

实验六：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 1 套，微型计算机 1 台。

4. 课时

课内 8 个课时，课外 8 个课时。

5. 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。

$$F_n = \begin{cases} 0, & n = 0 \\ 1, & n = 1 \\ F_{n-1} + F_{n-2}, & n > 1 \end{cases} \quad (\text{公式 1})$$

(1) 求 Fibonacci 数的矩阵算法

对于数列的初始条件对应公式 2 的矩阵运算：

$$\begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} \quad (\text{公式 2})$$

更一般化地，有公式 3：

$$\begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} F_{n-2} \\ F_{n-1} \end{pmatrix} \quad (\text{公式 3})$$

根据递推关系可以得到公式 4：

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a_n & b_n \\ c_n & d_n \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b_n \\ d_n \end{pmatrix} \quad (\text{公式 4})$$

由公式 4 可推出， $F_n = b_n$ 。

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

$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ 求 n 次幂。采用矩阵的快速幂算法，操作次数可优化为 $O(\log_2 n)$ 。

由于 $F(47)=(2971215073)_{10} < 2^{32}$, $F(48)=(4807526976)_{10} > 2^{32}$, 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 $2 \leq n \leq 47$, n 用 6 位二进制数表示。

(2) 算法描述

Fibonacci(){

初始化: $X = A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, Start=0;

For (i=5 downto 0)

{

if (Start==0) then

{

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

}

Else

{

if ($n[i]==1$)

then $X=X^2 \cdot A$;

else

$X=X^2$; }

}

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 $X = X^2 = A^2$;

step3: $i=3$, Start=1, $n[3]=1$, 此时 $X = X^2 \cdot A = (A^2)^2 \cdot A$;

step4: $i=2$, Start=1, $n[2]=1$, 此时 $X = X^2 \cdot A = ((A^2)^2 \cdot A)^2 \cdot A$;

step5: $i=1$, Start=1, $n[1]=0$, 此时 $X = X^2 = (((A^2)^2 \cdot A)^2 \cdot A)^2$;

step6: $i=0$, Start=1, $n[0]=0$, 此时 $X = X^2 = (((((A^2)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2$;

循环执行完后, $X = (((((A^2)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2 \cdot A)^2 = A^{44}$

(3) 矩阵计算模块

(a) 计算 X^2 模块 sqrX

$$X^2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^2 = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{pmatrix} \quad (\text{公式 5})$$

其相应的输入/输出如图 6.1 所示。

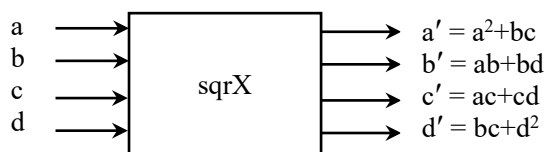


图 6.1 计算 X^2 模块 sqrX 输入/输出示意图

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

(b) 计算 $X^2 \cdot A$ 模块 $\text{sqrX} \cdot A$

$$X^2 \cdot A = \begin{pmatrix} a^2 + bc & ab + bd \\ ac + cd & bc + d^2 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} ab + bd & a^2 + bc + ab + bd \\ bc + d^2 & ac + cd + bc + d^2 \end{pmatrix} \quad (\text{公式 6})$$

其相应的输入/输出如图 6.2 所示。

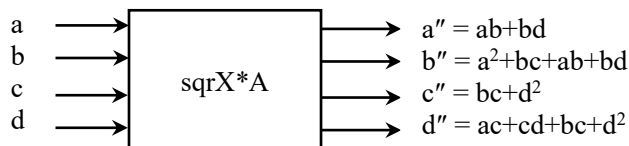


图 6.2 计算 $X^2 \cdot A$ 模块 $\text{sqrX} \cdot A$ 输入/输出示意图

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 6.3 所示。

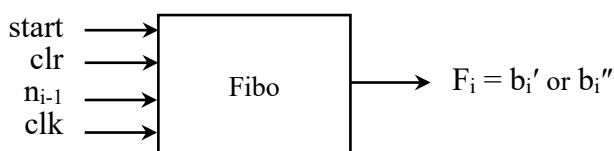


图 6.3 Fibo 输入/输出示意图

这里， start 为 $\text{Fibonacci}()$ 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； n_{i-1} 是 $\text{start}=1$ 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。 F_i 为 $\text{Fibonacci}()$ 算法迭代的中间结果，根据 n_{i-1} 取 0 或 1 来决定 F_i 是取 sqrX 或者 $\text{sqrX} \cdot A$ 运算后的矩阵元素 b_i ，在第 6 个时钟脉冲时， F_i 即为输入 n 的 Fibonacci 数 F_n 。

其内部逻辑结构图如图 6.4 所示。

(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 $F(n)$ 。

The diagram illustrates a 4-bit multiplier circuit. It includes a 2-to-1 MUX at the input, a 4-bit register (寄存器组) with inputs A, D, and ENB, and two 4-bit multipliers (sqrX and sqrX*A). The MUX selects between the input $ni-1$ and the output of the first multiplier. The register stores the output of the MUX. The second multiplier takes the register output and A as inputs. The final output is selected by another 2-to-1 MUX between the output of the second multiplier and the output of the first multiplier.

(6) 斐波那契(Fibonacci)数列计算器

The diagram illustrates the system architecture. The **Controller** block manages the process, receiving **n**, **clear**, and **clock** as inputs. It outputs **start**, **clear**, **n_{i-1}** , and **clk** to the **Fibo** block. The **Fibo** block calculates the Fibonacci sequence and outputs **F(n)** to the **Display** block. The **Display** block consists of 10 LEDs, each labeled **D9** through **D0**, which show the binary representation of the result.

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。

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。

6. 实验方案设计

具体要求:

4 / 6

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;

(2) 给出矩阵 X^2 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 X^2 模块进行封装;

(3) 给出矩阵 $X^2 \cdot A$ 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 $X^2 \cdot A$ 模块进行封装;

(4) 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 $X^2 \cdot A$ 模块进行封装;

(5) 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)。

7. 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	1 st clock	2 nd clock	3 rd clock	4 th clock	5 th clock	6 th clock	After 6 th clock
2								
5								
10								
17								
25								
32								
44								
45								
46								
47								

8. 实验结果提交

要求: (1) 本次实验的全部电路都在同一个 Logisim 文件中, 子电路结构如图 6.6 所示;

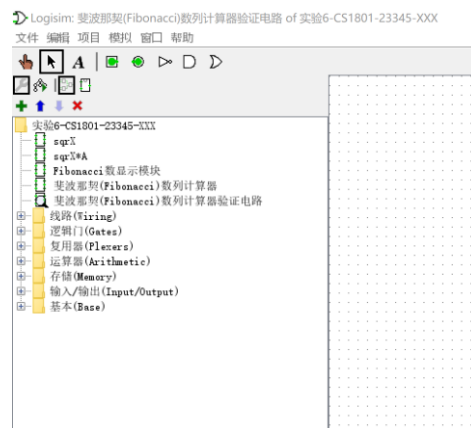


图 6.6 实验 6 子电路结构

- (2) 打印检查表并填写姓名等相关信息，实验验收完成后当堂提交。
- (3) 上交 Logisim 电路文件，命名格式：实验 6-班级-学号-姓名。