ChipScope Demo Instructions

Overview

ChipScope is an embedded, software based logic analyzer. By inserting an "intergrated 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. Even nicer is that 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 six main steps to using ChipScope, as detailed below.

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

Detailed Instructions: Step 1 – Generating the ICON

1. First you will need to start the **ChipScope Core Generator**
   a. Go to **Start** -> **All Programs** -> **ChipScope Pro 6.1i** -> **ChipScope Core Generator**
   b. This will present you with the ChipScope core generator wizard.
2. Select the "**ICON (Integrated Controller)**" option and click **Next**
3. **General Options**
   a. You will need to change the **Output Netlist** location.
      i. By default ChipScope Core Generator will try and write the output to a directory you are not allowed to modfy.
      ii. Change the box from "**.\icon.edn**" to something like "**c:\users\cs150-xxx\icon.edn**"
      iii. You MUST include the "**icon.edn**" at the end of the path.
   b. Select the **VirtexE Device Family**
   c. Select the correct number of **Control Ports**
      i. Each **ILA** requires one control port
      ii. Normally you will only need **1 Control Port**
   d. Click **Next**
4. **Example Template Options**
   a. Make sure **Generate HDL Example Files** is **Checked**
      i. Select a **Verilog** example file

          ii.   Use the **Synplicity Synplify** synthesis tool
- b. You do not need to **Generate Batch Mode Argument Example Files**
- c. Click **Generate Core**
5. 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
6. When you are done click **Start Over** and proceed directly to **step 2** below.

Detailed Instructions: Step 2 – Generating the ILA

1. First you will need to start the **ChipScope Core Generator** if you haven't already started from the previous section.
- a. Go to **Start** -> **All Programs** -> **ChipScope Pro 6.1i** -> **ChipScope Core Generator**
- b. This will present you with the ChipScope core generator wizard.
2. Select the "**ILA (Integrated Logic Analyzer)**" option and click **Next**
3. **General Options**
- a. You will need to change the **Output Netlist** location.
  - i. By default ChipScope Core Generator will try and write the output to a directory you are not allowed to modfy.
  - ii. Change the box from "**.\ila.edn**" to something like "**c:\users\cs150-xxx\ila.edn**"
  - iii. You MUST include the "**ila.edn**" at the end of the path.
- b. Select the **VirtexE Device Family**
- c. Select **Rising Edge** trigger.
  - i. **Falling Edge** should only be used by expert debuggers who know EXACTLY what they are doing.
- d. Click **Next**
4. **Trigger Options**
- a. Normally you will only need **1 Trigger Port**
  - i. Set the **Trigger Width** to any amount you desire (probably either **1bit** or **32bits** for Lab5)
  - ii. Set the **Match Type** to **Basic** unless you need any more advanced features
  - iii. You will most likely need **1 Match Unit**
  - iv. Set the **Counter Width** to **Disabled** unless you need the counter
- b. **Disable Trigger Sequencing**
  - i. This is very useful for more advanced FSM debugging
- c. **Disable** the **Trigger Output Port**
- d. Click **Next**
5. **Data Options**
- a. Select the desired **data depth**
  - i. 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. You will probably need relatively few for Lab5.

b. **Data Same as Trigger** should be selected if you are going to connect all the signals you are interested in to the trigger.

> The bench logic analyzers in the lab have 16bits of input used for both the data and the trigger. The ILA can have separate data and trigger inputs. You may wish to use a one bit trigger connected to a "detected error" signal, or you may wish to simply connect the outputs you are interested in to the trigger port. It is up to you. If you wish to use a "detected error" signal then you should connect that to the trigger and connect the counter output to the data port. Otherwise simple make a very large trigger port and have it be the same as the data.

c. **Data 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.

d. **Number of BlockRAMs** is simply an indicator of how large the ILA you are making will be. If this number is above 20 you may be doing something incorrect.

e. Click **Next**

6. **Example and Template Options**
   a. Make sure **Generate HDL Example Files** is **Checked**
      i. Select a **Verilog** example file
      ii. Use the **Synplicity Synplify** synthesis tool
   b. You will need to **Generate Bus/Signal Name Example File**
   c. You do not need to **Generate Batch Mode Argument Example Files**
   d. Click **Generate Core**

7. 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

8. When you are done, exit ChipScope Core Generator, you now have all the files you need to begin integrating the logic analyzer into your design.

Detailed Instructions: Step 3 – Connecting the Cores to Your Design

1. Declare a control bus for each ILA, similar to the following:
   a. `wire [35:0]     ILAControl;`
   b. Remember, each ILA you want to add to your design will require a control bus.
   c. 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
   a. `icon        i_icon(.control0(ILAControl));`
   b. Remember to only instiate **ONE** ICON, your design can never have more than one.

3. Instaniate the ILA core
   a. `ila        i_ila(.control(ILAControl), .clk(Clock),`
      `.trig0(/**/));`

b. You may instantiate ILAs wherever they are necessary, just make sure to route the control bus from the ICON appropriately
c. You may have one or more ILAs in your design.
d. **Your ILA may have different ports** be sure to read the verilog example produced by the ChipScope Core Generator.

4. *REMEMBER TO LOOK AT THE EXAMPLE VERILOG FILES GENERATED BY CHIPSCOPE CORE GENERATOR!*

Detailed Instructions: Step 4 – Synthesize and Implement Your Design

1. Make sure to **add the verilog examples** generated by ChipScope Core Generator to your Xilinx project as **Verilog Design Files**.
    a. This will ensure the Synplify has the correct block box models for the ICON and ILA cores.
2. Synthesize and implement your design as normal.

Detailed Instructions: Step 5 – Program the CaLinx Board

1. Make sure that the the **Parallel Cable IV** is connected to the **Slave Serial** Port on the CaLinx board and that the CaLinx board is on.
    a. The little light on the Parallel Cable IV will turn green when the cable detects that it is connected to a powered Xilinx chip.
2. Run **iMPACT** from **Project Navigator**
    a. When it asks select **FPGA_TOP.bit** as before
    b. **Right click** on the picture of the FPGA ant select **program**
3. *YOU MUST CLOSE IMPACT OR CHIPSCOPE WILL NOT BE ABLE TO WORK!*

Detailed Instructions: Step 6 – Run ChipScope

1. Once iMPACT is closed, double click **Analyze Design Using ChipScope** in Project Navigator.
2. Make sure that the the **Parallel Cable IV** is connected to the **JTAG** Port on the CaLinx board and that the CaLinx board is programmed, the **DONE** LED will light up green when the board is programmed..
    a. The little light on the Parallel Cable IV will turn green when the cable detects that it is connected to a powered Xilinx chip.
3. Once ChipScope Pro Analyzer is running you must connect to the parallel 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
        i. The **System_ACE-CF** is not used in this class
        ii. The **XCV2000E** 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.