数字逻辑电路专题实验报告

—— 交通灯控制器设计

班级	
姓名	
学号	
日期	2021 年 5 月 31 日

目录

1.实验目的

通过设计一个交通灯控制器，达到以下目标

- 能够应用所学的电路知识完成较复杂逻辑电路的设计
- 培养独立分析问题解决问题的能力
- 激发创造性思维
- 提高数字技术方面的创新实践能力。

2.实验项目名称与实现的功能目标

- 项目名称：交通灯控制器设计
- 功能目标：设计一个由一条主干道和一条支干道的汇合点形成的十字交叉路口的交通灯控制器，

  (1) 主、支道各设有一个绿、黄、红指示灯，两个显示数码管。

  (2) 主干道处于常允许通行状态，而支干道有车来才允许通行。

  (3) 当主、支道均有车时，两者交替允许通行，主干道每次放行 45 s，支干道每次放行 25 s，在每次由亮绿灯变成亮红灯的转换过程中，要亮 5 s 的黄灯作为过渡，并进行减计时显示。

3．系统设计方案

- 系统功能模块图

本系统功能模块图如图 3.1 所示

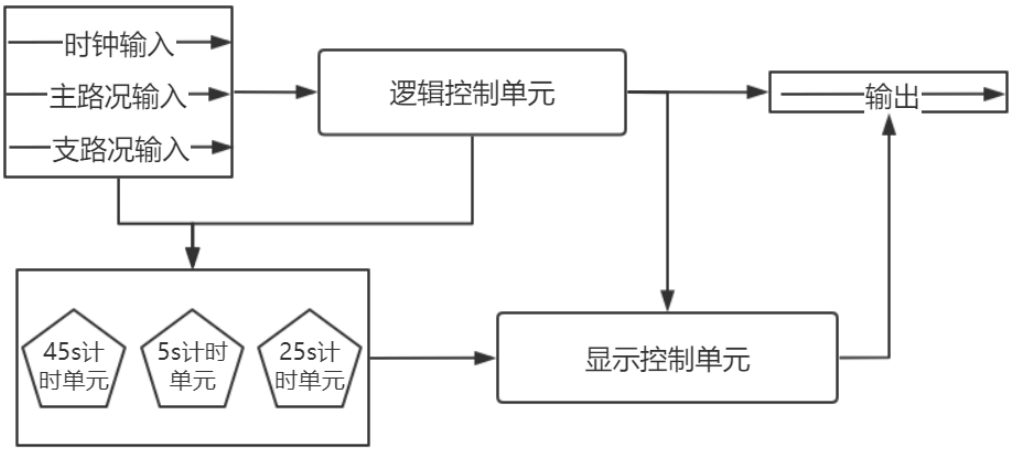

图 3.1 系统功能模块

- 逻辑关系与设计原理

本电路的输入包含时钟信号、主路是否有车以及支路是否有车。信号传入逻辑控制单元以完成逻辑判断。控制信号作用于各单元的使能端以达到整体控制的目的。各计时单元完成计时任务并产生两条路的交通灯信号，信号相差 5s 以体现黄灯。显示控制单元控制信号的输出，最终输出红绿灯信号以及倒计时信号。

4.各功能模块说明

4.1 计数器模块

计数器模块的 VHDL 代码如下

计数器模块
5s 计数模块
--CNT05S.VHD *LIBRARY IEEE;* *USE IEEE.STD_LOGIC_1164.ALL;* *USE IEEE.STD_LOGIC_UNSIGNED.ALL;* *ENTITY CNT05S IS* *PORT(CLK,EN05M,EN05B:IN STD_LOGIC;*

```
            DOUT5: OUT STD_LOGIC_VECTOR(7 DOWNTO 0));END ENTITY CNT
ARCHITECTURE ART OF CNT05S IS
   SIGNAL CNT3B: STD_LOGIC_VECTOR(2 DOWNTO 0);
   BEGIN
   PROCESS(CLK,EN05M,EN05B) IS
      BEGIN
      IF(CLK'EVENT AND CLK= '1')THEN
          IF EN05M='1' OR EN05B='1' THEN
             CNT3B<=CNT3B+1;
ELSE
             CNT3B<="000";
          END IF;
      END IF;
   END PROCESS;
   PROCESS(CNT3B) IS
      BEGIN
      CASE CNT3B IS
      WHEN "000"=>DOUT5<="00000101";
      WHEN "001"=>DOUT5<="00000100";
      WHEN "010"=>DOUT5<="00000011";
      WHEN "011"=>DOUT5<="00000010";
      WHEN "100"=>DOUT5<="00000001";
      WHEN OTHERS=>--DOUT5<="00000000";
      END CASE;
   END PROCESS;
END ARCHITECTURE ART;
```

25s 计数模块

```
--CNT25S.VHD
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT25S IS
   PORT(SB,SM,CLK,EN25:IN STD_LOGIC;
            DOUT25M,DOUT25B:OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
```

```
END ENTITY CNT25S;
ARCHITECTURE ART OF CNT25S IS
   SIGNAL CNT5B: STD_LOGIC_VECTOR(4 DOWNTO 0);
   BEGIN
   PROCESS(SB,SM,CLK,EN25) IS
      BEGIN
      IF SB='0' OR SM='0' THEN
         CNT5B<=CNT5B-CNT5B-1;
ELSIF(CLK'EVENT AND CLK= '1')THEN
         IF EN25='1' THEN
            CNT5B<=CNT5B+1;
         ELSIF EN25='0'THEN
            CNT5B<=CNT5B-CNT5B-1;
        END IF;
      END IF;
   END PROCESS;
   PROCESS(CNT5B) IS
      BEGIN
      CASE CNT5B IS
WHEN "00000"=>DOUT25B<="00100101"; DOUT25M<="00110000";
      WHEN "00001"=>DOUT25B<="00100100"; DOUT25M<="00101001";
      WHEN "00010"=>DOUT25B<="00100011"; DOUT25M<="00101000";
      WHEN "00011"=>DOUT25B<="00100010"; DOUT25M<="00100111";
      WHEN "00100"=>DOUT25B<="00100001"; DOUT25M<="00100110";
      WHEN "00101"=>DOUT25B<="00100000"; DOUT25M<="00100101";
      WHEN "00110"=>DOUT25B<="00011001"; DOUT25M<="00100100";
      WHEN "00111"=>DOUT25B<="00011000"; DOUT25M<="00100011";
      WHEN "01000"=>DOUT25B<="00010111"; DOUT25M<="00100010";
      WHEN "01001"=>DOUT25B<="00010110"; DOUT25M<="00100001";
      WHEN "01010"=>DOUT25B<="00010101"; DOUT25M<="00100000";
      WHEN "01011"=>DOUT25B<="00010100"; DOUT25M<="00011001";
WHEN "01100"=>DOUT25B<="00010011"; DOUT25M<="00011000";
      WHEN "01101"=>DOUT25B<="00010010"; DOUT25M<="00010111";
```

```
    WHEN "01110"=>DOUT25B<="00010001"; DOUT25M<="00010110";
    WHEN "01111"=>DOUT25B<="00010000"; DOUT25M<="00010101";
    WHEN "10000"=>DOUT25B<="00001001"; DOUT25M<="00010100";
    WHEN "10001"=>DOUT25B<="00001000"; DOUT25M<="00010011";
    WHEN "10010"=>DOUT25B<="00000111"; DOUT25M<="00010010";
    WHEN "10011"=>DOUT25B<="00000110"; DOUT25M<="00010001";
    WHEN "10100"=>DOUT25B<="00000101"; DOUT25M<="00010000";
    WHEN "10101"=>DOUT25B<="00000100"; DOUT25M<="00001001";
    WHEN "10110"=>DOUT25B<="00000011"; DOUT25M<="00001000";
    WHEN "10111"=>DOUT25B<="00000010"; DOUT25M<="00000111";
    WHEN "11000"=>DOUT25B<="00000001"; DOUT25M<="00000110";
    WHEN OTHERS=>DOUT25B<="00000000"; DOUT25M<="00000000";
END CASE;
  END PROCESS;
END ARCHITECTURE ART;
```

45s 计数模块

```
--CNT45S.VHD
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT45S IS
  PORT(SB,CLK,EN45:IN STD_LOGIC;
       DOUT45M,DOUT45B:OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END ENTITY CNT45S;
ARCHITECTURE ART OF CNT45S   IS
  SIGNAL CNT6B: STD_LOGIC_VECTOR(5 DOWNTO 0);
  BEGIN
  PROCESS(SB,CLK,EN45) IS
    BEGIN
    IF SB='0'THEN CNT6B<=CNT6B-CNT6B-1;
      ELSIF(CLK'EVENT AND CLK= '1')THEN
       IF EN45='1'THEN CNT6B<=CNT6B+1;
ELSIF EN45='0'THEN CNT6B<=CNT6B-CNT6B-1;
       END IF;
```

```
    END IF;
  END PROCESS;
  PROCESS(CNT6B) IS
    BEGIN
    CASE CNT6B IS
WHEN "000000"=>DOUT45M<="01000101"; DOUT45B<="01010000";
    WHEN "000001"=>DOUT45M<="01000100"; DOUT45B<="01001001";
    WHEN "000010"=>DOUT45M<="01000011"; DOUT45B<="01001000";
    WHEN "000011"=>DOUT45M<="01000010"; DOUT45B<="01000111";
    WHEN "000100"=>DOUT45M<="01000001"; DOUT45B<="01000110";
    WHEN "000101"=>DOUT45M<="01000000"; DOUT45B<="01000101";
    WHEN "000110"=>DOUT45M<="00111001"; DOUT45B<="01000100";
    WHEN "000111"=>DOUT45M<="00111000"; DOUT45B<="01000011";
    WHEN "001000"=>DOUT45M<="00110111"; DOUT45B<="01000010";
    WHEN "001001"=>DOUT45M<="00110110"; DOUT45B<="01000001";
    WHEN "001010"=>DOUT45M<="00110101";DOUT45B<="01000000";

  WHEN "001011"=>DOUT45M<="00110100"; DOUT45B<="01101001";
    WHEN "001100"=>DOUT45M<="00110011"; DOUT45B<="00111000";
    WHEN "001101"=>DOUT45M<="00110010"; DOUT45B<="00110111";
    WHEN "001110"=>DOUT45M<="00110001"; DOUT45B<="00110110";
    WHEN "001111"=>DOUT45M<="00110000"; DOUT45B<="00110101";
    WHEN "010000"=>DOUT45M<="00101001"; DOUT45B<="00110100";
    WHEN "010001"=>DOUT45M<="00101000"; DOUT45B<="00110011";
    WHEN "010010"=>DOUT45M<="00100111"; DOUT45B<="00110010";
    WHEN "010011"=>DOUT45M<="00100110"; DOUT45B<="00110001";
    WHEN "010100"=>DOUT45M<="00100101"; DOUT45B<="00110000";
    WHEN "010101"=>DOUT45M<="00100100"; DOUT45B<="00101001";
    WHEN "010110"=>DOUT45M<="00100011"; DOUT45B<="00101000";
WHEN "010111"=>DOUT45M<="00100010"; DOUT45B<="00100111";
    WHEN "011000"=>DOUT45M<="00100001"; DOUT45B<="00100110";
    WHEN "011001"=>DOUT45M<="00100000"; DOUT45B<="00100101";
    WHEN "011010"=>DOUT45M<="00011001"; DOUT45B<="00100100";
    WHEN "011011"=>DOUT45M<="00011000"; DOUT45B<="00100011";
```

```
        WHEN "011100"=>DOUT45M<="00010111"; DOUT45B<="00100010";
        WHEN "011101"=>DOUT45M<="00010110"; DOUT45B<="00100001";
        WHEN "011110"=>DOUT45M<="00010101"; DOUT45B<="00100000";
        WHEN "011111"=>DOUT45M<="00010100"; DOUT45B<="00011001";
        WHEN "100000"=>DOUT45M<="00010011"; DOUT45B<="00011000";
        WHEN "100001"=>DOUT45M<="00010010"; DOUT45B<="00010111";
        WHEN "100010"=>DOUT45M<="00010001"; DOUT45B<="00010110";
WHEN "100011"=>DOUT45M<="00010000"; DOUT45B<="00010101";
        WHEN "100100"=>DOUT45M<="00001001"; DOUT45B<="00010100";
        WHEN "100101"=>DOUT45M<="00001000"; DOUT45B<="00010011";
        WHEN "100110"=>DOUT45M<="00000111"; DOUT45B<="00010010";
        WHEN "100111"=>DOUT45M<="00000110"; DOUT45B<="00010001";
        WHEN "101000"=>DOUT45M<="00000101"; DOUT45B<="00010000";
        WHEN "101001"=>DOUT45M<="00000100"; DOUT45B<="00001001";
        WHEN "101010"=>DOUT45M<="00000011"; DOUT45B<="00001000";
        WHEN "101011"=>DOUT45M<="00000010"; DOUT45B<="00000111";
        WHEN "101100"=>DOUT45M<="00000001"; DOUT45B<="00000110";
        WHEN OTHERS=>DOUT45M<="00000000"; DOUT45B<="00000000";
END CASE;
   END PROCESS;
END ARCHITECTURE ART;
```

本计数器模块的作用是进行倒计时。用一个自增的信号进行计数，并将所计的数转换为两个交通灯的倒计时信号。

4.2 逻辑控制模块

逻辑控制模块的 VHDL 代码如下.

逻辑控制模块

```
--JTDKZ.VHD
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY JTDKZ IS
```

```
    PORT(CLK, SM, SB, RESET: IN STD_LOGIC;
         MR, MY, MG, BR, BY, BG: OUT STD_LOGIC);
END ENTITY JTDKZ;
ARCHITECTURE ART OF JTDKZ IS
  TYPE STATE_TYPE IS(A,B,C,D);
  SIGNAL STATE: STATE_TYPE;
  BEGIN
  CNT:PROCESS(CLK , RESET) IS
    VARIABLE S:INTEGER RANGE 0 TO 45;
    VARIABLE CLR,EN:BIT;
    BEGIN
IF RESET = '1' THEN
    S:= 0;
    STATE<= A;
    ELSIF(CLK'EVENT AND CLK='1')THEN
       IF CLR='0' THEN
        S:=0;
       ELSIF EN='0' THEN
          S:=S;
       ELSE
          S:=S+1;
       END IF;

CASE STATE IS
WHEN A=>MR<='0'; MY<='0';   MG<='1';
                    BR<='1'; BY<='0'; BG<='0';
                    IF(SB AND SM)='1' THEN
                       IF S=45 THEN
                          STATE<=B; CLR:='0'; EN:='0';
                       ELSE
                          STATE<=A; CLR:='1'; EN:='1';
                       END IF;
                    ELSIF(SB AND (NOT SM))='1' THEN
                       STATE<=B; CLR:='0'; EN:='0';
```

```
                    ELSE
                       STATE<=A; CLR:='1'; EN:='1';
                    END IF;
WHEN B=>MR<='0'; MY<='1'; MG<='0';
                    BR<='1'; BY<='0'; BG<='0';
                    IF S=5 THEN
                       STATE<=C;CLR:='0'; EN:='0';
                    ELSE
                       STATE<=B; CLR:='1'; EN:='1';
                    END IF;
WHEN C=>MR<='1'; MY<='0'; MG<='0';
                    BR<='0'; BY<='0'; BG<='1';
                    IF(SM AND SB)='1' THEN
                       IF S=25 THEN
                          STATE<=D; CLR:='0'; EN:='0';
                       ELSE
                          STATE<=C; CLR:='1'; EN:='1';
                       END IF;

                    ELSIF (SM AND (NOT SB)) = '1' THEN
                          STATE<=D; CLR:='0'; EN:='0';
                          ELSE
                                STATE<=C; CLR:='1'; EN:='1';
                          END IF;
WHEN D=>MR<='1'; MY<='0';    MG<='0';
                    BR<='0'; BY<='1'; BG<='0';
                    IF S=5 THEN
                       STATE<=A;CLR:='0'; EN:='0';
                    ELSE
                       STATE<=D; CLR:='1'; EN:='1';
                          END IF;
          END CASE;
      END IF;
   END PROCESS CNT;
```

```
END ARCHITECTURE ART;
```

本模块的作用在于建立两个交通灯的四个状态, 根据输入和现态来确定次态。

4.3 显示控制模块

本模块的 VHDL 代码如下

显示控制模块

```
--XSKZ.VHD
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY XSKZ IS
   PORT(EN45,EN25,EN05M,EN05B:IN STD_LOGIC;

AIN45M,AIN45B: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          AIN25M,AIN25B,AIN05: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          DOUTM,DOUTB: OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END ENTITY XSKZ;
ARCHITECTURE ART OF XSKZ IS
   BEGIN
   PROCESS(EN45,EN25,EN05M,EN05B) IS
      BEGIN
      IF EN45='1'THEN
DOUTM<=AIN45M(7 DOWNTO 0); DOUTB<=AIN45B(7 DOWNTO 0);
      ELSIF   EN05M='1'THEN
         DOUTM<=AIN05(7 DOWNTO 0); DOUTB<=AIN05(7 DOWNTO 0);
      ELSIF   EN25='1' THEN
         DOUTM<=AIN25M(7 DOWNTO 0); DOUTB<=AIN25B(7 DOWNTO (
      ELSIF   EN05B='1'THEN

DOUTM<=AIN05(7 DOWNTO 0); DOUTB<=AIN05(7 DOWNTO 0);
```

```
        END IF;
    END PROCESS;
END ARCHITECTURE ART;
```

本模块的作用在于根据使能信号选择正确的输出信号送给输出端。

5.测试结果的分析

本设计经过仿真和硬件检验，已经通过验收。

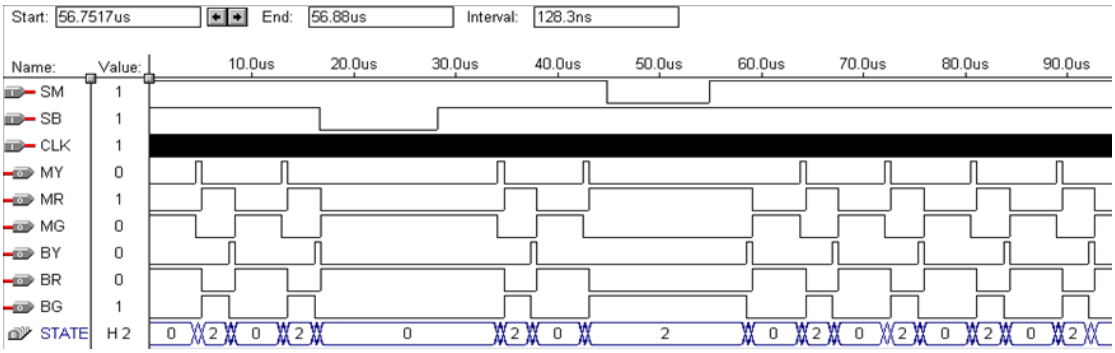

图 5.1 仿真时序图

图 5.1 中 MY, MR, MG 为主道黄、红、绿灯；BY, BR, BG 为支道黄、红、绿灯;SM, SB 代表主路，支路是否有车。图中主路绿灯、黄灯亮时，支路红灯亮；支路绿灯、黄灯亮时主路红灯亮；SM=0 时主路无车，主路红灯、支路绿灯亮；SB=0 时支路无车，主路绿灯、支路红灯亮。

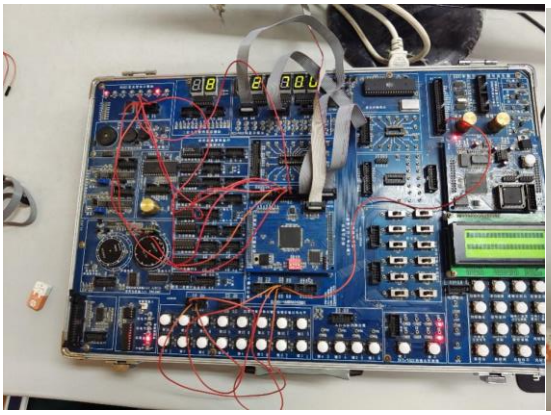

图 5.2（a） 测试结果图 1

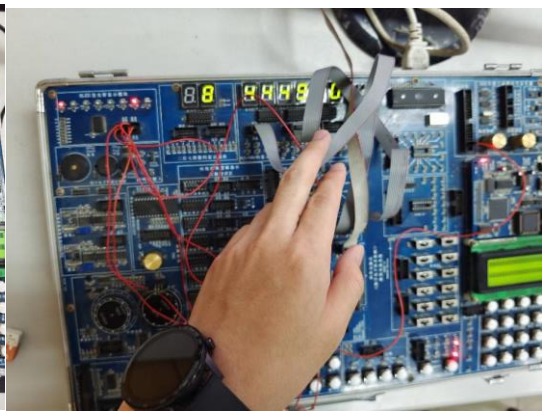

图 5.2（b） 测试结果图 2

图中左上角 LED 灯模块为灯光显示，灯 1~3 为主道绿、黄、红灯；灯 5~7 为支道绿、黄、红灯；中间上方前两位数字为主道计时，后两位数字为支道计时。测试结果图 2 中所示状态为主道绿灯，支道红灯；主道剩余通行时间为 44s，支道等待时间（包括黄灯）为 49s。

经测试，电路各项功能正常，达到预期目标。

6.实验总结与讨论

本实验设计主要参考了幻灯片文档和实验设计书的相关内容，在完成设计时遇到了一些困难。一是 Quartus II 软件的使用问题，有一些功能经多方查阅才知道怎样使用；二是程序编写的问题，这部分主要参考了幻灯片和数电教材的内容；三是引脚分配的问题，有一些引脚不可用，在分配时造成了一定障碍。经过本次数电实验设计，我了解了数字电路设计的基本方法，设计了简单的数字电路并成功实现了目标功能，深刻理解了所学的理论知识，为将来进一步学习做好了铺垫，也为实际工作积累了经验。

7.参考文献

[1]朱正东，伍卫国.数字逻辑与数字系统[M].北京:电子工业出版社,2015.8