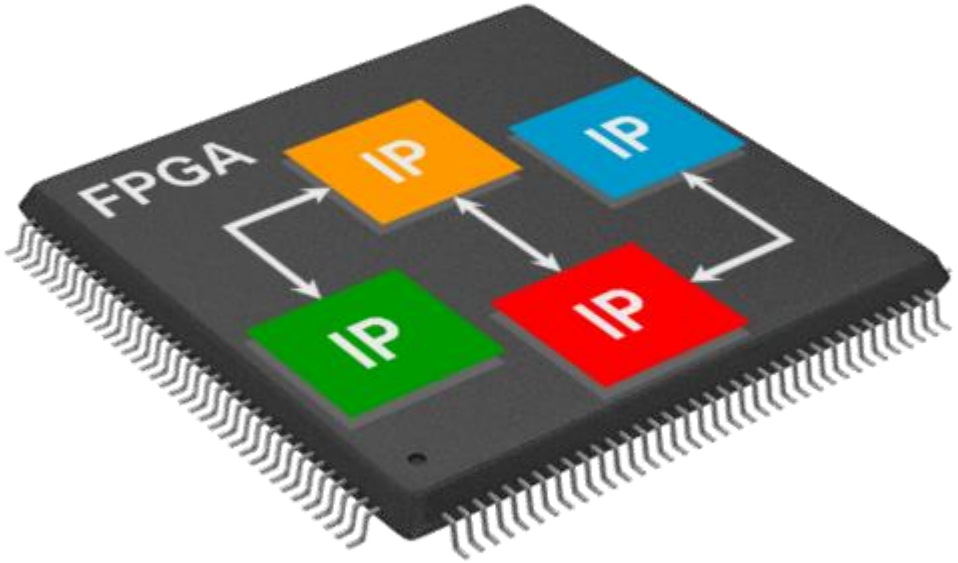
实验7：利用IP核设计



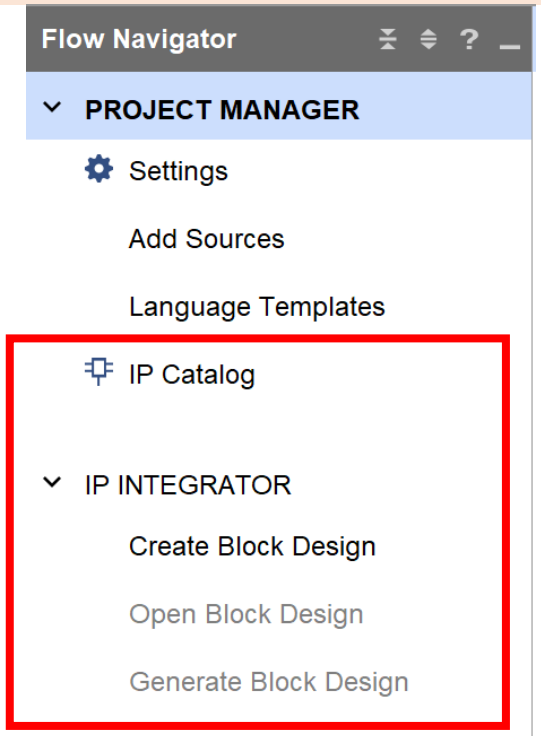
xgsun@fudan.edu.cn

孙晓光

2022-10-31

Intelligent Property Core

IP核：快速开发的法宝

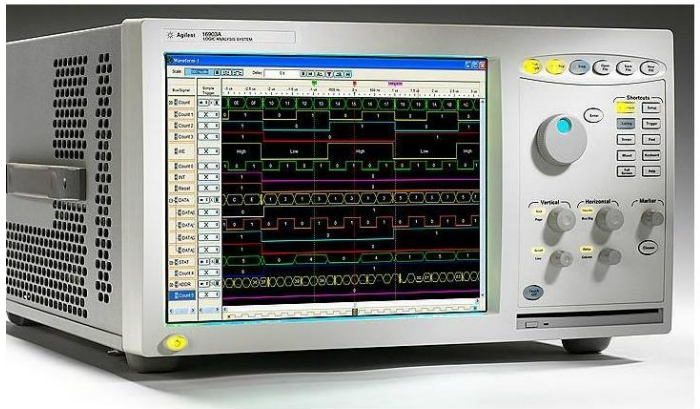


- 具有知识产权的集成电路芯核
- 反复验证过的、具有特定功能的宏模块
- **软核**(HDL语言)、固核(网表形式)、硬核(版图形式)

IP目录

- Vivado Repository
  - Alliance Partners
  - Audio Connectivity & Processing
  - Automotive & Industrial
  - AXI Infrastructure
  - AXIS Infrastructure
  - BaseIP
  - Basic Elements
  - Communication & Networking
  - Debug & Verification
  - Digital Signal Processing
  - Embedded Processing
  - FPGA Features and Design
  - Kernels
  - Math Functions
  - Memories & Storage Elements
  - Network on Chip (NoC)
  - Partial Reconfiguration
  - SDAccel DSA Infrastructure
  - Standard Bus Interfaces
  - Test NOC
  - Video & Image Processing
  - Video Connectivity

- **数学运算**： 整数运算、浮点运算， **加减法**、**乘除法**、比较器、移位器…
- **输入输出**： 时钟控制器、锁相环(**PLL**)、DDR…
- **设计调试**： 逻辑分析仪(**ILA**)
- **存储器类**： **ROM**、**RAM**、FIFO、Flash…
- **图像处理**： 视频监视器、视频IO、图像裁减器、滤波器、混合器、矫正器…
- **处理器**： MicroBlaze(嵌入在FPGA中的RISC处理器软核) 32位、64位
- **数字信号处理**：
- **通信+网络**：

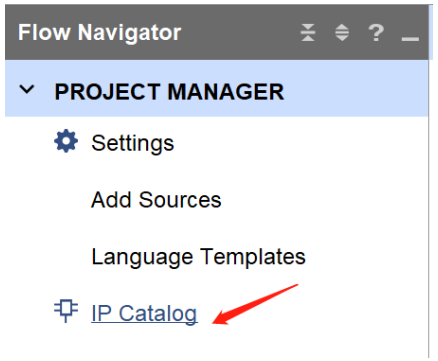


1

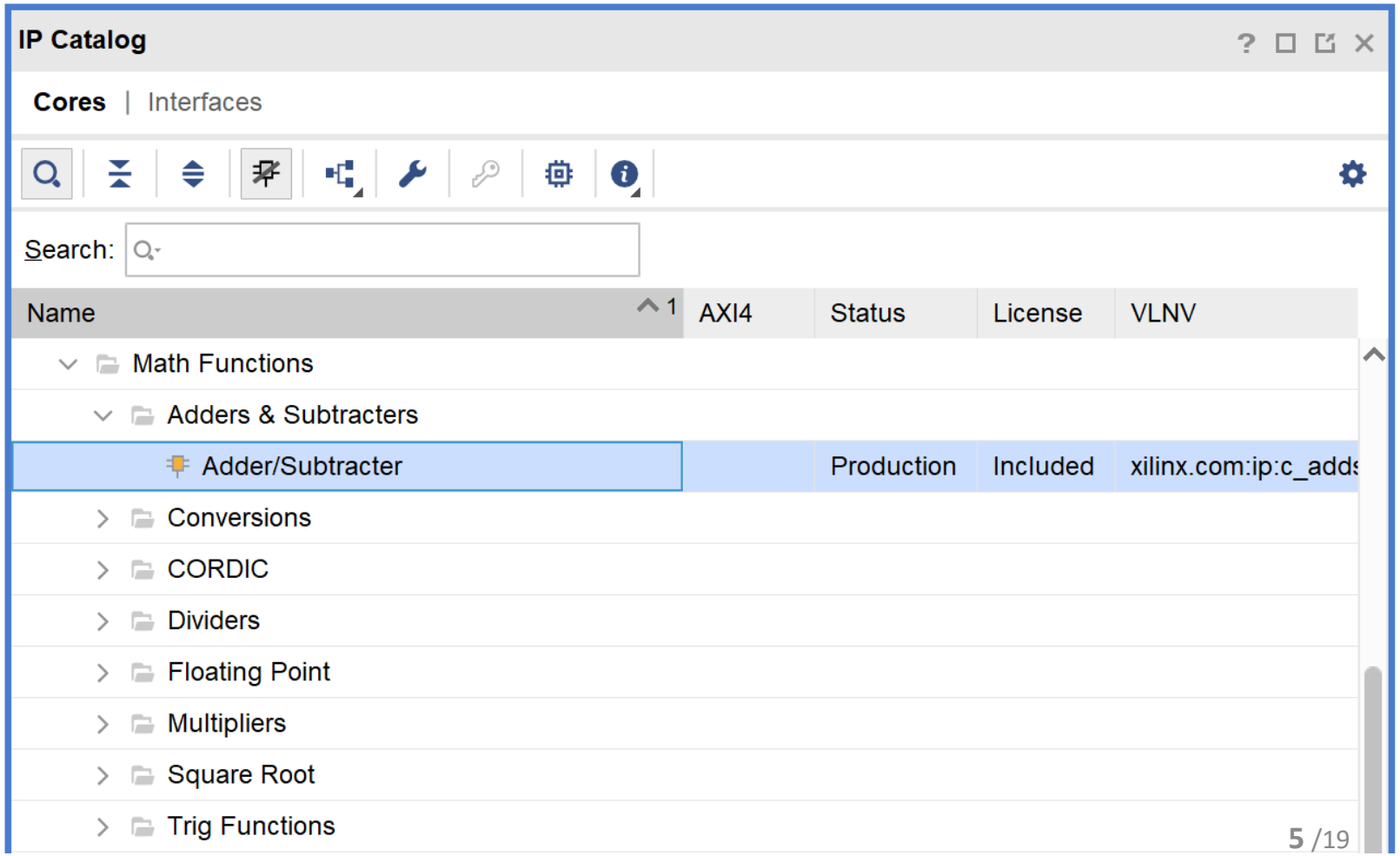
Adder

Add_IP

利用IP核实现“加法器”


Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog


双击


IP Catalog
Cores | Interfaces
Search:
Name
AXI4
Status
License
VLNV
Math Functions
Adders & Subtracters
Adder/Subtracter
Production
Included
xilinx.com:ip:c_adds
Conversions
CORDIC
Dividers
Floating Point
Multipliers
Square Root
Trig Functions

配置 IP 参数

Customize IP

Adder/Subtracter (12.0)

Documentation | IP Location | Switch to Defaults

IP Symbol | Information

Show disabled ports

A[7:0]
B[7:0] S[8:0]
CLK

Component Name: add8

Basic | Control

	S =	A	+/-	B
Implement using	Fabric			
Input Type		Signed		Signed
Input Width		8 [2,256]		8 [2,256]
Add Mode		Add		
Output Width		9 [8 - 9]		
Latency Configuration		Manual		
Latency		1 [0 - 9]		

Constant Input

Constant Value (Bin) 00000000

OK | Cancel

Component Name: add8

Basic | Control

Clock Enable (CE)

Carry In (C_IN)

Create Directory

OK to create the directory 'c:/_SamData/Vivado2022/Codes/Add_IP/Add_IP.srcs/sources_1/ip'?

OK | Cancel

Generate Output Products

The following output products will be generated.

Preview

add8.xci (OOC per IP)
- Instantiation Template
- Synthesized Checkpoint (.dcp)
- Structural Simulation
- Change Log

Synthesis Options

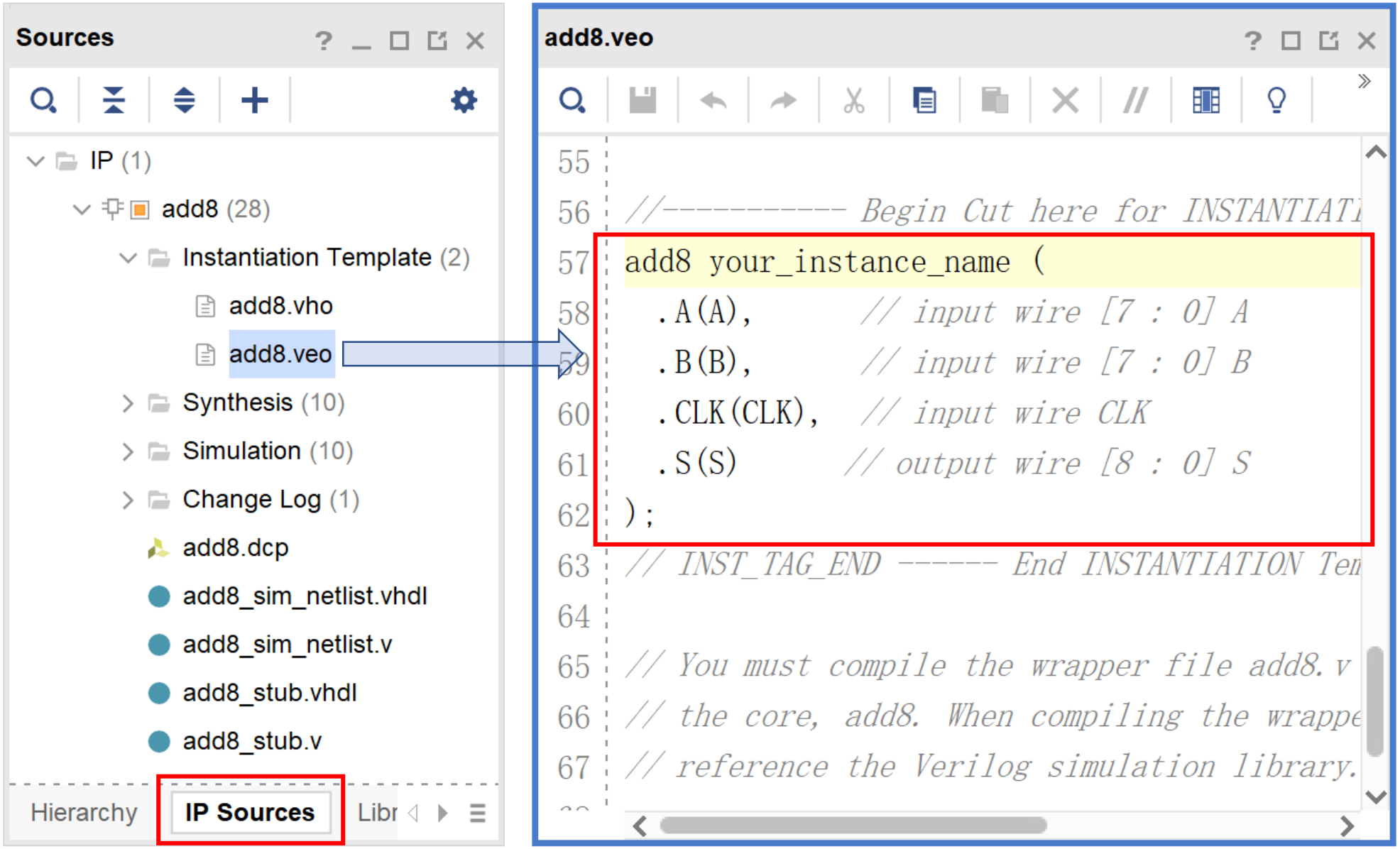
Global
Out of context per IP

Run Settings

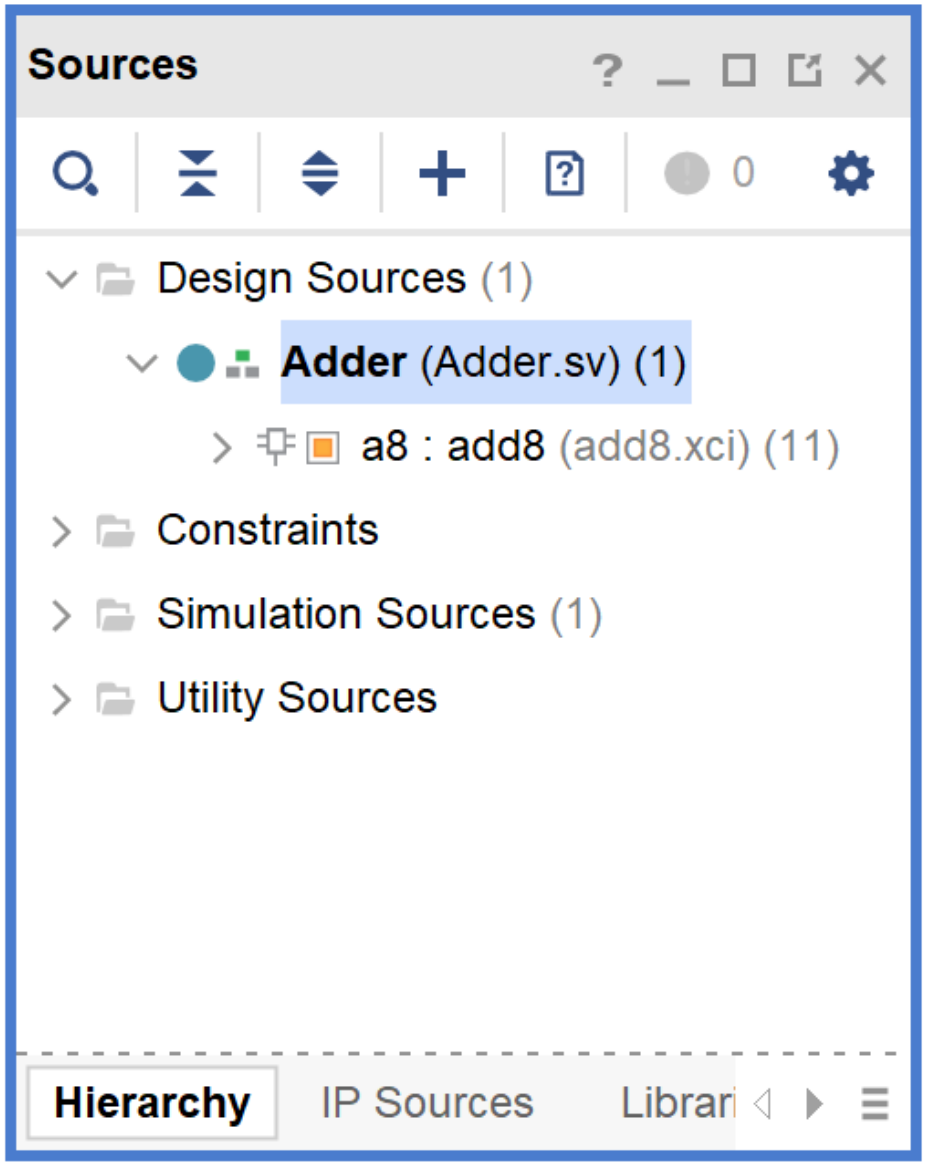
Number of jobs: 4

Apply | Generate | Skip

实例化 模板


Sources
IP (1)
add8 (28)
Instantiation Template (2)
add8.vho
add8.veo
Synthesis (10)
Simulation (10)
Change Log (1)
add8.dcp
add8_sim_netlist.vhdl
add8_sim_netlist.v
add8_stub.vhdl
add8_stub.v
Hierarchy
IP Sources
Libr
add8.veo
//---------- Begin Cut here for INSTANTIATI
add8 your_instance_name (
.A(A), // input wire [7 : 0] A
.B(B), // input wire [7 : 0] B
.CLK(CLK), // input wire CLK
.S(S) // output wire [8 : 0] S
);
// INST_TAG_END ------ End INSTANTIATION Tem
// You must compile the wrapper file add8.v
// the core, add8. When compiling the wrappe
// reference the Verilog simulation library.

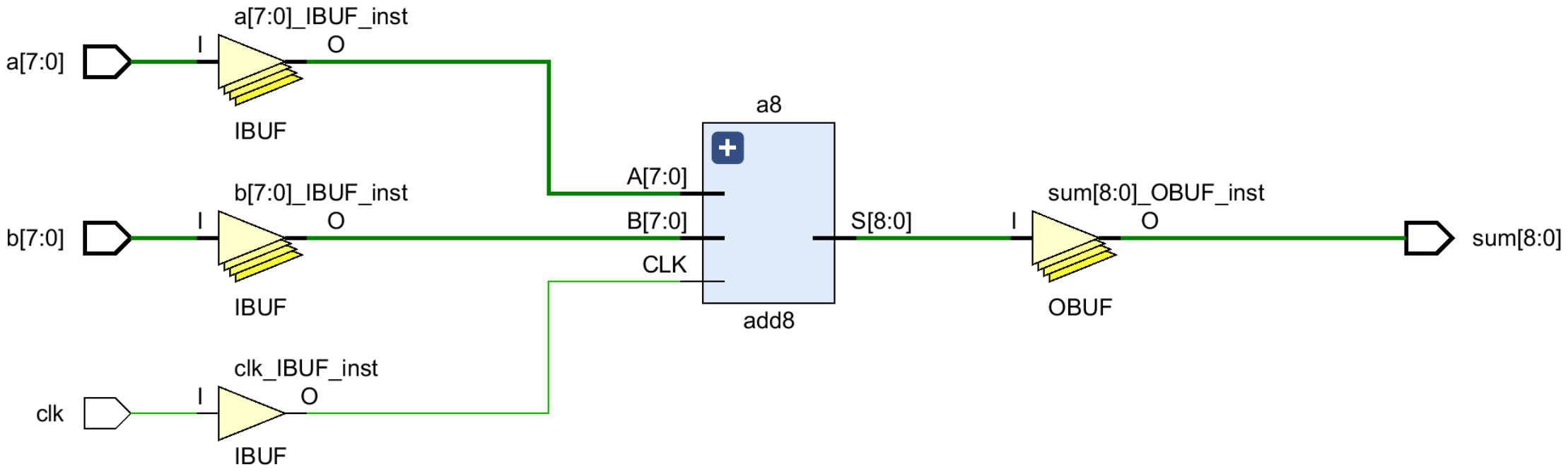
实例化 IP核



Adder.sv

```
1  module Adder(
2      input  logic [7:0] a, b,
3      input  logic       clk,
4      output logic [8:0] sum );
5
6      add8 a8( .A(a),   .B(b),
7               .CLK(clk),
8               .S(sum)   );
9  endmodule
```

原理图


a[7:0]
a[7:0]_IBUF_inst
I
O
IBUF
b[7:0]
b[7:0]_IBUF_inst
I
O
IBUF
clk
clk_IBUF_inst
I
O
IBUF
a8
A[7:0]
B[7:0]
CLK
add8
S[8:0]
sum[8:0]_OBUF_inst
I
O
OBUF
sum[8:0]

2

逻辑分析仪
ILA

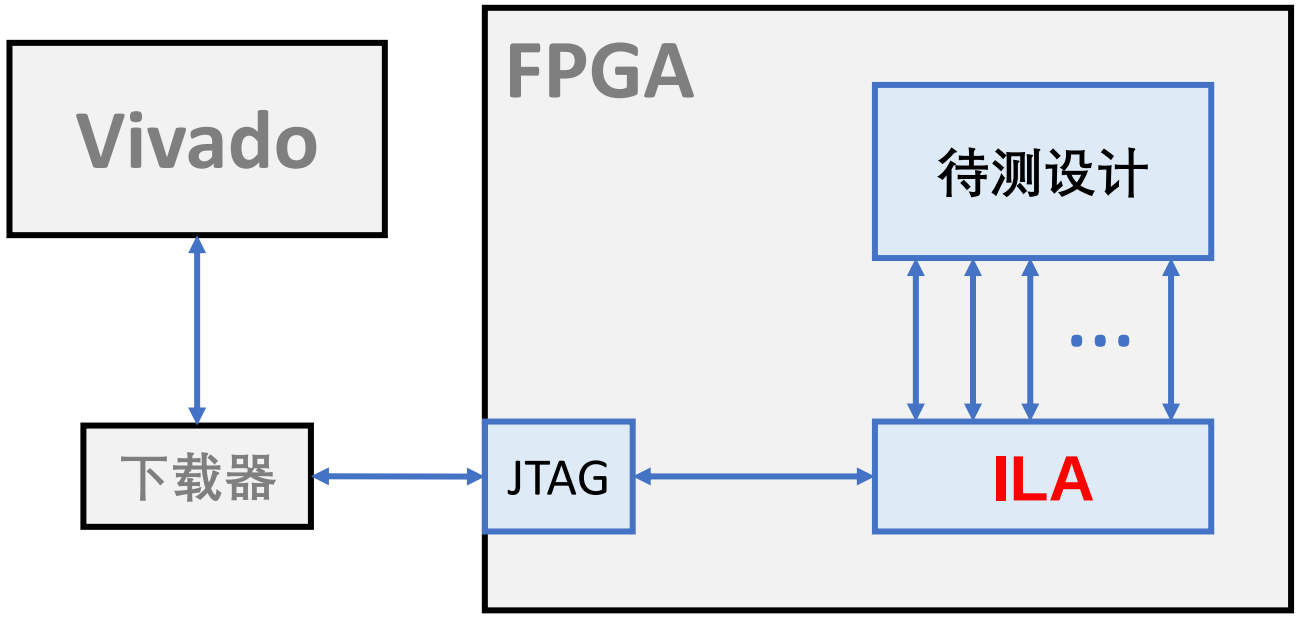
ILA

ILA 可定制集成逻辑分析器

ILA IP核是一款逻辑分析器内核，可用于监控设计中的内部信号。

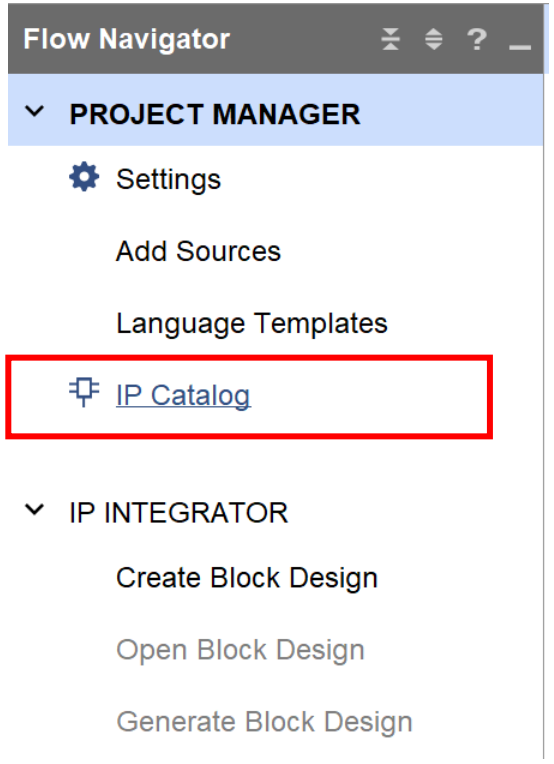
上板子后如何实时查错？

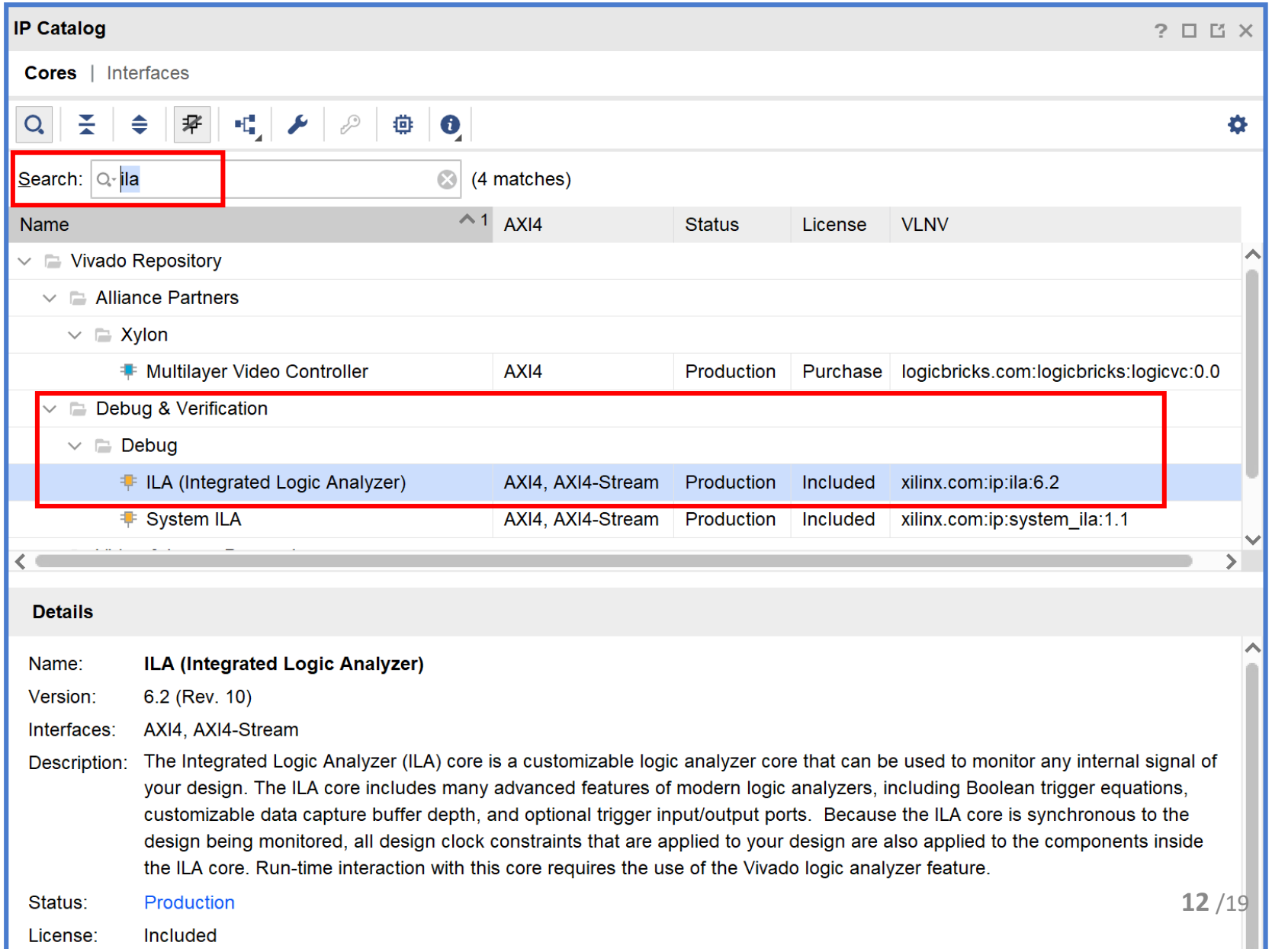
```
1  module Gray(
2      input  logic [3:0] Bin,
3      output logic [3:0] Gray );
4
5      assign Gray[3] = Bin[3];
6      assign Gray[2] = Bin[3] ^ Bin[2];
7      assign Gray[1] = Bin[2] ^ Bin[1];
8  //  assign Gray[0] = Bin[1] ^ Bin[0];
9
10     // 错误写法:
11     assign Gray[0] = Bin[1] * Bin[0]; // *
12 endmodule
```



ILA 会将所采集到的探针数据存放在 RAM 中，然后，通过 JTAG 和下载器上传到 Vivado。

查找 ILA


Flow Navigator
PROJECT MANAGER
Settings
Add Sources
Language Templates
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design


IP Catalog
Cores | Interfaces
Search: ila
(4 matches)
Name
AXI4
Status
License
VLNV
Vivado Repository
Alliance Partners
Xylon
Multilayer Video Controller
AXI4
Production
Purchase
logicbricks.com:logicbricks:logicvc:0.0
Debug & Verification
Debug
ILA (Integrated Logic Analyzer)
AXI4, AXI4-Stream
Production
Included
xilinx.com:ip:ila:6.2
System ILA
AXI4, AXI4-Stream
Production
Included
xilinx.com:ip:system_ila:1.1
Details
Name: ILA (Integrated Logic Analyzer)
Version: 6.2 (Rev. 10)
Interfaces: AXI4, AXI4-Stream
Description: The Integrated Logic Analyzer (ILA) core is a customizable logic analyzer core that can be used to monitor any internal signal of your design. The ILA core includes many advanced features of modern logic analyzers, including Boolean trigger equations, customizable data capture buffer depth, and optional trigger input/output ports. Because the ILA core is synchronous to the design being monitored, all design clock constraints that are applied to your design are also applied to the components inside the ILA core. Run-time interaction with this core requires the use of the Vivado logic analyzer feature.
Status: Production
License: Included

配置 ILA

Customize IP

ILA (Integrated Logic Analyzer) (6.2)

Documentation | IP Location | Switch to Defaults

Show disabled ports

Component Name: ila_G

To configure more than 64 probe ports use Vivado Tcl Console

General Options | Probe_Ports(0..1)

Monitor Type

Native / AXI

探针数量

Number of Probes: 2 [1...1024]

Sample Data Depth: 1024

Same Number of Comparators for All Probe Ports

Number of Comparators: 1

Trigger Out Port

Trigger In Port

Input Pipe Stages: 0

Trigger And Storage Settings

Capture Control

Advanced Trigger

GUI configuration mode is limited to 64 probe ports.

OK | Cancel

clk
probe0[0:0]
probe1[0:0]

Customize IP

ILA (Integrated Logic Analyzer) (6.2)

Documentation | IP Location | Switch to Defaults

Show disabled ports

Component Name: ila_G

To configure more than 64 probe ports use Vivado Tcl Console

General Options | **Probe_Ports(0..1)**

Probe Port	Probe Width [1..4096]	Number of Comparators	Probe Trigger or Data
PROBE0	4	1	DATA AND TRIGGER
PROBE1	4	1	DATA AND TRIGGER

探针宽度

clk
probe0[3:0]
probe1[0:0]

```
module Gray(
    input  logic [3:0] Bin,
    output logic [3:0] Gray );
```

实例化 ILA，生成bit，下载

Sources

- Design Sources (1)
  - Gray_Top (Gray_Top.sv) (3)
    - G1 : Gray (Gray.sv)
    - iG1 : ila_G (ila_G.xci) (1)
      - ila_G (ila_G.v) (1)
        - inst : ila_v6_2_10_ila
    - x7 : x7seg (x7seg.sv) (1)
- Constraints (1)
- Simulation Sources (1)
- Utility Sources

ila_G.v

```
80  C_PROBE806_TYPE=1, C_PROBE807_TYPE=1, C_PROBE808_TYPE=1, C_PROBE809_TYP
81  C_PROBE906_TYPE=1, C_PROBE907_TYPE=1, C_PROBE908_TYPE=1, C_PROBE909_TYP
82  C_PROBE1006_TYPE=1, C_PROBE1007_TYPE=1, C_PROBE1008_TYPE=1, C_PROBE1009
83
84  module ila_G (
85  clk,
86
87
88  probe0,
89  probe1
90  );
91
92  input clk;
93
```

实例化模板

Gray_Top.sv

```
1   module Gray_Top(
2       input  logic         CLK100MHZ,
3       input  logic [15:12] SW,
4       output logic [15:12] LED,
5       output logic [6:0]   A2G,
6       output logic [7:0]   AN     );
7
8       assign LED = SW;
9
10      logic [3:0] GrayData;
11      Gray G1(SW, GrayData);
12
13      ila_G iG1(.clk    (CLK100MHZ),
14                .probe0(SW        ),
15                .probe1(GrayData ) );
16
17      x7seg x7(CLK100MHZ,
18               {SW, GrayData},
19               A2G,
20               AN  );
21  endmodule
```

实例化

- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: ado2022/Codes/ILA/ILA.runs/impl_1/Gray_Top.bit

Debug probes file: ado2022/Codes/ILA/ILA.runs/impl_1/Gray_Top.ltx

Enable end of startup check

有调试探针信息的.ltx 文件

Program　Cancel

ILA 的调试窗口

自动触发开关

开始触发

立即触发

停止触发

dashboard_1

Waveform - hw_ila_1

Dashboard Options

ILA Status: Idle

Name	Value
GrayData_1[3:3]	
LED_OBUF[14:12]	
[14]	
[13]	
[12]	
GrayData[3:0]	
[3]	
[2]	
[1]	
[0]	

0 200 400 600 800 1,000 1,200 1,400 1,600 1,800

```
13        ila_G iG1(.clk     (CLK100MHZ),
14                  .probe0(SW[15:12]),
15                  .probe1(GrayData ) );
```

Settings - hw_ila_1 | Status - hw_ila_1

Core status Idle

Capture status - Window 1 of 1

Window sample 0 of 1024

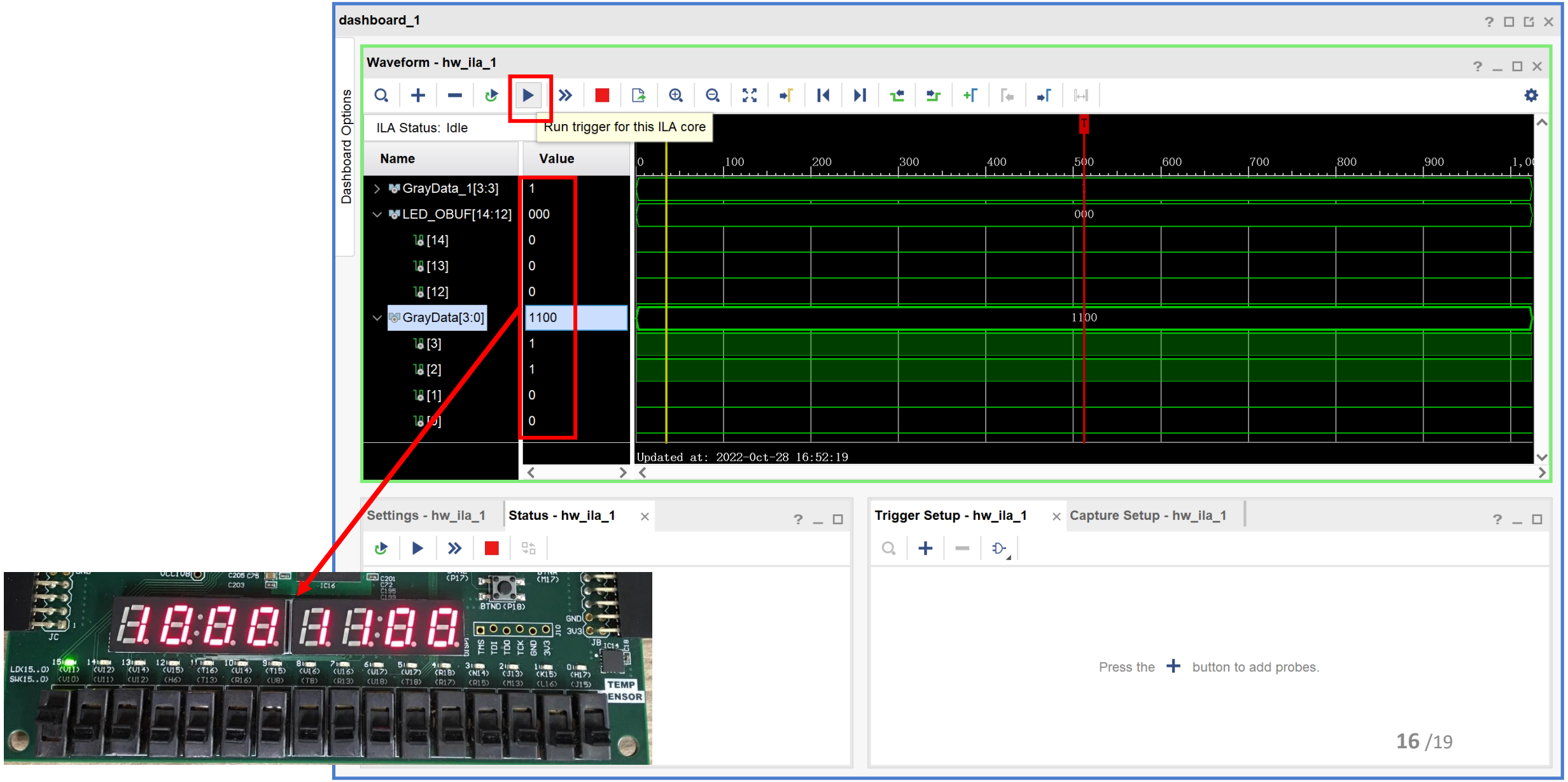
Idle

Trigger Setup - hw_ila_1 | Capture Setup - hw_ila_1

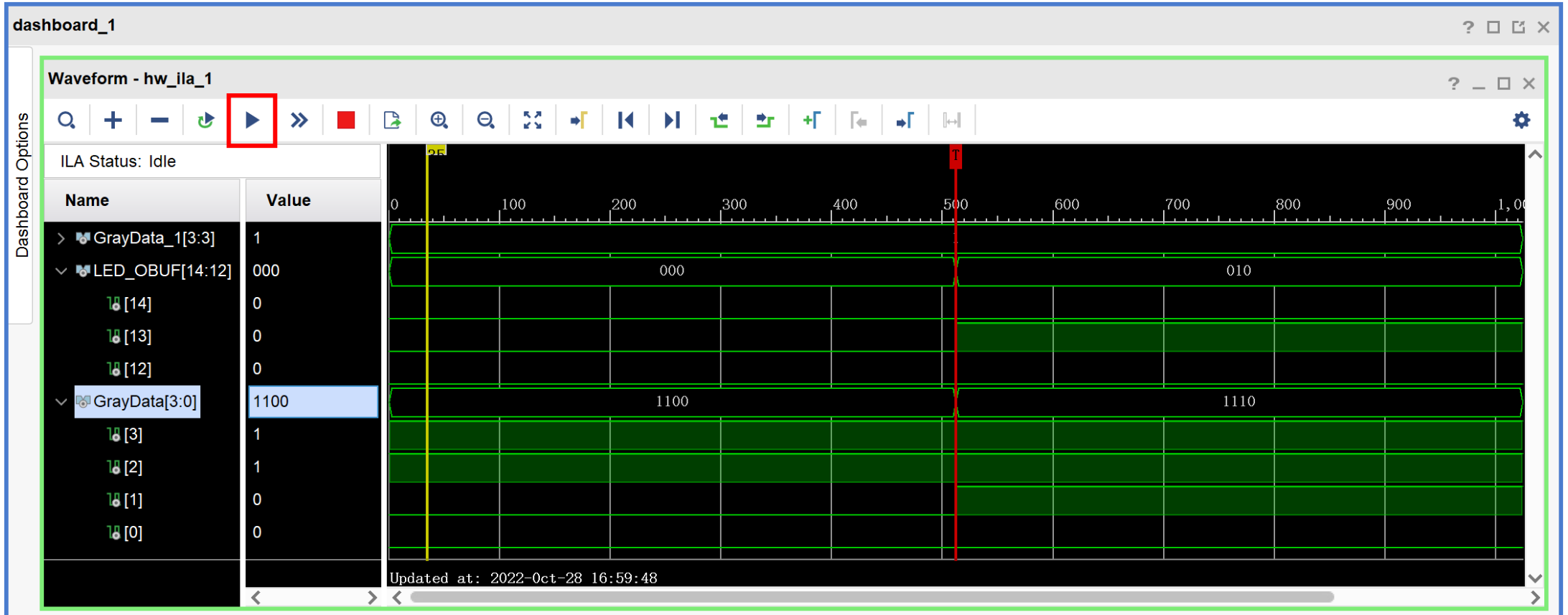
Press the + button to add probes

触发：决定 ILA 在什么时候将RAM中的探针值数据上传到 Vivado。当 ILA 检测到触发条件满足时，就会把 探针值数据上载到 Vivado，然后 Vivado 将探针数据的波形显示出来。

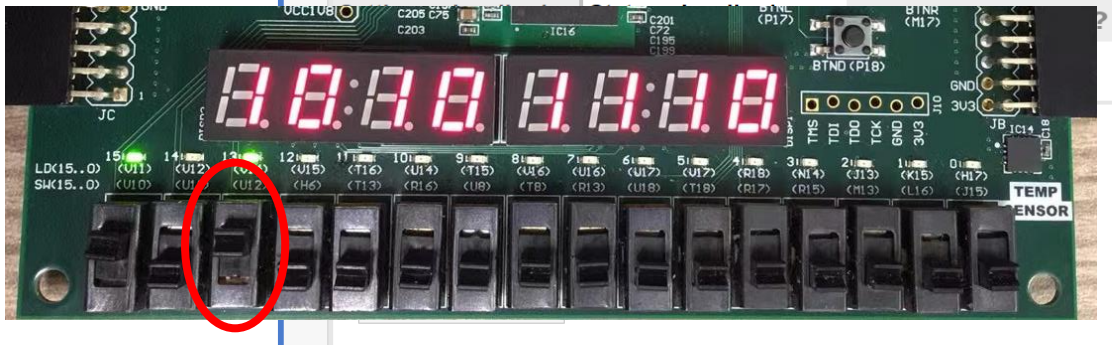
无触发条件时运行：板子现状

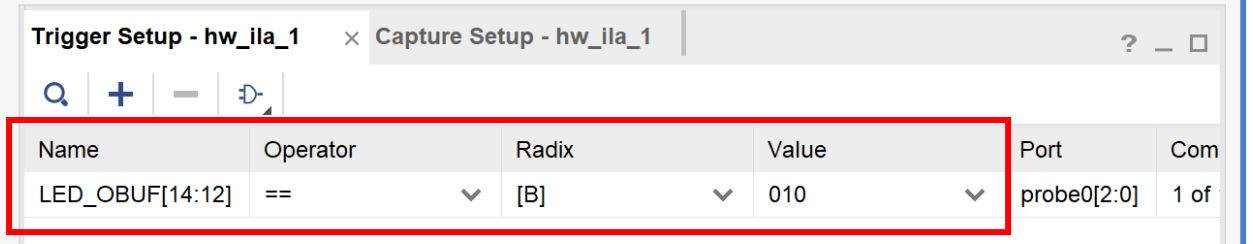


设置触发条件1： ==010

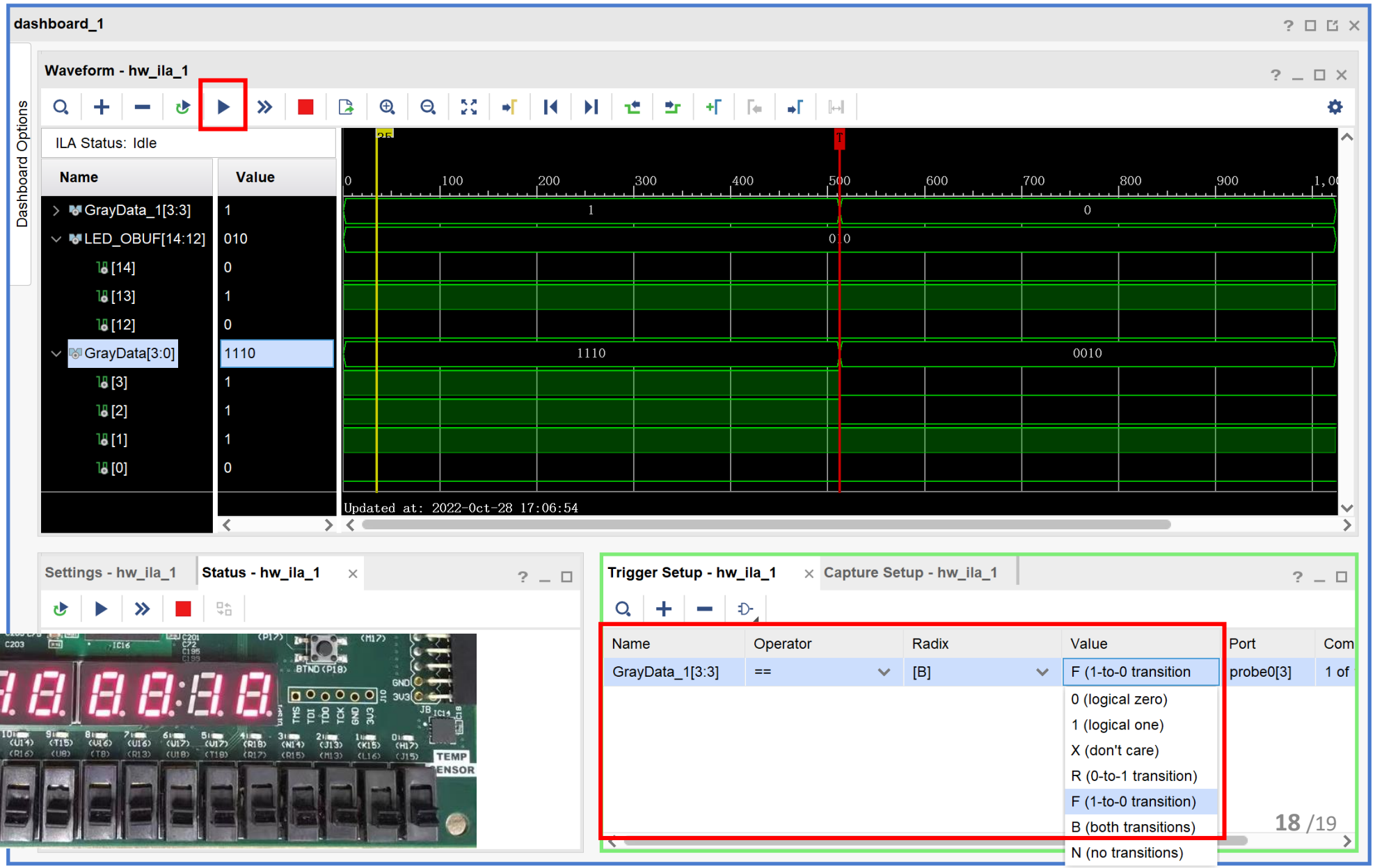


拨动后触发





设置触发条件1： 1-to-0



拨动后触发

收尾工作

Gray_Top.sv

```
1  module Gray_Top(
2      input  logic         CLK100MHZ,
3      input  logic [15:12] SW,
4      output logic [15:12] LED,
5      output logic [6:0]   A2G,
6      output logic [7:0]   AN    );
7
8      assign LED = SW;
9
10     logic [3:0] GrayData;
11     Gray G1(SW, GrayData);
12
13     ila_G iG1(.clk    (CLK100MHZ),
14               .probe0(SW        ),
15               .probe1(GrayData ) );
16
17     x7seg x7(CLK100MHZ,
18              {SW, GrayData},
19              A2G,
20              AN  );
21 endmodule
```

- 调试正确后，删除ila_G模块，及实例化代码，
- 再重新生成bit文件。