

本科实验报告

课程名称： **FPGA应用**

主讲教师： 夏书峰

学院、系： 微电子学院

专 业： 电子科学与技术类

班 级：

学 号：

姓 名：

联系电话：

电子邮箱：

2021～2022学年第3学期

实验成绩记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 内容 | 学时 | 满分 | 得分 |
| 1 | Vivado/ModelSim实践 | 4 | 15 |  |
| 2 | 时序逻辑设计1 — RTC | 4 | 20 |  |
| 3 | Xilinx嵌入式逻辑分析仪 | 4 | 15 |  |
| 4 | Xilinx嵌入式处理器实践 | 4 | 20 |  |
| 5 | 时序逻辑设计2 — FSM | 8 | 30 |  |
|  |  |  |  |  |
| 总计 | | 24 | 100 |  |

主讲教师签字：

实验二《实时钟逻辑设计》实验报告

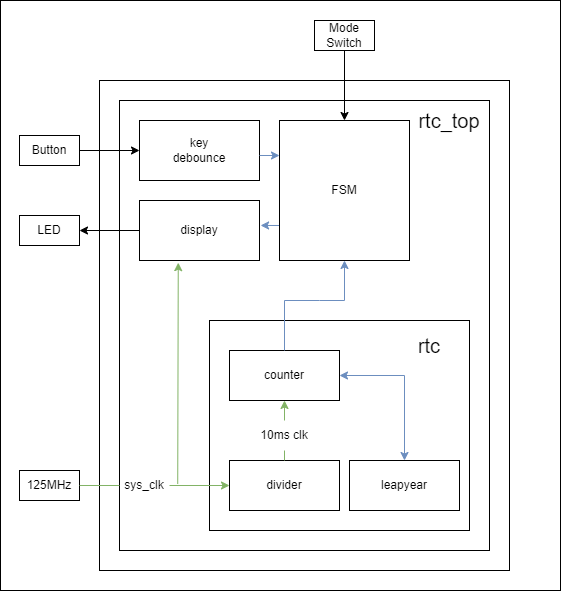
【实验内容】

设计一个计时用的实时钟（RTC：Real Time Clock）电路，计数年、月、日、时、分、秒、百分秒、星期几数据，有闰年修正功能，能正确计2月份日数，闰年判断用实验一里设计的闰年模块。

要求该模块可综合。再对其进行布线后仿真，实验报告里要给出Verilog HDL代码、测试向量代码、布线后仿真截图等必要信息。进行Behavioral（行为级）和Post-Route（布线后仿真）测试，对比结果。

【设计文件与实验过程】

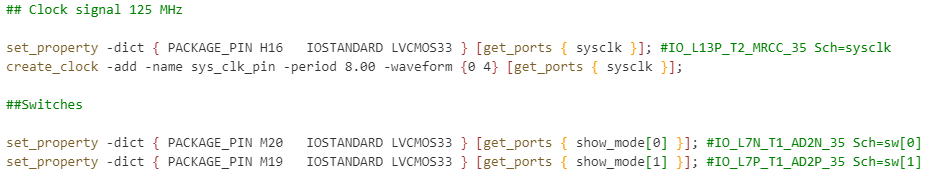
项目采用自顶向下设计方法，为实现功能规划如下子模块：

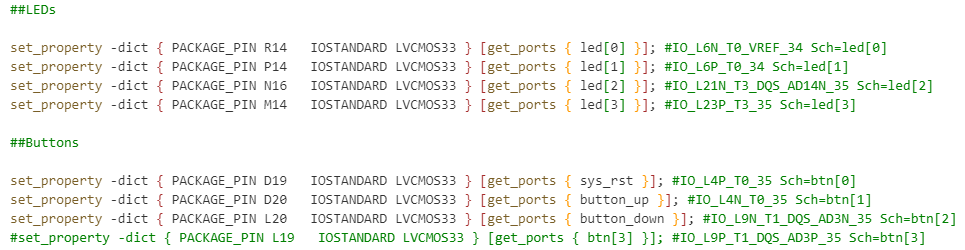


结构框图保存在工程目录**.src/report/report.dio**

【项目使用的板载资源】

约束文件如下所示





项目使用到四个LED显示四位二进制数；两个拨动开关SW0、SW1可以置为00、01、10、11四个值，指示要显示高四位还是低四位；两个按钮BTN1、BTN2用于控制要显示的变量。使用按钮BTN0实现**全局异步复位**。

比如显示小时，则可以通过按动按钮切换到小时显示模式，分别将拨动开关调到00和01来查看低四位值hour\_l和高四位值hour\_h。若要显示年份，可以按动按钮切换到年显示模式，分别将拨动开关调到00、01、10、11来显示y0[3:0]、y1[3:0]、y2[3:0]、y3[3:0]

系统时钟使用输入的125MHz时钟，直接供给分频器模块divider、时钟状态机模块rtc、显示状态机模块display、按键及防抖动模块key。闰年判断和输出译码器采用组合逻辑实现

【源文件divider.v与测试向量divider\_tb.v】

10ms时钟分频器源码如下所示：

localparam CLK\_10MS\_DIV = 32'd1\_250\_000;

reg [31: 0] cnt;

reg clk\_10ms\_r;

//T=10ms clk

always @(posedge sys\_clk\_125M) begin

    if (!sys\_rst\_n) begin

        cnt <= 32'd0;

        clk\_10ms\_r <= 1'b0;

    end

    else if (cnt == (CLK\_10MS\_DIV / 2 - 1)) begin

        cnt <= 32'd0;

        clk\_10ms\_r <= ~clk\_10ms\_r;

    end

    else begin

        cnt <= cnt + 1;

        clk\_10ms\_r <= clk\_10ms\_r;

    end

end

BUFG BUFG\_inst\_10m (

         .O(clk\_10ms), // Clock output

         .I(clk\_10ms\_r) // Clock input

     );

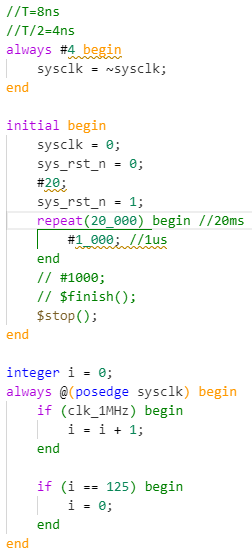
使用Xilinx的原语BUFG对时钟（）进行输出约束，对其中中高速时钟（1KHz、1MHz）进行额外的时序约束，xdc文件中约束如下所示：

# create\_clock -add -name clk\_1MHz\_w -period 100.00 [get\_pins rtc\_inst/u\_divider/clk\_1MHz]; # 1MHz == 100ns

# create\_clock -add -name clk\_1KHz\_w -period 100000.00 [get\_pins rtc\_inst/u\_divider/clk\_1KHz]; # 1KHz == 100ns

本项目中不需要使用这两个高速时钟，所以将其注释，只令10ms周期（100Hz）的低速脉冲通过一个BUFG

testbench文件divider\_tb.v如下图所示

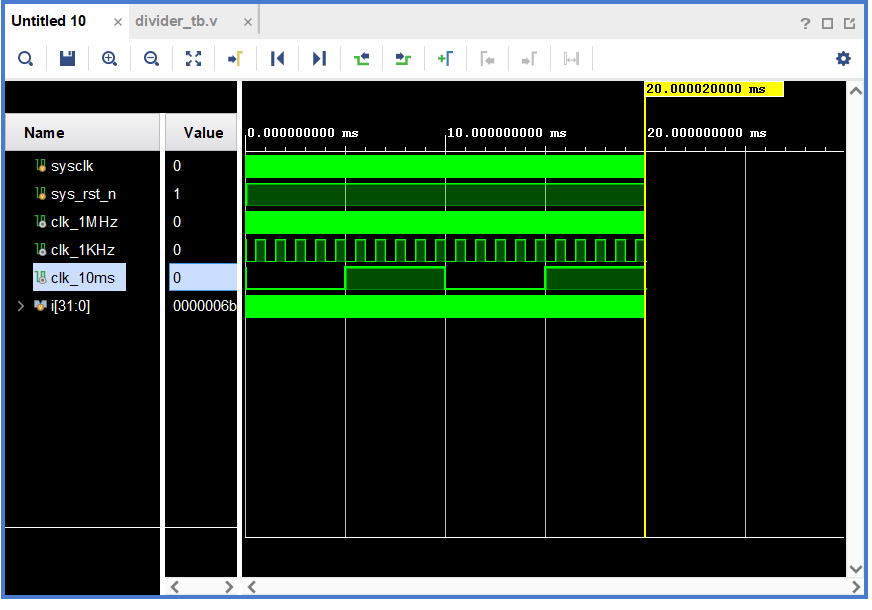


仿真共20ms，设置在20ms后自动结束（$stop();）

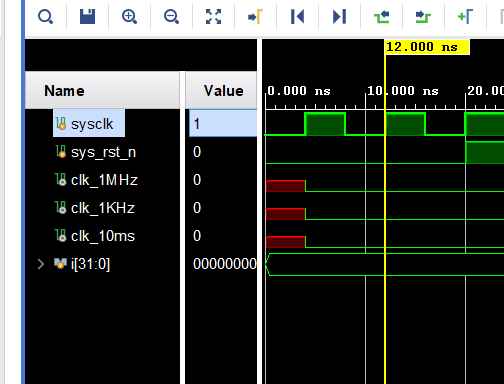
这里使用一个整数计数器i，每个sysclk到来时判断clk\_1MHz，如果为高则自增1，到达125归零

也就是当clk\_1Mhz为高结束前，i应当为125 / 2 + 1==63

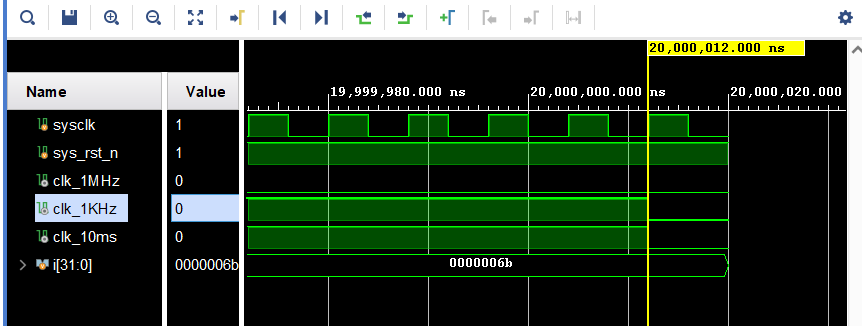
仿真结果如下所示：



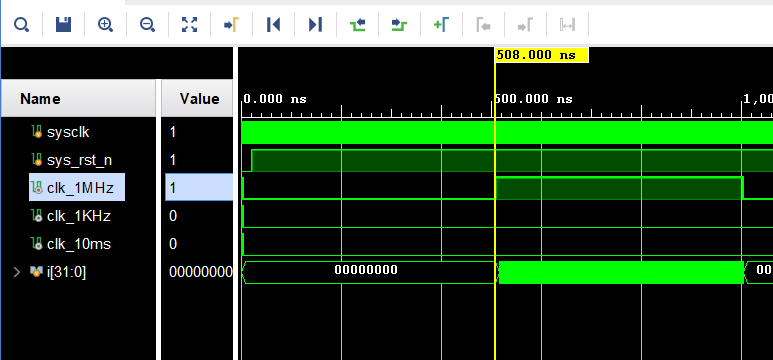
仿真片段如下：

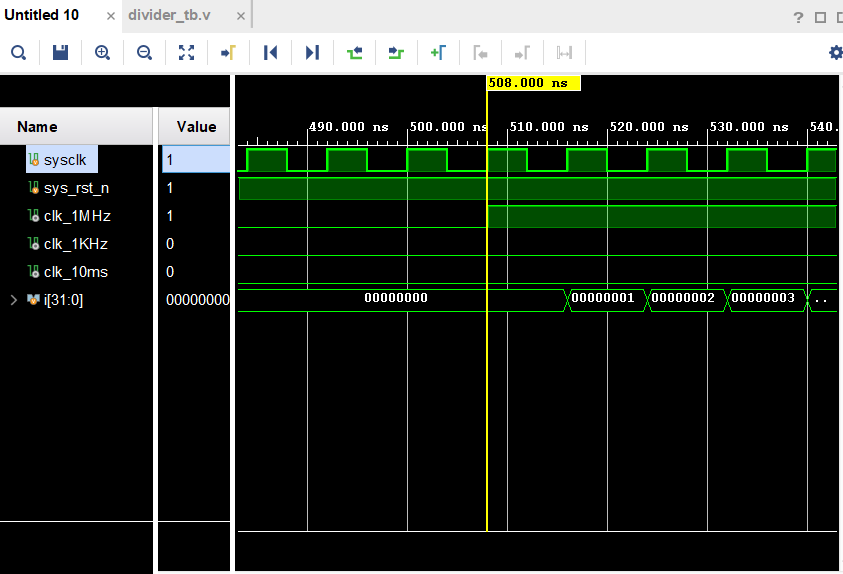


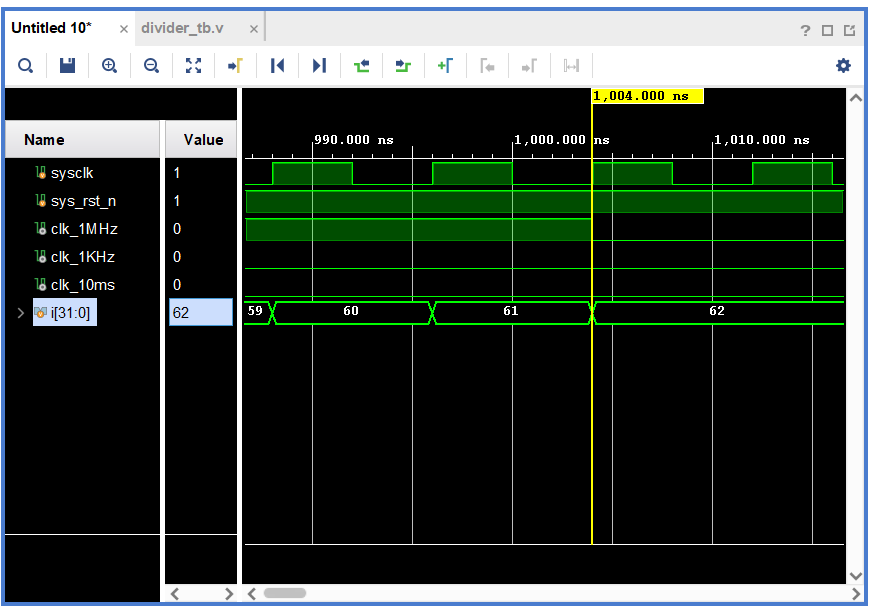
由于开始仿真后的sys\_rst\_n置1的时刻为20ns且在sysclk上升沿前到来，BUFG仿真延迟为一个时钟周期，因此10ms时钟延迟12ns



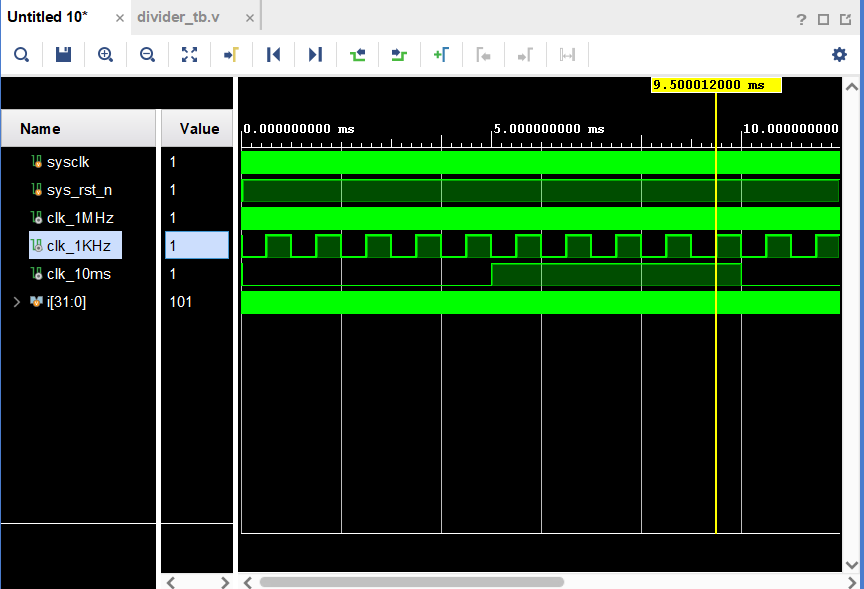
从下面的仿真中可以看到1KHz时钟可以稳定工作







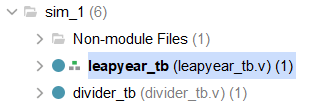
下面是1KHz时钟的仿真结果，每10ms有10个1KHz脉冲，符合要求



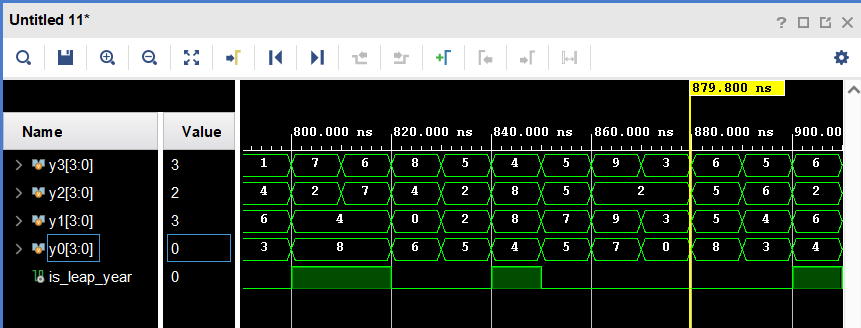
【源文件leapyear.v与测试向量leapyear\_tb.v】

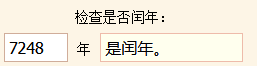
首先调整顶层文件为leapyear\_tb.v，运行10us后停止

如下图所示：

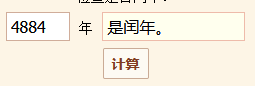


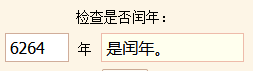
测试结果如下：











模块满足要求

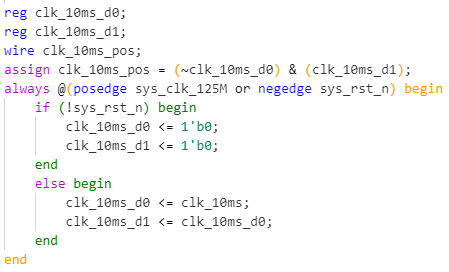
【源文件rtc.v与测试向量rtc\_tb.v】

rtc模块是leapyear和divider模块的顶层模块，采用一个行为级描述的时序逻辑计数器/选择器控制时钟逻辑变化。

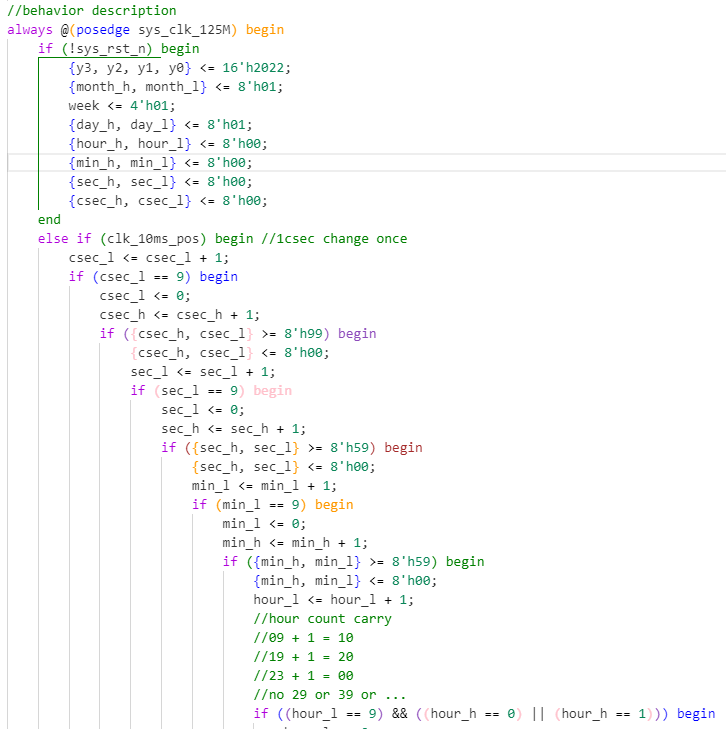
使用leapyear模块判断当年是否为闰年，从而控制二月天数变化。

使用divider模块输出的10ms时钟驱动百分秒变化，以百分秒溢出作为秒触发，以此类推完成年进位的控制逻辑。

由于divider模块的输出时钟不能直接用于百分秒检测，还加入了一个上升沿检测电路，使用系统时钟sys\_clk作为上升沿检测触发信号，这样判断信号clk\_10ms\_pos仅维持一个系统时钟，即可用于rtc逻辑判断，相关代码如下所示：

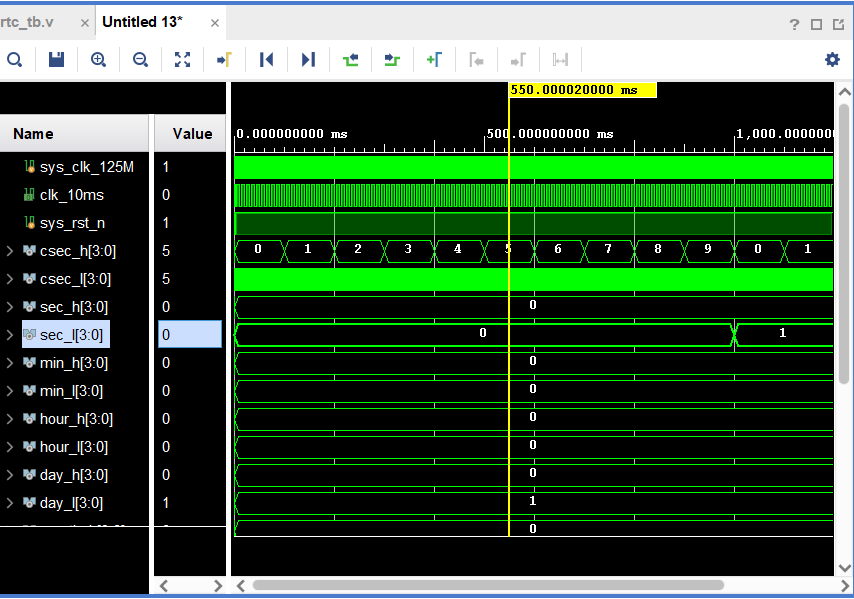


逻辑部分代码如下所示



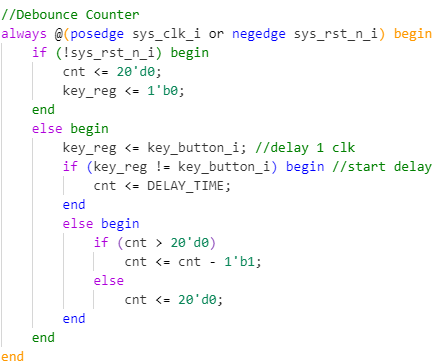
仿真结果如下所示：

这里仅仿真2s，模块实际效果在上板演示中给出



【源文件key.v】

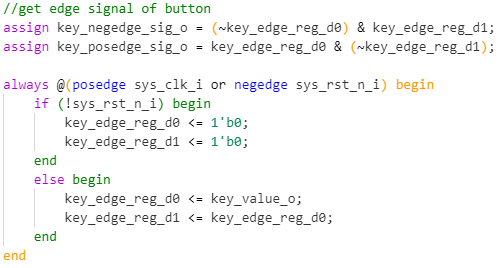
为处理板上按钮，加入key模块，提供按钮检测和按钮防抖动功能，同时能够处理按钮变化和上升沿、下降沿检测



模块主要使用debounce counter计数器实现按钮防抖动，延迟时间DELAY\_TIME

可以通过参数修改，初始固定为1\_000\_000时钟周期

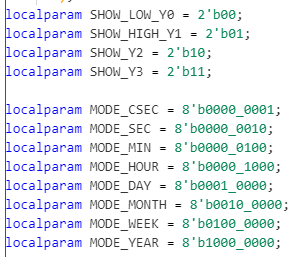
另有一个边沿检测移位寄存器，配合组合逻辑进行边沿捕获



模块较简单，故直接上板进行验证

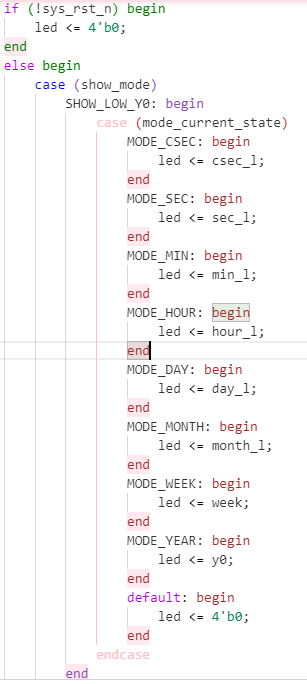
【源文件display.v与测试向量display\_tb.v】

显示模块是一个**三段式状态机**，由状态轮转时序逻辑、状态变换组合逻辑、输出多路选择器时序逻辑构成



状态机状态根据模式选择按钮给出的脉冲进行轮转变换，有百分秒、秒、分钟、小时、日期、月份、周、年份共8个状态。

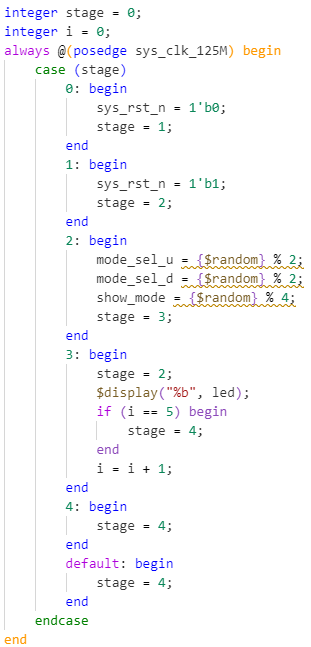
输出选择器会根据当前状态和输出多路选择器（复选信号由拨动开关输入）状态向led输出不同值，部分代码如下：



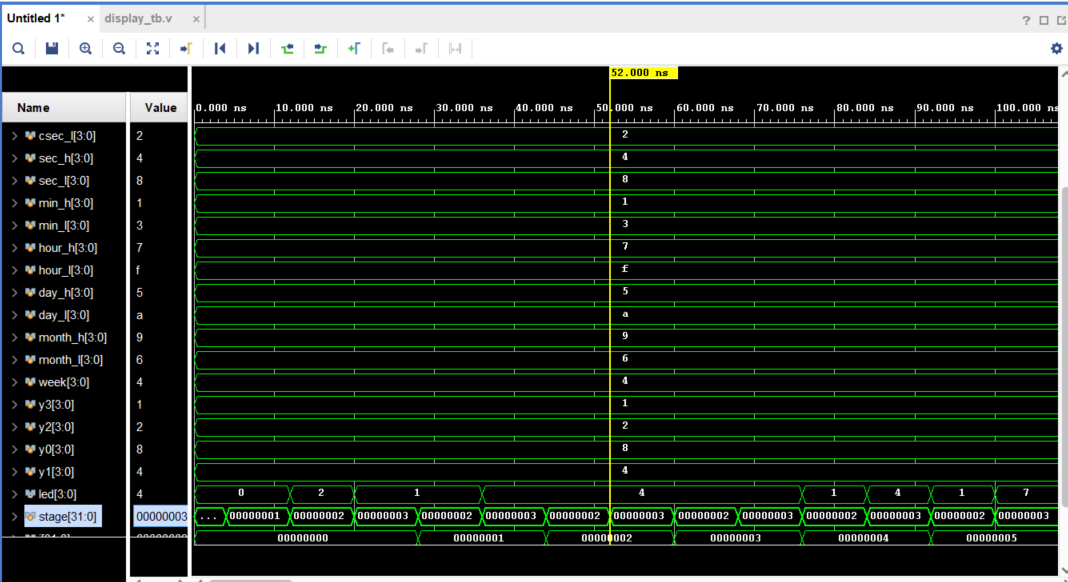
测试向量采用随机测试，将随机的show\_mode、mode\_sel\_d、mode\_sel\_u激励输入uut，人为设定每个字段的激励，使用$display()输出每轮led的显示值

测试向量一次测试中进行5轮，到达5轮输出后卡死，并根据Vivado要求时间停止。卡死时间用于测试激励不变时模块输出固定，防止状态机电路中可能的亚稳态

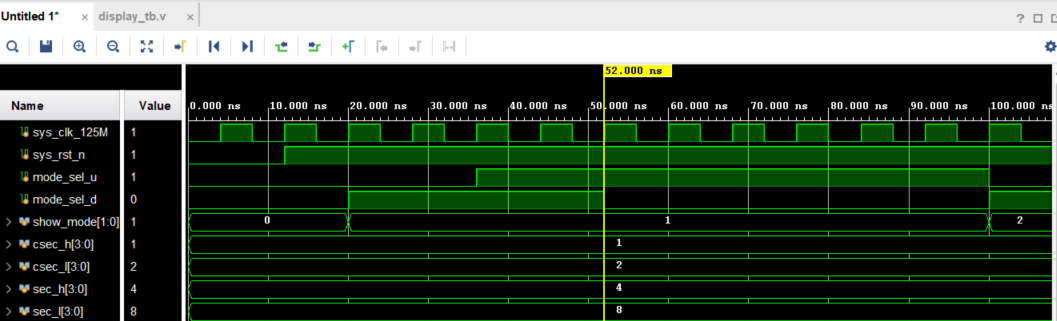
部分测试代码如下：



执行行为级仿真，输出结果如下



对应指令为：



在命令行中输出：

0001

0100

0100

0100

0100

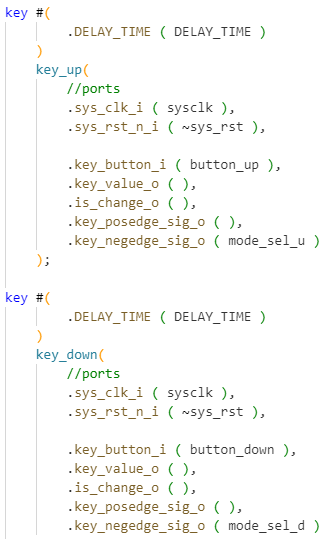
0111

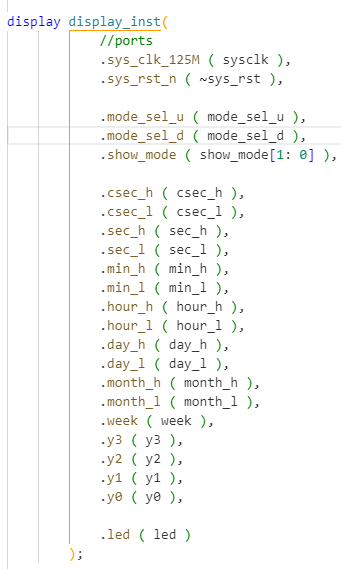
经过检验能满足描述要求

【源文件rtc\_top.v】

顶层模块放置在rtc\_top.v文件，分别根据开头给出的结构框图例化rtc模块、两个按钮模块、显示输出模块







这里输入的复位信号（按钮BTN0）默认为低电平，因此加入一个非逻辑

【结果分析】

将项目代码综合、实现后生成比特流，烧录到开发板，实验现象如下图和视频所示。

可见项目完成了既定要求，能够实现RTC实时时钟功能并以LED的形式展示当前输出

