ESRG

Embedded Systems Research Group

CENTROALGORITMI

Xilinx Vivado Logic Analyzer

Group Stage

Universidade do Minho
Escola de Engenharia

POPH
PROGRAMA OPERACIONAL POTENCIAL HUMANO

QREN
QUADRO DE REFERÊNCIA ESTRATÉGICO NACIONAL
PORTUGAL 2007.2013

UNIÃO EUROPEIA
Fundo Social Europeu

Introduction

- This document contains a tutorial designed to help debug complex FPGA designs
- This tutorial describes the steps involved in taking a small RTL design and the multiple ways of inserting the **Integrated Logic Analyzer** (ILA) core to help debug the design
- This debug method:
  - Allows to visualize the FPGA's internal signals;
  - Debugging in runtime that allows:
    - To validate system's behavior
    - To detect bugs that aren't spotted in simulation
    - To accelerate the debugging process

Introduction

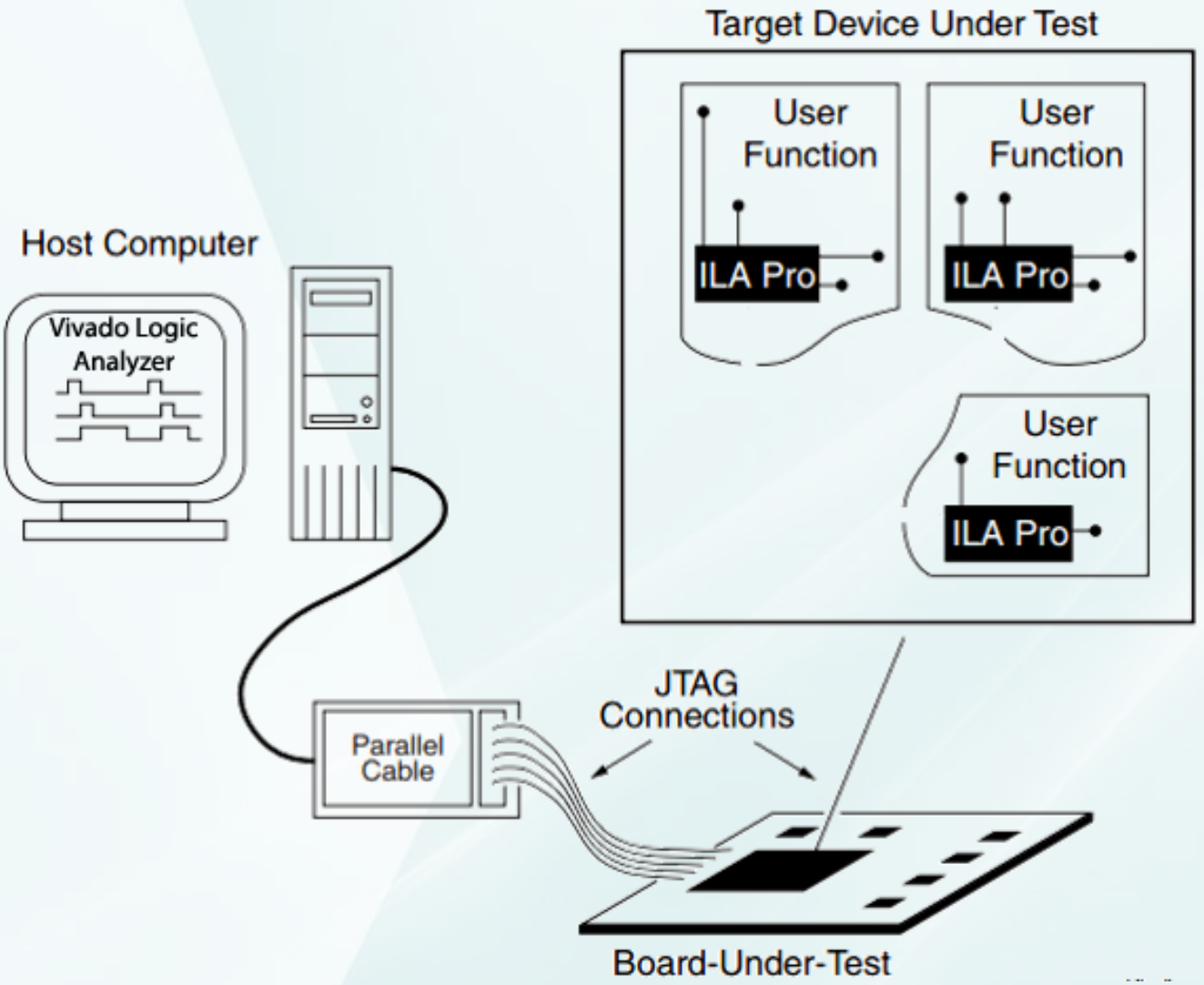
- Integrated Logic Analyzer (ILA):
  - Can be used to **monitor the internal signals of a design**.
  - Includes many advanced features of modern logic analyzers, including **Boolean trigger equations**, and **edge transition triggers**.
  - **All design clock constraints** that are applied to your design **are also applied** to the components inside the **ILA** core, since the ILA core is synchronous to the design being monitored,

Introduction

- Vivado Logic Analyzer uses the FPGA's resources:
  - Block RAM: Trigger and data;
  - Triggers Compare Logic (*Slice Logic*)

- So, it is necessary to reserve some FPGA resources for the logic associated with this tool.

Vivado Debug Design



Debug System Design

- **Create a Verilog RTL Project**
  - Ex **32-bits Counter**:

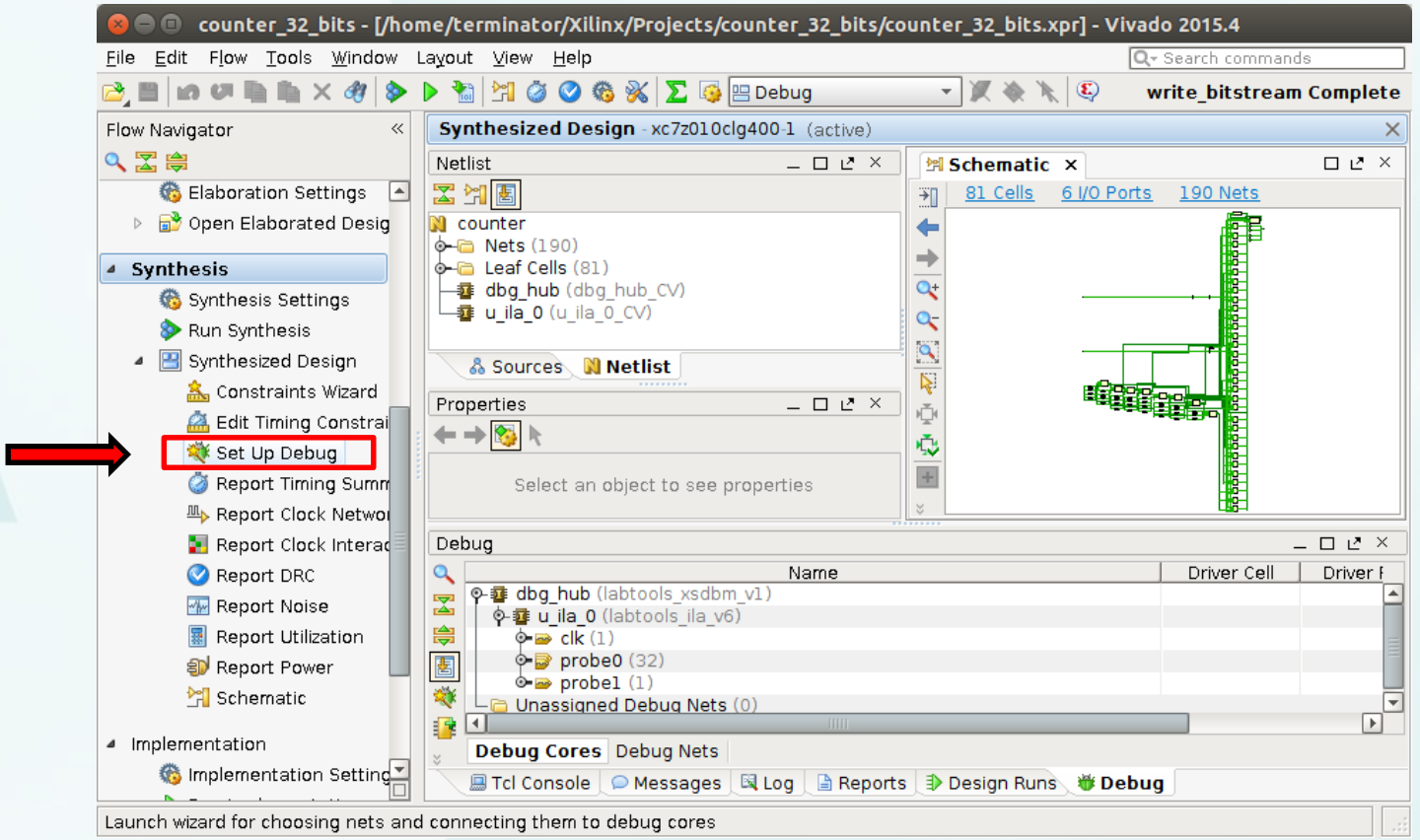
```
module top ( reset, clk, cnt );
        input clk;
        input reset;
        output [3:0] cnt;

        (* mark_debug="true" *) reg [31:0] counter;

        always @(posedge clk) begin
                if(reset)
                        counter <= 0;
                else
                        counter = counter + 1;
        end

        assign cnt = counter[31:28];

endmodule
```

Debug System Design

- Debug Main Steps:
  - Map I/O pins [**I/O Planning**]:
    - CLK → L16 (Zybo's External Clock)
    - Reset → T16 (Switch 1)
    - Cnt [Place automatically]

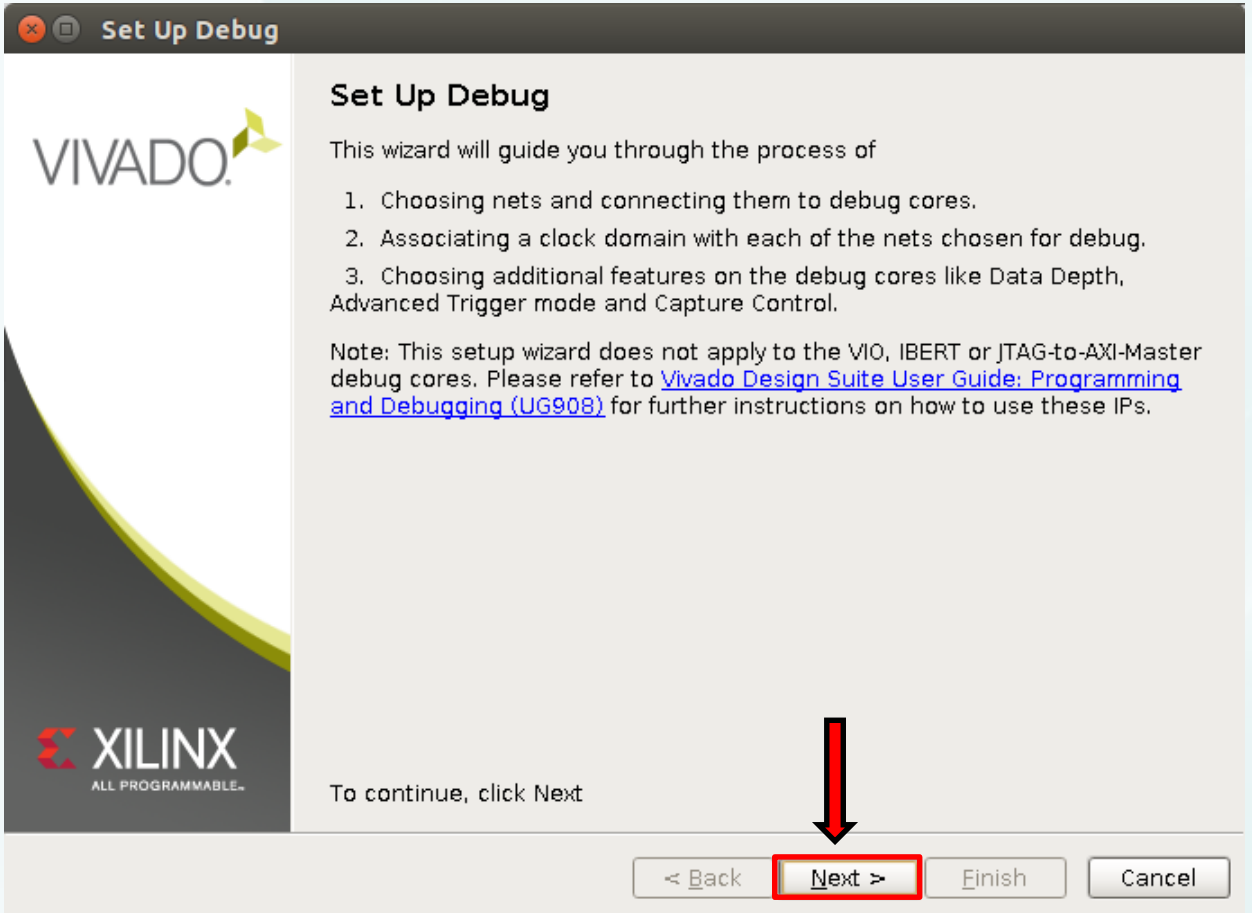
Debug System Design

- Debug Main Steps:
  - Synthesize Design
  - Open Synthesized Design
    - Setup Debug



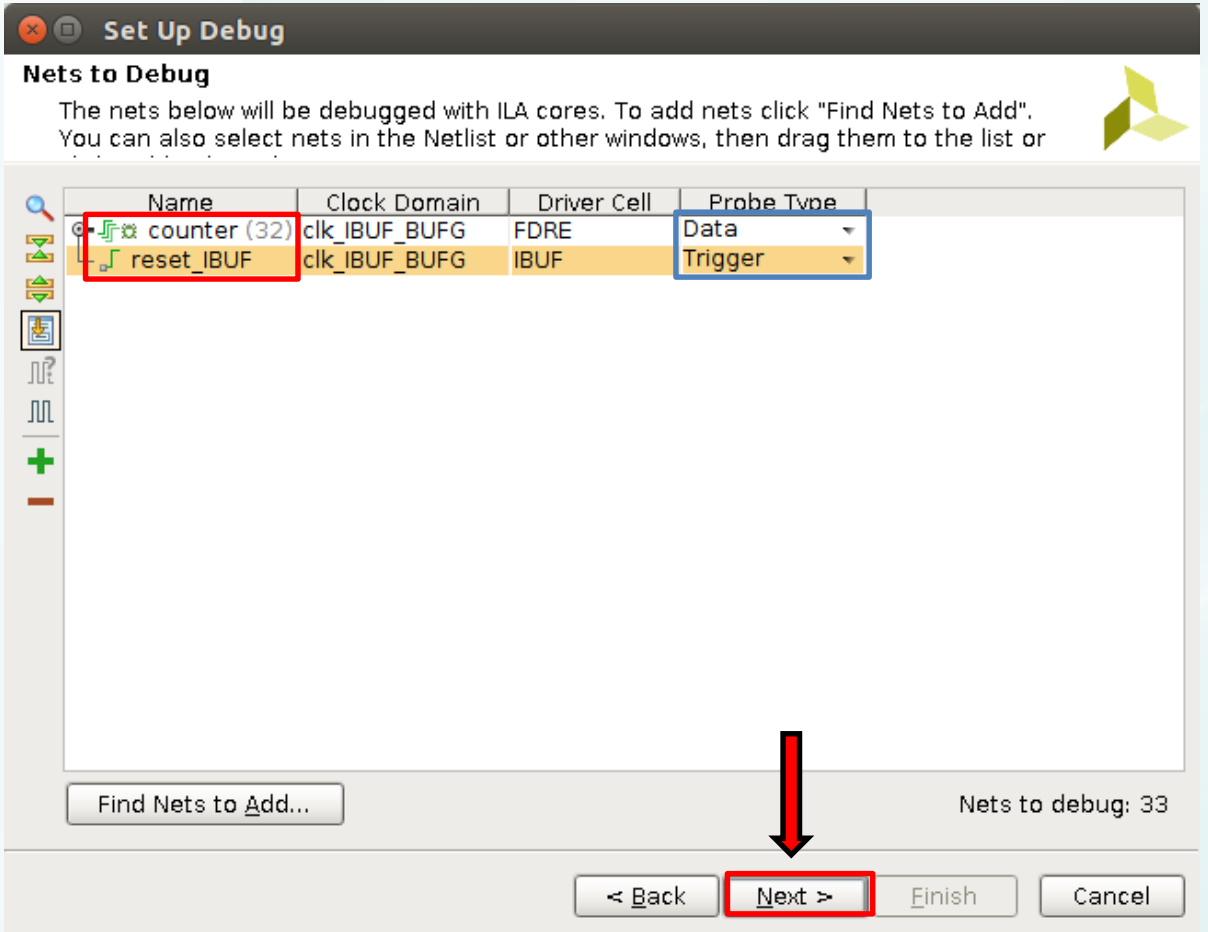
Debug System Design

- Debug Main Steps:
  - Click Next



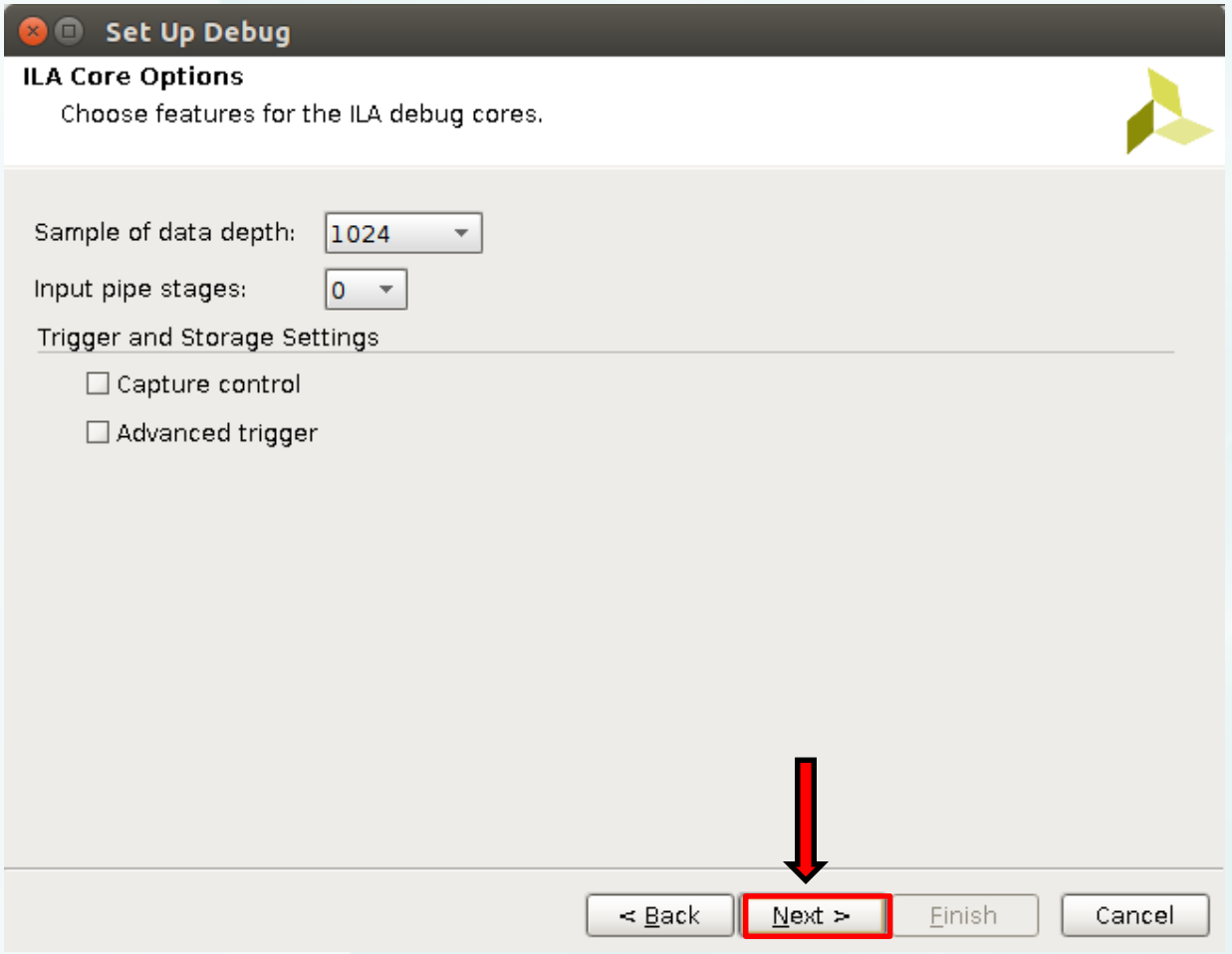
Debug System Design

- Debug Main Steps:
  - Select Data and Trigger signals
  - Click Next



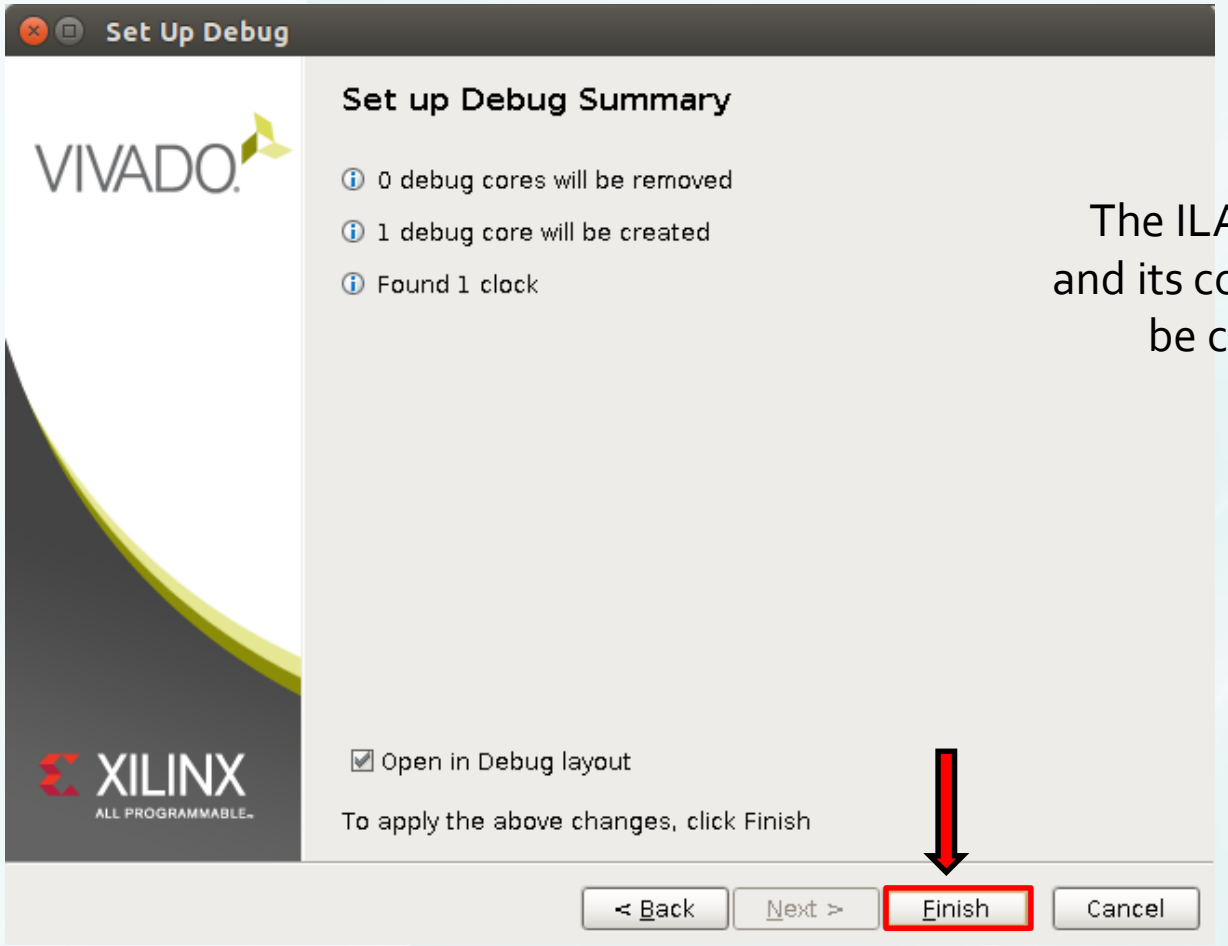
Debug System Design

- Debug Main Steps:
  - Click Next



Debug System Design

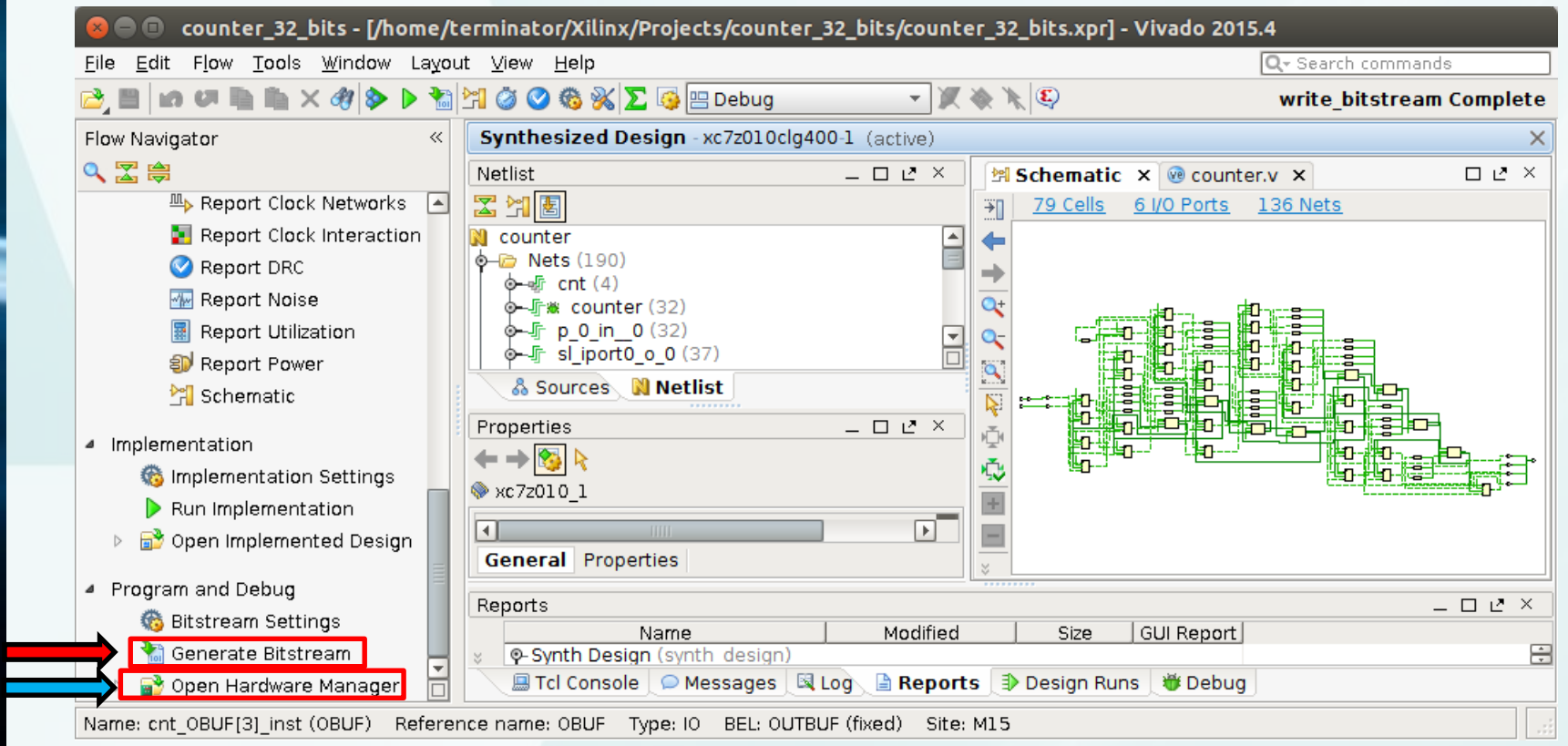
- Debug Main Steps:
  - Click Finish



The ILA debug core and its connections will be connected

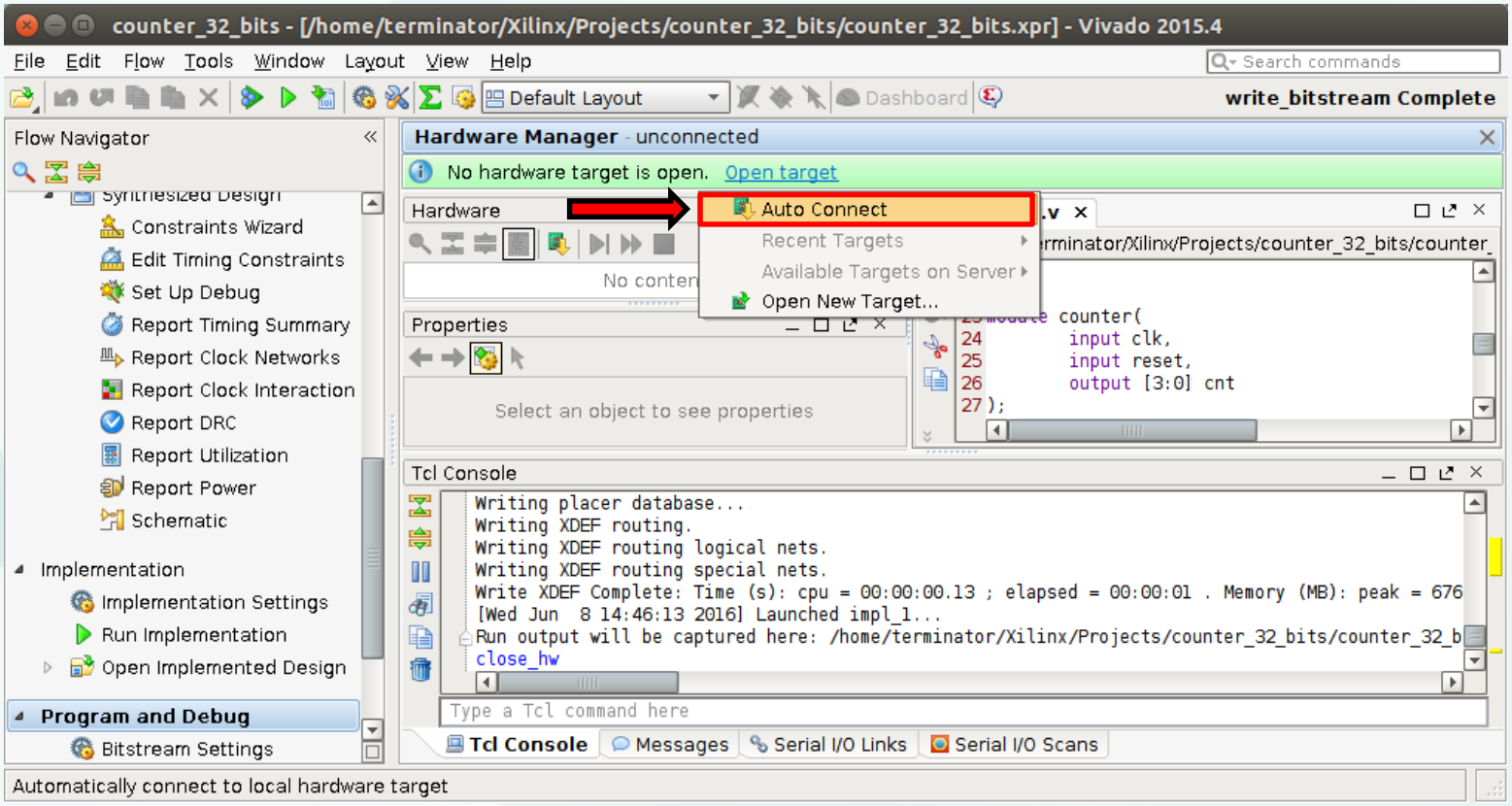
Debug System Design

- Debug Main Steps:
  - Implement Design
  - Generate Bitstream
  - Open Hardware Target



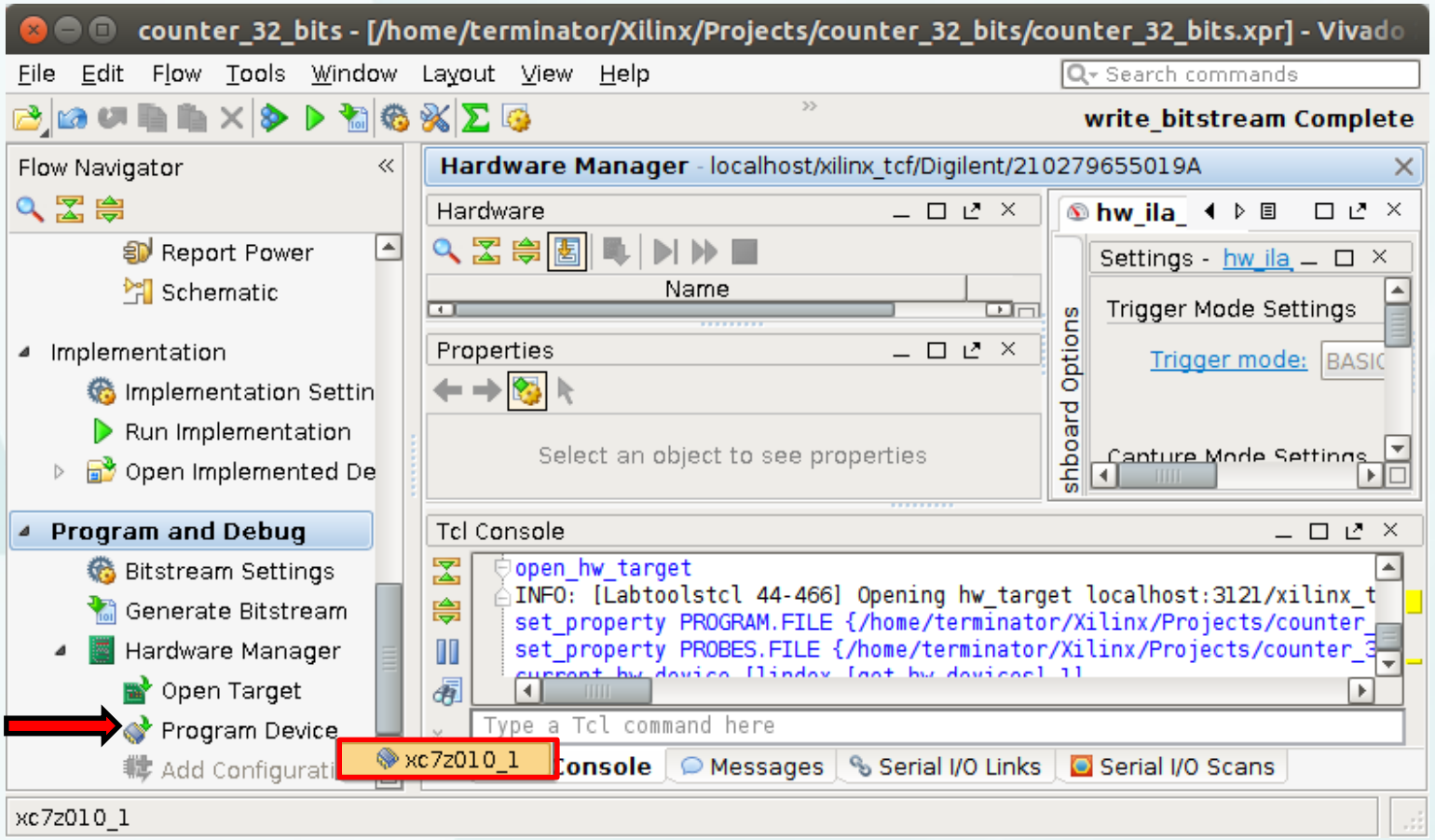
Debug System Design

- Debug Main Steps:
  - Open Target → Auto Connect



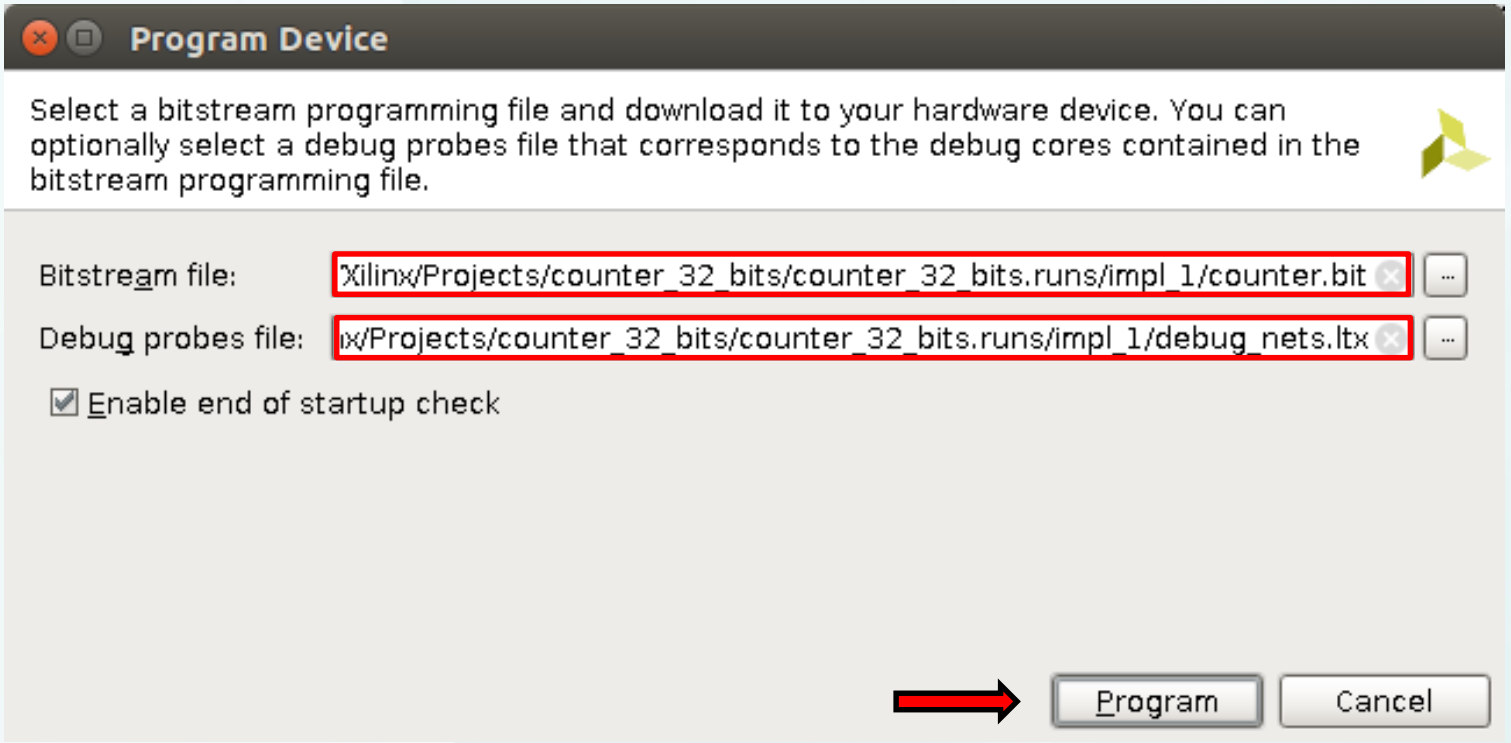
Debug System Design

- Debug Main Steps:
  - Program FPGA Device



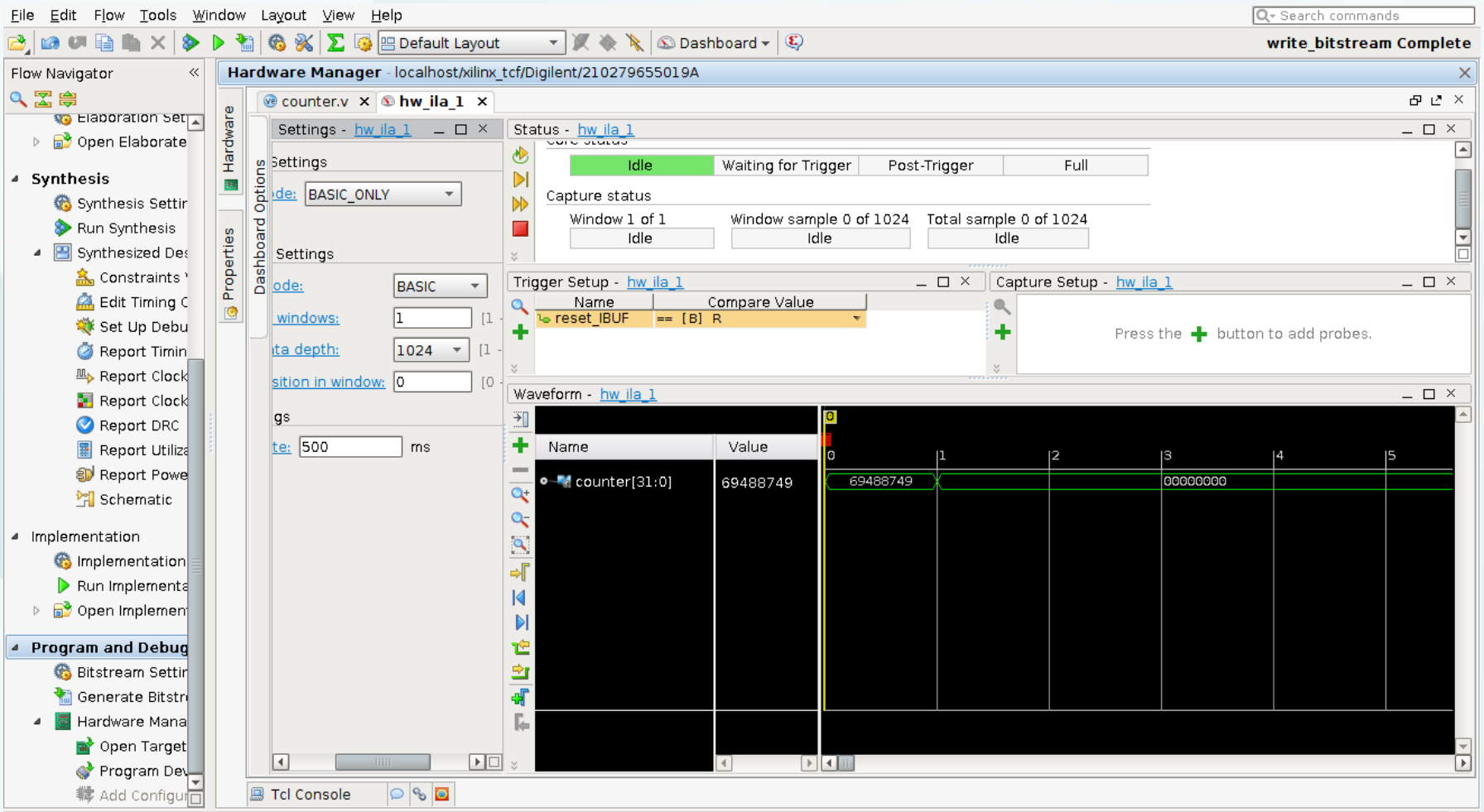
Debug System Design

- Debug Main Steps:
  - Program FPGA Device
    - Select **Bitstream** and **Debug Probes**



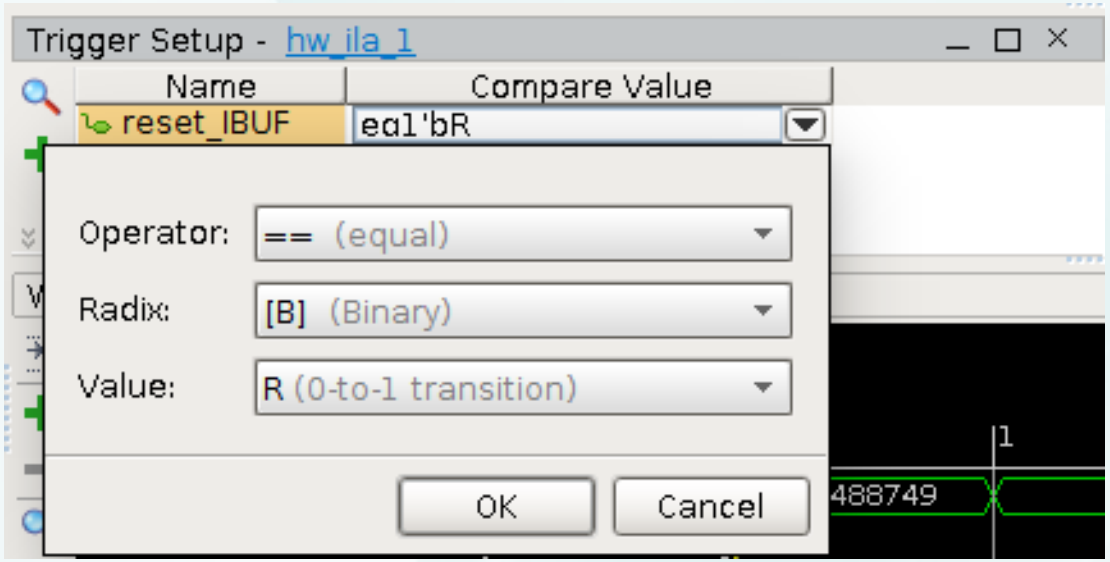
Debug System Design

- Vivado Logic Analyzer:



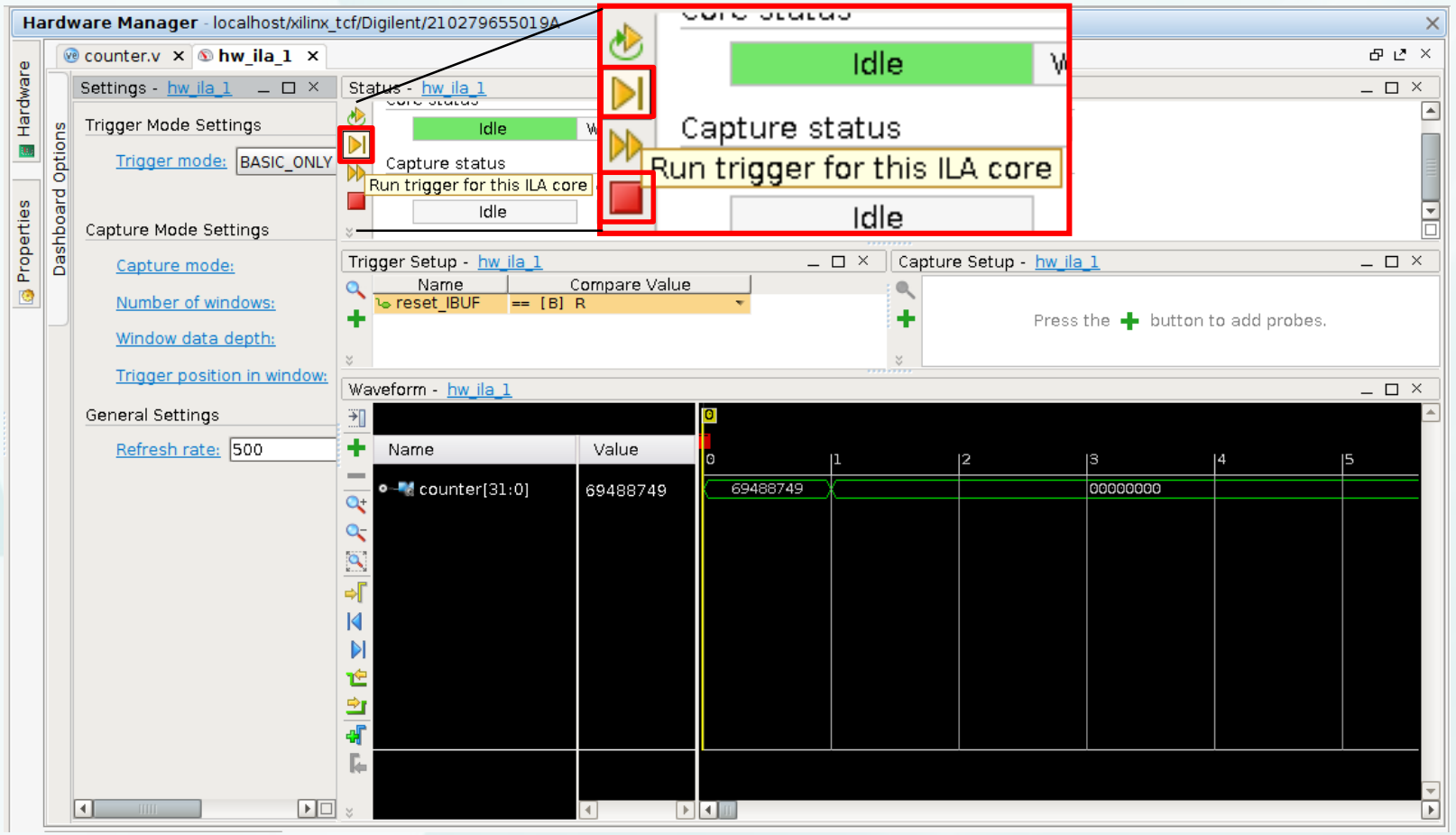
Debug System Design

- Vivado Logic Analyzer:
  - Setup Trigger
    - Select **Trigger Name** and **Compare Value**



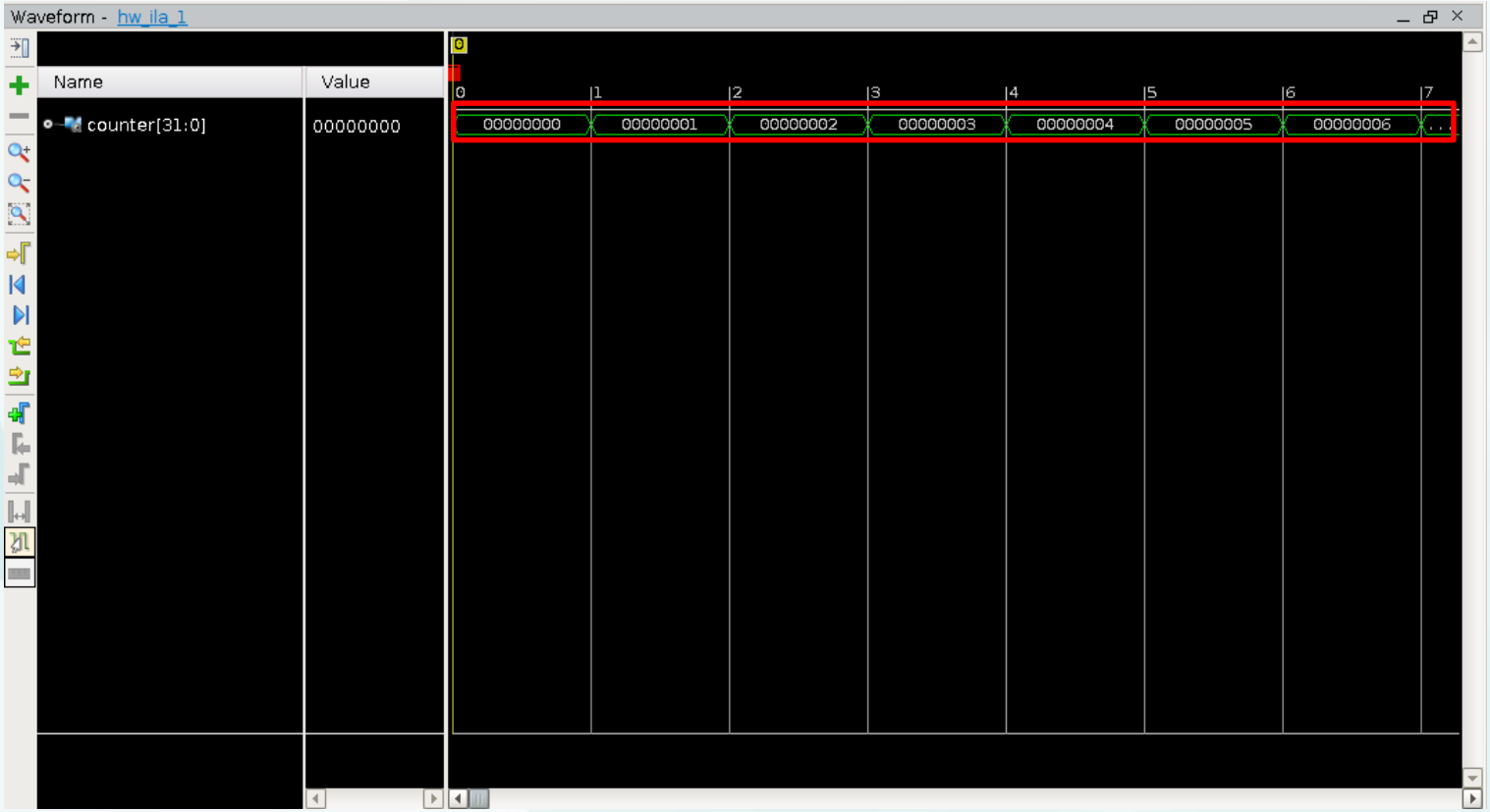
Debug System Design

- Vivado Logic Analyzer:
  - Start/Stop Capturing Samples



Debug System Design

- Vivado Logic Analyzer:
  - **Visualize Capture Data** after triggering the signal



Vivado Logic Analyzer

Questions?