

WWW.ESISTREAM.COM

ESIstream

The Efficient Serial Interface

ESIstream INTEL A10 V2-1

Version 2.1

ESIstream is an Open-source high efficiency serial interface protocol based on 14b/16b encoding.

Its main benefits are low overhead and ease of hardware implementation.

Reference documents

ESIstream Protocol specification V2.0: www.ESIstream.com

EV12AQ600 datasheet: [website link](#), [download link](#)

Document aim and comment

This document aims at explaining the architecture, the design, the environment of simulation and the environment of compilation of ESIstream transmitter (TX) and receiver (RX) IP package.

To make the adoption of ESIstream easier and to speed up the integration into customers' products, the ESIstream IP package include all the source code for both ends of the link transmitter (Tx) and receiver (Rx). A fully implementable project supporting INTEL FPGA Arria 10 (10AX115N4F40I3SG). Furthermore, the ESIstream IP package include detailed test-bench simulation for both the Tx and Rx.

This package allows testing of the communication with a lane rate up to 12.8Gbps.

Although the current IP requires a Intel FPGA Arria 10, the modular architecture allows an easy migration of the ESIstream IP to a different Intel target replacing the FPGA specific IPs.

Following the user guide section of this document, user will be able to generate two implementable projects. One using a KAYA loopback test board and the other one using a FMC EV12AQ600 test board, both project are based on a REFLEX CES Attila evaluation kit.

A test-bench simulation is also available with the KAYA loopback test board project using Vivado simulator.

For technical support on ESIstream, contact: GRE-HOTLINE-BDC@Teledyne.com

Packages supported by this document

ESISTREAM_INTEL_A10_V2-1

Introduction to ESIstream

ESIstream protocol initiated by Teledyne-e2v is born from a severe need of the following combination:

- Reduced data overhead on serial links, as low as possible.
- Increased rate of useful data when linking ADCs operating at GSPS speeds with FPGAs on a serial interface.
- Simplified hardware implementation, simple enough to be built on RF SiGe technologies.

ESIstream provides an efficient High-Speed serial 14b/16b interface. It is open-source and supports in particular serial communication between FPGAs and High-Speed data converters.

An ESIstream system is made up of a transmitter and a receiver.

- A transmitter can be an ADC or an FPGA or an ASIC
- A receiver can be a DAC or an FPGA or an ASIC
- A number of lanes ($L \geq 1$) to transmit serial data
- A synchronization signal (sync) to initialize the communication. Clock embedded in each lane data stream (need to be recovered by RX).

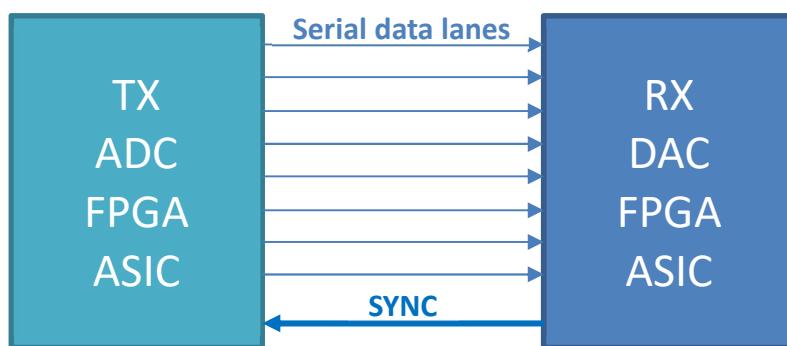


Figure 1: Basic ESIstream system

The main keys benefits are:

- FLEXIBILITY: Open-source and license free
- EFFICIENCY: 87.5%, with 14 bit of useful data and 2 bit overhead (clock bit and disparity bit).
- SIMPLICITY: Minimal hardware implementation.

The main specifications of ESIstream are:

- Deterministic latency
- Multi-device synchronization
- Demonstrated lane rate up to 12.8Gbps, depending on device abilities
- Multi-lanes synchronization
- Guaranteed DC balance transmission, ±16 bit running disparity
- Synchronization monitoring, using the clock bit (overhead bit)
- Sufficient number of transitions for CDR, max run length of 32.

Disclaimer

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled bitstream, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

THIS DISCLAIMER MUST BE RETAINED AS PART OF THIS FILE AT ALL TIMES.

TABLE OF CONTENTS

REFERENCE DOCUMENTS.....	1
DOCUMENT AIM AND COMMENT.....	1
PACKAGES SUPPORTED BY THIS DOCUMENT	1
INTRODUCTION TO ESISTREAM	2
DISCLAIMER.....	3
1. DESIGN GUIDE.....	6
1.1 TRANSMITTER IP (TX).....	6
1.1.1 <i>Description</i>	6
1.1.2 <i>Transmitter entity</i>	9
1.1.3 <i>Port description</i>	9
1.2 RECEIVER IP (RX).....	10
1.2.1 <i>Description</i>	10
1.2.2 <i>Receiver entity</i>	12
2. USER GUIDE	14
2.1 SOFTWARE VERSIONS AND DOWNLOAD LINK	14
2.2 KAYA FMC LOOPACK TEST BOARD PROJECT SETUP	15
2.2.1 <i>Hardware</i>	15
2.2.2 <i>Software</i>	16
2.3 EV12AQ600 FMC TEST BOARD PROJECT SETUP.....	26
2.3.1 <i>Hardware</i>	26
2.3.2 <i>Software</i>	28
2.3.3 <i>RAMP mode test</i>	30
3. ANNEX A: SWITCH PROJECT TO A 32-BIT SERDES	32
4. ANNEX B: ATTILA BOARD INFORMATION.....	36

ESIstream

The Efficient Serial Interface

FIGURES

Figure 1: Basic ESIstream system.....	2
Figure 2: Block diagram of ESIstream TX IP with 2 lanes	6
Figure 3: Synchronization sequence	7
Figure 4: ESIstream encoded data.....	8
Figure 5: Block diagram of ESIstream RX IP with 2 lanes.....	10
Figure 6: Frame alignment and output buffers alignment	11
Figure 7: ESIstream received data	12
Figure 8: Attila Arria 10 GX Board with Kaya loopback test board	15
Figure 9: Quartus II, Generate bitstream and program the FPGA	18
Figure 10: 64-bit SERDES testbench Block diagram.....	25
Figure 11: Attila Arria 10 GX Board with EV12AQ600 FMC test board	26
Figure 12: Chronogram of the Ramp test mode	30
Figure 13: Attila Arria 10 GX Board	36

Issue	Date	Comments
1.0	June 2019	Creation
2.1	October 2019	Publication

ESIstream

The Efficient Serial Interface

1. DESIGN GUIDE

1.1 Transmitter IP (TX)

1.1.1 Description

1.1.1.1 Architecture

The transmitter IP allows generation of the synchronization sequence on a sync pulse event (alignment and prbs) and encoding of the input data according to the ESIstream protocol specification (scrambling processing, clock bit concatenation, disparity bit processing and concatenation).

The encoding block encapsulates the encoding ESIstream protocol layer and makes the interface between ESIstream TX IP incoming data to encode and the encoded data to transmit through the transceiver IP.

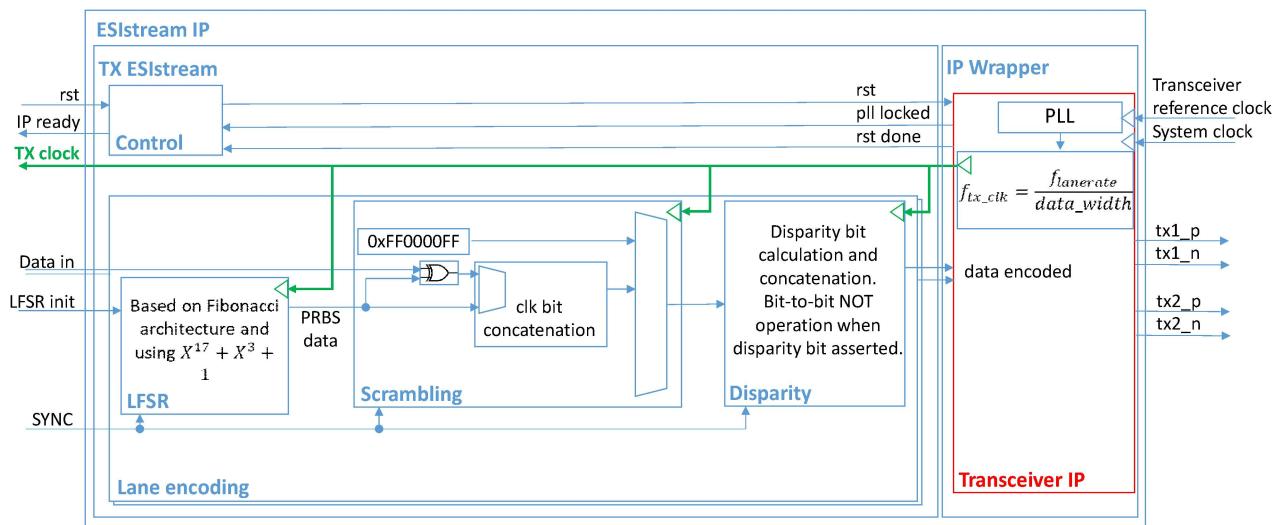


Figure 2: Block diagram of ESIstream TX IP with 2 lanes

The FPGA manufacturer provides the transceiver IP (in red in the above block diagram), and the transceiver IP can change depending on the FPGA target device manufacturer and reference.

Module	.vhd file name	Description
ESIstream IP	tx_esistream_with_xcvr	Serializes and transmits data, using an Intel FPGA transceiver IP, on the differential serial lane outputs (tx_n / tx_p).
TX ESIstream	tx_esistream	For each lane, encodes useful data (data_in signal) into an ESIstream frame vector (data_encoded signal).
Control	tx_control	Manages and monitors transceiver PLL(s) lock, reset, reset done, user ready and ip ready signals.
Encoding	tx_encoding	When SER_WIDTH = 32 : Encodes useful data 2x14-bits (data_in signal) into a 2x16-bits ESIstream frame vector (data_encoded signal) according to the ESIstream protocol specification. When SER_WIDTH = 64 : Encodes useful data 4x14-bits (data_in signal) into a 4x16-bits ESIstream frame vector (data_encoded signal) according to the ESIstream protocol specification.
LFSR	lfsr	Generates pseudo-random binary sequence (PRBS) to scramble data using a linear feedback shift register (LFSR) based on a

ESIstream

The Efficient Serial Interface

		Fibonacci architecture and using the polynomial $X^{17} + X^3 + 1$. Applying scrambling ensures a statistical DC balanced transmission. In case of multi-lane interfaces in order to reduce correlation between lanes, each lane may have different initial values for scrambling units (LFSR init input signal)
Scrambling	tx_scrambling	When a sync event occurs, generates the synchronization sequence for receiver frame alignment and PRBS initialization. When SER_WIDTH = 32 : it scrambles 2x14-bit ESIstream data with a pseudo-random binary sequence generated by the LFSR module and concatenates the overhead clock bit to each data stream. When SER_WIDTH = 64 : it scrambles 4x14-bit ESIstream data with a pseudo-random binary sequence generated by the LFSR module and concatenates the overhead clock bit to each data stream.
Disparity	tx_disparity	When SER_WIDTH = 32: Calculates and processes disparity (bit-to-bit NOT) for 2x16-bit ESIstream frames concatenated with the overhead disparity bit. When SER_WIDTH = 64: Calculates and processes disparity (bit-to-bit NOT) for 4x16-bit ESIstream frames concatenated with the overhead disparity bit. (ESIstream protocol specification - ANNEX C: Example of disparity bit implementation).
Disparity submodule	tx_disparity_word_16b	Calculates the current word disparity for a 16-bit ESIstream frame. (ESIstream protocol specification - ANNEX C: Example of disparity bit implementation).
IP Wrapper	tx_xcvr_wrapper	Contains transceiver IP and provides a generic entity interface with it.

1.1.1.2 Synchronization sequence

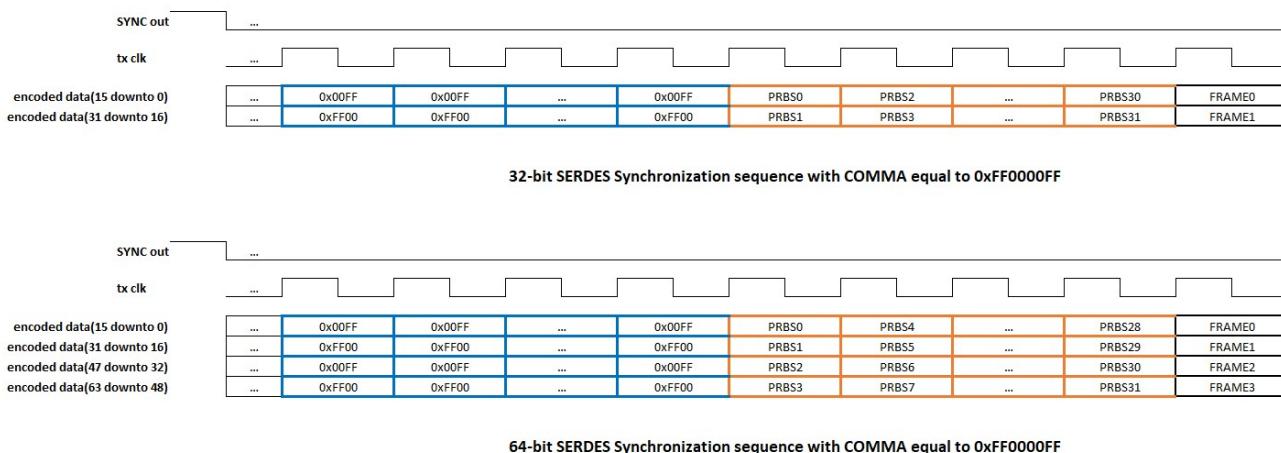


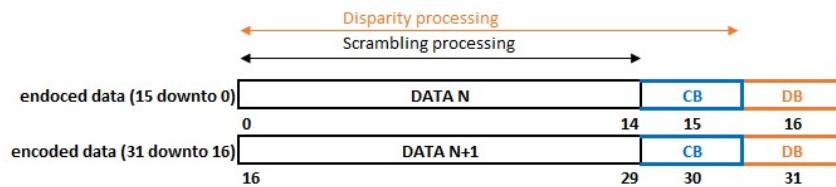
Figure 3: Synchronization sequence

When the transmitter receives a synchronization pulse, it will send a frame alignment sequence of 32x16-bit frames alternating between 0x00FF and 0xFF00 if COMMA equal to 0xFF0000FF with no scrambling and no disparity processing applied. The receiver can use the alignment pattern 0xFF0000FF or 0x00FFFF00 (COMMA) to align the received data and then to rebuild the frames sent by the transmitter.

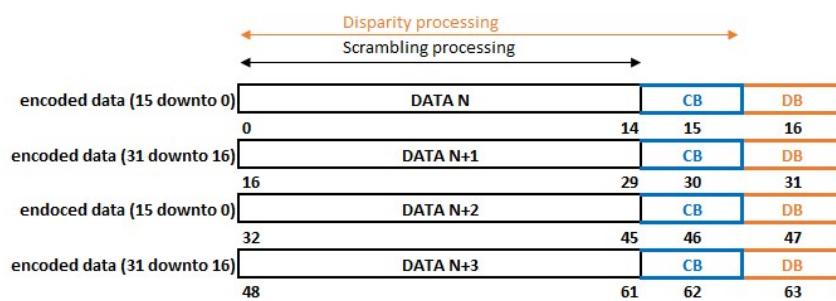
During the second part of the sequence it will send 32x16-bit additional frames containing PRBS values in the 14-bit data field plus two undefined overhead bits. This sequence allows initializing the receiver LFSR to generate the PRBS used by the descrambling process.

1.1.1.3 Encoded data

After the synchronization sequence, transmitter sends scrambled data plus overhead bits (clock bit & disparity bit).



32-bit SERDES encoded data



64-bit SERDES encoded data

Figure 4: ESIstream encoded data

ESIstream

The Efficient Serial Interface

1.1.2 Transmitter entity

1.1.2.1 Package

The ESIstream IP (tx_esistream_with_xcvr) and sub-modules use a common package (*esistream_pkg.vhd*) to share signal types, LFSR functions, and DESER_WIDTH / SER_WIDTH constants.

1.1.2.2 Generic description

Generic	Type	Values	Description
NB_LANES	natural	1 to (FPGA maximum number of transceivers)	Number of lanes
COMMA	std_logic_vector	x"00FFFF00" or x"FF0000FF"	Frame alignment synchronization pattern.

1.1.3 Port description

Port	IN / OUT	width	Description
rst	input	1-bit	Active high transceiver PLL asynchronous reset
refclk_n	input	1-bit	Transceiver reference clock n input.
refclk_p	input	1-bit	Transceiver reference clock p input.
sysclk	input	1-bit	Transceiver system clock [100MHz typ].
sync_in	input	1-bit	Active high pulse. Generates the synchronization sequence for the receiver, frame alignment and PRBS initialization.
prbs_en	input	1-bit	Active high, enables scrambling processing.
disp_en	input	1-bit	Active high, enables disparity processing.
lfsr_init	input	NB_LANES array of 17-bit	Selects LFSR initialization value for each lane.
data_in	input	NB_LANES array of 14-bit x SER_WIDTH/16	Useful data to encode. <ul style="list-style-type: none">- When SER_WIDTH = 32: 2x14-bit at each tx_clk cycle.<ul style="list-style-type: none">o data_in(NB_LANES_index)(0) : Data No data_in(NB_LANES_index)(1) : Data N+1- When SER_WIDTH = 64: 4x14-bit at each tx_clk cycle.<ul style="list-style-type: none">o data_in(NB_LANES_index)(0) : Data No data_in(NB_LANES_index)(1) : Data N+1o data_in(NB_LANES_index)(2) : Data N+2o data_in(NB_LANES_index)(3) : Data N+3
txn	output	NB_LANES array of 1-bit	Not used by Intel FPGAs which manage the differential pair in the constraint file with pin assignment.
txp	output	NB_LANES array of 1-bit	Transceiver serial p output for each lane. Encoded data sent LSB first.
tx_clk	output	1-bit	Transceiver SER_WIDTH-bit frame clock. This clock is not available during transceiver reset. $f_{tx_clk} = \frac{f_{lane_rate}}{SER_WIDTH}$
ip_ready	output	1-bit	Active high, when transceiver PLL(s) locked and transceiver reset done. Indicates that IP is ready to receive a sync pulse.

ESIstream

The Efficient Serial Interface

1.2 ReceiveR IP (RX)

1.2.1 Description

1.2.1.1 Architecture

The receiver IP sends the sync event to the transmitter and waits for the synchronization sequence to align the lanes and to initialize the LFSR. Then the receiver decodes the deserialized transceiver raw data according to the ESIstream protocol specification (descrambling processing, disparity bit processing).

The lane decoding block encapsulates the decoding ESIstream protocol layer and makes the interface between the encoded data received through the transceiver IP and ESIstream RX IP decoded data at the buffer output.

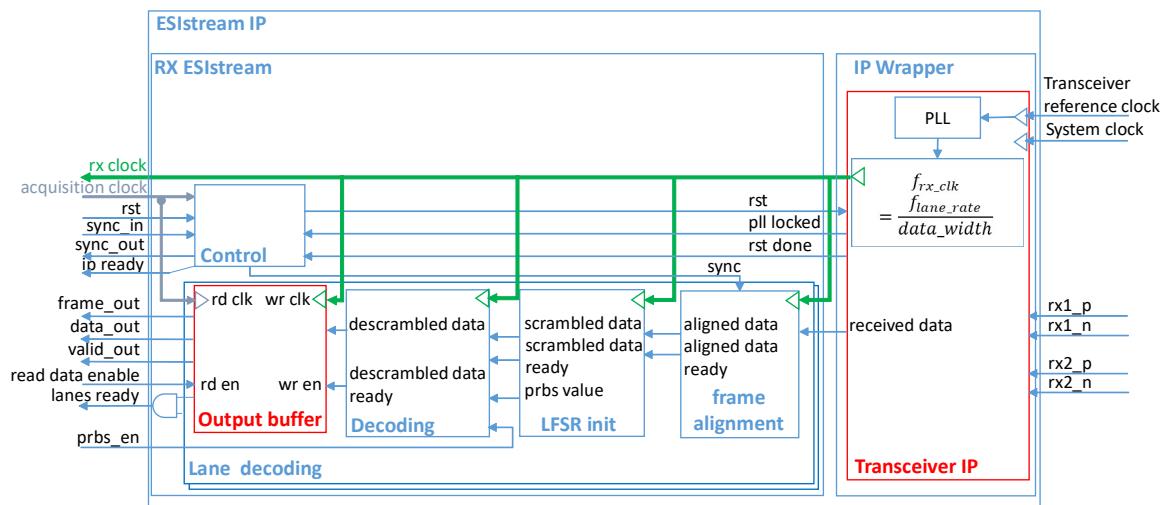


Figure 5: Block diagram of ESIstream RX IP with 2 lanes

The FPGA manufacturer provides the transceiver IP and the output buffer IP (in red in the above block diagram). User should replace these two IP blocks according to the targeted FPGA.

Module	.vhd file name	Description
ESIstream IP	rx_esistream_with_xcvr	Deserializes and receives data using an Intel FPGA transceiver IP from the differential serial lane inputs (rx_n / rx_p).
RX ESIstream	rx_esistream	For each lane: - When DESER_WIDTH=32, Decodes 32-bits raw data (2x16-bits ESIstream encoded frame vector) to provide decoded useful data 2x14-bits (data_out signal) and related overhead clock bit and disparity bit (through frame_out) when valid_out is high. - When DESER_WIDTH=64, Decodes 64-bits raw data (4x16-bits ESIstream encoded frame vector) to provide decoded useful data 4x14-bits (data_out signal) and related overhead clock bit and disparity bit (through frame_out) when valid_out is high.
Control	rx_control	Manages and monitors sync, transceiver PLL(s) lock, reset, reset done, user ready and ip ready signals.
Lane decoding	rx_lane_decoding	- When DESER_WIDTH = 32, Decodes useful data 2x14-bits, from a 2x16-bits ESIstream frame vector (32-bits) from transceiver output

ESIstream

The Efficient Serial Interface

		according to the ESIstream - When DESER_WIDTH = 64, Decodes useful data 4x14-bits, from a 4x16-bits ESIstream frame vector (64-bits) from transceiver output according to the ESIstream.
Frame alignment	rx_frame_alignment	After a sync event, waits and detects frame alignment synchronization sequence. Indicates to the LFSR init module to start PRBS initialization sequence when frames aligned.
LFSR init	rx_lfsr_init	After a sync event, waits for frames aligned event to initialize and synchronize the PRBS with scrambled data.
Decoding	rx_decoding	Applies descrambling and reverse disparity processing on each frame.
IP Wrapper	rx_xcvr_wrapper	Contains transceiver IP and provides a generic entity interface with it.
output buffer	rx_output_buffer_wrapper	Provides a generic interface for the output buffer. Embeds Intel FPGA IP (dual clock FIFO). Frames buffering until all lanes ready and read data enable signal asserted. Frames buffering always begins with the first frame of the PRBS initialization sequence. Independent clocks (rx_clk, wr_clk) built-in fifo, 16-bit width. rd_clk and wr_clk should be at the same synchronous frequency. Fifo not empty indicates synchronized data are available at buffer outputs. Acquisition clock (acq_clk) and rx_clk must be synchronous. Default rx_clk can be connected

1.2.1.2 Frame alignment

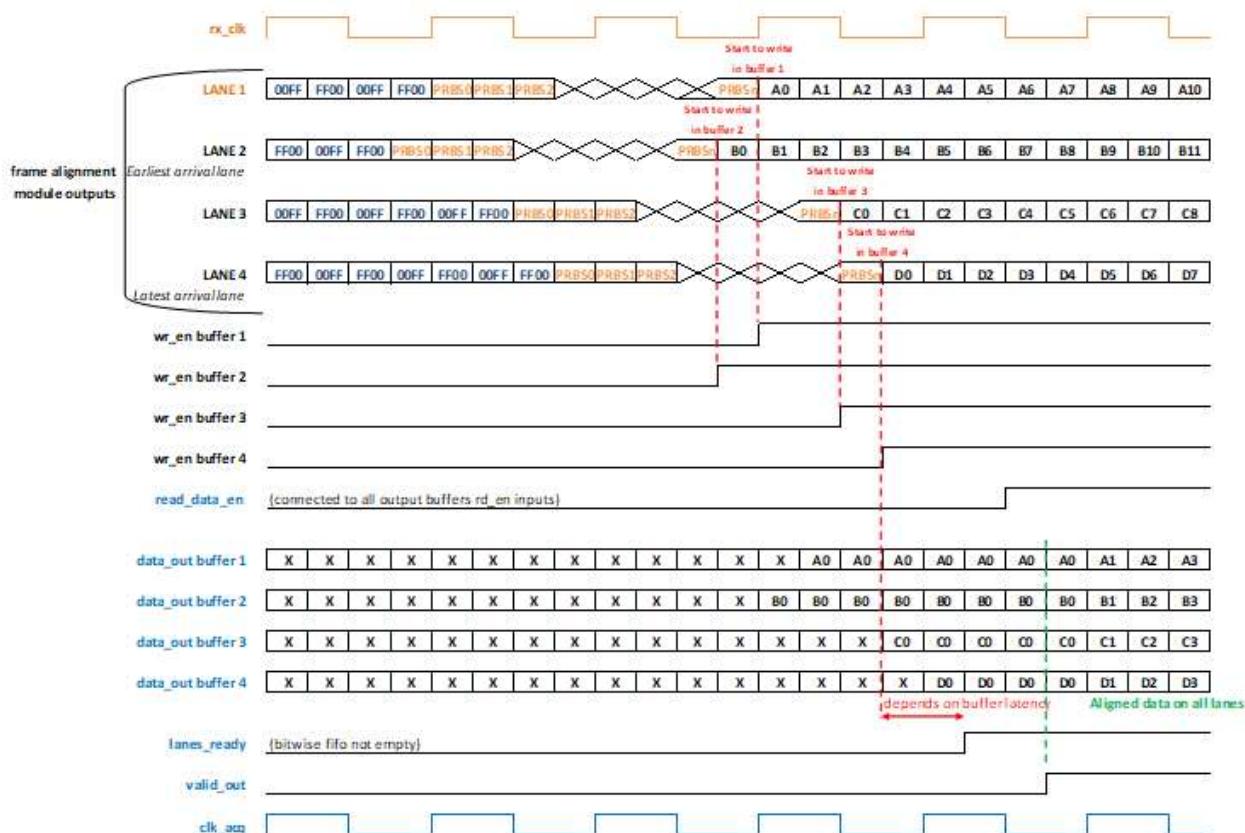


Figure 6: Frame alignment and output buffers alignment

ESIstream

The Efficient Serial Interface

1.2.1.3 Synchronization sequence

See [section 2.1.2](#).

1.2.1.4 Received data

After the synchronization sequence, transmitter send scrambled data plus overhead bits (clock bit & disparity bit).

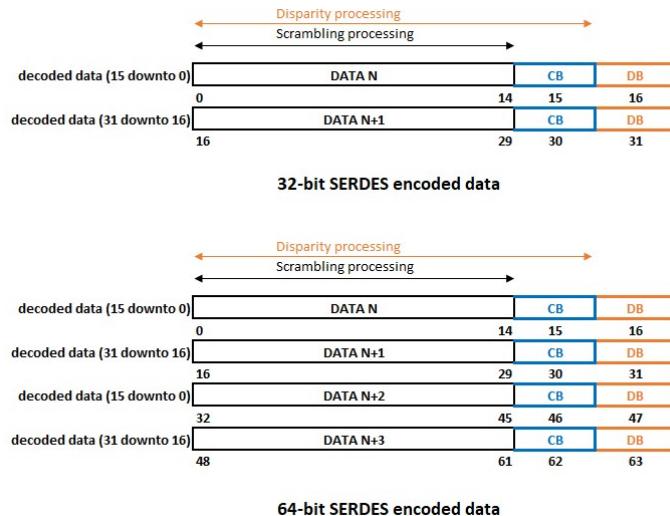


Figure 7: ESIstream received data

1.2.2 Receiver entity

1.2.2.1 Package

The ESIstream RX IP module (*rx_esistream_with_xcvr.vhd*) and sub-modules use a common package (*esistream_pkg.vhd*) to share signal types, LFSR functions, and DESER_WIDTH / SER_WIDTH constants.

1.2.2.2 Generic description

Generic	Type	Values	Description
NB_LANES	natural	1 to (FPGA maximum number of transceivers)	Number of lanes
SYNC_DELAY	natural	1 to n	Delay to add between sync output and sync process in lane decoding module.
COMMA	std_logic_vector	x"00FFFF00" or x"FF0000FF"	Frame alignment synchronization pattern.

1.2.2.3 Port description

Port	IN / OUT	width	Description
rst	input	1	Active high transceiver PLL asynchronous reset
sysclk	input	1	Transceiver system clock [100MHz].
refclk_n	input	1	Transceiver reference clock n input.
refclk_p	input	1	Transceiver reference clock p input.
rxn	input	NB_LANES array	Transceiver serial n input for each lane.

ESIstream

The Efficient Serial Interface

		of 1	
rxp	input	NB_LANES array of 1	Transceiver serial p input for each lane.
sync_in	input	1	Active high pulse. Generates the synchronization sequence for the receiver, frame alignment and PRBS initialization.
prbs_en	input	1	Active high, enables scrambling processing.
lanes_on	Input	NB_LANES array of 1	Active high, enable lane. If lane disabled, then no data is available in corresponding lane output buffer.
read_data_en	input	1	Active high, enables read of data at buffers outputs. When enabled and when valid_out output is high, received data is streamed on frame_out and data_out.
clk_acq	input	1	Acquisition clock, output buffer read port clock, should be the same frequency with no phase drift with respect to receive clock (default: clk_acq should take rx_clk).
rx_clk	output	1	Transceiver DESER_WIDTH-bit frame clock. This clock is not available during transceiver reset. $f_{rx_clk} = \frac{f_{lane\ rate}}{DESER\ WIDTH}$
sync_out	output	1	Active high pulse. Connect to transmitter sync_in to generate the synchronization sequence for the receiver, frame alignment and PRBS initialization
frame_out	output	NB_LANES array of 16 x DESER_WIDTH/16	Received and decoded ESIstream output frame: disparity bit (15) + clk bit (14) + data (13 down to 0).
data_out	output	NB_LANES array of 14 x SER_WIDTH/16	Received and decoded useful data. Bits (13 down to 0) of ESIstream frame.
valid_out	output	NB_LANES array of 1	Active high one rx_clk period later than read_data_en, frame_out and data_out stream valid output.
ip_ready	output	1	Active high, when transceiver PLL(s) locked and transceiver reset done. Indicates that IP is ready to receive a sync pulse.
lanes_ready	output	1	Active high, indicates lanes synchronization status. When high, all enabled lanes are synchronized and data are available at output_buffer(s) outputs.

ESIstream

The Efficient Serial Interface

2. USER GUIDE

2.1 Software versions and download link

- Quartus Prime 18.1 Standard Edition
- Modelsim Intel FPGA Edition 10.5b
- Arria 10 device support

To download this software, use the following link and select the files to download as show in the picture below:

<http://fpgasoftware.intel.com/?edition=standard>

Quartus Prime Standard Edition
Release date: September, 2018
Latest Release: v18.1

Intel® Quartus® Prime
Design Software

Select edition: Standard ▾
Select release: 18.1 ▾

Operating System: Windows Linux

Download Method: Akamai DLM3 Download Manager Direct Download

✓ The Quartus Prime Design Software Standard Edition v18.1 Update 1 has been updated with Curl 7.64.0, CygWin 3.0.3, expat 2.2.6, flexnet_publisher 11.16.2.1, glib 2.56.1, gstreamer 1.14.0, guava 27.0.1, jre (Oracle Java) 1.8.0u202, libicu 61.0, libpng 1.6.36, openssl 1.0.2r, perl 5.28.1, python 3.6.8, qt 5.11.3, ccl_sqlite3 3.26.0, sqlite3 (python) 3.26.0, sqlite3 (qt) 3.27.2, xerces-c++ 3.2.2, and zlib 1.2.11, which include functional and security updates. Users should update to the latest version of Quartus Prime design software as soon as possible.

✓ The Quartus Prime Standard software version 18.1 supports the following device families: Stratix IV, Stratix V, Arria II, Arria V, Arria V GZ, Arria 10, Cyclone 10 LP, Cyclone IV, Cyclone V, MAX II, MAX V, and MAX 10 FPGA. [More](#)

Combined Files **Individual Files** **Additional Software** **Updates** !

Download and install instructions: [More](#)
[Read Intel FPGA Software v18.1 Installation FAQ](#)
[Quick Start Guide](#)

Select All

Quartus Prime Standard Edition
Size: 2.2 GB MD5: 6804D9CE1DF40E81EA8E175C79A34171

ModelSim-Intel FPGA Edition (includes Starter Edition)
Size: 1.1 GB MD5: 7FDBE5899A9929AEDD517F410079AA35

Devices
You must install device support for at least one device family to use the Quartus Prime software

Arria II device support
Size: 669.7 MB MD5: 3F397B96CCD63E70205391B92BE5AC62

Arria 10 device support ⓘ
Arria 10 device support Part 1
Size: 3.2 GB MD5: EA1E93E1C56DBEEA925E8E68E2BA2837
Arria 10 device support Part 2
Size: 2.5 GB MD5: 907CBB967E5F0BE384CB74FD458CD639
Arria 10 device support Part 3
Size: 4.3 GB MD5: F22EEDCA1CB53493BC297ED9C3A15760

Updates Available

2.2 KAYA FMC Loopack test board project setup

2.2.1 Hardware

- **Warning:** To avoid power supplies hot plug, check all power supplies switches positions are OFF.
- **Warning:** Use the correct hardware configuration for each bitstream.

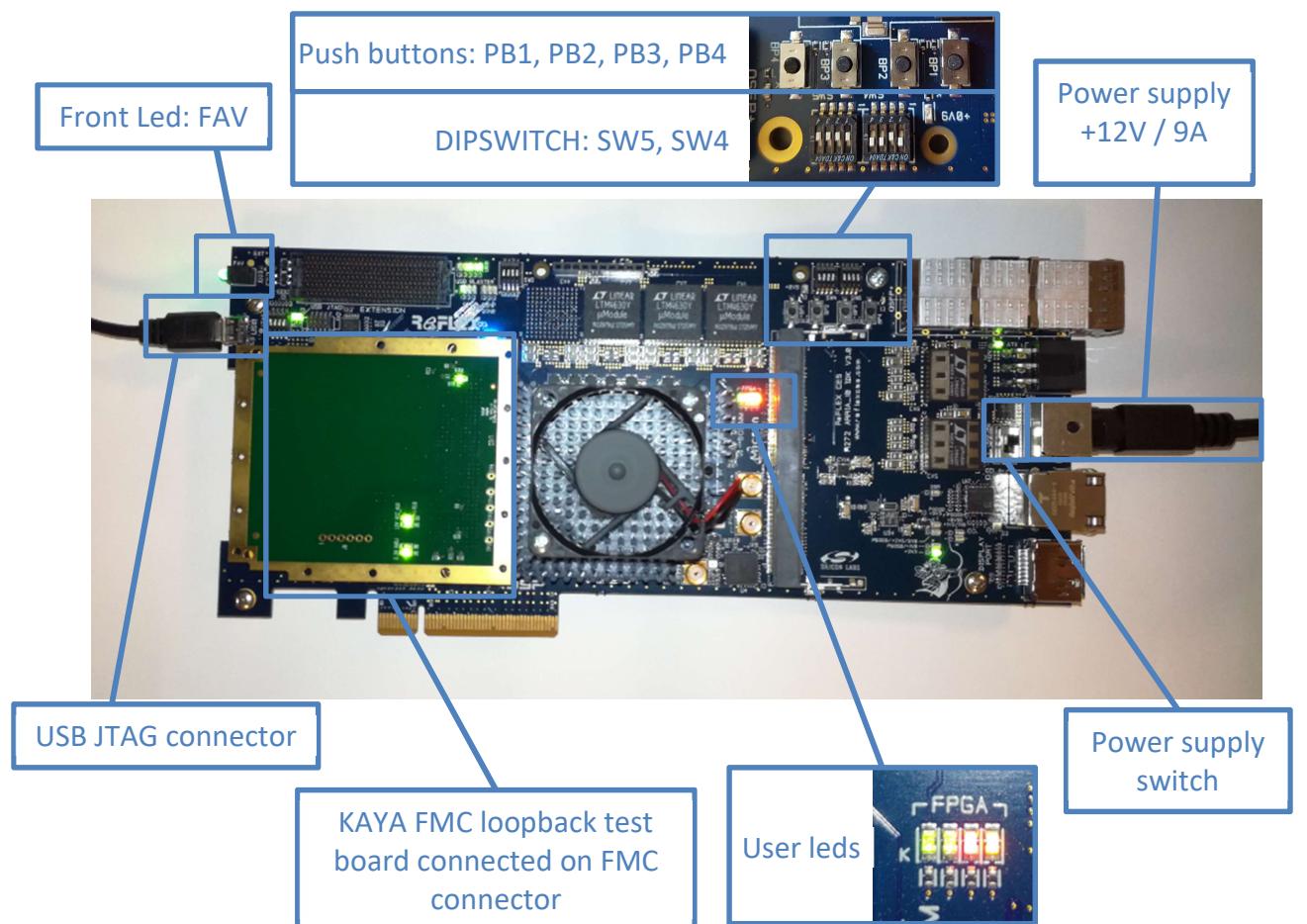


Figure 8: Attila Arria 10 GX Board with Kaya loopback test board

ESIstream

The Efficient Serial Interface

2.2.2 Software

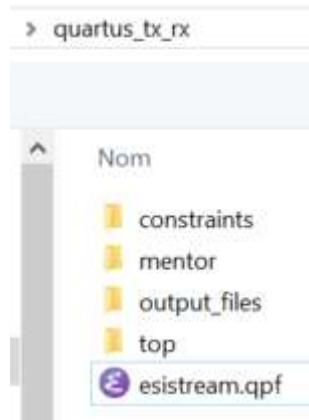
2.2.2.1 Quartus project

- Open Quartus 18.1 Standard Edition

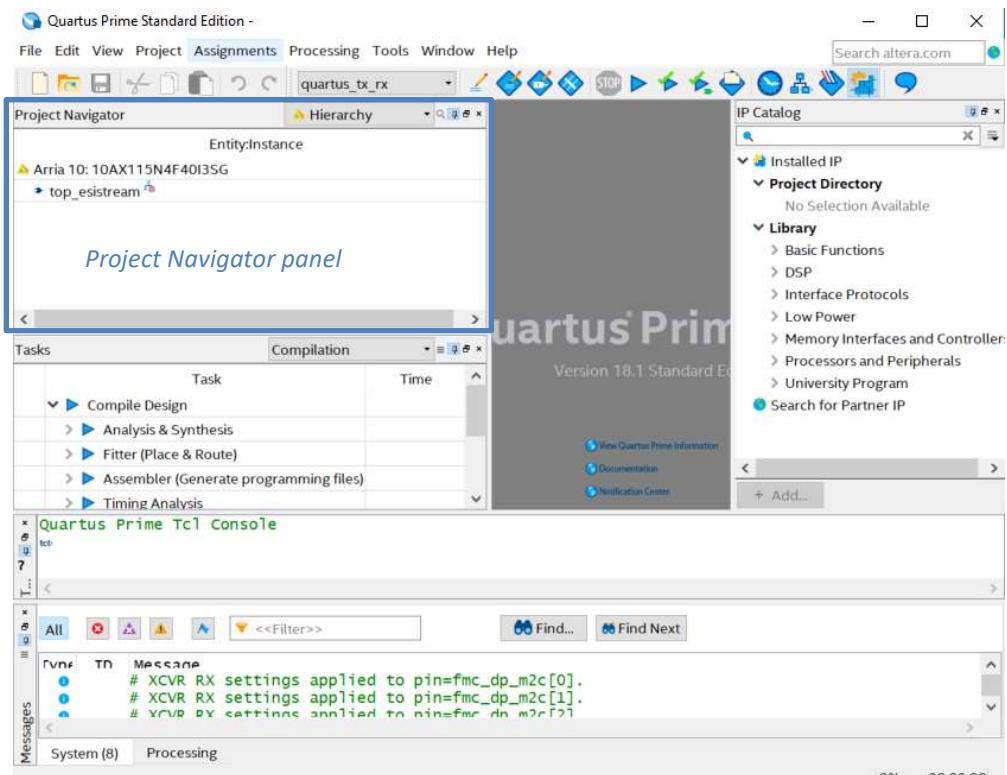


- File > Open Project... or Ctrl+J

- Go to package directory \quartus_tx_rx and open esistream.qpf project file



- Quartus should look like this:



ESIstream

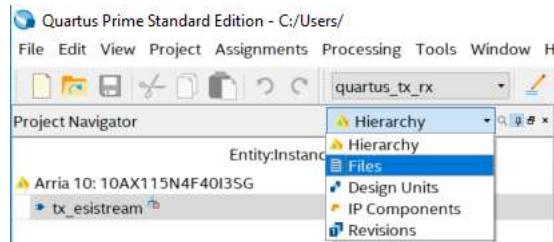
The Efficient Serial Interface

- Generate IP core synthesis and simulation files:

These files are used to compile the system in the quartus environment and to simulate the ESIstream IPs with the modelsim project. It is a one-time procedure if IPs are not modified. If the IP is modified, perform this procedure after modification of the IP.

- Select view Files:

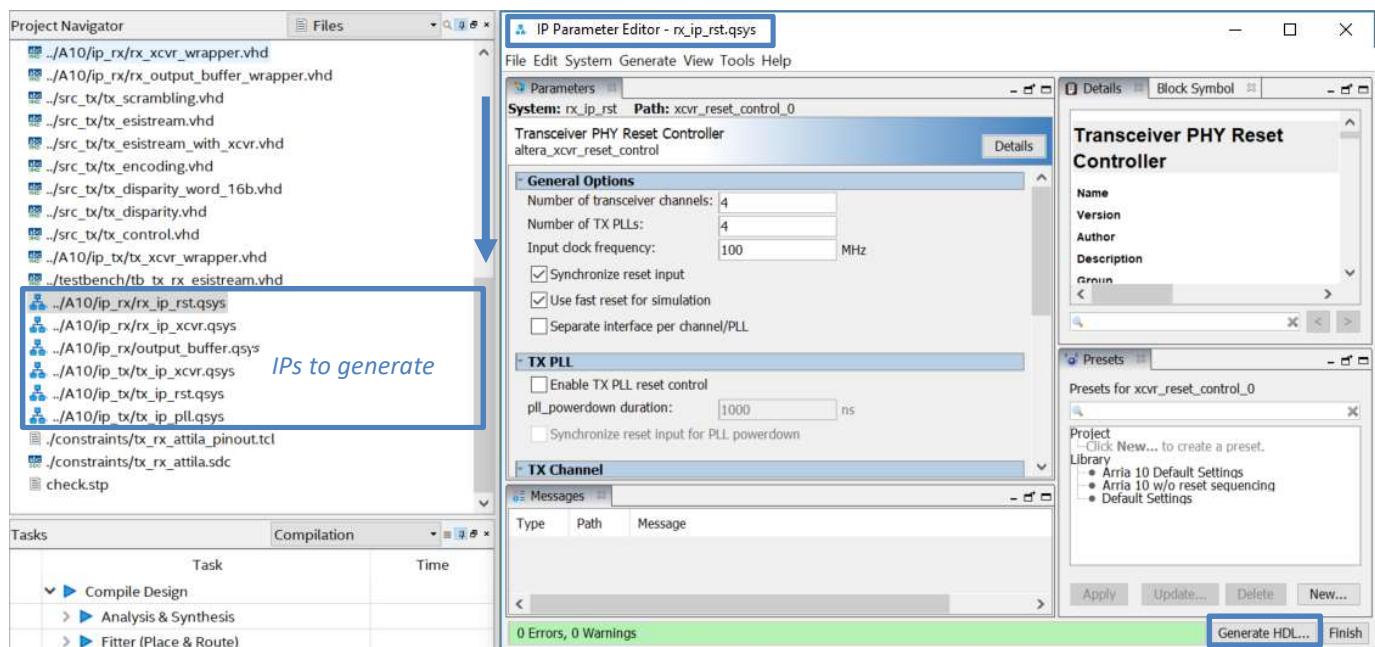
- In *Project Navigator* panel, select *Files* view in the drop-down menu:



- In the *Project Navigator Files* view down the scroll bar.

- For each IP:

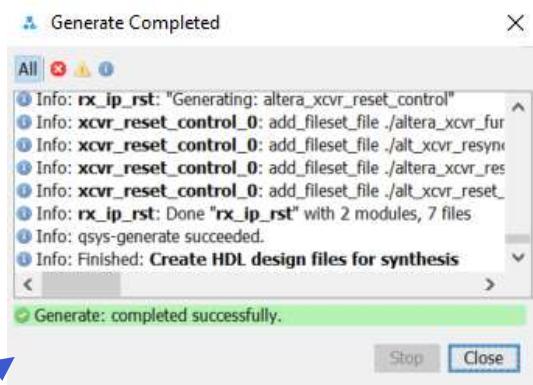
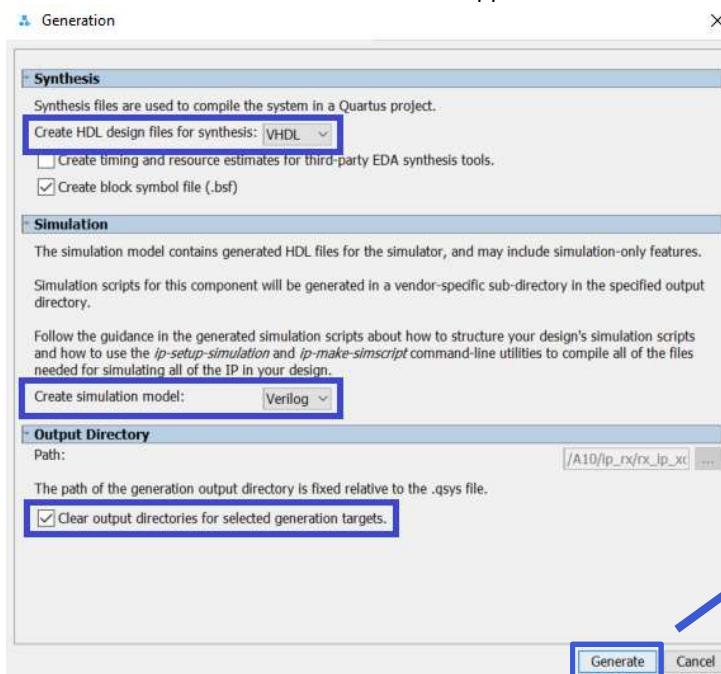
- Double-click on the IP or Right-click > Open
 - The *IP Parameter Editor* window appears.
 - Click on *Generate HDL...* button.



ESIstream

The Efficient Serial Interface

- The *Generation* window appears:



- Check the configuration as show on the above picture:
 - synthesis: *VHDL* - simulation: *Verilog* - Check checkbox: *clear output directories*...
 - Click on *Generate*
 - Wait completion of Generate and click on *Close* button
 - Click on *Finish* and **restart this procedure for all IPs**.

2.2.2.2 Generate bitstream

- Click on *Start Compilation (Ctrl + L)*.
- Wait end of design compilation.
 - Check Synthesis, Implementation end without critical warnings.
 - Check there is no Timing error.

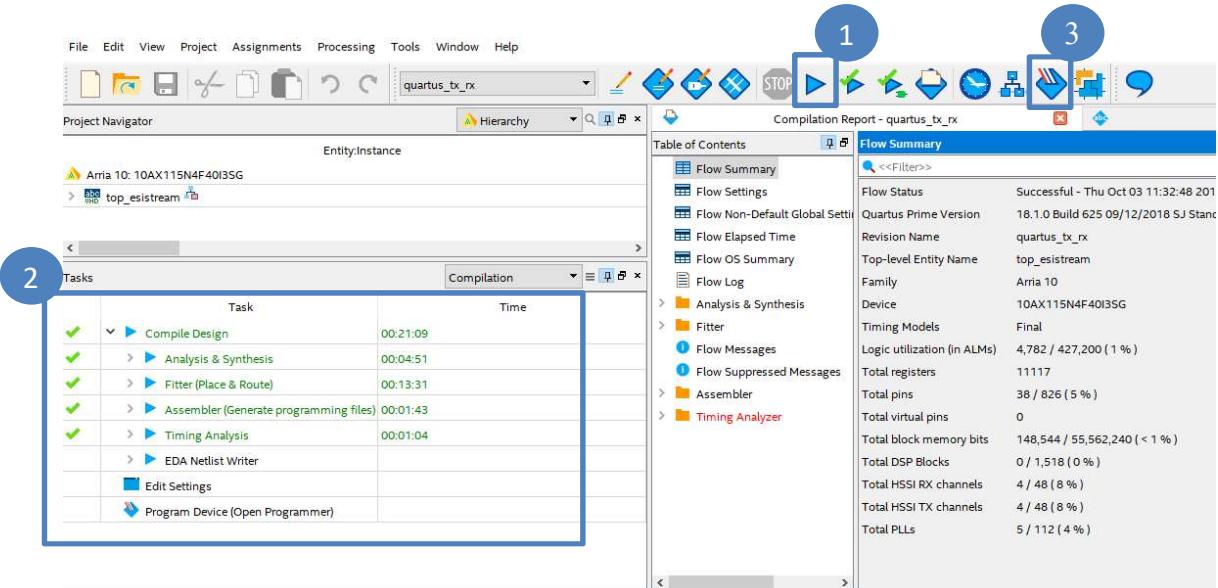
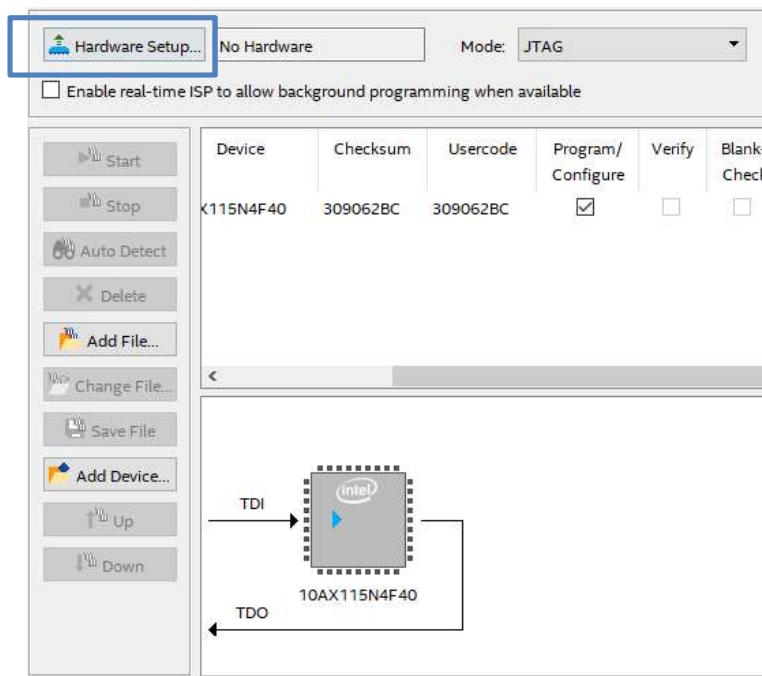


Figure 9: Quartus II, Generate bitstream and program the FPGA

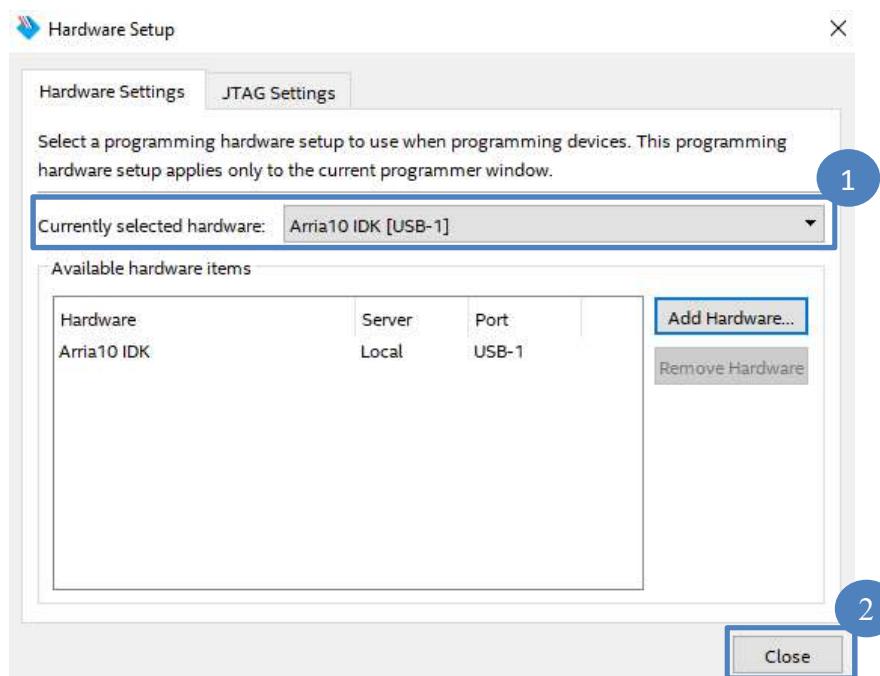
2.2.2.3 Test procedure

Once the bitstream is generated, the Arria 10 FPGA embedded on the Attila board can be loaded using JTAG programmer.

- Program the FPGA: click on Programmer button (see (3) on [Figure 10](#)), USB JTAG cable should be connected to the PC and the board should be powered on.
 - Click on *Hardware Setup*



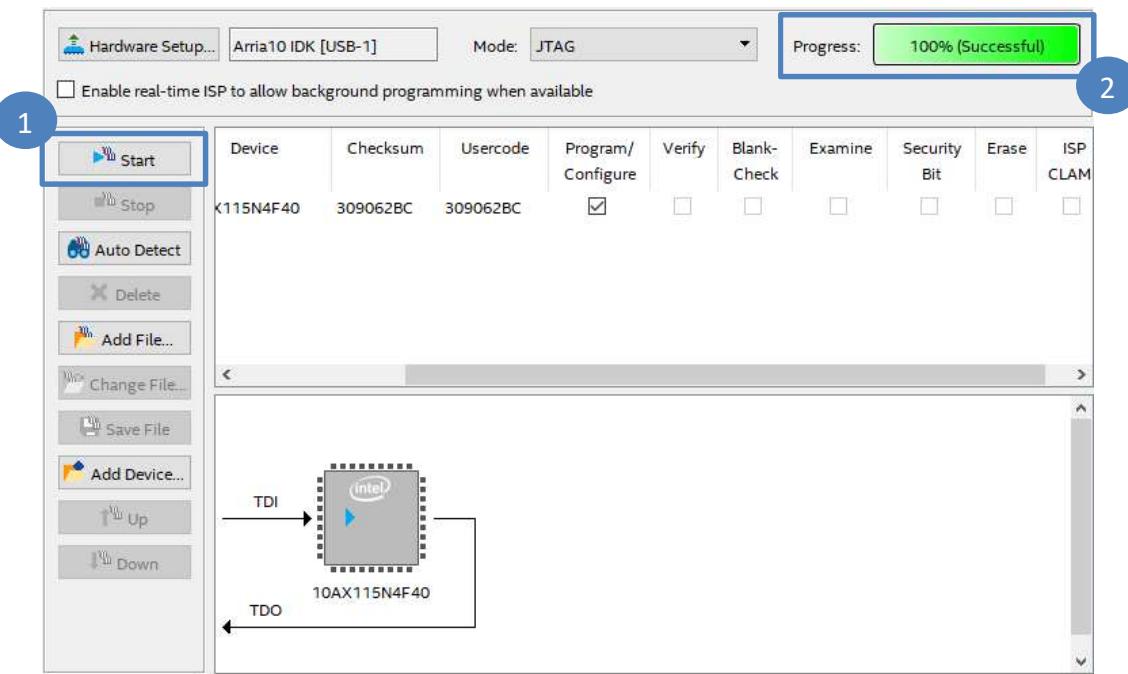
- Select hardware *Arria10 IDK [USB-1]*
 - Then, click on *Close* button



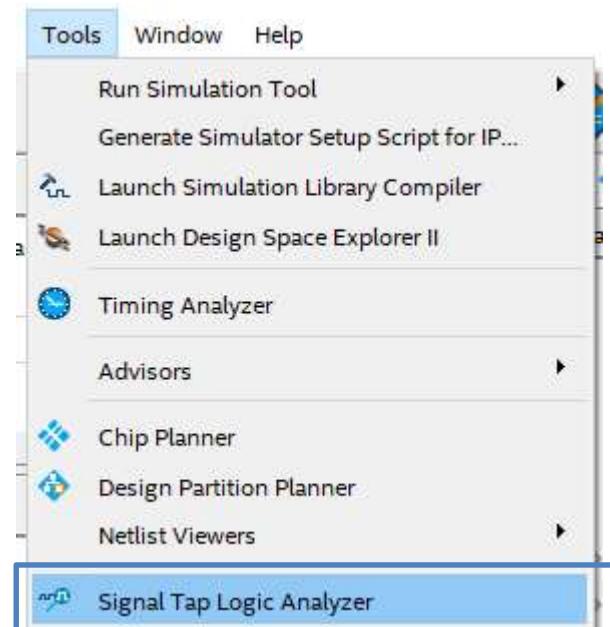
ESIstream

The Efficient Serial Interface

- Click on Start button to launch the FPGA programing.
 - Wait end of FPGA programing



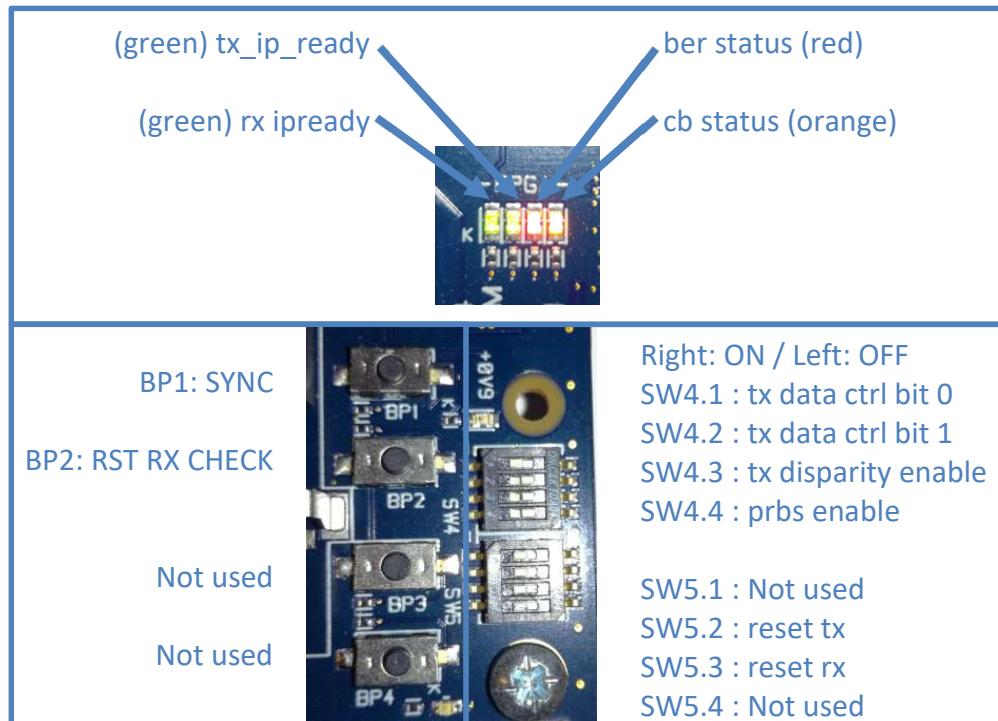
- Once the FPGA is programed, open the signal tap logic analyzer:
 - Tools > Signal Tap Logic Analyzer



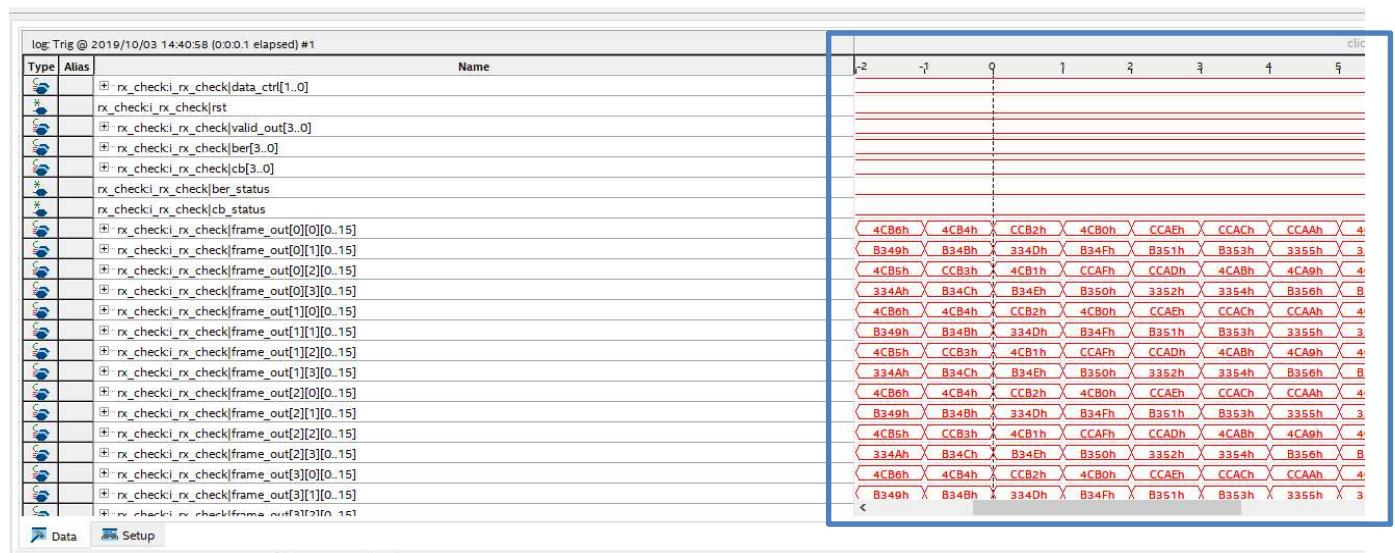
ESIstream

The Efficient Serial Interface

- Use switches SW4.1 and SW4.2 to configure the test pattern.
- SW4.3, SW4.4 should be ON and SW5.2,SW5.3 should be OFF.
- Start the ESIstream lanes synchronization sequence pushing BP1 (SYNC) push button.
 - Check Front Led FAV is ON and Green (means lanes are ready).
- Reset the rx_check module pushing BP1 push button to get a valid status on cb_status and ber_status.
- If all user led are ON, then there is no error.
- If led red is OFF then there is a bit error.
- If led orange is OFF then there is a clock bit error.

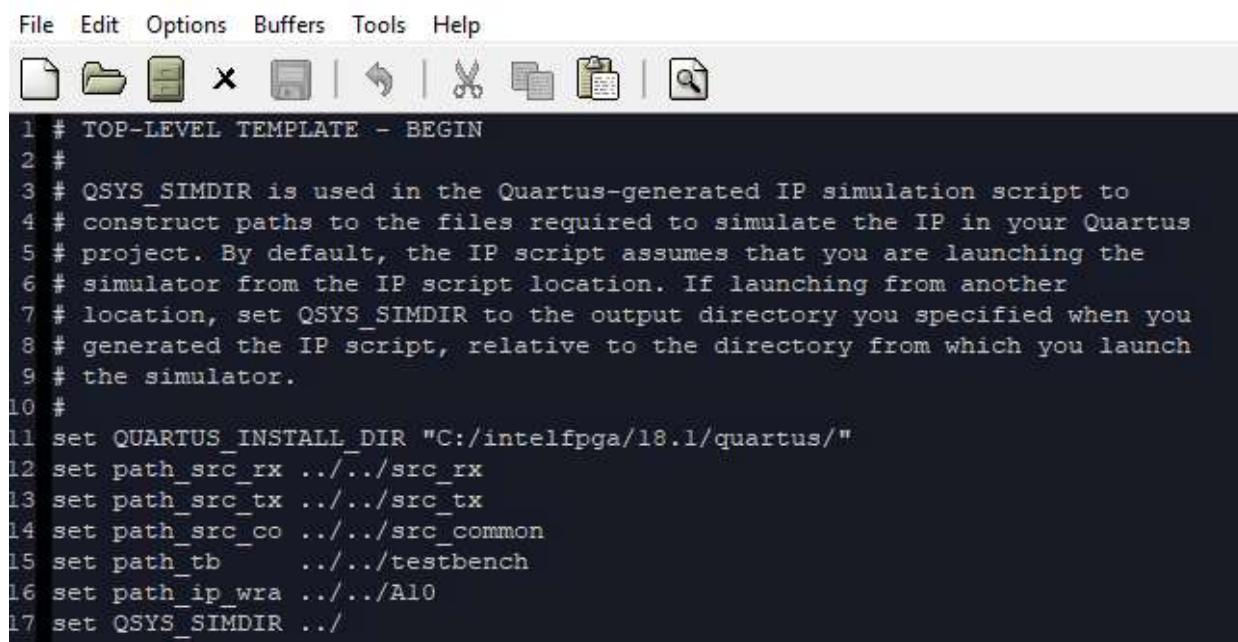


- In quartus, display signals in the Signal Tap Logic Analyser clicking on *Run analysis* button.



2.2.2.4 *Modelsim, simulation scripts setup*

Check and update the path of the variable *QUARTUS_INSTALL_DIR* with your Quartus installation directory path in the files *compile.do* and *simulate.do* located in package directory *\modelsim\tx_a10_rx_a10*



A screenshot of a text editor window showing a simulation script. The menu bar includes File, Edit, Options, Buffers, Tools, and Help. The toolbar contains icons for file operations like Open, Save, and Cut/Paste. The script content is as follows:

```
1 # TOP-LEVEL TEMPLATE - BEGIN
2 #
3 # QSYS_SIMDIR is used in the Quartus-generated IP simulation script to
4 # construct paths to the files required to simulate the IP in your Quartus
5 # project. By default, the IP script assumes that you are launching the
6 # simulator from the IP script location. If launching from another
7 # location, set QSYS_SIMDIR to the output directory you specified when you
8 # generated the IP script, relative to the directory from which you launch
9 # the simulator.
10 #
11 set QUARTUS_INSTALL_DIR "C:/intelfpga/18.1/quartus/"
12 set path_src_rx ../../src_rx
13 set path_src_tx ../../src_tx
14 set path_src_co ../../src_common
15 set path_tb      ../../testbench
16 set path_ip_wra ../../A10
17 set QSYS_SIMDIR ...
```

ESIstream

The Efficient Serial Interface

2.2.2.5 ESIstream 64-bit SERDES IP(s) simulation

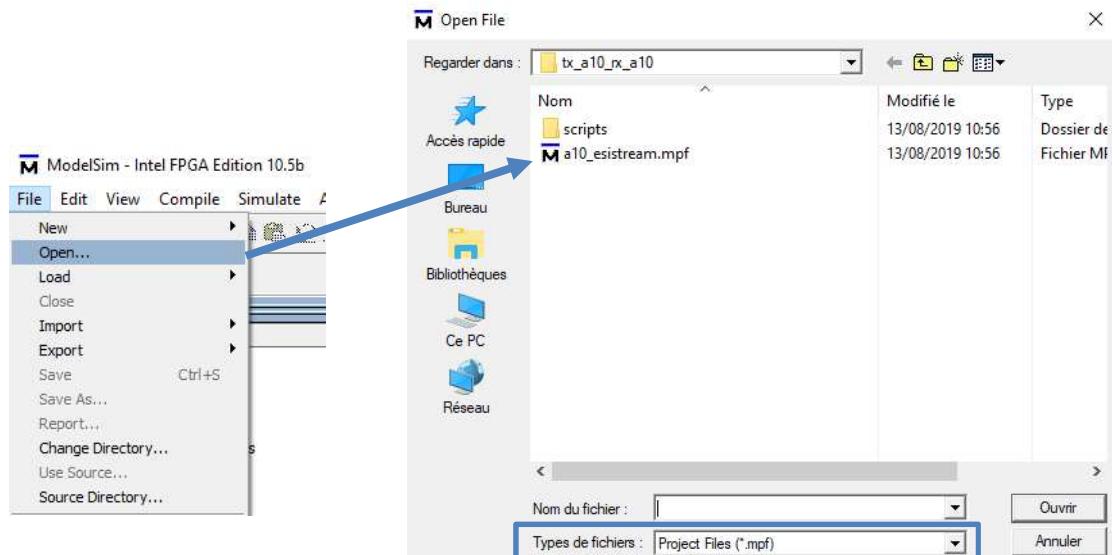
2.2.2.5.1 Launch simulation

- Open Modelsim Intel FPGA Edition 10.5b



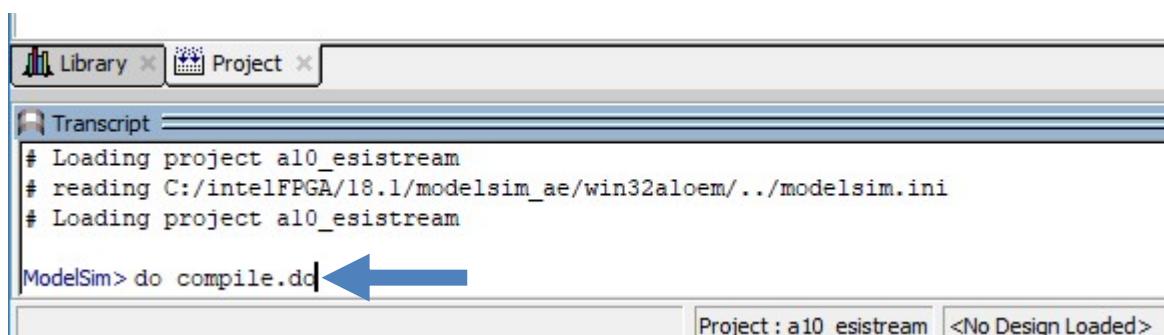
- Open project:

- File > Open...
- Go to package directory: \modelsim\tx_a10_rx_a10\
- Select *Project Files (*.mpf)* as files type.
- Open *a10_esistream.mpf*
- Open



- Then, in transcript panel, use command:

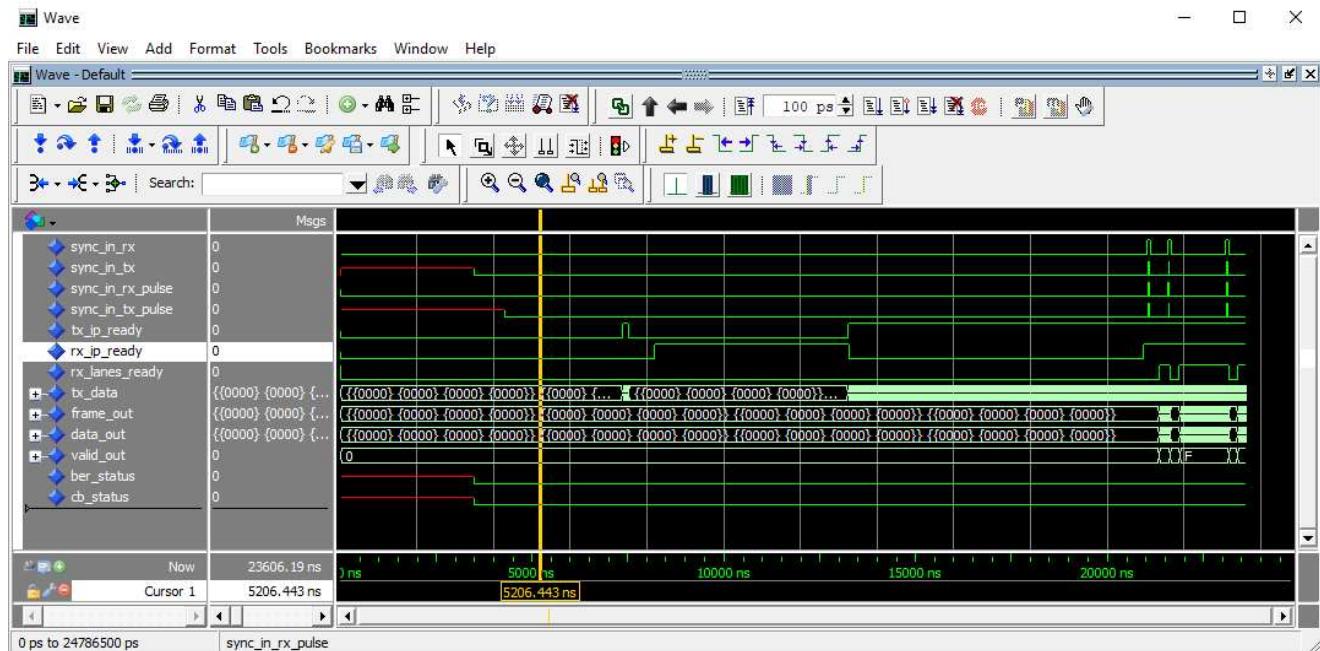
- *do compile.do* allow compilation of the whole project (including Intel FPGA IPs) and launch of the simulation.
- *do simulate.do* allows compilation of just the vhdl source files and launch of the simulation.



ESIstream

The Efficient Serial Interface

At the end of compilation, the simulation starts and ends automatically.



2.2.2.5.2 Test-bench description

The testbench instantiates both the ESIstream transmitter (tx) and receiver (rx) ips.

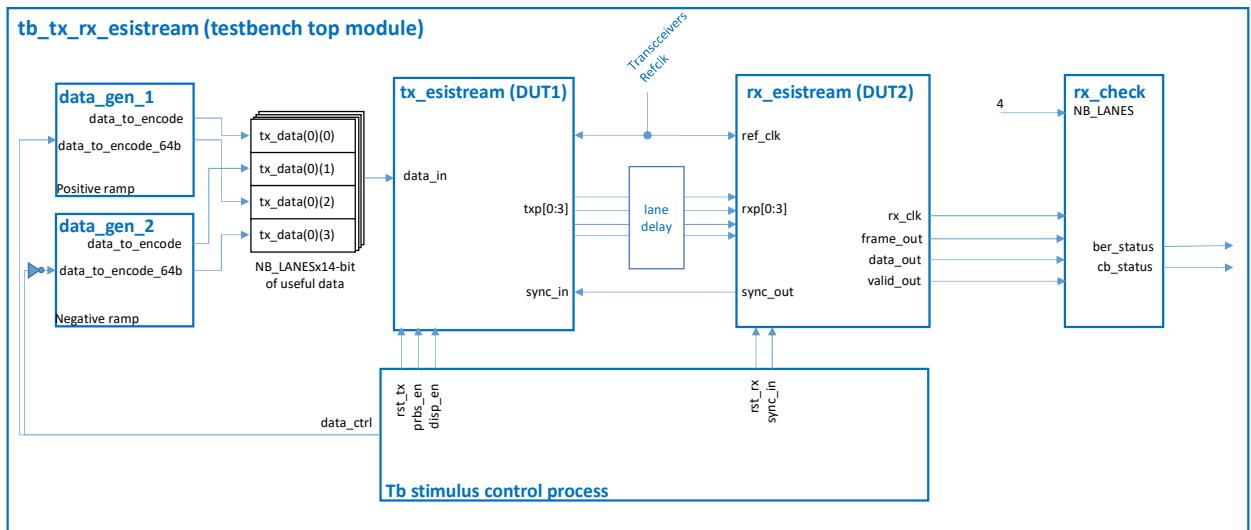


Figure 10: 64-bit SERDES testbench Block diagram

Two data generator modules (data_gen) generate known data at the transmitter input.

According to the value of data generator control input signal *d_ctrl* the generated data can be a 14-bit positive ramp, a 14-bit negative ramp, zero only or one only.

<i>d_ctrl</i> value	waveform
"00"	"0000000000000000"
"01"	Positive ramp For 64-bit implementation: - Odd values came out on <i>data_out</i> - Even values came out on <i>data_out_64b</i>
"10"	Negative ramp For 64-bit implementation: - Odd values came out on <i>data_out</i> - Even values came out on <i>data_out_64b</i>
"11"	"1111111111111111"

The validation module (rx_check) allows checking received data values. It reports an error when there is a bad value in a data field or a clock bit field of the ESIstream frames.

ber_status	'0': No error '1': Bit error
cb status	'0': No error '1': Clock bit error

2.3 EV12AQ600 FMC test board project setup

2.3.1 Hardware

- **Warning:** To avoid power supplies hot plug, check all power supplies switches positions are OFF.
- **Warning:** Use the correct hardware configuration for each bitstream.

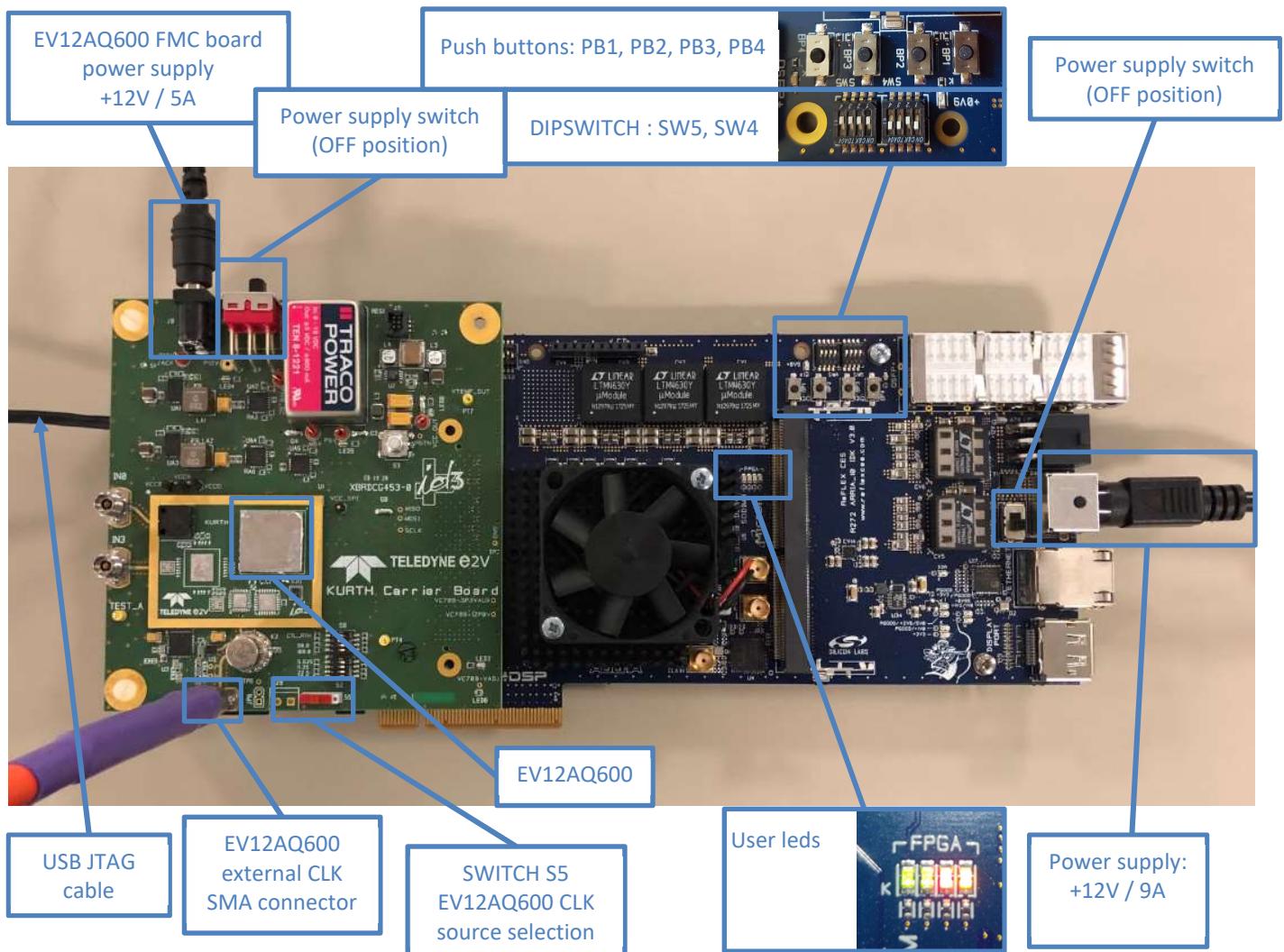


Figure 11: Attila Arria 10 GX Board with EV12AQ600 FMC test board

For more information on the EV12AQ600 FMC test board, contact GRE-HOTLINE-BDC@Teledyne.com.

2.3.1.1 Dipswitch configuration

Dipswitch	Function	Description
SW4.1	DC balance enable	EV12AQ600 disparity processing enabled when ON (a SPI write of DATA_MODE_SEL register must be done).
SW4.2	data enable	EV12AQ600 data enabled when ON else all data bits set to logic 0 (a SPI write of DATA_MODE_SEL register must be done).
SW4.3	external PLL enable	When ON external PLL located on the EV12AQ600 FMC board test board is used to clock the ADC EV12AQ600. Else an external clock should be provided on CLK SMA + S5 switch right position.
SW4.4	prbs enable	RX ESIstream IP scrambling processing enabled when ON. EV12AQ600 prbs processing enabled when ON (a SPI write of DATA_MODE_SEL register must be done).
SW5.1	NC	Not connected
SW5.2	NC	Not connected
SW5.3	NC	Not connected
SW5.4	OTP memory enable	Load OTP memory when ON.

2.3.1.2 Leds status

user leds	Output	Description
Green 1	Transceivers ip ready status	OFF: PLL unlocked or reset not done. ON: PLL locked and reset done
Green 2	isrunning status	OFF: aq600_interface.vhd isrunning fsm state not reached. ON: RX check tests can be launched pushing SW_W and data can be monitored using vivado ILA.
Orange	Bit error status	ON: No error OFF: Bit error After lanes synchronized, rst check push button (SW_W) should be pressed for 1-second minimum to start the test and get a valid status.
Red	Clock bit error status	ON: No error OFF Clock bit error After lanes synchronized, rst check push button (SW_W) should be pressed for 1-second minimum to start the test and get a valid status.

FAV	Output	Description
Red	User rst push button status	ON reset pushbutton PB4 on OFF reset pushbutton PB4 off
Blue	User sync push button status	ON SYNC pushbutton PB1 on OFF SYNC pushbutton PB1 off
Green	Lanes ready status	ON Lanes synchronization failed OFF Lanes synchronized succeed

2.3.1.3 Push-buttons description

S5 switch position	Control	Description
PB1	SYNC signal generation	
PB2	RX check module reset	Push button when led Green 1 – Green 2 are ON
PB3	Start / Stop	Push button to start or stop the test.
PB4	FPGA logic reset	

2.3.1.4 S5 switch configuration

S5 switch position	Function	Description
Left	PLL external CLK selected	This mode is not available without modification of the file spi_dual_master_fsm.vhd. To use this mode or for more information contact: GRE-HOTLINE-BDC@Teledyne.com
Right	SMA external CLK selected	Default position

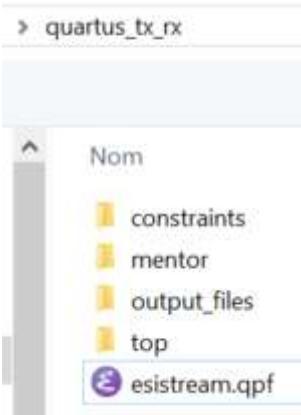
2.3.2 Software

2.3.2.1 Quartus project

- Open Quartus 18.1 Standard Edition



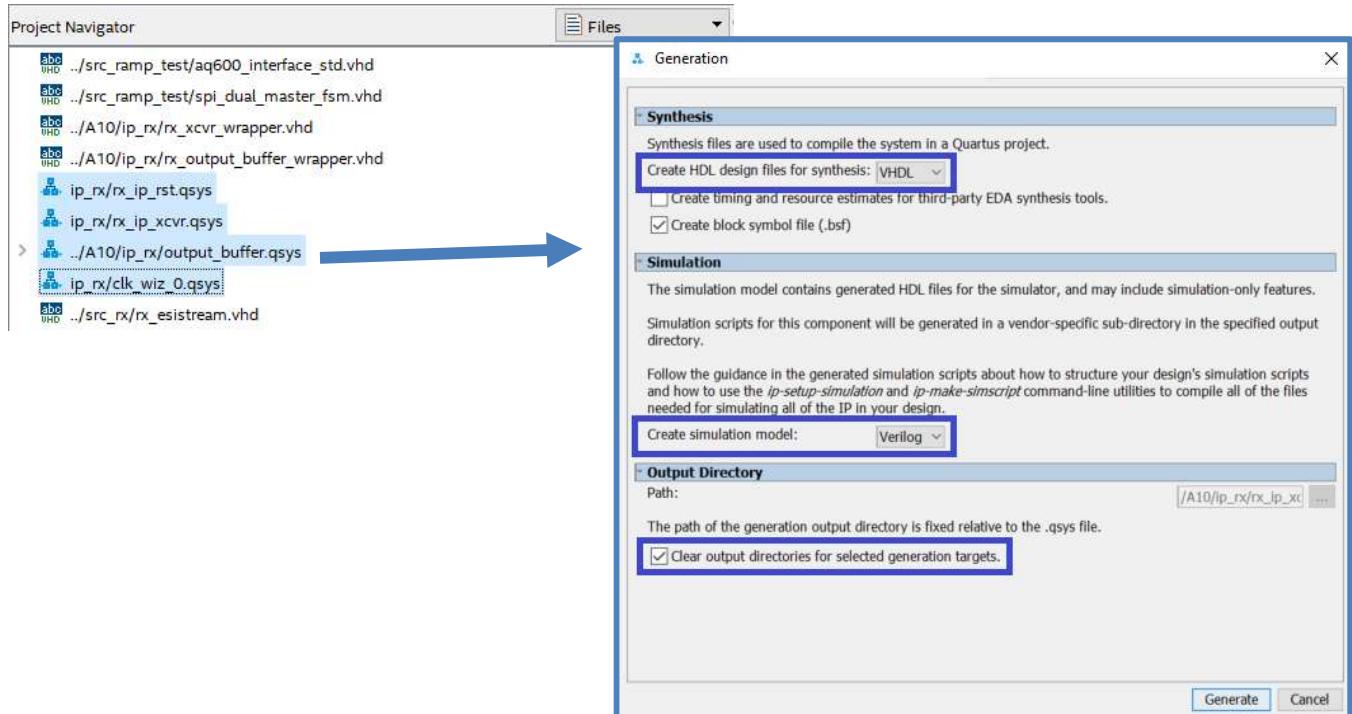
- File > Open Project... or Ctrl+J
 - Go to package directory `\quartus_rx_10g\` and open `esistream.qpf` project file



ESIstream

The Efficient Serial Interface

- Generate HDL for all IP using the configuration below:



When the project has been created and the IP generated, you can start compiling or modifying the example design like any Quartus project.

- Generate bitstream.
- Program the FPGA.
- Open Signal Tap Logic Analyzer

2.3.3 RAMP mode test

The ADC EV12AQ600 can be configured in ramp mode. In ramp mode, the data encoded corresponds to a 12-bit ramp value with only the even values on lane 1 and the odd values on lane 2.

See below the chronogram of the ramp test mode. The data shown in the following figure only presents the 14 bits data from the ADC (12 bits sample value plus 2 control bits CB1 and CB2) and does not include the encoding of the ESIstream protocol which is used on the serial interface.

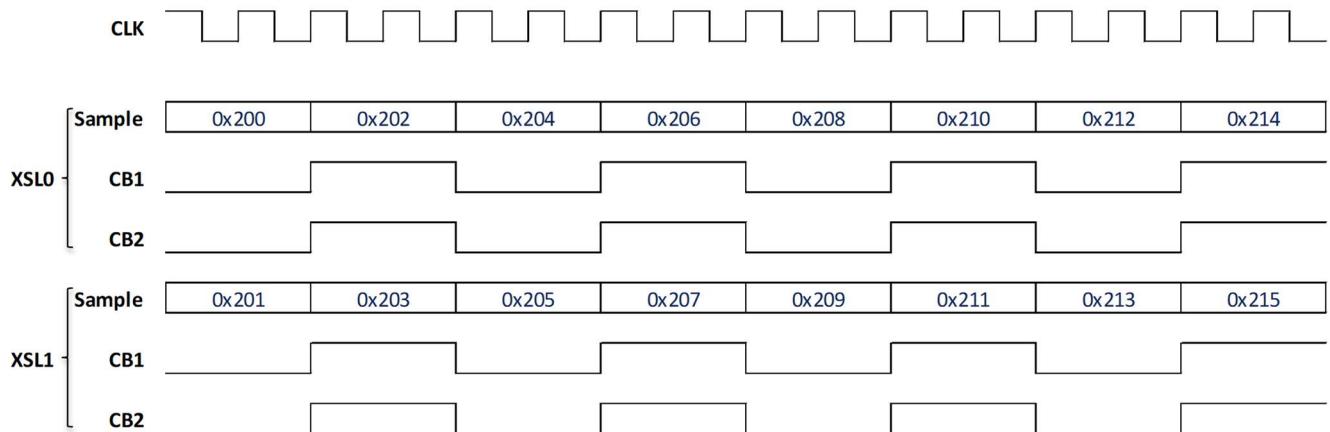


Figure 12: Chronogram of the Ramp test mode

The rx_check module allows checking that the received data match this pattern and that the clock bit is valid.

A state machine located in the aq600_interface.vhd file allows configuring the EV12AQ600 ADC in ramp mode and launching the ESIstream serial link synchronization (SYNC signal).

This state machine starts when the user pushes the “Start / Stop” push-button.

- Functional State Machine (FSM) description:
 - 1- Reset EV12AQ600 and FPGA transceivers (10 µs min). During the RSTN pulse, CSN must be held high and SCLK held low. The CLK must be provided before the RSTN pulse. The CLK can start before or after the power-up.
 - 2- Wait OTP memory wakes up (1 ms min).
 - 3- Send SPI instruction WRITE @0x0001 =0b1 to EV12AQ600 ADC. The EV12AQ600 OTP memory cells are loaded into registers. There must be at least 1 ms between the RSTN pulse and this instruction.
 - 4- Send SPI instruction WRITE @0x80BA =0b1 to enable ramp mode.
 - 5- Send SYNCTRIG pulse to the ESIstream IP and to the ADC (default: the trigger mode is disabled).
 - 6- Normal operation when the FSM reaches the *isrunning* state). Data monitoring at the output of the RX ESIstream IP is possible using the Vivado ILA interface.

ESIstream

The Efficient Serial Interface

- Ramp mode test procedure:
 - 1- Connect the EV12AQ600 FMC board to FPGA evaluation board FMC connector.
 - 2- Check that the EV12AQ600 FMC board power switch is on OFF position.
 - 3- Connect EV12AQ600 FMC board +12V / 5A power supply cable.
 - 4- Check that the FPGA evaluation board power switch is on OFF position.
 - 5- Connect FPGA evaluation board +12V / 5A power supply cable.
 - 6- Connect JTAG cable between the FPGA evaluation board and the PC.
 - 7- Slide EV12AQ600 FMC board switch (S5) to the position allowing the use of the SMA external CLK.
 - a. Use a signal generator to generate the EV12AQ600 CLK signal (6.4 GHz / +12 dBm).
 - 8- FPGA evaluation board dipswitch positions **SW4.3 should be OFF**, SW4.1, SW4.2 and SW4.4 positions should be all ON.
 - 9- Power on EV12AQ600 FMC board and the FPGA evaluation board.
 - 10- Load FPGA using Quartus programmer. The bitstream must be generated before opening the programmer.
 - 11- Push-button (2 second) "FPGA logic reset" (PB4).
 - 12- Push-button (2 second) "Start / Stop" (PB3).
 - 13- Check leds Green 1 and Green 2 are ON.
 - 14- Push button (2 second) "RX check module reset" (PB2)
 - 15- Signal Tap Logic Analyzer acquisition can be performed.

ESIstream

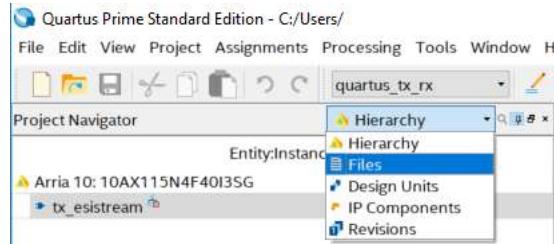
The Efficient Serial Interface

3. ANNEX A: SWITCH PROJECT TO A 32-BIT SERDES

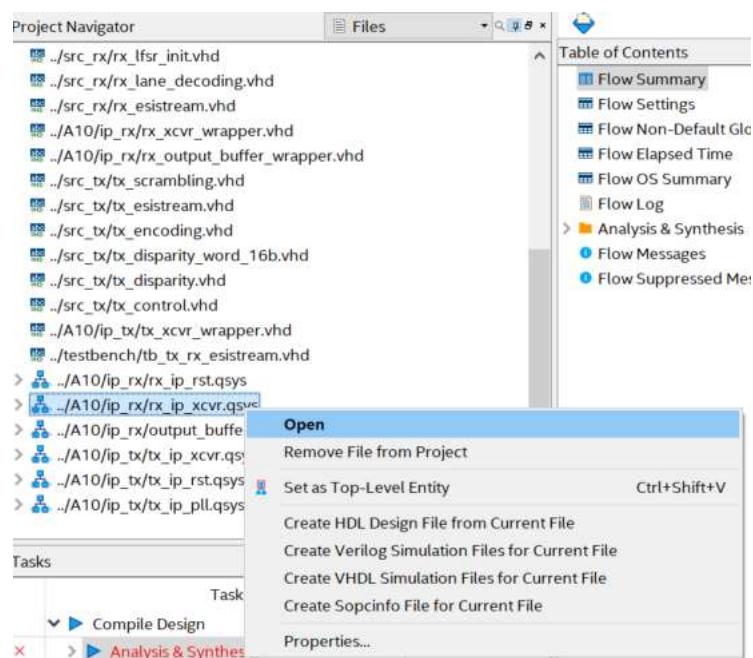
The logic has not been tested at higher logic speed than 64b serdes and may not support 32b without modification or optimization. It depends on the serial lanes data rate and FPGA speed grade.

Moving to 32b serdes allows to reduce the ESIstream IP latency.

- Open Quartus prime 18.1 standard edition:
- In project navigator panel, select Files view.



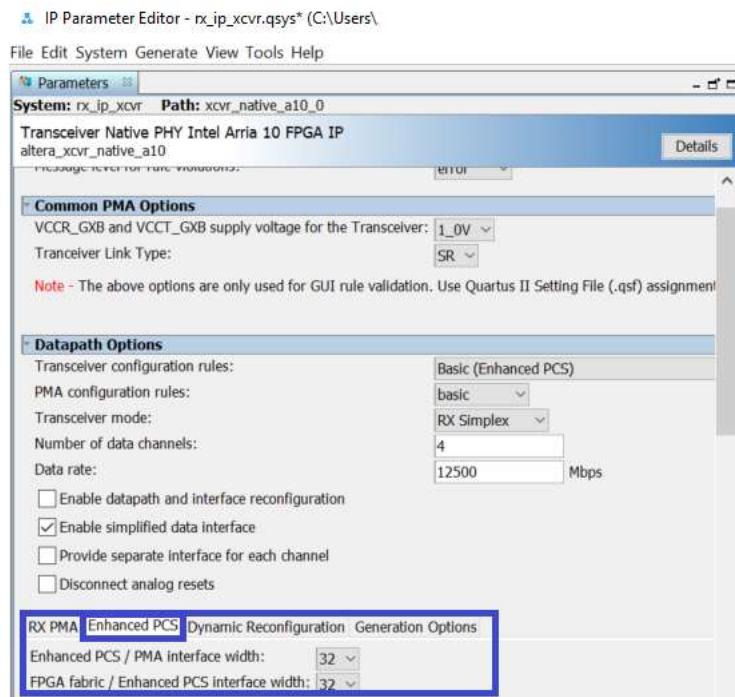
- Scroll down to find transceivers IP(s):
 - o Open rx transceiver:
 - o Right click + Open or Double click on rx_ip_xcvr.qsys



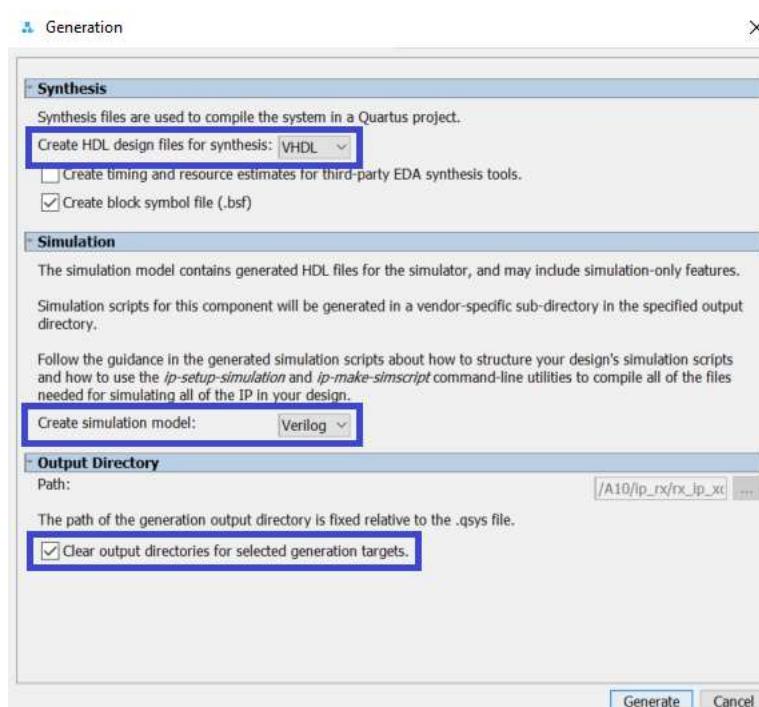
ESIstream

The Efficient Serial Interface

- Scroll down and go to Enhanced PCS panel
 - o Change Enhanced PCS / PMA interface width to 32
 - o Change FPGA fabric / Enhanced PCS interface width to 32



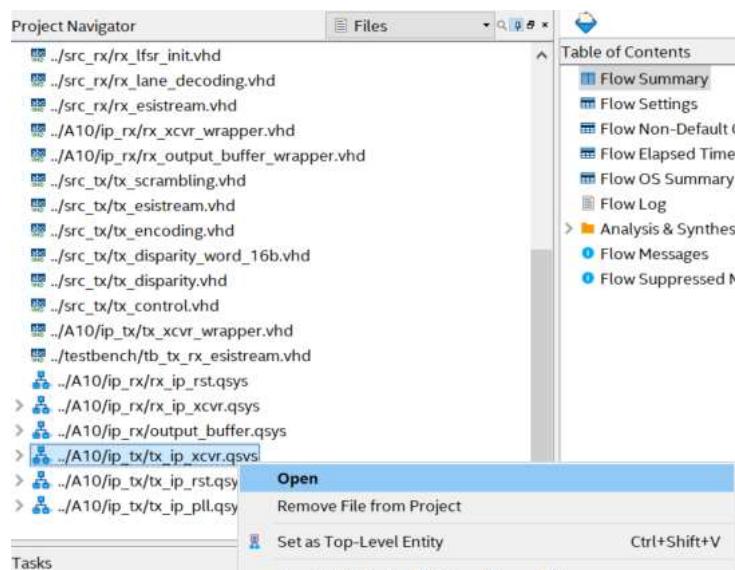
- Click on Generate HDL... button:
 - o Check configuration as picture below (synthesis: VHDL/ simulation: Verilog / clear output directories...):
 - Generate
 - Close
 - Wait for end of generate
 - Close
 - Finish



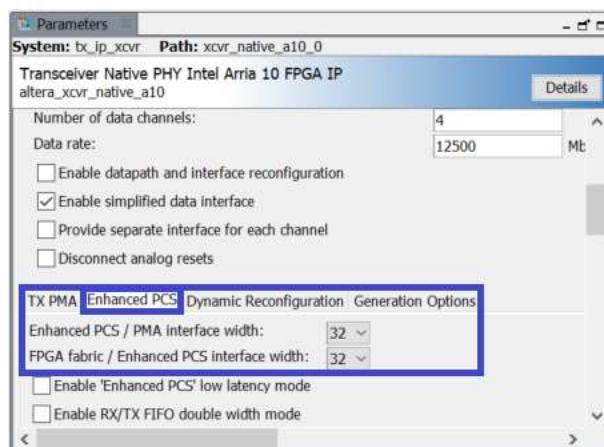
ESIstream

The Efficient Serial Interface

- Open tx transceiver:
 - o Right click + Open or Double click on rx_ip_xcvr.qsys
 - o Repeat operation for tx_ip_xcvr.qsys



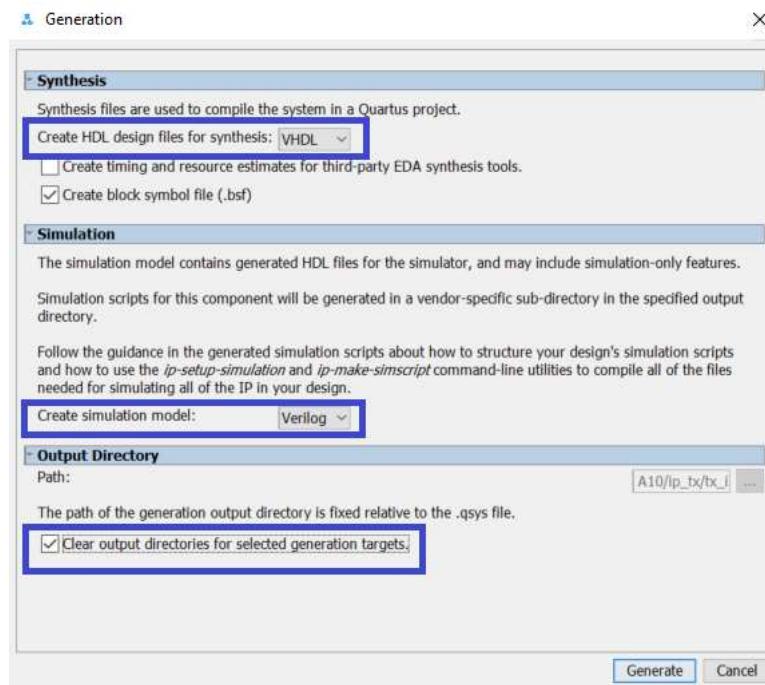
- o Scroll down and go to Enhanced PCS panel
 - Change Enhanced PCS / PMA interface width to 32
 - Change FPGA fabric / Enhanced PCS interface width to 32



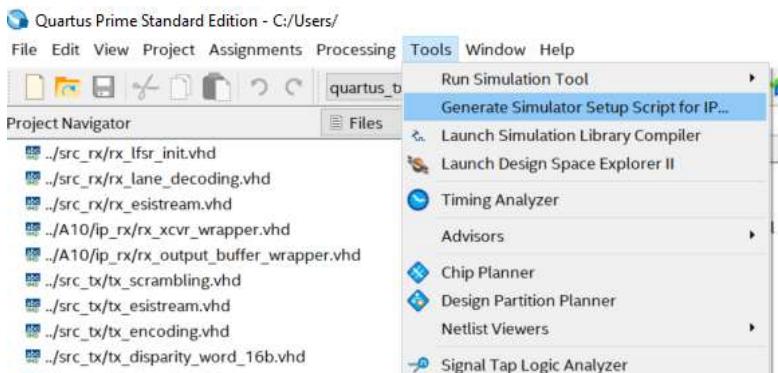
- o Check configuration as picture below (synthesis: VHDL/ simulation: Verilog / clear output directories...):
 - Generate
 - Close
 - Wait for end of generate
 - Close
 - Finish

ESIstream

The Efficient Serial Interface



- Tools > Generate Simulator Setup Script for IP...
 - OK
 - Only mentor folder is used.



- Open modelsim
 - Open project
 - Launch do compile.do script

ESIstream

The Efficient Serial Interface

4. ANNEX B: ATTILA BOARD INFORMATION

It is possible to test the TX and RX IP using an [Attila board](#) from [Reflex-CES](#) with a [KAYA FMC loopback test board](#) from [Kaya Instruments](#).

The board is based on the Intel Arria 10 GX FPGA in the FBGA F40 package (FPGA reference is 10AX115N4F40I3SG).

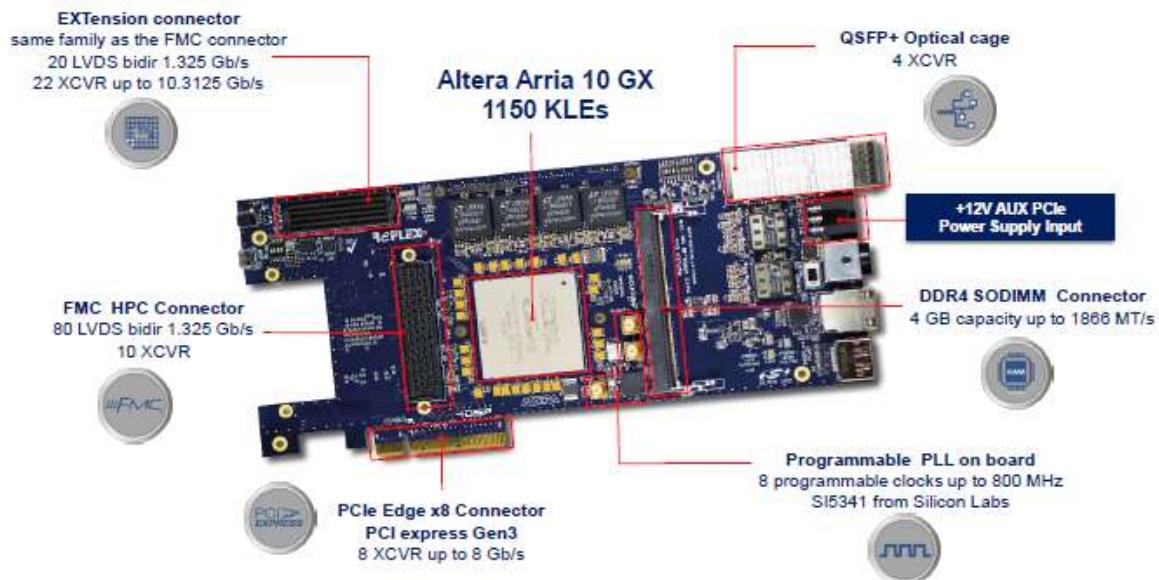


Figure 13: Attila Arria 10 GX Board