


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46



计算机系统结构实验指导书-LAB2

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	1 of 18
	CETC	修改日期	公开	
		03/21/2010		

1. OVERVIEW

1.1 实验名称

FPGA基础实验: 4-bitAdder

1.2 实验目的


- 1) 掌握Xilinx逻辑设计工具 Vivado的基本操作
- 2) 掌握 VerilogHDL进行简单的逻辑设计
- 3) 使用功能仿真;
- 4) 约束文件的使用和直接写法
- 5) 添加时序约束
- 6) 生成 Bitstream 文件
- 7) 上板验证 (暂定不做)

1.3 实验预计时间

约90分钟

1.4 实验报告与验收办法

- 1) 实验报告和工程文件在第十二周星期五晚上 23 点前提交
- 2) 无法验收, 但报告中需保留有效结果的仿真截图和有价值的心得段落

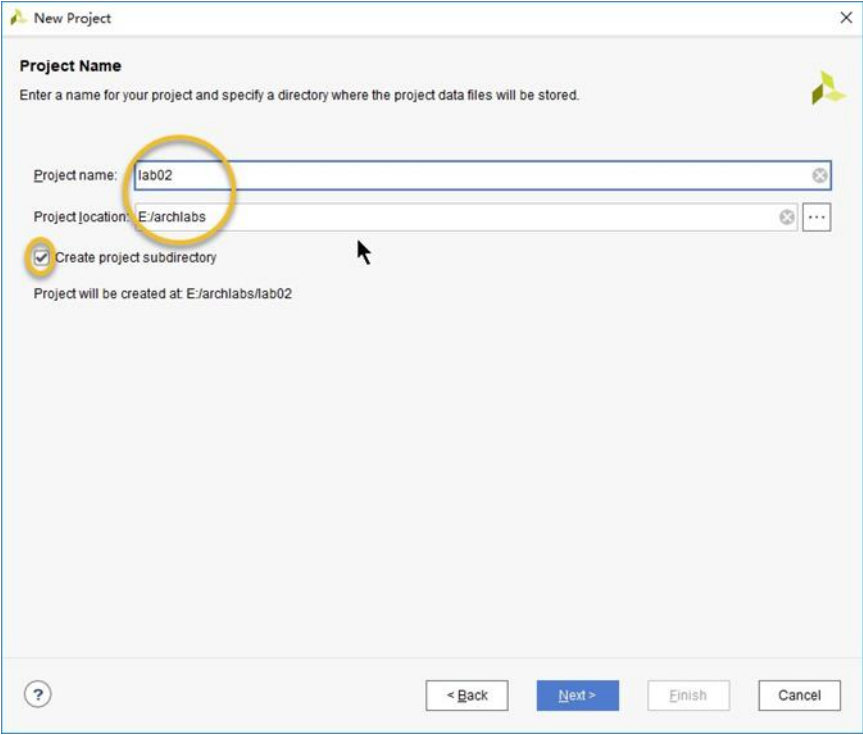
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2 CETC	CSE-COA-LAB-002 03/21/2010	0.5	2 of 18 公开


2. EXPERIMENTAL STEPS

2.1 新建工程

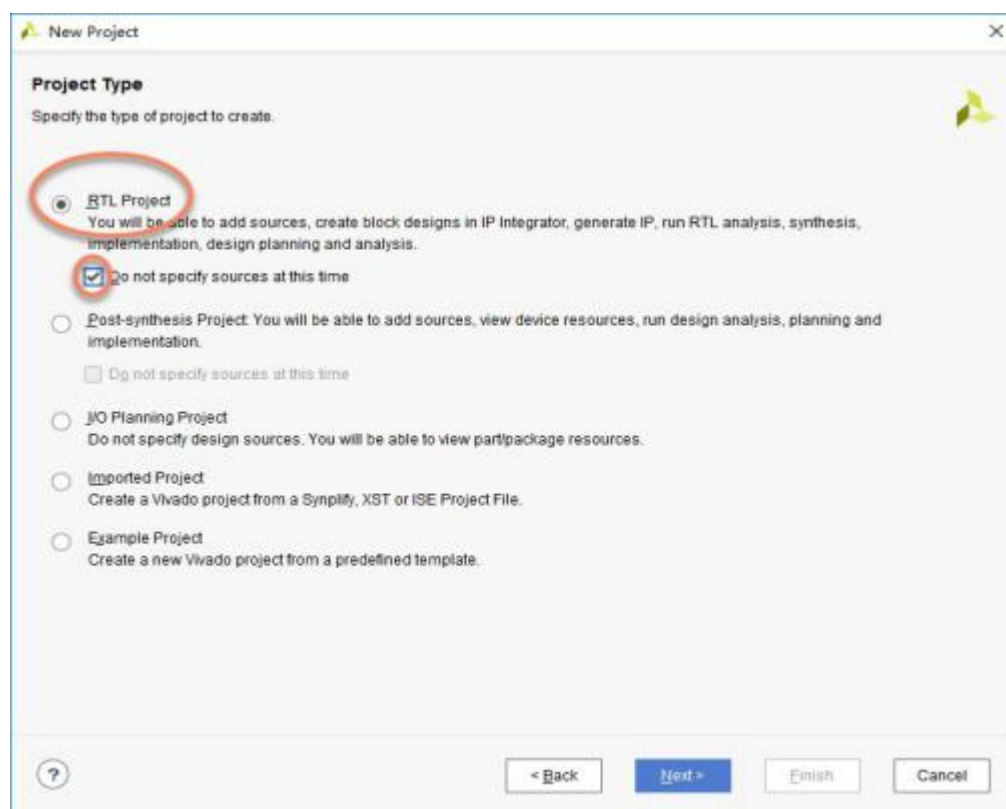
- 1) 启动桌面 Vivado 2018.3 开发工具
- 2) 点击 Create Project
- 3) 弹出 New Project, 建立一个新工程, 点击 Next
- 4) 输入工程名称 lab02, 确认勾选中 Create project subdirectory 后点击 Next

PS: 工程名称和存储路径中不能出现中文和空格, 建议工程名称以字母、数字、下划线来组成

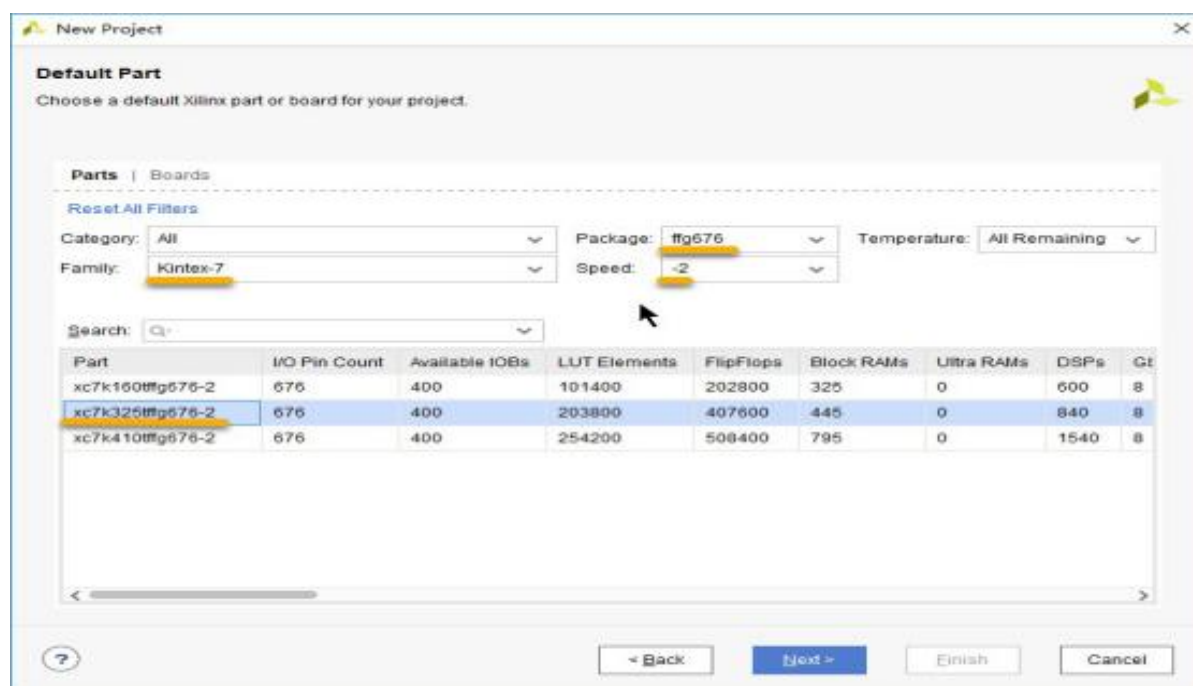



 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	3 of 18
	CETC	修改日期	公开	
		03/21/2010		

5) 选择RTL Project工程类型, 勾选Do not specify sources at this time, 点击Next

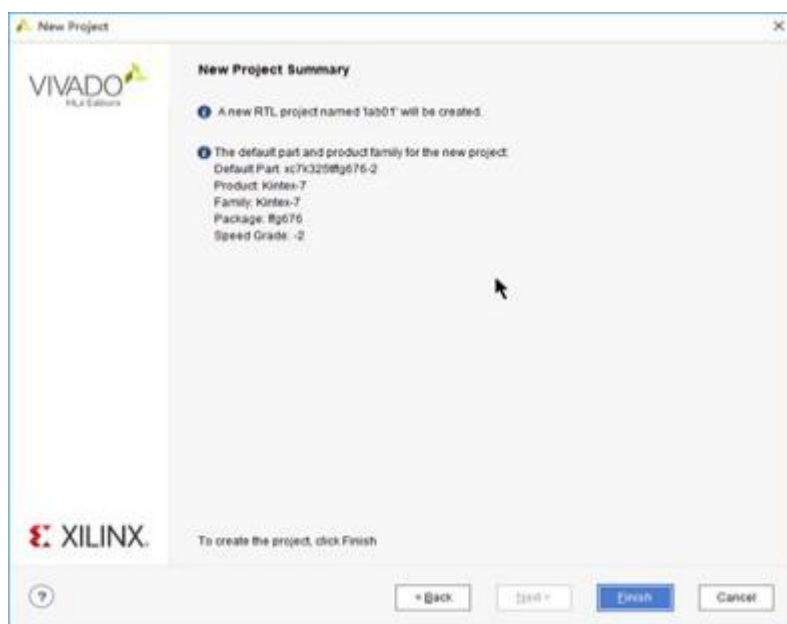


6) 选择 SWORD4.0 的 FPGA 参数: Family 选 Kintex-7, Package 选 ff676, Speed grade 选-2, 在找到的具体型号中选 xc7k325ff676-2, 点击 Next



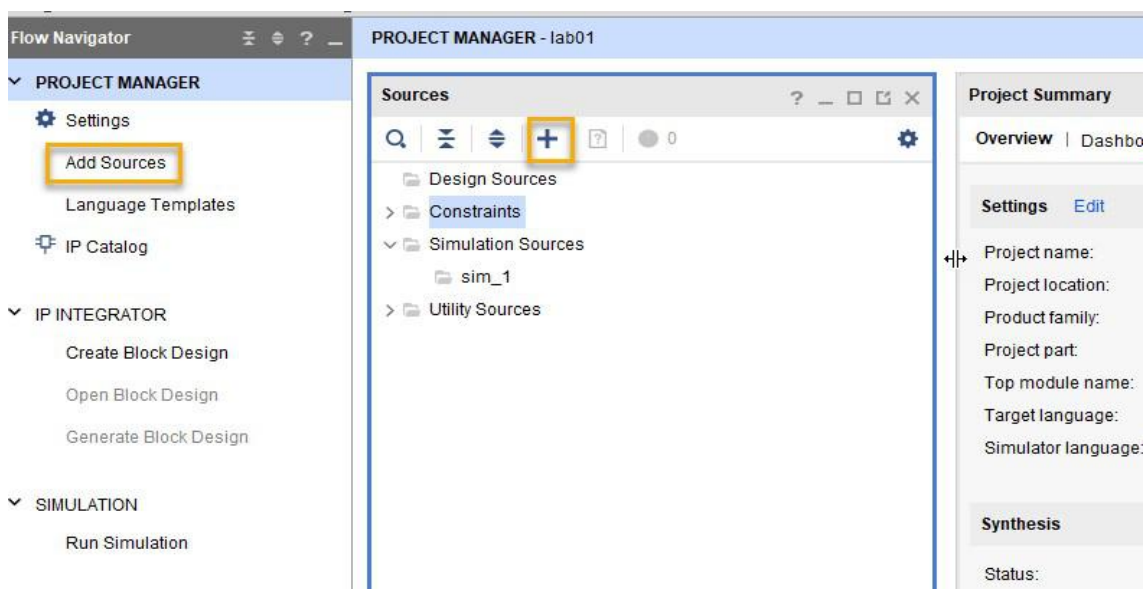
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	4 of 18
	CETC	修改日期 03/21/2010	公开	


7) 点击Finish结束工程的创建



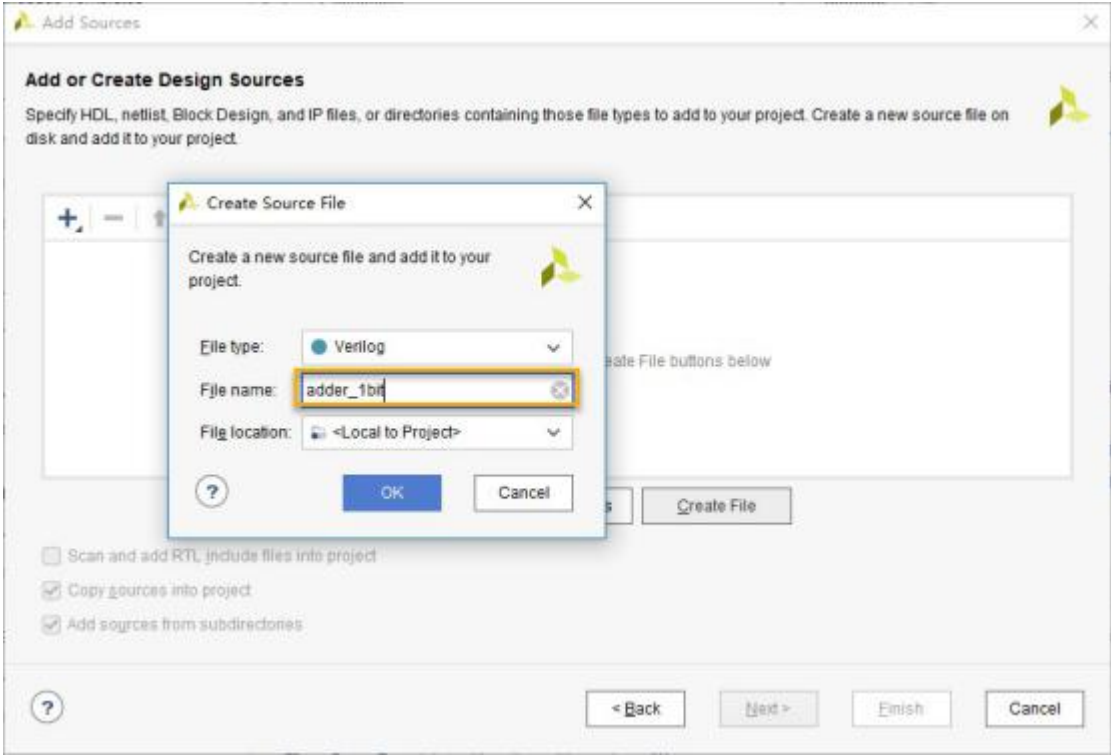
2.2 添加文件

- 1) 如下图，点击左侧区Flow Navigator下的Project Manager->Add Sources或中间区Sources的“+”号，打开Add Sources对话框




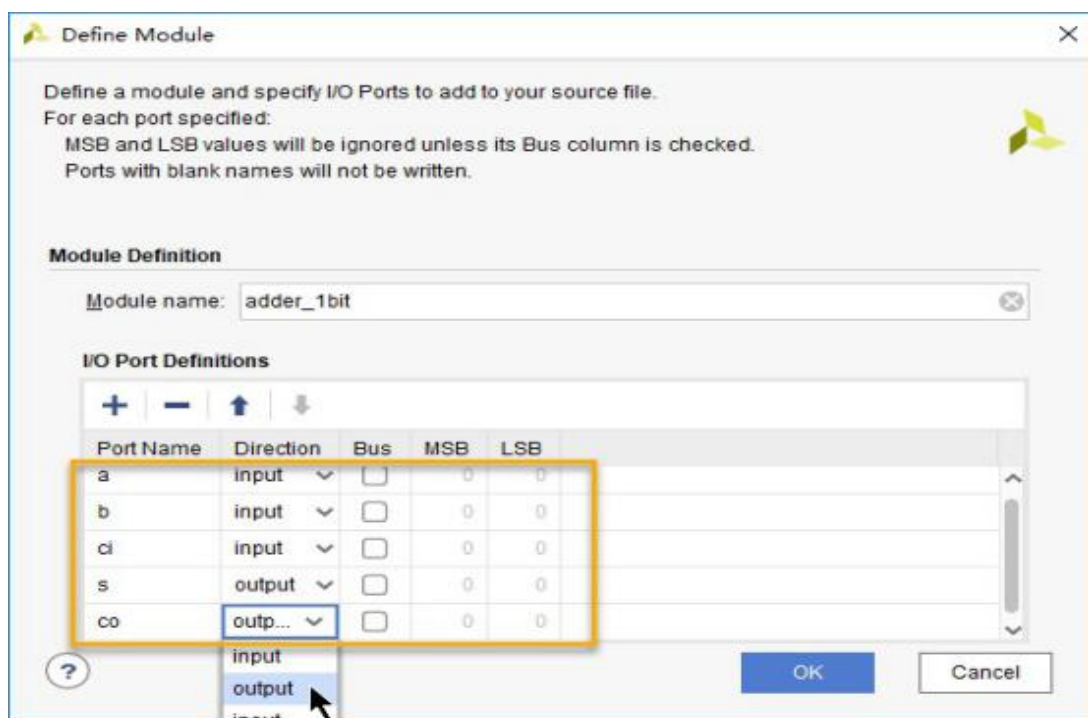
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	5 of 18
	CETC	修改日期 03/21/2010	公开	

- 2) 选择第二项 Add or Create Design Sources, 用来添加或新建 Verilog HDL 源文件, 点击 Next
- 3) 若已有源文件或内核文件, 可选 Add Files 项以添加文件。这里是要新建 1 位全加器模块文件, 选择 Create File
- 4) 在弹框 Create Source File 中输入 adder_1bit 文件名, 点击 OK



- 5) 点击 Finish
- 6) 在弹出的 Define Module 中输入设计模块所需的端口, 并设置端口方向; 完成后点击 OK

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	6 of 18
	CETC	修改日期 03/21/2010	公开	




7) 在源代码编辑区双击 adder_1bit，添加如下方框中的代码：

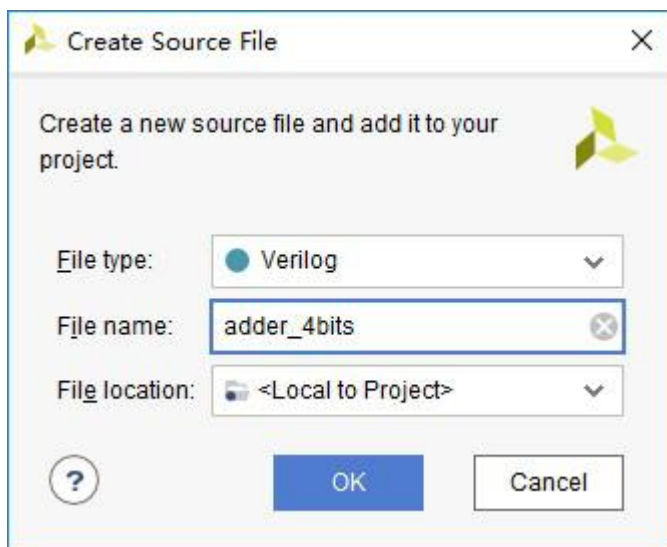
```

2  module adder_1bit(
3      input a,
4      input b,
5      input ci,
6      output s,
7      output co
8  );
9      wire s1, c1, c2, c3;
10     and (c1, a, b),
11         (c2, b, ci),
12         (c3, a, ci);
13
14     xor (s1, a, b),
15         (s, s1, ci);
16
17     or (co, c1, c2, c3);
18
19 endmodule
20

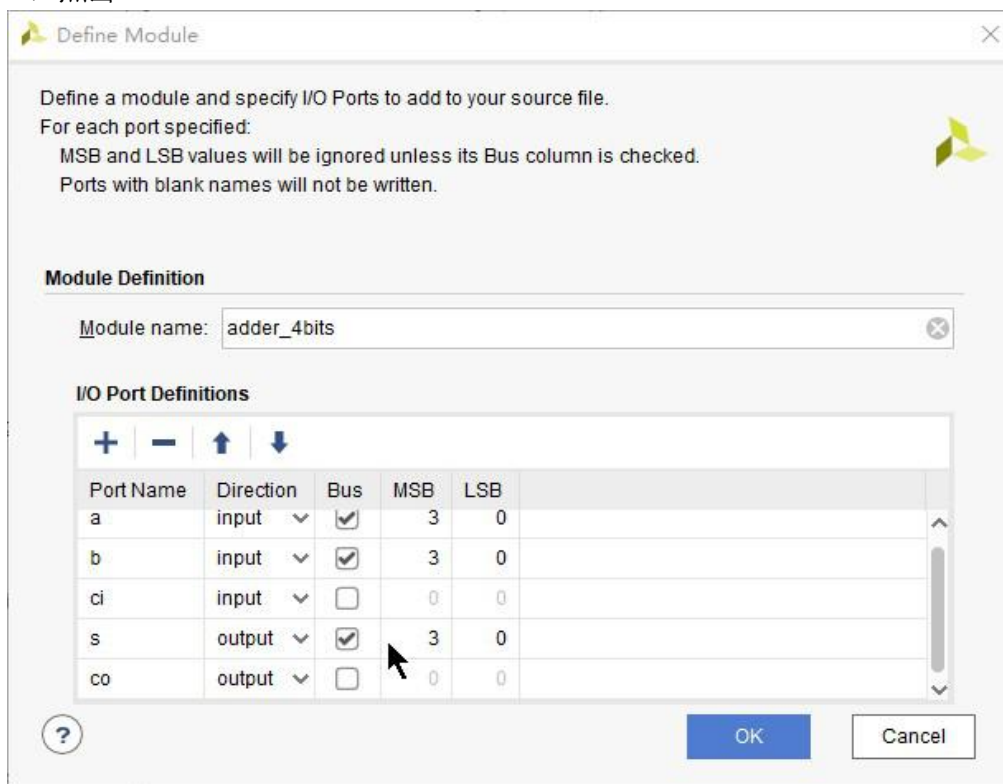
```

8) 接下来要实现4位加法器。继续创建源文件，命名为adder_4bits，点击 OK


	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	7 of 18
	CETC	修改日期 03/21/2010	公开	



- 9) 点击 Finish，输入模块所需的端口，并设置端口方向，若端口属总线型，勾选 Bus，并由 MSB 和 LSB 确定宽度（当然这一步也可忽略模块的端口定义和宽度确定，直接点击 OK 之后点击 Yes，再点击 Finish）点击 OK



- 10) 双击 adder_4bits，添加如下代码

	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	8 of 18
	CETC	修改日期 03/21/2010	公开	

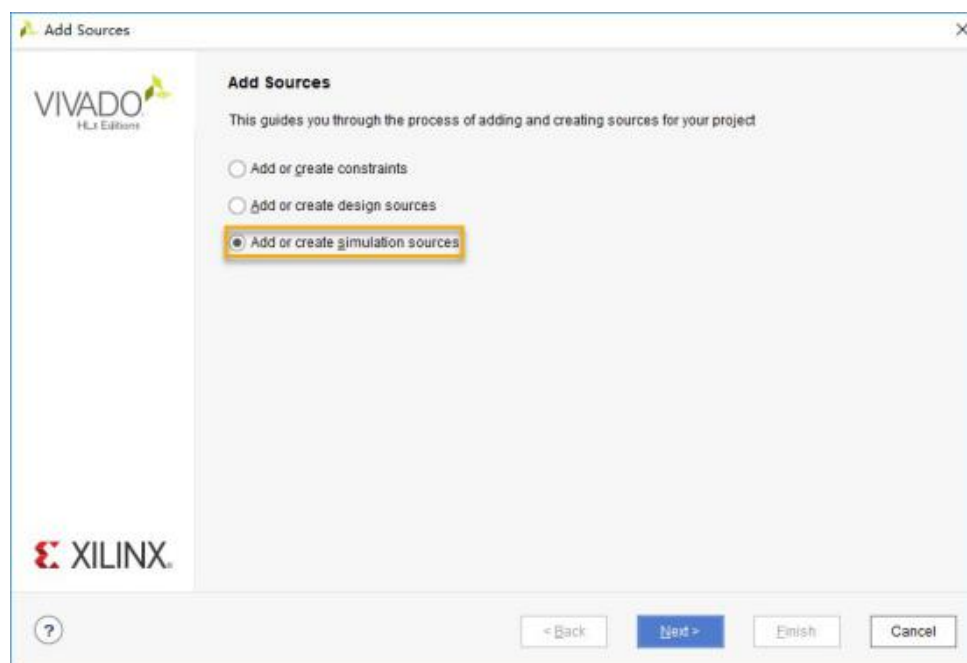

```

1  1
2  2 module adder_4bits(
3      input [3:0] a,
4      input [3:0] b,
5      input ci,
6      output [3:0] s,
7      output co
8  );
9
10     wire [2:0] ct;
11
12     adder_1bit a1(.a(a[0]), .b(b[0]), .ci(ci), .s(s[0]), .co(ct[0])),
13                 a2(.a(a[1]), .b(b[1]), .ci(ct[0]), .s(s[1]), .co(ct[1])),
14                 a3(.a(a[2]), .b(b[2]), .ci(ct[1]), .s(s[2]), .co(ct[2])),
15                 a4(.a(a[3]), .b(b[3]), .ci(ct[2]), .s(s[3]), .co(co));
16
17 endmodule
18


```

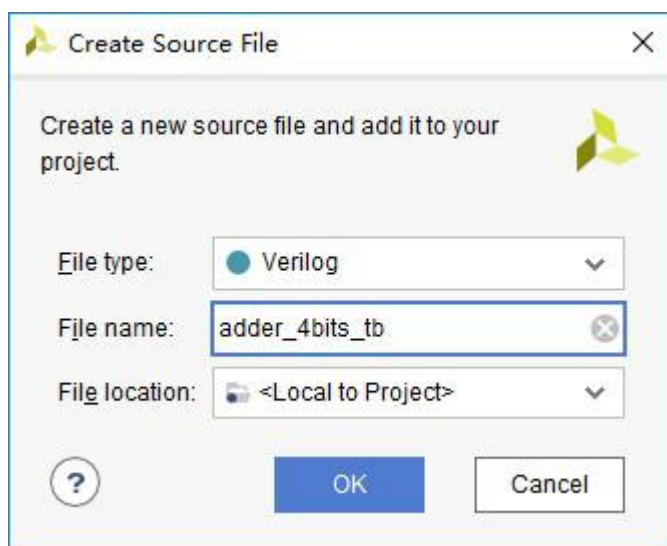
2.3 功能仿真

- 1) 创建激励测试文件。在中间区 Sources 具栏点击 “+” 号或于左侧区 PROJEUMANAGER 下选择 Add Source
- 2) 在 Add Sources 选择 Add or Create Simulation Source, 点击 Next




- 3) 选择 Create File 创建一个仿真激励文件
- 4) 激励文件测试名可输入 adder_4bits_tb, 点击 OK

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	9 of 18
	CETC	修改日期 03/21/2010	公开	



- 5) 点击 Finish，创建激励测试文件不需要对外端口，点击 OK
- 6) 在弹出的对话框点击 YES
- 7) 在 Source 区 Simulation Sources 下，打开测试文件 adder_4bit_tb，在其中对要仿真的模块进行实例化和激励代码的编写（并点击保存），如下图所示::

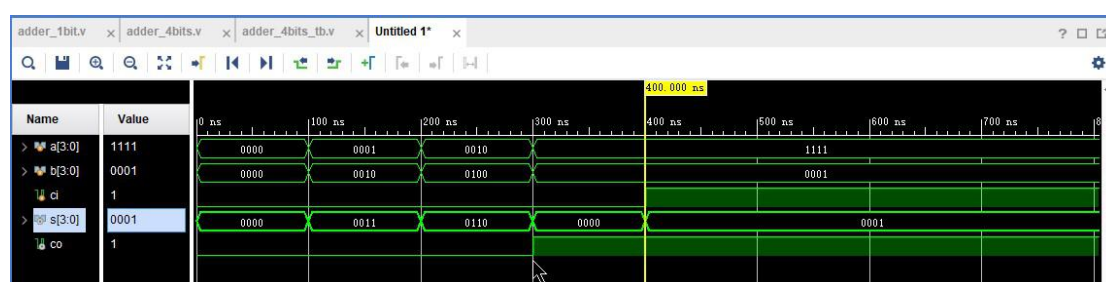
	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	10 of 18
	CETC	修改日期	公开	
		03/21/2010		

```

1  module adder_4bits_tb( );
2
3
4      reg [3:0] a;
5      reg [3:0] b;
6      reg ci;
7
8      wire [3:0] s;
9      wire co;
10
11     adder_4bits u0 (
12         .a(a),
13         .b(b),
14         .ci(ci),
15         .s(s),
16         .co(co)
17     );
18
19     initial begin
20         a = 0;
21         b = 0;
22         ci = 0;
23
24         #100;
25         a = 4'b0001;
26         b = 4'b0010;
27         #100;
28         a = 4'b0010;
29         b = 4'b0100;
30
31         #100;
32         a = 4'b1111;
33         b = 4'b0001;
34         #100;
35         ci = 1'b1;
36
37     end
38
39 endmodule


```

- 8) 点击左侧区的Run Simulation并选择Run Behavioral Simulation。下图为仿真运行后得到正确的仿真波形:



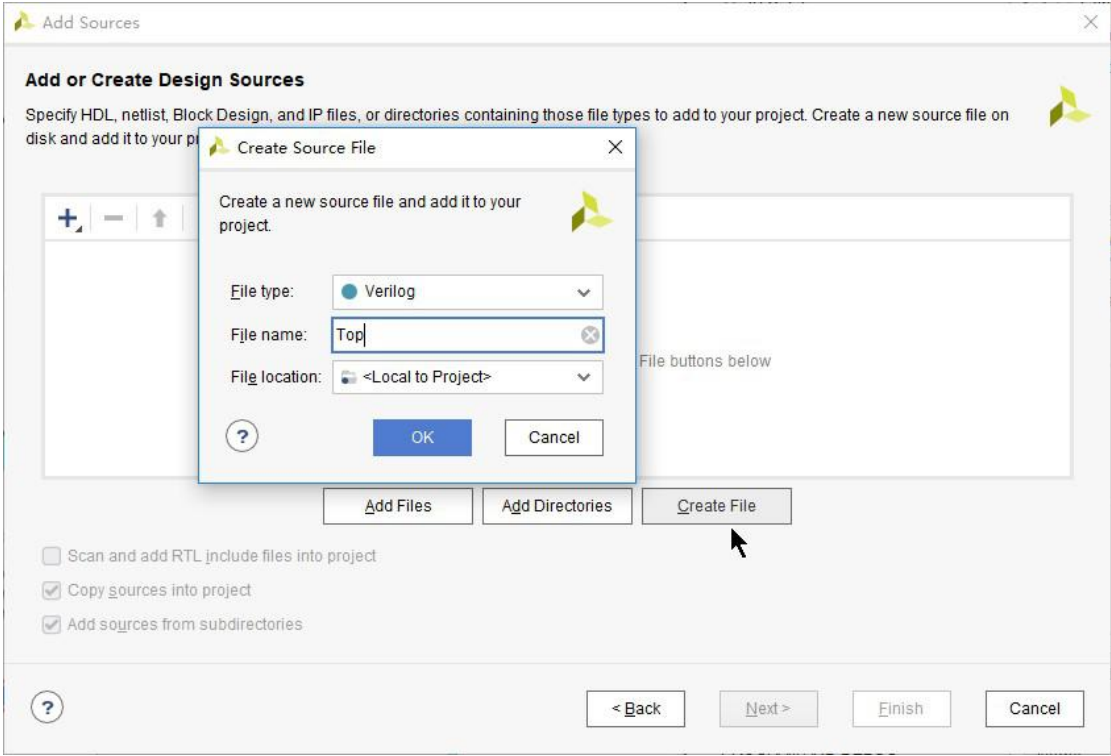
观察波形图, 可知仿真结果与逻辑功能一致(可以思考下该图为什么是正确的), 电路能正常工作。如果初看不一致可以放大或缩小、可以滑动黄色的时间运行竖线来查看输入与输出间的变化是否吻合, 若有误可修改逻辑代码。

2.4 工程实现

	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	11 of 18
	CETC	修改日期 03/21/2010	公开	

在实验二中是要用实验板上的 8 个 Switch 对应二组 4 位二进制输入，用 4 个 LED 发光二极管对应输出并用 2 个七段数码管显示运行结果。故本实验需要用到 display.v 这个七段数码管 SEGMENT 和 LED 显示模块（实验室提供该核，以网表文件形式给出）

- 1) 将 4 位加法器的输出赋予 LED 和七段数码管显示，需要创建一个顶层源文件，可命名 Top，如下图：




进入 Define Module，这里我们略过端口定义，将在源程序中自行添加；双击 Top 模块，添加以下端口变量和代码语句：

```
module Top(
    input clk_p,
    input clk_n,
    input [3:0] a,
    input [3:0] b,
    input reset,

    output led_clk,
    output led_do,
    output led_en,

    output wire seg_clk,
    output wire seg_en,
    output wire seg_do
);
```


 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	12 of 18
	CETC	修改日期 03/21/2010	公开	

```

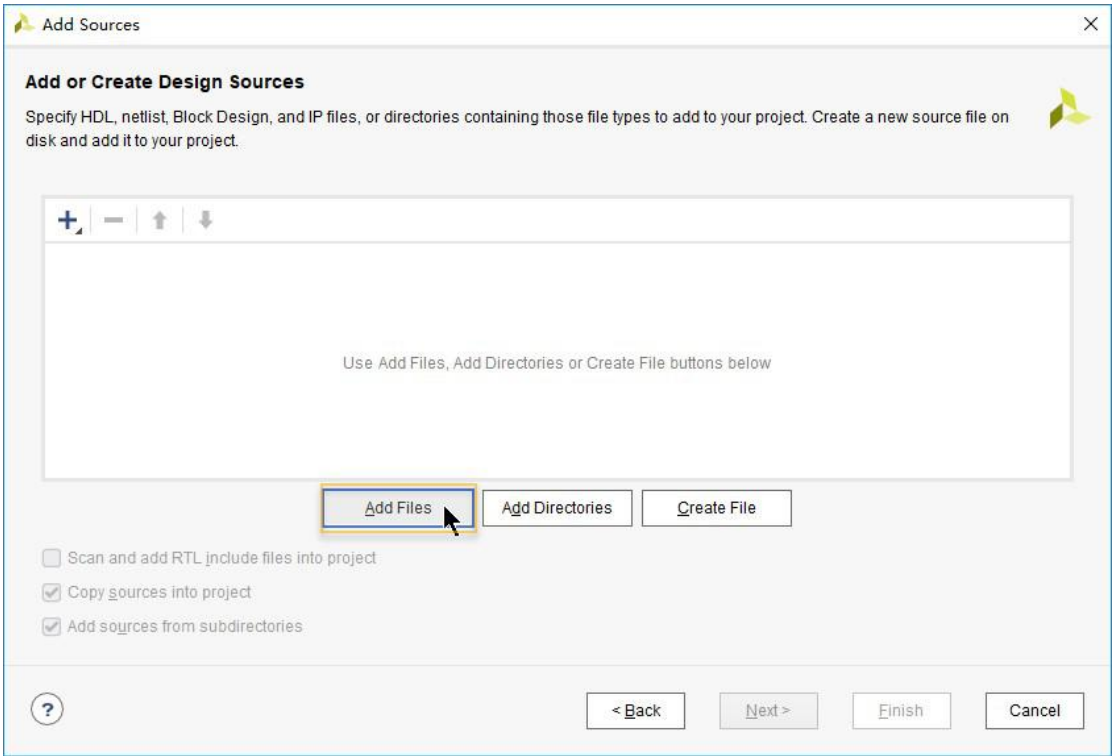
1      wire CLK_i;
2      wire Clk_25M;
3
4      IBUFGDS IBUFGDS_inst (
5          .O(CLK_i),
6          .I(clk_p),
7          .IB(clk_n)
8      );
9
10     wire [3:0] s;
11     wire co;
12     wire [4:0] sum;
13     assign sum = {co, s};
14
15     adder_4bits U1 (
16         .a(a),
17         .b(b),
18         .ci(1'b0),
19         .s(s),
20         .co(co)
21     );
22
23     reg [1:0] clkdiv;
24     always@(posedge CLK_i)
25         clkdiv<=clkdiv+1;
26     assign Clk_25M=clkdiv[1];
27
28     display DISPLAY
29     ( .clk(Clk_25M),
30       .rst(1'b0),
31       .en(8'b00000011),
32       .data({27'b0, sum}), .dot(8'b00000000),
33       .led(~{11'b0, sum}),
34       .led_clk(led_clk),
35       .led_en(led_en),
36       .led_do(led_do),
37       .seg_clk(seg_clk),
38       .seg_en(seg_en),
39       .seg_do(seg_do)
40     );

```

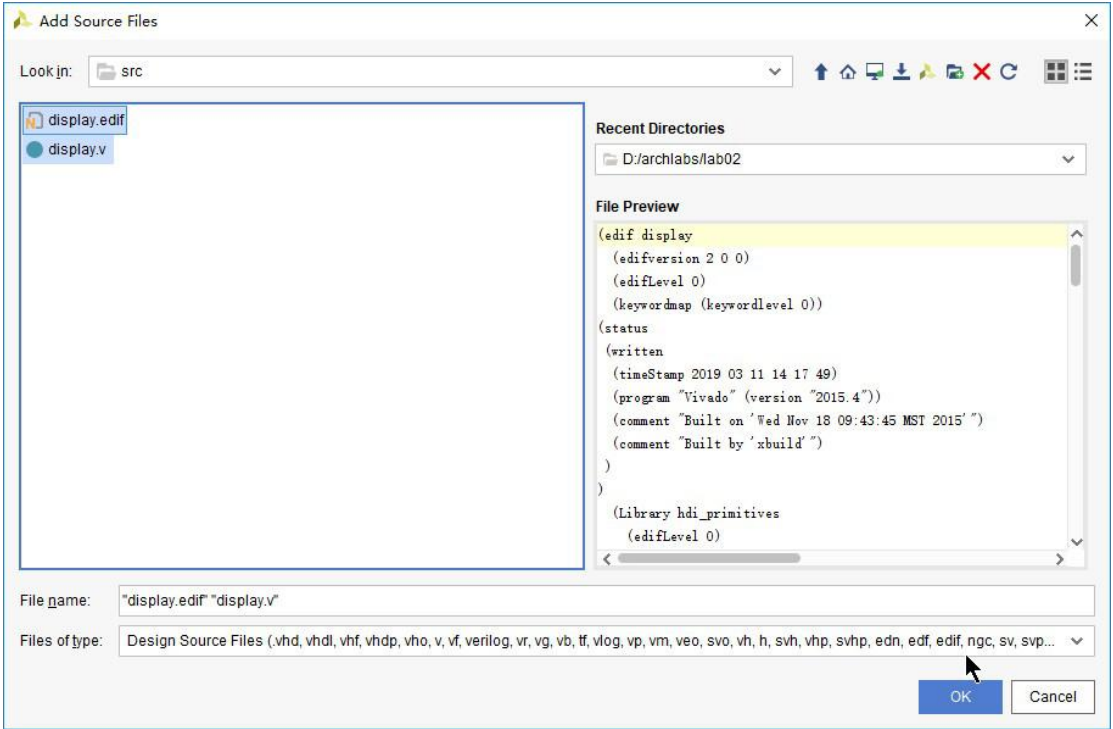
Endmodule

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	13 of 18
	CETC	修改日期	公开	
		03/21/2010		

2) Top.v 中用到了一个 display IP 核（已转换为网表），接下来需要添加该核。如下图，在 Add Sources 中，点击 Add Sources:



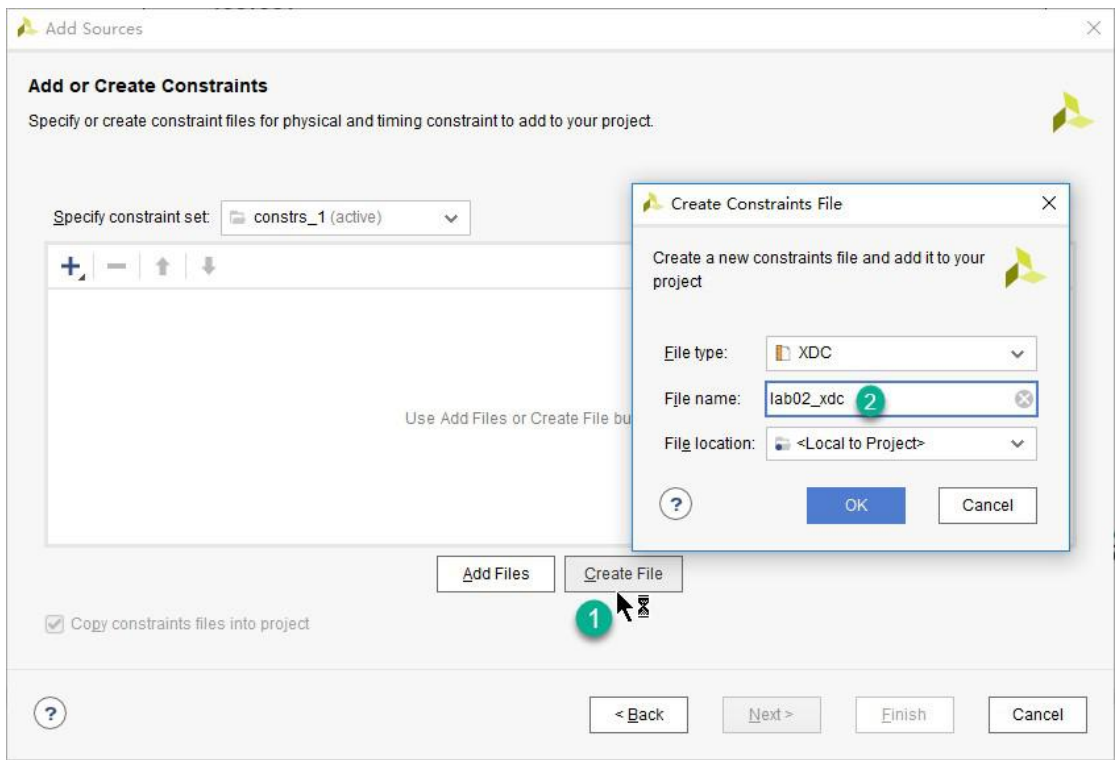
在 Add Source Files 中同时选择一对与该核同前缀名的端口和网表文件，如下图:



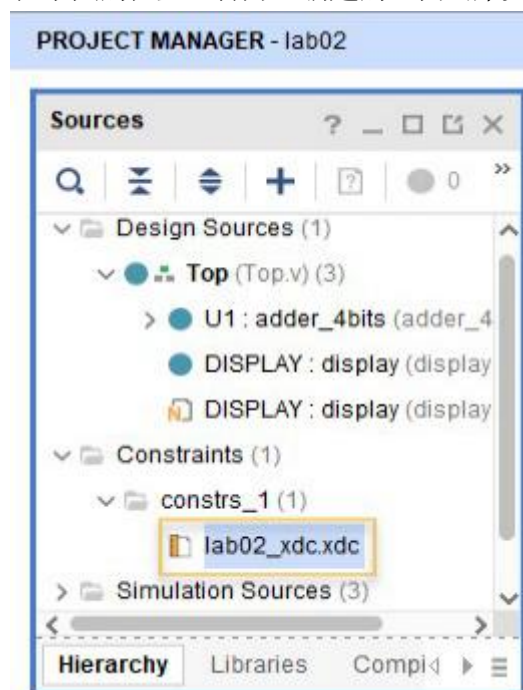
点击 Finish，则在 Sources 区的顶层源文件下便关联了这一对源文件。

上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	14 of 18
	CETC	修改日期	公开	
		03/21/2010		

3) 添加约束文件。打开 Add Sources 对话框, 如图选择第一项



在下图的位置，打开这新建的空白约束文件：




本实验需添加如下约束代码：

```
set_property PACKAGE_PIN AC18 [get_ports clk_p]
set_property IOSTANDARD LVDS [get_ports clk_p]
set_property PACKAGE_PIN AA12 [get_ports {a[3]}]
set_property PACKAGE_PIN AA13 [get_ports {a[2]}]
set_property PACKAGE_PIN AB10 [get_ports {a[1]}]
set_property PACKAGE_PIN AA10 [get_ports {a[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {a[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {a[1]}]
set_property IOSTANDARD LVCMOS15 [get_ports {a[2]}]
set_property IOSTANDARD LVCMOS15 [get_ports {a[3]}]

set_property PACKAGE_PIN AD10 [get_ports {b[3]}]
set_property PACKAGE_PIN AD11 [get_ports {b[2]}]
set_property PACKAGE_PIN Y12 [get_ports {b[1]}]
set_property PACKAGE_PIN Y13 [get_ports {b[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {b[0]}]
set_property IOSTANDARD LVCMOS15 [get_ports {b[1]}]
set_property IOSTANDARD LVCMOS15 [get_ports {b[2]}]
set_property IOSTANDARD LVCMOS15 [get_ports {b[3]}]

set_property PACKAGE_PIN N26 [get_ports led_clk]
set_property PACKAGE_PIN M26 [get_ports led_do]
set_property PACKAGE_PIN P18 [get_ports led_en]
set_property IOSTANDARD LVCMOS33 [get_ports led_clk]
set_property IOSTANDARD LVCMOS33 [get_ports led_do]

set_property IOSTANDARD LVCMOS33 [get_ports led_en]
set_property PACKAGE_PIN M24 [get_ports seg_clk]
```

	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	16 of 18
	CETC	修改日期 03/21/2010	公开	

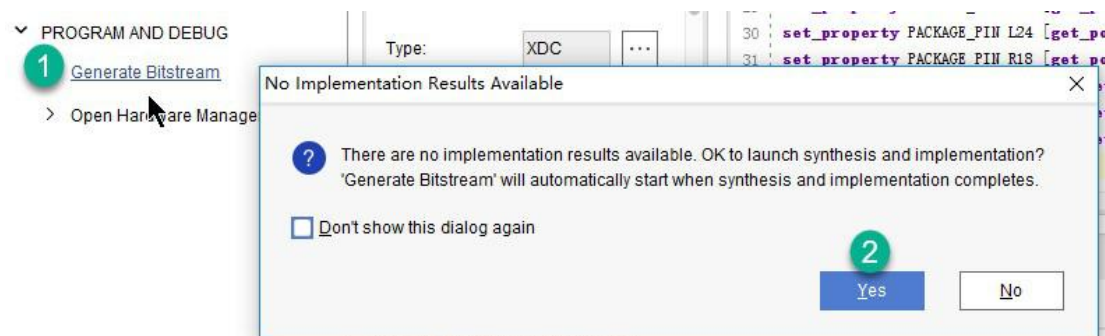
```

1 set_property PACKAGE_PIN L24 [get_ports seg_do]
2 set_property PACKAGE_PIN R18 [get_ports seg_en]
3 set_property IOSTANDARD LVCMOS33 [get_ports seg_clk]
4 set_property IOSTANDARD LVCMOS33 [get_ports seg_do]
5
6
7
8
9
10

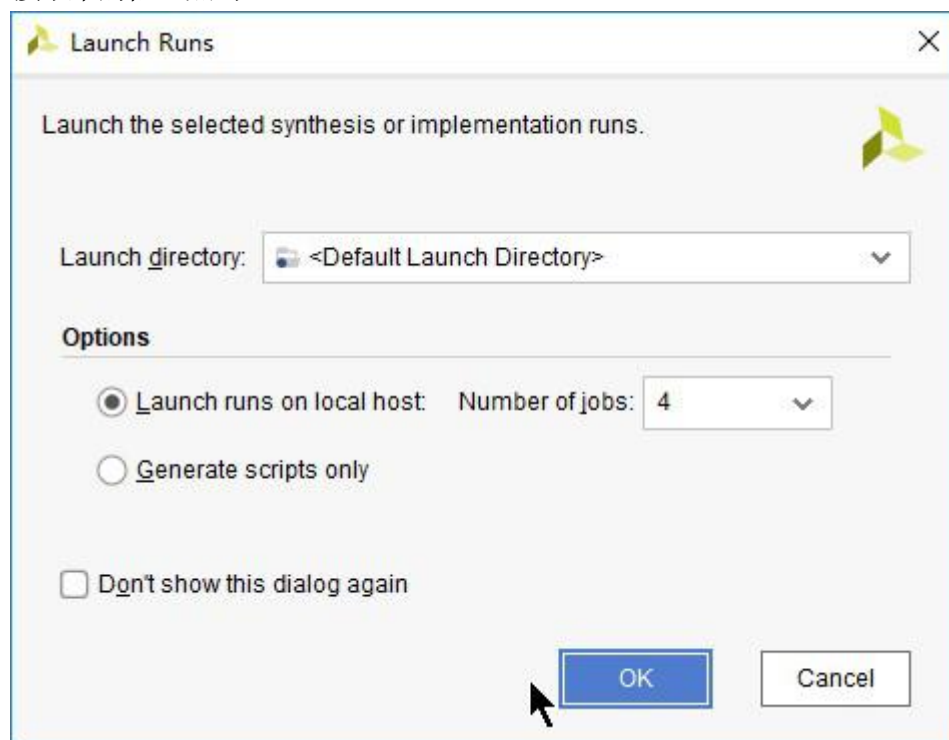
```

2.6 下载验证（暂无法做）


1. 在 Flow Navigator 区点击 Program and Debug 下 Generate Bitstream 选项，系统将自动完成综合、实现、并生成 FPGA 配置文件。出现以下框，点 Yes：

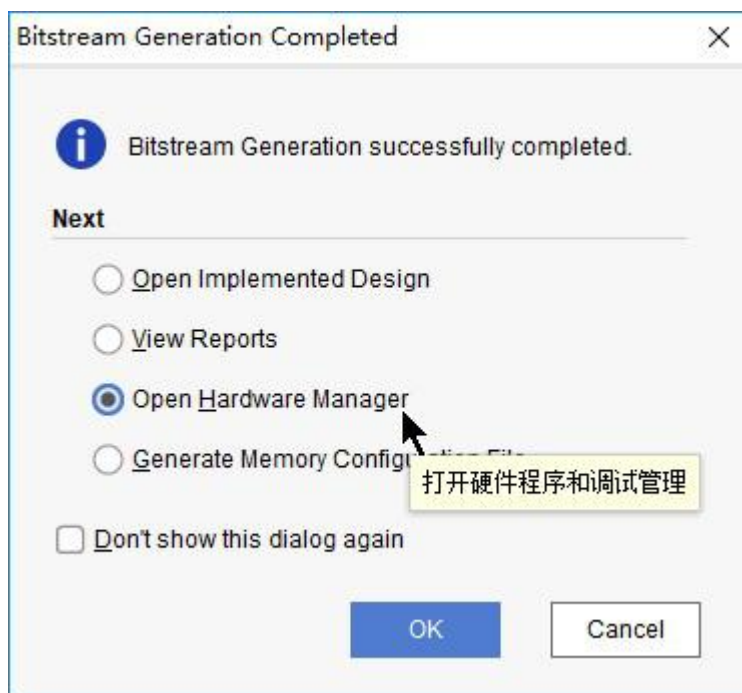


接下来的框，点击 OK：



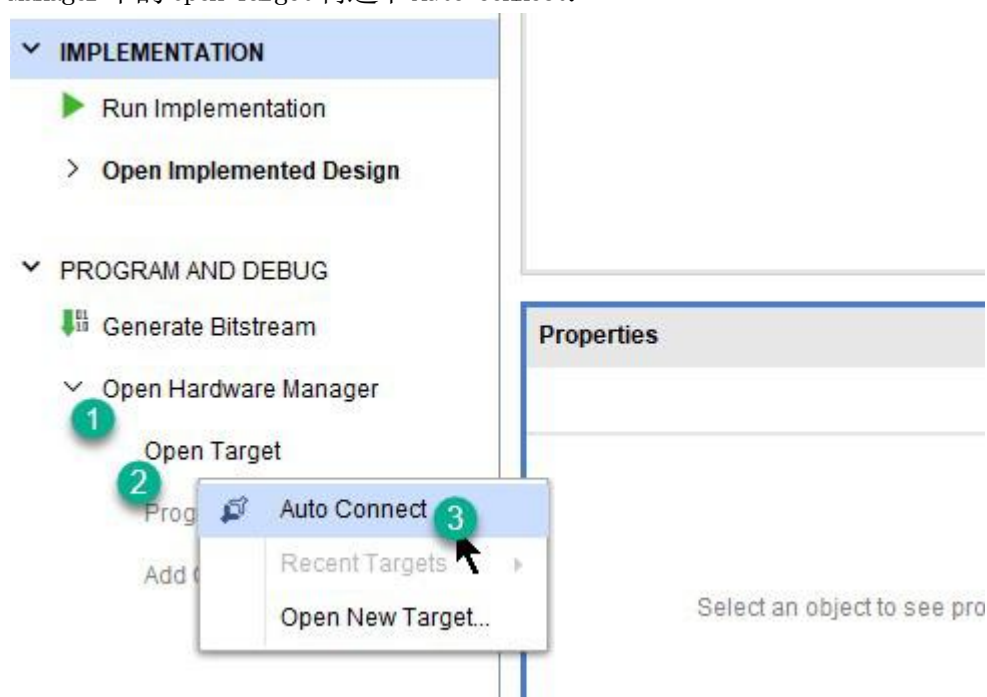
FPGA 配置文件生成后，便可将 bit 文件下载到实验板上以查看真实效果，此次选择第三项直接去“烧写”

	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	17 of 18
	CETC	修改日期 03/21/2010	公开	




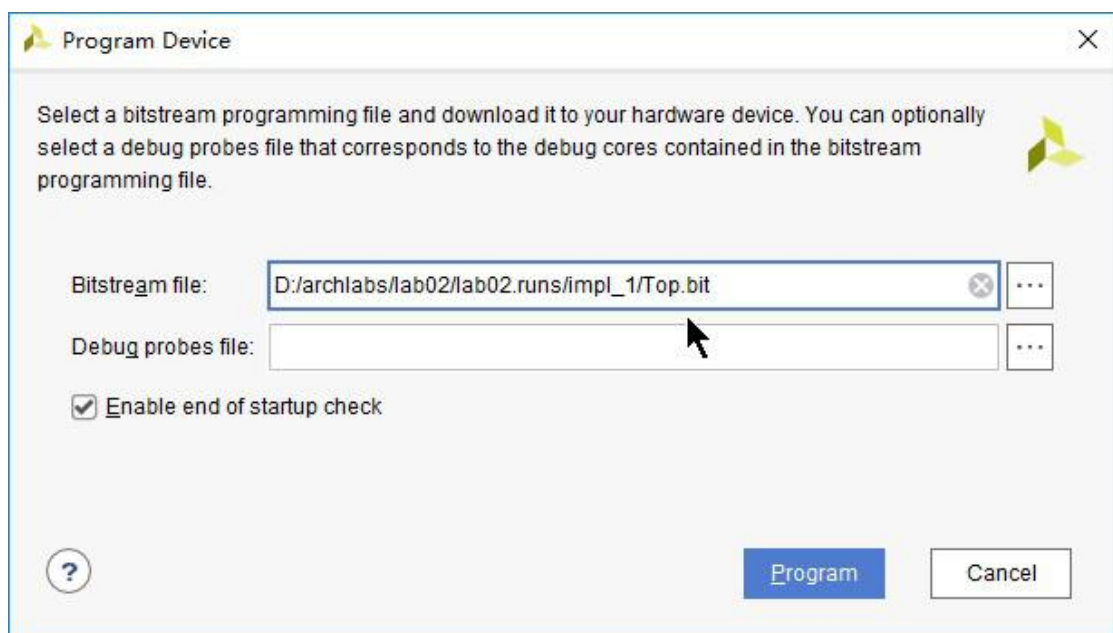
2. 连接好 SWORD 开发板的电源和 JTAG，然后打开电源开关

3. 将 bit 文件下载到开发板 FPGA 芯片上以查看真实效果。在 Flow Navigator 中点击 Open Hardware Manager 下的 Open Target 再选中 Auto Connect:




4. 点击 Open Hardware Manager 下的 Program Device 再选择 FPGA 芯片 xc7k325t_0，弹出的对话框中文件名处会自动加载本工程生成的 bit 文件，点击 Program 对 FPGA 芯片进行现场编程

	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	18 of 18
	CETC	修改日期 03/21/2010	公开	



5. 观察实验结果，显示是否预期。等待 bit 文件下载配置完成，尝试拨动开关，拨码开关的低 4 位和次高 4 位代表两个加数；LED 的低 4 位代表的是和，第 5 位代表的是进位；2 个七段数码管也代表和的结果。

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB2	CSE-COA-LAB-002	0.5	19 of 18
	CETC	修改日期 03/21/2010	公开	