

# **数字逻辑实验报告（4**）

|  |  |  |
| --- | --- | --- |
| **数字逻辑实验4** | | |
| **斐波那契(Fibonacci)数列计算器设计** | **成绩** |  |

评语：（包含：预习报告内容、实验过程、实验结果及分析）

教师签名

**姓 名： 葛松**

**学 号： U201714668**

**班 级： CS1703**

**指 导 教 师： 石宣化**

**计算机科学与技术学院**

**2019 年 6 月 4 日**



**数字逻辑实验报告**

斐波那契(Fibonacci)数列计算器设计

1、实验名称

斐波那契(Fibonacci)数列计算器设计。

2、实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件logisim进行斐波那契(Fibonacci)数列计算器设计和验证，记录实验结果，验证设计是否达到要求。

通过斐波那契(Fibonacci)数列计算器的设计、仿真、验证3个训练过程，掌握数字逻辑电路的设计、仿真、调试的方法。

3、实验所用设备

Logisim2.7.1软件一套。

4、实验内容

斐波那契(Fibonacci)数列中每项数值都是其两个直接前项的和，其生成规则如下公式1所示：

（公式1）

**（1）求Fibonacci数的矩阵算法**

　　首先，对于数列的初始条件对应公式2的矩阵运算：

（公式2）

更一般化地，有公式3：

（公式3）

所以，根据递推关系可以得到公式4：

（公式4）

由公式4可推出，。

因此，对求斐波那契数列的第n项的问题，可以转化为对一个二维矩阵求n次幂。采用矩阵的快速幂算法，操作次数可优化为O(log2 n)。

由于F(47)=(2971215073)10<232，F(48)=(4807526976)10>232，电路中采用32位二进制数表示一个整数。为了避免整数溢出，取2≤n≤47，n用6位二进制数表示。

**（2）算法描述**

Fibonacci(){

初始化：, Start=0；

For (i=5 downto 0)

{

if (Start==0) then

{

if (n[i]==1) then Start=1;

}

Else

{

if (n[i]==1)

then X=X2•A;

else

X=X2; }

}

return(X);

}

例如：n = (101100)2 = (44)10

step1：i=5，Start=0，n[5]=1，此时Start置1；

step2：i=4，Start=1，n[4]=0，此时X = X2 = A2；

step3：i=3，Start=1，n[3]=1，此时X = X2 •A = (A2)2•A；

step4：i=2，Start=1，n[2]=1，此时X = X2 •A = ((A2)2•A)2 •A；

step5：i=1，Start=1，n[1]=0，此时X = X2 = (((A2)2•A)2 •A)2；

step6：i=0，Start=1，n[0]=0，此时X = X2 = ((((A2)2•A)2 •A)2)2；

循环执行完后，X = ((((A2)2•A)2 •A)2)2 = A44

**（3）矩阵计算模块**

计算X2模块sqrX

（公式5）

其相应的输入/输出如图1所示。

sqrX

a

b

c

d

a′ = a2+bc

b′ = ab+bd

c′ = ac+cd

d′ = bc+d2

图1 计算X2模块sqrX输入/输出示意图

这里，a, b, c, d, a′, b′, c′, d′都为32位无符号二进制整数。

计算X2·A模块sqrX\*A

（公式6）

其相应的输入/输出如图2所示。

sqrX\*A

a

b

c

d

a″ = ab+bd

b″ = a2+bc+ab+bd

c″ = bc+d2

d″ = ac+cd+bc+d2

图2 计算X2·A模块sqrX\*A输入/输出示意图

这里，a, b, c, d, a″, b″, c″, d″都为32位无符号二进制整数。

**（4）矩阵快速幂算法迭代模块**

该模块Fibo输入/输出端如图3所示。

Fibo

start

clr

ni-1

clk

Fi = bi′ or bi″

图3 Fibo输入/输出示意图

这里，start为Fibonacci()算法中的6位二进制数n左移出的第一个（最高位的）1的标志信号；ni-1是start=1之后左移出的下一位；clr为初始化（清零）信号，此时X = A；clk为时钟脉冲信号。Fi为Fibonacci()算法迭代的中间结果，根据ni-1取0或1来决定Fi是取sqrX或者sqrX\*A运算后的矩阵元素bi，在第6个时钟脉冲时，Fi即为输入n的Fibonacci数Fn。

其内部逻辑结构图如图4所示。



图4 Fibo内部逻辑结构图

**（5）Fibonacci数显示模块**

将二进制数转换成十进制数在数码显示管上显示出来。

输入为32位二进制的Fibonacci数F(n)。

由于32位二进制Fibonacci数表示的最大十进制数的位数是10位，该模块的输出为10组8421BCD码D9、D8、D7、D6、D5、D4、D3、D2、D1、D0，每组8421BCD码表示1位10进制数。

**（6）主模块main**

主模块main的逻辑结构图5所示。



图5 主模块main的逻辑结构图

控制器Controller中包括三个功能块：6位二进制数n的左移控制电路、6个时钟脉冲控制电路、start信号产生电路。

6位二进制数n的左移控制电路，使用一个移位寄存器，在时钟脉冲作用下产生ni-1。用clear信号装入n，进行移位寄存器的初始化。

使用1个8位计数器、1个比较器和适当的门电路，可以控制Fibo只接收6个clock时钟脉冲（产生clk）。直至下一个clear信号初始化后，才准备产生下一组6个时钟脉冲。

使用1个D触发器加适当的门电路构成一个锁存器Latch，在接收到n的最高位1时start=1，直至下一个clear信号使start=0。

在6个clock时钟脉冲信号后，电路就产生了第n个Fibonacci数F(n)，并经过Display电路转换成十进制数在数码管上显示出来。

5、实验方案设计

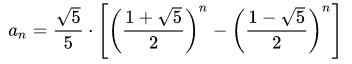
**（1）斐波那契(Fibonacci)数列计算**

**要求：**

给出Fibonacci数列通项公式；

斐波那契(Fibonacci)数列中每项数值都是其两个直接前项的和，其生成规则如下公式1所示：

数列的通项公式如下所示：



给出Fibonacci数列的递归算法（指数时间复杂度）形式化描述；

以下给出C语言的递归算法的实现：

int Fibo(int n){

if (n == 1 || n == 2){

return 1;

}

return Fibo(n-1) + Fibo(n-2);

}

时间复杂度为O()

给出Fibonacci数列的多项式时间复杂度算法形式化描述。

递推法：这个方法就是按照递推方程，从n = 0和n = 1开始，逐个求出所有小于n的Fibonacci数，最后就可以算出F(n)。由于每次计算值需要用到前两个Fibonacci数，更小的数就可以丢弃了，可以将空间复杂度降到最低。算法如下：

int Fibo(int n){

int f0 = 1;

int f1 = 1;

int f2 = 0;

for (int i = 2; i < n; i++){

f2 = f0 + f1;

f0 = f1;

f1 = f2;

}

return f2;

}

时间复杂度为O(n)

**（2）计算矩阵X2模块**

**要求：**

给出矩阵X2计算模块的设计思路；

矩阵X2 的计算公式如下：

其相应的输入/输出的函数关系式如下

a′ = a2+bc

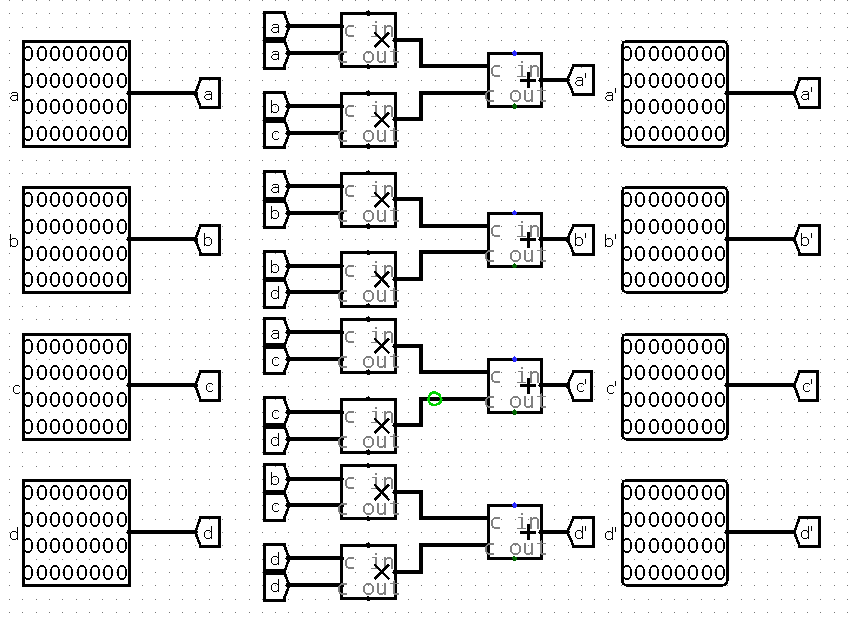
b′ = ab+bd

c′ = ac+cd

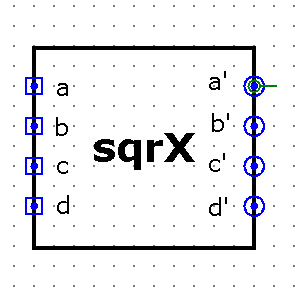
d′ = bc+d2

依次使用乘法器和加法器来实现电路

给出logisim软件绘制的电路图（经过仿真验证基本正确）；



对矩阵X2模块进行封装，给出封装后的模块图。



**（3）计算矩阵X2·A模块**

**要求：**

给出矩阵X2·A计算模块的设计思路；

其相应的输入/输出的函数关系式如下

a″ = ab+bd

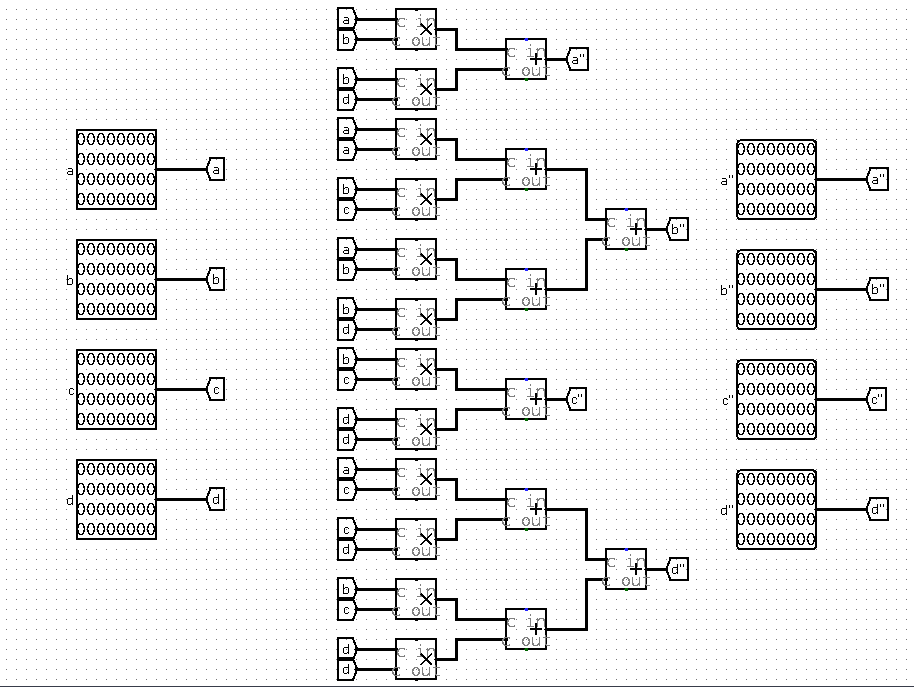
b″ = a2+bc+ab+bd

c″ = bc+d2

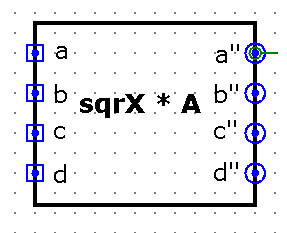
d″ = ac+cd+bc+d2

与上述模块一样，使用加法器和乘法器来实现

给出logisim软件绘制的电路图（经过仿真验证基本正确）；



对矩阵X2·A模块进行封装，给出封装后的模块图。



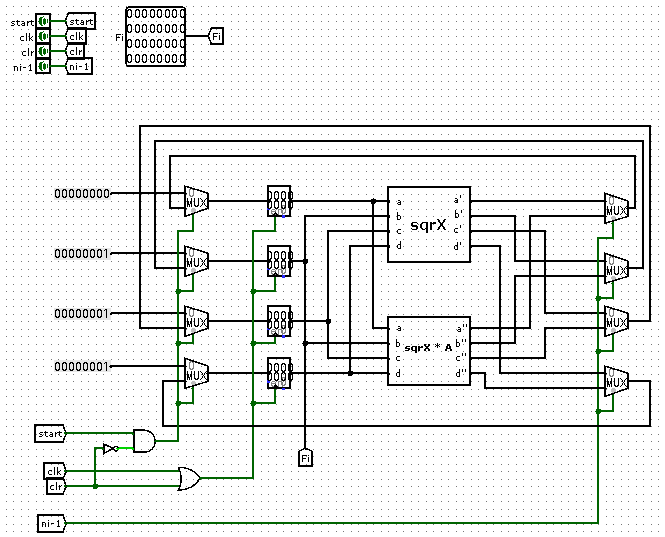
**（4）矩阵快速幂算法迭代模块设计**

**要求：**

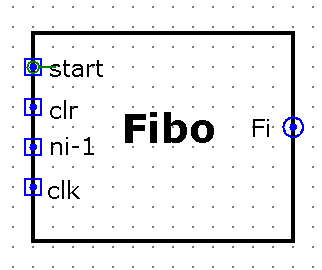
给出矩阵快速幂算法迭代模块设计思路；

start为Fibonacci()算法中的6位二进制数n左移出的第一个（最高位的）1的标志信号；ni-1是start=1之后左移出的下一位；clr为初始化（清零）信号，此时X = A；clk为时钟脉冲信号。Fi为Fibonacci()算法迭代的中间结果，根据ni-1取0或1来决定Fi是取sqrX或者sqrX\*A运算后的矩阵元素bi，在第6个时钟脉冲时，Fi即为输入n的Fibonacci数Fn。

给出logisim软件绘制的电路图（经过仿真验证基本正确）；



对矩阵X2·A模块进行封装，给出封装后的模块图。



**（5）主模块main设计**

**要求：**

说明主模块main中控制和显示部分的设计思路；

控制器Controller中包括三个功能块：6位二进制数n的左移控制电路、6个时钟脉冲控制电路、start信号产生电路。

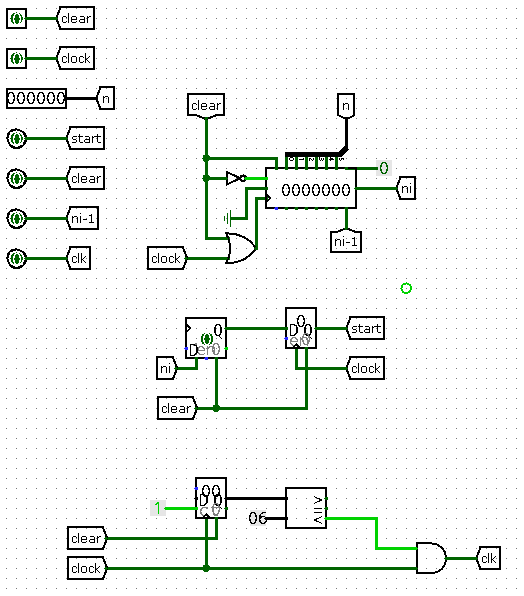
6位二进制数n的左移控制电路，使用一个移位寄存器，在时钟脉冲作用下产生ni-1。用clear信号装入n，进行移位寄存器的初始化。

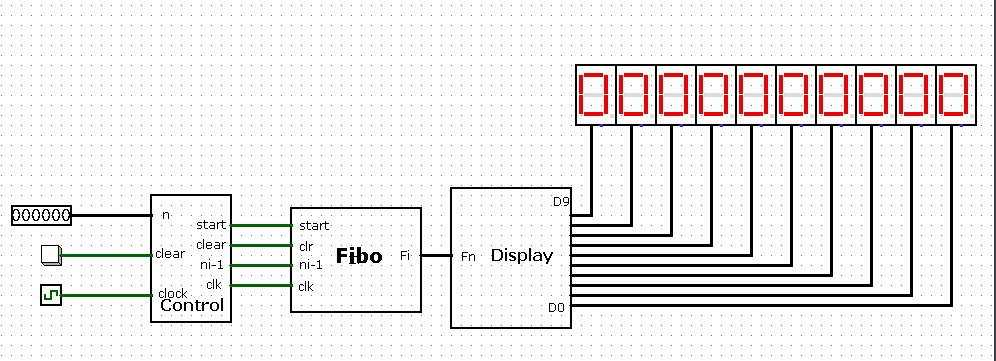
使用1个8位计数器、1个比较器和适当的门电路，可以控制Fibo只接收6个clock时钟脉冲（产生clk）。直至下一个clear信号初始化后，才准备产生下一组6个时钟脉冲。

使用1个D触发器加适当的门电路构成一个锁存器Latch，在接收到n的最高位1时start=1，直至下一个clear信号使start=0。

在6个clock时钟脉冲信号后，电路就产生了第n个Fibonacci数F(n)，并经过Display电路转换成十进制数在数码管上显示出来。

给出主模块的logisim软件绘制的电路图（经过仿真验证基本正确）。





6、实验结果记录

根据下表中所列内容，记录相应信号作用后输出数码管显示数据，并填入表1中（注：要求clear、clock使用按钮输入）。

表1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input n | clear | 1st clock | 2nd clock | 3rd clock | 4th clock | 5th clock | 6th clock | After  6th clock |
| 2 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 | 1 | 1 | 1 | 5 | 5 |
| 10 | 0 | 1 | 1 | 1 | 1 | 5 | 551 | 551 |
| 17 | 0 | 1 | 1 | 1 | 3 | 21 | 1597 | 1597 |
| 25 | 0 | 1 | 1 | 2 | 8 | 144 | 75025 | 75025 |
| 32 | 0 | 1 | 1 | 5 | 89 | 987 | 2178309 | 2178309 |
| 44 | 0 | 1 | 1 | 5 | 89 | 17711 | 701408733 | 701408733 |
| 45 | 0 | 1 | 1 | 5 | 89 | 17711 | 1134903170 | 1134903170 |
| 46 | 0 | 1 | 1 | 5 | 89 | 28657 | 1836311903 | 1836311903 |
| 47 | 0 | 1 | 1 | 5 | 89 | 28657 | 2971215073 | 2971215073 |

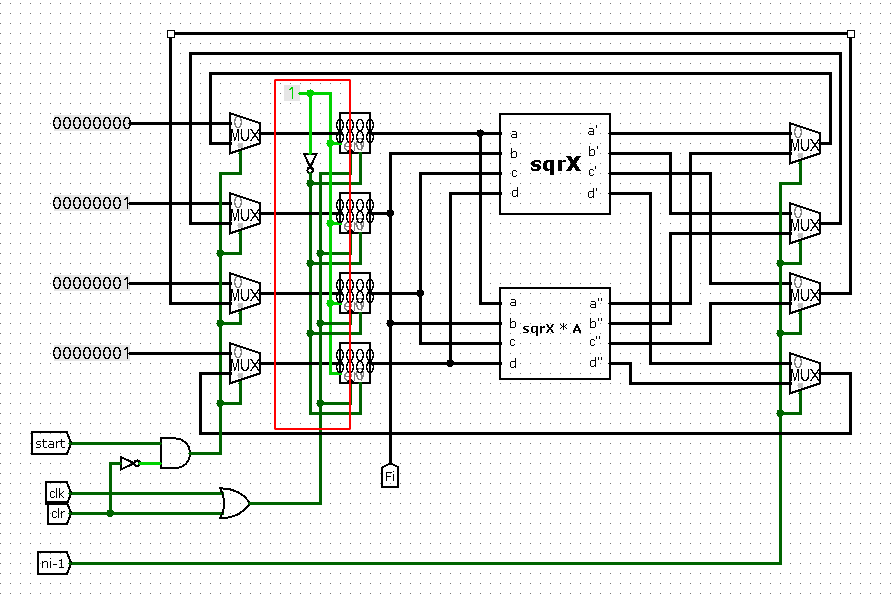
7、实验中遇到的问题及解决方法

（1）故障1

问题描述：最后检查的时候,整个模块不是很稳定,有时候是正常显示,有时候显示的是错误的

问题分析：有可能是某一些元器件的没有用到的输入没有给定一个常量,导致出现随机值,也有可能是因为使用了自动模拟的时钟,频率过快

解决方法：最终将电路中大部分的元器件的输入都用常量进行了填充

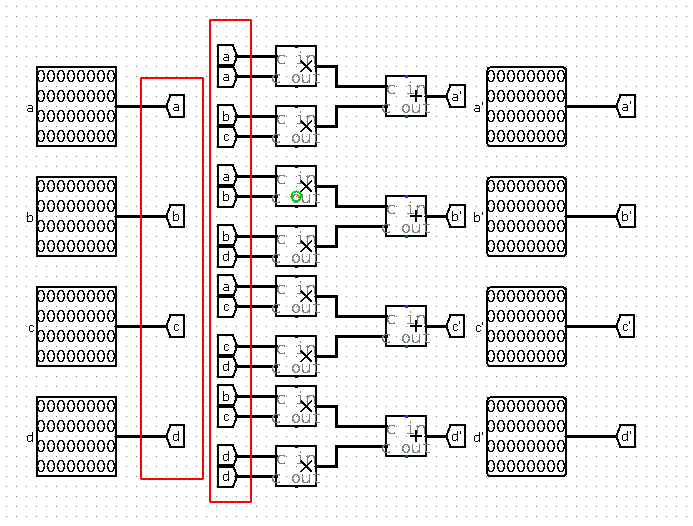


（2）故障2

问题描述：编写sqrtX 和 sqrtX \* A模块的时候,线路过于杂乱

问题分析：由于这两个模块就是不同的线路交错非常复杂,如果直接连线不可避免的会出现这种情况

解决方法：多使用通道,使整个电路(外观上)大大化简



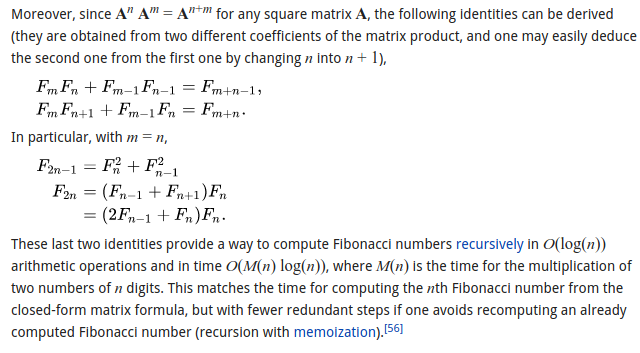
8、思考题

**（1）**如果n=48,电路的输出结果是多少？为什么？如果要求n=64时电路的输出结果也需正确，对你设计的电路应作哪些改进？

n=48时，电路的输出结果为512559680，此时,正确的结果已经超出了10位10进制数能够表示的范围,所以发生了溢出.

n = 64时,正常的输出为10610209857723,那么二进制位数只要需要被扩展到44位,否则就会发生溢出错误.

**（2）**还有没有其它时间复杂度为O(log2 n)的算法？如有，请描述该算法，并简要说明该算法的硬件实现思路。



其实本质上还是利用了分治法，使得整体的大问题可以变为两个小问题分开求解。硬件思路：由于输入就是2进制数，可以通过移位器多次位移得到二分过程中所需要算出的子问题，然后用同样的方法分析子问题，再使用减法器和乘法器，根据公式得到最终结果

**（3）**请谈谈对用硬件和用软件实现同一算法的优势和劣势。

硬件实现的话可以在短时间内得到较为真实的数据，执行时间也比较快，但是最开始的构造环节不可避免的会出现一些错误，再加上可能出现的后期的算法的改进，这就导致硬件实现的成本飞升。

1. 心得体会、意见与建议

这一次实验主要难度在于control模块的设计，由于其他的模块路大体都已经给出，只需要按照所给的思路画出电路图即可。这次实验是所有试验中第一次感觉到和计算机软件或者说算法知识相关的，一是因为实验的主题就是我们初学编程时都会遇到的斐波那契数列的计算问题，二是因为在思考题中还出现了时间复杂度的分析与计算，让人觉得很亲切。同时，这一次试验中多个地方引出的对于斐波那契数列的多种计算方法。也让我认识到对于某一种特定的问题，硬件实现的方法多种多样，并且常常与软件实现相比也就是写代码的方式速度更快。