**《脉冲电路与数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | **魏和琬 陈炫瑾** | | | **年级** | **2017** |
| **学号** | | **20174393 20174417** | | | **专业、班级** | **信安2班 物联1班** |
| **实验名称** | **摩尔状态机序列检测器** | | | | | |
| **实验时间** | **5.31** | | **实验地点** | **DS1410** | | |
| **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： 设计及过程完整,有结果分析  评价教师签名（电子签名）：谭玉娟 | | | | | | |
| 一、实验目的  通过实验，掌握状态机状态转移原理，并设计代码实现。 | | | | | | |
| 二、实验项目内容  1)设计“1101”序列检测的状态转换图；  2)调用并转串输出模块，使用Verilog HDL语言的行为描述方式实现一个摩尔状态机，能检测一个8位的二进制数据中是否存在“1101”序列，如果检测到该序列则指定的LED灯亮；  3)综合、实现、生成bit流，下载到Nexys4开发板进行验证； | | | | | | |
| 三、实验过程或算法(包括真值表、原理图、关键步骤、核心代码）  **实验原理：**   1. par2ser将并行输入的8位二进制码转串行输出。 2. 逻辑抽象后，根据5个状态画出状态转换表以及状态转换图，其中输入是输入的8位数据转串行的每一个，当出现”1101“，状态S4输出会变为1，即LED灯亮。 3. 按下rst键重置所有数据，按下set键开始并转串并读入第一个数，按下clk键开始进行状态转换。   状态：  S0=‘4’b0000’; S1=’4’b0001’ ;S2=’4’b0010’ ;S3=’4’b0011’ ;S4=‘4’b0100’;  **状态转移表：**  **2019-06-04 14:58:25.834000**  **状态转移图为：**  **2019-06-04 15:07:25.484000**  **代码：**  **Moore状态机：**  module moer(clk,rst,z,a);  input clk,rst,a;  output z;  reg z;  reg[3:0] cur\_sta,net\_sta;  parameter S0 = 4'b0000;  parameter S1 = 4'b0001;  parameter S2 = 4'b0010;  parameter S3 = 4'b0011;  parameter S4 = 4'b0100;    always@(posedge clk or posedge rst)  begin  if(rst)  cur\_sta <= S0;  else  cur\_sta <= net\_sta;  end  always@(cur\_sta or a or rst)  begin  if(rst)  net\_sta = S0;  else  case(cur\_sta)  S0:net\_sta = (a==1)?S1:S0;  S1:net\_sta = (a==1)?S2:S0;  S2:net\_sta = (a==0)?S3:S2;  S3:net\_sta = (a==1)?S4:S0;  S4:net\_sta = (a==1)?S2:S0;  default:net\_sta = S0;  endcase  end  always@(rst or cur\_sta)  begin  if(rst)  z = 0;  else  case(cur\_sta)  S0 : z = 0;  S1 : z = 0;  S2 : z = 0;  S3 : z = 0;  S4 : z = 1;  default : z = 0;  endcase  end  endmodule  **并转串：**  module par2ser(clk, set, d, q );  input clk;  input set;  input [7:0] d;  output reg q;  reg [7:0] data\_buff;  always @(posedge clk or posedge set) begin  if(set) begin  q<=1'b0;  data\_buff<=d;  end  else begin  data\_buff<=data\_buff<<1;  q<=data\_buff[7];  end  end  endmodule  **顶层模块：**  module top(clk,set,rst,d,z);  input set;  input rst;  input clk;  input [7:0]d;  output z;  wire q;  par2ser u(.clk(clk),.set(set),.d(d),.q(q));  moer fsm(.clk(clk),.rst(rst),.a(q),.z(z));  endmodule  **仿真：**  module moer\_sim();  reg set;  reg clk;  reg rst;  reg [7:0]d; //并行输入  wire z;  top u1(.clk(clk),.set(set),.rst(rst),.d(d),.z(z));  initial  begin  d=8'b11011101;set = 1'b1; clk = 1'b0; rst =0;  #10 set = 1'b1;  #10 set = 1'b0;  end  always  begin  #25 clk = ~clk;  end  endmodule  **管脚：**  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[7]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[6]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[5]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[4]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[3]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[2]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[1]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports {d[0]}]  set\_property IOSTANDARD LVCMOS33 [get\_ports clk]  set\_property IOSTANDARD LVCMOS33 [get\_ports set]  set\_property IOSTANDARD LVCMOS33 [get\_ports z]  set\_property IOSTANDARD LVCMOS33 [get\_ports rst]  set\_property PACKAGE\_PIN R13 [get\_ports {d[7]}]  set\_property PACKAGE\_PIN U18 [get\_ports {d[6]}]  set\_property PACKAGE\_PIN T18 [get\_ports {d[5]}]  set\_property PACKAGE\_PIN R17 [get\_ports {d[4]}]  set\_property PACKAGE\_PIN R15 [get\_ports {d[3]}]  set\_property PACKAGE\_PIN M13 [get\_ports {d[2]}]  set\_property PACKAGE\_PIN L16 [get\_ports {d[1]}]  set\_property PACKAGE\_PIN J15 [get\_ports {d[0]}]  set\_property PACKAGE\_PIN P17 [get\_ports clk]  set\_property PACKAGE\_PIN P18 [get\_ports rst]  set\_property PACKAGE\_PIN M17 [get\_ports set]  set\_property PACKAGE\_PIN H17 [get\_ports z] | | | | | | |
| 四、实验结果及分析和（或）源程序调试过程  **仿真：**    输入11011101，亮两次  **下载到开发板验证**  **C:\Users\35331\Documents\Tencent Files\353319310\FileRecv\MobileFile\BC7F4D0872CC7A48433EECBFB734A8AF.png**  **开关拨到11011101，按下rst重置，按下set键后，分别在第四下第八下clk，亮灯。**  **C:\Users\35331\Documents\Tencent Files\353319310\FileRecv\MobileFile\AABFDF36BBC6D92A1D9AB67F44879E5C.png**  **开关拨到01101101，按下rst重置，按下set键后，在第五下clk，亮灯。**  **五、小组分工：**  魏和琬：内容讨论，代码和仿真部分完成。  陈炫瑾：内容讨论，状态转移图表绘制，脚管设置和上板。  六、个人总结：  用到了前面并转串实验的代码，重点是加入了rst键，使得可以无限重置。  先画出状态转化图，可以顺利很多。 | | | | | | |