浙江大学实验报告

专业：电气工程及其自动化
姓名：潘谷雨
学号：3220102382
地点：紫金港东三 406

课程名称：电路与电子技术实验　　指导老师：张伟　　成绩：

实验名称：DE10-Lite 开发板步进电机脉冲分配器设计实验　　同组学生姓名：杨骐恺

一. 实验目的

1.了解FPGA工程的开发过程；
2.认识DE10-Lite开发板，学习Quartus-Prime集成开发软件的应用；
3.步进电机脉冲分配器设计与总程序设计。

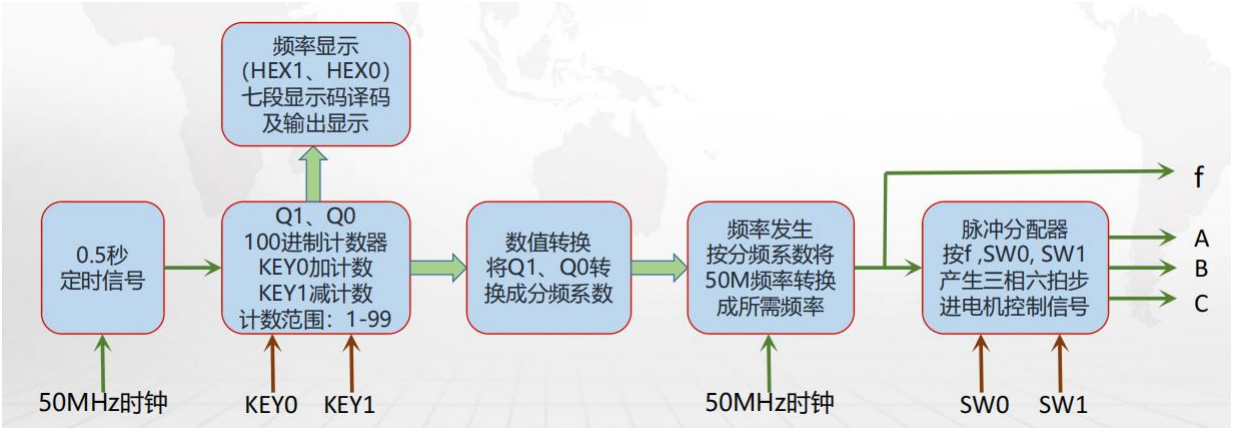
二. 实验仪器

MAX10M50DAF484C7G芯片，Quartus-Prime集成开发软件。

三. 实验原理

如果控制线路不停地按A→AB→B→BC→C→CA→A→…的顺序控制步进电机绕组的通断电，步进电机的转子便不停地顺时针转动。若通电顺序改为A→AC→C→CB→B→BA→A→…。同理，步进电机的转子将逆时针不停地转动。

在步进电机控制器FPGA程序结构如下图所示。



FPGA 程序要求：

1、采用 DE10-Lite FPGA 学习板;
2、两位数码管（HEX1、HEX0）显示脉冲频率，频率范围：1Hz -- 99Hz;
3、频率调节按钮（KEY1，KEY0）；
4、工作模式为三相六拍；
5、滑动开关 SW0 可控制 运行/停止 状态；
6、滑动开关 SW1 可控制 正转/反转 状态；
7、A、B、C 三相脉冲及频率 f 采用学习板上 GPIO（JP1）输出
A - GPIO_[1]; B - GPIO_[5]; C - GPIO_[9]; f - GPIO_[11]; GND - PIN12 。

四. 实验步骤与结果

（1）新建项目

启动Quartus软件，选择File > New Project Wizard…菜单命令，输入项目文件夹、项目名称和顶层实体名，新建空项目，器件选择10M50DAF484C7G芯片。

（2）编写设计文件

新建设计文件，类型选择VHDL File，另存为motor.vhd，步进电机脉冲分配器的VHDL源程序如图1.1至图1.7所示。

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity motor is
port(
    clk_50mHz:in std_logic;     -- 50MHz系统时钟输入
    key0_up,key1_down:in std_logic;-- 按键输入，key0_up为加键，key1_down为减键
    sw0_clr,sw1_x:in std_logic;-- 开关输入，sw0_clr用于清零，sw1_x用于控制逻辑
    a,b,c,f:out std_logic;      -- 步进电机控制信号输出
    hex0,hex1:out std_logic_vector(6 downto 0)-- 两位七段数码管显示频率
    );
end motor;

architecture behavior of motor is
     signal q1: std_logic_vector(3 downto 0):="0000";  -- 计数器1，用于十位
     signal q0: std_logic_vector(3 downto 0):="0001";  -- 计数器0，用于个位
     signal clk_2Hz: std_logic;                        -- 2Hz时钟信号，用于控制频率更新
     signal q2, q3, q4, q5: std_logic_vector(7 downto 0):="00000000";  -- 中间变量，用于计算频率
     signal k: integer range 0 to 25000000;            -- 频率计算变量
     signal freq: std_logic;                           -- 输出频率信号
     signal qa:std_logic:='1';    --初始A=1
     signal qb,qc:std_logic:='0'; --初始B=0,C=0
```

图1.1　输入输出与参数定义部分

```vhdl
    constant m:integer:=12500000;-- 用于生成2Hz时钟的计数常量
begin

-- 生成2Hz时钟信号的过程
process(clk_50mHz)
    variable cout:integer:=0;
    begin
       if rising_edge(clk_50mHz) then
          cout:=cout+1;
          if cout<=m then clk_2Hz<='0';
          elsif cout<m*2 then clk_2Hz<='1';
          else cout:=0;
          end if;
       end if;
end process;
```

图1.2　生成2Hz时钟部分

```vhdl
--100进制
process(clk_2Hz)
begin
   if rising_edge(clk_2Hz) then
      if key0_up='0' then                    --"+"
         if(q1="1001" and q0="1001") then --99->01
            q1<="0000";q0<="0001";
         elsif(q0="1001") then               --?9->?0
            q1<=q1+1;q0<="0000";
         else q0<=q0+1;                      -- +1
         end if;
      elsif key1_down='0' then               --"-"
         if(q1="0000" and q0="0001") then --01->99
            q1<="1001";q0<="1001";
         elsif(q0="0000") then               --?0->?9
            q1<=q1-1;q0<="1001";
         else q0<=q0-1;                      -- -1
         end if;
      end if;
   end if;
end process;
```

图1.3　100进制频率控制部分

```vhdl
--频率计算
q2(3 downto 0)<=q0;
q3(4 downto 1)<=q1;
q4(6 downto 3)<=q1;
q5<=q2+q3+q4;                          --q5=q1*10+q0
k<=25000000/conv_integer(q5); --分频
```

图1.4　分频参数计算部分

```vhdl
--生成频率信号freq的过程
process(clk_50mHz)
   variable cout:integer:=0;
   begin
      if rising_edge(clk_50mHz) then
         cout:=cout+1;
         if cout<=k then freq<='0';
         elsif cout<k*2 then freq<='1';
         else cout:=0;
         end if;
      end if;
end process;
```

图1.5　生成分频脉冲部分

```
84  --数码管显示处理，将二进制转换为七段码hex
85  process(q0)
86  begin
87     case q0 is
88           when "0000"=>hex0<="1000000"; --0
89           when "0001"=>hex0<="1111001"; --1
90           when "0010"=>hex0<="0100100"; --2
91           when "0011"=>hex0<="0110000"; --3
92           when "0100"=>hex0<="0011001"; --4
93           when "0101"=>hex0<="0010010"; --5
94           when "0110"=>hex0<="0000010"; --6
95           when "0111"=>hex0<="1111000"; --7
96           when "1000"=>hex0<="0000000"; --8
97           when "1001"=>hex0<="0010000"; --9
98           when others=>hex0<="1111111";
99     end case;
100 end process;
```

```
102 process(q1)
103 begin
104    case q1 is
105          when "0000"=>hex1<="1000000"; --0
106          when "0001"=>hex1<="1111001"; --1
107          when "0010"=>hex1<="0100100"; --2
108          when "0011"=>hex1<="0110000"; --3
109          when "0100"=>hex1<="0011001"; --4
110          when "0101"=>hex1<="0010010"; --5
111          when "0110"=>hex1<="0000010"; --6
112          when "0111"=>hex1<="1111000"; --7
113          when "1000"=>hex1<="0000000"; --8
114          when "1001"=>hex1<="0010000"; --9
115          when others=>hex1<="1111111";
116    end case;
117 end process;
```

图1.6　数码管显示部分

```
119 --步进电机驱动逻辑，基于频率信号进行状态翻转
120 process(freq)
121 begin
122    if rising_edge(freq) then
123    --sw1_x=1,qa<=!qb,qb<=!qc,qc<=!qa
124    --sw1_x=0,qa<=!qc,qb<=!qa,qc<=!qbS
125       qa<=not((sw1_x and qb) or ((not sw1_x) and qc));
126       qb<=not((sw1_x and qc) or ((not sw1_x) and qa));
127       qc<=not((sw1_x and qa) or ((not sw1_x) and qb));
128    end if;
129 end process;
130 -- 输出最终的步进电机控制信号，考虑清零开关
131 f<=freq;
132 a<=qa and sw0_clr;
133 b<=qb and sw0_clr;
134 c<=qc and sw0_clr;
135 end behavior;
```

图1.7　步进控制与输出部分

（3）项目编译

执行Processing>Start Compilation 菜单命令。

（4）管脚定义

①等到编译不报错，执行Assignments>Pin planner 菜单命令，进入引脚分配编辑器窗口，将PIN_P11作为50M时钟输入，两位数码管（HEX1、HEX0）显示脉冲频率，频率调节按钮按下KEY0增频，按下KEY1减频，滑动开关SW0控制运行/停止状态，滑动开关 SW1 控制正转/反转状态，三相脉冲A、B、C及频率f 分别在GPIO_[1]、GPIO_[5]、GPIO_[9]、GPIO_[11]输出。

a	Output	PIN_W10	3	B3_N0	PIN_W10	2.5 V
b	Output	PIN_W8	3	B3_N0	PIN_W8	2.5 V
c	Output	PIN_V5	3	B3_N0	PIN_V5	2.5 V
clk_50mHz	Input	PIN_P11	3	B3_N0	PIN_P11	2.5 V
f	Output	PIN_AA15	4	B4_N0	PIN_AA15	2.5 V
hex0[6]	Output	PIN_C17	7	B7_N0	PIN_C17	2.5 V
hex0[5]	Output	PIN_D17	7	B7_N0	PIN_D17	2.5 V
hex0[4]	Output	PIN_E16	7	B7_N0	PIN_E16	2.5 V
hex0[3]	Output	PIN_C16	7	B7_N0	PIN_C16	2.5 V
hex0[2]	Output	PIN_C15	7	B7_N0	PIN_C15	2.5 V
hex0[1]	Output	PIN_E15	7	B7_N0	PIN_E15	2.5 V
hex0[0]	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V
hex1[6]	Output	PIN_B17	7	B7_N0	PIN_B17	2.5 V
hex1[5]	Output	PIN_A18	7	B7_N0	PIN_A18	2.5 V
hex1[4]	Output	PIN_A17	7	B7_N0	PIN_A17	2.5 V
hex1[3]	Output	PIN_B16	7	B7_N0	PIN_B16	2.5 V
hex1[2]	Output	PIN_E18	6	B6_N0	PIN_E18	2.5 V
hex1[1]	Output	PIN_D18	6	B6_N0	PIN_D18	2.5 V
hex1[0]	Output	PIN_C18	7	B7_N0	PIN_C18	2.5 V
key0_up	Input	PIN_B8	7	B7_N0	PIN_B8	2.5 V
key1_down	Input	PIN_A7	7	B7_N0	PIN_A7	2.5 V
sw0_clr	Input	PIN_C10	7	B7_N0	PIN_C10	2.5 V
sw1_x	Input	PIN_C11	7	B7_N0	PIN_C11	2.5 V

图1.8　引脚分配编辑器窗口

②进行未用引脚的设置，执行Assignments>Device菜单命令，点击Device and Pin Options…按键，选中Unused Pins 条目，在Reserve all unused pins 下拉列表中选择As input tri-stated设置项。

（5）重新编译

重新对项目进行完整编译后生成motor.sof文件，经程序下载后在DE10-Lite开发板上验证程序。结果如下图所示。

①长按Key0，测试增频功能。

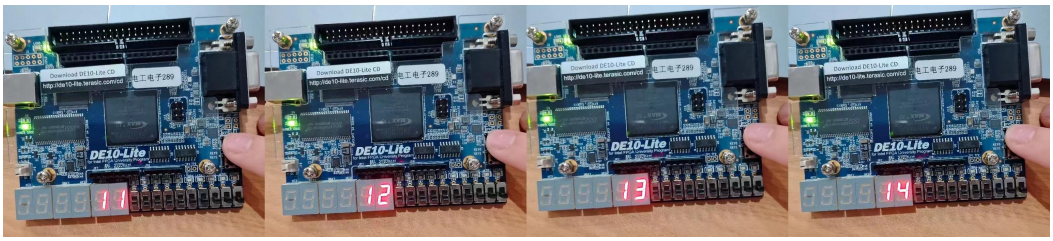


图1.9 测试增频功能

②长按Key1，测试减频功能。

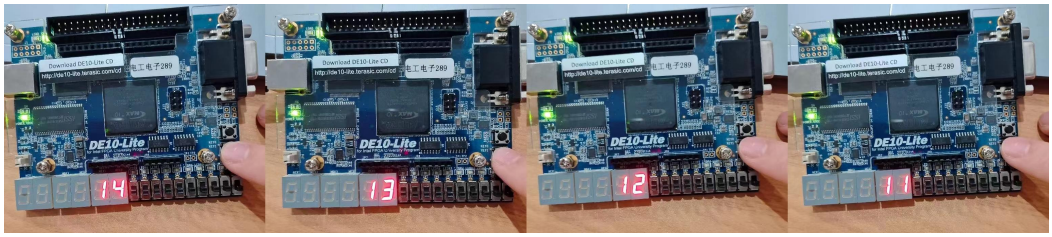


图1.10 测试减频功能

③设置不同频率，CH1检测输出分频脉冲f的波形，CH2检测输出A脉冲的波形。SW0键输入0时，能观察到分频脉冲f，A脉冲始终为0，上拨SW0键输入1时能观察到A脉冲波形，此后保持SW0键输入1的状态。

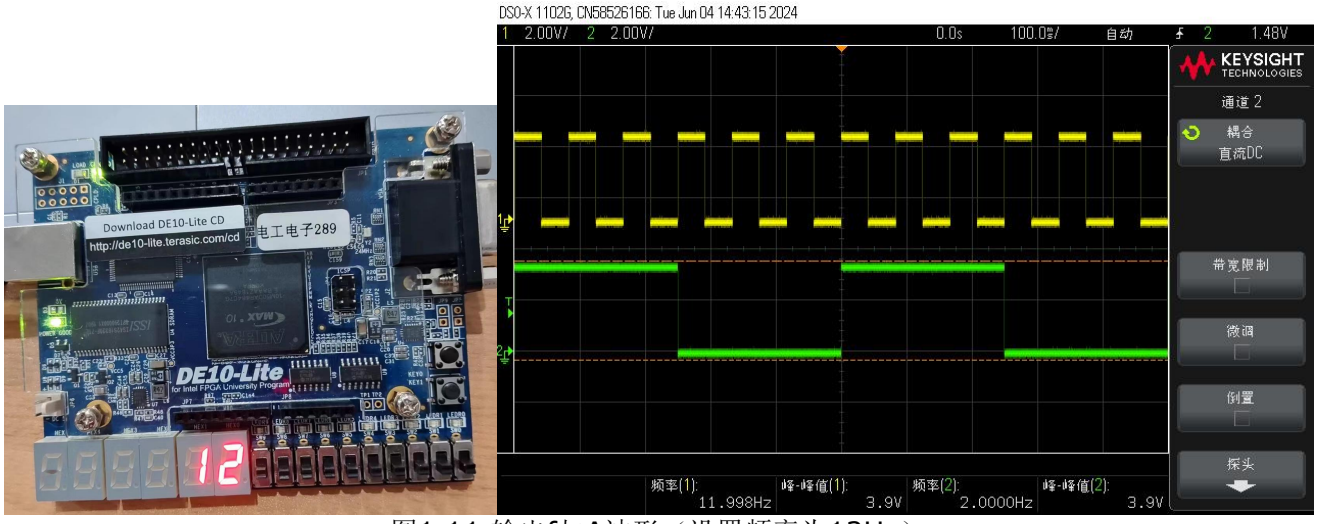


图1.11 输出f与A波形（设置频率为12Hz）

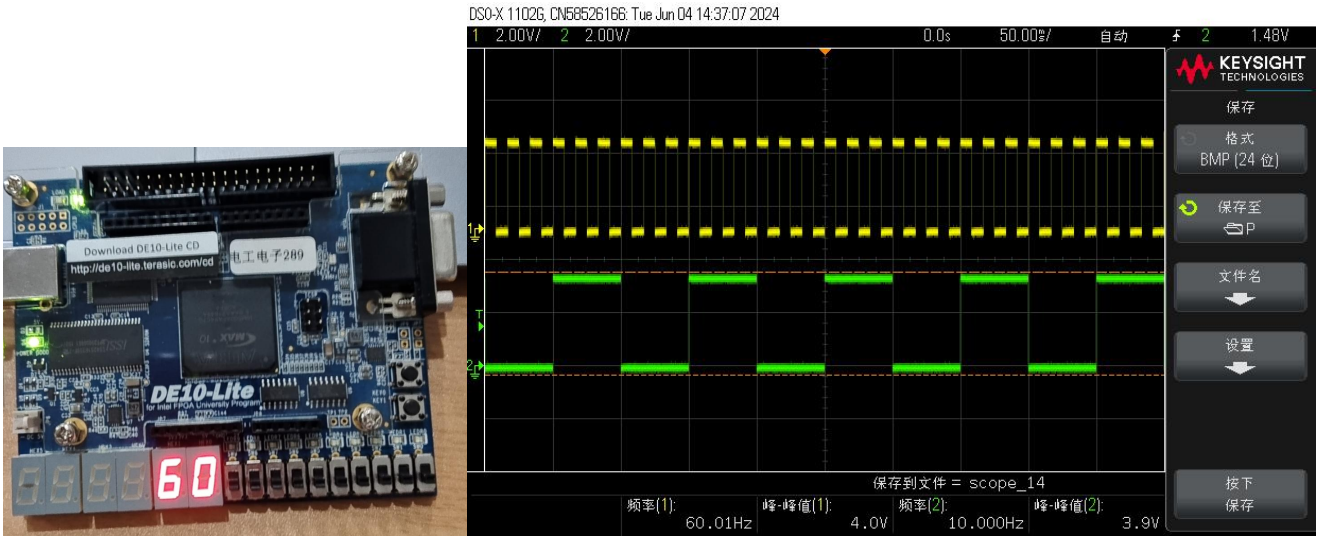


图1.12 输出f与A波形（设置频率为60Hz）

当分频脉冲f频率为11.998Hz时，A脉冲频率为2.0000Hz；分频脉冲f频率为60.01Hz时，A脉冲频率为10.000Hz，为上升沿触发，是f的六分频。

④设置频率为60Hz，SW1键输入0，电机反转，通电顺序为A→AC→C→CB→B→BA→A→…。

1.CH1检测输出A脉冲的波形，CH2检测输出B脉冲的波形，频率均为10.000Hz，A滞后B约120°。

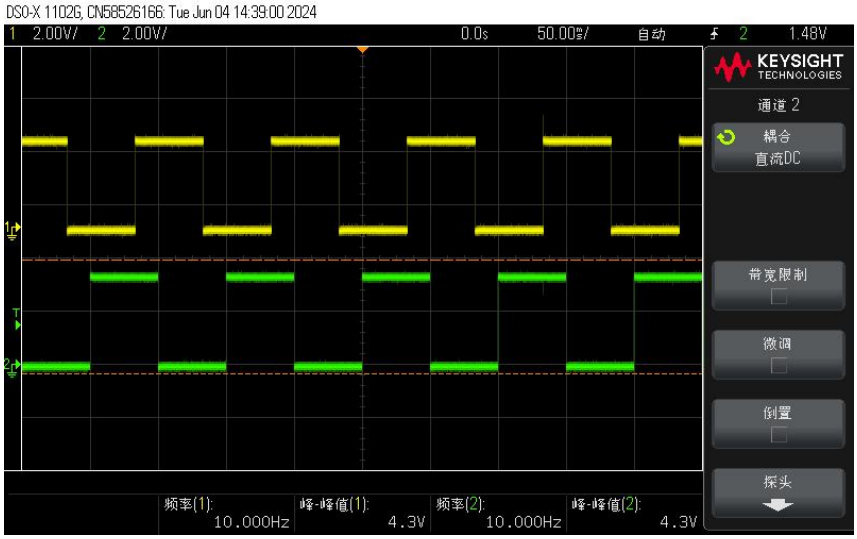


图1.12 输出A与B波形

2.CH1检测输出B脉冲的波形，CH2检测输出C脉冲的波形，频率均为10.000Hz，B滞后C约120°。

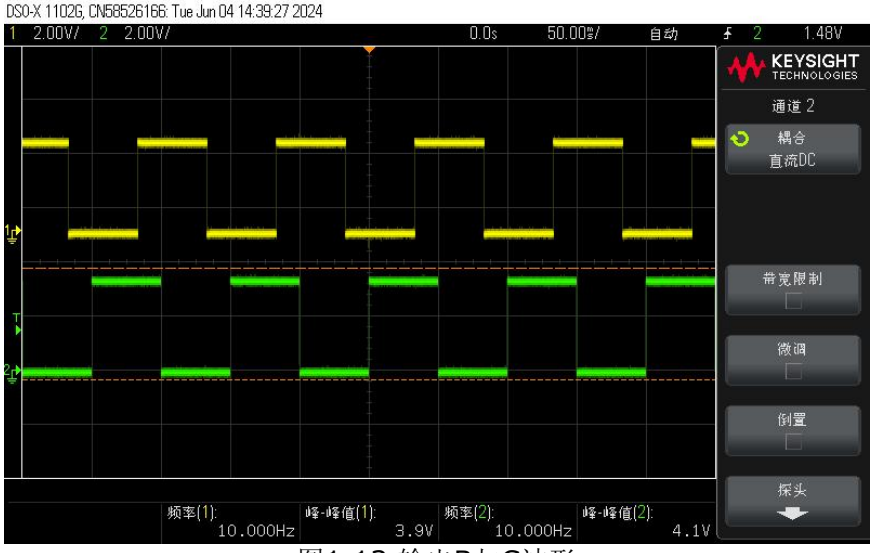


图1.13 输出B与C波形

3.CH1检测输出C脉冲的波形，CH2检测输出A脉冲的波形，频率均为10.000Hz，C滞后A约120°。

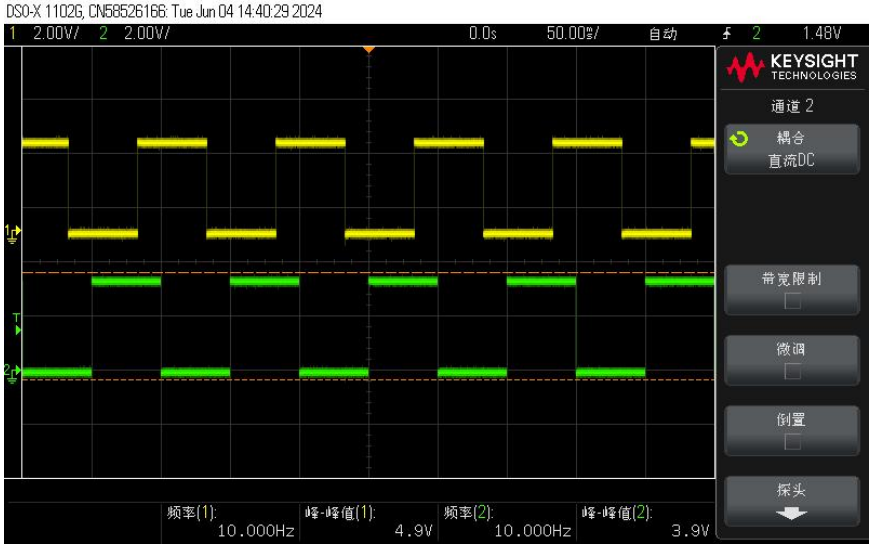


图1.14 输出C与A波形

⑤设置频率为60Hz，SW1键输入1，电机正转，通电顺序为A→AB→B→BC→C→CA→A→…。

1.CH1检测输出A脉冲的波形，CH2检测输出B脉冲的波形，频率均为10.000Hz，A超前B约120°。

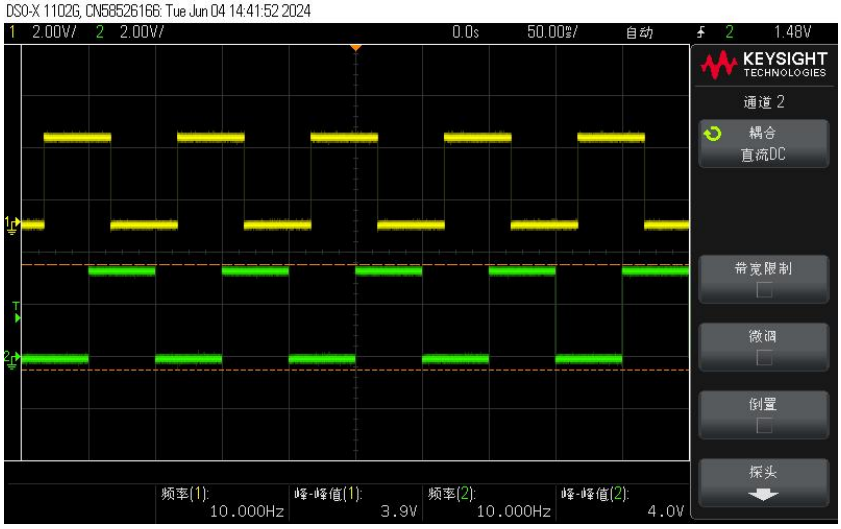


图1.15 输出A与B波形

2.CH1检测输出B脉冲的波形，CH2检测输出C脉冲的波形，频率均为10.000Hz，B超前C约120°。

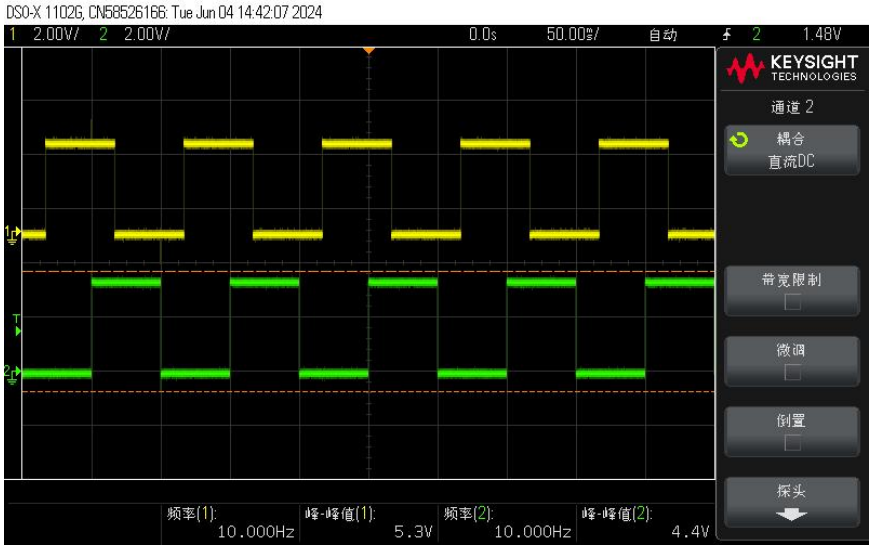


图1.16 输出B与C波形

3.CH1检测输出C脉冲的波形，CH2检测输出A脉冲的波形，频率均为10.000Hz，C超前A约120°。

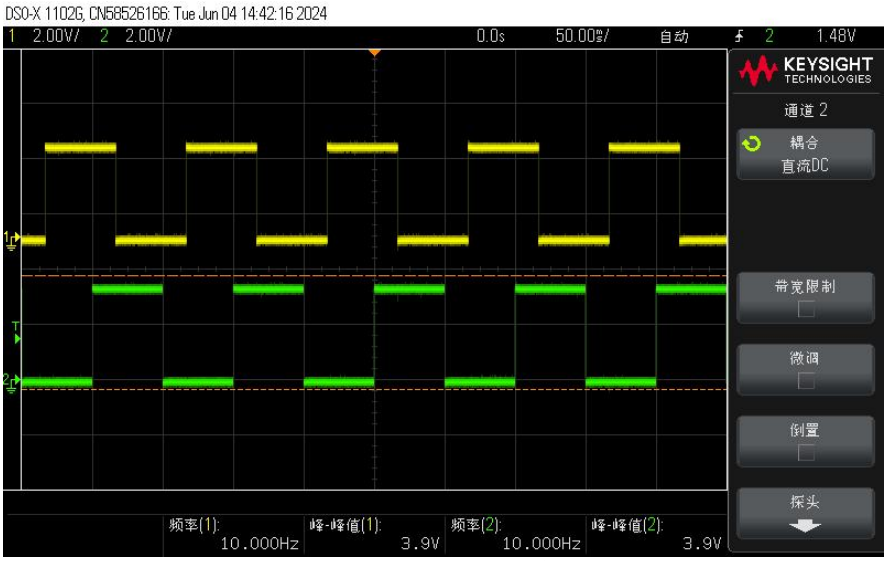


图1.17 输出C与A波形

4.CH1检测输出分频脉冲f的波形，频率为60.01Hz，CH2检测输出A脉冲的波形，频率为10.000Hz，，为上升沿触发，是f的六分频。

图1.18 输出f与A波形

步进电机脉冲分配器工作模式为三相六拍，两位数码管显示1Hz -- 99Hz脉冲频率，频率可由按钮调节，通过滑动开关可控制运行/停止状态与正转/反转状态，检测到A、B、C 三相脉冲及频率f符合预期，可以正常地运行，符合设计要求。

心得体会：

在本次步进电机控制器设计实验中，我学习了步进电机的工作原理，了解了其相序控制和脉冲信号的作用，学会了FPGA编程的基本流程，编写VHDL代码实现了步进电机的控制逻辑，在编写代码的过程中，我巩固了VHDL编程的基础，熟悉了VHDL语句的并行运行规律，最后将步进电机控制器与DE10-Lite开发板进行了验证，确保程序功能正确。通过设计和实现步进电机控制器，我对数字逻辑设计和FPGA编程有了更深入的了解。

通过这次实验，我不仅掌握了FPGA设计的基本流程，还学会了面对问题的解决思路和方法。未来，我将继续注重理论与实践的结合，不断深化对硬件设计的理解，提升自己的综合设计能力，为今后的学习和工作做好充分准备。