**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 陈鹏宇，  帕肉合·帕尔哈提 | | **年级** | | 2020级，  2020级 |
| **学号** | | 20204227，  20201641 | | **专业、班级** | | 计算机科学与技术01班，  计算机科学与技术05班 |
| **实验名称** | **实验十五 摩尔状态机序列检测器** | | | | | |
| **实验时间** | **2021/12/9** | | **实验地点** | | **Ds1408** | |
| **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，掌握有限状态机的设计方法，并用摩尔状态机设计实现“1101” 序列检测。 | | | | | | |
| 二、实验项目内容   1. 设计“1101”序列检测的状态转换图; 2. 设计一个 8 位并转串输出模块 par2ser。该器件有 8 位输入 d[7:0]，1 位输出 q，另有一个 clk 端，一个 set 端。set端上升沿将 8位输入锁存到逻辑右移移位寄存器中。 3. 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； 4. 综合、实现、生成bit流，下载到Nexys4开发板进行验证； | | | | | | |
| 1. 实验设计   化简后的状态转换图  1639062138955 | | | | | | |
| 四、实验过程或算法  莫尔型状态机检验1101：  module detector\_1101(clk, reset, in, out);  input clk;  input reset;  input [15:0] in;  output out;    wire out;  reg [2:0] state;  reg check; //检验位  reg [15:0] tmp\_in; //tmp\_in    parameter update\_interval = 50\_000\_000;  integer selcnt;  reg clk100;    parameter idle = 3'b000, s1 = 3'b001, s2 = 3'b010, s3 = 3'b011,//状态编码  s4 = 3'b100, s5 = 3'b101, s6 = 3'b110, s7 = 3'b111;    always @(posedge clk) //时钟分频模块  begin  if (selcnt == update\_interval)  begin  selcnt <= 0;  clk100 <=~clk100;  end  else  selcnt<=selcnt+1;  end    always@(posedge clk100 or posedge reset)//状态机实现  begin  if(reset) //并行载入数据,状态为初始状态  begin  state <= idle;  tmp\_in <= in;  end  else  begin  check = tmp\_in[15]; //将最高位输入到序列检测器中  tmp\_in = {tmp\_in[14 : 0], tmp\_in[15]};  case(state)  idle:  begin  if(check == 1) state <= s1;  else state <= idle;  end  s1:  begin  if(check == 1) state <= s2;  else state <= idle;  end  s2:  begin  if(check == 0) state <= s3;  else state <= s2;  end  s3:  begin  if(check == 1) state <= s4;  else state <= idle;  end  s4:  begin  if(check == 1) state <= s2;  else state <= idle;  end  default: state <= idle;  endcase  end  end  assign out = (state == s4 ) ? 1 : 0; // 莫尔机  endmodule | | | | | | |
| 五、实验过程中遇到的问题及解决情况  若按照老师的源代码，不能从in[15]开始检测，因为源代码是先进行移位再赋值，为了能从最高位开始检测，需要先赋值再移位，所以交换两行代码的位置如下 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  首先通过将reset置为1来读取十六位待检测序列，此后每一秒从最高位到最低为循环检测序列，若到达对应序列状态，则在一个或多个时钟周期内保持q为1。经上板验证后，逻辑功能实现成功。  对于莫尔型和米利型的区别，在完成两个实验后有了更深刻的印象，由于米利型的输出与输入有关，所以米利型最终结果输出只需由倒数第二个状态和输入决定，而莫尔型则必须达到最后一个状态才会输出1，可知米利型可以比莫尔型少用一个状态，且米利型的输出对输入更加敏感。   1. 小组分工情况说明   陈鹏宇：自主完成实验，完成实验报告  帕肉合·帕尔哈提：自主完成实验 | | | | | | |