**《电子线路设计、测试与实验》实验报告**

|  |  |
| --- | --- |
| 实验一名称： | 多功能数字钟 |
| 院（系）： | 电子信息与通信工程学院 |
| 专业班级： | 电磁1802 |
| 姓名： | 吴叶赛 |
| 学号：  时间： | U201813405  2020年9月16日 |

**一、实验目的**

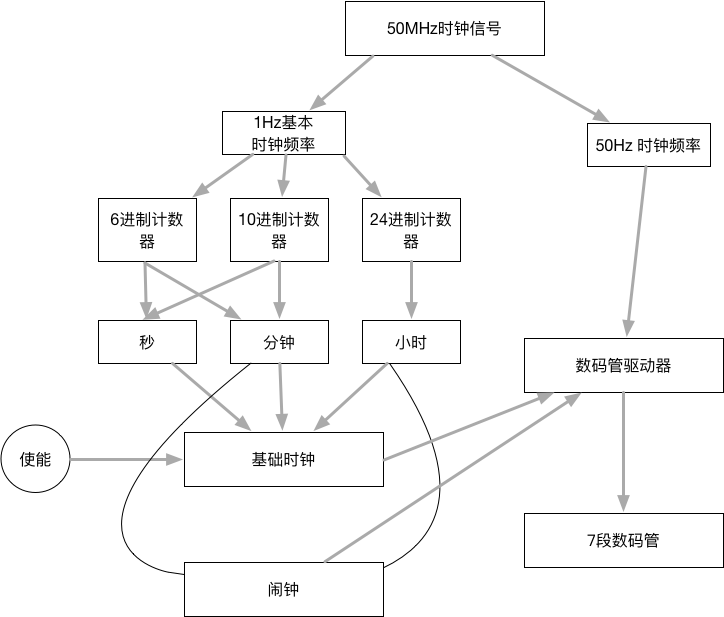
1. 熟悉verilog分层设计；
2. 熟悉功能模块的设计与重复调用；
3. 掌握数字电子钟的实现原理；
4. 实现闹钟，整点报时和时制切换。

**二、实验条件**

|  |  |
| --- | --- |
| 内容 | 型号 |
| 开发环境 | Vivado2018.1 |
| 开发语言 | Verilog HDL |
| 开发板 | NEXYS4 DDR |

**三、实验原理**

1）顶层模块



2）七段译码器

七段译码器用于将时间信号显示在数码管上。当输入D3D2D1D0接四位二进制数码十，输出低电平有效，用以驱动共阳极显示器。

3）动态扫描

动态数码扫描显示方式是利用了人眼的视觉暂留效应，把八个数码管按一定顺序（从左至右或从右至左）进行点亮，当频率足够大时，我们看到的将是全部同时显示（点亮），与传统方式得到的视觉效果完全一样。事实上，因为数码管点亮不是瞬间就可以的，它也需要一定的时间，该时间与数码管的选择有关系。为了折中这一对矛盾，实验中一般可将计数频率选择在100Hz左右可以满足上述两个要求。

4）分频器

该模块的任务是对100MHz的时钟信号进行分频，产生1Hz的秒脉冲信号，其占空比为50%，作为数字钟的计时基准。

设计一个模数为25×106的二进制递增计数器，其计数范围为0～24999999，每当计数器记到最大值时，输出信号翻转一次，即可产生1Hz的秒脉冲，其占空比为50%。

5）计数器设计

计数器分为三种计数器：模24计数器，模10计数器，模6计数器。

1. 模24计数器将八位寄存器分为两个四位二进制寄存器，分别存储时的个位和十位。个位范围从0000到1001，1001的下一位清零，同时十位进1，当十位为2且个位为3时，下一步个位十位同时清零。
2. 模10计数器使用一个四位二进制寄存器，用于存储分和秒的个位。计数范围从0000到1001，1001的下一位清零，同时控制模6计数器使能端使其进1。
3. 模6计数器使用一个四位二进制寄存器，用于存储分和秒的十位。计数范围从0000到0101，0101的下一位清零，同时控制模24计数器进1。

**四、实验内容**

1）拟定数字钟电路的组成框图，采用分层次、分模块的方法设计电路。

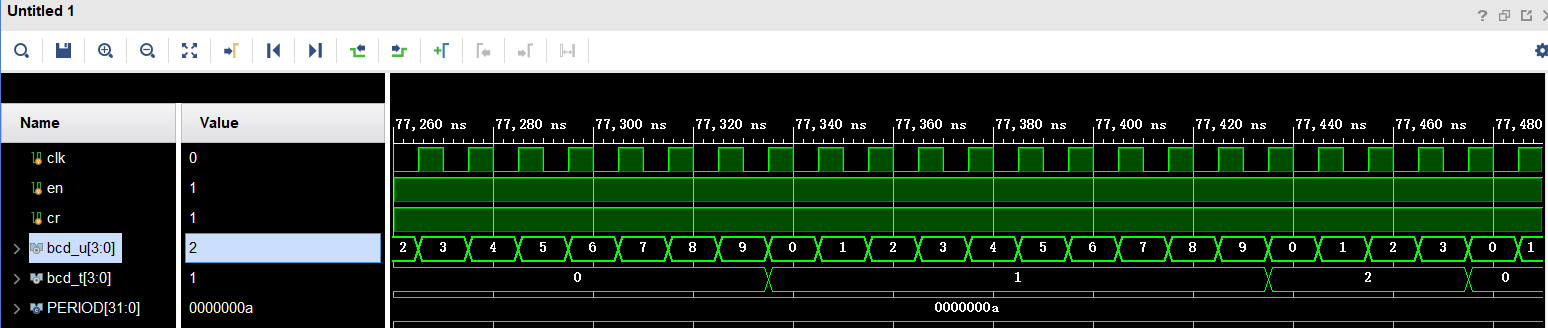
2）使用Verilog HDL设计各单元电路并进行仿真分析。

3）用FPGA开发板实现数字钟，并实际测试数字钟系统的逻辑功能。

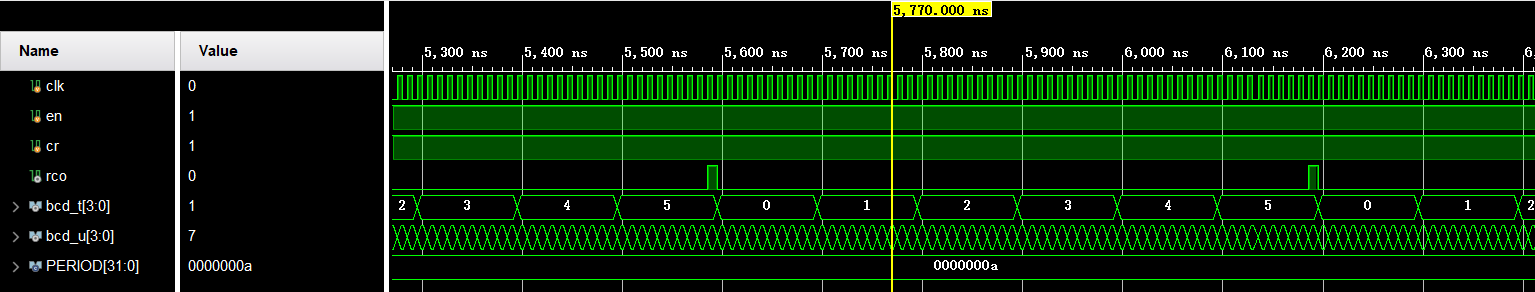
4）根据实验要求，实现闹钟，整点报时和时制切换三个拓展功能。

**五、仿真结果**

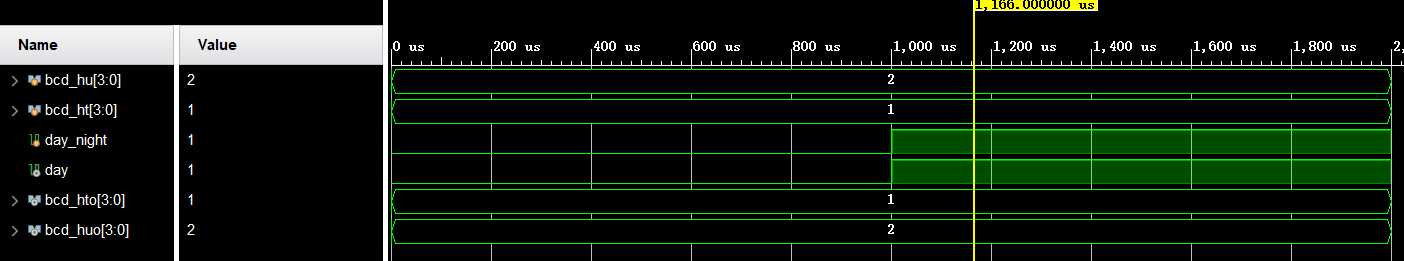
## 24小时波形图



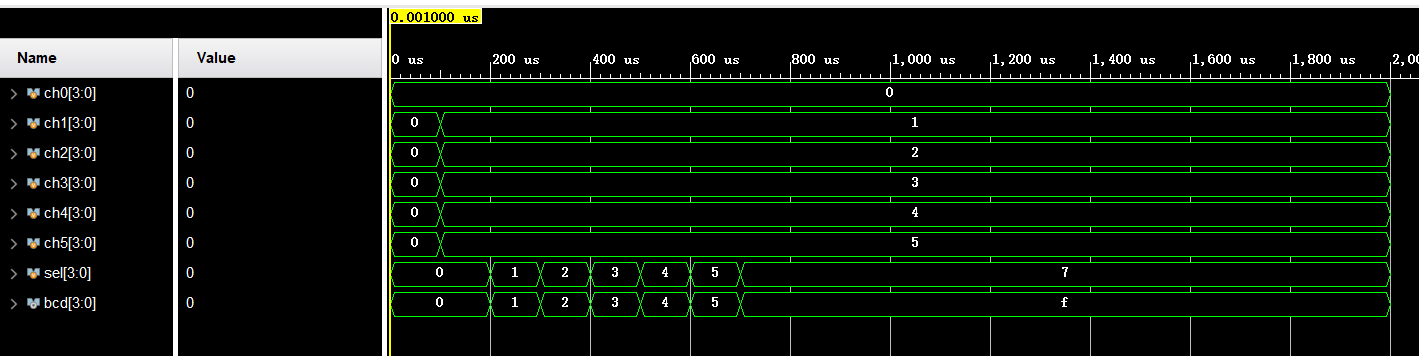
## 1分钟波形图



## 12进制／24进制仿真



## 6选1复用器仿真



**五、仿真激励代码**

1.//仿真激励代码

module top\_clocksim(

);

reg clk,reset,en,Adj\_Min,Adj\_Hour,timemode;

wire led,beep;

wire [7:0] HEX,AN;

top\_clock K0(clk,reset,en,led,Adj\_Min,Adj\_Hour,timemode,beep);

parameter PERIOD =10;

always begin

clk=1'b0;

#(PERIOD/2) clk=1'b1;

#(PERIOD/2);

end

initial

begin

reset=0;

en=0;

Adj\_Min=0;

Adj\_Hour=0;

timemode=0;

#20

reset=1;

en=1;

Adj\_Min=0;

Adj\_Hour=0;

timemode=0;

#200

reset=1;

en=1;

Adj\_Min=1;

Adj\_Hour=1;

timemode=0;

#230

reset=1;

en=1;

Adj\_Min=1;

Adj\_Hour=0;

timemode=0;

#360

reset=1;

en=1;

Adj\_Min=0;

Adj\_Hour=0;

timemode=0;

#600

reset=0;

en=0;

Adj\_Min=0;

Adj\_Hour=0;

timemode=0;

#20

reset=1;

en=1;

Adj\_Min=1;

Adj\_Hour=1;

timemode=0;

#80

reset=1;

en=1;

Adj\_Min=1;

Adj\_Hour=0;

timemode=0;

#500;

reset=1;

en=1;

Adj\_Min=0;

Adj\_Hour=0;

timemode=0;

end

endmodule

**六、模块代码**

1.//顶层文件

module top(

input clk\_100m,

input cr,

input en\_clock,

input clock\_adjust,

input clock\_alarm\_en,

input clock\_alarm\_set,

input min\_hour\_set,

input alarm\_adjust\_disp,

input day\_night,

output [7:0] pos,

output [6:0] seg,

output alarm,

output day

);

wire clk\_1hz,clk\_1k,clk\_5h,clock\_alarm,radio\_alarm,mux\_sel\_set,day\_radio;

wire [3:0] bcd\_day\_hu,bcd\_day\_ht,bcd\_day\_hu\_radio,bcd\_day\_ht\_radio;

wire [3:0] bcd\_smu,bcd\_smt,bcd\_shu,bcd\_sht,bcd\_mu,bcd\_mt,bcd\_su,bcd\_st;

wire [3:0] bcd\_hu,bcd\_ht,bcd\_temp\_mu,bcd\_temp\_mt,bcd\_temp\_hu,bcd\_temp\_ht;

assign alarm=radio\_alarm|clock\_alarm&clock\_alarm\_en;//整点报时或闹钟

assign mux\_sel\_set=clock\_alarm\_set|alarm\_adjust\_disp;//设置闹钟或显示闹钟

assign {bcd\_temp\_mu,bcd\_temp\_mt,bcd\_temp\_hu,bcd\_temp\_ht}=

mux\_sel\_set?{bcd\_smu,bcd

\_ smt,bcd\_shu,bcd\_sht}:{bcd\_mu,bcd\_mt,bcd\_hu,bcd\_ht};//显示2选1

clk\_div u\_clock(clk\_100m,clk\_1k,clk\_5h,clk\_1hz,cr);//分频

clock\_adjust u\_clk\_adj

(clk\_1hz,clock\_adjust,cr,min\_hour\_set,bcd\_su,bcd\_st,bcd\_mu,bcd\_mt,bcd\_hu,bcd\_ht);// 计时，校时

clock\_alarm u\_clk\_alarm(clk\_1hz,clk\_1k,clk\_5h,clock\_alarm\_en,clock\_alarm\_set,cr,min\_hour\_set,bcd\_mu,bcd\_mt,

bcd\_hu,bcd\_ht,bcd\_smu,bcd\_smt,bcd\_shu,bcd\_sht,clock\_alarm);

day24\_12 u\_day24\_12(bcd\_temp\_ht,bcd\_temp\_hu,day\_night,day,bcd\_day\_ht,bcd\_day\_hu);//显示24/12小时调整

day24\_12 radio\_day24\_12(bcd\_ht,bcd\_hu,day\_night,day\_radio,bcd\_day\_ht\_radio,bcd\_day\_hu\_radio);

radio u\_radio(bcd\_su,bcd\_st,bcd\_mu,bcd\_mt,bcd\_day\_ht\_radio,bcd\_day\_hu\_radio,clk\_1k,clk\_5h,clk\_1hz,en\_clock,cr,day\_night,radio\_alarm);

scan\_disp u\_scan\_disp(clk\_1k,cr,en\_clock,bcd\_day\_ht,bcd\_day\_hu,bcd\_temp\_mt,bcd\_temp\_mu,bcd\_st,bcd\_su,seg,pos);

endmodule

//时钟分频模块

module clk\_div(

input clk\_100m,

output clk\_1k,

output clk\_5h,

output clk\_1hz,

input cr

);

reg [15:0]count1k;

reg [8:0]count1hz;

reg clk\_1kr,clk\_5hr,clk\_1hzr;

assign clk\_1k=clk\_1kr;

assign clk\_5h=clk\_5hr;

assign clk\_1hz=clk\_1hzr;

always@(posedge clk\_100m or negedge cr)

if(!cr)

begin

count1k<=16'h0000;

clk\_1kr<=1'b0;

end

else begin

if(count1k==16'd49999)//100Mhz;100000分频得到1khz信号；

//if (count1k==4)

begin

count1k<=0;

clk\_1kr<= ~clk\_1kr;

end

else count1k<=count1k+1'b1;

end

always@(posedge clk\_1k or negedge cr)

if(!cr)

begin

count1hz<=9'h000;

clk\_1hzr<=1'b0;

end

else begin

if(count1hz==9'd499)//1000hz1000分频得到1hz

//if (count1hz==5)

begin

count1hz<=0;

clk\_1hzr<= ~clk\_1hzr;

end

else count1hz<=count1hz+1'b1;

end

always@(posedge clk\_1k or negedge cr)

if(!cr)

begin

clk\_5hr<=1'b0;

end

else

clk\_5hr<= ~clk\_5hr;//1khz2分频得到500hz信号

endmodule

/计时，校时模块

module clock\_adjust(

input clk,

input adjust,

input cr,

input min\_hour,

output [3:0] bcd\_su,

output [3:0] bcd\_st,

output [3:0] bcd\_mu,

output [3:0] bcd\_mt,

output [3:0] bcd\_hu,

output [3:0] bcd\_ht

);

wire m\_en,h\_en,rco\_s,rco\_m;

assign m\_en = rco\_s|(adjust&min\_hour); //分钟计时使能

assign h\_en = (rco\_s&rco\_m)|(adjust&~min\_hour); //小时计时使能

c60 second(clk,1'b1,cr,rco\_s,bcd\_st,bcd\_su);

c60 minute(clk,m\_en,cr,rco\_m,bcd\_mt,bcd\_mu);

c24 hour(clk,h\_en,cr,bcd\_hu,bcd\_ht);

endmodule

//闹钟模块

module clock\_alarm(

input clk\_1hz,

input clk\_1k,

input clk\_5h,

input en,

input set,

input cr,

input min\_hour,

input [3:0] bcd\_tmu,

input [3:0] bcd\_tmt,

input [3:0] bcd\_thu,

input [3:0] bcd\_tht,

output [3:0] bcd\_smu,

output [3:0] bcd\_smt,

output [3:0] bcd\_shu,

output [3:0] bcd\_sht,

output [0:0] alarm

);

wire equ;

assign equ=(bcd\_tmu==bcd\_smu)&(bcd\_tmt==bcd\_smt)&(bcd\_shu==bcd\_thu)&(bcd\_tht==bcd\_sht);//比较器

assign alarm=(equ&clk\_1k&clk\_1hz)|(equ&clk\_5h&~clk\_1hz);//1khz,500hz轮换输出闹铃

set\_time u\_set\_time(clk\_1hz,en,set,min\_hour,bcd\_smu,bcd\_smt,bcd\_shu,bcd\_sht,cr);//设置时分闹铃时间

endmodule

//24/12小时切换

module day24\_12(

input [3:0] bcd\_ht,

input [3:0] bcd\_hu,

input day\_night,

input day,

output [3:0] bcd\_hto,

output [3:0] bcd\_huo

);

wire modify,borrowu,mid\_night\_zero,noon;

wire [3:0] bcd\_temp\_hu,bcd\_temp\_ht,bcd\_temp\_huo,bcd\_temp\_hto;

assign noon=(bcd\_ht==1)&&(bcd\_hu==2);//中午12点

assign modify =(bcd\_ht>1)||(bcd\_hu>2)&&(bcd\_ht==1);//12点以后

assign borrowu=(bcd\_hu<2)?1:0;//BCD减法调整

assign bcd\_temp\_hu=borrowu?(bcd\_hu+4'b1010-4'b0010):(bcd\_hu-4'b0010);

assign bcd\_temp\_ht=borrowu?(bcd\_ht-4'b0010):(bcd\_ht-4'b0001);

assign mid\_night\_zero=(bcd\_ht==0)&&(bcd\_hu==0);//零点

assign bcd\_temp\_huo=(modify&day\_night)?bcd\_temp\_hu:bcd\_hu;

assign bcd\_temp\_hto=(modify&day\_night)?bcd\_temp\_ht:bcd\_ht;

assign {bcd\_hto,bcd\_huo}=(mid\_night\_zero&day\_night)?8'h12:{bcd\_temp\_hto,bcd\_temp\_huo};//午夜十二点

assign day=(modify|noon)&day\_night;//12点以后标志下午

endmodule

//仿电台报时

module radio(

input [3:0] bcd\_su,

input [3:0] bcd\_st,

input [3:0] bcd\_mu,

input [3:0] bcd\_mt,

input [3:0] bcd\_hu,

input [3:0] bcd\_ht,

input clk\_1k,

input clk\_5h,

input clk\_1hz,

input en,

input cr,

input day\_night,

output radio\_alarm

);

wire ld,min\_equ;

wire equ,grt,less,noon;

wire [3:0]bcd\_temp\_hu,bcd\_temp\_ht;

assign noon=({bcd\_ht,bcd\_hu}==8'h12);//中午12点

assign {bcd\_temp\_ht,bcd\_temp\_hu}=(noon&day\_night)?8'h00:{bcd\_ht,bcd\_hu};//中午12点且12点小时制则是下午0点

assign min\_equ=((bcd\_mt==5)&&(bcd\_mu==9))?1'b1:1'b0;//整点来临前

assign equ=({bcd\_st,bcd\_su,bcd\_mt,bcd\_mt}==16'h0000)?1'b1:1'b0;//整点

assign grt=(({bcd\_st,bcd\_su}+{bcd\_temp\_ht,bcd\_temp\_hu}+8'h1)>8'h5a)?1'b1:1'b0;//提示整点来临前的秒数

assign radio\_alarm=(((grt&min\_equ&clk\_5h)|(equ&clk\_1k))&clk\_1hz)&en;//整点来临前输出500HZ信号，整点时输出1kHZ信号

endmodule

module scan\_disp(

input clk,

input cr,

input en,

input [3:0] ch0,

input [3:0] ch1,

input [3:0] ch2,

input [3:0] ch3,

input [3:0] ch4,

input [3:0] ch5,

output [6:0] seg,

output [7:0] pos

);

wire rco;

wire[3:0] bcdsel,bcd\_data;

c6 u\_count(clk,cr,rco,en,bcdsel);//6进制同步计数器

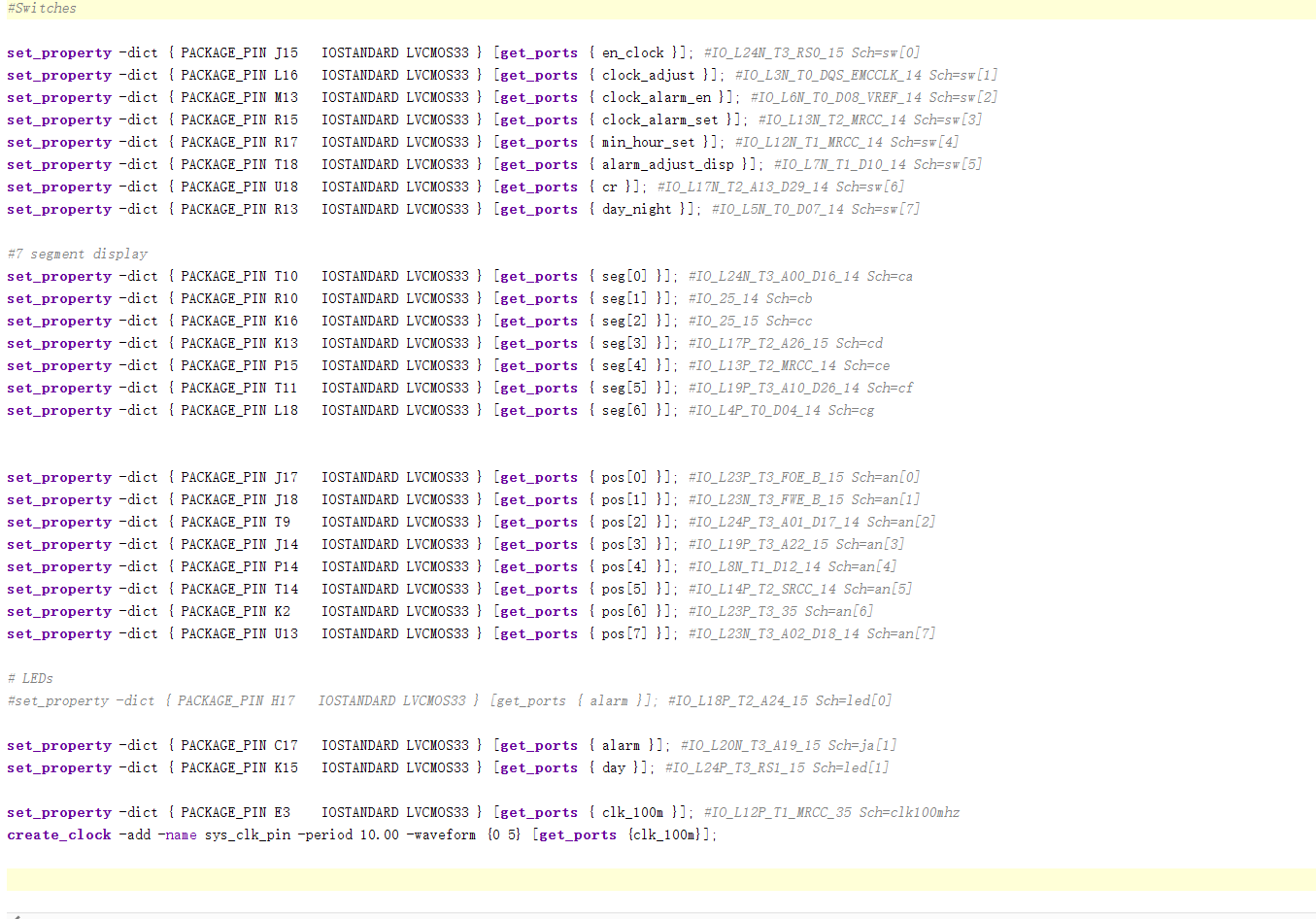
mux6\_1 u\_mux(ch0,ch1,ch2,ch3,ch4,ch5,bcdsel,bcd\_data);//6路四位复用器；

posdecode u\_pos(bcdsel,pos);//位码

bcd\_7seg u\_seg(bcd\_data,seg);//段码

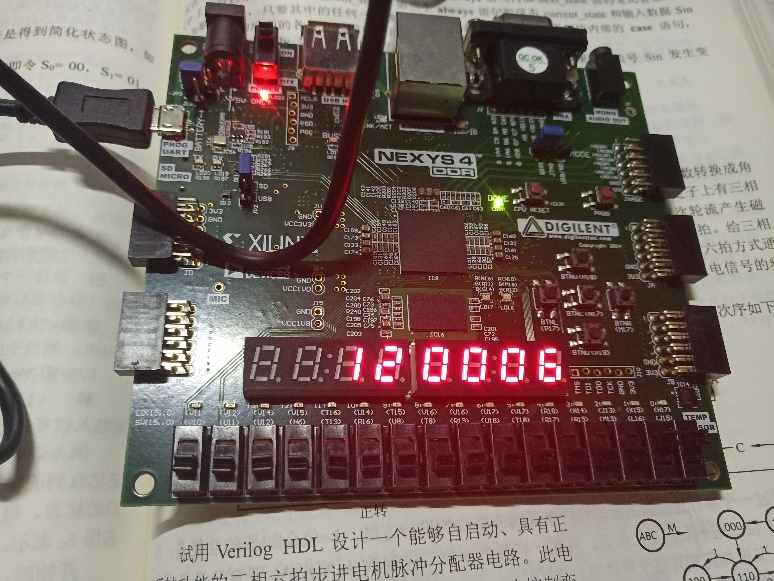
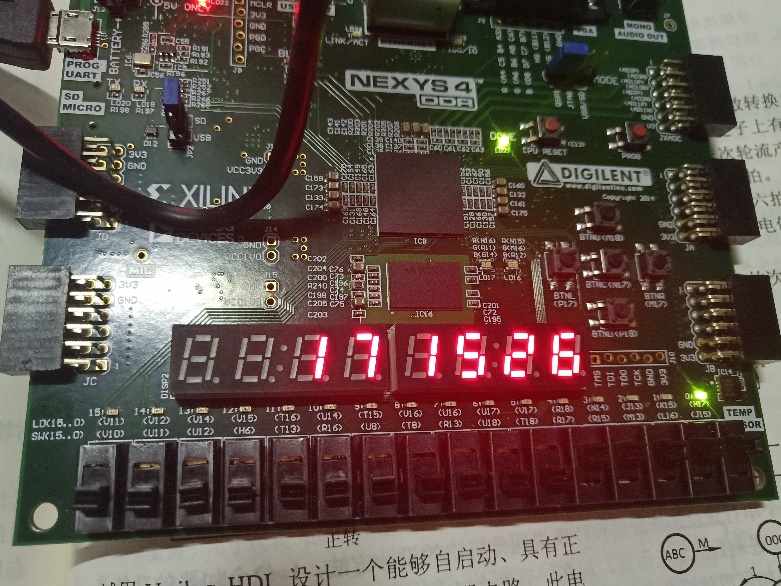
endmodule

**七、引脚约束**



**八、实验结果及分析**

将代码装载到开发板上进行实验，发现功能正确



第一个按钮为复位功能，第二个按钮为使能信号，第三个按钮为12/24小时切换，第四个按钮为分校准，第五个按钮为时校准

第一个LED灯为上下午切换，第二个LED灯为整点报时。

**九、小结**

本次试验综合性较强，参考的是mooc上的代码，自顶向下构建每个模块，最后实现了所有功能。可见，在电路设计中，一定要有目的性，要有很明细的架构设计。

而且有时候一步步照着慕课上做，最后也会有各种各样的问题出现，我们要自己多思考，真正理解了这个实验的原理和过程之后，做起来就又会有不一样的感觉，也会对EDA设计有不一样的理解。