实验六：斐波那契(Fibonacci)数列计算器设计

1．实验方案设计

**具体要求：**

（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；

通项公式：通项公式如图6.1所示。

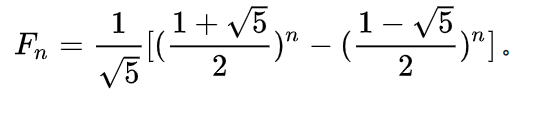


图6.1 通项公式

递归算法形式化描述：如果n等于0，则Fn为0；如果n等于1，则Fn为1；否则Fn=Fn-1+Fn-2。

int fb(int n)

{

if (n<=0)

return 0;

if (n==1)

return 1;

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

}

多项式时间复杂度算法形式化描述：

int non\_recursive\_method(int n)

{

int p = 0;

int q = 1;

int tmp；

if (n == 0)

return p;

if (n == 1)

return q;

else{

for(int i = 3; i < n; i++){

tmp = p;//将第一个值p赋给tmp

p = q; //将第二个值q赋给p,以后每一次赋值都将得到的最新的F(n)赋给p,从后面语句可//以看出,q储存的为最新的F(n)

q = tmp + q;

}

return q;

}

}

（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）。

控制部分设计思路：

将n的最高位送入移位寄存器的最右侧，其余依次，直到最低位送入移位寄存器的最左侧。Clear信号接到加载端，~clear接入移位端。时钟端接入clock与clear相或，保证clear可以进行移位寄存器的初始化。

使用计数器计算时钟脉冲，将结果与6比较，clk为该结果与clock相与，若时钟脉冲小于6，clk才有效。

D触发器时钟端接clock，清零端接clear，输入端D为n当前最高位与结果Q相或，在接收到n的最高位1之后start一直为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)。

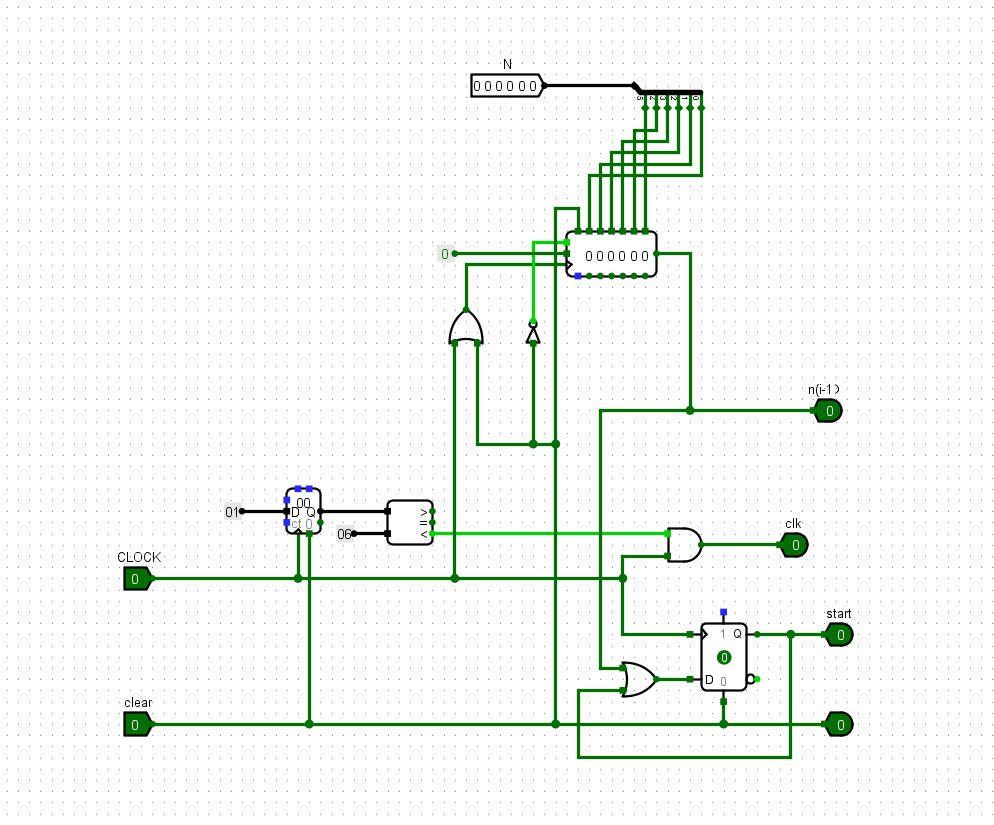


图6.2 控制电路

显示部分设计思路：

将32位二进制数除以1000000000，得到的商为十进制的最高位，余数为下一次的被除数；用被除数除以100000000，得到的商为十进制的下一位位，余数为下一次的被除数；以此类推，依次除以10000000，1000000，100000，10000，1000，100，10，1，得到最后的结果。

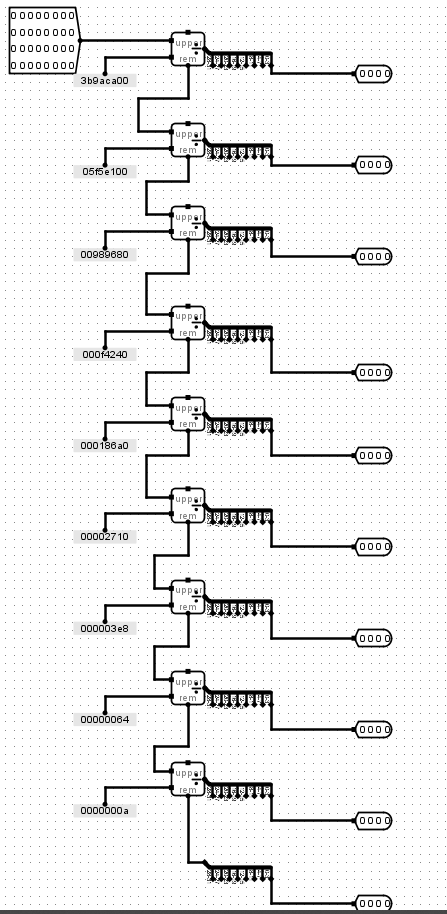


图6.3 显示电路

主模块电路图：主模块电路图如图6.4所示。

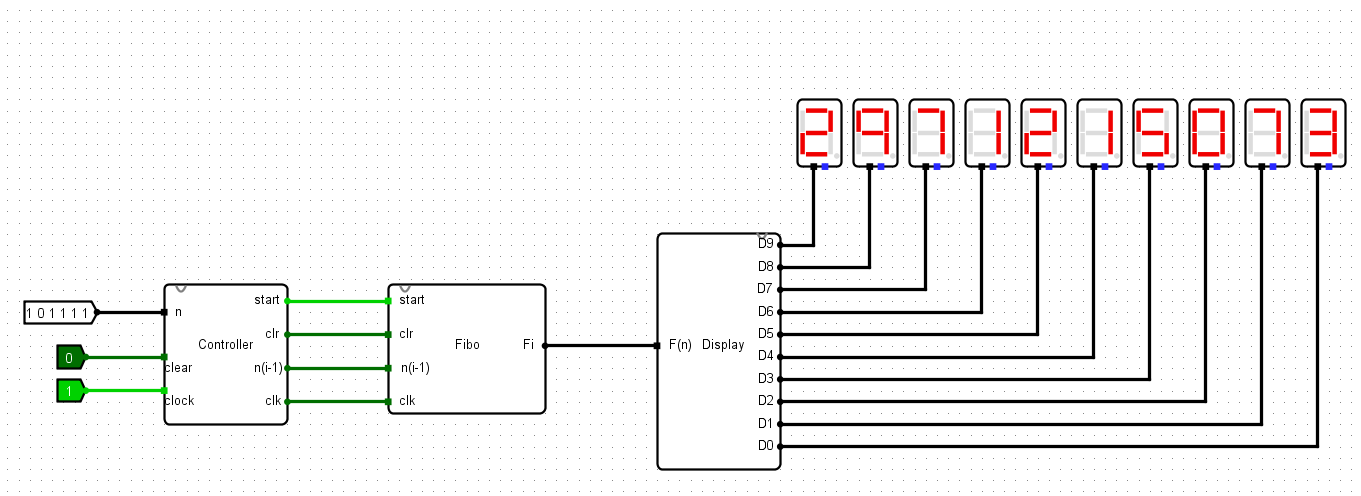


图6.4 主模块电路图

2．实验结果记录

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 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 5 | 0 | 1 | 1 | 1 | 1 | 1 | 5 | 5 |
| 10 | 0 | 1 | 1 | 1 | 1 | 5 | 55 | 55 |
| 17 | 0 | 1 | 1 | 1 | 3 | 21 | 1597 | 1597 |
| 25 | 0 | 1 | 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 |