浙江大学实验报告

课程名称：数字系统设计实验　　指导老师：屈民军、唐奕　　成绩：_____

实验名称：数字式秒表

一、实验目的和要求（必填）　　二、实验内容和原理（必填）

三、主要仪器设备（必填）　　四、操作方法和实验步骤

五、实验数据记录和处理　　六、实验结果与分析（必填）

七、讨论、心得

装

订

线

一、实验目的

（1）熟练掌握分频器、各种进制的同步计数器的设计。

（2）熟练掌握同步计数器的级联方法。

（3）掌握数码管的动态显示驱动方式。

（4）掌握计数器的功能和应用。

（5）理解开关防颤动的必要性。

（6）掌握简单控制器的设计方法。

二、实验内容与原理

（一）基本要求

设计一个数字秒表电路：

(1) 计时范围 0’0.0’’~9’59.9’’，分辨率 0.1s，用数码管显示计时值；

(2) 秒表设有一个功能按键开关 ButtonIn.

电路起始是处于“初始”状态时，通过第一次按键，开始自动地计时；再次按键，停止计时；第三次按键，计数器自动复位为 0’0.0’’，电脑回到“初始”状态。

（二）实验原理

1. 电路总体设计

(1) 秒表的电路的原理框图：

根据设计要求，可画出秒表电路的原理框图，如图 1 所示秒表电路由分频模块、按键处理模块、控制器、计时模块和动态显示模块组成。

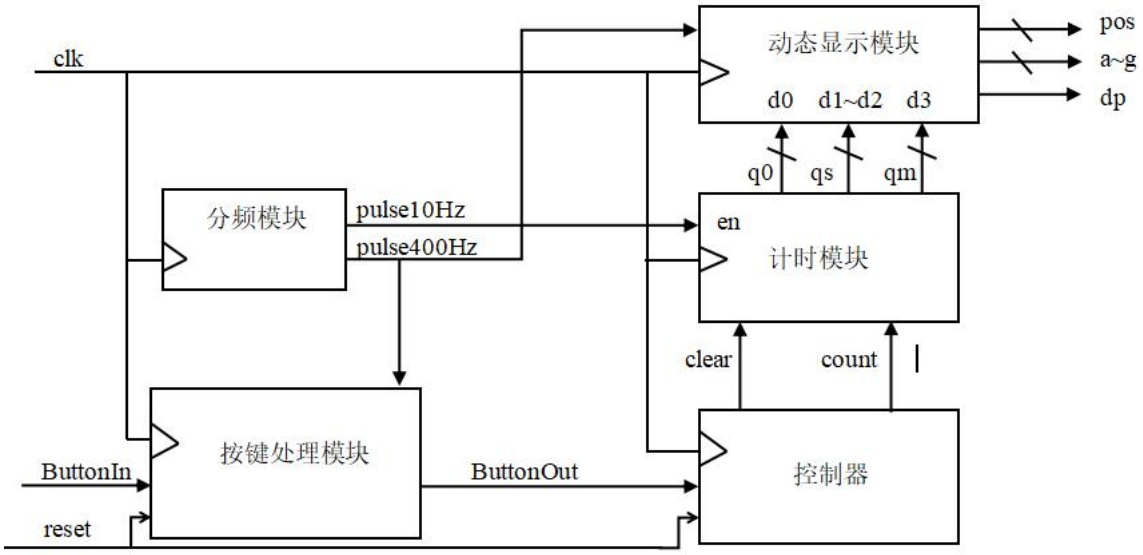


图 1 秒表电路的原理框图

(2) 设计思路

a. 由原理框图所示，操作者通过外部，每一次按键，信号 ButtonIn 输入按键处理模块，然后输出一个正脉冲信号 ButtonOut

b. ButtonOut 输入控制器，在按键控制下，控制器输出 clear 和 count 两个信号分别控制计时模块的工作状态。

c. 计时模块实现该秒表的计时功能，输出当前计时数 q0、qs、qm 到动态显示模块。显示的时候采用数码管动态显示技术，输出 pos 表示对应的数码管点亮；通过 a-g 的明灭，显示各位的数字；其中，本实验应该在右起第二位显示小数点，要求 dp=pos[1]

d. 分频模块分别产生 10Hz 和 400Hz 脉冲信号控制各模块工作

e. 电路时钟信号 clk，复位信号 reset 控制按键处理模块和控制器。

(3) Verilog HDL 代码描述

```
module stopwatch(ButtonIn,clk,reset,pos,a_g,dp);
    parameter sim=1'b0;         //便于仿真用的参数
    input reset,clk,ButtonIn;  //输入按键信号和复位信号
    output [6:0] a_g;          //控制七段译码管
    output dp;                 //控制小数点亮暗
    output [3:0] pos;          //控制四个数码管

/** 分频模块 产生 400Hz 和 10Hz 脉冲**/
    wire pulse400Hz,pulse10Hz;
    counter_n #(.n(sim?2:12_5000),.counter_bits(sim?1:17)) Div1(
          .co(pulse400Hz), ... );

    counter_n #(.n(sim?10:40),.counter_bits(sim?4:6)) Div2(
          .en(pulse400Hz),
```

```
          .co(pulse10Hz), ...);
/** 按键处理模块 **/
    wire ButtonOut;
    button_pro #(.sim(sim))  button_pro_inst(
        .clk(clk),
        .reset(reset),
        .ButtonIn(ButtonIn),
        .ButtonOut(ButtonOut));

/** 控制模块 **/
    wire count,clr;
    control control_inst(
        .in(ButtonOut),
        .reset(reset),
        .count(count),
        .clr(clr));

/** 计时模块 **/
    wire[3:0] q0,qm;
    wire[7:0] qs;
    timing  timer_inst(
        .en(count&&pulse10Hz),
        .clk(clk),
        .clr(clr),
        .q0(q0),
        .qs(qs),
        .qm(qm) ... );
/** 动态显示模块 **/

    seg7_display disp(
        .clk(clk),
        .d0(q0),
        .d1(qs[3:0]),
        .d2(qs[7:4]),
        .d3(qm),
        .scan(pulse400Hz),
        .pos(pos),
        .a_g(a_g),
        .dp(dp));

endmodule
```

三、主要仪器设备

1. 装有 Modelsim 、vivado 软件的计算机
2. Basys 3 开发板

四、实验设计过程

（一）按键处理模块

1. 实验目的

(1) 掌握减少亚稳态的方法，了解亚稳态给系统带来的危险

(2) 掌握开关颤动的概念和消除的方法

(3) 初步了解控制器的设计

2. 电路总体设计

(1) 实验组成框图

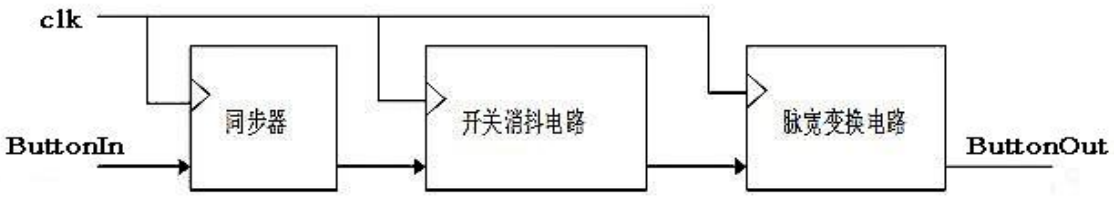


图 2 按键处理模块原理框图

(2) 电路功能概述

针对异步输入信号的特点，可采取以下措施解决：

① 输入增加同步器，减少触发器进入亚稳态状态的概率；

② 采用脉冲宽度变换电路，将异步输入信号的宽度变换为一个时钟周期；

③ 开关、按键输入，在同步器与脉冲宽度变换电路之间插入一个开关防颤动电路。

3. 子模块及其功能实现

(1) 同步器

异步设计，异步输入的时候，采用双锁存器的方法，将异步输入的信号用两个锁存器连续锁存两次，减少亚稳态问题，但同时给输入信号带来了延时。电路原理框图见图 3

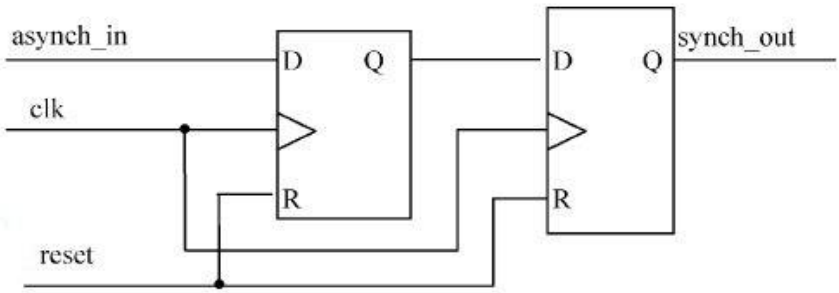


图 3 同步器原理框图

(2) 开关防颤动电路的设计

① 开关的颤动即开关防颤动电路的功能

人们完成一次按键操作的指压力如图 4 - (b)所示，按键开关从最初按下到接触稳态需要经过数毫秒的颤动，按键松开时也有同样的问题。因此，按键被按下或释放时，都有几毫秒的不稳定输出，从逻辑电平来看不稳定输出期间，其电平在“0”、“1”之间无规则摆动。因此，一次按键操作的输出如图 4 - (c)所示。一般情况，按键一次时间大于 100ms，颤动时间（按下或释放）小于 10ms。

电路目的：按一次按键，输出一个稳定的脉冲，即将开关的输出作为开关防颤动电路的输入，而开关防颤动电路的输出作为理想输出，如图 4 - (a)所示。

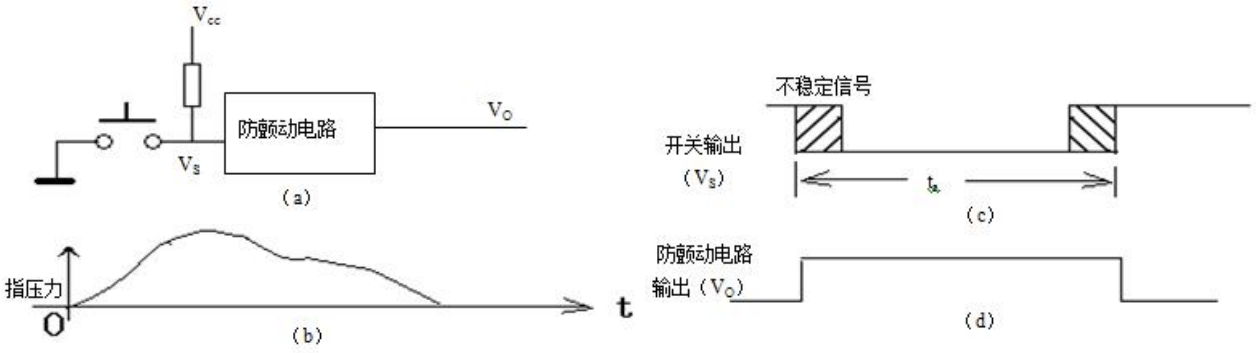


图 4 按键开关的颤抖

② 防颤电路原理框图

开关防颤动电路的关键是避免在颤动期采样，颤动期的长度一般是 10ms 左右。利用分频器将产生周期为 1ms 的脉冲信号 pulse1KHz。启动信号 timer_clr 由控制器提供，定时结束信号 timer_done 回馈给控制器。原理框图见图 5

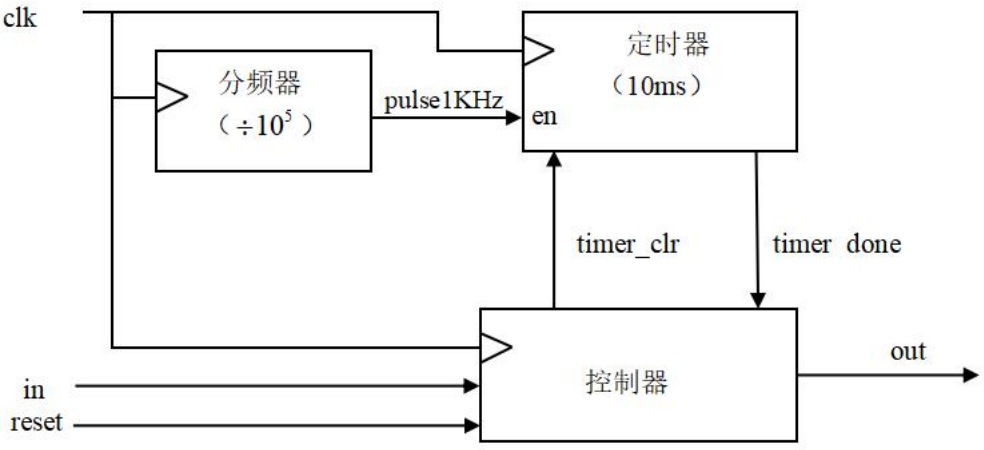


图 5 防颤动电路的原理框图

③ 防颤电路子模块设计

i) 控制器设计

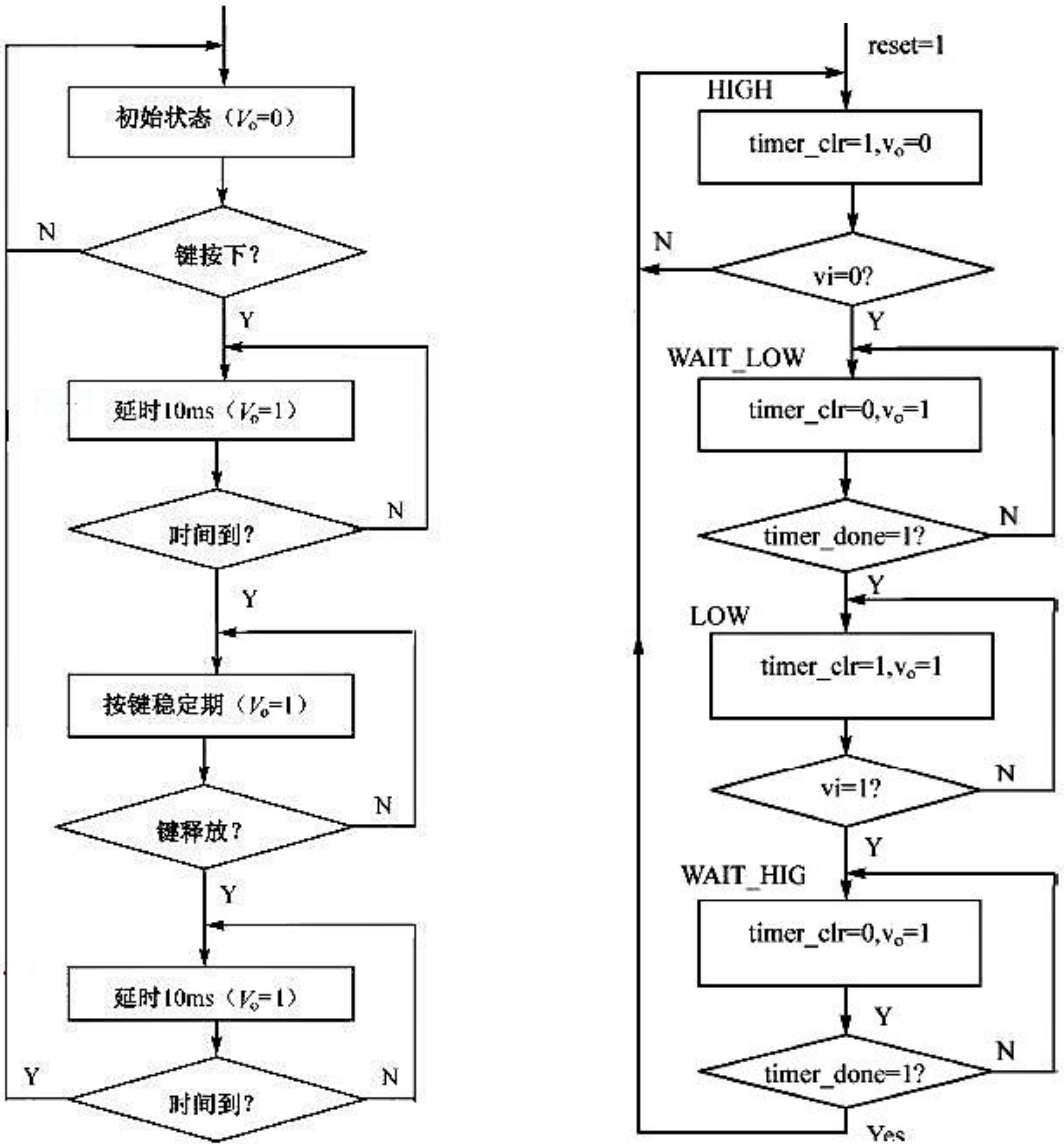


图 6 控制器工作流程图、算法流程图

ii) 定时器设计

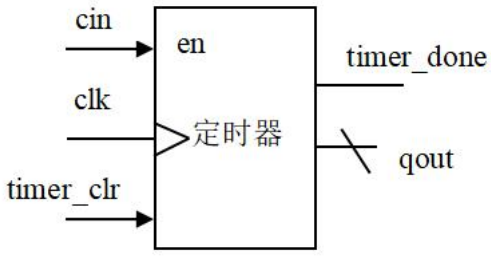


图 7 定时器框图

timer_clr	cin	clk	qout	功能
1	X	↑	qout* =0	清零（启动）
0	0		qout* =qout	保持
0	1		qout* =qout+1	计数

表 1 定时器功能表

(3) 脉冲变换电路的设计

开关或按键的输入信号宽度远远大于一个时钟周期，脉宽变换电路的作用是将输入信号宽度变换为一个时钟周期。由于输入已是同步的且宽度大于一个时钟周期的脉冲。脉宽变换电路的原理图如图 8 所示

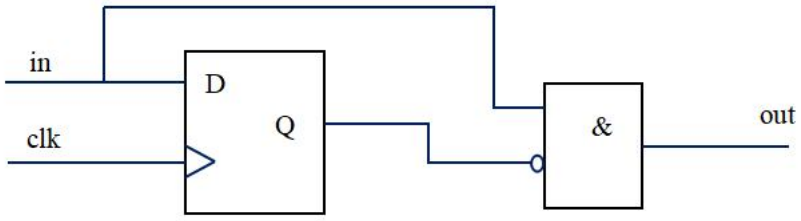


图 8 脉宽变换电路

（二）分频模块

1. 电路原理

分频器实际上就是计数器，分频比 n 就是计数器的模。

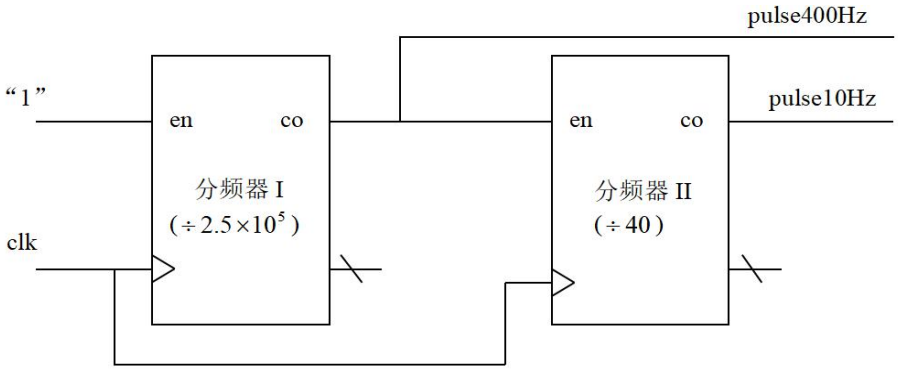


图 9 分配模块原理图

2. 功能描述

该模块产生用于计时的 0.1s 脉冲信号 pulse10Hz；产生用于显示模块的扫描脉冲信号 pulse400Hz。脉冲宽度均为一个系统主时钟信号 clk 的周期。

（三）控制器

控制器是电路的核心，在按键的控制下，输出 clr、count 两个控制信号分别控制计时模块的工作状态。控制器的算法流程图如图 10 所示，RESET、TIMING、STOP 分别代表秒表的初始、计时和停止三个状态。

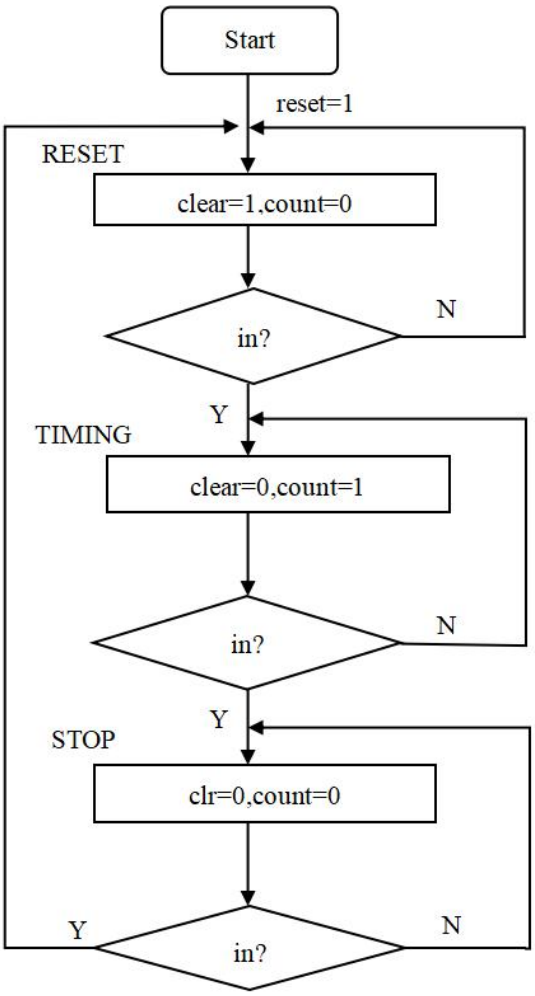


图 10 控制器 ASM 图

（四）计时模块

1. 计时模块电路设计

计时模块可由 2 个 10 进制计数器（0.1s 计时和 1min 计时）、1 个 60 进制 BCD 码计数器（精度 1s）级联而成。计时模块的输入控制信号 clr、count 为互斥信号。计时模块的原理框图如图 11 所示，其功能表如表 2 所示。

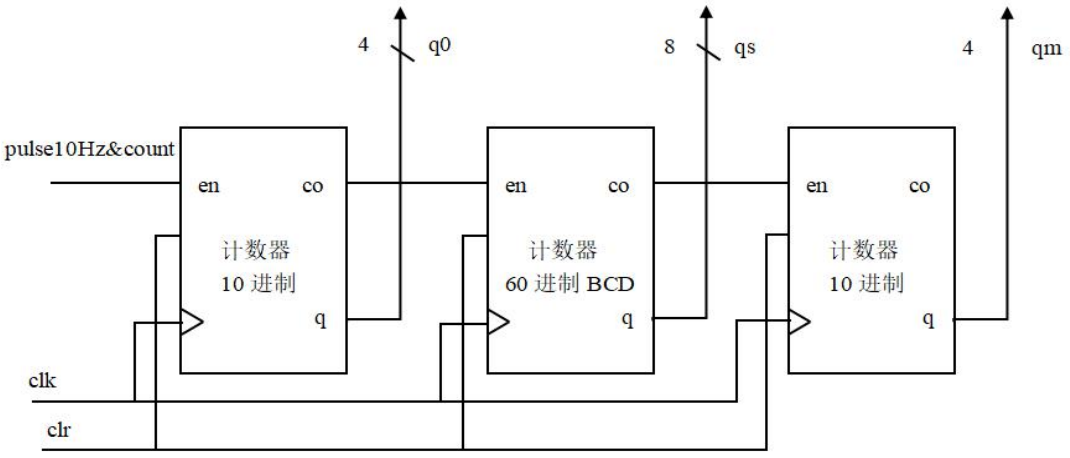


图 11 计时模块原理图

clr	count	pulse10Hz	clk	功能
1	×	×	↑	同步清零
0	0	×		保持
0	1	0		保持
0	1	1		计数

表 2 计时模块功能表

（五）动态显示模块

1. 电路原理

显示采用数码管动态显示技术，即所有数码管公用一组数据线（a~g），数码管轮流被点亮。因此，动态显示驱动方式中每一个数码管都要有一个点亮控制输入端，该端口即为数码管的共阴极端或共阳极端。本试验 LED 数码显示器为共阳极数码管，采用反相驱动，因此位选信号低电平有效。

“0.1 秒”、“秒”、“分”共有 4 位 BCD 码需显示。四进制计数器状态 q 控制数据选择器一次选出当前显示的 BCD 码（din）送入到显示译码器。计数器状态 q 同时表征显示在那个数码管上，通过 2-4 线译码器输出 pos（position）控制对应的数码管点亮。电路原理图见图 12

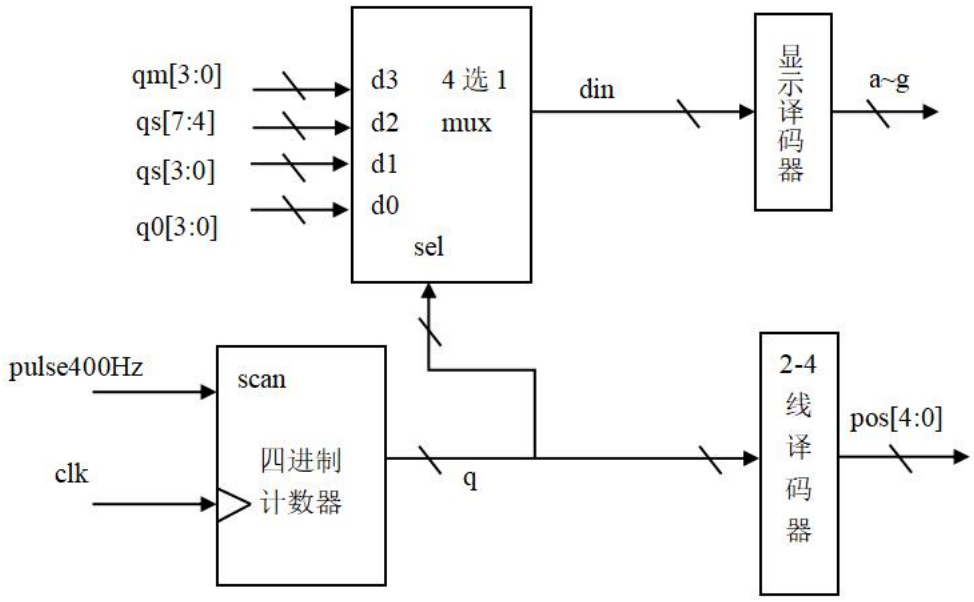


图 12 显示模块原理图

2. 显示译码器功能实现

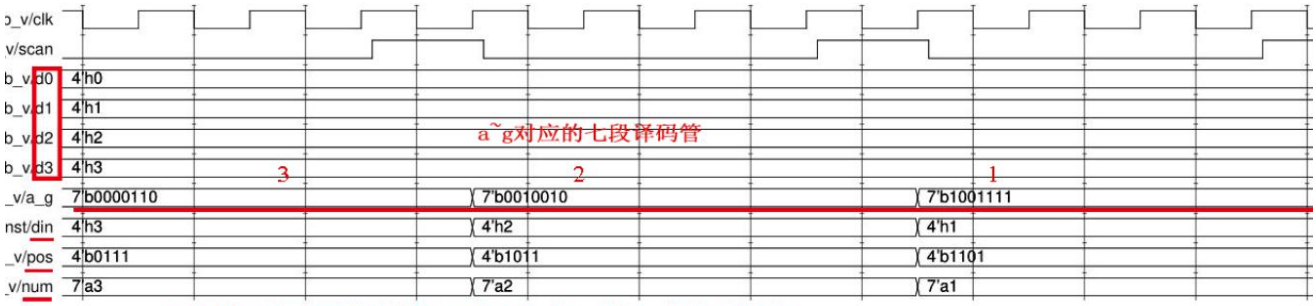
显示译码器输出的 a~g 信号（低电平有效）控制数码管亮灭，输入禁用码时，数码管全灭。其功能表见表 3

输入				输出							字形
D	C	B	A	F_a	F_b	F_c	F_d	F_e	F_f	F_g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

表 3 七段译码器功能表

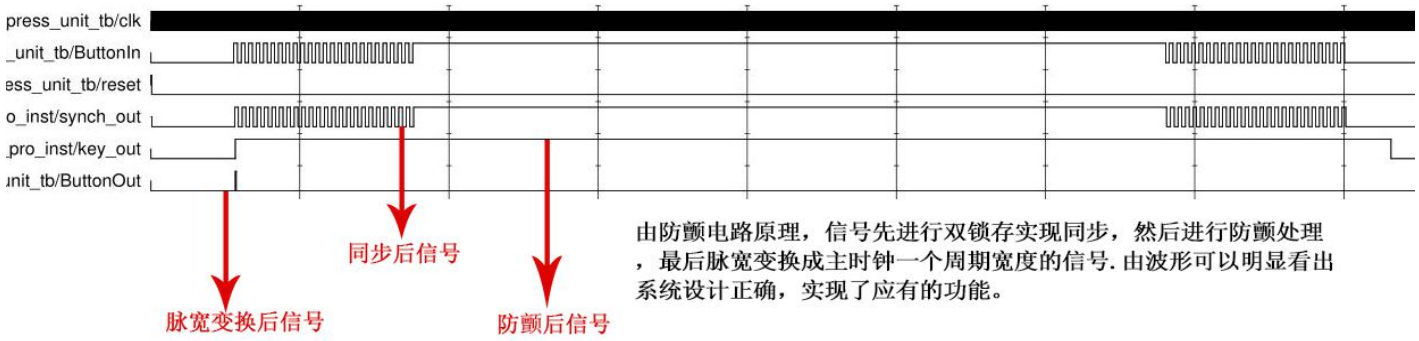
五、仿真分析

（一）Lab10 学号显示



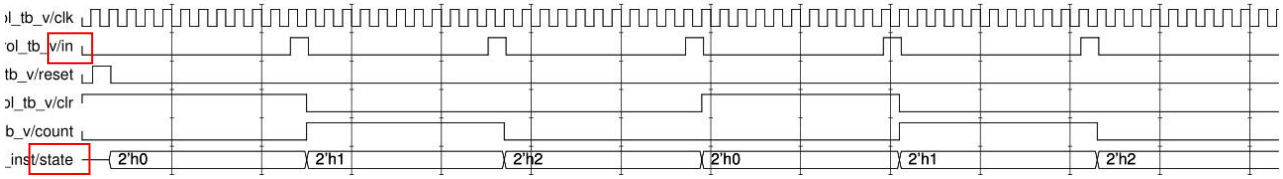
din为输出显示译码器的信号，理论上din与num相同，仿真结果正确。
num为pos信号对应数码管的数字，pos从0111~1101即从左边数码管到右边依次点亮，对应数字分别显示3、2、1，仿真结果正确。

（二）Lab11 防颤电路仿真



（三）Lab12 电路仿真

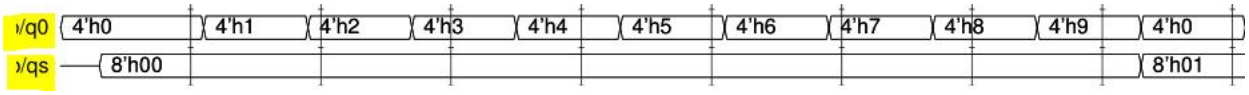
1. 控制器仿真



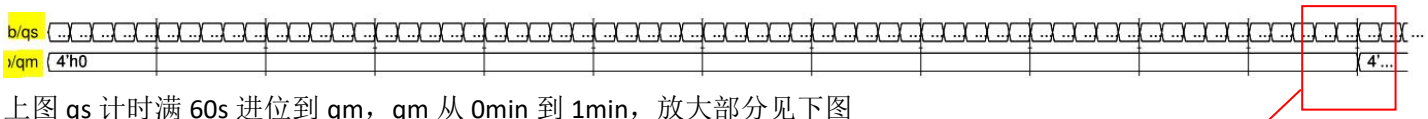
图中，state 即为当前状态。state 为 0,1,2 分别对应的是状态 RESET（复位）、TIMING（计时）、STOP（停止），信号 in 为按键输入，每按一次键，状态 state 就改变状态。

因此由仿真结果可见，控制器设计正确

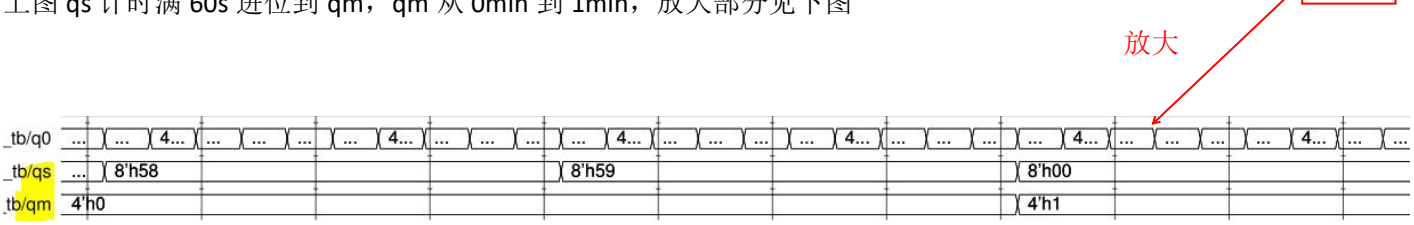
2. 定时器仿真



上图：q0 的精度 0.1s ， 计数 10 后进位到 qs ， qs 变成 1s

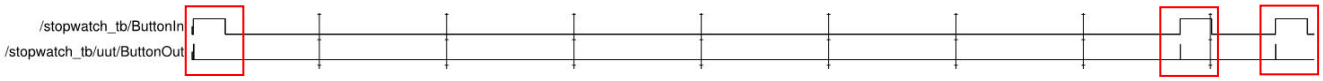


上图 qs 计时满 60s 进位到 qm，qm 从 0min 到 1min，放大部分见下图

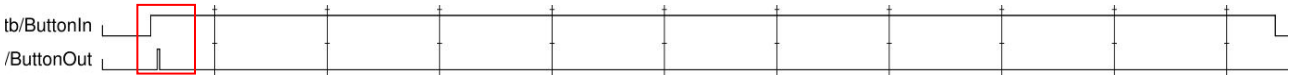


3. 顶层文件（秒表）仿真

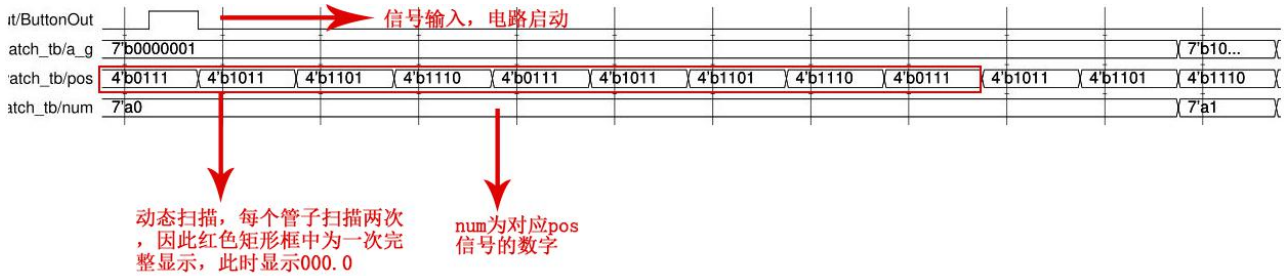
1）首先看按键处理部分。操作者按键输出的 ButtonIn 经过处理后变成理论上能够控制电路的短脉冲信号 ButtonOut.



放大图：



2）计时仿真



上图为仿真波形一部分，在 modelsim 中看后面的波形（未截图），电路功能正确。

（四）vivado 综合电路

下面均为 vivado 综合电路，与最初设计的原理图一样，这也是一个验证实验是否正确的好方法。

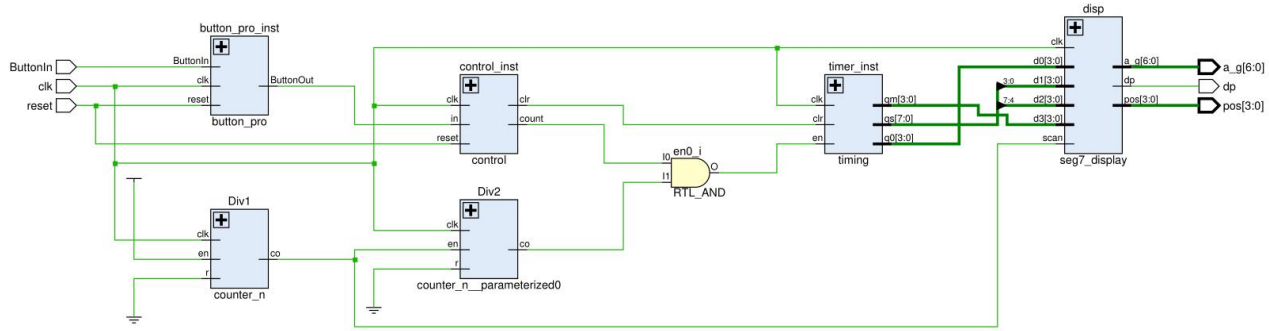


图 13 数字秒表电路（总电路）

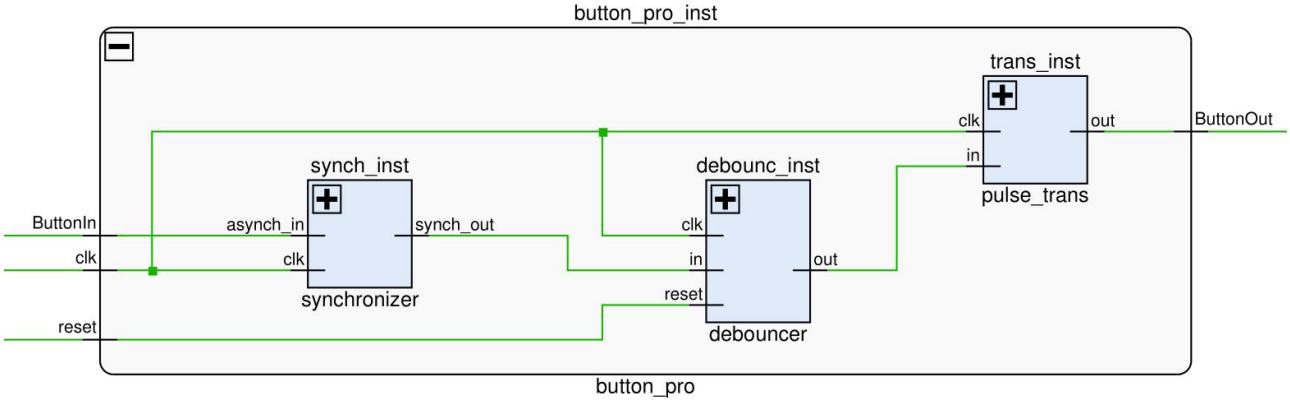


图 14 按键处理电路

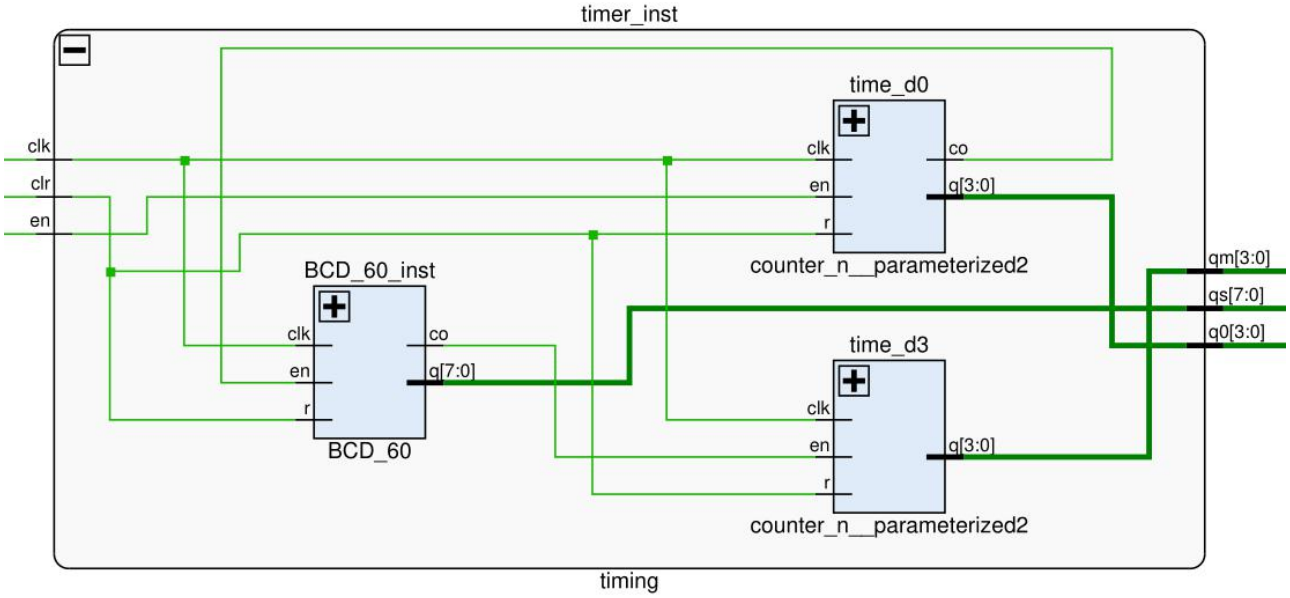


图 15 定时电路

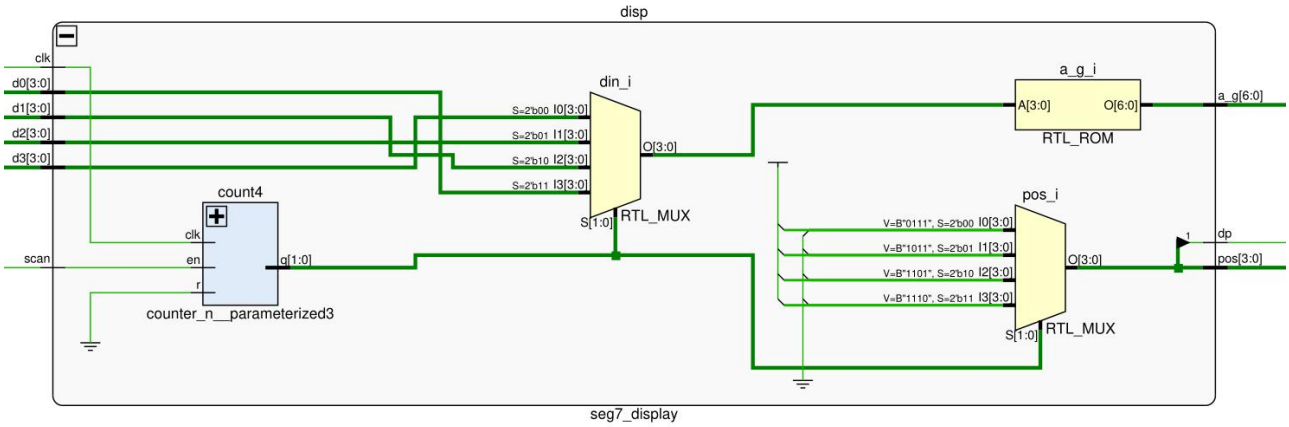


图 16 动态显示电路

六、 心得体会及思考题

（一）实验中遇到的问题及解决方法

1. **问题** ： 参数过多，常常把子模块的参数当成顶层模块的参数用于接口传递，为此导致了许多错误，其中最严重的是综合工具无法识别 clk 系统主时钟，因此无法进行时钟约束。

解决办法一：对照已经设计好的顶层模块和子模块的原理图，在调用子模块时，按照原理图上的参数来进行各个接口的传递；或者

解决办法二：代码写好后，进行电路综合，看每一个模块的接口是否没有数据传入/传出，说明该接口连接错误，再据此修改代码

2. **问题** ：写代码不规范，例如：count_n 实例的接口使能端 en 是’1’

我的代码写的是

… ,en(1),…);

然而在 vivado 电路综合时出现以下警告：

```
zer.v(15): [PCDPC] - Port size (1) does not match connection size (32) for port 'r'. The port definition

/synch_inst/ff2 File: C:/Users/xps/Desktop/Lab_12/button_pro/dffre.v
.v(20): [PCDPC] - Port size (1) does not match connection size (32) for port 'r'. The port definition is

/debounc_inst/Div File: C:/Users/xps/Desktop/Lab_12/timer/counter_n.v
ns.v(8): [PCDPC] - Port size (1) does not match connection size (32) for port 'en'. The port definition
```

提示电路综合接口位数不匹配，原因是自己传递数据时没有定义位数，对于我们这种初学者，需要注明参数位数，否则容易出错

解决办法：修改代码成 … ,en(1’b1),…);

3. **问题** ：引脚分配问题。在学号滚动显示实验中，数码管能亮，但是无法分辨数字，而且顺序从左到右滚动，与期望的从右到左方向相反。这说明自己 a~g 的引脚分配错误，以及 pos 信号出错。

解决办法：为了程序方便我将 a~g 定义成 reg[6:0] a_g 在对应管脚分配时，a~g 顺序弄反因此出现显示错误数字的结果。重新分配管脚就解决了这个问题。

以及 pos 信号（红色部分即为自己的错误所在）：

错误代码：

```
/** 2-4 译码 数码管选择 **/
  always @(*)
    begin
      case(sel)
       0: pos=4'b1110;
       1: pos=4'b1101;
       2: pos=4'b1011;
       3: pos=4'b0111;
      endcase
    end
```

正确代码：

```
/** 2-4 译码 数码管选择 **/
  always @(*)
    begin
      case(sel)
       0: pos=4'b0111;
       1: pos=4'b1011;
       2: pos=4'b1101;
       3: pos=4'b1110;
      endcase
    end
```

4. **问题**：在编写控制器模块代码时，发现教材里面状态机的描述中，顶层模块一般没有给 reset 赋初值，自己就很疑惑：那在人没有按下 reset 时如何决定控制器的初始状态？

解决：经查阅资料和询问老师，了解到硬件上会有上电复位电容，但凡刚启动，都会自动复位，因此不必担心电路进入错误的初始状态。

（二）收获

1. 在解决上述问题的过程中已经收获了许多自主解决问题的经验，经过电路设计，我意识到了开设实验课程以及提升动手能力的重要性，看似简单的几幅原理图，可能需要的是我们无数次的调试和修正才能完成。

2. 在把数据下载到电路板之前仿真是非常重要的，真正实现电路功能花费的时间比仿真多得多，然而如果你进行仿真，通过波形，综合电路图等方法发现错误，然后改正，比下到板子上节约了许多时间和精力，因此一定要提前仿真，保证结果正确。

3. 随着数电理论课进度的深入，以及前几次的多次练习，自己也渐渐掌握 Verilog HDL 的语法结构等。其次是对 modelsim、vivado 软件的运用，从生疏到熟练，从照着书上的步骤一步步点击，到能够自己独立进行操作，真正是熟能生巧。但自己也只是学到了基本的使用方法，软件还有许多功能等待自己去摸索。

（三）思考题

为什么计时模块采用 BCD 码计数方式？

因为是 BCD 数码管显控，四位 BCD 码对应一位数字，然后再译码通过 a~g 控制管子。