

实验手册

陈田

实验一 Vivado 平台入门

1. 实验内容

完成 Vivado 的安装

2. 实验目标

- 1) 安装 Xilinx 的 Vivado 开发套件。
- 2) 熟悉 Vivado 仿真，原理图，下载方法。

3. 实验步骤

登录 <http://china.xilinx.com/products/design-tools/vivado.html>

选择合适的版本下载。（下载需要 Xilinx 账户，请自行申请，后续授权也要用到。）

安装过程不再赘述。

Vivado 平台在工程文件完成编码后一般沿着综合、仿真、链接端口、生成 Bit 文件和下载至开发板的过程完成工程的实践。如图 1 所示，展现了打开一个 Vivado 工程的示例。

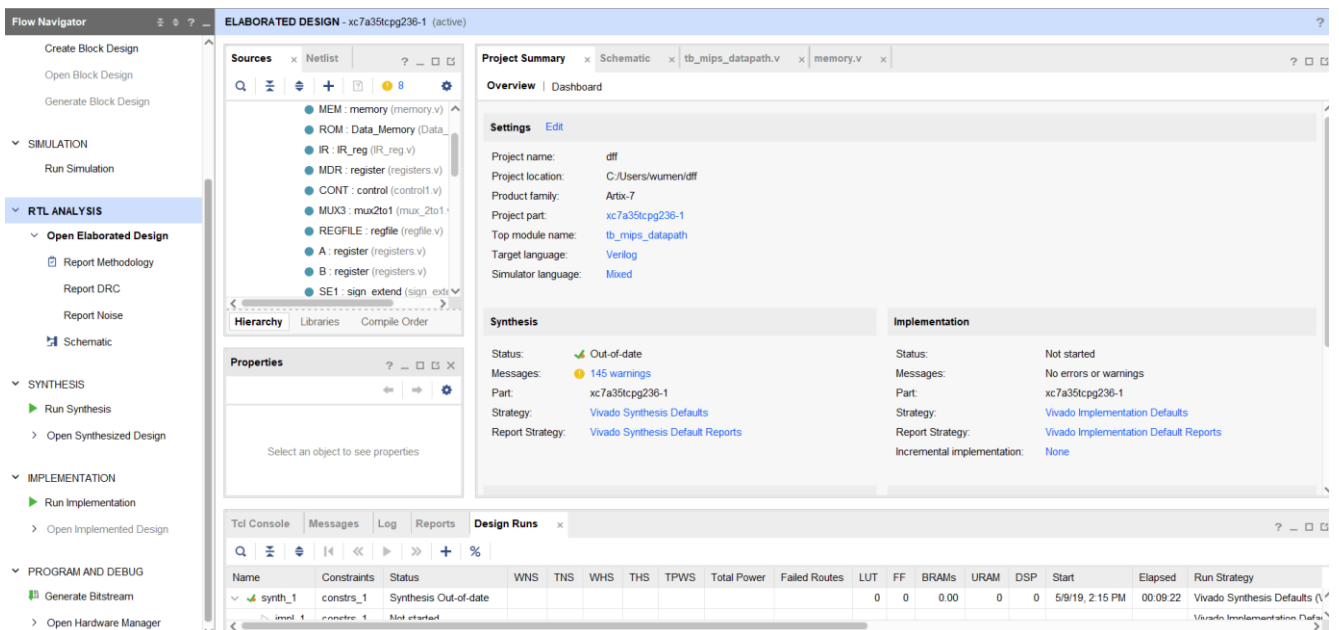


图 1 Vivado 主界面

Flow Navigator：设计流程导航；

Simulation 对顶层文件仿真，可以看到时序波形的变化；

RTL Analysis: 进行设计规则检测（DRCs），其下的 Schematic 可以生成器件连线的 RTL 原理图；

Synthesis 是对指定的顶层文件进行层次化综合；

Implementation 是对设计进行实现，链接相关端口；

Program and Debug: 对 bitstream 进行设置，Generate Bitstream 生成 bit 流文件，可以打开硬件管理器，将 bit 流文件烧写到指定设备。如果设计中加入了 Debug 核，这里会打开 Vivado 逻辑分析仪，对系统进行 Debug。

例如：点击红框位置进行 Run Implementation 操作。

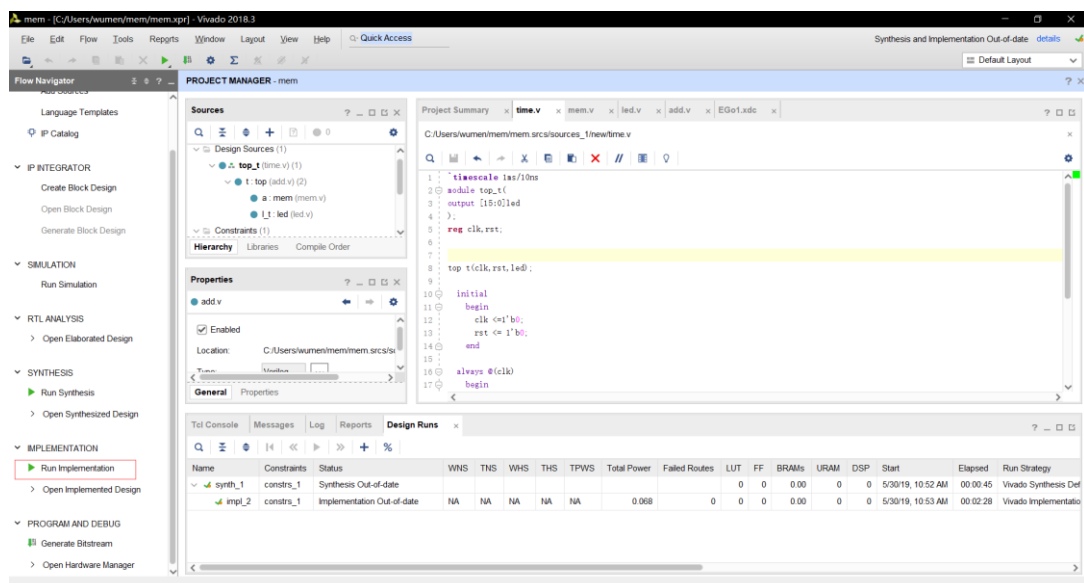


图 2 Run Implementation 操作

在这两部分设计验证都没有问题的情况下，可以生成 bit 文件，该文件就是用于写入开发板的部分。

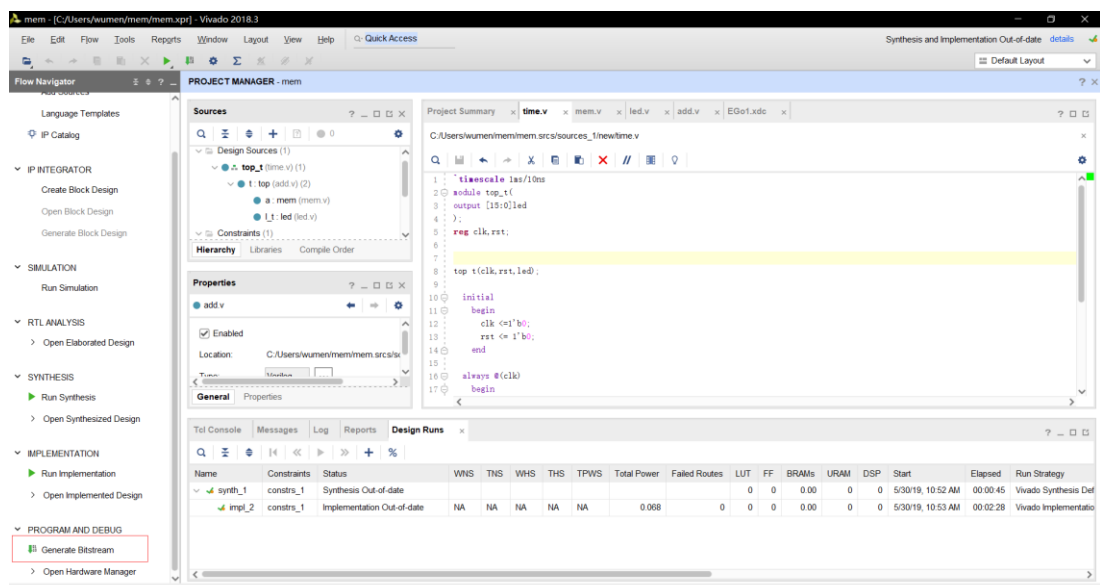


图 1.3 Generate Bitstream

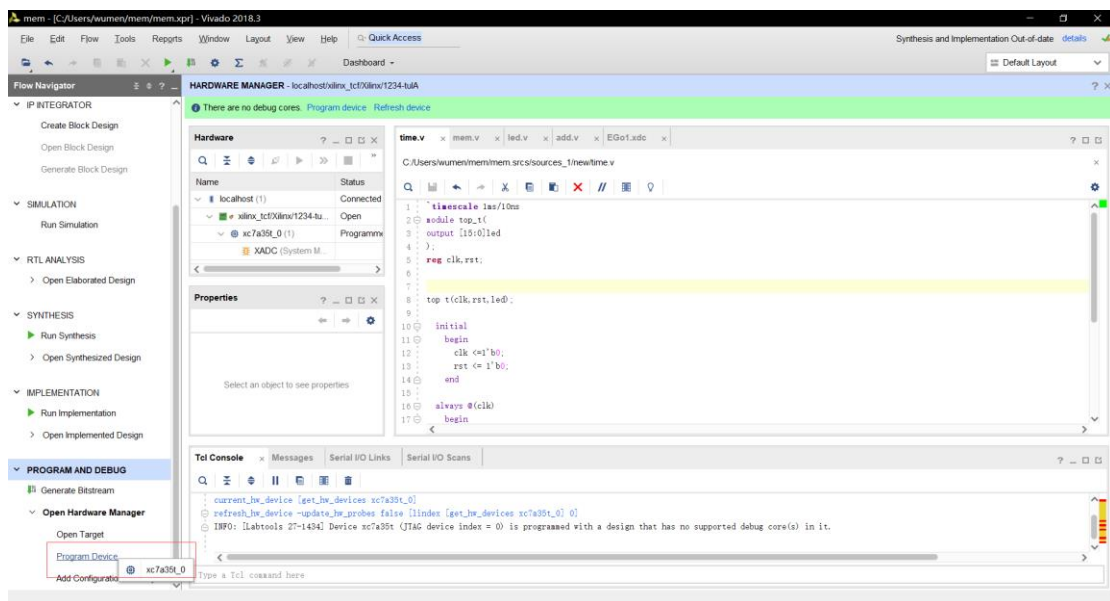


图 1.4 Program Device

生成 bit 文件后就能打开硬件向导，连接开发板后就可以进行下载运行。

实验二 多路选择器的实现

一. 实验内容

这个实验主要使用 Vivado 软件进行多路数据选择器的设计与实现。多路选择器是现代 RISC CPU 内部数据通用选择的重要部件，是基于总线的数据通路的一部分。

二. 实验目标

1. 了解 Vivado 平台开发环境，学习 Verilog HDL 语言。
2. 掌握多路选择器的工作原理和逻辑功能。
3. 了解计算机条件指令的实现。

三. 实验步骤

1. 建立新工程；
2. 设计代码与输入；
3. 代码综合
4. 软件仿真
5. 硬件约束与实现
6. 生成流代码与下载

四. 实验原理

数据选择器是计算机逻辑电路设计中最重要基本逻辑电路之一，也是基于数据选择数据通路 CPU 的重要部件。在 Verilog 硬件描述语言中可以用 if...else 或 case 语句来生成一个数据选择器。

以 2 选 1 和 4 选 1 选择器为例，图 2.2 和图 2.3 为多路选择器的原理图：

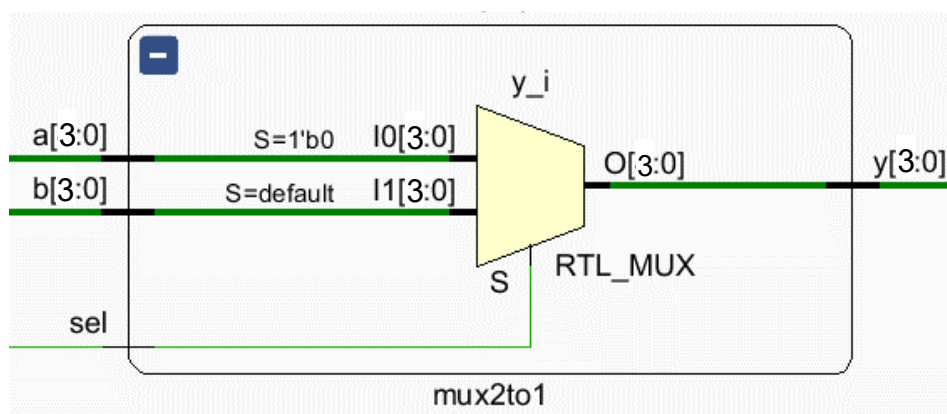


图 2.2 2 选 1

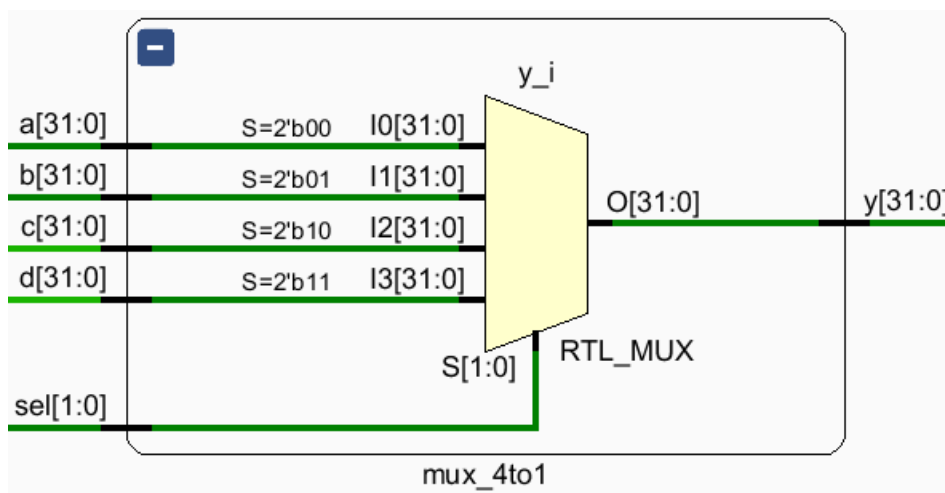


图 2.3 4 选 1

五. 实验步骤

5.1 创建工程

- (1) 双击桌面 Vivado 快捷图标 ，启动 Vivado。
- (2) Create Project 进入 New Project 向导，如图 2.4 所示。

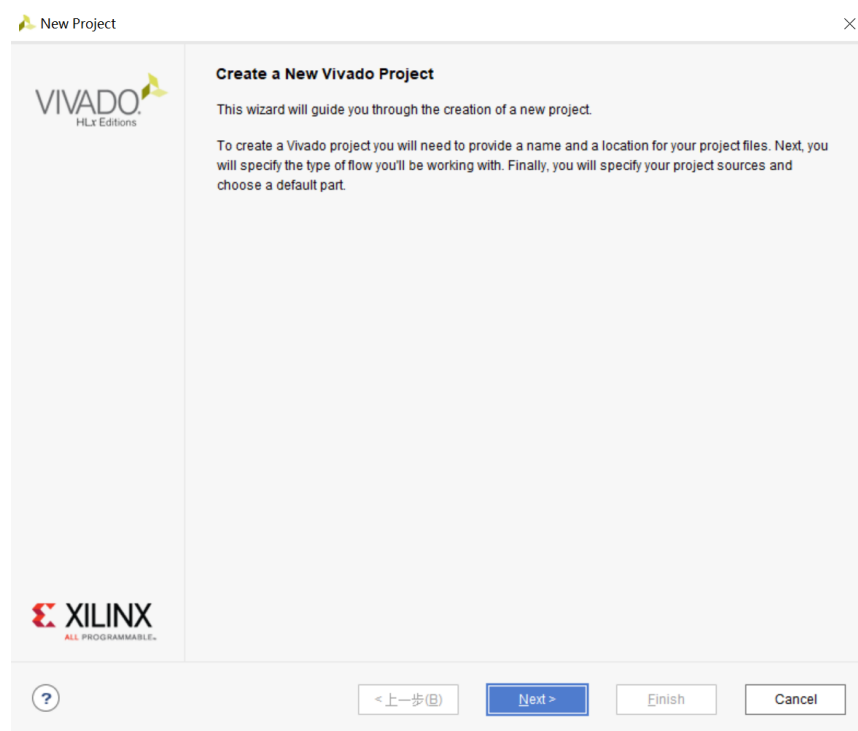


图 2.4 New Project

在 Project Name 中勾选 ☒ Create project subdirectory，如图 2.5 所示：

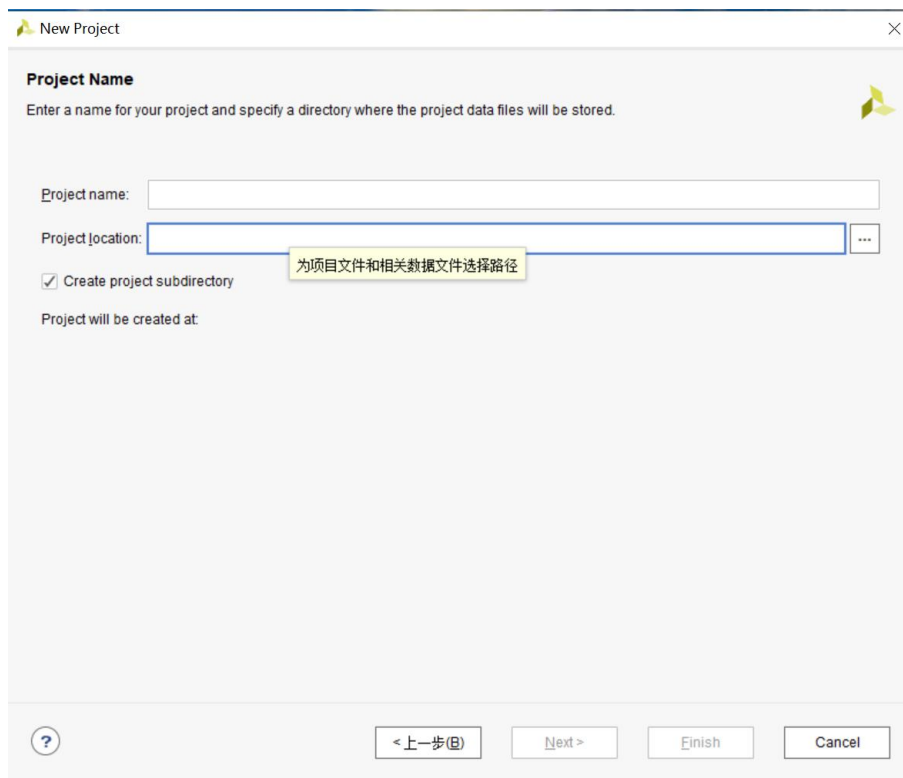


图 2.5 Project Name

在 Project Type 中选 RTL Project，如图 2.6

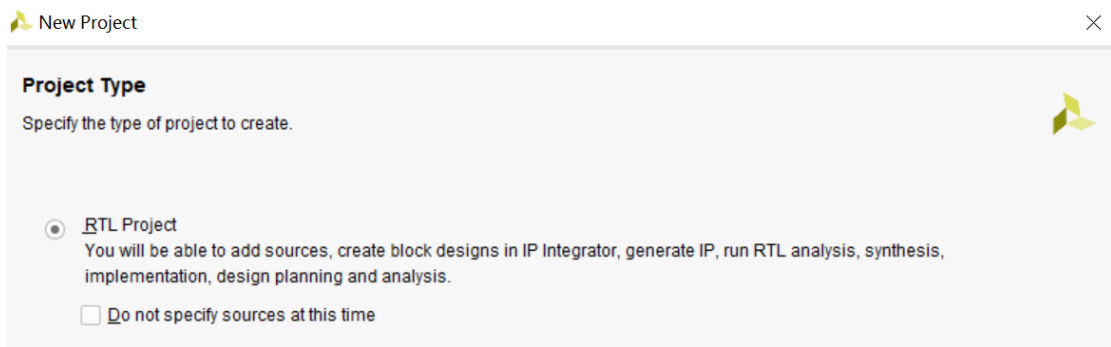


图 2.6 Project Type

在 Add Sources 中选 Verilog 作为目标语言，也可以选自己熟悉的其他语言，如图 2.7。如果有源文件，选 Add File，我们没有，所以选 Create File，如图 2.8。

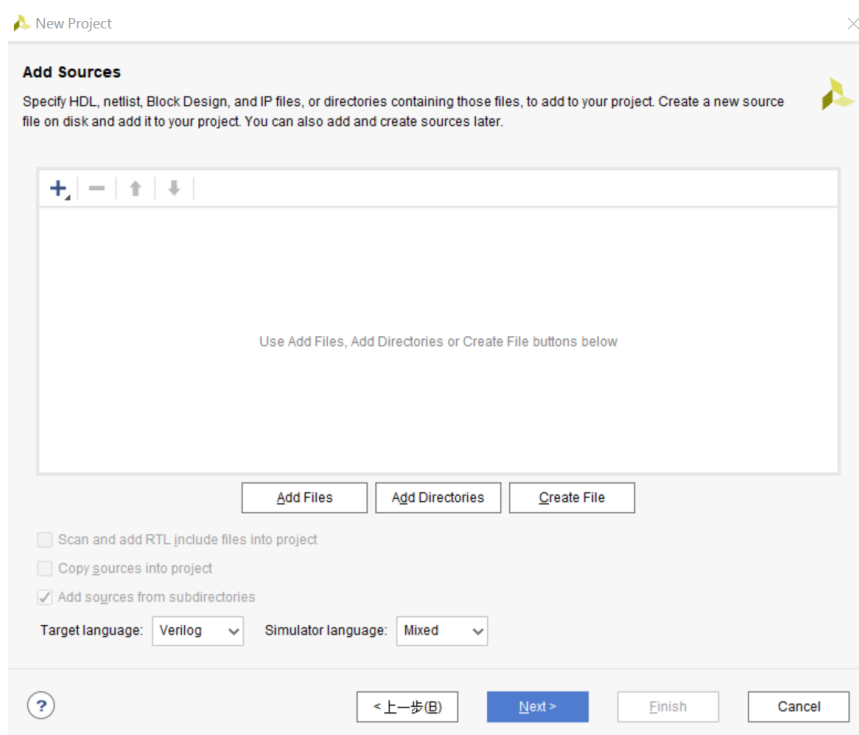


图 2.7 Add Sources

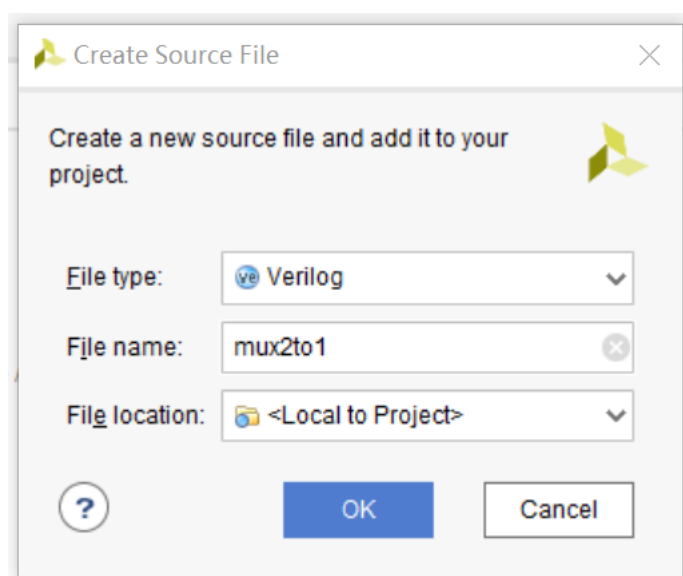


图 2.8 Create File

在 Add Constraints 里单击 Next。

在 Default Part 对话框中选择相应的芯片型号或者硬件平台。我们假设使用 EG01 开发平台，如图 2.9



FPGA数模混合口袋实验平台——EGO1

产品型号：EDK-A7-EGO1

2018-07-18

浏览量: 18330



2.9 EGO1 开发板及芯片规格

具体设置如图 2.10 所示。

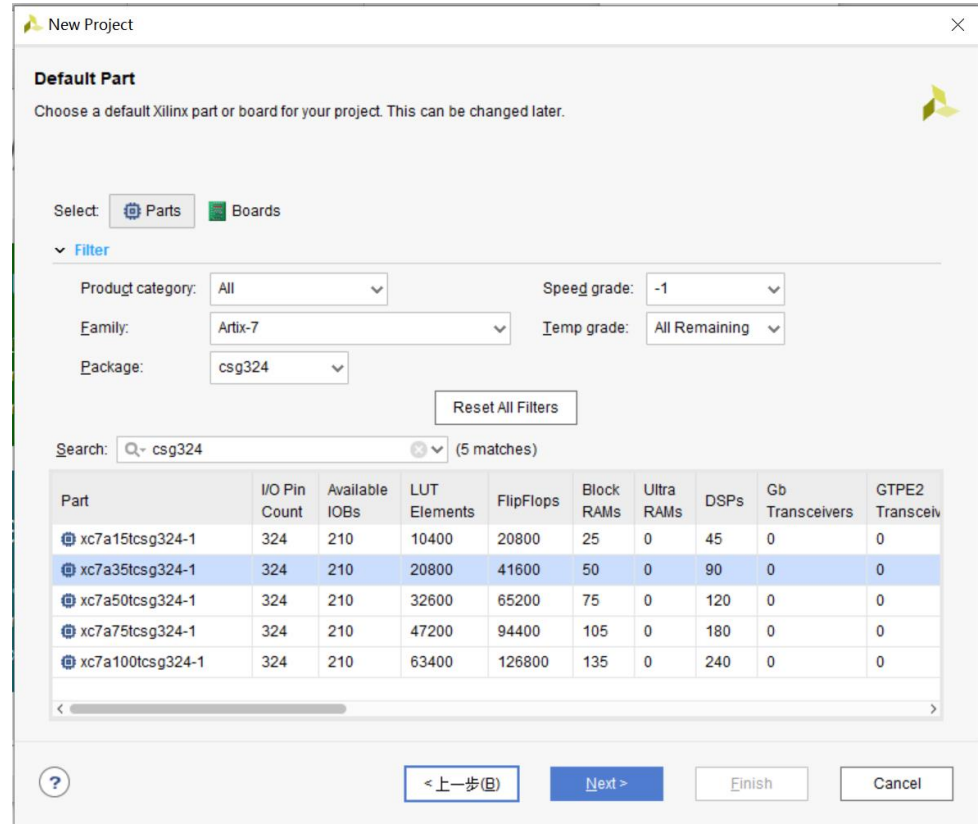


图 2.10 Default Part

Define Module 中 Module Name 栏用于输入模块名，下面的列表框用于端口的定义。Port Name 表示端口名称，Direction 表示端口方向(可选择为 input、output 或 inout)，MSB 表示信号最高位，LSB 表示信号最低位，对于单信号的 MSB 和 LSB 不用填写。当然，端口定义这一步也可以略过，在源程序中再行添加。如图 2.11。

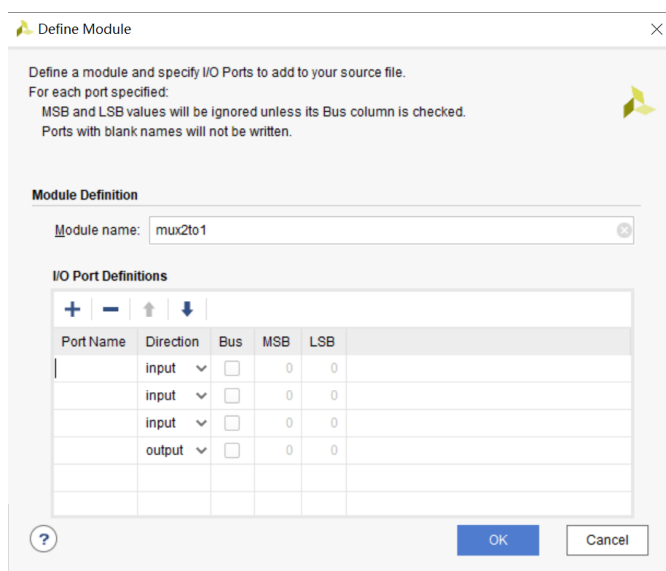


图 2.11 Define Module

定义了模块的端口后，单击 Next 进入下一步，点击 Finish 完成创建，进入主界面。

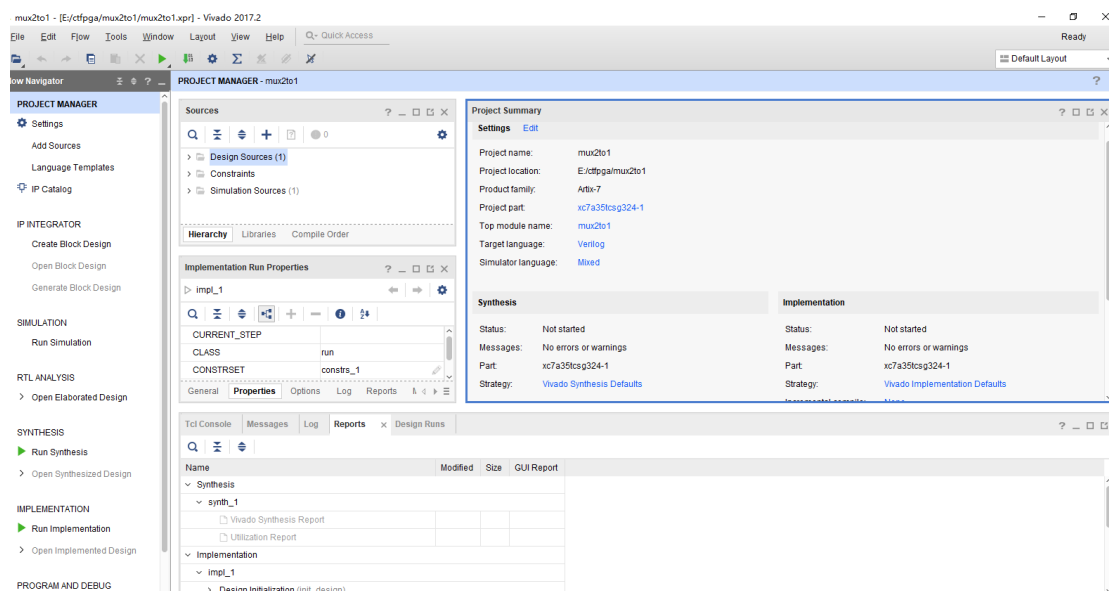


图 2.12 Vivado 工程界面

Vivado IDE 主要工作环境包括菜单栏、工具栏、设计流程导航、数据窗口区、工作空间、结果显示区。

(3) 添加或者创建设计文件，如图 2.13 所示

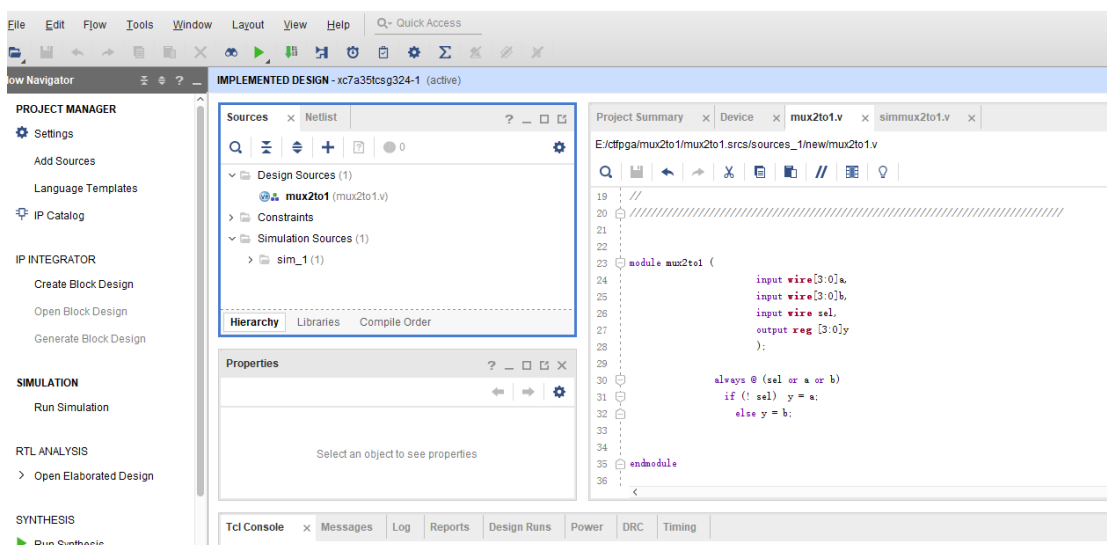


图 2.13 工程设计文件输入

2 选 1 的基本代码：

```
module mux2to1 (  
    input wire[3:0]a,  
    input wire[3:0]b,  
    input wire sel,  
    output reg [3:0]y  
);  
  
    always @ (sel or a or b)  
        if (! sel) y = a;  
        else y = b;  
  
endmodule
```

请仿照以上例子自己设计 4 选 1 或 8 选 1 多路选择器。

5.3 添加约束文件

对于工程中输入的源代码，需要给设计添加管脚和时序约束。管脚约束是将设计文件的输入输出信号设置到器件的某个管脚，（包括设置此管脚的电平标准、电流标准、上下拉特性）。时序约束的作用是为了提高设计的工作频率和获得正确的时序分析报告。

在这个实验中，我们将这一步略去。

5.4 综合与实现

综合就是针对输入设计以及约束条件，按照一定的优化算法进行优化处理，获得一个能满足预期功能的电路设计方案。在 FPGA 设计时，可以用硬件描述语言或者是原理图形式来表示电路的功能。综合工具将这些输入文件翻译成由 FPGA 内部逻辑资源（逻辑单元，RAM 存储单元，时钟单元等）按照某种连接方式组成的逻辑连接（网表），并根据用户的要求生成网表文件，这一过程称为综合过程。方法：Flow Navigator→Synthesis→Run Synthesis 进行工程综合。如图 2.14.

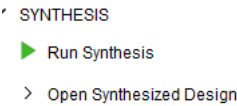


图 2.14 综合选项

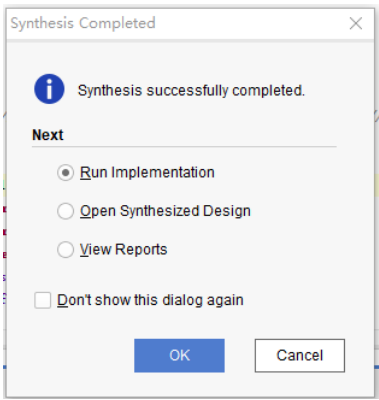


图 2.15 综合后的弹出框

实现就是通过翻译、映射、布局布线等过程来完成设计的固化。实现过程首先将综合生成的网表（Netlist）文件,通过翻译变成所选器件的内部资源和硬件单元，如可配置逻辑块（CLB），数字时钟单元（DCM），存储单元（RAM）等，这个步骤称为翻译过程；然后找到对应的硬件关系，将设计与这些硬件资源关系一一对应起来，这又称为映射过程（Map）；最后进行布局布线(Place&Route)，这样设计基本上就可以完全固化到 fpga 当中了。选择 Run Implementation，执行实现过程。

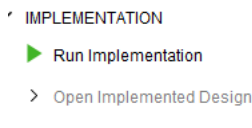


图 2.16 实现选项

5.5 仿真并查看波形

(1) 添加仿真文件，点图 2.17 的 “+”

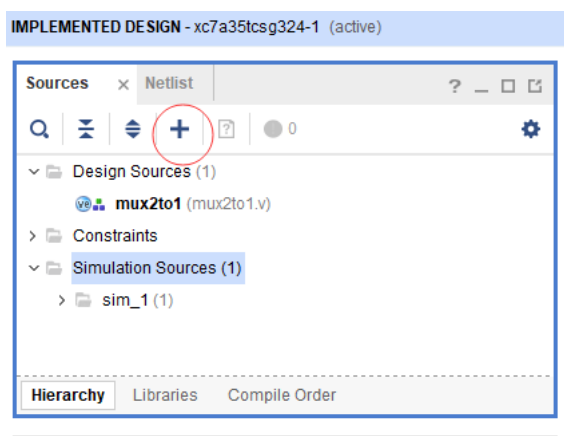


图 2.17 添加仿真文件

(2) 设置仿真文件

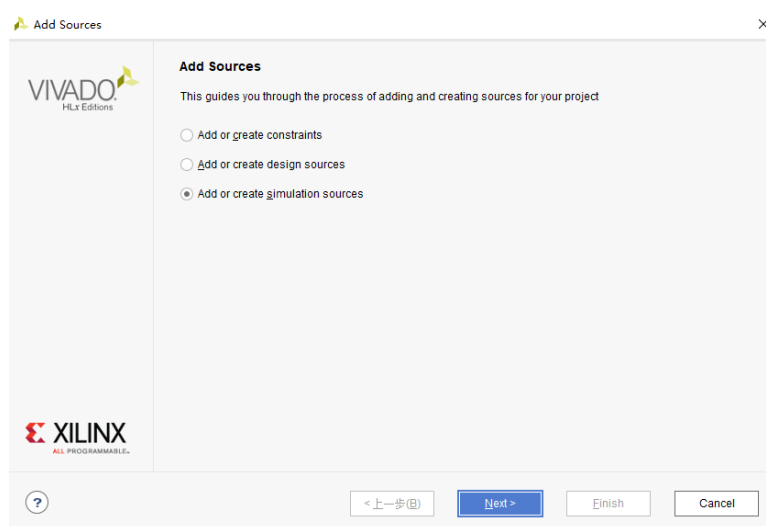


图 2.18 Add Source

(3) 建立仿真文件

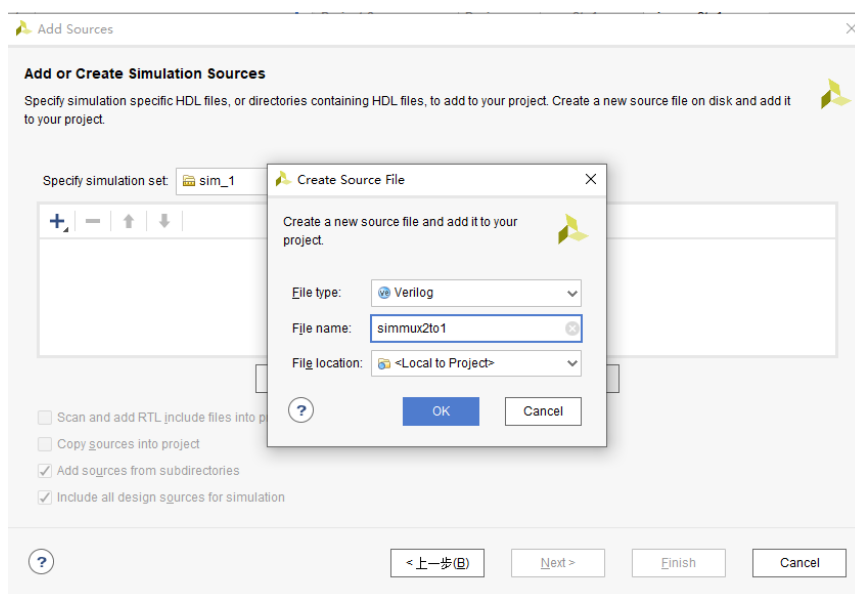


图 2.19 建立仿真文件

(4) 添加用于仿真的测试程序

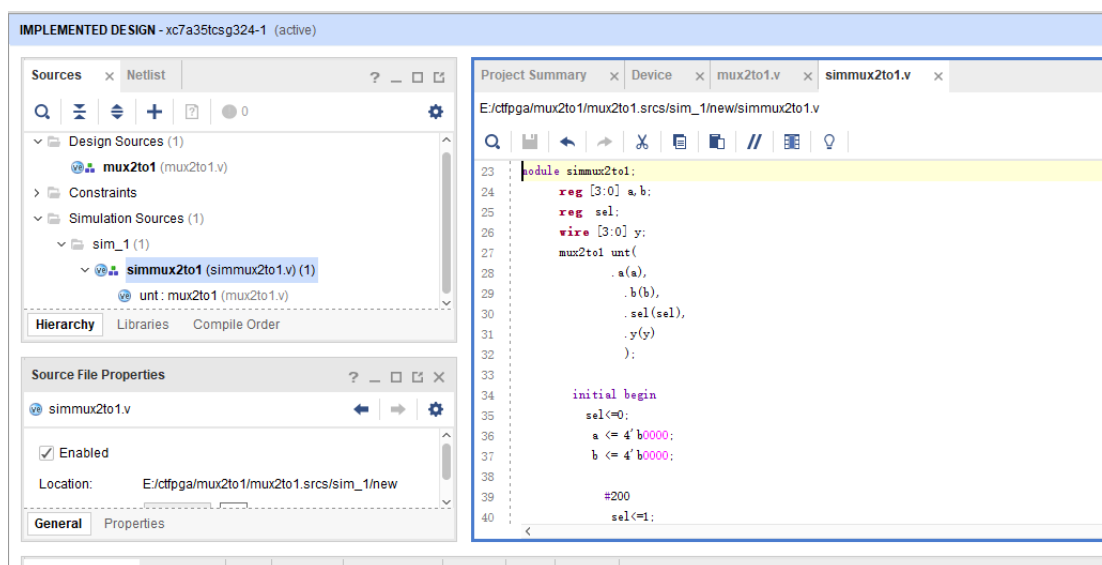


图 2.20 编写测试程序

测试代码如下：

```
module simmux2to1;
    reg [3:0] a,b;
    reg sel;
    wire [3:0] y;
    mux2to1 unt(
        .a(a),
        .b(b),
```

```

        .sel(sel),
        .y(y)
    );

initial    begin
    sel<=0;
    a <= 4'b0000;
    b <= 4'b0000;

    #200
    sel<=1;
    a <= 4'b0001;
    b <= 4'b1000;

    #200
    sel<=0;
    a <= 4'b0010;
    b <= 4'b0100;

    #200
    sel<=1;
    a <= 4'b0011;
    b <= 4'b1100;

end

endmodule

```

(5) 仿真

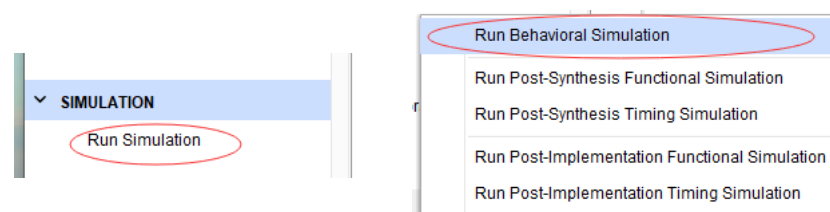


图 2.21 仿真

(6) 分析波形

根据波形中的逻辑值分析是否符合逻辑设计的功能。

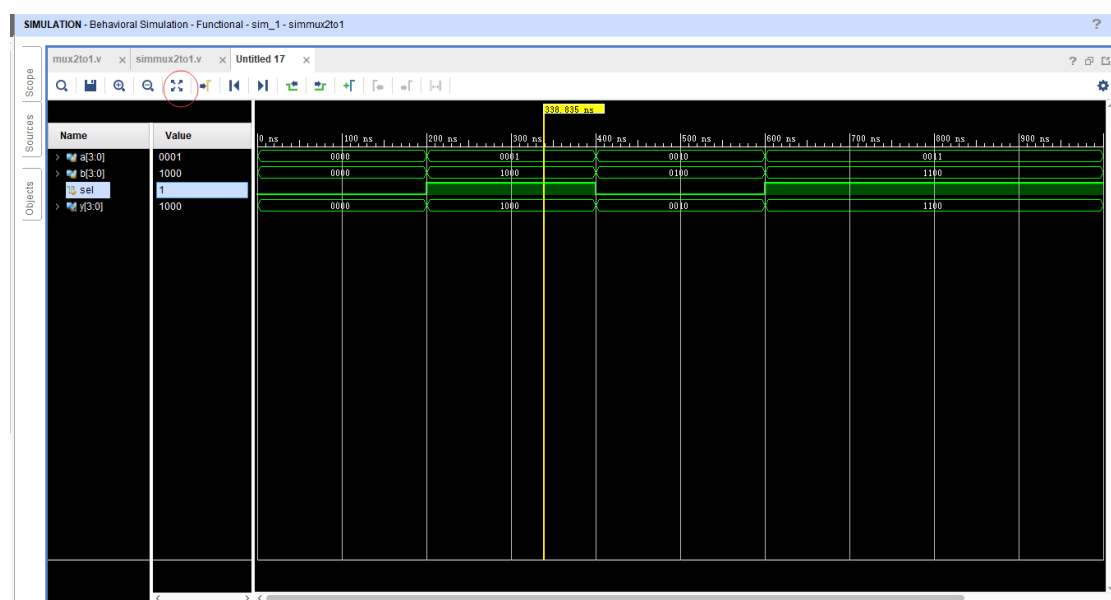
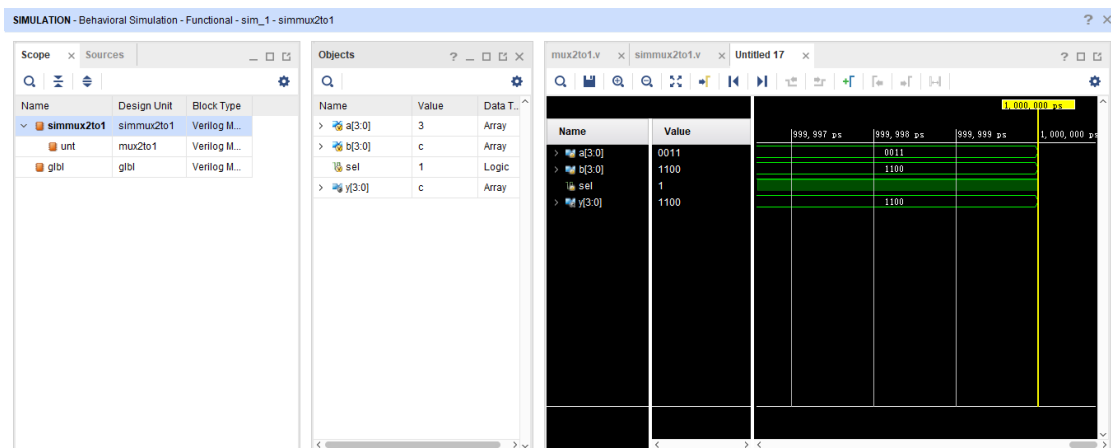


图 2.22 分析波形

查看原理图，如图 2.23 所示。

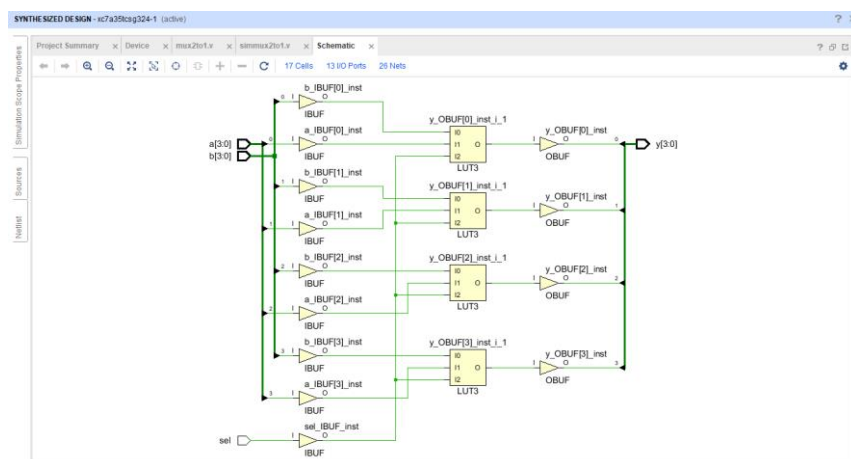


图 2.23 原理图

5.6 编译文件及下载

只有编译文件才能配置 FPGA，因此在完成综合之后，还要将生成的网表文件转换成可配置的 FPGA 文件。在实现过程完成后，用户可以查看设计实现的结果或者实现报告。如果需要继续进行生成编译时可选择生成 bitstream，点击 OK，执行编译过程。（在这个实验中因为没有做约束，也没有开发板用于下载，因此这一步可以跳过。）

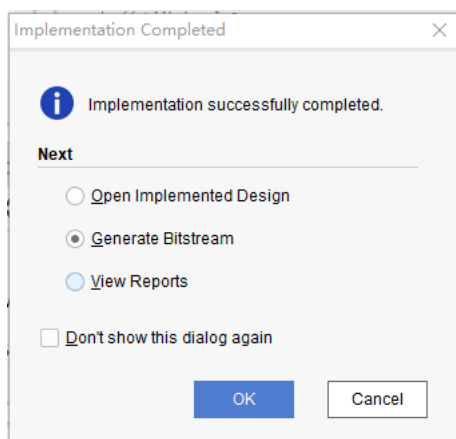


图 2.16 弹窗