

## 实验四 综合电路设计实验报告

组长：李天力

学号：1120190405

班级：07111901

手机：

组员：徐梦瑶

学号：1120190209

班级：07111901

手机：

组员：周一鸣

学号：1120192000

班级：07111904

手机：

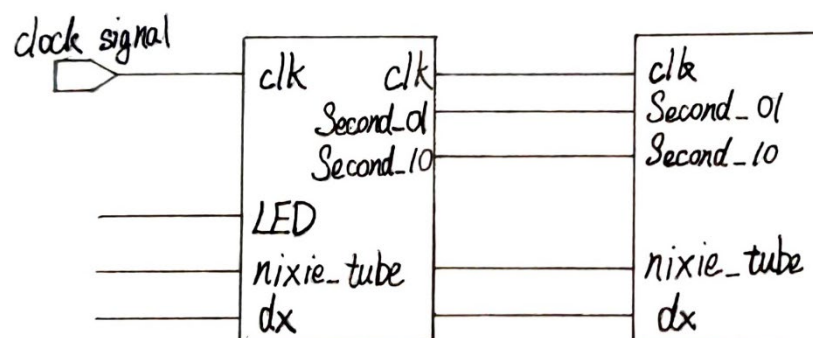
### 1. 实验题目

- 人行交通灯设计与实现（难度系数：0.8）

人行交通灯描述如下：

- “人行交通灯” 用两只不同颜色的 LED 灯显示；
- 红、绿两灯点亮时间比为 30:20；
- 红、绿两灯亮时，用两位数码管以“倒计时”方式显示剩余时间；
- 最后三秒时“闪烁”LED 灯，以表示临近结束
- 开机自动运行，显示时间单位为“秒（S）”。

### 2. 电路设计



### 3. 电路实现

顶层控制模块 Control 包含 LED 灯的控制与显示部分、计时器的控制部分和对 7 段数码管显示模块的调用。该模块的输入为板载时钟信号，输出为 LED 控制信号 LED、数码管显示信号 nixie\_tube 和数码管段选信号 dx。

顶层模块中的第一个 always 块控制寄存器 R\_Second\_01 和 R\_Second\_10。两者分别为数码管显示的个位和十位数字。寄存器 R\_current 表示当前显示的是红灯还是绿灯。第二个 always 块控制 LED 的点亮和闪烁。详见注释。

```
22 module Control(  
23     input      clk,           //clock signal from FPGA  
24     output [7:0] LED,         //LED control signal  
25     output [6:0] nixie_tube,  //nixie tube control signal  
26     output [1:0] dx           //nixie tube choosing signal  
27 );  
28  
29 reg [7:0] display;           //register of LED control signal  
30 reg [3:0] R_Second_01;       //register of the ones digit displayed by nixie tube  
31 reg [3:0] R_Second_10;       //register of the tens digit displayed by nixie tube  
32 reg [31:0] counter;          //time loop control  
33 reg      R_current;          //0 indicates red which should be displayed for 30s, 1 indicates green which should be displayed for 20s  
34  
35 parameter CLK_FREQUENCY      = 32'd1_0000_0000; //a cycle of time loop  
36 parameter LED_BLINK_FREQUENCY = 32'd5000_0000; //the frequency of which LED blinks when timer reaches the last three seconds  
37  
38 initial //starts with red light  
39 begin  
40     display      = 7'b00001111;  
41     R_Second_01  = 4'b0000;  
42     R_Second_10  = 4'b0011;  
43     R_current     = 1'b0;  
44     counter      = 32'd0;  
45 end  
46  
47 //nixie tube control part is as follows  
48 always@(posedge clk)  
49 begin  
50     if(counter == CLK_FREQUENCY && R_Second_01 > 4'b0) //ones digit minus 1  
51     begin  
52         R_Second_01 <= R_Second_01 - 1;  
53         counter <= 32'd0;  
54     end  
55     else if(counter == CLK_FREQUENCY && R_Second_01 == 4'b0000 && R_Second_10 > 4'b0000) //current ones digit equals 0, tens digit minus 1  
56     begin  
57         R_Second_10 <= R_Second_10 - 1;  
58         R_Second_01 <= 4'b1001;  
59         counter <= 32'd0;  
60     end  
61     else if(counter == CLK_FREQUENCY && R_Second_01 == 4'b0000 && R_Second_10 == 4'b0000) //all digits equals 0, switch light  
62     begin  
63         if(R_current == 1'b0) //red turns to green  
64         begin  
65             R_Second_01 <= 4'b0000;  
66             R_Second_10 <= 4'b0010;  
67             counter <= 32'd0;  
68         end  
69         else //green turns to red  
70         begin  
71             R_Second_01 <= 4'b0000;  
72             R_Second_10 <= 4'b0011;  
73             counter <= 32'd0;  
74         end  
75     end  
76 end
```

```

74   end
75   R_current <= ~R_current; //switch current light colour
76   end
77   else
78   begin
79       counter <= counter + 1;
80   end
81   end
82
83   //LED control part is as follows
84   always@(posedge clk)
85   begin
86       if(R_Second_10 > 4'b0000 || (R_Second_10 == 4'b0000 && R_Second_01 > 4'b0011))
87       begin
88           if(R_current == 1'b0)
89               display <= 8'b00001111;
90           else
91               display <= 8'b11110000;
92       end
93       else //timer reaches the last three seconds and LED starts to blink
94       begin
95           if(R_current == 1'b0 && counter <= LED_BLINK_FREQUENCY)
96               display <= 8'b00000000;
97           else if(R_current == 1'b0 && counter > LED_BLINK_FREQUENCY)
98               display <= 8'b00001111;
99           else if(R_current == 1'b1 && counter <= LED_BLINK_FREQUENCY)
100               display <= 8'b00000000;
101           else
102               display <= 8'b11110000;
103       end
104   end
105
106   //calling nixie tube display part
107   Nixie_tube tube
108   (
109       .clk          (clk),
110       .Second_01    (R_Second_01),
111       .Second_10    (R_Second_10),
112       .nixie_tube   (nixie_tube),
113       .dx           (dx)
114   );
115
116   assign LED = display;
117
118 endmodule

```

Nixie\_tube 模块中的 DISPLAY\_FREQUENCY 为分频系数。第一个 always 块控制

数码管的段选信号并选择相应数码管应显示的数字。第二个 always 块是数码管控制信

号译码器，将选定应显示的数字译成相应的 7 位控制信号。

```

22 module Nixie_tube(
23     input    clk,
24     input [3:0] Second_01,
25     input [3:0] Second_10,
26     output reg [6:0] nixie_tube,
27     output reg [1:0] dx
28 );
29
30     reg [3:0] R_temp; //current display digit
31     reg [31:0] counter; //for frequency divide control
32
33     parameter DISPLAY_FREQUECE = 32'd10_0000;
34
35     initial
36     begin
37         counter = 32'd0;
38         R_temp = 4'b0011;
39         dx = 2'b10;
40     end

```

```

42 always@(posedge clk)
43 begin
44     if(counter == DISPLAY_FREQUECE && dx == 2'b10) //display switch to ones digit
45     begin
46         dx      <= 2'b01;
47         R_temp  <= Second_01;
48         counter <= 32'd0;
49     end
50     else if(counter == DISPLAY_FREQUECE && dx == 2'b01) //display switch to tens digit
51     begin
52         dx      <= 2'b10;
53         R_temp  <= Second_10;
54         counter <= 32'd0;
55     end
56     else
57     begin
58         counter <= counter + 1;
59     end
60 end
61

```

## 4. 电路验证

### a) TestBench

```

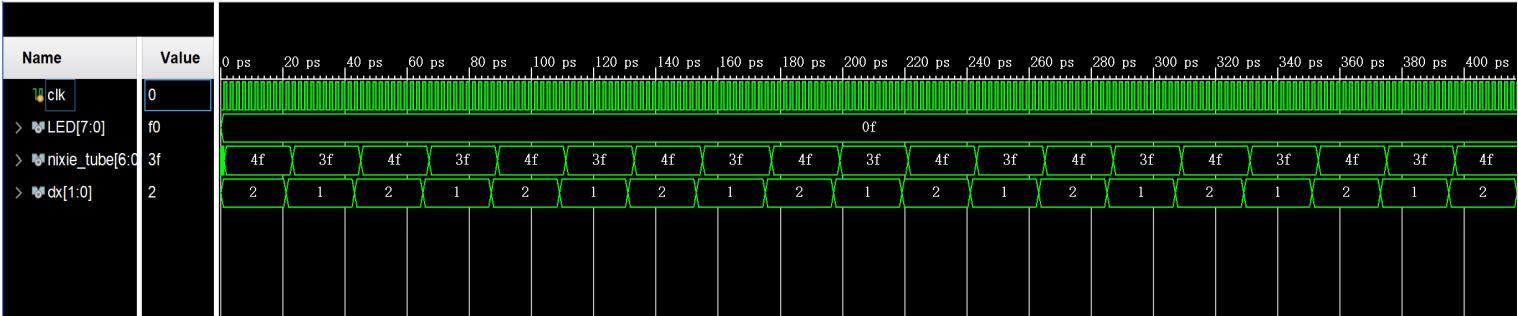
23 module testbench();
24
25     reg      clk;
26     wire [7:0] LED;
27     wire [6:0] nixie_tube;
28     wire [1:0] dx;
29
30     initial
31         clk = 1'b0;
32
33     always #0.001 clk = ~clk;
34
35     Control testbench
36     (
37         .clk      (clk),
38         .LED      (LED),
39         .nixie_tube (nixie_tube),
40         .dx      (dx)
41     );
42
43 endmodule

```

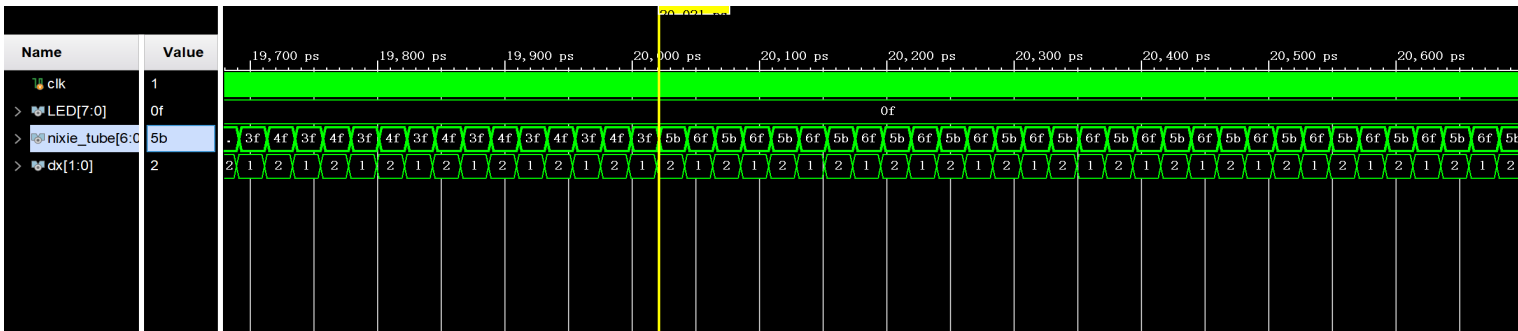
由于输入只有时钟信号 `clk`，所以仿真文件只需要对时钟信号进行模拟。此处设置时钟周期为 0.002ns，时钟信号每隔 0.001ns 跳变一次。在仿真时为了加快仿真信号变化速度，对所有源文件中的时间周期数值进行了缩放，缩小为下板所需的

$$\frac{1}{10000}^{\circ}$$

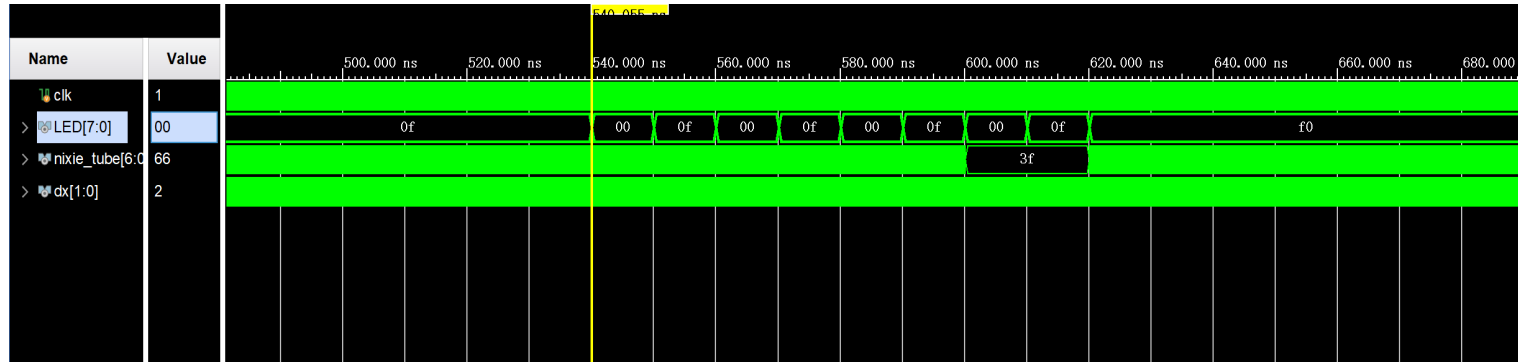
b) 仿真结果



在最初的 400ps 中，由于是模拟红灯，LED 信号恒为 0f，数码管信号为 4f 和 3f，即 30。段选信号以 40ps 为一周期反复跳变。



在 20000ps 左右，模拟的实际情况为 30s 跳变为 29s，数码管信号跳变为 5b 和 6f。



从 540ns 左右起，LED 信号开始呈周期性变化，模拟的实际情况为红灯进入最后三秒，信号灯闪烁。然后 LED 信号由 0f 变为 f0，表示红灯变为绿灯。

5. 电路上板

Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	
All ports (18)							
dx (2)	OUT			<input checked="" type="checkbox"/>	35	LVC MOS33*	-
dx[1]	OUT		C2	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
dx[0]	OUT		G2	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED (8)	OUT			<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[7]	OUT		K2	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[6]	OUT		J2	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[5]	OUT		J3	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[4]	OUT		H4	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[3]	OUT		J4	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[2]	OUT		G3	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[1]	OUT		G4	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
LED[0]	OUT		F6	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube (7)	OUT			<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[6]	OUT		B2	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[5]	OUT		B3	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[4]	OUT		A1	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[3]	OUT		B1	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[2]	OUT		A3	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[1]	OUT		A4	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
nixie_tube[0]	OUT		B4	<input checked="" type="checkbox"/>	35	LVC MOS33*	-
Scalar ports (1)							
clk	IN		T5	<input checked="" type="checkbox"/>	34	LVC MOS33*	-

dx 信号的分别对应最右侧两个数码管的段选管脚；LED 信号的对应 8 个 LED 的控制管脚；nixie\_tube 对应数码管显示的管脚；clk 对应板载时钟的输入管脚。

最终上板情况符合预期，贴合实验要求。

6. 实验心得

由于实验进行得比较仓促，很多相关知识都是现学现用，实验过程中出现过许多纰漏。

首先在实验中误以为只需要控制所使用的 LED 灯，因此最初只对 8 个 LED 灯中的 2 个分配了控制信号和相应管脚，导致在综合的时候出现了错误。在仔细查看了报错信息后对 8 个 LED 灯都分配了控制信号并解决了问题。

其次在设定数码管的频分系数时，由于对数码管的原理不熟，误以为应当设置尽可能小的频分系数，导致数码管出现了全亮的问题。在求助了学长后得知由于数码管的电

平变化速度跟不上切换频率导致出现了两个数码管的所有段同时点亮但仍有明暗的区别。在增大频分系数后解决了问题。

本次实验使我充分认识到了理论和实践之间的差距。在理论上可行的东西，在具体实现时仍有可能出现极大的偏差。