

# Vivado design flow

---

## 目录

1.硬件和软件环境.....	1
1.1 小车架构.....	1
1.2 软件环境.....	1
2.功能描述 .....	2
3.设计流程 .....	2
3.1 创建自己的 IP core. ....	2
3.2 使用 IP 设计工程.....	10

## 1.硬件和软件环境

### 1.1 小车架构

小车分为 3 层：

底层：车轮、电机、电池

中间：robot bottom board

上层：zedboard、RF 模块、bluetooth 模块

准备：具备蓝牙功能、安装好小车控制 app 的安卓手机或 serial port utility 串口调试助手，带 RF 模块和发射程序的板子。

注：本小车默认为 RF 接收，zynq 上的 LD5 向上拨选择为蓝牙接收。

### 1.2 软件环境

Vivado 2013.2

## 2.功能描述

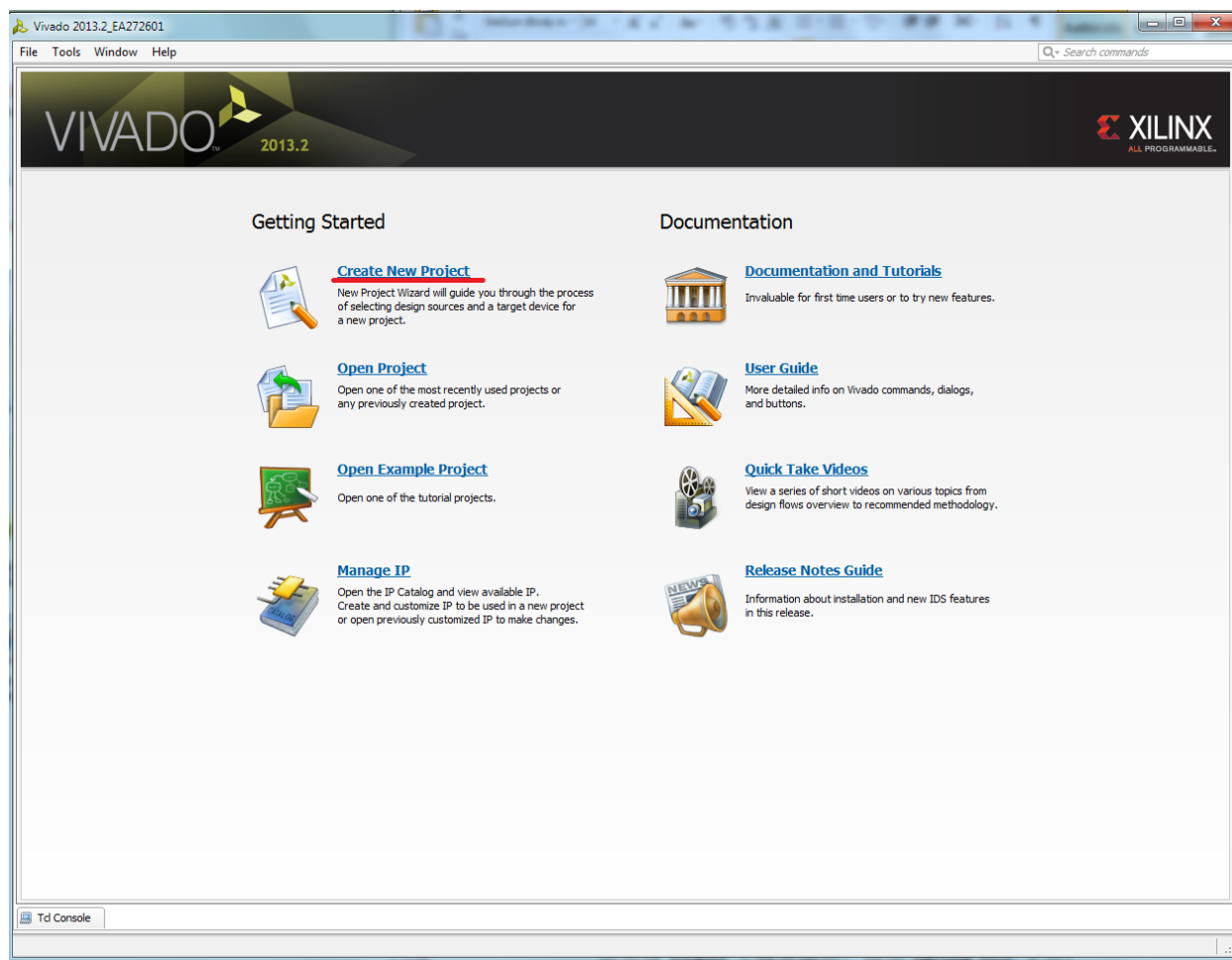
小车基本功能：前进、后退、左转、右转，正常速度档和加速档，可以通过 RF 模块在另外一块板子上的按钮控制，也可以通过电脑上的串口调试助手以发送命令的方式控制，也可以通过安卓手机客户端控制。

## 3.设计流程

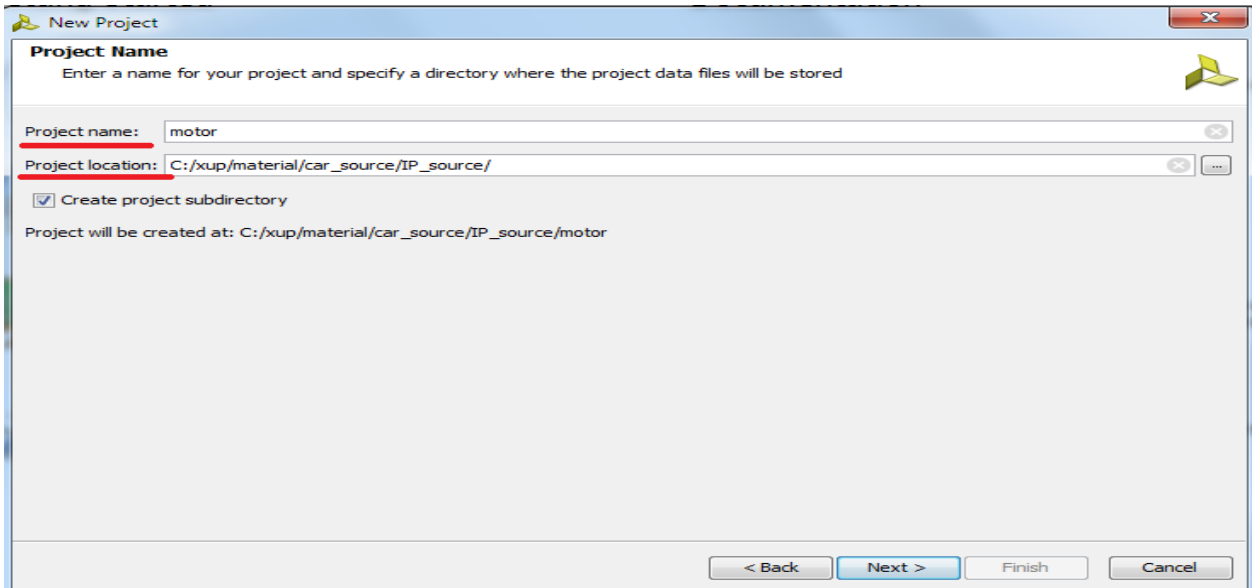
### 3.1 创建自己的 IP core.

这里我们以 motor 电机控制 IP 为例，一步一步教大家在 vivado 中创建自己的 IPcore.

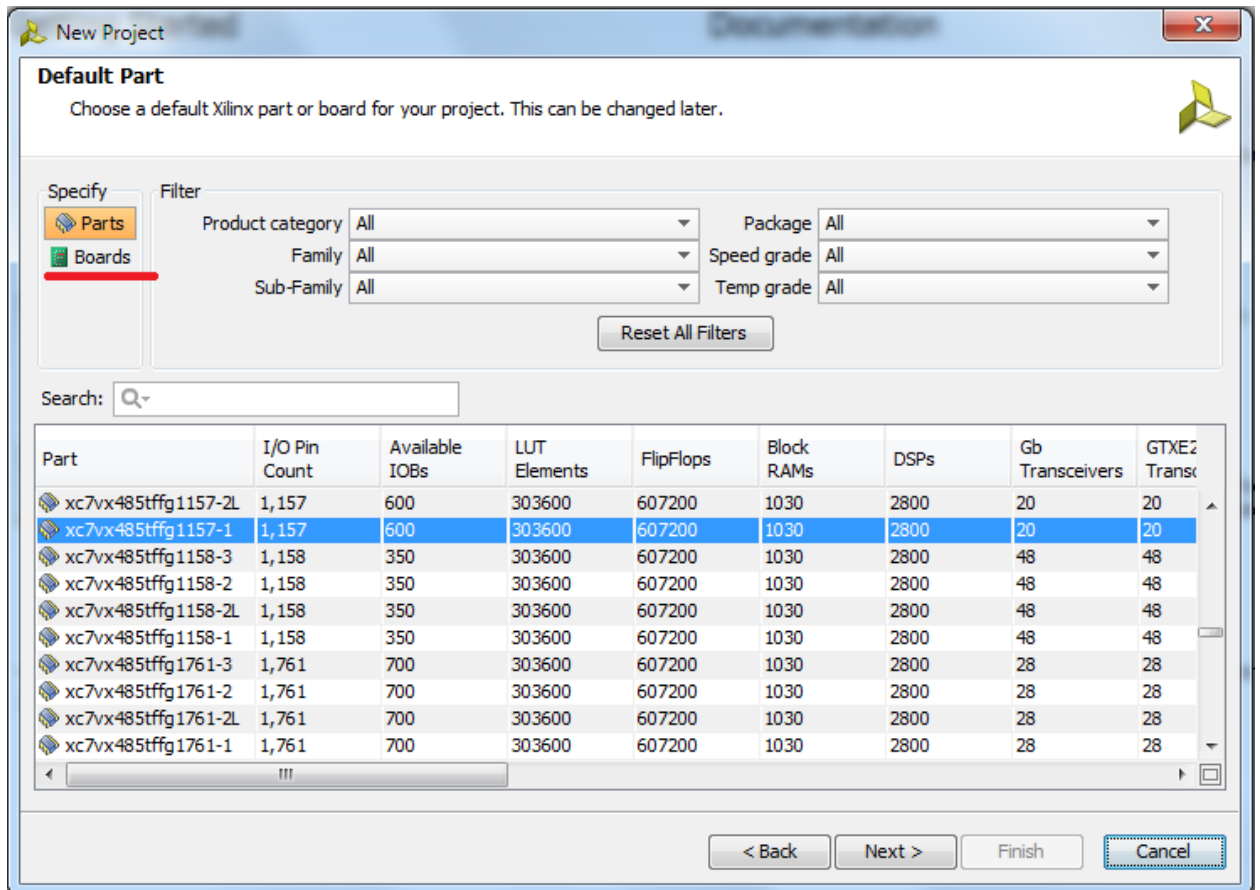
- 双击 vivado 图标，出现如下界面，点击 **Creat New Project**:

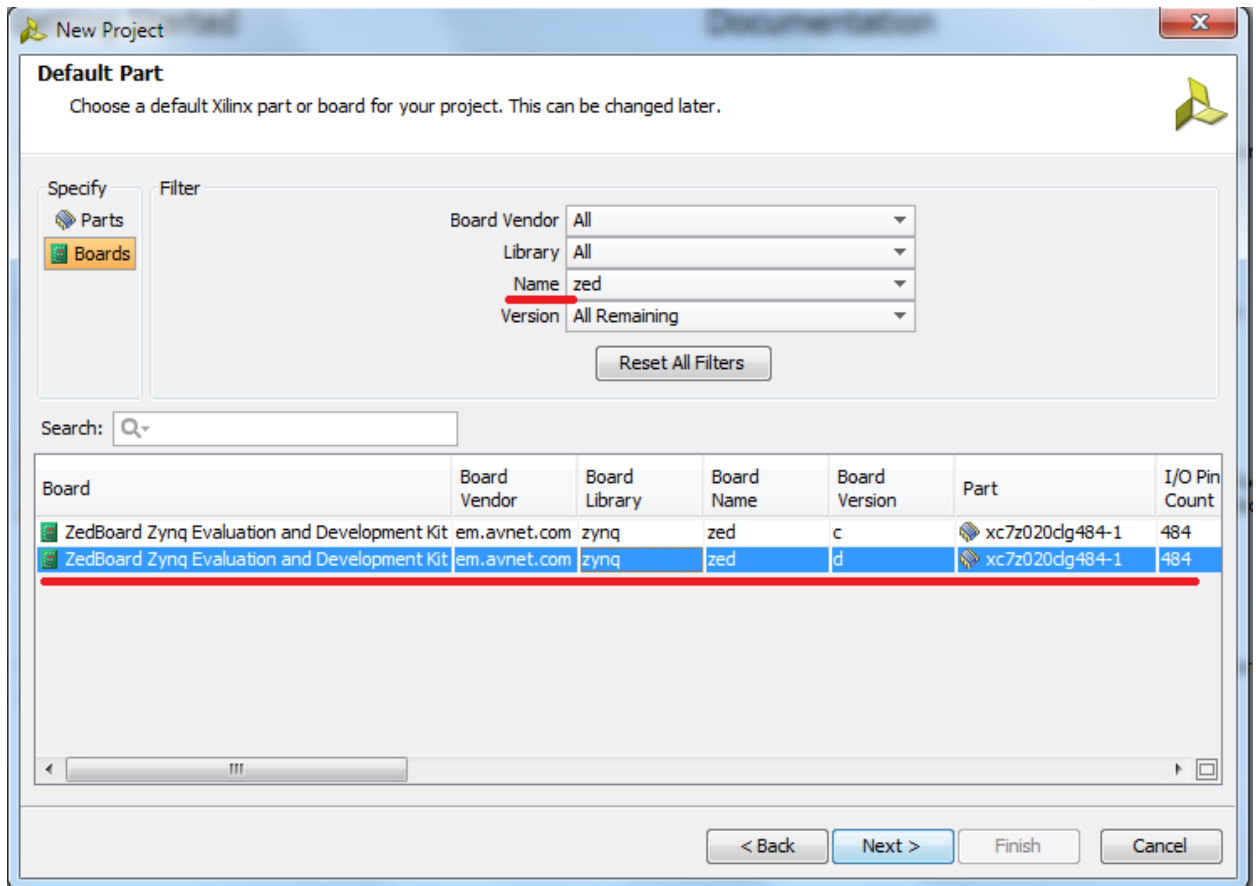


- 点击 Next，出现如下界面，project name 一栏输入 motor.

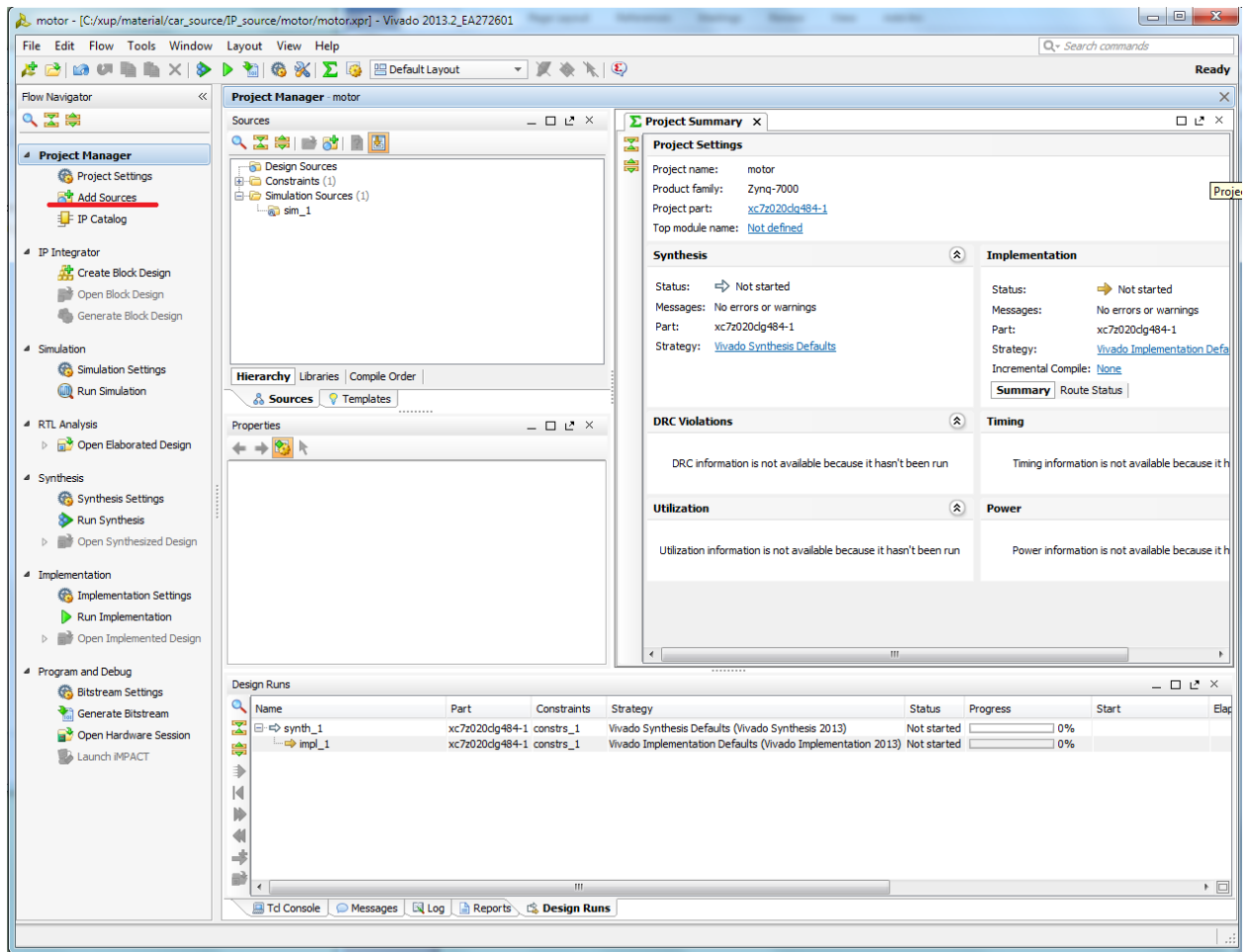


- 一直点击 Next，直到出现如下界面，单击 Boards，在 name 栏里选择 zed，并点击第二个 board version 为 d 的板子选项：

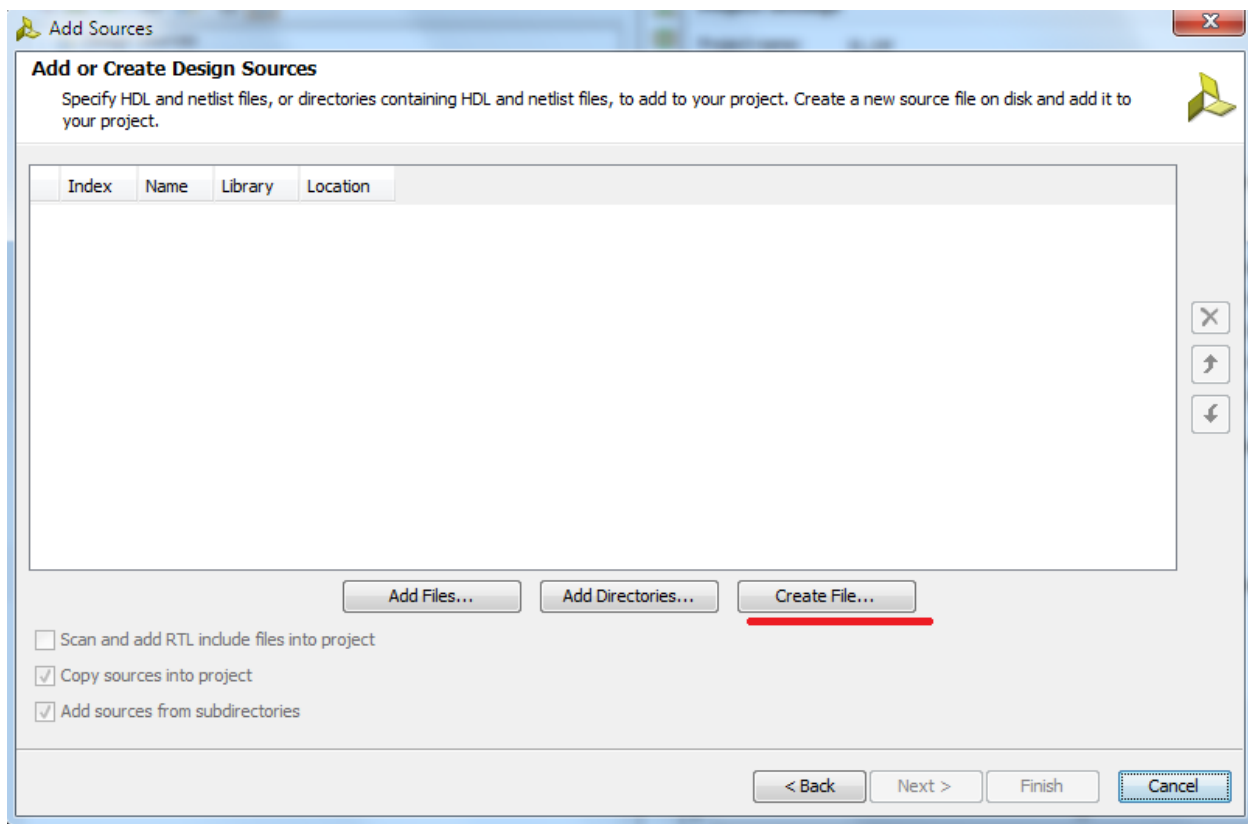




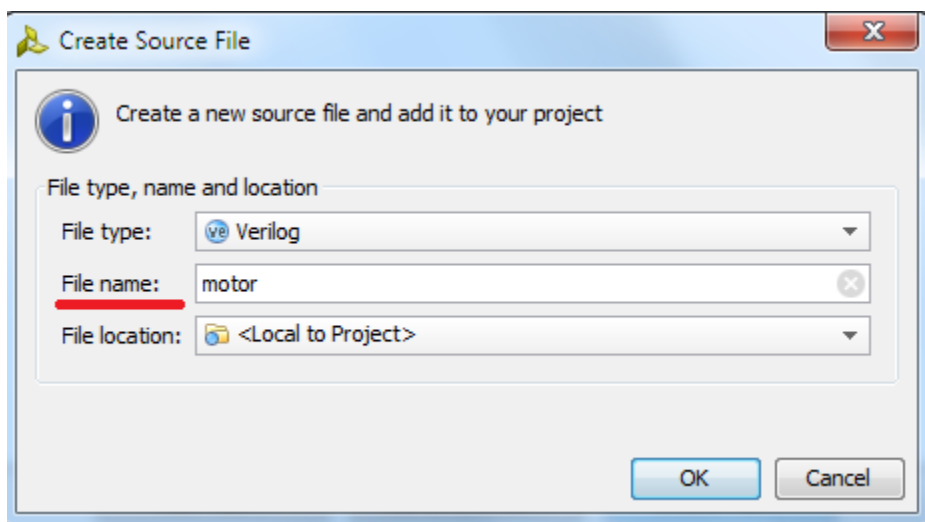
- 点击 **Next**，在接下来的界面中点击 **Finish**，这样我们的一个 IP 工程就新建好了。接下来我们要在 **vivado** 工程中创建自己的 **verilog** HDL 文件，在左边的控制面板中点击 **Add Sources**：



- 在 Add Source 界面中，选择 Add or Create Design Sources，点击 Next，在出现的界面中单击 Create File:

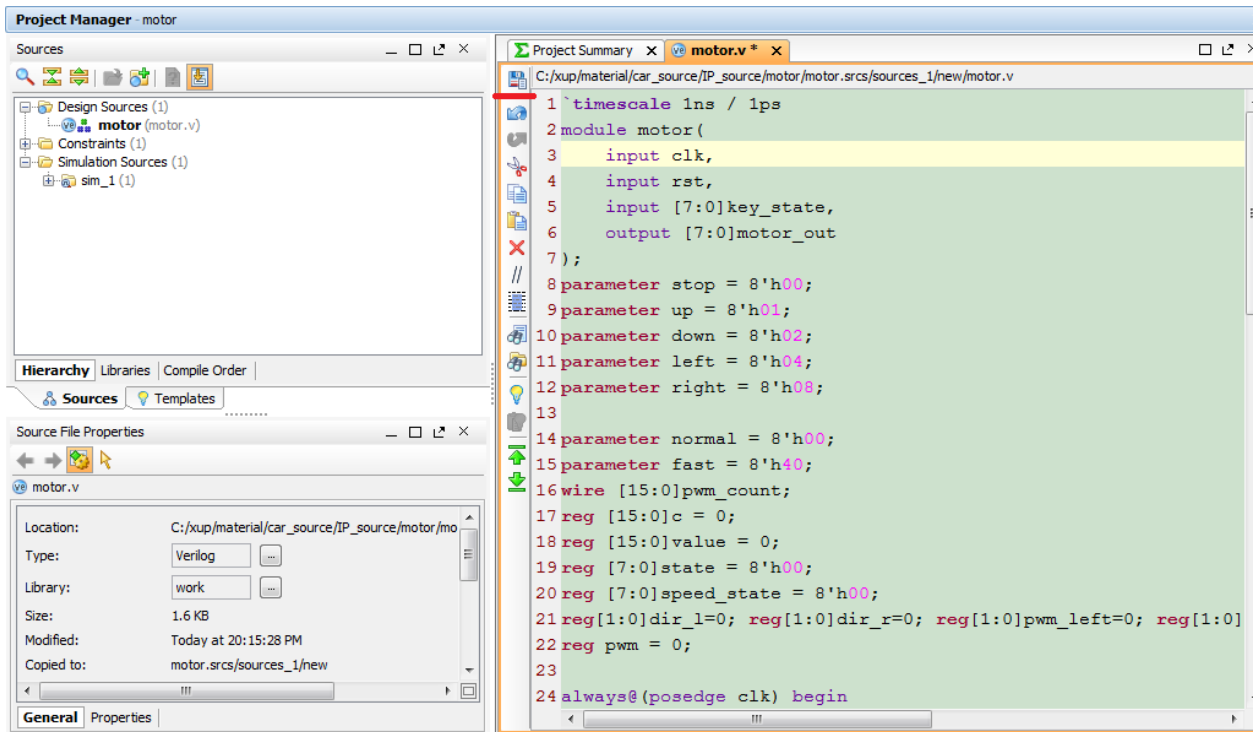


- 在 Create Source File 界面中，输入 File name :motor,点击 OK，再点击 Finish。之后出现 Define Moudle 界面，不做任何选项直接点击 OK，在弹出的界面中点击 Yes:



- 完成上面步骤后，可以在 Design Sources 下看到自己建立的 verilog hdl 文件，双击它，在右侧出现的界面中可以编辑他，可以自己写 verilog hdl 代

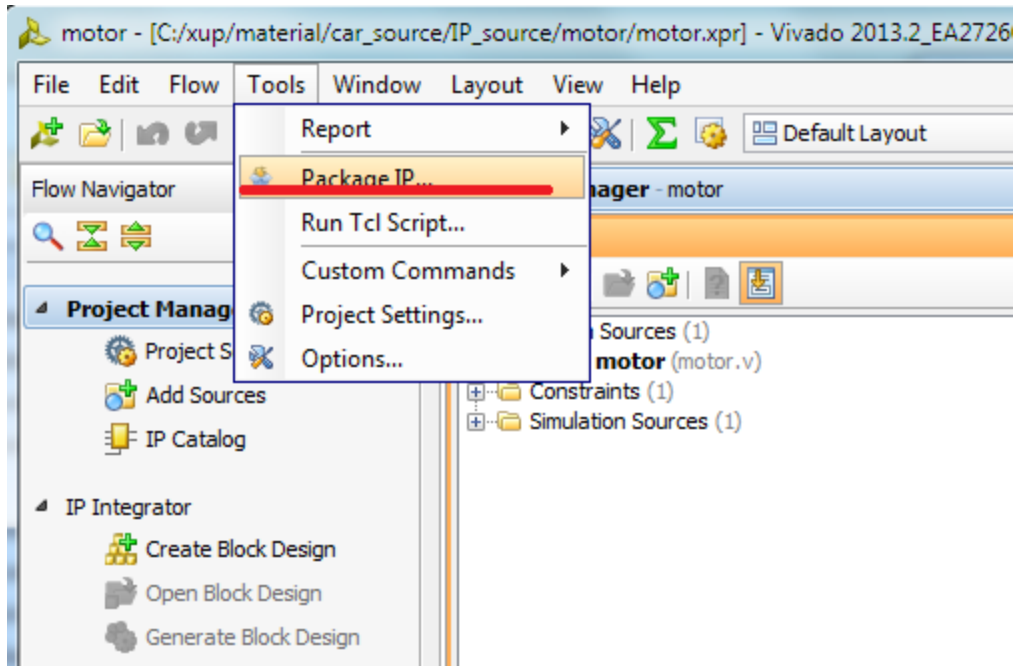
码,这里我们直接拷贝 workflow/HDL/motor.src/motor.v, 可以使用文本编辑器打开将代码直接复制过来就行了, 如下所示:



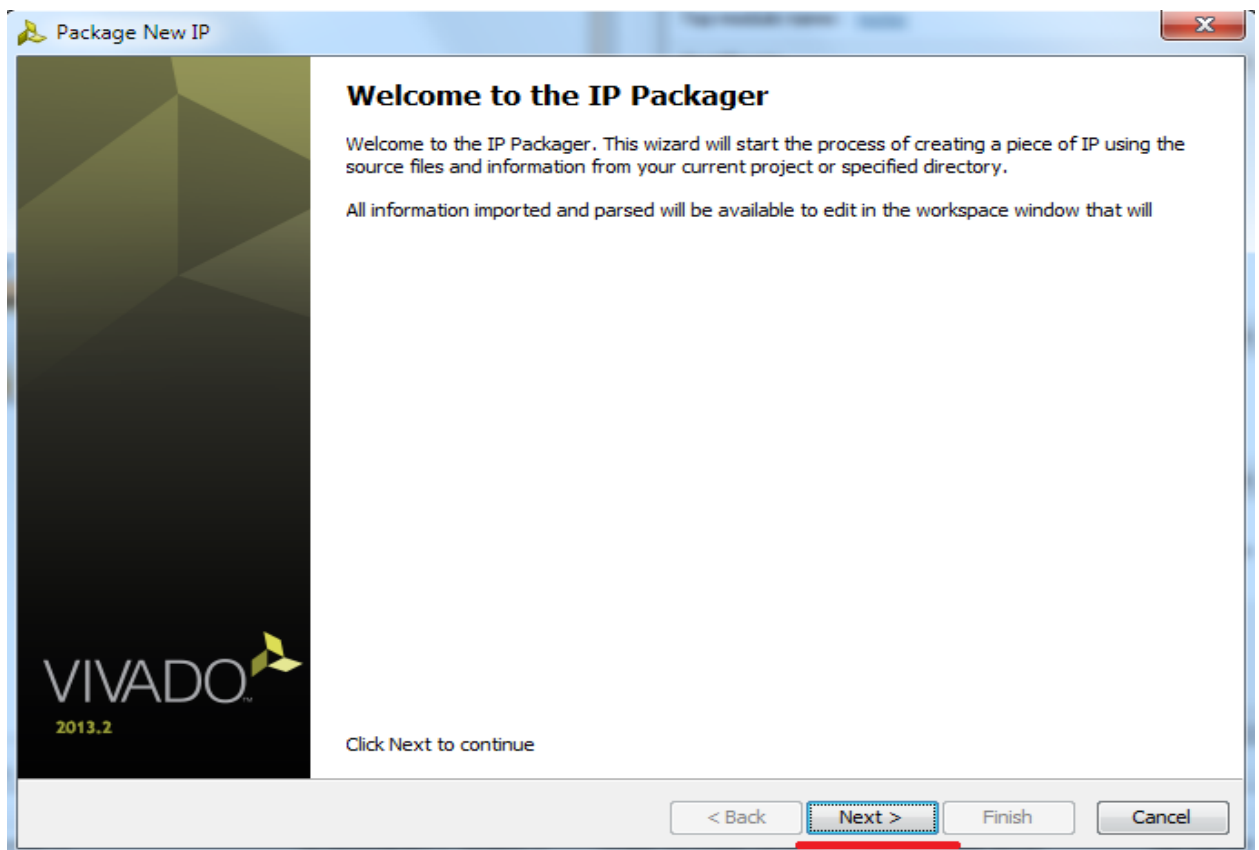
- 至此, 一个控制电机转动和方向、并根据输入按键信息进行 PWM 调速的电机工程完成了。可以通过点击左侧的 Run Synthesis 和 Run Impementation 来查看错误与否: 接下来我们将生成的 vivado 工程封装成 IPCore, 以便以后使用。

注: 可以添加约束单独跑这个工程, 但这不是本次的目的, 这里不展开了, 具体细节流程可以到 <http://www.xilinx.com>, 注册账号、查阅相关信息。

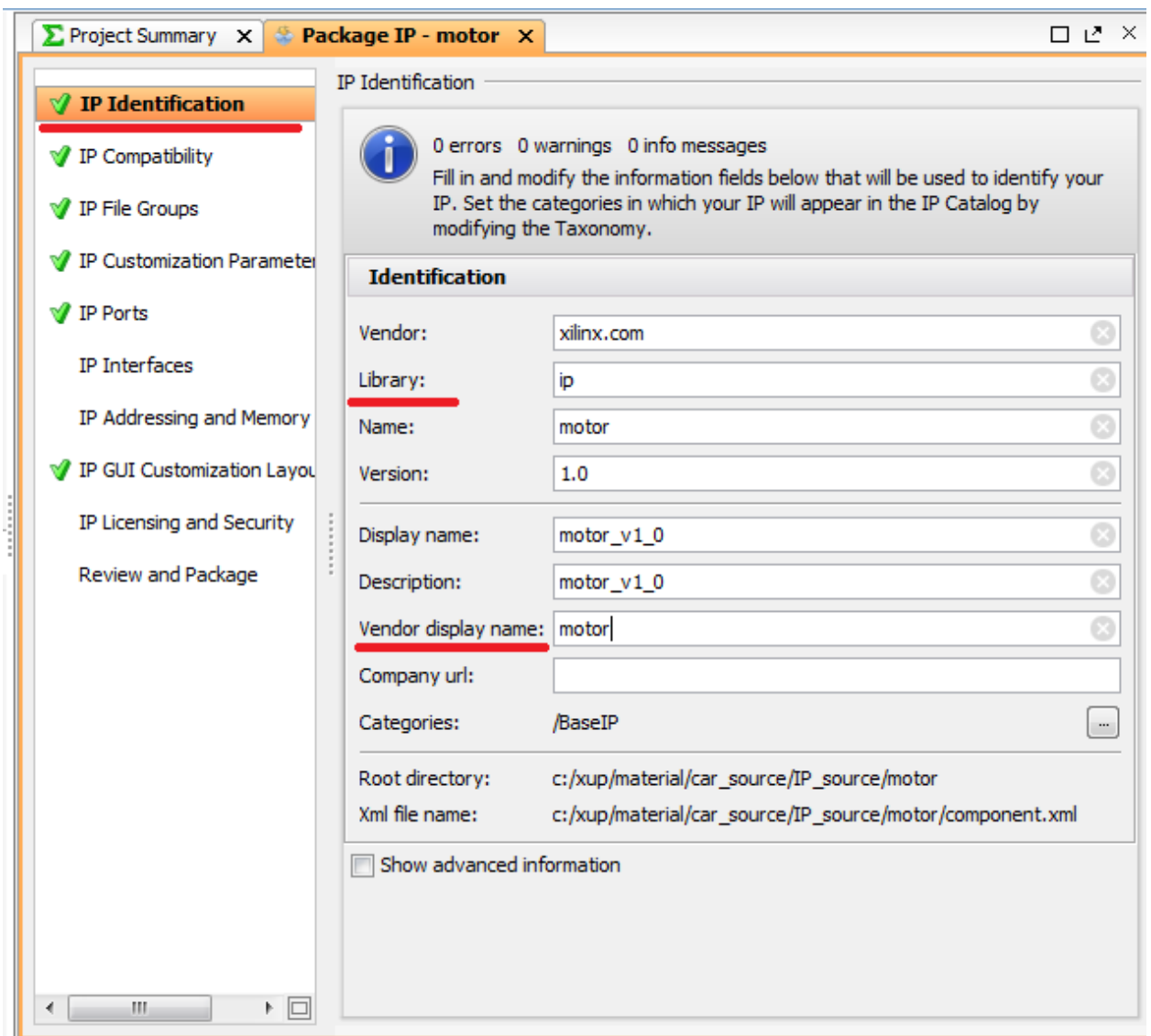
- 以下讲述封装 IP 的方法, 在菜单栏里单击 Tools, 在其下拉单击 package IP 选项:



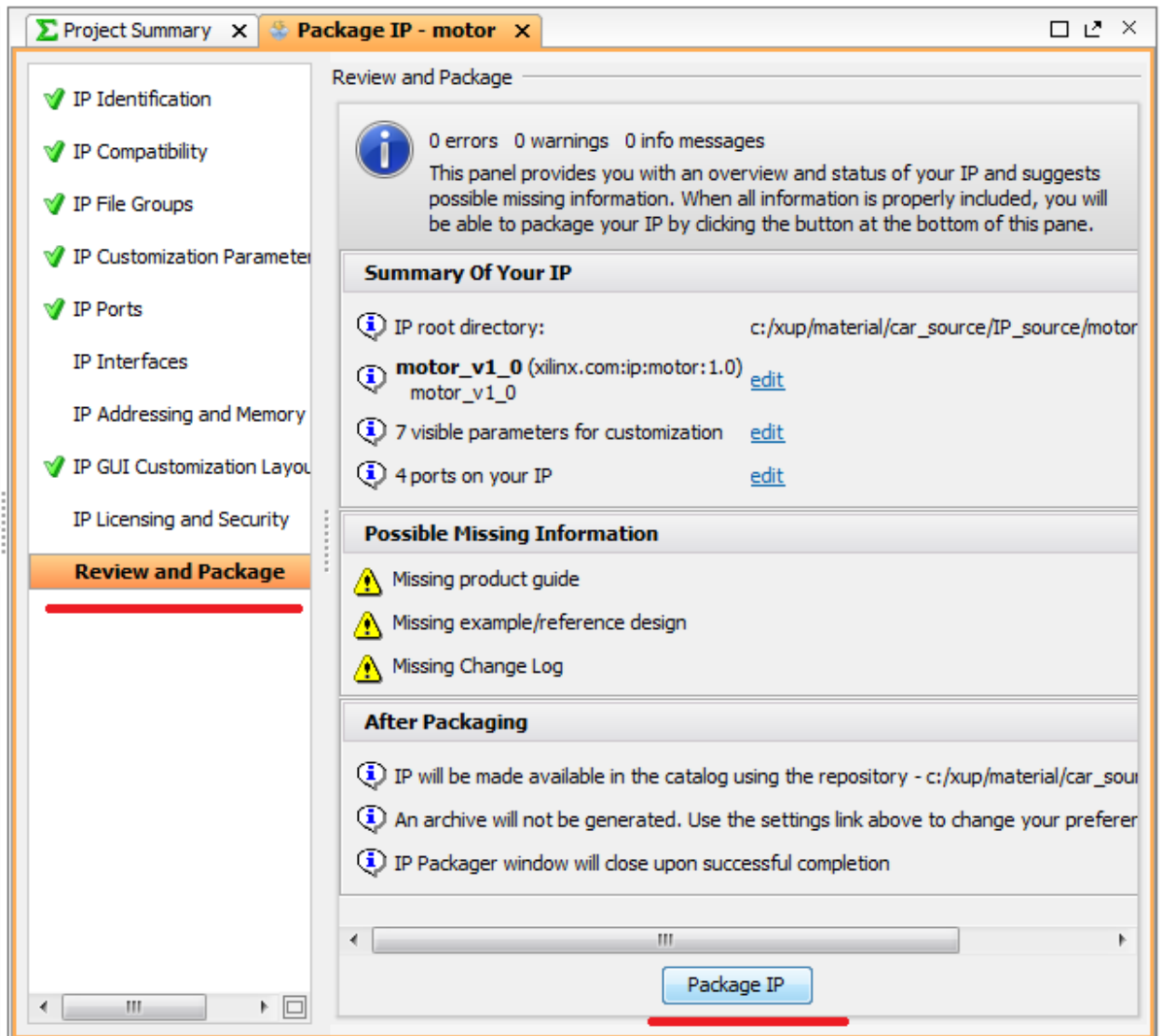
- 出现如下 Package New IP 欢迎界面：点击 Next，一直到点击 Finish，在出现的 IP Packager Summary 中单击 OK，出现如下界面，默认是 IP Identification 选项，library 和 vendor display name 可以随意修改：







- 修改后，选择 **Review and Package** 选项，并点击下方的 **package IP**：



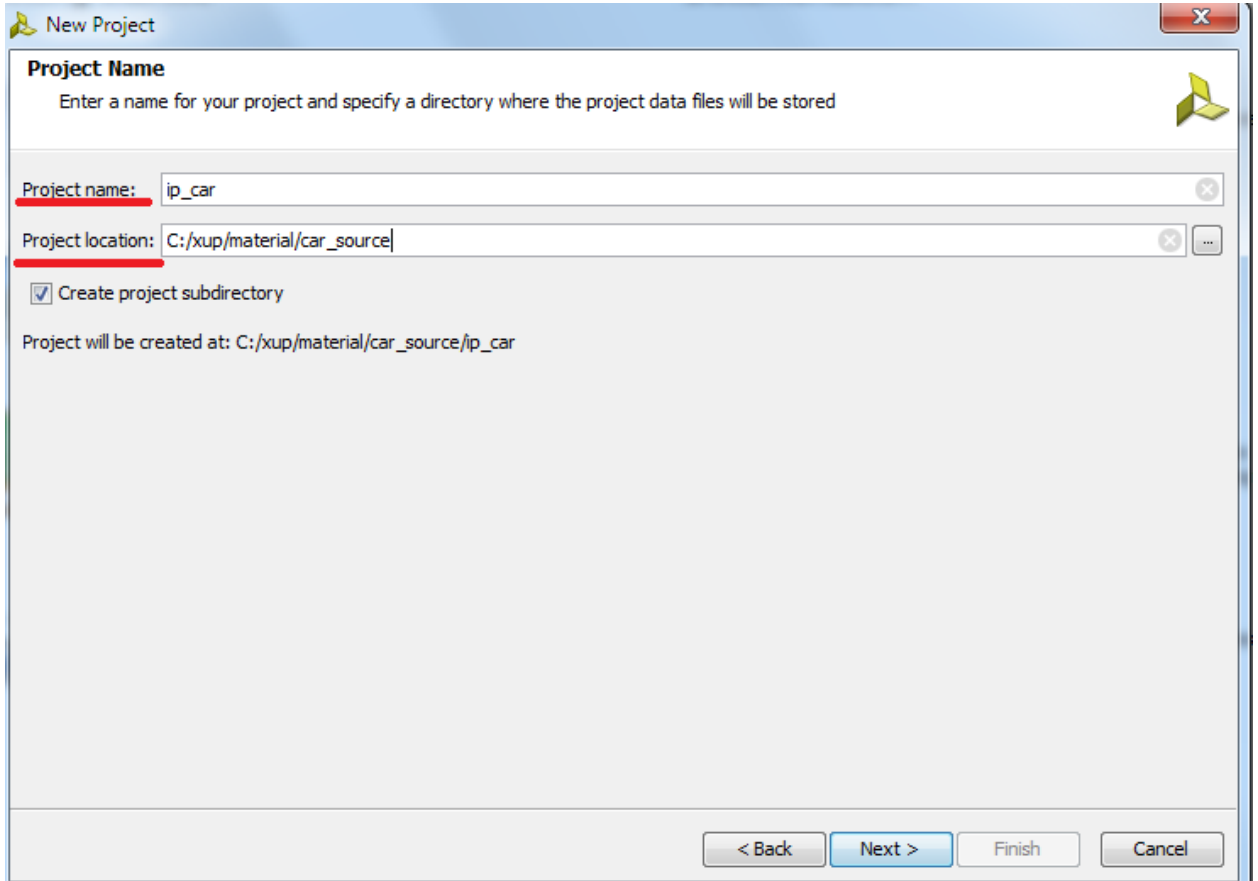
- 至此，我们的 IPcore 就生成完成了。在我们的工程目录下会出现一个 component.xml 文件，此文件就是我们的最终想要的 IPcore。

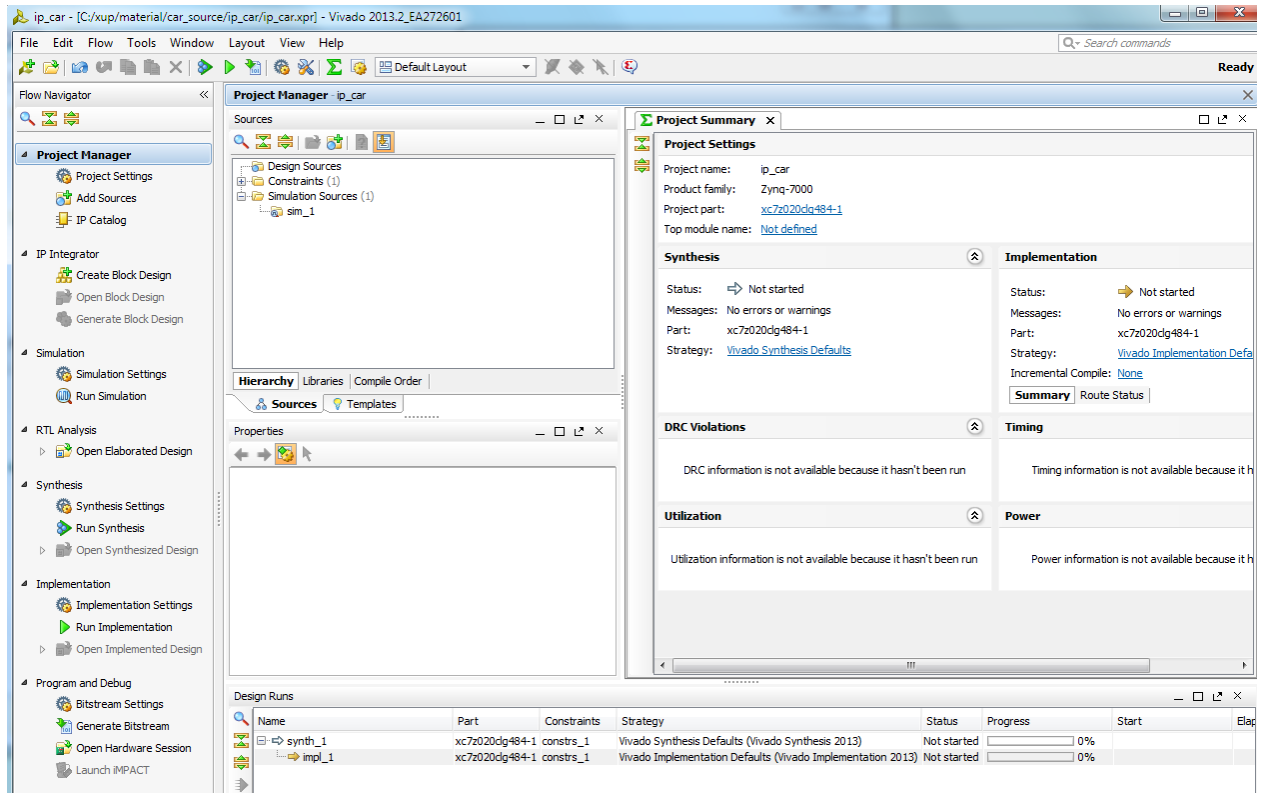
### 3.2 使用 IP 设计工程

- 在 3.1 节，我们学会了如何 create 一个 vivado 工程，如何添加自己的 HDL 代码，并且学会了如何封装 IP。下面我们来讲解如何将我们的各个功能模块联系起来，构建可视化 wrapper。
- Workflow/ip\_cores/下的文件是经过 3.1、3.2 实现的已经封装好 IP 的工程文件，包括 clkdiv、RF\_r、uart、motor，其功能分别为时钟分频、RF 接收、蓝牙收发和电机驱动，其中 centre 工程起到为这些工程建立简单的数据关联作用。

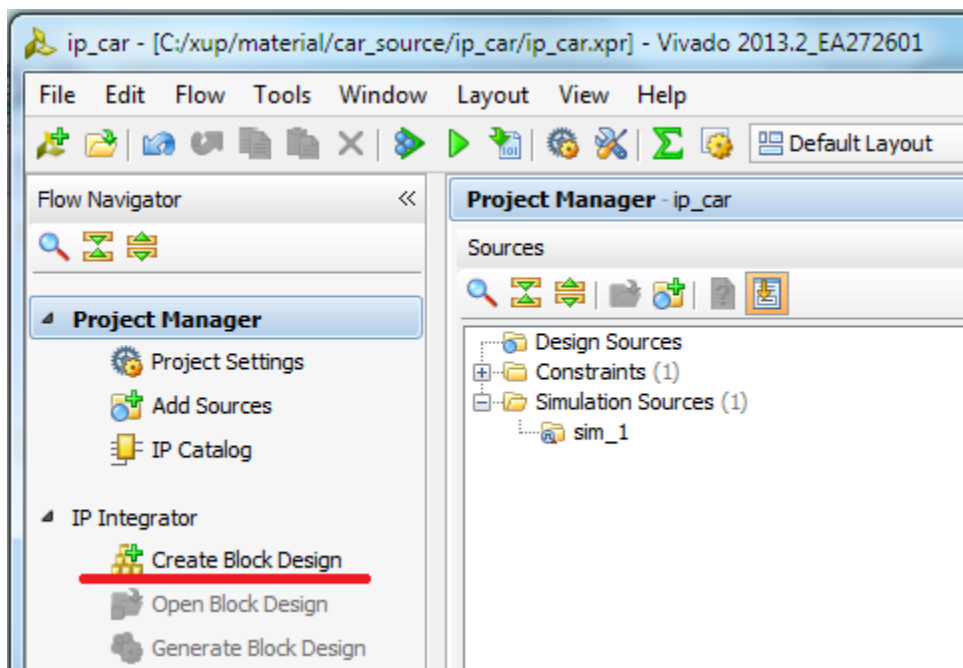
这里我们希望同学们能够根据 3.1 节的内容自己动手完成剩余 4 个 IPcore 的创建。这里我们不在一一累述，在 workflow/ip\_cores/目录下是我们已经做好的 ipcore。

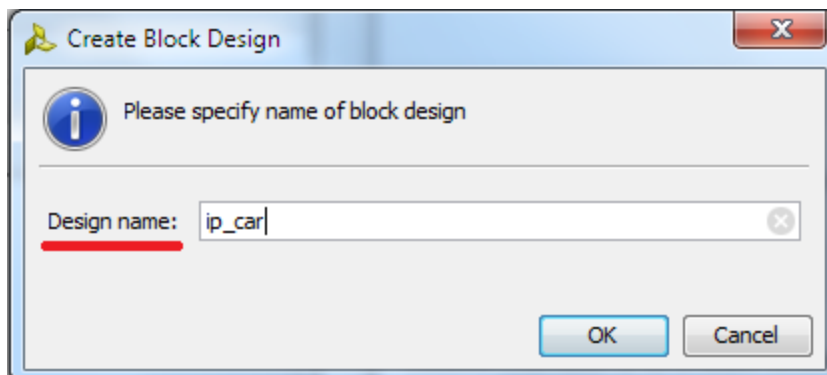
- 下面我们将要建立一个综合工程，然后将上面我们已经创建的 ipcore 联系起来一起来驱动我们的小车。以下是使用这些 IP 建立自己的小车工程，首先参考 3.1 建立一个空白工程。修改如下：直到出现如下空白工程界面：



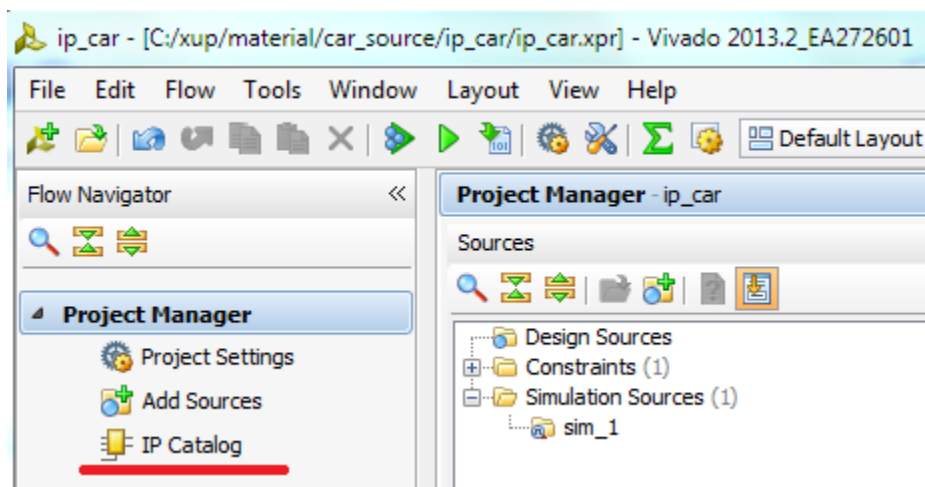


- 点击左侧 IP Integrator 下的 Create Block Design: 输入 Design name, 点击 OK。

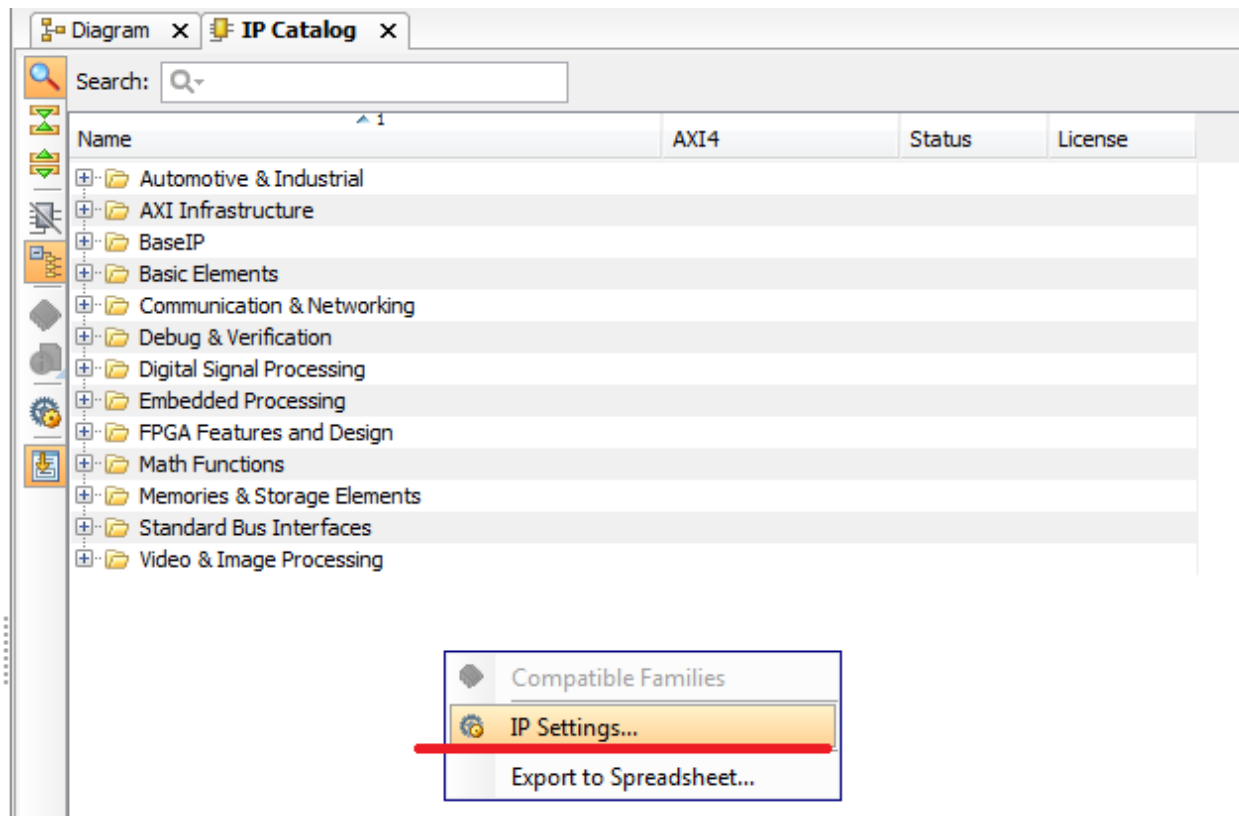




- 点击 Project Manager 下 IP Catalog:

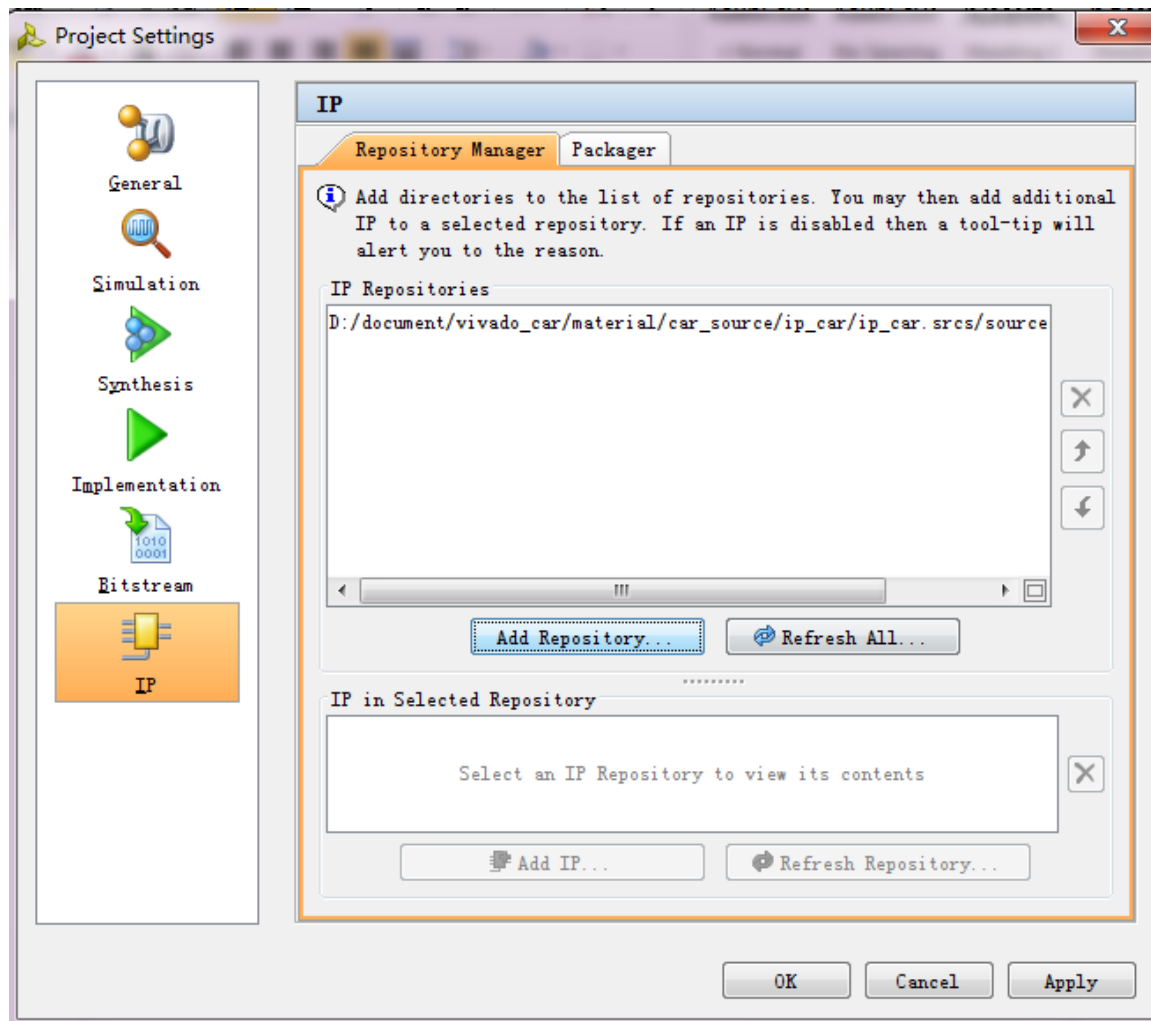


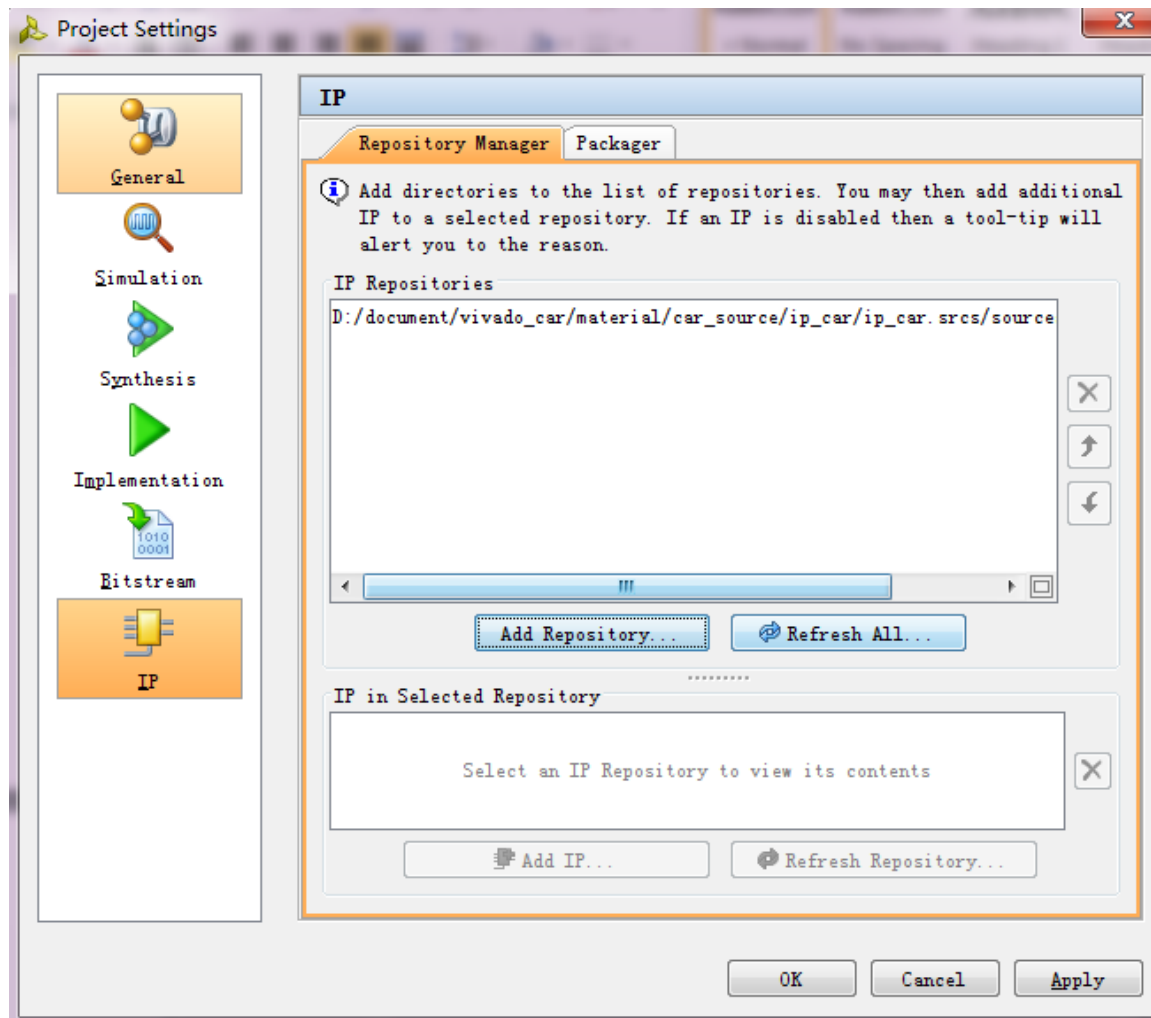
- 在右侧出现的 IP Catalog 界面中，在空白处右击出现的框图中点击选择 IP Settings...



- 在出现的界面中选择 Add Repository，之后会跳出选择路径的界面，可以自己选择，这里选择放在工程 workflow/ipcores/文件夹下，选择后点击

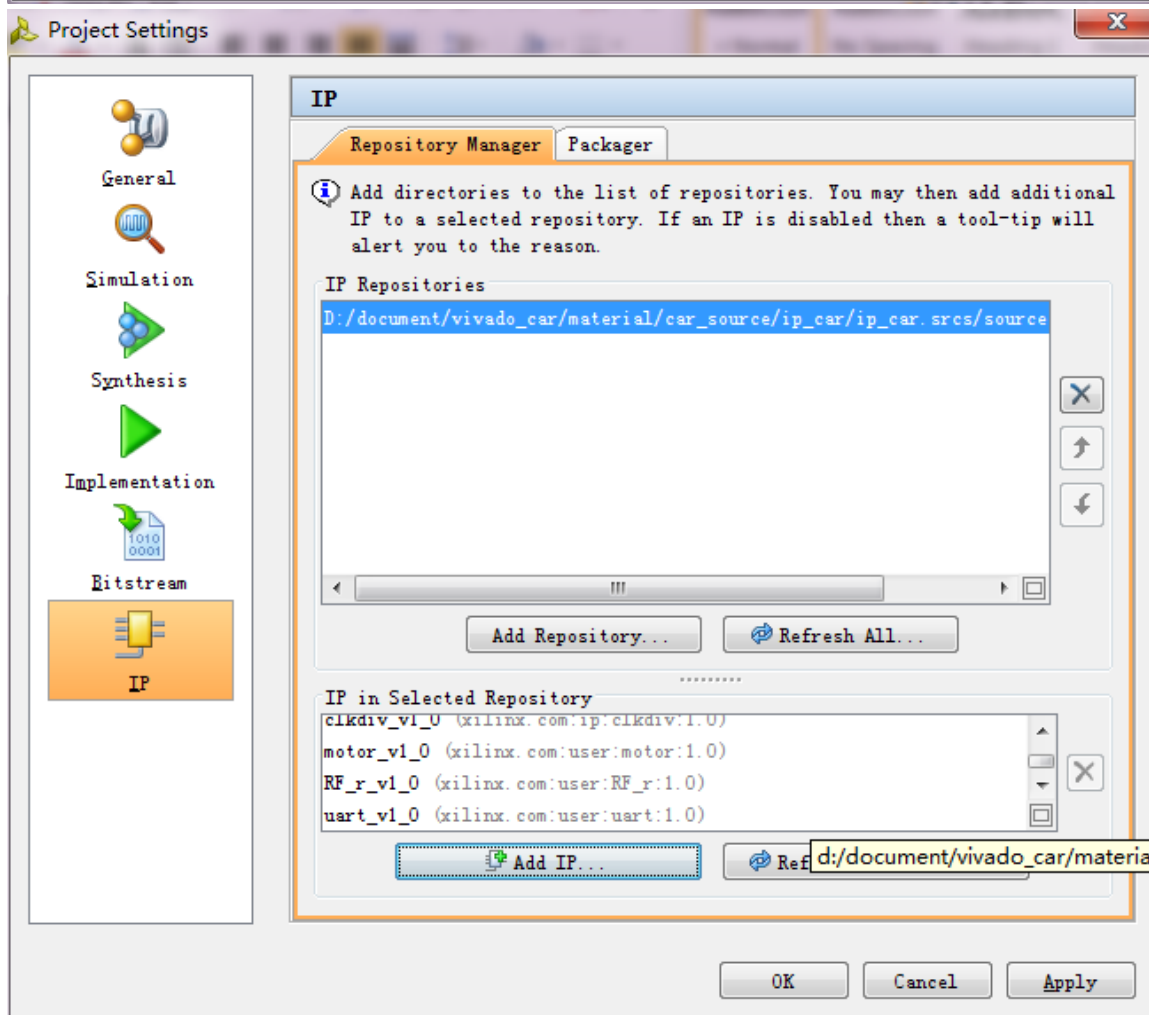
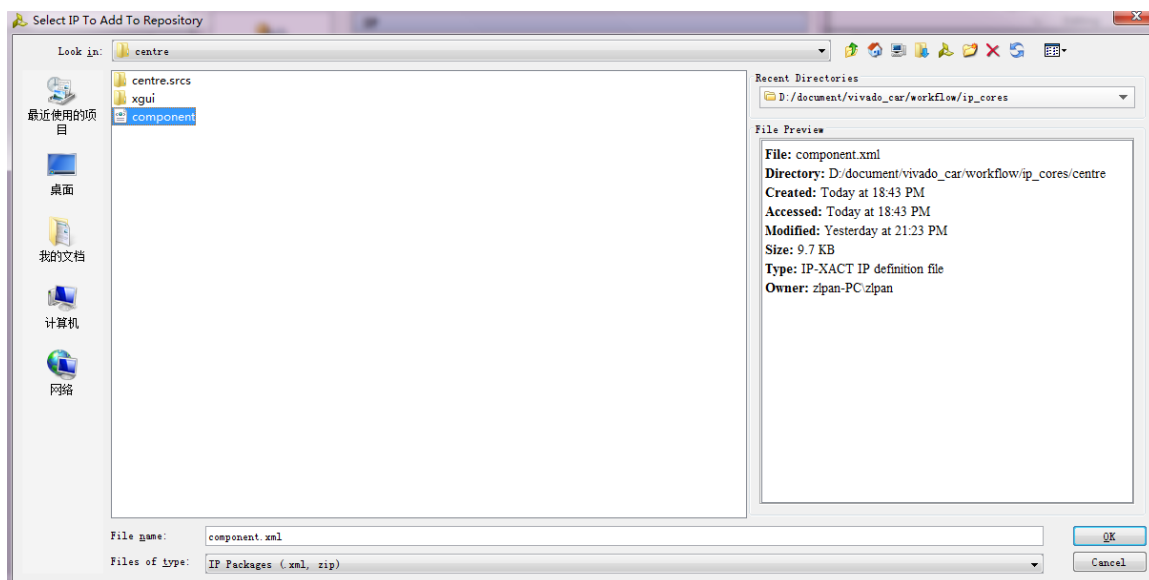
select



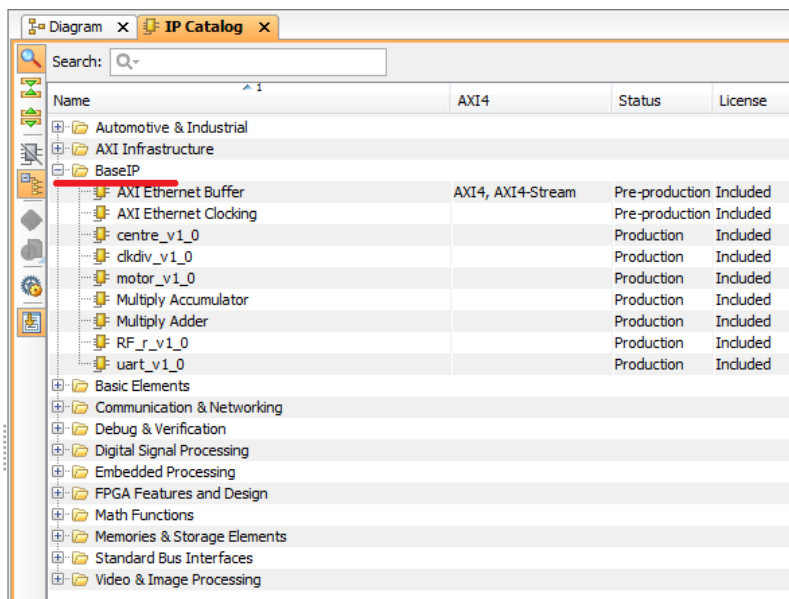


- 选择完后在 IP Repositories 框图中会出现自己选择的路径。接下来在该路径下逐一添加需要的 IP。
- 点击 Add IP，出现路径选择界面，点击默认路径栏：双击 component.xml，便可完成添加，在 IP in Selected Repository 下可以看到该 IP，用同样的方法可以完成 workflow/ipcores/ 目录下所有 IP 的添加：

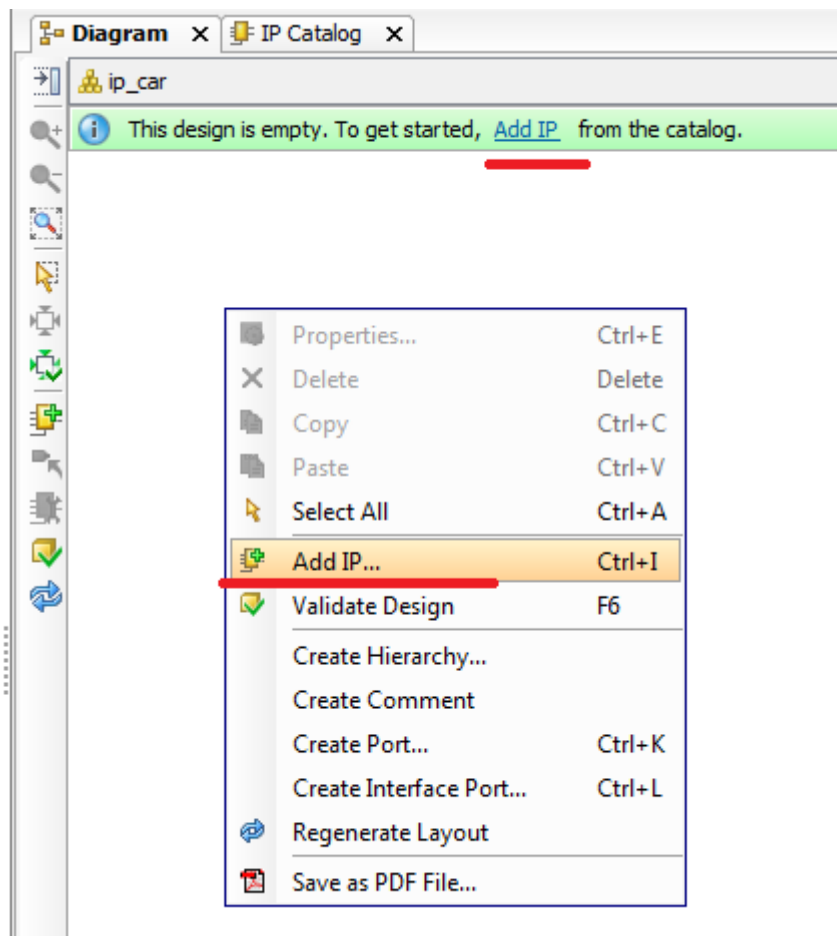




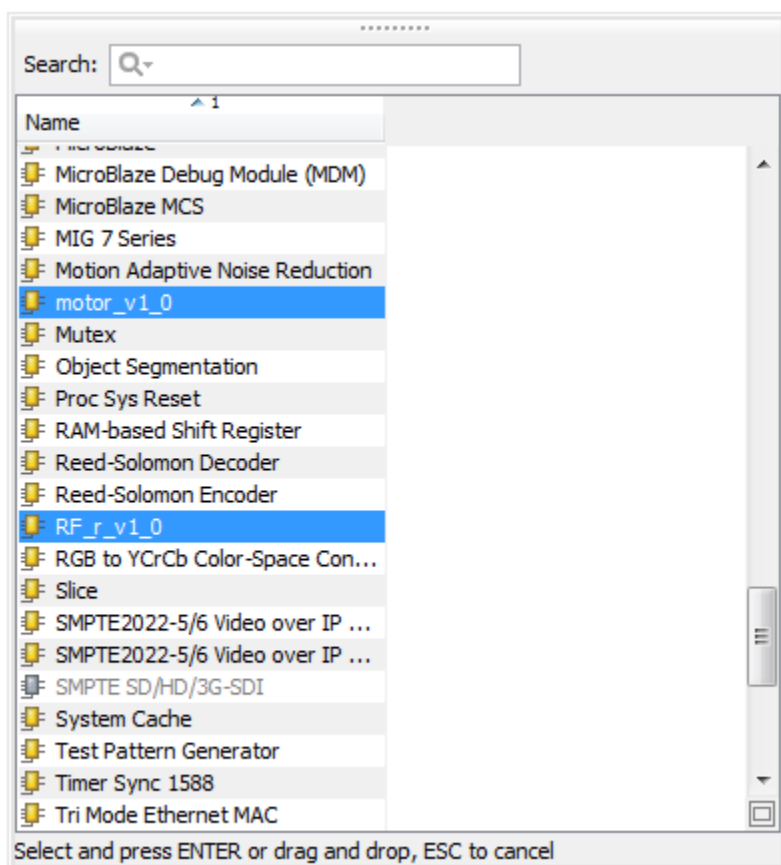
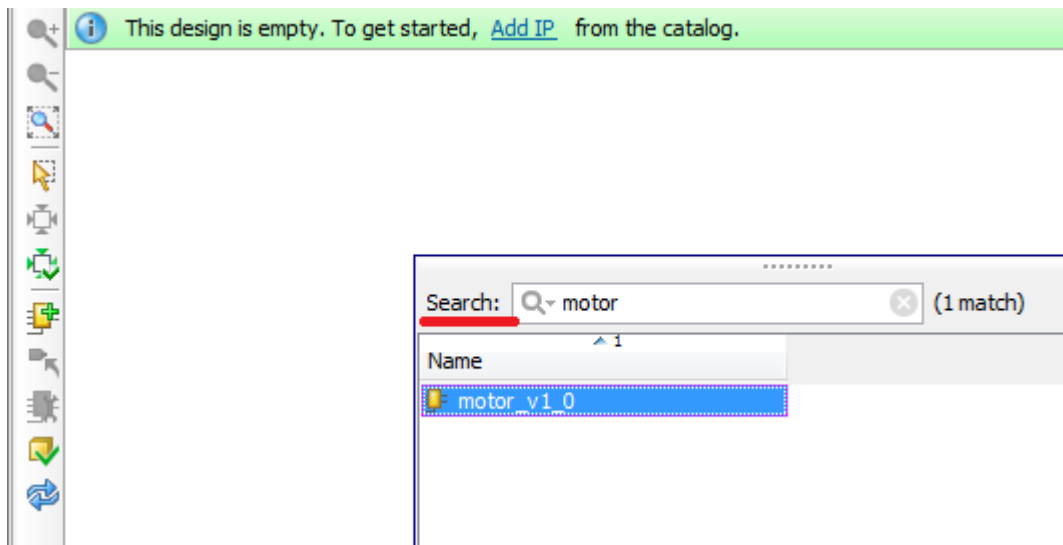
- 然后点击 OK，可以在 **BaseIP** 下看到添加的 IP，用户添加的 IP 都会出现在该目录下：



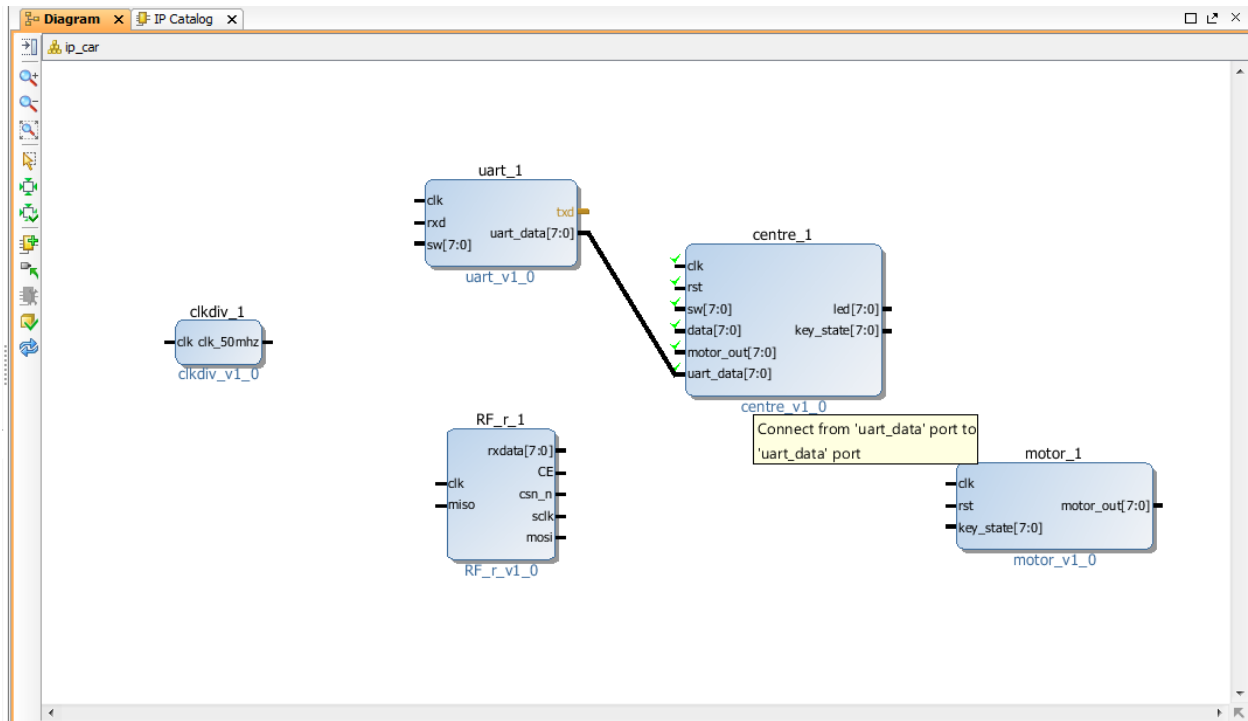
- 切换到 Diagram，右击选择 Add IP 或在上方点击 Add IP 的方式添加需要的 IP：



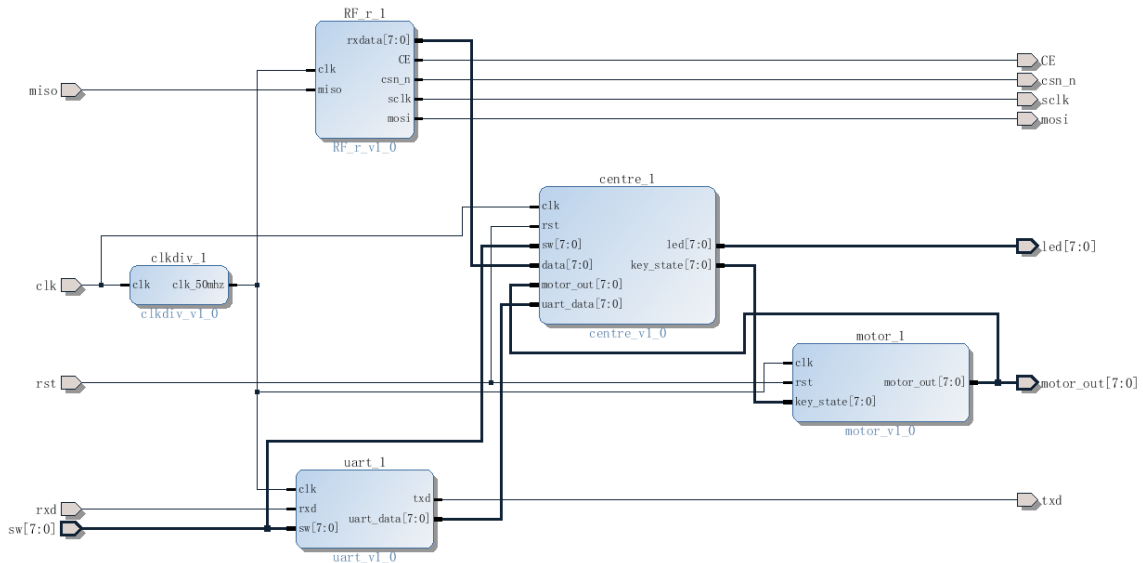
- 在 search 中输入 IP 名字，IP 显示后双击它就会被添加到面板上，或者按住 ctrl 选择完需要的 IP 后按 Enter 键：



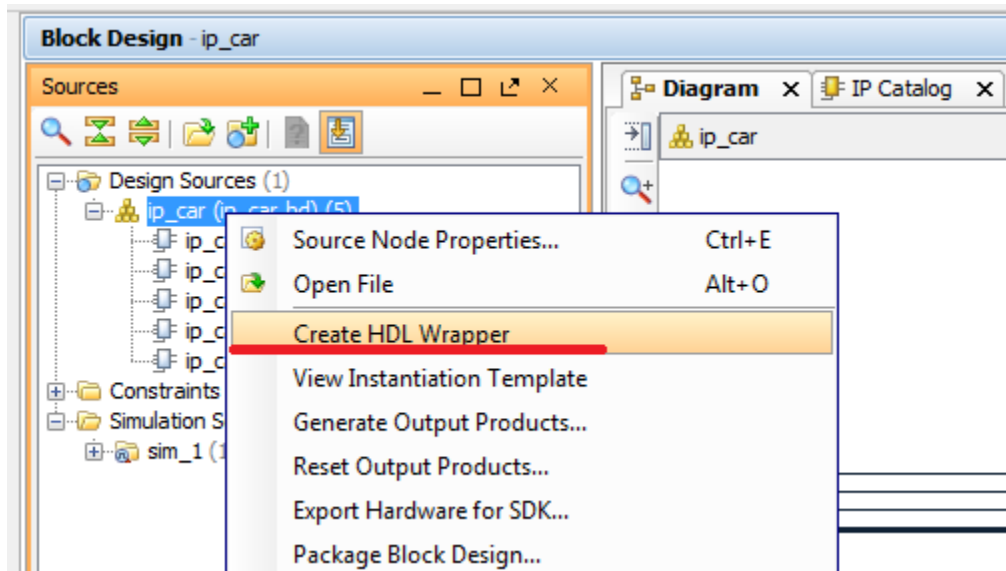
- 出现全部 IP 后连线，鼠标靠近接口直到变为笔的形状，拖到另一个需要接的口即可，取消连接右键选择 End connection Mode 即可：



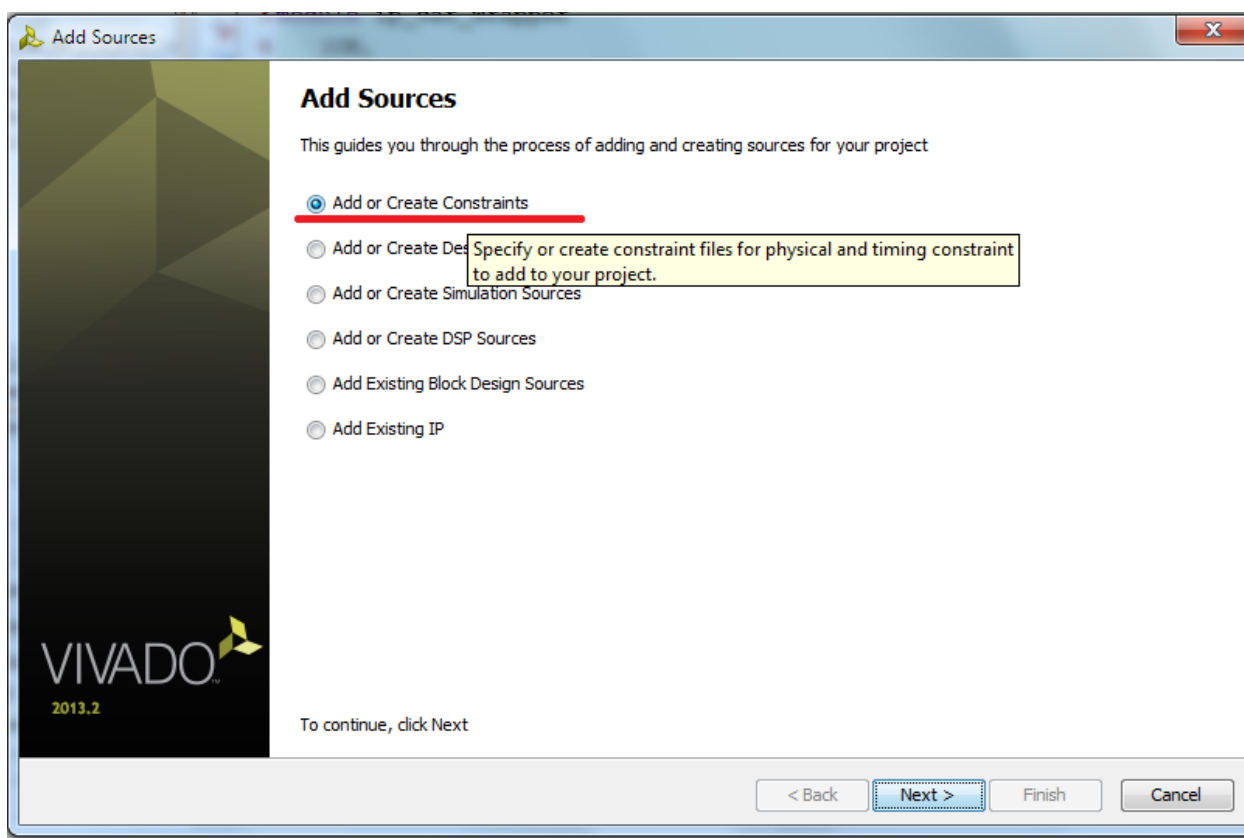
- 对于外接的引脚，鼠标靠近接口直到变为笔的形状右键选择 **Make External** 即可，以下是连好的图：



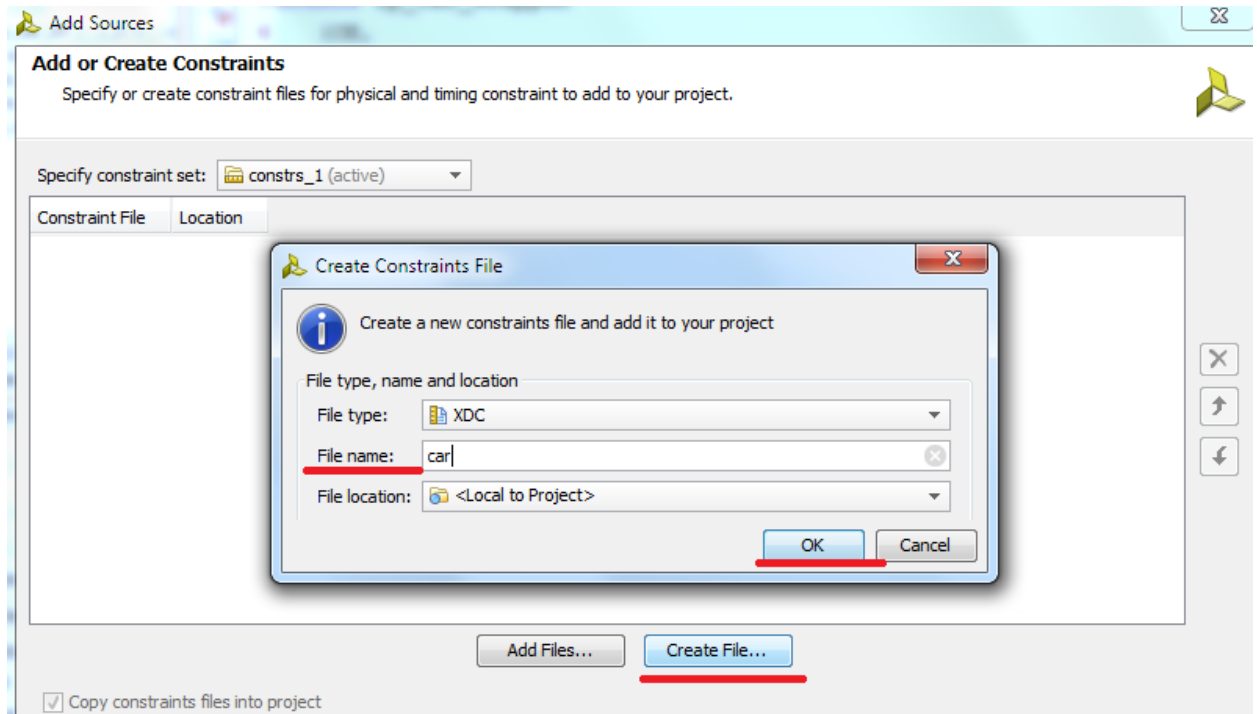
- 完成连接后，保存一下工程。右击 ip\_car 工程，点击 Create HDL Wrapper，在跳出的对话框中点击 OK，就会产生接口信息文件：



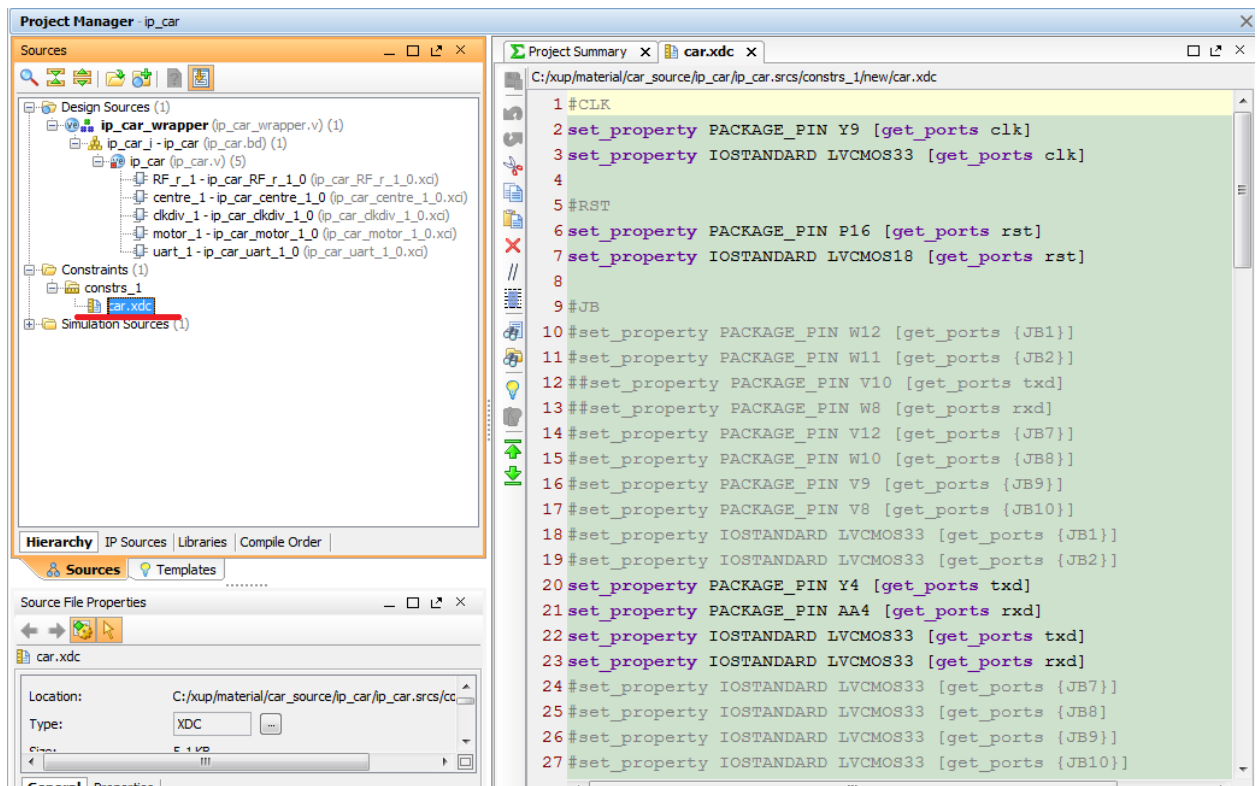
- 最后要给整个工程的外部引脚添加约束，点击 Add Sources，选择 Add or Create Constraints:



- 点击 Next，选择 Create File...，输入 File name，点击 OK，最后点击 Finish :

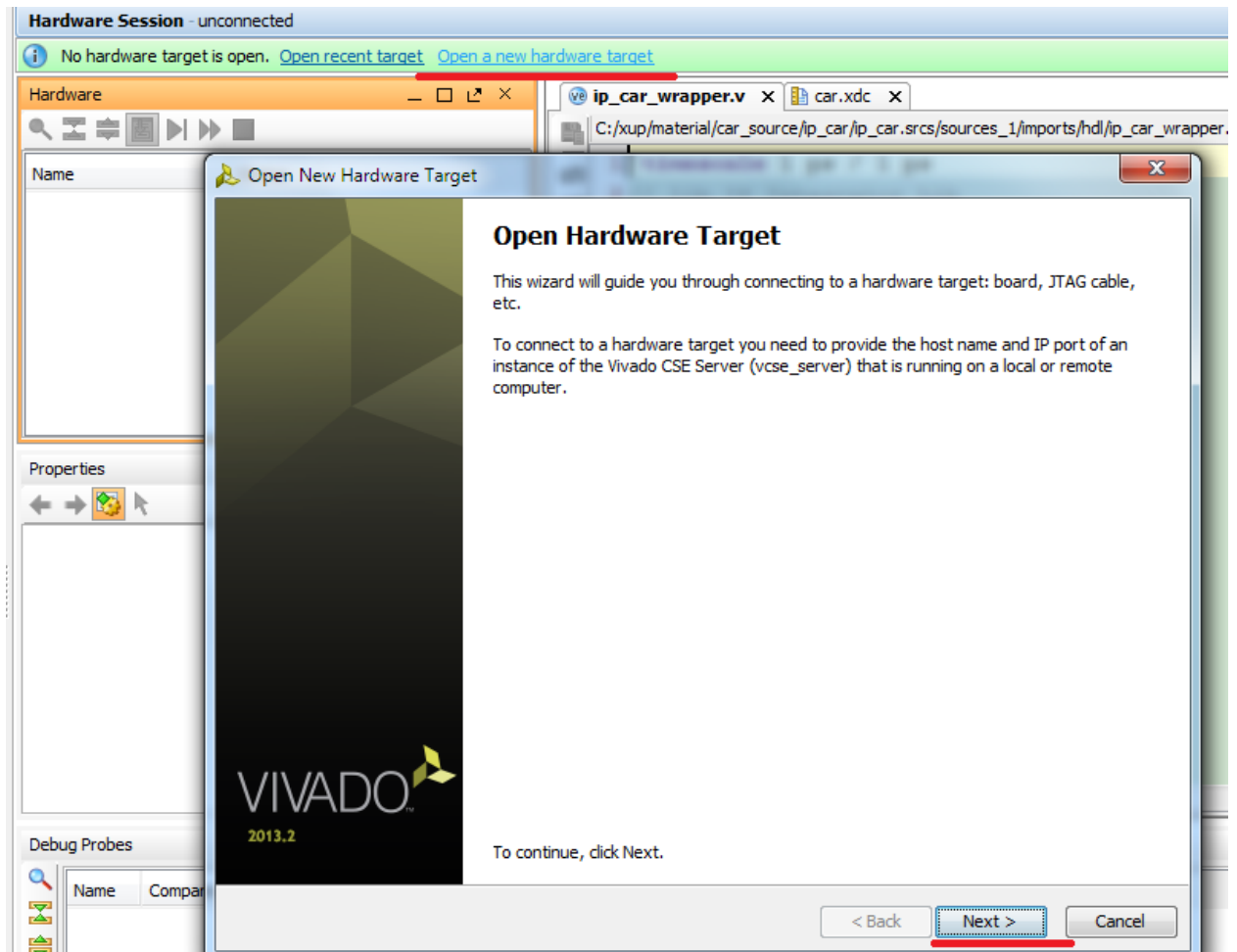


- 在 constraints 下出现自己建立的.xdc 文件，可以自己添加，也可以将 workflow\约束文件\new\car.xdc 复制过来直接使用。



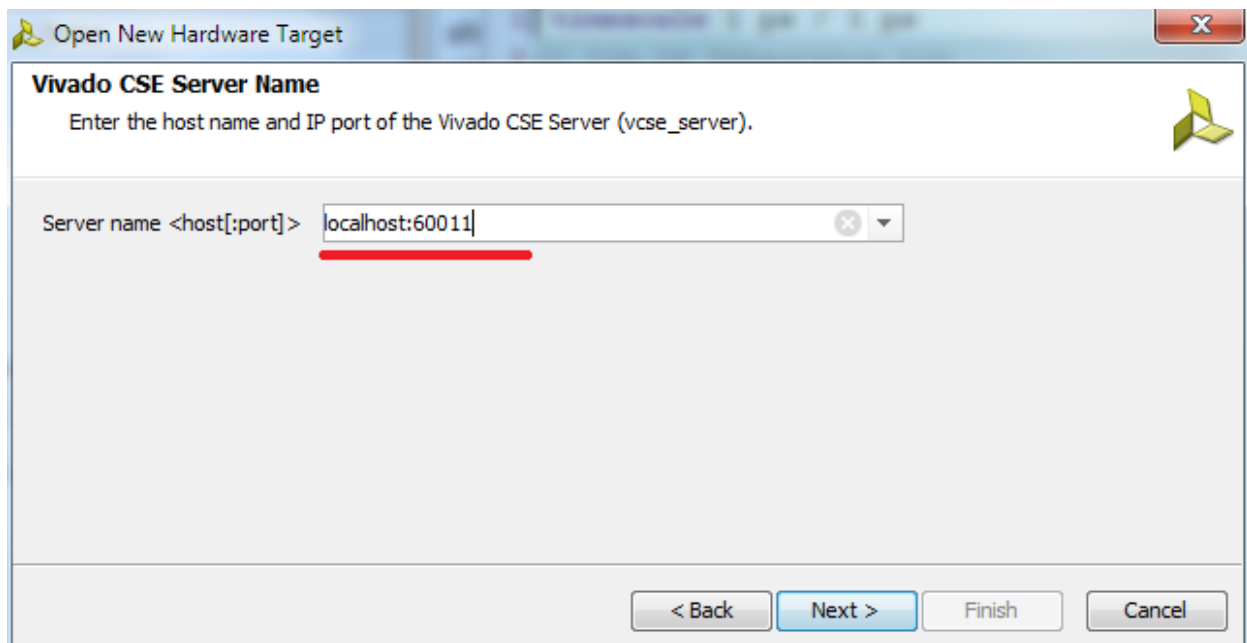
- 在 Flow Navigator 下点击 Run Synthesis，在弹出的对话框中选择 Run Implementation 并点击 OK，之后再在弹出的对话框中选择 Generate Bitstream 并点击 OK，最后在弹出的对话框中选择 Open Hardware Session 并点击 OK，将小车 zynq 的 prog 口 USB 线和电脑接好，打开电源，在工程上方点击 Open a new hardware target，在弹出界面中选择 Next。对于第一次使用的用户没有 Open recent

target 选项，之后可以选择该选项，不用另外创建：

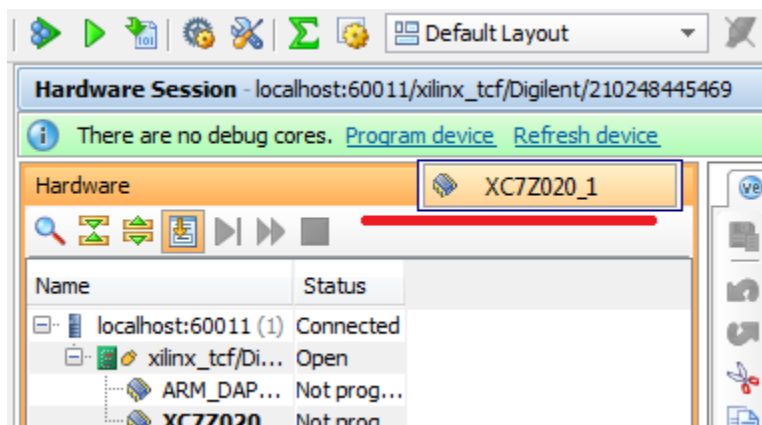




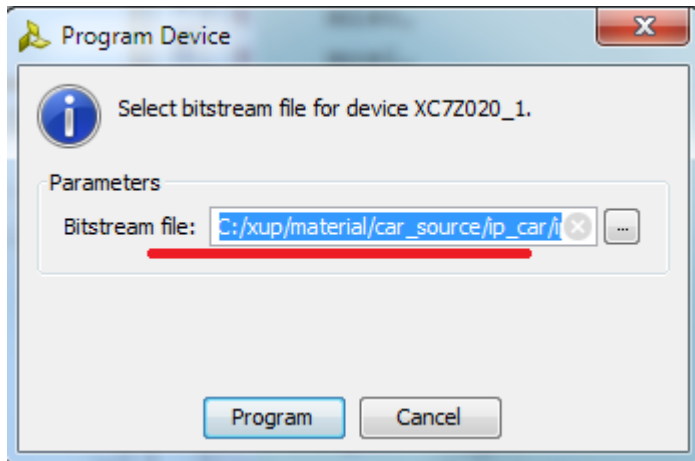
- 在下面弹出的界面中可以更改 port，这里选择 60011：



- 连续点击 Next，直到点击 Finish，之后点击 program device 并点击 xc7z020\_1:



- 确认好.bit 路径后点击 Program:



- 至此程序下载进 zynq 里了。可以使用带 RF 的控制板或带 app 的安卓手机控制该小车。