# 最新电子eda课程心得(五篇)

来源：网络 作者：独坐青楼 更新时间：2024-07-19

*人的记忆力会随着岁月的流逝而衰退，写作可以弥补记忆的不足，将曾经的人生经历和感悟记录下来，也便于保存一份美好的回忆。范文书写有哪些要求呢？我们怎样才能写好一篇范文呢？这里我整理了一些优秀的范文，希望对大家有所帮助，下面我们就来了解一下吧。电...*

人的记忆力会随着岁月的流逝而衰退，写作可以弥补记忆的不足，将曾经的人生经历和感悟记录下来，也便于保存一份美好的回忆。范文书写有哪些要求呢？我们怎样才能写好一篇范文呢？这里我整理了一些优秀的范文，希望对大家有所帮助，下面我们就来了解一下吧。

**电子eda课程心得篇一**

自动打铃系统设计说明书

学 生 姓 名：周文江

学

号：14112502521

专 业 班 级：1102

报告提交日期：2024.11.26

湖 南 理 工 学 院 物 电 学 院

目录

一、题目及要求简介……………3 1.设计题目…………………3 2.总体要求简介……………3

二、设计方案说明……………3

三、系统采用器件以及模块说明………3 1.系统框图…………4 2.选择的fpga芯片及配置………4 3.系统端口和模块说明…………5

四、各部分仿真结果………5

五、调试及总结………6

六、参考文献……7

七、附录………7

一、题目及要求简介

1、设计题目

设计一个多功能自动打铃系统

2、总体要求简介

① 基本计时和显示功能（24小时制显示），包括：

1.24小时制显示 2.动态扫描显示； 3.显示格式：88-88-88 ② 能设置当前时间(含时、分)③ 能实现基本打铃功能，规定：

06:00起床铃，打铃5s

二、设计方案说明

本次设计主要采用verilog hdl硬件描述性语言、分模块法设计的自动打铃系统。由于这次用的开发板提供的是50m晶振。首先要对时钟进行分频，当计时到2fa\_f07f时完成1s分频，通过计时到60s产生分钟进位信号，再通过60分钟产生时钟进位信号。最后通过6个寄存器对时分秒进行锁存最终输出到8个数码管上完成显示。当显示时钟和默认闹钟时钟相等时，驱动打铃模块。通过key\_mode,key\_turn,key\_change查看闹钟，时钟显示，调整时钟。

三、系统采用器件以及模块说明

1.系统框图如下：

：下如图框统系

2.选择的fpga芯片及配置：本次系统设计采用的fpga芯片是alter公司生产的cyclone ii ep2c8q208c8。该芯片是208个管脚，138个io，并且具有两个内部pll，而且内嵌乘法器，8k的逻辑门，资源相当丰富。完成这次自动打铃系统的设计总共消耗250个le单元，22个io口，131个寄存器。经过综合后，本系统最高能实现145m的运行速度。通过quartus ii 软件观察到内部的rtl图如下

3.系统端口和模块说明

(1)分频部分

分频器的作用是对50mhz的系统时钟信号进行分频，得到频率为1hz的信号，即为1s的计时信号。

(2)按键部分

按键key\_mode--0为显示计时，1为闹钟显示，2为调整时间。按键key\_turn—0为调整小时，1为调整分钟。按键key\_change—每按一次加1(3)计时部分

通过sec\_l,sec\_h,min\_l,min\_h,hour\_l,hour\_h 6个寄存器对时分秒进行锁存然后送入数码管显示

(4）闹钟模块

当设定的闹钟时间和数码管上显示的时间相等时驱动闹钟，完成打铃，持续时间5s。

（5）数码管显示模块

显示模块是由8个位选8个段选构成的显示模块，利用人眼的余晖效果完成动态扫描，显示时间。

四、各部分仿真结果

测试文件如下：

module clock\_tb;reg sysclk，rst\_b;reg key\_mode,key\_turn,key\_change;wire buzzer;

wire [7:0] led\_sel,led\_data;clock i\_clock(.sysclk(sysclk),.rst\_b(rst\_b),.key\_mode(key\_mode),.key\_change(key\_change),.key\_turn(key\_turn),.buzzer(buzzer),.led\_sel(led\_sel),.led\_data(led\_data));initial begin sysclk = 1\'b1;rst\_b = 1\'b0;//复位信号

#30 rst\_b = 1\'b1;end always #10 sysclk = ~sysclk;//输入的系统时钟，20ns的周期 endmodule

五、调试及总结

本次课程设计总共花费了四天左右的时间，设计了自动打铃系统。通过这次的设计更加熟悉了对eda技术的了解和认识，在中也发现许多不足的地方。使用了自顶而下的设计方法，使得设计更加的简单和明了。在调试过程中，有些代码的设计不规范性，导致时序相当缓慢，甚至编译综合都会报错。在不断的修改下，发现时序电路和组合逻辑最好分开写，这样便于查错，和修改代码。毕竟verilog hdl语言不同于c语言，不能以软件的思想来设计，而是要利用电路的思想来编程，这样可以更好的节省资源，使得时序也比较的简单明了。在以后的学习及程序设计当中，我们一定要倍加小心，在程序出现不正常运行的情况下要耐心调试，尽量做到精益求精。

最后通过这次eda方面的课程设计，提高了我们对eda领域及通信电路设计领域的认识，有利于培养我们在通信电路eda方面的设计能力。有利于锻炼我们独立分析问题和解决问题的能力。

六、文献参考

[1].王金明、左自强 编，《eda技术与verilog设计》科学出版社

2024.8 [2].杜慧敏、李宥谋、赵全良 编，《基于verilog的fpga设计基础》 西安电子科技大学出版社 2024.2 [3].韩彬 编，《从零开始走进fpga世界》杭州无线电爱好者协会出版社 2024.8.20

七、附录（实物图及源码）

module clock(//input

sysclk,rst\_b,key\_mode,key\_change,key\_turn，//output

buzzer,led\_sel,led\_data);

input sysclk,rst\_b;//sysclk--global system clock,rst\_b--global reset signal input key\_mode;//mode choose.0--timing function.1--alarm clock function.2--adjust function input key\_turn;//choose adjust minute or hour input key\_change;//count add 1 output buzzer;//device buzzer output [7:0] led\_sel;//led tube bit choose

output [7:0] led\_data;//led\_tube 8 bit data choose

parameter init\_hour = 8\'h12;parameter init\_min = 8\'h59;parameter init\_sec = 8\'h50;//initial time :12:59:50 parameter init\_alarm\_hour = 8\'h06;parameter init\_alarm\_min = 8\'h30;//initial alarm time : 06:30:0 parameter count\_1s = 28\'h2fa\_f07f;//count time 1s;

reg [7:0] sec;reg [7:0] min;reg [7:0] hour;reg [3:0] min\_l;//minute low 4 bit reg [3:0] min\_h;//minute high 4 bit reg [3:0] hour\_l;//hour low 4 bit reg [3:0] hour\_h;//hour high 4 bit reg [23:0] key\_time;//press key away shake reg key\_mode\_n;//press key\_mode next state reg key\_change\_n;//press key\_change next state reg key\_turn\_n;//press key\_turn next state wire key\_mode\_press;//sure button press key\_mode wire key\_turn\_press;//sure button press key\_turn wire key\_change\_press;//sure button press key\_change

always @(posedge sysclk)key\_mode\_n =“10”)then count1<=“00”;--count1为三进制加法计数器 else count1<=count1+1;end if;end if;end if;end process speaker;lamper:process(clk)begin if(rising\_edge(clk))then if(count<=“10”)then if(count=“00”)then lamp<=“001”;--elsif(count=“01”)then lamp<=“010”;elsif(count=“10”)then lamp<=“100”;end if;count<=count+1;else count<=“00”;end if;end if;end process lamper;end fun;

循环点亮三只灯

**电子eda课程心得篇五**

《eda课程设计》

课程设计题目：

基于单片机的温湿度采集系统

姓

名：

xxx

学

班

时

地

号：

xxxx

级：

xxxx

间：

2024.4.21~ 2024.5.5

点：

xxxxx

指 导

老

师：

xxxxx

目录

一、电路原理图..................................................................................2

二、电路pcb图（或实物图）.........................................................2

三、电路效果图..................................................................................3

四、设计总结......................................................................................3 附录（单片机源代码）......................................................................4

一、电路原理图

二、电路pcb图（或实物图）

三、电路效果图

四、设计总结

eda的实验还是挺有趣的，比较讲究动手能力，当然也不能忽略团体合作。总的来说本次实验还是成功了，虽然每个环节都遇到了困难。在生成原理图的过程中，就曾把导线画成了placeline而不是placewire，还有芯片的引脚应该用net符号而不是用文本符号，所以这些错误都导致我花在原理图上的时间多了点。而在生成pcb电路图的过程中遇到的困难则是自动布线之后，还有电源的几个脚需要手动布线，所以各个元件之间的位置要布置好，以免发生短路。腐蚀的时候，由于腐蚀的时间太长了，有些碳都化开了，导致里面的铜被腐蚀掉了，所以又为我的工作增加了困难。在焊接的时候，要注意元件的正负极，还要检测锡是否都与那些铜连接上了。最终把led和 dht11的程序烧进去就行了。

本次实验我还是能多多少少学到点什么的，总的来说还是希望能有多一点这样的实习。

附录（单片机源代码）

//51单片机控制温湿度传感器dht11

lcd1602上显示当前机最小系统。//lcd 读进去 写出来 #include

#include

typedef unsigned char byte;typedef unsigned int word;#define uint unsigned int

//定义无符号整型 #define uchar unsigned char typedef bit bool;

//此声明一个布尔型变量即真或假// uchar data\_byte,num,i;uchar rh,rl,th,tl,flag;uchar shuzi[4];unsigned char code num1[11]={0xc0,0xf9,0xa4,0xb0,0x99，0x92,0x82,0xf8,0x80,0x90,0x7f};

sbit dht=p2^4;

//dht11data端接单片机的p2^4口//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

延

时

函

数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* void delay(uchar ms)//延时模块//延时1毫秒

{

}

void delay1()

//一个for循环大概需要8个多机器周期

//一个机器周期为1us晶振为12mhz也就是说本函数延时8us{

} uchar i;

while(ms--)

for(i=0;i<110;i++);

uchar i;

for(i=0;i<1;i++);void display(void){ // if(flag==0)// {

p2=0x07;

p0=num1[shuzi[2]];delay(1);// }

// if(flag==1)// {

p2=0x0b;

p0=num1[shuzi[3]];delay(1);// } // if(flag==2)// {

p2=0x0d;

p0=num1[shuzi[0]];delay(1);// } // if(flag==3)// {

p2=0x0e;p0=num1[shuzi[1]];delay(1);// } }

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*dht11

测

试

某

块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*// void start()//开始信号

{

dht=1;

delay1();

//主机发出8us高电平，开始信号开始发出 dht=0;

delay(25);

// 主机把总线拉低必须大于18ms

dht11能检测到起始信号

dht=1;

//delay1();

//以下三个延时函数差不多为24usdelay1();delay1();

20-40us

}

uchar receive\_byte()

//接收一个字节 8位// {

uchar i,temp;

for(i=0;i<8;i++)//接收8bit的数据

{

while(!dht);

//等待40-50us的低电平开始信号结束

delay1();

//开始信号结束之后延时26us-28us

delay1();delay1();

temp=0;

//时间为26us-28usif(dht==1)

temp=1;

//如果26us-28us

\'0\'

数据为\'1\'

while(dht);

//

\'0\'为26us-28us

\'1\'为70us

} data\_byte<<=1;

//data\_byte|=temp;

//接收每一位的数据，相或保存数据

return data\_byte;}

void receive()//接收数据// {

uchar t\_h,t\_l,r\_h,r\_l,check,num\_check,i;start();

//开始信号//调用开始信号子函数

dht=1;

//主机设为输入判断从机dht11响应信号

if(!dht)

//判断从机是否有低电平响应信号// {

while(!dht);//判断从机发出 40us 的低电平响应信号是否结束//

while(dht);

//判断从机发出 40us 的高电平是否结束 如结束则从机进入发送数据状态，主机进入数据接收状态

数

//两个while语句加起来就是dht11的响应信号

r\_h=receive\_byte();//湿度高位

调用接受一个字节的子函

r\_l=receive\_byte();//湿度低位

t\_h=receive\_byte();//温度高位

t\_l=receive\_byte();//温度低位

check=receive\_byte();//校验位

//结束信号

dht=0;

//当最后一bit数据接完毕后主机拉低电平50us// for(i=0;i<7;i++)//差不多8us的延时

delay1();

dht=1;

//总线由上拉电阻拉高进入空闲状态

num\_check=r\_h+r\_l+t\_h+t\_l;

if(num\_check==check)//判断读到的四个数据之和是否与校验位相同

{

rh=r\_h;

rl=r\_l;

th=t\_h;

tl=t\_l;

check=num\_check;}

shuzi[0]=rh/10;shuzi[1]=rh%10;shuzi[2]=th/10;shuzi[3]=th%10;

} }

void main()//主函数模块// { while(1)

//进入死循环

{

receive();

//接收数据

display();

} }

本文档由站牛网zhann.net收集整理，更多优质范文文档请移步zhann.net站内查找