**Using the Integrated Logic Analyzer (ILA)**

**Acknowledgement**

This tutorial refers to Integrated Logic Analyzer v6.1 Product Guide from Xilinx.

**Goal**

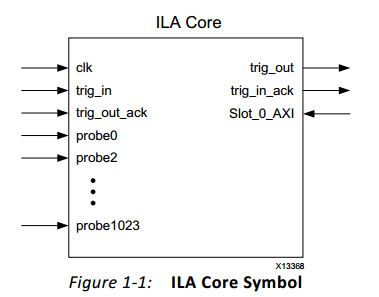
* Use the ILA to monitor the internal signals of a design during runtime
* Set up a trigger event using the Vivado logic analyzer

**Requirements**

* Xilinx Vivado software
* Xilinx SDK software
* Xilinx Nexys 4 board and a programming cable
* Enough disk space for the project files

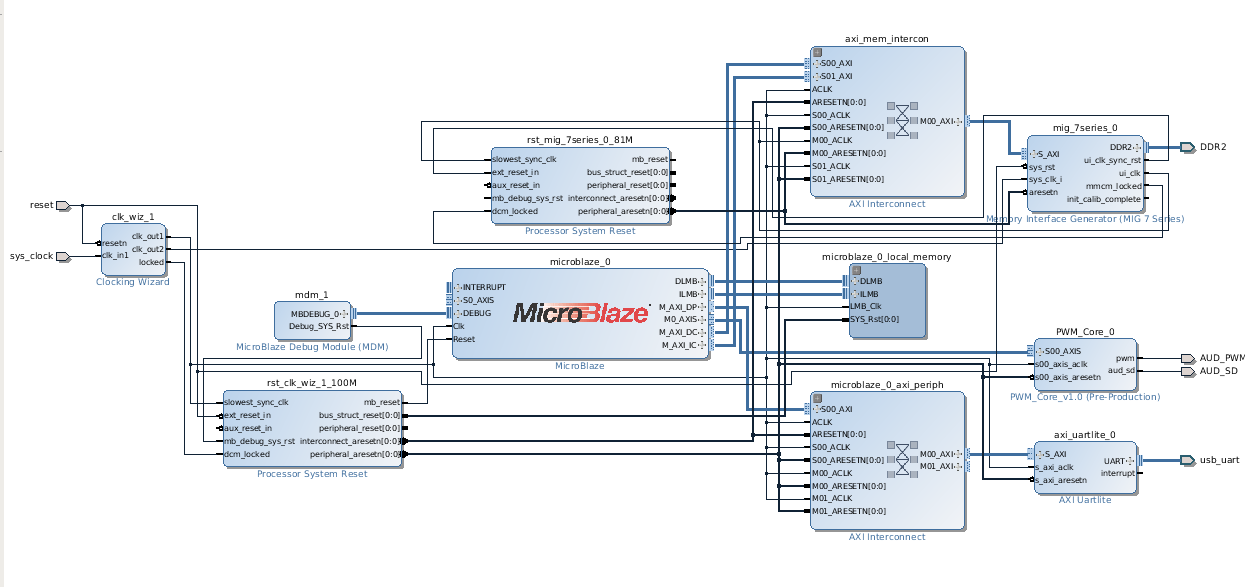
**Background**

The Vivado Integrated Logic Analyzer is a core that can be used to monitor the internal signals of your FPGA in real time through the Vivado logic analyzer software. The data is stored in the On-chip block RAM memory before it is uploaded by the software. After a trigger occurs, you can view the data captured in the waveform window of the Vivado Logic Analyzer.



**Connecting ILA Module to Previous Audio Project**

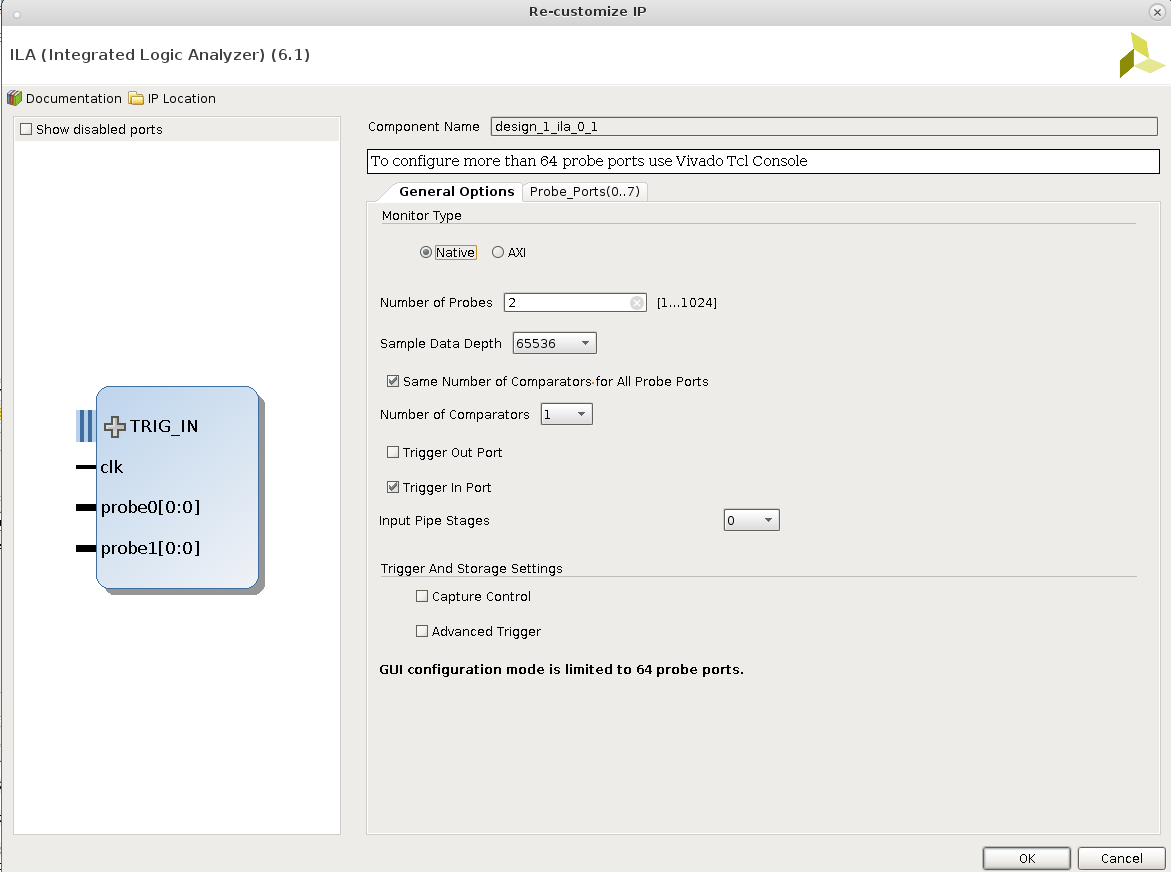
1. Open the block diagram of the previous audio project

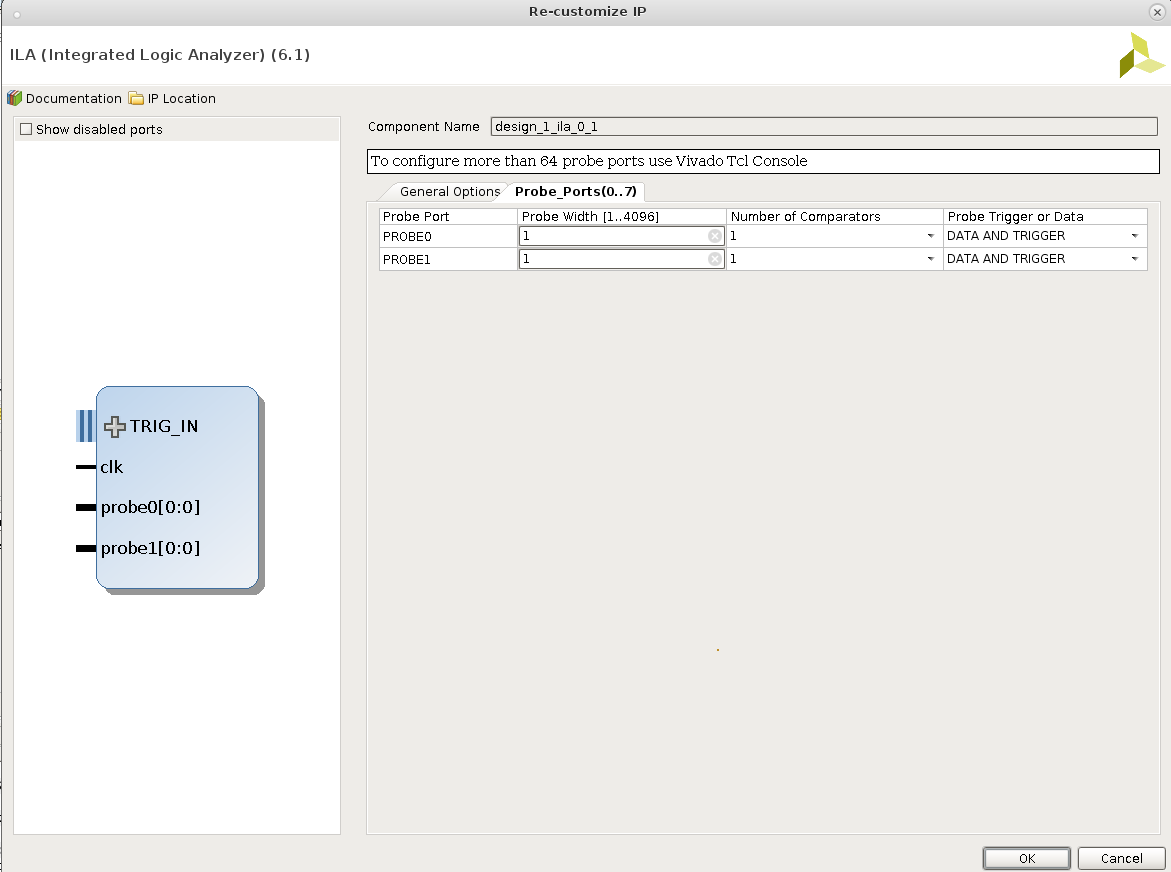


2. **Add IP**. Search for **ILA (Integrated Logic Analyzer)**

3. Double click on the ILA module and customize it.

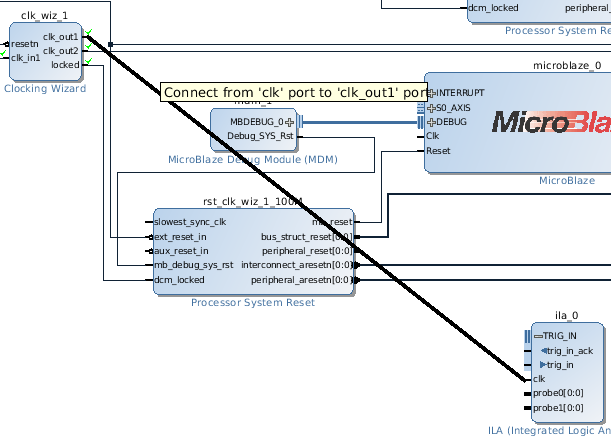
* Choose **Native** for Monitor Type
* Change the **number of probes** to 2
* Change **Sample Data Depth** to 64k
* Check the **Trigger In Port**
* Open **Probe** tab. Make sure it is same as in the picture

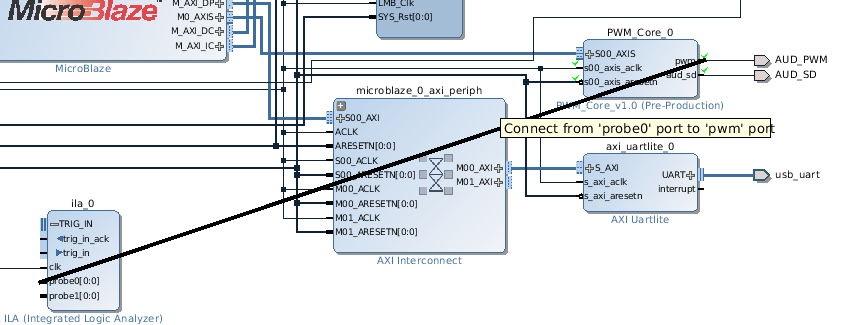




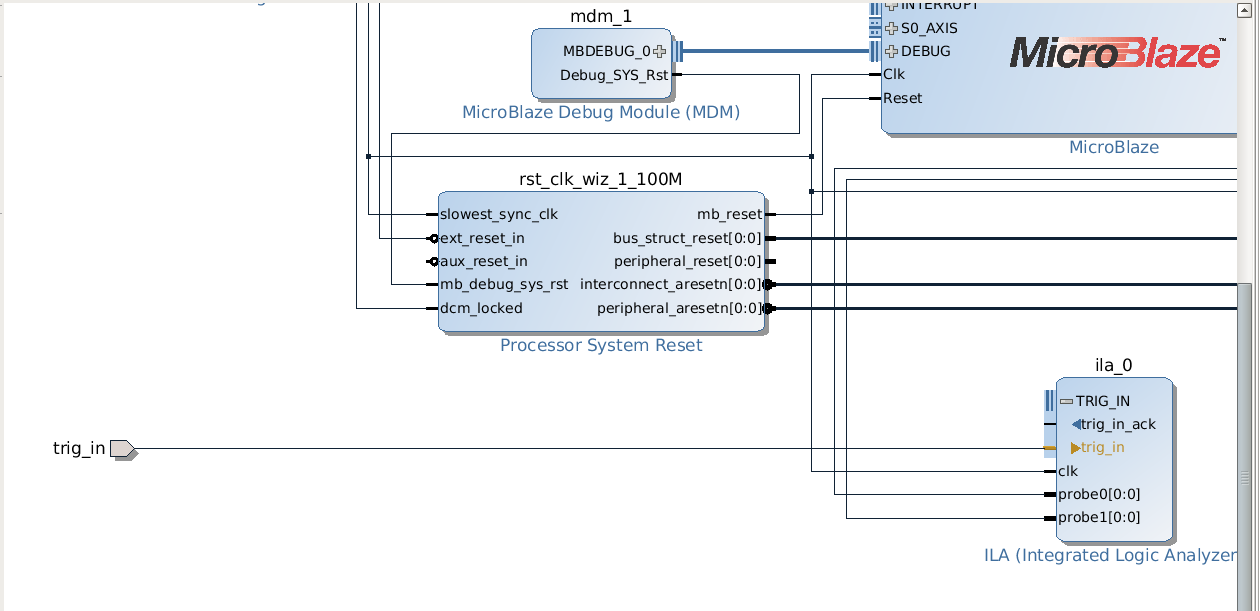
4. Connect the wires.

* clk to system clock 100mhz. This clock indicates sampling rate and must be sufficient for the PWM core
* Connect probe0 to AUD\_PWM
* Connect probe1 to AUD\_SD

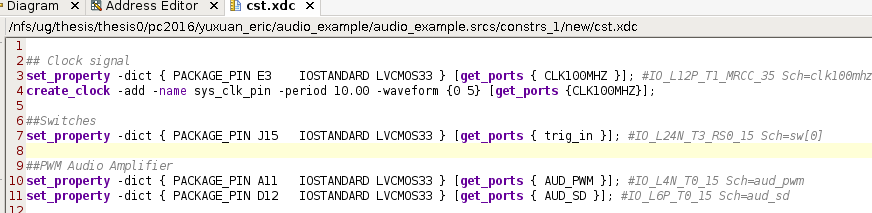




5. Expand trig\_in. Right click trig in, make external.



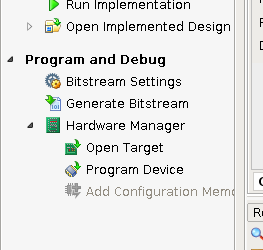
6. Change the constraint file. We connect the trig\_in signal to SW[0] (Port J15).



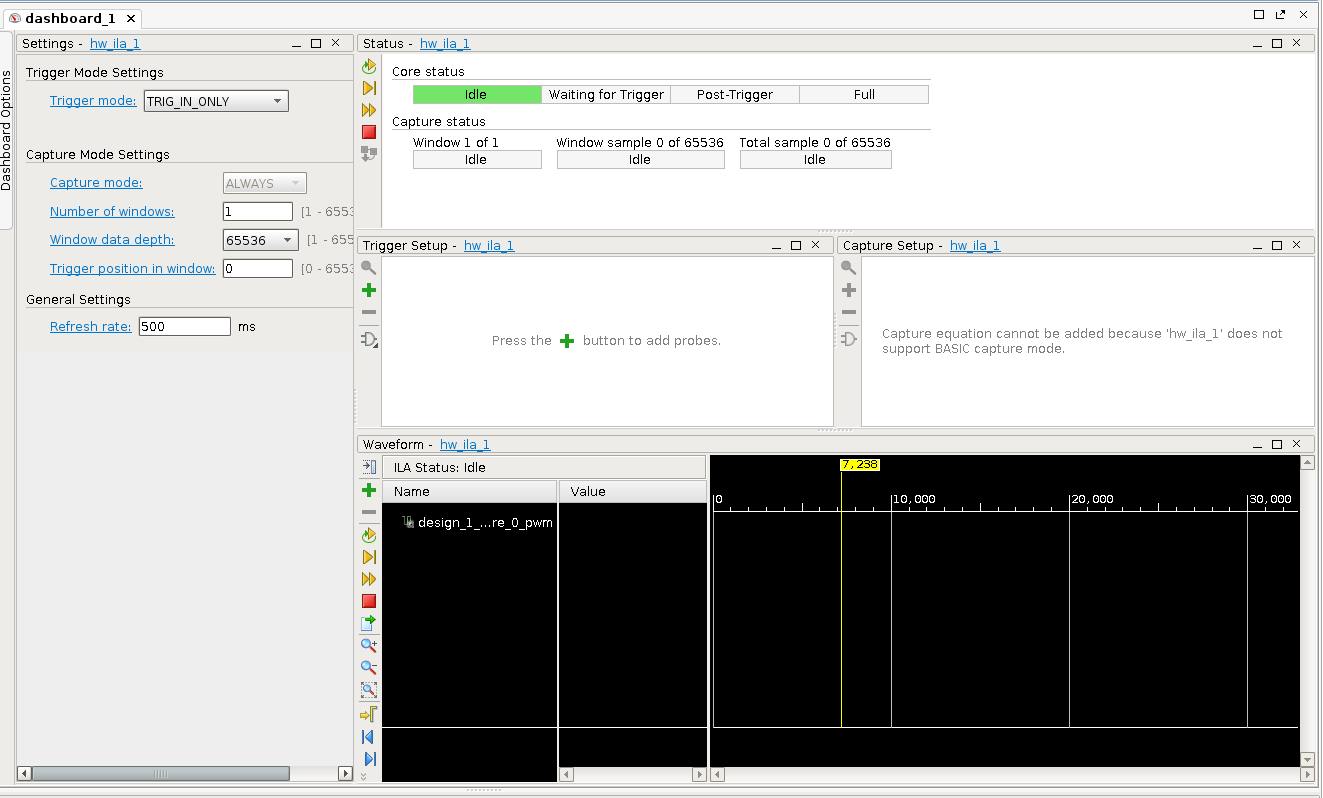
Remember to delete the original SDK folder from the project directory. Otherwise, there will be problems when launching SDK. We will rebuild the SDK project later.

7. Generate bit stream. Export to SDK, be sure to include the bitstream.

8. Going back to Vivado, make sure your device is connected and click **Program Device** under **Program and Debug > Hardware Manager**



Then, the new perspective of windows will be shown automatically as below:



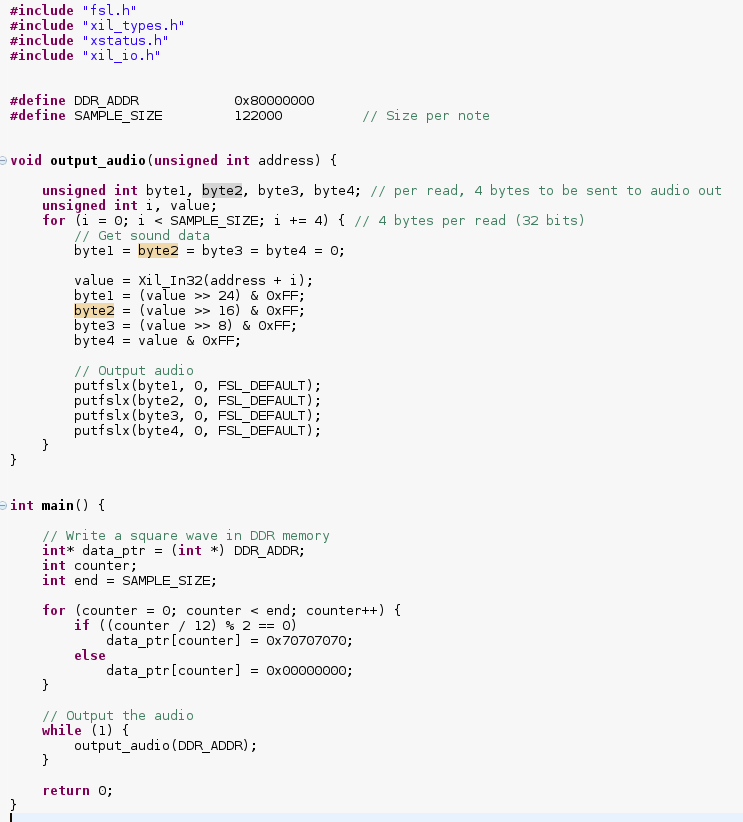
**SDK Configuration**

The SDK configuration is similar to the previous audio tutorial, except for a few changes.

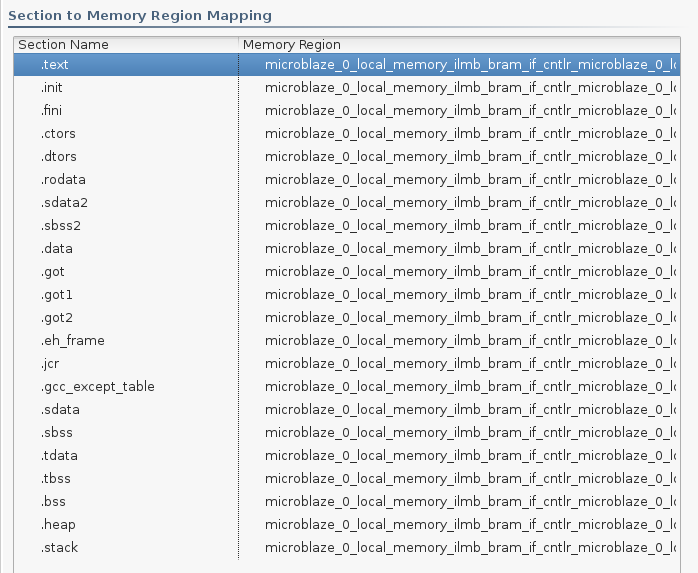
1. Open SDK and create a new project (Xilinx Application). Make it an empty application.

2. Right click the new project to create a new C source file, give it a name main.c.

3. Copy and paste the provided main.c into the project.

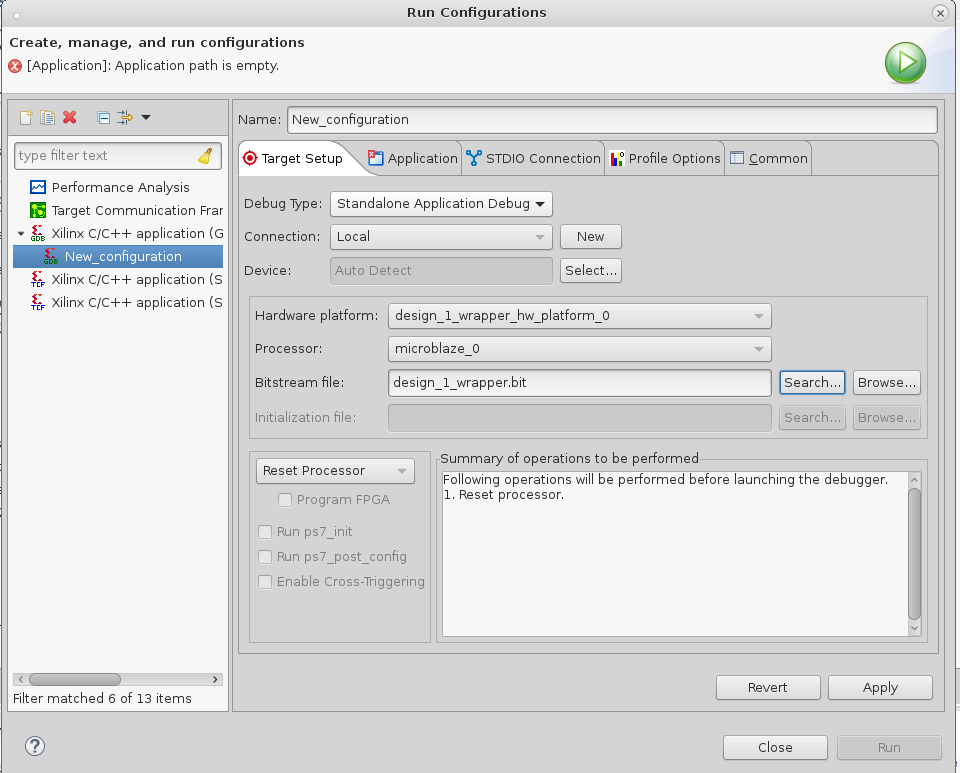


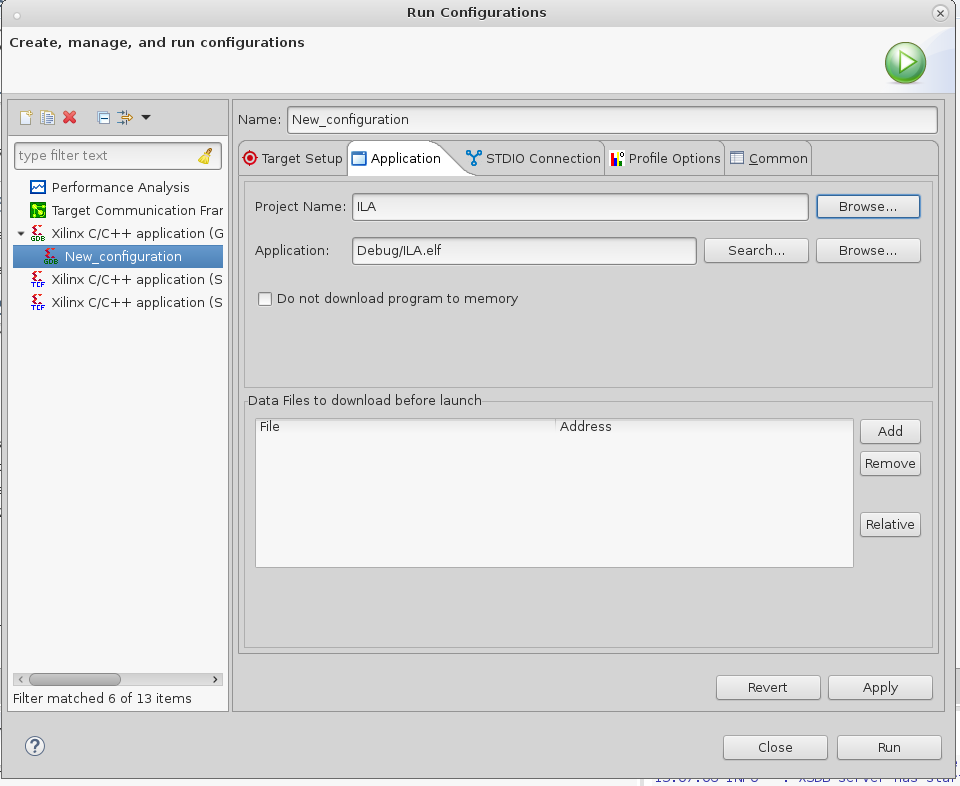
4. Open the lscript.ld (linker script) under src. Change all Memory Region Mapping to local memory as shown.



5. Save and build the project.

6. Run configuration. Create a new Xilinx C/C++ application (GDB) configuration. On the Target Setup page, choose the wrapper as Bitstream file. On the Application page, select this project. **Note: Do not Program FPGA in SDK, since the FPGA should have already been programmed by using the Hardware Manager - Program Device in Vivado. Program the FPGA again in SDK might affect the functionality of the ILA module.**





7. Click Run. If you have connected your headphone to the audio jack, you should hear the same sound as produced from the previous audio tutorial.

**Using ILA**

There are multiple ways to start the ILA. In this tutorial, we will give only two examples, the other methods can be explored from the Xilinx user manual.

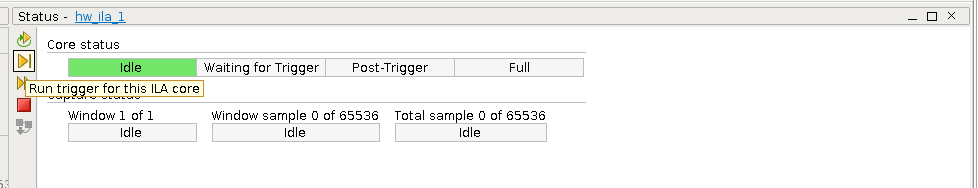
1. Make sure your SW[0] is low and the Trigger Mode is BASIC\_OR\_TRIG\_IN

Trigger Mode of BASIC\_OR\_TRIG\_IN means that the ILA can be evoked by both the trig\_in signal that is connected to SW0 or a combinational logic of the probes.

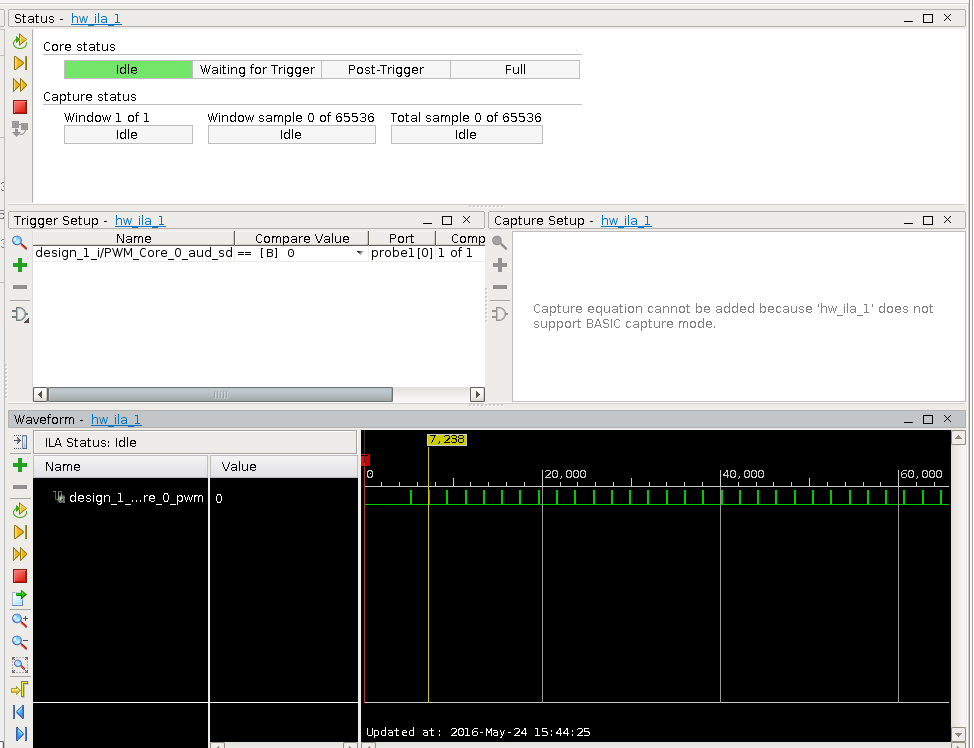
2. Under Trigger Setup, click on the “+” sign and add AUD\_SD. Recall that AUD\_SD is always set to be logic 1 by the PWM IP.

3. Under Trigger Setup -> Compare Value for AUD\_SD, select Operator “==”, Radix [B] and Value logic “0”.

With the above setup, this ILA will only record data when either the SW0 (Trig\_In) is high or AUD\_SD is low. But since AUD\_SD is always high, when we will start the ILA, we should see that the ILA will be in waiting for trigger state.

4. Under Status, click on **Run trigger for this ILA core**. You should see that the Core status should now be Waiting for Trigger and that nothing is generated in the Waveform window.

5. Now turn SW[0] to high. You should see that a sequence of waveform should be recorded and the core status should jump rapidly to Idle. Examine the waveform closely and what do you see?



6. Instead of using SW[0] to trigger the recording, you can also use the PWM output itself. Modify the Trigger Setup such that the ILA will start recording ONLY when PWM output become a logic “1”.