Nutaq® INNOVATION TODAY FOR TOMORROW®

MI125

Perseus Examples

October 2015

Revision history

Revision	Date	Comments
0.1	December 2012	First draft
1.0	December 2012	First release
1.1	February 2013	Ready for revision
1.2	February 2013	Linguistic revision
1.3	February 2013	Added Linux BSDK example
1.4	October 2013	Merged Windows and Linux example descriptions
1.5	June 2014	Added BSDK example details Modified setup section to add PicoDigitizer setup instructions
1.6	November 2014	Added *sudo* prefix to Linux shell calls Up to date for Software Tools Release 6.6
1.7	October 2015	Adjusted content and structure for BAS 7.0

Table of Contents

List of Figures

1 FPGA Designs Description and Generation

1.1 MI125 Demos Description

A set of FPGA designs showcasing how to use the MI125 and its corresponding FPGA cores and how they interface with user logic are made available in your BAS software suite. They reside in the *%BASROOT%**examples*\ *mi125* 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	RTDEx media	Carrier
perseus601x\bsdk	Xilinx Platform Studio (BSDK)	Gigabit Ethernet	Perseus601x
perseus601x\mbdk	Nutaq's Model Based Design Kit (MBDK)	Gigabit Ethernet	Perseus601x
perseus611x\bsdk_gige	Xilinx Platform Studio (BSDK)	Gigabit Ethernet	Perseus611x
perseus611x\mbdk	Nutaq's Model Based Design Kit (MBDK)	Gigabit Ethernet	Perseus611x

Table 1 Available FPGA designs depiction

The first two designs, which apply to Perseus601x carriers, support up to two stacked MI125. These designs are functionally identical to each other, and they can be illustrated with Figure 1-1.

The last two designs, which apply to Perseus611x carriers, support up to two double MI125 stacks, that is, four MI125 FMC. These designs are functionally identical to each other, and they can be illustrated with Figure 1-2.

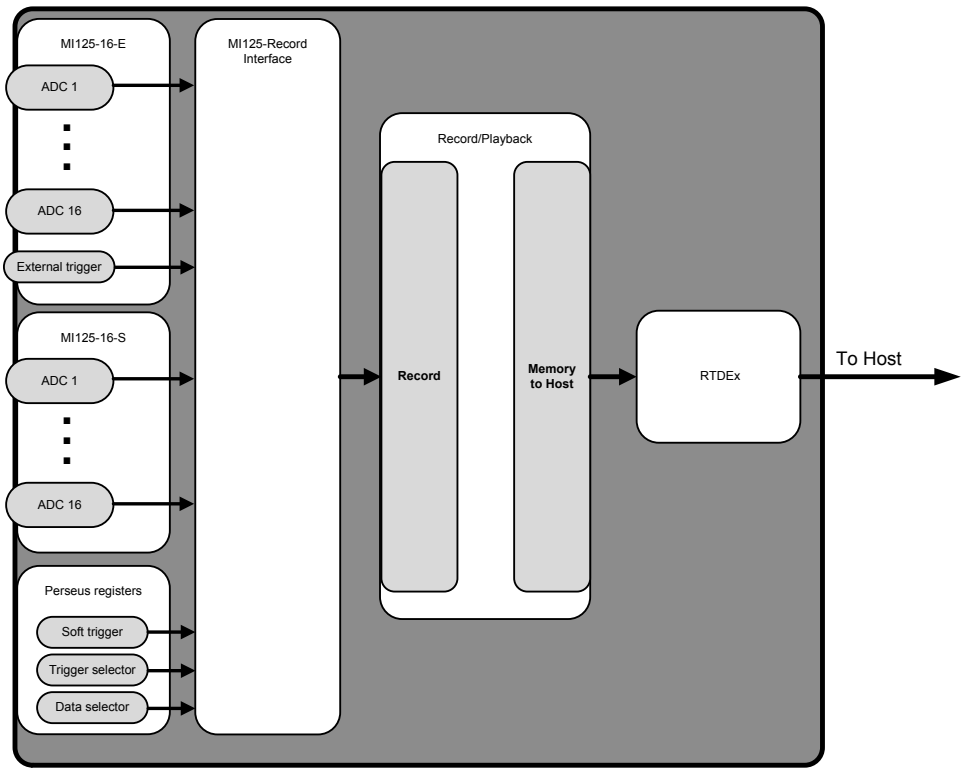

Figure 1-1 MI125 FPGA design schematic for Perseus601x

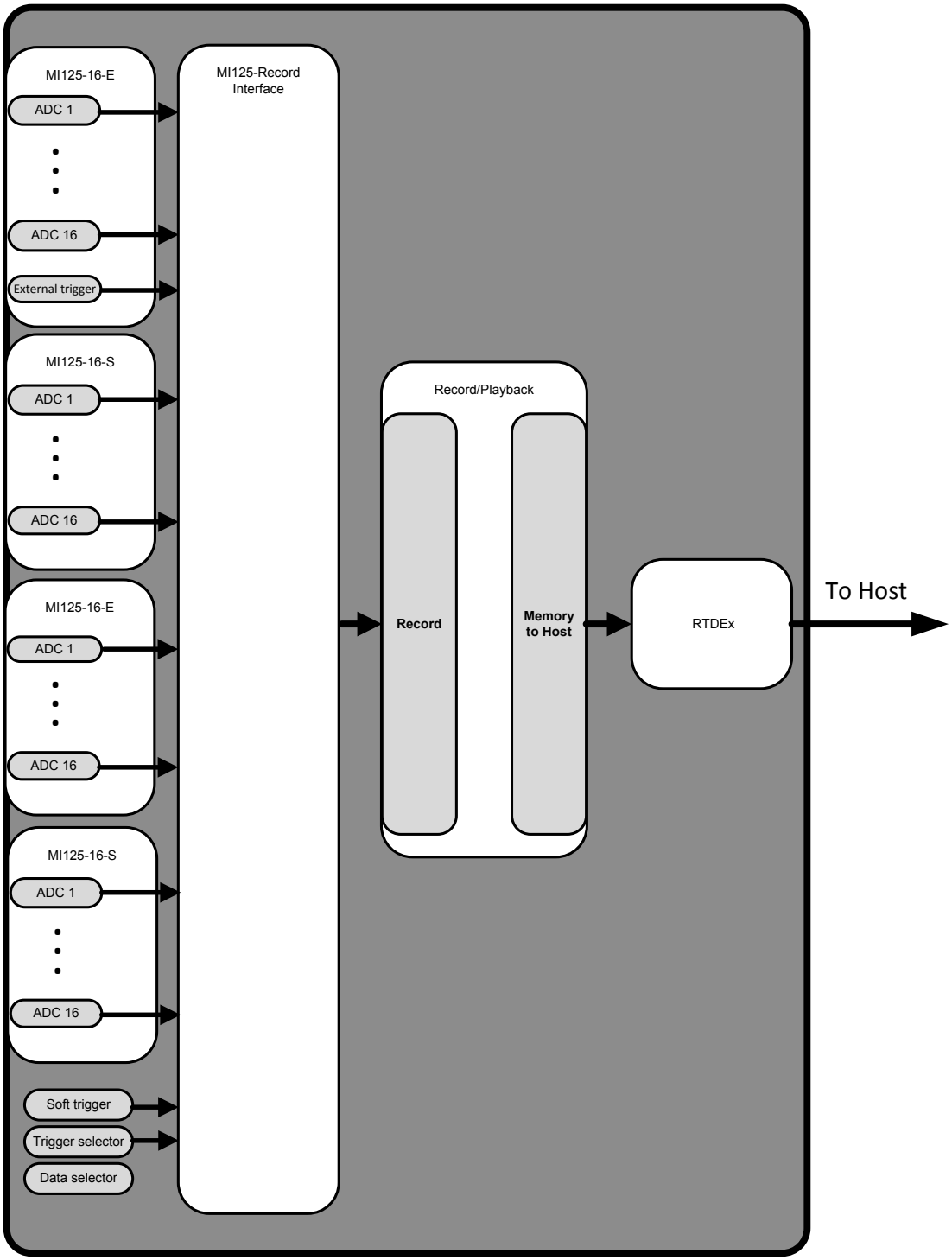

Figure 1-2 MI125 FPGA design schematic for Perseus611x

Data is captured by the MI125 ADCs and is transferred to the MI125 cores on the Perseus. The Record module is instantiated to record 16 ADCs data streams. The MI125-Record interface pcore acts as multiplexer between the MI125 ADCs and the Record/Playback core. More details on the MI125-Record interface pcore operates are available in section 1.1.1.

Configuration of the cores involved in this design is completely automated when running the demo. The example performs the following steps:

1. The host device configures the MI125 hardware.

2. The host device configures the MI125-Record interface to channels 1 to 16 and channels 17 to 32, or to a down-sampled version of channels 1 to 32.
3. The host device configures the MI125-Record interface to external trigger or software trigger.
4. The host device configures the Record/Playback FPGA core.
5. The host device or the external signal triggers the recording and waits for completion.
6. The host device configures the Record/Playback and RTDEx FPGA cores for a memory-to-host transfer.
7. Transfer is executed and data is received.
8. Received data is saved to a file.

1.1.1 MI125 Interface Pcore

MI125 interface pcore is used to connect the MI125 to the record ports. The memory used by the Record module has an effective bandwidth of 5.7 GB/s. A configuration using multiple MI125 quickly overloads this bandwidth. For example, 32 ADCs (two MI125 boards) with a 14-bit resolution at a sample rate of 125 MHz produce a bandwidth of 7 GB/s. Because the memory cannot support the required bandwidth, the MI125-Record interface pcore is used, acting as a data interleaver/downsampler between the ADCs and the Record/Playback core. Also, the MI125 interface pcore allows the Record module to be triggered by the MI125 external trigger or by a software trigger.

If using a Perseus601x carrier with double MI125 stack, the MI125-Record interface pcore may operate in either one of these three functionalities:

- Route all 16 channels from MI125 board 1 to the Record/Playback core. In this mode, no channels of MI125 board 2 are recorded.
- Route all 16 channels from MI125 board 2 to the Record/Playback core. In this mode, no channels of MI125 board 1 are recorded.
- Route all 32 channels from MI125 boards 1 and 2 to the Record/Playback core. This functionality effectively interleaves data from channels of boards 1 and 2 in the carrier memory. Also, this divides the effective sampling rate by two. For example, each channel will be recorded at a sampling rate of 62.5 mhz if the MI125 boards are clocked at 125 mhz.

If using a Perseus611x carrier with two double MI125 stacks, the MI125-Record interface pcore may operate in either one of these seven functionalities:

- Route all 16 channels from MI125 board 1 of connector 1 to the Record/Playback core. In this mode, no channels of the other three MI125 boards are recorded.
- Route all 16 channels from MI125 board 2 of connector 1 to the Record/Playback core. In this mode, no channels of the other three MI125 boards are recorded.
- Route all 16 channels from MI125 board 1 of connector 2 to the Record/Playback core. In this mode, no channels of the other three MI125 boards are recorded.
- Route all 16 channels from MI125 board 2 of connector 2 to the Record/Playback core. In this mode, no channels of the other three MI125 boards are recorded.
- Route all 32 channels from MI125 boards 1 and 2 of connector 1 to the Record/Playback core. This functionality effectively interleaves data from channels of boards 1 and 2 of connector 1 in the carrier memory. Also, this divides the effective sampling rate by two. For example, each channel will be recorded at a sampling rate of 62.5 mhz if the MI125 boards are clocked at 125 mhz. In this mode, no channels from MI125 boards 1 and 2 of connector 2 are recorded.
- Route all 32 channels from MI125 boards 1 and 2 of connector 2 to the Record/Playback core. This functionality effectively interleaves data from channels of boards 1 and 2 of connector 2 in the carrier memory. Also, this divides the effective sampling rate by two. For example, each

channel will be recorded at a sampling rate of 62.5 mhz if the MI125 boards are clocked at 125 mhz. In this mode, no channels from MI125 boards 1 and 2 of connector 1 are recorded.

- Route all 64 channels from the four MI125 boards to the Record/Playback core. This functionality effectively interleaves data from the 64 channels in the carrier memory. Also, this divides the effective sampling rate by four. For example, each channel will be recorded at a sampling rate of 31.25 mhz if the MI125 boards are clocked at 125 mhz.

The table below shows the Perseus custom register number associated with the MI125-Record interface ports. These ports are used to configure the behaviour of the pcore. Also, the table shows the different configurations values allowed by the pcore.

Perseus custom register	MI125-Record interface port	Configuration
13	iv3_MuxSel_p	0 = Select channels from MI125 board 1 of connector 1 1 = Select channels from MI125 board 2 of connector 1 2 = Select channels from MI125 board 1 of connector 2 (applicable on Perseus611x carriers) 3 = Select channels from MI125 board 2 of connector 2 (applicable on Perseus611x carriers) 4 = Select a down-sampled version of all 32 MI125 channels of connector 1 5 = Select a down-sampled version of all 32 MI125 channels of connector 2 (applicable on Perseus611x carriers) 6 = Select a down-sampled version of all 64 MI125 channels of connector 1 and 2 (applicable on Perseus611x carriers)
14	i_SoftOrExternalTriggSel_p	0 = The software trigger is used. 1 = The MI125 external trigger is used.
15	i_SoftTrigger_p	0 = Idle 1 = Software trigger asserted

Table 2 Perseus custom register description

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\mi125\perseus601x\bsdk* folder if using a Perseus601x carrier, or in the *%BASROOT%\examples\mi125\perseus611x\bsdk_gige* folder if using a Perseus611x carrier.
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.

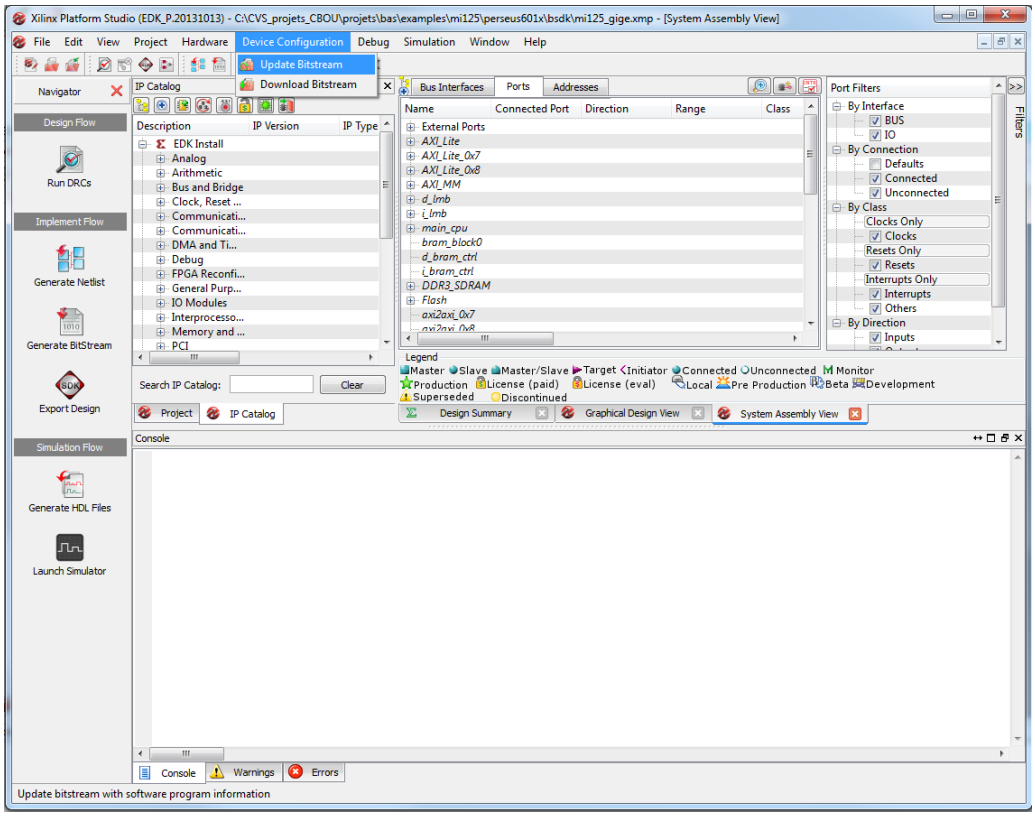

Figure 1-3 BSDK example

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* file 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\mi125\<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 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**.

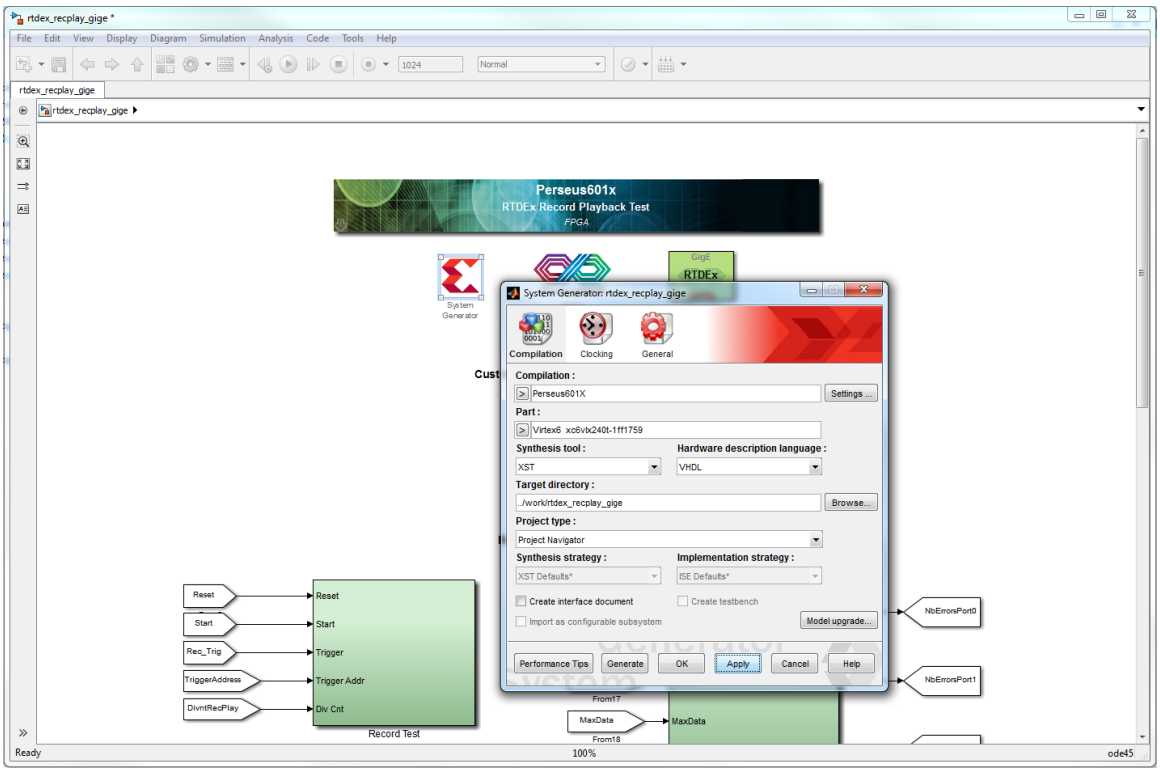

Figure 1-4 MBDK example

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 *mi125_gige.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

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

PicoDigitizer with external PC

- 1 PicoDigitizer125 (any version)
- 1 PC with a Gigabit Ethernet network card (jumbo frame capable)
- 1 Gigabit Ethernet cable
- 1 or 2 Edge Rate Contact™ breakout SMA cables
- 1 FPGA JTAG pod
- 1 2-MHz signal generator

Setup to run the example

PicoDigitizer with embedded CPU

- 1 PicoDigitizer125/16-E or /32-E
- 1 computer Xilinx Impact (to operate the FPGA JTAG pod)
- 1 FPGA JTAG pod
- 1 or 2 Edge Rate Contact™ breakout SMA cables
- 1 2-MHz signal generator

Setup to run the example

Perseus in a µTCA chassis with External PC

- 1 Perseus
- 1 or 2 MI125 FMC
- 1 MicroTCA chassis with its TCA Carrier Hub (MCH)
- 1 computer with a Gigabit Ethernet network card (jumbo frame capable)
- 1 Gigabit Ethernet cable
- 1 FPGA JTAG pod
- 1 or 2 Edge Rate Contact™ breakout SMA cables
- 1 2-MHz signal generator

Setup to run the example

Perseus and embedded PC in a µTCA chassis

- 1 Perseus
- 1 or 2 MI125 FMC
- 1 embedded PC running Linux Fedora or Ubuntu
- 1 MicroTCA chassis with its TCA Carrier Hub (MCH)
- 1 FPGA JTAG pod1 computer with Xilinx Impact (to operate the FPGA JTAG pod)
- 1 or 2 Edge Rate Contact™ breakout SMA cables
- 1 2-MHz signal generator

Setup to run the example

2.2 Hardware Setup

2.2.1 Setup for execution on a PicoDigitizer on an external PC

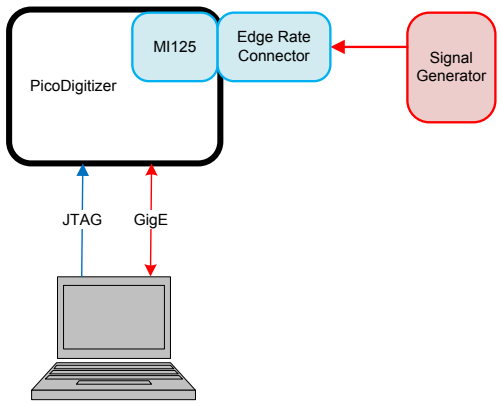

Figure 2-1 PicoDigitizer with an external PC

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

To set up the example:

1. Connect the FPGA JTAG pod to the external PC and then the PicoDigitizer back panel.
   Refer to the PicoDigitizer User's Guide documents for details about performing this operation.
2. Connect the Gigabit Ethernet cable between the external PC and the PicoDigitizer back panel.

2.2.2 Setup for execution on a PicoDigitizer on its embedded PC

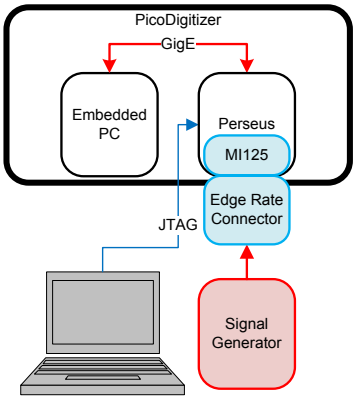

Figure 2-2 PicoDigitizer with an its embedded PC

The PicoDigitizer125-16E and PicoDigitizer125-32E contain an embedded PC and a Perseus. The following procedure must be performed to execute the examples on a PicoDigitizer system with its embedded PC.

To set up the example:

1. Connect the FPGA JTAG pod to the computer and then the PicoDigitizer back panel.
   Refer to the PicoDigitizer User's Guide documents for details about performing this operation.

2.2.3 Setup for execution in a MicroTCA chassis

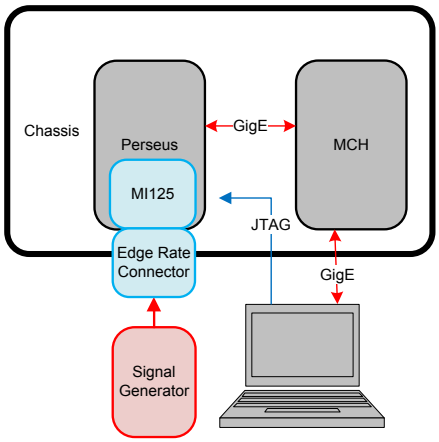

Figure 2-3 Perseus in a MicroTCA chassis, with an external PC

1. If not already done, install the MI125 cards on the Perseus.
2. If not already done, connect the FPGA JTAG pod to the computer and then to the Perseus.
   Refer to the FPGA JTAG pod documentation for details about performing this operation.
3. Connect the Gigabit Ethernet cable between the MCH's front panel Ethernet connector and your PC's Gigabit Ethernet network card.
4. If not already done, insert the Perseus into the µTCA chassis.

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

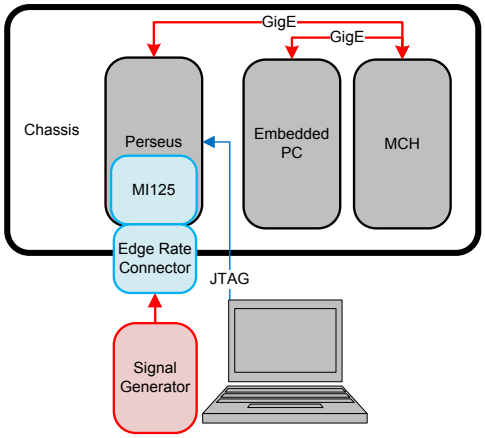

Figure 2-4 Perseus in a MicroTCA chassis, with an external PC

1. If not already done, install the MI125 cards on the Perseus.
2. If not already done, connect the FPGA JTAG pod to the computer and then to the Perseus. Refer to the FPGA JTAG pod documentation for details about performing this operation.
3. If not already done, insert the Perseus into the µTCA chassis.

2.2.5 Signal generator setup

1. Connect the Edge Rate Contact breakout SMA cables to the MI125 front panel connectors labeled 1 to16 and 17 to 32.
2. Connect a 1-Vpp (4 dBm) sine at 2 MHz to each channels.

   The channel inputs must be less than 2 Vpp in amplitude to avoid the risk of damaging the hardware. If you are unable to connect a signal to all the channels, connect signals to as many as possible, and then change the connections to test all the channels. If you have only one Edge Rate Contact breakout SMA cables, start testing channels 1 to 16 and then connect it to channels 17 to 32.
3. Turn on the µTCA chassis or the PicoDigitizer
4. Make sure that the Perseus has valid MAC and IP addresses.

3 Executing the Demos

Two steps are needed in the execution of MI125 demos.

1. The FPGA is configured with a pre-compiled *mi125_gige* 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 3.2 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.

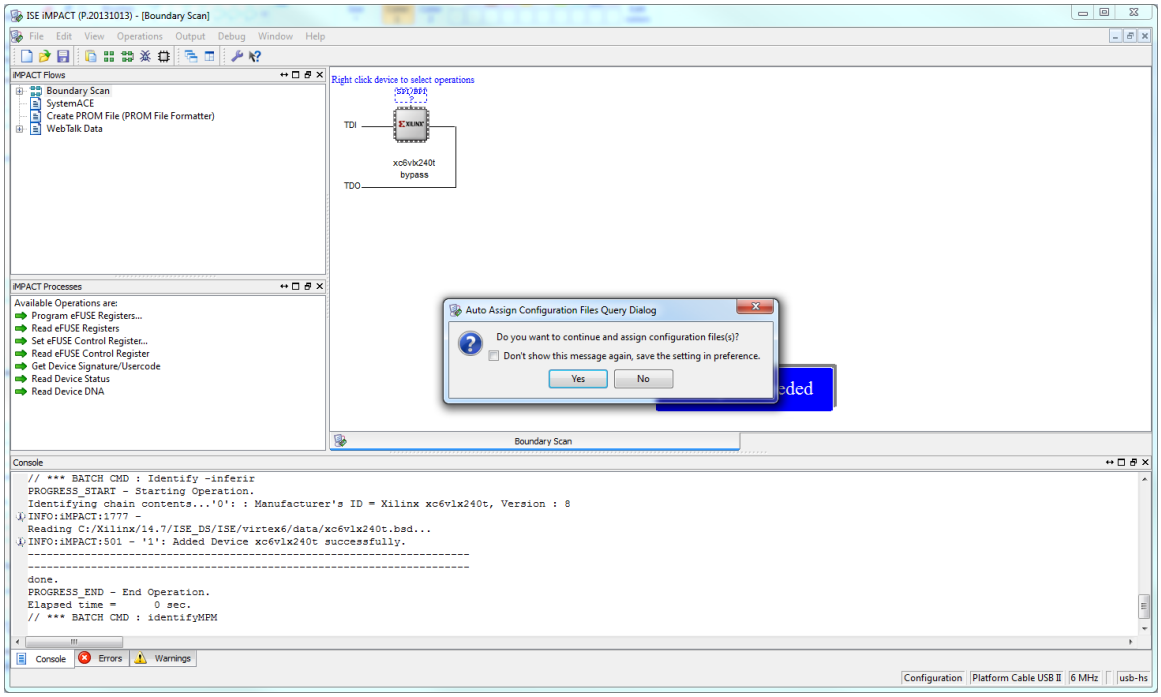

Figure 3-1 After iMPACT Boundary Scan

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\mi125\perseus601x* directory.
    - Users operating a system equipped with a Perseus611x must select their bitstream from the *%BASROOT%\examples\mi125\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

A script showcasing MI125 operation and data recording is available and it can be run using the bitstream configured in section 3.1. It resides in the *bas/examples/mi125/host/scripts* directory and it is in the form of a batch file (*.bat*) or a shell script (*.sh*) for Windows or Linux, respectively. By default, they are written to operate a two-board MI125 stack.

3.2.1 MI125_Record_32

1. Browse to *bas/examples/mi125/host/scripts*.
2. Using a text editor, open/display
   - *MI125_Record_32.bat* if you use Windows
   - *MI125_Record_32.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
MI125_Init	MI125	%BASROOT%\doc\cores\RTDExTest
RecordData	Record/Playback	%BASROOT%\doc\cores\RecordPlayback
RetrieveRecordedData	Record/Playback	%BASROOT%\doc\cores\RecordPlayback

The script:

- Initializes the MI125 cores and boards using application *MI125_Init* with configuration file *MI125_Init.ini*.
- Configures the MI125-Record interface pcore to route all 32 channels of the MI125 stack to the Record/Playback core, downsampling signals by 2. As parameters for the MI125-Record interface pcore are routed out to custom registers of the Perseus61xx_regs pcore, application *CustomRegister_Write* (on register 13) is used to perform the operation.
- Configures the MI125-Record interface pcore to use a software signal to trig the Record/Playback core, using application *CustomRegister_Write* (on register 14).
- Resets the software trigger using application *CustomRegister_Write* (on register 15).
- Configures the Record/Playback core to record 65600 bytes, with an external trigger signal, using application *RecordData*. In the FPGA design used in this demo, the trigger signal is always external as seen from the Record/Playback core, even if the trigger signal is chosen to be software. This is because it is managed externally by the MI125-Record interface core.

  Following this step, the Record/Playback core starts writing MI125 data to memory as soon as the MI125 cores have received samples and are ready to send them, regardless of when the trigger signal is detected.
- Sets the software trigger using application *CustomRegister_Write* (in register 15). In the FPGA design, custom register 15 is routed to the MI125-Record interface core which in turn routes the trigger signal to the Record/Playback core. When the trigger signal is detected by the Record/Playback core, the core stores the address in memory of its write pointer (the trigger address) and starts counting bytes it writes to memory. The Record/Playback core stops writing to memory when the record size is reached.

- Retrieves data from memory using application *RetrieveRecordedData*.

3.2.1.1 Execution

Execution consists simply 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 MI125_Record_32.bat file.

The test starts automatically.

3.2.1.1.2 Linux

1. In a Linux terminal, change directory to *bas/examples/mi125/host/scripts*.
2. To start the example, run the following command

   sudo ./MI125_Record_32.sh

3.2.1.1.3 Expected Results

```
---------- MI125_Init ----------

Parsing MI125_Init.ini file for needed parameters...Done!

Configuring the Perseus at IP = 192.168.0.101, please wait.
MI125 Init...
   - Board number : 1
         - Sampling clock source : 0
         - Trigger output IO : OFF
         - Is a clock master : Yes.
         - Core Version: 0x0201
         - Driver Version: 0x0023
         - PCB temp: 48.5C
   - Board number : 2
         - Sampling clock source : 2
         - Trigger output IO : OFF
         - Is a clock master : No.
         - Core Version: 0x0201
         - Driver Version: 0x0023
         - PCB temp: 45.5C
Done!


---------- CustomRegister_Write -----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
CustomRegister Write...
         - Custom Register ID : 13
         - Value to write : 4
Done!


---------- CustomRegister Write -----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
CustomRegister Write...
         - Custom Register ID : 14
         - Value to write : 0
Done!


---------- CustomRegister_Write -----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
CustomRegister Write...
         - Custom Register ID : 15
```

```
         - Value to write : 0
Done!


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

         - Trigger source : 0
         - Record size : 65600 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 trigger in external mode...
Start recording...


---------- CustomRegister_Write -----------

Configuring the Perseus at IP = 192.168.0.101, please wait.
CustomRegister Write...
         - Custom Register ID : 15
         - Value to write : 1
Done!


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

         - RTDEx channel : 0
         - Frame size : 8192 bytes
         - Record size : 65536 bytes
         - Start address : trigger
         - Record timeout : 1000 ms
         - Filename : ..\bin\record.bin

Connecting to the platform at IP address : 192.168.0.101 ...
Trigger position: 81604064
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:0A:01:5F'
MAC Address of Host is: '10:60:4B:79:68:3C'

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

Press any key to continue . . .
```

The file containing the sampled data is then saved in the repository specified in the last argument of RetrieveRecordedData (line Filename in the console output of RetrieveRecordedData).

If Matlab is installed on the host PC, it is possible to visualize the data.

1. In Matlab, change the current folder to the folder where record.bin resides.
2. Run the following command: bintofft('filename', num_of_channels_to_plot, data_type, framesize, acquisition_frequency);.

   For example, *bintofft('record.bin', 32, 'int16', inf, 62.5e6);*.

   This will plot the time and the frequency domain for the 32 acquired channels. If 32 channels are stored in the binary file, the number of channels in the function argument must be changed to 32.
3. Matlab will plot the received data and the FFT for each channel. The figure below shows the result for a sine wave of 2 MHz, 1 Vpp (4 dBm).

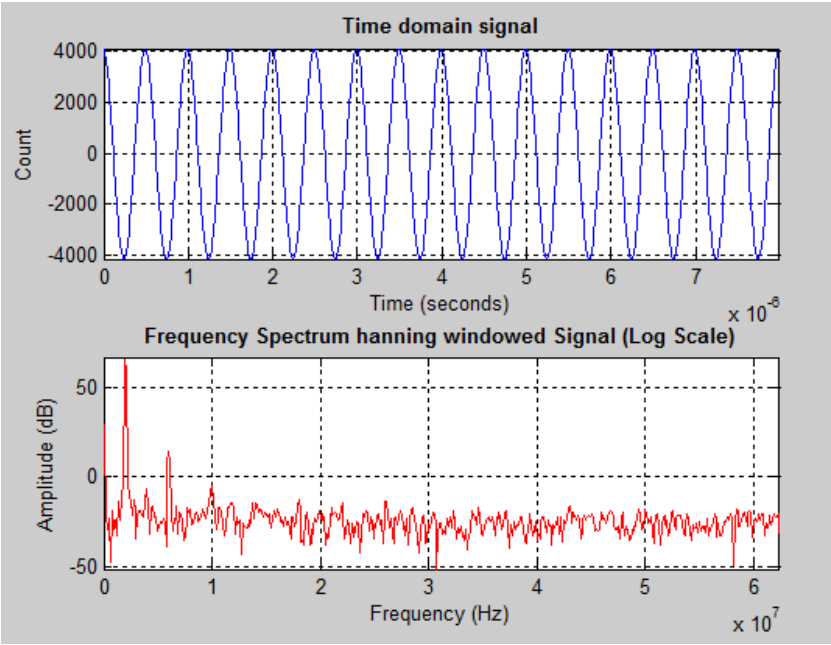

Figure 3-2 Matlab data plot result

3.3 Command Line Interface Scripts

A Command Line Interface (CLI) script showcasing MI125 operation and data recording is available and it can be run using the bitstream configured in section 3.1. It resides in the *%BASROOT%/examples/mi125/host/cli*. This example is in the form of a series of CLI commands saved in txt files. On Windows, an example batch file launches the CLI using another batch file called LaunchCLI.bat and passes the CLI commands to the CLI using the example txt files.

On Linux, the same principle applies, but the example uses shell scripts.

3.3.1 Executing launch_mi125

1. Browse to *%BASROOT%/examples/mi125/host/cli*.
2. Using a text editor, open/display
   - *launch_mi125_init_32ch.txt* and
   - *launch_mi125_record_32ch.txt*
3. In both files, make sure the IP address used as an argument to command *connect* at the beginning of the files matches your system configuration. This is the IP address of the carrier running the demo.
4. Save the script files.

3.3.1.1 Windows

Double-click the launch_mi125.bat file.

The test starts automatically.

3.3.1.2 Linux

1. In a Linux terminal, change directory to *bas/examples/mi125/host/cli*.
2. To start the example, run the following command

   sudo ./launch_mi125.sh

3.3.1.3 Expected Results

```
-----------------------------------
       Configuring MI125 FMCs
-----------------------------------
Target connecting...
Connected to Perseus601X carrier board
Detected MI125 FMC
DONE
Mi125 1 initialization...
MI125 fpga core version:  513
MI125 firmware driver version:  35
Board pcb temperature is:  37
Mi125 2 initialization...
Board pcb temperature is:  36
DONE
-----------------------------------
       Configuring Record mode
-----------------------------------
Target connecting...
Connected to Perseus601X carrier board
Detected MI125 FMC
Configure 32 channels mode, with downsample by 2...
DONE
Configure mi125 core trigger mode internal...
DONE
Recplay_record at 125MS/sec for 65536+64 bytes...
Resetting Record Playback module...
```

```
Setting the Record trigger in external mode...
Start recording...
DONE
Waiting for trigger and record data ready...
DONE
Retrieve data to host...
Trigger position: 53616960
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:0A:01:47'
MAC Address of Host is: '10:60:4B:79:68:3C'

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

DONE
```

The file containing the sampled data is then saved in the repository specified in the last argument of command RetrieveRecordedDataToFile of file *launch_mi125_record_32ch.txt*.

If Matlab is installed on the host PC, it is possible to visualize the data.

1. In Matlab, change the current folder to the folder where record.bin resides.
2. Run the following command: bintofft('filename', num_of_channels_to_plot, data_type, framesize, acquisition_frequency);.

   For example, *bintofft('record.bin', 32, 'int16', inf, 62.5e6);*.

   This will plot the time and the frequency domain for the 32 acquired channels. If 32 channels are stored in the binary file, the number of channels in the function argument must be changed to 32.
3. Matlab will plot the received data and the FFT for each channel. The figure below shows the result for a sine wave of 2 MHz, 1 Vpp (4 dBm).

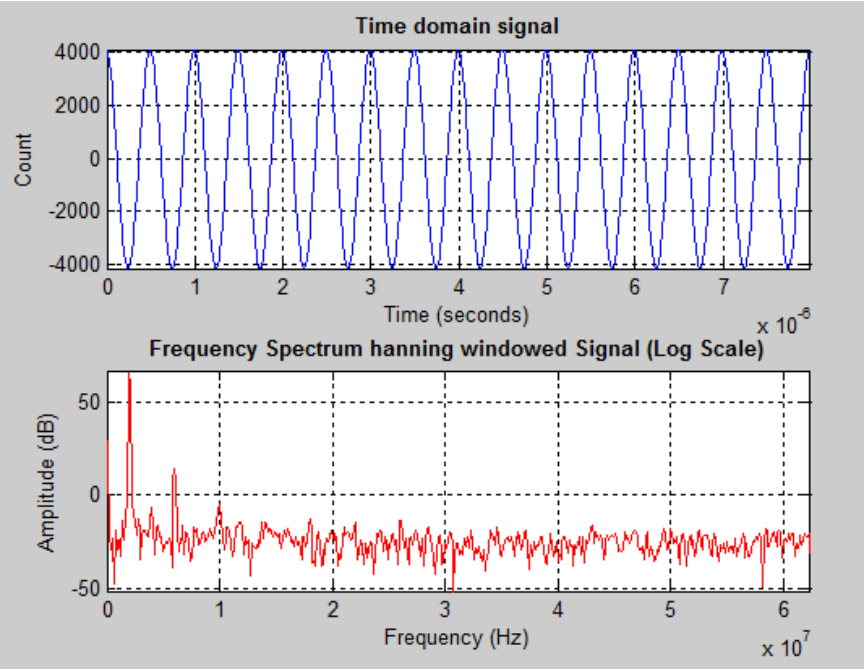

Figure 3-3 Matlab data plot result