

奋斗的小孩之 altera 系列

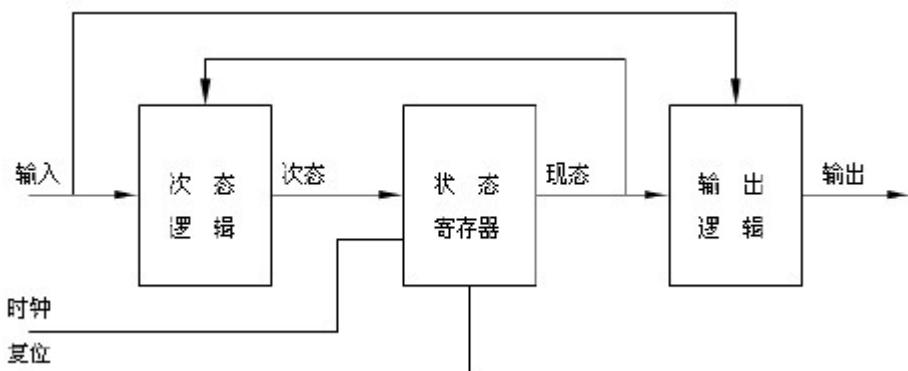
第十八篇 流水灯

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：流水灯。

具体要求：复位时，四个灯全亮。正常工作时，四个灯做流水。
每一个灯亮一秒钟。用状态机实现。

状态机的结构如下：



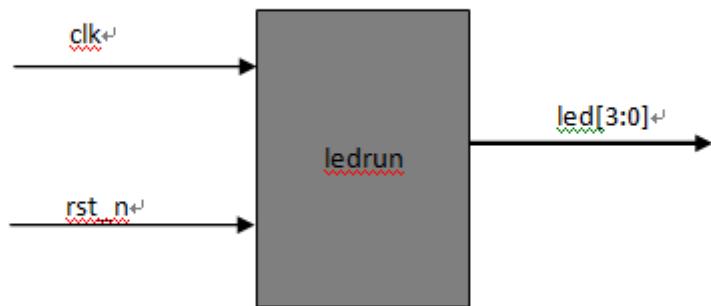
在实际的应用中，根据有限状态机是否使用输入信号，设计人员经常将其分为 Moore 型有限状态机和 Mealy 型有限状态机两种类型。

Moore 型和 Mealy 型有限状态机的区别：Moore 型有限状态机仅与当前状态有关，而与输入信号无关；Mealy 型有限状态机不但与当前

状态有关，而且还与状态机的输入信号有关。

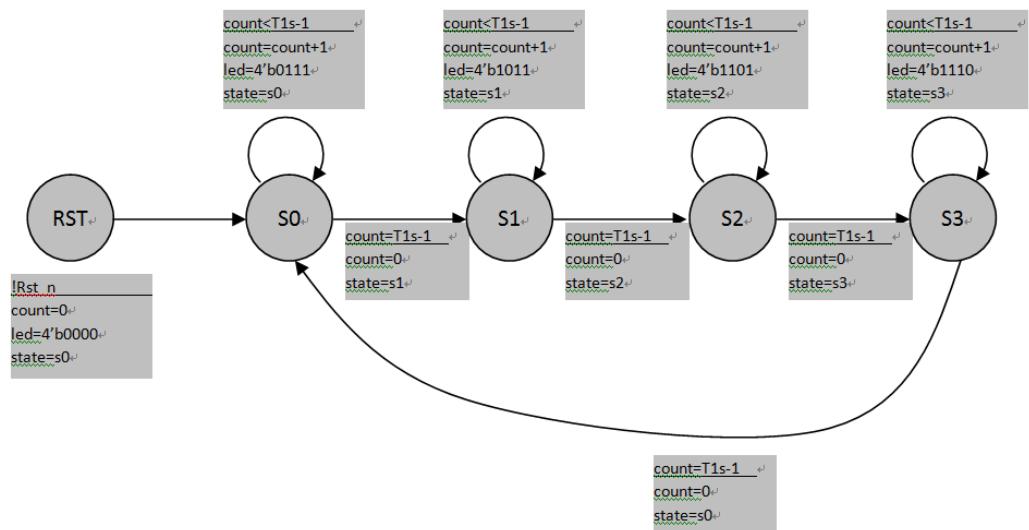
Moore 状态机是 Mealy 状态机的一种独特类型。

通过分析上述的“项目名称”和“具体要求”，我们可以设计出如下的架构：



系统设计：

1. 工程的名称：ledrun。
2. 在项目中，四个灯做流水，故而设置为四个状态。
3. 用一个计数器，计数一秒钟。
4. 状态转移图如下：



设计代码如下：

/ *

模块名称：ledrun

模块功能：四个灯做流水，每一个灯亮一秒钟

编写时间：2016-08-12

作者：至芯科技——奋斗的小孩

邮箱: zxopenhxs@126.com

* /

mod

```
module ledrun (clk, rst_n, led);
```

```
input clk;
```

```
input rst_n; //低电平有效
```

```
output reg [3:0] led;
```

```
parameter T1s = 50_000_000;//参数设置

reg [25:0] count;//计数器（需要计数 50M 个周期）

reg [1:0] state;//四个状态

localparam s0 = 2' b00, //定义四个状态机参数
         s1 = 2' b01,
         s2 = 2' b10,
         s3 = 2' b11;

always @ (posedge clk or negedge rst_n)
begin
    if (!rst_n)
        begin
            state <= s0;
            count <= 26' d0;
            led <= 4' b0000;
        end
    end
else
begin//状态机的关键字 case () ••• endcase
```

case(state) //按照设计的状态转移图进行描述

```
s0 : begin
    if (count < T1s - 1)
        begin
            led <= 4'b0111;
            state <= s0;
            count <= count + 1;
        end
    else
        begin
            count <= 0;
            state <= s1;
        end
    end

s1 : begin
    if (count < T1s - 1)
        begin
            led <= 4'b1011;
            state <= s1;
            count <= count + 1;
        end
    end
```

```
        end

        else

            begin

                count <= 0;

                state <= s2;

            end

        end

    s2 : begin

        if (count < T1s - 1)

            begin

                led <= 4'b1101;

                state <= s2;

                count <= count + 1;

            end

        else

            begin

                count <= 0;

                state <= s3;

            end

        end

    end
```

```
s3 : begin
    if (count < T1s - 1)
        begin
            led <= 4'b1110;
            state <= s3;
            count <= count + 1;
        end
    else
        begin
            count <= 0;
            state <= s0;
        end
    end
default : state <= s0;
endcase
end
end
```

endmodule

解析：

parameter T1s = 50_000_000; 参数设置。我们可以在例化时改变它。下板时，我们让参数为 50_000_000, 仿真时，我们让参数为 5. 如果仿真 50_000_000 时间将会很长，我们只做一下简单的验证就可以。

计数器计数为什么到 50_000_000 - 1? 我们的本地晶振是 50MHz, 周期为 20ns。如果计数到 1 秒钟的话, 应该需要计 50_000_000 个数。假设我们要计 5 个数, 从 0 开始计, 那么到 4 (5-1) 就可以了。

下面在激励中，笔者给大家讲解怎么样在例化时改变参数。

激励代码如下：

/*

模块名称：ledrun_tb

模块功能：为 ledrun 模块提供激励信号

编写时间：2016-08-12

作者：至芯科技----奋斗的小孩

邮箱：zxopenhxs@126. com

*/

`timescale 1ns/1ps

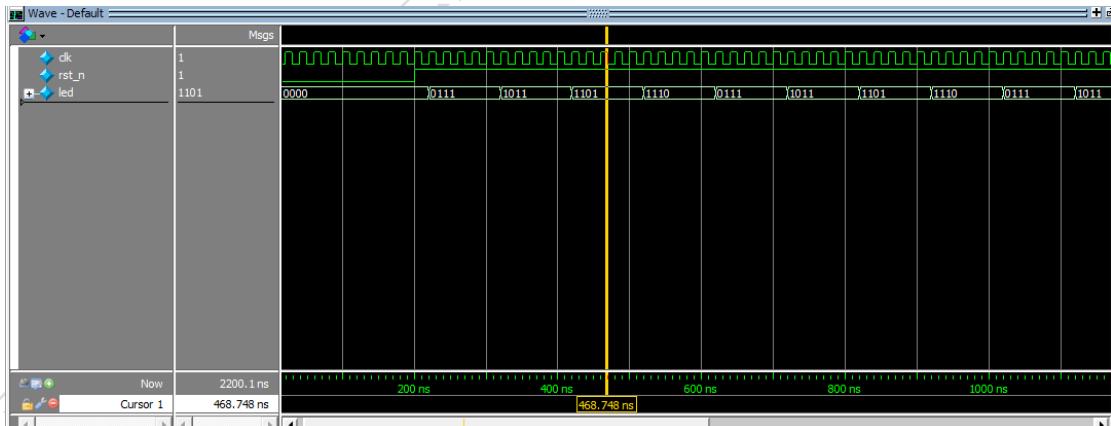
```
module ledrun_tb;  
  
    reg clk;  
  
    reg rst_n; //低电平有效  
  
    wire [3:0] led;  
  
    parameter T1s = 5;//参数设置（仿真时）  
  
    initial begin  
        clk = 1' b1;  
        rst_n = 1' b0;  
        # 200.1//复位 200.1ns  
  
        rst_n = 1' b1;  
        # 2000  
        $stop;  
    end  
  
    always #10 clk = ~clk;//50M
```

```
ledrun #(T1s) //参数传递

ledrun_dut(
    .clk(clk),
    .rst_n(rst_n),
    .led(led)
);

endmodule
```

仿真波形如下：



从仿真出来的波形来看，第一： led 灯在流水（低电平在流水）。
第二：每一个 led 灯亮 5 个周期（我们设置的仿真时是 5 个周期）。

设计正确，下板即可验证。如果小伙伴的电路原理和笔者的不一



ZHI XIN TECHNOLOGY *FPGA 培训专家* www.zxopen.com

样，请自行更改设计。如果还是有不明白的小伙伴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