



奋斗的小孩之 altera 系列

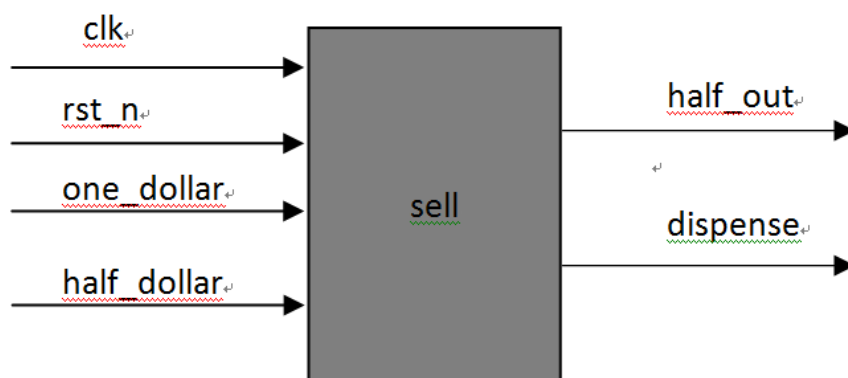
第二十四篇 自动售货机

对于每一个的小实验，我们都可以把它看作是一个小项目，逐步的去分析，设计，调试，最后完成功能。下面我们就开始我们的“小项目”。

项目名称：自动售货机

具体要求：一个饮料自动售货机，每听饮料的售价 2.5 美元，可以使用两种硬币：1 美元（one_dollar），0.5 美元（half_dollar），同时售货机有找零的功能。

架构图如下：



one_dollar: 代表投入 1 美元硬币

half_dollar: 代表投入 0.5 美元硬币



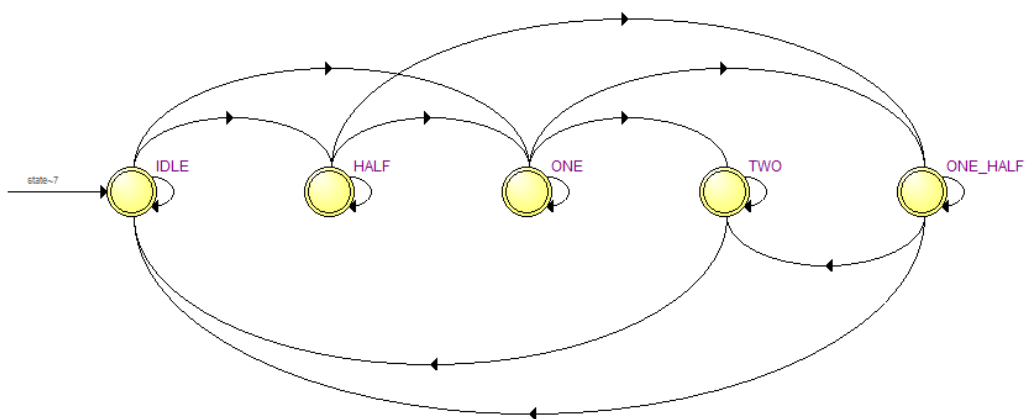
half_out: 表示找零信号

dispense: 表示机器售出一听饮料

系统设计:

1. 工程的名称: sell

2. 状态转移图如下:



IDLE: 没有钱的状态

HALF: 0.5 美元的状态

ONE: 1 美元的状态

ONE_HALF: 1.5 美元的状态

TWO: 2 美元的状态

由于此状态转移图较为简单, 故而没有写出转移条件, 读者简单分析就可理解。例如: 当前已经投入 1 美元, 若再次投入 1 美元时, 应该进入 2 美元的状态。



设计代码如下：

```
/*
```

模块名称：sell

模块功能：一个饮料自动售货机，每听饮料的售价 2.5 美元，可以使用两种硬币：1 美元 (one_dollar)，0.5 美元 (half_dollar)，同时售货机有找零的功能。

编写时间：2016-08-24

作者：至芯科技----奋斗的小孩

邮箱：zxopenhxs@126.com

```
*/
```

```
module sell (clk, rst_n, half_dollar, one_dollar, dispense,  
             half_out);
```

```
input clk;
```

```
input rst_n;
```

```
input half_dollar;
```

```
input one_dollar;
```

```
output reg dispense;
```

```
output reg half_out;
```



```
reg [2:0] state;//状态寄存器
```

```
localparam IDLE = 3'b000, //定义状态
```

```
    HALF = 3'b001,
```

```
    ONE = 3'b010,
```

```
    ONE_HALF = 3'b011,
```

```
    TWO = 3'b100;
```

```
always @ (posedge clk or negedge rst_n)
```

```
begin
```

```
    if (!rst_n)
```

```
        begin
```

```
            state <= IDLE;
```

```
            dispense <= 1'b0;
```

```
            half_out <= 1'b0;
```

```
        end
```

```
    else
```

```
        begin
```

```
            case (state)
```

```
                IDLE : begin
```



```
if (one_dollar)

    begin

        state <= ONE;

        dispense <= 1'b0;

        half_out <= 1'b0;

    end

else

    begin

        if (half_dollar)

            begin

                state <= HALF;

                dispense <= 1'b0;

                half_out <= 1'b0;

            end

        else

            begin

                state <= IDLE;

                dispense <= 1'b0;

                half_out <= 1'b0;

            end

        end

    end

end
```



end

HALF : begin

if (one_dollar)

begin

state <= ONE_HALF;

dispense <= 1'b0;

half_out <= 1'b0;

end

else

begin

if (half_dollar)

begin

state <= ONE;

dispense <= 1'b0;

half_out <= 1'b0;

end

else

begin

state <= HALF;

dispense <= 1'b0;



```
half_out <= 1'b0;

end

end

end

ONE : begin

    if (one_dollar)

        begin

            state <= TWO;

            dispense <= 1'b0;

            half_out <= 1'b0;

        end

    else

        begin

            if (half_dollar)

                begin

                    state <= ONE_HALF;

                    dispense <= 1'b0;

                    half_out <= 1'b0;

                end

            else


```



```
begin

    state <= ONE;

    dispense <= 1'b0;

    half_out <= 1'b0;

end

end

end

ONE_HALF : begin

    if (one_dollar)
        begin

            state <= IDLE;

            dispense <= 1'b1;

            half_out <= 1'b0;

        end
    else
        begin

            if (half_dollar)

                begin

                    state <= TWO;

                    dispense <= 1'b0;
```



```
half_out <= 1'b0;

end

else

begin

state <= ONE_HALF;

dispense <= 1'b0;

half_out <= 1'b0;

end

end

end

TWO : begin

if (one_dollar)

begin

state <= IDLE;

dispense <= 1'b1;

half_out <= 1'b1;

end

else

begin

if (half_dollar)
```



```
begin
    state <= IDLE;
    dispense <= 1'b1;
    half_out <= 1'b0;
end
else
begin
    state <= TWO;
    dispense <= 1'b0;
    half_out <= 1'b0;
end
end
end

default : state <= IDLE;

endcase
end
end

endmodule
```



激励代码如下：

```
/*
```

模块名称：sell_tb

模块功能：为 sell 模块提供激励信号

编写时间：2016-08-24

作者：至芯科技----奋斗的小孩

邮箱：zxopenhxs@126.com

```
*/
```

```
`timescale 1ns/1ps
```

```
module sell_tb;
```

```
    reg clk;
```

```
    reg rst_n;
```

```
    reg half_dollar;
```

```
    reg one_dollar;
```

```
    wire dispense;
```

```
    wire half_out;
```



```
initial begin
```

```
    clk = 1'b1;
```

```
    rst_n = 1'b0;
```

```
    half_dollar = 1'b0; //复位期间，不投入任何硬币
```

```
    one_dollar = 1'b0;
```

```
    # 200.1
```

```
    rst_n = 1'b1;
```

```
    # 200
```

```
    half_dollar = 1'b1; //0.5 美元
```

```
    # 20
```

```
    half_dollar = 1'b0;
```

```
    # 200
```

```
    one_dollar = 1'b1; //1 美元
```

```
    # 20
```

```
    one_dollar = 1'b0;
```

```
    # 200
```

```
    half_dollar = 1'b1; //0.5 美元
```

```
    # 20
```



```
half_dollar = 1'b0;

# 200

one_dollar = 1'b1; //1 美元

# 20

one_dollar = 1'b0;

# 2000

$stop;

end

always # 10 clk = ~clk; //系统时钟为 50MHz

sell sell_dut(
    .clk(clk),
    .rst_n(rst_n),
    .half_dollar(half_dollar),
    .one_dollar(one_dollar),
    .dispense(dispense),
    .half_out(half_out)
);
```



```
endmodule
```

解析：

由于驱动时钟（本地晶振）为 50MHz，我们外部投入硬币的信号的高电平的时间必须和驱动时钟的周期相同，若时间过短，系统采集不到外部硬币投入信号，若过长，系统会认为外部连续多次投入硬币。

所以：

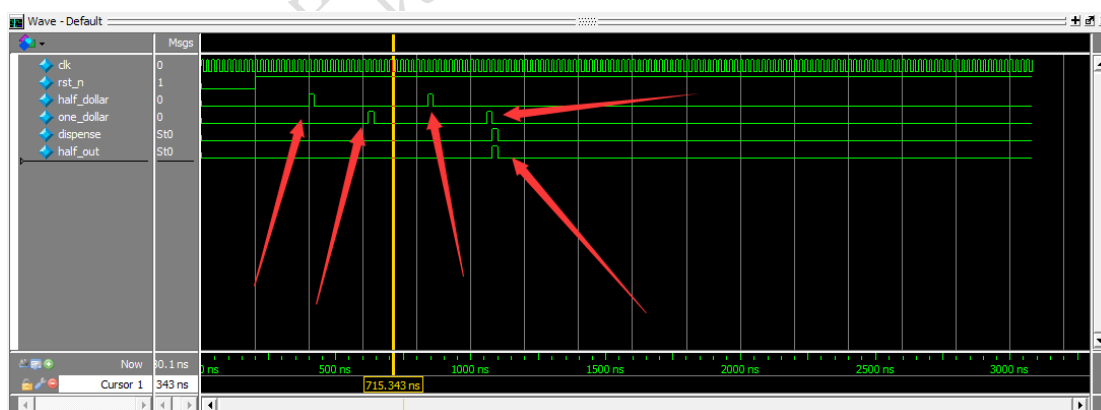
```
one_dollar = 1'b1; //1 美元
```

```
# 20 //驱动时钟为 50MHz，故而高电平持续时间为 20ns
```

```
one_dollar = 1'b0;
```

```
# 200
```

仿真波形如下：



当输入 $0.5 + 1 + 0.5 + 1 = 3$ 时，输出饮料并找零。



如果本地晶振和笔者的设计不同，请自行更改设计，以保证设计的正确性。如果还是有不明白的读者可以发邮件到我邮箱或者加群询问。

制作人: 奋斗的小孩

fpga 交流群: 282124839

至芯科技论坛—www.fpgaw.com