EXOSTIV Dashboard

Hands-on - MICA board

Rev. 1.0.7 - April 17, 2019

Table of Contents

Revision History

Revision	Modifications
1.0.0	• Initial revision
1.0.1	• Minor updates
1.0.2	• Update for EXOSTIV Dashboard v. 1.5.3 • Added section describing the RTL flow.
1.0.3	• Update for EXOSTIV Dashboard v. 1.5.4
1.0.4	• General review and update
1.0.5	• Update for EXOSTIV Dashboard v. 1.9.1
1.0.6	• Legal and brand names update.
1.0.7	• Update with new versions of software

EXOSTIV Dashboard – Hands-on

Introduction

This session provides a documented practical example on how to use EXOSTIV Dashboard software for Xilinx FPGA.

In this session, we review the general flow used by EXOSTIV to instrument FPGA and extract debug data out of it.

Then we show how to use EXOSTIV Analyzer with a pre-configured demonstration board and configuration.

Finally, we describe how to use EXOSTIV Dashboard 'Core Inserter' to configure and insert a simple EXOSTIV IP core into a reference design.

EXOSTIV for Xilinx FPGA – Overview

Typically, the target FPGA design is instrumented with an 'EXOSTIV IP' core after synthesis or from the RTL source code. This IP is configured with the EXOSTIV Dashboard core inserter to reach FPGA internal nodes, sample them and send the sampled data to outside with gigabit transceivers.

Unlike JTAG-based solution, this approach does not require growing the FPGA memory resources with the size of the capture, as the recorded data is progressively extracted towards a memory located outside of the target FPGA. When the average required bandwidth does not exceed what the reserved transceivers are capable of, data can be extracted from the FPGA as a continuous flow or in bursts until the external memory is full.

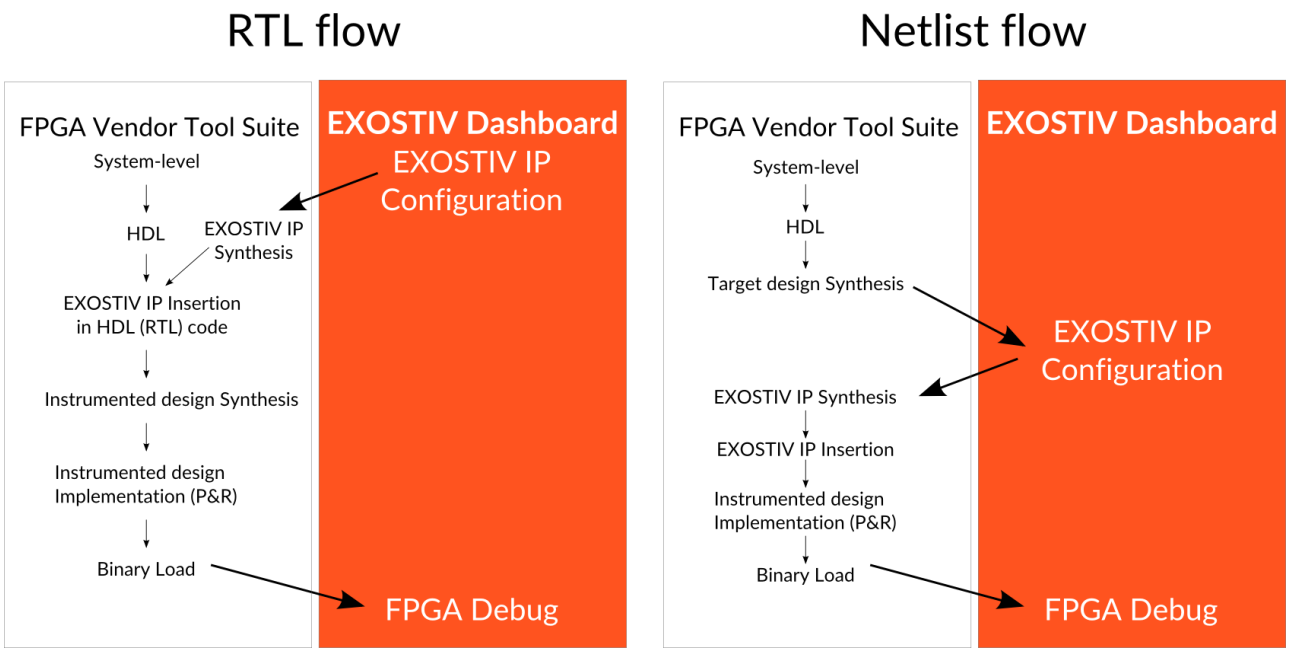


Figure 1: EXOSTIV for Xilinx FPGA – flows overview

2 alternate flows can be used with EXOSTIV: the 'RTL flow', and the 'netlist flow':

- **The 'RTL flow'** (or 'HDL flow') is used to insert the EXOSTIV IP in the RTL (VHDL or Verilog) source code. EXOSTIV Dashboard software is used to set up a **generic IP** provided as an output netlist and a top-level component (VHDL) or module (Verilog). This output IP has to be instantiated 'manually' in the source RTL code – after which the synthesis and implementation of the design are to be run.

- **The 'netlist flow'** is used to insert the EXOSTIV IP into a synthesized target design (= in the target design netlist). EXOSTIV Dashboard is used to configure the EXOSTIV IP and to insert / connect it in the target design netlist. This flow is more automatically managed from the EXOSTIV Dashboard and does not require any manual insertion. From a flow point of view, it has the advantage to work from a synthesized design – and hence – to save the time needed to synthesize the target FPGA every time a new EXOSTIV IP must be inserted. Once the target design

netlist is instrumented with EXOSTIV IP, it has to be implemented (place, route, bitstream generation). This flow is depicted at at

- **Figure 1**.

The general structure of EXOSTIV is depicted at **Figure 2**. With a large external 8GB memory, EXOSTIV provides up to 200,000 times more visibility than tradition embedded instrumentation solutions.

Figure 3 shows the general structure of the EXOSTIV IP core inserted into the FPGA.

Figure 2 : EXOSTIV

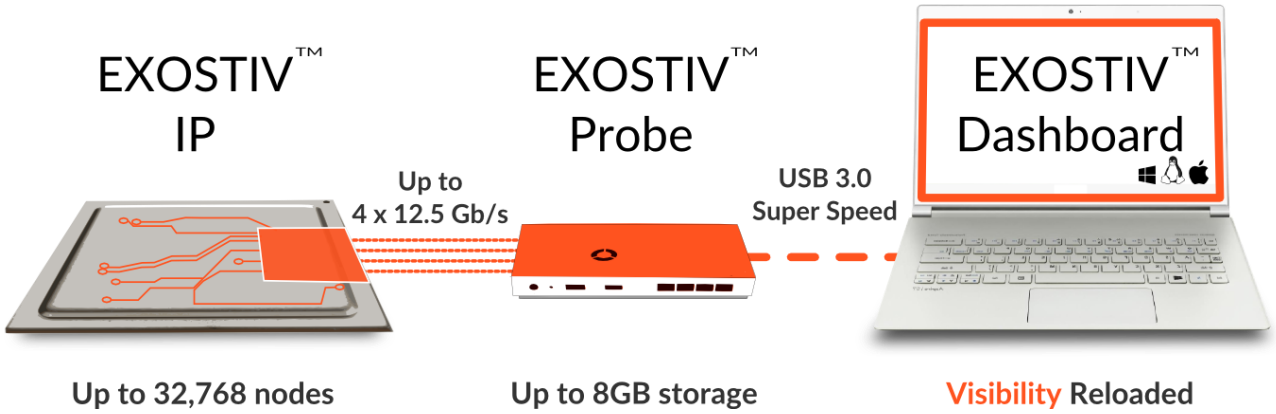
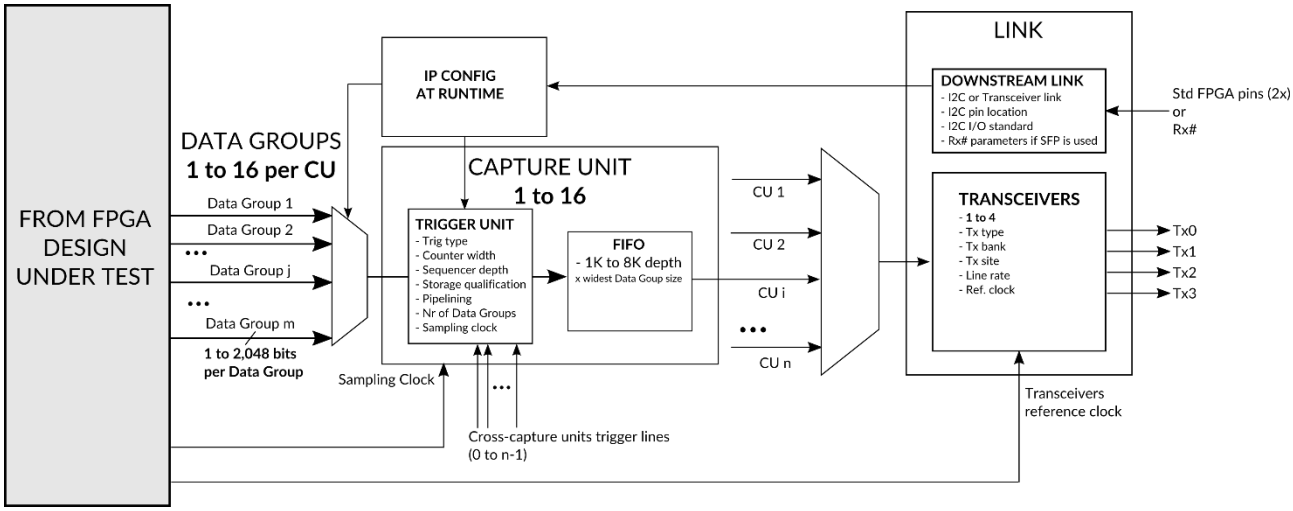


Figure 3 : EXOSTIV IP general structure



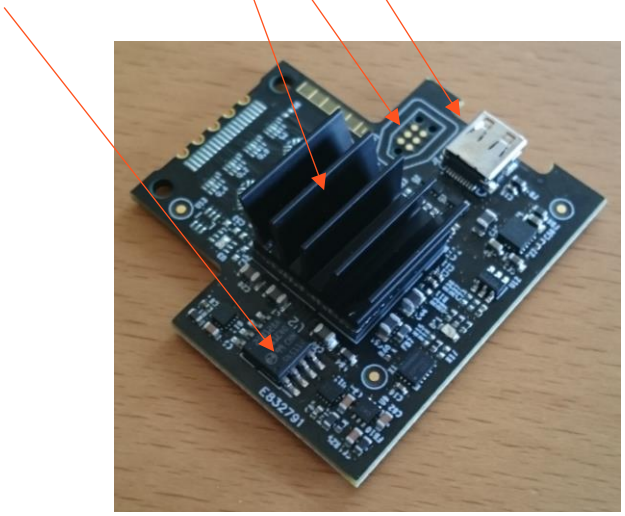
The MICA board setup

Board Overview

The 'MICA' board is a compact target board that can be used to get started with EXOSTIV and demo the system. It is pre-configured with an example FPGA design.

The 'MICA' board is a target board mounted with an Artix-7 FPGA.
Features:

- Artix-7 FPGA – xc7a35tcsg325-2
- Micro-HDMI type connector, with 3 GTP lines for use with EXOSTIV Dashboard. The board power is supplied through this connector by the EXOSTIV probe.
- JTAG interface to reprogram the FPGA and the FPGA configuration EEPROM
- FPGA configuration EEPROM, pre-loaded with an example configuration for demo.



Connecting the board to the EXOSTIV probe

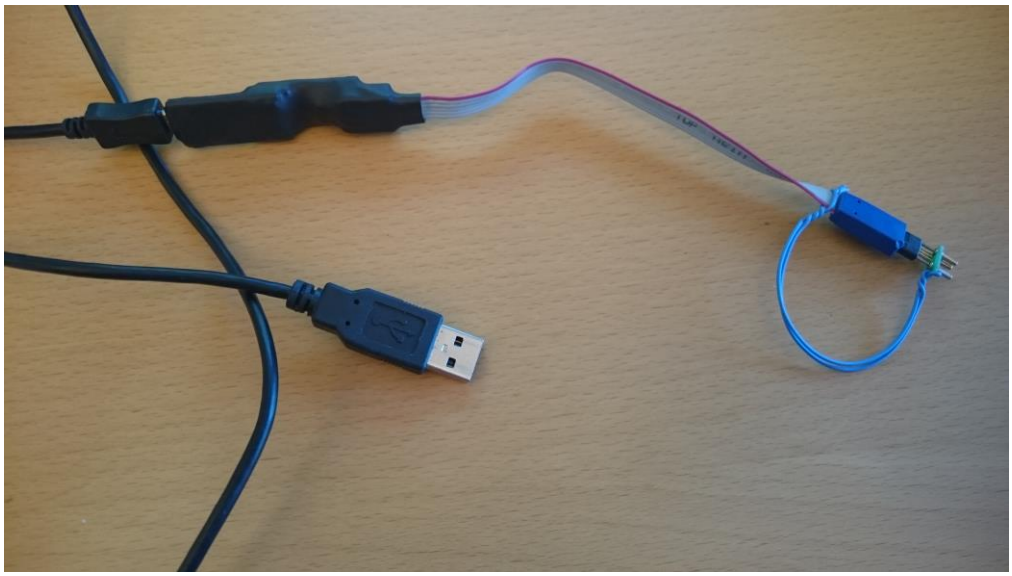
1. Place the HDMI to micro HDMI adapter on one end of the HDMI cable
2. Plug the HDMI end into the EXOSTIV probe HDMI connector and the micro-HDMI end into the MICA board

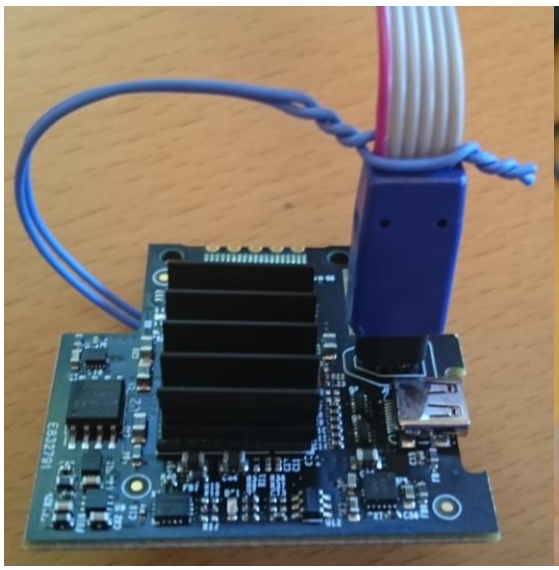
This setup can be used for demonstrating EXOSTIV: up to 3 transceivers are connected to the EXOSTIV probe through the HDMI connector and cables. The MICA board is powered through this cable too. Providing power through the HDMI cable requires using the proper option setting from EXOSTIV Dashboard (see Options menu).

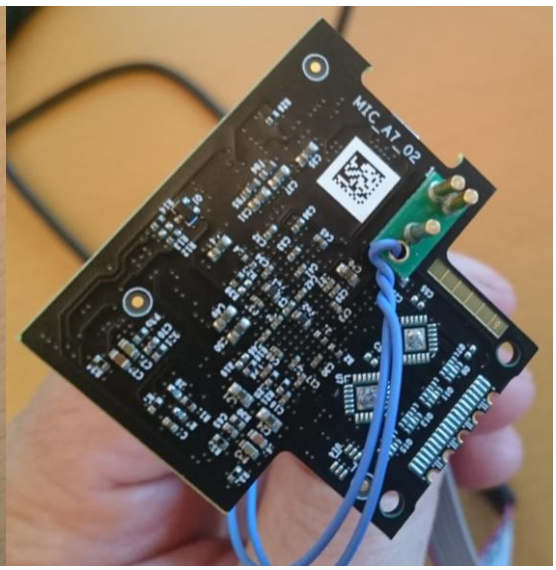
Connecting the configuration cable to the MICA board

The demonstration kit includes a JTAG programming cable adapter for configuring the MICA board FPGA and its configuration EEPROM. **This is only necessary when the FPGA configuration has to be changed.**

It has to be connected to the PC with the provided micro USB cable.
Please check the following article to know how to update the MICA board configuration:
https://www.exostivlabs.com/knowledgebase/how-do-i-update-the-mica-board-configuration/







Quick Start – Run simple captures from the MICA board

This section of the tutorial shows how to use the MICA board setup with its standard configuration for capturing data with EXOSTIV Analyzer. This part loads a predefined project and does not use the EXOSTIV Core Inserter, which is described later in this document, from on page 19.

Demonstration kit contents and files

The demonstration kit includes:

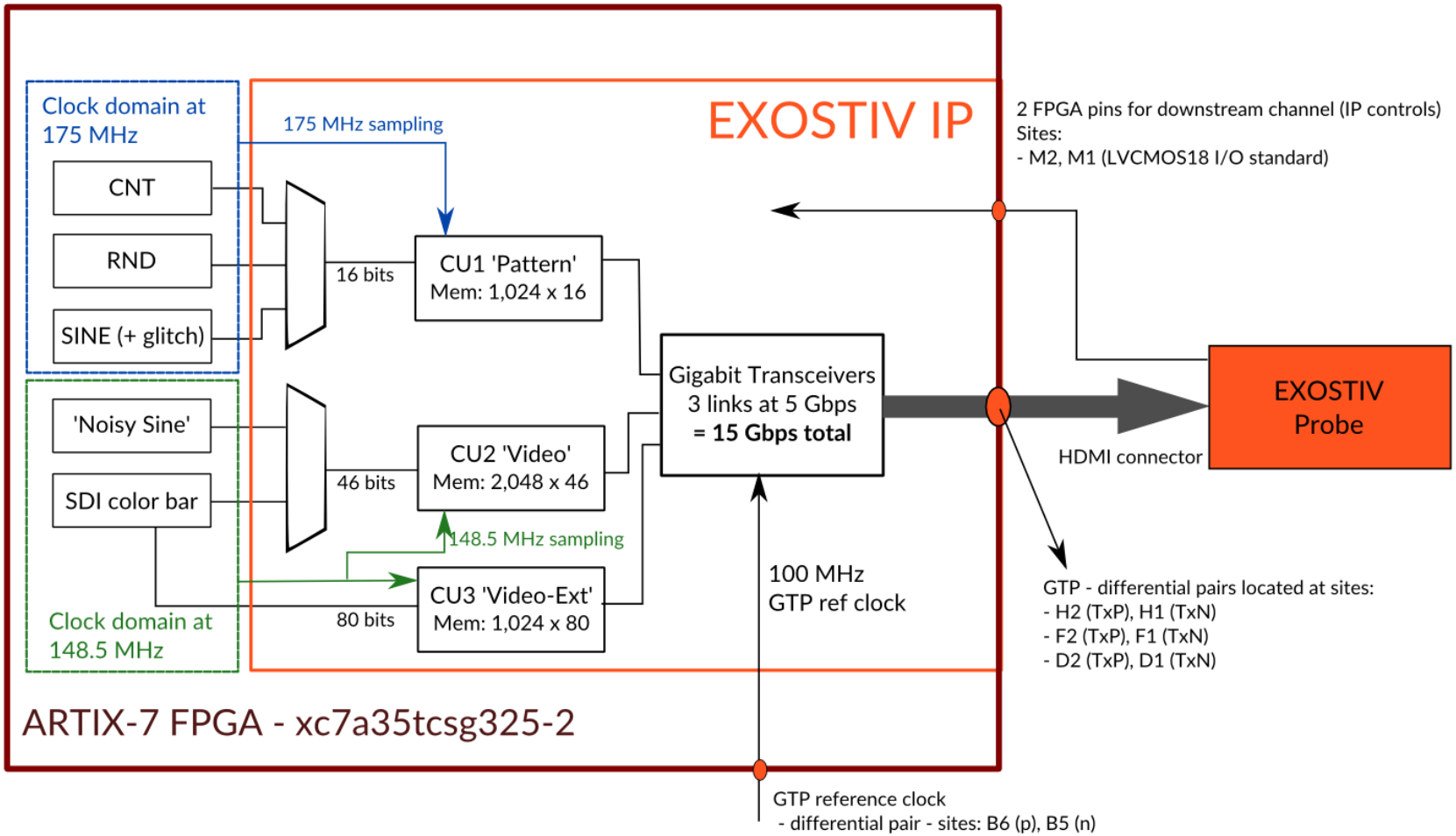
1. 1x EXOSTIV probe with a power supply
2. 1x USB 3.0 cable for the EXOSTIV probe
3. 1x HDMI cable
4. 1x HDMI to mico-HDMI adapter
5. 1x MICA FPGA board
6. 1x programming kit for the MICA board

The following software and files are used with the demonstration kit:

Vivado v.2015.4 or newer (from 2018.2 recommended)	Please go to www.xilinx.com to download the software. The Vivado Webpack (free) is ok for using this demonstration kit. You'll need to register to download the software.
EXOSTIV Dashboard v. 1.9.1 or newer	Please download from this page: https://www.exostivlabs.com/support/downloads/ Please contact support@exostivlabs.com to receive the latest download link and a license key. Please check the 'UG501 - Getting started guide' for installation instructions.
EXOSTIV Dashboard reference project file	Please download from this page: https://www.exostivlabs.com/support/downloads/ File archive : **Demo-MICA-3links_1.9.x.zip** Files: **demo_mica702-3links-1.9.1.epf (or newer):** EXOSTIV Dashboard project file. ***.xml :** EXOSTIV MYRIAD Waveform Viewer wave formatting files. Includes the MICA board binaries : **demo_mica702-3links-1.9.1.bin and demo_mica702-3links-1.9.1.bit** – or newer. The MICA board is pre-configured. Use these files if the board configuration was changed and you want to revert back to the original configuration.
MICA board Artix-7 FPGA reference design	Please download from this page: https://www.exostivlabs.com/support/downloads/ File archive: **DEMO_MIC_A7_02-export.zip**

Overview of the reference design used for the demo instrumented with EXOSTIV IP.

Figure 4 : Overview of the reference design instrumented with EXOSTIV IP.



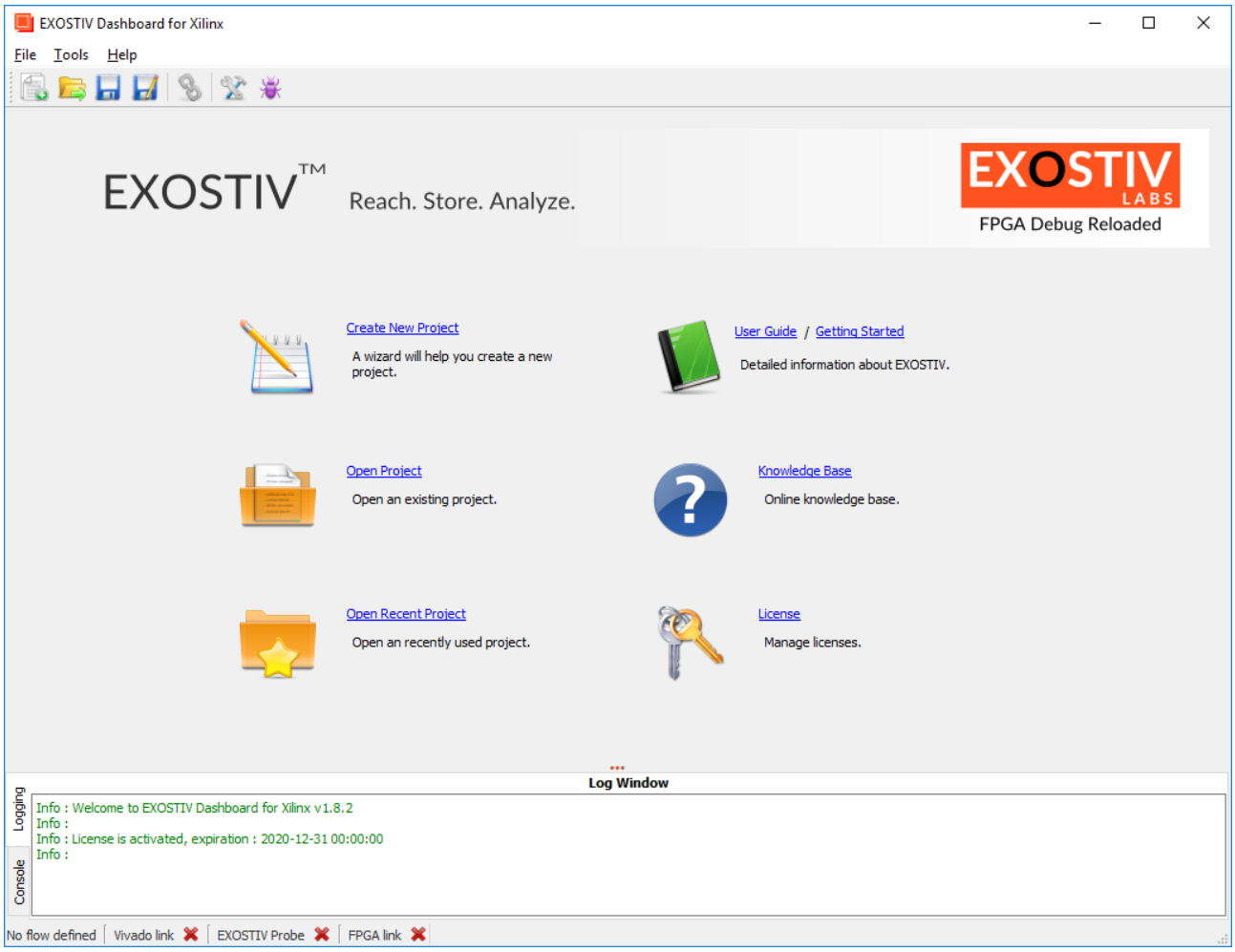
REMARKS:

1) The original design provided in the DEMO_MIC_A7_02-export.zip Vivado project does not include the EXOSTIV IP inserted.
2) The binary files provided as demo_mica702-3links.bit and demo_mica702-3links.bin include the EXOSTIV IP . The MICA board is provided pre-configured with these files.

Start EXOSTIV Dashboard and load the reference project file

All project files used with the EXOSTIV Dashboard contains the settings of one EXOSTIV IP.

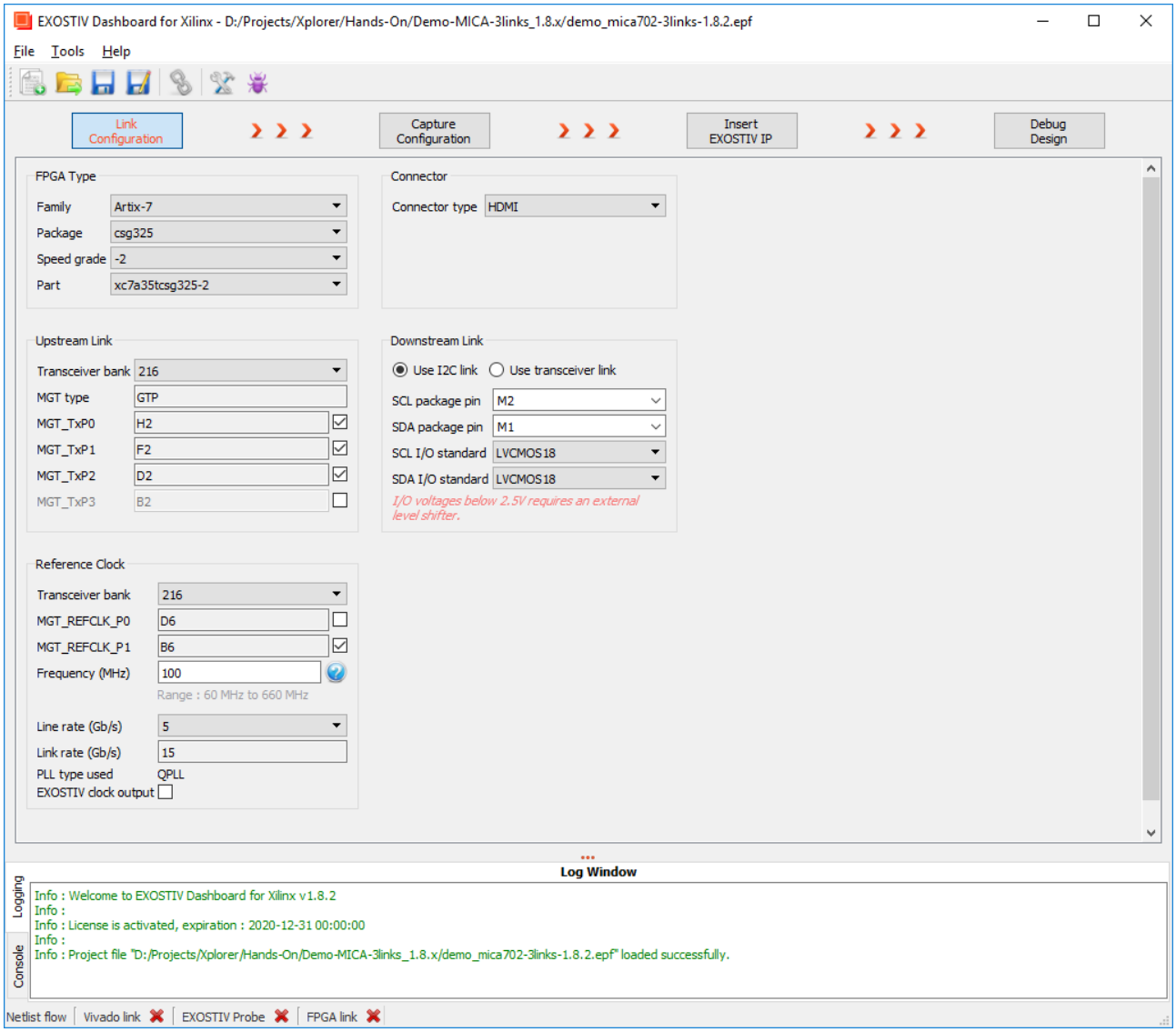
1. Start EXOSTIV Dashboard for Xilinx :



2. On the Welcome Screen, click on 'Open Project'

3. Locate and load '**demo_mica702-3links-1.9.1.epf**'

The Dashboard opens on the Core Inserter 'Link Configuration' window. Please note that this example is based on the 'netlist flow'.



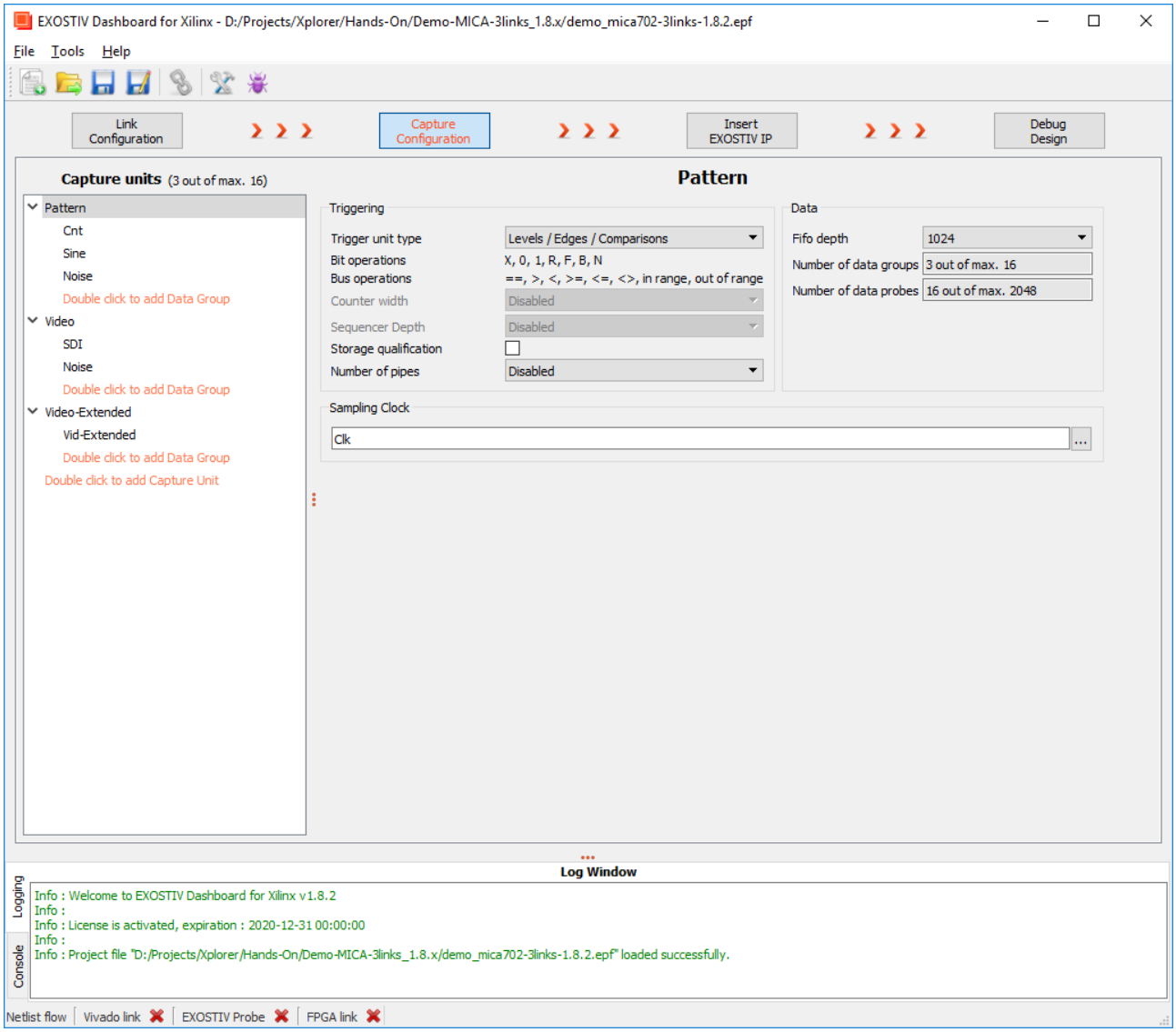
EXOSTIV IP configuration review

The 3 buttons on the top of the window show the flow for configuring EXOSTIV IP, run insertion and then use the EXOSTIV Dashboard analyzer.

The steps required to configure and insert an IP are detailed from section 'Creating a 'netlist flow' project with EXOSTIV ' at page 19 below.

Clicking on the 'Link Configuration' and 'Capture Configuration' buttons switch the display and allow to check the EXOSTIV IP configuration as defined in the demonstration project and loaded into the MICA demonstration board.

The overall settings match the description of Figure 4.

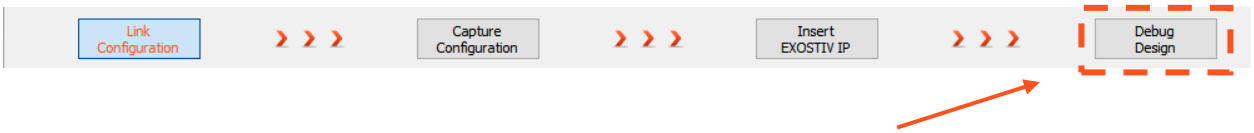

EXOSTIV Dashboard for Xilinx - D:/Projects/Xplorer/Hands-On/Demo-MICA-3links_1.8.x/demo_mica702-3links-1.8.2.epf
File Tools Help
Link Configuration
Capture Configuration
Insert EXOSTIV IP
Debug Design
Capture units (3 out of max. 16)
Pattern
Cnt
Sine
Noise
Double click to add Data Group
Video
SDI
Noise
Double click to add Data Group
Video-Extended
Vid-Extended
Double click to add Data Group
Double click to add Capture Unit
Pattern
Triggering
Trigger unit type
Levels / Edges / Comparisons
Bit operations
X, 0, 1, R, F, B, N
Bus operations
==, >, <, >=, <=, <>, in range, out of range
Counter width
Disabled
Sequencer Depth
Disabled
Storage qualification
Number of pipes
Disabled
Data
Fifo depth
1024
Number of data groups
3 out of max. 16
Number of data probes
16 out of max. 2048
Sampling Clock
Clk
Log Window
Logging
Console
Info : Welcome to EXOSTIV Dashboard for Xilinx v1.8.2
Info :
Info : License is activated, expiration : 2020-12-31 00:00:00
Info :
Info : Project file "D:/Projects/Xplorer/Hands-On/Demo-MICA-3links_1.8.x/demo_mica702-3links-1.8.2.epf" loaded successfully.
Netlist flow
Vivado link
EXOSTIV Probe
FPGA link

Switch to the EXOSTIV Dashboard Analyzer

The EXOSTIV Dashboard 'Analyzer' is used with the EXOSTIV Probe to sample and capture the internal nodes connected to the EXOSTIV IP.

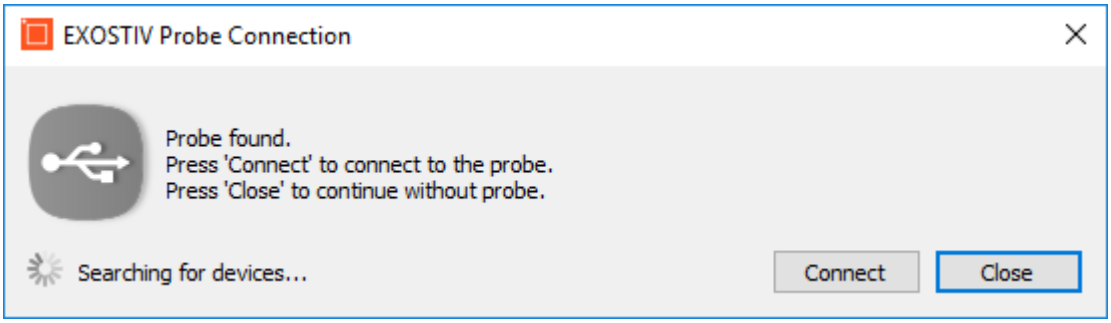
Once you have loaded the target FPGA with the instrumented configuration, access the Analyzer by clicking on the 'bug icon' lets you access the EXOSTIV Analyzer.

Alternatively, the Analyzer can be accessed by clicking on 'Debug Design' in the top tool flow in the Core Inserter.

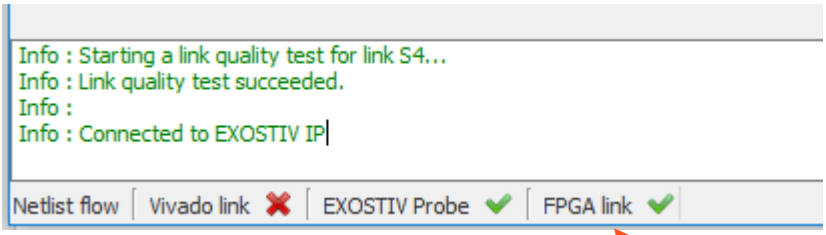


Connect the Probe

1. Connect the MICA board to the EXOSTIV probe
2. Power on the EXOSTIV Probe
3. Click on the 'connect' button in EXOSTIV Dashboard toolbar: 
4. Click on 'Connect'



Once connected to the probe, EXOSTIV automatically attempts to connect the probe to the IP that is in the target FPGA.



EXOSTIV Dashboard is connected to the EXOSTIV Probe through USB

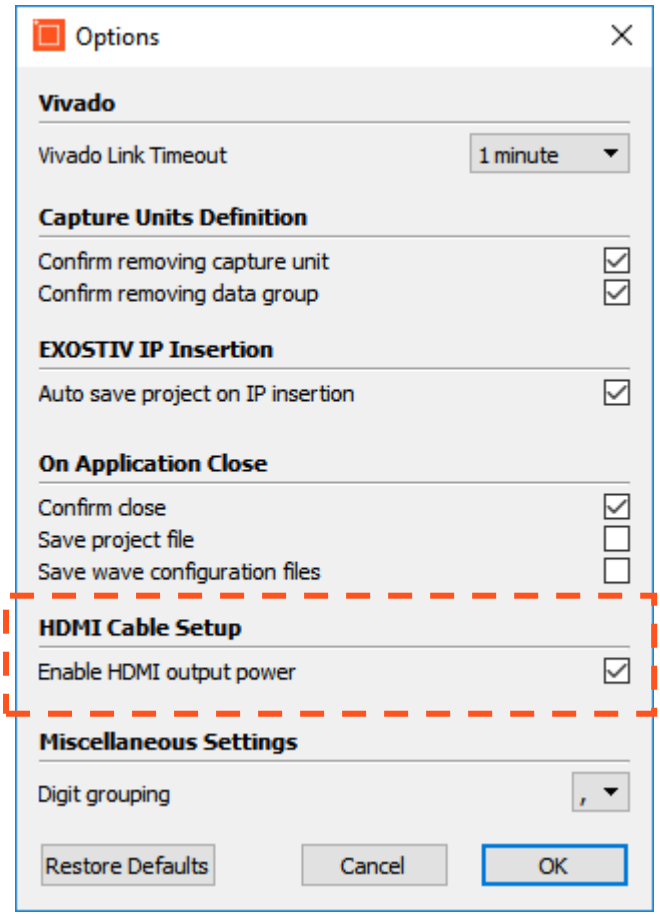
EXOSTIV Dashboard has found a valid EXOSTIV IP in the target design and there is A valid communication with it

Remarks:

For the 'FPGA link' to be established, the EXOSTIV Probe must be able to:

1. properly communicate with an IP inserted in the target FPGA – and:

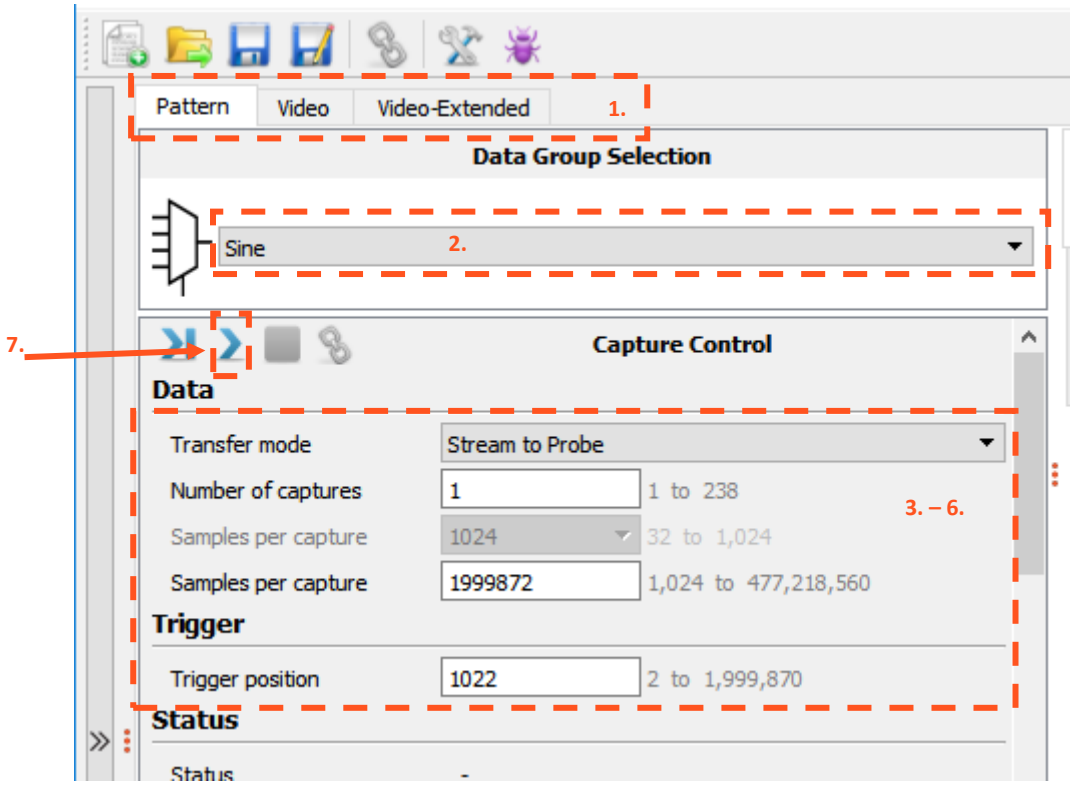
2. check that the EXOSTIV IP inserted in the FPGA matches with the IP settings (CU, data groups, …) as defined in the project file. For instance, if a project file is used, that does not match the EXOSTIV IP inserted in the FPGA, the communication will not be established and the EXOSTIV Dashboard Analyzer will not be usable.
   To check if the EXOSTIV IP inserted in the target FPGA matches with the project settings, a unique identifier (UUID) is programmed in the generated IP and read back with the EXOSTIV Probe. This UUID is also saved in the project file when a new EXOSTIV IP is generated.
3. The MICA board is powered through the HDMI link. The power line of the HDMI link is enabled through the EXOSTIV Dashboard menu: **Tools > Menu**



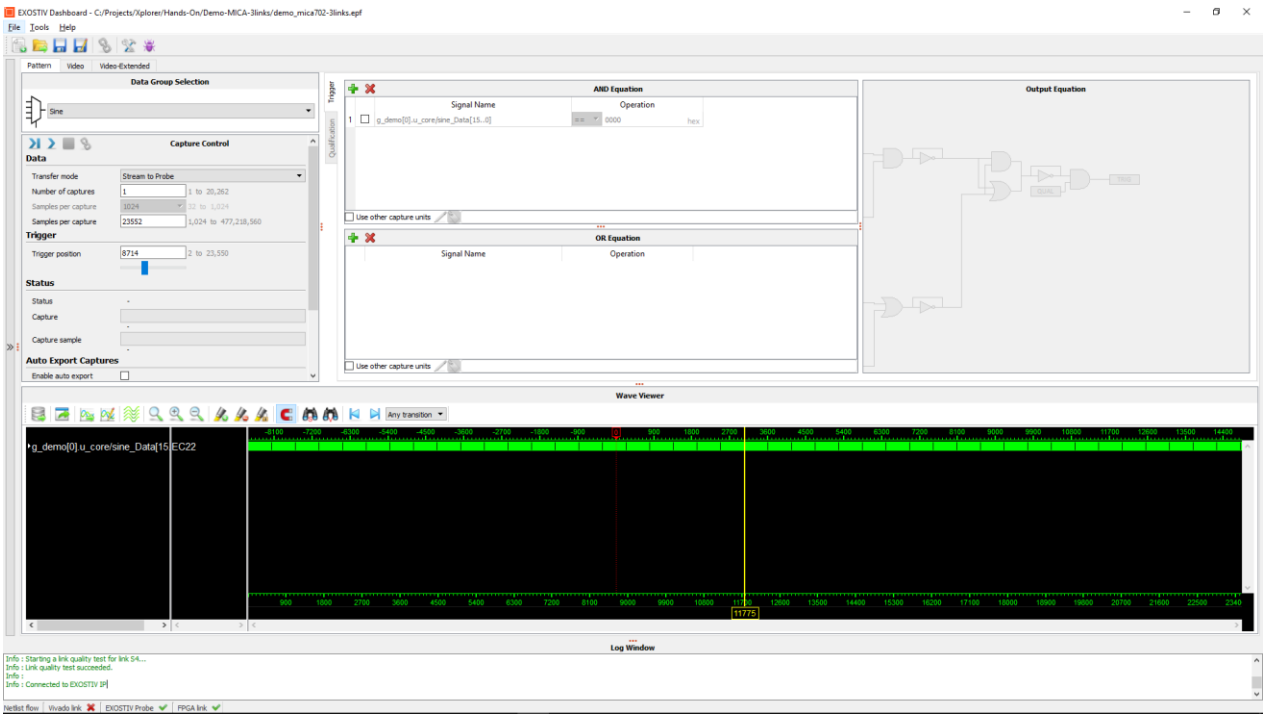
To power on the MICA board, the project file **must** be loaded into the EXOSTIV Dashboard before attempting to connect to the EXOSTIV Probe. When the EXOSTIV Dashboard connects to the EXOSTIV Probe through USB, it first checks in the project settings if the HDMI cable should be used – and if the output power option is selected. If it is the case, the power is enabled before the probe attempts to communicate with the EXOSTIV IP loaded in the target FPGA.

Run a simple capture

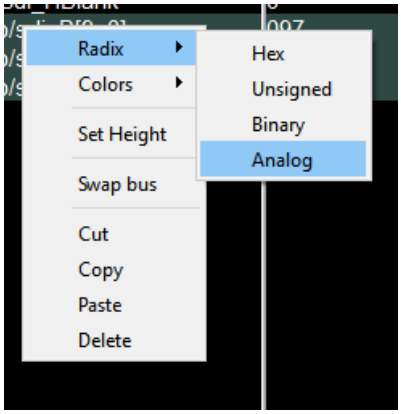
1. Select ‘Pattern’ capture unit tab
2. Select ‘Sine’ Data Group
3. Select ‘Stream to probe’ Transfer mode
4. Specify 1 as number of captures
5. Specify the following number of samples per capture: 1999872
6. Set trigger position to 1022
7. Click on ‘Run immediately’.

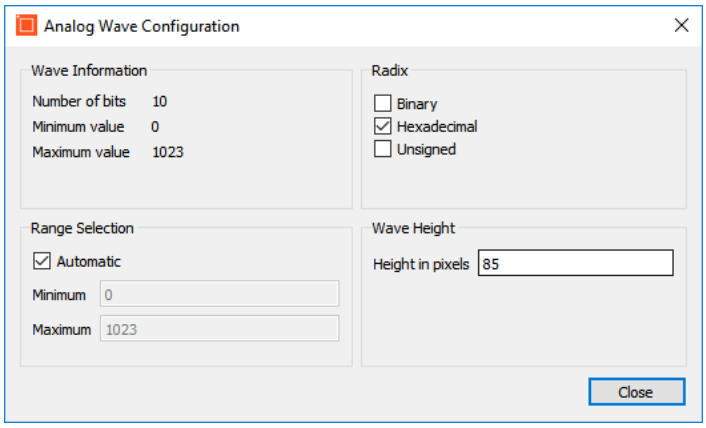


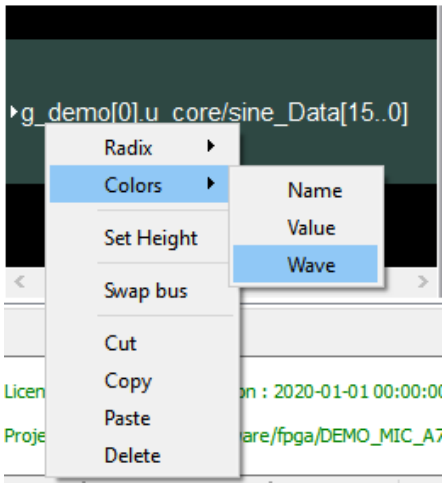
The data is captured, uploaded to the PC and encoded as waves. They appear in the waveform viewer.

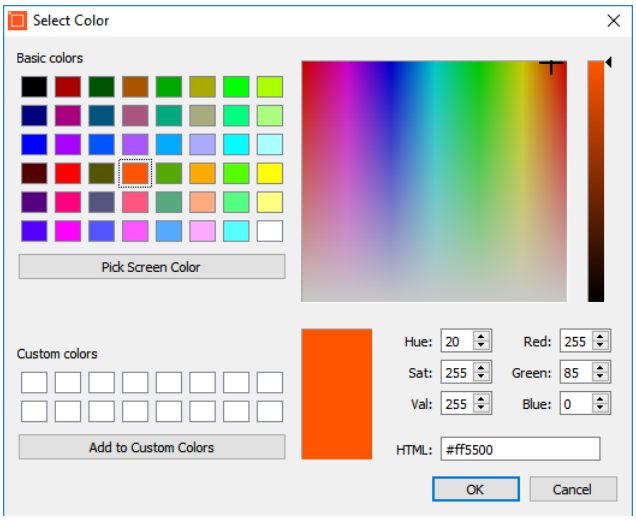


8. Right-click on the sine_data bus in the waveform window and change radix to 'Analog' to display the digital sine wave as analog signal. Then do it again and select Color > Wave to change the wave color. This is a simple example of how the wave can be formatted.

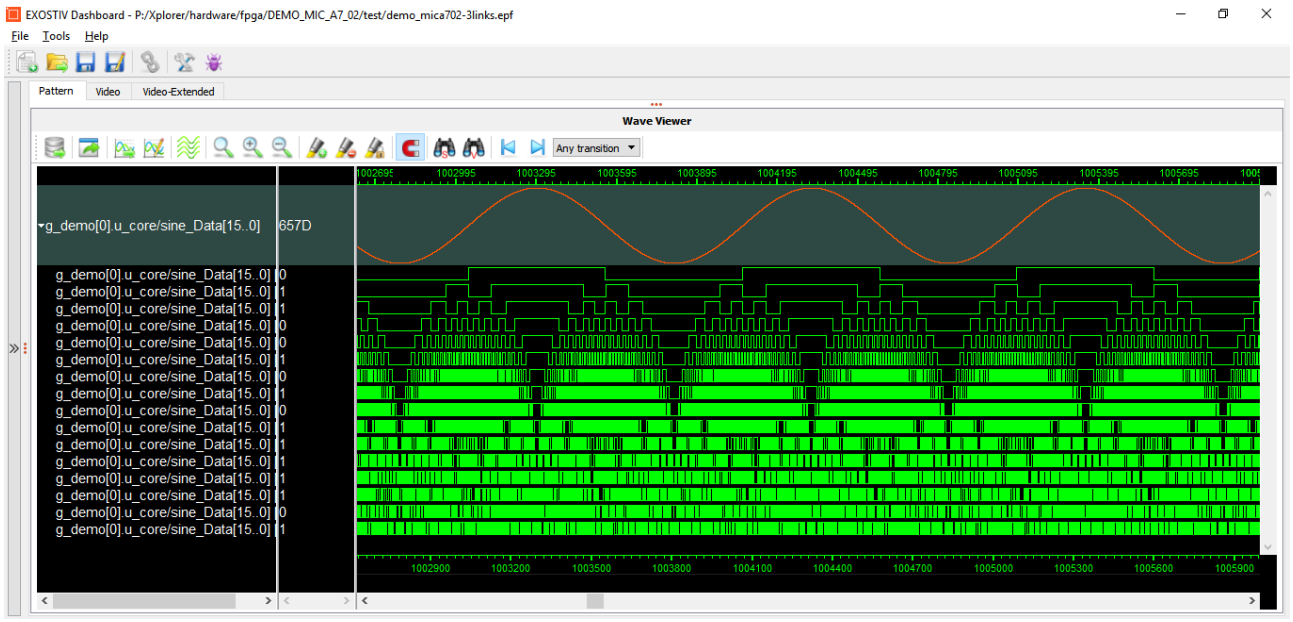






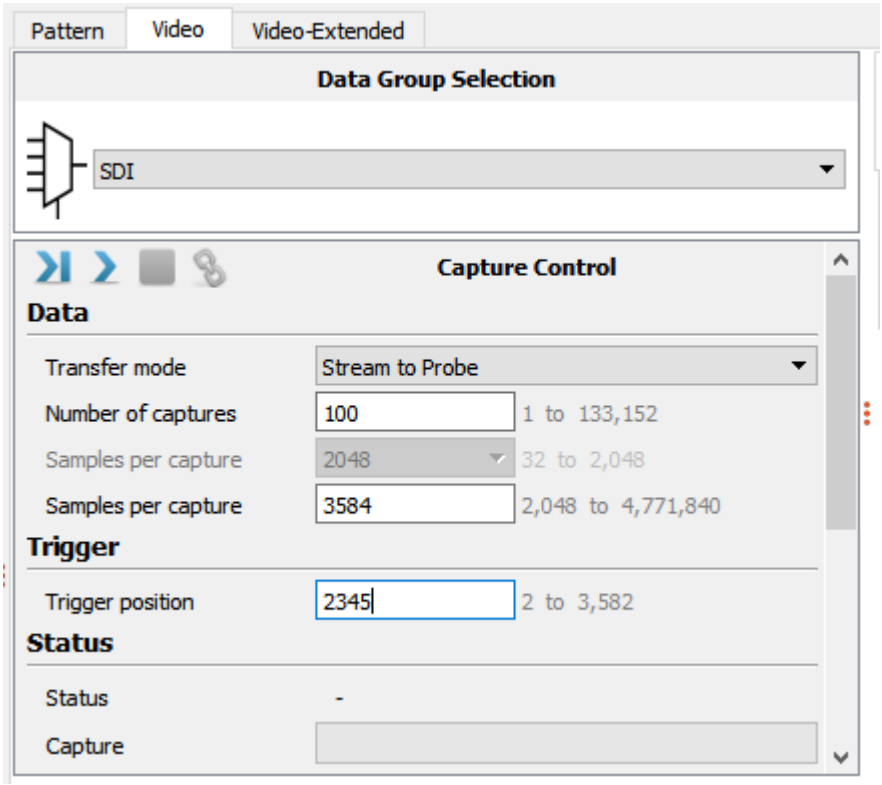


As a result the Sine Wave is displayed as analog format, and can be expanded bit by bit as well…

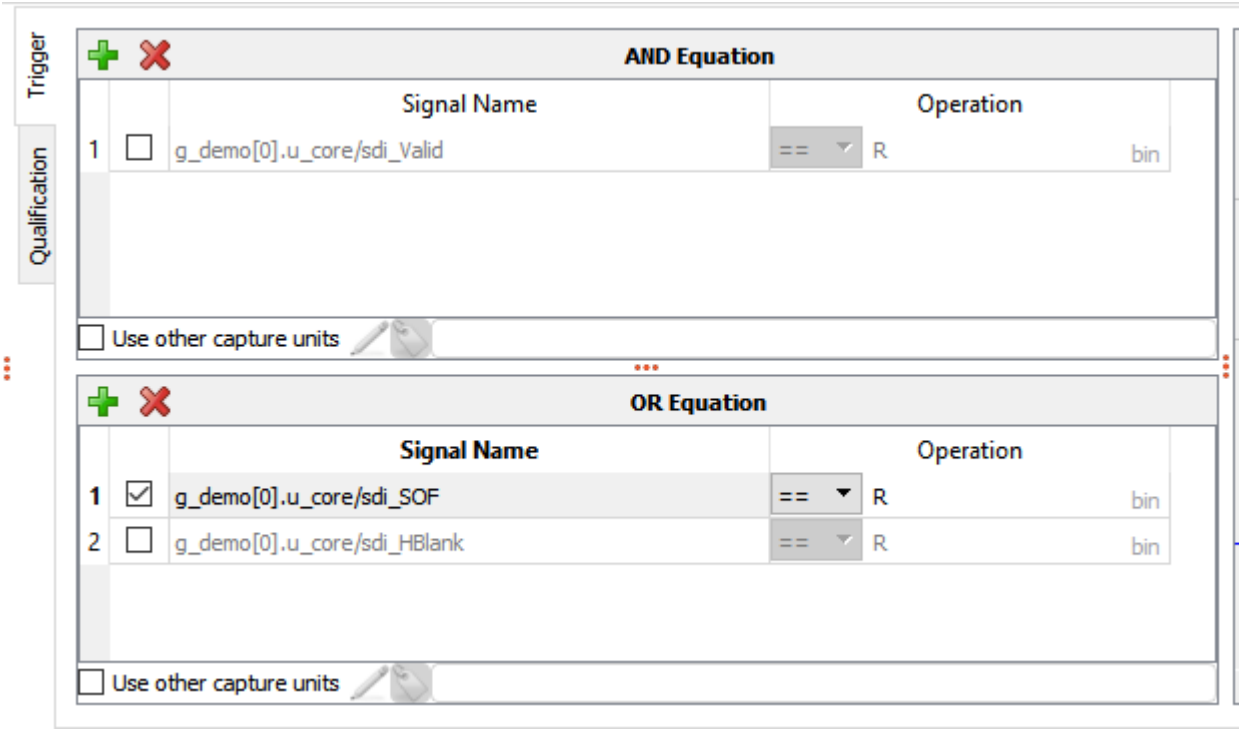


Run a burst capture with trigger

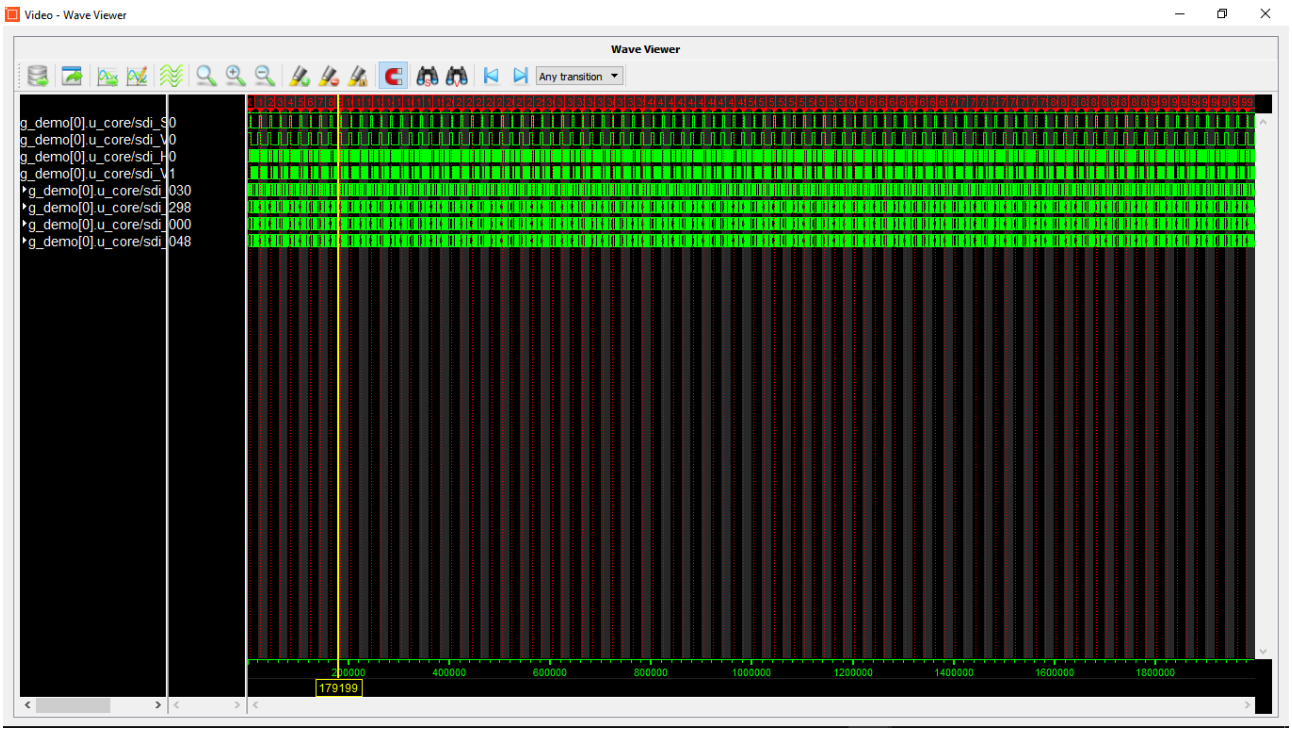
1. Select ‘Video’ tab
2. Select ‘SDI’ data group
3. Set up a capture with e.g. 100 captures of 3584 samples
4. Position trigger at sample 2345



5. Go to the ‘trigger controls’ select the following pre-saved trigger equation:

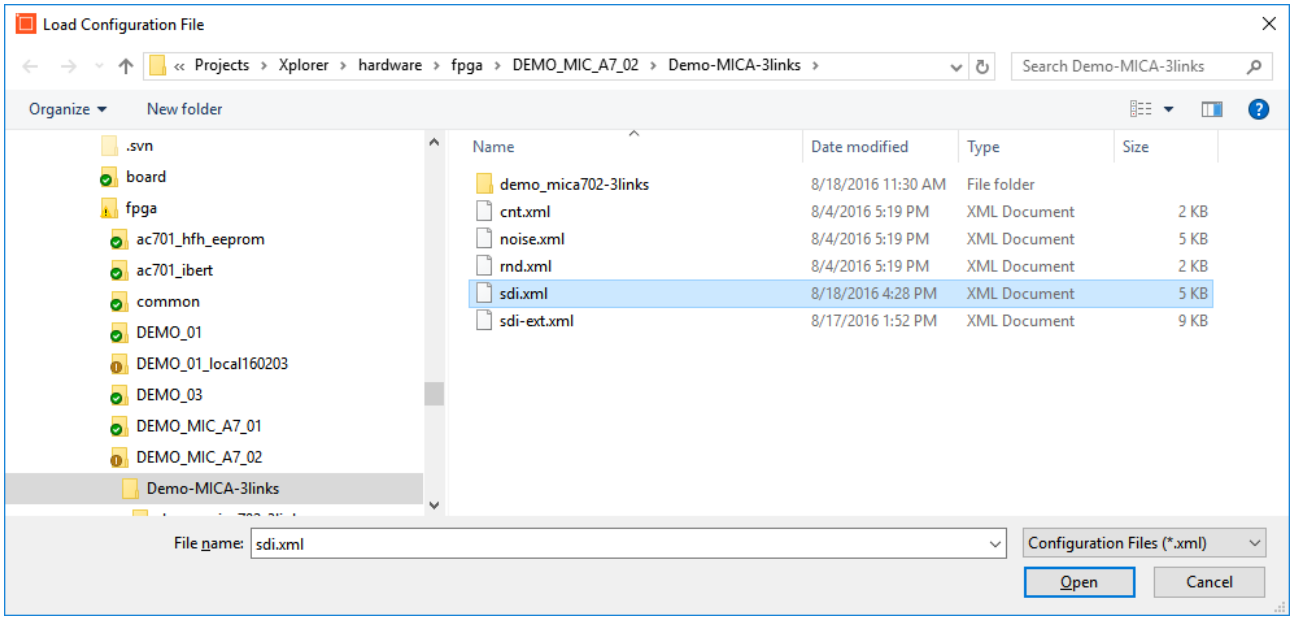


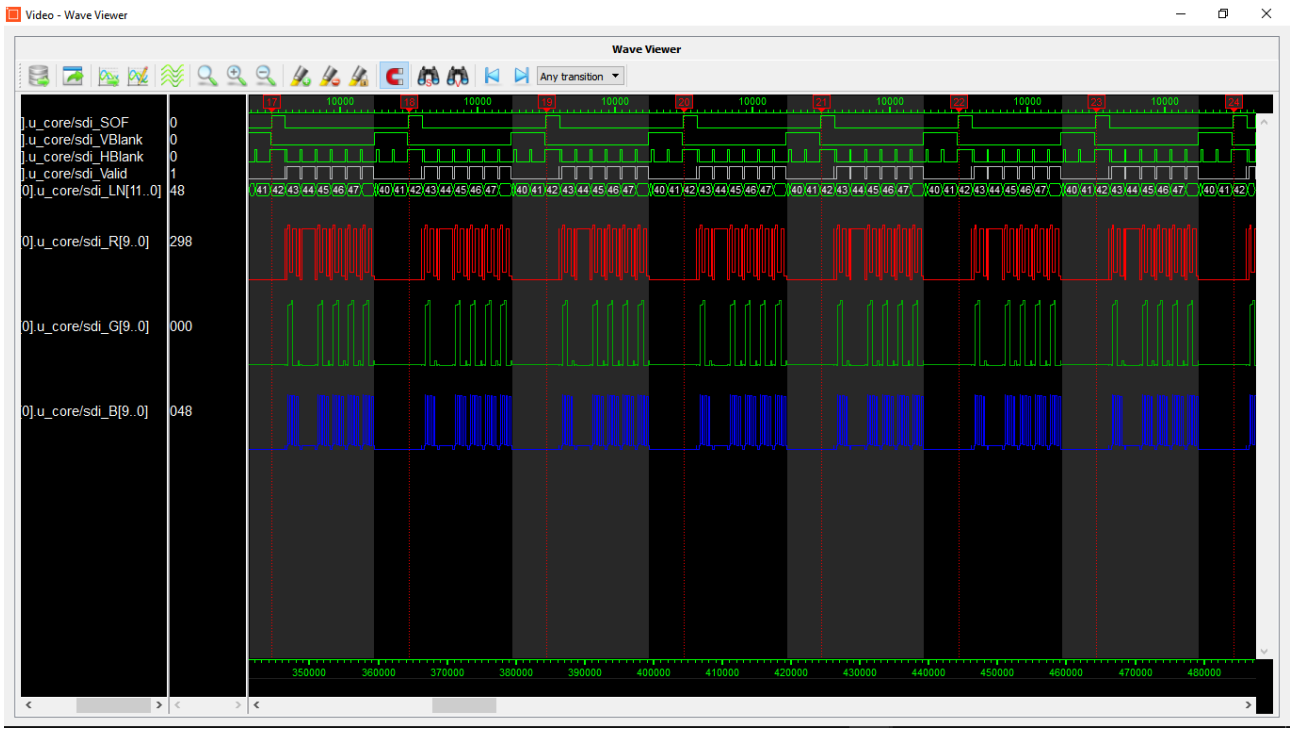
6. Run the capture with trigger by clicking on the '>|' button:



7. To change the wave formatting, click on the 'load wave format' button :

8. Locate an select file 'sdi.xml', open, then zoom…


Load Configuration File
Projects › Xplorer › hardware › fpga › DEMO_MIC_A7_02 › Demo-MICA-3links ›
Search Demo-MICA-3links
Organize
New folder
.svn
board
fpga
ac701_hfh_eeprom
ac701_ibert
common
DEMO_01
DEMO_01_local160203
DEMO_03
DEMO_MIC_A7_01
DEMO_MIC_A7_02
Demo-MICA-3links
Name
Date modified
Type
Size
demo_mica702-3links 8/18/2016 11:30 AM File folder
cnt.xml 8/4/2016 5:19 PM XML Document 2 KB
noise.xml 8/4/2016 5:19 PM XML Document 5 KB
rnd.xml 8/4/2016 5:19 PM XML Document 2 KB
sdi.xml 8/18/2016 4:28 PM XML Document 5 KB
sdi-ext.xml 8/17/2016 1:52 PM XML Document 9 KB
File name: sdi.xml
Configuration Files (*.xml)
Open
Cancel


Video - Wave Viewer
Wave Viewer
Any transition
].u_core/sdi_SOF
].u_core/sdi_VBlank
].u_core/sdi_HBlank
].u_core/sdi_Valid
0].u_core/sdi_LN[11..0]
0].u_core/sdi_R[9..0]
0].u_core/sdi_G[9..0]
0].u_core/sdi_B[9..0]

Creating a ‘netlist flow’ project with EXOSTIV Dashboard

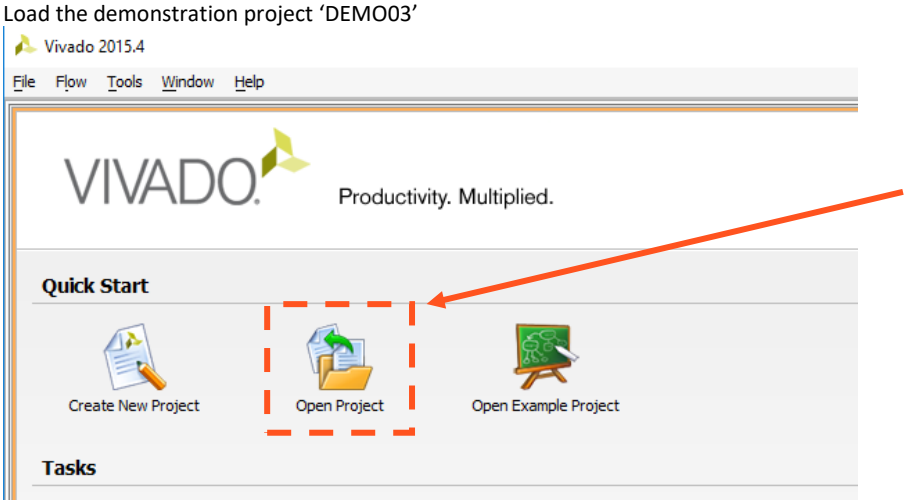
In this section, we’ll describe how to create a ‘netlist flow’ project from zero using EXOSTIV Dashboard.
It shows how to configure an IP and insert it into the target design with EXOSITV Dashboard Core Inserter.

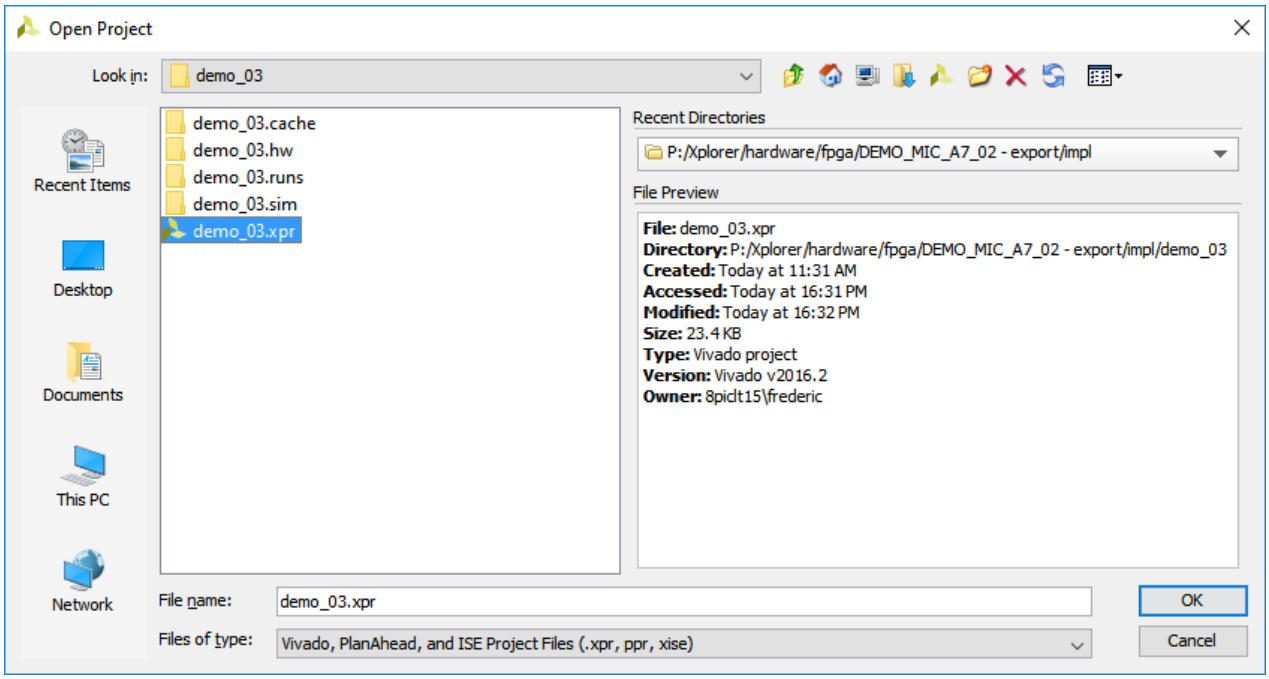
In this section, we’ll use the MICA board reference design, Xilinx Vivado and the EXOSTIV Dashboard.

Start Vivado and create a new netlist flow project

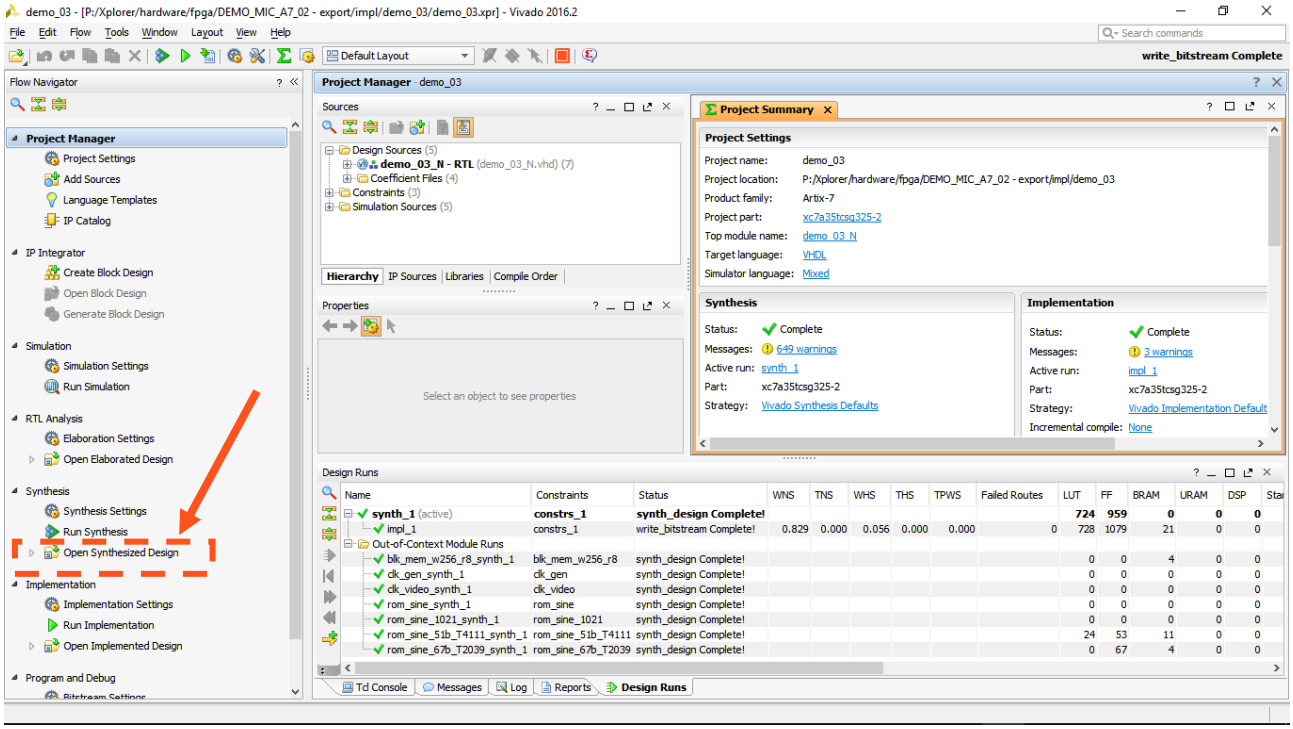
Xilinx Vivado version 2015.4 or newer must be used. The free Webpack version is sufficient for this demonstration, as the Artix-7 FPGA device mounted on the MICA board is supported with this version.

1. Load the demonstration project ‘DEMO03’

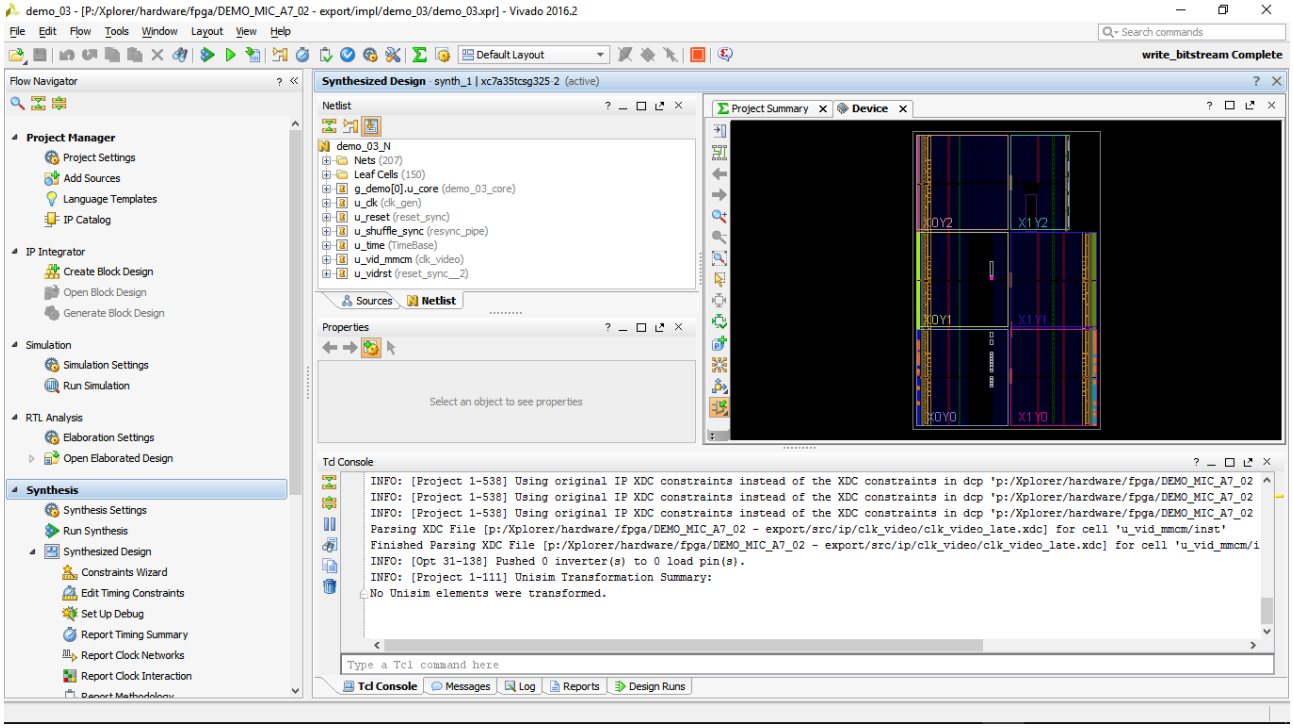




2. In Vivado, open ‘Synthesized Design’

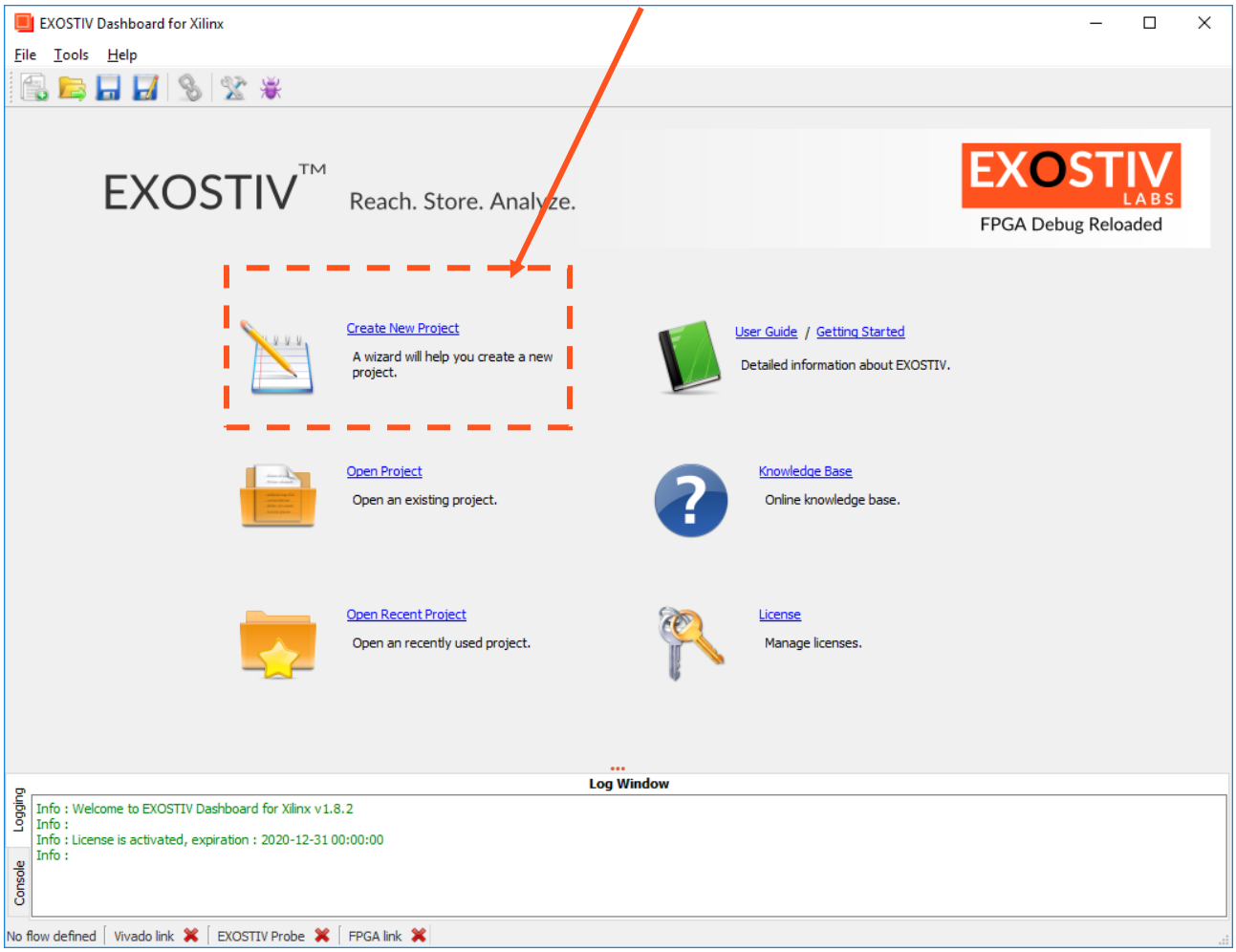


After some time, the synthesized reference design is loaded into Vivado.

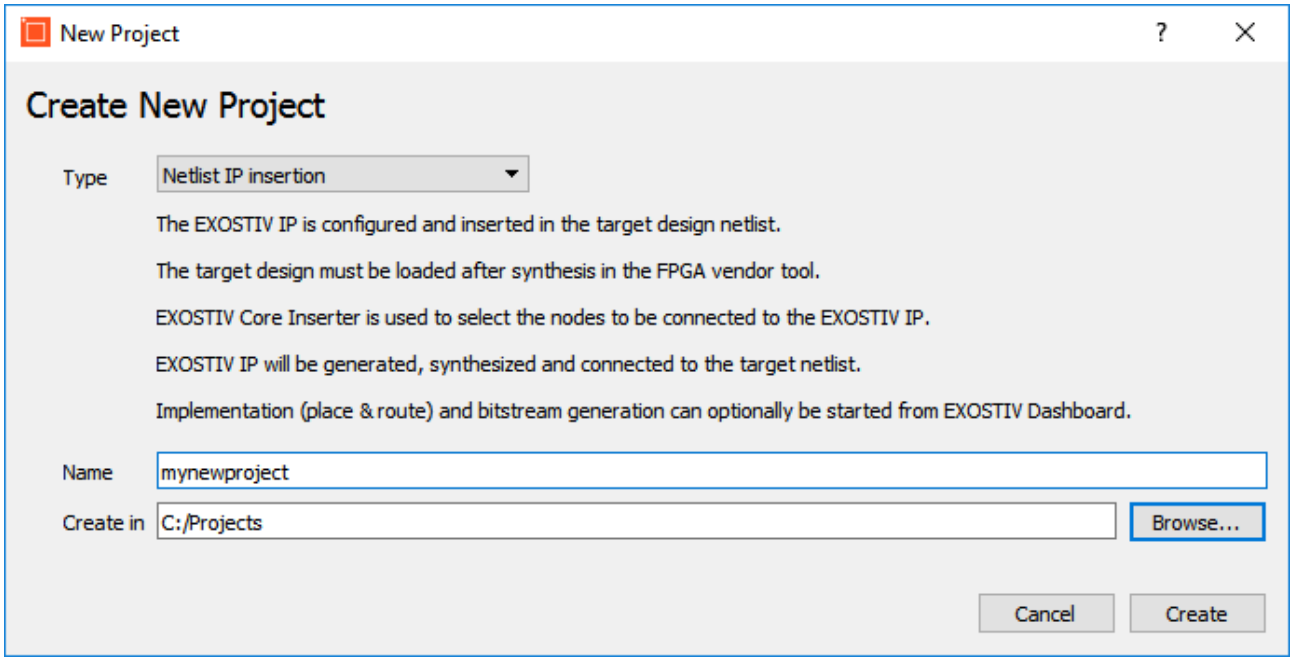


3. Start EXOSTIV Dashboard 

4. In the welcome screen, click on ‘Create New Project’

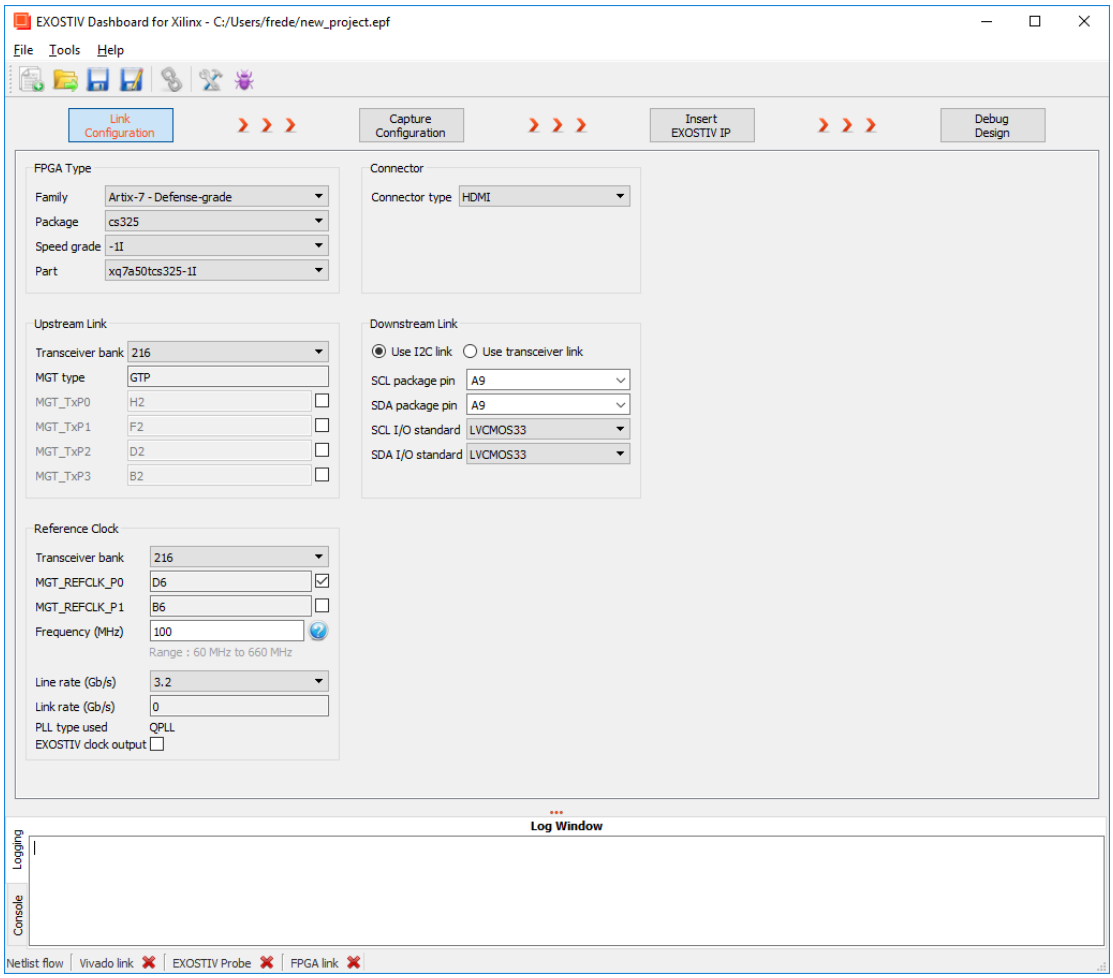


The ‘Create New Project’ window appears, prompting to select the desired flow (Netlist flow or RTL flow)



- Select ‘Netlist IP Insertion’ from the type drop-down box.
- Specify new project name and pick a location.
- For permission issues reasons please do not create project in Program Files (x86) or Program Files directory.

The ‘Link Configuration window’ of the core inserter appears.



Using the Core Inserter

Overview

To access the ‘Core Inserter’, click on the following icon in the main toolbar: 

Inserting the EXOSTIV IP requires 3 successive steps:

- **Step 1 :** Link Configuration
- **Step 2 :** Capture Configuration
- **Step 3 :** Run Insertion

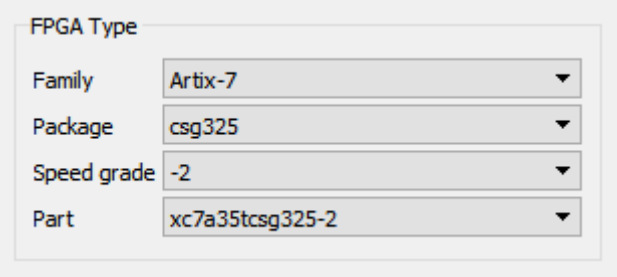
These 3 steps are accessible through the top flow overview in the EXOSTIV Dashboard window

Step 1: Link Configuration

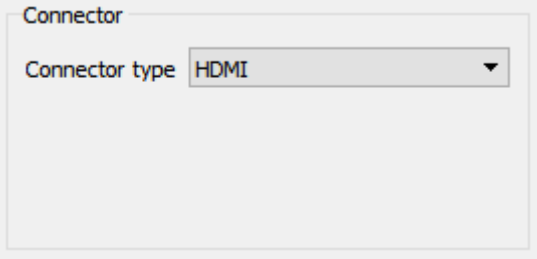
This step defines the characteristics of the target FPGA and of the interface between EXOSTIV IP and EXOSTIV Probe. The information required to complete this step depends on the target FPGA and the target FPGA board. For this hands-on session, we'll use the MICA board from Exostiv Labs. Please refer to the MICA 702 board user's guide for a details (https://www.exostivlabs.com/files/documents/Demo%20Kit%20User's%20Guide%20-%20MICA702.pdf).

Please set up as follows:

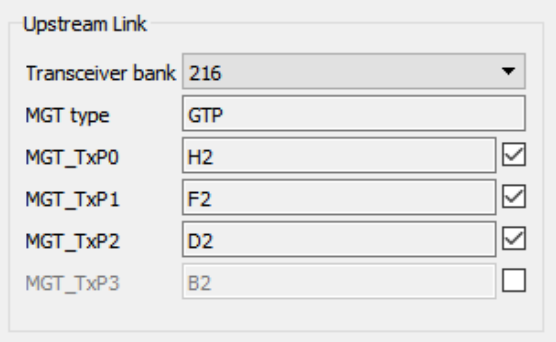
1. **FPGA Type**



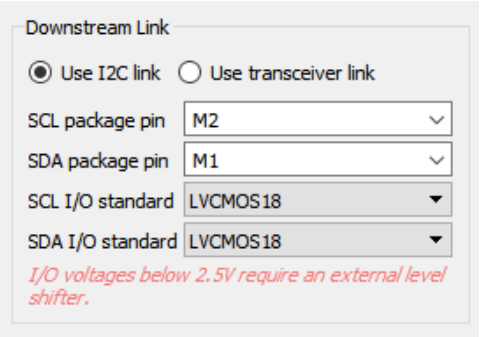
2. **Connector : HDMI, as we'll use the HDMI type of connection between the MICA board and EXOSTIV Probe**



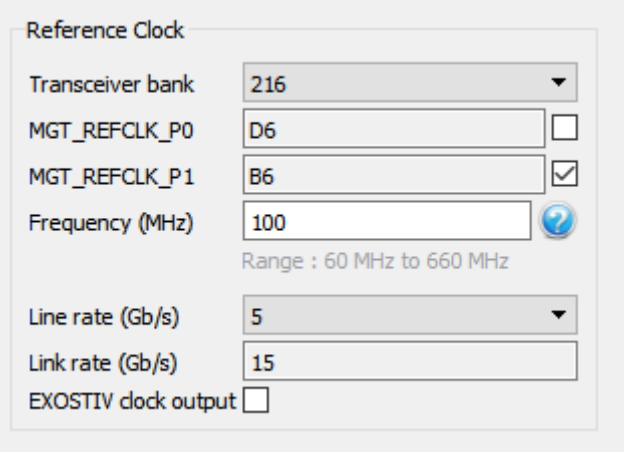
3. **Upstream link:** this setting defines the location of the used gigabit transceivers on the FPGA package. In this case, we'll use the transceivers connected to the micro-HDMI on the MICA board. **They are at sites H2, F2 and D2 of bank 216.**



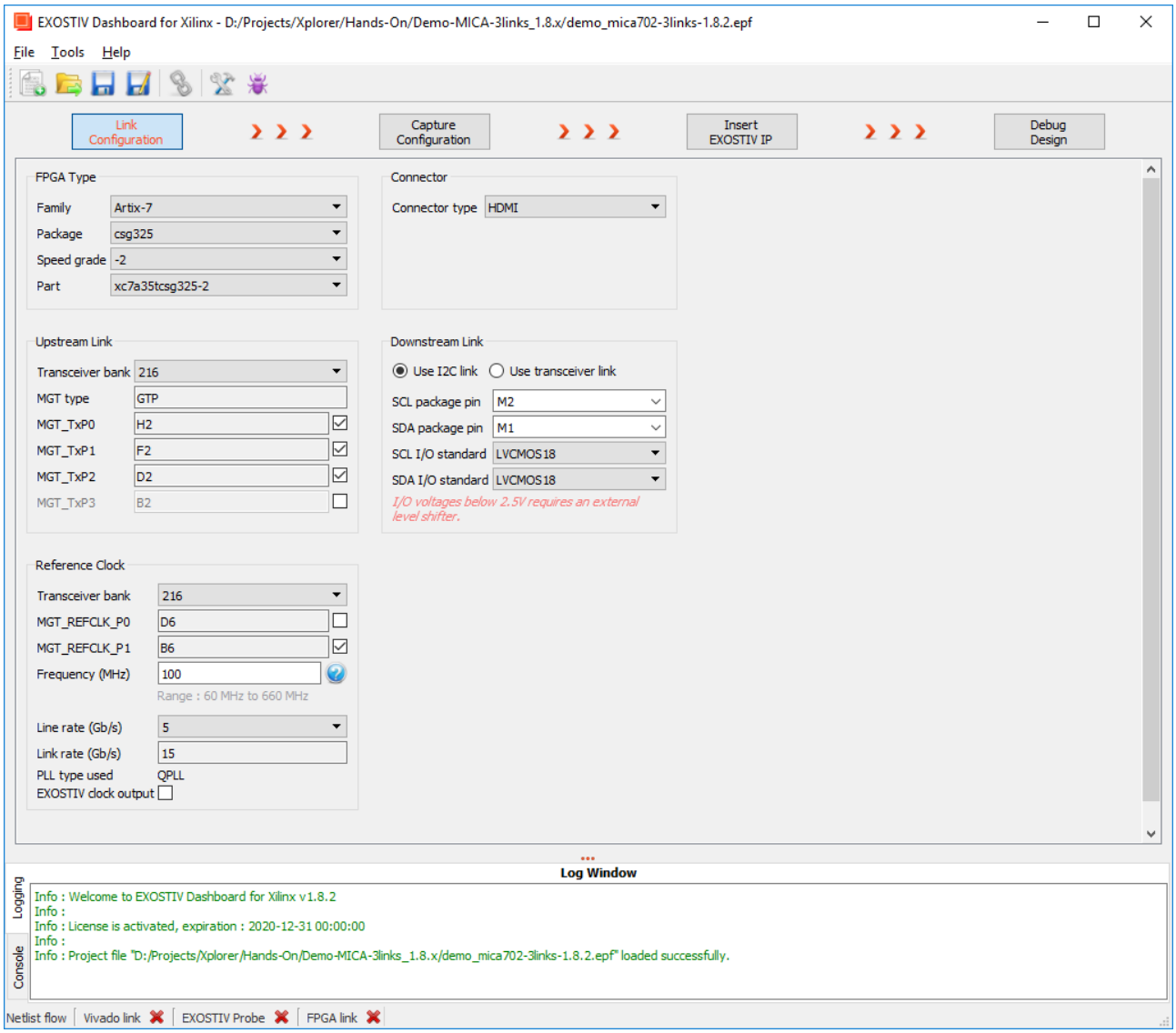
4. **Downstream link:** this setting defines the location of the 2 pins connected to the micro-HDMI connector used for the downstream 'I2C-like' link used to configure EXOSTIV IP at run time. SCL and SDA are respectively at sites M2 and M1 of the Artix-7 FPGA package. The I/O standard is LVCMOS18.



5. **Reference Clock:** this defines the pin input of the reference clock used for the transceiver used for EXOSTIV, as well as its frequency (100 MHz). From the frequency, we can choose the link rate setting. We choose the maximum link rate available for this frequency / FPGA / EXOSTIV Probe model, that is 5 Gbps.



Here is an overview of the 'Link Configuration' for this example:



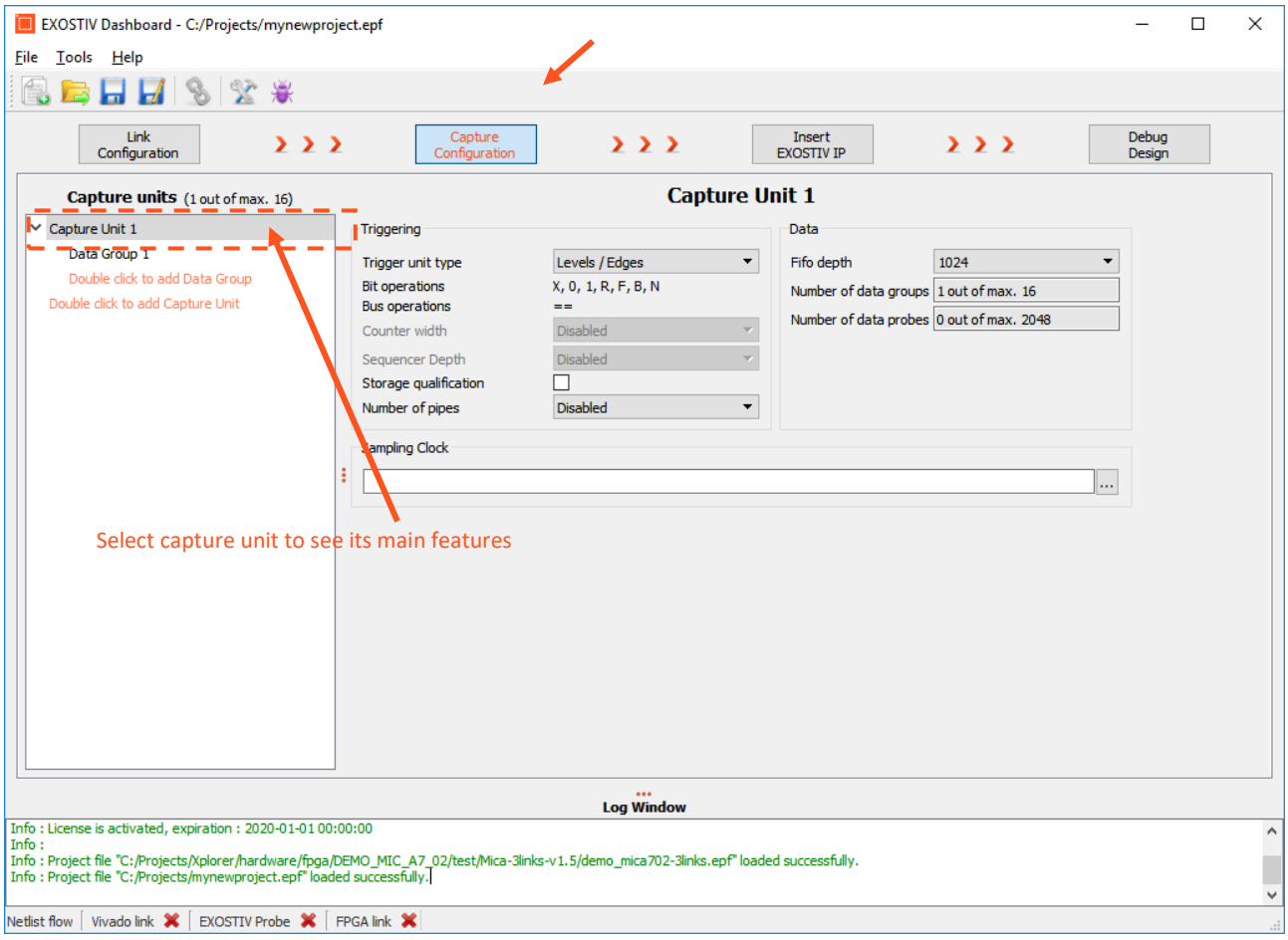
✓ LINK CONFIGURATION: DONE !

To save the project: 

Step 2 : Capture Configuration

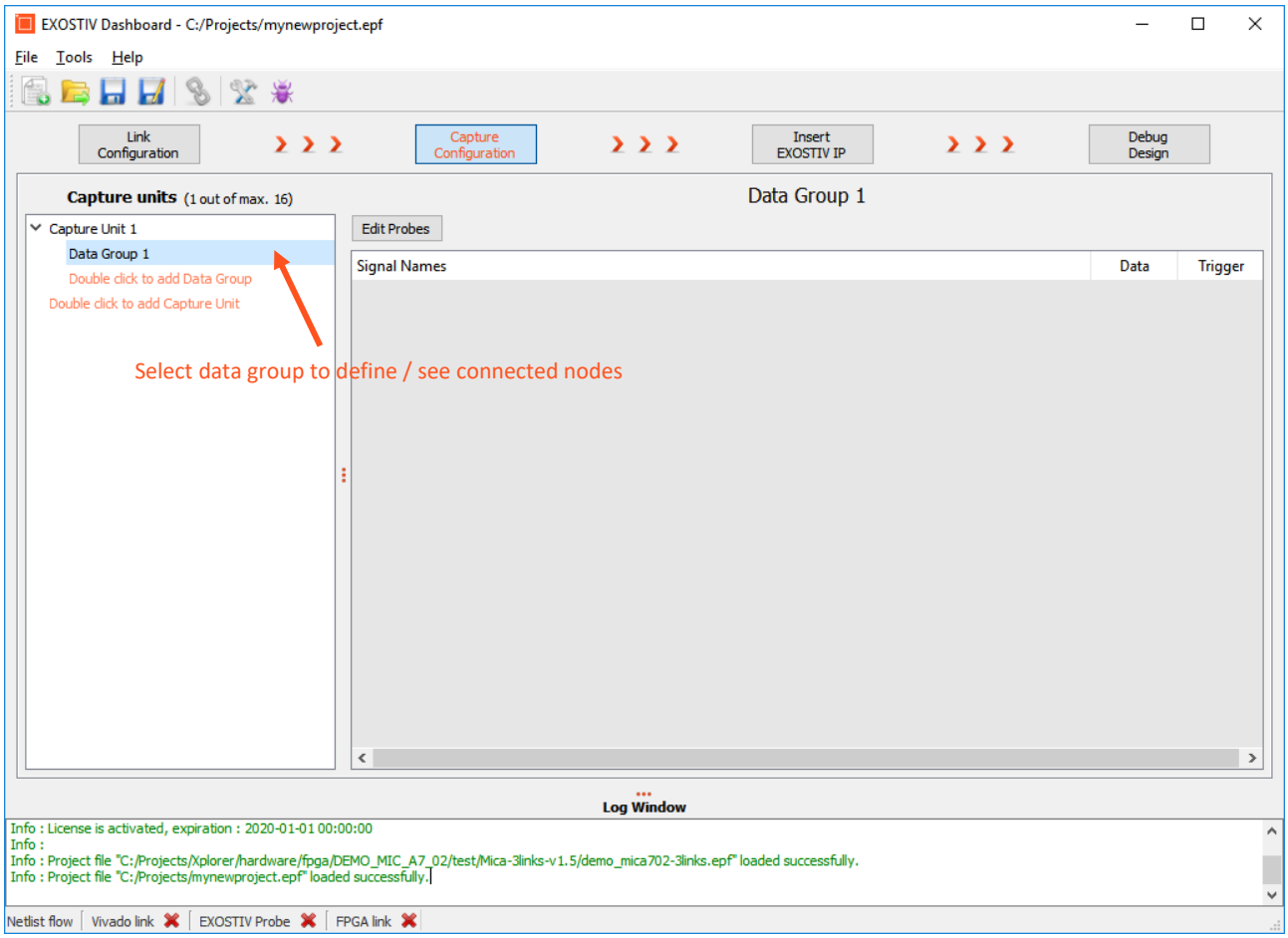
This part of the flow defines the characteristics of the EXOSTIV IP core – namely:

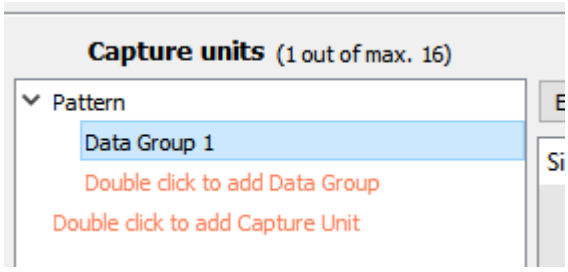
- The capture units and their features:
  - Trigger resources
  - Enable / Disable storage qualification
  - FIFO depth
  - The sampling clock for each capture unit
  - …
- The data groups for each capture units and the signals from the target design that are part of each data group

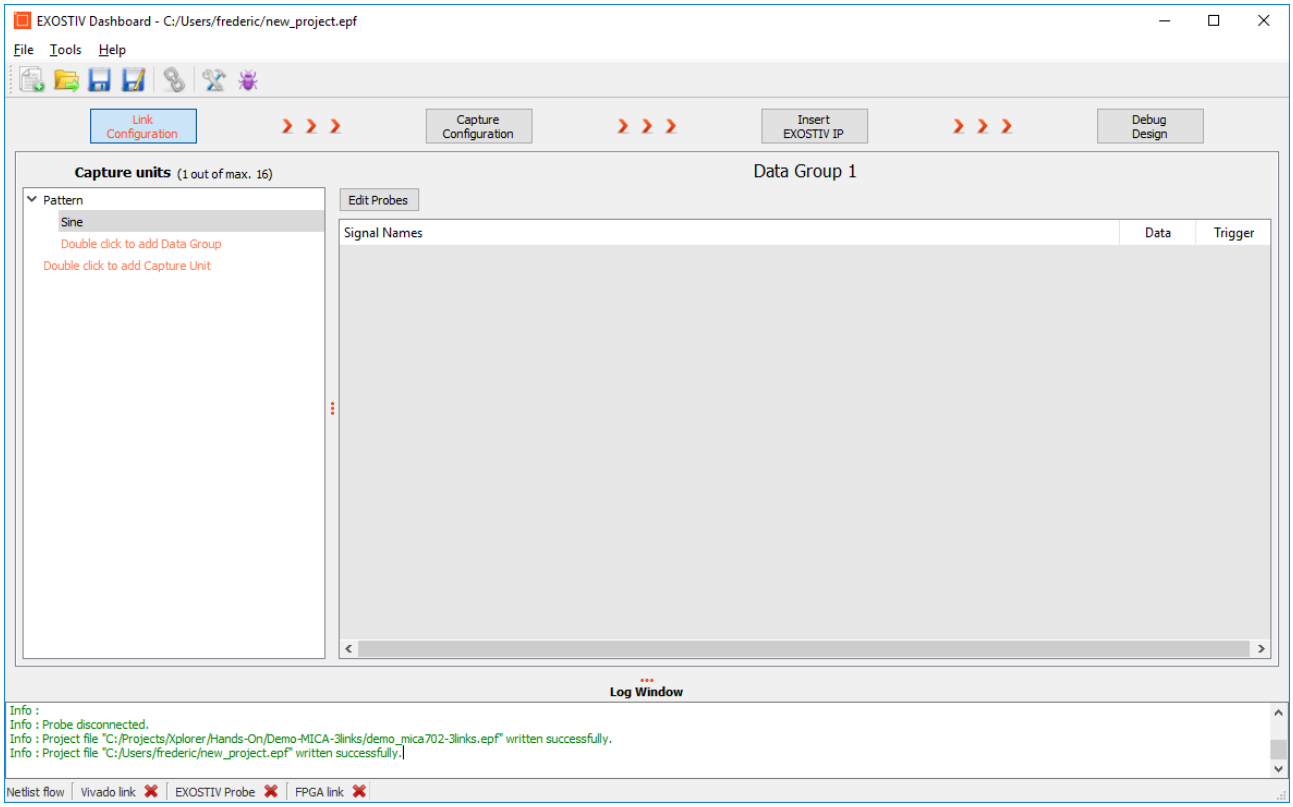


For 'Capture Unit 1', select the following:

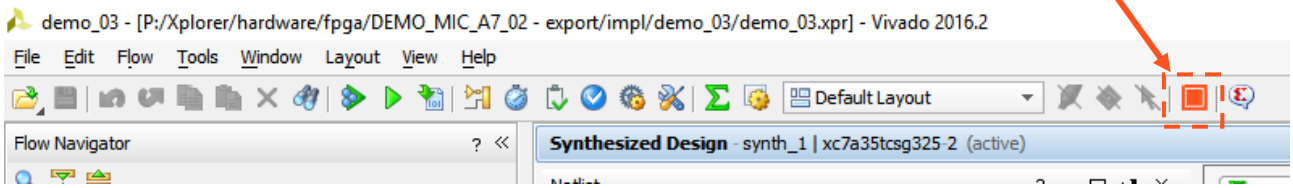
1. Trigger unit type: **Levels / Edges / Comparisons**
2. Fifo Depth : **1024**
3. Leave the other settings as they are.
4. Double-click on the Capture Unit name to change it. Change it to 'Pattern'.
5. Click on the data group name to define or see the connected nodes
6. Double-click on the Data Group name to change it. Change it to 'Sine'.



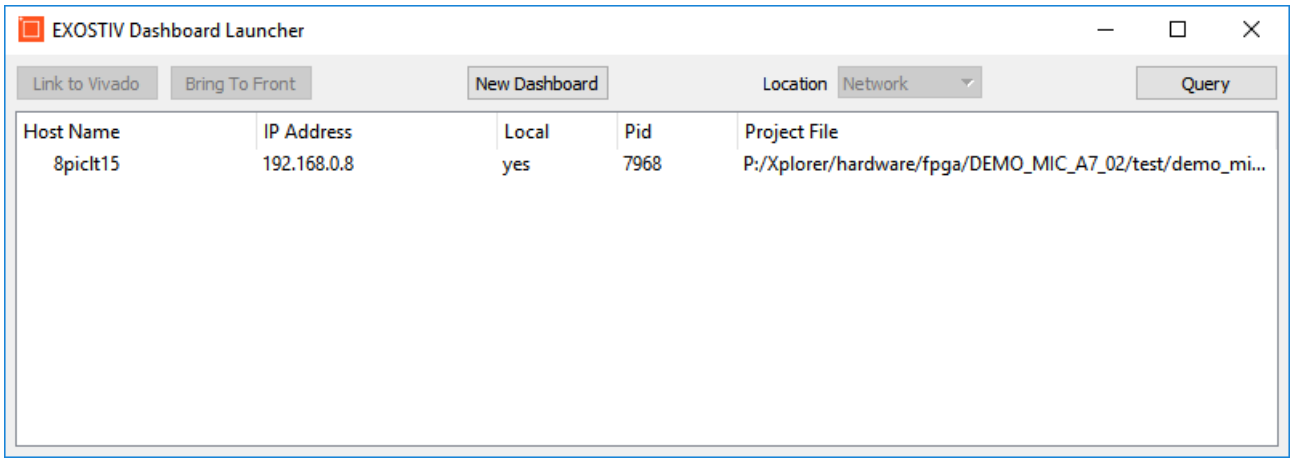


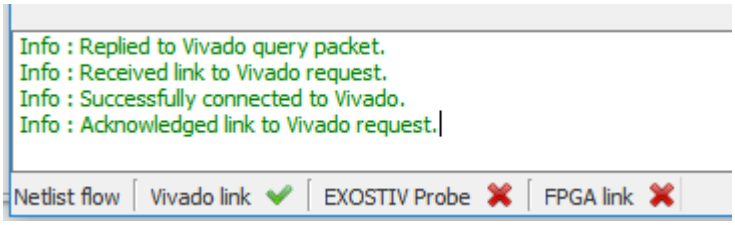


7. We now have to connect EXOSTIV Dashboard to Vivado, in order to select the nodes to be observed. Switch back to Vivado and click on 'EXOSTIV Dashboard' shortcut in the main toolbar.

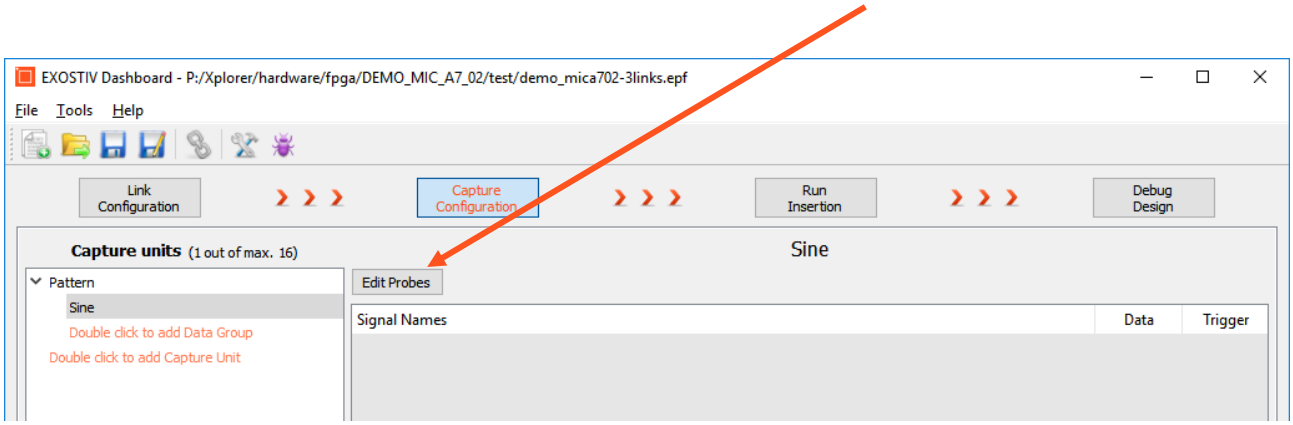


8. In the 'EXOSTIV Dashboard Launcher' window, select the running instance of the EXOSTIV Dashboard and click on 'Link to Vivado'. This establishes a link between Vivado and EXOSTIV Dashboard.

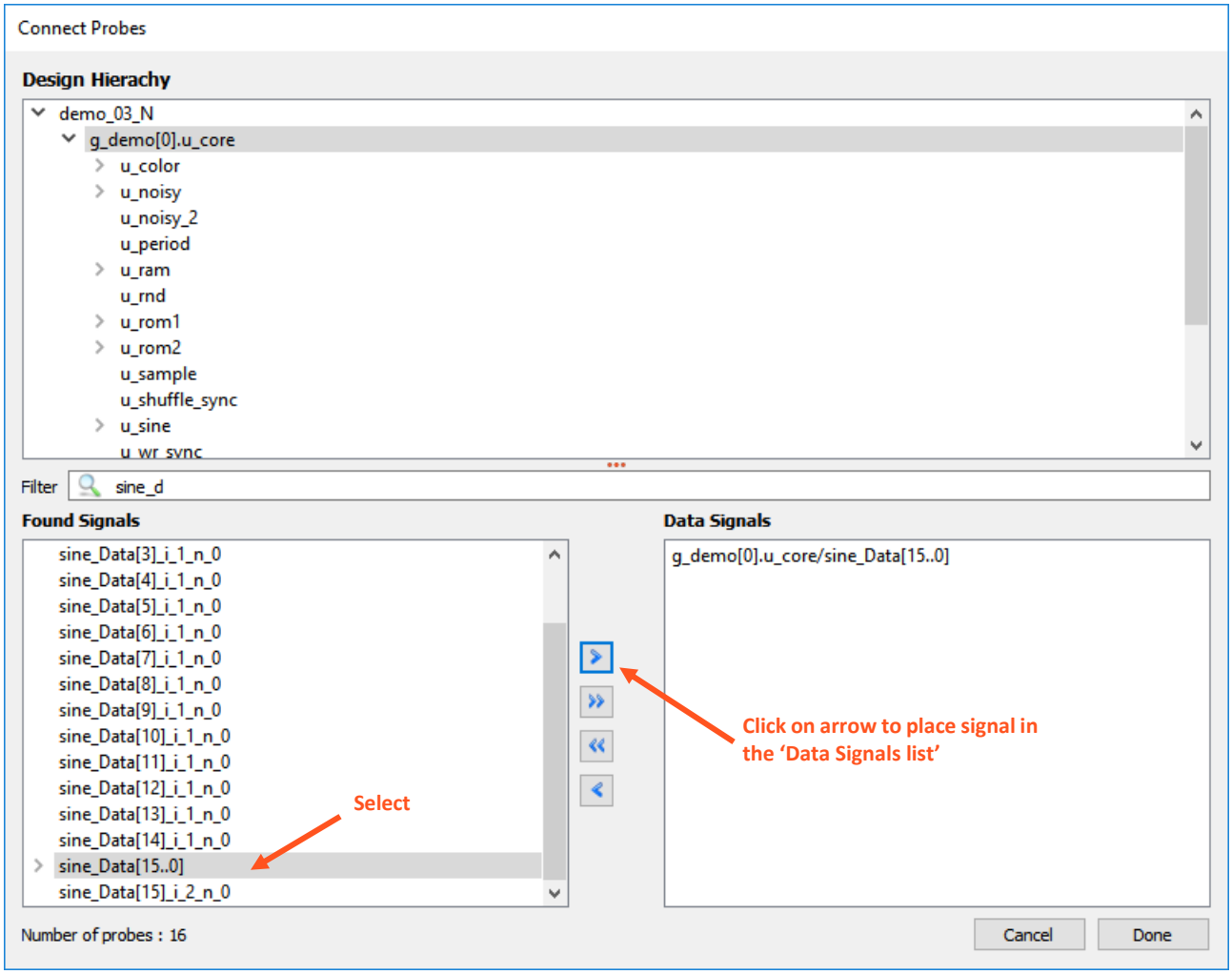




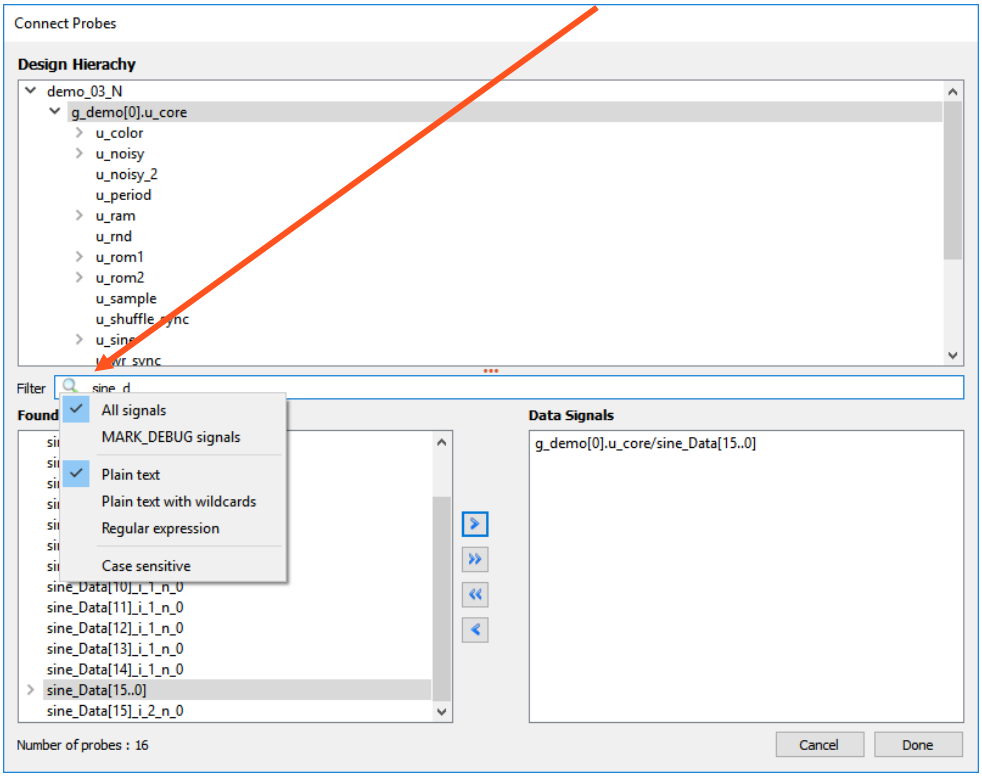
9. In EXOSTIV Dashboard, select the 'Sine' data group and click on 'Edit Probes'



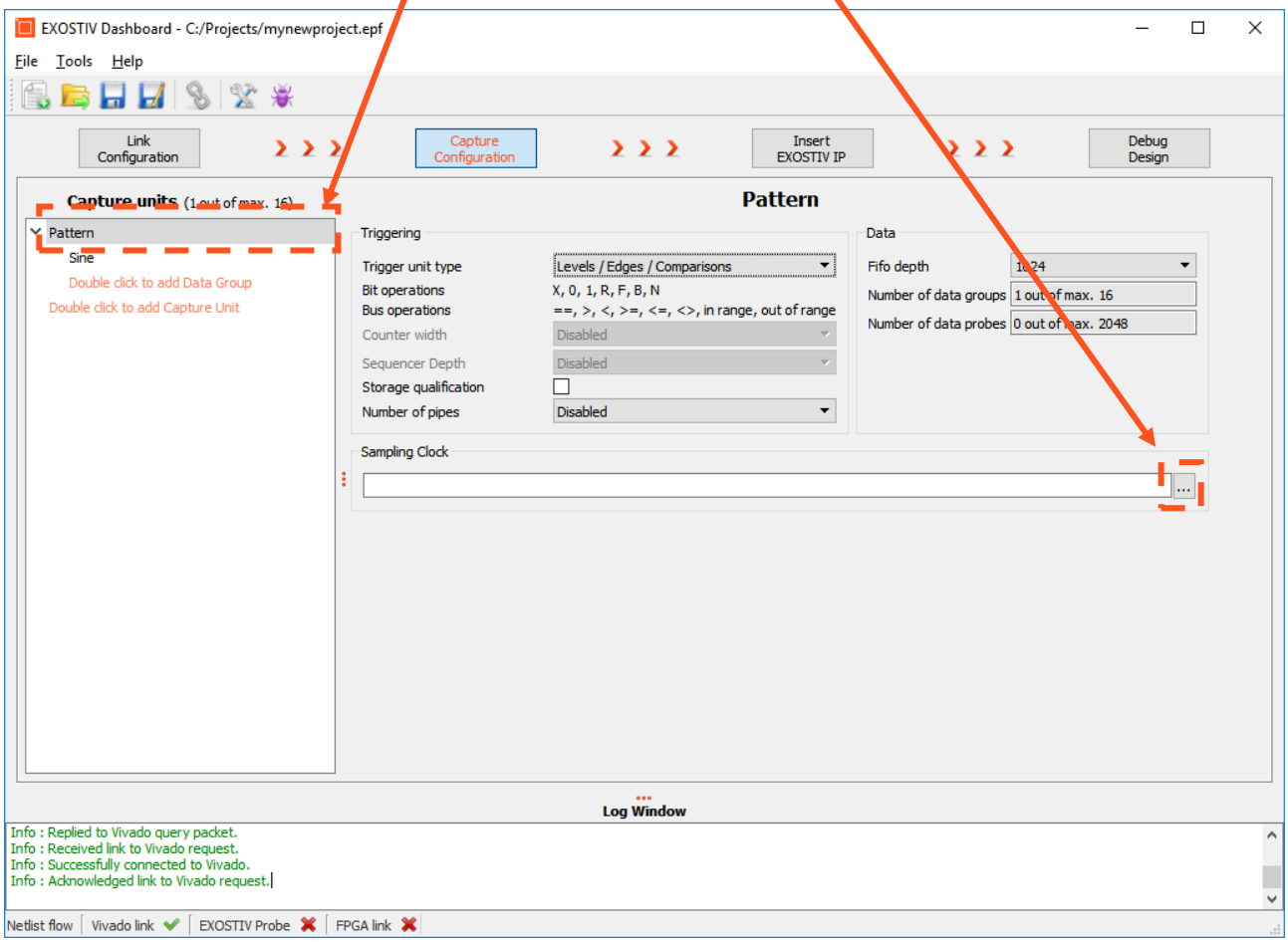
10. From the 'Connect Probes' window, browse the design and select the signal 'u_demo/sine_Data[15:0]



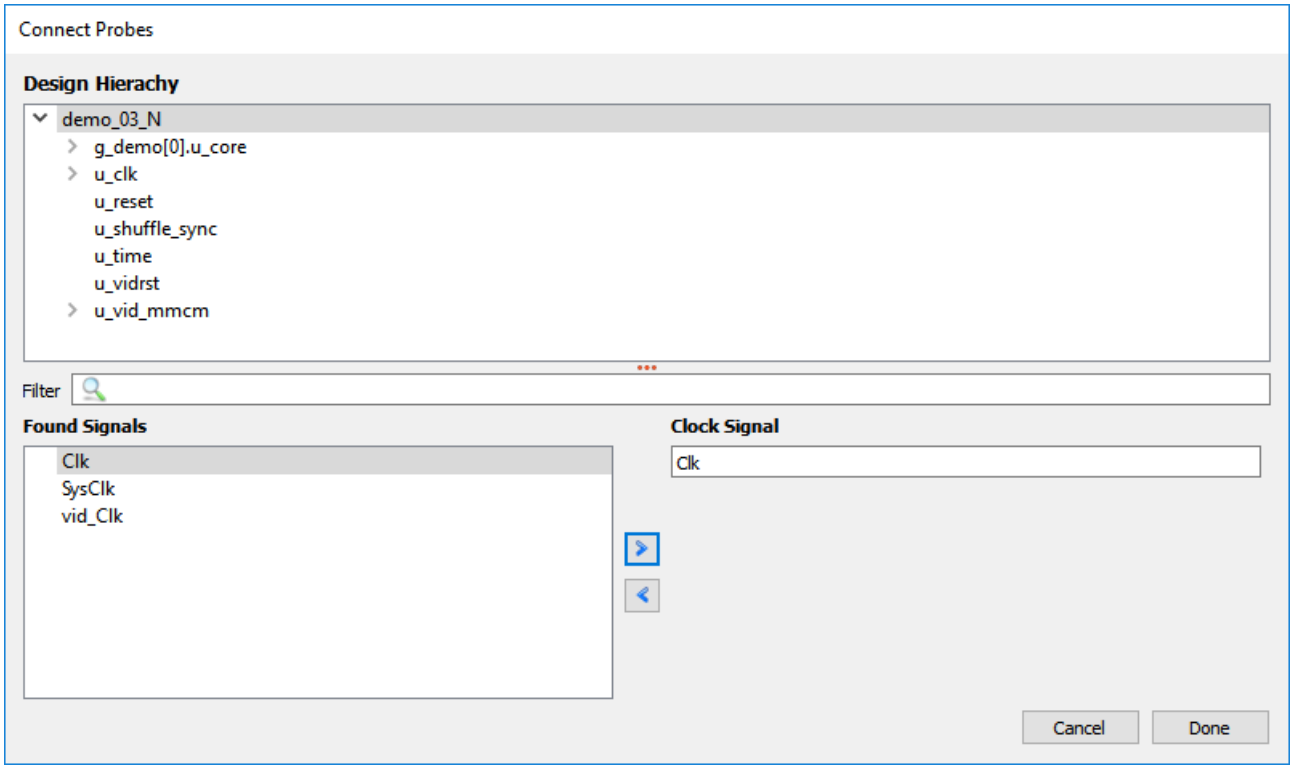
11. If necessary, change the signal filter by clicking on the magnifier icon:



12. Select 'Pattern' capture unit and click on the '…' button to define the capture unit's sampling clock:



13. Select ‘demo_03_N/Clk’, bring it to the ‘Clock signal’ box with the ‘>’ and click on ‘Done’.



14. Set up 2 additional capture units, with the parameters of the table below (or load the reference project file ‘**demo_mica702-3links.epf’** if you want to skip this step)

	Name	**Trigger unit type**	**Storage qualification**	**Number of pipes**	**Fifo depth**	**Number of data groups**	**Number of data probes**	**Sampling Clock**
CU1	Pattern	Levels/Edges/Comparisons	NO	Disabled	1024	3	16	Clk
CU2	Video	Levels/Edges/Comparisons	NO	Disabled	2048	2	46	vid_Clk
CU3	Video-Extended	Levels/Edges	NO	Disabled	1024	1	80	vid_Clk

Capture Unit	**Data group**	**Nodes**	**Data**	**Trigger**
CU1 ‘Pattern’	Cnt	g_demo[0].u_core/Cnt[15..0]	YES	YES
	Sine	g_demo[0].u_core/sine_Data[15..0]	YES	YES
	Noise	g_demo[0].u_core/Rnd[15..0]	YES	YES
CU2 ‘Video’	SDI	g_demo[0].u_core/sdi_SOF	YES	YES
		g_demo[0].u_core/sdi_VBlank	YES	YES
		g_demo[0].u_core/sdi_HBlank	YES	YES
		g_demo[0].u_core/sdi_Valid	YES	YES
		g_demo[0].u_core/sdi_LN[11..0]	YES	YES
		g_demo[0].u_core/sdi_R[9..0]	YES	NO
		g_demo[0].u_core/sdi_G[9..0]	YES	NO
		g_demo[0].u_core/sdi_B[9..0]	YES	NO
	Noise	g_demo[0].u_core/vid_Sine[15..0]	YES	YES
		g_demo[0].u_core/vid_Noise[17..0]	YES	NO
		g_demo[0].u_core/vid_Addr[9..0]	YES	NO
CU3 ‘Video-Extended’	Vid-Extended	g_demo[0].u_core/sdi_SOF	YES	YES
		g_demo[0].u_core/sdi_VBlank		YES
		g_demo[0].u_core/sdi_HBlank		YES
		g_demo[0].u_core/sdi_Valid		YES
		g_demo[0].u_core/sdi_LN[11..0]		YES
		g_demo[0].u_core/sdi_R[9..0]		NO
		g_demo[0].u_core/sdi_G[9..0]		NO

		g_demo[0].u_core/sdi_B[9..0] g_demo[0].u_core/vid_Sine[15..0] g_demo[0].u_core/vid_Noise[17..0]		NO NO NO

✓ CAPTURE CONFIGURATION: DONE !

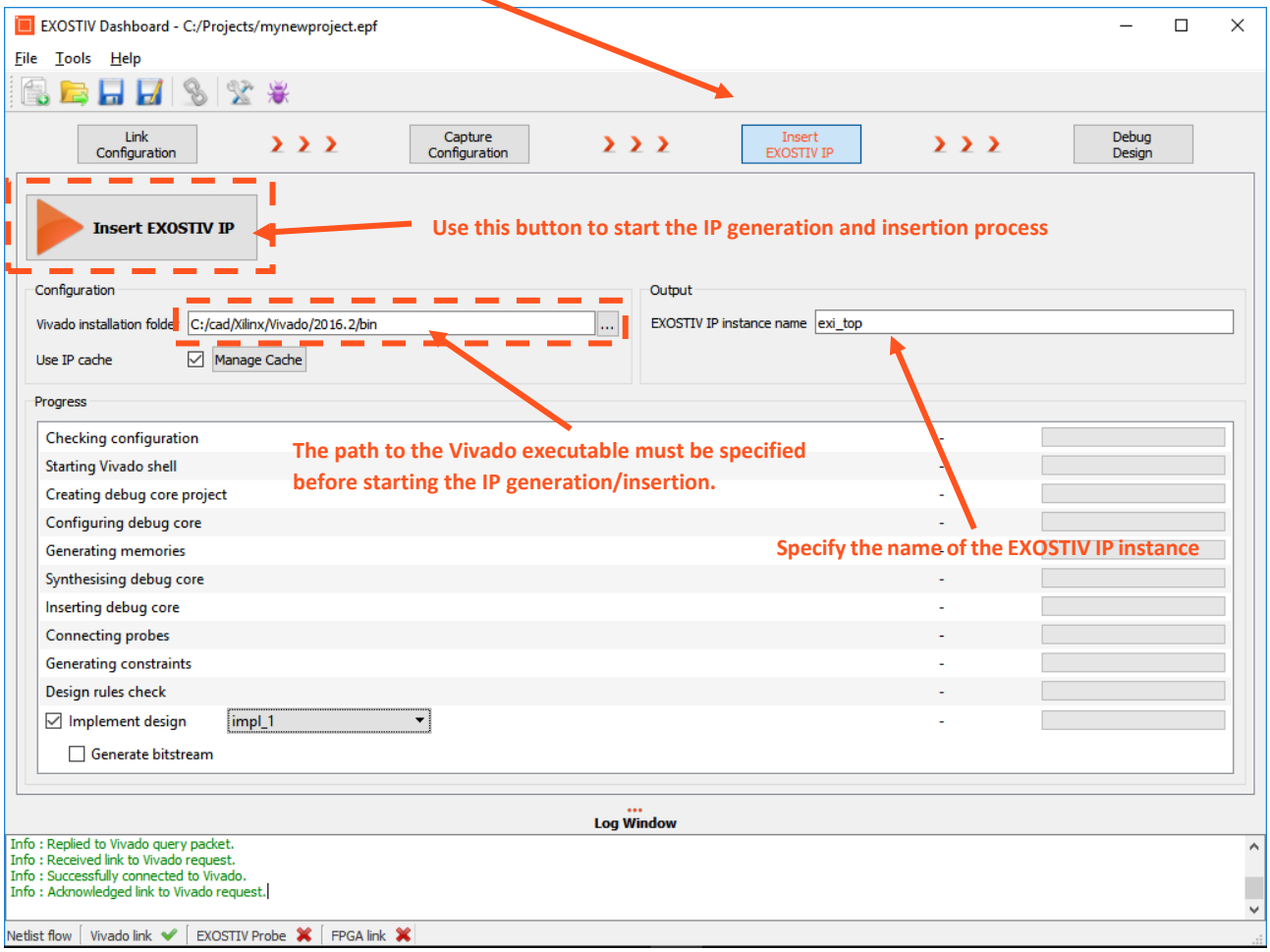
(! Don't forget to save your project !)

Step 3: Run Insertion

This part of the flow executes the following:

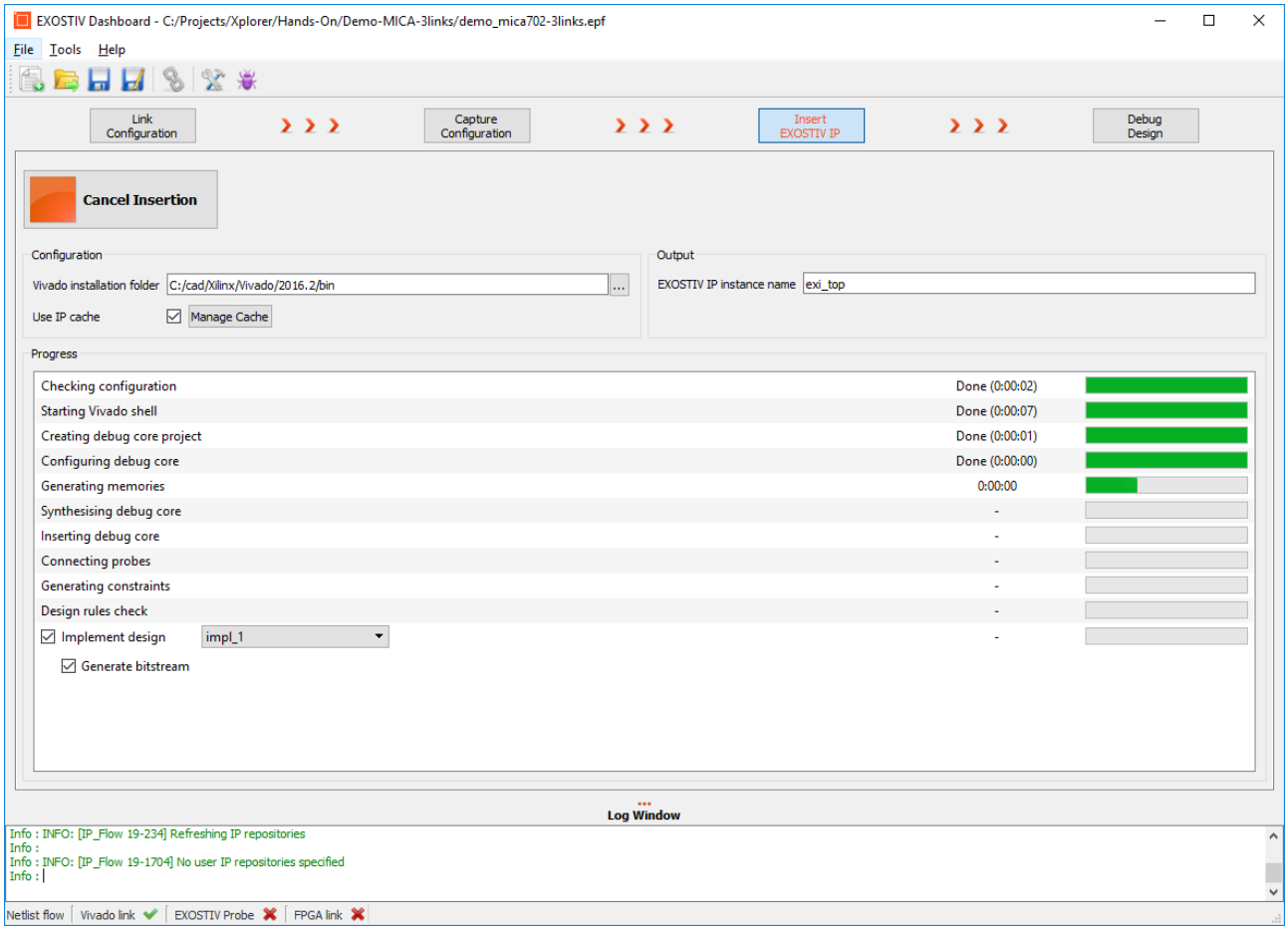
- IP configuration checks
- EXOSTIV IP synthesis with Vivado
- EXOSTIV IP insertion in the target FPGA netlist
- Additional constraints generation
- Instrumented design implementation and bitstream generation.

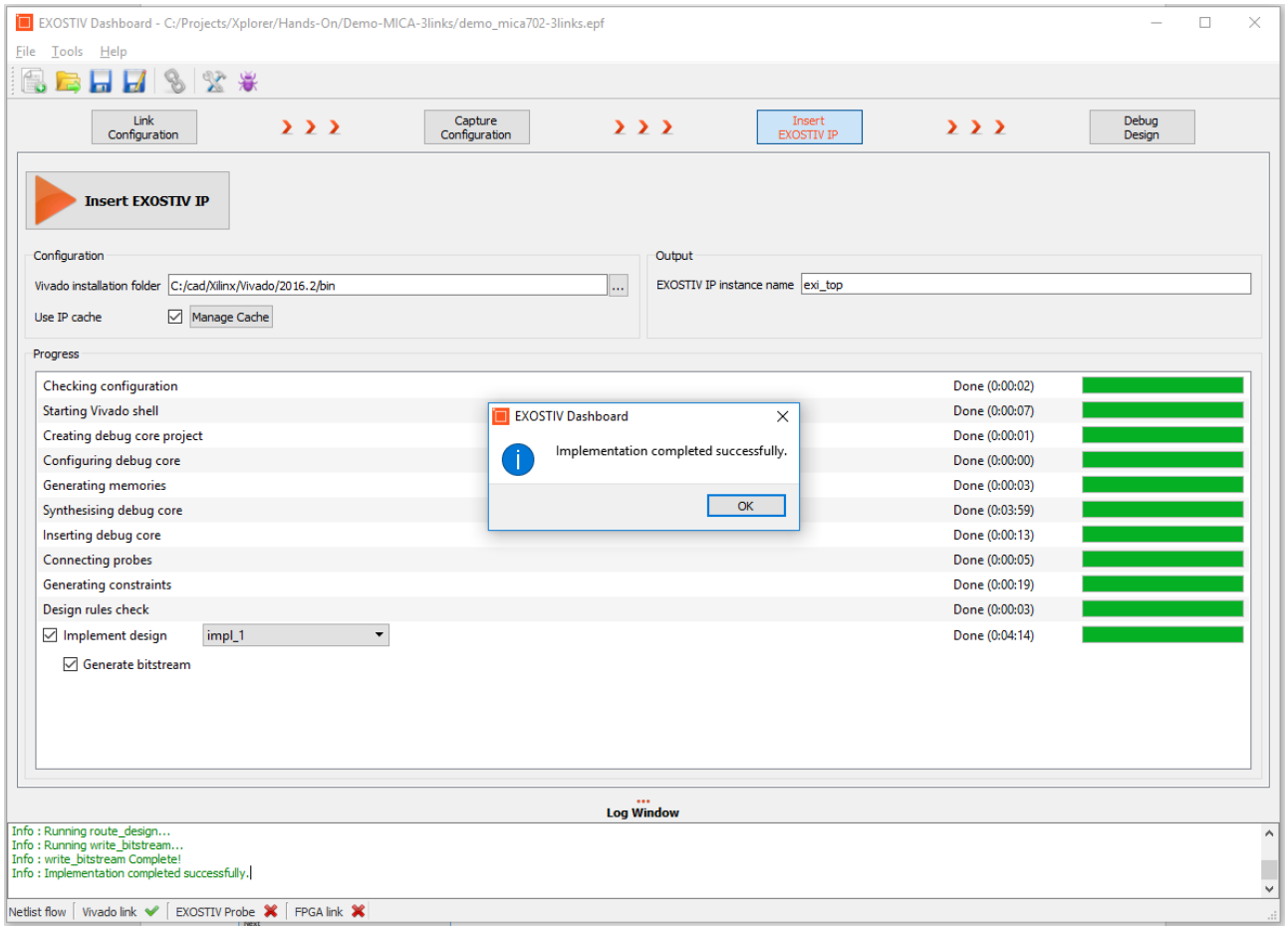
1. Click on 'Insert EXOSTIV IP'



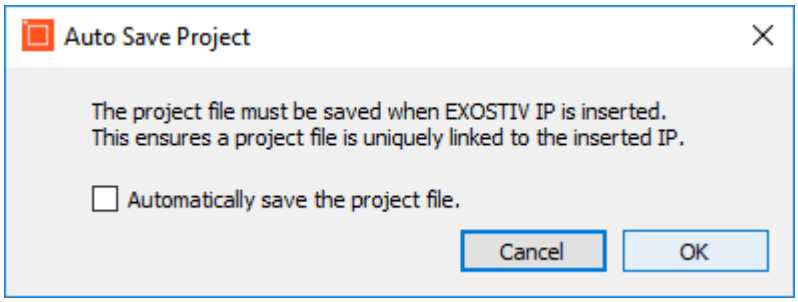
2. Enable Implementation and select 'impl_1' from the drop down list:
   Select 'Implement Design' and 'Generate bitstream'.
3. Click on 'Insert EXOSTIV IP' button.

Insertion running … and completed.

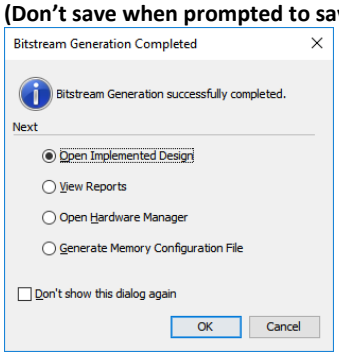

EXOSTIV Dashboard - C:/Projects/Xplorer/Hands-On/Demo-MICA-3links/demo_mica702-3links.epf
File Tools Help
Link Configuration
Capture Configuration
Insert EXOSTIV IP
Debug Design
Cancel Insertion
Configuration
Vivado installation folder C:/cad/Xilinx/Vivado/2016.2/bin
Use IP cache Manage Cache
Output
EXOSTIV IP instance name exi_top
Progress
Checking configuration Done (0:00:02)
Starting Vivado shell Done (0:00:07)
Creating debug core project Done (0:00:01)
Configuring debug core Done (0:00:00)
Generating memories 0:00:00
Synthesising debug core -
Inserting debug core -
Connecting probes -
Generating constraints -
Design rules check -
Implement design impl_1 -
Generate bitstream
Log Window
Info : INFO: [IP_Flow 19-234] Refreshing IP repositories
Info :
Info : INFO: [IP_Flow 19-1704] No user IP repositories specified
Info :
Netlist flow Vivado link EXOSTIV Probe FPGA link

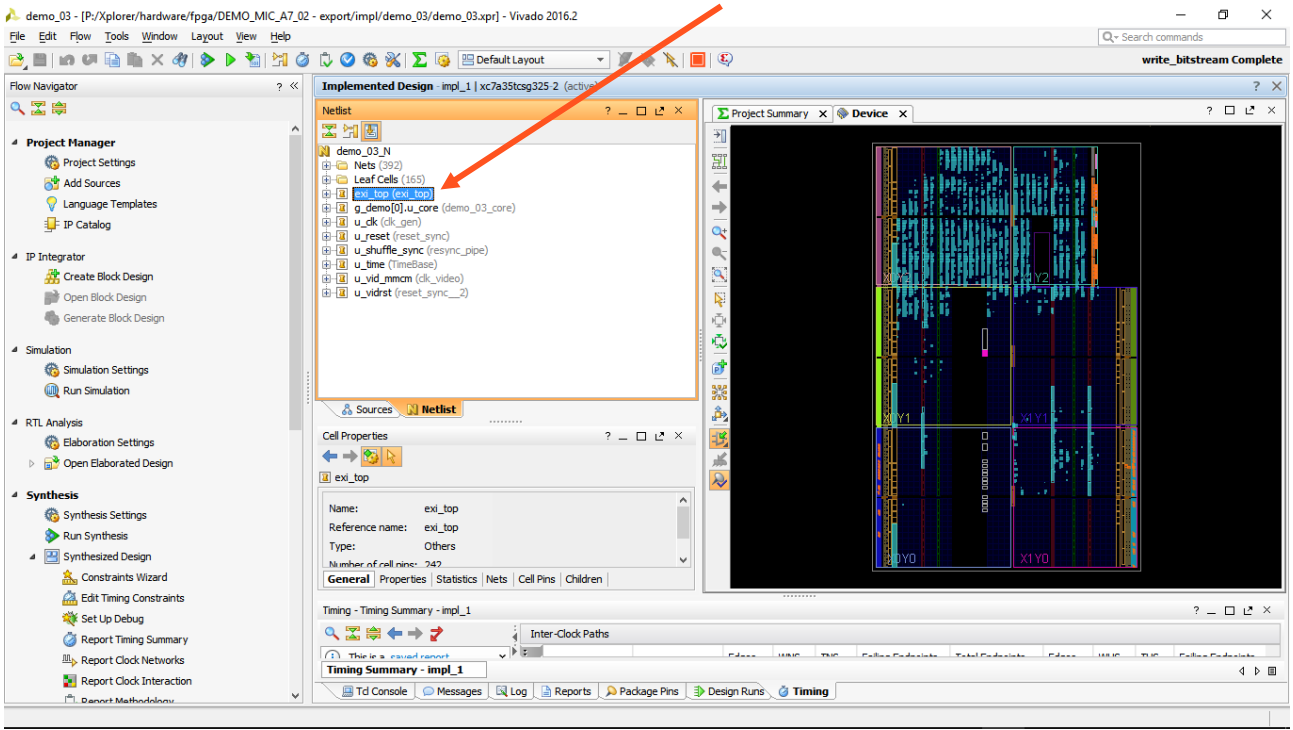

EXOSTIV Dashboard - C:/Projects/Xplorer/Hands-On/Demo-MICA-3links/demo_mica702-3links.epf
File Tools Help
Link Configuration
Capture Configuration
Insert EXOSTIV IP
Debug Design
Insert EXOSTIV IP
Configuration
Vivado installation folder C:/cad/Xilinx/Vivado/2016.2/bin
Use IP cache Manage Cache
Output
EXOSTIV IP instance name exi_top
Progress
Checking configuration Done (0:00:02)
Starting Vivado shell Done (0:00:07)
Creating debug core project Done (0:00:01)
Configuring debug core Done (0:00:00)
Generating memories Done (0:00:03)
Synthesising debug core Done (0:03:59)
Inserting debug core Done (0:00:13)
Connecting probes Done (0:00:05)
Generating constraints Done (0:00:19)
Design rules check Done (0:00:03)
Implement design impl_1 Done (0:04:14)
Generate bitstream
EXOSTIV Dashboard
Implementation completed successfully.
OK
Log Window
Info : Running route_design...
Info : Running write_bitstream...
Info : write_bitstream Complete!
Info : Implementation completed successfully.
Netlist flow Vivado link EXOSTIV Probe FPGA link

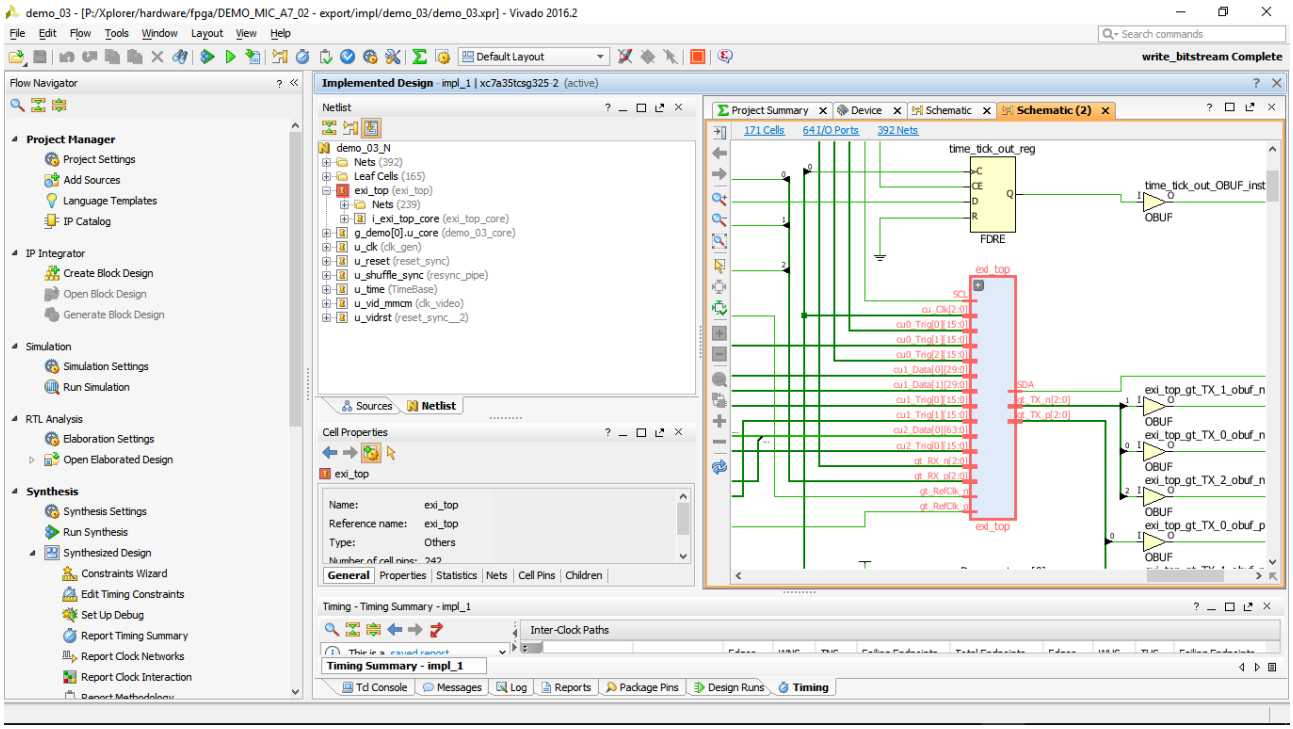
Click on 'OK' at prompt to save the project.



We can check that 'exi_top' was inserted into the design if you open the 'design implementation' in Vivado **(Don't save when prompted to save synthesized design)**:



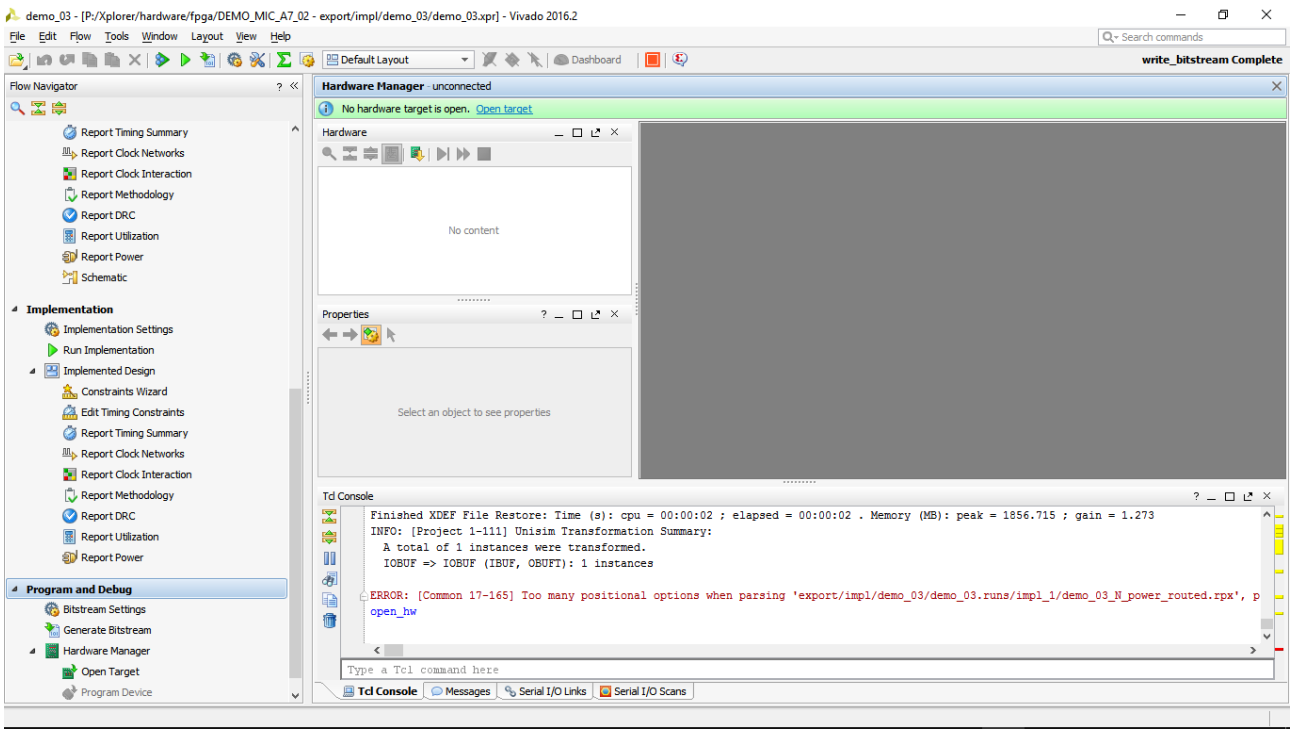




Program the target board

We now need to use the generated binary to program the target board.

- Return to Vivado and open the 'Hardware Manager'.
- Connect the programming cable to the MICA board
- Turn on the MICA board by enabling the output HDMI power and connecting to the EXOSTIV Probe.
- Once connected, right-click on the target FPGA and select 'Program'.
- Then, select the generated bit file and load the FPGA configuration.

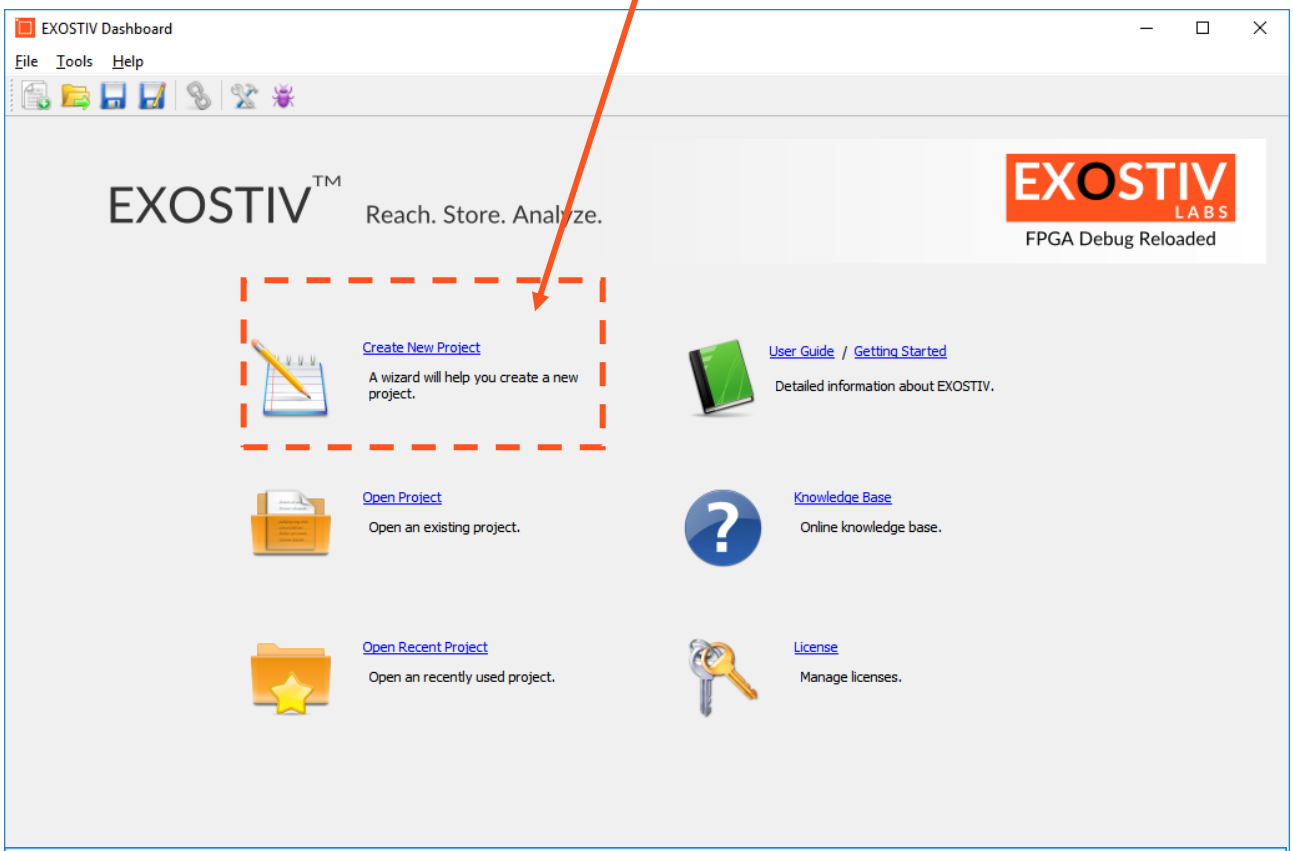


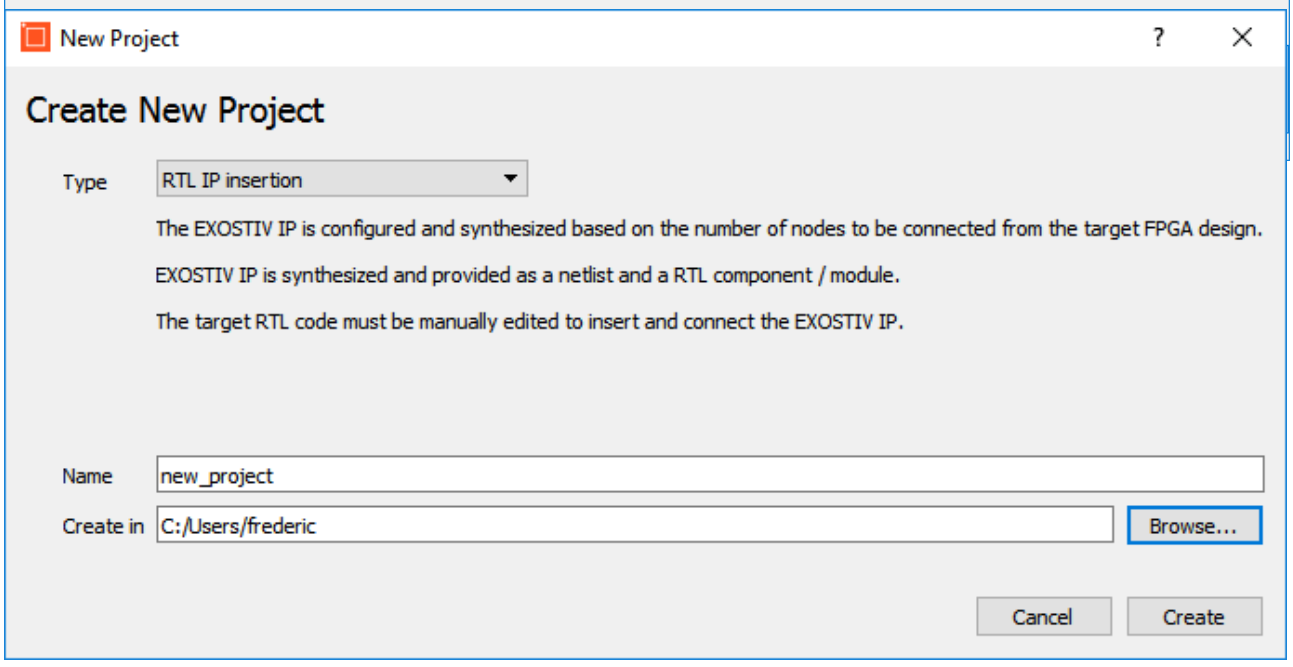
Creating a ‘RTL flow’ project with EXOSTIV Dashboard

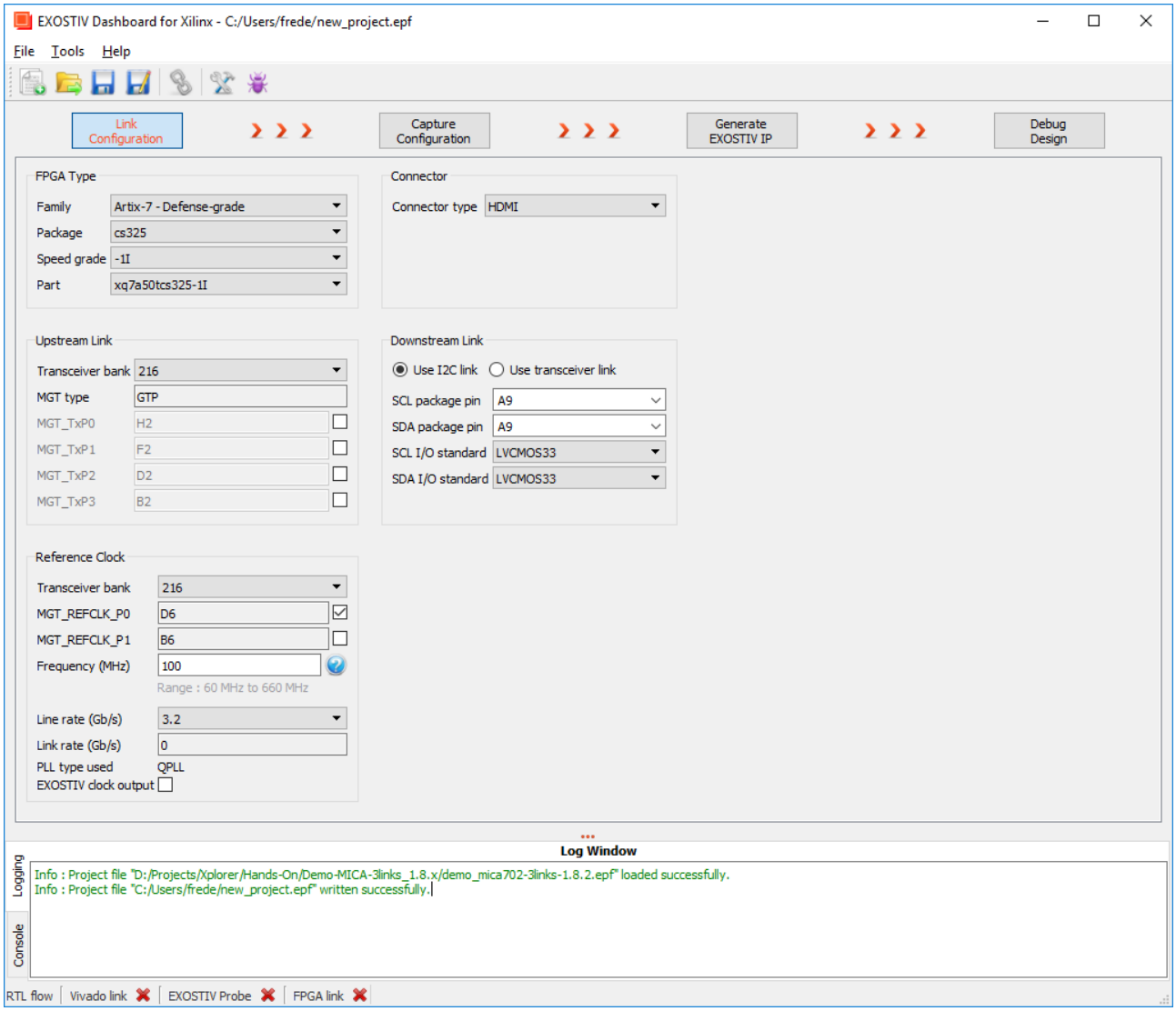
The ‘RTL flow’ enables configuring and synthesizing EXOSTIV IP for insertion in RTL source code. The ‘RTL flow’ is somewhat simpler than the ‘netlist flow’ as it does not involve user interactions with Vivado. EXOSTIV IP is inserted and connected to the target FPGA design nodes by instantiating EXOSTIV IP in the RTL code – a ‘manual’ process.

1. From the Welcome screen, click on ‘Create New Project’

The ‘Create New Project’ window appears, prompting to select the desired flow (Netlist flow or RTL flow).
Specify a project name and pick a location.
The ‘Link Configuration window’ of the core inserter appears.







Using the IP generator (RTL flow)

In RTL flow, the EXOSTIV Dashboard allows to **generate EXOSTIV IP.** Unlike the 'netlist flow', RTL flow won't automatically insert EXOSTIV IP into the target design. This has to be done 'manually', in the RTL code (VHDL / Verilog).

Generating EXOSTIV IP in RTL flow requires 3 steps:

- **Step 1 :** Link Configuration
- **Step 2 :** Capture Configuration
- **Step 3 :** Generate EXOSTIV IP

These 3 steps are accessible through the top flow overview in the EXOSTIV Dashboard window

Step 1: Link Configuration

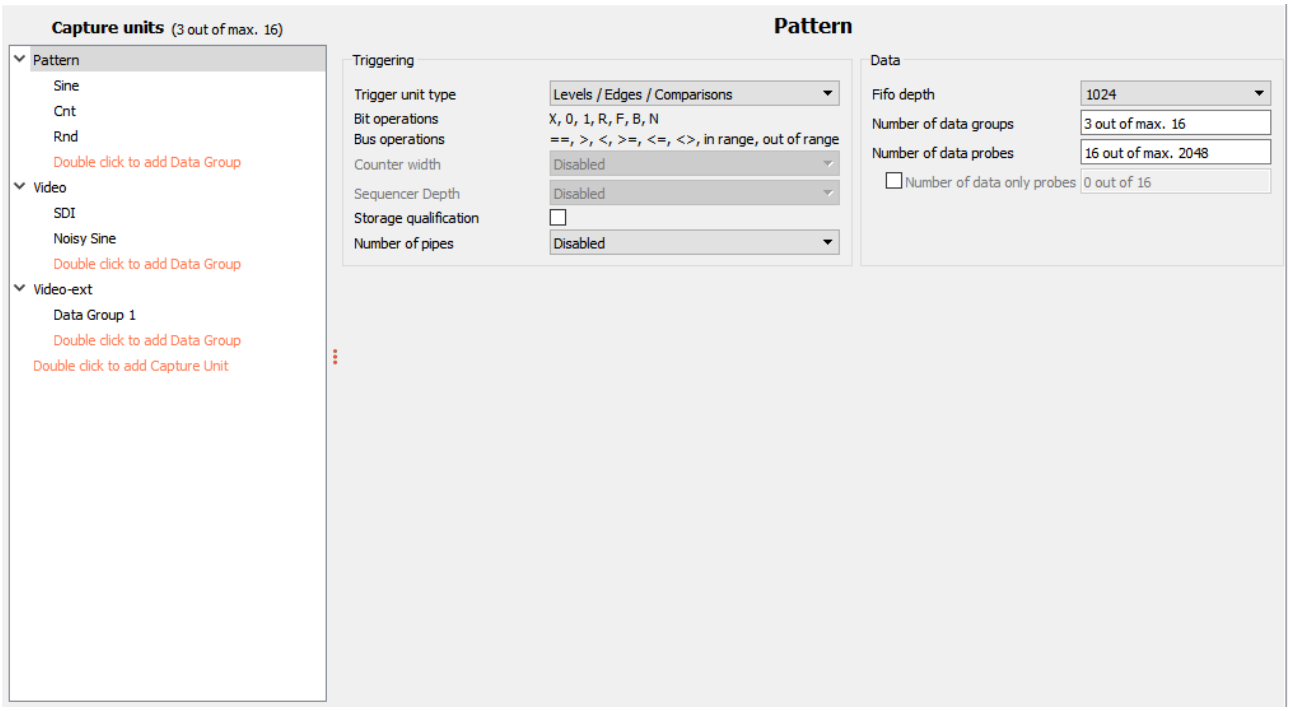
This step in RTL flow does not differ from the same step in netlist flow. Please refer to Step 1: Link Configuration, page 23 for an illustrated example on how to use it.

Step 2 : Capture Configuration

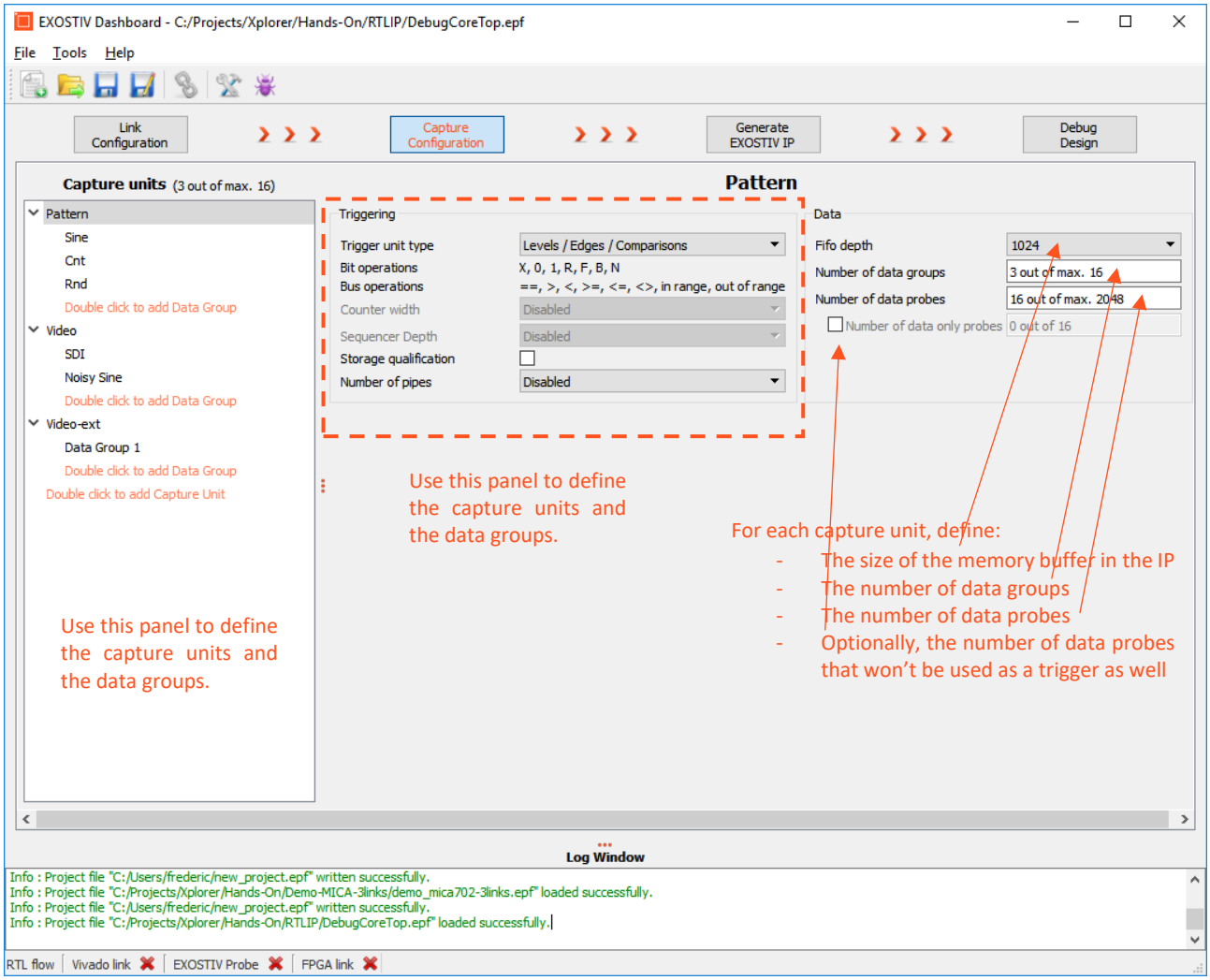
During this step, the characteristics of EXOSTIV IP are defined:

- Capture units
  - Number and names
  - Trigger options
  - Size of the memory buffer

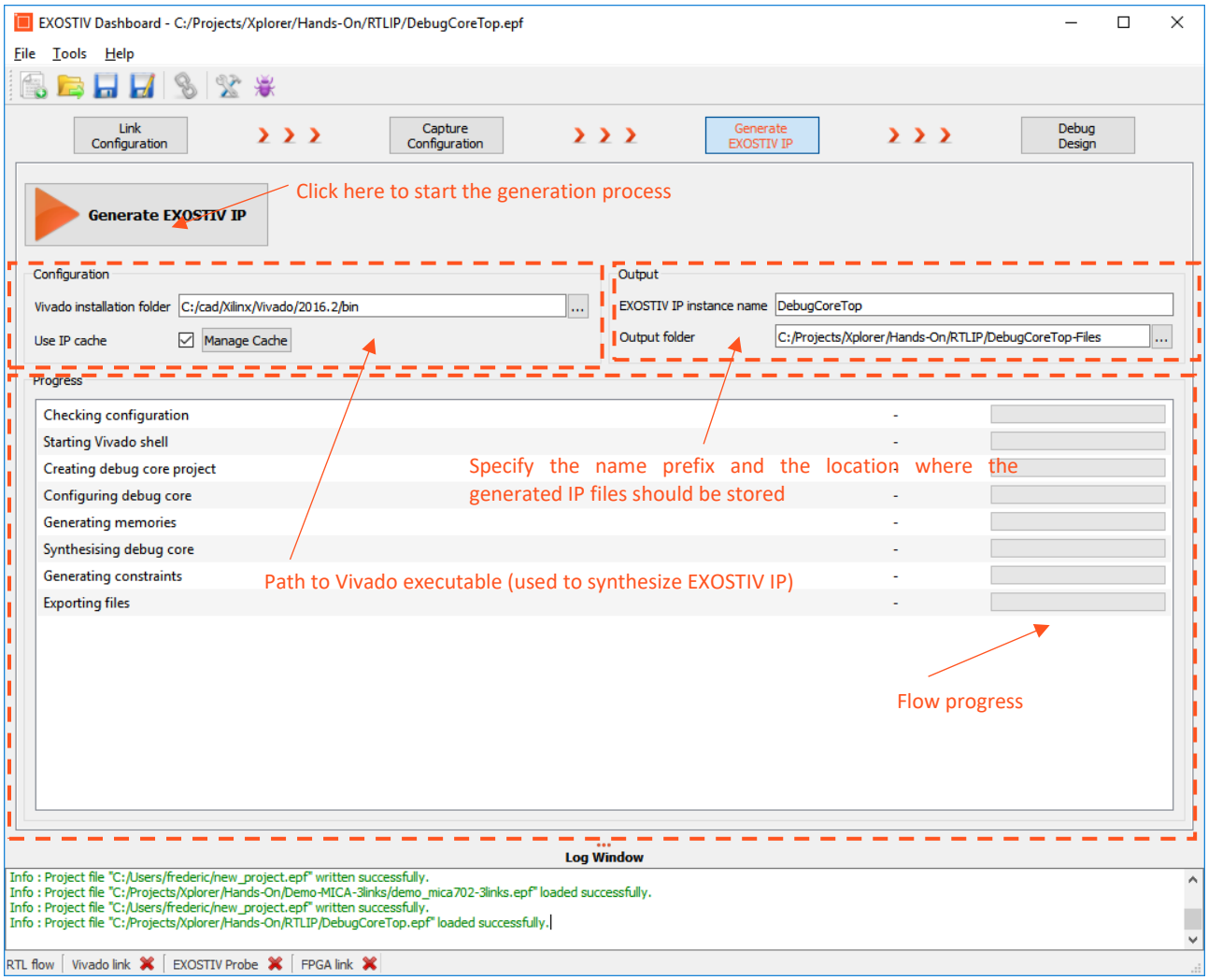
- Data groups
  - Number and names
  - **As opposed to the 'netlist flow' described at page (-), data group definition is limited to:**
    - **The number of bits in each data group**
    - **The quality of the inputs of each data group: the number of 'data / trigger' nodes and the number of 'data only' nodes.**

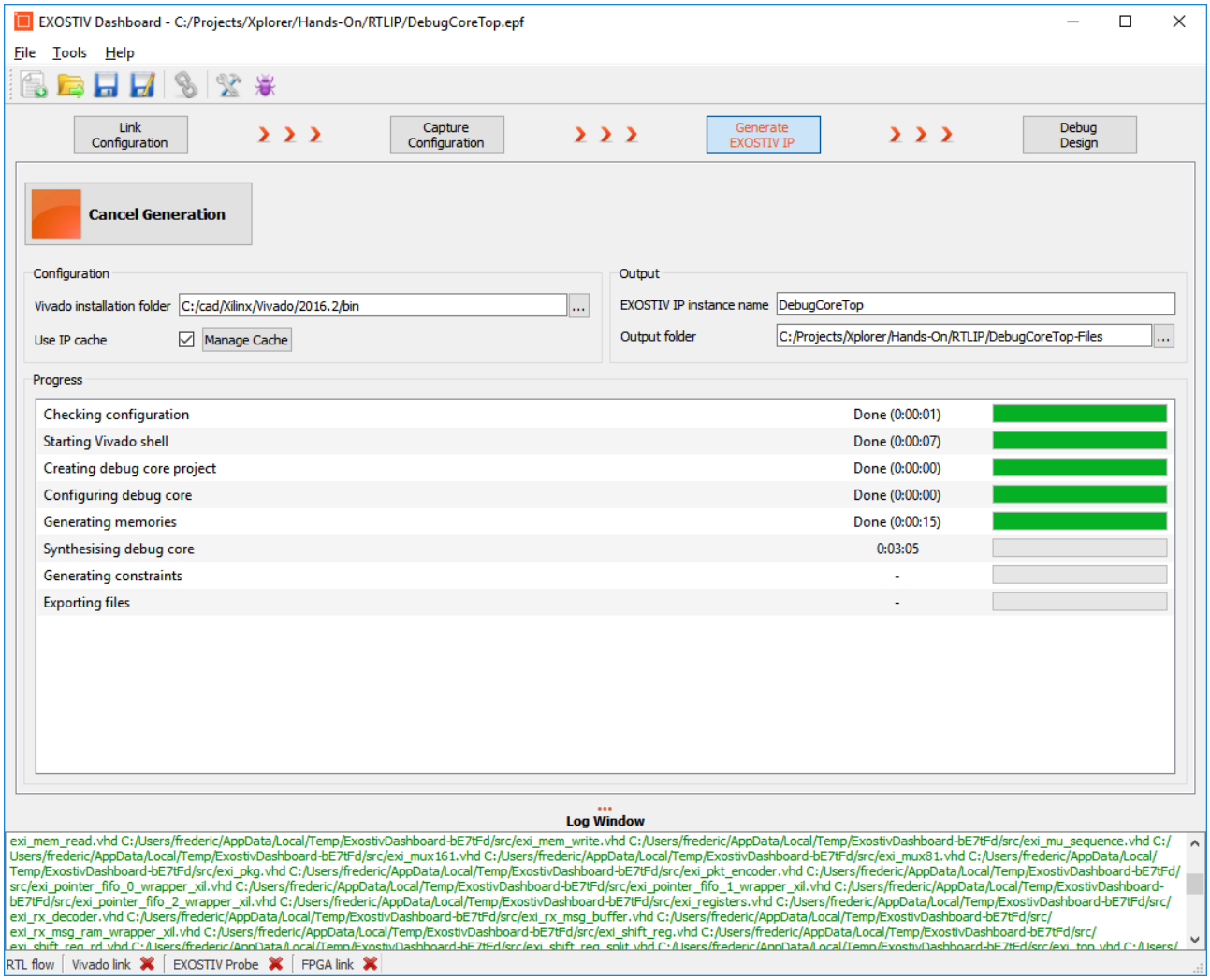


Use this panel to define the Capture units and the data groups.

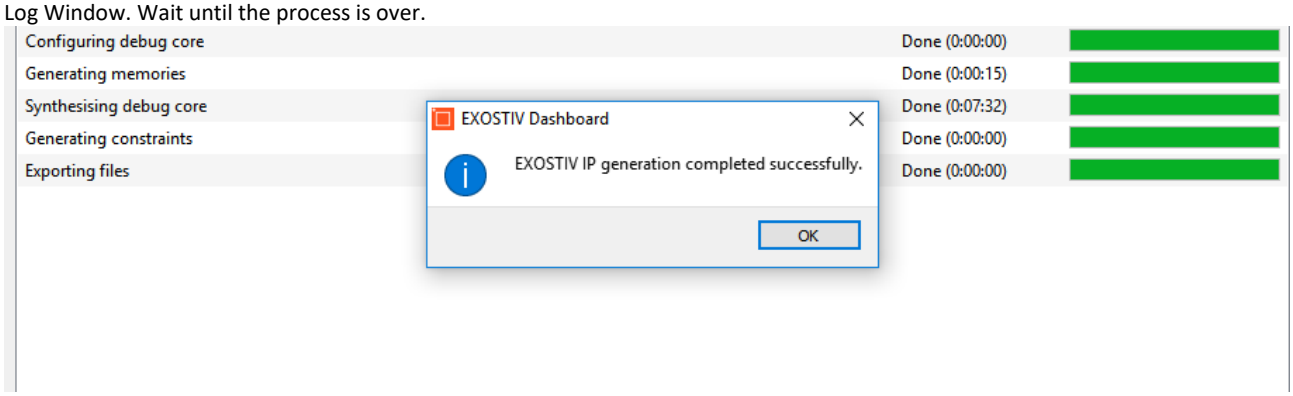

EXOSTIV Dashboard - C:/Projects/Xplorer/Hands-On/RTLIP/DebugCoreTop.epf
File Tools Help
Link Configuration
Capture Configuration
Generate EXOSTIV IP
Debug Design
Capture units (3 out of max. 16)
Pattern
Sine
Cnt
Rnd
Double click to add Data Group
Video
SDI
Noisy Sine
Double click to add Data Group
Video-ext
Data Group 1
Double click to add Data Group
Double click to add Capture Unit
Pattern
Triggering
Trigger unit type
Levels / Edges / Comparisons
Bit operations
X, 0, 1, R, F, B, N
Bus operations
==, >, <, >=, <=, <>, in range, out of range
Counter width
Disabled
Sequencer Depth
Disabled
Storage qualification
Number of pipes
Disabled
Data
Fifo depth
1024
Number of data groups
3 out of max. 16
Number of data probes
16 out of max. 2048
Number of data only probes
0 out of 16
Use this panel to define the capture units and the data groups.
Use this panel to define the capture units and the data groups.
For each capture unit, define:
- The size of the memory buffer in the IP
- The number of data groups
- The number of data probes
- Optionally, the number of data probes that won't be used as a trigger as well
Log Window
Info : Project file "C:/Users/frederic/new_project.epf" written successfully.
Info : Project file "C:/Projects/Xplorer/Hands-On/Demo-MICA-3links/demo_mica702-3links.epf" loaded successfully.
Info : Project file "C:/Users/frederic/new_project.epf" written successfully.
Info : Project file "C:/Projects/Xplorer/Hands-On/RTLIP/DebugCoreTop.epf" loaded successfully.
RTL flow
Vivado link
EXOSTIV Probe
FPGA link

Step 3 : Generate EXOSTIV IP





Once started, the IP generation progress is displayed in the window. Messages from the running session of Vivado appear in the Log Window. Wait until the process is over.



Overview of the files generated with the RTL flow

A series of files are generated by the EXOSTIV Dashboard RTL flow IP generation process. These files must be used with the target FPGA design in order to instrument the target FPGA and use EXOSTIV to debug the FPGA.

All the files names are prefixed with the EXOSTIV IP instance name specified in the project. Here is the list of generated files and their usage:

File name	Usage	Add to Vivado project?
<Instance name>.edf	EXOSTIV IP synthesized netlist.	YES
<Instance name>.vhd	Example template on how to instantiate the EXOSTIV IP in the target design.	NO this is an example template.
<Instance name>_pinout.xdc	Constraint file containing the pinout required by the EXOSTIV IP (like transceiver location).	YES
<Instance name>_pkg.vhd	VHDL Package file containing types used for the EXOSTIV IP instantiation.	YES
<Instance name>_timing.xdc	Constraint file containing the timing constraints relative to EXOSTIV IP.	NO, should be 'sourced' in Vivado, not just added.

RTL flow: inserting EXOSTIV IP and implementing the design

From here, the RTL code of the target design has to be modified manually. An instance of EXOSTIV IP has to be created. The connection with the design internal nodes should be done at this step, from the RTL code.

Then the required files should be added to the Vivado project and the synthesis / implementation / bitstream generation of the instrumented design can be run from the Vivado interface.

Finally, the bitstream has to be used with the target board, and EXOSTIV analyzer can be used for extracting trace data (refer to page 12).

Copyright

Exostiv Labs sprl is a company registered in Belgium, 18 Avenue Molière, 1300 Wavre.
VAT / REG nr: BE0873.279.914.

Disclaimer

https://www.exostivlabs.com