数字逻辑与数字系统设计

实验大作业报告

（2020年）

课程名称：数字逻辑与数字系统设计

任课教师：张英涛

作业题目：显示倒计时，自定义密码电子密码锁

完成人：陈一帆 夏时雨

学号：1191000606 1181000705

班级：1903003 1803201

报告日期：2020年 12 月 16 日

|  |  |
| --- | --- |
| **报告成绩** |  |
| **教师评语** |  |

目录

[1 设计要求 2](#_Toc59115806)

[2 工作原理及系统方框图 2](#_Toc59115807)

[2.1 工作原理 2](#_Toc59115808)

[2.2 系统方框图 3](#_Toc59115809)

[3 各部分模块具体功能及设计思路 4](#_Toc59115810)

[3.1 Operate 4](#_Toc59115811)

[3.2 ClkControl 5](#_Toc59115812)

[3.3 Slide 5](#_Toc59115813)

[3.4 Modify 6](#_Toc59115814)

[3.5 Valuetodiode 7](#_Toc59115815)

[3.6 Lightdiode 7](#_Toc59115816)

[3.7 Checkorchange 8](#_Toc59115817)

[3.8 Countdown 9](#_Toc59115818)

[4 调试过程 10](#_Toc59115819)

[5 设计结论 10](#_Toc59115820)

[6 设计心得与总结 11](#_Toc59115821)

[6.1 陈一帆 11](#_Toc59115822)

[6.2 夏时雨 12](#_Toc59115823)

[7 参考文献 12](#_Toc59115824)

[8 附录 12](#_Toc59115825)

[8.1 设计代码、仿真测试代码、约束文件 12](#_Toc59115826)

[8.1.1 设计代码 12](#_Toc59115827)

[8.1.2 仿真文件 27](#_Toc59115828)

[8.1.3 约束文件 30](#_Toc59115829)

[8.2 总体设计图 32](#_Toc59115830)

[8.3 模块仿真截图 32](#_Toc59115831)

[8.3.1 Slide 33](#_Toc59115832)

[8.3.2 Modify 33](#_Toc59115833)

[8.3.3 Opreate 33](#_Toc59115834)

[8.3.4 Valuetodiode 34](#_Toc59115835)

[8.4 小组分工 34](#_Toc59115836)

[8.4.1 陈一帆 34](#_Toc59115837)

[8.4.2 夏时雨 35](#_Toc59115838)

1. 设计要求

通过综合设计性实验，培养学生灵活运用所学知识解决比较复杂的实际问题的能力，使得学生灵活运用学过的计数器、触发器、译码电路等方面的知识，独立完成从设计、选片、连线、调试、排除故障到实现一个数字系统的全过程，详细书写项目报告。

基于EGO1开发板与Vivado集成开发环境，设计一个开锁密码至少为4位数字（或更多）的密码锁。当开锁按扭开关（可设置8位或更多，其中只有4位有效，其余位为虚设）的输入代码等于所设密码时启动开锁控制电路，并且用绿灯亮、红灯灭表示开锁状态。从第一个按扭触动后的5秒内若未能将锁打开，则电路自动复位并发出报警信号，同时用绿灯灭、红灯亮表示关锁状态。可适当添加一些附加功能，如：当操作者开始按动按键能进行倒计时显示。密码锁中的4位密码可以修改，可以记录按键次数等等。

1. 工作原理及系统方框图
   1. 工作原理

该电子密码锁共由四个部分构成：输入控制模块，时钟分频模块，比较修改模块，译码显示模块。

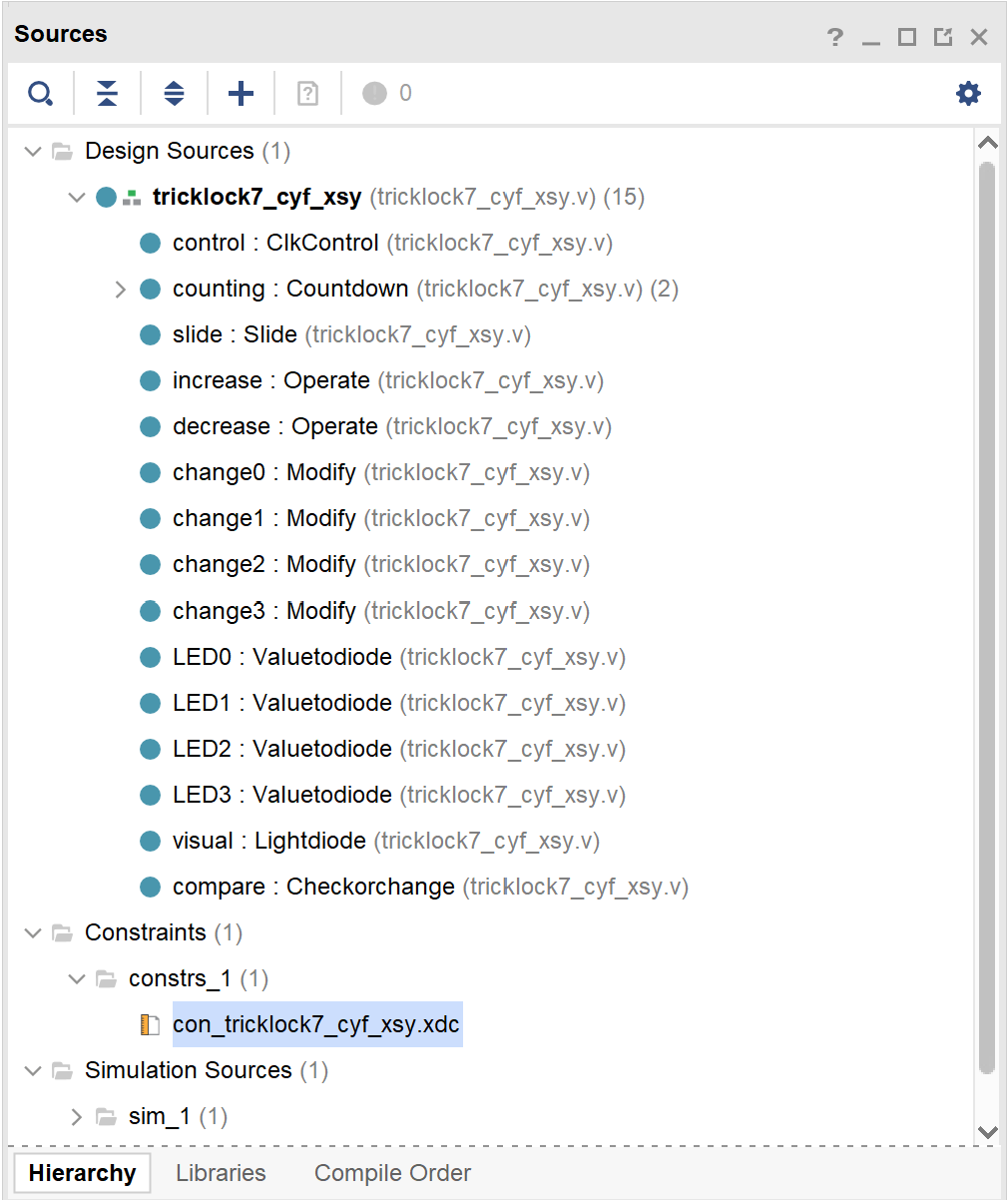
输入控制模块可根据EGO1板上S0~S4上的按键进行输入，进行加，减，左移位，右移位，确认五项操作。密码共4位，每位密码为0~9，故左右移位采用模4计数器的方式，加减采用模10计数器的方式，R1，N4开关分别为模式，重置开关。

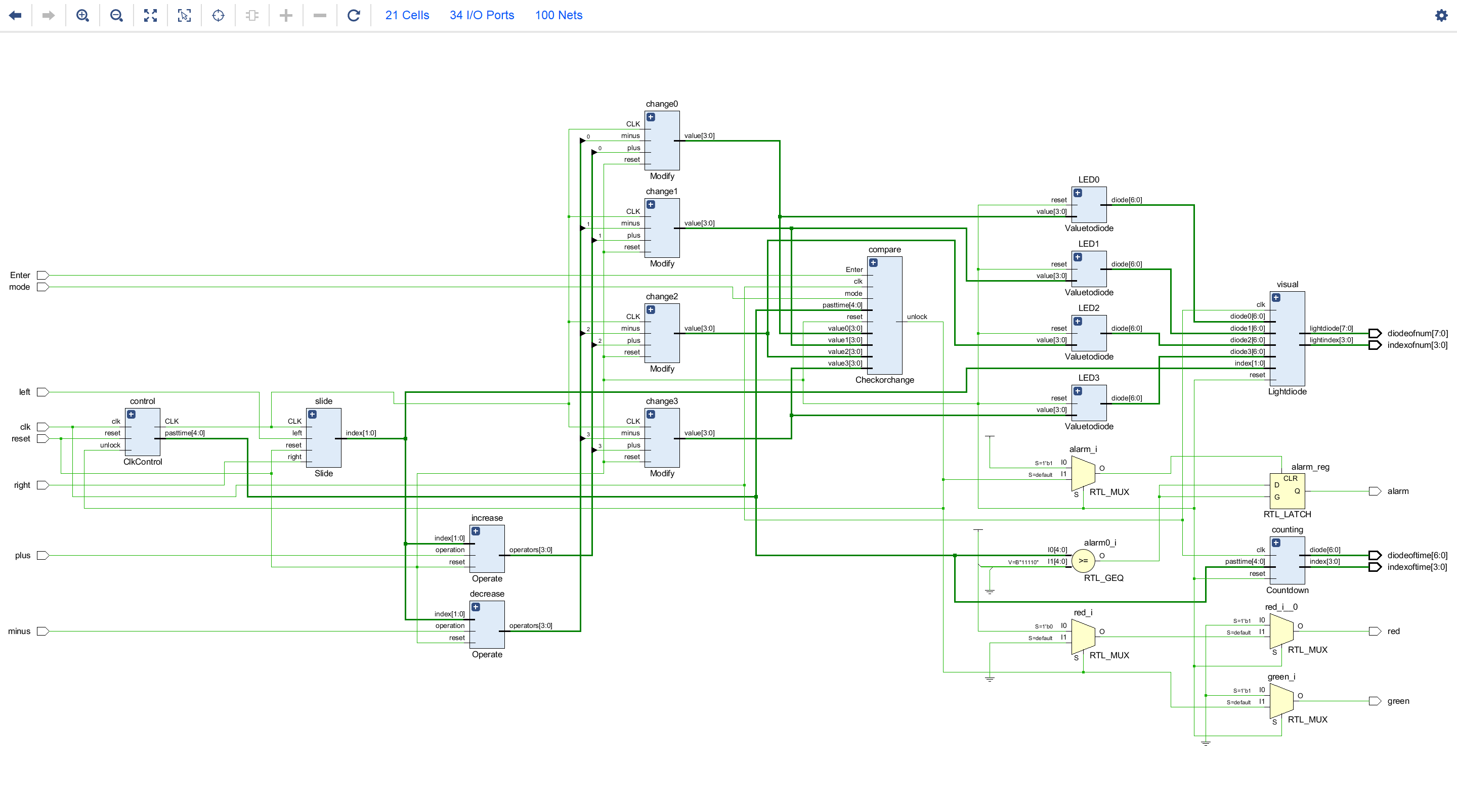
时钟分频模块，对EGO1板上的100MHz时钟进行分频操作，产生20Hz，1Hz的时钟信号。20Hz的时钟信号用于输入等操作时使用。1Hz的时钟信号用于记录第一个按键按下后经过的时间，生成30s的倒计时。

比较修改模块，30s内，当mode为0时可将输入与密码进行比对，不成功可继续输入，mode为1时将密码修改为当前输入。30s后或打开密码锁后，停止一切操作。

译码显示模块，生成十进制的密码及倒计时对应的七段数码管段码，运用分时显示的思想，采取调整位码的方式，从而让密码，倒计时可视化。

## 系统方框图

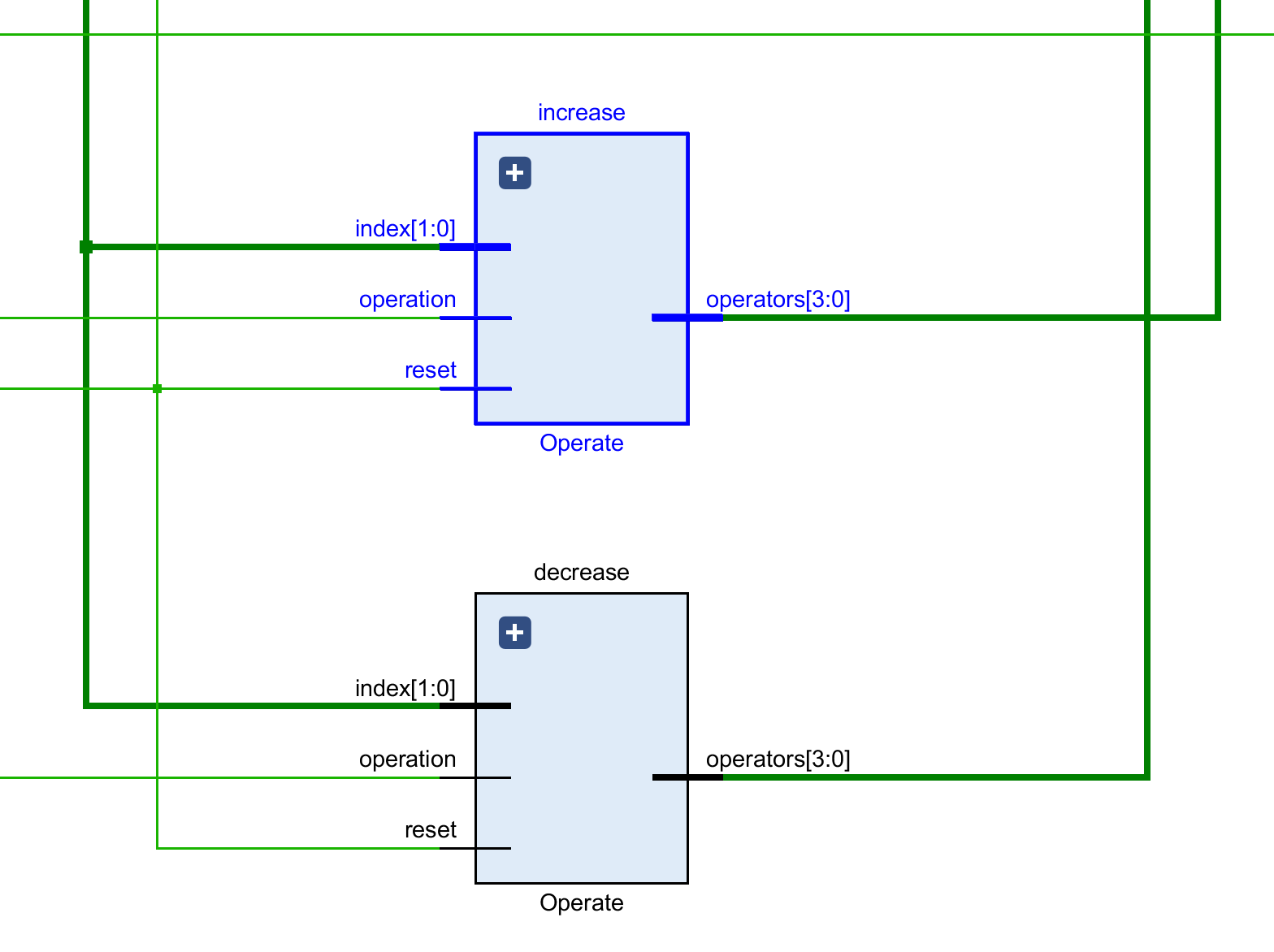




1. 各部分模块具体功能及设计思路

四大部分共涵盖了8个模块：Operate，ClkControl，Slide，Modify，Valuetodiode，Lightdiode，Checkorchange，Countdown。下面逐一介绍其具体功能和设计思路。

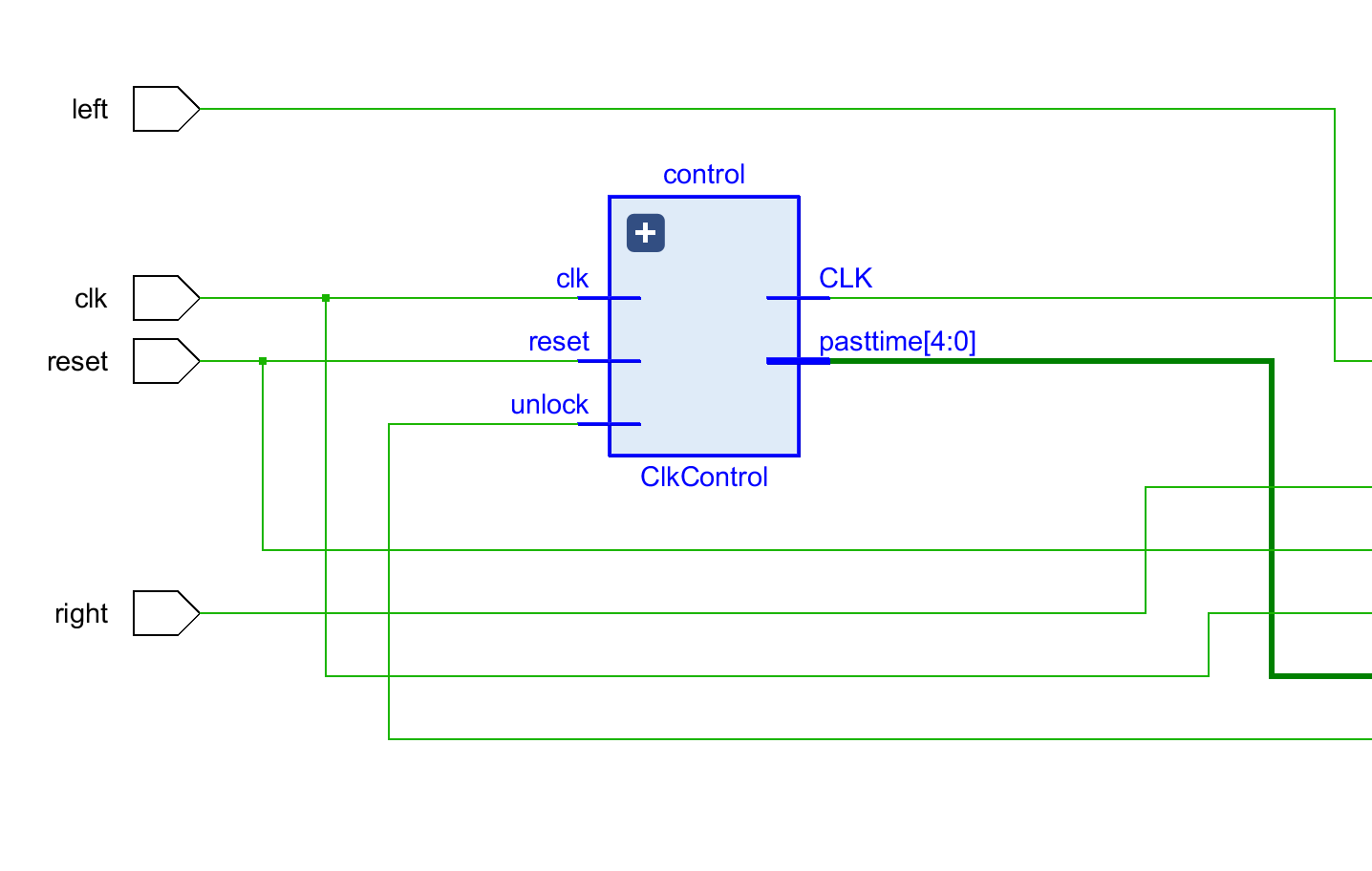
* 1. Operate



操作模块，输入为[1:0]index（操作对象下标），operation（加减操作），reset（重置，电平触发，高有效），输出为reg [3:0] operators。将加减操作（operation）以及操作对象的下标（index）等信息存储在数组operators中。reset表示重置。

输入的下标使用二进制表示，00表示最低位，01表示次低位，10表示次高位，11表示最高位，数组operators根据输入的下标，将输入的操作保存在对应部分。

* 1. ClkControl

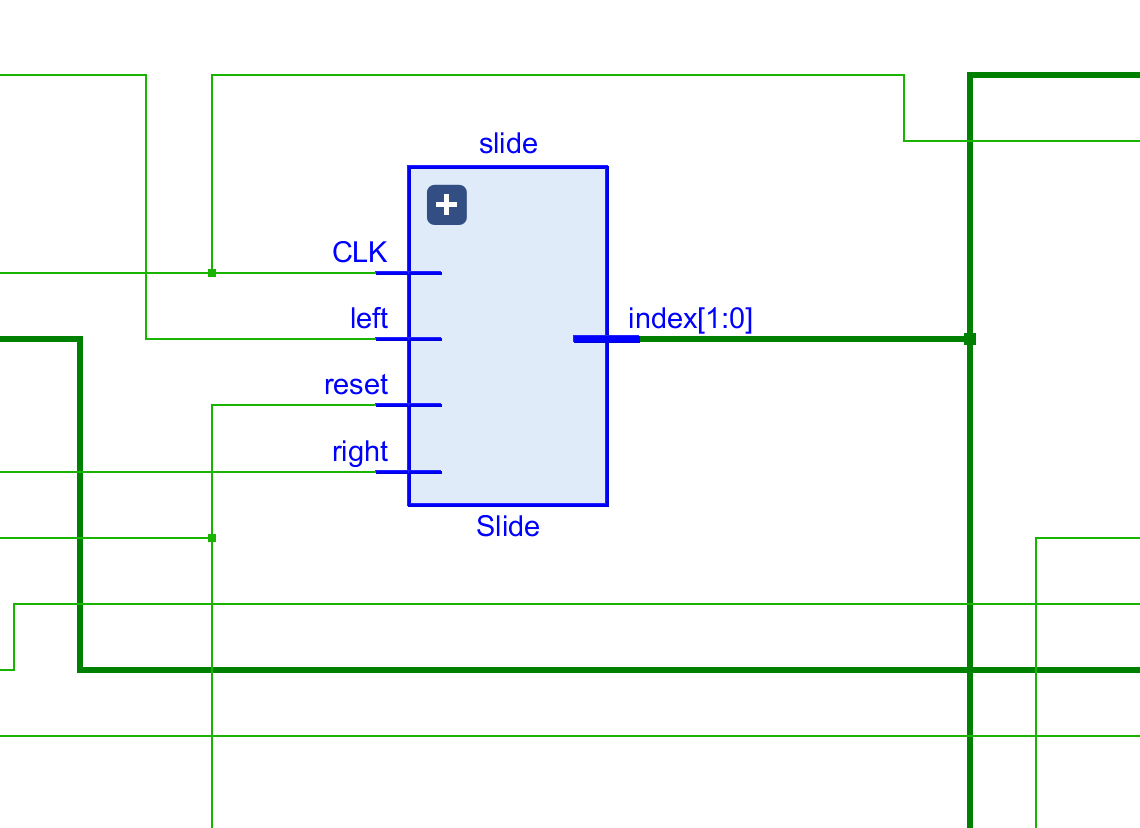


时钟分频模块，输入为reset（重置），clk（EGO1板自带时钟），unlock（密码锁是否解锁）。输出为CLK（20Hz），reg [4:0] pasttime(经过的时间)。此模块生成两个时钟，频率分别为20Hz（CLK），1Hz。分别用于后续操作和倒计时。

CLK之所以为20Hz,因为20Hz是实际测试时较好的一个时钟周期信号。EGO1开发板上的时钟信号每经过5000000个周期，CLK反转一次，当按下第一个按键30s后该时钟分频结束。

EGO1板上时钟1s产生10^8个周期,故每10^8个周期pasttime加一，直到30s或者密码锁被成功打开则停止。

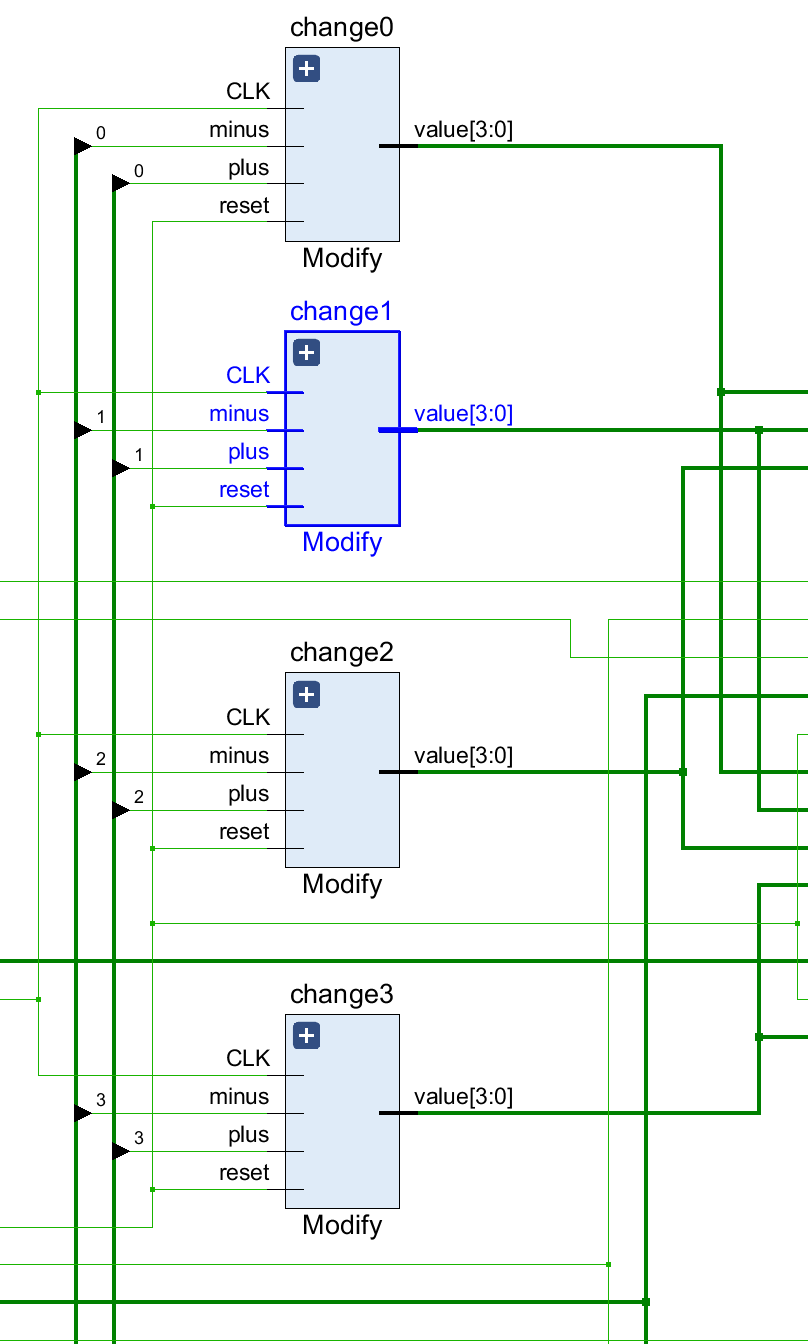
* 1. Slide



移位模块，输入为CLK（20Hz时钟信号），left（左移操作），right（右移操作），reset（重置），输出为reg [1:0] index（下标）。根据输入进行左移，右移，得到加减操作的对象，即输入的第几位。

index初始值为00，表示处于最低位，左移则依次移动到次地位，次高位，高位。根据用户输入的习惯，最高位左移到最低位，最低位右移到最高位，由于是四位密码，此处可使用模4计数器的方法。

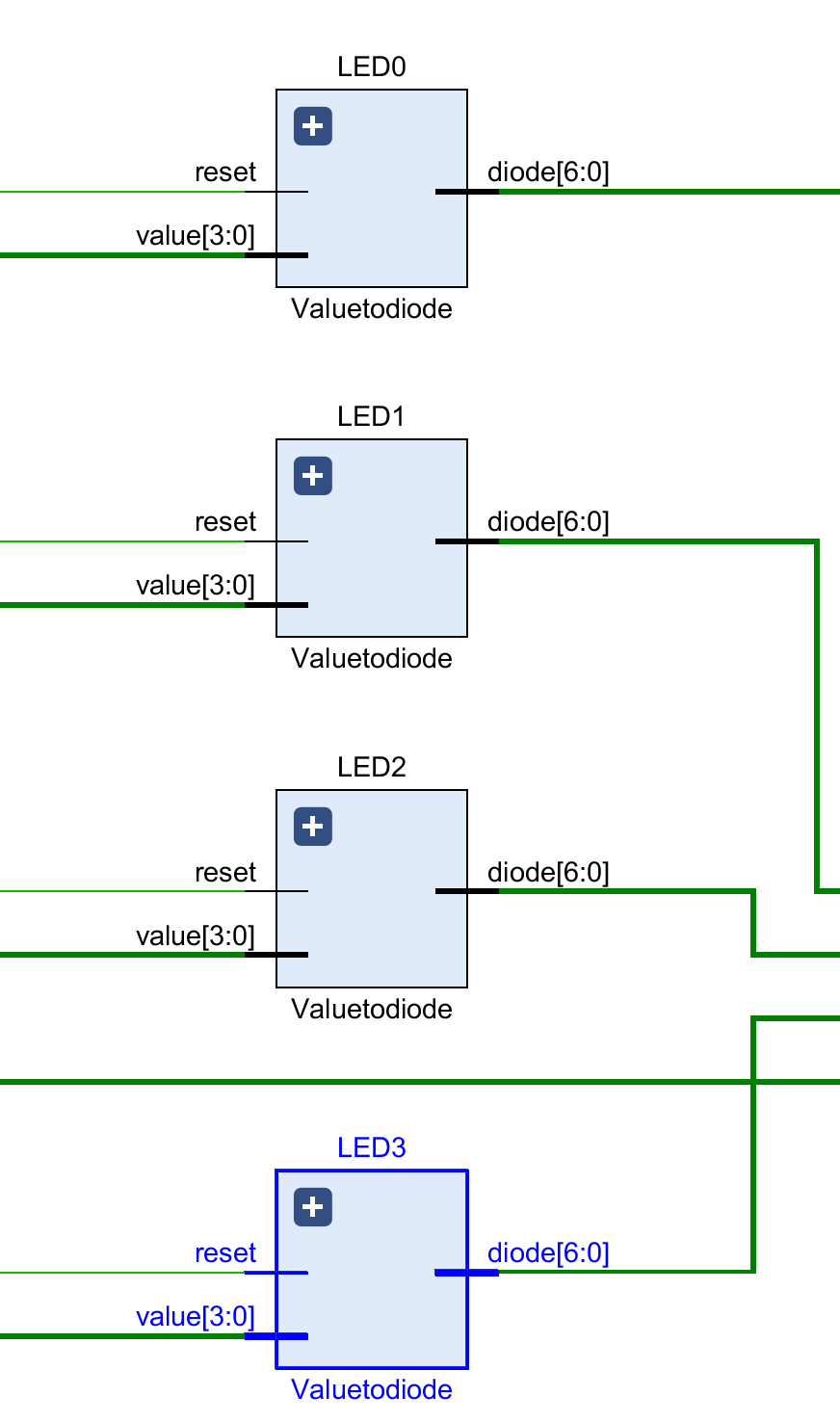
* 1. Modify



修正模块，输入为CLK（20Hz时钟信号），plus（加操作），minus（减操作），reset（重置），输出为reg [3:0] value。对某一位上的密码在0~9范围内进行调整。

初始值value为0，当操作为plus时，value值加一，为minus时，value值减一，考虑到用户输入的习惯，0减一为9，9加一为0，由于密码每一位都在0~9上波动，此处可采用模10计数器的方法。

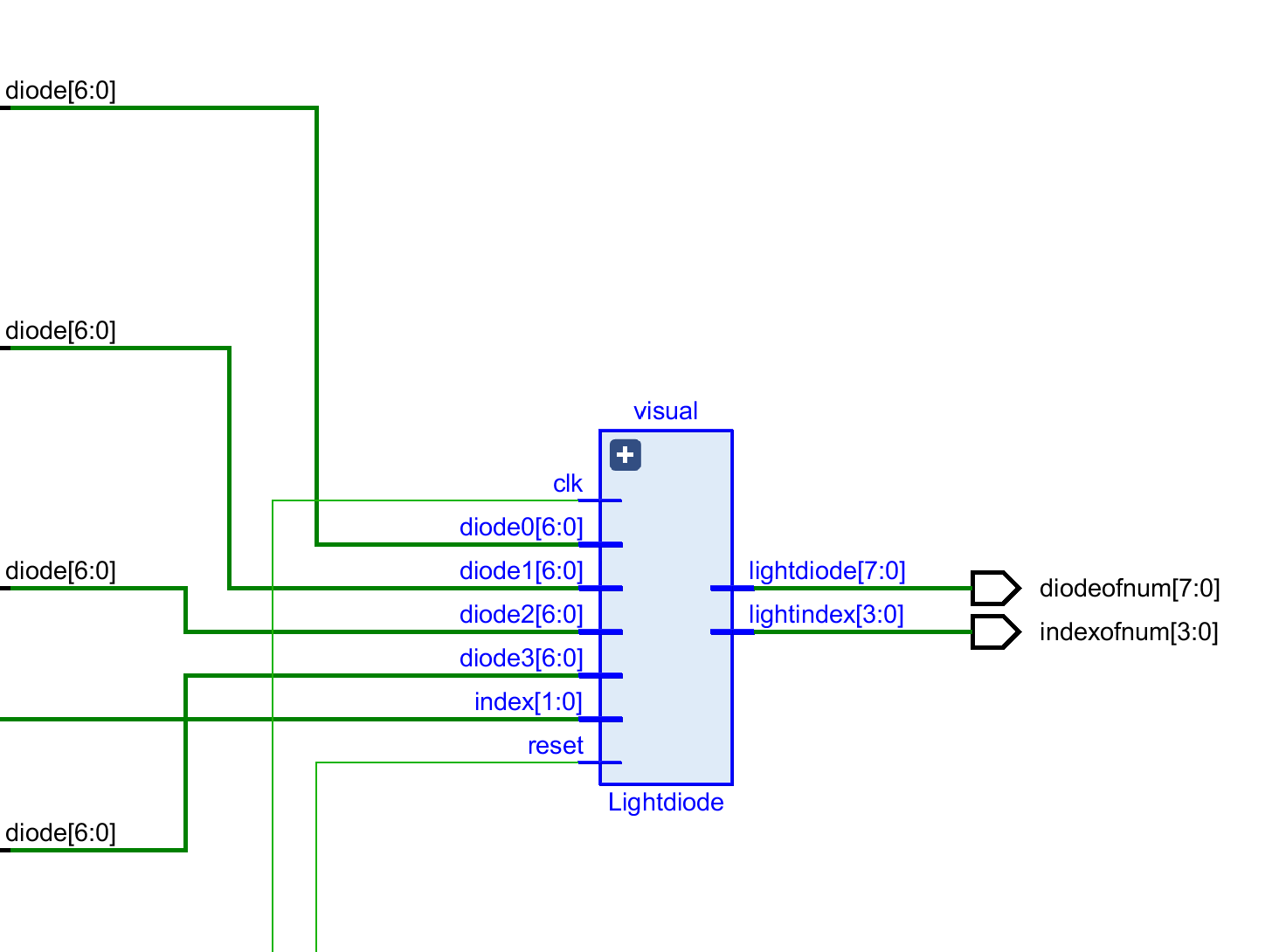
* 1. Valuetodiode



译码模块，输入为[3:0]value（数码管要显示的值），reset（重置），输出为reg [6:0] diode（七段数码管的段码）。根据value生成其对应的段码。

value取值为0~9，每一个十进制数对应一个七位段码，使用switch case语句，将value与diode一一对应。

* 1. Lightdiode

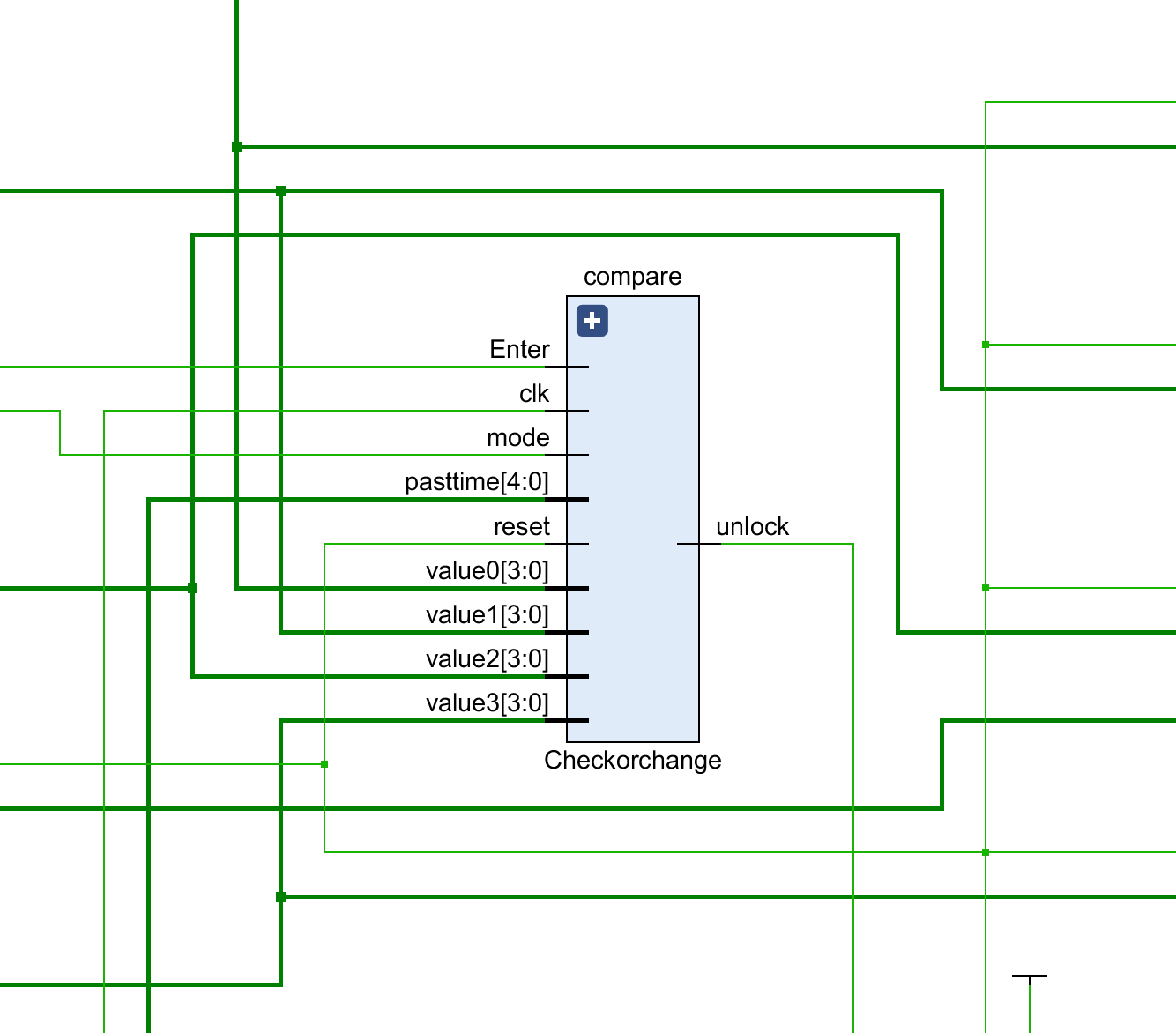


可视化模块，输入为clk（100MHz时钟模块），[6:0] diode0（最低位段码），[6:0] diode1（次低位段码），[6:0] diode2（次高位段码），[6:0]diode3（最高位段码），reset（重置），[1:0]index（被点亮数码管的编号），输出为reg [7；0] lightdiode（被点亮数码管的段码），lightindex（被点亮数码管的位码）。

四个七段数码管，采用分时点亮的方式，每2.5x10^（-4）s点亮一个数码管，由于该时间远小于人的视觉反应时间，因此可以达到同时点亮4个数码管的效果。

首先分频，得到4000Hz的时钟，然后依次点亮四个数码管2.5x10^（-4）s。被点亮数码管的下标，即位码，由index[1:0]来存储，index等于00时点亮最低位，01时点亮次低位，10时点亮次高位，11时点亮最高位。被点亮数码管的段码由对应位的值，经过Valuetodiode模块译码得到。同时，被选中，将要（或正在）修改的值对应的七段数码管的DH位也会被被点亮。

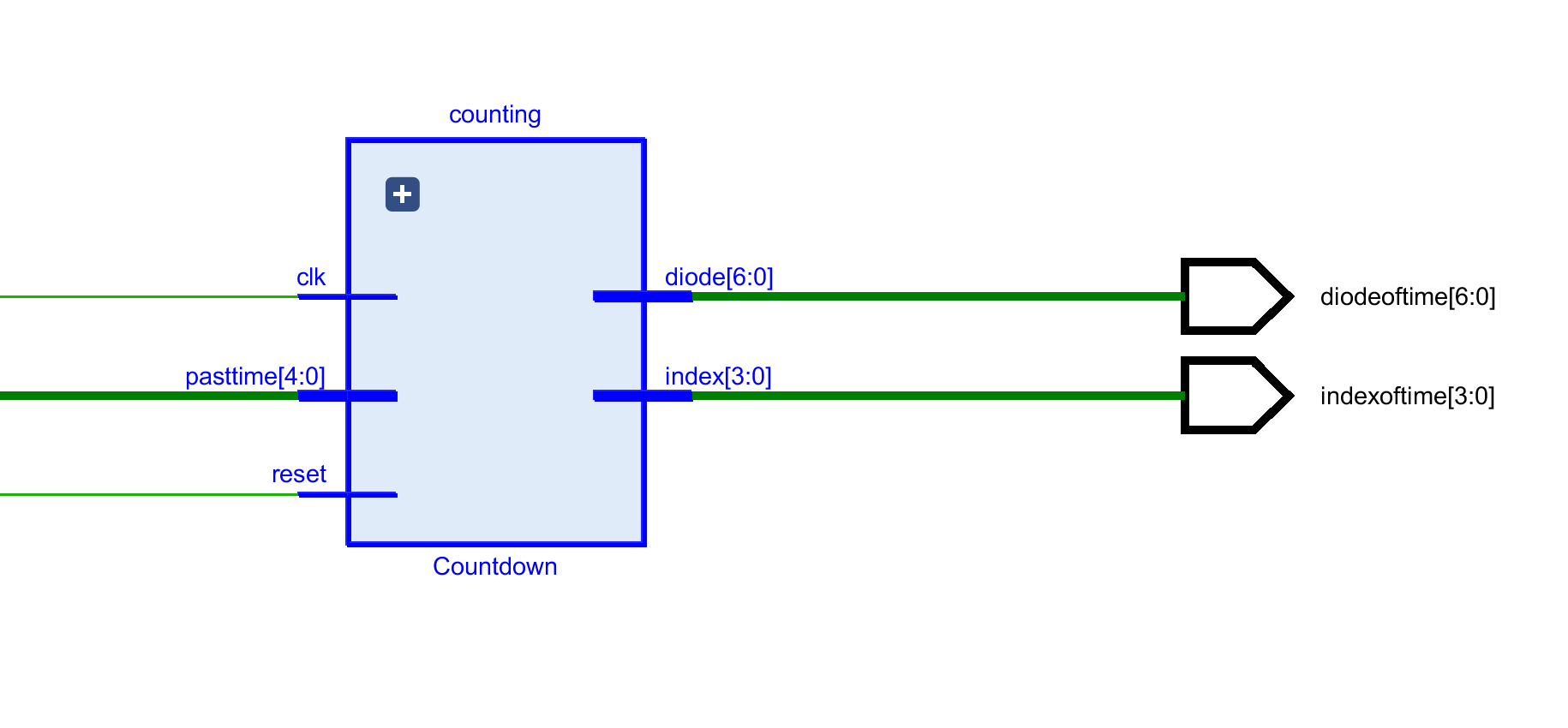
* 1. Checkorchange



比较更新模块，输入有mode（模式，mode=1时更新密码,mode=0时比较核对密码）,[3:0] value0（最低位的值），[3:0] value1（次低位的值），[3:0] value2（次高位的值），[3:0] value3（最高位的值），Enter（确认），clk(100Mhz时钟)，reset（重置），[4:0]pastime（第一个按键按下后经过的时间），输出为unlock（unlock=0表示没有开启，unlock=1表示开启）。

Password初使值为0805，当时间小于30s时可进行操作，当mode=1时按下Enter将密码更新为当前密码,mode=0时按下Enter比较核对密码，成功unlock=1,密码锁打开，失败unlock=0,无任何后果。30s后或开锁停止一切操作。

* 1. Countdown



倒计时及其可视化模块，输入为[4:0] pasttime（自第一个按键按下后后经过的时间），clk（100Hz时钟信号），reset（重置），输出为reg [6:0] diode（被点亮数码管的段码），reg [3:0] index（被点亮数码管的位码）。得到倒计时的十位值，个位值，及其对应的段码，分时点亮时的位码。

首先分频得到4000Hz的时钟，依次点亮两个数码管2.5x10^（-4）s。然后使用数学操作得到十位值，个位值，分别存储，再经过Valuetodiode译码得到十位，个位七段数码管的段码。index根据4000Hz时钟信号，表示位码。

1. 调试过程

在模块的调试过程中，我们遇到过诸多情况。

早期的设计中，输入错误便会停止计时，但是此时可以继续确认，使得绿灯亮起，并且警报灯会报错，多次修改的过程中，也出现过按下一次确认，警报会闪烁一次的情况。最后，我们将之修改为输入错误可以继续输入，只有倒计时结束且密码锁未打开时，警报灯才会亮起。

数码管点亮的过程中曾出现错误，比如数字的显示不正确，后来发现是管脚的绑定，比如diode [6:0] = 7’b0000001，此时diode[0]=1，我们之前在此出现了疏忽。

设计代码中也出现过代码书写错误的情况，如clk（100MHz）与CLK（20Hz）混用的情况。

1. 设计结论

该设计完成全部指定功能，包括：设计一个开锁密码为4位数字的密码锁；当开锁按扭开关的输入代码等于所设密码时启动开锁控制电路，并且用绿灯亮、红灯灭表示开锁状态；从第一个按扭触动后的30秒内若未能将锁打开，则电路复位并发出报警信号，同时用绿灯灭、红灯亮表示关锁状态。

同时实现两个附加功能：当操作者开始按动按钮会经行30s倒计时显示；密码锁中四位密码可修改。

我们设计的密码锁输入端共7个，包括S0~S4，R1，N4。S0为右移位，S1为减操作，S2为确认键，S3为左移位，S4为加操作，通过这五个按键可以修改输入。R1为mode键，当mode=0时，输入正确的密码可以打开密码锁，输入错误密码无影响。当mode=1时，将密码修改为当前输入。N4为重置，当N4=0时密码锁正常运行，当N4=1时，密码锁关闭。

输出端共有9个，包括6个七段数码管，3个灯。左边两位数码管用来显示倒计时，30s倒数计结束时，如果未能打开密码锁，密码锁锁死。右边四为数码管可实时显示输入。J3，J2，K2分别为警报，红灯，绿灯。当密码锁未开启且倒计时未结束时，红灯亮，绿灯灭，警报灯灭。当密码锁开启后，红灯灭，绿灯亮，警报灯灭。当倒计时结束且密码锁未开启时，红灯亮，绿灯灭，警报灯亮。

1. 设计心得与总结
   1. 陈一帆

在设计过程中需要与同学交流，交流对一些模块的构思，对整个密码锁的构想等等。如对输入端的构思，我之前的想法是每次输入一位，输入的值由EGO1板上的10位开关控制，这样显然过于繁琐，后来经过不断的交流，才有了目前的加减，左右移位的输入模式。再者对于数码管的显示，我们之前认为只有两个数码管可点亮，经过同学交流，网上自学，终于找到了分时显示的方法，分时点亮不同的数码管，达到同时点亮的效果。

在网上学习代码的过程中，要看懂代码，并且最好自己写一遍，否则原代码可能和自己的代码不匹配，日后的调试过程中也难以发现问题。

在调试的过程中要保持耐心，即使仿真正确，也不一定能达到预期效果，每次改动代码均需要重新仿真，综合，实现，生成编码文件，在一次次修改的过程要保持耐心。也要注意休息，不能熬夜。

漫长而痛苦的调试过程中，既有看到正确波形图的喜悦，有一次次改动仍不停的高阻态的迷惑困苦，有折腾自己亲手做的密码锁时的喜悦。

* 1. 夏时雨

这次实验让我学会了如何运用vivado设计各种功能的模块，同时通过实验培养我的耐心，面对电脑上的导线和元器件，我需要十足的耐心一步步一条条看，通过查阅资料也让我对更多的设计模块有了更多的了解。也暴露了我的知识短板，增进了我学习的斗志。

1. 参考文献

https://wenku.baidu.com/view/18eb7d1310a6f524ccbf8509.html

https://wenku.baidu.com/view/c2a4d314336c1eb91b375d28.html

http://blog.csdn.net/qq\_23851075/article/details/52783323

https://wenku.baidu.com/view/0294cbb3bb4cf7ec4bfed01a.html

https://www.xilinx.com/support/answers/54074.html

https://www.baidu.com/link?url=IplK3P4\_E2BNlzKR7NrdoUyy\_GKT

-evWrnremCXfPdJ0\_SeRftZRHDqmOrzFqqQd\_jFxRwepg7JhQeN5cwvc7a&

wd=&eqid=943f89cd000758a30000000359493b12

1. 附录
   1. 设计代码、仿真测试代码、约束文件
      1. 设计代码

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 2020/12/17 13:35:30

// Design Name:

// Module Name: tricklock7\_cyf\_xsy

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

//操作模块，对某一位上的值进行操作

module Operate(

input [1:0] index,//加减操作作用的位数

input operation,//加or减操作

input reset,//重置

output reg[3:0] operators//得到加操作数组和减操作数组

);//把加减操作用数组表示

initial

begin

operators<=0;

end

always@(\*)

begin

if(reset==1)

begin

operators<=4'b0;

end

else

begin

case(index)

2'b00:

begin

operators[0]<=operation;

operators[1]<=0;

operators[2]<=0;

operators[3]<=0;

end

2'b01:

begin

operators[0]<=0;

operators[1]<=operation;

operators[2]<=0;

operators[3]<=0;

end

2'b10:

begin

operators[0]<=0;

operators[1]<=0;

operators[2]<=operation;

operators[3]<=0;

end

2'b11:

begin

operators[0]<=0;

operators[1]<=0;

operators[2]<=0;

operators[3]<=operation;

end

default:

operators=0;

endcase

end

end

endmodule

//得到一个分频时钟信号CLK,以及第一个按键按下后经过的时间（0~30s）

module ClkControl(

input reset,//重置

input clk,//100MHz时钟

input unlock,//是否开锁，开锁则停止计时

output reg CLK,//分频后的时间

output reg[4:0] pasttime//30秒倒计时中过了多久

);

reg[30:0]count;

reg [30:0]count1;

parameter n\_CLK=5000000;

parameter n\_countdown=100000000;

initial

begin

CLK=0;

count=0;

count1=0;

pasttime=0;

end

//分频

always @(posedge clk or posedge reset)

begin

if(reset==1)

begin

CLK<=0;

count<=0;

end

else

begin

if(pasttime<30&&unlock==0)

begin

if(count==n\_CLK)

begin

CLK= ~CLK;

count=0;

end

else

begin

count=count+1;

end

end

else

CLK=0;

end

end

always @(posedge clk or posedge reset)

begin

if(reset==1)

begin

count1=0;

pasttime=0;

end

else if(pasttime<30&&unlock==0)

begin

if(count1==n\_countdown)

begin

pasttime<=pasttime+1;

count1<=0;

end

else

count1<=count1+1;

end

end

endmodule

//移位模块，用以选择输入位

module Slide(

input CLK,//时钟

input left,//左移

input right,//右移

input reset,//重置

output reg [1:0]index//下标，表示第几位

);

initial

begin

index<=2'b00;

end

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

index<=2'b00;

end

else

begin

if(left==1)

begin

if(index==2'b11)

index<=2'b00;

else

index<=index+1;

end

if(right==1)

begin

if(index==2'b00)

index<=2'b11;

else

index<=index-1;

end

end

end

endmodule

//修正模块，用以设置某一位上密码的大小

module Modify(

input CLK,//时钟

input plus,//加

input minus,//减

input reset,//重置

output reg[3:0]value//某一位上的值

);

initial

begin

value<=0;

end

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

value<=0;

end

else

begin

if(plus==1)

begin

if(value==4'b1001)

value<=0;

else

value<=value+1;

end

if(minus==1)

begin

if(value==0)

value<=9;

else

value<=value-1;

end

end

end

endmodule

//将某一位上的值变为数码管的七位显示码

module Valuetodiode(

input [3:0] value,//值

input reset,//重置

output reg [6:0] diode//数码管显示码

);

always@(\*)

begin

if(reset==1)

begin

diode = 0;

end

else

begin

case(value)

4'b0000:

diode = 7'b1111110;//reg[6:0]A，则最左边的为A[6]，最右边为A[0]

4'b0001:

diode = 7'b0110000;

4'b0010:

diode = 7'b1101101;

4'b0011:

diode = 7'b1111001;

4'b0100:

diode = 7'b0110011;

4'b0101:

diode = 7'b1011011;

4'b0110:

diode = 7'b1011111;

4'b0111:

diode = 7'b1110000;

4'b1000:

diode= 7'b1111111;

4'b1001:

diode= 7'b1111011;

default:

diode = 7'b1111111;

endcase

end

end

endmodule

//七段数码管分时点亮

module Lightdiode(

input clk,//时钟

input [6:0] diode0,//待表示的数码管最低位

input [6:0] diode1,//待表示的数码管次低位

input [6:0] diode2,//待表示的数码管次高位

input [6:0] diode3,//待表示的数码管最高位

input [1:0] index,//数码管的编号，即位数

input reset,//重置

output reg [7:0] lightdiode,//输出的数码管码

output reg [3:0] lightindex//输出的数码管的编号

);

reg[14:0]count;

reg CLK;

reg [1:0]state;

initial

begin

lightdiode=0;

lightindex=4'b0001;

count=0;

CLK=0;

state=0;

end

parameter n=25000;

always @(posedge clk or posedge reset)//分频

begin

if(reset==1)

begin

count = 0;

CLK = 0;

end

else

begin

if(count==n)

begin

CLK<=~CLK;

count<=0;

end

else

count<=count+1;

end

end

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

lightdiode=0;

lightindex=4'b0001;

state=0;

end

else

begin

if(state==3)

state=0;

else

state=state+1;

case(state)

2'b00:

begin

lightindex<=4'b0001;

if(index==2'b00)

lightdiode<={1'b1,diode0};

else

lightdiode<={1'b0,diode0};

end

2'b01:

begin

lightindex <= 4'b0010;

if(index==2'b01)

lightdiode<={1'b1,diode1};

else

lightdiode<={1'b0,diode1};

end

2'b10:

begin

lightindex <= 4'b0100;

if(index==2'b10)

lightdiode<={1'b1,diode2};

else

lightdiode<={1'b0,diode2};

end

2'b11:

begin

lightindex <= 4'b1000;

if(index==2'b11)

lightdiode<={1'b1,diode3};

else

lightdiode<={1'b0,diode3};

end

endcase

end

end

endmodule

//对输入好的字符进行操作，mode=0表示为输入密码，mode=1表示密码修改为当前字符

module Checkorchange(

input mode,//mode=1修改密码，mode=0输入密码

input [3:0] value0,//最低位数字

input [3:0] value1,//次低位数字

input [3:0] value2,//次高位数字

input [3:0] value3,//最高位数字

input Enter,//确认

input clk,//时钟

input reset,//重置

input [4:0] pasttime,//30s内才可以开锁或者换密码

output reg unlock//解锁状态

);

reg [3:0] password0;

reg [3:0] password1;

reg [3:0] password2;

reg [3:0] password3;

initial

begin

password0<=5;

password1<=0;

password2<=8;

password3<=0;

unlock<=0;

end

always@(posedge clk or posedge reset)

begin

if(reset==1)

begin

unlock<=0;

end

else if(pasttime<30)

begin

if(mode==1)

begin

if(Enter==1)

begin

password0<=value0;

password1<=value1;

password2<=value2;

password3<=value3;

end

end

else

begin

if(Enter==1)

begin

if(password0==value0&&password1==value1&&password2==value2&&password3==value3)

unlock=1;

else

unlock=0;

end

end

end

end

endmodule

//得到倒计时，以及倒计时的段码位码，十位，个位分时显示

module Countdown(

input [4:0] pasttime,//倒计时30s过了多久

input clk,//时钟

input reset,//重置

output reg [6:0] diode,//输出数码管

output reg [3:0] index//输出数码管的位数

);

reg [3:0] value0;

reg [3:0] value1;

wire [6:0] diode0;

wire [6:0] diode1;

reg [14:0] count;

reg CLK;

reg state;

initial

begin

diode=0;

index=0;

value0=0;

value1=0;

count=0;

CLK=0;

state=0;

end

parameter n=25000;

always @(posedge clk or posedge reset)//分频

begin

if(reset==1)

begin

count=0;

CLK=0;

end

else

begin

if(count==n)

begin

CLK=~CLK;

count=0;

end

else

begin

count=count+1;

end

end

end

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

value0<=0;

value1<=0;

end

else

begin

value0<=(30-pasttime)%10;

value1<=(30-pasttime)/10;

end

end

Valuetodiode Decoder0(value0,reset,diode0);

Valuetodiode Decoder1(value1,reset,diode1);

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

index<=0;

diode<=0;

state<=0;

end

else

begin

if(state>=1)

state=0;

else

state=state+1;

case(state)

1'b0:

begin

index<=4'b0100;

diode<=diode0;

end

1'b1:

begin

index<=4'b1000;

diode<=diode1;

end

default:

begin

index <= 4'b0000;

diode <= 0;

end

endcase

end

end

endmodule

module tricklock7\_cyf\_xsy(

input clk,

input plus,

input minus,

input left,

input right,

input mode,

input Enter,

input reset,

output [7:0] diodeofnum,

output [3:0] indexofnum,

output [6:0] diodeoftime,

output [3:0] indexoftime,

output reg green,

output reg red,

output reg alarm

);

//ClkControl

wire CLK;

wire [4:0] pasttime;

//Slide

wire [1:0] index;

//Operate

wire [3:0] plusoperators;

wire [3:0] minusoperators;

//Modify

wire [3:0] value0;

wire [3:0] value1;

wire [3:0] value2;

wire [3:0] value3;

//Valuetodiode

wire [6:0] diode0;

wire [6:0] diode1;

wire [6:0] diode2;

wire [6:0] diode3;

//Output

initial

begin

green=0;

red=0;

alarm=0;

end

ClkControl control(reset,clk,unlock,CLK,pasttime);

Countdown counting(pasttime,clk,reset,diodeoftime,indexoftime);

Slide slide(CLK,left,right,reset,index);

Operate increase(index,plus,reset,plusoperators);

Operate decrease(index,minus,reset,minusoperators);

Modify change0(CLK,plusoperators[0],minusoperators[0],reset,value0);

Modify change1(CLK,plusoperators[1],minusoperators[1],reset,value1);

Modify change2(CLK,plusoperators[2],minusoperators[2],reset,value2);

Modify change3(CLK,plusoperators[3],minusoperators[3],reset,value3);

Valuetodiode LED0(value0,reset,diode0);

Valuetodiode LED1(value1,reset,diode1);

Valuetodiode LED2(value2,reset,diode2);

Valuetodiode LED3(value3,reset,diode3);

Lightdiode visual(clk,diode0,diode1,diode2,diode3,index,reset,diodeofnum,indexofnum);

Checkorchange compare(mode,value0,value1,value2,value3,Enter,clk,reset,pasttime,unlock);

always@(unlock or Enter or mode or reset or pasttime)

begin

if(reset==1)

begin

green<=0;

red<=0;

alarm<=0;

end

else

begin

if(unlock==0)

begin

green<=0;

red<=1;

if(pasttime>=30)

begin

alarm<=1;

end

end

else

begin

green<=1;

red<=0;

alarm<=0;

end

end

end

endmodule

* + 1. 仿真文件

module sim\_Slide(

);

reg CLK,left,right,reset;

wire [1:0] index;

Slide Slide\_test(CLK,left,right,reset,index);

initial

begin

CLK=0;left=0; right=0;reset=0;

end

always #7.5 CLK<=~CLK;

always #120 reset<=~reset;

always@(posedge CLK or posedge reset)

begin

if(reset==1)

begin

CLK=0;left=0; right=0;reset=0;

end

else

begin

#15 left <= 1;right <= 0;

#15 left <= 1;right <= 0;

#15 left <= 1;right <= 0;

#15 left <= 1;right <= 0;

#15 right <= 1;left <= 0;

#15 right <= 1;left <= 0;

#15 right <= 1;left <= 0;

#15 right <= 1;left <= 0;

end

end

endmodule

module sim\_Modify();

reg CLK,plus,minus ,reset;

wire [3:0]value;

Modify test(CLK,plus,minus,reset,value);

initial

begin

CLK=0;plus=0;minus=0;reset=0;

end

always #7.5 CLK <= ~CLK;

always #300 reset=~reset;

always

begin

if(reset==1)

begin

CLK=0;plus=0;minus=0;reset=0;

end

else

begin

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 plus <= 1;minus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

#15 minus <= 1;plus <= 0;

end

end

endmodule

module sim\_Opreate();

reg [1:0] index;

reg operation;

reg reset;

wire [3:0] Operators;

Operate test(index,operation,reset,Operators);

initial

begin

index = 0;

operation = 0;

reset=0;

end

always #80 reset=~reset;

always

begin

if(reset==0)

begin

#10 index = 0;operation = 1;

#10 index = 1;operation = 1;

#10 index = 2;operation = 1;

#10 index = 3;operation = 1;

#10 index = 0;operation = 0;

#10 index = 1;operation = 0;

#10 index = 2;operation = 0;

#10 index = 3;operation = 0;

end

else

begin

index = 0;

operation = 0;

reset=0;

end

end

endmodule

module sim\_valuetodiode();

reg [3:0] value;

reg reset;

wire [6:0] diode;

Valuetodiode test(value,reset,diode);

initial

begin

value=0;

reset=0;

end

always #100 reset=~reset;

always

begin

if(reset==0)

begin

#10 value<=0;

#10 value<=1;

#10 value<=2;

#10 value<=3;

#10 value<=4;

#10 value<=5;

#10 value<=6;

#10 value<=7;

#10 value<=8;

#10 value<=9;

end

else

begin

value=0;

reset=0;

end

end

endmodule

* + 1. 约束文件

set\_property PACKAGE\_PIN P17 [get\_ports clk]

set\_property IOSTANDARD LVCMOS33 [get\_ports clk ]

set\_property PACKAGE\_PIN K2 [get\_ports green]

set\_property IOSTANDARD LVCMOS33 [get\_ports green]

set\_property PACKAGE\_PIN J2 [get\_ports red]

set\_property IOSTANDARD LVCMOS33 [get\_ports red]

set\_property PACKAGE\_PIN J3 [get\_ports alarm]

set\_property IOSTANDARD LVCMOS33 [get\_ports alarm]

set\_property PACKAGE\_PIN R1 [get\_ports mode]

set\_property IOSTANDARD LVCMOS33 [get\_ports mode]

set\_property PACKAGE\_PIN R15 [get\_ports Enter]

set\_property IOSTANDARD LVCMOS33 [get\_ports Enter]

set\_property PACKAGE\_PIN N4 [get\_ports reset]

set\_property IOSTANDARD LVCMOS33 [get\_ports reset]

set\_property PACKAGE\_PIN G6 [get\_ports {indexofnum[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexofnum[0]}]

set\_property PACKAGE\_PIN E1 [get\_ports {indexofnum[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexofnum[1]}]

set\_property PACKAGE\_PIN F1 [get\_ports {indexofnum[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexofnum[2]}]

set\_property PACKAGE\_PIN G1 [get\_ports {indexofnum[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexofnum[3]}]

set\_property PACKAGE\_PIN D2 [get\_ports {diodeofnum[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[0]}]

set\_property PACKAGE\_PIN E2 [get\_ports {diodeofnum[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[1]}]

set\_property PACKAGE\_PIN F3 [get\_ports {diodeofnum[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[2]}]

set\_property PACKAGE\_PIN F4 [get\_ports {diodeofnum[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[3]}]

set\_property PACKAGE\_PIN D3 [get\_ports {diodeofnum[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[4]}]

set\_property PACKAGE\_PIN E3 [get\_ports {diodeofnum[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[5]}]

set\_property PACKAGE\_PIN D4 [get\_ports {diodeofnum[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[6]}]

set\_property PACKAGE\_PIN H2 [get\_ports {diodeofnum[7]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeofnum[7]}]

set\_property PACKAGE\_PIN B2 [get\_ports {diodeoftime[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[0]}]

set\_property PACKAGE\_PIN B3 [get\_ports {diodeoftime[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[1]}]

set\_property PACKAGE\_PIN A1 [get\_ports {diodeoftime[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[2]}]

set\_property PACKAGE\_PIN B1 [get\_ports {diodeoftime[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[3]}]

set\_property PACKAGE\_PIN A3 [get\_ports {diodeoftime[4]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[4]}]

set\_property PACKAGE\_PIN A4 [get\_ports {diodeoftime[5]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[5]}]

set\_property PACKAGE\_PIN B4 [get\_ports {diodeoftime[6]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {diodeoftime[6]}]

set\_property PACKAGE\_PIN C1 [get\_ports {indexoftime[0]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexoftime[0]}]

set\_property PACKAGE\_PIN H1 [get\_ports {indexoftime[1]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexoftime[1]}]

set\_property PACKAGE\_PIN C2 [get\_ports {indexoftime[2]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexoftime[2]}]

set\_property PACKAGE\_PIN G2 [get\_ports {indexoftime[3]}]

set\_property IOSTANDARD LVCMOS33 [get\_ports {indexoftime[3]}]

set\_property PACKAGE\_PIN V1 [get\_ports left]

set\_property IOSTANDARD LVCMOS33 [get\_ports left]

set\_property PACKAGE\_PIN R11 [get\_ports right]

set\_property IOSTANDARD LVCMOS33 [get\_ports right]

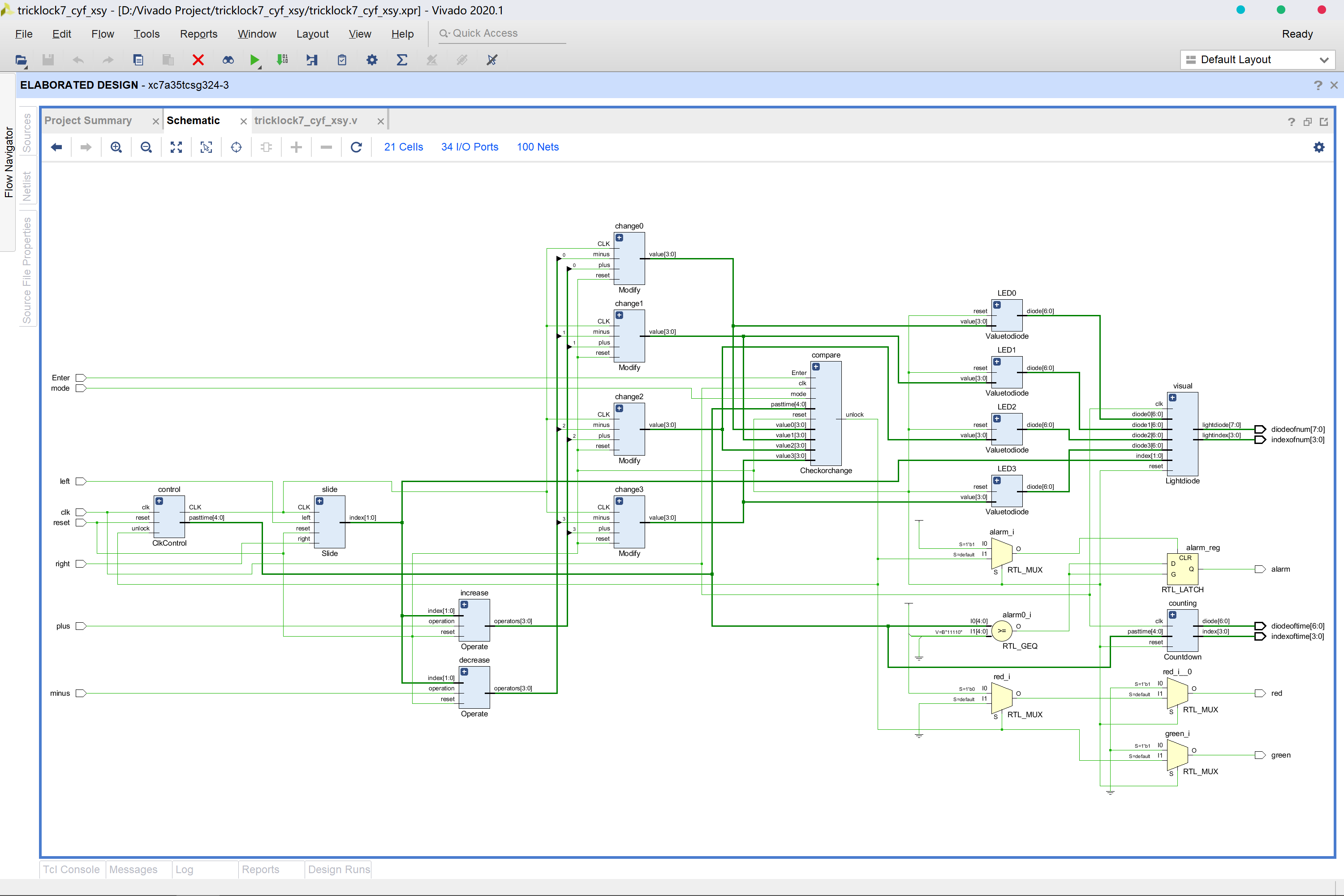
set\_property PACKAGE\_PIN U4 [get\_ports plus]

set\_property IOSTANDARD LVCMOS33 [get\_ports plus]

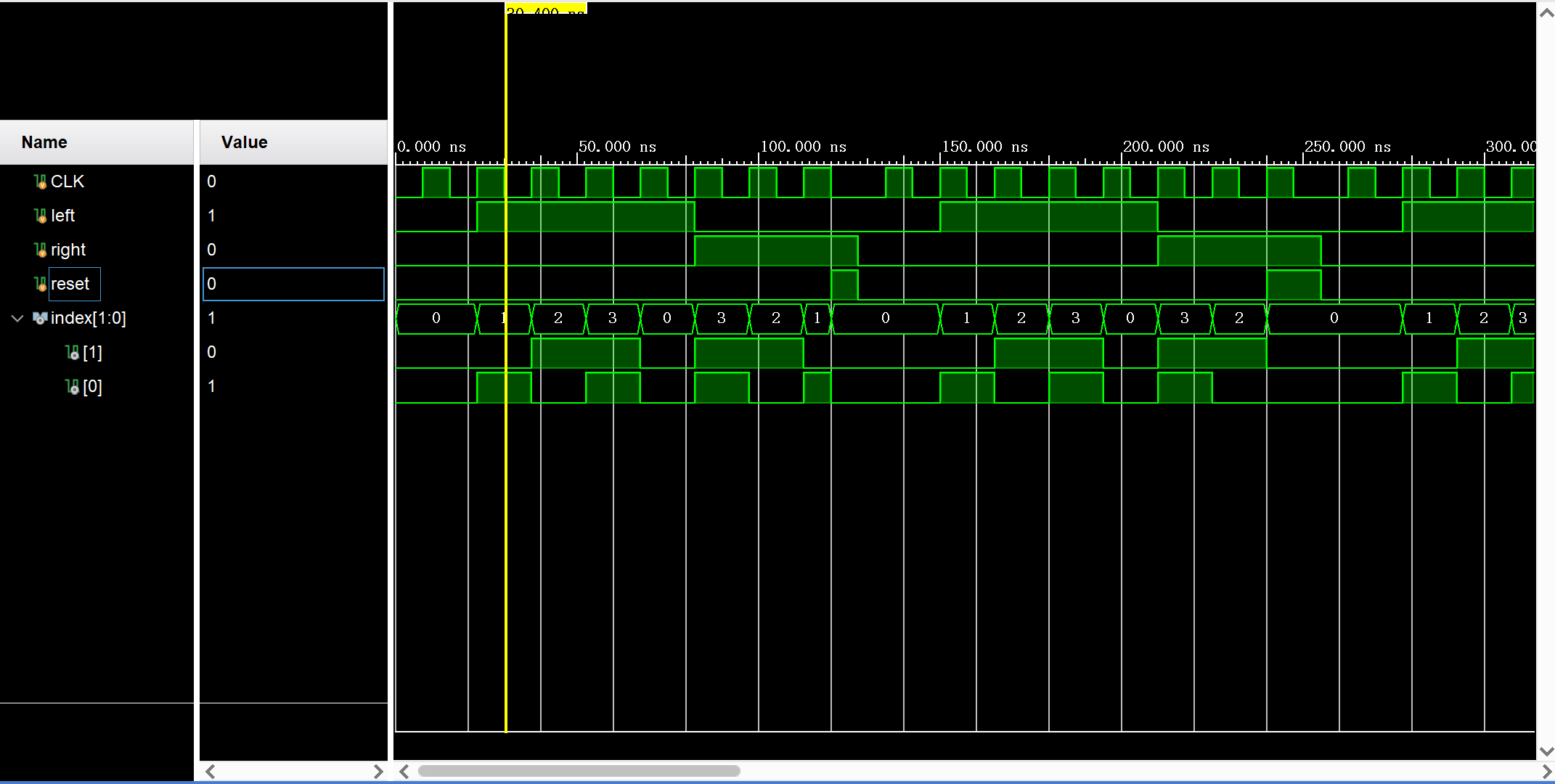
set\_property PACKAGE\_PIN R17 [get\_ports minus]

set\_property IOSTANDARD LVCMOS33 [get\_ports minus]

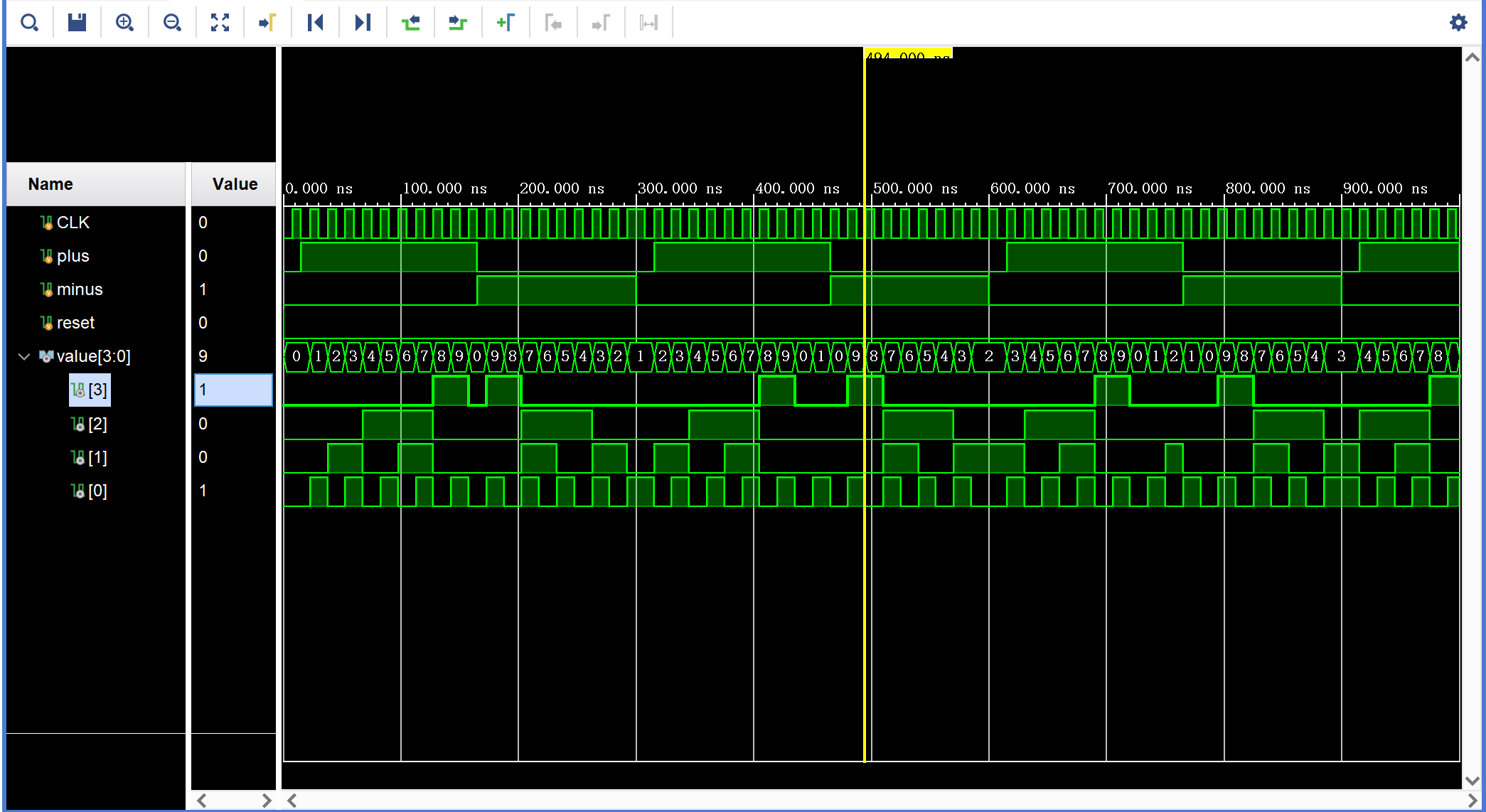
* 1. 总体设计图



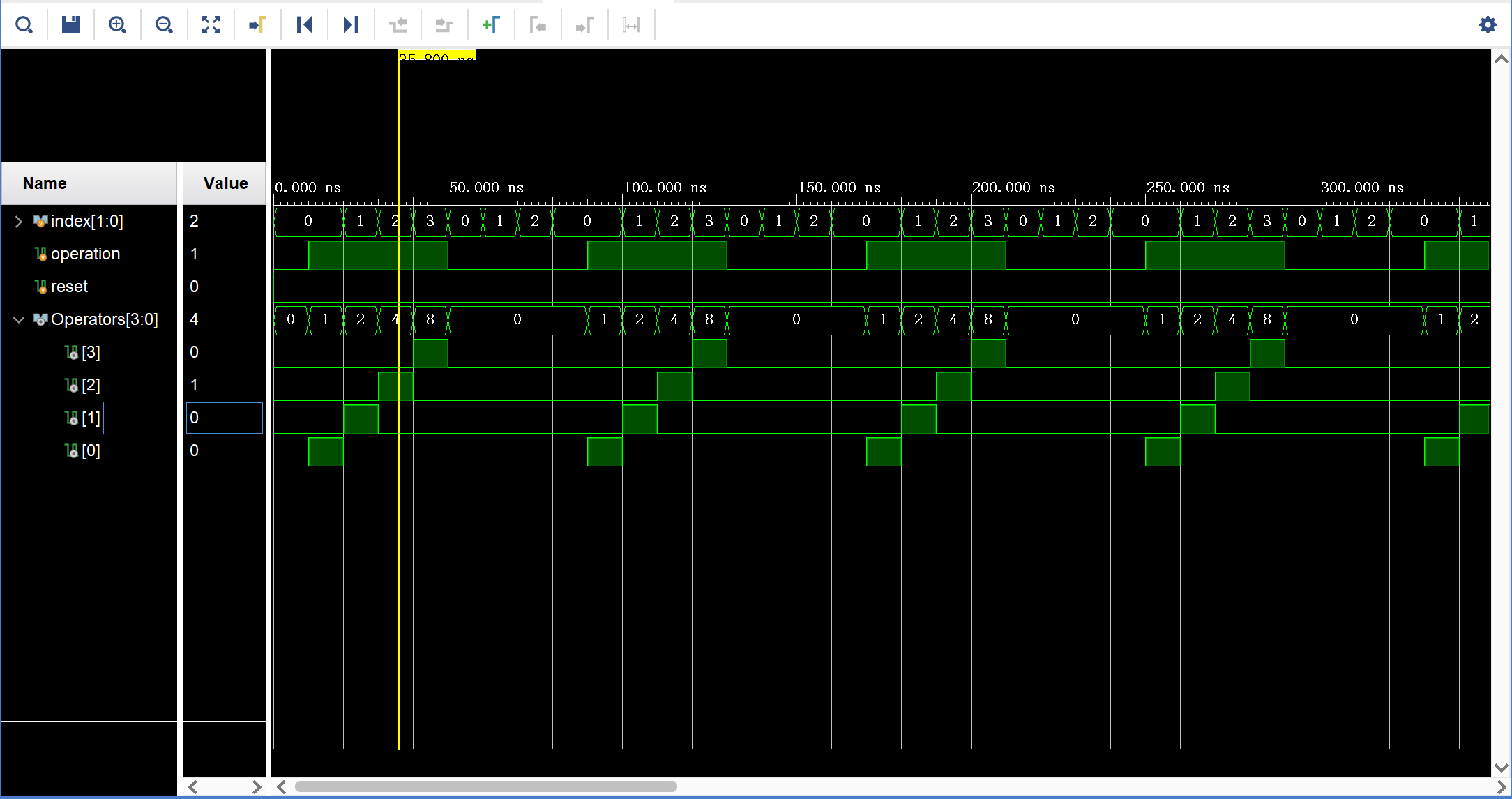
* 1. 模块仿真截图
     1. Slide



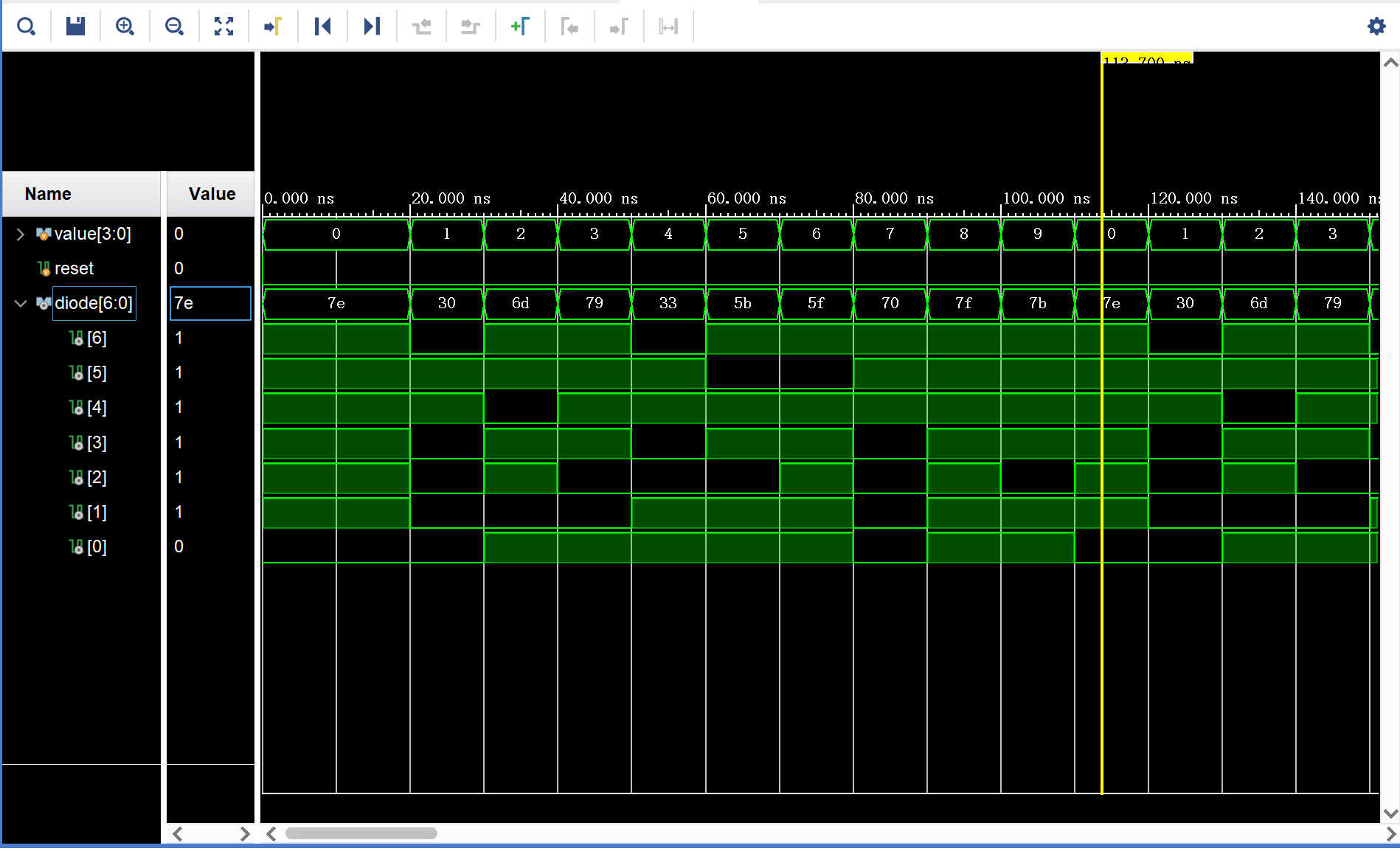
* + 1. Modify



* + 1. Opreate



* + 1. Valuetodiode



(注：此处diode[0]对应7号管，diode[6]对应1号管)

* 1. 小组分工
     1. 陈一帆

负责主体构思，Opreate，Slide，Modify，Valuetodiode，Lightdiode，Checkorchange设计文件的编写，仿真文件的编写，调试，报告编写。

权重分配：1.0

* + 1. 夏时雨

负责ClkControl，Countdown设计文件的编写，约束文件的编写。

权重分配：0.8