**《数字逻辑》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | |  |
|  | |  | |  | |  |
| **实验名称** | **实验三．多路选择器** | | | | | |
| **实验时间** | 2023年10月28日 | | **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：  评价教师签名（电子签名）： | | | | | | |
| 1. 实验目的   通过实验，学会多路选择器的功能，以及使用 Block Design 和 verilog HDL 语言设计多路选择器。 | | | | | | |
| 二、实验项目内容  **1.** 分别使用 Block Design 设计方法和Verilog HDL语言编写代码的方式设计一个一位 4 选 1 多路选择器，并通过写仿真文件、看 RTL 电路图、下载到板子验证其正确性。  **2.** 使用 Verilog HDL 语言的行为描述方法设计一个 3 位数据的 4 选 1 多路选择器。  3.比较两种方法在RTL电路图、仿真波形等方面的异同。 | | | | | | |
| 1. 实验设计   1.block design设计：  （1）实验原理：  Block design设计方案使用三个二选一多路选择器拓展实现四选一多路选择器。输入信号为a、b、c、d、s0、s1，输出信号为Y。   1. 原理图：屏幕截图 2023-10-28 092709   图1 Block Design图  （3）真值表   |  |  |  | | --- | --- | --- | | S1（SW15） | S0（SW14） | Y | | 0 | 0 | d(SW2~SW0) | | 0 | 1 | c(SW5~SW3) | | 1 | 0 | b(SW8~SW6) | | 1 | 1 | a(SW11~SW9) |   图2 Block Design 实现的真值表  （4）RTL电路图  屏幕截图 2023-10-28 092739  图3 Block Design 实现的RTL 电路图  2.Verilog HDL 语言实现  （1）实验原理：  四选一多路选择器  组合逻辑表达式：  1d=sw14&sw15&a+sw14&¬sw15&b+¬sw14&sw15&c+¬sw14&¬sw15&d  逻辑语言实现：  通过双目运算符，先判断第一位是否为1，若是，则判断第二位是否为1，这样可选择实现需要的输入a,b,c,d。  屏幕截图 2023-10-28 093508  图4 组合逻辑语言实现4选1多路选择器  （2）真值表（同图2）   1. 电路图：   屏幕截图 2023-10-28 093146  图5 Verilog HDL实现的电路图 | | | | | | |
| 四、实验过程或算法  1.Block Design 设计  (1) block design核心算法：  module m4x1  (Y,  a,  b,  c,  d,  s0,  s1);  output [2:0]Y;  input [2:0]a;  input [2:0]b;  input [2:0]c;  input [2:0]d;  input s0;  input s1;  wire [2:0]a\_1;  wire [2:0]b\_1;  wire [2:0]c\_1;  wire [2:0]d\_1;  wire [2:0]mux2x1\_wrapper\_0\_c;  wire [2:0]mux2x1\_wrapper\_1\_c;  wire [2:0]mux2x1\_wrapper\_2\_c;  wire s0\_1;  wire s1\_1;  assign Y[2:0] = mux2x1\_wrapper\_2\_c;  assign a\_1 = a[2:0];  assign b\_1 = b[2:0];  assign c\_1 = c[2:0];  assign d\_1 = d[2:0];  assign s0\_1 = s0;  assign s1\_1 = s1;  m4x1\_mux2x1\_wrapper\_0\_0 mux2x1\_wrapper\_0  (.a(b\_1),  .b(a\_1),  .c(mux2x1\_wrapper\_0\_c),  .s({s0\_1,s0\_1,s0\_1}));  m4x1\_mux2x1\_wrapper\_1\_0 mux2x1\_wrapper\_1  (.a(d\_1),  .b(c\_1),  .c(mux2x1\_wrapper\_1\_c),  .s({s0\_1,s0\_1,s0\_1}));  m4x1\_mux2x1\_wrapper\_2\_0 mux2x1\_wrapper\_2  (.a(mux2x1\_wrapper\_1\_c),  .b(mux2x1\_wrapper\_0\_c),  .c(mux2x1\_wrapper\_2\_c),  .s({s1\_1,s1\_1,s1\_1}));  endmodule   1. Simulation代码：   module m4x1\_sim();  reg [2:0] a= 001;  reg [2:0] b= 010;  reg [2:0] c= 100;  reg [2:0] d= 000;  reg s0;  reg s1;  wire [2:0]Y;  initial  begin  s0=0;  s1=0;  #100  s0=1;  s1=0;  #100;  s0=0;  s1=1;  #100  s0=1;  s1=1;  #100  $finish();  end  m4x1 u  (.Y(Y),  .a(a),  .b(b),  .c(c),  .d(d),  .s0(s0),  .s1(s1));  endmodule   1. 波形图：IMG_256   图6 行为仿真波形图   1. 查看RTL电路图屏幕截图 2023-10-28 092739 2. Run Synthesis、Run Implementation 3. 配置IO口   屏幕截图 2023-10-28 092818  图7 Block Design 实现配置相关IO口   1. 生成比特流文件 2. 烧录到FPGA板中 3. 在开发板中验证结果   功能正常，结果正确，符合真值表，验证成功  IMG_256  图8 验证结果  2.使用Verilog HDL 语言实现  （1）核心代码：  module mux4x1verilog(  input wire [2:0]a,  input wire [2:0]b,  input wire [2:0]c,  input wire [2:0]d,  input wire sw15,  input wire sw14,  output wire [2:0]ld  );  assign ld=sw14?(sw15?a:b):(sw15?c:d);  endmodule   1. simulation 代码：   module mux4x1\_sim(  );  reg [2:0] a = 001;  reg [2:0] b = 010;  reg [2:0] c = 100;  reg [2:0] d = 000;  reg sw15;  reg sw14;  wire [2:0] OUT;  initial  begin  sw15<=0;  sw14<=0;  #100  sw15<=1;  sw14<=0;  #100  sw15<=0;  sw14<=1;  #100  sw15<=1;  sw14<=1;  #100  $finish();  end  mux4x1verilog test(  .a(a),  .b(b),  .c(c),  .d(d),  .sw15(sw15),  .sw14(sw14),  .ld(OUT)  );  endmodule   1. 波形图   屏幕截图 2023-10-28 092646  图 9 Verilog HDL 实现行为仿真图   1. RTL分析：   屏幕截图 2023-10-28 093221  图 10 Verilog HDL语言实现RTL电路图   1. Run Synthesis,Run Implementation 2. 配置IO口   屏幕截图 2023-10-28 093337   1. 生成比特流文件 2. 烧录到FPGA板中 3. 上板验证   功能正常，结果正确，符合真值表，验证成功  IMG_256  图11 Verilog HDL 实现上板验证结果 | | | | | | |
| 1. 实验过程中遇到的问题及解决情况   1.block design生成仿真波形图时呈现的是十进制结果而非二进制  解决情况：咨询老师同学，发现二进制结果被折叠了，在波形图中点击展开就可以看见对应二进制的结果  2.vivado语言为全英文，出现报错时阅读困难  解决情况：使用翻译软件翻译整句，或者查找专业术语逐字翻译。  3.IP封装后调用出错不知道出错原因，且改动IP核内容却不知道如何更新IP核  解决情况：封装IP核前一定要先验证其正确性，逐步排错。查阅网上资料或询问同学得以解决。  4.偶然出现过闪退问题  解决情况：一般都是重新打开软件后就不会再出现闪退  5.配置IO口后生成比特流文件总是出现报错  解决情况：选择合适匹配的IO电平标准和管脚电压  6.不熟悉verilog HDL语法编写代码时费力且总是报错  解决情况：咨询同学，以及查阅资料 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  1.实验结果：  用block design 和Verilog HDL 语言成功在开发板上实现四选一多路选择器功能：IMG_256  2.得到两种方法下的RTL电路图、仿真波形图，对其分析异同：  （1）仿真波形图  二者得到的波形图一致，为：  屏幕截图 2023-10-28 092646   1. RTL电路图：   Block Design 实现的RTL电路分为四级，Verilog HDL实现的RTL电路分为两级，显然后者效果更佳，运行速度更快。  屏幕截图 2023-10-28 093221  图12 Verilog HDL 语言实现RTL电路图  屏幕截图 2023-10-28 092739  图13 Block Design 实现的RTL电路图  七、小组分工情况说明  ：主要负责Verilog和block design两种方式实现四选一多路选择器、仿真波形、RTL电路图观测、上版验证，大部分实验报告编写；  ：主要负责辅助编写实验报告，尝试实现verilog和block design 两种方式实现四选一多路选择器； | | | | | | |