

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： 数字钟

同组学生： _____

日期： 2024 年 6 月 2 日

1 实验目的

1. 学会使用 VHDL 语言描述数字电路；
2. 学会使用 FPGA 编写基本数字电路；
3. 掌握 Quartus 软件的基本用法。

2 实验原理

2.1 1Hz 时钟信号的产生

```
1  --clk1hz.vhd--
2  library IEEE;
3  use IEEE.std_logic_1164.ALL;
4  --实体定义--
5  entity clk1hz is
6      port(
7          clk : in std_logic;
8          clock1hz : out std_logic
9      );
10 end clk1hz;
11 --结构定义--
12 architecture Behavioral of clk1hz is
13     constant m : integer :=
14         25000000; -- 50MHz
15         /2--
16     signal tmp : std_logic;
17 begin
18     --时序定义需用process--
19     process(clk,tmp)
20         variable cout : integer := 0;
21     begin
22         if rising_edge(clk) then
23             cout := cout+1;
24             if cout <= m then
25                 tmp <= '0';
26             elsif cout < m*2 then
27                 tmp <= '1';
28             else
29                 cout := 0;
30             end if;
31         end if;
32     end process;
33     clock1hz <= tmp;
34 end Behavioral;
```

2.2 7 位数码管

```
1  --hex7seg.vhd--
2  LIBRARY IEEE;
3  USE IEEE.STD_LOGIC_1164.ALL;
4
5  ENTITY hex7seg IS
6      PORT(
7          data_in : IN STD_LOGIC_VECTOR(3
8              DOWNTO 0);
9          seg_dis : OUT STD_LOGIC_VECTOR(6
10              DOWNTO 0)
11      );
12 END hex7seg;
13 ARCHITECTURE Behavioral OF hex7seg IS
14 BEGIN
15     PROCESS (data_in)
16     BEGIN
17         CASE data_in IS
18             when "0000" => seg_dis <= "1000000"; --0
19             when "0001" => seg_dis <= "1111001"; --1
20             when "0010" => seg_dis <= "0100100"; --2
21             when "0011" => seg_dis <= "0110000"; --3
22             when "0100" => seg_dis <= "0011001"; --4
23             when "0101" => seg_dis <= "0010010"; --5
24             when "0110" => seg_dis <= "0000010"; --6
25             when "0111" => seg_dis <= "1111000"; --7
26             when "1000" => seg_dis <= "0000000"; --8
27             when "1001" => seg_dis <= "0010000"; --9
28             when "1010" => seg_dis <= "0001000"; --A
29             when "1011" => seg_dis <= "0000011"; --B
30             when "1100" => seg_dis <= "1000110"; --C
31             when "1101" => seg_dis <= "0100001"; --D
32             when "1110" => seg_dis <= "0000110"; --E
33             when "1111" => seg_dis <= "0001110"; --F
34         end case;
35     END PROCESS;
36 END Behavioral;
```

2.3 数字钟代码编写

在上述两个代码的基础上，我们可以编写数字钟的代码，如下所示：

```
1  --clock.vhd--
2  LIBRARY IEEE;
3  USE IEEE.STD_LOGIC_1164.ALL;
4  USE IEEE.STD_LOGIC_ARITH.ALL;
5  USE IEEE.STD_LOGIC_UNSIGNED.ALL;
6  --实体定义--
7  ENTITY Clock IS
8      PORT(
9          clk50m: IN STD_LOGIC; -- FPGA时钟信号输入
10         clr: IN STD_LOGIC; -- 分隔移位信号
11         inc_min: IN STD_LOGIC; -- 分隔调整增加
12         dec_min: IN STD_LOGIC; -- 小时调整减小
13         inc_hour: IN STD_LOGIC; -- 小时调整增加
14         dec_hour: IN STD_LOGIC; -- 小时调整减小
15         seg_dis_hour0: OUT STD_LOGIC_VECTOR(6
16             DOWNT0 0); -- 小时位数码管显示
17         seg_dis_hour1: OUT STD_LOGIC_VECTOR(6
18             DOWNT0 0); -- 小时个位数数码管显示
19         seg_dis_min0: OUT STD_LOGIC_VECTOR(6
20             DOWNT0 0); -- 分钟十位数数码管显示
21         seg_dis_min1: OUT STD_LOGIC_VECTOR(6
22             DOWNT0 0); -- 分钟个位数数码管显示
23         seg_dis_sec0: OUT STD_LOGIC_VECTOR(6
24             DOWNT0 0); -- 秒十位数数码管显示
25         seg_dis_sec1: OUT STD_LOGIC_VECTOR(6
26             DOWNT0 0) -- 秒个位数的数码管显示
27     );
28 END Clock;
29
30 ARCHITECTURE Behavioral OF Clock IS
31     -- 秒信号，引用clk1hz.vhd
32     COMPONENT clk1hz
33     PORT(
34         clk: IN STD_LOGIC;
35         clock1hz: OUT STD_LOGIC
36     );
37 END COMPONENT;
38
39 -- 数码管译码，引用hex7seg.vhd
40 COMPONENT hex7seg
41 PORT(
42     data_in: IN STD_LOGIC_VECTOR(3
43         DOWNT0 0);
44     seg_dis: OUT STD_LOGIC_VECTOR(6
45         DOWNT0 0)
46 );
47 END COMPONENT;
48
49 signal clock1hz : std_logic;
50 -- 秒、分、时的十位和个位 --
51 signal qsec0 : std_logic_vector(3 downto 0);
52 signal qsec1 : std_logic_vector(3 downto 0);
53 signal qmin0 : std_logic_vector(3 downto 0);
54 signal qmin1 : std_logic_vector(3 downto 0);
55 signal qhour0 : std_logic_vector(3 downto 0);
56 signal qhour1 : std_logic_vector(3 downto 0);
57
58 BEGIN
59     u0 : clk1hz port map ( clk => clk50m, clock1hz =>
60         clock1hz );
61     u1 : hex7seg port map ( data_in => qsec0, seg_dis
62         => seg_dis_sec0 );
63     u2 : hex7seg port map ( data_in => qsec1, seg_dis
64         => seg_dis_sec1 );
65     u3 : hex7seg port map ( data_in => qmin0, seg_dis
66         => seg_dis_min0 );
67     u4 : hex7seg port map ( data_in => qmin1, seg_dis
68         => seg_dis_min1 );
69     u5 : hex7seg port map ( data_in => qhour0, seg_dis
70         => seg_dis_hour0 );
71     u6 : hex7seg port map ( data_in => qhour1, seg_dis
72         => seg_dis_hour1 );
73
74     process(clock1hz,clr) -- 将clock1hz和clr信号作为敏感
75         信号传入
76     begin
77         -- 清零 --
78         if (clr = '0') then
79             qsec0 <= "0000";
80             qsec1 <= "0000";
81             qmin0 <= "0000";
82             qmin1 <= "0000";
83             qhour0 <= "0000";
84             qhour1 <= "0000";
85         elsif rising_edge(clock1hz) then -- 上升沿触发
86             -- 分调整增加
87             if (inc_min = '1') then
88                 if (qmin0 = "1001" and qmin1 = "0101")
89                     then
90                     qmin0 <= "0000";
91                     qmin1 <= "0000"; -- 59分时归零
92                 elsif (qmin0 = "1001") then
93                     qmin0 <= "0000";
94                     qmin1 <= qmin1 + 1; -- 一个位为9时
95                     进位
96                 else
97                     qmin0 <= qmin0 + 1;
98             end if;
99             -- 分调整减小
100            elsif (dec_min = '1') then
101                if (qmin0 = "0000" and qmin1 = "0000")
102                    then
103                    qmin0 <= "1001";
104                    qmin1 <= "0101"; -- 0分时变为59分
105                elsif (qmin0 = "0000") then
106                    qmin0 <= "1001";
107                    qmin1 <= qmin1 - 1; -- 一个位为0时
108                    借位
109                else
110                    qmin0 <= qmin0 - 1;
111            end if;
112        end process;
```

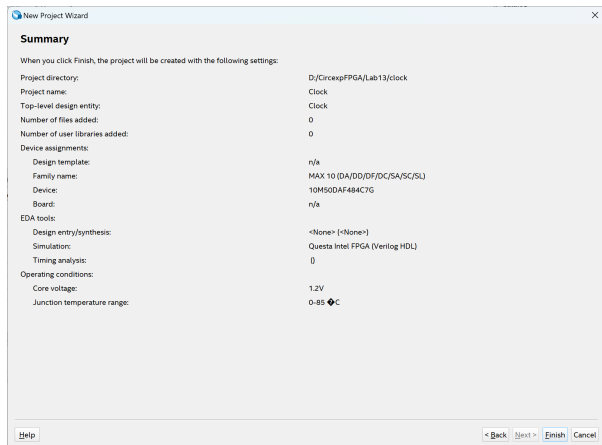
```

91         end if; 130
92
93         -- 小时调整增加 131
94         elsif(inc_hour='1') then 132
95             -- 23清零 -- 133
96             if(qhour0="0011" and qhour1="0010") 134
97                 then 135
98                     qhour0 <= "0000"; 136
99                     qhour1 <= "0000"; 137
100                 -- 9 进位 -- 138
101             elsif(qhour0="1001") then 139
102                 qhour0 <= "0000"; 140
103                 qhour1 <= qhour1 + 1; 141
104             else 142
105                 qhour0 <= qhour0 + 1; 143
106             end if; 144
107
108         -- 小时调整减小 145
109         elsif(dec_hour='1') then 146
110             if(qhour0="0000" and qhour1="0000") 147
111                 then 148
112                     qhour0 <= "0011"; 149
113                     qhour1 <= "0010"; 150
114             elsif(qhour0="0000") then 151
115                 qhour0 <= "1001"; 152
116                 qhour1 <= qhour1 - 1; 153
117             else 154
118                 qhour0 <= qhour0 - 1; 155
119             end if; 156
120
121         -- 时钟信号处理 计时到 23:59:59 时清零-- 157
122         elsif(qsec0="1001" and qsec1="0101" and 158
123             qmin0="1001" and qmin1="0101" 159
124             and qhour0="0011" and qhour1="0010") 160
125             then 161
126                 qsec0 <= "0000"; 162
127                 qsec1 <= "0000"; 163
128                 qmin0 <= "0000"; 164
129                 qmin1 <= "0000"; 165
130                 qhour0 <= "0000"; 166
131                 qhour1 <= "0000"; 167
132
133         -- 计时到 "9:59:59" 和 "19:59:59" 时分、秒清 168
134         零, 小时时位增一 --

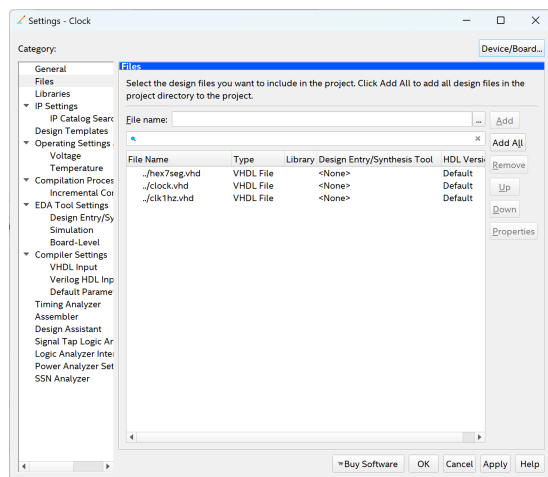
```

3 实验过程与结果

1. 创建项目，结果如下图所示：
2. 将实验所需的 VHDL 文件加入工程中；



(a) 创建项目，简介



(b) 添加 VHDL 文件

3. 设定顶层设计

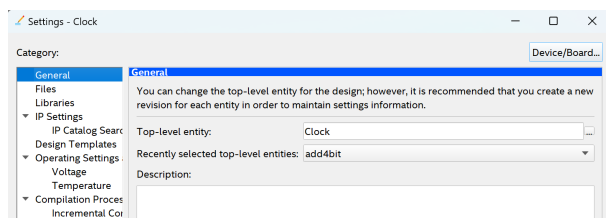


图 2: 设定顶层设计

4. 运行代码，编译成功

5. 对引脚进行分配

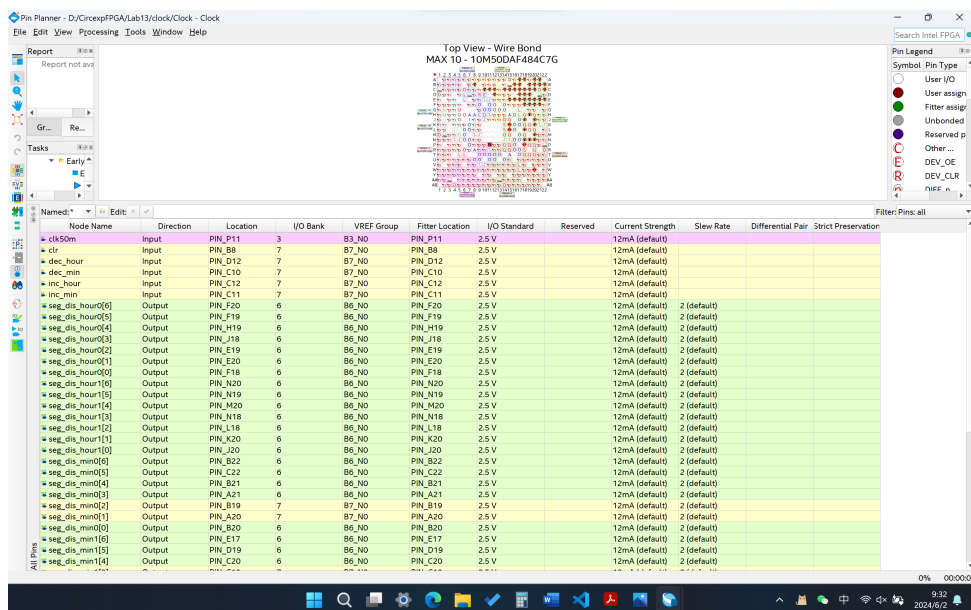


图 3: 引脚分配

6. 下载到 FPGA 板上，运行代码，结果如下图所示：

7. 实验结果

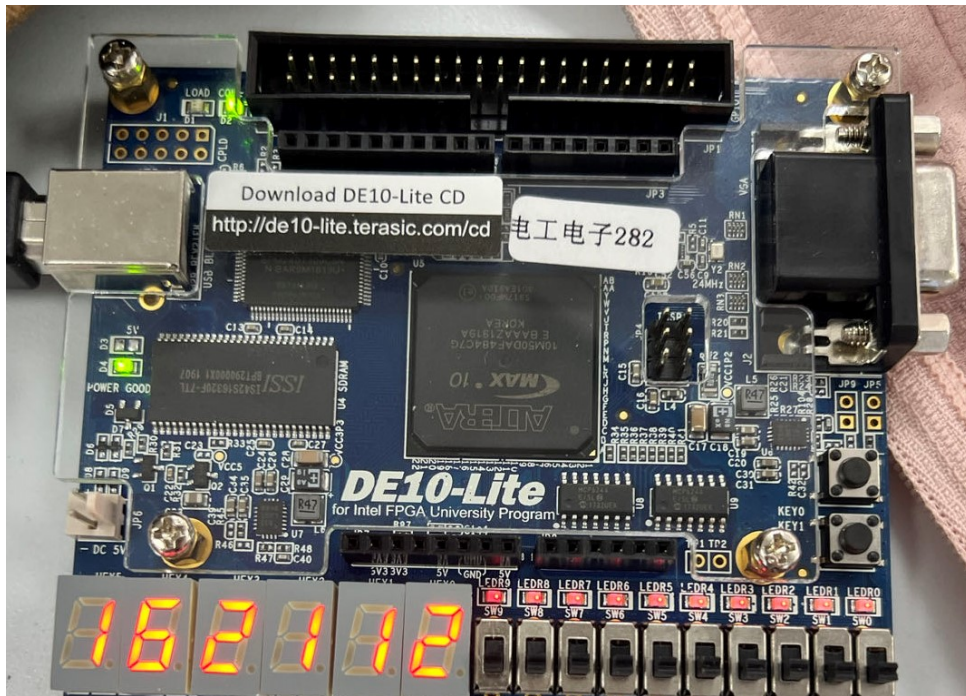


图 4: 实验结果