

# Lab 3 Report

PB22020514 郭东昊

## 一、 实验目的与内容

### 实验内容

编写以下五个序列检测状态机，每个状态机都能够正确识别非重叠的 10101011 序列

Mealy 型两段式顺序码状态机

Mealy 型两段式独热码状态机

Moore 型一段式顺序码状态机

Moore 型两段式顺序码状态机

Moore 型三段式顺序码状态机

### 实验目的

理解 Mealy 型和 Moore 型时序电路的区别

熟练掌握 Verilog 描述有限状态机(FSM)的方法

熟练掌握利用 EDA 工具，进行逻辑电路的设计、仿真、调试、下载测试等基本方法

熟练掌握查看生成电路及其性能和资源使用情况

二、 逻辑设计

Mealy 型状态机的状态图

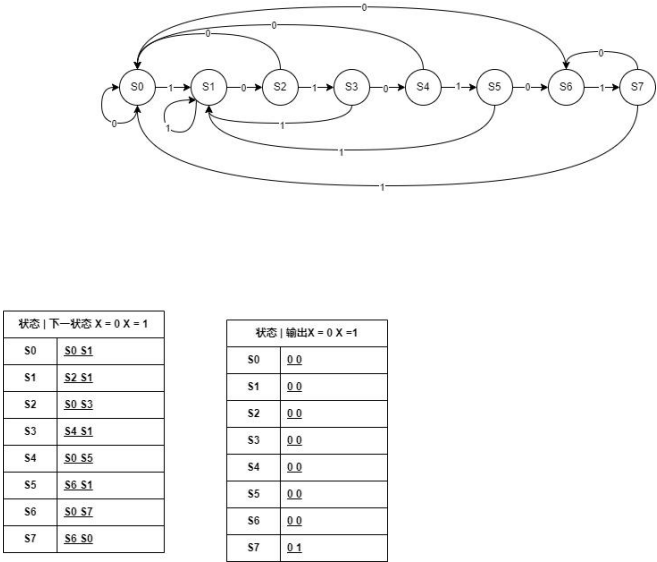


图 1 Mealy 型状态机的状态图、状态表、输入输出表

Moore 型状态机的状态图

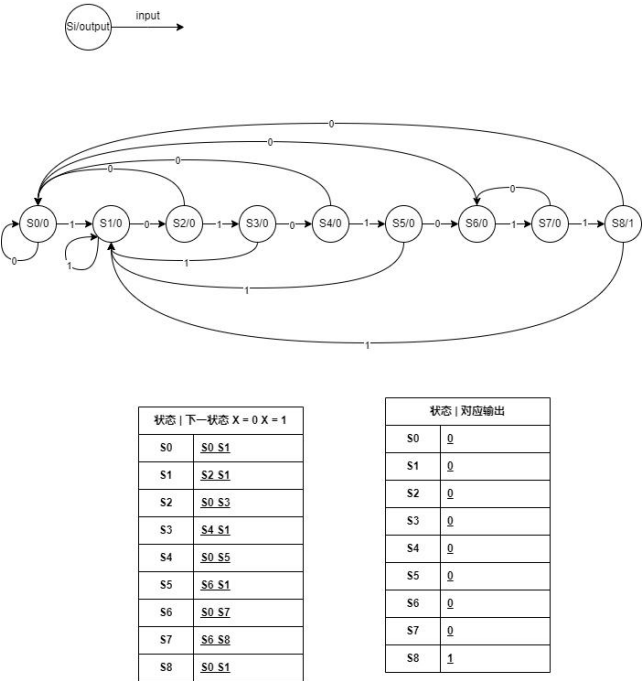


图 2 Moore 型状态机的状态图、状态表、输入输出表

### 三、 Verilog 代码实现

#### Mealy 型两段式顺序码状态机

Mealy 型，状态+输入

两段式状态机，就是将状态更新和状态转移、输出分开进行描述，见以下代码注释：

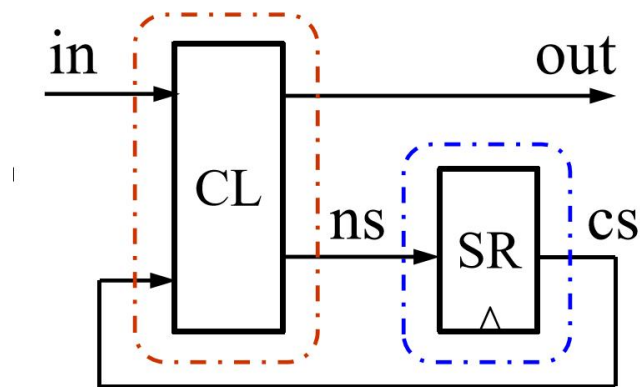
```
//状态定义，顺序码采用八进制书写，代码简洁
parameter S0 = 4'o0;
parameter S1 = 4'o1;
parameter S2 = 4'o2;
parameter S3 = 4'o3;
parameter S4 = 4'o4;
parameter S5 = 4'o5;
parameter S6 = 4'o6;
parameter S7 = 4'o7;

//状态更新
always @(posedge clk, negedge rstn) begin
    if (!rstn) cs <= S0;
    else cs <= ns;
end
```

//状态转移与输出

```
always_comb begin
    out = 0;
    ns = cs;
    case(cs)
        S0:begin
            if (in == 0) begin
                out = 0;
                ns = S0;
            end
            else begin
                out = 0;
                ns = S1;
            end
        end
    end
end
.....
end
```

示意图如下：



## Mealy 型两段式独热码状态机

实现思路同上，在 `//状态定义` 处将状态编码改为独热码即可。

## Moore 型两段式顺序码状态机

这里注意 Mealy 型与 Moore 型在代码实现上的区别，在 `case(cs)` 语句中

```
S0:begin
  if (in == 0) begin
    out = 0;
    ns = S0;
  end
  else begin
    out = 0;
    ns = S1;
  end
end
```

Mealy 型

```
S0:begin
  out = 0;
  if (in == 0)begin
    ns = S0;
  end
  else begin
    ns = S1;
  end
end
```

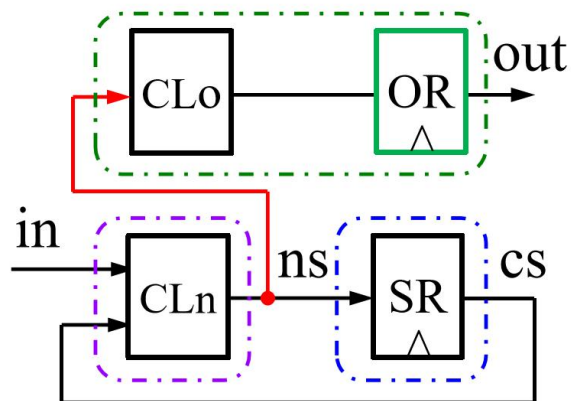
Moore 型

## Moore 型一段式顺序码状态机

一段式状态机，在代码中与两段式、三段式的最大区别就是没有 **ns** 这个中间变量，对 **cs** 直接赋（下一个状态的）值，其余大同小异。

## Moore 型三段式顺序码状态机

三段式状态机使用寄存器进行输出，示意图如下



代码实现上，每一个 **always** 语句块的职能都更加清晰，可以说，三段式将二段式的第二段又裂为两段：

```
98         end
99         S8:begin
100             if (in == 0)
101                 ns = S0;
102             else
103                 ns = S1;
104             end
105             default;
106         endcase
107     end
108
109     always @(posedge clk, negedge rstn) begin
110         if(!rstn)
111             out <= 0;
112         else begin
113             case(ns)
114                 S0: out <= 0;
115                 S1: out <= 0;
116                 S2: out <= 0;
117                 S3: out <= 0;
118                 S4: out <= 0;
119                 S5: out <= 0;
120                 S6: out <= 0;
121                 S7: out <= 0;
122                 S8: out <= 1;
123                 default;
124             endcase
125         end
126     end
127
128 endmodule
129
```

## 四、 电路设计与分析

略。

## 五、 测试结果与分析

检查通过，略。

## 六、 总结

通过本实验，我学会了状态机的 Verilog 书写，了解到在代码实现上，状态机有一段式、两段式、三段式的分别，并认识到不同的代码实现实际上对应着不同的硬件结构。本实验加深了我对 Verilog 这门硬件语言的理解，序列状态检测机的成功上板也让我获得了不小的成就感。