**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 孙莹莹 | | **年级** | | *2020* |
| **学号** | | *20204104* | | **专业、班级** | | *计算机科学与技术卓越1班* |
| **实验名称** | **项目三 电梯控制器设计** | | | | | |
| **实验时间** | **2021.12.09** | | **实验地点** | | **DS1404** | |
| **实验成绩** |  | | **实验性质** | | **□验证性 □设计性 □综合性** | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的  通过实验，巩固有限状态机设计方法，并设计实现一个电梯控制器。 | | | | | | |
| 二、实验项目内容  利用 BASYS 开发板资源设计一个 5 层楼的电梯控制器系统，并能在开发板上 模拟电梯运行状态，具体要求如下：  1) 利用开发板的 5 个按键作为电梯控制器的呼叫按钮；  2) 利用 led 灯分别显示楼层 1~5 的呼梯状态；  3) 利用数码管显示电梯运行时电梯所在楼层；  4) 利用时钟分频设计电梯控制器控制电梯每秒运行一层。 | | | | | | |
| 1. 实验设计   1、电梯控制器系统控制流程图（电梯厢内视角）  电梯停在某一层  电梯上升  电梯下降  电梯停  电梯停  电  初始化  上层叫梯  下层叫梯  叫梯服务  无  本层是否叫梯  本层是  无  无  上层是否叫梯  下层是  有  2.设计思路   1. 系统输入/输出变量   时钟输入clk；按键输入btn；数码管显示输出seg；叫梯楼层状态灯输出nfloor。   1. 按键设计   本实验使用板上 5 个按键按钮模拟电梯的叫梯按键，1 层按键为 BTNU， 2 层按键为 BTNL，3 层按键为 BTNC，4 层按键为 BTNR，5 层按键为 BTND。 所以，定义一个 5 位按键寄存器 btn\_pre\_re，同时考虑到防抖（按键按下去和松开会产生抖动现象会影响到我们的操作），故防抖设计为每 200ms 读取一次叫梯按键信息。  对于电梯按键，当没有叫梯时，按键相应的 LED 指示灯应处于熄灭 状态；当有叫梯时，按键相应的 LED 指示灯应处于点亮状态；当在某一层 已经叫梯，但是由于某种原因发现所叫梯不是自己想要的梯层时，能够取 消此层的叫梯状态。   1. 显示设计   电梯控制器包括两种显示，即数码管显示电梯所在楼层和 LED 灯显 示所叫楼层服务。  3.模块功能图  IMG_0843(20211225-085549)  4.所用器件  RTL Memory,RTL Register,RTL Multiplexer | | | | | | |
| 四、实验过程或算法  （1）顶层模块  module elevator\_top(  input clk,  input [5:1] btn,  output [5:1] led,  output [3:0] an,  output [7:0] segs  );  wire flag;  wire [2:0] pos;  reg [5:1] btn\_off;  wire [5:1] led;  wire clk\_1Hz;    always@(pos) begin  btn\_off=5'b11111;  btn\_off[pos] =0;  end    clock u0(clk,~flag,clk\_1Hz);  segs u1(clk,{9'b0,pos},an,segs);  debkey u2(clk,btn,btn\_off,led);  elevator u3(led,clk\_1Hz,pos,flag);  Endmodule   1. 分频模块   module clock(  input clk,  input rst,  output clk\_1Hz  );  parameter N=49999999;    reg clk\_1Hz=1'b0;  integer count=0;    always@ (posedge clk or posedge rst)  begin  if(rst)  begin  clk\_1Hz<=0;  count<=0;  end  else  begin  if(count < N)  count <= count + 1;  else  begin  count<=0;  clk\_1Hz<=~clk\_1Hz;  end  end  end  endmodule   1. 七段管输出控制   module segs(  input clk,  input [11:0] s,  output reg [3:0] an,  output reg [7:0] segs  );  reg [20:0] count=0; //计数，实现降频功能  always@(posedge clk)  begin  count=count+1;  end    reg [3:0] data=4'b0000; //显示的数值  initial begin  an=4'b1110;  end    always@(posedge clk) //段控制  case(count[20:19]) //通过count来控制现在输出的数据和位置  2'b00:  begin an=4'b1101; data=s[5:3];end  2'b01:  begin an=4'b1011; data=s[8:6];end  2'b10:  begin an=4'b0111; data=s[11:9];end  2'b11:  begin an=4'b1110; data=s[2:0];end  endcase    always@(posedge clk) //位控制  case(data) //将data的值显示在数码管上  3'h0: segs=8'b11111111; //没有显示  3'h1: segs=8'b10011111; //显示1，下同  3'h2: segs=8'b00100101;  3'h3: segs=8'b00001101;  3'h4: segs=8'b10011001;  3'h5: segs=8'b01001001;  3'h6: segs=8'b01000001;  3'h7: segs=8'b00011111;  3'h8: segs=8'b00000001;  3'h9: segs=8'b00001001;  default:  segs=8'b11111111;  endcase  （4）电梯按键控制模块  module debkey(  input clk,  input [5:1] btn,  input [5:1] btn\_off,  output [5:1] btn\_pre\_re  );  //每 200ms 读取一次叫梯按键信息  parameter N=99\_999999;  integer count=0;  reg clk\_200ms;  always@(posedge clk) begin  clk\_200ms<=0;  if(count<N/5)  count<=count+1;  else begin  count<=0;  clk\_200ms<=1;  end  end    //叫梯按键赋值程序如下  reg [5:1] btn\_pre\_re;  always@(posedge clk\_200ms) begin  btn\_pre\_re=btn\_pre\_re^btn;  btn\_pre\_re=btn\_pre\_re&btn\_off;  end    endmodule  （5）电梯运行控制模块----更改当前所在楼层  module elevator(  input [5:1] call,  input clk,  output reg [2:0] floor,  output flag  );    integer i;  wire up;  wire down;    initial  floor=1;    judge j(floor,call,up,down);    always@(posedge clk) begin  if(up) begin  floor=floor+1;  end  else if(down)begin  floor=floor-1;  end  else  floor=floor;  end    assign flag=up|down;  Endmodule   1. 电梯运行控制子模块---判断上行或下行   module judge(  input [2:0] floor,  input [5:1] call,  output up,  output down  );    integer i;  reg up;  reg down;    always@(\*)  begin  down=0;  up=0;  for(i=1;i<=5;i=i+1)  begin  if(call[i]==1 && i<floor) begin  down=1;  end  else if(call[i]==1 && i>floor) begin  up=1;  end  end  end  endmodule | | | | | | |
| 五、实验过程中遇到的问题及解决情况  1.为使led灯显示清晰，分频为1s刷新1次。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  1.仿真文件如下  // Module Name: elevator\_sim  `timescale 1ns / 1ps  module elevator\_sim();  reg [4:0] call;  reg clk=0;  wire [2:0] floor;    always #20 clk=~clk;    elevator u(call,clk,floor,up,down);    initial begin  #20 call=5'b00100;  #220 call=5'b10000;  #100 call=5'b00010;    end  endmodule  2.仿真结果如下    3.资源利用如下    4.上板测试   * 呼叫，取消呼叫   见附件视频   * 可同时多层呼叫     七、小组分工情况说明  *刘淑文：代码编写，上板测试，实验报告撰写*  *孙莹莹：代码编写，上板测试，实验报告撰写* | | | | | | |