

DIGILENT Basys 3 FPGA 基于 Vivado 上手教程 (一) Vivado 设计流程

Vivado 设计流程

一、 实验目的

通过 LED 流水灯设计实验了解 Vivado 设计流程，熟悉 Basys 3 的外设接口及设置。

二、 实验内容

- 创建一个新的 Vivado 工程项目
- 编写 LED 流水灯的 Verilog 源文件
- 建立仿真
- 添加引脚约束文件
- 编写 testbench 进行激励仿真测试
- 添加时序约束
- 在 Basys 3 FPGA 开发板上运行实验设计

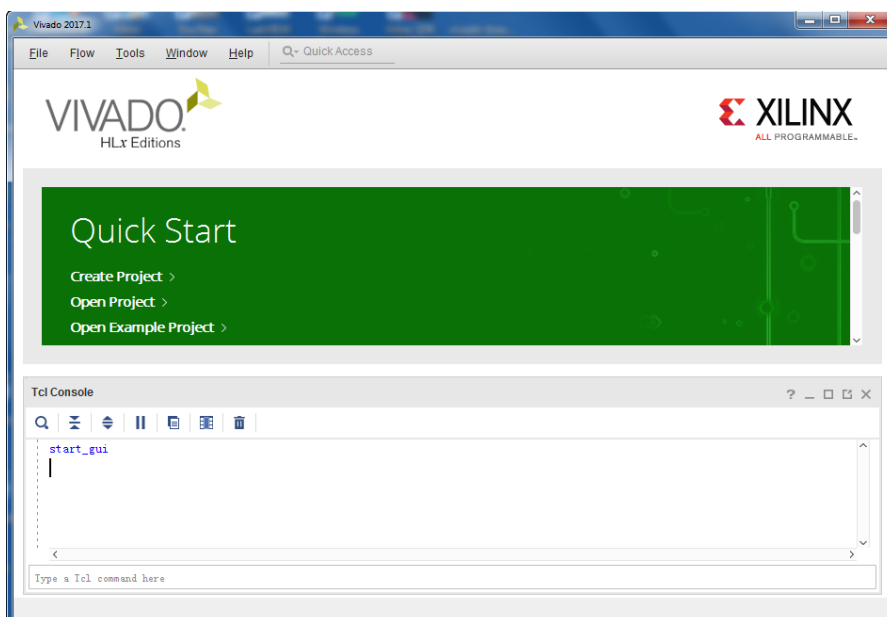
三、 实验步骤

1. 创建新的工程项目

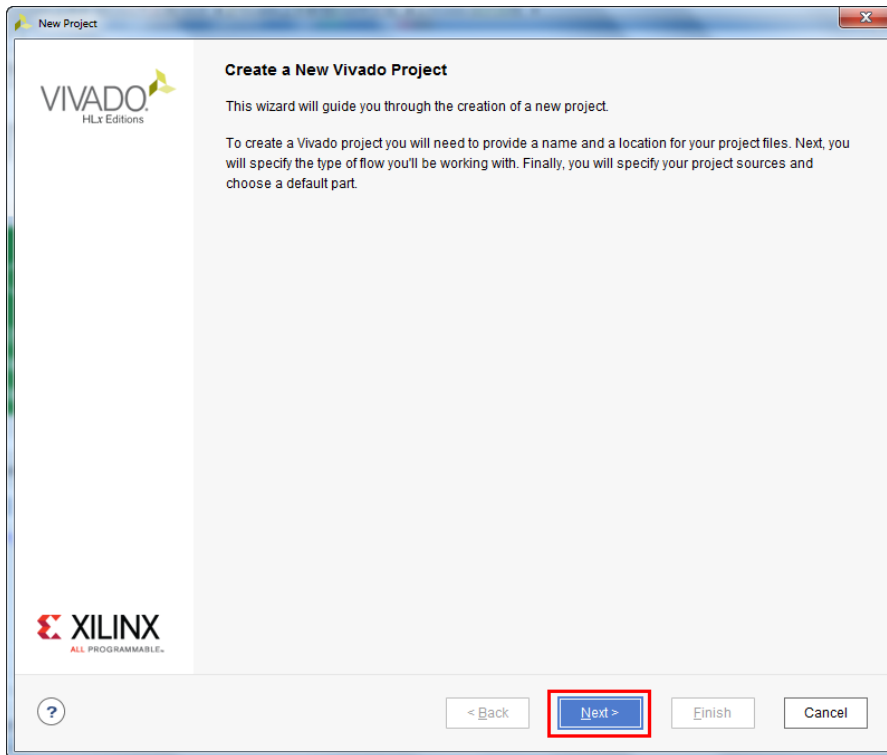
- 1) 双击桌面图标打开 Vivado 2017.1，或者选择开始>所有程序>Xilinx Design Tools> Vivado 2017.1>Vivado 2017.1



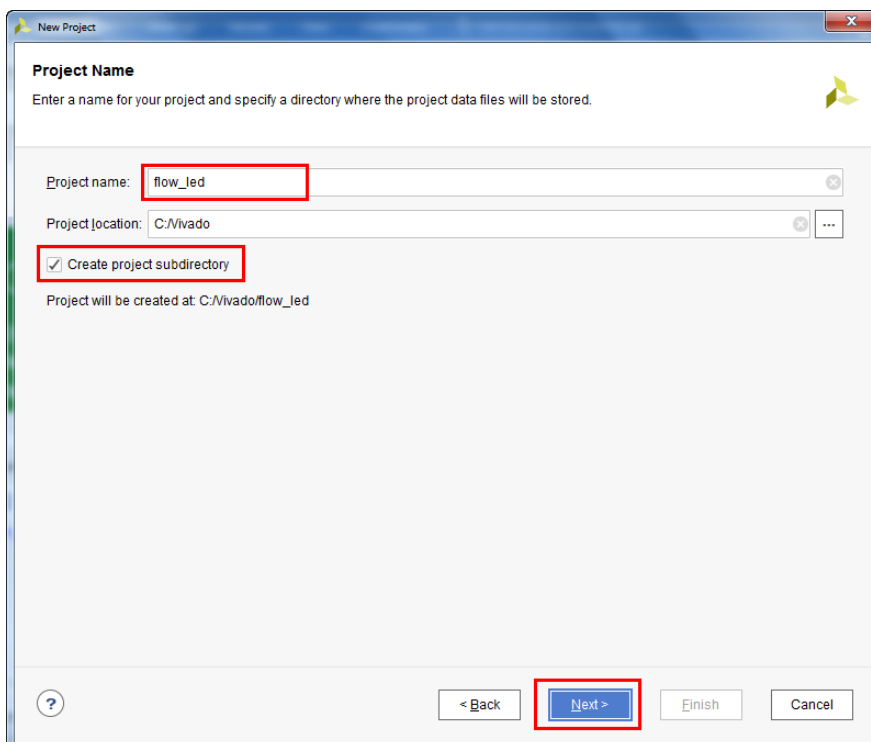
- 2) 点击‘Create Project’，或者单击 File>New Project 创建工程文件。



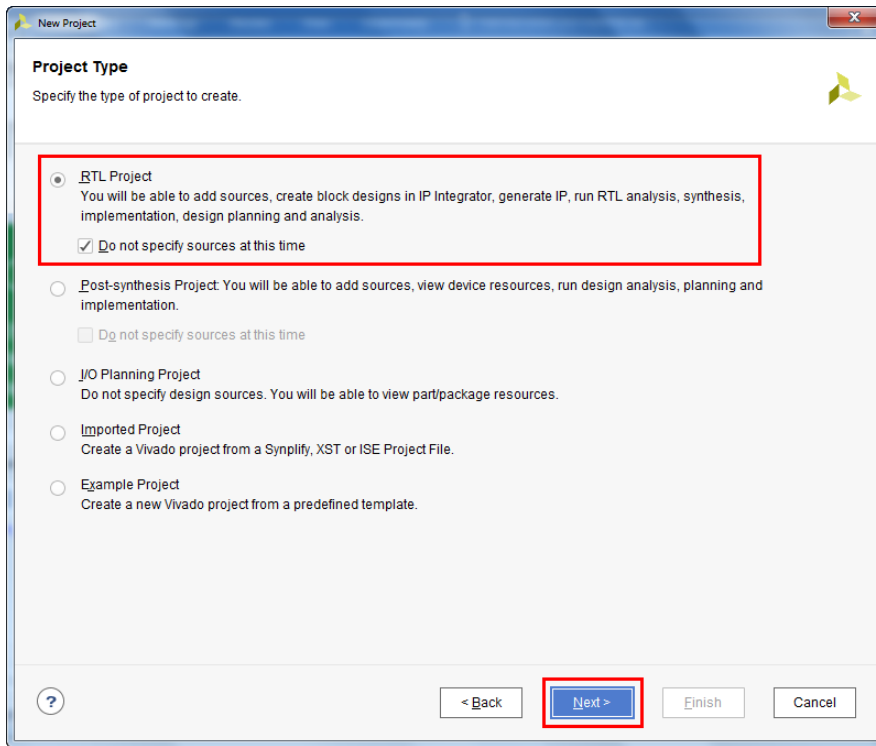
3) 弹出工程导向窗口，点击 **Next** 继续。



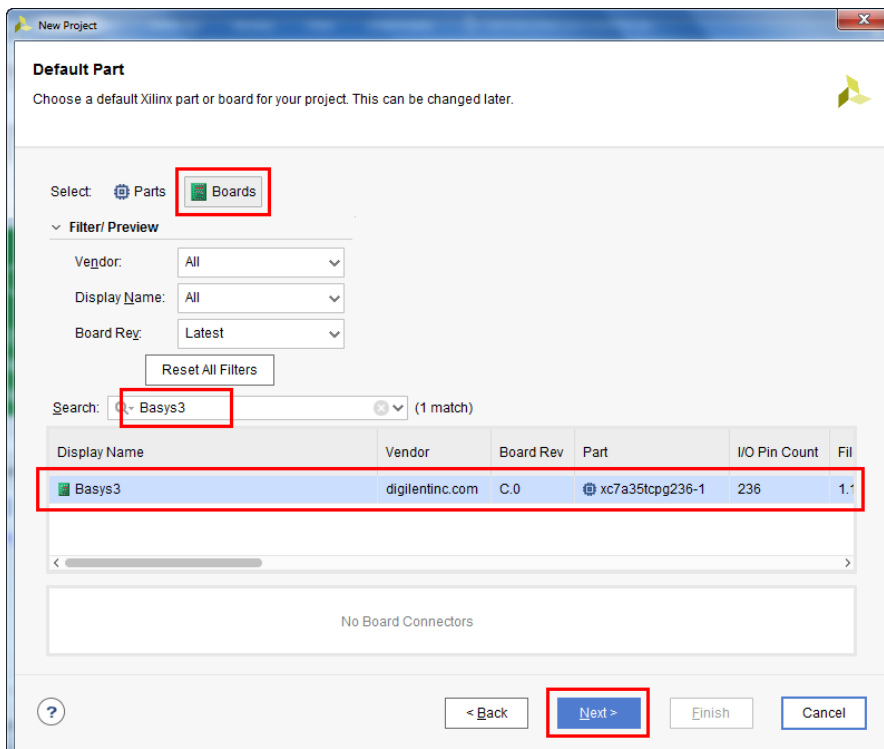
4) 将新的工程项目命名为‘flow_led’，勾选创建工程子文件夹，点击 **Next** 继续。



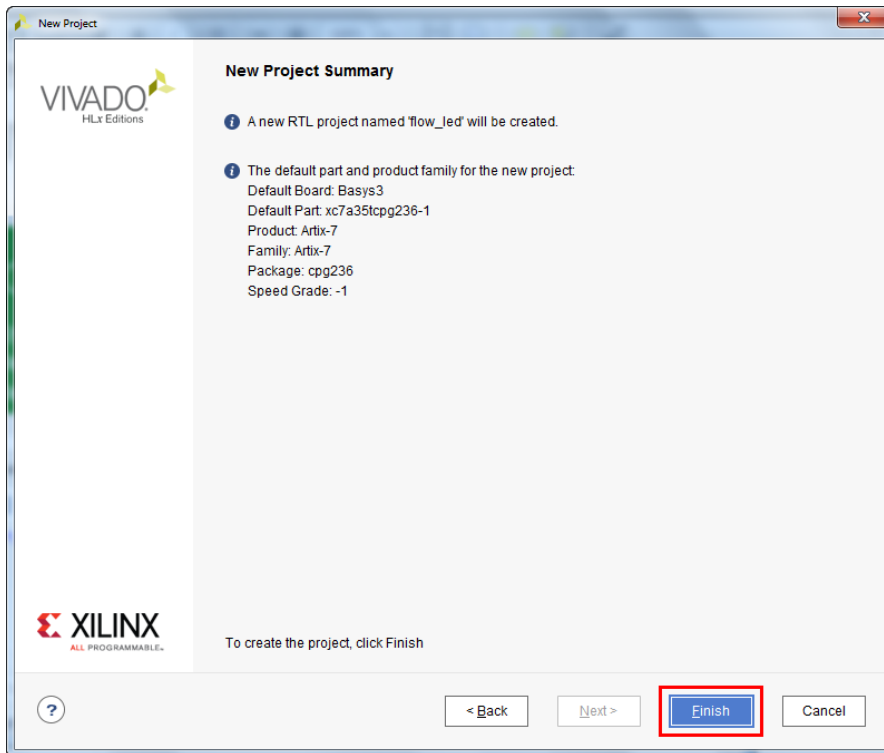
- 5) 新建一个 RTL 工程，勾选‘Do not specify sources at this time’先不添加源文件，点击 **Next** 继续。



- 6) 点击 **Board**，在搜索框输入‘Basys3’，选择 **Basys3** 开发板，点击 **Next** 继续。注意需要先安装 Digilent board files。

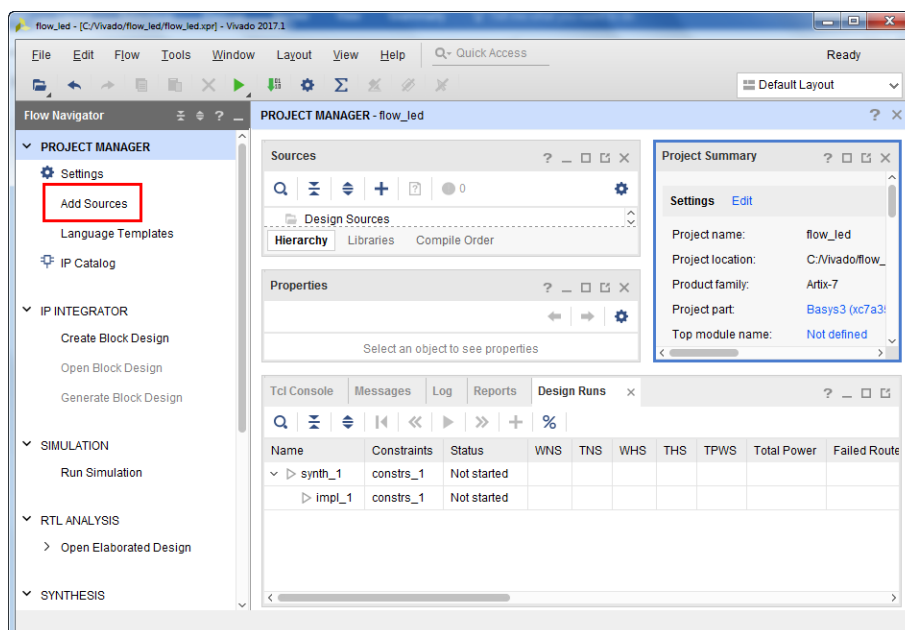


7) 确认相关信息无误后点击 **Finish** 完成工程项目的创建。

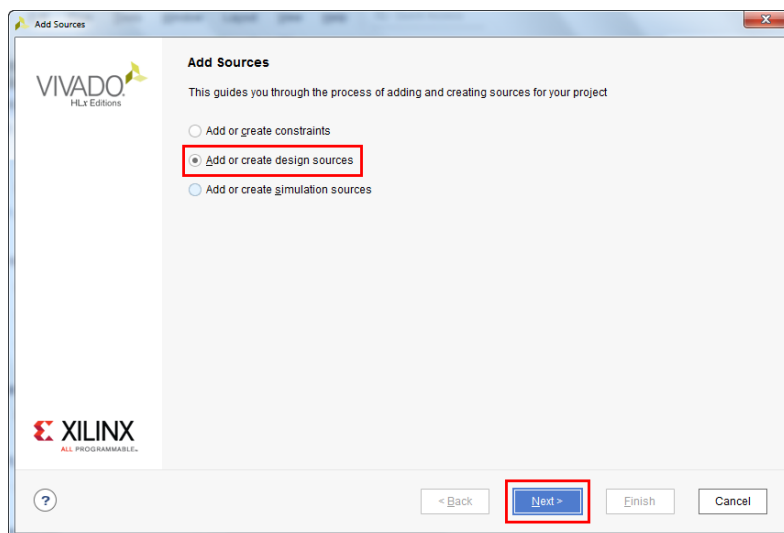


2. 添加源文件

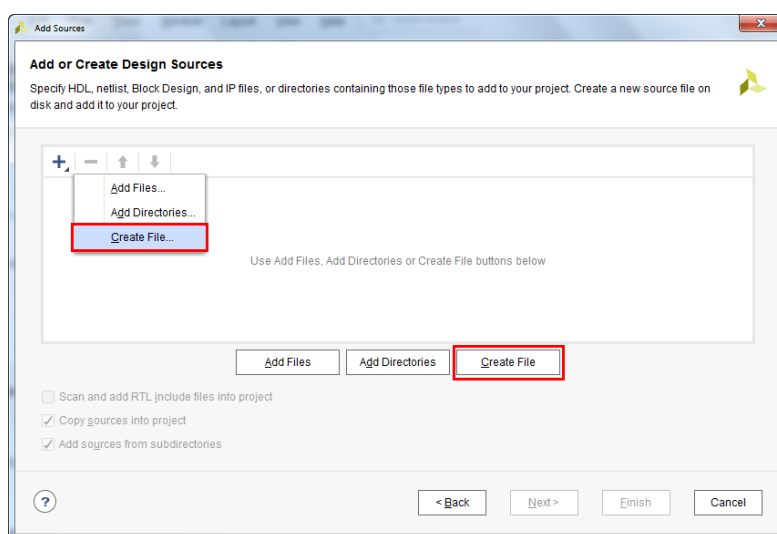
1) 完成工程创建后会弹出如下界面，在 **Flow Navigator** 中展开 **Project Manger**，点击 **Add Source** 添加源文件。



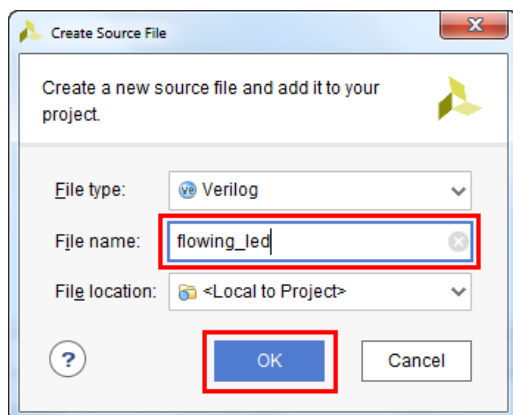
2) 选择‘Add or create design source’, 点击 Next 继续。



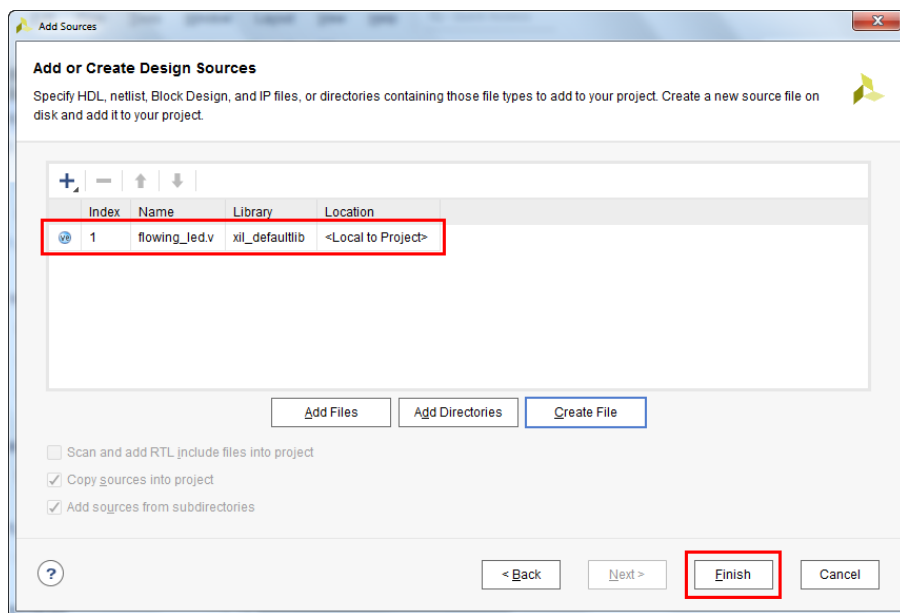
3) 点击 Create File 或者点击‘+’号，选择‘Create File’。



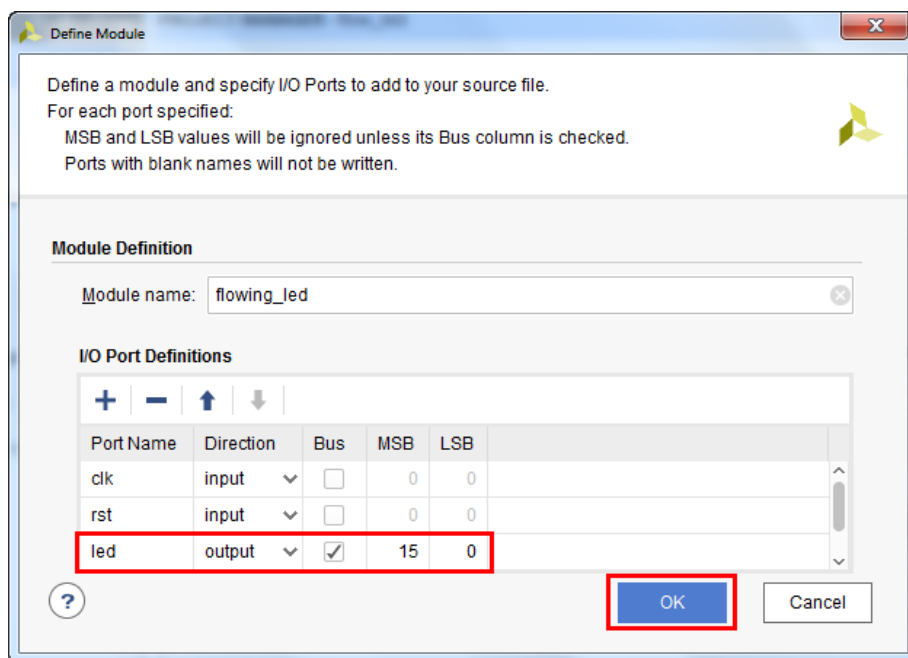
4) 在弹出的窗口中输入文件名‘flowing_led’，点击 OK。



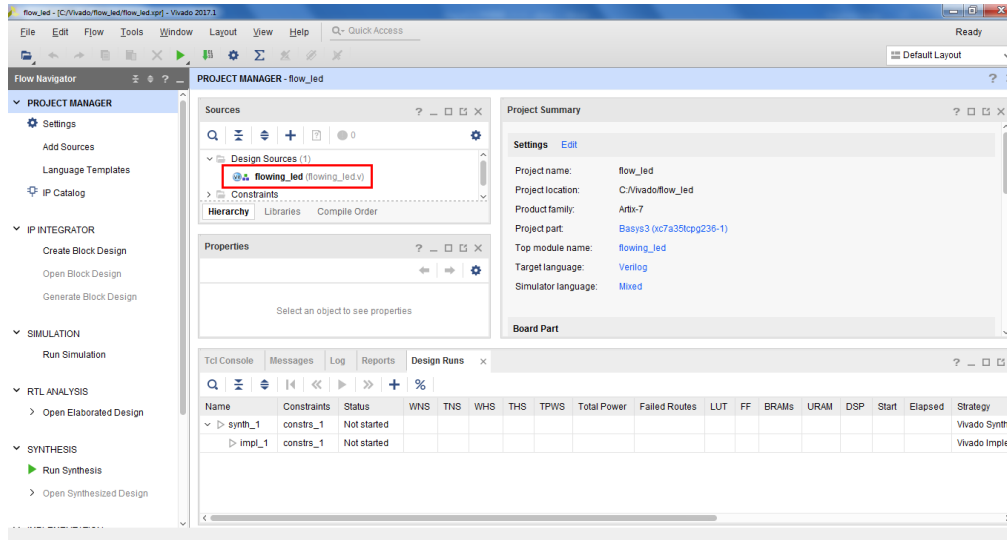
5) 新的源文件已经被添加到列表中，点击 **Finish** 完成添加。



6) 在弹出的窗口中定义模块的输入/输出端口，点击 **OK** 继续。



7) 在 Source 中展开 Design Sources，双击 `flowing_led` 打开‘`flowing_led.v`’文件。



8) 在‘`flowing_led.v`’文件中已经定义好了输入/输出端口，接下来需要完成模块的功能设计，在文件中输入相应的代码。

参考代码：

```
`timescale 1ns / 1ps
module flowing_light(
    input clk,
    input rst,
    output [15:0] led
);

reg[23:0] cnt_reg;
reg[15:0] light_reg;

always@(posedge clk)
begin
    if(rst)
        cnt_reg<=0;
    else
        cnt_reg<=cnt_reg+1;
end

always@(posedge clk)
begin
    if(rst)
        light_reg<=16'h0001;
    else if(cnt_reg == 24'hfffffff)
    begin
        if(light_reg == 16'h8000)
            light_reg<=16'h0001;
        else
            light_reg<=light_reg<<1;
        end
    end
end
```



```
assign led = light_reg;
```

```
endmodule
```

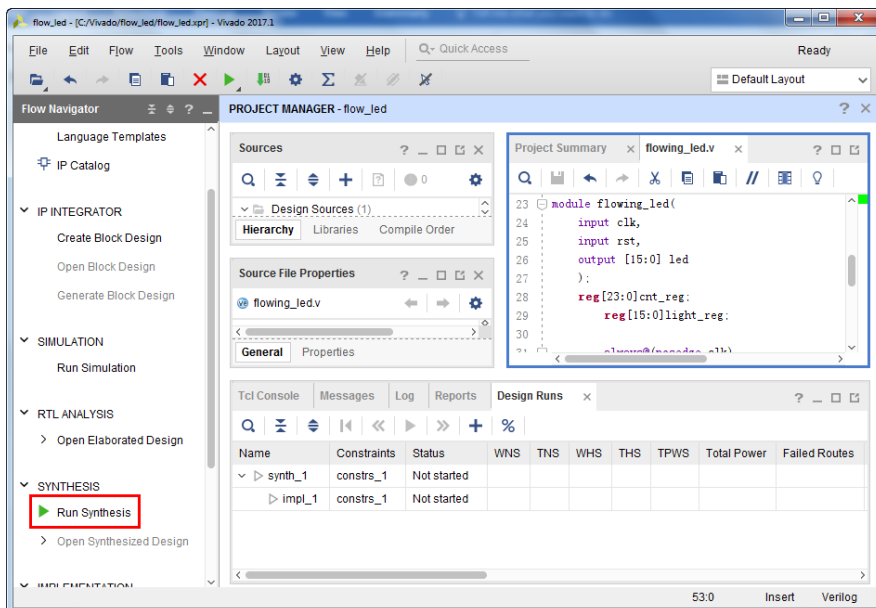
输入完成后 **Ctrl+S** 保存。

3. 添加约束文件

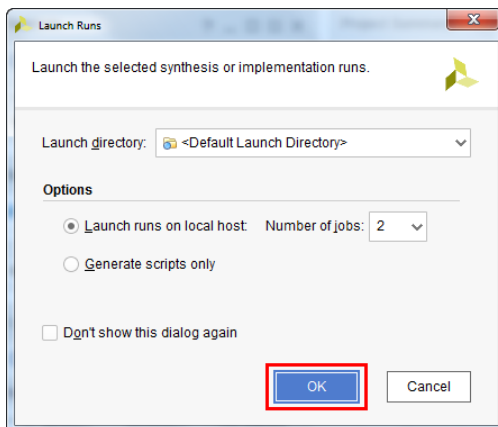
在 Vivado 2017.1 中有两种方式可以添加约束文件，一种是使用 Vivado I/O Planning 功能，另一种是新建约束文件，手动输入约束命令。下面将分别介绍两种约束文件的添加步骤：

a) 使用 Vivado I/O Planning 添加约束文件

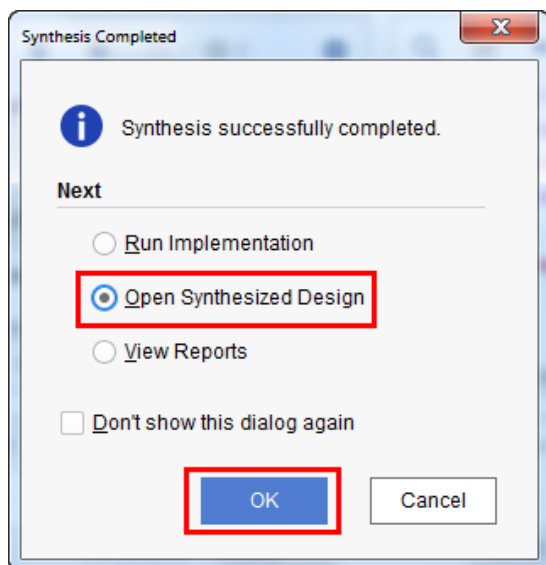
1) 在 Flow Navigator 中点击‘Run Synthesis’对工程进行综合。



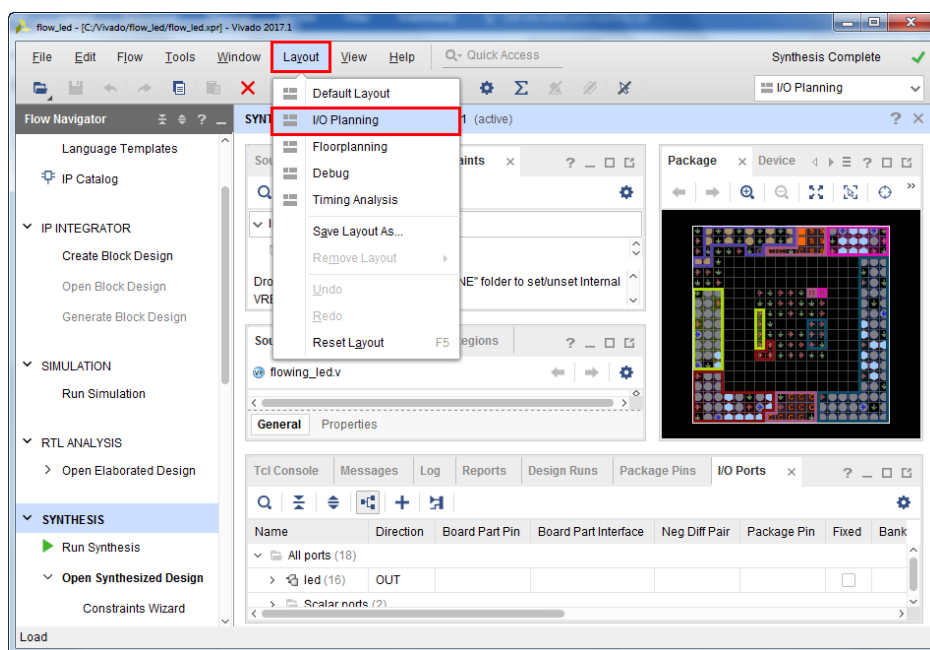
2) 使用默认设置，点击 OK 开始综合。



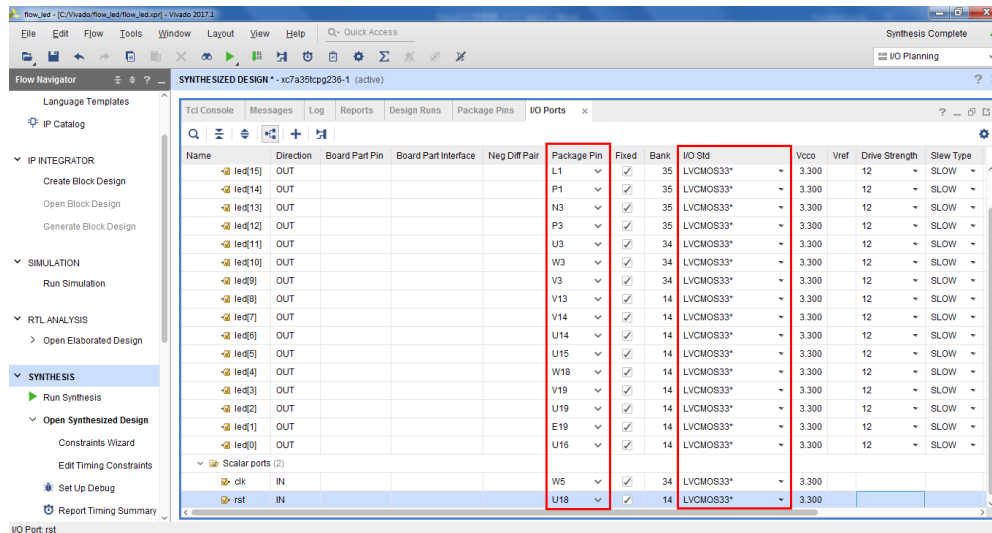
3) 综合完成后，选择打开综合后的结果，点击 OK 查看。



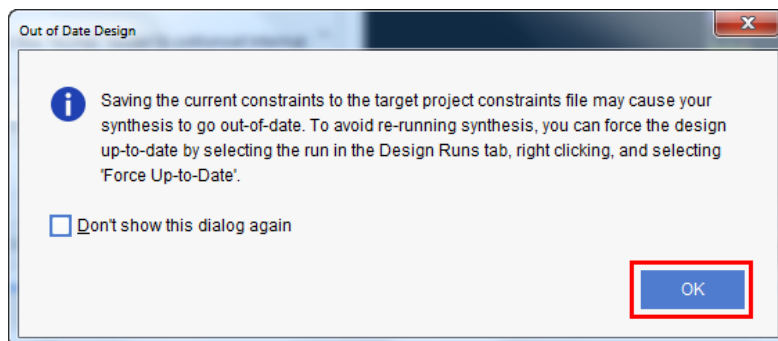
4) 此时应看到如下界面，如果没有出现如下界面可以在顶部工具栏中点击 Layout>I/O Planning。



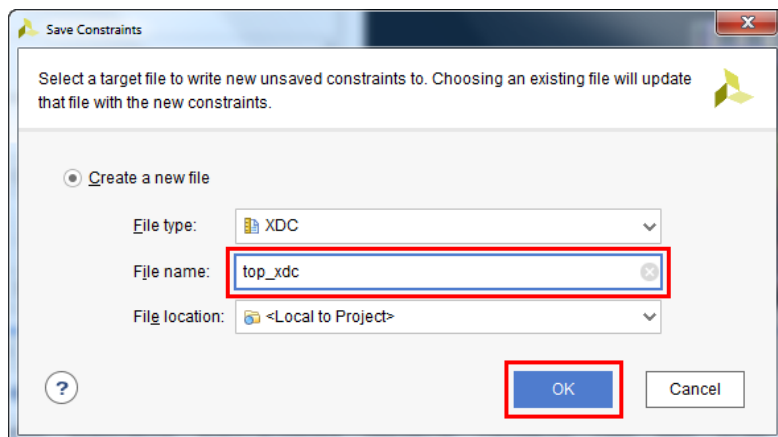
- 5) 在右下方选项卡中选中 I/O Ports 一栏，在对应的信号后面输入对应的 Basys 3 FPGA 引脚标号，选择对应的电平标准，Basys 3 使用 LVCMOS33(3.3V) 的电平标准。



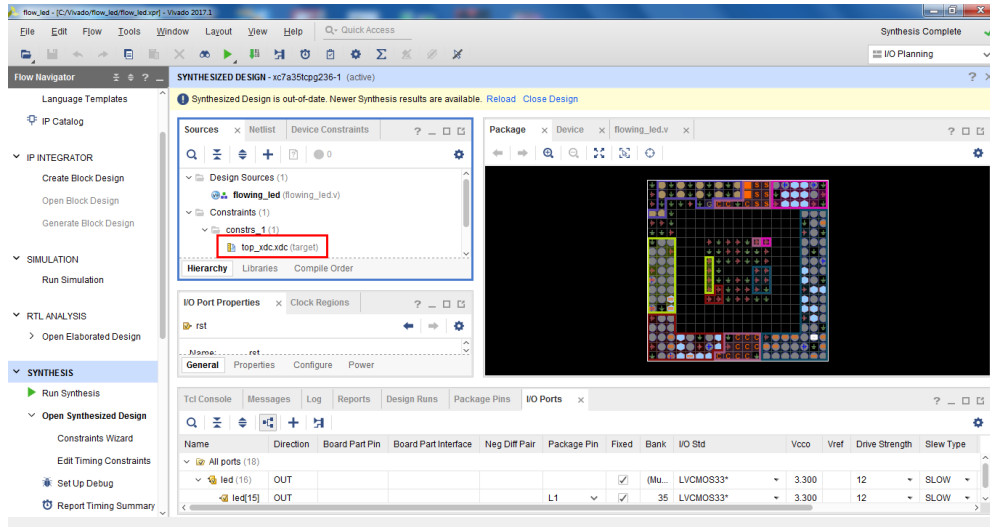
- 6) 添加约束文件需要重新综合，更新设计，点击 OK 继续。



- 7) 创建约束文件，命名约束文件为'top_xdc.xdc'，点击 OK 继续。

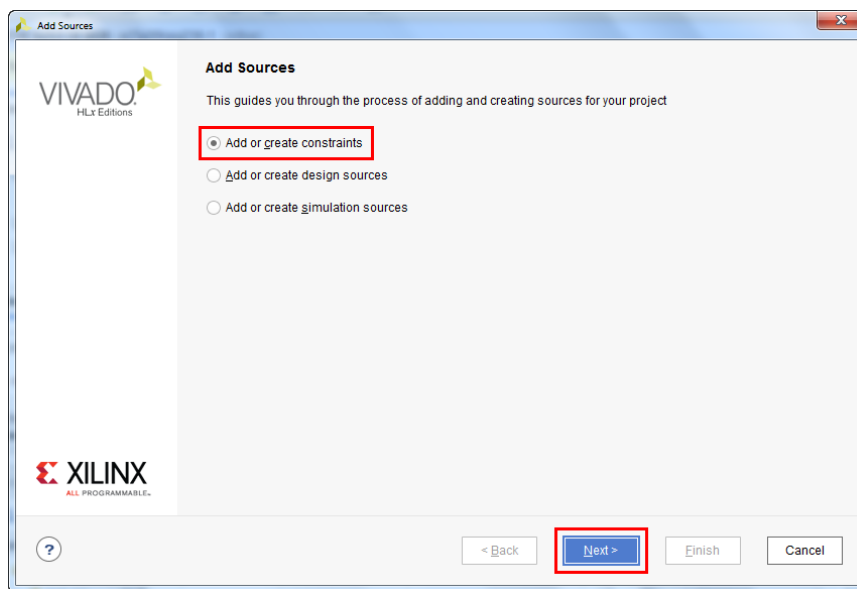


- 8) 创建完成后，可以在 Source 窗口中展开 Constraints 找到‘top_xdc.xdc’文件。按照步骤 1) 点击‘Run Synthesis’重新综合，更新设计工程。

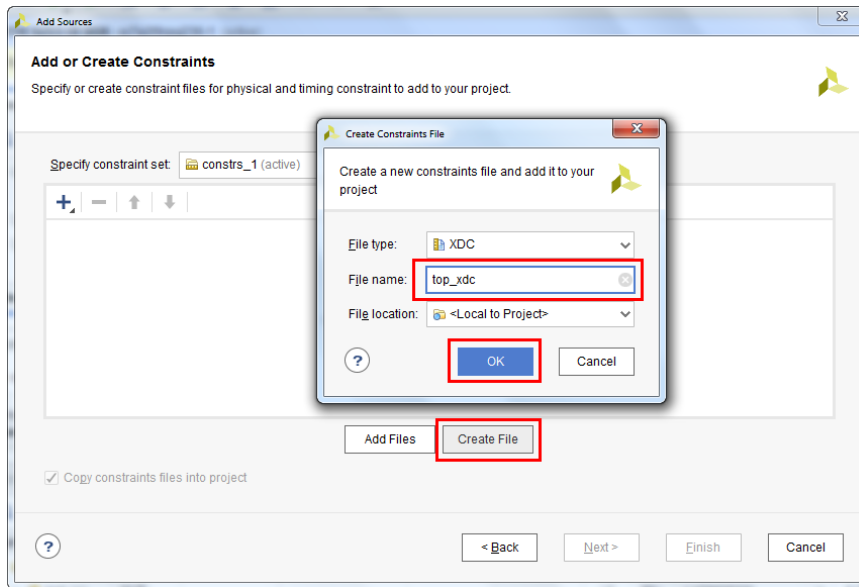


b) 新建手动输入约束文件

- 1) 在 Project Manager 中点击 Add Source，选择第一项‘Add or create constraints’，点击 Next 继续。



- 2) 点击 **Create File**，在弹出窗口中输入文件名‘top_xdc.xdc’，点击 **OK** 完成新建约束文件。



- 3) 点击 **Finish** 完成约束文件创建。
- 4) 在 **Source** 窗口中展开 **Constraints**，双击‘top_xdc.xdc’文件开始编辑，输入相应的 **Basys 3 FPGA** 引脚信息和电平标准。

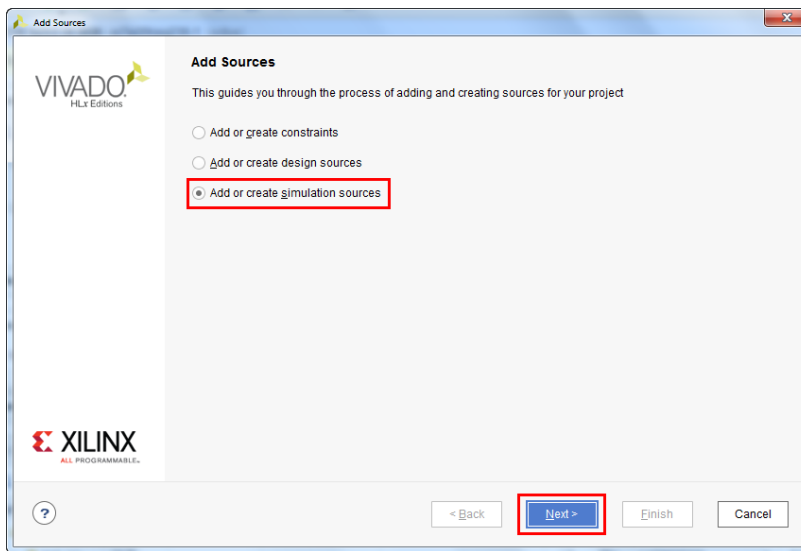
参考代码：

```
set_property PACKAGE_PIN L1 [get_ports {led[15]}]
set_property PACKAGE_PIN P1 [get_ports {led[14]}]
set_property PACKAGE_PIN N3 [get_ports {led[13]}]
set_property PACKAGE_PIN P3 [get_ports {led[12]}]
set_property PACKAGE_PIN U3 [get_ports {led[11]}]
set_property PACKAGE_PIN W3 [get_ports {led[10]}]
set_property PACKAGE_PIN V3 [get_ports {led[9]}]
set_property PACKAGE_PIN V13 [get_ports {led[8]}]
set_property PACKAGE_PIN V14 [get_ports {led[7]}]
set_property PACKAGE_PIN U14 [get_ports {led[6]}]
set_property PACKAGE_PIN U15 [get_ports {led[5]}]
set_property PACKAGE_PIN W18 [get_ports {led[4]}]
set_property PACKAGE_PIN V19 [get_ports {led[3]}]
set_property PACKAGE_PIN U19 [get_ports {led[2]}]
set_property PACKAGE_PIN E19 [get_ports {led[1]}]
set_property PACKAGE_PIN U16 [get_ports {led[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[15]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[14]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[13]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[12]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[11]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[10]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[9]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[8]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[6]}]
```

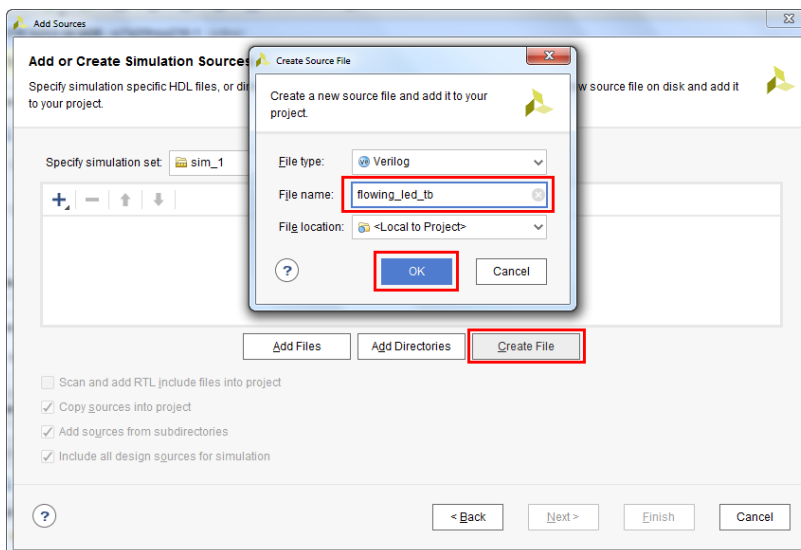
```
set_property IOSTANDARD LVCMOS33 [get_ports {led[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {led[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports rst]
set_property PACKAGE_PIN W5 [get_ports clk]
set_property PACKAGE_PIN U18 [get_ports rst]
```

4. 利用 Vivado 进行功能仿真

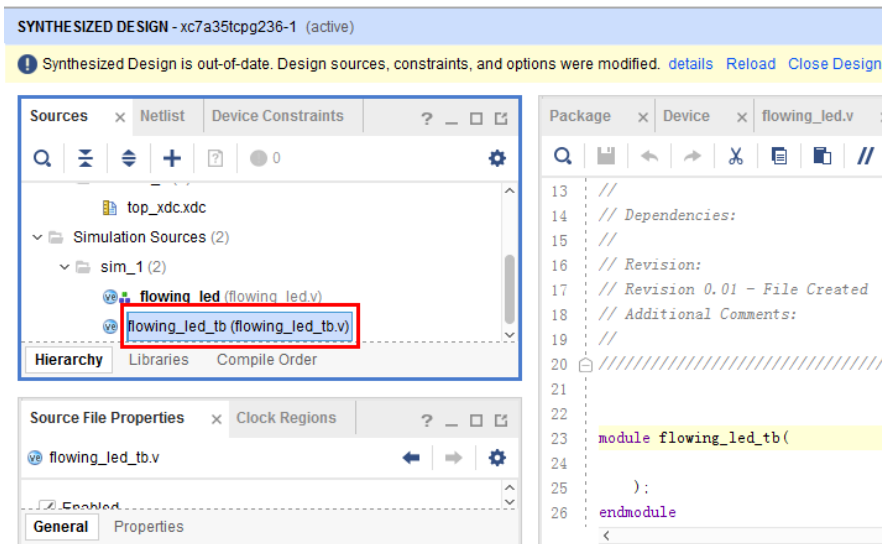
- 1) 创建测试平台 (testbench) 文件，点击 **Add Source**。选择 'Add or create simulation sources'，点击 **Next** 继续。



- 2) 点击 **Create File**，将测试文件命名为 'flowing_led_tb.v'，点击 **Finish** 完成测试文件的创建。



- 3) 在弹出的输入输出端口设置的窗口中设置为空，直接点击 OK。
- 4) 在 **Source** 一栏中展开 **Simulation Sources**，展开 **sim_1** 文件夹，双击打开 'flowing_led_tb.v' 文件开始编辑。



- 5) 编辑测试平台文件，参考代码如下：

```
`timescale 1ns / 1ps
module flowing_led_tb();
reg clk;
reg rst;
wire[3:0]led;

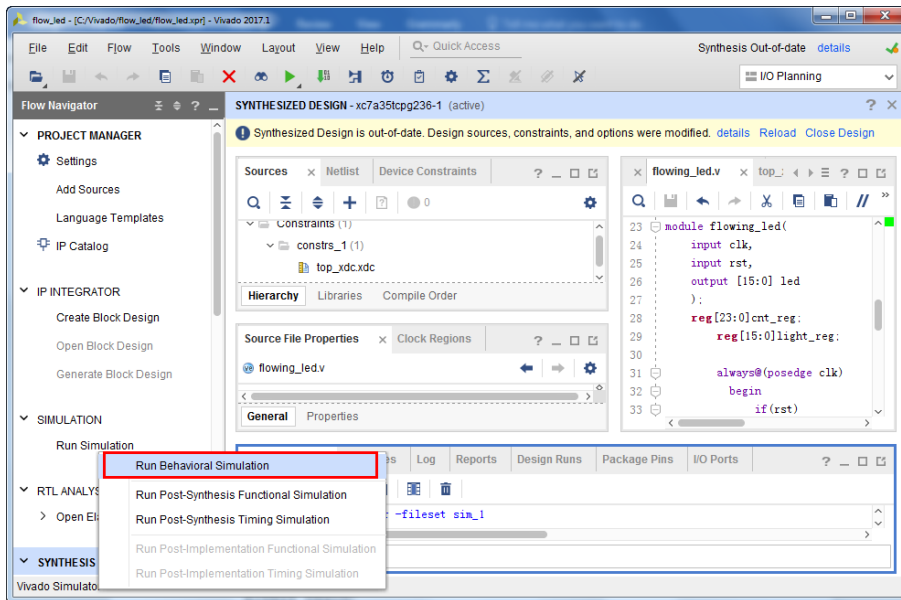
flowing_led u0(
    .clk(clk),
    .rst(rst),
    .led(led));

parameter PERIOD = 10;

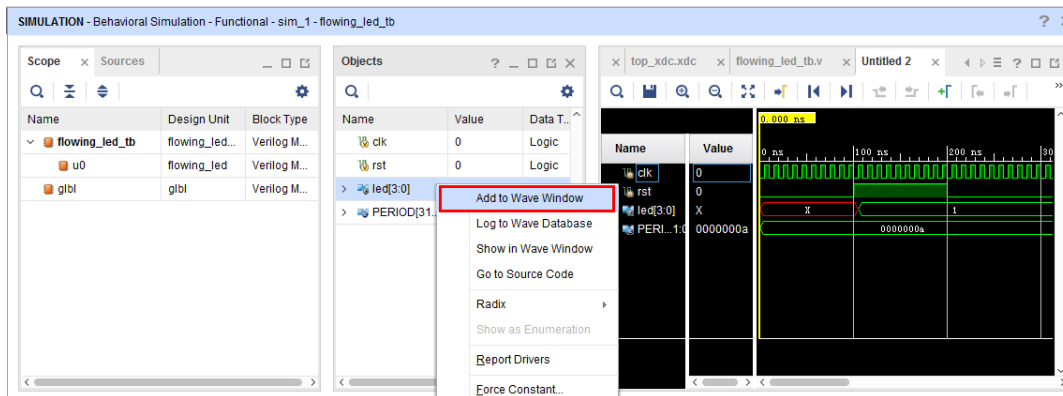
always begin
    clk = 1'b0;
    #(PERIOD/2) clk = 1'b1;
    #(PERIOD/2);
end

initial begin
    clk = 1'b0;
    rst = 1'b0;
    #100;
    rst = 1'b1;
    #100;
    rst = 1'b0;
end
endmodule
```

- 6) 在 Flow Navigator 中，展开 Simulation，点击 Run Simulation，选择‘Run Behavioral Simulation’。



- 7) 可以通过左侧的 Scope 一栏中找到想要查看的模块的寄存器，在 Objects 一栏中右击选择‘Add to Wave Window’添加需要查看的信号。

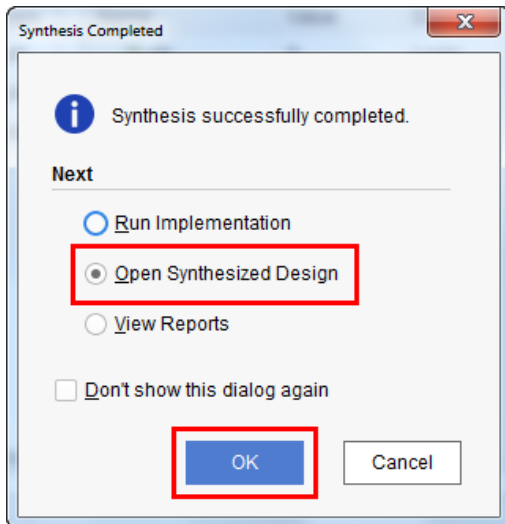


可通过选择工具栏中的如下选项来进行波形的仿真时间控制。如下工具条，分别是复位波形（即清空现有波形）、运行仿真、运行特定时长的仿真、仿真时长设置、仿真时长单位、单步运行、暂停。

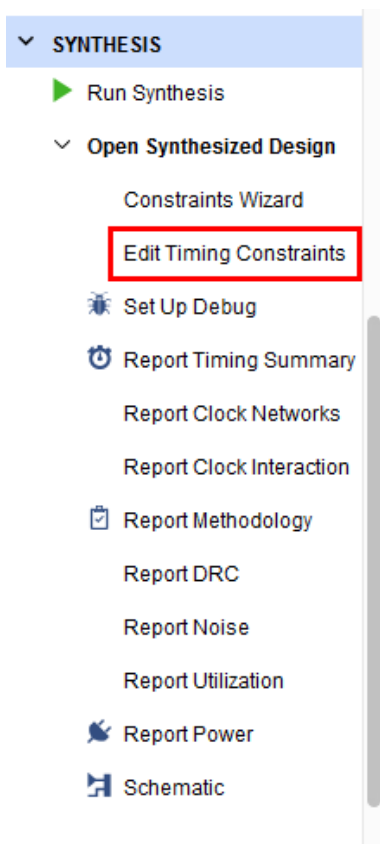


5. 添加时序约束

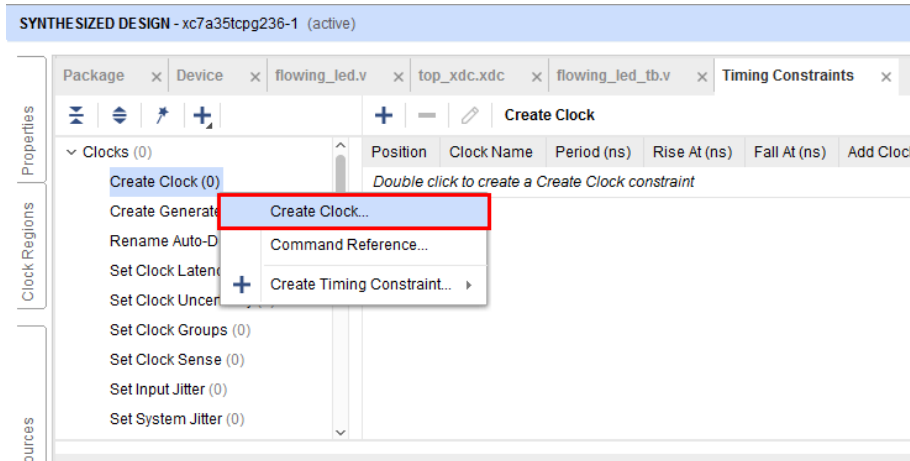
1) 重新综合完成后选择‘Open Synthesized Design’，点击 OK。



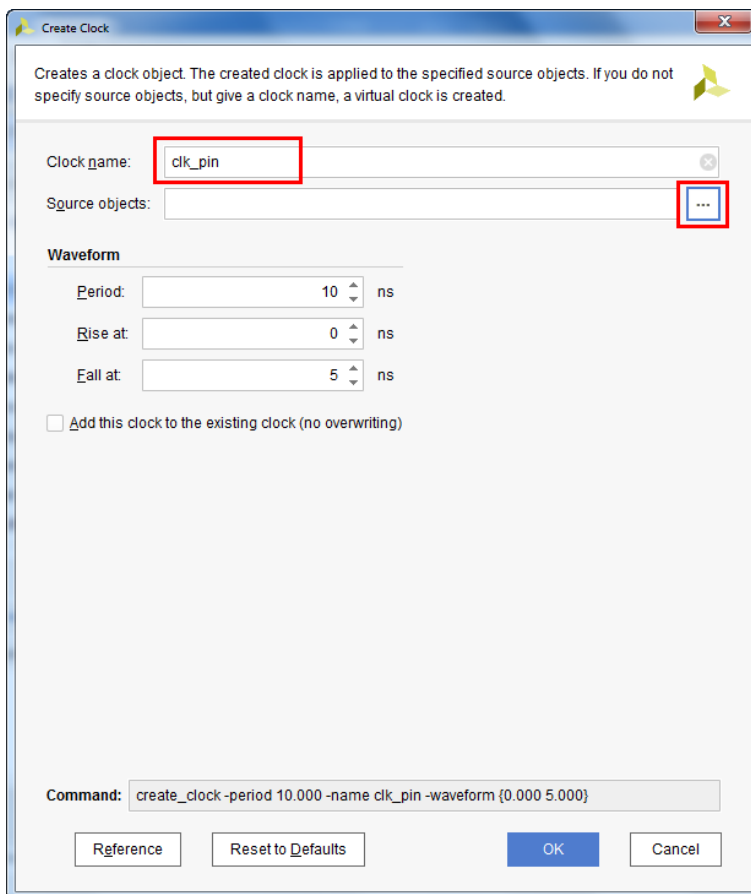
2) 在左侧的 Flow Navigator 中依次选择 Synthesis>Open Synthesized Design>Edit Timing Constraints。



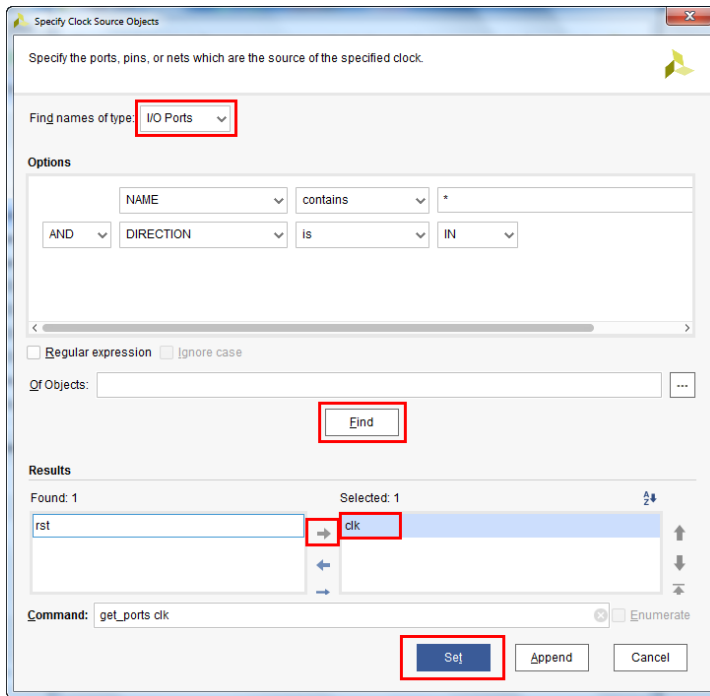
3) 在时序约束窗口中展开 Clocks，右键 Create Clock，选择‘Create Clock’。



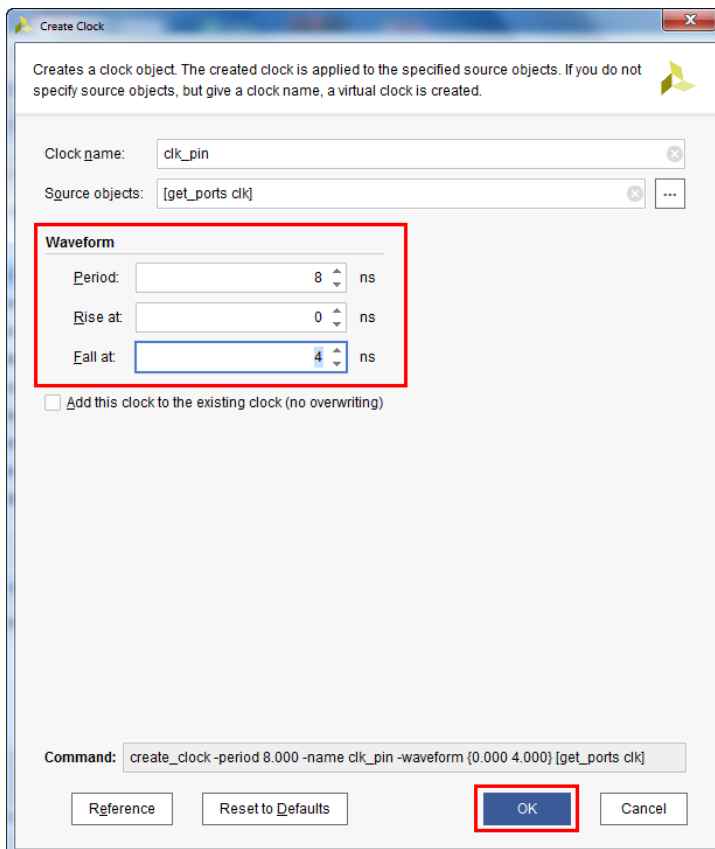
4) 在弹出的窗口中将时钟命名为‘clk_pin’，在 Source objects 一项中点击右侧的按钮选择目标源。



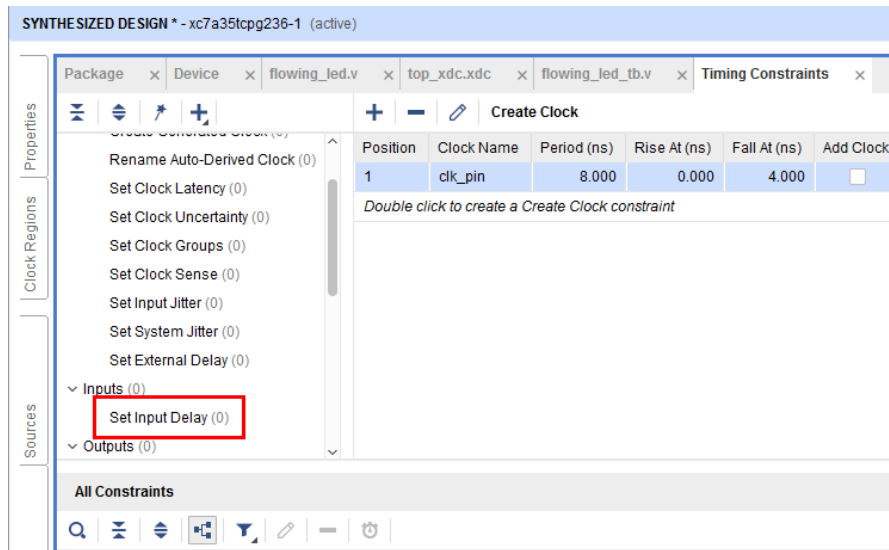
- 5) 在弹出窗口中选择 I/O Ports 类型，点击‘Find’，选中‘clk’端口并将其移到右侧，点击‘Set’继续。



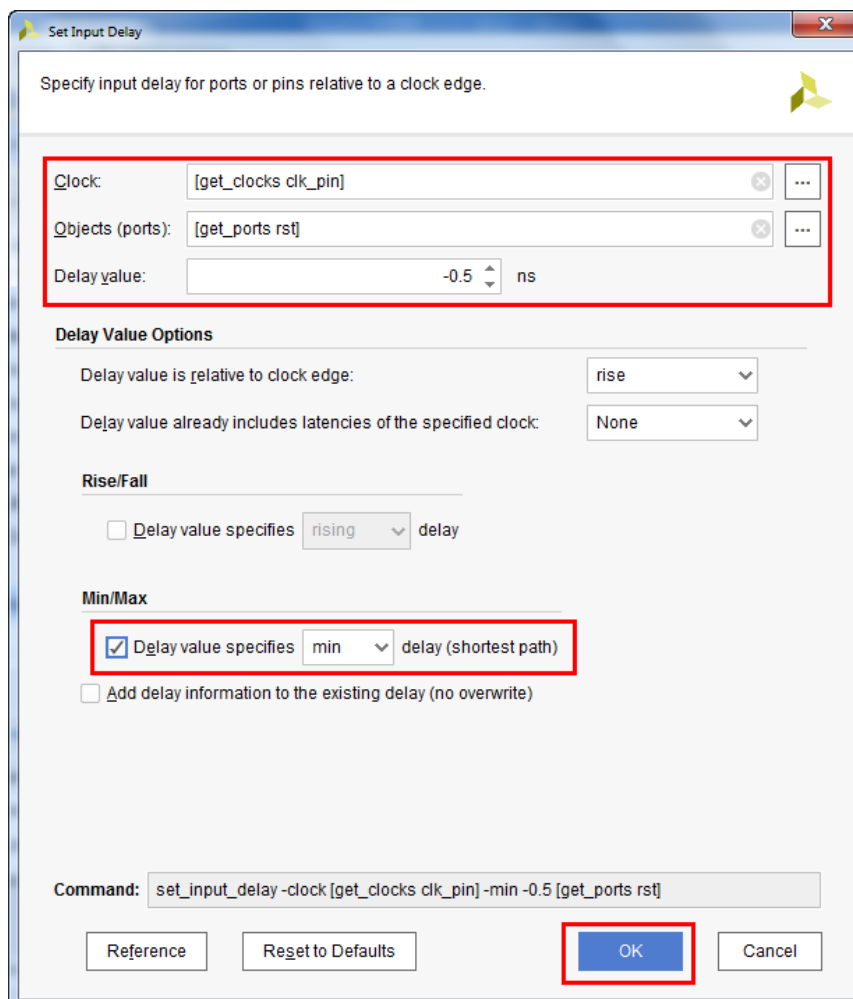
- 6) 将周期设为 8ns，上升设为 0ns，下降设为 4ns，点击 OK 继续。



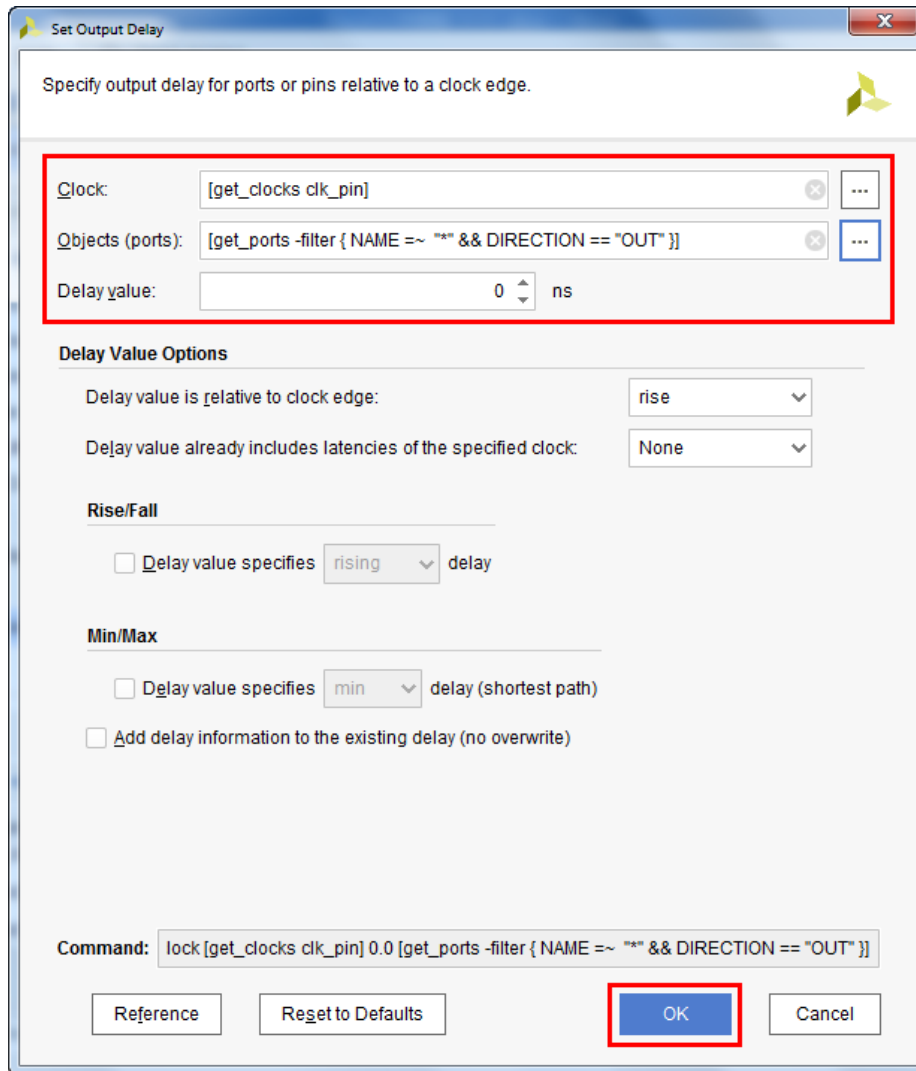
7) 接下来设置输入延迟，展开 Inputs，双击‘Set Input Delay’。



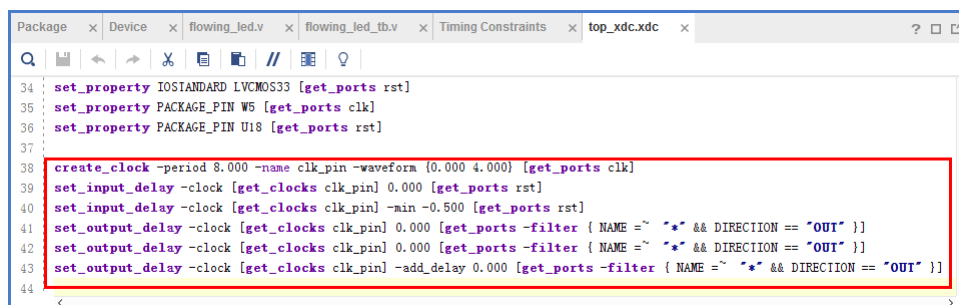
8) 按照下图设置输入延迟，Clock 选择 clk_pin，Objects 选择 rst。Delay value 设为-0.5ns，勾选最小延迟(最短路径)，点击 OK。



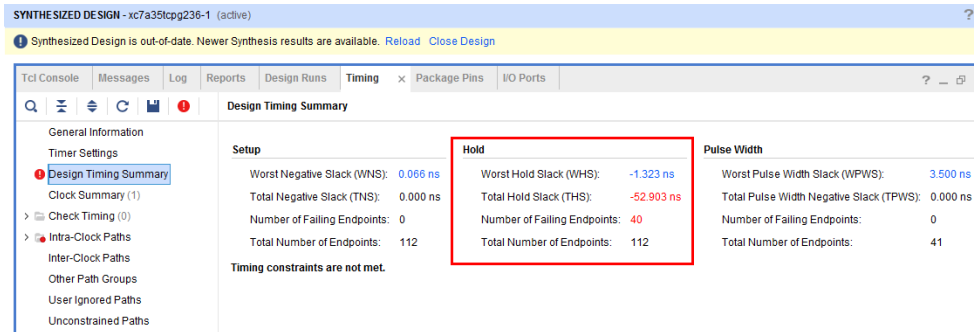
- 9) 展开 Outputs, 双击‘Set Output Delay’, Clock 选择 clk_pin, Objects 选择所有的输出(led[0]~led[15]), 延迟设为 0ns, 点击 OK 继续。



- 10) 在工具栏中选择 File>Save Constraints, 或者 Ctrl+S 保存。然后打开‘top_xdc.xdc’文件可以看到约束命令已被写入约束文件。

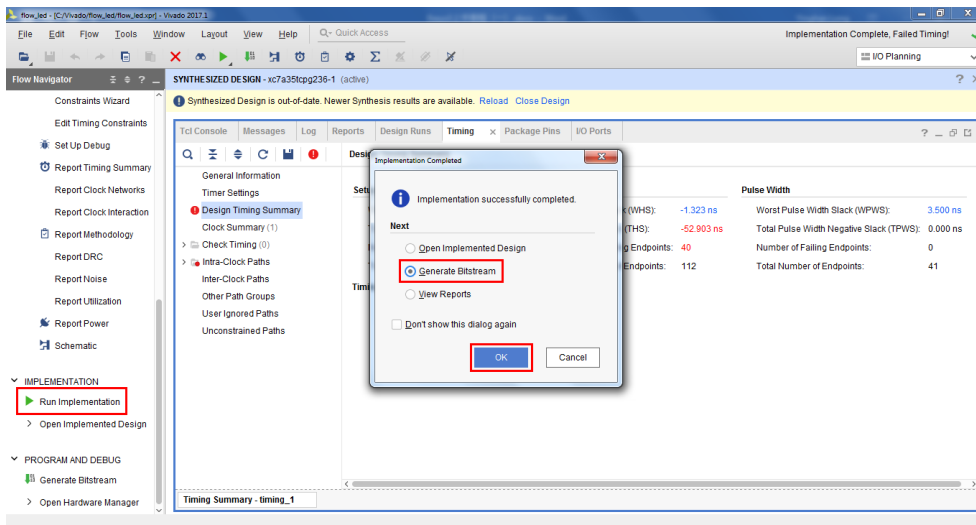


- 11) 完成时序报告后，可以看到 **Hold** 一项中显示为红色，并且在‘Design Timing Summery’旁有一个红色的感叹号，表示时序约束后无法满足需求，在 Implementation(实现)时，Vivado 会自动优化布线路径，来满足用户设定的时序约束条件。

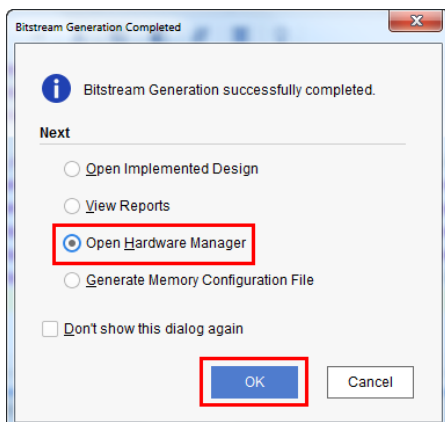


6. 工程实现

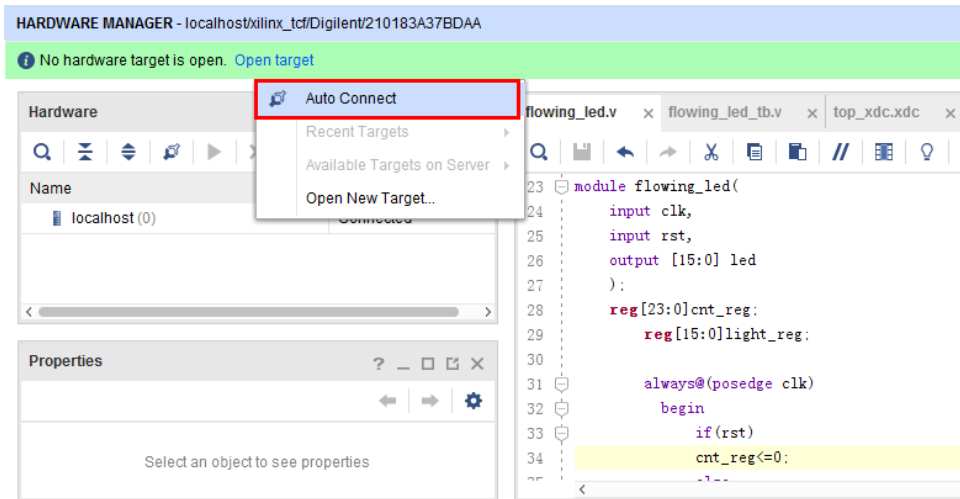
- 1) 在 Flow Navigator 中展开 Implementation，点击‘Run Implementation’。完成后，在弹出窗口中选择‘Generate Bitstream’生成比特流文件。



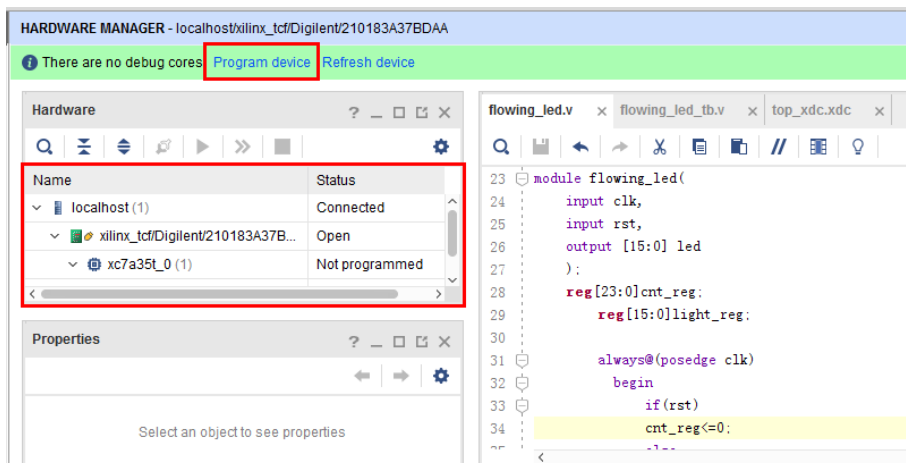
- 2) 生成比特流文件后选择‘Open Hardware Manager’，点击 OK。



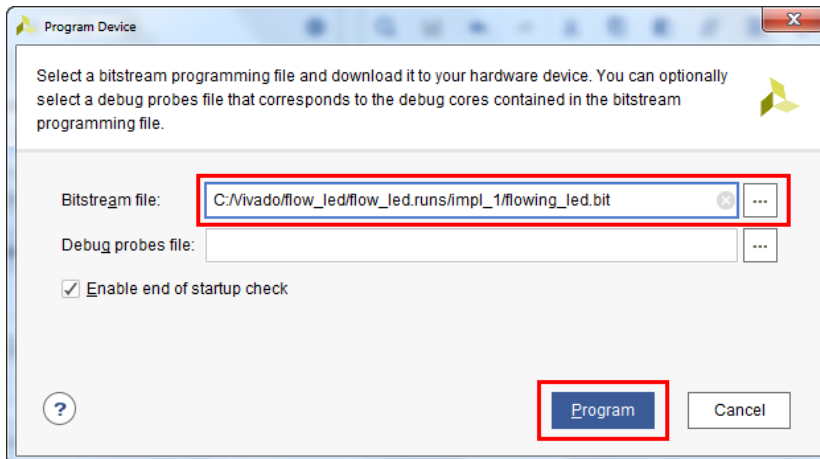
- 3) 将 Basys 3 FPGA 开发板 PROG 接口(J4)通过 micro-USB 线连接到电脑 USB 接口，确认电源(JP2)跳线为 USB 供电模式，模式(JP1)跳线为 JTAG。打开 Basys 3 开关通电。
- 4) 在 Hardware Manager 中点击‘Open Target’，选择‘Auto Connect’连接 Basys 3 FPGA 开发板。



- 5) 在 Hardware 一项中看到，Basys 3 FPGA 已经成功的连接到了电脑，点击‘Program device’将工程下载到 Basys 3 开发板上。



6) 选择需要下载的比特流文件，点击‘Program’开始下载。



7) 下载完成后，在 Basys 3 FPGA 开发板上的运行演示。

