

# 实验五

## 计数器和时钟的设计

实验报告

日期：2020 年 10 月 15 日

姓名：朱嘉琦

学号：191220185

班级：数电实验一班

邮箱：1477194584@qq.com



# 实验五报告——计数器和时钟

191220185 朱嘉琦

## 一、实验目的

本实验的目的是复习计数器的工作原理，通过介绍几种简单计数器的工作过程和设计方法、以及开发板系统时钟的使用，学习计数器的设计和定时器的工作原理。

## 二、实验原理

### 1. 加法计数器

利用触发器可以构成简单的计数器。图5-1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  $Q_2Q_1Q_0$  加1。

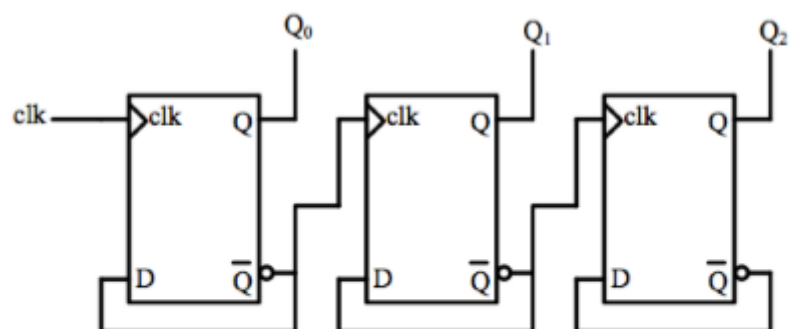


图 5-1: 3 位二进制加法计数器

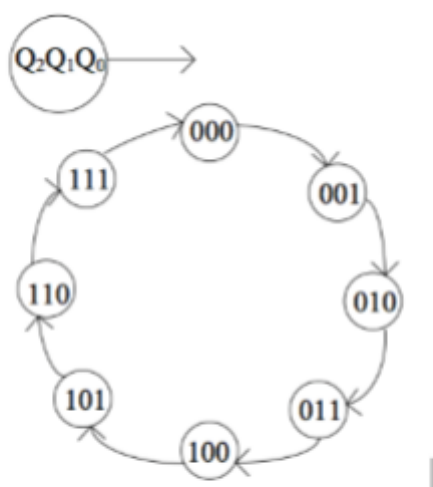


图 5-2: 3 位二进制加法计数器状态图

### 2. 减法计数器

利用D触发器同样可以构成减法计数器，图5-3是由3个上升沿触发的D触发器组成的3位二进制异步减法计数器 图5-4是此3位二进制异步减法计数器的状态转移图。

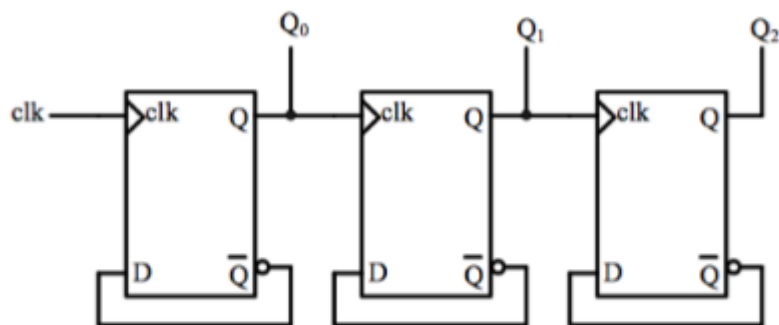


图 5-3: 3 位二进制异步减法计数器

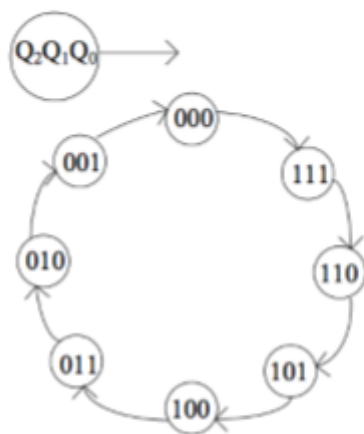


图 5-4: 3 位二进制减法计数器状态图

### 3. 定时器

如果在计数器的时钟输入端输入一个固定周期的时钟，那么计数器就变成了定时器。

DE-10开发板提供了四个频率位 50MHz 的外部输入时钟，这些时钟均可供用户使用。另外还给开放平台上的HPS 提供了一个25MHz 的时钟。

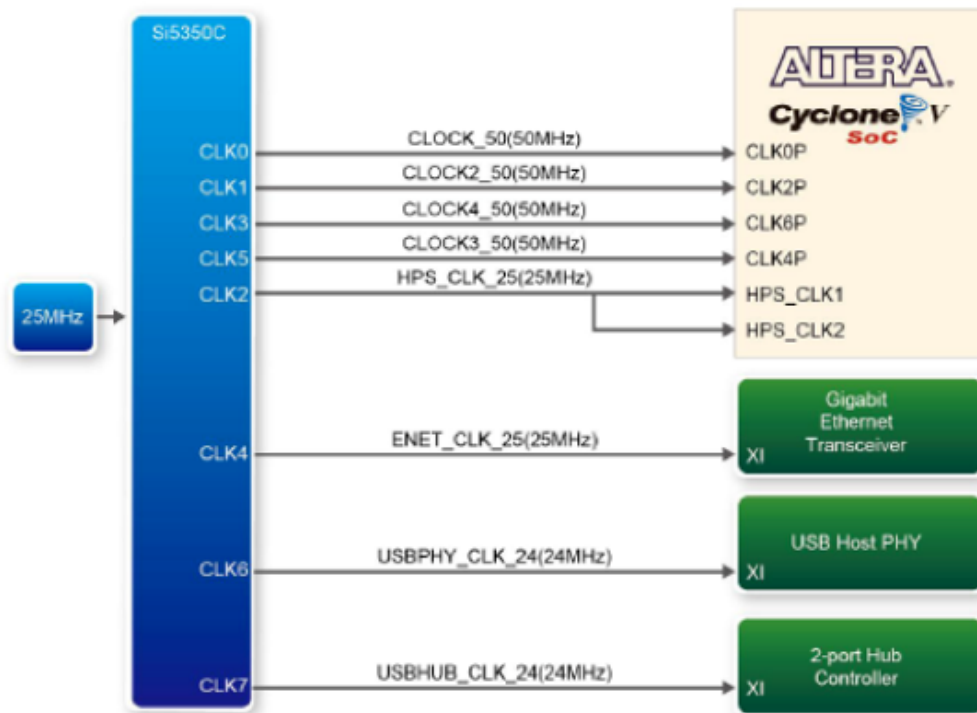


图 5-6: 开发板时钟连接

### 三、实验环境/器材

实验环境是Quartus 17.1 Lite，实验器材是DE10 开发板

### 四、程序代码或流程图

#### 1. 计时器

7段码编码器

```

module decoder(A,B);
    input [3:0]A;
    output reg[6:0]B;
    always @(*) begin
        case(A)
            0: B<=7'b1000000;           //0
            1: B<=7'b1111001;           //1
            2: B<=7'b0100100;           //2
            3: B<=7'b0110000;           //3
            4: B<=7'b0011001;           //4
            5: B<=7'b0010010;           //5
            6: B<=7'b0000010;           //6
            7: B<=7'b1111000;           //7
            8: B<=7'b0000000;           //8
            9: B<=7'b0010000;           //9

            default: B<=7'b1000000;     //0

        endcase
    end
endmodule

```

分频器

```

module divider(clk, clk_1hz); //divide the clk
    input clk;
    output reg clk_1hz=0;
    reg[36:0] count_clk=0;
    always@(posedge clk)
    begin
        if(count_clk==25000000) begin
            count_clk<=0;
            clk_1hz=~clk_1hz;
        end
        else
            count_clk=count_clk+1;
        end
    end
endmodule

```

计时器

```

module timer(clk,start,pause,reset,out0,out1,isEnd);
    input clk,start,pause,reset;
    output [6:0]out0,out1;
    output reg isEnd=0;
    reg [3:0]a0=0;
    reg [3:0]a1=0;
    wire clk_1s;
    divider Div(clk,clk_1s);

    always@(posedge clk_1s) begin
        if(reset==1) begin
            a0<=0;
            a1<=0;
        end else if(pause==1) begin
            a0<=a0;
            a1<=a1;
        end else if(start==1) begin
            if(a0==9)begin
                if(a1==9)begin
                    a0<=0;
                    a1<=0;
                    isEnd<=1;
                end
                else begin
                    a0<=0;
                    a1<=a1+1;
                    isEnd=0;
                end
            end
            end
        else begin
            a0<=a0+1;
            isEnd=0;
        end
    end
    decoder D1(a0,out0);
    decoder D2(a1,out1);
endmodule

```

## 2. 电子时钟

systembuilder创建的工程文件代码

```

//=====
// This code is generated by Terasic System Builder
//=====
module MyClock(

    //////////// CLOCK ////////////
    input          CLOCK2_50,
    input          CLOCK3_50,
    input          CLOCK4_50,
    input          CLOCK_50,

    //////////// SW ////////////
    input [9:0]    SW,
    input          BTN,

    //////////// LED ////////////
    output [9:0]   LEDR,

    //////////// Seg7 ////////////
    output [6:0]   HEX0,
    output [6:0]   HEX1,
    output [6:0]   HEX2,
    output [6:0]   HEX3,
    output [6:0]   HEX4,
    output [6:0]   HEX5
);

//=====
// REG/WIRE declarations
//=====

clock myClock (CLOCK2_50,HEX4,HEX5,HEX2,HEX3,HEX0,HEX1,SW[9:2],BTN,SW[1:0]);

//=====
// Structural coding
//=====
endmodule

```

分频器和七段编码器和计数器一致，如上图

### 电子时钟

```

module clock(clk,phr0,phr1,pmin0,pmin1,psec0,psec1,set,confirm,type);
    //input clk,start,pause,reset,confirm;
    input clk,confirm;
    input [7:0]set;
    input [1:0]type;
    output [6:0]phr0,phr1,pmin0,pmin1,psec0,psec1;
    reg[3:0] hr0,hr1,min0,min1,sec0,sec1;
    wire clk_1s;
    reg isEndSec=0;
    reg isEndMin=0;
    divider div(clk,clk_1s);
    always@(posedge clk_1s) begin
        if(confirm==0) begin
            case(type)
                1:begin sec1=set[7:4]; sec0=set[3:0]; end
                2:begin min1=set[7:4]; min0=set[3:0]; end
                3:begin hr1=set[7:4]; hr0=set[3:0]; end
            default;;
            endcase
        end
        if(sec0==9)begin
            if(sec1==5)begin
                sec0=0;
                sec1=0;
                isEndSec=1;
                isEndMin=0;
            end
            else begin
                sec0=0;
                sec1=sec1+1;
                isEndSec=0;
                isEndMin=0;
            end
        end
        else begin
            sec0=sec0+1;
            isEndSec=0;
            isEndMin=0;
        end
    end
end

```

```

end
if(isEndSec==1) begin
    if(min0==9)begin
        if(min1==5)begin
            min0=0;
            min1=0;
            isEndMin=1;
        end
        else begin
            min0=0;
            min1=min1+1;
            isEndMin=0;
        end
    end
    else begin
        min0=min0+1;
        isEndMin=0;
    end
end

if(isEndMin==1) begin
    if(hr0==9) begin
        hr0=0;
        hr1=hr1+1;
    end
    else if(hr0==3) begin
        if(hr1==2) begin
            hr0=0;
            hr1=0;
        end
    end
    else hr0=hr0+1;
end
end

decoder D1(hr0,phr0);
decoder D2(hr1,phr1);
decoder D3(min0,pmin0);
decoder D4(min1,pmin1);
decoder D5(sec0,psec0);
decoder D6(sec1,psec1);

endmodule

```

## 五、实验步骤

- 根据实验讲义复习计数器的相关内容。
- 了解如何利用开发板上提供的频率为 50MHz 的时钟信号和定时器，设计任何需要的时钟信号。

表 5-2: 1 秒时钟生成代码

```

1 always @(posedge clk)
2     if(count_clk==25000000)
3         begin
4             count_clk <=0;
5             clk_1s <= ~clk_1s;
6         end
7     else
8         count_clk <= count_clk+1;

```



- 根据以上内容设计一个具有开始、暂停和清零功能的计时器，能从 00 计数到 99 并且在七段数码管上直接以十进制显示。
- 进行分析与综合，检查是否有语法错误，最终编译通过。
- 利用ModelSim进行功能仿真，编写testbench文件，进行仿真测试。
- 接着为计时器分配引脚。
- 进行编译，待编译通过后将二进制文件写入开发板。
- 在开发板上进行硬件验证。
- 利用计时器所写的部分代码改写，编写一个具有设置时间的电子时钟，直接利用system builder分配引脚。
- 进行分析与综合，检查是否有语法错误，最终编译通过。
- 利用ModelSim进行功能仿真，编写testbench文件，进行仿真测试。
- 进行编译，待编译通过后将二进制文件写入开发板。
- 在开发板上进行硬件验证。

## 六、测试代码

### 1. 激励代码




















通过在激励代码中枚举所有可能的情况，在仿真波形图中对结果进行验证

#### 计时器的激励代码

```
divider`i1 (
// port map - connection between master ports and signals/registers
.clk(clk),
.clk_1hz(clk_1hz)
);
initial
begin
// code that executes only once
// insert code here --> begin
clk=0; #100000000;
// --> end
//$display("Running testbench");
end
always
// optional sensitivity list
// @(event1 or event2 or .... eventn)
begin
// code executes for every event on sensitivity list
// insert code here --> begin
#1 clk=~clk;
//@eachvec;
// --> end
end
endmodule
```

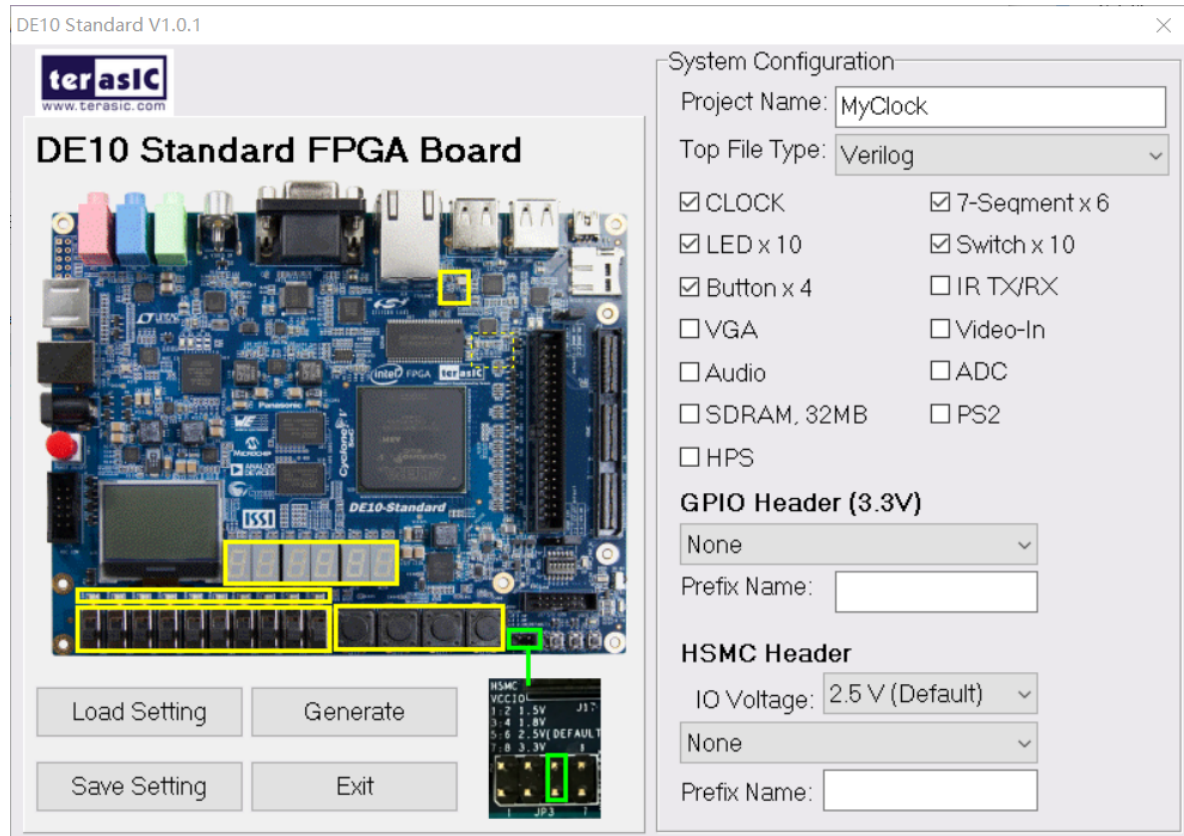
### 2. 引脚分配

#### 计时器的引脚分配

Node Name	Direction	Location	I/O Bank
 clk	Input	PIN_AF14	3B
 isEnd	Output	PIN_AA24	5A
 out0[6]	Output	PIN_AH18	4A
 out0[5]	Output	PIN_AG18	4A
 out0[4]	Output	PIN_AH17	4A
 out0[3]	Output	PIN_AG16	4A
 out0[2]	Output	PIN_AG17	4A
 out0[1]	Output	PIN_V18	4A
 out0[0]	Output	PIN_W17	4A
 out1[6]	Output	PIN_V17	4A
 out1[5]	Output	PIN_AE17	4A
 out1[4]	Output	PIN_AE18	4A
 out1[3]	Output	PIN_AD17	4A
 out1[2]	Output	PIN_AE16	4A
 out1[1]	Output	PIN_V16	4A
 out1[0]	Output	PIN_AF16	4A
 pause	Input	PIN_Y27	5B
 reset	Input	PIN_AB28	5B
 start	Input	PIN_AB30	5B

### 电子时钟的引脚分配

由于要使用6个七段码显示数字，手动分配引脚非常麻烦，所以利用system bulider作为工具直接分配引脚。



## 八、思考

无

## 九、实验中遇到的问题及解决方法

之前的实验由于所需要的接口都比较少，所以手动分配引脚都很方便，而在实现电子时钟的时候需要利用到6个七段码显示器，共计42个引脚，而且还要利用十个开关和部分按钮，如果每个都自己手动分配非常麻烦而且容易出错，于是学习了system builder的使用方法，一次性分配完毕而且保证正确性。

## 十、意见和建议

无