

《数字逻辑与数字系统》实验报告

学院 计算机科学与技术 年级 2022 级 班级 4 班

姓名 陆子毅 学号 3022206045

实验项目名称 自动贩售机的设计与实现

一. 实验目的

1. 掌握基于 SystemVerilog HDL 的时序逻辑电路建模方法；
2. 掌握计数器设计方法，并能够使用计数器设计使能时钟（用于时钟分频）；
3. 掌握移位寄存器设计方法，并能够利用移位寄存器设计边沿检测电路；
4. 掌握 7 段数码管的动态显示。
5. 掌握有限状态机的设计方法。；
6. 能够使用 SystemVerilog 进行三段式状态机的建模。

二. 实验内容

采用有限状态机，基于 SystemVerilog HDL 设计并实现一个报纸自动贩售机。整个工程的顶层模块如图 3-8 所示，输入/输出端口如表 3-1 所示。使用 4 个七段数码管实时显示已付款和找零情况。其中，两个数码管对应“已付款”，另两个数码管对应“找零”，单位为分。通过 1 个拨动开关对数字钟进行复位控制。使用两个按键模拟投币，其中一个按键对应 5 分，另一个按键对应 1 角。使用 1 个 LED 灯标识出售是否成功，灯亮表示出售成功，则表示已付款不够，出售失败。假设报纸价格为 15 分，合法的投币组合包括：

- 1 个 5 分的硬币和一个 1 角的硬币，不找零
- 3 个五分的硬币，不找零
- 两个 5 分的硬币和 1 个 1 角的硬币，找零 5 分。
- 两个 1 角的硬币是合法的，找零 5 分。

当投入硬币的组合为上面 4 种之一时，则购买成功，LED 灯亮。购买成功后，LED 灯持续亮 10 秒，然后自动熄灭，同时 4 个数码管也恢复为 0。

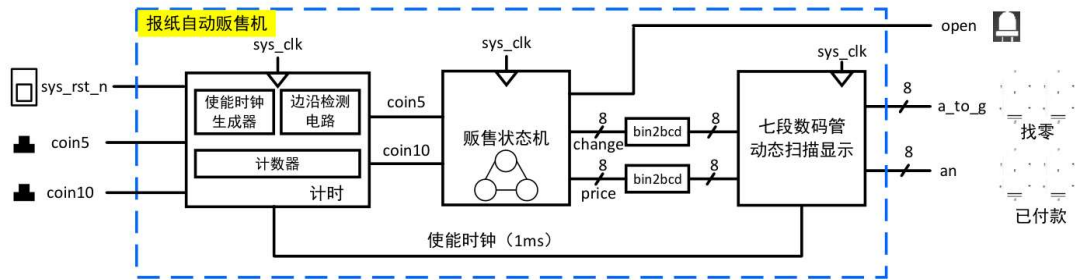


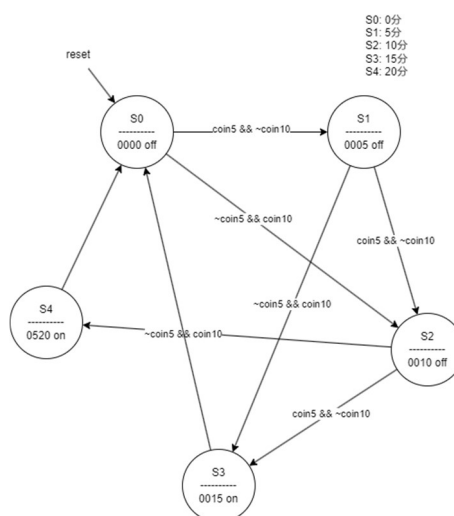
图 3-8 报纸自动贩售机顶层模块

报纸自动贩售机由 4 部分构成。

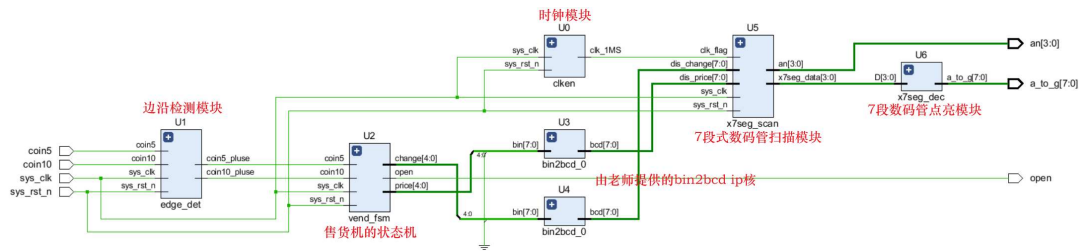
- 第一部分是计时器模块，该模块又由 3 个子模块构成，分别是计数器电路、使能时钟生成电路和边沿检测电路。
- 第二部分是整个自动贩售机电路的核心——贩售机状态机。状态机根据投币情况产生“已付款”和“找零”输出。此外，如果已付款超过 15 分，则将 LED 灯点亮，表示出售成功。
- 第三部分是两个 8 位二进制转 BCD 模块，分别将二进制的“已付款”和“找零”值转化为 BCD 编码，即 10 进制数。本实验中，该模块不需要实现，由教师直接提供 IP 使用。
- 第四部分是 7 段数码管动态扫描显示模块，它实现“已付款”和“找零”值的最终显示。

三. 实验原理与步骤（注：不用写工具的操作步骤，而是设计步骤）

1. 画出自动贩售机的状态转换图



2.2. 画出自动贩售机电路的原理图（模块级别即可，如使能时钟模块、边沿检测模块等）。



3. 报纸自动贩售机的 SystemVerilog 代码。

顶层模块：

```

module vend(
    input sys_clk, sys_rst n,
    input coin5, coin10,
    output [3 : 0] an,
    output [7 : 0] a_to_g,
    output open
);

    logic clk 1MS;
    logic coin5 pluse, coin10 pluse;
    logic [4 : 0] price, change;
    logic [7 : 0] dis_price, dis_change;
    logic [3 : 0] x7seg_data;

    clkgen U0(
        .sys_clk      (sys_clk)
        .sys_rst n    (sys_rst n) ,
        .clk 1MS      (clk 1MS)
    );

    edge_det U1(
        .sys_clk      (sys_clk)
        .sys_rst n    (sys_rst n)
        .coin5        (coin5)
        .coin10       (coin10)
        .coin5 pluse   (coin5 pluse)
        .coin10 pluse  (coin10 pluse)
    );

    vend_fsm U2(
        .sys_clk      (sys_clk)
        .sys_rst n    (sys_rst n),
        .coin5        (coin5 pluse)
        .coin10       (coin10 pluse)
        .price        (price)
        .change       (change)
        .open         (open)
    );

    bin2bcd 0 U3(
        .bin({3'b000, price}),
        .bcd(dis_price)
    );

    bin2bcd 0 U4(
        .bin({3'b000, change}),

```

```

        .bcd(dis change)
    );

    x7seg_scan U5(
        .sys_clk      (sys_clk),
        .sys_rst_n    (sys_rst_n),
        .clk_flag      (clk_1MS),
        .dis_price      (dis_price),
        .dis_change     (dis_change),
        .x7seg_data     (x7seg_data),
        .an              (an)
    );

    x7seg_dec U6(
        .D              (x7seg_data),
        .a_to_g         (a_to_g)
    );

endmodule

```

状态机 FSM 模块:

```

module vend_fsm(
    input                sys_clk,
    input                sys_rst_n,
    input                coin5,
    input                coin10,
    output [4 : 0]       price,
    output [4 : 0]       change,
    output               open
);
    reg r_open;
    reg [4:0] r_price;
    reg [4:0] r_change;
    // State
    logic [2:0] S0 = 3'b000, S1 = 3'b001, S2 = 3'b010,
    S3 = 3'b011, S4 = 3'b100;
    logic [2:0] state;
    logic [2:0] next_state;
    // State Register
    always ff @(posedge sys_clk) begin
        if (!sys_rst_n) begin
            state <= S0;
        end
        else begin
            state <= next_state;
        end
    end
    // State Transition
    always comb begin
        case(state)
            S0: begin
                if (coin5) begin
                    next_state = S1;
                end
                else if (coin10) begin
                    next_state = S2;
                end
                else next_state = S0;
            end
            S1: begin
                if(coin5) begin

```

```

        next state = S2;
    end
    else if(coin10) begin
        next state = S3;
    end
    else next state = S1;
end
S2: begin
    if(coin5) begin
        next state = S3;
    end
    else if(coin10) begin
        next state = S4;
    end
    else next state = S2;
end
S3: begin
    next state = S3;
end
S4: begin
    next state = S4;
end
endcase
end
// Output
always ff @(posedge sys clk) begin
    if(!sys rst n) begin
        r open <= 1'b0;
        r price <= 5'd0;
        r change <= 5'd0;
    end
    else begin
        case(next state)
            S0: begin
                r open <= 1'b0;
                r price <= 5'd0;
                r change <= 5'd0;
            end
            S1: begin
                r open <= 1'b0;
                r price <= 5'd5;
                r change <= 5'd0;
            end
            S2: begin
                r open <= 1'b0;
                r price <= 5'd10;
                r change <= 5'd0;
            end
            S3: begin
                r open <= 1'b1;
                r price <= 5'd15;
                r change <= 5'd0;
            end
            S4: begin
                r open <= 1'b1;
                r price <= 5'd20;
                r change <= 5'd5;
            end
        endcase
    end
end
assign open = r open;
assign price = r price;

```

```

    assign change = r change;
endmodule

```

边沿检测模块:

```

module edge det(
    input sys clk,
    input sys rst n,
    input coin5,
    input coin10,
    output coin5 pluse,
    output coin10 pluse
);
    reg coin5 q1, coin5 q2;
    reg coin10 q1, coin10 q2;
    always ff @(posedge sys clk) begin
        if (!sys rst n) begin
            coin5 q1 <= 1'b0;
            coin5 q2 <= 1'b0;
            coin10 q1 <= 1'b0;
            coin10 q2 <= 1'b0;
        end
        else begin
            coin5 q1 <= coin5;
            coin5 q2 <= coin5 q1;
            coin10 q1 <= coin10;
            coin10 q2 <= coin10 q1;
        end
    end
    assign coin5 pluse = coin5 q1 & ~coin5 q2;
    assign coin10 pluse = coin10 q1 & ~coin10 q2;
endmodule

```

7 段数码管扫描模块:

```

module x7seg scan(
    input sys clk,
    input sys rst n,
    input clk flag,
    input [7 : 0] dis price,
    input [7 : 0] dis change,
    output logic [3 : 0] x7seg data,
    output logic [3 : 0] an
);
    localparam STUBNUM = 4;
    localparam AN0 = 4'b0001,
               AN1 = 4'b0010,
               AN2 = 4'b0100,
               AN3 = 4'b1000;

    logic [1 : 0] r cnt;
    always ff @(posedge sys clk) begin
        if (!sys rst n) r cnt <= 2'b0;
        else if (clk flag) begin
            if (r cnt == STUBNUM - 1) r cnt <= 2'b0;
            else r cnt <= r cnt + 1;
        end
        else r cnt <= r cnt;
    end
end

```

```

always ff @(posedge sys clk) begin
    if (!sys rst n) an <= 4'b1111;
    else begin
        case (r cnt)
            2'b00 : an <= AN0;
            2'b01 : an <= AN1;
            2'b10 : an <= AN2;
            2'b11 : an <= AN3;
        endcase
    end
end
always ff @(posedge sys clk) begin
    if (!sys rst n) x7seg data <= 4'b0000;
    else begin
        case (r cnt)
            2'b00 : x7seg data <= dis price[3 : 0];
            2'b01 : x7seg data <= dis price[7 : 4];
            2'b10 : x7seg data <= dis change[3 : 0];
            2'b11 : x7seg data <= dis change[7 : 4];
        endcase
    end
end
endmodule

```

CLKEN 1ms 时钟生成模块:

```

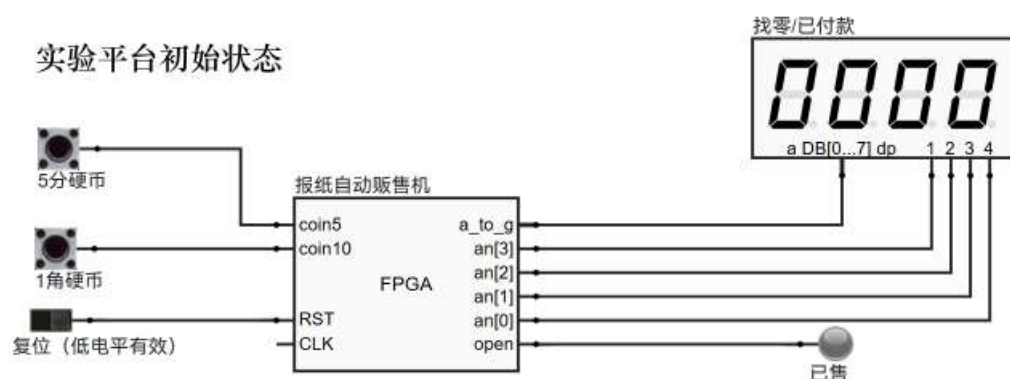
module clken (
    input logic sys clk,
    input logic sys rst n,
    output logic clk 1MS
);
    reg [19:0] cnt;
    always ff @(posedge sys clk) begin
        if (!sys rst n) begin
            clk 1MS <= 1'b0;
            cnt <= 0;
        end else begin
            if (cnt == 24999) begin
                clk 1MS <= 1;
                cnt <= 0;
            end else begin
                cnt <= cnt + 1;
                clk 1MS <= 0;
            end
        end
    end
end
endmodule

```

四. 仿真与实验结果（注：仿真需要给出波形图截图，截图要清晰，如果波形过长，可以分段截取；实验结果为远程 FPGA 硬件云平台的截图）

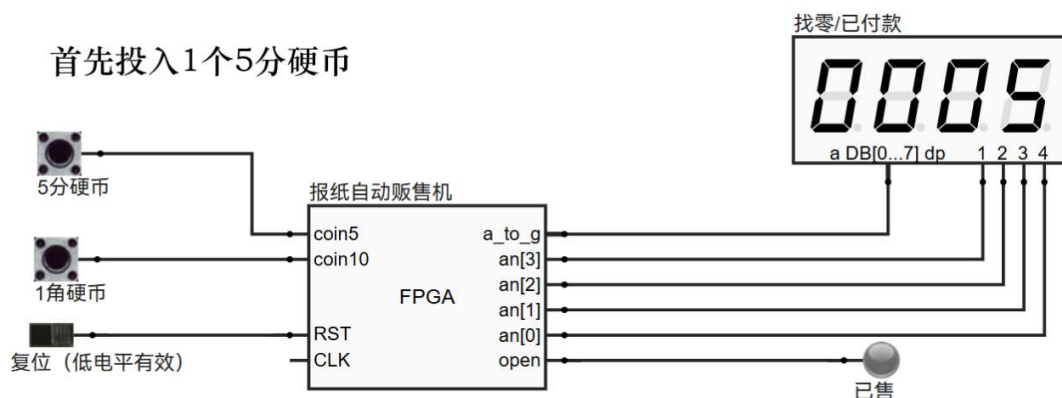
注：所有投币组合都要给出截图

实验平台初始状态

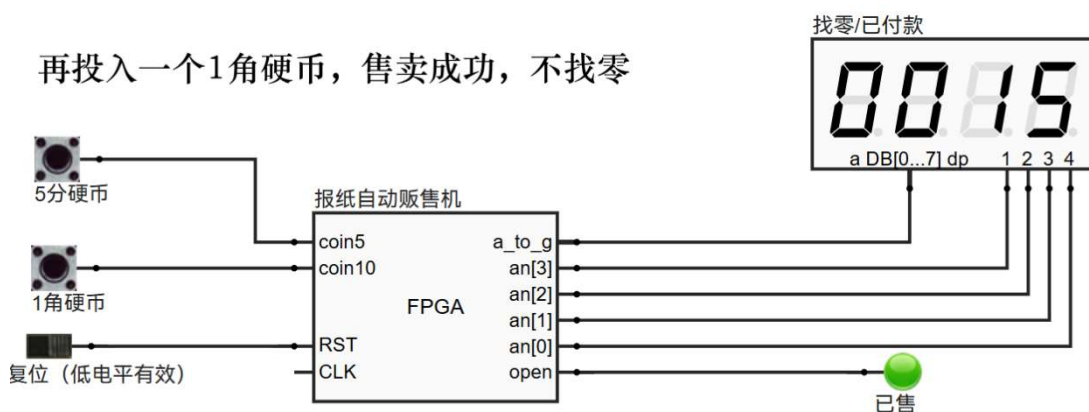


第一种情况，投入 1 个 5 分，1 个 1 角：

首先投入1个5分硬币

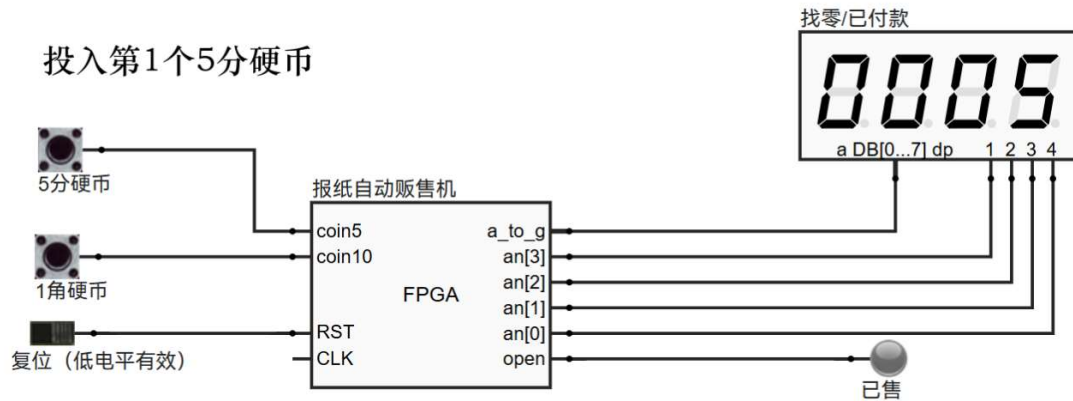


再投入一个1角硬币，售卖成功，不找零

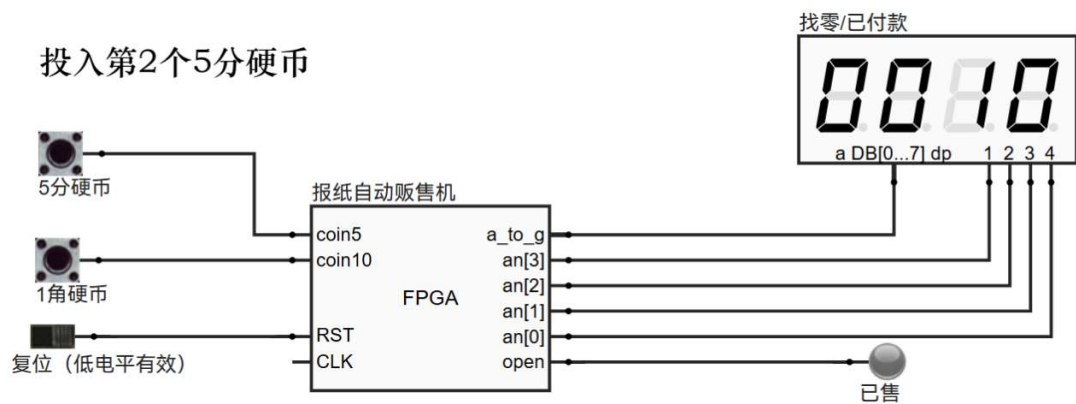


第二种情况，投入 3 个 5 分：

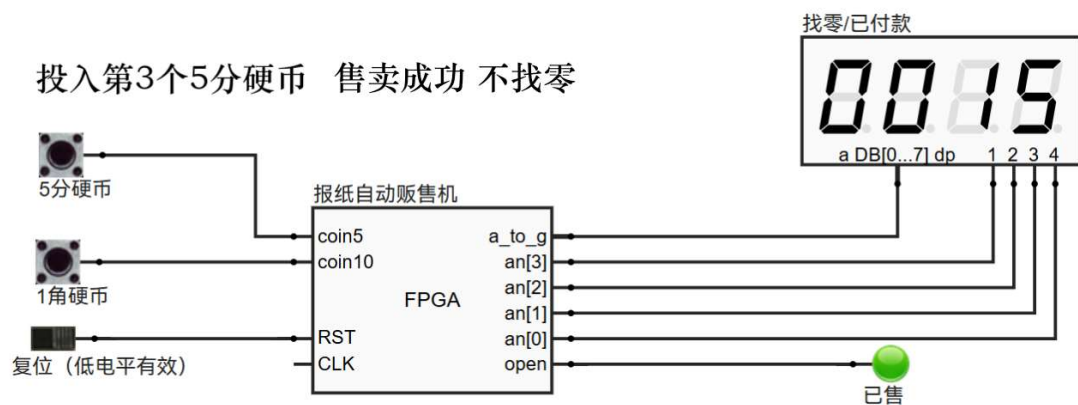
投入第1个5分硬币



投入第2个5分硬币

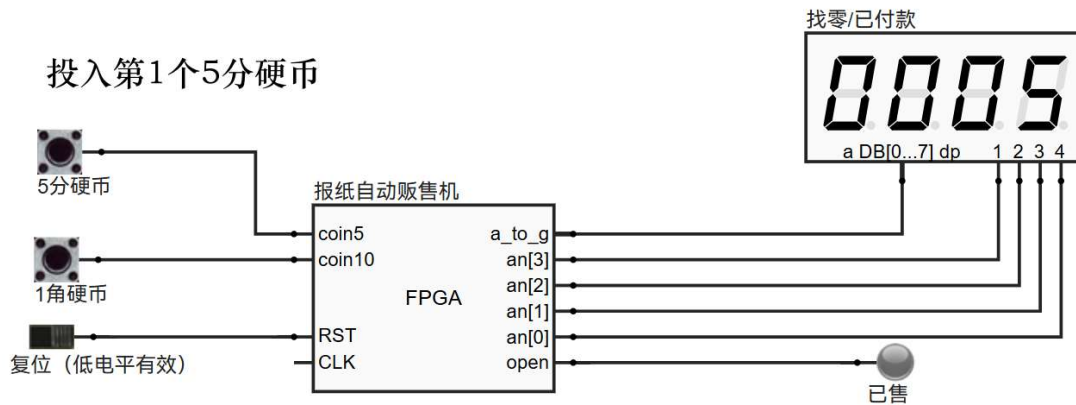


投入第3个5分硬币 售卖成功 不找零

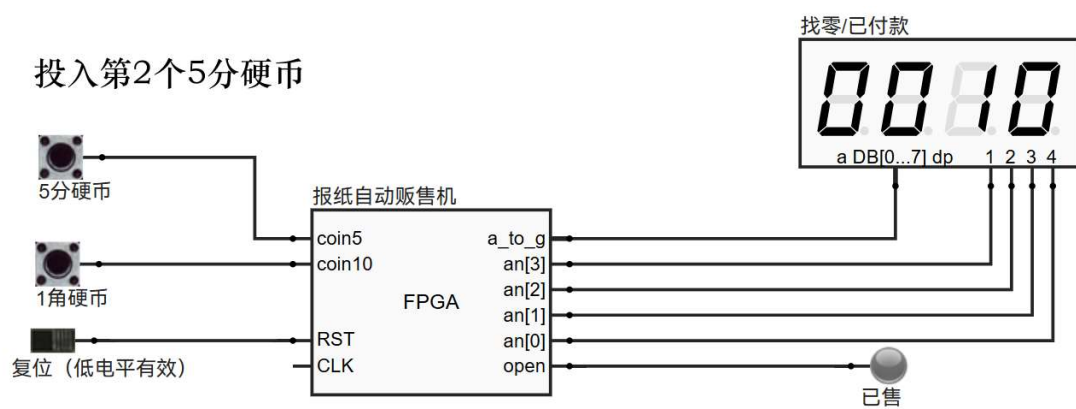


第三种情况，投入2个5分，1个1角：

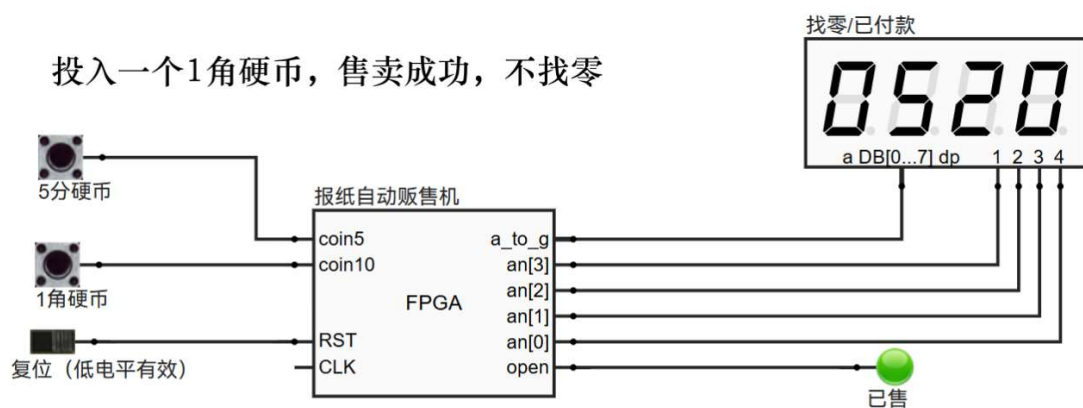
投入第1个5分硬币



投入第2个5分硬币

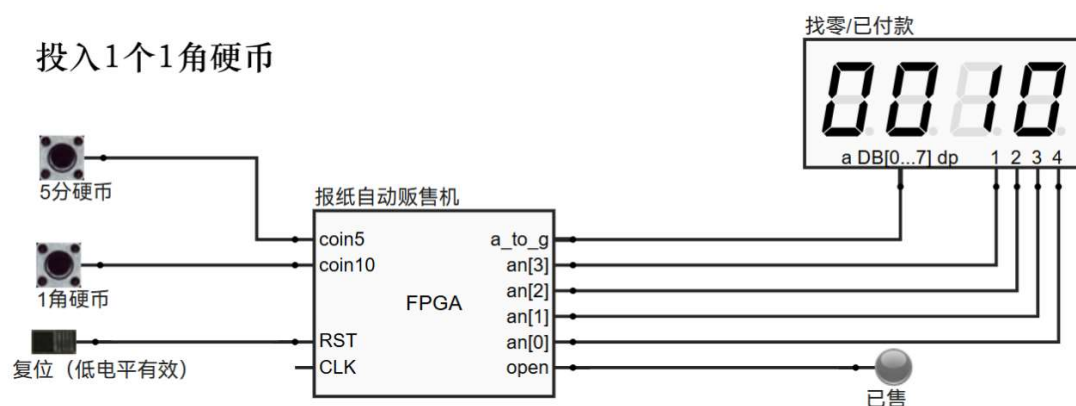


投入一个1角硬币，售卖成功，不找零

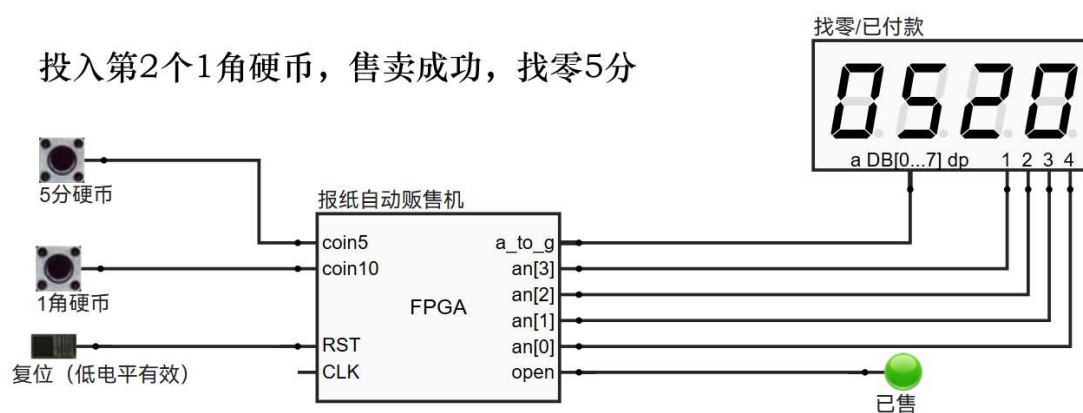


第四种情况，投入 2 个 1 角：

投入 1 个 1 角硬币



投入第 2 个 1 角硬币，售卖成功，找零 5 分



五. 实验中遇到的问题和解决办法

- 1、在构建状态机时候，利用常量进行标记，但是在定义常量的时候，出现了定义的位数少于状态所需要的位数，导致状态机不能正常运作。
- 2、在实现 `clken` 模块的时候，没有将计数器清零，导致实验时二极管出现轮流闪烁的情况。
- 3、进行模拟的时候，要将模拟文件设置为顶层文件，否则模拟结果将会不正确。
- 4、模拟完成以后，可以在左侧选中一些没有在波形图里面展示的变量，方便调试和寻找问题所在。

六. 附加题

无