

《专用集成电路设计基础》电路设计实验

实验 1——简单状态机实验

(循环彩灯控制电路)

1、电路功能

设计一套循环彩灯控制电路，要求：

- (1) 共有 16 个彩色灯组，每只彩灯用一个控制信号控制器点亮和关闭（“1”——点亮；“0”——关闭）；

注：彩灯排放顺序

D	E	F	0	1	2
C				3	
B					4
A	9	8	7	6	6

- (2) 彩灯闪烁频率可以通过编程进行控制；

- (3) 彩灯闪烁顺序及图案可以进行修改。

2、电路设计

采用 Verilog HDL 硬件描述语言进行，用 ModelSim EDA 软件进行仿真。

- (1) 彩灯控制电路设计：

```
//***** ASIC Exercise - 1 *****
//*** Counter + Decoder ***
//*** can be used to control colour lamps ***
//*** colour lamps arrangement ***
//*****
//*** d e f 0 1 2 ***
//*** c           3 ***
//*** b           4 ***
//*** a 9 8 7 6 5 ***
//*****

module exercise_1 (reset,clk,load,data_in,
                    lamp_ctl);

/* input/output ports definition *
input      reset,clk,load;
input      [3:0]   data_in;

output     [15:0]  lamp_ctl;
reg       [15:0]  lamp_ctl;

/* in-circuit signals definition *
wire      ena;
reg       [3:0]   cnt1,cnt2;
reg       [15:0]  lamp;
```

```

/* circuit RTL description *
/* 4-bit mode-16 counter with sync_data load*
/* used as freq_divider *
always @(posedge reset or posedge clk)
  if (reset)
    cnt1 <= 4'b0000;
  else if (load)
    cnt1 <= data_in;
  else if (cnt1 == 4'b1111)
    cnt1 <= data_in;
  else
    cnt1 <= cnt1 + 1'b1;

assign ena = &cnt1;

/* 4-bit mode-16 counter *
always @(posedge reset or posedge clk)
  if (reset)
    cnt2 <= 4'b0000;
  else if (ena)
    begin
      if (cnt2 == 4'b1111)
        cnt2 <= 4'b0000;
      else
        cnt2 <= cnt2 + 1'b1;
    end

/* 4-16 decoder *
always @(reset or cnt2)
begin
  if (reset)
    lamp = 16'h0000;
  else
    case (cnt2)
      4'b0000 : lamp = 16'b0000_0000_0000_0001;
      4'b0001 : lamp = 16'b0000_0000_0000_0010;
      4'b0010 : lamp = 16'b0000_0000_0000_0100;
      4'b0011 : lamp = 16'b0000_0000_0000_1000;
      4'b0100 : lamp = 16'b0000_0000_0001_0000;
      4'b0101 : lamp = 16'b0000_0000_0010_0000;
      4'b0110 : lamp = 16'b0000_0000_0100_0000;
      4'b0111 : lamp = 16'b0000_0000_1000_0000;
      4'b1000 : lamp = 16'b0000_0001_0000_0000;
      4'b1001 : lamp = 16'b0000_0010_0000_0000;
      4'b1010 : lamp = 16'b0000_0100_0000_0000;
      4'b1011 : lamp = 16'b0000_1000_0000_0000;
      4'b1100 : lamp = 16'b0001_0000_0000_0000;
      4'b1101 : lamp = 16'b0010_0000_0000_0000;
      4'b1110 : lamp = 16'b0100_0000_0000_0000;
      4'b1111 : lamp = 16'b1000_0000_0000_0000;
      default : lamp = 16'b0000_0000_0000_0000;
    endcase
  end

/* latch the output control signal *
always @(posedge reset or posedge clk)
  if (reset)
    lamp_ctl <= 16'h0000;
  else

```

```

    lamp_ctl <= lamp;
endmodule

```

(2) 测试向量设计:

```

//*** test bench ***
`timescale 1ns/1ps

module testbench ();
//*** test signal definition ***
reg      t_reset, t_clk, t_load;
reg [3:0] t_data_in;
wire [15:0] lamp_show;

//*** test signal generation ***

/* t_reset */
//*** t_reset ***
initial
begin
#0      t_reset = 1'b1;
#2000   t_reset = 1'b0;
end

//*** t_clk ***
initial
begin
t_clk = 1'b0;
end

always #500 t_clk = ~t_clk;

//*** t_ena ***
initial
begin
#0      t_load = 1'b0;
#2000   t_load = 1'b1;
#3000   t_load = 1'b0;
end

/* t_data_in */
initial
begin
#0      t_data_in = 4'b0000;
#3000   t_data_in = 4'b1000;
end

//*** connect with the circuit to be tested ***
//*** use component instantiation statement ***
exercise_1 u0 (.reset(t_reset),.clk(t_clk),.load(t_load),.data_in(t_data_in),
               .lamp_ctl(lamp_show));

endmodule

```

3、实验内容

- (1) 详细分析实验电路的设计，明确电路结构及各个功能部件的功能，并在 ModelSim EDA 软件环境下对其进行初步的仿真验证；
- (2) 更改测试文件相关部分的数值，改变彩灯的闪烁频率（与系统时钟信号 clk 进行比较）；
- (3) 将 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow \dots \rightarrow E \rightarrow F$ 的顺时针点亮方式更改为 $F \rightarrow E \rightarrow \dots \rightarrow 2 \rightarrow 1 \rightarrow 0$ 的逆时针点亮方式；
方法 1：将彩灯点亮顺序计数器改变为递减计数模式；
方法 2：你能想到采用其他什么样的电路改进方法吗？试一试!!!
- (4) 改变彩灯的显示方式，使你的彩灯点亮的与众不同！大胆尝试吧!!!

4、实验报告中需要解答的问题

- (1) 根据对实验电路的分析，绘制出彩灯控制电路详细的电路结构框图，并对每一功能部件的功能及相关参数的意义进行说明；
- (2) 采用 Verilog HDL 硬件描述语言设计一个具有上电异步复位，同步复位、同步置数及技术允许控制的 4-Bit 递增计数器；
- (3) 在上面计数器电路的基础上，说明分别实现模-4、模-9 及模-13 计数的方式与方法；
- (4) 将题 (2) 中的计数器更改为 4-bit 递减计数器；
- (5) 说明实验中彩灯闪烁方式是如何进行更改的？改进后彩灯的显示效果是什么？