

**计算机与信息 学院实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 实验课程： | 计算机组成原理实验 | | | | | |
| 实验编号： | Project 1 | | | | | |
| 实验名称： | Xilinx Vivado建模与仿真工具使用 | | | | | |
| 实验人员： | 学号 | 18111303044 | |  | |  |
| 姓名 | 邵一波 | |  | |  |
| 班级 | 计算机类一班 | |  | |  |
| 实验日期： | 2019-9-9 | | | | | |
| 实验室： | 2号实验楼202 | | | | | |
|  |  | | | | | |
| 实验评价： |  | | | | | |
| 实验成绩： | |  | 评价日期： |  | |
|  | 指导教师： | |  | | | |

# Xilinx Vivado建模与仿真工具使用

# 一、实验目的

## 1. 学习Xilinx Vivado软件的使用。

## 2.掌握Xilinx Vivado建模与仿真技术。

## 3.了解Verilog HDL。

# 二、实验工具

## 1.Xilinx Vivado软件。

## 2.Windows系统PC机。

# 三、实验要求

## 1. 安装Xilinx Vivado。

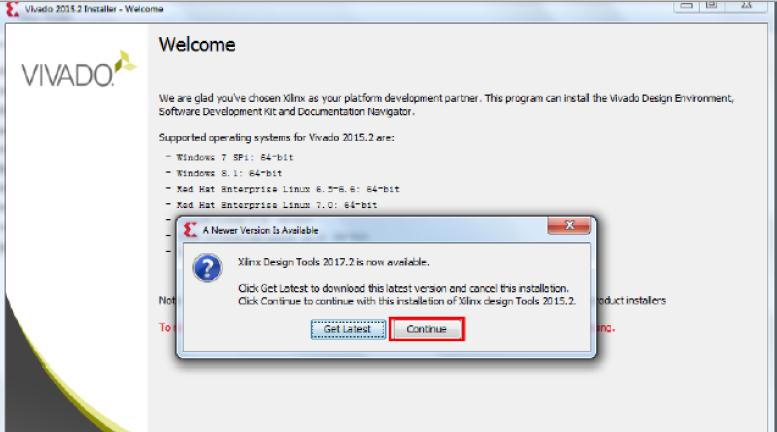
## 2.建立工程，编辑工程文件，进行编译、仿真，查看波形，进行综合，查看RTL视图。记录操作步骤，剪贴操作界面。

# 四、实验内容

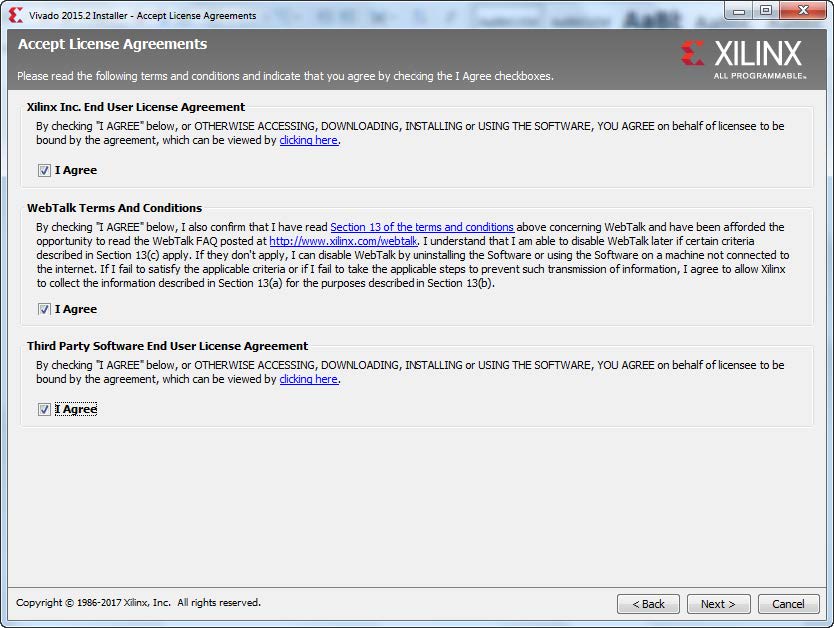
## 1. Xilinx Vivado软件的安装与配置。

【请给出步骤和截图】

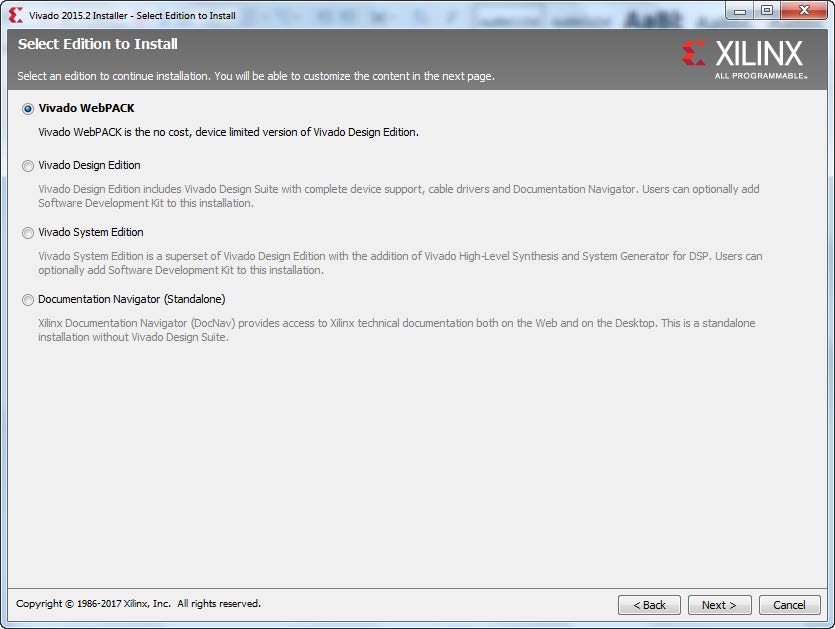
启动安装工具



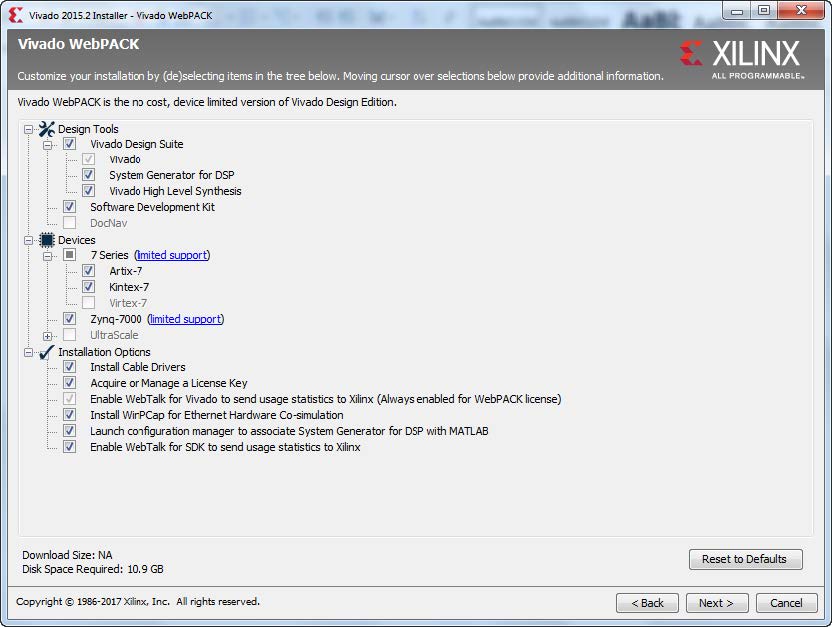
勾选I Agree，点击Next至下一步。



选择Vivado WebPACK，点击Next至下一步。



勾选所有的工具和器件，点击Next继续。



‘

等待安装完成



单击 windows 开始->所有程序->Xilinx Design Tools->Vivado 2015.2->Manage Xilinx Licenses 打开许可证管理器。单击“Load License”,然后单击“Copy License…”



## 2. Xilinx Vivado软件的使用。

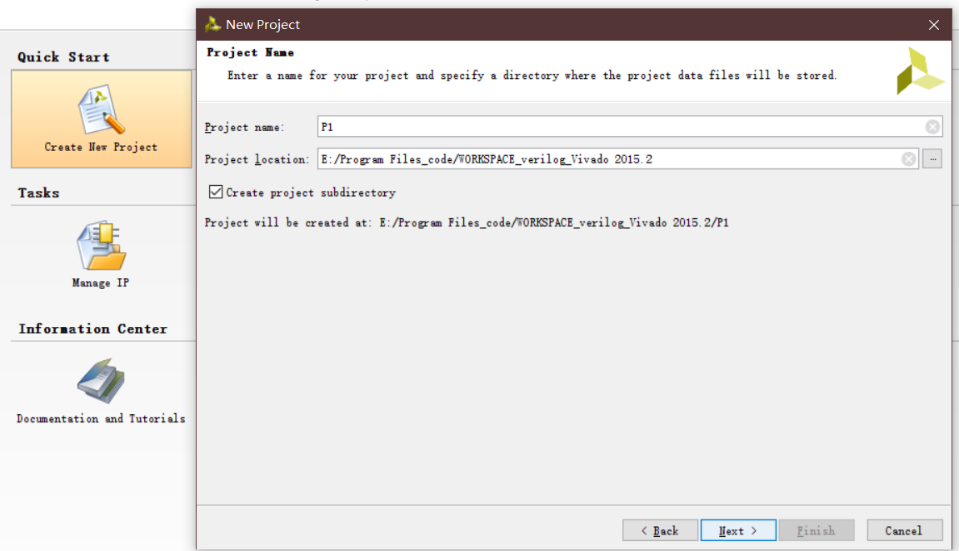
### (1)创建工程

注：工程名P1。

【请给出步骤和截图】

Creat New Project –> next ->命名P1->选择RTL Project并勾选Do not…

->选择 XC7A100TCSG324-1 -> Finish



### (2)工程文件编辑

建立f\_adder.v文件

module f\_adder(ain,bin,cin,cout,sum);

output cout,sum;

input ain,bin,cin;

wire ain,bin,cin,cout,sum;

wire d,e,f;

h\_adder u0(ain,bin,d,e);

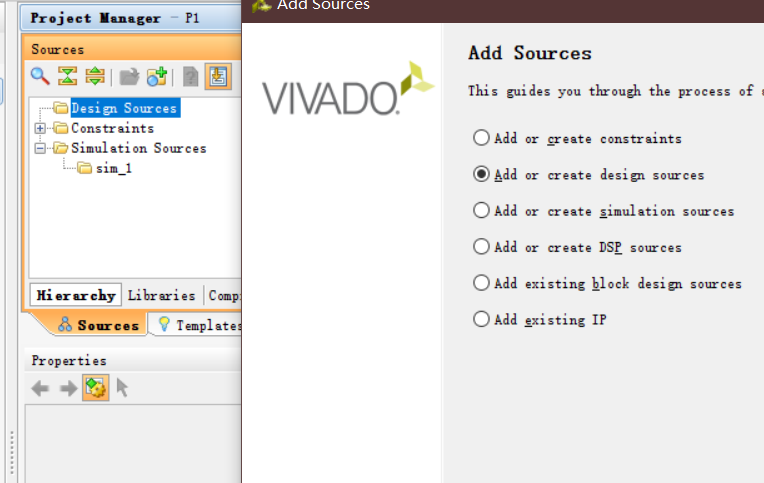
h\_adder u1(e,cin,f,sum);

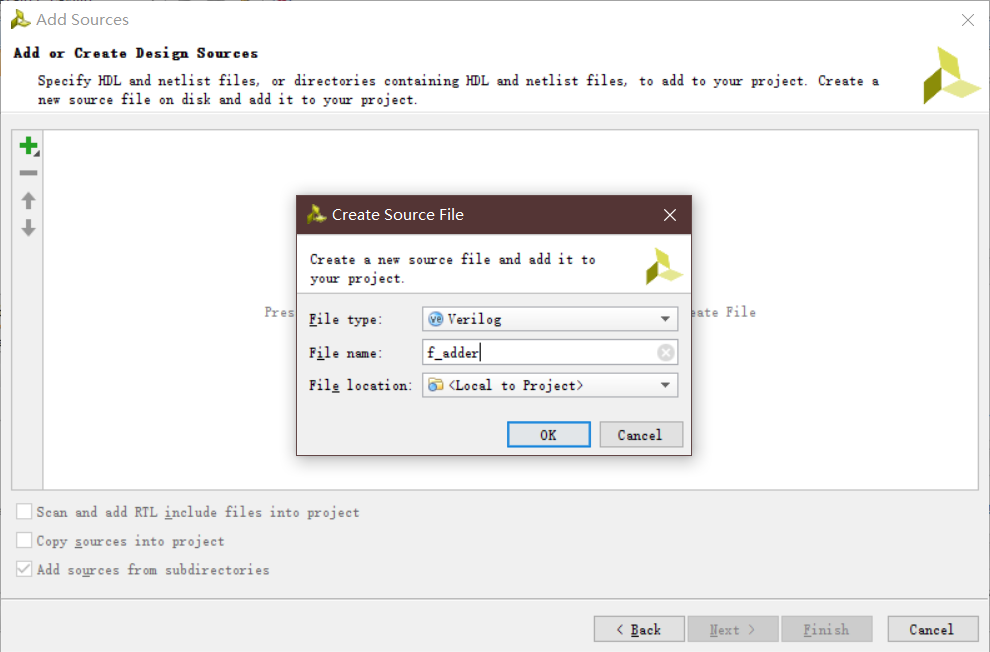
or2a u2(d,f,cout);

endmodule

【请给出步骤和截图】

右键Design Sources -> Add Sources ->命名为f\_adder





建立h\_adder.v文件

/\*以下为半加器模块\*/

module h\_adder(a,b,co,so);

output co,so;

input a,b;

wire a,b,co,so,bbar;

and and2(co,a,b);

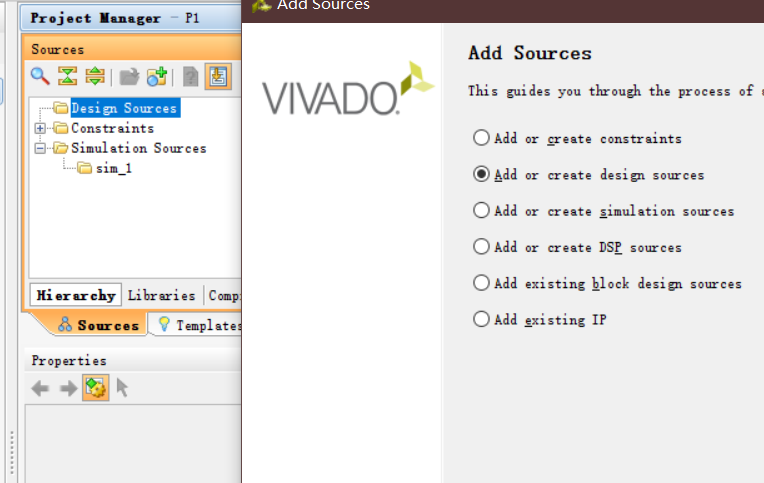
not not1(bbar,b);

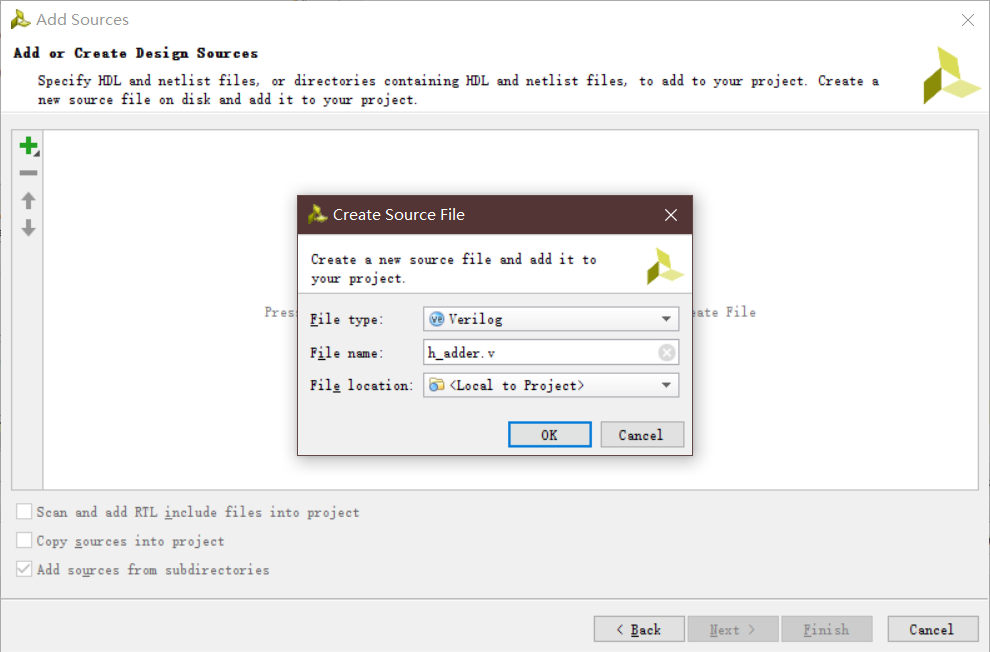
xnor xnor2(so,a,bbar);

endmodule

【请给出步骤和截图】

右键Design Sources -> Add Sources ->命名为h\_adder





建立or2a.v文件

/\*以下为或门模块\*/

module or2a(a,b,c);

output c;

input a,b;

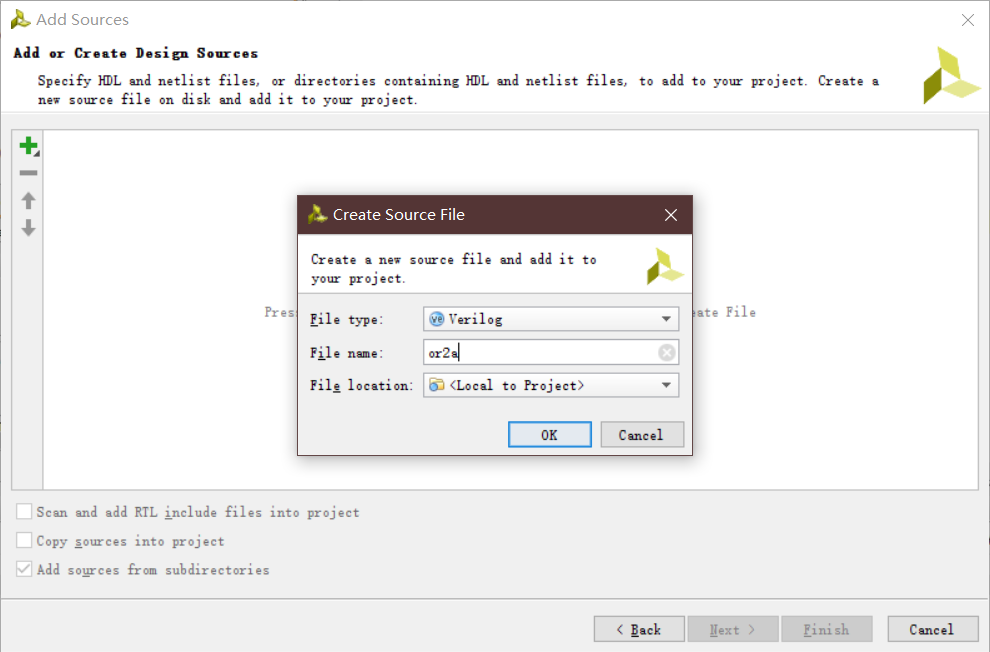
wire a,b,c;

assign c=a|b;

endmodule

【请给出步骤和截图】

右键Design Sources -> Add Sources ->命名为or2a



建立tb\_f\_adder.v文件

`timescale 1ns/100ps

module tb\_f\_adder r();

reg a,b,cin;

wire cout,sum;

f\_adder u0(a,b,cin,cout,sum);

initial

begin

#100; a=1’b0;b=1’b0;cin=1’b0;

#100; a=1’b0;b=1’b1;cin=1’b0;

#100; a=1’b1;b=1’b0;cin=1’b0;

#100; a=1’b1;b=1’b1;cin=1’b0;

#100; a=1’b0;b=1’b0;cin=1’b1;

#100; a=1’b0;b=1’b1;cin=1’b1;

#100; a=1’b1;b=1’b0;cin=1’b1;

#100; a=1’b1;b=1’b1;cin=1’b1;

end

endmodule

【请给出步骤和截图】

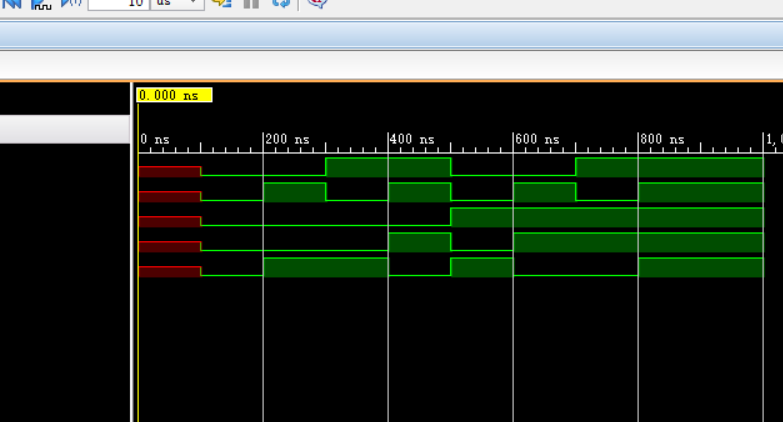
右键Simulation Sources -> Add Sources ->命名为tb\_f\_adder



### (3)仿真，查看波形

【请给出步骤和截图】

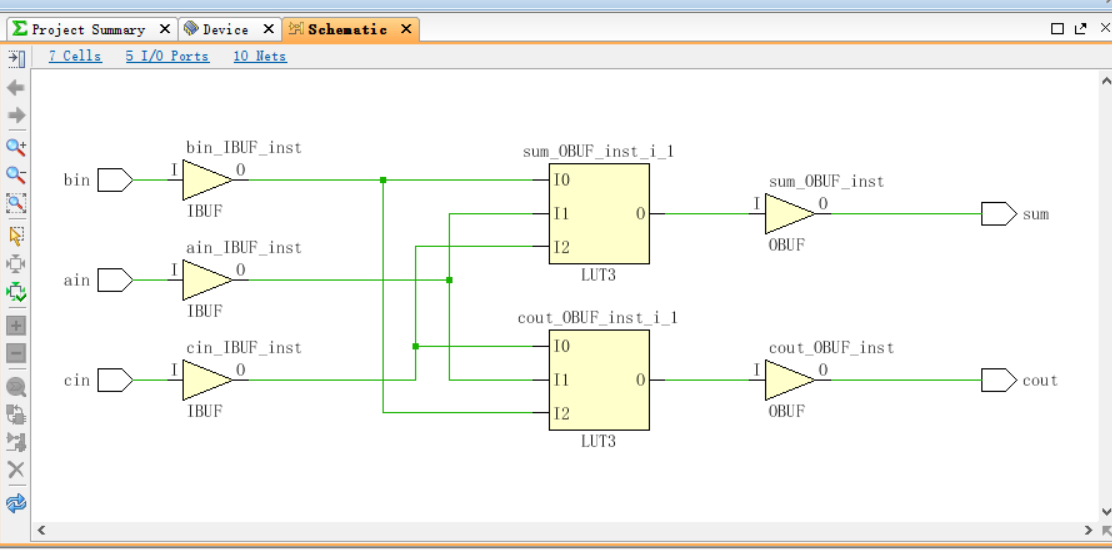
Project Manager -> Simulation -> Run Simulation



### (4)综合，查看RTL视图

【请给出步骤和截图】

Project Manager -> Synthesis -> Run Synthesis -> Open Synthesiszed… -> Schematic



# 五、实验思考

## 1.什么是Verilog HDL?

【答】 Verilog HDL是一种以文本形式来描述数字系统硬件的结构和行为的硬件描述语言

## 2.到Xilinx公司的主页上查看Xilinx的其它支持Verilog HDL的工具，查找其获取方法。

【答】 ISE Design Suite

## 3.到Altera公司的主页上查看支持Verilog HDL的工具，查找其获取方法。

【答】 Quartus

## 4.Baidu搜索支持Verilog HDL的工具，查找其获取方法。

【答】 ISE Design Suite ，Quartus，ModelSim，Synplify