**《数字逻辑》实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | **潘兴烨** | | **年级** | | **2022级** |
| **学号** | **20220589** | | **专业、班级** | | **计算机科学与技术 卓越01** |
| **实验名称** | **实验三 多路选择器** | | | | |
| **实验时间** | **2023.10.12** | **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）： | | | | | |
| 一、实验目的  1.掌握门电路的设计开发。  2.学习掌握IP核的封装与调用。  3. 调用封装的IP核，用Block Design设计一个三位的4选1多路选择器。  4.熟悉仿真文件的写法，并熟悉将程序写入开发板的流程。 | | | | | |
| 二、实验项目内容  设计一个三位4选1多路选择器电路，分别用Block Design和verilog HDL语言编写代码两种方式实现，通过仿真、看RTL电路图、下载到板子验证其正确性，并比较不同方法实现的异同（逻辑资源、RTL电路图、仿真波形等方面）。 | | | | | |
| 三、实验设计  6617b0ee1d4ad20c4b3a6469902b010  实验原理图。  实验原理是，利用2个与2选1多路选择器进行综合，用两个输入信号控制输入的情况。  但是考虑到我们需要做出三位数据的多路选择器，所以我们需要控制w信号和f信号为3位。  根据    逻辑函数，w若为3为，那么为了完成正确的位运算，s必须也为3位。考虑到开发板案件数量。我们仍然输入1位的s0和s1,但是在输入之后利用译码器拓展位宽，实现逻辑功能即可。  真值表如图： | | | | | |
| 四、实验过程或算法  Block design语言编写：首先实现2选1的设计与IP封装，注意实现选择器的位数即可。然后编写转换器，将1位拓展为3位，只需1到111，0到000就可以。  然后建立block design项目，导入并调用IP核，画出电路图。  屏幕截图 2023-10-12 160133  进行RTL分析，观察结果与自己所连电路的差异。  屏幕截图 2023-10-12 160324  检查无误后，编写仿真文件，看是否与真值表相对应。  module Mux4x13\_tb();  reg [2:0]w0;  reg [2:0]w1;  reg [2:0]w2;  reg [2:0]w3;  reg s0;  reg s1;  wire [2:0]re;  Mux4x13\_wrapper Mux4x13\_wrapper(.w0(w0),.w1(w1),.w2(w2),.w3(w3),.s0(s0),.s1(s1),.re(re));  initial  begin  w0 = 001;w1 = 010;w2 = 100; w3=111;s0=0;s1=0;  #10 s0=1;s1=0;  #10 s0=0;s1=1;  #10 s0=1;s1=1;  #10  $finish;  end  endmodule  以下为仿真图：  屏幕截图 2023-10-12 160548  之后连接开发板，并综合程序原文件，输入引脚并产生约束文件。  屏幕截图 2023-10-12 161012  确认后，保存位xdc文件。之后生成比特流。接着打开硬件程序并打开一个新的硬件目标。然后向开发板中编入文件即可。在开发板上验证自己的逻辑与程序是否正确。    1  输入w0=001,w1=010,w2=100,w3=111  当s1=0,s0=0时，选择的是w0，w0=001，所以re输出001  2  s1=0,s1=1,re=w1=010  3  s1=1,s0=0,re=w2=100  4  s1=1,s0=1,re=w3=111  5  经检验，与真值表和仿真生成图相符。  Verilog语言程序设计：  利用了case语句实现选择器功能  module fourone(w0,w1,w2,w3,t,e,f);  input [2:0]w0;  input [2:0]w1;  input [2:0]w2;  input [2:0]w3;  input [1:0]t;  input e;  output reg[2:0]f;  always@(w0,w1,w2,w3,e,t,f)  begin  if(e==0)  f = 0;  else  begin  case(t)  2'b00:f=w0;  2'b01:f=w1;  2'b10:f=w2;  2'b11:f=w3;  endcase  end  end  endmodule  对fourone模块进行RTL分析，生成Schematic：  RTL电路图如下：  屏幕截图 2023-10-12 162706  编写仿真文件：  module fourone\_tb();  reg [1:0]t;  reg [2:0]w0;  reg [2:0]w1;  reg [2:0]w2;  reg [2:0]w3;  wire [2:0]f;  reg e;  fourone fourone(.t(t),.w0(w0),.w1(w1),.w2(w2),.w3(w3),.e(e),.f(f));  initial  begin  w0 = 000;  w1 = 010;  w2 = 100;  w3 = 111;  e = 1;  t = 2'b00;  #10 t = 2'b01;  #10t = 2'b10;  #10 t = 2'b11;  #10e = 0;  #10  $finish;  end  Endmodule  时序图如下：  屏幕截图 2023-10-12 163623  综合源文件，在开发板上验证正确性。按键位置与block design一致。 | | | | | |
| 五、实验过程中遇到的问题及解决情况  使用block design设计mux4x1的过程中，一开始将2选一多路选择器的位宽锁死在了1位，在后续的bd设计中无法灵活的调用，所以后面采用位宽不定的方法解决了问题 | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  Block design设计：仿真测试和开发板验证  屏幕截图 2023-10-12 160548  12345  Verilog 设计：仿真测试和开发板验证：  屏幕截图 2023-10-12 163623  微信图片_20231012165352微信图片_20231012165354微信图片_20231012165356微信图片_20231012165357微信图片_20231012165359  经过电路分析级及真值表情况，判断知结果正确。  逻辑资源对比：  Block design实现：  屏幕截图 2023-10-12 164534屏幕截图 2023-10-12 164906  屏幕截图 2023-10-12 201635  Verilog实现：  屏幕截图 2023-10-12 164644  屏幕截图 2023-10-12 164912  屏幕截图 2023-10-12 202010  可见block design会比Verilog实现消耗更多的逻辑资源。这在情理之中，因为block design在某种程度上是未经过简化、优化的逻辑表达式，而纯Verilog在生成RTL的过程中会有优化逻辑的过程。纯粹的block design因为完全是由ip核搭建所以某种意义上类似ASSP的设计方式，没有将FPGA上的逻辑资源利用效率最大化。不同的Verilog语言编程也会有不一样的资源消耗。  RTL异同  Block design：  屏幕截图 2023-10-12 160324  Verilog:  屏幕截图 2023-10-12 162706  可以看到block design的RTL电路图就是一开始设计是绘制的逻辑电路图形，而verilog则直接生成了一个mux。Mux是由case语句生成的，如果使用if-else语句则电路又会有不同。  七、小组分工情况说明  由潘兴烨完成block design  由张斯宇和曹泽阳完成verilog HDL语言编写和数据汇总 | | | | | |