# 硬件描述训练 2(第八次 1、2)实验报告

软件 1801 肖云杰 201826010113

(本次实验做完了基于 D、C 级任务的 B 级任务, 报告如下)

(D、C 级任务的文件部分已经根据 B 更改, 图片可能会超出等级)

### D级任务、C级任务

使用状态图生成实验 VHDL 文件的基本框架,将状态图作修改,使其不包括输入输出(变得容易处理),同时输入输出文件,单独写 VHDL 文件,分别作为模块,加入到顶层文件中。

加入时钟信号,手动重写模八计数器,使模八计数器能将时钟信号转化为8位的位选信号,并根据位选信号,从8位二进制(0,1)输入中选取相应位,选取其段选信号位,显示。

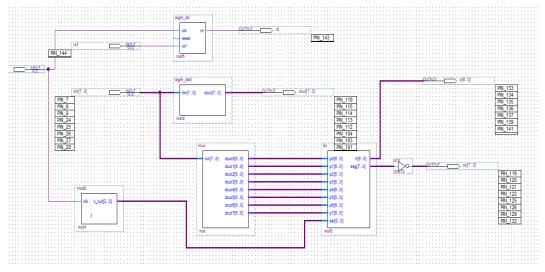
### 状态图和部分 VHDL 文件如下

```
library IEEE;
     use IEEE.std logic 1164.all;
    entity eight_ddd is
 3
    mport (
         din:in std_logic_vector(7 downto 0);
 5
 6
         dout: out std logic vector(7 downto 0));
     end entity;
    ■architecture bhv of eight ddd is
 9
    ■begin
10
        dout<=din ;
11
     end bhv;
```

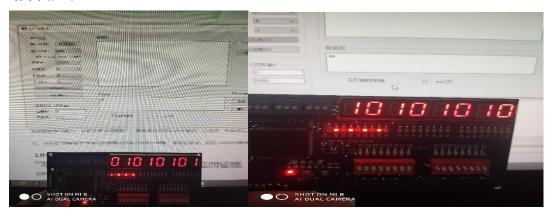
```
library IEEE;
                                                                                use ieee.std logic 1164.all;
                                                                              entity mux is
                                                                              port (
       library IEEE;
                                                                                          dout0:out STD_LOGIC_VECTOR(6 downto 0);
dout1:out STD_LOGIC_VECTOR(6 downto 0);
dout2:out STD_LOGIC_VECTOR(6 downto 0);
dout3:out STD_LOGIC_VECTOR(6 downto 0);
dout4:out STD_LOGIC_VECTOR(6 downto 0);
       use ieee.std logic 1164.all;
       use ieee.std logic unsigned.all;
      use ieee std logic arith all;
                                                                       11
                                                                                          dout5:out STD_LOGIC_VECTOR(6 downto 0);
dout6:out STD_LOGIC_VECTOR(6 downto 0);
 5 ≣entity mod8 is
 6 ■port(
                                                                       13
                                                                                           dout7:out STD_LOGIC_VECTOR(6 downto 0));
                                                                       14
                                                                                           end mux;
            clk:in STD LOGIC;
                                                                             ■ architecture rtl of mux is
 8
            c out:out STD LOGIC VECTOR(2 downto 0));
                                                                       16
                                                                             ■ begin
      end mod8;
                                                                       18
                                                                                     if cin(0)='0' then dout0<="0|111111";
10 ■architecture mo8 of mod8 is
                                                                       20
                                                                                     elsif cin(0)='1' then dout0<="0000110";
                                                                       21
                                                                                     end if;
11 signal iq:integer;
                                                                       22
                                                                                     if cin(1)='0' then dout1<="0111111";
elsif cin(1)='1' then dout1<="0000110";</pre>
                                                                       23
12 ■begin
                                                                                     end if;
if cin(2)='0' then dout2<="0111111";
elsif cin(2)='1' then dout2<="0000110";
13 process(clk)
                                                                       25
14
            begin
                                                                       27
28
                                                                                      end if:
                                                                                     if cin(3)='0' then dout3<="0111111";
15
          if clk'event and clk='l' then
                                                                                      elsif cin(3)='1' then dout3<="0000110";
16
            if iq=8 then iq<=0;
                                                                       30
                                                                                     end if;
                                                                                     if cin(4)='0' then dout4<="0111111";
elsif cin(4)='1' then dout4<="0000110";
17 = else iq<=iq+1;
                                                                       32
18
            end if;
                                                                       34
35
                                                                                     if cin(5)='0' then dout5<="01111111";
elsif cin(5)='1' then dout5<="0000110";</pre>
19
            end if:
                                                                                      if cin(6)='0' then dout6<="0111111";
20
            c out <= conv std logic vector(iq,3);
                                                                       37
                                                                                      elsif cin(6)='1' then dout6<="0000110";
21
            end process;
                                                                       39
                                                                                      end if:
                                                                                     if cin(7)='0' then dout7<="0111111";
elsif cin(7)='1' then dout7<="0000110";
                                                                       40
      end mo8;
                                                                       41
                                                                       42
                                                                                     end if;
                                                                             end rtl;
```

```
library IEEE;
use ieee.std_logic_ll64.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
  2
   3
   4
   5
              mentity dx is
                port(
    p0,pl,p2,p3,p4,p5,p6,p7:in STD_LOGIC_VECTOR(6 downto 0);
    sel:in STD_LOGIC_VECTOR(2 downto 0);
    h:out STD_LOGIC_VECTOR(6 downto 0);
    seg:out std_logic_vector(7 downto 0));
end dx;
architecture
   6
7
             port(
   8
 10
 11
             ■ architecture dac of dx is
12
 13
              ■ begin
14
              process(sel)
                             process(sel)
begin
   if sel="000" then h<=p0;
   elsif sel="001" then h<=p1;
   elsif sel="010" then h<=p2;
   elsif sel="011" then h<=p3;
   elsif sel="100" then h<=p4;
   elsif sel="101" then h<=p5;
   elsif sel="110" then h<=p6;
   elsif sel="111" then h<=p7;
   end if:</pre>
15
16
17
              =
18
              19
              =
20
21
              _
              =
22
23
              24
                                         end if;
25
                              end process;
26
              process(sel)
                             process(sel)
begin

if sel="000" then seg<="00000001";
elsif sel="001" then seg<="0000010";
elsif sel="010" then seg<="00000100";
elsif sel="011" then seg<="000001000";
27
28
29
              30
              31
                                         elsif sel="011" then seg<="000010000";
elsif sel="100" then seg<="000100000";
elsif sel="101" then seg<="010000000";
elsif sel="110" then seg<="010000000";
elsif sel="111" then seg<="100000000";
32
              =
              33
34
              35
              36
37
                               end process;
38
                end dac;
39
```



#### 结果图如下



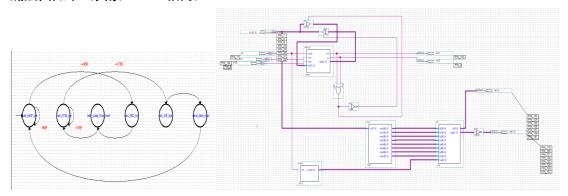
部分分析

数码管显示正确,选用 AA 和 55 是因为 AA 和 55 包括了所有的数码管,可以全方位检测问题。

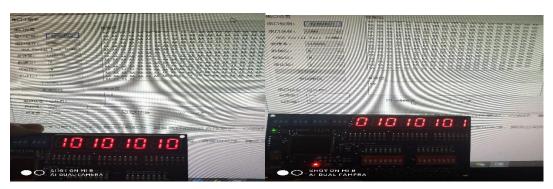
## B 级任务

采用状态图,实现基本底层的 VHDL 文件,在其中修改,得到正确的 VHDL,数码管显示电路同 C 级任务,连接输入,分配管脚即可

### 底层图如下(其余 VHDL 相同)



### 验证正确图如下



### 总结

本次实验中, 遇到问题, 其中

- 1. 位选信号低电平有效 (debug 了一个多小时才发现) -> 需要注重基础知识
- 2. 手写模八计数器和译码器和位选选择时,发现 VHDL 语法还是不是特别熟悉,部分代码需要参考之前写过的代码来写。

了解了状态图输入法的优点,极容易生成基本底层状态图,不过细节处理比较困难(可能是因为我还不太熟悉),可以在通过状态图生成的 VHDL 的文件上,修改 VHDL 的基本代码。然后再整合相关模块文件,进行不同模块的测试。

实现了上位机和实验板的相互通信,看到数据传输成功,在实验板的数码管上成功显示,有一种终于成功了的自豪感(修改数码管耗时 4h)