

# **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： 聂鸿勇**

**学 号： U201714785**

**班 级： CS1707**

**指 导 教 师： 胡迎松**

**计算机科学与技术学院**

**2019年 6月 6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）、



F0=0

F1=1

FN=FN−1+FN−2

（2）、int Fib(int n)

{

if (n == 0)

{

return 0;

}

if (n == 1)

{

return 1;

}

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

}

（3）、F(n)=F(n-1)+F(n-2)，利用递归思想，每次计算当前的值时候，就要引用之前的两个值，一步一步的递归，一直到最起始处，直至F(1)和F(2)。记计算第n个数的所需时间为T(n)，那么T(n) = T(n-1) + T(n-2) ,由T(n-1) > T(n-2) 可以推出T(n) < 2T(n-1) < 2^2 \* T(n-2)<2^(n-1)T(1)，可以推出T(n)的时间复杂度为O(2^n)。

**（2）计算矩阵X2模块**

矩阵X2 的计算公式如下：

a′ = a2+bc

b′ = ab+bd

c′ = ac+cd

d′ = bc+d2

根据公式可得到输入与输出的关系，利用乘法器和加法器构建出对应电路图，如图4-3所示。

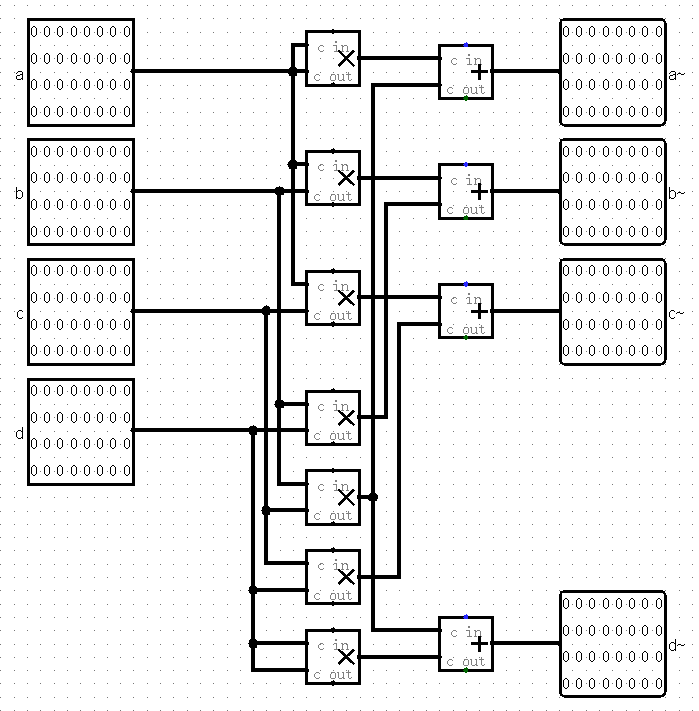


图4-3 矩阵X2设计电路

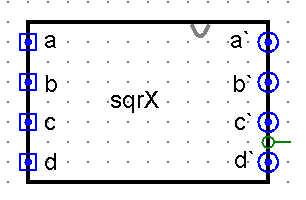


图4-4矩阵X2设计电路封装

**（3）计算矩阵X2·A模块**

矩阵X2·A 的计算公式如下：

a″ = ab+bd

b″ = a2+bc+ab+bd

c″ = bc+d2

d″ = ac+cd+bc+d2

根据公式可得到输入与输出的关系，利用乘法器和加法器构建出对应电路图，如图4-5所示。

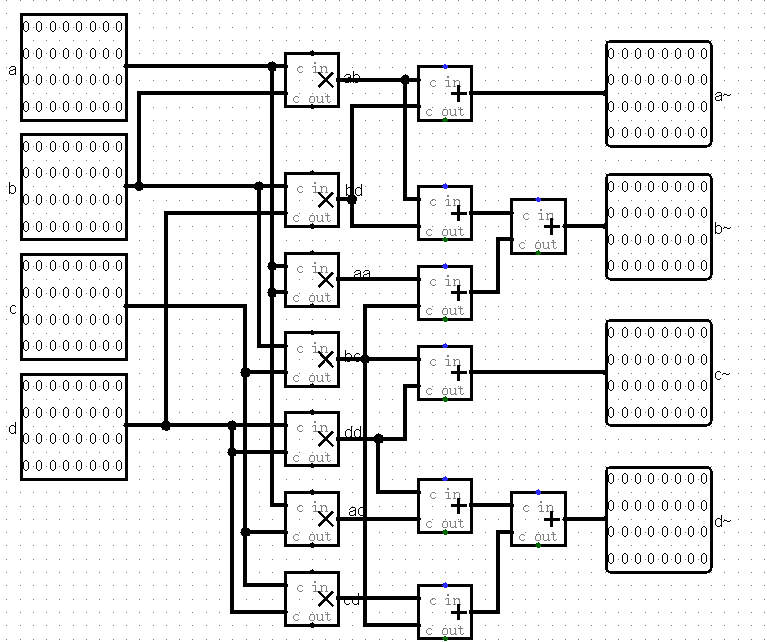


图4-5 矩阵设计电路

**（4）矩阵快速幂算法迭代模块设计**

在矩阵幂运算迭代算法中需要设计一个存储器来存储当前的计算结果和下一次迭代的运算结果，并且通过脉冲更新存储器中的内容；在运算过程中需要根据来判断是进行X2运算还是X2·A运算，通过一个选择器来选择将哪一数据存入存储器中。

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。使用logisim绘制电路图如图4-6所示。

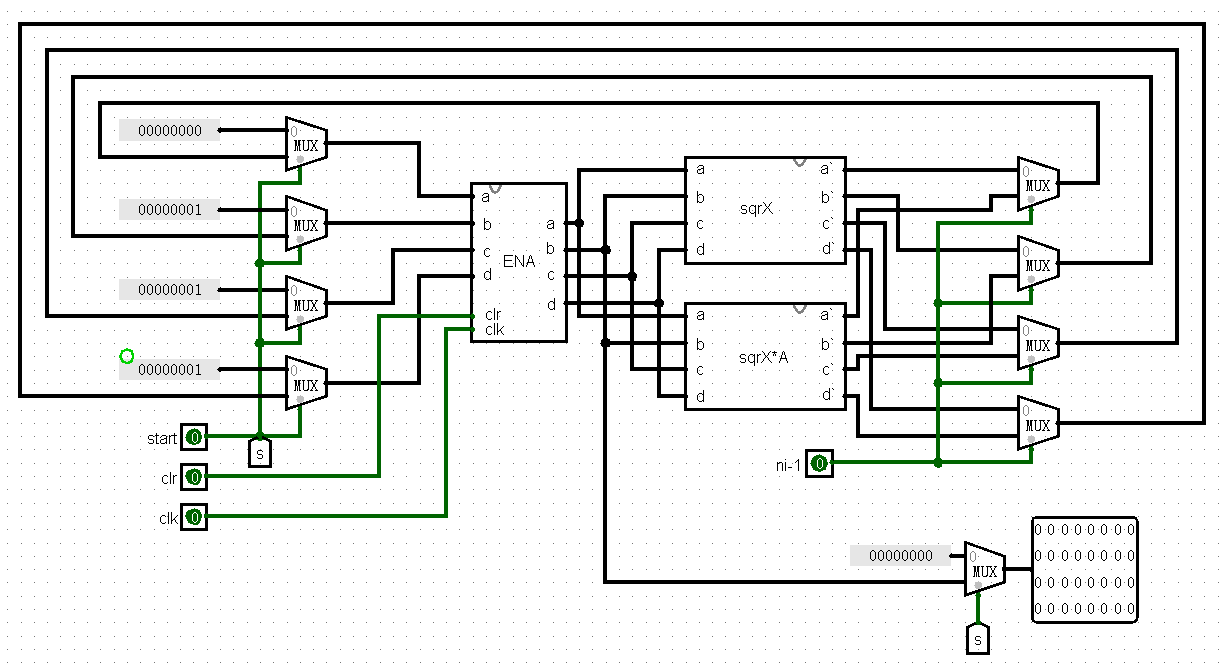


图4-6 Fibo模块电路设计

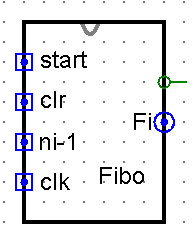


图4-7 Fibo模块电路设计

**（5）主模块main设计**

（1）、显示器Display的电路图由10个除法器构成，输入计算所得的结果，对其进行对十求模，再对其所得余求模重复十次。将十次的求模结果输出。

（2）、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。如图4-8所示。

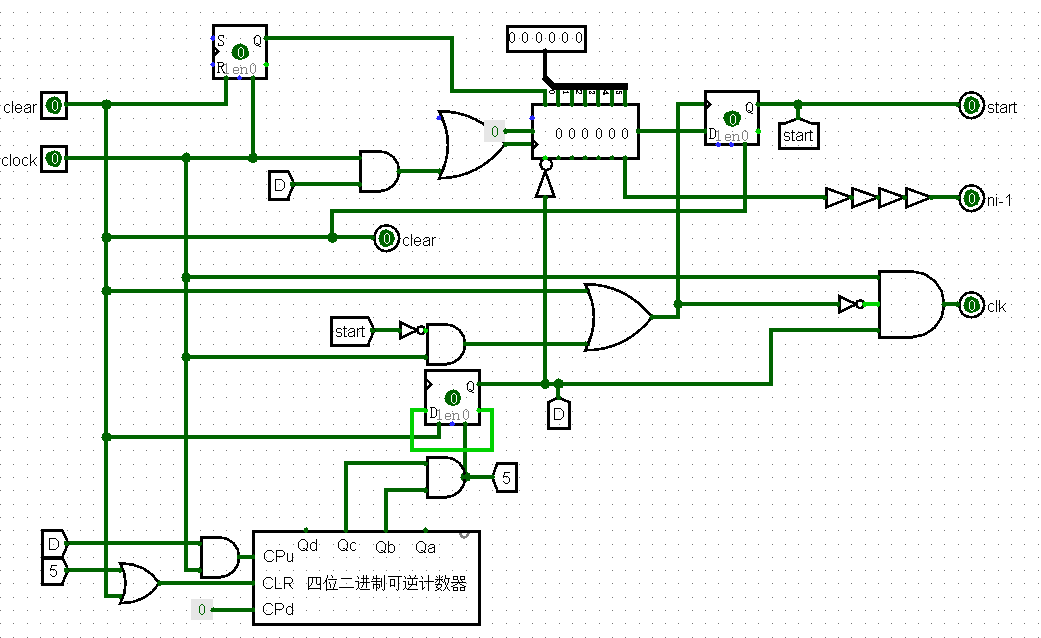


图4-8 主模块设计电路

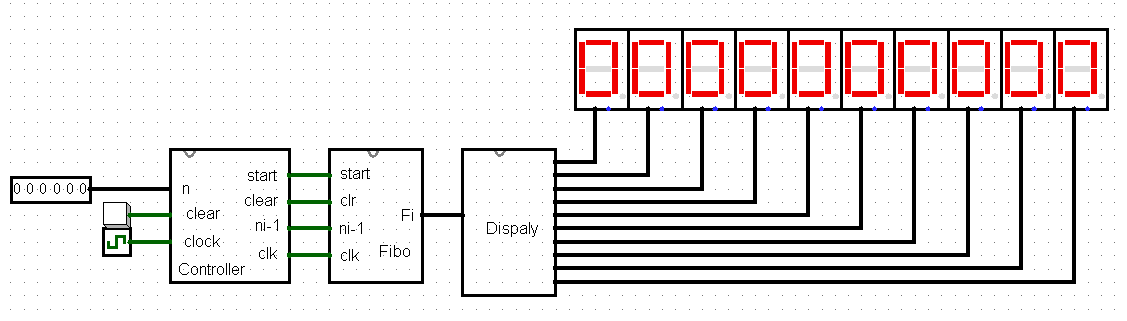


图4-9 斐波拉契数列计算封装图

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 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 0 | 0 | 0 | 1 | 1 | 5 | 5 |
| 10 | 0 | 0 | 0 | 1 | 1 | 5 | 55 | 55 |
| 17 | 0 | 0 | 1 | 1 | 3 | 21 | 1597 | 1597 |
| 25 | 0 | 0 | 1 | 2 | 8 | 144 | 75025 | 75025 |
| 32 | 1 | 1 | 1 | 3 | 21 | 987 | 2178309 | 2178309 |
| 44 | 1 | 1 | 1 | 5 | 89 | 17711 | 701408733 | 701408733 |
| 45 | 1 | 1 | 1 | 5 | 89 | 17711 | 1134903170 | 1134903170 |
| 46 | 1 | 1 | 1 | 5 | 89 | 28657 | 1836311903 | 1836311903 |
| 47 | 1 | 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：在最初设计好控制器Controller的时候，cleck点击五下即可得到最终结果。

问题分析：在clear的时候需要一次脉冲来装入n，进而将clear也接在了移位器的脉冲端，点击clear的时候相当于已经点击了一次cleck，所以只需五次即可得到结果 。

解决方法： 利用一个锁存器来控制clear输入。最终修改如图4-10所示：

（2）故障2

问题描述：每次运算的时候结果都不相同

问题分析：在设计控制器Controller的时候，其中的D触发器有些是上升沿有些是下降沿，导致结果有误。

解决方法：将所有D触发器改为下降沿。

（2）故障3

问题描述：在进行完一次运算后，再次计算结果有误。

问题分析：在进行完一次运算后，start无法变回0状态。

解决方法：控制移位器在运算结束后清零，并且连接start的D触发器也受clear控制清零。

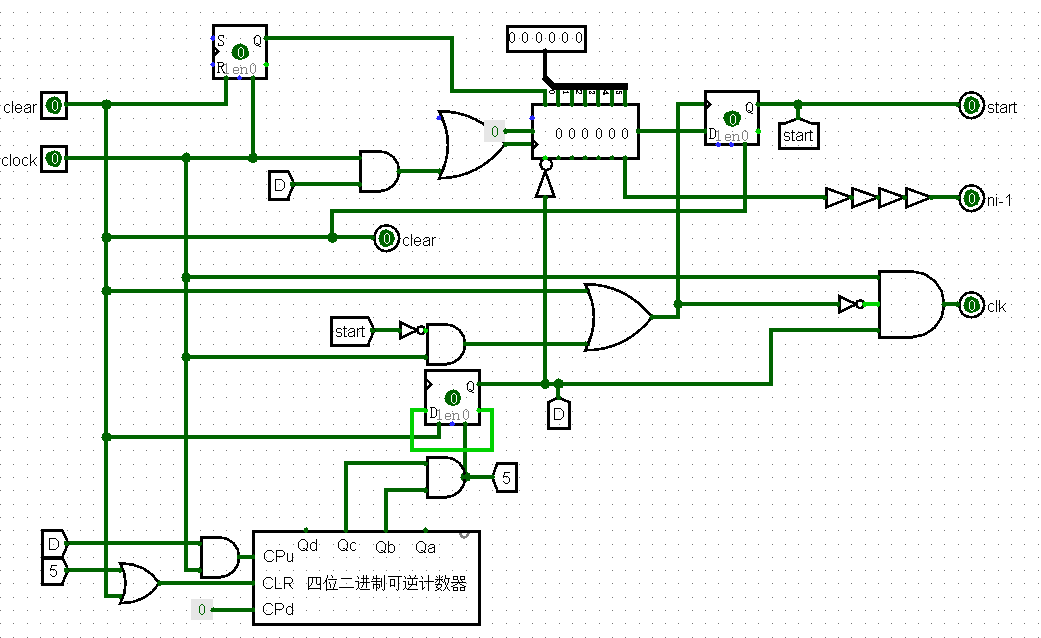


图4-10修改后的控制器Controller

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？

答：48对应的斐波那契数值为4807526976，而使用该电路计算输入48，输出为0512559680，与实际结果不符，因为48对应的结果超过了32位所能表示的最大值。

若要使n=64时输出也正确需要将32位的输入输出位数改为位宽更大的，其计算次数也要增加。

**（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。

答：无

**（3）**请谈谈对用硬件和用软件实现同一算法的优势和劣势。

答：通过软件实现的算法更加具有可读性，在复杂度方面可以不断的优化，而使用硬件实现算法时需要考虑器件使用多少的问题，因而在设计过程中需要尽量使用简单的运算，可读性比较低。

9、心得体会、意见与建议

本次实验在给出算法的基础上，理解了算法之后电路的设计就变的比较简单。

跟着文档的步骤一步一步做下来能够在该过程中不断熟悉了解该算法，在最后的控制器设计阶段将之前的设计综合起来完成本次实验的斐波拉契数列计算器，虽然过程中有一些问题，但最后都被解决了。

总的来说，本次数电实验课程让我收获很多。我会在今后的学习中更加努力。 最后，感谢老师一个学期以来的教导。