

# **Xilinx ISE 14.7**

## **开发环境使用说明**

**2024.09.25**

# 目 录

1. Xilinx ISE 14.7 开发环境安装.....	3
1.1 系统需求.....	3
1.2 软件安装.....	3
1.3 软件注册.....	11
1.4 bug 修复.....	13
2. Xilinx ISE 14.7 开发环境使用简介.....	15
2.1 界面说明.....	15
2.2 新建工程.....	16
2.3 添加模块.....	20
2.4 模块编辑.....	24
2.5 模块仿真.....	27
2.6 仿真结果分析.....	30
2.6.1 仿真结果界面.....	30
2.6.2 观察与调整仿真波形图.....	31
2.6.3 修改显示数值进制.....	32
2.6.4 观察中间信号.....	33
2.6.5 观察存储器变量.....	34

# 1. Xilinx ISE 14.7 开发环境安装

## 1.1 系统需求

Xilinx ISE 14.7 开发环境经测试可以在 Windows 10 操作系统下顺利运行，在 Windows 11 操作系统下无法安装。

## 1.2 软件安装

1) 安装压缩包大致为 6.15GB，安装占用空间。首先将安装包解压，展开文件目录，如图 1-1 所示。

名称	修改日期	类型	大小
Digilent	2017/5/5 13:35	文件夹	
Digilent Adept	2017/5/5 13:35	文件夹	
iso	2018/11/28 19:22	文件夹	

图 1-1 安装包压缩文件展开目录

2) 进入 “iso” 文件夹，双击 “xsetup.exe” 文件进行安装，如图 1-2 所示。

名称	修改日期	类型	大小
.install	2017/5/5 13:35	文件夹	
bin	2017/5/5 13:35	文件夹	
common	2017/5/5 13:35	文件夹	
Crack	2017/5/5 13:35	文件夹	
data	2017/5/5 13:35	文件夹	
edk	2017/5/5 13:35	文件夹	
idata	2017/5/5 13:38	文件夹	
ise	2017/5/5 13:38	文件夹	
labtools	2017/5/5 13:38	文件夹	
Microsoft.VC90.CRT	2017/5/5 13:38	文件夹	
Microsoft.VC90.MFC	2017/5/5 13:38	文件夹	
msg	2017/5/5 13:38	文件夹	
-p	2015/11/23 10:20	文件夹	
planahead	2017/5/5 13:38	文件夹	
planahead_wp	2017/5/5 13:38	文件夹	
sysgen	2017/5/5 13:38	文件夹	
webpack	2017/5/5 13:38	文件夹	
autorun.inf	2013/10/14 2:47	安装信息	1 KB
xinfo.exe	2013/10/14 2:50	应用程序	741 KB
xsetup.exe	2013/10/14 2:50	应用程序	748 KB

图 1-2 打开安装文件进行安装

3) 安装程序运行后弹出欢迎界面，选择 “Next>” 进入下一步即可，如图 1-3 所示。

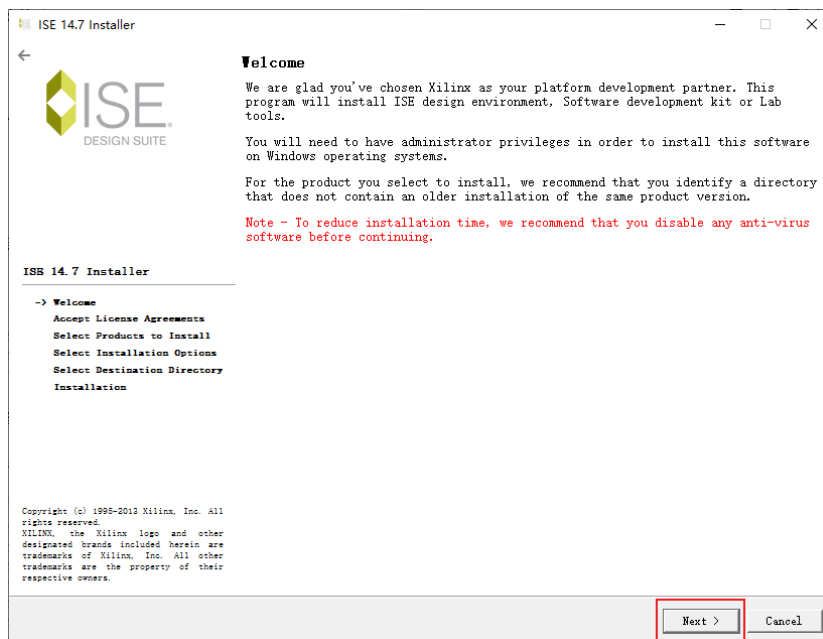


图 1-3 安装程序运行弹出欢迎页面

4) 接收许可协议页面，这里需要将图中红框中所示的勾选框选中，然后点击“Next>”进入下一步，如图 1-4 所示。

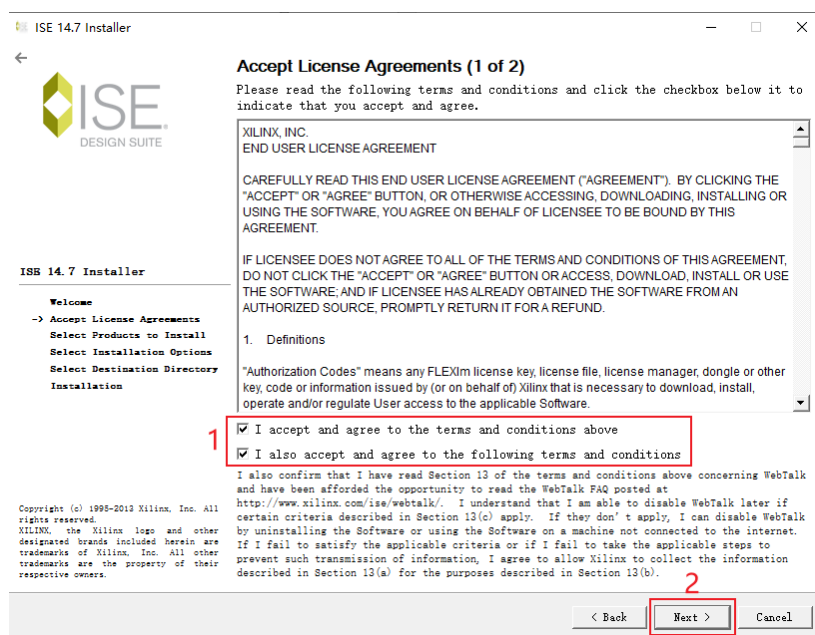


图 1-4 同意安装许可协议 1

5) 继续点击红框中的勾选框并点击“Next>”，如图 1-5 所示。

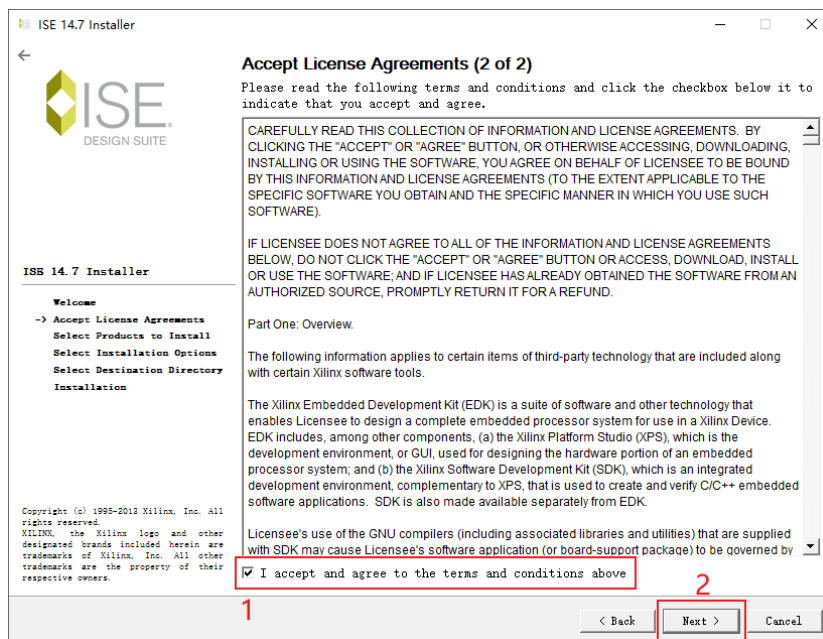


图 1-5 同意安装许可协议 2

6) 选择安装产品页面，这里保持默认选择，直接点击“Next >”，如图 1-6 所示。

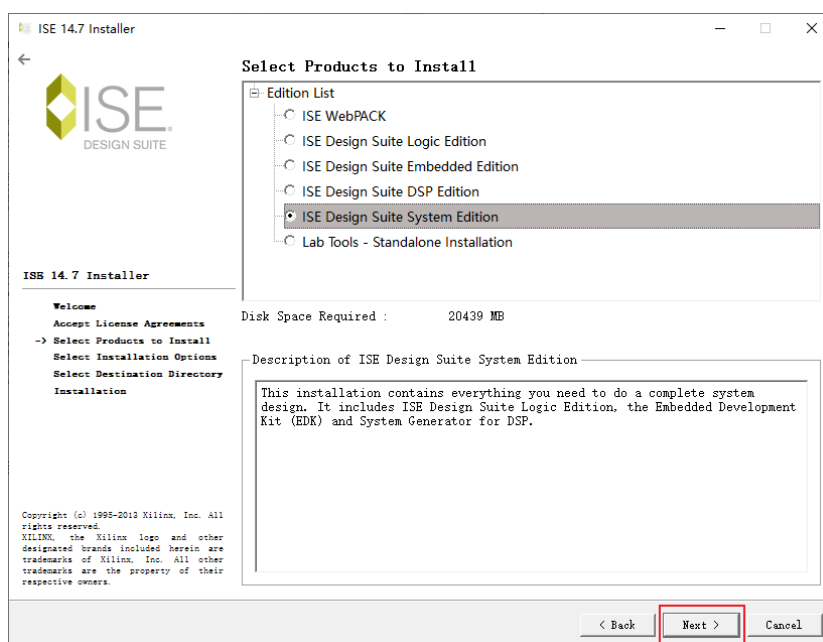


图 1-6 选择安装产品

7) 安装选项页面，这里保持默认选择，直接点击“Next>”，如图 1-6 所示。

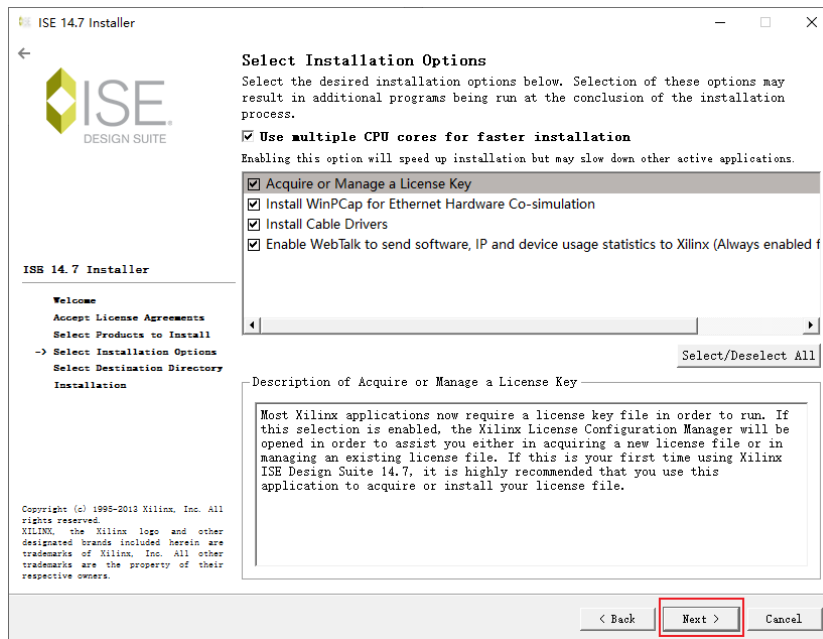


图 1-7 安装选项

8) 选择安装目录页面，在文本框 1 中显示了当前安装目录，可以自行更改安装目录，但需**注意安装路径当中不得含有中文字符**！其余选项保持默认选项即可，点击“Next>”，如图 1-8 所示。

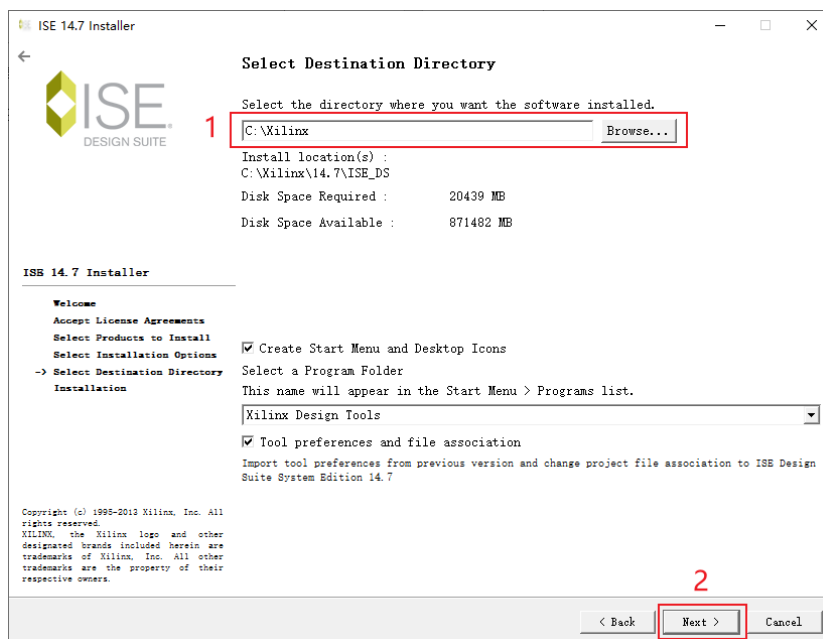


图 1-8 选择安装路径

9) 安装选项概览页面，确认无误后，点击“Install”按钮开始安装，如图 1-9 所示。

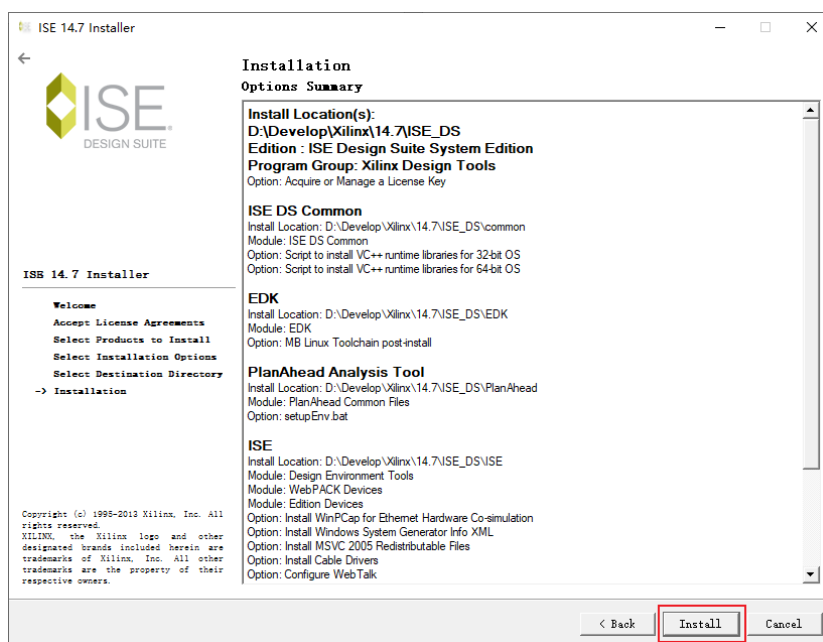


图 1-9 安装选项概览

10) 开始进行安装，等待进度条，如图 1-10 所示。

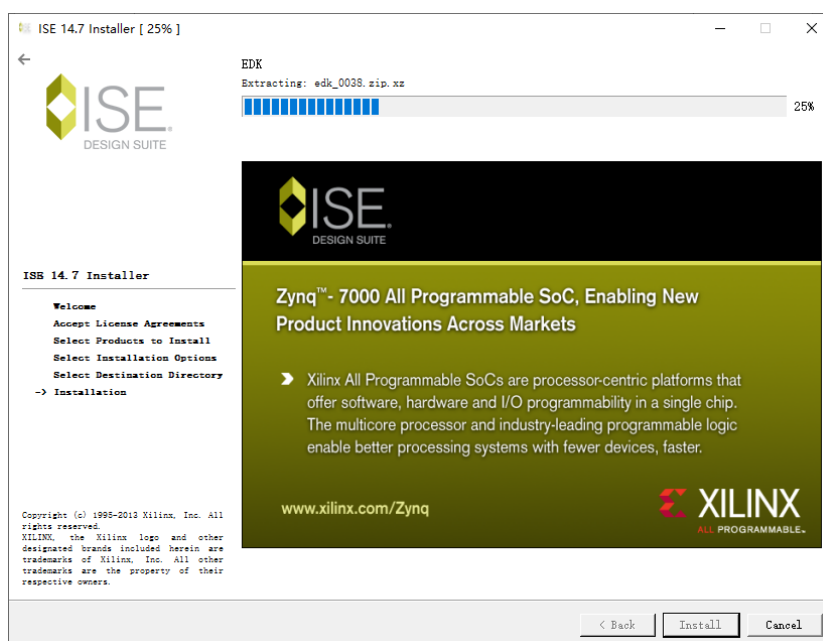


图 1-10 等待安装过程

11) 安装过程中会弹出辅助软件“WinPcap 4.1.1”的安装页面，点击红框按钮进行默认安装即可，如图 1-11、1-12、1-13、1-14、1-15 所示。

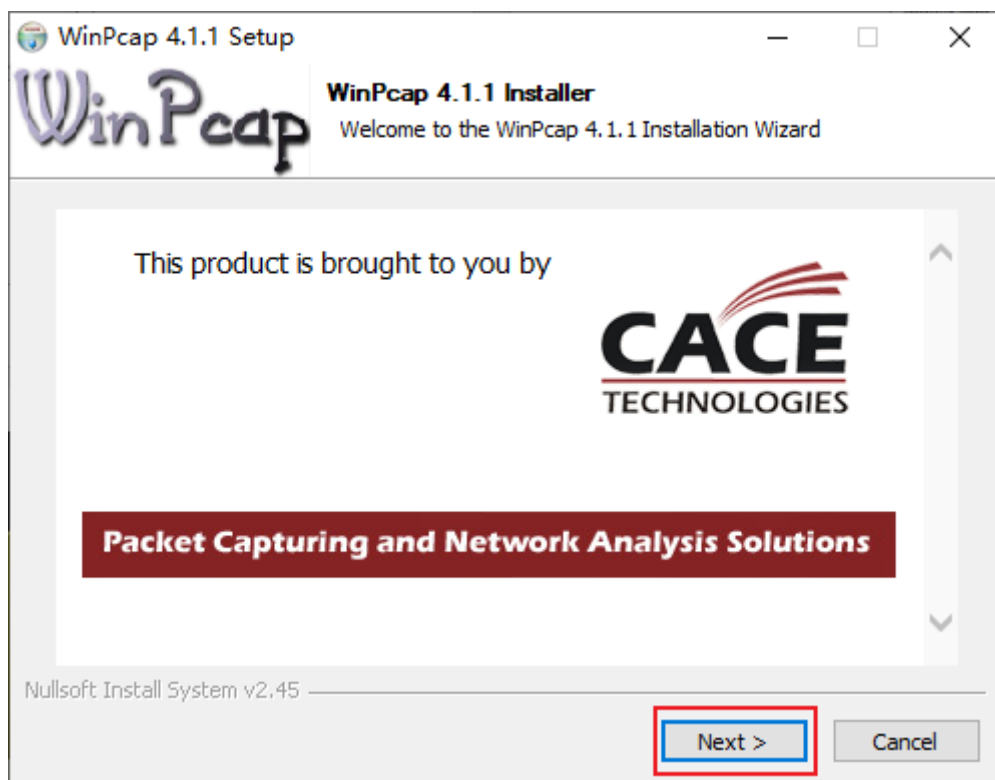


图 1-11 安装辅助软件“WinPcap”-1

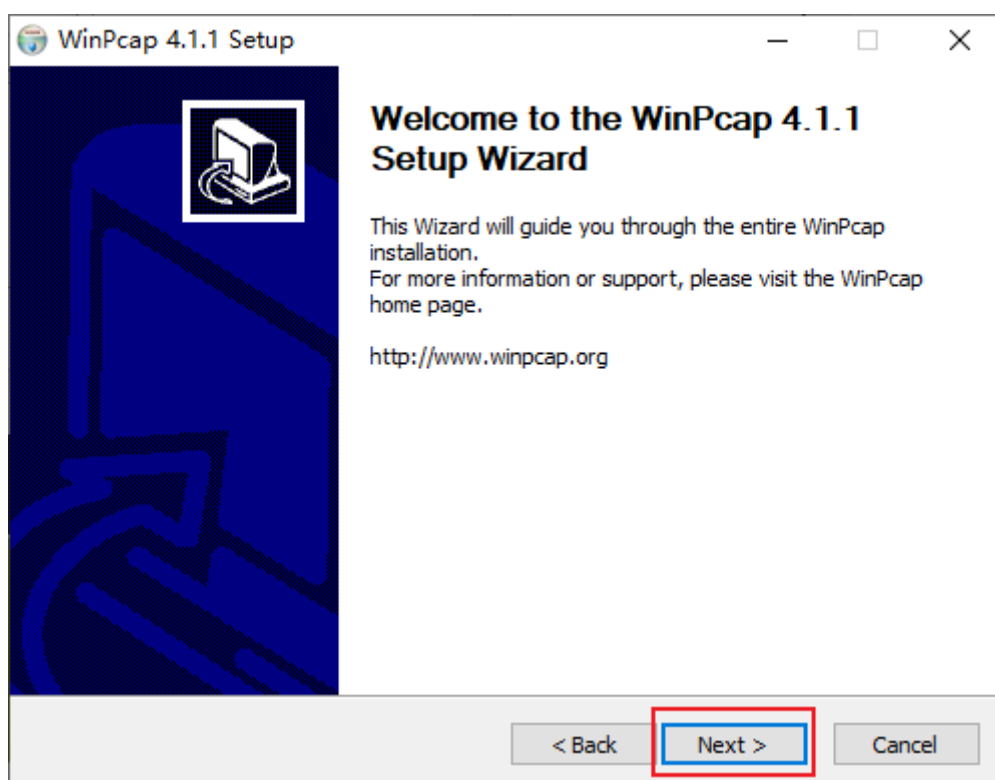


图 1-12 安装辅助软件“WinPcap”-2



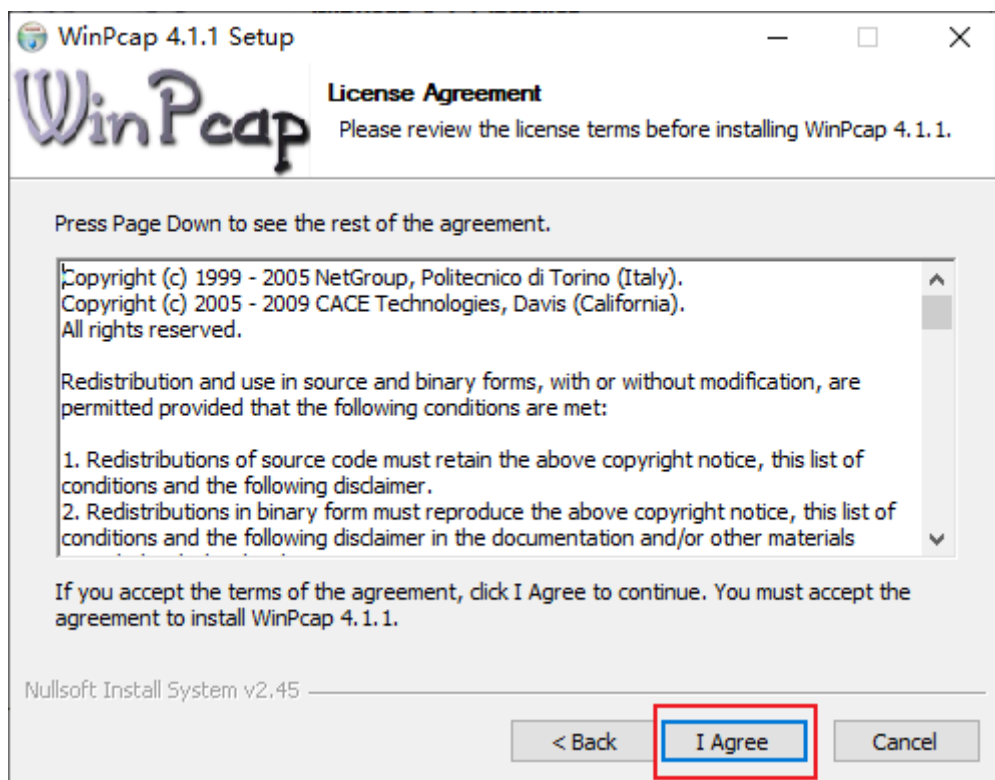


图 1-13 安装辅助软件“WinPcap”-3

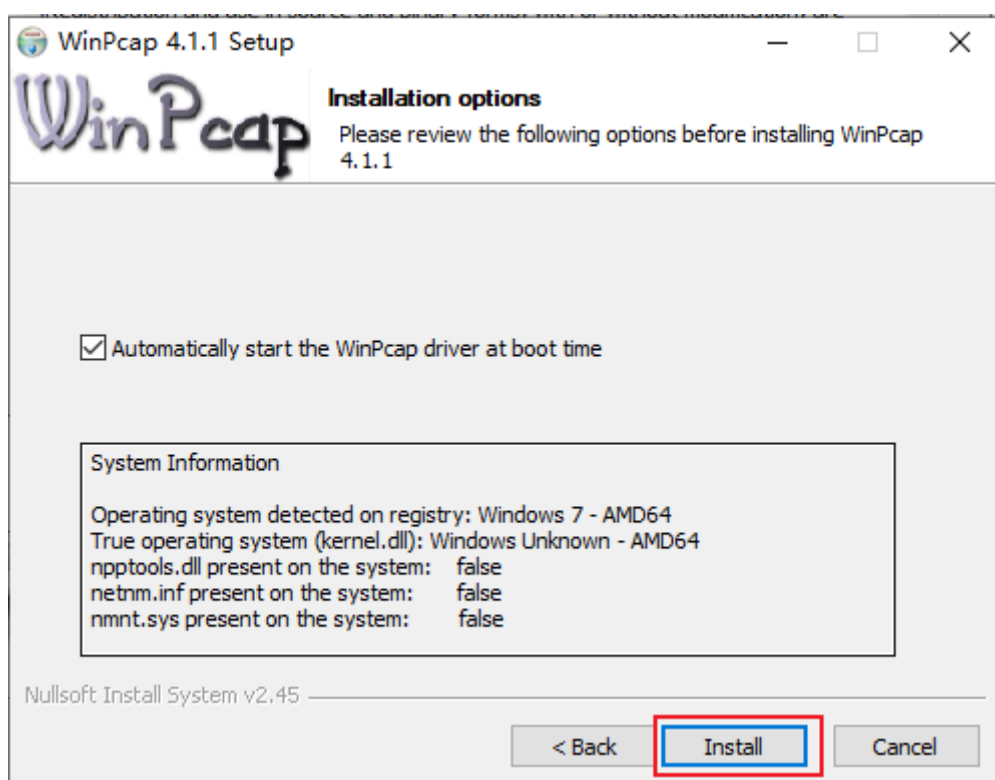


图 1-14 安装辅助软件“WinPcap”-4

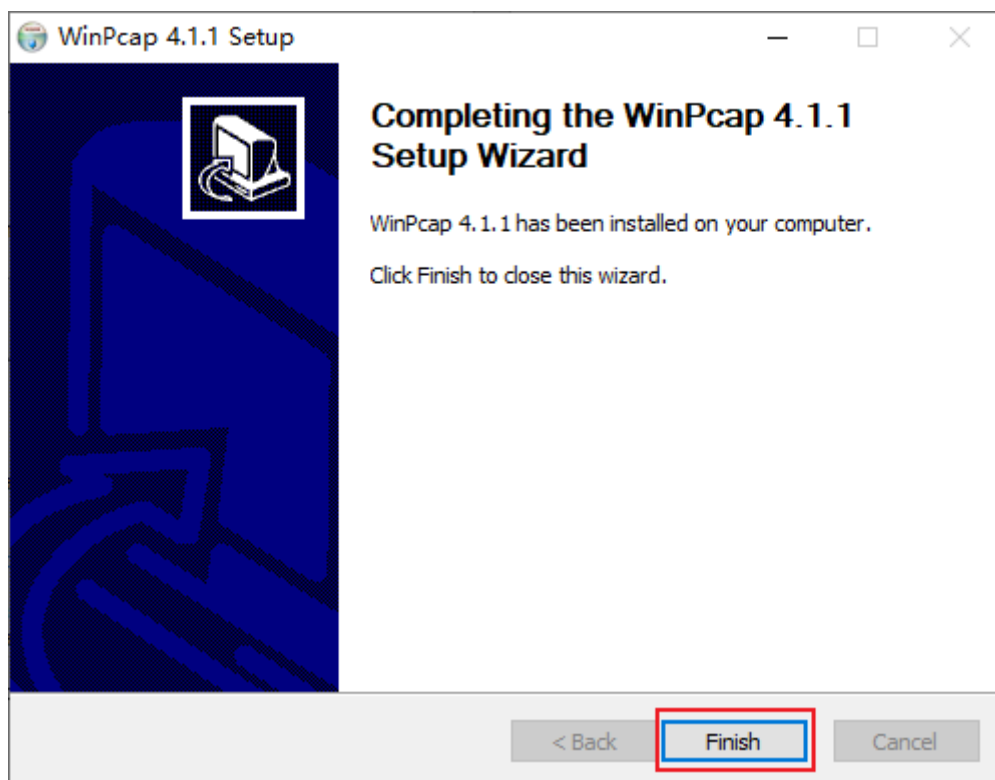


图 1-15 安装辅助软件“WinPcap”-5

12) 随后会弹出断开已连接设备的对话框，点击“确定”后继续。如图 1-16 所示。

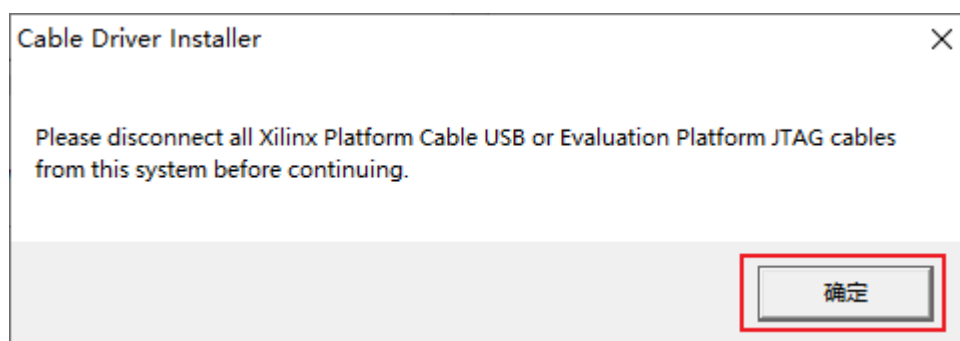


图 1-16 断开连接提示对话框

13) 弹出 MATLAB 定位对话框，可以与 MATLAB 协同工作，点击“Ok”跳过。如图 1-17 所示。

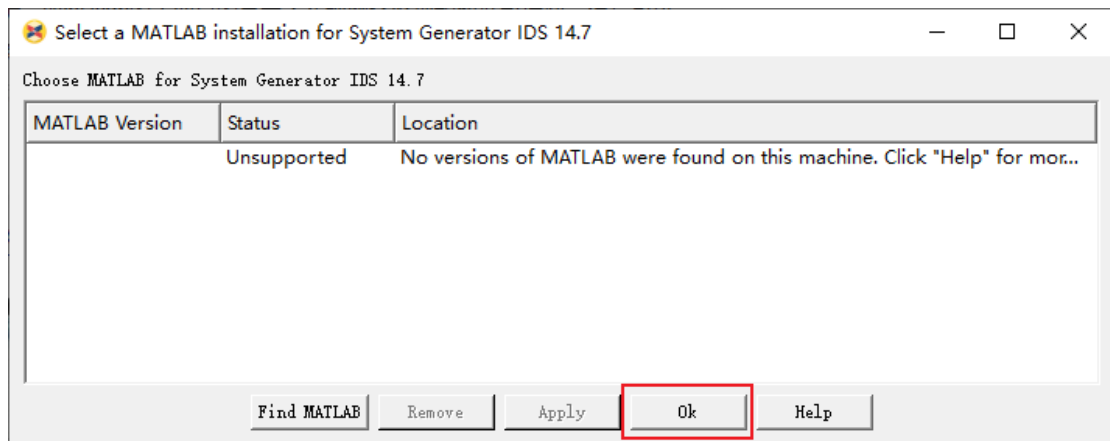


图 1-17 MATLAB 定位对话框

14) 进度达到 100%后，安装完成，点击“Finish”按钮关闭安装程序。如图 1-18 所示。

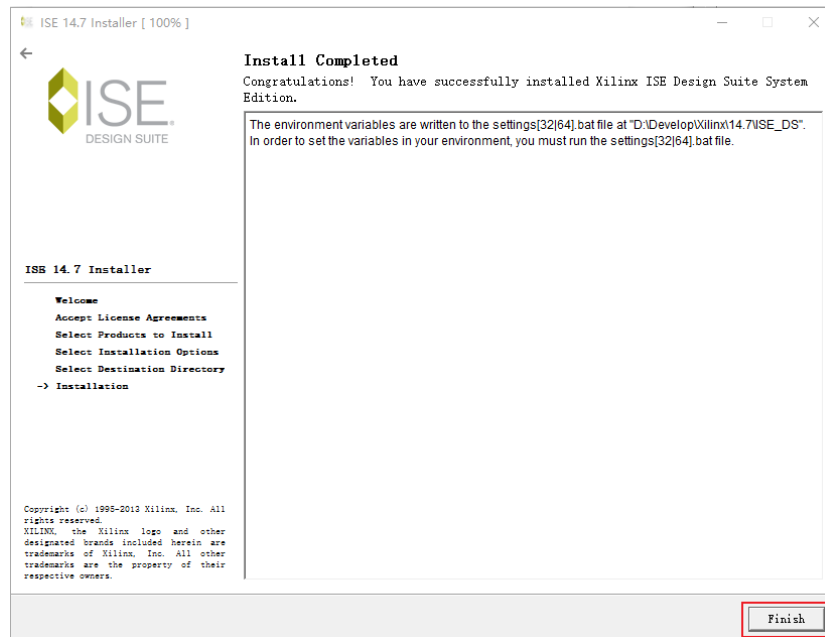


图 1-18 安装结束

### 1.3 软件注册

在软件安装结束时，会自动弹出注册页面，按以下流程对软件进行注册。

1) 在注册页面中选择“Manage Licenses”选项卡，如图 1-19 所示。

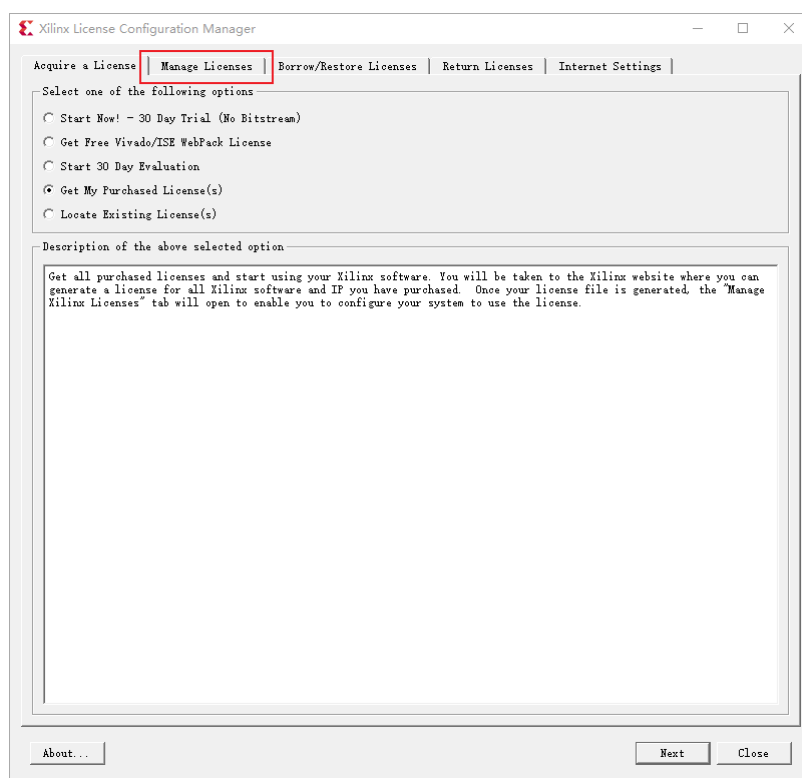


图 1-19 注册对话框页面

2) 在此页面中，首先在“XILINXD\_LICENSE\_FILE”文本框中输入注册文件 xilinx\_ise.lic 所在目录，该文件存放在安装目录中的\iso\Crack 文件夹下，如图 1-20 所示。

名称	修改日期	类型	大小
.xinstall	2017/5/5 13:35	文件夹	
bin	2017/5/5 13:35	文件夹	
common	2017/5/5 13:35	文件夹	
Crack	2017/5/5 13:35	文件夹	
data	2017/5/5 13:35	文件夹	
edk	2017/5/5 13:35	文件夹	
idata	2017/5/5 13:38	文件夹	
ise	2017/5/5 13:38	文件夹	
labtools	2017/5/5 13:38	文件夹	
Microsoft.VC90.CRT	2017/5/5 13:38	文件夹	
Microsoft.VC90.MFC	2017/5/5 13:38	文件夹	
msg	2017/5/5 13:38	文件夹	
-p	2015/11/23 10:20	文件夹	
planahead	2017/5/5 13:38	文件夹	
planahead_wp	2017/5/5 13:38	文件夹	
sysgen	2017/5/5 13:38	文件夹	
webpack	2017/5/5 13:38	文件夹	
autorun.inf	2013/10/14 2:47	安装信息	1 KB
xinfo.exe	2013/10/14 2:50	应用程序	741 KB
xsetup.exe	2013/10/14 2:50	应用程序	748 KB

图 1-20 注册文件所在路径

2) 在此页面中，首先在“XILINXD\_LICENSE\_FILE”文本框中输入注册文件 xilinx\_ise.lic 所在目录，该文件存放在安装目录中的\iso\Crack 文件夹下，而安

装文件解压在“D:\Tools\Xilinx ISE 14.7”路径下，所以对话框中的完整路径为“D:\Tools\Xilinx ISE 14.7\iso\Crack”。

填入注册文件路径后点击文本框右侧的“Set”按钮完成设置，如注册成功注册许可列表中的某些选项背景会变为绿色，表示许可操作，注册成功，此时可以点击“Close”按钮关闭注册页面，如图 1-21 所示。

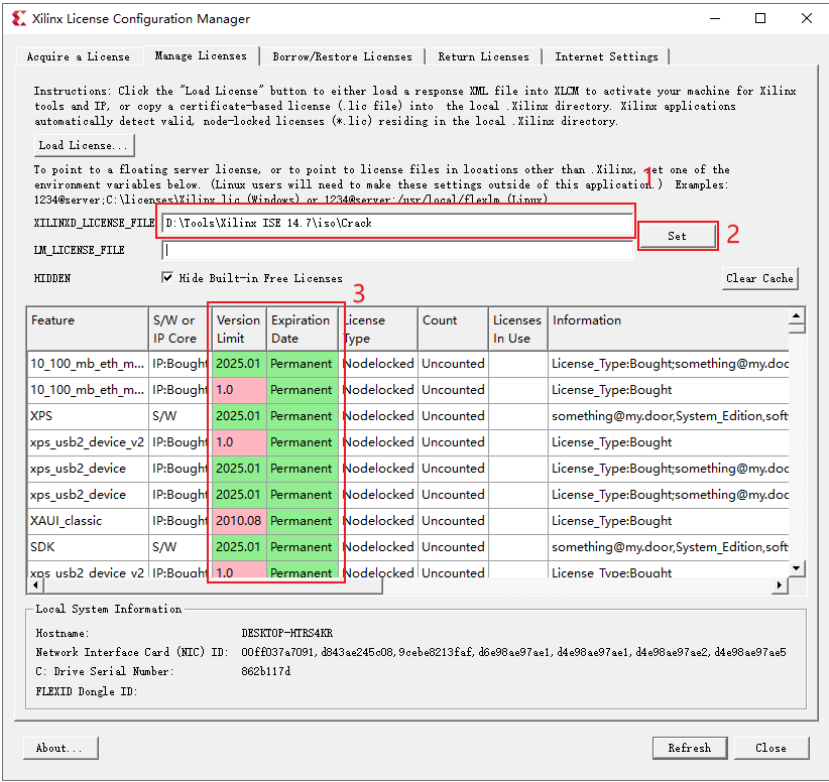


图 1-21 注册成功页面

## 1.4 bug 修复

在 Windows 10 操作系统环境下，安装 Xilinx ISE 14.7 后直接使用的过程中会产生一些 bug 问题，具体如程序崩溃，无法新建工程等问题，需要对安装程序进行一些修改。

找到程序安装路径下的两个文件夹，如我的程序安装在“C:\Xilinx”目录下，则两个文件夹的路径分别为：

- 1) C:\Xilinx\14.6\ISE\_DS\ISE\lib\nt64
- 2) C:\Xilinx\14.6\ISE\_DS\common\lib\nt64

替换第一个文件夹中的“libPortability.dll”文件，将该文件夹中的“libPortabilityNOSH.dll”复制一个副本，并更名为“libPortability.dll”替换原文件。

将第一个文件夹中更名后的“libPortability.dll”文件覆盖至第二个文件夹中。

通过以上操作可以修复程序 bug，使 Xilinx ISE 14.7 开发环境在 Windows 10 操作系统下顺利运行。

## 2. Xilinx ISE 14.7 开发环境使用简介

本节主要介绍 Xilinx ISE 14.7 开发环境的基础使用方法。通过实现图 2-1 中的电路结构，展示从新建工程、添加模块、到仿真验证的一个操作流程。

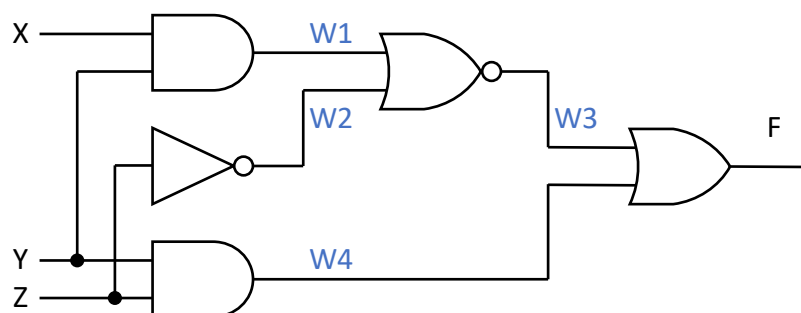


图 2-1 电路例图

### 2.1 界面说明

Xilinx ISE 14.7 开发环境可以双击桌面应用程序图标或从“开始菜单 -> Xilinx Design Tools -> 32-bit Project Navigator/64-bit Project Navigator”（选择 32 位应用程序还是 64 位应用程序根据操作系统环境而定）启动，桌面应用程序图标和菜单栏启动程序图标分别如图 2-2、2-3 所示。

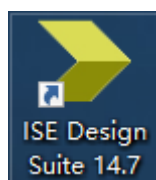


图 2-2 桌面应用程序图标

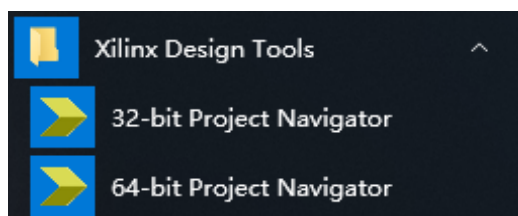


图 2-3 开始菜单程序图标

软件启动后界面界面如图 2-4 所示，

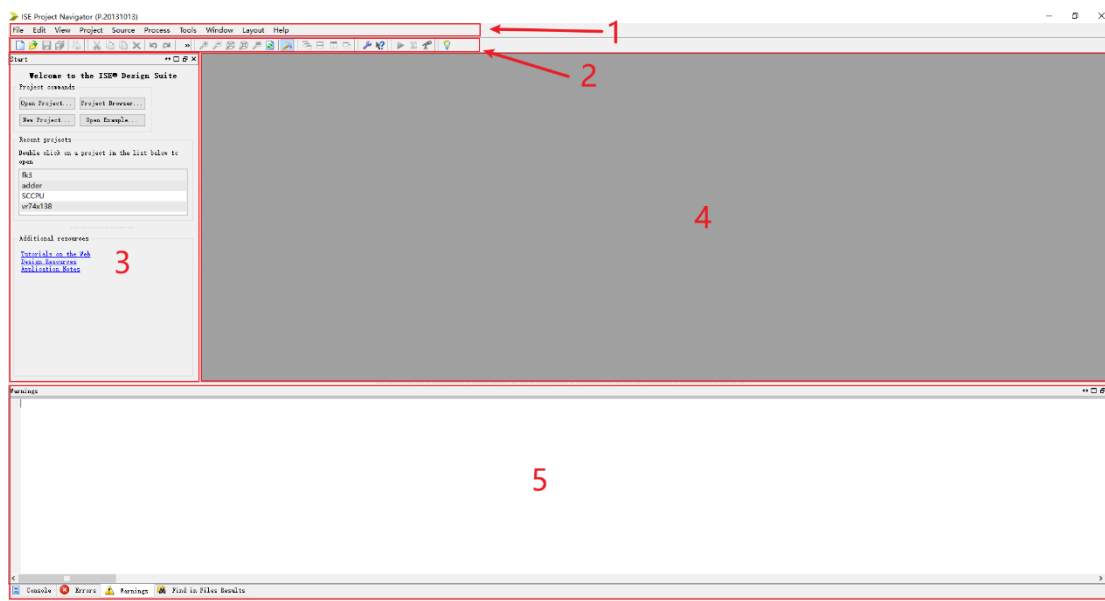


图 2-4 软件启动页面

其大致可划分为 5 个区域，以下分别做简要介绍：

- 1) 菜单栏，软件包含的所有功能都分列在各菜单项中。
- 2) 工具栏，部分软件功能为方便操作会以图标形式显示在工具栏当中。（如保存、放大页面、缩小页面等）
- 3) 操作栏，对具体文件的相关操作出现在此栏中。
- 4) 显示与编辑区域，查看或编辑特定文件时，其内容显示在此区域当中。
- 5) 控制台/信息提示区域，在对特定文件进行操作的过程中，相关的提示信息会显示在此区域当中，切换底部的选项卡可分类查看相关信息。

软件初始界面的所有内容都可以修改定制，如需要恢复至软件默认界面，可在菜单栏当中进行如下操作：“Layout -> Load Layout -> 1- Default Open Project”。

如需对开发过程中相关显示内容或形式进行修改，可在菜单栏 “Edit -> Preferences...” 中进行修改，如需要修改显示字体或字体大小可在菜单栏中查找如下关键字路径：“Edit -> Preferences... -> ISE Text Editor”。

## 2.2 新建工程

1) 开始设计工作首先需要新建一个工程，可在初始界面中点击“New Project”按钮，或在菜单栏当中选择 “File -> New Project...”，如图 2-5 所示。



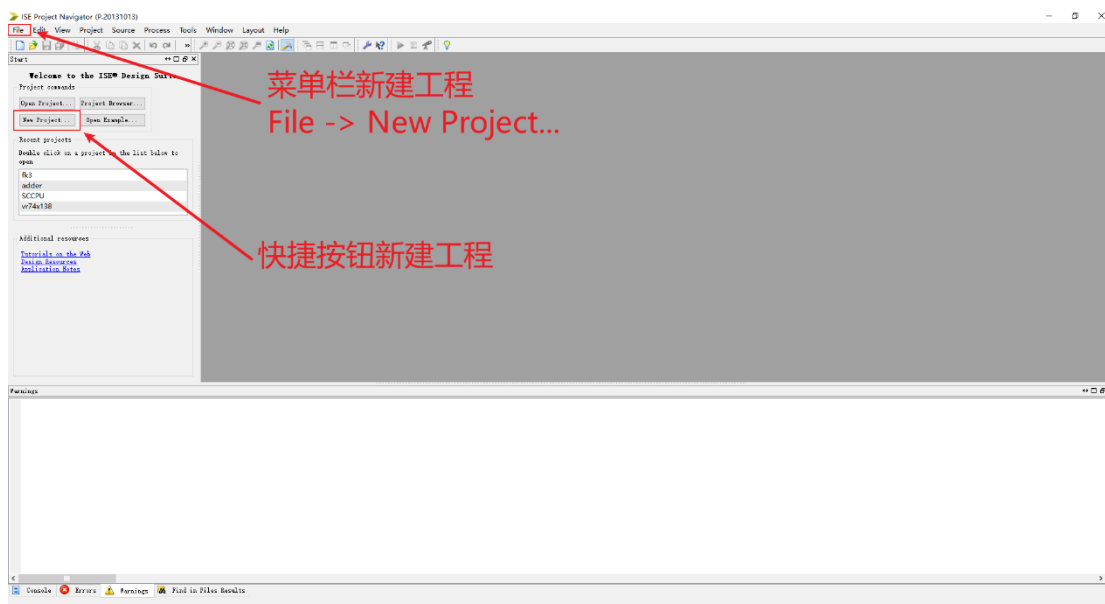


图 2-5 新建工程

2) 新建工程操作后弹出对话框，需要在文本框中输入工程名，工程保存地址与工作空间。

这里注意命名规则，名称当中可以出现数字、字母与部分特殊符号，但不能以数字与特殊符号作为第 1 个字符。

其次工程的保存路径当中不能够出现中文字符，否则会出现工程不能打开，工程文件无法编辑等情况。

其余选项保持默认即可，点击“Next >”进入下一步。如图 2-6 所示。

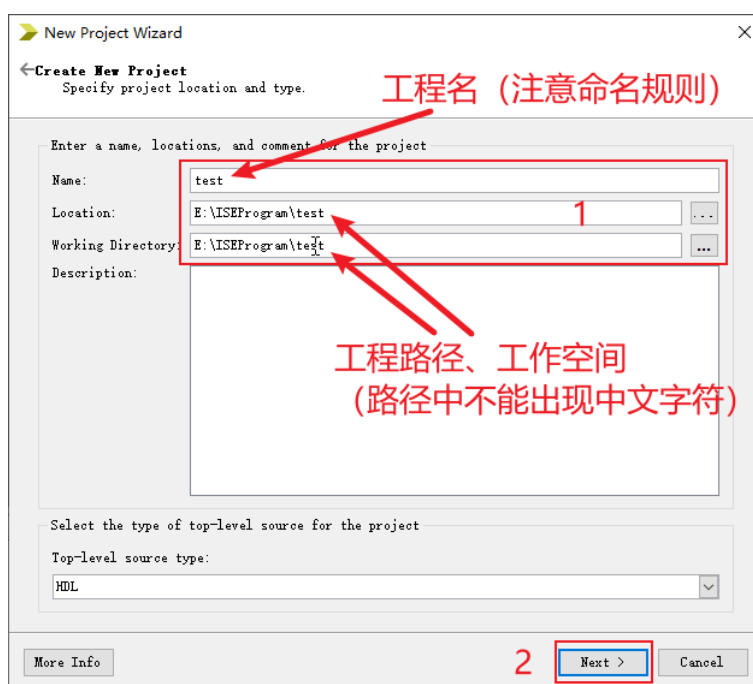


图 2-6 输入工程名与工程路径

3) 工程设置选项卡，其中的设置选项分为两大类，上半部分红框 1 中的设置主要与硬件开发平台相关，如只是软件设计并进行仿真不必在意其中选项。

红框 2 中的选项与软件设计相关，其中注意“Simulator”选项卡选择“ISim(VHDL/Verilog)”，“Preferred Language”选项卡选择“Verilog”，其余保持默认，点击“Next >”进入下一步，如图 2-7 所示。

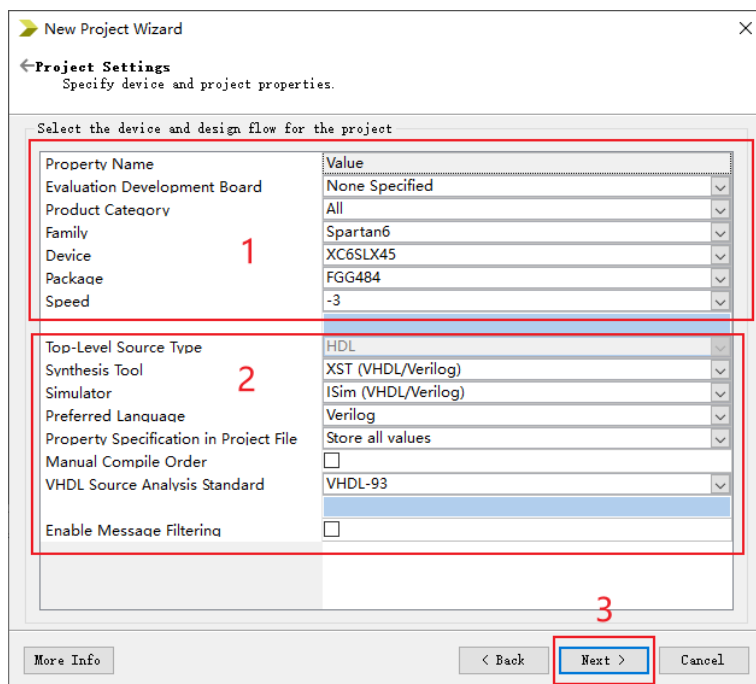


图 2-7 工程设置

4) 工程概览对话框，确认相关设置无误后点击“Finish”按键完成工程创建，如图 2-8 所示。

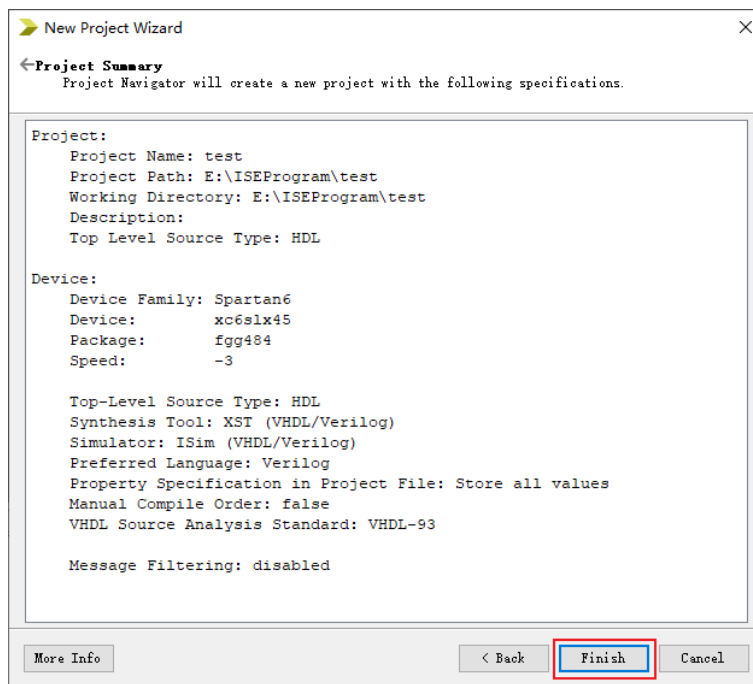


图 2-8 完成工程创建

5) 图 2-9 是新建工程后的软件界面，可以看到原先的“操作栏”产生了变化，分为上下半个区域。

上半区域为工程目录窗口，当前工程中的所有文件都会以树状图的形式展示在该窗口中。可以看到当前该窗口中的内容，第一行显示“Hierarchy”，表示该窗口中的内容以树状层次展示，第二行的“test”表示当前工程名，第三行的型号表示当前工程所采用的硬件开发平台，由于设计采用软件仿真的形式，不必关心具体开发平台的型号。当前工程目录是一个空目录，不包含任何文件，因此窗口内容显示“Empty View”。

下半区域为文件操作窗口。一个工程中可以包含各种不同的文件，针对不同类型的文件可以进行的操作也互不相同。当在工程目录窗口中选中某个文件时，该文件可进行的操作会显示在文件操作窗口中。

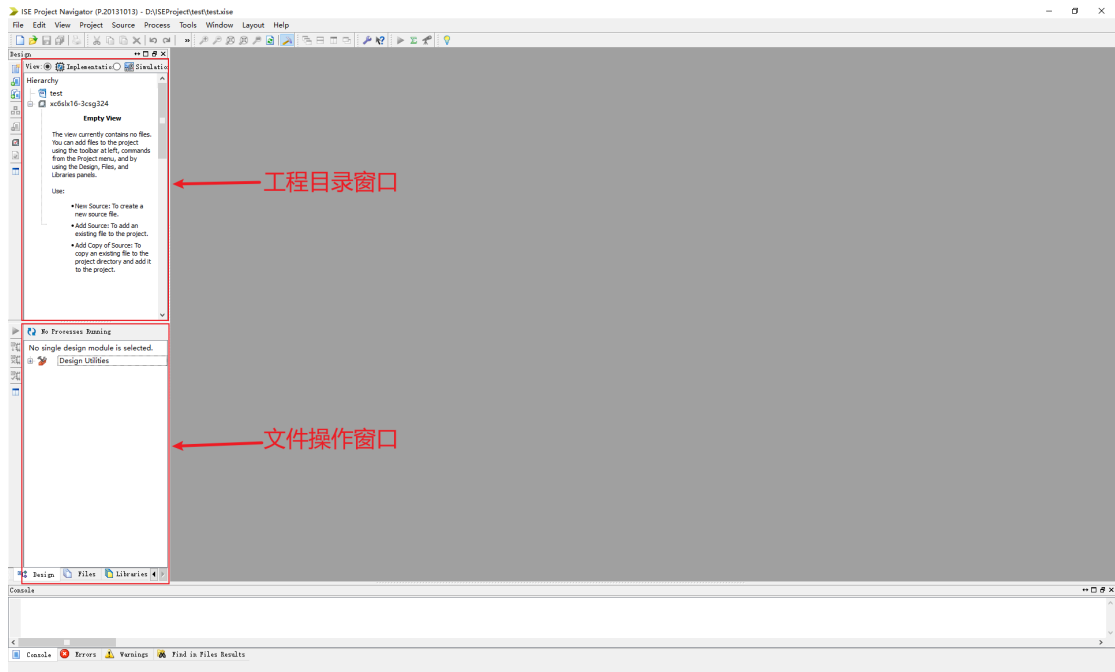


图 2-9 工程创建后的软件界面

## 2.3 添加模块

往工程中添加文件，本文档中只考虑添加采用 Verilog HDL 进行描述的模块设计或测试文件。添加文件既可以添加已经编辑好的 Verilog 模块文件，也可以新建一个空文件进行编辑。

1) 添加已经编辑好的文件可以利用工程目录左侧的工具栏按键，或利用菜单栏选项，选择“Project -> Add Source...”或“Project -> Add Copy of Source”，其中的区别在“Add Source...”可以将其他路径下的文件添加至工程中，但文件本身不会存放在工程目录下，“Add Copy of Source...”则会在文件添加至工程的同时将一份文件的副本拷贝至工程目录当中。一般情况下在添加文件时推荐采用添加副本的方式，如图 2-10 所示

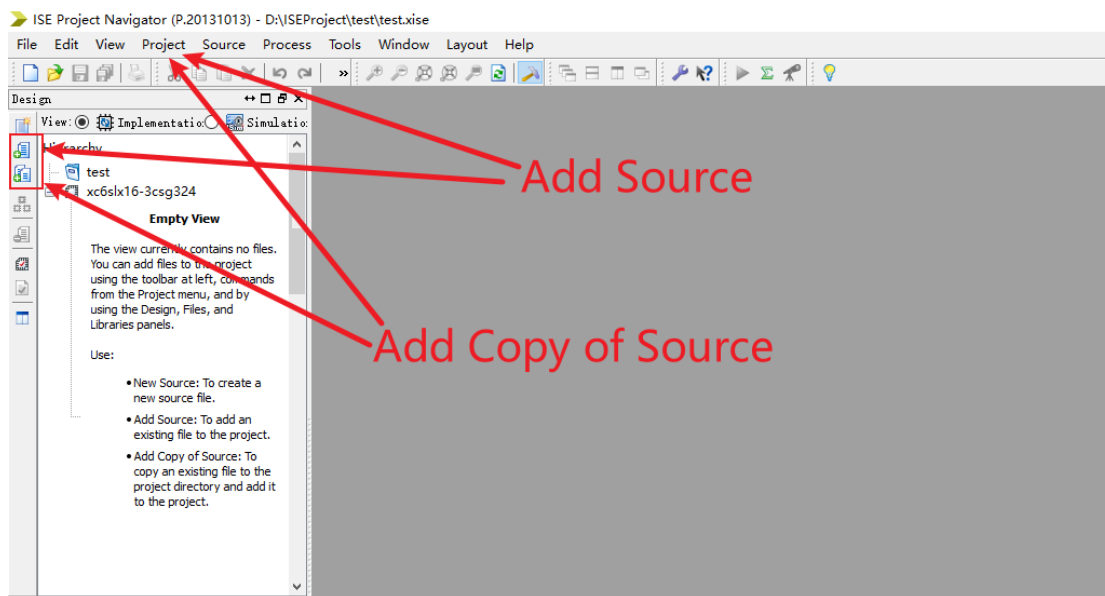


图 2-10 添加文件

2) 也可以采用新建空文件的方式向工程中添加文件，同样采用工程目录左侧的工具栏按钮或菜单栏选项中的“Project -> New Source...”，如图 2-11 所示。

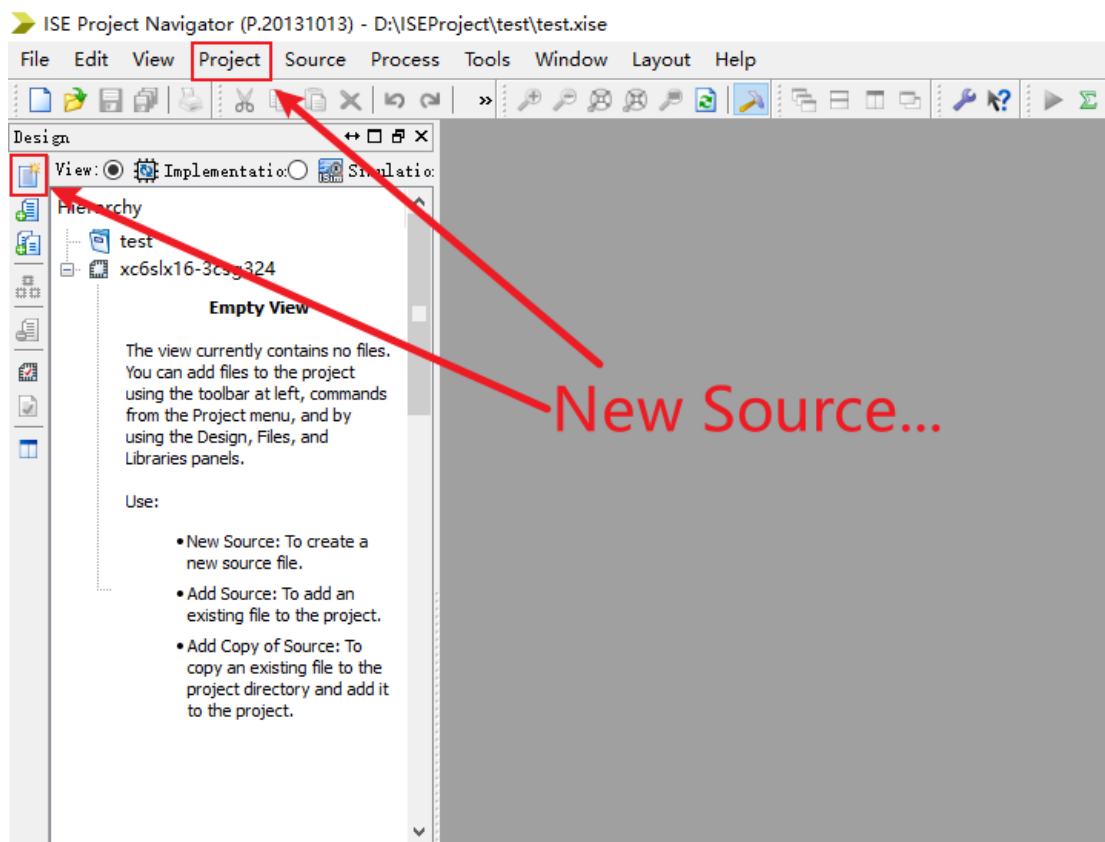


图 2-11 新建文件

在弹出的新建文件对话框中，左侧显示可以新建文件的类型，选择“Verilog Module”表示新建文件类型是一个 Verilog 模块。在“File name:”文本框中输入新建文件的名称，同 1.2 小节“新建工程”中的命名注意事项，注意命名规则。默认保持勾选框“Add to project”，表示将新建的文件添加至工程中。确认以上选项完成后点击“Next >”按钮。

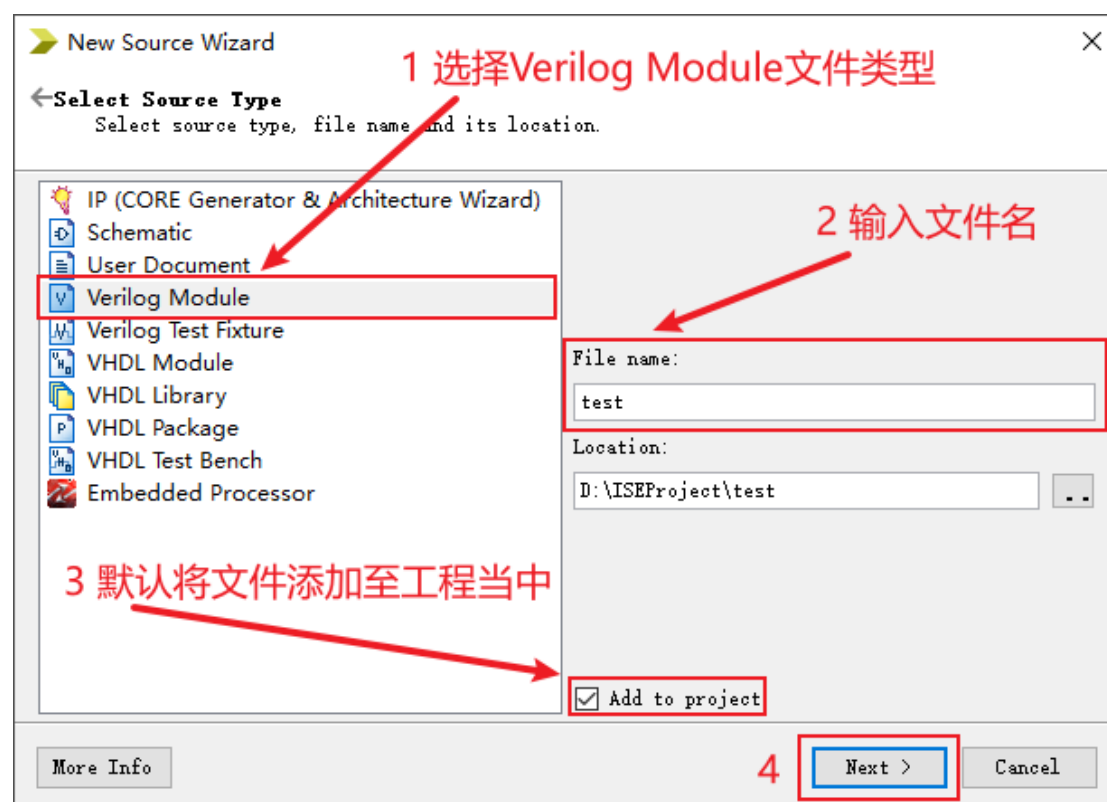


图 2-11 新建文件

在弹出的模块定义对话框中可以对模块接口结构进行描述定义，这一工作在模块建立后进行代码编辑同样可以实现，故对话框中接口预定义的工作可以跳过，直接点击“Next >”进入下一步，如图 2-12 所示。

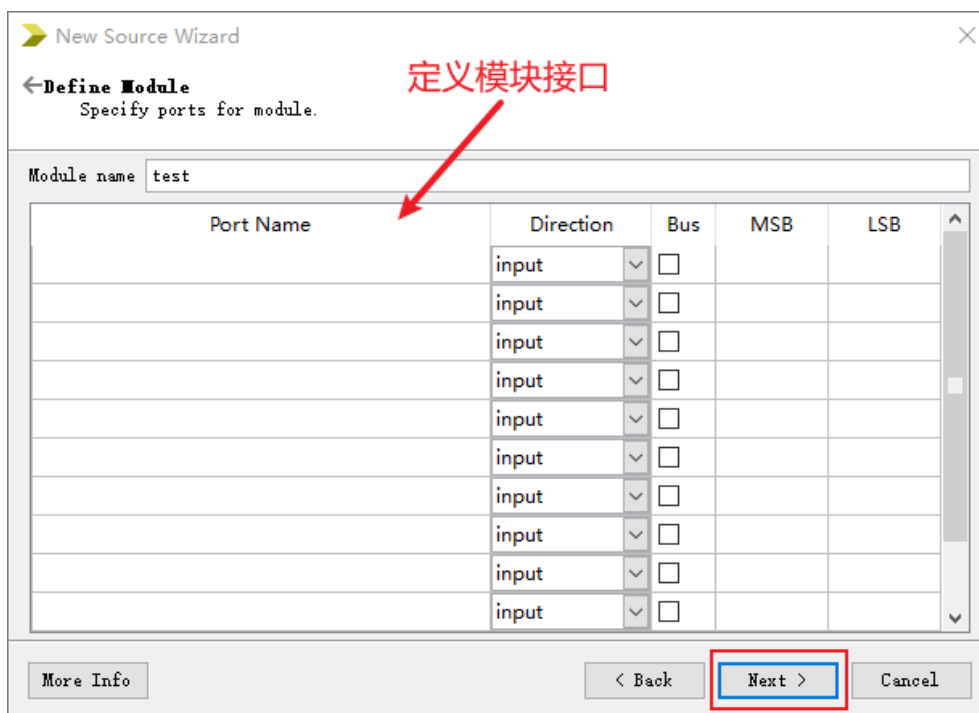


图 2-12 模块接口预定义

新建模块概览对话框，确认信息无误后点击“Finish”完成文件创建，如图 2-13 所示。

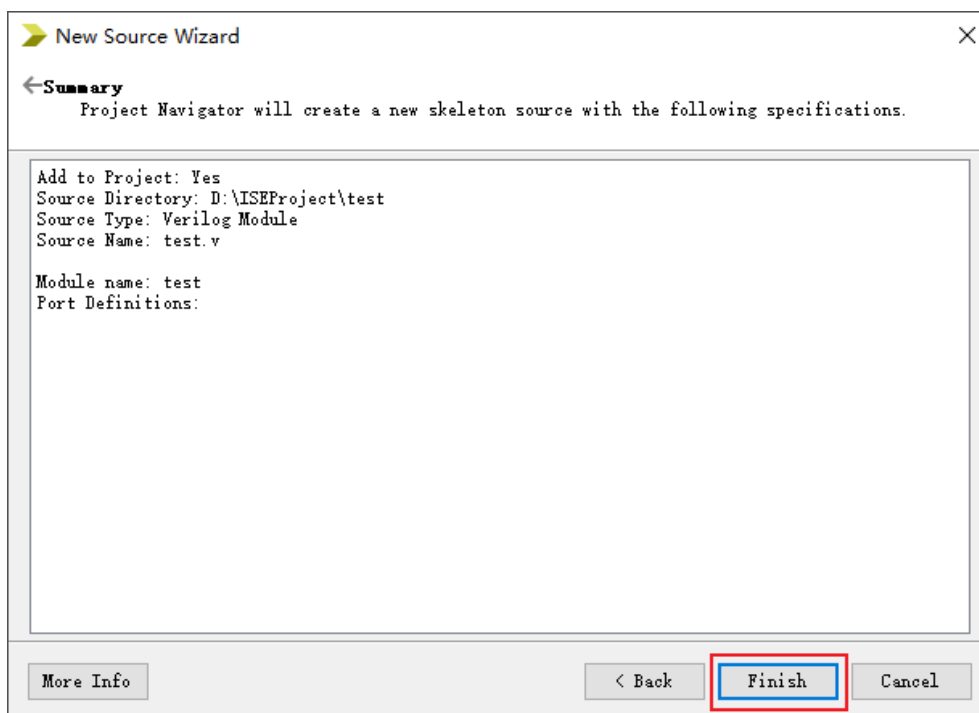


图 2-13 完成文件创建

## 2.4 模块编辑

完成 Verilog Module 文件的添加后，需要对该文件进行编辑以进行电路描述与设计，可参照如下步骤与说明进行操作，如图 2-14 所示。

1) 新添加的文件出现在工程目录窗口中，最前边的符号“V”表示当前文件类型是一个 Verilog Module 文件，其后的标识表明当前模块是一个顶层模块（在一个 ISE 工程当中只允许有一个顶层模块）。后边的字符串“test (test.v)”，前面的“test”表示当前 Verilog Module 的模块名是“test”，后面的“test.v”表示该文件在工程目录下的完整名称，可利用名称对相应文件进行检索。

2) 双击相应的文件名，该文件的内容即展示在代码编辑区域中，可以对相应文件的内容进行编辑。

3) 文件编辑完成后需要进行保存，可以使用工具栏中的快捷按钮，或在菜单选项卡中，选择“File -> Save”或者“File -> Save All”保存文件，两者的区别位“Save”按钮仅保存当前正在编辑的文件，“Save All”将工程目录中的所有文件进行保存。

4) Synthesize（综合）是集成电路设计过程中的重要步骤，将 HDL 描述翻译成门级网表，并输出网表文件。可以简单理解为根据当前模块文件内容，检测其中是否存在逻辑错误，或无法由基础门电路实现的描述，若没有错误则综合通过。在 ISE 开发环境下需要选中“顶层模块文件”，然后在操作窗口中双击“Synthesize - XST”进行综合操作，综合过程一般情况下需要一定时间，综合过程中的关键信息会显示在底部状态信息栏中。

5) 若综合结果出现错误或警告，可在状态信息栏中选择“Errors”或“Warnings”选项卡查看对应提示信息，这通常对代码调试工作有非常大的帮助，部分提示信息可以直接定位至可能产生警告或错误的代码段。



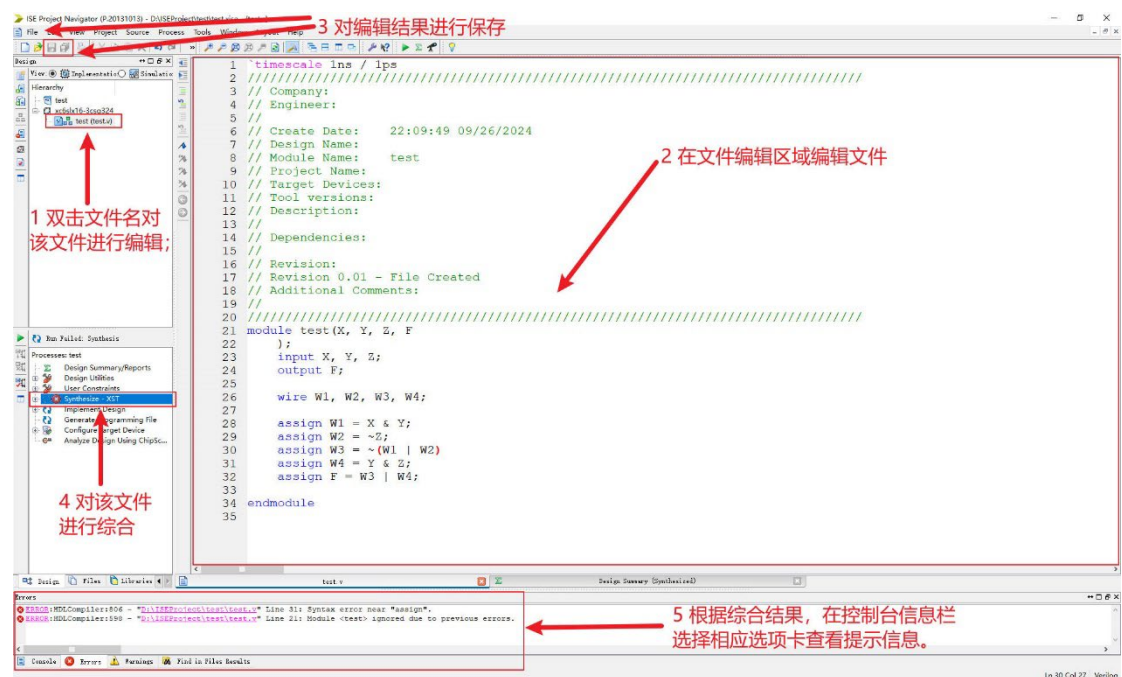


图 2-14 模块编辑

在本例中 Verilog Module 输入的完整模块代码如下：

```
module test(X, Y, Z, F      //接口变量
);
    input X, Y, Z;      //输入接口声明
    output F;           //输出接口声明
    wire W1, W2, W3, W4;      //中间连接信号定义
    //以下部分采用连续赋值语句对电路逻辑功能进行描述
    assign W1 = X & Y;
    assign W2 = ~Z;
    assign W3 = ~(W1 | W2);
    assign W4 = Y & Z;
    assign F = W3 | W4;
endmodule
```

6)在工程中存在多个模块文件的情况下，只有顶层模块可以进行综合操作，对非顶层模块只能进行“Check Syntax”（语法检查）操作，如图 2-15 所示。模块“abc”是一个非顶层模块，无法进行综合，如想对该模块进行综合则需将其设置为顶层模块，具体操作方法为在工程目录窗口中选中文件，鼠标右键展开选项菜单，选择“Set as Top Module”选项并进行确认。



图 2-15 设置模块为顶层模块

7) 若综合过程通过，操作窗口中“Synthesize - XST”选项前会出现对应的标识，同时控制栏中会显示提示信息“Process “Synthesize - XST” completed successfully”，表示综合过程通过，模块编辑工作阶段完成，可以进入模块仿真工作阶段，如图 2-16 所示。

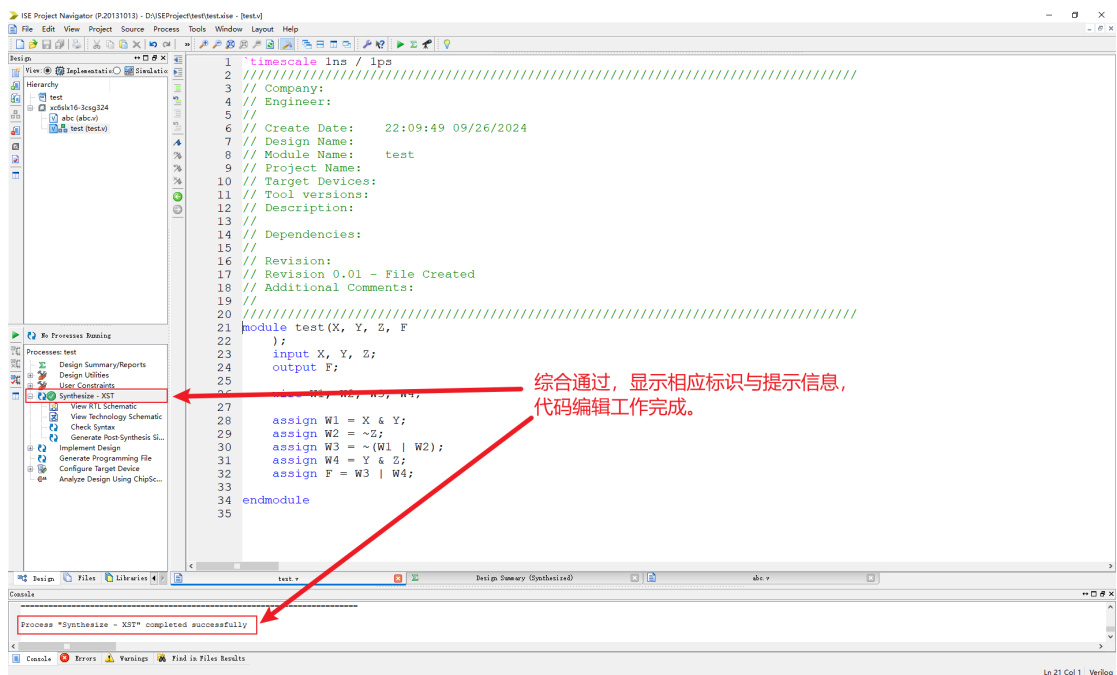


图 2-16 综合通过

## 2.5 模块仿真

1) ISE 开发环境具有设计视图与仿真视图两种不同的显示模式，这两种模式在现实内容方面具有一定区别。准备模仿仿真工作时需要将显示模式更改为仿真视图，在工程目录窗口的顶部，将“View”视图选项卡中的选项由“Implementation”设计视图变为“Simulation”仿真视图，如图 2-17 所示。

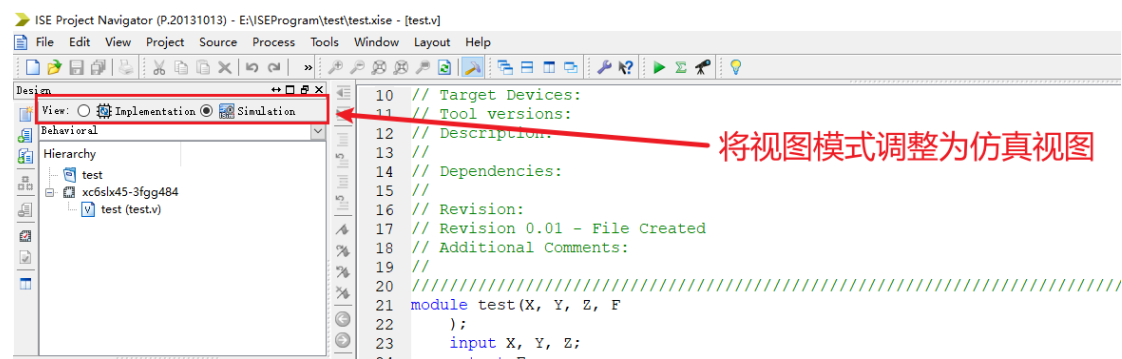


图 2-17 仿真视图模式

2) 对 Verilog Module 进行仿真需要向工程中添加与模块对应的仿真激励文件，添加文件的方式与 2.3 小节“添加模块”中所描述的操作方式基本相同，差异在于文件类型选择“Verilog Test Fixture”，如图 2-18 所示。

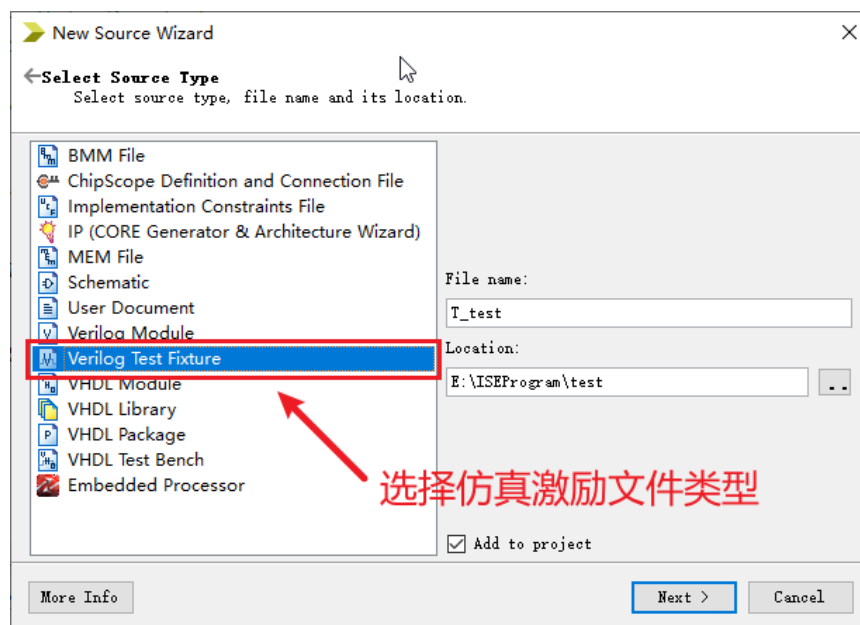


图 2-18 新建仿真激励文件

3) 随后弹出的对话框需要选择与新建仿真激励文件相匹配的 verilog 模块文件，根据所选择的模块，激励文件会自动生成相应的测试框架，减少仿真过程的

工作量，如图 2-19 所示。

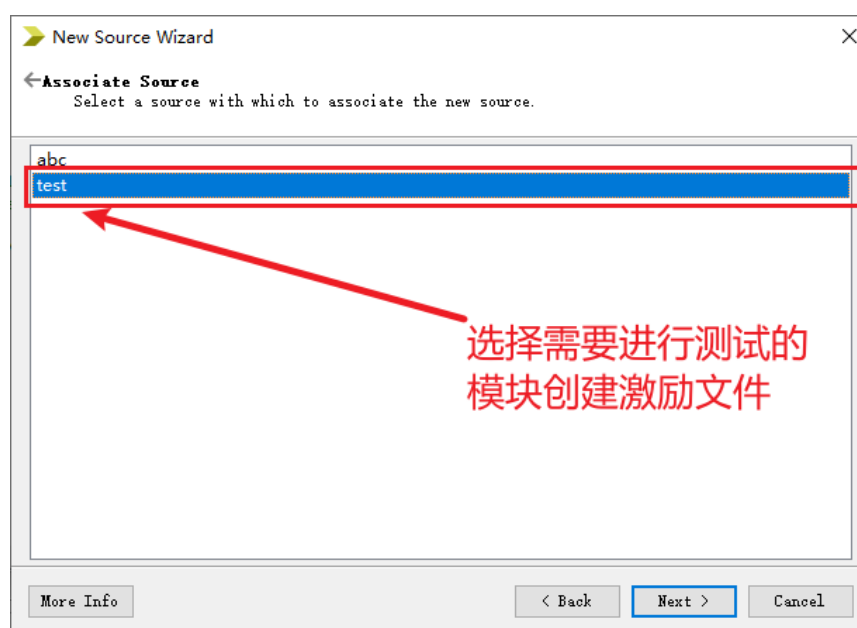


图 2-19 选择测试模块

4) 确认信息无误后，点击“Finish”按钮完成仿真激励文件的创建，如图 2-20 所示。

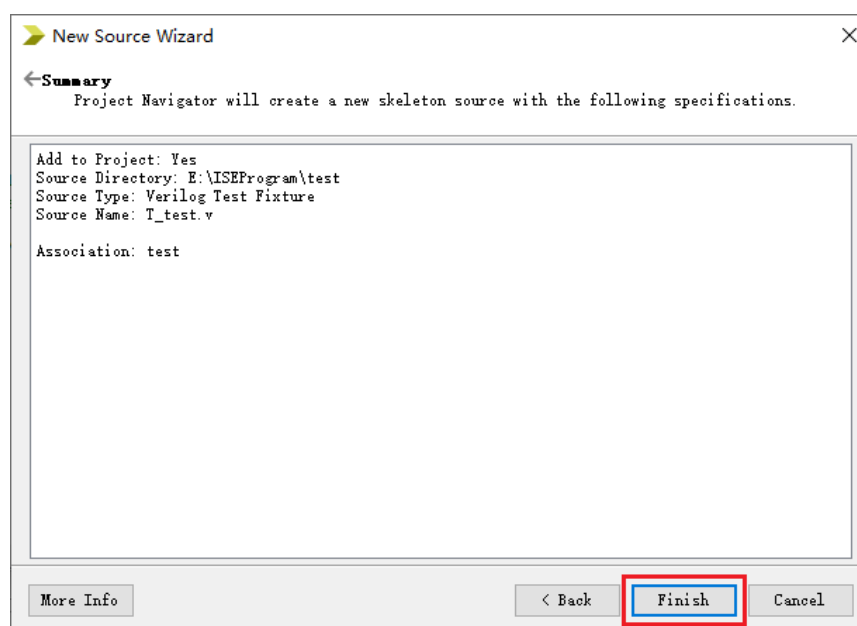


图 2-20 完成仿真激励文件的创建

5) 可以看到新创建的仿真激励文件出现在仿真视图下的工程目录窗口中，其命名方式与 2.4 小节“模块编辑”小节中描述相同。即仿真激励文件也是 Verilog 模块文件，其模块结构和描述语法与 Verilog Module 相同，只是用法有所区别，Verilog Module 文件用于对电路结构与功能的描述，而 Verilog Test Fixture 文件用

于对 Verilog 模块进行仿真。

与 2.4 小节所述操作方法相同，双击工程目录窗口中的文件名即可在右侧代码编辑区域编辑文件，根据提示将激励信号代码写入文件并进行保存。

在仿真视图下工程目录窗口中选中模块激励文件，可以看到操作栏中显示出能对激励文件进行的操作。其中第一行“ISim Simulator”表示当前使用 Isim 的仿真器进行仿真工作，展开后能进行两种操作；第一种操作“Behavioral Check Syntax”表示对激励文件进行语法检查，检测其是否符合仿真测试文件的语法规范；第二种操作“Simulate Behavioral Model”表示利用仿真激励文件对所要测试的 Verilog 模块进行行为仿真。双击操作栏中的选项即可进行相应的操作。

若要对仿真相关参数进行调整（如仿真时长等），可在操作栏中选中第二项“Simulate Behavioral Model”并右键展开菜单，选择其中的“Process Properties...”选项。相关操作如图 2-21 所示。

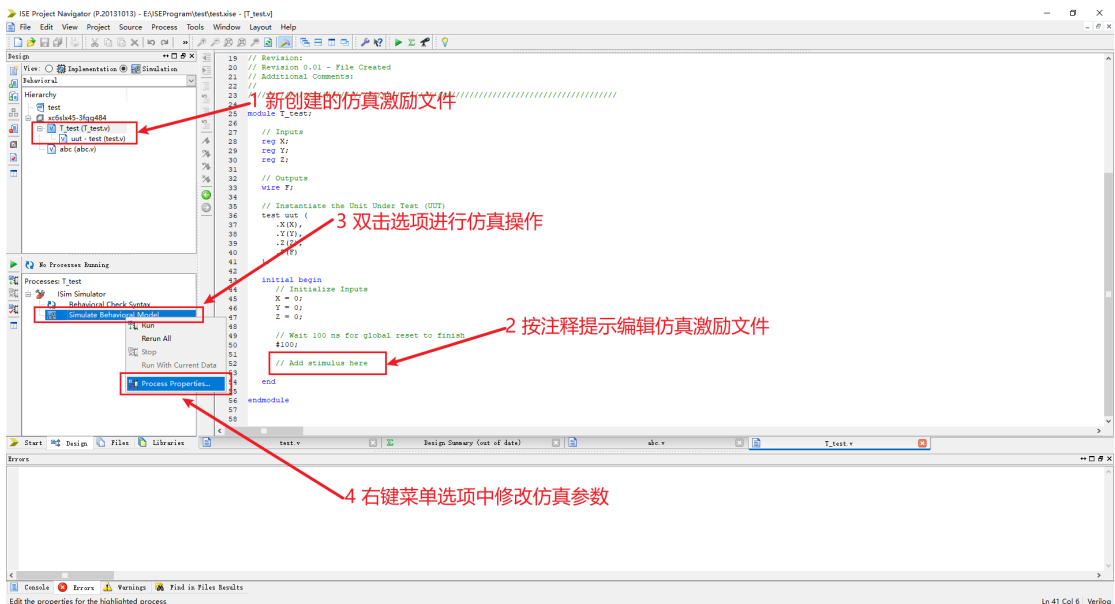


图 2-21 模块仿真操作

在本例中，在注释说明后输入如下激励文件代码：

```
X = 0; Y = 0; Z = 1;
#100; X = 0; Y = 1; Z = 0;
#100; X = 0; Y = 1; Z = 1;
#100; X = 1; Y = 0; Z = 0;
#100; X = 1; Y = 0; Z = 1;
#100; X = 1; Y = 1; Z = 0;
#100; X = 1; Y = 1; Z = 1;
```

若选择对仿真参数进行修改，则弹出仿真参数对话框。如要修改仿真时长，可修改“Simulation Run Time”参数栏的属性值，默认为 1000ns，如图 2-22 所示。

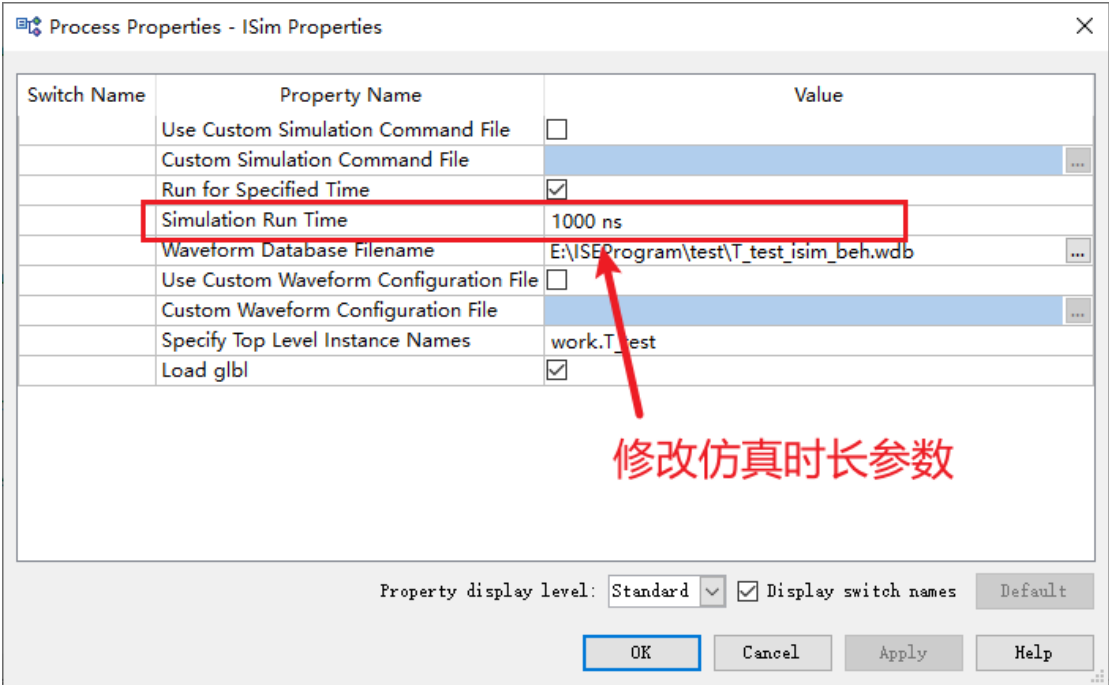


图 2-22 修改仿真参数

## 2.6 仿真结果分析

在集成电路设计流程中，仿真工作占有很大的比重。能够读懂仿真结果，运用相关仿真工具对仿真结果进行分析有助于发现电路设计过程中的漏洞与问题。

### 2.6.1 仿真结果界面

如使用仿真激励文件对 Verilog 模块进行仿真，会弹出仿真结果界面，整个界面大致分为 5 个区域，分别如下：

- 1) 仿真模块，此区域中展示参与仿真过程的相关模块。
- 2) 仿真对象变量及其值，此区域中展示参与仿真过程的对象及其最终取值。
- 3) 仿真波形结果，默认情况下为黑色背景，其展示整个仿真过程中各接口和对象在不同时刻的取值，又可分为三部分内容：
  - a. 显示接口及对象名称；
  - b. 显示某一时刻对应对象的取值；
  - c. 随时间变化的取值波形图，其中绿色线段在上表示当前时刻取指为高电平 1，绿色线段在下表示当前时刻取指为低电平 0，红色线段表示当前时刻取指

为不定值 x，蓝色线段表示对象接口处于高阻态，呈断开状态，无法输入输出。  
相关界面如图 2-23 所示。

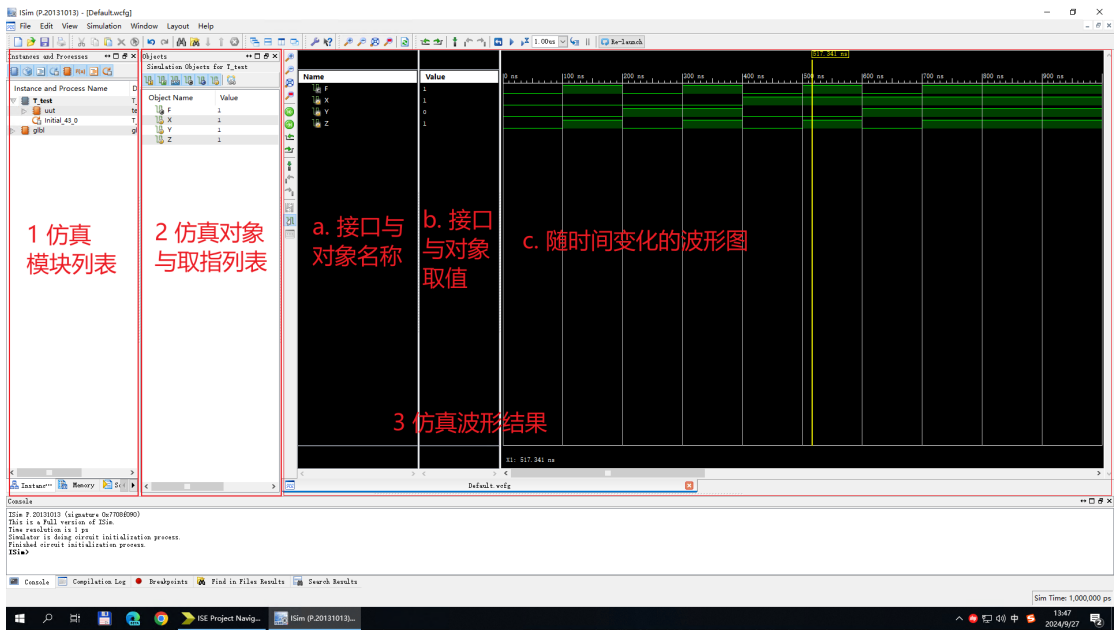


图 2-23 仿真结果界面

### 2.6.2 观察与调整仿真波形图

仿真结果是一个随时间变化的波形图，其总时长默认为 1000ns，在波形图顶部显示具体时刻，当用鼠标左键在波形图中某一处单击，会产生一条黄色的竖线经过该点，并在竖线顶部显示所选取的具体时刻。波形图中 Value 栏（对象取值）会变化显示各接口对象在该时刻所取得的值，也可以直接观察对应时刻各接口对象的波形得出取值，如图 2-24 所示。

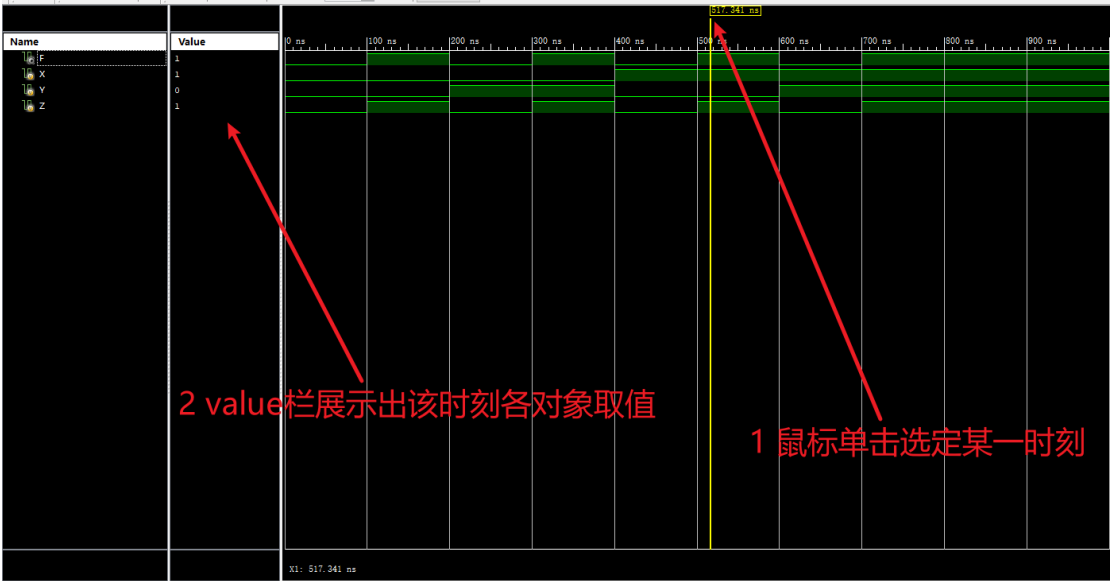


图 2-24 观察某一时刻接口与对象的状态

波形图中时间尺度可以进行放大或缩小以便观察，具体方法为利用工具栏中的 Zoom In 按钮缩小时间尺度，Zoom Out 按钮放大时间尺度，Zoom to Full View 按钮将整个仿真时长显示在当前波形图窗口中。其中最常用的为 Zoom to Full View 操作，键盘快捷键为 F6.，如图 2-25 所示。

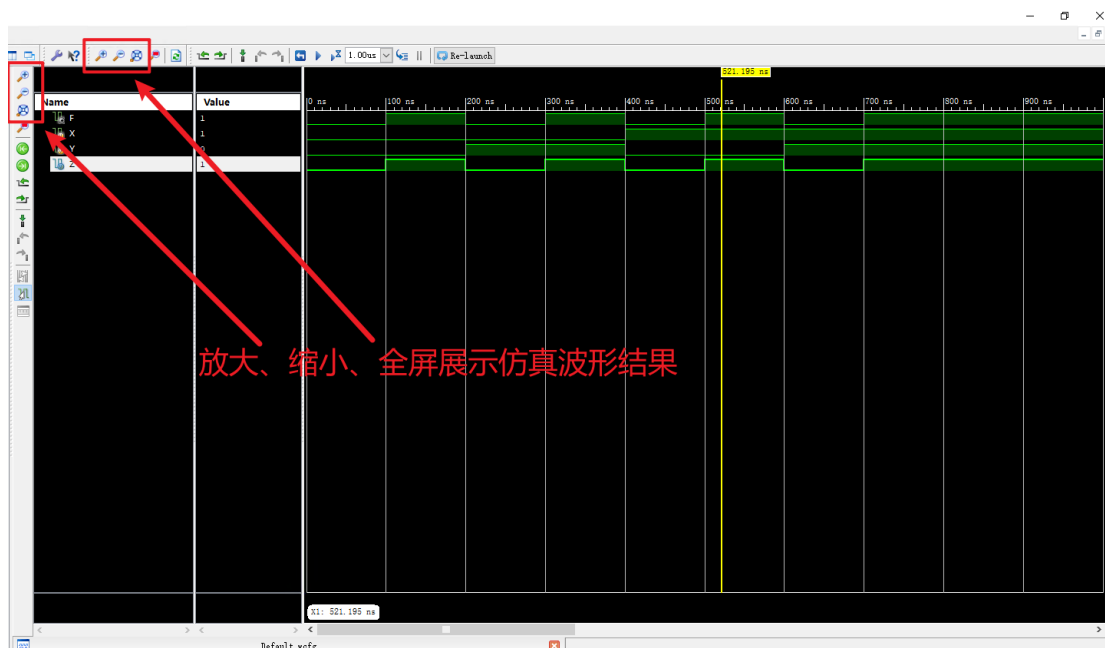


图 2-25 调整波形仿真结果

### 2.6.3 修改显示数值进制

波形图中各变量取值默认按照二进制进行显示，而某些时刻在变量位宽较大，如 16 位、32 位等情况下，直接观察数值的二进制结果就比较困难，此时可以将数值更改为其他进制(如十进制、十六进制等)进行显示。具体操作方式为在 Value 栏中选中希望更改进制显示的对象，鼠标右键点击展开菜单，从“Radix”一栏中选取具体进制，如图 2-26 所示，由于本例中所有数据与对象的位宽都是 1 位，无法修改进制，故图中展示为其他案例，仅做操作演示。



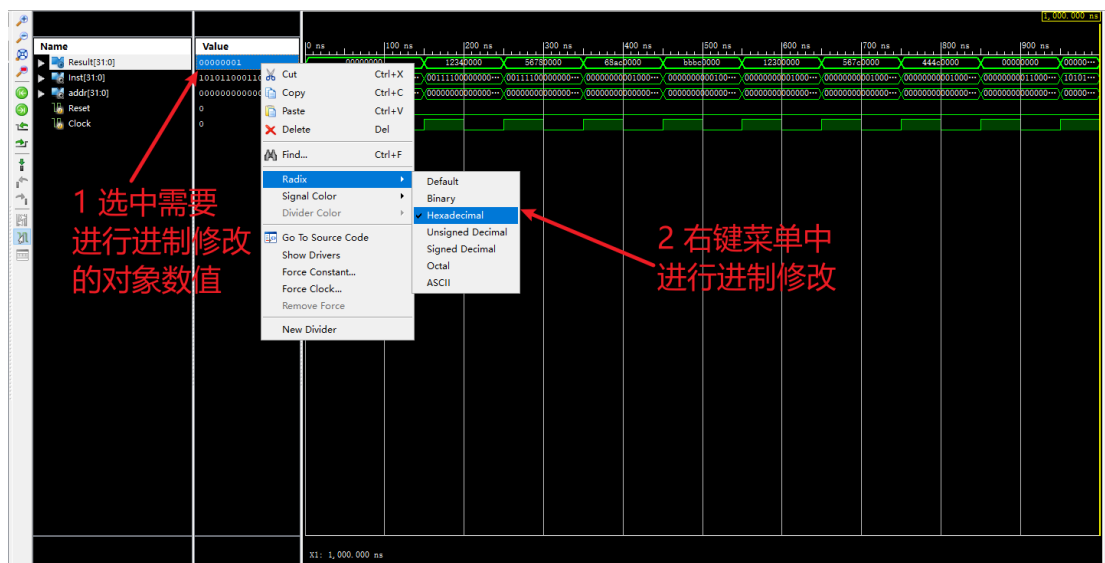


图 2-26 修改数值显示进制

## 2.6.4 观察中间信号

在仿真过程中，波形图默认展示模块接口变量的波形图，有时希望观察模块内部中间信号的波形图以帮助分析，可以将中间信号添加至波形图中。具体操作方式如下：

1) 在仿真结果界面的模块列表中，展开激励文件模块，从中选中“uut”模块（uut - unit under test，即被测试模块的实例化）；

2) 此时发现被测试的 Verilog 模块所有的接口信号与中间信号均出现在仿真对象与取值列表中，左键选中希望进行观察的中间信号，将其拖动至波形图界面当中；

3) 点击工具栏中的“Re - launch”按键重新运行仿真过程；

4) 此时可以在波形图界面中观察到所选中间信号。

以本文档案例进行说明，其中模块接口仅有 X，Y，Z 三个输入，F 一个输出共 4 个接口信号，内部具有 W1，W2，W3，W4 四个中间信号，按以上方式可以在波形图中观察到中间信号的值，如图 2-27 所示。

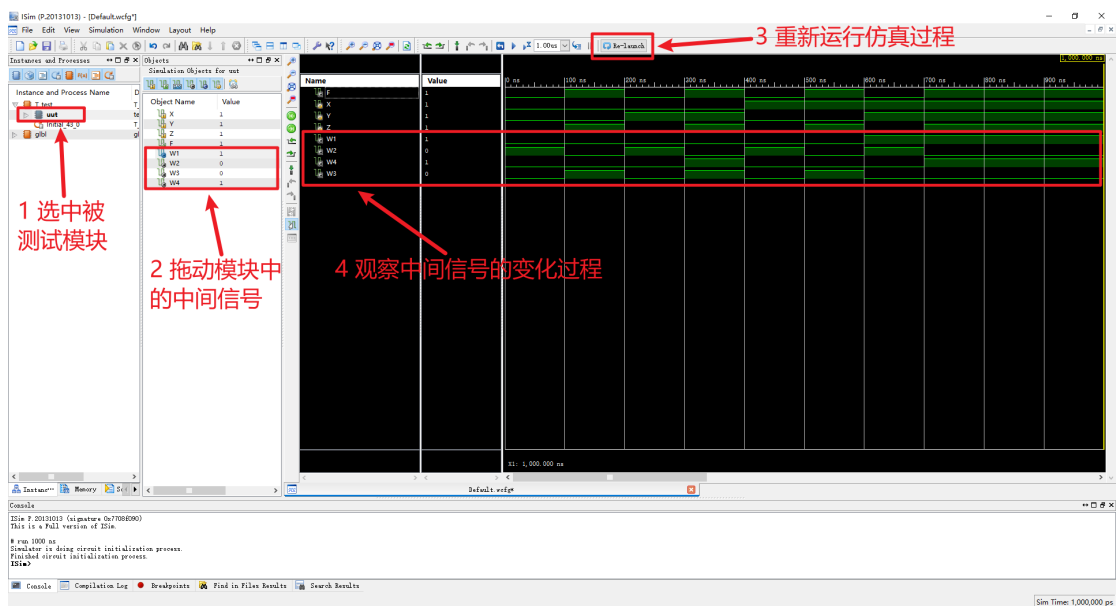


图 2-27 观察中间信号

## 2.6.5 观察存储器变量

在电路设计过程中有时需要对存储器进行建模产生存储器变量，如需要观察存储器中各地址中存放的值可按如下说明进行操作：

- 1) 仿真模块列表底部存在多个选项卡，默认情况下选中“Instances and Processes”选项卡，将其更改为“Memory”选项卡；
- 2) 此时若被测试的 Verilog Module 中存在存储器类型的变量，该变量名会出现在仿真对象与取值列表中；
- 3) 双击希望观测的存储器变量名，此时原波形图界面处会生成一个新的页面，在该页面中显示存储器变量中所有地址的值；
- 4) 在存储器观测页面也可对地址进制、数值进制等参数进行调整以便观测。

由于本例中没有建立存储器类型的变量，故图中展示为其他案例，仅做操作演示，如图 2-28 所示。

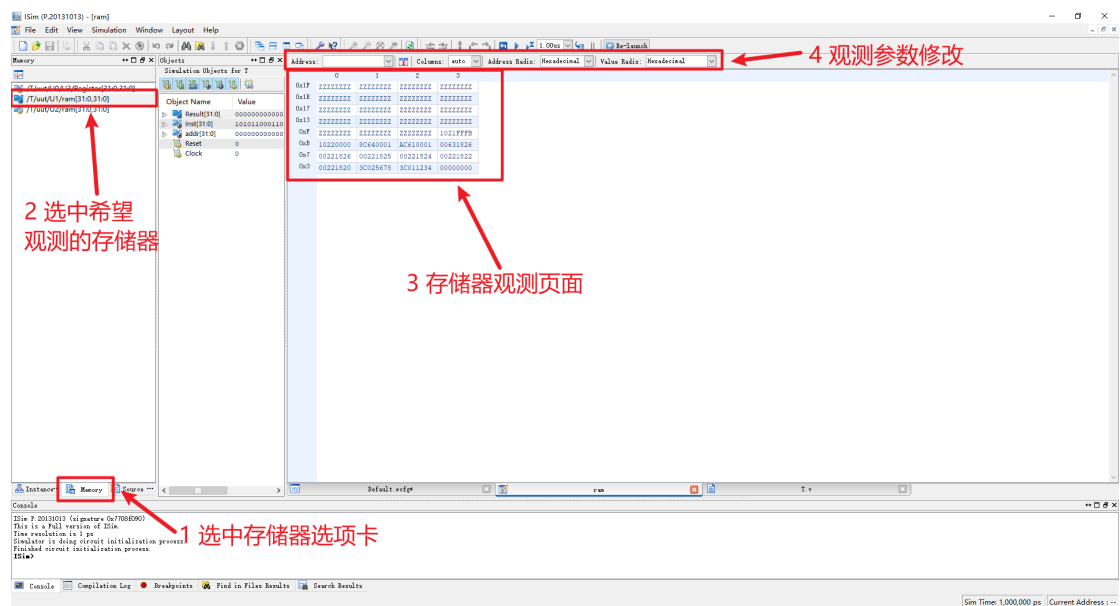


图 2-28 存储器变量观测