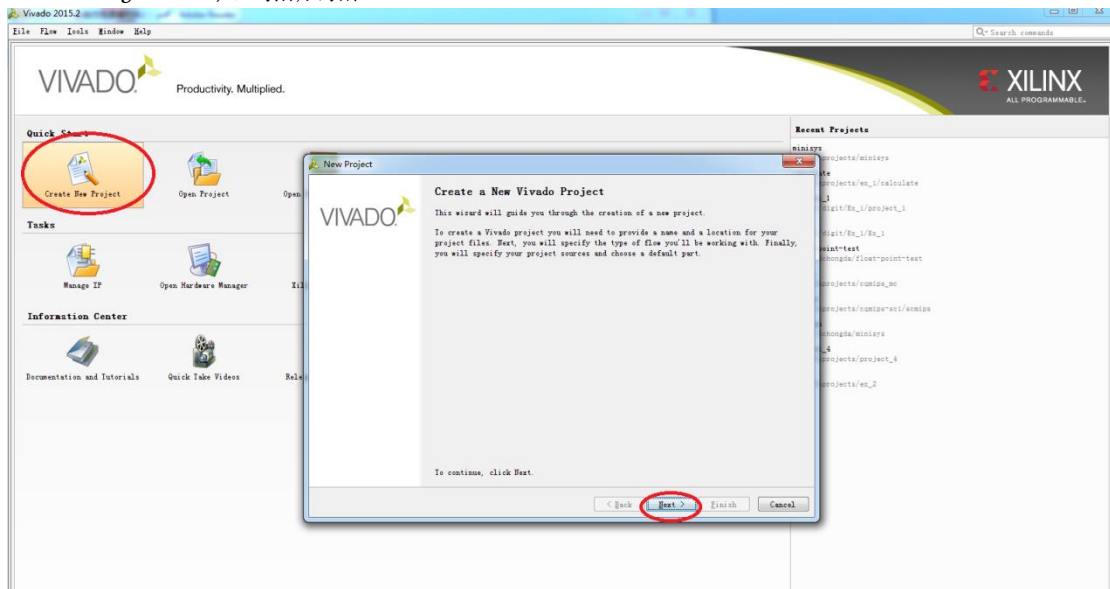


# Vivado 设计流程

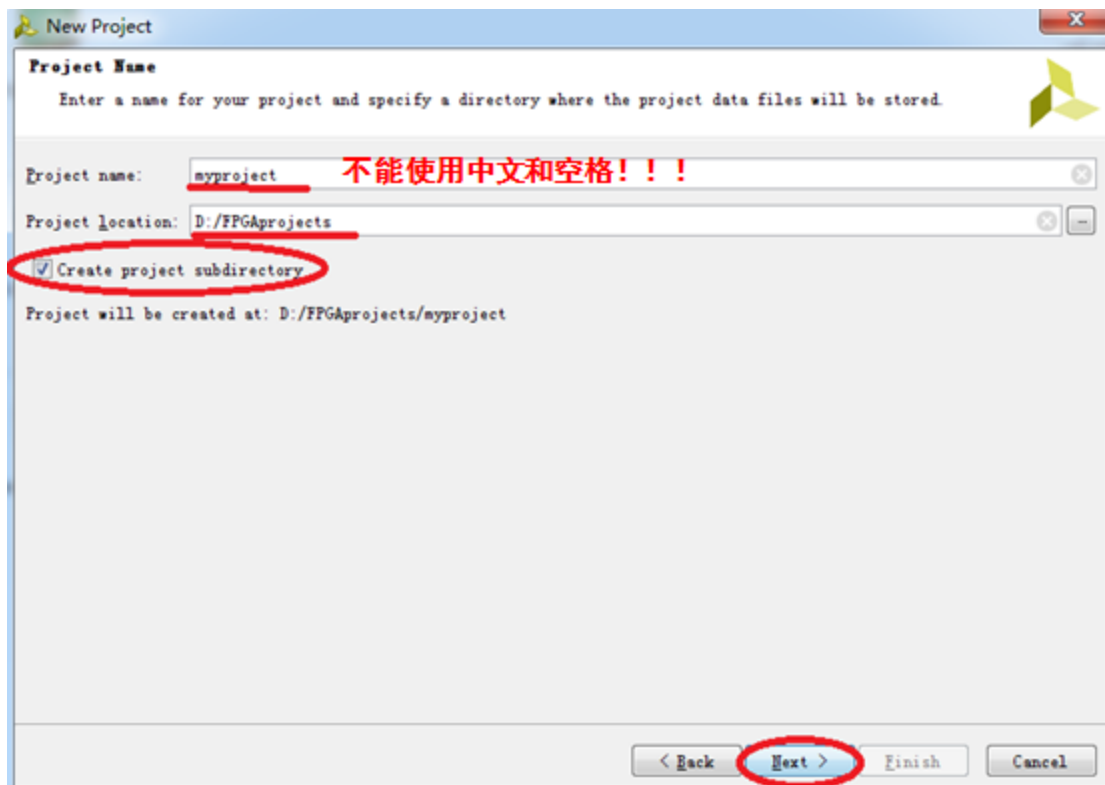
以一个简单的 16 位拨码开关的读和 16 位 LED 灯的输出电路为例，利用 Verilog HDL 语言，在 Vivado 中创建简单的 16 位拨码开关的输入和 16 位 LED 灯的输出电路,将设计下载的 Xilinx 大学计划 ARTIX-7 平台。

## 一、 新建工程

- 1、打开Vivado，然后点击创建一个新项目（或者在菜单栏选择File->New Project...），然后点next。

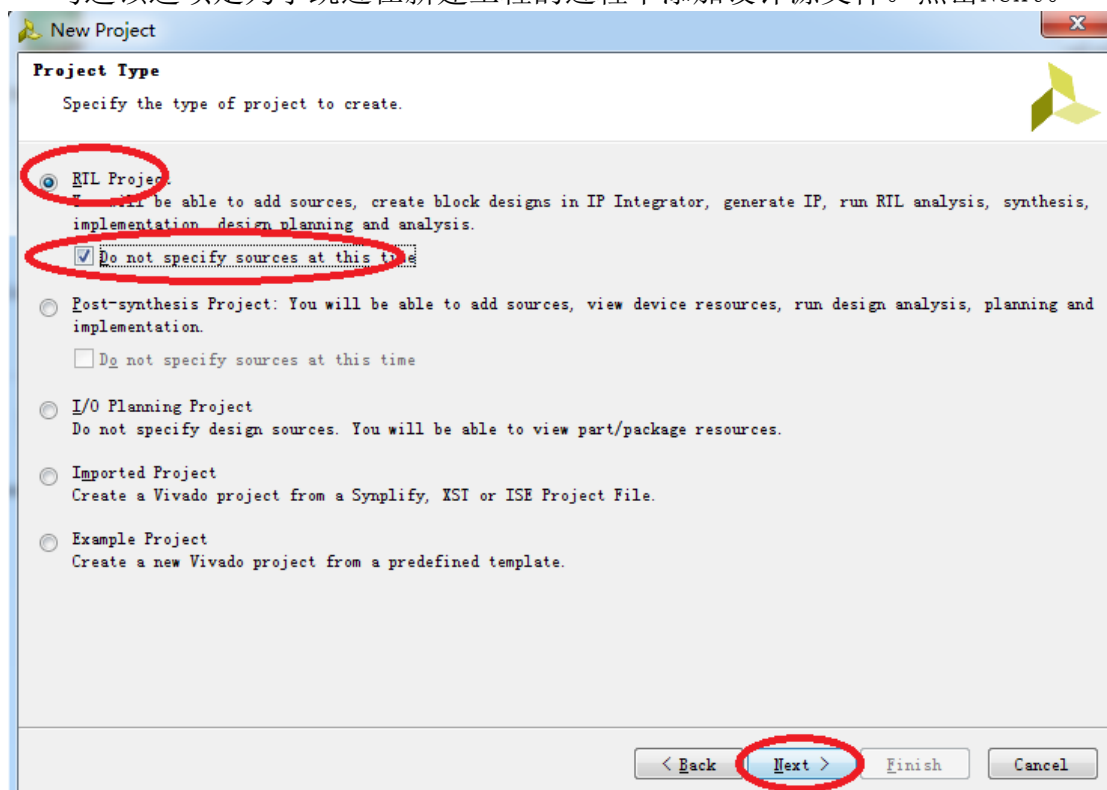


- 2、输入工程名称、选择工程存储路径，并勾选Create project subdirectory选项，为工程在指定存储路径下建立独立的文件夹。设置完成后，点击Next。最后，如图，整个项目将在 D:/FPGAprojects/myproject中。



注意：工程名称和存储路径中不能出现中文和空格，建议工程名称以字母、数字、下划线来组成。

- 3、选择RTL Project一项，并勾选Do not specify sources at this time，勾选该选项是为了跳过在新建工程的过程中添加设计源文件。点击Next。



- 4、根据使用的FPGA开发平台，选择对应的FPGA目标器件。Nexys4开发板请选择Artix-7 XC7A100TCSG324-1的器件，即Family和Subfamily均为Artix-7，封装形式（Package）为CSG324，速度等级（Speed grade）为-1，温度等级（Temp Grade）为C，**Basys3则为xc7a35tcpg236-1**，可查看Basys3官方指导手册）。点击Next。

**New Project**

**Default Part**

Choose a default Xilinx part or board for your project. This can be changed later.

Select: ☒ Parts ☐ Boards

Filter

Product category: All Package: csg324

Family: Artix-7 Speed grade: -1

Sub-Family: Artix-7 Temp grade: C

Si Revision: All Remaining

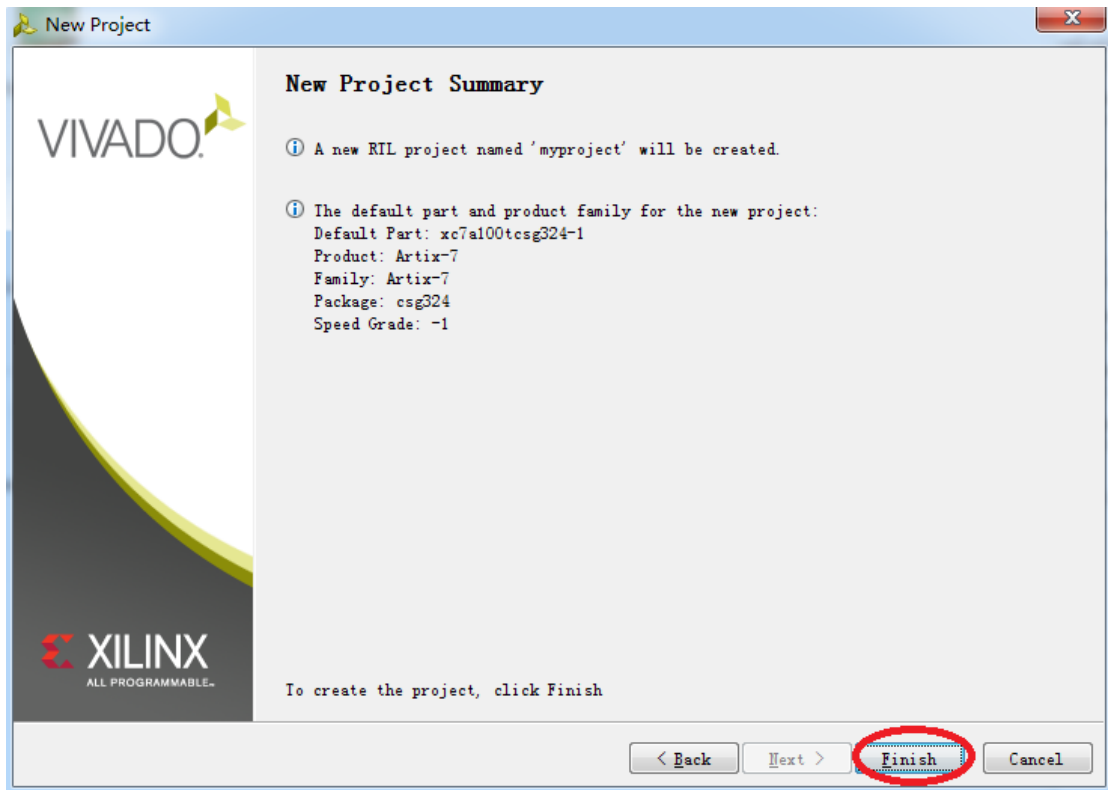
Reset All Filters

Search:

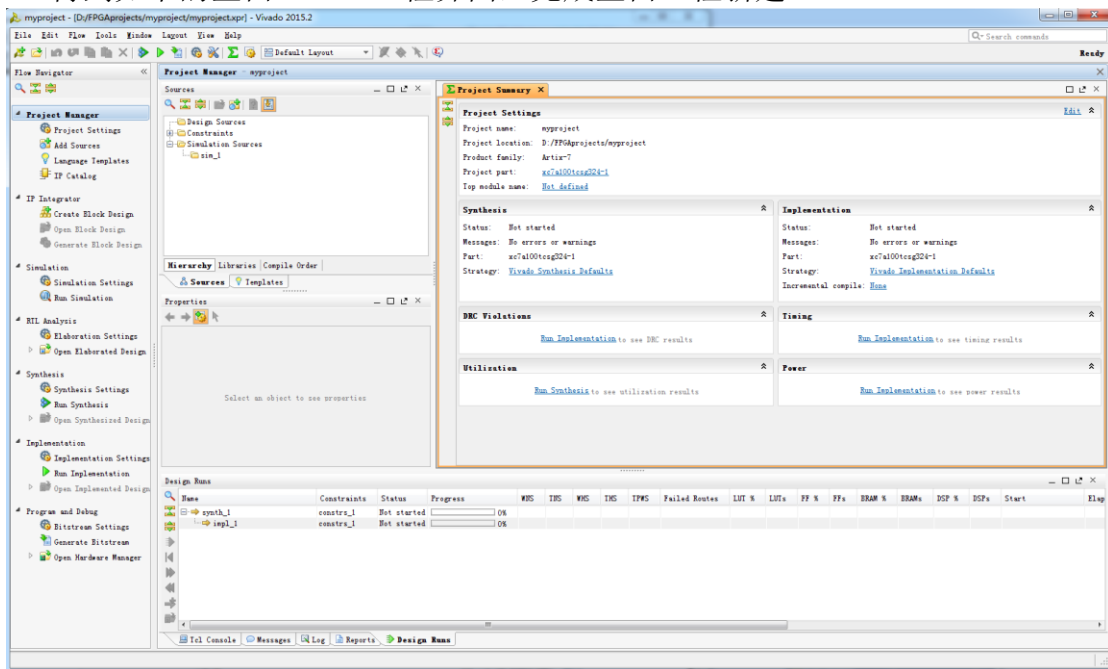
Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	DSPs	Gb Transceivers
xc7a15tcsg324-1	324	210	10400	20800	25	45	0
xc7a35tcsg324-1	324	210	20800	41600	50	90	0
xc7a50tcsg324-1	324	210	32600	65200	75	120	0
xc7a75tcsg324-1	324	210	47200	94400	105	180	0
<b>xc7a100tcsg324-1</b>	<b>324</b>	<b>210</b>	<b>63400</b>	<b>126800</b>	<b>135</b>	<b>240</b>	<b>0</b>

< Back **Next >** Finish Cancel

- 5、确认相关信息与设计所用的的FPGA器件信息是否一致，一致请点击Finish，不一致，请返回上一步修改。

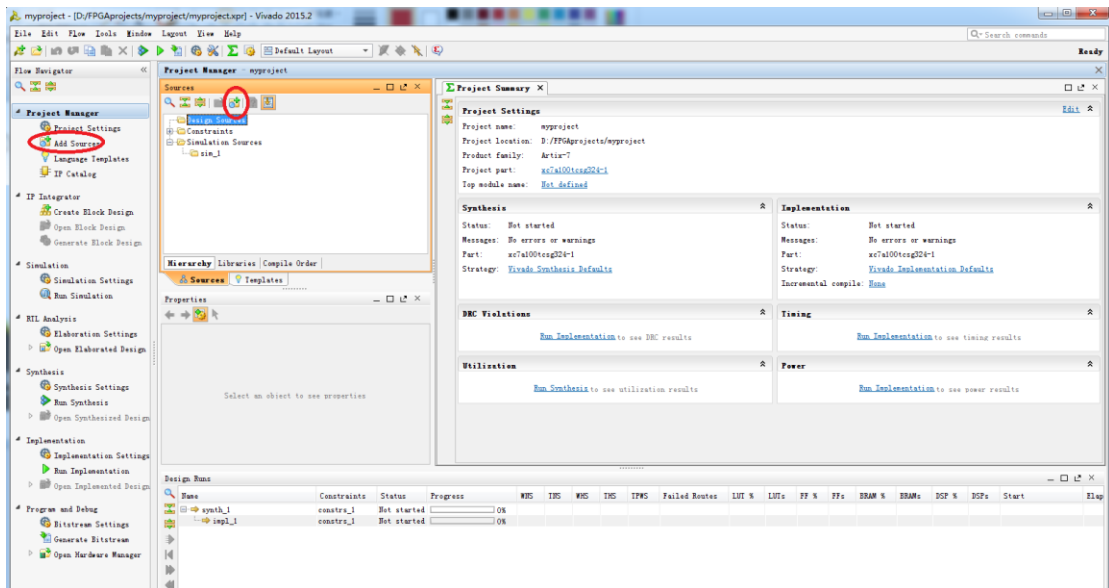


6、得到如下的空白Vivado工程界面，完成空白工程新建。

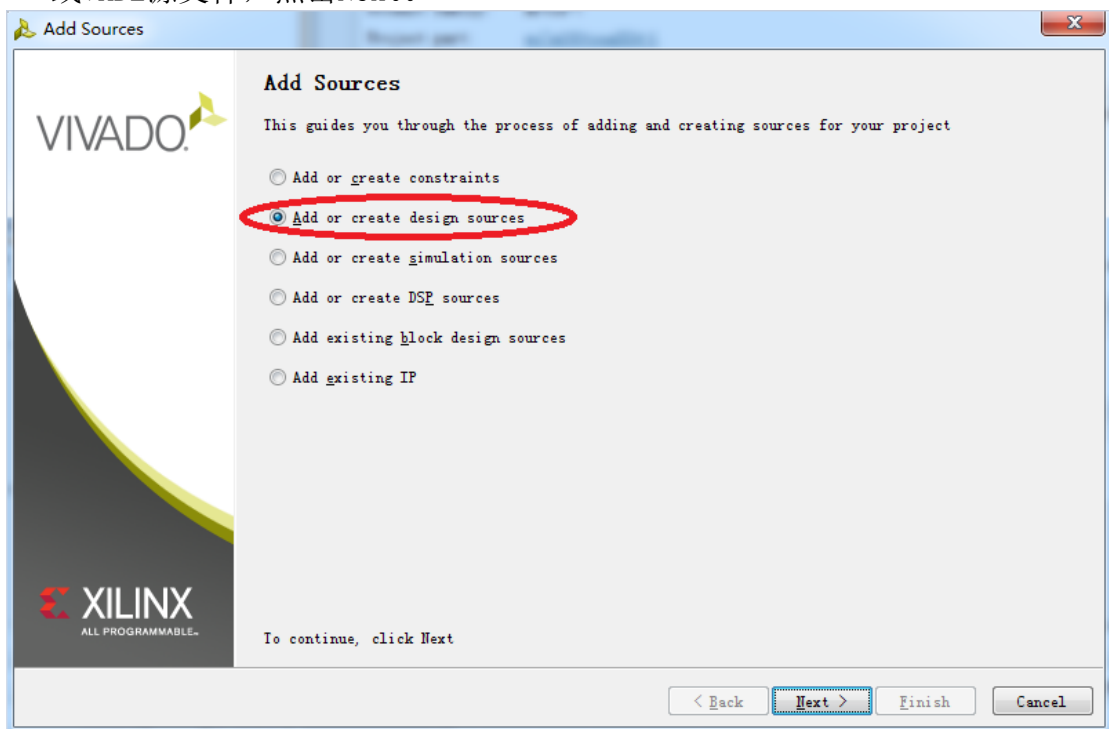


## 二、设计文件输入

- 1、点击Flow Navigator下的Project Manager->Add Sources或中间Sources中的对话框打开设计文件导入添加对话框。

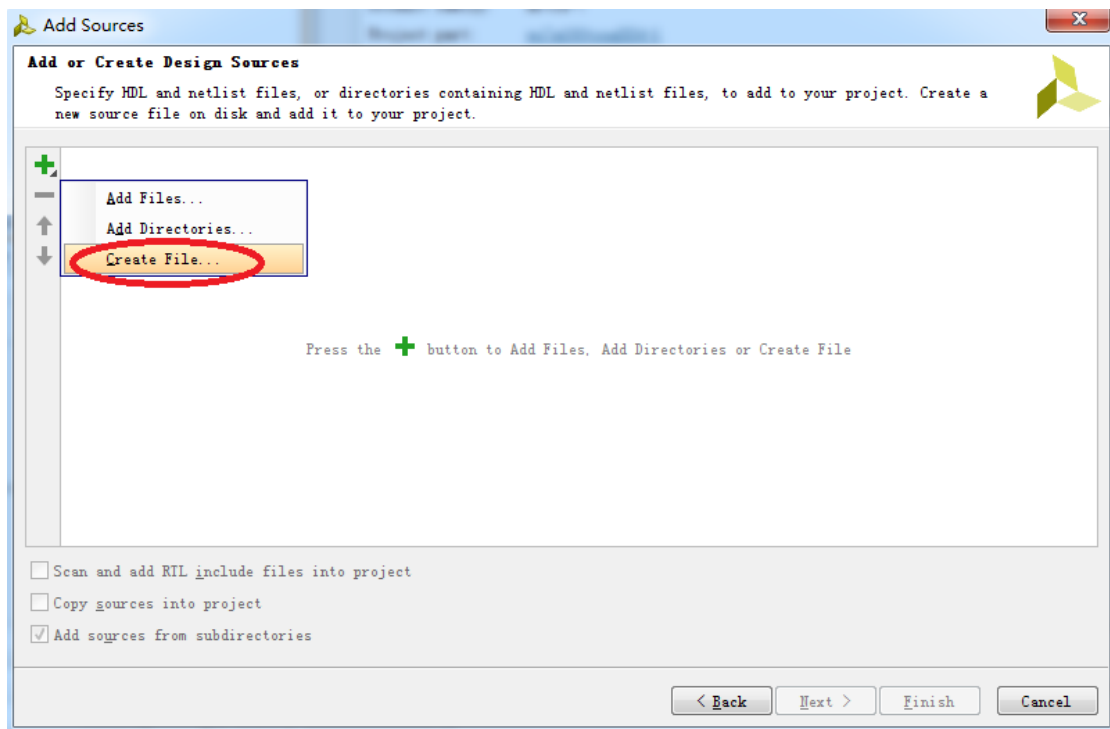


2、选择第二项Add or Create Design Sources，用来添加或新建Verilog或VHDL源文件，点击Next。

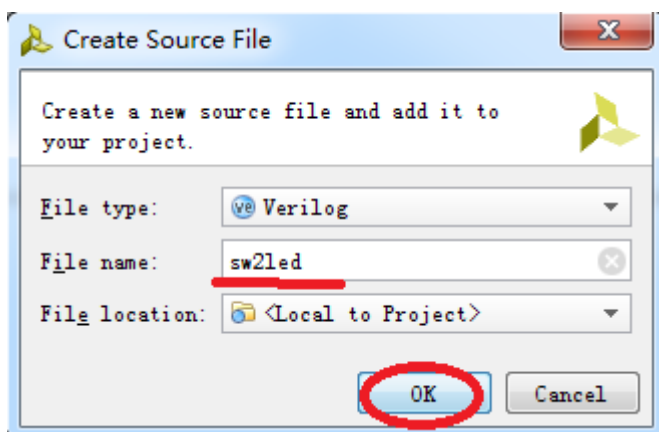


注意：第一项为约束文件，第三项为仿真文件，后面会用到。

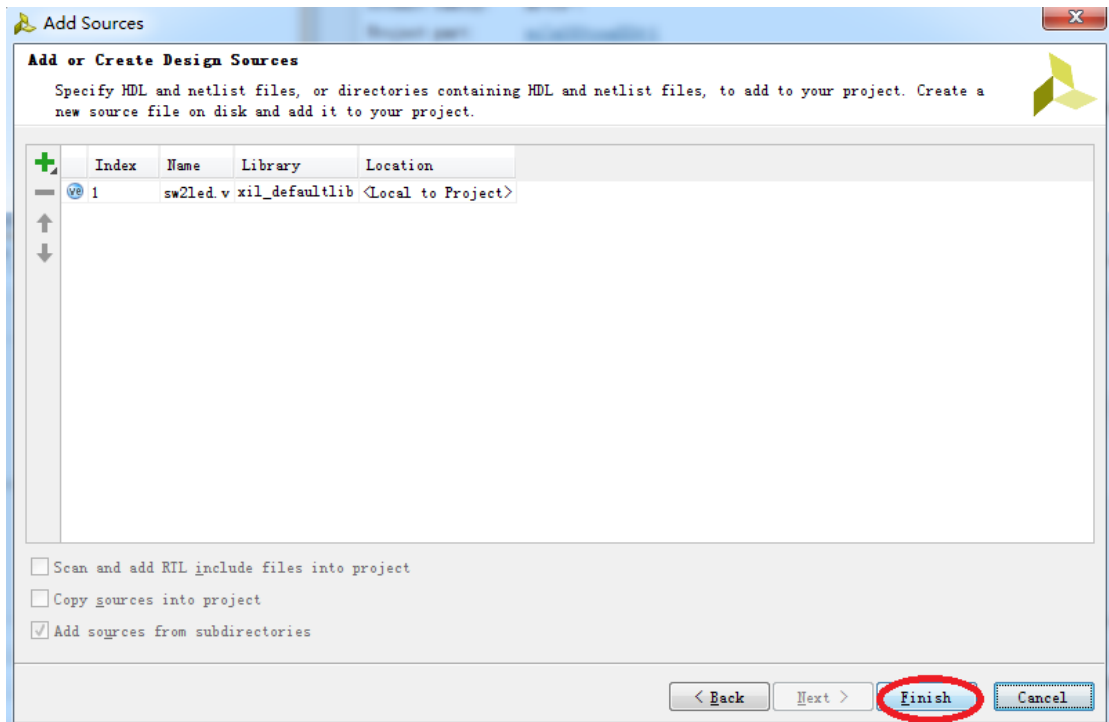
3、如果有现有的V/VHD文件，可以通过Add Files一项添加。在这里，我们要新建文件，所以选择Create File一项。



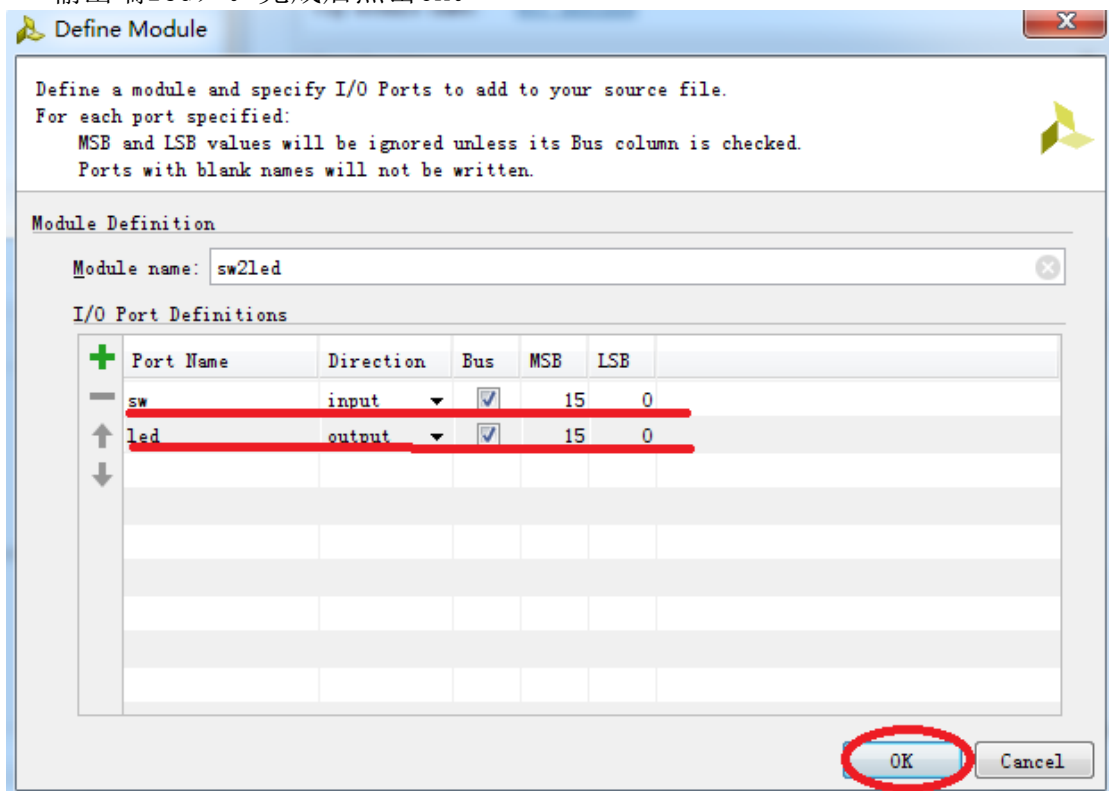
4、在Create Source File中输入File Name，点击OK。



5、点击finish

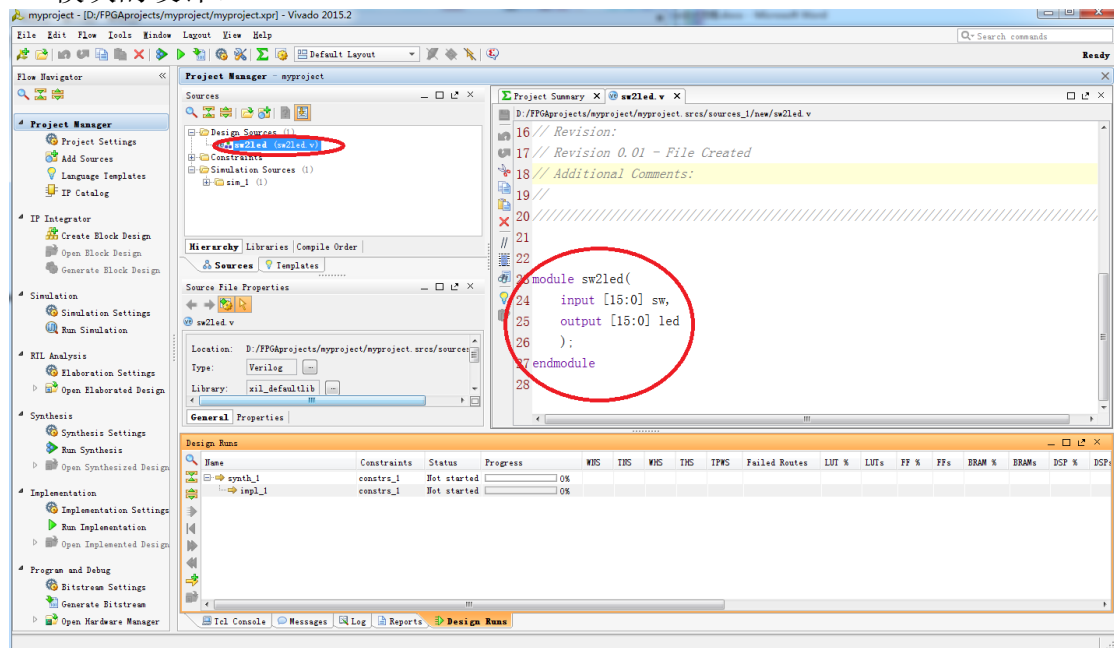


- 6、在弹出的Define Module中的I/O Port Definition, 输入设计模块所需的端口, 并设置端口防线, 如果端口为总线型, 勾选Bus选项, 并通过MSB和LSB确定总线宽度(如图定义了一个16位的输入sw, 和一个16位的输出端led)。完成后点击OK。



注意：此处也可以不定义端口，直接点击OK，而直接在后面的设计文件中定义输入输出端口。

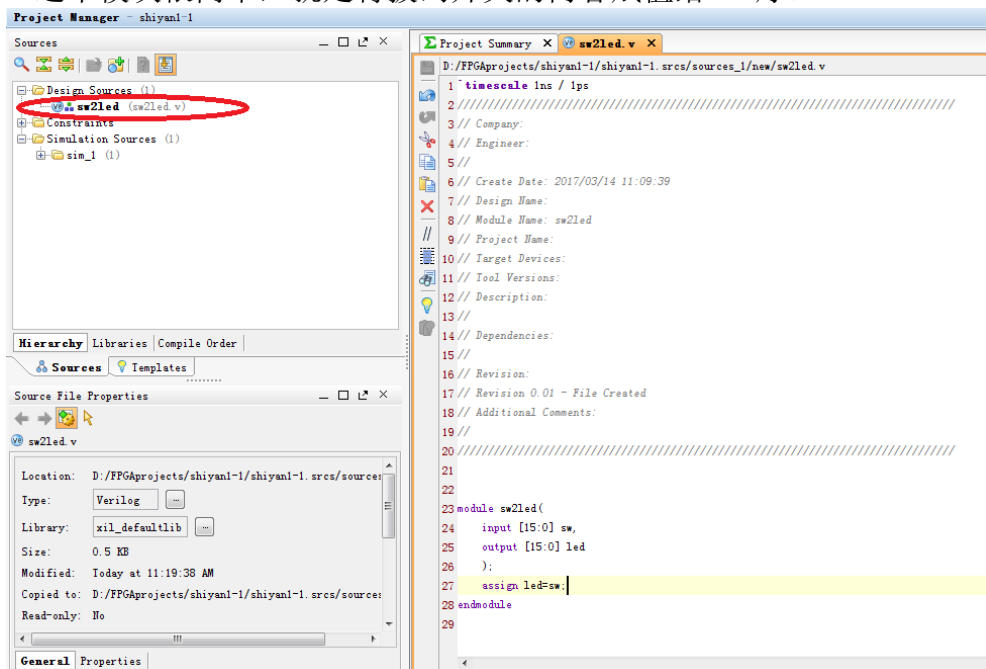
- 7、新建的设计文件（此处为sw2led.v）即存在于Sources中的Design Sources中。双击打开该文件，可以看到该文件只有刚刚定义的一个16位的输入sw，和一个16位的输出端led，添加相应的设计代码以完成该模块的设计。



添加完整的模块代码为：

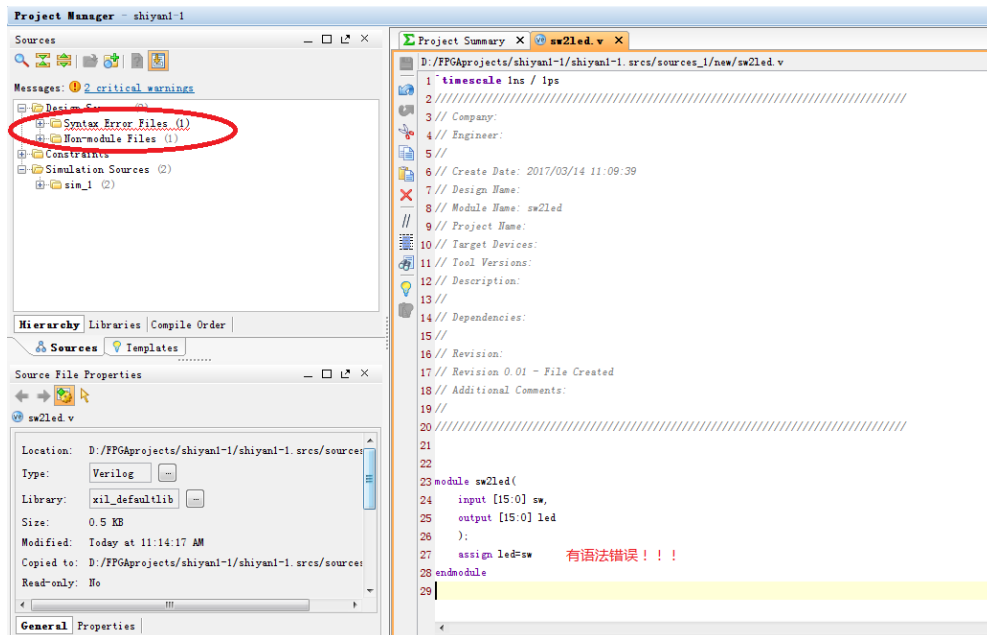
```
module sw2led(  
    input [15:0] sw,  
    output [15:0] led  
);  
    assign led=sw;  
endmodule
```

这个模块很简单，就是将拨码开关的内容赋值给led灯。





注意：当设计文件有语法错误的时候点击保存后，工程管理窗口内会出现语法错误的文件夹如下图，应仔细检查语法错误！！

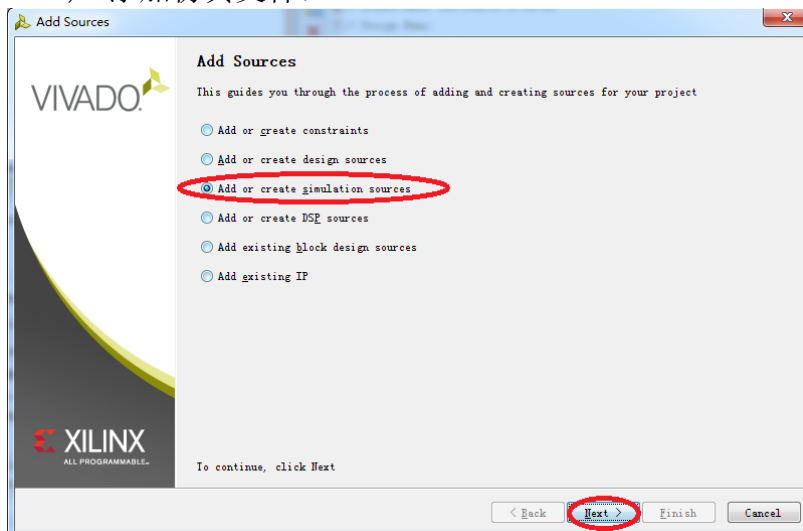


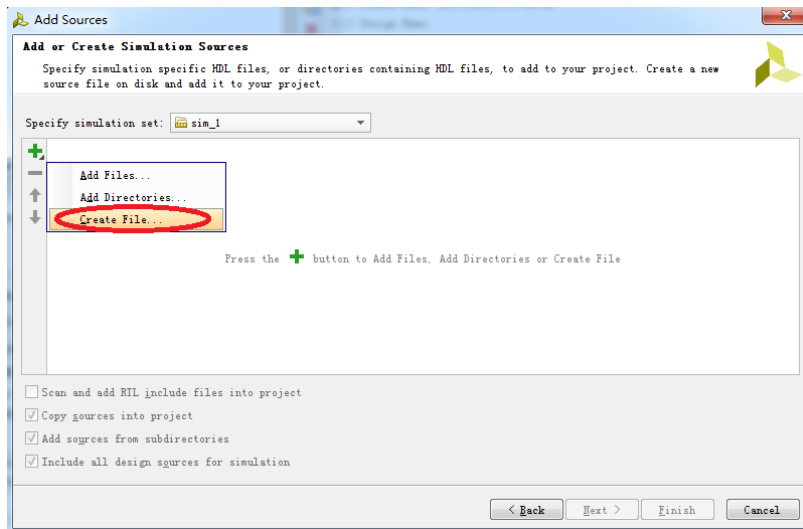
注意：修改源文件之后一定记得要先保存，否则修改无效，运行报错！

### 三、 仿真

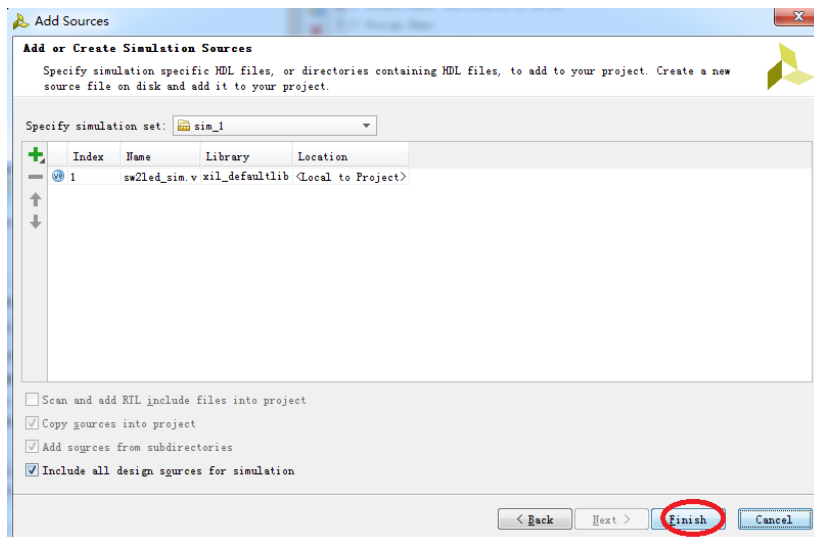
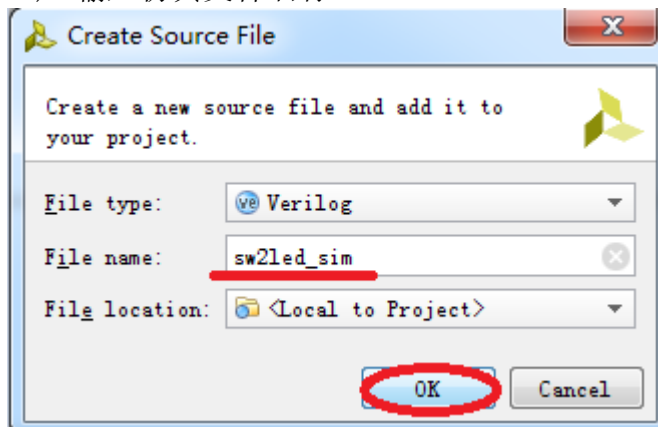
编写仿真文件(sw2led\_sim.v)，验证电路设计的正确性。如不仿真验证，则可直接添加约束文件综合、实现、生成 BIT 流下载到开发板。

#### 1) 添加仿真文件：

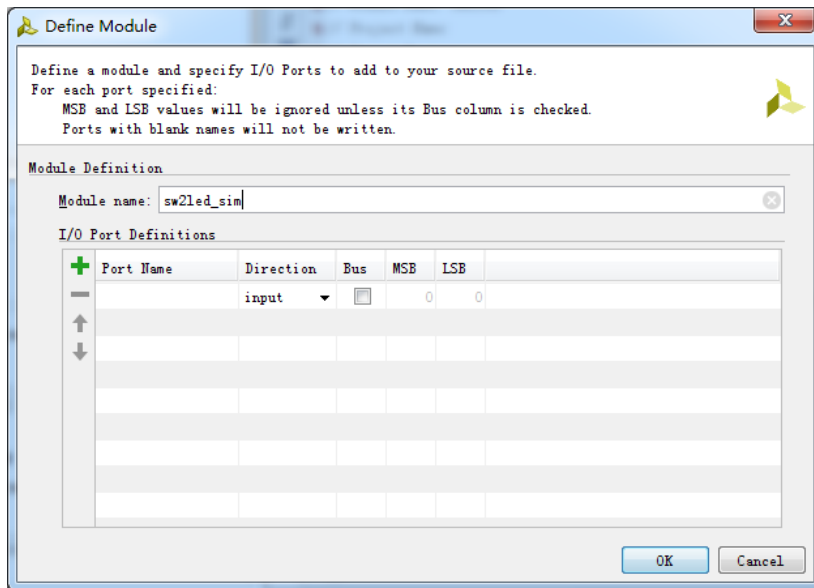




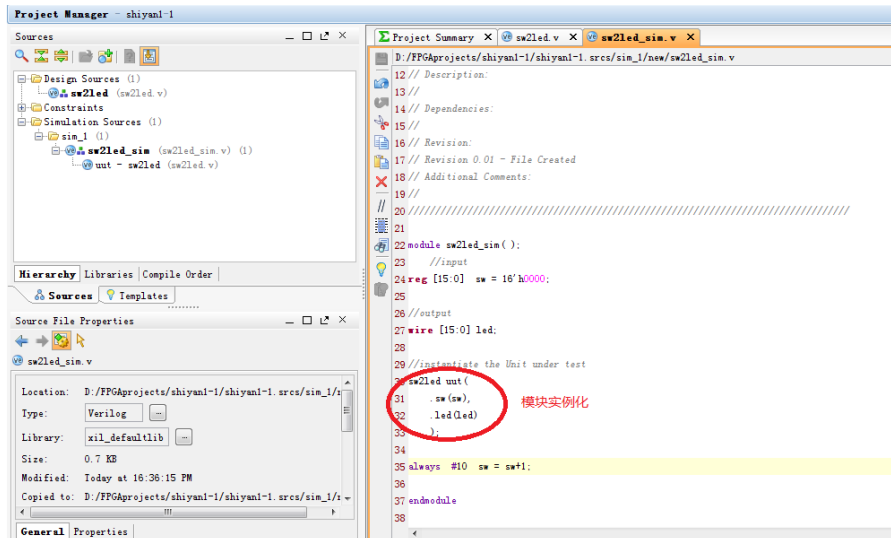
2) 输入仿真文件名称:



3) 仿真模块不需要定义输入输出端口:

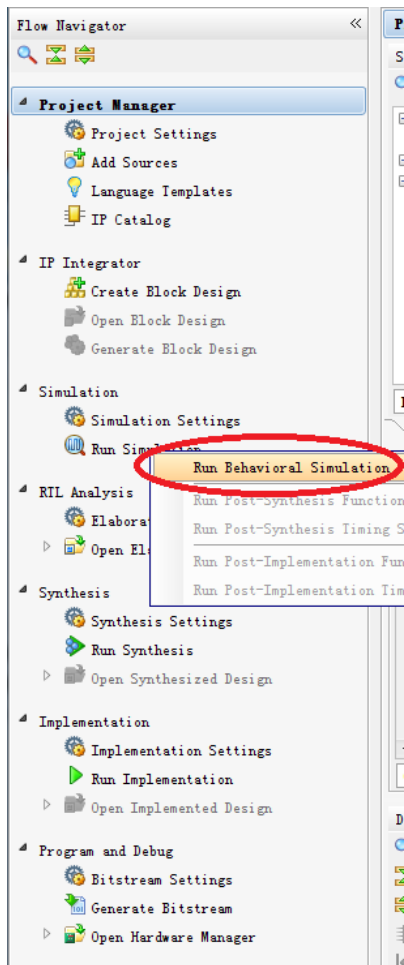


#### 4) 编写仿真代码:

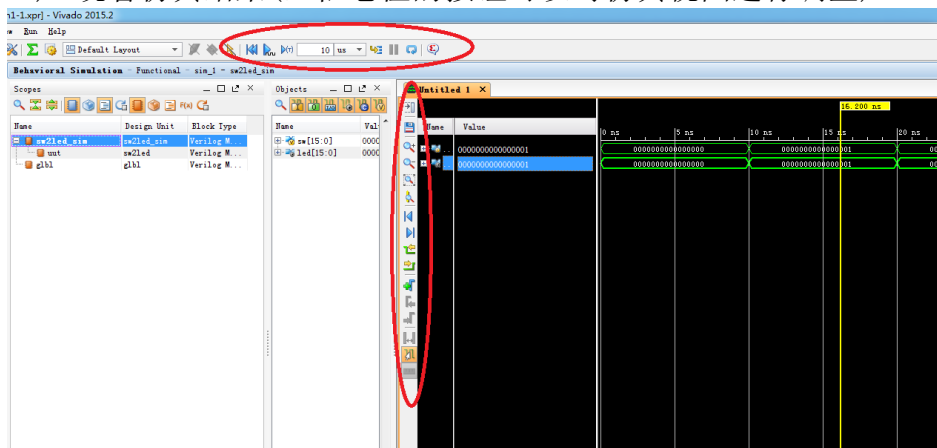


仿真文件中，不需要定义输入输出端口，但一定要对仿真的模块进行实例化（本次实验是对 sw2led 模块进行实例化），因此，要定义传给实例化模块的输入输出变量，并给出输入变量的初始值 and 变化值，以便观察输出值。

#### 5) 在导航栏里点击仿真（行为仿真）:



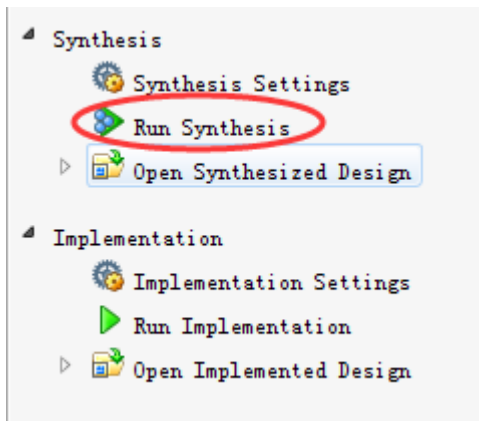
6) 观看仿真结果(红框卷住的按钮可以对仿真视图进行调整):



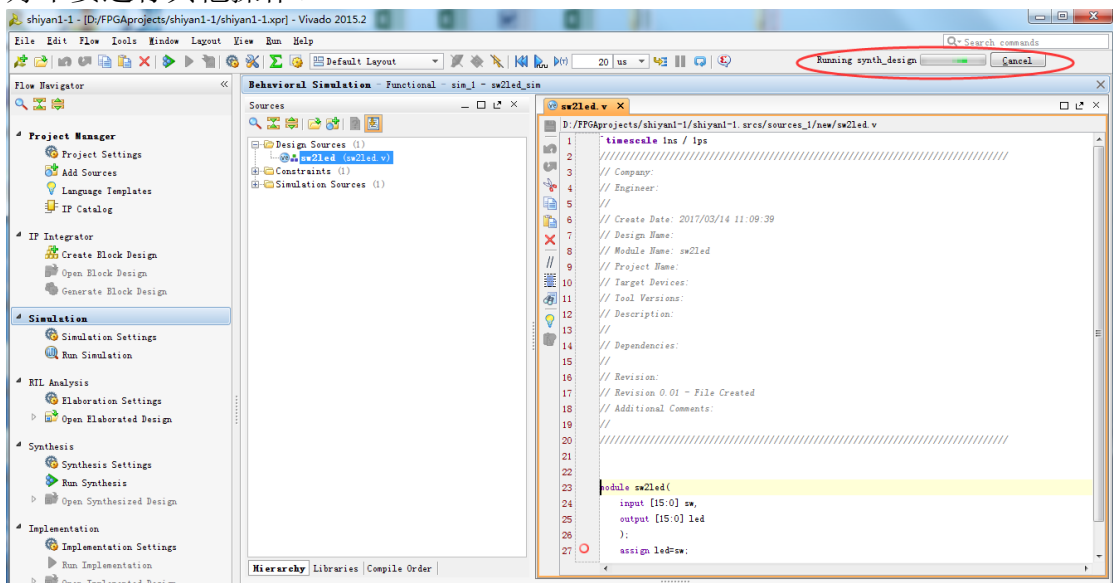
由于我们设计的电路就是将 sw 的值传给 led，因此，可以看到，led[15:0]的值始终等于 sw[15:0]的值，根据仿真文件的代码可以看到 sw 的值每隔 10ns 增加 1，led 的值也是每隔 10ns 增加 1。

#### 四、 综合

在Flow Navigator中点击Synthesis下的Run Synthesis选项，将电路设计文件综合成一个完整的电路，将设计的高层次描述转化为优化的门级网表。如果报错则

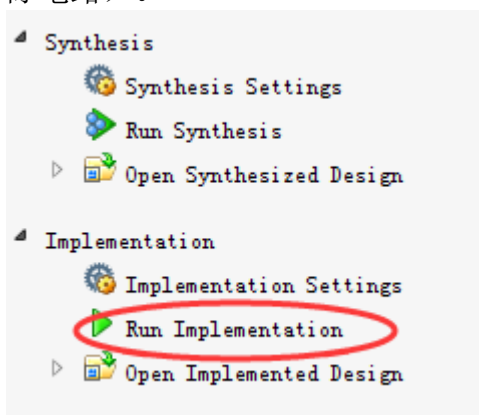


如图，绿色进度条在滚动，表示vivado环境正在运行，此时只需等待，最好不要进行其他操作。



## 五、 实现

在Flow Navigator中点击Implementation下的Run Implementation选项，实现设计综合网表（开发板资源对所设计的电路进行布局布线，转化为实际电路）。



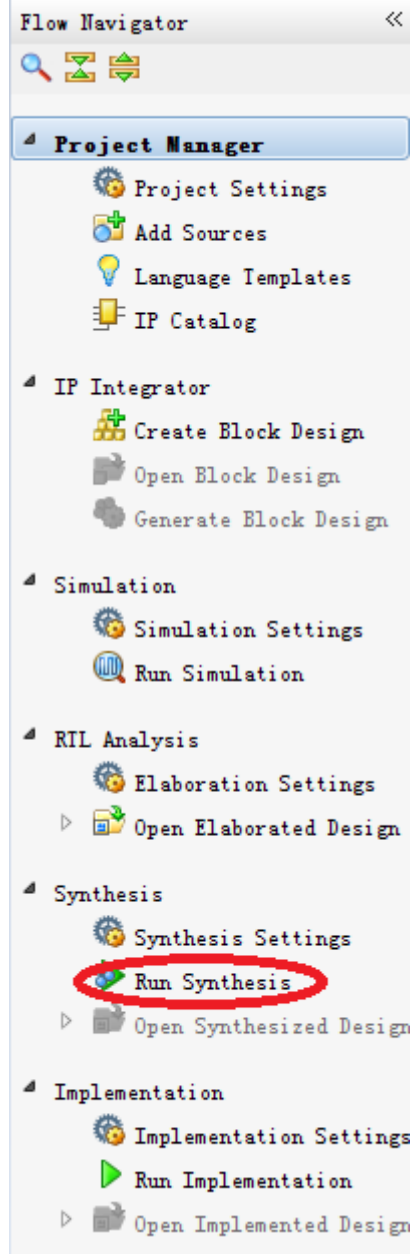
## 六、 添加约束文件

有两种方法可以添加约束文件，一是可利用Vivado中IO planning功能，二

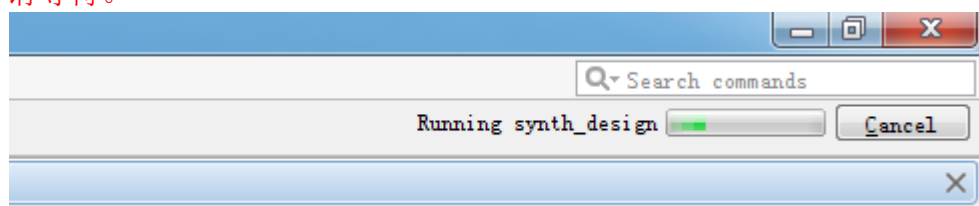
是可以直接新建XDC的约束文件，手动输入约束命令。

A. 分配管脚之第一种方法，利用IO planning。

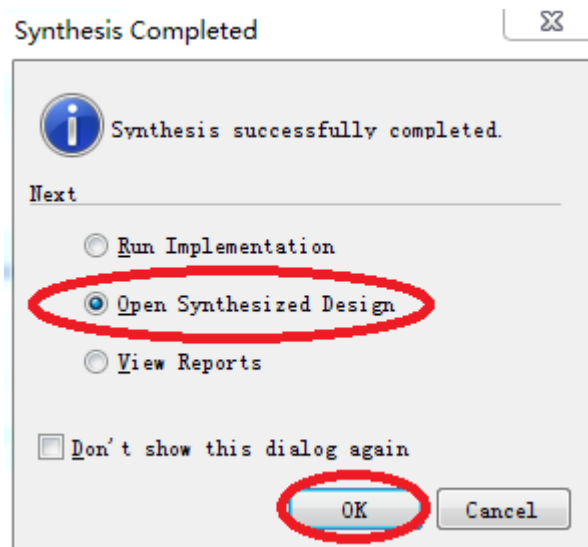
- 1) 点击Flow Navigator中Synthesis中的Run Synthesis，先对工程进行综合。



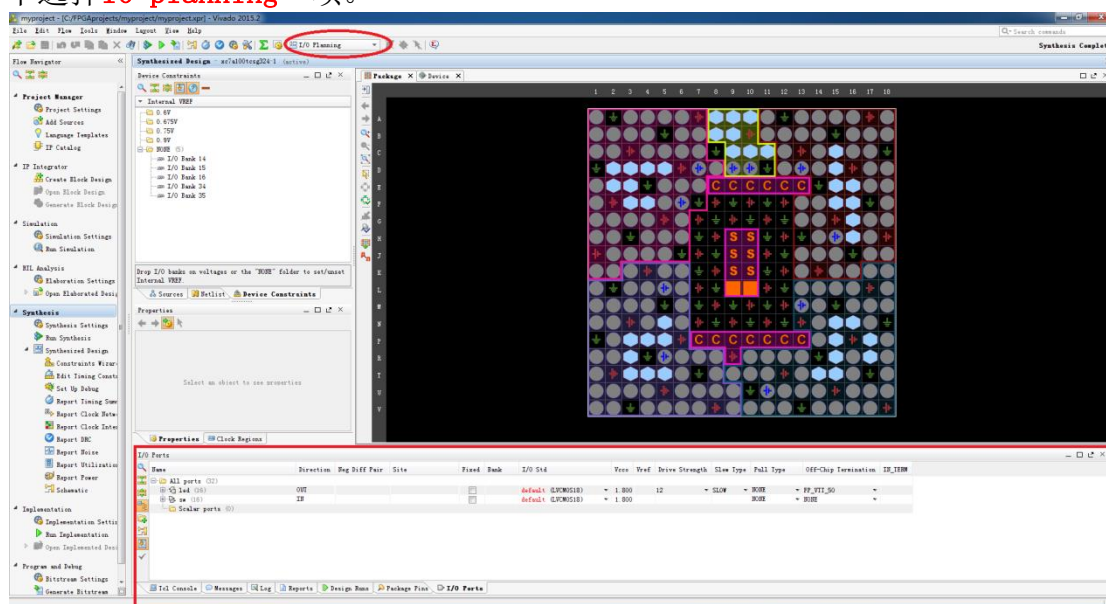
注意：窗口右上方有一个进度条显示当前状态，当处于滚动状态时请等待。



- 2) 综合完成之后，选择Open Synthesized Design，打开综合结果。



- 3) 此时应看到如下界面，如果没出现如下界面，在图示位置的layout中选择IO planning一项。



- 4) 在右下方的选项卡中切换到I/O ports一栏，并在对应的信号后，输入对应的FPGA管脚标号（或将信号拖拽到右上方Package图中对应的管脚上），并指定I/O std。

Name	Direction	Reg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	TR_TERM
led (16)	OUT					Multiple) LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[15]	OUT		V11			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[14]	OUT		V12			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[13]	OUT		V14			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[12]	OUT		V15			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[11]	OUT		V16			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[10]	OUT		V14			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[9]	OUT		V15			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[8]	OUT		V16			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[7]	OUT		V17			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[6]	OUT		V16			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[5]	OUT		V17			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[4]	OUT		V18			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[3]	OUT		V14			14 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[2]	OUT		V13			15 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[1]	OUT		V15			15 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
led[0]	OUT		V17			15 LVCNMOS33	3.300	12		SLOW	NOISE	PP_VTT_50	
sw (16)	IN					Multiple) LVCNMOS33	3.300				NOISE		
sw[15]	IN		V10			14 LVCNMOS33	3.300				NOISE		
sw[14]	IN		V11			14 LVCNMOS33	3.300				NOISE		
sw[13]	IN		V12			14 LVCNMOS33	3.300				NOISE		
sw[12]	IN		V16			25 LVCNMOS33	3.300				NOISE		
sw[11]	IN		V13			14 LVCNMOS33	3.300				NOISE		
sw[10]	IN		V16			14 LVCNMOS33	3.300				NOISE		
sw[9]	IN		U8			34 LVCNMOS33	3.300				NOISE		
sw[8]	IN		V8			34 LVCNMOS33	3.300				NOISE		
sw[7]	IN		V13			14 LVCNMOS33	3.300				NOISE		
sw[6]	IN		V18			14 LVCNMOS33	3.300				NOISE		
sw[5]	IN		V18			14 LVCNMOS33	3.300				NOISE		
sw[4]	IN		V17			14 LVCNMOS33	3.300				NOISE		
sw[3]	IN		V15			14 LVCNMOS33	3.300				NOISE		
sw[2]	IN		V13			14 LVCNMOS33	3.300				NOISE		
sw[1]	IN		V16			14 LVCNMOS33	3.300				NOISE		

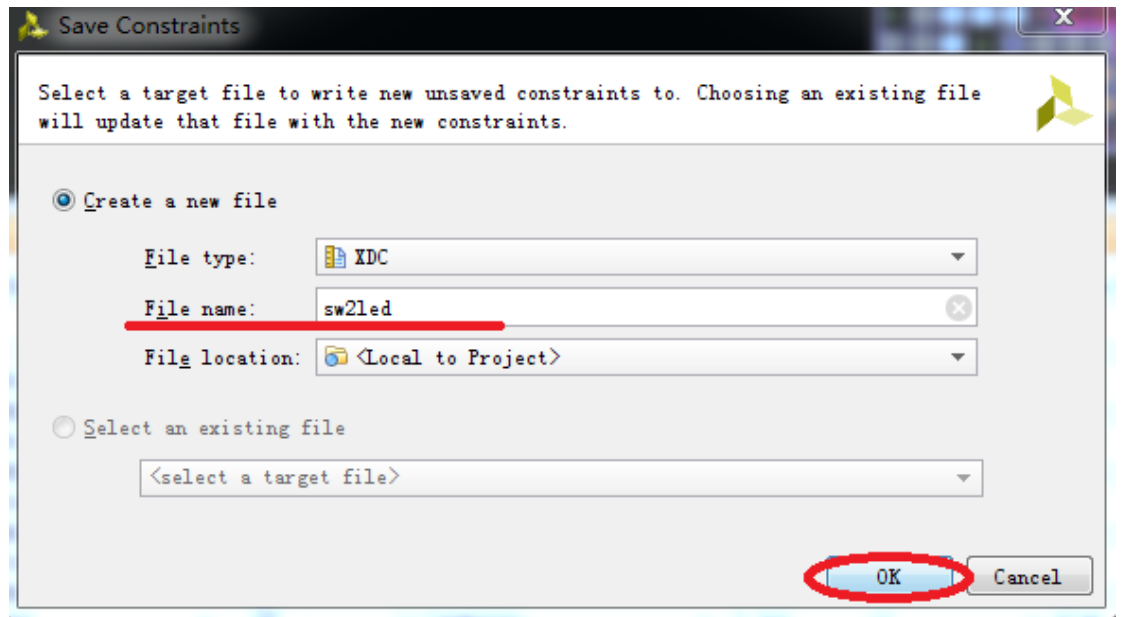
注意：具体的FPGA约束管脚和IO电平标准与板卡型号有关，不同型号的板卡（N4或者B3）需查看其相关的分配表，管脚电压设置均为LVCNMOS33。

如下表为Basys3对应的引脚分配表（详情可查看预习资料“Basys3实验官方指导手册”）：

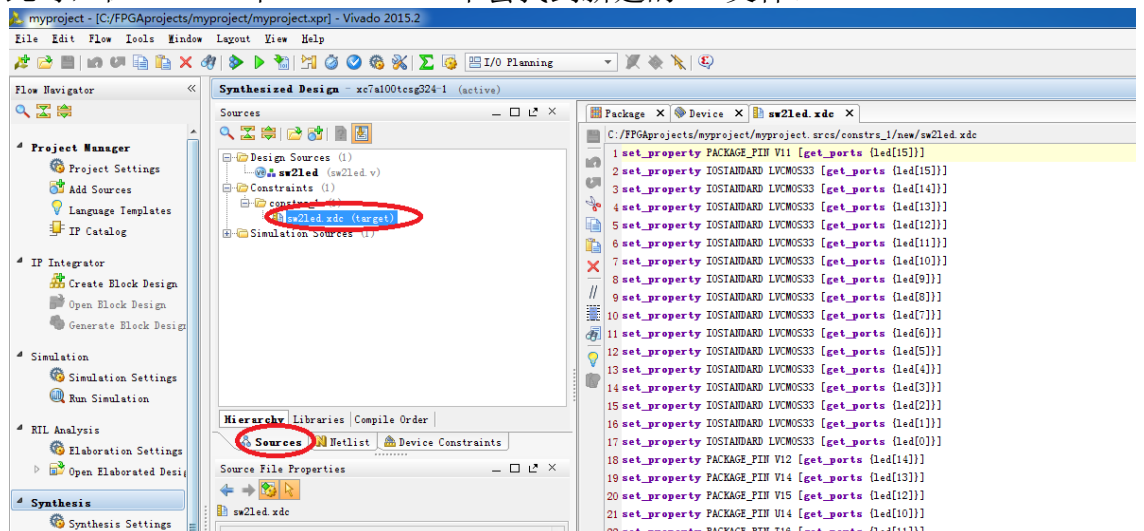
LED	PIN	CLOCK	PIN	SWITCH	PIN	BUTTON	PIN	Seven-segment digital tube	PIN
LD0	U16	MRCC	W5	SW0	V17	BTNU	T18	AN0	U2
LD1	E19			SW1	V16	BTNR	T17	AN1	U4
LD2	U19			SW2	W16	BTND	U17	AN2	V4
LD3	V19			SW3	W17	BTNL	W19	AN3	W4
LD4	W18			SW4	W15	BTNC	U18	CA	W7
LD5	U15			SW5	V15			CB	W6
LD6	U14			SW6	W14			CC	U8
LD7	V14			SW7	W13			CD	V8
LD8	V13	USB (J2)	PIN	SW8	V2			CE	U5
LD9	V3	PS2_CLK	C17	SW9	T3			CF	V5
LD10	W3	PS2_DAT	B17	SW10	T2			CG	U7
LD11	U3			SW11	R3			DP	V7
LD12	P3			SW12	W2				
LD13	N3			SW13	U1				
LD14	P1			SW14	T1				
LD15	L1			SW15	R2				

5) 完成之后，点击左上方工具栏中的**保存**按钮，工程提示新建XDC文件或选择工程中已有的XDC文件。在这里，我们要Create a new file，输入File name，点击OK完成约束过程。



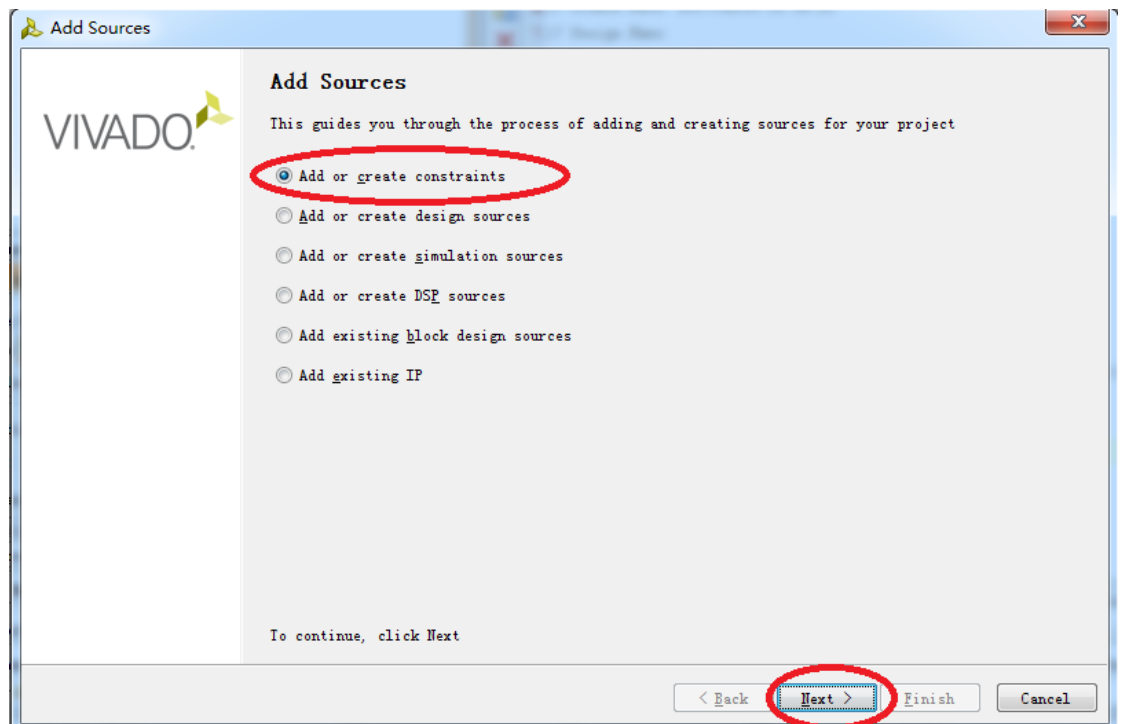


6) 此时，在Sources下Constraints中会找到新建的XDC文件。

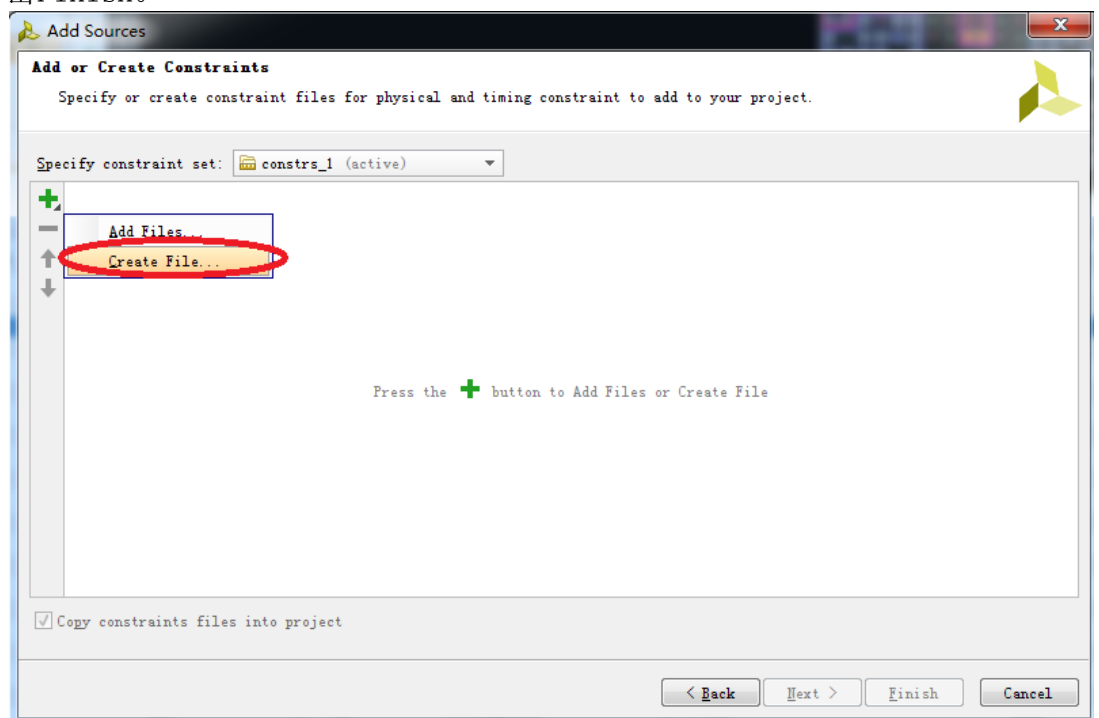


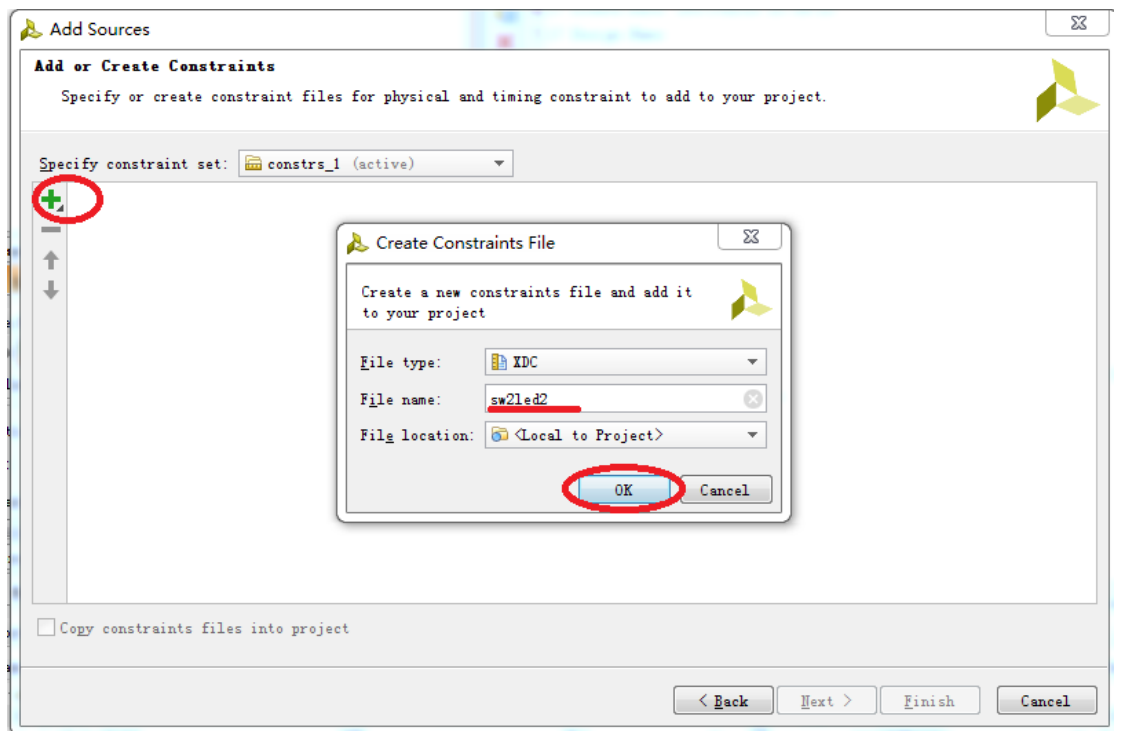
B. 如何利用第二种方法添加约束文件。

- 1) 点击Add Sources，选择第一项Add or Create Constraints一项，点击Next。

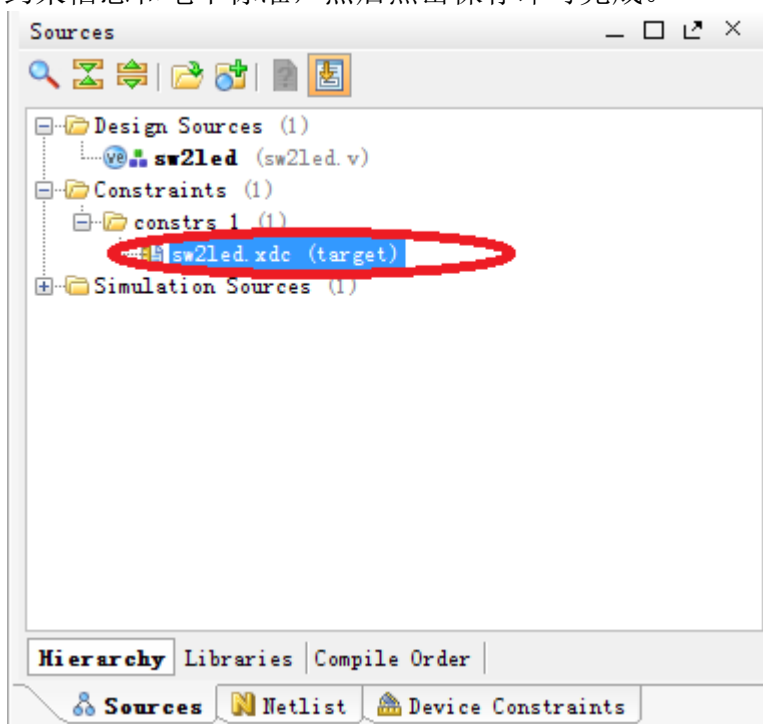


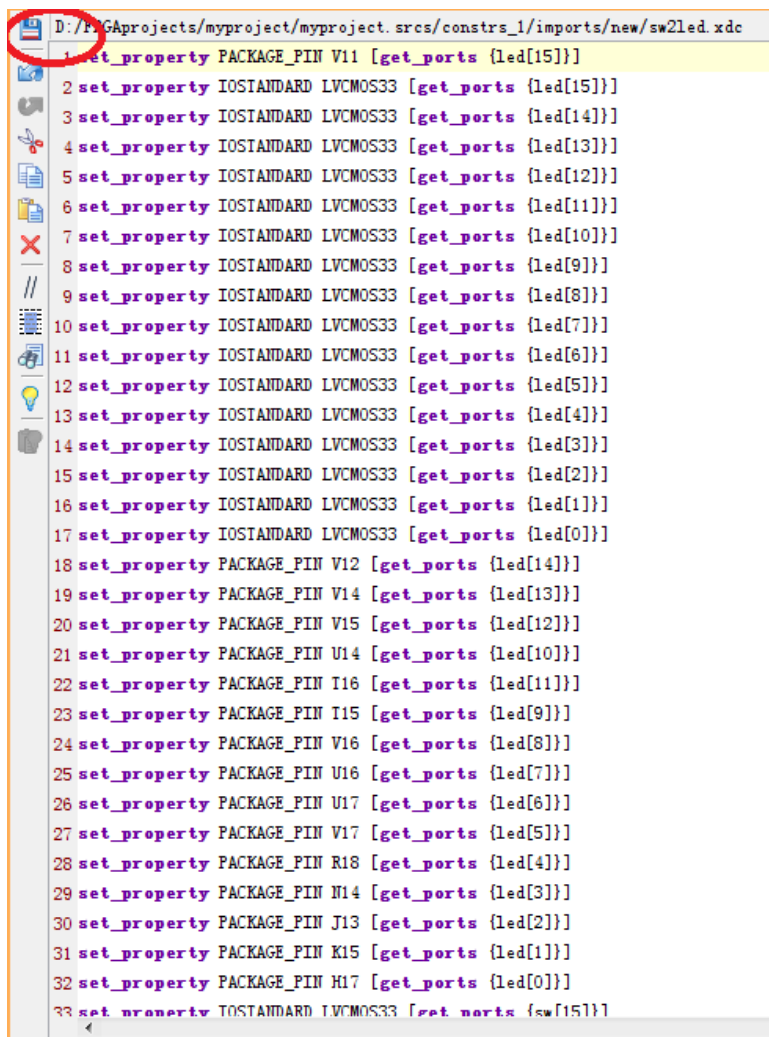
- 2) 点击Create File，新建一个XDC文件，输入XDC文件名，点击OK。点击Finish。





- 3) 双击打开新建好的XDC文件，并按照如下规则，输入相应的FPGA管脚约束信息和电平标准，然后点击保存即可完成。

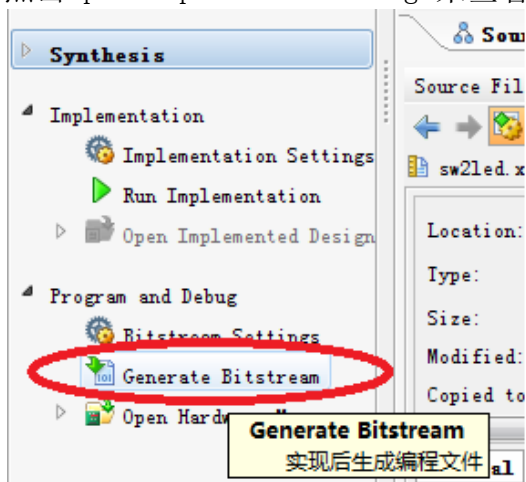


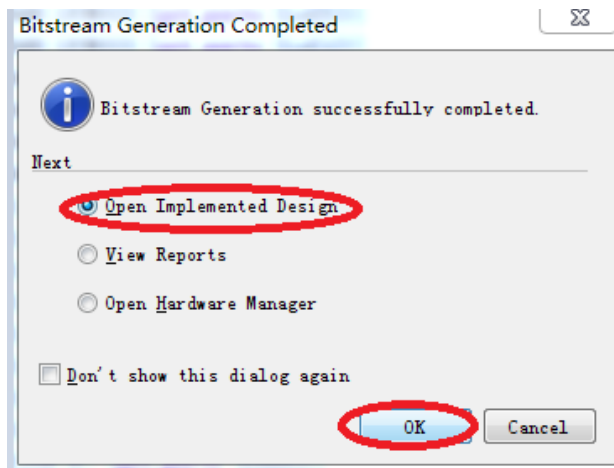


```
D:/FPGAprojects/myproject/myproject.srcs/constrs_1/imports/new/sw2led.xdc
1 set_property PACKAGE_PIN V11 [get_ports {led[15]}]
2 set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]
3 set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
4 set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
5 set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
6 set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
7 set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
8 set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
9 set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
10 set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
11 set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
12 set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
13 set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
14 set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
15 set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
16 set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
17 set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
18 set_property PACKAGE_PIN V12 [get_ports {led[14]}]
19 set_property PACKAGE_PIN V14 [get_ports {led[13]}]
20 set_property PACKAGE_PIN V15 [get_ports {led[12]}]
21 set_property PACKAGE_PIN U14 [get_ports {led[10]}]
22 set_property PACKAGE_PIN T16 [get_ports {led[11]}]
23 set_property PACKAGE_PIN T15 [get_ports {led[9]}]
24 set_property PACKAGE_PIN V16 [get_ports {led[8]}]
25 set_property PACKAGE_PIN U16 [get_ports {led[7]}]
26 set_property PACKAGE_PIN U17 [get_ports {led[6]}]
27 set_property PACKAGE_PIN V17 [get_ports {led[5]}]
28 set_property PACKAGE_PIN R18 [get_ports {led[4]}]
29 set_property PACKAGE_PIN N14 [get_ports {led[3]}]
30 set_property PACKAGE_PIN J13 [get_ports {led[2]}]
31 set_property PACKAGE_PIN K15 [get_ports {led[1]}]
32 set_property PACKAGE_PIN H17 [get_ports {led[0]}]
33 set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
```

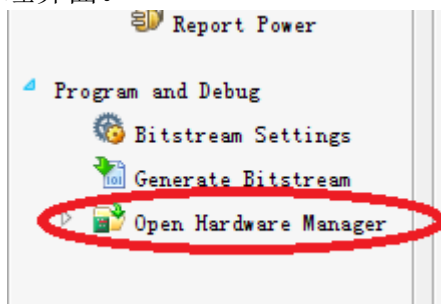
## 七、 下载程序

- 1、在Flow Navigator中点击Program and Debug下的Generate Bitstream选项，工程会自动完成综合、实现、Bit文件生成过程，完成之后，可点击Open Implemented Design来查看工程实现结果。

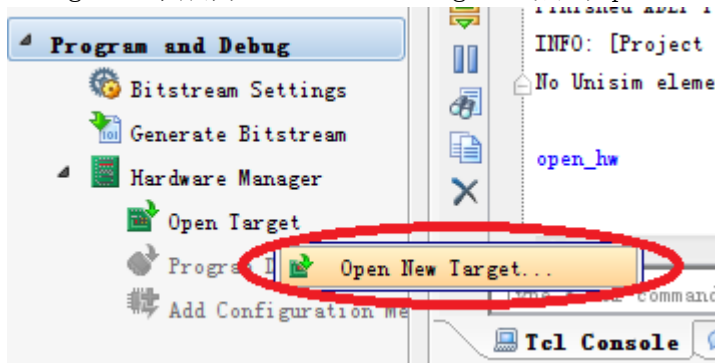




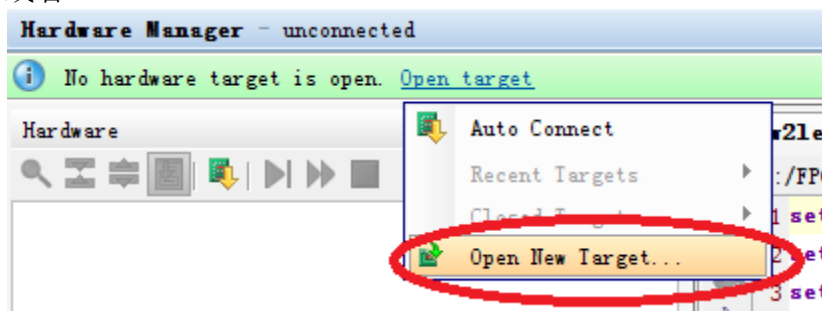
- 2、点击Flow Navigator中Open Hardware Manager一项，进入硬件编程管理界面。



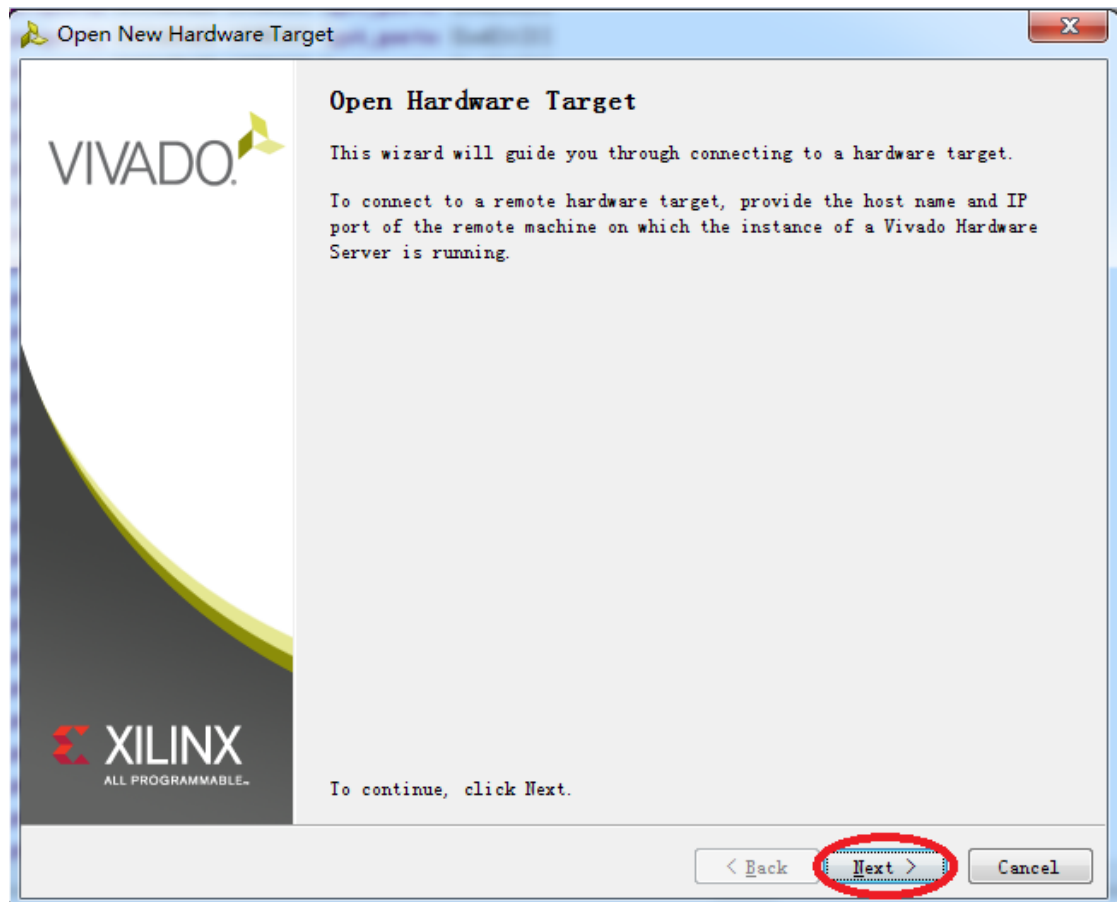
- 3、在提示的信息中，选择Open a new hardware target（或在Flow Navigator中展开Hardware Manager，点击Open New Target）。



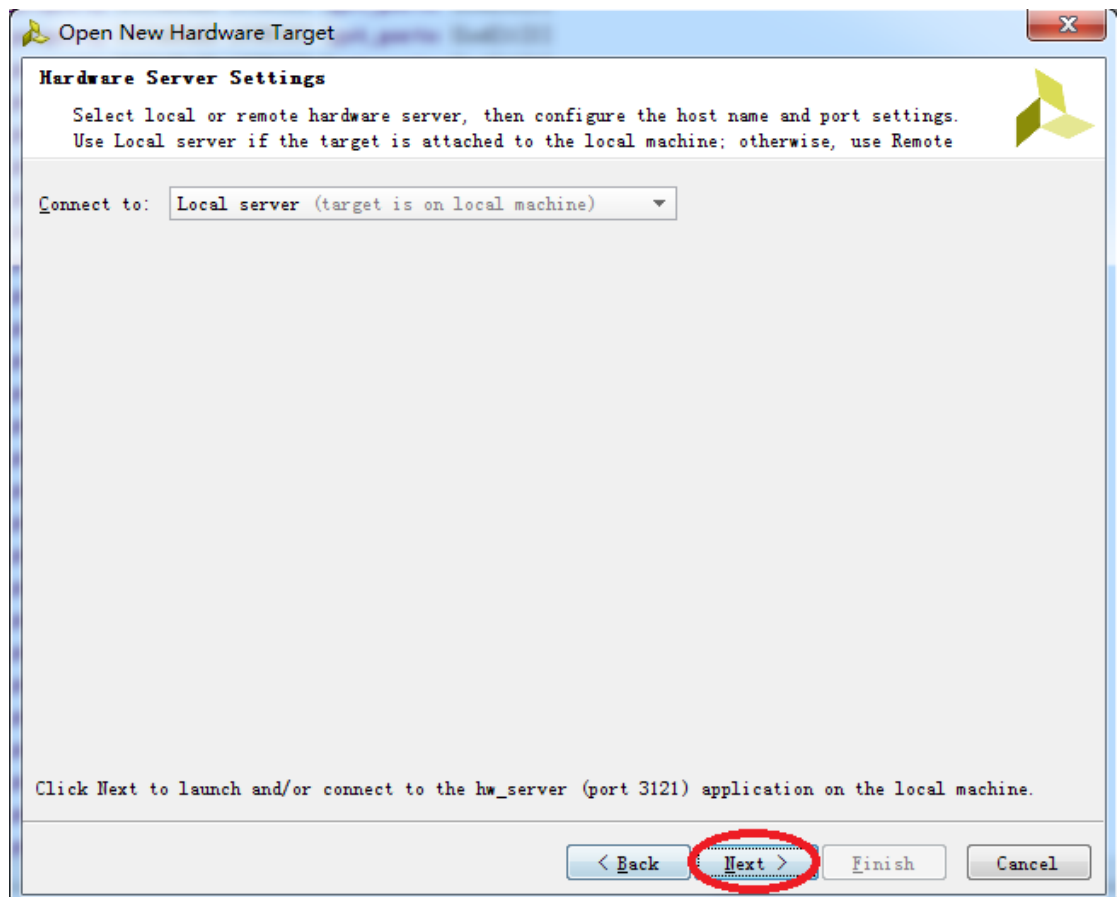
或者



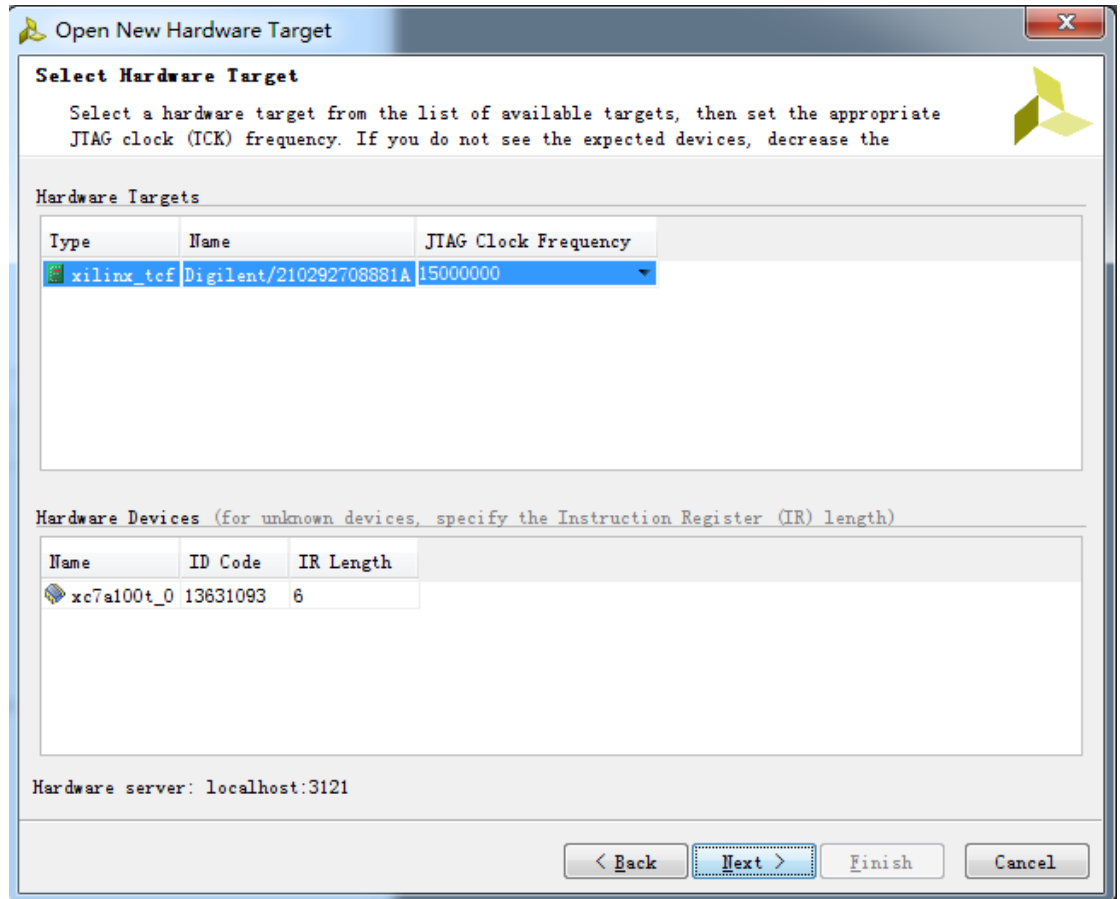
- 4、在弹出的Open hardware target向导中，先点击Next，进入Server选择向导。

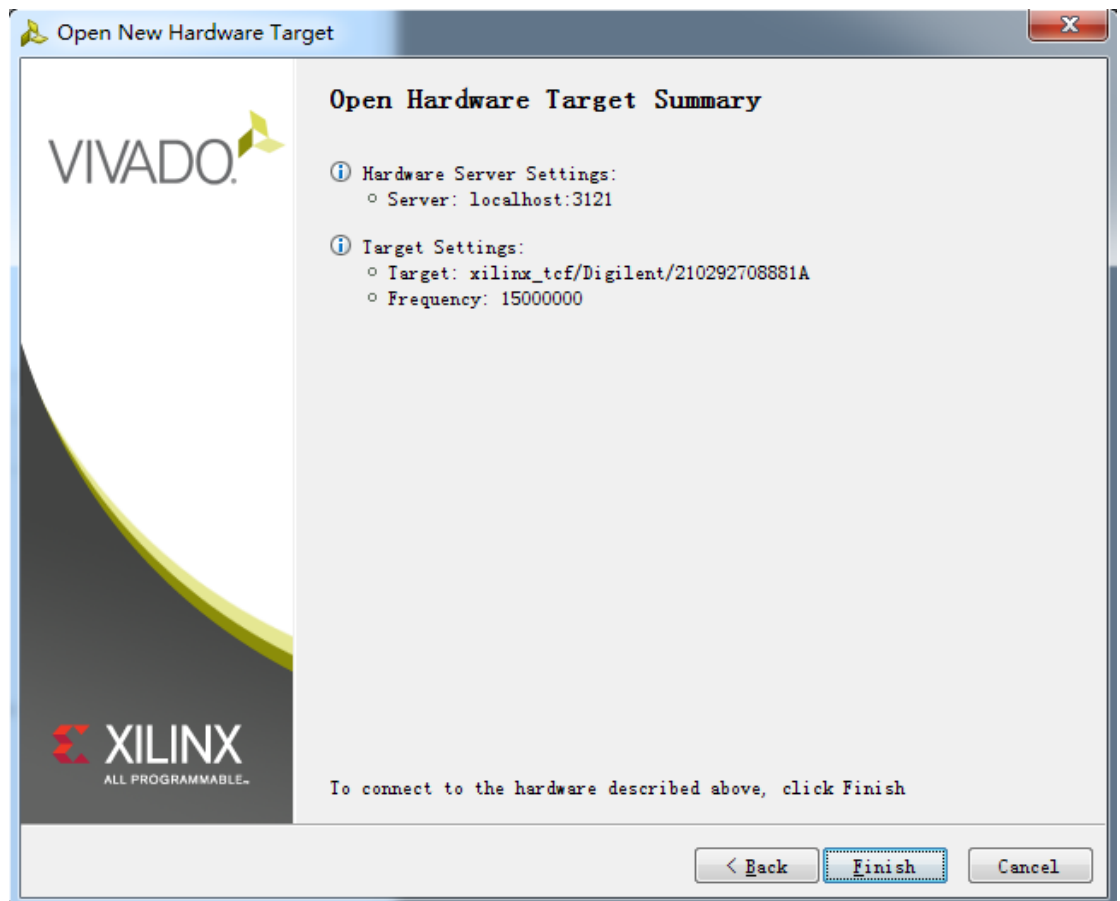


- 5、保持默认的Server name为localhost, 连接好板卡的PROG端口, 并上电。点击Next。注意：此时，电脑上不要使用其他USB设备以防查找硬件出错。如出错了，只有重启软件，或者重启电脑（软件的BUG。。。）



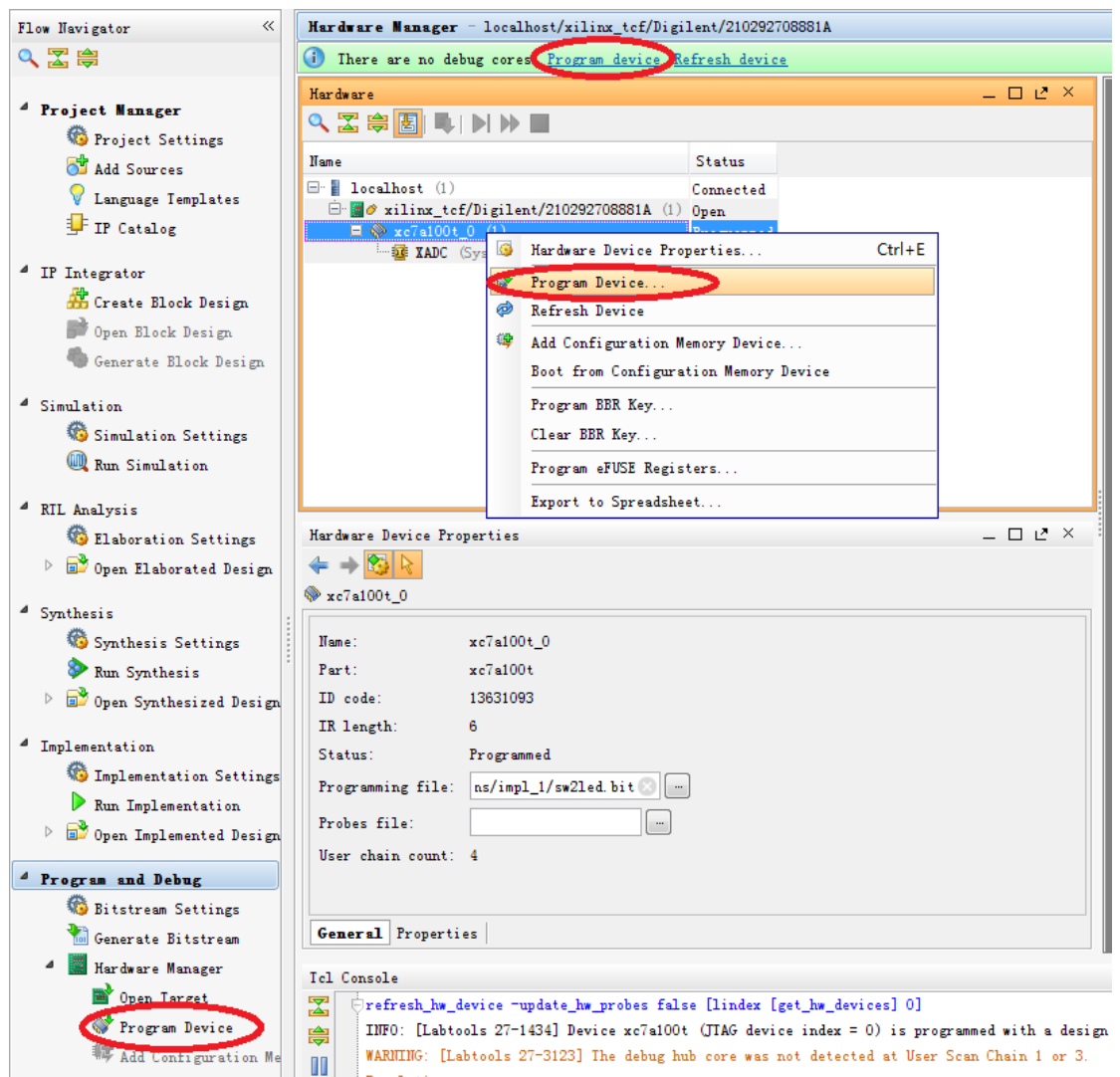
6、依次点击Next、Finish完成新建Hardware Target。





- 7、此时，Hardware一栏中出现硬件平台上可编程的器件。在对应的FPGA器件上右击，选择Program Device，或在Flow Navigator下的Hardware Manager中点击Program Device一项。





- 8、指定所需的bit文件（系统默认已存在该工程的bit文件，如不需更改，点击Program）。



- 9、设计完成，即可观察开发板运行情况。