



电子科技大学

University of Electronic Science and Technology of China

本科生实验报告

实验课程： 复杂数字集成电路设计（挑战性课程）

实验名称： 实验 7：模 7 检测器的设计

实验地点： 无

学生姓名： 周岳恒

学 号： 2021340105016

指导教师： 廖永波

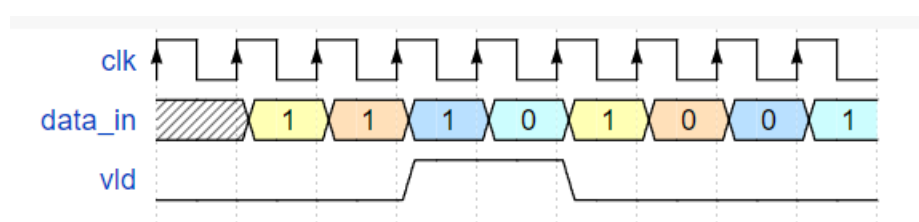
实验时间： 2023 年 10 月 23 日

一、实验目的

学习体会状态机的思想，能够根据问题的规模设计状态机的状态和转移逻辑，使用算法状态机实现一般算法的设计。

二、实验任务

实现模 7 检测器的设计，输入序列无穷，输入时钟决定输入序列的更新。当输入序列组合的数为 7 的倍数时输出有效，其他时刻输出无效。



三、实验原理

本算法很难由组合逻辑实现，需要使用状态机根据外界输入的时钟实现算法。由于有效信号在新的序列输入时就产生，所以状态机的输出取决于输入，使用 Mealy 机实现设计。

一种简单的思想是：模 7 说明一个数对 7 的余数范围为 0~6，则需要至少 7 个状态，状态寄存器至少要 4 位。每次输入的值决定次态跳转到哪个状态，对应余数是多少。把所有余数对应的状态位确定即可。每次移位输入后原来的余数要 $\times 2$ ，不过这一过程固化在真值表里，穷举即可，不需要通用逻辑。

另一种思想具有一般性，可以实现除 7 以外的模数的设计，且改动更少，更直观。使用寄存器记录上一时刻序列对 7 的余数，并使用组合逻辑判断这一次序列更新后对 7 的余数。

算法的正确性证明如下：

$$a = 7m + n$$

$$2a = 14m + 2n = 7(2m) + 2n$$

$$\textcircled{1} \quad 8 \leq 2n \leq 12$$

$$2a = 7(2m + 1) + q$$

$$\text{其中 } q = 2n - 7(0 < q < 7)$$

$$q = 2n \% 7$$

② $2n \leq 6$:

$$2a = 7(2m) + 2n$$

$$2n = 2n \% 7$$

四、实验过程

verilog 代码:

```
module mol7(
    input clk,
    input rst_n,
    input data_in,
    output vld
);
    reg [2:0] state_cur, state_nxt; // 0~6
    wire [3:0] mol_nxt; // 0~13

    always @(posedge clk, negedge rst_n) begin
        if(~rst_n)
            state_cur <= 0;
        else
            state_cur <= state_nxt;
    end

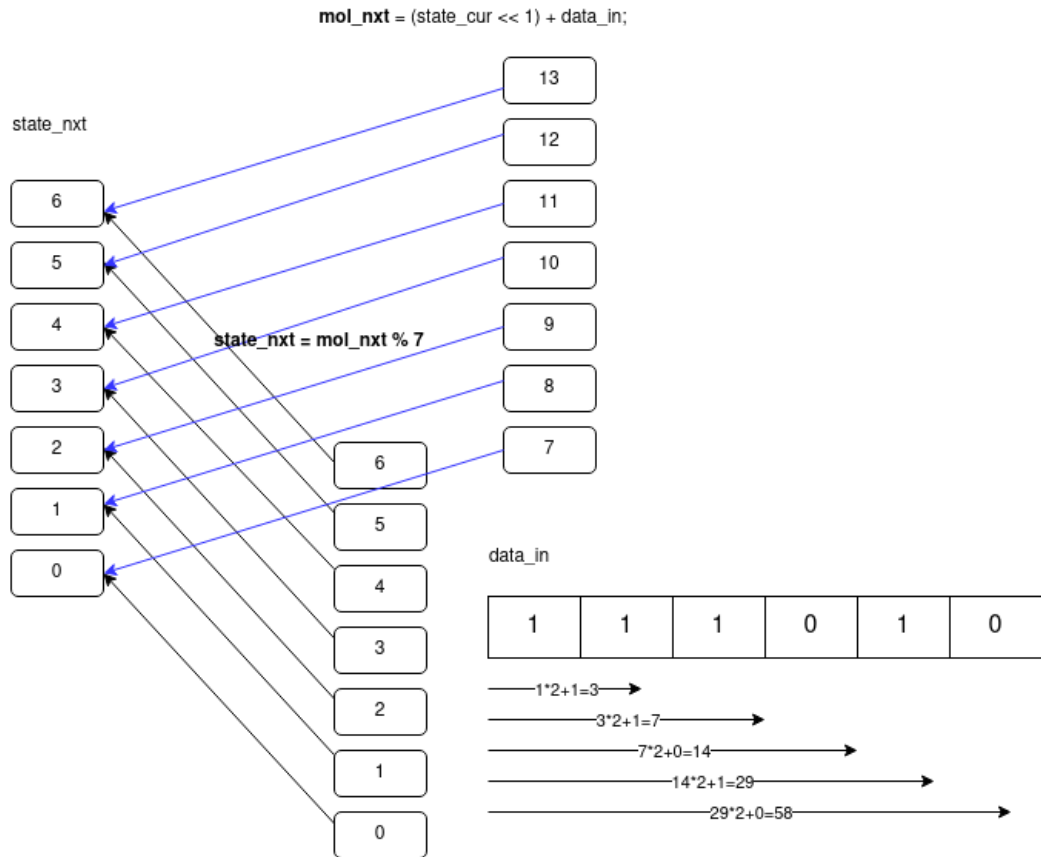
    always @(*) begin
        if(mol_nxt < 7)
            state_nxt = mol_nxt[2:0];
        else // mol_nxt < 14
            state_nxt = mol_nxt - 7;
    end

    assign mol_nxt = (state_cur << 1) + data_in;

    assign vld = state_nxt == 0;

endmodule
```

state_cur 表示现态的模数, state_nxt 表示次态的模数 q (0~6), mol_nxt 表示移位 $\times 2$ 后的 $2n$ (0~13), 还要再模一次 2 再赋给 state_nxt。



spyglass 报告: PASS

Message	File	Line
Message Tree (Total: 2, Displayed: 2, Waived: 0)		
Design Read (2)		
DetectTopDesignUnits (1) : Identify the top-level design units in user design.		
Module mol7 is a top level design unit	proj7/mol7_sc...	1
ElabSummary (1) : Generates Elaborated design units Summary data		
Please refer file './spyglass-1/mol7/lint/rtl/spyglass_reports/SpyGlass/elab_summary.rpt' for elab summary report	./spyglass-1/m...	0

五、 仿真结果

testbench:

```

`timescale 1ps/1ps
module test ();
  reg clk;
  reg rst_n;
  reg data_in;
  wire vld;

  mol7 u1(
    .clk(clk),

```

```

        .rst_n(rst_n),
        .data_in(data_in),
        .vld(vld)
    );

    initial begin
        clk = 1'b0;
        forever #5 clk = ~clk;
    end

    // initial begin
    //     rst_n = 1'b0;
    //     data_in = 1'b0;
    //     #6 rst_n = 1;
    //     data_in = 1'b1;
    //     #10 data_in = 1'b1;
    //     #10 data_in = 1'b1;
    //     #10 data_in = 1'b0;
    //     #10 data_in = 1'b1;
    //     #10 data_in = 1'b0;
    //     #10 data_in = 1'b0;
    //     #10 data_in = 1'b1;
    // end

    reg [31:0] seq_num;
    initial begin
        rst_n = 1'b0;
        data_in = 1'b0;
        seq_num = 0;
        #6 rst_n = 1'b1;
        forever begin
            data_in = {$random};
            seq_num = (seq_num << 1) + data_in;
            #10;
        end
    end

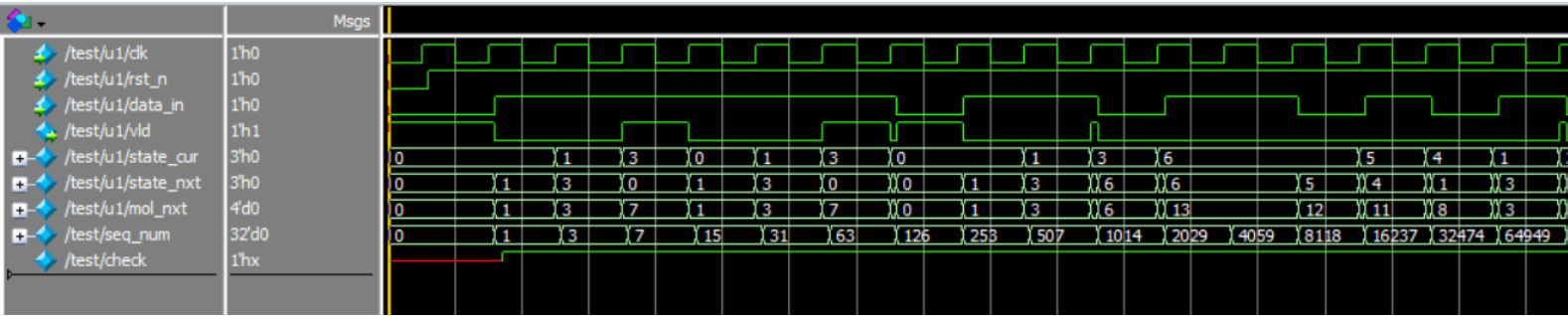
    reg check;
    always @(posedge clk) begin
        if(rst_n) begin
            #2;
            check = ((seq_num % 7 == 0 && vld) || (seq_num % 7 != 0 &&
~vld)) ? 1'b1 : 1'b0;
        end
    end

```

```
end

endmodule
```

仿真波形：



可以看到使用 seq_num 在 testbench 记录序列组成的数字模 7 后的结果和电路设计结果相一致，验证设计算法的正确性。每个时钟边沿到来时 data_in 移位组成序列，vld 信号及时反馈当前组成序列模 7 的结果是否为 0。vld 的毛刺是由于 data_in 滞后时钟导致的，不影响结果。state_next 很好地反映了每次模数的余的结果。

六、实验总结

本次实验进一步体会 Mealy 机的使用场景，了解状态机的使用场景，并设计模数算法实现无穷长序列的模数检测，具有一般性，且易于扩展，对电路的算法设计有进一步的掌握。