**数字电子电路**

**课程设计**

***2022.9.19~2022.9.23***

**课题： 交通灯控制器**

**学号： 21190630 21190634**

**姓名： 黄艺杰 董帅帅**

**指导老师： 黄为民**

1. **实验目的**

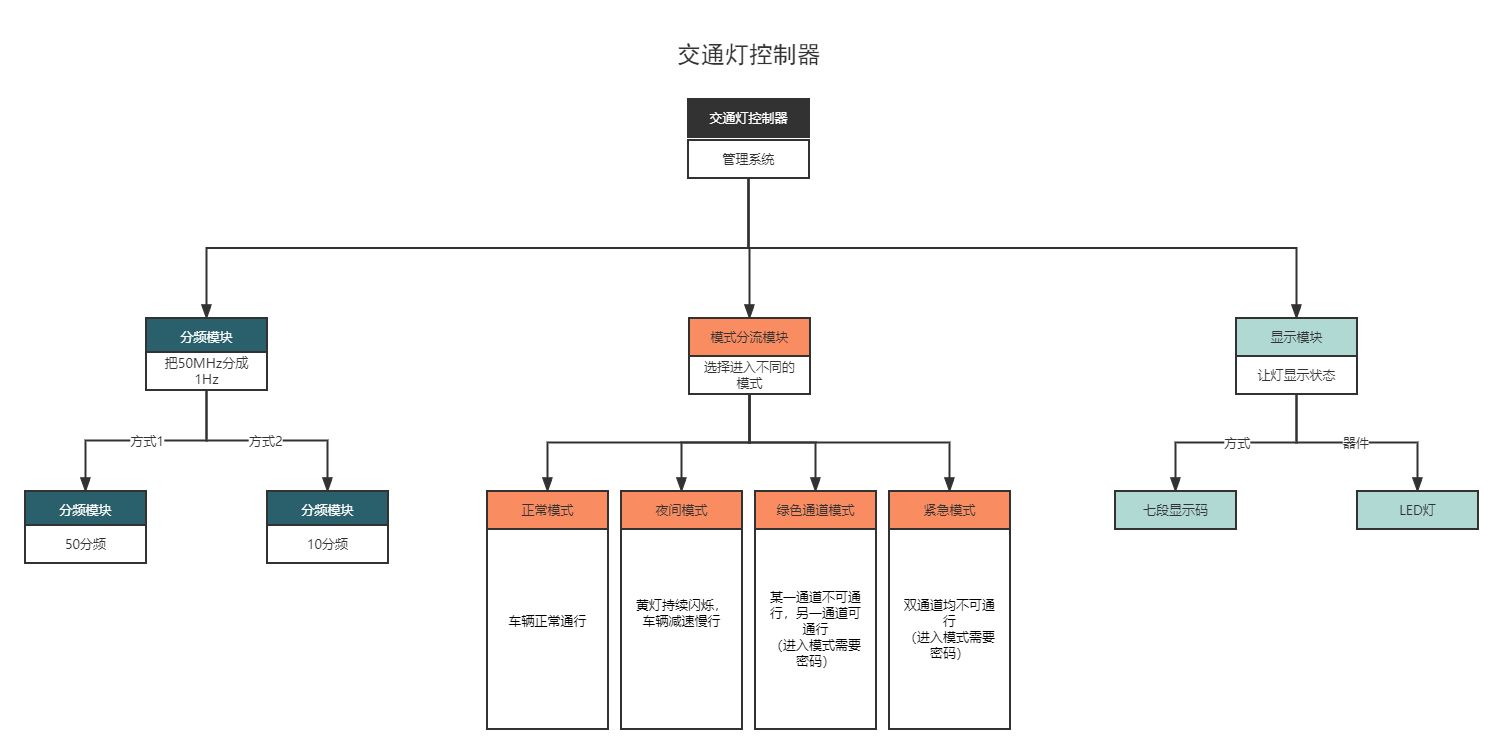
交通灯的显示有很多方式，如十字路口、丁字路口等，而对于同一个路口又有很多不同的显示要求，比如十字路口，车辆如果只要东西和南北方向通行就很简单，而如果车子可以左右转弯的通行就比较复杂，本实验针对最简单的南北和东西直行、左转的情况。

利用计数器以及状态机设计一个交通灯控制器，控制两条通道的交通灯，每条道路有四个交通灯，分别用来控制车辆的左拐通行，直行通行，黄灯以及红灯。其中两条通道的倒计时分别使用LED数码管来显示，两条道路的交通灯分别使用二极管来显示。交通灯有一个复位键，复位时所有交通灯停止工作，再次复位恢复工作。

1. **实验原理**

此实验是针对十字路口的交通灯控制。根据对交通灯的亮的规则的观察，及verilog自身的特点，主要实验设计原理如下：首先应设置输入端以及输出端，输入端包含时钟信号clk以及使能端EN,EN1,EN2，用来控制各种模式之间的转换，控制时钟由频率信号提供。输出端应包含交通灯以及倒计时的显示。尤为要注意的是，在本实验中还使用了频率转换模块将50MHz的晶振转换为1Hz的时钟信号，使能端使用开关控制。接着设计使用分配状态模块分配两条通道各种状态所需要的时间以及各个交通灯的初始状态。另外，使用BCD码来计算时间，时间到1之后转换下一个状态。周而复始，最终达到自动控制的效果。

1. **系统架构**



1. **实现功能及特色**

下面以下面的十字路口交通灯系统为例，来说明上述功能的实现：东西方向（a道）循环为绿灯45s, 黄灯5s，左拐灯15s，黄灯5s，红灯40s，黄灯5s南北方向（b道）循环为红灯65s，黄灯5s，绿灯20s，黄灯5s，左拐灯15s，黄灯5s。如下图所示

红 65s 黄5s 绿20s 黄5s 左拐15s 黄5s

绿 45s 黄 5s 左拐15秒 黄 5s 红40s 黄5s

S0 S1 S2 S3 S4 S5 S6 S7

甲道

乙道

状态表示

可以将一个周期划分成以下8种状态（甲设为a道，乙设为b道）：

S0：cnt=0-44：a道通行，b道禁止；

S1：cnt=45-49：a道停车，b道禁止；

S2：cnt=50-64；a道左拐，b道禁止；

S3：cnt=65-69；a道停车，b道禁止；

S4：cnt=70-89；a道禁止，b道通行；

S5：cnt=90-94；a道禁止，b道停车；

S6：cnt=95-109；a道禁止，b道左拐；

S7：cnt=110-114；a道禁止，b道停车；

该路口一共有四个功能，既可以实现正常的倒计时、红绿黄灯交替，也可进行不同模式的转换，下面一一阐述：

**1、正常模式**

表示该交通控制器正在正常工作，倒计时和红绿灯信号都在正常工作。

倒计时：

begin

if(numb>1)

if(!numb[3:0])

begin

numb[3:0]<=9;

numb[7:4]<=numb[7:4]-1;

end

else

numb[3:0]<=numb[3:0]-1;

if(numb==2)

tempb<=0;

end

红绿灯切换：

case (countb)

0: begin numb<=bred; LAMPB<=1; countb<=1; end

1: begin numb<=byellow; LAMPB<=2; countb<=2; end

2: begin numb<=bgreen; LAMPB<=4; countb<=3; end

3: begin numb<=byellow; LAMPB<=2; countb<=4; end

4: begin numb<=bleft; LAMPB<=8; countb<=5; end

5: begin numb<=byellow; LAMPB<=2; countb<=0; end

default: LAMPB<=1;

endcase

**2、夜间模式**

表示该交通控制器正处在夜间提醒状态，倒计时和红绿灯信号都不工作，只有黄灯不停闪烁，以提醒过往车辆注意行路安全。

LAMPA<=4'b0010;

if(LAMPA==4'b0010)

LAMPA<=4'b0000;

if(LAMPA==4'b0000)

LAMPA<=4'b0010;

counta<=0;

tempa<=0;

numa<=0;

**3、紧急模式**

表示该交通灯正处在手动控制阶段，输入密码后，通过手动控制使所有信号灯全部立即转成红灯，并且倒计时暂停，以便急行车辆同行，之后可以恢复，倒计时正常继续进行。

LAMPA<=4'b0001;

counta<=0;

tempa<=1;

numa<=0;

1. **绿色通道模式**

表示该交通灯处于手动控制阶段，输入密码后，两条车道上面有一条全面禁止，另一条可以通行，倒计时暂停，协助处理突发事件。

if(EN1&&EN&&!EN2) begin

LAMPA<=4'b0001;

counta<=0;

tempa<=0;

numa<=0;

end

else if(EN1&&!EN&&!EN2)

begin

LAMPA<=4'b0100;

counta<=0;

tempa<=1;

numa<=0;

end

**4、LED显示**

将每条通道四位BCD码倒计时转换为七段码显示在实验板的数码管上，主要方法是通过七段码的基本分配方式。其中，为了使界面整洁美观，还消去了小数点：

reg[3:0] dis;

dis<=4'b1111;

**5、绿灯闪烁**

当绿灯通行时，从倒计时5秒开始绿灯闪烁，用来提醒行人。

以b道为例：

if(numb==5 && LAMPB==4)

LAMPB<=0;

if(numb==4 && LAMPB==0)

LAMPB<=4;

if(numb==3 && LAMPB==4)

LAMPB<=0;

if(numb==2 && LAMPB==0)

LAMPB<=4;

if(numb==1 && LAMPB==4)

LAMPB<=0;

**五、分析与设计思路**

本次设计是针对十字路口，进行南北和东西直行情况下交通灯控制。设定东西方向为主干道方向，根据交通灯的亮的规则，进入正常工作状态后，当主干道上绿灯亮时，支干道上红灯亮，之后，主干道和支干道上的黄灯都亮启，持续一段时间后，主干道上红灯亮启，支干道上绿灯亮启，之后主干道和支干道上的黄灯都亮启，一个循环完成。循环往复的直行这个过程。为让交通灯的工作能力更加突出，于是增加了多种工作模式，让交通灯的设计更加贴近实际，方便使用。

此实验中需要利用计数器以及状态机设计一个交通灯控制器，控制两条通道的交通灯，其中两条通道的倒计时分别使用LED数码管来显示，两条道路的交通灯分别使用二极管来显示。交通灯有一个复位键，复位时所有交通灯停止工作，再次复位恢复工作。

根据以上分析，我们的设计中大致分三个模块：

1. 频率转换模块，系统时钟计时模块需要1HZ的脉冲。此模块用来将实验中提供的50MHz的晶振转换成为1Hz，以提供每次一秒的时钟信号用于倒计时。

2）显示模块，可以根据控制信号，驱动交通信号灯，以及倒计时数码管的显示。使用用七段码将八位BCD码转换成两位十进制数显示在LED上。

3）顶层模块，包含状态分配，初始化倒计时以及模式转换三个功能。其中，状态分配使用八位BCD码初始化每个状态的时间，初始化倒计时分别对两个通道BCD码的时间进行倒计时，状态转换是用于对时间倒计时结束后分配交通灯的下一个模式。

**六、设计代码**

**1、频率转换模块**

模块简介：这个模块的基本功能就是将DE2-70实验板上的50MHz的晶振转换为1Hz的时钟信号，主要方法是通过递归来降低频率，为了使时钟信号更加准确，本次实验使用实验板内部自带的晶振信号。

module clk\_1hz(CLK, RST, clk\_1Hz);

input CLK, RST;

output clk\_1Hz;

wire clk\_1MHz, clk\_100KHz, clk\_10KHz, clk\_1KHz, clk\_100Hz, clk\_10Hz;

divide\_by\_50 d6 (clk\_1MHz, CLK, RST);

divide\_by\_10 d5 (clk\_100KHz, clk\_1MHz, RST);

divide\_by\_10 d4 (clk\_10KHz, clk\_100KHz, RST);

divide\_by\_10 d3 (clk\_1KHz, clk\_10KHz, RST);

divide\_by\_10 d2 (clk\_100Hz, clk\_1KHz, RST);

divide\_by\_10 d1 (clk\_10Hz, clk\_100Hz, RST);

divide\_by\_10 d0 (clk\_1Hz, clk\_10Hz, RST);

endmodule

module divide\_by\_10 (Q, CLK, RST);

input CLK,RST;

output Q;

reg Q;

reg [2:0] count;

always @(posedge CLK or negedge RST)

begin

if (~RST)

begin

Q <= 1'b0;

count <= 3'b000;

end

else if (count < 4)

begin

count <= count + 1'b1;

end

else

begin

count <= 3'b000;

Q <= ~Q;

end

end

endmodule

module divide\_by\_50 (Q, CLK, RST);

input CLK, RST;

output Q;

reg Q;

reg [4:0] count;

always @(posedge CLK or negedge RST)

begin

if (~RST)

begin

Q <= 1'b0;

count <= 5'b00000;

end

else if (count < 24)

begin

count <= count + 1'b1;

end

else

begin

count <= 5'b00000;

Q <= ~Q;

end

end

endmodule

**2、LED显示模块**

模块简介：这个模块的基本功能就是将每条通道四位BCD码倒计时转换为七段码显示在实验板的数码管上，主要方法是通过七段码的基本分配方式。

module hex\_7seg (digit, seg);

input [3:0] digit;

output [6:0] seg;

reg [6:0] seg;

always @(digit)

case(digit)

4'h0: seg = ~7'h3F;

4'h1: seg = ~7'h06;

4'h2: seg = ~7'h5B;

4'h3: seg = ~7'h4F;

4'h4: seg = ~7'h66;

4'h5: seg = ~7'h6D;

4'h6: seg = ~7'h7D;

4'h7: seg = ~7'h07;

4'h8: seg = ~7'h7F;

4'h9: seg = ~7'h6F;

4'ha: seg = ~7'h77;

4'hb: seg = ~7'h7C;

4'hc: seg = ~7'h39;

4'hd: seg = ~7'h5E;

4'he: seg = ~7'h79;

4'hf: seg = ~7'h71;

endcase

endmodule

**3、顶层模块**

模块简介：这个模块是整个课程设计的核心，定义了输入端以及输出端，额外考虑到美观因素，将LED屏幕小数点消除了。

module Traffic\_Control(CLK,EN,LAMPA,LAMPB,ACOUNT,BCOUNT,dis,EN1,EN2,M3,M4,M5);

output[13:0] ACOUNT,BCOUNT;

output[3:0] LAMPA,LAMPB;

output [3:0]dis;

input CLK,EN,EN1,EN2,M3,M4,M5;

reg[7:0] numa,numb; //八位BCD码倒计时

reg tempa,tempb;

reg[2:0] counta,countb; //控制亮灯的顺序

reg[7:0] ared,ayellow,agreen,aleft,bred,byellow,bgreen,bleft;

//各状态的时间

reg[3:0] LAMPA,LAMPB; //左拐通行，绿灯，黄灯，红灯

reg[3:0] dis; //用于去除LED小数点

assign RST=1;

//clk\_1hz clk (CLK,RST,clk\_1Hz);

hex\_7seg h7 (numa[7:4],ACOUNT[13:7]);

hex\_7seg h6 (numa[3:0],ACOUNT[6:0]);

hex\_7seg h5 (numb[7:4],BCOUNT[13:7]);

hex\_7seg h4 (numb[3:0],BCOUNT[6:0]);

always @(EN or EN1 or EN2 )

begin

if(!EN&&!EN1&&!EN2)

begin //设置各种灯的计数器的预置数

ared <=8'b01000101; //45 秒

ayellow <=8'b00000101; //5 秒

agreen <=8'b01000101; //45 秒

aleft <=8'b00010101; //15 秒

bred <=8'b01110000; //70 秒

byellow <=8'b00000101; //5 秒

bleft <=8'b00010101; //15 秒

bgreen <=8'b00100000; //20 秒

dis<=4'b1111;

end

end

//always @(posedge clk\_1Hz) //该进程控制 A 方向的四种灯

always @(posedge CLK)

begin

if(EN&&!EN1&&!EN2)

begin

if(!tempa)

begin

tempa<=1;

case(counta) //控制亮灯的顺序

0: begin numa<=agreen; LAMPA<=4; counta<=1; end

1: begin numa<=ayellow; LAMPA<=2; counta<=2; end

2: begin numa<=aleft; LAMPA<=8; counta<=3; end

3: begin numa<=ayellow; LAMPA<=2; counta<=4; end

4: begin numa<=ared; LAMPA<=1; counta<=0; end

default: LAMPA<=1;

endcase

end

else

begin //倒计时

if(numa==5 && LAMPA==4)

LAMPA<=0;

if(numa==4 && LAMPA==0)

LAMPA<=4;

if(numa==3 && LAMPA==4)

LAMPA<=0;

if(numa==2 && LAMPA==0)

LAMPA<=4;

if(numa==1 && LAMPA==4)

LAMPA<=0;

if(numa>1)

if(numa[3:0]==0)

begin

numa[3:0]<=9;

numa[7:4]<=numa[7:4]-1;

end

else

numa[3:0]<=numa[3:0]-1;

if(numa==2)

tempa<=0;

end

end

else

if(M3&&M4&&M5)

begin

if(EN1&&EN&&!EN2) //011

begin

LAMPA<=4'b0001;

counta<=0;

tempa<=0;

numa<=0;

end

else if(EN1&&!EN&&!EN2) //010

begin

LAMPA<=4'b0100;

counta<=0;

tempa<=1;

numa<=0;

end

else if(EN2&&!EN1&&EN) //101

begin

LAMPA<=4'b0001;

counta<=0;

tempa<=1;

numa<=0;

end

end

else

begin

LAMPA<=4'b0010;

if(LAMPA==4'b0010)

LAMPA<=4'b0000;

if(LAMPA==4'b0000)

LAMPA<=4'b0010;

counta<=0;

tempa<=0;

numa<=0;

end

end

//always @(posedge clk\_1Hz) //该进程控制 B 方向的四种灯

always @(posedge CLK)

begin

if (EN&&!EN1&&!EN2)

begin

if(!tempb)

begin

tempb<=1;

case (countb) //控制亮灯的顺序

0: begin numb<=bred; LAMPB<=1; countb<=1; end

1: begin numb<=bgreen; LAMPB<=4; countb<=2; end

2: begin numb<=byellow; LAMPB<=2; countb<=3; end

3: begin numb<=bleft; LAMPB<=8; countb<=4; end

4: begin numb<=byellow; LAMPB<=2; countb<=0; end

default: LAMPB<=1;

endcase

end

else

begin //倒计时

if(numb==5 && LAMPB==4)

LAMPB<=0;

if(numb==4 && LAMPB==0)

LAMPB<=4;

if(numb==3 && LAMPB==4)

LAMPB<=0;

if(numb==2 && LAMPB==0)

LAMPB<=4;

if(numb==1 && LAMPB==4)

LAMPB<=0;

if(numb>1)

if(!numb[3:0])

begin

numb[3:0]<=9;

numb[7:4]<=numb[7:4]-1;

end

else

numb[3:0]<=numb[3:0]-1;

if(numb==2)

tempb<=0;

end

end

else

if(M3&&M4&&M5)

begin

if(EN1&&EN&&!EN2) //011

begin

LAMPB<=4'b0100;

countb<=0;

tempb<=0;

numb<=0;

end

else if(EN1&&!EN&&!EN2) //010

begin

LAMPB<=4'b0001;

countb<=0;

tempb<=1;

numb<=0;

end

else if(EN2&&!EN1&&EN)

begin

LAMPB<=4'b0001;

countb<=0;

tempb<=1;

numb<=0;

end

end

else

begin

LAMPB<=4'b0010;

if(LAMPB==4'b0010)

LAMPB<=4'b0000;

if(LAMPB==4'b0000)

LAMPB<=4'b0010;

tempb<=0;

countb<=0;

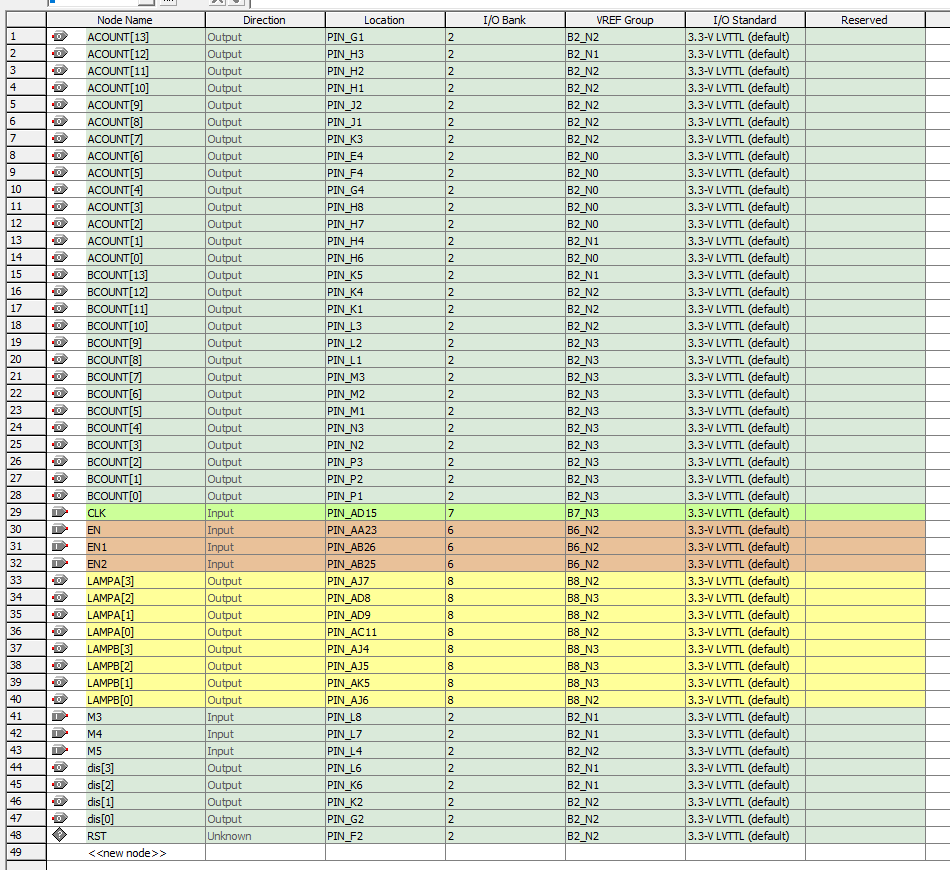
numb<=0;

end

end

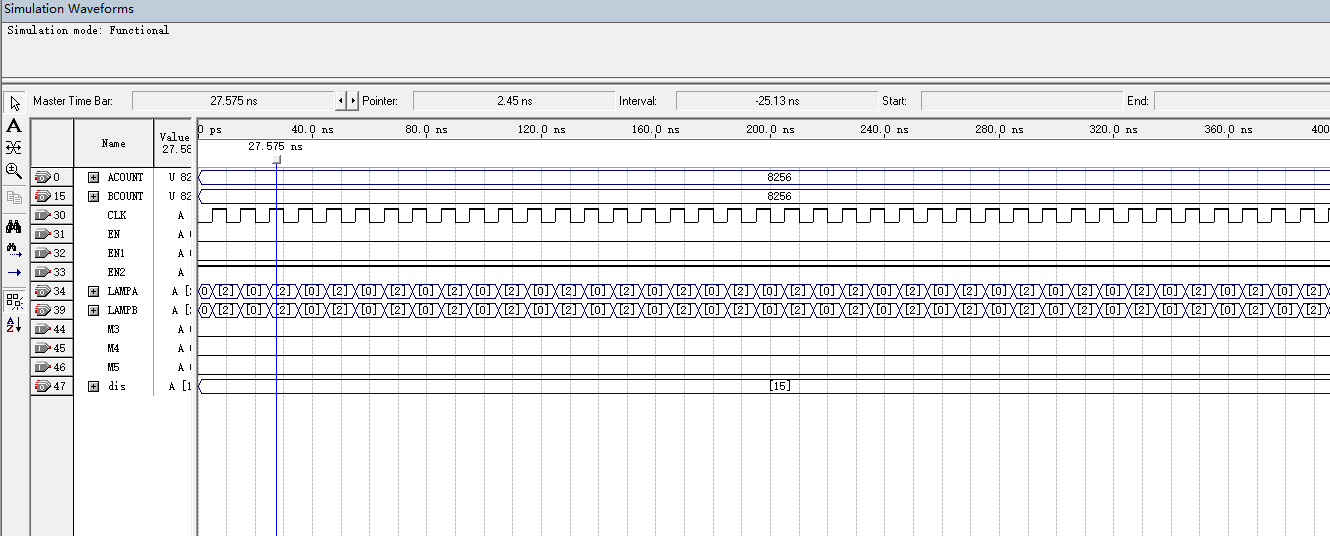
endmodule

**七、管脚分配**

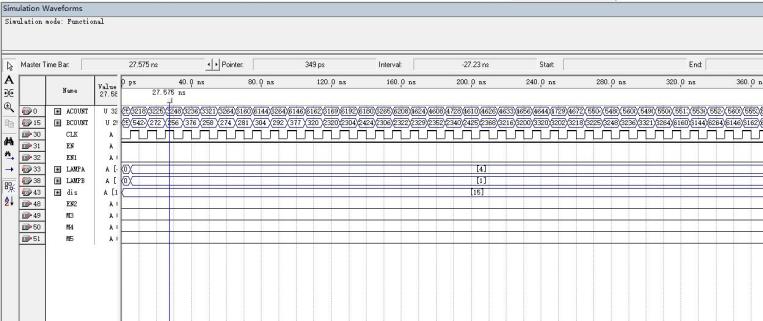


**八、功能仿真结果**

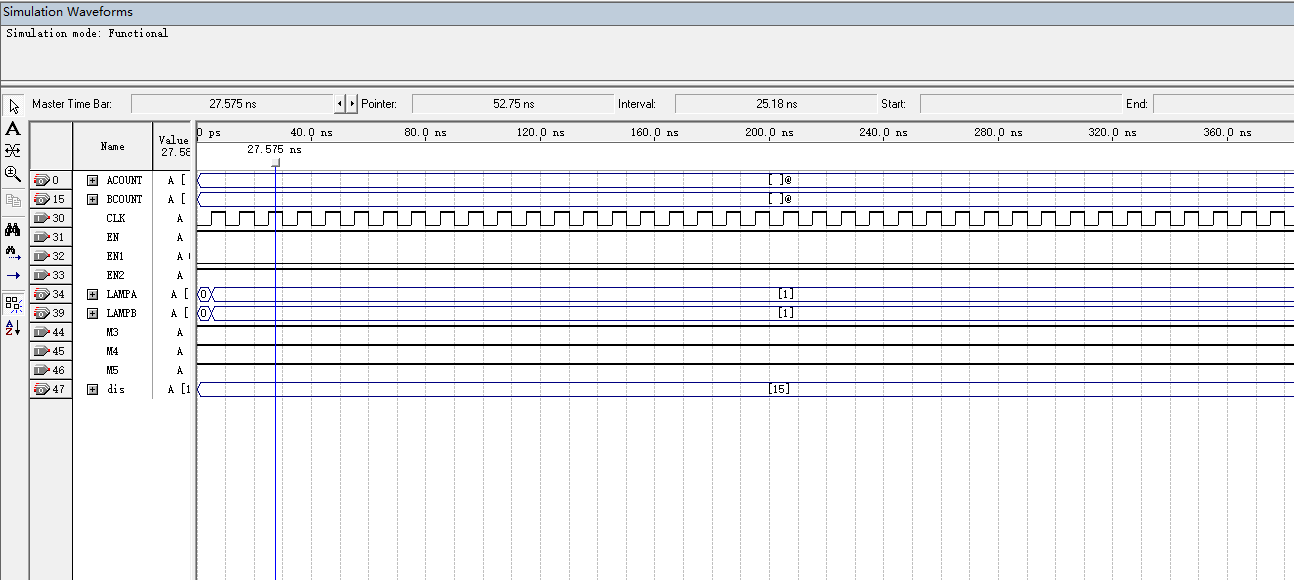
本设计采用QuartusII软件进行编译并进行时序仿真。其中，时钟周期此用1us。整个工作流程仿真结果如下：



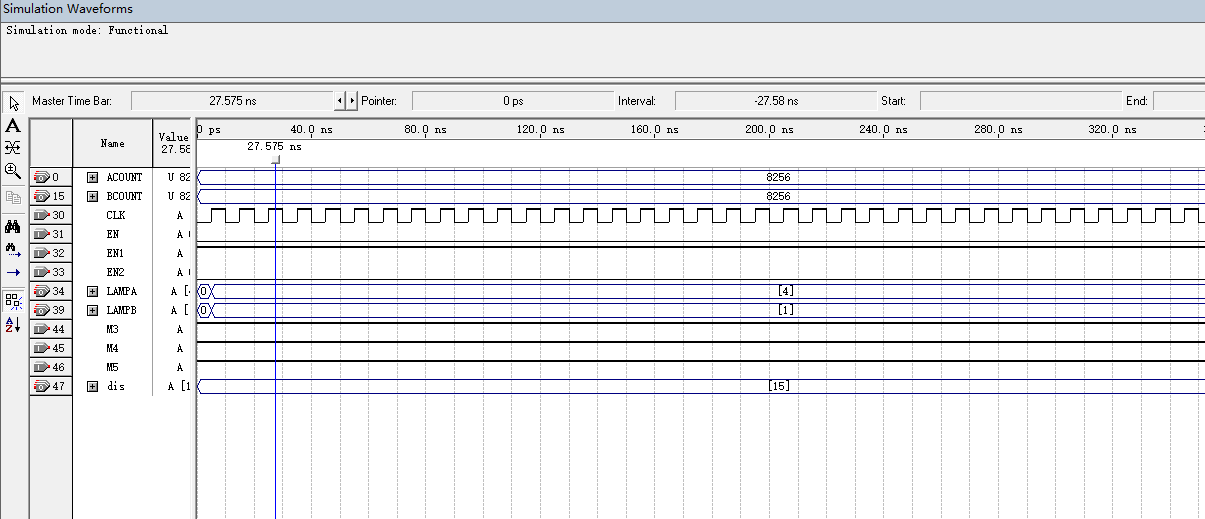
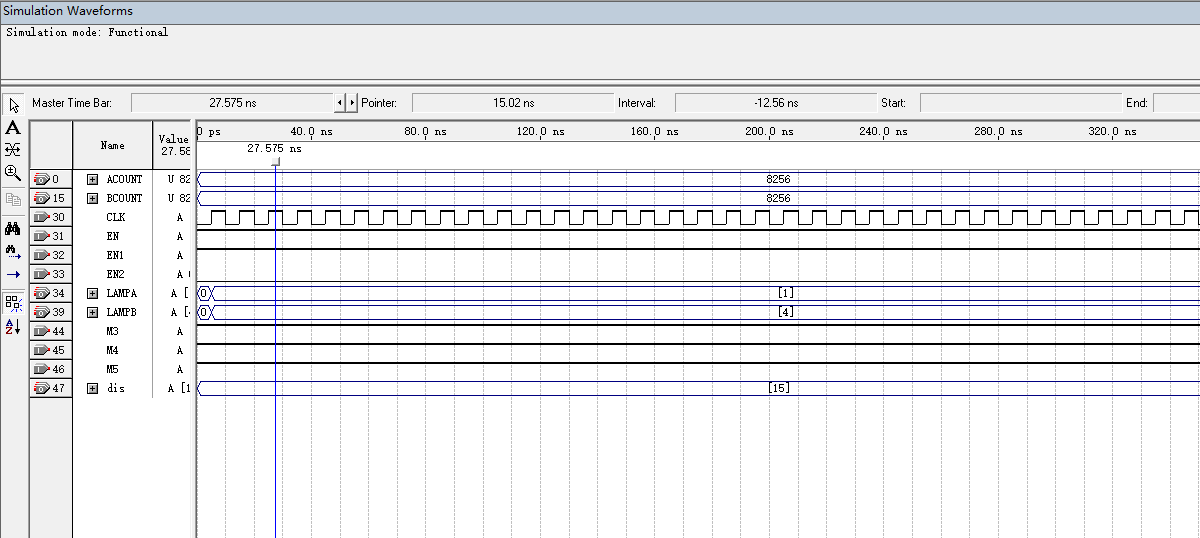
状态A：交通灯夜间工作状态



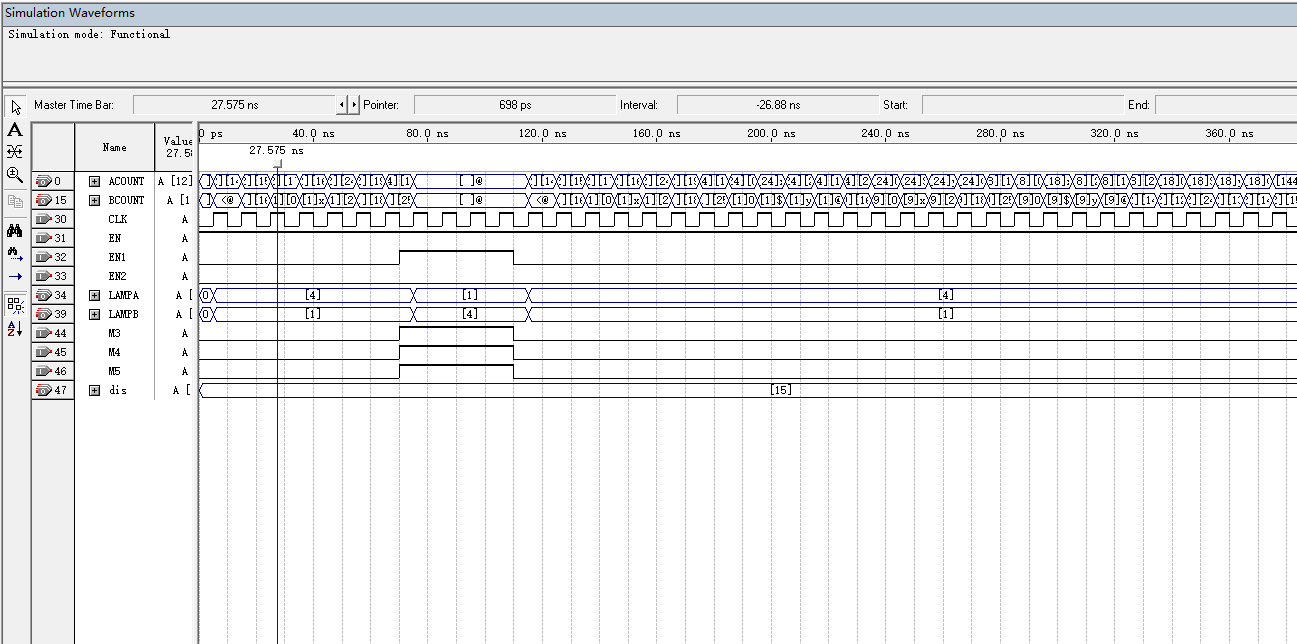
状态B：交通灯正常工作状态



状态C：交通灯紧急工作状态



状态D：交通灯绿色通道工作状态

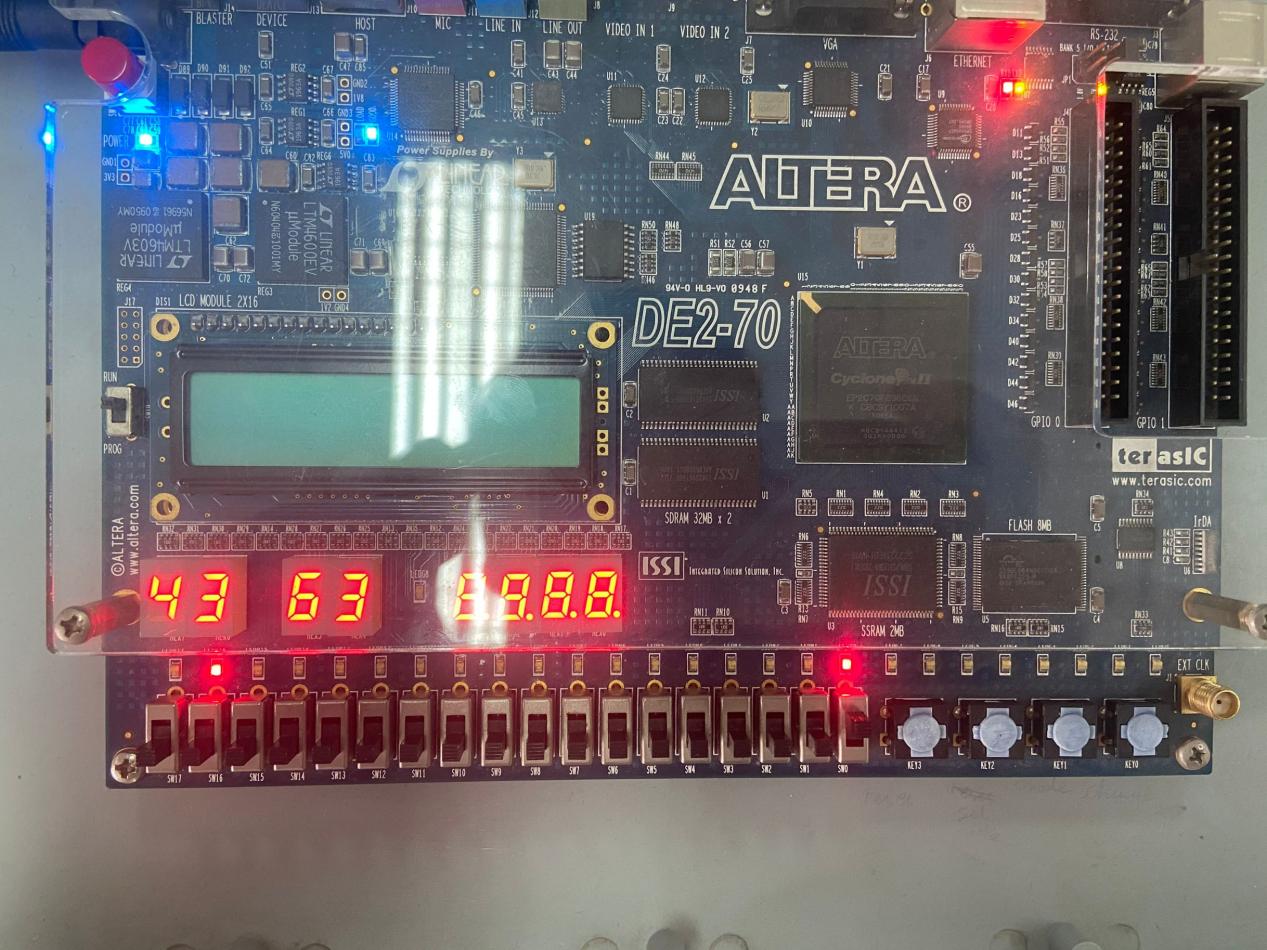


状态E：交通灯正常工作状态到绿色通道工作状态再返回正常

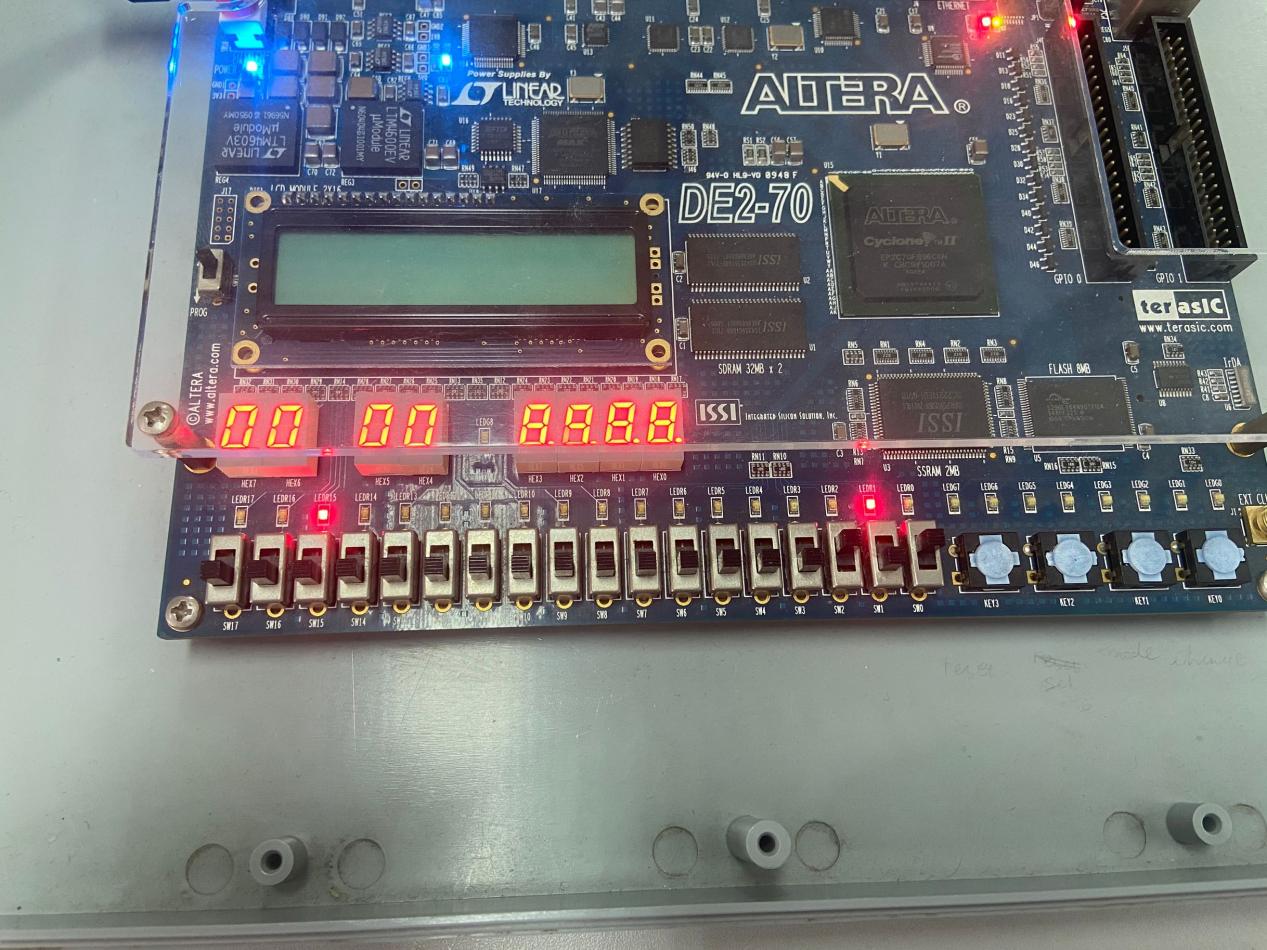
交通灯控制器的仿真图

**九、实验结果展示**

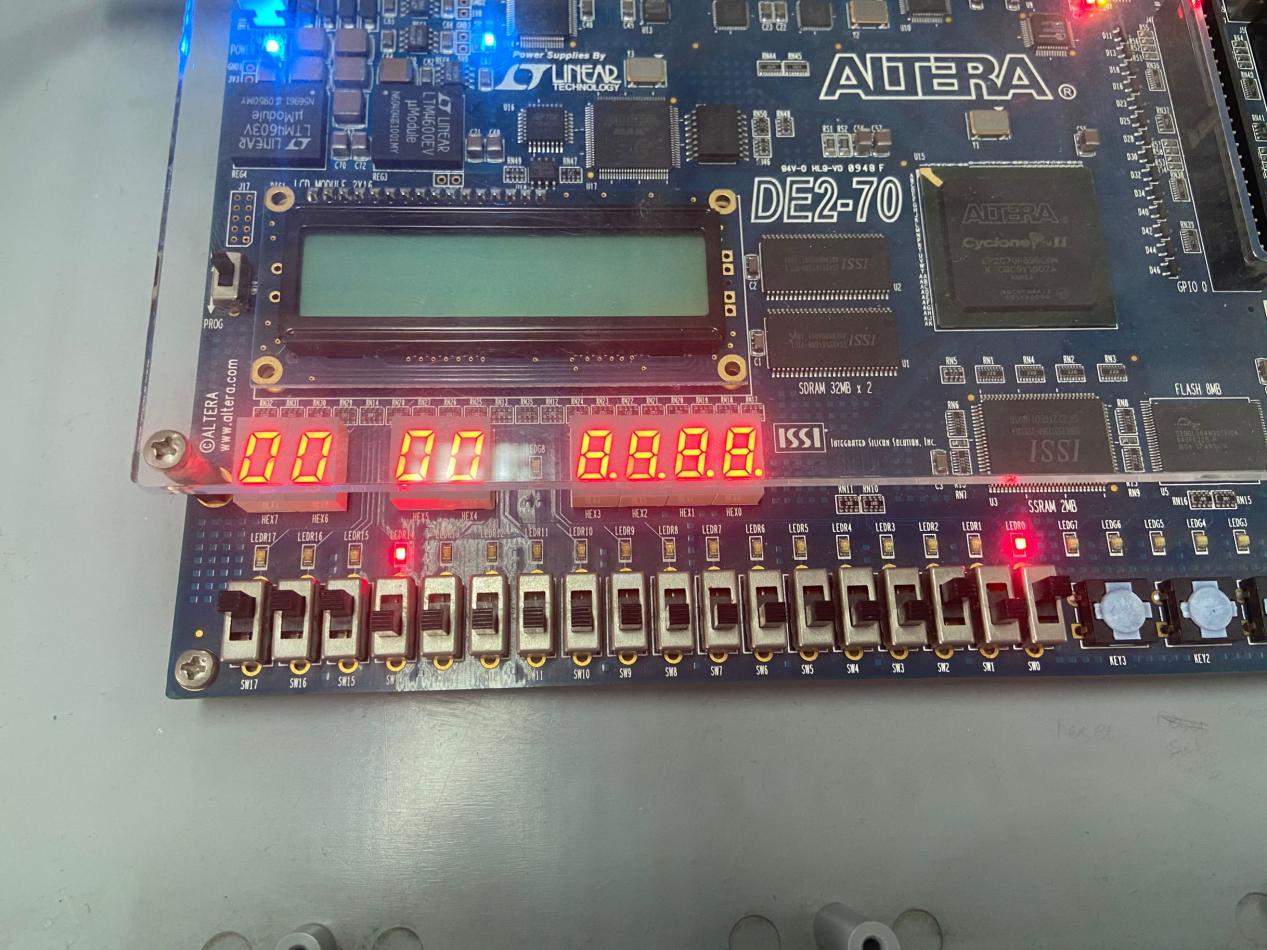
1、正常模式



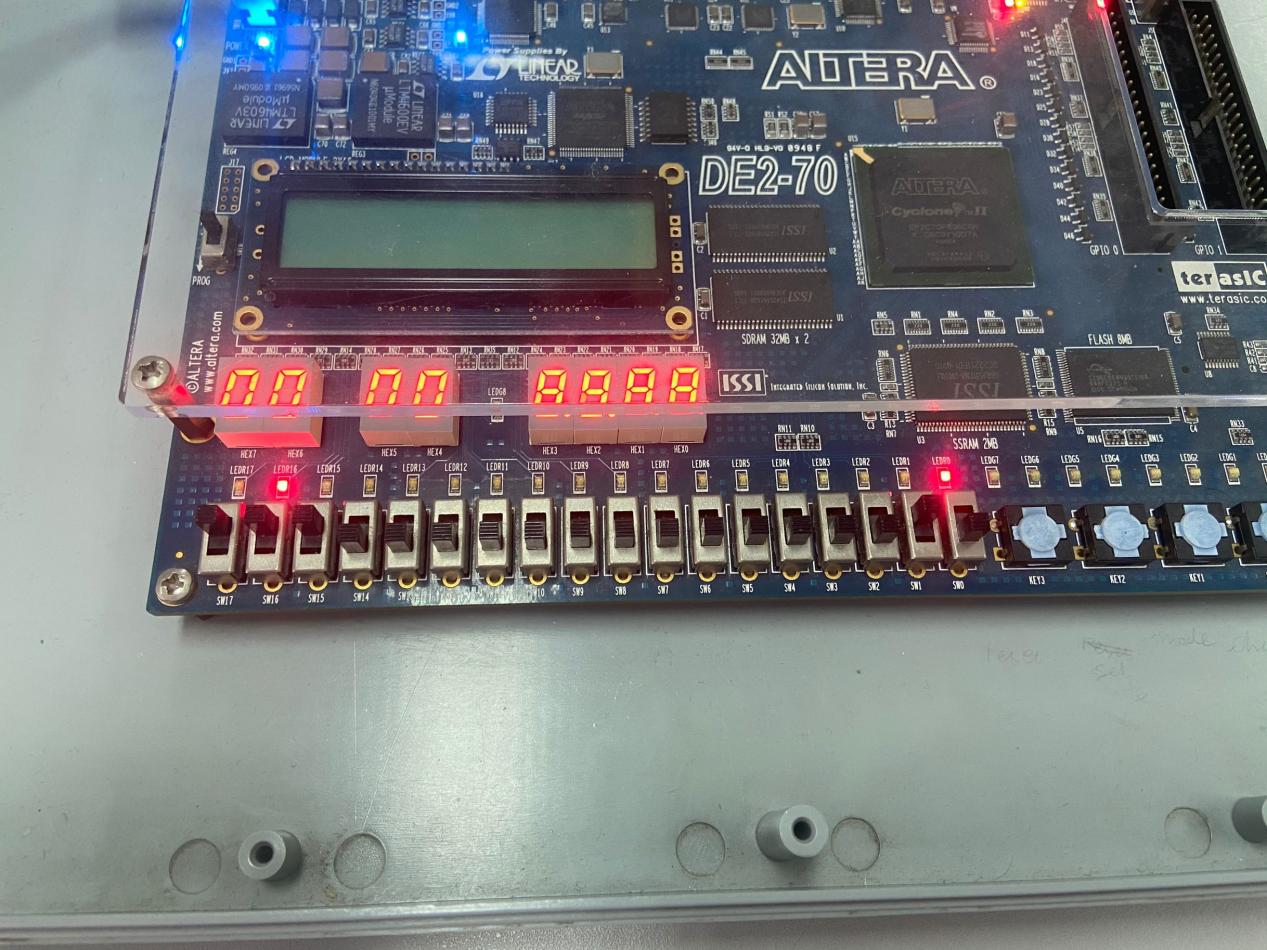
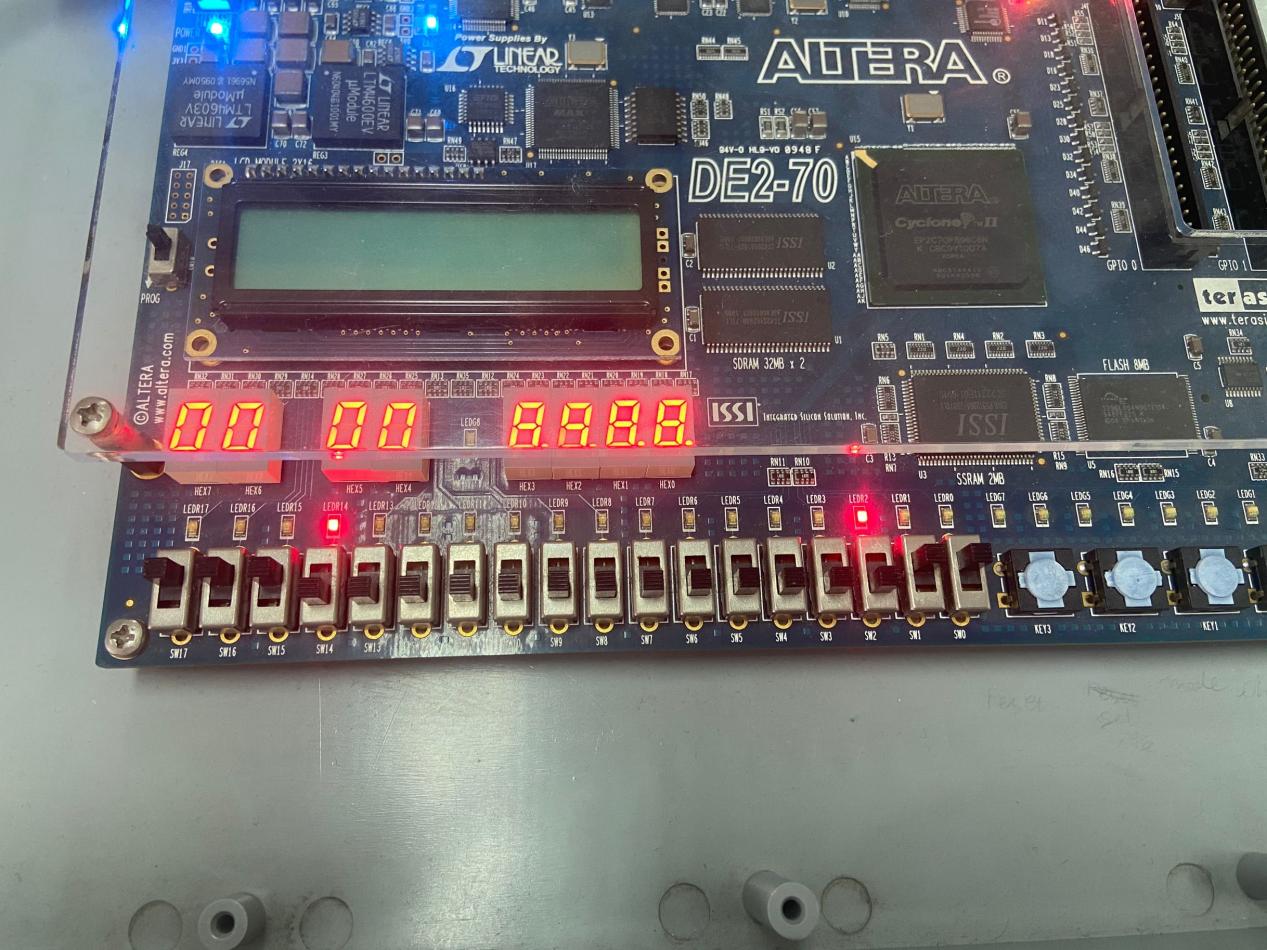
2、夜间模式



3、紧急模式



1. 绿色通道模式



**十、出现的问题及解决方案**

1. 开始时无法分频出一个1Hz的时钟信号，后来老师共享了代码，我们添加了一个分频模块，将实验板内部的50MHz的晶振信号转换成了1Hz的信号，也就是一秒一次的时钟信号。
2. 实验后发现LED界面上的数字之间总有小数点消不去，后来上网查阅了资料，改进了代码，使得界面整洁美观。
3. 仿真实验的时候不显示结果，总是不显示结果，后来经过老师指点，才发现是时钟信号的问题，写到板子和在电脑上仿真用的是两种时钟。

**十一、总结与感悟**

通过这次课程设计，我再一次体验到了细心对一个编程者的重要性，以及程序的规范性对于程序的重要性，这些平时我们忽略的问题，其实有时候关乎着我们编程的成功率；再者，在verilog语言中，我们必须注意其与C语言的异同，比如格式和变量定义，还有模块的调用，和时钟信号的应用。还有，verilog HDL设计语言是一门很好的硬件描述语言，可以直白的描述实际的电路，实际的系统模型，易懂而且易于实现，颠覆了我对硬件描述语言的认知，感受到了这种语言的强大能力。

课程设计培养了学生综合运用所学知识，发现，提出，分析和解决实际问题，锻炼实践能力的重要环节，是对学生实际工作能力的具体训练和考察过程。然而从理论到实践，我在做课程设计的几天里，使我对数字逻辑电路这门课有了更进一步的认识和了解，要想学好它要重在实践，要通过不断的实际操作才能更好地学习它，不仅可以巩固了以前所学过的知识，而且学到了很多在书本上所没有学到过的知识。

这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，好多次编程感觉没问题，但是一到板子上就出事，有错误，只有把所学的理论知识与实践相结合起来，从而提高自己的实际动手能力和独立思考的能力。在设计的过程中遇到问题，可以说得是困难重重，遇到了各种各样的问题，同时在设计的过程中发现了自己的不足之处，对以前所学过的知解得不够深刻，掌握得不够牢固，在编写程序的过程中我几乎是重新复习课本上的知识，再加上询问老师、到网上查询资料以及和同学的不断交流才慢慢的完善这次课程设计，也让我有了些许的成就感。

**十二、参考资料**

[1] 康华光. 电子技术基础(数字部分) [M ]. 第五版

[2]Verilog HDL程序设计教程 （第一版） 王金明主编

[3] 夏宇闻. 复杂数字电路与系统的V erilog HDL设计技术 [M ].北京: 北京航空航天大学出版社, 1998

[4] 张明. V erilog HDL 实用教程[M ]. 成都: 电子科技大学出版社, 1999

[5] FPGA设计及应用（第二版）西安电子科技大学出版社