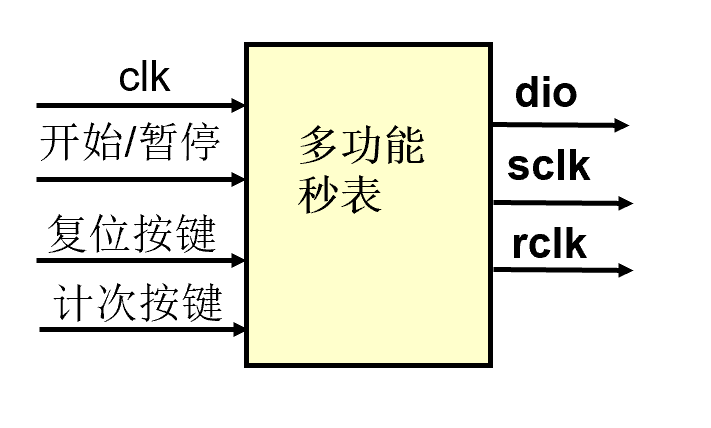
**实验五 电子秒表设计与实现**

1. **实验内容**

1.1设计要求

采用自顶向下的设计方法，完成多功能秒表的设计与实现，其功能包括计时与计次，输出用外扩8个数码管显示。

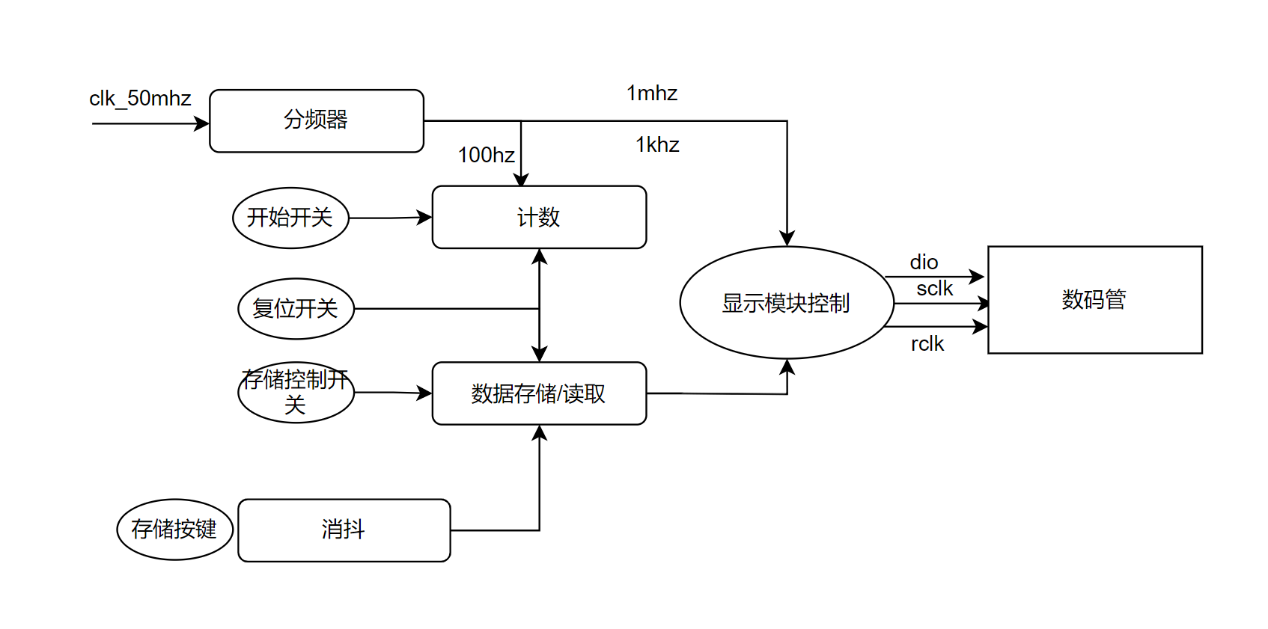


1.2设计指标

* 秒表精度为0.01秒
* 秒表计时范围为：1小时
* 设置开始计时/停止计时、复位两个按钮
* 显示工作方式：使用六位BCD七段数码管显示读数，显示格式00-00-00
* 扩展功能：按键消抖，计次：分别存储三次以上时间并分时回放显示。

1. **实验原理与方案**

2.1系统组成



2.2计数器

采用十进制计数器和六进制计数器异步级联方式实现1小时计时，最大显示59-59-99，实现秒表的精度为0.01秒，设置开始/暂停和复位两个开关，来控制计数器计数。

2.3存储功能

使用三个24位数的寄存器实现按键按下时记录当前秒表值，当存储使能开关无效时，再次按下按键会显示存储的秒表值，存储使能开关有效时，继续计数。

2.4外扩8位数码管显示

显示控制模块收到要显示的数据后，使用数码管动态扫描的方式产生并行的位选和段选信号，使用1mhz时钟来将并行信号转化为串行信号，并通过dio送出，实现外扩数码管的动态显示功能。

1. **实验过程(源程序)**

3.1秒表顶层模块

module miaobiaotop(

clk\_50mhz,

switch0,

switch1,

switch2,

btn0,

rclk,

sclk,

dio

);

//端口

input switch0,switch1,switch2,btn0;//使能、复位、读取显示、存储\读取

input clk\_50mhz;

output rclk;//外扩数码管接口

output sclk;

output dio;

//分频

wire clk100hz;

wire clk1mhz;

wire clk1khz;

div100hz clk0(

.clk\_50mhz(clk\_50mhz),

.clk100hz(clk100hz)

);

div1mhz clk1(

.clk\_50mhz(clk\_50mhz),

.clk1mhz(clk1mhz)

);

div1khz clk2(

.clk\_50mhz(clk\_50mhz),

.clk1khz(clk1khz)

);

//计数

wire [3:0] q0;

wire [3:0] q1;

wire [3:0] q2;

wire [3:0] q3;

wire [3:0] q4;

wire [3:0] q5;

wire carry\_out1;

wire carry\_out2;

wire carry\_out3;

wire carry\_out4;

wire carry\_out5;

wire over;

counter10 u1(

.clk100hz(clk100hz),

.rst(switch1),

.en(switch0),

.cnt(q0),

.carry\_out(carry\_out1)

);

counter10 u2(

.clk100hz(carry\_out1),

.rst(switch1),

.en(switch0),

.cnt(q1),

.carry\_out(carry\_out2)

);

counter10 u3(

.clk100hz(carry\_out2),

.rst(switch1),

.en(switch0),

.cnt(q2),

.carry\_out(carry\_out3)

);

counter6 u4(

.clk100hz(carry\_out3),

.rst(switch1),

.en(switch0),

.cnt(q3),

.carry\_out(carry\_out4)

);

counter10 u5(

.clk100hz(carry\_out4),

.rst(switch1),

.en(switch0),

.cnt(q4),

.carry\_out(carry\_out5)

);

counter6 u6(

.clk100hz(carry\_out5),

.rst(switch1),

.en(switch0),

.cnt(q5),

.carry\_out(over)

);

//存储

reg [23:0]data\_memory0=0;

reg [23:0]data\_memory1=0;

reg [23:0]data\_memory2=0;

reg [23:0]memory=0;

reg [2:0] cnt\_mem=0;

wire key\_xd;

xiaodou x0(

.clk100hz(clk100hz),

.btn0(btn0),

.key\_out(key\_xd)

);

always@(negedge key\_xd or posedge switch2)

begin

if(switch2==1)begin

cnt\_mem<=2;

end

else if(cnt\_mem==0)

cnt\_mem<=2;

else

cnt\_mem<=cnt\_mem-1;

end

//移位

always@(negedge key\_xd)

begin

if(switch2==1)

begin

data\_memory0<={q5,q4,q3,q2,q1,q0};

data\_memory1<=data\_memory0;

data\_memory2<=data\_memory1;

end

end

//数码管动态显示

wire [23:0] dispnum;

reg[23:0] disnum\_tmp;

assign dispnum=disnum\_tmp;

always@(posedge rclk)

begin

if(switch2==1)

disnum\_tmp<={q5,q4,q3,q2,q1,q0};

else

case(cnt\_mem)

0:disnum\_tmp<=data\_memory0;

1:disnum\_tmp<=data\_memory1;

2:disnum\_tmp<=data\_memory2;

endcase

end

leddisplaynew d0(

.rclk(rclk),

.sclk(sclk),

.clk1mhz(clk1mhz),

.clk1khz(clk1khz),

.dio(dio),

.dispnum(dispnum)

);

endmodule

3.2分频

module div100hz(

clk\_50mhz,

clk100hz

);

input clk\_50mhz;

output reg clk100hz=0;

integer cnt2=1;

always@(posedge clk\_50mhz)

begin

if(cnt2==250000)//25000000

begin

cnt2=1;

clk100hz=~clk100hz;

end

else

cnt2=cnt2+1;

end

endmodule

module div1mhz(

clk\_50mhz,

clk1mhz

);

input clk\_50mhz;

output reg clk1mhz=0;

integer cnt4=1;

always@(posedge clk\_50mhz)

begin

if(cnt4==25)//25000000

begin

cnt4=1;

clk1mhz=~clk1mhz;

end

else

cnt4=cnt4+1;

end

endmodule

module div1khz(

clk\_50mhz,

clk1khz

);

input clk\_50mhz;

output reg clk1khz=0;

integer cnt3=1;

always@(posedge clk\_50mhz)

begin

if(cnt3==25000)//25000000

begin

cnt3=1;

clk1khz=~clk1khz;

end

else

cnt3=cnt3+1;

end

endmodule

3.3计数

module counter10(

rst,

clk100hz,

en,

cnt,

carry\_out

);

input rst,clk100hz,en;

output reg carry\_out;

output reg[3:0] cnt=0;

always@(posedge clk100hz or negedge rst)

begin

if(!rst)

begin

cnt=0;

carry\_out=0;

end

else if(en)

begin

if(cnt==9)

begin

cnt=0;

carry\_out=1;

end

else

begin

cnt=cnt+1;

carry\_out=0;

end

end

end

endmodule

module counter6(

rst,

clk100hz,

en,

cnt,

carry\_out

);

input rst,clk100hz,en;

output reg carry\_out;

output reg[3:0] cnt=0;

always@(posedge clk100hz or negedge rst)

begin

if(!rst)

begin

cnt=0;

carry\_out=0;

end

else if(en)

begin

if(cnt==5)

begin

cnt=0;

carry\_out=1;

end

else

begin

cnt=cnt+1;

carry\_out=0;

end

end

end

endmodule

3.4消抖

module xiaodou(

clk100hz,

btn0,

key\_out

);

input btn0;

input clk100hz;

output key\_out;

reg tmp1,tmp2;

always @(posedge clk100hz) begin

tmp1 <= btn0;

tmp2 <= tmp1;

end

assign key\_out = tmp1 | tmp2;

endmodule

3.5数码管显示

module leddisplaynew(

dispnum,

rclk,

sclk,

dio,

clk1mhz,

clk1khz

);

input[23:0] dispnum;

output reg rclk;

output sclk;

output reg dio;

input clk1mhz;

input clk1khz;

//数码管动态显示

reg[7:0] dig;

reg[7:0] seg;

reg[2:0] count1=0;

always@(posedge clk1khz)

begin

if(count1>=5)

count1=0;

else

count1=count1+1;

end

always@(count1)

begin

case(count1)

0: dig=8'b00000001;//00000001

1: dig=8'b00000010;//00000010

2: dig=8'b00000100;//00000100

3: dig=8'b00001000;//00001000

4: dig=8'b00010000;

5: dig=8'b00100000;

default: dig=8'b00000000;

endcase

end

reg[3:0] disp\_data;

always@(count1)

begin

case(count1)

0: disp\_data=dispnum[3:0];

1: disp\_data=dispnum[7:4];

2: disp\_data=dispnum[11:8];

3: disp\_data=dispnum[15:12];

4: disp\_data=dispnum[19:16];

5: disp\_data=dispnum[23:20];

default: disp\_data=0;

endcase

end

always@(disp\_data)

begin

case(disp\_data)

0: seg<=8'b00000011;

1: seg<=8'b10011111;

2: seg<=8'b00100101;

3: seg<=8'b00001101;

4: seg<=8'b10011001;

5: seg<=8'b01001001;

6: seg<=8'b01000001;

7: seg<=8'b00011111;

8: seg<=8'b00000001;

9: seg<=8'b00011001;

default: seg<=8'b11111111;

endcase

end

//外扩数码管并串转换

assign sclk=clk1mhz;

reg[3:0] cnt1=0;

always@(negedge clk1mhz)

begin

if(cnt1==15)

begin

cnt1<=0;

rclk<=1'b1;

end

else

begin

cnt1<=cnt1+1;

rclk<=1'b0;

end

end

always@(cnt1)

begin

case(cnt1)

0:dio<=seg[0];

1:dio<=seg[1];

2:dio<=seg[2];

3:dio<=seg[3];

4:dio<=seg[4];

5:dio<=seg[5];

6:dio<=seg[6];

7:dio<=seg[7];

8:dio<=dig[7];

9:dio<=dig[6];

10:dio<=dig[5];

11:dio<=dig[4];

12:dio<=dig[3];

13:dio<=dig[2];

14:dio<=dig[1];

15:dio<=dig[0];

default: dio<=0;

endcase

end

endmodule

1. **实验结果与分析(仿真程序与仿真图，硬件测试图)**

4.1仿真程序

module miaobiao\_testbench;

// Inputs

reg clk\_50mhz;

reg switch0;

reg switch1;

reg switch2;

reg btn0;

// Outputs

wire rclk;

wire sclk;

wire dio;

// Instantiate the Unit Under Test (UUT)

miaobiaotop uut (

.clk\_50mhz(clk\_50mhz),

.switch0(switch0),

.switch1(switch1),

.switch2(switch2),

.btn0(btn0),

.rclk(rclk),

.sclk(sclk),

.dio(dio)

);

initial begin

// Initialize Inputs

clk\_50mhz = 0;

switch0 = 0;

switch1 = 0;

switch2 = 0;

btn0 = 0;

// Wait 100 ns for global reset to finish

#100;

// Add stimulus here

switch0=1;

switch1=1;

switch2=1;

end

always

begin

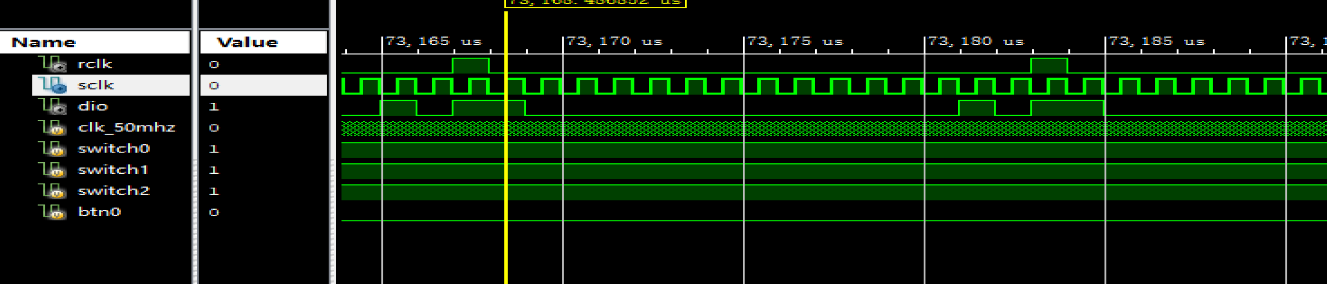
#10;

clk\_50mhz=~clk\_50mhz;

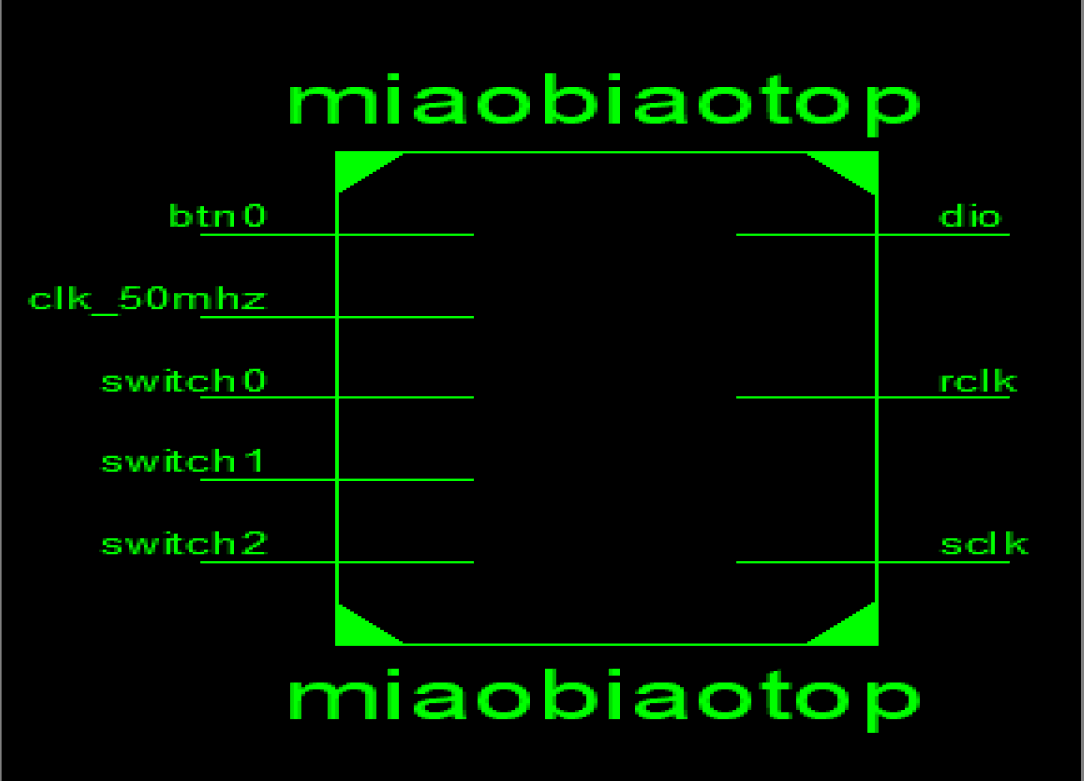
end

endmodule

4.2仿真图



4.3RTL



4.4硬件测试图

