Nutaq® INNOVATION TODAY FOR TOMORROW®

RTDEx and Record/Playback

Perseus Examples

January 2017

Revision history

Revision	Date	Comments
1.0	December 2012	First released revision.
1.1	February 2013	Ready for linguistic revision.
1.2	March 2013	Linguistic revision.
1.3	March 2013	Added PCIe example.
1.4	April 2013	Simplified example.
1.5	September 2013	Merged Linux and Windows examples.
1.6	June 2014	Modified Record Playback Example. Added PicoSDR/Digitizer setup section.
1.7	November 2014	Added sudo prefix to Linux shell calls. Up to date for Software Tools Release 6.6.
1.8	October 2015	Restructured document and adjusted content for BAS 7.0.
1.9	February 2016	Up to date for PicoSDR official release.
2.0	January 2017	Up to date for PicoSDR 2nd Generation 1.1 official release. Modified expected results to match example application verbose.

Table of Contents

List of Figures

1 FPGA Designs Description and Generation

1.1 RTDEx and Record/Playback Demos Description

A set of FPGA designs showcasing how to use the Record/Playback and RTDEx cores and how they interface with user logic are made available in your BAS software suite. They reside in the *%BASROOT%\examples\rtdex_recplay* directory. These FPGA designs are available for both Perseus601x and Perseus611x, can be used with both PCIe and Gigabit Ethernet depending on the selected FPGA project and are designed using either Nutaq's Model Based Design Kit or Xilinx Platform Studio directly (BSDK).

Project files path	Design methodology	RTDEx media	Carrier
perseus601x\bsdk_gige	Xilinx Platform Studio (BSDK)	Gigabit Ethernet	Perseus601x
perseus601x\bsdk_pcie	Xilinx Platform Studio (BSDK)	PCI-Express	Perseus601x
perseus601x\mbdk_gige	Nutaq's Model Based Design Kit (MBDK)	Gigabit Ethernet	Perseus601x
perseus601x\mbdk_pcie	Nutaq's Model Based Design Kit (MBDK)	PCI-Express	Perseus601x
perseus611x\bsdk_gige	Xilinx Platform Studio (BSDK)	Gigabit Ethernet	Perseus611x
perseus611x\bsdk_pcie	Xilinx Platform Studio (BSDK)	PCI-Express	Perseus611x
perseus611x\mbdk_gige	Nutaq's Model Based Design Kit (MBDK)	Gigabit Ethernet	Perseus611x
perseus611x\mbdk_pcie	Nutaq's Model Based Design Kit (MBDK)	PCI-Express	Perseus611x

Table 1 Available FPGA designs depiction

These eight designs are functionally identical, and they can be illustrated with Figure 1-1. This schematic exposes the principal cores involved in this demo.

The RTDEx core is the utility providing the data pipes linking the FPGA and the host. An instance of the RTDEx Test core is connected to RTDEx channels 1 and 2 in order to test raw data transfers using RTDEx.

The Record/Playback core is connected to RTDEx channel 0 and is responsible for

- loading data in memory from host using RTDEx, then playbacking it to user logic;
- recording data from user logic, then transferring it to host using RTDEx.

In this FPGA design, user logic solely consists of an instance of the Record/Playback Test core and an instance of the RTDEx test core.

As no FMC is operated in this demo, data is synthesized and consumed internally, by the RTDEx Test and Record/Playback Test cores, depending on selected test mode. These cores generate and validate a simple binary ramp, that is ultimately transferred to and received from the host PC, respectively.

As in all provided Nutaq's FPGA designs, a Microblaze core is instantiated and accomplishes AXI exchanges with other cores.

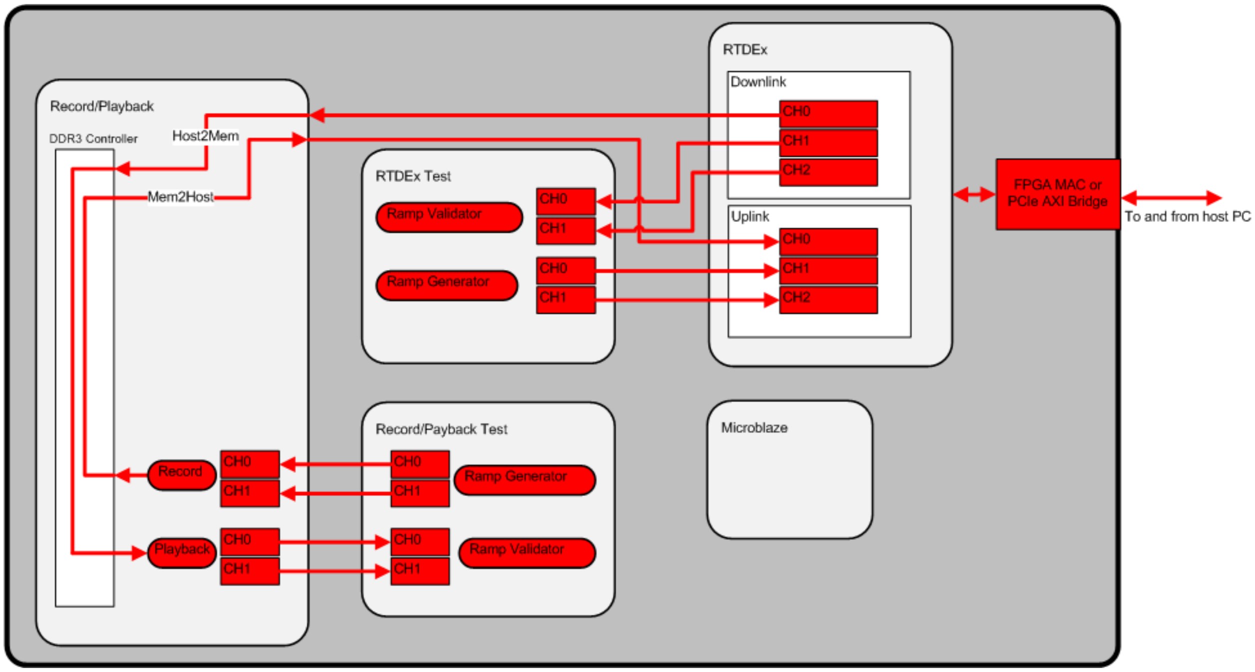


Figure 1-1 FPGA design schematic

1.2 Generating the Demos

BSDK versions of FPGA designs described in section 1.1 come pre-compiled in your BAS software suite. They are available in the *%BASROOT%\tools\bitstreams* folder.

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

Sections 1.2.1 and 1.2.2 explain how to open the projects and regenerate them.

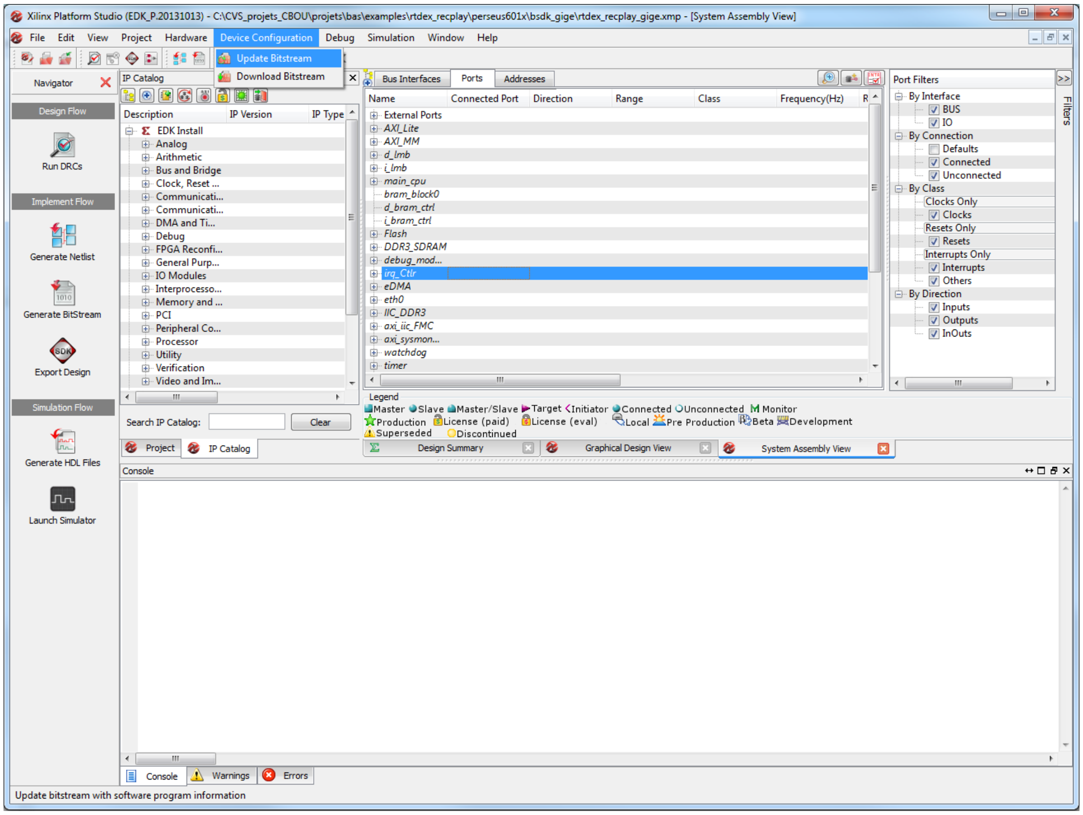
1.2.1 BSDK

1. Open the Xilinx Platform Studio 14.7 project file in the *%BASROOT%\examples\rtdex_recplay\<carrier>\bsdk_<media>* folder.

   Where

   - *<carrier>* is either
     - *perseus601x* or
     - *perseus611x*
   - *<media>* is either
     - *gige* or
     - *pcie*
2. Explore the content of 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 *%BASROOT%\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.



Xilinx Platform Studio (EDK_P.20131013) - C:\CVS_projets_CBOU\projets\bas\examples\rtdex_recplay\perseus601x\bsdk_gige\rtdex_recplay_gige.xmp - [System Assembly View]

Update bitstream with software program information

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

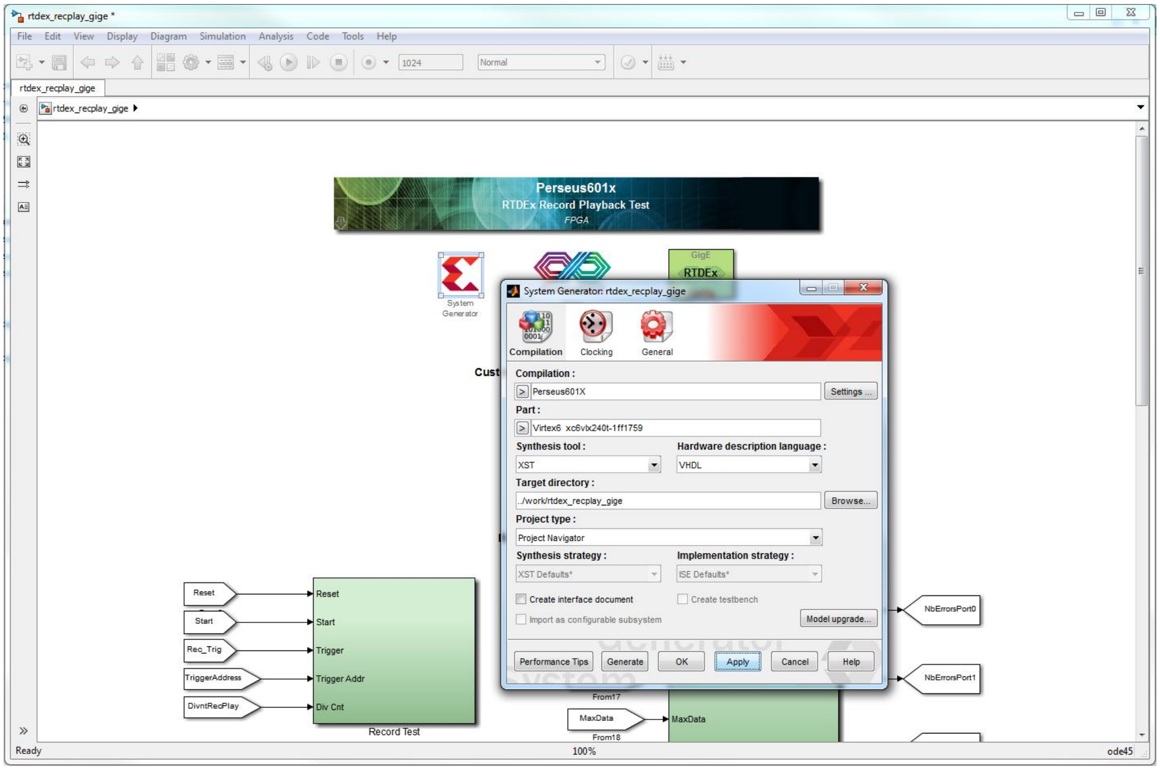
If a Nutaq's Model Based Design Kit license is available with your software package,

1. Run Xilinx System Generator as Administrator.
2. Open the Xilinx System Generator 14.7 project file in the *%BASROOT%\examples\rtdex_recplay\<carrier>\mbdk_<media>* folder. The project file is a *.slx* file.

   Where

   - *<carrier>* is either
     - ○ *perseus601x* or
     - ○ *perseus611x*
   - *<media>* is either
     - ○ *gige* or
     - ○ *pcie*
3. Explore the content of the project.
4. Double click on the **System Generator** symbol at the top left corner of 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: rtdex_recplay_gige

Compilation :
Perseus601X

Part :
Virtex6 xc6vlx240t-1ff1759

Synthesis tool : XST

Hardware description language : VHDL

Target directory :
../work/rtdex_recplay_gige

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 at least 30 minutes and up to a few hours depending on project complexity and computer performance. When completed, a *rtdex_gige_recplay.bit* file becomes available directly in 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 Hardware Requirements when using GigE

These are the hardware requirements to run the examples depending on the setup used.

PicoSDR/Digitizer with external PC

- 1 PicoSDR / PicoDigitizer
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 PC with a Gigabit Ethernet network card (jumbo frame capable)

Setup to run the example

PicoSDR/Digitizer with embedded CPU

- 1 PicoSDR / PicoDigitizer
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 computer Xilinx Impact (to operate the FPGA JTAG pod)

Setup to run the example

Perseus in a µTCA chassis with External PC

- 1 Perseus
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 TCA chassis with its TCA Carrier Hub (MCH)
- 1 computer with a Gigabit Ethernet network card (jumbo frame capable)

Setup to run the example

Perseus and embedded PC in a µTCA chassis

- 1 Perseus
- 1 embedded PC running Linux Fedora or Ubuntu
- 1 FPGA JTAG pod
- 1 TCA chassis with its TCA Carrier Hub (MCH)
- 1 computer with Xilinx Impact (to operate the FPGA JTAG pod)

Setup to run the example

2.2 Hardware Setup when using GigE

2.2.1 Setup for execution on a PicoSDR or PicoDigitizer with an external PC

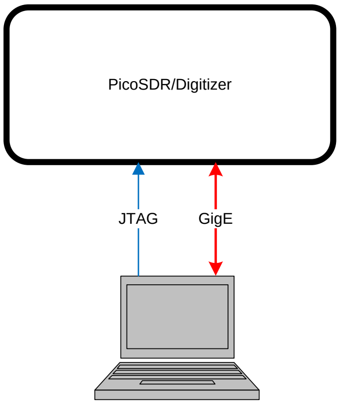


Figure 2-1 PicoSDR/Digitizer with an external PC

The following procedure must be performed to execute the examples on a PicoSDR or PicoDigitizer system with an external PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then the PicoSDR/Digitizer back panel.

   Refer to the PicoSDR User's Guide or PicoDigitizer User's Guide documents for details about performing this operation.

2. Connect the Gigabit Ethernet cable between the PicoSDR/Digitizer back panel Ethernet connector and your PC's Gigabit Ethernet network card.
3. Turn on the PicoSDR/Digitizer.
4. Make sure that the PicoSDR/Digitizer Perseus has a valid MAC and IP addresses.

2.2.2 Setup for execution on a PicoSDR or PicoDigitizer on its embedded PC

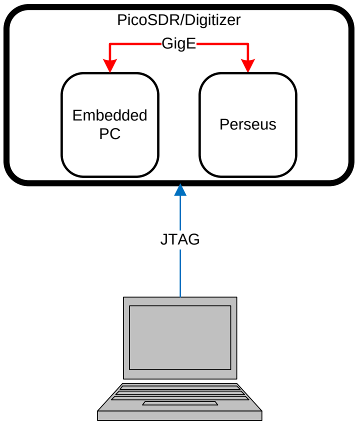


Figure 2-2 PicoSDR/Digitizer with an its embedded PC

The PicoSDR 2x2E, as well as the PicoDigitizer125-16E, 125-32E and 250E contain an embedded PC and a Perseus601x. The following procedure must be performed to execute the examples on a PicoSDR or PicoDigitizer system with its embedded PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then the PicoSDR/Digitizer back panel.

   Refer to the PicoSDR User's Guide or PicoDigitizer User's Guide documents for details about performing this operation.
2. Turn on the PicoSDR/Digitizer.
3. Make sure that the PicoSDR/Digitizer Perseus has a valid MAC and IP addresses.

2.2.3 Setup for execution on a Perseus601x in a MicroTCA chassis, with an external PC

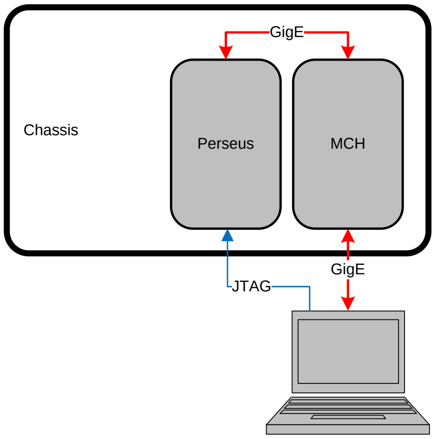


Figure 2-3 Perseus in a MicroTCA chassis, with an external PC

The following procedure must be performed to execute the examples in a chassis system with an external PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then to the Perseus.

   Refer to the Perseus User's Guide document for details about performing this operation.
2. Insert the Perseus into the µTCA chassis.
3. Connect the Gigabit Ethernet cable between the MCH's front panel Ethernet connector and your PC's Gigabit Ethernet network card.
4. Turn on the µTCA chassis.
5. Make sure that the Perseus has a valid MAC and IP addresses.

2.2.4 Setup for execution on a Perseus601x in a MicroTCA chassis, with an embedded PC

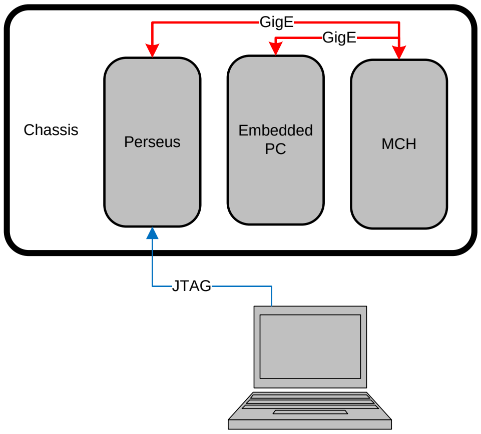


Figure 2-4 Perseus in a MicroTCA chassis, with an embedded PC

The following procedure must be performed to execute the examples in a chassis system with an embedded PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then to the Perseus.

   Refer to the Perseus User's Guide document for details about performing this operation.

2. Insert the Perseus into the µTCA chassis.
3. Turn on the µTCA chassis.
4. Make sure that the Perseus has a valid MAC and IP addresses.

2.3 Hardware Requirements when using PCIe

These are the hardware requirements to run the examples depending on the setup used.

PicoSDR/Digitizer with external PC

- 1 PicoSDR / PicoDigitizer
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 PC with PCI express expansion cable

Setup to run the example

PicoSDR/Digitizer with embedded CPU

- 1 PicoSDR / PicoDigitizer
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 computer (to operate the FPGA JTAG pod)

Setup to run the example

Perseus and embedded PC in a µTCA chassis

- 1 Perseus
- 1 embedded PC running Linux Fedora or Ubuntu
- 1 FPGA JTAG pod
- 1 TCA chassis with its TCA Carrier Hub (MCH)
- 1 computer (to operate the FPGA JTAG pod)

Setup to run the example

2.4 Hardware Setup when using PCIe

2.4.1 Setup for execution on a PicoSDR or PicoDigitizer with an external PC

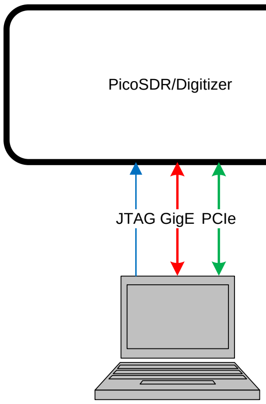


Figure 2-5 PicoSDR/Digitizer with an external PC

The following procedure must be performed to execute the examples on a PicoSDR or PicoDigitizer system with an external PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then the PicoSDR/Digitizer back panel.

   Refer to the PicoSDR User's Guide or PicoDigitizer User's Guide documents for details about performing this operation.

2. Connect the Gigabit Ethernet cable between the PicoSDR/Digitizer back panel Ethernet connector and your PC's Gigabit Ethernet network card.
3. Connect the PCI Express expansion cable between the PicoSDR/Digitizer and the your PC's PCIe expansion card.
4. Turn on the PicoSDR/Digitizer.
5. Make sure that the PicoSDR/Digitizer Perseus has a valid MAC and IP addresses.

2.4.2 Setup for execution on a PicoSDR or PicoDigitizer on its embedded PC

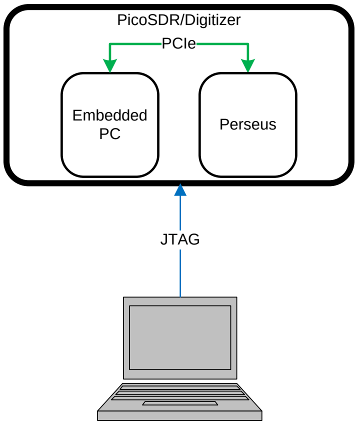


Figure 2-6 PicoSDR/Digitizer with an its embedded PC

The PicoSDR 2x2E, as well as the PicoDigitizer125-16E, 125-32E and 250E contain an embedded PC and a Perseus. The following procedure must be performed to execute the examples on a PicoSDR or PicoDigitizer system with its embedded PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then the PicoSDR/Digitizer back panel.

   Refer to the PicoSDR User's Guide or PicoDigitizer User's Guide documents for details about performing this operation.

2. Turn on the PicoSDR/Digitizer.
3. Make sure that the PicoSDR/Digitizer Perseus has a valid MAC and IP addresses.

2.4.3 Setup for execution on a Perseus in a MicroTCA chassis, with an embedded PC

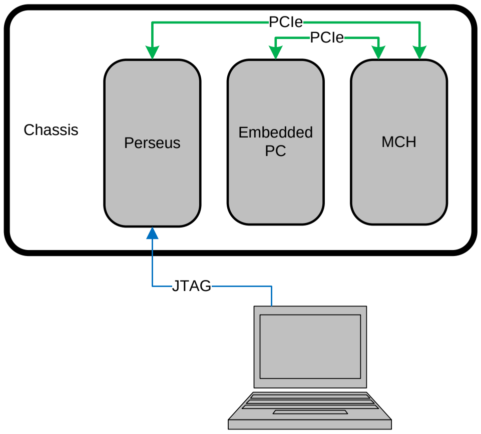


Figure 2-7 Perseus in a MicroTCA chassis, with an embedded PC

The following procedure must be performed to execute the examples in a chassis system with an embedded PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then to the Perseus.

   Refer to the Perseus User's Guide document for details about performing this operation.

2. Insert the Perseus into the µTCA chassis.
3. Turn on the µTCA chassis.
4. Make sure that the Perseus has a valid MAC and IP addresses.

3 Executing the Demos

Two steps are needed in the execution of RTDEx and Record/Playback demos.

1. The FPGA is configured with a pre-compiled *rtdex_recplay* bitstream, or one generated following steps depicted in section 1.2.

   Section 3.1 explains this operation.

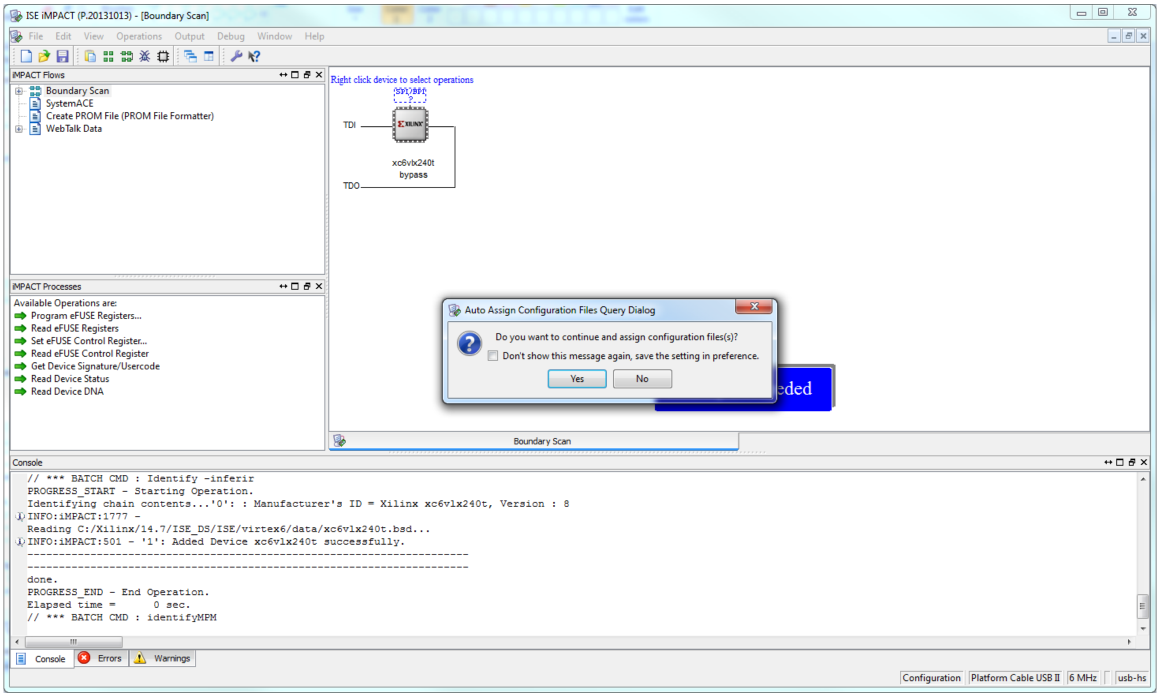
2. A host script is executed. It will configure and enable the various FPGA cores involved in the demo.

   Section 0 explains this operation.

3.1 Configuring the FPGA

1. Make sure your JTAG pod is properly connected to your system as described in section 2.2.
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 a single FPGA device detected. Depending on your setup, iMPACT may detect other devices such as CPLDs and more 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
Get Device Signature/Usercode
Read Device Status
Read Device DNA
Right click device to select operations
SPI/BPI
?
TDI
XILINX
xc6vlx240t
bypass
TDO
Auto Assign Configuration Files Query Dialog
Yes
No
eded
Boundary Scan
Console
Errors
Warnings
Configuration
Platform Cable USB II
6 MHz
usb-hs

Do you want to continue and assign configuration files(s)?

□ Don't show this message again, save the setting in preference.

```
// *** BATCH CMD : Identify -inferir
PROGRESS_START - Starting Operation.
Identifying chain contents...'0': : Manufacturer's ID = Xilinx xc6vlx240t, Version : 8
INFO:iMPACT:1777 -
Reading C:/Xilinx/14.7/ISE_DS/ISE/virtex6/data/xc6vlx240t.bsd...
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 BAS 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 *%BASROOT%\examples\rtdex_recplay\perseus601x* directory.
    - Users operating a system equipped with a Perseus611x must select their bitstream from the *%BASROOT%\examples\rtdex_recplay\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.

3.2 Host Scripts

Scripts showcasing raw RTDEx transfers and data recording and playbacking using the Record/Playback core are available and they can all be run using the bitstream configured in section 3.1. They reside in the *%BASROOT%/examples/rtdex_recplay/host/scripts* directory and they are in the form of batch files (*.bat*) or shell scripts (*.sh*) for Windows or Linux, respectively.

3.2.1 RtdexRxTest

1. Browse to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
2. Using a text editor, open/display
   - *RtdexRxTest.bat* if you use Windows
   - *RtdexRxTest.sh* if you use Linux
3. Change the value of variable *CARRIERIPADDRESS* 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 *%BASROOT%\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	*%BASROOT%*\doc\cores\RTDExTest
RxStreaming	RTDEx	*%BASROOT%*\doc\cores\RTDEx

The script:

- Initializes the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestUplink.ini*.

  Notable parameters of file *RtdexTestUplink.ini*:
  - *adc_dac_mode* is enabled. The RTDEx Test core will simulate an ADC and so will generate data at a precise data rate. The RTDEx core is expected to be able to transfer data to the host at the rate dictated by the RTDEx core.
  - *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.
  - *test_rate* is set to 8. This is the number of clock cycles the RTDEx Test core skips after sending each sample. Because the RTDEx Test core is clocked at 100mhz, the nominal test rate in samples per second is then 100Mhz/(8+1) = 11.11 Msps = 44.44 MBps.
  - *filenames* is not used in this application call. It is used, however, in the last call to the application. See below.
- Enables the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestUplink.ini*.
- Enables RTDEx channel 1 using application *RxStreaming* with configuration file *RxStreaming.ini*.

  Notable parameters of file *RxStreaming.ini*:
  - *RTDEx_framegap* is set to 0. Because the test rate is dictated by the RTDEx Test core, the RTDEx core doesn't use gaps between its frames.
- Stops the test using application *RtdexTestUtil*. The application also validates the ramp contained in the file specified in *RtdexTestUplink.ini* and displays error statistics.

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 RtdexRxTest.bat file.

The test starts automatically.

Expected results:

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

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

Connecting to the board... Done
Action: init



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

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

Connecting to the board... Done
Action: enable



---------- RxStreaming.exe ----------

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

    - Perseus Ip Address : 192.168.0.101
         - RTDEx Media is Gigabit Ethernet

    - General streaming parameters
         - RTDEx frame size : 8192 bytes
         - RTDEx queue size : 10000
         - RTDEx transfer size : 204800000 bytes
         - RTDEx burst size : 65536 bytes
         - Real time flag : 1
         - Display stats flag : 1

    - Total channels number : 1
         - Channel : #1
           - Carrier position : #1
           - Carrier IP address : 192.168.0.101
           - Filename : ../bin/receivedramp.bin
Carrier    :  Channel       DataRate
   #1      :    #1          45.57MBps
Stats
  - RTDEx Packets received (From FPGA, Ch 1): 25000
  - Packets lost (From FPGA, Ch 1): 1
  - Byte(s) received by host (From FPGA, Ch 1): 204800000


  - Full from FPGA throughput:  45.420 MB/s


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

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

Connecting to the board... Done
Action: stop

  - Sample(s) in error (Uplink), Ch 0):                 0
  - Uplink Overflow (4 Bytes sample) (Uplink, Ch 0):      0

Press any key to continue . . .
```

Figure 3-2 RtdexRxTest Windows expected results

NOTE

The success of this test is especially susceptible to the quality of the Ethernet link on the host PC. If poor performance is observed, like Sample(s) in error, try terminating processes that use the Ethernet stack.

3.2.1.1.2 Linux

If using Gigabit Ethernet as RTDEx media, skip to step 4.

1. Once the Perseus is configured and has booted, power cycle the PC to allow it to enumerate the Perseus PCI Express device.
2. Validate the PCI Express link presence. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
3. Install the PCI Express driver on the host PC. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
4. In a Linux terminal, change directory to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
5. To start the example, run the following command

   ```
   sudo ./RtdexRxTest.sh
   ```

Expected results with Gigabit Ethernet:

See expected results for execution on Windows.

NOTE

The success of this test is especially susceptible to the quality of the Ethernet link on the host PC. If poor performance is observed, like Sample(s) in error, try terminating processes that use the Ethernet stack.

3.2.2 RtdexTxTest

1. Browse to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
2. Using a text editor, open/display
   - *RtdexTxTest.bat* if you use Windows
   - *RtdexTxTest.sh* if you use Linux
3. Change the value of variable *CARRIERIPADDRESS* 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 *%BASROOT%\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	*%BASROOT%*\doc\cores\RTDExTest
TxStreaming	RTDEx	*%BASROOT%*\doc\cores\RTDEx

The script:

- Initializes the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestUplink.ini*.

  Notable parameters of file *RtdexTestDownlink.ini*:
  - *adc_dac_mode* is enabled. The RTDEx Test core will simulate a DAC and so will consume data at a precise data rate. The RTDEx core is expected to be able to transfer data from the host to the RTDEx Test core at a rate dictated by it.
  - *rtdex_test_base_custom_register* is set to 0. This parameter specifies to *RtdexTestUtil* the first custom register in the FPGA design that is used for the RTDEx Test core configuration.
  - *test_rate* is set to 8. This is the number of clock cycles the RTDEx Test core skips after consuming each sample. Because the RTDEx Test core is clocked at 100 Mhz, the nominal test rate in samples per second is then 100 Mhz/(8+1) = 11.11 Msps = 44.44 MBps.
  - *filename* is *../bin/generatedramp.bin*. This is the path and the name of the file the *RtdexTestUtil* application will generate. This file will contain a binary ramp which will be transmitted by the *TxStreaming* application.
- Enables the RTDEx Test core using application *RtdexTestUtil* with configuration file *RtdexTestDownlink.ini*.
- Enables RTDEx channel 1 using application T*xStreaming* with configuration file *TxStreaming.ini*.

  Notable parameters of file T*xStreaming.ini*:
  - *RTDEx_flowcontrol* is set to 1 (enabled). This is necessary because the data transfer rate is dictated by the RTDEx Test core. The RTDEx core cannot receive from host its payload and transmit it out faster than the RTDEx Test core consumes it. So, a flow control between the host and the FPGA is enabled to make sure the host doesn't overload the RTDEx data pipe.
- Stops the test using application *RtdexTestUtil*. The application also gathers and displays error statistics.

3.2.2.1 Execution

Execution consists simply 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 RtdexTxTest.bat file.

The test starts automatically.

Expected results:

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

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

Connecting to the board... Done
Action: init



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

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

Connecting to the board... Done
Action: enable



---------- TxStreaming.exe ----------

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

    - Perseus Ip Address : 192.168.0.101
         - RTDEx Media is Gigabit Ethernet

    - General streaming parameters
         - RTDEx frame size : 8192 bytes
         - RTDEx queue size : 10000
         - RTDEx transfer size : 204800000 bytes
         - RTDEx burst size : 65536 bytes
         - Real time flag : 1
         - Display stats flag : 1

    - Total channels number : 1
         - Channel : #1
           - Carrier position : #1
           - Carrier IP address : 192.168.0.101
           - Filename : ../bin/generatedramp.bin
Carrier     :   Channel        DataRate
   #1       :     #1           12.12MBps

Threads stopped
Stats
  - RTDEx Packets transmitted (To FPGA, Ch 1): 25000
  - Packets lost (To FPGA, Ch 1): 0
  - Byte(s) transmitted by host (To FPGA, Ch 1): 204800000


  - Full to FPGA throughput:  12.133 MB/s


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

Parsing RtdexTestDownlink.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):51200000
  - Downlink Underflow (4 Bytes sample) (Downlink, Ch 0):      0

Press any key to continue . . .
```

Figure 3-3 RtdexTxTest Windows expected results

NOTE

The success of this test is especially susceptible to the quality of the Ethernet link on the host PC. If poor performance is observed, like Sample(s) in error, try terminating processes that use the Ethernet stack.

3.2.2.1.2 Linux

If using Gigabit Ethernet as RTDEx media, skip to step 4.

1. Once the Perseus is configured and has booted, power cycle the PC to allow it to enumerate the Perseus PCI Express device.
2. Validate the PCI Express link presence. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
3. Install the PCI Express driver on the host PC. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
4. In a Linux terminal, change directory to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
5. To start the example, run the following command

   sudo ./RtdexRxTest.sh

Expected results with Gigabit Ethernet:

See expected results for execution on Windows.

NOTE

The success of this test is especially susceptible to the quality of the Ethernet link on the host PC. If poor performance is observed, like Sample(s) in error, try terminating processes that use the Ethernet stack.

3.2.3 RecordTest

1. Browse to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
2. Using a text editor, open/display
   - *RecordTest.bat* if you use Windows
   - *RecordTest.sh* if you use Linux
3. Change the value of variable *CARRIERIPADDRESS* 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 *%BASROOT%\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
RecPlayTestUtil	Record/Playback Test	*%BASROOT%*\doc\cores\RecordPlaybackTest

RecordData	Record/Playback	*%BASROOT%*\doc\cores\RecordPlayback
RetrieveRecordedData	Record/Playback	*%BASROOT%*\doc\cores\RecordPlayback

The script:

- Initializes the Record/Playback Test core using application *RecPlayTestUtil* with configuration file *RecplayTestRecord.ini*.
- Configures the Record/Playback core to record 1048576 bytes. Record will start when user logic, in this case, the Record/Playback Test core, starts sending data.
- Enables the Record/Playback Test core.
- Retrieves data from memory using application *RetrieveRecordedData*.
- Stops the test and gathers/displays statistics using application *RecPlayTestUtil*.

3.2.3.1 Execution

Execution consists simply 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 RecordTest.bat file.

The test starts automatically.

Expected results:

```
---------- RecPlayTestUtil ----------

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

Action: init

Connecting to the board... Done


---------- RecordData ----------

         - Trigger source : 2
         - Record size : 1048576 bytes
         - Start address : 0 bytes
         - Trigger delay : 0 bytes

Connecting to the platform at IP address : 192.168.0.101 ...
Resetting Record Playback module...
Setting the Record in no trigger mode...
Start recording...


---------- RecPlayTestUtil ----------

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

Action: enable

Connecting to the board... Done


---------- RetrieveRecordedData ----------

         - RTDEx channel : 0
         - Frame size : 8192 bytes
         - Record size : 1048576 bytes
         - Start address : 0 bytes
         - Record timeout : 1000 ms
         - Filename : ..\bin\recordeddata.bin

Connecting to the platform at IP address : 192.168.0.101 ...
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:20:11:87'
MAC Address of Host is: 'AC:16:2D:06:12:EE'

Setting the Record Playback module in memory to host transfer mode...
1.000 MB received.
Done.



---------- RecPlayTestUtil ----------

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

Action: stop
```

```
Connecting to the board... Done
Error(s) detected when analyzing received ramp file: 0
Success.
Press any key to continue . . .
```

Figure 3-4 RecordTest Windows expected results

3.2.3.1.2 Linux

If using Gigabit Ethernet as RTDEx media, skip to step 4.

1. Once the Perseus is configured and has booted, power cycle the PC to allow it to enumerate the Perseus PCI Express device.
2. Validate the PCI Express link presence. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
3. Install the PCI Express driver on the host PC. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
4. In a Linux terminal, change directory to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
5. To start the example, run the following command

   sudo ./RecordTest.sh

Expected results with Gigabit Ethernet:

See expected results for execution on Windows.

3.2.4 PlaybackTest

1. Browse to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
2. Using a text editor, open/display
   - *PlaybackTest.bat* if you use Windows
   - *PlaybackTest.sh* if you use Linux
3. Change the value of variable *CARRIERIPADDRESS* 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 *%BASROOT%\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
RecPlayTestUtil	Record/Playback Test	*%BASROOT%*\doc\cores\RecordPlaybackTest
LoadDataToPlayback	Record/Playback	*%BASROOT%*\doc\cores\RecordPlayback
RetrieveRecordedData	Record/Playback	*%BASROOT%*\doc\cores\RecordPlayback

The script:

- Initializes the Record/Playback Test core using application *RecPlayTestUtil* with configuration file *RecplayTestRecord.ini*.
- Uses application *LoadDataToPlayback* to load the content of the file generated by *RecPlayTestUtil*.
- Enables the Record/Playback Test core to receive data from the Record/Playback core and analyze it. Analysis begins when the Record/Playback core starts playbacking data on the next step of this script.
- Uses application *PlaybackData* to enable the Record/Playback core to start data transfer from memory and playback.
- Stops the test and gathers/displays statistics using application *RecPlayTestUtil*.

3.2.4.1 Execution

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

3.2.4.1.1 Windows

Double-click the RecordTest.bat file.

The test starts automatically.

Expected results:

```
---------- RecPlayTestUtil ----------

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

Action: init

Connecting to the board... Done


---------- RecordData ----------

         - Trigger source : 2
         - Record size : 1048576 bytes
         - Start address : 0 bytes
         - Trigger delay : 0 bytes

Connecting to the platform at IP address : 192.168.0.101 ...
Resetting Record Playback module...
Setting the Record in no trigger mode...
Start recording...


---------- RecPlayTestUtil ----------

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

Action: enable

Connecting to the board... Done


---------- RetrieveRecordedData ----------

         - RTDEx channel : 0
         - Frame size : 8192 bytes
         - Record size : 1048576 bytes
         - Start address : 0 bytes
         - Record timeout : 1000 ms
         - Filename : ..\bin\recordeddata.bin

Connecting to the platform at IP address : 192.168.0.101 ...
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:20:11:87'
MAC Address of Host is: 'AC:16:2D:06:12:EE'

Setting the Record Playback module in memory to host transfer mode...
1.000 MB received.
Done.



---------- RecPlayTestUtil ----------

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

Action: stop
```

```
Connecting to the board... Done
Error(s) detected when analyzing received ramp file: 0
Success.
Press any key to continue . . .
```

Figure 3-5 PLaybackTest Windows expected results

3.2.4.1.2 Linux

If using Gigabit Ethernet as RTDEx media, skip to step 4.

1. Once the Perseus is configured and has booted, power cycle the PC to allow it to enumerate the Perseus PCI Express device.
2. Validate the PCI Express link presence. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
3. Install the PCI Express driver on the host PC. Follow the instructions given in the *App Note - Using PCI Express* document at the *%BASROOT%/doc/app_notes* location.
4. In a Linux terminal, change directory to *%BASROOT%/examples/rtdex_recplay/host/scripts*.
5. To start the example, run the following command

   ```
   sudo ./PlaybackTest.sh
   ```

Expected results with Gigabit Ethernet:

See expected results for execution on Windows.