Nutaq® INNOVATION TODAY FOR TOMORROW®

PicoDigitizer125-64 2nd Generation

User's Guide

March 2017

Revision history

Revision	Date	Comments
0.9	July 2016	First draft.
1.0	March 2017	Revised document Removed temporary comments and highlighting Diagram of section 7.1 redone

To ensure the accuracy of the information contained herein, particular attention was given to usage in preparing this document. It corresponds to the product version manufactured prior to the date appearing on the title page. There may be differences between the document and the product, if the product was modified after the production of the document.

Nutaq reserves itself the right to make changes and improvements to the product described in this document at any time and without notice.

Trademarks

Acrobat, Adobe, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. IBM is a registered trademark of International Business Machines Corporation in the United States, other countries, or both. Intel and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Microsoft, MS-DOS, Windows, Windows NT, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. MATLAB, Simulink, and Real-Time Workshop are registered trademarks of The MathWorks, Inc. Xilinx, Spartan, and Virtex are registered trademarks of Xilinx, Inc. Texas Instruments, Code Composer Studio, C62x, C64x, and C67x are trademarks of Texas Instruments Incorporated. All other product names are trademarks or registered trademarks of their respective holders.

The TM and ® marks have been omitted from the text.

Warning

Do not use Nutaq products in conjunction with life-monitoring or life-critical equipment. Failure to observe this warning relieves Nutaq of any and all responsibility.

FCC warning

This equipment is intended for use in a controlled environment only. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of personal computers and peripherals pursuant to subpart J of part 15 of the FCC rules. These rules are designed to provide reasonable protection against radio frequency interference. Operating this equipment in other environments may cause interference with radio communications, in which case the user must, at his/her expense, take whatever measures are required to correct this interference.

This page was left intentionally blank.

Table of Contents

List of Figures and Tables

1 Introduction

Congratulations on the purchase of the PicoDigitizer125!

This document contains all the information necessary to understand and use the PicoDigitizer125. It should be read carefully before using the card and stored in a handy location for future reference.

1.1 Conventions

In a procedure containing several steps, the operations are numbered (1, 2, 3…). The diamond (♦) is used to indicate a procedure containing only one step, or secondary steps. Lowercase letters (a, b, c…) can also be used to indicate secondary steps in a complex procedure.

The abbreviation NC is used to indicate no connection.

Capitals are used to identify any term marked as is on an instrument, such as the names of connectors, buttons, indicator lights, etc. Capitals are also used to identify key names of the computer keyboard.

All terms used in software, such as the names of menus, commands, dialog boxes, text boxes, and options, are presented in bold font style.

The abbreviation N/A is used to indicate something that is not applicable or not available at the time of press.

Note:
The screen captures in this document are taken from the software version available at the time of press. For this reason, they may differ slightly from what appears on your screen, depending on the software version that you are using. Furthermore, the screen captures may differ from what appears on your screen if you use different appearance settings.

1.2 Glossary

This section presents a list of terms used throughout this document and their definition.

Term	Definition
Advanced Mezzanine Card (AMC)	AdvancedMC is targeted to requirements for the next generation of "carrier grade" communications equipment. This series of specifications are designed to work on any carrier card (primarily AdvancedTCA) but also to plug into a backplane directly as defined by MicroTCA specification.
Advanced Telecommunications Computing Architecture (or AdvancedTCA, ATCA)	AdvancedTCA is targeted primarily to requirements for "carrier grade" communications equipment, but has recently expanded its reach into more ruggedized applications geared toward the military/aerospace industries as well. http://en.wikipedia.org/wiki/Advanced_Telecommunications_Computing_Architecture - cite_note-3 This series of specifications incorporates the latest trends in high speed interconnect technologies, next-generation processors, and improved Reliability, Availability and Serviceability (RAS).
Application Programming Interface (API)	An application programming interface is the interface that a computer system, library, or application provides to allow requests for services to be made of it by other computer programs or to allow data to be exchanged between them.
Board Software Development Kit (BSDK)	The board software development kit gives users the possibility to quickly become fully functional developing C/C++ for the host computer and HDL code for the FPGA through an understanding of all Nutaq boards major interfaces.
Boards and Systems (BAS)	Refers to the division part of Nutaq which is responsible for the development and maintenance of the hardware and software products related to the different Perseus carriers and their different FMC daughter cards.
Carrier	Electronic board on which other boards are connected. In the FMC context, the FMC carrier is the board on which FMC connectors allow a connection between an FMC card and an FPGA. Nutaq has two FMC carriers, the Perseus6113 (1 FMC site) and the Perseus611x (2 FMC sites).
Central Communication Engine (CCE)	The Central Communication engine (CCE) is an application that executes on a virtual processor called a MicroBlaze in the FPGA of the Perseus products. It handles all the behavior of the Perseus such as module initialization, clock management, as well as other tasks.
Chassis	Refers to the rigid framework onto which the CPU board, Nutaq development platforms, and other equipment are mounted. It also supports the shell-like case—the housing that protects all the vital internal equipment from dust, moisture, and tampering.
Command Line Interface (CLI)	The Command Line Interface (or CLI) is a basic client interface for Nutaq's FMC carriers. It runs on a host device. It consists of a shell where commands can be typed, interacting with the different computing elements connected to the system.
FPGA Mezzanine Card (FMC)	FPGA Mezzanine Card is an ANSI/VITA standard that defines I/O mezzanine modules with connection to an FPGA or other device with re-configurable I/O capability. It specifies a low profile connector and compact board size for compatibility with several industry standard slot card, blade, low profile motherboard, and mezzanine form factors.
HDL	Stands for hardware description language.
Host	A host is defined as the device that configures and controls a Nutaq board. The host may be a standard computer or an embedded CPU board in the same chassis system where the Nutaq board is installed. You can develop applications on the host for Nutaq boards through the use of an application programming interface (API) that comprises protocols and functions necessary to build software applications. These API are supplied with the Nutaq board.
MicroTCA (or μTCA)	The MicroTCA (μTCA) specification is a PICMG Standard which has been devised to provide the requirements for a platform for telecommunications equipment. It has been created for AMC cards.
Model-Based Design	Refers to all the Nutaq board-specific tools and software used for development with the boards in MATLAB and Simulink and the Nutaq model-based design kits.
Model-Based Development Kit (MBDK)	The model-based development kit gives users the possibility to create FPGA configuration files, or bitstreams, without the need to be fluent in VHDL. By combining Simulink from Matlab, System Generator from Xilinx and Nutaq's tools, someone can quickly create fully-functional FPGA bitstreams for the Perseus platforms.
NTP	Network Time Protocol. NTP is a protocol to synchronize the computer time over a network.

Term	Definition
Peer	A host peer is an associated host running RTDEx on either Linux or Windows. An FPGA peer is an associated FPGA device.
PicoDigitizer / PicoSDR Systems	Refers to Nutaq products composed of Perseus AMCs and digitizer or SDR FMCs in a table top format.
PPS	Pulse per second. Event to indicate the start of a new second.
Reception (Rx)	Any data received by the referent is a reception.
Reference Design	Blueprint of an FPGA system implemented on Nutaq boards. It is intended for others to copy and contains the essential elements of a working system (in other words, it is capable of data processing), but third parties may enhance or modify the design as necessary.
Transmission (Tx)	Any data transmitted by the referent is a transmission. Abbreviated TX.
µDigitizer / µSDR Systems	Any Nutaq system composed of a combination of µTCA or ATCA chassis, Perseus AMCs and digitizer or SDR FMCs.
VHDL	Stands for VHSIC hardware description language.

Table 1 Glossary

1.3 Technical Support

Nutaq is firmly committed to providing the highest level of customer service and product support. If you experience any difficulties using our products or if it fails to operate as described, first refer to the documentation accompanying the product. If you find yourself still in need of assistance, visit the technical support page in the Support section of our Web site at www.nutaq.com.

2 Product Description

The PicoDigitizer125 system combines different Nutaq components to offer its functionalities. These cards are:

- The Perseus6113 carrier board equipped with an SX475T Virtex-6 FPGA
- Two MI125 FMC stack
- The AMC726 embedded i7 PC (optional)
- The custom µTCA backplane board offering two AMC slots for a Perseus6113 and an AMC726.

These components are detailed in the following section of this document.

2.1 Component Details

This section presents details about the inner workings of the PicoDigitizer125. FPGA developers may want to study this section to get a better understanding of how the various interfaces are arranged and wired inside the PicoDigitizer125.

2.1.1 Perseus6113 Carrier

The Perseus6113 AMC carrier is at the heart of the PicoDigitizer125 system. The Perseus6113 advanced mezzanine card (AMC or AdvancedMC) is designed around the powerful Virtex-6 SX475T FPGA, combining unsurpassed fabric flexibility and a colossal external memory, as well as benefiting from two high-pin-count (HPC), modular, add-on FMC-based I/O cards.

The Perseus is intended for high-performance, high-bandwidth, low-latency processing applications. The card also takes full advantage of the Virtex-6 FPGA power, which, when combined with Nutaq's advanced software development tools, makes the Perseus perfect for reducing size, complexity, risks and costs associated to leading-edge telecommunications, networking, industrial, defense and medical applications. On top of this, the Perseus' FMC expansion sites offer almost endless I/O possibilities.

The user's guide for this product can be found in the %BASROOT%\doc\perseus611x folder of the Software Tools installation.

2.1.2 AMC726

The AMC726 is a mid-size processor AMC board designed around the fourth generation Intel Core i7 processor (Haswell) that comes with the PicoDigitizer125-E units (embedded PC option). This module follows the AMC.1, AMC.2 and the AMC.3 specifications and provides 16 GB of DDR3 memory with ECC and 64 GB of Flash memory for the OS.

Ubuntu 12.04 LTS as well as the latest PicoDigitizer125 Software tools come installed on the AMC726 of your PicoDigitizer125.

The following table lists the AMC726 default passwords.

AMC726 Linux passwords			
User	nutaq	admin (sudo)	superuser (su)
Password	nutaq	nutaq	nutaqrd

Table 2 AMC726 Linux passwords

For details on the AMC726 embedded computer operation and its specifications, please refer to the AMC726 datasheet.

2.1.3 PicoDigitizer125 Backplane and Rear Transition Module (RTM)

The backplane supplies the power to the Perseus6113, the AMC726 embedded PC, the fans, and the Ethernet switch. It can also be used to select power cycles for the Perseus6113 through the push button or the console menu. The backplane and RTM also provide interfaces to multiple devices. Please see Figure 2-1 for a detailed block diagram of the backplane and RTM functionalities.

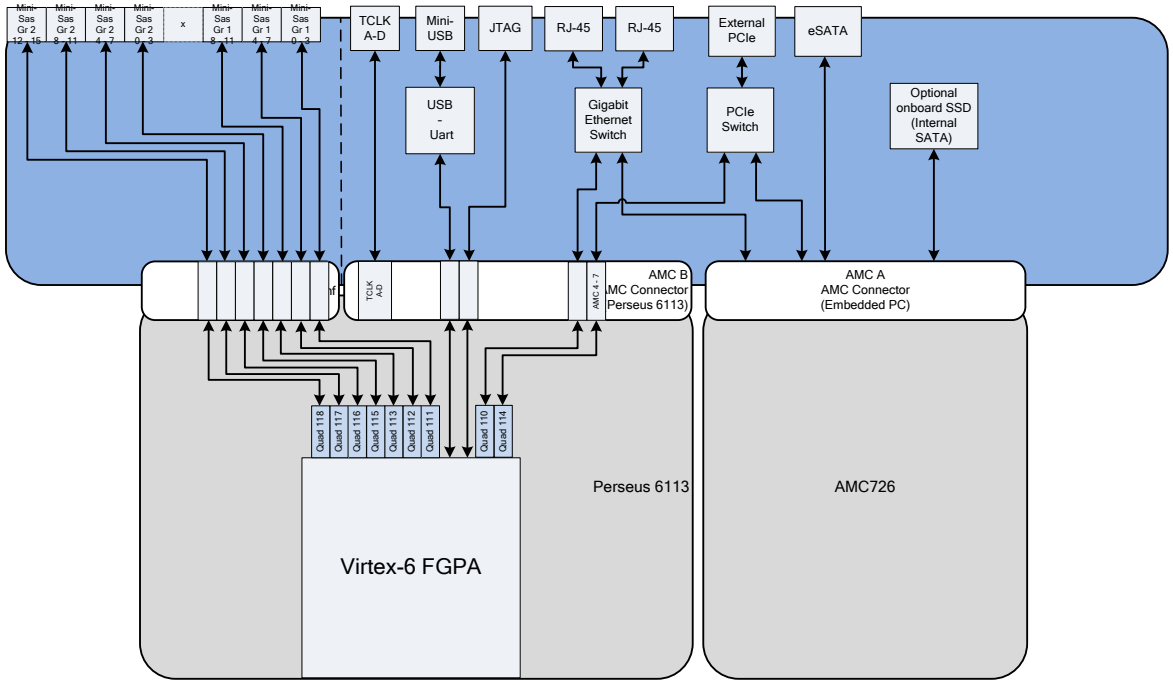

Figure 2-1 PicoDigitizer125 Backplane and Rear Transition Module

PicoDigitizer125 Back Panel

On both PicoDigitizer125 models, the back panel gives access to the Perseus6113 debug console, control and data streaming ports.

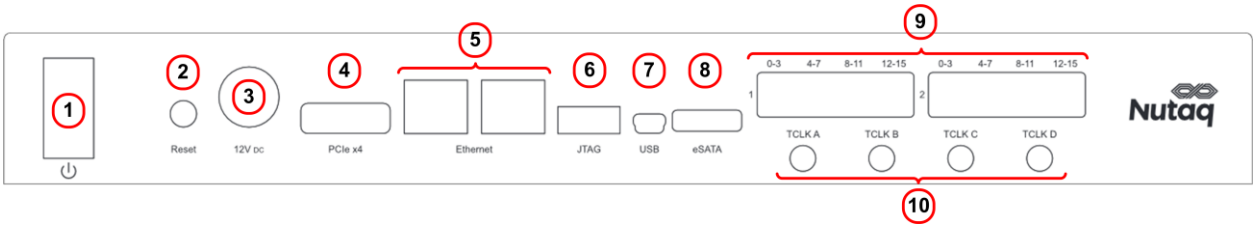

Figure 2-2 PicoDigitizer125 back panel

1 - Power switch

The power switch manages the power of the entire system. By default, when placed in ON position from the OFF position, the system will go through a boot sequence ensuring proper functionality. If the system doesn't go through the boot sequence, verify the autostart flag in your backplane configuration.

2 - Reset button

The Reset button allows to power cycle the embedded computer; to allow for PCIe enumeration to be executed on the embedded computer after a bitstream supporting PCIe has been loaded to the FPGA of the Perseus.

3 - Power-supply DIN-8 connector

Connector compatible with the DIN-8 power-supply provided with the system.

4 - External PCIe connector

The external PCIe connector allows the exchange of data between the Perseus6113 FPGA and a Linux PC with the help of a 4x PCIe cable.

5 - Ethernet connectors

The Ethernet RJ-45 connectors allow Gigabit Ethernet communication between the Perseus6113 FPGA and an external Windows or Linux PC.

6 - JTAG

This connector enables the connection of a JTAG pod to program and interact with the FPGA of the Perseus6113 in the system.

7 - USB connector

Gives access to the backplane and Perseus serial consoles.

8 - eSATA connector

The external eSATA offers an additional connector to the backplane which can be used to access an external SATA hard disk drive from the embedded computer of a PicoDigitizer125-64-E. The eSATA connector functionality is disabled in the PicoDigitizer125-64 models since no embedded PC is present.

9 - Mini-SAS/RTM connectors

The mini-SAS connectors allow the inter-connection of the PicoDigitizer125 to other systems through the use of the Aurora bus. Group 1 connector 12-15 is not connected.

10- TCLK connectors

The TCLK connectors on the back panel make it possible to connect trigger or clock signals to the PicoDigitizer125.

GTX Interfaces

In the PicoDigitizer125, the Rear Transition Module (RTM) provides a direct interface to 7 GTX Quads of the Perseus 6113. The interface is provided through 7 Mini-SAS connectors divided into 2 groups. These 7 Mini-SAS connectors are installed on the PicoDigitizer125 back panel. Please note that Mini-SAS group 1 connector 12 – 15 is not connected and should not be used. Please refer to Table 3 for the complete mapping.

Connector groups	Virtex-6 FPGA MGT number (GTX Quad)
Mini-SAS group 1 connector 0-3	111
Mini-SAS group 1 connector 4-7	112
Mini-SAS group 1 connector 8-11	113
Mini-SAS group 2 connector 0-3	115
Mini-SAS group 2 connector 4-7	116
Mini-SAS group 2 connector 8-11	117
Mini-SAS group 2 connector 12-15	118

Table 3 Mini-SAS connectors to MGT number mapping

The backplane also uses one GTX Quad for implementing PCI-express. PCIe supports a 4x connection on generation 1 (2.5 Gbps) between a Perseus6113 and a Linux PC.

An external PCIe connection is also available to the Perseus6113. It can be enabled with the use of an analog switch controllable from the backplane menu.

Ethernet Switch and RJ-45 Connectors (AMC ports 0 and 1)

The Ethernet switch allows communication between slots A-B and the RJ-45 connectors through the AMC ports 0 and direct communication between slots A and B through port 1. The two RJ-45 connectors connected to the Ethernet switch allow the daisy-chaining of multiple PicoDigitizers, reducing the need for unnecessary cabling.

FPGA JTAG Connector

The FPGA JTAG connection allows the user to program or monitor the FPGA on the Perseus6113.

USB - Uart Serial Connection (AMC port 15)

The USB-Uart serial connection allows the user to connect to the PicoDigitizer125 console menu or the Perseus6113 Linux console. The PicoDigitizer console menu gives the possibility to power cycle each slot or to turn off the system. For more information, refer to section 6.3 “PicoDigitizer125 Serial Console Menu”.

Internal SATA Connection (AMC port 2)

The internal SATA connection allows the user to access the optional 250 GB SSD from the embedded computer on the PicoDigitizer125-64-E.

External eSATA Connection (AMC port 3)

The external eSATA offers an additional connector to the backplane which can be used to access an external SATA hard disk drive from the embedded computer of a PicoDigitizer125-64-E. The eSATA functionality is disabled in the PicoDigitizer125-64 models since no embedded PC is present.

2.1.4 MI125 FMC

The MI125 FPGA mezzanine card (FMC) is a 16 channels phased aligned A/D card designed around the high-performance LTM9012 QUAD ADC from Linear Technology. The MI125 is stackable, which allows a Perseus6113 to house two stacks of two MI125 (4 total). The MI125 takes full advantage of the LTM9012 integrated low-noise amplifiers which are suitable for single-ended drive and pulse train signals such as required for imaging applications. Combined with multiple clocks and trigger modes, the MI125 is at its best in DSP applications such as medical/industrial imaging (PET/ultrasound systems), multichannel DAQ, nondestructive testing, radar beamformers, phased array antennas and multichannel pulse detectors (linear accelerators, synchrotron).

For details on the MI125 FMC operation and its specifications, please refer to the *MI125 User's Guide.pdf* and the *MI125 Programmer's Reference Guide.pdf* documents available in the doc/fmc/MI125 folder of the PicoDigitizer125 Software tools USB dongle.

2.1.5 Embedded SSD (optional)

The optional embedded SSD is a Samsung 250GB SATA NAND Flash solid state drive, supporting SATA3. It comes in the 2.5 inch form factor. The SSD is only present in the PicoDigitizer125-64-E, which contains an AMC726 embedded computer. It offers the following sequential read and write performance as reported in the Samsung spec sheet for the product.

Direction	Performance
Sequential Read	540 MB/s
Sequential Write	520 MB/s

Table 4 Embedded SSD performances

3 PicoDigitizer125 Versions

The PicoDigitizer125 comes in two different versions.

- **PicoDigitizer125-64**
- **PicoDigitizer125-64-E**

The following sections fully detail the hardware of each PicoDigitizer125-64 version.

3.1 PicoDigitizer125-64

The PicoDigitizer125-64 combines the following hardware:

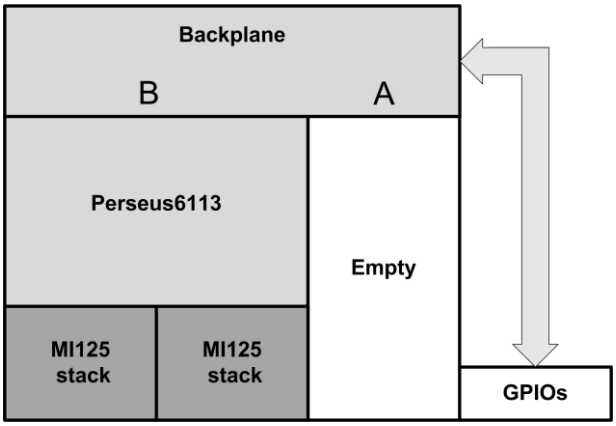

Figure 3-1 PicoDigitizer125-64 component diagram

- In slot A
  - Slot A is empty
- In slot B
  - 1 Perseus6113
  - 2 MI125 stacks (32 RX channels per stack, total of 64)
- 8 GPIO signals
- 1 Pico Extended Backplane using the following configuration:
  - External PCI Express: AMC ports 4-7 are routed by default from the Perseus6113 in slot B to the external PCIe connector.
  - Both PicoDigitizer125 Ethernet ports are connected to the Perseus6113 through the Ethernet switch.
  - JTAG is routed directly to slot B for use with the Perseus6113 FPGA.
  - The Perseus6113 serial console is available on the PicoDigitizer125-64 Mini-USB connector.
  - The external eSATA port and the internal SATA port are unused.

3.1.1 PicoDigitizer125-64 Front Panel

This section presents a description of the PicoDigitizer125-64 front panel.

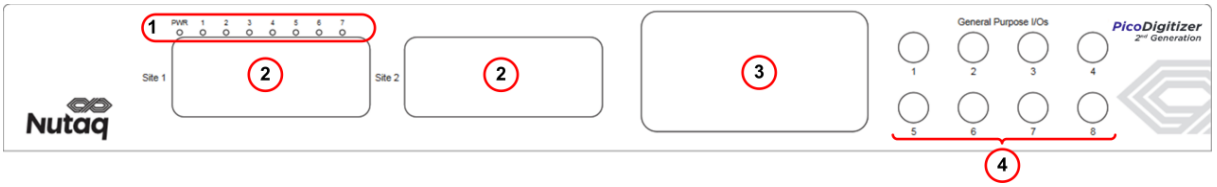

Figure 3-2 PicoDigitizer125-64 front panel

1- Front panel LEDs

8 front-panel LEDs are available for user control. The Power LED is solid green when the sub-system FPGA has been powered.

2- FMC sites hosting two MI125 FMC stacks

Sites 1 and 2 of the PicoDigitizer125-64 each host a MI125 FMC stack with 32 input channels. Details about the connections are presented here.

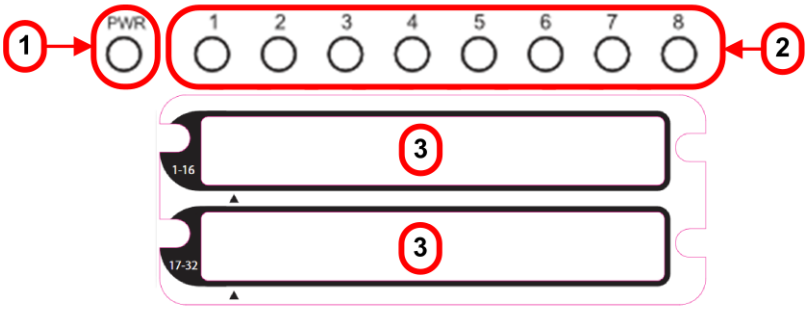

Figure 3-3 PicoDigitizer125-64 MI125 connections

1- Perseus6113 power LED

The power LED is green when the Perseus6113 is turned on.

2- Perseus6113 user LEDs

8 bicolor LEDs are available to the user from the FPGA logic.

3- MI125 Custom Connector

The front panel of the MI125 is equipped with a single connector from Samtec's ERF8 Series. The exact part number of the connector is ERF8-049-01-L-D-RA-L. You can learn more about the connector on Samtec's Web site by following this link:

http://www.samtec.com/ProductInformation/TechnicalSpecifications/Overview.aspx?series=ERF8

The connector has the following signals available (which can be broken out with the provided custom cable):

External clock (EC)

The external clock input connector, EC on the custom breakout cable of the MI125 and has an impedance of 50 Ω. The input signal is LVCMOS 3.3 V and is AC coupled.

External trigger (ET)

The external trigger pin of the front panel connector can be used either as an input or an output. The trigger must be used as a 2.5-V signal.

Channels 1 to 16 (single-ended configuration)

Signals 1 to 16 are connected to the analog signal inputs of the MI125 and have an input impedance of 50 Ω. The characteristics of these signals should respect the configuration of your MI125. With the default configuration, their amplitude should not exceed 2 Vpp.

Nutaq offers a custom cable with its PicoDigitizer125 products. Please contact us for more details about this solution.

Figure 3-4 MI125 custom breakout cable

The figure below presents the pinout of the front panel connector. The first pinout represents the pinout for a MI125 of a PicoDigitizer125 with a single-ended configuration and the second one represents the pinout for a differential configuration.

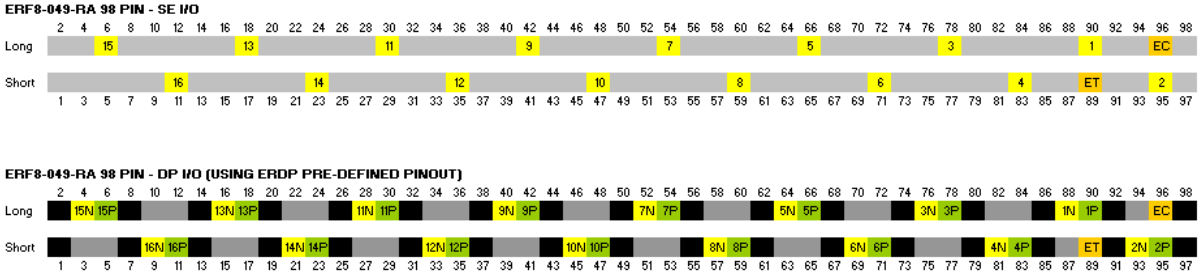

Figure 3-5 MI125 front panel connector pinout

3- Blank CPU membrane

A blank membrane covers the cutout that's reserved for the embedded computer.

4- GPIOs

8 GPIOs are available on the current PicoDigitizer125 product.

3.2 PicoDigitizer125-64-E

The PicoDigitizer125-64-E combines the following hardware:

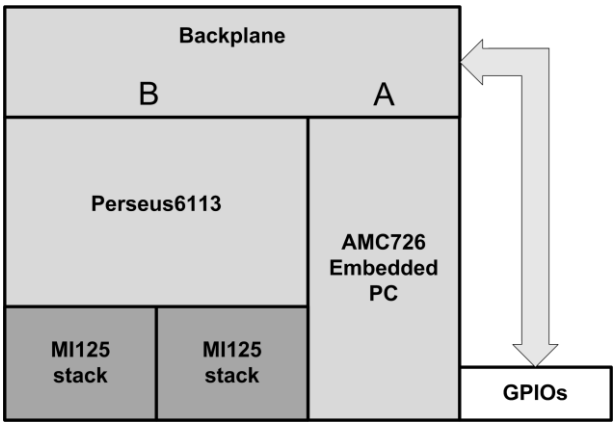

Figure 3-6 PicoSDR 8x8-E component diagram

- In slot A
  - 1 AMC726 embedded PC.
- In slot B
  - 1 Perseus6113
  - 2 MI125 stack (32 RX channels per stack, total of 64)
- 8 GPIO signals
- 1 Pico Extended Backplane using the following configuration:
  - External PCI Express: A PCI Express link with the Perseus6113 in slot B is possible either using the external PCIe connector or internally with the AMC726 embedded PC. This configuration is available from a backplane control menu. Please contact support@nutaq.com for information on how to proceed.
  - Both PicoDigitizer125 Ethernet ports are connected to the Perseus6113 and the Embedded PC through the Ethernet switch.
  - JTAG is routed directly to slot B for use with the Perseus6113 FPGA.
  - The Perseus6113 serial console is available on the PicoDigitizer125-64-E Mini-USB connector.
  - The external eSATA port is present and directly connected to the Embedded PC.
  - The internal SATA port with 250GB SSD is available as an option.

3.2.1 PicoDigitizer125-64-E Front Panel

This section presents a description of the PicoDigitizer125-64-E front panel.

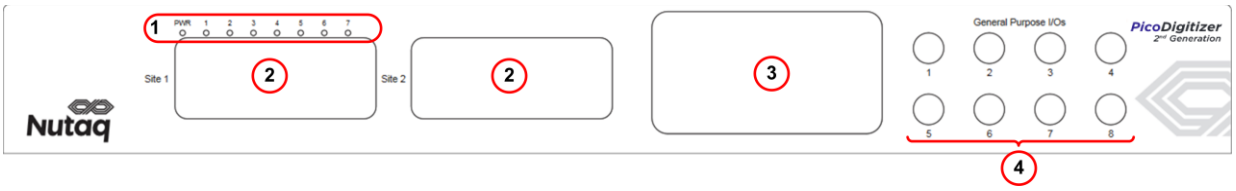

Figure 3-7 PicoDigitizer125-64-E front panel

1- Front panel LEDs

8 front-panel LEDs are available for user control. The Power LED is solid green when the sub-system FPGA has been powered.

2- FMC sites hosting two MI125 FMC stacks

Sites 1 and 2 of the PicoDigitizer125-64 host each a MI125 FMC stack with 32 input channels. Details about the connections are presented here.

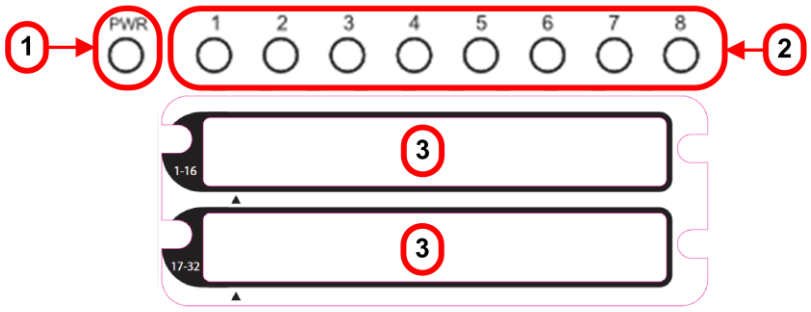

Figure 3-8 PicoDigitizer125-64-E MI125 connections

1- Perseus6113 power LED

The power LED is green when the Perseus6113 is turned on.

2- Perseus6113 user LEDs

8 bicolor LEDs are available to the user from the FPGA logic.

3- MI125 Custom Connector

The front panel of the MI125 is equipped with a single connector from Samtec's ERF8 Series. The exact part number of the connector is ERF8-049-01-L-D-RA-L. You can learn more about the connector on Samtec's Web site by following this link:

http://www.samtec.com/ProductInformation/TechnicalSpecifications/Overview.aspx?series=ERF8

The connector is described in the previous section.

3- ACM726 embedded processor

Figure 3-9 PicoSDR 8x8 AMC726 connections

1- Ethernet connectors

The AMC726 offers one RJ-45 Ethernet connector on its front panel. This connector is typically used to connect the embedded computer to the network.

2- Mini DisplayPort connector for video monitor

3- USB type C connectors for peripherals

4- Micro-USB connectors for PCH and IPMI debug

4- GPIOs

8 GPIOs are available on the current PicoDigitizer125 product.

4 PicoDigitizer125-64 Software

The following image illustrates the PicoDigitizer125-64 software stack.

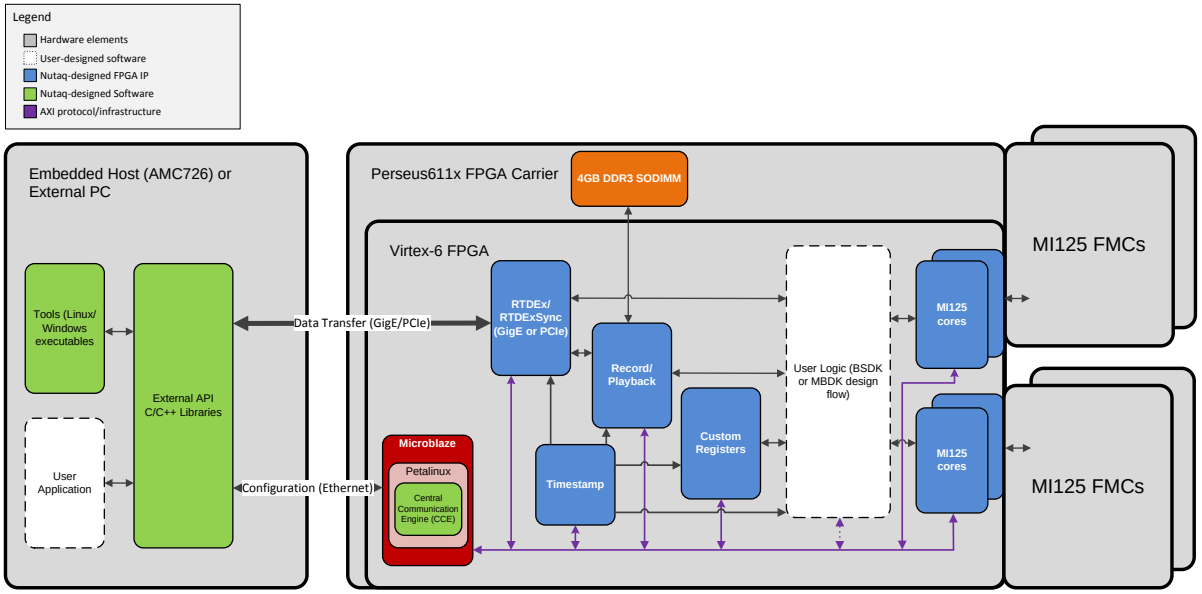

Figure 4-1. PicoDigitizer125 software stack

4.1 Host Software Tools

The PicoDigitizer125 2nd Generation Software Tools support the following host development environments:

- Ubuntu 12.04 LTS, either pre-installed on the PicoDigitizer125-64-E embedded computer or a PC.
- Windows 7 on a PC.

The host software tools relevant to the PicoDigitizer125 are:

4.1.1 Host Libraries

- **External API (EAPI):** The External API is the host library which instantiates the control interface between the PC and the FPGA. The following parts of the EAPI library are used in the PicoDigitizer:
  - **MI125 EAPI:** The MI125 portion of the EAPI is used to configure the ADC front-end of the PicoDigitizer125-64. The library ensures the communication between the host computer and the front-end hardware through an Ethernet link with the PicoDigitizer's FPGA.
  - **Custom Registers EAPI:** The custom registers functions are used to execute generic reads and writes in the FPGA, for example to reset and configure logic blocks.
- **Data recording library (Record/Playback)** The Record/Playback library configures and operates the FPGA's record and playback core and allows recording of raw data onto the onboard memory.
- **Data streaming library (RTDEx):** The RTDEx library configures and operates the RTDEx FPGA core and executes real-time data transfers between the host and the FPGA on the Gigabit Ethernet or PCI Express medium.

4.1.2 Sample Applications

The Software Tools showcase the use of the cores and libraries through its sample applications. The following applications are the main ones used in the PicoDigitizer125-64 Reference design. The applications are modifiable and recompilable by the user.

MI125

- MI125_Init: Handles the configuration of the PicoDigitizer125-64 ADC front-end.

Data recording and playback

- RecordData: Records data to the FPGA DDR3 from an interface of the user's choosing (typically the FMCs DAC channels).
- RetrieveRecordedData: Reads data from the FPGA DDR3 and stores it in a file on the host PC.
- LoadDataToPlayback: Writes data to the FPGA DDR3 from a file on the host PC.
- PlaybackData: Configures and executes a data playback from the FPGA DDR3.
- PlaybackStop: Stops a current data playback

Data streaming

- RXStreaming: Streams real-time data from the FPGA (and its front end) to a file on the host PC.
- TXStreaming: Streams real-time data to the FPGA (and its front end) from a file on the host PC.
- RxSyncStreaming: Streams synchronous real-time data from the FPGA (and its front end) to a file on the host PC.
- TxSyncStreaming: Streams synchronous real-time data to the FPGA (and its front end) from a file on the host PC.
- RxTsStreaming: Streams synchronous timestamped real-time data from the FPGA (and its front end) to a file on the host PC.
- TxTsStreaming: Streams synchronous timestamped real-time data to the FPGA (and its front end) from a file on the host PC.

Generic applications

- CustomRegister_Write: Executes a 32-bit wide data write to a FPGA register.
- Timestamp_Util: Configures the timestamp FPGA core and retrieves stamp values.

The applications are in the *%BASROOT%/tools/apps* folder.

4.1.3 The Command Line Interface (CLI)

This feature is not supported for host development on the PicoDigitizer125-64 systems. For maintenance of the Perseus6113 carrier board such as flashing FPGA bit files, changing the carrier's IP address and updating the CCE, please refer to the documents App Note - Configuring the Perseus IP address.pdf and App Note - Perseus Firmware Update.pdf located in the %BASROOT%\doc\app_notes directory.

4.2 Embedded Firmware

4.2.1 The Central Command Engine (CCE)

When using Nutaq's BSDK or MBDK toolsets to create the FPGA implementation, an embedded MicroBlaze processor is instantiated in the FPGA logic. This processor runs an embedded Linux distribution which interfaces the processor's volatile and non-volatile memories as well as the network connectivity.

Within this distribution is deployed the Central Command Engine application. The CCE handles the direct control of the FPGA implementation and the FMC hardware through the FPGA's internal AXI bus and the I2C or SPI buses on the hardware.

4.3 FPGA Tools

The Software Tools Windows release includes support for all Nutaq IP cores in Xilinx ISE 14.7 and System Generator in Nutaq's Model Based Design Kit design flow.

The main IP cores used in the system are:

- **Front end core (MI125):** Handles the front end's initialization and data acquisition.
- **Data recording core (Record/Playback):** Controls the DDR3 memory interface and allows reads and writes.
- **Data streaming cores (RTDEx PCI Express, RTDEx Gigabit Ethernet and RTDEx Sync):** Used to stream data at high-speed between FPGA and the host PC.
- **Timestamp:** Maintains an internal time base for timestamped events.

In the FPGA is an amount of logic cells and resources which is reserved to the user, the **User Logic.** In this space, the PicoDigitizer's user can implement their signal processing algorithm as well as interface to the BSDK and MBDK FPGA cores.

BSDK FPGA tools and cores are located in the *%BASROOT%/sdk/fpga/NutaqIPLib/pcores* folder of the Windows installation while the MBDK tools are located in the *%BASROOT%*/sdk/fpga/mbdk/blocks

5 Specifications

This chapter presents the main technical specifications of the PicoDigitizer125.

Note:
The specifications presented here are subject to change without notice.

5.1 Mechanical Specifications

This section provides the dimensions and mass specifications.

5.1.1 Dimensions

PicoDigitizer125 product	Width (mm)	Height (mm)	Depth (mm)
PicoDigitizer125-64	365.00	45.00	378 with connectors 397 without connectors
PicoDigitizer125-64-E	365.00	45.00	378 with connectors 397 without connectors

Table 5 PicoDigitizer125 dimensions

5.1.2 Mass

PicoDigitizer125 product	Mass (lbs)	Mass (kg)
PicoDigitizer125-64	11.0	5.0
PicoDigitizer125-64-E	12.4	5.6

Table 6 PicoSDR125 mass

5.2 Power Consumption Specifications

The following table lists the power consumption of the PicoDigitizer. The minimum power consumption was measured while the PicoDigitizer was turned off. The typical power consumption was measured while the PicoDigitizer was turned on and the FPGA was programmed. The maximum power consumption was measured while recording data on all channels.

Specification	Typical	Maximum
PicoDigitizer125-64	**67 watts**	**141 watts**
PicoDigitizer125-64-E	**92 watts**	**166 watts**

Table 7 PicoDigitizer125 power consumption

5.3 Temperature Specifications

The PicoDigitizer equipment is designed for lab use only. The PicoDigtizer125 should always operate in an ambient temperature setting between 15°C and 25°C. Operating the unit outside of this range can severely damage the hardware and is not supported within the PicoDigtizer125 warranty.

5.4 Analog Specifications

For the full PicoDigtizer125 analog specifications please refer to the **MI125 User's Guide.pdf** document available in the doc/fmc/MI125 folder of the PicoDigtizer125 Software tools USB dongle.

6 PicoDigitizer125 Setup

This chapter presents the contents of the product shipment and the procedures to set up the instrument.

6.1 Shipment Contents

The PicoDigitizer125 shipping box contains:

- **The PicoDigitizer125 2nd Generation enclosure**
- **A universal power supply with a power cord**.
  - Maximum output power: 150 Watts
  - Output voltage: 12 VDC
  - Input voltage: 100 to 240 VAC
  - Frequency: 50 to 60 Hz
  - Operation temperature: 0 to 40°C
- **The PicoDigitizer125 2nd Generation Software tools USB dongle**
  - Windows 7 Installer (Host and FPGA development).
  - Ubuntu 12.04 LTS Installer (Host development).
- **The Digitizer125 2nd Generation Quick Start Guide**
- **A USB mini-B to A male-male cable**
- **An Ethernet cable**
- **Other documents** (Terms of use and product policies)
- **MI125 custom cable (if purchased)**
- **Additional content for PicoDigitizer125-64-E:**
  - **Mini DisplayPort to HDMI/VGA adapter**
  - **USB-C to 4 x USB 3.0 adapter and USB type C to USB type A male-male cable**

6.2 Software Tools Installation

The full Installation procedure of the PicoDigitizer125 Software Tools is on the Nutaq USB dongle delivered with the PicoDigitizer125 system, in the documentation folder.

6.3 PicoDigitizer125 Serial Console Menu

The PicoDigitizer125 RS-232 menu is available through the back panel USB connector. On a Linux computer, four RS232 ports appear upon the USB connection. For a Windows computer, the user sees four virtual com ports created. The configuration of all RS-232 ports is the same (115200,N,8,1).

6.3.1 Linux Console Menu

On Linux, to view the port list, execute the following command: #ls /dev/ | grep ttyUSB

The typical port list is:

Port name	Device	Connection
ttyUSB0	USB serial converter A	Reserved for future use
ttyUSB1	USB serial converter B	Perseus6113 RS-232 communication
ttyUSB2	USB serial converter C	Backplane RS-232 communication
ttyUSB3	USB serial converter D	Reserved for future use

Table 8 PicoDigitizer125 typical com ports list for Linux

6.3.2 Virtual COM Port Detector Tool

A tool called ComPortsDetector is available in the PicoDigitizer125 software installation package to detect and list the currently used Virtual COM ports. It can be found in the *%BASROOT%/tools/support/script* directory.

The capture below shows the result of executing the ComPortsDetector tool on a PicoDigitizer125. The *Slot* B entry indicates the Perseus COM port for the PicoDigitizer125 and the *Backplane* entry indicates the PicoDigitizer125 backplane COM port.

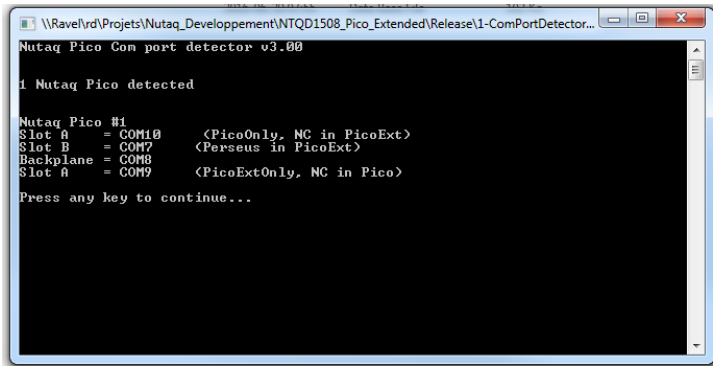

Figure 6-1 Virtual COM port detector expected results

6.3.3 Windows Console Menu

On a Windows computer, 4 virtual com ports will be added upon the PicoDigitizer125 USB connection. Unfortunately, Windows will not always create the virtual com ports in the same order as at the first detection. The user should check in the **Device Manager** window to open the **Properties** dialog box to verify which port is connected to what device. Usually the location (A to D) will be indicated on the **General** tab.

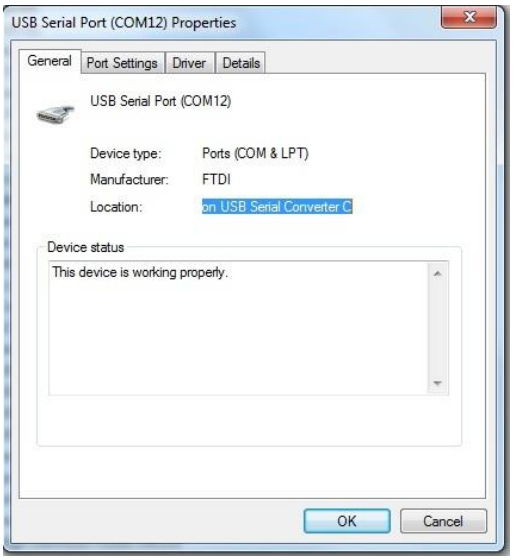

Figure 6-2 Windows virtual COM port general properties

However, for certain setups the location information always lists "0". In such a case, the user must select the **Details** tab and, in the **Property** list, select **Device Instance Path** to display which com port is connected to which device under **Value**. The number indicated by the arrow in Figure 6-3 identifies the device (1-2-3 link to A-B-C).

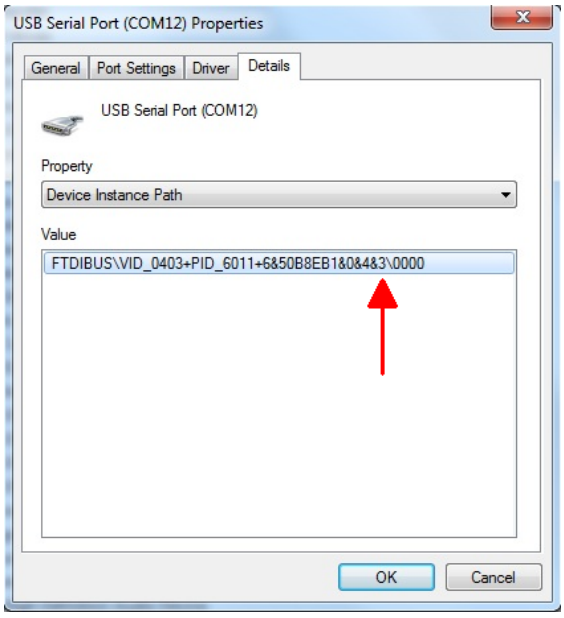

Figure 6-3 Virtual COM port link to PicoSDR hardware

Device	Connection
USB serial converter A	AMC slot A RS-232 communication
USB serial converter B	AMC slot B RS-232 communication
USB serial converter C	Backplane RS-232 communication

Table 9 PicoDigitizer125 com ports list for Windows

6.3.4 Menu Description

The following is a representation of the menu of the PicoDigitizer125-64-E.

PicoDigitizer125-64-E
Serial Number : NTQ0127xxx

Hardware Revision : RevA
Firmware Version : 1.05

A - Power cycle CPU
B - Power cycle Perseus
C - Power cycle System
D - Shutdown (menu will still be available)
S - Toggle Autostart mode (Status : ON)

+ - Increase the fan velocity
- - Decrease the fan velocity
Fan velocity set to 70%
Minimum fan velocity is 70%

Choice :

The first group of 4 lines indicates some basic information on the content of the PicoDigitizer125 such as the product itself (-64 or -64-E). The serial number of the PicoDigitizer and the hardware revision and firmware version of the backplane board are also listed.

The user is then prompted to choose between the options listed by entering the choice in either upper or lower case on the console command line. Options A, B, and C allow the user to start a power cycle for each slot or the whole system. Option D turns off the PicoDigitizer125, turning off both slots. Even when turned off, the menu is still available. Pressing the space bar in the menu or clicking the back panel push button will start the PicoDigitizer125.

The autostart mode provides the ability to start the PicoDigitizer125 automatically upon power connection. It is useful when one wants to use the PicoDigitizer125 and control it remotely via an Ethernet link. This way, in case of a power failure, it will always restart automatically when the power returns.

If the External PCIe option is available on the PicoDigitizer125, the console will print an extra line in the menu. This option can be used to toggle between the PCIe clock that comes from the external cable and the one generated on the backplane. The current clock source use will be shown on the menu to inform the user.

E - Toggle PCIe clock source (Status : CABLE PCIE CLOCK)

On the PicoDigitizer125-64 system, the PCIe clock source is set to ‘external cable’ and cannot be changed.

On the PicoDigitizer125-64-E system, the PCIe clock source is set to ‘backplane’ and cannot be changed directly by the user. Please contact Nutaq Support (support@nutaq.com) if you wish to use external PCIe with a PicoDigitizer125-64-E system.

7 FPGA Demos Description and Generation

7.1 PicoDigitizer125-64 Demos Description

A set of FPGA designs showcasing how to use the MI125 and its corresponding FPGA cores are made available in the Software Tools installation. These designs also show how to interface user made logic to the FPGA core of the MI125. They reside in the *bas**examples**mi125* directory. They are designed using either Nutaq's Model Based Design Kit (MBDK) or Xilinx Platform Studio directly (BSDK).

Project files path	Design methodology	RTDEx media
perseus611x\bsdk_gige	Xilinx Platform Studio (BSDK)	Gigabit Ethernet
perseus611x\mbdk	Nutaq's Model Based Design Kit (MBDK)	Gigabit Ethernet

Table 10 Available FPGA designs depiction

BSDK and MBDK FPGA designs are functionally identical to each other. The inner workings of the example design can be illustrated with

Figure 7-1. The diagram is aimed at showing the flow of data within the FPGA design and omits some control signals and key components such as the Microblaze processor, the AXI bus, the carrier core, etc. Note also that the bus widths expressed in this diagram refer to the conceptual widths, from a high-level point of view. The relationship with HDL bus widths is not one-to-one. For example, a line/arrow in diagram of

Figure 7-1 may actually be implemented as three separate buses in HDL code. Also, a bus having a width of 24 bits in the diagram may actually be wider in the actual implementation to include, for example, "Data Valid" signals.

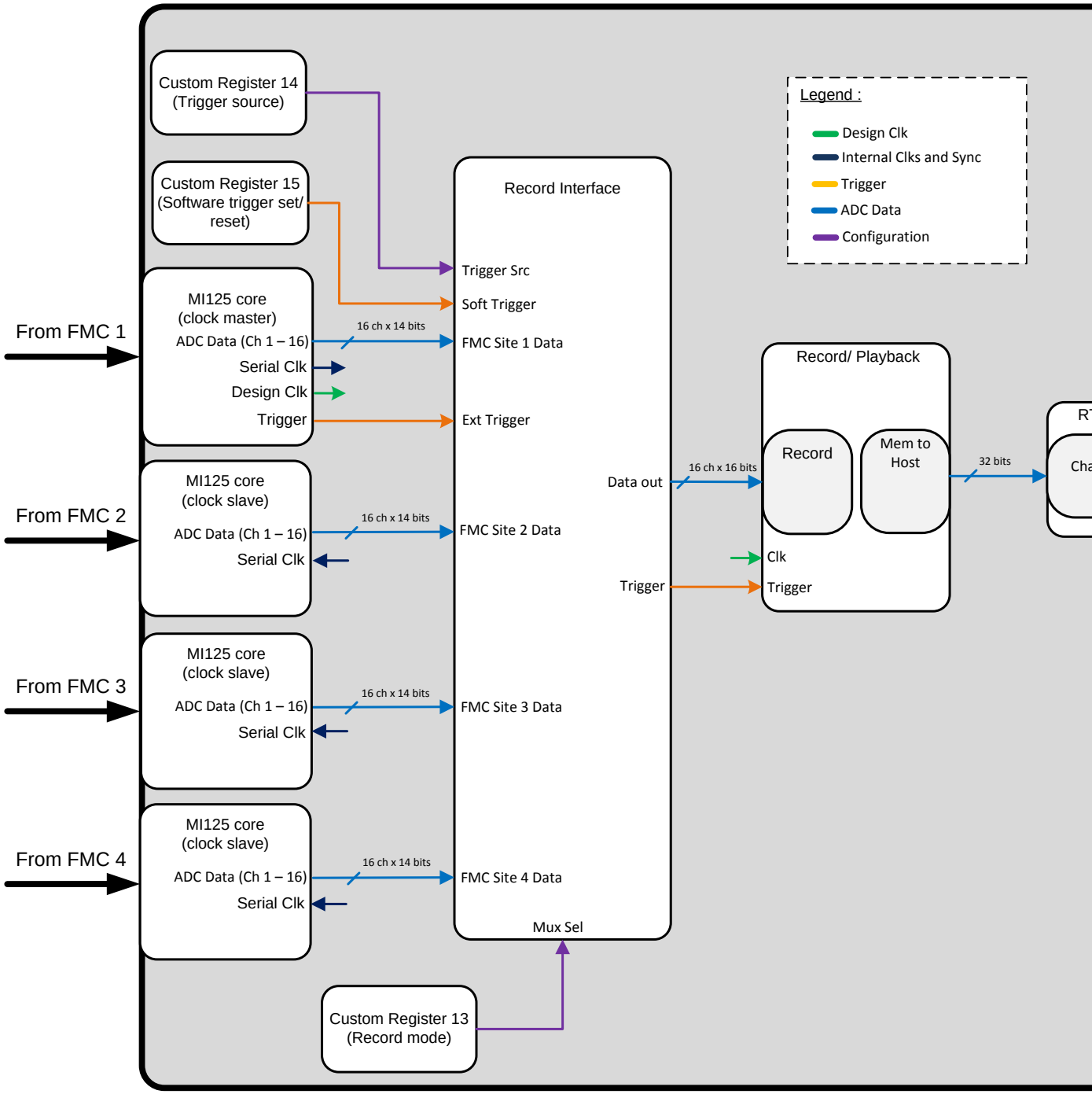

Figure 7-1 MI125 Perseus611x example diagram

The MI125 example illustrates the interactions between the MI125, the Record/Playback core and the RTDEx core using different modes of operation. In this demo, no processing is implemented between the ADCs and the Record/Playback core. Data is simply transmitted from the ADCs to the onboard DDR3 memory in real time. The host may then retrieve the data after acquisition.

64 channels, each with 14 bits of precision, come out of the MI125 cores. The channels are then fed through the Record interface core before being written to the onboard memory. As the total throughput of 64 channels at 125 Msps largely exceeds the maximum write speed on the DDR3 memory, multiplexing or downsampling is needed.

This is accomplished thanks to the block named “Record Interface” in Figure 7-1. In this regard, 7 modes of operation are possible. Table 11 lists them all.

Recorded Data (FMC sites)	Total number of channels	Downsampling	Custom Register 13 value
1	16	1 (none)	0
2	16	1 (none)	1
3	16	1 (none)	2
4	16	1 (none)	3
1, 2	32	2	4
3, 4	32	2	5
1, 2, 3, 4	64	4	6

Table 11. Record Interface Core Modes

The Record Interface block is also responsible for the following duties:

- Converting the 14-bit buses to 16-bit buses by sign-extending the samples.
- Multiplexing the trigger signals (software and front-panel) and feeding to the Record/Playback core.

7.2 Generating the Demos

MBDK versions of the FPGA designs described in section 7.1 come pre-compiled with the software suite. They are available in the *%BASROOT%\tools\bitstreams\PicoDigitizer125* folder.

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

Sections 7.2.1 and 7.2.2 explain how to open the projects and regenerate them.

7.2.1 BSDK

1. Open the Xilinx Platform Studio 14.7 project file in the *%BASROOT%\examples\mi125\perseus611x\bsdk_gige* folder.
2. Explore the contents 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. In the case of the PicoDigitizer125-64, the device should be ‘xc6vsx475t’.
5. Click **OK**.
6. In the **Device Configuration** menu, click **Update Bitstream**.

   Platform Studio will generate a bitstream (*.bit* file) and attach *perseus_default_linux.elf* from the *bas\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.

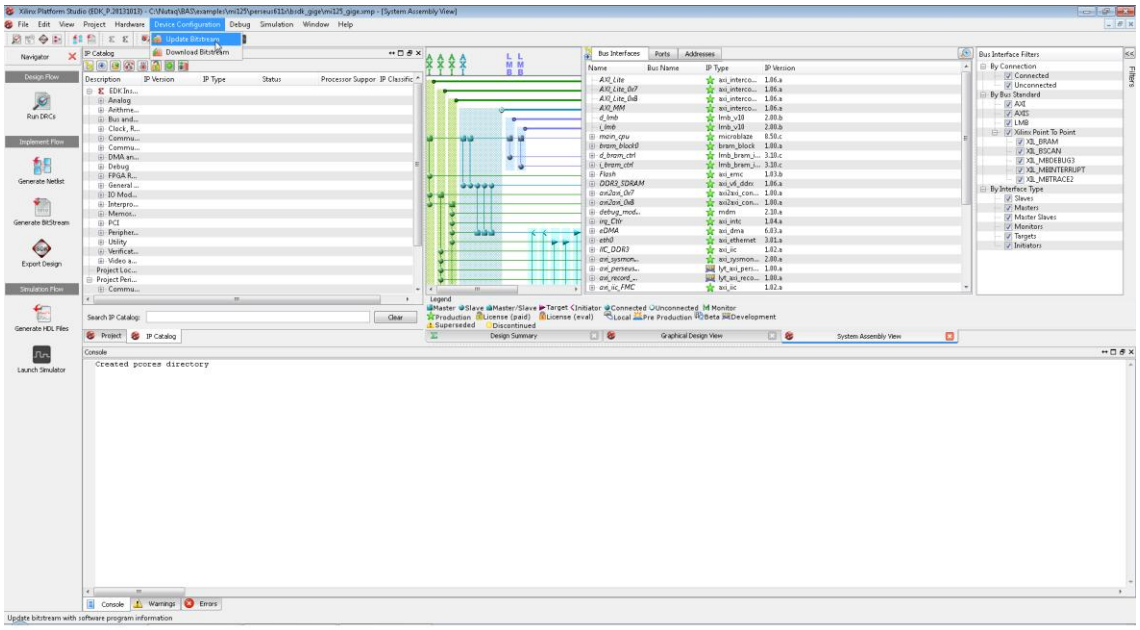

Figure 7-2 MI125 BSDK example generation

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 9 explains how to use this file and how to run the demo.

7.2.2 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\perseus611x\mbdk_gige* folder. The project file is a *.slx* file.
3. Explore the contents 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. In the case of the PicoDigitizer125-64, the device should be 'Virtex6 xc6vsx475t-1ff1759'.
6. Click **Apply**.
7. Click **Generate**

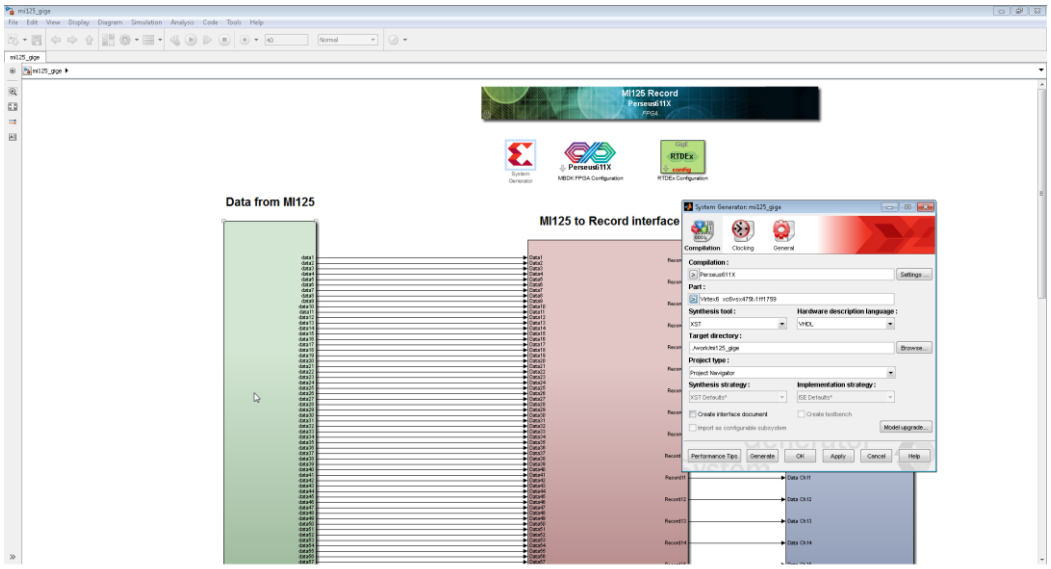

Figure 7-3 MI125 MBDK example generation

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* file becomes available directly in the given project repository. Section 3 explains how to use this file and how to run the demo.

8 Preparing to Execute the Demos

8.1 Hardware Requirements

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

PicoDigitizer125-64 with external PC

- 1 PicoDigitizer125-64 chassis
- 1 PC with a Gigabit Ethernet capable network card (jumbo frame capable)
- 1 Gigabit Ethernet cable
- ERM8 cables to connect the MI125s to the signal generator
- 1 Signal Generator able to output a signal of frequency lower than 62.5 MHz
- 1 125 MHz external clock source

PicoDigitizer-64-E (with embedded PC)

- 1 PicoDigitizer125-64-E chassis
- ERM8 cables to connect the MI125s to the signal generator
- 1 Signal Generator able to output a signal of frequency lower than 62.5 MHz
- 1 125 MHz external clock source

8.2 Hardware Setup

8.2.1 Setup for Execution on a PicoDigitizer125-64 with an External PC

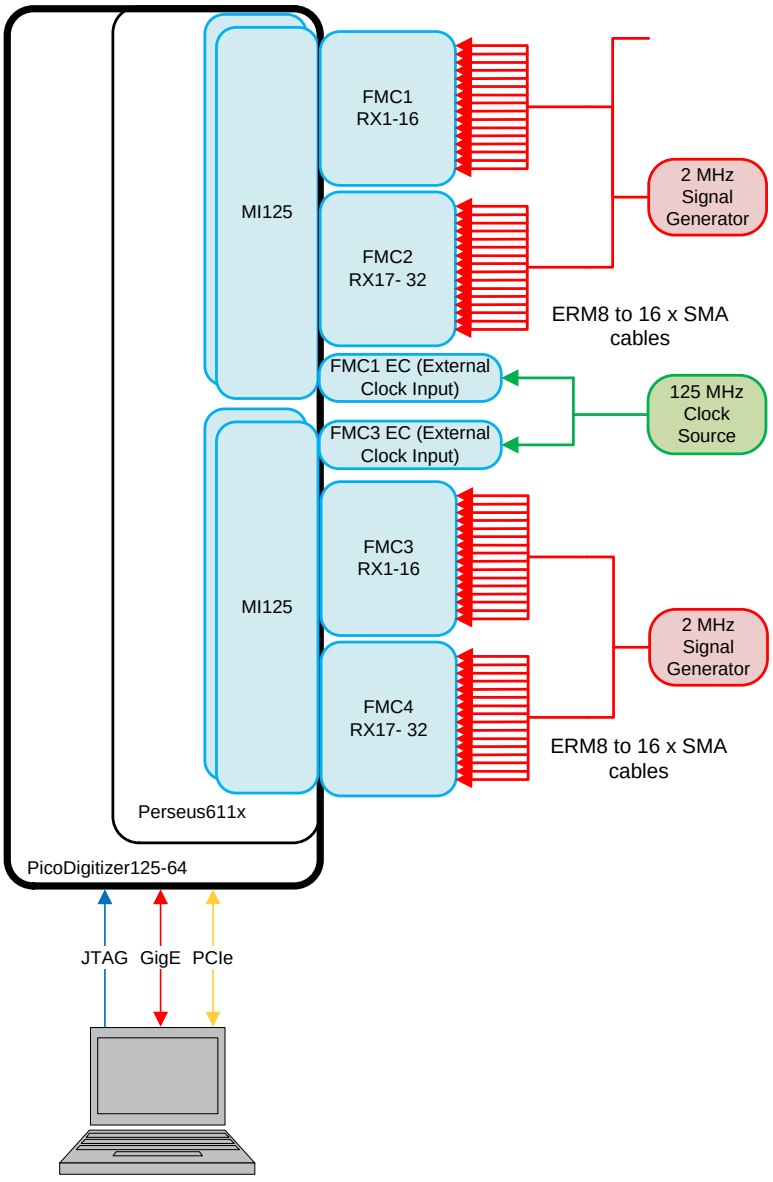

Figure 8-1 Setup for execution on a PicoDigitizer125-64 with an external PC

The following procedure must be performed to execute the examples on a PicoDigitizer125-64 system with an external PC.

To setup the example:

1. Connect the FPGA JTAG pod to the external PC and then to the PicoDigitizer125-64 back panel.
   Refer to section 9.1 for details about performing this operation.
2. Connect the Gigabit Ethernet cable between the external PC and the PicoDigitizer125-64 back panel.
3. Optionally, connect a USB cable between the external PC and the PicoDigitizer125-64 back panel (using the system mini-USB connector).

8.2.2 Setup for Execution on a PicoDigitizer125-64-E

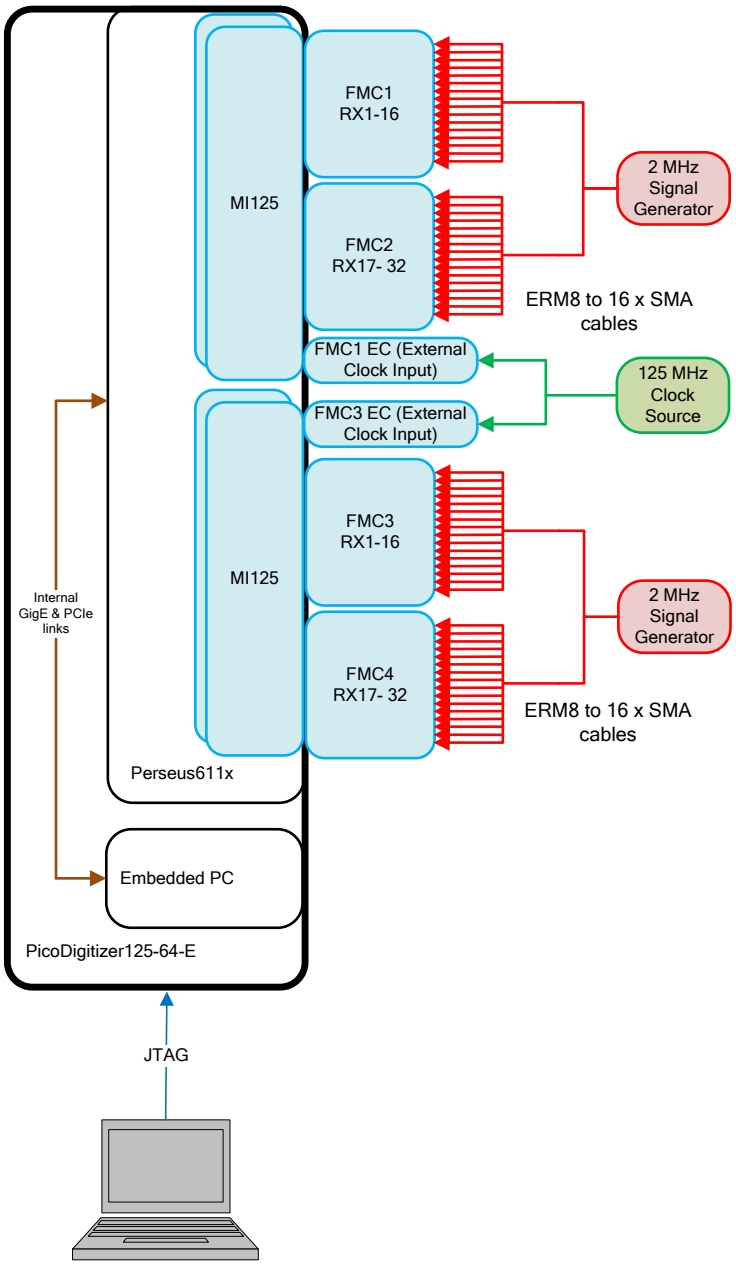

Figure 8-2 Setup for execution on a PicoDigitizer125-64-E

The following procedure must be performed to execute the examples on a PicoDigitizer125-64-E system with an Embedded PC.

To setup the example:

1. Connect the FPGA JTAG pod to the external PC and then the PicoDigitizer125-64-E back panel.
   Refer to section 9.1 for details about performing this operation.
2. Optionally, connect a USB cable between the external PC and the PicoDigitizer125-64-E back panel (using the system mini-USB connector).

8.2.3 Equipment Setup

To setup the example:

1. Connect the signal generator output to one or more MI125 channels using an the custom breakout cable. Any of the 64 channels of the PicoDigitizer125-64 channels may be connected. Set the power amplitude of the signal generator to 1-Vpp (4 dBm) and the frequency to 2 Mhz.

8.2.4 External Clocking Setup

If you purchased the clocking solution proposed by Nutaq, configuration files are provided in the Software Tools files to set it up quickly and enable the reference design to work properly.

Before validating the MI125 reference design, execute the reference clock module generator setup procedure.

1. Install the host software
    a. Download the evaluation software for the HMC7044 from Analog Device's website : http://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-hmc7044.html#eb-relatedsoftware
    b. Extract the .zip folder contents and run setup.exe. Leave all options to their default values.
    c. After the software setup, you will be prompted to install software drivers. Click Yes.
    d. The evaluation software for the HMC7044 is now installed.
2. Connect the device to the MI125 front-end
    a. Connect the provided power adapter to a power outlet and to the evaluation module.
    b. Connect the HMC7044 evaluation module to the host computer with the provided USB cable.
    c. Connect the module to a 10 MHz reference clock.
    d. Connect one custom breakout cable to Site 1, channels 1 - 16 (FMC1) and another one to Site 2, channels 1 – 16 (FMC3) of the PicoDigitizer64.
    e. Identify the "External Clock" (EC) ports on both connected breakout cables.
    f. Connect any of the module's enabled outputs (0, 3, 10, 13) to the "External Clock" (EC) port of each MI125 identified as MI125 "1-16" (one for each stack).
3. Configure the device for use with the reference design.

    Note
    The configuration has to be done every time the evaluation board is powered on.
    a. Launch the HMC704X GUI by going to Start Menu – All programs – HMC704X.
    b. Click on the Load configuration file button :

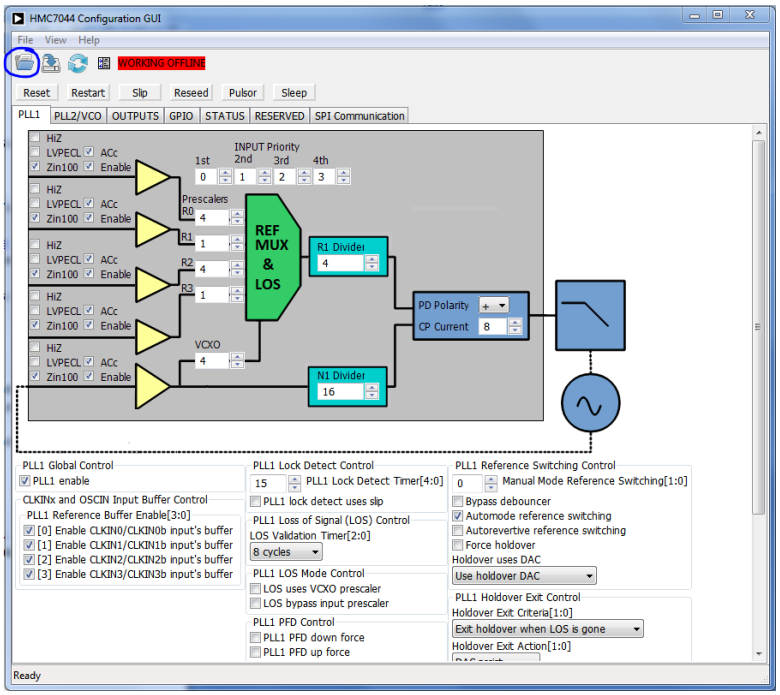

Figure 8-3 HMC7044 Configuration GUI Load Configuration

c. Select the provided configuration file. Use file %BASROOT%\examples\mi125\host\scripts\HMC7044_125MHZ_CMOS_0_3_10_13.py.

d. Click on the Restart button :

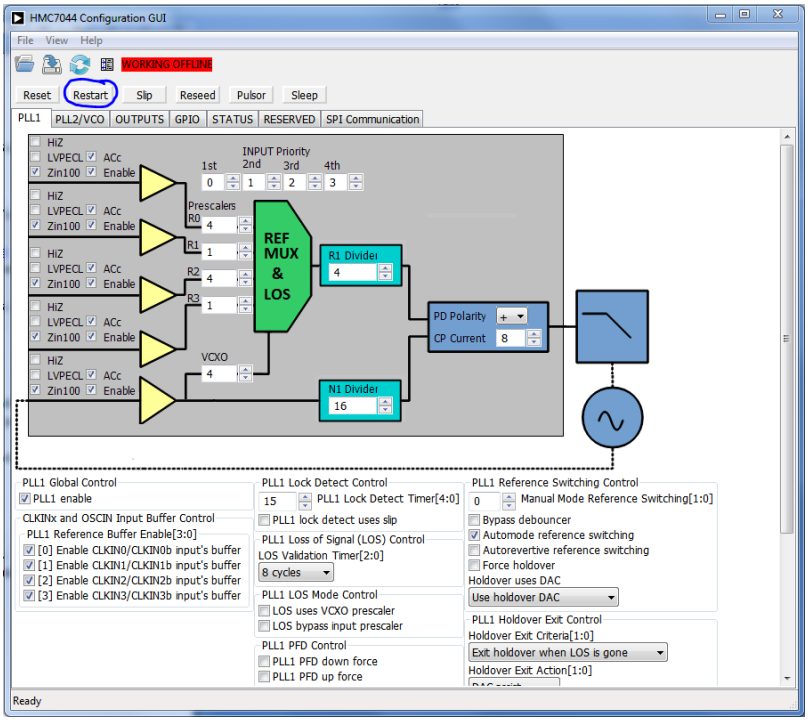

Figure 8-4 HMC7044 Configuration GUI Restart Button

e. Click on the STATUS tab :

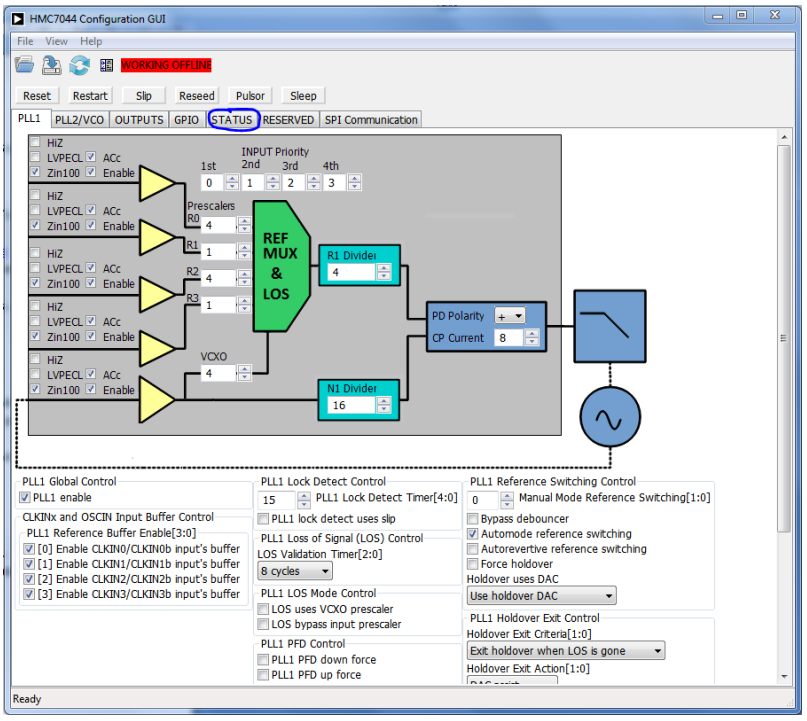

Figure 8-5 HMC7044 Configuration GUI Status Tab

f. Click on the Read Registers from device button :

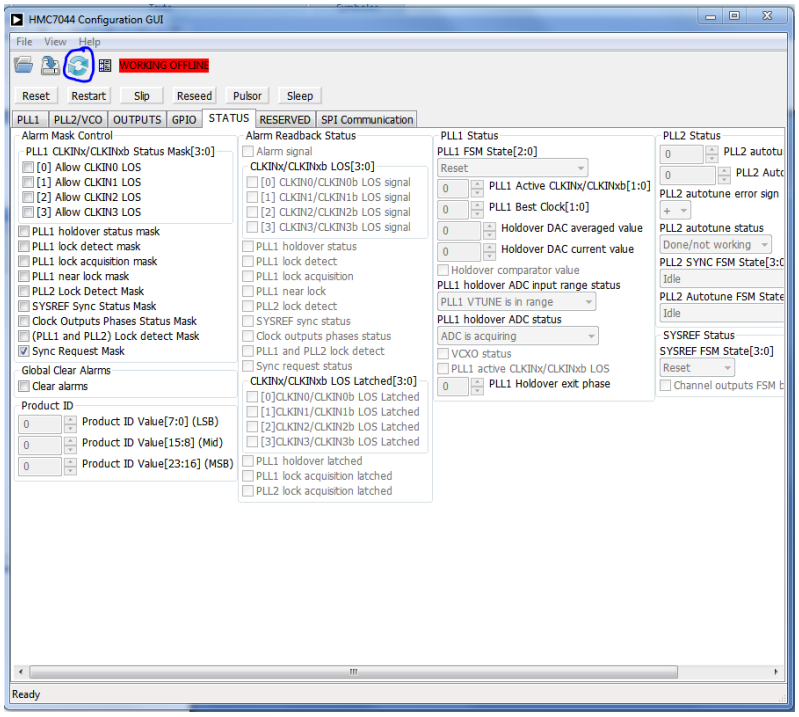

Figure 8-6 HMC7044 Configuration GUI Read registers device button

g. Validate that the onboard PLLs have locked with the PLL1 and PLL2 lock detect checkbox:

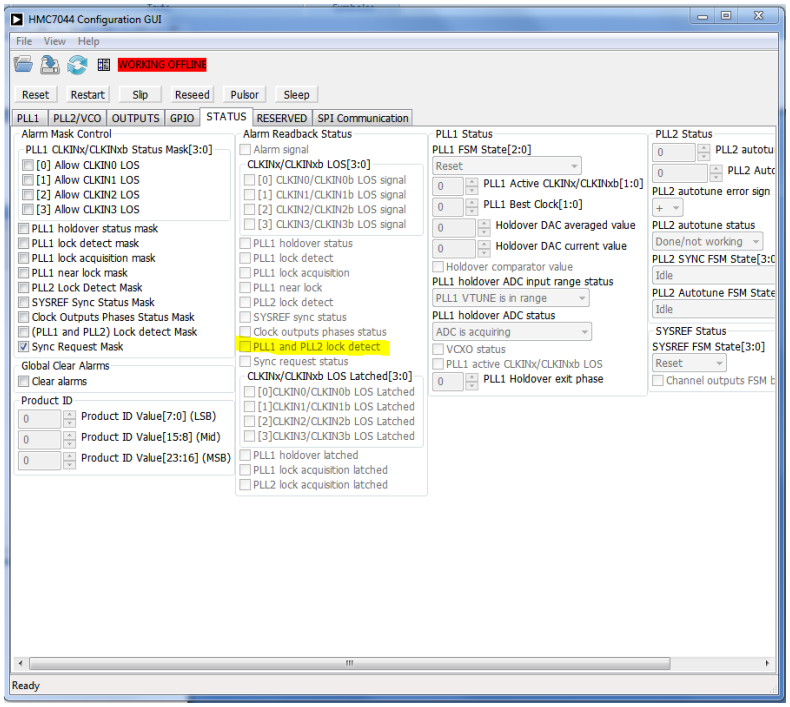

Figure 8-7 HMC7044 Configuration GUI PLL Lock

9 Executing the Demos

Two steps are needed to run the PicoDigitizer125-64 demo.

1. The FPGA is configured with a pre-compiled *mi125* bitstream, or one generated following steps depicted in section 7.2.

   Section 9.1 explains this operation.

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

   Section 9.2 explains this operation.

   Important:
   For applications requiring more than 1 MI125 stack to acquire synchronized data, ensure that the "External Clock" (EC) of the MI125 identified as MI125 "1-16" for each stack gets a copy of a phase-aligned version of the same clock signal. This is described in section 8.2.4.

9.1 Configuring the FPGA

1. Make sure your JTAG pod is properly connected to your system as described in section 8.2.
2. Open Xilinx iMPACT.
3. The software displays a prompt asking 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. On a PicoDigitizer64, iMPACT should detect a single xc6vsx475t device.

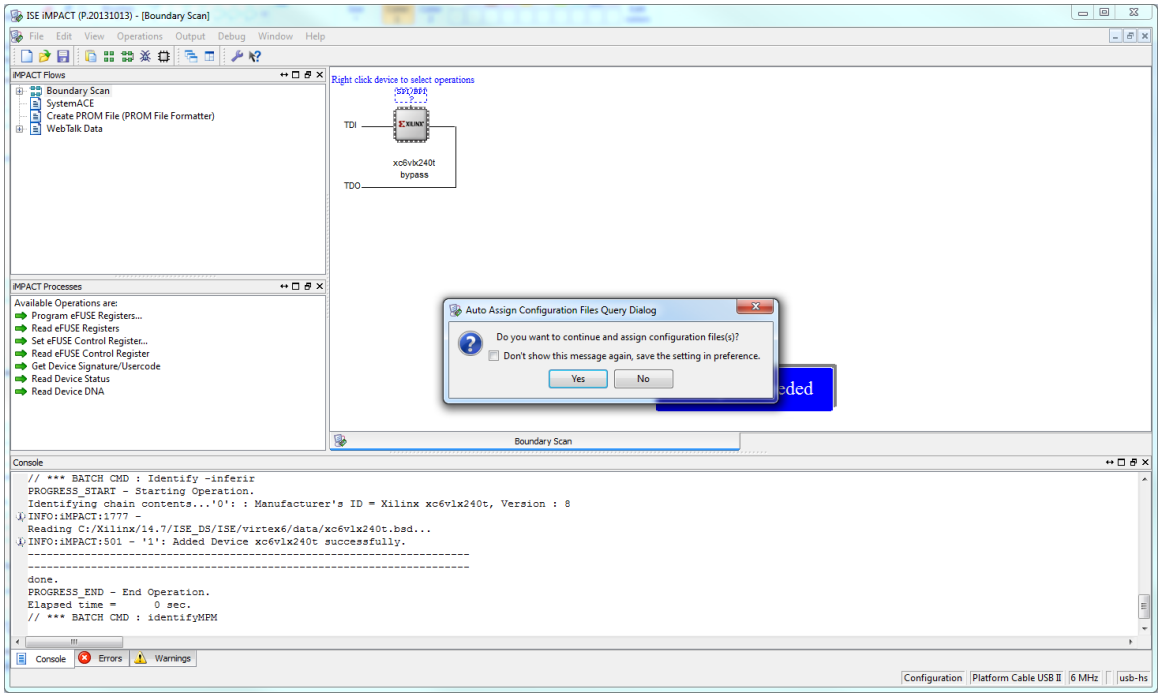

Figure 9-1 After iMPACT Boundary Scan

7. If the 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…**.

    Browse to the bitstream to download to the device and click **Open**. Section 7.2 explains how to generate this bitstream and where to find pre-compiled bitstreams shipped with your BAS software tools.

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

    - PicoDigitizer125-64 users must select their bitstream from the *bas**examples**mi125**perseus611x* directory.

    Downloading a bitstream designed for a carrier different from the carrier you are using may permanently damage the system.
11. A dialog prompting you to attach an SPI or BPI PROM to the device may appear. Click **No**.
12. In the JTAG chain, right click again on the device to program, and click **Program**.
13. If a Device Programming Properties dialog appears, click **OK**.
14. Impact is now programming the bitstream. This may take up to a minute depending on the FPGA type.
15. The bitstream is now programmed to the FPGA.

9.2 Host Scripts

Scripts showcasing MI125 operation and data acquisition are available and they can all be run using the bitstream configured in section 9.1. They reside in the *%BASROOT%\examples\mi125\host\scripts* directory and they are in the form of batch files (*.bat*) or shell scripts (*.sh*) for Windows or Linux, respectively.

9.2.1 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 the *Execution* portion of this section. 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\fmc\MI125
RecordData	Record/Playback	%BASROOT%\doc\cores\RTDEx_RecordPlayback
RetrieveRecordedData	Record/Playback	%BASROOT%\doc\cores\RTDEx_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. Inside the FPGA, as parameters for the MI125-Record interface core are routed out to custom registers of the core, application *CustomRegister_Write* (on register 13) is used to perform the operation.
- Configures the MI125-Record interface core 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*.

Execution

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

Windows

Double-click the MI125_Record_32.bat file.

The test starts automatically.

Linux

In a Linux terminal, change directory to *bas/examples/mi125/host/scripts*.

To start the example, run the following command

sudo ./MI125_Record_32.sh

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: 51.5C
   - Board number : 2
         - Sampling clock source : 2
         - Trigger output IO : OFF
         - Is a clock master : No.
         - Core Version: 0x0201
         - Driver Version: 0x0023
         - PCB temp: 52.5C
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: 93708448
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:BA:BE:02'
MAC Address of Host is: 'AC:16:2D:06:12:EE'

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.

In Matlab, change the current folder to the folder where record.bin resides.

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.

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).

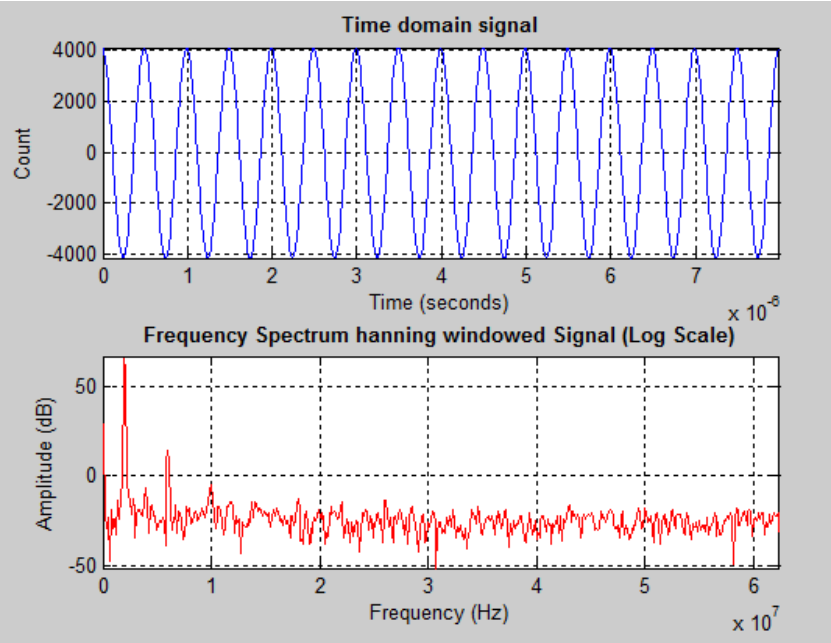

Figure 9-2 Matlab data plot result

9.2.2 MI125_Record_64

1. Browse to *bas/examples/mi125/host/scripts*.
2. Using a text editor, open/display
   - *MI125_Record_64.bat* if you use Windows
   - *MI125_Record_64.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 the *Execution* portion of this section. 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\fmc\MI125
RecordData	Record/Playback	%BASROOT%\doc\cores\RTDEx_RecordPlayback
RetrieveRecordedData	Record/Playback	%BASROOT%\doc\cores\RTDEx_RecordPlayback

The script:

- Initializes the MI125 cores and boards using application *MI125_Init* with configuration file *MI125_Init_64.ini*.
- Configures the MI125-Record interface core to route all 64 channels of the MI125 stacks to the Record/Playback core, downsampling signals by 4. Inside the FPGA, as parameters for the MI125-Record interface core are routed out to custom registers, application *CustomRegister_Write* (on register 13) is used to perform the operation.
- Configures the MI125-Record interface core 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*.

Execution

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

Windows

Double-click the MI125_Record_64.bat file.

The test starts automatically.

Linux

In a Linux terminal, change directory to *bas/examples/mi125/host/scripts*.

To start the example, run the following command

sudo ./MI125_Record_64.sh

Expected Results

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

Parsing MI125 Init 64.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 : 1
         - Trigger output IO : OFF
         - Is a clock master : Yes.
         - Core Version: 0x0201
         - Driver Version: 0x0023
         - PCB temp: 44.0C
   - 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
   - Board number : 3
         - Sampling clock source : 1
         - Trigger output IO : OFF
         - Is a clock master : No.
         - Core Version: 0x0201
         - Driver Version: 0x0023
         - PCB temp: 36.0C
   - Board number : 4
         - Sampling clock source : 2
         - Trigger output IO : OFF
         - Is a clock master : No.
         - Core Version: 0x0201
         - Driver Version: 0x0023
```

```
         - PCB temp: 37.0C
Done!


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

Configuring the Perseus at IP = 192.168.0.101, please wait.
CustomRegister Write...
         - Custom Register ID : 13
         - Value to write : 6
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: 103998208
The data will be transferred through RTDEx Gigabit Ethernet
MAC Address of Perseus is: '00:D0:CC:BA:BE:02'
MAC Address of Host is: 'AC:16:2D:06:12:EE'

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

In Matlab, change the current folder to the folder where record.bin resides.

Run the following command: bintofft('filename', num_of_channels_to_plot, data_type, framesize, acquisition_frequency);.

For example, *bintofft('record.bin', 64, 'int16', inf, 31.25e6);*.

This will plot the time and the frequency domain for the 64 acquired channels.

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).

The ADC sampling rate is 125 MHz but the specified data in divided by 4 because of the interface logic configuration. In 64-channel mode, the data is down-sampled by 4 due to the onboard board memory transfer rate limitation.

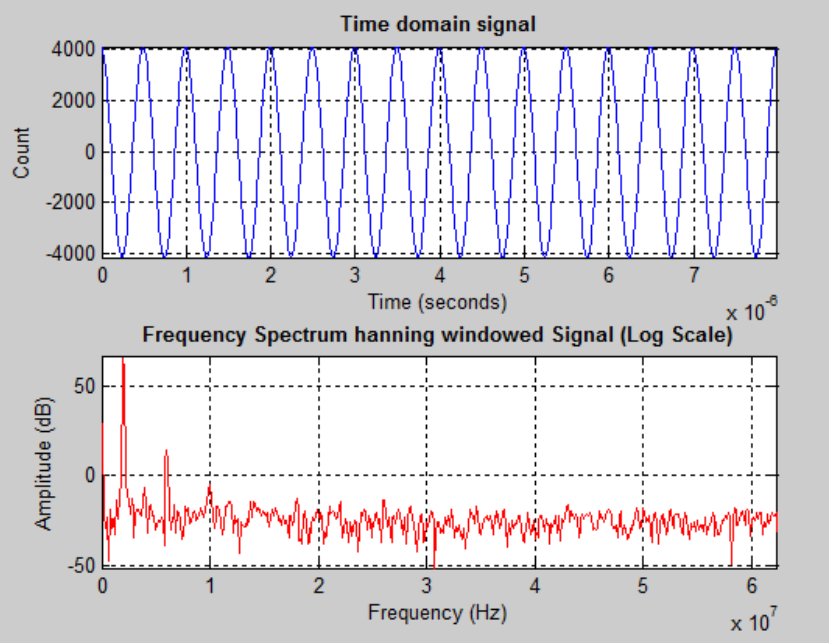

Figure 9-3 Matlab data plot result