AURORA

Perseus Examples

October 2016

Revision history

Revision	Date	Comments
0.9	September 2013	First draft
1.0	October 2013	Formatting and released version
1.1	June 2014	Added *sudo* to all shell script launch procedures in Linux Up to date for Software Tools Release 6.6
1.2	August 2015	Restructured document and content updated for BAS 7.0
1.3	November 2015	Add modifications for Perseus611X
1.4	October 2016	Added Aurora Loopback example for PicoSDR 2nd Generation Software Tools

Table of Contents

List of Figures

1 FPGA Designs Description and Generation

1.1 Inter-carrier Aurora Transfers Demos Description

A set of FPGA designs showcasing usage of the Aurora core and how it interfaces with user logic are made available in the software suite installation files. They reside in the *examples**aurora* directory. These FPGA designs are available for both Perseus601x and Perseus611x and are designed using either Nutaq's Model Based Design Kit or Xilinx Platform Studio directly (BSDK).

Project files path	Design methodology	Carrier
perseus601x\bsdk	Xilinx Platform Studio (BSDK)	Perseus601x
perseus601x\mbdk	Nutaq's Model Based Design Kit (MBDK)	Perseus601x
perseus611x\bsdk	Xilinx Platform Studio (BSDK)	Perseus611x
perseus611x\mbdk	Nutaq's Model Based Design Kit (MBDK)	Perseus611x

Table 1. Available FPGA designs depiction

These four designs demonstrate bidirectional transfers between two carriers using Aurora. In both directions, ramps are generated internally in the FPGA and transmitted using each Aurora channel and verified on the other Perseus. The RTDEx Test core is used to transmit and validate ramps even though no RTDEx core is instantiated in the FPGA and no RTDEx transfers actually take place in this demo.

Perseus601x designs use three Aurora core, connected to FPGA GTX ports, which are themselves routed to AMC backplane ports 4 to 7, 8 to 11 and 17 to 20. A single 3-channel RTDEx Test core is instantiated in each FPGA. Figure 1-1. Perseus601x Aurora Demo schematicFigure 1-1 shows the Perseus601x demos schematic.

Perseus611x designs use seven Aurora cores, connected to FPGA GTX ports, which are themselves routed to connectors:

- RTM #1 0-3
- RTM #1 4-7
- RTM #1 8-11
- RTM #2 0-3
- RTM #2 4-7
- RTM #2 8-11
- RTM #2 12-15

A single 7-channel RTDEx Test core is instantiated in each FPGA. Figure 1-2 shows the Perseus611x demos schematic.

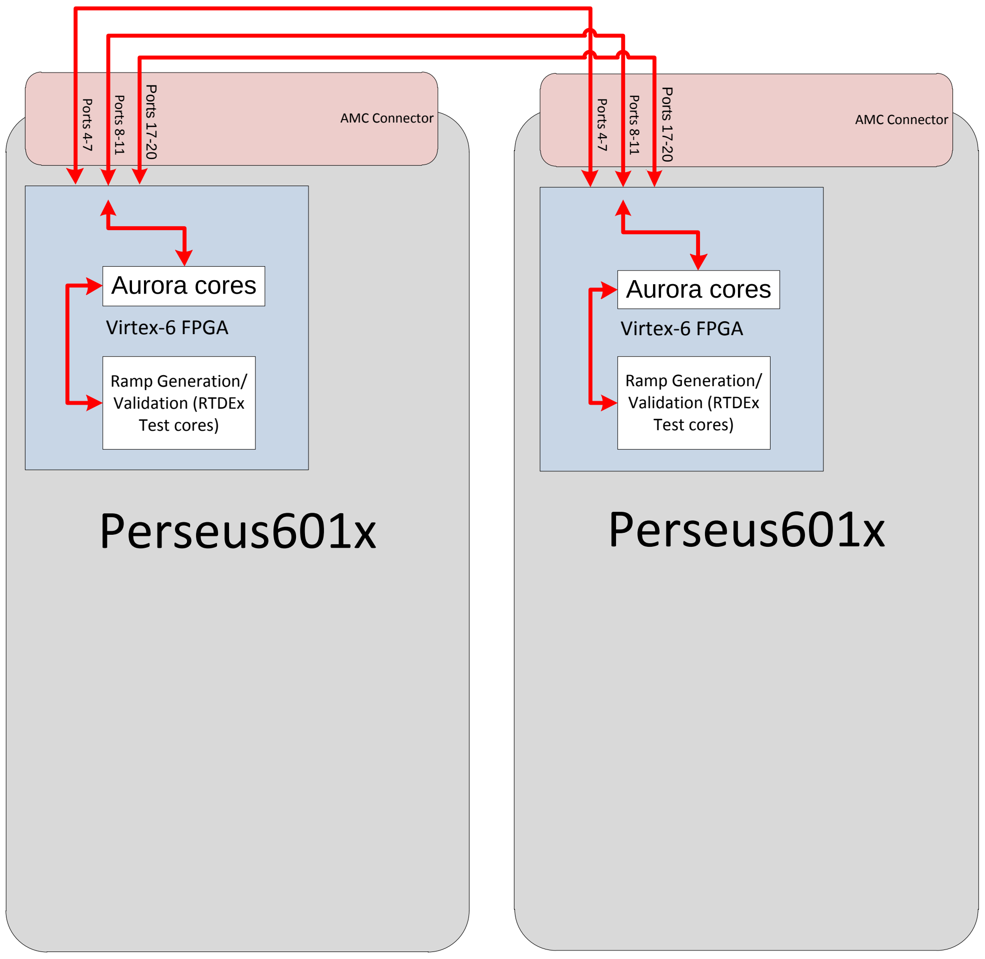


Figure 1-1. Perseus601x Aurora Demo schematic

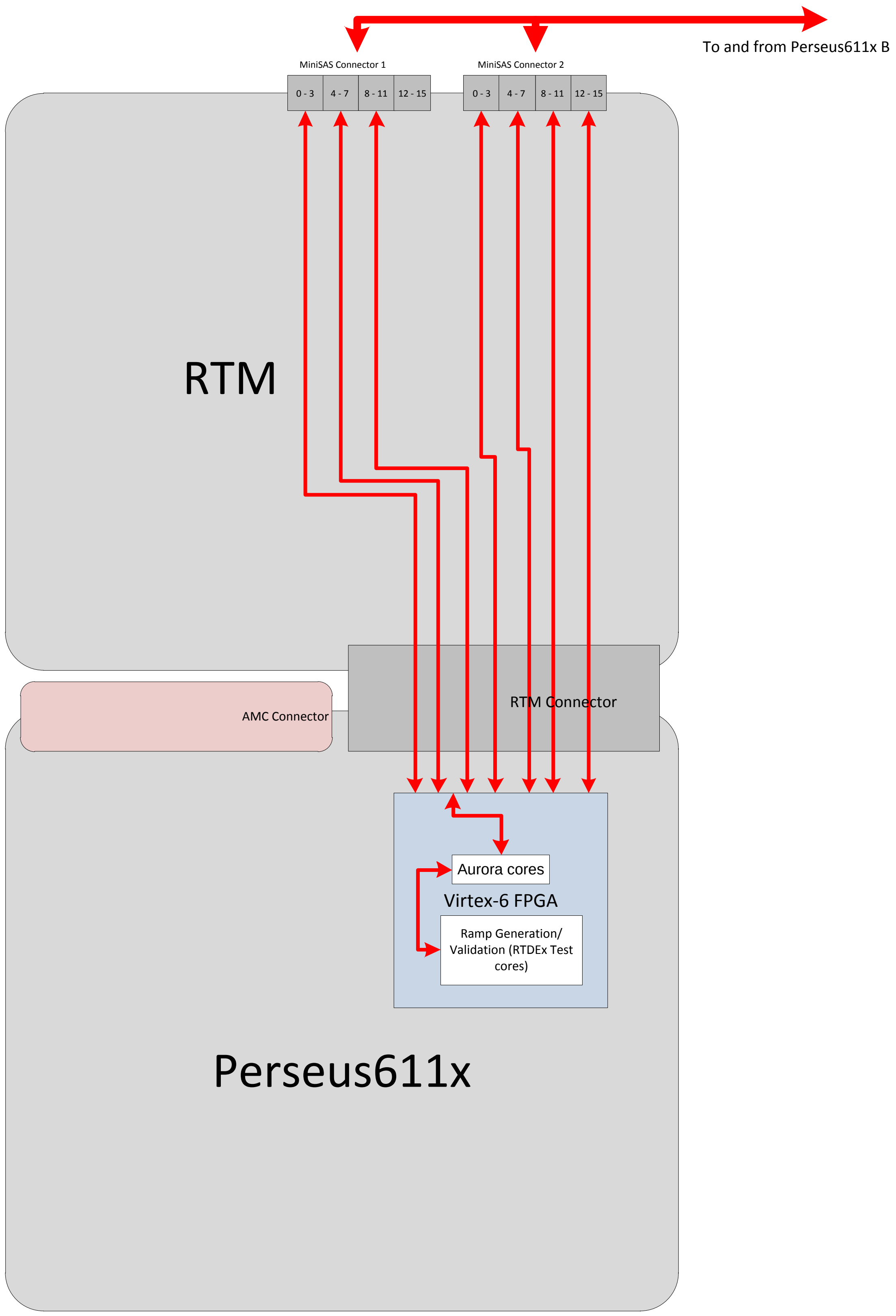


Figure 1-2 Perseus611x Aurora Demo schematic

1.2 Generating the Demos

BSDK versions of FPGA designs described in section 1.1 come pre-compiled in the software suite installer files. They are available in the */tools/bitstreams* folder.

MBDK versions of these FPGA designs do not come pre-compiled as they are functionally identical to their BSDK versions counterparts.

If only the MBDK version of an FPGA design is present in the software installer files, the MBDK precompiled bit file is located in the */tools/bitstreams* folder.

Sections 1.2.1 and 7 explain how to open the projects and regenerate them.

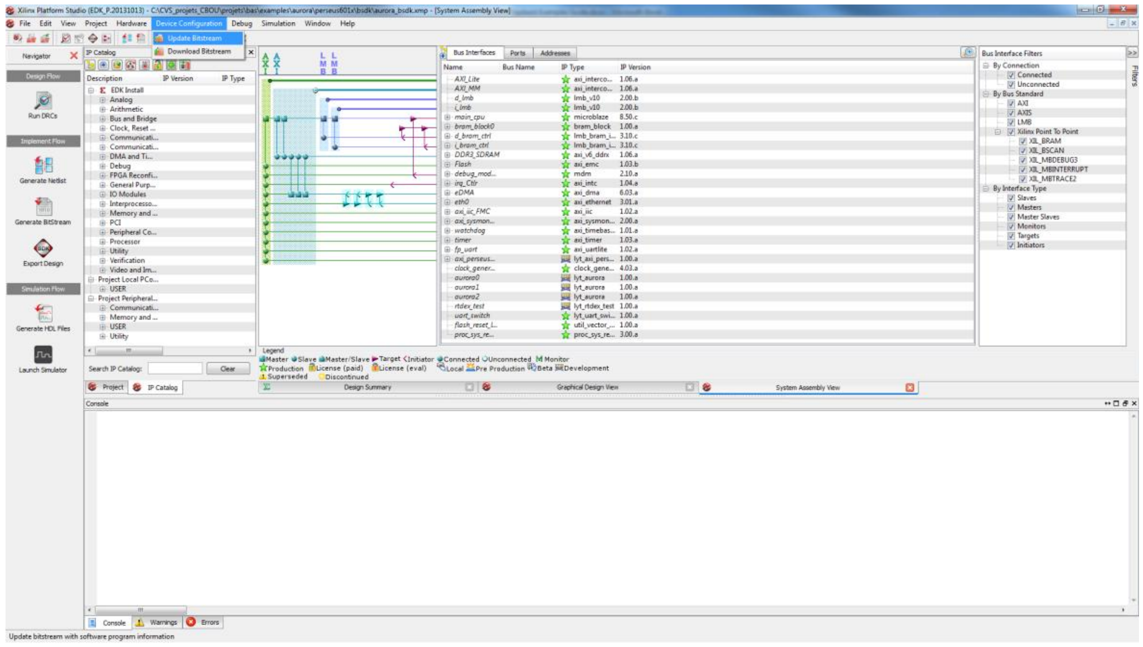
1.2.1 BSDK

1. Open the Xilinx Platform Studio 14.7 project file in the *\examples\aurora\<carrier>\bsdk* folder.

   Where *<carrier>* is either

   - *perseus601x* or
   - *perseus611x*
2. Explore the content the project.
3. On the **Project** menu, click **Project Options**
4. In the **Device Size** scrolling menu of the **General** tab, make sure the right device size, corresponding to the FPGA size installed on your carrier, is selected.
5. Click **OK**.
6. On the **Hardware** menu, click **Update Bitstream**

   Platform Studio will generate a bitstream (*.bit* file) and attach *perseus_default_linux.elf* from the *\sdk\embedded\bin* directory. This will allow the instantiated Microblaze soft processor to boot Petalinux from the onboard flash when the bitstream is downloaded to the FPGA.



Name	Bus Name	IP Type	IP Version
AXI_Lite		axi_interco...	1.06.a
AXI_MM		axi_interco...	1.06.a
d_lmb		lmb_v10	2.00.b
i_lmb		lmb_v10	2.00.b
main_cpu		microblaze	8.50.c
bram_block0		bram_block	1.00.a
d_bram_ctrl		lmb_bram_i...	3.10.c
i_bram_ctrl		lmb_bram_i...	3.10.c
DDR3_SDRAM		axi_v6_ddrx	1.06.a
Flash		axi_emc	1.03.b
debug_mod...		mdm	2.10.a
irq_Ctlr		axi_intc	1.04.a
eDMA		axi_dma	6.03.a
eth0		axi_ethernet	3.01.a
axi_iic_FMC		axi_iic	1.02.a
axi_sysmon...		axi_sysmon...	2.00.a
watchdog		axi_timebas...	1.01.a
timer		axi_timer	1.03.a
fp_uart		axi_uartlite	1.02.a
axi_perseus...		lyt_axi_pers...	1.00.a
clock_gener...		clock_gene...	4.03.a
aurora0		lyt_aurora	1.00.a
aurora1		lyt_aurora	1.00.a
aurora2		lyt_aurora	1.00.a
rtdex_test		lyt_rtdex_test	1.00.a
uart_switch		lyt_uart_swi...	1.00.a
flash_reset_l...		util_vector_...	1.00.a
proc_sys_re...		proc_sys_re...	3.00.a

7. This process usually takes between 30 minutes and up to a few hours depending on project complexity and computer performance. When completed, a *download.bit* becomes available in the *implementation* folder of the given project repository. Section 3 explains how to use this file and how to run the demo.

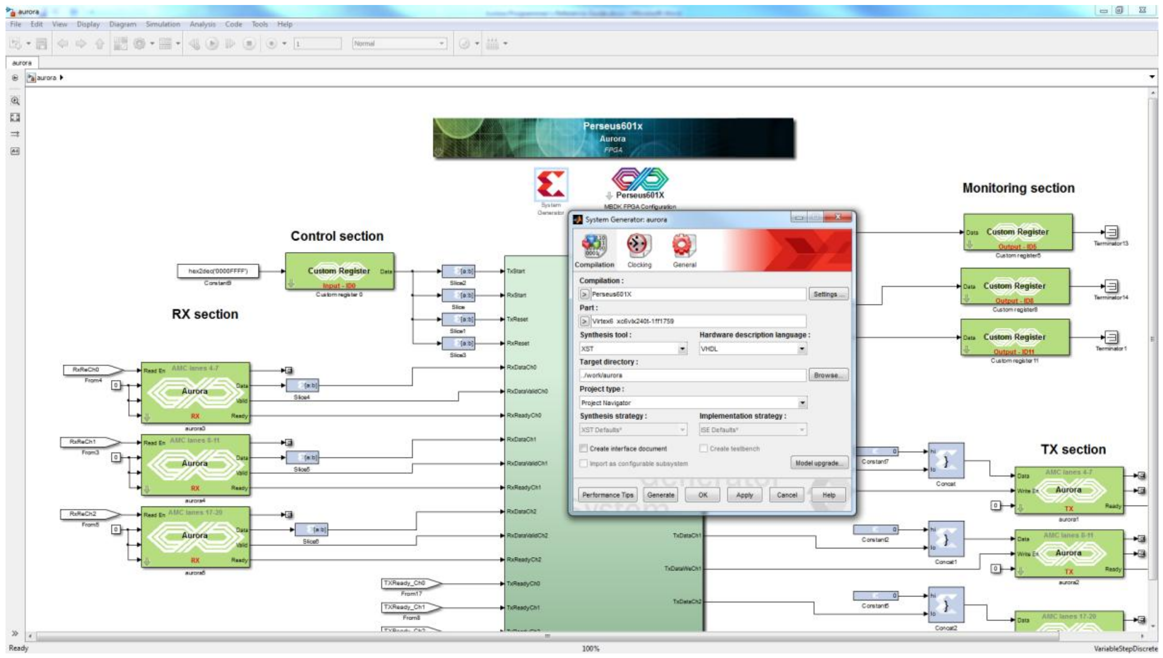
1.2.2 MBDK

If a Nutaq's Model Based Design Kit license is available with your software package,

1. Run Xilinx System Generator as Administrator.
2. Open the Xilinx System Generator 14.7 project file in the *\examples\aurora\<carrier>\mbdk* folder. The project file is a *.slx* file.

   Where *<carrier>* is either

   - *perseus601x* or
   - *perseus611x*
3. Explore the content of the project.
4. Double click on the **System Generator** symbol at the top left corner if the MBDK model.
5. In the **Part** section of the System Generator window, make sure the right device size, corresponding to the FPGA size installed on your carrier, is selected.
6. Click **Apply**.
7. Click **Generate**.



System Generator: aurora

Compilation : Perseus601X

Part : Virtex6 xc6vlx240t-1ff1759

Synthesis tool : XST

Hardware description language : VHDL

Target directory : ./work/aurora

Project type : Project Navigator

Synthesis strategy : XST Defaults*

Implementation strategy : ISE Defaults*

☐ Create interface document

☐ Create testbench

☐ Import as configurable subsystem

8. This process usually takes between 30 minutes and up to a few hours depending on project complexity and computer performance. When completed, a *aurora.bit* becomes available in the *implementation* folder of the given project repository. Section 3 explains how to use this file and how to run the demo.

2 Preparing to Execute the Demos

2.1 Requirements

The following hardware requirements must be met to perform the example.

2.1.1 Hardware for Perseus601X

- 2 Perseus601X
- 1 Nutaq PicoSDR/PicoDigitizer or a uTCA chassis with direct backplane connection on AMC4 to 11 and 17 to 20)
- 1 FPGA JTAG pod
- 1 USB cable (mini-USB to USB-A)

To setup the example (with the PicoSDR or PicoDigitizer chassis):

1. Connect the FPGA JTAG pod to the computer and then to the back of the PicoSDR/PicoDigitizer
2. Refer to the FPGA JTAG pod documentation for details about performing this operation.
3. Connect a mini-usb cable to the back of the PicoSDR/PicoDigitizer (for RS232 support)
4. Turn on the PicoSDR/PicoDigitizer.

2.1.2 Hardware for Perseus611X

AuroraTest Demo

- 2 Perseus611X
- 2 RTM boards
- MiniSAS cables to connect the RTM boards together.
- 1 Nutaq PicoSDR/PicoDigitizer or a uTCA chassis
- 1 FPGA JTAG pod
- 1 USB cable (mini-USB to USB-A) Setup

To setup the example (with the PicoSDR or PicoDigitizer chassis):

1. Connect the FPGA JTAG pod to the computer and then to the back of the PicoSDR/PicoDigitizer
2. Refer to the FPGA JTAG pod documentation for details about performing this operation.
3. Connect a mini-usb cable to the back of the PicoSDR/PicoDigitizer (for RS232 support)
4. Turn on the PicoSDR/PicoDigitizer.
5. Connect the RTM boards together with the MiniSAS cables.

AuroraTestLoopback Demo

- 1 Perseus611X
- 1 RTM boards
- MiniSAS cables to connect the RTM boards together.
- 1 Nutaq PicoSDR/PicoDigitizer or a uTCA chassis
- 1 FPGA JTAG pod
- 1 USB cable (mini-USB to USB-A) Setup

To setup the example (with the PicoSDR or PicoDigitizer chassis):

1. Connect the FPGA JTAG pod to the computer and then to the back of the PicoSDR/PicoDigitizer
2. Refer to the FPGA JTAG pod documentation for details about performing this operation.
3. Connect a mini-usb cable to the back of the PicoSDR/PicoDigitizer (for RS232 support)
4. Turn on the PicoSDR/PicoDigitizer.
5. Connect the RTM board to itself with the MiniSAS cables RTM Connector 1, port 12-15/8-11/4-7.

3 Executing the Demos

Two steps are needed in the execution of Aurora demos.

1. FPGAs of both carriers are configured with a pre-compiled *aurora* bitstream, or one generated following steps depicted in section 1.2.

   Section **Erreur ! Source du renvoi introuvable.** explains this operation.

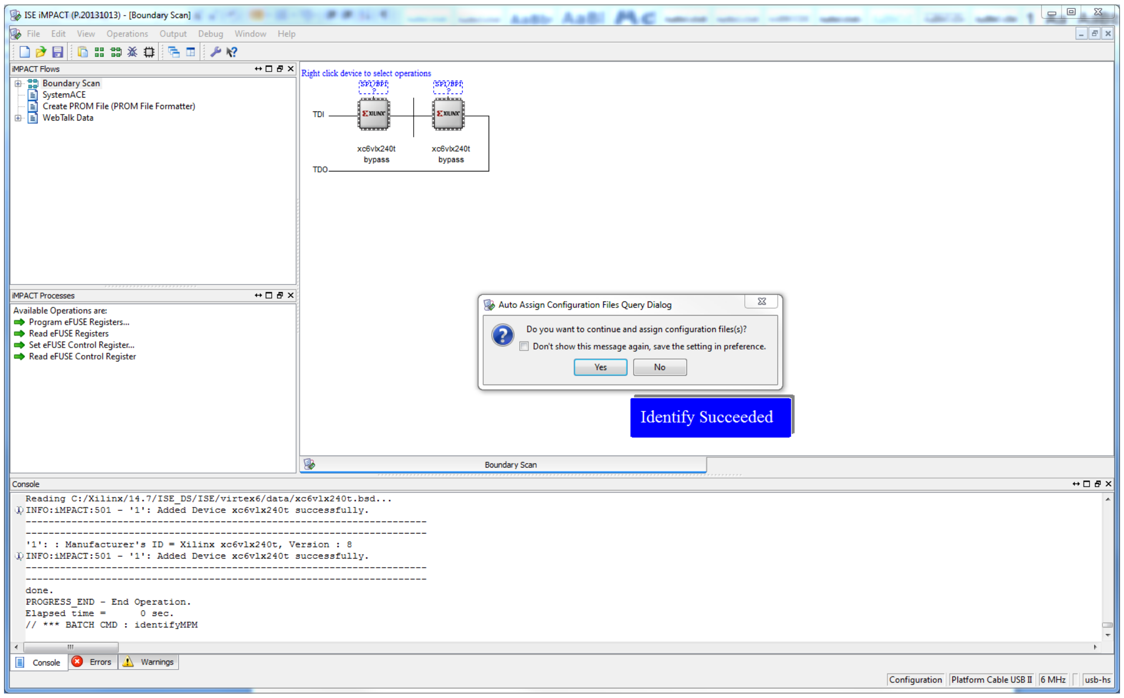
2. A host script is executed. It will configure and enable the various FPGA cores involved in the demo.

   Section **Erreur ! Source du renvoi introuvable.** explains this operation.

3.1 Configuring the FPGA

1. Make sure your JTAG pod is properly connected to your system as described in section **Erreur ! Source du renvoi introuvable.**.
2. Open Xilinx iMPACT.
3. The software make prompt you to create a project file. This is optional. Click **No**.
4. The software may then show a dialog called **New iMPACT Project**. Loading an already existing project or creating a new project file is again optional. Click **Cancel**.
5. In the left pane of iMPACT main window, in the **iMPACT Flows** pane, double click **Boundary Scan**.
6. In the toolbar, click the **Initialize Chain** button.

   iMPACT then shows devices detected in your JTAG chain. Figure 3-1 shows a JTAG chain with two FPGAs detected. Depending on your setup, iMPACT may detect only one device (if you perform this test using carriers not on the same JTAG chain) or more devices such as CPLDs and other FPGAs.


ISE iMPACT (P.20131013) - [Boundary Scan]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PROM File Formatter)
WebTalk Data
iMPACT Processes
Available Operations are:
Program eFUSE Registers...
Read eFUSE Registers
Set eFUSE Control Register...
Read eFUSE Control Register
Right click device to select operations
TDI
TDO
xc6vlx240t
bypass
xc6vlx240t
bypass
Auto Assign Configuration Files Query Dialog
Do you want to continue and assign configuration files(s)?
Don't show this message again, save the setting in preference.
Yes
No
Identify Succeeded
Boundary Scan
Console
Errors
Warnings
Configuration
Platform Cable USB II
6 MHz
usb-hs

```
Reading C:/Xilinx/14.7/ISE_DS/ISE/virtex6/data/xc6vlx240t.bsd...
INFO:iMPACT:501 - '1': Added Device xc6vlx240t successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
'1': : Manufacturer's ID = Xilinx xc6vlx240t, Version : 8
INFO:iMPACT:501 - '1': Added Device xc6vlx240t successfully.
----------------------------------------------------------------------
----------------------------------------------------------------------
done.
PROGRESS_END - End Operation.
Elapsed time =       0 sec.
// *** BATCH CMD : identifyMPM
```

Figure 3-1. After iMPACT Boundary Scan

7. If Auto Assign Configure Files Query Dialog appears, click **No**.
8. A Device Programming Properties dialog may appear, click **Cancel**.
9. In the JTAG chain, right click on the device to program, that is, the Virtex 6 device installed on the Perseus carrier.
10. Click **Assign New Configuration File…**.
11. Browse to the bitstream to download to the device and click **Open**. Section 1.2 explains how to generate this bitstream and where to find pre-compiled bitstreams shipped with your software tools.

    WARNING

    Be very careful in selecting the bitstream to program to your FPGA device.

    - Users operating a system equipped with a Perseus601x must select their bitstream from the *\examples\aurora\perseus601x* directory.
    - Users operating a system equipped with a Perseus611x must select their bitstream from the *\examples\aurora\perseus611x* directory.

    Downloading a bitstream designed for a carrier different from the carrier you are using may permanently damage the system.
12. A dialog prompting you to attach an SPI or BPI PROM to the device may appear. Click **No**.
13. In the JTAG chain, right click again on the device to program, and click **Program**.
14. If a Device Programming Properties dialog appears, click **OK**.
15. The bitstream is programmed to the FPGA.
16. Repeat steps 9 to 14 to program the other FPGA that is present on the same JTAG chain.

    If you perform this test on carriers not on the same JTAG chain, connect your JTAG pod to the other system holding the device left to program and repeat steps 6 to 14.

3.2 Host scripts

Scripts showcasing raw Aurora transfers between carriers are available and they can be run using the bitstream configured in section **Erreur ! Source du renvoi introuvable.**. They are located in the */examples/rtdex_recplay/host/scripts* directory and take the form of a batch file (.*bat*) or a shell script (.*sh*) for Windows or Linux, respectively.

3.2.1 AuroraTest

1. Browse to */examples/aurora/host/scripts*.
2. Using a text editor, open/display
   - *AuroraTest.bat* if you use Windows
   - *AuroraTest.sh* if you use Linux
3. Change the value of variables *CARRIER1IPADDRESS* and *CARRIER2IPADDRESS* to match your system configuration. This is the IP addresses of the carriers running the demo.
4. Save the script file.
5. To execute the demo right away, skip to section 3.2.1.1.

The script uses applications of which source code is made available in the *tools**apps* repository. Details about those applications are given in the Programmer's Reference Guide of their respective associated FPGA core.

Application	Associated FPGA core	Path to Programmer's Reference Guide
RTDExTestUtil	RTDEx Test	\doc\cores\RTDExTest
AuroraInitChannel	Aurora	\doc\cores\Aurora
AuroraMonitor	Aurora	\doc\cores\Aurora

The script:

- Initializes the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*, the first time for carrier 1, then for carrier 2.

  Notable parameters of file *RtdexTestRx.ini*:

  - *adc_dac_mode* is disabled. The RTDEx Test core will generate data as fast as it can but will be limited by the Aurora core capacity. Aurora pipes throughput will be maxed out with no overflow.
  - *rtdex_test_base_custom_register* is to 0. This parameter specifies to *RtdexTestUtil* the first custom register in the FPGA design that is used for the RTDEx Test core configuration.
  - number_of_channels_to_analyze is set to 3. When the *RtdexTestUtil* application is called in "stop" mode, this tells the *RtdexTestUtil* application to gather statistics from 3 RTDEx Test core channels.

- Initializes all three Aurora channels, in both directions (RXTX) using application *AuroraInitChannel*, the first time for carrier 1, then for carrier 2.

At this point, Aurora channels are initialized but no data is being transmitted yet.

- Enables the RTDEx Test core in Rx mode using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*, the first time for carrier 1, then for carrier 2.
- Enables the RTDEx Test core in Tx mode using application *RtdexTestUtil* with configuration file *RtdexTestTx.ini*, the first time for carrier, then for carrier 2. These two application calls enable data generation by RTDEx Test cores on both carriers and so enable bidirectional transmission between the two systems.
- Uses utility script *new_console_with_pause* to execute application *AuroraMonitor* in another terminal window, in parallel. The application is called for carrier 1, then for carrier 2. This allows monitoring of Aurora channels in independent terminal instances.
- Stops the test and gathers statistics using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*, the first time for carrier 1, then for carrier 2. For this test, statistics are only relevant in downlink direction, as seen from the RTDEx Test core. We then use the rx version of the configuration when calling *RtdexTestUtil*. However, the “stop” mode of this application actually disables the whole core, thus disabling data transmission in both directions.

3.2.1.1 Execution

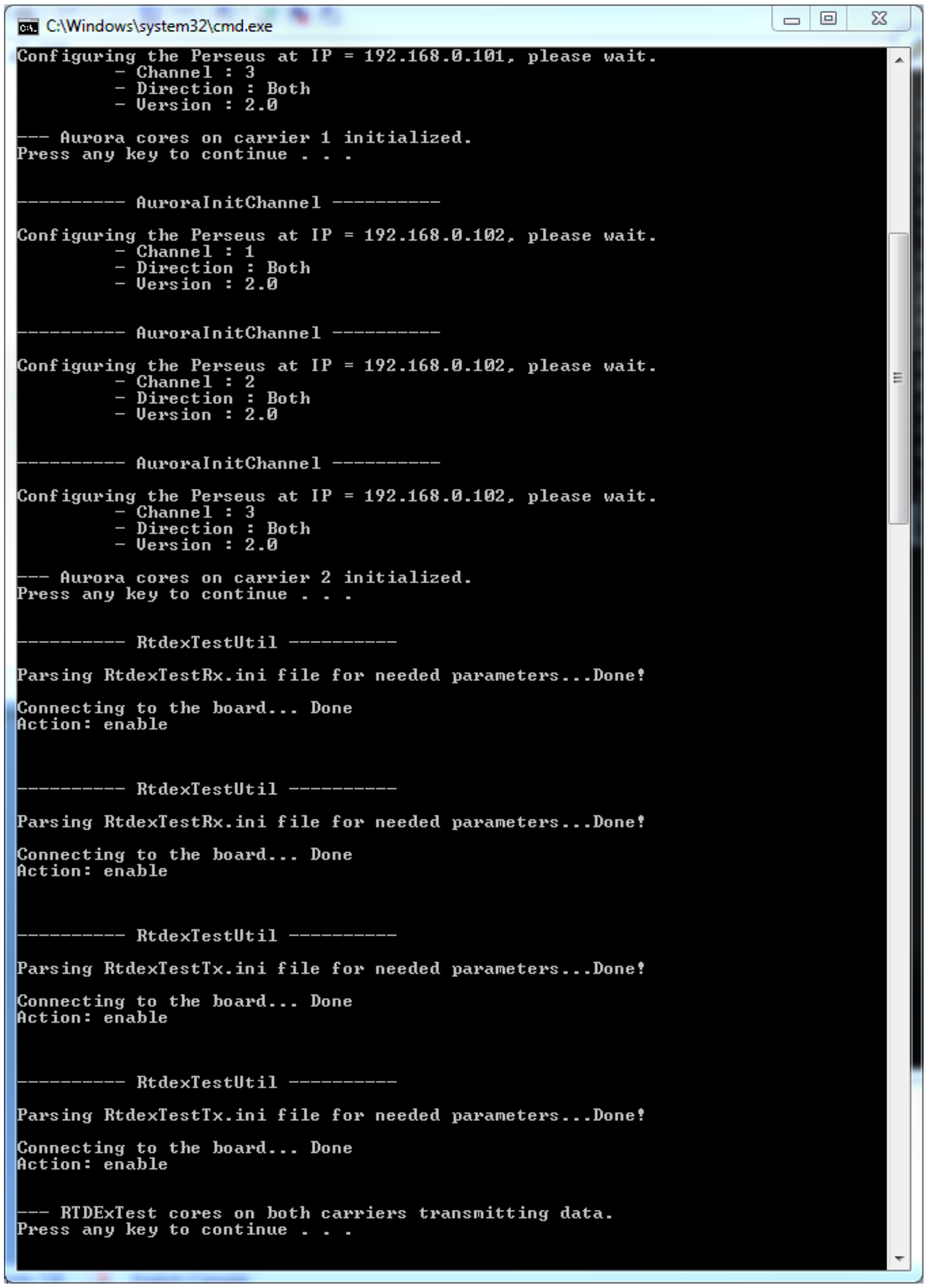
Execution simply consists of launching the script described in the previous section. Instructions differ whether Windows or Linux is used.

3.2.1.1.1 Windows

Double-click the *AuroraTest.bat* file.

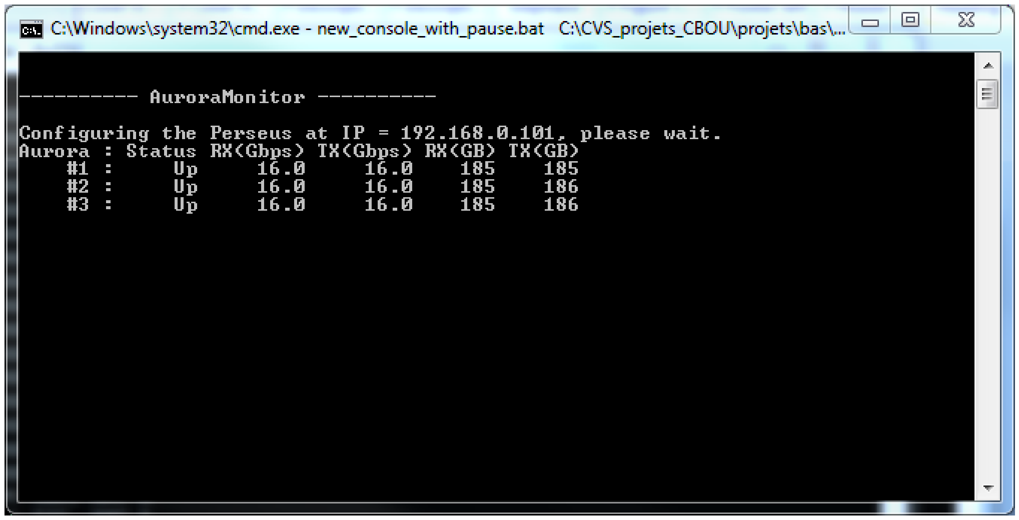
The test starts in a terminal window and is designed as a step by step demo. Following some steps, the script prompts the user to hit any key in order to continue.

Expected display while data transmission is ongoing:


C:\Windows\system32\cmd.exe

```
Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 3
         - Direction : Both
         - Version : 2.0

--- Aurora cores on carrier 1 initialized.
Press any key to continue . . .

---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 1
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 2
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 3
         - Direction : Both
         - Version : 2.0

--- Aurora cores on carrier 2 initialized.
Press any key to continue . . .

---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable

--- RTDExTest cores on both carriers transmitting data.
Press any key to continue . . .
```

Expected display of one of the two Aurora monitoring windows:

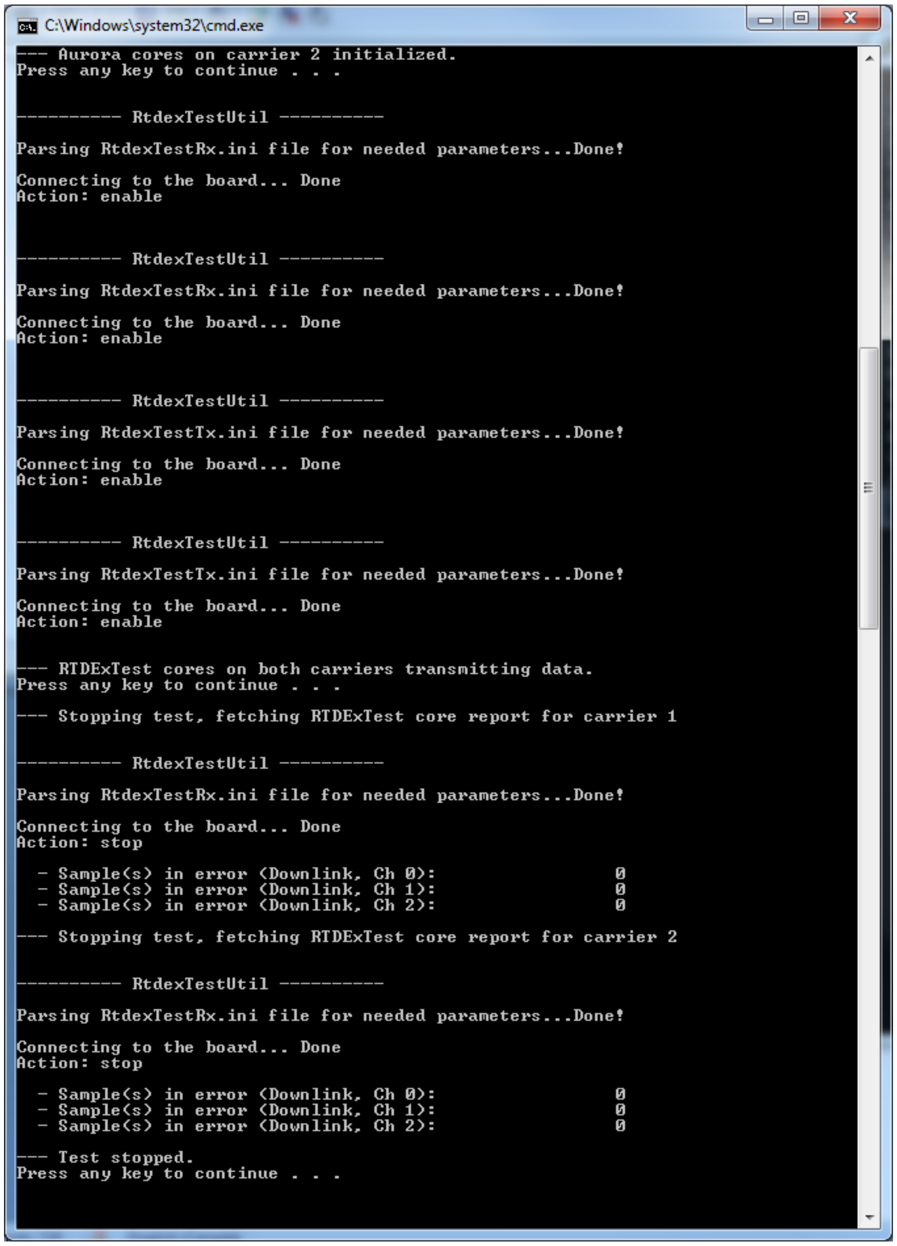


C:\Windows\system32\cmd.exe - new_console_with_pause.bat   C:\CVS_projets_CBOU\projets\bas\...

```
---------- AuroraMonitor ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
Aurora : Status RX(Gbps) TX(Gbps) RX(GB) TX(GB)
    #1 :     Up     16.0     16.0    185    185
    #2 :     Up     16.0     16.0    185    186
    #3 :     Up     16.0     16.0    185    186
```

Speeds should keep stable at around 16 Gbps while RX and TX data counts should be rising continuously.

Expected results once the test has been terminated by user:



```
--- Aurora cores on carrier 2 initialized.
Press any key to continue . . .

---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable

--- RTDExTest cores on both carriers transmitting data.
Press any key to continue . . .

--- Stopping test, fetching RTDExTest core report for carrier 1

---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Downlink, Ch 0):                 0
  - Sample(s) in error (Downlink, Ch 1):                 0
  - Sample(s) in error (Downlink, Ch 2):                 0

--- Stopping test, fetching RTDExTest core report for carrier 2

---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Downlink, Ch 0):                 0
  - Sample(s) in error (Downlink, Ch 1):                 0
  - Sample(s) in error (Downlink, Ch 2):                 0

--- Test stopped.
Press any key to continue . . .
```

3.2.1.1.2 Linux

1. In a Linux terminal, change directory to *bas/examples/aurora/host/scripts*.
2. To start the example, run the following command

   sudo ./AuroraTest.sh

Expected results are the same as on Windows.

3.2.2 AuroraTestLoopback

1. Browse to */examples/aurora/host/scripts*.

2. Using a text editor, open/display
   - *AuroraTestLoopback.bat* if you use Windows
   - *AuroraTestLoopback.sh* if you use Linux
3. Change the value of variable *CARRIER1IPADDRESS* to match your system configuration. This is the IP address of the carrier running the demo.
4. Save the script file.
5. To execute the demo right away, skip to section 3.2.1.1.

The script uses applications of which source code is made available in the *\tools\apps* repository. Details about those applications are given in the Programmer's Reference Guide of their respective associated FPGA core.

Application	Associated FPGA core	Path to Programmer's Reference Guide
RTDExTestUtil	RTDEx Test	\doc\cores\RTDExTest
AuroraInitChannel	Aurora	\doc\cores\Aurora
AuroraMonitor	Aurora	\doc\cores\Aurora

The script:

- Initializes the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*, the first time for carrier 1, then for carrier 2.

  Notable parameters of file *RtdexTestRx.ini*:

  - *adc_dac_mode* is disabled. The RTDEx Test core will generate data as fast as it can but will be limited by the Aurora core capacity. Aurora pipes throughput will be maxed out with no overflow.
  - *rtdex_test_base_custom_register* is to 0. This parameter specifies to *RtdexTestUtil* the first custom register in the FPGA design that is used for the RTDEx Test core configuration.
  - number_of_channels_to_analyze is set to 3. When the *RtdexTestUtil* application is called in "stop" mode, this tells the *RtdexTestUtil* application to gather statistics from 3 RTDEx Test core channels.
- Initializes all seven Aurora channels, in both directions (RXTX) using application *AuroraInitChannel*.

  At this point, Aurora channels are initialized but no data is being transmitted yet.
- Enables the RTDEx Test core in Rx mode using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*.
- Enables the RTDEx Test core in Tx mode using application *RtdexTestUtil* with configuration file *RtdexTestTx.ini*. These two application calls enable data generation by RTDEx Test core and so enable bidirectional transmission.
- Uses utility script *new_console_with_pause* to execute application *AuroraMonitor* in another terminal window, in parallel. This allows monitoring of Aurora channels in independent terminal instances.

- Stops the test and gathers statistics using application *RtdexTestUtil* with configuration file *RtdexTestRx.ini*. For this test, statistics are only relevant in downlink direction, as seen from the RTDEx Test core. We then use the rx version of the configuration when calling *RtdexTestUtil*. However, the “stop” mode of this application actually disables the whole core, thus disabling data transmission in both directions.

3.2.2.1 Execution

Execution simply consists of launching the script described in the previous section. Instructions differ whether Windows or Linux is used.

3.2.2.1.1 Windows

Double-click the *AuroraTestLoopback.bat* file.

The test starts in a terminal window and is designed as a step by step demo. Following some steps, the script prompts the user to hit any key in order to continue.

Expected display while data transmission is ongoing:

```
---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: init


--- RTDExTest core on FPGA carrier initialized.
Press any key to continue . . .


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 1
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 2
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 3
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 4
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 5
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 6
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 7
         - Direction : Both
         - Version : 2.0

--- Aurora cores on FPGA carrier initialized.
Press any key to continue . . .
```

```
---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable



---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


--- RTDExTest cores on FPGA carrier transmitting data.

--- Press enter to end test and display test results.
Press any key to continue . . .
```

Expected display of one of the two Aurora monitoring windows:

```
---------- AuroraMonitor ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
Aurora : Status RX(Gbps) TX(Gbps) RX(GB) TX(GB)
    #1 :   Down      0.0      0.0      0      0
    #2 :   Down      0.0      0.0      0      0
    #3 :   Down      0.0      0.0      0      0
    #4 :   Down      0.0      0.0      0      0
    #5 :     Up     16.0     16.0      8      8
    #6 :   Down     16.0     16.0      0      0
    #7 :     Up      0.0      0.0      8      8
```

Speeds should keep stable at around 16 Gbps while RX and TX data counts should be rising continuously.

Expected results once the test has been terminated by user:

```
--- Stopping test, in uplink direction


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop


--- Stopping test, in downlink direction


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Downlink, Ch 0):                        0
  - Sample(s) received by RtdexTest Core (Downlink, Ch 0):      0
  - Sample(s) in error (Downlink, Ch 1):                        0
  - Sample(s) received by RtdexTest Core (Downlink, Ch 1):      0
  - Sample(s) in error (Downlink, Ch 2):                        0
  - Sample(s) received by RtdexTest Core (Downlink, Ch 2):      0

--- Test stopped.
Press any key to continue . . .
```

3.2.2.1.2 Linux

3. In a Linux terminal, change directory to *bas/examples/aurora/host/scripts*.
4. To start the example, run the following command

   sudo ./AuroraTestLoopback.sh

Expected results are the same as on Windows.

3.2.3 Modifying default example when using more than 3 Aurora cores

If you are on a carrier board that has more than three Aurora cores, it possible to modify the script and initialization file to configure all the Aurora cores. The Perseus611X example has 7 Aurora core instantiated on its MiniSAS RTM connectors.

1. Open the "AuroraTest" script with a text editor.
2. Copy the following lines and modify the channel index (1 in this case) to the desired new channel index:

```
%APP_ROOT%\AuroraInitChannel.exe %CARRIER1IPADDRESS% 1 RXTX
IF %ERRORLEVEL% NEQ 0 GOTO End
```

3. The 7 Aurora channels of the first carrier board and the 7 channels of the second carrier board should be initialized.
4. Save the modified script.
5. Open the "RtdexTestRx.ini" and "RtdexTestTx.ini" files with a text editor.

6. For both files, modify the “number_of_channels_to_analyze” parameters from 3 to 7.
7. Save the modified initialization files.

3.2.3.1 Execution

Execution simply consists of launching the script described in the previous section. Instructions differ whether Windows or Linux is used.

3.2.3.1.1 Windows

Double-click the *AuroraTest.bat* file.

The test starts in a terminal window and is designed as a step by step demo. Following some steps, the script prompts the user to hit any key in order to continue.

Expected display while data transmission is ongoing:

```
---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: init



---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: init


--- RTDExTest cores on both carriers initialized.
Press any key to continue . . .


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 1
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 2
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 3
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------
```

```
Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 4
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 5
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 6
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
         - Channel : 7
         - Direction : Both
         - Version : 2.0

--- Aurora cores on carrier 1 initialized.
Press any key to continue . . .


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 1
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 2
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 3
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 4
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 5
         - Direction : Both
         - Version : 2.0
```

```
---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 6
         - Direction : Both
         - Version : 2.0


---------- AuroraInitChannel ----------

Configuring the Perseus at IP = 192.168.0.102, please wait.
         - Channel : 7
         - Direction : Both
         - Version : 2.0

--- Aurora cores on carrier 2 initialized.
Press any key to continue . . .


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable



---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable



---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable



---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: enable


--- RTDExTest cores on both carriers transmitting data.
Press any key to continue . . .
```

Expected display of one of the two Aurora monitoring windows:

```
---------- AuroraMonitor ----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
Aurora : Status RX(Gbps) TX(Gbps) RX(GB) TX(GB)
    #1 :     Up     16.0     16.0    173    173
    #2 :     Up     16.0     16.0    173    173
    #3 :     Up     16.0     16.0    173    173
    #4 :     Up     16.0     16.0    173    173
    #5 :     Up     16.0     16.0    173    173
    #6 :      Up      16.0      16.0     173     173
    #7 :    Down       0.0       0.0       0       0
```

Speeds should keep stable at around 16 Gbps while RX and TX data counts should be rising continuously.

In the above example, the first six MiniSAS connectors are connected the other RTM board while the seventh connector is left unconnected.

Expected results once the test has been terminated by user:

```
---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop


---------- RtdexTestUtil ----------

Parsing RtdexTestTx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Downlink, Ch 0):                 0
  - Sample(s) in error (Downlink, Ch 1):                 0
  - Sample(s) in error (Downlink, Ch 2):                 0
  - Sample(s) in error (Downlink, Ch 3):                 0
  - Sample(s) in error (Downlink, Ch 4):                 0
  - Sample(s) in error (Downlink, Ch 5):                 0
  - Sample(s) in error (Downlink, Ch 6):                 0


---------- RtdexTestUtil ----------

Parsing RtdexTestRx.ini file for needed parameters...Done!

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Downlink, Ch 0):                 0
  - Sample(s) in error (Downlink, Ch 1):                 0
  - Sample(s) in error (Downlink, Ch 2):                 0
  - Sample(s) in error (Downlink, Ch 3):                 0
  - Sample(s) in error (Downlink, Ch 4):                 0
  - Sample(s) in error (Downlink, Ch 5):                 0
  - Sample(s) in error (Downlink, Ch 6):                 0

--- Test stopped.
Press any key to continue . . .
```

3.2.3.1.2 Linux

1. In a Linux terminal, change directory to *bas/examples/aurora/host/scripts*.
2. To start the example, run the following command

```
sudo ./AuroraTest.sh
```

Expected results are the same as on Windows.