

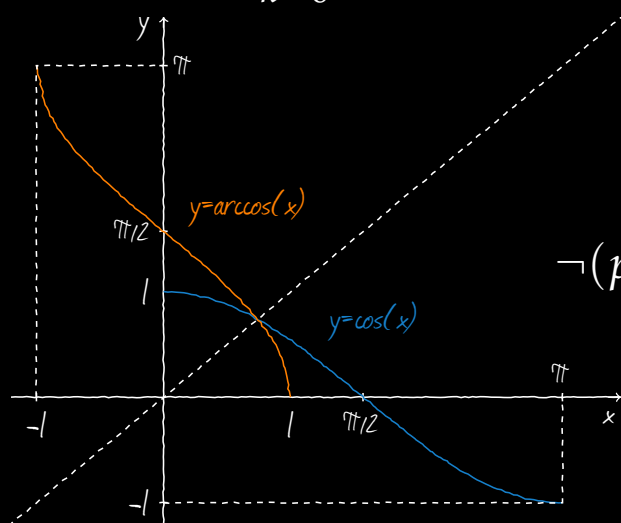
ZJUNIX

# 实验操作流程

浙江大学

2017.08.20

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k}$$

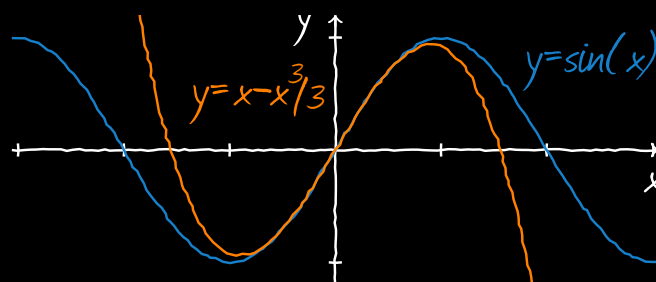


$$\zeta_k = |a|^{1/n} e^{i(\arg(a) + 2k\pi)/n}$$

$$e^{i\pi} + 1 = 0$$

$$\neg(p \vee q) \equiv (\neg p) \wedge (\neg q)$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$



Chapter 1	引言	Page 2
-----------	----	--------

1.1	硬件简介	2
1.2	操作系统（软件）简介	2

Chapter 2	硬件综合实验	Page 3
-----------	--------	--------

2.1	工具环境	3
2.2	Linux 实验流程（硬件）	3

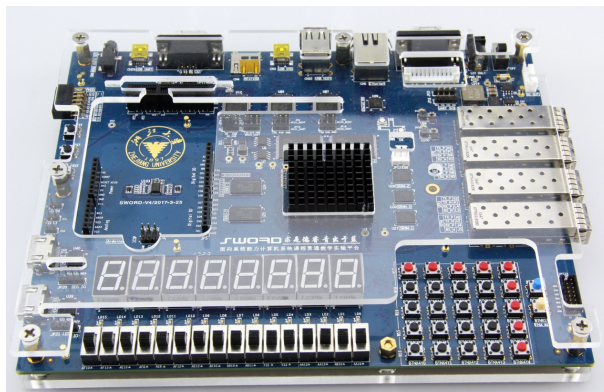
开启 vivado – 新建工程【第一步】 – 赋值硬件核模块及相关文件【第二步】 – 添加 IP 网表层及代码源文件【第三步】  
– 添加 DDR3 IP 核文件【第四步】 – 添加引脚约束文件【第五步】 – 选择开发板（xc7k325tffg676-2）【第六步】 –  
修改编译设置【第七步】 – 综合布线生成流文件【第八步】 – 下载开发板

2.3	Windows 实验流程（硬件）	15
-----	------------------	----

开启 vivado – 新建工程【第一步】 – 赋值硬件核模块及相关文件【第二步】 – 添加 IP 网表层及代码源文件【第三步】  
– 添加 DDR3 IP 核文件【第四步】 – 添加引脚约束文件【第五步】 – 选择开发板（xc7k325tffg676-2）【第六步】 –  
修改编译设置【第七步】 – 综合布线生成流文件【第八步】 – 下载开发板

## 1.1 硬件简介

ZJUNIX 操作系统基于浙江大学开发的 SWORD 通贯式教学板卡开发，在 SWORD 上开发了一套具有基本功能的 PC，包括一个 MIPS 指令集的 CPU，基本的输入输出外设等。在此基础上进行进一步的开发。培训所使用的 SWORD 版本为 SWORD V4.0，板卡为蓝色。



## 1.2 操作系统（软件）简介

ZJUNIX 是由浙江大学计算机系统兴趣小组同学自行开发的操作系统，可以安装到自行开发的 PC 机上，实现进程调度，内存管理，文件系统管理等功能。整个操作系统由 C 语言编写，嵌入少量 MIPS 汇编代码，配套有完整的 Makefile。

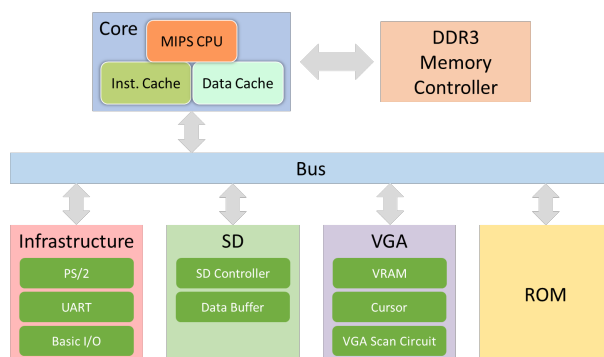


图 1.1: SOC 架构

2.1 工具环境

环境	版本
硬件综合工具	Vivado 2016.2

表 2.1: 工具环境

2.2 Linux 实验流程（硬件）

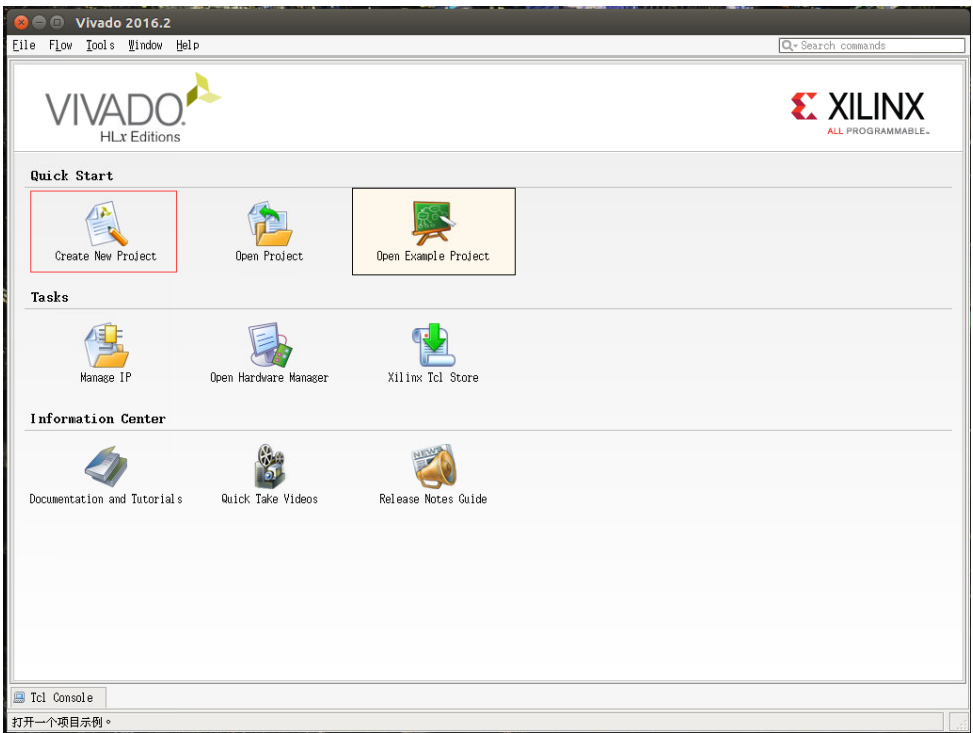
2.2.1 开启 vivado

本次实验的操作系统是 Ubuntu 系统，在开启电脑之后，登录界面的用户名为 zju，密码为 zjusig

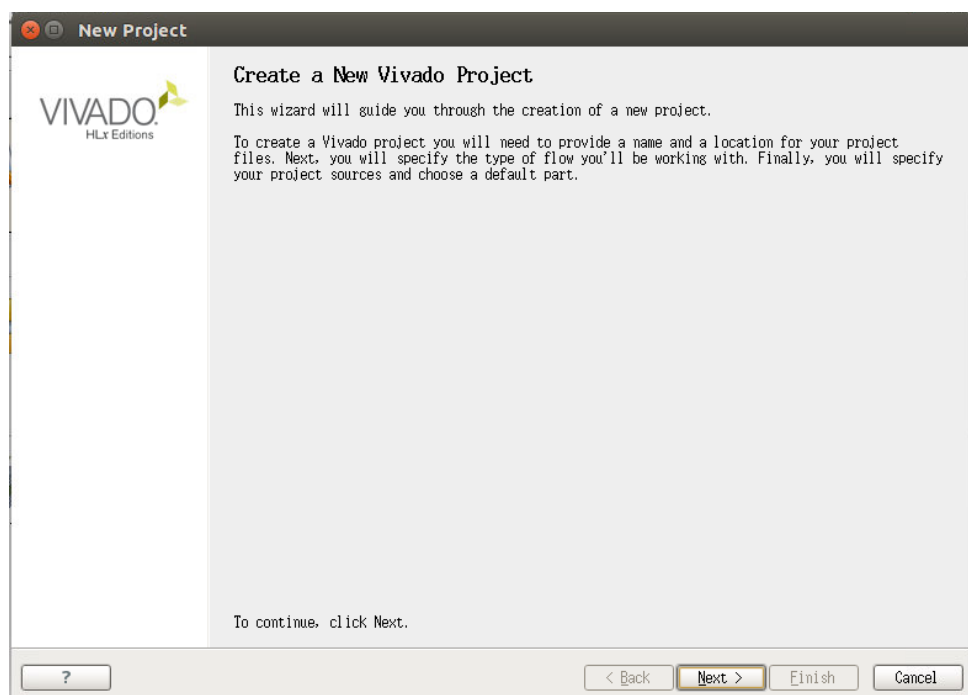
开启 Ubuntu 系统之后，双击桌面图标 vivado，即可开启 vivado 2016.2

2.2.2 新建工程【第一步】

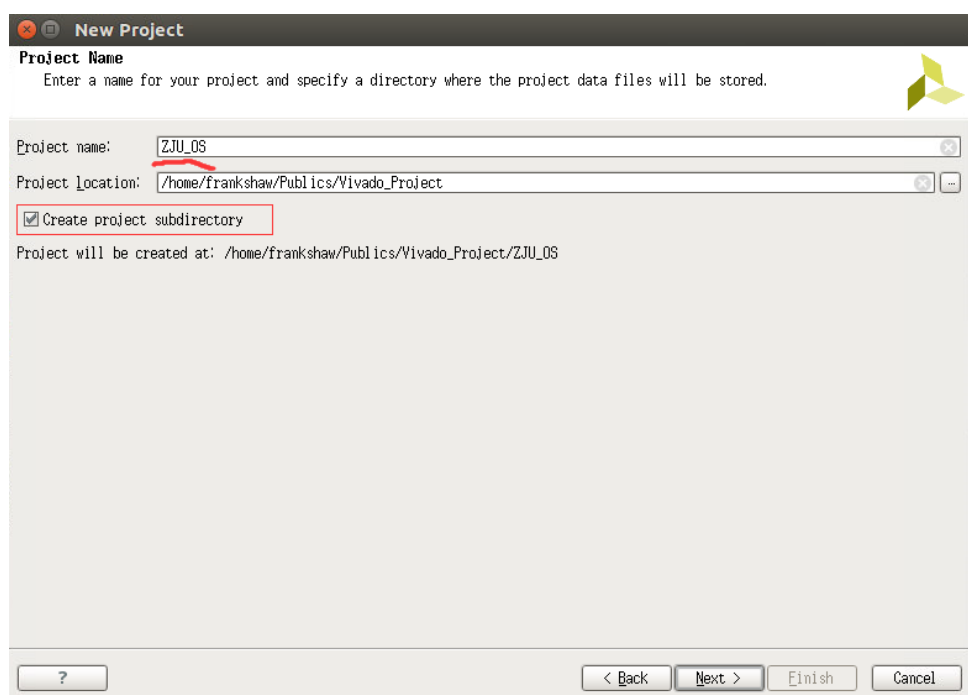
在开启 vivado 界面之后，点击 Create New Project 按钮，创建一个新的工程。



第一页向导，点击 Next



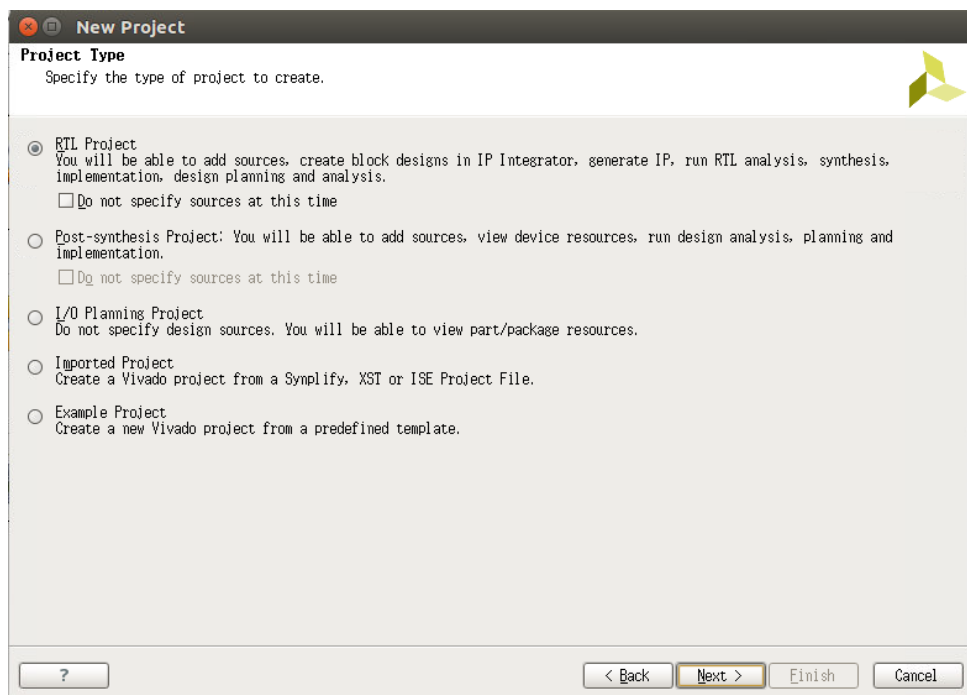
下一步是给工程命名，记住命名规则（不能以数字开头，中间不能有空格，不能有中文字符）。这里以 ZJU\_OS 举例，下面的 Create Project Subdirectory 表示创建子目录。目录位置选择好，继续点击 Next



这个时候，工程已经在指定目录下面创建了一个 ZJU\_OS 的文件夹（目前为空）

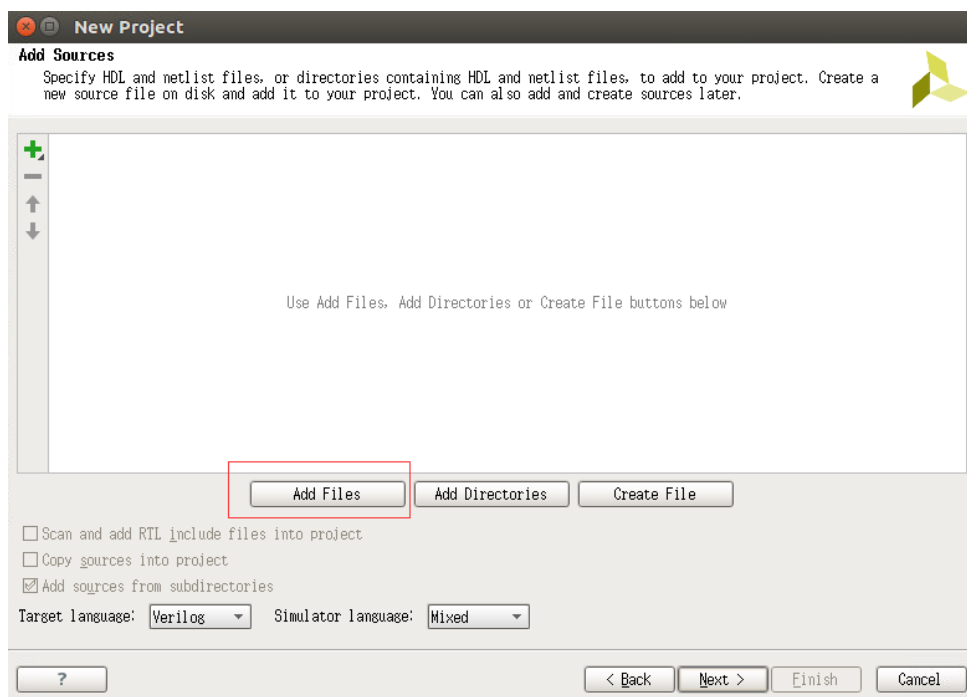
## 2.2.3 赋值硬件核模块及相关文件【第二步】

打开文件管理器，把提供的 code 文件夹里面的内容，全部赋值到 ZJU\_OS 文件夹内。  
回到 Vivado，工程类型默认不变，点击 Next

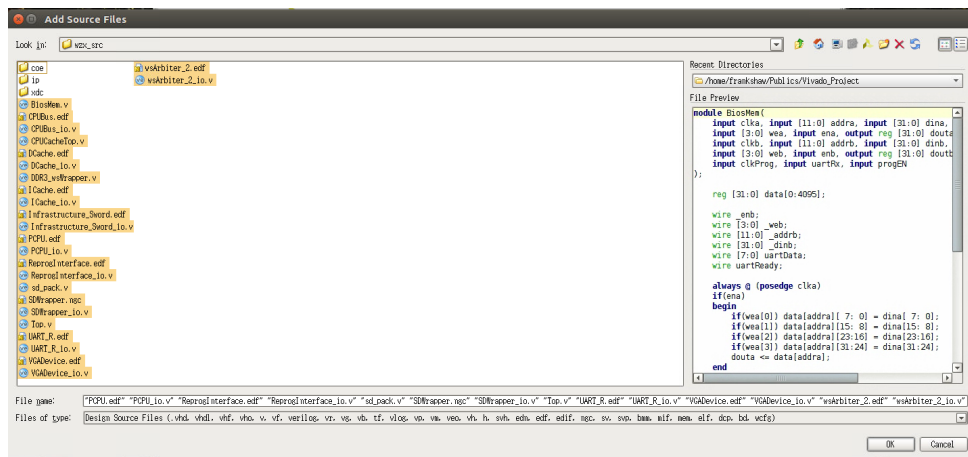


下一步添加源文件，点击 Add Files

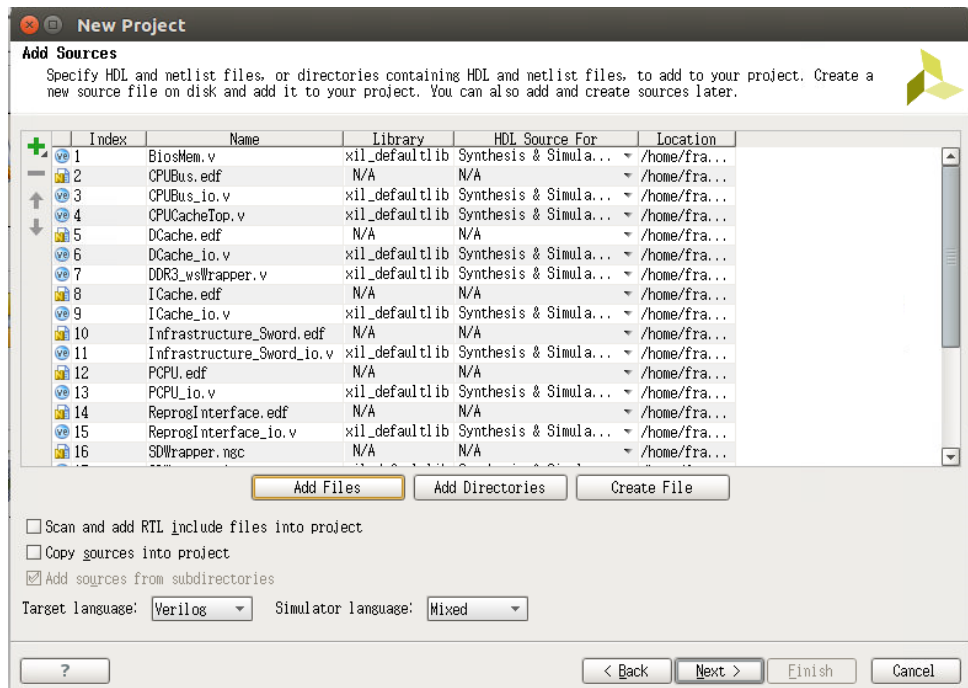
## 2.2.4 添加 IP 网表层及代码源文件【第三步】



把 ZJU\_OS 文件夹里面的除了 ddr3 文件夹之外的，所有.v 和.edf 文件都添加进来

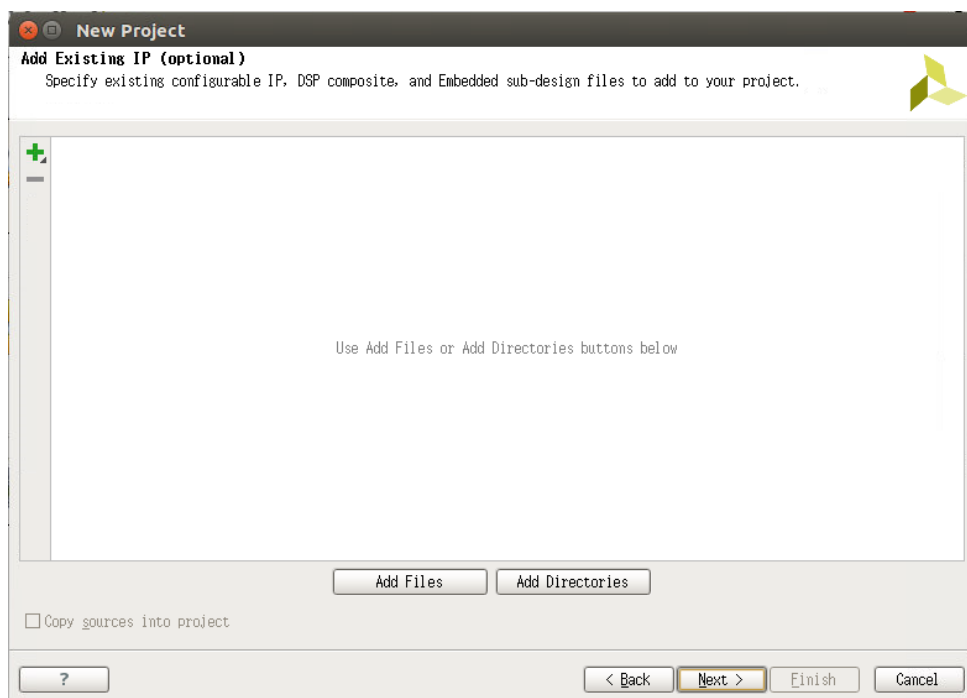


点击 OK

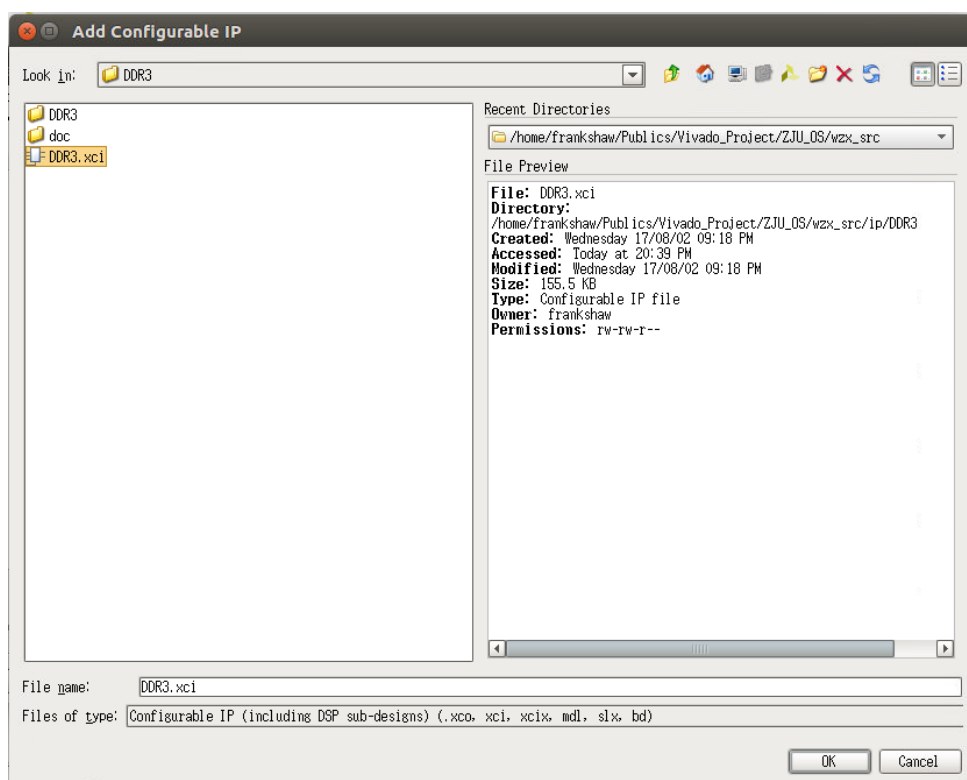


然后点击 Next，注意缺省是选择 Add sources from subdirectories

## 2.2.5 添加 DDR3 IP 核文件【第四步】

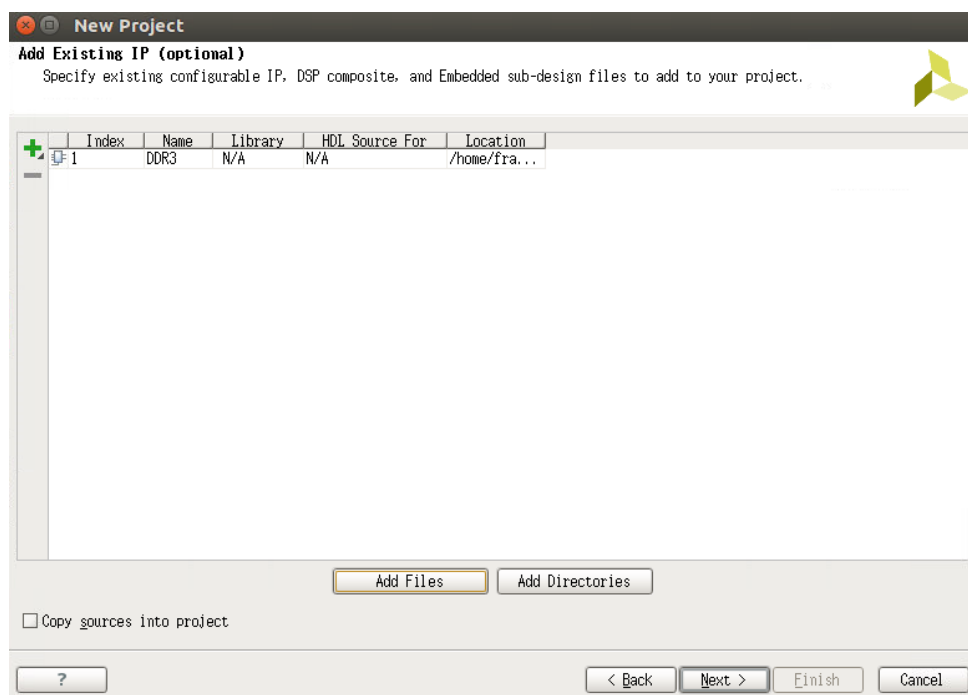


这里，我们需要把现有的储存器 DDR3 IP 核导入，点击 Add Files



选择 IP/DDR3/DDR3.xci 文件，点击 OK



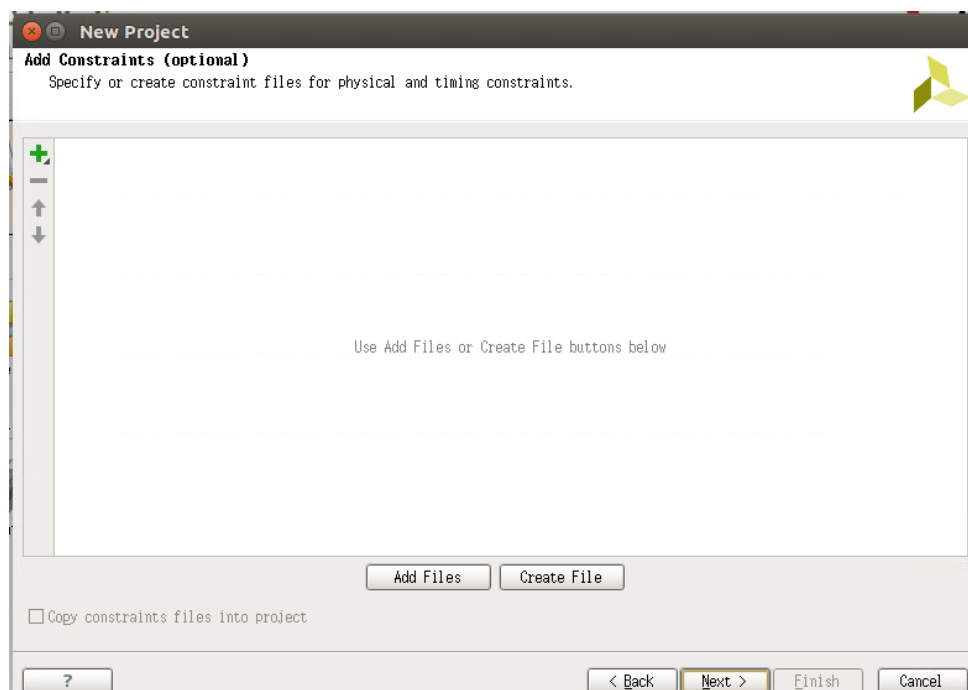


点击 Next

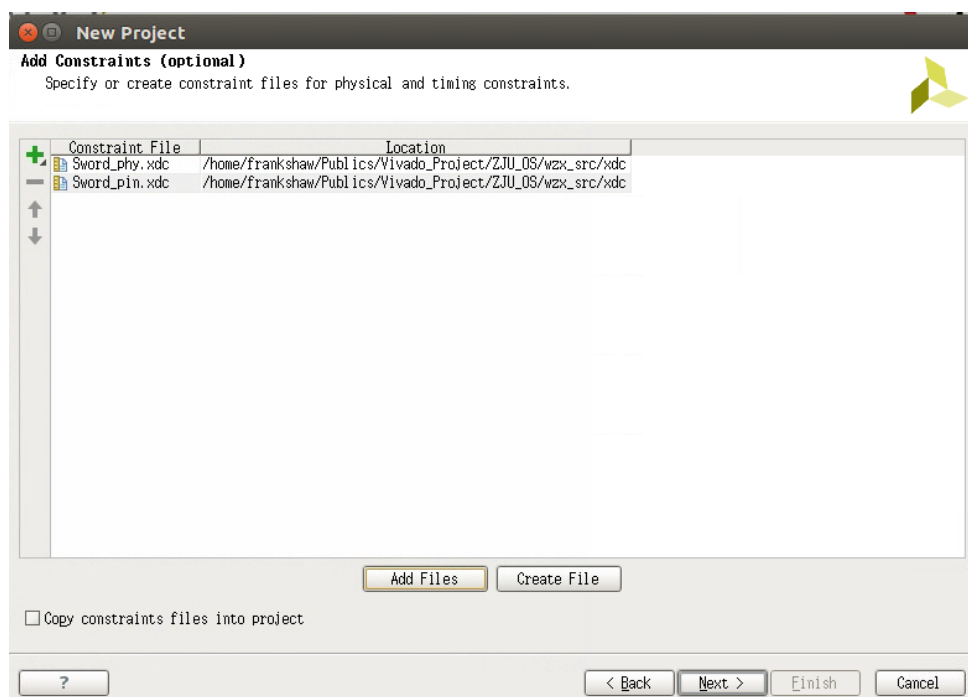
下一步，添加引脚约束文件

## 2.2.6 添加引脚约束文件【第五步】

点击 Add Files

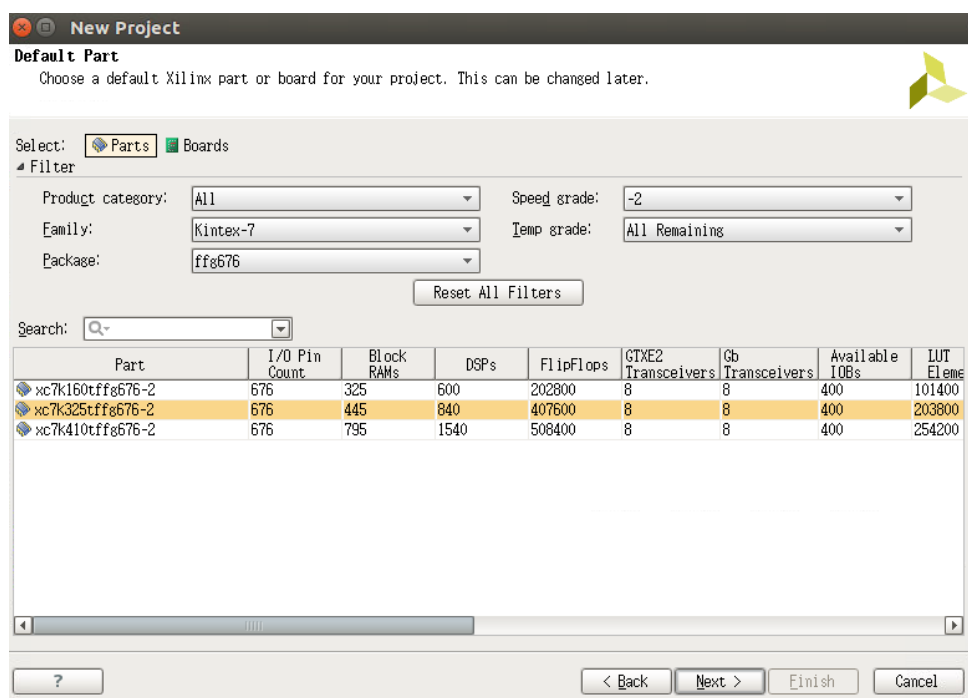


选择 xdc 文件夹下面的两个 xdc 文件，添加进来，点击 OK



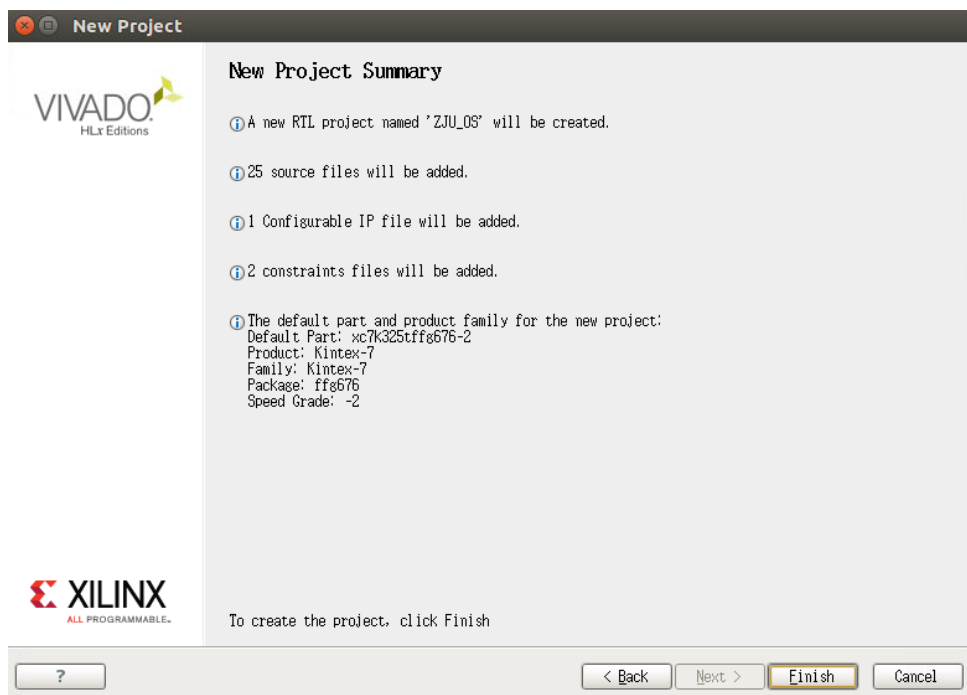
再点击 Next

## 2.2.7 选择开发板（xc7k325tffg676-2）【第六步】



点击 Next

最后确认一下工程类型

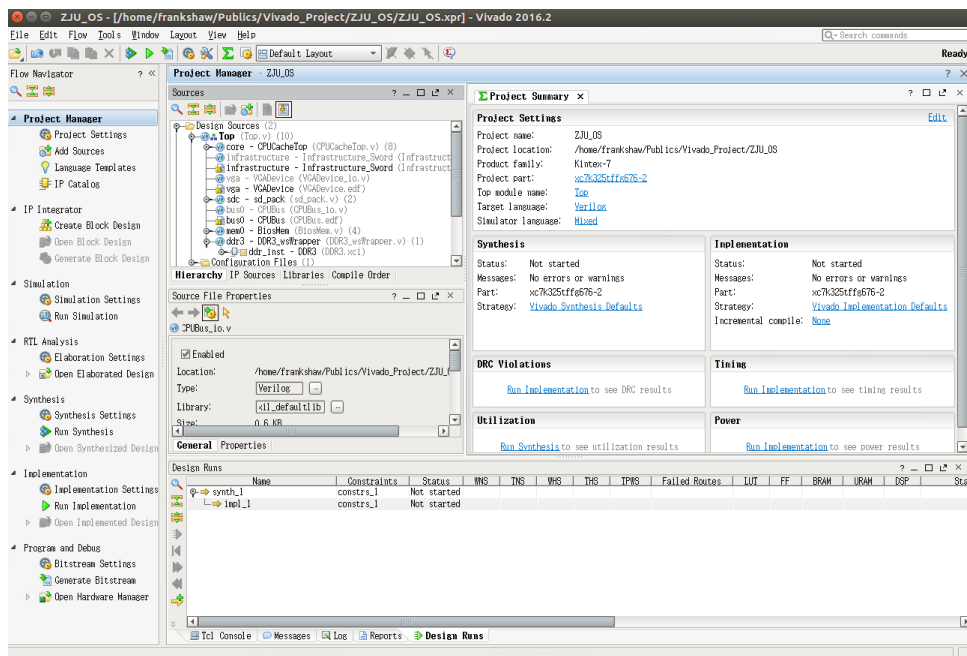


点击 Finish

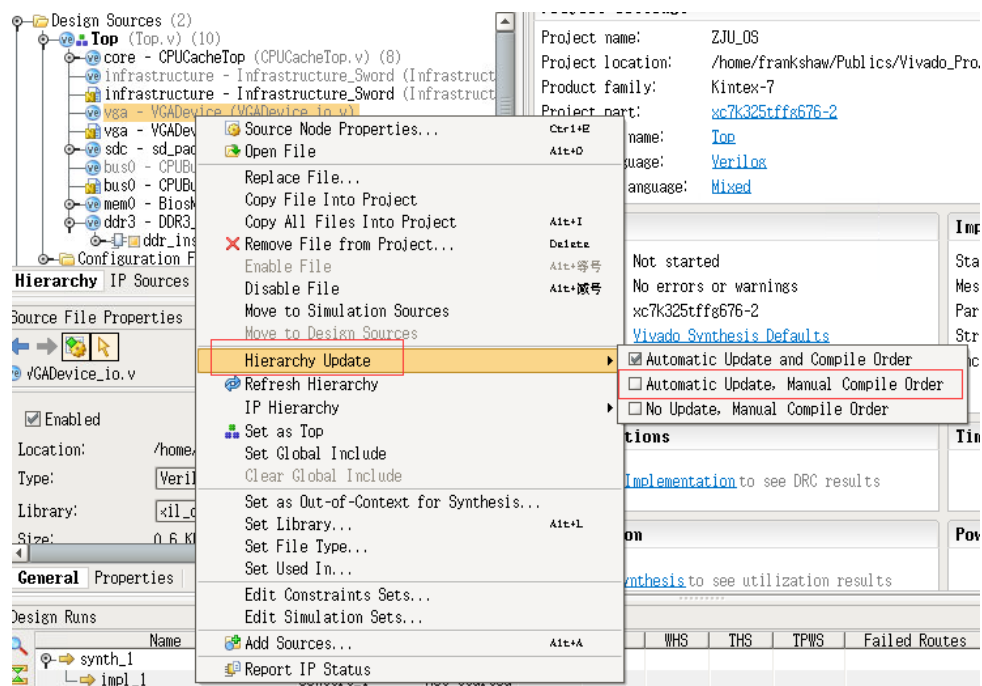
稍等片刻 Vivado 会自动创建好工程。

## 2.2.8 修改编译设置【第七步】

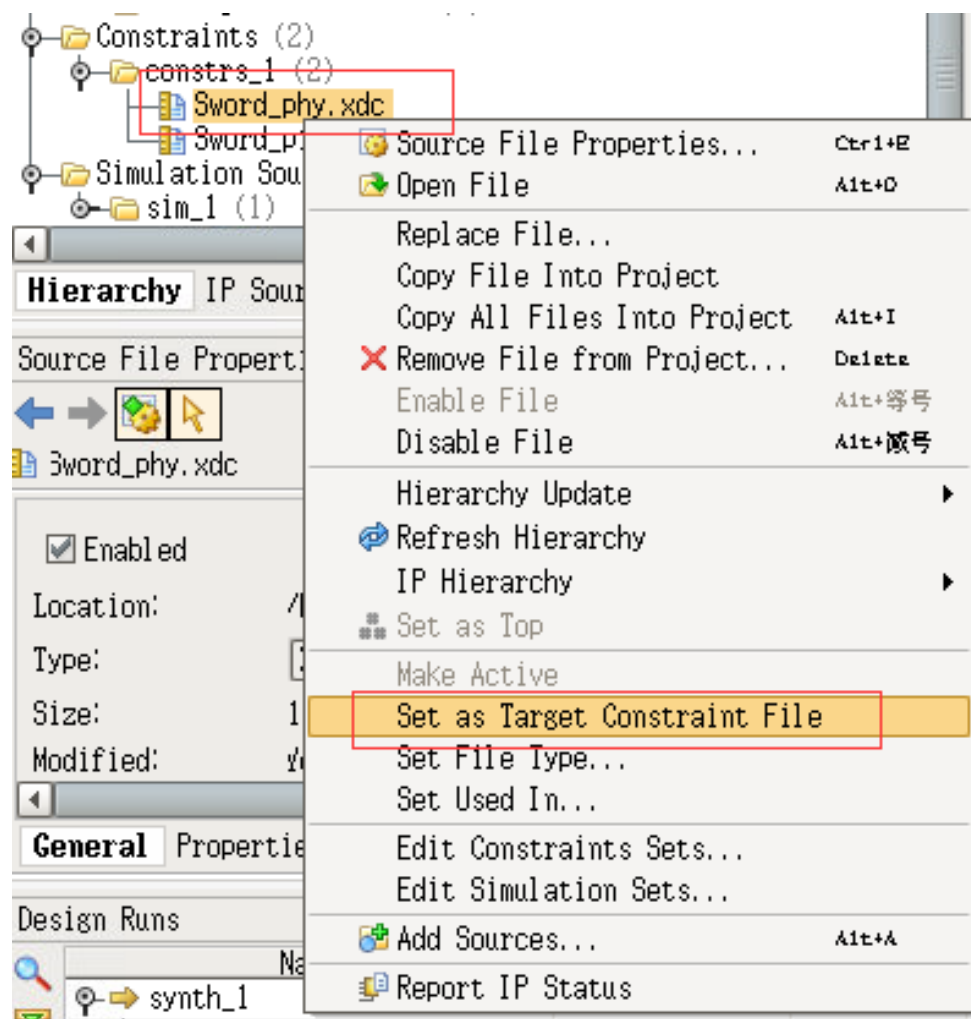
vivado 的主界面如下



在 Source 窗口的 Design Sources 内，单击右键，选择 Hierarchy Update -> Automatic Update, Manual Compile Order

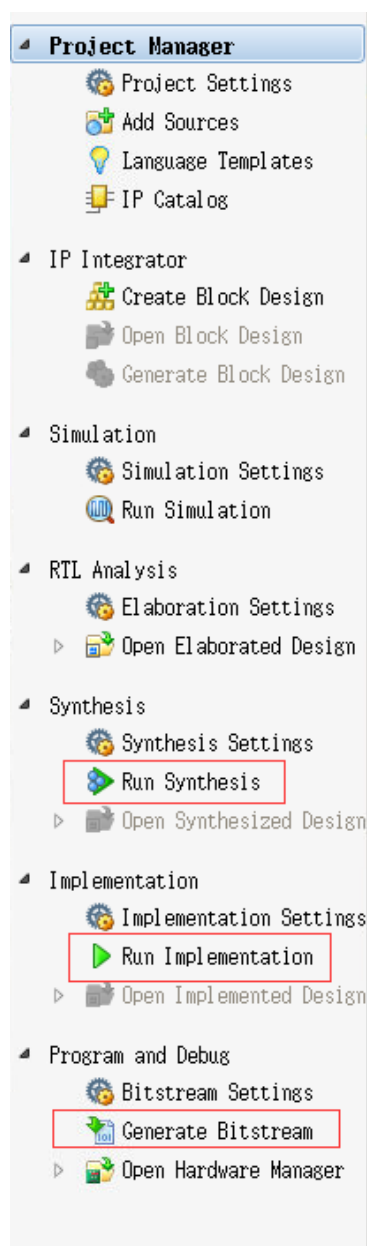


然后在 Constraints 的 Sword\_phy.xdc 文件上面单击右键，选择 Set as Target Constraint File

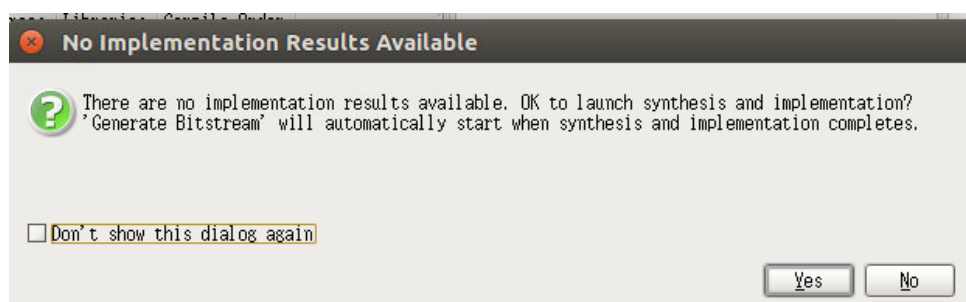


## 2.2.9 综合布线生成流文件【第八步】

接下来，你可以依次点击 Run Synthesis -> Run Implementation -> Generate Bitstream，也可以直接点击 Generate Bitstream 来生成流文件。

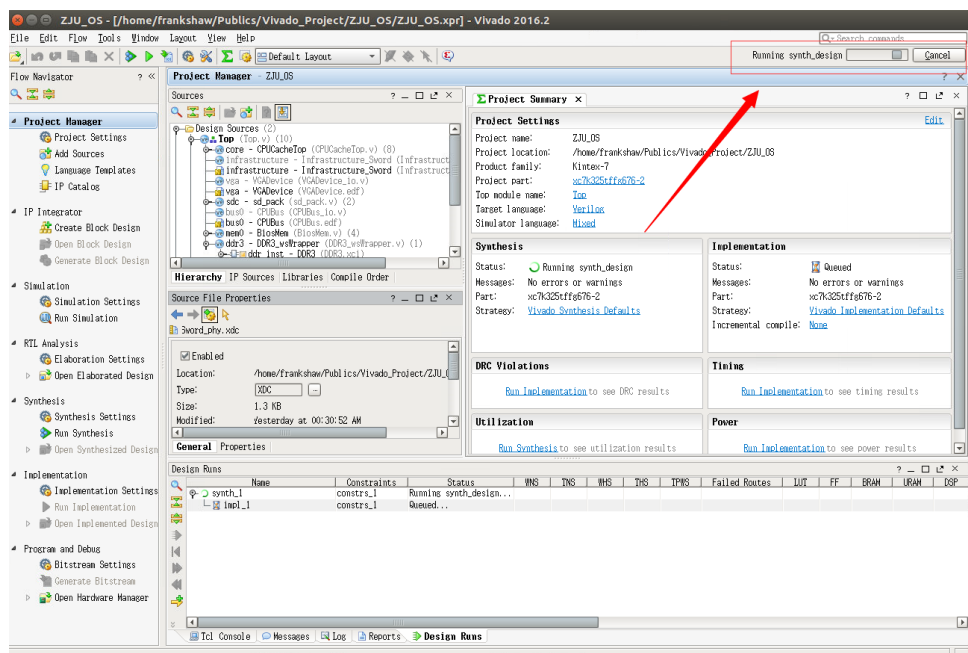


如果直接点击 Generate Bitstream 会提示你没有实现结果，需要先综合与实现，这里点击 Yes 即可



然后开始综合布线生成流文件，这个时间会比较长，大约需要 30 到 40 分钟不等（可能更长，根据机器配置而定）

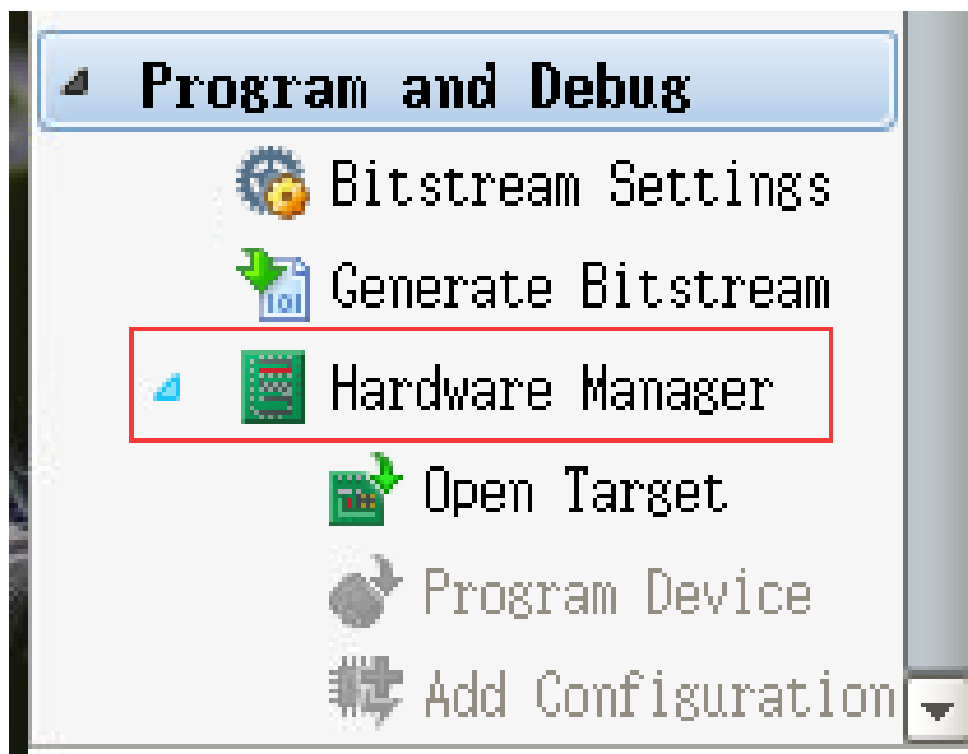
下图这个位置是当前的进度状态。



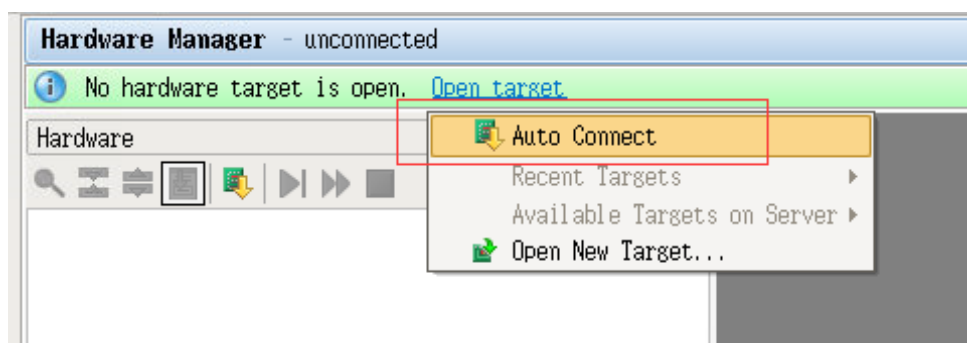
当生成完成时，就可以在 ZJU\_OS 工程目录的 ZJU\_OS.runs -> impl\_1 文件夹下面，找到 Top.bit 流文件了。

## 2.2.10 下载开发板

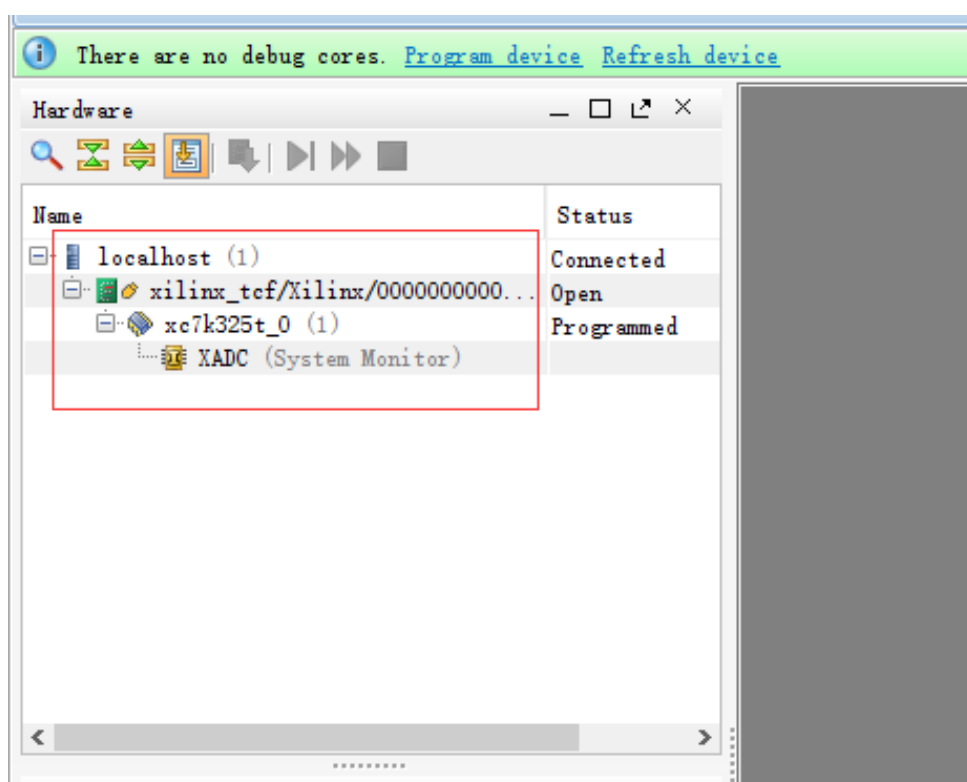
点击 Hardware Manager



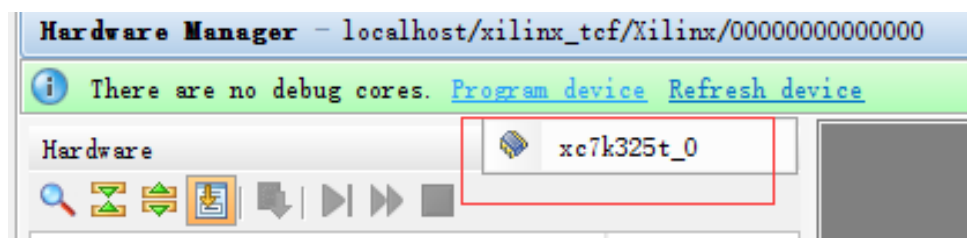
在弹出的硬件设备管理窗口里面，点击



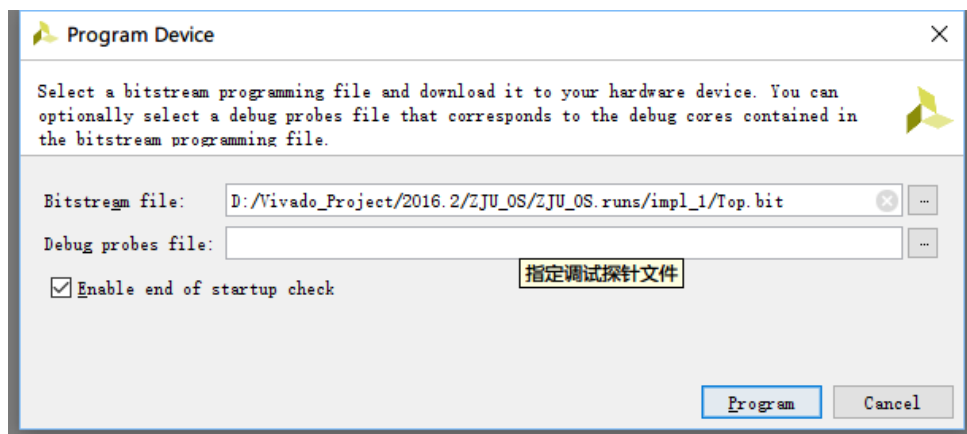
这样可以看到连接的设备（如果看不到，请检查连线 and 驱动）



点击 Program device



在弹出窗口请检查路径是否正确



单击 Program



等待机器讲流文件烧写入板子（很快，约十几秒）

## 2.3 Windows 实验流程（硬件）

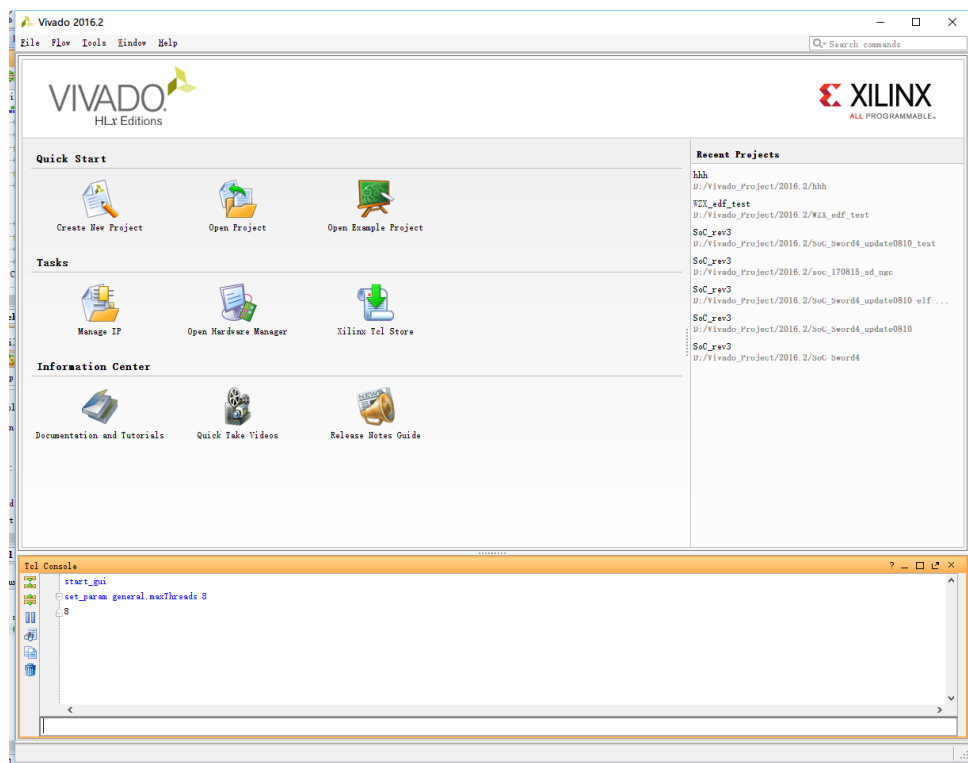
### 2.3.1 开启 vivado

本次实验的操作系统是 Windows 系统，打开 vivado 2016.2

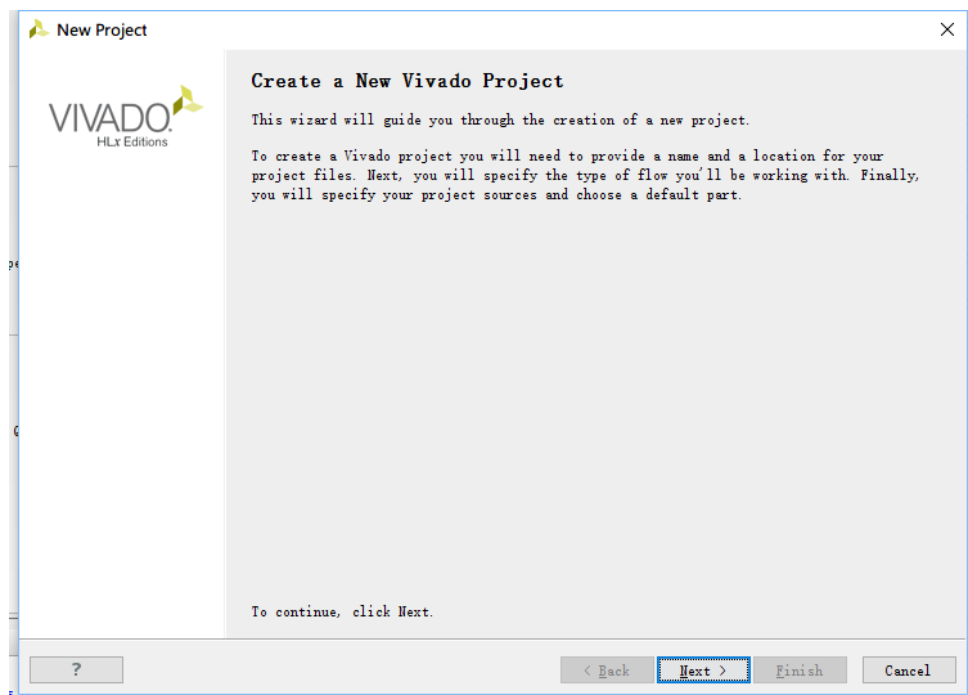
### 2.3.2 新建工程【第一步】

在开启 vivado 界面之后，点击 Create New Project 按钮，创建一个新的工程。

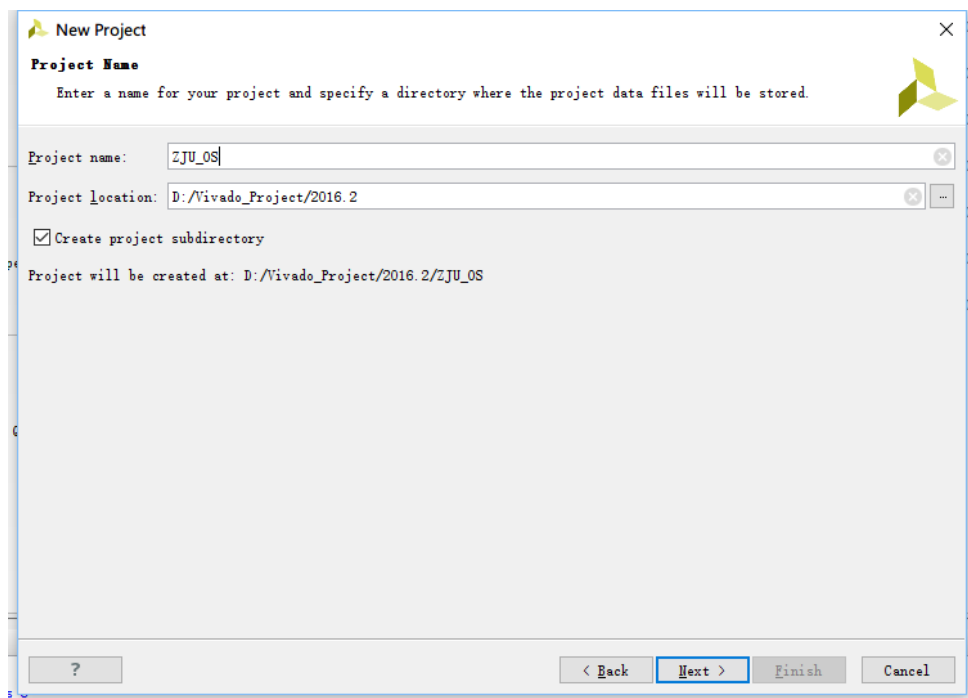




第一页向导，点击 Next



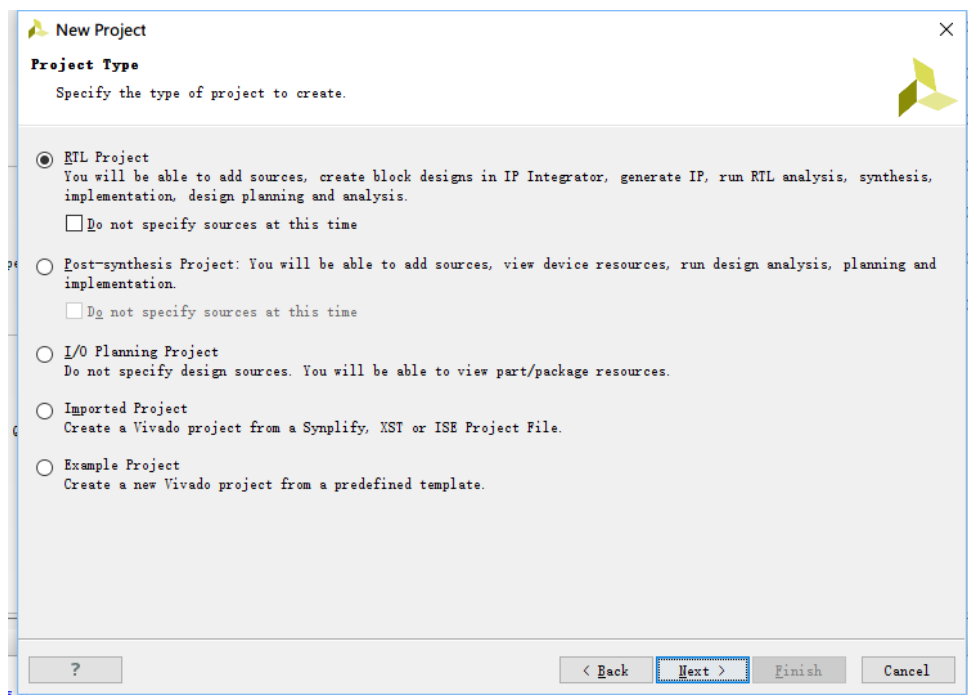
下一步是给工程命名，记住命名规则（不能以数字开头，中间不能有空格，不能有中文字符）。这里以 ZJU\_OS 举例，下面的 Create Project Subdirectory 表示创建子目录。目录位置选择好，继续点击 Next



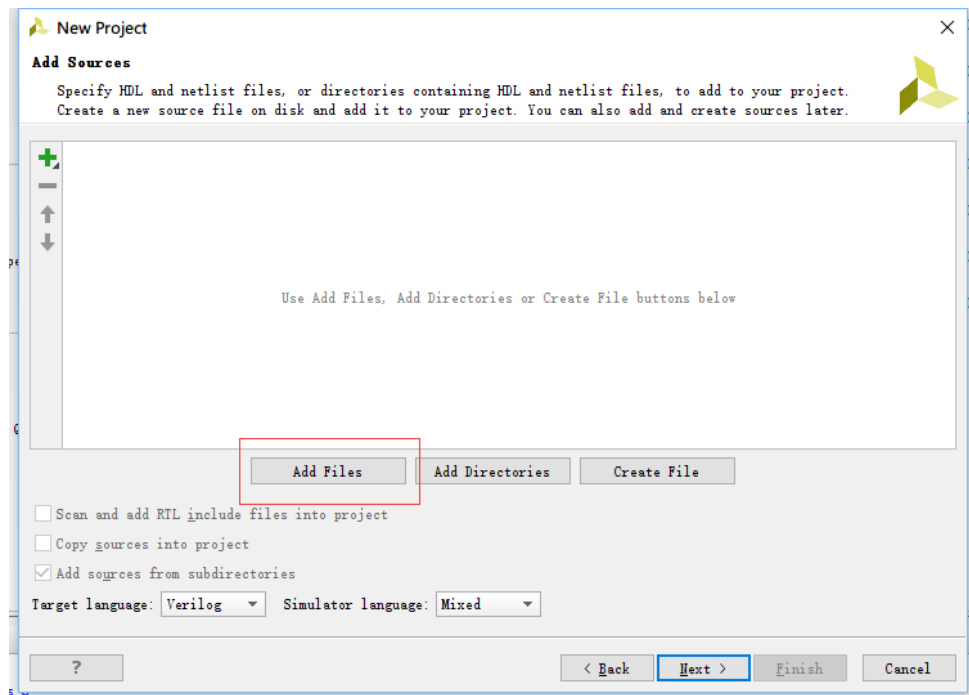
这个时候，工程已经在指定目录下面创建了一个 ZJU\_OS 的文件夹（目前为空），

### 2.3.3 赋值硬件核模块及相关文件【第二步】

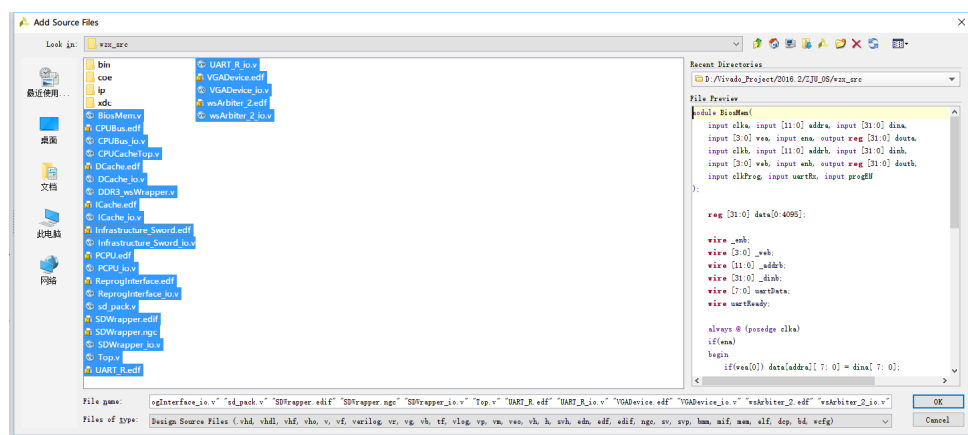
打开文件管理器，把提供的 code 文件夹里面的内容，全部赋值到 ZJU\_OS 文件夹内。  
回到 Vivado，工程类型默认不变，点击 Next



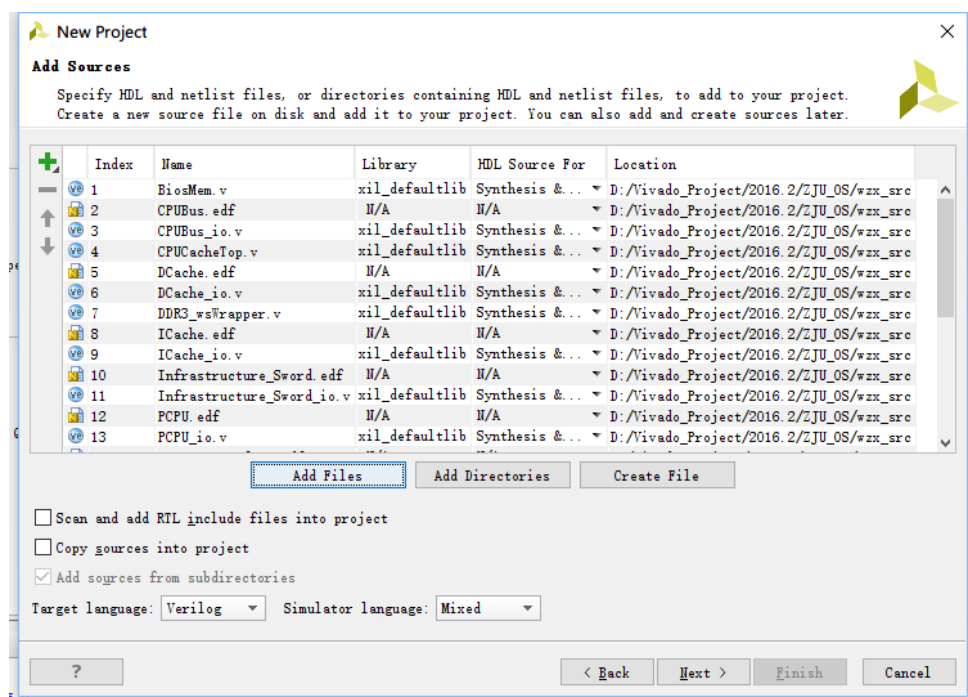
### 2.3.4 添加 IP 网表层及代码源文件【第三步】



把 ZJU\_OS 文件夹里面的除了 ddr3 文件夹之外的, 所有.v 和.edf 文件都添加进来

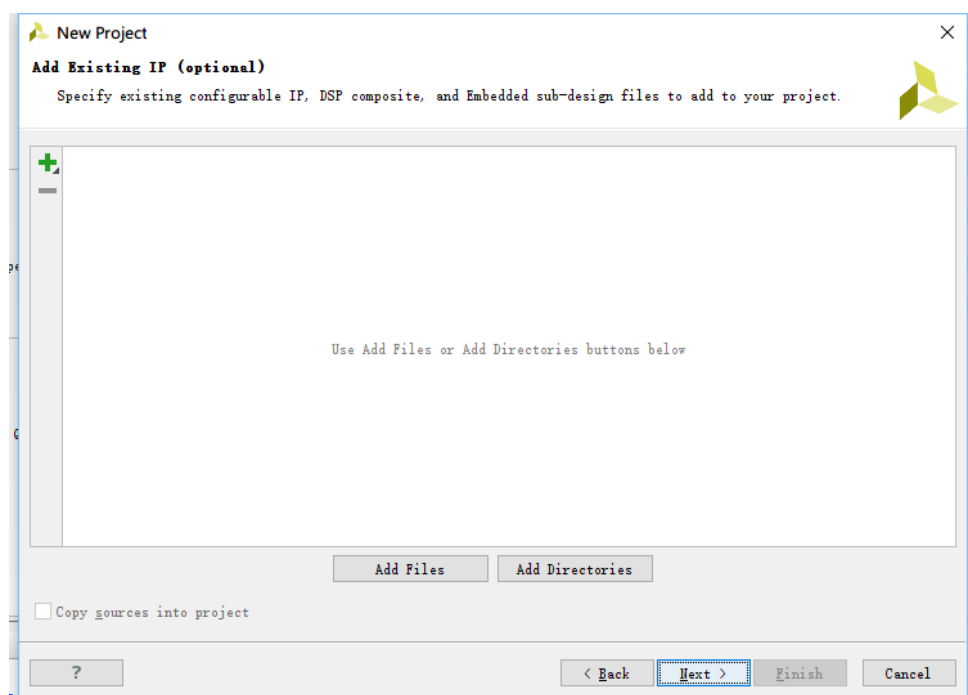


点击 OK

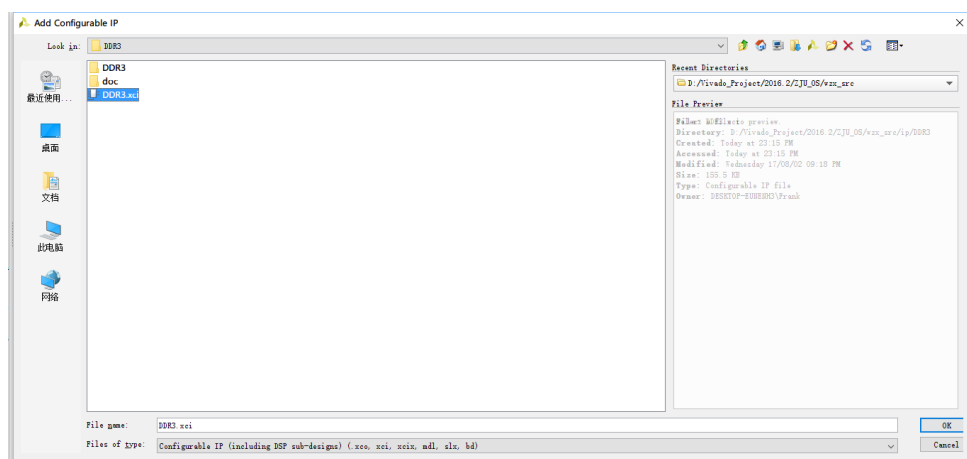


然后点击 Next，注意缺省是选择 Add sources from subdirectories

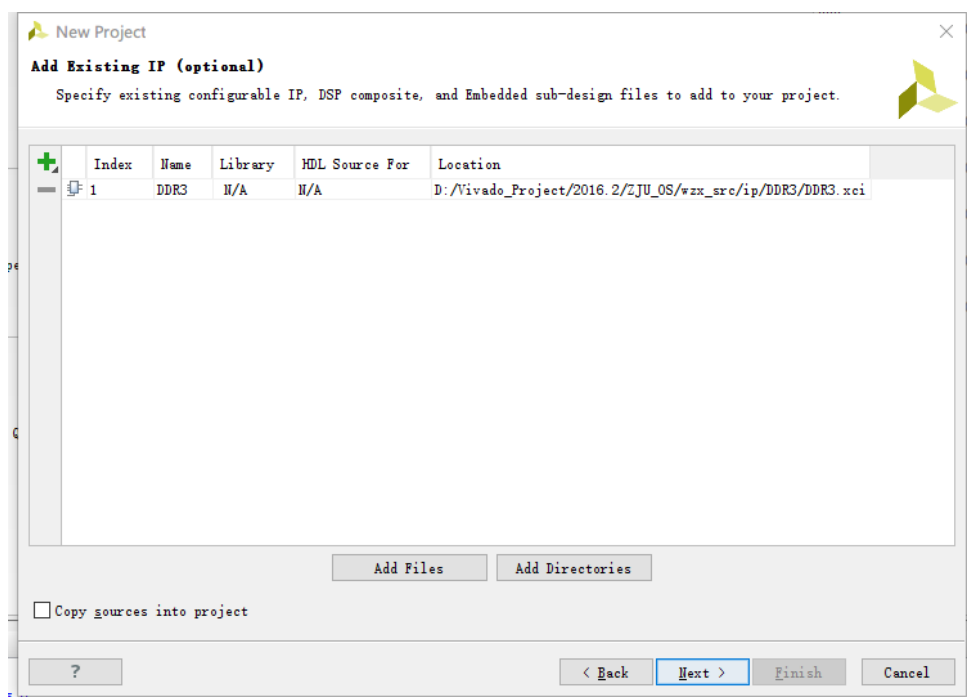
## 2.3.5 添加 DDR3 IP 核文件【第四步】



这里，我们需要把现有的储存器 DDR3 IP 核导入，点击 Add Files



选择 IP/DDR3/DDR3.xci 文件，点击 OK

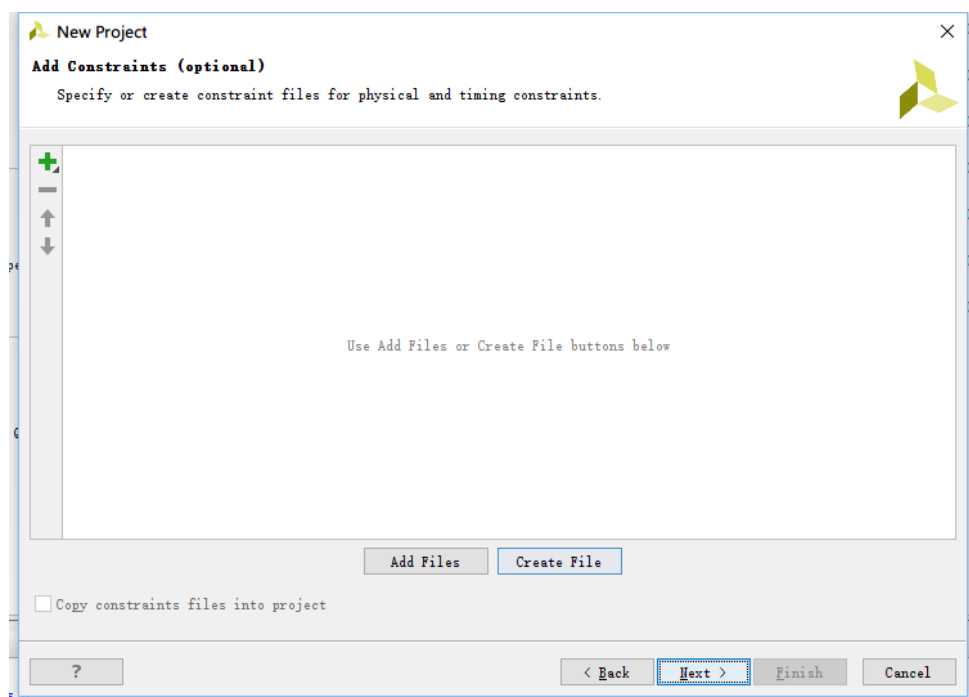


点击 Next

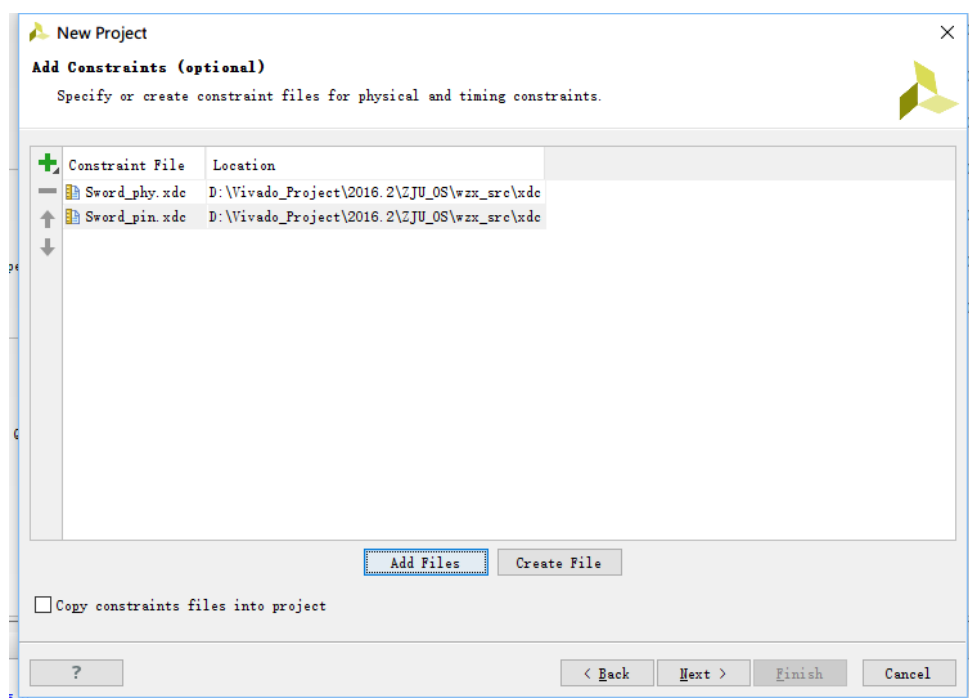
下一步，添加引脚约束文件

## 2.3.6 添加引脚约束文件【第五步】

点击 Add Files

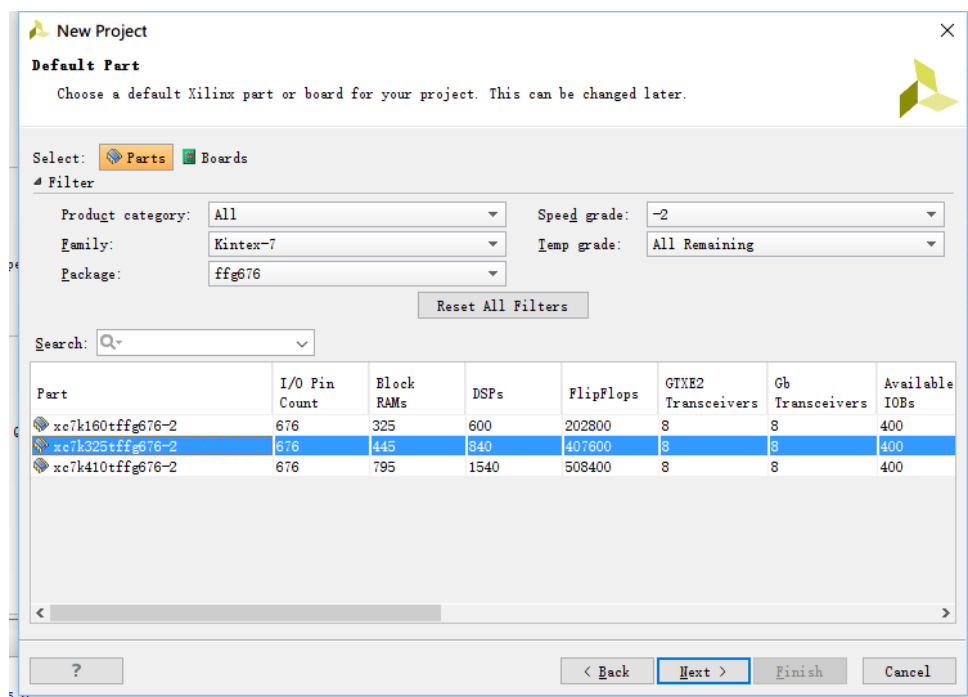


选择 xdc 文件夹下面的两个 xdc 文件，添加进来



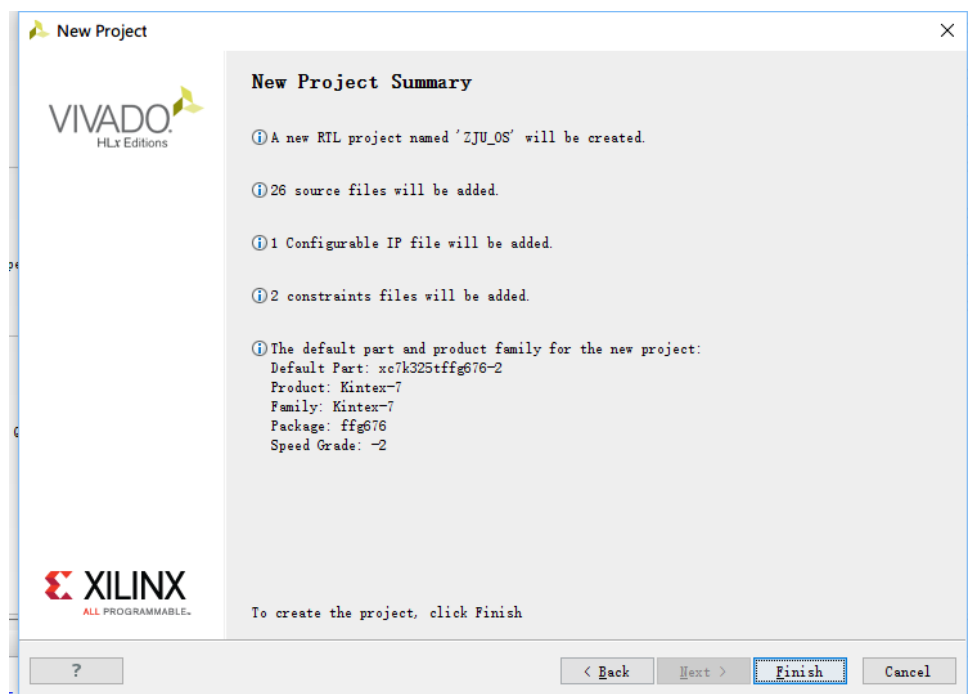
再点击 Next

## 2.3.7 选择开发板（xc7k325tffg676-2）【第六步】



点击 Next

最后确认一下工程类型

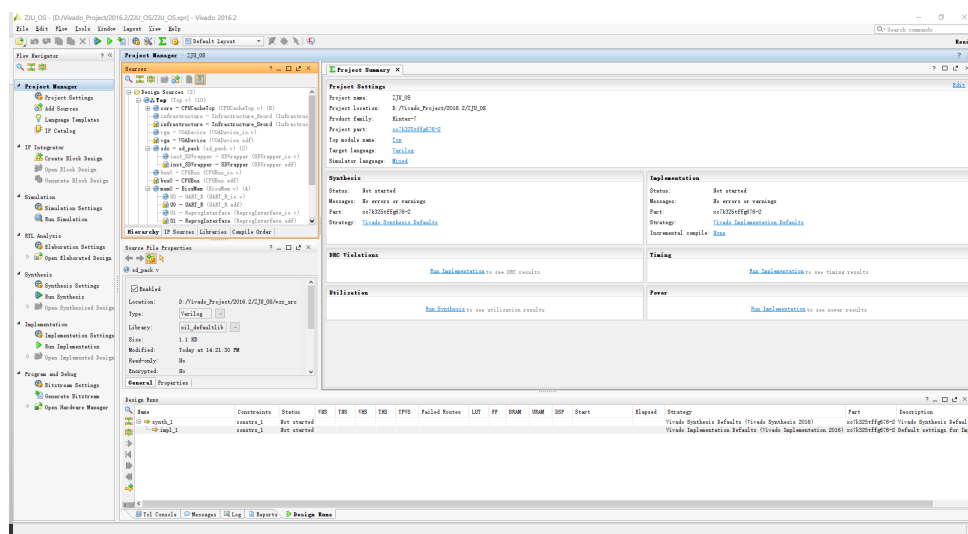


点击 Finish

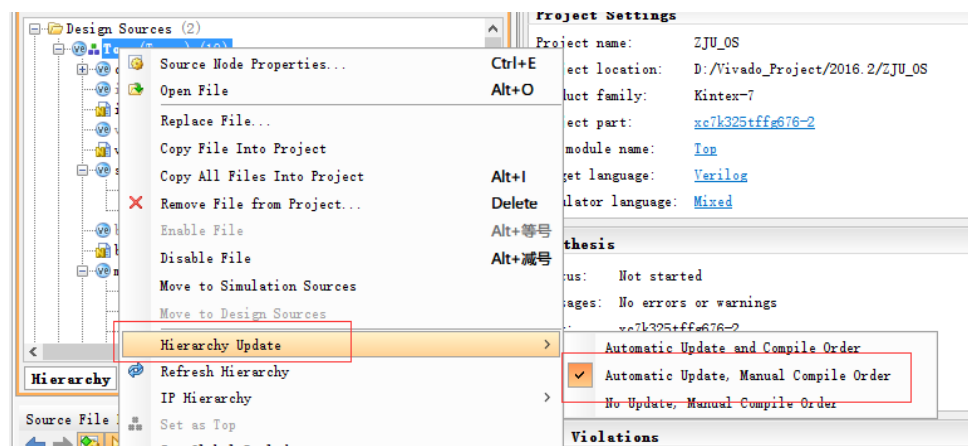
稍等片刻 Vivado 会自动创建好工程。

## 2.3.8 修改编译设置【第七步】

vivado 的主界面如下

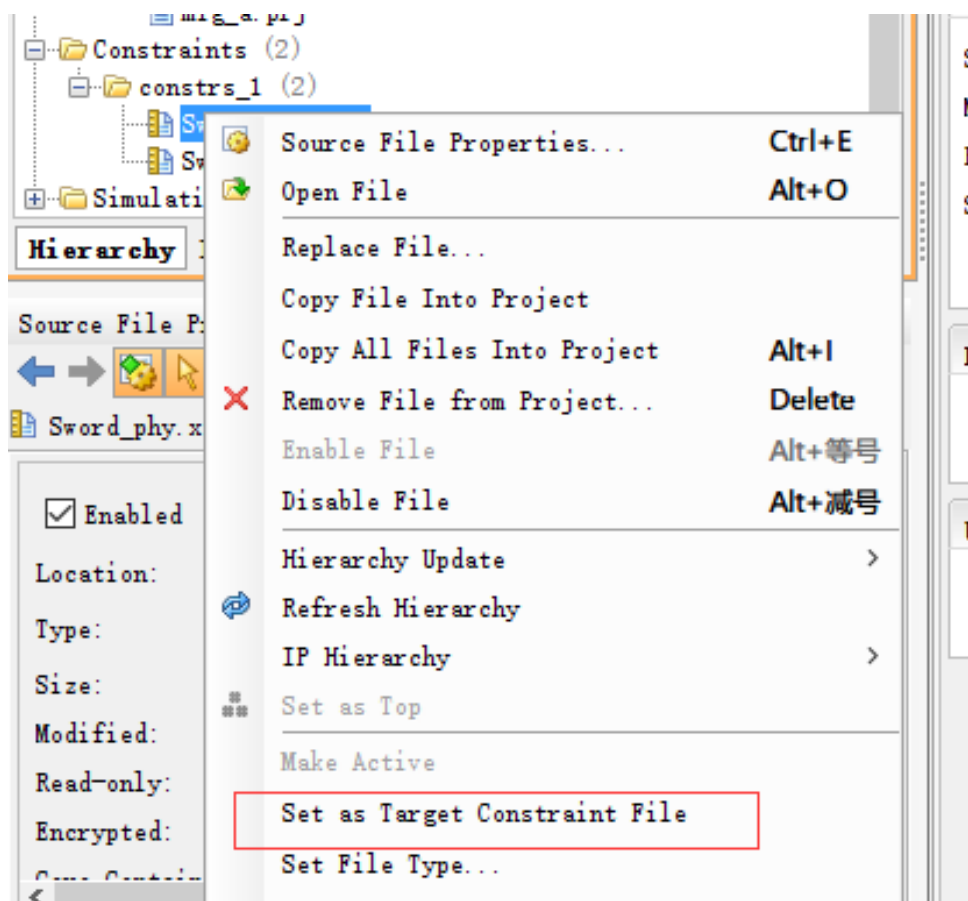


在 Source 窗口的 Design Sources 内，单击右键，选择 Hierarchy Update -> Automatic Update, Manual Compile Order



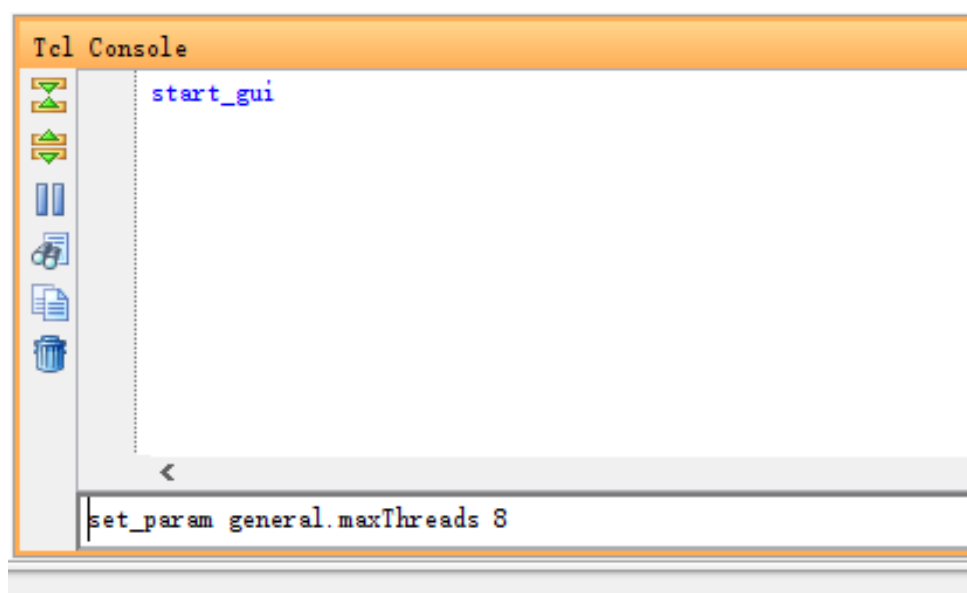
然后在 Constraints 的 Sword\_phy.xdc 文件上面单击右键，选择 Set as Target Constraint File



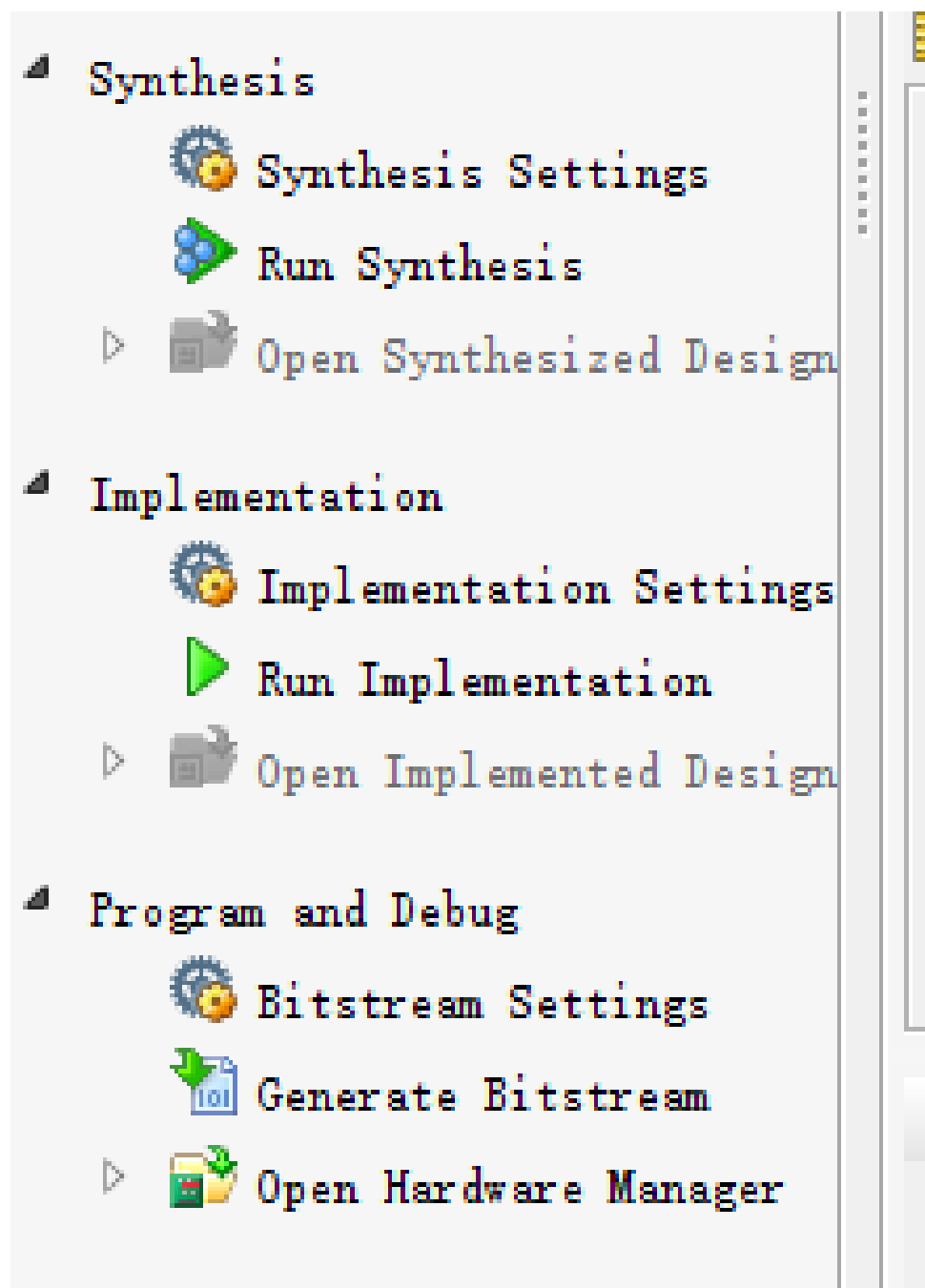


## 2.3.9 综合布线生成流文件【第八步】

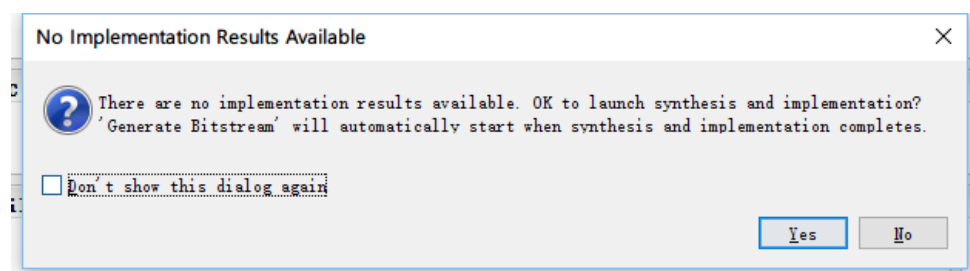
接下来，你可以依次点击 Run Synthesis -> Run Implementation -> Generate Bitstream，也可以直接点击 Generate Bitstream 来生成流文件。（Windows 下面综合默认是 2 线程，可以打开 Tcl Console，输入 `set_param general.maxThreads 8`，增加其综合速度。）



如果直接点击 Generate Bitstream 会提示你没有实现结果，需要先综合与实现，这里点击

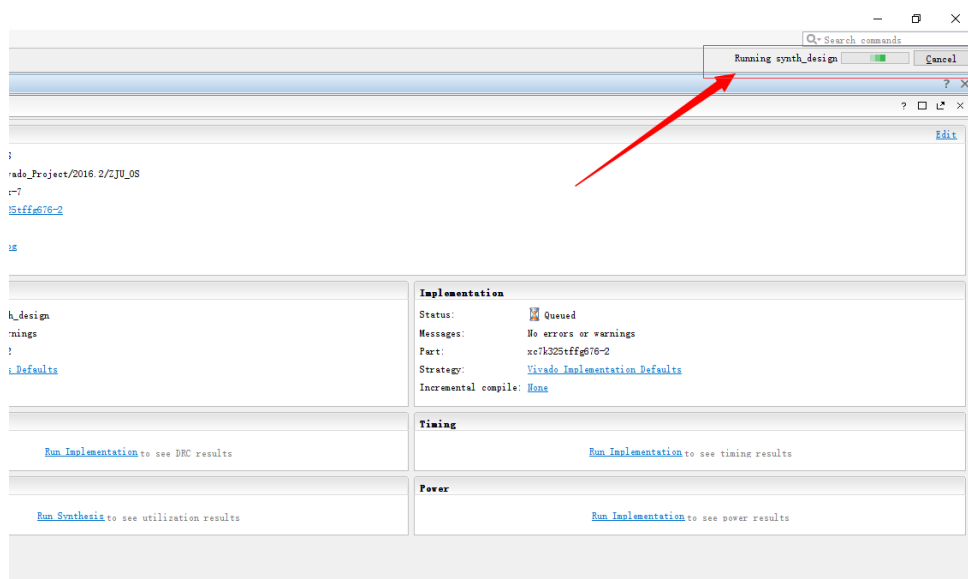


Yes 即可

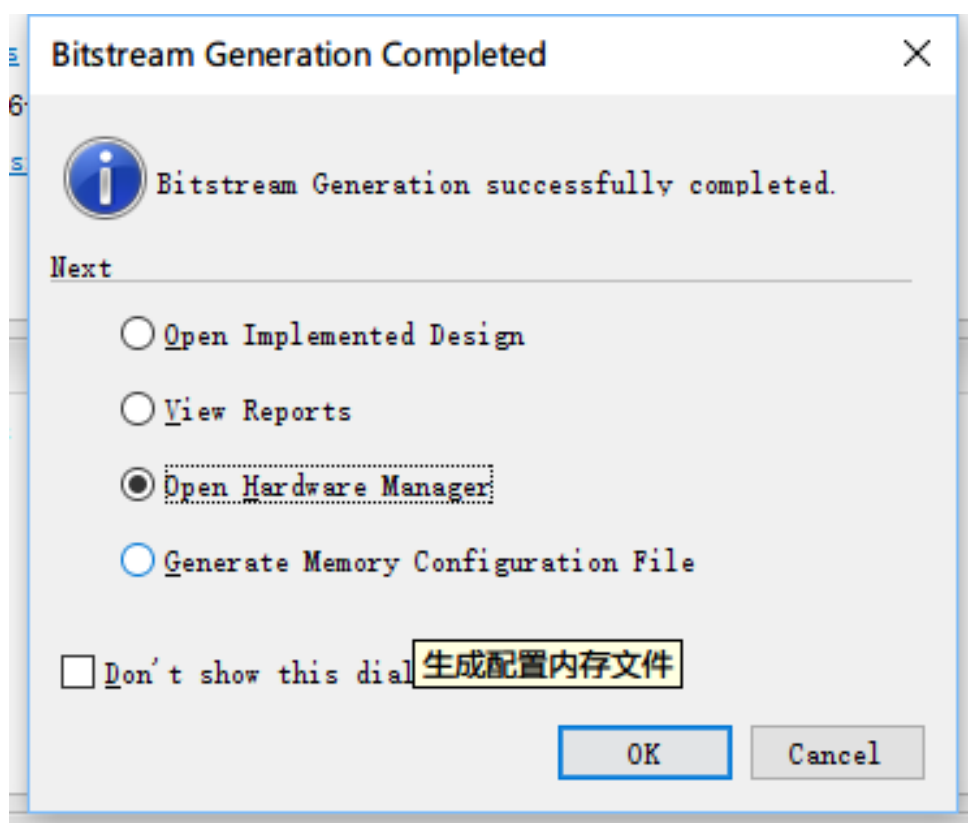


然后开始综合布线生成流文件，这个时间会比较长，大约需要 30 到 40 分钟不等（可能更长，根据机器配置而定）

下图这个位置是当前的进度状态。

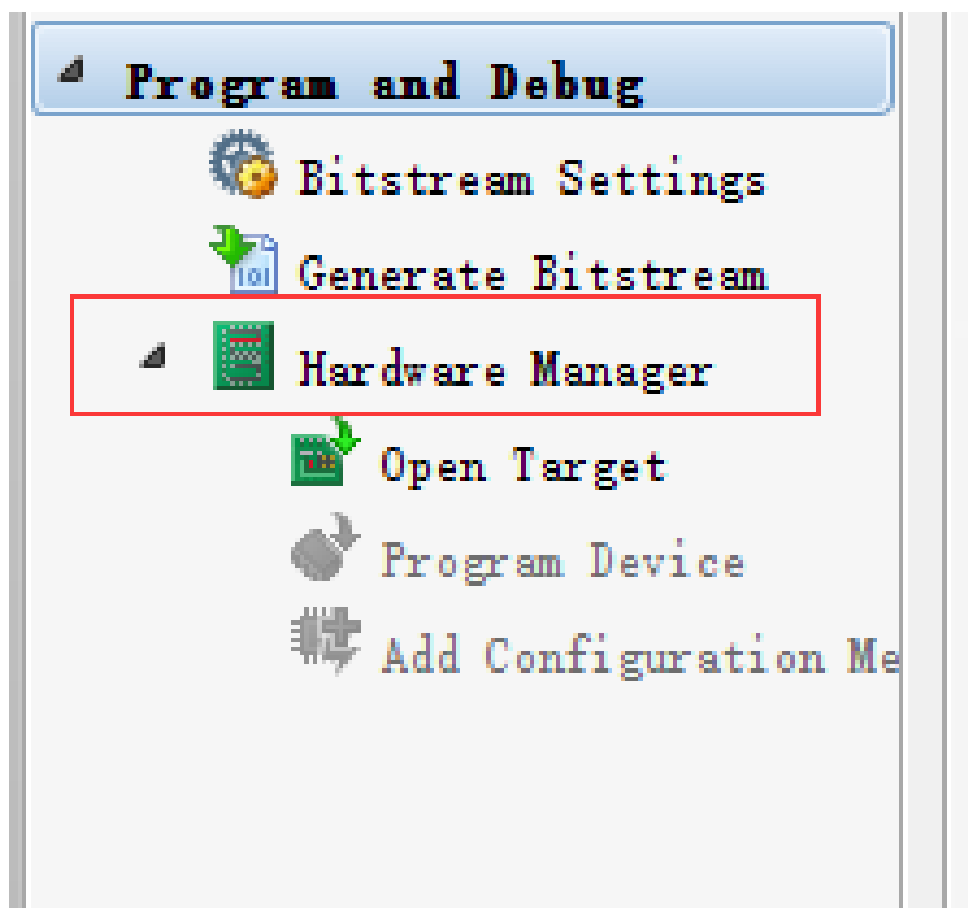


当生成完成时，就可以在 ZJU\_OS 工程目录的 ZJU\_OS.runs -> impl\_1 文件夹下面，找到 Top.bit 流文件了。

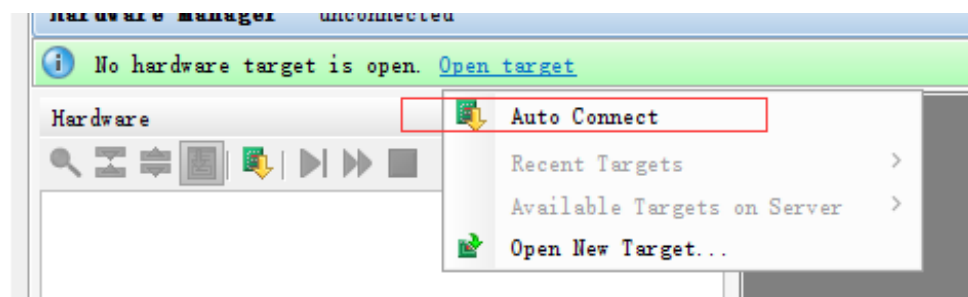


## 2.3.10 下载开发板

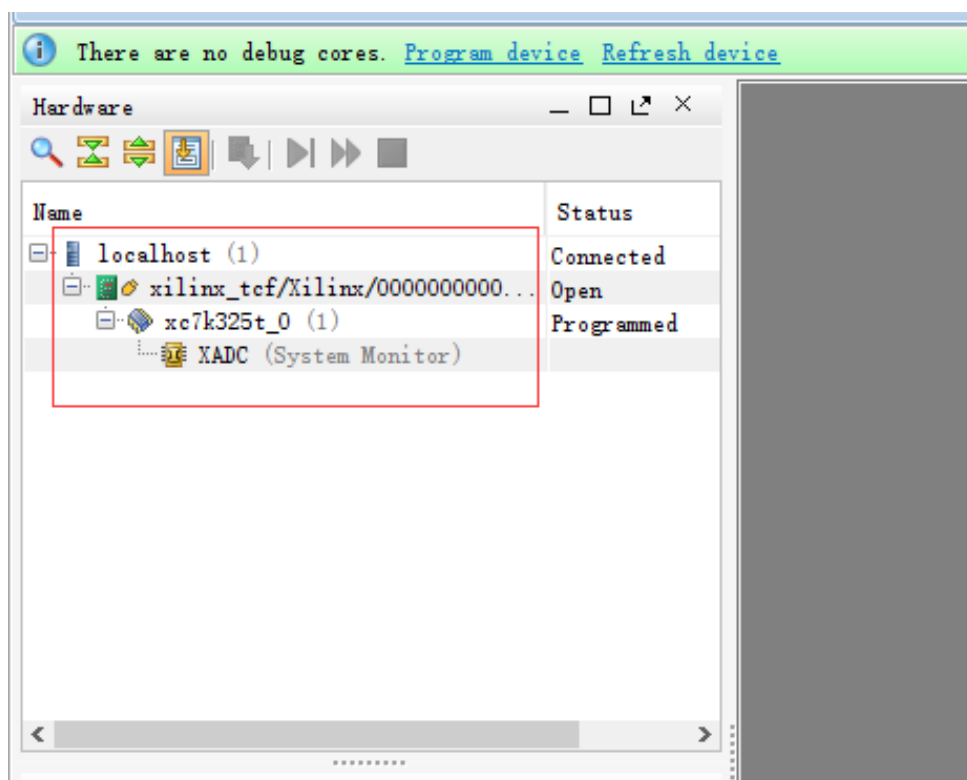
点击 Hardware Manager



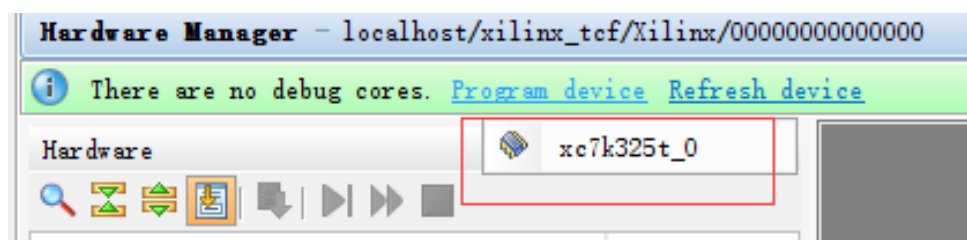
在弹出的硬件设备管理窗口里面，点击



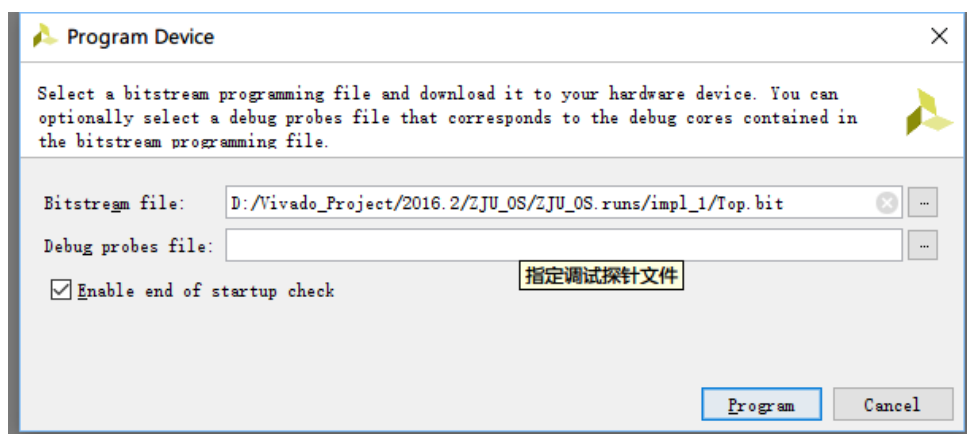
这样可以看到连接的设备（如果看不到，请检查连线 and 驱动）



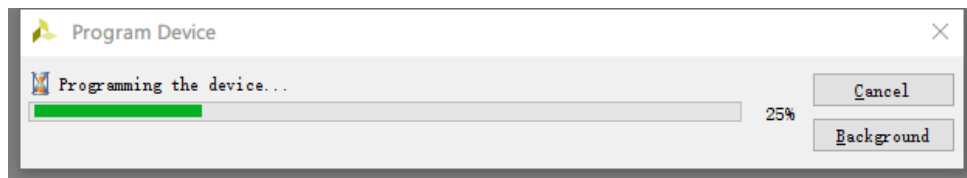
点击 Program device



在弹出窗口请检查路径是否正确



单击 Program



等待机器讲流文件烧写入板子（很快，约十几秒）