EDA 大作业

学生姓名：吴程锴

学　　号：18029100040

班　　级：1802015

授课教师：宗　汝

提交日期：2020 年 11 月 12 日

目录

一、实验目的

利用所学的 EDA 知识，制作一个交通控制器 。

二、设计任务

设计一个十字路口交通控制系统，其东西、南北两个方向除了有红、黄、绿灯指示是否允许通行外，还设有时钟，以倒计时方式显示每一路允许通行的时间，绿灯，黄灯，红灯的持续时间分别是 40、5 和 45 秒，且在绿灯的最后 5 秒让绿灯闪烁，提醒司机将要不能通行，红灯的最后 5 秒让黄灯也亮，提醒司机准备通行。当东西或南北两路中任一道上出现紧急情况，此时交通控制系统应可进入紧急状态，即两条道上的所有车辆皆停止通行，红灯全亮，时钟停止计时，且其数字在闪烁。当紧急状态结束后，控制系统恢复原来的状态，继续正常运行。

三、程序设计

3.1 分频器

输入：时钟信号，重置信号。

输出：分频时钟信号。

功能：50MHz 分频。

由于板子的时钟频率为 50MHz，为便于观察，需要设计一个 50M 的分频器，输出一个 1Hz 的时钟信号。

分频器其实就是计数器，只要计数器计数到某一模值时，把计数器清零，并且对输出进行反转即可。

分频器模值、系统时钟和期望输出时钟频率关系为

$$\text{分频器模值}=\frac{\text{系统时钟频率/期望输出时钟频率}}{2}\text{-1。}$$

所以，把 50MHz 时钟分频，输出 1Hz 的时钟，分频器的模值为

$$M=\frac{50000000/1}{2}\text{-1=24999999}$$

为了保证分频器正常工作，计数器寄存器所能表示的最大值必须大于分频器的模值。这里，设置把计数器寄存器的位数设定为 32 位，计数器寄存器可表示的最大数值为 2^{32}-1=4294967295>24999999 。具体代码见附录一。流程图如图 1 所示。

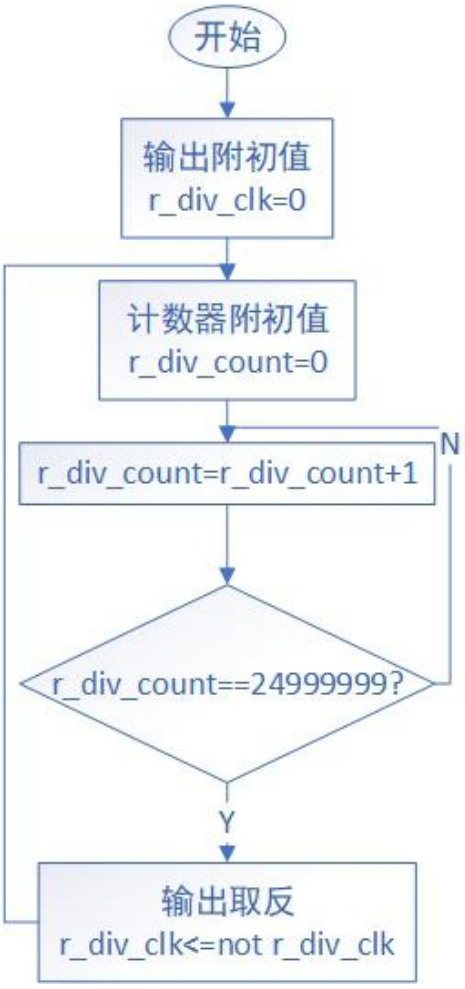

图 1 分频器

3.2 45 秒倒计时

输入：时钟信号，重置信号，紧急信号

输出：十位 8421BCD 码，个位 8421BCD 码，交换通行信号

功能：44 计数到 0，可通过紧急信号暂停计数或通过重置信号使计数清零。

一个方向的红绿灯红灯 45 秒，绿灯 40 秒，黄灯 5 秒，一直重复。我设计了一个 45 秒计时器。当紧急信号为 0 时，计时器从 45 倒计时到 0，当紧急信号为 1 时，计时器停止计时。倒计时过程中输出倒计时的十位和个位的 8421BCD 码。当计数器计数到 0 时，输出一个周期高电平，代表需要交换通行方向。具体代码见附录二。流程图如图 2 所示。

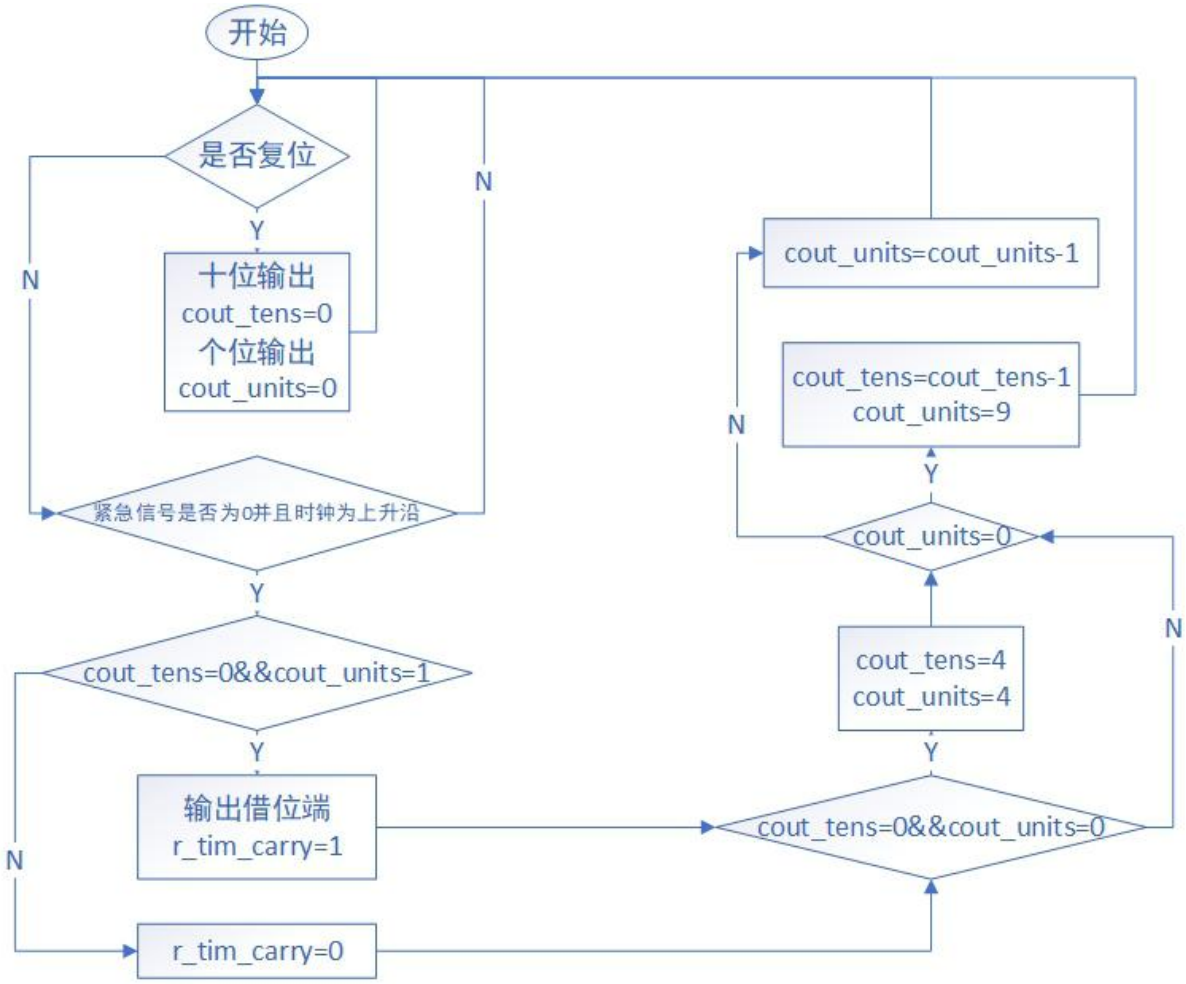

图 2 45 秒倒计时

3.3 数码管译码器

输入：数字的 8421BCD 码，重置信号，使能信号。

输出：数码管 7 段编码。

功能：把 8421BCD 码转化为数码管编码输出。

为了实现输入 8421BCD 码，时数码管显示相应的数字，需要对 8421BCD 码进行译码，使用 switch-case 语句，当输入一个 8421BCD 码时，输出相应的数码管的编码。输入还有一个使能信号，当使能为 1 时，输出对应的数码管编码，当是能为 0 时，输出全 1，时数码管熄灭。具体代码见附录三。流程图如图 3 所示。

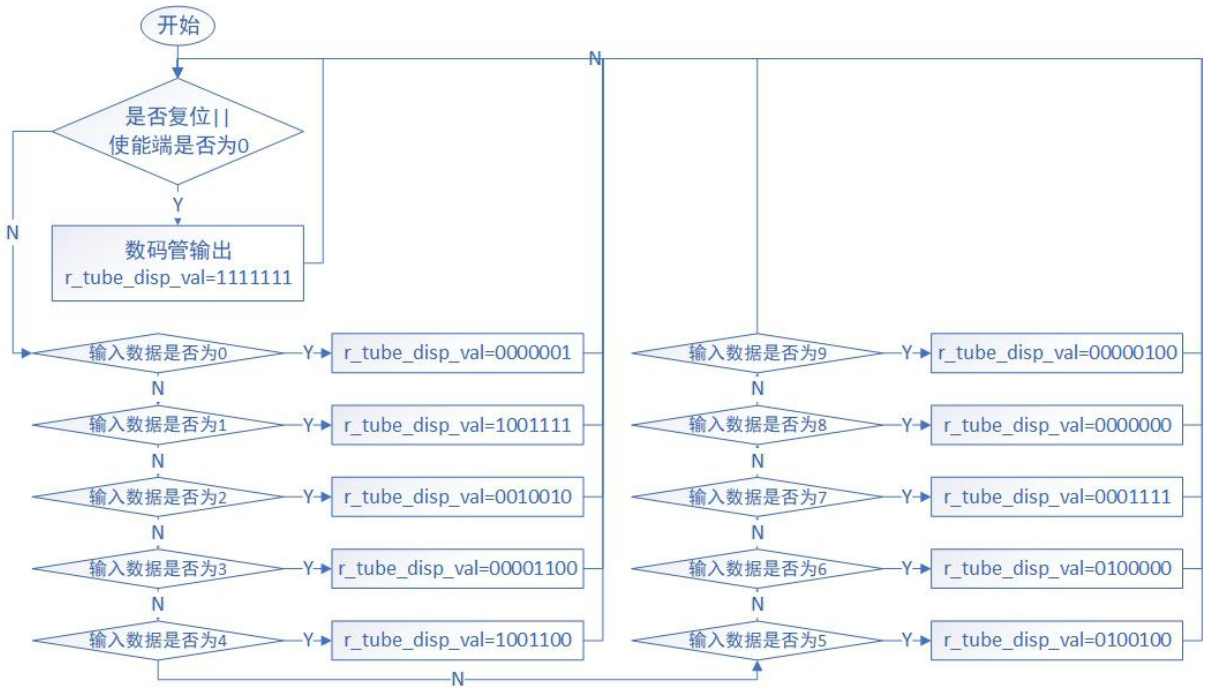

图 3 数码管显示

3.4 总控制器

输入：重置信号，时钟信号，倒计时十位 8421BCD 码，倒计时个位 8421BCD 码，交换通行信号，紧急信号。

输出：东西向十位 8421BCD 码，东西向个位 8421BCD 码，南北向十位 8421BCD 码，南北向个位 8421BCD 码，数码管使能信号，两个方向的红黄绿信号。

功能：根据从倒计时模块输入的数值和交换通行信号和规则以及紧急信号，按照规则控制 6 个灯和 4 个数码管的亮灭和闪烁。

主要使用 if 语句实现。

设一个允许通行方向标志位，当标志位为 1 时，表示东西方向允许通行，当标志位为 0 时，表示南北方向允许通行，标志位的切换由 45 秒计时器输出的交换通行信号的下降沿来决定，当 45 秒计时器输出一个周期高电平，下降沿到达时标志位取反。

当紧急信号为 0 时，在允许通行的方向的 45 秒内，前 40 秒绿灯亮，数码管使能端为 1，输出的显示从 39 倒计时到 0 的十位和个位的 8421BCD 码；在最后 5 秒，只有黄灯亮，数码管使能端为时钟信号，输出从 4 倒计时到 0 的十位和个位的 8421BCD 码。在禁止通行的方向的 45 秒内，输出 44 倒计时到 0 的十位和个位的 8421BCD 码；在前 40 秒，红灯常亮，数码管使能端为 1；在最后 5 秒，红灯和黄灯都亮，数码管使能端为时钟信号，让数码管闪烁。

当紧急信号为 1 时，数码管使能端和两个方向的红灯均为时钟信号，让他们闪烁，其他灯熄灭。

具体代码见附录四。流程图如图 4 所示。

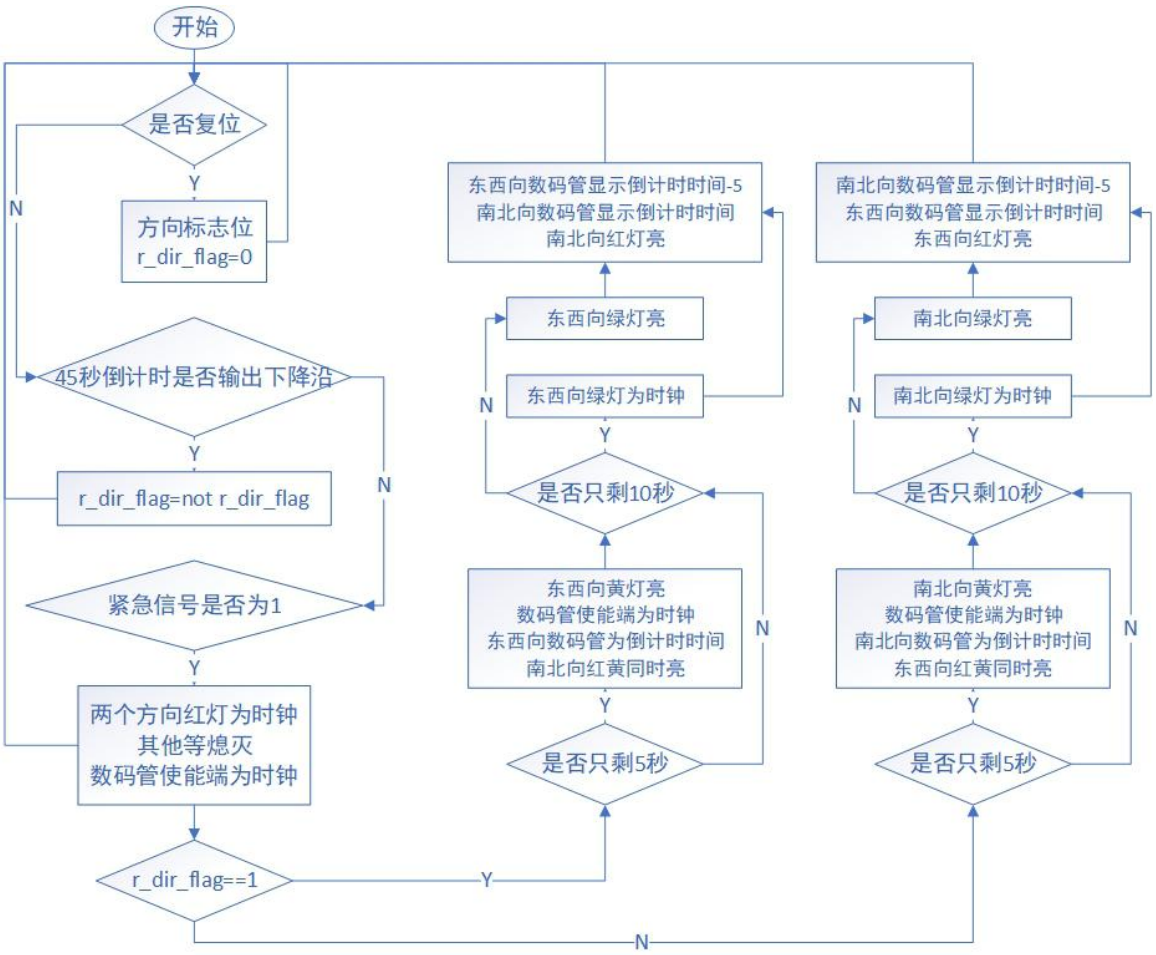

图 4 总控制器

3.5 顶层设计

把以上四个代码生成模块，并按照逻辑连接，连接如图 5 所示

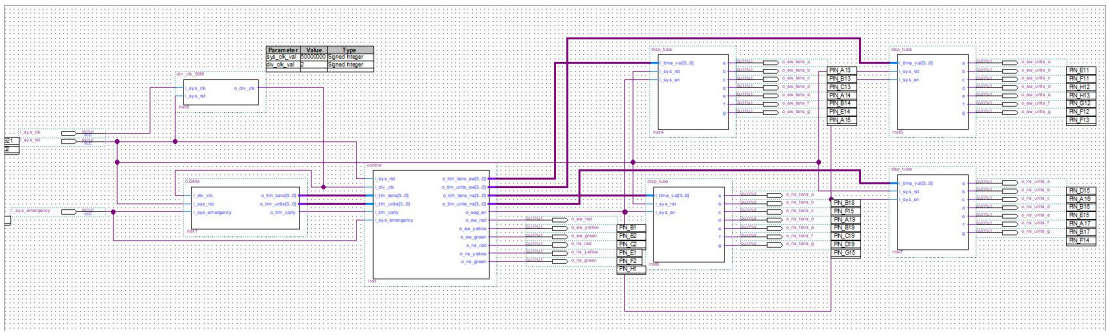

图 5 顶层设计

由于太长，把图从中间截为两张，如图 6 和图 7 所示

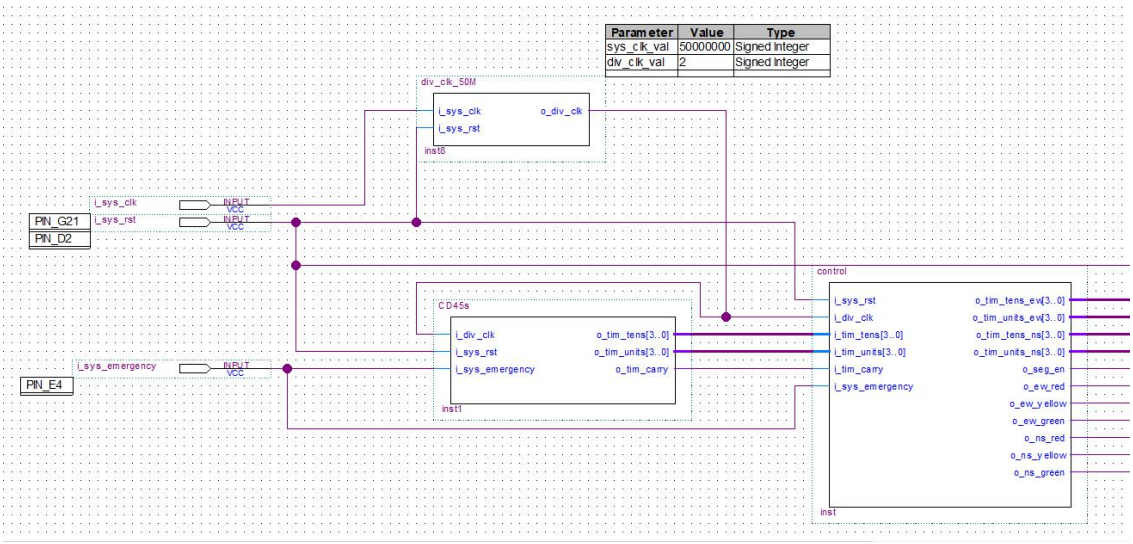

图 6 分频器、倒计时、总控制器

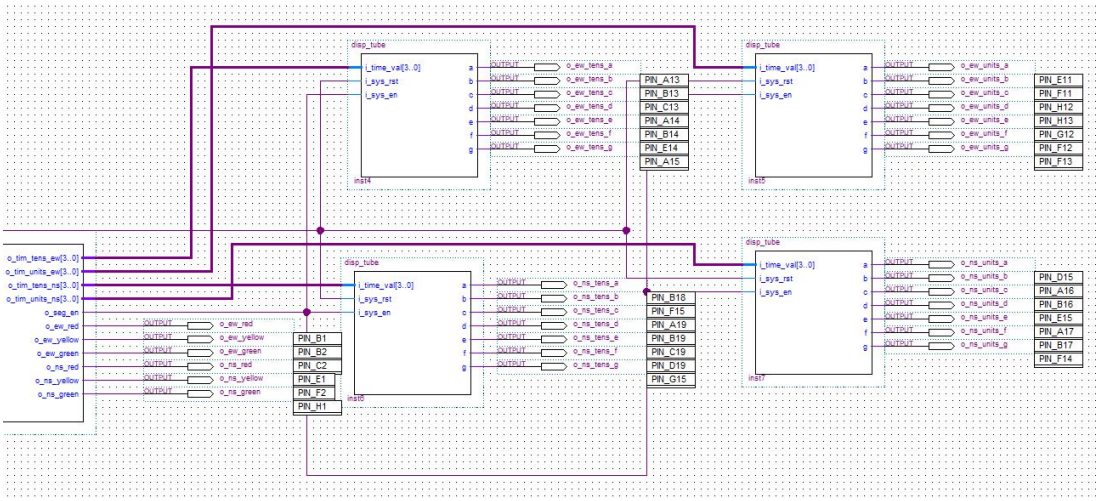

图 7 数码管显示

3.6 引脚分配

根据开发手册配置相关引脚如图 8 所示。

i_sys_clk	Input	PIN_G21
i_sys_emergency	Input	PIN_E4
i_sys_rst	Input	PIN_D2
o_ew_green	Output	PIN_C2
o_ew_red	Output	PIN_B1
o_ew_tens_a	Output	PIN_F13
o_ew_tens_b	Output	PIN_F12
o_ew_tens_c	Output	PIN_G12
o_ew_tens_d	Output	PIN_H13
o_ew_tens_e	Output	PIN_H12
o_ew_tens_f	Output	PIN_F11
o_ew_tens_g	Output	PIN_E11
o_ew_units_a	Output	PIN_A15
o_ew_units_b	Output	PIN_E14
o_ew_units_c	Output	PIN_B14
o_ew_units_d	Output	PIN_A14
o_ew_units_e	Output	PIN_C13
o_ew_units_f	Output	PIN_B13
o_ew_units_g	Output	PIN_A13
o_ew_yellow	Output	PIN_B2
o_ns_green	Output	PIN_H1
o_ns_red	Output	PIN_E1
o_ns_tens_a	Output	PIN_F14
o_ns_tens_b	Output	PIN_B17
o_ns_tens_c	Output	PIN_A17
o_ns_tens_d	Output	PIN_E15
o_ns_tens_e	Output	PIN_B16
o_ns_tens_f	Output	PIN_A16
o_ns_tens_g	Output	PIN_D15
o_ns_units_a	Output	PIN_G15
o_ns_units_b	Output	PIN_D19
o_ns_units_c	Output	PIN_C19
o_ns_units_d	Output	PIN_B19
o_ns_units_e	Output	PIN_A19
o_ns_units_f	Output	PIN_F15
o_ns_units_g	Output	PIN_B18
o_ns_yellow	Output	PIN_F2
<<new node>>		

图 8 配置引脚

四、波形仿真

由于仿真的原因，需要把 50M 分频器去除。

在命令行输入代码如下

```
1. force i_sys_clk 0 0,1 10 -r 20
2. force i_sys_rst 1 1,0 10 -r 200000000
3. force i_sys_emergency 1 1,0 500 -r 20000
4. run 30ns
```

4.1 引脚说明

i_sys_rst：重置信号

i_sys_clk：时钟信号

i_sys_emergency：紧急信号

o_xx_yy：红绿灯，其中 xx=ew,ns 表示东西或南北方向，yy=red,yellow,green 表示红黄绿灯

o_xx_yy_zz：数码倒计时，其中 xx=ew,ns 表示东西或南北方向，yy=tens,units 表示十位和个位，zz=a,b,c,d,e,f,g 表示数码管输出的 7 个段。

4.2 无紧急状况

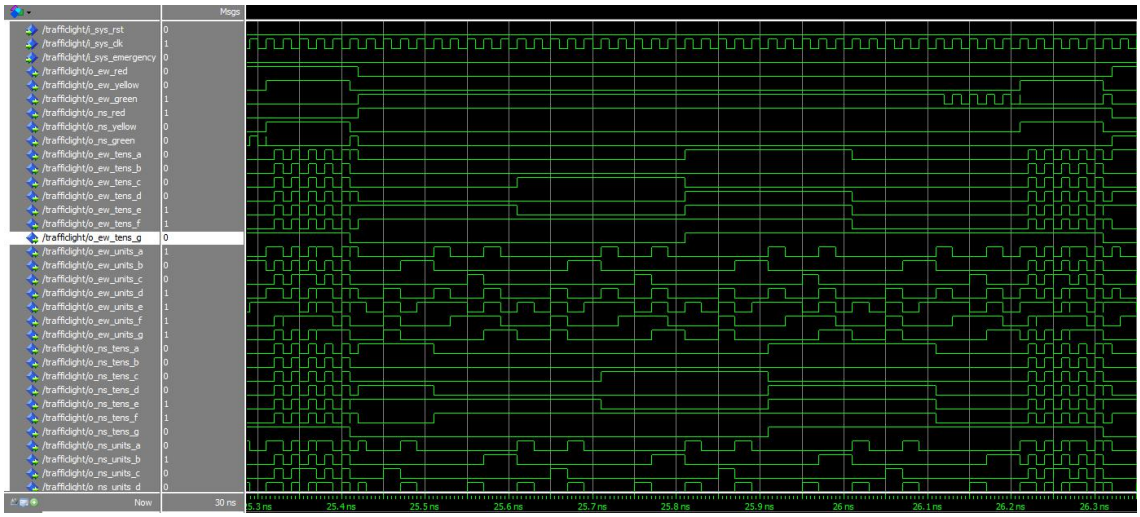

图 9 无紧急状况波形仿真

如图 9 可以看出从南北向变为红灯后，东西向由红灯变为绿灯，经过 35 秒，绿灯开始闪烁，到达 40 秒后绿灯熄灭黄灯亮起，这期间南北向红灯一直亮且在 40 秒后黄灯也亮起。再过 5 秒后，东西向由黄灯变为红灯，南北向由红灯和黄灯变为绿灯。这一期间东西向的数码由 39 减小到 00，又从 04 码减小到 00；南北向的数码从 44 减小到 00。在最后 5 秒 4 个数码管都开始闪烁。

4.3 出现紧急状况

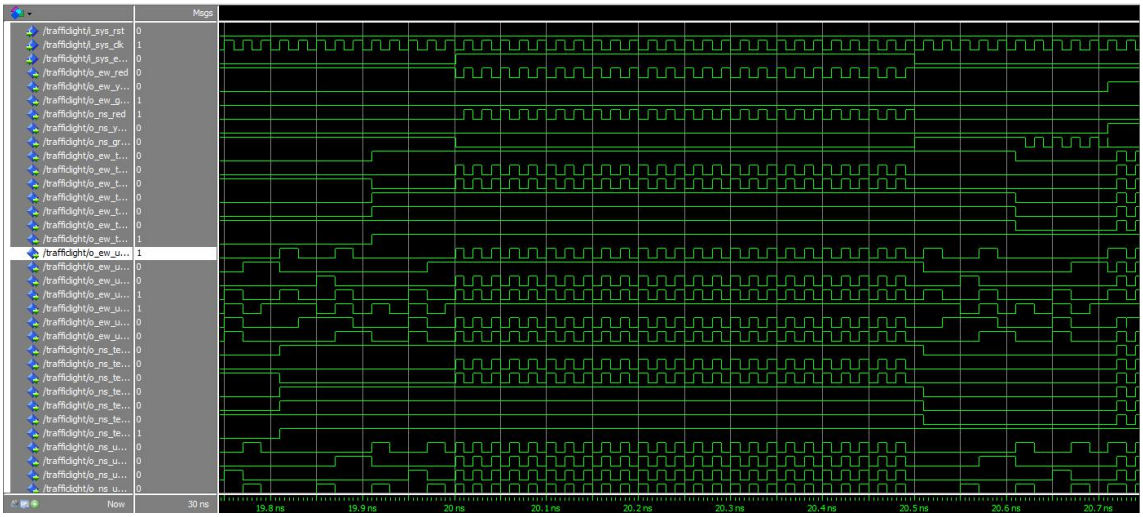

图 10 紧急状况波形仿真

如图 10 可以看出，当 i_sys_emergency 从 0 变为 1 时，东西和南北方向都变为红灯且跟随时钟信号开始闪烁，数码停止改变并闪烁。当 i_sys_emergency 变为 0 后，恢复紧急前的状况。

五、分析与总结

这次实验完成了设计任务，且在实验板上顺利运行，没有任何 bug。

通过这次设计，我对 VHDL 编程更加得熟练，掌握了基本的语法，学会通过波形仿真来修改 bug。

六、附录

附录一

```
1.  LIBRARY IEEE;
2.  USE IEEE.std_logic_1164.all;
3.  USE IEEE.std_logic_arith.all;
4.  USE IEEE.std_logic_unsigned.all;
5.
6.  --分频器
7.  entity div_clk_50M is
8.      generic(
9.          sys_clk_val:INTEGER:=50000000;
10.         div_clk_val:INTEGER:=1
11.     );
12.     port(
13.         i_sys_clk:in std_logic;
14.         i_sys_rst:in std_logic;
15.         --i_sys_emergency:in std_logic;
16.         o_div_clk:out std_logic
17.     );
18. end entity div_clk_50M;
19.
20. architecture behavior of div_clk_50M is
21.     signal r_div_count:std_logic_vector(31 downto 0);
22.     signal r_div_clk:std_logic;
23.     begin
24.     process(i_sys_rst,i_sys_clk)
25.         begin
26.             if(i_sys_rst='1')then--复位键
27.                 r_div_count<=x"00000000";
28.                 r_div_clk<='0';
29.             elsif(i_sys_clk'event AND i_sys_clk='1')then--上升沿计数
30.                 if(r_div_count=sys_clk_val/div_clk_val/2-1)then--到达 24999999 清零，输
    出反相
31.                     r_div_count<=x"00000000";
32.                     r_div_clk<=NOT r_div_clk;
33.                 else
34.                     r_div_count<=r_div_count+1;
35.                 end if;
36.             end if;
37.     end process;
38.     o_div_clk<=r_div_clk;
39. end architecture behavior;
```

附录二

```
1.  LIBRARY IEEE;
2.  USE IEEE.std_logic_1164.all;
3.  USE IEEE.std_logic_arith.all;
4.  USE IEEE.std_logic_unsigned.all;
5.
6.  --倒计时 45 秒
7.  entity CD45s is
8.      port(
9.          i_div_clk:in std_logic;                                  --1hz 时钟输入
10.         i_sys_rst:in std_logic;                                  --复位
11.         i_sys_emergency:in std_logic;                            --紧急状态
12.         o_tim_tens:out std_logic_vector(3 downto 0);     --输出倒计时十位 BCD
13.         o_tim_units:out std_logic_vector(3 downto 0);--输出倒计时个位 BCD
14.       o_tim_carry:out std_logic                                --输出进位
15.     );
16. end entity CD45s;
17.
18. architecture behavior of CD45s is
19.     signal cout_tens,cout_units:std_logic_vector(3 downto 0);
20.     signal r_tim_carry:std_logic;
21.     begin
22.         process(i_sys_rst,i_div_clk,i_sys_emergency)
23.             begin
24.                 if i_sys_rst='1'then
25.                     cout_tens<="0000";
26.                     cout_units<="0000";
27.                 elsif(i_div_clk'event AND i_div_clk='1')then
28.                     if i_sys_emergency='0'then
29.
30.                         if(cout_tens=0 AND cout_units=1)then
31.                             r_tim_carry<='1';
32.                         else
33.                             r_tim_carry<='0';
34.                         end if;
35.
36.                         if(cout_tens=0 AND cout_units=0)then
37.                             cout_tens<="0100";
38.                             cout_units<="0100";
39.                             --r_tim_carry<='1';
40.                         elsif(cout_units=0)then
41.                             cout_tens<=cout_tens-1;
42.                             cout_units<="1001";
```

```
                            --r_tim_carry<='0';
                        else
                            cout_units<=cout_units-1;
                            --r_tim_carry<='0';
                        end if;
                    end if;
                end if;
        end process;
        o_tim_tens<=cout_tens;
        o_tim_units<=cout_units;
        o_tim_carry<=r_tim_carry;
end architecture behavior;
```

附录三

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
USE IEEE.std_logic_unsigned.all;

--七段数码管显示
entity disp_tube is
    port(
        i_time_val:in std_logic_vector(3 downto 0);
        i_sys_rst:in std_logic;
        i_sys_en:in std_logic;
        a,b,c,d,e,f,g:out std_logic
        --o_tube_disp_val:out std_logic_vector(6 downto 0)
    );
end entity disp_tube;

architecture behavior of disp_tube is
    signal r_tube_disp_val:std_logic_vector(6 downto 0);
    begin
    process(i_sys_rst,i_time_val)
        begin
            if(i_sys_en='1')then
                if(i_sys_rst='1')then
                    r_tube_disp_val<="1111111";
                else
                    case i_time_val is
                        when "0000"=>r_tube_disp_val<="1000000";
                        when "0001"=>r_tube_disp_val<="1111001";
                        when "0010"=>r_tube_disp_val<="0100100";
```

```
                        when "0011"=>r_tube_disp_val<="0110000";
                        when "0100"=>r_tube_disp_val<="0011001";
                        when "0101"=>r_tube_disp_val<="0010010";
                        when "0110"=>r_tube_disp_val<="0000010";
                        when "0111"=>r_tube_disp_val<="1111000";
                        when "1000"=>r_tube_disp_val<="0000000";
                        when "1001"=>r_tube_disp_val<="0010000";
                        when others=>r_tube_disp_val<="1111111";
                    end case;
                end if;
            else
                r_tube_disp_val<="1111111";
            end if;
    end process;
    a<=r_tube_disp_val(6);
   b<=r_tube_disp_val(5);
   c<=r_tube_disp_val(4);
   d<=r_tube_disp_val(3);
   e<=r_tube_disp_val(2);
   f<=r_tube_disp_val(1);
   g<=r_tube_disp_val(0);
    --o_tube_disp_val<=r_tube_disp_val;
end architecture behavior;
```

附录四

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
USE IEEE.std_logic_unsigned.all;

entity control is
    port(
        i_sys_rst:in std_logic;
        i_div_clk:in std_logic;                                    --1hz 时钟输入
        i_tim_tens:in std_logic_vector(3 downto 0);      --输入倒计时十位 BCD
        i_tim_units:in std_logic_vector(3 downto 0);         --输入倒计时个位 BCD
      i_tim_carry:in std_logic;                                    --输入进位
        i_sys_emergency:in std_logic;                              --紧急状态
        o_tim_tens_ew:out std_logic_vector(3 downto 0); --输出东西倒计时十位 BCD
        o_tim_units_ew:out std_logic_vector(3 downto 0);--输出东西倒计时个位 BCD
        o_tim_tens_ns:out std_logic_vector(3 downto 0); --输出南北倒计时十位 BCD
        o_tim_units_ns:out std_logic_vector(3 downto 0);--输出南北倒计时个位 BCD
        o_seg_en:out std_logic;
```

```
19.         o_ew_red,o_ew_yellow,o_ew_green,o_ns_red,o_ns_yellow,o_ns_green:out std_logi
    c--输出东西南北红绿灯
20.
21.     );
22. end entity control;
23.
24. architecture behavior of control is
25.     signal r_tim_tens_ew:std_logic_vector(3 downto 0);
26.     signal r_tim_units_ew:std_logic_vector(3 downto 0);
27.     signal r_tim_tens_ns:std_logic_vector(3 downto 0);
28.     signal r_tim_units_ns:std_logic_vector(3 downto 0);
29.     signal r_dir_flag:std_logic;                                    --方向标志位
30.     signal r_ew_red,r_ew_yellow,r_ew_green,r_ns_red,r_ns_yellow,r_ns_green:std_logic;
    --输出东西南北红绿灯
31.     signal r_seg_en:std_logic;
32.     begin
33.     process(i_sys_rst,i_div_clk,i_tim_tens,i_tim_units,i_tim_carry,i_sys_emergency)

34.         begin
35.             if(i_sys_rst='1')then
36.                 r_dir_flag<='0';
37.             else
38.                 if(i_tim_carry'event AND i_tim_carry='0')then--交换通行
39.                     r_dir_flag<=NOT r_dir_flag;
40.                 end if;
41.                 if(i_sys_emergency='1')then
42.                     r_seg_en<=i_div_clk;
43.                     --r_ew_red<='1';
44.                     r_ew_red<=i_div_clk;
45.                     r_ew_yellow<='0';
46.                     r_ew_green<='0';
47.                     --r_ns_red<='1';
48.                     r_ns_red<=i_div_clk;
49.                     r_ns_yellow<='0';
50.                     r_ns_green<='0';
51.                 else
52.                     r_seg_en<='1';
53.                     if(r_dir_flag='1')then--东西通行
54.                         if(i_tim_tens=0 AND i_tim_units<5)then
55.                             r_ew_red<='0';
56.                             r_ew_yellow<='1';
57.                             r_ew_green<='0';
58.                             r_tim_tens_ew<=i_tim_tens;
59.                             r_tim_units_ew<=i_tim_units;
60.                             r_seg_en<=i_div_clk;
```

```
61.
62.                             r_ns_red<='1';
63.                             r_ns_yellow<='1';
64.                             r_ns_green<='0';
65.                         else
66.                             r_ew_red<='0';
67.                             r_ew_yellow<='0';
68.                             if(i_tim_tens=0 AND i_tim_units<=9)then--最后 5 秒绿灯闪
    烁
69.                                 r_ew_green<=i_div_clk;
70.                             else
71.                                 r_ew_green<='1';
72.                             end if;
73.
74.                             if(i_tim_units<5)then
75.                                 r_tim_tens_ew<=i_tim_tens-1;
76.                                 r_tim_units_ew<=i_tim_units+5;
77.                             else
78.                                 r_tim_tens_ew<=i_tim_tens;
79.                                 r_tim_units_ew<=i_tim_units-5;
80.                             end if;
81.                             r_ns_red<='1';
82.                             r_ns_yellow<='0';
83.                             r_ns_green<='0';
84.                         end if;
85.                         r_tim_tens_ns<=i_tim_tens;
86.                         r_tim_units_ns<=i_tim_units;
87.
88.                     else--南北通行
89.                         if(i_tim_tens=0 AND i_tim_units<5)then
90.                             r_ns_red<='0';
91.                             r_ns_yellow<='1';
92.                             r_ns_green<='0';
93.                             r_tim_tens_ns<=i_tim_tens;
94.                             r_tim_units_ns<=i_tim_units;
95.                             r_seg_en<=i_div_clk;
96.
97.                             r_ew_red<='1';
98.                             r_ew_yellow<='1';
99.                             r_ew_green<='0';
100.                         else
101.                             r_ns_red<='0';
102.                             r_ns_yellow<='0';
103.                             if(i_tim_tens=0 AND i_tim_units<=9)then--最后 5 秒绿灯闪
    烁
```

```
                                r_ns_green<=i_div_clk;
                            else
                                r_ns_green<='1';
                            end if;
                            if(i_tim_units<5)then
                                r_tim_tens_ns<=i_tim_tens-1;
                                r_tim_units_ns<=i_tim_units+5;
                            else
                                r_tim_tens_ns<=i_tim_tens;
                                r_tim_units_ns<=i_tim_units-5;
                            end if;
                            r_ew_red<='1';
                            r_ew_yellow<='0';
                            r_ew_green<='0';
                        end if;
                        r_tim_tens_ew<=i_tim_tens;
                        r_tim_units_ew<=i_tim_units;
                    end if;
                end if;
            end if;
    end process;
    o_tim_tens_ew<=r_tim_tens_ew;
    o_tim_units_ew<=r_tim_units_ew;
    o_tim_tens_ns<=r_tim_tens_ns;
    o_tim_units_ns<=r_tim_units_ns;
    o_seg_en<=r_seg_en;
    o_ew_red<=r_ew_red;
    o_ew_yellow<=r_ew_yellow;
    o_ew_green<=r_ew_green;
    o_ns_red<=r_ns_red;
    o_ns_yellow<=r_ns_yellow;
    o_ns_green<=r_ns_green;
end architecture behavior;
```