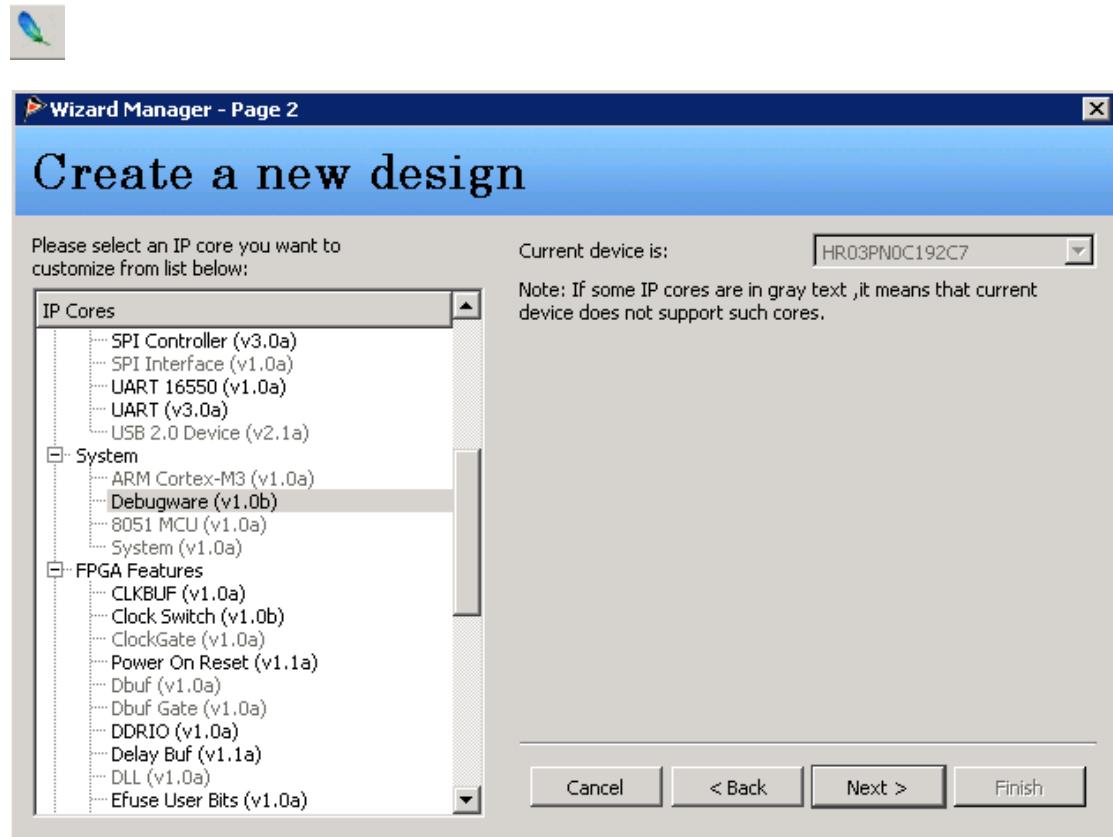
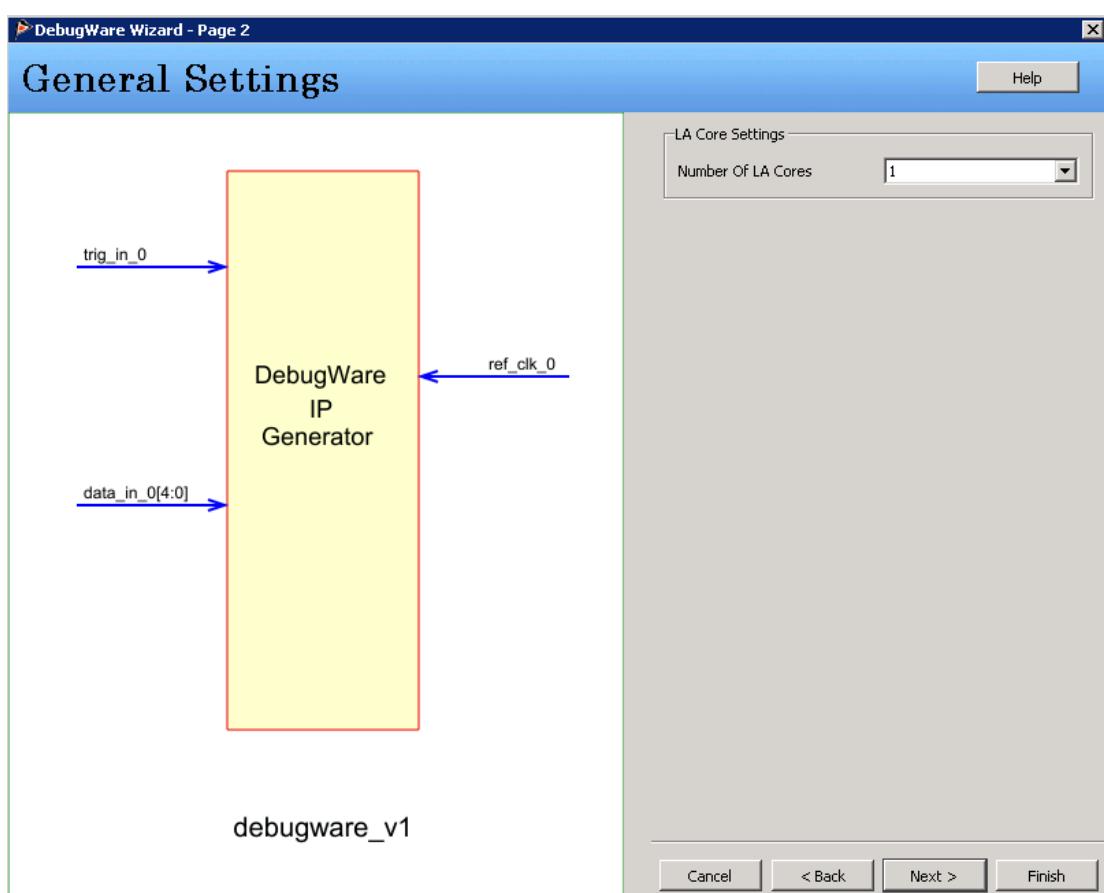
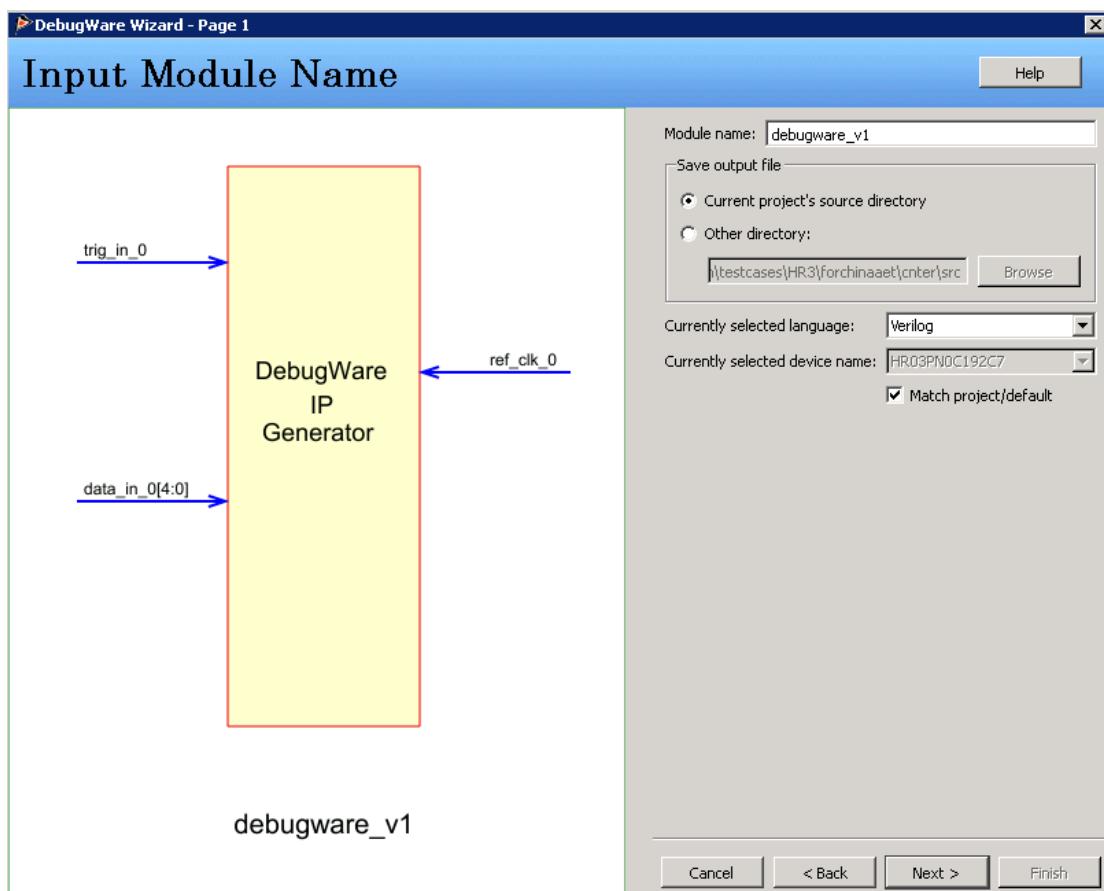


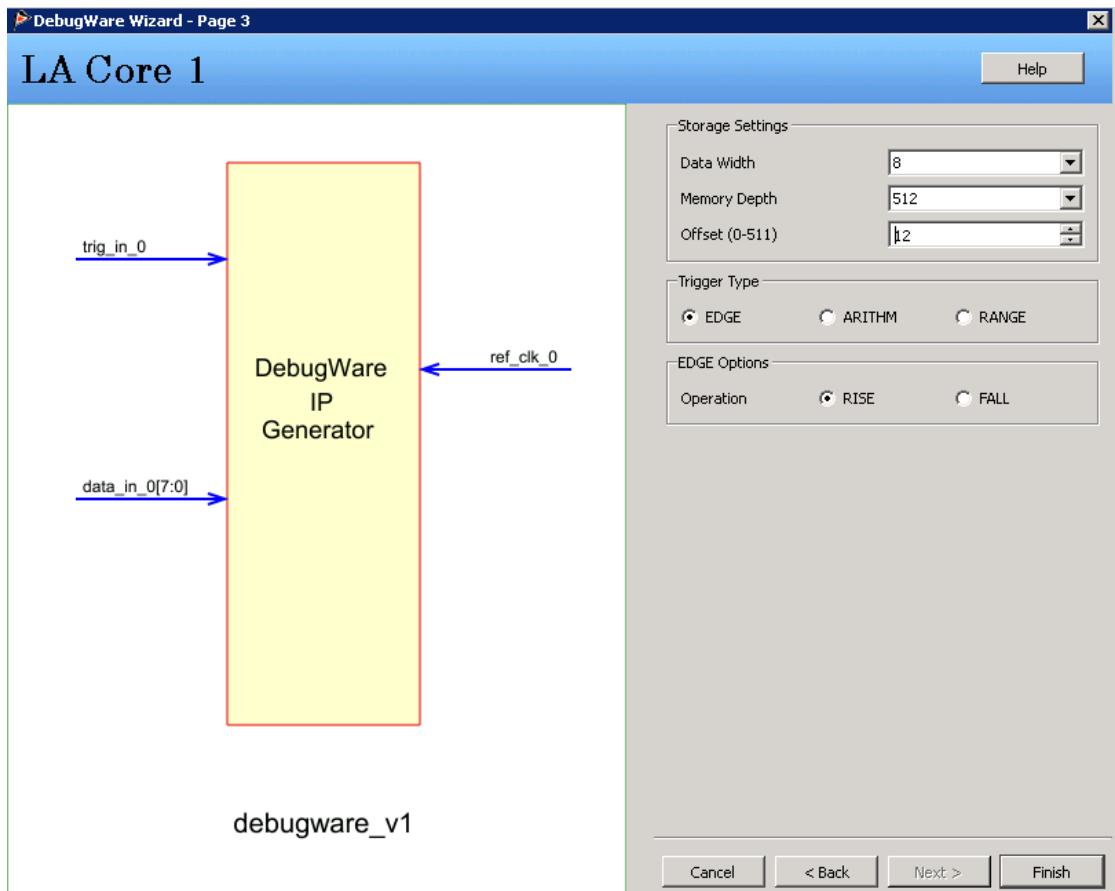
Debugware 是一个嵌入式逻辑分析仪, 设计者有时候会希望探测 FPGA 内部的信号传输情况, Debugware 使用 EMB(FPGA 内部的 memory)存储这些信号波形, 当触发事件发生后, 通过 JTAG 从这些 EMB 中读取波形数据。

最简单的例子, 比如一个 counter, 我们知道它的数据是累加的, 可以降低频率把数据接到 LED 灯上来看这些信号的变化, 也可以通过把某些位的信号连接到 debugware 的 data 端口, 通过设置相应的触发条件来 dump 波形来分析数据。

现在, 我们先用 wizard 例化一个 debugware.







最后 dialog 设置页面，从左边图中也可以看出我们设置的 Data Width 宽度为 8 (`data_in_0[7:0]`)，触发信号 (`trig_in_0`) 类型为上升沿触发。`ref_clk_0` 为向 EMB 写数据的时钟。

然后，我们在我们的 cnter 例子中实例化一个 debugware,

The screenshot shows the Quartus II software interface. The top menu bar includes File, Edit, View, Project, Flow, Tools, Window, and Help. Below the menu is a toolbar with various icons. The main area has two tabs: 'src/cnter.v' and 'src/debugware_v1.v'. The 'src/debugware_v1.v' tab is active, displaying the following VHDL code:

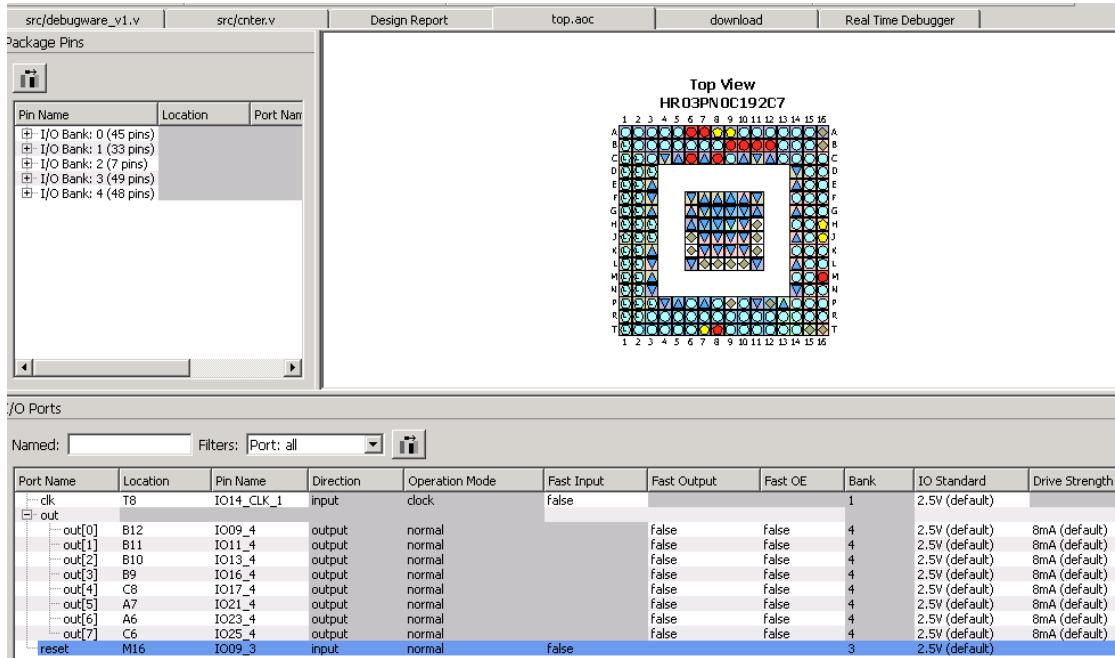
```
1 module top(out, clk, reset);
2   output [7:0] out;
3   input clk;
4   input reset;
5   reg [7:0] out;
6   reg [31:0] count;
7
8   debugware_v1 mydw(
9     .trig_in_0(count[25]),
10    .data_in_0(count[7:0]),
11    .ref_clk_0(clk)
12 );
13
14 always @ (negedge reset or posedge clk) begin
15   if (!reset)
16     begin
17       count <= 0;
18     end
19   else
20     begin
21       count <= count + 1;
22     end
23
24 end
25
26 assign out[7:0] = count[31:24];
27
28 endmodule
29
```

On the left, there is a project tree labeled 'project' containing a 'cnter' folder with files 'cnter.v' and 'top', and a file 'debugware_v1.v'. Below the code editor is a 'Progress View' window with a 'Files' tab selected, showing the following table:

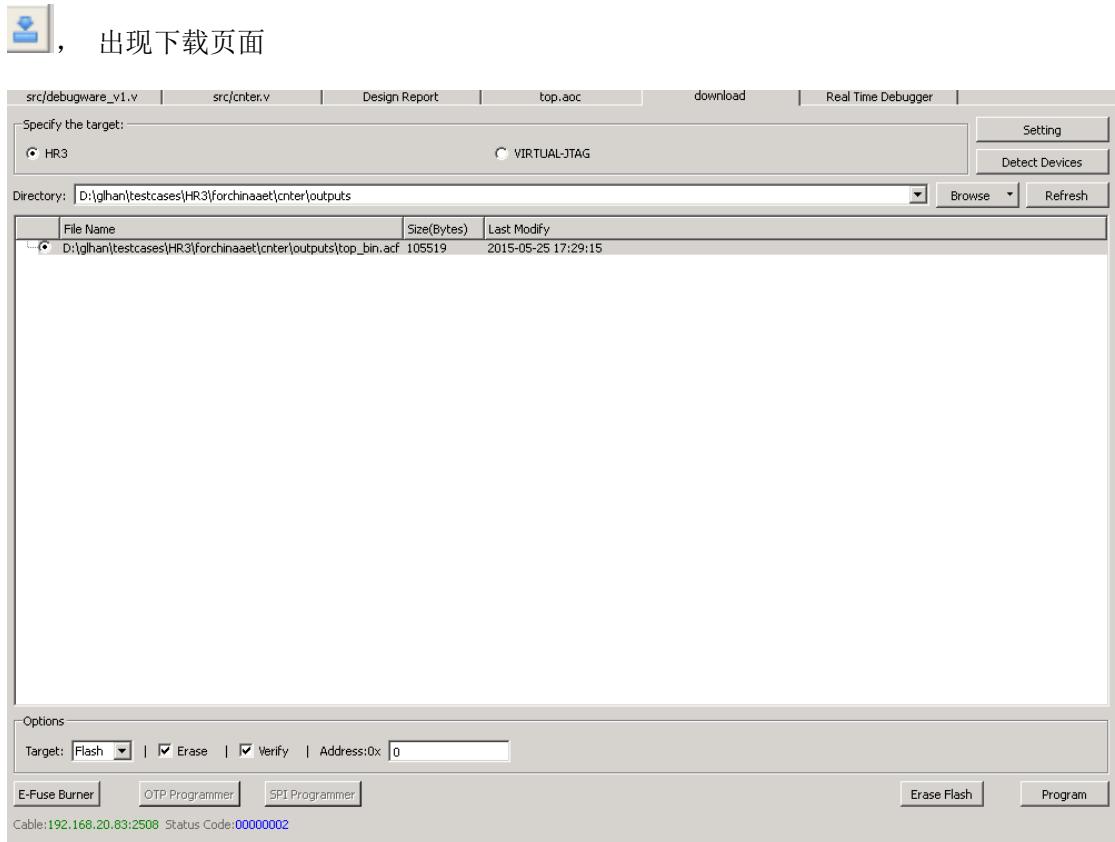
| Process | Progress | Time |
|------------|----------|-------|
| Placement: | 100% | 00:05 |
| Routing: | 100% | 00:08 |
| Bitgen: | 100% | 00:02 |
| Download: | 100% | 00:13 |

同 counter 一样，其他用户逻辑也可以通过把某些信号连接到 debugware 的 data 端口，设置相应触发条件来看数据。

综合（Synthesis）完成之后，点击 ioEditor 分配 IO。需要查看 PCB 原理图。



这样，跑完 Flow， 需要点击

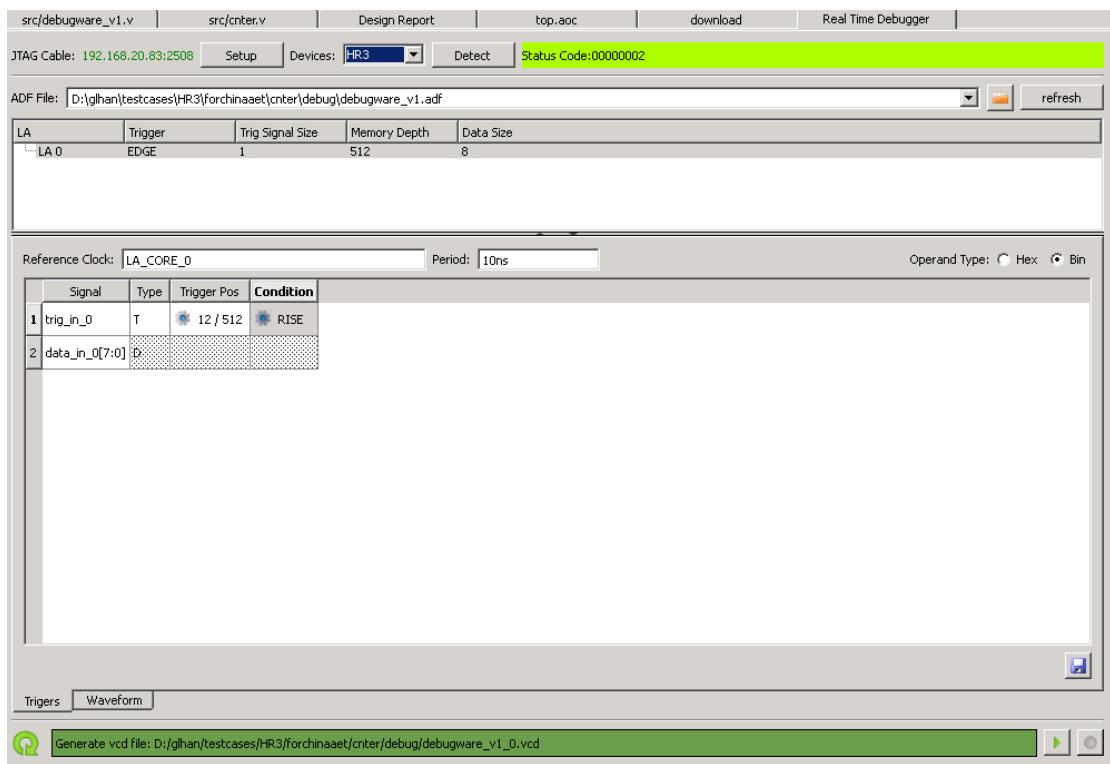


Debugware 模块的使用需要一个开发板。连接好 JTAG 线，点击按钮 Program 下载。

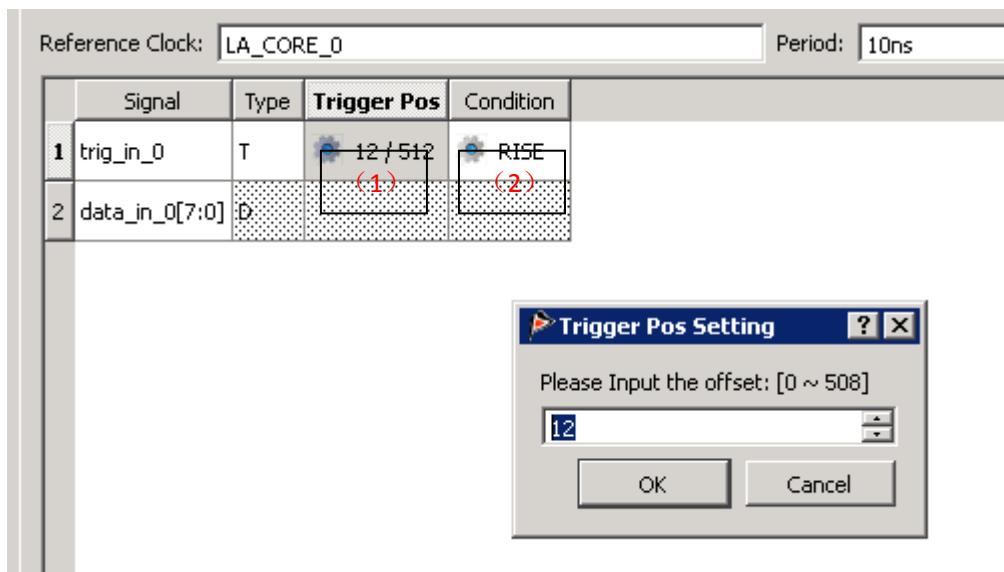
然后我们点击



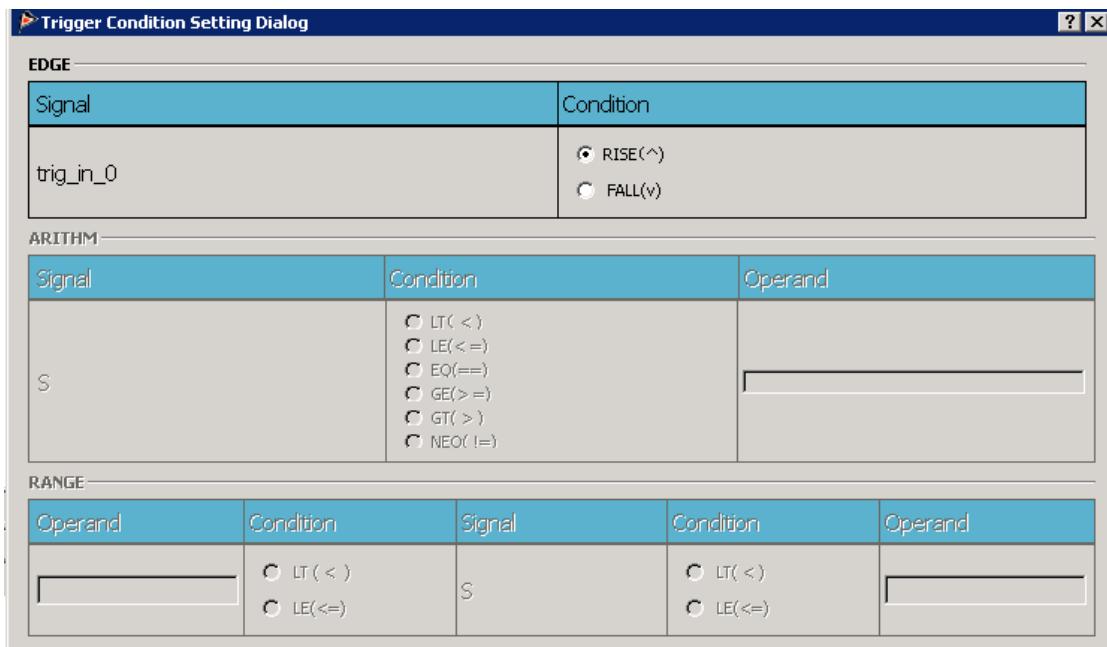
打开 debugware 界面，



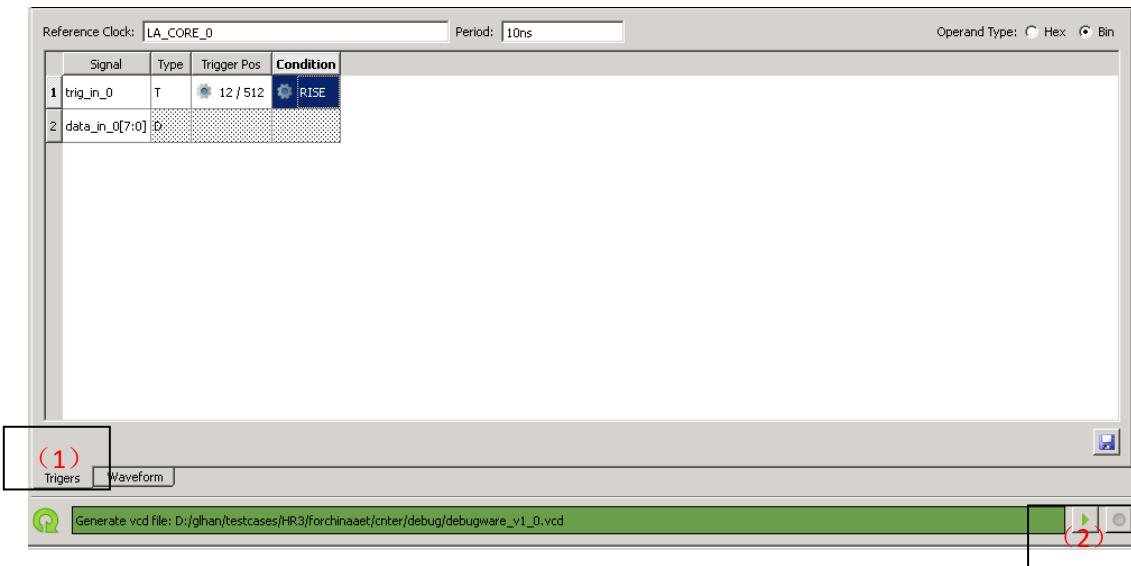
可以单击 (1) 处修改 trigger offset;



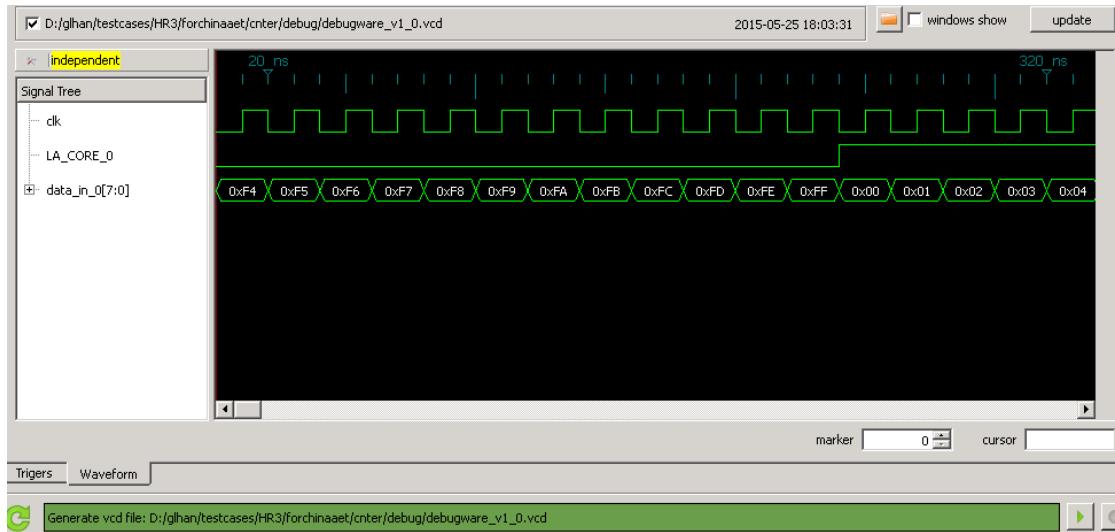
也可以单击 (2) 处弹出下图所示的触发条件设置框图。



设置完毕，回到（1）处 trigger 标签页，点击（2）处 Trigger 按钮



然后页面跳转到 Waveform 标签页，可以观察数据变化。



当然这只是一个用法上的简单示例，例子本身经不起详细考究，如要满足采样定理之类。

我今天在写这个的时候有一个想法，其实对于 EDA 来讲，UI 做的好坏会对易用性有影响，但是只要流程能够打通，UI 的不完美从来不是阻止有创意的开发人员做出好的设计的障碍。或者说，即使 UI 做的再完美，倘若不懂的，还是不会使用。若是想要弄懂，外文可以翻译，百度可以查找，他人可以请教……总之方法总比困难多。愿，您是一个积极乐观的国产 FPGA 的粉丝，愿您在国产 FPGA 上做出精彩设计；愿，国产 FPGA 得到您青睐的同时，更多的人也因您的推荐而认识 FPGA。