

计数器的设计 实验报告

2018011365 张鹤潇

一、实验目的

- 掌握时序电路的基本分析和设计方法；
- 理解同步时序电路和异步时序电路的区别；
- 掌握计数器电路的设计原理，用硬件描述语言实现指定功能的计数器设计；
- 利用软件实现对数字电路的逻辑功能进行验证分析。

二、实验内容

- 使用实验平台上两个未经译码处理的数码管显示计数，手动单次时钟进行计数，时钟上升沿计数一次；当计数到59时，要求恢复到00的状态。实验还要求设置一个复位键，可以随时重新恢复到00的状态继续计数。
- 借助实验平台上的1MHz时钟，将计数器改成秒表，并增加开关控制秒表的暂停和启动。

三、原理及代码实现

1. 计数器

用逻辑输入mode来控制计数器的模式。mode为0设置为手动计数，为1设置为秒表计数。

```
entity counter4 is
    port(
        clk, rst, pause: in std_logic;
        mode: in std_logic; -- 0: 手动 1:秒表
        d0, d1: buffer std_logic_vector(3 downto 0) -- 输出
    );
end counter4;

architecture count of counter4 is
    signal cnt: integer := 0;
begin
    process(clk, rst)
    begin
        if (rst = '0') then -- 复位
            d0 <= "0000";
            d1 <= "0000";
            cnt <= 0;
        elsif (clk'event and clk = '1' and pause = '0') then
            if (mode = '1') then -- 秒表计数
                if (cnt < 1000000) then
                    cnt <= cnt + 1;
                else

```

```

        cnt <= 0;
    end if;
end if;
if (mode = '0' or cnt = 0) then -- 手动计数
    if (d0 = "1001") then -- 更新输出
        d0 <= "0000";
        if (d1 = "0101") then
            d1 <= "0000";
        else
            d1 <= d1 + 1;
        end if;
    else
        d0 <= d0 + 1;
    end if;
end if;
end process;
end;

```

2. 数码管译码器

沿用了第一次实验的代码。

```

entity digit7 is
    port(
        digit: in std_logic_vector(3 downto 0);
        light_natural_7: out std_logic_vector(6 downto 0)
    );
end digit7;

architecture lighting of digit7 is
begin
    process(digit)
    begin
        case (digit) is
            when "0000" => light_natural_7<="1111110";
            when "0001" => light_natural_7<="0110000";
            when "0010" => light_natural_7<="1101101";
            when "0011" => light_natural_7<="1111001";
            when "0100" => light_natural_7<="0110011";
            when "0101" => light_natural_7<="1011011";
            when "0110" => light_natural_7<="1011111";
            when "0111" => light_natural_7<="1110000";
            when "1000" => light_natural_7<="1111111";
            when "1001" => light_natural_7<="1111011";
            when "1010" => light_natural_7<="1110111";
            when "1011" => light_natural_7<="0011111";
            when "1100" => light_natural_7<="1001110";
            when "1101" => light_natural_7<="0111101";
            when "1110" => light_natural_7<="1001111";
            when "1111" => light_natural_7<="1000111";
            when others => light_natural_7<="0000000";
        end case;
    end process;
end lighting;

```

```
end;
```

3. 顶层结构

顶层entity将计数器和数码管的接口整合在一起，实现模块化、结构化的设计。

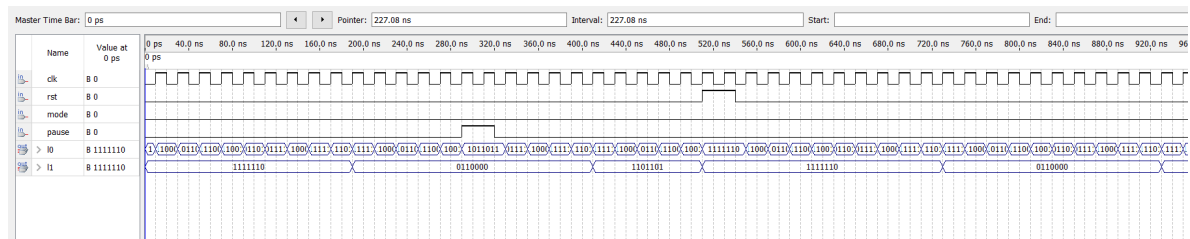
```
entity counter is
    port(
        clk, rst, pause: in std_logic;
        mode: in std_logic;
        l0, l1: out std_logic_vector(6 downto 0)
    );
end counter;

architecture count of counter is
    signal d0: std_logic_vector(3 downto 0) := "0000";
    signal d1: std_logic_vector(3 downto 0) := "0000";
    component counter4 -- 元件例化
    port(
        clk, rst, mode, pause: in std_logic;
        d0, d1: buffer std_logic_vector(3 downto 0)
    );
end component;
component digit7
    port(
        digit: in std_logic_vector(3 downto 0);
        light_natural_7: out std_logic_vector(6 downto 0)
    );
end component;
begin
    FF: counter4 port map(clk=>clk, rst=>rst, d0=>d0, d1=>d1, mode=>mode, pause=>pause);
    light0: digit7 port map(digit=>d0, light_natural_7=>l0);
    light1: digit7 port map(digit=>d1, light_natural_7=>l1);
end;
```

四、仿真实验

本机仿真

在本机上进行time simulation, 测试程序正确性。



JieLab实验

实验如图，左边三个开关分别控制复位、暂停和模式。

设置模式为秒表计时(mode=1)，数码管显示的数字每秒加1。

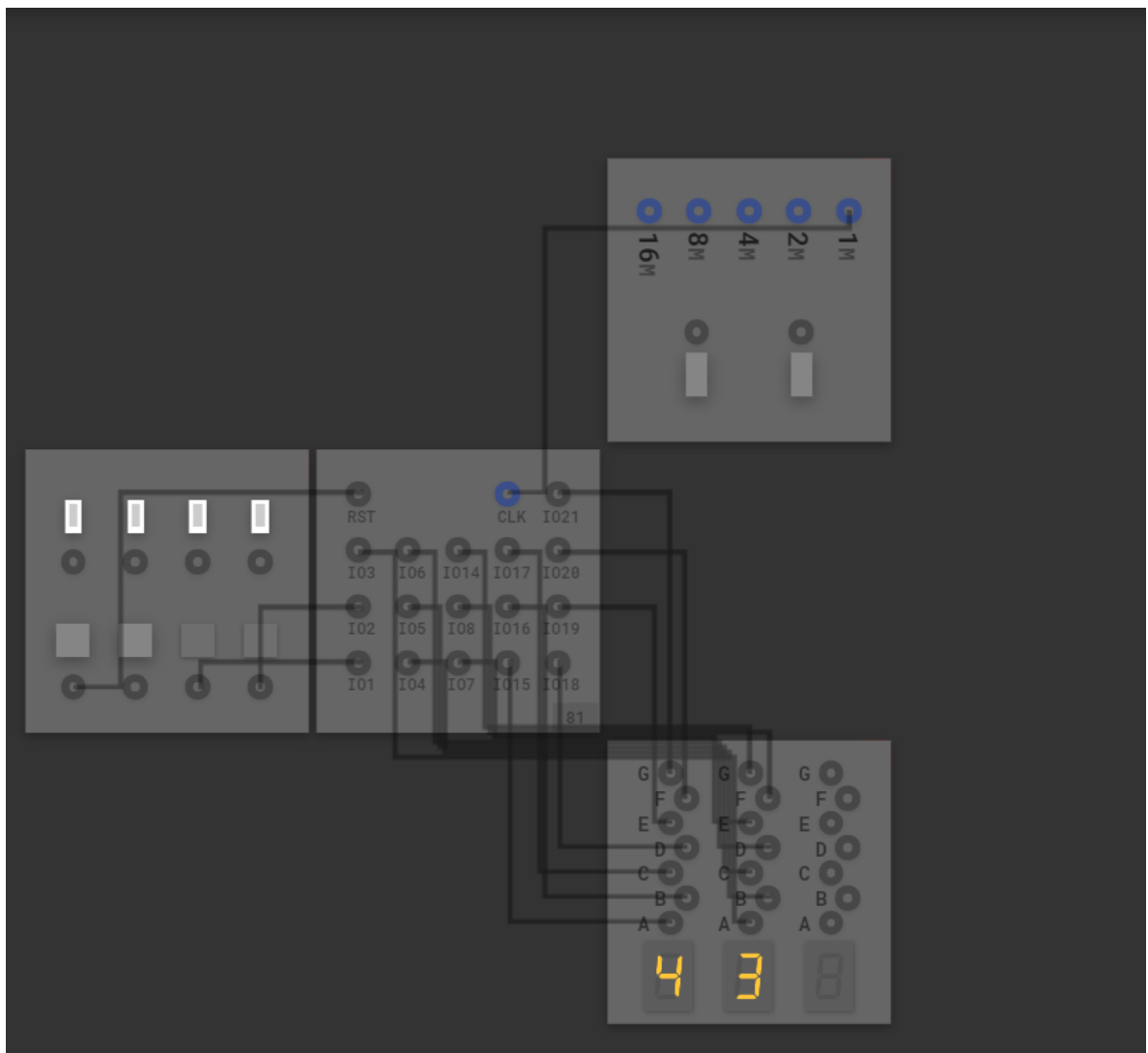
JieLabs 数电实验
#3973 ✔ Labview Build 1

```

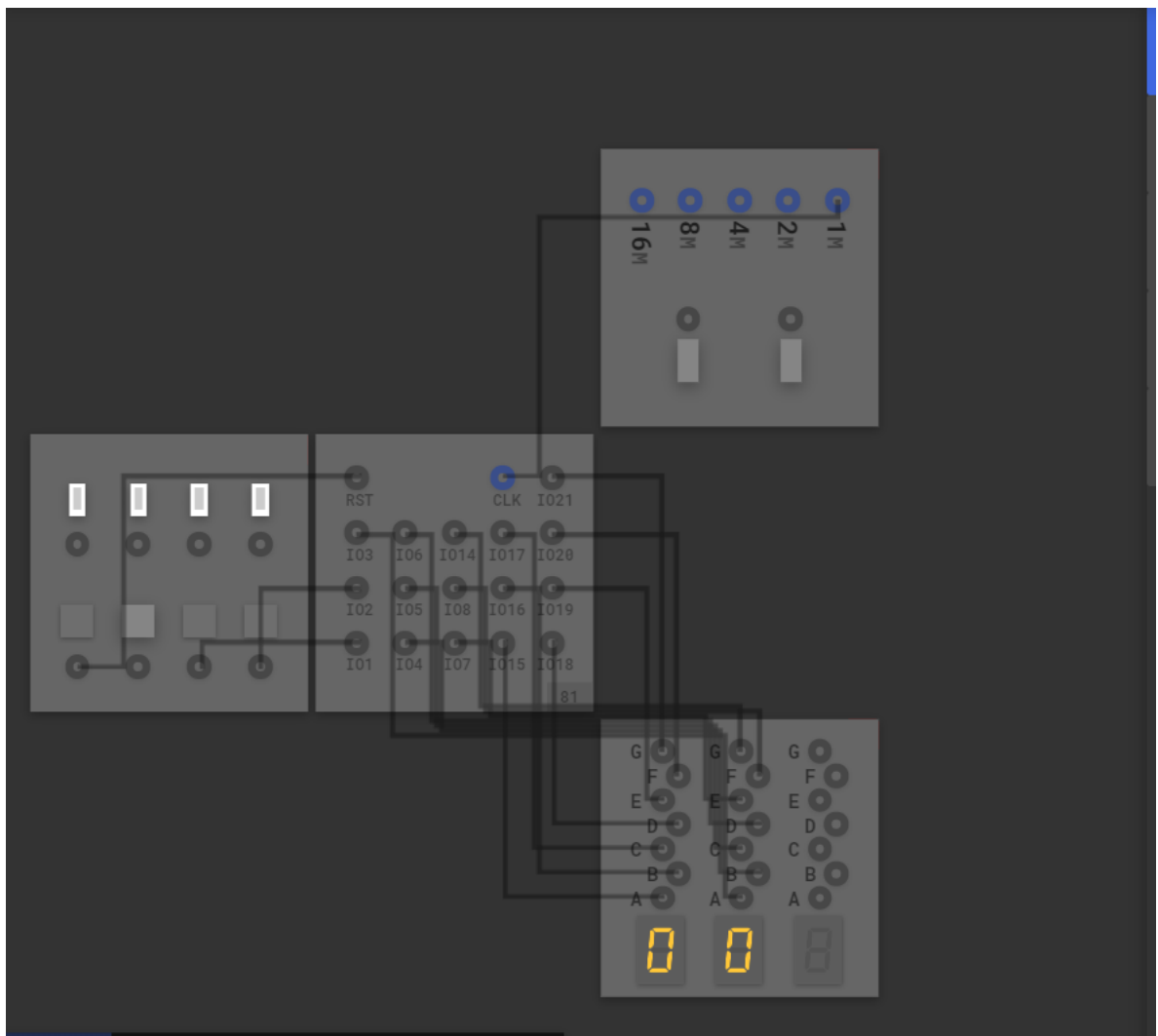
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  use ieee.std_logic_unsigned.all;
5
6  Set as top
7  entity counter4 is
8      port(
9          clk, rst, pause: in std_logic;
10         mode: in std_logic;
11         d0, d1: buffer std_logic_vector(3 downto 0)
12     );
13 end counter4;
14
15 architecture count of counter4 is
16     signal cnt: integer := 0;
17 begin
18     process(clk, rst)
19     begin
20         if (rst = '1') then
21             d0 <= "0000";
22             d1 <= "0000";
23             cnt <= 0;
24         elsif (clk'event and clk = '1' and pause = '0') then
25             if (mode = '1') then -- 秒表计数
26                 if (cnt < 1000000) then
27                     cnt <= cnt + 1;
28                 else
29                     cnt <= 0;
30                 end if;
31             end if;
32             if (mode = '0' or cnt = 0) then -- 手动计数
33                 if (d0 = "1001") then

```

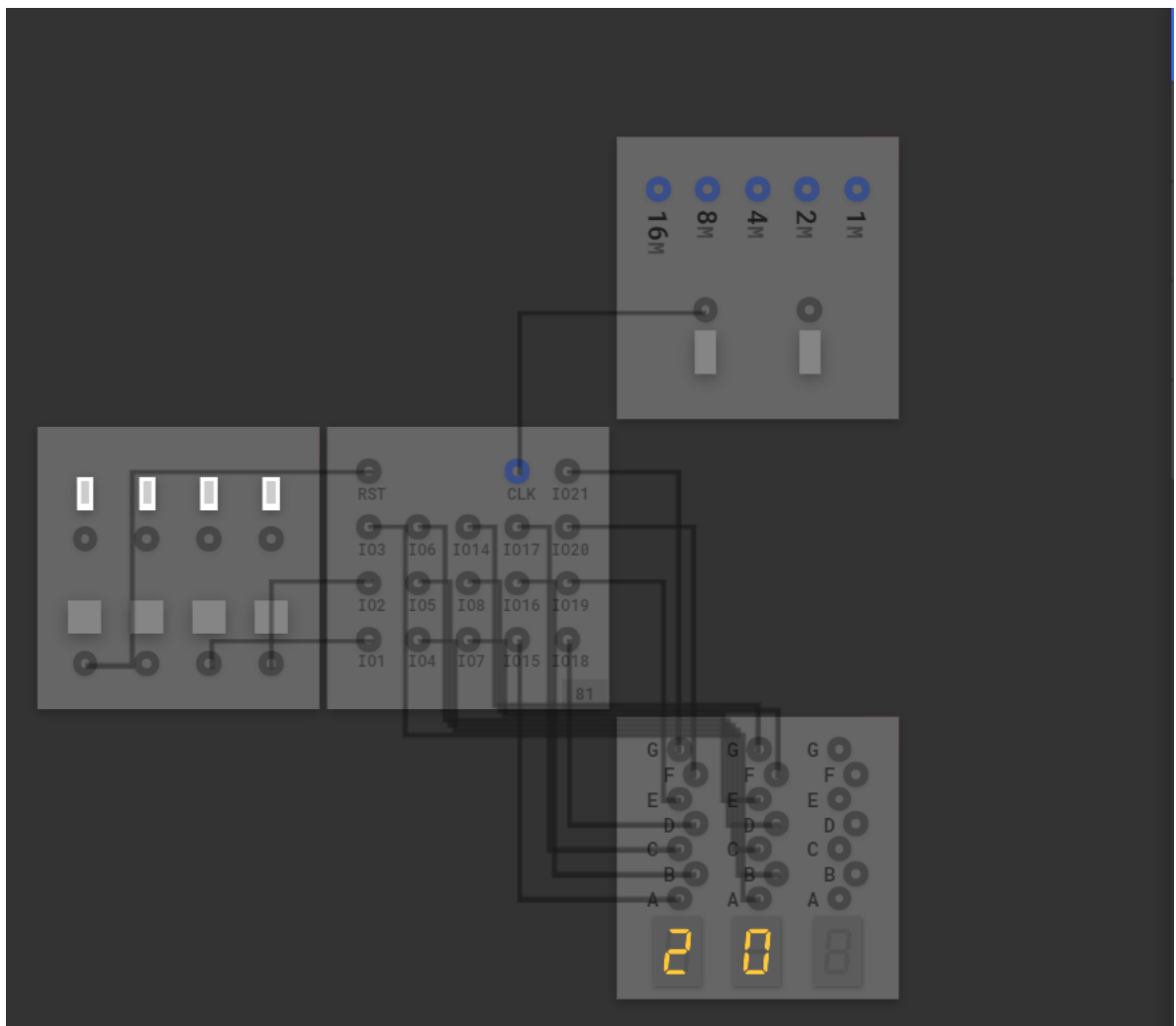
设置pause为1，暂停计数。



设置rst为1，计数器复位。



更改模式，进行手动计数(mode=0)。



五、总结反思

这是我第三次进行CPLD 实验，在之前基础上又有了提高。这次我没有用configuration语句控制功能，而是通过一个逻辑输入mode来控制，这避免了二次编译，也方便了测试和仿真。