

14.呼吸灯实验例程

14.1实验简介

在过去的手机上，常常看到呼吸灯的身影，它缓慢的从亮到灭、从灭到亮，像人的呼吸一样，用于提醒用户有信息或者其他情况，虽然随着科技的发展，呼吸灯逐渐淡出大众的视野，但不可否定的是它便携的表达信号的方式仍被许多人怀念。

14.2实验目的

使用 FPGA 开发板完成 LED 灯实验呼吸灯实验。

14.3实验原理

呼吸灯是一个缓慢地从灭到亮、从亮到灭周而复始的过程，我们可以使用 FPGA 开发板的 LED 灯实验这一过程，涉及到的原理为脉冲宽度调制（PWM），人眼视觉暂留特性。

占空比是指脉冲信号在一个周期内维持高电平的时间与周期的比值。



脉冲宽度调制即调整脉冲占空比的比值，占空比越大，脉冲对应电压也就越大。

根据开发板原理图知高电平时 LED 灯亮，低电平 LED 灯灭。在使用 PWM 信号对 LED 灯进行亮灭控制时，信号占空比越大，对应电压越大，LED 灯越亮，信号占空比越小，LED 灯越暗，因此 LED 灯从暗到亮的过程，是 LED 灯控制信号占空比越来越大的过程，LED 灯从亮到暗的过程，是 LED 灯控制信号占空比越来越小的过程。

值得注意的是若 PWM 信号频率太小，人眼可以观察到 LED 灯的亮灭，会观察到闪烁感，因此需要调整信号的频率，已知由于视觉暂留特性，人眼帧率为 24 帧/s，因此 PWM 信号的频率需要大于 24Hz。

我们选取 PWM 信号频率为 100Hz，在 1s 内，占空比逐渐增加，1s 内占空比逐渐减小，就可以实现呼吸灯了。

14.4实验源码

```
module breath_light(  
    input    wire    clk ,  
    input    wire    rst_n ,  
    output   wire    led  
);  
  
parameter time_1s = 26'd49_999_999 ;  
parameter time_10ms = 19'd499_999 ;  
parameter time_100us = 13'd4_999 ;  
  
reg [18:0]cnt_time_10ms ;  
reg [12:0]cnt_time_100us ;  
  
reg [6:0]cnt_10ms ;  
reg [6:0]cnt_100us ;  
  
reg flag ;  
reg breath_led ;  
  
//10ms*100=1s, The counter clocks 10ms and records the number of 10ms//  
always @(posedge clk ) begin  
    if (~rst_n) begin  
        cnt_time_10ms <= 19'd0 ;  
        cnt_10ms <= 7'd0 ;  
    end  
    else if ((cnt_10ms == 7'd99)&&(cnt_time_10ms == time_10ms)) begin  
        cnt_10ms <= 7'd0 ;  
        cnt_time_10ms <= 19'd0 ;  
    end  
    else if (cnt_time_10ms == time_10ms) begin  
        cnt_10ms <= cnt_10ms + 7'd1 ;  
        cnt_time_10ms <= 19'd0 ;  
    end  
    else begin  
        cnt_10ms <= cnt_10ms ;  
        cnt_time_10ms <= cnt_time_10ms + 19'd1 ;  
    end  
end  
  
//100us*100=10ms, The counter clocks 100us and records the number of 100us//  
always @(posedge clk ) begin  
    if (~rst_n) begin
```

```

        cnt_time_100us <= 13'd0 ;
        cnt_100us <= 7'd0 ;
    end
    else if ((cnt_100us == 7'd99)&&(cnt_time_100us == time_100us)) begin
        cnt_100us <= 7'd0 ;
        cnt_time_100us <= 13'd0 ;
    end
    else if (cnt_time_100us == time_100us) begin
        cnt_100us <= cnt_100us + 7'd1 ;
        cnt_time_100us <= 13'd0 ;
    end
    else begin
        cnt_100us <= cnt_100us ;
        cnt_time_100us <= cnt_time_100us + 13'd1 ;
    end
end
end

/*-----When the flag signal is 0, the LED light is dark to light; -----
-----when the flag signal is 1, the LED light is light to dark; -----*/
always @(posedge clk ) begin
    if (~rst_n)
        flag <= 1'b0 ;
    else if ((cnt_10ms == 7'd99)&&(cnt_time_10ms == time_10ms))
        flag <= ~flag ;
end

//-----Set the signal value to 1 every 10ms-----//
//control the signal to low level by counting the number of 10ms and 100us//
always @(posedge clk ) begin
    if (~rst_n)
        breath_led <= 1'b0 ;
    else if (cnt_time_10ms == time_10ms)
        breath_led <= 1'b1 ;
    else if (cnt_10ms + 1'b1 == cnt_100us)
        breath_led <= 1'b0 ;
end

assign led = (!flag) ? breath_led : ~breath_led ;

endmodule

```

14.5实验现象

观察到 LED0 似呼吸般平滑的由暗到亮，又由灭到亮。