数字电路与数字系统实验

实验五 时钟

姓名： 你猜

学号： 你猜

班级： 你猜

邮箱： 你猜

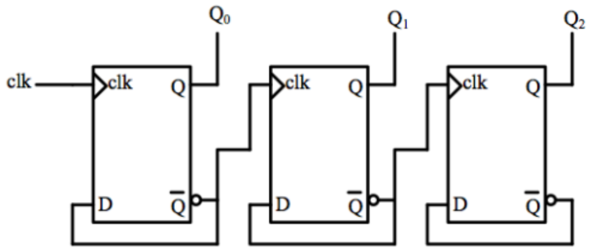
实验时间：你猜

一、 实验目的

学习计数器的原理，通过 verilog 语言设计、实现一个计数器和电子时钟。

二、 实验原理

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。



三、 实验环境

Quartus 18.1、FPGA 开发板

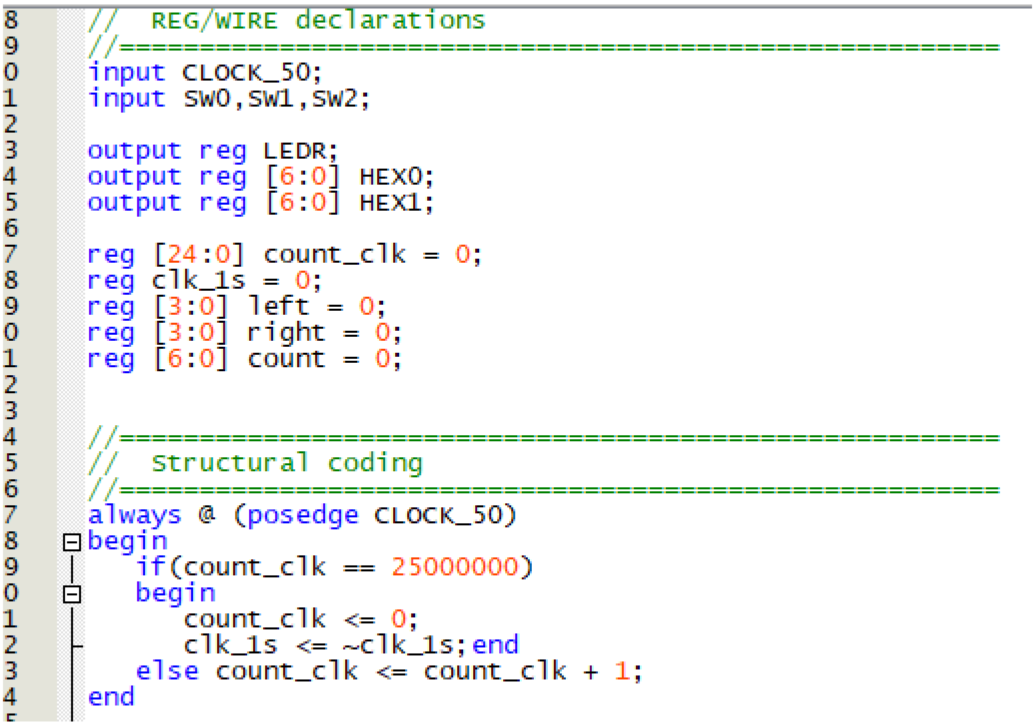
四、 实验过程

基础实验部分：

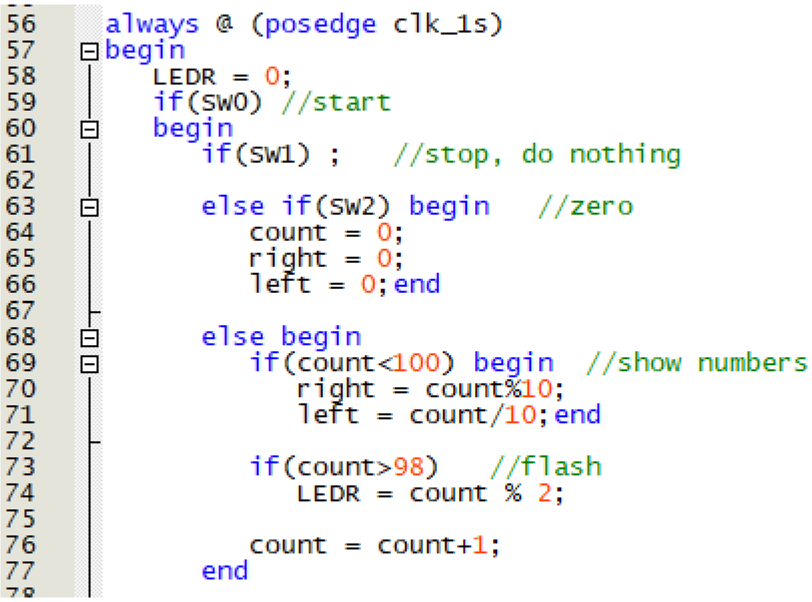
设计思路：
输入：CLOCK_50 为时钟信号，sw0,1,2 分别为开始，暂停和清零。
输出：led，hex0,1
用周期为 1 秒的时钟信号作为设计的时钟信号，若 sw1=1，暂停，则什么都不做；若 sw2=1，清零，则 count 置为零；否则，正常计数，当 count>98 时，led 开始闪烁，直到 count 溢出，重新从 0 开始计数。

设计代码：

clock.v



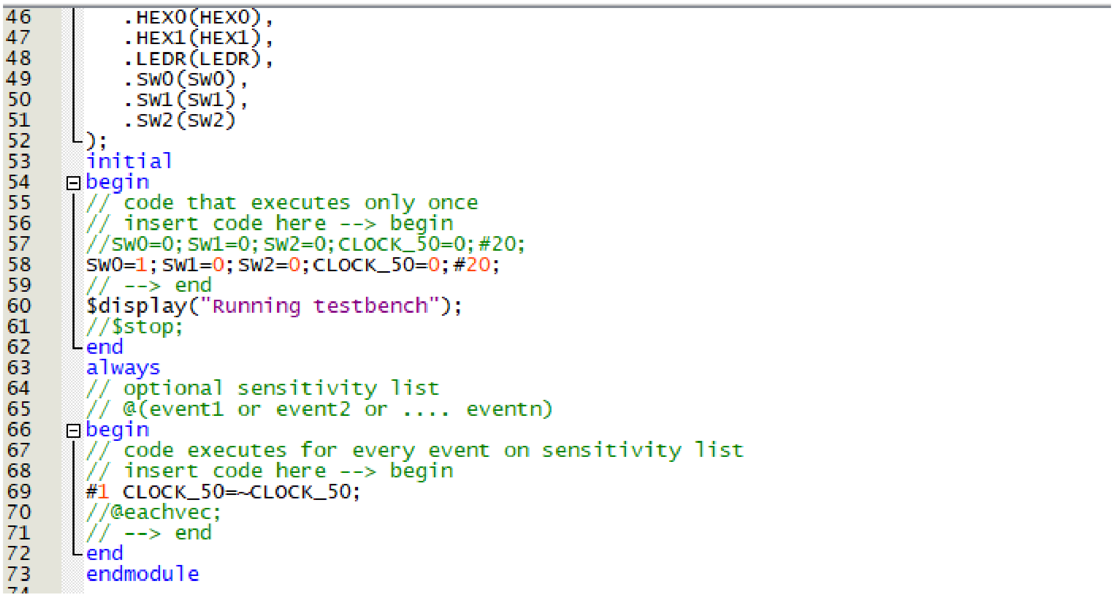
```
//  REG/WIRE declarations
//=========================================================
input CLOCK_50;
input SW0,SW1,SW2;

output reg LEDR;
output reg [6:0] HEX0;
output reg [6:0] HEX1;

reg [24:0] count_clk = 0;
reg clk_1s = 0;
reg [3:0] left = 0;
reg [3:0] right = 0;
reg [6:0] count = 0;


//=========================================================
//  Structural coding
//=========================================================
always @ (posedge CLOCK_50)
begin
   if(count_clk == 25000000)
   begin
      count_clk <= 0;
      clk_1s <= ~clk_1s;end
   else count_clk <= count_clk + 1;
end
```



```
always @ (posedge clk_1s)
begin
   LEDR = 0;
   if(SW0) //start
   begin
      if(SW1) ;    //stop, do nothing

      else if(SW2) begin   //zero
         count = 0;
         right = 0;
         left = 0;end

      else begin
         if(count<100) begin  //show numbers
            right = count%10;
            left = count/10;end

         if(count>98)    //flash
            LEDR = count % 2;

         count = count+1;
      end
```

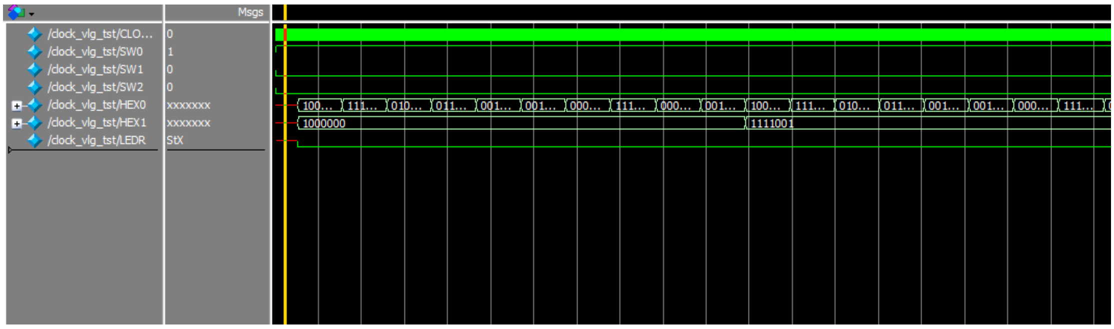
数码管部分略…

激励代码：

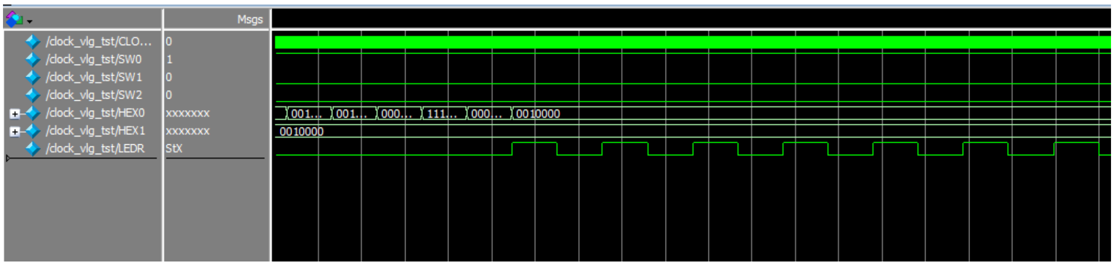


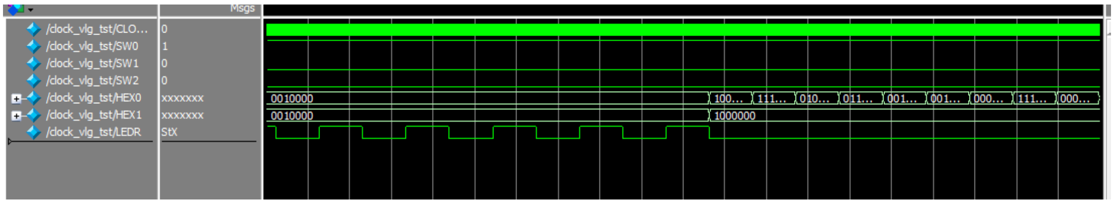
```
    .HEX0(HEX0),
    .HEX1(HEX1),
    .LEDR(LEDR),
    .SW0(SW0),
    .SW1(SW1),
    .SW2(SW2)
);
initial
begin
// code that executes only once
// insert code here --> begin
//SW0=0;SW1=0;SW2=0;CLOCK_50=0;#20;
SW0=1;SW1=0;SW2=0;CLOCK_50=0;#20;
// --> end
$display("Running testbench");
//$stop;
end
always
// optional sensitivity list
// @(event1 or event2 or .... eventn)
begin
// code executes for every event on sensitivity list
// insert code here --> begin
#1 CLOCK_50=~CLOCK_50;
//@eachvec;
// --> end
end
endmodule
```

ModelSim 仿真：

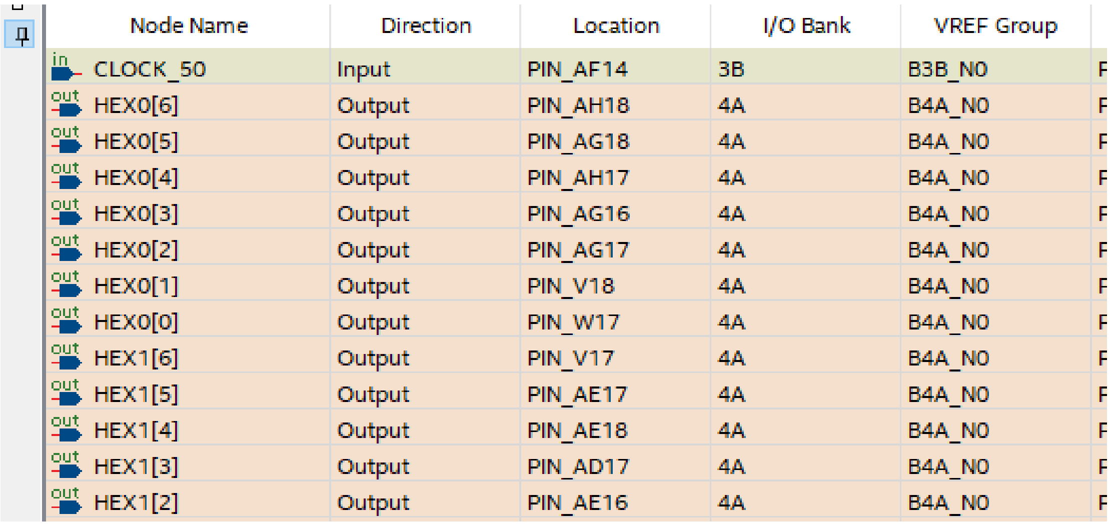


计数到达 99 之后，led 闪烁，经过 28 秒的闪烁之后，计数器清零，再次开始计数：

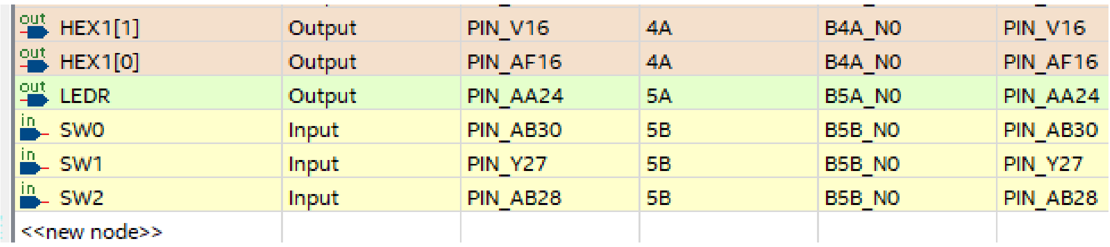




引脚分配：



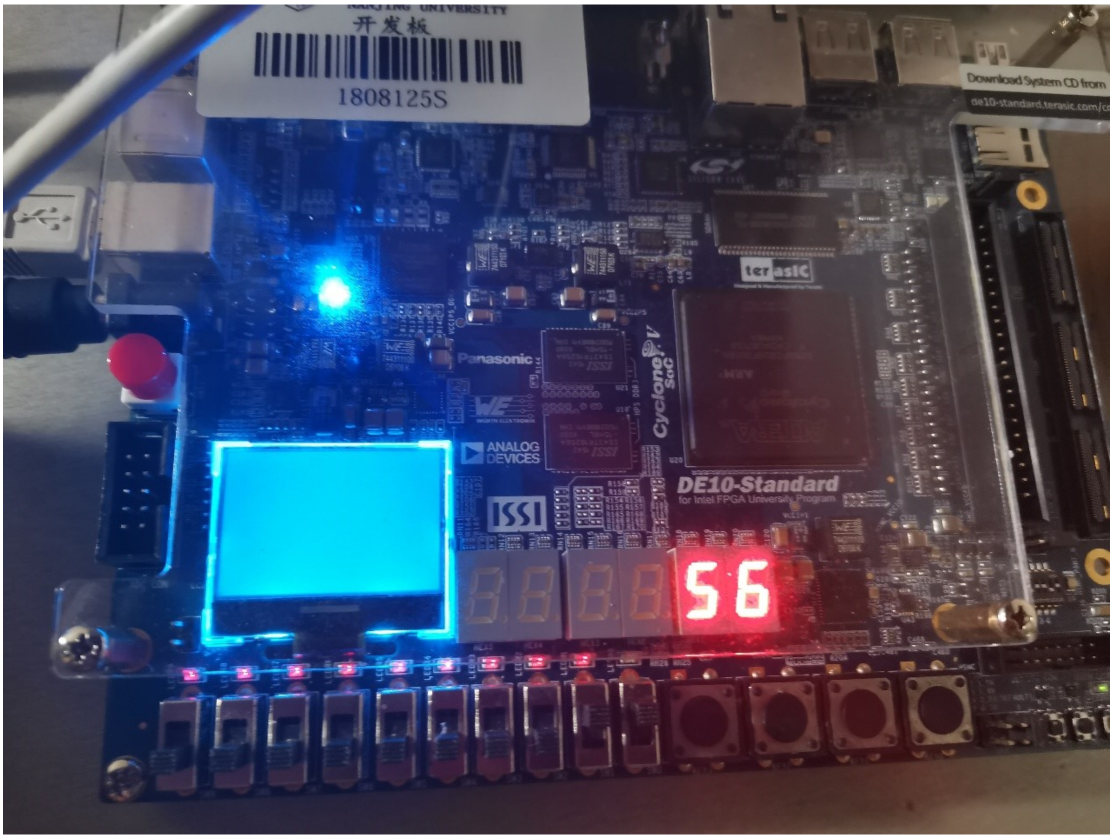
Node Name	Direction	Location	I/O Bank	VREF Group	
CLOCK_50	Input	PIN_AF14	3B	B3B_N0	F
HEX0[6]	Output	PIN_AH18	4A	B4A_N0	F
HEX0[5]	Output	PIN_AG18	4A	B4A_N0	F
HEX0[4]	Output	PIN_AH17	4A	B4A_N0	F
HEX0[3]	Output	PIN_AG16	4A	B4A_N0	F
HEX0[2]	Output	PIN_AG17	4A	B4A_N0	F
HEX0[1]	Output	PIN_V18	4A	B4A_N0	F
HEX0[0]	Output	PIN_W17	4A	B4A_N0	F
HEX1[6]	Output	PIN_V17	4A	B4A_N0	F
HEX1[5]	Output	PIN_AE17	4A	B4A_N0	F
HEX1[4]	Output	PIN_AE18	4A	B4A_N0	F
HEX1[3]	Output	PIN_AD17	4A	B4A_N0	F
HEX1[2]	Output	PIN_AE16	4A	B4A_N0	F



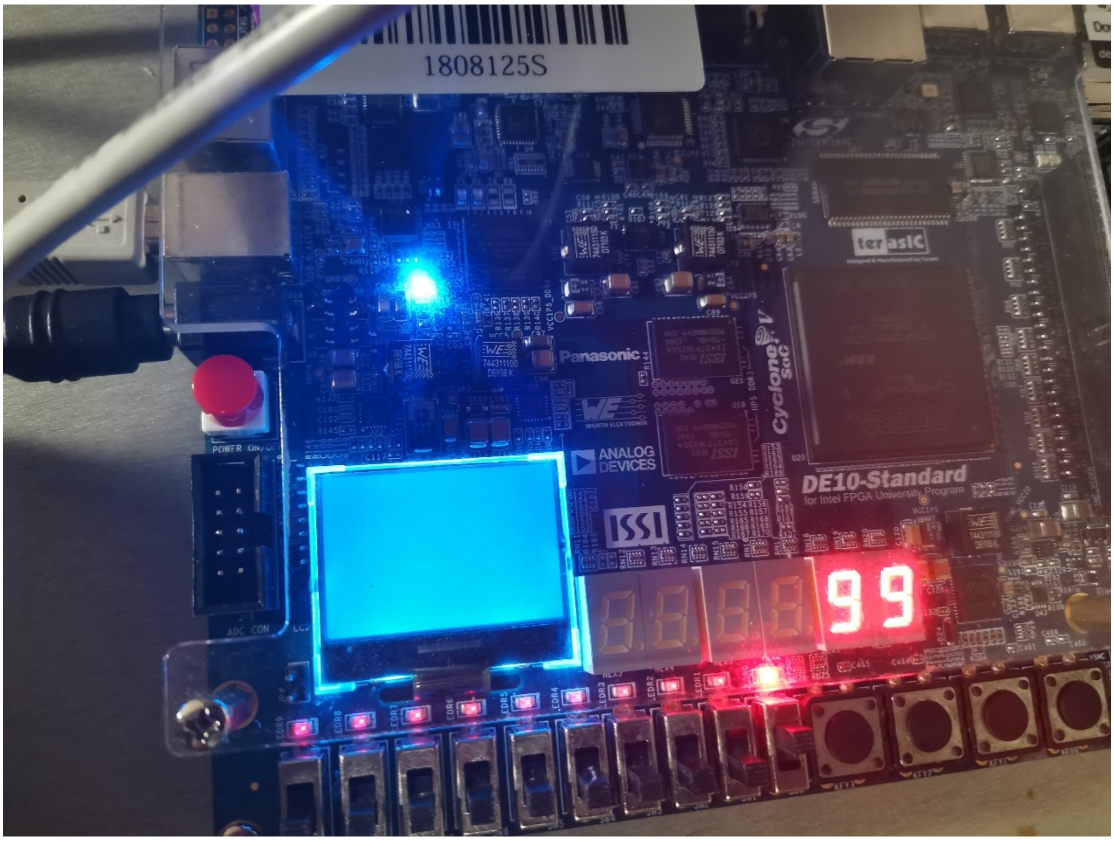
HEX1[1]	Output	PIN_V16	4A	B4A_N0	PIN_V16
HEX1[0]	Output	PIN_AF16	4A	B4A_N0	PIN_AF16
LEDR	Output	PIN_AA24	5A	B5A_N0	PIN_AA24
SW0	Input	PIN_AB30	5B	B5B_N0	PIN_AB30
SW1	Input	PIN_Y27	5B	B5B_N0	PIN_Y27
SW2	Input	PIN_AB28	5B	B5B_N0	PIN_AB28
<<new node>>					

实验结果：

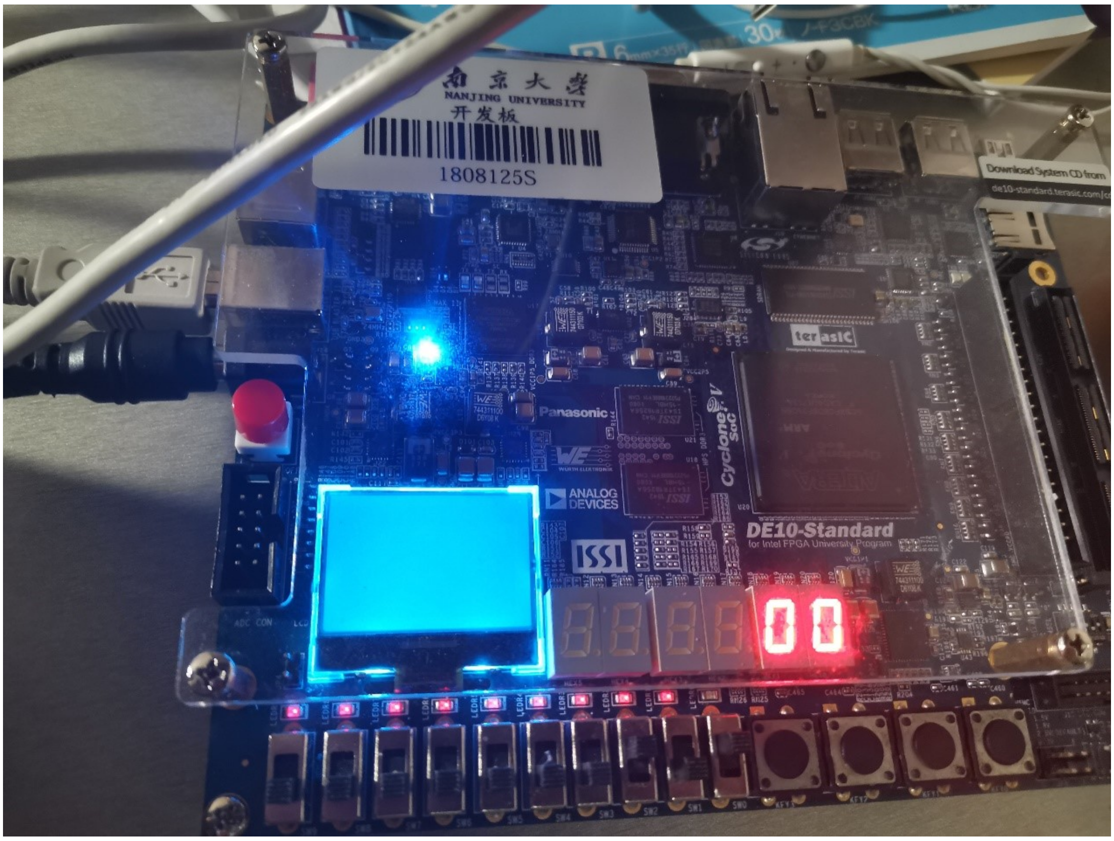
开始计数：



计数到 99，led 闪烁 28 秒：



Led 结束闪烁后，重新开始计数：



扩展实验部分：

设计思路：

输入为：clk 时钟信号，两位的 select 进行功能选择，两位 change_select 选择设置的时、分、秒，change_in 是设置时间的输入，sec_in 是秒表开关。

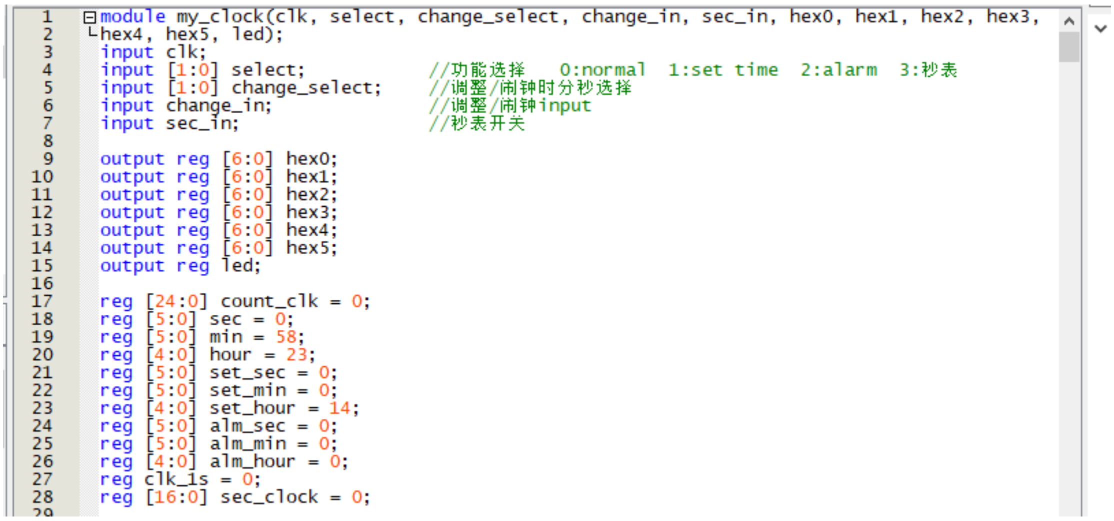
输出：hex0-5 是数码管，led 是闹钟输出。

用周期为 1 秒的时钟信号作为设计的时钟信号，若 select=1 时，将 sec，min，hour 赋值为手动设置的时分秒；否则 clock 正常运行。若 select=3 且秒表开关为 1，秒表同时也计数。当 clock 的时分秒与闹钟设置的时分秒相等时，led 亮。

检测到 change_in 上升沿时，若 select=1 或 2，则进行手动时间设置或闹钟设置。

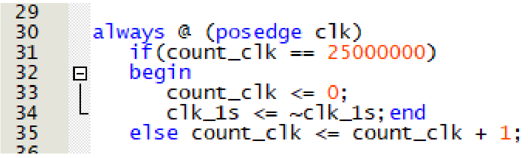
设计代码：

my_clock.v



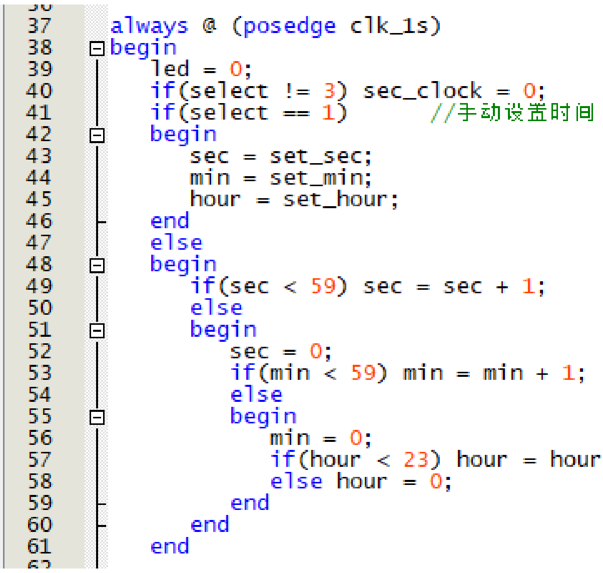
```
module my_clock(clk, select, change_select, change_in, sec_in, hex0, hex1, hex2, hex3,
hex4, hex5, led);
input clk;
input [1:0] select;          //功能选择   0:normal  1:set time  2:alarm  3:秒表
input [1:0] change_select;   //调整/闹钟时分秒选择
input change_in;             //调整/闹钟input
input sec_in;                //秒表开关

output reg [6:0] hex0;
output reg [6:0] hex1;
output reg [6:0] hex2;
output reg [6:0] hex3;
output reg [6:0] hex4;
output reg [6:0] hex5;
output reg led;

reg [24:0] count_clk = 0;
reg [5:0] sec = 0;
reg [5:0] min = 58;
reg [4:0] hour = 23;
reg [5:0] set_sec = 0;
reg [5:0] set_min = 0;
reg [4:0] set_hour = 14;
reg [5:0] alm_sec = 0;
reg [5:0] alm_min = 0;
reg [4:0] alm_hour = 0;
reg clk_1s = 0;
reg [16:0] sec_clock = 0;
```

生成 1 秒的时钟信号：



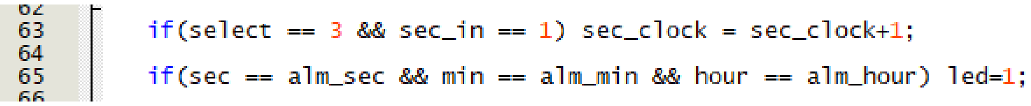
```

always @ (posedge clk)
   if(count_clk == 25000000)
   begin
      count_clk <= 0;
      clk_1s <= ~clk_1s;end
   else count_clk <= count_clk + 1;
```

时钟正常状态和设置手动设置的时间：



```
always @ (posedge clk_1s)
begin
    led = 0;
    if(select != 3) sec_clock = 0;
    if(select == 1)      //手动设置时间
    begin
        sec = set_sec;
        min = set_min;
        hour = set_hour;
    end
    else
    begin
        if(sec < 59) sec = sec + 1;
        else
        begin
            sec = 0;
            if(min < 59) min = min + 1;
            else
            begin
                min = 0;
                if(hour < 23) hour = hour + 1;
                else hour = 0;
            end
        end
    end
```

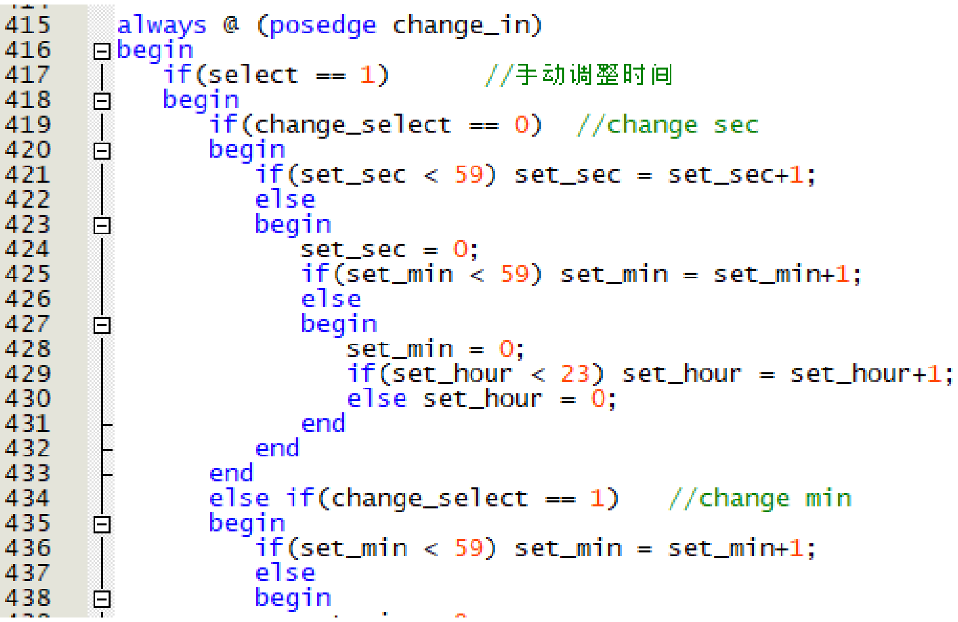
秒表和闹钟的显示：



```
    if(select == 3 && sec_in == 1) sec_clock = sec_clock+1;

    if(sec == alm_sec && min == alm_min && hour == alm_hour) led=1;
```

数码管部分略…

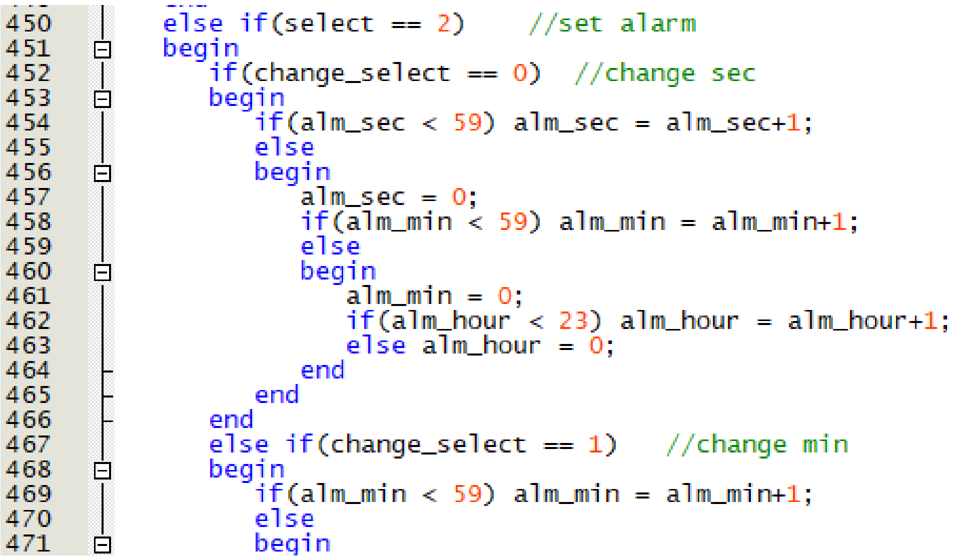
手动设置时间：



```
always @ (posedge change_in)
begin
    if(select == 1)      //手动调整时间
    begin
        if(change_select == 0)  //change sec
        begin
            if(set_sec < 59) set_sec = set_sec+1;
            else
            begin
                set_sec = 0;
                if(set_min < 59) set_min = set_min+1;
                else
                begin
                    set_min = 0;
                    if(set_hour < 23) set_hour = set_hour+1;
                    else set_hour = 0;
                end
            end
        end
        else if(change_select == 1)   //change min
        begin
            if(set_min < 59) set_min = set_min+1;
            else
            begin
```

change min、hour 部分略…

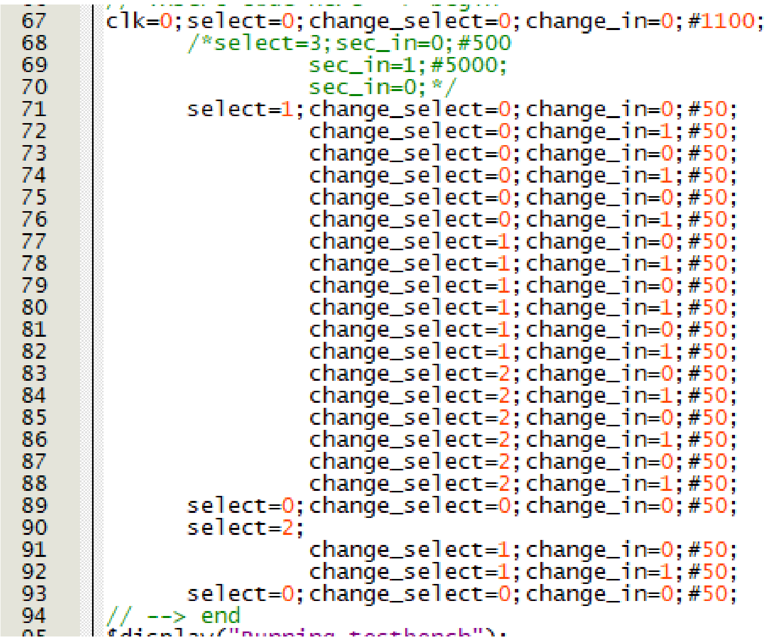
设置闹钟：



```
else if(select == 2)     //set alarm
begin
   if(change_select == 0)  //change sec
   begin
      if(alm_sec < 59) alm_sec = alm_sec+1;
      else
      begin
         alm_sec = 0;
         if(alm_min < 59) alm_min = alm_min+1;
         else
         begin
            alm_min = 0;
            if(alm_hour < 23) alm_hour = alm_hour+1;
            else alm_hour = 0;
         end
      end
   end
   else if(change_select == 1)   //change min
   begin
      if(alm_min < 59) alm_min = alm_min+1;
      else
      begin
```

change min、hour 部分略…

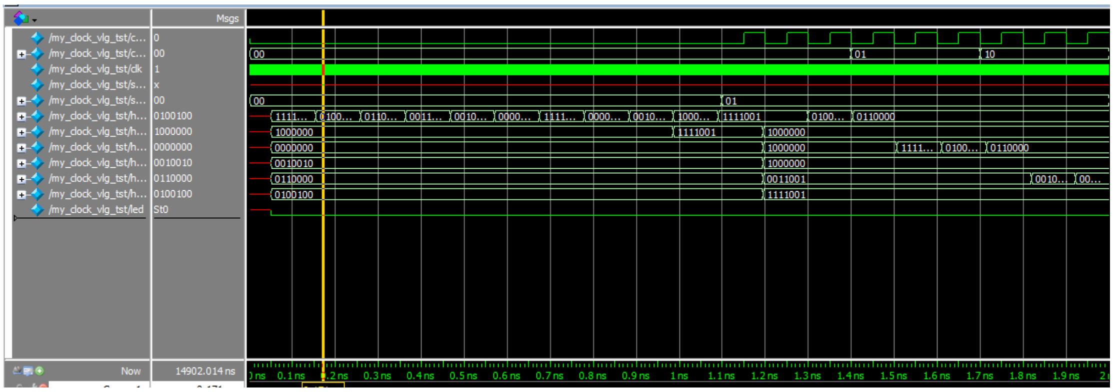
激励代码：



```
clk=0;select=0;change_select=0;change_in=0;#1100;
      /*select=3;sec_in=0;#500
               sec_in=1;#5000;
               sec_in=0;*/
      select=1;change_select=0;change_in=0;#50;
               change_select=0;change_in=1;#50;
               change_select=0;change_in=0;#50;
               change_select=0;change_in=1;#50;
               change_select=0;change_in=0;#50;
               change_select=0;change_in=1;#50;
               change_select=1;change_in=0;#50;
               change_select=1;change_in=1;#50;
               change_select=1;change_in=0;#50;
               change_select=1;change_in=1;#50;
               change_select=1;change_in=0;#50;
               change_select=1;change_in=1;#50;
               change_select=2;change_in=0;#50;
               change_select=2;change_in=1;#50;
               change_select=2;change_in=0;#50;
               change_select=2;change_in=1;#50;
               change_select=2;change_in=0;#50;
               change_select=2;change_in=1;#50;
      select=0;change_select=0;change_in=0;#50;
      select=2;
               change_select=1;change_in=0;#50;
               change_select=1;change_in=1;#50;
      select=0;change_select=0;change_in=0;#50;
// --> end
$display("Running testbench");
```

ModelSim 仿真：

时钟正常运行和手动设置时间：

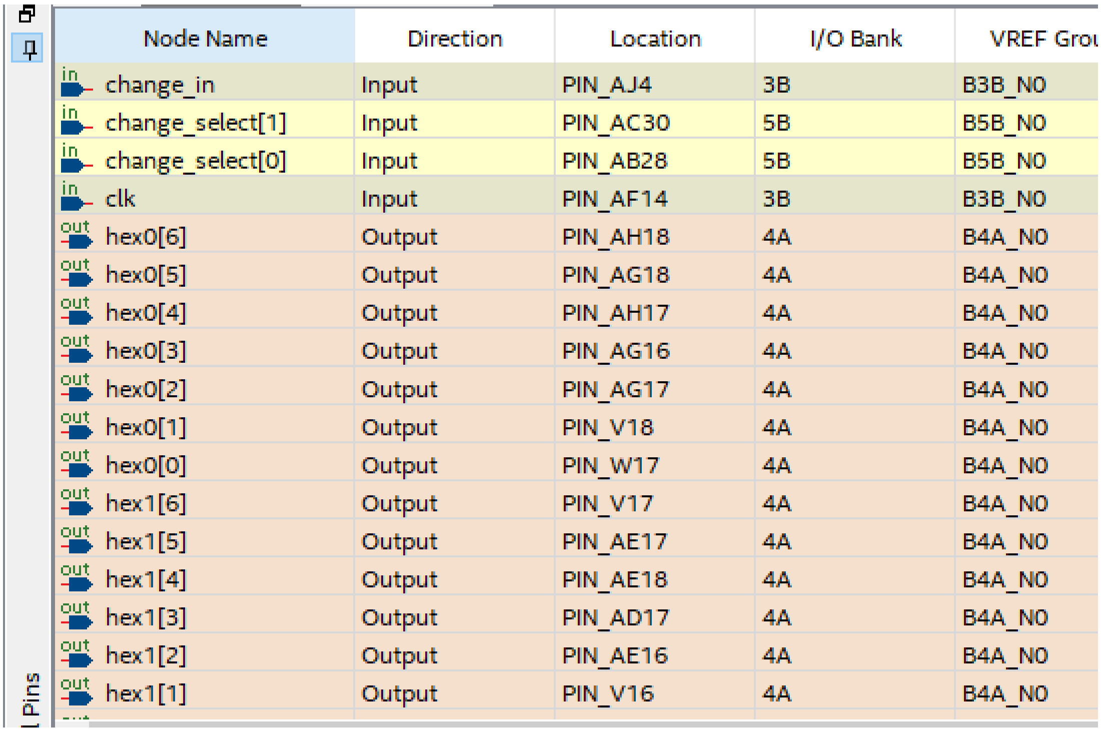


	Msgs
/my_clock_vlg_tst/c...	0
/my_clock_vlg_tst/c...	00
/my_clock_vlg_tst/clk	1
/my_clock_vlg_tst/s...	x
/my_clock_vlg_tst/s...	00
/my_clock_vlg_tst/h...	0100100
/my_clock_vlg_tst/h...	1000000
/my_clock_vlg_tst/h...	0000000
/my_clock_vlg_tst/h...	0010010
/my_clock_vlg_tst/h...	0110000
/my_clock_vlg_tst/h...	0100100
/my_clock_vlg_tst/led	St0

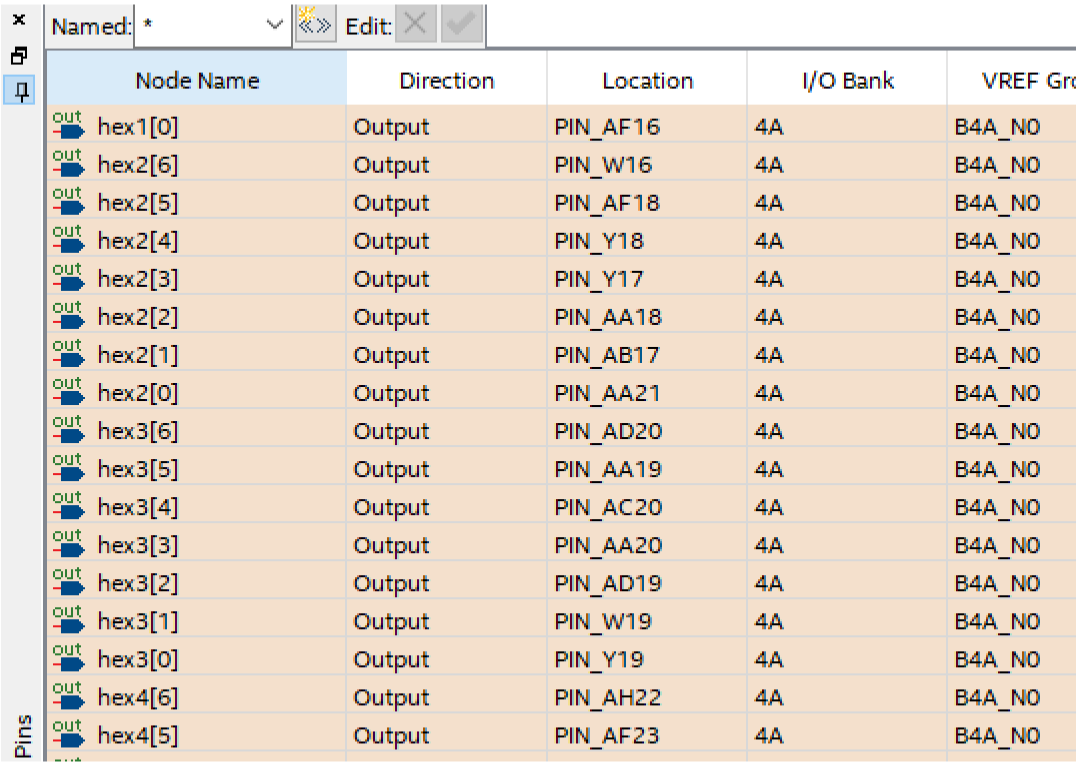
Now 14902.014 ns

闹钟和秒表略，跟基础实验部分差不多…

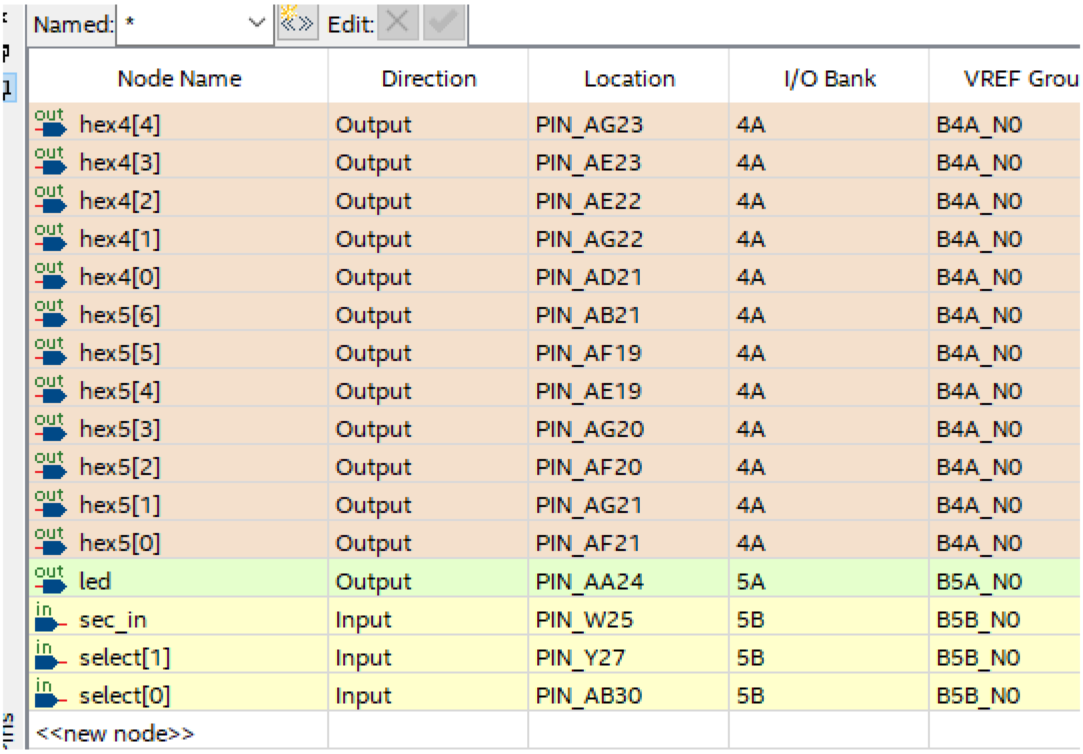
引脚分配：



Node Name	Direction	Location	I/O Bank	VREF Grou
change_in	Input	PIN_AJ4	3B	B3B_N0
change_select[1]	Input	PIN_AC30	5B	B5B_N0
change_select[0]	Input	PIN_AB28	5B	B5B_N0
clk	Input	PIN_AF14	3B	B3B_N0
hex0[6]	Output	PIN_AH18	4A	B4A_N0
hex0[5]	Output	PIN_AG18	4A	B4A_N0
hex0[4]	Output	PIN_AH17	4A	B4A_N0
hex0[3]	Output	PIN_AG16	4A	B4A_N0
hex0[2]	Output	PIN_AG17	4A	B4A_N0
hex0[1]	Output	PIN_V18	4A	B4A_N0
hex0[0]	Output	PIN_W17	4A	B4A_N0
hex1[6]	Output	PIN_V17	4A	B4A_N0
hex1[5]	Output	PIN_AE17	4A	B4A_N0
hex1[4]	Output	PIN_AE18	4A	B4A_N0
hex1[3]	Output	PIN_AD17	4A	B4A_N0
hex1[2]	Output	PIN_AE16	4A	B4A_N0
hex1[1]	Output	PIN_V16	4A	B4A_N0



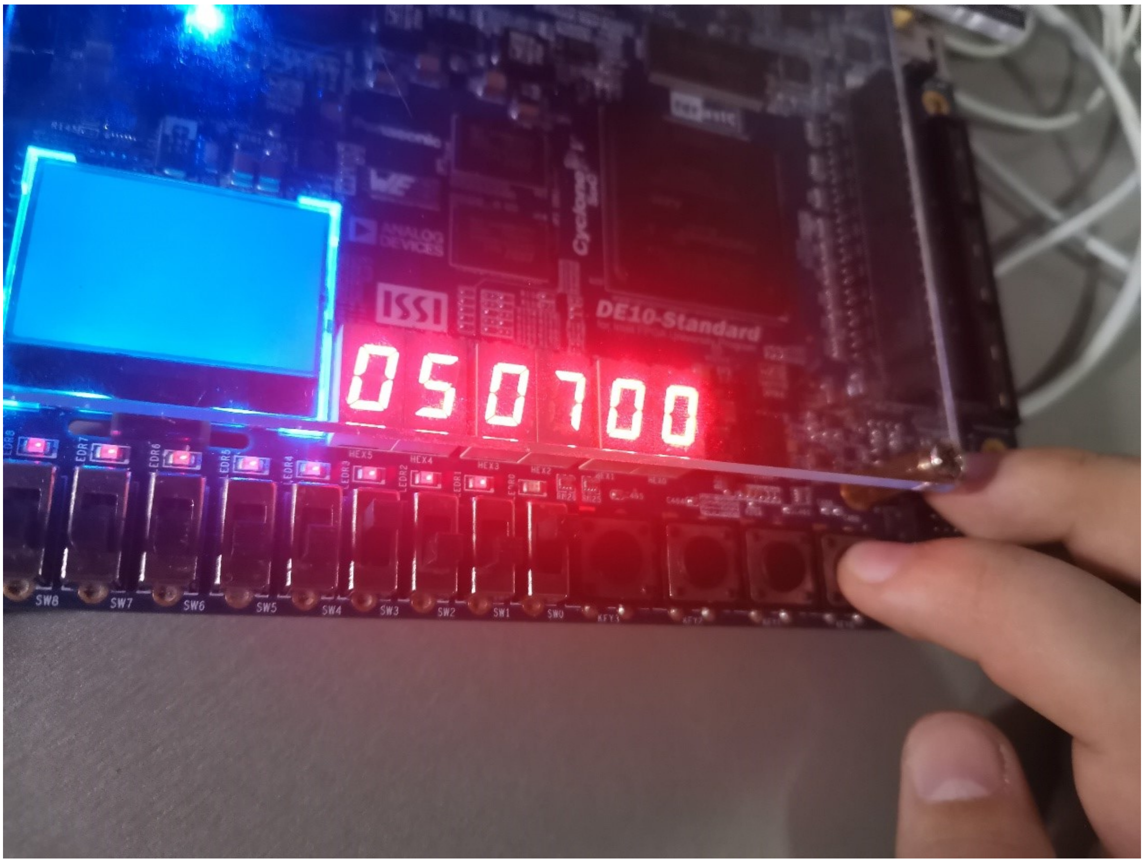
Node Name	Direction	Location	I/O Bank	VREF Gro
hex1[0]	Output	PIN_AF16	4A	B4A_N0
hex2[6]	Output	PIN_W16	4A	B4A_N0
hex2[5]	Output	PIN_AF18	4A	B4A_N0
hex2[4]	Output	PIN_Y18	4A	B4A_N0
hex2[3]	Output	PIN_Y17	4A	B4A_N0
hex2[2]	Output	PIN_AA18	4A	B4A_N0
hex2[1]	Output	PIN_AB17	4A	B4A_N0
hex2[0]	Output	PIN_AA21	4A	B4A_N0
hex3[6]	Output	PIN_AD20	4A	B4A_N0
hex3[5]	Output	PIN_AA19	4A	B4A_N0
hex3[4]	Output	PIN_AC20	4A	B4A_N0
hex3[3]	Output	PIN_AA20	4A	B4A_N0
hex3[2]	Output	PIN_AD19	4A	B4A_N0
hex3[1]	Output	PIN_W19	4A	B4A_N0
hex3[0]	Output	PIN_Y19	4A	B4A_N0
hex4[6]	Output	PIN_AH22	4A	B4A_N0
hex4[5]	Output	PIN_AF23	4A	B4A_N0

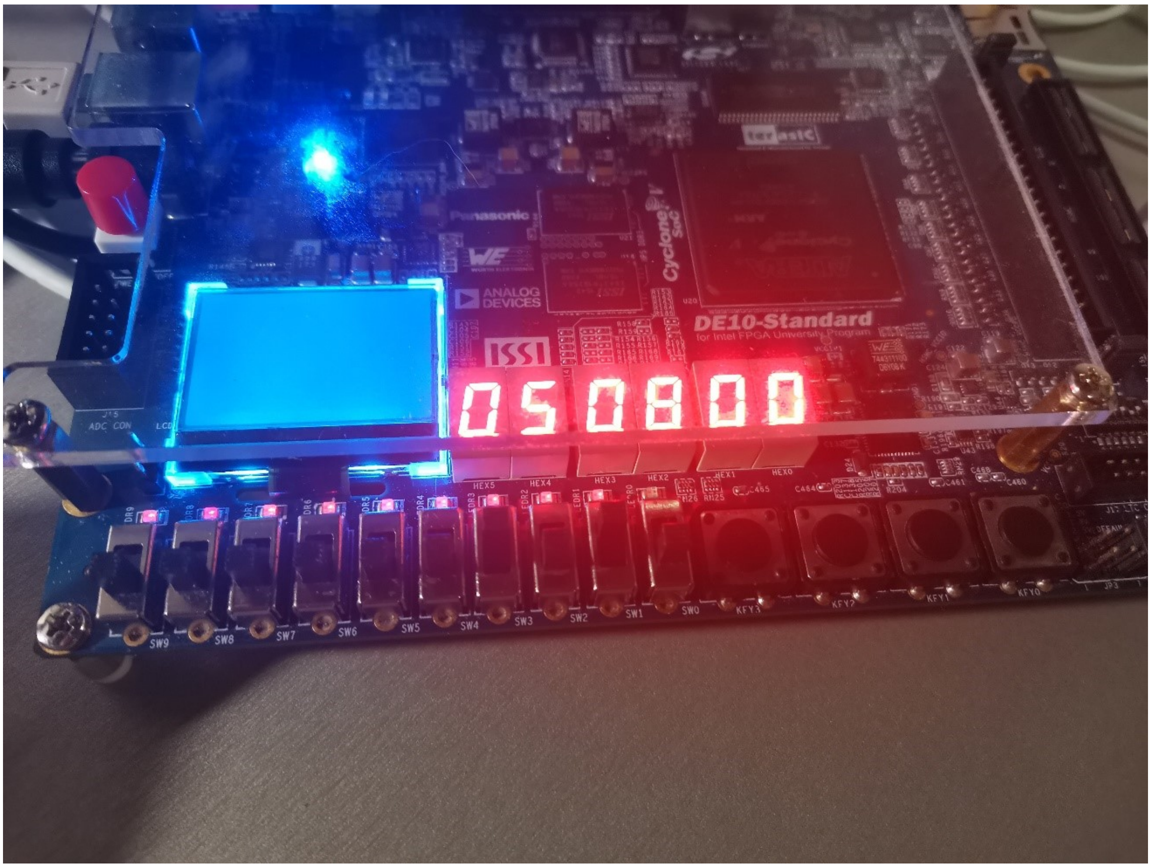


Node Name	Direction	Location	I/O Bank	VREF Grou
hex4[4]	Output	PIN_AG23	4A	B4A_N0
hex4[3]	Output	PIN_AE23	4A	B4A_N0
hex4[2]	Output	PIN_AE22	4A	B4A_N0
hex4[1]	Output	PIN_AG22	4A	B4A_N0
hex4[0]	Output	PIN_AD21	4A	B4A_N0
hex5[6]	Output	PIN_AB21	4A	B4A_N0
hex5[5]	Output	PIN_AF19	4A	B4A_N0
hex5[4]	Output	PIN_AE19	4A	B4A_N0
hex5[3]	Output	PIN_AG20	4A	B4A_N0
hex5[2]	Output	PIN_AF20	4A	B4A_N0
hex5[1]	Output	PIN_AG21	4A	B4A_N0
hex5[0]	Output	PIN_AF21	4A	B4A_N0
led	Output	PIN_AA24	5A	B5A_N0
sec_in	Input	PIN_W25	5B	B5B_N0
select[1]	Input	PIN_Y27	5B	B5B_N0
select[0]	Input	PIN_AB30	5B	B5B_N0
<<new node>>				

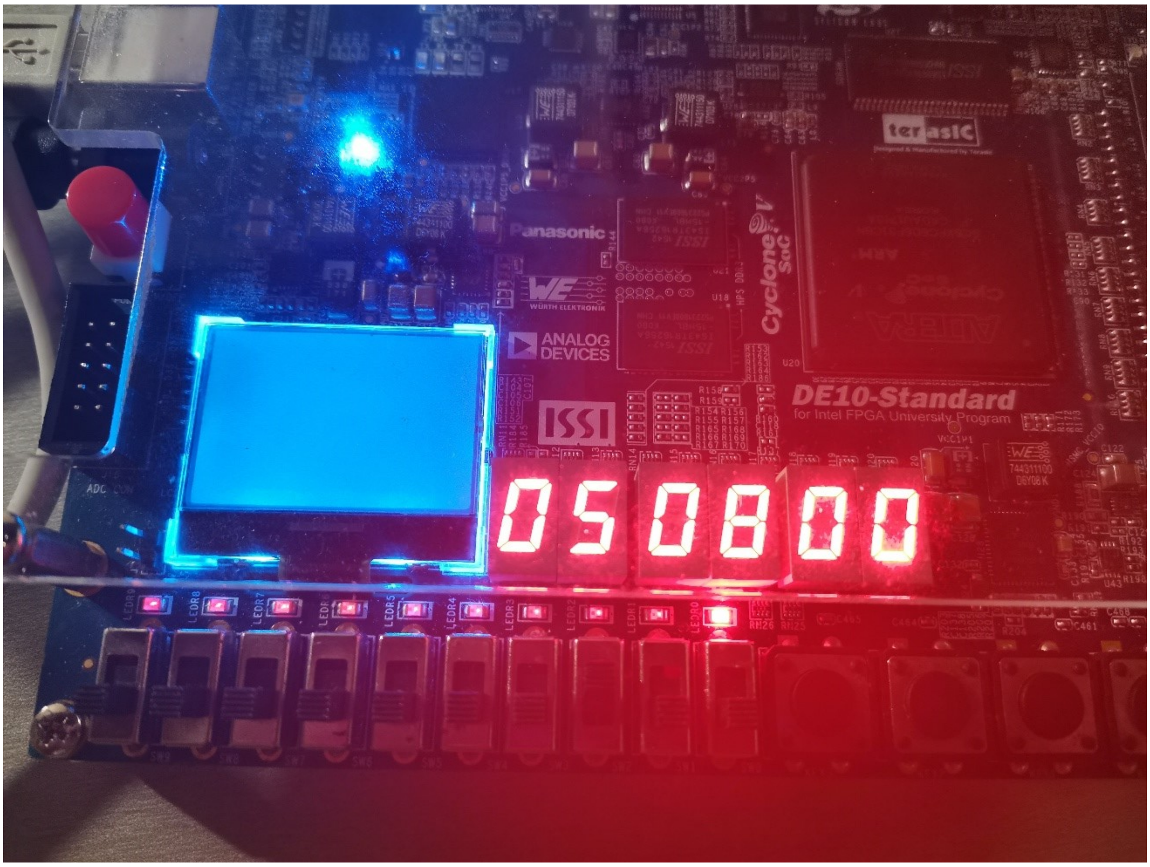
实验结果：

设置时间和闹钟：

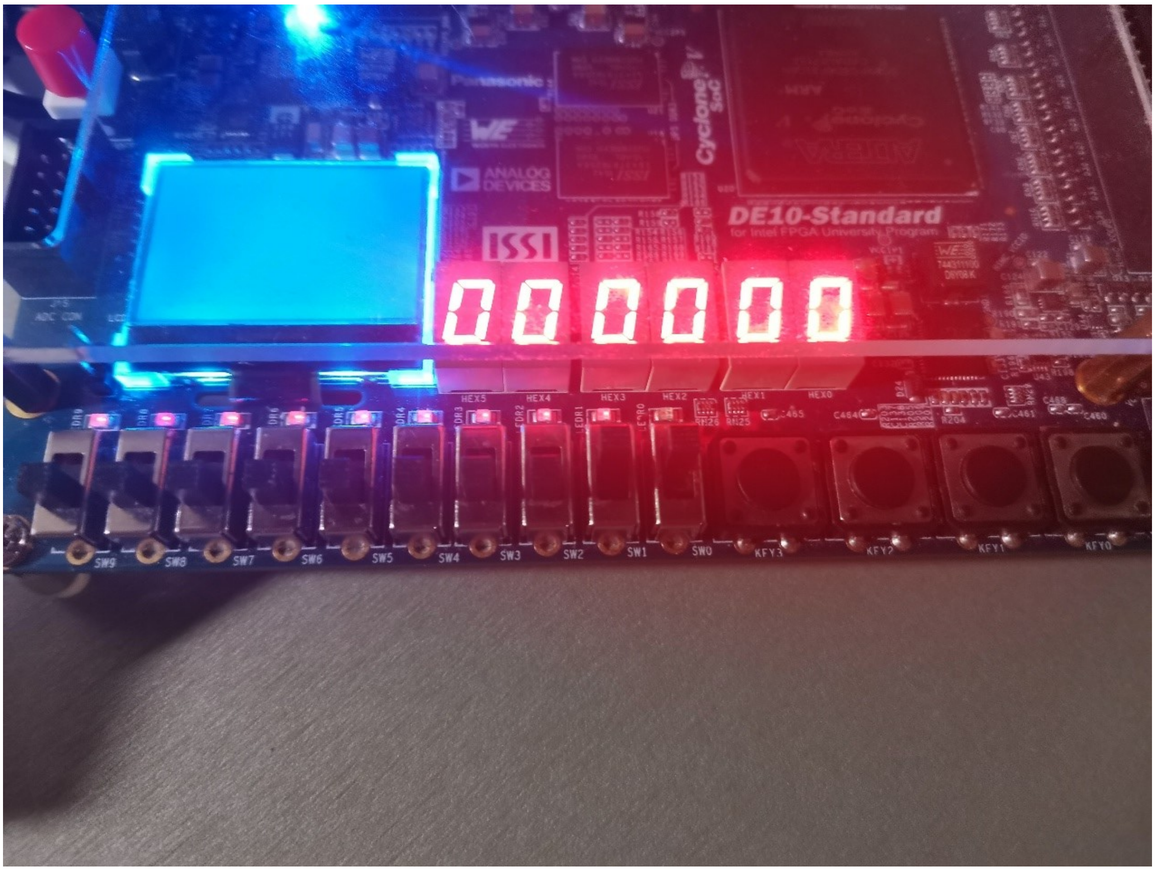


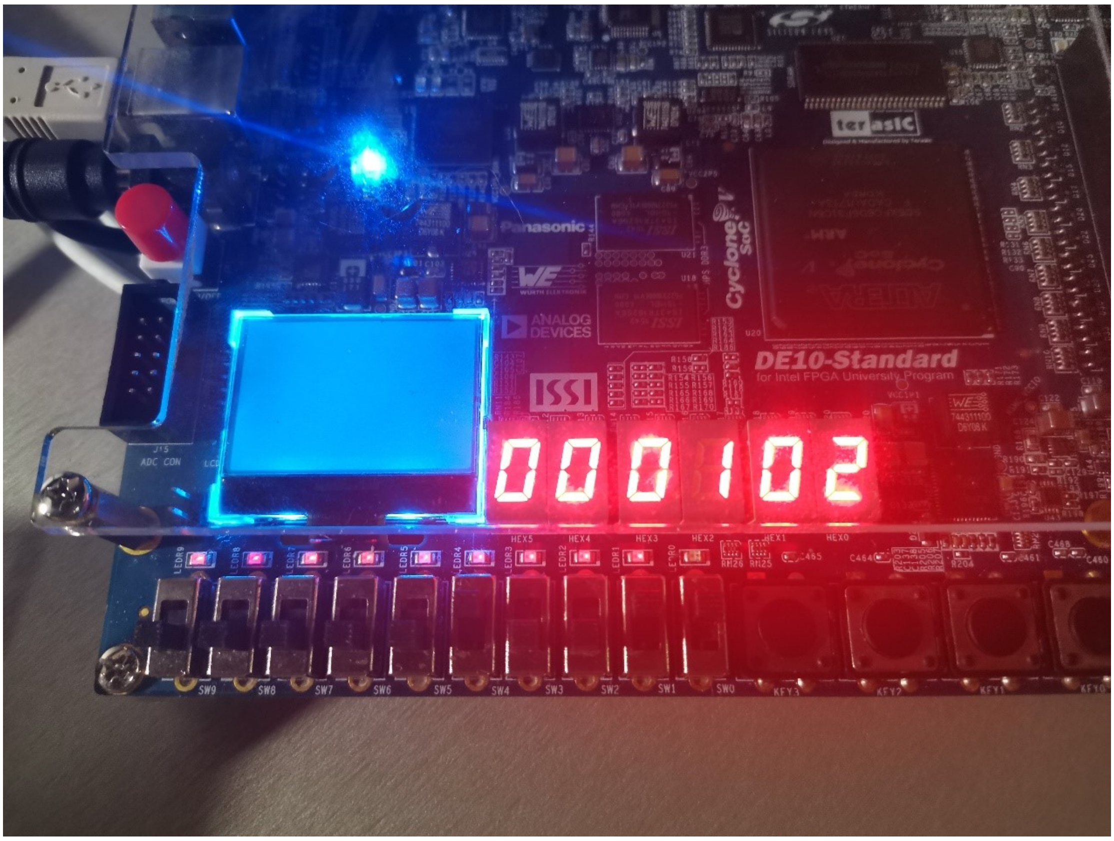


设定的闹钟时间到，led 亮：



秒表：





五、 实验中遇到的问题及解决办法

1、一开始计数器到 98 就停止，重新开始计数时也直接从 1 开始，后来发现是因为自己不细心，在判断 count>98 的 if 语句中，也对 count 加了 1，导致 count>98 以后，每次都是加 2。

2、感觉自己电子时钟的设计思路不是很好，也有很多冗余的代码…

六、 启示

写的时候一定要想好了再写，不然边想边写的话很容易混乱…

七、 意见与建议

虽然之前我并没有上过数字电路这门课，但是实验手册前面的讲解非常的清楚，由浅入深，帮助我学习和完成了这次实验。设计电子时钟挺好玩的！