**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 罗帆靖 | | **年级** | | 21级 |
| **学号** | | 20214865 | | **专业、班级** | | 计算机科学与技术卓越一班 |
| **实验名称** | 电梯控制器设计 | | | | | |
| **实验时间** | 2022.11.30 | | **实验地点** | | 线上 | |
| **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。 | | | | | | |
| 三、实验设计  1、电梯控制器系统控制流程图（电梯厢内视角）    2、根据电梯模型，设计出模块结构模型。    系统输入/输出变量：  clk系统时钟，rst异步置位，en异步使能（作为系统的开关），in[4:0]/cancel[4:0]分别为五个楼层的叫楼和取消叫楼；  en\_led指示系统工作状态，led[4:0] 五位叫楼指示灯，a2g/sel分别为数字灯的使能和选择信号。  信号输入预处理：  在①模块中，主要对输入信号进行处理，其中主要是对systemclk分频，获取用户输入并且对机械按键输入的信号做消抖处理。  呼楼状态更新：  在②模块中，主要根据①模块传输过来的用户呼楼和取消呼楼的情况，对内部各个楼层的呼楼状态进行更新，并且将各个楼层的呼楼状态根据LED灯输出。在具体代码上，我们采用for循环搭建五个相同的电路对每个楼层的状态进行检测，需要保证取消信号的优先级要高于生成信号。  电梯运动状态和楼层更新：  根据显示电梯模型，我们将电梯的运动状态抽象出五个状态，分别是：  stop：停状态，此时电梯的运动趋势为0，用物理的术语来说是加速度为0；  upRun：上行运动状态，此时电梯正在向上走；  upStop：此时电梯有向上运动的趋势，但是在这里停留（有客要下）；  downRun：下行运动状态，此时电梯正在向下走；  downStop：此时电梯有向下运动的趋势，但是在这里停留；  定义输入为：  cur:当前楼层呼楼状态为1；  up:当前楼层向上有楼层呼楼状态为1；  down:当前楼层向下有楼层呼楼状态为1；  同时可以画出状态转换图：    在图中有一个细节，我们在设计时，由于考虑到up、down同时出现的情况，需要设置一个优先级（优先向上还是向下），我们采用优先向上的方式，因此down状态需要以up==0为前提。  此时电梯运动状态转移的输出out将直接导致电梯楼层的变化，我们设想中电梯运动状态转移应该伴随着楼层改变的实时响应，由于三段式状态机会导致state的响应晚一个时序，为了避免时序的错乱，我们采用两段式状态机设计。  而电梯楼层的变化并不能简单只根据运动状态的改变，还需要根据(up、down、cur)信号，因此我们需要采用mealy型状态机，输出与现态和输入都有关。具体关系如下：  若当前楼层为X，当状态为upRun且输入up==1&&cur==0时，将next\_floor设为X+1；当状态为downRun且输入down==1&&cur==0时，将next\_floor设为X-1；其他情况下，next\_floor均为X；  数字灯输出：  在④模块中，用相关信号展示系统运行状态（LED）、呼梯状态（五个LED），当前楼层以及当前的运行状态（七段数码管）；  其中，对应的运行状态仅显示 上、下、停，对应的符号展示详见实验结果分析部分。 | | | | | | |
| 四、实验过程或算法(关键步骤、核心代码注解等）  一、顶层设计elevator\_design.v  module elevator\_design(  input clk,rst,en,  input [4:0] in,//五位叫楼信号  input [4:0] cancel,//五位取消叫楼信号  output reg en\_led,  output reg [4:0] led,//五位叫楼指示灯  output reg [7:0] a2g,//数字显示使能  output reg [3:0] sel//数字显示选择      );      parameter   stop     = 3'b100,                   upRun    = 3'b001,                   upStop   = 3'b000,                   downRun  = 3'b011,                   downStop = 3'b010;      //第0位-->1:run 0:stop      //第1位-->1:down 0:up      //第2位-->1:top  0:yundong        reg [4:0] floor;      wire [4:0] next\_floor;      reg [2:0] state;      wire [2:0] next\_state;        reg [4:0] floor\_call;//五个楼层的叫梯状态      reg [4:0] temp\_floor\_call;      //时钟 start///////////////////////////////////////////////////      wire clk\_100ms;      parameter clk\_100ms\_counter=49\_9999;      counter#(clk\_100ms\_counter) CLK\_100ms(clk,rst,en,clk\_100ms);      wire clk\_1s;      parameter clk\_1s\_counter=4999\_9999;      counter#(clk\_1s\_counter) CLK\_s(clk,rst,en,clk\_1s);      //时钟 end///////////////////////////////////////////////////          //更新叫梯状态 start///////////////////////////////////////      genvar i;      for(i = 0;i<5;i=i+1)begin          always@(posedge clk\_100ms)begin              if(!rst||!en||((state==upStop||state==downStop)&&floor[i]==1))begin                  //若没开启，或当前已经移动到此楼，取消此楼的叫楼状态                  temp\_floor\_call[i]<=0;              end              else if(cancel[i]==1)begin                  temp\_floor\_call[i]<=0;              end              else if(in[i]==1)begin                  temp\_floor\_call[i]<=1;              end              else begin                  temp\_floor\_call[i]<=temp\_floor\_call[i];              end          end          //利用组合逻辑，使rst和en能够异步处理          always@(\*)begin              if(!rst||!en||((state==upStop||state==downStop)&&floor[i]==1))begin                  floor\_call[i]=0;              end              else floor\_call[i]=temp\_floor\_call[i];          end      end      //更新叫梯状态 end///////////////////////////////////////        //状态和楼层转移 start///////////////////////////////////////      always@(negedge clk\_1s or negedge en) begin          if (!rst || !en) state = stop;          else state = next\_state;      end      always@(negedge clk\_1s or negedge en) begin          if (!rst) floor = 'b00001;          else if (!en) floor = floor;          else floor = next\_floor;      end        mealy\_state\_change mealy\_state\_change\_inst01(      .clk(clk\_1s),      .floor\_call(floor\_call),      .floor(floor),      .state(state),//当前状态      .next\_state(next\_state),      .next\_floor(next\_floor)      );      //状态和楼层转移 end///////////////////////////////////////      //digital\_show模块 start///////////////////////////////////////      wire en\_led2;      wire [4:0] led2;      wire [7:0] a2g2;      wire [3:0] sel2;      digital\_show(      .clk(clk\_100ms),      .en(en),      .floor\_call(floor\_call),      .floor(floor),      .state(state),      .en\_led(en\_led2),      .led(led2),      .a2g(a2g2),      .sel(sel2)      );      always@(\*) begin          en\_led = en\_led2;          led = led2;          a2g = a2g2;          sel = sel2;      end        //digital\_show模块 end///////////////////////////////////////      endmodule  二、时钟分频器counter.v  module counter  #(parameter clk\_counter=499)  (      input clk,rst,en,      output reg CLK      );      integer i;      always@(posedge clk)begin          if(!en||!rst) begin              i<=0;              CLK<=0;          end          else begin              i<=i+1;              if(i==clk\_counter)begin                  CLK=!CLK;                  i<=0;              end          end      end  endmodule  三、将绝对的呼梯信号转换成相对信号(up,down,cur) abs2rela.v  `timescale 1ns / 1ps  //////////////////////////////////////////////////////////////////////////////////  // Company:  // Engineer:  //  // Create Date: 2022/11/25 23:04:17  // Design Name:  // Module Name: abs2rela  // Project Name:  // Target Devices:  // Tool Versions:  // Description:  //  // Dependencies:  //  // Revision:  // Revision 0.01 - File Created  // Additional Comments:  //  //////////////////////////////////////////////////////////////////////////////////  module abs2rela(  input[4:0] floor,  input[4:0] floor\_in,  output reg up,  output reg down      );  always@(\*) begin      case(floor)          'b10000: begin                   up = 0;                   if (floor\_in[3:0] == 'b0000) down=0;                   else down = 1;                   end          'b01000: begin                   if (floor\_in[4]=='b0) up = 0;                   else up = 1;                   if (floor\_in[2:0]=='b000) down = 0;                   else down = 1;                   end          'b00100: begin                   if (floor\_in[4:3]=='b00) up = 0;                   else up = 1;                   if (floor\_in[1:0]=='b00) down = 0;                   else down = 1;                   end          'b00010: begin                   if (floor\_in[4:2]=='b000) up = 0;                   else up = 1;                   if (floor\_in=='b0) down = 0;                   else down = 1;                   end          'b00001: begin                   if (floor\_in[4:1]=='b0000) up = 0;                   else up = 1;                   down = 0;                   end          default: begin                   up = 0;                   down = 0;                   end      endcase  end  endmodule  四、电梯状态变化Mealy状态机 mealyStateChange.v  module mealy\_state\_change(      input clk,      input [4:0] floor\_call,      input [4:0] floor,      input [2:0] state,//当前状态      output reg [2:0] next\_state,      output reg [4:0] next\_floor      );          //通过各楼叫梯型号输出相对叫楼信号 start///////////////////////////////////////      wire up;      wire down;      abs2rela abs2rela\_inst(floor,floor\_call,up,down);      //通过各楼叫梯型号输出相对叫楼信号 end///////////////////////////////////////        //有限状态机设计 start///////////////////////////////////////      parameter   stop     = 3'b100,                   upRun    = 3'b001,                   upStop   = 3'b000,                   downRun  = 3'b011,                   downStop = 3'b010;          always@(posedge clk)begin          case(state)               stop:      begin                          if(up) next\_state <= upRun;                          else if (down) next\_state <= downRun;                          else next\_state <= stop;                          next\_floor <= floor;                          end               upRun:     begin                          //下面这行代码意味着到达了需要上升的楼层                          if ((floor\_call & floor) != 'b00000) begin next\_state <= upStop;next\_floor <= floor; end                          else if (up)  begin next\_state <= upRun; next\_floor <= (floor<<1); end                          else begin next\_state <= stop;next\_floor <= floor; end                          end               upStop:    begin                          if (up) next\_state <= upRun;                          else if (down) next\_state <= downRun;                          else next\_state <= stop;                          next\_floor <= floor;                          end               downRun:   begin                          if ((floor\_call & floor) != 'b00000) begin next\_state <= downStop; next\_floor <= floor; end                          //下面的代码意味着，在向下的过程中，下面依旧还有信号                          else if (down) begin next\_state <= downRun; next\_floor <= (floor>>1); end                          else begin next\_state <= stop; next\_floor <= floor; end                          end               downStop:  begin                          if (down) next\_state <= downRun;                          else if (up) next\_state <= upRun;                          else next\_state <= stop;                          next\_floor <= floor;                          end  //             default:   begin  //                        next\_state<=stop;  //                        next\_floor<=floor;  //                        end            endcase        end      //有限状态机设计 end///////////////////////////////////////  endmodule  五、展示模块 degital\_show.v  `timescale 1ns / 1ps  //////////////////////////////////////////////////////////////////////////////////  module digital\_show(      input clk,en,      input [4:0] floor\_call,      input [4:0] floor,      input [2:0] state,      output en\_led,      output [4:0] led,      output reg [7:0] a2g,      output reg [3:0] sel      );      reg sel\_change;      //右位显示上升下降，左位显示所在楼层      reg [3:0]data;      always@(posedge clk)begin          if(!en) sel\_change<=0;          else sel\_change <= ~sel\_change;      end        always@(\*)begin          if(en) begin              if (sel\_change) begin                  if (floor[4]) data<=4'b0101;                  else if (floor[3]) data<=4'b0100;                  else if (floor[2]) data<=4'b0011;                  else if (floor[1]) data<=4'b0010;                  else if (floor[0]) data<=4'b0001;              end              else begin                  if (state=='b001) data<=4'b1010;                  else if (state=='b011) data<=4'b1111;                  else data<=4'b1110;              end          end      end      assign en\_led=!en;      assign led = en?floor\_call:5'b00000;      always@(posedge clk) begin          if(!en)a2g = 8'b1111\_1111;          else begin              case(data)                  'b0001:a2g<=8'b10011111;                  'b0010:a2g<=8'b00100101;                  'b0011:a2g<=8'b00001101;                  'b0100:a2g<=8'b10011001;                  'b0101:a2g<=8'b01001001;                  'b1010:a2g<=8'b00010001;                  'b1110:a2g<=8'b11111101;                  'b1111:a2g<=8'b10000001;                  default:a2g<=8'b11111111;              endcase          end      end      always @(posedge clk) begin          if (en) begin              if (sel\_change) sel='b1011;              else sel='b1101;          end      end  endmodule  RTL电路图：    根据RTL电路图，可以看出各个模块，设计与初期预计相似。 | | | | | | |
| 五、实验过程中遇到的问题及解决情况(主要问题及解决情况)  1. 电梯交互逻辑较为复杂，主要表现在时序之间的交叉，不同状态之间的转化。  解决情况：多次修改优化设计结构和状态图，理清相互的逻辑关系；  2. 按键取消的功能一直无法实现，尝试通过将信号取反来实现，但设计总出现Bug。  解决情况：同样使用板机调试工具ILA进行Debug，发现cancel信号会直接被顶出，因此我们采取了cancel信号优先级高的解决方式，并且直接采用拨码开关输入。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  按键说明：    说明：其中是否处于工作状态灯的指示为，工作时不亮，非工作时亮，可以当作故障报警灯使用。  演示各个工作模式：   1. 电梯向上走。按下四楼呼楼。   最初电梯停留在1F    4F呼叫。    电梯上行。    到达4F，呼楼灯灭，电梯状态改变。     1. 电梯从5F连续停4F,2F,1F。   电梯停留在5F。    4F，2F，1F呼楼，电梯下行。    到达4F，暂停，4F呼楼灯灭。    到达3F，继续下行。    到达2F，暂停，2F呼楼灯灭。    到达1F，停止，1F呼楼灯灭。     1. 电梯在3F，5F、1F呼楼。   电梯停在3F。    5F，1F呼楼，电梯先上行。  到达5F，暂停，5F呼楼灯灭。    下行，到达1F，停止，1F呼楼灯灭。     1. en关闭。     不显示楼层，电梯停止，系统运行灯（故障灯）亮。  七、小组分工情况说明  罗帆靖：项目思路设计，代码书写，实验报告书写。  张恩宁：项目思路设计，代码书写，实验报告书写。  耿协铭：项目思路设计，代码书写，实验报告书写。 | | | | | | |