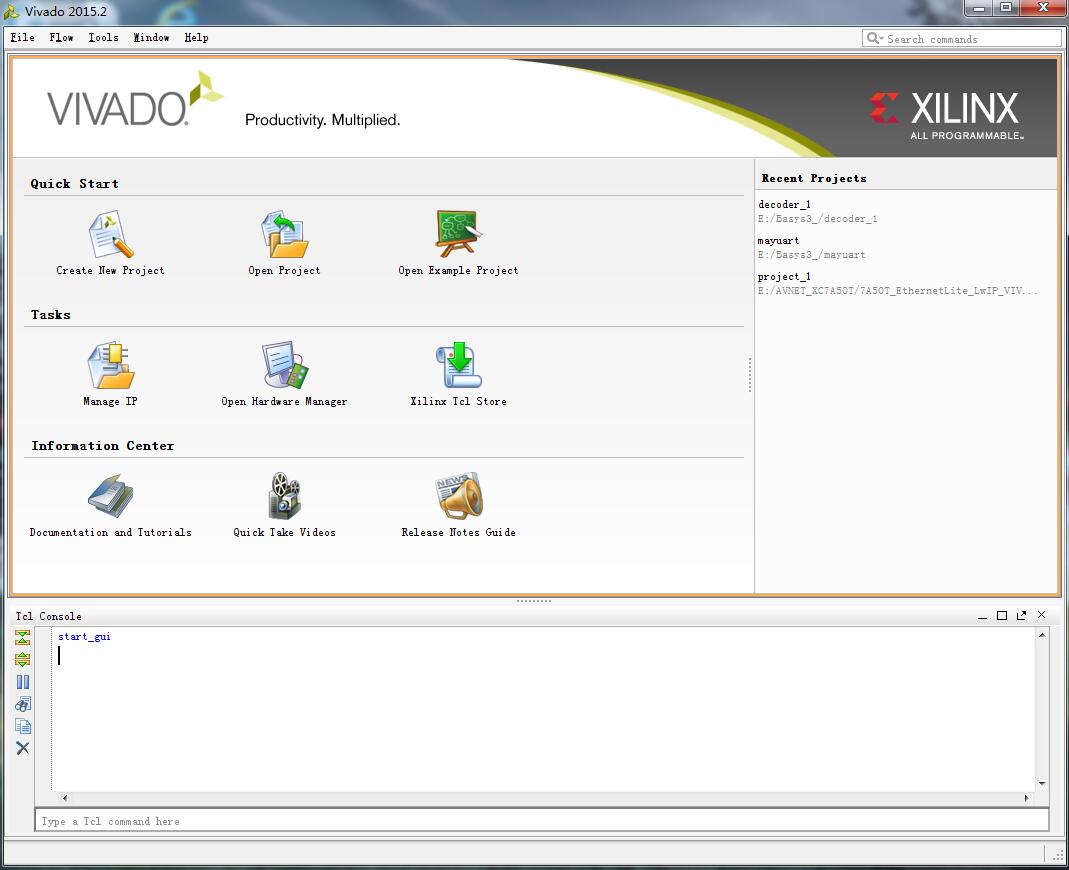
* **Basys3开发板指导手册**

**——通过Vivado 2015.1 设计译码器**

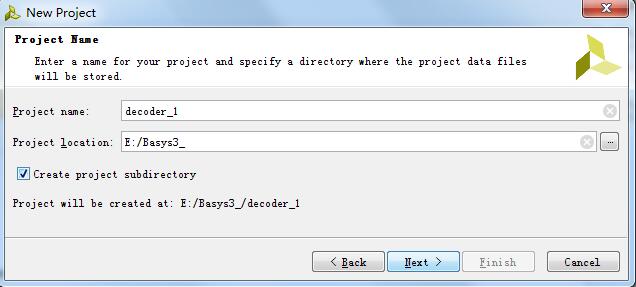
  这个手册将展示 如何在Basys3开发板上创建一个3-8译码器（通过拨码开关和led控制显示）。

  启动Vivado设计软件：

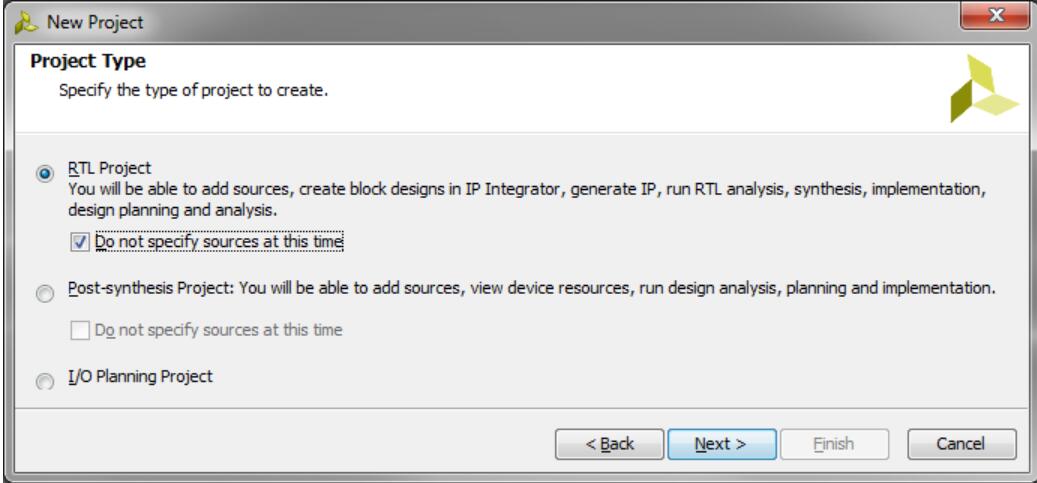


  择创建新的工程。

  点击Next，输入工程文件名和文件地址。

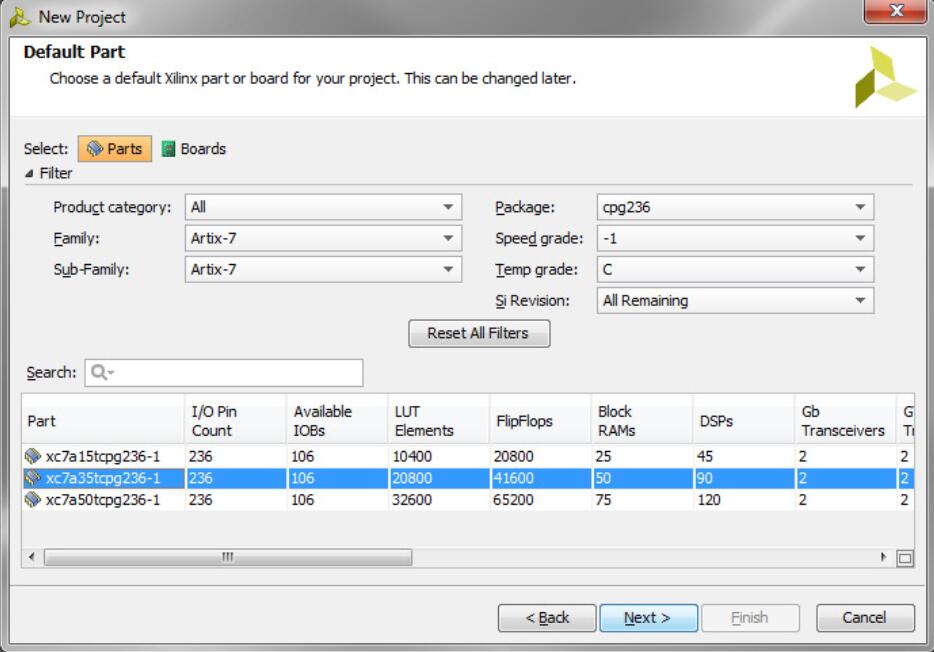


  点击Next，选择RTL工程文件类型。

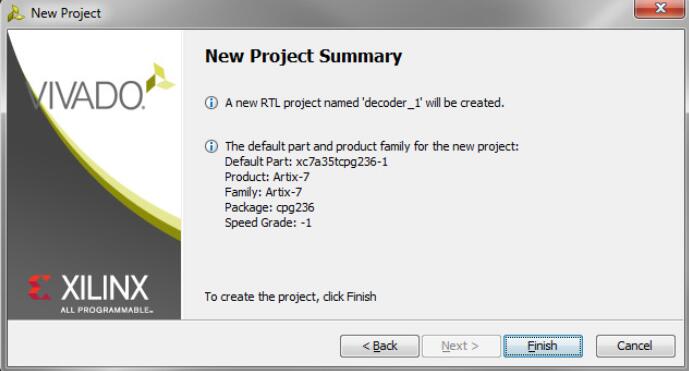


  勾选“Do not specify sources at this time“，点击next。（此时不需要指定源文件）

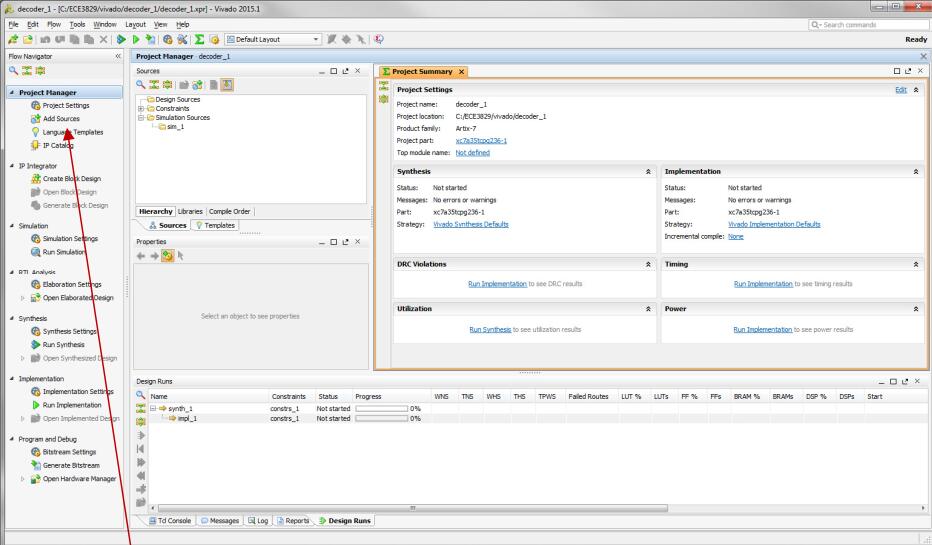
  选择正确的Basys3开发板上正确的Xilinx FPGA型号（XC7A35T-1CPG236C）。



  点击 Next，接着点Finish完成工程创建。

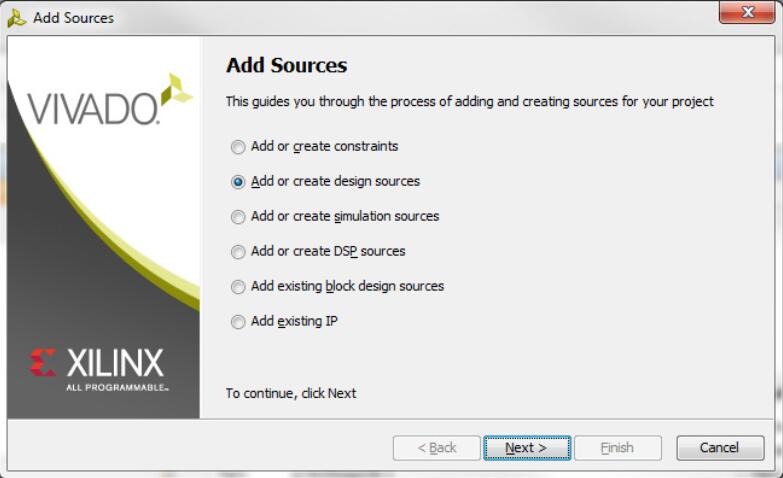


  工程文件打开窗口如下所示：



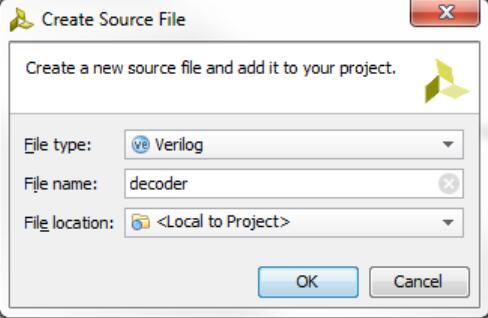
  我们现在需要添加一个Verilog文件描述译码器如何实现。

  点击左侧Project Manager 窗口中的Add Sources按钮（或点击菜单栏中File=>Add Sources）:



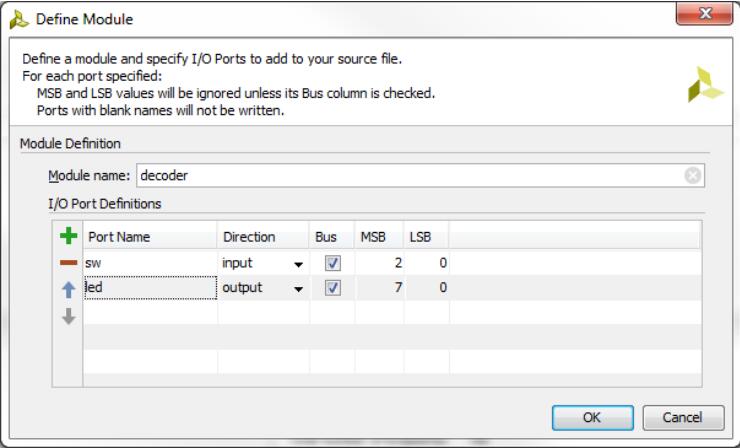
 选Next。

 点击左侧“+ ”按钮选择创建新文件，同时输入文件名decoder。



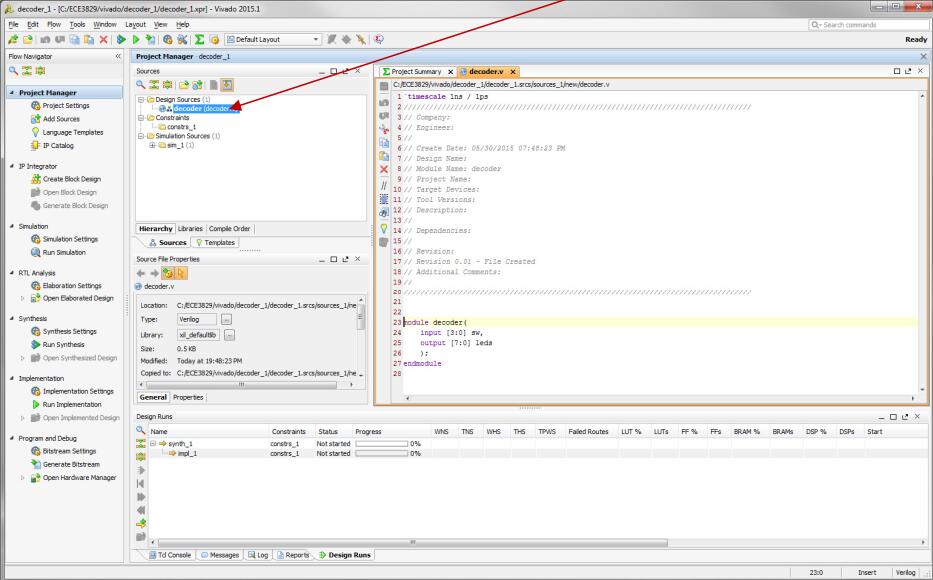
  接着点击OK完成新的源文件创建。

  这个时候我们可以为3-8译码器指定输入输出信号（我们使用3个拨码开关和8个led灯）。



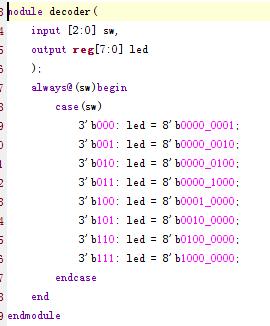
  点击OK。

  回到工程文件管理窗口（Project Manager），双击新建的decoder.v文件，你会在右侧看到Verilog文件。



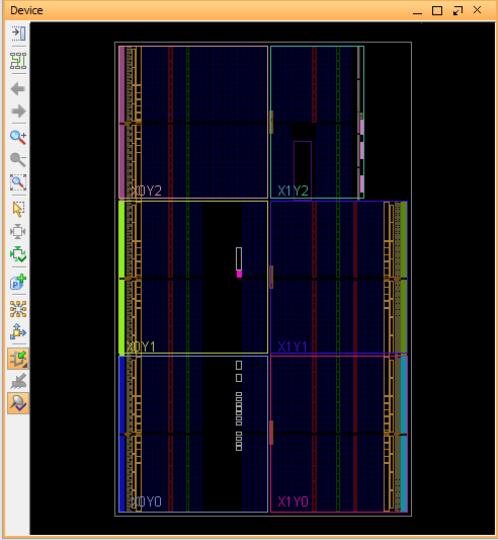
  你可以在Verilog的开头注释中添加你的项目信息。

  现在，我们可以添加Verilog语句去设计我们的3-8译码器。这里有许多方法可以设计这个译码器，下面是一个使用case语句的实例。

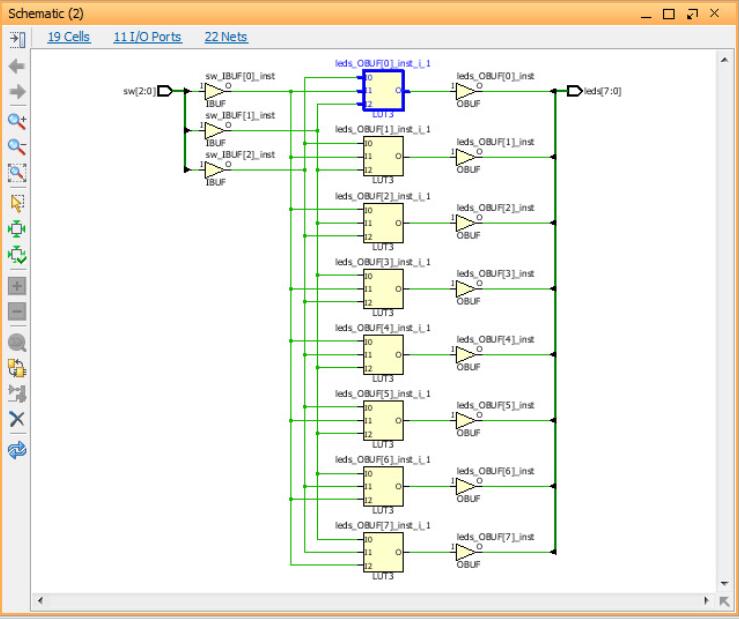


  现在我们可以综合这个设计文件。

  点击Run Synthesis。综合完成之后 应该没有错误和警告。如果你打开综合设计，你可以看到器件资源使用描述。



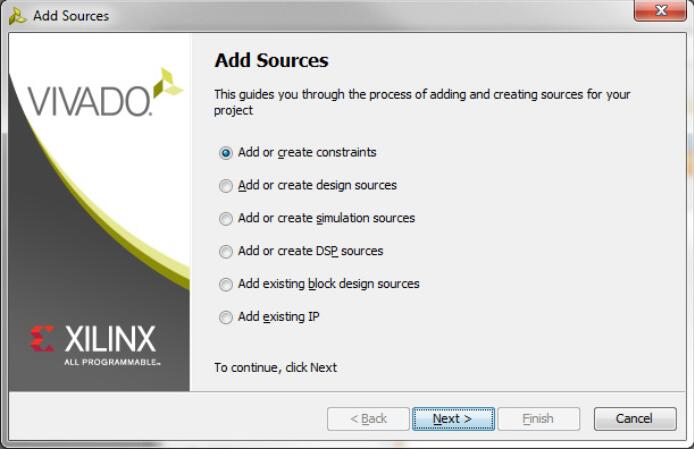
  你也可以看到综合得到的原理图，输入输出缓存和查找表都被用到：



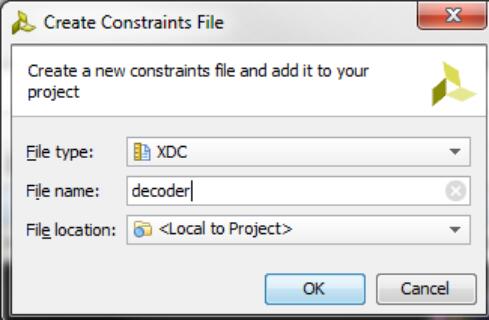
  在我们实现设计之前，需要为SW输入和LED输出指定FPGA的引脚。

  查询Basys3手册去决定FPGA的引脚。

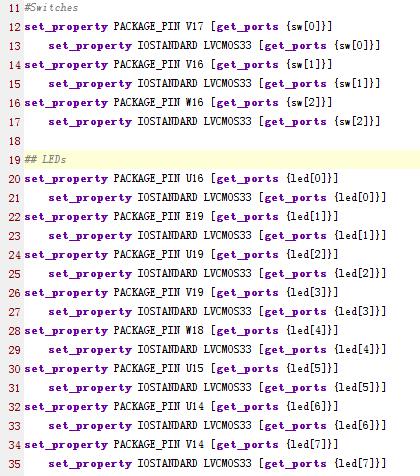
  点击Add Sources，选择”Add or create constraints”：



  给该约束文件命名decoder：



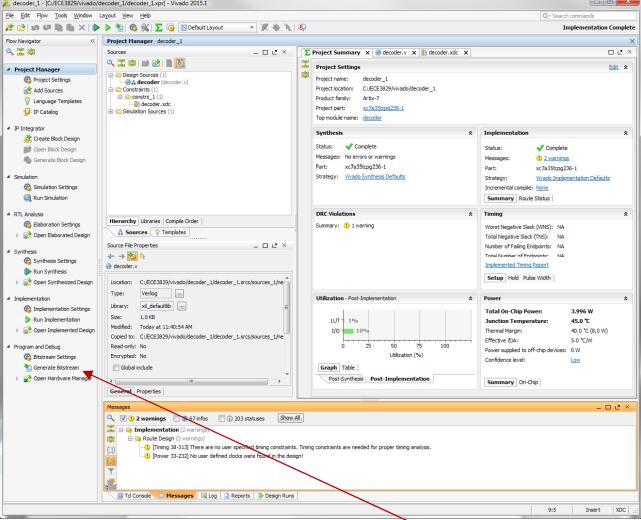
  打开 decoder.xdc 文件，按照下面的语句输入，实现对3-8译码器输入输出信号的约束和电平接口的描述。



  现在，我们已经正确分配了FPGA的引脚，接下来实现这个设计。

  在左侧栏中，点击 Run Implementation 。

  你会发现有两个警告，这是因为我们没有做任何时序约束。但本设计只是组合逻辑，用不着时钟，所以可以忽略。



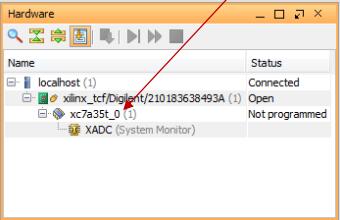
  现在，我们生产比特流，

  点击 Generate Bitstream。

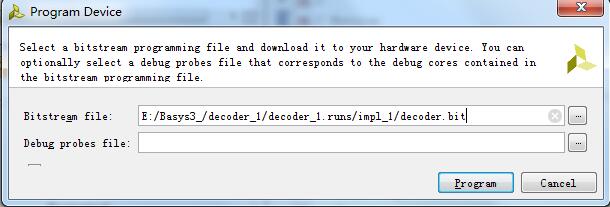
  下一步就是将比特流下载进FPGA。

  将Basys3开发板上的跳线帽JP1处选择JTAG模式，用USB数据线将PC与FPGA连接。

  在硬件窗口，你会看到 Xilinx xc7a35t\_0：



  右击箭头指示处，选择Program Device 并选择decoder.bit



  接着，点击Program。

  只需要几秒钟时间就可以下载完毕，开发板上led灯亮起。

  这时候，你就发现你设计的3-8译码器已经在FPGA上实现了。

  改变三个拨码开关（SW2，SW1， SW0），8个led的亮灭会相应变化。

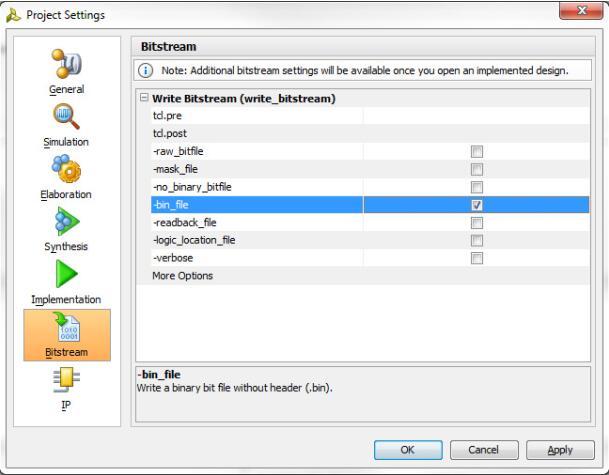
   你已经完成设计、综合、实现、生成比特流、下载进FPGA这一整个流程。这虽然是个简单的设计示例，但是任何设计都是重复这个过程的。

   关闭硬件窗口，回到工程管理窗口（Project Manager）。

**烧写串行FLASH**

  FPGA是易失性器件，所以比特流(.bit)在卡发版断电之后不会保存。我们可以将.bit文件加载在Basys3开发板的串行FLASH上，上电时装载flash中的.bit。

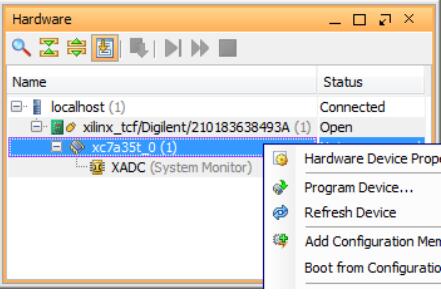
  点击 Bitstream Settings，勾选-bin\_file：



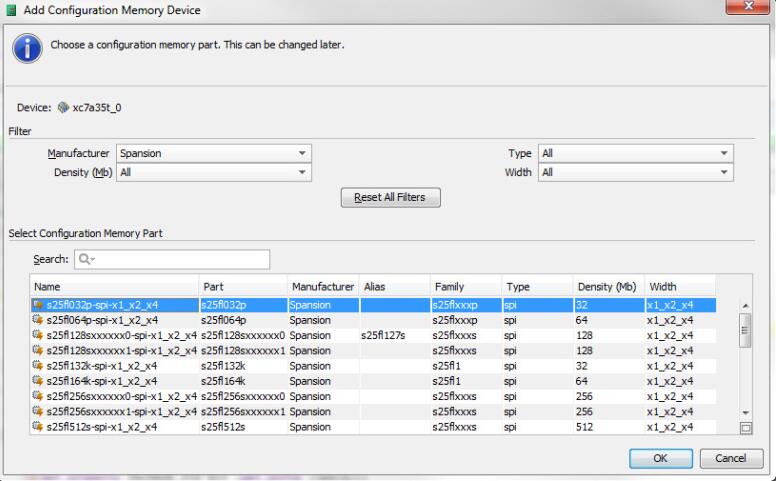
  点击OK，重新点击 Generate Bitstream。

  生成完毕之后，你可以在路径 decoder\_1\decoder\_1.runs\impl\_1下看到decoder.bit和decoder.bin文件。

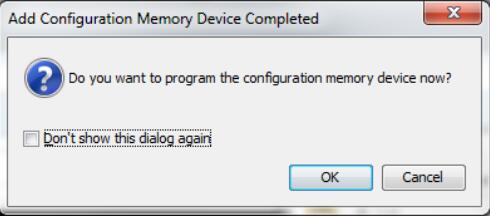
  打开 Hardware Manager窗口，连接Basys3开发板。右击FPGA型号选择Add Configuration Memory Device：



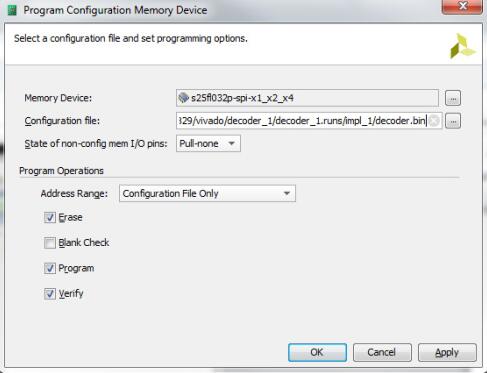
  选择串行FLASH的型号：



  点击OK。



  点击OK。  
  选择配置文件（decoder.bin）：



  点击OK。

  注意：该过程会擦出之前任何存在QSPI flash中的设计文件。

  成功烧写之后，你可以给Basys3开发板断电，将JP1跳线帽选择QSPI模式。

  开发板供电数秒之后Done灯亮起，意味着你的编码器设计文件自动从flash加载到FPGA中。你可以移动拨码开关观察LED来验证设计设计的正确性。

**代码分析http://img.baidu.com/hi/face/i_f24.gif**

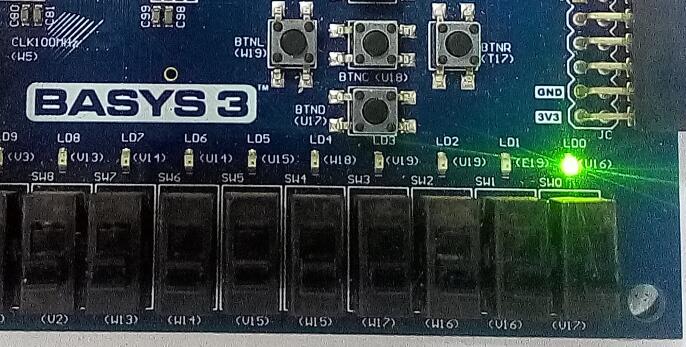
  输入信号 wire [2:0]sw，输出信号 reg [7:0]led。

  这里的led必须为reg型，因为在always中不断改变。

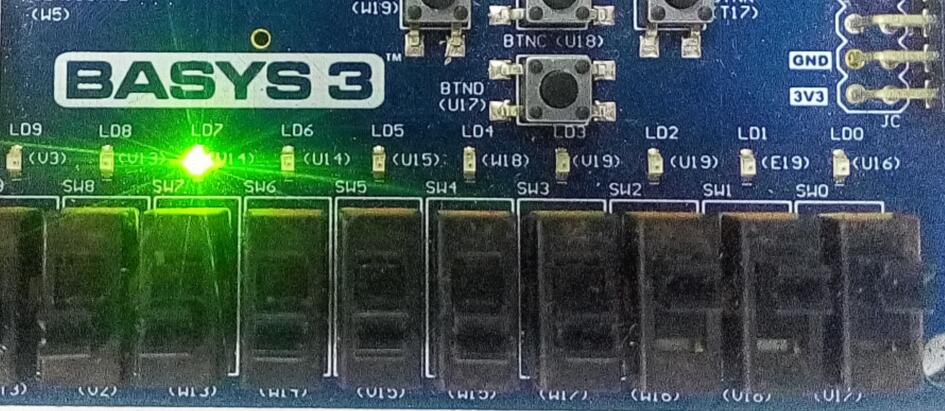
  这里使用case语句实现 3-8译码器。

  sw的电平是该组合逻辑的触发信号。每种SW信号对应一种led亮灯模式。

  比如将拨码开关都关闭，对应000，那么输出为 0000\_0000，led0亮。



  拨码开关都打开，对应111，则输出1000\_0000，led7亮。



上传的压缩包是Vivado 2015.2 工程文件,内含Basys3\_Master.xdc。