

|  |  |
| --- | --- |
| 成 绩 |  |
| 批阅教师 |  |
| 日 期 |  |

**桂林电子科技大学**

**实验报告**

**2020 -2021 学年第 二 学期**

开 课 单 位 桂林电子科技大学北海校区

适用年级、专业 18级专升本

课 程 名 称 FPGA技术综合设计实训

主 讲 教 师 余华

实 验 名 称 计时器和倒计时的系统

实 验 学 时 2学时

学 号

姓 名

**实验八 计时器和倒计时的系统**

# 一、实验目的

①掌握用Verilog HDL文本输入法设计计时电路的方法，并 通过电路仿真和硬件验证，进一步了解计时器的功能和特性。

②掌握用Verilog HDL文本输入法设计倒计时电路的方法，并通过电路仿真和硬件验证，进一步了解倒计时电路的功能和特性。

# 二、实验原理

计时器24小时计时器的电路框图如图8.1所示。

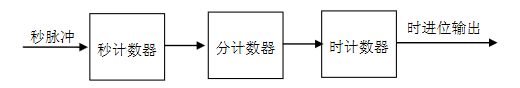


图8.1 24小时计时器的电路框图

桂林电子科技大学信息科技学院电子工程系《EDA技术及应用》实验指导书-11 -24小时计时器由2个60进制加计数器和1个24进制加计数器构成，输入CLK为1Hz（秒）的时钟，经过60进制加计数后产生1分钟的进位时钟信号，再经过60进制加计数后产生1小时的 进位时钟信号送给24进制加计数器进行加计数，当加计数到达23：59：59后，再来一个秒脉冲，产生时的进位输出。将两个60进制加 计数器和一个24进制加计数器的输出送数码管显示，得到计时器的显示结果。其中，秒脉冲由EDA实训仪上的20MHz晶振分频得到。

倒计时器24小时倒计时器的电路框图如图8.2所示。

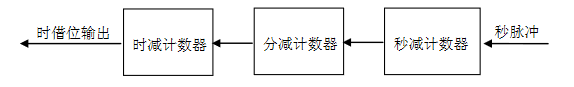


图8.2

24小时倒计时器由2个60进制减计数器和1个24进制减计数器构成，输入CLK为1Hz（秒）的时钟，经过60进制减计数后产生1分钟的借位时钟信号，再经过60进制减计数后产生1小时的 借位时钟信号送给24进制减计数器进行减计数，当减计数到达00：00：00后，产生时的借位输出，同时24小时倒 计时器停止倒计时，并发出提醒信号。将两个60进制减计数器和一个24进制减计数器的输出送数码管显示，得到倒计时的显示结果。其中，秒脉冲由EDA实训仪上的20MHz晶振分频得到。

# 三、实验设备

①DEDA实训仪1台。

②计算机1台(装有QuartusⅡ软件)。

# 四、实验内容

计时器在Quartus II软件中，按照实验原理中24小时计时器的电路框图，用Verilog HDL编程设计计时器电路，然后进行编辑、编译（综合）、仿真，引脚的锁定，并下载到EDA实训仪中进行验证。注：用EDA实训仪上的20MHz晶振作为计时器的时钟输入端，按键S8~S6分别作为计时器的校时、校分、校秒输入端，拨动开关S0作为计时器的清零输入端，拨动开关S1作为计时器的暂停输入端，用数码管SEG5~SEG0分别作为时、分、秒的输出端，用发光二极管L0作为进位输出端COUT

倒计时器在Quartus II软件中，按照实验原理中24小时倒计时器的电路框图，用Verilog HDL编分减计数器秒脉冲秒减计数器时借位输出图8.2 24小时倒计时器的电路框图时减计数器程设计倒计时器电路，然后进行编辑、编译（综合）、仿真，引脚的锁定，并下载到EDA实训仪中进行验证。注：用EDA实训仪上的20MHz晶振作为倒计时器的时钟输入端，按键S8~S6分别作为倒计时器的校时、校分、校秒输入端，拨动开关S0作为倒计时器的复位输入端，拨动开关S1作为倒计时器的暂停输入端，用数码管SEG5~SEG0分别作为时、分、秒的输出端，用发光二极管L0作为借位输出端COUT。

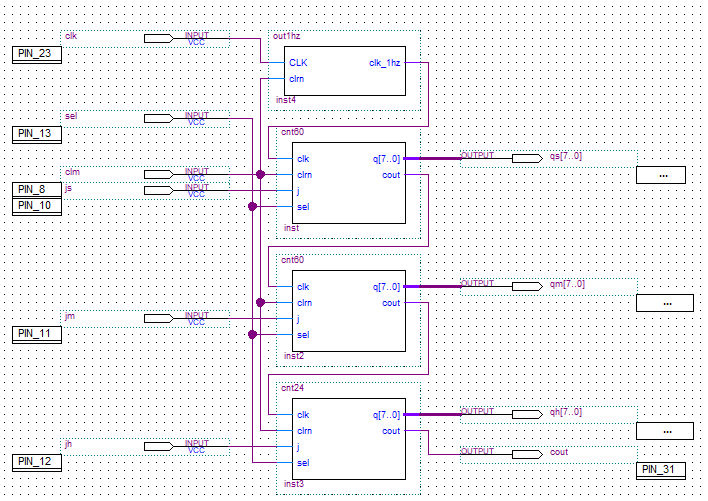
# 五、实验总结

通过本次课程设计的学习，我深深的体会到设计课的重要性和目的性所在。本次设计课不仅仅培养了我们实际操作能力，也培养了我们灵活运用课本知识，理论联系实际，独立自主的进行设计的能力。它不仅仅是一个学习新知识新方法的好机会，同时也是对我所学知识的一次综合的检验和复习，使我明白了自己的缺陷所在，从而查漏补缺。希望学校以后多安排一些类似的实践环节，让同学们学以致用。

在设计中要求我要有耐心和毅力，还要细心，稍有不慎，一个小小的错误就会导致结果的不正确，而对错误的检查要求我要有足够的耐心，通过这次设计和设计中遇到的问题，也积累了一定的经验，对以后从事工作会有一定的帮助。在应用VHDL的过程中让我真正领会到了其在电路设计上的优越性。用VHDL硬件描述语言的形式来进行数字系统的设计方便灵活，利用EDA软件进行编译优化仿真极大地减少了电路设计时间和可能发生的错误，降低了开发成本，这种设计方法必将在未来的数字系统设计中发挥越来越重要的作用。

# 六、仿真波形与实物图

使用模式5。



24小时模块

module cnt24(clk,clrn,j,sel,q,cout);

input clk,clrn,j,sel;

output reg[7:0] q;

output reg cout;

always @(posedge clk^j or

negedge clrn)

begin

if(!clrn) q=0;

else if(sel)

begin

if(q=='h23) q=0;

else q=q+1;

if(q[3:0]=='ha)

begin

q[3:0]=0;

q[7:4]=q[7:4]+1;

end

if(q=='h23) cout=1;

else cout=0;

end

else if(~sel)

begin

if(q==0) q='h23;

else q=q-1;

if(q[3:0]=='hf)

q[3:0]=9;

q[7:4]=q[7:4];

if(q==0) cout=cout-1;

else cout=0;

end

end

endmodule

时/分钟/秒

module cnt60(clk,clrn,j,sel,q,cout);

input clk,clrn,j,sel;

output reg[7:0] q;

output reg cout;

always @(posedge clk^j or negedge clrn)

begin

if(!clrn) q=0;

else if(sel)

begin

if(q=='h59) q=0;

else q=q+1;

if(q[3:0]=='ha)

begin

q[3:0]=0;

q[7:4]=q[7:4]+1;

end

if(q=='h59) cout=1;

else cout=0;

end

else if(~sel)

begin

if(q==0) q='h59;

else q=q-1;

if(q[3:0]=='hf)

q[3:0]=9;

q[7:4]=q[7:4];

if(q==0) cout=cout-1;

else cout=0;

end

end

endmodule

一秒分频

module out1hz(CLK,clrn,clk\_1hz);

input CLK,clrn;

reg [25:0]j;

output reg clk\_1hz;

always@(posedge CLK or negedge clrn)

begin

if(!clrn) begin clk\_1hz<=0;

j<=0;

end

else

begin if(j>=9\_999\_999)

begin j<=0;

clk\_1hz<=~clk\_1hz;

end

else j<=j+1;

end

end

endmodule

你的实物图