

# 点亮数字人生实验报告

翁家翌 2016011446

2018-03-31

## 目录

<b>1</b>	<b>实验目的</b>	<b>2</b>
<b>2</b>	<b>实验内容</b>	<b>2</b>
<b>3</b>	<b>代码及注释</b>	<b>2</b>
3.1	实验内容 1 . . . . .	2
3.2	实验内容 2 . . . . .	3
<b>4</b>	<b>实验小结</b>	<b>6</b>

# 1 实验目的

1. 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。
2. 掌握 EDA 软件的使用方法和工作流程。
3. 进一步理解可编程芯片的工作原理。

# 2 实验内容

1. 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。数码管从 0 到 f（带译码的显示 0 到 9）。
2. 设计一个数码管显示实验，要求 7 段数码管有规律的显示数列（例如：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 代码及注释

## 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 lighten is
    port(
        key: in std_logic_vector(3 downto 0); --控制开关
        display: out std_logic_vector(6 downto 0); --不带译码器
        display_4: out std_logic_vector(3 downto 0) --带译码器
    );
end lighten;

architecture fire of lighten is
begin
    display_4<=key; --带译码器的可以直接输出开关值
    process(key)    --不带译码器的需要进行译码处理
    begin
```

```

case key is --以下是 0~F 的编码规则
    when "0000"=>display<="1111110";
    when "0001"=>display<="0110000";
    when "0010"=>display<="1101101";
    when "0011"=>display<="1111001";
    when "0100"=>display<="0110011";
    when "0101"=>display<="1011011";
    when "0110"=>display<="0011111";
    when "0111"=>display<="1110000";
    when "1000"=>display<="1111111";
    when "1001"=>display<="1110011";
    when "1010"=>display<="1110111";
    when "1011"=>display<="0011111";
    when "1100"=>display<="1001110";
    when "1101"=>display<="0111101";
    when "1110"=>display<="1001111";
    when "1111"=>display<="1000111";
    when others=>display<="0000000"; --其他情况全灭
end case;
end process;
end fire;

```

### 3.1.2 工作原理

设置两个输出端口 display(不带译码器) 和 display\_4 (带译码器), 一个开关输入端口 key, 通过 4 个开关, 控制 4 个 key 的 0,1 输入值, 数码管显示相应的 (0-f) 的值, 其中带译码器的 display\_4 直接输入, 不带译码器的通过进程对 key 值进行码制转换。

### 3.1.3 测试结果

带译码器和不带译码器的两个数码管可以根据开关的输入值准确地显示相同的数值。

## 3.2 实验内容 2

### 3.2.1 代码

```

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

```

```

USE IEEE.STD_LOGIC_UNSIGNED.ALL;

entity lighten is
    port(
        display: out std_logic_vector(6 downto 0); --不带译码器
        display_4: out std_logic_vector(3 downto 0); --带译码器
        display_4_2: out std_logic_vector(3 downto 0); --带译码器
        clk: in std_logic --输入时钟信号
    );
end lighten;

architecture fire of lighten is
    signal display_4_buf: std_logic_vector(3 downto 0):="0000"; --自然数列
    signal display_4_buf_odd: std_logic_vector(3 downto 0):="0001"; --奇数列
    signal display_4_buf_even: std_logic_vector(3 downto 0):="0000"; --偶数列
    signal cnt: integer:=0; --计数器
begin
    process(clk)
    begin
        if (clk'event and clk='1') then --'
            if (cnt<1000000) then
                cnt<=cnt+1; --输入时钟频率为 1M 的时候每秒跳动一个数字
            else
                cnt<=0;
                if (display_4_buf="1001") then --自然数列 9->0
                    display_4_buf<="0000";
                else
                    display_4_buf<=display_4_buf+1;
                end if;
                if (display_4_buf_even="1000") then --偶数列 8->0
                    display_4_buf_even<="0000";
                else
                    display_4_buf_even<=display_4_buf_even+2;
                end if;
                if (display_4_buf_odd="1001") then --奇数列 9->1
                    display_4_buf_odd<="0001";
                else
                    display_4_buf_odd<=display_4_buf_odd+2;
                end if;
            end if;
        end if;
    end process;
end fire;

```

```

        end if;
        display_4<=display_4_buf_odd;    --带译码器显示奇数列
        display_4_2<=display_4_buf_even; --带译码器显示偶数列
    end if;
end process;
process(display_4_buf) --不带译码器的需要进行译码处理
begin
    case display_4_buf is --以下是 0~F 的编码规则
        when "0000"=>display<="1111110";
        when "0001"=>display<="0110000";
        when "0010"=>display<="1101101";
        when "0011"=>display<="1111001";
        when "0100"=>display<="0110011";
        when "0101"=>display<="1011011";
        when "0110"=>display<="0011111";
        when "0111"=>display<="1110000";
        when "1000"=>display<="1111111";
        when "1001"=>display<="1110011";
        when "1010"=>display<="1110111";
        when "1011"=>display<="0011111";
        when "1100"=>display<="1001110";
        when "1101"=>display<="0111101";
        when "1110"=>display<="1001111";
        when "1111"=>display<="1000111";
        when others=>display<="0000000"; --其他情况全灭
    end case;
end process;
end fire;

```

### 3.2.2 工作原理

通过 display\_4\_buffer\_odd 和 display\_4\_buffer\_even 两个 signal 保存奇数列和偶数列的值，然后通过 display\_4 和 display\_2 两个带译码器的输出显示。display\_4\_buffer 保存自然数列的值，通过 display 显示。clk 时钟信号遇到上升沿的时候计数器 +1，当计数器达到 1M 的时候三个信号的值 +1，当值即将溢出的时候使它的值从开始的数值开始即可。

### 3.2.3 测试结果

两个带译码器的数码管分别显示奇数列和偶数列，不带译码器的数码管显示自然数列，跳动的频率由输入的时钟信号控制，输入时钟信号为 1MHz 的时候，三个数列的数字每秒跳动一次。

## 4 实验小结

1. 这是我第一次进行可编程器件，在编程的过程中遇到了诸如软件的安装和破解、VHDL 语法的学习、USB 驱动的下载和安装、代码的编译和导入等等，我和舍友分工合作，共同探讨研究，在较短的时间内解决了这些问题，算是以较高的效率入了门吧；
2. 书后的示例代码对我的帮助很大，在上手一门语言的过程中，都是从依葫芦画瓢学习开始的，从书后简单的程序入手，节省了我不少对语法结构等学习的时间；
3. 对于提高要求功能的实现，我自己查阅了书上和网络上关于 signal 和时钟信号的 event 的相关资料，在纷繁复杂的资料中筛选对这次实验有用的信息，使我能成功完成这次实验，但是也因此没有系统地进行这些语法的学习，正所谓“路漫漫其修远兮，吾将上下而求索”。