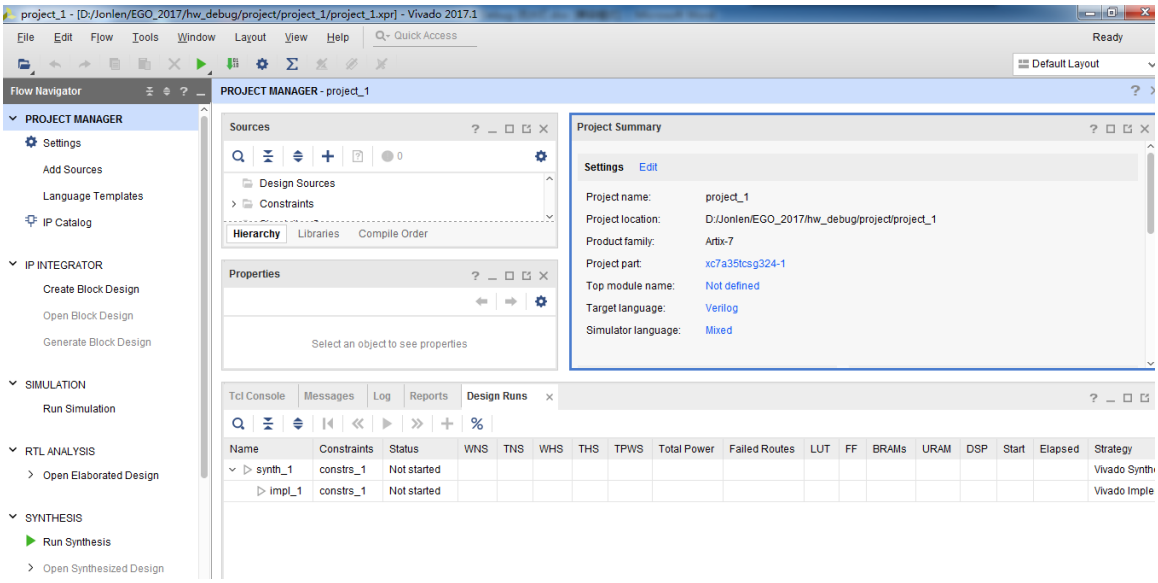
HW Debug-流水灯

一、创建工程

1、创建一个新的工程“hw_debug”。



二、添加源文件和约束文件

2、创建源文件“flow_led_top.v”,“clk_div.v”和“flow_led.v”，并用 Verilog 语言输入流水灯设计。各模块源代码如下：

flow_led_top.v:

```
module   flow_led_top(
      input clk,                  //100MHz
      output [15:0] led
      );

      wire clk_pulse;

      clk_div clk_div(
            .clk(clk),
            .clk_pulse(clk_pulse)          //1Hz
            );

       flow_led flow_led(
            .clk(clk),
            .clk_pulse(clk_pulse),
            .led_r(led)
           );

endmodule
```

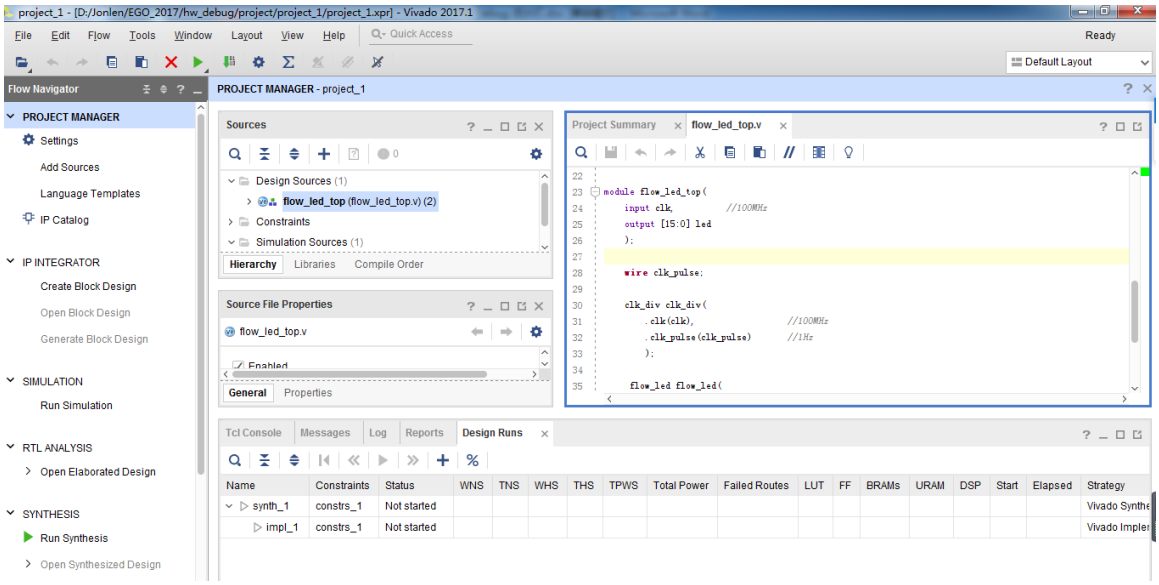
clk_div.v:

```
module   clk_div(
      input clk,                  //100MHz
      output clk_pulse            //1Hz
```

```
    );

    reg clk_pulse = 0;
    reg [25:0]    div_counter = 0;

    always @(posedge clk) begin
       if(div_counter>=50000000) begin
            clk_pulse<=1;
            div_counter<=0;
          end else begin
            clk_pulse<=0;
            div_counter <= div_counter + 1;
          end
    end

endmodule
```

flow_led.v:

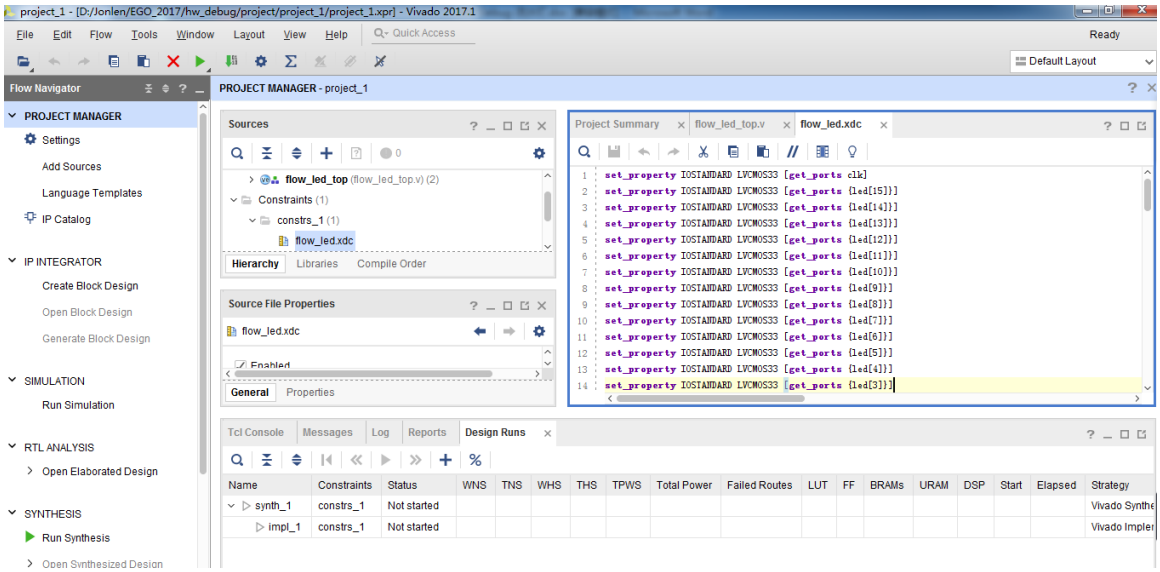
```
module    flow_led(
    input clk,
    input clk_pulse,                    //100MHz
    output reg [15:0] led_r
    );

    reg [15:0]    led_r = 16'h000f;

    always @(posedge clk) begin
      if(clk_pulse==1)
         led_r <= { led_r[11:0], led_r[15:12] };
      else
         led_r <= led_r;
    end

endmodule
```



3、添加约束文件“flow_led.xdc”。

```
set_property IOSTANDARD LVCMOS33 [get_ports clk]
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 PACKAGE_PIN P17 [get_ports clk]
set_property PACKAGE_PIN F6 [get_ports {led[15]}]
set_property PACKAGE_PIN G4 [get_ports {led[14]}]
set_property PACKAGE_PIN G3 [get_ports {led[13]}]
set_property PACKAGE_PIN J4 [get_ports {led[12]}]
set_property PACKAGE_PIN H4 [get_ports {led[11]}]
set_property PACKAGE_PIN J3 [get_ports {led[10]}]
set_property PACKAGE_PIN J2 [get_ports {led[9]}]
set_property PACKAGE_PIN K2 [get_ports {led[8]}]
set_property PACKAGE_PIN K1 [get_ports {led[7]}]
set_property PACKAGE_PIN H6 [get_ports {led[6]}]
set_property PACKAGE_PIN H5 [get_ports {led[5]}]
set_property PACKAGE_PIN J5 [get_ports {led[4]}]
set_property PACKAGE_PIN K6 [get_ports {led[3]}]
set_property PACKAGE_PIN L1 [get_ports {led[2]}]
set_property PACKAGE_PIN M1 [get_ports {led[1]}]
set_property PACKAGE_PIN K3 [get_ports {led[0]}]
```

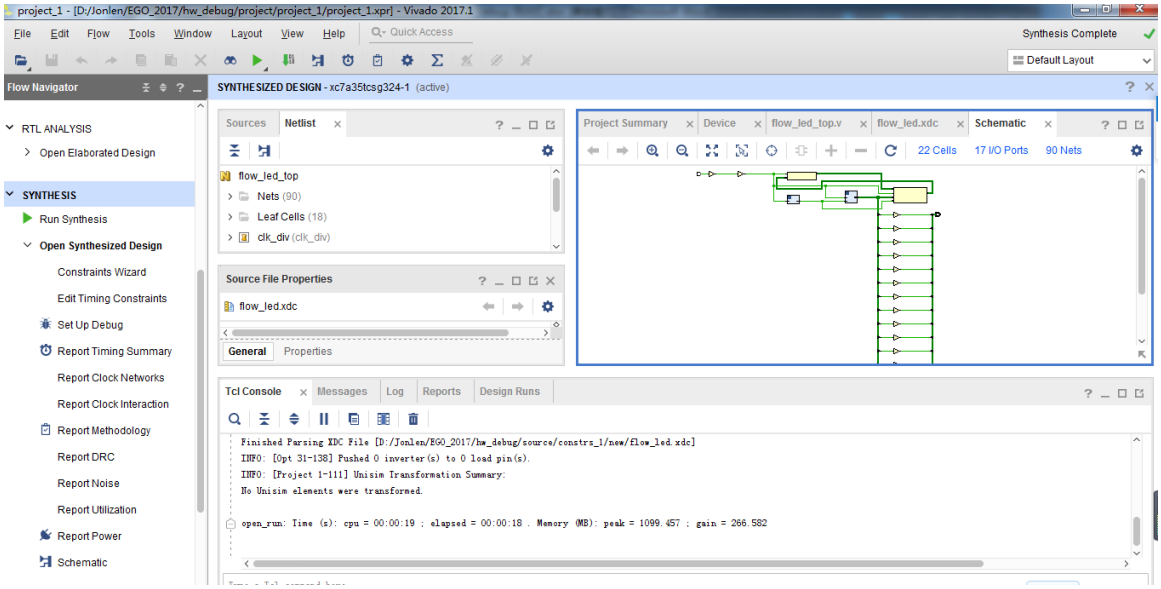


三、综合

4、在“Flow Navigator”栏中的“Synthesis”下点击“Run Synthesis”。右上角的进度条

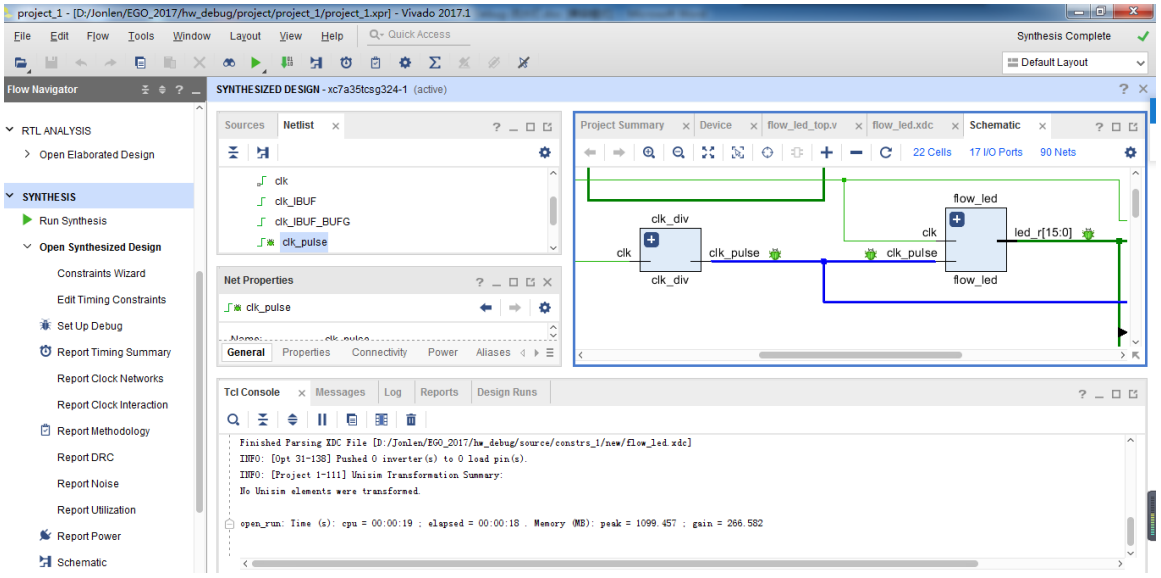
“Running synth_design”指示正在对工程进行综合。

综合完成之后在弹出的对话框中点击“Cancel”取消。在“Flow Navigator”一栏中，找到“Synthesis”-> “Open Synthesised Design”-> “Schematic”，点击“Schematic”。

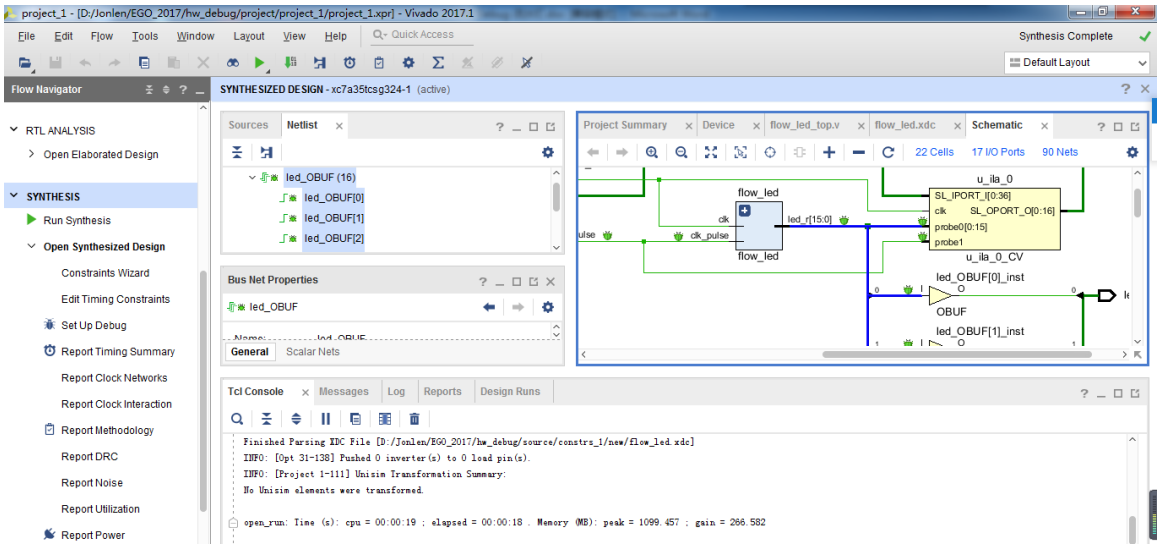


四、Mark Debug

5、在“Schematic”标签页中，点击左侧工具栏中的放大镜图标，将电路图放大到合适大小。找到“clk_div”模块和“flow_led”模块之间的连线“clk_pulse”，选中后右击，选择“Mark Debug”。

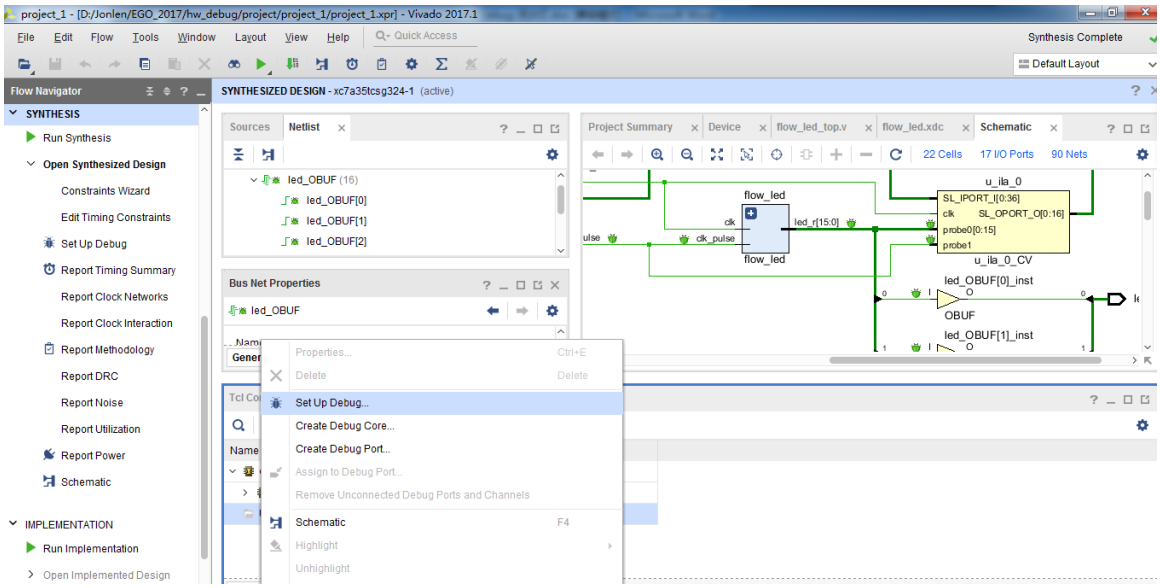


6、同样，找到与“flow_led”模块的输出端口相连的信号线“led_OBUF”，选中后右击，选择“Mark Debug”。

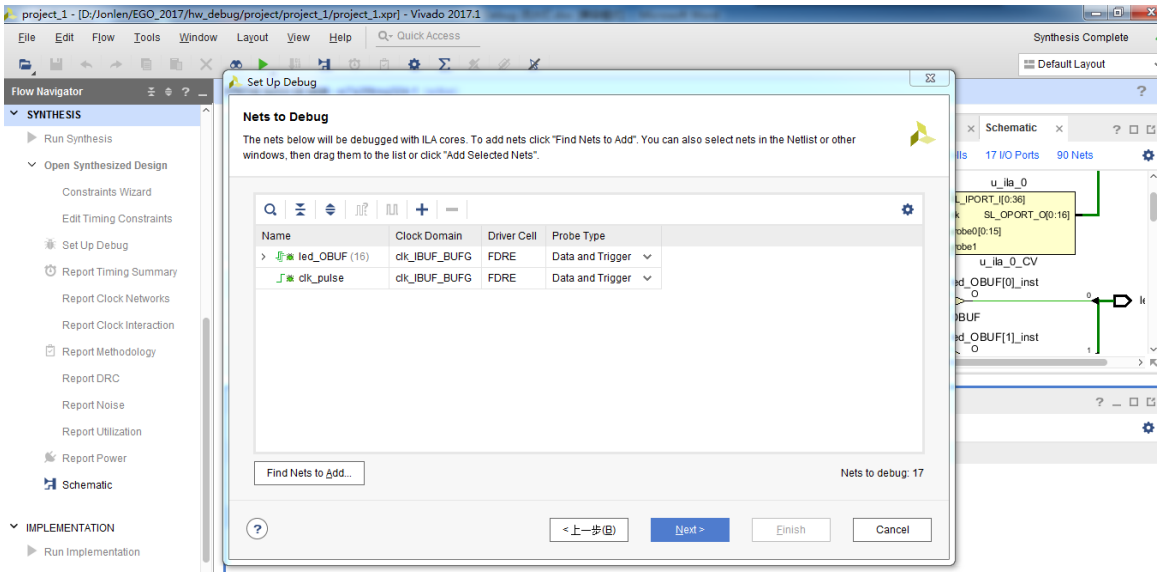


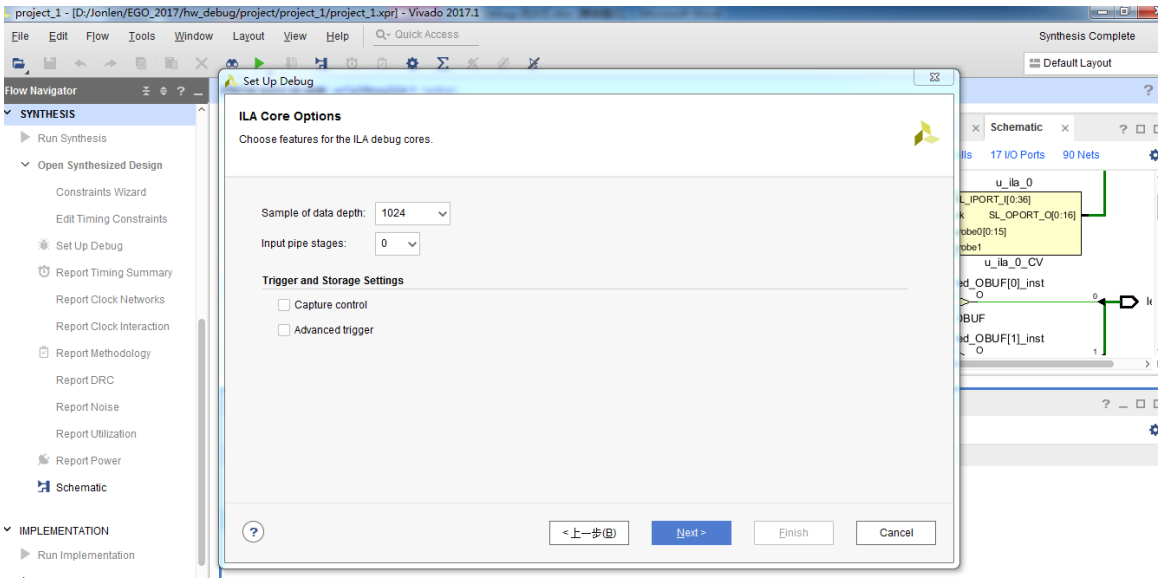
五、Set up Debug

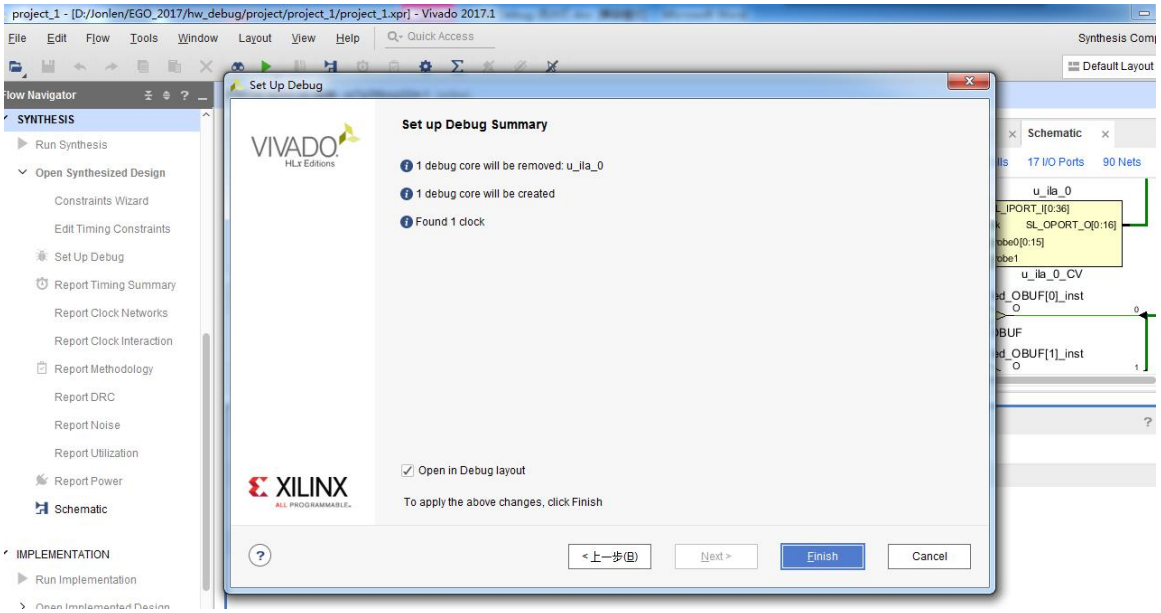
7、在“Debug”窗口中（可通过在菜单栏中点击“Layout”-> “Debug”打开），单击选中“Unassigned Debug Nets”，然后右击选择“Set Up Debug”。



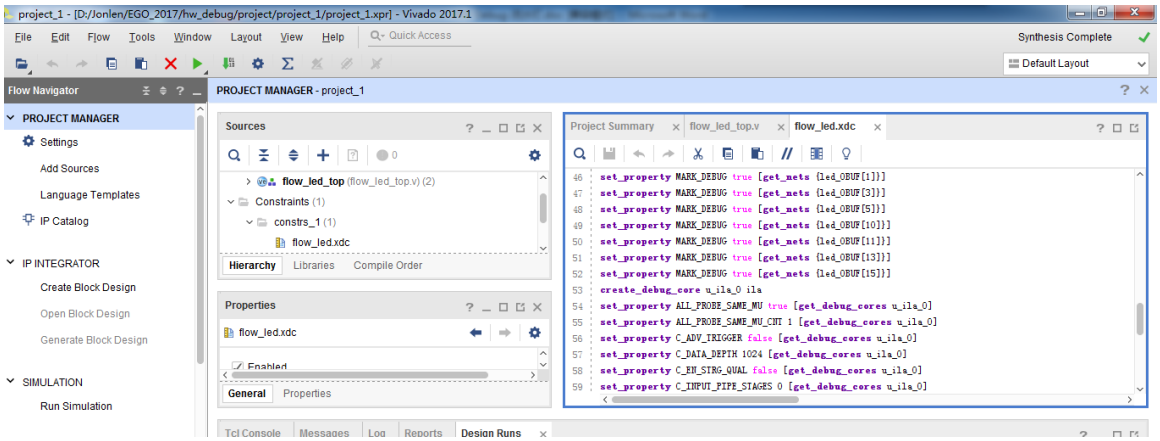
8、在 “Set Up Debug”向导中连续点击“Next”，最后点击“Finish”。


project_1 - [D:/Jonlen/EGO_2017/hw_debug/project/project_1/project_1.xpr] - Vivado 2017.1
Set Up Debug
Nets to Debug
The nets below will be debugged with ILA cores. To add nets click "Find Nets to Add". You can also select nets in the Netlist or other windows, then drag them to the list or click "Add Selected Nets".
Name
Clock Domain
Driver Cell
Probe Type
led_OBUF (16)
clk_IBUF_BUFG
FDRE
Data and Trigger
clk_pulse
Find Nets to Add...
Nets to debug: 17
< 上一步(B)
Next >
Finish
Cancel


project_1 - [D:/Jonlen/EGO_2017/hw_debug/project/project_1/project_1.xpr] - Vivado 2017.1
Set Up Debug
ILA Core Options
Choose features for the ILA debug cores.
Sample of data depth:
1024
Input pipe stages:
0
Trigger and Storage Settings
Capture control
Advanced trigger
< 上一步(B)
Next >
Finish
Cancel

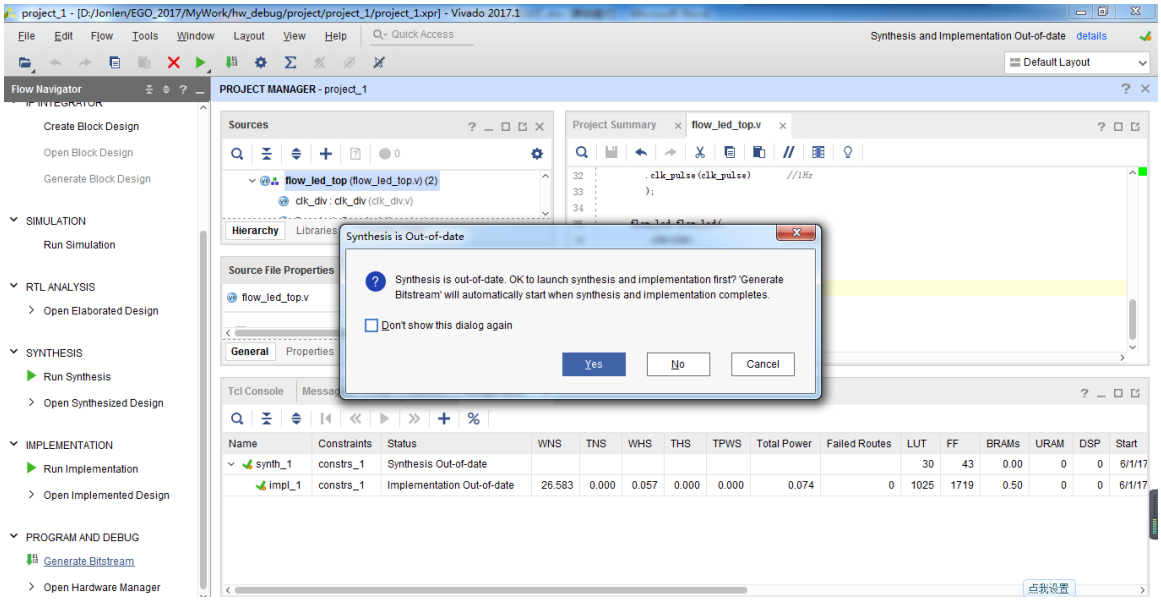

project_1 - [D:/Jonlen/EGO_2017/hw_debug/project/project_1/project_1.xpr] - Vivado 2017.1
Set Up Debug
VIVADO
HLx Editions
Set up Debug Summary
1 debug core will be removed: u_ila_0
1 debug core will be created
Found 1 clock
XILINX
Open in Debug layout
To apply the above changes, click Finish
< 上一步(B)
Next >
Finish
Cancel

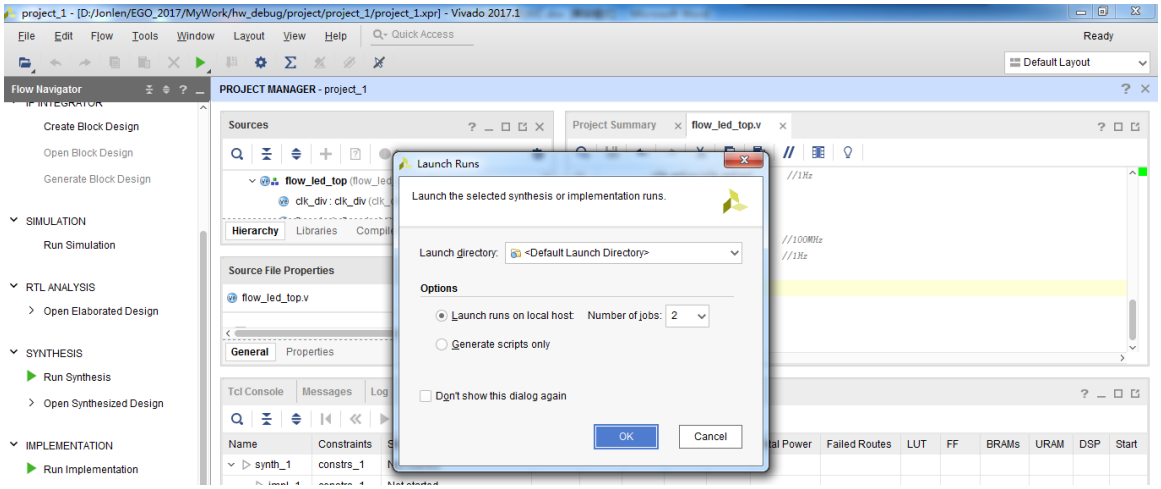
9、在菜单栏中点击“File”-> “Save Constraints”。在弹出的对话框中点击“OK”。然后在“Source”窗口中打开“flow_led.xdc”，在文档的底部可以看到“Mark Debug”及“Set Up Debug”的相关信息被添加上去了。



六、生成 bit 文件

10、在“Flow Navigator”一栏中的“Program and Debug”下点击“Generate Bitstream”，此时会提示工程没有实现，点击“Yes”，然后点击“OK”，系统会自动执行实现过程。



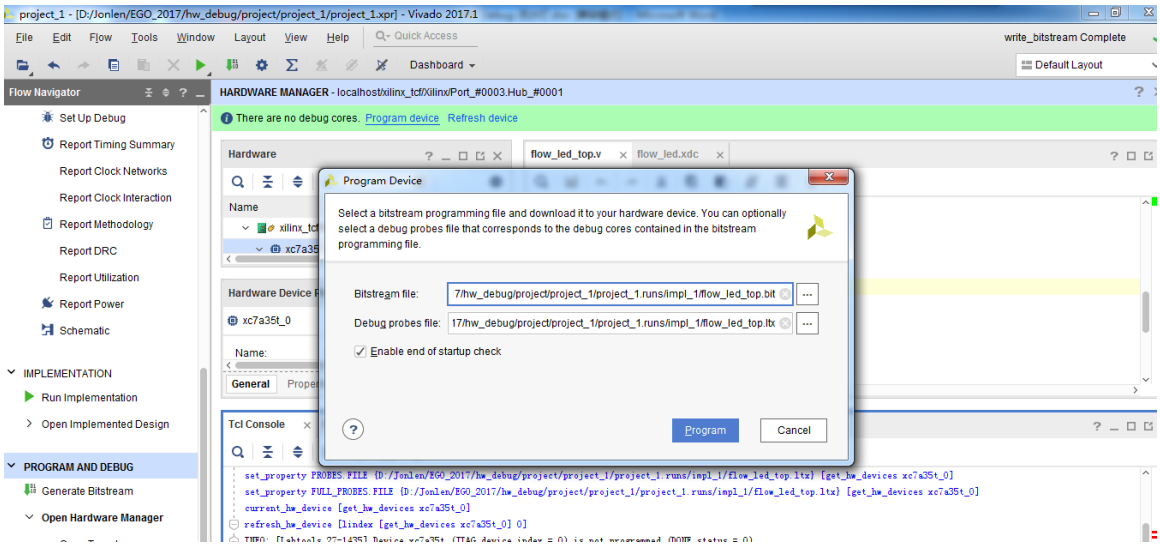


七、下载

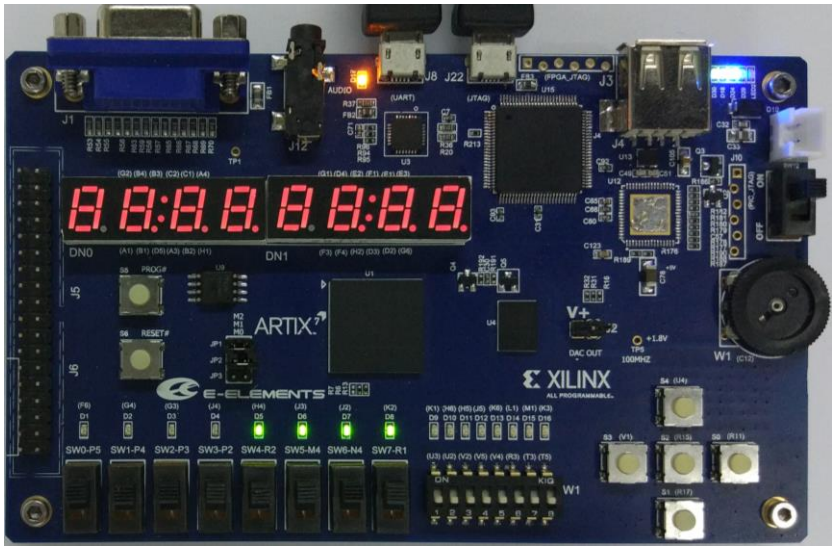
11、用 Micro USB 线连接电脑与板卡上的 JTAG 端口，打开电源开关。

12、生成比特流文件完成后，打开“Hardware Manager”。在“Hardware Manager”界面点击“Open target”，选择“Auto Connect”。

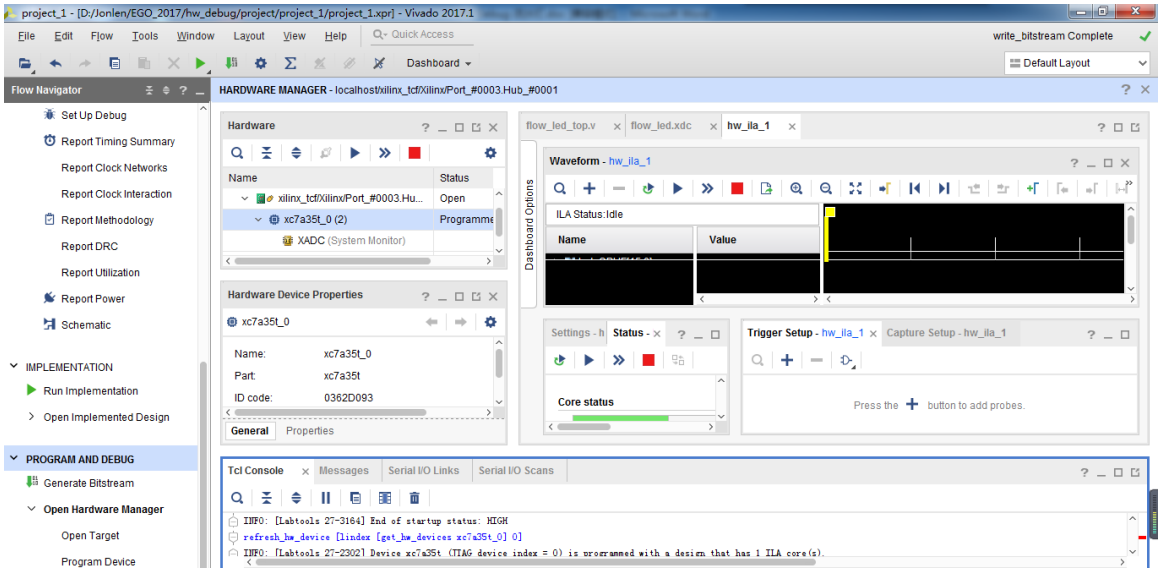
连接成功后，在目标芯片上右击，选择“Program Device”。在弹出的对话框中“Bitstream File”一栏已经自动加载本工程生成的比特流文件，点击“Program”对 FPGA 芯片进行编程。



13、下载完成后在板卡上可观察到四位 led 灯为一组从右向左循环点亮。

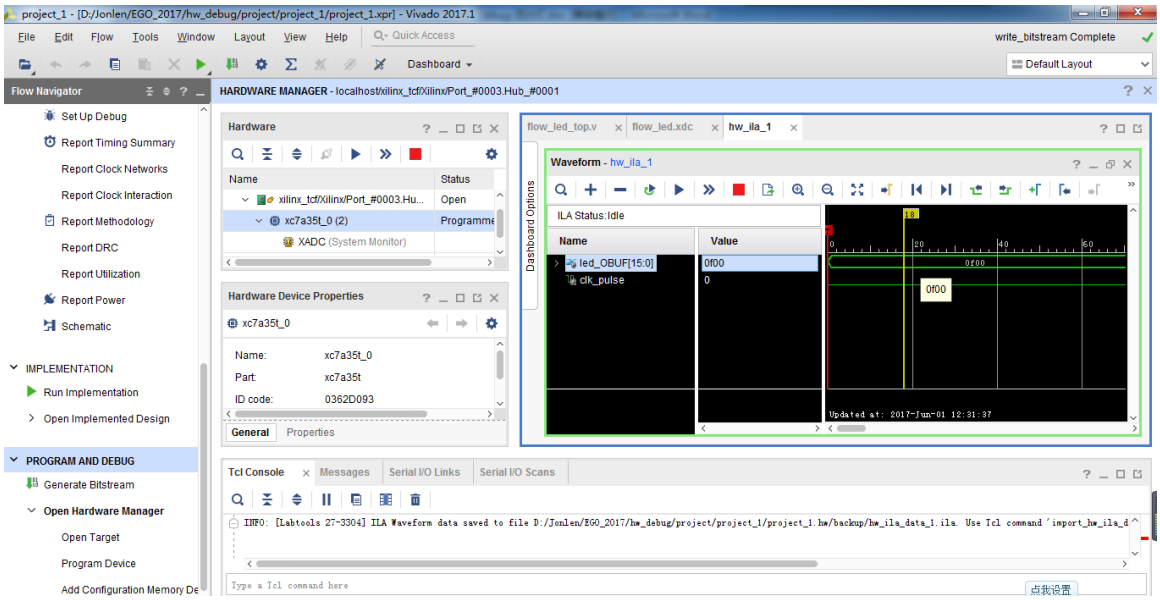


14、下载完成后“Hardware Manager”界面如下图所示。

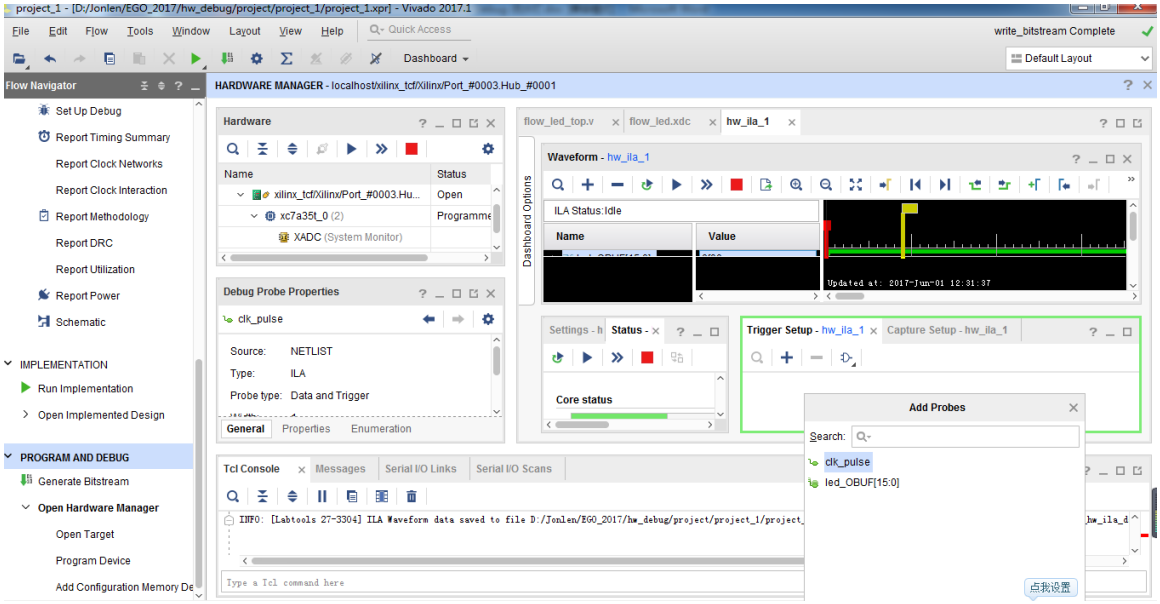


八、Hardware Debug

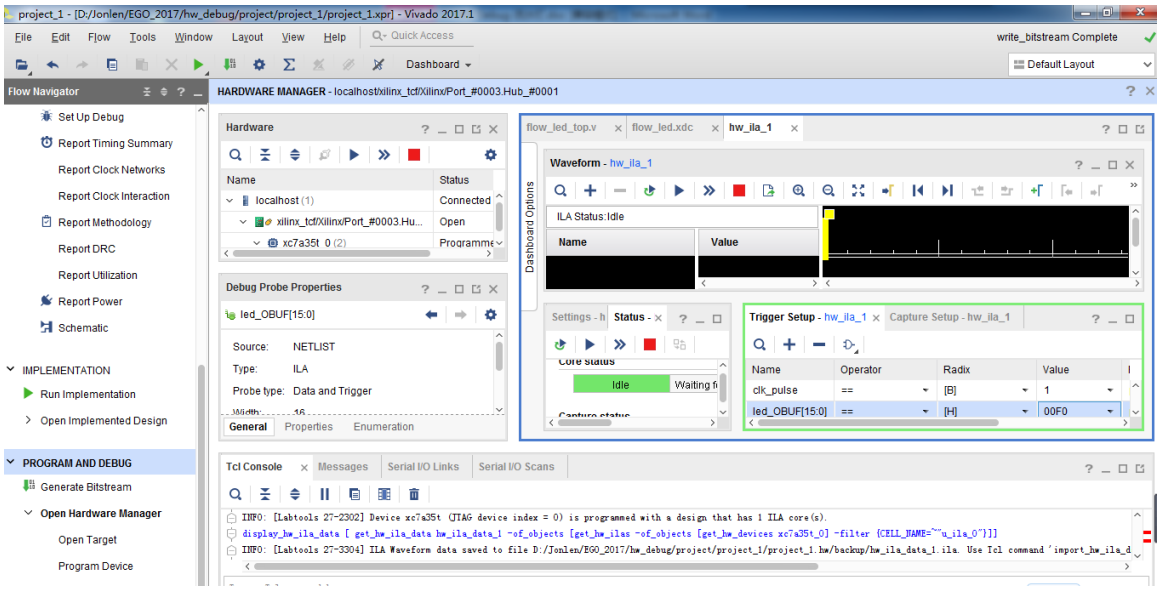
15、选中目标芯片，点击上方的“Run Trigger Immediate”，在波形窗口中可看到触发信号。



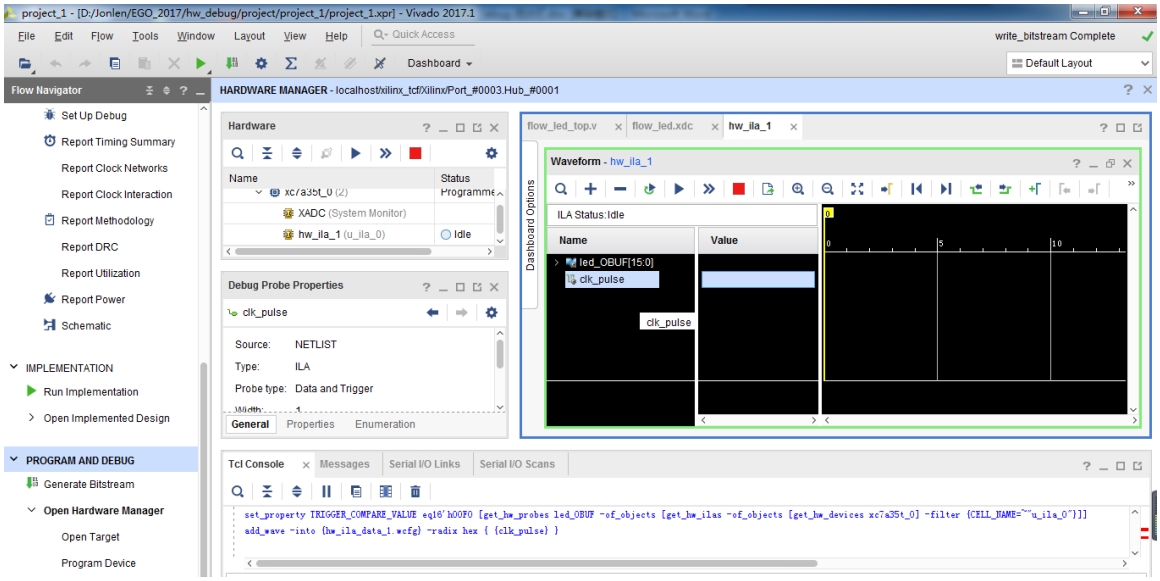
16、在“Trigger Setup”窗口中点击加号，选中“clk_puse”，双击添加到窗口中。



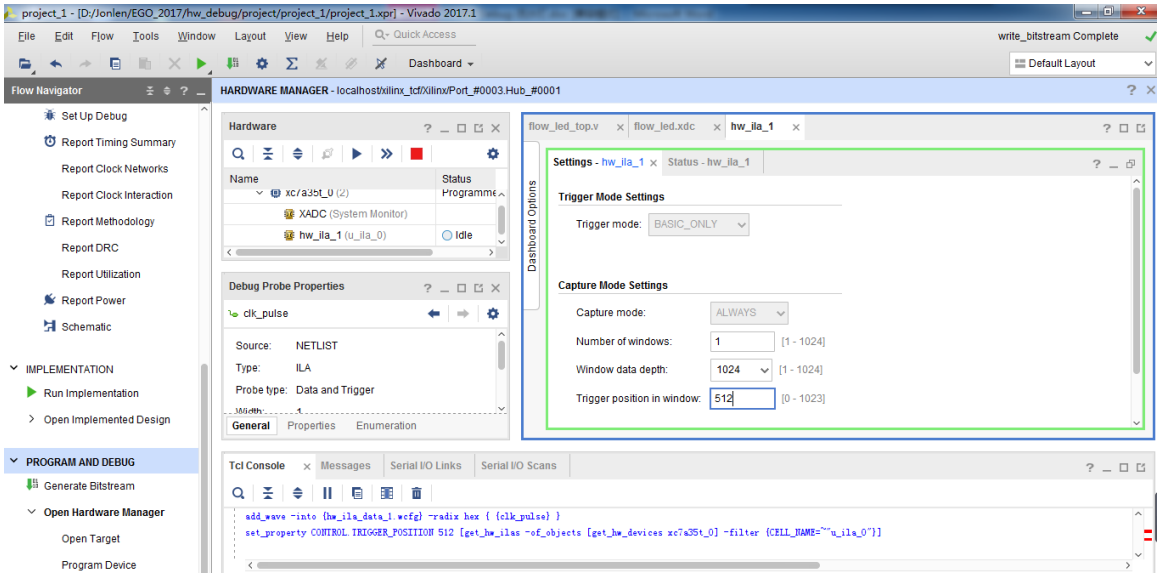
17、在“Trigger Setup”窗口中将“led_OBUF”的“Compare Value”设为“[H] 00F0”，将“clk_puse”的“Compare Value”设为 1。



18、在“Waveform”窗口点击加号，在“Add Probes”列表中双击“clk_pulse”将其添加到波形窗口中。



19、在“hw_ila_1”的“Settings”窗口中，将“Trigger position in window”一栏设为 512.



20、在“Hardware”窗口选中“hw_ila_1”，然后点击上方的“Run Trigger”按钮。 在“Status”窗口可观察到状态由“Idle”跳转到“ Waiting for Trigger”。

当状态跳转到“Full”后回到“Idle”状态，此时将波形图中红色竖线标注出了触发的时刻。将波形图放大之后，可以看到触发时刻的信号“led_OBUF”和“clk_pulse”与设置的触发条件一致。