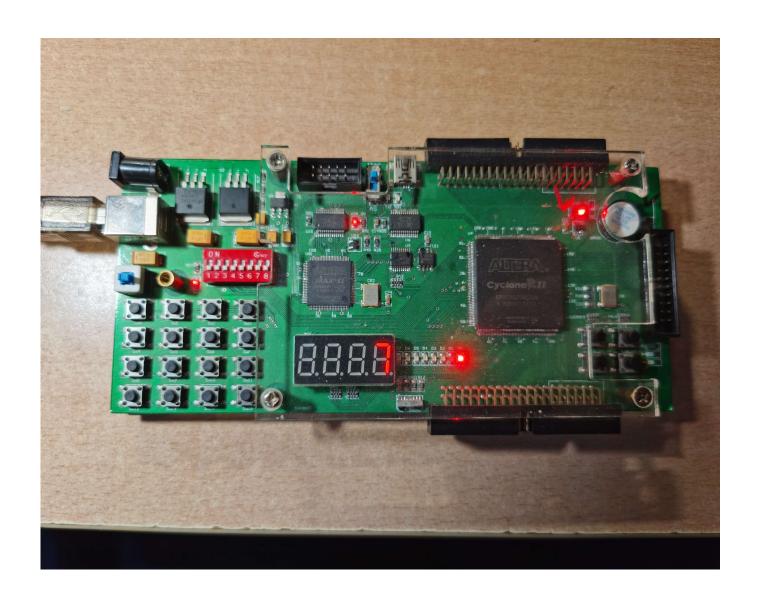
EDA03实验报告



自动化系

李昭阳 2021013445

(1) 分频器Clock

分频器模块的作用是将FPGA板的晶振产生的 50MHZ 的高频时钟信号降成 10000HZ 和 250HZ 的适合充电仪工作的较低频时钟信号,代码如下。

```
module Clock(input wire BoardCLK, output reg clk 250Hz, output reg clk 10000Hz);
initial clk_250Hz=1'b0;
initial clk_10000Hz=1'b0;
reg [17:0] i=18'b0;
reg [17:0] j=18'b0;
always @(posedge BoardCLK)
begin
   if(i==100000)
      begin
         clk 250Hz <= ~clk 250Hz;
         i<=18'b0;
   else
      i<=i+1:
end
always @(posedge BoardCLK)
   if(j==2500)
      begin
         clk 10000Hz<=~clk 10000Hz;
         j<=18'b0;
      end
      j<=j+1;
endmodule
```

(2) 键盘扫描模块KeyBoard

该键盘扫描实现是,在每一个时钟信号下扫描键盘中的一列,如果这一列对应的行有某位为低电平,则将代表前一时钟键盘有按下的变量,last置为1。在接下来的时间内,如果该按键能够维持20次时钟脉冲,则认为该信号不是按键抖动的误触,实现了防抖。按键信息到输出数据的转换,是通过case语句进行实现的。大部分逻辑相似,故给出部分精简代码如下。

```
module KeyBoard (input wire clk, input [3:0] Row, output reg
initial Col=4'blll1;
initial NoShut=1'b0;
initial ReadFromKeyBoard=4'b0000;
reg last=1'b0;
reg [0:4] i=5'b0;
                                                                                          4'b0111:
                                                                                             begin
 always @(posedge clk)
begin
if(last==0)
                                                                                                           ReadFromKeyBoard<=4'dl;
       begin
                                                                                                       end
          if (Row==4'b1111)
              begin
                                                                                                       begin
                 case (Col)
                                                                                                           ReadFromKeyBoard<=4'd2;
                     4'b0111: Col<=4'b1011;
                                                                                                    end
4'b1101:
                     4'bl011: Col<=4'bl101;
4'bl101: Col<=4'bl110;
                                                                                                       begin
                     4'b1110: Col<=4'b0111;
                                                                                                           ReadFromKeyBoard<=4'd3;
                     default: Col<=4'b0111;
                                                                                                       end
                 endcase
                                                                                                    4'b1110:
              end
                                                                                                       begin
          else
                                                                                                           ReadFromKeyBoard<=4'd4;
              begin
                 last<=l'bl;
             end
                                                                                             end
                                                                                          4'b1011:
    else
                                                                                             begin
                                                                                                 case (Row)
           if (Row==4'blll1)
              begin
                                                                                                       begin
                 last<=1'b0;
                                                                                                           ReadFromKeyBoard<=4'd5;
                 i<=5'b0:
                 NoShut<=1'b0;
                                                                                                    4'b1011:
              end
          else
                                                                                                           ReadFromKeyBoard<=4'd6;
                                                                                                       end
                 case (Col)
                                                                                                    4'b1101:
                                                                                                       begin
                        begin
                                                                                                           ReadFromKeyBoard<=4'd7;
                            case (Row)
                                                                                                    end
4'blll0:
                               4'b0111:
                                  begin
                                                                                                       begin
                                      ReadFromKeyBoard<=4'd1;
                                                                                                           ReadFromKeyBoard<=4'd8;
```

(3) 核心处理模块Core

状态机共四个状态,分别为初始状态Initial,开始输入的状态Start,表示已经输入至少一个数后状态的GetNum,表示充电倒计时的Counting。在Initial状态下,当按下开始键后,进入Start状态;

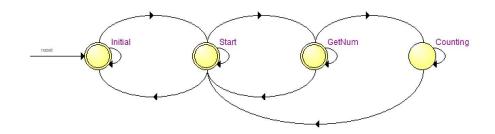
在Start状态下,如果十秒没有按下任何按键,则返回Initial状态;如果有表示数字的按键按下,则进入GetNum状态;

在GetNum状态下,如果按清零键,则返回Start状态;如果按下确认键,则进入Counting状态;

在Counting状态下,当倒计时结束时,播放20秒音乐,播放结束后返回Initial状态。代码如下。

```
module Core(input wire CLK, input NoShut, input [3:0] ReadFromKeyBoar
parameter Initial=2'b00;
parameter Satt=2'b01;
parameter GetNum=2'b10;
parameter GetNum=2'b10;
parameter Counting=2'b11;
reg [1:0]state=Initial;
reg preNoShur=1'b0;
assign availablee(preNoShut=1'b1&&NoShut=1'b0);
initial TimeLeft=19'b0;
initial Money=19'b0;
initial Play=1'b0;
initial Play=1'b0;
always @(posedge CLK)
begin
sf'...
                                                                                                                                                                                                                                                                       else
begin
if(TimeLeft==0)
hegin
                                                                                                                                                        _ase if(NoShu
begin
i<=i+1;
end
else
                                                                                                                                                                                                                                                                                               Mone,
if (j==20,
begin
j<
                                                                                                                                                                                                                                                                                                            j<=0;
state<=Start;
        if(state==Initial)
                                                                                                                                                                                                                                                                                                       Play<=0;
end
                                                                                                                                                        if(state==GetNum)
                  if(available&&ReadFromKeyBoard==11)
                                                                                                                                                         gin
if(available&&ReadFromKeyBoard==12)
                        begin
    state<=Start;
end</pre>
                                                                                                                                                                                                                                                                                                      begin
                                                                                                                                                                 egin
state<=Start;
Money<=0;
TimeLeft<=0;
                                                                                                                                                                                                                                                                                                           j<=j+1;
Play<=1;
                   light<=0;
TimeLeft<=0;
             Money<=0;
preNoShut<=NoShut;
end
                                                                                                                                                         end
else if(available&&ReadFromKeyBoard==13)
begin
                                                                                                                                                              state<=Counting;
end
       else if(state==Start)
begin
                                                                                                                                                         else
begin
if(availableiiReadFromKeyBoard>=0iIReadFromKeyBoard<=9)
begin
...
                  gin
light<=1;
TimeLeft<=0;
Money<=0;
if(i==100000)
begin
i<=0;
                                                                                                                                                                                                                                                                                                           j<=0;
TimeLeft=TimeLeft-1;
                                                                                                                                                                            gin
if (Money<20'd2)
begin
Money<=10'Money+ReadFromKeyBoard;
TimeLeft<=10'TimeLeft+2'ReadFromKeyBoard;
and
                                                                                                                                                                                                                                                                                                      begin
                                                                                                                                                                                                                                                                                                    j<=j+1;
end
                                                                                                                                                                            end
else if (Money<20'd10)
begin
                               state<=Initial;
                                                                                                                                                                                                                                                                                   preNoShut <= NoShut;
                                                                                                                                                                                                                                                                            end
                              state<=GetNum;
Money<=ReadFromKeyBoard;
                                                                                                                                                                                                                                                                  endmodule
```

状态机如下。



(4) 显示模块Show

该模块的作用是实时地将投币数和倒计时数显示在数码管上(Initial状态不显示)。通过高频选通信号扫描,利用视觉残留,使得数码管"假并真串"使数码管同时显现四个不同数字。代码如下。

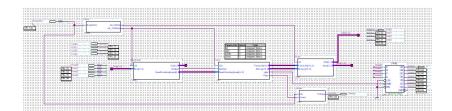
```
module Show(input wire clk, input [19:0] TimeLeft, input [19:0] Money, output reg [3:0] DIG, output reg [3:0] Num);
initial DIG=4'b0000;
initial Num=4'b0000;
always @(posedge clk)
begin
   case (DIG)
         begin
             DIG<=4'b0001;
             Num<=TimeLeft%10;
      4'b0001:
             DIG<=4'b0010;
             Num<=TimeLeft/10;
          end
      4'b0010:
begin
             DIG<=4'b0100:
             Num<=Money%10;
          end
         begin
DIG<=4'bl000;
             Num<=Money/10;
      default:
         begin
DIG<=4'b0001;
             Num<=0;
          end
   endcase;
end
endmodule
```

(5) 音乐播放模块Music

该模块的作用是播放一段20s的音乐,部分代码如下。

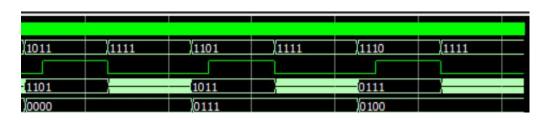
```
module Music(input wire clk, input music, output reg ToPlay);
initial ToPlay=1'b0;
localparam M1=95600;
                                                                              begin
localparam M2=85150;
                                                                                 if(cnt0<midtemp)
localparam M3=75850;
                                                                                    begin
localparam M5=63750;
                                                                                        ToPlay<=1'bl;
localparam M6=56800;
                                                                                    end
localparam M7=50600;
localparam H1=47750;
                                                                                    begin
reg [16:0] cnt0=0;
                                                                                        ToPlay<=1'b0;
reg [10:0] cntl=0;
                                                                                    end
reg [5:0] cnt2=0;
                                                                              end
reg [16:0] temp=0;
                                                                           else
wire [16:0] midtemp;
                                                                              begin
reg [10:0] num=0;
                                                                                 ToPlay<=1'b0;
wire [10:0] musicnum;
                                                                           if (cnt0==temp)
reg [10:0] nota=60;
assign midtemp=temp/2;
                                                                              begin
assign musicnum=num*7/8;
                                                                                 cnt0<=0;
always @(posedge clk)
                                                                                 if (cntl==num)
                                                                                    begin
begin
                                                                                        cnt1<=0;
   if (music==0)
                                                                                        if(cnt2==nota-1)
      begin
                                                                                           begin
         cnt0<=0;
                                                                                              cnt2<=0;
         cntl<=0;
                                                                                           end
         cnt2<=0;
                                                                                        else
         ToPlay<=1'b0;
                                                                                          begin
      end
                                                                                             cnt2<=cnt2+1;
   else
                                                                                           end
      begin
                                                                                    end
         case (cnt2)
                                                                                 else
            0:temp=M1;
                                                                                    begin
            1:temp=M2;
                                                                                        cntl<=cntl+1;
            2:temp=M3;
                                                                                    end
            3:temp=M5;
                                                                              end
            4:temp=M3;
                                                                           else
            5:temp=M2;
            6:temp=M1;
                                                                                 cnt0<=cnt0+1;
            7:temp=M1;
                                                                               end
            8:temp=M1;
                                                                        end
            9:temp=M2;
                                                                  end
            10:temp=M3;
                                                                  endmodule
```

(8) 整体设计图

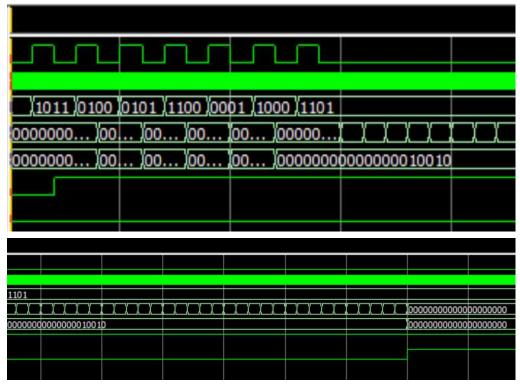


波形图与验证文件

键盘扫描模块仿真



核心模块仿真



图中为开始、长按4、长按5、清零、长按1、长按8、确认。当按下开始键 ("1011")后进入数字输入状态。长按数字键("0100"和"0101"),输出相应的金 额和时间;按下清零键("1100"),金额和时间被清零;再按下数字键("0001"和 "0100")重新输入金额和时间。仿真符合预期的需求。按下确认键("1101")后,进 入计时状态,充电时间逐渐减小,减为0时,播放音乐。音乐播放结束,变回输入状态,然 后经过10s,数码管熄灭,变为初始状态。仿真与预期相符,设计合理。

总结

通过本次实验,我对verilog语言的理解更加深刻,也逐渐脱离原理图进行纯语言的设计。我相信这次EDA3作业培养了我可编程逻辑器件的操作能力,同时在解决bug的过程中也锻炼了我的耐心和查阅资料的能力。