# 

# 电

# 子

# 线

# 路

# 实

# 验

# 报

# 告

# 学院：电子信息与通信学院

# 班级：电信2005班

# 姓名：张智博

# 学号：U202011950

# 实验时间：2022年3月30日

目录

[一、 实验名称 1](#_Toc100238706)

[二、 实验目的 1](#_Toc100238707)

[三、 实验任务 1](#_Toc100238708)

[1. 功能要求 1](#_Toc100238709)

[2. 验收内容 1](#_Toc100238710)

[四、 实验原理 1](#_Toc100238711)

[1. 状态图的描述方法 1](#_Toc100238712)

[2. 脉冲分配器设计步骤 1](#_Toc100238713)

[3. 自顶而下的设计方法 2](#_Toc100238714)

[五、 实验过程 2](#_Toc100238715)

[1. 步进电机代码 2](#_Toc100238716)

[2. 步进电机仿真输出结果 4](#_Toc100238717)

[3. 数字钟代码 4](#_Toc100238718)

[4. 数字钟的插板结果 14](#_Toc100238731)

[六、 实验小结 15](#_Toc100238732)

# 第七次实验：步进电机及数字钟

# 实验名称

步进电机及数字钟

# 实验目的

1. 掌握用verilog HDL描述数字逻辑电路与系统的方法;
2. 有限状态机概念;
3. 掌握用verilog HDL描述有限状态机的方法；
4. 掌握分层次电路设计方法;
5. 熟练掌握数字钟的设计与调试方法。

# 实验任务

步进电机脉冲分配器设计与多功能数字钟设计

## 功能要求

1.使用组合逻辑 能显示小时、分钟、秒钟

2.能调整小时、分钟的时间

## 验收内容

代码及实验板实操结果

# 实验原理

## 状态图的描述方法

1.利用parameter描述状态机中各个状态的名称，并指定状态编码。例如，对序列检测器的状态分配使用最简单的自然二进制码，其描述如下:parameter SO=2'b00,S1=2'b01,S2=2'b10,S3 = 2'b11;

2.用always块描述状态触发器实现状态存储。

3.使用敏感表和case语句(也可以采用if-else等价语句)描述的状态转换逻辑。

4.描述状态机的输出逻辑。

## 脉冲分配器设计步骤

1.创建子目录E:\EDA\_Lab\Lab5，并新建一个工程项目。

2.使用Verilog HDL设计电路，并进行仿真分析。

3.用FPGA开发板实现步进电机脉冲分配器，并实际测试逻辑功能。(A、B、C用发光二极管代替)。

4.根据实验流程和实验结果，写出实验总结报告，并对波形图和实验现象进行说明。

## 自顶而下的设计方法

先设计顶层总框图,该框图由若干个具有特定功能的源模块组成。下一步针对这些具有不同功能的模块进行设计,对于有些功能复杂的模块,还可以将该模块继续化分为若干个功能子模块，这样就形成模块套模块的层次化设计方法。

# 实验过程

## 步进电机代码

module bujin (Sin,CP,nCR,Out);

input Sin,CP,nCR;//声明输入

output Out;//声明输出

reg [2:0] Current\_state,Next\_state,Out;

parameter S0=3'b000,S1=3'b001,S2=3'b010,S3=3'b011,

S4=3'b100,S5=3'b101,S6=3'b110,S7=3'b111;

always@(posedge CP or negedge nCR)//状态转换

begin

if(~nCR)

Current\_state<=S7;//异步清零

else

Current\_state<=Next\_state;//触发器翻转

end

always@(Current\_state)

begin

case(Current\_state)

S0:begin

Out=Current\_state;

Next\_state=S6;

end

S1:begin

Out=Current\_state;

Next\_state=(Sin==1)?S3:S5;

end

S2:begin

Out=Current\_state;

Next\_state=(Sin==1)?S6:S3;

end

S3:begin

Out=Current\_state;

Next\_state=(Sin==1)?S2:S1;

end

S4:begin

Out=Current\_state;

Next\_state=(Sin==1)?S5:S6;

end

S5:begin

Out=Current\_state;

Next\_state=(Sin==1)?S1:S4;

end

S6:begin

Out=Current\_state;

Next\_state=(Sin==1)?S4:S2;

end

S7:begin

Out=Current\_state;

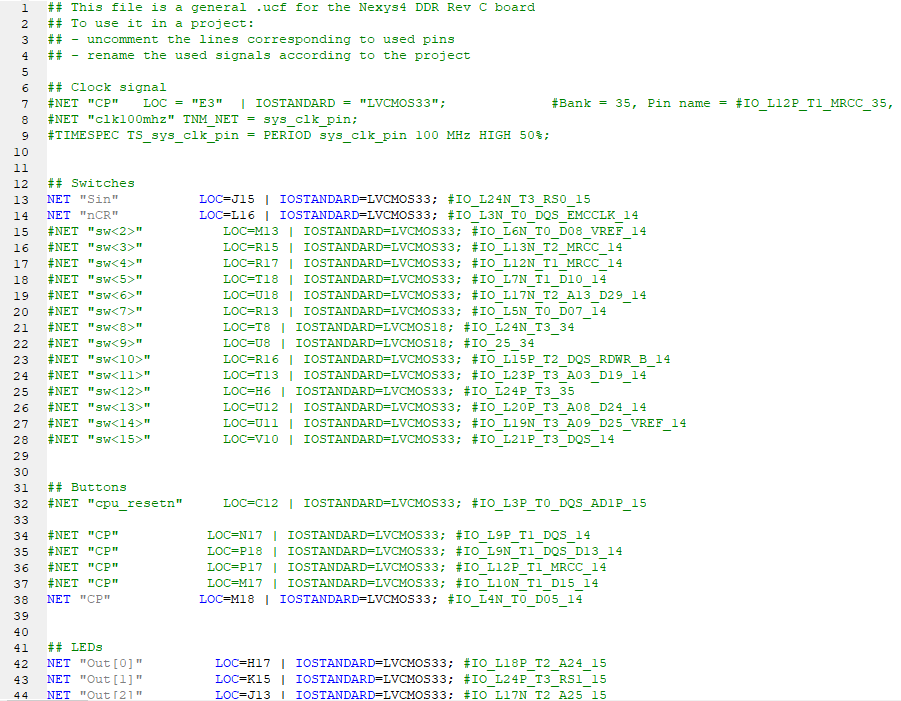
Next\_state=S0;

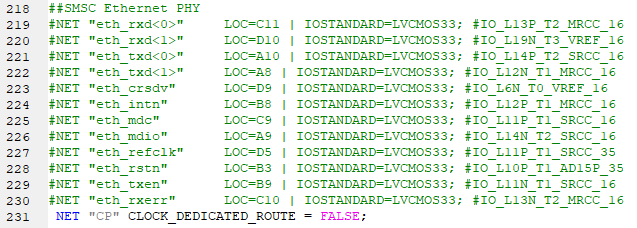
end

endcase

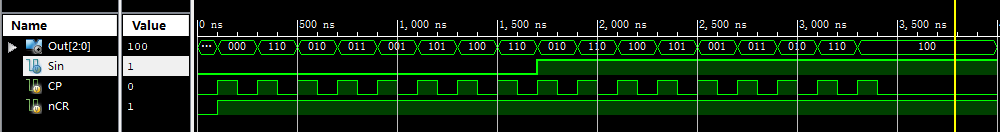
end

Endmodule





## 步进电机仿真输出结果



## 数字钟代码

### 1.顶层文件

module clock(

input CP,CR,EN,stop,setclock,dangwei,adjmode,mode,seta,showmode,set,setmode, //CR清零，EN熄屏，stop停止计时，setclock调整时钟开关，dangwei切换小时/时间调整，adjmode切换加减调整,mode切换显示时间、闹钟,seta调整闹钟开关,startalarm开启关闭闹钟,showmode12、24小时显示切换,set手动调整CP输入,setmode切换手动、自动调整

output [6:0] show,

output [7:0] guan,

output zhengLED

);//CP时钟，CR置零，EN停止显示，stop时钟停止，show显示的部分，guan显示的数码管

wire [7:0] miao,fen,hour;//存放秒、分、时

wire [3:0] number;//存放显示的数字

wire JIN1,JIN2;//JIN1秒进分，JIN2分进时(时钟模式下)

wire CP2,CP1,CP0;//CP1 1hz,CP0 1000hz,CP2 约3hz

wire psetclock,pseta; //防抖动后的信号

assign CPCLOCK=(~psetclock&&setmode)?set:CP1; //手动调整模式下，CPCLOCK和CPALARM由set开关产生

assign CPALARM=(~pseta&&setmode)?set:CP1;

assign JIN1=(miao[3:0]==4'b1001&&miao[7:4]==4'b0101)||((~setclock)&&mode); //产生时钟进分信号，当在时钟模式下设置时或到59秒

assign JIN2=(fen[3:0]==4'b1001&&fen[7:4]==4'b0101&&miao[3:0]==4'b1001&&miao[7:4]==4'b0101)||((~setclock)&&mode);//产生时钟进时信号，当在时钟模式下设置时或到59分59秒

hz1000 U0(CP,CP0); //产生1000hz

pinglv U1(CP0,CR,stop,CP1); //产生1hz

miao U2(CP1,CR,miao[7:4],miao[3:0]); //秒记录

fenzhong U3(CPCLOCK,CR,setclock,dangwei,adjmode,JIN1,fen[7:4],fen[3:0]); //分记录

hour U4(CPCLOCK,CR,setclock,dangwei,adjmode,JIN2,hour[7:4],hour[3:0]); //时记录

xianshi U5(number,show); //显示管控制

showtime U6(CP0,EN,mode,showmode,miao,fen,hour,alarmfen,alarmhour,number,guan); //显示控制

zheng U6(CP2,CR,setclock,showmode,fen,miao,hour,zhengLED); //整点报时模块

zhengCP U7(CP0,CP2); //产生整点报时的CP信号,约3HZ

prevent U8(CP0,0,0,setclock,psetclock);//防抖动

prevent U9(CP0,0,0,seta,pseta); //防抖动

endmodule

### 2.分频为1khz

module hz1000(

input CP,

output reg CLK=0

);

reg [15:0] state;

always@(posedge CP)

begin

if(state<49999) state<=state+1'b1;//由1MHZ脉冲产生1KHZ脉冲，每计数50000输出翻转一次

else

begin

CLK<=~CLK;

state<=0;

end

end

endmodule

### 3.分频为1hz

module pinglv(

input CP,CR,stop,

output reg CLK

);

reg [8:0] state;

always@(posedge CP,negedge CR)

begin

if(~CR)

begin

CLK<=0;

state<=0;

end

else if(~stop) //stop可以控制使得停止计时（便于调整秒）

begin

CLK<=CLK;

state<=state;

end

else if(state<499) state<=state+1'b1; //由1kHZ脉冲产生1HZ脉冲，每计数500输出翻转一次

else

begin

CLK<=~CLK;

state<=0;

end

end

endmodule

### 4.秒计数

module miao(

input CP,CR,

output reg [3:0] fen1,fen0

);

always @(posedge CP,negedge CR) //1Hz脉冲控制，秒分情况+1

begin

if(~CR)

begin

fen1<=0;

fen0<=0;

end

else if(fen1<6&&fen0<9) fen0<=fen0+1'b1;

else if(fen1<5&&fen0==9)

begin

fen0<=0;

fen1<=fen1+1'b1;

end

else

begin

fen1<=0;

fen0<=0;

end

end

endmodule

### 5.分计数

module fenzhong(

input CP,CR,setclock,dangwei,adjmode,JIN1,

output reg [3:0] fen1,fen0

);

always @(posedge CP,negedge CR)

begin

if(~CR) //清零

begin

fen1<=0;

fen0<=0;

end

else if(JIN1==0); //无进位信号保持

else if({setclock,dangwei,adjmode}==3'b000) //在设置分钟模式且调整模式为减时，分钟分情况-1

begin

if(fen0>0) fen0<=fen0-1'b1;

else if(fen1>0&&fen0==0)

begin

fen0<=4'b1001;

fen1<=fen1-1'b1;

end

else

begin

fen1<=4'b0101;

fen0<=4'b1001;

end

end

else if(setclock==0&&dangwei==1); //设置小时时，保持

else //分钟分情况+1

begin

if(fen1<6&&fen0<9) fen0<=fen0+1'b1;

else if(fen1<5&&fen0==9)

begin

fen0<=0;

fen1<=fen1+1'b1;

end

else

begin

fen1<=0;

fen0<=0;

end

end

end

endmodule

### 6.小时计数

module hour(

input CP,CR,setclock,dangwei,adjmode,JIN2,

output reg [3:0] hour1,hour0

);

always @(posedge CP,negedge CR)

begin

if(~CR) //清零

begin

hour1<=0;

hour0<=0;

end

else if(JIN2==0); //无进位信号，保持

else if({setclock,dangwei,adjmode}==3'b010)//在设置模式且调整模式为减时，小时分情况-1

begin

if(hour0>0) hour0<=hour0-1'b1;

else if(hour1>0&&hour0==0)

begin

hour0<=4'b1001;

hour1<=hour1-1'b1;

end

else

begin

hour1<=4'b0010;

hour0<=4'b0011;

end

end

else if(setclock==0&&dangwei==0); //设置分钟时，保持

else //小时分情况+1

begin

if(hour1<2&&hour0<9) hour0<=hour0+1'b1;

else if(hour1<2&&hour0==9)

begin

hour0<=0;

hour1<=hour1+1'b1;

end

else if(hour1==2&&hour0<3) hour0<=hour0+1'b1;

else

begin

hour1<=0;

hour0<=0;

end

end

end

endmodule

### 7.译码显示

module xianshi( //显示译码，将BCD码转为数码管的数字

input [3:0] Q,

output reg[6:0] show=0

);

always@(Q)

begin

case(Q)

4'h0:show<=7'b100\_0000;

4'h1:show<=7'b111\_1001;

4'h2:show<=7'b010\_0100;

4'h3:show<=7'b011\_0000;

4'h4:show<=7'b001\_1001;

4'h5:show<=7'b001\_0010;

4'h6:show<=7'b000\_0010;

4'h7:show<=7'b111\_1000;

4'h8:show<=7'b000\_0000;

4'h9:show<=7'b001\_0000;

4'ha:show<=7'b000\_1000;

4'hb:show<=7'b000\_1100;

default:show<=7'b111\_1111;

endcase

end

endmodule

### 8.通过扫掠显示数字

module showtime(

input CP0,EN,mode,showmode,

input [7:0]miao,fen,hour,alarmfen,alarmhour,

output reg[3:0]number=0,

output reg[7:0]guan=0

);

integer k;

always@(posedge CP0) //在1000Hz脉冲下，依次显示8个管子内容

begin

if(~EN) guan<=8'b1111\_1111; //熄屏

case(k)

0:begin

number<=miao[3:0];

guan<=8'b1111\_1011;

k<=k+1;

end

1:begin

number<=miao[7:4];

guan<=8'b1111\_0111;

k<=k+1;

end

2:begin

number<=fen[3:0];

guan<=8'b1110\_1111;

k<=k+1;

end

3:begin

number<=fen[7:4];

guan<=8'b1101\_1111;

k<=k+1;

end

4:begin

if(showmode==1) number<=hour[3:0];

else

begin

if((hour[7:4]==1&&hour[3:0]>2)||(hour[7:4]==2&&hour[3:0]>=2)) number<=hour[3:0]-4'b0010;

else if(hour[7:4]==2&&hour[3:0]<2) number<=hour[3:0]+4'b1000;

else if(hour[7:4]==0&&hour[3:0]==0) number<=4'b0010;

else number<=hour[3:0];

end

guan<=8'b1011\_1111;

k<=k+1;

end

5:begin

if(showmode==1) number<=hour[7:4];

else

begin

if((hour[7:4]==1&&hour[3:0]>2)||(hour[7:4]==2&&hour[3:0]>=2)) number<=hour[7:4]-4'b0001;

else if(hour[7:4]==2&&hour[3:0]<2) number<=0;

else if(hour[7:4]==0&&hour[3:0]==0) number<=4'b0001;

else number<=hour[7:4];

end

guan<=8'b0111\_1111;

k<=k+1;

end

6:begin

if(showmode==1) k<=0;

else

begin

guan<=8'b1111\_1110;

if(hour[7:4]==2||(hour[7:4]==1&&hour[3:0]>=2)) number<=4'b1011;

else number<=4'b1010;

k<=0;

end

end

default:k<=0;

endcase

end

endmodule

### 8.整点亮灯

module zheng(

input CP2,CR,setclock,showmode,

input [7:0] fen,miao,hour,

output reg zhengLED

);

integer k;

always@(posedge CP2,negedge CR)

begin

if(~CR||~setclock)

begin

k<=0;

zhengLED<=0;

end

else if(fen==0&&miao==0)

k<=2;

else if(k>0)

begin

zhengLED<=~zhengLED;

k<=k-1'b1;

end

else zhengLED<=0;

end

endmodule

### 9.整点亮灯的时钟产生

module zhengCP(

input CP,

output reg CLK=0

);

reg [7:0] state;

always@(posedge CP)

begin

if(state<165) state<=state+1'b1;//由1MHZ脉冲产生1KHZ脉冲，每计数166输出翻转一次

else

begin

CLK<=~CLK;

state<=0;

end

end

endmodule

### 10.rs触发器防抖动

module prevent( //RS触发器，防止抖动

input CP,R,S,CPCLOCK,

output reg pCPCLOCK=0

);

always@(posedge CP)

begin

case({R,S})

2'b00:pCPCLOCK<=CPCLOCK;

2'b01:pCPCLOCK<=1'b1;

2'b10:pCPCLOCK<=0;

2'b11:pCPCLOCK<=1'bx;

endcase

end

endmodule

### 约束条件

NET "guan[0]" LOC = J17;

NET "guan[1]" LOC = J18;

NET "guan[2]" LOC = T9;

NET "guan[3]" LOC = J14;

NET "guan[4]" LOC = P14;

NET "guan[5]" LOC = T14;

NET "show[0]" LOC = T10;

NET "show[1]" LOC = R10;

NET "show[2]" LOC = K16;

NET "show[3]" LOC = K13;

NET "show[4]" LOC = P15;

NET "show[5]" LOC = T11;

NET "show[6]" LOC = L18;

NET "CP" LOC = E3;

NET "CR" LOC = J15;

NET "EN" LOC = L16;

NET "stop" LOC = M13;

NET "setclock" LOC = R15;

NET "dangwei" LOC = R17;

NET "adjmode" LOC = T18;

NET "showmode" LOC=U18;

NET "mode" LOC = R13;

NET "zhengLED" LOC=J13;

NET "set" LOC=N17;

NET "setmode" LOC=V10;

NET "guan[5]" IOSTANDARD = LVCMOS18;

NET "guan[4]" IOSTANDARD = LVCMOS18;

NET "guan[3]" IOSTANDARD = LVCMOS18;

NET "guan[2]" IOSTANDARD = LVCMOS18;

NET "guan[1]" IOSTANDARD = LVCMOS18;

NET "guan[0]" IOSTANDARD = LVCMOS18;

NET "show[6]" IOSTANDARD = LVCMOS18;

NET "show[5]" IOSTANDARD = LVCMOS18;

NET "show[4]" IOSTANDARD = LVCMOS18;

NET "show[3]" IOSTANDARD = LVCMOS18;

NET "show[2]" IOSTANDARD = LVCMOS18;

NET "show[1]" IOSTANDARD = LVCMOS18;

NET "show[0]" IOSTANDARD = LVCMOS18;

NET "CP" IOSTANDARD = LVCMOS18;

NET "CR" IOSTANDARD = LVCMOS18;

NET "EN" IOSTANDARD = LVCMOS18;

NET "stop" IOSTANDARD = LVCMOS18;

NET "setclock" IOSTANDARD = LVCMOS18;

NET "dangwei" IOSTANDARD = LVCMOS18;

NET "adjmode" IOSTANDARD = LVCMOS18;

NET "mode" IOSTANDARD = LVCMOS18;

NET "showmode" IOSTANDARD = LVCMOS18;

NET "zhengLED" IOSTANDARD = LVCMOS18;

NET "set" IOSTANDARD = LVCMOS18;

NET "setmode" IOSTANDARD = LVCMOS18;

# PlanAhead Generated physical constraints

NET "guan[6]" LOC = K2;

NET "guan[7]" LOC = U13;

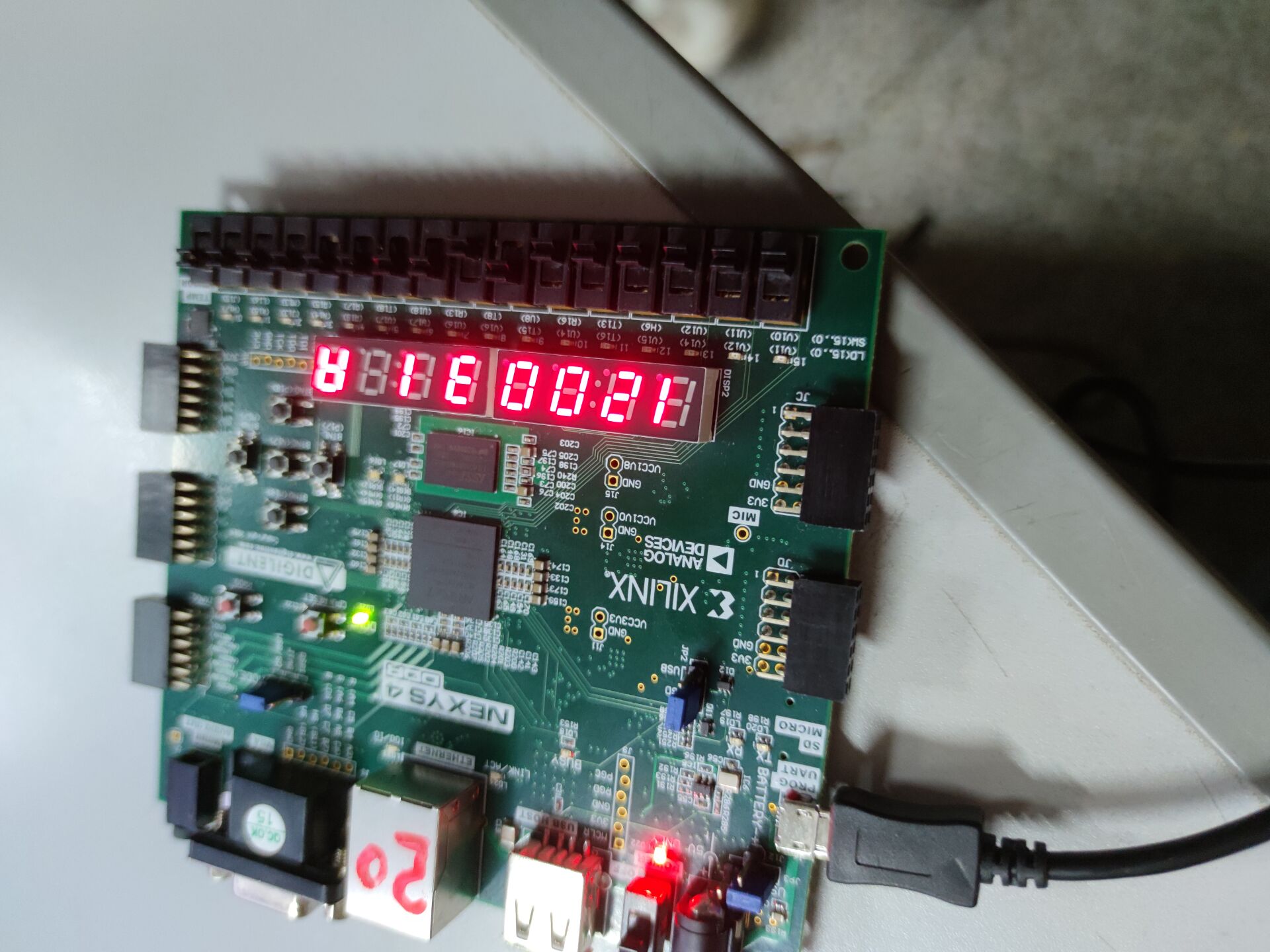
# PlanAhead Generated IO constraints

NET "guan[7]" IOSTANDARD = LVCMOS18;

NET "guan[6]" IOSTANDARD = LVCMOS18;

NET "set" CLOCK\_DEDICATED\_ROUTE = FALSE;

## 数字钟的插板结果



# 实验小结

通过本实验，我切身体验到了数字电路设计的难度，不过也正是通过实验，我深刻学习到了ise等软件的使用方法，体会到了数字电路软件设计方法，获得一些常见错误和重点的掌握。为学习后面的课程打下了坚实的基础。