实验六：斐波那契(Fibonacci)数列计算器设计

1. 给出Fibonacci数列通项公式、Fibonacci数列的递归算法（指数时间复杂度）形式化描述、Fibonacci数列的多项式时间复杂度算法形式化描述；

数列通项公式：

递归算法的形式化描述：

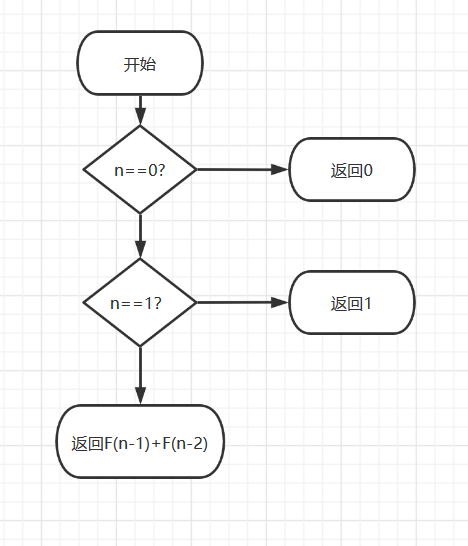


图6-1 递归算法形式话描述

时间复杂度为：求解通项公式得到时间复杂度为 ，即O（2^n）

数列多项式伪代码：

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);

}

时间复杂度：O(n)

2. 说明斐波那契(Fibonacci)数列计算器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） 控制部分：

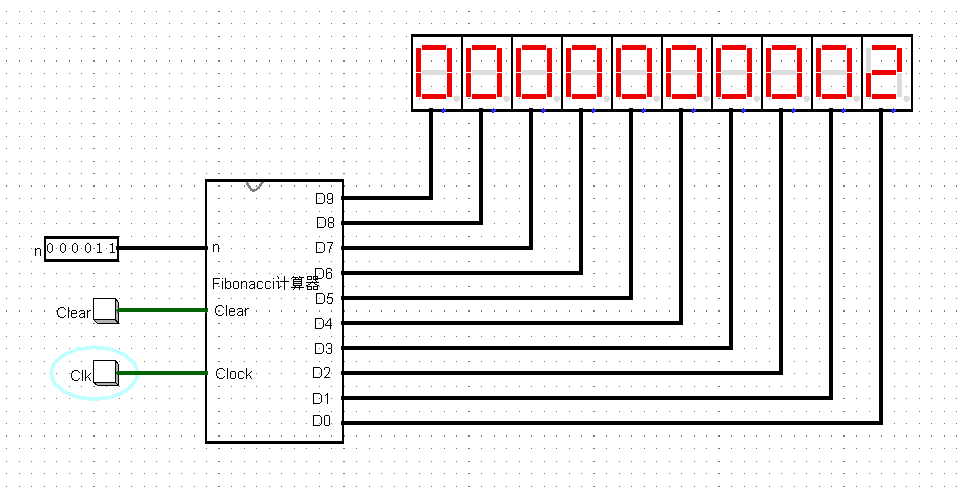
左移控制部分：采用一个移位寄存器，寄存器的load端接clear实现输入n的初始化，在一开始的尝试中发现，该移位寄存器需要在时钟的上升沿时加载端为高电平才可以实现加载，所以在这里采用clear和clock或的方式接入寄存器的时钟端，从而只要按下clear键即可加载，移位寄存器的右侧输出端即为ni-1。

clk控制电路：为了实现fibo模块只接受六个clock脉冲，采用一个8位计数器和一个比较器，8位计数器的时钟端接clock，清零端接clear，计数端接常量0x1，其输出接至比较器的一个输入端，比较器的另外一个输入端接0x5（接6的话，会导致之后合并控制模块和fibo模块时，计时输入六个clock脉冲之后再输入clock输出的clk电平为0，但是还是可以触发fibo的寄存器更新数据，所以接5，而且发现和5比较与和6比较对最终的计算结果没有影响）。比较从小于号取出输出，再取非之后再和clock相与作为clk的输出，其工作过程大致是：当clock计数小于等于5的时候，比较部分输出为0，clk的输出波形和clock的输入波形相同。当clock输入到第六个脉冲及以上的时候，比较部分的输出为1，与门结果为0，clk始终为低电平，这样即可保证输入到fibo模块的时钟端仅有6个脉冲。

锁存器：为了实现start在接收到n的最高位标志信号时变为1，直到再一次clear脉冲到来，start变为0，采用一个D触发器和一个或门。移位寄存器的输出端口和start相或作为D触发器的D输入端，clock作为时钟端，clear作为清零端，触发器的输出即为要输出的start。初始时start为0，所以第一次向左移位通过一个或门之后，如果n的最高位为1，start即为1，否则为0，而当start变为1之后，触发器而定D输出端始终为1，所以输出始终为1，即start保持1不变知道clear是触发器置零。

ni-1：观测移位寄存器我们不难得出ni-1就是移位寄存器的右侧输出值。

1. 显示部分：显示部分要求将一个32位的二进制数转化为10个8421BCD码，可以采用模十取余的方法一次获得这个32位二进制数的每一位的数值，最高位商，其余9位取余数。
2. 测试电路图：



**图6-1 测试结果图**

3. 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 实验结果记录表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 0 | 1 | 1 | 3 | 21 | 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 |