Using ChipScope

Greg Gibeling & Chris Fletcher

February 21, 2009

Overview

ChipScope is an embedded, software based logic analyzer. By inserting an "integrated controller core" (**icon**) and an "integrated logic analyzer" (**ila**) into your design and connecting them properly, you can monitor any or all of the signals in your design. ChipScope provides you with a convenient software based interface for controlling the "integrated logic analyzer," including setting the triggering options and viewing the waveforms.

There are seven main steps to using ChipScope:

1. Create a new Xilinx COREGen project.
2. Generate an "integrated controller core" or **icon**.
3. Generate one or maybe more "integrated logic analyzers" or **ila**s.
4. Connect the **ila**s to the **icon** and make all of these modules part of your design.
5. Synthesize, and implement your design (including the **icon** and **ila**) as normal.
6. Program the FPGA board.
7. Run the ChipScope software to access and use the **ila**s (the ChipScope software requires the **icon** to gain access to the **ila**s).

1: Creating a new Project

1. Open **Start** ⟶ **Programs** ⟶ **Xilinx ISE Design Suite 10.1** ⟶ **ISE** ⟶ **Accessories** ⟶ **CORE Generator**.
2. Open **File** ⟶ **New Project.**
   (a) Choose where to save project files.
      i. **Create a new directory** and open it.
      ii. Save the coregen.cgp file in this directory.
      iii. This is where your ICON and ILA cores (and their metadata files) will be generated.

   (b) Under the **Part** tab, select all of the same device settings as you do when you setup a normal Xilinx ISE Project.
   (c) Under the **Generation** tab, change **Design Entry** to **Verilog**.
   (d) Select Ok to return to the main screen.
3. On the main screen, go to the **View By Function** tab.
   (a) Select the **Debug and Verification** folder.
   (b) Select **Chip Scope Pro**.
   (c) You should now be looking at a list of Chip Scope "cores." Among them should be **ICON**, **ILA**, and **VIO**. We will be using ICONs and ILAs.

2: Generating the ICON

1. Double-click on **ICON (ChipScope Pro Integrated Controller)**. A window that will allow you to customize your ICON should have appeared.
    (a) **Component Name**: Assign your ICON a name (this is arbitrary).
    (b) Select the correct **Number of Control Ports**.
    - Each **ILA** requires one control port.
    - Normally you will only need **1 Control Port**.
    - **If you generate an ICON with multiple control ports, you must connect every control port to an ILA. If you have "hanging" control ports in your design, Xilinx ISE will fail during synthesis.**

    (c) Leave **Disable Boundary Scan** unchecked.
    (d) Leave **Enable Unused Boundary Scan Ports** unchecked.
2. The ChipScope Pro Core Generator will now generated the ICON core according to the settings you specified. If you have errors go back and make sure you followed the above instructions.
3. Click **Finish** to Return to the main screen.
4. Your ICON will have been created in the directory you specified to store your project.

3: Generating the ILA

1. Once again on the main screen, double-click on **ILA (ChipScope Pro - Integrated Logic Analyzer).** A window that will allow you to customize your ILA should have appeared.
    (a) **Component Name:** Same as with the ICON, you may set this to whatever you like.
    (b) Normally you will only need **1 Trigger Port**.
    (c) Leave **Enable Output Trigger Port** unchecked.
    (d) Leave **Sample On to Rising** (you will sample on the rising edge).
    (e) Select the desired **Sample Data Depth**.
    - This is the number of samples the ILA will capture after it receives the trigger. It will capture one sample per clock cycle until it captures this many samples.

    (f) Leave **Data Same as Trigger** unchecked.
    - The ILA can have separate data and trigger inputs. **For clarity's sake, it is typical to always have different trigger and data ports.**

    (g) **Data Port Width** is only available when you are using separate trigger and data ports. Set this to the number of bits you will need to see in the wave window of ChipScope.
    (h) **Click Next** to go to the next page.
    (i) Set the **Trigger Port Width** to the number of bits that you need to trigger on.
    (j) You will most likely need **1 Match Unit**.
    (k) Leave **Counter Width** to **Disabled**.
    (l) Leave Match Type set to basic.
2. The ChipScope Pro Core Generator will now generated the ICON core according to the settings you specified. If you have errors go back and make sure you followed the above instructions.
3. **Click Finish** to Return to the main screen.
4. Your ILA will have been created in the directory you specified to store your project.

4: Connecting the Cores to Your Design

1. Declare a control bus for each ILA, similar to: `wire [35:0] ILAControl;`
   - Remember, each ILA you want to add to your design will require a control bus.
   - You can route control busses as input/outputs if you want to instantiate the ICON somewhere other than where you instantiate the ILA.
2. Instantiate the ICON core: `icon i_icon(.CONTROL0(ILAControl));`.
   - Remember to only instantiate **one** ICON. **Your design can never have more than one.**
3. Instantiate the ILA core: `ila i_ila(.CONTROL(ILAControl), .CLK(Clock), .TRIG0(/**/), .DATA(/* */));`.
   - You may instantiate ILAs wherever they are necessary, just make sure to route the control bus from the ICON appropriately
   - You may have one or more ILAs in your design.
   - **Your ILA may have different ports. Be sure to read the verilog example produced by the ChipScope Core Generator.**
4. On both the ILA and ICON core instantiations, add **Synthesis no-prune directives**. Since the ICON and ILA do not output anything from your design, Synplify Pro will think that they have no impact on the circuit and will try to "prune" or remove them. Program 1 shows example ICON/ILA instantiations that have the necessary synthesis directives added.

Program 1 Example ICON/ILA instantiations with the **noprun** Synthesis directives properly added

```
icon_1 icon(.CONTROL0(  ILAControl)) /* synthesis syn_noprune=1 */;

ila_1 ila ( .CLK(       Clock),
            .CONTROL(   ILAControl),
            .TRIG0(     /**/),
            .DATA(      /**/)) /* synthesis syn_noprune=1 */;
```

Note that synthesis directives are normal **block comments** in Verilog, placed **after** the instantiation but **before** its closing semi-colon.

5. **Remember to look at the example Verilog files generated by the ChipScope Core Generator!**

5: Synthesize and Implement Your Design

1. Make sure to **add the verilog examples** generated by ChipScope Core Generator to your Xilinx ISE project as **Verilog Design Files**. (this is done through the **Add Source** option that you have been using to add modules to your design up to this point).
2. Make sure you set your **Macro Search Path** so that the search path directory contains all of the metadata files created for your cores.
   (a) In the **Sources** box, select **Implementation**.
   (b) Click on `FPGA_TOP_ML505` in the Sources box.
   (c) In the **Processes box**, right click on **Implementation**.
   (d) Under **Translate Properties**, set the **Macro Search Path** to the directory containing the ICON/ILA core files.
      i. Make sure to add the following shell Verilog file(s) to your project:

A. All `.v` files created for your ICON.
B. All `.v` files created for your ILAs.

ii. Set the Macro Search Path
  A. Make sure is highlighted in the **Sources** box.
  B. **Right-Click** on Implement Design in the **Processes** box.
  C. Go to the Translate Properties tab.
  D. Set the Macro Search Path to the **exact** directory where your copy(ies) of the `.edf` `/.ngc` files reside.

iii. Your project should now be able to build with black box files and core files properly.

3. Synthesize and implement your design as normal.

6: Program the ML505 Board

1. Make sure that the the programming Cable is connected to the **JTAG** Port on the `FPGA_TOP_ML505` board and that the `FPGA_TOP_ML505` board is on.
2. Run iMPACT from Xilinx ISE.
3. **You must close iMPACT or ChipScope will be unable to work correctly!**

7: Run ChipScope

1. Open **Start** $\longrightarrow$ **Programs** $\longrightarrow$ **Xilinx ISE Design Suite 10.1** $\longrightarrow$ **ChipScope Pro** $\longrightarrow$ **Analyzer.**
2. Make sure that the the programming Cable is connected to the **JTAG** Port on the `FPGA_TOP_ML505` board and that the `FPGA_TOP_ML505` board is programmed.
3. Once ChipScope Pro Analyzer is running you must connect to the programming cable.
   (a) Go to the **JTAG Chain** menu.
   (b) Select **Xilinx Parallel Cable**.
   (c) Select the **Xilinx Parallel Cable IV**.
   (d) Set the **Speed to 5MHz**.
   (e) Make sure the **Port** is set to **LPT1**.
   (f) Click **OK**.
   (g) In the next window you will see two chips listed.
      - The **System_ACE-CF** is not used in this class.
      - The **xc5vlx110t** is the FPGA.
   (h) Click **OK**.
4. The ChipScope Pro Analyzer should now be connected to the FPGA and running.
   (a) You can move the **Trigger Setup** and **Waveform** windows around as needed to be able to see the information you're looking for.
   (b) First you must use the **Trigger Setup** window to set a **trigger function**, just like with the Bench Logic Analyzers
   (c) When you have a trigger, click the **Run** button in the toolbar to start waiting for that trigger
   (d) When the **trigger occurs** ChipScope will start downloading data from the FPGA and show it in the **Waveform** window, much like ModelSim.
5. **Please experiment with ChipScope**, it is an invaluable tool for FPGA debugging.