

东南大学自动化学院

# 《数字系统课程设计》

## 设计报告

项目名称： 交通灯控制系统

姓 名： 刘页恺 学 号： 08014233

专 业： 自动化 实 验 室： 416

组 别：           \           同组人员：           \          

设计时间： **2016 年 8 月 26 日** —— **2016 年 9 月 2 日**

评定成绩：                                  审阅教师： 孙培勇

# 目 录

一. 设计方案及论证.....	3
1.1 设计使用环境.....	3
1.2 设计任务分析.....	3
1.3 测量控制原理.....	3
1.4 顶层设计方案框图及说明.....	3
二. 模块设计.....	4
2.1 模块功能及端口说明.....	4
2.2 主要功能的设计方法.....	4
2.3 Verilog 设计程序及说明.....	5
2.4 仿真图及说明.....	13
三. 总体设计与仿真.....	15
3.1 顶层设计图及说明.....	15
3.2 仿真图及说明.....	16
3.3 实验结果.....	17
四. 总结.....	18
4.1 实验结果分析.....	18
4.2 问题解决方法.....	18
4.3 心得体会.....	18
参考书目.....	19

# 一、设计方案及论证

## 1.1 设计使用环境

基于 verilog 硬件描述语言利用 QuartusII 集成开发工具设计的交通灯控制系统可以用于主干道与乡村公路交叉的十字路口，以帮助车辆安全通过和避免交通拥堵，系统配备有红、绿、黄三色信号灯以及数码管倒计时显示器（乡村公路安装了传感器，检测车辆是否到来的信号可用乒乓开关代替）。

## 1.2 设计任务分析

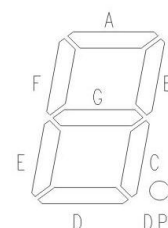
- (1) 当乡村公路无车时，始终保持乡村公路红灯亮，主干道绿灯亮。
- (2) 。主干道最短通车时间为 25s 。
- (3) 当乡村公路和主干道都有车时，按主干道通车 25s，乡村公路通车 16s 交替进行。
- (4) 不论主干道情况如何，乡村公路通车最长时间为 16s。
- (5) 在每次由绿灯亮变成红灯亮的转换过程中间，要亮 5s 时间的黄灯作为过渡。
- (6) 用开关代替传感器作为检测车辆是否到来的信号。用红、绿、黄三种颜色的发光二极管作交通灯。
- (7) 要求显示时间，倒计时。

## 1.3 测量控制原理

(1) 时钟分频：实验板提供的时钟脉冲频率为 2MHz，而交通灯倒计时以 1 秒为单位时间，所以状态机和数码管译码模块输入的时钟信号要先经过分频模块进行分频处理。

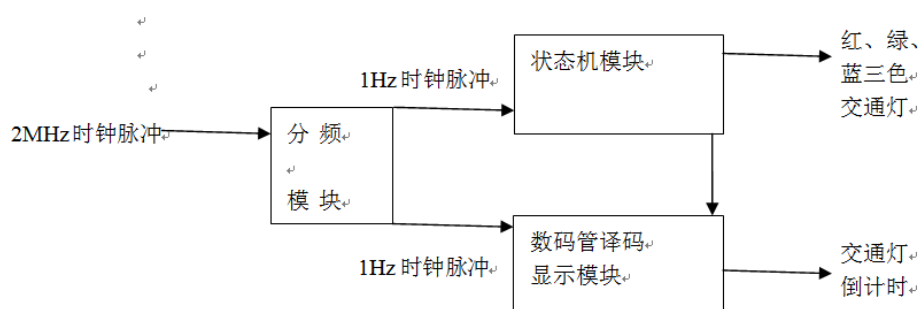
(2) 状态机控制：有限状态机（Finite State Machine, FSM）适于设计数字系统的控制模块，采用一位热码编码（ONE-HOT Encoding）可以有效提高 FPGA 器件电路的速度和可靠性，也有利于提高器件资源的利用率，状态转换的依据是计数时间的改变。

(3) 数码管显示被控量：共阴极八段数码管（如右图）用于显示交通灯倒计时，某段赋上低电平时，该段点亮，赋上高电平，该段熄灭。参照数码管八段译码对照表可以设计出译码显示模块。



## 1.4 顶层设计方案框图及说明

(1) 顶层设计方案框图：



(2) 顶层设计方案框图说明：顶层设计方案中包含三个模块设计，分别是分频模块，

状态机模块和数码管译码显示模块。下面说明三个模块的联系：其中分频模块的输入端口接 2MHz 时钟脉冲（由实验板产生），输出端口输出 1Hz 时钟脉冲；状态机模块的时钟输入端口接 1Hz 时钟脉冲，状态输出端口接 6 盏交通灯（用实验箱上 6 个 LED 替代），还输出交通灯倒计时信息到数码管译码显示模块；数码管译码显示模块对状态机传来的倒计时信息进行译码使之可以通过八段数码管显示出来。

## 二、 模块设计

### 2.1 模块功能及端口说明

（1）模块功能：分频模块的功能是将实验板提供的频率为 2MHz 的时钟脉冲转为交通灯控制系统需要的频率为 1Hz 的时钟脉冲；状态机模块的功能是接受乡村道路车辆传感器（用乒乓开关代替）传递的状态信息，并根据该状态信息按设计的交通灯控制系统状态图在 4 种状态中实现自动无误地转换；数码管译码显示模块的功能是根据数码管八段译码对照表将状态模块传递的交通灯倒计时信息准确同步地用数码管显示出来。

（2）模块端口说明：分频模块的端口有 F1 和 F2，其中 F1 是 2MHz 时钟脉冲的输入端口，F2 是 1Hz 时钟脉冲的输出端口；状态机模块的端口有 C, RET, CLK, MG, MY, MR, CG, CY, CR, T1, T2，其中 C、RET、CLK 是输入端口，MG, MY, MR, CG, CY, CR, T1, T2 是输出端口，C 端口接乡村道路车辆传感器（用乒乓开关代替），RET 端口接初始化开关，CLK 端口接 1MHz 的时钟脉冲，MG, MY, MR, CG, CY, CR 端口分别输出 6 盏交通灯的亮灭信息，T1、T2 端口分别输出主干道和乡村道路交通灯的倒计时信息；数码管译码显示模块的端口有 RET, CLK, MainCountdown, CountryCountdown, ML, MR, CL, CR, C，其中 RET, CLK, MainCountdown, CountryCountdown, C 是输入端口，ML, MR, CL, CR 是输出端口，RET 端口接初始化开关，CLK 端口接 1MHz 的时钟脉冲，MainCountdown, CountryCountdown 端口分别接主干道和乡村道路交通灯的倒计时信息 T1、T2，ML, MR, CL, CR 端口分别输出显示在 4 个数码管上数字 0-9。

### 2.2 主要功能的设计方法

（1）分频功能的设计：设定合适的分频系数和初值为 0 的变量，待分频的输入时钟脉冲信号上升沿触发该变量值加 1，变量的值小于分频系数时，输出信号值不变（固定为 0 或 1），当变量的值等于分频系数时，输出信号取反，变量值归零。如此循环即可输出相应频率的时钟脉冲信号，从而实现分频功能。

（2）状态机的设计：交通灯控制系统包含四个状态 S1、S2、S3、S4

S1：主干道绿灯 (MG)，乡村道路红灯 (CR)

S2：主干道黄灯 (MY)，乡村道路红灯 (CR)

S3：主干道红灯 (MR)，乡村道路绿灯 (CG)

S4：主干道红灯 (MR)，乡村道路黄灯 (CY)

首先系统处于 S1 状态，主干道绿灯 (MG)，乡村道路红灯 (CR)，如果 C 一直为 0（乡村道路无车），则一直保持 S1 状态；当 C 信号第一次被置高（表示乡村公路有车），系统保持 S1 状态 25 秒（当乡村公路有车时，要求主干道通车时间已经超过它的最短通车时间 25 秒时，才禁止主干道通行，让乡村公路通行），接着系统进入 S2 状态，主干道黄灯 (MY)，乡村道路

红灯(CR)， 5 秒的黄灯倒计时结束后，系统进入 S3 状态，主干道红灯(MR)，乡村道路绿灯(CG)，乡村道路绿灯 16 秒倒计时结束后系统进入 S4 状态，主干道红灯(MR)，乡村道路黄灯(CY)，开始 5 秒的黄灯倒计时，如果在乡村道路绿灯倒计时的 S3 过程中 C 信号突然被置高，则直接进入 S4 状态，S4 状态黄灯 5 秒倒计时结束后进入 S1 状态；如果 C 信号保持为 1，则系统继续 S1→S2→S3→S4→S1 的循环，如果 C 信号为 0，则一直保持在 S1 状态；S1 状态下 C 信号再次（不是第一次）被置高则系统立刻进入 S2 状态，5 秒的黄灯倒计时后进入 S3 状态让乡村道路通行 16 秒，之后再依次进入 S4、S1 状态，如此循环。

(3) 数码管显示功能的设计：将交通灯倒计时的数值拆分成十位和个位两位数分别在两个数码管上显示，参照数码管八段译码对照表可以设计出译码显示模块。

## 2.3 Verilog 设计程序及说明

(1) 分频模块设计程序

```
module frequency(F1, F2);    // 模块名为 frequency
input F1;                  //F1,F2 分别为输入、输出变量
output F2;
reg F2;
reg [31:0] division;        //division 变量帮助分频
parameter K = 1000000;      //常量 K 是分频系数

initial
begin
    division <= 0;          //division 变量初始化为 1
end

always @(posedge F1)        //F1 时钟脉冲的上升沿触发过程块执行
    if(division == (K - 1) )
    begin
        division <= 0;
        F2 <= ~F2; //division 变量累加到分频系数的值时归零，F2 信号取反
    end
    else
        division <= (division + 1);
    end
endmodule
```

(2) 状态机模块设计程序

```
module FSM(C, RET, CLK, MG, MY, MR, CG, CY, CR, T1, T2);

input C, RET, CLK;
output MG, MY, MR, CG, CY, CR;
```

```

output [4:0]T1, T2;

reg MG, MY, MR, CG, CY, CR, F;
  reg [10:0] G;
reg [4:0]T1, T2;
reg [3:0] state;
/* 状态机模块的端口有 C, RET, CLK, MG, MY, MR, CG, CY, CR, T1, T2, 其中 C、
RET、CLK 是输入端口, MG, MY, MR, CG, CY, CR, T1, T2 是输出端口, C 端口接乡村道路
车辆传感器(用乒乓开关代替), RET 端口接初始化开关, CLK 端口接 1MHz 的时钟脉冲, MG,
MY, MR, CG, CY, CR 端口分别输出 6 盏交通灯的亮灭信息, T1、T2 端口分别输出主干道和
乡村道路交通灯的倒计时信息, 变量 G 的设定是为了实现当 C 信号第一次被置高(表示乡村
公路有车), 系统保持 S1 状态 25 秒(当乡村公路有车时, 要求主干道通车时间已经超过它
的最短通车时间 25 秒时, 才禁止主干道通行, 让乡村公路通行)这一条件, 变量 F 用于记录
C 置高之前的状态 */

```

```

parameter s1 = 4'b0001, // 状态 1:MG, CR
          s2 = 4'b0010, //状态 2:MY, CR
          s3 = 4'b0100, //状态 3:MR, CG
          s4 = 4'b1000; //状态 4:MR, CY

initial
begin
G=512; //G 初始化为 512(只要初始化为一个小于 2^10 的正数即可)
end
always @(posedge CLK) //时钟脉冲的上升沿触发过程块执行
begin

if (!RET)
begin
state = s1;
T1 = 25;
T2 = 30;
MG = 1; CR = 1; MY = 0; MR = 0; CG = 0; CY = 0;
end
else //初始化信号 RET 为 0 时, 系统保持在 S1 状态

case(state) //借助 case 语句实现状态编码
s1:

if(C == 0) //S1 状态下, C 为 0 时系统保持 S1 状态
begin
T1 = 25;
T2 = 30;
MG = 1; CR = 1; MY = 0; MR = 0; CG = 0; CY = 0; //主干道通行
end

```

```

else if((C == 1)&&((T1 == 0) || (F==0))&&(G<=487))
    begin
        state = s2;
        T1 = 5;           //设置 5s 的黄灯倒计时长
        T2 = 5;
        MY = 1; CR = 1; MG = 0; MR = 0; CG = 0; CY = 0;
        end           //S1 状态下 C 不是第一次置高
                    //且 25s 倒计时结束或 C 置高之前值为 0 才进入 S2 状态
else
    begin
        state = s1;
        T1 = (T1 - 1);
        T2 = (T2 - 1);
        MG = 1; CR = 1; MY = 0; MR = 0; CG = 0; CY = 0; //主干道
    end

s2: if(T1 == 0)
    begin
        state = s3;
        T1 = 21;
        T2 = 16;
        MR = 1; CG = 1; MG = 0; MY = 0; CR = 0; CY = 0;
        end           //5s 黄灯倒计时结束时进入状态 3

else
    begin
        state = s2;
        T1 = (T1 - 1);
        T2 = (T2 - 1);
    end           //5s 黄灯倒计时中

s3: if(C == 0)           //乡村道路通行时，如果 C 为 0（表示乡村道路无车）
    begin
        state = s4;       //则进入 S4 状态（5s 黄灯倒计时），从而让主干道通行
        T1 = 5;
        T2 = 5;
        MR = 1; CY = 1; MG = 0; MY = 0; CG = 0; CR = 0;
        end
else if(T2 == 0)
    begin
        state = s4;           //乡村公路绿灯倒计时结束后进入状态 4
        T1 = (T1 - 1);
        T2 = 4;
        MR = 1; CY = 1; MG = 0; MY = 0; CG = 0; CR = 0;
    end

```

```

        end
    else
        begin
            state = s3;                //乡村公路绿灯倒计时中
            T1 = (T1 - 1);
            T2 = (T2 - 1);
        end
s4: if(T2 == 0)
    begin
        state = s1;
        T1 = 25;
        T2 = 30;
        MG = 1; CR = 1; MY = 0; MR = 0; CG = 0; CY = 0;
    end                                //如果 5s 黄灯倒计时结束重新进入状态 1

    else
        begin
            state = s4;
            T1 = (T1 - 1);
            T2 = (T2 - 1);            //5s 黄灯倒计时中
        end

    end
default:
    begin
        state = s1;
        T1 = 25;
        T2 = 30;
        MG = 1; CR = 1; MY = 0; MR = 0; CG = 0; CY = 0;
    end                                //其他情况进入状态 1
endcase
F=C;                                //把 C 之前的值赋给 F
G=(G-1);                            //G 每经过 1s 值减一
end
endmodule

```

### (3) 数码管译码显示模块设计程序

```

module nixie_tube(RET, CLK, MainCountdown, CountryCountdown, ML, MR, CL, CR, C);
input CLK, RET, C;
input [4:0]MainCountdown, CountryCountdown;
output [7:0]ML, MR, CL, CR;

```



```

reg [7:0]ML, MR, CL, CR;
reg [4:0]m;                                //变量 m 是中间变量
/* 数码管译码显示模块的端口有 RET, CLK, MainCountdown, CountryCountdown, ML,
MR, CL, CR, C, 其中 RET, CLK, MainCountdown, CountryCountdown, C 是输入端口, ML, MR,
CL, CR 是输出端口, RET 端口接初始化开关, CLK 端口接 1MHz 的时钟脉冲, MainCountdown,
CountryCountdown 端口分别接主干道和乡村道路交通灯的倒计时信息 T1、T2, ML, MR, CL,
CR 端口分别输出显示在 4 个数码管上数字 0-9 */

```

```

task display;                                //定义任务 display 帮助数码管译码输出倒计时
output [7:0]N1, N2;                          //任务有两个输出量 N1, N2
case(m)                                       //借助 case 语句对可能出现的倒计时数值 00, 01,
                                              //02..... 29, 30 逐一参照数码管八段译码对
                                              //照表进行译码, 从而使倒计时数值通过数码管
                                              //准确地显示出来, 下面为逐一译码的过程:

0: begin
    N1 = 8'b11000000;                        //倒计时数值为 00, 则 N1=0, N2=0, 所以
    N2 = 8'b11000000; // N1 = 8'b11000000=N2, 之后的数值 01 至 30 亦如此译码
End                                           //故不再赘述

1: begin
    N1 = 8'b11000000;
    N2 = 8'b11111001;
end

2: begin
    N1 = 8'b11000000;
    N2 = 8'b10100100;
end

3: begin
    N1 = 8'b11000000;
    N2 = 8'b10110000;
end

4: begin
    N1 = 8'b11000000;
    N2 = 8'b10011001;
end

5: begin
    N1 = 8'b11000000;
    N2 = 8'b10010010;
end

6: begin
    N1 = 8'b11000000;
    N2 = 8'b10000010;
end

```

```

7:  begin
    N1 = 8' b11000000;
    N2 = 8' b11111000;
    end
8:  begin
    N1 = 8' b11000000;
    N2 = 8' b10000000;
    end
9:  begin
    N1 = 8' b11000000;
    N2 = 8' b10010000;
    end
10: begin
    N1 = 8' b11111001;
    N2 = 8' b11000000;
    end
11: begin
    N1 = 8' b11111001;
    N2 = 8' b11111001;
    end
12: begin
    N1 = 8' b11111001;
    N2 = 8' b10100100;
    end
13: begin
    N1 = 8' b11111001;
    N2 = 8' b10110000;
    end
14: begin
    N1 = 8' b11111001;
    N2 = 8' b10011001;
    end
15: begin
    N1 = 8' b11111001;
    N2 = 8' b10010010;
    end
16: begin
    N1 = 8' b11111001;
    N2 = 8' b10000010;
    end
17: begin
    N1 = 8' b11111001;
    N2 = 8' b11111000;
    end

```

```

18: begin
    N1 = 8'b11111001;
    N2 = 8'b10000000;
end
19: begin
    N1 = 8'b11111001;
    N2 = 8'b10010000;
end
20: begin
    N1 = 8'b10100100;
    N2 = 8'b11000000;
end
21: begin
    N1 = 8'b10100100;
    N2 = 8'b11111001;
end
22: begin
    N1 = 8'b10100100;
    N2 = 8'b10100100;
end
23: begin
    N1 = 8'b10100100;
    N2 = 8'b10110000;
end
24: begin
    N1 = 8'b10100100;
    N2 = 8'b10011001;
end
25: begin
    if(C==0)                                     //如果C为0则数码管不显示25倒计时信息
begin
N1 = 8'b11111111;                             //可以通过数码管八段码全部置高实现
N2 = 8'b11111111;
end
else
begin
    N1 = 8'b10100100;
    N2 = 8'b10010010;
end
end
26: begin
    N1 = 8'b10100100;
    N2 = 8'b10000010;
end

```

```

27: begin
    N1 = 8'b10100100;
    N2 = 8'b11111000;
end
28: begin
    N1 = 8'b10100100;
    N2 = 8'b10000000;
end
29: begin
    N1 = 8'b10100100;
    N2 = 8'b10010000;
end
30: begin
if(C==0)    //如果 C 为 0 则数码管不显示 25 倒计时信息
begin
N1 = 8'b11111111;
N2 = 8'b11111111;    //可以通过数码管八段码全部置高实现
end
else
begin
N1 = 8'b10110000;
N2 = 8'b11000000;
end
end
default: begin
    N1 = 8'b11000000;
    N2 = 8'b11000000;
end
endcase
endtask

always @(posedge CLK)
if(!RET)
begin
    ML = 8'b11111111;
    MR = 8'b11111111;
    CL = 8'b11111111;
    CR = 8'b11111111;    //如果 RET 为 0 则 4 个数码管都不显示
end
else
begin
m = MainCountdown;    //将主干道倒计时数值赋给 m
display(ML, MR);    //调用过程块 display 显示倒计时信息

```

```

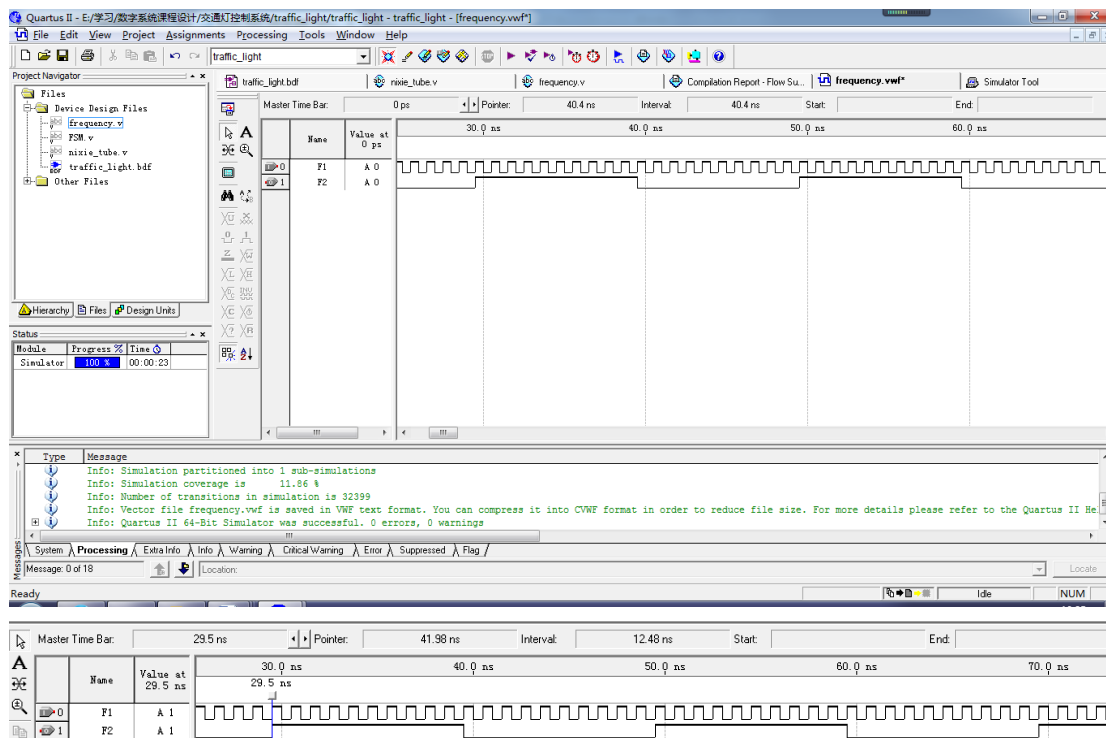
m = CountryCountdown; //将乡村公路倒计时数值赋给 m
display(CL, CR);       //调用过程块 display 显示倒计时信息
end
endmodule

```

## 2.4 仿真图及说明

### (1) 分频模块

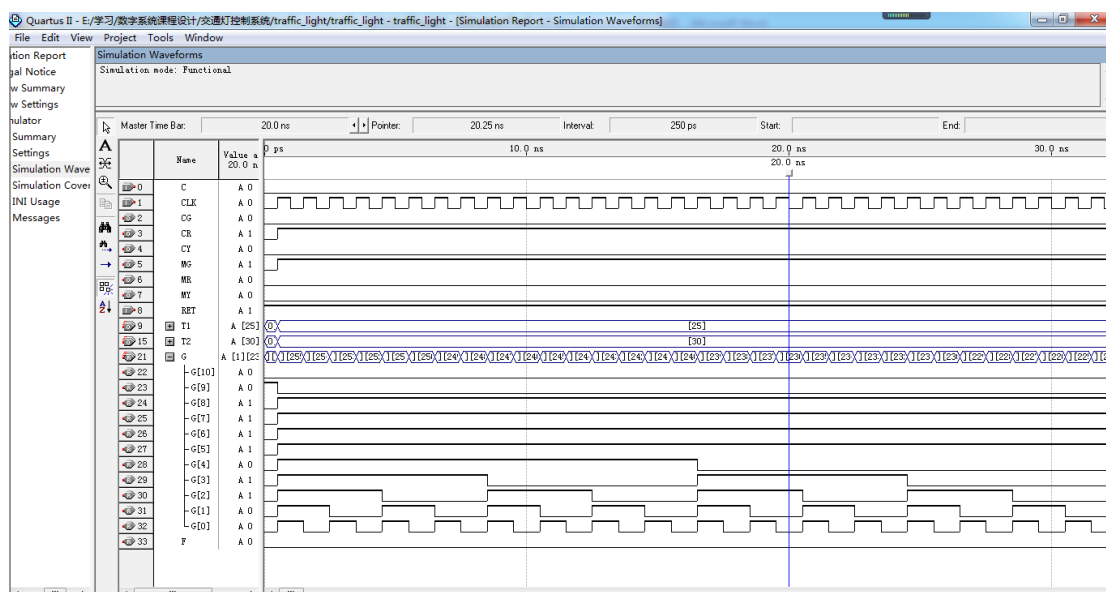
分频系数 K 设定为 10（因为源代码中的 1000000 太大，仿真效果不好）



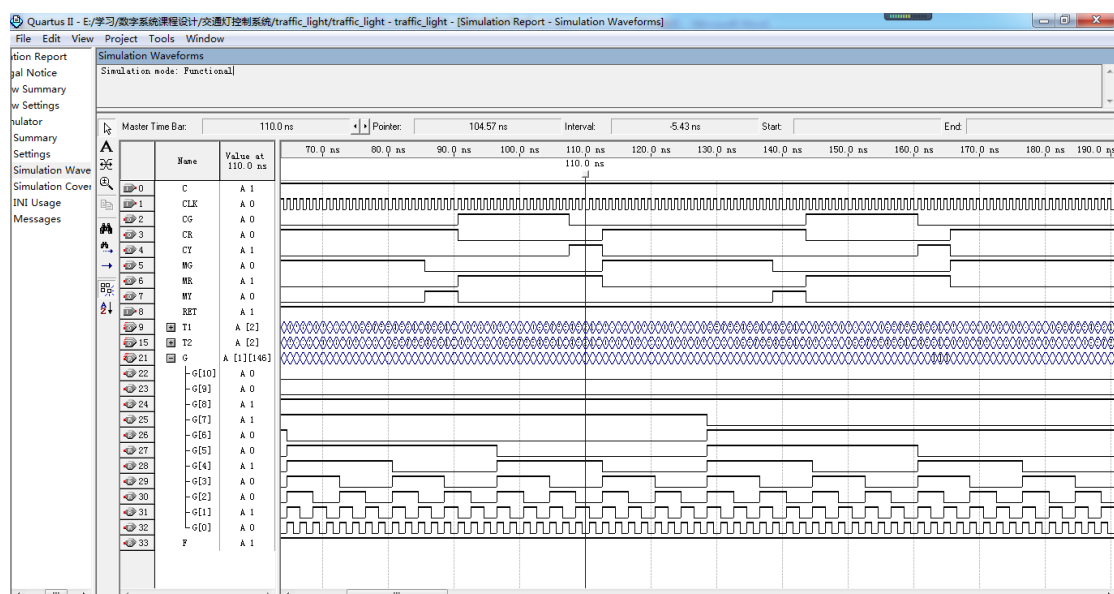
由功能仿真波形图可见分频得到的F2时钟脉冲信号的频率是分频前的F1频率的2K倍，所以源代码中K取1000000是合适的。

### (2) 状态机模块

当 RET=1, C=0 时:



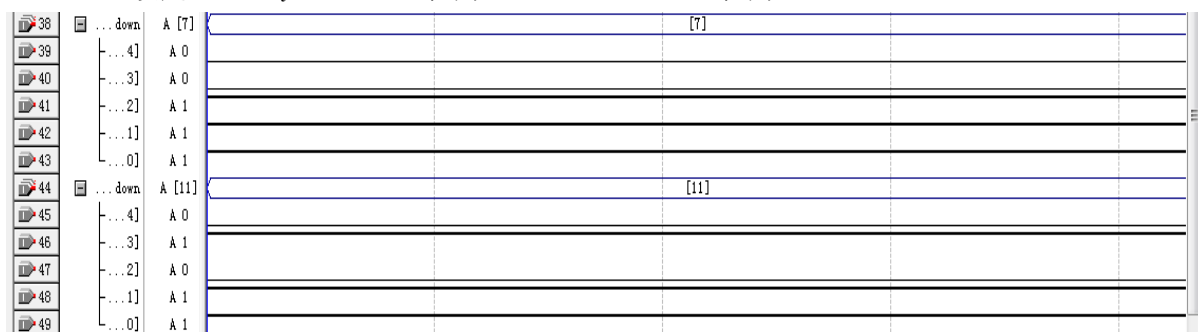
可见系统保持在 S1 状态，与实验要求一致；  
当 RET=1, C=1 时：



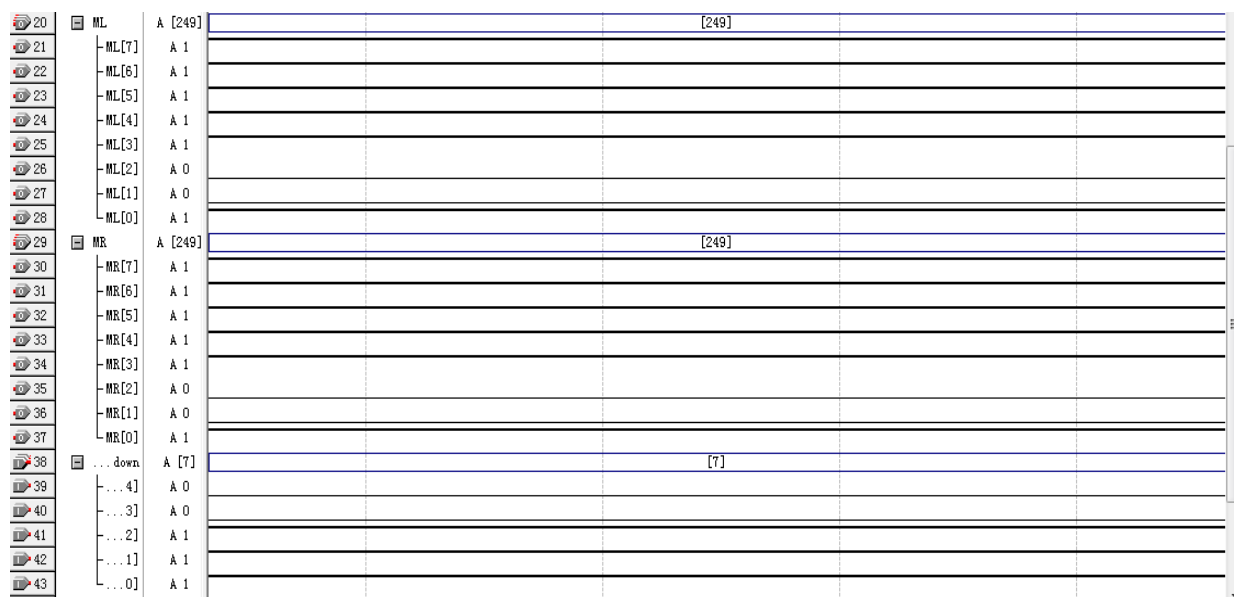
可见系统在四个状态中依次循环，与实验要求一致，状态机模块程序设计无误。

### (3) 数码管译码显示模块

设定 CountryCountdown 值为 7，MainCountdown 值为 11：



功能仿真图中对应的 CL, CR 和 ML, MR 分别为: 11000000, 11111000 和 11111001, 11111001

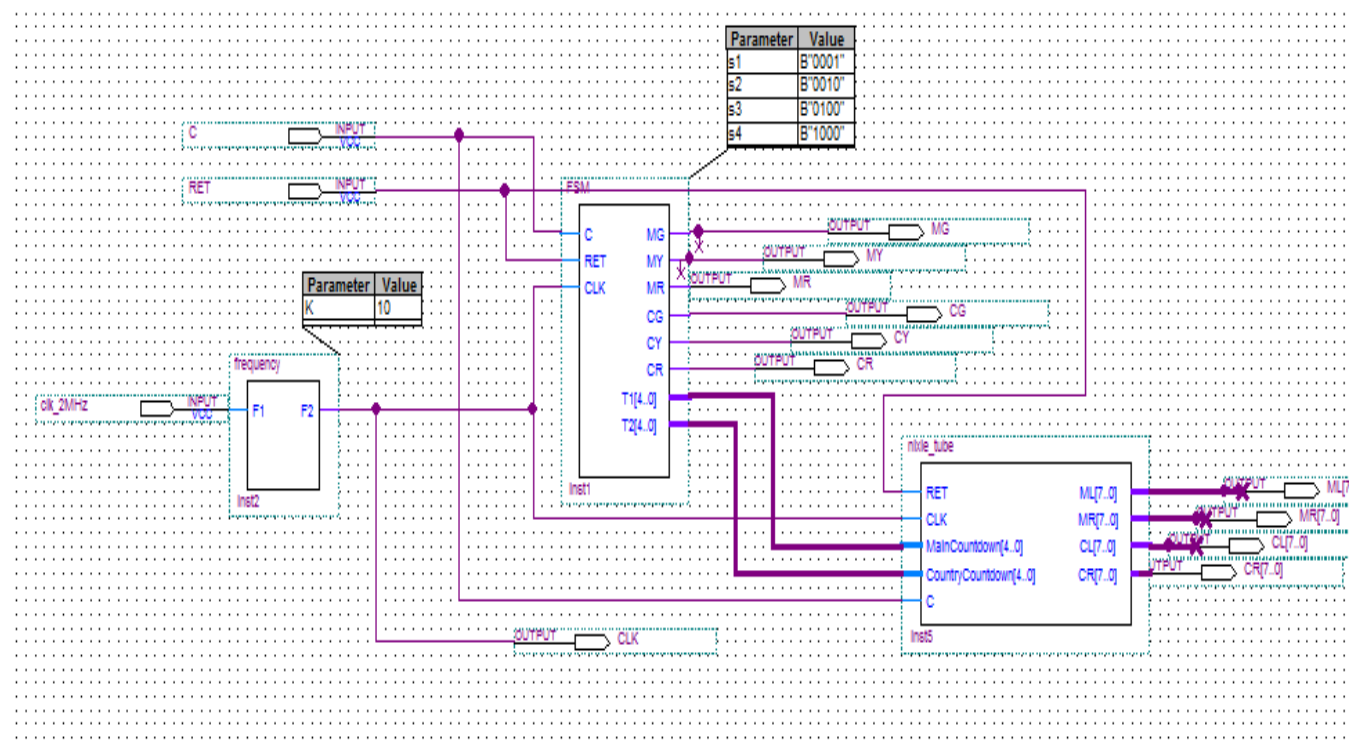


2	CL	A [192]	[192]
3	CL[7]	A 1	
4	CL[6]	A 1	
5	CL[5]	A 0	
6	CL[4]	A 0	
7	CL[3]	A 0	
8	CL[2]	A 0	
9	CL[1]	A 0	
10	CL[0]	A 0	
11	CR	A [248]	[248]
12	CR[7]	A 1	
13	CR[6]	A 1	
14	CR[5]	A 1	
15	CR[4]	A 1	
16	CR[3]	A 1	
17	CR[2]	A 0	
18	CR[1]	A 0	
19	CR[0]	A 0	

参考数码管八段译码对照表知数码管显示的分别是 0，7 和 1，1；与设定的交通灯倒计时数值 CountryCountdown（值为 7）和 MainCountdown（值为 11）一致，故数码管译码显示模块程序设计无误。

### 三、 总体设计与仿真

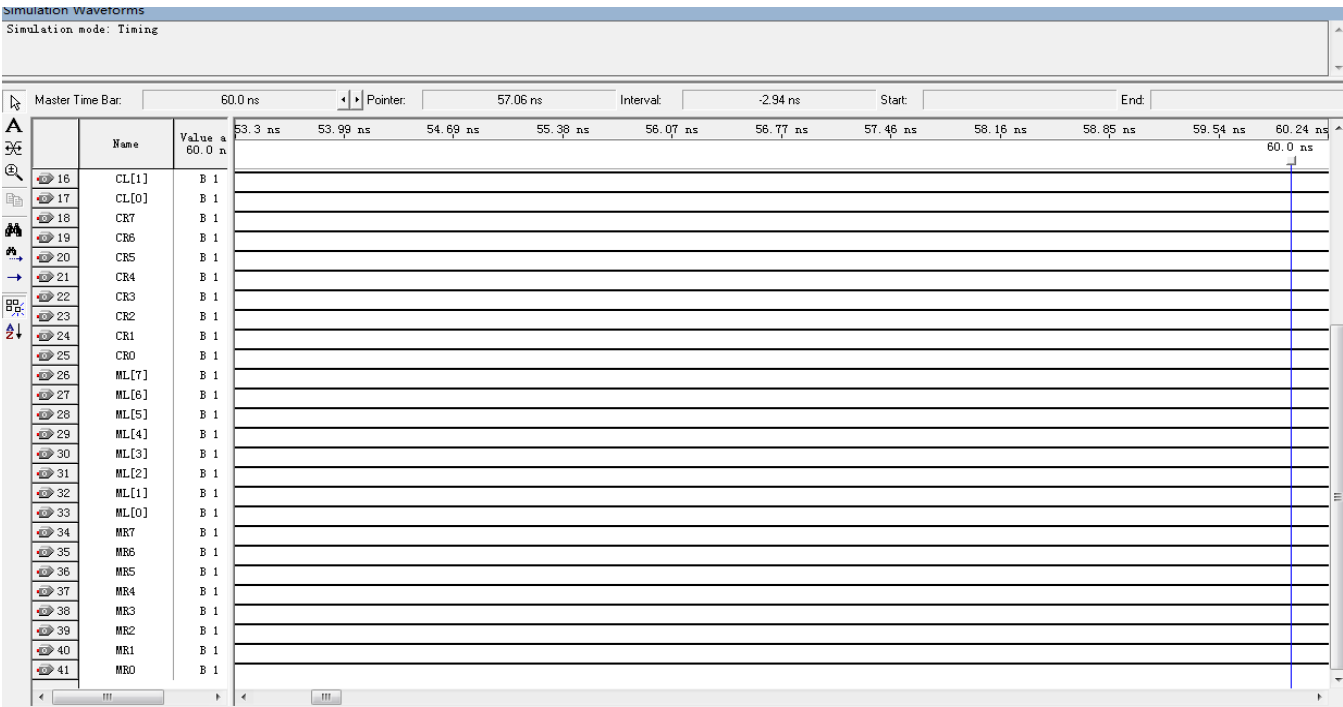
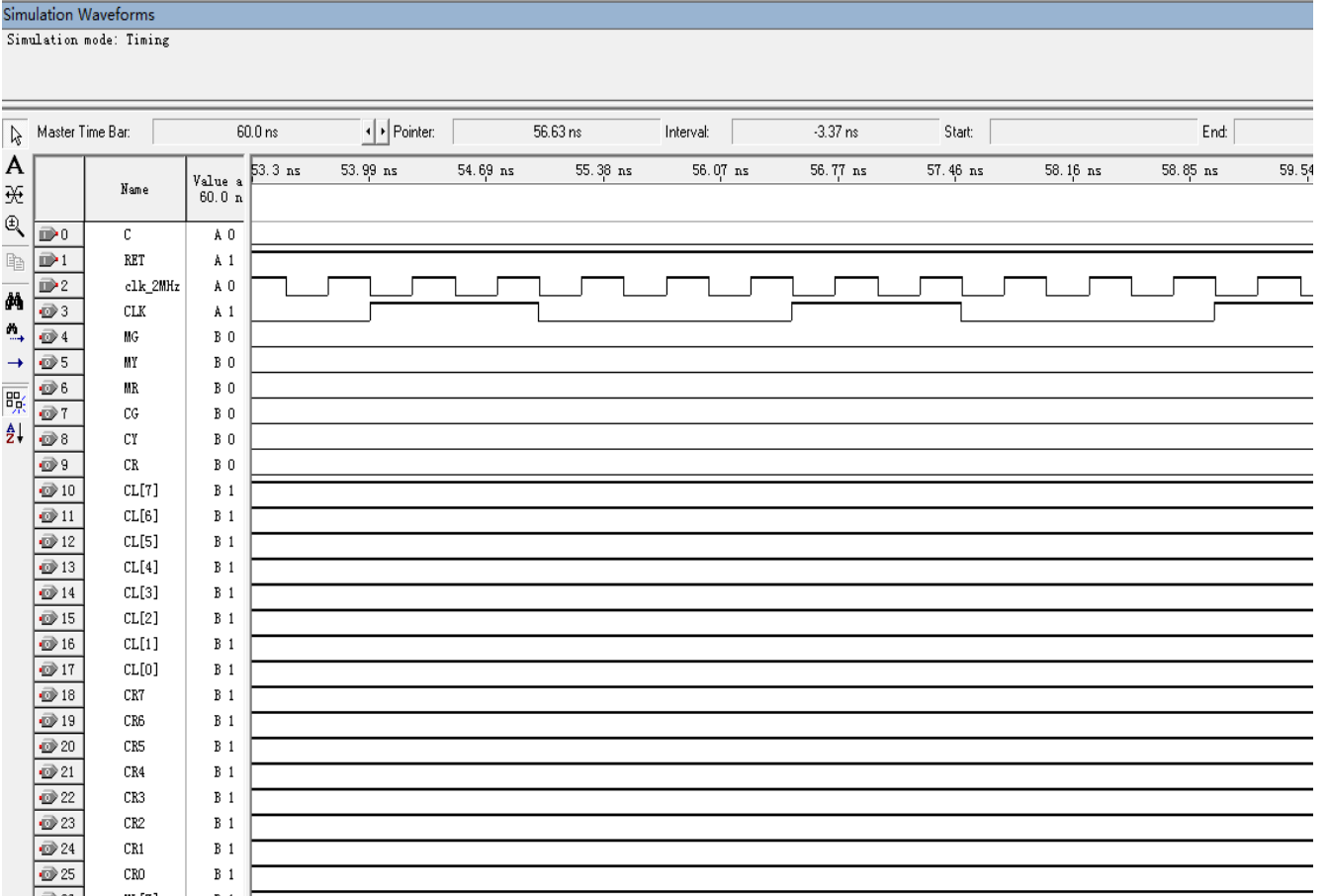
#### 3.1 顶层设计图及说明



FPGA 顶层设计图包含三个模块，从左到右依次为：分频模块、状态机模块和数码管译码显示模块；顶层设计图共有三个输入端口：C, RET, clk\_2MHz；11 个输出端口：CLK(为了便于在仿真波形图中显示分频后的时钟脉冲频率), MG, MY, MR, CG, CY, CR, ML[7:0], MR[7:0], CL[7:0], CR[7:0]；应注意 MR(CR)与 MR[7:0] (CR[7:0]) 同名但不同位宽，是两个不同的变量，前者是主干道(乡村道路)红灯信号，后者接显示主干道(乡村道路)交通灯倒计时个位数的数码管)。

3.2 仿真图及说明

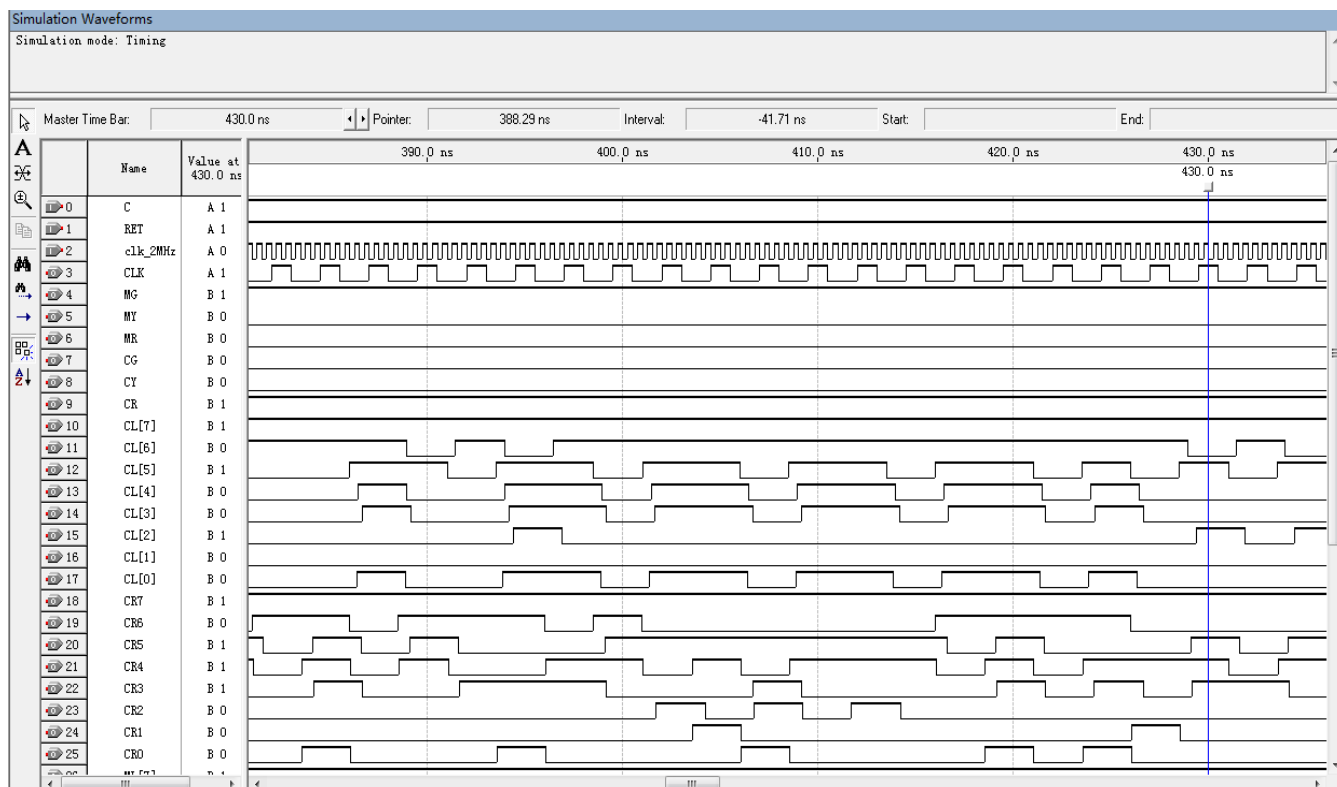
为了便于显示和检查仿真波形，设定分频系数 K 为 5，待分频的输入时钟脉冲 clk\_2MHz 的周期设为 0.5ns，分频后的时钟脉冲为 CLK 信号，RER=1,C=0 时：



可见系统保持在 S1 状态，与实验要求一致；



当 RET=1, C=1 时:



可见系统在四个状态中依次循环，与实验要求一致，顶层程序设计无误。

### 3.3 实验结果

将交通灯控制系统下载到实验板上进行实验，获得实验结果：

首先系统处于 S1 状态，主干道绿灯 (MG)，乡村道路红灯 (CR)，如果 C 一直为 0（乡村道路无车），则一直保持 S1 状态；当 C 信号第一次被置高（表示乡村公路有车），系统保持 S1 状态 25 秒（当乡村公路有车时，要求主干道通车时间已经超过它的最短通车时间 25 秒时，才禁止主干道通行，让乡村公路通行），接着系统进入 S2 状态，主干道黄灯 (MY)，乡村道路红灯 (CR)，5 秒的黄灯倒计时结束后，系统进入 S3 状态，主干道红灯 (MR)，乡村道路绿灯 (CG)，乡村道路绿灯 16 秒倒计时结束后系统进入 S4 状态，主干道红灯 (MR)，乡村道路黄灯 (CY)，开始 5 秒的黄灯倒计时，如果在乡村道路绿灯倒计时的 S3 过程中 C 信号突然被置高，则直接进入 S4 状态，S4 状态黄灯 5 秒倒计时结束后进入 S1 状态；如果 C 信号保持为 1，则系统继续 S1→S2→S3→S4→S1 的循环，如果 C 信号为 0，则一直保持在 S1 状态；S1 状态下 C 信号再次（不是第一次）被置高则系统立刻进入 S2 状态，5 秒的黄灯倒计时后进入 S3 状态让乡村道路通行 16 秒，之后再依次进入 S4、S1 状态，如此循环。且乡村道路和主干道交通灯倒计时数值通过四个数码管准确同步地显示出来。此外还实现了拓展部分：若乡村道路无车且主干道通行则熄灭四个数码管，不显示倒计时数值。

## 四. 总结

### 4.1 实验结果分析

经验收，实验结果完全满足交通灯控制系统课题设计要求，反映出各个模块功能完善，整体衔接合适，此外还完成了实验拓展部分：若乡村道路无车且主干道通行则熄灭四个数码管，不显示倒计时数值。

### 4.2 问题解决方法

（1）问题 1：如何实现拓展部分——若乡村道路无车且主干道通行则熄灭四个数码管，不显示倒计时数值

解决方法：为数码管译码模块编写一个输入端口接受 C 的数值，当乡村道路无车时，C=0，译码部分输出 8'b11111111，将 4 个八段数码管全部置高位，就可以熄灭显示的数字。

（2）问题 2：如何实现数码管显示倒计时数值

解决方法：参照数码管八段译码对照表，将输入数码管译码显示模块的两位倒计时数值拆分为个位和十位两个 0-9 数字，分别显示在两个数码管上，如 01 可以拆为 0 和 1 两个数字分别同步显示出来。

### 4.3 心得体会

（1）设计逻辑比较复杂的程序时，千万不可以一上手就在电脑上打字，一定要先拿起笔在纸上写出逻辑关系和注意要点，帮助脑子理清思路再上手电脑编程，这样可以提高编程效率和准确率，事半功倍地完成任务。

（2）比起长学期的课程，短学期的课程要求我们在较短的时间里系统地学习大量知识，完成课题研究，这教会了我一种新的学习方法——边做边学：在快速地阅读教材，掌握了基本知识之后立刻上手编程，遇见不懂的再在书中学习相应的知识点，学会了就继续攻克任务，碰见下一个问题再学习。。。。。。如此循环，直到完成全部任务。我发现这样学习可以大大提高工作效率也可以紧密联系书本知识和实际需求，给了我学习新知识的动力，让学习和研究变得无比有趣。

**参考书目：**

- [1] 王金明., 数字系统设计与 Verilog HDL(第 5 版), 北京, 电子工业出版社, 2014