

实验三 时序电路设计实验报告

姓名	学号
班级	手机

1. 实验题目

某自动售货机出售四元一瓶的饮料，但是每次只能投入一元硬币。当连续投入四个硬币后，售货机才会给出一瓶饮料。如果投入的硬币不足四元，售货机不会退回之前所投入的硬币。假设输入 $X = 1$ 表示投入一元硬币， $X = 0$ 表示未投入一元硬币；输出 $Z = 1$ 表示售货机给出一瓶饮料， $Z = 0$ 表示未给出饮料。

2. 电路设计

a) 规范化

- 输入 X ：投硬币的操作，1 表示投入一元硬币，0 表示未投入硬币
- 输出 Z ：是否出饮料的判断，1 表示给出一瓶饮料，0 表示未给出饮料
- 电路行为：模拟自动售货机，对于投入硬币操作的时序行为，经过状态转移输出是否会出饮料的判断，标准为是否能在输入序列中识别到目标序列 1111 的出现。

b) 形式化

该电路共有四个状态：

- S_0 ：初始状态（连续投入硬币数为 0）
- S_1 ：连续投入 1 个硬币
- S_2 ：连续投入 2 个硬币
- S_3 ：连续投入 3 个硬币
- S_4 ：连续投入 4 个硬币

采用状态机图，对电路行为建模如下：

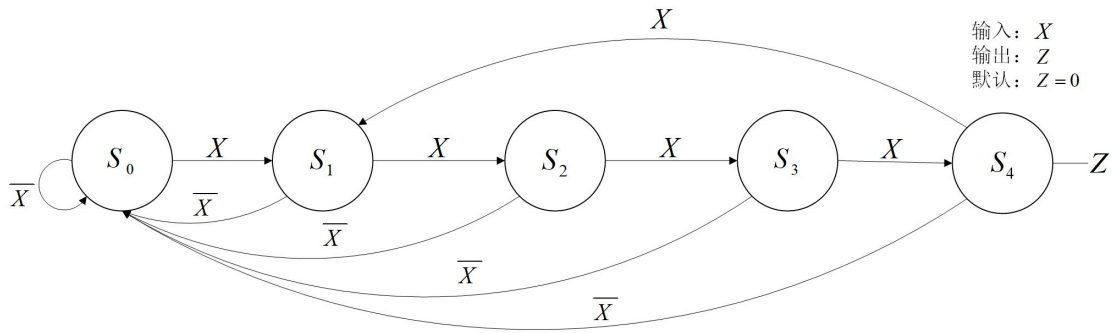


图 1 自动售货机的状态机图

c) 状态分配

采用计数赋值的方法给每个状态分配唯一的二进制编码。由于该电路共有四个状态，因此需要用两位二进制表示状态。按照 S_0 、 S_1 、 S_2 、 S_3 、 S_4 的顺序给各状态赋值，并将图 1 所示的状态机图转换为状态表如下：

表 1 自动售货机的状态表

状态	状态码	转换条件	下一个状态	状态码	非0输出
S_0	000	X	S_1	001	
		\bar{X}	S_0	000	
S_1	001	X	S_2	010	
		\bar{X}	S_0	000	
S_2	010	X	S_3	011	
		\bar{X}	S_0	000	
S_3	011	X	S_4	100	
		\bar{X}	S_0	000	
S_4	100	X	S_1	001	Z
		\bar{X}	S_0	000	

3. 电路实现

通过 Verilog 语言基于状态机对电路进行实现。

```
//针对自动售货机时序电路的设计程序
`timescale 1ns / 1ps
module machine(CLK,RESET,X,Z);
    input CLK,RESET,X;
```

```

    output Z;
    reg[2:0] state,next_state;//当前状态、下一状态
    parameter
S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,S4=3'b100;//状态的计数赋值
    reg Z;

//异步电路
always @(posedge CLK or posedge RESET)
begin
    if (RESET)
        state <= S0;//复位，回到初始状态 S0（用于初始化）
    else
        state <= next_state;//在时钟的上升沿转移状态
end

//下一状态的判断：若 X 为 1 则转移至下一状态，否则回到初始状态
always @(X or state)
begin
    case (state)
        S0: next_state = X ? S1 : S0;
        S1: next_state = X ? S2 : S0;
        S2: next_state = X ? S3 : S0;
        S3: next_state = X ? S4 : S0;
        S4: next_state = X ? S1 : S0;
    endcase
end

//输出：当且仅当连续投入 4 个硬币，输出为 1
always @(X or state)
begin
    case (state)
        S0: Z = 1'b0;
        S1: Z = 1'b0;
        S2: Z = 1'b0;
        S3: Z = 1'b0;
        S4: Z = 1'b1;// Moore 型输出
    endcase
end
endmodule

```

4. 电路验证

a) TestBench

通过编写 Verilog TestBench, 来对实现的电路模块进行功能验证。

```
//针对自动售货机时序电路的测试程序
`timescale 1ns / 1ps
module testbench();
    wire Z;
    reg clock, X, reset;
    //测试序列: 0 10 110 1110 11110 11111
    reg[0:19] test_sequence=20'b0_10_110_1110_11110_11111;
    //用于遍历测试序列的变量 i
    integer i;
    //时钟信号的一个周期
    parameter PERIOD = 100;

    machine mc(clock,reset,X,Z);
        initial
        begin
            //初始化
            reset = 1'b1;
            X = 1'b0;
            #(5*PERIOD/4);
            reset = 1'b0;
            //按照测试序列依次输入
            for (i = 0; i < 20; i = i + 1)
            begin
                X = test_sequence[i];
                //每隔时钟变化的一个周期, x 取下一个值
                #PERIOD;
            end
            //在输入序列结束时, 结束模拟
            $stop;
        end

    //时钟信号, 周期长度为 PERIOD
    always
    begin
        clock = 1'b1;
```

```

        #(PERIOD/2);
        clock = 1'b0;
        #(PERIOD/2);
    end

endmodule

```

b) 仿真结果

利用上一小节的 Verilog TestBench 进行 Vivado 仿真，得到 Wave 如下：

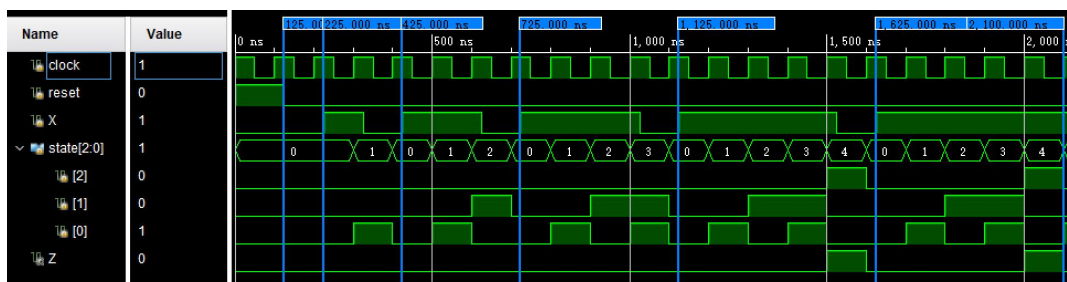


图 2 自动售货机的仿真结果

输入为 clock、reset、X，当前状态为 state，输出为 Z，对 Wave 进行分析解释如下：

首先，设置 reset 为 1，状态进行初始化，二进制表示为 000。之后，reset 设置为 0，开始输入测试序列。

测试序列为 0 10 110 1110 11111，用于遍历状态图所有可能的情况。

- i. 0 (图 2 中标记 1-标记 2)：未投入硬币，状态保持 S_0 不变，输出为 0
- ii. 10(图 2 中标记 2-标记 3)：仅连续投入一枚硬币，状态变化为 $S_0 \rightarrow S_1 \rightarrow S_0$ ，输出为 0
- iii. 110 (图 2 中标记 3-标记 4)：连续投入两枚硬币，状态变化为 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$ ，输出为 0
- iv. 1110 (图 2 中标记 4-标记 5)：连续投入三枚硬币，状态变化为 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_0$ ，输出为 0

- v. 11110 (图 2 中标记 5-标记 6) : 连续投入四枚硬币, 状态变化为 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_0$, 前四个状态输出为 0, 状态为 S_4 的时候输出为 1
- vi. 11111 (图 2 中标记 6-标记 7) : 连续投入五枚硬币, 状态变化为 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1$, 状态为 S_4 的时候输出为 1, 其余的时候输出为 0

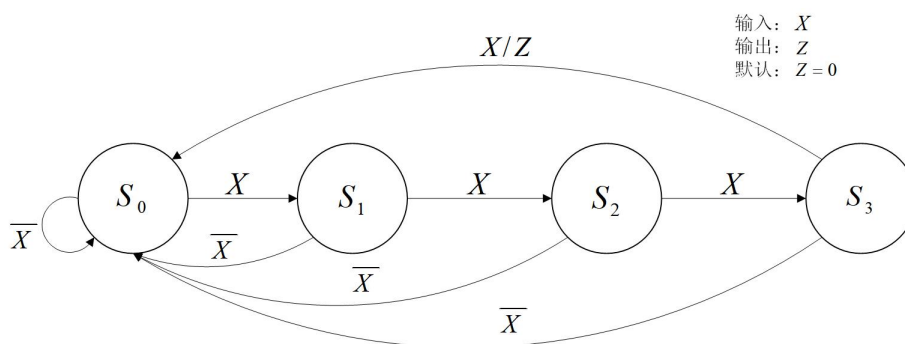
经过上述分析, 可以发现仿真的输出与期望相同, 电路设计无误。

5. 实验心得

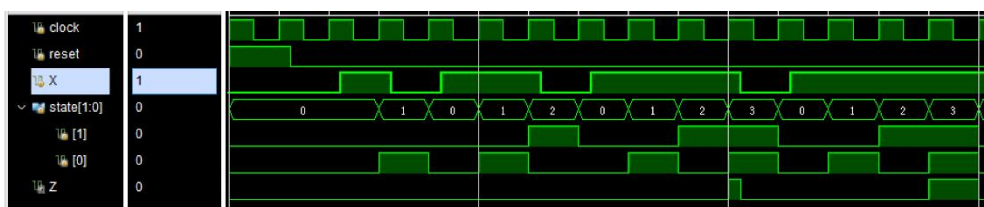
在该实验中, 我完成了对时序电路从设计到验证的全过程。

在开始设计时序电路的过程时, 我采用了 Mealy 型输出进行设计。其状态机

图如下:



然而, 在实际仿真的时候出现了问题。我采用了 0 10 110 1110 1111 的测试序列进行了验证, 然而电路在 1110 的时候出现来错误的输出 1。本质而言, 状态更新只在 clock 的上升沿进行, 而输出在一切 clock=1 的时候进行。



同时, 在调试过程中, 我学会了通过 tcl console 查看错误信息, 通过查阅相关资料, 找出代码错误。