点亮数字人生

2017011341, 陈旭

2019 年 4 月

1 实验目的

- 通过数码管点亮程序, 熟悉 VHDL 语言, 了解掌握硬件程序的编写规范;
- 进一步理解可编程芯片的工作原理。

2 实验内容

- 同时点亮一个经过译码的数码管和一个未经过译码的数码管;
- 设计一个数码管显示实验, 要求有规律地显示数列（奇数列、偶数列、自然数列等), 尽可能多地点亮数码管。要求试验中至少使用一个不带译码的数码管。

3 代码与分析

3.1 实验内容 1

3.1.1 代码

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity test is
    port(
        key: in std_logic_vector(3 downto 0);
        display: out std_logic_vector(6 downto 0);
        display_4: out std_logic_vector(3 downto 0)
    );
end test;
architecture fire of test is
begin
```

```
14      display_4<=key;
15      process(key)
16      begin
17          case key is
18              when "0000"=>display<="1111110";
19              when "0001"=>display<="0110000";
20              when "0010"=>display<="1101101";
21              when "0011"=>display<="1111001";
22              when "0100"=>display<="0110011";
23              when "0101"=>display<="1011011";
24              when "0110"=>display<="0011111";
25              when "0111"=>display<="1110000";
26              when "1000"=>display<="1111111";
27              when "1001"=>display<="1110011";
28              when others=>display<="0000000";
29          end case;
30      end process;
31  end fire;
```

3.1.2 分析

设置 display 用于控制不带译码器的数码管，display_4 用于控制带译码器的数码管，key 表示开关的输入端口，通过 4 个开关来控制 4 个 key 的值，数码管显示相应的值。

3.2 实验内容 2

3.2.1 代码

```
1  LIBRARY IEEE;
2  USE IEEE.STD_LOGIC_1164.ALL;
3  USE IEEE.STD_LOGIC_ARITH.ALL;
4  USE IEEE.STD_LOGIC_UNSIGNED.ALL;
5  entity test is
6      port(
7          display: out std_logic_vector(6 downto 0);
8          display_4_oe: out std_logic_vector(2 downto 0);
9          oe: out std_logic;
10         clk: in std_logic;
11         rst: in std_logic;
12         btn: in std_logic
```

```
13      );
14  end test;
15  architecture fire of test is
16      signal display_4_buf: std_logic_vector(3 downto 0):="0000";
17      signal display_4_buf_oe: std_logic_vector(2 downto 0):="000";
18      signal cnt:integer:=0;
19      signal j:integer range 0 to 10:=1;
20      signal rst1:boolean:=false;
21      signal rst2:boolean:=false;
22      signal btn1:boolean:=false;
23      signal btn2:boolean:=false;
24      signal go:boolean:=true;
25  begin
26      process(clk)
27      begin
28          if (clk'event and clk='1') then
29              if (rst1 = NOT rst2) then
30                  display_4_buf<="0000";
31                  display_4_buf_oe<="000";
32                  j<=1;
33                  rst2<=rst1;
34                  oe<='0';
35                  cnt<=-1000000;
36                  go<=true;
37              else
38                  if (btn1 = NOT btn2) then
39                      go <= NOT go;
40                      btn2 <= btn1;
41                  end if;
42                  if (go) then
43                      cnt<=cnt+1;
44                      if (cnt=500000) then
45                          oe<='1';
46                      end if;
47                      if (cnt=1000000) then
48                          cnt<=0;
49                          j<=j+1;
50                          if (j=5) then
51                              j<=1;
```

```
52                              if (display_4_buf="1001") then
53                                  display_4_buf<="0000";
54                              else
55                                  display_4_buf<=display_4_buf+1;
56                              end if;
57                          end if;
58                          if (display_4_buf_oe="100") then
59                              display_4_buf_oe<="000";
60                          else
61                              display_4_buf_oe<=display_4_buf_oe+1;
62                          end if;
63                          oe<='0';
64                      end if;
65                  end if;
66              end if;
67              display_4_oe<=display_4_buf_oe;
68          end if;
69      end process;
70      process(rst)
71      begin
72          if (rst'event and rst='1') then
73              rst1<=NOT rst2;
74          end if;
75      end process;
76      process(btn)
77      begin
78          if (btn'event and btn='1') then
79              btn1<=NOT btn2;
80          end if;
81      end process;
82      process(display_4_buf)
83      begin
84          case display_4_buf is
85              when "0000"=>display<="1111110";
86              when "0001"=>display<="0110000";
87              when "0010"=>display<="1101101";
88              when "0011"=>display<="1111001";
89              when "0100"=>display<="0110011";
90              when "0101"=>display<="1011011";
```

```
91                when "0110"=>display <="1011111";
92                when "0111"=>display <="1110000";
93                when "1000"=>display <="1111111";
94                when "1001"=>display <="1111011";
95                when others=>display <="0000000";
96            end case;
97        end process;
98    end fire;
```

3.2.2 分析

这里控制了一个带译码器的数码管显示自然数列，另外两个带译码器的数码管将 2、4、8 端口与该译码器接在一起，1 端口分别接入高电平和低电平，这样就能轻易实现自然数列、奇数列、偶数列的显示了。不带译码器的数码管额外处理。

该代码还实现了通过按键来控制重置与暂停继续的功能（数列随着时间自动增长）。