

1 FPGA 在线调试说明

在 FPGA 开发过程中，经常碰到仿真上板行为不一致，更多时候是仿真正常，上板异常。由于上板调试手段薄弱，导致很难定位错误。这时候可以借助 Xilinx 的下载线进行在线调试，在线调试是在 FPGA 上运行过程中探测预定好的信号，然后通过 USB 编程线缆显示到上位机上。

本文档给出简单使用在线调试的方法：RTL 里设定需探测的信号，综合并建立 Debug core，实现并生产 bit 流文件，下载 bit 流和 debug 文件，上板观察。

1.1 抓取需探测的信号

在 RTL 源码中，给想要抓取的探测信号声明前增加(*mark_debug = "true"*)。

比如，我们想要在 FPGA 板上观察 debug 信号、PC 寄存器和数码管寄存器，需要在代码里这样设置。

```
//debug interface
(*mark_debug = "true"*)output [31:0] debug_wb_pc;
(*mark_debug = "true"*)output [3:0] debug_wb_rf_wen;
(*mark_debug = "true"*)output [4:0] debug_wb_rf_wnum;
(*mark_debug = "true"*)output [31:0] debug_wb_rf_wdata;
wire inst_ex_dib;
(*mark_debug = "true"*)reg [31:0] inst_pc_r;
reg [31:0] inst_code_r;
```

(1) 抓取写回信息

(2) 抓取 PC 寄存器

```
4 reg [31:0] led_rgo_data;
5 reg [31:0] led_rgl_data;
6 (*mark_debug = "true"*) reg [31:0] num_data;
7 wire [31:0] switch_data;
8 wire [31:0] btn_key_data;
```

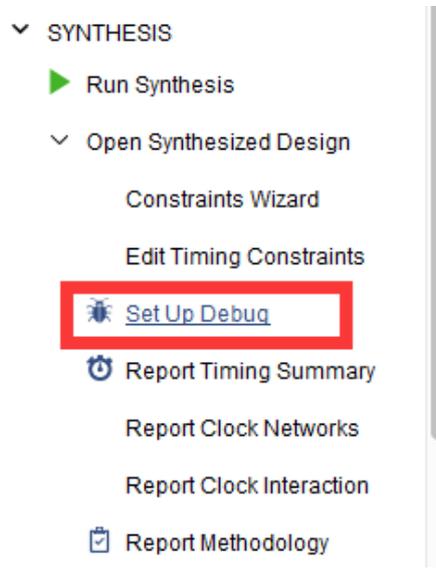
(3) 抓取数码管寄存器

设定完成后，就可以运行综合了。

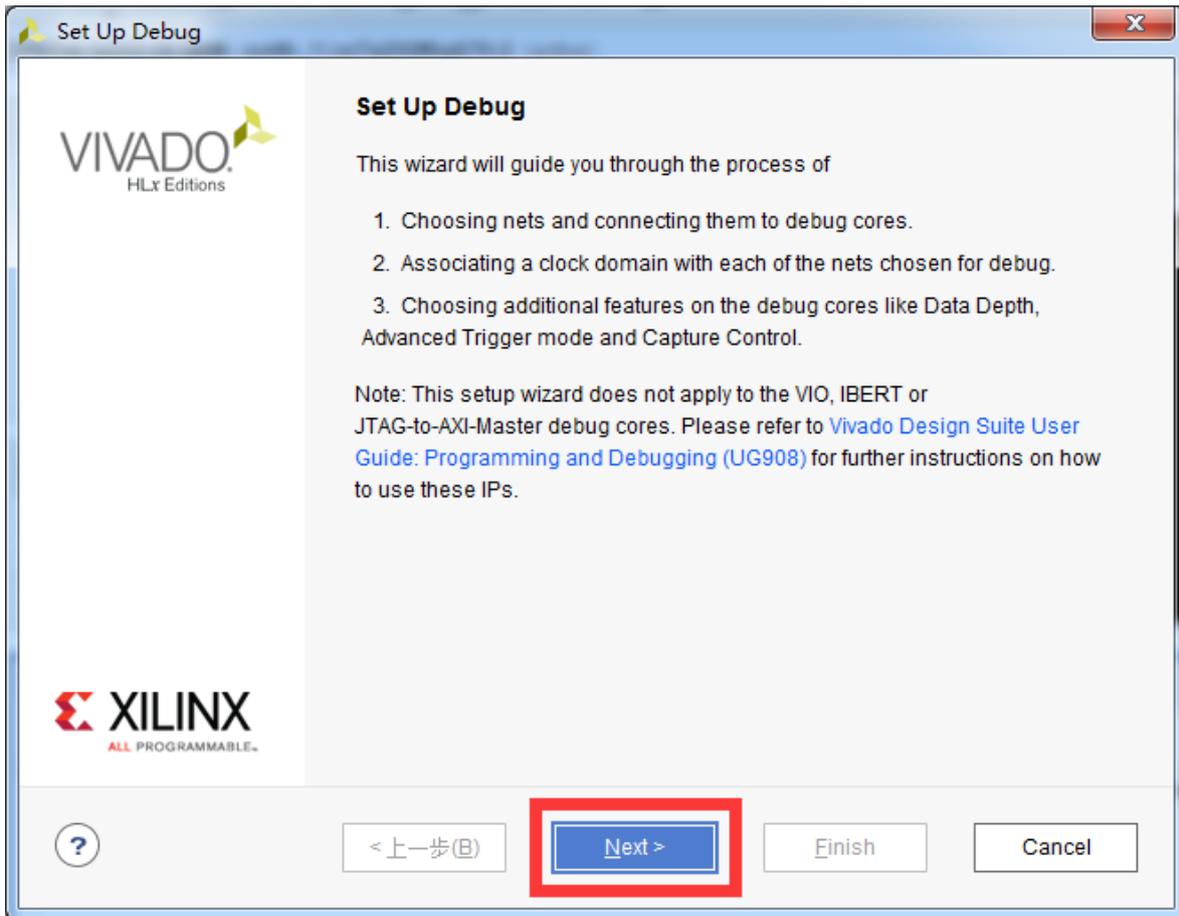
1.2 综合并建立 debug

在综合完成后，需建立 debug。

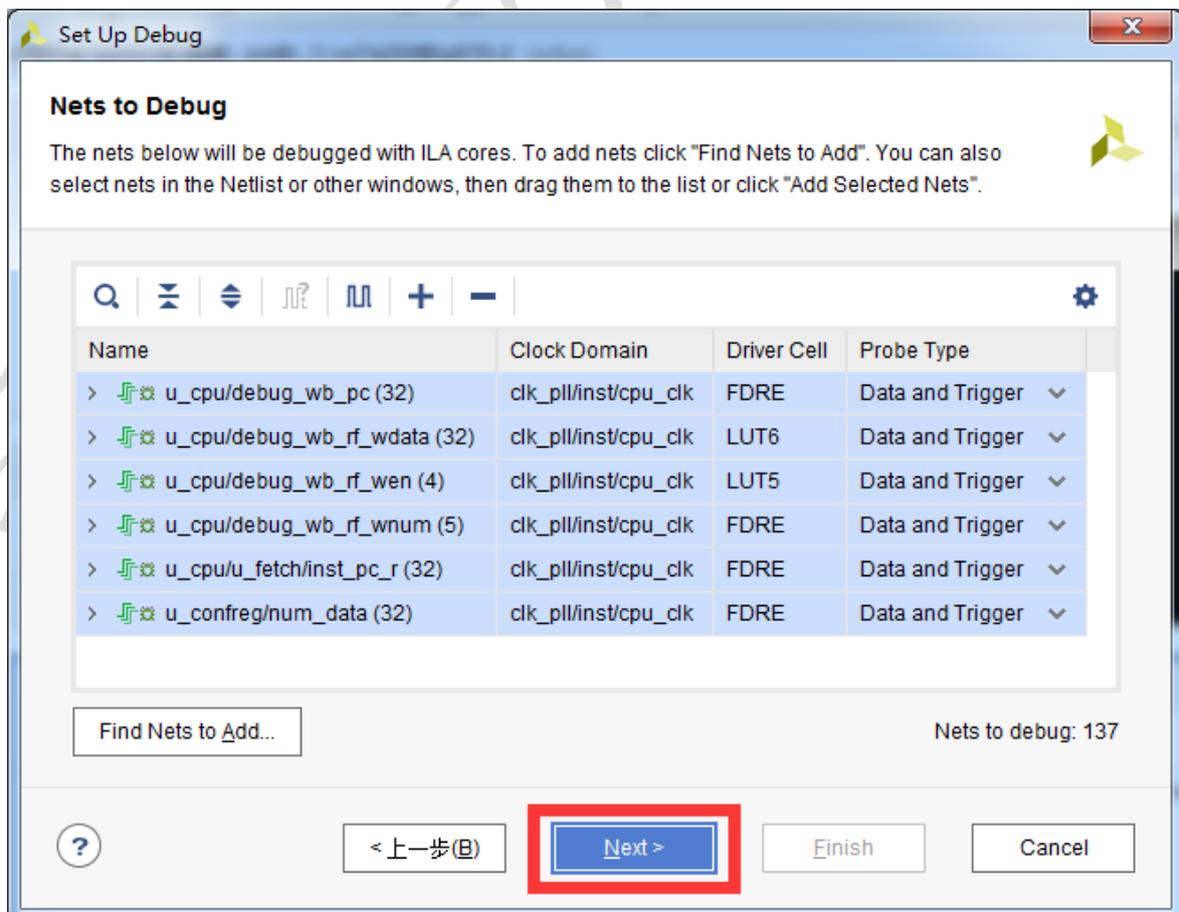
点击工程左侧 synthesis->Open Synthesized Design->Set Up Debug。



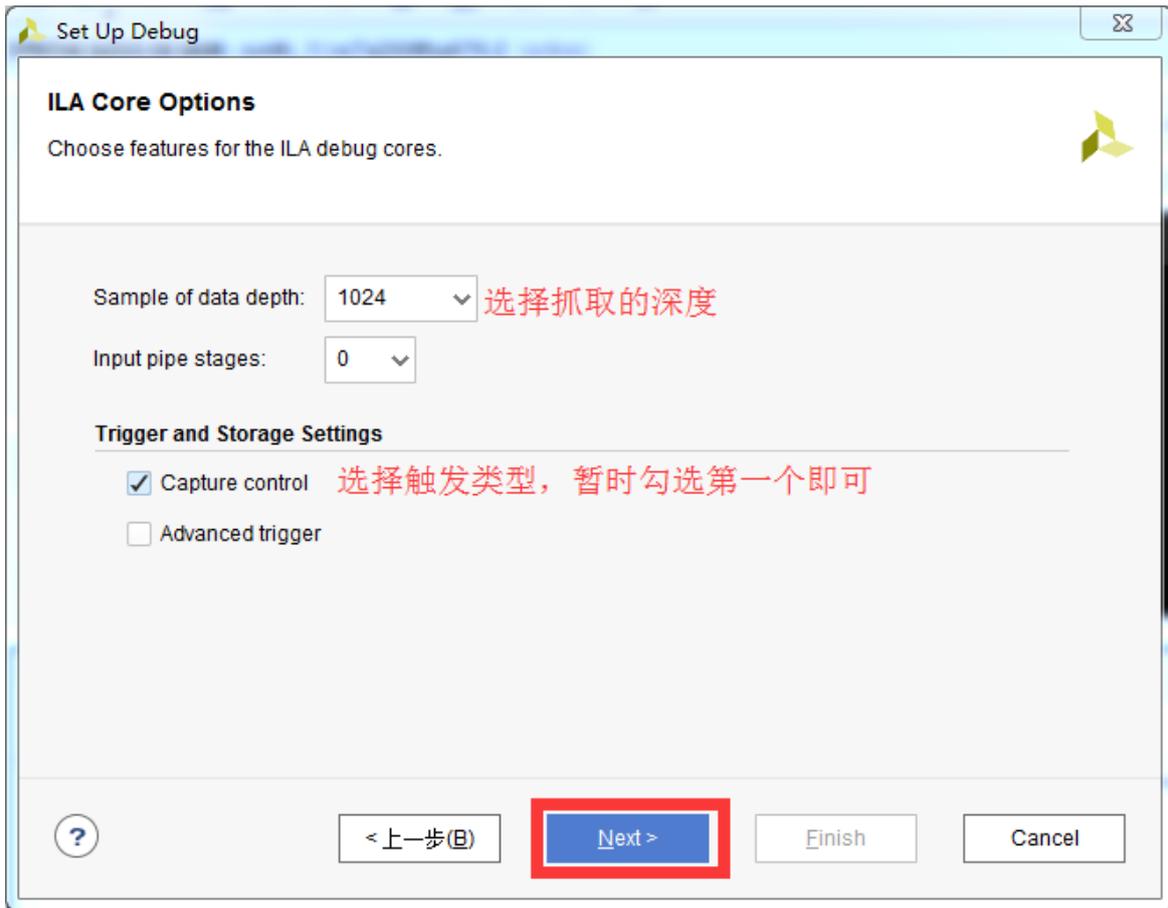
随后会出现如下界面，点击 Next:



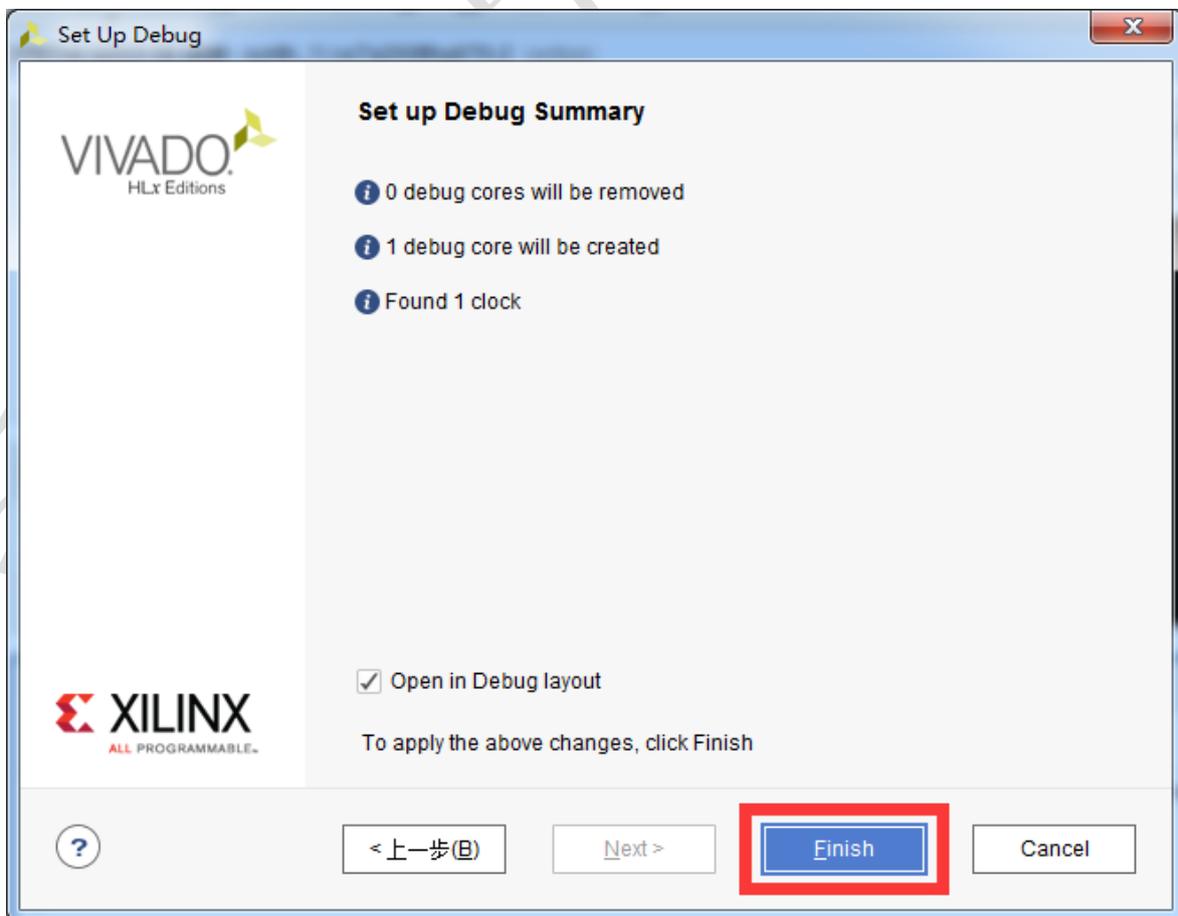
随后会列出抓取的 Debug 信息，点击 Next



选择抓取的深度和触发控制类型，点击 Next（更高级的调试可以勾选“Advanced trigger”）：

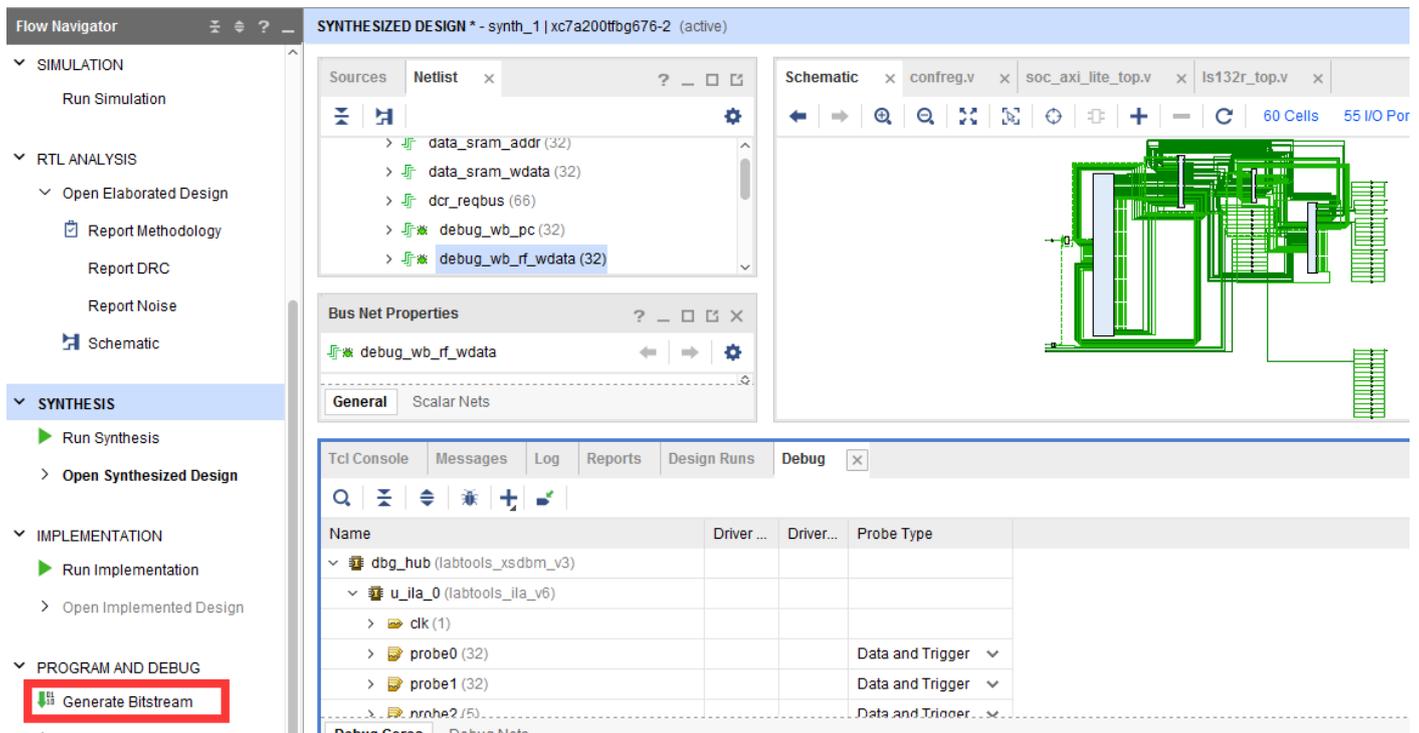


最后，点击 Finish:

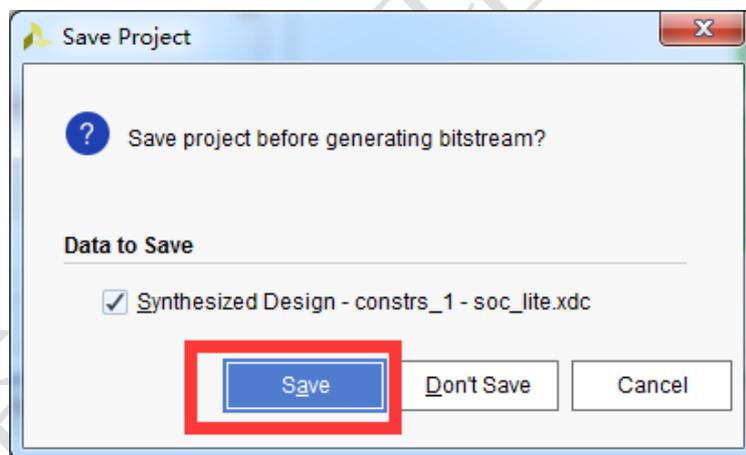


1.3 实现并生产 bit 流文件

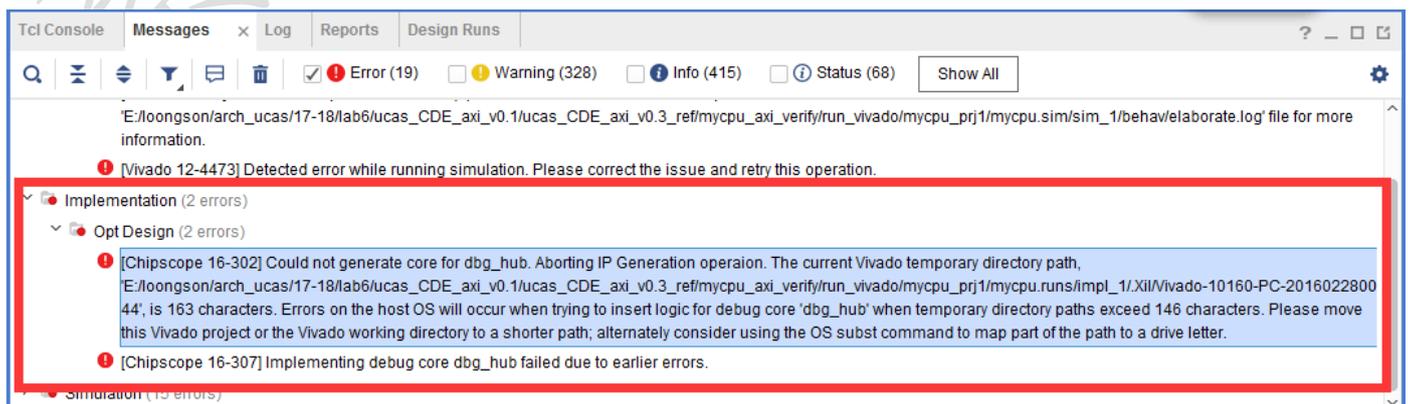
在 1.2 节完成后会出现类似下图界面，直接点击 **Generate Bitstream**：



弹出如下界面，点击 **Save**：

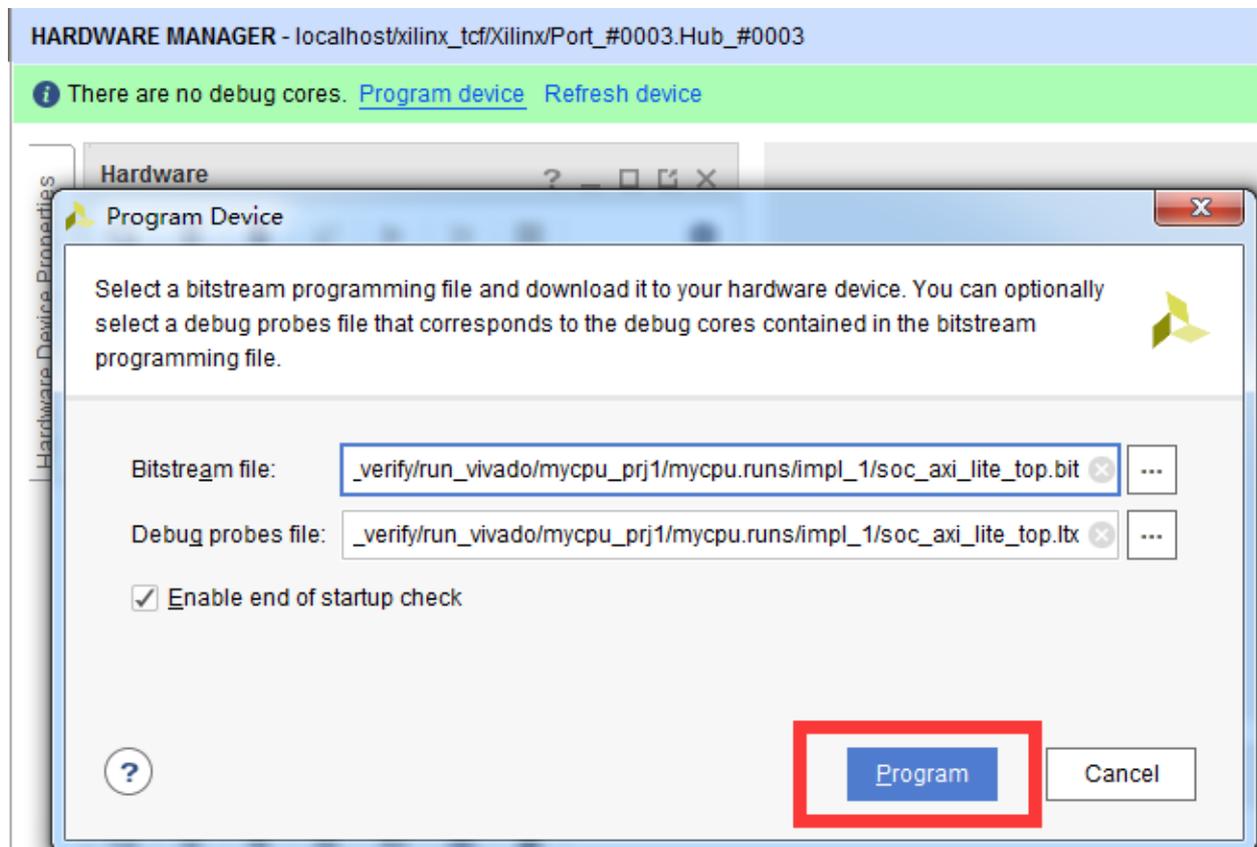


如果有后续弹出界面，继续点击 **OK** 或 **Yes** 即可。这时就进入后续生产 bit 文件的流程了，此时 Vivado 界面里的 **synthesis design** 界面就可以关闭了。如果发现以下错误，则是因为路径太深，引用起来名字太长，降低工程目录深度即可：



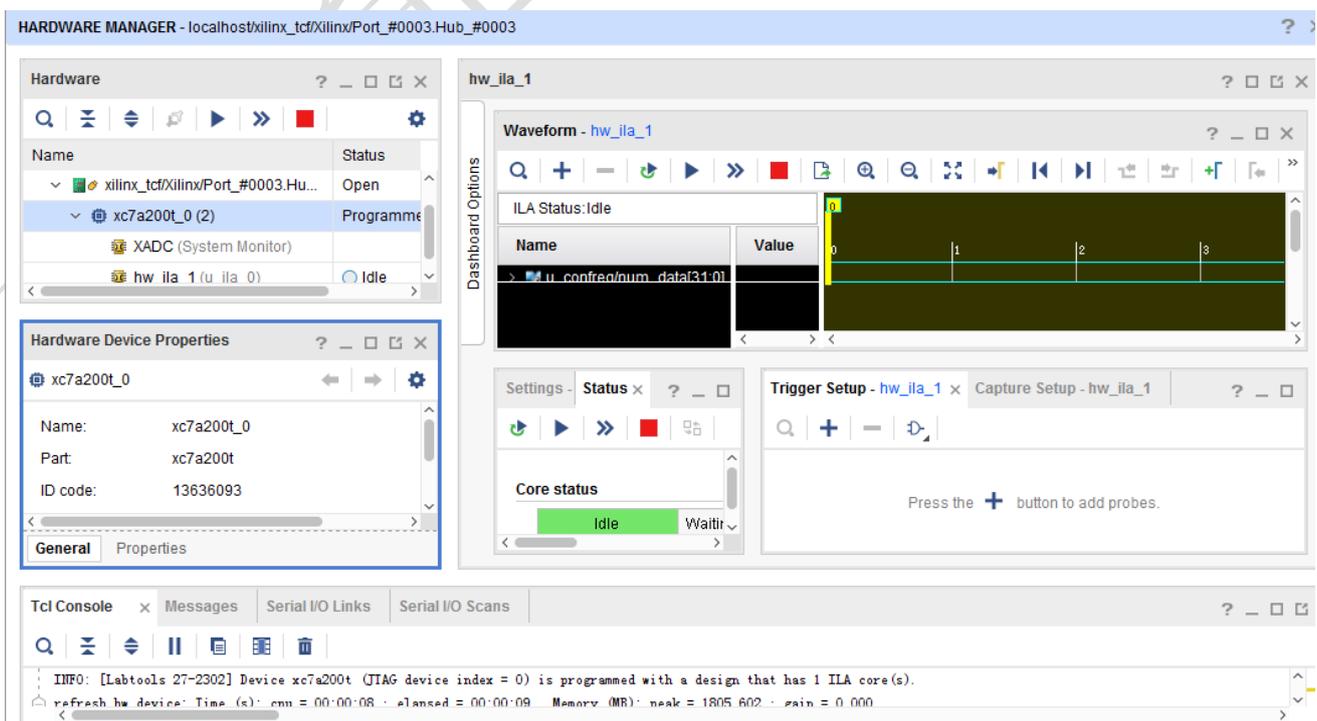
1.4 下载 bit 流和 debug 文件

在完成 1.3 节后，会生成 bit 流文件和调试使用 ltx 文件。这里，打开 Open Hardware Manager，连接好 FPGA 开发板后，选择 Program device，如下图。自动加载了 bit 流文件和调试的 ltx 文件。选择 Program，等待下载完成。

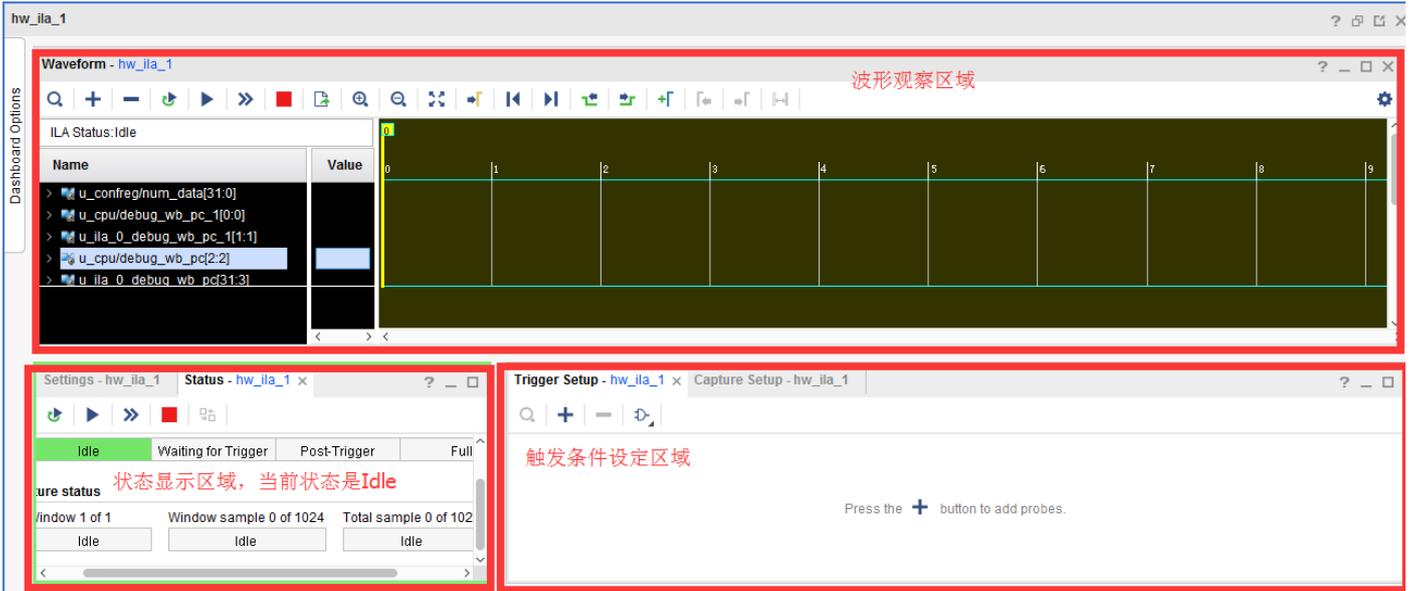


1.5 上板观察

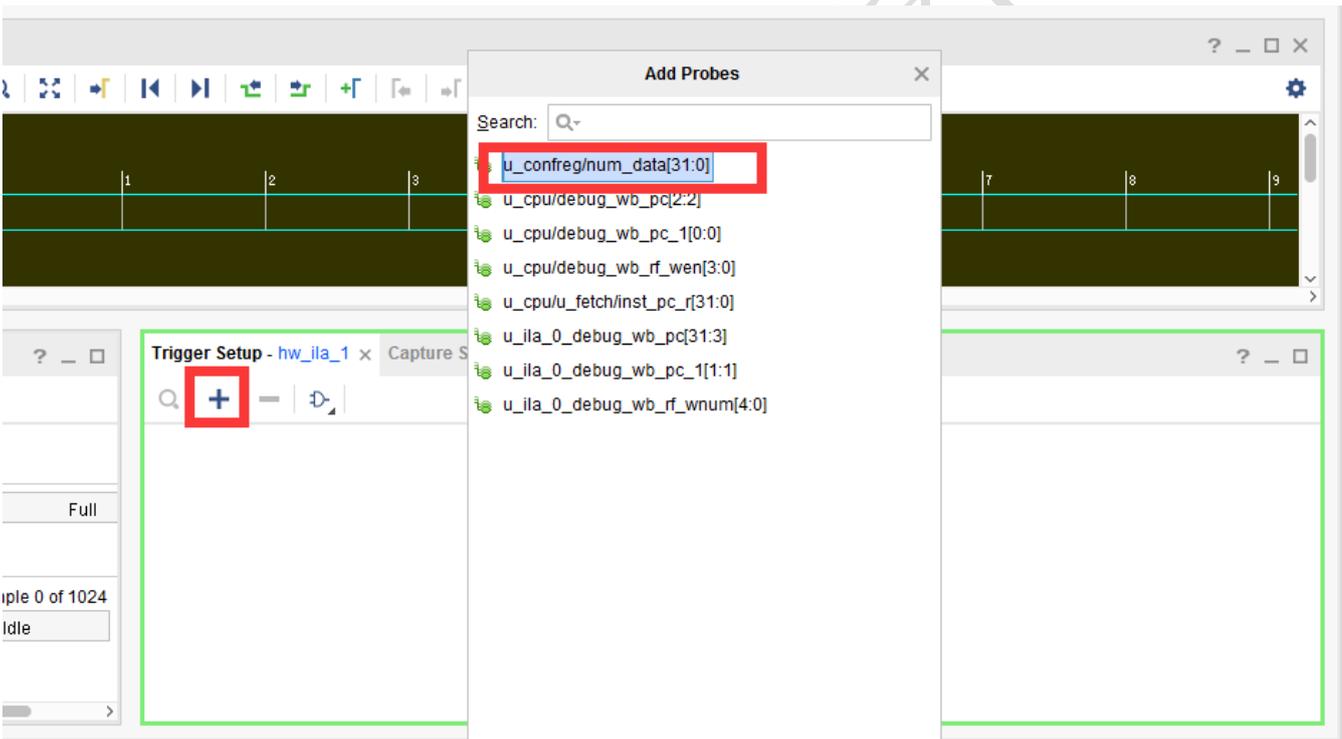
在下载完成后，vivado 界面如下，在线调试就是在 hw_ila_1 界面里进行的。



hw_ila_1 界面主要分为 3 个界面，分别如下：



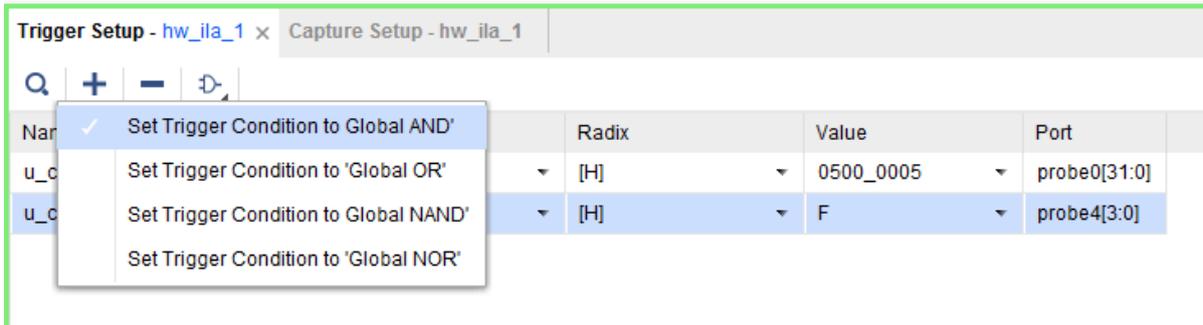
首先，我们需要在右下角区域设定触发条件。所谓触发条件，就是设定该条件满足时获取波形，比如我先设定触发条件是数码管寄存器到达 0x0500_0005。在下图中，先点击“+”，在双击 num_data。



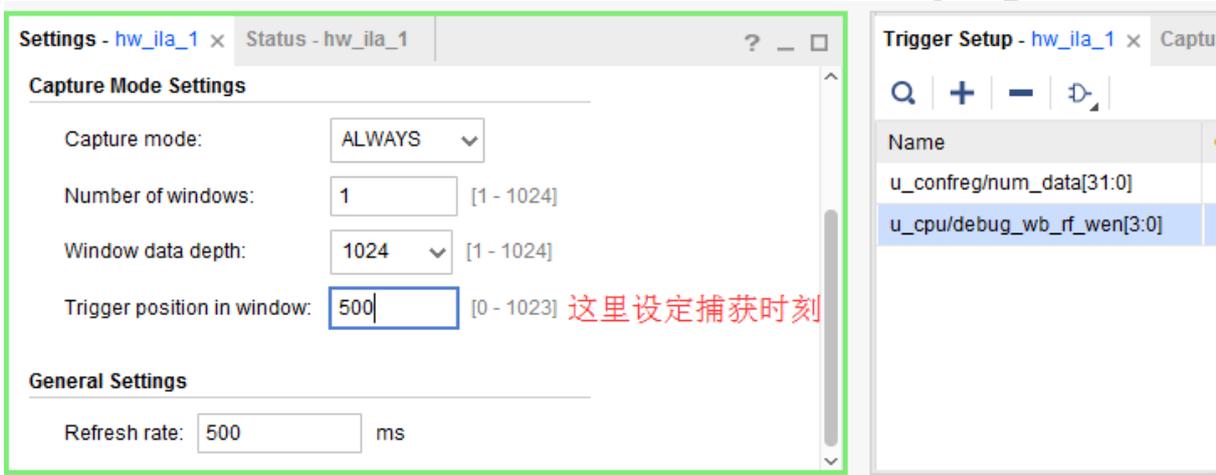
之后会出现如下界面，设定好触发条件。



可以设定多个触发条件，比如，再加一个除法条件是写回使能是 0xf，可以设定多个触发条件直接的关系，比如是任意一个满足、两个都是满足等等，如下图：



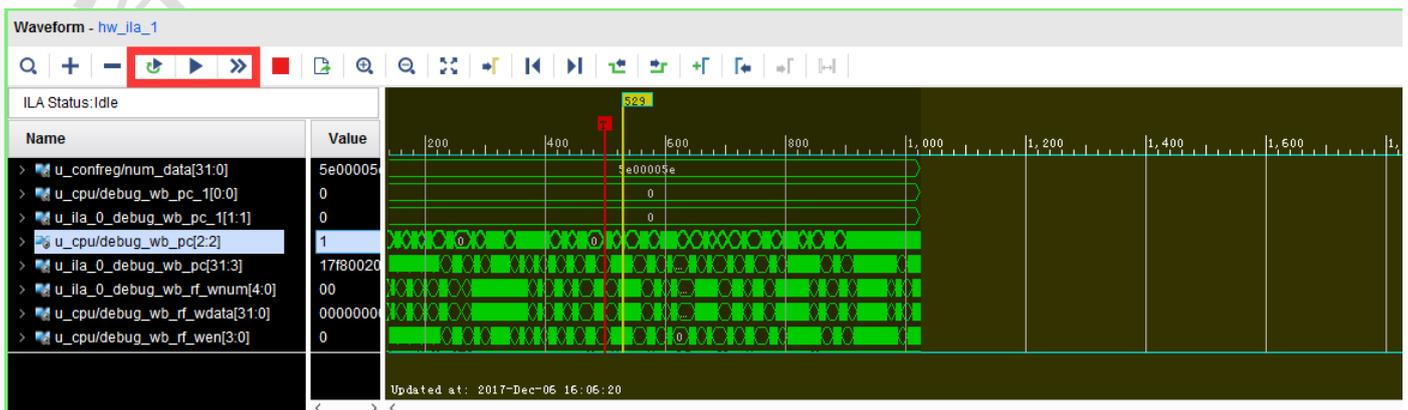
在左下角窗口，选择 settings，可以设定 Capture 选项，可能经常用到的是 Trigger position in window，用来设定触发条件满足的时刻在波形窗口的位置。比如，下图设定为 500，当触发条件满足时，波形窗口的第 500 个 clk 的位置是该条件，言下之意，将触发条件满足前的 500 个 clk 的信号值也抓出来了，这样可以看到触发条件之前的电路行为。Refresh rate 设定了波形窗口的刷新频率。



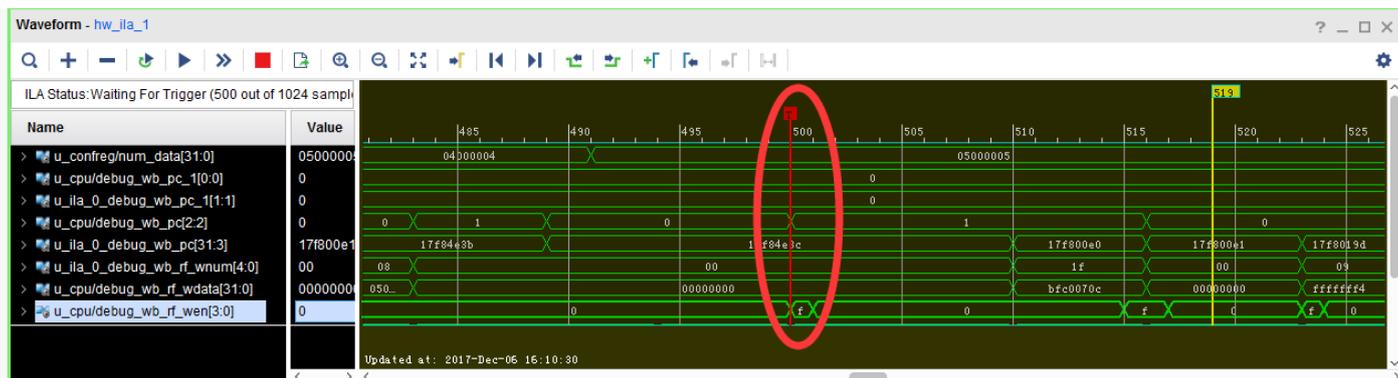
触发条件建立后，就可以启动波形抓取了，最关键的有三个触发按键，即下图圈出的 3 个按键：

- 左起第一个，设定触发模式，有两个选项：单触发；循环触发。当该按键按下时，表示循环检测触发，那么只要触发条件满足，波形窗口就会更新。当设置为单触发时，就是触发一次完成后，就不会再检测触发条件了。比如，如果我们设定触发条件是 PC=0xbf00690，那么如果该 PC 被多次执行到。如果设定为单触发，那按下 FPGA 板上的复位键，波形窗口只会展示第一次触发时的情况。如果设定循环触发，那么波形窗口会以 Refresh rate 不停刷新新捕获的触发条件。
- 左起第二个，等待触发条件被满足。点击该按键，就是等待除法条件被满足，展示出波形。
- 左起第三个，立即触发。点击该按键，表示不管触发条件，立即抓取一段波形展示到窗口中。

下图就是点击第三个按键得到的波形，因为是立即触发，所以 num_data 不是 0x0500_0005，且有一条标注为“T”的红色，就是触发的时刻。由于触发时刻位于波形窗口的 500 clk 位置，所以红色的位置正好是 500 clk 处。



从上图也能看到，num_data 是 0x5c00_005c，表示一次测试已经完成了。所以这时候，点击第二个触发按键等待触发，会发现波形窗口没有反应。这是因为触发条件没有被满足，这时按下 FPGA 板上的复位键即可。结果如下图。红色圈出的就是触发条件：num_data==32'h5c00_005c && rf_wen==4'hf。



剩下的 debug 过程，就和仿真 debug 类似了，去观察波形。但在线调试时，你无法添加在 1.1 节未被添加 debug mark 的信号。在线调试过程中，可能需要不停的更换触发条件，不停的按复位键。

1.6 注意事项

在 1.1 节中添加要抓取的信号时，不要给太多信号标注 debug mark 了。在线调试时抓取波形是需要消耗电路资源和存储单元的，因而能抓取的波形大小是受限的。应当只给必要的 debug 信号添加 debug mark。

在 1.2 节中抓取波形的深度，不宜太深。如果设定得太深，那么会存在存储资源不够，导致最后生成 bit 流和 debug 文件失败。

抓取的信号数量和抓取的深度是一对矛盾的变量。如果抓取深度相对较低，抓取的信号数量就可以相对多些。

在 1.5 节中，相对仿真调试，在线调试对调试思想和技巧有更高的要求，请好好整理思路，多多总结技巧。特别强调以下几点：

- 触发条件的设定有很多组合，请根据需求认真考虑，好好设计。
- 通常只需要使用单触发模式，但循环触发有时候也很有用，必要时好好利用。
- 在线调试界面里很多按键，请自行学习，可以网上搜索资料，xilinx 官网上搜索，查找官方文档等。

最后再提醒一点，仿真通过但上板失败时，请先重点排查其他问题，最后再使用在线调试的方法。也就是仿真通过，上板异常时，应按以下流程排查：

- 1) 复核生成、下载的 bit 文件是否正确。
- 2) 复核仿真结果是否正确。
- 3) 检查实现时的时序报告（Vivado 界面左侧“IMPLEMENTATION”“Open Implemented Design”“Report Timing Summary”）。
- 4) 认真排查综合和实现时的 Warning
- 5) 排查 RTL 代码规范，避免多驱动、阻塞赋值乱用。
- 6) 使用 Vivado 的逻辑分析仪进行板上在线调试