浙江大学 实验报告

专业：电子信息工程
姓名：
学号：

课程名称：电路与电子技术实验 II　　指导老师：张伟　　地点：紫金港东三 406
实验名称：步进电机脉冲分配器　　同组学生：　　日期：2024 年 6 月 10 日

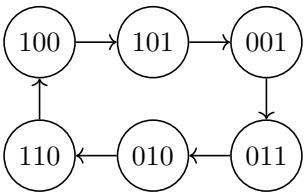
1 实验目的

1. 学会使用 VHDL 语言描述复杂逻辑电路——步进电机脉冲分配器。
2. 掌握步进电机脉冲分配器的工作原理。
3. 掌握步进电机脉冲分配器的设计方法。

2 实验原理

2.1 步进电机脉冲分配器

电路的状态转移图为：



在上图的基础上设计脉冲分配器

2.2 程序流程分析

设计基本程序流程如下所示，完成以下基本功能：

1. 频率改变从 1Hz 99Hz
2. 正转、反转
3. 基本的脉冲分配器功能

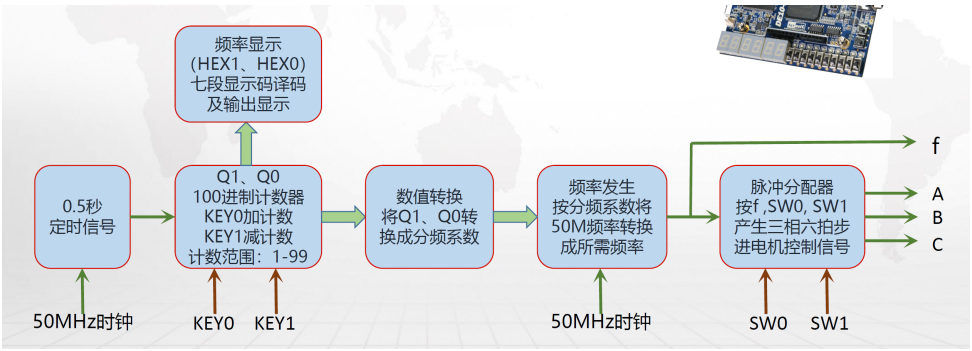


图 1: 程序流程图

2.3 程序设计

2.3.1 2Hz 脉冲产生程序

```
entity clk2hz is
    port(
        clk : in std_logic;
        clock2hz : out std_logic
    );
end clk2hz;

architecture Behavioral of clk2hz is
    constant m : integer :=
        12500000;
    signal tmp : std_logic;
begin
    process(clk,tmp)
        variable cout : integer := 0;
    begin
        if rising_edge(clk) then
        cout := cout+1;
            if cout <= m then
                tmp <= '0';
            elsif cout < m*2 then
                tmp <= '1';
            else
                cout := 0;
            end if;
        end if;
    end process;
    clock2hz <= tmp;
end Behavioral;
```

2.4 1-99Hz 脉冲产生程序

```
entity Freqg is
    port(
        clk50m : in std_logic; -- 50 MHz时钟输入
        key0: in STD_LOGIC; -- 增加信号输入
        key1: in STD_LOGIC; -- 减小信号输入
        q0: out STD_LOGIC_VECTOR (3 DOWNTO 0); -- 个位数码管显示
        q1: out STD_LOGIC_VECTOR (3 DOWNTO 0); -- 十位数码管显示
        freq : out std_logic
    );
end Freqg;

architecture Behavioral of Freqg is
    COMPONENT clk2hz
        port(
            clk : in std_logic;
            clock2hz : out std_logic
        );
    end COMPONENT;
    COMPONENT KeyInput
    PORT(
        clk2Hz: IN STD_LOGIC;
        key0: IN STD_LOGIC;
        key1: IN STD_LOGIC;
        q0: OUT STD_LOGIC_VECTOR (3 DOWNTO 0); -- 个位数码管显示
        q1: OUT STD_LOGIC_VECTOR (3 DOWNTO 0) -- 十位数码管显示
    );
    end COMPONENT;

    signal clock2hz : std_logic;
    signal q2,q3,q4,q5 : std_logic_vector(7 downto 0) := "00000000";
        signal q00,q11 : std_logic_vector(3 downto 0);
    signal k : integer range 0 to 25000000;
begin
    u0 : clk2hz port map ( clk => clk50m, clock2hz => clock2hz );
    u1 : KeyInput port map ( clk2Hz => clock2hz, key0 => key0, key1 => key1, q0 => q00, q1 => q11 );
    q0<=q00;
        q1<=q11;
        process(key0,key1)
----- 输入数据进制转换 -----
    begin
        if key0 = '1' and key1 = '1' then
            q2(3 downto 0) <= q00;
            q3(4 downto 1) <= q11;
            q4(6 downto 3) <= q11;
            q5 <= q2+q3+q4;
            k <= 25000000/conv_integer(q5);
        end if;
    end process;
----- 产生1-99Hz的脉冲 -----
    process(clk50m)
    variable cout: integer := 0;
    begin
      --略--
    end process;
end Behavioral;
```

2.4.1 脉冲分配器

```
entity PulseAssign is
    port(
        freq : in std_logic;
        sw1_x : in std_logic;
        sw0_x : in std_logic;
        A : out std_logic;
        B : out std_logic;
        C : out std_logic
    );
end PulseAssign;
architecture Behavioral of PulseAssign is
    signal qa: std_logic :='1';
        signal qb,qc : std_logic :='0';
begin
    process(freq)
    begin
        if rising_edge(freq) then
            qa <= not((sw1_x and qb) or ((not sw1_x) and qc));
            qb <= not((sw1_x and qc) or ((not sw1_x) and qa));
            qc <= not((sw1_x and qa) or ((not sw1_x
```

```
                ) and qb));
        end if;
    end process;
    A <=qa and sw0_x;
    B <=qb and sw0_x;
    C <=qc and sw0_x;
end Behavioral;
```

2.4.2 主程序

```
entity Motor is
    port(
        clk50m : in std_logic;
        key0,key1: in std_logic;
        sw0,sw1: in std_logic;
        A : out std_logic;
        B : out std_logic;
        C : out std_logic;
        freq : out std_logic;
        seg_dis_q1 : OUT
            STD_LOGIC_VECTOR
            (6 DOWNTO 0);
        seg_dis_q0 : OUT
            STD_LOGIC_VECTOR
            (6 DOWNTO 0)
    );
end Motor;

architecture Behavioral of Motor is
    COMPONENT Freqg
        PORT(
            clk50m : in std_logic;
            key0: in STD_LOGIC;
            key1: in STD_LOGIC;
            q0: out
                STD_LOGIC_VECTOR
                (3 DOWNTO 0);
            q1: out
                STD_LOGIC_VECTOR
                (3 DOWNTO 0);
        freq : out std_logic
    );
END COMPONENT;
COMPONENT PulseAssign
port(
    freq : in std_logic;
    sw1_x : in std_logic;
    sw0_x : in std_logic;
    A : out std_logic;
    B : out std_logic;
    C : out std_logic
);
end COMPONENT;
COMPONENT hex7seg
    PORT(
        data_in : IN
            STD_LOGIC_VECTOR
            (3 DOWNTO 0);
        seg_dis : OUT
            STD_LOGIC_VECTOR
            (6 DOWNTO 0)
    );
end COMPONENT;
signal q0,q1 : std_logic_vector(3
    DOWNTO 0);
signal freq_in : std_logic;
begin
    Freqg1: Freqg PORT MAP(
        clk50m => clk50m,
        key0 => key0,
        key1 => key1,
        q0 => q0,
        q1 => q1,
        freq => freq_in
    );
    PulseAssign1: PulseAssign
        PORT MAP(
        freq => freq_in,
        sw1_x => sw1,
        sw0_x => sw0,
        A => A,
        B => B,
        C => C
    );
    hex7seg1: hex7seg PORT MAP(
        data_in => q0,
        seg_dis => seg_dis_q0
    );
    hex7seg2: hex7seg PORT MAP(
        data_in => q1,
        seg_dis => seg_dis_q1
    );
    freq<=freq_in;
end Behavioral;
```

其他部分程序略。

3 实验过程与结果

1. 创建工程文件

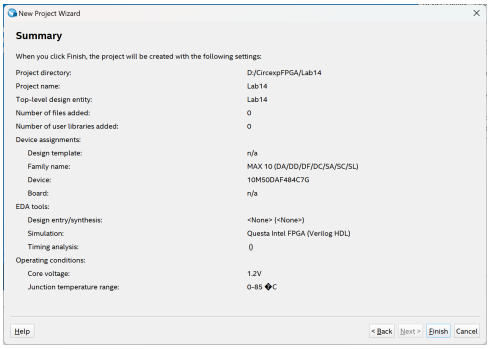


图 2: 创建工程文件结果图

2. 添加源文件

3. 编译，得到基本结构如下图所示

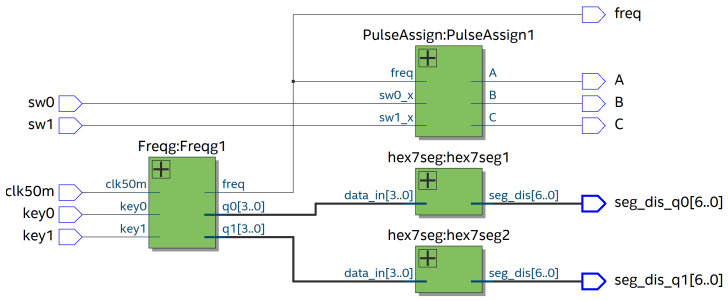
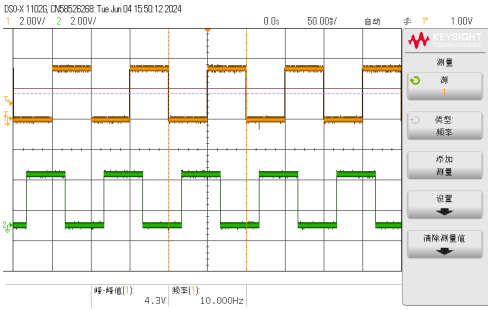
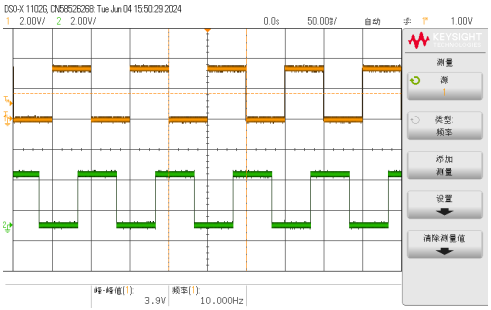


图 3: 编译结果

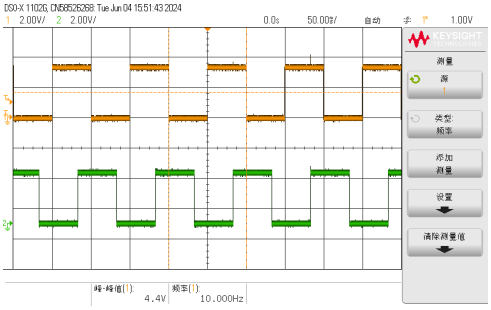
4. 下载到 FPGA 板上，实验效果如下图所示



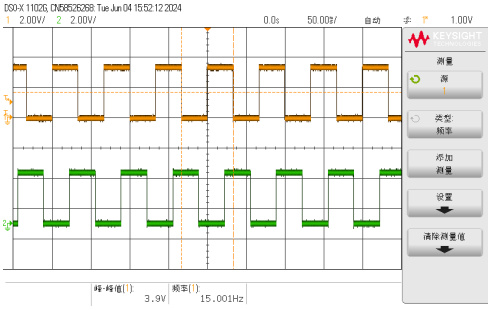
(a) CHA-CHB，反转结果



(b) CHA-CHC，反转结果



(c) CHA-CHB，正转结果，输入频率：60Hz



(d) CHA-CHB，正转结果，输入频率：90Hz

图 4: 实验结果