

基于课本 2.2 节内容的实验报告

基于课本第 2.2 节，本次实验的过程中将完成 MY_AND2 和 MY_OR2 模块的 VHDL 描述程序，并利用这两个模块完成顶层模块 AND_OR 的结构化描述

一、实验目标

通过本次实验学习以下几点：

- 编写简单门电路的 RTL 级描述程序
- 创建简单电路的结构级 VHDL 描述程序
- 用 VHDL 语言建立电路的层级描述
- 熟悉并使用 ISE 集成环境和其 HDL 编辑器
- 熟悉 ISE 第三方集成的仿真工具 Modelsim SE 的使用
- 熟悉一般项目开发的步骤

二、实验过程

1) 创建工程

Top-Level Source Type 选择 HDL

Family:表示 FPGA 器件属于 Xilinx 的哪一个产品系列，选择 Spartan3

Device:表示 FPGA 器件的器件类型，选择 XC3S200

Package:表示 FPGA 的封装类型，选择 PQ208

Speed:表示 FPGA 的速度等级，选择-5

Synthesis Tool:为工程选择的综合工具，选择 XST(VHDL/Verilog)

Simulation:为工程仿真所用的仿真工具，选择 Modelsim-SE VHDL

Perferred Language:为工程源文件的语言类型，选择 VHDL

2) 编写 MY_AND2 和 MY_OR2 模块的 RTL 级描述和 AND_OR 的结构级描述

源代码如下

MY_AND2:

//...

```
entity MY_AND2 is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           C : out STD_LOGIC);
end MY_AND2;
architecture Behavioral of MY_AND2 is
begin
    C <= A and B; //实现与逻辑
end Behavioral;
```

MY_OR2:

//...

```
entity MY_OR2 is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           C : out STD_LOGIC);
end MY_OR2;
architecture Behavioral of MY_OR2 is
```

```

begin
C <= A or B; //实现或逻辑
end Behavioral;
AND_OR:
//...
entity AND_OR is
    Port ( INP : in  STD_LOGIC_VECTOR (3 downto 0);
          Z : out  STD_LOGIC);
end AND_OR;
architecture STRUCT of AND_OR is
-- 元件声明
component MY_AND2
    port(A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);
end component;
component MY_OR2
    port(A: in STD_LOGIC;
          B: in STD_LOGIC;
          C: out STD_LOGIC);
end component;
--信号声明
signal SIG1,SIG2:STD_LOGIC;
begin
U0:MY_AND2 port map(A=>INP(0), B=>INP(1),C=>SIG1);
U1:MY_AND2 port map(A=>INP(2), B=>INP(3), C=>SIG2);
U2:MY_OR2 port map(A=>SIG1, B=>SIG2, C=>Z);
end STRUCT;

```

3) 语法检查

在源文件输入完成后，需要进行语法检查，以验证源文件语法的正确性。

通过 Processes Synthesize-XST 里面的 Check Syntax

4) 功能仿真

功能仿真可以对工程设计文件进行初步的功能验证，仿真软件将 HDL 语言转换成抽象的逻辑电路，忽略信号在逻辑器件和传输线上的延迟，并应对各种可能得输入信号仿真得到设计工程的输出信号，检验输出信号是否满足预期要求。

在对工程文件进行仿真之前，首先要为仿真器输入激励源。通过创建仿真平台文件 tb_AND_OR.vhd,作为输入测试激励；

文件内容如下：

```

ENTITY tb_AND_OR IS
END tb_AND_OR;
ARCHITECTURE behavior OF tb_AND_OR IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT AND_OR
    PORT(

```

```

        INP : IN  std_logic_vector(3 downto 0);
        Z : OUT  std_logic
    );
END COMPONENT;
--Inputs
signal INP_in : std_logic_vector(3 downto 0) := (others => '0');
--Outputs
signal Z_out : std_logic;
-- No clocks detected in port list. Replace <clock> below with
-- appropriate port name
--constant <clock>_period : time := 10 ns;
--增加仿真信号
signal A_STIM,B_STIM:STD_LOGIC_VECTOR(1 downto 0);
BEGIN
    -- 接入仿真信号,注意并运算
    INP_in<=A_STIM & B_STIM;
    -- Instantiate the Unit Under Test (UUT), "接口端口定义=》输入或输出外部信
号"
    uut: AND_OR PORT MAP (
        INP => INP_in,
        Z => Z_out
    );
    -- A_STIM 仿真信号时序
    A_STIM <="00","01" after 50 ns,"10" after 100 ns,"11" after 150 ns;
    -- B_STIM 仿真信号时序
    TB:process
    begin
        B_STIM <="11";
        wait for 50 ns;
        B_STIM <="10";
        wait for 50 ns;
        B_STIM <="01";
        wait for 50 ns;
        B_STIM <="00";
        wait for 50 ns;
        B_STIM <="00";
        wait; --will wait forever
    end process;
END

```

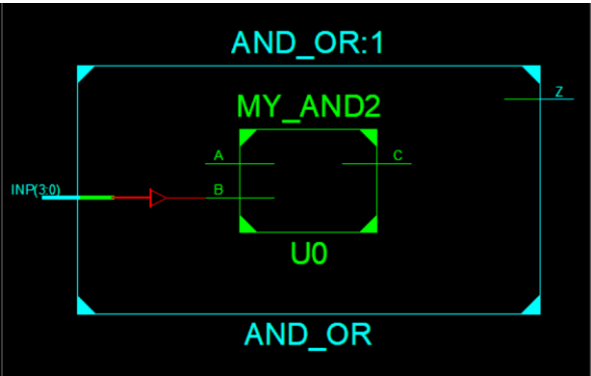
5) 时序仿真

时序仿真是对经过布局布线后的仿真模型加入延时文件进行仿真的过程,它将最基本的门级延时计算在内,模拟工程设计在 FPGA 器件内实现过程。经过时序仿真后的设计基本与实际电路是一致的

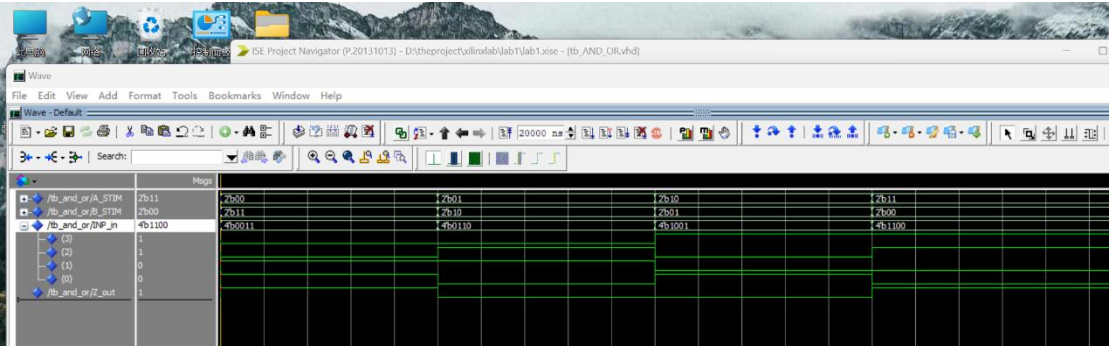
目的是要验证,在信号有延时的情况下,系统是否也能完成给定的功能

三、实验仿真结果图及分析

器件细节：



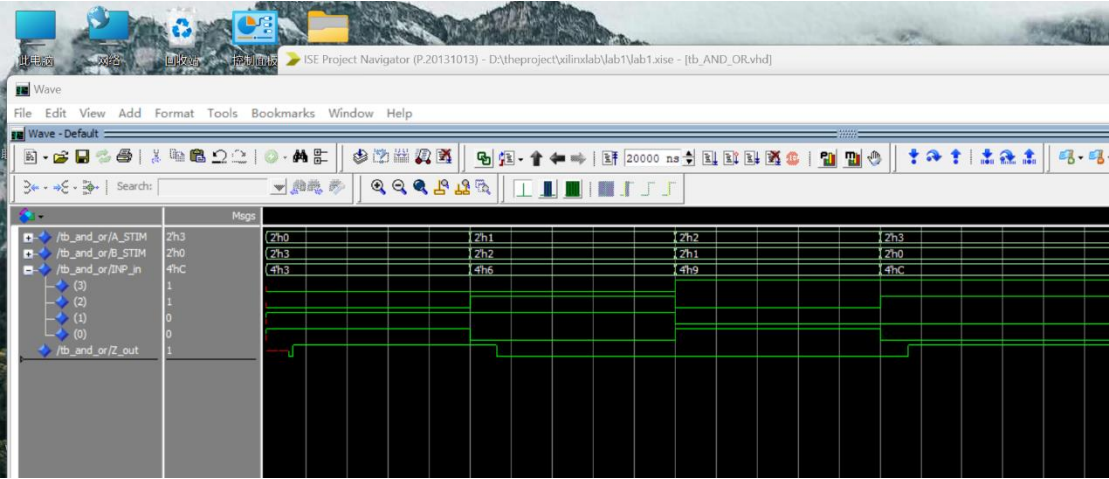
功能仿真

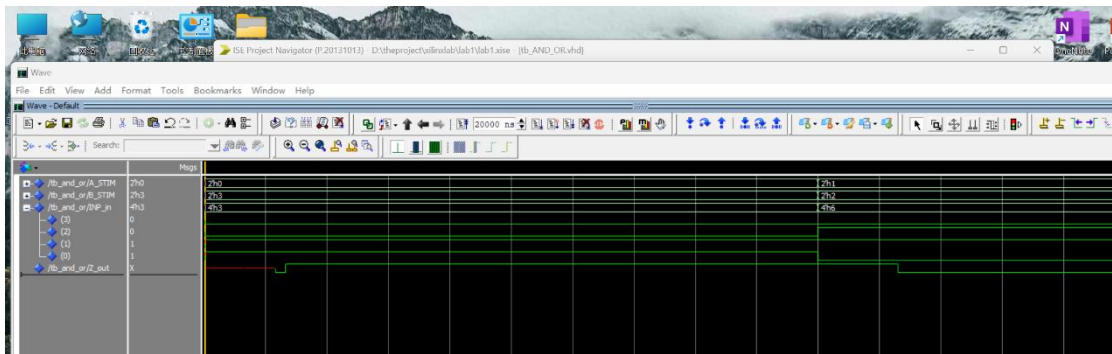


分析：

根据器件 AND_OR 的逻辑：AB 各自先相与的结果在相或，可以通过信号 Z 的输出验证 AND_OR 功能仿真正确

时序仿真





可以看到，大体器件逻辑正确，只不过时序仿真将最基本的门级延时计算在内，模拟了工程设计在 FPGA 器件内实现过程。而在信号有延时的情况下，系统也能完成给定的功能 AND_OR

四、实验经验总结

1. 在编写仿真平台测试文件时，注意各个输入信号的周期合不合适，能否使得仿真得到我们所希望的，对研究有意义的波形
2. 仿真过程中，对文件设定的运行时间决定波形是否完整——在设定的运行时间内程序逻辑可能没有跑完
3. 将仿真文件的视图缩小才能看见完整的波形，不能在小视图下误以为波形没有变化