Vivado从此开始（To Learn Vivado From Here）

本书围绕Vivado四大主题

- 设计流程
- 时序约束
- 时序分析
- Tcl脚本的使用

作者：高亚军（Xilinx战略应用高级工程师）

- 2012年2月，出版《基于FPGA的数字信号处理（第1版）》
- 2012年9月，发布网络视频课程《Vivado入门与提高》
- 2015年7月，出版《基于FPGA的数字信号处理（第2版）》
- 2016年7月，发布网络视频课程《跟Xilinx SAE学HLS》

- 内容翔实全面：涵盖Vivado所有基本功能
- 讲解深入浅出：结合大量案例，帮助读者加强对基本概念的理解
- 描述图文并茂：给出具体操作步骤，易于快速动手实践

Programming and Debugging

Lauren Gao

Agenda

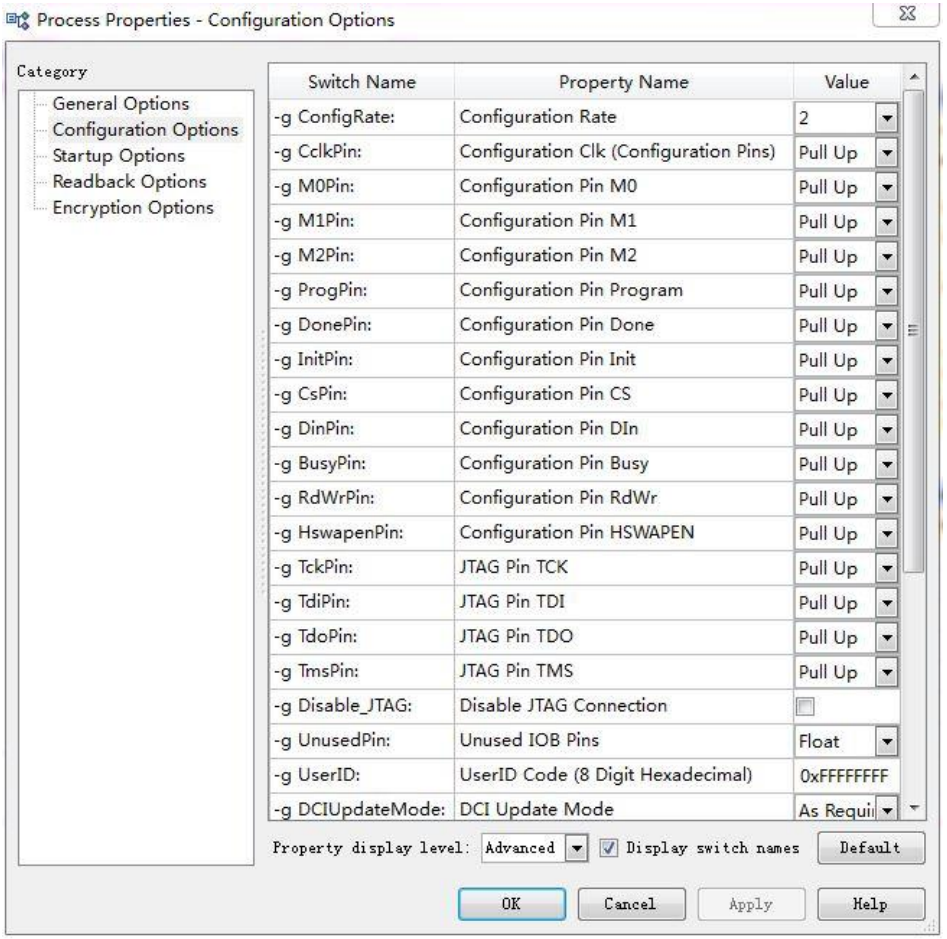
- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- Using a VIO Core for Debugging a Design in Vivado
- Using Tcl to Create Debug Unit
- Demo

Agenda

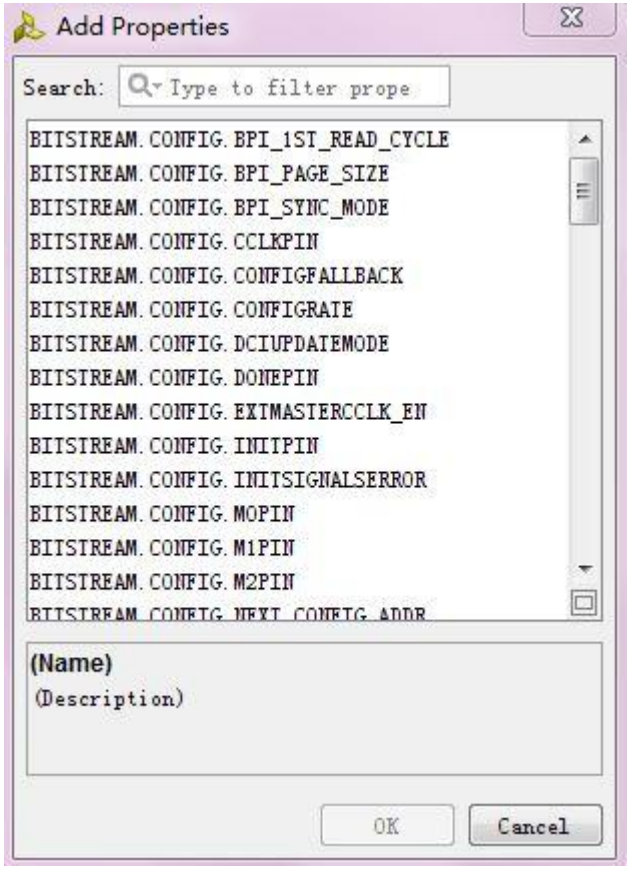
- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- Using a VIO Core for Debugging a Design in Vivado
- Using Tcl to Create Debug Unit
- Demo

Changing Device Configuration Bitstream Settings

- ISE: Generate Programming File → Process Properties



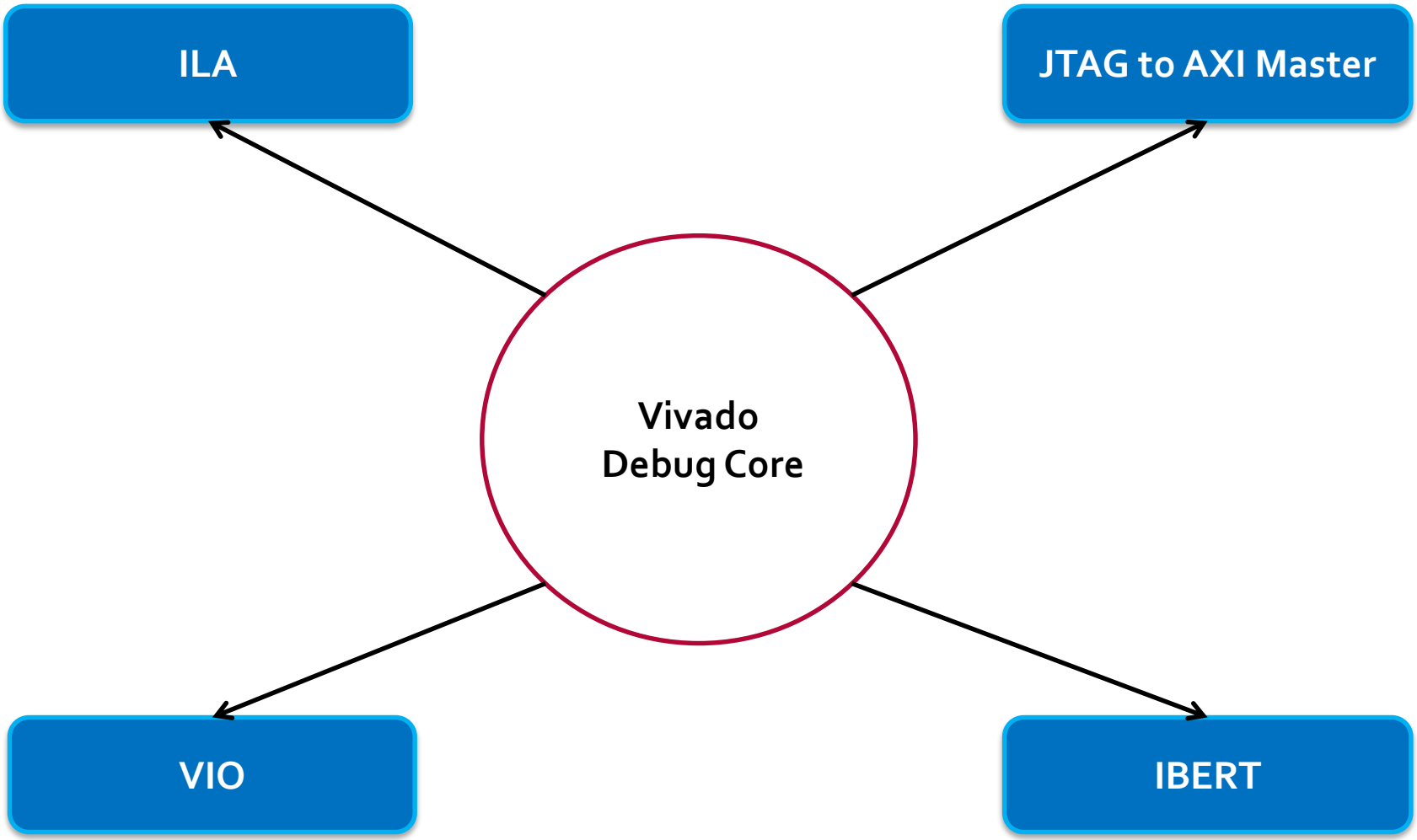
- Vivado: top netlist → Add property



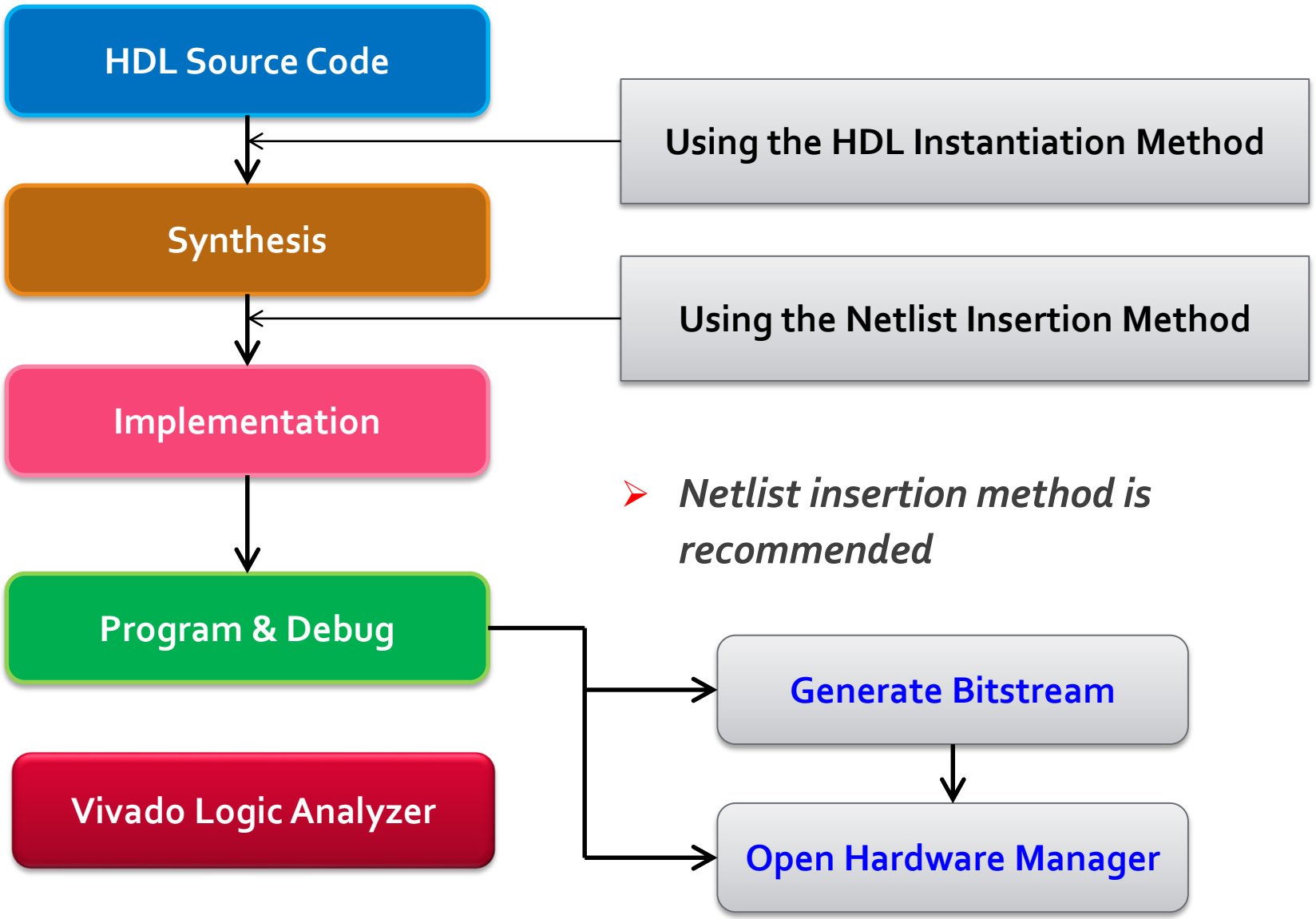
Agenda

- Changing Device Configuration Bitstream Settings
- **Using the Netlist Insertion Method for Debugging a Design in Vivado**
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- Using a VIO Core for Debugging a Design in Vivado
- Using Tcl to Create Debug Unit
- Demo

Vivado Debug Core



Two Methods to Add Debug Core in the Design



Using the Netlist Insertion Method

Synthesizing the Design

- **Synthesis settings:**
  - ✓ **-flatten_hierarchy**
    - **none**
    - **rebuilt**

Probing and Adding Debug IP

- **How to find target nets**
  - ✓ **In HDL source code**
  - ✓ **In Netlist view**
  - ✓ **In Schematic view**

Marking HDL Signals for Debug (Pre-Synthesis)

- For an RTL netlist-based project

Vivado: VHDL/Verilog

```
attribute mark_debug : string;
attribute mark_debug of char_fifo_dout: signal is "true";
```

```
(* mark_debug = "true" *) wire [7:0] char_fifo_dout;
```

Synplify: VHDL/Verilog

```
attribute syn_keep : boolean;
attribute mark_debug : string;
attribute syn_keep of char_fifo_dout: signal is true;
attribute mark_debug of char_fifo_dout: signal is "true";
```

```
(* syn_keep = "true", mark_debug = "true" *)
wire [7:0] char_fifo_dout;
```

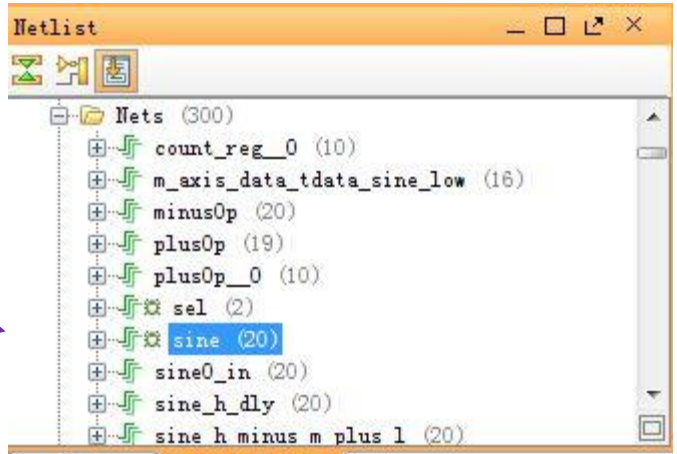
Marking Nets for Debug in the Synthesized Design (Post-Synthesis)

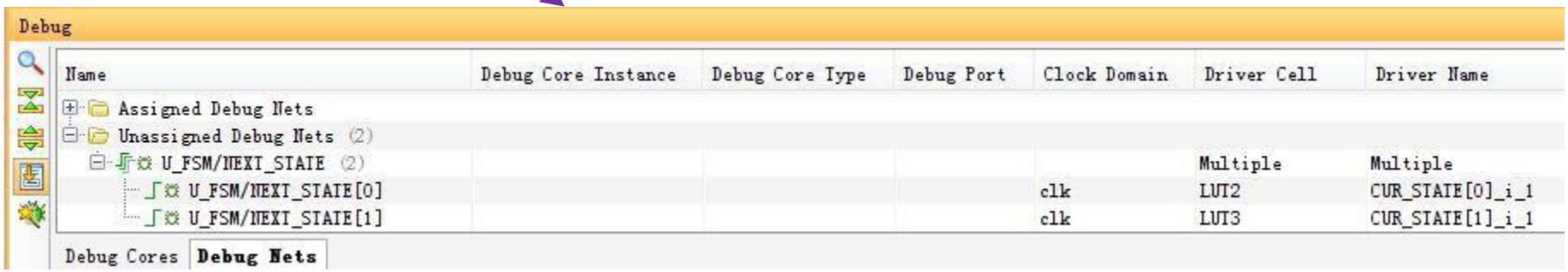
- For a synthesized design (Open synthesized design firstly)

Netlist view → Select net →
Right click → Mark Debug

Schematic view → Select net →
Right click → Mark Debug

Netlist/Schematic view → Select net
→ Dragging into Unassigned Debug Nets folder





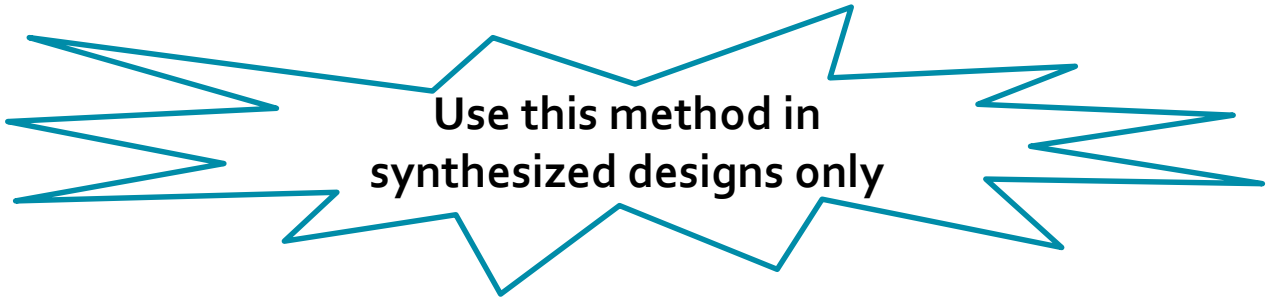
Using Tcl to Set mark_debug Attribute

- For a synthesized design (Open synthesized design firstly)

```
set_property mark_debug true [get_nets sine*]
```

- Confirm get_nets as expected

```
select_objects [get_nets sine*]
```



- You can use get_nets in conjunction with other get_* to find your target nets effectively

Agenda

- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- **Using the HDL Instantiation Method for Debugging a Design in Vivado**
- Using a VIO Core for Debugging a Design in Vivado
- Using Tcl to Create Debug Unit
- Demo

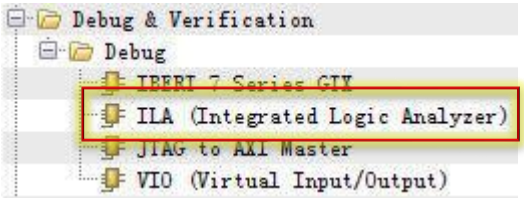
Using the HDL Instantiation Method

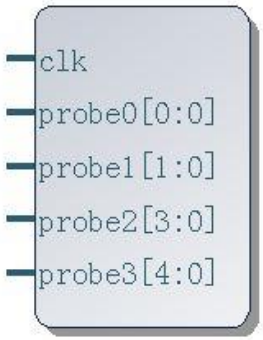
- In HDL source code, use "KEEP" or "DON'T_TOUCH" to avoid target signals being optimized

```
attribute keep : string;
attribute keep of sineSel : signal is "true";
attribute keep of sine : signal is "true";
```

- Get a probe list. Each probe connects to one signal. They have the same width
- Generate ILA ip core and add it to the project
- Instantiate the ILA in HDL source code

Where is ICON?

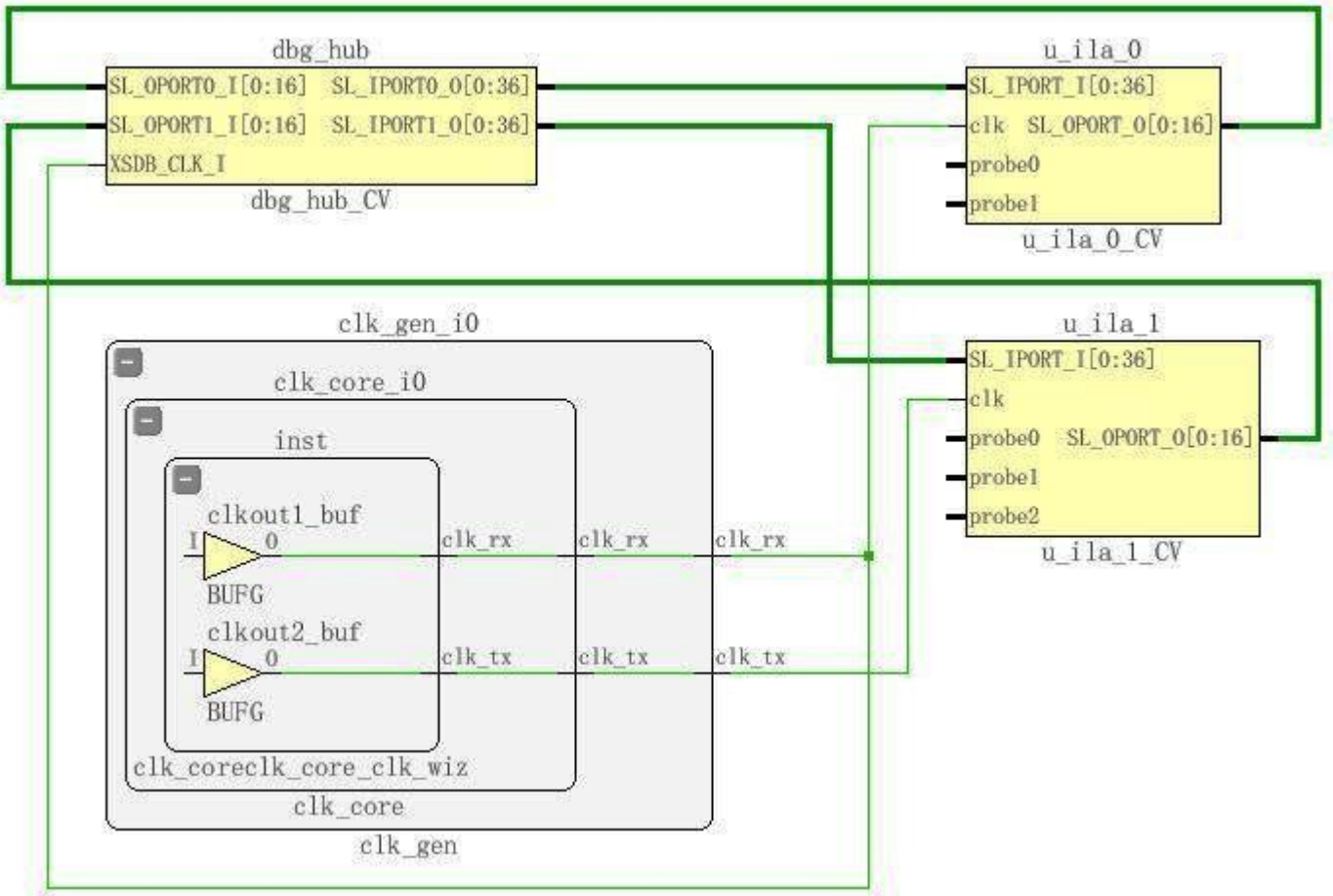





```
207  U_ILA : ila_0
208    port map
209    (
210      CLK => clk,
211      PROBE0 => sineSel,
212      PROBE1 => sine,
213      PROBE2 => GPIO_BUTTONS_db,
214      PROBE3 => GPIO_BUTTONS_re,
215      PROBE4 => GPIO_BUTTONS_dly,
216      PROBE5 => GPIO_BUTTONS
217    );
```

Migrate from ICON to dbg_hub

- Vivado can automatically generate dbg_hub after generating ILA
- ICON is replaced by dbg_hub which is easy to use

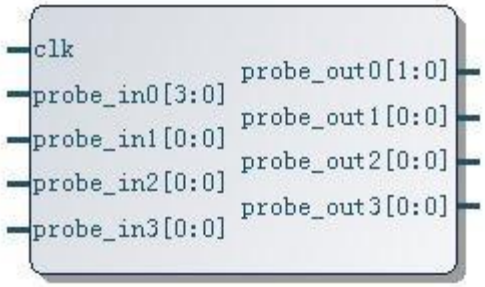


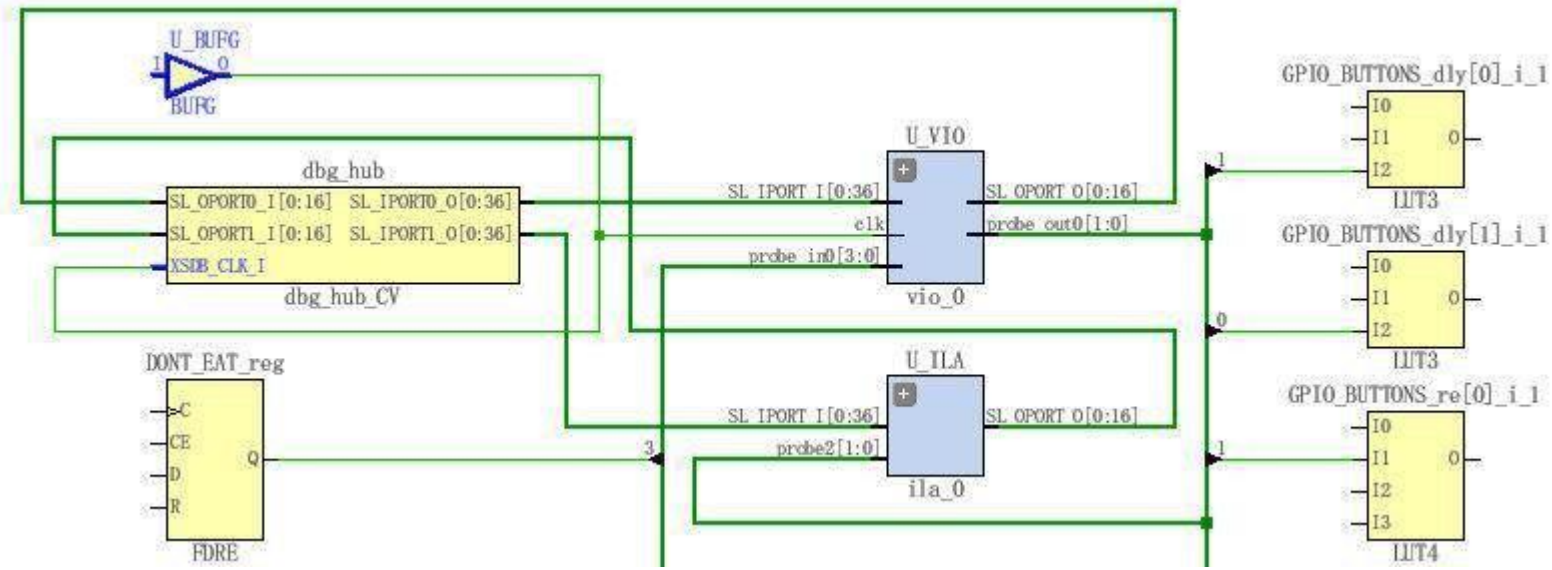
Agenda

- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- **Using a VIO Core for Debugging a Design in Vivado**
- Using Tcl to Create Debug Unit
- Demo

VIO

- **Virtual Input/Output (VIO) core**
  - Both monitor and drive internal FPGA signals in real time
  - Synchronous to the design being monitored and/or driven
  - Can only be used with HDL source code





Agenda

- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- Using a VIO Core for Debugging a Design in Vivado
- **Using Tcl to Create Debug Unit**
- Demo

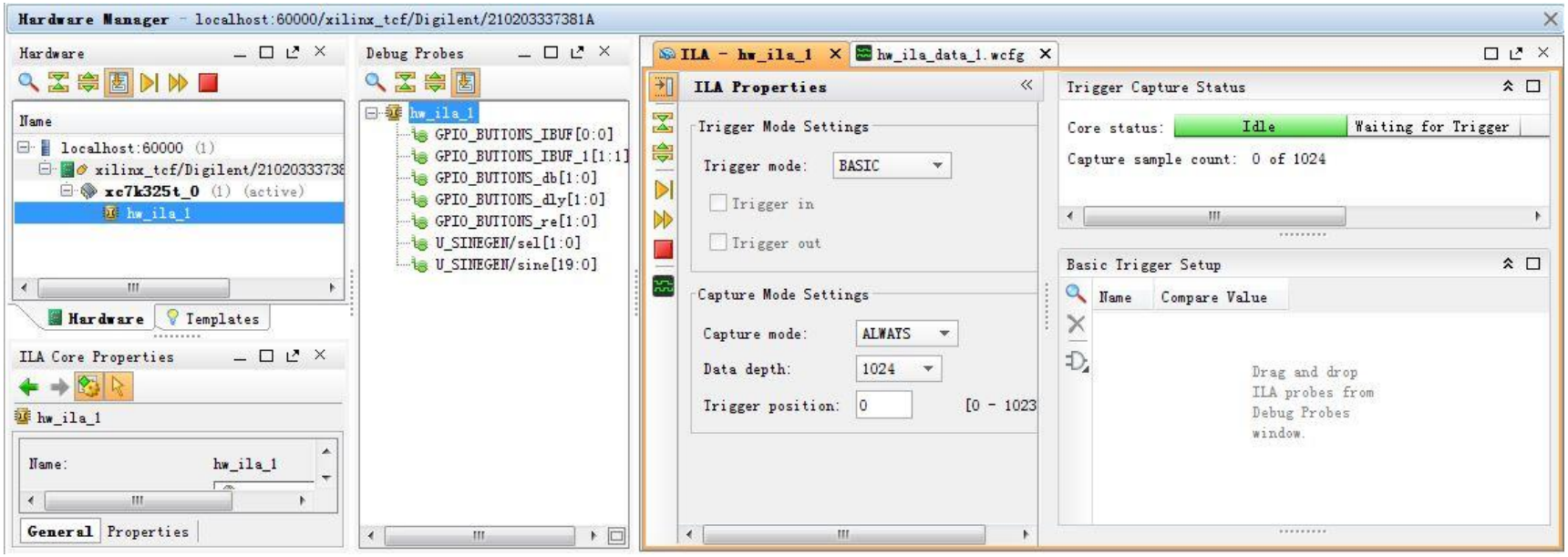
Using Tcl to Create Debug Unit

```
create_debug_core u_ila_0 labtools_ila_v3
set_property port_width 1 [get_debug_ports u_ila_0/clk]
connect_debug_port u_ila_0/clk [get_nets [list clk]]
set_property port_width 2 [get_debug_ports u_ila_0/probe0]
connect_debug_port u_ila_0/probe0 \
[get_nets [list {sel[0]} {sel[1]}]]
create_debug_port u_ila_0 probe
set_property port_width 2 [get_debug_ports u_ila_0/probe1]
connect_debug_port u_ila_0/probe1 \
[get_nets [list {GPIO_BUTTONS_db[0]} {GPIO_BUTTONS_db[1]}]]
```

- **Disconnect debug port**

```
disconnect_debug_port u_ila_0/probe1
```

Hardware Manager



- Two files are necessary
  - ✓ .bit
  - ✓ debug_nets.ltx: probes information files
  - ✓ .ltx can be generated by write_debug_probes Tcl command

```
set_property PROGRAM.FILE {C:/design.bit} [lindex [get_hw_devices] 0]
set_property PROBES.FILE {C:/design.ltx} [lindex [get_hw_devices] 0]
```

Agenda

- Changing Device Configuration Bitstream Settings
- Using the Netlist Insertion Method for Debugging a Design in Vivado
- Using the HDL Instantiation Method for Debugging a Design in Vivado
- Using a VIO Core for Debugging a Design in Vivado
- Using Tcl to Create Debug Unit
- Demo