

点亮数字人生

梁业升 2019010547 (计 03)

2022 年 4 月 12 日

1 实验内容

- 依次显示学号和实验室房间号 (2019010547+9208)
- 使用时钟控制, 每 1s 切换一次
- 支持复位

2 实现及其原理

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 digital_7 is
    port(
        display: out std_logic_vector(0 to 6); -- 译码后的信号
        clk: in std_logic;                      -- 时钟信号
        rst: in std_logic                       -- 复位信号
    );
end digital_7;

architecture bhv of digital_7 is
    -- 时钟上升沿次数
    signal cnt: integer := 0;
    -- 当前显示第几个数字 (共14个)
```

```
    signal index: std_logic_vector(3 downto 0) := "0000";
begin
    process(clk, rst)
    begin
        -- 如果有复位信号，恢复到初始状态
        if (rst = '1') then
            cnt <= 0;
            index <= "0000";

            -- 否则，判断时钟信号
        elsif (clk 'event and clk = '1') then
            if (cnt < 1000000) then
                -- 如果还没达到 1s (1MHz)
                cnt <= cnt + 1; -- 增加时钟计数
            else
                -- 达到了 1s
                cnt <= 0; -- 复位计数

                if (index = "1101") then
                    -- 当前是最后一个
                    index <= "0000"; -- 复位
                else
                    -- 否则
                    index <= index + 1; -- 下一个数字
                end if;
            end if;
        end if;
    end process;

    process(index)
    begin
        case index is
            when "0000" => display <= "1101101"; -- 2
            when "0001" => display <= "1111110"; -- 0
            when "0010" => display <= "0110000"; -- 1
            when "0011" => display <= "1110011"; -- 9
            when "0100" => display <= "1111110"; -- 0
            when "0101" => display <= "0110000"; -- 1
            when "0110" => display <= "1111110"; -- 0
            when "0111" => display <= "1011011"; -- 5
            when "1000" => display <= "0110011"; -- 4
```

```
when "1001" => display <= "1110000"; -- 7
when "1010" => display <= "1110011"; -- 9
when "1011" => display <= "1101101"; -- 2
when "1100" => display <= "1111110"; -- 0
when "1101" => display <= "1111111"; -- 8
when others => display <= "0000000";
end case;
end process;

end bhv;
```

2.2 实现原理

使用变量 `cnt` 记录时钟变化的次数，使用变量 `index` 记录当前显示的数字位。在一个以时钟信号 `clk` 和复位信号 `rst` 为敏感信号的 `process` 中，判断复位信号和时钟信号，当复位信号为 1，或距离上次复位或更新过了 1s 后，改变 `index`。

在另一个以 `index` 为敏感信号的 `process` 中，根据 `index` 的值改变输出信号，即可改变显示的数字。

3 实验总结

由于是第一次进行可编程器件实验，本次实验中大量时间花在了配置环境和学习语言上。通过课本的介绍，在这个过程中遇到的问题都逐一解决了，没有遇到太多的障碍。

实际实现时，遇到的问题主要是对硬件语言机制的不熟悉导致错误的做法，比如在多个 `process` 中同时更改某个 `signal` 的值。通过在 Stackoverflow 等网站进行搜索解决了相应的问题。