点亮数字人生

计01　容逸朗　2020010869

1　实验内容

1. 使用至多两个带译码的数码管和至少一个不带译码的数码管，分别循环显示奇数列、偶数列和自然数列；
2. 使用一个不带译码的数码管，循环显示学号和实验室房号。（即 20200108699208）

其中 2 的优先级较 1 高，故只需要实现任务 2 即可。

2　实验步骤

2.1　实验过程

1. 首先编写实验代码；
2. 根据代码用到的接口适当更改 pin 的方案；
3. 编绎代码，若不通过则回到步骤 1；
4. 若编绎通过，则把程序导入芯片中；
5. 根据 pin 方案在合适位置上插线。

2.2　实验代码

实验所用的完整代码如下：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

entity id is
port(
  display: out std_logic_vector(6 downto 0); --不带译码器
  clk, rst: in std_logic
);
end id;

architecture bhv of id is

  signal cnt: integer := 1; --下一码的位置
  signal key: std_logic_vector(3 downto 0) := "0010"; --初值为 2
  begin

```

```
19    process(clk, rst)
20    begin
21
22      if (clk'event and clk='1') then --若按下 clk 键
23        cnt <= cnt + 1; --则显示下一位
24
25        if (cnt >= 13) then --到达循环终点
26          cnt <= 0; --置下一位为 0
27        end if;
28
29        case (cnt) is
30          when 0=> key <= "0010"; --2
31          when 1=> key <= "0000"; --0
32          when 2=> key <= "0010"; --2
33          when 3=> key <= "0000"; --0
34          when 4=> key <= "0000"; --0
35          when 5=> key <= "0001"; --1
36          when 6=> key <= "0000"; --0
37          when 7=> key <= "1000"; --8
38          when 8=> key <= "0110"; --6
39          when 9=> key <= "1001"; --9
40          when 10=> key <= "1001"; --9
41          when 11=> key <= "0010"; --2
42          when 12=> key <= "0000"; --0
43          when 13=> key <= "1000"; --8
44          when others=> key <="0111"; --输出不同的值用于检查错误
45        end case;
46
47      end if;
48
49      if (rst='1') then --若按下重置键
50        cnt <= 1; --下一个输出的位置
51        key <= "0010"; --则回到最初的情况
52      end if;
53    end process;
54
55
56    process(key) --不带译码器的需要进行译码处理
57    begin
58      case key is --以下是各数的编码规则
59        when"0000"=> display<="1111110"; --0
60        when"0001"=> display<="0110000"; --1
61        when"0010"=> display<="1101101"; --2
62        when"0011"=> display<="1111001"; --3
63        when"0100"=> display<="0110011"; --4
```

```
        when"0101"=> display<="1011011"; --5
        when"0110"=> display<="1011111"; --6
        when"0111"=> display<="1110000"; --7
        when"1000"=> display<="1111111"; --8
        when"1001"=> display<="1110011"; --9
        when"1010"=> display<="1110111"; --a
        when"1011"=> display<="0011111"; --b
        when"1100"=> display<="1001110"; --c
        when"1101"=> display<="0111101"; --d
        when"1110"=> display<="1001111"; --e
        when"1111"=> display<="1000111"; --f
        when others=>display<="0000000"; --其他情况全灭
      end case;
    end process;
end bhv;
```

2.3 代码说明

在实验中我主要利用了两个输入端和七个输出端：

```
entity id is
port(
  display: out std_logic_vector(6 downto 0); --不带译码器
  clk, rst: in std_logic
);
end id;
```

其中 `clk` 代表计数器，`rst` 代表重置键，`display` 是对应数码管的输出。

同时为了代码的简洁性，我加入了两个临时变量 `cnt` 和 `key` ，前者代表计数器当前值（对应下一位的位置），后者代表应输出的数码。

```
signal cnt: integer := 1; --下一码的位置
signal key: std_logic_vector(3 downto 0) := "0010"; --初值为 2
```

按下计数器键后，需要增加计数器的值，若计数器到达循环的最后一位时重置为 0 。可以这样操作的原因是由于同一个 process 内的代码是并行计算的，因此可以在计数器（代表下一个位置），才不会影响输出。

```
process(clk, rst)
  begin
    if (clk'event and clk='1') then --若按下 clk 键
      cnt <= cnt + 1; --则显示下一位

    if (cnt >= 13) then --到达循环终点
```

```
      cnt <= 0; --置下一位为 0
    end if;

    -- 按 cnt 的值更改 key

    -- 若 rst 被按下，则重设

  end if;
end process;
```

在过程中还需要按计数器更改 `key` 的值：

```
case (cnt) is
  when 0=> key <= "0010"; --2
  when 1=> key <= "0000"; --0
  when 2=> key <= "0010"; --2
  when 3=> key <= "0000"; --0
  when 4=> key <= "0000"; --0
  when 5=> key <= "0001"; --1
  when 6=> key <= "0000"; --0
  when 7=> key <= "1000"; --8
  when 8=> key <= "0110"; --6
  when 9=> key <= "1001"; --9
  when 10=> key <= "1001"; --9
  when 11=> key <= "0010"; --2
  when 12=> key <= "0000"; --0
  when 13=> key <= "1000"; --8
  when others=> key <="0111"; --输出不同的值用于检查错误
end case;
```

还需要考虑按下重置键的情况，这时相当于回到变量的初始值：

```
if (rst='1') then --若按下重置键
  cnt <= 1; --下一个输出的位置
  key <= "0010"; --则回到最初的情况
end if;
```

我们还需要一个程序将四位数据绎为七位编码：

```
process(key) --不带译码器的需要进行译码处理
begin
  case key is --以下是各数的编码规则
    when"0000"=> display<="1111110"; --0
    when"0001"=> display<="0110000"; --1
```

```
    when"0010"=> display<="1101101"; --2
    when"0011"=> display<="1111001"; --3
    when"0100"=> display<="0110011"; --4
    when"0101"=> display<="1011011"; --5
    when"0110"=> display<="1011111"; --6
    when"0111"=> display<="1110000"; --7
    when"1000"=> display<="1111111"; --8
    when"1001"=> display<="1110011"; --9
    when"1010"=> display<="1110111"; --a
    when"1011"=> display<="0011111"; --b
    when"1100"=> display<="1001110"; --c
    when"1101"=> display<="0111101"; --d
    when"1110"=> display<="1001111"; --e
    when"1111"=> display<="1000111"; --f
    when others=>display<="0000000"; --其他情况全灭
  end case;
end process;
```