



# Vivado 设计流程

**Vivado 2013.4 Version**

# 目标

## ➤ 完成了这个单元之后，你将能够

- 阐述Vivado IDE的设计分析功能如何能帮助FPGA设计开发
- 列举Vivado IDE的主要特性
- 描述Vivado IDE设计流程
- 引入脚本化的Vivado IDE设计流程

# 目录

- **Vivado IDE 功能和收益**
- **Vivado 设计套件介绍**
- **Vivado 设计流程**
- **介绍**

# Vivado IDE 方案

## ➤ 交互式设计和分析

- 时序分析、连接性、资源利用、时序约束分析和入口

## ➤ RTL开发与分析

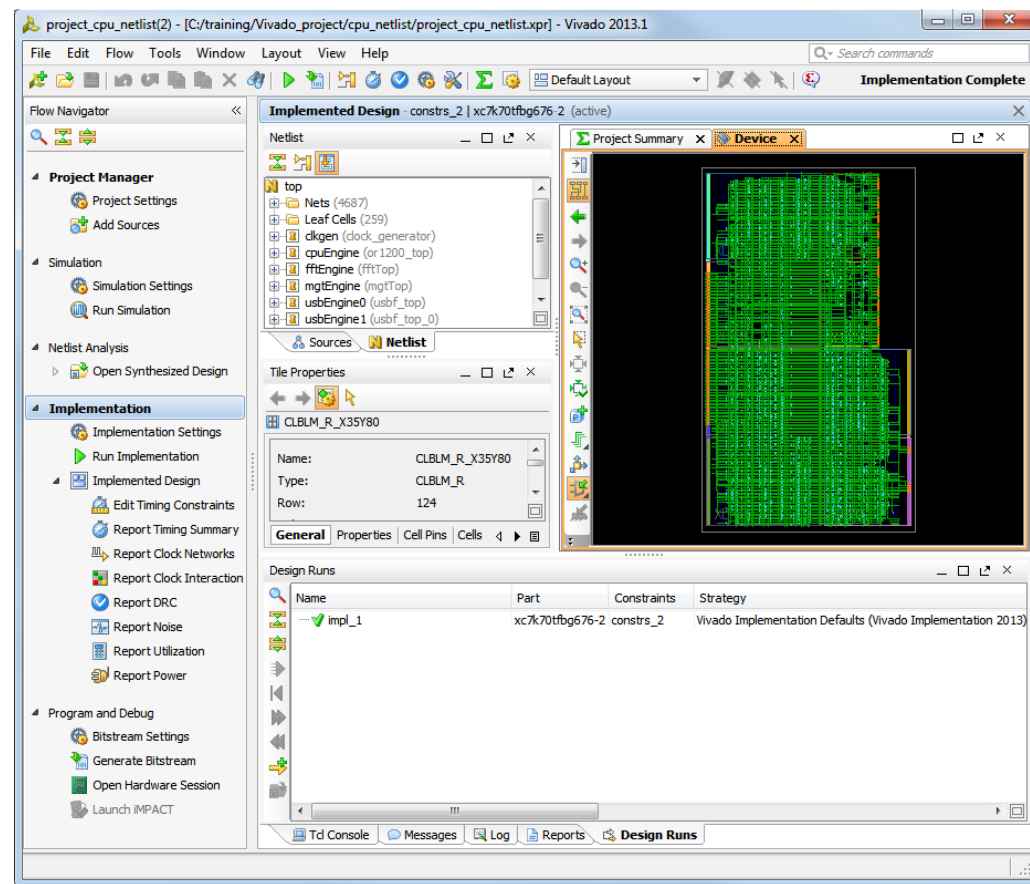
- 精致的HDL
- 逐层探索
- 产生电路图

## ➤ XSIM仿真器集成

## ➤ 在一个封装中合成和实现

## ➤ I/O引脚规划

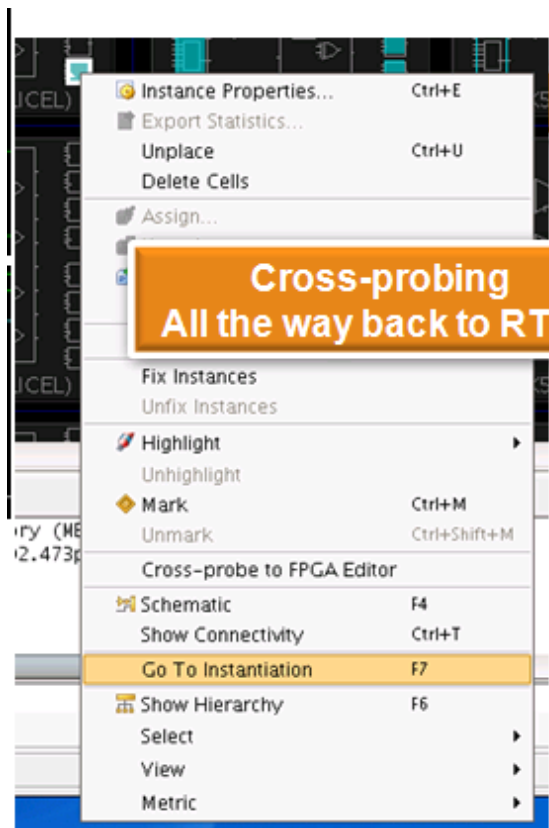
- 交互式基于规则的I/O分配



分层的设计分析和实现环境

# Vivado's 可视化特性

- 在流程的任何阶段都可以可视化和调试你的设计
  - 在网络表/电路图/RTL之间交叉对比

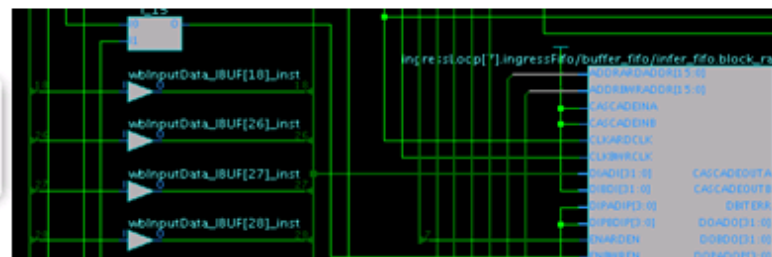
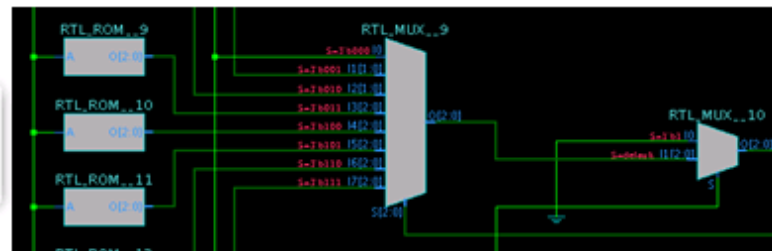


**Cross-probing  
All the way back to RTL!!**

**Elaborated  
RTL**

**Netlist  
Schematic**

**Implemented  
Design**



# Tcl 特性

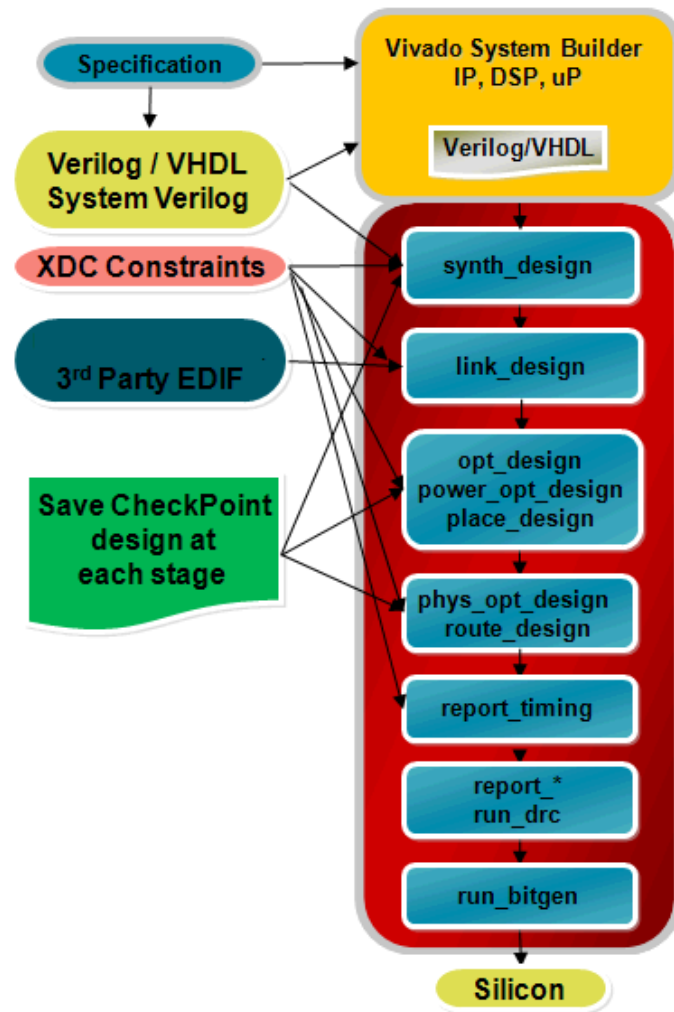
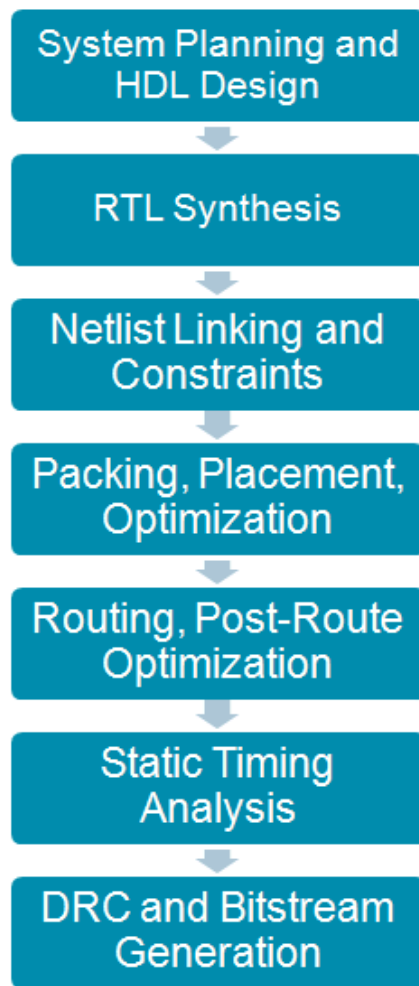
- **Tcl**终端使得设计者得以主动查询设计的网络表
- 完整的**Tcl**脚本支持两种设计流程
  - Vivado IDE为基于项目的设计流程提供方便的项目管理
  - 非项目的批处理设计流程使得整个流可以在内存中执行
- 日记和日志可以用来做脚本构建
- **Tcl**脚本在本课程最后一个单元来介绍

# 目录

- **Vivado IDE 功能和收益**
- **Vivado 设计套件介绍**
- **Vivado 设计流程**
- **介绍**

# Vivado设计流程vs传统流程

- 交互式IP即插即用环境
  - AXI4, IP\_XACT
- 贯穿流程的普通约束语言（XDC）
  - 在任何阶段施加约束
- 在任何阶段报告
  - 强壮的Tcl API
- 贯穿流程的普通数据模型
  - “内存中”模型改进了速度
  - 在所有阶段产生报告
- 在任何阶段保存关键点设计
  - 网络表、约束、位置和连结结果





# 什么是网络表？

➤ 网络表是你的设计的一种描述

➤ 包括单元、引脚、端口和网络

- 单元是设计目标

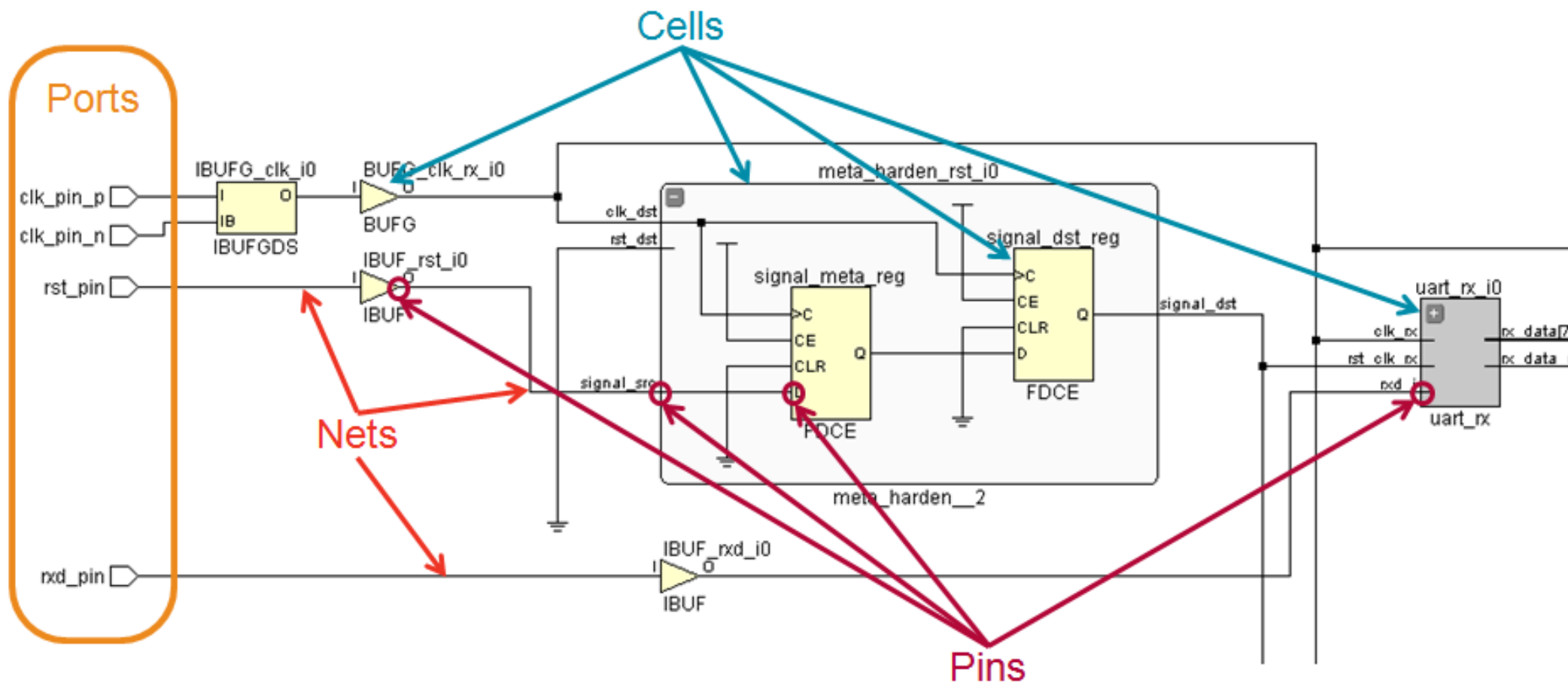
- 用户模块/设计内容的实例表达
- 库单元（BEL）的实例
  - LUT, FF, RAM, DSP单元等...
- 硬件功能的通用技术表达
- 黑盒

- 引脚是单元的连结点

- 端口是设计的顶层端口

- 网络形成引脚之间及引脚到端口的连结

# 网络表对象



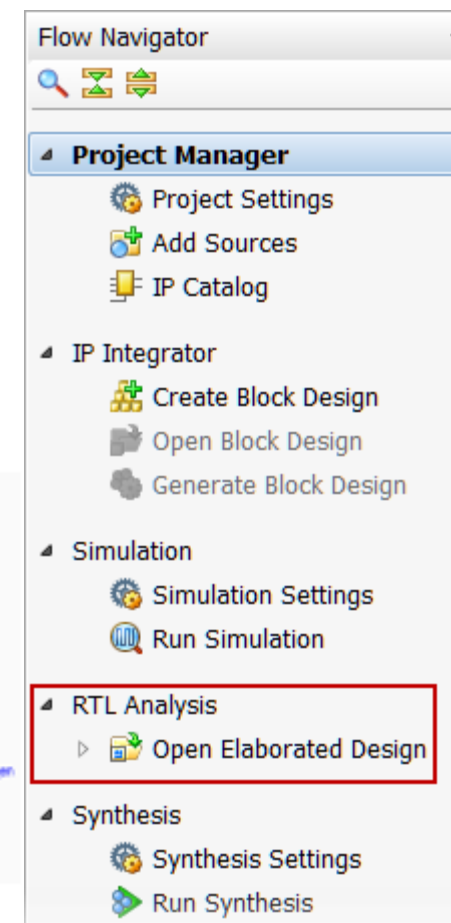
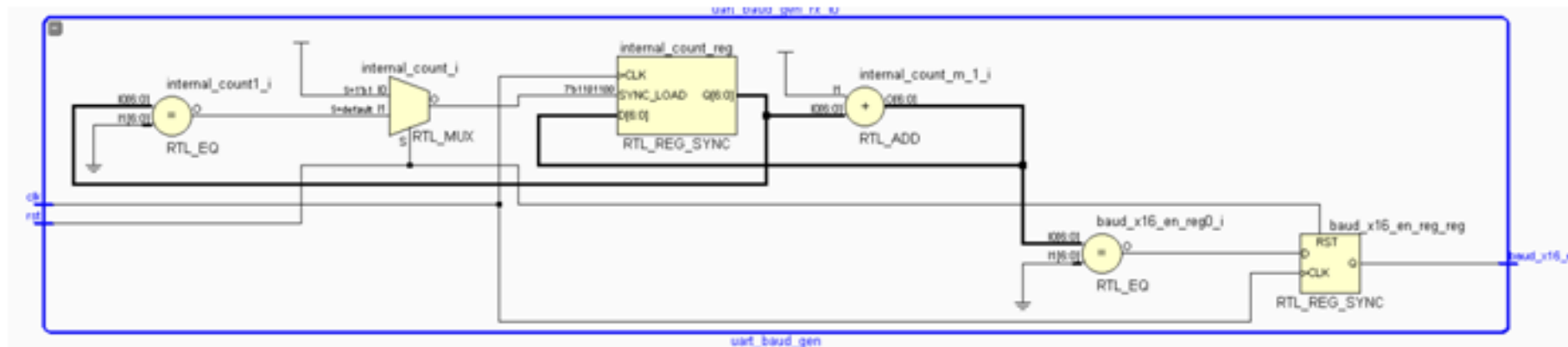
# 详细设计

➤ 选择**Open Elaborated Design**（打开详细设计）来使用流程导航

➤ 在综合之前表述设计

– 层级通用的单元之间互联的网络表

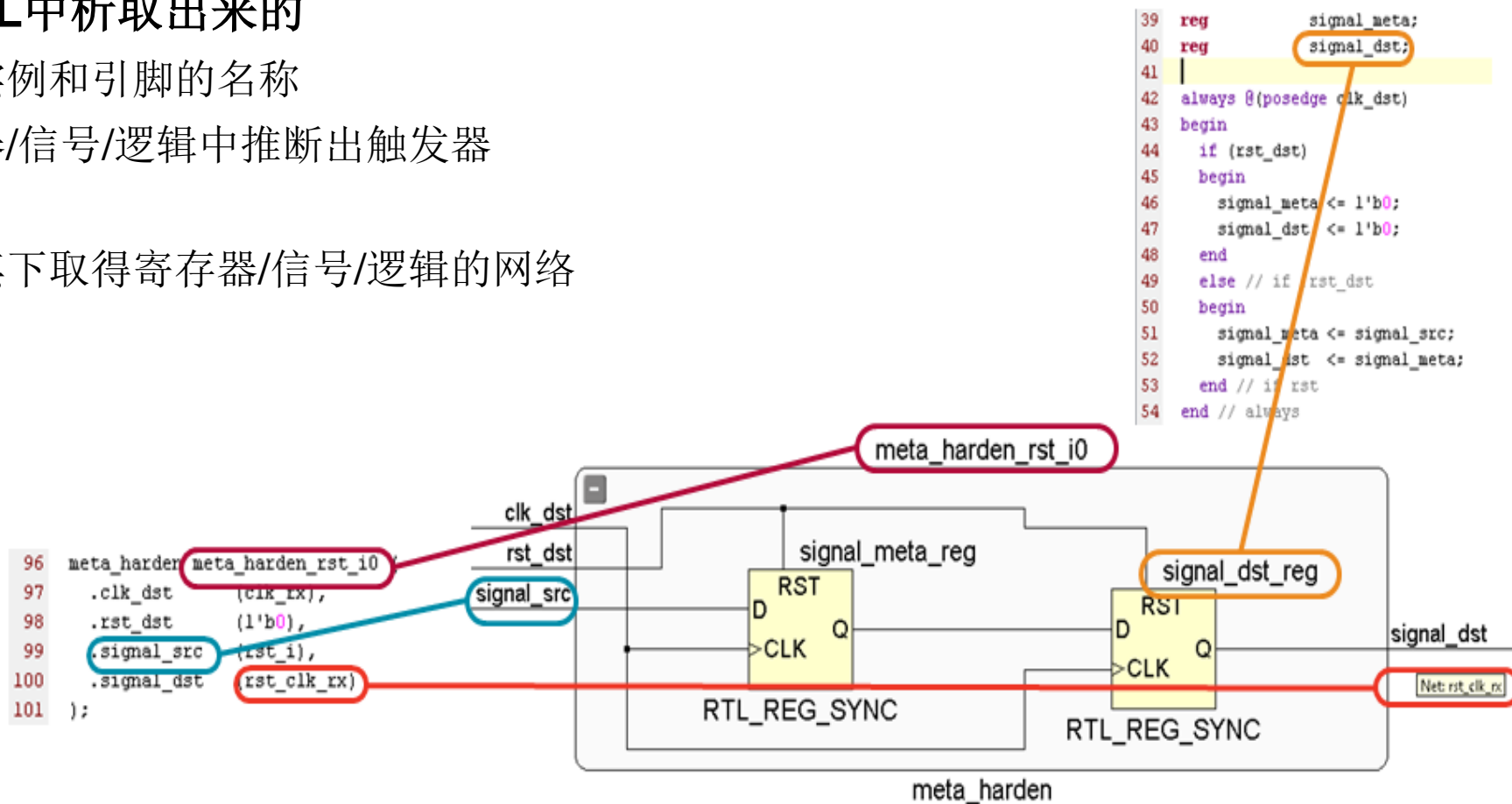
- 单元/设计项的实例
- 硬件部件的通用技术表述
  - AND, OR, 缓冲, 复用器, 加法器, 比较器等



# 详细设计中的对象名称

## ➤ 对象名称是从RTL中析取出来的

- 层级中对象的实例和引脚的名称
- 从其下的寄存器/信号/逻辑中推断出触发器
  - 加上\_reg后缀
- 可能的话，从其下取得寄存器/信号/逻辑的网络



# 综合设计

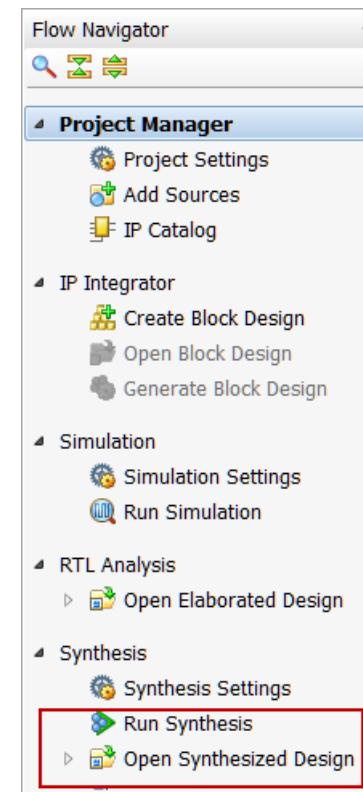
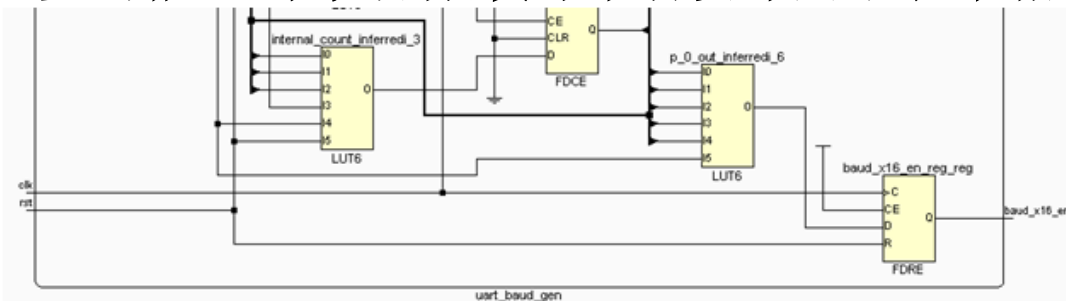
➤ 选择**Open Synthesized Design**（打开综合设计）来使用流程导航

➤ 在合成之后表达设计

– 层级和基础元件（BEL）的互连网络表

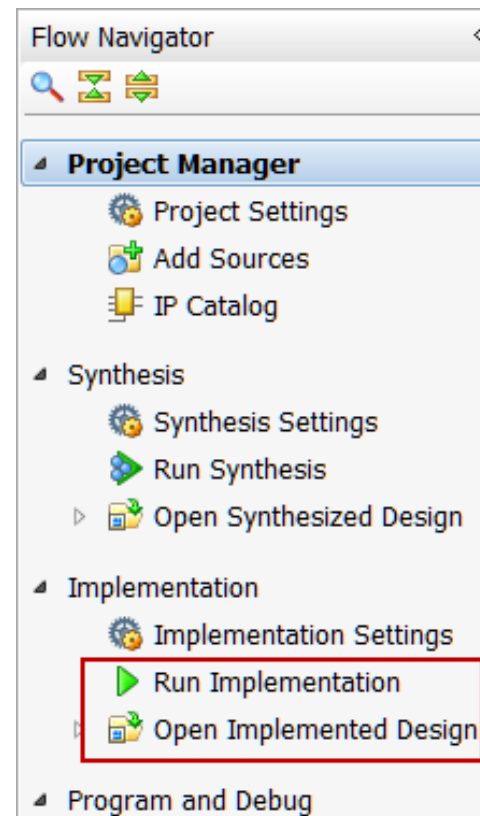
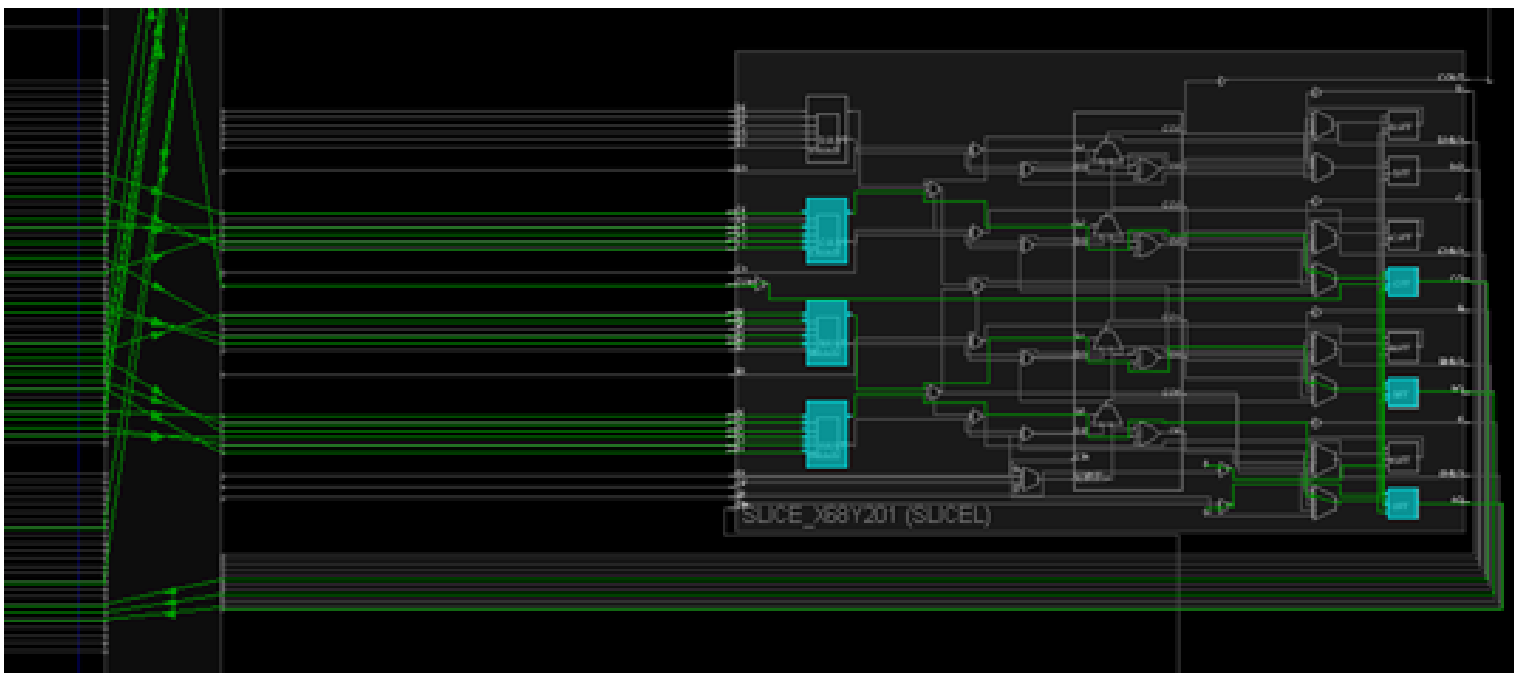
- 模块/设计项的实例
  - LUT, 触发器, 进位链元件, 宽复用器
  - 块RAM, DSP单元
  - 时钟元件 (BUFG, BUFR, MMCM, ...)
  - I/O 元件 (IBUF, OBUF, I/O 触发器)
- 基础元件

➤ 只要可能，对象名称采用详细设计网络表中相同的名称



# 实现设计

- 选择**Open Implemented Design**（打开实现设计）来使用流程导航
- 在实现过程其间和之后表示设计
  - 结构上和综合设计类似
  - 单元有位置，网络映射到具体的连线上



# 项目数据

- 项目的所有数据保存在以项目名称为名字的目录中，包括
  - *project\_name.xpr* 文件: 用来打开项目的文件（Vivado IDE 项目文件）
  - *project\_name.runs* 目录: 包括所有的运行数据
  - *project\_name.srcs* 目录: 包括所有引入的本地HDL源文件、网络表和XDC文件
  - *project\_name.data* 目录: 保存平面图和网络表数据

# 日记和日志文件

## ➤ 日记文件 (*vivado.jou*)

- 只包含Vivado IDE执行的Tcl命令

## ➤ 日志文件 (*vivado.log*)

- 包含Vivado IDE产生的所有消息，包括Tcl命令及结果，信息/警告/错误消息等

## ➤ 位置

- Linux: Vivado IDE启动时的目录
- Windows双击图标启动的: *%APPDATA%\Xilinx\Vivado* 或 *C:\Users\<user\_name>\AppData\Roaming\Xilinx\Vivado*
- Windows从命令行启动的: Vivado IDE启动时的目录
- 从GUI中
  - 选择File > Open Log File
  - 选择File > Open Journal File



# 目录

- **Vivado IDE 功能和收益**
- **Vivado 设计套件介绍**
- ***Vivado 设计流程***
- **介绍**

# 从跳转页开始

➤ 创建新项目或打开已有项目的链接

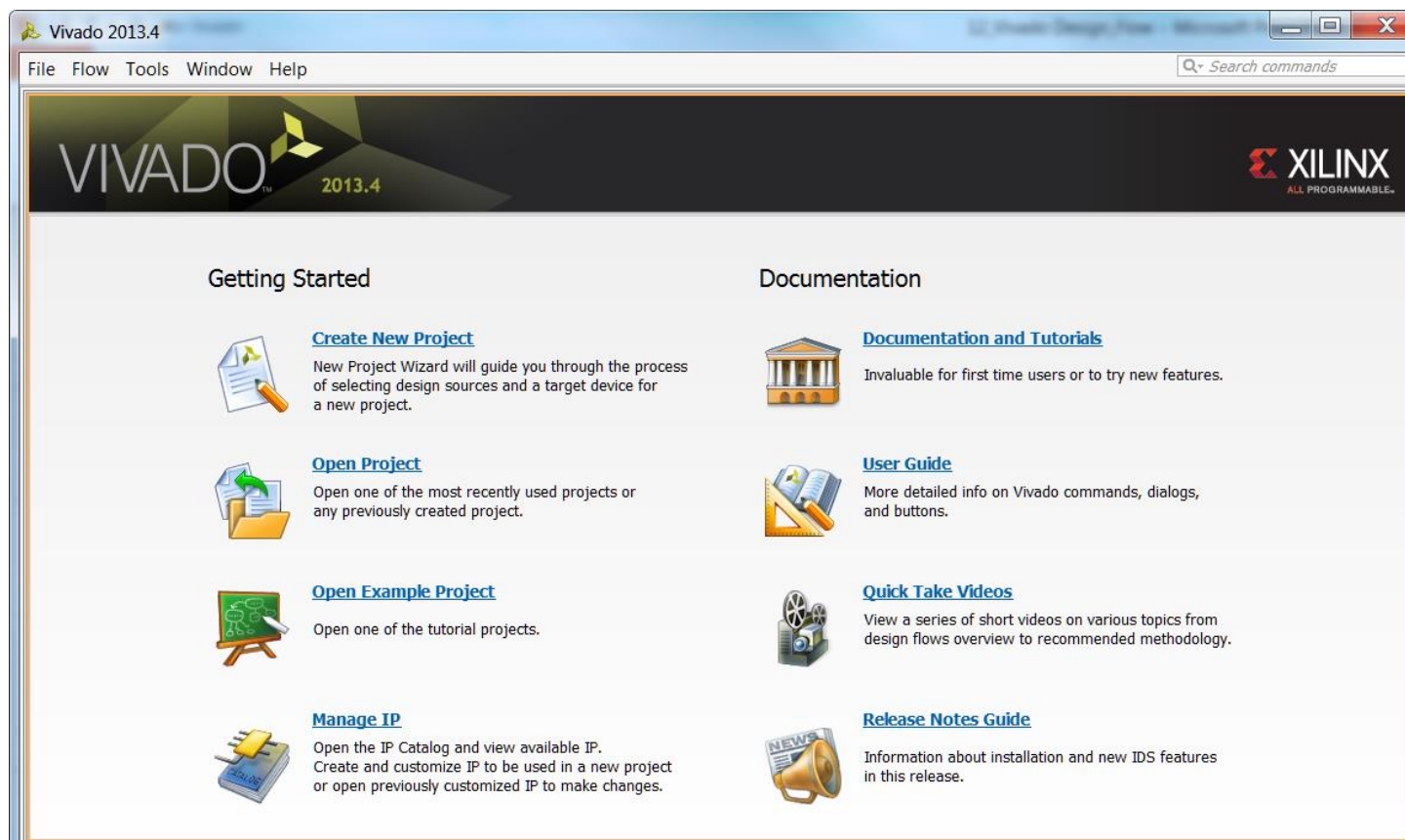
➤ 打开例子项目

– 用于教程

➤ 文档链接

– 调用PDF阅读软件来看

- 版本说明
- 用户指南
- 方法指南
- 教程

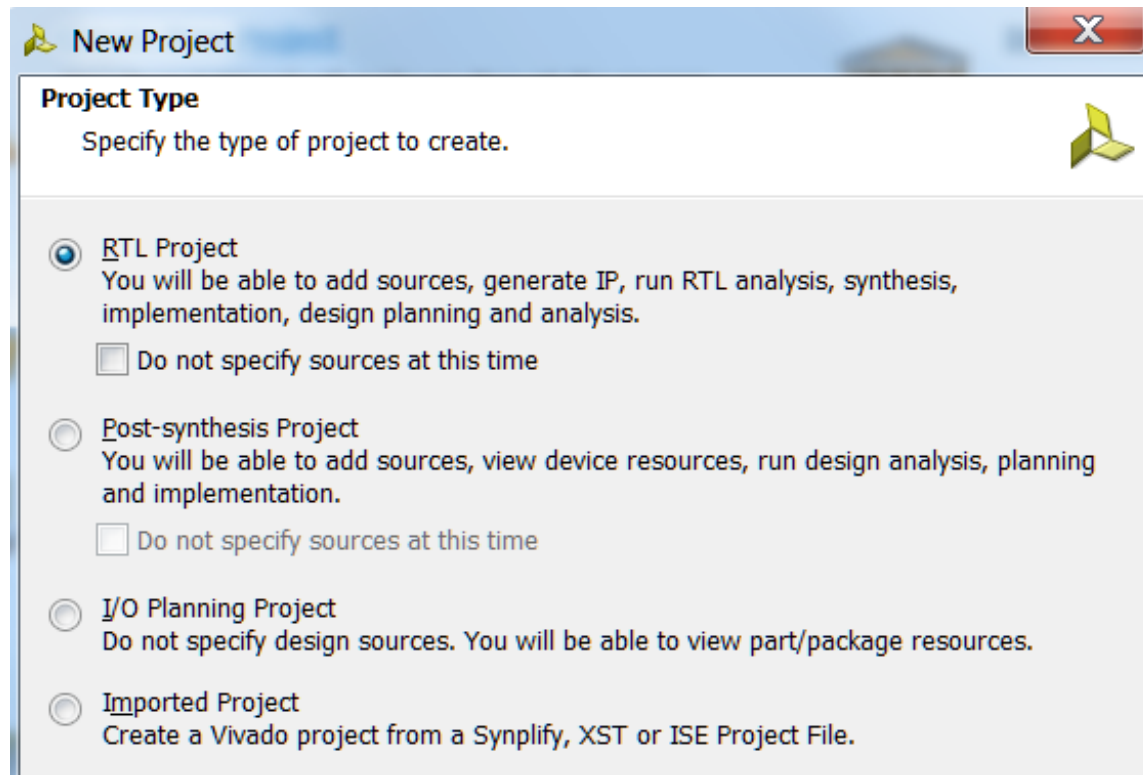


# 创建新项目的向导

## ➤ 可以创建四种不同类型的项目

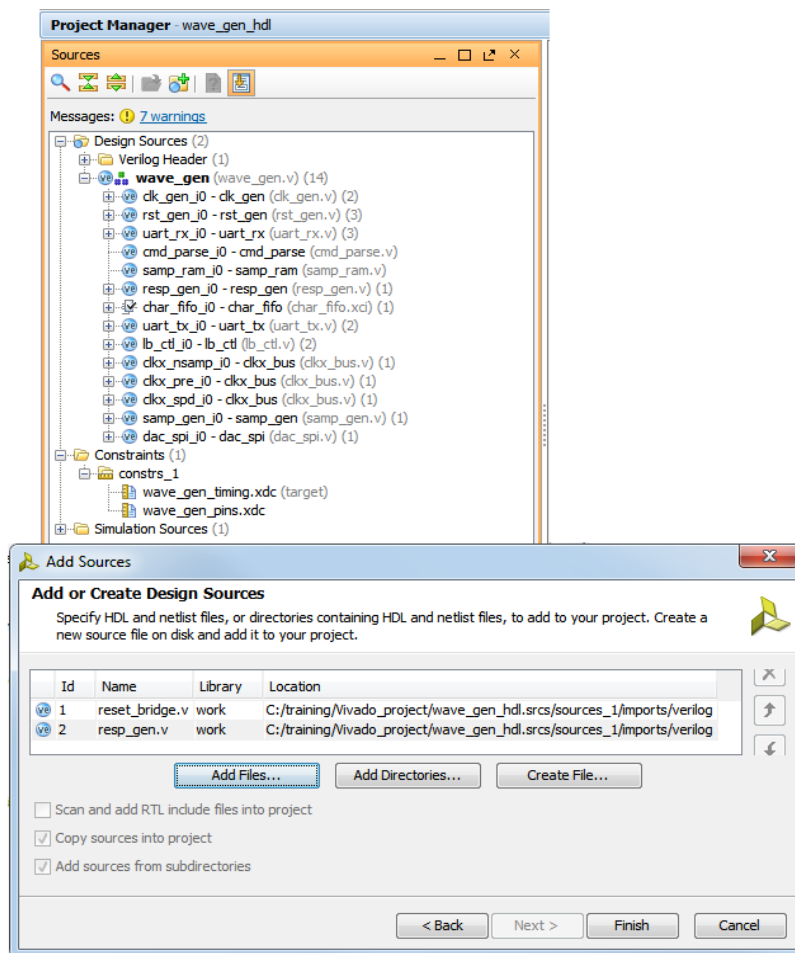
- RTL
  - 从前到后
- 综合后Post-synthesis
  - EDIF或 NGC
- I/O 规划
  - 用于早期引脚测试
  - 没有设计资源
- 输入项目
  - 从Synlify、XST或ISE项目文件中输入已有的项目

## ➤ 接下来，加入资源文件、约束文件，然后选择FPGA



# 创建有HDL资源和/或综合设计的项目

- 定义项目的名称和位置
- 在RTL项目创建中选择源文件
  - 所有能识别的源文件，Verilog、VHDL，在目录及其子目录中的，都可以加入
- 在**Post-synthesized project creation**那里选择合成后的网络表
  - 加入目录和子目录中所有的综合文件
- 选择约束文件
  - 可以加一个或多个带有IP内容的约束文件
- 选择系列和目标器件，或预定义的板
- 索引原本存在的文件，或引入拷贝这些文件到项目中来



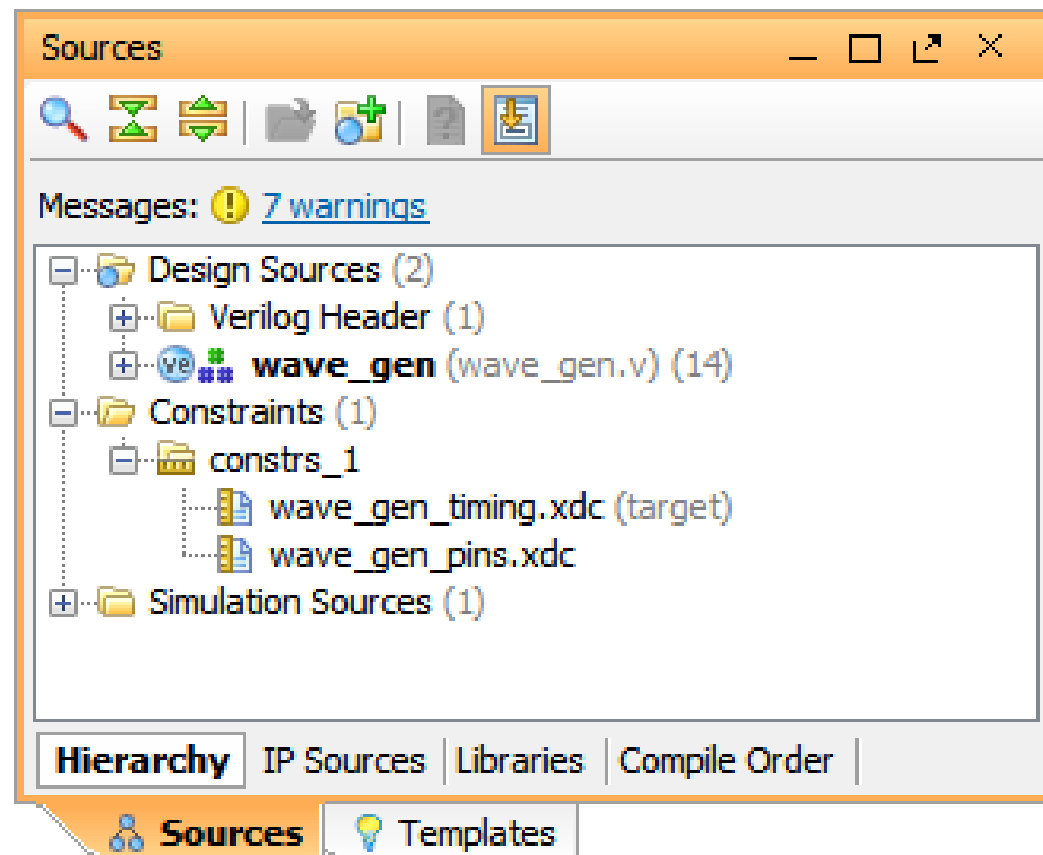
# 约束文件管理

## ➤ 约束集时XDC文件的集合

- 一个项目可以包含多个约束集，但是所有的约束集必须是相同类型的
- 要施加一个约束集，它必须是“活跃”的
- 右键点击然后选择“Make Active”就能使那个约束是活跃的

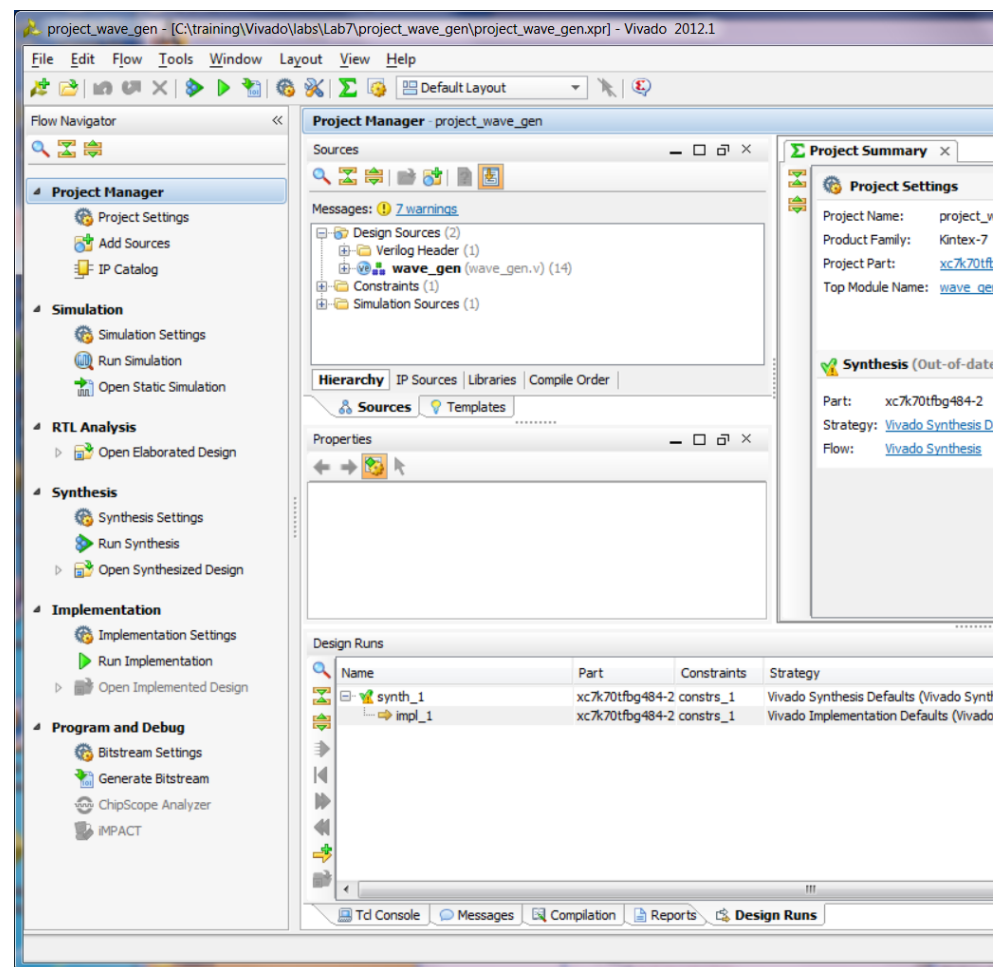
## ➤ 目标XDC

- 新建的约束会写到约束集的XDC文件里去
- 修改已有的约束会写回到原本的约束文件中去



# 项目管理

- 用来在**Project Summary**中管理源，定制IP，以及查看项目细节
- 流程导航
- 源视图
  - 源的层级显示
  - IP源和库的视图
  - HDL和网络表包含了到库和位置的索引
  - 可以访问约束文件
- 项目总结
  - 可以访问器件工具（源）、时序总结和策略信息
- **Tcl**终端、消息、编译、报告和设计运行

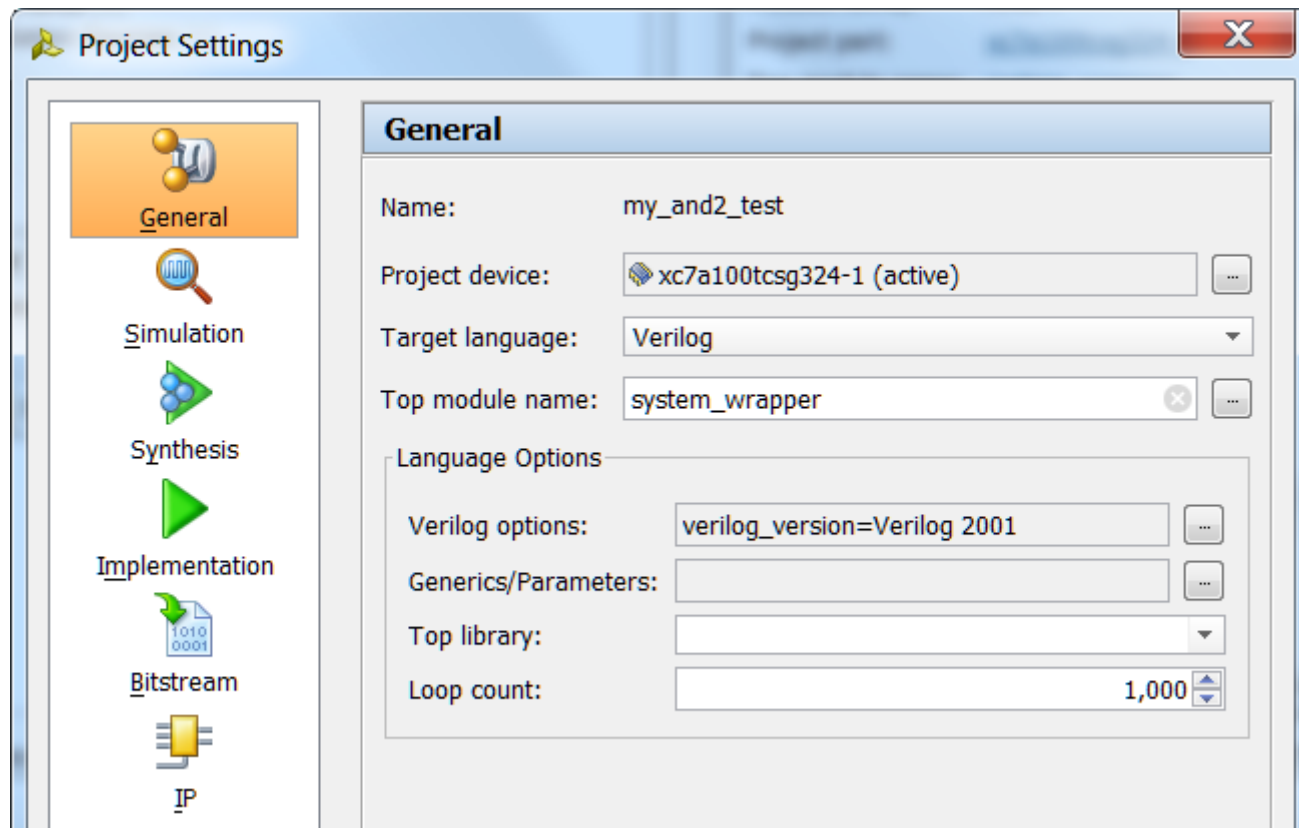


# 项目设置

## ➤ 基本设置General settings

- 选择器件
- 目标HDL 语言
- 顶层模块名称
- 语言选项

## ➤ 其他的设置在相应的单元中覆盖



# 流程导航——RTL项目

## ➤ 配置项目源

- 加入HDL源文件、约束文件、仿真文件、块设计

## ➤ IP集成器

- 创建、打开、产生一个块设计

## ➤ 运行仿真

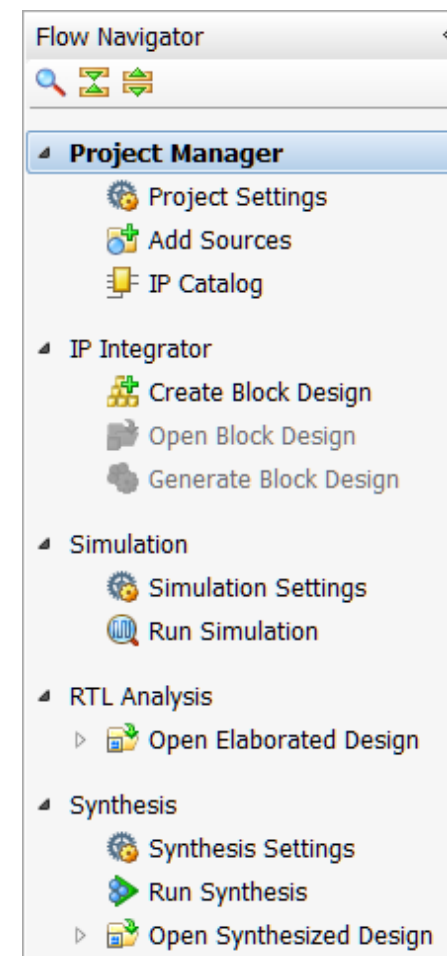
- 包括了XSIM仿真器
- 行为、合成后、实现后

## ➤ RTL分析

- Open Elaborated Design按钮：装载详细RTL设计

## ➤ 运行合成

- 时序驱动
- Open Synthesized Design按钮：装载合成网络表





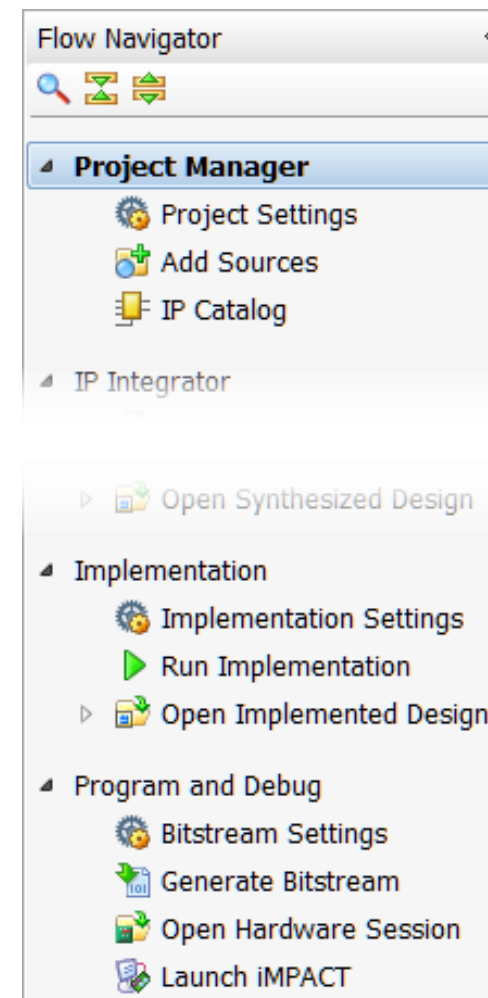
# 流程导航——RTL项目

## ➤ Run Implement按钮：运行实现工具

- link, opt, power\_opt, place, phys\_opt和布线
- Open Implemented Design 按钮: 装载实现设计

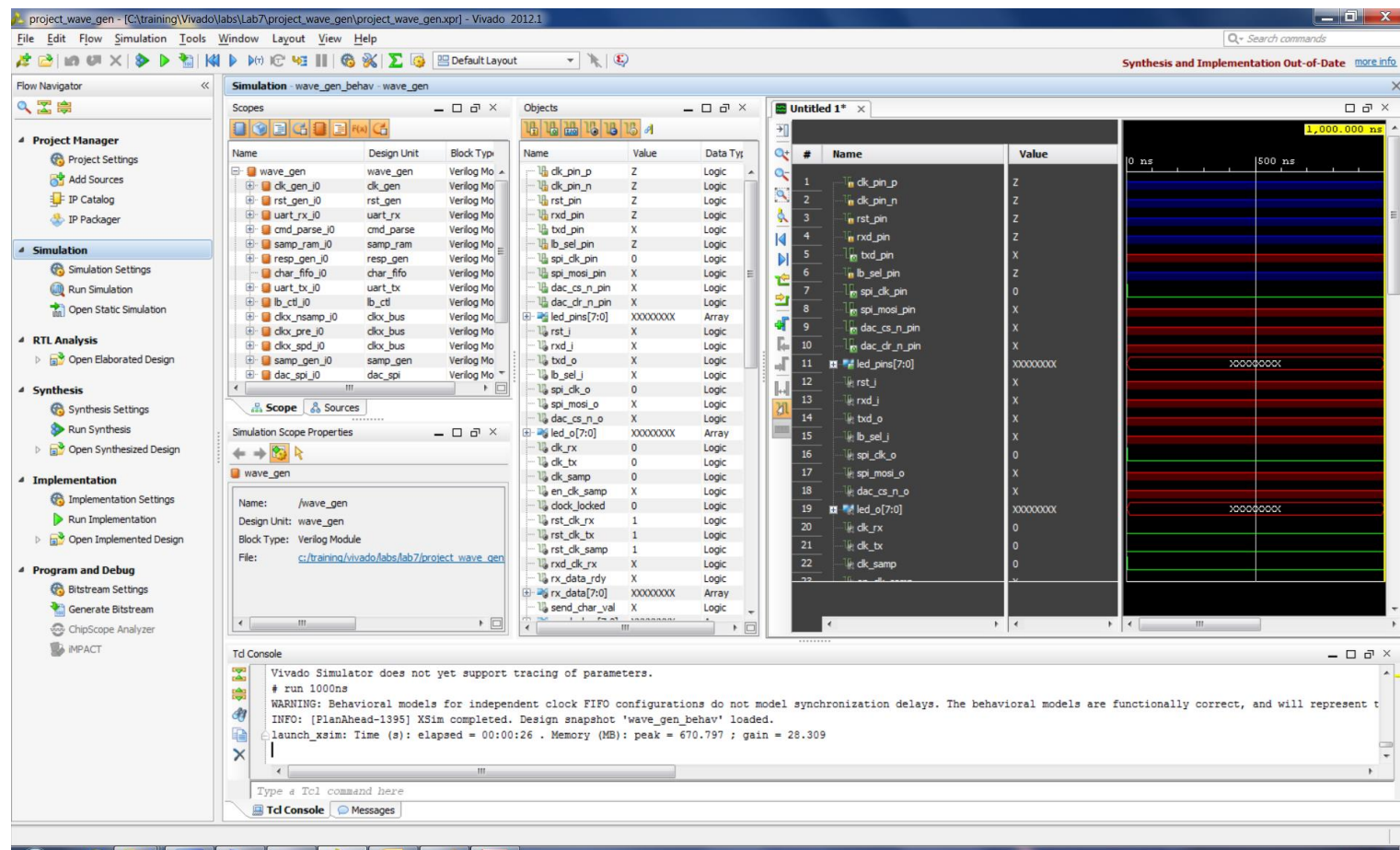
## ➤ Program and Debug: 启动编程和调试工具

- 开始硬件会话来对FPGA编程
- 如果安装了iMPACT，也可以用它来给FPGA编程



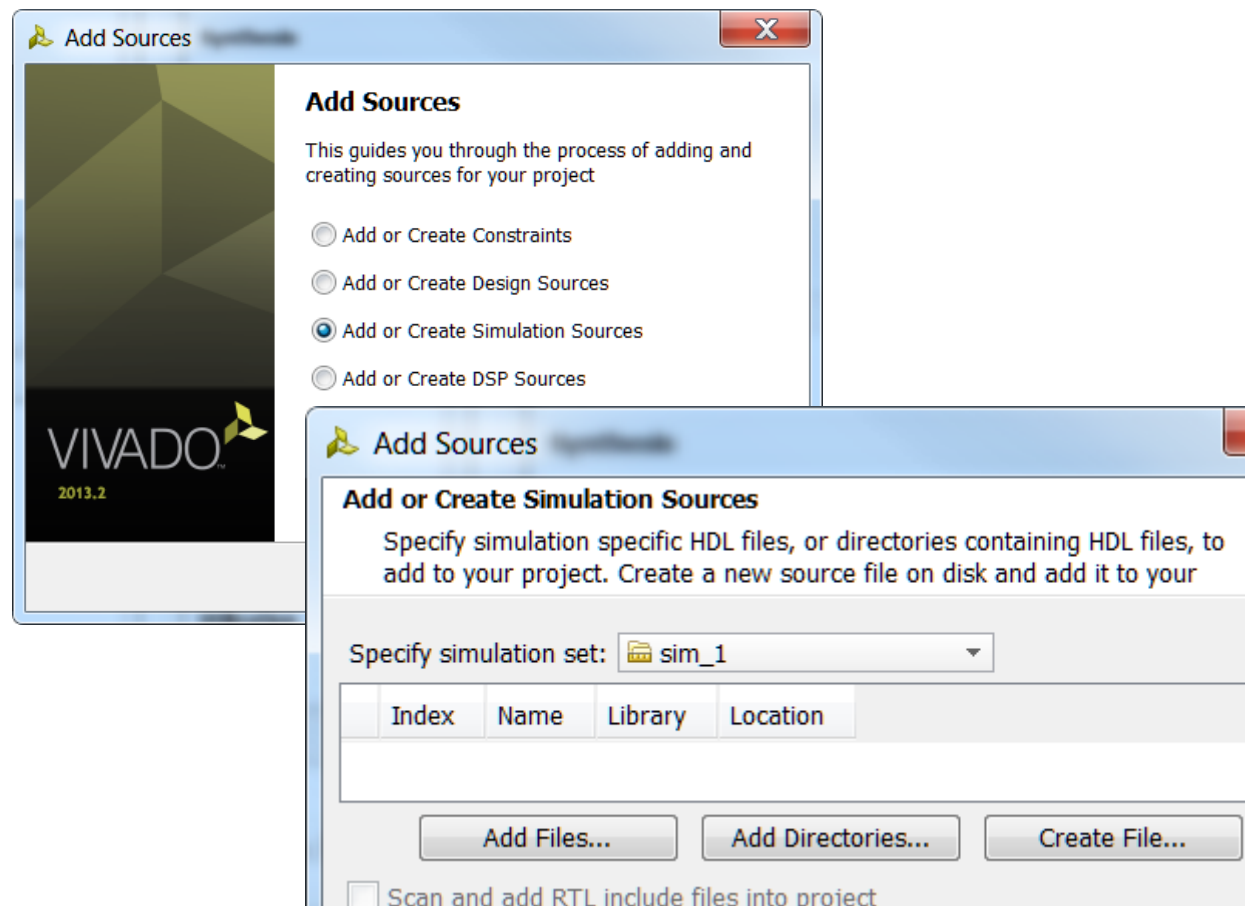
# 仿真

- Vivado仿真器 XSIM支持RTL、网络表和时序仿真
- 图形化的波形显示
- 工具条按钮可以添加标记、测量延迟和缩放
- 总线可以展开来观察每个信号
- 可以插入分离器来可视化地隔离相关的信号组
- 默认显示顶层信号



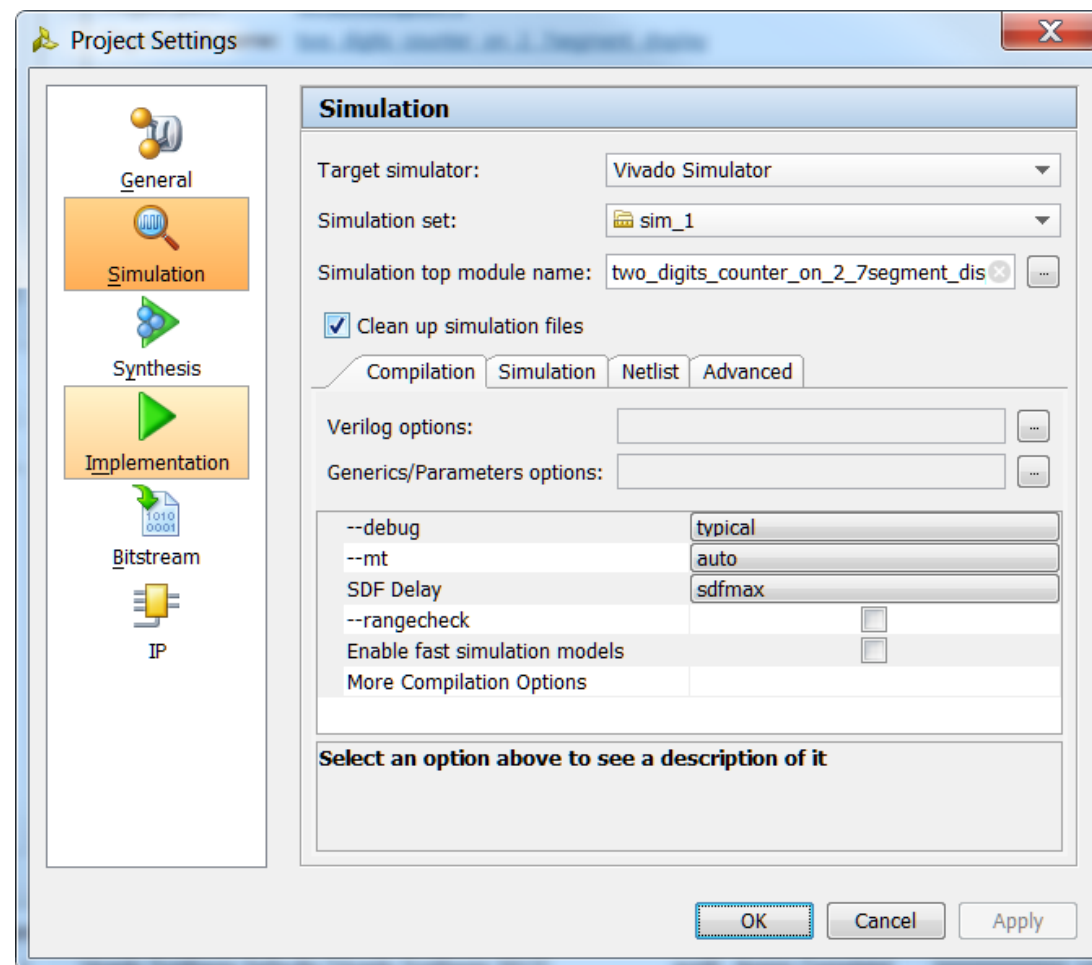
# 加入Testbench文件

- 在流程导航中，点击 **Add Sources**
- 选择**Add or Create Simulation Sources**，然后点击 **Next**
- 如果**testbench**（测试集）文件已经存在了，点击**Add Files...**, **Add directories...**按钮，否则点击 **Create File...** 按钮来创建一个新的测试集文件
- 选择文件类型- **Verilog, Verilog Header, SystemVerilog, or VHDL**
- 浏览已有的测试集或输入文件名来创建



# 仿真设置

- 从流程导航，点击 **Simulation Settings**
- 可以选择编译和仿真的属性
- 在编译页的**More Compilation Options**输入框和仿真页的**More Simulation Options**输入框可以输入额外的选项
- 更多的信息请参考 *Vivado Design Suite Simulation Guide (UG900)*



# 目录

- **Vivado IDE 功能和收益**
- **Vivado 设计套件介绍**
- **Vivado 设计流程**
- *介绍*

# 总结

## ➤ Vivado IDE 的特性与好处包括

- 性能可预计
- 能让设计者加速实现时限目标的设计、分析功能
- 使得创建脚本更容易和更强大的Tcl功能（命令）

## ➤ Vivado工具在整个FPGA设计过程中使用共同的数据模型

- 这使用户在运行时间和存储资源上受益

## ➤ Vivado工具支持无项目批处理时的脚本和基于项目的设计流程

- Vivado工具支持所有命令中使用Tcl

## ➤ Vivado工具在设计过程中使用共同的约束语言（XDC）

- 这样就使得合成优化极大地优于ISE软件

## ➤ 大多数设计过程只需要按下按钮