**实验三 时序电路设计实验报告**

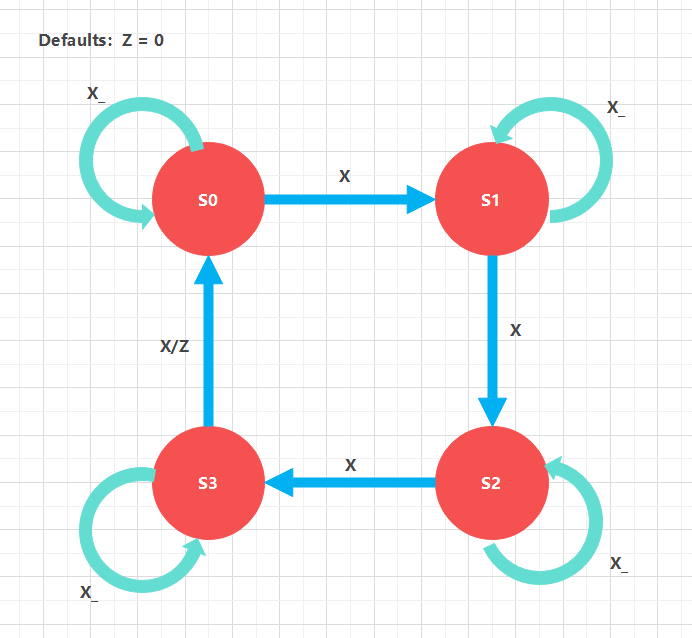
**姓名**：xxx  **学号**：xxx

**班级**：xxx **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。

1. **电路设计**
   1. **规范化**
2. **输入**：X=1表示投入一元硬币，X=0表示未投入一元硬币
3. **输出**：Z=1表示售货机给出一瓶饮料，Z=0表示未给出饮料
4. **电路的时序行为**：X累计四次为1，输出Z=1，否则Z=0
   1. **形式化**

****

* + 1. **S0：连续有0个1或连续有4个1**
    2. **S1：连续有1个1**
    3. **S2：连续有2个1**
    4. **S3：连续有3个1**
  1. **状态分配**

采用计数赋值，S0为00，S1为01，S2为10，S3为11

1. **电路实现**

`timescale 1ns / 1ps

module coin(CLK, RESET, X, Z);

input CLK, RESET, X;

output reg Z;

// 声明当前状态和下一状态

reg [1:0] state, next\_state;

// 声明4种状态

parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

always @ (posedge CLK or posedge RESET) begin

if(RESET)

state <= S0;

else

state <= next\_state;

end

// 下一状态与输入和当前状态有关

always @ (X or state) begin

case (state)

S0: next\_state = X ? S1 : S0;

S1: next\_state = X ? S2 : S1;

S2: next\_state = X ? S3 : S2;

S3: next\_state = X ? S0 : S3;

endcase

end

// 输出与当前状态和输入有关

always @ (X or state) begin

case (state)

S0: Z = 1'b0;

S1: Z = 1'b0;

S2: Z = 1'b0;

S3: Z = X ? 1'b1 : 1'b0;

endcase

end

endmodule

1. **电路验证**
   1. **TestBench**

`timescale 1ns / 1ps

module testbench();

wire Z;

reg clk = 1'b0, X, reset;

reg [0:23] test\_seq = 24'b1010\_1010\_1111\_0011\_1011\_1101;

integer i = 0;

parameter period = 10;

coin coin(clk, reset, X, Z);

initial begin

reset <= 1'b1;

# (period/2);

reset <= 1'b0;

// 将test\_seq依次赋给输入

for(i = 0; i < 24; i = i + 1) begin

X <= test\_seq[i];

# period;

end

X = 1'b0;

# (period/2);

$stop;

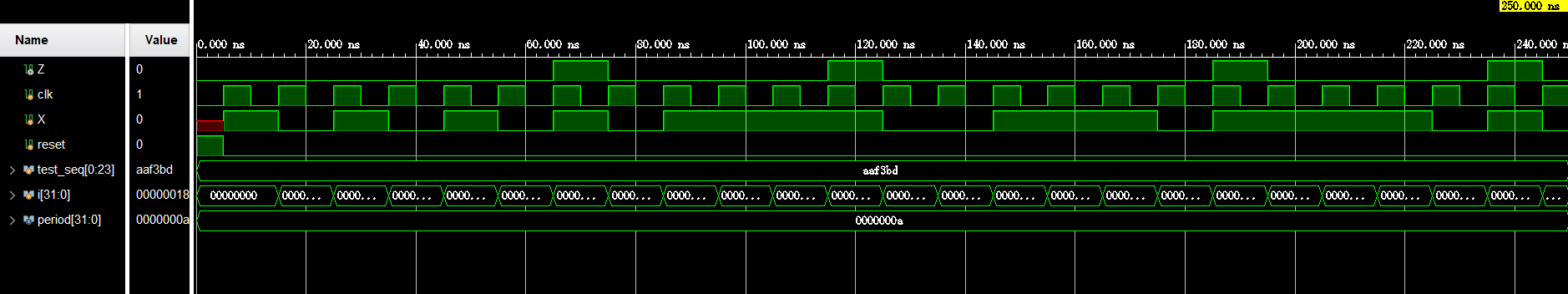
end

// 模拟时钟信号

always # (period/2) clk = ~clk;

endmodule

* 1. **仿真结果**

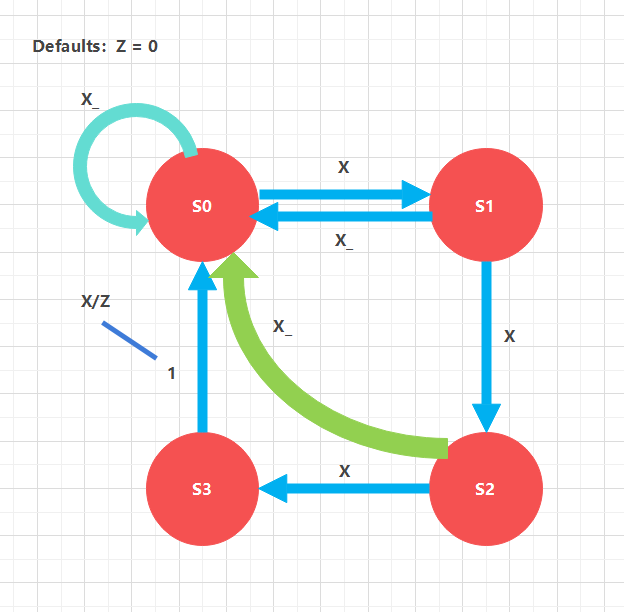


从该仿真波形可以看出，当输入累计为4个1时，输出为1；对于连续序列和不连续序列，它们的行为都是相同的。

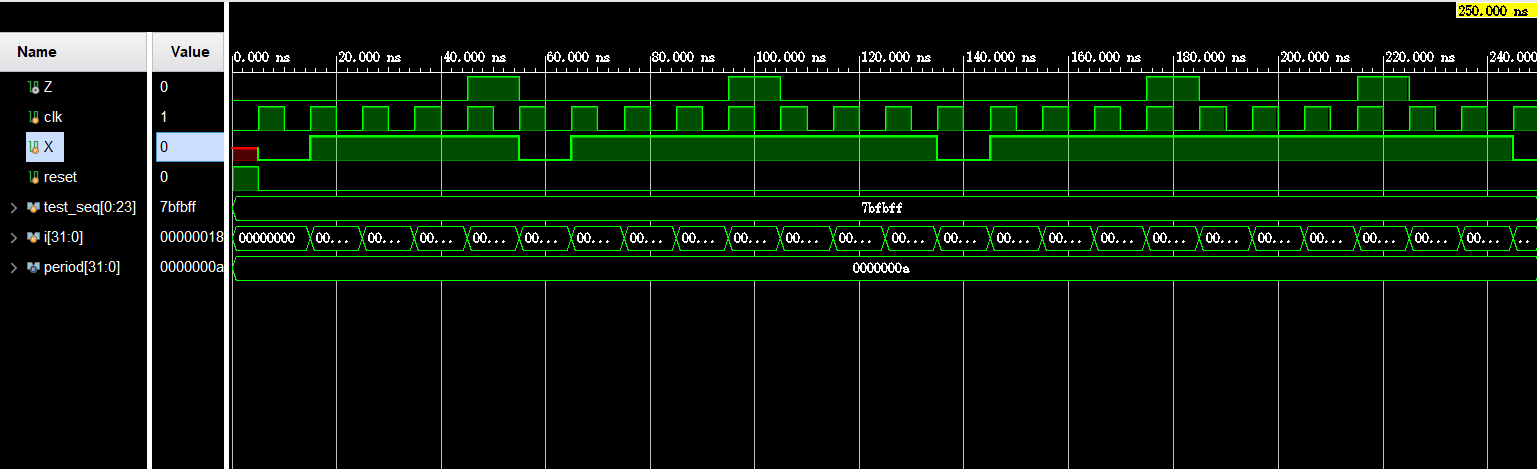
1. **实验心得**

本次实验让我掌握了时序电路从设计到验证的全过程，让我加深了对时序电路原理和设计的理解，让我掌握了Verilog对时序电路的编写方法，其中关键一点就是：每一个always块处理不同的问题，如一个always块处理状态转移，一个always块处理输出。

本次实验我在开始时理解错了题意，尤其是通过对本次实验两个题目的比较，我误认为这两个实验的关键区别就是重叠序列的识别与否，即认为题目要求的是X连续为1时Z输出1，这让我设计出如下的状态机图：

****

仿真结果如下：



不过我发现当S3转移到S0时该状态机图有些漏洞，按照如上理解，**X = 0表示投入的不是1元硬币**，而对于没有投入硬币的情况则没有考虑。然后我又仔细分析了题意，重新理解的题意如下：售货机只能识别一元硬币，对于其他类型的硬币不予接收，即**X = 0表示没有投入硬币**。这样，当X累计4次为1时输出Z为1，这样理解更符合现实生活。