

# zynq 7000 的流水灯实验

原创

leon\_zeng0 于 2018-08-31 03:14:39 发布 2378 收藏 15

分类专栏: [fpga zynq arm](#) 文章标签: [zynq vivado en](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/leon\\_zeng0/article/details/82230372](https://blog.csdn.net/leon_zeng0/article/details/82230372)

版权



[fpga](#) 同时被 3 个专栏收录

49 篇文章 13 订阅

订阅专栏



[zynq](#)

48 篇文章 10 订阅

订阅专栏



[arm](#)

20 篇文章 0 订阅

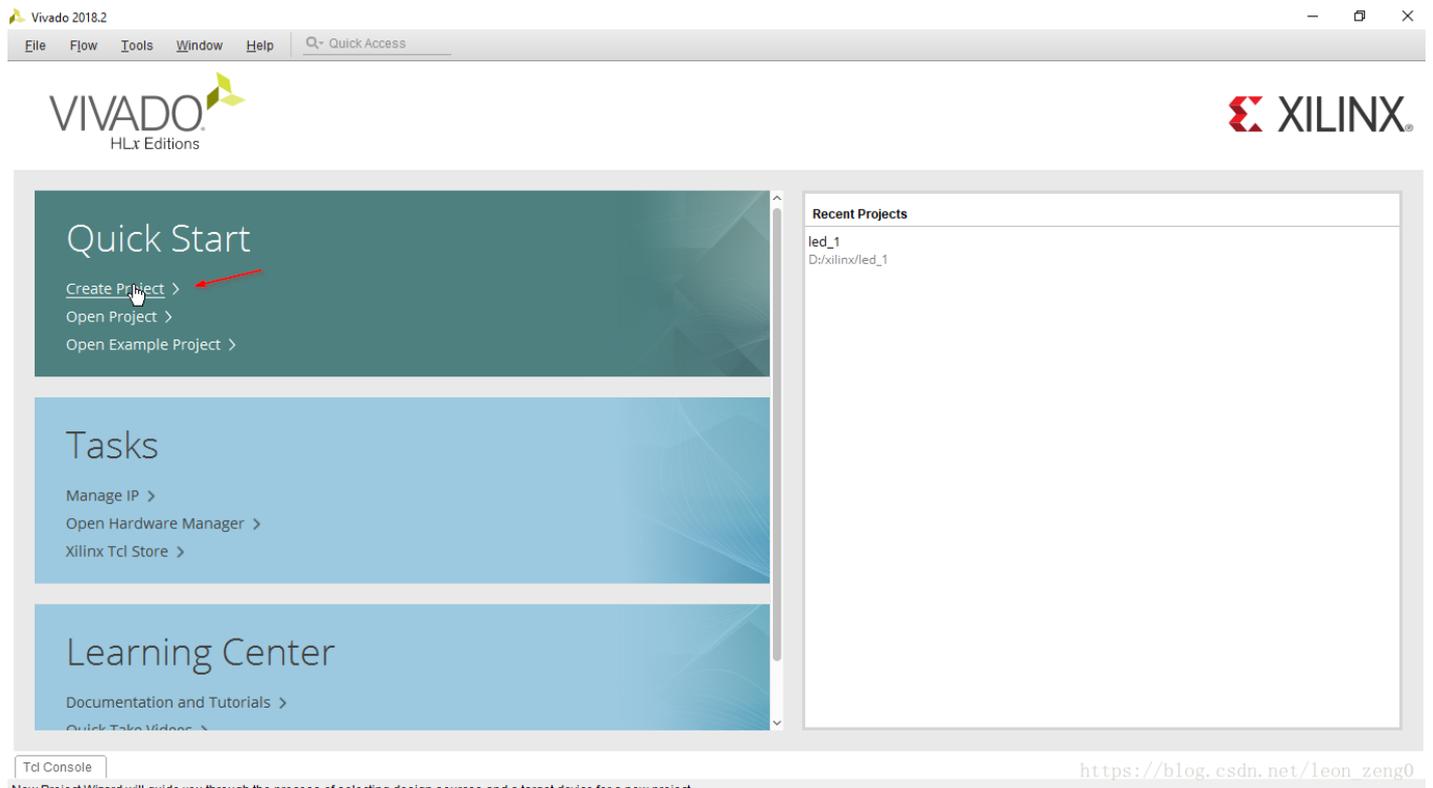
订阅专栏

流水灯实验是FPGA 的入门实验。最简单并且得到实在的体验, 我们就从这里开始zynq 7000的体验吧。

本实验包括3部分: 创建工程, 添加和建立文件, 比特流产生和编程。

1: 创建工程: 打开Vivado, 在如下界面里选择建立工程, Create Project。

由于不同的vivado的版本, 可能界面不一, 但都是找到Create Project。



出现create project 界面时，选择Next。在如下界面输入工程的名字，这里是run\_led，同时也要选择工程的存放文件夹。

New Project

**Project Name**  
Enter a name for your project and specify a directory where the project data files will be stored.

Project name: run\_led

Project location: D:/xilinx

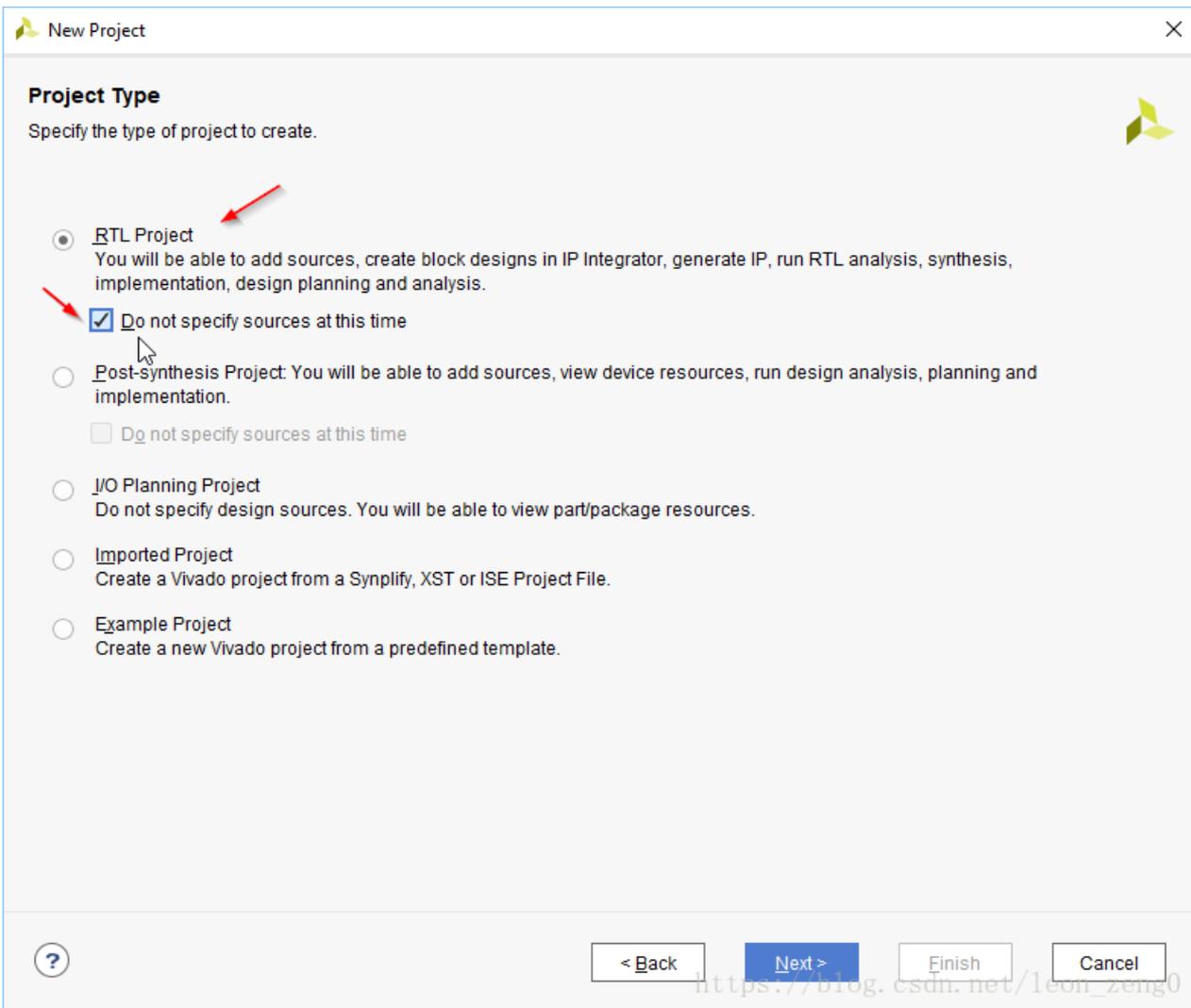
Create project subdirectory

Project will be created at: D:/xilinx/run\_led

? < Back Next > Finish Cancel

[https://blog.csdn.net/leon\\_zeng0](https://blog.csdn.net/leon_zeng0)

到如下界面时，选择RTL Project，勾取Do not specify sources at this time。



到这个界面，选择Part，太多了，不好选择，可以指定一些条件，我们这里选择系列为Zynq-7000。

**New Project**

### Default Part

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

**Parts** | Boards

[Reset All Filters](#)

Category: All Package: All Temperature: All  
Family: All Speed: All

Search:

Part	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7z007s	14400	28800	50	0	66
xc7z007s	14400	28800	50	0	66
xc7z010clg400-2	17600	35200	60	0	80
xc7z010clg400-1	17600	35200	60	0	80
xc7z010clg225-1L	17600	35200	60	0	80
xc7z010clg400-1L	17600	35200	60	0	80

< Back   Next >   Finish   Cancel

[https://blog.csdn.net/leon\\_zeng0](https://blog.csdn.net/leon_zeng0)

再选择封装为clg400，就看到了我的Part， 我的是:xc7z010clg400-1。不同的板子这里可能不同，你要查询好你的Part 型号。

New Project

### Default Part

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

Parts | Boards

Reset All Filters

Category: All Package: clg400 Temperature: All Remaining  
Family: Zynq-7000 Speed: All Remaining

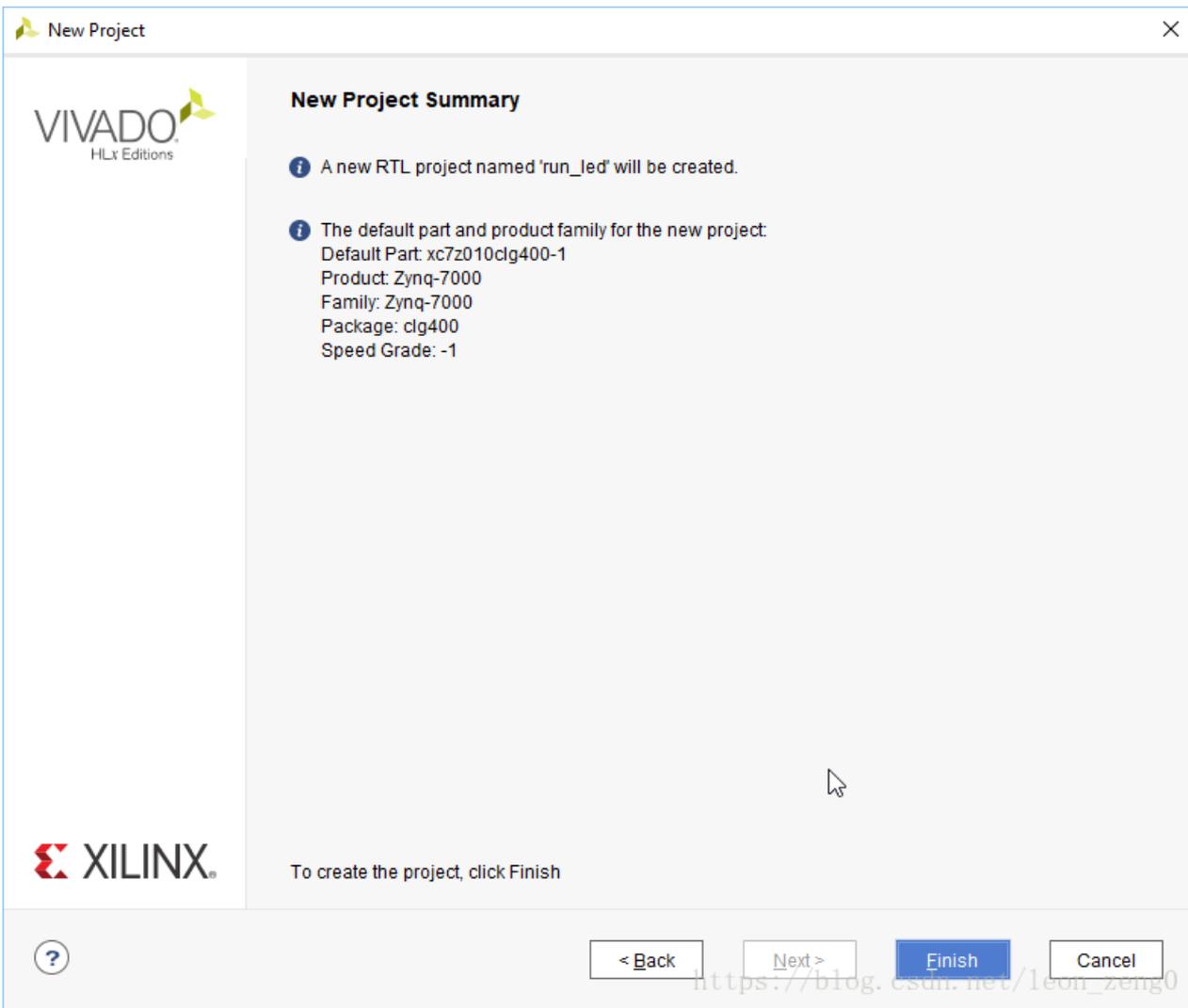
Search: Q

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs	Ultra RAMs	DSPs
xc7z007sclg400-2	400	100	14400	28800	50	0	66
xc7z007sclg400-1	400	100	14400	28800	50	0	66
xc7z010clg400-3	400	100	17600	35200	60	0	80
xc7z010clg400-2	400	100	17600	35200	60	0	80
xc7z010clg400-1	400	100	17600	35200	60	0	80
xc7z010iclg400-1L	400	100	17600	35200	60	0	80
xc7z014sclg400-2	400	125	40600	81200	107	0	170
xc7z014sclg400-1	400	125	40600	81200	107	0	170
xc7z020clg400-3	400	125	53200	106400	140	0	220
xc7z020clg400-2	400	125	53200	106400	140	0	220

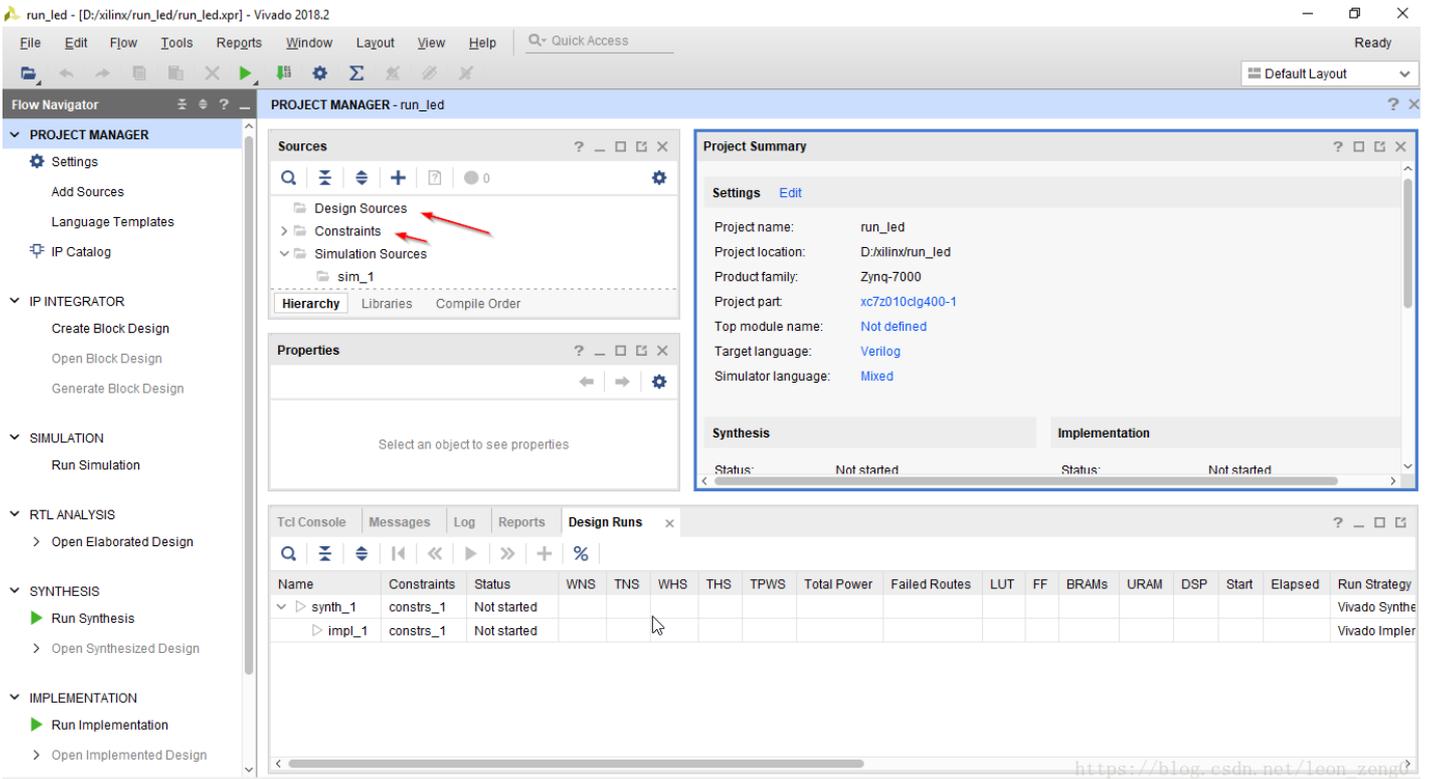
? < Back Next > Finish Cancel

[https://blog.csdn.net/leon\\_zeng0](https://blog.csdn.net/leon_zeng0)

到这里是个汇总，有不对的，可以返回选择。

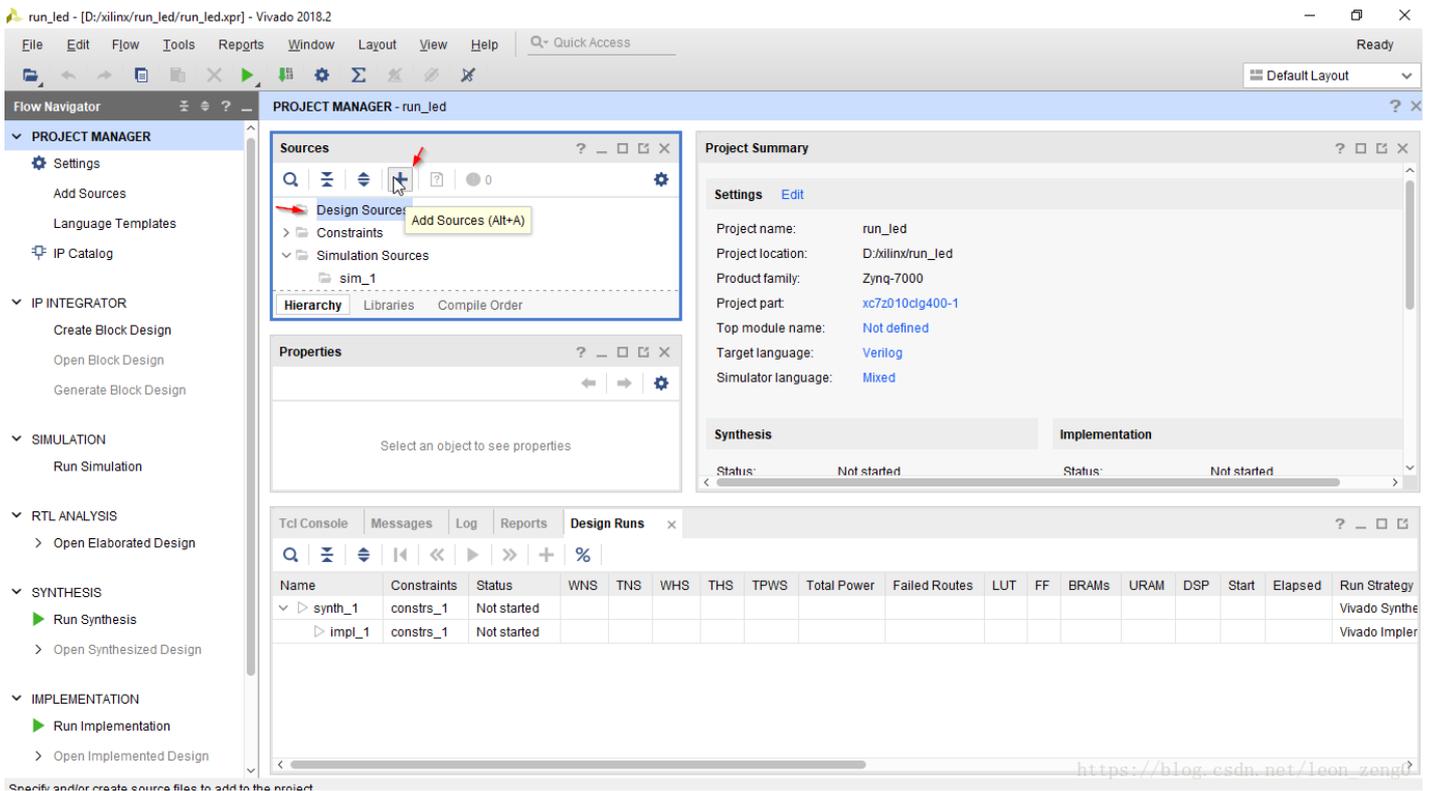


工程创建完了，就是这个样子。看看图中2个红箭，下面我们要添加2个文件在这里。

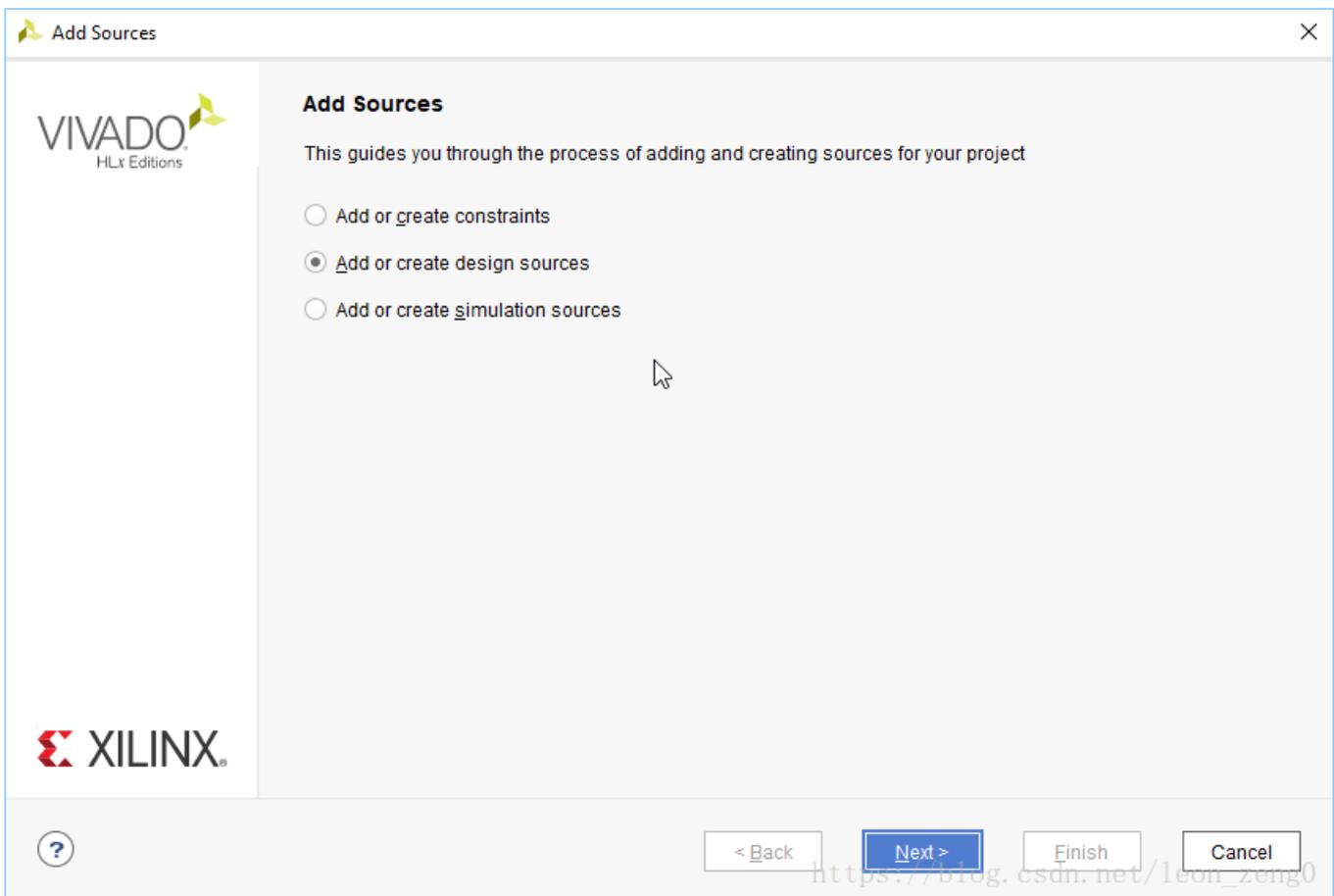


2: 添加或创建文件:

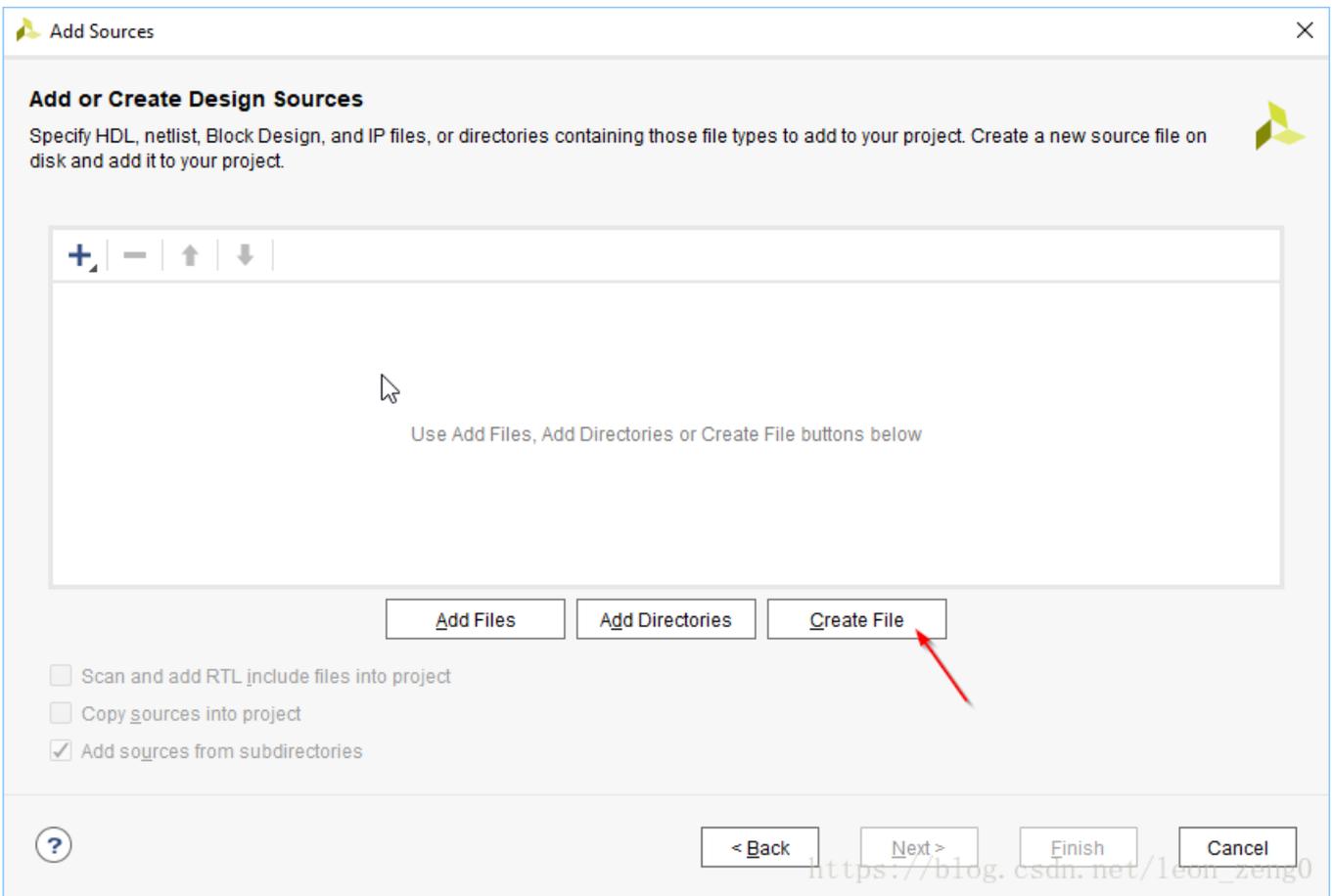
在工程管理界面里，添加设计源码文件。先选择Design Source，然后右键或下图中+按钮，Add Source。



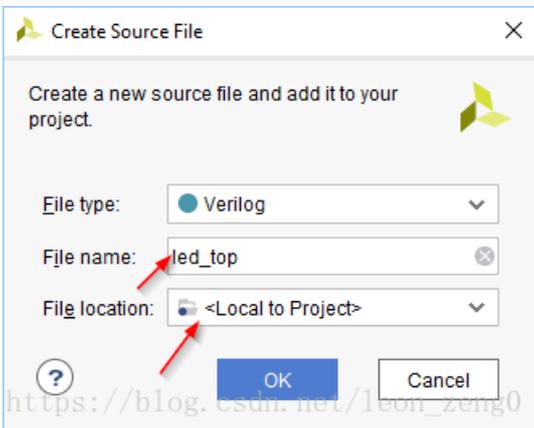
这里选择添加文件类型，这里是 Add or create design sources。



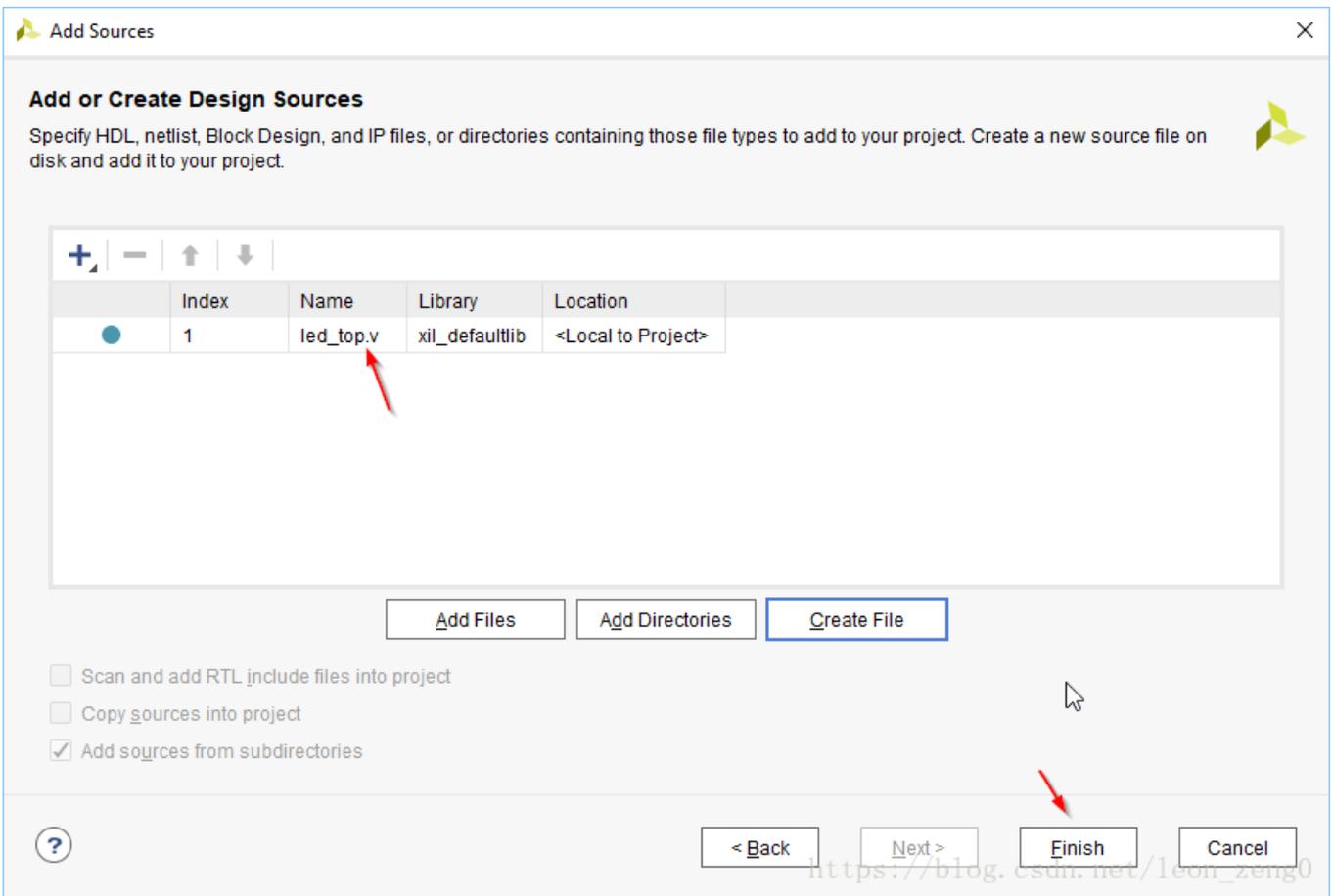
选择 Next 后到如下界面，可以添加或者创建文件，这里选择Create File



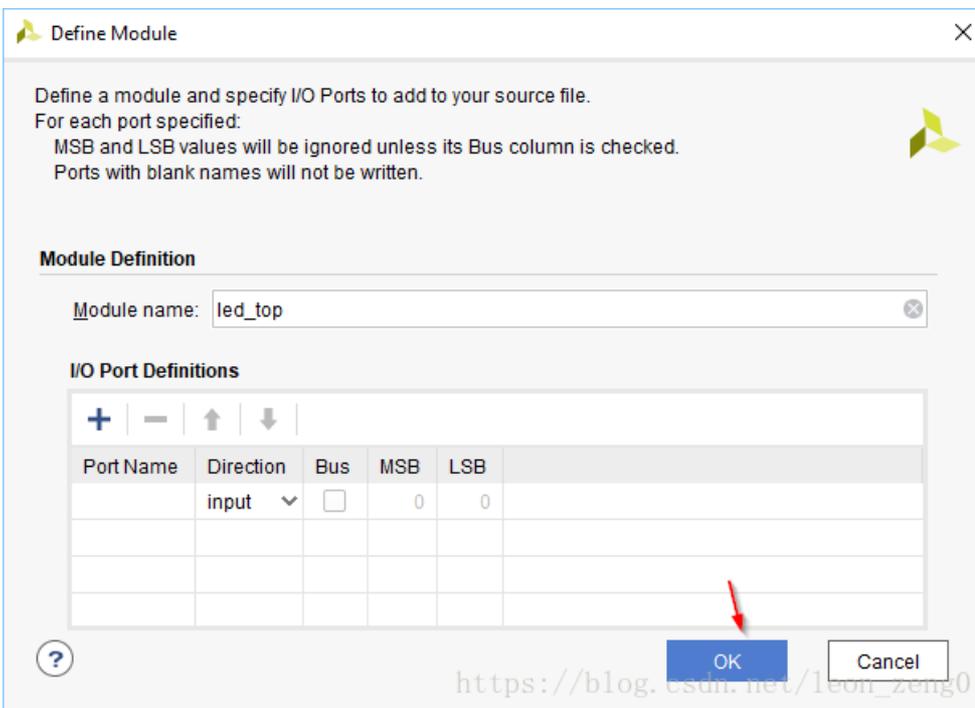
出现对话框，这里输入文件名led\_top，还有文件位置。



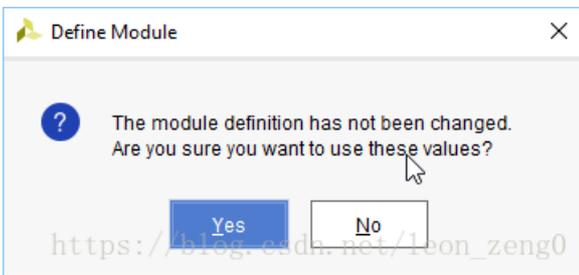
OK 之后，回到如下界面。文件已在列表中。



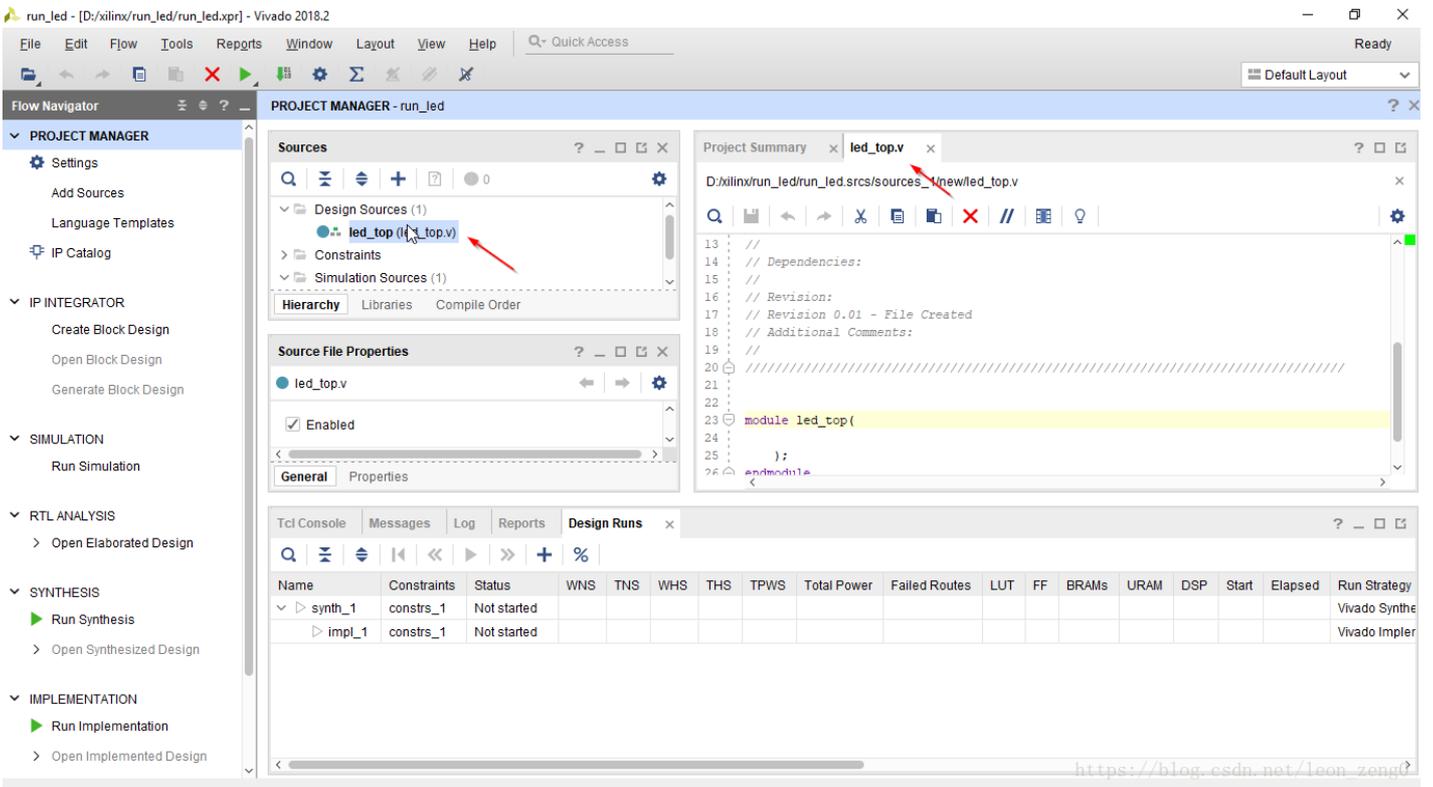
选择Finish 后，出现如下界面，OK就好了。



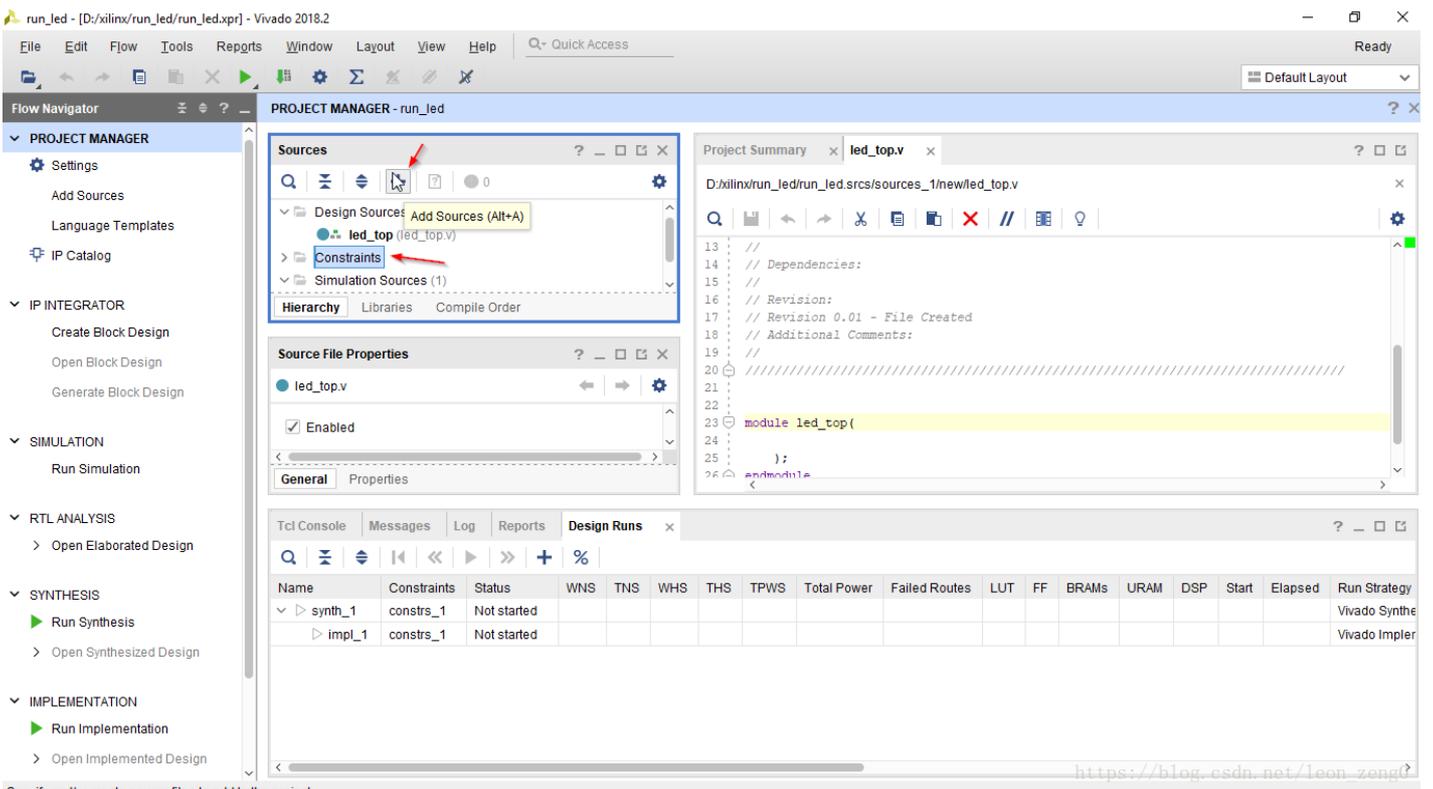
还有这一个对话框，Yes。



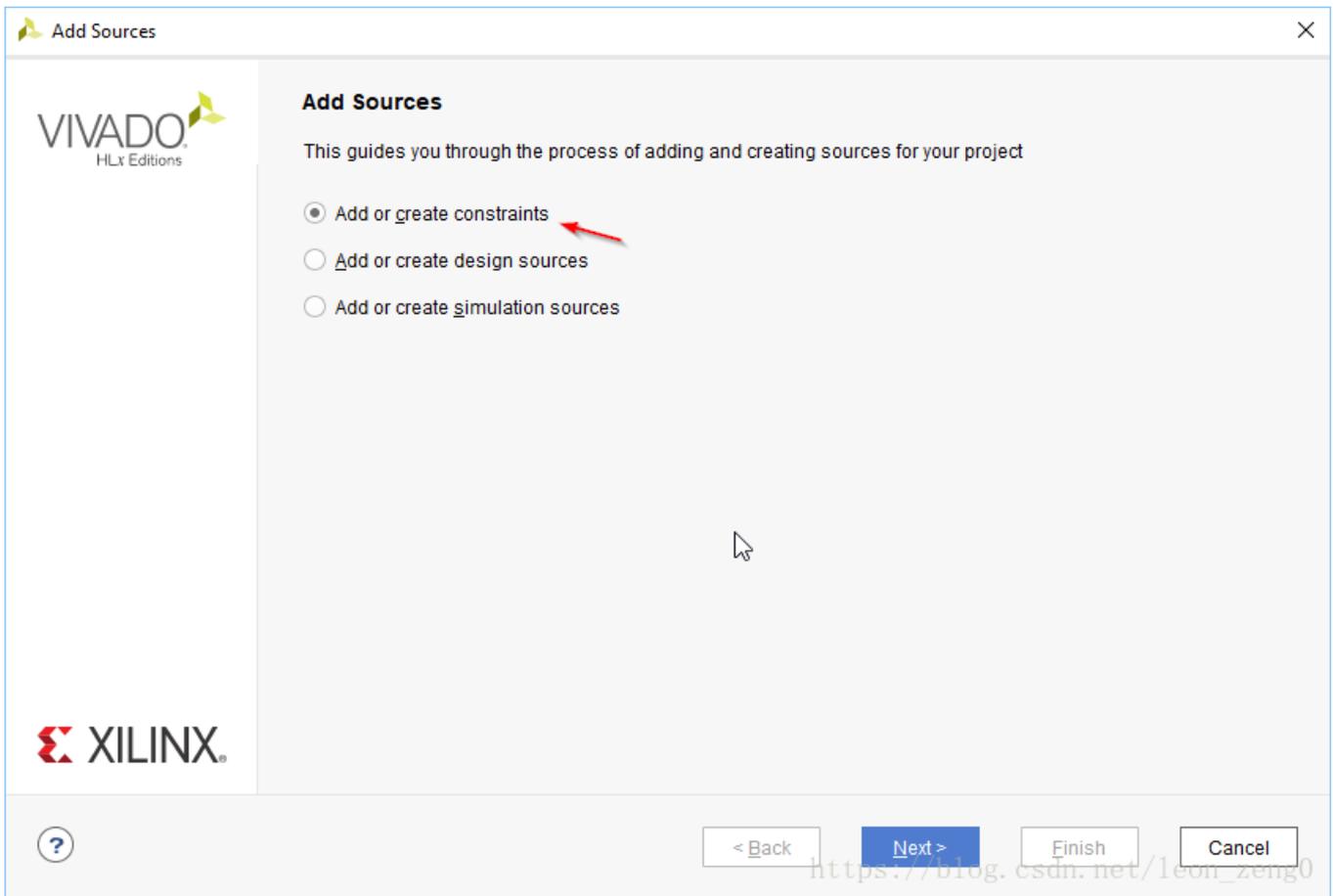
回到管理界面了。Design Sources 下有了文件，点一下，右边就会显示出来，可以编辑修改。



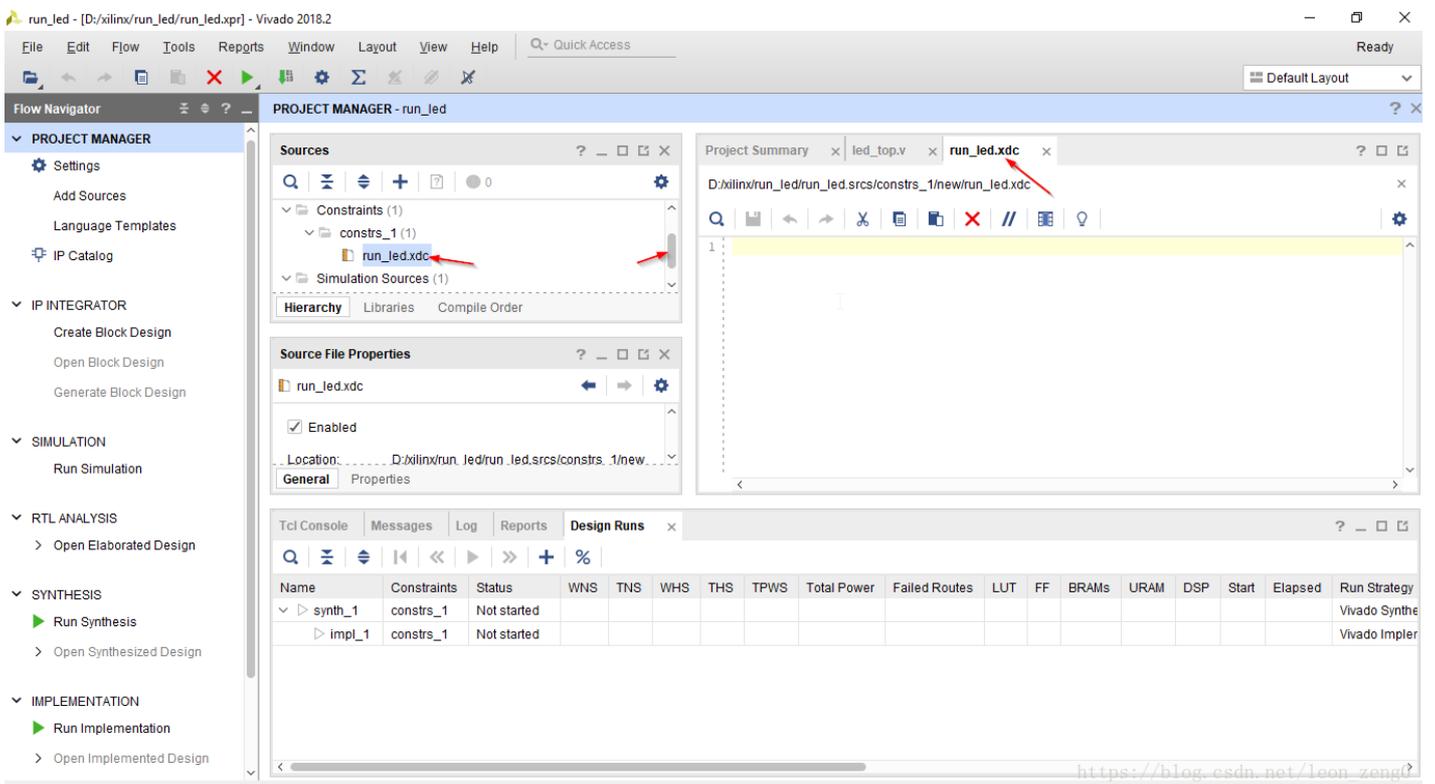
在约束下还要添加文件。点一下 Constraints，与上面添加文件一样操作。右键或+按钮，Add Source。



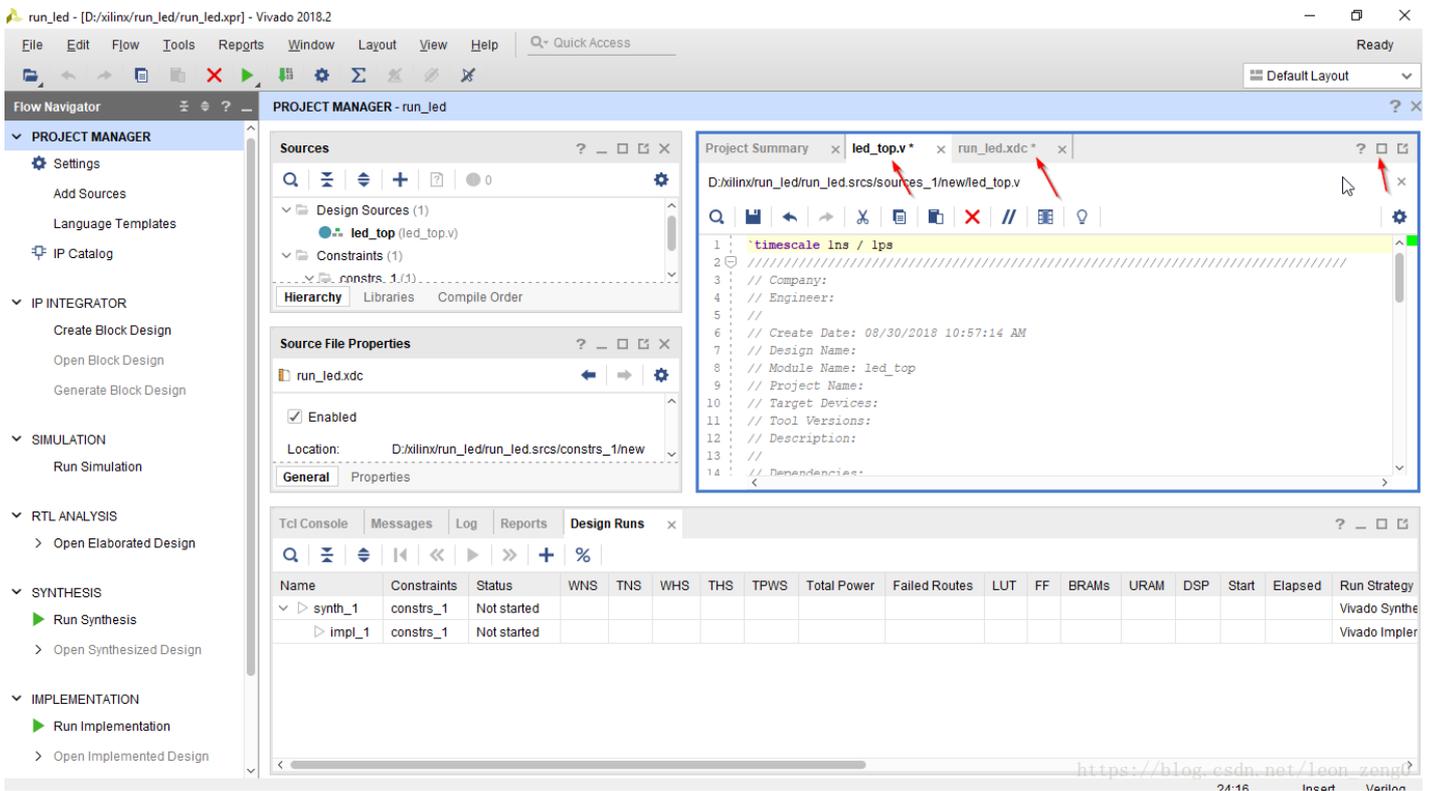
这里文件类型选择是 Add or create constraints。



约束文件取名run\_led，点开可以右边显示。



看看我们创建的2个文件都显示在右边。右上可以最大化。



文件都建立好了，现在要修改文件的内容了。

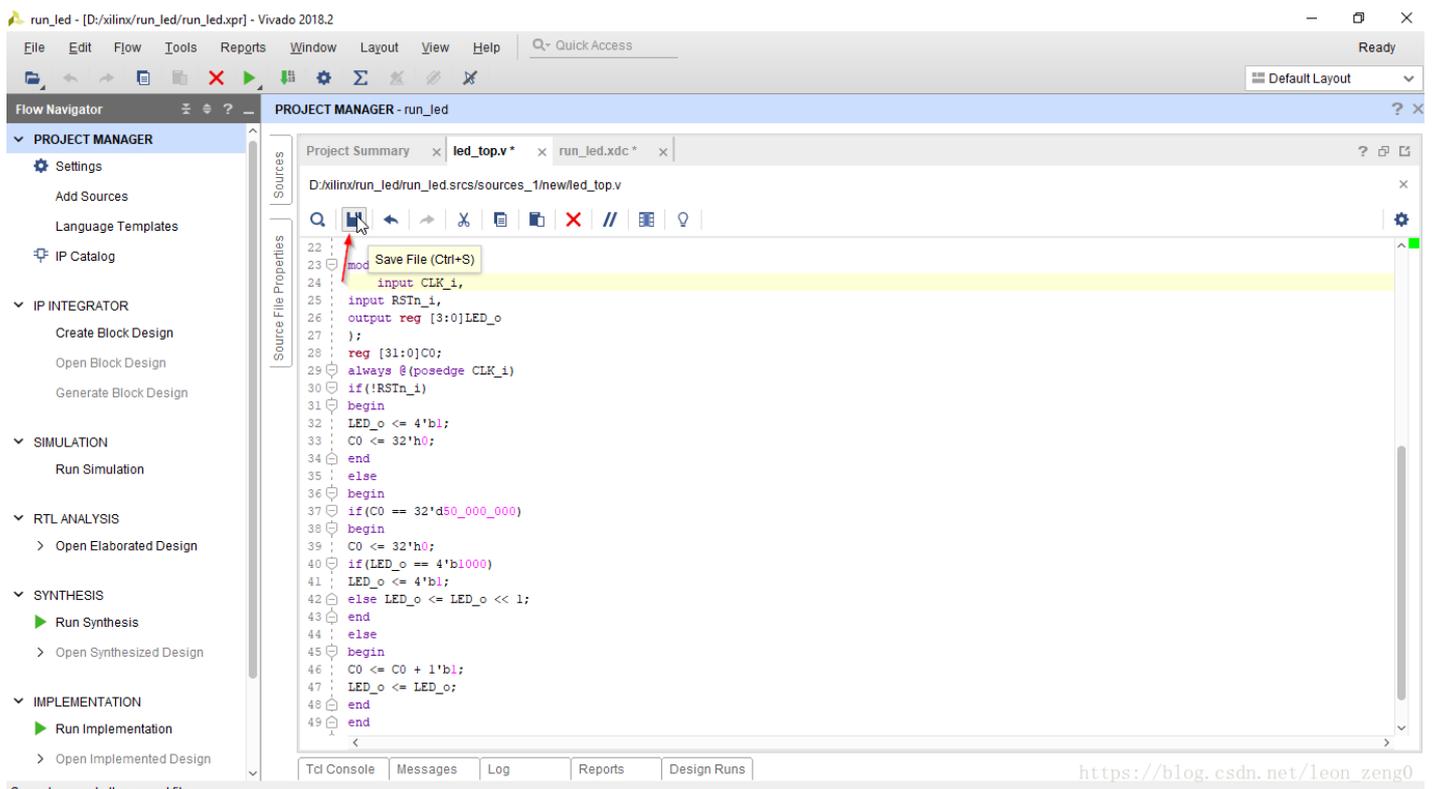
led\_top.v文件内容如下：

```

module led_top(
    input CLK_i,
input RSTn_i,
output reg [3:0]LED_o
);
reg [31:0]C0;
always @(posedge CLK_i)
if(!RSTn_i)
begin
LED_o <= 4'b1;
C0 <= 32'h0;
end
else
begin
if(C0 == 32'd50_000_000)
begin
C0 <= 32'h0;
if(LED_o == 4'b1000)
LED_o <= 4'b1;
else LED_o <= LED_o << 1;
end
else
begin
C0 <= C0 + 1'b1;
LED_o <= LED_o;
end
end
endmodule

```

你可以这里复制，粘贴过去。显示如下，多了一些注释之类的。



run\_led.xdc 的内容与硬件相关。在这里要注意你的硬件管脚对应关系。我这里是米联 MZ7XA 的硬件定义。

```
CLK_i H16
```

```
T19 RSTn_i
```

```
J16 LED_o[0]
```

```
K16 LED_o[1]
```

```
H15 LED_o[2]
```

```
G15 LED_o[3]
```

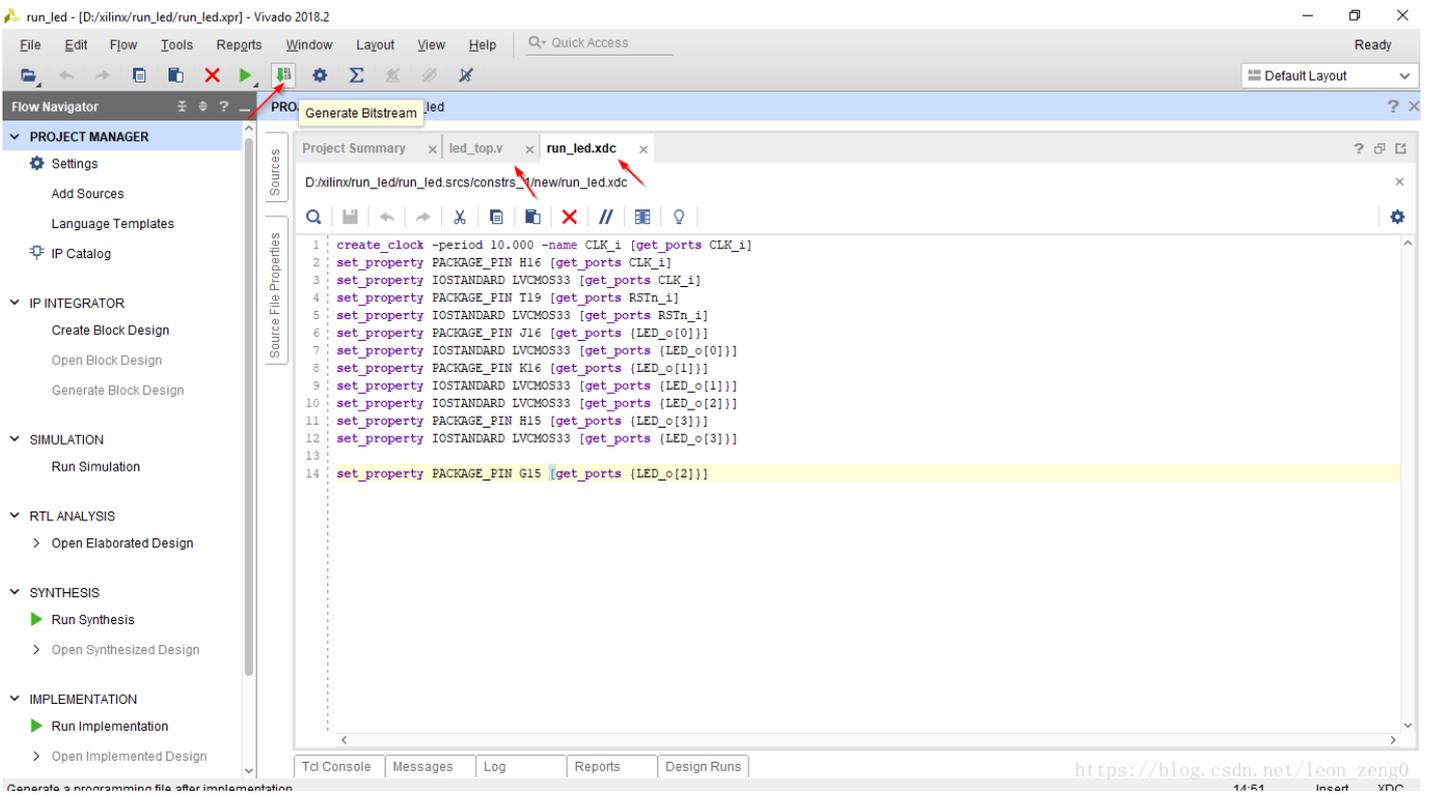
run\_led.xdc 的内容如下，复制粘贴过去。如果管脚对应不一样，修改这个内容。

```
create_clock -period 10.000 -name CLK_i [get_ports CLK_i]
set_property PACKAGE_PIN H16 [get_ports CLK_i]
set_property IOSTANDARD LVCMOS33 [get_ports CLK_i]
set_property PACKAGE_PIN T19 [get_ports RSTn_i]
set_property IOSTANDARD LVCMOS33 [get_ports RSTn_i]
set_property PACKAGE_PIN J16 [get_ports {LED_o[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_o[0]}]
set_property PACKAGE_PIN K16 [get_ports {LED_o[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_o[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_o[2]}]
set_property PACKAGE_PIN H15 [get_ports {LED_o[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LED_o[3]}]
set_property PACKAGE_PIN G15 [get_ports {LED_o[2]}]
```

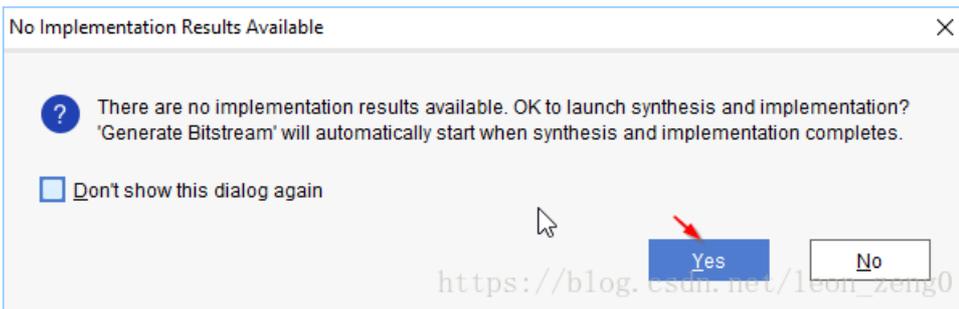
文件内容修改后，记得保存。

### 3: 比特流产生和编程运行

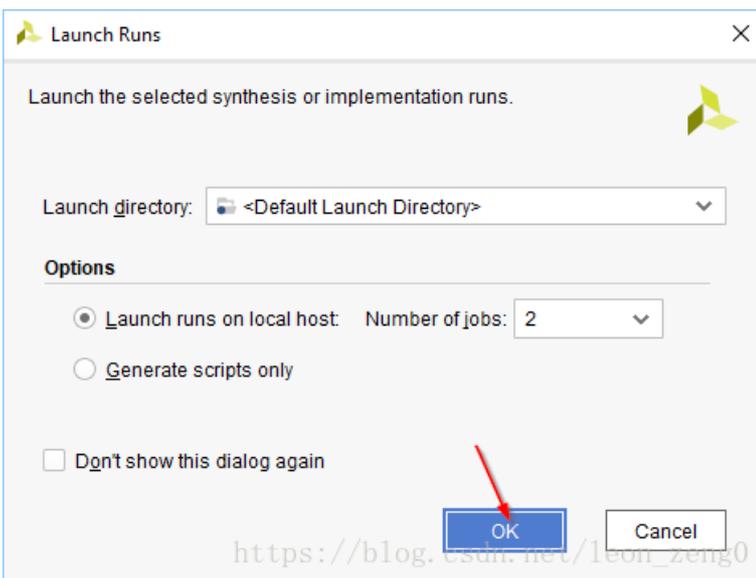
文件内容都做好了，保存文件，然后点击Generate BitStream。



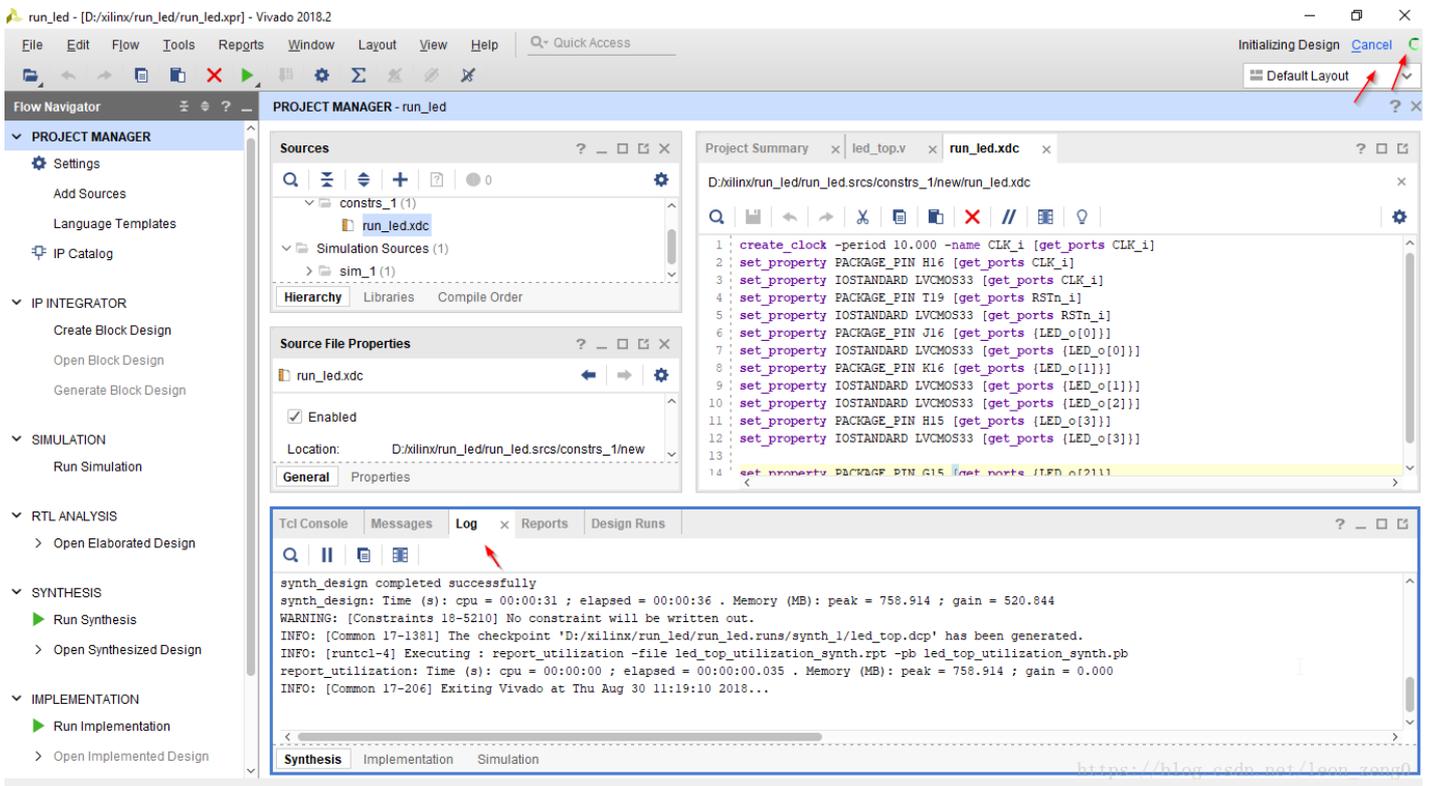
他会产生提示对话框，Yes就好。



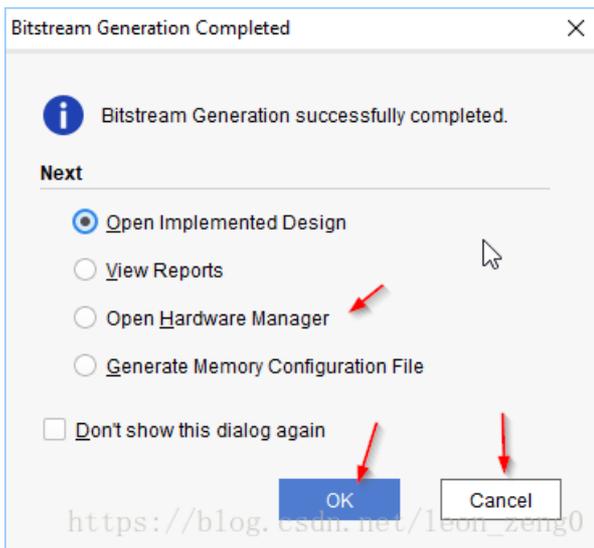
还出现这个，OK



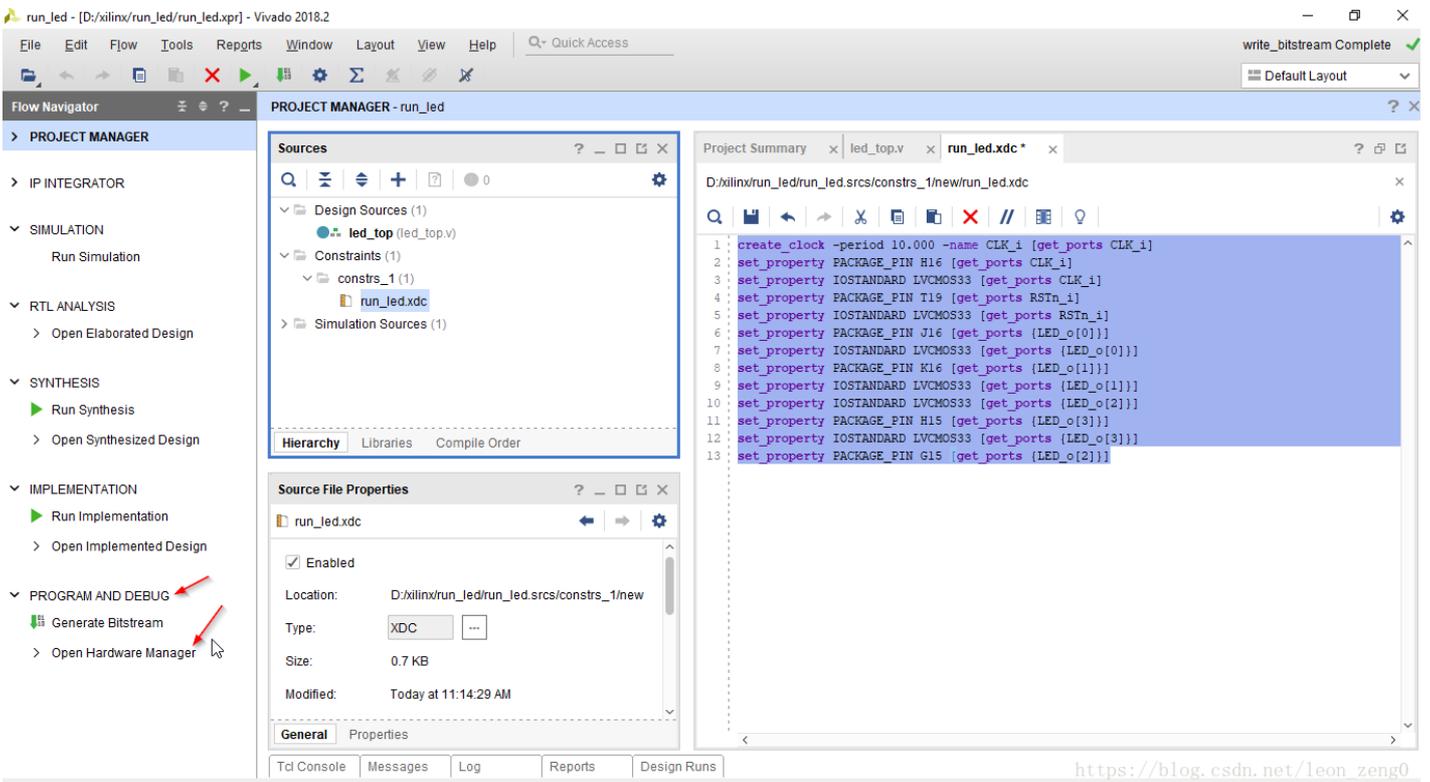
这个需要些时间，你可以看到右上在旋转，表示在计算，看看日志在更新。



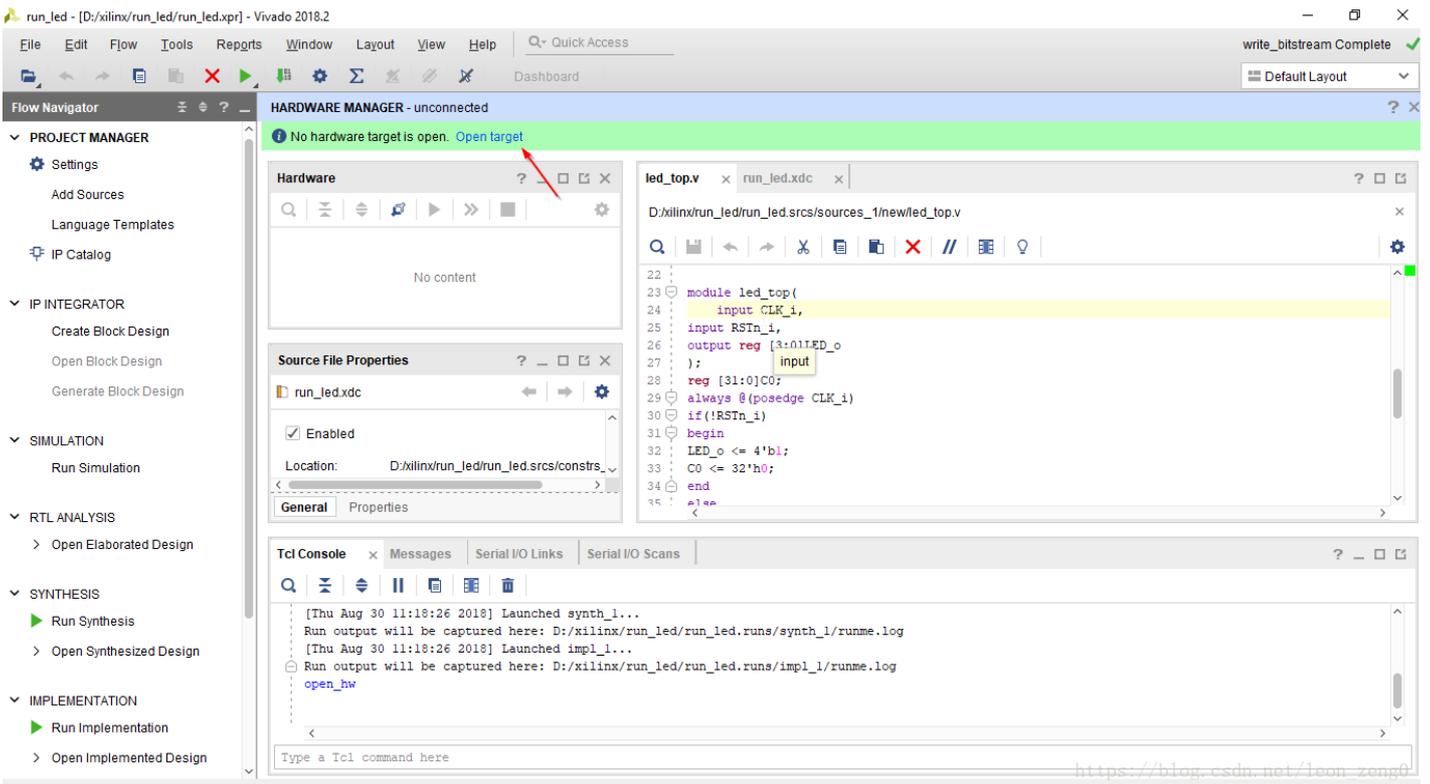
运行结束后会有如下完成对话框。这是告诉你结束了，下面怎么做，可以Cancel，也可以选择一个下面的内容。这里选择红箭指示的 Open Hardware manager，完成了，我就打开硬件管理器。



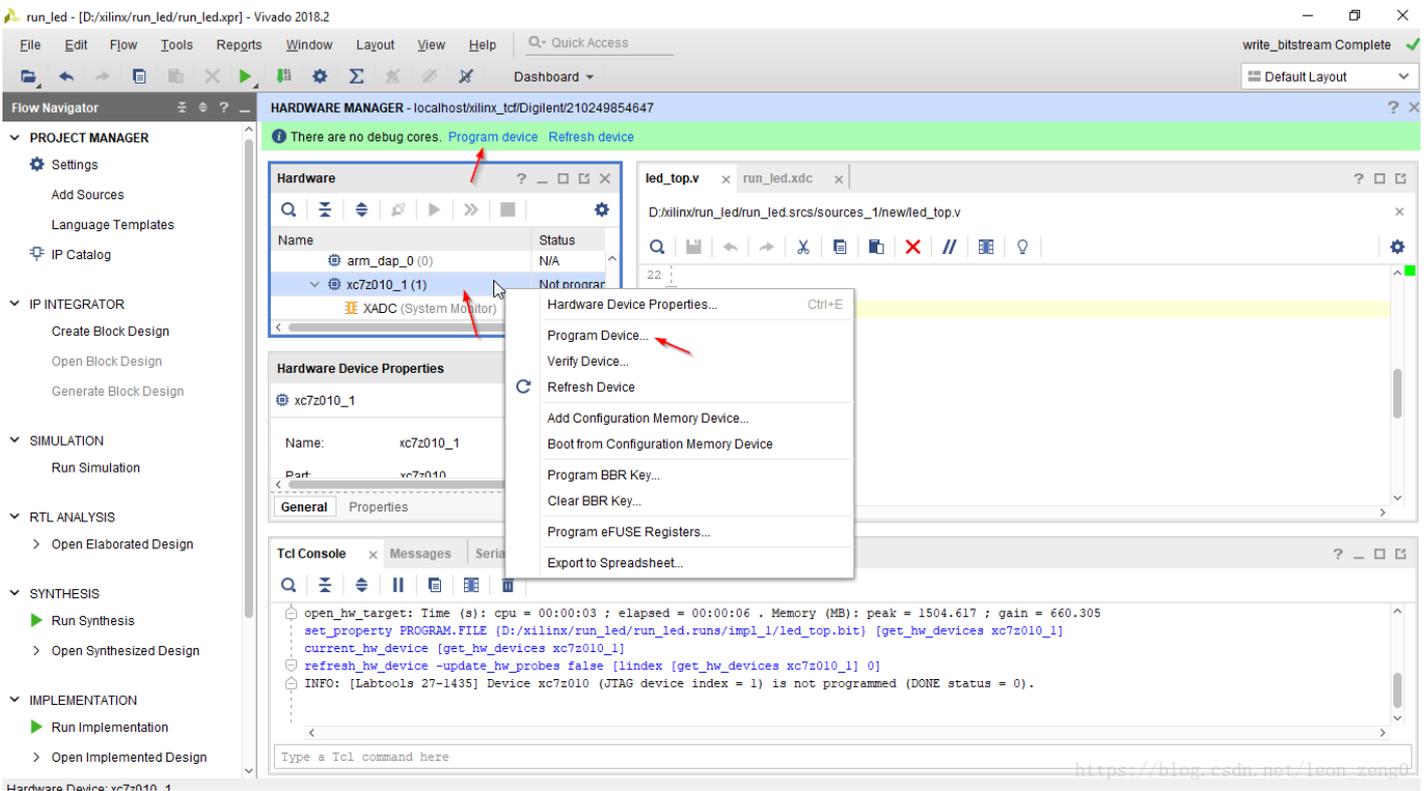
如果你选择了Cancel或者没有选择打开硬件管理，也可以如下图打开硬件管理界面。Flow Navigator->Program and debug->Open Hardware Manager。还可以菜单Flow -> Open Hardware Manager。



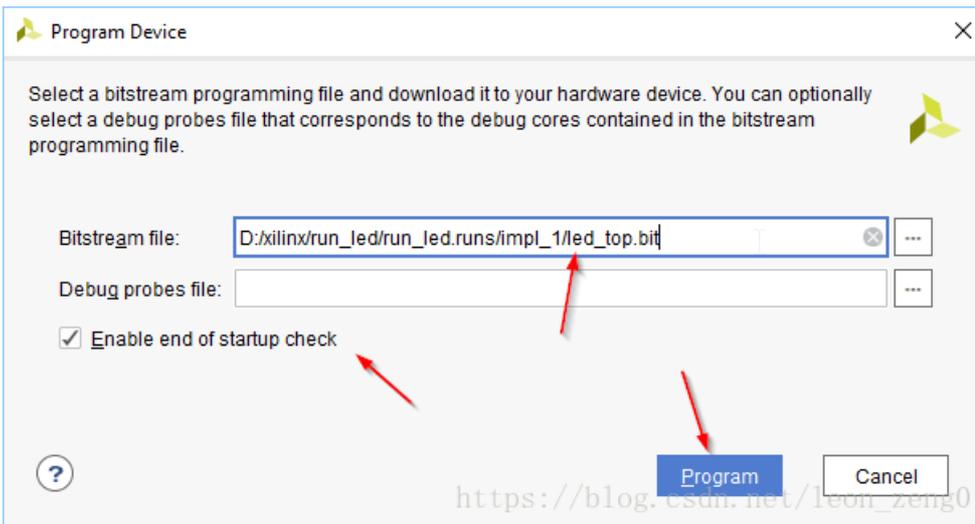
硬件管理面板如下。这里说没有连接硬件，你要连接好你的硬件，然后Open target



现在硬件连接上了，但还没有编程设备。按照红箭的位置，点击 Program Devices。



选择比特率文件，如果没有自动输入，那你手动选择，记得下图文件位置和名字。



点击Program，程序就下载并运行了。实验也就结束了。



[创作打卡挑战赛](#)  
[赢取流量/现金/CSDN周边激励大奖](#)