

# **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： 王占成**

**学 号： U201714786**

**班 级： CS1707**

**指 导 教 师： 胡迎松**

**计算机科学与技术学院**

**2019年6月5日**



**数字逻辑实验报告**

斐波那契(Fibonacci)数列计算器设计

## 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所示：

**（1）求Fibonacci数的矩阵算法**

　　首先，对于数列的初始条件对应以下的矩阵运算：

更一般化地，有：

所以，根据递推关系可以得到：

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

其相应的输入/输出如下：

sqrX

a

b

c

d

a′ = a2+bc

b′ = ab+bd

c′ = ac+cd

d′ = bc+d2

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

其相应的输入/输出如下：

sqrX\*A

a

b

c

d

a″ = ab+bd

b″ = a2+bc+ab+bd

c″ = bc+d2

d″ = ac+cd+bc+d2

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/输出端如下：

Fibo

start

clr

ni-1

clk

Fi = bi′ or bi″

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4-1所示。



图4-1 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4-2所示。



图4-2 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

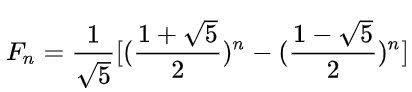
**（1）斐波那契(Fibonacci)数列计算**

**要求：**

给出Fibonacci数列通项公式；

给出Fibonacci数列的递归算法（指数时间复杂度）形式化描述；

给出Fibonacci数列的多项式时间复杂度算法形式化描述。

1. Fibonacci数列通项公式；
2. Fibonacci数列的递归算法（指数时间复杂度）形式化描述：T(n+1)=T(n)+T(n-1)=2\*T(n-1)+T(n-2)=…=F(n)+F(n-1)=F(n+1)

由于直接递归调用,结果中的每一个1都来自最底层的F(1)和F(2),那么为了求第n个数,就要调用F(n)次函数.由于斐波那契数列是指数增长,所以该算法的时间复杂度也是指数增长,即O(2^n)；

伪代码描述；

Fibo (n):

if n <= 1: *// 0,1 直接返回*

return n

result = np.mat([[1,1],[1,0]])\*\*(n-1) \* np.mat([1,0])

return (result[0,0])

1. Fibonacci数列的多项式时间复杂度算法形式化描述：

Fibonacci(){

int fbnq(int n){

int a = 1;

int b = 1;

int c = a;

while (n > 2)

{

c = a + b;//从第三个数开始，斐波那契数等于前两个数的和；

a = b;//将前一个数给到a，开始下一次求值

b = c;//将斐波那契数给b，开始下一次求值

n--;//每求一次，n都要减一

}

return c;

}return(X);

}

**（2）计算矩阵X2模块**

**要求：**

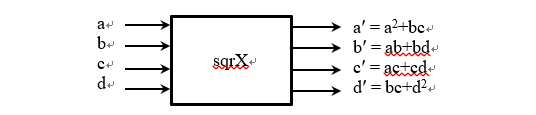
给出矩阵X2计算模块的设计思路；

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。

1. 矩阵X2计算模块的设计思路：

其相应的输入/输出如下：



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

1. logisim软件绘制的电路图；如图4-1矩阵X2计算模块；

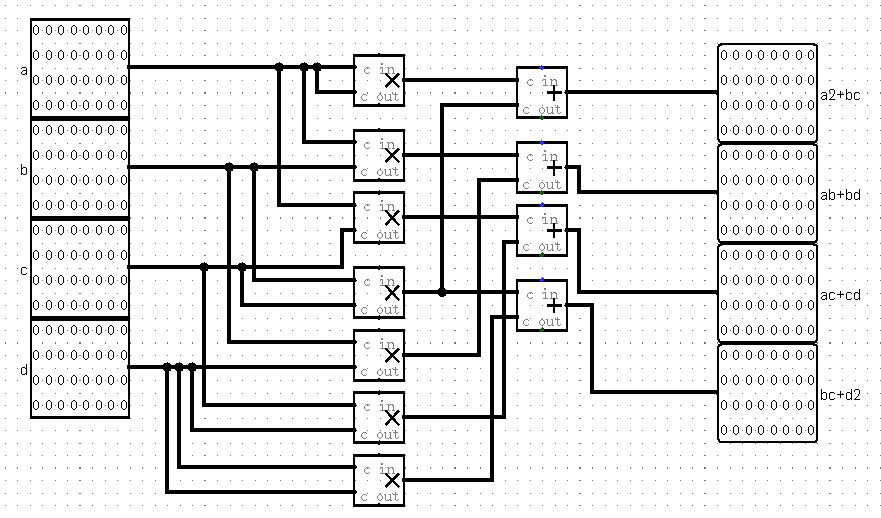


图4-1矩阵X2计算模块内部结构

1. 对矩阵X2模块进行封装，给出封装后的模块图；

如图4-2矩阵X2计算模块封装；

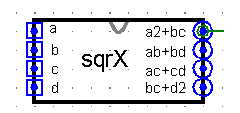


图4-2矩阵X2计算模块封装

**（3）计算矩阵X2·A模块**

**要求：**

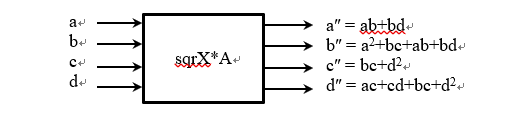
给出矩阵X2·A计算模块的设计思路；

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。

1. 矩阵X2·A计算模块的设计思路

其相应的输入/输出如下：



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

1. logisim软件绘制的电路图；如图4-3矩阵X2·A计算内部结构模块；

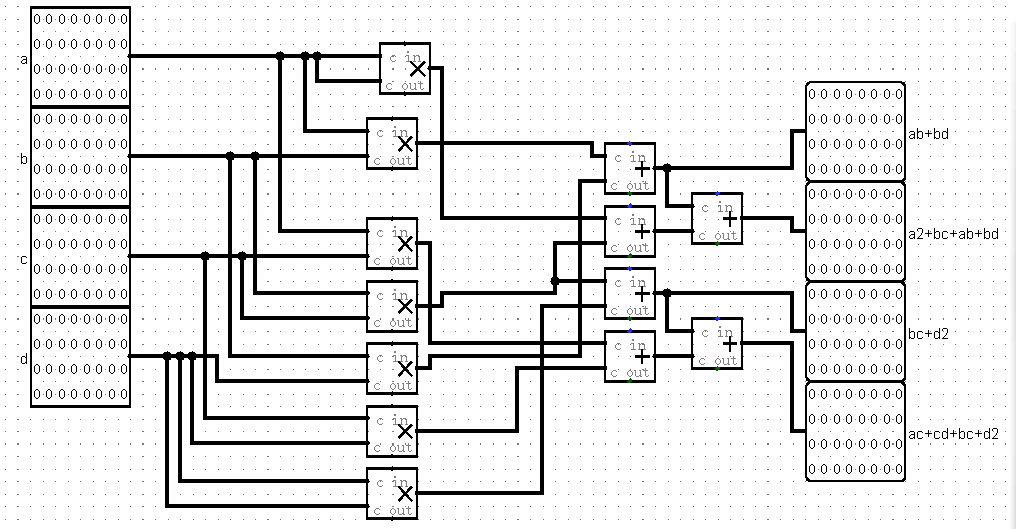


图4-3矩阵X2·A计算内部结构模块

1. 矩阵X2·A模块进行封装，如图4-4矩阵X2·A计算模块封装；

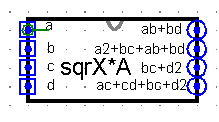


图4-4矩阵X2·A计算模块封装

**（4）矩阵快速幂算法迭代模块设计**

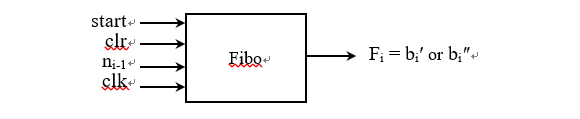
**要求：**

给出矩阵快速幂算法迭代模块设计思路；

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。

1. 矩阵快速幂算法迭代模块设计思路：该模块Fibo输入/输出端如下：



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。

1. logisim软件绘制的电路图，如图4-5所示；

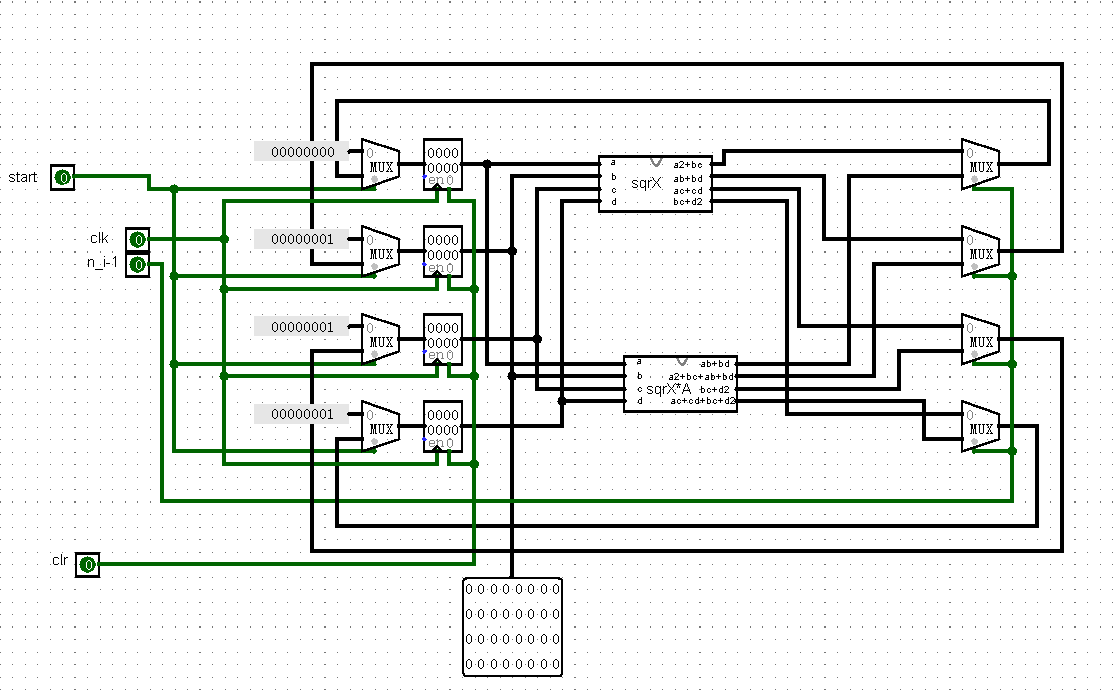


图4-5矩阵快速幂算法迭代模块

1. 矩阵X2·A模块进行封装，如图4-6矩阵快速幂算法迭代模块封装；

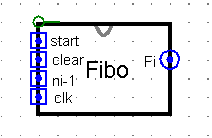


图4-6矩阵快速幂算法迭代模块封装

**（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；

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）。

1. main中控制部分设计思路；

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。

控制电路内部结构如图4-7所示；

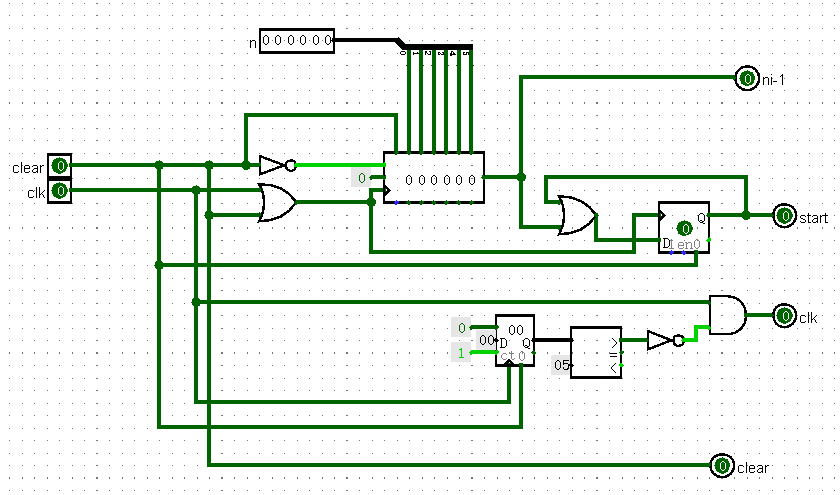


图4-7控制部分内部结构

其封装如图4-8所示；

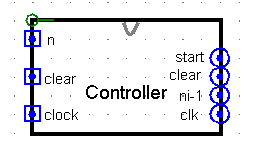


图4-8控制部分封装电路图

main中显示模块设计的思路：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，采用模十取余法，依次取出D9、D8、D7、D6、D5、D4、D3、D2、D1、D0；

控制部分内部电路结构如图4-9所示；

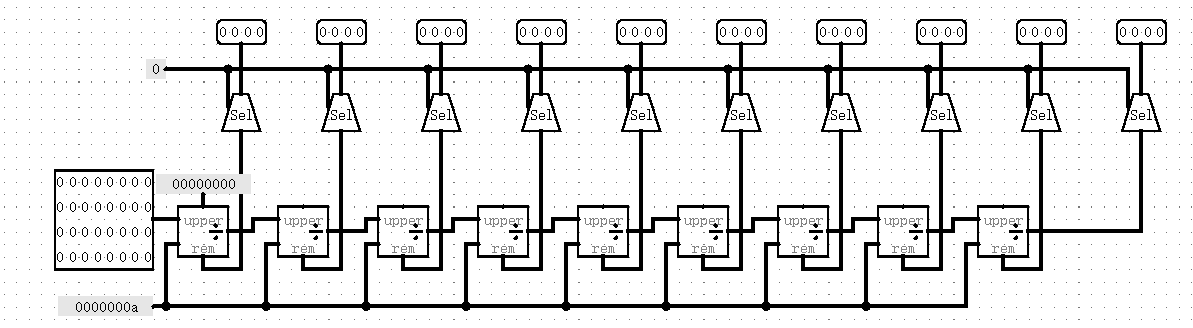


图4-9 数字显示部分电路

其封装如图4-8所示；

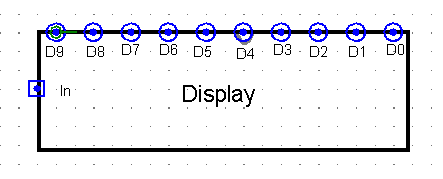


图4-10 数字显示部分封装电路

1. 主模块的logisim软件绘制的电路图，将各个模块组合得图4-13所示主模块电路；

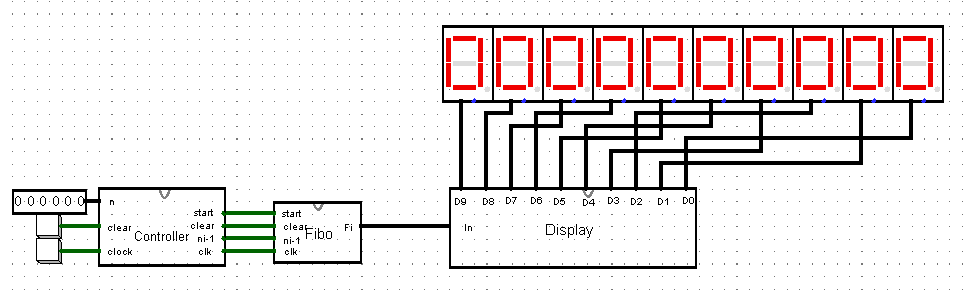


图4-11 main模块绘制的电路图

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6-1中（注：要求clear、clock使用按钮输入）。

表6-1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 1 | 5 |
| 10 | 0 | 1 | 1 | 1 | 1 | 1 | 5 | 55 |
| 17 | 0 | 1 | 1 | 1 | 1 | 3 | 2 | 1597 |
| 25 | 0 | 1 | 1 | 1 | 2 | 8 | 144 | 75025 |
| 32 | 0 | 1 | 1 | 1 | 3 | 21 | 987 | 2178309 |
| 44 | 0 | 1 | 1 | 1 | 5 | 89 | 17711 | 701408733 |
| 45 | 0 | 1 | 1 | 1 | 5 | 89 | 17711 | 1134903170 |
| 46 | 0 | 1 | 1 | 1 | 5 | 89 | 28657 | 1836311903 |
| 47 | 0 | 1 | 1 | 1 | 5 | 89 | 28657 | 2971215073 |

7、实验中遇到的问题及解决方法

（1）故障1

问题描述：显示电路输出的是乱码。

问题分析：显示电路的高位和低位弄反，导致电路显示混乱。

解决方法：第一次与10相除时，得到的余是最低位。在设计电路外观时，只需要将各个输出的位置调换即可。

（2）故障2

问题描述：时钟输入的时候，经过控制部分输出时，会多产生一次脉冲。

问题分析：当与门两个输入端的输入值相反，并且同时发生变换时，会产生毛刺现象。

解决方法：将毛刺现象作为一次脉冲输出，将控制部分比较的数值改为5即可。实例如下：

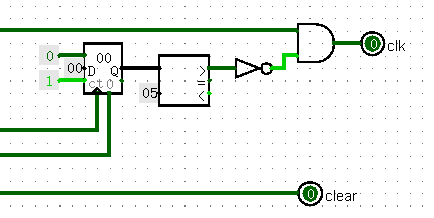


图4-12 问题解决实例

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？

1. n=48512559680；电路显示结果为512559680；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,发生了溢出。

2. 或许可以通过修改位宽，通过多次移位实现

**（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。

没有算法复杂度为O(log2 n)的算法，但有时间复杂度交题目低的迭代算法和时间复杂度为1的直接通过斐波拉契数列的通项公式求解；

**（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：快，但是成本高、耗电、占用空间

软件实现：灵活，成本低，但是慢。

9、心得体会、意见与建议

通过本次的实验，对硬件的综合设计能力有了一定的提升。+在设计电路时，有些问题不要想的过于复杂，想的复杂了设计的电路会更加复杂和难懂。在设计电路之前，不要盲目地开始，在纸上写出了一定的设计思路之后，再开始进行设计。正所谓磨刀不误砍柴工。

这也是数字模拟电路设计的最后一次实验课，真心地感谢胡老师对我的耐心指导和同学们对我的帮助。谢谢胡老师！