Exostiv Core Inserter

User's Guide

Rev. 1.0.1 - March 13, 2025

Table of Contents

Revision History

Revision	Modifications
1.0.0	• Initial revision
1.0.1	• Provided details about netlist and RTL flows. • Added details about the ECO flow • Added details about Versal devices flow
1.0.2	• Added description for Extended width IP

Exostiv Core Inserter

Introduction

This guide provides instructions on how to use Exostiv Core Inserter for use with EP16000 probes and Exostiv Blade units. Exostiv Core Inserter – or simply 'Core Inserter' allows generating, inserting and modifying Exostiv IP cores.

For instructions about how to install Exostiv Core Inserter, refer to Exostiv Labs website knowledge base.

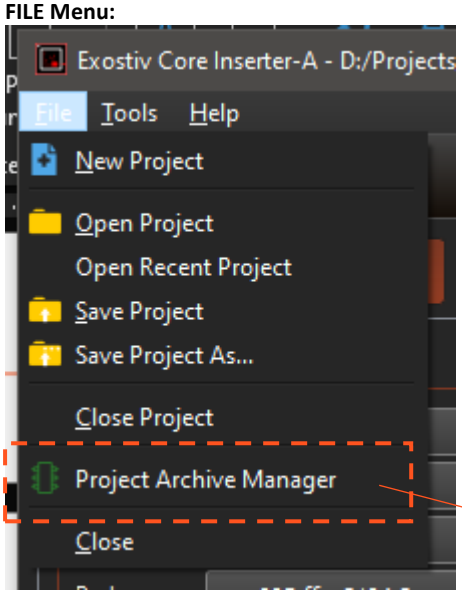
The Exostiv Labs online searchable knowledge base provides support information complementary to this guide.
https://www.exostivlabs.com/knowledgebase

Core Inserter – Welcome screen

From there, you can create or open a project, access the documentation and manage your software license.

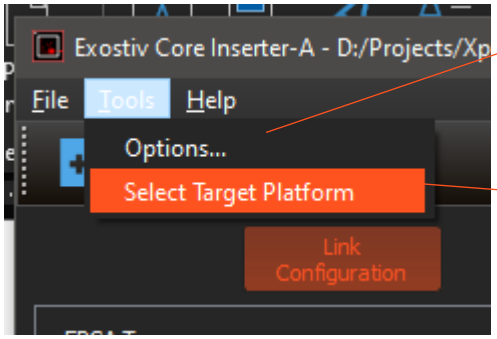
Exostiv Core Inserter: Menus

FILE Menu:



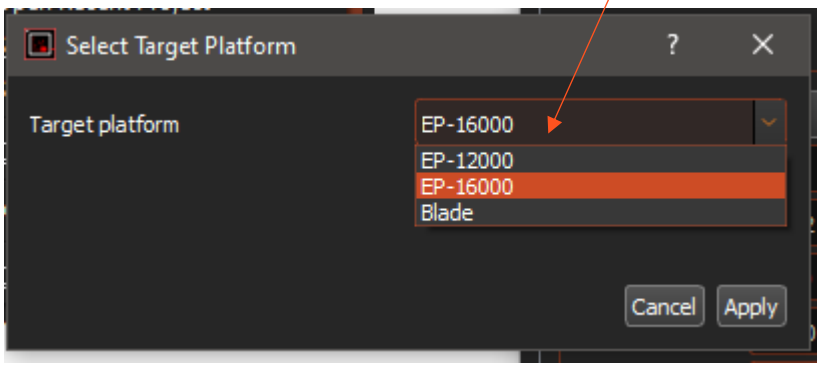
Opens the Project Archive Manager – see description down below in the section.

Tools Menu:

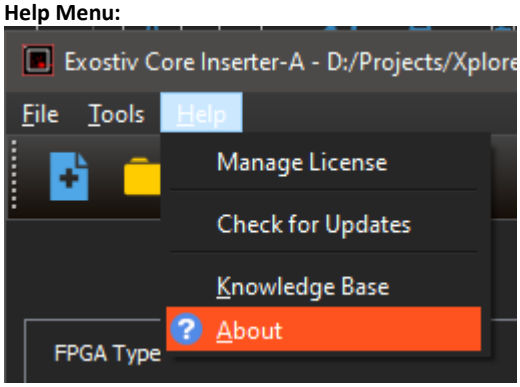


Options… - see description down below in the section

Opens target platform selection controls. Exostiv Core Inserter is foreseen for EP16000 Probes, Exostiv Blade, and migrated EP12000 probes. Please contact us for releases, versions and availability on all platforms.



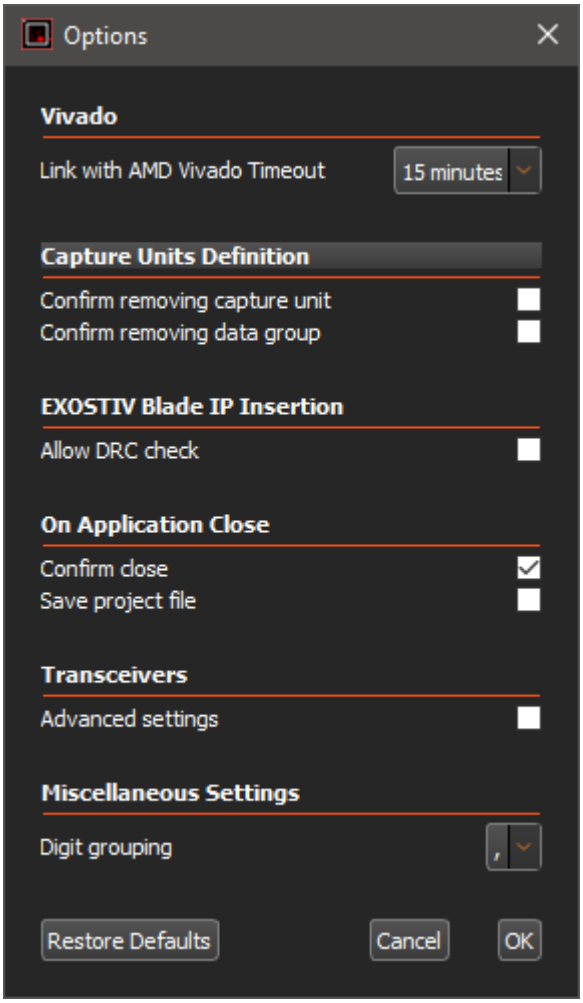
Help Menu:



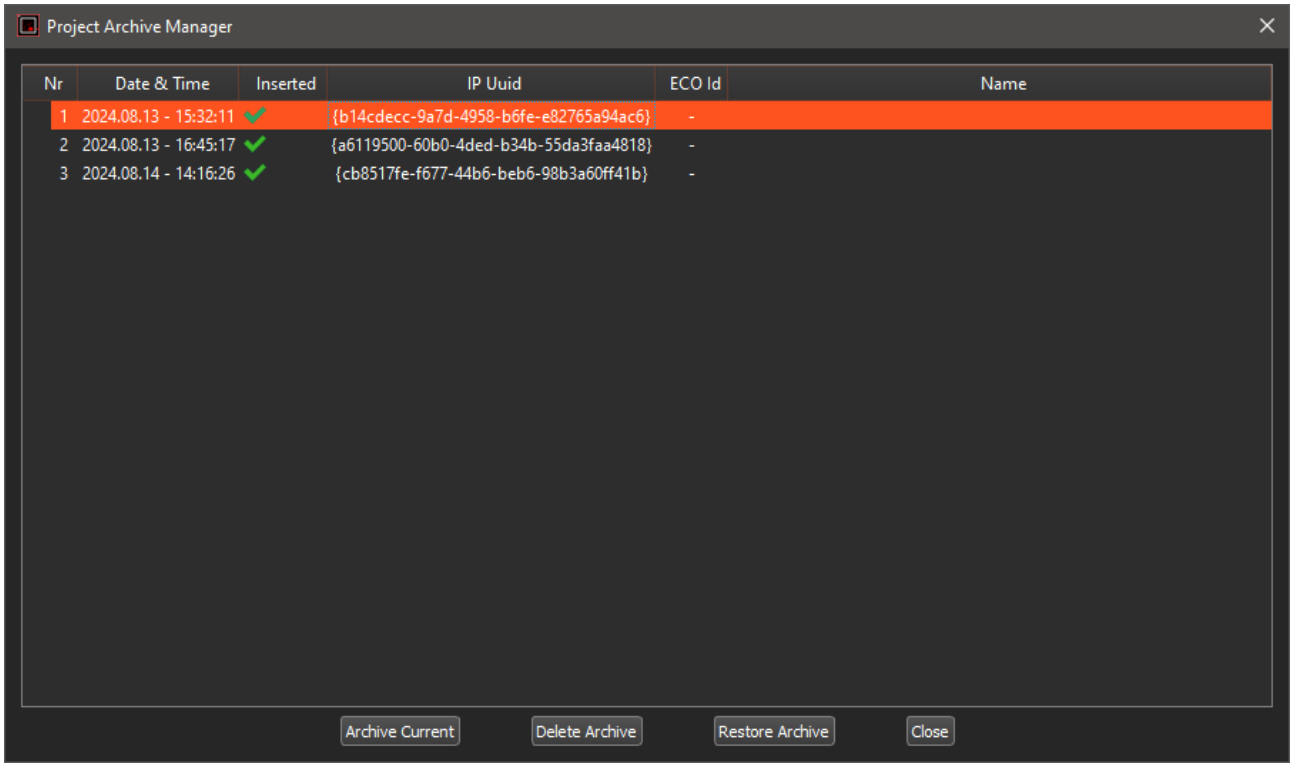
Options: interface and project options

This menu item opens the 'Options' dialog that provides general interface and project options.

Option	Description
Vivado ➔ Vivado Link Timeout	Sets up the timeout for the 'Vivado Link'. When Vivado fails to react to a command from Exostiv Core Inserter after the specified timeout, a dialog opens, prompting for cancelling the command or waiting longer.
Capture Units Definition	Sets up the dialog boxes used for confirming actions.
Exostiv IP Insertion	
On Application Close	
Transceivers	Enables the display of TxDiffCtrl settings for the transceivers. (Advanced – please check: https://www.exostivlabs.com/knowledgebase/what-are-the-transceivers-advanced-settings/)
Miscellanous Settings ➔ **Digit grouping**	Defines the digit / comma separators for numbers in the interface



Project Archive Manager



The 'Project Archive Manager' helps managing, saving and recalling project configurations.

Each generated Exostiv IP core has got an 'Universal Unique Identification' ('Uuid') that enables Exostiv Core Inserter identify an IP once it is inserted inside the target FPGA and check if the settings used in the Dashboard matches it. For instance, Exostiv Core Inserter must make sure that the capture unit settings match these of the loaded IP before it is able to capture data and properly process it.

If the Uuid of the Exostiv IP loaded in the target FPGA does not match the project's Uuid, Exostiv Core Inserter will refuse to establish a connection with the Exostiv IP and won't be able to extract data from the target FPGA.

This ID is defined automatically when the IP is generated.

The Project Archive Manage enables archiving, restoring and deleting project settings. When connecting to an FPGA, it tries to suggest the valid configuration available in the archive that matches the Uuid that is readback from the target instrumented FPGA. It does NOT save and archive the FPGA programming files generated with the FPGA vendor tool after implementation, which should be managed separately.

Core Inserter flows

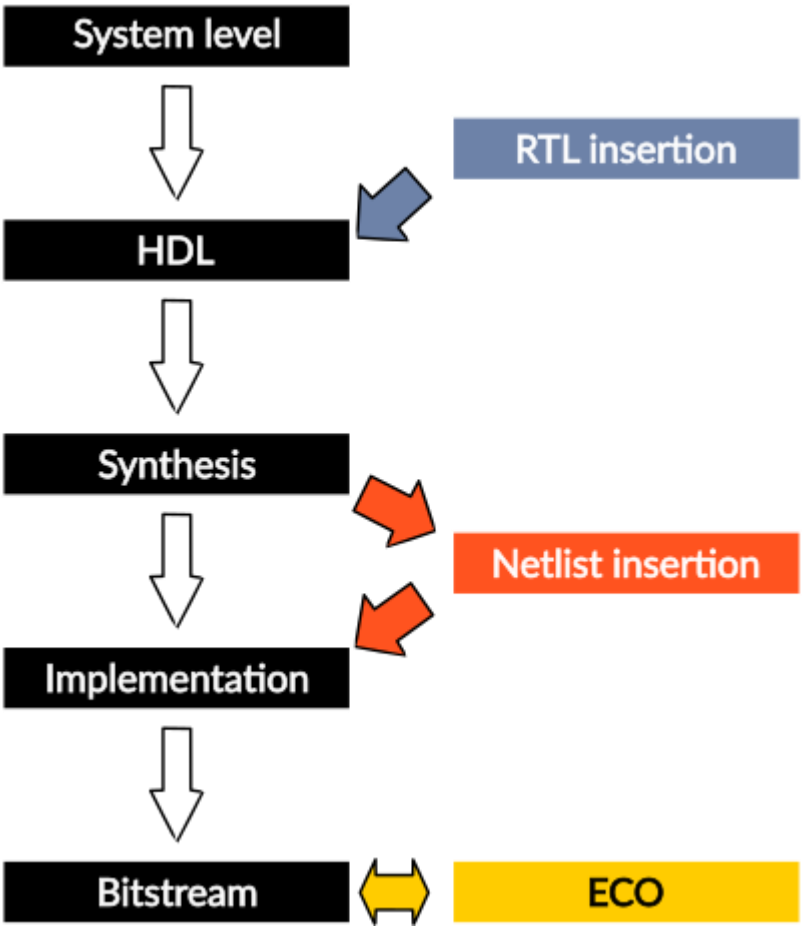
3 IP creation / insertion and modification flows are provided for Exostiv IP :

1. **Netlist flow:**
   In this type of flow, Exostiv IP is configured with Exostiv Core Inserter and inserted into the target design after synthesis. Exostiv IP is not provided as a separate set of files, but directly integrated into the target design netlist by connecting a list of selected nodes to it.

2. **RTL flow:**
   In this type of flow, Exostiv IP is configured with Exostiv Core Inserter and provided to the user as a netlist with a component / module top-level in Verilog or VHDL, together with constraint files and possibly additional example files. The insertion into the target design is done at RTL level by the user.

3. **ECO Flow:**
   The 'ECO flow' (Engineering Change Order flow) allows modifying the connections of a previously inserter IP with the target design under test, after place and route.

The figure below provides an overview of the alternative Exostiv flows.
The Table below sums up the differences between each flow.



Exostiv Core Inserter allows inserting one or multiple Exostiv IP instances into the target design at RTL and netlist levels – and modify the connections to these IPs after implementation (place and route).

1 Netlist flow

In this flow, the Core Inserter synthesizes the Exostiv IP and inserts it automatically into the synthesized target design. Nodes to be captured are selected from the target netlist thanks to an interaction with Vivado. Once the IP s inserted, the design is implemented and the target FPGA bitstream is generated.

For Versal devices, a wrapper must be inserted into the RTL code prior to synthesis.

1.1 Netlist flow for all supported AMD devices, except Versal devices

Using this flow requires using Exostiv Core Inserter ('Core Inserter') and Vivado interfaces.

Netlist flow, step by step (applies to all AMD devices except Versal):

Step nr	Tool	Step	Description
1	Core Inserter	Link configuration	Selects target and defines transceivers setup
2	Vivado	Design synthesis	Sythesizes the target design (without Exostiv IP core) and/or load synthesis or synthesized design check point (DCP)
3	Core Inserter	Capture configuration	Link Core Inserter to Vivado Defines Exostiv IP resources, and input ports. Selects the nodes to be captured interactively from the loaded synthesized design.
4	Core Inserter	Insert Exostiv IP	Calls Vivado to synthesize the configured Exostiv IP. Inserts the synthesized IP into the synthesized target design. Optionally runs implementation (place& route, bitstream generation)
5	Vivado	(Optional): Instrumented design implementation and bitstream generation.	If not done at step 4, runs instrumented design implementation (place and route) and bitstream generation.

1.2 Netlist flow for Versal devices

The Versal flow for the IP insertion in netlist mode differs from the other flow used for the other AMD/Xilinx FPGA families. Even if the Exostiv IP core is inserted and connected to the user's design nets directly in the synthesized netlist, the transceiver bank used by the Exostiv IP must be inserted at RTL level in the user's design.

The Exostiv IP generation and insertion is now performed in two steps.

1. Generation of the Exostiv IP top level wrapper and the transceiver link.

   This wrapper must be inserted in the user's design at RTL level. HDL templates ('.vo' and '.vho' files) are generated to help the user insert the IP top level wrapper in its code. When the wrapper is inserted in the user's HDL, the generated TCL script to add generated files to the user's design must be sourced from Vivado TCL console. A script is generated for project mode and for non-project mode Vivado usage.

   Please select the script matching your case. This script generates a block design containing the transceiver link used by the Exostiv IP and adds all the required HDL files to the user's design. At this point, the Exostiv IP top level contains two modules, the transceiver link and a dummy version of the IP core. This dummy core will be replaced by the real IP core during the netlist insertion step.

   The synthesis can then be launched.

2. Generation and insertion of the Exostiv IP core.
   This step is performed after synthesis of the user's design. After completing the core configuration and the selection of the nets to observe, the IP core is generated. It is then inserted in the user's design netlist. Selected nets are automatically connected to the IP core. Implementation can then be launched.

Files generated by the Versal netlist flow:

File	Description / Usage
Verilog files	
<instance_name>_core_dummy.v	Wrapper of the dummy IP core.
<instance_name>_gt_wrapper.v	Wrapper for the transceiver link.
<instance_name>_wrapper.v	Wrapper for the IP top level.
<instance_name>_wrapper.vo	Template for top level insertion in user's design.
VHDL files	
<instance_name>_core_dummy.vhd	Wrapper of the dummy IP core.
<instance_name>_gt_wrapper.vhd	Wrapper for the transceiver link.
<instance_name>_wrapper.vhd	Wrapper for the IP top level.
<instance_name>_wrapper.vho	Template for top level insertion in user's design.
Constraint Files	
<instance_name>_wrapper_pinout.xdc	Constraint file defining the transceiver pins mapping and reference clock.
Scripts	
Generate_gt_link_bd.tcl	Script generating the transceiver's block design.
Non_project_mode_read_files.tcl	Script to add all the generated files to the user design when non-project mode is used.
Project_mode_add_files.tcl	Script to add all the generated files to the user design when project mode is used.

Note:
When the files are generated by the Core Inserter, the tag <instance_name> used in the table hereabove is replaced by the instance name defined by the user in the application.

Versal netlist flow - step by step:

Step nr	Tool	Step	Description
1	Core Inserter	Link configuration	Selects target and defines transceivers setup
2	Core Inserter	Exostiv IP Wrapper generation	Generates RTL IP Wrapper for Versal devices
3	A code editor + Vivado	Insert the generated Exostiv IP Wrapper into the target design	To enable netlist insertion with versal device, the generated wrapper must be inserted into the target design and a script adds the required files to the Vivado project.
4	Vivado	Design synthesis	Synthesizes the target design with the generated wrapper. This wrapper instance allows the subsequent automatic insertion of the Exostiv IP.
5	Core Inserter	Capture configuration	Links Core Inserter to Vivado. Defines Exostiv IP resources, and input ports. Selects the nodes to be captured interactively from the loaded synthesized design.
6	Core Inserter	Insert Exostiv IP	Calls Vivado to synthesize the configured Exostiv IP. Inserts the synthesized IP into the synthesized target design. Optionally runs implementation (place& route, bitstream generation)
7	Vivado	(Optional): Instrumented design implementation and bitstream generation.	If not done at step 6, runs instrumented design implementation (place and route) and bitstream generation.

2 RTL flow

2.1 RTL flow for all supported AMD devices except Versal devices

In this flow, The Core Inserter generates produces the Exostiv IP as a synthesized netlist with constraints, template and scripts. The target design RTL code must be edited to insert the generated IP and connect the nodes to be captured to exostiv IP. Reference templates are provided for the insertion.

The instrumented target design must then be synthesized and implemented from the vendor tool.

Files generated by the RTL flow:

File name	Description	Usage
<instance name>_wrapper.edf	EXOSTIV IP synthesized netlist.	Add to Vivado project for synthesis / P&R in VHDL or Verilog flow.
<instance name>_wrapper_pkg.vhd	VHDL package containing the types used for the EXOSTIV IP instantiation.	Add to Vivado project for synthesis / P&R in VHDL flow.
<instance name>_wrapper_module.v	Verilog module for EXOSTIV IP netlist	Add to Vivado project for synthesis / P&R in Verilog flow.
<instance name>_wrapper_pinout.xdc	Constraint file for EXOSTIV IP pinout. (like transceivers location).	Add to Vivado project for synthesis / P&R in VHDL or Verilog flow.
<instance name>_wrapper.xdc	Constraint file with the timing constraints of EXOSTIV IP.	Add to Vivado project but do not use for synthesis. You have to 'scope' this file on the Exostiv IP wrapper. Here are the commands: — Case 1: project mode: add_files -fileset constrs_1 -norecurse <file_path>/<instance_name>_wrapper.xdc set_property SCOPED_TO_REF <instance name>_wrapper [get_files <file_path>/<instance_name>_wrapper.xdc] set_property used_in_synthesis false [get_files <file_path>/<instance_name>_wrapper.xdc] — Case 2: non-project mode: read_xdc -ref <instance name>_wrapper <file_path>/<instance_name>_wrapper.xdc set_property used_in_synthesis false [get_files <file_path>/<instance_name>_wrapper.xdc]
<instance name>_wrapper.vho	Example template on how to create a VHDL instance of EXOSTIV IP in the target design.	Open with a text editor and follow the provided example.
<instance name>_wrapper.vo	Example template on how to create a Verilog instance of EXOSTIV IP in the target design.	Open with a text editor and follow the provided example.

RTL flow, step by step (applies to all AMD devices, except Versal devices):

Step nr	Tool	Step	Description
1	Core Inserter	Link configuration	Selects target and defines transceivers setup
2	Core Inserter	Capture configuration	Defines Exostiv IP resources, and input ports.
3	Core Inserter	Generate Exostiv IP	Calls Vivado to generate the configured Exostiv IP. Saves synthesized IP, VHDL/Verilog module, constraints and scripts to an output directory.
4	A code editor	Design update	Target design update to instantiate the generated Exostiv IP.
5	Vivado	Synthesis, implementation, bitstream generation	Instrumented design generation.

2.2 RTL flow for AMD Versal devices

The RTL flow is very similar to the first step of the netlist flow.
Instead of generating a dummy wrapper for the IP core, an EDIF netlist is generated. Using the generated templates, the user must insert the IP top level wrapper in its design at HDL level. In RTL mode, the nets to observe must be connected to the Exostiv IP wrapper. When the RTL insertion is done, the user must run the script corresponding to its Vivado use case (project mode or non-project mode) to generate the transceiver block design and add all generated files to its design. The RTL project is instrumented with the Exostiv IP core. Synthesis and implementation can be launched.

Files generated by the Versal RTL flow:

File	Description / Usage
Verilog files	
<instance_name>_gt_wrapper.v	Wrapper for the transceiver link.
<instance_name>_wrapper.v	Wrapper for the IP top level.
<instance_name>_wrapper.vo	Template for top level insertion in user's design.
VHDL files	
<instance_name>_gt_wrapper.vhd	Wrapper for the transceiver link.
<instance_name>_wrapper.vhd	Wrapper for the IP top level.
<instance_name>_wrapper.vho	Template for top level insertion in user's design.
Netlist	
<instance_name>_core_wrapper.edf	EDIF netlist of the Exostiv IP Core
Constraint Files	
<instance_name>_wrapper_pinout.xdc	Constraint file defining the transceiver pins mapping and reference clock.
<instance_name>_core_wrapper.xdc	Timing constraints for the IP core netlist. This constraint must be scoped to the core wrapper module.
Scripts	
Generate_gt_link_bd.tcl	Script generating the transceiver's block design.
Non_project_mode_read_files.tcl	Script to add all the generated files to the user design when non-project mode is used.
Project_mode_add_files.tcl	Script to add all the generated files to the user design when project mode is used.

Versal RTL flow, step by step:

Step nr	Tool	Step	Description
1	Core Inserter	Link configuration	Selects target and defines transceivers setup
2	Core Inserter	Capture configuration	Defines Exostiv IP resources, and input ports.
3	Core Inserter	Generate Exostiv IP	Calls Vivado to generate the configured Exostiv IP. Saves synthesized IP, VHDL/Verilog module, constraints and scripts to an output directory.
4	A code editor	Design update	Target design update to instantiate the generated Exostiv IP.
5	Vivado	Call script generated with the code inserter.	Adds all the generated files to the user design in either project or non-project mode.
6	Vivado	Synthesis, implementation, bitstream generation	Instrumented design generation.

3 ECO flow

Available for all designs with an inserted Exostiv IP.

Step nr	Tool	Step	Description
0	(Potentially multiple tools)	Complete the generation of a design instrumented with one or multiple Exostiv IP instances	Complete implementation – see previous descriptions of flows.
1	Vivado	Load design checkpoint or project after place and route with a Exostiv IP instance	ECO flow allows modifications after P&R. This step of the design must be loaded into Vivado.
2	Core Inserter	Select ECO mode and establish link with Vivado (and the loaded implemented design)	This step allows making the ECO.
3	Core Inserter	Run ECO	This steps calls Vivado to generated the modified bitstream.

Core Inserter : create a project

When creating a new project, the target platform and the type of flow must be selected. Available choices:

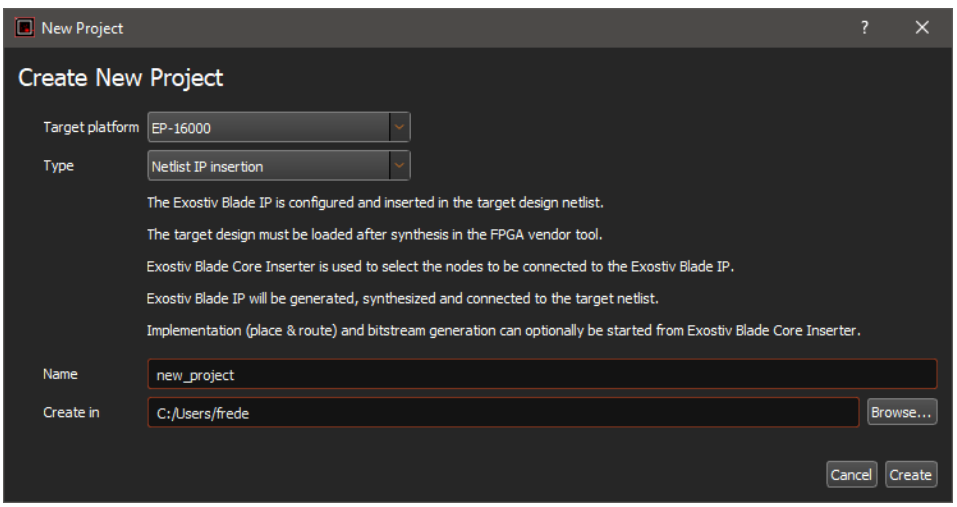
Platform

- **EP12000 :** EP12000 legacy Exostiv Probe. Requires EP12000 to be upgraded to software environment v2 (contact us).
- **EP16000 :** EP16000 Exostiv Probe.
- **Exostiv Blade :** Exostiv Blade unit.

Flow

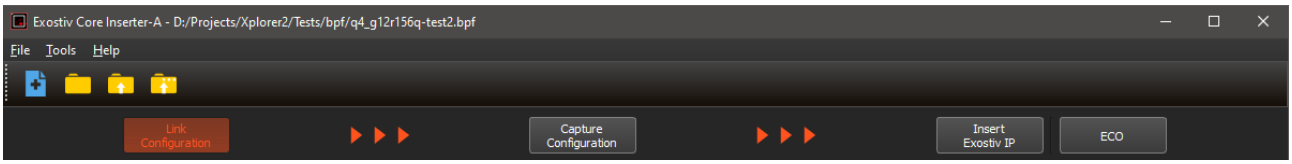
- **Netlist insertion flow**
- **RTL flow.**

The ECO ('Engineering Change Order') flow is not an IP 'creation' flow – and hence can be used only after an IP is first created and inserted.

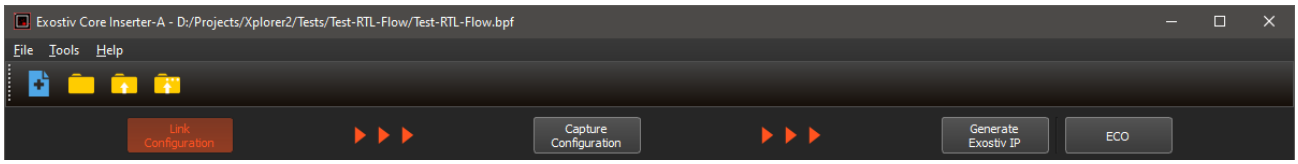


Core Inserter - Overview

The Core Inserter is made of multiple screens to be used sequentially. This sequential flow is summarized in the top area of the Core Inserter. Clicking in each of the flow steps switches the display to a specific page. The top flow bar depends on the chosen Core Inserter flow (RTL flow or netlist flow).

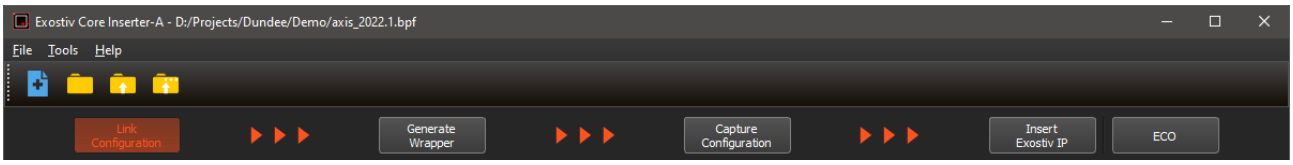


Flow bar in 'Netlist Flow'



Flow bar in 'RTL flow'

When targeting Versal devices in netlist mode of insertion, the sequential flow features one additional step ('Generate Wrapper'), as shown below:



Flow bar in 'Netlist Flow' for Versal devices

Schematically, the following steps must be followed to set up and insert Exostiv IP ('Core') in a target design:

1) **Link Configuration: sets up the link between Exostiv IP the hardware.**
This step is used to define the parameters of the physical link between the target FPGA and the chosen hardware (Exostiv Probe or Exostiv Blade): transceivers parameters, pin locations, type of interface, data rates and so on.

2) **Generate Wrapper**: (Versal devices in netlist mode of insertion only). **Generates the design wrapper required to properly insert the IP in Versal devices in netlist flow.**

3) **Capture Configuration: sets up the capture inside the target FPGA.**
This step is used to set up Exostiv IP 's connections with the internal logic of the target FPGA. It also defines the resources reserved for the IP to sample data, define triggering and filtering events and locally store trace data.

   If the 'Netlist IP Insertion flow' is used, a connection with the FPGA vendor design tool is required.

   If the 'RTL IP Insertion flow' is used, this step is used to define the Exostiv IP structure and resources (FIFO size, trigger unit complexity, number of ports, …)

4) **Insert Exostiv IP (Netlist flow): synthesizes Exostiv IP, inserts it into the target design and run the instrumented design implementation.**
During this step, the Core Inserter calls the FPGA vendor tool to synthesize the IP, insert it into the target FPGA and connect it to the target FPGA nodes chosen at step 2). Thereafter, Core Inserter runs the implementation of the target design instrumented with the Exostiv IP.

During this step, Exostiv Core Inserter calls the FPGA vendor tool to synthesize Exostiv IP , generate a HDL (VHDL or Verilog) top-level file and constraint files. These files must be used to instrument the target design RTL source code: Exostiv IP should then be instantiated into the target code and the instrumented design should be synthesized and implemented with the FPGA vendor tool.

Once the full core inserter flow is over, the target FPGA can be programmed with the generated programming files. Thereafter, the client application corresponding to the chosen hardware (Exostiv Probe Client or Exostiv Blade Client) can be used to capture data (refer to the documentation relative to the client applications).

Core Inserter – Linking to FPGA vendor tool in 'Netlist flow'

(Applies to 'Netlist flow Insertion' – Core Inserter-A, used for AMD FPGA targets.)

Some of the Core Inserter functionalities - such as selecting the IP connections to the target FPGA nodes or selecting the sampling clock of a capture unit – work based on queries sent by Exostiv Core Inserter to the FPGA Vendor tool (AMD Vivado).
*This link is used in **'Netlist flow'** only.*

To use these features, a link must be established between Exostiv Core Inserter and the FPGA Vendor tool.

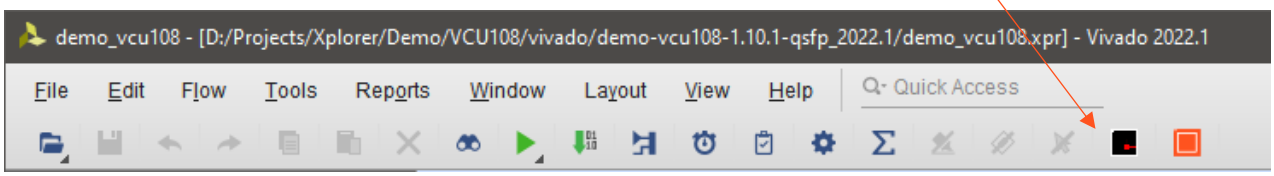
To establish a link between Exostiv Core Inserter and Vivado, please proceed as follows:

1. **Open Vivado and open the project containing the target FPGA design.**

2. **From the flow navigator in Vivado, open synthesized design or a design checkpoint saved after the synthesis of your design.**

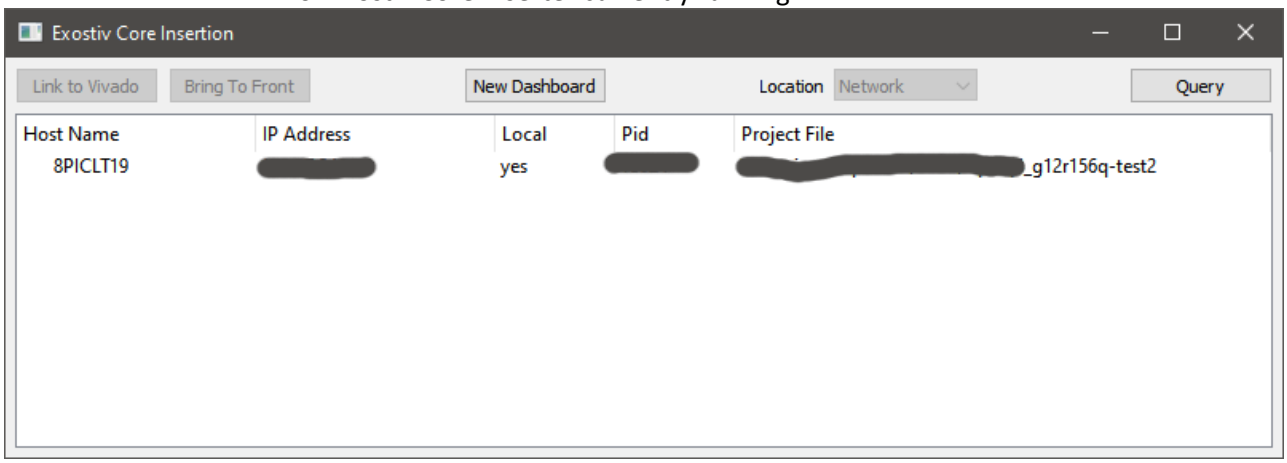
   If the design was not synthesized, you'll have to synthesize it.

3. **Click on the 'Open Exostiv Core Inserter' icon in the toolbar.**
   This shortcut is installed with Exostiv Core Inserter.
   Can't find it? Refer to Error! Reference source not found.



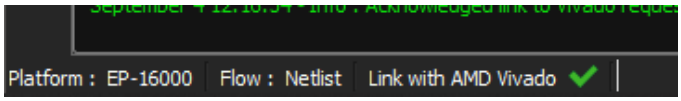
If you cannot find it, you can install it from the Exostiv Core Inserter menu:
Tools > Install Dashboard Shortcut
You'll have to shut down Vivado and restart from 1.

4. **After clicking on the shortcut, the following window appear**s, listing the instances of Exostiv Core Inserter currently running:



5. **Select the desired instance and click on 'Link to Vivado'.**

6. **Switch back to the Exostiv Core Inserter**. You are now connected to Vivado with Exostiv Core Inserter, as it can be seen from the bottom status bar.



An alternative way to come to the same situation consists in
1) opening the project in Vivado and loading the post-synthesized checkpoint
2) clicking on the Exostiv Core Inserter shortcut

Exostiv Core Inserter starts automatically and links to Vivado. Thereafter, the right .bpf project has to be loaded. **In this case, the 'link to Vivado' window above does not pop up.**

Core Inserter – Linking to FPGA vendor tool in 'ECO'mode.

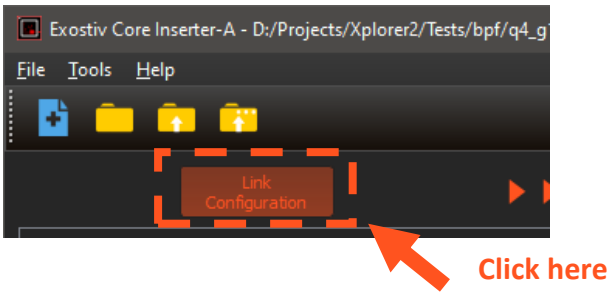
The 'ECO' mode allows modifying a previously inserted Exostiv IP after place and route.
To use it:
- Load the implemented design in Vivado, that contains the previously inserted Exostiv IP
- Alternatively, if saved, load the design checkpoint (DCP) after P&R
- Proceed as described in the previous section to establish a link between Vivado and the Core Inserter application. From there, the controls of the 'ECO' mode of operation are fully available.

Please refer to section Core Inserter – ECO mode below for more details.

Core Inserter – Link Configuration

Link Configuration: locate and access

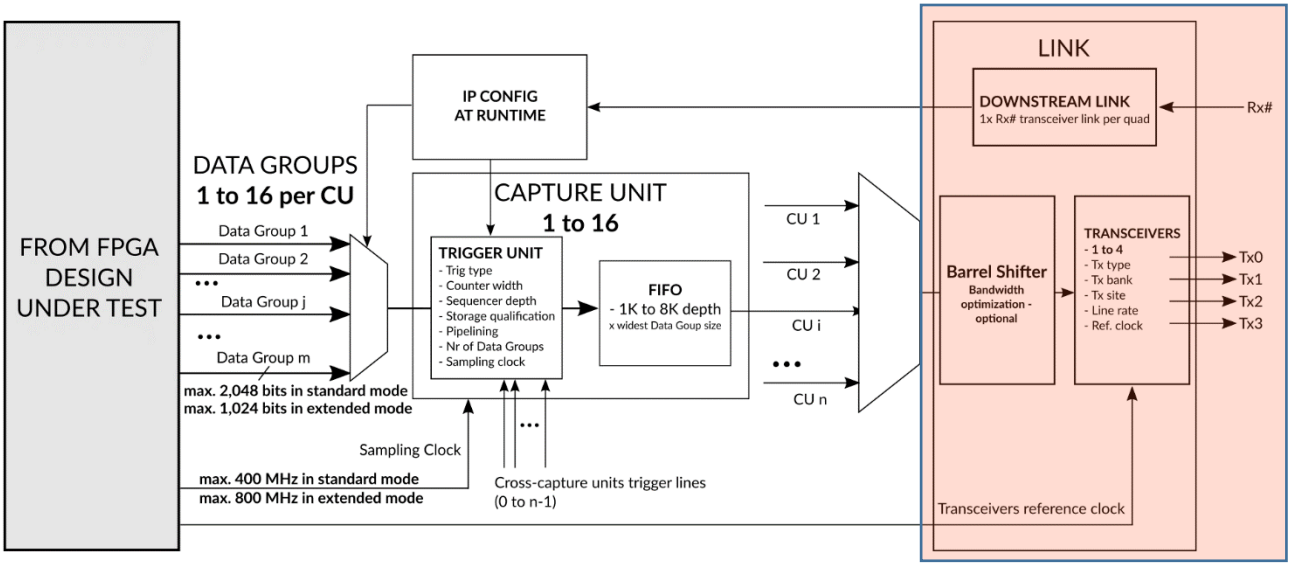
- Click on the 'Link Configuration' button of the top bar of the Core Inserter.

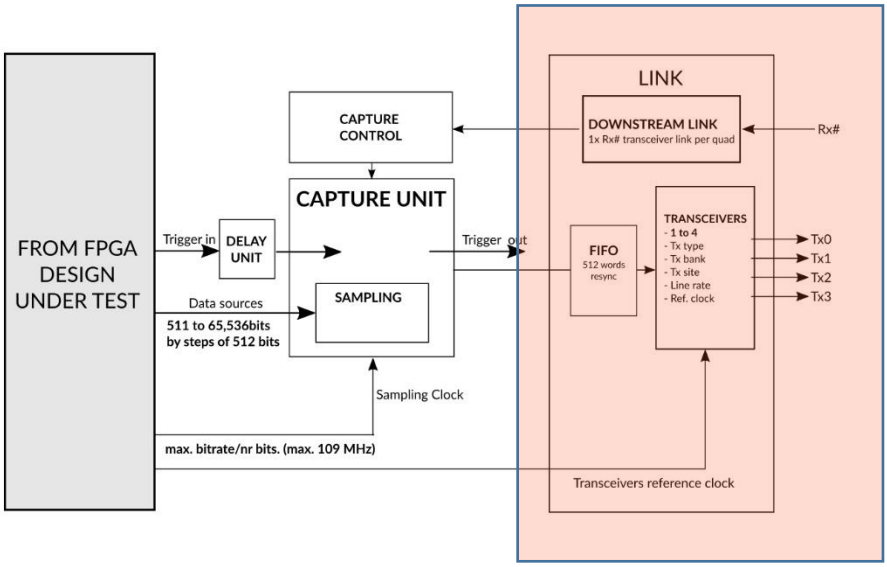


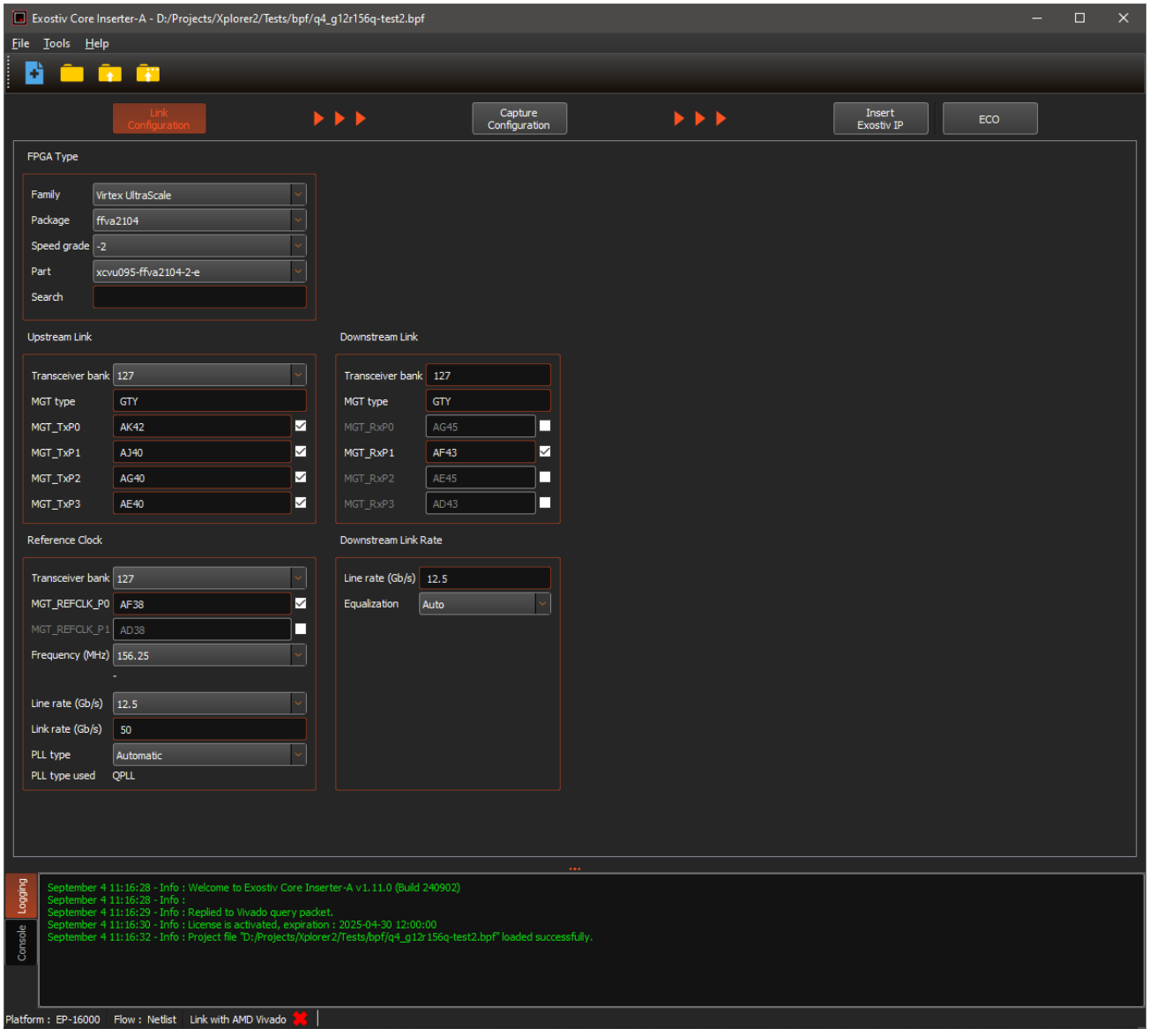
Link Configuration: overview

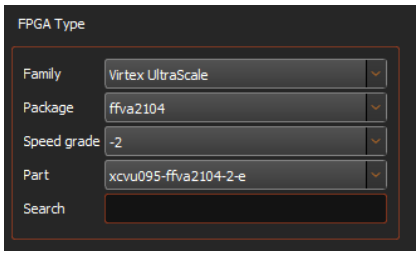
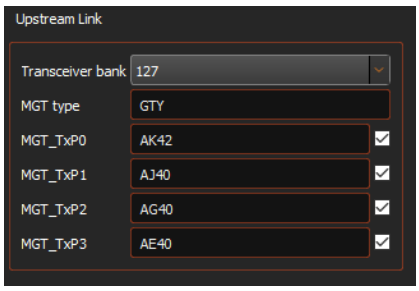
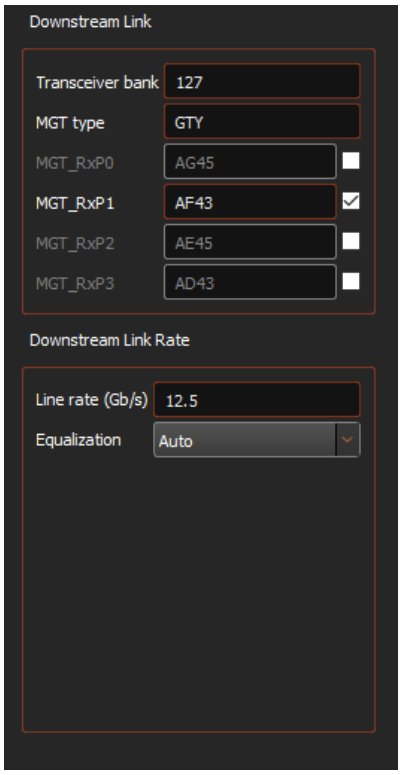
The 'Link Configuration window' is used to configure the interface between Exostiv IP and Exostiv Probe. It sets up the transceivers. Please refer to **Exostiv IP user's guide** for a detailed description of the IP's parameters.

Link Configuration: parameters




FROM FPGA DESIGN UNDER TEST
CAPTURE CONTROL
CAPTURE UNIT
Trigger in
DELAY UNIT
Trigger out
Data sources
511 to 65,536bits by steps of 512 bits
SAMPLING
Sampling Clock
max. bitrate/nr bits. (max. 109 MHz)
LINK
DOWNSTREAM LINK
1x Rx# transceiver link per quad
Rx#
FIFO
512 words resync
TRANSCEIVERS
- 1 to 4
- Tx type
- Tx bank
- Tx site
- Line rate
- Ref. clock
Tx0
Tx1
Tx2
Tx3
Transceivers reference clock


Exostiv Core Inserter-A - D:/Projects/Xplorer2/Tests/bpf/q4_g12r156q-test2.bpf
File Tools Help
Link Configuration
Capture Configuration
Insert Exostiv IP
ECO
FPGA Type
Family Virtex UltraScale
Package ffva2104
Speed grade -2
Part xcvu095-ffva2104-2-e
Search
Upstream Link
Transceiver bank 127
MGT type GTY
MGT_TxP0 AK42
MGT_TxP1 AJ40
MGT_TxP2 AG40
MGT_TxP3 AE40
Downstream Link
Transceiver bank 127
MGT type GTY
MGT_RxP0 AG45
MGT_RxP1 AF43
MGT_RxP2 AE45
MGT_RxP3 AD43
Reference Clock
Transceiver bank 127
MGT_REFCLK_P0 AF38
MGT_REFCLK_P1 AD38
Frequency (MHz) 156.25
Line rate (Gb/s) 12.5
Link rate (Gb/s) 50
PLL type Automatic
PLL type used QPLL
Downstream Link Rate
Line rate (Gb/s) 12.5
Equalization Auto
Logging
Console
September 4 11:16:28 - Info : Welcome to Exostiv Core Inserter-A v1.11.0 (Build 240902)
September 4 11:16:28 - Info :
September 4 11:16:29 - Info : Replied to Vivado query packet.
September 4 11:16:30 - Info : License is activated, expiration : 2025-04-30 12:00:00
September 4 11:16:32 - Info : Project file "D:/Projects/Xplorer2/Tests/bpf/q4_g12r156q-test2.bpf" loaded successfully.
Platform : EP-16000
Flow : Netlist
Link with AMD Vivado

Control group		Description
FPGA Type		Use these controls to define the target FPGA family, device, speed grade and package.
Upstream link	 **Upstream link transceivers choice grouped by quad.**	Use these controls to set up the 'Upstream link' – that is the link between Exostiv IP and the chosen hardware. This link uses from 1 to 4 transceivers. The transceivers are designated from their I/O bank on the FPGA and the I/O site of their 'P' channel. Please refer to the chosen FPGA pinout and its connections on the target board. At IP insertion, there is no restriction on the number of transceivers. All the transceivers used for a single IP instance must be located in the same FPGA quad. **Transceiver bank:** select the I/O bank of the target FPGA that is reserved for use with Exostiv **MGT_TxP#** : click on the corresponding tick box to use this transceiver in Exostiv IP . The I/O site for the P pin is derived from the 'transceiver bank' choice above.
Downstream link		Exostiv IP has got a downstream link that is used to control the IP during analysis (e.g.: to change the trigger condition at runtime.). This downstream link uses one of the transceivers, selected here. Additional advanced settings for the downstream link can be found here as well. Please refer to UG201 : Interfacing Exostiv Probe EP Series for a detailed description of the pin allocation and the required physical connections.

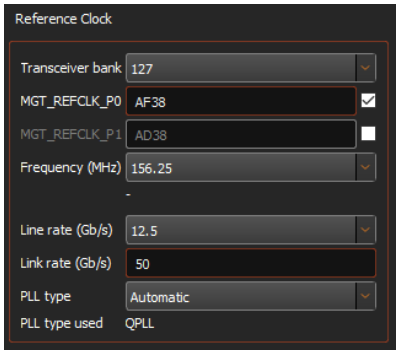
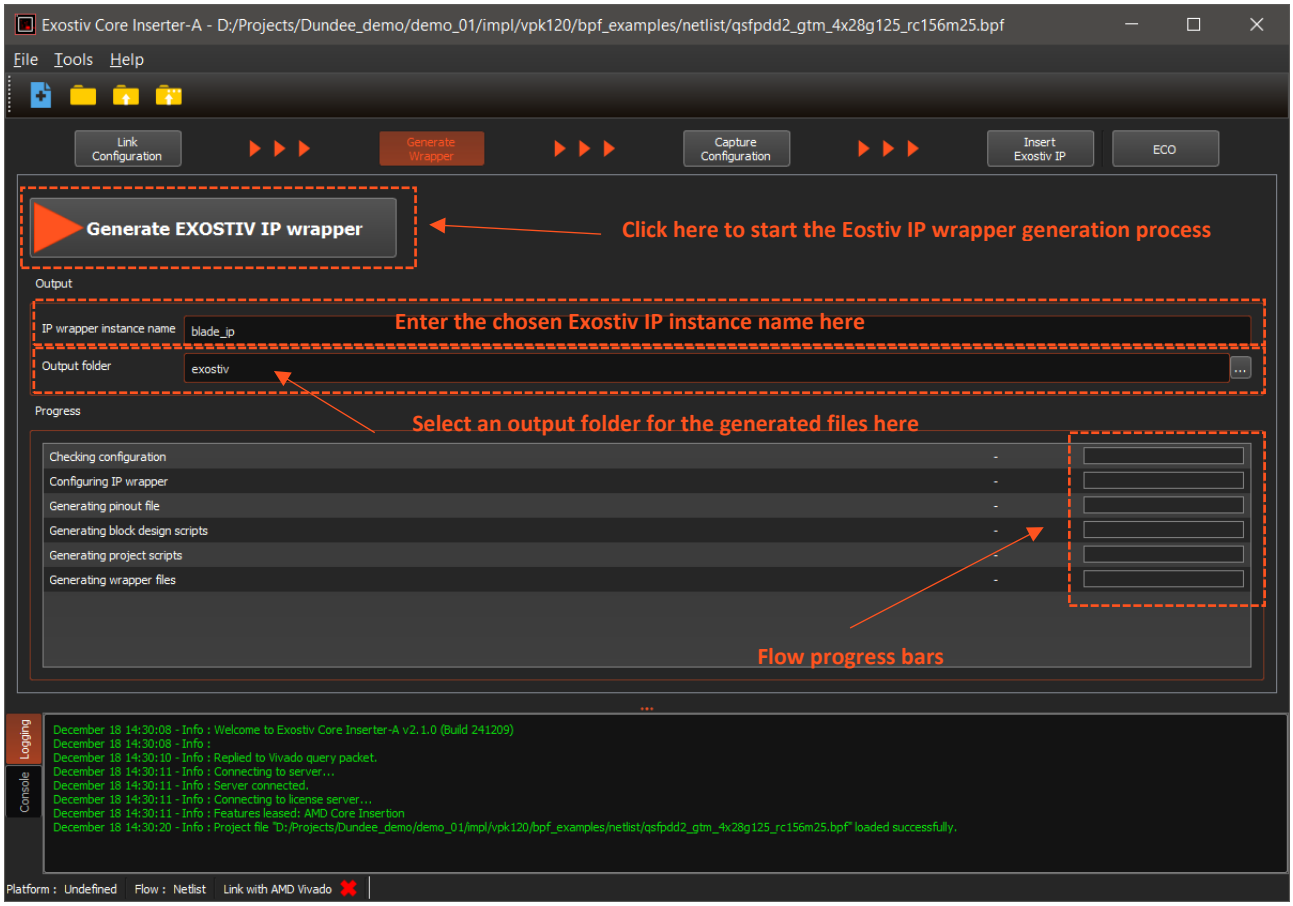
Reference clock	 **Reference clock controls for AMD FPGA**	This group of control sets up the transceivers' reference clock. The chosen clock depends on the board design. Specify the location it is taken from and the clock frequency. Based on the specified frequency, the desired data rate (line rate) can be derived. The available line rates in the list depend on the reference clock frequency, the target FPGA capabilities, and the chosen Exostiv hardware (EP12000 probe, EP16000 probe or Exostiv Blade). Additionally, Exostiv Core Inserter provides controls to force the usage of the quad QPLL vs. CPLL.

Table 1: Link Configuration parameters

Core Inserter : Generate EXOSTIV IP Wrapper



This page is only available for IP generated for Versal devices in netlist mode of insertion.

Based on the Link Configuration settings, a series of files are generated during this process, that need to be used with the target design to properly instantiate the transceivers used by the exostiv Ip in the Versal device.

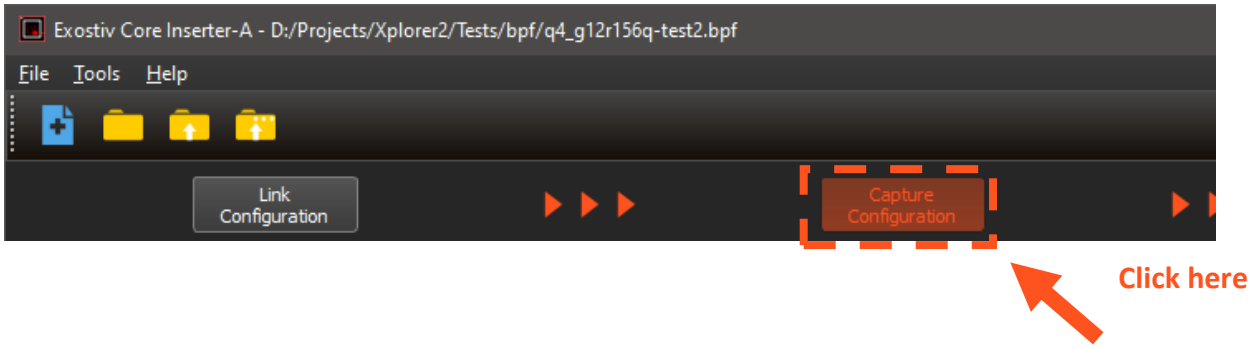
To proceed:

1. Specify an instance name for the wrapper.
2. Select an output folder.
3. Click on 'Generate EXOSTIV IP Wrapper' button

Progress bars provide the status of this process. Once over, the following files are available in the chosen output folder:
(files)

Core Inserter – Capture Configuration

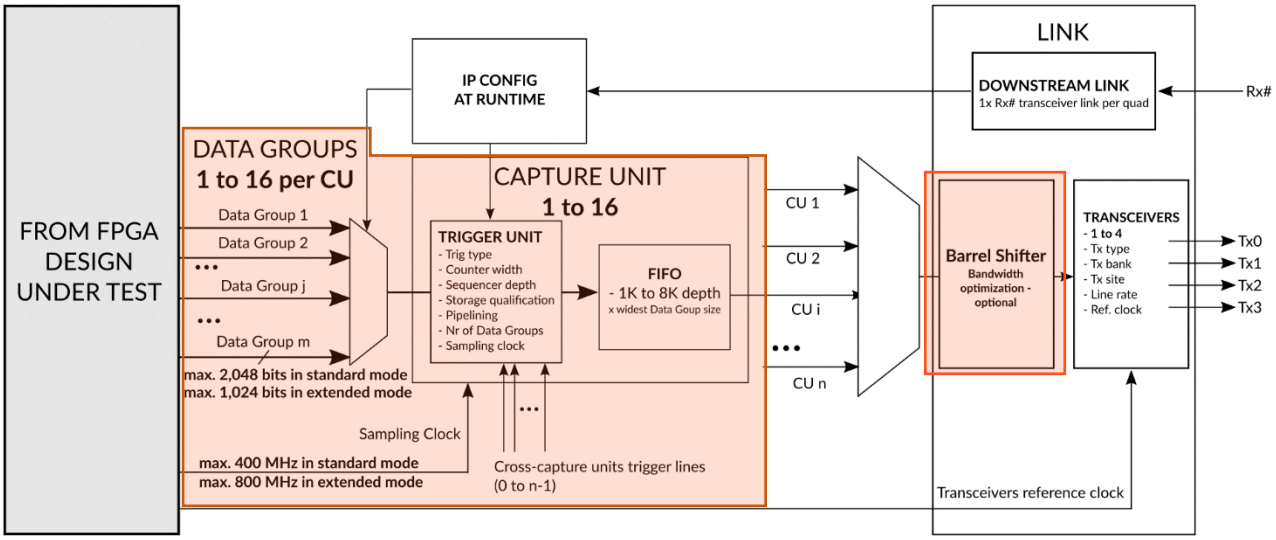
Capture Configuration: locate and access



Capture Configuration: overview

The 'Capture Configuration window' is used to set up the Exostiv IP 's 'capture' logic.
It is used to:

- Select the IP type that has to be generated and/or inserter: 'Standard IP' vs. 'Extended width IP'. According to the chosen IP type, the provided options differ.
- Define the number of nodes from the target FPGA that will be sampled and observed.
- If netlist mode is used, select the nodes to be observed from the target netlist;
- Define the resources allocated to the Exostiv IP to detect a trigger condition, sample data from the target FPGA and to send it outside the FPGA to Exostiv Probe.
- Specify the sampling clock parameters for each capture unit and parameters related to bandwidth usage.



Standard IP Capture Configuration: adding & configuring Capture Units

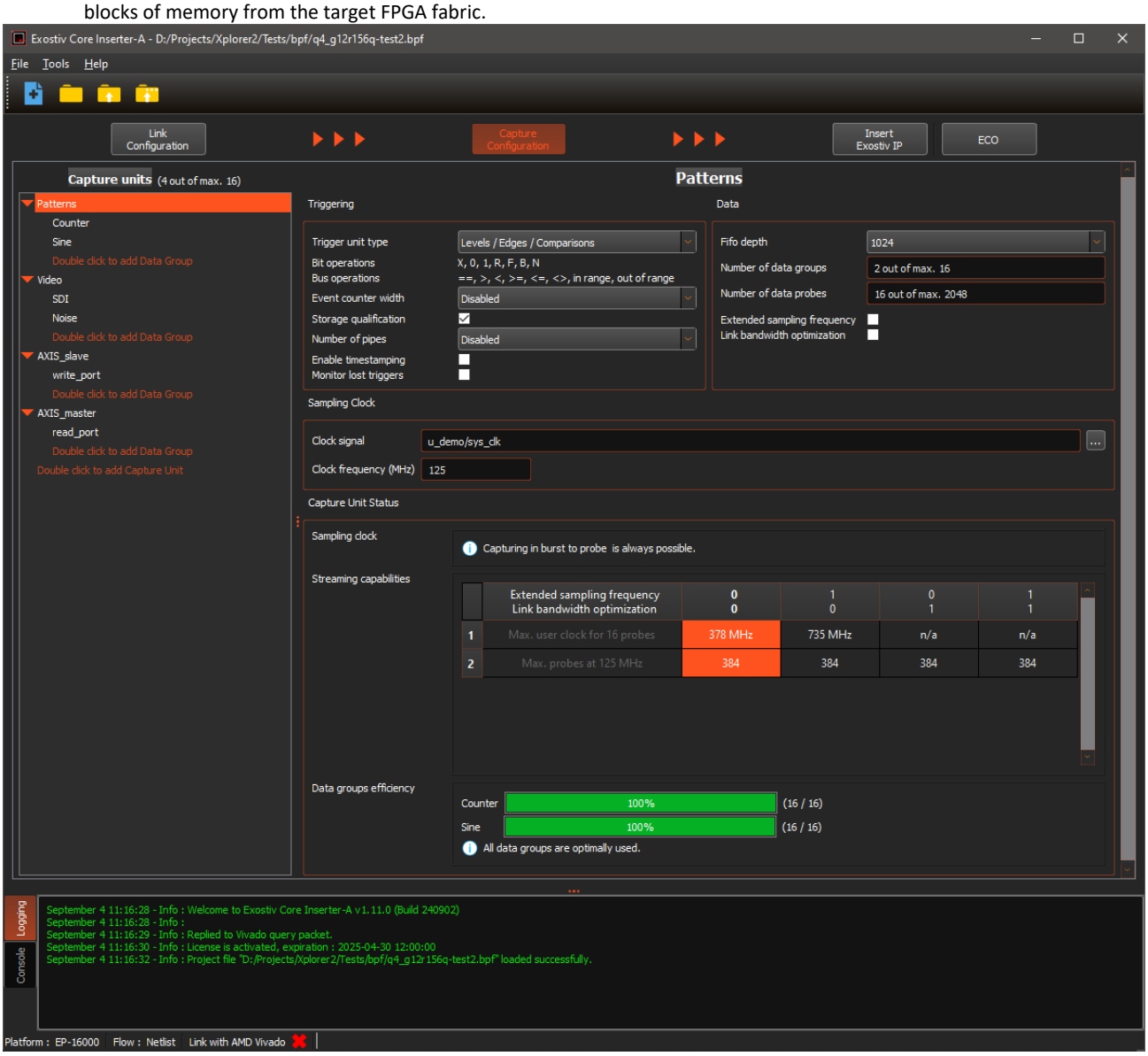
A 'Capture Unit' is a logic unit that samples data from the target FPGA and sends it to the external hardware.
Each Capture Unit:

- **is connected to 1 to 16 'Data Group(s)'**.
  The Data Groups are multiplexed and the user can switch from one data group to another during analysis without having to recompile the Exostiv IP . Each Data Group can be connected to up to 2,048 FPGA internal logic nodes.

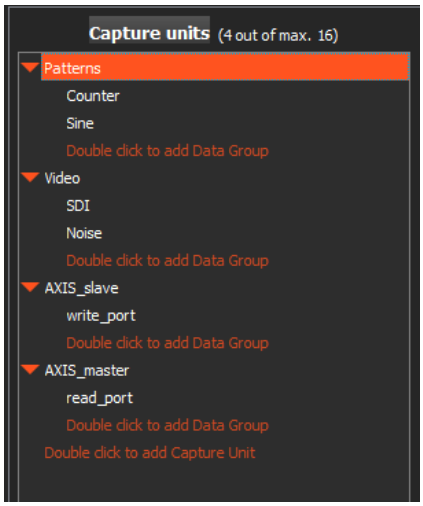
- **uses 1 single clock for sampling data**.
  This clock is chosen from the clocks available internally in the target FPGA. If Exostiv must collect data **from more than one clock domain**, a separate capture unit has to be defined for each clock domain.
  Multiple capture units can use the same reference clock for sampling data.

- **includes logic for triggering and data qualification.**
  This logic is used to detect the conditions for starting a capture and for filtering the data.

- **includes one FIFO**, that is used for buffering data before sending to the transceivers. This FIFO is implemented by using blocks of memory from the target FPGA fabric.

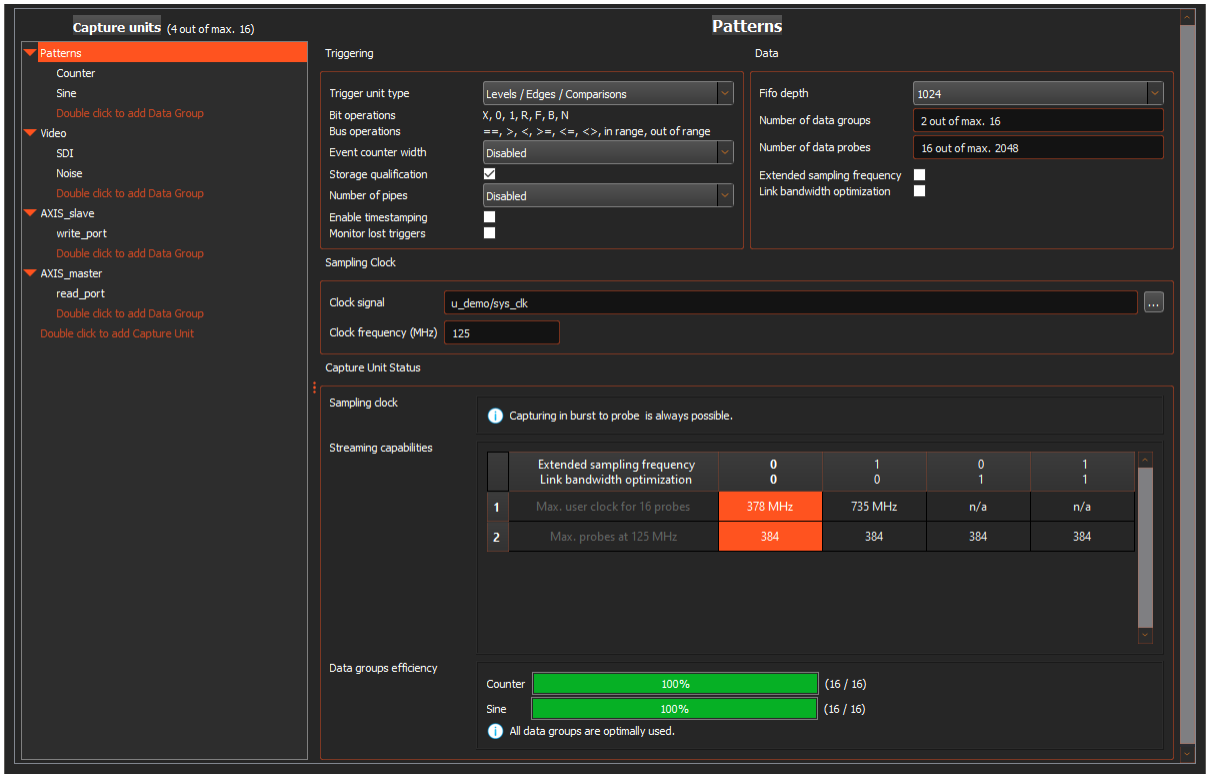


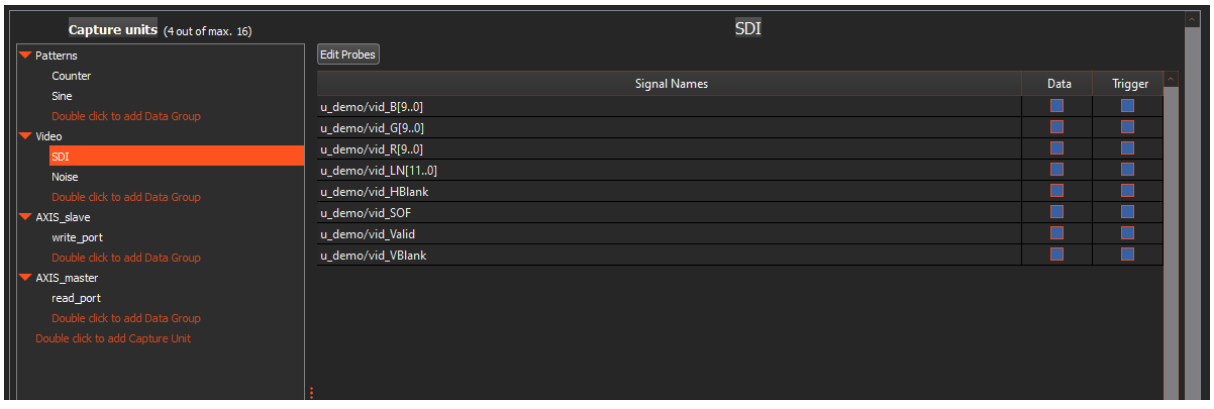
To add or remove capture units and/or data groups: use the controls provided in the 'Capture Unit' section.

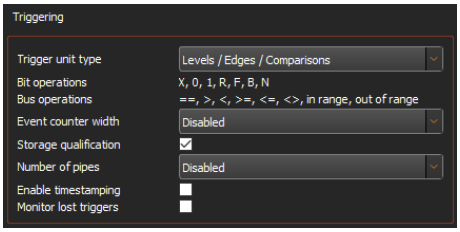


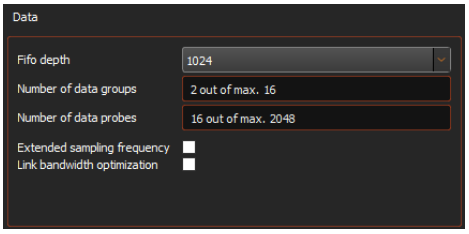
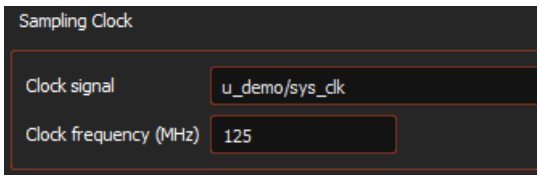
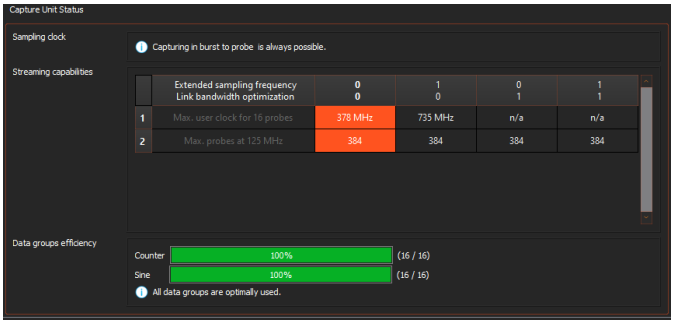
To set up a capture unit or a data group:

Click on the capture unit or the data group that you wish to set up. The Dashboard controls switch to the selected element's settings page.

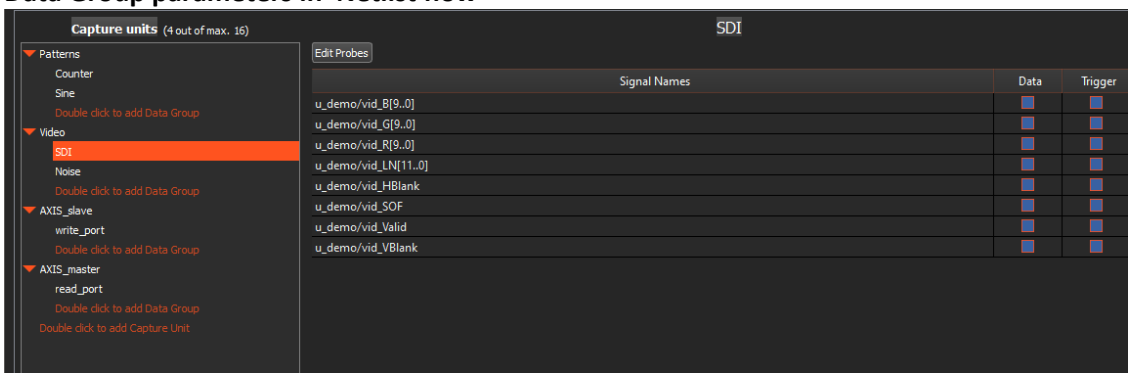




Control group		Description
Triggering		**Defines the trigger settings for the selected capture unit.** - **Trigger unit type:** use this drop-down list to select between 'Levels / Edges / Comparisons' or 'Levels / Edges' only. This setting sets the complexity of the trigger unit by enabling / disabling 'comparison operators. 'Levels / Edges / Comparisons' requires more FPGA resources. The available operations are summarized under the drop-down control. - **'Event Counter width'** specifies the width of the events counter. - **Storage qualification:** use this tick box to enable/disable storage qualification for the selected capture unit. 'Storage qualification' allows defining logic conditions on the capture unit inputs in order to filter the incoming data for capture. - **Number of pipes:** defines the number of pipelining stages placed in each of the data groups paths before the capture unit's multiplexer. Leaving this parameter to 'Disabled' automatically adds such pipes if necessary. Use this parameter to add more pipelining stages. - **Enable timestamping:** enables capture timestamping for the selected capture unit *(check release notes for function availability)* - **Monitor lost triggers:** allows monitoring triggers lost during interrupted captures. *(check release notes for function availability).*

Data		**Sets up the Capture Unit FIFO and summarizes the capture unit's size** - **Fifo depth:** specifies the capture unit's FIFO depth. Selectable values: 1024 to 8192. - **Number of data groups (informative only)** shows the number of data groups defined for this capture unit. - **Number of data probes (informative only)** shows the width of the capture unit in bits. This value is the width of the largest data group connected to the capture unit. It also defines the FIFO width. - **Extended sampling frequency /** - **Link bandwidth optimization** Combining these parameters allows reaching higher sampling clock frequencies and optionally adding the link barrel shifter unit. The right combination of parameters is best chosen from the **'Capture unit status' controls** (below).
Sampling clock		**Defines the reference clock from the target FPGA design used to sample data with the selected capture unit.** - If data from more than one clock domain must be sampled, a separate capture unit for each clock domain should be defined. - To select this clock from the design loaded in Vivado, link dashboard to Vivado first. Then click on "…" Refer to **'How to select capture unit clocks and data groups signals'** below. - The parameter 'clock frequency' must be specified – please check this value from the target design.
Capture unit status		**Diplays additional status and optional configurations about the selected capture unit.**

Data Group parameters in 'Netlist flow'



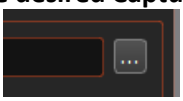
Each signal connected to the selected data group can be defined as 'Data only' or 'Data and Trigger' with 2 tick boxes. When defined as 'trigger', this signal can be used to define trigger condition during analysis. Unselecting the 'trigger' option for a signal helps reduce the logic resources required for implementing Exostiv IP .

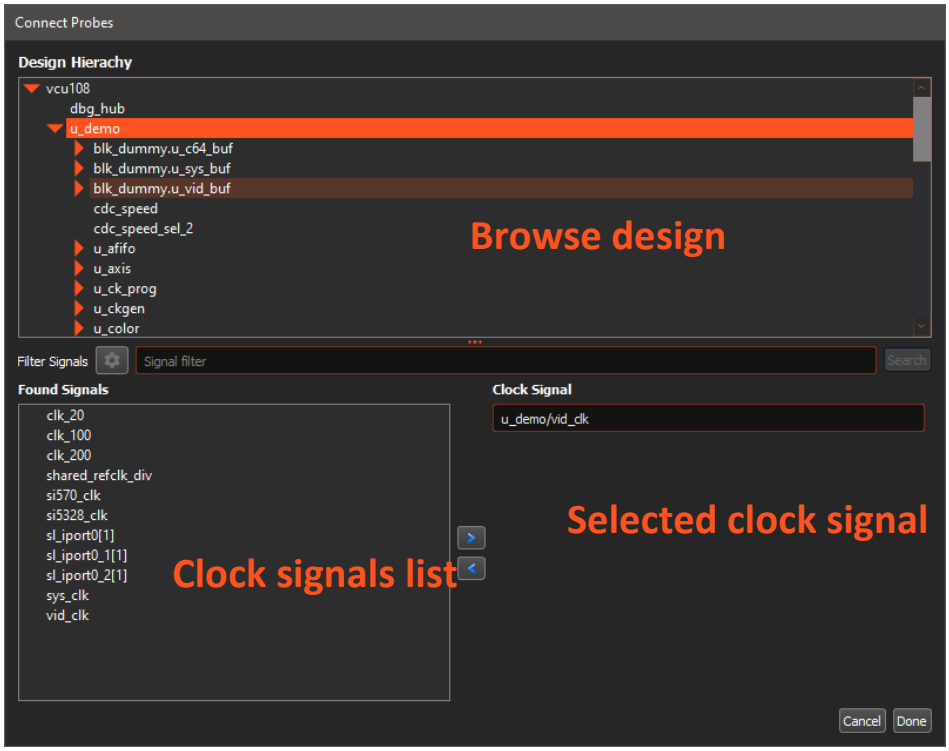
How to select capture unit clocks and data groups signals?

Defining the Capture Unit's sampling clock and selecting the signals (nodes) connected to a data group requires browsing the target FPGA design. To do this, Exostiv Core Inserter establishes a link with the FPGA vendor tool (Vivado for AMD FPGAs) and sends queries to it.

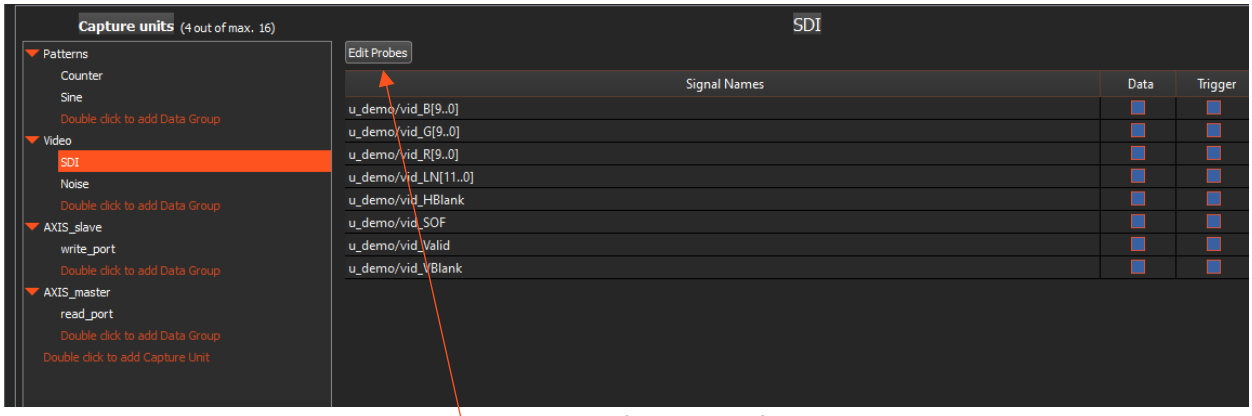
Please refer to 'Core Inserter – Linking to FPGA vendor tool' to know how to establish this link.

To select the Capture Unit's sampling clock:

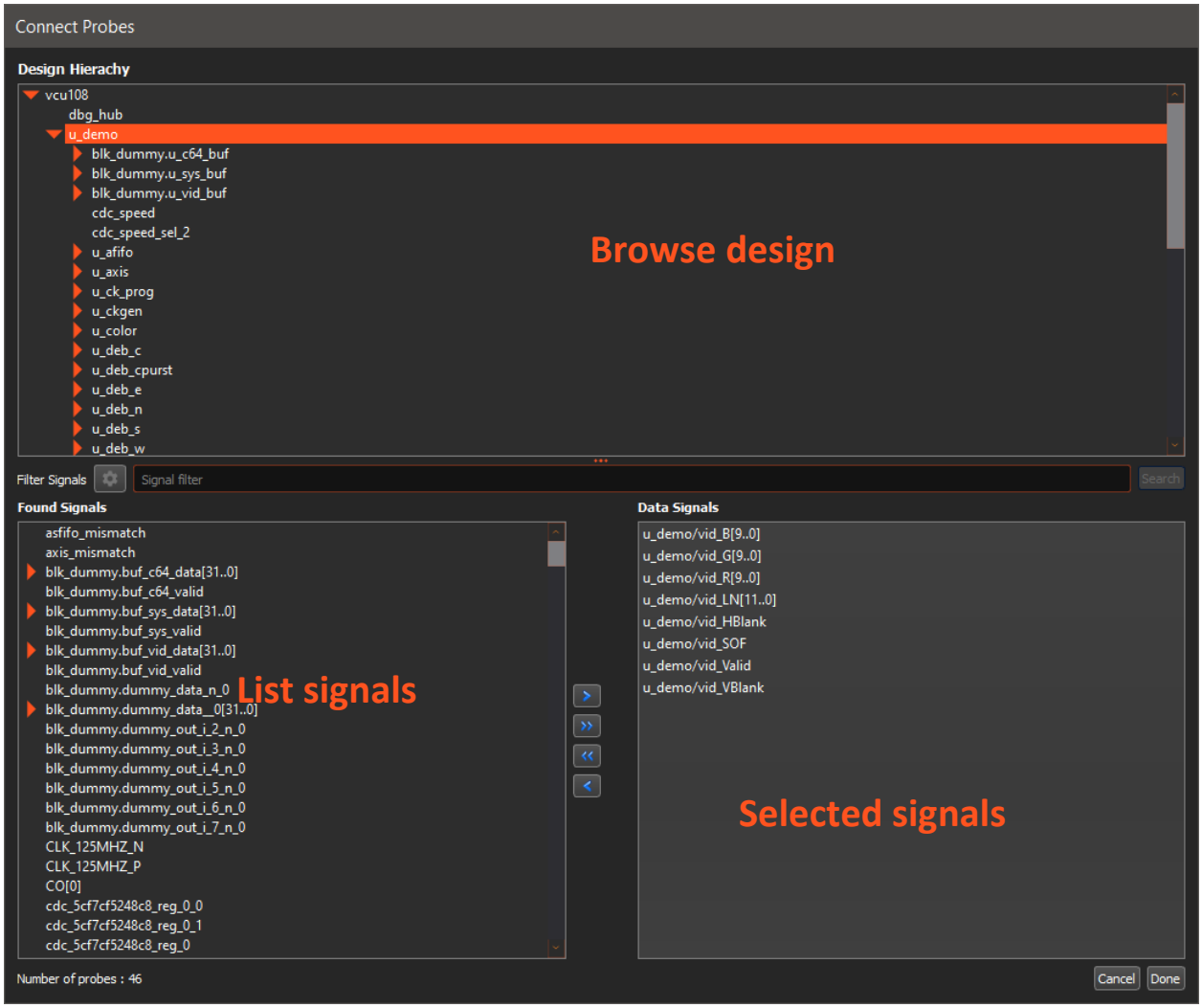
1. **Select the desired Capture Unit from the left column**.
2. **Click on**  **in the 'Sampling Clock' controls group**. It opens the 'Connect Probes' window, from which you can browse the target FPGA design and select the desired clock signal. Click on 'Done'.



To select a Data Group's signals:

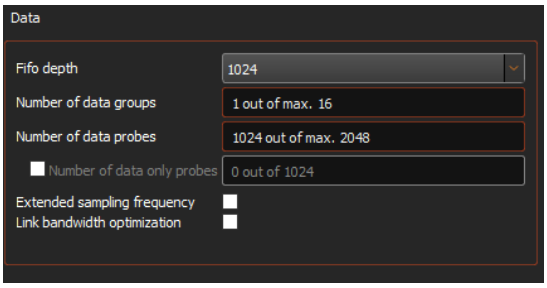


1. **Select the desired Data Group from the left hand column**
2. **Click on 'Edit Probes'**
3. In the 'Connect Probes' window, the design loaded in Vivado can be browsed and its internal nodes / signals are listed. Use the window controls to select the signals you wish to insert in the selected Data Group from the selected Capture Unit. Click on 'Done' once you're finished.



Note: *The sampling clock and the signals connected to a capture unit should be chosen as part of the same clock domain. If a clock is selected and the chosen signals of a data group are not in the corresponding clock domain, it will result a longer implementation of the Exostiv IP in the target design – and likely – a timing error. Exostiv Core Inserter future releases will include an automatic clock selection from data group signals to facilitate the definition and setup of data groups.*

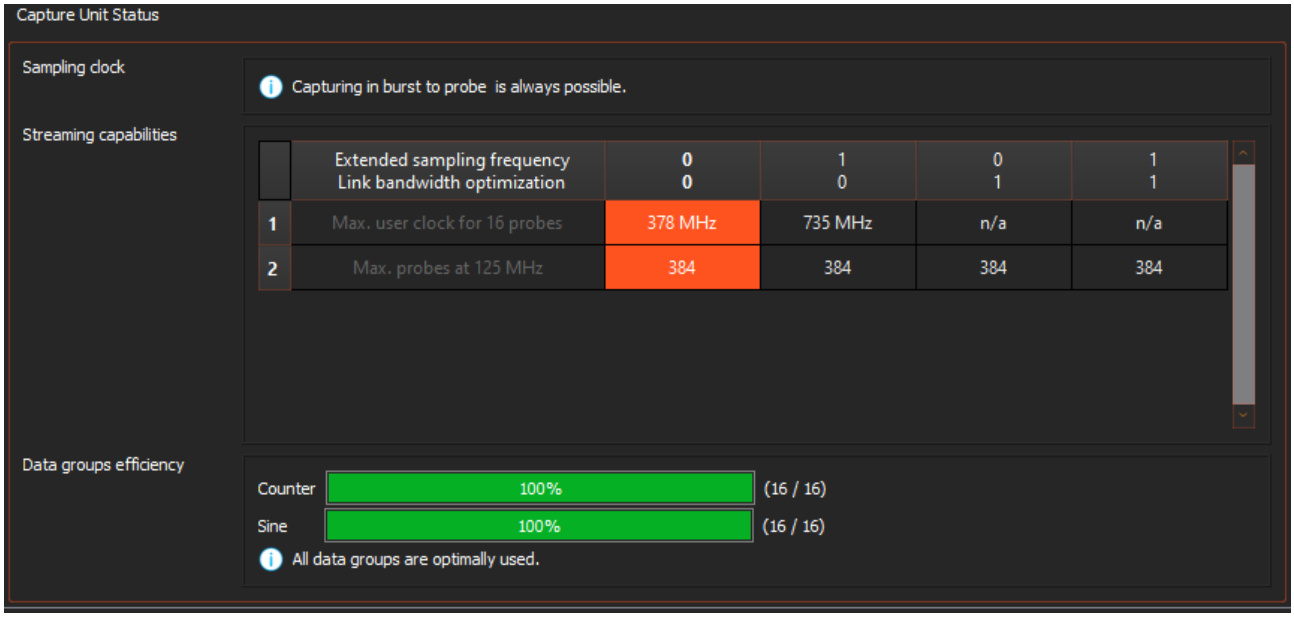
Differences in controls for Capture Unit parameters in 'RTL flow'

Control group		Description
Data		As opposed to the netlist mode of insertion, the RTL flow does not require to select the nodes from the target design. Consequently, the user must specify the size of the IP input ports – the 'number of data probes field is editable. It is possible to specify a number of these ports as 'data only' (unused for trigger).

Data Group parameters in 'RTL flow'

There are no additional parameters for the data groups. The target design nodes are selected by instantiating the IP in the RTL code. Selecting a data group displays the following message, reminding of the specified total width (number of nodes that can be connected) of the data group.

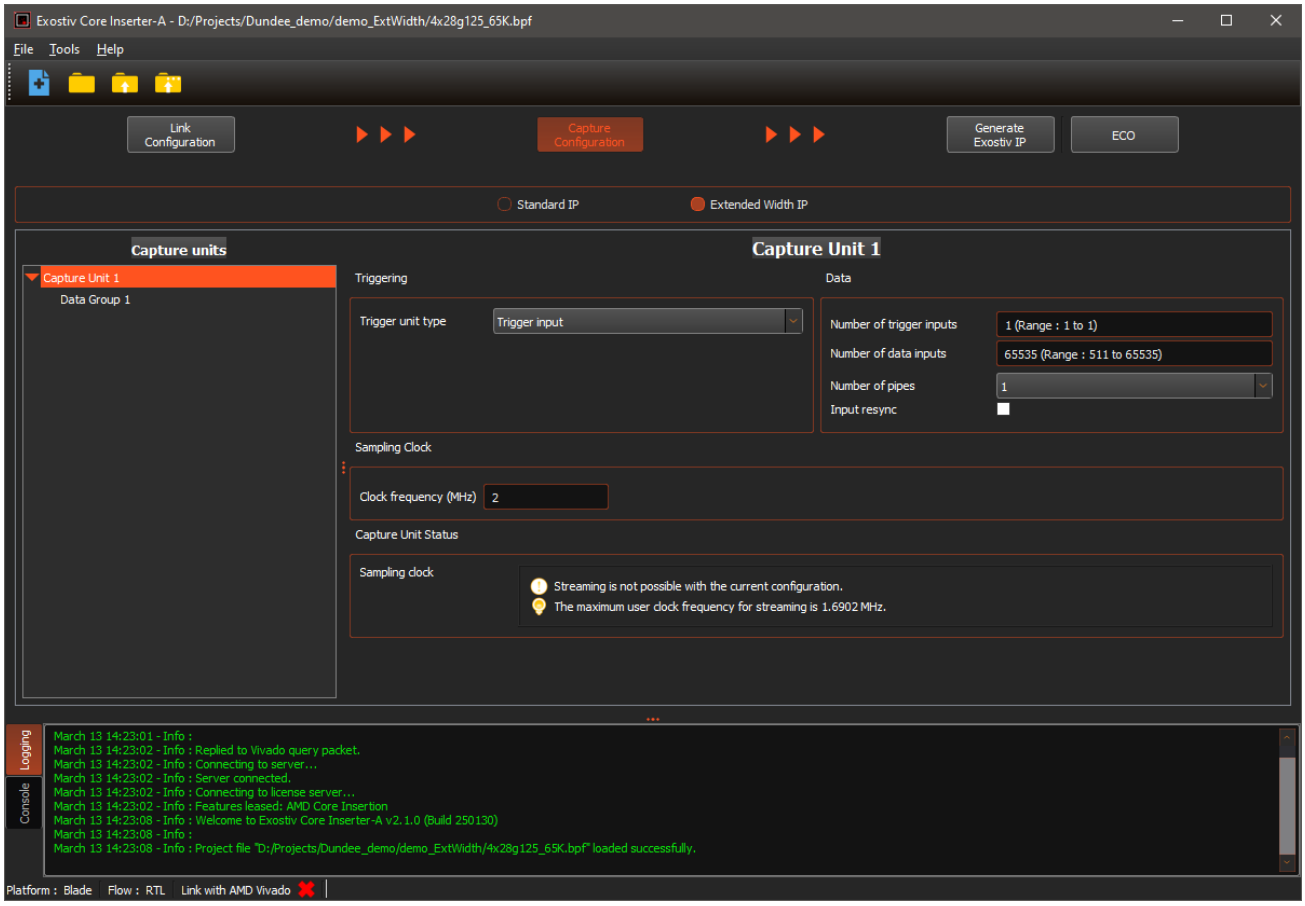
Capture Unit status

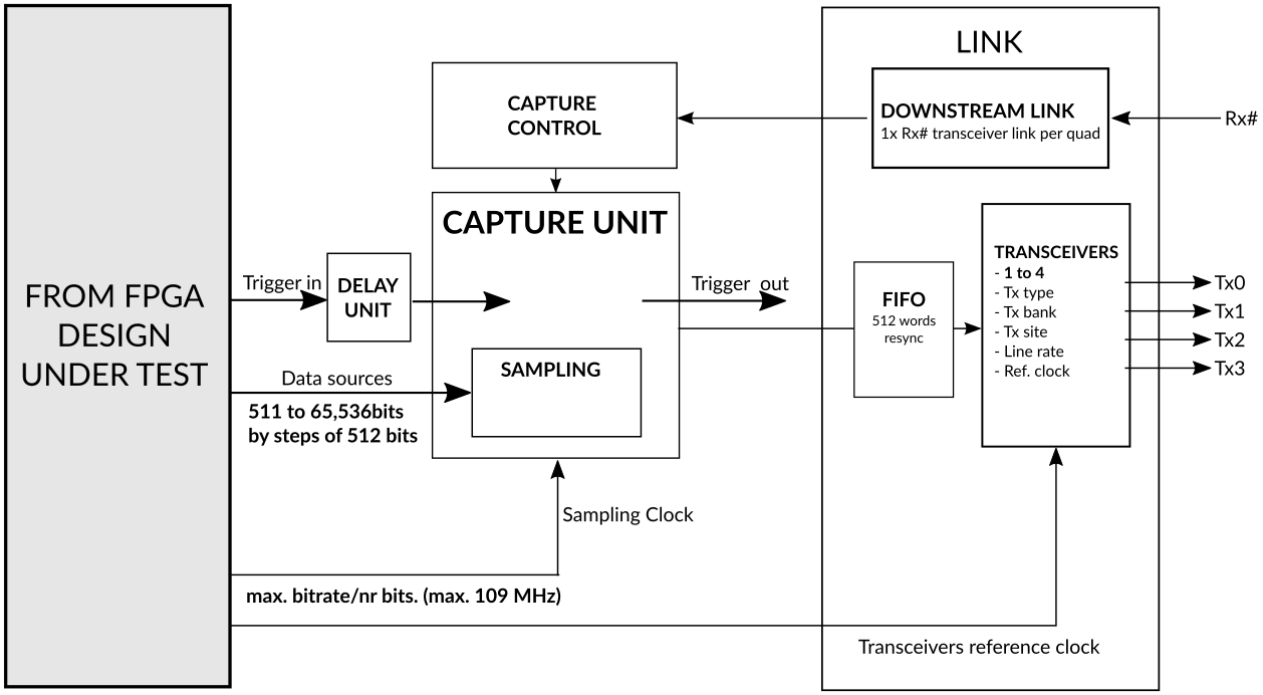


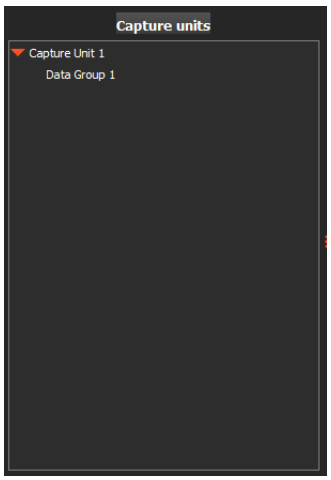
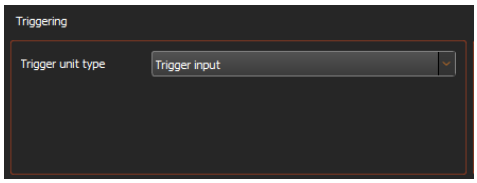
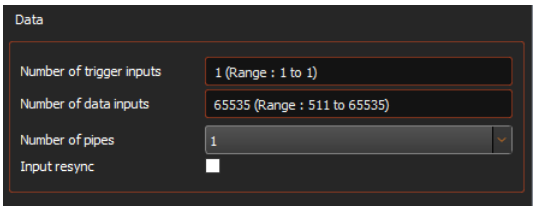
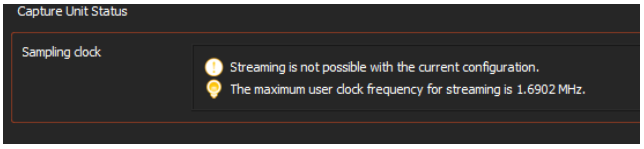
This section provides additional options and information about the streaming capabilities of the capture unit. For details, please refer to the following article: https://www.exostivlabs.com/knowledgebase/how-to-understand-the-capture-unit-status-section-in-the-core-inserter/

Extended width IP Capture Configuration: adding & configuring Capture Units

The ‘Extended width IP’ is a lightweight IP that can connect to a larger number of nodes than the ‘Standard IP’.
It is selected with the top radio button available from Exostiv Core Inserter version 2.1.0 **build 250130.**

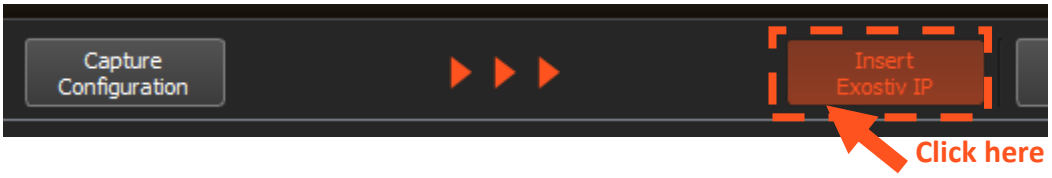




Control group		Description
Capture Units		The extended width IP features one single capture unit with one single data group.
Triggering		The trigger source for the extended width IP is a single signal, the trigger is sensitive to the rising edge. Currently, only this option ('Trigger input') is available.
Data		**The number of trigger inputs** is 1. **The number of data inputs** ranges from 511 to 65535. In RTL mode of insertion, this field is editable to specify the desired number of inputs. In Netlist mode of insertion, this number represents the number of data nodes connected to the data group (selection from the netlist in Vivado, in the same fashion as for the Standard IP above). **Number of pipes** are optional additional pipeline stage to be inserted at the inputs of the IP instance for the data. **Input resync:** when selected, this adds a single resynchronization stage at the inputs of the IP.
Sampling clock		Specifies the sampling clock frequency in MHz.
Capture unit status		This section provides additional information about the single capture unit of the IP, namely its streaming capability (based on the transceivers available bandwidth, number of nodes and the sampling frequency).

Core Inserter – Insert Exostiv IP (Netlist flow)

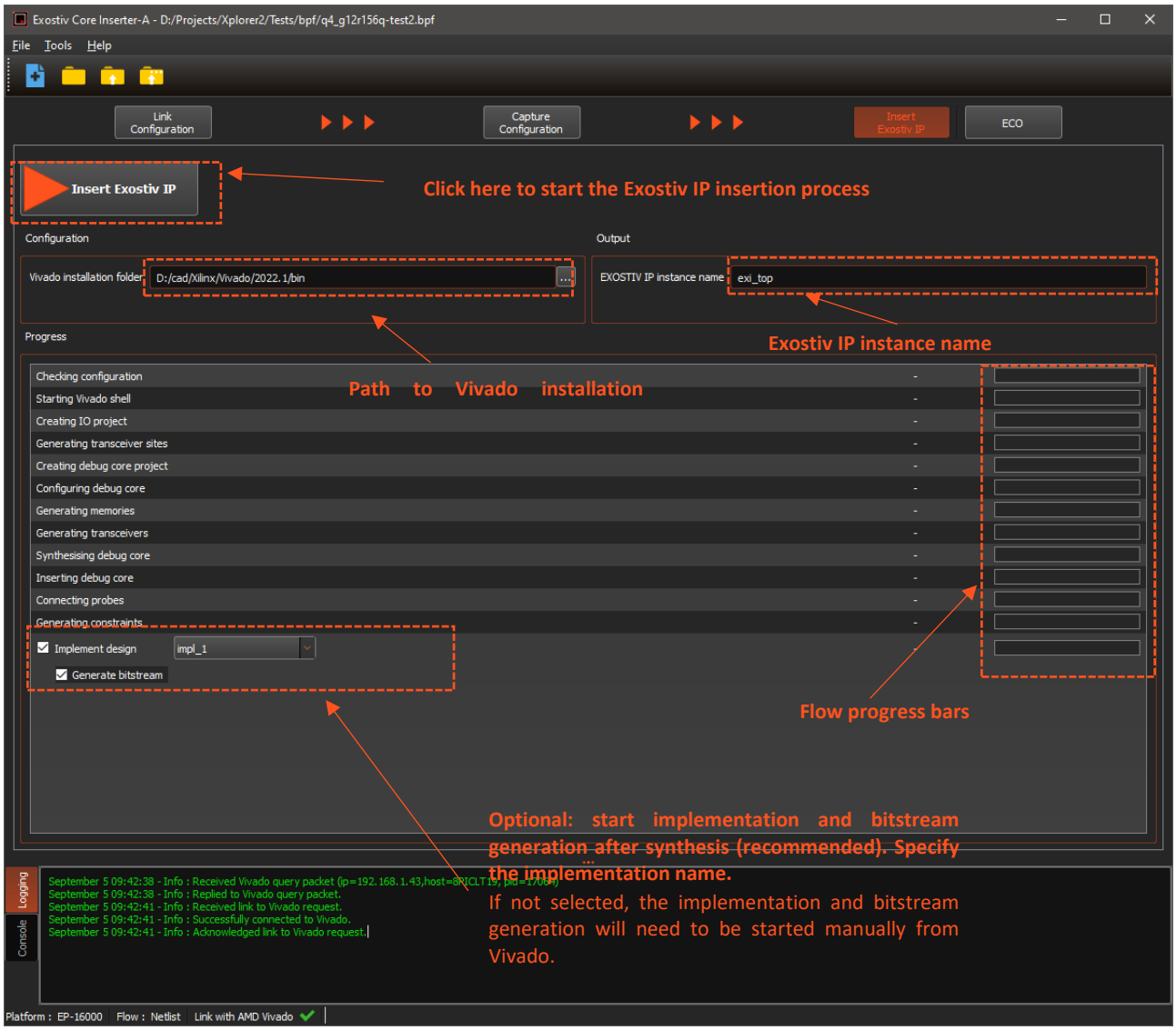
Insert Exostiv IP : locate and access



Insert Exostiv IP : overview

The 'Run Insertion window' is used to start the insertion of Exostiv IP in the target FPGA design.
For that purpose, there must be a link with the FPGA vendor tool (Vivado). To link Exostiv Core Inserter to Vivado, please refer Core Inserter – Linking to FPGA vendor tool.

The 'Run Insertion window' controls are summarized below:

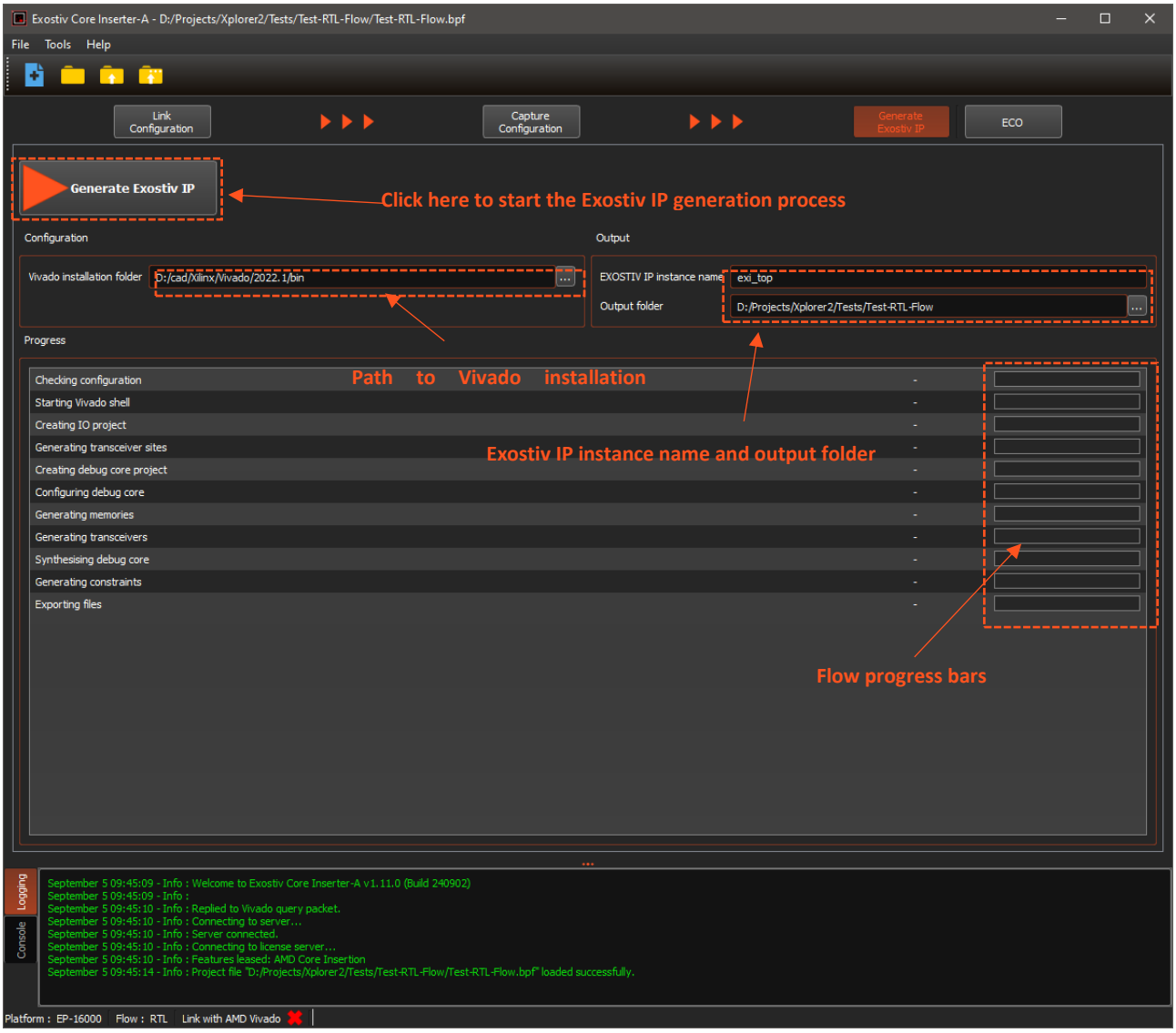


Schematically, when hitting the 'Insert Exostiv IP ', Exostiv Core Inserter will:

1. Synthesize the Exostiv IP
2. Insert the Exostiv IP into the target FPGA design loaded in Vivado
3. Run the chosen implementation (if selected)
4. Generate the bitstream (if selected).

Core Inserter – Generate Exostiv IP (RTL flow)

The 'Generate Exostiv IP ' is available for RTL flow projects only. The 'Generate Exostiv IP ' window and controls are accessible by clicking on the corresponding button on the flow bar.



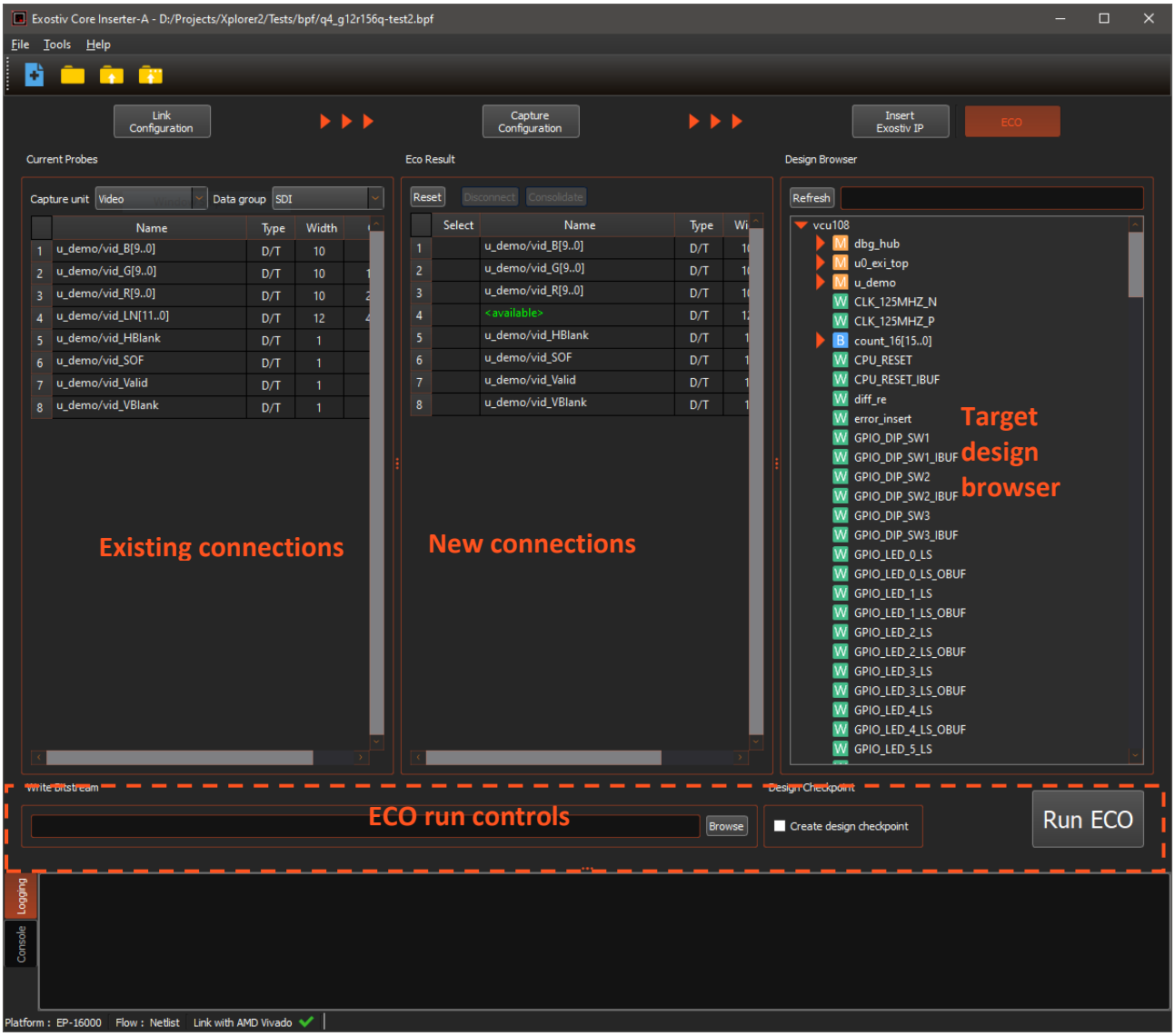
Core Inserter – ECO mode

The ‘ECO mode’ or ‘ECO flow’ refers to ‘Engineering Change Order’.
It allows modifying the connections of an Exostiv IP already inserted into the design after place and route, for fast turn around time.

This functionality does not modify the structure and the previously generated features of the existing Exostiv IP. It allows re-routing the connections of a specific IP to the design under test. It is typically used to avoid having to go through a new IP generation / insertion if a new set of nodes from the design under test need to be captured and observed.

To use the ECO mode:

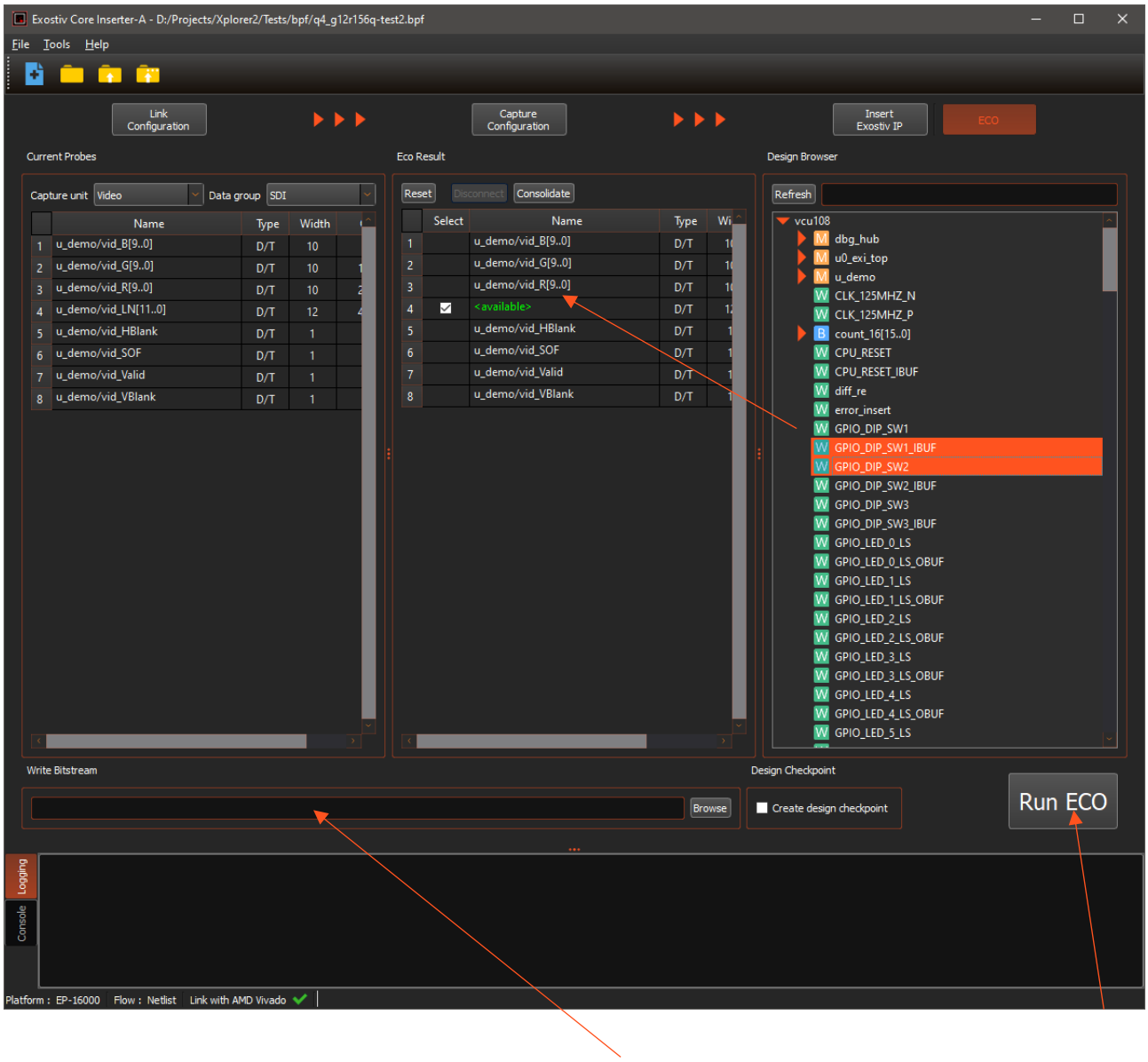
1) Load post place-and-route design checkpoint into Vivado (or ‘open implemented design’).
2) Load the corresponding .bpf project file in Exostiv Core Inserter
3) Link Vivado to Exostiv Core Inserter with the link button.



To use the ECO flow, please follow these steps:

1) From ‘New connections’ section, select one or multiple existing connections and click on ‘**disconnect’** to make them available for a new connection. The **Reset** button reinitializes the connections to the existing ones (‘Existign connections’). The **‘Consolidate’** button allows merging multiple disconnected connections to a single one.

2) Select one available connection from the **'New connections' (tick box) and click and drag the desired new connection signals from the target design browser.**



3) Once all new connections are defined, specify an output bitsteam file in the '**Write bitstream' prompt** and click on '**Run ECO'.**

The tool will call Vivado to modify the existing connections with the design under test and generate a new bitstream.

Copyright

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

Disclaimer

THIS DOCUMENT IS PROVIDED "AS IS". Exostiv LABS PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Exostiv LABS makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

The contents of this document are subject to change without notice. This document may contain information on an Exostiv Labs product under development by Exostiv Labs. Exostiv Labs reserves the right to change or discontinue work on any product without notice.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL Exostiv LABS BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF Exostiv LABS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Exostiv Labs products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Exostiv Labs products in such critical applications.

https://www.exostivlabs.com