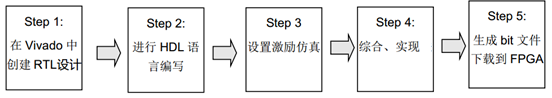
Vivado设计流程

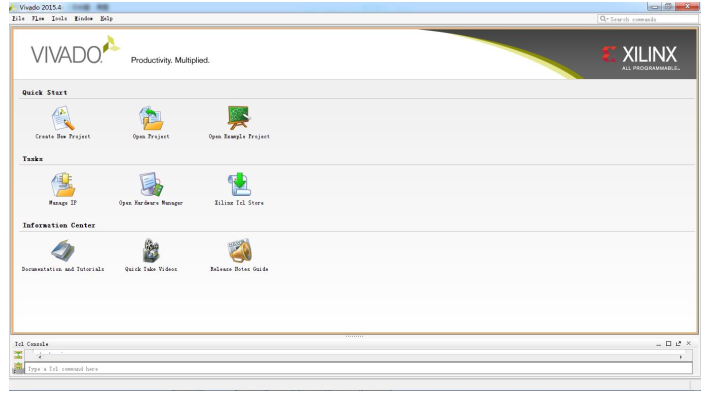
Vivado 设计分为 Project Mode 和 Non-project Mode （即tcl脚本模式）两种模式，一般设计中，我们常用的是 Project Mode。我们将以一个简单的实验案例，一步一步的完成 Vivado 的整个设计流程。  
在本次实验中，将会学习如何使用 Xilinx Vivado 创建工程、编写设计文件、编写testbench、 行为仿真、综合、实现等功能。通过编写一个4位加法器来展示使用 Xilinx Vivado 来进行基本的 FPGA 设计。  
目标：

* 熟悉vivado设计流程
* 掌握利用Vivado创建设计的方法
* 掌握行为仿真方法

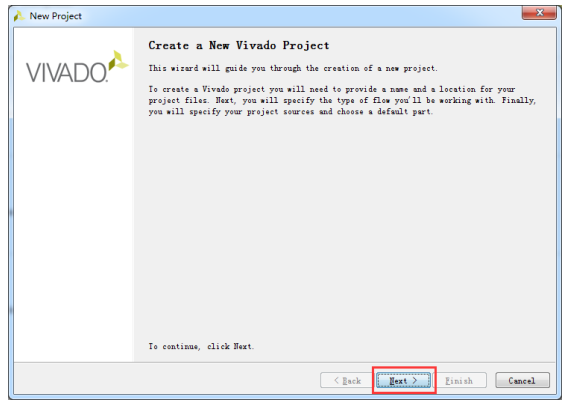
Vivado设计流程



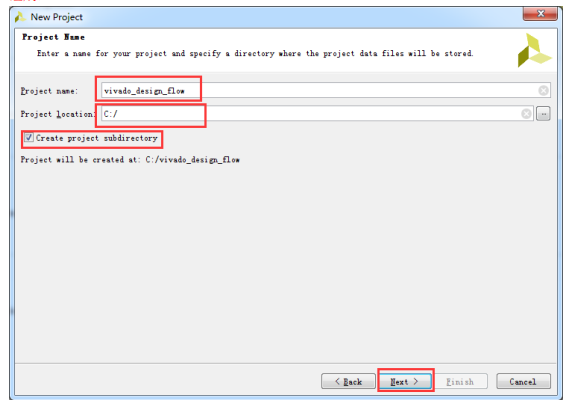
1. 新建工程  
   1、打开 Vivado 2015.4 开发工具，可通过桌面快捷方式或开始菜单中 Xilinx Design Tools->Vivado  
   2014.2 下的 Vivado 2015.4 打开软件，开启后，软件如下所示：



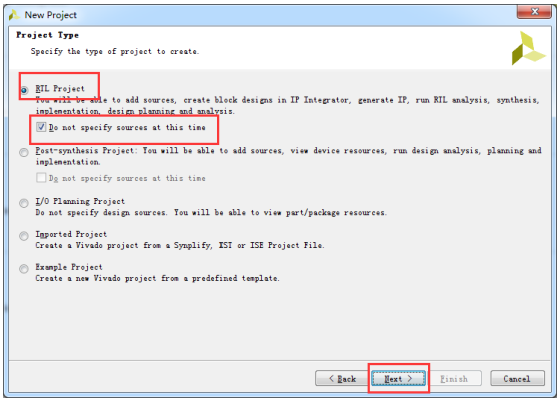
2、单击上述界面中 Create New Project 图标，弹出新建工程向导，点击 Next。



3、输入工程名称、选择工程存储路径，并勾选 Create project subdirectory 选项，为工程在指定存储路径下建立独立的文件夹。设置完成后，点击 Next。  
注意：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。

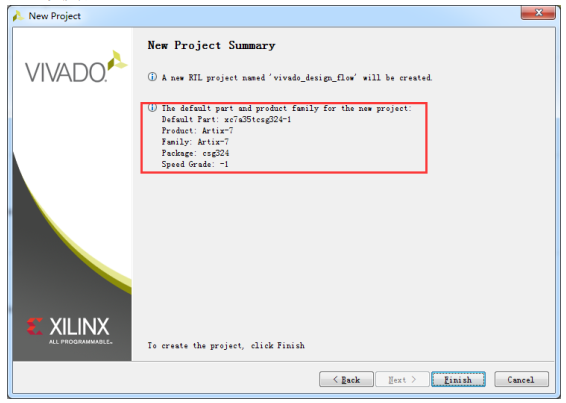


4、选择 RTL Project 一项，并勾选 Do not specify sources at this time，勾选该选项是为了跳过在新建工程的过程中添加设计源文件。点击 Next。

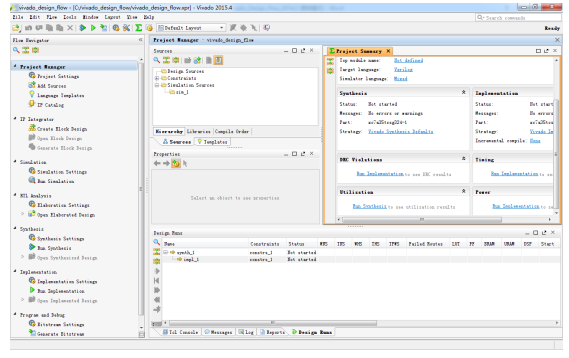


5、根据使用的 FPGA 开发平台，选择对应的 FPGA 目标器件。 在本手册中，用默认的设置就行。

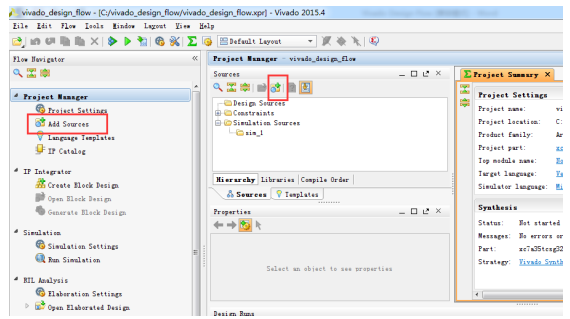
6、确认相关信息与设计所用的的 FPGA 器件信息是否一致，一致请点击 Finish，不一致，请返回上一步修改。



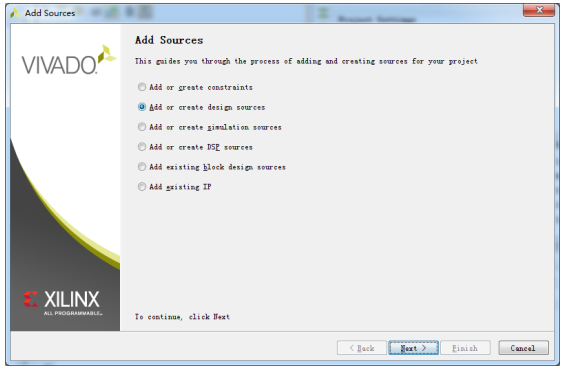
7、得到如下的空白 Vivado 工程界面，完成空白工程新建。



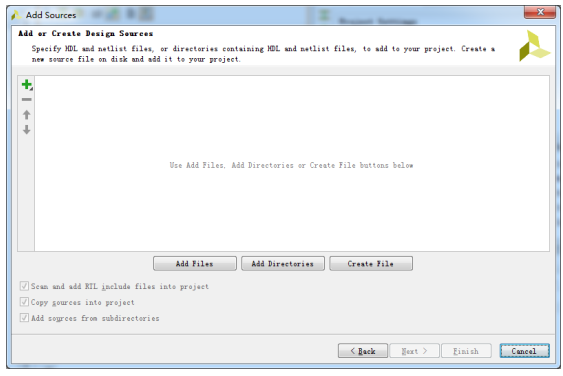
二、设计文件输入  
1、如下图所示，点击 Flow Navigator 下的 Project Manager->Add Sources 或中间 Sources 中的对话框打开设计文件导入添加对话框。



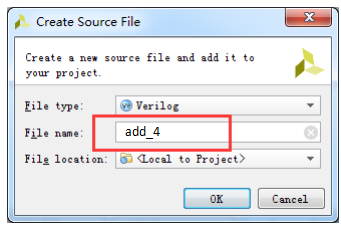
2、选择第二项 Add or Create Design Sources，用来添加或新建 Verilog 或 VHDL 源文件，点击Next。



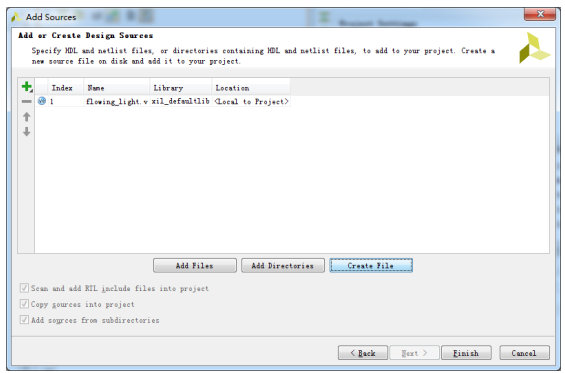
3、如果有现有的 V/VHD 文件，可以通过 Add Files 一项添加。在这里，我们要新建文件，所以选择 Create File 一项。



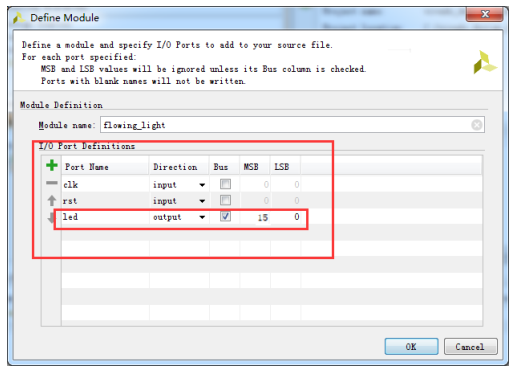
4、在 Create Source File 中输入 File Name，点击 OK。 注：名称中不可出现中文和空格。



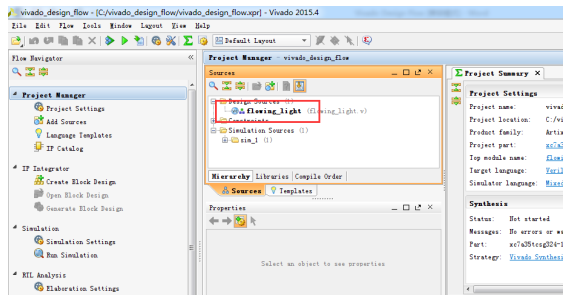
5、点击 Finish。

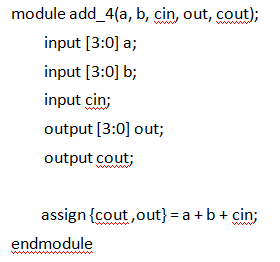


6、在弹出的 Define Module 中的 I/O Port Definition，输入设计模块所需的端口，并设置端口方向，如果端口为总线型，勾选 Bus 选项，并通过 MSB 和 LSB 确定总线宽度。完成后点 OK。注： led 实际宽度与代码中一致，也可在代码中修改。端口的名称、位宽等信息应根据具体的电路填写，下图只是个示例，不是4位加法器的端口。

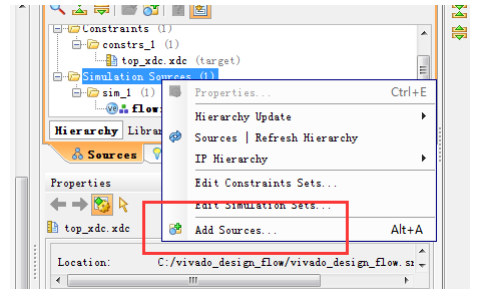


7、新建的设计文件（此处为add\_4.v）即存在于 Sources 中的 Design Sources 中。双击打开该文件，输入相应的设计代码。

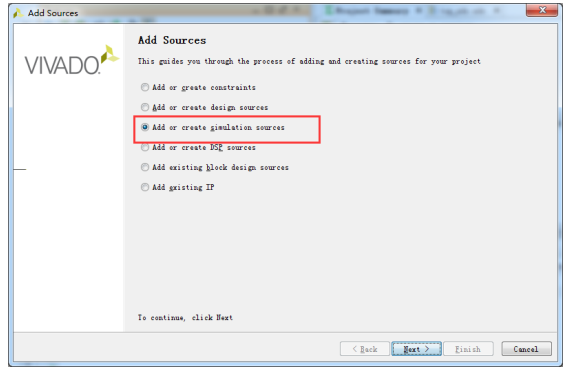




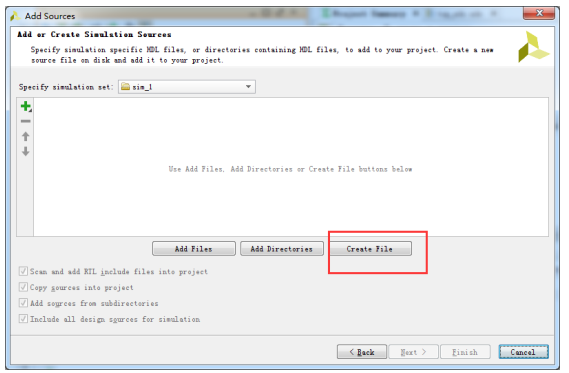
1. 利用 Vivado 进行功能仿真  
   1、创建激励测试文件，在 Source 中右击选择 Add Source



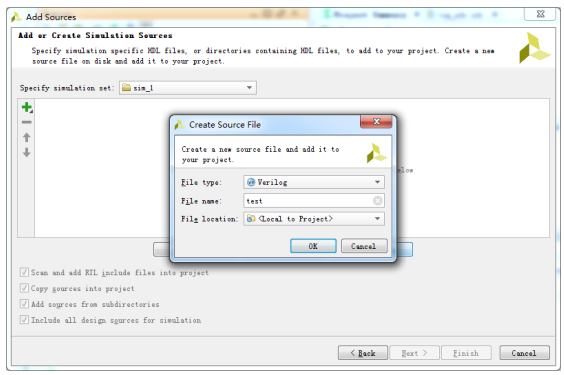
2、在 Add Source 界面中选择第三项 Add or Create Simulation Source,点击 Next。



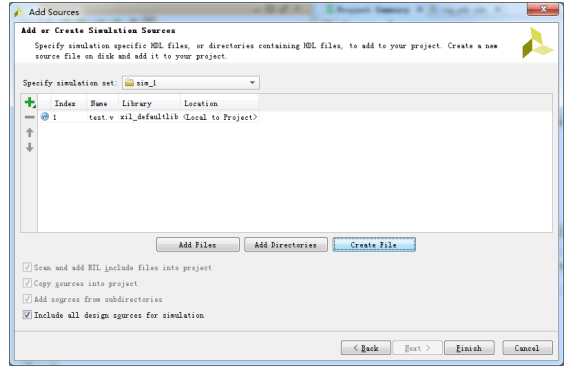
3、选择 Create File 创建一个仿真激励文件

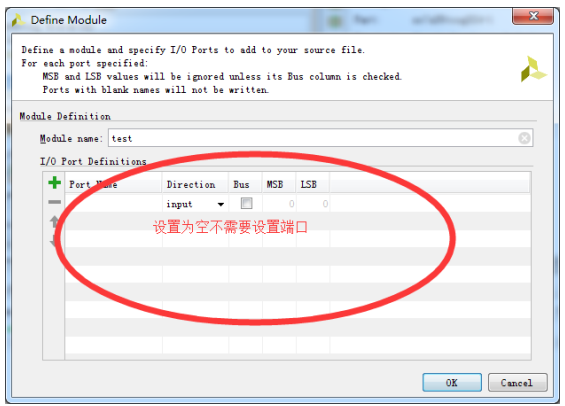


4、输入激励文件名称，点击 OK。



5、 确认添加完成之后点击 Finish，因为是激励文件不需要对外端口，所以直接 Port 部分直接空着，点击 ok。





5、在 Source 下双击打开空白的激励测试文件，完成对将要仿真的 module 的实例化和激励代码的编写，如下图和下述代码所示:

`timescale 1ns / 1ps

module test\_top;

reg cin;

reg [3:0] a;

reg [3:0] b;

wire [3:0] out;

wire cout;

reg clk;

initial

begin

clk = 0;

a = 0;

b = 0;

cin = 0;

#10000;

$finish;

end

always #50 clk = ~clk;

always@(posedge clk)

begin

a = {$random} % 16;

b = {$random} % 16;

cin = {$random} % 2;

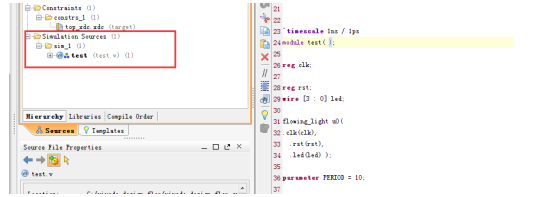
$display("a=%d, b=%d, cin=%d, out=%d, cout=%d\n", a, b, cin, out, cout);

end

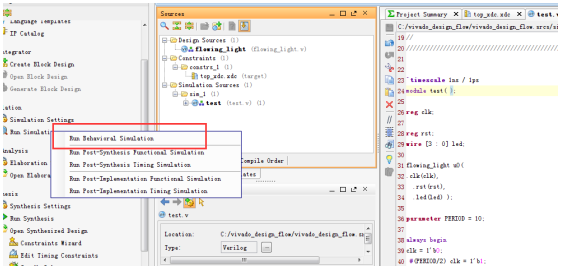
add\_4 u\_add\_4(.a(a),.b(b),.cin(cin),.out(out),.cout(cout));

endmodule

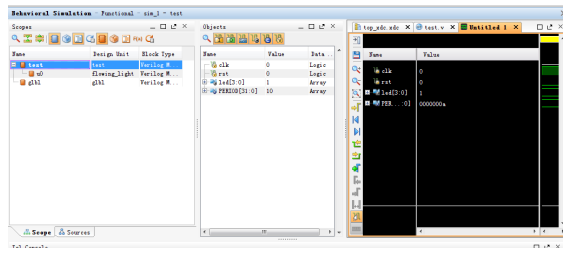
激励文件完成之后，工程目录如下图所示。



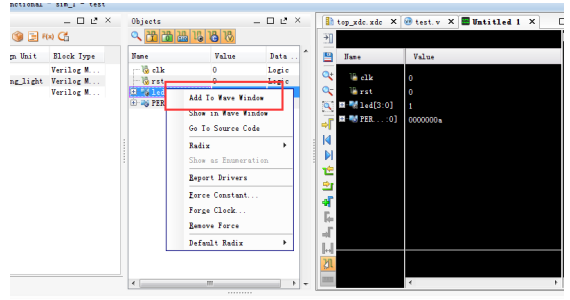
6、此时，进入仿真。在左侧 Flow Navigator 中点击 Simulation 下的 Run Simulation 选项，并选择 Run Behavioral Simulation 一项，进入仿真界面。



7、下图所示为仿真界面。



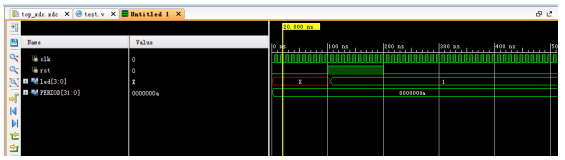
可通过左侧 Scope 一栏中的目录结构定位到设计者想要查看的 module 内部寄存器，在 Objects对应的信号名称上右击选择 Add To Wave Window，将信号加入波形图中。



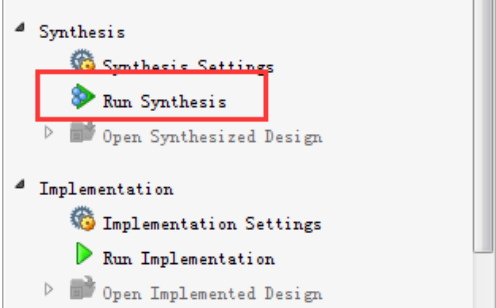
可通过选择工具栏中的如下选项来进行波形的仿真时间控制。如下工具条，分别是复位波形（即清空现有波形）、运行仿真、运行特定时长的仿真、仿真时长设置、仿真时长单位、单步运行、暂停



8、最终得到的仿真效果图如下。核对波形与预设的逻辑功能是否一致。仿真完成。



1. 运行综合（Run Synthesis）& 运行实现（Run Implementation）



综合完成之后，选择Run Implementation。

