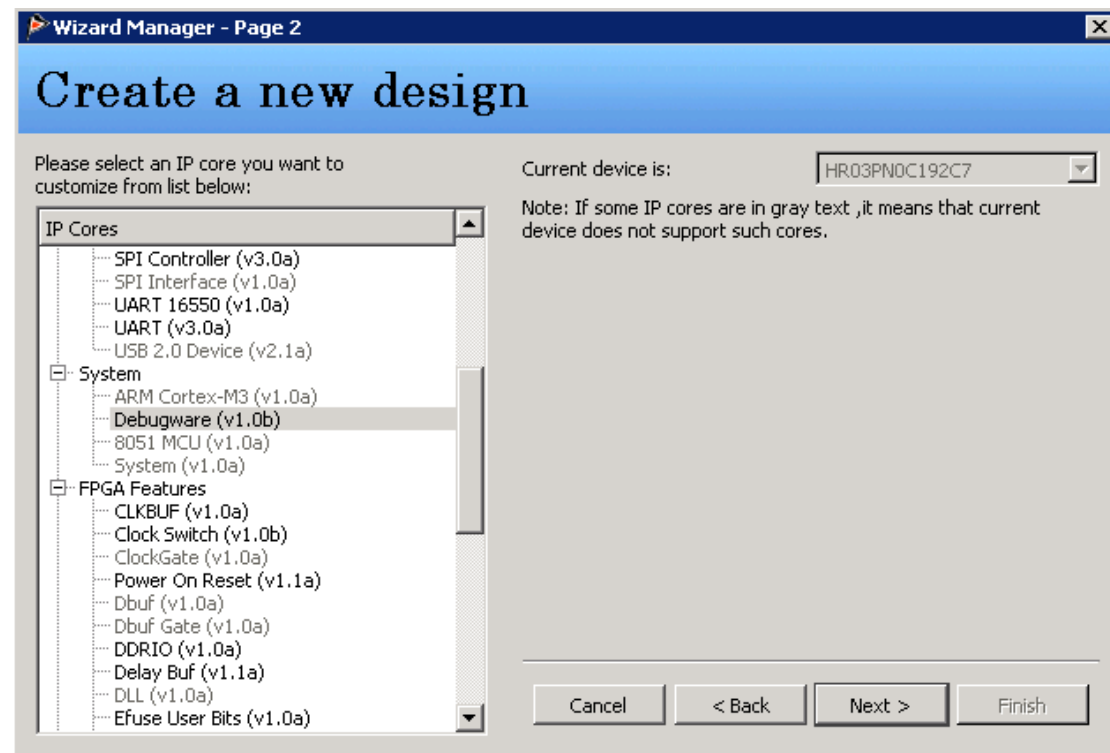


Debugware 是一个嵌入式逻辑分析仪,设计者有时候会希望探测 FPGA 内部的信号传输情况, Debugware 使用 EMB(FPGA 内部的 memory)存储这些信号波形,当触发事件发生后,通过 JTAG 从这些 EMB 中读取波形数据。

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。

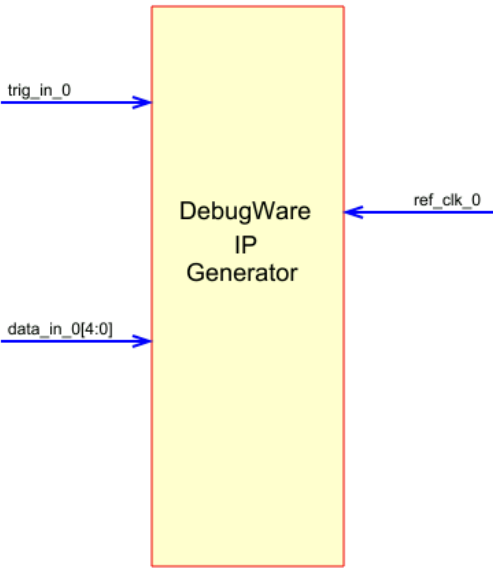
现在,我们先用 wizard 例化一个 debugware.



DebugWare Wizard - Page 1

Input Module Name

Help



Module name:

Save output file:

- ☒ Current project's source directory
- ☐ Other directory:

Currently selected language:

Currently selected device name:

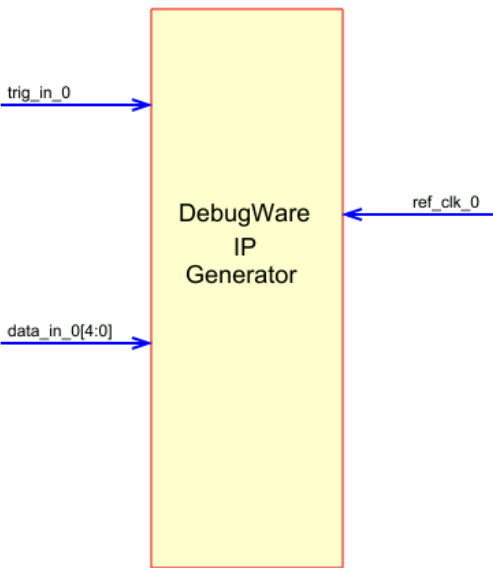
☒ Match project/default

Cancel < Back Next > Finish

DebugWare Wizard - Page 2

General Settings

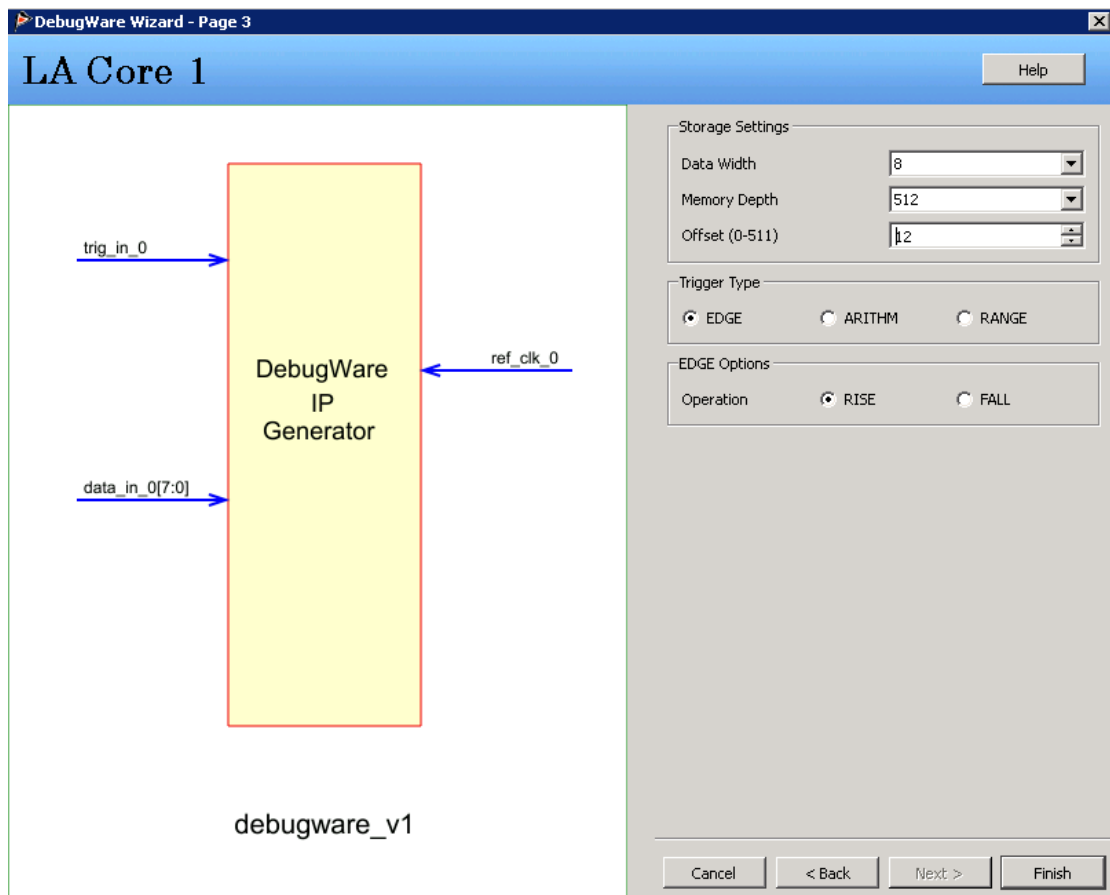
Help



LA Core Settings

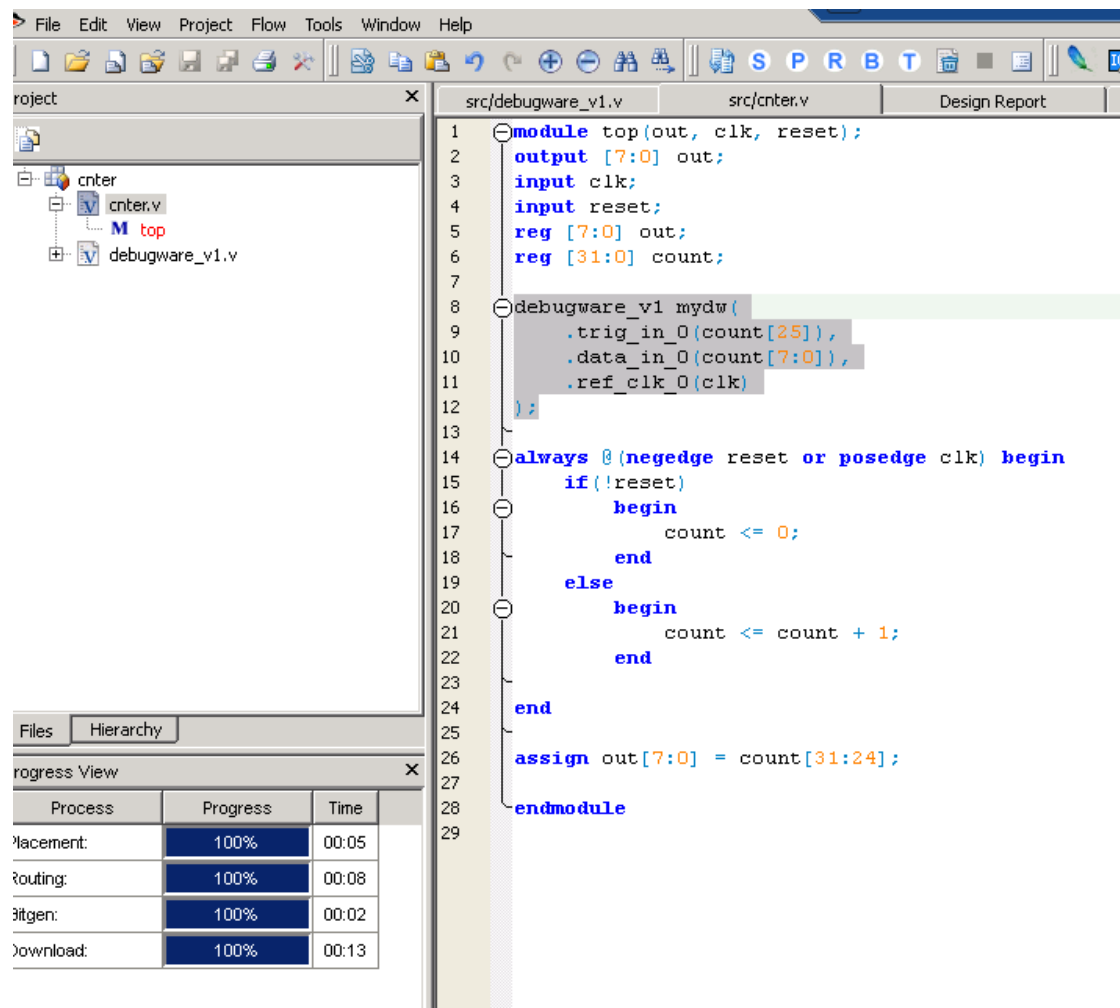
Number Of LA Cores

Cancel < Back Next > Finish



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,



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。

src/debugware_v1.v | src/cnter.v | Design Report | top.aoc | download | Real Time Debugger

Package Pins

Pin Name	Location	Port Name
+ I/O Bank: 0 (45 pins)		
+ I/O Bank: 1 (33 pins)		
+ I/O Bank: 2 (7 pins)		
+ I/O Bank: 3 (49 pins)		
+ I/O Bank: 4 (48 pins)		

Top View
HR03PN0C192C7

I/O Ports

Named: Filters: Port: all

Port Name	Location	Pin Name	Direction	Operation Mode	Fast Input	Fast Output	Fast OE	Bank	IO Standard	Drive Strength
clk	T8	IO14_CLK_1	input	clock	false			1	2.5V (default)	
out										
out[0]	B12	IO09_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[1]	B11	IO11_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[2]	B10	IO13_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[3]	B9	IO16_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[4]	C8	IO17_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[5]	A7	IO21_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[6]	A6	IO23_4	output	normal		false	false	4	2.5V (default)	8mA (default)
out[7]	C6	IO25_4	output	normal		false	false	4	2.5V (default)	8mA (default)
reset	M16	IO09_3	input	normal	false			3	2.5V (default)	

这样，跑完 Flow， 需要点击



， 出现下载页面

src/debugware_v1.v | src/cnter.v | Design Report | top.aoc | download | Real Time Debugger

Specify the target:

☒ HR3 ☐ VIRTUAL-JTAG

Setting
Detect Devices

Directory: D:\gihan\testcases\HR3\forchinaaet\cnter\outputs

Browse Refresh

File Name	Size(Bytes)	Last Modify
D:\gihan\testcases\HR3\forchinaaet\cnter\outputs\top_bin.acf	105519	2015-05-25 17:29:15

Options

Target: Flash | ☒ Erase | ☒ Verify | Address: 0x 0

E-Fuse Burner | OTP Programmer | SPI Programmer

Erase Flash | Program

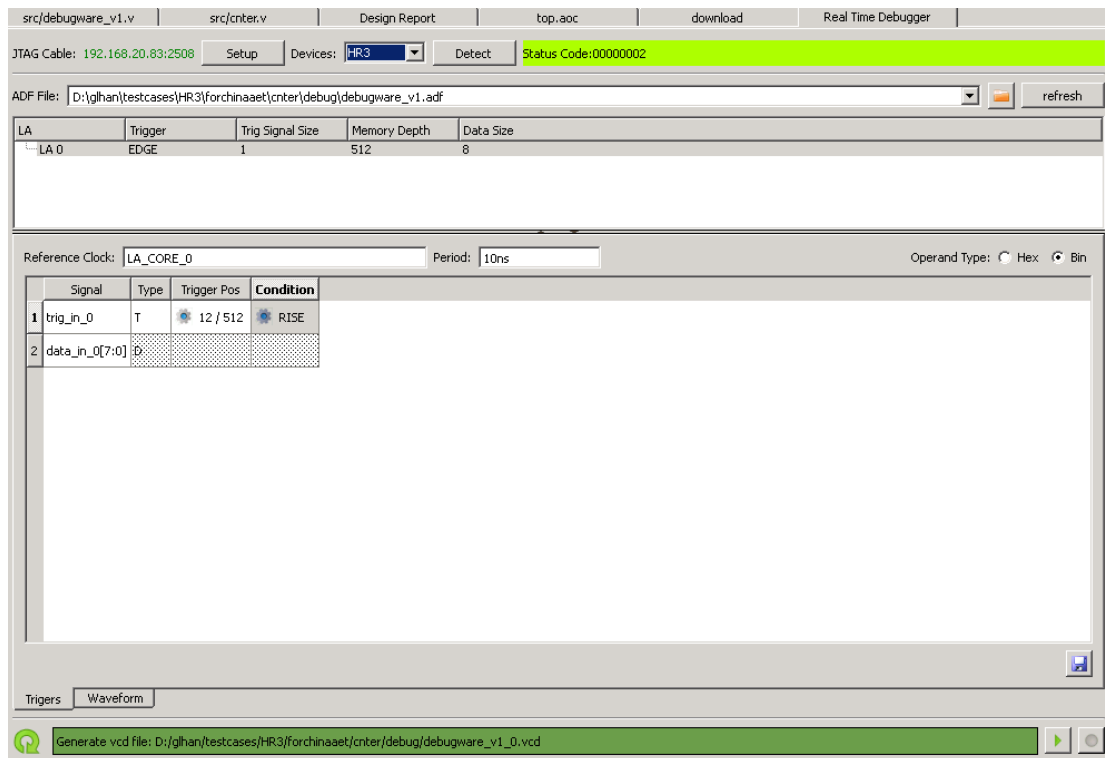
Cable: 192.168.20.83:2508 Status Code: 00000002

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。

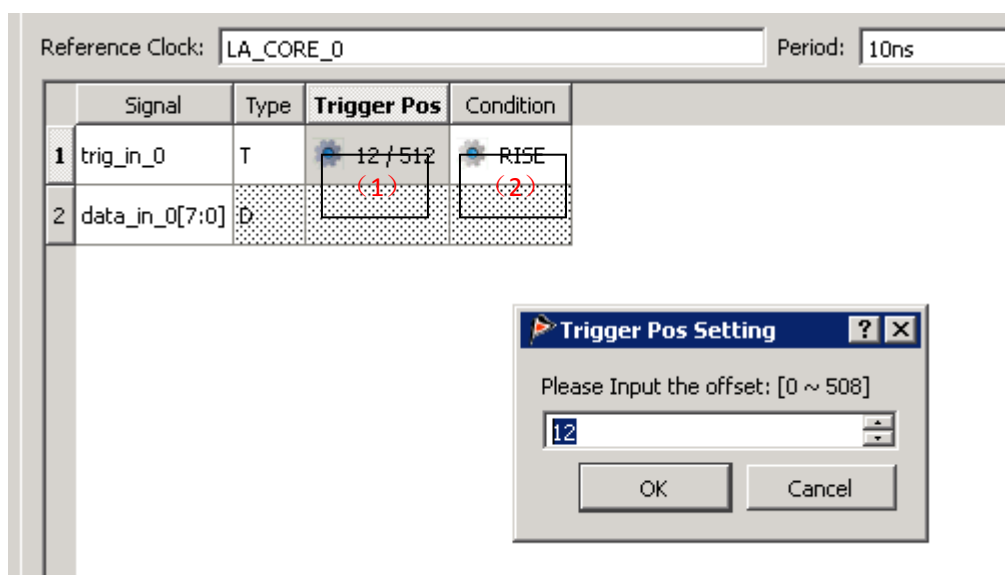
然后我们点击



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。

Trigger Condition Setting Dialog

EDGE

Signal	Condition
trig_in_0	<input checked="" type="radio"/> RISE(^) <input type="radio"/> FALL(v)

ARITHM

Signal	Condition	Operand
S	<input type="radio"/> LT(<) <input type="radio"/> LE(<=) <input type="radio"/> EQ(==) <input type="radio"/> GE(>=) <input type="radio"/> GT(>) <input type="radio"/> NEO(!=)	

RANGE

Operand	Condition	Signal	Condition	Operand
	<input type="radio"/> LT(<) <input type="radio"/> LE(<=)	S	<input type="radio"/> LT(<) <input type="radio"/> LE(<=)	

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

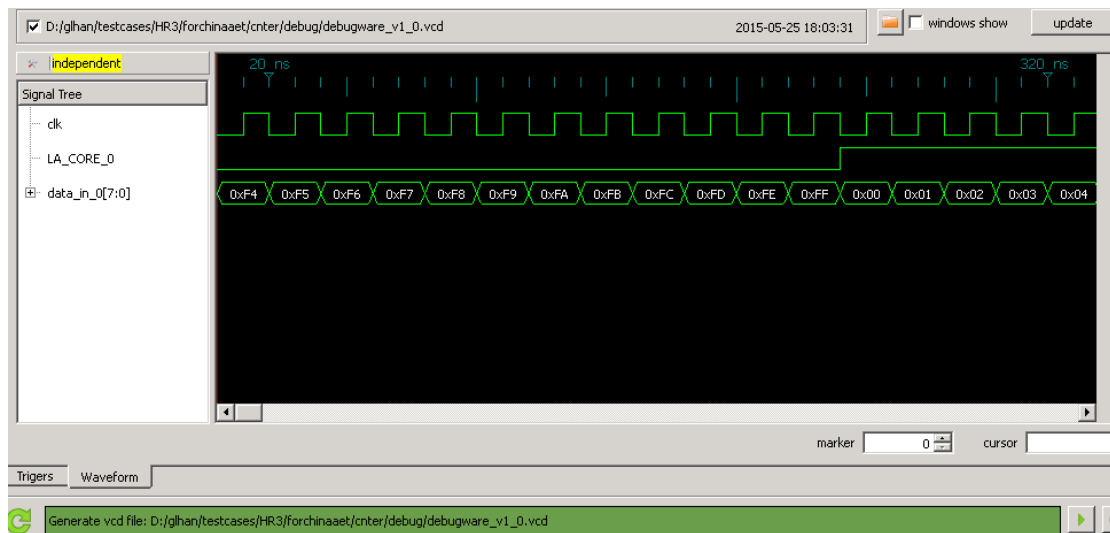
Reference Clock: LA_CORE_0 Period: 10ns Operand Type: ☐ Hex ☒ Bin

	Signal	Type	Trigger Pos	Condition
1	trig_in_0	T	12 / 512	RISE
2	data_in_0[7:0]	D		

(1) Triggers Waveform

Generate vcd file: D:/ghay/testcases/HR3/forchinaaet/cnter/debug/debugware_v1_0.vcd (2)

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。