实验一　组合逻辑电路实验（一）

一、实验目的

1、掌握编码器、译码器逻辑功能及使用方法；

2、掌握利用编码器、译码器进行电路设计的方法；

3、熟悉组合逻辑电路的测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC148、74HC138、74HC08、74HC20、74HC30。

三、实验内容

1、用两片优先编码器 74HC148 和必要的门电路，设计一个 2421 码的编码器电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

当 10 个输入信号中任一为 0 时，输出端即可输出对应的 2421 码。如图 7- 1 所示，在使用 Proteus 仿真时，当输入端 5 为 0 时，输出端 F3、F2、F1、F0 为 1011，符合 2421 码编码规则。

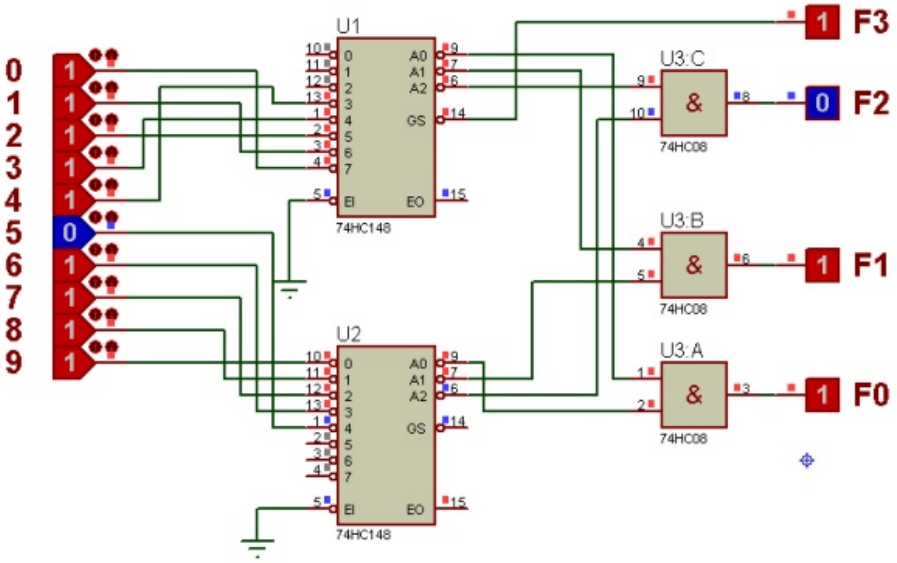

图 7- 1 2421 码编码器电路

2、用一片 3-8 线译码器 74HC138 和门电路设计一个一位全减器电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

输入端被减数 A、减数 B 和低位借位信号 C_{i-1}，输出端 S_i 表示差，C_i 表示借位信号。如图 7- 2 所示，在使用 Proteus 仿真时，当输入端为 010 时，输出为 11，实现全减器的功能。

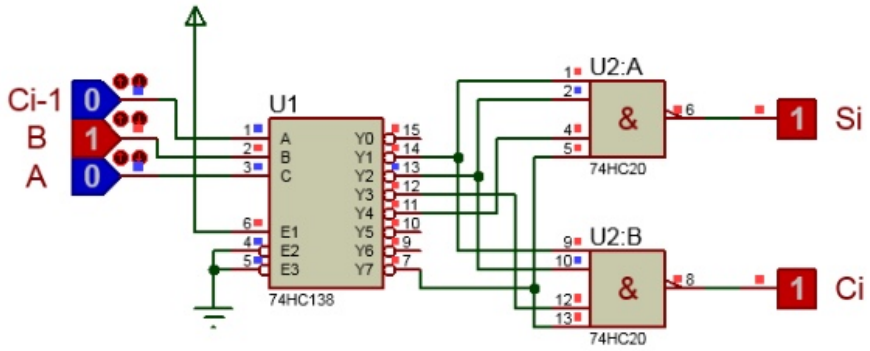

图 7- 2 一位全减器电路

3、用两片 3-8 线译码器 74HC138 和门电路设计一个奇偶校验电路，当输入的 4 个变量中有偶数个 1 时输出为 1，否则输出为 0。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

输入端四变量为 A3、A2、A1、A0，输出端 F。如图 7- 3 所示，在使用 Proteus 仿真时，当输入端为 1100 时，4 个变量中有两个 1，因此输出端 F 为 1。

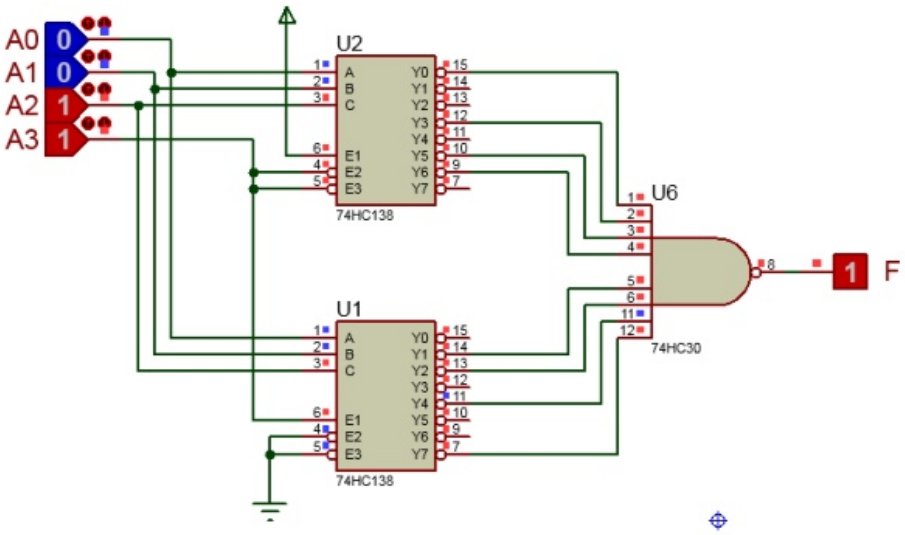

图 7- 3 奇偶校验电路

4、用两片 3-8 线译码器 74HC138 和门电路设计一个 8421 码转换为余 3 码的电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

输入端 8421 码为 A3、A2、A1、A0，输出端余 3 码为 F3、F2、F1、F0。如图 7- 4 所示，在使用 Proteus 仿真时，当输入的 8421 码为 0110 时，输出的余 3 码为 1001，符合余 3 码的编码规则。

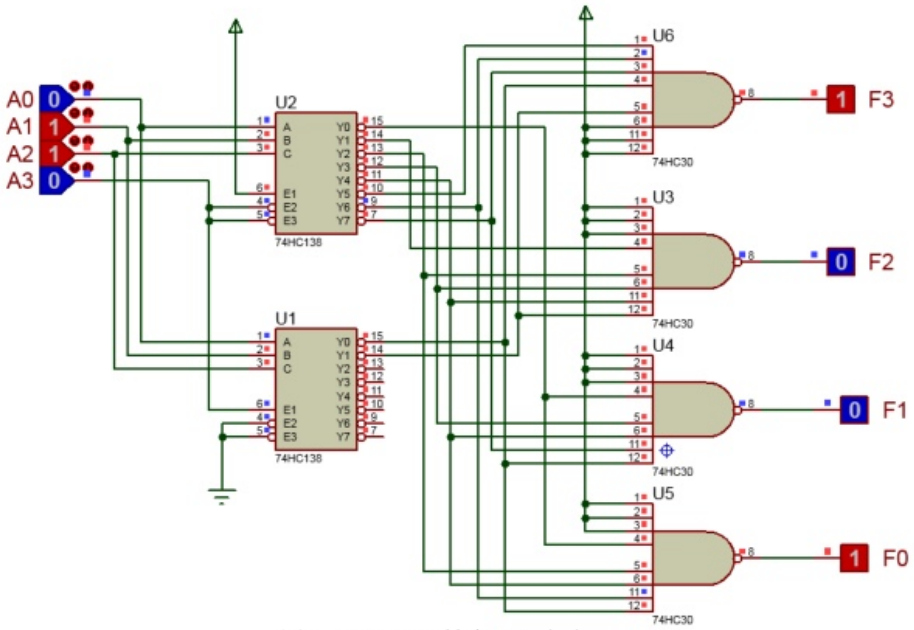

图 7- 4 8421 码转余 3 码电路

四、思考题

1、修改 2421 码的编码器电路，设计一个 8421 码的编码器电路。

2、修改一位全减器电路，设计一位全加器电路。

实验二　组合逻辑电路实验（二）

一、实验目的

1、掌握数据选择器逻辑功能及使用方法；

2、掌握利用数据选择器进行电路设计的方法；

3、进一步掌握组合逻辑电路的设计和测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC151、74HC153、74HC 138、74HC04。

三、实验内容

1、用一片 4 选 1 数据选择器 74HC153 设计一个数值大小判断电路，输入为 8421BCD 码，当输入的数字大于 1 小于 9 时，输出为 1，否则输出为 0。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

输入端 8421 码为 D、C、B、A，输出端为 F，74HC153 的地址输入端 B、A 分别接 8421 码的高位 D、C。如图 7- 1 所示，在使用 Proteus 仿真时，当输入的 8421 码为 0100 时，数值 6 在 2-8 之间，因此输出为 1。

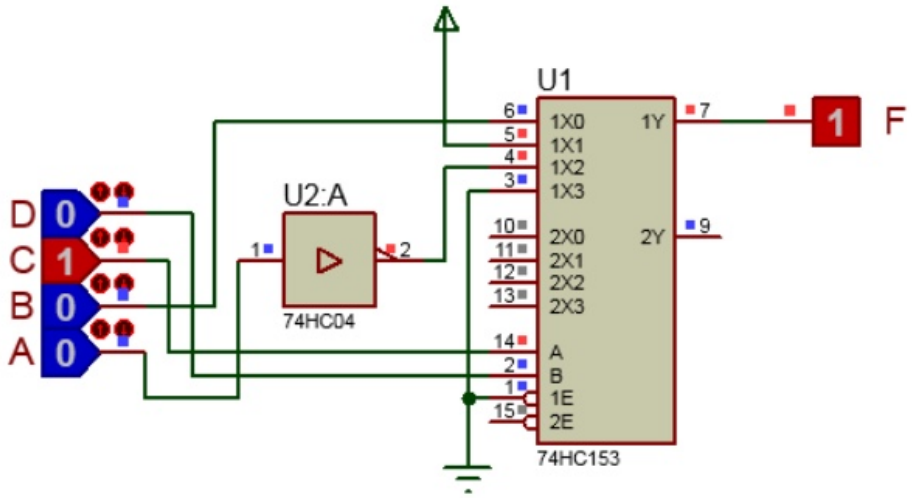

图 7- 1 数值大小判断电路

2、用一片数据选择器 74HC151 和门电路设计一个运算判断电路。输入为 4 位二进制数，当输入的数能被 5 或者 2 整除时输出 1，否则输出为 0。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

输入端二进制数为 D、C、B、A，输出端为 F，74HC151 的地址输入端 C、B、A 分别接二进制数的 D、C、B。如图 7- 2 所示，在使用 Proteus 仿真时，当输入的二进制数为 0101 时，数值 5 能被 5 整除，因此输出为 1。

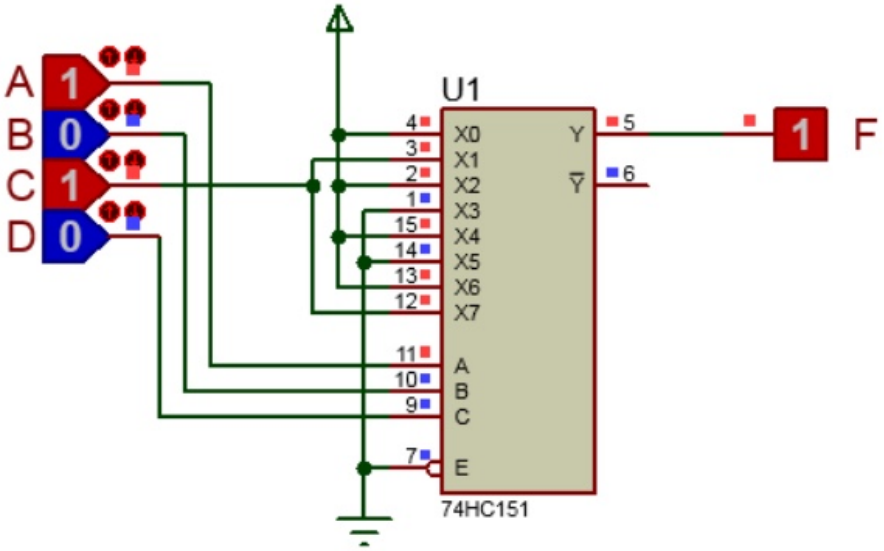

图 7- 2 运算判断电路

3、用一片数据选择器 74HC151 和一片译码器 74HC138 组成一个 6 路分时传送系统。测试在相同地址 A2A1A0 控制下，74HC151 的数据输入 X5~X0 和 74HC138 的输出 Y5~Y0 的对应关系。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

数据输入端为 X5、X4、X3、X2、X1、X0，地址输入端 C、B、A，数据输出端为 Y5、Y4、Y3、Y2、Y1、Y0。如图 7- 3 所示，在使用 Proteus 仿真时，当地址选择为 001 时，数据输入端 X1 的数据可以传输给 Y1，即 X1=Y1，其他输出端保持为 1 不变。

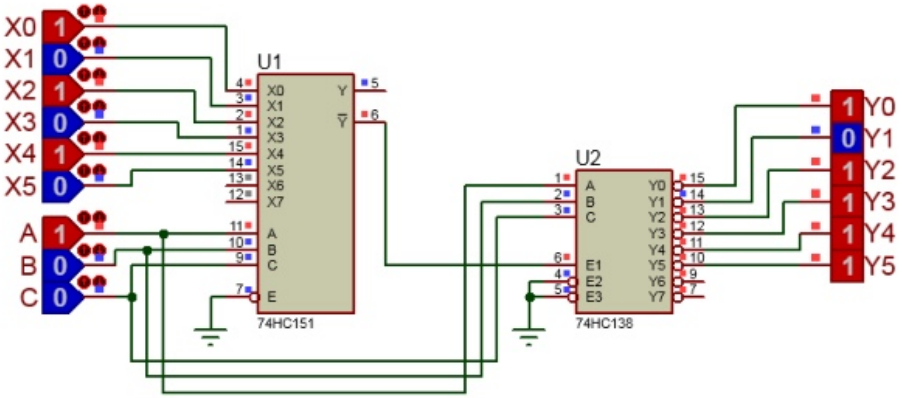

图 7- 3 6 路分时传送系统

四、思考题

1、设计的 6 路分时传送系统，若将数据选择器 74HC151 的输出$\overline{Y}$改为 Y，则译码器 74HC138 的串行输入端应该如果改接，才能保持系统功能不变。

2、用数据选择器 74HC151 和译码器 74HC138，设计一个 3 位二进制数等值比较器。

实验三　组合逻辑电路实验（三）

一、实验目的

1、掌握加法器、比较器逻辑功能及使用方法；

2、掌握利用加法器、比较器进行电路设计的方法；

3、进一步掌握组合逻辑电路的设计和测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC283、74HC85、74HC00、74HC04、74HC08、74HC10、74HC32、74HC86。

三、实验内容

1、用一片 4 位加法器 74HC283 和门电路设计一个 8421 码加法和的调整电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

两个 8421 码相加的结果为输入端 S4、S3、S2、S1、S0，调整后的结果为输出端 F4、F3、F2、F1、F0。如图 7- 1 所示，在使用 Proteus 仿真时，当输入的 8421 码加法和为 10010 时，是伪码，需要加 6 调整，因此输出为 11000。

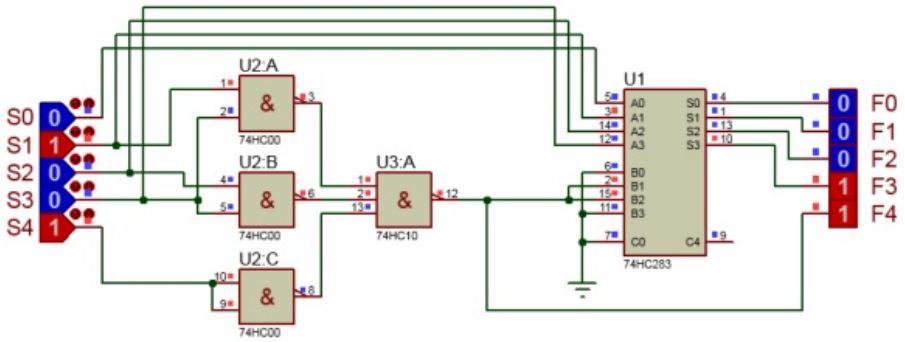

图 7- 1 8421 码加法和调整电路

2、用一片数据选择器 74HC151、一片 4 位加法器 74HC283 和门电路设计一个 2421 码加法和调整电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

两个 2421 码相加的结果为输入端 E、D、C、B、A，调整后的结果为输出端 F4、F3、F2、F1、F0。如图 7- 2 所示，在使用 Proteus 仿真时，当输入的 2421 码加法和为 00101 时，是伪码且没有进位，需要加 6 调整，因此输出为 01011。

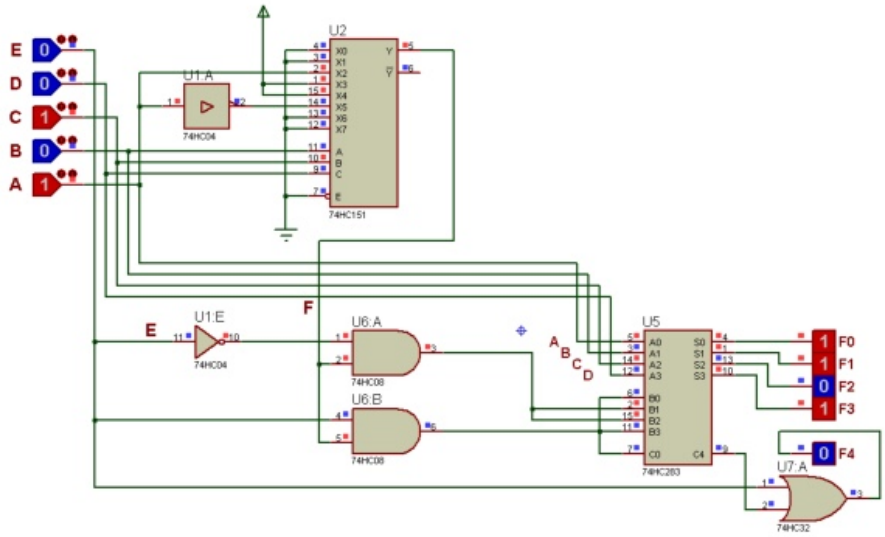

图 7- 2 2421 码加法和调整电路

3、用一片加法器 74HC283 和异或门 74HC86 组成一个 4 位二进制数加法/减法器，当控制端 C 为 0 时做加法运算，C 为 1 时做减法运算。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

两个数据输入端分别为 A3、A2、A1、A0 和 B3、B2、B1、B0，控制端为 C，运算的结果为 F3、F2、F1、F0。如图 7- 3 所示，在使用 Proteus 仿真时，当控制端 C 时做减法，即 1110-1100，因此运算结果为 0010，S4 为 1 表示完成减法运算。

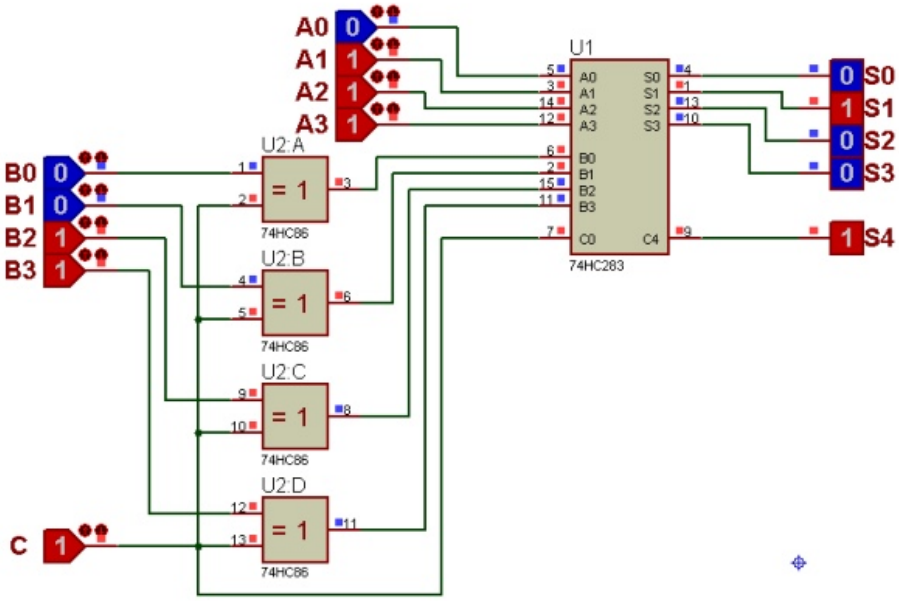

图 7- 3 加/减法器电路

4、用一片加法器 74HC283 和一片四位比较器 74HC85 设计一个 8421BCD 码转换为 5421 码的电路。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

数据输入端为 B8、B4、B2、B1，转换后的结果输出为 F3、F2、F1、F0。如图 7- 4 所示，在使用 Proteus 仿真时，当输入的 8421 码为 1001 时，转换后的 5421 码为 1100。

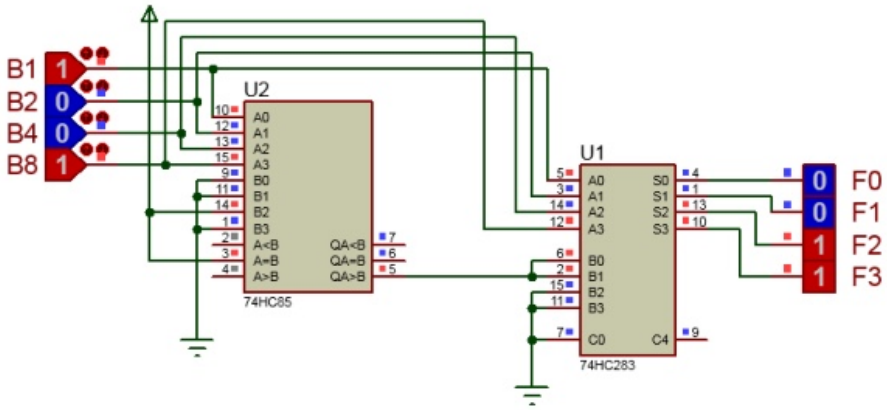

图 7- 4 8421BCD 码转 5421 码电路

四、思考题

1、用一片 4 位加法器 74HC283 设计一个 8421BCD 码转换为余 3 码的电路。

2、利用 8421 码加法和调整电路，设计一个 8421 码加法器电路。

实验四　组合逻辑电路实验（四）

一、实验目的

1、了解组合逻辑电路的设计方法，学会用 VHDL 语言进行简单的逻辑电路设计；

2、熟悉在 Quartus 软件中使用硬件描述语言进行电路设计，并能进行功能和时序仿真；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言设计一个三变量奇偶校验电路，当输入的 3 个变量中有偶数个 1 时电路输出为 1，否则输出为 0。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板中验证其功能。

三变量奇偶校验电路的参考源代码如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。当输入信号 C、B、A 出现偶数个 1 时，输出信号 F 为 1；其他情况输出为 0。由于采用时序仿真，因此输出信号的变化延迟输入信号约 10ns。

```
1  --     3变量奇偶校验电路
2  ------------------------------------------------
3   LIBRARY IEEE;
4   USE IEEE.STD_LOGIC_1164.ALL;
5   USE IEEE.STD_LOGIC_ARITH.ALL;
6   USE IEEE.STD_LOGIC_UNSIGNED.ALL;
7   ------------------------------------------------
8  ENTITY JOJY_3 IS
9     PORT(
10          A,B,C        :IN std_logic;
11          F            :OUT std_logic
12        );
13 END  JOJY_3 ;
14  ------------------------------------------------
15 ARCHITECTURE abc OF JOJY_3 IS
16 signal  INTEMP :std_logic_VECTOR(2 DOWNTO 0);
17 BEGIN
18     INTEMP<=C&B&A;
19     PROCESS(INTEMP)
20     BEGIN
21        CASE INTEMP IS
22        WHEN"000"=>F<='1';
23        WHEN"001"=>F<='0';
24        WHEN"010"=>F<='0';
25        WHEN"011"=>F<='1';
26        WHEN"100"=>F<='0';
27        WHEN"101"=>F<='1';
28        WHEN"110"=>F<='1';
29        WHEN"111"=>F<='0';
30        WHEN OTHERS=>F<='0';
31        END CASE;
32     END PROCESS;
33 end abc;
```

图 7- 1 三变量奇偶校验电路参考源代码

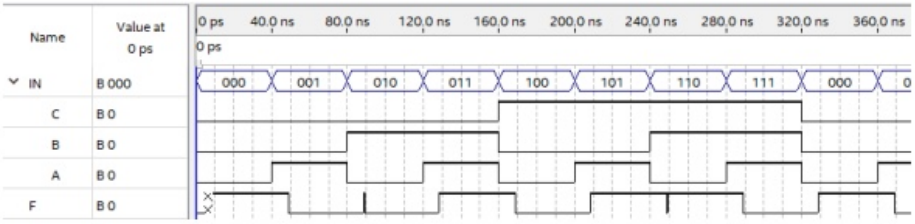

图 7- 2 三变量奇偶校验电路时序仿真波形图

2、用 VHDL 语言设计一个可控的代码转换电路，完成 4 位二进制码和格雷码的互相转

换。当控制端 CON 为 0 时，完成二进制码转换为格雷码；当控制端 CON 为 1 时，完成格雷码转换为二进制码。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板中验证其功能。

可控的代码转换电路的参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。当控制端 CON 为 1 时，实现二进制码转换为格雷码，当输入为二进制码 0010 时，输出为格雷码 0011；当控制端 CON 为 0 时，实现格雷码转换为二进制码，当输入为格雷码 0101 时，输出为二进制码 0111。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*****************************************
ENTITY  BINARY_TO_GRAY  IS
   PORT( A3,A2,A1,A0,CON    :IN std_logic;
         F3,F2,F1,F0     :OUT std_logic
       );
END  BINARY_TO_GRAY ;
--*****************************************
ARCHITECTURE abc OF BINARY_TO_GRAY IS
BEGIN
   PROCESS(A3,A2,A1,A0,CON)
   BEGIN
   IF(CON='0')THEN
   F3<=A3;
   F2<=A3 XOR A2;
   F1<=A2 XOR A1;
   F0<=A1 XOR A0;
   ELSE
   F3<=A3;
   F2<=A3 XOR A2;
   F1<=A3 XOR A2 XOR A1;
   F0<=A3 XOR A2 XOR A1 XOR A0;
   END IF;
   END PROCESS;
end abc;
```

图 7- 3 可控的代码转换电路参考源代码

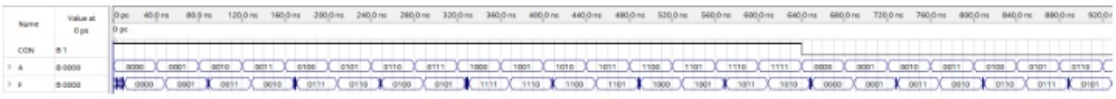

图 7- 4 可控的代码转换电路仿真波形图

四、思考题

1、FPGA 开发板中，芯片没有用到的引脚和双功能引脚应如何处理。

组合逻辑电路实验（五）

一、实验目的

1、熟悉在 Quartus 软件中使用硬件描述语言进行电路设计，并能进行功能和时序仿真；

2、掌握编码器的编程设计方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言设计一个 2421 码的普通编码器。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，用 8 个拨动开关表示编码器的其中 8 个输入端，用 4 个 LED 灯来表示编码器的 4 个输出端，验证其功能。

2421 码的普通编码器的参考源代码如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。当输入信号为 1111111011 时，表示第 2 个按键按下，此时输出为 0010，表示 2421 码中的 2。

```
--     2421码编码器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY ENCODER_2421 IS
   PORT(
         IN10        :IN std_logic_vector( 9 downto 0);
         F         :OUT std_logic_vector( 3  downto 0)
       );
END  ENCODER_2421 ;
--*********************************************
ARCHITECTURE abc OF ENCODER_2421 IS
BEGIN
   PROCESS(IN10)
   BEGIN
      CASE IN10 IS
      WHEN"1111111110"=>F<="0000";
      WHEN"1111111101"=>F<="0001";
      WHEN"1111111011"=>F<="0010";
      WHEN"1111110111"=>F<="0011";
      WHEN"1111101111"=>F<="0100";
      WHEN"1111011111"=>F<="1011";
      WHEN"1110111111"=>F<="1100";
      WHEN"1101111111"=>F<="1101";
      WHEN"1011111111"=>F<="1110";
      WHEN"0111111111"=>F<="1111";
      WHEN OTHERS=>F<="ZZZZ";
      END CASE;
   END PROCESS;
end abc;
```

图 7- 1 2421 码的普通编码器参考源代码

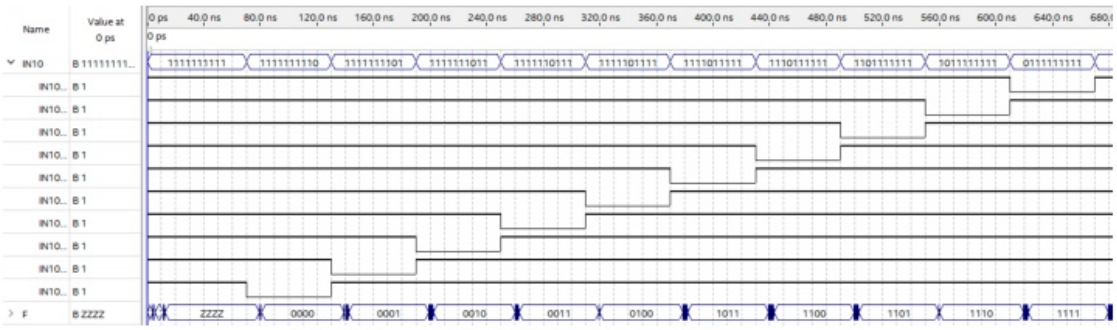

图 7- 2 2421 码的普通编码器仿真波形图

2、用 VHDL 语言设计一个 2421 码的优先编码器。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，用 8 个拨动开关表示编码器的其中 8 个输入端，用 4 个 LED 灯来表示编码器的 4 个输出端，验证其功能。

2421 码的优先编码器的参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。当输入信号为 1011111111 时，表示第 8 个按键按下，此时输出为 1110，表示 2421 码中的 8；当输入信号为 1100111111 时，表示第 7 个和地 6 个按键同时按下，由于第 6 个按键优先级更高，因此输出为 1100，表示 2421 码中的 6。

```
--     2421优先编码器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY PRIORITY_ENCODER_2421 IS
   PORT(
            CLK        :IN std_logic;
          IN10         :IN std_logic_vector( 9 downto 0);
          F         :OUT std_logic_vector( 3  downto 0)
       );
END  PRIORITY_ENCODER_2421 ;
--*********************************************
ARCHITECTURE abc OF PRIORITY_ENCODER_2421 IS
BEGIN
   PROCESS(IN10,CLK)
   BEGIN
   IF CLK'EVENT AND CLK='1' THEN
      IF IN10(0)='0' THEN F<="0000";
      ELSIF IN10(1)='0' THEN F<="0001";
      ELSIF IN10(2)='0' THEN F<="0010";
      ELSIF IN10(3)='0' THEN F<="0011";
      ELSIF IN10(4)='0' THEN F<="0100";
      ELSIF IN10(5)='0' THEN F<="1011";
      ELSIF IN10(6)='0' THEN F<="1100";
      ELSIF IN10(7)='0' THEN F<="1101";
      ELSIF IN10(8)='0' THEN F<="1110";
      ELSIF IN10(9)='0' THEN F<="1111";
      ELSE F<="ZZZZ";
      END IF;
   END IF;

   END PROCESS;
end abc;
```

图 7- 3 2421 码的普通编码器参考源代码

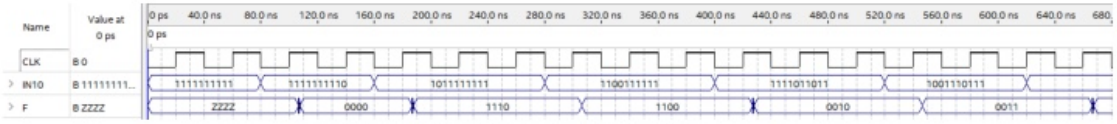

图 7- 4 2421 码的优先编码器仿真波形图

3、用 VHDL 语言设计一个七路抢答器，主持人和七个人各一个按钮。当主持人给有效信号后开始抢答，否则抢答无效。当抢答者按下抢答按钮后，判断第一个按下的抢答者，并显示其对应的二进制码。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板中验证其功能。

七路抢答器的参考源代码如图 7- 5 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。

```
--     7路抢答器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY QDQ_7 IS
   PORT(
         CLK,HOST     :IN std_logic;
         IN7          :IN std_logic_vector( 6 downto 0);
         F         :OUT std_logic_vector( 2  downto 0)
       );
END  QDQ_7 ;
--*********************************************
ARCHITECTURE abc OF QDQ_7 IS
signal LOCK :std_logic;
BEGIN
   PROCESS(CLK,HOST,IN7)
   BEGIN
   IF (HOST='0') THEN F<="111";LOCK<='0';
   elsif (HOST='1') THEN
      IF (LOCK='0') THEN
         IF CLK'EVENT AND CLK='1' THEN
         CASE IN7 IS
         WHEN"1111110"=>F<="000";LOCK<='1';
         WHEN"1111101"=>F<="001";LOCK<='1';
         WHEN"1111011"=>F<="010";LOCK<='1';
         WHEN"1110111"=>F<="011";LOCK<='1';
         WHEN"1101111"=>F<="100";LOCK<='1';
         WHEN"1011111"=>F<="101";LOCK<='1';
         WHEN"0111111"=>F<="110";LOCK<='1';
         WHEN OTHERS=>F<="111";
         END CASE;
         END IF;
      END IF;
   END IF;
   END PROCESS;
end abc;
```

图 7- 5 七路抢答器参考源代码

图 7- 6 七路抢答器仿真波形图

四、思考题

1、用 VHDL 语言设计一个 8421 码的普通编码器，编写程序并建立波形文件仿真验证。

实验六 组合逻辑电路实验（六）

一、实验目的

1、熟悉在 Quartus 软件中使用原理图进行电路设计和仿真；

2、掌握译码器、数据选择器的编程设计方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言分别设计一个 8 选 1 数据选择器和一个 3 线-8 线译码器。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后分别生成两个元件。

利用生成的 8 选 1 数据选择器和 3 线-8 线译码器元件组成一个 6 路信号分时传送系统，两个元件连接相同的数据输入端，观察该系统的输入和输出波形关系，最后将电路下载到 FPGA 开发板中验证其功能。

8 选 1 数据选择器的参考源代码如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。由于没有添加采样信号 CLK，所以输出存在冒险。

```
--8选1数据选择器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY MUX8  IS
   PORT( S      :IN std_logic;
         A      :IN std_logic_vector(2 downto 0);
         D      :IN std_logic_vector(7 downto 0);
         Y,YN   :OUT std_logic

       );
END  MUX8 ;
--*********************************************
ARCHITECTURE abc OF MUX8 IS
BEGIN
   PROCESS(S,A,D)
   BEGIN
   IF (S='0') THEN
      CASE A IS
      WHEN"000"=>Y<=D(0);YN<=NOT D(0);
      WHEN"001"=>Y<=D(1);YN<=NOT D(1);
      WHEN"010"=>Y<=D(2);YN<=NOT D(2);
      WHEN"011"=>Y<=D(3);YN<=NOT D(3);
      WHEN"100"=>Y<=D(4);YN<=NOT D(4);
      WHEN"101"=>Y<=D(5);YN<=NOT D(5);
      WHEN"110"=>Y<=D(6);YN<=NOT D(6);
      WHEN"111"=>Y<=D(7);YN<=NOT D(7);
      WHEN OTHERS=>NULL;
      END CASE;
   ELSE Y<='0';YN<='1';
   END IF;
   END PROCESS;
end abc;
```

图 7- 1 8 选 1 数据选择器参考源代码

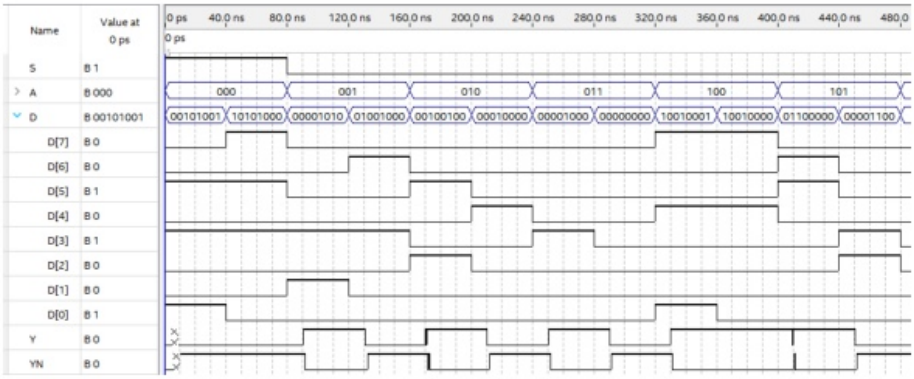

图 7- 2 8 选 1 数据选择器仿真波形图

3 线-8 线译码器的参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。

```
1  --3-8译码器
2  __*************************************************
3   LIBRARY IEEE;
4   USE IEEE.STD_LOGIC_1164.ALL;
5   USE IEEE.STD_LOGIC_ARITH.ALL;
6   USE IEEE.STD_LOGIC_UNSIGNED.ALL;
7   __*************************************************
8  ENTITY DECODER_3to8 IS
9  PORT(E1,NE2,NE3    :  IN STD_LOGIC;
10         CBA     :  IN STD_LOGIC_VECTOR(2 DOWNTO 0);
11     Y           :  OUT STD_LOGIC_VECTOR(7 DOWNTO 0)
12     );
13 END DECODER_3to8;
14  __*************************
15  ARCHITECTURE rtl OF DECODER_3to8 IS
16 BEGIN
17 PROCESS(CBA,E1,NE2,NE3)
18 BEGIN
19 IF(E1='1' AND NE2='0' AND NE3='0') THEN
20     CASE CBA IS
21     WHEN "000"=>Y<="11111110";
22     WHEN "001"=>Y<="11111101";
23     WHEN "010"=>Y<="11111011";
24     WHEN "011"=>Y<="11110111";
25     WHEN "100"=>Y<="11101111";
26     WHEN "101"=>Y<="11011111";
27     WHEN "110"=>Y<="10111111";
28     WHEN "111"=>Y<="01111111";
29     WHEN OTHERS=>Y<="XXXXXXXX";
30     END CASE;
31 ELSE
32     Y<="11111111";
33 END IF;
34 END PROCESS;
35 END rtl;
```

图 7- 3 3 线-8 线译码器参考源代码

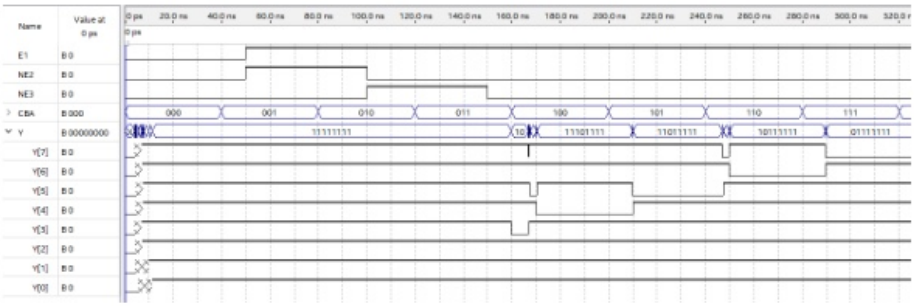

图 7- 4 3 线-8 线译码器仿真波形图

将生成元件的数据选择器和译码器放入新建的原理图文件中，并添加输入输出引脚和地端后连接电路，连接完成的 6 路信号分时传送系统电路如图 7- 5 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。当地址为 010 时，输出端 Y2 随输入信号 D2 的改变而变化，其他输出端均保持 1 不变。

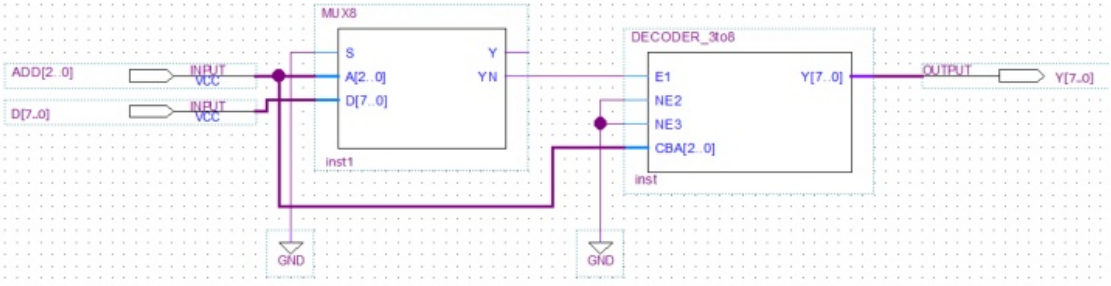

图 7- 5 6 路信号分时传送系统电路图

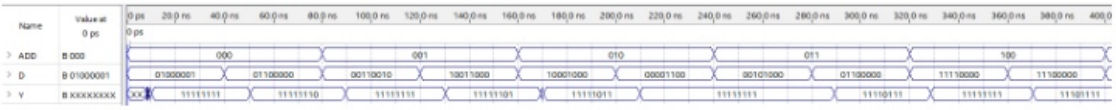

图 7- 6 6 路信号分时传送系统仿真波形图

2、用 VHDL 语言设计一个 2421 码的译码器。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

2421 码译码器的参考源代码如图 7- 7 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 8 所示。

```
--     2421译码器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY   DECODER_2421   is
   PORT( CLK:in std_logic;
         I  :in std_logic_vector( 3 downto 0);
         F  :OUT std_logic_vector( 9 downto 0)
         );
 END  DECODER_2421 ;
--*********************************************
ARCHITECTURE abc OF DECODER_2421    IS
BEGIN
PROCESS(I)
BEGIN
IF CLK'event and CLK='1' then
   CASE I IS
   WHEN "0000"=>F<="1111111110";
   WHEN "0001"=>F<="1111111101";
   WHEN "0010"=>F<="1111111011";
   WHEN "0011"=>F<="1111110111";
   WHEN "0100"=>F<="1111101111";
   WHEN "1011"=>F<="1111011111";
   WHEN "1100"=>F<="1110111111";
   WHEN "1101"=>F<="1101111111";
   WHEN "1110"=>F<="1011111111";
   WHEN "1111"=>F<="0111111111";
   WHEN OTHERS=>F<="1111111111";
   END CASE;
end IF;
END PROCESS;
end abc;
```

图 7- 7 2421 码译码器参考源代码

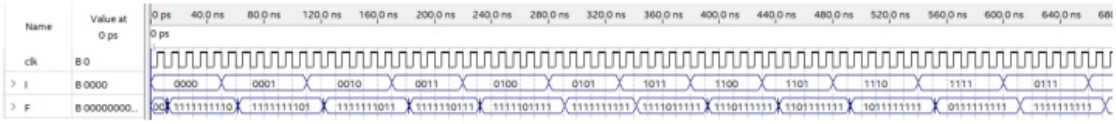

图 7- 8 2421 码译码器仿真波形图

3、用 VHDL 语言设计一个 8421 码共阴极七段数码管的显示译码器。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

显示译码器的参考源代码如图 7- 9 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 10 所示。当输入为 8421 码的 0-9 中的任一个数值，输出为对应的共阴七段码，如果不是 8421 码，则输出为“0000000”，共阴极数码管将不显示。

```
--          共阴极 7段译码器
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY   BCD_7SEG IS
PORT ( bcd_led :  IN STD_LOGIC_VECTOR(3 DOWNTO 0);
ledseg : OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
END BCD_7SEG;
ARCHITECTURE behavior OF BCD_7SEG IS
BEGIN
PROCESS(bcd_led)
BEGIN
CASE bcd_led IS
WHEN "0000"=>ledseg<="0111111" ;--0
WHEN "0001"=>ledseg<="0000110" ;--1
WHEN "0010"=>ledseg<="1011011" ;--2
WHEN "0011"=>ledseg<="1001111" ;--3
WHEN "0100"=>ledseg<="1100110" ;--4
WHEN "0101"=>ledseg<="1101101" ;--5
WHEN "0110"=>ledseg<="1111101" ;--6
WHEN "0111"=>ledseg<="0000111" ;--7
WHEN "1000"=>ledseg<="1111111" ;--8
WHEN "1001"=>ledseg<="1101111" ;--9
WHEN OTHERS=>ledseg<="0000000" ;
END CASE;
END PROCESS;
END behavior;
```

图 7- 9 显示译码器参考源代码

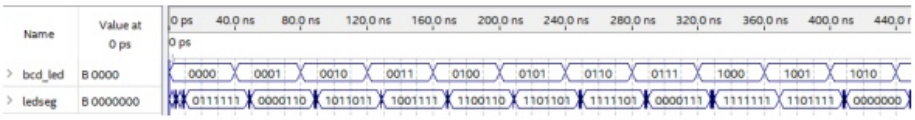

图 7- 10 显示译码器仿真波形图

四、思考题

1、用 VHDL 语言设计一个 4-16 译码器，编写程序并建立波形文件仿真验证。

2、采用动态显示的方法，用 VHDL 语言设计一个能同时显示 2 位数码管的显示译码器。

实验七 组合逻辑电路实验（七）

一、实验目的

1、熟悉在 Quartus 软件中使用原理图进行电路设计和仿真；

2、掌握加法器的编程设计方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言设计一个 2421 码加法和的调整电路。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

2421 码加法和调整电路的参考源代码如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。输入为两个 2421 码相加的结果，输出为根据调整规则进行调整后的结果，当输入为 10111 时，出现伪码且有进位，需要减 0110，因此输出为 10001。

```
--2421码加法和调整电路
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY v2421T IS
   PORT(                    --输入输出引脚说明
          Z    :IN std_logic_vector(4 downto 0);
         ZT  :OUT std_logic_vector(4 downto 0)
        );
END v2421T  ;
--*********************************************
ARCHITECTURE abc OF v2421T  IS
BEGIN
  PROCESS(Z)
   BEGIN
     IF Z(3 DOWNTO 0)="0101"  OR
        Z(3 DOWNTO 0)="0110"  OR
        Z(3 DOWNTO 0)="0111"  OR
        Z(3 DOWNTO 0)="1000"  OR
        Z(3 DOWNTO 0)="1001"  OR
        Z(3 DOWNTO 0)="1010"  THEN
       IF Z(4)='1'  THEN
         ZT<=Z-"0110";
       ELSE ZT<=Z+"0110";
       END IF;
     ELSE ZT<=Z;
     END IF;
  END PROCESS;
end abc;
```

图 7- 1 2421 码加法和调整电路参考源代码

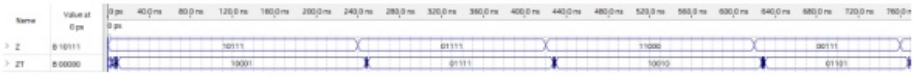

图 7- 2 显示译码器仿真波形图

2、用 VHDL 语言设计两个 8421 码相加的加法电路。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

两个 8421 码相加的加法电路参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。输入两个 2421 码及低位的进位信号，输出加法和及向高位的进位信号，当输入的两个 8421 码为 0100 和 0101、进位信号为 1 时，输出的加法和为 0000，向高位进位信号为 1。

```
--*****8421加法器*************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--****************************************
ENTITY   V8421ADD is
PORT( cin   :in std_logic;
      op1,op2  :in unsigned( 3 downto 0);
      co:OUT std_logic;
      sum:OUT std_logic_vector( 3 downto 0)
);
END V8421ADD   ;
ARCHITECTURE abc OF V8421ADD  IS
signal binadd,res :unsigned( 4 downto 0);
BEGIN
binadd<=('0' & op1)+op2+cin;
process(binadd)
   begin
   if binadd >9 then res<=binadd+6;
   else res<=binadd;
   end if;
end process;
sum<=res( 3 downto 0)+0;
co<=res(4) ;
end abc;
```

图 7- 3 两个 8421 码相加的加法电路参考源代码

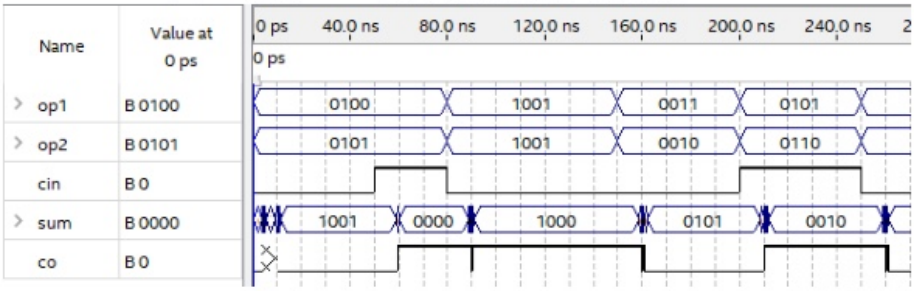

图 7- 4 两个 8421 码相加的加法电路仿真波形图

3、用 VHDL 语言设计可控的 4 位加法/减法电路。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

可控的 4 位加法/减法电路参考源代码如图 7- 5 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。输入信号为控制信号 CON、两个二进制数 A、B，输出信号为结果 S 及符号位 FUHAO，当 CON 为 0，做加法，输入的二进制数为 0001 和 0101，输出的加法和为 00110，不是负数，符号位为 0；当 CON 为 1，做减法，输入的二进制数为 0110 和 1010，输出减法的差为 00100，是负数，符号位为 1。

```
--        4bit 的加/减法器
--*****************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--****************************************
entity ADD_SUB_4BIT is
port(A,B :in std_logic_vector(3 downto 0);
       CON:IN std_logic;
    FUHAO : out std_logic;
       S  :out std_logic_vector(4 downto 0));
end ADD_SUB_4BIT;
architecture rtl of ADD_SUB_4BIT is
signal TEMP2:std_logic_vector(4 downto 0);
begin
P1:process(CON,A,B)
begin
   IF(CON='0') then
   TEMP2<=('0'&A)+B;
   ELSE
       TEMP2<=('0'& A)+NOT B+1;
   end if;
end process P1;
P2:PROCESS(TEMP2,CON)
BEGIN
   IF CON='1' THEN
      IF TEMP2(4)='0'THEN
         S<= '0' & NOT TEMP2(3 downto 0) +1;
         FUHAO<='1';
      ELSE S<='0' &  TEMP2(3 downto 0);
         FUHAO<='0';
      END IF;
   ELSE S<=TEMP2;FUHAO<='0';
   END IF;
END PROCESS P2;
end rtl;
```

图 7- 5 可控的 4 位加法/减法电路参考源代码

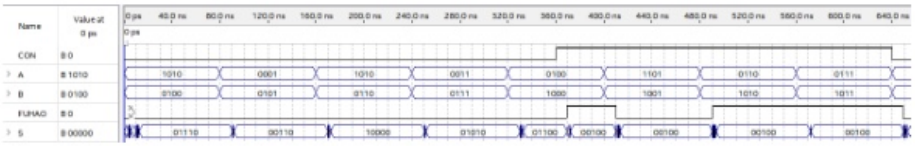

图 7- 6 可控的 4 位加法/减法电路仿真波形图

四、思考题

1、用 VHDL 语言设计两个 2421 码相加的加法电路，编写程序并建立波形文件仿真验证。

实验八　时序逻辑电路实验（一）

一、实验目的

1、掌握 JK 触发器、D 触发器逻辑功能及使用方法；

2、掌握利用 JK 触发器、D 触发器设计计数器的方法；

3、掌握时序逻辑电路的设计和测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC76、74HC74、74HC08、74HC11、74HC32、74HC86。

三、实验内容

1、用双 JK 触发器 74HC76 设计一个同步四进制加法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Q1、Q0 为计数器的输出。如图 7- 1 所示，在使用 Proteus 仿真时，输出 Q1Q0 状态为 00-01-10-11-00，实现四进制加法。

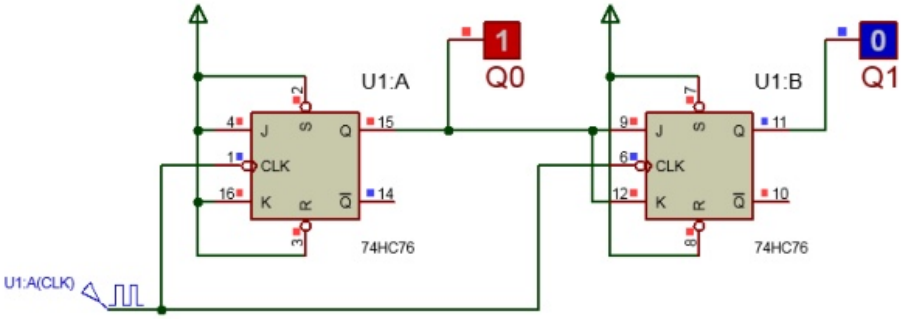

图 7- 1 同步四进制加法计数器电路

2、用双 JK 触发器 74HC76 和门电路设计一个 8421BCD 码的同步十进制加法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Q3、Q2、Q1、Q0 为计数器的输出。如图 7- 2 所示，在使用 Proteus 仿真时，输出 Q3Q2Q1Q0 状态 0000-1001，实现十进制加法。

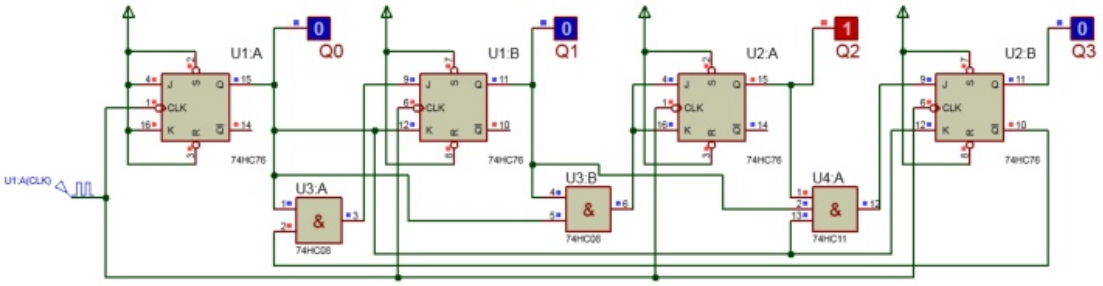

图 7- 2 8421BCD 码同步十进制加法计数器电路

3、用双 D 触发器 74HC74 和门电路设计一个同步模 3 的减法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号， Q1、Q0 为计数器的输出。如图 7- 3 所示，在使用 Proteus

仿真时，输出端 Q1Q0 状态为 11-10-00-111，实现模 3 的减法计数。

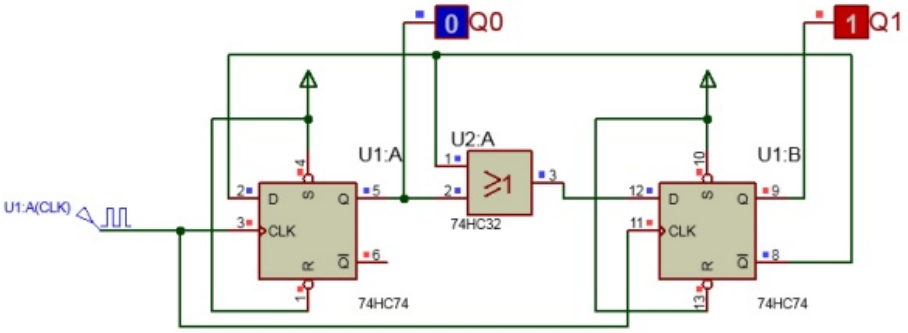

图 7- 3 模 3 的减法计数器电路

4、用双 D 触发器 74HC74 和门电路设计一个同步五进制加法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Q2、 Q1、Q0 为计数器的输出。如图 7- 4 所示，在使用 Proteus 仿真时，输出端 Q2Q1Q0 状态为 000-001-010-011-100-000，实现五进制加法计数。

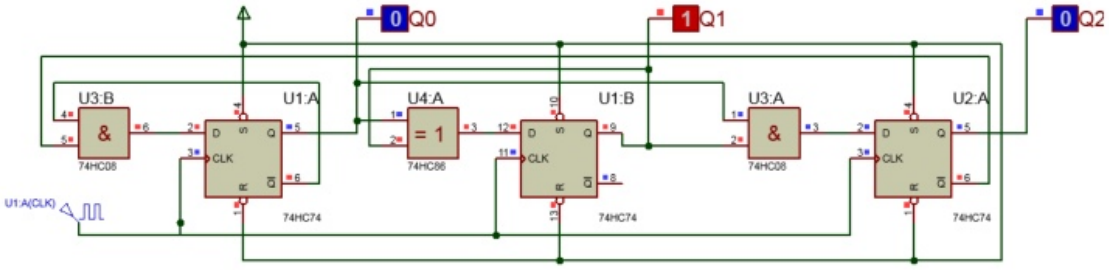

图 7- 4 同步五进制加法计数器电路

四、思考题

1、74HC74 和 74HC76 的时钟 CLK 在什么边沿有效？

2、用双 JK 触发器 74HC76 设计一个单脉冲发生器。

实验九　时序逻辑电路实验（二）

一、实验目的

1、掌握集成移位寄存器逻辑功能及使用方法；

2、掌握利用集成移位寄存器设计计数器和信号发生器的方法；

3、进一步掌握时序逻辑电路的设计和测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC194、74HC138、74HC00、74HC04、74HC08、74HC20、74HC32。

三、实验内容

1、用移位寄存器 74HC194 和门电路设计 M=6 的具有自启动功能的移位型计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Q2、Q1、Q0 为计数器的输出。如图 7- 1 所示，在使用 Proteus 仿真时，输出 Q2Q1Q0 状态为 000-001-011-111-110-100-000，实现模为 6 的计数器。

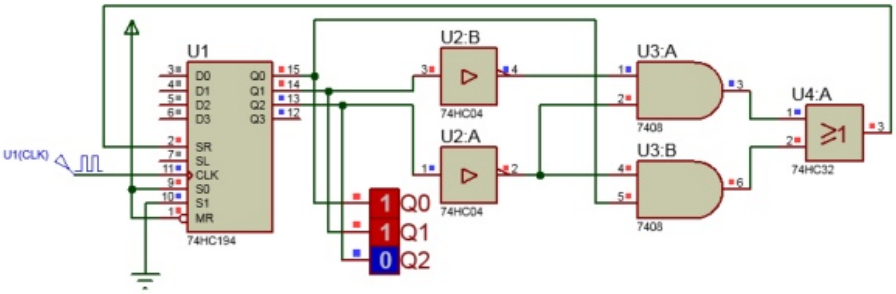

图 7- 1 模 6 移位型计数器电路

2、用移位寄存器 74HC194 和门电路一个能产生 100111 序列的序列信号发生器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Q3、Q2、Q1、Q0 为计数器的输出。如图 7- 2 所示，在使用 Proteus 仿真时，从计数器任一输出端观察均能产生 100111 的序列。

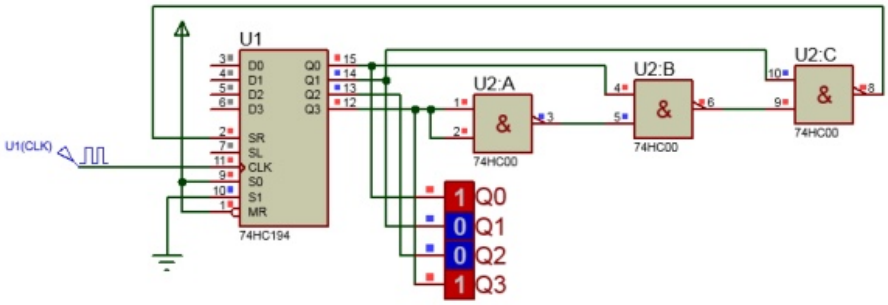

图 7- 2 序列信号发生器电路

3、用移位寄存器 74HC194、3-8 线译码器 74HC138 和门电路设计一个能同时产生两组序列的双序列信号发生器，两个序列分别为 110101 和 010110。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，Y1、Y2 为两个序列的输出。如图 7- 3 所示，在使用 Proteus

仿真时，Y1 输出序列 110101，Y2 输出序列 010110。

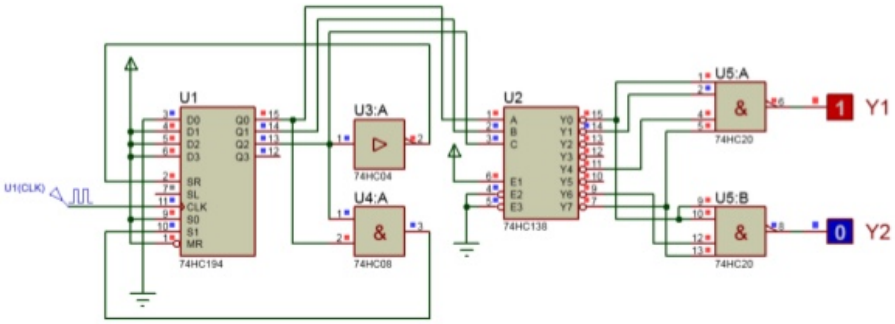

图 7- 3 双序列信号发生器电路

四、思考题

1、使用移位寄存器 74HC194 和 8 选 1 数据选择器 74HC151 设计模 12 计数器。

实验十　时序逻辑电路实验（三）

一、实验目的

1、掌握集成计数器逻辑功能及使用方法；

2、掌握利用集成计数器设计任意进制计数器和信号发生器的方法；

3、进一步掌握时序逻辑电路的设计和测试方法。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、面包板；

5、集成电路 74HC161、74HC151、74HC00、74HC04、74HC32。

三、实验内容

1、用四位同步二进制计数器 74HC161 和门电路设计一个起始状态为 0 的 12 进制加法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，74HC161 的 Q3、Q2、Q1、Q0 为计数器的输出。如图 7- 1 所示，在使用 Proteus 仿真时，使用 74HC161 的异步清零端，将 Q3 和 Q2 通过与非门后接入清零端$\overline{MR}$，计数器输出为 1100 时清零，实现模为 6 的计数器。

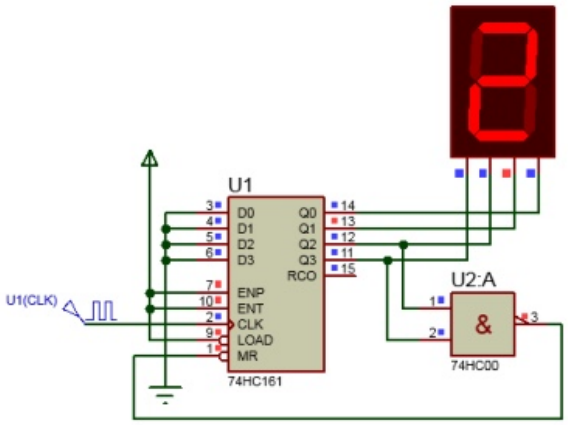

图 7- 1 12 进制加法计数器电路

2、用两片四位同步二进制计数器 74HC161 和门电路设计一个起始状态为 0 的 8421 码 60 进制同步加法计数器。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号，两片 74HC161 的 Q3、Q2、Q1、Q0 分别表示计数器的十位和各位。如图 7- 2 所示，在使用 Proteus 仿真时，使用 U1：74HC161 的同步预置端，将 U1 的 Q3 和 Q0 通过 U3A 与非门后接入 U1 预置端$\overline{LOAD}$，计数器输出 1001 时在下一个时钟下降沿 74HC161 预置零，实现模为 10 的计数器；使用 U21：74HC161 的同步预置端，U2 的将 Q2 和 Q0 通过 U3B 与非门后，再 U3A、U3B 通过或门接入 U2 预置端$\overline{LOAD}$，计数器输出 01011001 时在下一个时钟下降沿 74HC161 预置零，实现模为 59 的计数器。

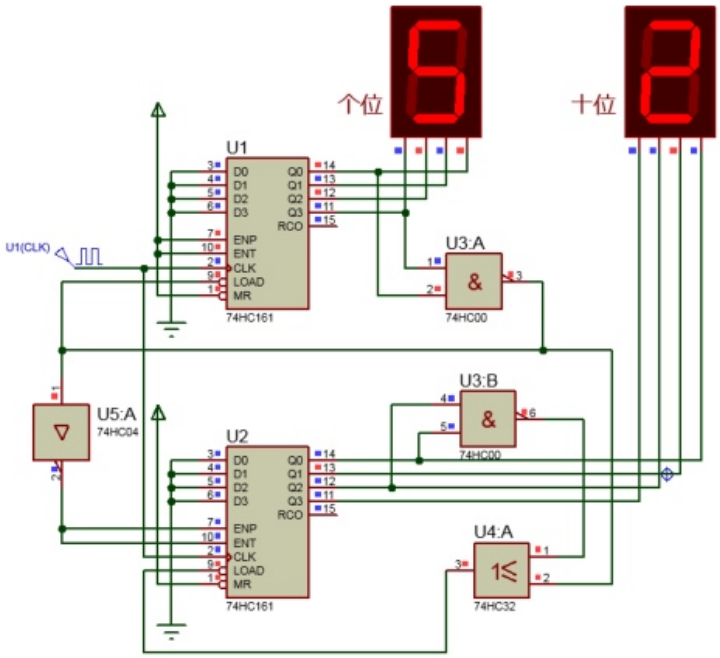

图 7- 2 12 进制加法计数器电路

3、用四位同步二进制计数器 74HC161、8 选 1 数据选择器 74HC151 及门电路设计一个计数器型序列信号发生器，产生序列信号 100011010101。要求先在 Proteus 中画出原理图，验证其功能；再用面包板连接电路并测试。

CLK 接 1Hz 的时钟信号， 74HC161 设置为 12 进制计数器，74HC151 地址输入端连接 74HC161 的 Q3Q1Q0。如图 7- 3 所示，在使用 Proteus 仿真时，74HC161 采用同步预置 0 实现 0000-1011 的模为 12 的计数器，74HC151 实现每一个计数值输出序列信号的一位，实现 100011010101 的产生。

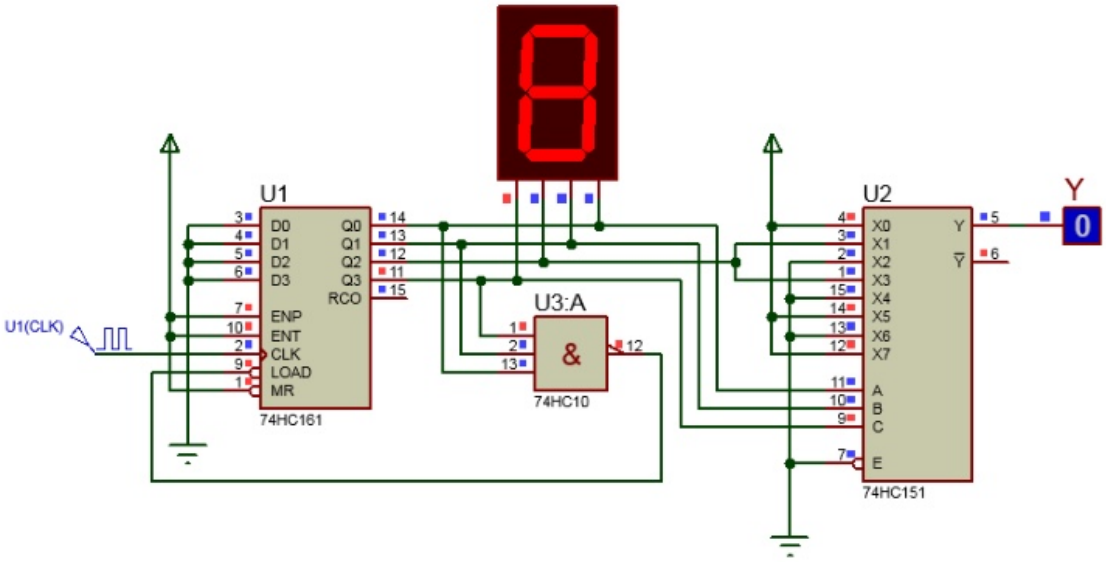

图 7- 3 计数器型序列信号发生器电路

四、思考题

1、用两片四位同步二进制计数器 74HC161 和门电路设计一个起始状态为 0 的二进制码 60 进制同步加法计数器。

实验十一　时序逻辑电路实验（四）

一、实验目的

1、熟悉在 Quartus 软件中使用 VHDL 或原理图进行电路设计和仿真；

2、掌握触发器的逻辑功能和使用方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用双 JK 触发器设计一个单脉冲发生器电路。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

单脉冲发生器电路图如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。CLK 接频率较高的时钟脉冲信号，SI 接按键，当按键按下一次后输出 Q1 只产生一个脉冲，与按键按下的时间长短无关，还能消除按键的抖动。

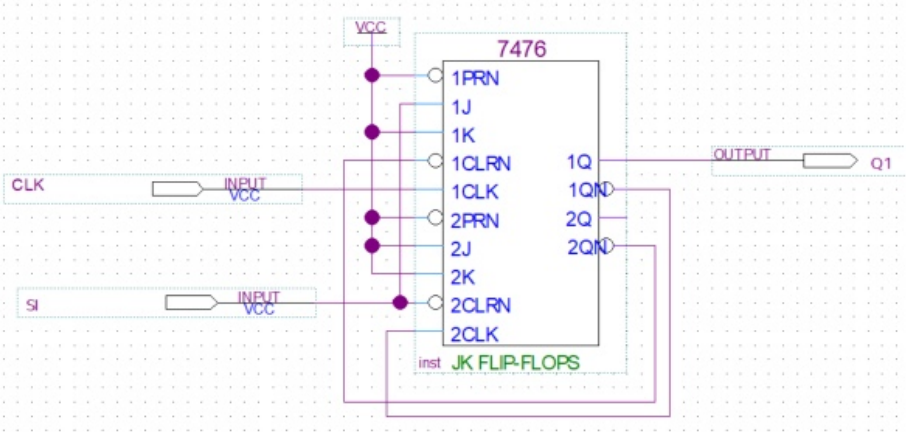

图 7- 1　单脉冲发生器电路图

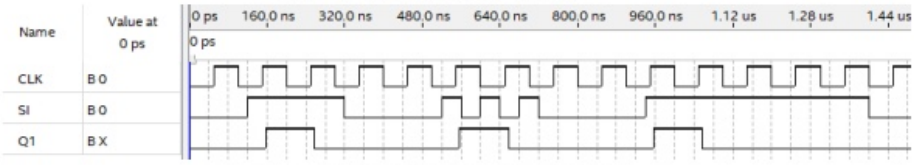

图 7- 2　单脉冲发生器电路仿真波形图

2、用双 D 触发器设计 2/4 分频电路。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

2/4 分频电路电路图如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。CLK 为时钟信号，Q1 频率是 CLK 的 1/2，Q1 频率是 CLK 的 1/4。

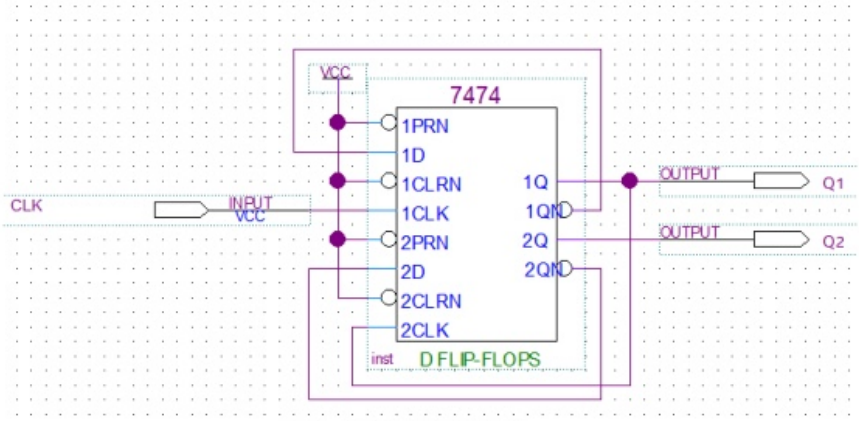

图 7- 3 2/4 分频电路图

图 7- 4 2/4 分频电路仿真波形图

3、用 VHDL 语言设计一个投币自动售饮料机的逻辑电路。投币自动售饮料机只允许投 0.5 元的硬币，只要累积投币到 2.5 元就自动出一罐饮料。使用 Quartus 软件进行程序编写、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

投币自动售饮料机的逻辑电路的参考源代码如图 7- 5 所示。CLK 时钟信号，A 为投币信号，1 表示投入一个 0.5 元硬币，0 表示没有投入；Q 表示计数状态，Y 表示出饮料信号，1 表示出，0 表示不出。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。当 A 为 1 的时间累计到 5 个时钟信号后，Y 输出 1，表示可以出饮料。

```
--投币机 5进制
--*****************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*************************************
ENTITY TBJ_5 IS
PORT (CLK,A     : IN STD_LOGIC;
          Y     : OUT STD_LOGIC;
          Q: OUT STD_LOGIC_VECTOR(2 DOWNTO 0)
          );
END TBJ_5;
--***************
ARCHITECTURE dataflow_1 OF TBJ_5 IS
SIGNAL QQ   :STD_LOGIC_VECTOR(2 DOWNTO 0);
BEGIN
PROCESS(CLK,QQ,A)
BEGIN
IF CLK'EVENT AND CLK='1' THEN
   case A&QQ IS
   WHEN "0000"=>QQ<="000";
   WHEN "1000"=>QQ<="001";
   WHEN "0001"=>QQ<="001";
   WHEN "1001"=>QQ<="010";
   WHEN "0010"=>QQ<="010";
   WHEN "1010"=>QQ<="011";
   WHEN "0011"=>QQ<="011";
   WHEN "1011"=>QQ<="100";
   WHEN "0100"=>QQ<="100";
   WHEN "1100"=>QQ<="101";
   WHEN "0101"=>QQ<="000";
   WHEN "1101"=>QQ<="001";
   WHEN OTHERS=>QQ<="000";
   END CASE;
   IF QQ="101" THEN Y<='1';
   ELSE Y<='0';
   END IF;
END IF;
END PROCESS;
 Q<=QQ;
END dataflow_1;
```

图 7- 5 投币自动售饮料机参考源代码

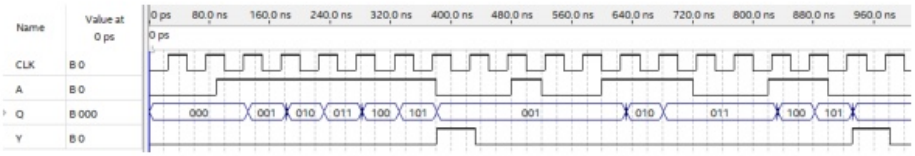

图 7- 6 投币自动售饮料机仿真波形图

四、思考题

1、用 VHDL 语言设计单脉冲发生器，编写程序并建立波形文件仿真验证。

2、用 VHDL 语言设计 2/4 分频器，编写程序并建立波形文件仿真验证。

实验十二　时序逻辑电路实验（五）

一、实验目的

1、熟悉在 Quartus 软件中使用 VHDL 或原理图进行电路设计和仿真；

2、掌握移位寄存器的逻辑功能和使用方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言设计一个序列信号检测器，当序列信号检测器连续收到一组 1101010 后,输出为 1;否则输出为 0，允许输入的序列重叠。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

序列信号检测器的参考源代码如图 7- 1 所示。CLK 时钟信号，RES 为复位信号，X 为序列信号输入端；Q 表示计数状态，Y 表示是否检测到正确的序列信号，检测到输出 1，否则输出 0。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图图 7- 2 所示。当 X 连续输入序列 1101010 后，Q 为 111，下一个时钟信号时 Y 输出 1。

```
--        序列信号检测 1101010
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY SEQ_DETECT_1101010      IS
   PORT( CLK,X,RES    :IN std_logic;
         Y         :OUT std_logic;
         Q  :OUT std_logic_vector( 2 downto 0)
       );
END SEQ_DETECT_1101010  ;
--*********************************************
ARCHITECTURE abc OF  SEQ_DETECT_1101010   IS
signal Q1    :std_logic_vector( 2 downto 0);
signal Q2    :std_logic_vector( 3 downto 0);
BEGIN
PROCESS(CLK,RES)
BEGIN
   Q2<=x&Q1;
   IF CLK'EVENT AND CLK='1' THEN
      IF RES='1' THEN Q1<="001";
      ELSE
      CASE Q2 IS
      WHEN "0001"=>Q1<="001";
      WHEN "1001"=>Q1<="010";
      WHEN "0010"=>Q1<="001";
      WHEN "1010"=>Q1<="011";
      WHEN "0011"=>Q1<="100";
      WHEN "1011"=>Q1<="011";
      WHEN "0100"=>Q1<="001";
      WHEN "1100"=>Q1<="101";
      WHEN "0101"=>Q1<="110";
      WHEN "1101"=>Q1<="011";
      WHEN "0110"=>Q1<="001";
      WHEN "1110"=>Q1<="111";
      WHEN "0111"=>Q1<="001";
      WHEN "1111"=>Q1<="011";
      WHEN OTHERS=>Q1<="001";
      END CASE;
      END IF;
   Y<=NOT X AND Q1(2) AND Q1(1) AND Q1(0);
   END IF;
END PROCESS;
Q<=Q1;
end abc;
```

图 7- 1 序列信号检测器参考源代码

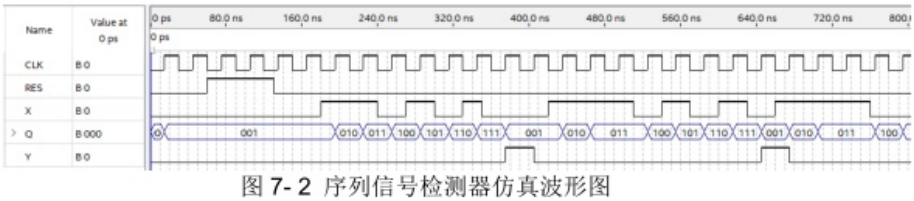

图 7- 2 序列信号检测器仿真波形图

2、用 VHDL 语言设计一个序列信号发生器，产生的序列为 100111。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

序列信号发生器的参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。Q 为模六的计数器，其任一位输出均为序列 100111。

```
--     序列信号发生器 100111
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY SEQ_GENERATOR_100111   is
PORT( CLK    :in std_logic;
      Y      :OUT std_logic;
      Q:OUT std_logic_vector( 3 downto 0)
      );
END SEQ_GENERATOR_100111   ;
--*********************************************
ARCHITECTURE abc OF  SEQ_GENERATOR_100111 IS
signal Q1 :std_logic_vector( 3 downto 0);
BEGIN
process(CLK,Q1)
begin
   if CLK'event and CLK='1' then
   case Q1 is
   when "1001"=>Q1<="0011";
   when "0011"=>Q1<="0111";
   when "0111"=>Q1<="1111";
   when "1111"=>Q1<="1110";
   when "1110"=>Q1<="1100";
   when "1100"=>Q1<="1001";
   when others=>Q1<="1001";
   end case;
   end if;
end process;
Q<=Q1;y<=Q1(3);
end abc;
```

图 7- 3 序列信号发生器参考源代码

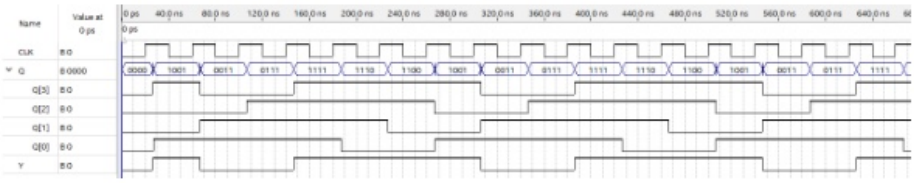

图 7- 4 序列信号发生器仿真波形图

3、用 VHDL 语言设计一个 6 位串并转换电路。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

6 位串并转换电路的参考源代码如图 7- 5 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。CLK 时钟信号， DATA 为串行信号输入端；Q 表示计数状态，DOUT 并行输出端。每 6 个时钟信号，DOUT 输出一组 6 位的串行转并行的信号。

```
--    串转并电路 6位
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY  CTOB_6    IS
   PORT( CLK,DATA    :IN std_logic;
         DOUT  :OUT std_logic_vector( 5 downto 0);
         Q  :OUT std_logic_vector( 2 downto 0)
       );
END CTOB_6  ;
--*********************************************
ARCHITECTURE abc OF CTOB_6    IS
signal Q1   :std_logic_vector( 2 downto 0);
signal Q2   :std_logic_vector( 5 downto 0);
signal C6   :std_logic;
BEGIN
P1:PROCESS(CLK,Q1)
BEGIN
   IF CLK'EVENT AND CLK='1' THEN
      IF Q1<"101" THEN  Q1<=Q1+1;C6<='0';
      ELSE Q1<=( OTHERS =>'0'); C6<='1';
      END IF;
   END IF;
   Q<=Q1;
END PROCESS;
P2:PROCESS(CLK,Q2)
BEGIN
   IF CLK'EVENT AND CLK='1' THEN
   Q2(0)<=DATA;Q2(5 DOWNTO 1)<=Q2(4 DOWNTO 0);
   END IF;
END PROCESS;
P3:PROCESS(C6,Q2)
BEGIN
   IF C6'EVENT AND C6='1' THEN DOUT<=Q2;
   END IF;
END PROCESS;
end abc;
```

图 7- 5 6 位串并转换电路参考源代码

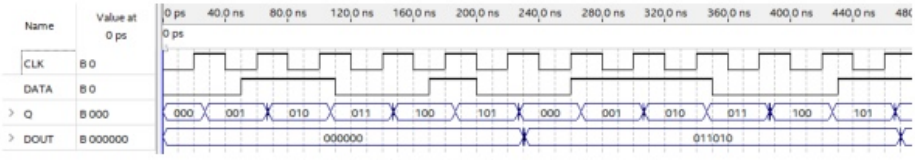

图 7- 6 6 位串并转换电路仿真波形图

4、用 VHDL 语言设计一个 6 路输出的脉冲分配器。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

6 路脉冲分配器的参考源代码如图 7- 7 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 8 所示。CLK 时钟信号， RESET 为复位信号；Q 表示计数状态，Y 为 6 路脉冲输出端。每 6 个时钟信号，Y 的 6 个输出端轮流输出 1 个时钟周期的低电平。

```
--    六路脉冲分配
--*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--*********************************************
ENTITY MCFP    IS
   PORT(  CLK,RESET  :IN std_logic;
          Q  :OUT std_logic_vector( 2 downto 0);
          Y  :OUT std_logic_vector( 5 downto 0)
       );
END MCFP  ;
--*********************************************
ARCHITECTURE abc OF  MCFP   IS
signal Q1 :std_logic_vector( 2 downto 0):="000";
BEGIN
PROCESS(CLK,Q1,RESET)
BEGIN
IF RESET='0' THEN Q1<="000";Y<="111111";
ELSIF CLK'EVENT AND CLK='1' THEN
   CASE Q1 IS
   WHEN "000"=>Q1<="001";Y<="111111";
   WHEN "001"=>Q1<="010";Y<="111110";
   WHEN "010"=>Q1<="011";Y<="111101";
   WHEN "011"=>Q1<="100";Y<="111011";
   WHEN "100"=>Q1<="101";Y<="110111";
   WHEN "101"=>Q1<="110";Y<="101111";
   WHEN "110"=>Q1<="000";Y<="011111";
   WHEN OTHERS=>Q1<="000";Y<="111111";
   END CASE;
END IF;
END PROCESS;
Q<= Q1;
end abc;
```

图 7- 7 6 路脉冲分配器参考源代码

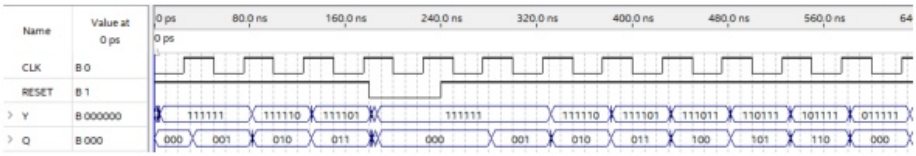

图 7- 8 6 路脉冲分配器仿真波形图

四、思考题

1、用 VHDL 语言设计一个可以自启动的 6 位 M 序列信号发生器，编写程序并建立波形文件仿真验证。

2、用设计的序列信号发生器和串并转换电路采用原理图，设计一个将 6 位序列信号转换为并行信号的电路。

实验十三　时序逻辑电路实验（六）

一、实验目的

1、熟悉在 Quartus 软件中使用 VHDL 或原理图进行电路设计和仿真；

2、掌握计数器的逻辑功能和使用方法；

3、熟悉 FPGA 开发板的使用。

二、实验仪器及元器件

1、双踪示波器；

2、万用表；

3、电路电子综合实验箱；

4、PC 及 Quartus 软件；

5、FPGA 开发板；

三、实验内容

1、用 VHDL 语言设计一个占空比为 40%、分频比为 10 的方波输出分频器。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

方波输出分频器的参考源代码如图 7- 1 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 2 所示。CLK 时钟信号；Q1_ OUT 表示计数状态，Q 为分频输出端。由波形图可以观察到 Q 的频率为 CLK 的 1/10，其占空比为 40%。

```
--          10分频器   占空比40%
__*********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
__*********************************************
ENTITY  FENPQ_10    IS
   PORT(  CLK  :IN std_logic;
          Q1_OUT:OUT std_logic_VECTOR(3 DOWNTO 0);
          Q     :OUT std_logic
        );
END FENPQ_10  ;
__*********************************************
ARCHITECTURE abc OF  FENPQ_10   IS
signal Q1 :INTEGER RANGE 0 TO 9;
BEGIN
PROCESS(CLK)
BEGIN
IF CLK'EVENT AND CLK='1' THEN
   IF Q1<=3 THEN Q<='1';Q1<=Q1+1;
   --改变Q1后的数值即可改变Q的占空比
   ELSIF Q1=9 THEN Q<='0';Q1<=0;
   --改变Q1后的数值即可Q的频率
   ELSE Q<='0';Q1<=Q1+1;
   END IF;
END IF;
END PROCESS;
Q1_OUT<=CONV_STD_LOGIC_VECTOR(Q1,4);
  end abc;
```

图 7- 1 分频器参考源代码

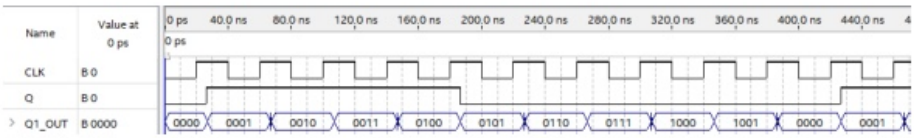

图 7- 2 分频器仿真波形图

2、用 VHDL 语言一个具有异步清零的 12 进制加法计数器。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

12 进制加法计数器的参考源代码如图 7- 3 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 4 所示。CLK 时钟信号，CLR 为异步清零入端；Q 表示计数状态。

```
--          异步清零的12进制计数器
--********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--********************************************
ENTITY COUNT_12 IS
   PORT( CLK,CLR  :IN std_logic;
         Q  :OUT std_logic_vector(3 downto 0)
       );
END COUNT_12  ;
--********************************************
ARCHITECTURE abc OF COUNT_12 IS
signal QQ :std_logic_vector(3 downto 0);
BEGIN
   PROCESS(CLK,CLR)
   BEGIN
   IF CLR='0' THEN QQ<="0000";
   ELSIF CLK'EVENT AND CLK='1' THEN
      IF QQ="1011" THEN QQ<="0000";
      ELSE QQ<=QQ+1;
      END IF;
   END IF;
   END PROCESS;
Q<=QQ;
end abc;
```

图 7- 3 12 进制加法计数器参考源代码

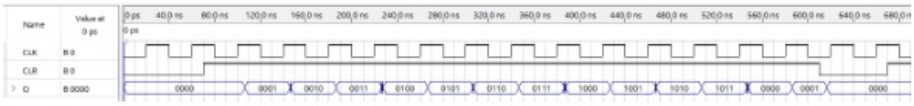

图 7- 4 12 进制加法计数器仿真波形图

3、用 VHDL 语言设计一个具有异步清零的 60 进制 BCD 码加法计数器。使用 Quartus 软件进行设计、编译，编译通过后进行仿真，最后将程序下载到 FPGA 开发板，验证其功能。

60 进制 BCD 码加法计数器的参考源代码如图 7- 5 所示。程序编译通过后，建立波形文件进行时序仿真，仿真波形如图 7- 6 所示。CLK 时钟信号，CLR 为异步清零入端；Q 表示计数状态。

```
--        异步清零 BCD60进制计数器
--********************************************
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
--********************************************
ENTITY  COUNT_60BCD    IS
   PORT( CLK,CLR  :IN std_logic;
         Q      :OUT std_logic_vector( 7 downto 0)
       );
END COUNT_60BCD  ;
--********************************************
ARCHITECTURE abc OF COUNT_60BCD    IS
signal QQ  :std_logic_vector( 7 downto 0);
BEGIN
PROCESS(CLK,CLR)
BEGIN
   IF CLR='0' THEN QQ<="00000000";
   ELSIF CLK'EVENT AND CLK='1' THEN
      IF QQ>="01011001" THEN QQ<="00000000";
      ELSIF QQ(3 DOWNTO 0)>="1001" THEN
      QQ(3 DOWNTO 0)<="0000";
      QQ(7 DOWNTO 4)<=QQ(7 DOWNTO 4)+1;
      ELSE QQ<=QQ+1;
      END IF;
   END IF;
END PROCESS;
Q<=QQ;
end abc;
```

图 7- 5 60 进制 BCD 码加法计数器参考源代码

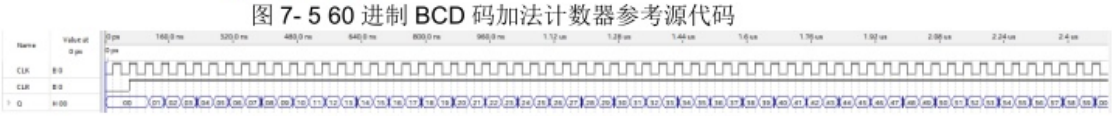

图 7- 6 60 进制 BCD 码加法计数器仿真波形图

四、思考题

1、用 VHDL 语言设计一个具有异步清零的 12 进制码减法计数器，编写程序并建立波形文件仿真验证。

2、用 VHDL 语言设计一个加/减法可控的 60 进制 BCD 码计数器，编写程序并建立波形文件仿真验证。