

北京航空航天大学
2014 ~2015 学年第 二 学期
《电子电路设计训练》期末考试试卷
(2015 年 07 月 06 日)

A. 数字部分 (共 50 分)

参考答案和评分标准

一. 填空题 (共 10 分, 每空 1 分)

1. 使用 Verilog HDL 语言在 RTL 级描述数据如何在寄存器之间流动, 以及描述如何处理、控制这些数据流动, “RTL” 的含义是: 寄存器传输级 (填写 “Register Transfer Level” 也对), 这属于 行为 描述 (选填: 系统、算法、行为、逻辑、门级、开关级)。
2. 定义一个存储宽度为 16 位、存储深度为 8 的存储器空间, 地址范围是 0~7, 存储器名为 rom_a, 定义语句为: reg [0:15] rom_a [0:7]; (填写 “reg [1:16] rom_a [0:7];”, 以及位宽的指标集为 “[15:0]” 等, 只要宽度和数组个数满足要求都对)。
3. 运算符 “!=” 和 “!==” 的语义存在着区别, 如果 reg [1:0] a = 2'b0x; reg [1:0] b = 2'b0z; , 则 a != b 的值为: x (填写 “1'bx”、“不确定” 也对); a !== b 的值为: 1 (填写 “1'b1”、“真” 也对)。
4. 循环次数不确定的 while 循环语句是 不可综合 的 (选填: 可综合、不可综合); fork...join 形式的块语句是 不可综合 的 (选填: 可综合、不可综合)。
5. 函数的定义用关键字 function 和 endfunction; 函数调用后的返回值是通过函数名变量传递给调用语句, 默认的函数调用返回值的类型为 寄存器。(选填: 线网、寄存器)

二. 组合逻辑设计和测试（共 12 分，第 1 小题 6 分，第 2 小题 6 分）

某保险库房的通道由两道门组成，如图 1，其设计要求为当 A 门打开时，B 门一定是关闭的，反之亦然。

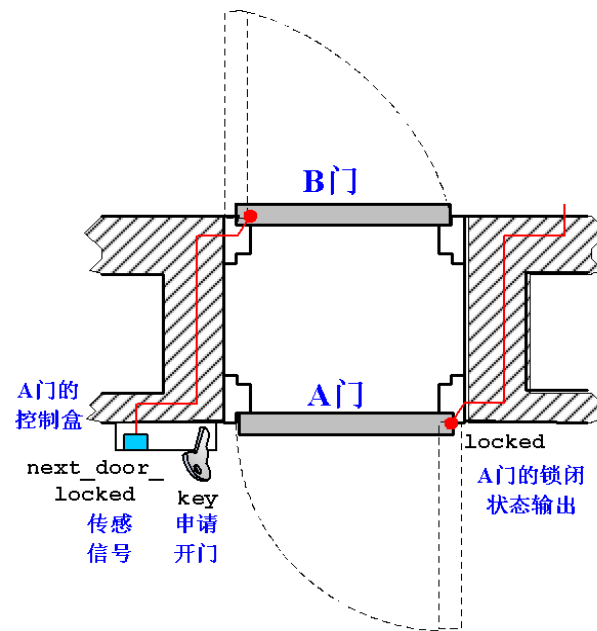


图 1 库房的 A 门和 B 门

(1) 设计问题

设每扇门均有开门请求的输入信号 `key`、表示门是否为锁定状态的输出信号 `locked`，以及来自另一扇门的锁定状态输入信号 `next_door_locked`，请用 Verilog HDL 语言设计门的控制器模块 `door_controller`。

参考答案 1:

```
module door_controller ( key, next_door_locked, locked );  
    input key, next_door_locked ;  
    output locked ;  
  
    assign locked = !( key && next_door_locked );  
  
endmodule
```

参考答案 2:

```
module door_controller ( key, next_door_locked, locked );
```

```

input key, next_door_locked ;
output locked ;

reg locked ;
always @ ( key or next_door_locked )
    locked = !( key && next_door_locked );

endmodule

```

说明：

① 如果将两个门锁一起考虑设计组合逻辑，也是可以的，但请严格检查语法和逻辑是否正确，全部正确可得 6 分，但一旦有误，属于没有正确理解题意，酌情加重扣分；

② 如果设计成时序逻辑电路，先扣 1 分，随后检查是否有合理的部分，分语法、逻辑进行检查。

(2) 测试问题

编写测试程序 (test bench)，测试某通道的 A 门和 B 门开启闭合的关系，需要监测 ① 只有一扇门开启 (A 门或 B 门) 的情况 `one_door_unlocked`、② 两扇门同时都开启 (正常情况下不应出现) 的情况 `both_doors_unlocked`。

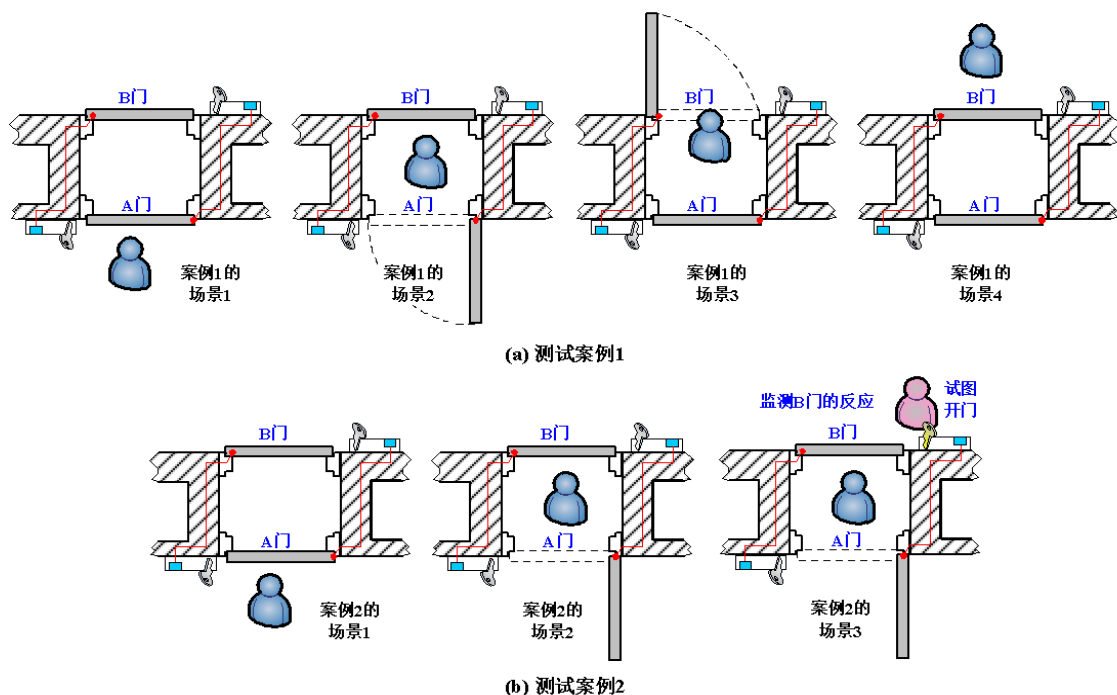


图 2 库房的 A 门和 B 门控制器的测试案例

测试案例需要包括：（如图 2）

- 案例 1——依次打开 A 门、关闭 A 门、打开 B 门和关闭 B 门的过程；
- 案例 2——打开 A 门且尚未关闭之前，某人试图打开 B 门的情况。

请编写相应的测试模块 tb_door_controllers 。

参考答案：

```
module tb_door_controllers ;
    reg key_A, key_B ;
    wire locked_A, locked_B ;
    wire one_door_open, both_doors_open ;

    door_controller uut_A ( .key( key_A ),
                           .next_door_locked( locked_B ),
                           .locked( locked_A ) );
    door_controller uut_B ( key_B, locked_A, locked_B );
    initial begin
        #10 key_A = 0 ; key_B = 0 ;
        #10 key_A = 1 ;
        #10 key_A = 0 ;
        #10 key_B = 1 ;
        #10 key_B = 0 ;
        #20 ;
        #10 key_A = 1 ;
        #5 key_B = 1 ;
        #5 key_A = 0 ;
        #10 key_A = 0 ;
        $stop ;
    end

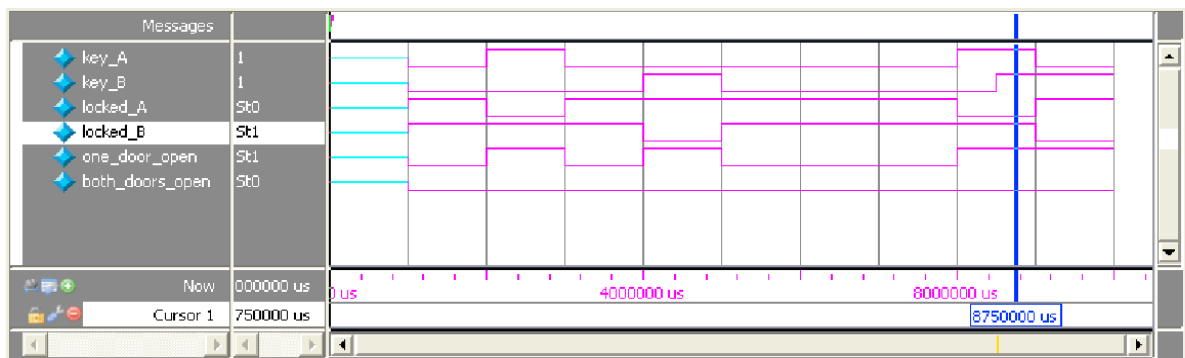
    assign one_door_open = locked_A ^ locked_B ;
    //或 !locked_A && locked_B || locked_A && !locked_B

    assign both_doors_open = ! locked_A && ! locked_B ;

endmodule
```

说明：

- ① 参考仿真波形。



三. 分析与设计（共 15 分，第 1 小题 5 分，第 2 小题 10 分）

循环冗余校验（cyclic redundancy checks, CRC）在信息码元后附加专门的校验码元，构成“循环码”，进行检错与纠错。例如：(7, 3)循环码的码长为 7，包含 3 个信息码元和 4 个校验码元，如表 1 所示。

表 1 (7, 3)循环码

	信息码元 [7:5]	校验码元 [4:1]	合成后的码字 [7:1]
0	000	0000	7'b000_0000
1	001	0111	7'b001_0111
2	010	1110	7'b010_1110
3	011	1001	7'b011_1001
4	100	1011	7'b100_1011
5	101	1100	7'b101_1100
6	110	0101	7'b110_0101
7	111	0010	7'b111_0010

循环码编码器的核心部件是“模 2 除法器”，对于(7, 3)循环码，图 3 给出了采用门级描述实现的模 2 除法器的 Verilog HDL 代码。

```

module div_mod2_gates ( clk, rst_n, enable, in, out );
    input clk ;
    input in ;
    input rst_n ;
    input enable ;
    output out ;

    wire [1:4] d, q ;
    wire feedback ;

```

```

D_FF D_buf ( .clk(clk), .rst_n(rst_n), .D(q[4]), .Q(buf_out));

D_FF D4 ( .clk(clk), .rst_n(rst_n), .D(d[4]), .Q(q[4]) ) ;
D_FF D3 ( .clk(clk), .rst_n(rst_n), .D(d[3]), .Q(q[3]) ) ;
D_FF D2 ( .clk(clk), .rst_n(rst_n), .D(d[2]), .Q(q[2]) ) ;
D_FF D1 ( .clk(clk), .rst_n(rst_n), .D(d[1]), .Q(q[1]) ) ;

mux2 switch1 ( .d0( 1'b0 ), .d1( feedback ),
               .sel( enable ), .y( d[1] ) ) ;
mux2 switch2 ( .d0( buf_out ), .d1( in ),
               .sel( enable ), .y( out ) ) ;

xor gate4 ( feedback, q[4], in ) ;
xor gate2 ( d[3], q[2], d[1] ) ;
xor gate1 ( d[2], q[1], d[1] ) ;

assign d[4] = q[3] ;

endmodule

// module D_FF ( Delay Flip-Flop )
module D_FF ( clk, rst_n, D, Q, Q_n ) ;
    input clk, rst_n, D ;
    output Q, Q_n ;
    reg Q ;
    assign Q_n = ~Q ;
    always @ ( posedge clk or negedge rst_n )
        if ( !rst_n ) Q <= 0 ;
        else Q <= D ;
endmodule

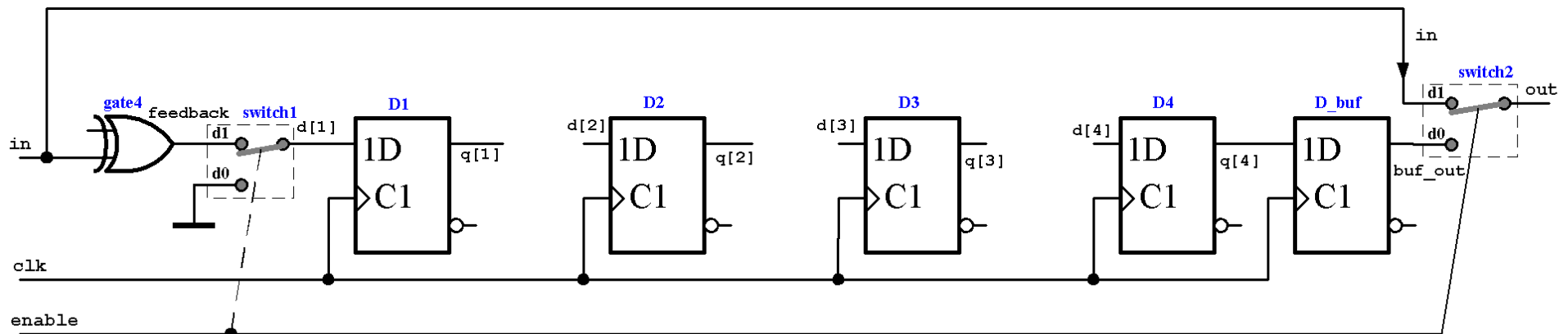
// module 2-output multiplexer
module mux2 ( d0, d1, sel, y );
    input d0, d1, sel ;
    output y ;
    assign y = ( sel == 0 ) ? d0 : d1 ;
endmodule

```

图 3 门级描述实现的模 2 除法器

生成(7,3)循环码的模2除法器电路如图4所示（注意：此图待补全），其主体部分由一串D触发器和门电路组成，工作原理为：

- 1) 当 enable 为高电平时，双联开关都接通到 d1 触点，串行的 3 个信息码元在 3 个时钟边沿的触发下，依次通过 gate4 输入，同时直接通过 out 端口输出；在这 3 个时钟周期内，D 触发器 D1、D2、D3、D4 通过逻辑运算生成 4 位校验码元；
- 2) 随后，使 enable 为低电平，双联开关都切换到 d0 触点，switch1 切断来自 in 的输入，switch2 使输出端 out 接到 D_buf 的输出端（D_buf 将 D4 的内容延迟一个时钟周期输出，用于拼接信息码元和校验码元的时序），寄存于 D1、D2、D3、D4 中的校验码元在 4 个时钟边沿的触发下，依次输出到 out 端口。



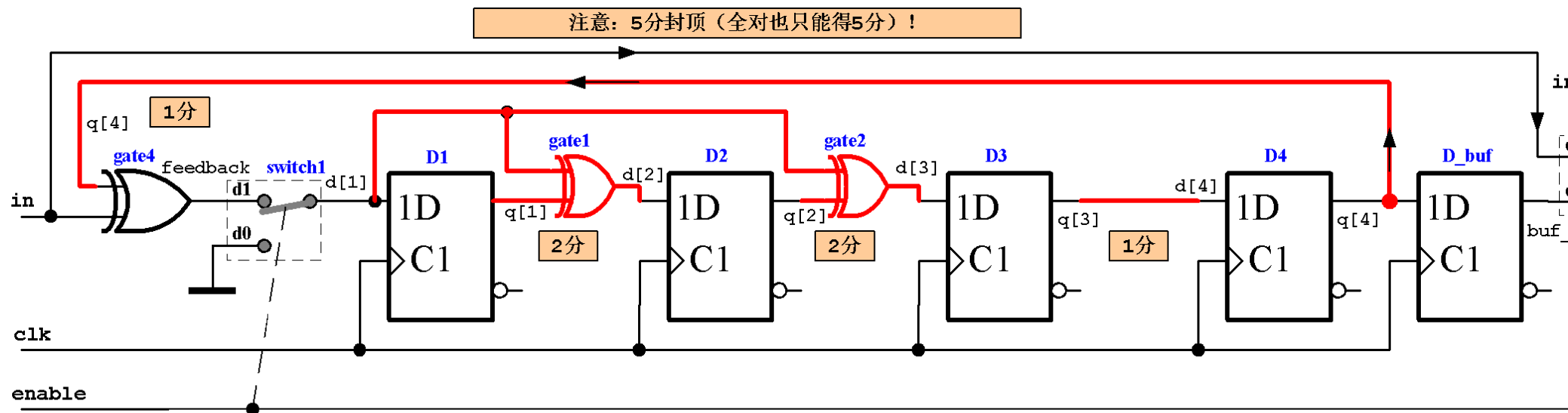
说明：1. 由enable信号控制双联开关；当enable为高电平，接通d1；当enable为低电平，接通d0。 2. 为了简洁，图中没有画出D触发器的复位信号

图4 采用时序逻辑电路实现的(7,3)循环码模2除法器（待补全）

(1) 分析问题

请根据门级描述实现的模2除法器模块 `div_mod2_gates` 的代码（参见图 3），补全图 4 所示的电路原理图。（可以直接在图 4 中补充）

参考答案：



说明：由enable信号控制双联开关：当enable为高电平，接通d1；当enable为低电平，接通d0。

(2) 设计问题

请采用 RTL 级描述方法，不允许使用“xor”、“D_FF”、“mux2”等元件，设计 (7, 3) 循环码的模 2 除法器模块，并编写出相应的 Verilog HDL 代码，该模块的名称、输入输出信号请按照图 5 中的定义。（可以将代码填写到图 5 中）

参考答案：（推荐）

```
module div_mod2_behavior ( clk, rst_n, enable, in, out );
    input clk , in, rst_n, enable ;
    output out ;

    reg out ;

    reg [1:4] register ;

    always @( posedge clk or negedge rst_n ) begin
        if ( !rst_n ) register <= 4'b0000 ;
        else if ( enable == 1 ) begin
            out <= in ;
            register [4] <= register [3] ;
            register [3] <= in ^ register [4] ^ register [2] ;
            register [2] <= in ^ register [4] ^ register [1] ;
            register [1] <= in ^ register [4] ;
        end
        else
            out <= register [4] ;
            register <= register >> 1 ;
        end
    end

endmodule
```

图 5 RTL 级描述实现的模 2 除法器（参考答案，已补全）

四. 综合设计（共 13 分，第 1 小题 5 分，第 2 小题 8 分）

利用第三题所述的“模 2 除法器”构造完整的“(7, 3)循环码编码器”必须妥善地解决信息码元和校验码元的时序配合关系。

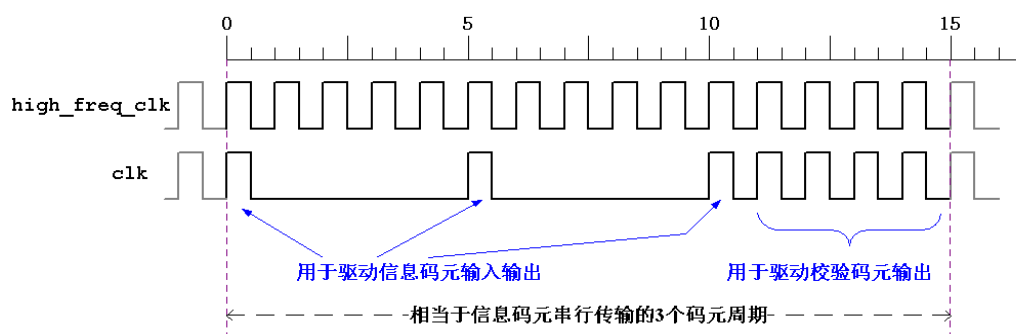


图 6 不均匀的时钟脉冲驱动方案

为了在 3 个信息码元周期内串行地输出 7 位码字，可以采用不均匀的时钟脉冲驱动方案，如图 6 所示；具体思路为：在主频时钟 `high_freq_clk` 连续的 15 个周期内，去除第 2、3、4、5 个和第 7、8、9、10 个时钟脉冲，只用第 1、6、11 个时钟脉冲驱动信息码元的输入和输出，而用剩下的第 12、13、14、15 个时钟脉冲驱动校验码元的输出。可见，在这种方案中，信息码元的周期为 `high_freq_clk` 时钟周期的 5 倍。

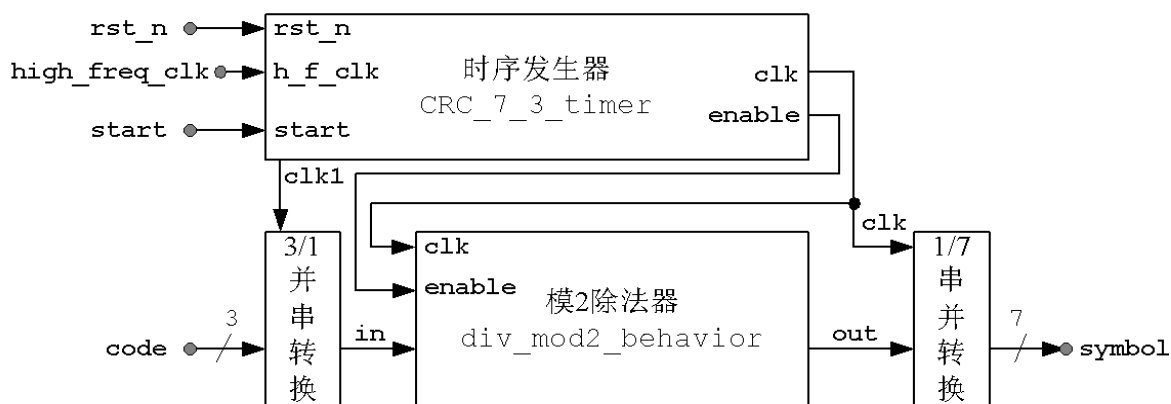


图 7 完整的(7, 3)循环码编码器框图

根据这种方案设计的(7, 3)循环码编码器顶层结构框图如图 7 所示。对于其中的“时序发生器”模块 `CRC_7_3_timer`，输入输出波形的时序关系如图 8 所示，其中：

- `start` 为宽脉冲信号，其宽度大于一个 `high_freq_clk` 时钟周期，

且与 high_freq_clk 时钟异步；

- enable 信号从 start 的上升沿开始从 0 变为 1，从第 3 个 clk 脉冲（或 clk1 脉冲）的下降沿开始从 1 变为 0。

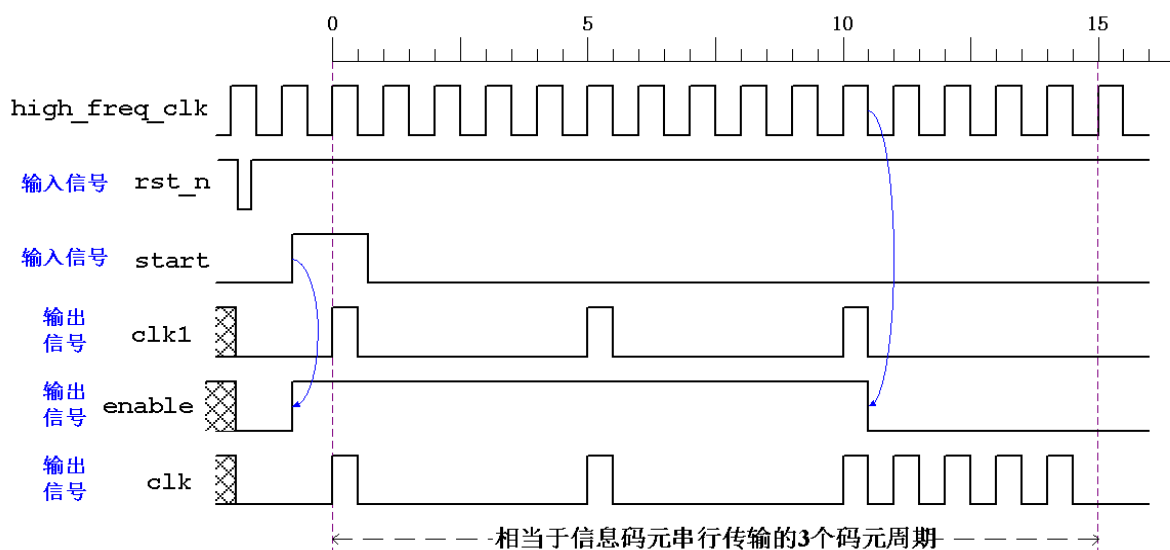


图 8 CRC_7_3_timer 模块的输入输出信号之间的时序关系

(1) 有限状态机设计

请观察图 8 所示的信号时序关系，设计 CRC_7_3_timer 的有限状态机

(finite state machine, FSM)，要求：

- 列出所设定的状态和计数器的名称；
- 绘制状态转换图。

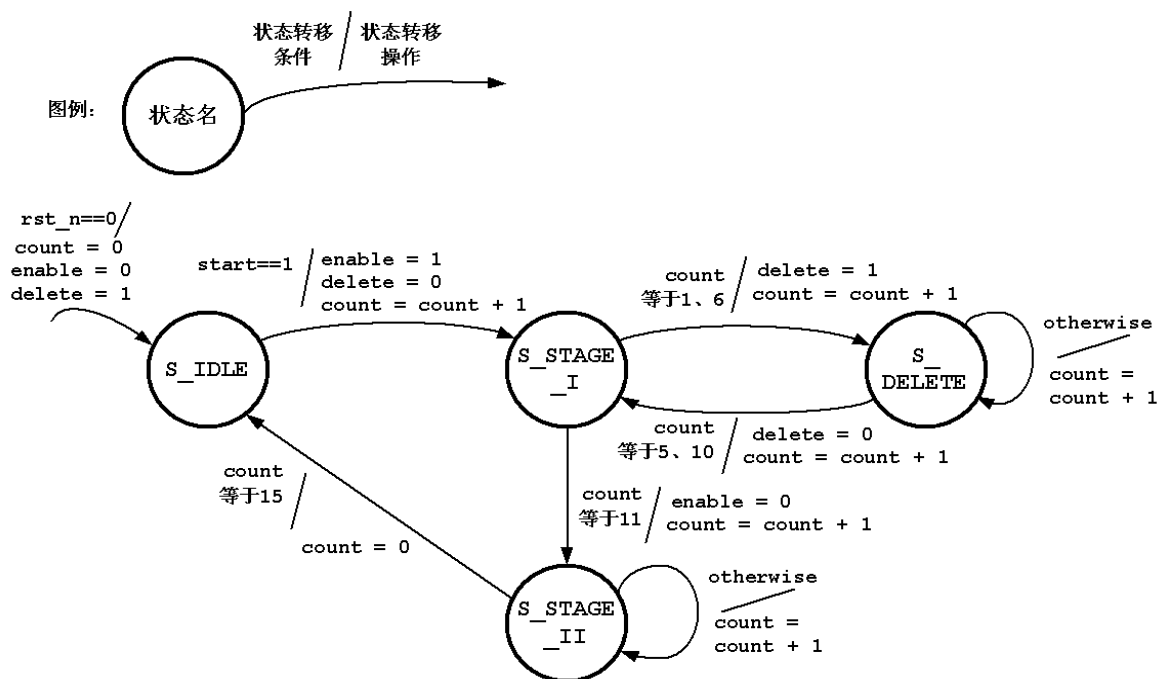
参考答案：（本题解法不唯一，以评价知识点给分为主）

答：设置状态变量 state，位宽为 3-bit，共有四种状态，即：

```
parameter [2:0] S_IDLE = 3'b000, S_STAGE_I = 3'b010,  
                S_DELETE = 3'b011, S_STAGE_II = 3'b110 ;
```

并配合 4-bit 计数器 count。

状态转换图如下：



说明:

- ① 这里，定义内部信号 `delete`，对应 `count=1` 至 `count=4` 的 4 个脉冲，以及对应 `count=6` 至 `count=9` 的 4 个脉冲，这样外部用组合逻辑可以去掉相应的 8 个时钟脉冲；
- ② 该状态图中的 `enable` 脉冲起始于时钟 (`high_freq_clk`) 的边沿，而题目要求的 `enable` 信号在 `start` 有效后立即变为 1，所以需要利用组合逻辑对 `enable` 的输出作一些修正；在第(2)部分的有限状态机实现中，状态机中对变量 `ori_enable` 进行操作，而利用组合逻辑根据 `ori_enable` 和 `start` 的关系生成题目要求的 `enable` 信号。

(2) 有限状态机的实现

请采用 Verilog HDL 语言实现 `CRC_7_3_timer` 模块，要求：

- 其中的有限状态机请采用“一段式”的风格实现；
- 要求该有限状态机的实现可综合。

参考答案：（本题解法不唯一，以评价知识点给分为主）

```
module CRC_7_3_timer ( high_freq_clk, rst_n,
                        start, clk1, enable, clk );

    input high_freq_clk, rst_n ;
    input start ;
    output clk1 ;
    output enable ;
    output clk ;

    reg enable ;

    parameter [2:0] S_IDLE = 3'b000, S_STAGE_I = 3'b010,
                   S_DELETE = 3'b011, S_STAGE_II = 3'b110 ;

    reg [2:0] state ;
    reg [3:0] count ;
    reg ori_enable ;
    reg delete ;

    always @ ( negedge high_freq_clk or negedge rst_n ) begin
        if( !rst_n ) begin
            ori_enable <= 0 ;
            delete <= 1 ;
            count <= 0 ;
            state <= S_IDLE ;
            delete <= 0 ;
        end
        else begin
            case ( state )
                S_IDLE : begin
                    if ( start ) begin
                        state <= S_STAGE_I ;
                        ori_enable <= 1 ;
                        count <= count + 1 ;
                        delete <= 0 ;
                    end
                end
                S_STAGE_I : begin
                    if ( count == 1 || count == 6 ) begin
                        delete <= 1 ;
                        state <= S_DELETE ;
                    end
                    else if ( count >= 11 ) begin
                        ori_enable <= 0 ;
                    end
                end
            endcase
        end
    end
```

```

        delete <= 0 ;
        state <= S_STAGE_II ;
    end
    count <= count + 1 ;
end
S_DELETE : begin
    if ( count == 5 || count == 10 ) begin
        delete <= 0 ;
        state <= S_STAGE_I ;
    end
    count <= count + 1 ;
end
S_STAGE_II : begin
    if ( count == 15 ) begin
        state <= S_IDLE ;
        count <= 0 ;
    end
    else count <= count + 1 ;
end
default : begin
    state <= S_IDLE ;
    count <= 0 ;
end
endcase
end
end

always @ ( state or start or ori_enable ) begin
    if ( state == S_IDLE ) enable = start || ori_enable ;
    else enable = ori_enable ;
end

assign clk1 = high_freq_clk && ori_enable && ( !delete ) ;

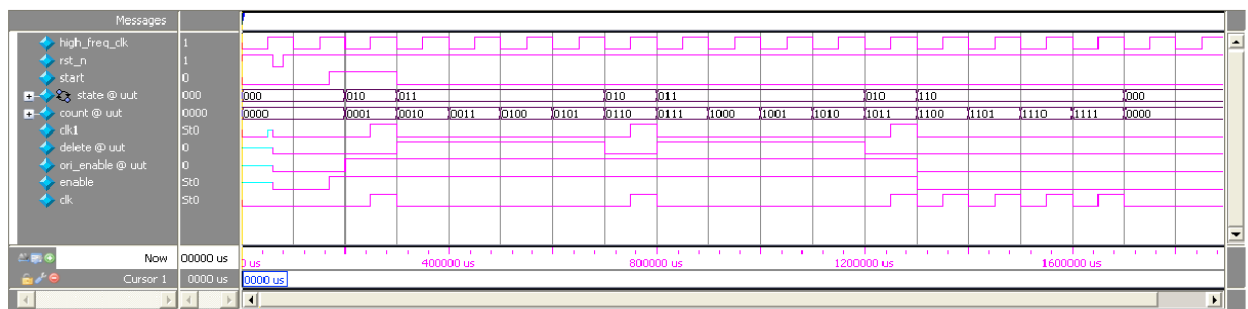
assign clk = high_freq_clk && ( !delete ) &&
            ( state != S_IDLE );

endmodule

```

说明:

① 参考仿真波形。



②解法不唯一，如果按照“两段式”或“三段式”求解，先惩罚性扣未审题 1 分，但其它分数仍按“套子”、变量类型、状态变量/计数器、复位、多路分支、FSM 正确性、整体合理性给分，但是封顶到 7 分（有罚分）。