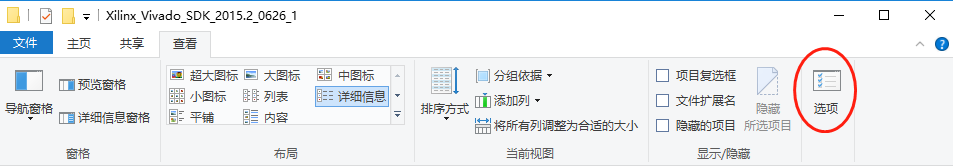
**Vivado 2015.2安装注意事项：**

首先网盘中下载Xilinx\_Vivado\_SDK\_2015.2\_0626\_1.rar和xilinx\_ise.rar

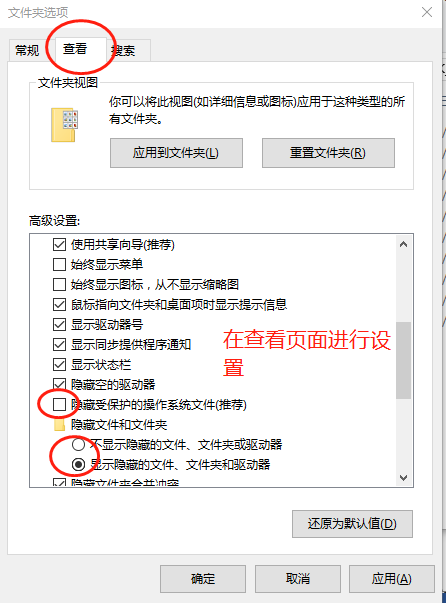
* 网盘上提供的Xilinx\_Vivado\_SDK\_2015.2\_0626\_1.rar 压缩包里面部分文件不知为何被设置成系统保护文件且被隐藏了，导致解压后找不到相应的vxsetup安装程序。

解决方法是设置显示系统保护的隐藏文件：

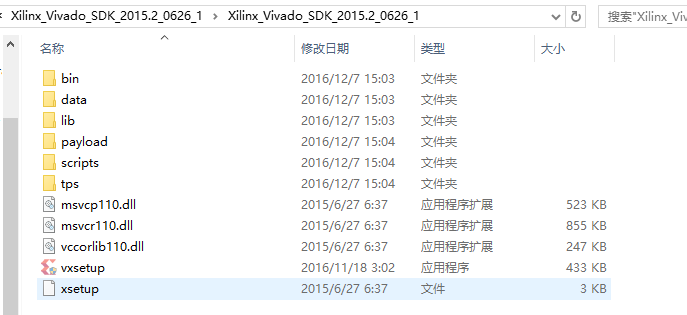
1. 打开文件夹选项



1. 在查看页面进行设置



1. 设置完后显示出安装程序vxsetup.exe：



* 如果出现双击vxsetup.exe无反应，原因是安装路径过长。。。

解决方法是将安装文件夹放在分区盘的根目录下再双击安装。

* 安装教程参考：https://blog.csdn.net/taowei1314520/article/details/74860356
* 安装过程注意事项：

1. 安装之前请关闭任何防护杀毒软件，如360。

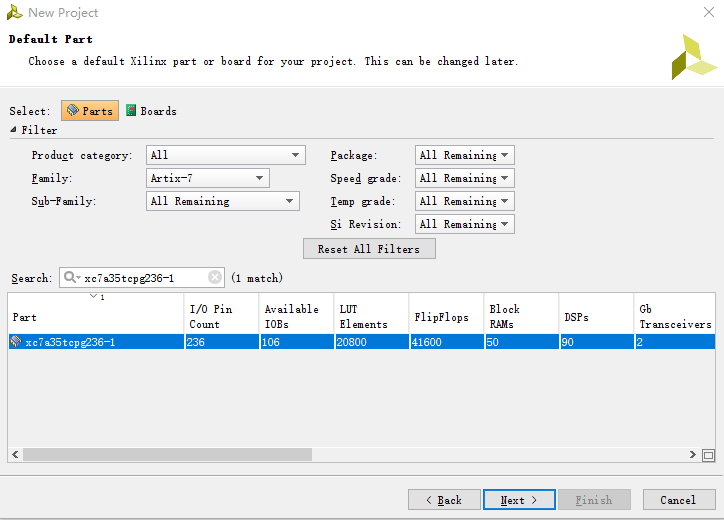
2. 请使用英文安装路径，以防出错。

3. 添加license时把xilinx\_ise.lic和Xilinx.lic都添加进去。

**使用Vivado block design的一个简单例子：**

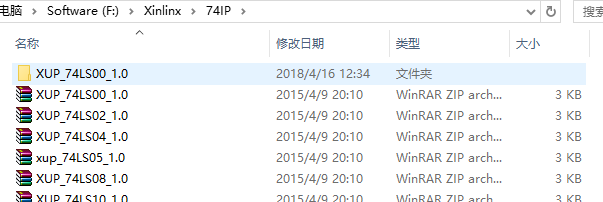
* 新建工程

参考实验指导书的实验五“新建工程部分”。最后选FPGA的芯片时候，我们选择目前使用的开发板FPGA型号：xc7a35tcpg236-1，如下图所示。

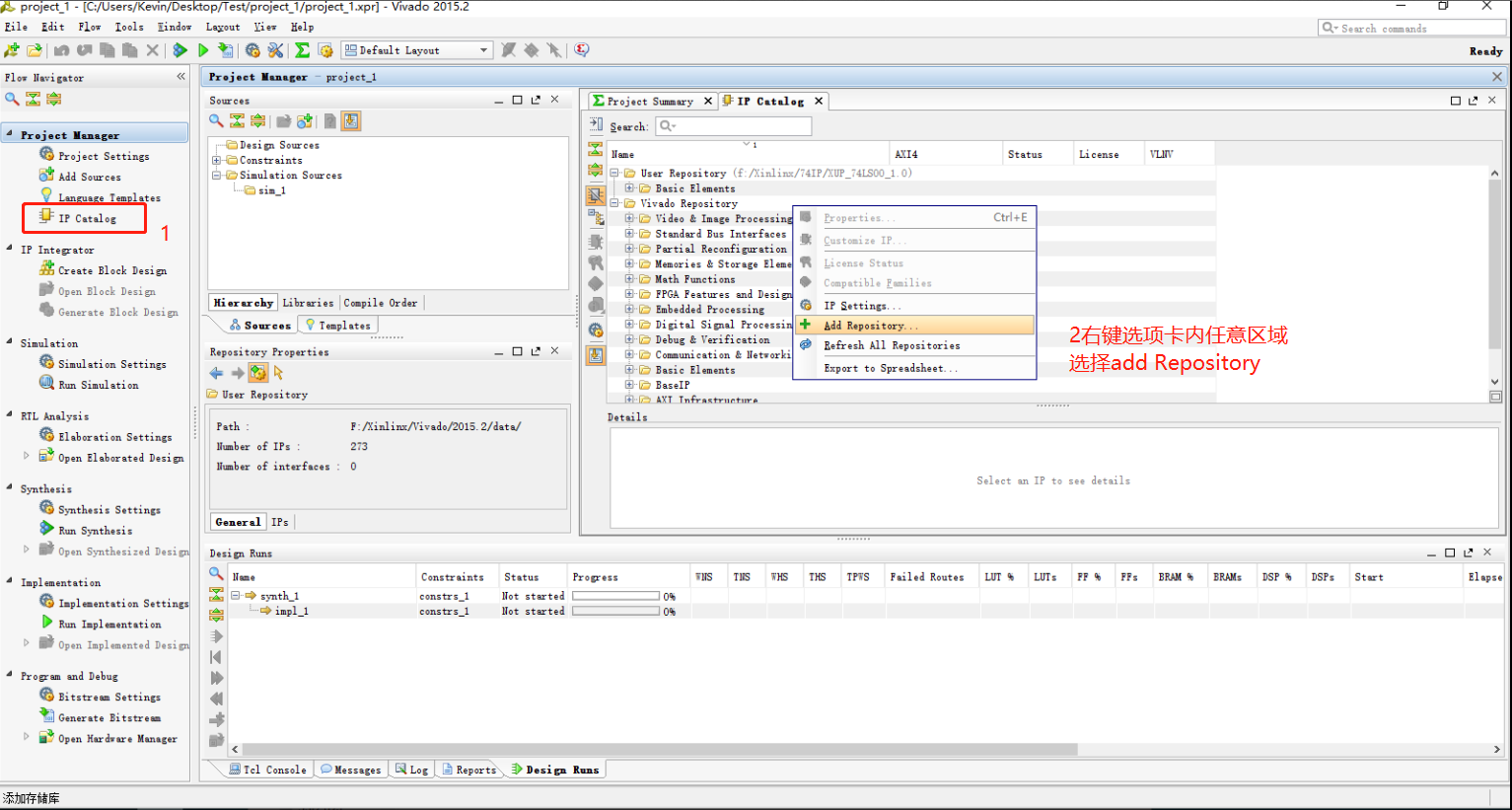


* 添加IP核

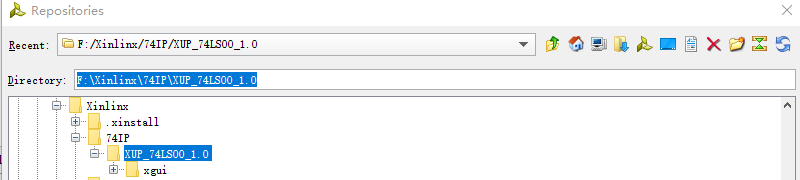
1. 将下载的74IP文件夹内的压缩包都解压出来，下图显示只解压出74LS00\_1.0。



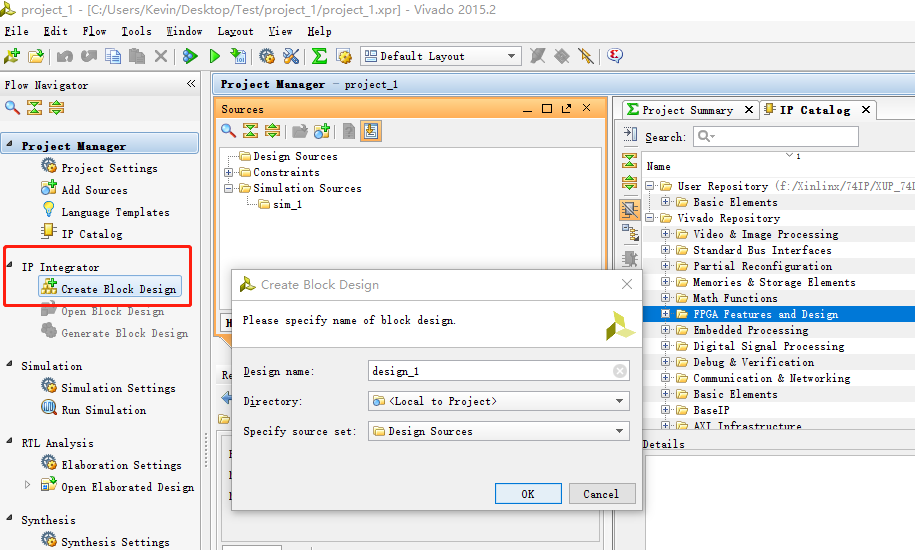
2. 在vivado主界面中加载该IP核：



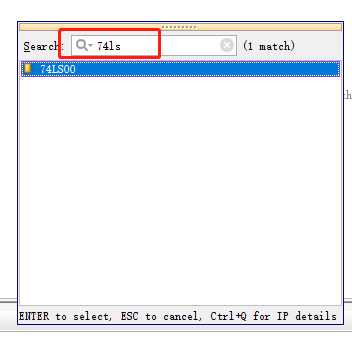
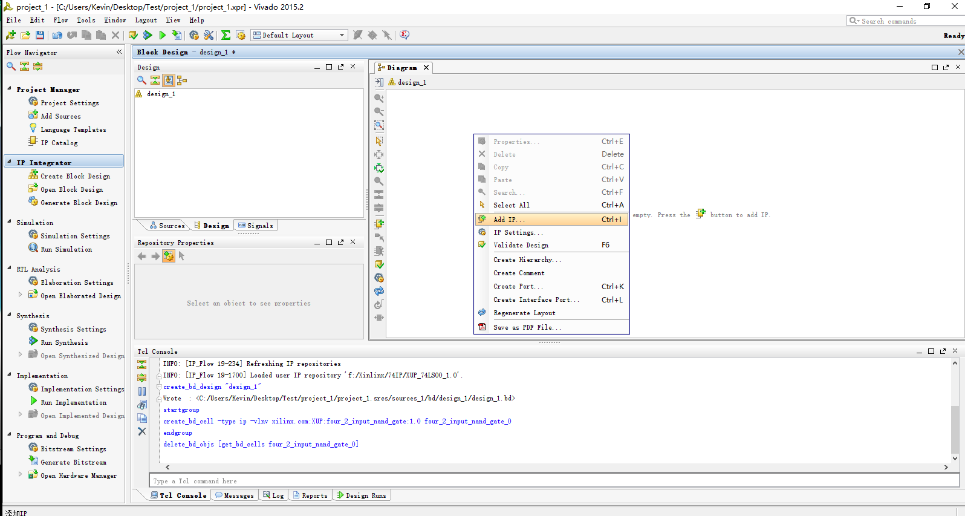
注意加载的是整个文件夹。

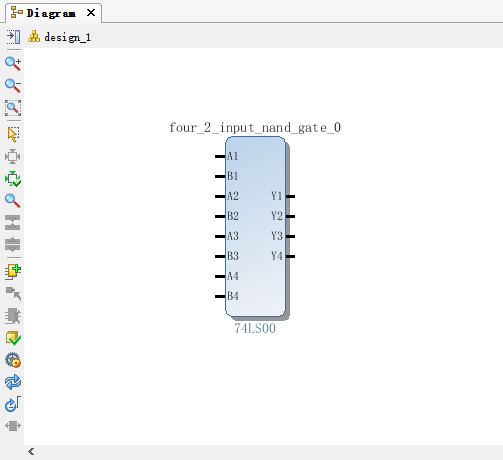


3. 新建block design文件：



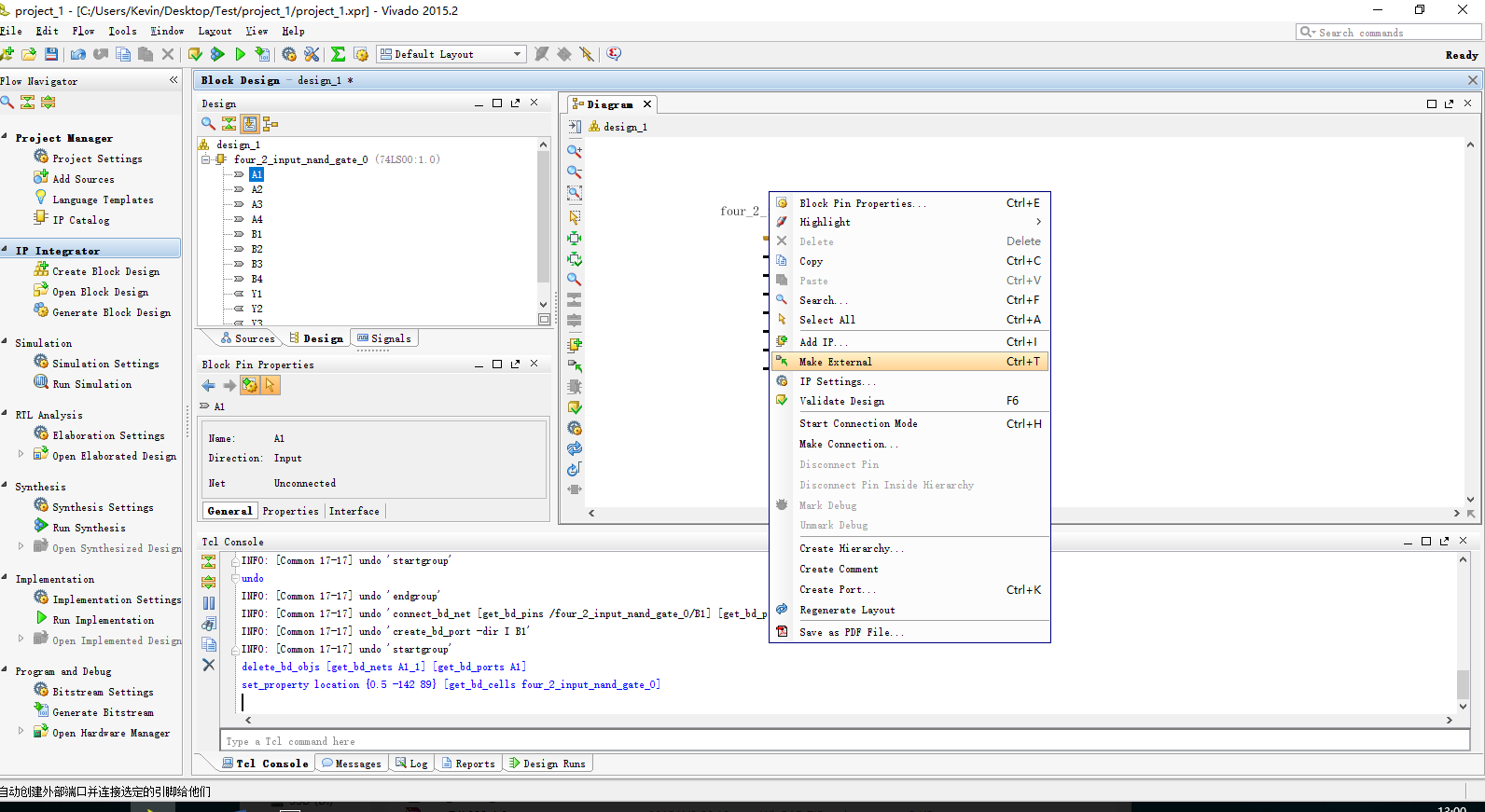
4. 在新建之后diagram框图内加入74LS00

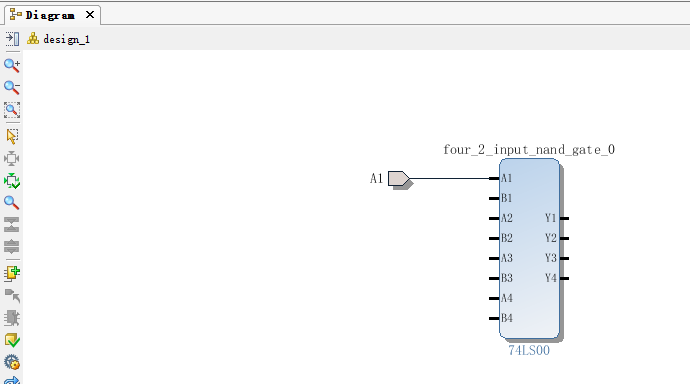
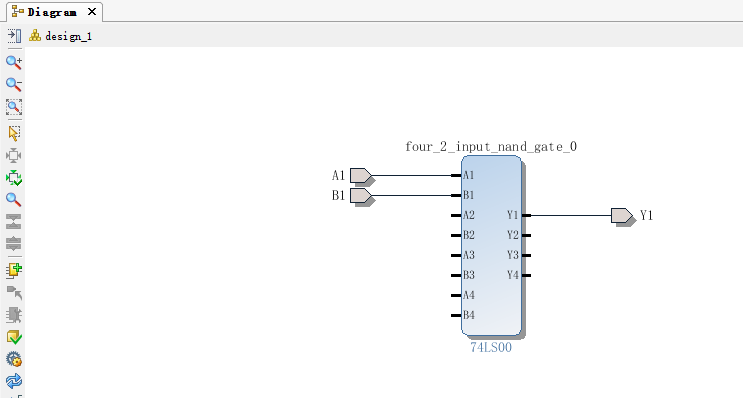


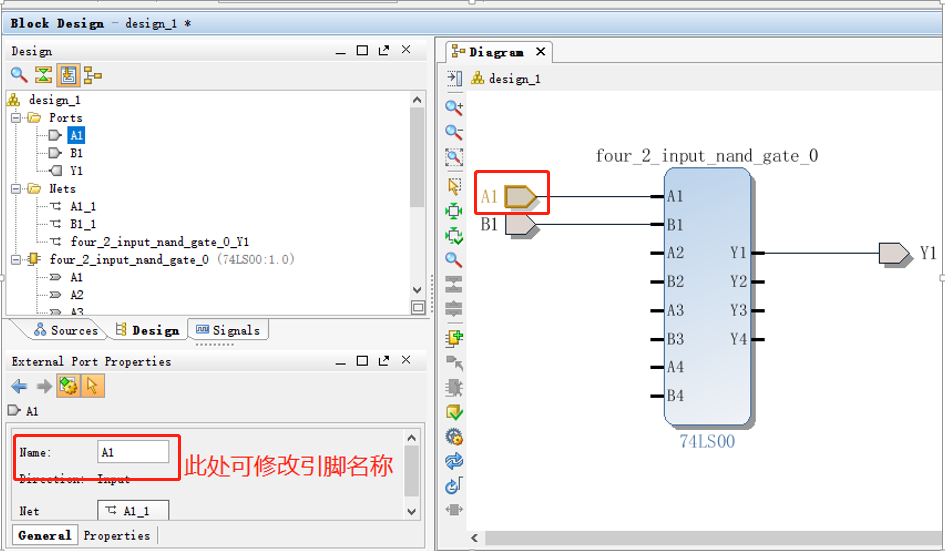


5. 添加输入输出端口及进行连线（和proteus进行仿真的思想基本相同）：本例中只以添加A1,B1,Y1为例。

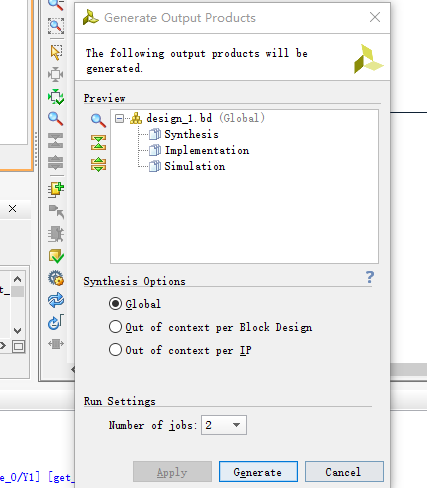
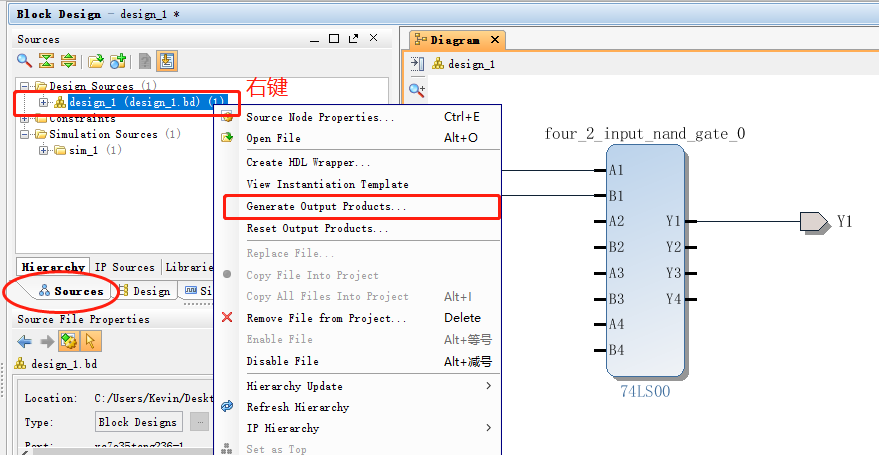
* 右键端口->选择make external。（PS: 后面所有需要进行芯片之间连线是，左键单击引脚并拖动就会出现连线）



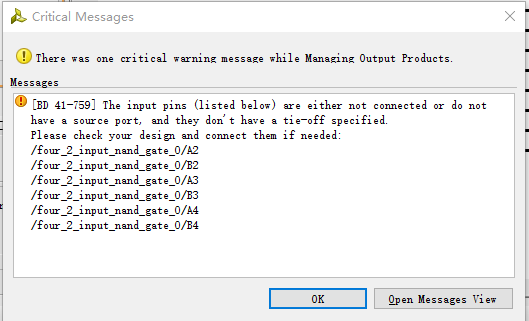
 



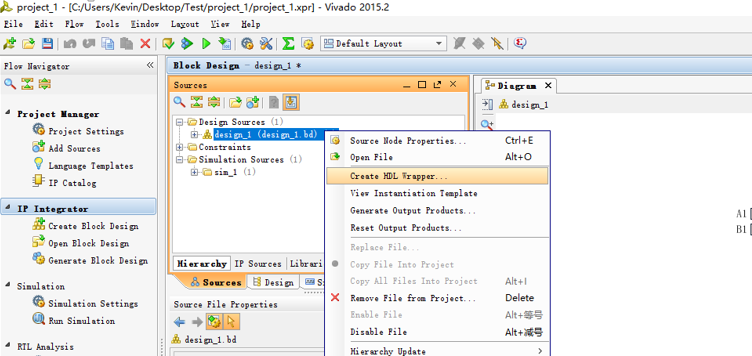
6. 完成连线和设置端口，点击source界面，右键.bd文件，选择generate Output Products=>generate生成后续需要的文件。



注意：如果design中有配置连线或端口的引脚，软件会报bug，本例中忽略。

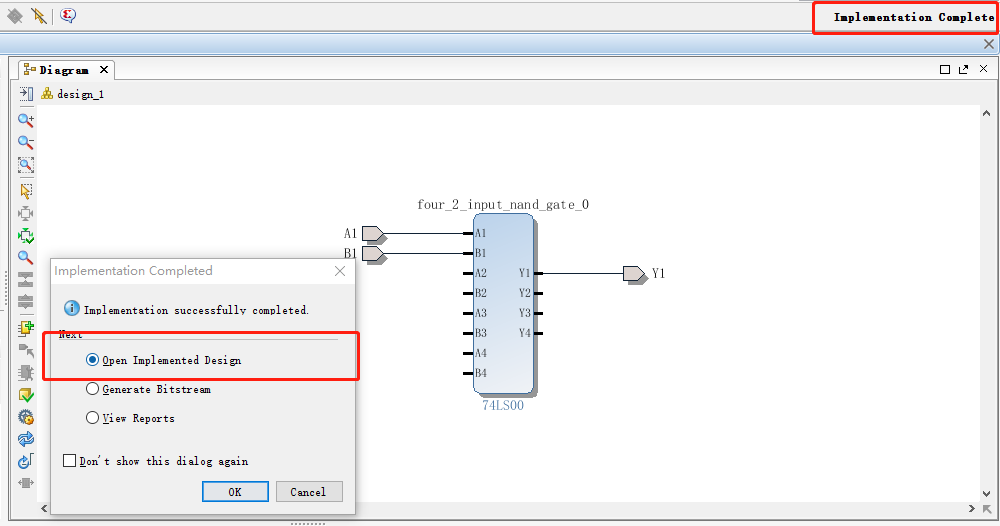


7. 再次右键.bd文件，选择Create HDL wrapper，默认选择，点OK，生成HDL的顶层封装文件。

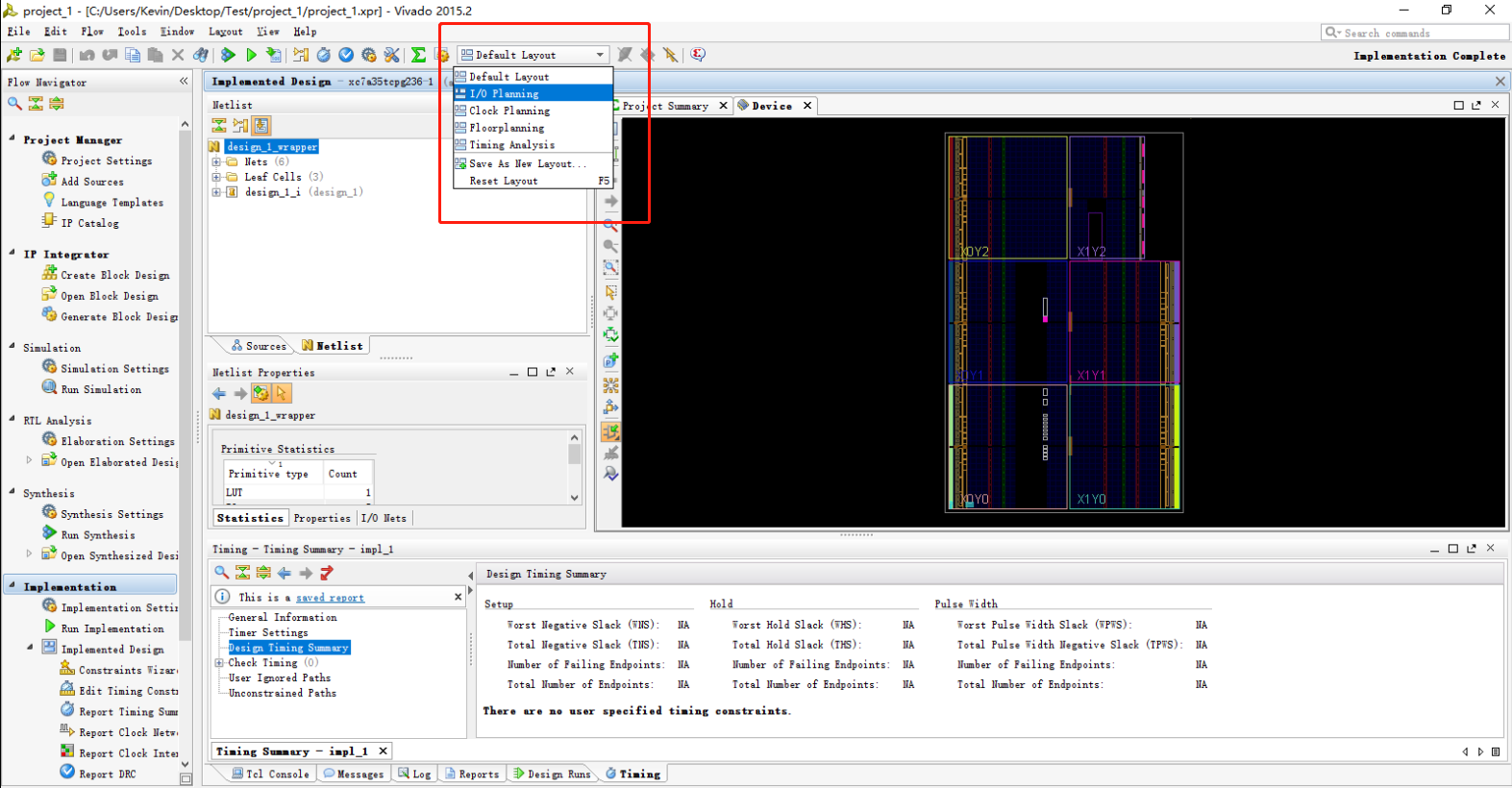


8. 生成成功后，开始按照Synthesis（综合）=>Implementation（实现）=>Program and Debug的顺序进行操作，最终把设计程序载入板中。

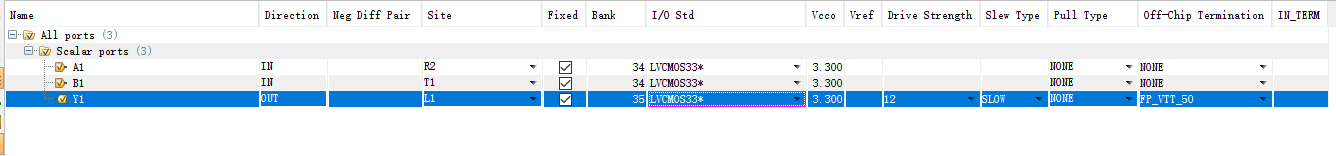
* 在完成“实现”之后，我们需要进行端口分配，即将设计中的逻辑端口映射到实际板上的物理端口：



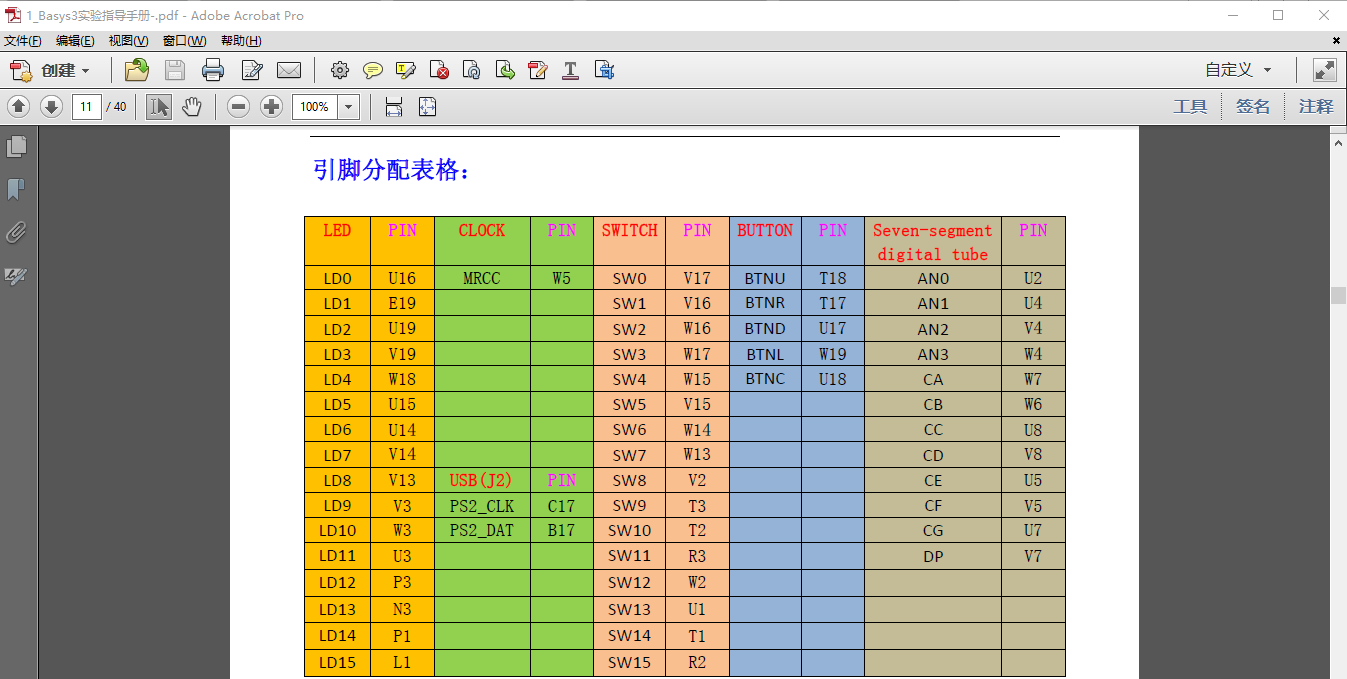
打开I/O planning界面



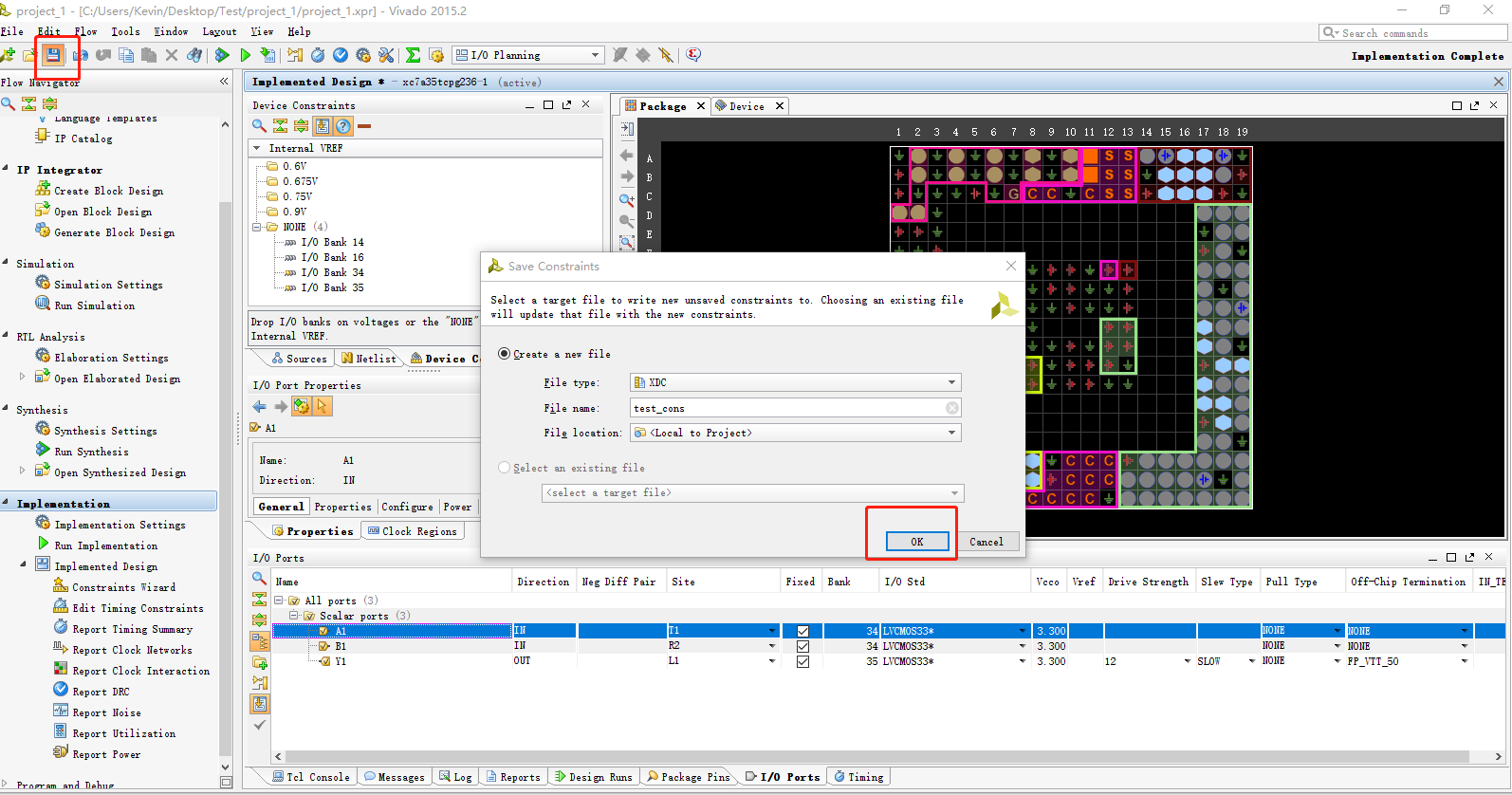
* 在下方进行IO端口配置，即block design中设计的端口具体连接板上的哪个端口。如本例中:
  + 我们将A1,B1映射为拨码开关SW14（T1）和SW15(R2)，作为输入
  + 将Y1映射到LD15（L1）作为输出
  + IO的模式设为LVCMOS3.3（即IO的输入输出电压是3.3V）
  + 配置如下：



**PS：**映射索引时使用的是物理端口的**引脚编号**，它们由板子生产商提供，软件根据选定的芯片型号进行索引。我们使用的这块板的**引脚分配表**可请参见资料“Basys3实验指导手册”。**另外也可见板子PCB上面的对应器件的下的丝印**。

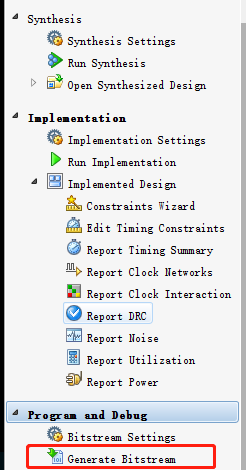
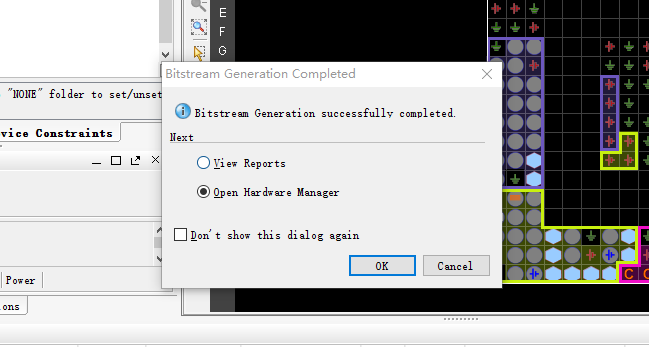


* 配置好端口之后保存约束文件，此时会提醒之前的综合和实现会过期，后面要生成load入板中的.bit文件之前要重新跑一次。

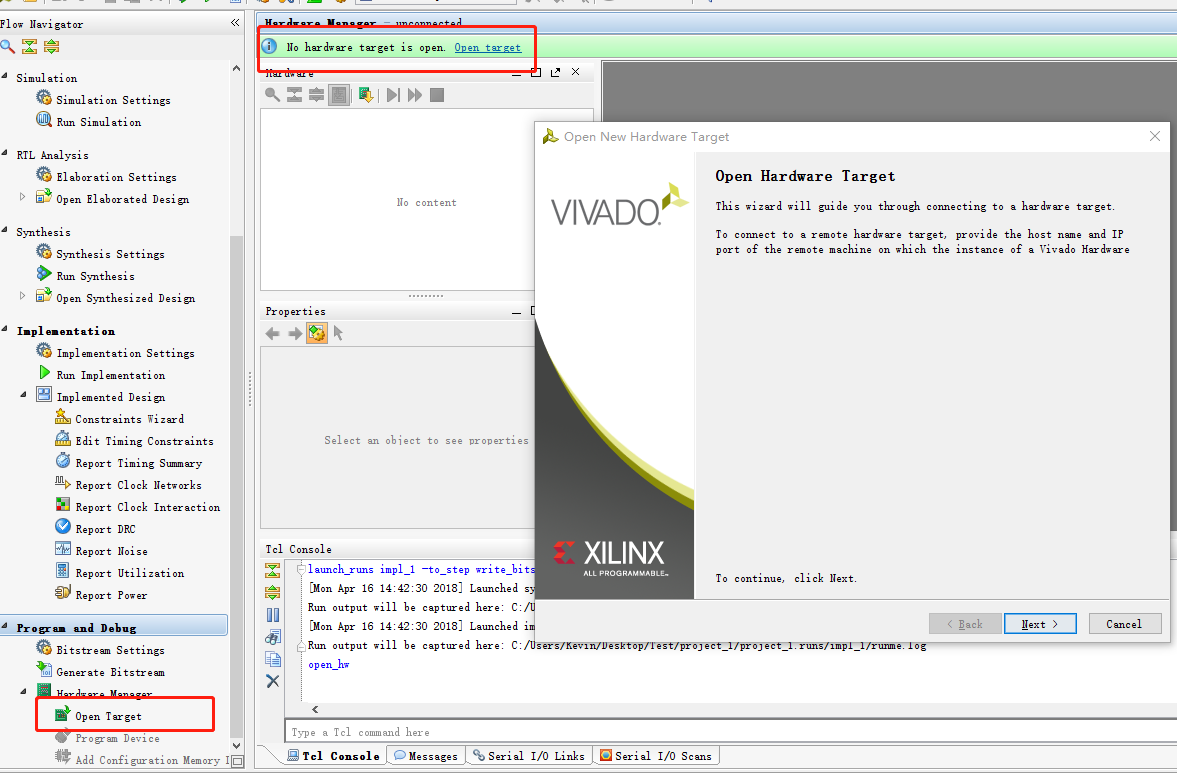


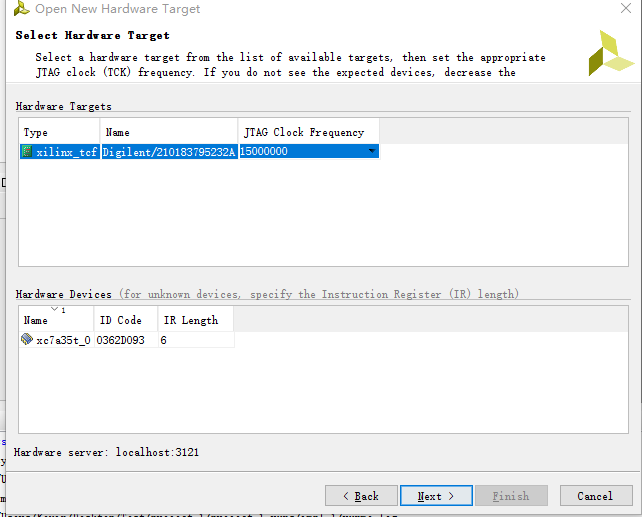
9. 生成.bit文件

* 双击generate bitstream, 重新跑综合和实现，完成之后，选择打开open hardware manager。

 => 

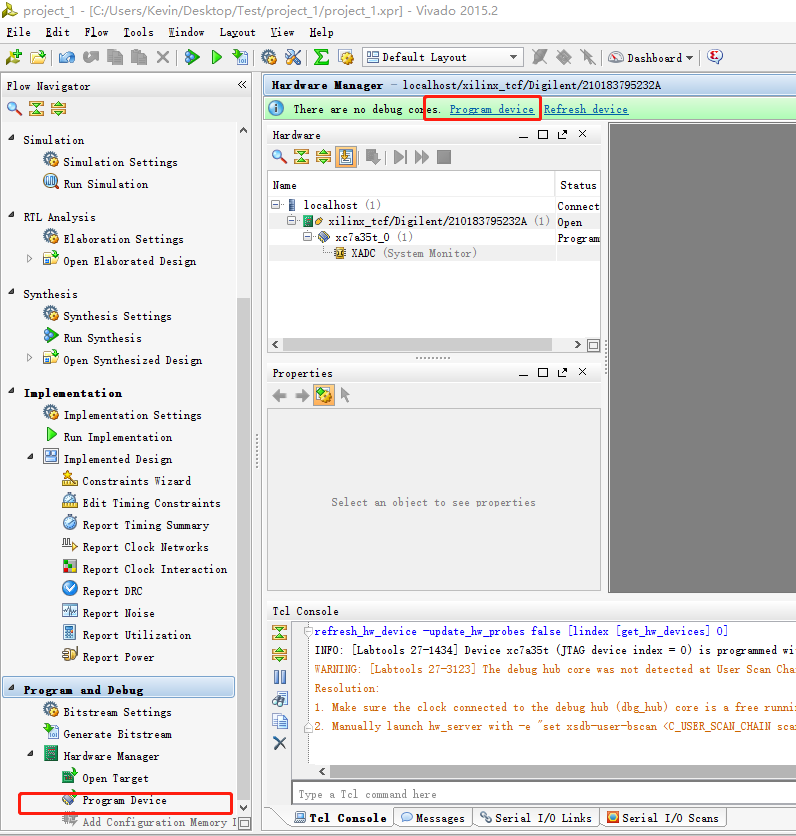
* 选择open target=>next，选择connect to local server =>next，成功发现我们的板子=>next=>finish.





10. 将.bit文件load入板中进行调试

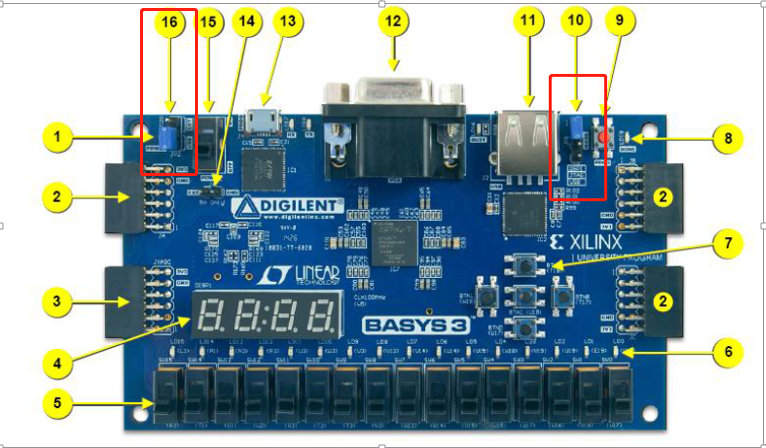
* 选择program device->板子型号=>program，在板上对设计进行初步验证。



**烧录板子之前注意事项：**

1. 将basys3板卡配置模式选择**JTAG模式**（貌似QSPI模式也行），由板上**跳线帽10**决定。

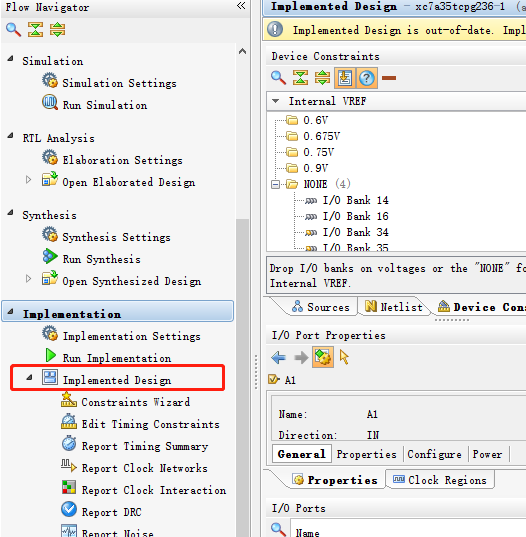
2. 供电来源选择**USB供电**（由**跳线帽16**决定），如果选择EXT供电，打开电源无反应。



11. 以上过程烧进fpga里面的程序在**掉电会丢失**。在我们调试成功之后，**将程序烧到板上的ROM中，这样程序就能掉电不丢失**。

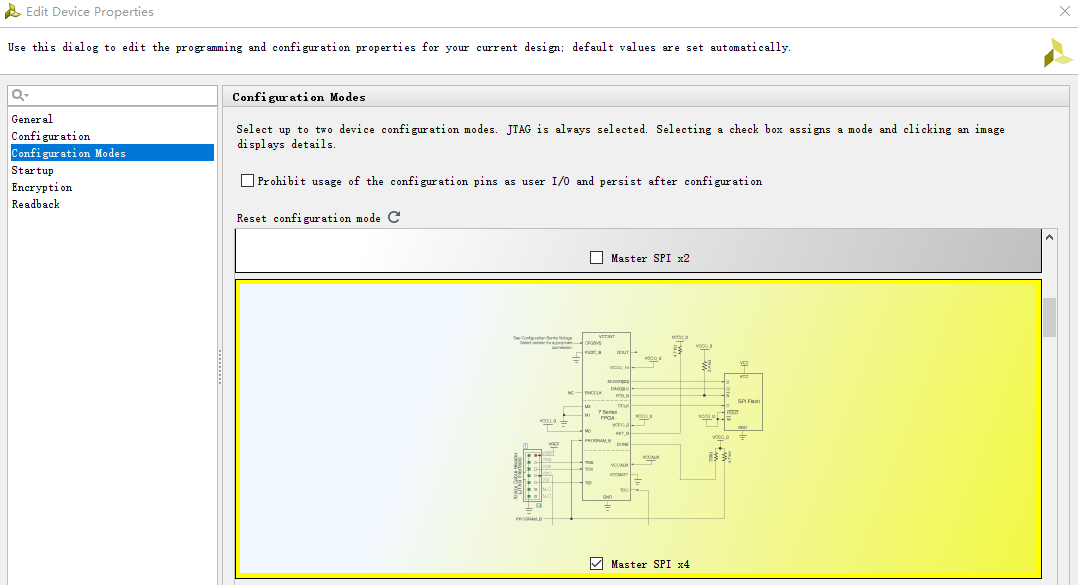
Note: 由于该过程时间比较长，随着后面设计越来越复杂，烧录一次时间可能要半小时甚至以上，因此建议在调试成功之后再将程序烧到ROM中。

* 首先在Flow Navigator中选择打开Implement design。



* 上方打开Tools->Edit Device Properties->Configuration Modes:

原默认选择JTAG/Boundary Scan模式，多选择一个Master SPI x4模式。



* 剩下过程参见“**basys3实验指导手册**”P35-37内容。