计数器实验报告

计 71 张程远 2017011429

一、实验目的

1、掌握时序电路的基本分析和设计方法

2、理解同步时序电路和异步时序电路的区别

3、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

4、学会利用软件仿真实现对数字电路的逻辑功能进行验证分析

二、实验内容

1、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 的时候，要求两个数码管都能复位到 00 的状态，重新计数；实验还要求设置复位键 rst，可以随时重新恢复到 00 的状态继续计数。

2、使用实验平台上的 1MHz 时钟，将计数器改成秒表，并可以在秒表中使用开关控制秒表的启动和暂停。

三、实验代码

1、D 触发器

```
library IEEE ;
use IEEE.STD_LOGIC_1164.ALL ;

entity Dtrigger is
    port(clk, rst: IN STD_LOGIC ;
    q, notq: OUT STD_LOGIC
    ) ;
end Dtrigger ;

architecture arc of Dtrigger is
signal tmp: STD_LOGIC := '0' ;
begin
    process(clk, rst)
    begin
        if(rst = '0') then
            tmp <= '0' ;
        elsif (clk'event) and (clk = '1') then
            tmp <= not tmp ;
        end if ;
    end process ;
    process(tmp)
    begin
        q <= tmp ;
        notq <= not tmp ;
    end process ;
end arc ;
```

2、Counter 部分

```
library IEEE ;
use IEEE.STD_LOGIC_1164.ALL ;
use IEEE.STD_LOGIC_ARITH.ALL ;
use IEEE.STD_LOGIC_UNSIGNED.ALL ;

entity Counter is
    port(c_clk, c_rst: IN STD_LOGIC ;
    outh, outl: OUT STD_LOGIC_VECTOR(6 downto 0)
    ) ;
end Counter ;

architecture arcc of Counter is
component Dtrigger -Dtrigger作为元件例化
    port(clk, rst: IN STD_LOGIC ;
        q, notq: OUT STD_LOGIC
        ) ;
end component ;
signal h, l:INTEGER range 0 to 9 := 0 ; --十位与个位
signal q_v, notq_v: STD_LOGIC_VECTOR(5 downto 0); --Q和非Q
signal t_rst, t_clk: STD_LOGIC := '1' ;
signal num : INTEGER range 0 to 60:= 0 ; --计数

begin
    s0: D port map(t_clk, t_rst, q_v(0), notq_v(0)) ;
    s1: D port map(nq_v(0), t_rst, q_v(1), notq_v(1)) ;
    s2: D port map(nq_v(1), t_rst, q_v(2), notq_v(2)) ;
    s3: D port map(nq_v(2), t_rst, q_v(3), notq_v(3)) ;
    s4: D port map(nq_v(3), t_rst, q_v(4), notq_v(4)) ;
    s5: D port map(nq_v(4), t_rst, q_v(5), notq_v(5)) ;

    process(c_clk, c_rst)
    begin
        t_rst <= c_rst ;
        t_clk <= c_clk ;
        num <= CONV_INTEGER(q_v) ;
        if(num > 59) then
```

```
            t_rst <= '0' ;
            num <= 0 ; --置零
        end if ;
        h <= num / 10 ;
        l <= num mod(10) ; --计算十位与个位
        --译码
        case h is
            when 0 => outh <= "1111110" ;
            when 1 => outh <= "0110000" ;
            when 2 => outh <= "1101101" ;
            when 3 => outh <= "1111001" ;
            when 4 => outh <= "0110011" ;
            when 5 => outh <= "1011011" ;
            when 6 => outh <= "0011111" ;
            when 7 => outh <= "1110000" ;
            when 8 => outh <= "1111111" ;
            when 9 => outh <= "1110011" ;
            when others => outh <= "0000000" ;
        end case ;

        case l is
            when 0 => outl <= "1111110" ;
            when 1 => outl <= "0110000" ;
            when 2 => outl <= "1101101" ;
            when 3 => outl <= "1111001" ;
            when 4 => outl <= "0110011" ;
            when 5 => outl <= "1011011" ;
            when 6 => outl <= "0011111" ;
            when 7 => outl <= "1110000" ;
            when 8 => outl <= "1111111" ;
            when 9 => outl <= "1110011" ;
            when others => outl <= "0000000" ;
        end case ;
    end process ;
end arcc ;
```

原理：首先实现底层的D触发器作为例化元件，在Counter的实现中采用6个计数器进行计数，并用num计算时钟上升沿的次数，超过59立即置成0。计算num的十位和个位作为h和l的值，然后再实现译码，从而获得结果。

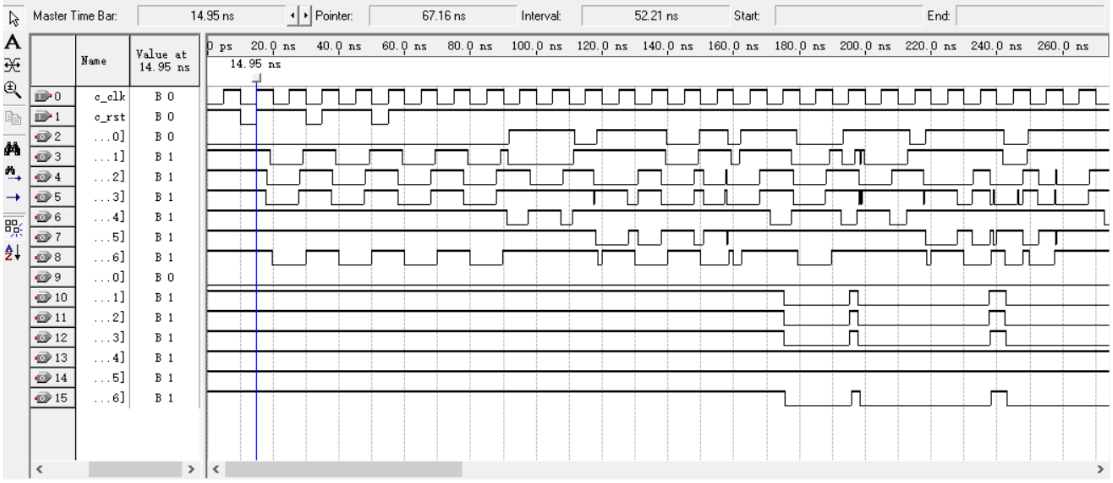
3、Timer部分

```
library IEEE ;
use IEEE.STD_LOGIC_1164.ALL ;
use IEEE.STD_LOGIC_ARITH.ALL ;
use IEEE.STD_LOGIC_UNSIGNED.ALL ;

entity Timer is
    port(c_clk, c_rst: IN STD_LOGIC ;
    outh, outl: OUT STD_LOGIC_VECTOR(6 downto 0);
    pause: IN STD_LOGIC
    ) ;
end Timer ;

architecture arct of Timer is
component Dtrigger
    port(clk, rst: IN STD_LOGIC ;
        q, notq: OUT STD_LOGIC
        ) ;
end component ;
signal h, l:INTEGER range 0 to 9 := 0 ; --同上
signal q_v, notq_v: STD_LOGIC_VECTOR(5 downto 0);
signal t_rst, t_clk: STD_LOGIC := '1' ;
signal num : INTEGER range 0 to 60:= 0 ;
signal t : INTEGER range 0 to 1000000 := 0 ;

begin
    u0: D port map(t_clk, t_rst, q_v(0), notq_v(0)) ;
    u1: D port map(nq_v(0), t_rst, q_v(1), notq_v(1)) ;
    u2: D port map(nq_v(1), t_rst, q_v(2), notq_v(2)) ;
    u3: D port map(nq_v(2), t_rst, q_v(3), notq_v(3)) ;
    u4: D port map(nq_v(3), t_rst, q_v(4), notq_v(4)) ;
    u5: D port map(nq_v(4), t_rst, q_v(5), notq_v(5)) ;

    process(c_clk, c_rst, pause)
    begin
        t_rst <= c_rst ;
        if(c_clk'event) and (c_clk = '1') then
            if(pause = '0') then
                t <= t+1 ;
                t_clk <= '1' ;
                if(t = 1000000) then --1M时钟
                    t_clk <= '0' ;
                    t <= 0 ;
                end if ;
            end if ;
        end if ;
        num <= CONV_INTEGER(q_v) ;
        h <= num / 10 ;
        l <= num mod 10 ;

        if(num > 59) then
            num <= 0 ;
            t_rst <= '0' ;
            t_clk <= '1' ;
            t <= 0 ;
            h <= 0 ;
            l <= 0 ;
        end if ;

        case h is
            when 0 => outh <= "1111110" ;
            when 1 => outh <= "0110000" ;
            when 2 => outh <= "1101101" ;
            when 3 => outh <= "1111001" ;
            when 4 => outh <= "0110011" ;
            when 5 => outh <= "1011011" ;
            when 6 => outh <= "0011111" ;
            when 7 => outh <= "1110000" ;
            when 8 => outh <= "1111111" ;
            when 9 => outh <= "1110011" ;
            when others => outh <= "0000000" ;
        end case ;

        case l is
            when 0 => outl <= "1111110" ;
            when 1 => outl <= "0110000" ;
            when 2 => outl <= "1101101" ;
            when 3 => outl <= "1111001" ;
            when 4 => outl <= "0110011" ;
            when 5 => outl <= "1011011" ;
            when 6 => outl <= "0011111" ;
            when 7 => outl <= "1110000" ;
            when 8 => outl <= "1111111" ;
            when 9 => outl <= "1110011" ;
            when others => outl <= "0000000" ;
        end case ;
    end process ;
end arct ;
```

原理：利用pause作为时钟的暂停键，模块1MHz的时钟频率代替手动触发的clk信号，因此设置一个计数用的变量t，每当时钟上升沿时t就加1，加到1000000的时候意味着经过

了一秒，此时计数值加1。

4、手动计数器仿真



Master Time Bar: 14.95 ns Pointer: 67.16 ns Interval: 52.21 ns Start: End:

	Name	Value at 14.95 ns
0	c_clk	B 0
1	c_rst	B 0
2	...0]	B 0
3	...1]	B 1
4	...2]	B 1
5	...3]	B 1
6	...4]	B 1
7	...5]	B 1
8	...6]	B 1
9	...0]	B 0
10	...1]	B 1
11	...2]	B 1
12	...3]	B 1
13	...4]	B 1
14	...5]	B 1
15	...6]	B 1

上面的 7 位数字是个位的仿真结果，下面的则是十位仿真结果。同时，给 reset 设置了 3 个置零操作，以观察 reset 的效果。从上面的波形图可以看出，在忽略仿真延迟的情况下，仿真结果基本符合理论值。

四、实验小结

通过本次实验，我对设计时序电路有了更丰富的经验和更深刻的理解。在实验中我遇到了几个问题：首先是 1MHz 时钟模块频率过快，t 以 100 万进行计数时数字跳变速度很快，经过我仔细检查代码，认为可能是模块本身的问题，因此我借用了其他同学的模块进行实验，获得了正常秒表的效果；另外一个问题是 mod 开销过大的问题，最开始时我使用了两处 mod，然而烧出的代码总是出现奇怪的 bug，推测是 mod 消耗的资源和时间过长造成的。后来我只在十位的计算里使用了一处 mod，终于获得了较好的效果。

由此看来，时序逻辑电路本身还有很多值得学习和思考的东西，也许这意味着将来还会在这里“踩坑”，但只有踩到坑才会有所进步，所以这也未尝不是一件好事。

感谢助教的指导和讲解！