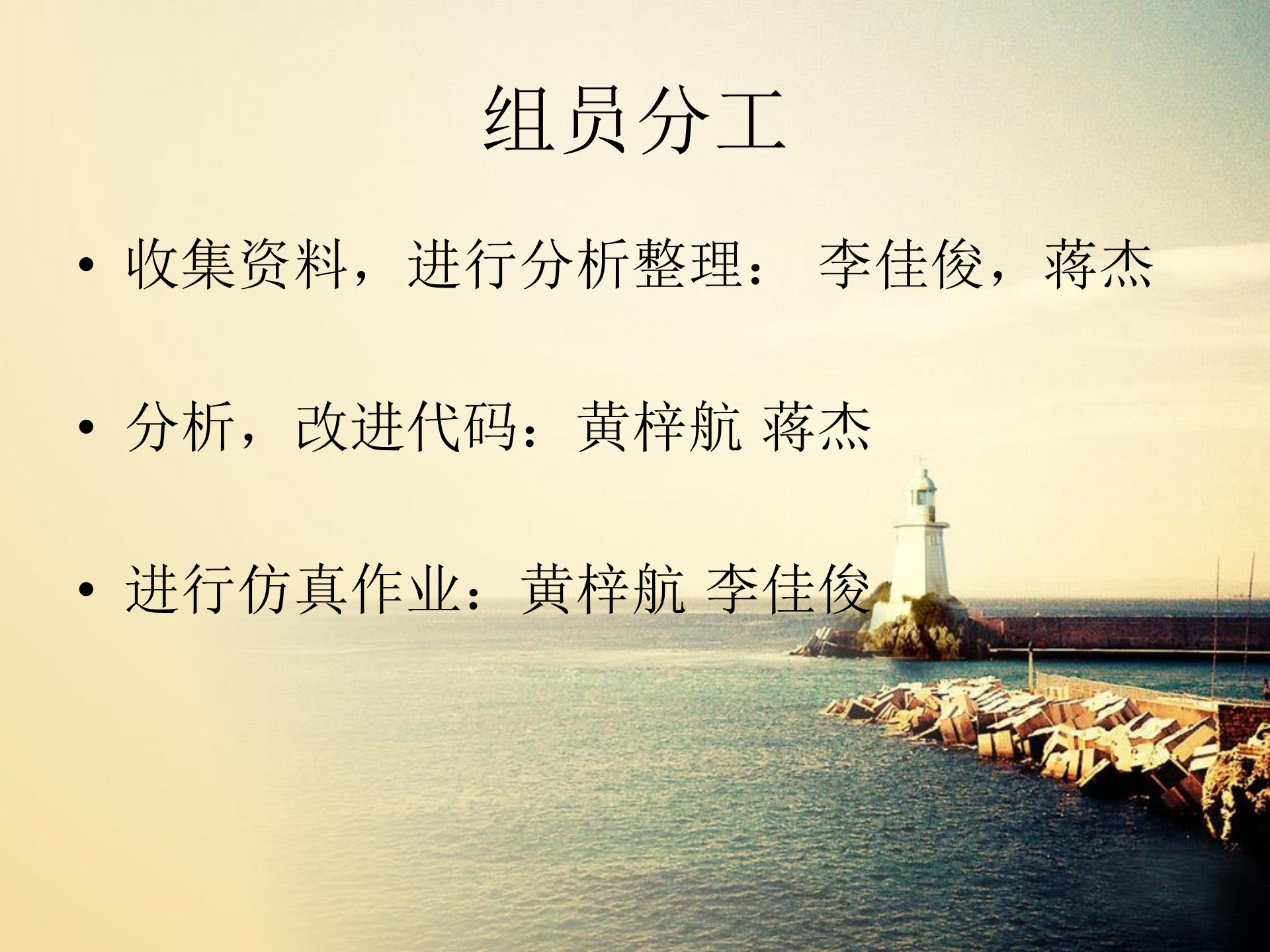
开平方算法器的设计

老师：郑海永
队员：　　李佳俊
　　蒋杰
　　黄梓航

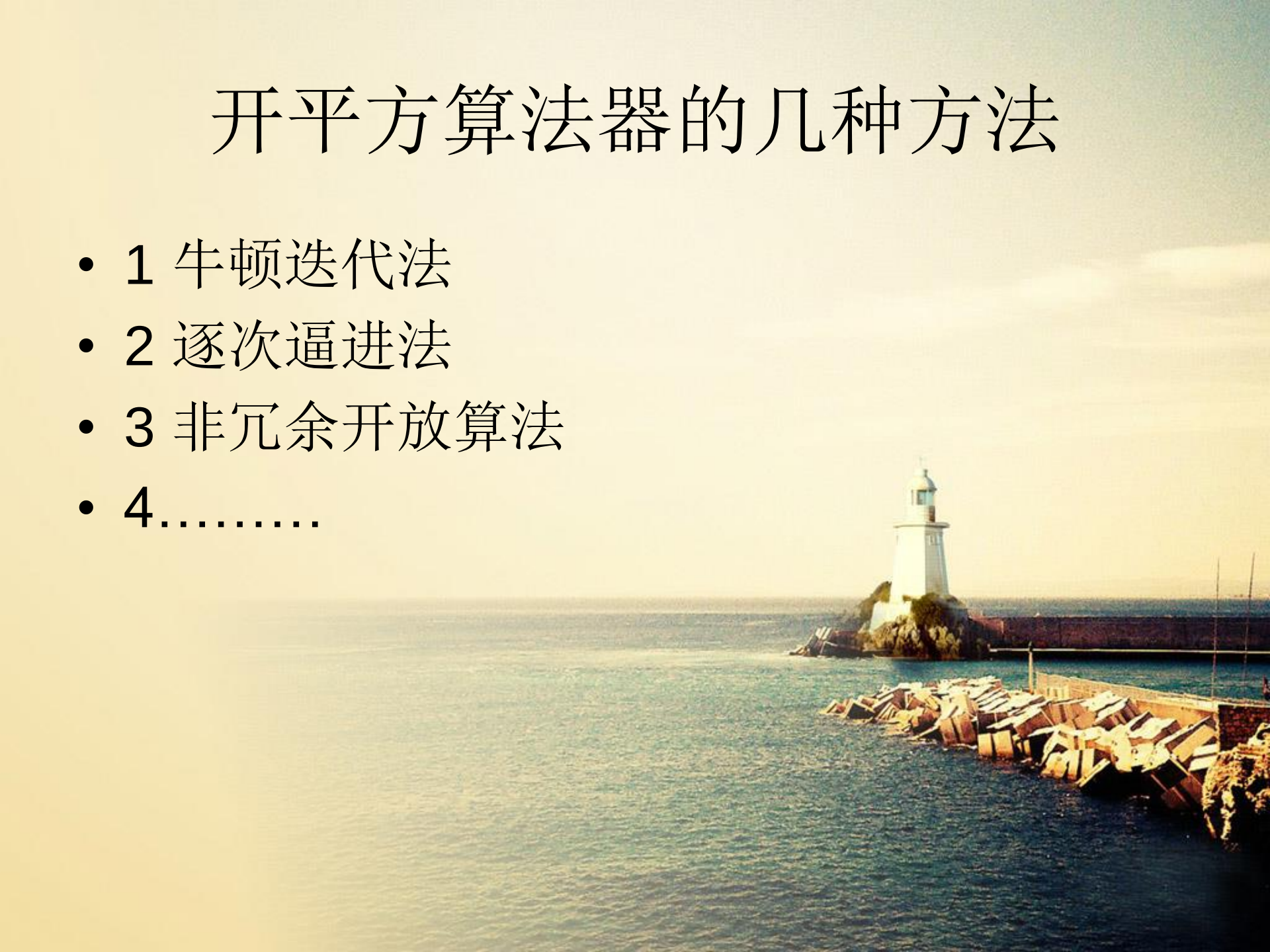
组员分工

- 收集资料，进行分析整理： 李佳俊，蒋杰
- 分析，改进代码：黄梓航 蒋杰
- 进行仿真作业：黄梓航 李佳俊



开平方算法器的几种方法

- 1 牛顿迭代法
- 2 逐次逼进法
- 3 非冗余开放算法
- 4……



二分逐次逼进法的原理

- 一般地，对于函数f(x),如果存在实数c,当x=c时，若f(c)=0,那么把x=c叫做函数f(x)的零点。
- 解方程即要求f(x)的所有零点。
- 假定f(x)在区间（x，y）上连续
- 先找到a、b属于区间（x，y），使f(a)，f(b)异号，说明在区间(a,b)内一定有零点，然后求f[(a+b)/2],
- 现在假设f(a)<0,f(b)>0,a<b
- ①如果f[(a+b)/2]=0，该点就是零点，
- ②如果f[(a+b)/2]<0,则在区间（(a+b)/2，b)内有零点，(a+b)/2赋给a，从①开始继续使用中点函数值判断。
- ③如果f[(a+b)/2]>0，则在区间(a,(a+b)/2)内有零点，(a+b)/2赋给b，从①开始继续使用中点函数值判断。
- 这样就可以不断接近零点。当区间小于一定值时，结束迭代过程。

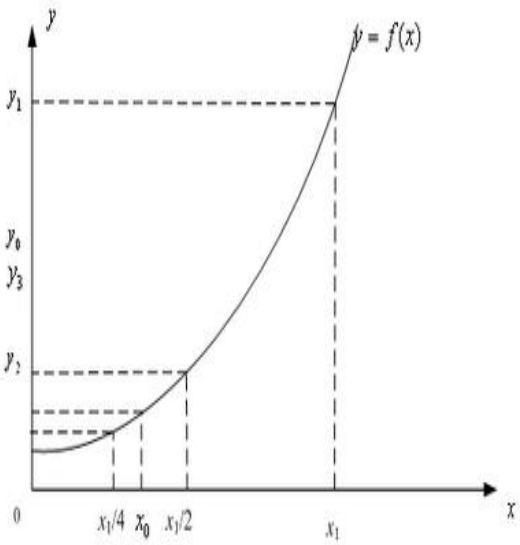


图1 二分逐次逼近原理

牛顿迭代法

- 一、确定迭代变量
- 在可以用迭代算法解决的问题中，至少存在一个可直接或间接地不断由旧值递推出新值的变量，这个变量就是迭代变量。
- 二、建立迭代关系式
- 所谓迭代关系式，指如何从变量的前一个值推出其下一个值的公式（或关系）。迭代关系式的建立是解决迭代问题的关键，通常可以使用递推或倒推的方法来完成。
- 三、对迭代过程进行控制
- 在什么时候结束迭代过程？这是编写迭代程序必须考虑的问题。不能让迭代过程无休止地执行下去。迭代过程的控制通常可分为两种情况：一种是所需的迭代次数是个确定的值，可以计算出来；另一种是所需的迭代次数无法确定。对于前一种情况，可以构建一个固定次数的循环来实现对迭代过程的控制；对于后一种情况，需要进一步分析得出可用来结束迭代过程的条件。

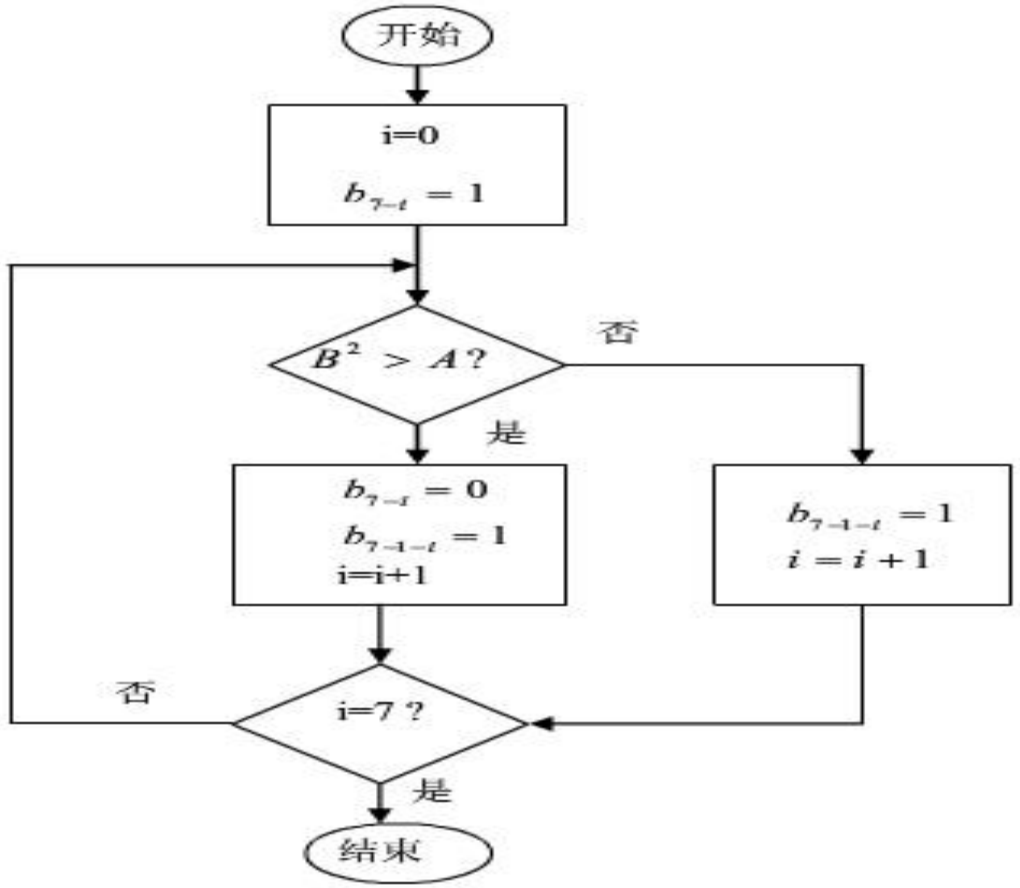
程序流程设计原理

- 。

由于开平方运算是对正数进行操作，所以我们这里的定点数都是正数，也就是说对于 n 位的定点数，其实际可以表示的整数的范围是 $0 \sim 2^{n-1}$。假设我们的被开方数 A 是 n 位，其平方根的近似值 B 也是 m 位的整数，则 m 与 n 之间的关系是：m=n/2，n 为偶数；m=（n+1）/2，n 为奇数。

其开方原理是我们先假设 B 的最高位是 1，然后计算 B 的平方，如果 B 的平方大于 A，则将 B 的最高位设为 0；如果 B 的平方小于 A，则 B 的最高位不变。接下来处理 B 的次高位，同最高位一样，先赋值 1，如果 B 的平方大于 A，则此位为 0，否则为 1，依此类推，直到 B 的最低位，这样计算出的 B 的值最接近 A 的平方根。

程序设计的流程图



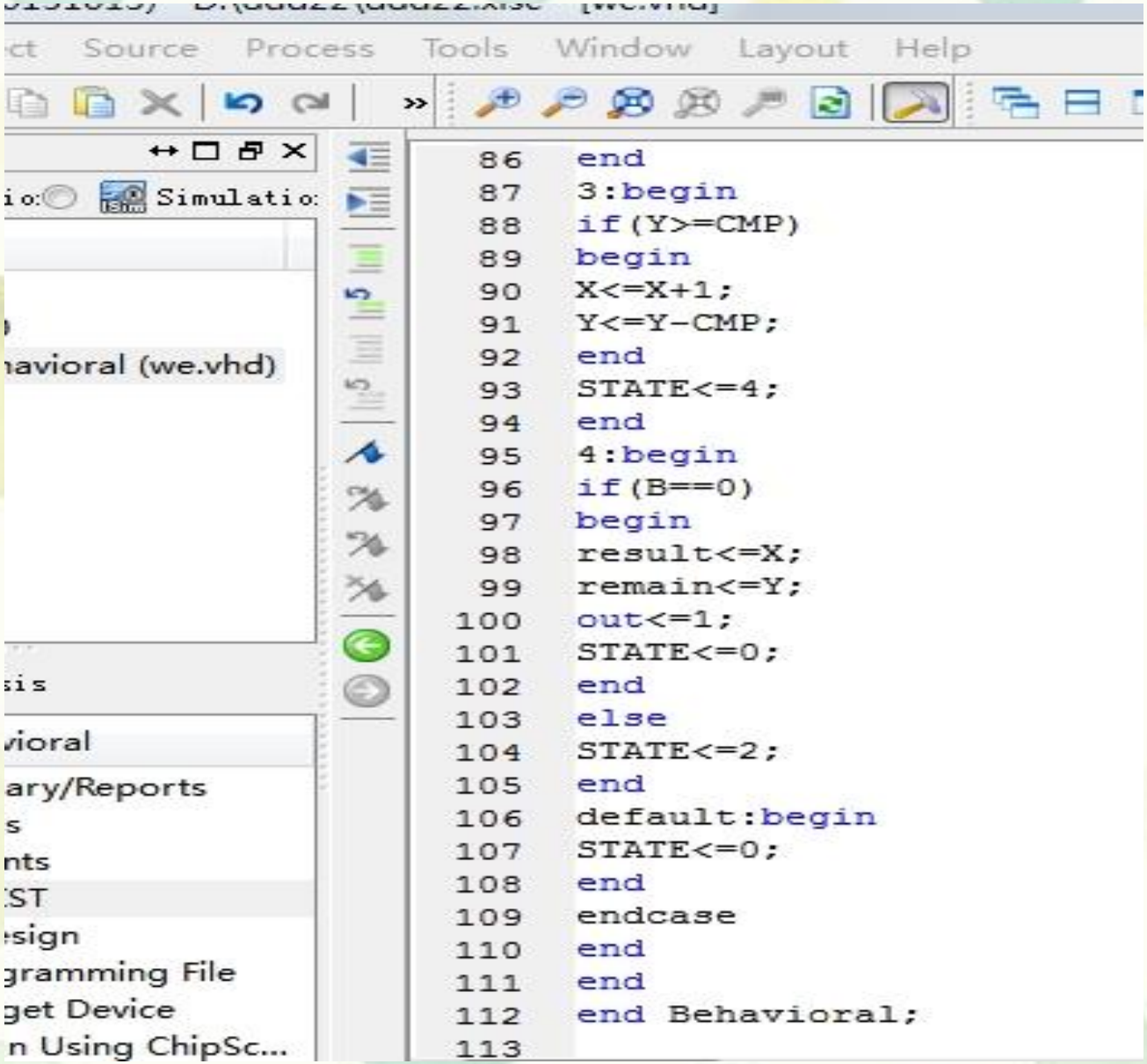
程序的输入输出

```
    Port ( rst_n : in  STD_LOGIC;
           clk : in  STD_LOGIC;
           start : in  STD_LOGIC;
           out : out  STD_LOGIC;
           data : in  STD_LOGIC_VECTOR (79 downto 0);
           remain : out  STD_LOGIC_VECTOR (79 downto 0);
           result : out  STD_LOGIC_VECTOR (79 downto 0));
```

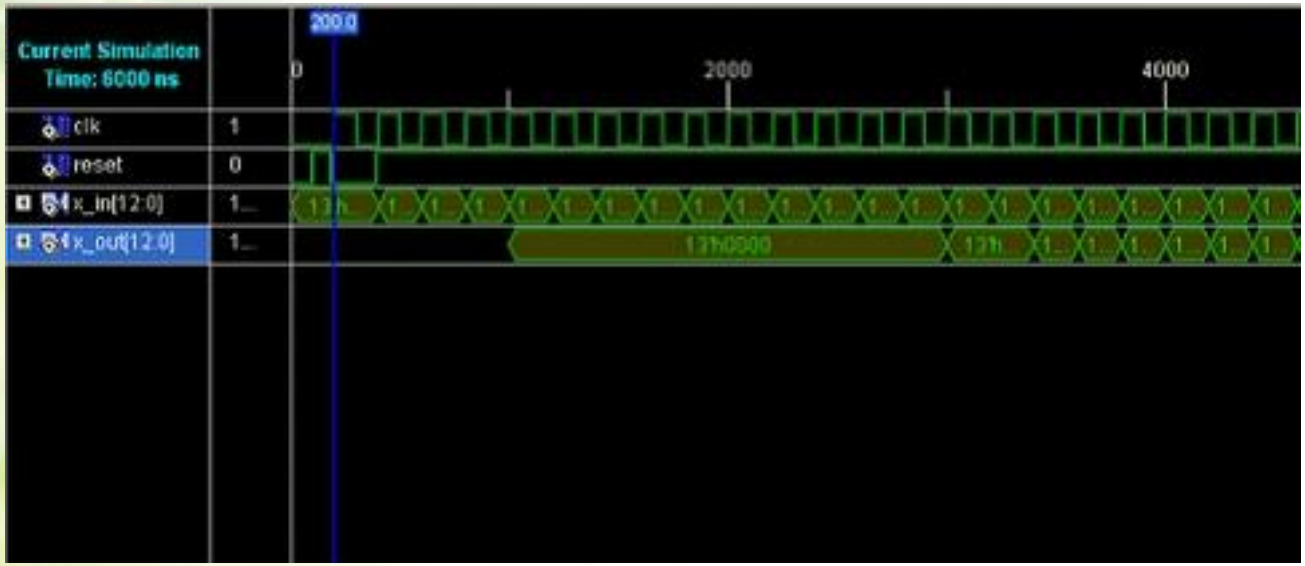
部分程序代码

```
begin
STATE=0;
out=0;
end
always@(posedge clk or negedge rst_n)
if(!rst_n)
begin
STATE<=0;
out<=0;
end
else
begin
case(STATE)
0:begin
out<=0;
if(start)
begin
STATE<=1;
X<=0;
Y<=data;
A<=data>>78;
B<=78;
end
end
1:begin
if(A>=1)
begin
```

```
out<=0;
end
else
begin
case(STATE)
0:begin
out<=0;
if(start)
begin
STATE<=1;
X<=0;
Y<=data;
A<=data>>78;
B<=78;
end
end
1:begin
if(A>=1)
begin
X<=1;
Y<=Y-(80'd1<<B);
end
STATE<=2;
end
2:begin
B<=B-2;
X<=X<<1;
```


ct Source Process Tools Window Layout Help
Simulatio:
havioral (we.vhd)
sis
vioral
ary/Reports
s
nts
ST
sign
gramming File
get Device
n Using ChipSc...
86 end
87 3:begin
88 if(Y>=CMP)
89 begin
90 X<=X+1;
91 Y<=Y-CMP;
92 end
93 STATE<=4;
94 end
95 4:begin
96 if(B==0)
97 begin
98 result<=X;
99 remain<=Y;
100 out<=1;
101 STATE<=0;
102 end
103 else
104 STATE<=2;
105 end
106 default:begin
107 STATE<=0;
108 end
109 endcase
110 end
111 end
112 end Behavioral;
113

ISE仿真图


Current Simulation Time: 6000 ns
2000
0
2000
4000
clk
1
reset
0
x_in[12:0]
1...
x_out[12:0]
1...

遇到的困难

- 1. ISE 新版本的软件跟旧版有很大的不同，操作起来比较生疏。
- 2.开平方的算法比较困难，方法繁多，不好实现。
- 3.网上的代码基本上都不完全，可以参照的比较少。
- 4.ise的安装也失败过几次

thank you!

The end