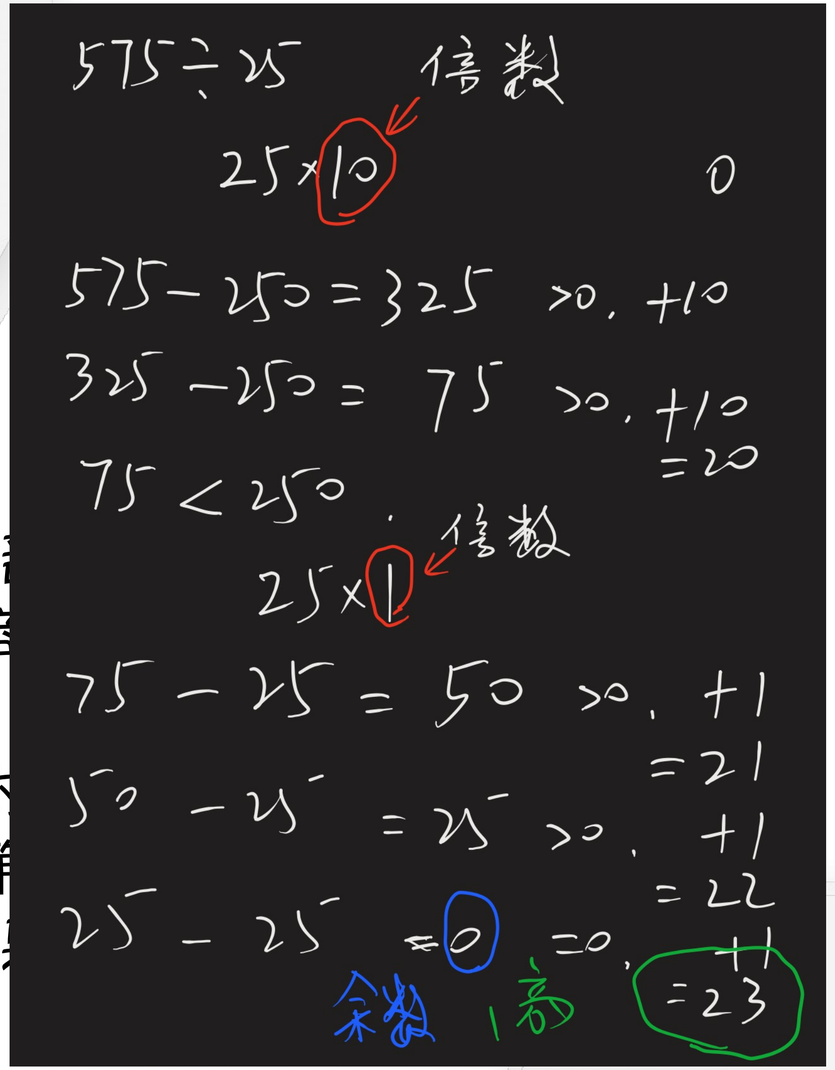
0011010。

2.5.3 十进制数用加减处理除（恢复余数）

1、商的符号为两数符号异或
2、值为两数绝对值除

例：575÷25

1、将除数和被除数对齐
2、商清0
3、被除数减对齐后的除数得到
4、如果部分被除数大于0，则商
第4步，直到部分被除数小于0.
5、将小于0的部分被除数加上对
6、对除数右移，并重复第3到第
7、将所有得到的商相加得到结
加上原除数得到余数。



2.5.3 定点数原码除法（恢复余数）

1、商的符号为两数符号异或
2、值为两数绝对值除

例：01100111÷1001

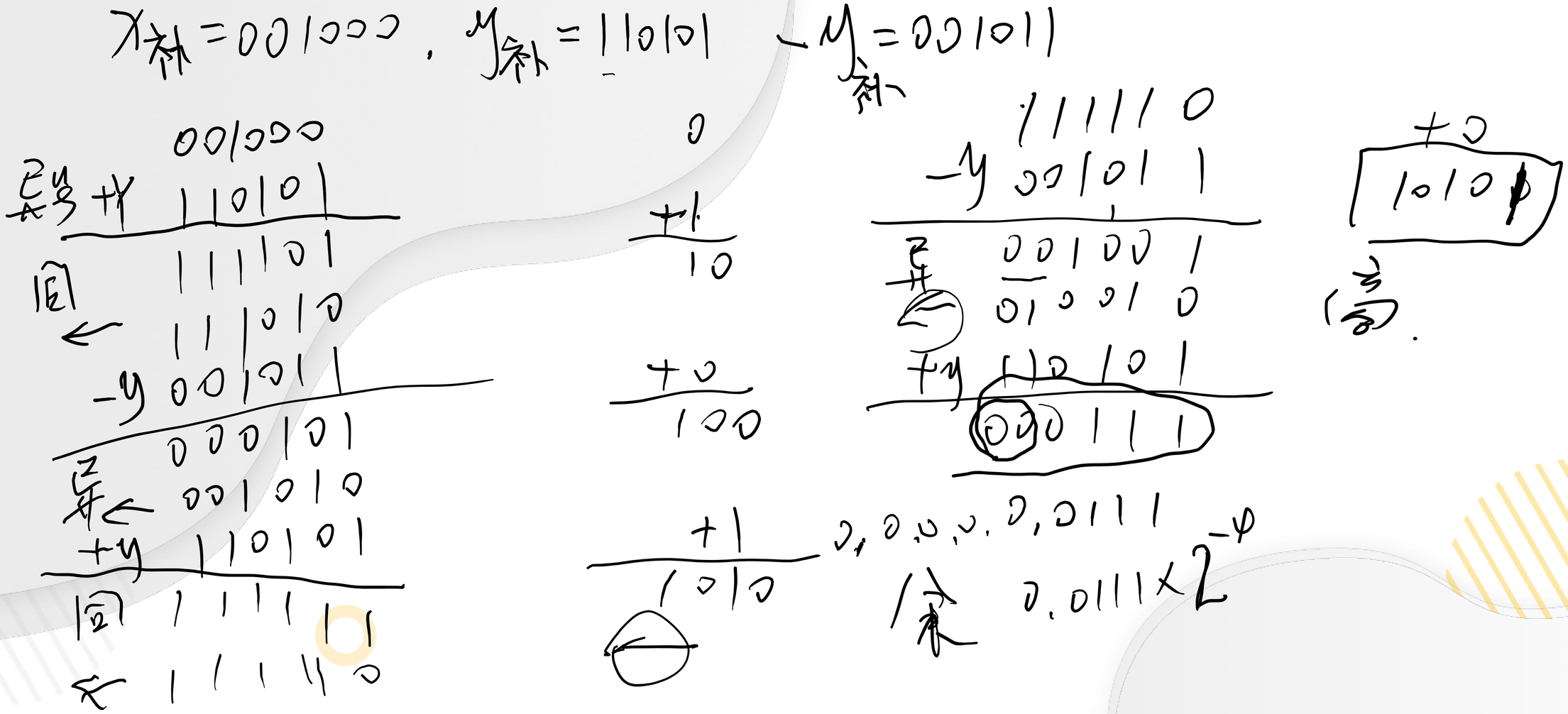
1、将除数和被除数对齐
2、商清0
3、被除数减对齐后的除数得到部分被除数
4、如果部分被除数为正，则商左移补1.
5、如果部分被除数为负，则商左移补0. 并将小于0的部分被除数加上对齐后的除数
6、对除数右移，并重复第3到第5，直到减原被除数为负，
7、最后的负部分被除数加上原除。

2.5.3 定点数补码除法（加减交替）

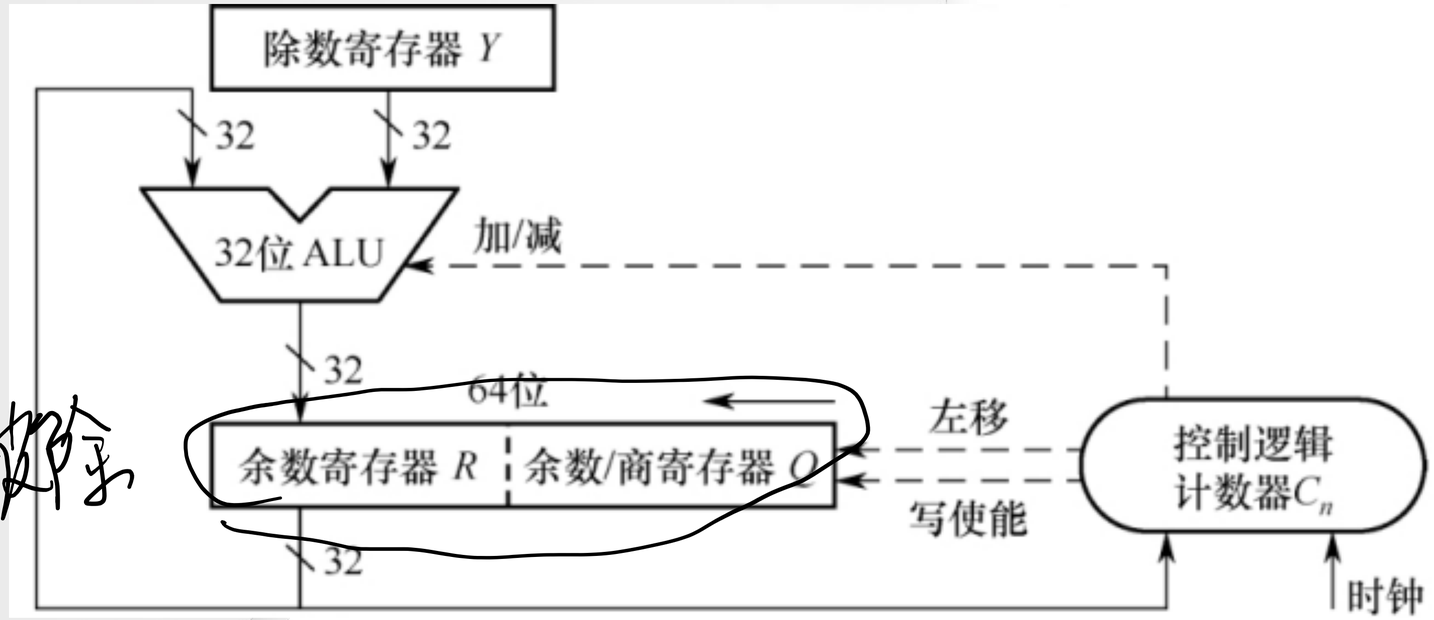
1、符号位参与运算，且采用双符号位。
2、若被除数与除数同号，则被除数减除数；否则就加上除数。
3、余数与除数同号，商加1左移，余数左移减去除数；余数与除数异号，商加0左移，余数左移加上除数。
4、重复第3步n次
5、商末位恒置1，余数右移n位。

2.5.3 定点数补码除法（加减交替）

例：x = 0.1000,y = -0.1011，采用补码加减交替法求x/y。



2.5.3 32位补码除法逻辑结构图



2.5. 本节总结

1. 原码一位乘法可以使用三个等长寄存器通过移位和加运算得到最后乘积
2. 积是原数据的两倍字长
3. 原码积符号位要单独处理，补码一位乘法可带符号运算。掌握Booth算法运算过程。

海贼宝藏
专注IT教育在线学习平台

欢迎参与学习

WELCOME FOR YOUR JOINING

船说：计算机基础