实验四、数字钟

一、实验目的

1.学习动态数码管的工作原理；

2.实现FPGA对八位动态数码管的控制；

3.熟悉模块化编程的操作流程；

4.掌握按键去抖的方法；

5.掌握计数器的使用。

二、实验内容

1.实现FPGA对八位动态数码管的控制，使其能够正常工作；

2.应用八位数码管作为显示器件设计一个简单数字钟

三、实验要求

1.准确计时，以数字形式显示时、分、秒的时间；

2.小时计时可以是“12翻1”或“23翻0”的形式，可选择；

3.时分秒的校时功能；

四、实验原理

秒脉冲生成

Clk

秒信号计数

S\_puls

S\_h

S\_L

M\_puls

分信号计数

M\_h

M\_L

H\_puls

时信号计数

H\_h

H\_L

S\_adj

M\_adj

H\_adj

校时使能生成

Adjust\_en

Adjust

Mode

图1 数字钟实现原理

**1、按键消抖**

外部按键进入FPGA内部，都需要进行消抖。消抖思路为，在按键变化期间不采样按键，隔1ms在采样。

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*按键消抖-输出为key\_press信号\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//adjust:调节按键，用于产生调节使能

//H\_adj：小时加按键,M\_adj:分加信号,S\_adj:秒加按键\*/

reg[19:0] cnt\_key\_reg;//计数器--消抖

wire[3:0] key= {adjust,H\_adj,M\_adj,S\_adj}; //输入按键

reg[3:0] key\_reg1,key\_reg2;//输入按键2拍信号

reg[3:0] key\_press; //按键消抖输出

always @(posedge sclk) key\_reg1 <= key;

always @(posedge sclk) key\_reg2 <= key\_reg1;

always @(posedge sclk or negedge rst\_n)

if(!rst\_n) cnt\_key\_reg <= 20'b0;

else if(key\_reg1 != key\_reg2) cnt\_key\_reg <= 20'b0;

else if(cnt\_key\_reg == 20'd9\_9999) cnt\_key\_reg <= 20'b0;//1ms:消抖时间

else cnt\_key\_reg <= cnt\_key\_reg +1'b1;

//采样按键

always @(posedge sclk or negedge rst\_n)

if(!rst\_n) key\_press <= 5'b0;

else if(cnt\_key\_reg == 20'd9\_9999) key\_press <= key\_reg2;//1ms:消抖时间

**2、校时信号产生的原理**

通过外部按键（Adjust）生成校时使能信号，在校时使能信号有效期间，通过外部按键（S\_adj：秒校时，M\_adj：分校时，H\_adj：时校时）进行校时。

在按键消抖功能完成后，由key\_press信号生成校时脉冲信号，秒校时脉冲，分校时脉冲，时校时脉冲，具体方法采样上升沿检测电路。

/\*\*\*\*\*\*\*\*\*\*校时信号产生，输出为adjust\_en ,H\_adj\_pos,M\_adj\_pos,S\_adj\_pos信号\*\*\*\*\* \*/

reg adjust\_r1,adjust\_r2,H\_adj\_r1,H\_adj\_r2,M\_adj\_r1,M\_adj\_r2,S\_adj\_r1,S\_adj\_r2;

reg adjust\_pos,H\_adj\_pos,M\_adj\_pos,S\_adj\_pos;//四个按键的上升沿脉冲信号

always @(posedge sclk) {adjust\_r1,H\_adj\_r1,M\_adj\_r1,S\_adj\_r1} <= key\_press;

always@(posedgesclk){adjust\_r2,H\_adj\_r2,M\_adj\_r2,S\_adj\_r2}<= {adjust\_r1,H\_adj\_r1,M\_adj\_r1,S\_adj\_r1};

always @(posedge sclk)

if({adjust\_r2,adjust\_r1}==2'b01) adjust\_pos <= 1'b1;

else adjust\_pos <= 1'b0;

always @(posedge sclk)

if({H\_adj\_r2,H\_adj\_r1}==2'b01) H\_adj\_pos <= 1'b1;

else H\_adj\_pos <= 1'b0;

always @(posedge sclk)

if({M\_adj\_r2,M\_adj\_r1}==2'b01) M\_adj\_pos <= 1'b1;

else M\_adj\_pos <= 1'b0;

always @(posedge sclk)

if({S\_adj\_r2,S\_adj\_r1}==2'b01) S\_adj\_pos <= 1'b1;

else S\_adj\_pos <= 1'b0;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*调时使能生成\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

reg adjust\_en; //校时使能信号。

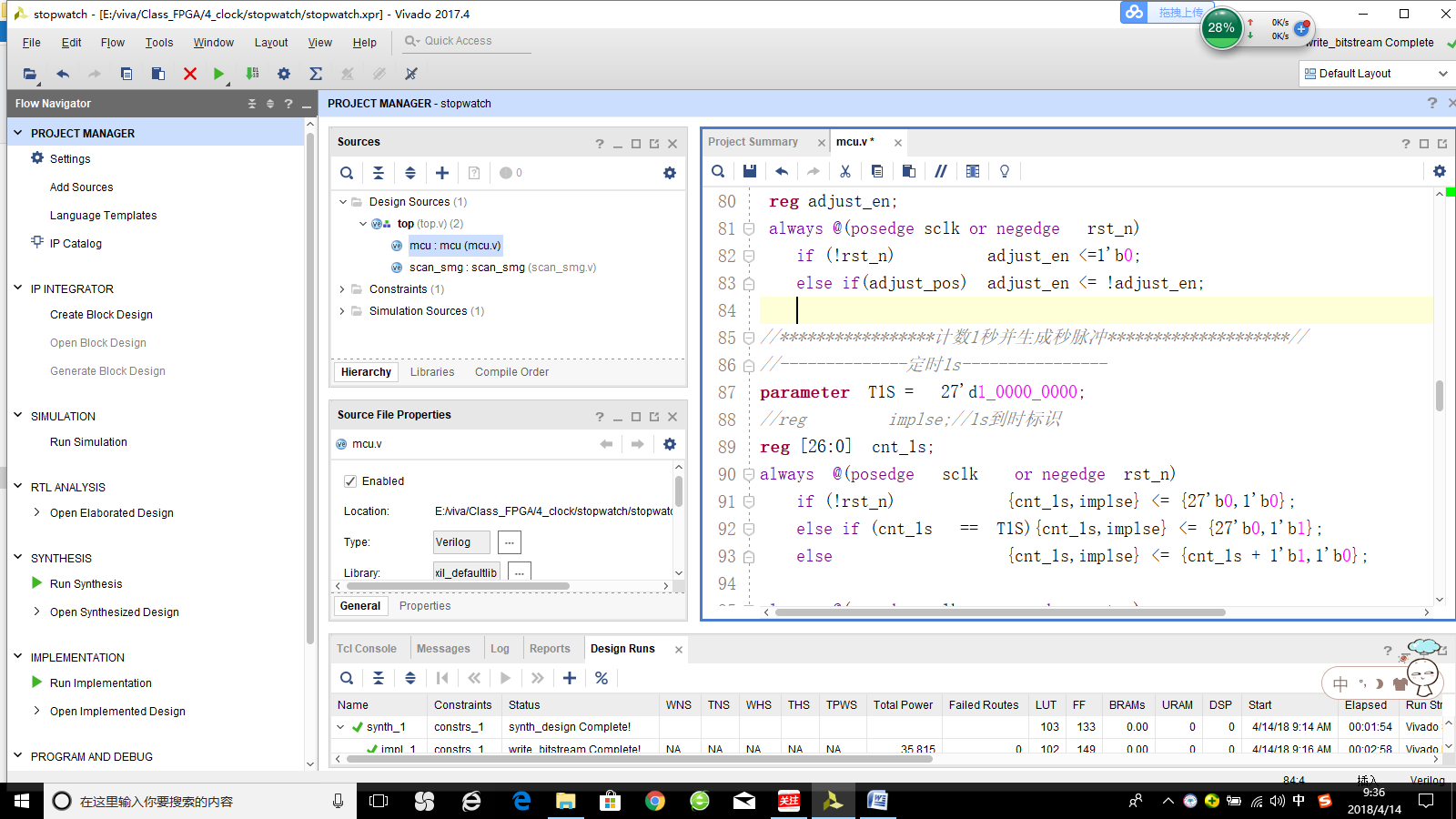
always @(posedge sclk or negedge rst\_n)

if (!rst\_n) adjust\_en <=1'b0;

else if(adjust\_pos) adjust\_en <= !adjust\_en;

**3、时钟信号产生**

1）秒脉冲产生秒脉冲信号（S\_puls）；



2）秒信号计数模块产生数字钟的秒信号的十位数字（S\_h）和个位数字（S\_L），以及分脉冲信号（M\_puls）。

always @(posedge sclk or negedge rst\_n)

if (!rst\_n) {sl,sh,min\_pulse} <= 9'd0;

else if (S\_adj\_pos && adjust\_en)//校时分支：adjust\_en=1且由按键生成秒信号

if (sl == 4'd9)

begin

sl <= 0;

if (sh == 4'd5) sh <= 0;

else sh <= sh + 1;

end

else sl <= sl + 1;

else if (implse && (!adjust\_en))//正常分支：adjust\_en=0且由秒脉冲生成秒信号

if (sl == 4'd9)

begin

sl <= 0;

if (sh == 4'd5) {sh,min\_pulse} <= 2'b01;

else sh <= sh + 1;

end

else sl <= sl + 1;

else min\_pulse <= 0;

3）分信号计数模块产生数字钟的分信号的十位数字（M\_h）和个位数字（M\_L），以及分脉冲信号（H\_puls）。

always @(posedge sclk or negedge rst\_n)

if (!rst\_n) {ml,mh,hour\_pulse} <= {4'd9,4'd5,1'b0};//初始值为59分

else if (M\_adj\_pos && adjust\_en) //校时分支：adjust\_en=1且由按键生成分信号

if (ml == 4'd9)

begin

ml <= 0;

if (mh == 4'd5) mh <= 1'b0;

else mh <= mh + 1;

end

else ml <= ml + 1;

else if (min\_pulse && (!adjust\_en)) //正常分支：adjust\_en=0且由分脉冲生成分信号

if (ml == 4'd9)

begin

ml <= 0;

if (mh == 4'd5) {mh,hour\_pulse} <= 2'b01;

else mh <= mh + 1;

end

else ml <= ml + 1;

else hour\_pulse <= 0;

4）时信号计数模块根据mode信号（12/24制，外部输入），产生数字钟的时信号的十位数字（H\_h）和个位数字（H\_L），以及天脉冲信号（H\_puls）。

always @(posedge sclk or negedge rst\_n)

if (!rst\_n && Mode) {hh,hl,day\_pulse} <= {2'd2,4'd3,1'b0}; //24小时制初值为23时

else if (!rst\_n && (!Mode)) {hh,hl,day\_pulse} <= {2'd1,4'd2,1'b0}; //12小时制初值为12时

else if (hour\_pulse && Mode && (!adjust\_en)) //正常分支：24制，adjust\_en=0且由时脉冲生成分信号

if (hh == 2'd2 && hl == 4'd3) {hh,hl,day\_pulse} <= {2'd0,4'd0,1'b1}; //23翻0

else if (hl == 4'd9)

begin

hl <= 0;

hh <= hh + 1;

end

else hl <= hl + 1;

else if (hour\_pulse && (!Mode) && (!adjust\_en)) //正常分支：12制，adjust\_en=0且由时脉冲生成分信号

if (hh == 2'd1 && hl == 4'd2) {hh,hl,day\_pulse} <= {2'd0,4'd1,1'b1}; //12翻1

else if (hl == 4'd9)

begin

hl <= 0;

hh <= hh + 1;

end

else hl <= hl + 1;

else if (H\_adj\_pos && Mode && adjust\_en) //校时分支：24制，adjust\_en=1且由时脉冲生成分信号

if (hh == 2'd2 && hl == 4'd3) {hh,hl} <= {2'd0,4'd0}; //23翻0

else if (hl == 4'd9)

begin

hl <= 0;

hh <= hh + 1;

end

else hl <= hl + 1;

else if (H\_adj\_pos && (!Mode) && adjust\_en) //校时分支：12制，adjust\_en=1且由时脉冲生成分信号

if (hh == 2'd1 && hl == 4'd2) {hh,hl} <= {2'd0,4'd1}; //12翻1

else if (hl == 4'd9)

begin

hl <= 0;

hh <= hh + 1;

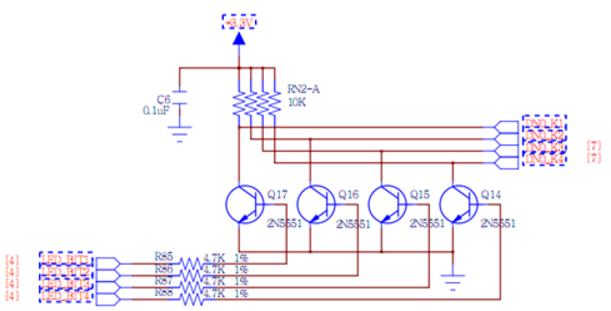
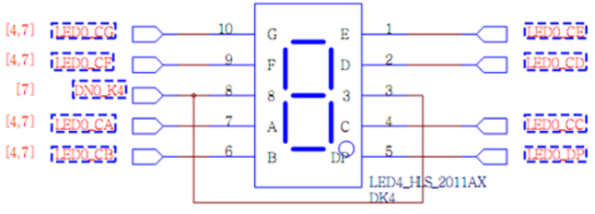
end

else hl <= hl + 1;

else day\_pulse <= 0;

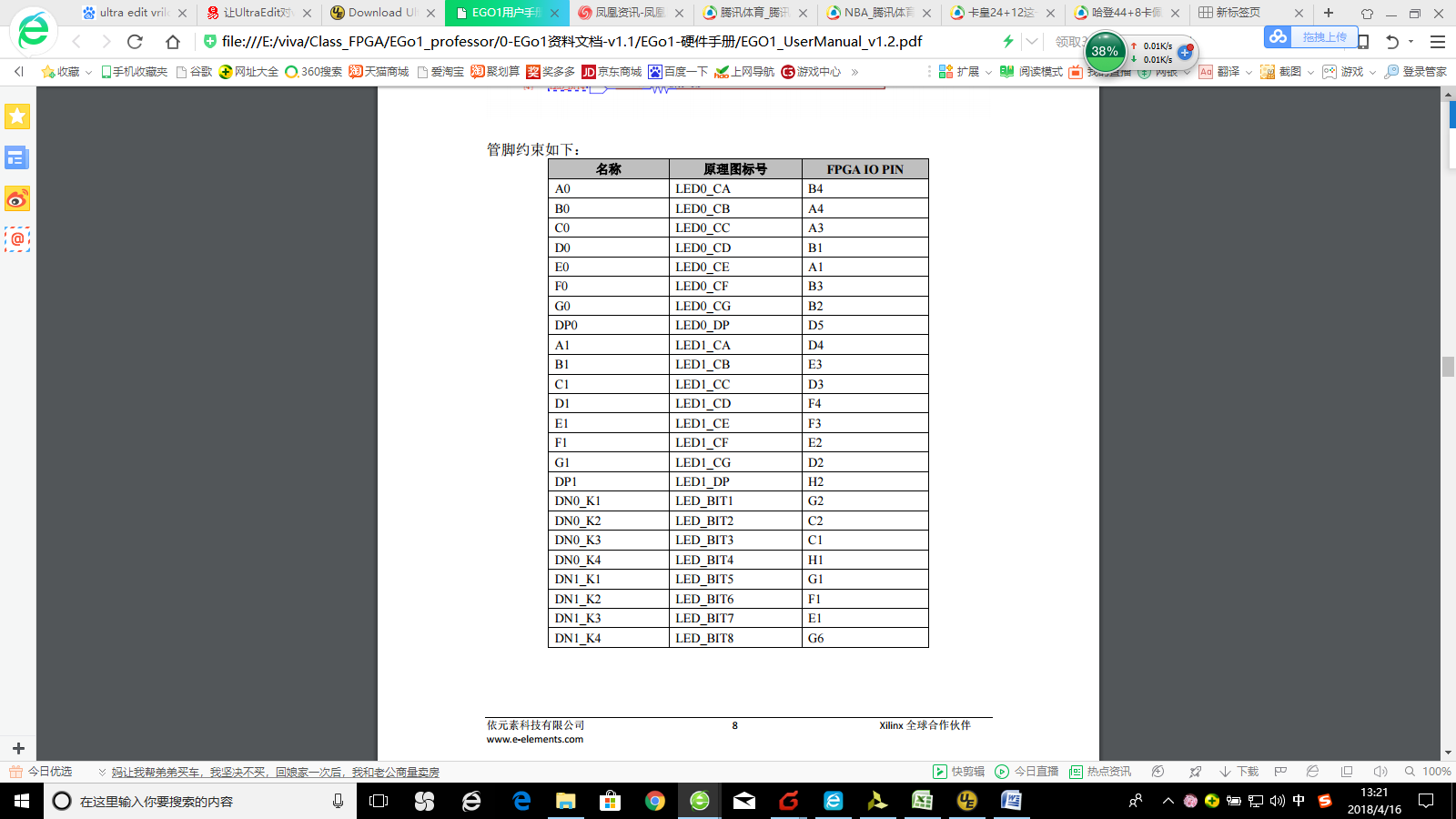
4、动态扫描数码管

EGO1开发板的数码管为共阴极数码管，即公共级输入低电平。共阴管由三极管驱动，FPGA需要提供正向信号。同时段选连接高电平，数码管上的对应位置才可以被点亮。因此，FPGA输出有效的片选信号和段选信号都应该是高电平。



实验板上8个数码管，4个为一组，每组有8bit的段选信号，4bit位选信号，在本次实验中，时分秒信号各用2位数码管，至少用到6位数码管，而且每位数码管显示内容不一样，必须用到动态扫描。

动态显示的特点是将所有位数码管的段选线并联在一起，由位选线控制是哪一位数码管有效。动态显示的特点是将所有位数码管的段选线并联在一起，由位选线控制是哪一位数码管有效。EGO1开发板的数码管共8个，4个数码管的段选硬件上是连在一起的。下表给出了数码管信号的管脚号：

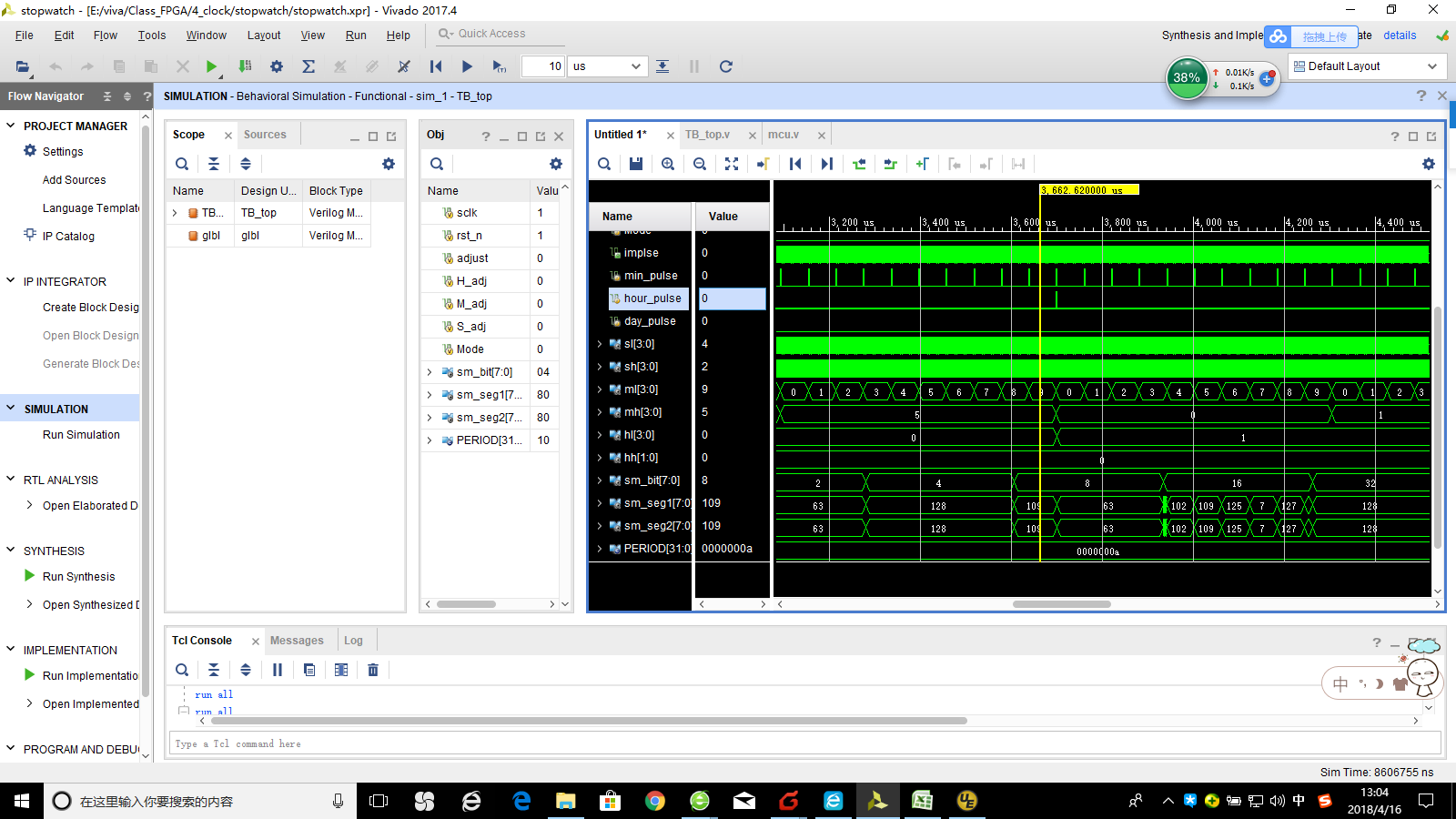


点亮数码管采用动态扫描显示。所谓动态扫描显示即轮流向各位数码管送出字形码和相应的位选，只要扫描显示速度够快，利用发光管的余辉和人眼视觉暂留作用，使人的感觉好像各位数码管同时都在显示。动态扫描显示时刷新频率最好大于50HZ,即没显示一轮的时间不超过20ms，每个数码管显示时间不能太长也不能太短，时间太长会影响刷新率，导致总体显示呈现闪烁的现象，时间太短发光二极管的电流导通时间也就短，会影响总体的显示亮度。一般控制在1ms左右最佳。

五、实验结果

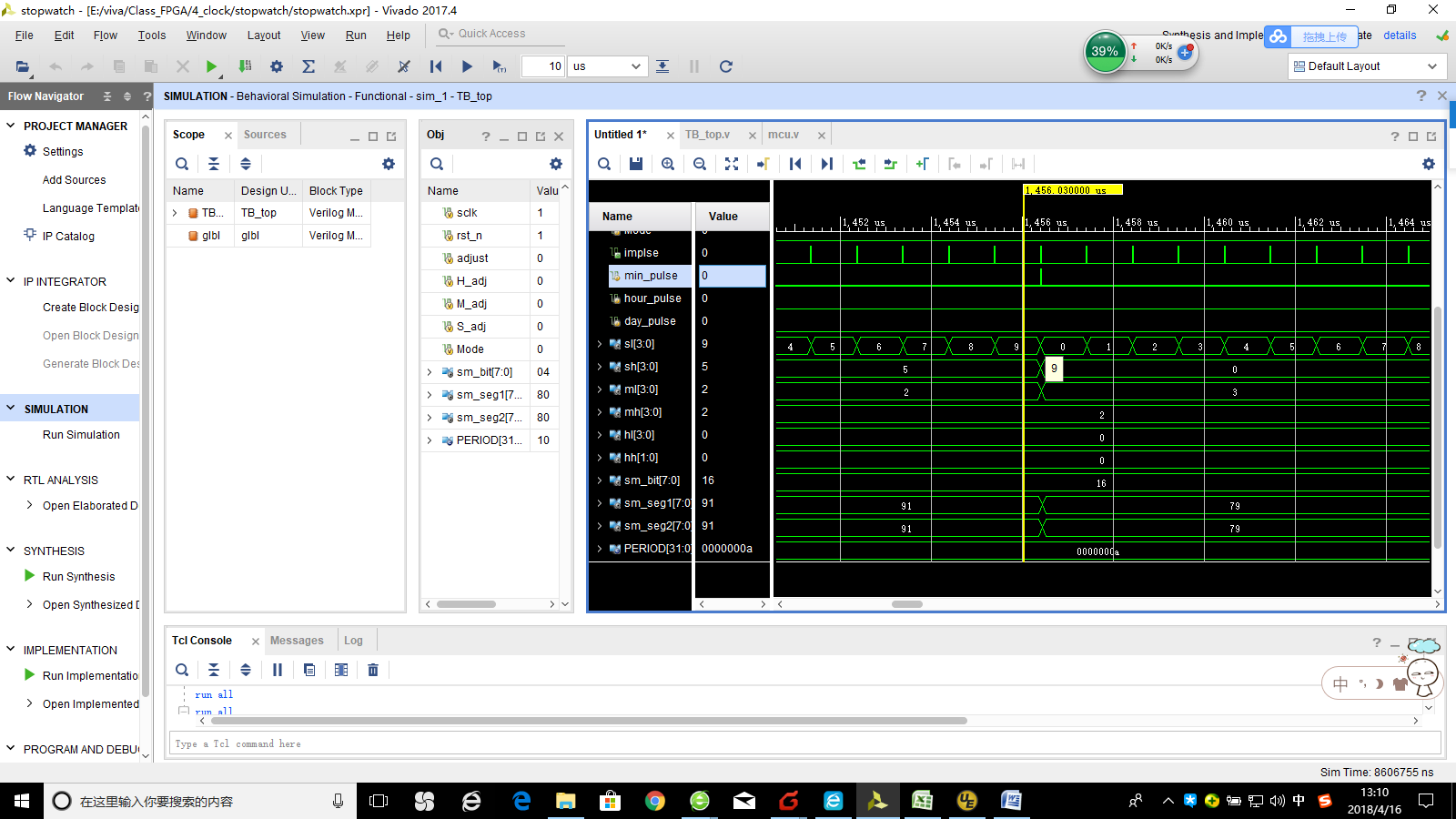
1、仿真

1）小时计数器时序仿真



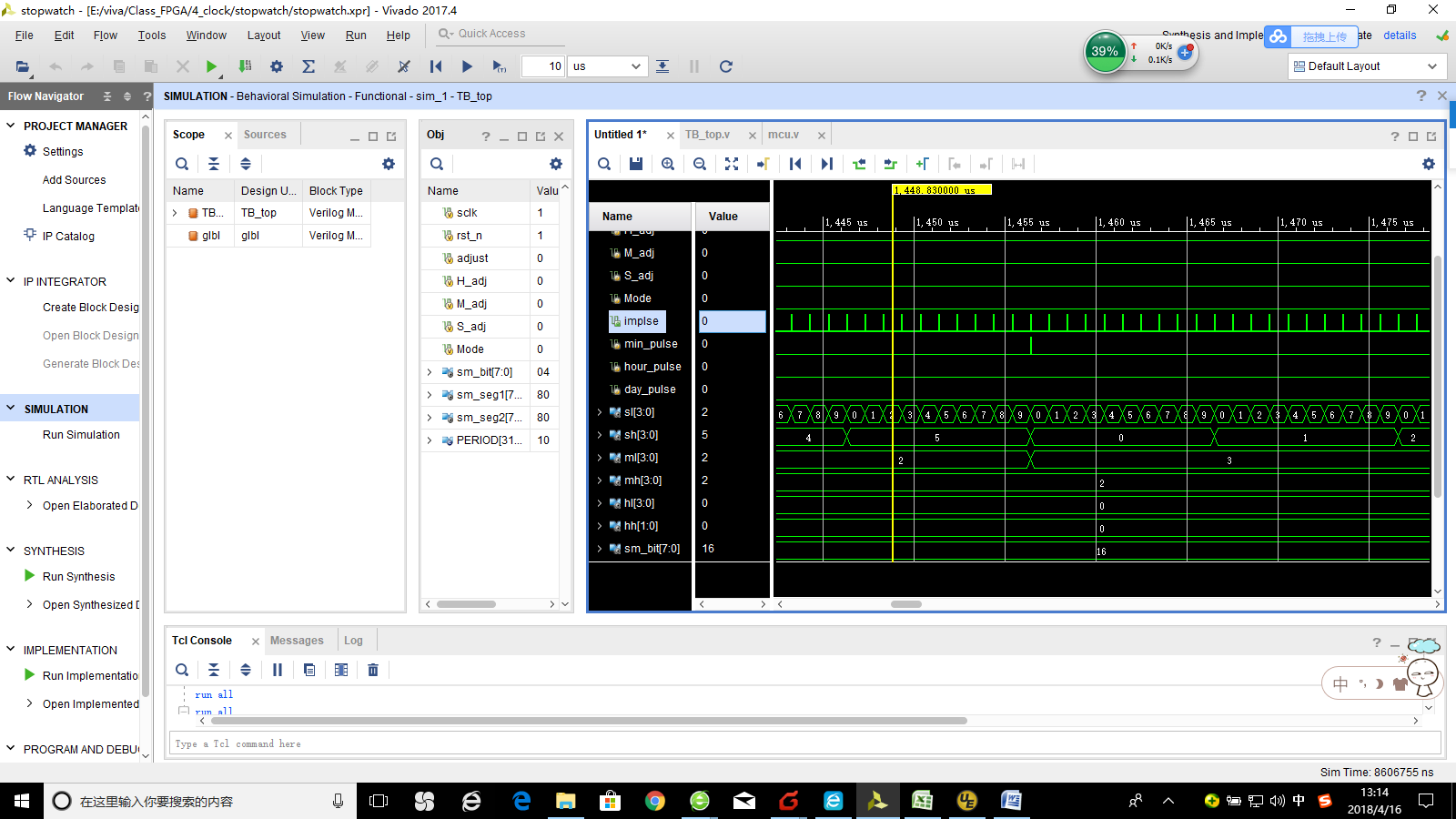
小时计数器(hl和hh)的变化，在黄线右边，小时脉冲（hour\_pulse）来的时候变化，且此时分钟计数器（ml和mh）是59变化到0，说明小时计数器时序正确。

2）分钟计数器时序仿真



分钟计数器(ml和mh)的变化，在黄线右边，分钟脉冲（hour\_pulse）来的时候变化，且此时秒钟计数器（sl和sh）是59变化到0，说明分钟计数器时序正确。

3）秒钟计数器时序仿真



秒钟计数器(ml和mh)的变化，在黄线右边，秒钟脉冲（imulse）来的时候变化，且图中可以看出秒钟计数器从0变到59，然后又到0，说明秒钟计数器时序正确。注意，为了缩减仿真时间，仿真的时候秒脉冲信号并没有计数到1s。

2 实验

