

# 例一、例二论文

刘正浩

2021 年 6 月 6 日

## Contents

1 例一	1
2 例二	2

### 1 例一

$Q_2 Q_1 Q_0$	$Q_2^* Q_1^* Q_0^*$	C
0 0 0	0 0 1	0
0 0 1	0 1 0	0
0 1 0	0 1 1	0
0 1 1	1 0 0	0
1 0 0	1 0 1	0
1 0 1	1 1 0	0
1 1 0	1 1 1	0
1 1 1	0 0 0	1

图 1: 例一转移/输出表

在得到如图的状态转移表之后，可以通过卡诺图求得状态转移方程和输出方程。三个状态转移方程和一个输出方程如下：

$$Q_0^* = Q_0' \quad (1)$$

$$Q_1^* = Q_1' Q_0 + Q_1 Q_0' \quad (2)$$

$$Q_2^* = Q_2' Q_1 Q_0 + Q_2 Q_1' + Q_2 Q_0' \quad (3)$$

$$C = Q_2 Q_1 Q_0 \quad (4)$$

如果使用 D 触发器搭建状态机，则激励方程为

$$D_0 = Q_0' \quad (5)$$

$$D_1 = Q_1' Q_0 + Q_1 Q_0' \quad (6)$$

$$D_2 = Q_2' Q_1 Q_0 + Q_2 Q_1' + Q_2 Q_0' \quad (7)$$

由于使用结构化描述方式进行设计，则在代码中组合逻辑部分如下：

——第一个触发器之前的组合逻辑——  
 $d\_0 \leq \text{not } q\_0;$   
 ——第二个触发器之前的组合逻辑——  
 $d\_1 \leq ((\text{not } q\_1) \text{ and } (q\_0)) \text{ or } ((q\_1) \text{ and } (\text{not } q\_0));$   
 ——第三个触发器之前的组合逻辑——  
 $d\_2 \leq ((\text{not } q\_2) \text{ and } (q\_1) \text{ and } (q\_0)) \text{ or } ((q\_2) \text{ and } (\text{not } q\_1)) \text{ or } ((q\_2) \text{ and } (\text{not } q\_0));$   
 $c \leq q\_0 \text{ and } q\_1 \text{ and } q\_2;$

设计完成后，对状态机进行仿真。仿真结果如下图：

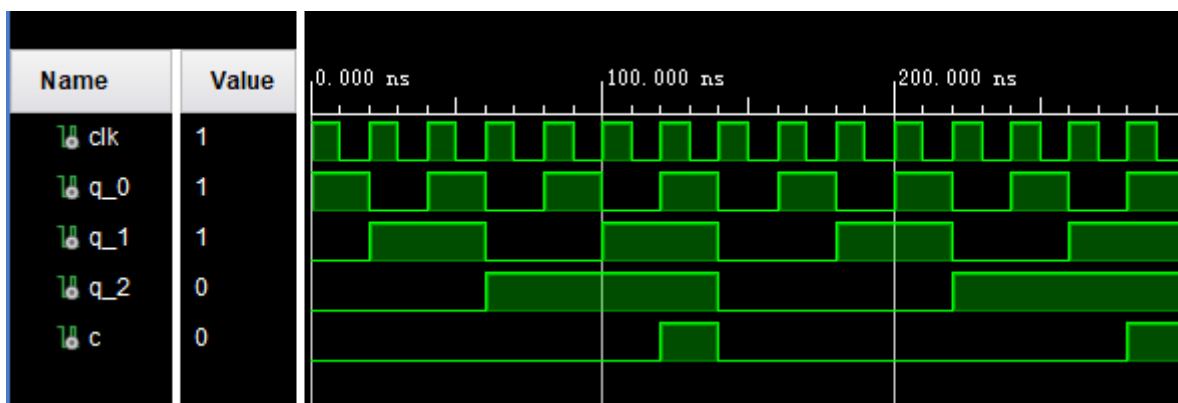


图 2: 例一仿真结果

## 2 例二

状态S	A		Z
	0	1	
→ STA	STA	A1	0
→ A1	STA	A11	0
→ A11	OK	A11	0
→ OK	STA	A1	1
	S*		

图 3: 例二状态转换表

用 A 表示串行输入的数据，用 Z 表示检测结果，可以得到 Moore 机的状态转换表。有了状态转换表，可以很容易写出三段式状态机中的状态转移部分。

```

STATE_TRANS : process(current_state, a) —状态转换
begin
    case current_state is
        when sta =>
            if(a = '1')then
                next_state <= a1;
            else
                next_state <= current_state;
            end if;
        when a1 =>
            if(a = '1')then
                next_state <= a11;
            else
                next_state <= sta;
            end if;
        when a11 =>
            if(a = '1')then
                next_state <= current_state;
            else
                next_state <= ok;
            end if;
        when ok =>
            if(a = '1')then
                next_state <= a1;
            else
                next_state <= sta;
            end if;
        when others =>
            next_state <= current_state;
        end case;
    end process;
end process;

```

其余代码详见项目代码。

仿真的结果如下图。（测试序列为 01110010110）

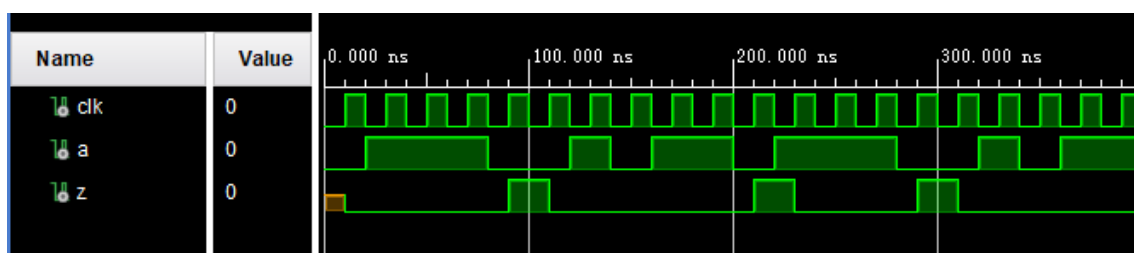


图 4: 例二仿真结果