



南京大學

数电实验四：计数器和时钟

课程名称： 数字逻辑与计算机组成实验

姓名： 孙文博

学号： 201830210

班级： 数电一班

邮箱： 201830210@smail.nju.edu.cn

实验时间： 2022. 3. 23

一、实验目的

1. 复习数字电路中计数器的工作原理；
2. 了解几种简单计数器的工作过程和设计方法；
3. 熟悉开发板上系统时钟的使用；
4. 设计一个 0-99 计时器及电子时钟并通过七段数码管显示。

二、实验环境

设计\编译环境：Quartus (Quartus Prime 17.1) Lite Edition

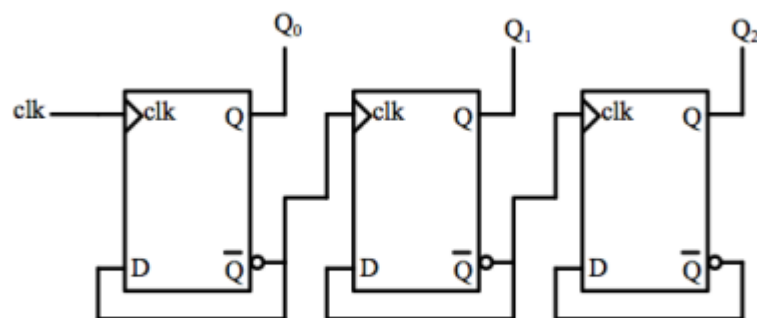
开发平台：DE10-Standard

FPGA 芯片：Cyclone II 5CSXFC6D6

三、实验原理

1. 加法计数器

最简单的计数器由时序逻辑电路中的触发器构成，如图 1 是由 3 个上升沿触发的 D 触发器组成的 3 位二进制异步加法计数器，即在每个 Clock 的上升沿，计数器输出 Q_2 Q_1 Q_0 加 1。



图表 1：3 位二进制加法器

根据上面的电路图，设计得到的 Verilog 代码如下，其逻辑不难理解，即每当 clk 上升沿来临时计数 out_q 自增 1，计数到 111 时由于溢出自动清零。

```
//加法计数器
module my_exp(clk,en,out_q);
    input clk;
    input en;
    output reg [2:0] out_q;

    always @ (posedge clk)
        if (en)
            out_q <= out_q + 1;
        else
            out_q <= 0;
endmodule
```

图表 2：3 位二进制带使能端的加法计数器代码

2. 开发板时钟信号

DE-10 Standard 开发板为 Cyclone V SOC FPGA 提供了四个频率为 50MHz 的外部输入时钟，这些时钟均可供使用，将此时钟信号作为计数器的时钟信号，即可构成一个定时器，原理如下：

$$\text{计时时间} = \text{脉冲个数} \times \text{脉冲周期}$$

四、实验过程

1. 分频器

开发板上的时钟频率为 50MHz，我们需要先把它改装成一个 1Hz 的时钟信号，一种方法如下：

```
always @(posedge clk)
begin
    if(count_clk==49999999)
    begin
        count_clk <= 0;
        clk_1s <= ~clk_1s;|
    end
    else
        count_clk <= count_clk+1;
    end
end
```

图表 3：分频器

其中 count_clk 变量的作用是记录开发板时钟信号 clk 的脉冲个数，当到达 49999999 时说明时间过去了 1 秒（这里 clk 时钟频率为 50 MHz），因此 clk_1s 取反，从而封装出一个 1Hz 的时钟信号 clk_1s。

2. 0-99 计数器

使用我们 1Hz 的时钟信号设计一个 0-99 计数器，要求实现开始、暂停和清零功能，计数到 99 时自动清零，且让一个发光二极管闪烁一个时钟周期，提示计时结束，将数字显示在两个七段数码管上。

根据要求，我们的输入端包括开发板时钟 clk，开始按键 start，暂停按键 start，清零按键 start，输出端包括计时结束标志 RCO，两个七段数码管 LED_0，LED_1。

```
module exp_4(
    input clk, //50MHZ
    input start,
    input pause,
    input rst,
    output RCO,
    output [6:0] LED_0,
    output [6:0] LED_1
);
```

图表 4：输入端与输出端

接下来分为三个模块，分别是分频器封装时钟信号，cnt 记录当前数值以及根据 cnt 的值在数码管上显示，但由于分开写为三个部分时出现 bug，最终的代码实现中我们将三个部分放在同一个 always 语块中：

```
always @(posedge clk)
begin
    if(count_clk==49999999)
    begin
        count_clk = 0;
        if(rst==1) cnt=0;
        else begin
            if(pause==0) begin
                if(cnt==99) begin
                    cnt=0;
                    rco=1;
                end
                else begin
                    cnt=cnt+1;
                    rco=0;
                end
            end
        end
    end
end
```

图表 5：分频器及计数

上面的代码负责根据开发板时钟信号计数 count_clk 判断是否到 1 秒，若到了 1 秒（也即一个周期），cnt 自增一，rco 置为 0 或 cnt 满 99 清零，同时 rco 点亮一个周期。

```

led_0=7'b1111111;
led_1=7'b1111111;
if(start) begin

    case(cnt%10)
    0: led_0=7'b1000000;
    1: led_0=7'b1111001;
    2: led_0=7'b0100100;
    3: led_0=7'b0110000;
    4: led_0=7'b0011001;
    5: led_0=7'b0010010;
    6: led_0=7'b0000010;
    7: led_0=7'b1111000;
    8: led_0=7'b0000000;
    9: led_0=7'b0010000;
    endcase

    case(cnt/10)
    0: led_1=7'b1000000;
    1: led_1=7'b1111001;
    2: led_1=7'b0100100;
    3: led_1=7'b0110000;
    4: led_1=7'b0011001;
    5: led_1=7'b0010010;
    6: led_1=7'b0000010;
    7: led_1=7'b1111000;
    8: led_1=7'b0000000;
    9: led_1=7'b0010000;
    endcase

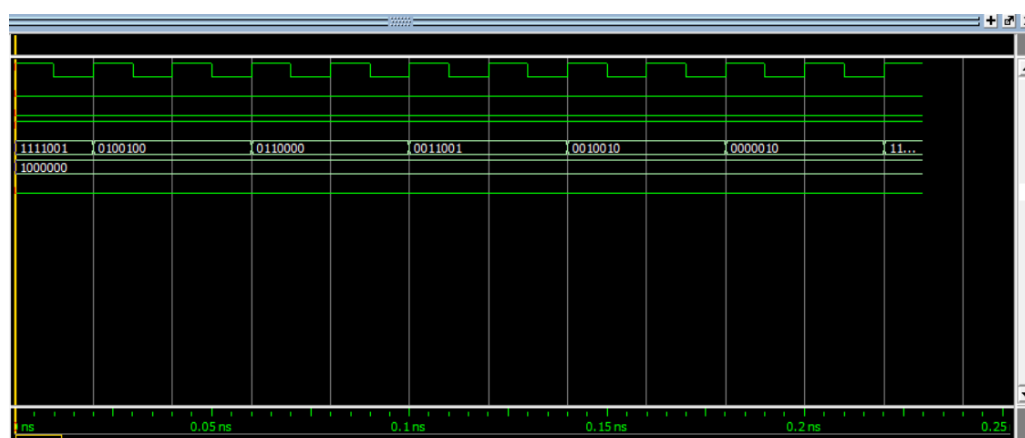
end

```

图表 6：七段数码管显示数字

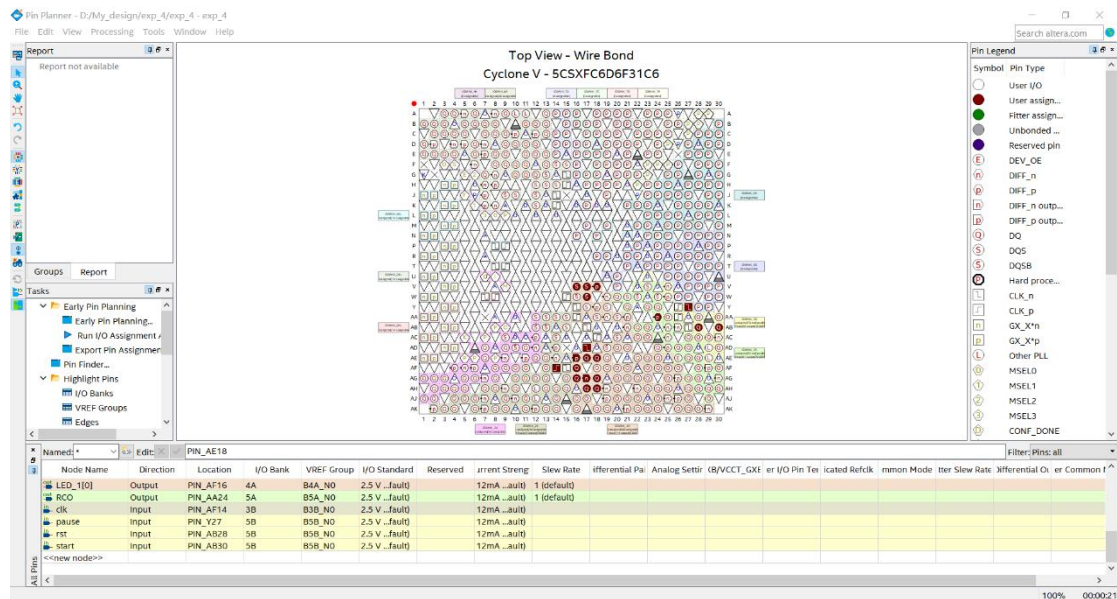
这部分代码负责将 cnt 的值显示到两个七段数码管上，其中十位 cnt/10 和个位 cnt%10 分开判断。

根据以上代码设计 testbench 仿真检测结果如图，符合预期：



图表 7：testbench 仿真检测

最后进行引脚分配及芯片烧录，等待上板验收。



图表 8：引脚分配

3. 电子时钟

在计数器的基础上我们可以改进为一个电子时钟，显示时分秒，并可以通关开关调整时间，这里给出实现的 Verilog 代码：

```

always @(posedge clk)
begin
    if(choice!=0) begin
        if(count_clk==49999999)
        begin
            count_clk = 0;

            if(choice==1)
                minute=data_0+data_1*10;
            if(choice==2)
                hour=data_0+data_1*10;
            if(choice==3) begin

                if(second==59) begin
                    second=0;
                    minute=minute+1;
                    if(minute==59) begin
                        minute=0;
                        hour=hour+1;
                        if(hour==23)
                            hour=0;
                    end
                end
            end
        else
            second=second+1;
        end
    end
end

```

其中 choice 开关控制时钟的调整时间，可以将分钟或者小时调整到我们输入的时间，调整后时钟正常运行。

五、实验结果

1. 思考题

在实验手册 4.3.2 中给出了一种生成 1Hz 时钟信号的方法，其代码如下：

表 4-2: 1 秒时钟生成代码

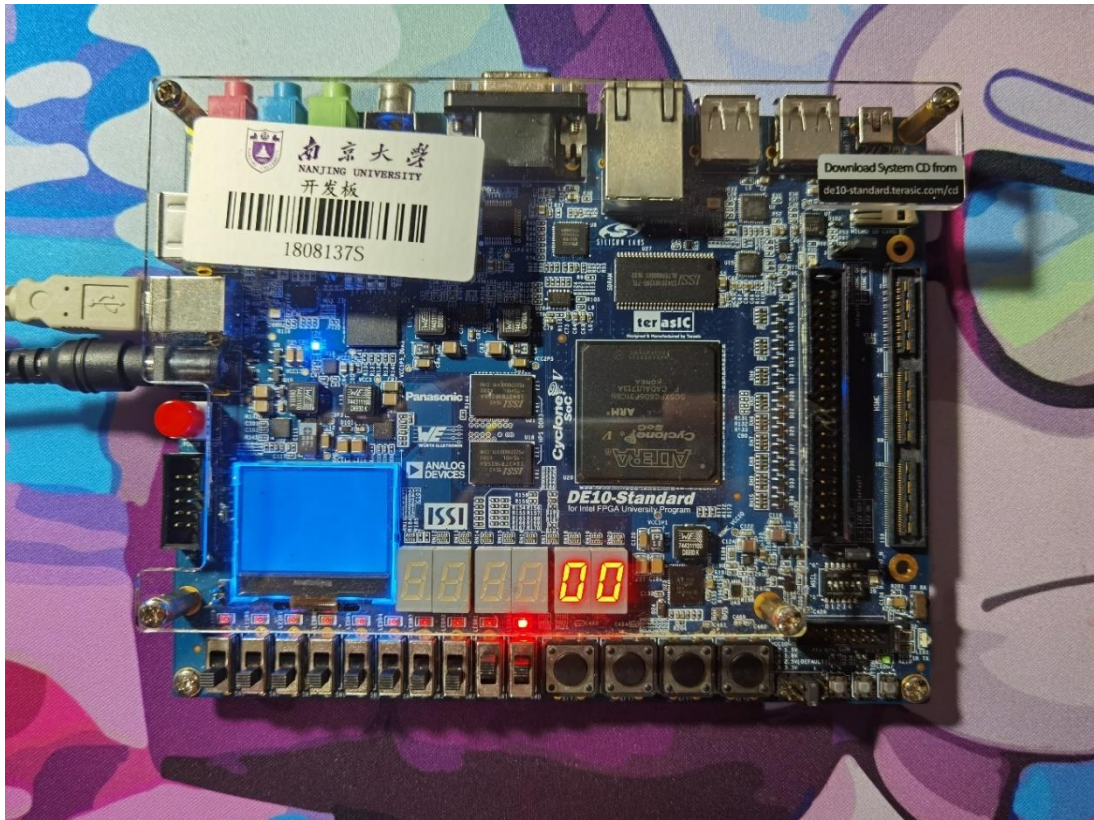
```
1 always @(posedge clk)
2     if(count_clk==24999999)
3     begin
4         count_clk <=0;
5         clk_1s <= ~clk_1s;
6     end
7     else
8         count_clk <= count_clk+1;
```

其中 count_clk 变量的作用是记录开发板时钟信号 clk 的脉冲个数，当到达 24999999 时说明时间过去了 1 秒（这里 clk 时钟频率为 25 MHz），因此 clk_1s 取反。为满足 0~24999999 的计数要求，变量 count_clk 的宽度至少为 $\log_2 24999999 \approx 24.5$ ，即位宽至少取 25 位。

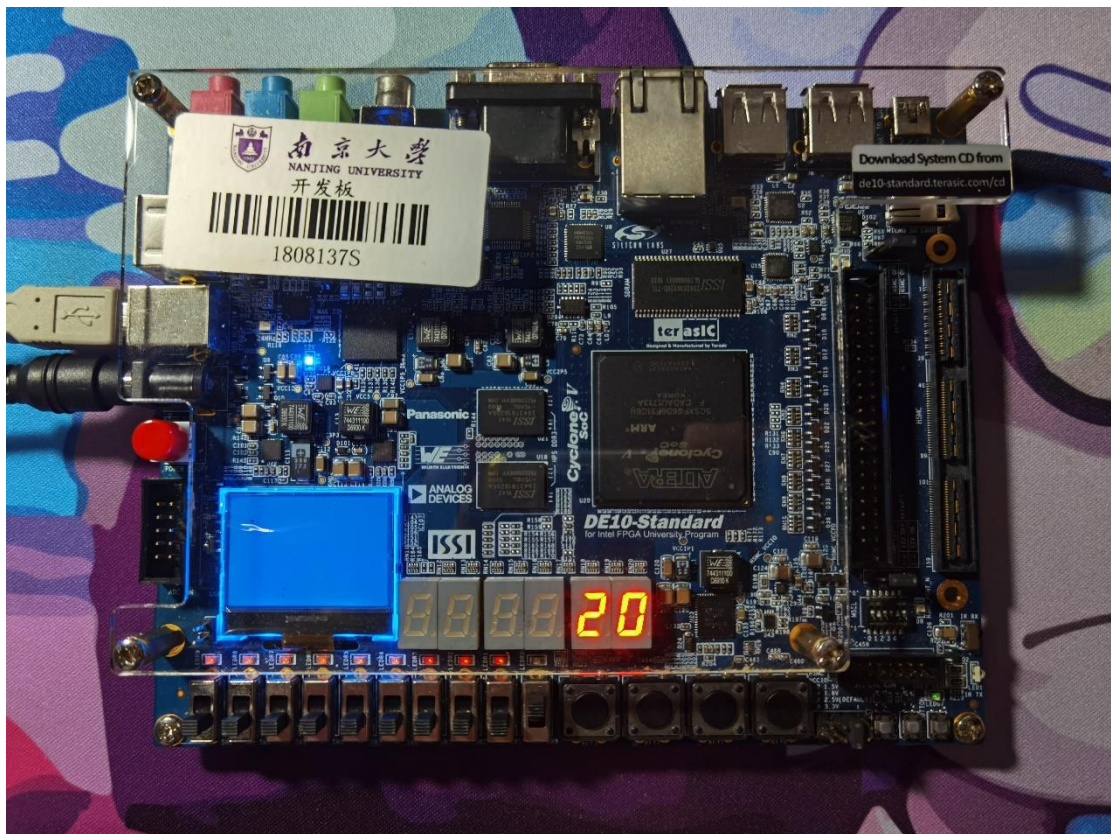
2. 上板验收（基础）

我们实现的 1 Hz 时钟有三个控制端 start, pause 和 rst，分别对应前三个开关，对应的功能是开始，暂停和清零，正常运行时候数字每秒自增一，满 99 自动清零，同时 LED 灯闪烁一个周期，以表示计时完成。

完整版上板验收视频放在压缩包中同步上传至 cms 网站。



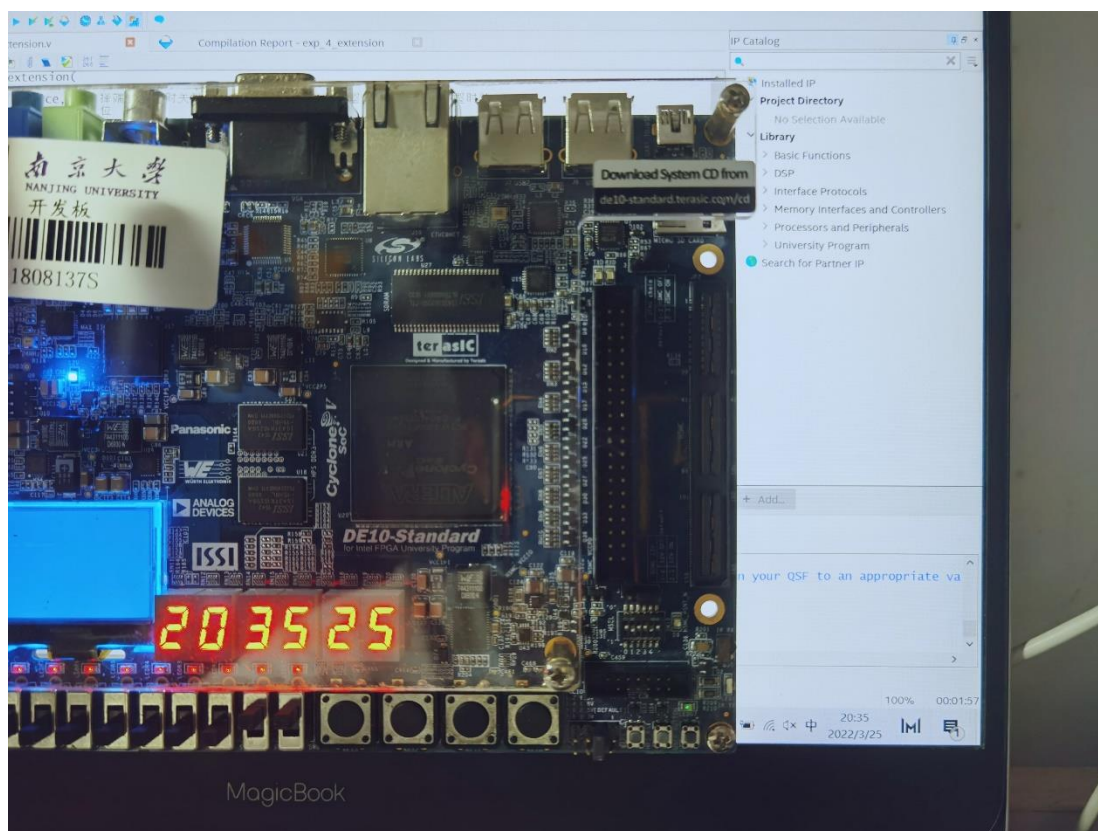
图表 10：满 99 自动清零，指示灯 RCO 亮起



图表 9：正常工作状态

3. 上板验收（拓展）

上板验收如图，调整时间为当前电脑时间后正常运行：



视频版本放在附件中提交到 cms 网站中。

六、总结与反思

本次实验中我们掌握了时序电路的实现过程，并改造开发板上自带的时钟成为我们自己的 1Hz 时钟，最后将其封装为一个可以调整时间的电子时钟，理论上等价于我们日常用的电子时钟（但是比它里面的电路元件复杂很多），另外我们对该时钟进行了长达 24h 的测试时间，发现一天会有大约 2min 的延迟，应该是由于电路内部部件的原因，总体来说实验四相对顺利！✧