

from 20241280汤英琦

• 分析输入输出信号

```
输入信号: clk, rst;
输入信号: 操作开始: op_start; //定义1开始操作
输入信号: 投币币值: coin_val; //定义2'b01表示0.5元; 2'b10表示1元
输入信号: 取消操作指示 : cancel_flag; //定义1为取消操作
输出信号: 机器是否占用: hold_ind; //定义0为不占用, 可以使用
输出信号: 取饮料信号: drinktk_ind; //定义1为取饮料
输出信号: 找零与退币标志信号: charge_ind; //定义1为找零
输出信号: 找零与退币币值: charge_val;
//定义3'b001表示找0.5元; 3'b010表示找1元
// 3'b011表示找1.5元; 3'b100表示找2.0元;
```

题解:

1. design代码,有限状态机最基本的三段式结构

```
module auto_vending_machine(
    clk,
    rst_n,
    op_start,
    cancel_flag,
    coin_val,
    hold_ind,
    drinktk_ind,
    charge_ind,
    charge_val
);
    input clk,rst_n,op_start,cancel_flag;
    input [1:0] coin_val;
    output reg hold_ind;
    output reg drinktk_ind;
    output reg charge_ind;
    output reg [2:0] charge_val;

    parameter S0=3'b000;
    parameter S1=3'b001;
    parameter S2=3'b010;
    parameter S3=3'b011;
    parameter S4=3'b100;
    parameter S5=3'b101;
    parameter S6=3'b110;
    reg [2:0] state;
    reg [2:0] next_state;
    //时序逻辑, 状态转移
    always@(posedge clk or posedge rst_n)
```

```
if(rst_n)
    state<=S0;
else
    state<=next_state;

//组合逻辑，输入 和 现态 决定次态
always@(state,op_start,coin_val,cancel_flag,rst_n)
    if(rst_n)
        next_state=S0;
    else begin
        case(state)
            S0:begin
                if(op_start==1)begin
                    if(coin_val==2'b01)next_state=S1;
                    else if(coin_val==2'b10)next_state=S2;
                    else next_state=S0;
                end
                else next_state=S0;
            end
            S1:begin
                if(cancel_flag==1)next_state=S0;
                else begin
                    if(coin_val==2'b01)next_state=S2;
                    else if(coin_val==2'b10)next_state=S3;
                    else next_state=S1;
                end
            end
            S2:begin
                if(cancel_flag==1)next_state=S0;
                else begin
                    if(coin_val==2'b01)next_state=S3;
                    else if(coin_val==2'b10)next_state=S4;
                    else next_state=S2;
                end
            end
            S3:begin
                if(cancel_flag==1)next_state=S0;
                else begin
                    if(coin_val==2'b01)next_state=S4;
                    else if(coin_val==2'b10)next_state=S5;
                    else next_state=S3;
                end
            end
            S4:begin
                if(cancel_flag==1)next_state=S0;
                else begin
                    if(coin_val==2'b01)next_state=S5;
                    else if(coin_val==2'b10)next_state=S6;
                    else next_state=S4;
                end
            end
            S5:begin
                next_state=S0;
            end
        endcase
    end
end
```

```
S6:begin
    next_state=S0;
end
default:
    next_state=S0;
endcase
end

//组合逻辑,输出仅和state和某个输入(cancel_flag)有关
//由于有多个输出,为保证速度和代码的可阅读性,采用多个always块来实现
//drinktk_ind
always@(state)
if(state==S5 || state==S6)
    drinktk_ind=1;
else
    drinktk_ind=0;

//hold_ind
always@(state)
if(state==S0)
    hold_ind=0;//只有在s0是空闲态
else
    hold_ind=1;

//charge_ind && charge_val
always@(state,cancel_flag)//找零和现态和cancel_flag有关
if(state==S0) begin
    charge_ind=0;
    charge_val=3'b000;
end
else if(state==S6) begin
    charge_ind=1;
    charge_val=3'b001;
end
else if(state==S5) begin
    charge_ind=0;
    charge_val=3'b000;
end
else if(cancel_flag) begin
    charge_ind=1;
    case(state)
        S1:charge_val=3'b001;
        S2:charge_val=3'b010;
        S3:charge_val=3'b011;
        S4:charge_val=3'b100;
        default:charge_val=3'b000;
    endcase
end
else begin
    charge_ind=0;
    charge_val=3'b000;
end
```

```
endmodule
```

2. 仿真代码,注意造这种实际应用上的东西, 可以将时钟周期提高以更好地实现功能

```
`timescale 1ms / 1ns

module auto_vending_machine_tb();
    reg clk,rst_n,op_start,cancel_flag;
    reg [1:0] coin_val;
    wire hold_ind,drinktk_ind,charge_ind;
    wire [2:0] charge_val;
    auto_vending_machine inst01(
        .clk(clk),
        .rst_n(rst_n),
        .op_start(op_start),
        .cancel_flag(cancel_flag),
        .coin_val(coin_val),
        .hold_ind(hold_ind),
        .drinktk_ind(drinktk_ind),
        .charge_ind(charge_ind),
        .charge_val(charge_val)
    );
    initial clk=1;
    always #5 clk=!clk;

    initial begin
        rst_n=0;
        op_start=0;
        cancel_flag=0;
        #20;
        rst_n=1;
        #20;
        rst_n=0;

        //第一次0.5 0.5 1 0.5
        #50;
        op_start=1;
        #2
        coin_val=2'b01;
        #10;
        coin_val=2'b01;
        #10
        coin_val=2'b10;
        #10
        coin_val=2'b01;
        #10
        op_start=0;
    end
endmodule
```

```
//第二次 1 1 1
#50;
op_start=1;
#2
coin_val=2'b10;
#10;
coin_val=2'b10;
#10
coin_val=2'b10;
#10
op_start=0;

//第三次 1 0.5 cancel
#50
op_start = 1;
#2
coin_val=2'b10;
#10;
coin_val=2'b01;
#10;
cancel_flag=1;
op_start=0;
#10;
cancel_flag=0;

//第四次 1 1 cancel
#50
op_start = 1;
#2
coin_val=2'b10;
#10;
coin_val=2'b10;
#10;
cancel_flag=1;
op_start=0;
#10;
cancel_flag=0;

end
endmodule
```

P.s.该作业由markdown书写