

数字电路与数字系统

实验报告

实 验： 计数器和时钟

姓 名： 周心同

学 号： 201220069

1.实验目的

复习计数器的工作原理

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

2.实验原理

2.1 计时器

利用触发器可以构成简单的计数器。

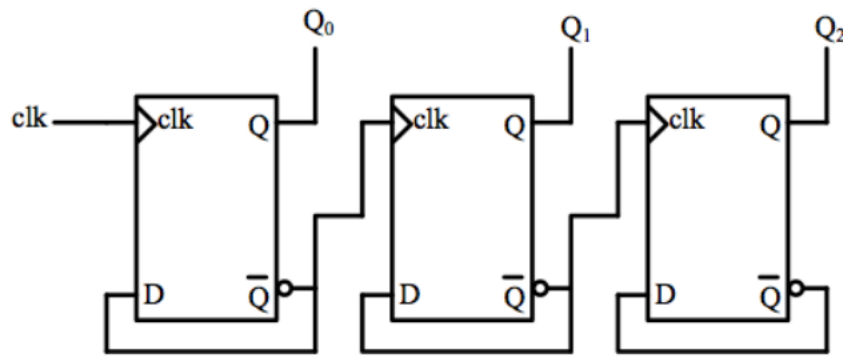


图 4-1: 3 位二进制加法计数器

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。

2.2 分频器

利用开发板上提供的频率为 50MHz 时钟信号和定时器，我们可以通过计数设计任何我们需要的时钟信号。

例如令计数为 24999999，就能将输入为 50MHz 的时钟，输出为一个频率为 1Hz，周期为 1 秒的时钟信号。

秒表也相当于一个分频器。

表 4-2: 1 秒时钟生成代码

```
always @(posedge clk)
    if(count_clk==24999999)
        begin
            count_clk <=0;
            clk_1s <= ~clk_1s;
        end
    else
        count_clk <= count_clk+1;
```

2.3 电子时钟，闹钟，秒表

三个时间电子时钟，闹钟，秒表，独立计算，独立表示。

当电子时钟时间为闹钟时，闪烁。

3.实验环境

- 软件环境

Quartus 17.1 Lite

- 硬件环境

开发板：DE10 Standard

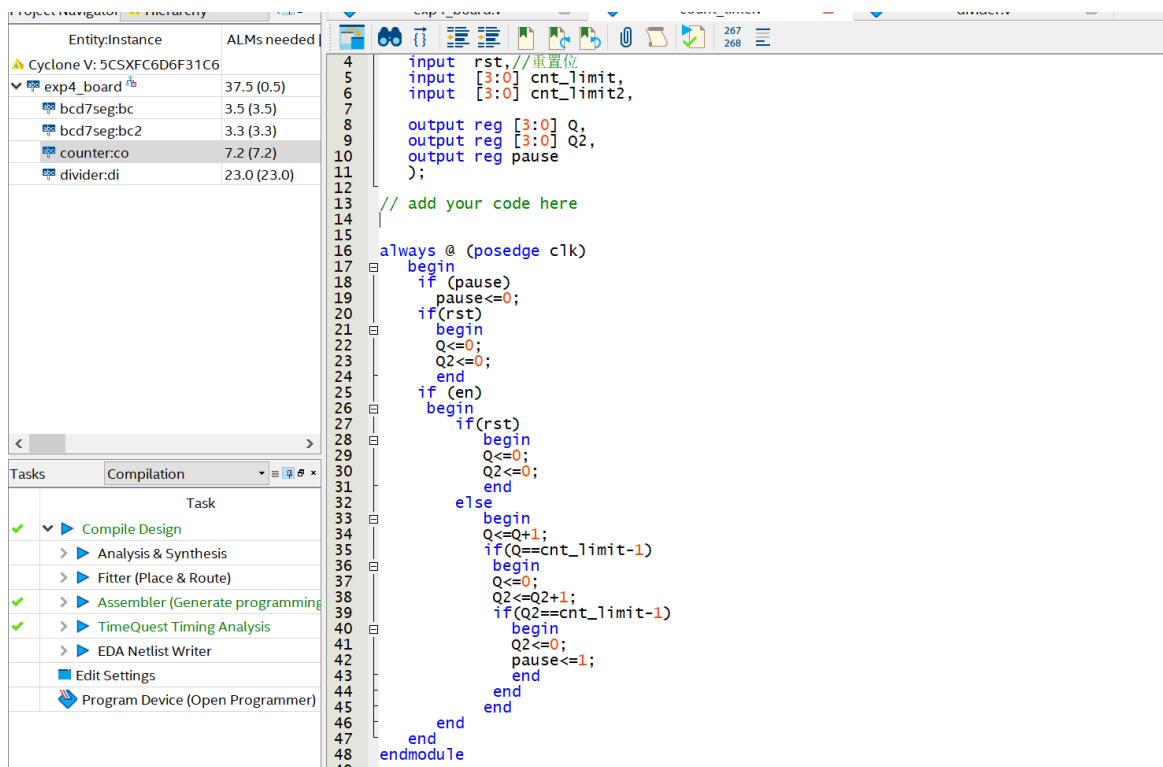
FPGA：Intel Cyclone V SE 5CSXFC6D6F31C6N

4.实验步骤和结果

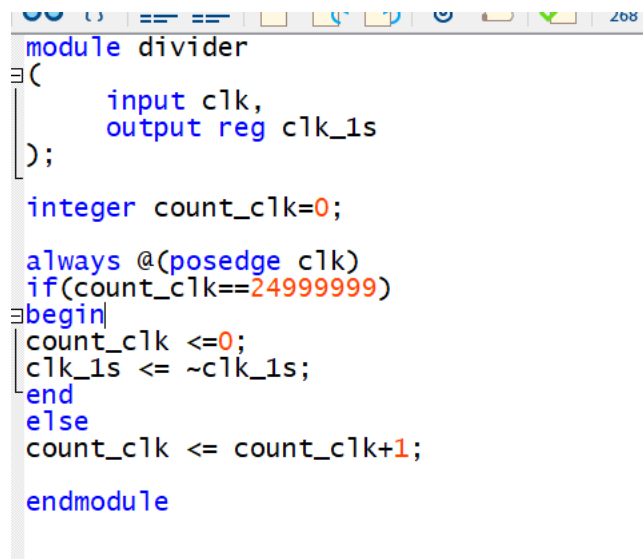
4.1 计时器

4.1.1 代码

计时器模块：



分频器模块：



```

module divider
(
    input clk,
    output reg clk_1s
);
    integer count_clk=0;
    always @(posedge clk)
    if(count_clk==24999999)
begin
count_clk <=0;
clk_1s <= ~clk_1s;
end
else
count_clk <= count_clk+1;
endmodule

```

接口：

```

module exp4_board(

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

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

    //////////// 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
//=====

wire clock;
divider di(CLOCK_50,clock);
wire [3:0] a;
wire [3:0] b;
counter co(clock,SW[0],SW[1],10,10,a,b,LEDR[0]);
bcd7seg bc(a,HEX0);
bcd7seg bc2(b,HEX1);

//=====
// structural coding
//=====

endmodule

```

4.1.1 仿真

由于波形图不能直观看结果，所以直接在上板仿真。测试计时 重置 暂停 三个功能。

4.2 电子时钟

4.2.1 代码

代码过长，下面贴主要部分。

```

module mux3_1 (
    input [1:0] Y,
    input clk,
    input [3:0] key,
    input en, //使能端
    input rst, //重置位

    output reg twinkle,
    output reg [3:0] a,
    output reg [3:0] a2,
    output reg [3:0] a3,
    output reg [3:0] a4,
    output reg [3:0] a5,
    output reg [3:0] a6
); //0:电子时钟,调整时间; 1:调闹钟(在特定时间 LED 闪烁); 2:秒表(提供百分之一秒精度,可以停止重启).
    reg [3:0] Q;
    reg [3:0] Q2;
    reg [3:0] Q3;
    reg [3:0] Q4;
    reg [3:0] Q5;
    reg [3:0] Q6;

    reg [3:0] q;
    reg [3:0] q2;
    reg [3:0] q3;
    reg [3:0] q4;
    reg [3:0] q5;
    reg [3:0] q6;

    reg [3:0] m;
    reg [3:0] m2;
    reg [3:0] m3;
    reg [3:0] m4;
    reg [3:0] m5;
    reg [3:0] m6;

    integer count_clk1=0;
    integer count_clk2=0;

```



```

41  always @(posedge clk)
42  begin
43  | if(count_clk1==2499999*2+1)
44  | begin
45  |     count_clk1 <=0;
46  |     //闪烁重置
47  |     twinkle<=0;
48  |     //电子时钟
49  |     if(Y==0)
50  |     begin
173  |         //调闹钟
174  |         if(Y==1)
175  |         begin
176  |         if(rst)
177  |         begin
185  |         //调时
186  |         if(key[0]==0)
187  |         begin
220  |         if(key[1]==0)
221  |         begin
244  |         if(key[2]==0)
245  |         begin
258  |             a<=q; a2<=q2; a3<=q3; a4<=q4; a5<=q5; a6<=q6;
259  |             end
260  |         end
261  |         end
262  |     else
263  |     count_clk1 <= count_clk1+1;
264  |     if(count_clk2==249999*2+1)
265  |     begin
266  |     count_clk2<=0;
267  |     if(Y==2)
268  |     begin
269  |     if(rst)
270  |     begin
271  |     if(en)
279  |     begin
280  |     begin
314  |         a<=m; a2<=m2; a3<=m3; a4<=m4; a5<=m5; a6<=m6;
315  |         end
316  |     end
317  |     else
318  |     count_clk2<=count_clk2+1;
319  |     end
320  endmodule
321

```

接口:

```

wire [3:0] a;
wire [3:0] b;
wire [3:0] c;
wire [3:0] d;
wire [3:0] e;
wire [3:0] f;
mux3_1 m(
    SW[9:8], //选择段
    CLOCK_50,
    KEY[3:0],
    SW[0], //使能端
    SW[1], //重置位

    LEDR[0],
    a,
    b,
    c,
    d,
    e,
    f
);
bcd7seg b1(a, HEX0);
bcd7seg b2(b, HEX1);
bcd7seg b3(c, HEX2);
bcd7seg b4(d, HEX3);
bcd7seg b5(e, HEX4);
bcd7seg b6(f, HEX5);

```

4.2.2 仿真

同样波形图不便仿真，通过上板仿真。

测试电子时钟 闹钟和秒表三个功能，通过两位选择端进行选择功能，同时每个功能也有清零和暂停键。

闹钟可以通过按钮调整时间。

5. 实验中遇到的问题及解决办法

问题：秒表和电子时钟的 clk 不统一，但是输出接口是同一个，必须放在一个 always 里面。

解决办法：直接在 always 里面分频计数，即 always 里面两个分频器。

6.实验启示和建议

always 内部语句是并行的，所以 always 里面不能有 always，可以写两个 always 都在外面，但是这两个 always 内部变量不能有交叉。