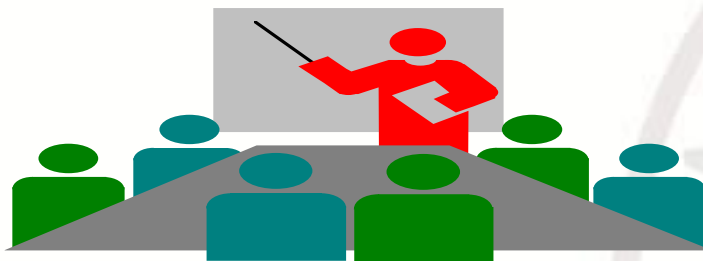




浙江大学
ZHEJIANG UNIVERSITY



逻辑与计算机设计基础

逻辑与计算机设计基础实验 与课程设计

实验一

EDA工具与实验平台

施青松

Asso. Prof. Shi Qingsong

College of Computer Science and Technology, Zhejiang University

zjsqs@zju.edu.cn

Course Outline





实验目的

1. 熟悉掌握Xilinx ISE EDA开发平台
2. 熟悉了解SWORD4.0 FPGA 开发平台
3. 学习运用Xilinx ISE平台输入逻辑设计、综合、时序
(行为) 仿真及设计约束(时序、引脚)
4. 学习运用Xilinx ISE平台下载流代码进行物理验证
5. 初步了解Verilog HDL语言



实验环境

实验设备

1. 计算机（Intel Core i3以上，1GB内存以上）系统
2. SWORD4.0 开发板
3. Xilinx ISE14.7及以上开发工具

材料

无

Course Outline





实验任务

1. 学习Xilin ISE 开发平台安装，并了解运行环境
2. 设计简单组合逻辑电路，并建立Xilin ISE设计管理工程（Project）
3. 采用逻辑图形输入逻辑功能描述， 并进行编辑、调试、综合、行为仿真，设计约束(时序、引脚)、实现及流代码下载运行与物理验证

Course Outline



一个简单的逻辑实例：逻辑图

- ◎ 某楼房的单元三层楼梯通道共用一盏灯，每一层楼都各安装一只开关，它能独立地控制这盏的开关，设计楼道灯的控制电路

S3	S2	S1	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$F = \bar{S}_3\bar{S}_2S_1 + \bar{S}_3S_2\bar{S}_1 + S_3\bar{S}_2\bar{S}_1 + S_3S_2S_1$$

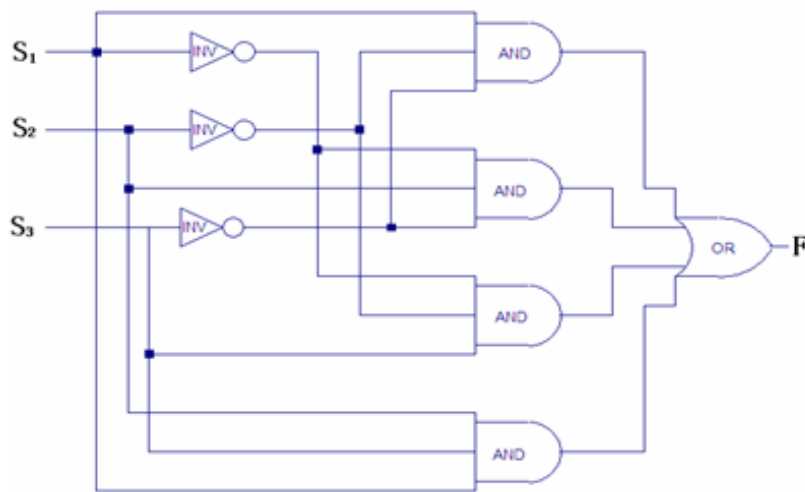


图 2-6 楼道灯控逻辑电路



一个简单的逻辑实例：硬件描述

- ◎ 改变控制要求，一旦灯打开后，延时若干秒后自动关闭，重新设计楼道灯的控制电路

```
`timescale 1ns / 1ps
module lamp_Controller(input wire clk,
                        input wire S1,
                        input wire S2,
                        input wire S3,
                        output wire F
                        );

parameter COUNTER=8;
reg y;
reg [COUNTER-1:0] count;
Initial begin                                //初始化
    count<=8'hFFFFFFFFF;
    y=0;
end
```

```
    assign w=S1||S2||S3;
    always@(posedge clk)
        if(w || count < 8'hFFFFFFFFF)
            begin
                y=1;
                count<=count+1;
            end
        else begin
            y=0;
            count<=count;
        end
    assign F=y;
```

endmodule

注：wire 信号变量，导线类型，ISE称为“NET”，说明可以省略
reg 寄存器变量，结果不一定是寄存器，用“<=”赋值

Course Outline



ISE启动

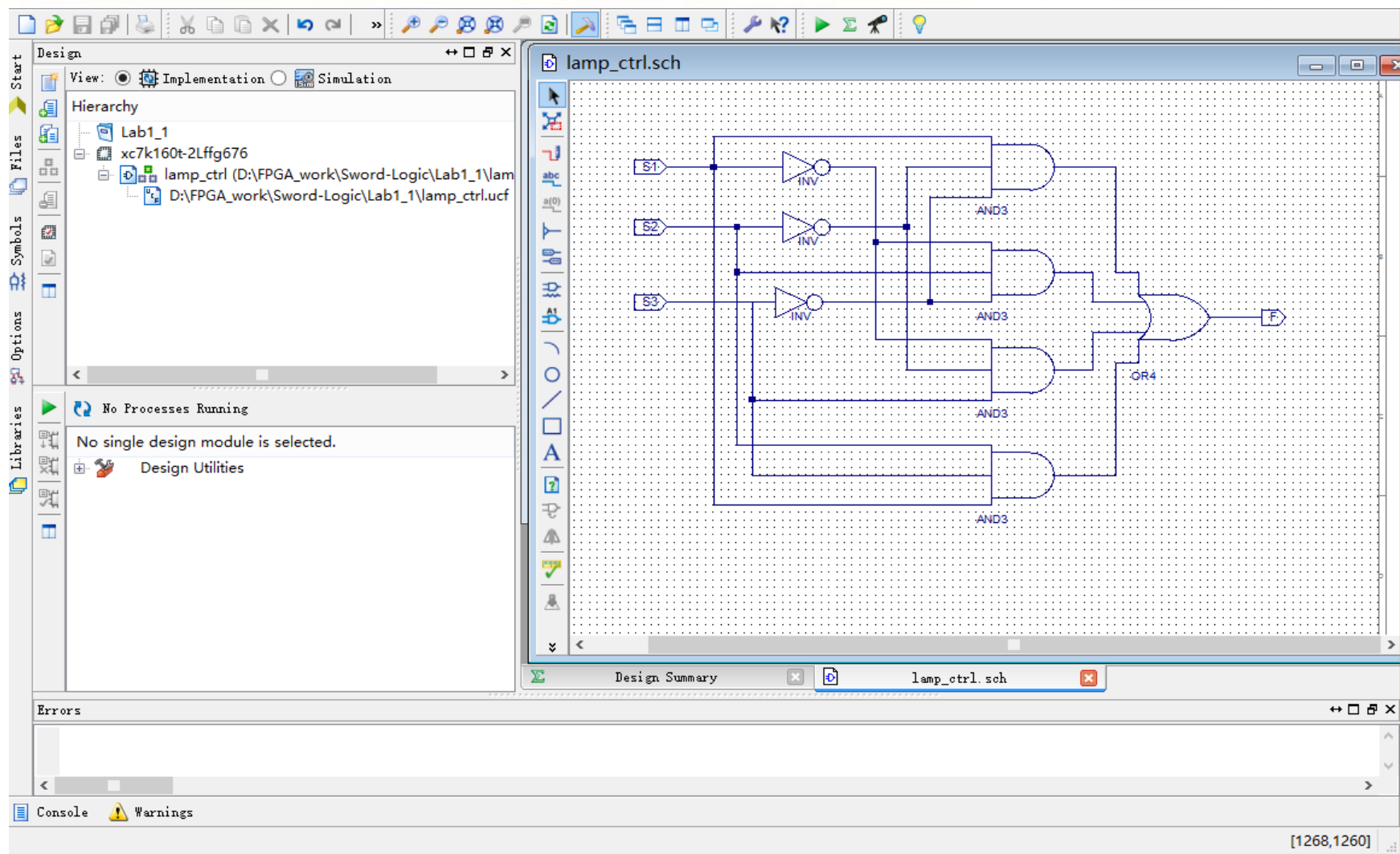
◎ 启动Xilinx ISE系统软件，双击桌面的图标



进入ISE开发环境



打开ISE集成开发平台





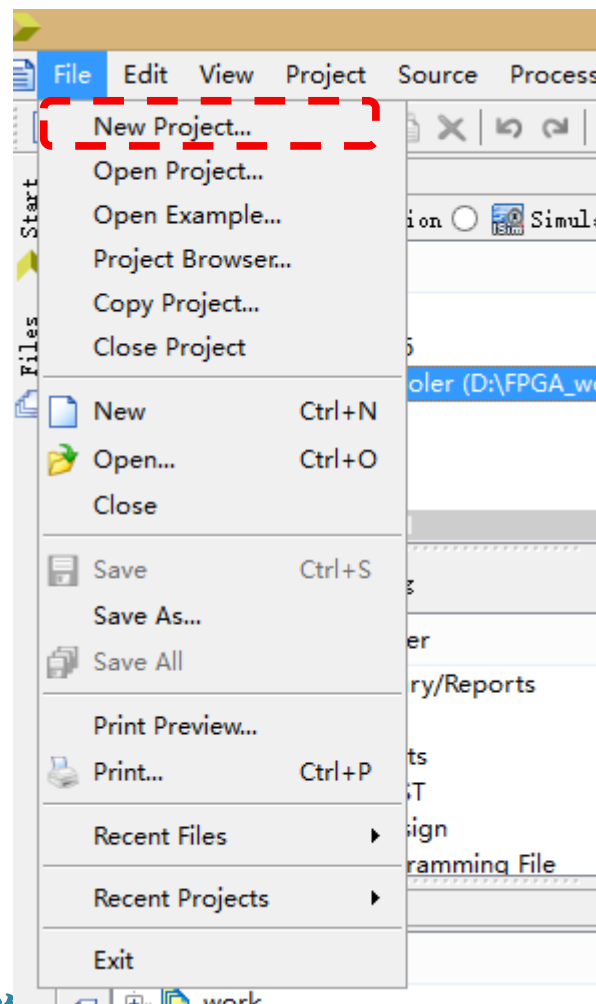
采用原理图输入逻辑功能描述

建立ISE实现数字系统设计工程



□ 【步骤1】 以图形描述方式建立楼道控制的工程文件：**lamp_Ctrl.isc**

(1) 点击左上方file，
选择new project新建project



(2) Project命名 :Lab1-1

top-level source type选择:schematic



The image shows a 'New Project Wizard' dialog box with the following fields and options:

- Create New Project** (with a back arrow icon)
Specify project location and type.
- Enter a name, locations, and comment for the project**
 - Name:** lab1-1 (highlighted with a red dashed box)
 - Location:** D:\FPGA_work\Sword-Logic\lab1-1
 - Working Directory:** D:\FPGA_work\Sword-Logic\lab1-1
 - Description:** 楼道灯定时控制
- Select the type of top-level source for the project**
 - Top-level source type:** Schematic (highlighted with a red dashed box)
- Buttons:** More Info, Next >, Cancel

(3) 点击Next到设备属性页

选择开发平台参数



The image shows the 'New Project Wizard' dialog box, specifically the 'Project Settings' step. The title bar says 'New Project Wizard' with a close button. The main heading is 'Project Settings' with a subtitle 'Specify device and project properties.' Below this, it says 'Select the device and design flow for the project'. The dialog contains a table of settings:

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Kintex7
Device	XC7K160T
Package	FFG676
Speed	-2L
Top-Level Source Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

At the bottom, there are three buttons: 'More Info', '< Back', and 'Next >'. The 'Next >' button is highlighted with a blue border. A red dashed circle is drawn around the 'Family', 'Device', 'Package', and 'Speed' rows in the table.

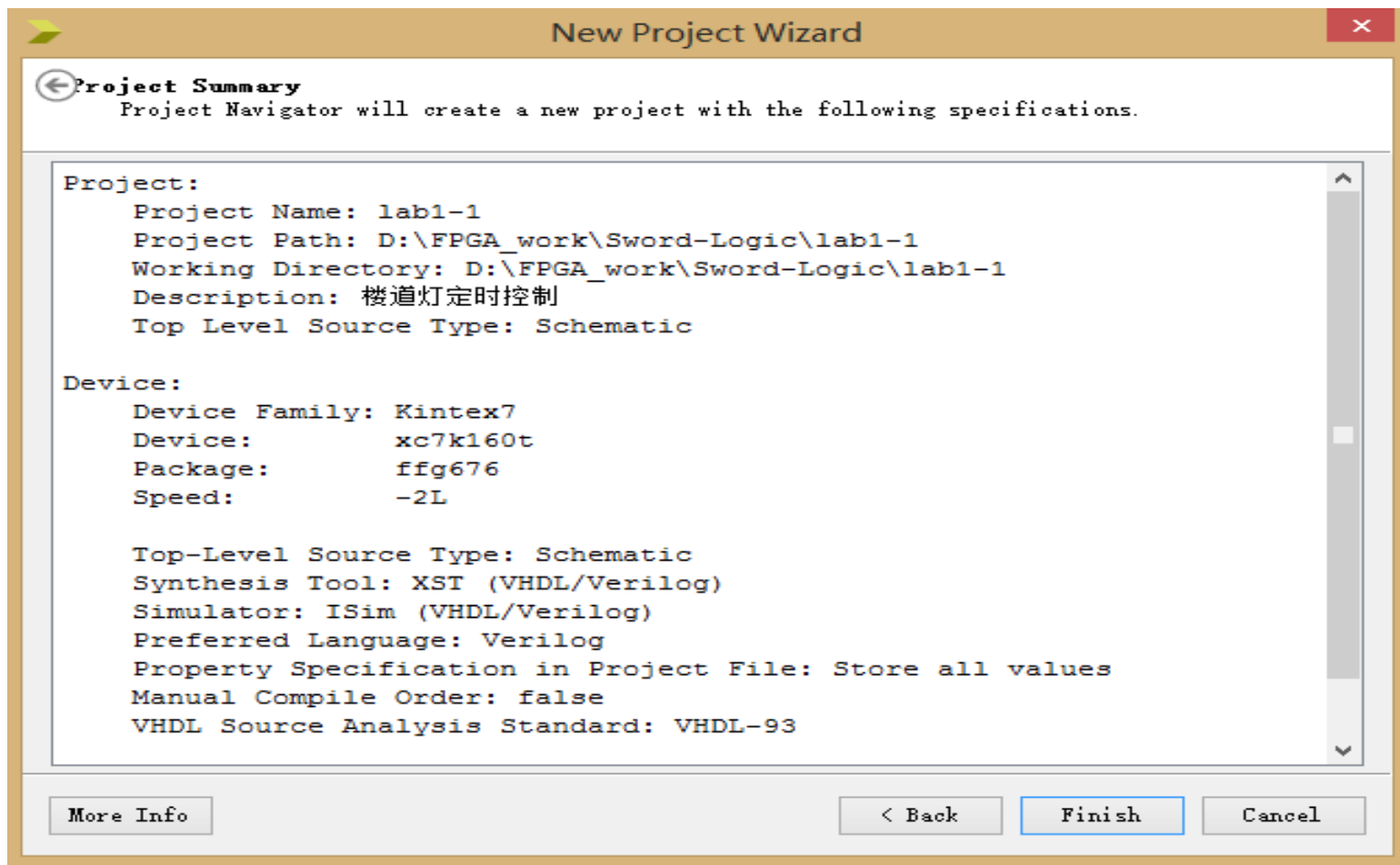
Family: Kintex7 Device: XC7K325T

Package: FFG676 Speed: -2

(4) 设置完开发平台详细属性信息后， 点击**Finish**新建工程模板

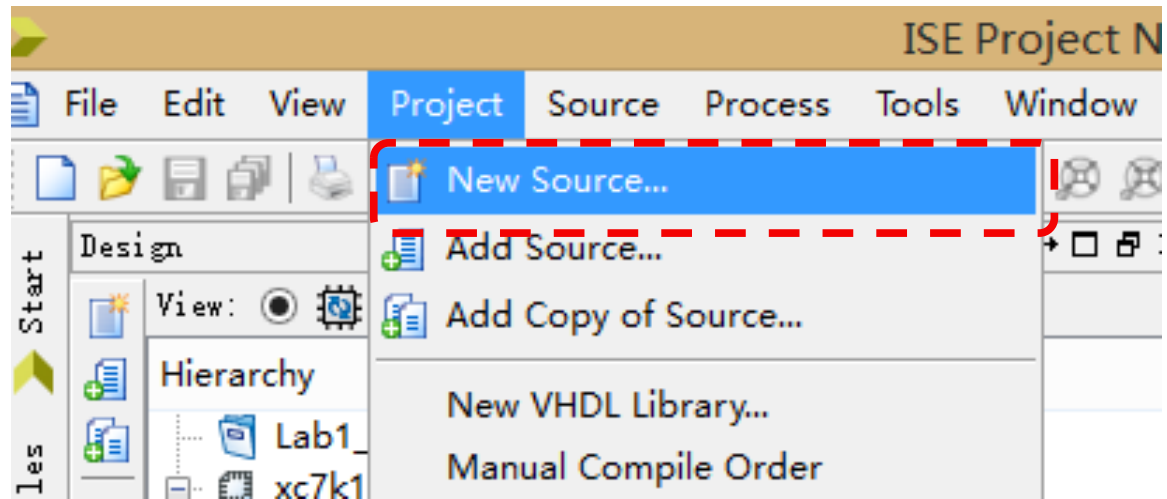


显示工程属性如下图所示，如果有误可以退回重新选择：

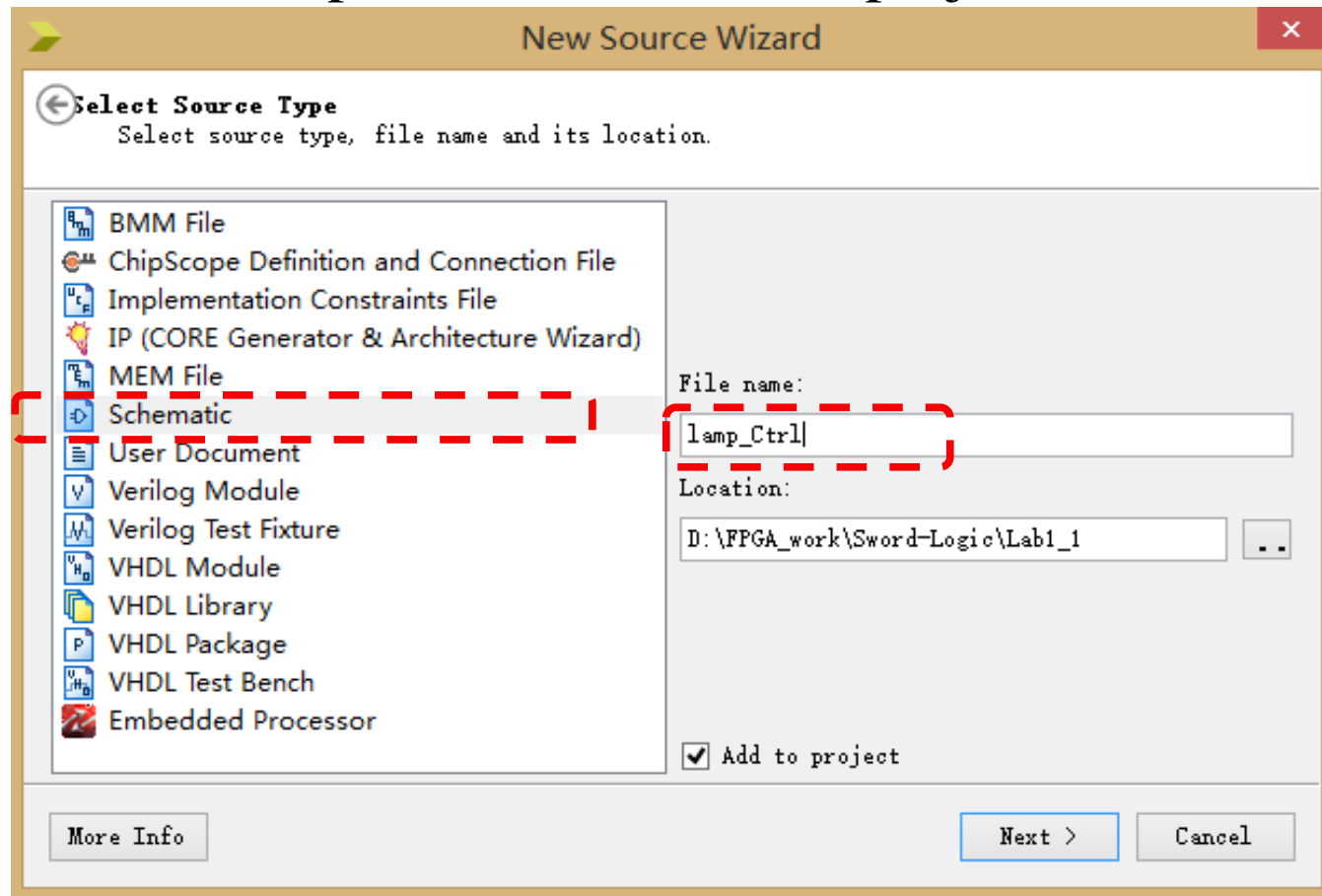


□ 【步骤2】 创建Schematic输入模板及源文件： **lamp_Ctrl.sch**

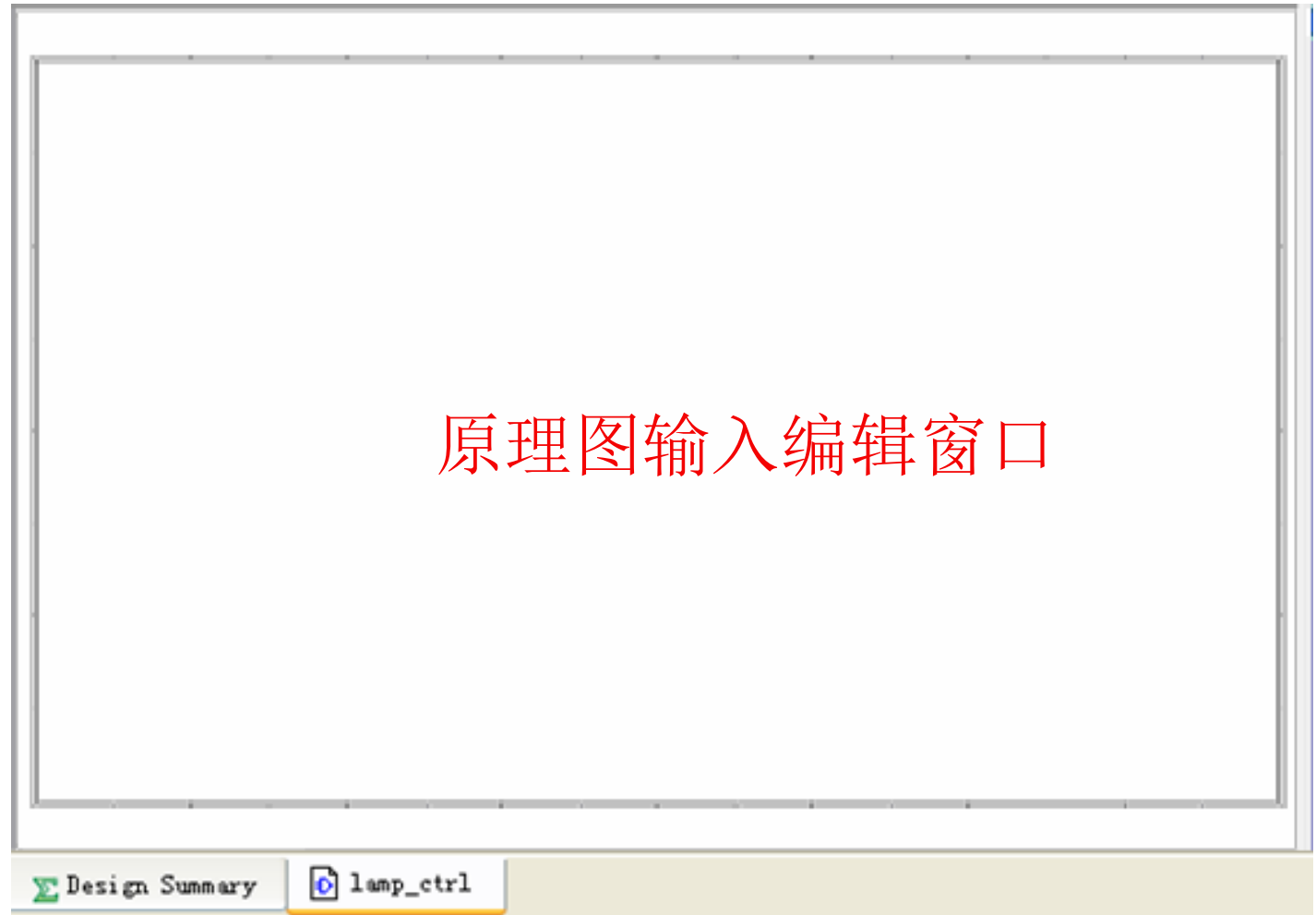
(1) 在Source窗口下空白处用鼠标右键点击New Source如下图所示（也可以从Project菜单进行）：



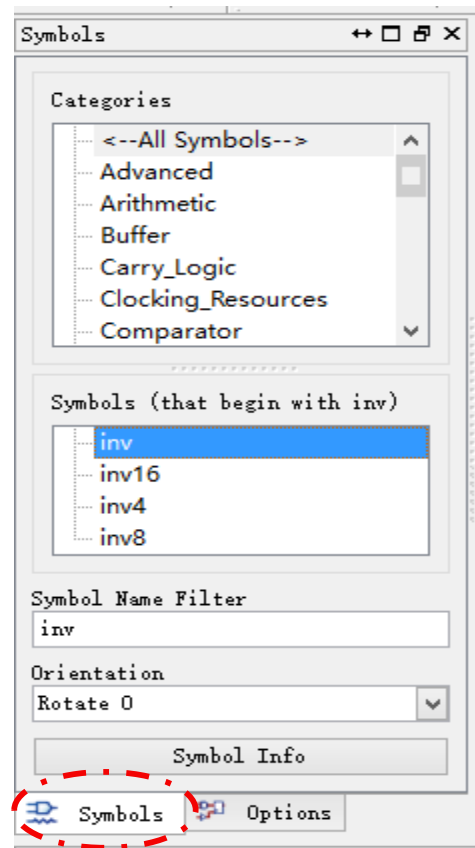
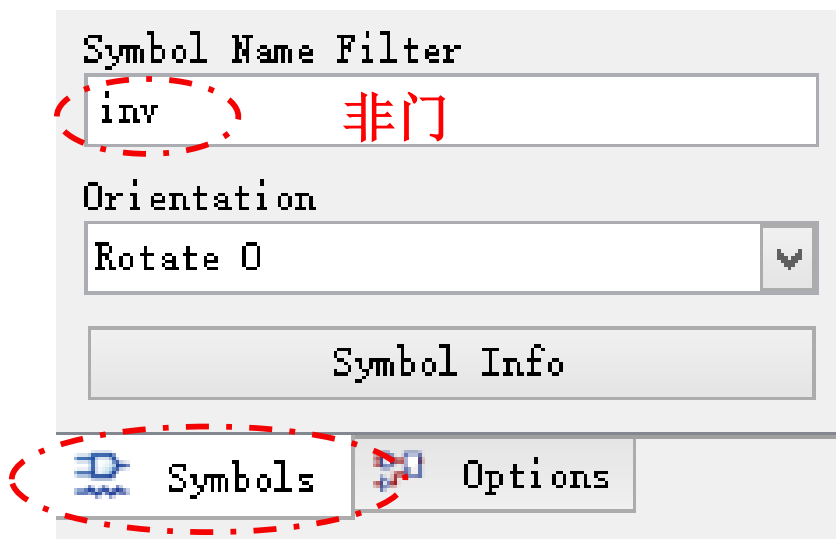
(2) 点击后，出现如图 窗口，选择schematic作为源类型、输入文件名lamp_Ctrl，确认Add to project被选中：



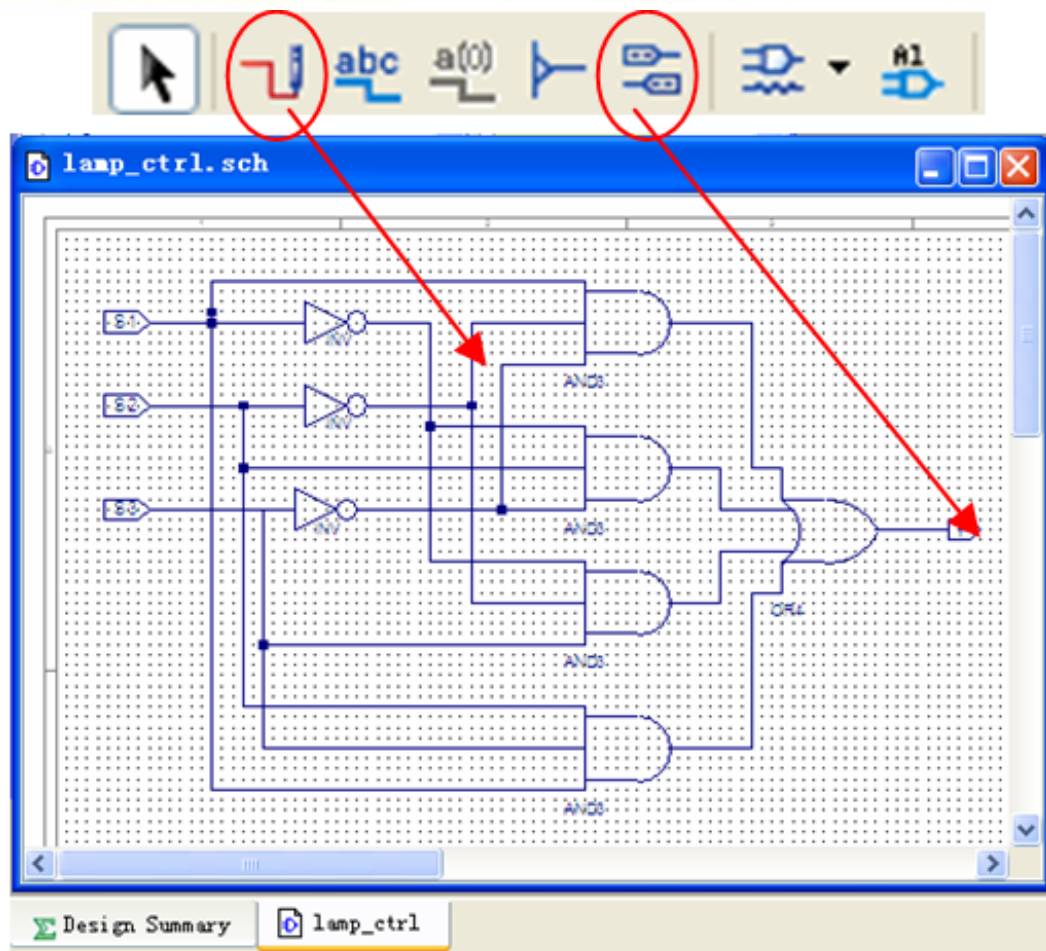
连续点击Next，接着点Finish出现如下编辑窗口：



- **【步骤3】** 用逻辑电路图描述输入楼道灯控逻辑电路，并进行编辑
 - 在Sources窗口中，点击右向箭头至Symbols界面，进入选择器件窗口

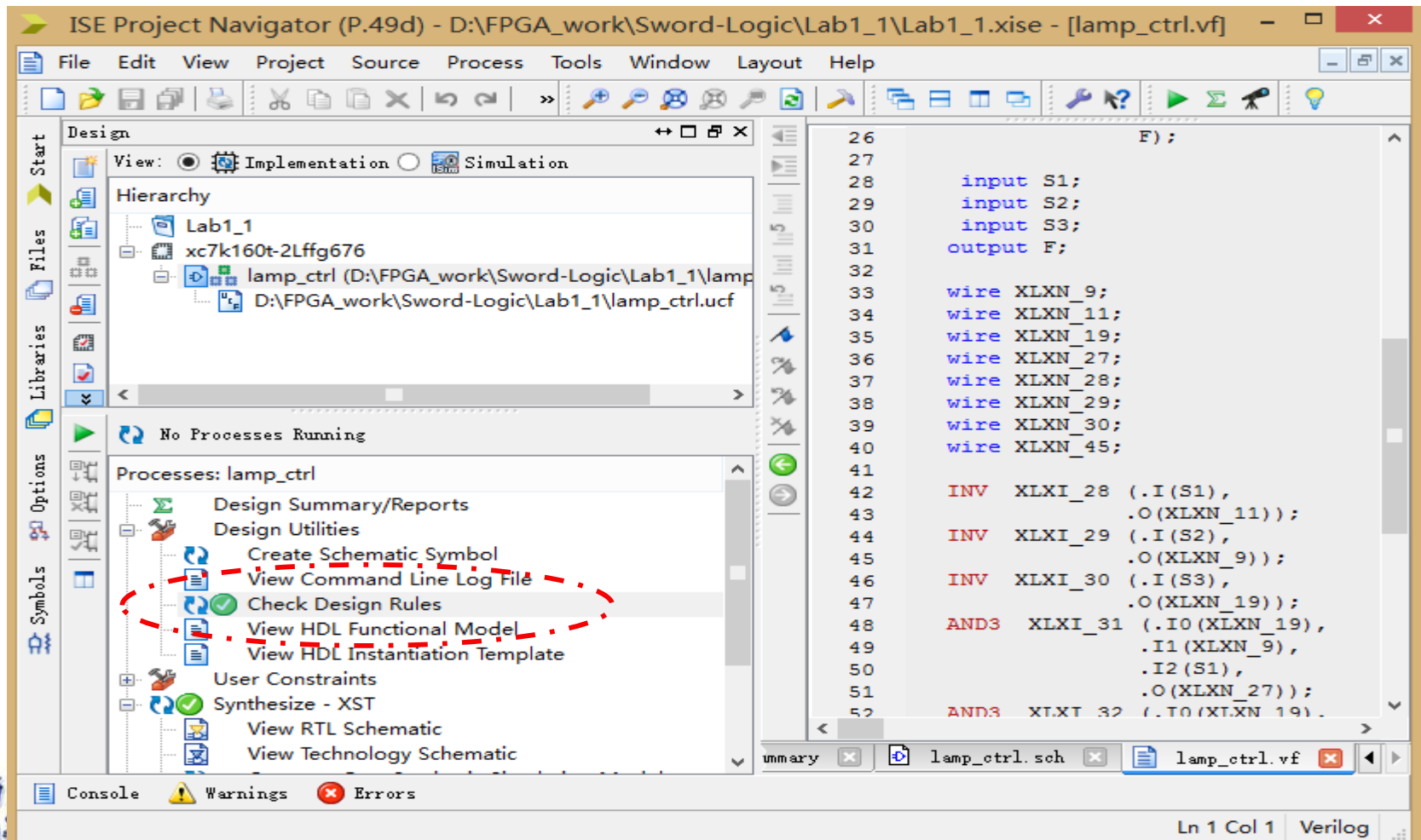


- 在器件窗口中选择所需的器件拖到图形输入窗口
- 当一个器件被安置后按下ESC键来取消，以便选择下一个器件
- 用导线(add wire)连接逻辑端口
- 对顶层输入/输出需要加上相应的缓冲：
add IO Marker(buffer和Iomake) 并修改命名
- 在Tools菜单选择Check Schematic可检查连线规则



- 楼道灯控逻辑电路输入完成后，无错误后保存

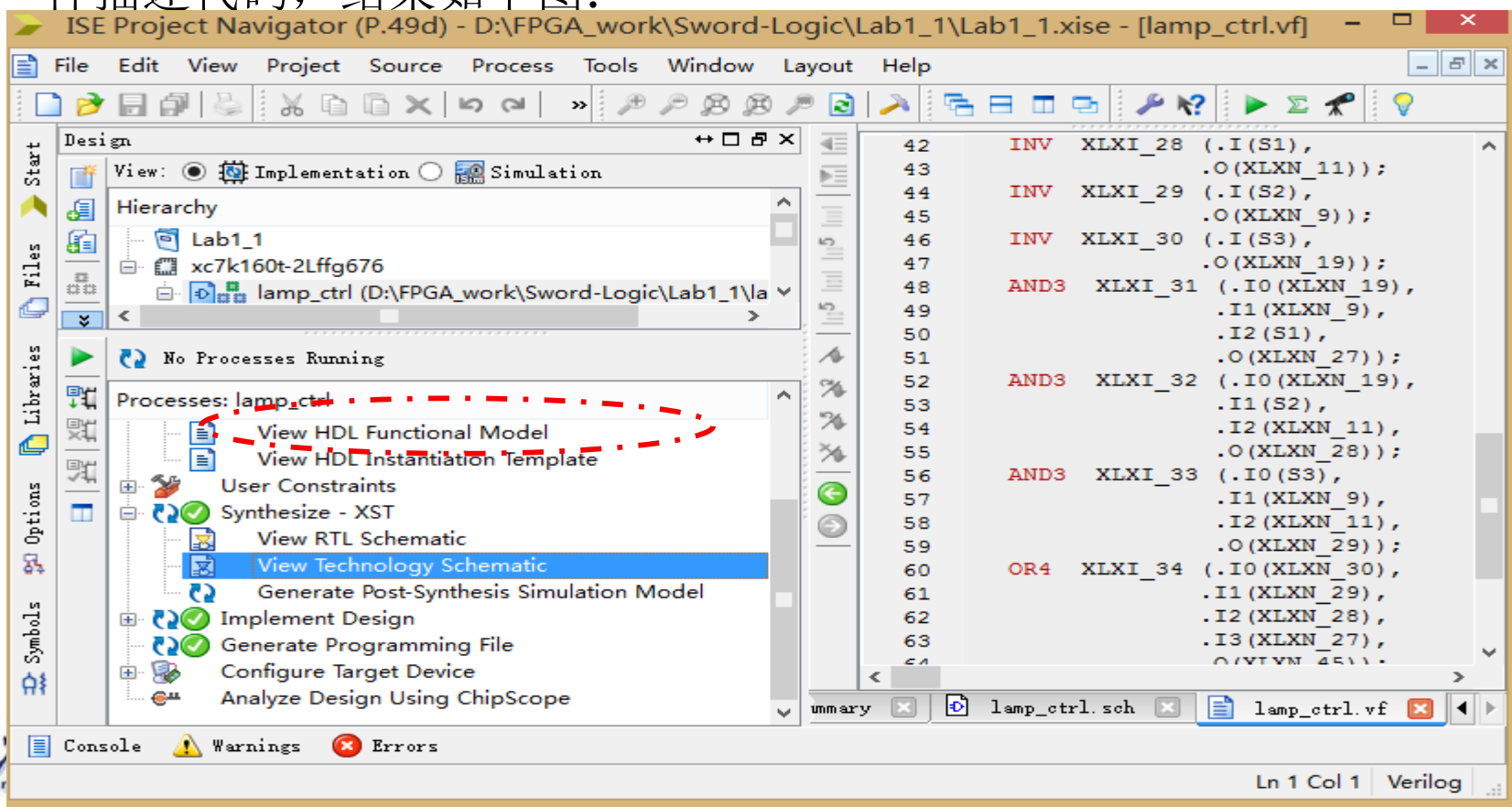
- 在Sources窗口，选择Synthesis/Implementation，并选中文件 lamp_Ctrl.sch；
- 在Processes窗口中用鼠标点击“+Design Utilities”的“+”号，展开菜单，并双击“Check Design Rules”，检查输入电路规则。若有错误排除输入错误



查看电路综合后是如何描述

□ 【步骤4】 查看分析输入电路的硬件描述代码

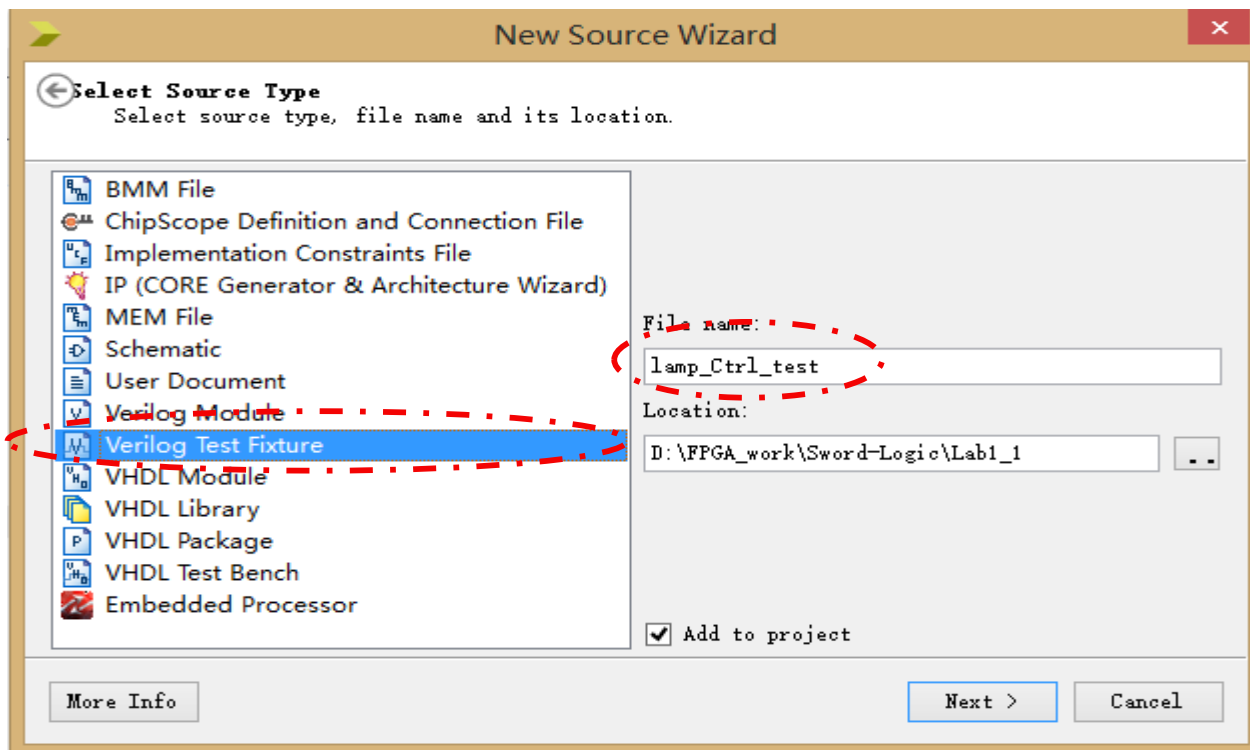
- 在Sources窗口，选择Synthesis，并选中文件lamp_Ctrl.sch
- 在Processes窗口中用鼠标点击“+Design Utilities”的“+”号，展开菜单，并双击“View HDL Functional Model”，可查看分析输入电路编译后的硬件描述代码，结果如下图：



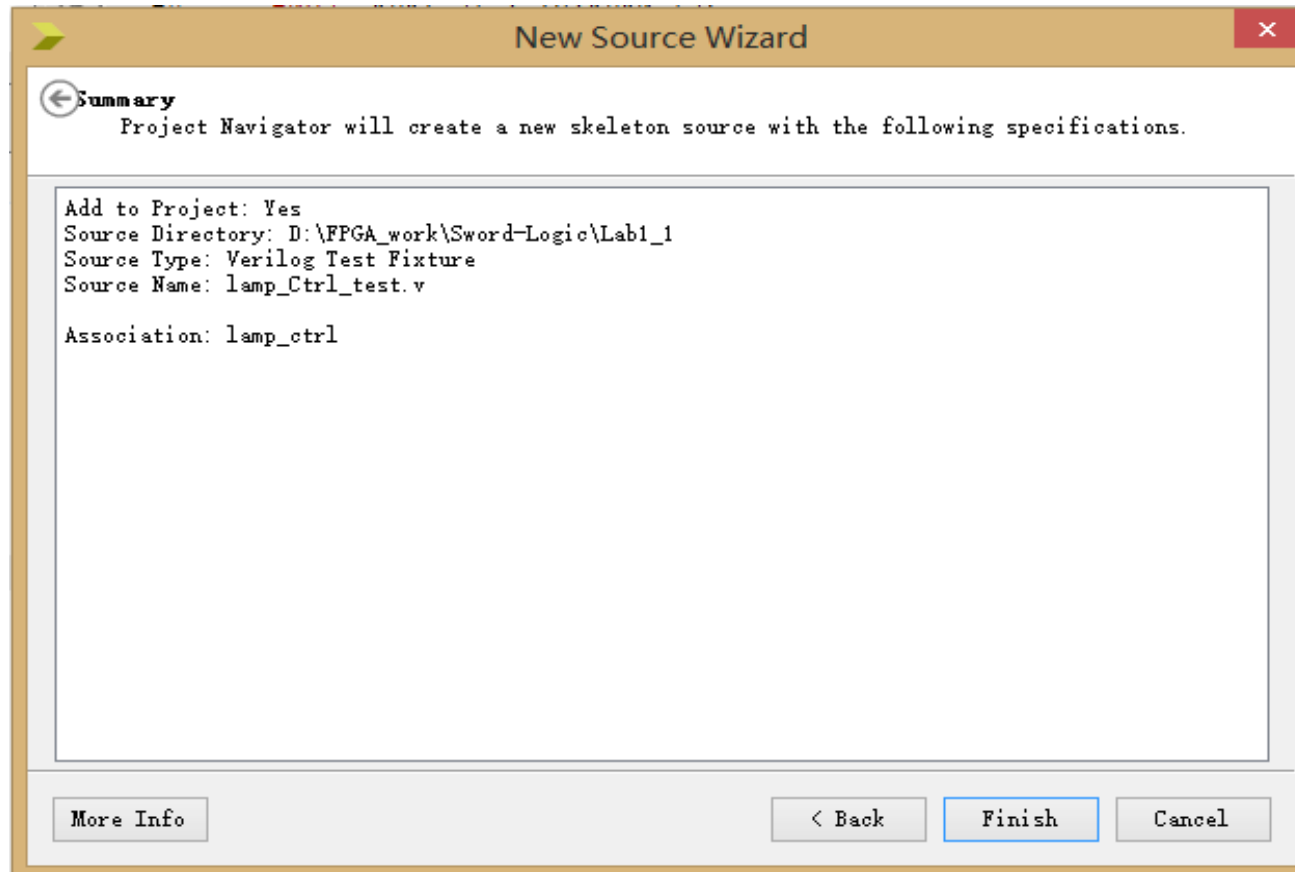
设计仿真激励代码

□ 【步骤5】 设计楼道灯控制电路的仿真激励代码： lamp_Ctrl_test.tbw

- (1) 在源窗口选择Verilog HDL(schematic)文件(要选中.v或.sch文件)。
- (2) 在Source窗口下空白处用鼠标右键点击New Source
- (3) 在New Source Type向导中, 选择Verilog Test Fixture作为源类型, 输入文件名: lamp_Ctrl_test到文件名框



(4) 点击Next，选择要仿真的模块，再点击Next弹出summary摘要窗口。点击Finish进入激励代码模板编辑窗口





(5) 设计激励代码。在激励代码模板编辑窗口输入激励代码

□ 参考代码:

.....

```
integer i;  
initial begin
```

```
    S1 = 0;
```

```
    S3 = 0;
```

```
    S2 = 0;
```

```
    for(i=0;i<=7;i=i+1)begin
```

```
        {S1,S2,S3} <= i;
```

```
    #20;
```

```
    end
```

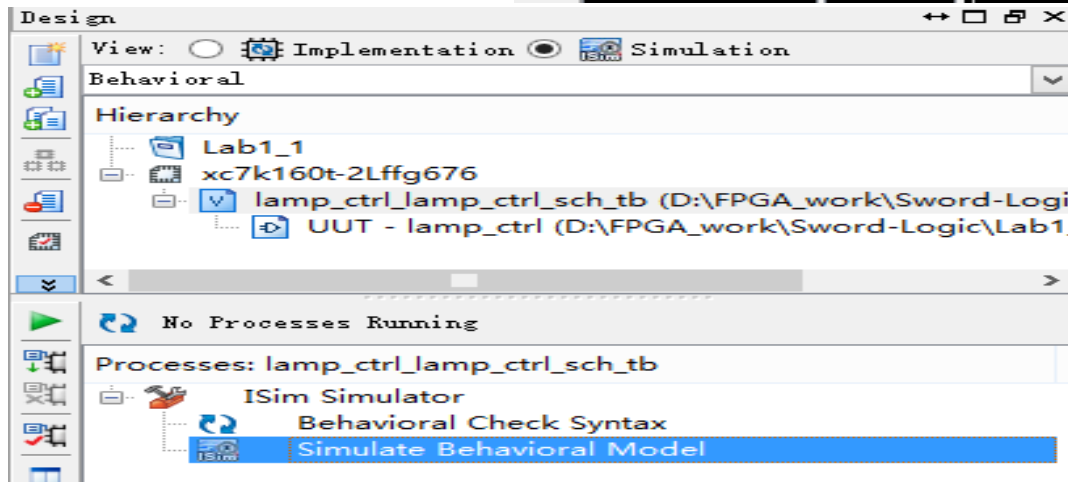
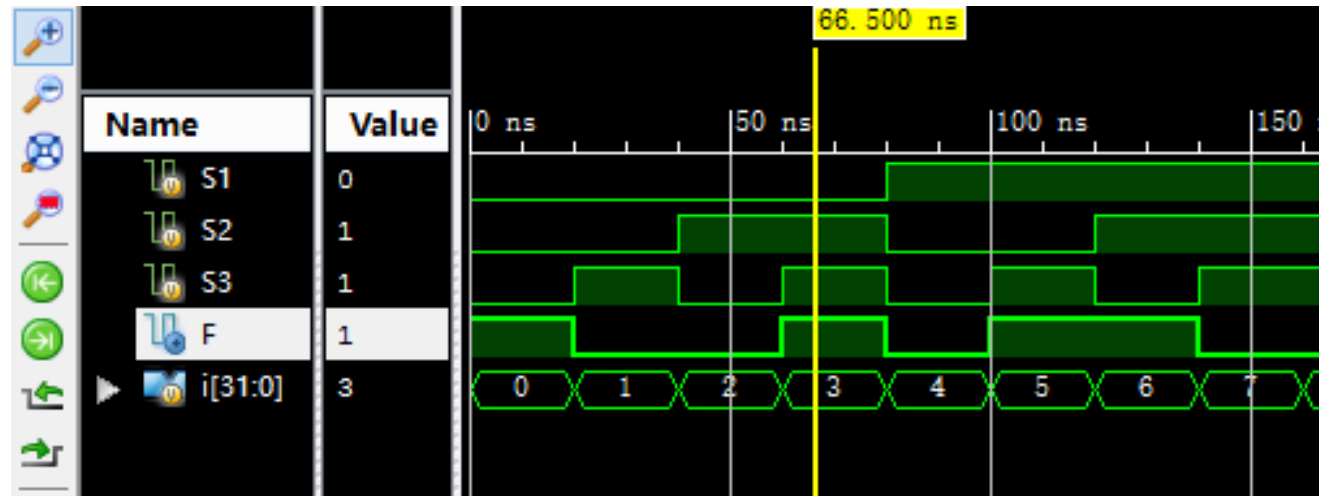
```
end
```

(6) 运行激励代码，作行为仿真(模拟)

在Design窗口选中Simulation

在Processes窗口双击Simulate Behavioral Model

仿真结果如下：





□ 【步骤6】 建立用户时序约束并为模块端口指定引脚分配

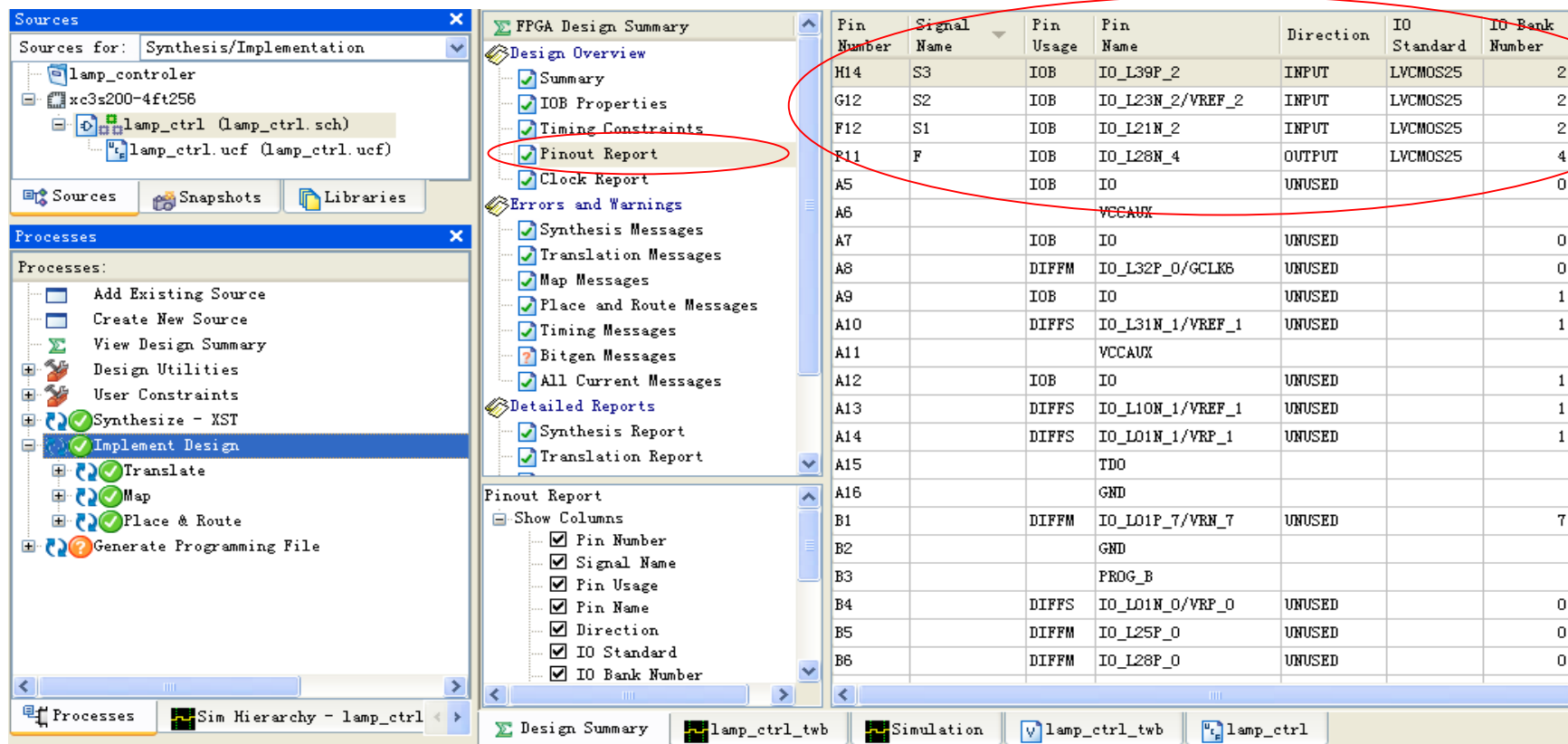
- 建立User Constraints files，输入顶层输出信号与芯片封装引脚关系
- 编辑完成后保存为lamp_Ctrl.ucf 文件
- 参考参数如下：

```
NET "S3" LOC = AE13    | IOSTANDARD = LVCMOS15 ;  
NET "S2" LOC = AF13    | IOSTANDARD = LVCMOS15 ;  
NET "S1" LOC = AF10    | IOSTANDARD = LVCMOS15 ;  
NET "F"  LOC = U21     | IOSTANDARD = LVCMOS18 ;
```

当端口信号非常多时，建议直接编辑UCF文件

□ 【步骤7】 设计实现并检查约束结果

- 在Source窗口下选Implementation;
- 在Processes窗口下选择→Implement Design。



The screenshot displays the Xilinx ISE software interface during the implementation phase. The 'Sources' window on the left shows the project files, including 'lamp_ctrl.sch' and 'lamp_ctrl.ucf'. The 'Processes' window on the left shows the 'Implement Design' process selected. The 'FPGA Design Summary' window in the center shows the 'Pinout Report' selected under 'Detailed Reports'. The 'Pinout Report' window on the right displays the pin configuration details, including Pin Number, Signal Name, Pin Usage, Pin Name, Direction, IO Standard, and IO Bank Number.

Pin Number	Signal Name	Pin Usage	Pin Name	Direction	IO Standard	IO Bank Number
H14	S3	IOB	IO_L39P_2	INPUT	LVCN0525	2
G12	S2	IOB	IO_L23N_2/VREF_2	INPUT	LVCN0525	2
F12	S1	IOB	IO_L21N_2	INPUT	LVCN0525	2
F11	F	IOB	IO_L28N_4	OUTPUT	LVCN0525	4
A5		IOB	IO	UNUSED		0
A6			VCCAUX			
A7		IOB	IO	UNUSED		0
A8		DIFFM	IO_L32P_0/GCLK6	UNUSED		0
A9		IOB	IO	UNUSED		1
A10		DIFFS	IO_L31N_1/VREF_1	UNUSED		1
A11			VCCAUX			
A12		IOB	IO	UNUSED		1
A13		DIFFS	IO_L10N_1/VREF_1	UNUSED		1
A14		DIFFS	IO_L01N_1/VRP_1	UNUSED		1
A15			TDO			
A16			GND			
B1		DIFFM	IO_L01P_7/VRN_7	UNUSED		7
B2			GND			
B3			PROG_B			
B4		DIFFS	IO_L01N_0/VRP_0	UNUSED		0
B5		DIFFM	IO_L25P_0	UNUSED		0
B6		DIFFM	IO_L28P_0	UNUSED		0

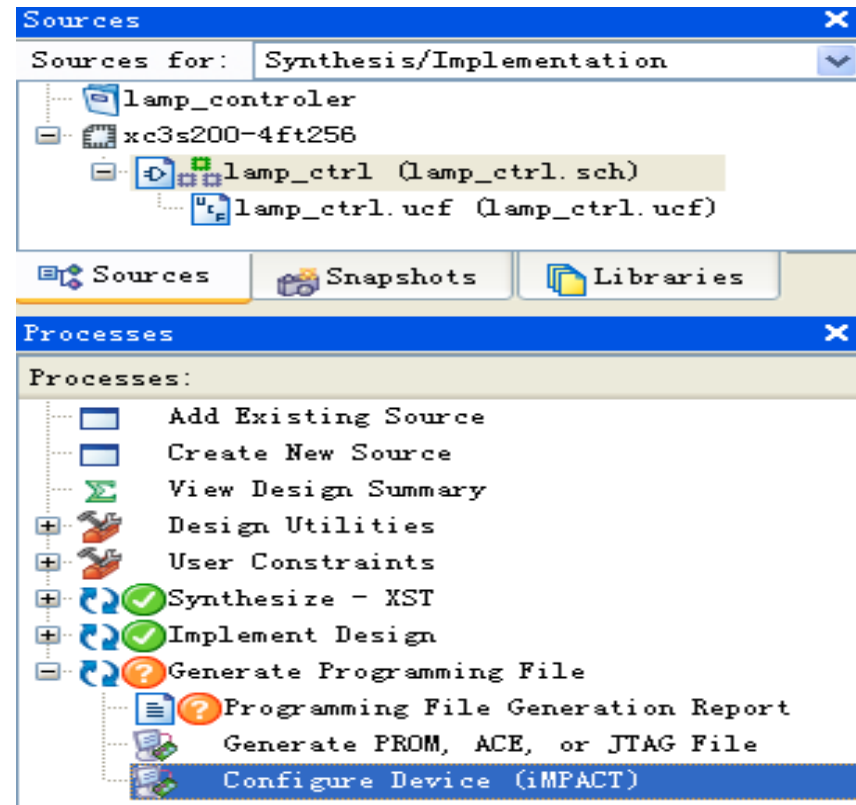
□ 【步骤8】 生成模块的FPGA流代码 并下载进行物理验证



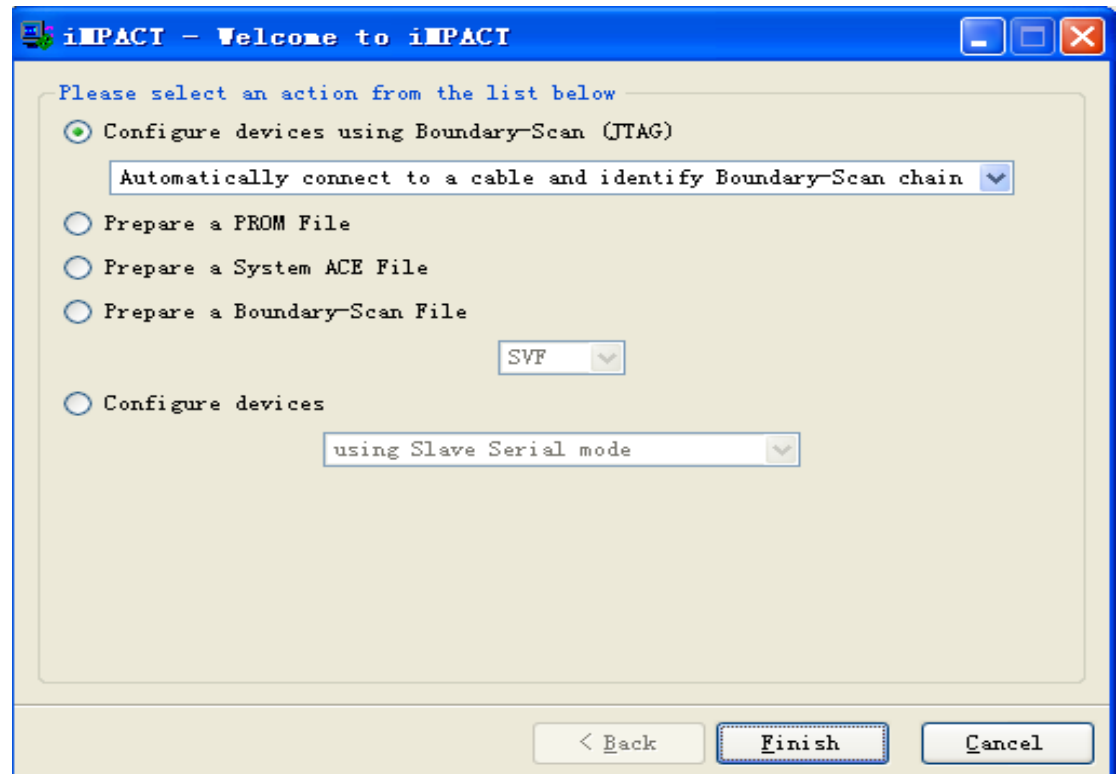
- 1) 用JTAG下载数据线将PC机USB口与SWORD4.0的“J11”端口连接。
- 2) 将12伏的DC电源连接到SWORD4.0的“CN1”电源插座
- 3) 打开SWORD4.0平台电源

下载步骤

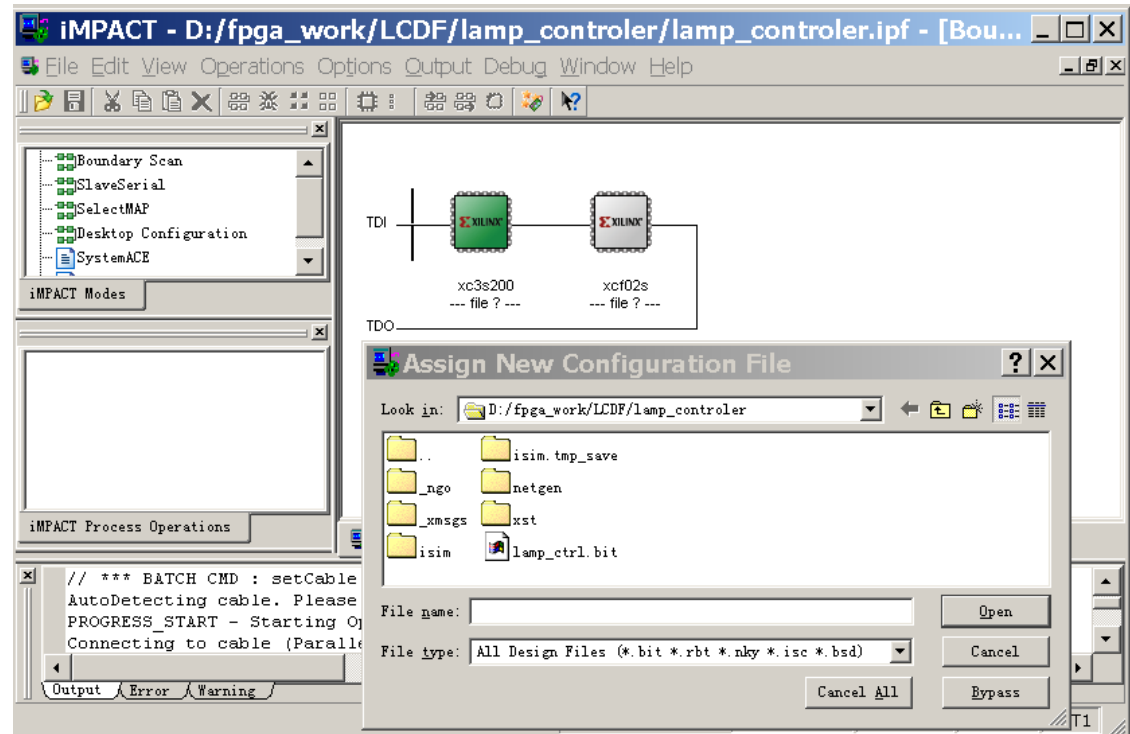
- 1) 从Sources窗口的选择**Implementation**
- 2) 在Sources窗口中选择**lamp_Ctrl.sch**
- 3) 在Processes窗口中，用鼠标点击
“+Generate Programming File”
- 4) 双击 **Configure Device (iMPACT)** 选项



- 4) 双击 Configure Device (iMPACT) 选项。出现如下图Configure Devices 对话框。
- 5) 在图 中选择**Configure devices using Boundary-Scan chain (JTAG)**。
- 6) 确认**Automatically connect to a cable and identify Boundary-Scan chain**已被选中。
- 7) 点击Finish按钮。

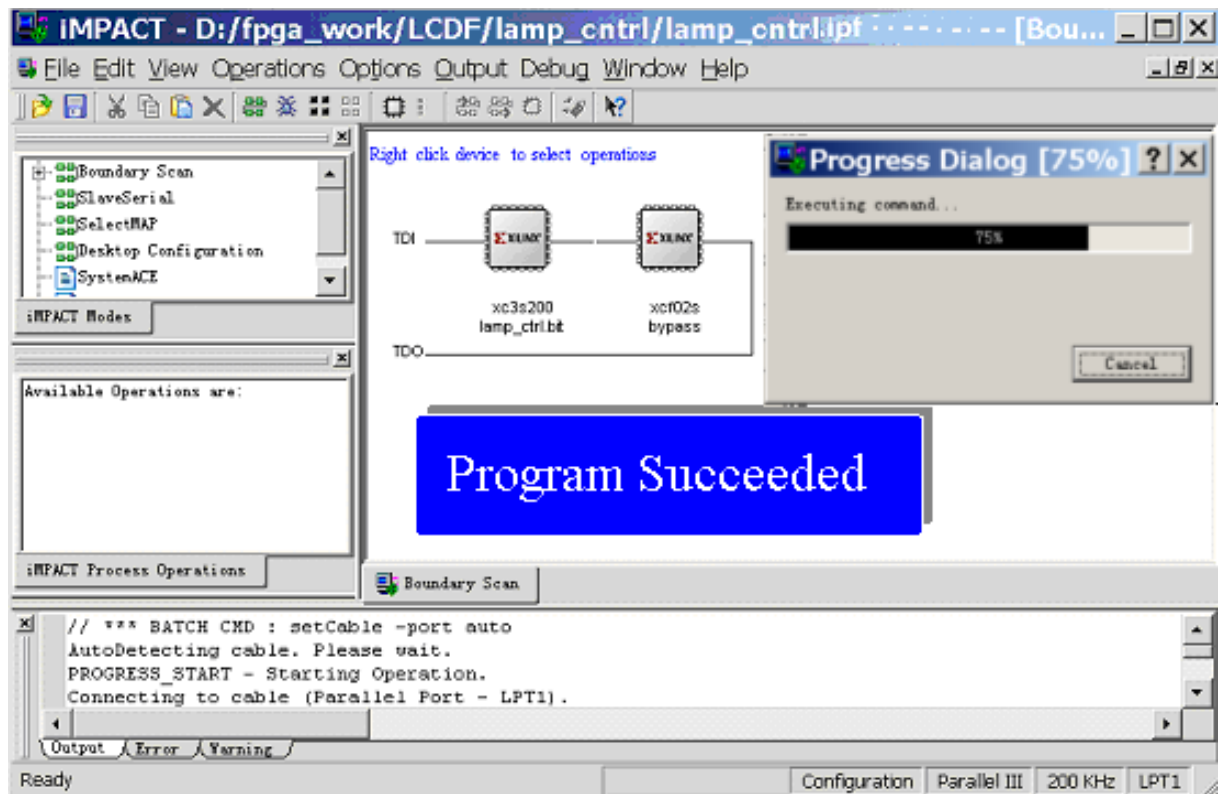


- 8) 如果跳出一个消息框显示“已找到两个设备”，请点击 **Ok**按钮继续。而连接到JTAG chain 的设备将在 iMPACT窗口中显示。
- 9) 接下来将会出现一个**Assign New Configuration File**对话框。这时从文件列表中选择**lamp_Ctrl.bit**文件并打开，将会为JTAG chain 上的xc3s200 设备指定配置文件





- 10) 如果有警告窗口弹出，点击 **OK** 按钮即可。
- 11) 选择 **Bypass** 以跳过其他余下的设备。
- 12) 鼠标右键点击xc7k325t device 的图标，然后在弹出菜单中选择 **Program...**，此时将会打开**Programming Properties**对话框。
- 13) 点击**OK**按钮可以对硬件设备进行下载编程，如下图：





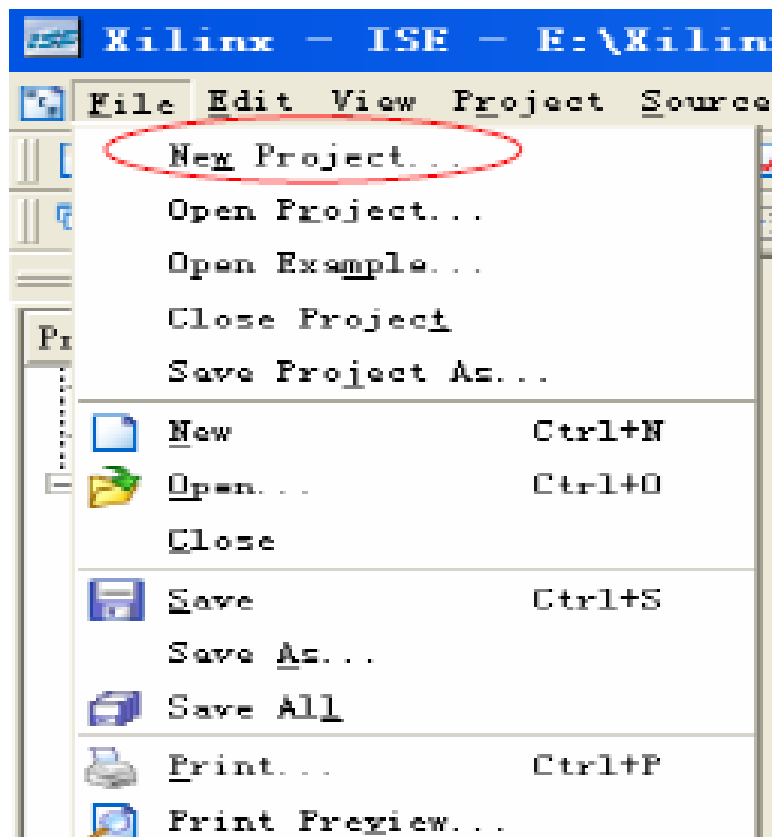
采用 Verilog HDL 输入逻辑功能描述

除设计描述输入不同其余完全类同

建立FPGA实现数字系统设计工程

□ 【步骤1】 Verilog描述方式建立楼道控制的工程文件：lamp_Controller.isc

(1) 点击左上方file，
选择new project新建
一个project。



(2) Project命名 :lamp_Ctrl.isc top-level source type选择:HDL



The image shows a screenshot of the 'New Project Wizard - Create New Project' dialog box. The dialog is divided into two main sections. The first section, 'Enter a Name and Location for the Project', contains two text boxes: 'Project Name' with the value 'lamp_Controller' and 'Project Location' with the value 'E:\Xilinx\Test01\lamp_Controller'. The second section, 'Select the Type of Top-Level Source for the Project', contains a dropdown menu labeled 'Top-Level Source Type' with 'HDL' selected. Below the dropdown, a list of options is visible: 'HDL', 'Schematic', 'EDIF', and 'NGC/NGO'. At the bottom of the dialog, there are four buttons: 'More Info', '< Back', 'Next >', and 'Cancel'.

New Project Wizard - Create New Project

Enter a Name and Location for the Project

Project Name: lamp_Controller

Project Location: E:\Xilinx\Test01\lamp_Controller

Select the Type of Top-Level Source for the Project

Top-Level Source Type:

HDL

HDL

Schematic

EDIF

NGC/NGO

More Info < Back Next > Cancel

(3) 点击Next到设备属性页

选择开发平台参数



Property Name	Value
Product Category	All
Family	Spartan3
Device	XC3S200
Package	FT256
Speed	-4
Top-Level Source Type	Schematic
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISE Simulator (VHDL/Verilog)
Enable Enhanced Design Summary	<input checked="" type="checkbox"/>
Enable Message Filtering	<input type="checkbox"/>
Display Incremental Messages	<input type="checkbox"/>

More Info < Back Next > Cancel

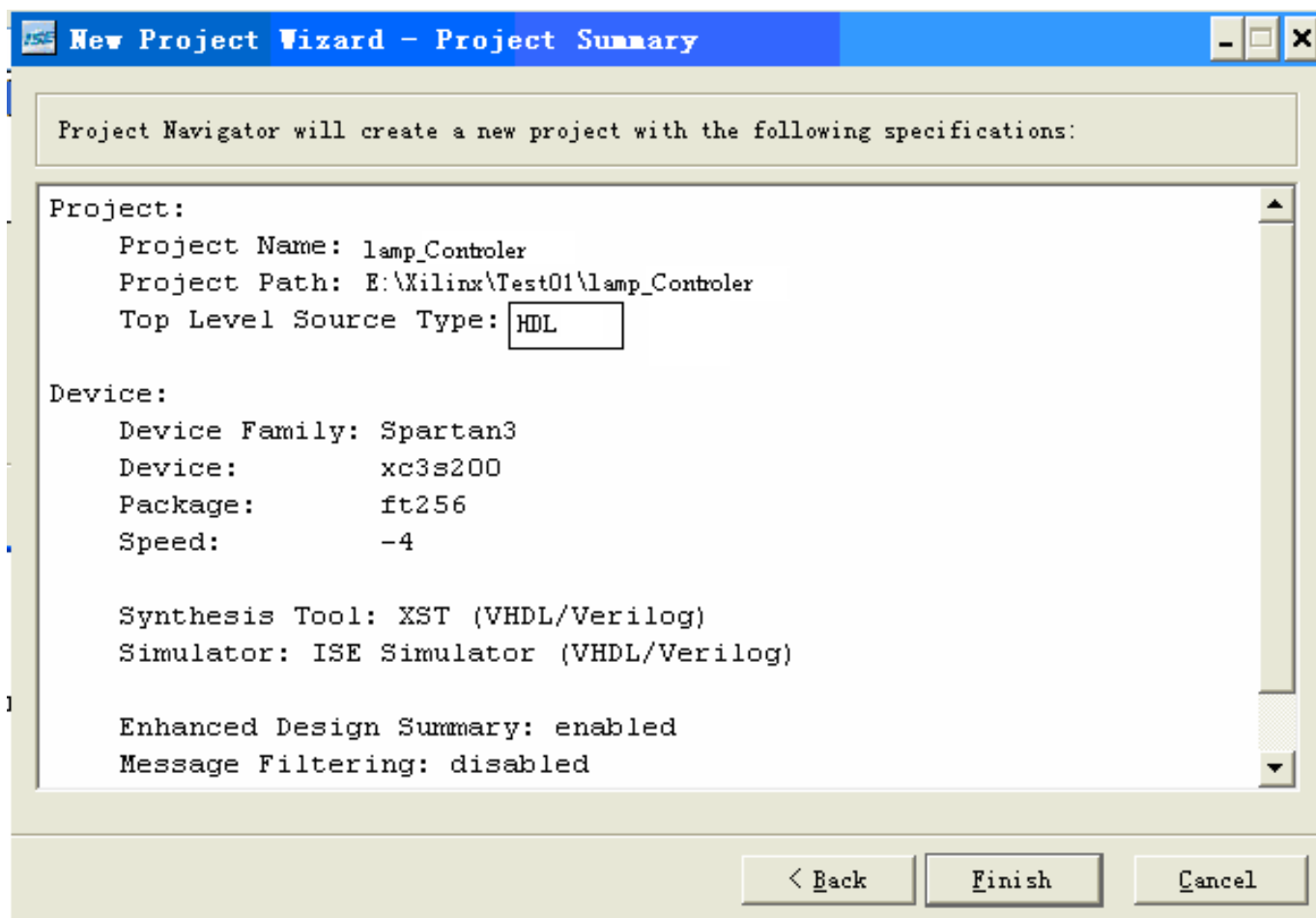
Family: Kintex-7

Device: XC7k325t

Package:FFG676

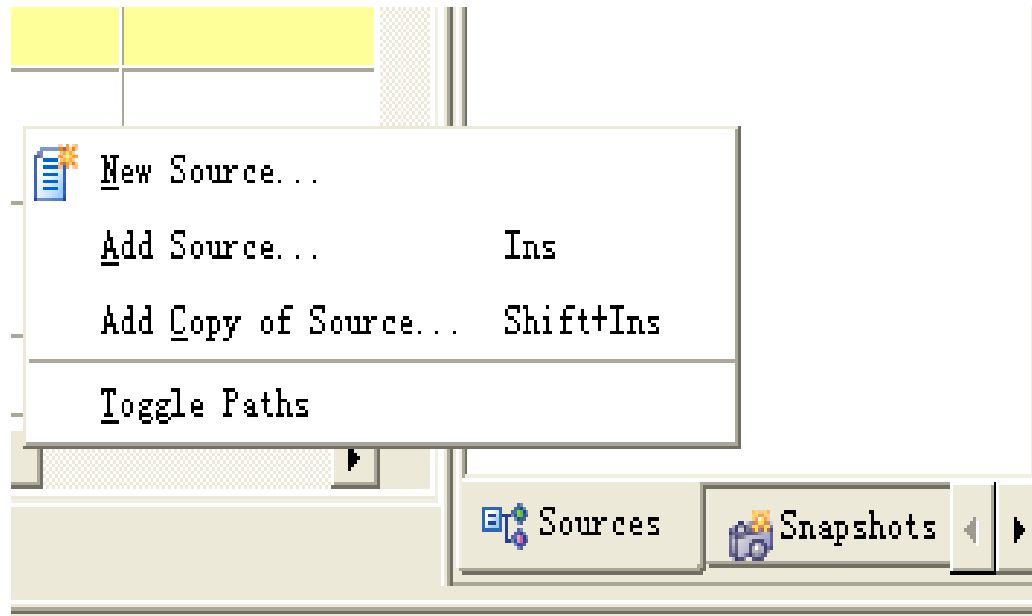
Speed:-2

设置完开发平台详细属性信息后，
点击NEXT，再点击NEXT进入

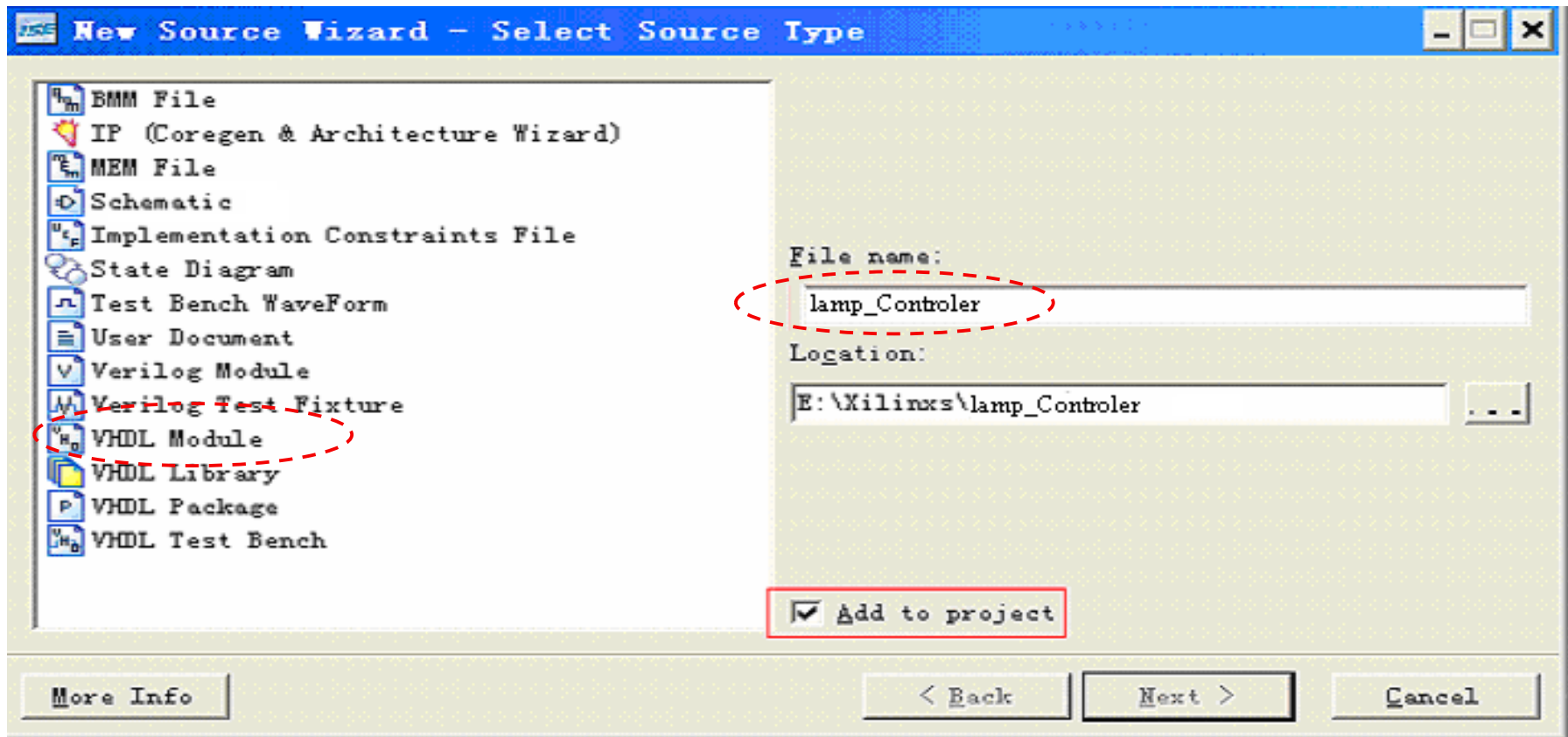


□ 【步骤2】 创建VerilogHDL输入模板及源文件：lamp_Controller.v

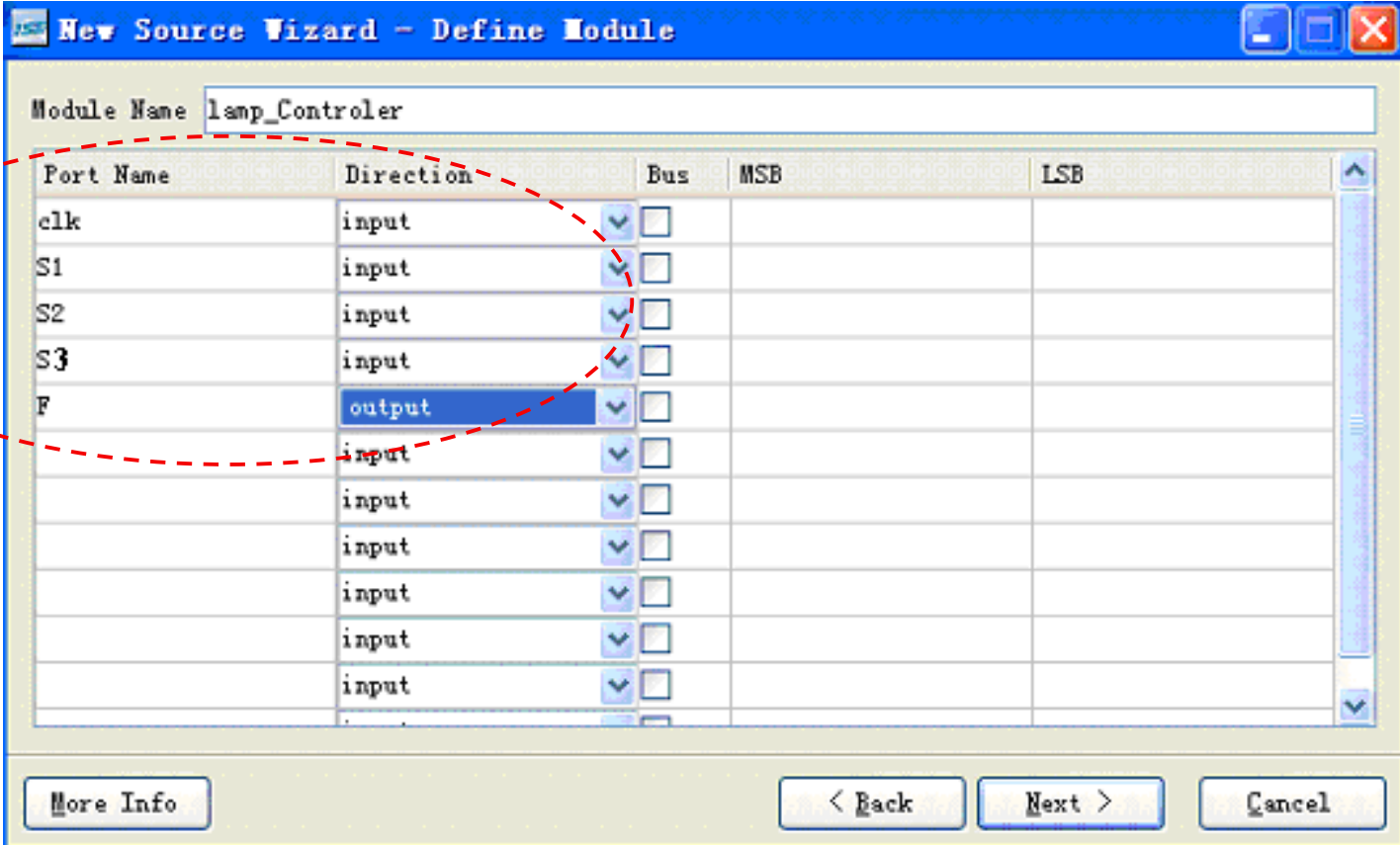
(1) 在Source窗口下空白处用鼠标右键点击New Source



(2) 点击后，在对话框，选择Verilog Modul作为源类型、输入文件名**lamp_Controller.v**、确认Add to project被选中



连续点击Next，接着点Finish出现如下编辑窗口：
定义输入输出端口引脚名(Port Name)



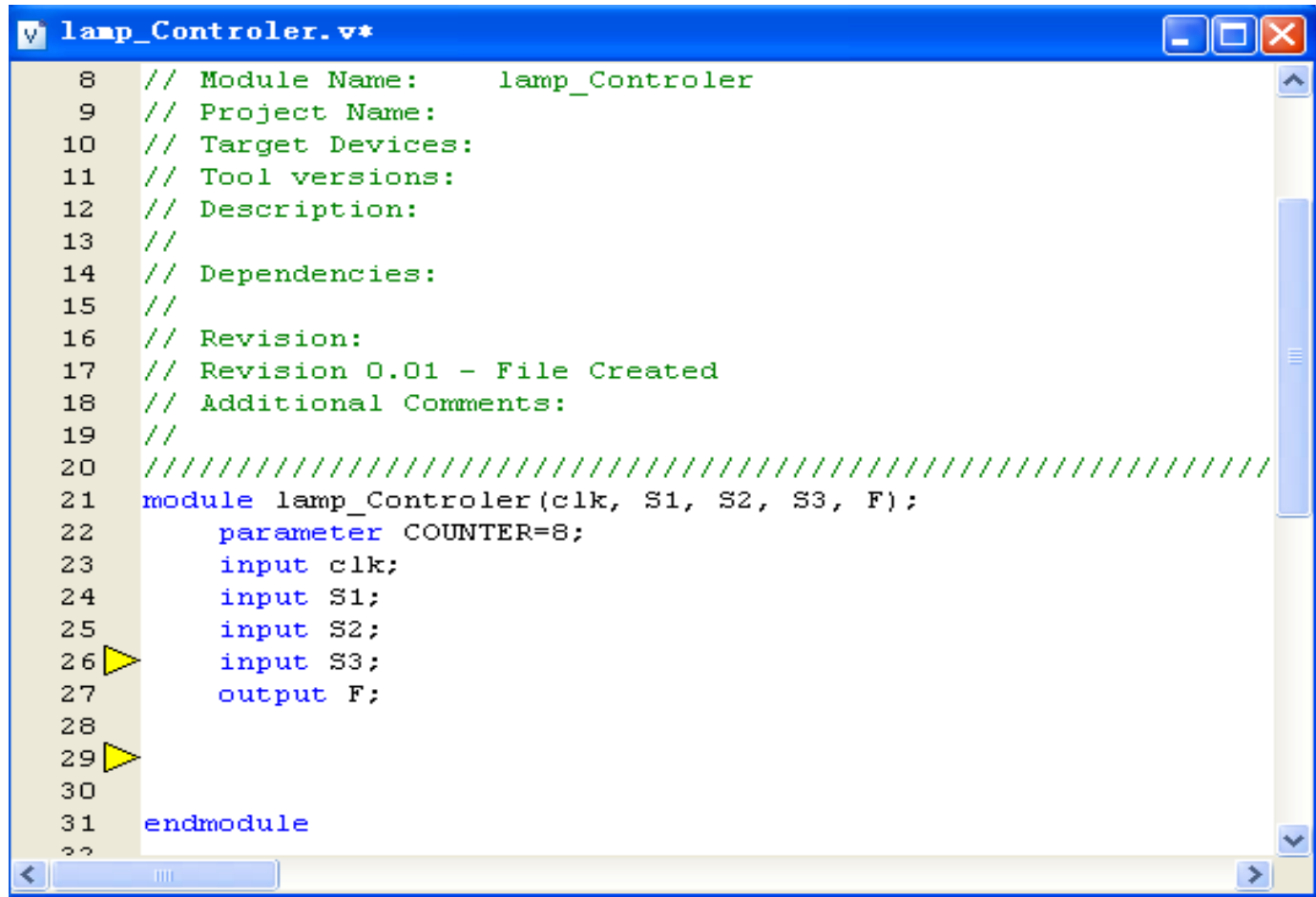
The dialog box titled "New Source Wizard - Define Module" shows the "Module Name" field set to "lamp_Controller". Below it is a table for defining ports. A red dashed oval highlights the first five rows of the table. The table has columns for "Port Name", "Direction", "Bus", "MSB", and "LSB".

Port Name	Direction	Bus	MSB	LSB
clk	input	<input type="checkbox"/>		
S1	input	<input type="checkbox"/>		
S2	input	<input type="checkbox"/>		
S3	input	<input type="checkbox"/>		
F	output	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		
	input	<input type="checkbox"/>		

At the bottom of the dialog, there are buttons for "More Info", "< Back", "Next >", and "Cancel".

此时可以不填写，在生成模块编辑模板中输入
计算机学院 系统结构与系统软件实验室

点击Next，然后在New SourceInformation对话框点击Finish，完成New Source File Verilog HDL模板的创建



```
8 // Module Name:      lamp_Controller
9 // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21 module lamp_Controller(clk, S1, S2, S3, F);
22     parameter COUNTER=8;
23     input  clk;
24     input  S1;
25     input  S2;
26     input  S3;
27     output F;
28
29
30
31 endmodule
32
```

□ 【步骤3】 用VerilogHDL描述方式输入 楼道灯控逻辑电路代码



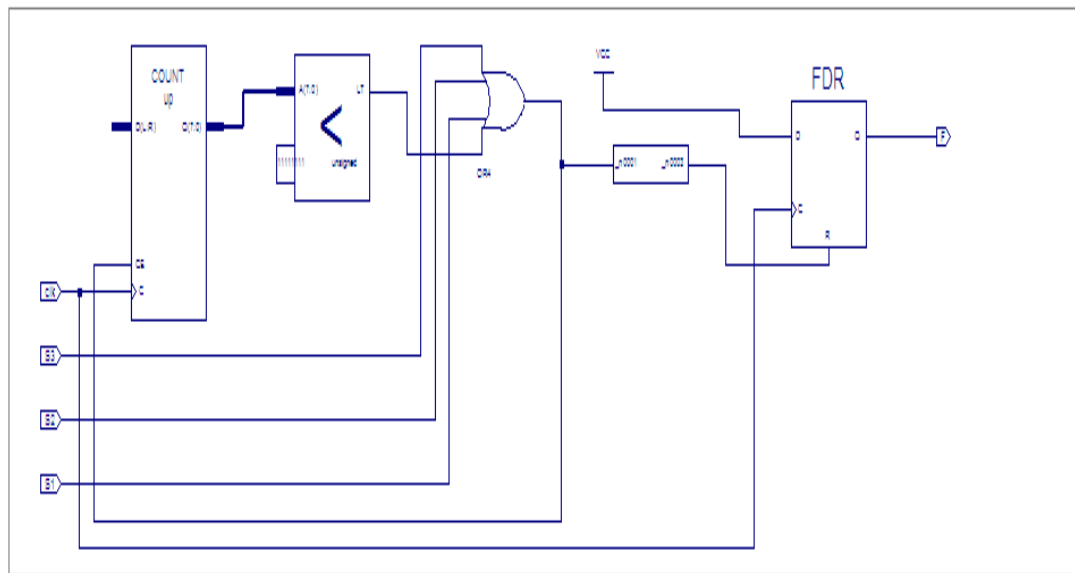
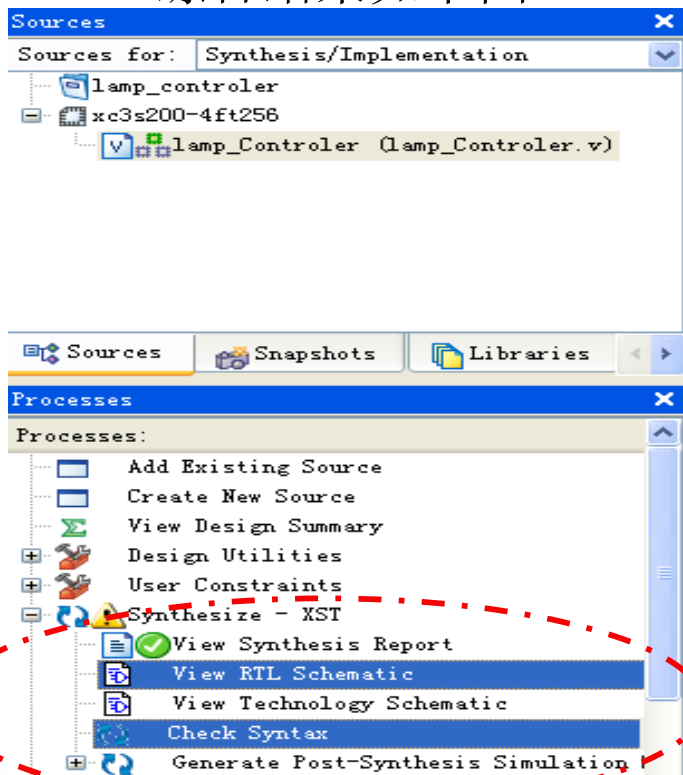
□ 进行编辑

- 在代码编辑窗口中用Verilog HDL语言，输入，编辑电路逻辑功能。电路代码输入完后保存编辑文件。

```
lamp_Controller.v*
21 module lamp_Controller(clk, S1, S2, S3, F);
22     parameter COUNTER=8;
23     input clk;
24     input S1;
25     input S2;
26     input S3;
27     output F;
28     wire w;
29     reg y;
30     reg [COUNTER-1:0] count;
31
32     initial
33         count<=0;
34
35     assign w=S1||S2||S3;
36     always@(posedge clk)
37         if(w || count < 8'hFFFF)
38             begin
39                 y=1;
40             end
41         else
42             y=0;
43     always@(posedge clk)
44         if(w || count<8'hFFF)
45             count<=count+1;
46         else
47             count<=count;
48     assign F=y;
49 endmodule
--
```

□ 【步骤4】 楼道控制电路代码综合

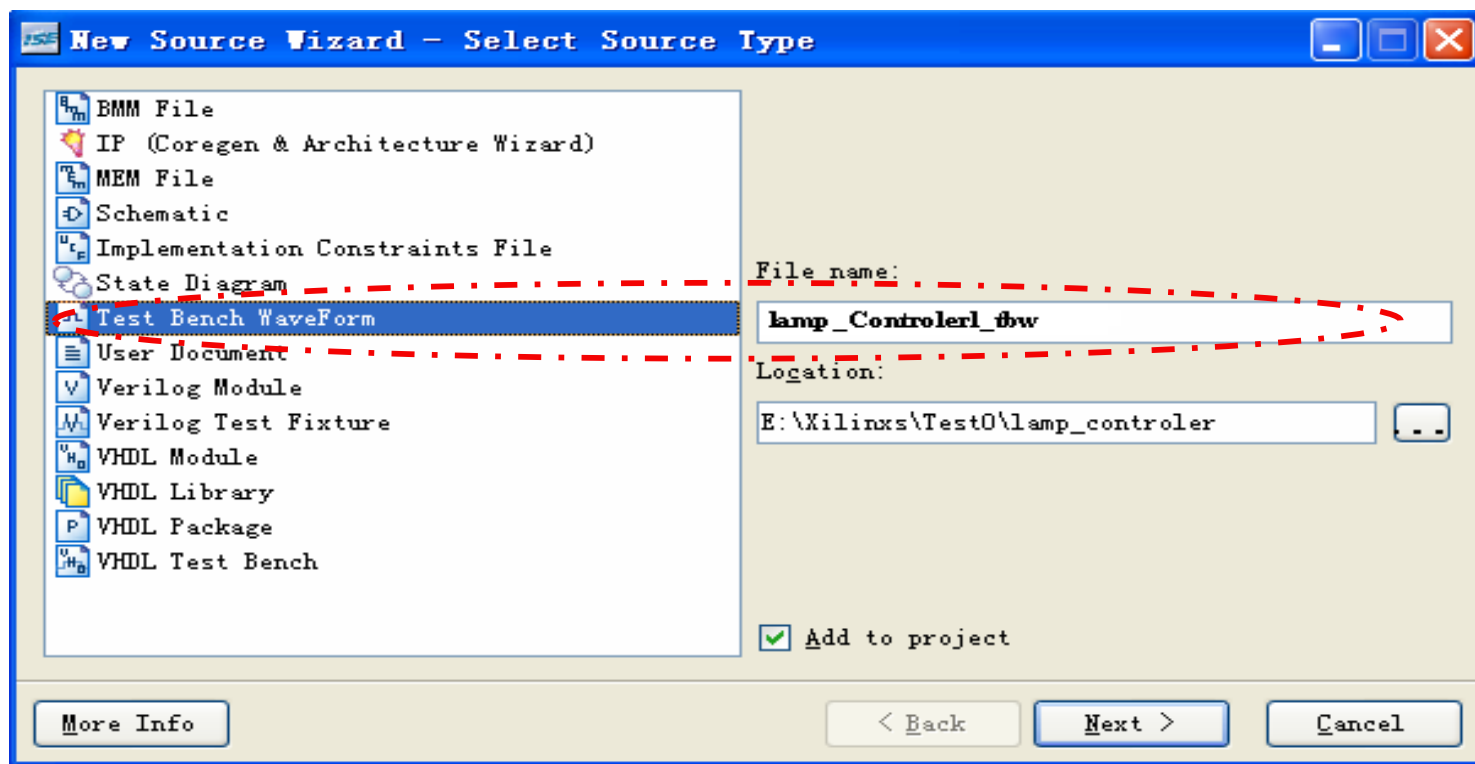
- 在Sources窗口，选择Synthesis/Implementation，并选中文件 **lamp_Controller.v**;
- 在Processes窗口中用鼠标点击 “+Synthesize-XST” 的 “+” 号，展开菜单，并双击 “View RTL Schematic”
- 综合结果如下图:



□ 【步骤5】 建立楼道灯控制电路的基准测试波形文件: lamp_Controller_tbw.tbw

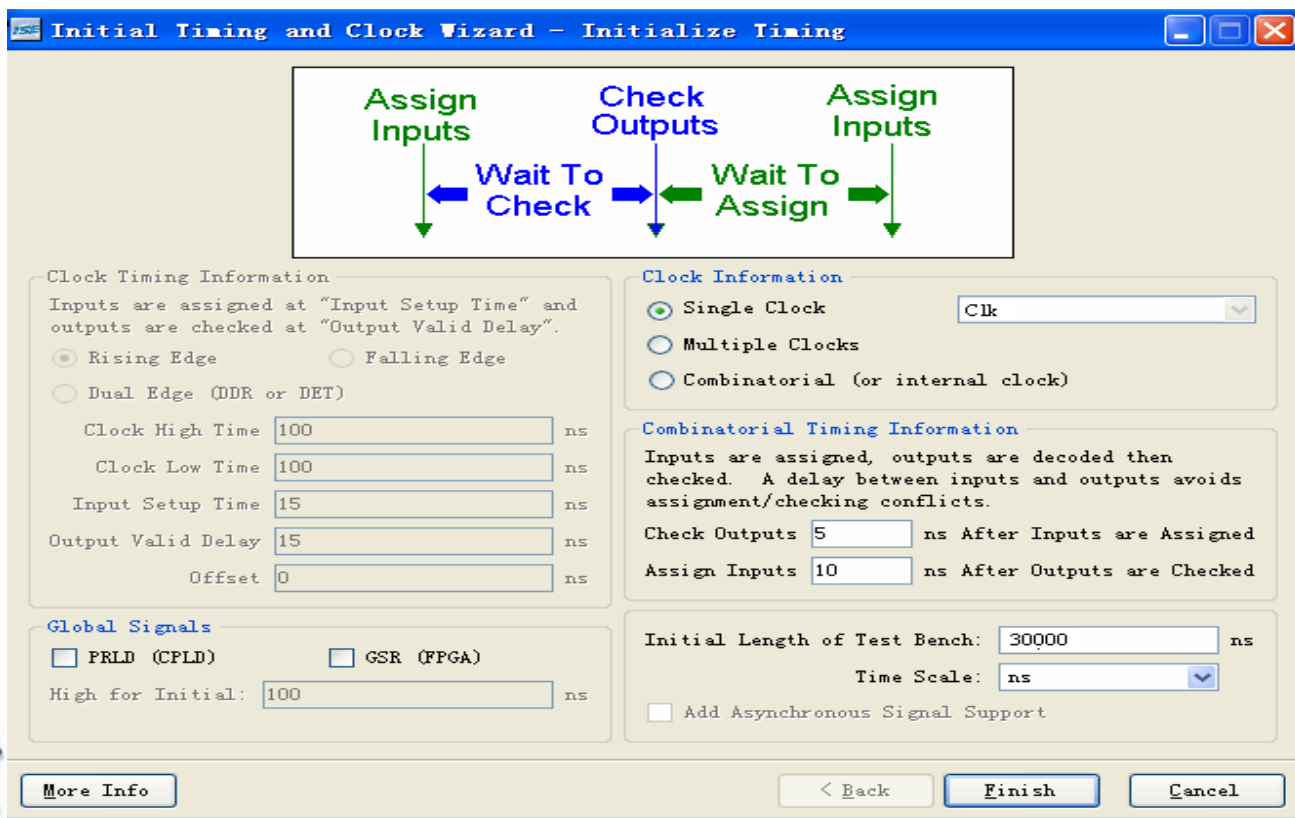


- (1) 在源窗口选择Verilog HDL文件(要选中.v或.sch文件)。
- (2) 在Source窗口下空白处用鼠标右键点击New Source如。所示。
- (3) 在New Source Type向导中, 选择Test Bench WaveForm作为源类型, 输入文件名: lamp_Controller_tbw到文件名框



(4) 点击Next，出现如下图所示，设置时间参数。

- 在这里根据电路的器件类型或电路的要求选择触发类型(上升沿触发、下降沿触发、双边沿触发)。
- 需要设置时钟频率、启动时间和输出延时
- 那个输入信号设置作为基准 single clock
- 本示例电路是时序电路，在Clock Information设为Single Clock，其余选项如图所示。点击finish，系统弹出Test Bench WaveForm编辑窗口。



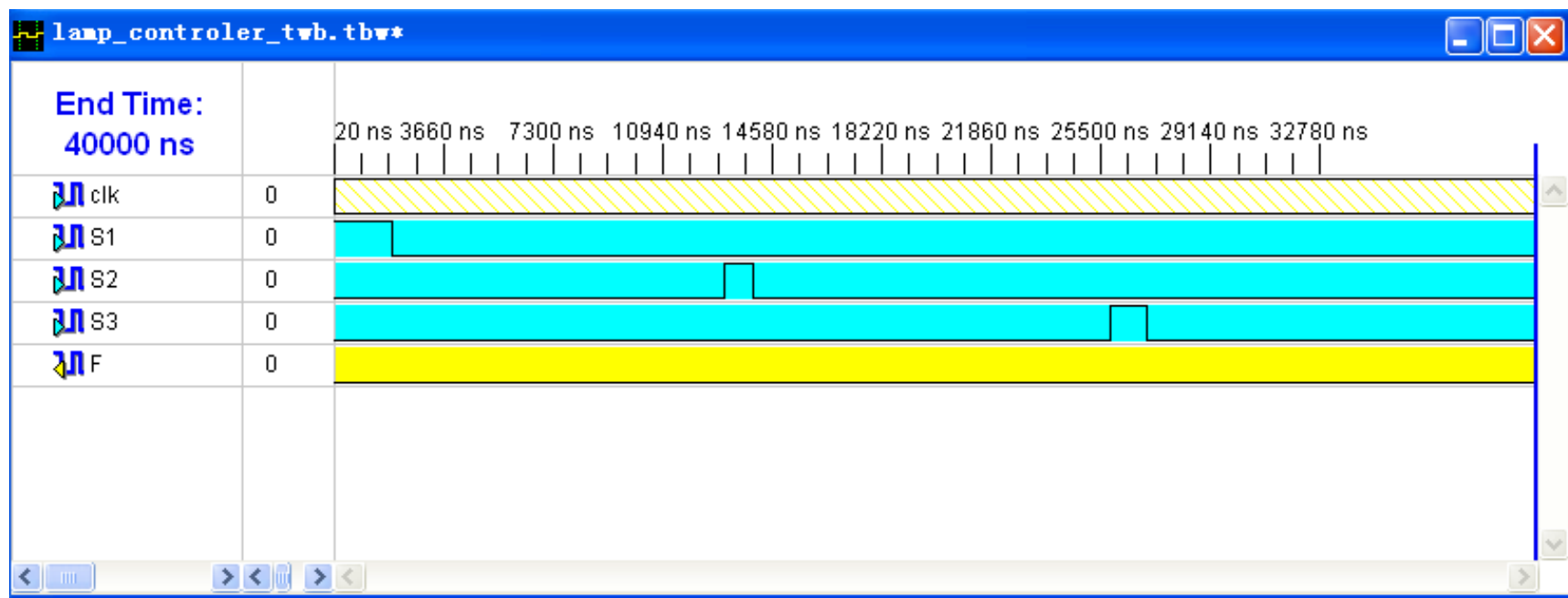
The dialog box titled "Initial Timing and Clock Wizard - Initialize Timing" contains the following sections:

- Process Flow Diagram:** A central diagram showing the sequence: Assign Inputs (green arrow) → Wait To Check (blue arrow) → Check Outputs (blue arrow) → Wait To Assign (green arrow) → Assign Inputs (green arrow).
- Clock Timing Information:**
 - Inputs are assigned at "Input Setup Time" and outputs are checked at "Output Valid Delay".
 - ☒ Rising Edge ☐ Falling Edge
 - ☐ Dual Edge (QDR or DET)
 - Clock High Time: 100 ns
 - Clock Low Time: 100 ns
 - Input Setup Time: 15 ns
 - Output Valid Delay: 15 ns
 - Offset: 0 ns
- Clock Information:**
 - ☒ Single Clock (dropdown: Clk)
 - ☐ Multiple Clocks
 - ☐ Combinatorial (or internal clock)
- Combinatorial Timing Information:**
 - Inputs are assigned, outputs are decoded then checked. A delay between inputs and outputs avoids assignment/checking conflicts.
 - Check Outputs: 5 ns After Inputs are Assigned
 - Assign Inputs: 10 ns After Outputs are Checked
- Global Signals:**
 - ☐ PRLD (CPLD) ☐ GSR (FPGA)
 - High for Initial: 100 ns
- Test Bench Settings:**
 - Initial Length of Test Bench: 30000 ns
 - Time Scale: ns
 - ☐ Add Asynchronous Signal Support
- Buttons:** More Info, < Back, Finish, Cancel

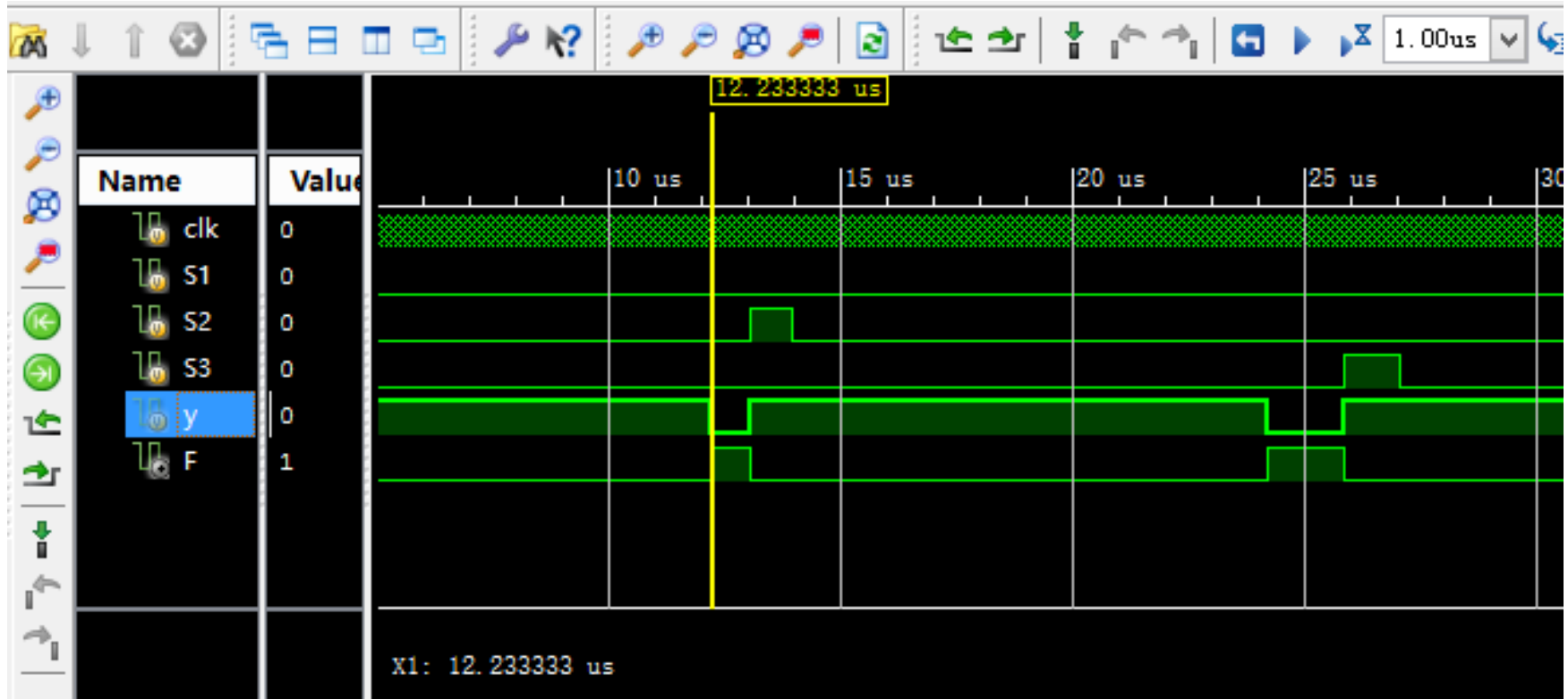


□ 【步骤6】 建立仿真激励输入波形，并运行行为模拟”，验证电路功能

- (1) 在Test Bench WaveForm编辑窗口中，通过点击左键改变输入端的高低电平。根据设计要求，这里用真值表的输入取值设置S1、S2、S3为脉冲。输入激励设置并保存，它的文件扩展名为.tbw。



(2) 进行波形行为仿真(模拟)，结果如下：

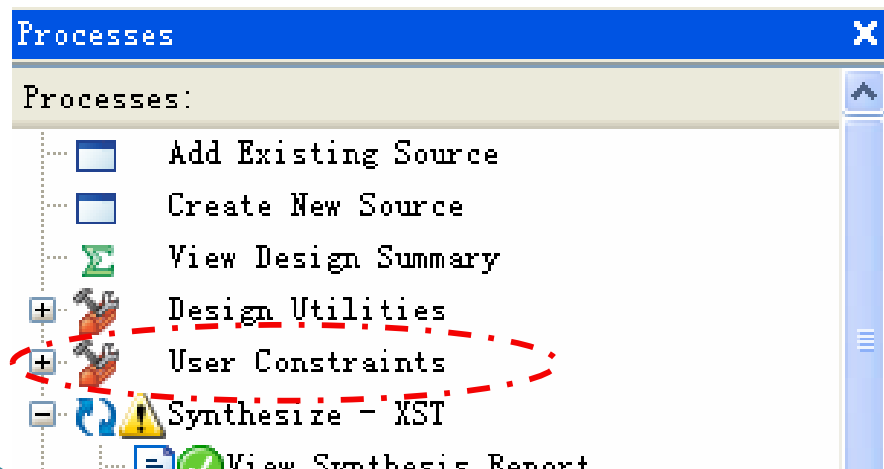
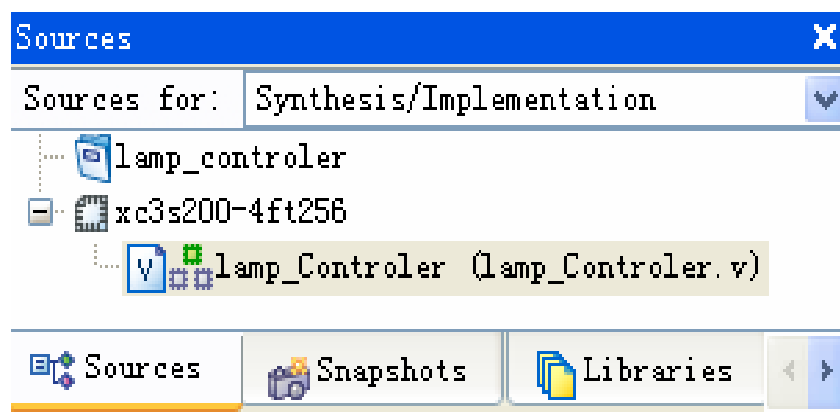


注意S1、S2、S3的脉冲间隔

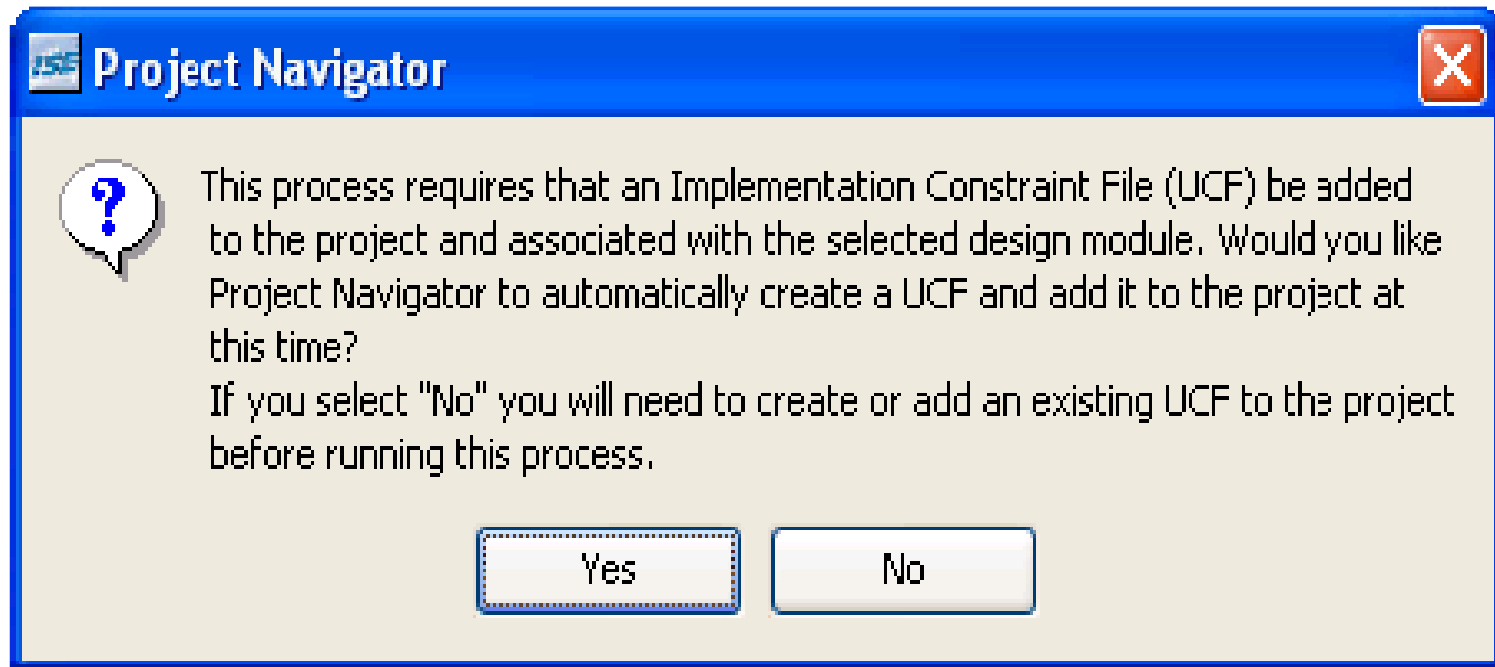


【步骤7】建立用户时序约束并为模块的端口指定引脚分配

- 本示例是时序电路首先要做时钟约束，然后再做电路的FPGA引脚约束。
- 时序约束(Entering Timing Constraints)过程如下：
 1. 在Sources窗口，选择Synthesis/Implementation并选中**lamp_Controller1** HDL源文件。
 - 在processes窗口中点击User Constraints processes group 边上的“+”符号，双击**Create Timing Constraints** 运行建立时间约束

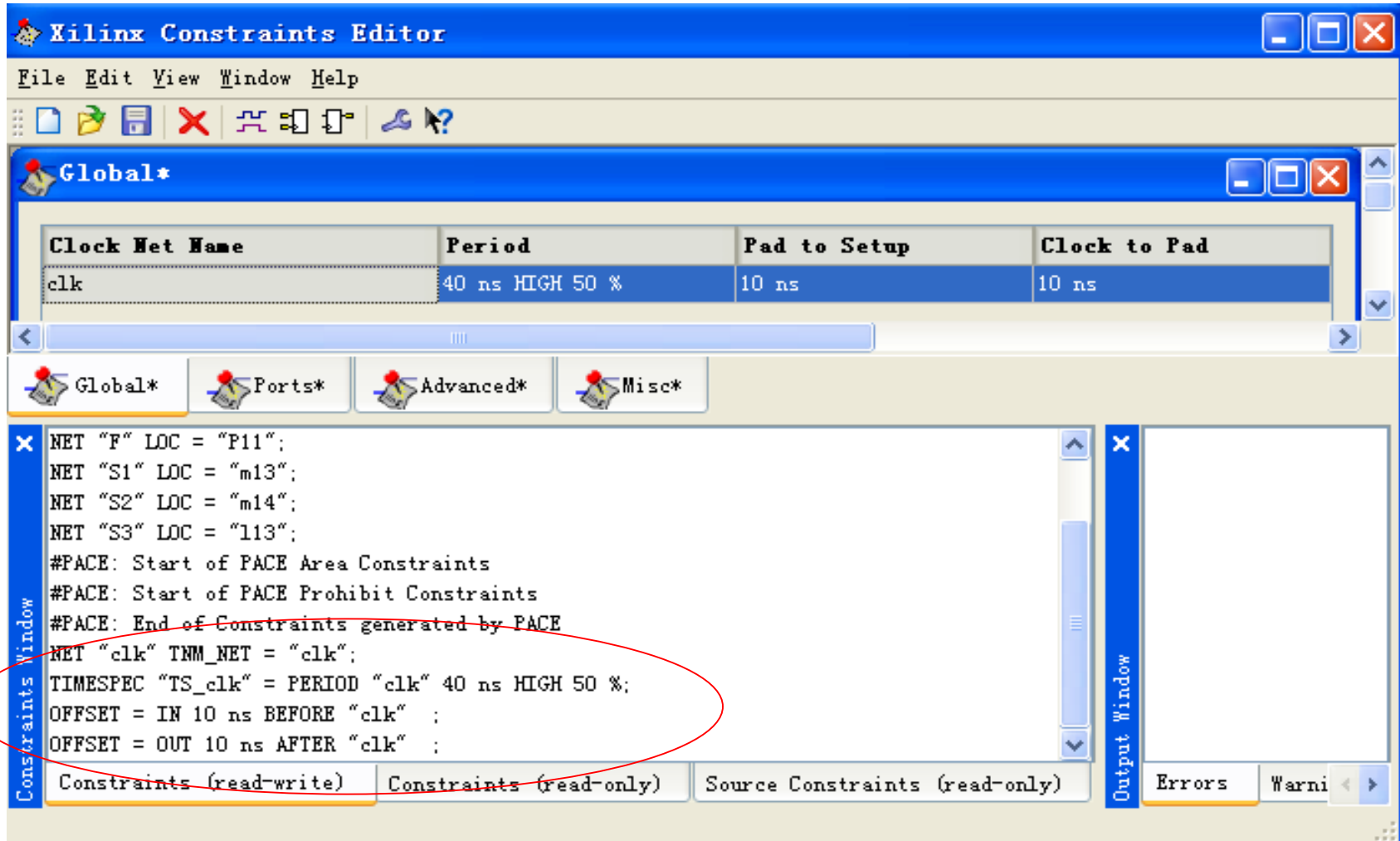


- ISE 运行编译、综合后自动生成用户约束文件(User Constraints File (UCF))并将得到如图 2-40的提示信息（**注意:**你也可以通过选择**Project-> Create New Source** 来生成新的UCF 文件。）：

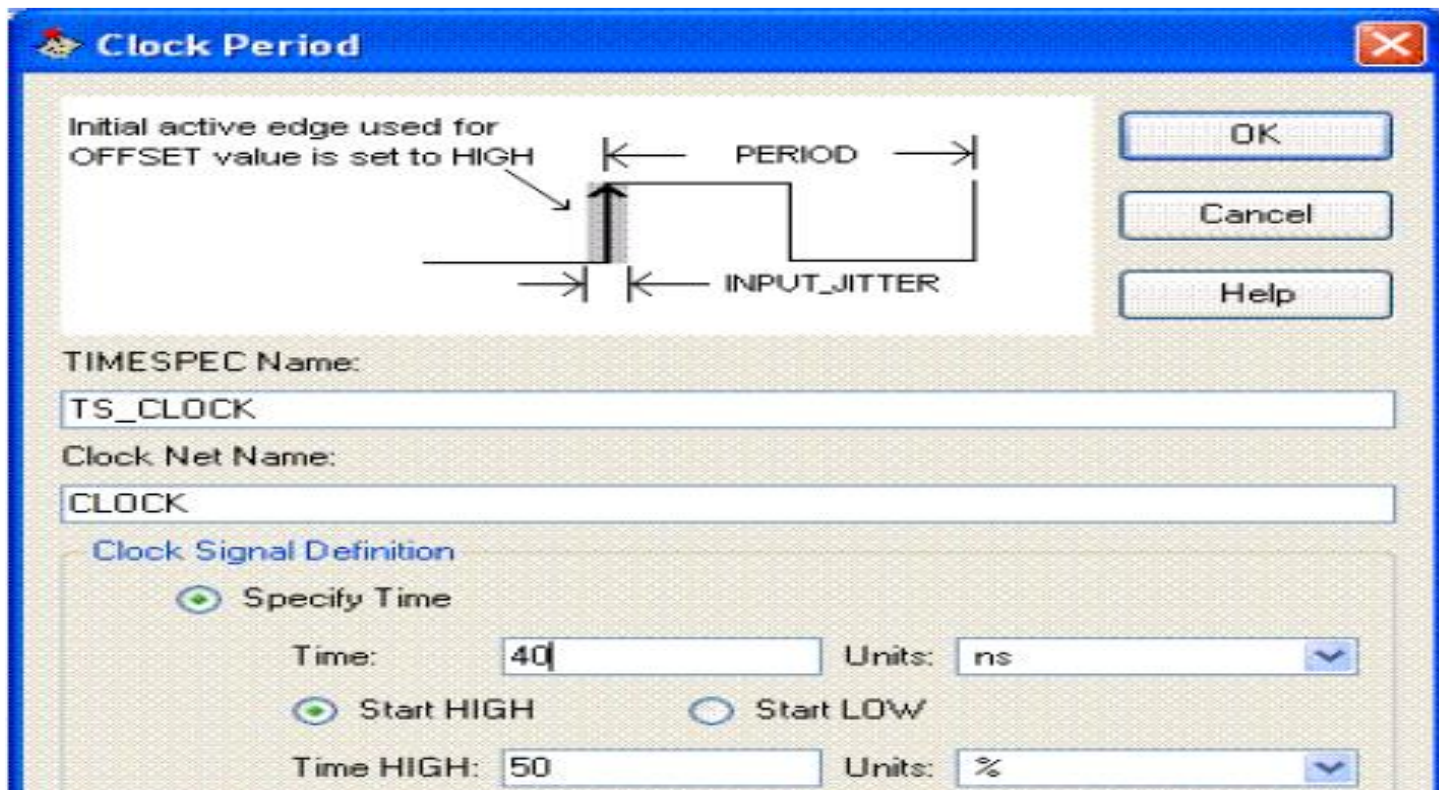




- 点击是(Yes) lamp_Controller1.ucf文件将加入到你的工程中，同时Xilinx约束编辑器将自动打开

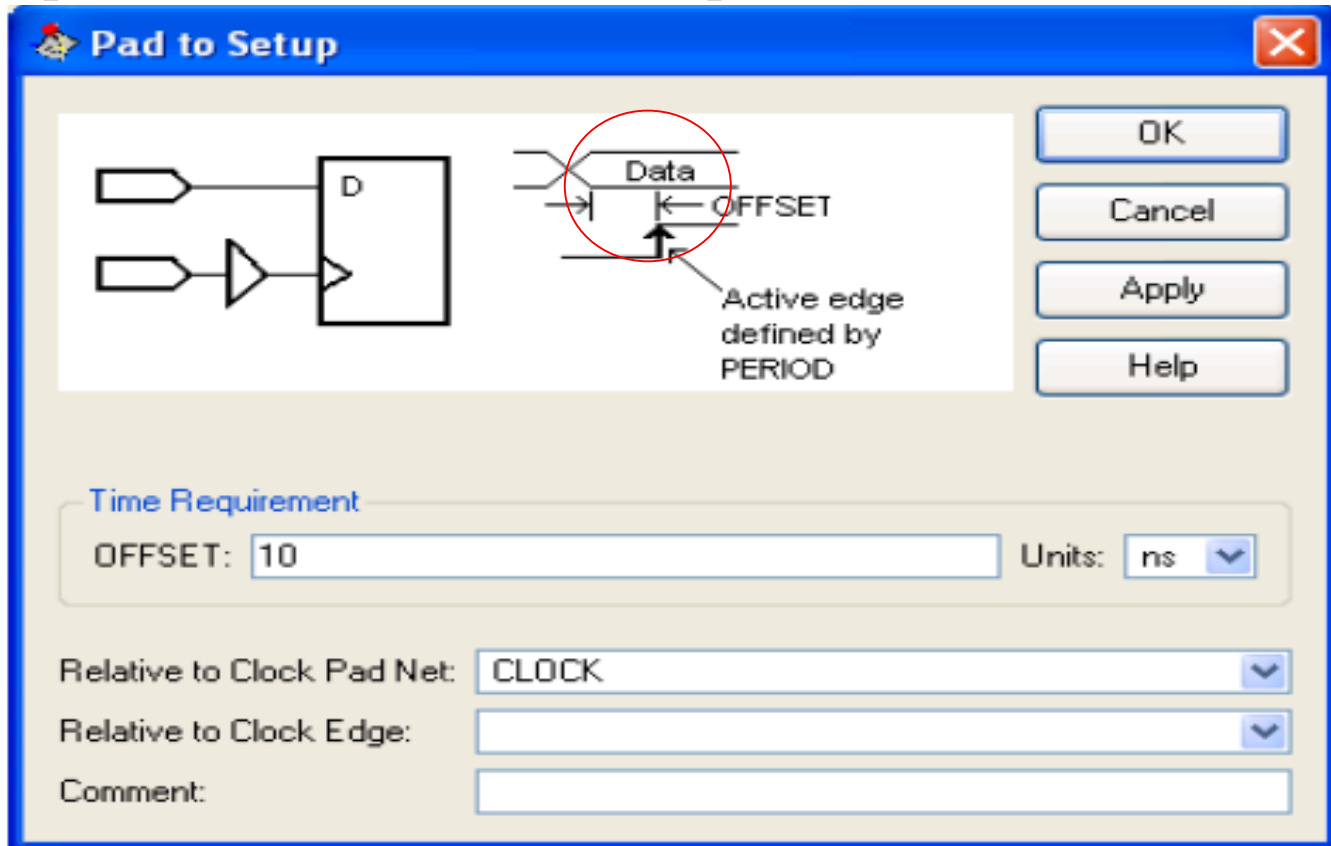


2. 在时钟节点名域(Clock Net Name field)中选中作为时钟的 **clk**, 然后选择工具栏上的 **Period** 按钮或双击空的周期域(empty Period field)系统显示时钟周期对话框



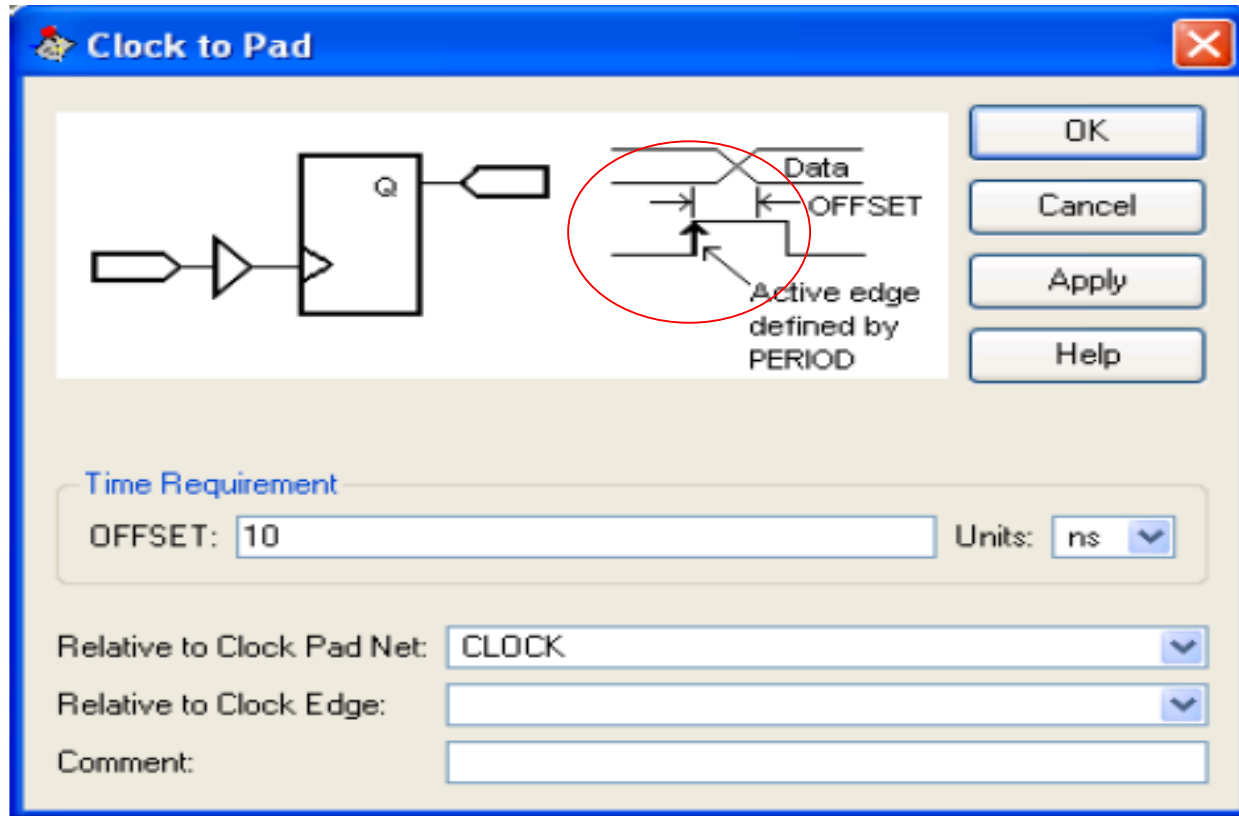
在时间域里键入40ns，并点击 OK

3. 选择工具栏上的 Pad to Setup  按钮，或者双击空的 Pad to Setup域，弹出显示 Pad to Setup对话框：



OFFSET设置：10，然后点击OK

选择工具栏上的 Clock to Pad 按钮，或者双击空的 Clock to Pad 域，弹出显示 Clock to Pad对话框：



OFFSET设置：10，然后点击OK

保存时间约束。关闭约束编辑器。



□ 【步骤7】 ~ 【步骤11】 与前面图形输入时相同。

■ lamp_Controller.ucf 引脚约束内容

NET "clk" LOC = "T9" ; (T代表板子上的系统时钟信号) 。

NET "F" LOC = "P11" ; (P11代表板子上的LD7指示灯) 。

NET "S1" LOC = "m13" ; (m13代表板子上的push0按钮) 。

NET "S2" LOC = "m14" ; (m14代表板子上的push1按钮) 。

NET "S3" LOC = "l13" ; (L13代表板子上的push2按钮) 。



● END