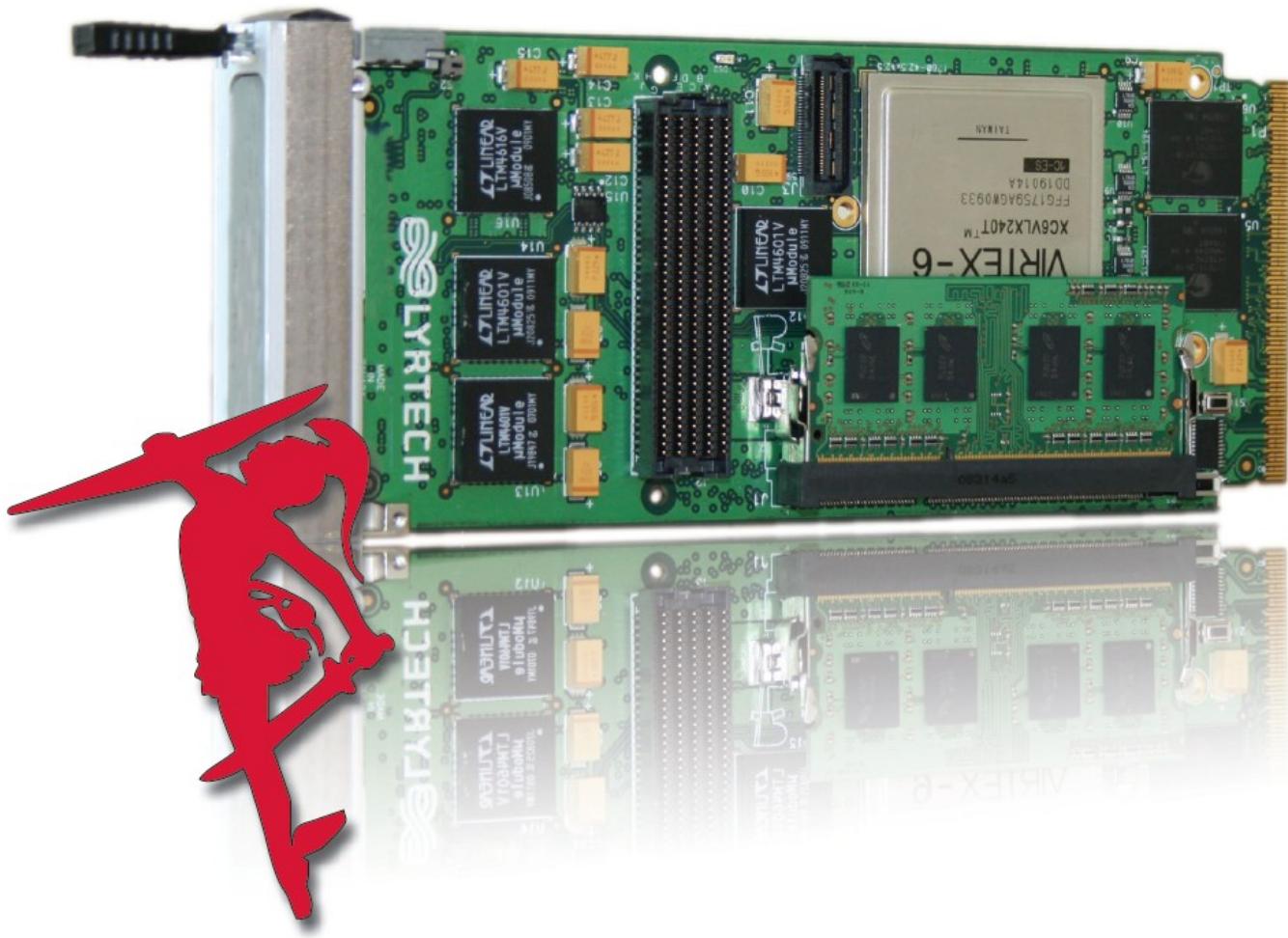


Perseus 601X

User's Guide



December 2015

Revision history

Revision	Date	Comments
1.0	August 2010	First edition.
1.1	October 2010	Minor corrections in relation to the flash memory.
1.2	January 2011	Updated for software release 5.1.0.
1.3	February 2011	Added hardware setup procedures.
1.4	May 2011	Updated for software release 5.2.0.
1.5	July 2011	Updated for software release 5.3.0.
1.6	October 2011	Updated for software release 5.4.0.
1.7	March 2012	Updated for software release 5.5.
1.8	July 2012	Updated for software release 5.6.
1.9	September 2012	Updated page layout.
2.0	December 2012	Updated to AXI bus.
2.1	April 2013	New software release
2.2	October 2013	Updated for software Release 6.4
2.3	March 2014	Information correction Up to date for Release 6.5
2.4	November 2014	Added QDR memory and Mestor Breakout Box precision concerning non-support in Perseus software tools. Added MO1000 support Creation of the Mechanical Assembly section Up to date for Release 6.6
2.5	January 2015	Corrected some section titles with capital letter missing
2.6	October 2015	Changed Glossary Removed all software related information and relocated to Perseus Programmer's Reference Guide Up to date for Release 7
2.7	December 2015	Removed QDR2 – no longer populated on Perseus601x and is not supported for earlier version

© Nutaq All rights reserved.

No part of this document may be reproduced or used in any form or by any means—graphical, electronic, or mechanical (which includes photocopying, recording, taping, and information storage/retrieval systems)—without the express written permission of Nutaq.

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 ™ 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

1	Introduction	1
1.1	Conventions	1
1.2	Glossary	1
1.3	Technical Support	2
2	Product Description.....	4
2.1	Overview	4
2.1.1	Outstanding Features	4
2.1.2	Options	4
2.2	Hardware Description	6
2.2.1	Perseus Top	6
2.2.2	Perseus Bottom	9
2.2.3	Front Panels.....	9
2.3	Component Details	11
2.3.1	Mestor Interface.....	11
2.3.2	AMC Backplane Connector	12
2.3.3	Module Management Controller.....	14
2.3.4	Perseus Clock Tree	15
2.3.5	Virtex-6 High-Speed Serial Transceivers (GTX)	16
2.4	Perseus Software Architecture	24
2.4.1	Distributed Computing System.....	24
2.4.2	Command-Line Interface	24
2.4.3	Perseus Software Platform	25
3	Mechanical Assembly	26
3.1	Replacing Face Plates.....	26
3.1.1	Removing a Face Plate.....	26
3.1.2	Installing a Face Plate	28
3.2	Installing FMCs	32
3.3	Installing the Mestor.....	34
3.3.1	Connecting a Mestor Adapter	34
3.3.2	Installing the Mestor Expander	34
3.3.3	Connecting a Mestor Breakout Box.....	38
3.4	Replacing the DDR3 SODIMM	39
3.4.1	Removing the DDR3 SODIMM	39
3.4.2	Inserting the DDR3 SODIMM	39
3.5	Perseus in a μTCA Chassis	40
3.5.1	Inserting a Perseus in a μTCA Chassis	40
3.5.2	Removing a Perseus from a μTCA Chassis	41
4	Mestor Connection	42
4.1	Hardware Description	45
4.1.1	Mestor Expander	45
4.1.2	Mestor Breakout Box.....	49
4.1.3	Mestor Breakout Box Perseus FPGA Core	51

5	Specifications	52
5.1	Mechanical Specifications.....	52
5.1.1	Overall Clearance.....	52
5.1.2	Board Dimensions.....	53
5.2	FPGA Pin Assignments	54
Appendix A.	FPGA Address Spaces	71
Processor Boot Memory.....	71	
Register User Space.....	71	
Perseus Registers	71	
Configuration Registers.....	71	
Nutaq IP Cores.....	77	
System Peripherals.....	78	
DDR3 Memory Processor	79	
Flash Memory.....	79	
Appendix B.	FMC Connector	80
Pin Assignments	80	
Compatibility Considerations	88	

List of Figures and Tables

Figure 2-1 Perseus module5
Figure 2-2 Perseus block diagram6
Figure 2-3 Hardware parts on the top of the Perseus6
Figure 2-4 External 12 V power connector pin assignments7
Figure 2-5 Hardware parts on the bottom of the Perseus.....	.9
Figure 2-6 Half-size front panel description10
Figure 2-7 Full-size front panel description11
Figure 2-8 Perseus JTAG chain12
Figure 2-9 Perseus clock scheme15
Figure 2-10 Multiple GTX with shared reference clock17
Figure 2-11 GTX detailed diagram20
Figure 2-12 Absolute gain (voltage transfer function).....	.21
Figure 2-13 LTC2641 serial timing diagram21
Figure 2-14 Software platform25
Figure 3-1 Removing screws26
Figure 3-2 Pulling face plate outward.....	.27
Figure 3-3 Disengaging the face plate assembly from the Perseus27
Figure 3-4 Unhooking the face plate assembly from the Perseus28
Figure 3-5 Installing an AMC handle29
Figure 3-6 Installing the LED pipe29
Figure 3-7 Aligning the AMC handle and LED pipe29
Figure 3-8 Secure assembly30
Figure 3-9 Installing the support.....	.30
Figure 3-10 Securing the support30
Figure 3-11 Turning the Perseus30
Figure 3-12 Pulling the AMC handle outward.....	.31
Figure 3-13 Aligning the face plate31
Figure 3-14 AMC handle and face plate.....	.31
Figure 3-15 Lifting the edge of the face plate.....	.32
Figure 3-16 Locking the face plate into position.....	.32
Figure 3-17 Removing the screws securing the blank front plate32
Figure 3-18 ADAC250, as shipped33
Figure 3-19 Aligning Perseus and FMC33
Figure 3-20 Securing the FMC on the Perseus.....	.33
Figure 3-21 Mating connectors.....	.34
Figure 3-22 Placing the washers on the FMC.....	.34
Figure 3-23 Mating the Mestor expander and the Perseus35
Figure 3-24 Screws and positions35
Figure 3-25 Mestor expander secured over an ADAC250 FMC35
Figure 3-26 Full size face plate over Mestor expander.....	.36
Figure 3-27 Securing the blank front plate to the Perseus36
Figure 3-28 Placing the rubber washers36
Figure 3-29 Placing the metal washers37
Figure 3-30 Mating the Mestor expander and the Perseus37
Figure 3-31 Screws and positions37
Figure 3-32 Securing the Mestor expander to the face plate.....	.38

Figure 3-33 Connecting and securing the breakout box cable to the Perseus	38
Figure 3-34 Connecting and securing the breakout box cable to the breakout box	38
Figure 3-35 Releasing the DDR3 SODIMM	39
Figure 3-36 Removing the SODIMM	39
Figure 3-37 Inserting the SODIMM in its connector	40
Figure 3-38 Securing the DDR3 SODIMM in place	40
Figure 3-39 Free μTCA chassis slot.....	40
Figure 3-40 Inserting the Perseus into the chassis	41
Figure 3-41 Pushing in AMC handle	41
Figure 4-1 Mestor connector	42
Figure 4-2 Mestor JTAG adapter	44
Figure 4-3 Mestor expander	44
Figure 4-4 Mestor breakout box	44
Figure 4-5 Mestor expander connectivity.....	45
Figure 4-6 Mestor breakout box	49
Figure 4-7 Connecting to the JTAG chain with the breakout box	50
Figure 5-1 Top and side dimensions	53
Figure 5-2 Bottom dimensions and assembly hole locations	54
Figure 5-3 I ² C bus connections.....	89
 Table 1 Glossary	2
Table 2 Supported Virtex-6s	8
Table 3 JTAG device order	12
Table 4 AMC backplane connector pin assignments	14
Table 5 MMC sensor definition records.....	14
Table 6 LVDS buffer behavior	16
Table 7 AMC and FMC port map (LX240 and SX315 Virtex-6)	18
Table 8 AMC and FMC port map (LX550 and SX475 Virtex-6)	19
Table 9 LVDS buffer behavior	23
Table 10 Mestor interface pin assignments of the Perseus board	43
Table 11 Pin assignments.....	47
Table 12 DPIO naming convention between the Perseus board and the Mestor Expander	48
Table 13 Mapping between VHDCI Mestor Expander connector and Perseus FPGA DPIO	48
Table 14 Pin assignments.....	50
Table 15 Pin assignments.....	50
Table 16 Pin assignments.....	51
Table 17 External reset	54
Table 18 AMC connector (LX240/SX315 Virtex-6 Perseus).....	56
Table 19 AMC connector (LX550/SX475 Virtex-6 Perseus).....	58
Table 20 AVR (IPMI)	58
Table 21 DDR3 memory processor	60
Table 22 Control pins	61
Table 23 DDR3 SODIMM	66
Table 24 DPIO connector	67
Table 25 Backplane clocks	67
Table 26 Flash memory	69
Table 27 200 MHz reference clock.....	69
Table 28 GTX reference clocks.....	70
Table 29 Front panel UART connector	70
Table 30 DAUGTHERS_ABSENT status register.....	72
Table 31 Offset 0x00 descriptions	72
Table 32 CLK_CTRL status register	73
Table 33 Offset 0x04 descriptions	73
Table 34 VADJ_AND_FPGA_PROG_CTRL status register	73
Table 35 Offset 0x08 descriptions	74
Table 36 LED_CTRL status register.....	74

Table 37 Offset 0x0C descriptions	74
Table 38 CLK_CTRL status register.....	75
Table 39 Offset 0x10 descriptions	75
Table 40 MEMORY_INIT_DONE_STATUS register	75
Table 41 Offset 0x14 descriptions	75
Table 42 User-defined custom registers.....	76
Table 43 Reserved registers.....	76
Table 44 MMC control register.....	76
Table 45 Offset 0xA0 descriptions	76
Table 46 Access status	77
Table 47 Offset 0xA4 descriptions.....	77
Table 48 Last read address register	77
Table 49 Offset 0xA8 descriptions	77
Table 50 Main IP cores provide by Nutaq.....	78
Table 51 FMC connector (LX240/SX315 Virtex-6 Perseus)	84
Table 52 FMC connector (LX550/SX475 Virtex-6 Perseus)	88

1 Introduction

Congratulations on the purchase of the Perseus 601X.

This document contains all the information necessary to understand and use the Perseus 601X. 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. 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

Term	Definition
	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 Perseus601x (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.
Peer	A host peer is an associated host running RTDEEx 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

This chapter presents the hardware description of the Perseus 601X module.

2.1 Overview

The Perseus 601X advanced mezzanine card (AMC or AdvancedMC) is designed around the powerful Virtex-6 FPGA, combining unsurpassed fabric flexibility and a colossal external memory, as well as benefiting from multiple high-pin-count, 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 this, the Perseus' FMC expansion site offers almost endless I/O possibilities.

2.1.1 Outstanding Features

- Mid-size AMC for μTCA and AdvancedTCA platforms
- Choice of powerful LXT and SXT Virtex-6 FPGAs
- High-pin-count VITA 57.1 FMC expansion site for I/Os
- DDR3 SODIMM interface to upgrade system memory
- Support for AMC R2.0 and R1.0 through onboard clock switch
- Available GTX base clocks — 100 MHz, 125 MHz, 156.25 MHz (PCIe/GigE/XAUI/SRIO)
- Fabric clock — RX or TX (100 MHz PCIe, default)
- IPMI controller (based on the AVR version of the Pigeon Point AdvancedMC MMC)
- FPGA and IPMI JTAGs on the Mestor interface



2.1.2 Options

The Perseus benefits from an extensive line of options that allow you to further enhance your development experience and final applications.

Optional Nutaq development software

The optional Perseus MBDK allows you to easily design high-performance digital signal processing systems within the card FPGA with the MATLAB/Simulink design environment and extensive DSP IP libraries from Xilinx.

Visit www.nutaq.com for details about this software kit.



Optional debugging hardware

From the AMC backplane or from the onboard Mestor interface. Nutaq offers two optional Mestor debugging modes:

Mestor-to-FPGA JTAG adapter

Offers direct onboard access to the FPGA JTAG chain.

**Mestor expander**

Offers front-panel access to the FPGA and IPMI JTAGs, 14 user LVDS I/Os, one clock, and an FPGA UART interface (serial RX/TX—Mini-B USB). Note, however, that using the Mestor expander, transforms the Perseus form factor into a fullsize AMC, thus the expander is supplied with a full-size face plate for the Perseus.



Contact Nutaq for details about these options at info@nutaq.com.

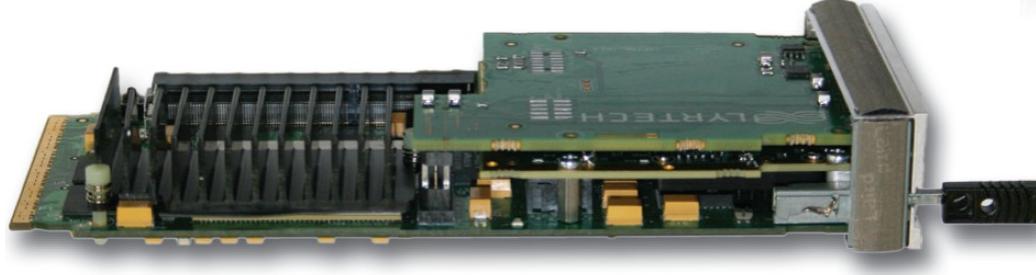


Figure 2-1 Perseus module

Optional FMC modules

Through its VITA 57.1 high-pin-count (HPC) FMC expansion site, the Perseus can interface with a wide variety of FMCs, such as the ADAC250 high-speed dual A/D and D/A FMC module.

Visit the Nutaq Web site at www.nutaq.com for more FMC modules for the Perseus as they become available.

2.2 Hardware Description

This section presents the Perseus hardware from a functional standpoint, introducing its parts and their functions.

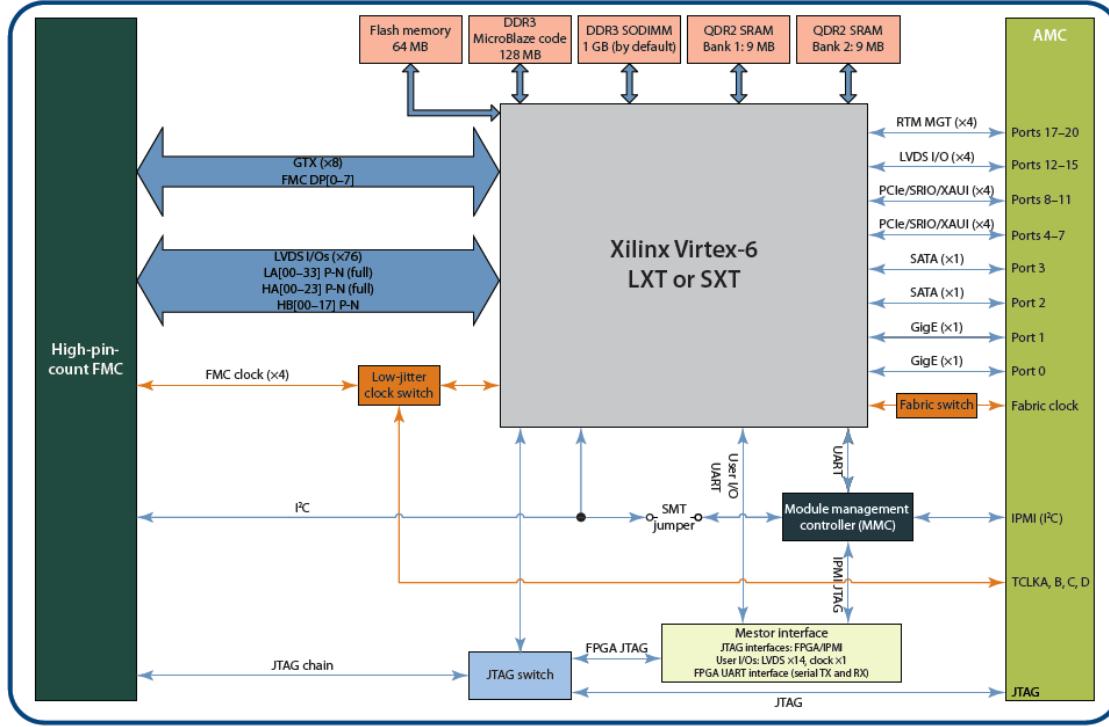


Figure 2-2 Perseus block diagram

2.2.1 Perseus Top

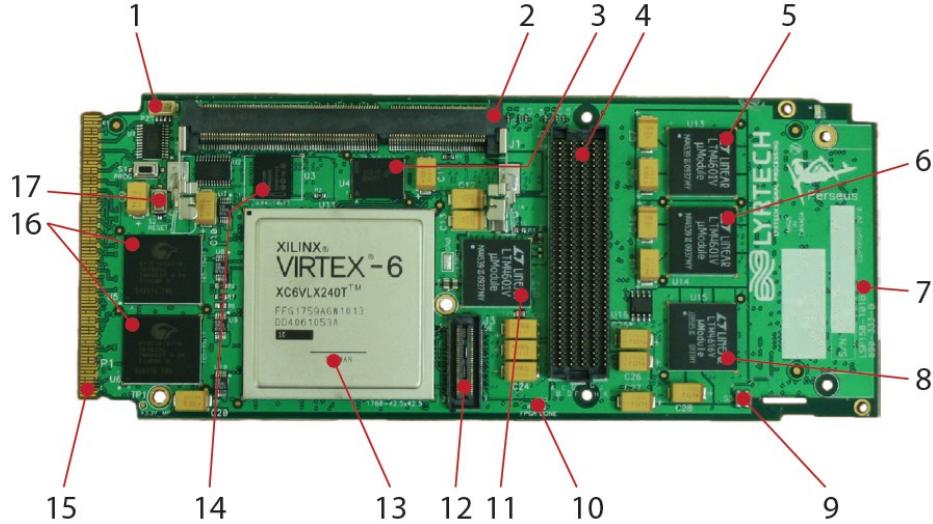


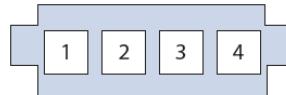
Figure 2-3 Hardware parts on the top of the Perseus

1 External 12-V power connector

This connector supplies 12 V for expansion. It could, for example, add power to an external device connected to the Mestor expansion connector. The connector is rated at 0.7 A per pin and the figure below illustrates its pin assignments.

Important:

This connector cannot be used to power the Perseus from an external power supply. Connecting an external power supply to this connector may seriously damage your card.



- 1: 12 V
- 2: Ground
- 3: 12 V
- 4: Ground

Figure 2-4 External 12 V power connector pin assignments

2 64-bit DDR3 SODIMM interface

By default, this interface is connected to a 64-bit, 1 GB SODIMM for the FPGA, which can be used to support acquisition processing. The interface supports SODIMMs up to 8 GB. Because of the Virtex-6 architecture, the maximum speed of the memory is limited to 400 MHz.

Note:

At the time of writing, the BAS Software Tools only support 1 and 4 GB SODIMMs.

3 8-bit DDR3 SDRAM

8-bit, 128 MB DDR3 SDRAM for the FPGA. By default, this memory is used to support embedded processing. Because of the Virtex-6 architecture, the maximum speed of the memory is limited to 400 MHz.

4 High-pin-count FMC (VITA 57.1) connector

FMC interface fully complying with VITA 57.1 specifications. Using FMC modules minimizes design efforts and resources. Refer to the VITA specification for details. The Perseus supports 76 FMC LVDS I/Os (LA00-33/HA00-23/HB00-17) with four FPGA global clocks (two FMC-to-FPGA clocks and two bidirectional clocks), and eight FMC GTX transceivers DP[0-7] with two FMC-to-FPGA GTX reference clocks.

5 1.5 V power supply

Switching regulator that can supply up to 12 A at 1.5 V from the 12 V input. This power is used to supply the DDR3 SDRAM and its associated I/O bank.

6 3.8 V power supply

Switching regulator that can supply up to 12 A at 3.8 V from the 12 V input. This power is used as intermediate tension to supply different linear and switching regulators that are sensitive to large voltage drops.

7 Board revision

The letter following LSP123-456 indicates the revision of the board. A indicates a revision A Perseus, B indicates a revision B Perseus and so on.

8 2.5 V and Vadj power supply

Switching regulator that can supply up to 6 A at 2.5 V from the 3.8 V input. The 2.5 V power is used to supply the FPGA LVDS bank and its LVCMOS signal. The Vadj output is used to supply the FPGA and the card FMC site. The output can supply up to 6 A at 1.2 V, 1.5 V, 1.8 V, or 2.5 V, depending on the FPGA selection pin (00;01;10;11).

9 AMC handle switch

Switch enabling hot swap. When you press the switch (through the AMC handle), you instruct the system to connect the Perseus to the AMC backplane and to apply power.

10 FPGA done LED

LED indicating the status of the FPGA done signal. When a bitstream is correctly loaded to the FPGA without error, the LED lights.

11 1 V power supplies

Switching regulator that can supply up to 12 A at 1 V from the 12 V input. This power is used to supply the FPGA core.

12 Mestor interface

Expansion connector that supplies outside-world connectivity to the Perseus, offering FPGA JTAG and IPMI JTAG accesses. Additionally, you can use this interface as a high-speed expansion port, using its 14 LVDS I/Os and one clock. The connector is a Samtec QSH-030-01-L-D-A-K. See Mestor interface for details.

A serial RX/TX port is also directly connected to the FPGA, making it possible for the MicroBlaze soft processor core to connect a UART to the outside world.

13 FPGA

Main processing unit of the Perseus. This Virtex-6 device supplies all the necessary processing power to design your algorithms and incorporates high-speed serial transceivers (GTX).

Perseus	Device	Package	Speed	Flip flop	Logic cell	Block RAM bit (#)	XtremeDSP slices	MMCM
6010	LX240T	SX475T	-1	301440	241152	14976 (416)	768	12
6011	LX550T	SX475T	-1	687360	549888	22752 (632)	864	18
6012	SX315T	SX475T	-1	393600	314880	25344 (704)	1344	12
6013	SX475T	SX475T	-1	595200	476160	38304 (1064)	2016	18

Table 2 Supported Virtex-6s

Important:

Make sure that the supplied heatsink is correctly installed on your FPGA. An incorrectly installed heatsink could damage your card FPGA. Verify that the heatsink two fixing pins are correctly in place.

Note:

By default, Perseus always come equipped with -1 speed FPGAs, but this can differ if you have requested it.

14 Flash memory

A 64 MB bottom-boot NOR flash memory (PC28F512P30BFA) is available to the FPGA. The memory is used to hold bitstreams and the embedded software image. By default, bitstreams are held in the upper part of the memory (the FPGA is configured BPI down, which means it loads its bitstream from address 0xFFFFFFF), while the embedded software image is held in the lower part of the memory. Contact Nutaq for more flash memory options.

Note:

At this time, ISE Foundation 12.x cannot program a bottom-boot flash memory through JTAG. To use the flash memory to boot your FPGA, you must use an external core (MicroBlaze or custom made) to program your bitstream or boot loader into the flash memory. Refer to answer record 36599 on the Xilinx Web site for details. Nutaq supplies the Linux CCE fpgsetFlash function to perform this operation.

15 AMC backplane connector

This connector supplies a huge bandwidth to the Perseus. A total of 24, gigabit-capable high-speed ports are available. Many clock inputs and outputs are also available, as well as a JTAG interface.

Important:

The Ethernet connection of the Perseus is only capable of Gigabit Ethernet (GigE). It is important, when connecting the chassis containing the Perseus point-to-point with your computer Ethernet card or Ethernet switch that it is Gigabit Ethernet capable. If not, you may encounter Ethernet communication problems.

16 18-bit QDR2 SRAM

Two, 18-bit, 9 MB banks of QDR2 SRAM are available to the FPGA. This memory is used to support high-bandwidth processing.

Important:

This memory is not supported by the Perseus software tools and is not populated from Perseus Revision C6

17 Buttons

The Perseus is equipped with two buttons. S1 is used to force the FPGA to reload from the flash memory, while S2 is directly routed to the FPGA and is normally used to reset the MicroBlaze.

2.2.2 Perseus Bottom

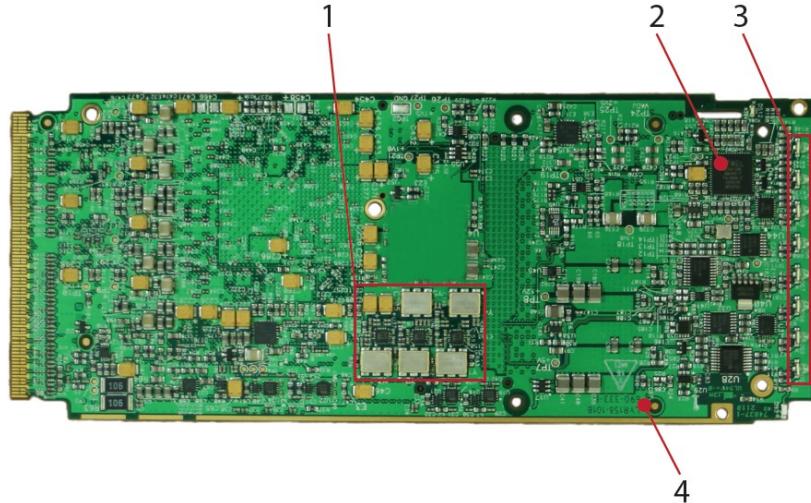


Figure 2-5 Hardware parts on the bottom of the Perseus

1 Local oscillators

The Perseus is equipped with five local oscillators that supply the source clocks for your designs. A 200 MHz clock drives logic and the multiport memory controller (MPMC). A 400 MHz clock drives the DDR3 SDRAM interface. A 100 MHz clock, a 125 MHz clock, and a 156.25 MHz clock drive the GTX section of the Virtex-6.

Note:

Revisions A and B of the Perseus have a 625 MHz clock instead of the 156.25 MHz clock.

2 Module management controller

More commonly known as the MMC, this controller manages the intelligent platform management interface (IPMI) of the Perseus and is necessary for hot swap. The IPMI controller is based on the AVR version of the Pigeon Point AdvancedMC MMC. The controller firmware may be modified. Nutaq is equipped to deal with such requests or you can do it yourself by buying the Pigeon Point AdvancedMC MMC.

The MMC is also responsible for managing AMC-specific LEDs on the front panel. See Front panels for details.

3 Front panel LEDs

The Perseus is equipped with eight dual-color LEDs connected to the FPGA through a buffer. The buffer must be enabled to drive the LEDs. LEDs light when a zero is driven to the appropriate I/Os of the FPGA.

4 Board revision

The letter following LSP123-456 indicates the revision of the board. A indicates a revision A Perseus, B indicates a revision B Perseus and so on.

2.2.3 Front Panels

Depending on the configuration of your Perseus, you may need to use one of the two available front panels: half size and full size.

Perseus half-size front panel

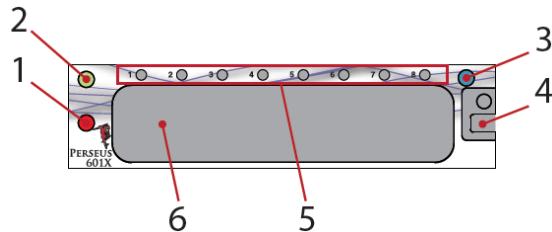


Figure 2-6 Half-size front panel description

1 Malfunction LED

This red LED lights when a critical power supply signal is not asserted as good or the board is overheating.

2 Power LED

This green LED lights when all the signals from critical power supplies are asserted as good and that a bitstream can be loaded in the FPGA. The LED blinks when the MicroBlaze is functioning properly.

3 AMC state LED

This blue LED is governed by the operational state (hotswap behavior) of the Perseus MMC (see previous page).

AMC state LED conditions:

Condition	Description
Off	The Perseus is running properly. At this point, the power LED is also on.
Blinking	The Perseus is: - Powering up (after pressing the AMC handle, upon insertion) - Powering down (after pulling the AMC handle)
On	When the LED is on, you can: - If you inserted the Perseus in the chassis: press the AMC handle to power up the card. - If you are attempting to remove the Perseus from the chassis, do so.

Note:

If the AMC state LED does not stop blinking, it is likely that the Perseus cannot power up. Refer to your chassis IPMI log for details as to the reason(s). Usually, using a more powerful chassis or removing cards from your current chassis will resolve the situation.

4 AMC handle

The handle of the Perseus fits here.

5 Front panel LEDs

The Perseus is equipped with eight dual-color LEDs connected to the FPGA through a buffer. The buffer must be enabled, through the output enable pin of the buffer, to drive the LEDs. LEDs light when a zero is driven to the appropriate I/Os of the FPGA.

Note:

LED behavior is defined by you for your applications.

6 FMC module faceplate

The faceplate of your chosen FMC module fits here.

Perseus full-size front panel

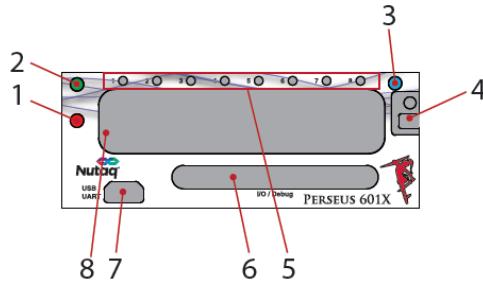


Figure 2-7 Full-size front panel description

1
Malfunction LED

2
Power LED

3
AMC state LED

4
AMC handle

5
Front panel LEDs

6 I/O / Debug port

When the Mestor is connected to the Perseus, this port is used to connect a Mestor breakout box, which offers several debugging capabilities.

7 USB/UART port

When the Mestor is connected to the Perseus, this port is used as a USB-to-UART bridge for your remote computer.

8
FMC module faceplate

2.3 Component Details

2.3.1 Mestor Interface

The Mestor interface is the Perseus connection. It offers JTAG capabilities, 14 differential I/Os or 28 single-ended I/Os, a differential clock I/O, and an FPGA UART interface (serial TX and RX) based on LVCMOS 2.5 V signaling. The Mestor expander allows you to interface directly with these I/Os. See chapter 3 for details.

JTAG

The Mestor interface JTAG facilities allow you to use JTAG in one of three ways:

- Through the backplane
- Through the Mestor adaptor
- Through Mestor breakout boxes

The FPGA JTAG chain is used to download bitstreams and debug the FPGA. It is also connected to the FMC connector and to the Mestor interface. JTAG configurations are automatically adjusted according to your hardware. The chain also supports two configurations: the AMC backplane or the Mestor interface as master. By default, the chain is routed through the AMC backplane but when a JTAG pod is present at the Mestor interface, it automatically becomes the chain master. Figure 7 shows the JTAG chain configuration.

Board revision	Device order
B	FMC -> FPGA
C	FPGA -> FMC

Table 3 JTAG device order

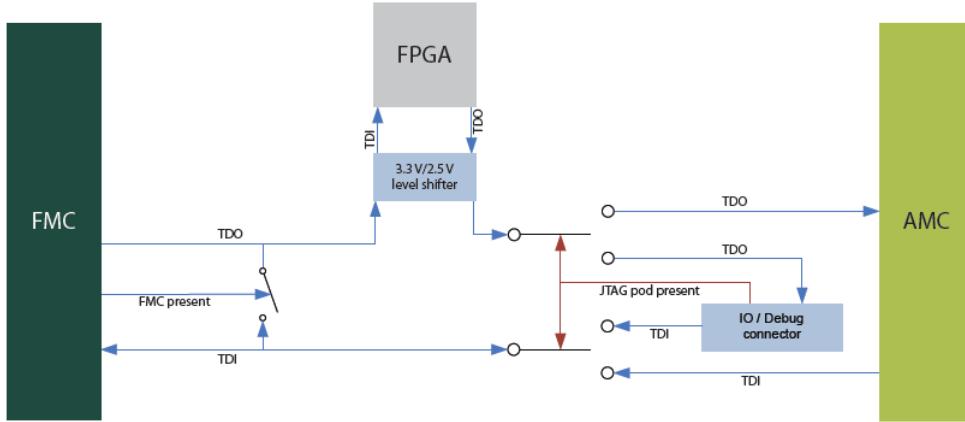


Figure 2-8 Perseus JTAG chain

The Mestor interface also offers 14 LVDS I/Os and one clock used as a high-speed expansion port. See the chapter on the Mestor for details. A serial TX/RX port, directly connected to the FPGA, also allows the MicroBlaze to connect a UART to the outside world. See the Mini-B USB interface on the Mestor.

2.3.2 AMC Backplane Connector

The Perseus comes with a fully compliant AMC backplane connector. The following pin assignments are provided as reference, so that you can ascertain whether your backplane is compatible with the Perseus.

Note:

The pin assignments below take into account that you are familiar with AMC signals. Refer to the PICMG AMC.0 R2.0 specifications for signal descriptions.

Pin	Assignment								
1	GND	35	TX3+	69	RX7-	103	TX10+	137	GND
2	12 V	36	TX3-	70	GND	104	GND	138	TCLKD-
3	~PS1	37	GND	71	IPMI_SDA	105	RX11-	139	TCLKD+
4	3.3V_MP	38	RX3+	72	12 V	106	RX11+	140	GND
5	GA0	39	RX3-	73	GND	107	GND	141	RX17-
6	RSVD	40	GND	74	TCLKA+	108	TX11-	142	RX17+
7	GND	41	~Enable	75	TCLKA-	109	TX11+	143	GND
8	RSVD	42	12 V	76	GND	110	GND	144	TX17-

Pin	Assignment								
9	12 V	43	GND	77	TCLKB+	111	RX12-	145	TX17+
10	GND	44	TX4+	78	TCLKB-	112	RX12+	146	GND
11	TX0+	45	TX4-	79	GND	113	GND	147	RX18-
12	TX0-	46	GND	80	FCLKA+	114	TX12-	148	RX18+
13	GND	47	RX4+	81	FCLKA-	115	TX12+	149	GND
14	RX0+	48	RX4-	82	GND	116	GND	150	TX18-
15	RX0-	49	GND	83	~PS0	117	RX13-	151	TX18+
16	GND	50	TX5+	84	12 V	118	RX13+	152	GND
17	GA1	51	TX5-	85	GND	119	GND	153	RX19-
18	12 V	52	GND	86	GND	120	TX13-	154	RX19+
19	GND	53	RX5+	87	RX8-	121	TX13+	155	GND
20	TX1+	54	RX5-	88	RX8+	122	GND	156	TX19-
21	TX1-	55	GND	89	GND	123	RX14-	157	TX19+
22	GND	56	IPMI_SCL	90	TX8-	124	RX14+	158	GND
23	RX1+	57	12 V	91	TX8+	125	GND	159	RX20-
24	RX1-	58	GND	92	GND	126	TX14-	160	RX20+
25	GND	59	TX6+	93	RX9-	127	TX14+	161	GND
26	GA2	60	TX6-	94	RX9+	128	GND	162	TX20-
27	12 V	61	GND	95	GND	129	RX15-	163	TX20+
28	GND	62	RX6+	96	TX9-	130	RX15+	164	GND
29	TX2+	63	RX6-	97	TX9+	131	GND	165	JTAG_TCK
30	RX2+	64	GND	98	GND	132	TX15-	166	JTAG_TMS
31	GND	65	TX7+	99	RX10-	133	TX15+	167	JTAG_TRSTN
32	RX2+	66	TX7-	100	RX10+	134	GND	168	JTAG_TDO
33	RX2-	67	GND	101	GND	135	TCLKC-	169	JTAG_TDI

Pin	Assignment								
34	GND	68	RX7+	102	TX10-	136	TCLKC+	170	GND

Table 4 AMC backplane connector pin assignments

2.3.3 Module Management Controller

The module management controller (MMC) is responsible for the AMC hot swap capabilities and the supervision of its functionalities. In an AMC chassis, each card is known as a field-replaceable unit (FRU). Each FRU can be queried for its status. (Refer to the user's guide of your chassis to learn how to read your FRU data.)

The MMC is equipped with a suite of sensors that monitor the card parameters. Each sensor is linked to an ID number so that it can read its value through the IPMI link. The following table presents the ID of each sensor.

ID	Name	Description
1	Hot swap	Hot swap handle switch.
2	3.3 V	Voltage reading of the 3.3V_MP power supply.
3	12 V	Voltage reading of the 12 V power supply.
4	DS75 temp	I ² C temperature sensor (U16).
5	BMC watchdog	AVR watchdog.
6	PG_DDR3_VTT	Power good signal from the DDR3_VREF supply.
7	PG_3V3	Power good signal from the 3V3 supply.
8	PG_1V8	Power good signal from the 1V8 supply.
9	PG_VADJ	Power good signal from the V _{adj} supply.
10	PG_2V5	Power good signal from the 2V5 supply.
11	PG_3V8	Power good signal from the 3V8 supply.
12	PG_1V5	Power good signal from the 1V5 supply.
13	PG_1V0	Power good signal from the 1V0 supply.
14	FMC_PG_M2C	Power good signal from the 3V3 supply of the FMC module on the Perseus.
15	PG_MGT_AVTT_N	Power good signal from the 1V2 supply for the MGTAVTT_N supply.
16	PG_MGT_AVCC_N	Power good signal from the 1V0 supply for the MGTAVCC_N supply.
17	PG_MGT_AVCC_S	Power good signal from the 1V0 supply for the MGTAVCC_S supply.
18	PG_MGT_AVTT_S	Power good signal from the 1V2 supply for the MGTAVTT_S supply.

Table 5 MMC sensor definition records

2.3.4 Perseus Clock Tree

The Perseus is equipped with many clock sources. Depending on your application, you can select almost any of the FPGA clocks.

As you can see in the diagram below, from a clocking point of view, the Perseus is very flexible. Each TCLKx can be routed three different ways—all you need to do is configure the clock switches.

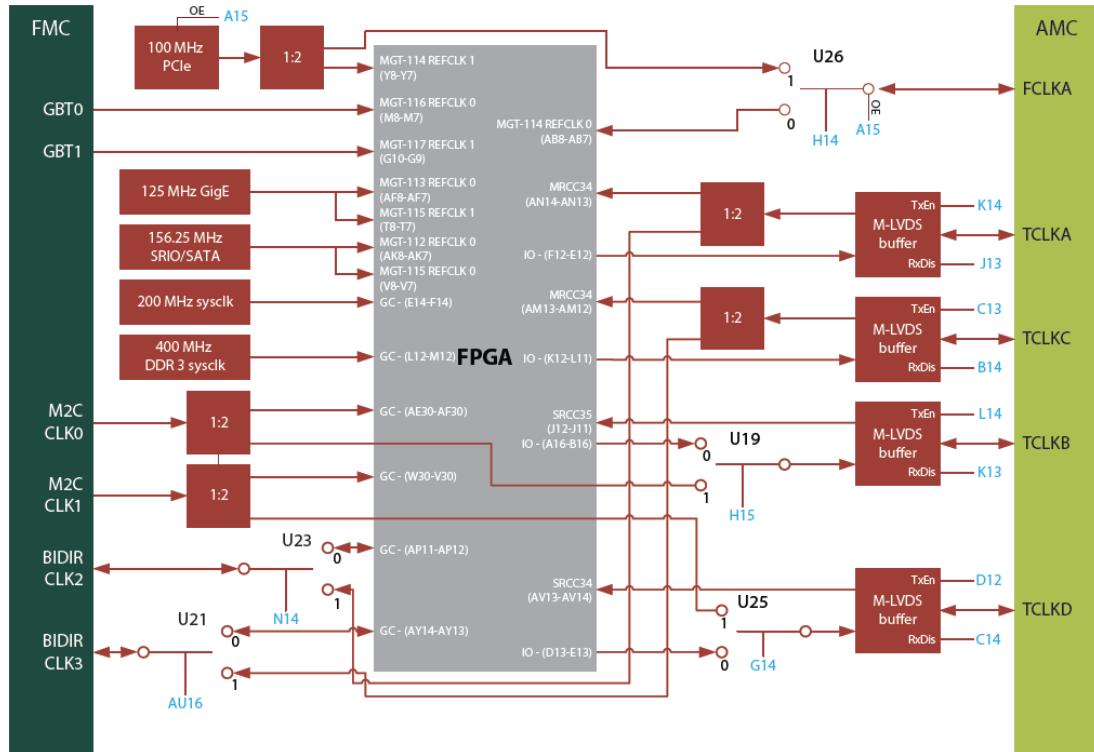


Figure 2-9 Perseus clock scheme

Type of reference clocks

There are four types of clocks on the Perseus: global clocks (GC), the multi-region capable clocks (MRCC), single-region capable clocks (SRCC), and Gigabit transceiver (MGT) reference clocks. Clock outputs are only normal I/Os.

Global clock

In the Perseus FPGA, global clocks are a dedicated network specifically designed to route all the clock inputs to the various resources of the FPGA. This network is designed for low skew and low-duty-cycle distortion, to consume very little power, and to have an improved tolerance to jitter compared to older generations. They are also designed to support very-high-frequency signals. On the Perseus, FMC clocks are used to forward A/D, D/A, and communication reference clocks, making it necessary to connect FMC clocks to GC inputs. The DDR3 400 MHz reference must also be connected to the DDR3 memory controller through a GC. A 200 MHz pulse is also connected to the FPGA as a reference clock for the I/O delays and as the MicroBlaze system clock.

Multi-region capable clocks and single-region capable clocks

Regional clocks are a different network that does not depend on the global clock network. Unlike global clocks, the span of regional clock signals are limited to three clock regions, while two I/O clock signals drive a single region (SRCC) and two other I/O clocks drive the regions/banks above/below (MRCC). The I/O banks in Virtex-6 FPGAs are the same size as a clock region.

Regional clocks are especially useful in source-synchronous interface designs, but, as TCLKxs are reference clocks, they are often used by the MMCM. The platform connects each TCLKx to unused MRCCs and SRCCs. These pins have, internally, a direct connection to an MMCM, thus you can reach the global clock tree and use the TCLKx as your design system clock.

Note:

To learn more about regional clocks, refer to Xilinx documentation.

MGT reference clock

MGTs, also called GTX, offer several available reference clock inputs. This feature allows the same MGT to be used under more than one standard. MGTs can, in fact, be used by six external reference clocks. As MGTs are organized in groups of four, they can use two reference clocks in their own group or two reference clocks in the groups directly before or after. For details, see Virtex-6 high-speed serial transceivers (GTX).

Clock controls

There exists a series of I/Os to control the clock routing on the Perseus, divided in two types of clock switching devices: SPDTs and LVDS buffers. SPDTs allow you to select between two clock trees, while the bidirectional LVDS buffers allow you to configure the AMC backplane telecommunication clocks (TCLKx) as inputs or outputs, or to select between three buffer states.

SPDTs

The switches are bidirectional and the selected input port switches the input to the desired output 0 or 1.

LVDS buffers

LVDS buffers act somewhat like multiplexers. They allow you to handle the different behaviors of the telecommunications clocks (TCLKx) of the AMC backplane. With the RX_DIS and TX_EN inputs, LVDS buffers also allow TCLKx to be input, output, or three states.

Control pin RX_DIS	Control pin TX_EN	Behavior
0	0	The backplane clock is received and transmitted to the clock output.
0	1	Loopback mode (the clock input is transmitted and a copy of the clock is received at the clock output).
1	0	All outputs are in high Z.
1	1	The backplane clock transmits the input clock.

Table 6 LVDS buffer behavior

2.3.5 Virtex-6 High-Speed Serial Transceivers (GTX)

This section explains Perseus from its high-speed serial transceivers perspective. These transceivers are usually called GTX in the range of speeds of the Virtex-6 FPGAs on the Perseus.

Unit interconnection

The AdvancedMC (AMC) standard relies heavily on high-speed serial communications to enable multiple communication schemes. AMC allows direct interconnections, point to point. Direct interconnection implies that Perseus A can communicate directly with Perseus B through a given high-speed port, while communicating with Perseus C through a different port and, at the same time, Perseus B communicates with Perseus C through yet another port in the same chassis. Of course, each link must go through a dedicated backplane interconnection (static) or through a switch in an MCH (dynamic). This architecture eliminates the bottle necks usual in other architectures such as cPCI, for example, where a processor must orchestrate all the data transfers. Such a topology makes it unintentionally impossible to use all the throughput of all the units at the same time in a given chassis; whereas, with AMC it becomes easy.

The ability to easily interconnect units also makes it possible to vastly expand systems. For example, if Perseus A receives data from its FMC interface, transmits this data to Perseus B, which transmits the result on its FMC interface. Now, imagine the processing performed by these two Perseus getting larger as you approach your

project goals (or through natural product evolution), so large that two units are no longer sufficient despite efficient partition design. AMC and the Virtex-6 GTX offer a simple solution: add a third Perseus between the existing two, properly connect the high-speed ports and spread the processing over the three cards. This way, you can easily daisy chain many units to suit your needs.

GTX layout

The Virtex-6 GTX are in groups of four to form what is called a GTX quad and are numbered 112 to 117 in the Xilinx documentation. Each quad has two dedicated clock reference inputs, called MgtRefClk. These inputs are used to supply the GTX with a high-quality clock signal. A GTX clock can come from the MgtRefClk of its own quad or from any one of the two adjacent quads, as long as the adjacent quad supplying the clock is populated with at least one GTX, (even if it is a dummy one). In other words, a given MgtRefClk can feed its own quad and the two adjacent ones, as illustrated here.

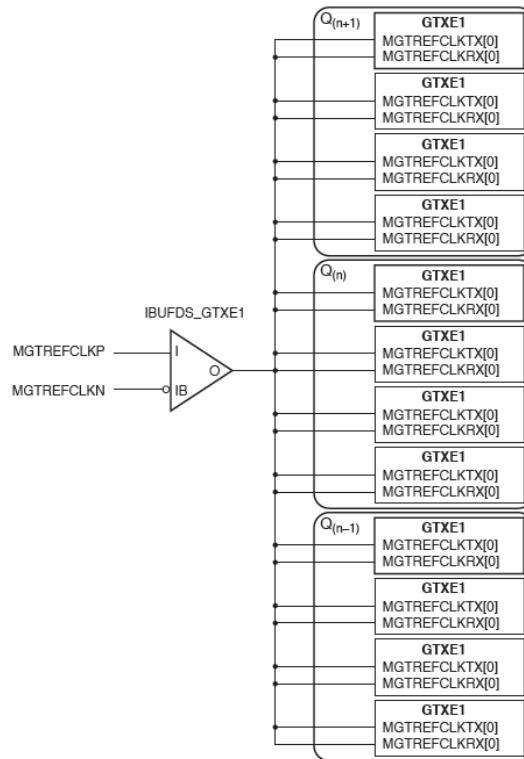


Figure 2-10 Multiple GTX with shared reference clock

The table below lists the various clock inputs and frequencies available to the GTX of the Perseus, according to the various high-speed serial ports. The FClkA label represents the frequency of the clock supplied by the backplane and is typically 100 MHz. The Fmc0 and Fmc1 labels represent the frequency of the clocks supplied to the Perseus from the FMC interface, which is, obviously, implementation specific.

Quad	GTX CX	GTX add.	MgtRefClk			Frequency	
			RefClk0_Q1	RefClk0_Q2	RefClk0_Q4	FClkA	Fmc0
112	Amc17	GTXE1_X0Y0	N/A	RefClk0_Q2	RefClk0_Q4	100 MHz	125 MHz
	Amc18	GTXE1_X0Y1					N/A
	Amc19	GTXE1_X0Y2					N/A
	Amc20	GTXE1_X0Y3					N/A
113	Amc0	GTXE1_X0Y4	N/A	RefClk0_Q2	RefClk0_Q4	125 MHz	156.25 MHz
	Amc1	GTXE1_X0Y5					N/A
	Amc2	GTXE1_X0Y6					N/A
	Amc3	GTXE1_X0Y7					N/A
114	Amc4	GTXE1_X0Y8	RefClk0_Q3	RefClk0_Q5	RefClk0_Q4	125 MHz	156.25 MHz
	Amc5	GTXE1_X0Y9					N/A
	Amc6	GTXE1_X0Y10					N/A
	Amc7	GTXE1_X0Y11					N/A
115	Amc8	GTXE1_X0Y12	N/A	RefClk0_Q5	RefClk0_Q4	125 MHz	156.25 MHz
	Amc9	GTXE1_X0Y13					N/A
	Amc10	GTXE1_X0Y14					N/A
	Amc11	GTXE1_X0Y15					N/A
116	Fmc0	GTXE1_X0Y16	N/A	RefClk0_Q5	RefClk0_Q4	125 MHz	156.25 MHz
	Fmc1	GTXE1_X0Y17					N/A
	Fmc2	GTXE1_X0Y18					N/A
	Fmc3	GTXE1_X0Y19					N/A
117	Fmc4	GTXE1_X0Y20	N/A	RefClk0_Q5	RefClk0_Q4	125 MHz	156.25 MHz
	Fmc5	GTXE1_X0Y21					N/A
	Fmc6	GTXE1_X0Y22					N/A
	Fmc7	GTXE1_X0Y23					N/A

Table 7 AMC and FMC port map (LX240 and SX315 Virtex-6)

Quad	GTX CX	GTX add.	MgtRefClk		Frequency	
			N/A	N/A	125 MHz	Fmc0
112	Amc17	GTXE1_X0Y8	RefClk0_Q1	RefClk0_Q2	100 MHz	FciKA
	Amc18	GTXE1_X0Y9				
	Amc19	GTXE1_X0Y10				
	Amc20	GTXE1_X0Y11				
113	Amc0	GTXE1_X0Y12	RefClk1_Q2	RefClk1_Q3	125 MHz	Fmc1
	Amc1	GTXE1_X0Y13				
	Amc2	GTXE1_X0Y14				
	Amc3	GTXE1_X0Y15				
114	Amc4	GTXE1_X0Y16	RefClk0_Q4	RefClk0_Q5	156.25 MHz	N/A
	Amc5	GTXE1_X0Y17				
	Amc6	GTXE1_X0Y18				
	Amc7	GTXE1_X0Y19				
115	Amc8	GTXE1_X0Y20	RefClk1_Q3	N/A	125 MHz	N/A
	Amc9	GTXE1_X0Y21				
	Amc10	GTXE1_X0Y22				
	Amc11	GTXE1_X0Y23				
116	Fmc0	GTXE1_X0Y24	RefClk0_Q3	N/A	100 MHz	Fmc0
	Fmc1	GTXE1_X0Y25				
	Fmc2	GTXE1_X0Y26				
	Fmc3	GTXE1_X0Y27				
117	Fmc4	GTXE1_X0Y28	RefClk1_Q4	N/A	125 MHz	Fmc1
	Fmc5	GTXE1_X0Y29				
	Fmc6	GTXE1_X0Y30				
	Fmc7	GTXE1_X0Y31				

Table 8 AMC and FMC port map (LX550 and SX475 Virtex-6)

RX/TX PLLs

Among other features, each GTX comprises an RX PLL and a TX PLL that can generate numerous data rates from the reference clock. The figure below illustrates how each PLL can be sourced by the two MgtRefClk of its own quad, as well as by those of the two adjacent quads.

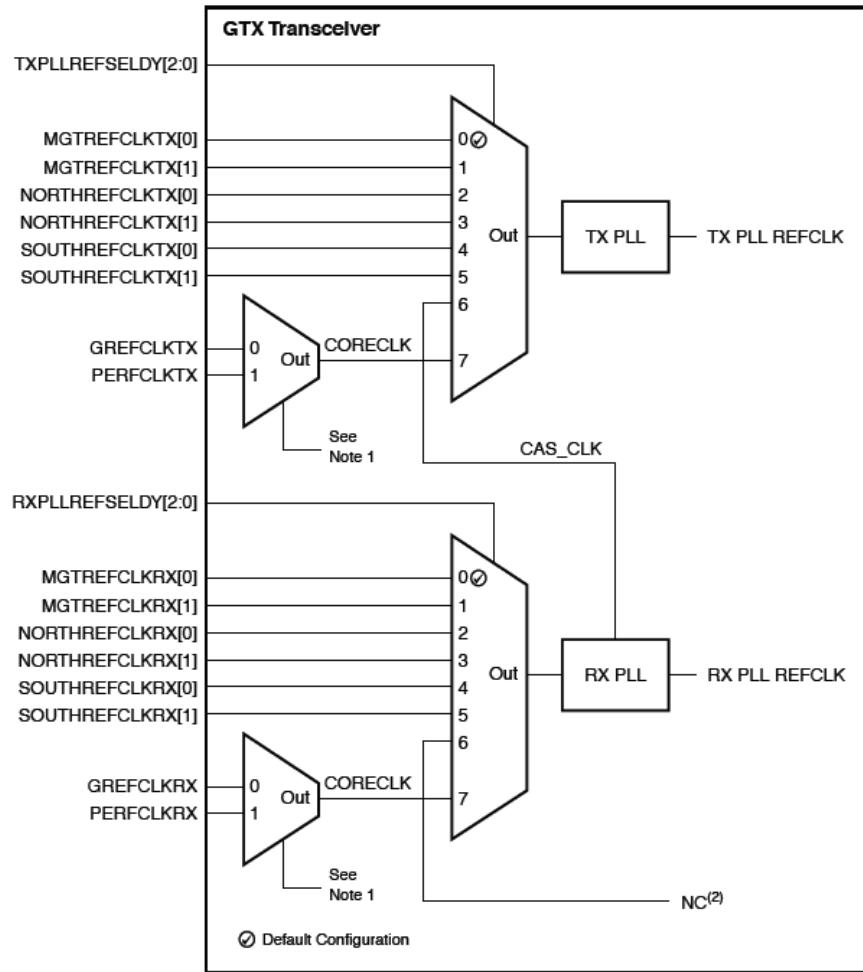


Figure 2-11 GTX detailed diagram

When you use the same clock in RX and TX, you can share the RX PLL and turn off the TX PLL to reduce power consumption.

Equalization

To ensure good data reception by the GTX, its RX section is equipped with an equalizer. You can tune this equalizer so that it fits the operating frequency of the GTX while more or less cutting the lower frequency portion of the spectrum. Available equalizer settings are illustrated here:

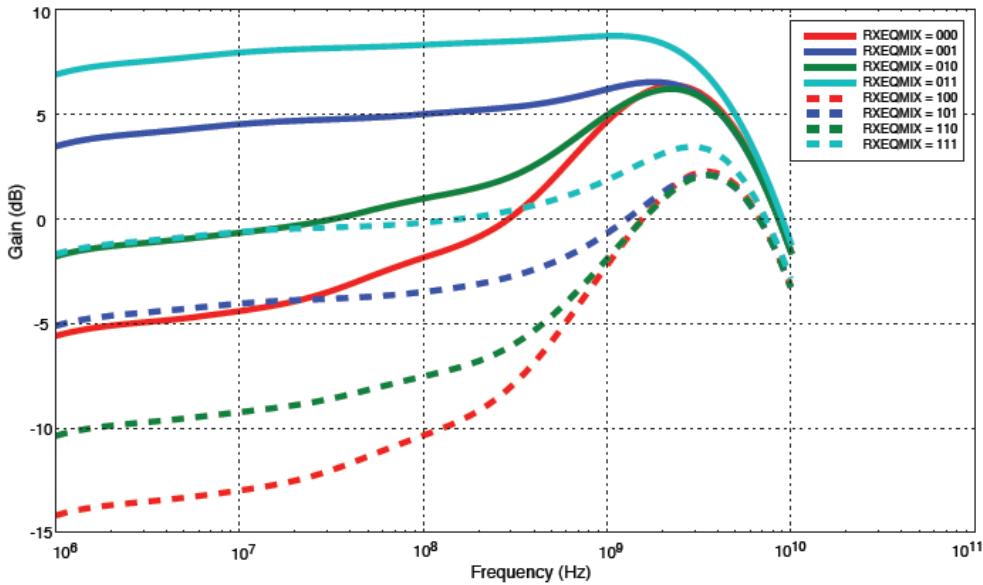


Figure 2-12 Absolute gain (voltage transfer function)

Serial, 16-bit D/A converter for reference TCVCXO

The onboard reference clock can be fine tuned by this small serial D/A. It is based on the Linear technologies LTC2641, 16 bit version. The D/A is controlled by its own SPI bus. The interface supports only write operations and consists of 3 wires: CS(chip select), DIN(data) and SCLK(clock). The serial interface can support a maximum frequency of 50MHz. The diagram below illustrates the timing requirement for a write operation.

When used with the proper FPGA core, this serial interface can be used to slave the reference clock to a GPS clock. Consult Nutaq on the availability of this core.

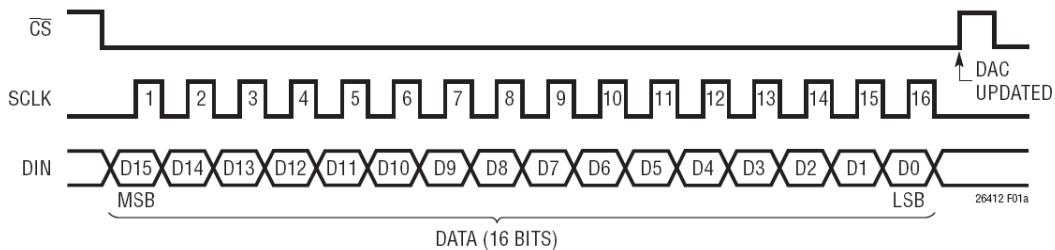


Figure 2-13 LTC2641 serial timing diagram

Perseus interfaces

The GTX on the backplane support several communications standards such as GigE, PCIe, XAUI, SATA, and SRIO. Each standard already has a prescribed position in the port map of the backplane, as described in the AdvancedMC standard. The generic nature of the GTX, however, also allows you to use them for other purposes (custom or not), as long as you do not need interoperability with other systems on these ports. For example, the Xilinx Virtex-6 Embedded Tri-Mode Ethernet MAC Wrapper core supports typical Ethernet data rates, but it also supports overclocking at rates up to 2.5 Gbps. In other words, using two Perseus on a backplane that supports 2.5 Gbps, you can leverage transferring data over Ethernet at twice the standard speed.

From the FMC standpoint, the GTX have no protocols, which is consistent with the GTX philosophy. The FMC interface is dynamic — it is designed to support a vast variety of I/O standards, voltages, directions, differential/single-ended combinations, within limits.

The table below presents the FPGA pin assignments of each GTX according to their quads and channels. This table is for informational purposes only; the pin assignments are available and ready to use in a UCF file supplied with the Perseus.

Quad	Mode	Channel	Pin (P/N)
112	RX	0	AN5/AN6
		1	AM7/AM8
		2	AL5/AL6
		3	AJ5/AJ6
	TX	0	AP3/AP4
		1	AN1/AN2
		2	AM3/AM4
		3	AL1/AL2
113	RX	0	AG5/AG6
		1	AF3/AF4
		2	AE5/AE6
		3	AD3/AD4
	TX	0	AK3/AK4
		1	AJ1/AJ2
		2	AH3/AH4
		3	AG1/AG2
114	RX	0	AC5/AC6
		1	AB3/AB4
		2	AA5/AA6
		3	Y3/Y4
	TX	0	AE1/AE2
		1	AC1/AC2
		2	AA1/AA2
		3	W1/W2
115	RX	0	W5/W6
		1	V3/V4

Quad	Mode	Channel	Pin (P/N)
116	TX	2	U5/U6
		3	R5/R6
	RX	0	U1/U2
		1	T3/T4
		2	R1/R2
		3	P3/P4
	TX	0	P7/P8
		1	N5/N6
		2	L5/L6
		3	J5/J6
		0	N1/N2
117	RX	1	M3/M4
		2	L1/L2
		3	K3/K4
		0	H7/H8
	TX	1	G5/G6
		2	F7/F8
		3	E5/E6
		0	J1/J2
		1	H3/H4
		2	G1/G2
		3	F3/F4

Table 9 LVDS buffer behavior

For details about GTX, refer to the Virtex-6 FPGA GTX user's guide available from the Xilinx Web site at www.xilinx.com.

2.4 Perseus Software Architecture

2.4.1 Distributed Computing System

Even when located in a single chassis, a Perseus running under Nutaq µTCA edition software tools can be viewed as a distributed computing system composed of several computing elements connected to each other through communication channels.

Communication channels

Available communication channels are:

- µTCA: the µTCA standard defines several fabrics such as Gigabit Ethernet, PCI Express, serial rapid I/O, and serial ATA.
- FMC: the FMC standard offers a means to communicate with more subsystems.
- Other channels: several channels can be supplied by µTCA cards through external connectors, such as Ethernet, EIA-232, and USB.

Computing elements

Different types of computing elements can be connected to this software system:

- Remote x86 host: a standard x86 computer can be connected to the µTCA system. Such a host device generally runs on Windows or Linux and is connected to the system through Ethernet.
- µTCA x86 host: an x86 host device can also take the form of a µTCA card. Such a host device usually runs on Windows or Linux, and is connected through the µTCA GigE or PCIe fabrics.
- µTCA PowerPC host: a PowerPC host device can take the form of a µTCA card. Such a host device usually runs on Linux and is connected through the µTCA SRIO, GigE, or PCIe fabrics.
- Perseus with MicroBlaze and central communication engine (CCE): the Perseus usually runs a MicroBlaze software processor running Linux and the Nutaq CCE.
- Perseus without MicroBlaze, but with other Nutaq cores: a Perseus can also use usual Nutaq cores (such as RTDEx) without depending on a MicroBlaze, Linux, CCE subsystem.
- Perseus custom: a Perseus can be used without any Nutaq software tools cores.
- Other subsystems: other computing elements can be part of this software system.

2.4.2 Command-Line Interface

The command-line interface (CLI) is a basic client interface for the Perseus. It runs on a host device or directly on the Perseus. It consists of a shell where you can type commands, interacting with the different computing elements connected to your system. The CLI offers many useful features:

- Programming an FPGA bitstream in the onboard flash memory. This allows configuring the FPGA during the startup sequence.
- Reading and writing at specified addresses on the AXI.
- Loading data at specified addresses in the Perseus DDR3 SDRAM.
- Configuring and controlling the ADAC250, MI250, Radio420, MI125, MO1000 and LVDS FMC modules.
- RTDEx
- Recording and playback.
- Running batch files.

Requirements

- Python 2.7 (supplied)
- Ethernet connectivity to the Perseus

CLI operation

The functionalities of a Perseus system can be highly customized and the standard software evolve over time. The command-line interface (or CLI, for short), a light interface that helps debug and configure the FPGA cores of various peripherals of the Perseus, reflects this by incorporating various command sets.

Please refer to the *Command Line Interface Programmer's Reference Guide.pdf*.

2.4.3 Perseus Software Platform

The board software development kit of the Perseus contains a software stack that allows remote controlling the Perseus and its component through a GigE connection.

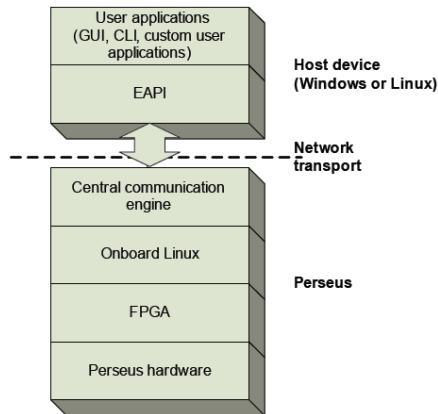


Figure 2-14 Software platform

Software libraries

Several application programming interfaces (APIs) are supplied to make programming and communicating with the Perseus easier. Through the help of the Perseus API, you can configure the card hardware and software for your specific needs.

3 Mechanical Assembly

3.1 Replacing Face Plates

Note:

Before you begin any of the mechanical procedures below, make sure that you have a set of precision screwdrivers handy.

Depending on your configuration (mid size or full size), you may have to replace the front plate of the Perseus, as it is shipped with the mid-size front plate installed.

3.1.1 Removing a Face Plate

Note:

The mid-size face plate and the full size face plate are removed in the same fashion.

To remove a face plate:

1. With a precision Phillips screwdriver, remove the two screws illustrated below.

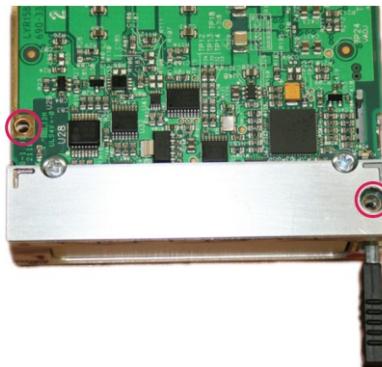


Figure 3-1 Removing screws

Important:

Make sure that the AMC handle is pulled fully outward before you proceed.

2. Gently pull the face plate and post assembly slightly toward you until it blocks.

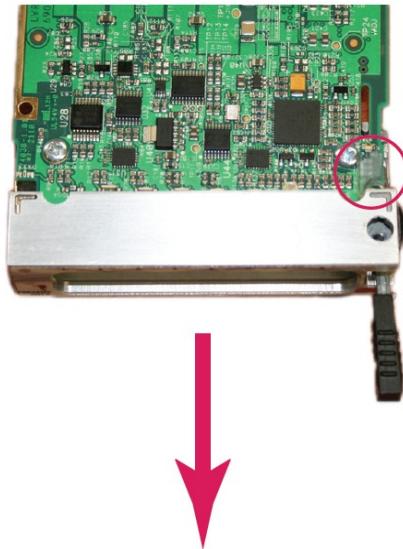


Figure 3-2 Pulling face plate outward

Important:

Make sure that you are careful when pulling the face plate as careless removal could result in damage to the LEDs under the Perseus. If you are removing a full-size face plate and that the Mestor expander is installed, make sure that you remove the two screws securing the expander to the front plate. See below for details.

3. Remove the LED pipe circled above. Place it in a safe place.
You will need this part again later.
4. Disengage the left portion of the assembly as illustrated.

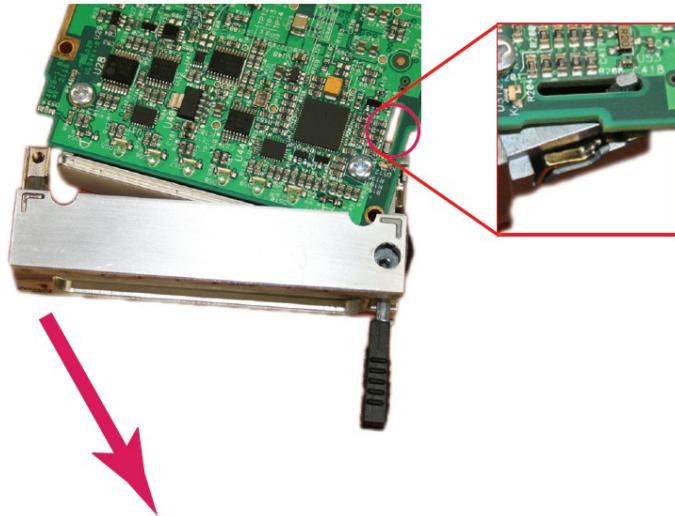


Figure 3-3 Disengaging the face plate assembly from the Perseus

5. Unhook the right portion of the face plate assembly from the groove in the Perseus, as illustrated.

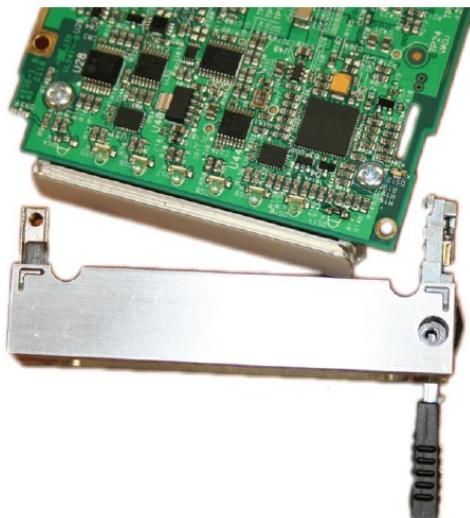


Figure 3-4 Unhooking the face plate assembly from the Perseus

3.1.2 Installing a Face Plate

Before installing a face plate once it is removed, you may want to read the procedures on installing FMCs and the Mestor, as you must install them prior to installing face plates on the Perseus.

Before you begin installing a face plate, make sure that you have the following parts at hand:

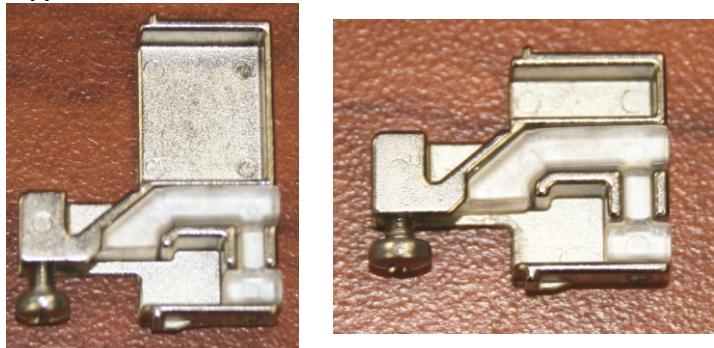
LED pipe



AMC handles



Supports



To install the face plate:

1. If not already so, disassemble the face plate-support-AMC handle assembly.
2. Remove any screws from the support and AMC handle, and then place them in a safe location for later use.
3. With the Perseus upside-down, locate the sliding groove on the card edge and place it in front of you.
4. Slide the AMC handle along the edge of the card until it is hooked in the sliding groove, as illustrated.

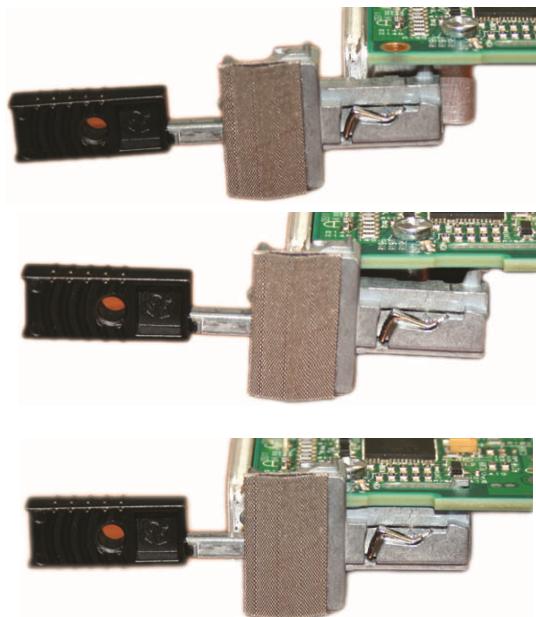


Figure 3-5 Installing an AMC handle

5. Move the AMC handle slightly outward.
6. Place the LED pipe on the Perseus and then align it with the screw hold, as illustrated.

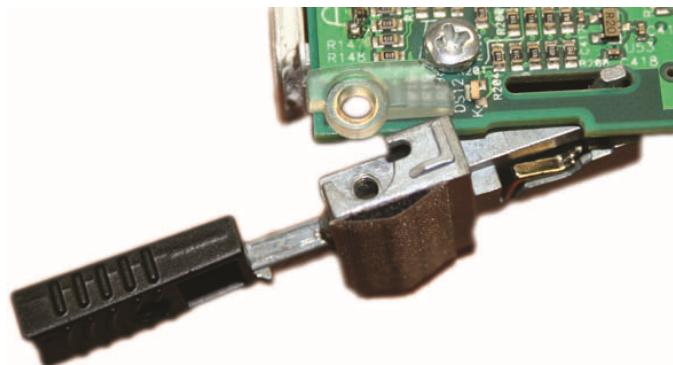


Figure 3-6 Installing the LED pipe

7. Holding the LED pipe in place, gently slide the AMC handle over the pipe until it is back in its original position.

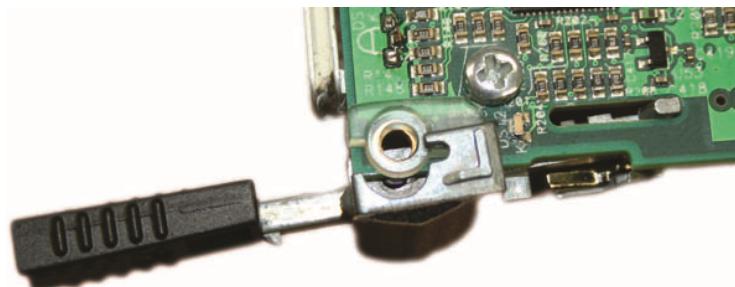


Figure 3-7 Aligning the AMC handle and LED pipe

8. Secure with the supplied screw.

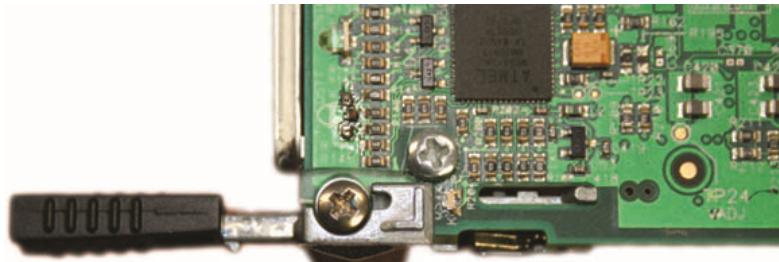


Figure 3-8 Secure assembly

9. Install the support on the opposite site of the Perseus, as illustrated.

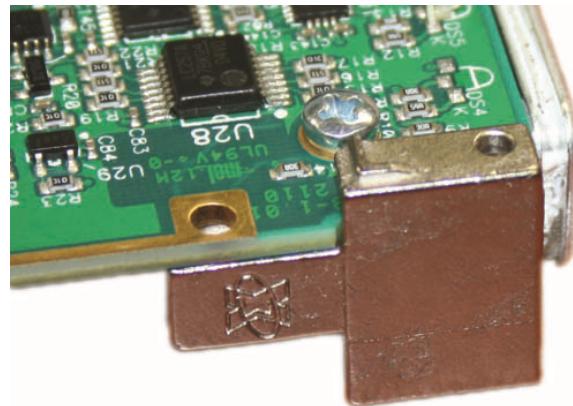


Figure 3-9 Installing the support

10. Secure the support on the card with the supplied screw.

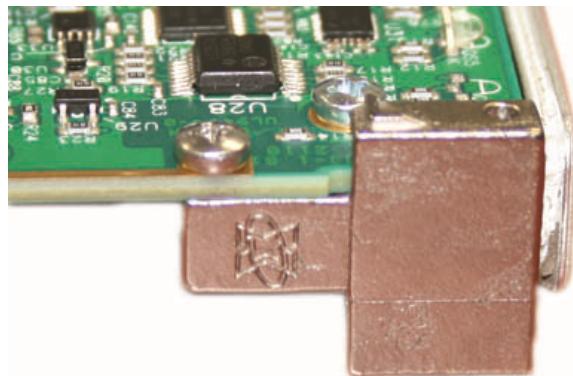


Figure 3-10 Securing the support

11. Turn the Perseus so that it faces you as illustrated.



Figure 3-11 Turning the Perseus

Note:

The Perseus is illustrated above with a cover plate installed. Depending on whether you have installed an FMC or the Mestor expander on your card, the cover plate could be absent or the cover plate of the FMC. Refer to the procedures below to install other modules on the Perseus before installing the face plate. The procedure to install the face plate remains, however, the same.

12. If not already so, make sure that the AMC handle is completely pull outward.

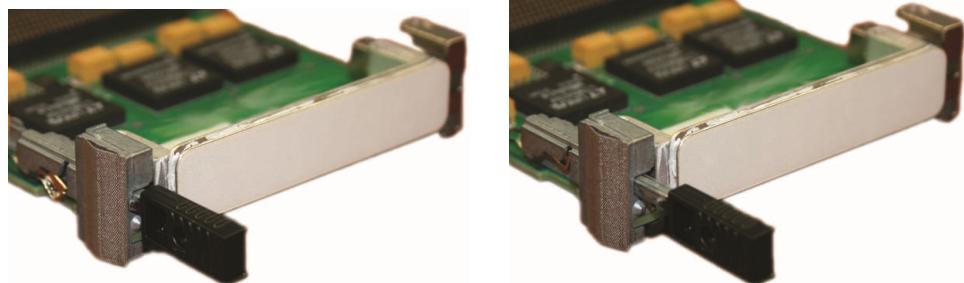


Figure 3-12 Pulling the AMC handle outward

13. Locate the face plate that you want to install (full size or half size), align it to the width of the Perseus and then slide it along the edge, as illustrated.



Figure 3-13 Aligning the face plate

14. Make sure that the AMC handle is properly aligned with the face plate.

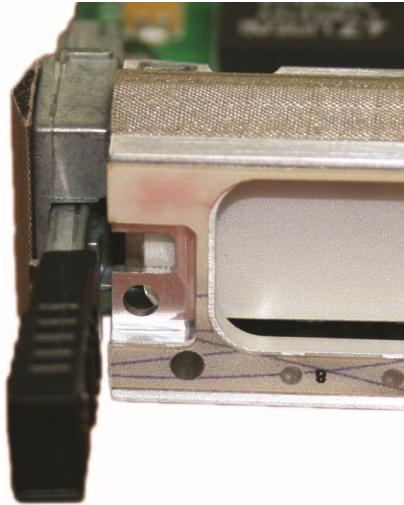


Figure 3-14 AMC handle and face plate

15. Turn the Perseus upside-down.
16. Carefully lift the inner edge of the face plate so that it overlaps the AMC handle screw, as illustrated.



Figure 3-15 Lifting the edge of the face plate

17. Gently push the face plate towards the back of the Perseus so that the slots in the plate slide into the grooves etched in the supports.
18. Repeat for the upper portion of the face plate.

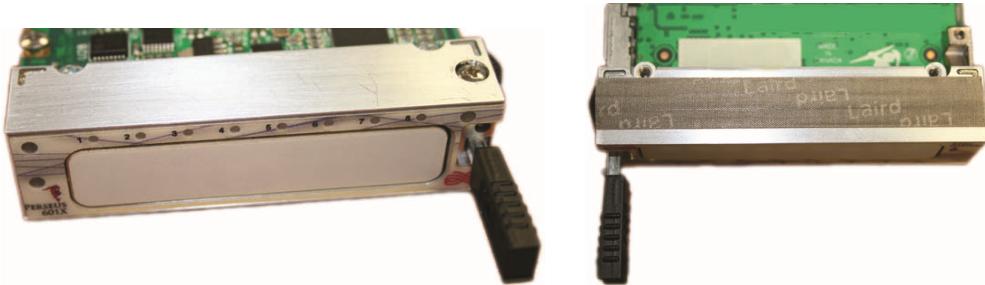


Figure 3-16 Locking the face plate into position

The face plate is now installed.

3.2 Installing FMCs

FMCs conforming to the VITA specifications are all installed in the same manner on the Perseus.

Note:

Make sure that you perform this procedure before you install the face plate on the Perseus. If your face plate is already on the Perseus, you must remove it before you can proceed. See the appropriate procedure above.

To install FMCs:

1. Remove the blank front plate from the Perseus.

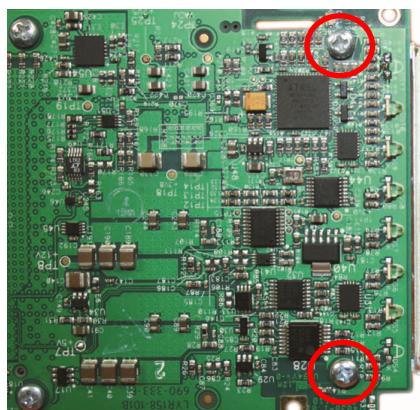


Figure 3-17 Removing the screws securing the blank front plate

2. Place your FMC one a flat surface and remove the four screws indicated in the figure below.

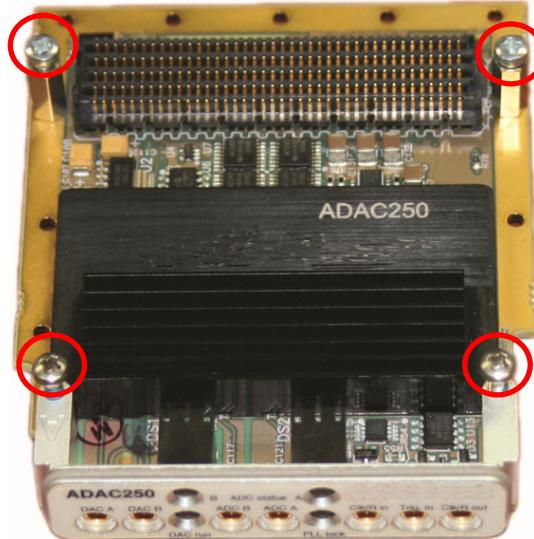


Figure 3-18 ADAC250, as shipped

Note:

Make sure that you place the screws in a safe location, as you will need them later on.

3. Align the FMC connectors of the Perseus and the FMC, as illustrated.



Figure 3-19 Aligning Perseus and FMC

4. Gently press down on the FMC above the FMC connector until the two connectors mate.
5. Turn the Perseus-FMC assembly over and secure the FMC to the Perseus, as indicated.

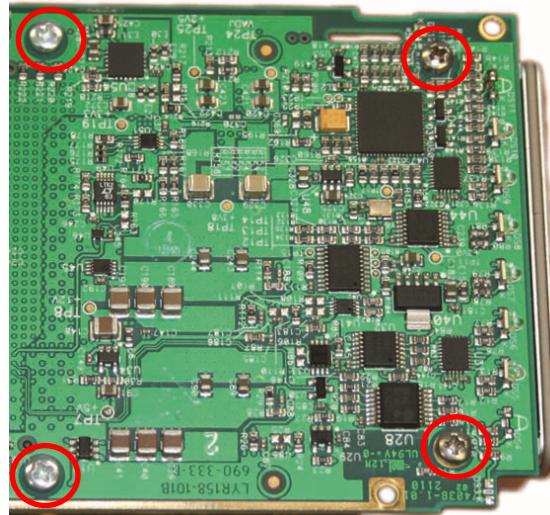


Figure 3-20 Securing the FMC on the Perseus

6. Install the mid-size face plate (see above) or proceed to install the Mestor expander (see below).

3.3 Installing the Mestor

3.3.1 Connecting a Mestor Adapter

To connect a Mestor adapter:

1. Locate the adapter that you want to connect.
There are two adapters: the FPGA JTAG adapter and the IPMI JTAG adapter.
2. Locate the expansion connector directly next to the FPGA heatsink.
3. Align the adapter with the connector.
Make sure that the adapter is correctly oriented.

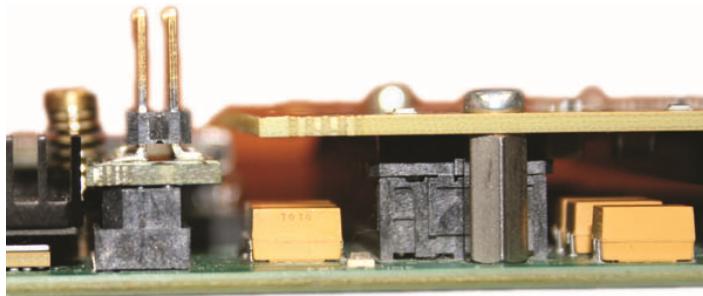


Figure 3-21 Mating connectors

4. Carefully press down on the adapter so that it securely mates with the connector on the Perseus.

3.3.2 Installing the Mestor Expander

The installation can be done with or without an FMC.

To install the Mestor expander with an FMC:

1. Locate the four supplied cylindrical metal washers.
2. Remove the four screws securing your FMC to the Perseus and place them in a safe place.
3. Place the washers as indicated below.

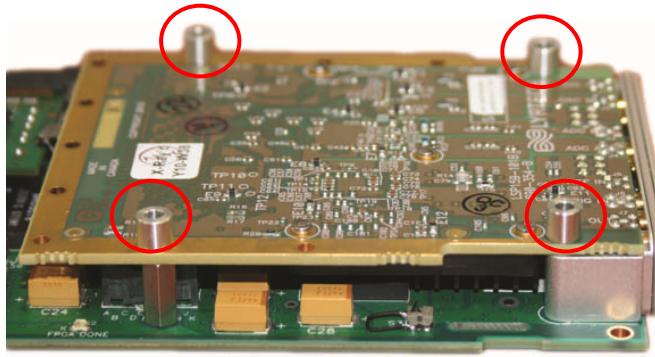


Figure 3-22 Placing the washers on the FMC

4. Mate the expander QTH connector with the Mestor interface connector on the Perseus.
5. Gently lower the Mestor expander so that its four screw holes are aligned with the washers.

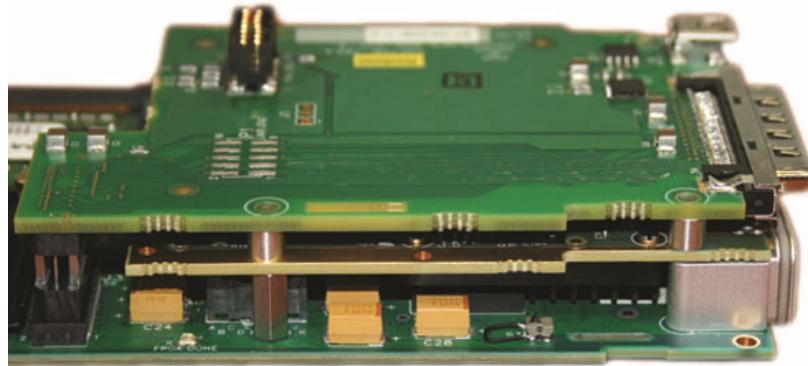


Figure 3-23 Mating the Mestor expander and the Perseus

6. Press down over the Mestor connector to mate the two connectors securely.
7. Place the supplied M2.5×14 mm Phillips screws in the four free screw holes.

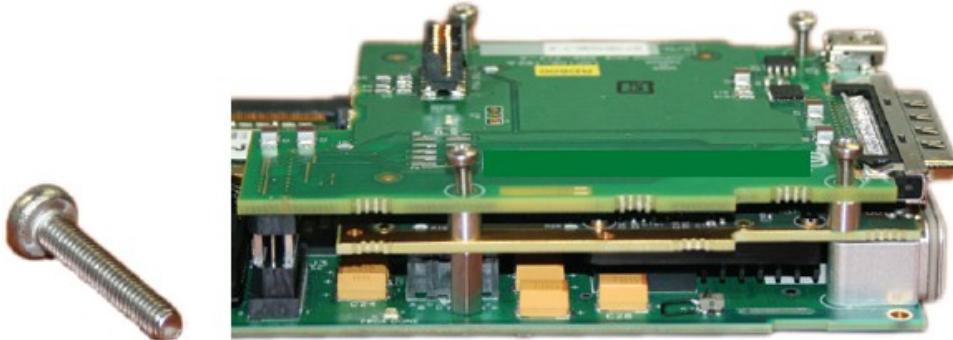


Figure 3-24 Screws and positions

8. Secure the Mestor expander in place.

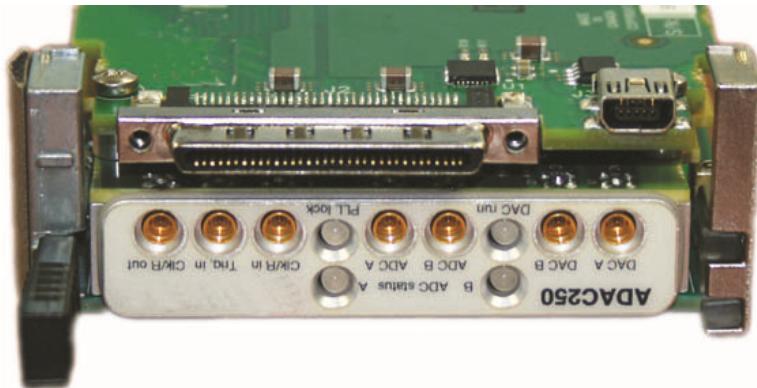


Figure 3-25 Mestor expander secured over an ADAC250 FMC

9. Install the full-size face plate as described above.

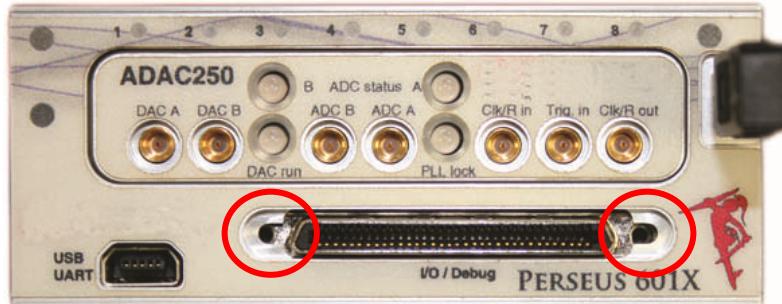


Figure 3-26 Full size face plate over Mestor expander

10. Once the face plate is secured to the Perseus, use the two supplied latching posts (circled red) to secure the
11. Mestor expander to the face plate.

To install the Mestor expander without an FMC:

1. Locate the four supplied cylindrical rubber washers and the four cylindrical metal washers.
2. Install the blank front plate on the Perseus.

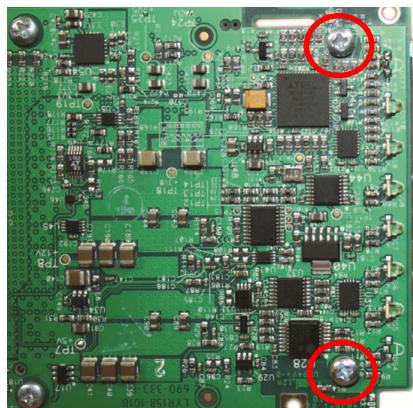


Figure 3-27 Securing the blank front plate to the Perseus

3. Place the rubber washers as indicated below.

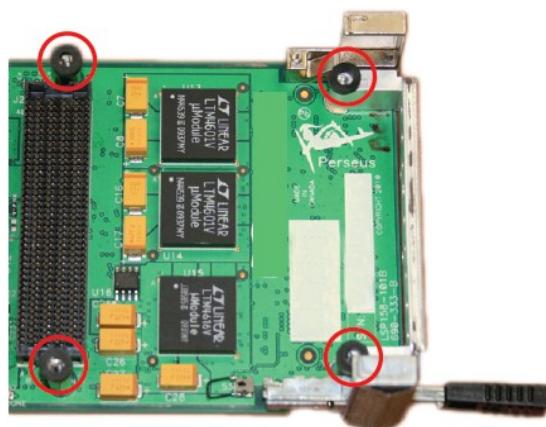


Figure 3-28 Placing the rubber washers

4. Atop the four rubber washers, place the four metal washers.

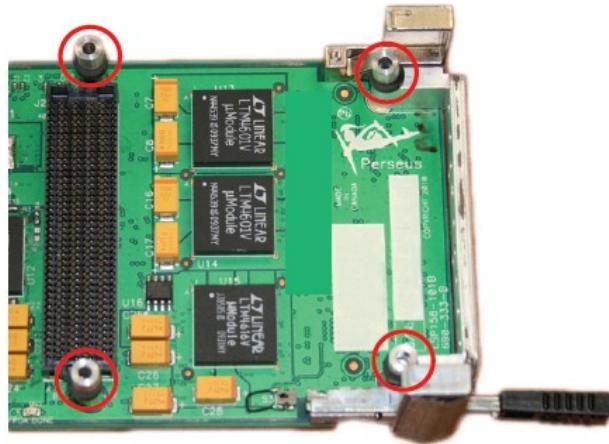


Figure 3-29 Placing the metal washers

5. Mate the expander QTH connector with the Mestor interface connector on the Perseus.
6. Gently lower the Mestor expander so that its four screw holes are lined with the washers.

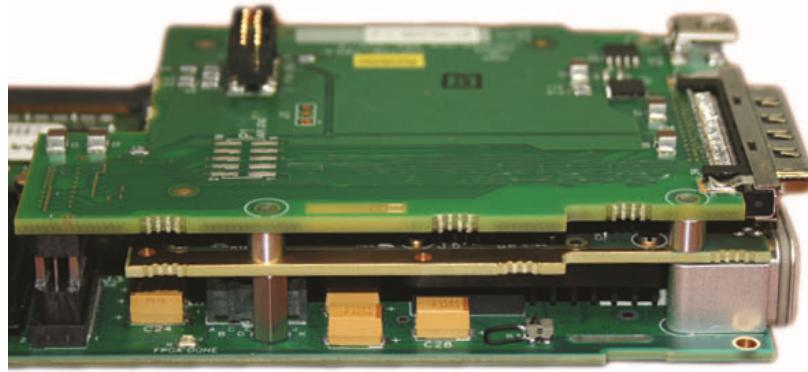


Figure 3-30 Mating the Mestor expander and the Perseus

7. Press down over the Mestor connector to mate the two connectors securely.
8. Place the supplied M2.5×14 mm Phillips screws in the four free screw holes.

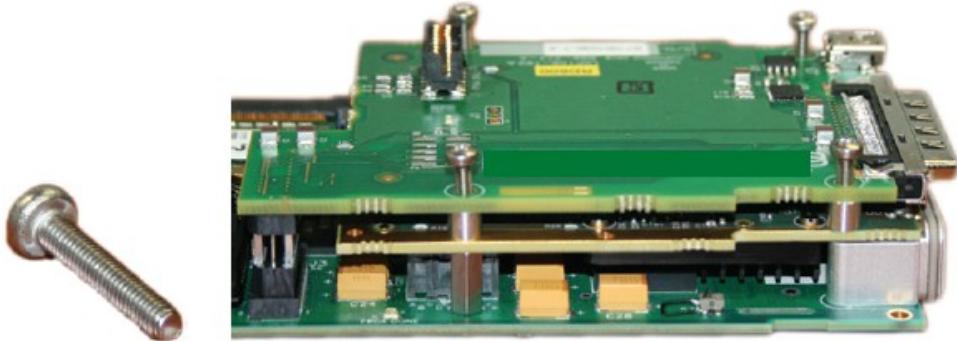


Figure 3-31 Screws and positions

9. Secure the Mestor expander in place.
10. Install the full-size face plate as described above.
11. Once the face plate is secured to the Perseus, use the two supplied latching posts to secure the Mestor expander to the face plate.

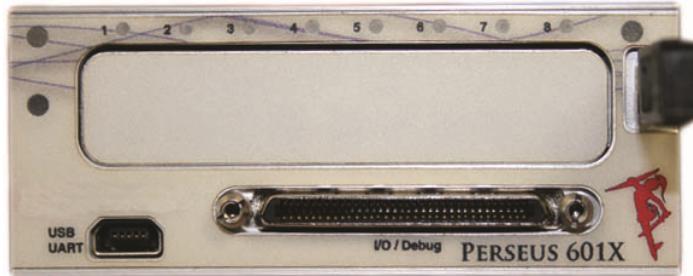


Figure 3-32 Securing the Mestor expander to the face plate

3.3.3 Connecting a Mestor Breakout Box

To connect a Mestor breakout box:

1. Make sure that the Mestor expander and the full-size front plate are installed on the Perseus.
2. Connect and secure the supplied cable to the Mestor expander.

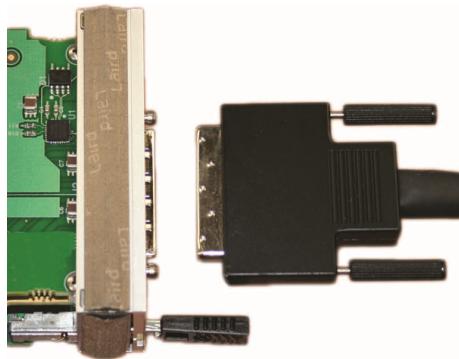


Figure 3-33 Connecting and securing the breakout box cable to the Perseus

3. Connect and secure the cable to the breakout box.



Figure 3-34 Connecting and securing the breakout box cable to the breakout box

Note:

Take care to secure the connector. If you do not, the cable is likely to disconnect itself.

3.4 Replacing the DDR3 SODIMM

3.4.1 Removing the DDR3 SODIMM

To remove the DDR3 SODIMM:

1. Remove the Mestor expander and any FMC from the Perseus.
2. Gently bend the supports on each side of the DDR3 SODIMM outward until it pops upward.

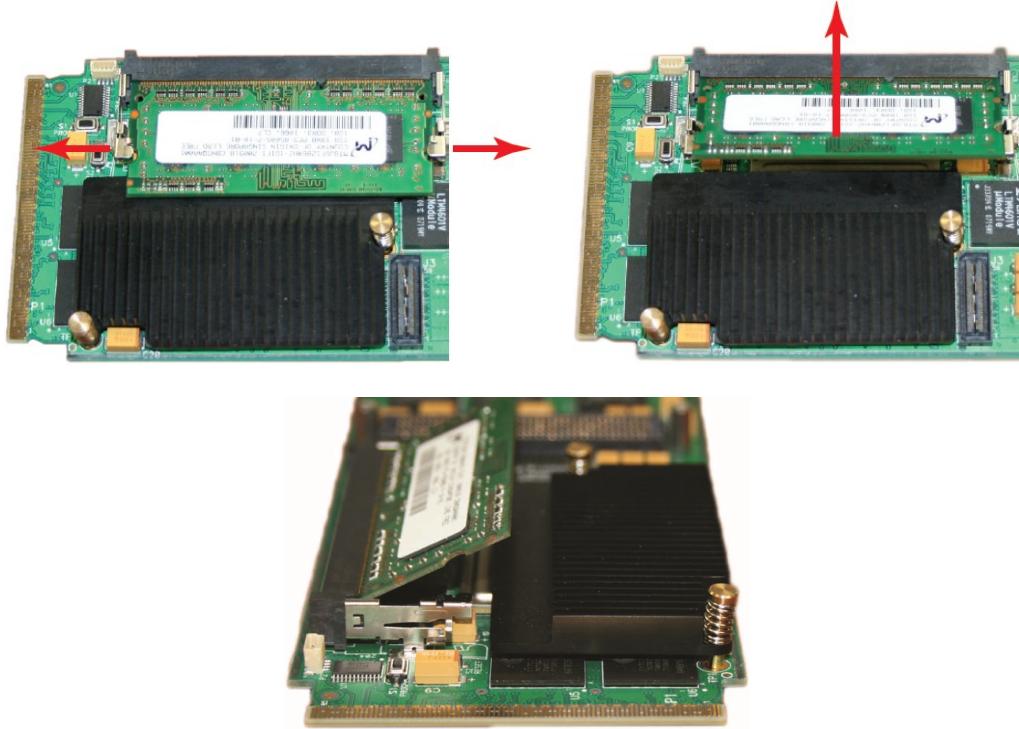


Figure 3-35 Releasing the DDR3 SODIMM

3. To remove the DDR3 SODIMM, gently pull it out of its connector.

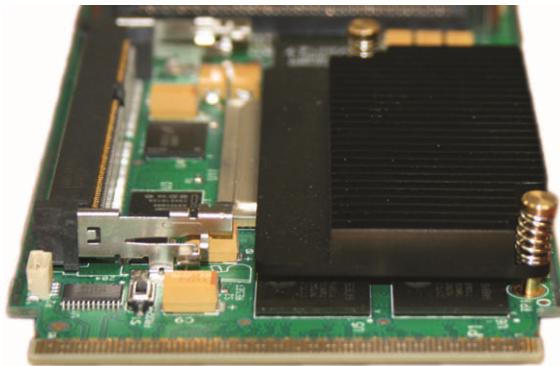


Figure 3-36 Removing the SODIMM

3.4.2 Inserting the DDR3 SODIMM

To insert the DDR3 SODIMM:

1. Insert the DDR3 SODIMM at an angle into the connector.



Figure 3-37 Inserting the SODIMM in its connector

2. Gently press down on the DDR3 SODIMM until you hear it click into place.

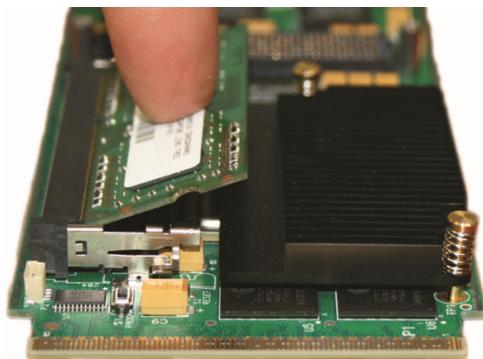


Figure 3-38 Securing the DDR3 SODIMM in place

3.5 Perseus in a μTCA Chassis

This section is supplied strictly for your information. For more detailed instructions, refer to the documentation accompanying your μTCA chassis.

3.5.1 Inserting a Perseus in a μTCA Chassis

To insert the Perseus:

1. Locate an open slot in your μTCA chassis.

Note:

If you are using the Perseus in its full size configuration, make sure that you have enough space to insert it in the chassis.



Figure 3-39 Free μTCA chassis slot

2. Pull the AMC handle completely outward.

3. Slide the Perseus into the free slot.

When the Perseus is correctly inserted, the AMC state LED lights.



Figure 3-40 Inserting the Perseus into the chassis

4. Push the AMC handle.

The AMC state LED starts blinking.

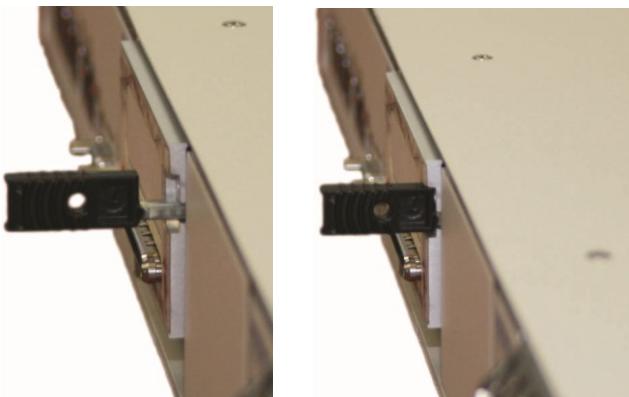


Figure 3-41 Pushing in AMC handle

When the AMC state LED goes out, the Perseus is running properly. The power LED is also on.

3.5.2 Removing a Perseus from a μTCA Chassis

To remove the Perseus:

1. Complete your work on the Perseus and disconnect any programs communicating with it.
2. Pull out the AMC handle.
The AMC state LED starts blinking.
3. When the AMC state LED stops blinking, remove the Perseus from the chassis.

4 Mestor Connection

The Mestor is the debugging connection for the Perseus 601X. It is a unique, onboard interface that regroups standard debugging ports and other interfaces. The Mestor interface offers access to:

- FPGA JTAG
- IPMI JTAG (typically used to debug the Atmel AVR MCU)
- FPGA UART (serial TX and RX)
- User LVDS I/Os (differential data × 14, differential clock × 1)
- 2.5 V and 3.3 V power (used to supply Mestor breakout boxes)

Mestor interface

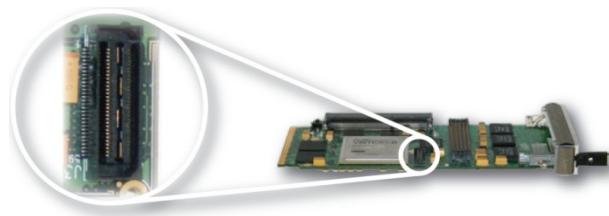


Figure 4-1 Mestor connector

The following table presents the Mestor interface pin assignment.

Pin	Assignment	Pin	Assignment
1	DPIO_0_P	2	DPIO_10_P
3	DPIO_0_N	4	DPIO_10_N
5	GND	6	GND
7	DPIO_1_P	8	DPIO_11_P
9	DPIO_1_N	10	DPIO_11_N
11	+2.5V	12	GND
13	DPIO_2_P	14	DPIO_12_P
15	DPIO_2_N	16	DPIO_12_N
17	GND	18	GND
19	DPIO_3_P	20	DPIO_13_P

Pin	Assignment	Pin	Assignment
21	DPIO_3_N	22	DPIO_13_N
23	+2.5V	24	GND
25	DPIO_4_P	26	DPIO_CLK_P
27	DPIO_4_N	28	DPIO_CLK_N
29	GND	30	JTAG_PRSNT_N
31	DPIO_5_P	32	SERIAL_RX
33	DPIO_5_N	34	SERIAL_TX
35	+2.5V	36	+3.3V
37	DPIO_6_P	38	+3.3V
39	DPIO_6_N	40	+3.3V
41	GND	42	AVR_TDI
43	DPIO_7_P	44	FPGA_TDI
45	DPIO_7_N	46	AVR_TMS
47	+2.5V	48	FPGA_TMS
49	DPIO_8_P	50	AVR_TDO
51	DPIO_8_N	52	FPGA_TDO
53	GND	54	AVR_TCK
55	DPIO_9_P	56	FPGA_TCK
57	DPIO_9_N	58	AVR_TRST
59	Daughter_prsnt_n	60	FPGA_TRST
61	GND	62	GND
63	GND	64	GND

Table 10 Mestor interface pin assignments of the Perseus board

Note:

The DPIO name convention of the Mestor Expander board does not follow the same convention than the Mestor DPIO name of the Perseus board (See Table 12).

Mestor JTAG adapters

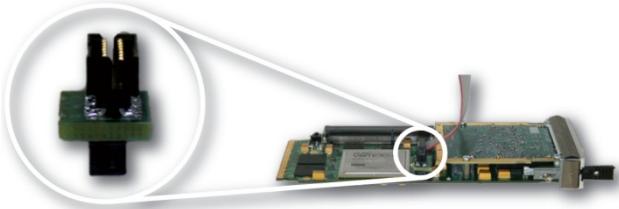


Figure 4-2 Mestor JTAG adapter

To access the Perseus' JTAG chains, you can use the Mestor JTAG adapters (LSP158-901) that includes a Mestor-to-FPGA JTAG adapter and a Mestor-to-AVR JTAG adapter. Connected to a Xilinx platform cable, you can easily troubleshoot your FPGA software through the FPGA JTAG chain with the Mestor-to-FPGA adapter. With an Atmel AVR MCU and the Mestor-to-AVR adapter, you can as easily troubleshoot the IPMI stack of the Perseus.

Mestor expander



Figure 4-3 Mestor expander

The Mestor expander saves you from having to free μTCA slots or to open the chassis shelf to access debugging interfaces. The kit includes the necessary front panel cables.

The Mestor expander offers two front panel connections:

- FPGA UART (mini-AB USB). This connector maps the FPGA UART pins through a USB-to-UART bridge (CP2110 from Silicon Labs).
- Mestor debugging I/Os (VHDCI connector), regrouping IPMI and FPGA JTAGs, user LVDS I/Os and power. This connector allows you or Nutaq to expand the capabilities of the Perseus. For example, Nutaq offers the breakout boxes to connect its RF front ends to the card.

Mestor breakout box 1



Figure 4-4 Mestor breakout box

For individual access to each port of the Mestor debugging I/Os, Nutaq offers a breakout box interfacing directly with the I/Os through self-powered VHDCI cables. In addition, the breakout box makes Nutaq's line of RF front ends compatible with the Perseus.

The following I/Os are available through Mestor breakout box 1:

- Xilinx FPGA JTAG connector
- Atmel AVR MCU IPMI JTAG connector
- Nutaq GPIO-32 interface (for LVC MOS user I/Os or to interface with external Nutaq RF front ends)
- Four-channel mux A/Ds (1 MSPS, 12 bits; 250 KSPS per channel, if using four ADCs)

4.1 Hardware Description

The Mestor is composed of the following components.

4.1.1 Mestor Expander

When the Mestor expander and the appropriate front panel are installed on your Perseus, you can connect:

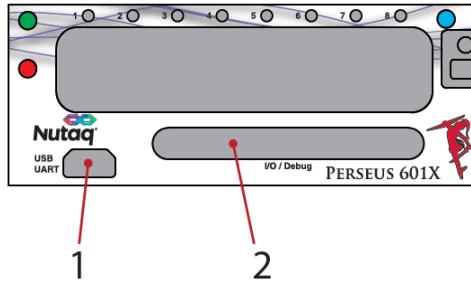


Figure 4-5 Mestor expander connectivity

USB/UART port

This port is used as a USB-to-UART bridge for your remote computer. It maps the FPGA UART pins through a USB-to-UART bridge (Silicon Labs CP2110). If your computer does not automatically detect the device, visit www.silabs.com to update your bridge driver.

I/O / Debug port

Use this port to connect a Mestor breakout box, which offers several debugging capabilities, or a custom I/O board. VHDCI connector to the Perseus I/O expansion connector. The following table introduces the connector pin assignments.

Pin	Assignment	Pin	Assignment
1	DPIO_CLK_P	35	DPIO_CLK_N
2	GND	36	GND
3	DPIO_0_P	37	DPIO_0_N
4	+2V5	38	+2V5
5	DPIO_1_P	39	DPIO_1_N

Pin	Assignment	Pin	Assignment
6	+2V5	40	+2V5
7	DPIO_2_P	41	DPIO_2_N
8	GND	42	GND
9	DPIO_3_P	43	DPIO_3_N
10	GND	44	GND
11	DPIO_4_P	45	DPIO_4_N
12	GND	46	GND
13	DPIO_5_P	47	DPIO_5_N
14	GND	48	GND
15	DPIO_6_P	49	DPIO_6_N
16	GND	50	GND
17	DPIO_7_P	51	DPIO_7_N
18	DAUGTHER_PRSNT_N	52	JTAG_PRSNT_N
19	DPIO_8_P	53	DPIO_8_N
20	GND	54	GND
21	DPIO_9_P	55	DPIO_9_N
22	+3V3	56	+3V3
23	DPIO_10_P	57	DPIO_10_N
24	+3V3	58	+3V3
25	DPIO_11_P	59	DPIO_11_N
26	SERIAL_TX	60	SERIAL_RX
27	DPIO_12_P	61	DPIO_12_N
28	GND	62	GND
29	DPIO_13_P	63	DPIO_13_N
30	NC	64	AVR_RST
31	FPGA_TCK	65	AVR_TCK

Pin	Assignment	Pin	Assignment
32	FPGA_TDO	66	AVR_TDO
33	FPGA_TMS	67	AVR_TMS
34	FPGA_TDI	68	AVR_TDI

Table 11 Pin assignments**Note:**

The DPIO naming convention of the Mestor expander card does not follow the same convention than the Mestor DPIO names of the Perseus board (See Table 12).

When connected to a Perseus board, the naming convention change when DPIO signal cross the Mestor connector. The table 12 shows the name assignment between the Mestor Expander DPIO names and the Perseus Mestor names.

Mestor Expander Pin Name	Perseus Mestor Pin Name
DPIO 0	DPIO 9
DPIO 1	DPIO 8
DPIO 2	DPIO 7
DPIO 3	DPIO 6
DPIO 4	DPIO 5
DPIO 5	DPIO 4
DPIO 6	DPIO 3
DPIO 7	DPIO 2
DPIO 8	DPIO 1
DPIO 9	DPIO 0
DPIO 10	DPIO 13
DPIO 11	DPIO 12
DPIO 12	DPIO 11
DPIO 13	DPIO 10

Mestor Expander Pin Name	Perseus Mestor Pin Name
DPIO CLK	DPIO CLK

Table 12 DPIO naming convention between the Perseus board and the Mestor Expander

By using the Mestor Expander with a Perseus board, the following mapping between the VHDCI Mestor Expander connector pins and the Perseus FPGA pins can be established using Table 11, Table 12 and Table 24.

VHDCI Mestor Expander pins	Perseus FPGA pins	VHDCI Mestor Expander pins	Perseus FPGA pins
1	AP36	35	AP35
3	AN35	37	AN36
5	AP37	39	AR37
7	AR35	41	AT35
9	AU36	43	AT36
11	AT37	45	AR38
13	AU37	47	AU38
15	AV39	49	AV38
17	AW37	51	AW38
19	AY39	53	BA39
21	AY38	55	AY37
23	AW36	57	AV36
25	BB36	59	BA36
27	BB39	61	BB38
29	BA37	63	BB37

Table 13 Mapping between VHDCI Mestor Expander connector and Perseus FPGA DPIO

4.1.2 Mestor Breakout Box

Note:

The control of the Mestor Breakout Box ADCs and GPIOs from the Perseus is not supported in the Perseus Software Tools.

The Mestor breakout boxes can be represented thus:



Figure 4-6 Mestor breakout box

As you can see from the figure, the box is equipped with several direct links to the Perseus most useful I/Os. In the case of multi-pin interfaces, a dot marks the position of pin number one.

MBB program button

This button forces the onboard Spartan-3AN to reload itself from its onboard flash. Normally, you need to reprogram the breakout box FPGA.

MBB reset button

This button resets the Spartan-3AN internal logic. This will force the FPGA to re-align itself on the Perseus data stream.

JTAG FPGA connector

Gives access to the Perseus FPGA JTAG chain. The following table presents the pin assignments of the connector and is based on a standard Xilinx platform cable.

Pin	Assignment	Pin	Assignment
1	GND	2	VRef-3.3 V
3	GND	4	TMS
5	GND	6	TCK

Pin	Assignment	Pin	Assignment
7	GND	8	TDO
9	GND	10	TDI
11	GND	12	NC
13	PresGND	14	NC

Table 14 Pin assignments

The breakout box Spartan-3AN is added at the beginning of the JTAG chain when you connect it to the Perseus. This changes the JTAG debugging chain. To continue using the Perseus with the EDK, you must configure an XMD server: instead of using the **Launch XMD** command on the **Debug** menu, start the XMD program directly from a command prompt window, typing **connect mb mdm -debugdevice deviceNr 2**. This starts the XMD server and instructs it to connect the MicroBlaze to the second FPGA in the chain (the Perseus FPGA).

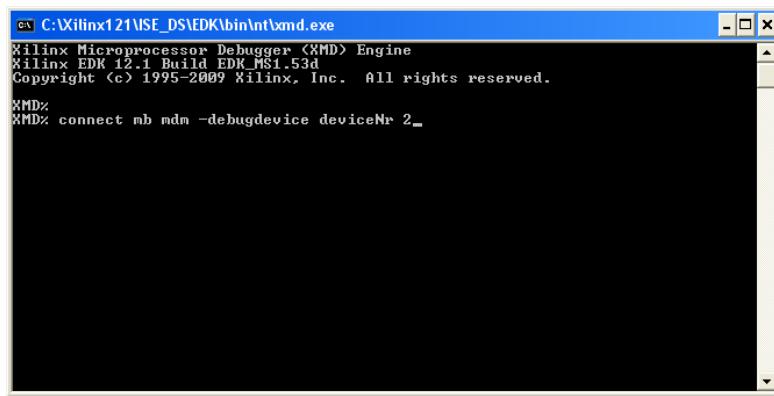


Figure 4-7 Connecting to the JTAG chain with the breakout box

JTAG IPMI connector

Gives access to the Perseus AVR JTAG chain. This allows you to debug your IPMI stack, if necessary. The following table introduces the connector pin assignments.

Pin	Assignment	Pin	Assignment
1	TCK	2	GND
3	TDO	4	3.3 V
5	TMS	6	Reste
7	3.3 V	8	NC
9	TDI	10	GND

Table 15 Pin assignments

ADC 1 to 4 connectors

These connector are linked to an Analog Devices AD7924BR A/D converter. This simple serial converter is capable of running at 1 MSPS. By enabling the four available channels, you can access 250 KSPS per 12-bit channel. You can

use these channels to sample an RF power monitor, for example. They are very low-cost channels, so performances such as the SNR or channel separation are limited. Use them for noncritical tasks.

GPIO-32 connector

This connector is a CMOS expansion port used to control external devices such as RF front ends. Each port is directly connected to the Spartan-3AN so special care should be taken not to overload the port. Each pin is 3.3 V and can be used as an input or an output. As outputs each pin can yield as much as 12 mA of current.

Pin	Assignment	Pin	Assignment
1	GPIO_0	2	GPIO_1
3	GND	4	GPIO_2
5	GPIO_3	6	GPIO_4
7	GPIO_5	8	GPIO_6
9	GPIO_7	10	GPIO_8
11	GPIO_9	12	GPIO_10
13	GPIO_11	14	GPIO_12
15	GPIO_13	16	GPIO_14
17	GPIO_15	18	GPIO_16
19	GPIO_17	20	GPIO_18
21	GPIO_19	22	GPIO_20
23	GPIO_21	24	GPIO_22
25	GPIO_23	26	GPIO_24
27	GPIO_25	28	GPIO_26
29	GPIO_27	30	GPIO_28
31	GPIO_29	32	GND
33	GPIO_30	34	GPIO_31

Table 16 Pin assignments

4.1.3 Mestor Breakout Box Perseus FPGA Core

Nutaq supplies an FPGA core to control the breakout box. This core allows you to access the GPIO-32 port and the A/D data. As there are not enough pins to transfer all the information between the breakout box and the Perseus, the information is serialized between the Perseus' FPGA and the breakout box FPGA.

5 Specifications

This chapter presents the main technical specifications of the Perseus 601X.

Note:

The specifications presented here are subject to change without notice.

5.1 Mechanical Specifications

5.1.1 Overall Clearance

Mid-size height: 20 mm

Full-size height: 30 mm

With heatsink: 18 mm

5.1.2 Board Dimensions

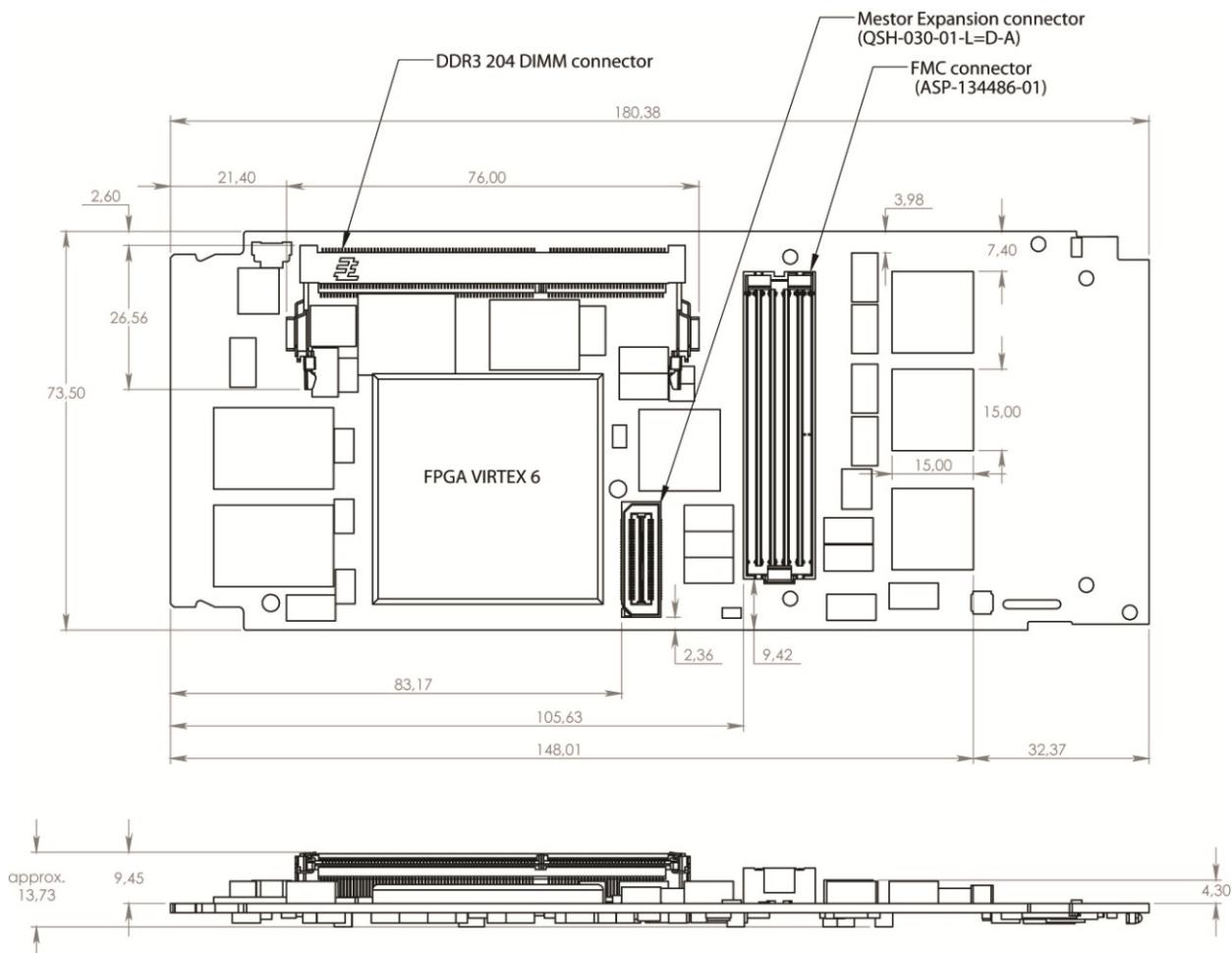


Figure 5-1 Top and side dimensions

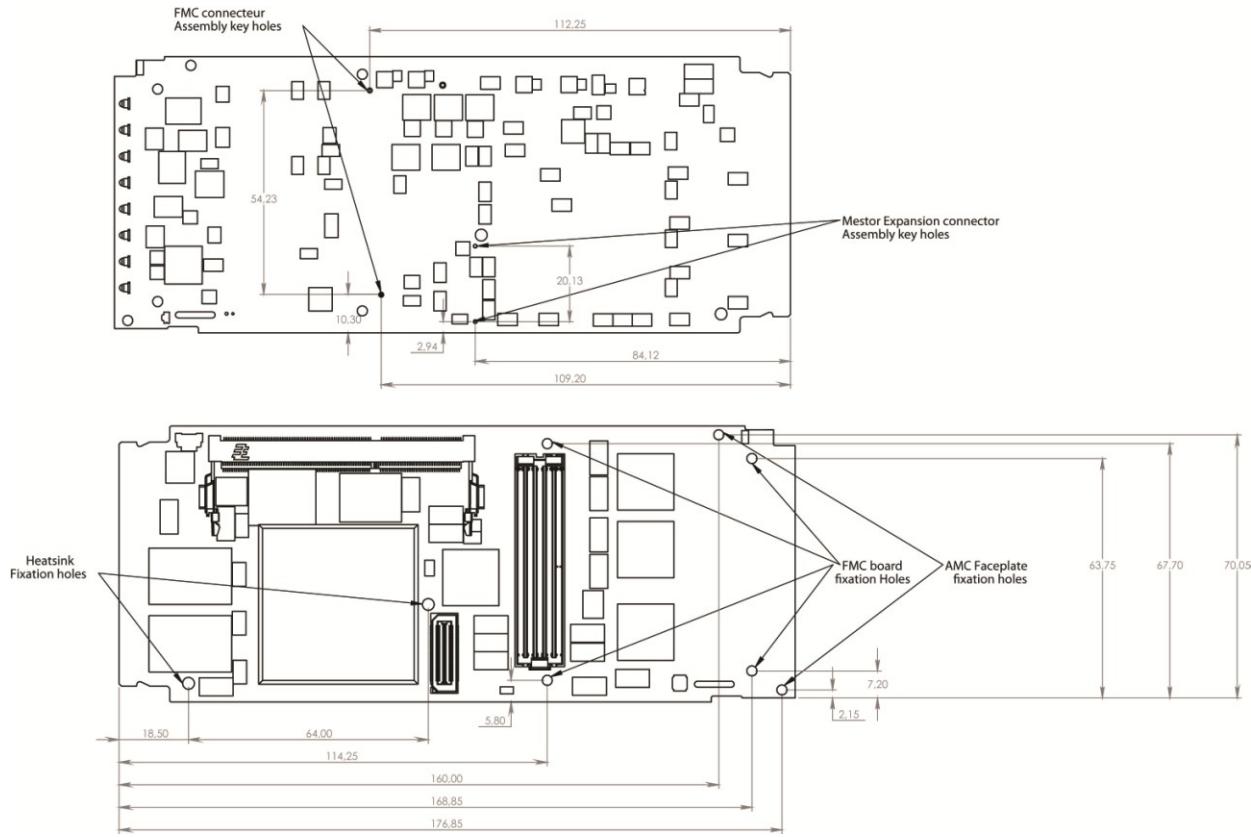


Figure 5-2 Bottom dimensions and assembly hole locations

5.2 FPGA Pin Assignments

In this section, we present the FPGA pins and their I/O assignments. For differential pairs, the LOC+ symbol represents the positive “p” pin and the LOC– symbol represents the negative “n” pin. “GTEx” indicates the GTX address in the core, but has no valid standard for the UCF.

Note:

AMC12 to AMC15 are LVDS ports. As such, they are not connected to a GTX transceiver.

Pin name	LOC+	I/O buffer	I/O standard	Resistor
EXT_RESET	AV16	IN	LVCMS25	PULLDOWN

Table 17 External reset

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_RX	0	AE5	AE6	INDS	GTXE1_X0Y6
AMC_RX	1	AD3	AD4	INDS	GTXE1_X0Y7
AMC_RX	2	AG5	AG6	INDS	GTXE1_X0Y4
AMC_RX	3	AF3	AF4	INDS	GTXE1_X0Y5

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_RX	4	AC5	AC6	INDS	GTXE1_X0Y8
AMC_RX	5	AB3	AB4	INDS	GTXE1_X0Y9
AMC_RX	6	AA5	AA6	INDS	GTXE1_X0Y10
AMC_RX	7	Y3	Y4	INDS	GTXE1_X0Y11
AMC_RX	8	W5	W6	INDS	GTXE1_X0Y12
AMC_RX	9	V3	V4	INDS	GTXE1_X0Y13
AMC_RX	10	U5	U6	INDS	GTXE1_X0Y14
AMC_RX	11	R5	R6	INDS	GTXE1_X0Y15
AMC_RX	12	R32	T32	INDS	LVDS_25
AMC_RX	13	Y30	AA30	INDS	LVDS_25
AMC_RX	14	AD31	AD30	INDS	LVDS_25
AMC_RX	15	AK33	AJ32	INDS	LVDS_25
AMC_RX	17	AJ5	AJ6	INDS	GTXE1_X0Y3
AMC_RX	18	AL5	AL6	INDS	GTXE1_X0Y2
AMC_RX	19	AM7	AM8	INDS	GTXE1_X0Y1
AMC_RX	20	AN5	AN6	INDS	GTXE1_X0Y0
AMC_TX	0	AH3	AH4	OUTDS	GTXE1
AMC_TX	1	AG1	AG2	OUTDS	GTXE1
AMC_TX	2	AK3	AK4	OUTDS	GTXE1
AMC_TX	3	AJ1	AJ2	OUTDS	GTXE1
AMC_TX	4	AE1	AE2	OUTDS	GTXE1
AMC_TX	5	AC1	AC2	OUTDS	GTXE1
AMC_TX	6	AA1	AA2	OUTDS	GTXE1

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_TX	7	W1	W2	OUTDS	GTXE1
AMC_TX	8	U1	U2	OUTDS	GTXE1
AMC_TX	9	T3	T4	OUTDS	GTXE1
AMC_TX	10	R1	R2	OUTDS	GTXE1
AMC_TX	11	P3	P4	OUTDS	GTXE1
AMC_TX	12	AA31	AB31	OUTDS	LVDS_25
AMC_TX	13	AC31	AC30	OUTDS	LVDS_25
AMC_TX	14	AH29	AG29	OUTDS	LVDS_25
AMC_TX	15	AJ31	AK30	OUTDS	LVDS_25
AMC_TX	17	AL1	AL2	OUTDS	GTXE1
AMC_TX	18	AM3	AM4	OUTDS	GTXE1
AMC_TX	19	AN1	AN2	OUTDS	GTXE1
AMC_TX	20	AP3	AP4	OUTDS	GTXE1

Table 18 AMC connector (LX240/SX315 Virtex-6 Perseus)

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_RX	0	AE5	AE6	INDS	GTXE1_X0Y14
AMC_RX	1	AD3	AD4	INDS	GTXE1_X0Y15
AMC_RX	2	AG5	AG6	INDS	GTXE1_X0Y12
AMC_RX	3	AF3	AF4	INDS	GTXE1_X0Y13
AMC_RX	4	AC5	AC6	INDS	GTXE1_X0Y16
AMC_RX	5	AB3	AB4	INDS	GTXE1_X0Y17
AMC_RX	6	AA5	AA6	INDS	GTXE1_X0Y18
AMC_RX	7	Y3	Y4	INDS	GTXE1_X0Y19

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_RX	8	W5	W6	INDS	GTXE1_X0Y20
AMC_RX	9	V3	V4	INDS	GTXE1_X0Y21
AMC_RX	10	U5	U6	INDS	GTXE1_X0Y22
AMC_RX	11	R5	R6	INDS	GTXE1_X0Y23
AMC_RX	12	R32	T32	INDS	LVDS_25
AMC_RX	13	Y30	AA30	INDS	LVDS_25
AMC_RX	14	AD31	AD30	INDS	LVDS_25
AMC_RX	15	AK33	AJ32	INDS	LVDS_25
AMC_RX	17	AJ5	AJ6	INDS	GTXE1_X0Y11
AMC_RX	18	AL5	AL6	INDS	GTXE1_X0Y10
AMC_RX	19	AM7	AM8	INDS	GTXE1_X0Y9
AMC_RX	20	AN5	AN6	INDS	GTXE1_X0Y8
AMC_TX	0	AH3	AH4	OUTDS	GTXE1
AMC_TX	1	AG1	AG2	OUTDS	GTXE1
AMC_TX	2	AK3	AK4	OUTDS	GTXE1
AMC_TX	3	AJ1	AJ2	OUTDS	GTXE1
AMC_TX	4	AE1	AE2	OUTDS	GTXE1
AMC_TX	5	AC1	AC2	OUTDS	GTXE1
AMC_TX	6	AA1	AA2	OUTDS	GTXE1
AMC_TX	7	W1	W2	OUTDS	GTXE1
AMC_TX	8	U1	U2	OUTDS	GTXE1
AMC_TX	9	T3	T4	OUTDS	GTXE1
AMC_TX	10	R1	R2	OUTDS	GTXE1
AMC_TX	11	P3	P4	OUTDS	GTXE1
AMC_TX	12	AA31	AB31	OUTDS	LVDS_25

Pin name	Index	LOC+	LOC-	I/O buffer	GTX address/ I/O standard
AMC_TX	13	AC31	AC30	OUTDS	LVDS_25
AMC_TX	14	AH29	AG29	OUTDS	LVDS_25
AMC_TX	15	AJ31	AK30	OUTDS	LVDS_25
AMC_TX	17	AL1	AL2	OUTDS	GTXE1
AMC_TX	18	AM3	AM4	OUTDS	GTXE1
AMC_TX	19	AN1	AN2	OUTDS	GTXE1
AMC_TX	20	AP3	AP4	OUTDS	GTXE1

Table 19 AMC connector (LX550/SX475 Virtex-6 Perseus)

Pin name	LOC+	I/O buffer	I/O standard
AVR_RXD	AW35	IN	LVCMOS25
AVR_TXD	AY34	OUT	LVCMOS25
E_KEY0_FPGA	D15	INOUT	LVCMOS25
E_KEY1_FPGA	C15	INOUT	LVCMOS25
E_KEY2_FPGA	G12	INOUT	LVCMOS25
E_KEY3_FPGA	H13	INOUT	LVCMOS25

Table 20 AVR (IPMI)

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
CCE_DDR3_A	0	E24	–	OUT	SSTL15
CCE_DDR3_A	1	D22	–	OUT	SSTL15
CCE_DDR3_A	2	C24	–	OUT	SSTL15
CCE_DDR3_A	3	H23	–	OUT	SSTL15
CCE_DDR3_A	4	B23	–	OUT	SSTL15
CCE_DDR3_A	5	K20	–	OUT	SSTL15
CCE_DDR3_A	6	C23	–	OUT	SSTL15
CCE_DDR3_A	7	J21	–	OUT	SSTL15

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
CCE_DDR3_A	8	B24	-	OUT	SSTL15
CCE_DDR3_A	9	K22	-	OUT	SSTL15
CCE_DDR3_A	10	F21	-	OUT	SSTL15
CCE_DDR3_A	11	A24	-	OUT	SSTL15
CCE_DDR3_A	12	B22	-	OUT	SSTL15
CCE_DDR3_A	13	L21	-	OUT	SSTL15
CCE_DDR3_A	14	D21	-	OUT	SSTL15
CCE_DDR3_BA	0	J20	-	OUT	SSTL15
CCE_DDR3_BA	1	A22	-	OUT	SSTL15
CCE_DDR3_BA	2	E23	-	OUT	SSTL15
CCE_DDR3_CAS_N	-	G22	-	OUT	SSTL15
CCE_DDR3_CK	-	H20	G21	OUTDS	DIFF_SSTL15
CCE_DDR3_CKE	-	D23	-	OUT	SSTL15
CCE_DDR3_CS_N	-	H21	-	OUT	SSTL15
CCE_DDR3_DQ	-	J16	-	OUT	SSTL15
CCE_DDR3_DQ	0	K18	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	1	G18	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	2	G19	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	3	E18	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	4	H18	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	5	E19	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	6	J18	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQ	7	F19	-	INOUT	SSTL15_T_DCI
CCE_DDR3_DQS	-	G16	F16	INOUTDS	SSTL15_T_DCI
CCE_DDR3_ODT	-	F22	-	OUT	SSTL15

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
CCE_DDR3_RAS_N	-	K19	-	OUT	SSTL15
CCE_DDR3_RESET_N	-	L20	-	OUT	SSTL15
CCE_DDR3_WE_N	-	G23	-	OUT	SSTL15

Table 21 DDR3 memory processor

Pin name	LOC	I/O buffer	I/O standard
CTRL_100MHZ_OUT_EN	H14	OUT	LVCMOS25
CTRL_AMCTCLKA_2_FMCLK2_EN	N14	OUT	LVCMOS25
CTRL_AMCTCLKC_2_FMCLK3_EN	AU16	OUT	LVCMOS25
CTRL_FCLKA_HIGHZ	A15	OUT	LVCMOS25
CTRL_FMCLK0_2_AMCTCLKB_EN	H15	OUT	LVCMOS25
CTRL_FMCLK1_2_AMCTCLKD_EN	G14	OUT	LVCMOS25
CTRL_LED1_GRN_N	AP40	OUT	LVCMOS25
CTRL_LED1_RED_N	AN40	OUT	LVCMOS25
CTRL_LED2_GRN_N	AP41	OUT	LVCMOS25
CTRL_LED2_RED_N	AN41	OUT	LVCMOS25
CTRL_LED3_GRN_N	AM39	OUT	LVCMOS25
CTRL_LED3_RED_N	AN39	OUT	LVCMOS25
CTRL_LED4_GRN_N	AR42	OUT	LVCMOS25
CTRL_LED4_RED_N	AP42	OUT	LVCMOS25
CTRL_LED5_GRN_N	AM38	OUT	LVCMOS25
CTRL_LED5_RED_N	AL37	OUT	LVCMOS25
CTRL_LED6_GRN_N	AU42	OUT	LVCMOS25
CTRL_LED6_RED_N	AT42	OUT	LVCMOS25
CTRL_LED7_GRN_N	AL35	OUT	LVCMOS25
CTRL_LED7_RED_N	AM34	OUT	LVCMOS25

Pin name	LOC	I/O buffer	I/O standard
CTRL_LED8_GRN_N	AW41	OUT	LVC MOS25
CTRL_LED8_RED_N	AW42	OUT	LVC MOS25
CTRL_LED_BUF_OD	AW40	OUT	LVC MOS25
CTRL_TCLKA_RX_DIS	J13	OUT	LVC MOS25
CTRL_TCLKA_TX_EN	K14	OUT	LVC MOS25
CTRL_TCLKB_RX_DIS	K13	OUT	LVC MOS25
CTRL_TCLKB_TX_EN	L14	OUT	LVC MOS25
CTRL_TCLKC_RX_DIS	B14	OUT	LVC MOS25
CTRL_TCLKC_TX_EN	C13	OUT	LVC MOS25
CTRL_TCLKD_RX_DIS	C14	INOUT	LVC MOS25
CTRL_TCLKD_TX_EN	D12	INOUT	LVC MOS25
CTRL_VADJ_EN	A14	OUT	LVC MOS25
CTRL_VADJ_SEL0	AV34	OUT	LVC MOS25
CTRL_VADJ_SEL1	AV35	OUT	LVC MOS25
CUSTOM0_FPGA_IO	F15	OUT	LVC MOS25
CUSTOM1_FPGA_IO	AW16	OUT	LVC MOS25
DAUGHTER_IO_ABSENT	AU34	IN	LVC MOS25
FPGA_PROG_N	M11	OUT	LVC MOS25

Table 22 Control pins

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_A	0	D40	–	OUT	SSTL15
DDR3_A	1	G41	–	OUT	SSTL15
DDR3_A	2	G42	–	OUT	SSTL15
DDR3_A	3	F37	–	OUT	SSTL15
DDR3_A	4	D41	–	OUT	SSTL15

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_A	5	F35	–	OUT	SSTL15
DDR3_A	6	F40	–	OUT	SSTL15
DDR3_A	7	E42	–	OUT	SSTL15
DDR3_A	8	G37	–	OUT	SSTL15
DDR3_A	9	G36	–	OUT	SSTL15
DDR3_A	10	B42	–	OUT	SSTL15
DDR3_A	11	F41	–	OUT	SSTL15
DDR3_A	12	H34	–	OUT	SSTL15
DDR3_A	13	B39	–	OUT	SSTL15
DDR3_A	14	F42	–	OUT	SSTL15
DDR3_A	15	F36	–	OUT	SSTL15
DDR3_BA	0	C41	–	OUT	SSTL15
DDR3_BA	1	A39	–	OUT	SSTL15
DDR3_BA	2	H36	–	OUT	SSTL15
DDR3_CAS_N	–	C40	–	OUTDS	SSTL15
DDR3_CK0	–	E39	E38	OUTDS	DIFF_SSTL15
DDR3_CK1	–	A40	A41	OUT	DIFF_SSTL15
DDR3_CKE0	–	H35	–	OUT	SSTL15
DDR3_CKE1	–	E40	–	OUT	SSTL15
DDR3_DM	0	J38	–	OUT	SSTL15
DDR3_DM	1	K33	–	OUT	SSTL15
DDR3_DM	2	F32	–	OUT	SSTL15
DDR3_DM	3	P28	–	OUT	SSTL15
DDR3_DM	4	E33	–	OUT	SSTL15
DDR3_DM	5	H30	–	OUT	SSTL15
DDR3_DM	6	B17	–	OUT	SSTL15

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_DM	7	P17	–	OUT	SSTL15
DDR3_DQ	0	M33	–	INOUT	SSTL15_T_DCI
DDR3_DQ	1	L37	–	INOUT	SSTL15_T_DCI
DDR3_DQ	2	H39	–	INOUT	SSTL15_T_DCI
DDR3_DQ	3	J42	–	INOUT	SSTL15_T_DCI
DDR3_DQ	4	M32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	5	M34	–	INOUT	SSTL15_T_DCI
DDR3_DQ	6	H38	–	INOUT	SSTL15_T_DCI
DDR3_DQ	7	K42	–	INOUT	SSTL15_T_DCI
DDR3_DQ	8	J40	–	INOUT	SSTL15_T_DCI
DDR3_DQ	9	H40	–	INOUT	SSTL15_T_DCI
DDR3_DQ	10	L34	–	INOUT	SSTL15_T_DCI
DDR3_DQ	11	L32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	12	H41	–	INOUT	SSTL15_T_DCI
DDR3_DQ	13	J37	–	INOUT	SSTL15_T_DCI
DDR3_DQ	14	K32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	15	L31	–	INOUT	SSTL15_T_DCI
DDR3_DQ	16	F31	–	INOUT	SSTL15_T_DCI
DDR3_DQ	17	D32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	18	B32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	19	A32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	20	E35	–	INOUT	SSTL15_T_DCI
DDR3_DQ	21	E32	–	INOUT	SSTL15_T_DCI
DDR3_DQ	22	C33	–	INOUT	SSTL15_T_DCI
DDR3_DQ	23	B33	–	INOUT	SSTL15_T_DCI

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_DQ	24	P31	–	INOUT	SSTL15_T_DC1
DDR3_DQ	25	P30	–	INOUT	SSTL15_T_DC1
DDR3_DQ	26	P27	–	INOUT	SSTL15_T_DC1
DDR3_DQ	27	R27	–	INOUT	SSTL15_T_DC1
DDR3_DQ	28	M31	–	INOUT	SSTL15_T_DC1
DDR3_DQ	29	N31	–	INOUT	SSTL15_T_DC1
DDR3_DQ	30	R29	–	INOUT	SSTL15_T_DC1
DDR3_DQ	31	N28	–	INOUT	SSTL15_T_DC1
DDR3_DQ	32	B36	–	INOUT	SSTL15_T_DC1
DDR3_DQ	33	A36	–	INOUT	SSTL15_T_DC1
DDR3_DQ	34	D36	–	INOUT	SSTL15_T_DC1
DDR3_DQ	35	E34	–	INOUT	SSTL15_T_DC1
DDR3_DQ	36	C35	–	INOUT	SSTL15_T_DC1
DDR3_DQ	37	C36	–	INOUT	SSTL15_T_DC1
DDR3_DQ	38	F34	–	INOUT	SSTL15_T_DC1
DDR3_DQ	39	D37	–	INOUT	SSTL15_T_DC1
DDR3_DQ	40	G32	–	INOUT	SSTL15_T_DC1
DDR3_DQ	41	G31	–	INOUT	SSTL15_T_DC1
DDR3_DQ	42	J30	–	INOUT	SSTL15_T_DC1
DDR3_DQ	43	L30	–	INOUT	SSTL15_T_DC1
DDR3_DQ	44	H31	–	INOUT	SSTL15_T_DC1
DDR3_DQ	45	J32	–	INOUT	SSTL15_T_DC1
DDR3_DQ	46	L29	–	INOUT	SSTL15_T_DC1
DDR3_DQ	47	M29	–	INOUT	SSTL15_T_DC1
DDR3_DQ	48	D18	–	INOUT	SSTL15_T_DC1
DDR3_DQ	49	A19	–	INOUT	SSTL15_T_DC1

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_DQ	50	B18	-	INOUT	SSTL15_T_DCI
DDR3_DQ	51	C18	-	INOUT	SSTL15_T_DCI
DDR3_DQ	52	A17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	53	D17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	54	G17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	55	J17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	56	M17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	57	N15	-	INOUT	SSTL15_T_DCI
DDR3_DQ	58	N18	-	INOUT	SSTL15_T_DCI
DDR3_DQ	59	P16	-	INOUT	SSTL15_T_DCI
DDR3_DQ	60	M16	-	INOUT	SSTL15_T_DCI
DDR3_DQ	61	L17	-	INOUT	SSTL15_T_DCI
DDR3_DQ	62	J15	-	INOUT	SSTL15_T_DCI
DDR3_DQ	63	K17	-	INOUT	SSTL15_T_DCI
DDR3_DQS0	-	L35	L36	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS1	-	K35	K34	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS2	-	A34	A35	INOUTV	DIFF_SSTL15_T_DCI
DDR3_DQS3	-	N29	N30	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS4	-	B34	C34	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS5	-	K29	K30	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS6	-	C19	B19	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_DQS7	-	L16	L15	INOUTDS	DIFF_SSTL15_T_DCI
DDR3_EVENT_N	-	C38	-	OUT	SSTL15
DDR3_ODT	0	A37	-	OUT	SSTL15
DDR3_ODT	1	B37	-	OUT	SSTL15

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DDR3_RAS_N	-	B38	-	OUT	SSTL15
DDR3_REFCLK	-	L12	M12	INDS	SSTL15
DDR3_RESET_N	-	J35	-	OUT	SSTL15
DDR3_SO_N	-	D38	-	OUT	SSTL15
DDR3_S1_N	-	C39	-	OUT	SSTL15
DDR3_SCL	-	C20	-	INOUT	SSTL15
DDR3_SDA	-	D20	-	INOUT	SSTL15
DDR3_VREF	-	J41	-	IN	SSTL15
DDR3_WE_N	-	B41	-	OUT	SSTL15

Table 23 DDR3 SODIMM

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DPIO	0	AY38	AY37	INOUTDS	LVDS_25
DPIO	10	BA37	BB37	INOUTDS	LVDS_25
DPIO	11	BB39	BB38	INOUTDS	LVDS_25
DPIO	12	BB36	BA36	INOUTDS	LVDS_25
DPIO	13	AW36	AV36	INOUTDS	LVDS_25
DPIO	1	AY39	BA39	INOUTDS	LVDS_25
DPIO	2	AW37	AW38	INOUTDS	LVDS_25
DPIO	3	AV39	AV38	INOUTDS	LVDS_25
DPIO	4	AU37	AU38	INOUTDS	LVDS_25
DPIO	5	AT37	AR38	INOUTDS	LVDS_25
DPIO	6	AU36	AT36	INOUTDS	LVDS_25
DPIO	7	AR35	AT35	INOUTDS	LVDS_25
DPIO	8	AP37	AR37	INOUTDS	LVDS_25
DPIO	9	AN35	AN36	INOUTDS	LVDS_25

Pin name	Index	LOC+	LOC-	I/O buffer	I/O standard
DPIO_CLK	-	AP36	AP35	INOUTDS	LVDS_25

Table 24 DPIO connector

Pin name	LOC+	LOC-	I/O buffer	I/O standard
FCLKA_IN	AB8	AB7	INOUTDS	BYPASS
TCLKA_IN	AN14	AN13	INDS	LVDS_25
TCLKA_OUT	F12	E12	INOUTDS	LVDS_25
TCLKB_IN	J12	J11	INDS	LVDS_25
TCLKB_OUT	A16	B16	INOUTDS	LVDS_25
TCLKC_IN	AM13	AM12	INDS	LVDS_25
TCLKC_OUT	K12	L11	INOUTDS	LVDS_25
TCLKD_IN	AV13	AV14	INDS	LVDS_25
TCLKD_OUT	D13	E13	INOUTDS	LVDS_25

Table 25 Backplane clocks

Pin name	Index	LOC+	I/O buffer	I/O standard	Resistor
FLASH_A	0	AW13	OUT	LVCMOS25_F_8	-
FLASH_A	1	AW12	OUT	LVCMOS25_F_8	-
FLASH_A	2	BB14	OUT	LVCMOS25_F_8	-
FLASH_A	3	BB13	OUT	LVCMOS25_F_8	-
FLASH_A	4	AU13	OUT	LVCMOS25_F_8	-
FLASH_A	5	AU12	OUT	LVCMOS25_F_8	-
FLASH_A	6	AW15	OUT	LVCMOS25_F_8	-
FLASH_A	7	AY15	OUT	LVCMOS25_F_8	-
FLASH_A	8	AR13	OUT	LVCMOS25_F_8	-
FLASH_A	9	AP13	OUT	LVCMOS25_F_8	-
FLASH_A	10	AU14	OUT	LVCMOS25_F_8	-

Pin name	Index	LOC+	I/O buffer	I/O standard	Resistor
FLASH_A	11	AV15	OUT	LVCMS25_F_8	-
FLASH_A	12	AT12	OUT	LVCMS25_F_8	-
FLASH_A	13	AR12	OUT	LVCMS25_F_8	-
FLASH_A	14	BA14	OUT	LVCMS25_F_8	-
FLASH_A	15	BA15	OUT	LVCMS25_F_8	-
FLASH_A	16	AW17	OUT	LVCMS25_F_8	-
FLASH_A	17	AY17	OUT	LVCMS25_F_8	-
FLASH_A	18	AR15	OUT	LVCMS25_F_8	-
FLASH_A	19	AP15	OUT	LVCMS25_F_8	-
FLASH_A	20	BB17	OUT	LVCMS25_F_8	-
FLASH_A	21	BB16	OUT	LVCMS25_F_8	-
FLASH_A	22	AT14	OUT	LVCMS25_F_8	-
FLASH_A	23	AR14	OUT	LVCMS25_F_8	-
FLASH_A	24	BA17	OUT	LVCMS25_F_8	-
FLASH_A	25	BA16	OUT	LVCMS25_F_8	-
FLASH_A	26	V	OUT	LVCMS25_F_8	-
FLASH_CE_N	-	AH30	OUT	LVCMS25	-
FLASH_DQ	0	U31	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	1	T31	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	2	AL32	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	3	AK32	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	4	R33	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	5	P32	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	6	AH33	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	7	AJ33	INOUT	LVCMS25_F_12	PULLDOWN
FLASH_DQ	8	P33	INOUT	LVCMS25_F_12	PULLDOWN

Pin name	Index	LOC+	I/O buffer	I/O standard	Resistor
FLASH_DQ	9	N33	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	10	AG31	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	11	AH31	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	12	R30	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	13	T30	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	14	AF31	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_DQ	15	AG32	INOUT	LVCMOS25_F_12	PULLDOWN
FLASH_OE_N	-	AJ30	OUT	LVCMOS25_F_12	-
FLASH_RST_N	-	W31	OUT	LVCMOS25_F_12	-
FLASH_WE-N	-	V31	OUT	LVCMOS25_F_12	-

Table 26 Flash memory**Note:**

The table below uses the following conventions in the Function column.

- MRCC XX: Identifies pins that can be used as a multiregion clock-capable pins in banks XX and as normal I/Os.
- SRCC XX: Identifies pins that can be used as a single region clock-capable pins in banks XX and as normal I/Os.
- VRN/VRP XX: Identifies pins that can be used as a voltage reference pins in banks XX with the high-speed I/O standard and as normal I/Os.

Pin name	LOC+	LOC-	I/O buffer	I/O standard
REFCLK200	E14	F14	INDS	LVDS_25

Table 27 200 MHz reference clock

Pin name	LOC+	LOC-	I/O buffer	I/O standard
MGTAVTTRCAL	A12	-	IN	BYPASS
MGTREFCLK1_IN_125M	AF8	AF7	INDS	BYPASS
MGTREFCLK1_IN_625M	AK8	AK7	INDS	BYPASS
MGTREFCLK2_IN_100M	Y8	Y7	INDS	BYPASS
MGTREFCLK2_IN_125M	T8	T7	INDS	BYPASS

Pin name	LOC+	LOC-	I/O buffer	I/O standard
MGTREFCLK2_IN_625M	V8	V7	IN/DS	BYPASS

Table 28 GTX reference clocks

Pin name	LOC+	I/O buffer	I/O standard
UBLAZE_RXD	BB34	IN	LVCMOS25
UBLAZE_TXD	BA34	OUT	LVCMOS25

Table 29 Front panel UART connector

Appendix A. FPGA Address Spaces

This chapter presents the physical address space of the Perseus 601X FPGA. This address space is divided into various sections — some representing peripherals that you can access, others representing simple registers that you can access. You can access the address space through the MicroBlaze code.

This chapter presents the QSFP S38FP+ module assembly.

Processor Boot Memory

This section of the FPGA address space ranges from 0x00000000 to 0x00007FFF and contains the processor boot code. At startup, the MicroBlaze runs the code in this section. You can also build your own application and store it within this section. By default, a boot loader occupies this section of memory and allows the MicroBlaze to jump to the program stored at the beginning of the flash memory, which is, by default, a second-stage boot loader (U-Boot).

Register User Space

In this section, ranging from 0x60000000 to 0x6FFFFFFF, you can map sets of AXI registers intended to access your own customized pcores.

Note:

In some cases, if you want to use specific drivers in this space, you must rebuild the OS kernel when running Linux on the MicroBlaze.

Perseus Registers

This section ranges from 0x70000000 to 0x7FFFFFFF and contains the registers specific to the Perseus FPGA. They are used to program the various interfaces of the FPGA. The registers presented in this section of this guide follow the register bit description table as it appears here.

Bit number
Bit description
Read/write functions
Initial values at reset or FPGA internal software reset

Configuration Registers

Configuration registers are implemented in the FPGA and wrapped in the LYR_axi_perseus6010_regs module (AXI subsystem). The configuration register software module also allows for more flexibility in design by offering 32 user-defined custom registers; the lower address of which contain special FPGA registers. You can access these registers through the address (0x70000000) in the MicroBlaze.

Offset 0x00 — DAUGTHER_ABSENT

Indicates the presence of an FMC or Mestor at the connector.

31 to 3	2	1	0
RESERVED	FMC STACK ABSENT	FMC ABSENT	DAUGHTER ABSENT
R	R	R	R
00000000000000000000000000000000	0	0	0

Table 30 DAUGTHERS_ABSENT status register

Bit	Description	Return
DAUGHTER ABSENT	Indicates whether a Mestor or Mestor expander is present at the connector.	0x0: Mestor present. 0x1: Mestor absent.
FMC ABSENT	Indicates whether an FMC is present at the connector.	0x0: FMC present. 0x1: FMC absent.
FMC STACK ABSENT	Indicates whether a second FMC is present at the connector, creating an FMC stack.	0x0: FMC stack present. 0x1: FMC stack absent.

Table 31 Offset 0x00 descriptions

Offset 0x04 — CLK_CTRL

Allows you to configure the clocks.

31 to 15	14	13	12
RESERVED	CTRL_GATED_CLK_EN	CTRL_100_MHZ_OUT_EN	CTRL_FCLKA_HIGHZ
R	R/W	R/W	R/W
00000000000000000000000000000000	1	0	0

11	10
CTRL_AMC_TCLKA_2_FMC_CLK2_EN	CTRL_AMC_TCLKC_2_FMC_CLK3_EN
R/W	R/W
0	0

9	8
CTRL_FMC_CLK0_2_AMC_TCLKB_EN	CTRL_FMC_CLK1_2_AMC_TCLKD_EN
R/W	R/W
0	0

7	6	5	4
CTRL_TCLKA_RX_DIS	CTRL_TCLKA_TX_EN	CTRL_TCLKB_RX_DIS	CTRL_TCLKB_TX_EN

R/W	R/W	R/W	R/W
1	1	1	0
3	2	1	0
CTRL_TCLKC_RX_DIS	CTRL_TCLKC_TX_EN	CTRL_TCLKD_RX_DIS	CTRL_TCLKD_TX_EN
R/W	R/W	R/W	R/W
1	1	1	0

Table 32 CLK_CTRL status register

Bit	Description	Configuration
XXX_TX_EN	Enables or disables the TX clock.	0x0: Disables the TX clock. 0x1: Enables the TX clock.
XXX_RX_DIS	Enables or disables the RX clock.	0x0: Enables the RX clock. 0x1: Disables the RX clock.
CTRL_FCLKA_HIGH_Z	Modifies the state of fclkA.	0x0: Sets fclkA to low impedance. 0x1: Sets fclkA to high impedance.
CTRL_100_MHZ_OUT_EN	Enables or disables the 100 MHz clock.	0x0: Disables the 100 MHz clock. 0x1: Enables the 100 MHz clock.
CTRL_FMC_CLK1_2_AMC_TCLKD_EN	Enables or disables the FMC clock 1 clock path to AMC Tclk D.	0x0: Enables the path. 0x1: Disables the path.
CTRL_FMC_CLK0_2_AMC_TCLKB_EN	Enables or disables the FMC clock 0 clock path to AMC Tclk B.	0x0: Enables the path. 0x1: Disables the path.
CTRL_AMC_TCLKA_2_FMC_CLK2_E	Enables or disables the AMC Tclk A clock path to FMC clock 2.	0x0: Enables the path. 0x1: Disables the path.
CTRL_AMC_TCLKC_2_FMC_CLK3_EN	Enables or disables the AMC Tclk C clock path to FMC clock 3.	0x0: Enables the path. 0x1: Disables the path.
CTRL_GATED_CLK_EN	Enables or disables the clock coming from the Radio410X module.	0x0: Disables the clock. 0x1: Enables the clock. (default)

Table 33 Offset 0x04 descriptions**Offset 0x08 — VADJ_AND_FPGA_PROG_CTRL**

Allows you to configure the value of the V_{adj} port. To prevent damaging your FMC, make sure that the current CTRL_VADJ_SEL configuration is correct for your card before connecting it to the Perseus and enabling V_{adj} .

31 to 4	3	2	1	0
RESERVED	FPGA_PROG_N	CTRL_VADJ_EN	CTRL_VADJ_SEL	
R	R/W	R/W	R/W	
00000000000000000000000000000000	1	0	00	

Table 34 VADJ_AND_FPGA_PROG_CTRL status register

Bit	Description	Configuration
FPGA_PROG_N	Reprograms the FPGA with the bistream stored in the flash memory	0x0: Reprograms the FPGA. 0x1: Does not reprogram the FPGA.
CTRL_VADJ_EN	Enables or disable the V_{adj} power supply.	0x0: Disables V_{adj} . 0x1: Enables V_{adj} .
CTRL_VADJ_SEL	Sets the V_{adj} power supply value.	0x0: Set the V_{adj} to 1.2 V. 0x1: Set the V_{adj} to 1.5 V. 0x2: Set the V_{adj} to 1.8 V. 0x3: Set the V_{adj} to 2.5 V.

Table 35 Offset 0x08 descriptions

Offset 0x0C — LED_CTRL

Allows you to configure the state of the front panel LEDs.

31 to 17																16							
RESERVED																CTRL_LED_BUF_OD							
R																R/W							
0000000000000000																0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
N_CTRL_LED_GRN																N_CTRL_LED_RED							
R/W																R/W							
0xFF																0xFF							

Table 36 LED_CTRL status register

Bit	Description	Configuration
CTRL_LED_BUF_OD	Enables or disables the front panel LED buffers. When disabled, the LEDs do no light.	0x0: Enables the LED buffers. 0x1: Disables the LED buffers.
N_CTRL_LED_GRN	Configures the state of each front panel green LED.	When the bits are 0 the LEDs light.
N_CTRL_LED_RED	Configures the state of each front panel red LED.	When the bits are 0 the LEDs light.

Table 37 Offset 0x0C descriptions

Offset 0x10 — CLK_STATUS

Allows you to acquire the clock status.

31 to 12	11	10	9
RESERVED	FMC_CLK3_STOP	FMC_CLK2_STOP	FMC_CLK1_STOP
R	R	R	R
0000000000000000	0	0	0

8	7	6	5	
FMC_CLK0_STOP	DPIO_CLK_STOP	QDR2_MMCM_LOCK	DDR3_SODIMM_MMCM_LOCK	
R	R	R	R	
0	0	0	0	
4	3	2	1	0
DDR3_CCE_MMCM_LOCK	TCLK_A_IN_STOP	TCLK_B_IN_STOP	TCLK_C_IN_STOP	TCLK_D_IN_STOP
R	R	R	R	R
0	0	0	0	0

Table 38 CLK_CTRL status register

Bit	Description	Return
XXX_MMCM_LOCK	Indicates that the MMCM is locked and running smoothly.	0x0: MMCM is unlocked and no clock is output. 0x1: MMCM is locked and running smoothly.
XXX_CLK_XX_STOP	Indicates the clock source status.	0x0: Clock in is OK. 0x1: Clock in is stopped.

Table 39 Offset 0x10 descriptions**Offset 0x14 — MEMORY_INIT_DONE_STATUS**

Allows you to acquire the memory initialization status.

31 to 4	3	2
RESERVED	QDR2_B1_INIT_DONE	QDR2_B0_INIT_DONE
R	R	R
00000000000000000000000000000000	0	0
1	0	
DDR3_SODIMM_INIT_DONE	DDR3_CCE_INIT_DONE	
R	R	
0	0	

Table 40 MEMORY_INIT_DONE_STATUS register

Bit	Description	Configuration
XXX_INIT_DONE	Indicates the memory controller initialization status.	0x0: The memory controller initialization is underway. 0x1: The memory controller initialization is done.

Table 41 Offset 0x14 descriptions

Offsets 0x18 to 0x94 — Custom registers 0 to 31

You can use these 32 registers as general-purpose user-defined registers (control or data).

31 to 0
User_reg0 to 31
R/W
X"00000000"

Table 42 User-defined custom registers

Offsets 0x98 to 0x9C — Reserved registers

These registers are internal and, as such, unconnected to external ports.

31 to 0
Reserved Reg0 to 1
R/W
X"00000000"

Table 43 Reserved registers

Offset 0xA0 — MMC_CTRL

This register allows accessing the MMC I²C bus.

31 to 2	1	0
RESERVED	MmcI2cReleaseAck	MmcI2cReleaseReq
R	R	R/W
00000000000000000000000000000000	0	0

Table 44 MMC control register

Bit	Description	Configuration
MmcI2cReleaseAck	When a request is sent to the MMC I ₂ C bus, the MMC responds by setting this bit.	When this bit is 1, the MMC acknowledges the request and grants access to the MMC I ₂ C bus. When this bit value is 0 after the request, the MMC denies access to the MMC I ₂ C bus. Once the request is sent, the bit reverts to 0, acknowledging the end of request.
MmcI2cReleaseReq	This bit enables a request to the MMC bus. It must be set to 1 to request I ₂ C bus ownership	When the bit is 1, a bus request is sent to the MMC. When the bit value changes from 1 to 0, an end of request is sent to the MMC.

Table 45 Offset 0xA0 descriptions

Offset 0xA4 — ACCESS_STATUS

This register indicates whether access at an invalid address of the AXI occurs.

31 to 1	0
---------	---

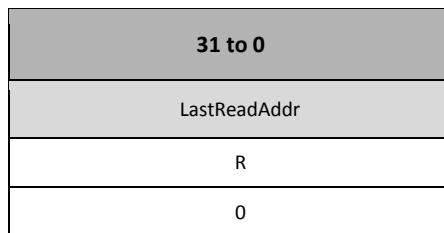
RESERVED	AccessFailed
R	R/W
00000000000000000000000000000000	0

Table 46 Access status

Bit	Description	Configuration
AccessFailed	Indicates whether an invalid access occurred at the AXI. Writing a value at the ACCESS_STATUS address clears the bit.	0x0: No invalid access. 0x1: Invalid access.

Table 47 Offset 0xA4 descriptions**Offset 0xA8 — LAST_RW_ADDRESS**

This register registers the address of invalid accesses on the AXI.

**Table 48 Last read address register**

Bit	Description	Configuration
LastReadAddr	When an invalid access occurs on the AXI, this register contains the invalid address where the access takes place.	Invalid address value.

Table 49 Offset 0xA8 descriptions

Nutaq IP Cores

This section presents Nutaq IP cores, which allow detection of these specific cores.

Name	Description
lyt_axi_adac250_v1_00_a	Provide status and control for the ADAC250 FMC card
lyt_axi_emac_rtdecr_v1_00_a	Used for data Transfer between the host PC and the Perseus board
lyt_axi_mi250_v1_00_a	Provide status and control for the MI250 FMC card
lyt_axi_qsfp_plus_v1_00_a	Provide status and control for the QSFP FMC card
lyt_axi_sfp_plus_v1_00_a	Provide status and control for the SFP FMC card
lyt_axi_radio420x_v1_00_a	Provide status and control for the radio 420 FMC card
lyt_axi_record_playback_v1_00_a	Provides the interface with the SODIMM memory
lyt_axi_mi125_v1_00_a	Provide status and control for the MI125 FMC card

lyt_axi_mo1000_v1_00_a	Provide status and control for the MO1000 FMC card
lyt_axi_lvds_io_v1_00_a	Provide status and control for the LVDS-xIn-xOut FMC card

Table 50 Main IP cores provide by Nutaq

System Peripherals

This section ranges from 0x80000000 to 0xFFFFFFF and is reserved to access Perseus peripherals. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x80000000–0x800FFFF — processor debugging module

This range of addresses is reserved for the processor debugging module, which allows you to debug your application code with a JTAG module. The module can also send command to the processor such as start, stop, reset, read/write registers. It is also capable of loading a binary image of Linux. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x81600000–0x8160FFFF — FMC I2C

This address range is reserved for the FPGA core communicating with FMCs through I2C. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x81800000–0x8180FFFF — interrupt controller

This address range is reserved for the FPGA interrupt controller, which handles interrupt sources from the FMC I2C interrupt, the DDR3 memory processor SDMA interrupt, the Ethernet AMC 0 TMAC interrupt, the watchdog timer interrupt, the front panel UART interrupt and the IPMI AVR UART interrupt. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x81C00000–0x81C7FFFF — Ethernet AMC 0

This address range is reserved for the FPGA Ethernet port 0, which handles communications between a host application and the MicroBlaze through UDP and TCP. For details, refer to the Xilinx documentation accompanying Platform Studio.

Note:

Not supported at the time of writing.

0x83600000–0x836000FF — watchdog timer

This address range is reserved for the watchdog timer, which allows the system to be reset after system failures. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x83C00000–0x83c000FF — real-time timer

This address range is reserved for the real-time timer, which is necessary to allow process switching by the operating system scheduler. For details, refer to the Xilinx documentation accompanying Platform Studio.

0x84000000–0x8400FFFF — front panel UART

This address range is reserved for the front panel UART controller.

0x84100000–0x8410FFFF — IPMI AVR UART

This address range is reserved for the IPMI AVR UART controller.

0x85000000–0x85FFFFFF — FMC

This address range is reserved to accessing the FMC connected to the Perseus.

0x8500X000 — FMC bottom/top

This address is reserved to indicate which FMC card is targeted 1 = top and 0 = bottom.

DDR3 Memory Processor

This section ranges from 0x90000000 to 0x97FFFFFF and is reserved for the MicroBlaze application code. By default, Linux runs in this section, but process code and data are stored in the section at runtime.

Flash Memory

This section ranges from 0x00000000 to 0x03FFFFFF and is reserved for the flash memory. The base address of the flash memory controller is 0x00000000, thus to find the real position of each section in the flash memory, you must subtract this value from the given address. For example, the U-Boot image actual flash memory address range is 0x00000000 to 0x03FFFFFF.

The following is a list of all the Perseus address ranges.

0x0000 0000 – 0x0003 FFFF — U-Boot image

This section contains the binary code of the U-Boot application (second-state boot loader) that loads Linux to the DDR3 SDRAM.

0x0004 0000 – 0x000 5FFFF — U-Boot environment

This section contains the environment variables (data section) of the U-Boot application.

0x0006 0000 – 0x0007 FFFF — dtb**0x0008 0000 – 0x00AF FFFF — Linux image**

This section contains the Linux image (code) that is loaded to the DDR3 SDRAM.

0x00B0 0000 – 0x00B1 FFFF — configuration**0x00B2 0000 – 0x017F FFFF — jffs2****0x0180 0000 – 0x02BF FFFF — fpga2: second bitstream in BPI down configuration****0x02C0 0000 – 0x03FF FFFF — fpga1: first bitstream in BPI down configuration**

Each **fpga** section contains an FPGA bitstream stored in BPI down mode. The bitstream can be loaded to the FPGA at startup. The LX240 FPGA needs 9 MB of flash memory to hold a bitstream and as the Perseus also supports the SX475 FPGA, enough space was reserved to contain 19 MB of data (each partition has 20 MB).

Note:

The three last address ranges can be resized to suit your needs. If you do not need to boot from the flash memory, your Linux file system can easily top off the flash memory. If only LX240 FPGA is used, the last 40 MB could be divided in more than 2 partitions. Modifying partition sizes require recompiling a PetaLinux device tree and the Central Communication Engine (CCE) provided by Nutaq will no longer work properly.

Appendix B. FMC Connector

This appendix presents information specific to the Perseus FMC connector.

Pin Assignments

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_ABSENT	–	AT34	–	IN	LVCMOS25
FMC_CLK0	–	AE30	AF30	INDS	APP/SPEC
FMC_CLK1	–	W30	V30	INDS	APP/SPEC
FMC_CLK2	–	AP11	AP12	INOUTDS	APP/SPEC
FMC_CLK3	–	AY14	AY13	INOUTDS	APP/SPEC
FMC_DP0_C2M	–	N1	AY13	OUTDS	GTXE1_X0Y16
FMC_DP0_M2C	–	P7	P8	INDS	GTXE1_X0Y16
FMC_DP1_C2M	–	M3	M4	OUTDS	GTXE1_X0Y17
FMC_DP1_M2C	–	N5	N6	INDS	GTXE1_X0Y17
FMC_DP2_C2M	–	L1	L2	OUTDS	GTXE1_X0Y18
FMC_DP2_M2C	–	L5	L6	INDS	GTXE1_X0Y18
FMC_DP3_C2M	–	K3	K4	OUTDS	GTXE1_X0Y19
FMC_DP3_M2C	–	J5	J6	INDS	GTXE1_X0Y19
FMC_DP4_C2M	–	J1	J2	OUTDS	GTXE1_X0Y20
FMC_DP4_M2C	–	H7	H8	INDS	GTXE1_X0Y20
FMC_DP5_C2M	–	H3	H4	OUTDS	GTXE1_X0Y21
FMC_DP5_M2C	–	G5	G6	INDSINDS	GTXE1_X0Y21
FMC_DP6_C2M	–	G1	G2	OUTDS	GTXE1_X0Y22
FMC_DP6_M2C	–	F7	F8	INDS	GTXE1_X0Y22

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_DP7_C2M	-	F3	F4	OUTDS	GTXE1_X0Y23
FMC_DP7_M2C	-	E5	E6	INDS	GTXE1_X0Y23
FMC_GBTCLK0_IN_M2C	-	M8	M7	INOUTDS	LVDS_25
FMC_GBTCLK1_IN_M2C	-	G10	G9	INOUTDS	LVDS_25
FMC_HA	0	AH34	AJ35	INOUTDS	MRCC 14
FMC_HA	1	AJ37	AK37	INOUTDS	MRCC 14
FMC_HA	2	AC34	AC33	INOUTDS	APP/SPEC
FMC_HA	3	AE40	AE39	INOUTDS	APP/SPEC
FMC_HA	4	AG34	AF34	INOUTDS	APP/SPEC
FMC_HA	5	AF40	AG41	INOUTDS	APP/SPEC
FMC_HA	6	AF39	AG39	INOUTDS	APP/SPEC
FMC_HA	7	AG42	AH41	INOUTDS	APP/SPEC
FMC_HA	8	AH40	AJ41	INOUTDS	APP/SPEC
FMC_HA	9	AF37	AG37	INOUTDS	APP/SPEC
FMC_HA	10	AJ42	AK42	INOUTDS	APP/SPEC
FMC_HA	11	AK38	AJ38	INOUTDS	SRCC 14
FMC_HA	12	AJ36	AH35	INOUTDS	VRN/VRP 14
FMC_HA	13	AL42	AM42	INOUTDS	APP/SPEC
FMC_HA	14	AL41	AM41	INOUTDS	APP/SPEC
FMC_HA	15	AF35	AF36	INOUTDS	APP/SPEC
FMC_HA	16	AK40	AL40	INOUTDS	APP/SPEC
FMC_HA	17	AH39	AJ40	INOUTDS	SRCC 14
FMC_HA	18	AF32	AG33	INOUTDS	APP/SPEC
FMC_HA	19	AK39	AL39	INOUTDS	APP/SPEC
FMC_HA	20	AL34	AK34	INOUTDS	MRCC 13

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_HA	21	AK35	AL36	INOUTDS	APP/SPEC
FMC_HA	22	AM37	AM36	INOUTDS	APP/SPEC
FMC_HA	23	AN38	AP38	INOUTDS	VRN/VRP 13
FMC_HB	0	R39	P38	INOUTDS	SRCC 17
FMC_HB	1	M36	M37	INOUTDS	APP/SPEC
FMC_HB	2	L39	L40	INOUTDS	APP/SPEC
FMC_HB	3	M38	M39	INOUTDS	APP/SPEC
FMC_HB	4	L41	L42	INOUTDS	APP/SPEC
FMC_HB	5	N38	N39	INOUTDS	APP/SPEC
FMC_HB	6	P36	P35	INOUTDS	MRCC 17
FMC_HB	7	N36	P37	INOUTDS	APP/SPEC
FMC_HB	8	M41	M42	INOUTDS	APP/SPEC
FMC_HB	9	N40	N41	INOUTDS	SRCC 17
FMC_HB	10	R37	T37	INOUTDS	VRN/VRP 17
FMC_HB	11	P40	P41	INOUTDS	APP/SPEC
FMC_HB	12	P42	R42	INOUTDS	APP/SPEC
FMC_HB	13	U36	T36	INOUTDS	APP/SPEC
FMC_HB	14	R40	T40	INOUTDS	APP/SPEC
FMC_HB	15	T34	T35	INOUTDS	APP/SPEC
FMC_HB	16	T41	T42	INOUTDS	APP/SPEC
FMC_HB	17	T39	R38	INOUTDS	MRCC 17
FMC_LA	0	AD32	AE32	INOUTDS	MRCC 15
FMC_LA	1	AE37	AD37	INOUTDS	MRCC 15
FMC_LA	2	AE34	AE35	INOUTDS	APP/SPEC
FMC_LA	3	AF42	AF41	INOUTDS	APP/SPEC
FMC_LA	4	AE33	AD33	INOUTDS	APP/SPEC

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_LA	5	AB39	AA40	INOUTDS	APP/SPEC
FMC_LA	6	AA41	AB41	INOUTDS	APP/SPEC
FMC_LA	7	AC41	AD41	INOUTDS	SRCC 15
FMC_LA	8	AD42	AE42	INOUTDS	APP/SPEC
FMC_LA	9	AC36	AB36	INOUTDS	SRCC 15
FMC_LA	10	AA42	AB42	INOUTDS	APP/SPEC
FMC_LA	11	AE38	AD38	INOUTDS	APP/SPEC
FMC_LA	12	AC40	AD40	INOUTDS	APP/SPEC
FMC_LA	13	AB32	AB33	INOUTDS	APP/SPEC
FMC_LA	14	AC35	AB34	INOUTDS	VRN/VRP 15
FMC_LA	15	AB37	AB38	INOUTDS	APP/SPEC
FMC_LA	16	W37	Y37	INOUTDS	APP/SPEC
FMC_LA	17	W32	Y33	INOUTDS	MRCC 16
FMC_LA	18	V34	U34	INOUTDS	MRCC 16
FMC_LA	19	Y38	AA39	INOUTDS	APP/SPEC
FMC_LA	20	W42	Y42	INOUTDS	APP/SPEC
FMC_LA	21	W35	V35	INOUTDS	APP/SPEC
FMC_LA	22	Y40	Y39	INOUTDS	APP/SPEC
FMC_LA	23	V40	W40	INOUTDS	SRCC 16
FMC_LA	24	V38	W38	INOUTDS	APP/SPEC
FMC_LA	25	AA35	Y35	INOUTDS	SRCC 16
FMC_LA	26	V41	W41	INOUTDS	APP/SPEC
FMC_LA	27	U39	V39	INOUTDS	APP/SPEC
FMC_LA	28	U42	U41	INOUTDS	APP/SPEC
FMC_LA	29	W36	V36	INOUTDS	APP/SPEC

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_LA	30	V33	W33	INOUTDS	APP/SPEC
FMC_LA	31	U37	U38	INOUTDS	APP/SPEC
FMC_LA	32	U32	U33	INOUTDS	APP/SPEC
FMC_LA	33	AA32	Y32	INOUTDS	VRN/VRP 16
FMC_SCL	–	BA35	–	INOUT	LVCMOS25
FMC_SDA	–	AY35	–	INOUT	LVCMOS25
FMC_VREF_A_M2C	–	AH36	–	ANALOG	N/A
FMC_VREF_B_M2C	–	R34	–	ANALOG	N/A

Table 51 FMC connector (LX240/SX315 Virtex-6 Perseus)

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_ABSENT	–	AT34	–	IN	LVCMOS25
FMC_CLK0	–	AE30	AF30	INDS	APP/SPEC
FMC_CLK1	–	W30	V30	INDS	APP/SPEC
FMC_CLK2	–	AP11	AP12	INOUTDS	APP/SPEC
FMC_CLK3	–	AY14	AY13	INOUTDS	APP/SPEC
FMC_DP0_C2M	–	N1	N2	OUTDS	GTXE1_X0Y24
FMC_DP0_M2C	–	P7	P8	INDS	GTXE1_X0Y24
FMC_DP1_C2M	–	M3	M4	OUTDS	GTXE1_X0Y25
FMC_DP1_M2C	–	N5	N6	INDS	GTXE1_X0Y25
FMC_DP2_C2M	–	L1	L2	OUTDS	GTXE1_X0Y26
FMC_DP2_M2C	–	L5	L6	INDS	GTXE1_X0Y26
FMC_DP3_C2M	–	K3	K4	OUTDS	GTXE1_X0Y27
FMC_DP3_M2C	–	J5	J6	INDS	GTXE1_X0Y27
FMC_DP4_C2M	–	J1	J2	OUTDS	GTXE1_X0Y28
FMC_DP4_M2C	–	H7	H8	INDS	GTXE1_X0Y28

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_DP4_M2C	-	H7	H8	INDS	GTXE1_X0Y28
FMC_DP5_C2M	-	H3	H4	OUTDS	GTXE1_X0Y29
FMC_DP5_M2C	-	G5	G6	INDS	GTXE1_X0Y29
FMC_DP6_C2M	-	G1	G2	OUTDS	GTXE1_X0Y30
FMC_DP6_M2C	-	F7	F8	INDS	GTXE1_X0Y30
FMC_DP7_C2M	-	F3	F4	OUTDS	GTXE1_X0Y31
FMC_DP7_M2C	-	E5	E6	INDS	GTXE1_X0Y31
FMC_GBTCLK0_IN_M2C	-	M8	M7	INOUTDS	LVDS_25
FMC_GBTCLK1_IN_M2C	-	G10	G9	INOUTDS	LVDS_25
FMC_HA	0	AH34	AJ35	INOUTDS	MRCC 14
FMC_HA	1	AJ37	AK37	INOUTDS	MRCC 14
FMC_HA	2	AC34	AC33	INOUTDS	APP/SPEC
FMC_HA	3	AE40	AE39	INOUTDS	APP/SPEC
FMC_HA	4	AG34	AF34	INOUTDS	APP/SPEC
FMC_HA	5	AF40	AG41	INOUTDS	APP/SPEC
FMC_HA	6	AF39	AG39	INOUTDS	APP/SPEC
FMC_HA	7	AG42	AH41	INOUTDS	APP/SPEC
FMC_HA	8	AH40	AJ41	INOUTDS	APP/SPEC
FMC_HA	9	AF37	AG37	INOUTDS	APP/SPEC
FMC_HA	10	AJ42	AK42	INOUTDS	APP/SPEC
FMC_HA	11	AK38	AJ38	INOUTDS	SRCC 14
FMC_HA	12	AJ36	AH35	INOUTDS	VRN/VRP 14
FMC_HA	13	AL42	AM42	INOUTDS	APP/SPEC
FMC_HA	14	AL41	AM41	INOUTDS	APP/SPEC
FMC_HA	15	AF35	AF36	INOUTDS	APP/SPEC

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_HA	16	AK40	AL40	INOUTDS	APP/SPEC
FMC_HA	17	AH39	AJ40	INOUTDS	SRCC 14
FMC_HA	18	AF32	AG33	INOUTDS	APP/SPEC
FMC_HA	19	AK39	AL39	INOUTDS	APP/SPEC
FMC_HA	20	AL34	AK34	INOUTDS	MRCC 13
FMC_HA	21	AK35	AL36	INOUTDS	APP/SPEC
FMC_HA	22	AM37	AM36	INOUTDS	APP/SPEC
FMC_HA	23	AN38	AP38	INOUTDS	VRN/VRP 13
FMC_HB	0	R39	P38	INOUTDS	SRCC 17
FMC_HB	1	M36	M37	INOUTDS	APP/SPEC
FMC_HB	2	L39	L40	INOUTDS	APP/SPEC
FMC_HB	3	M38	M39	INOUTDS	APP/SPEC
FMC_HB	4	L41	L42	INOUTDS	APP/SPEC
FMC_HB	5	N38	N39	INOUTDS	APP/SPEC
FMC_HB	6	P36	P35	INOUTDS	MRCC 17
FMC_HB	7	N36	P37	INOUTDS	APP/SPEC
FMC_HB	8	M41	M42	INOUTDS	APP/SPEC
FMC_HB	9	N40	N41	INOUTDS	SRCC 17
FMC_HB	10	R37	T37	INOUTDS	VRN/VRP 17
FMC_HB	11	P40	P41	INOUTDS	APP/SPEC
FMC_HB	12	P42	R42	INOUTDS	APP/SPEC
FMC_HB	13	U36	T36	INOUTDS	APP/SPEC
FMC_HB	14	R40	T40	INOUTDS	APP/SPEC
FMC_HB	15	T34	T35	INOUTDS	APP/SPEC
FMC_HB	16	T41	T42	INOUTDS	APP/SPEC
FMC_HB	17	T39	R38	INOUTDS	MRCC 17

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_LA	0	AD32	AE32	INOUTDS	MRCC 15
FMC_LA	1	AE37	AD37	INOUTDS	MRCC 15
FMC_LA	2	AE34	AE35	INOUTDS	APP/SPEC
FMC_LA	3	AF42	AF41	INOUTDS	APP/SPEC
FMC_LA	4	AE33	AD33	INOUTDS	APP/SPEC
FMC_LA	5	AB39	AA40	INOUTDS	APP/SPEC
FMC_LA	6	AA41	AB41	INOUTDS	APP/SPEC
FMC_LA	7	AC41	AD41	INOUTDS	SRCC 15
FMC_LA	8	AD42	AE42	INOUTDS	APP/SPEC
FMC_LA	9	AC36	AB36	INOUTDS	SRCC 15
FMC_LA	10	AA42	AB42	INOUTDS	APP/SPEC
FMC_LA	11	AE38	AD38	INOUTDS	APP/SPEC
FMC_LA	12	AC40	AD40	INOUTDS	APP/SPEC
FMC_LA	13	AB32	AB33	INOUTDS	APP/SPEC
FMC_LA	14	AC35	AB34	INOUTDS	VRN/VRP 15
FMC_LA	15	AB37	AB38	INOUTDS	APP/SPEC
FMC_LA	16	W37	Y37	INOUTDS	APP/SPEC
FMC_LA	17	W32	Y33	INOUTDS	MRCC 16
FMC_LA	18	V34	U34	INOUTDS	MRCC 16
FMC_LA	19	Y38	AA39	INOUTDS	APP/SPEC
FMC_LA	20	W42	Y42	INOUTDS	APP/SPEC
FMC_LA	21	W35	V35	INOUTDS	APP/SPEC
FMC_LA	22	Y40	Y39	INOUTDS	APP/SPEC
FMC_LA	23	V40	W40	INOUTDS	SRCC 16
FMC_LA	24	V38	W38	INOUTDS	APP/SPEC

Pin name	Index	LOC+	LOC-	I/O buffer	Function
FMC_LA	25	AA35	Y35	INOUTDS	SRCC 16
FMC_LA	26	V41	W41	INOUTDS	APP/SPEC
FMC_LA	27	U39	V39	INOUTDS	APP/SPEC
FMC_LA	28	U42	U41	INOUTDS	APP/SPEC
FMC_LA	29	W36	V36	INOUTDS	APP/SPEC
FMC_LA	30	V33	W33	INOUTDS	APP/SPEC
FMC_LA	31	U37	U38	INOUTDS	APP/SPEC
FMC_LA	32	U32	U33	INOUTDS	APP/SPEC
FMC_LA	33	AA32	Y32	INOUTDS	VRN/VRP 16
FMC_SCL	–	BA35	–	INOUT	LVCMOS25
FMC_SDA	–	AY35	–	INOUT	LVCMOS25
FMC_VREF_A_M2C	–	AH36	–	ANALOG	N/A
FMC_VREF_B_M2C	–	R34	–	ANALOG	N/A

Table 52 FMC connector (LX550/SX475 Virtex-6 Perseus)

Compatibility Considerations

I²C bus management

This section discusses the Perseus I²C bus and some of the design considerations that must be taken into account when using it. The following diagram illustrates the I²C bus connections.

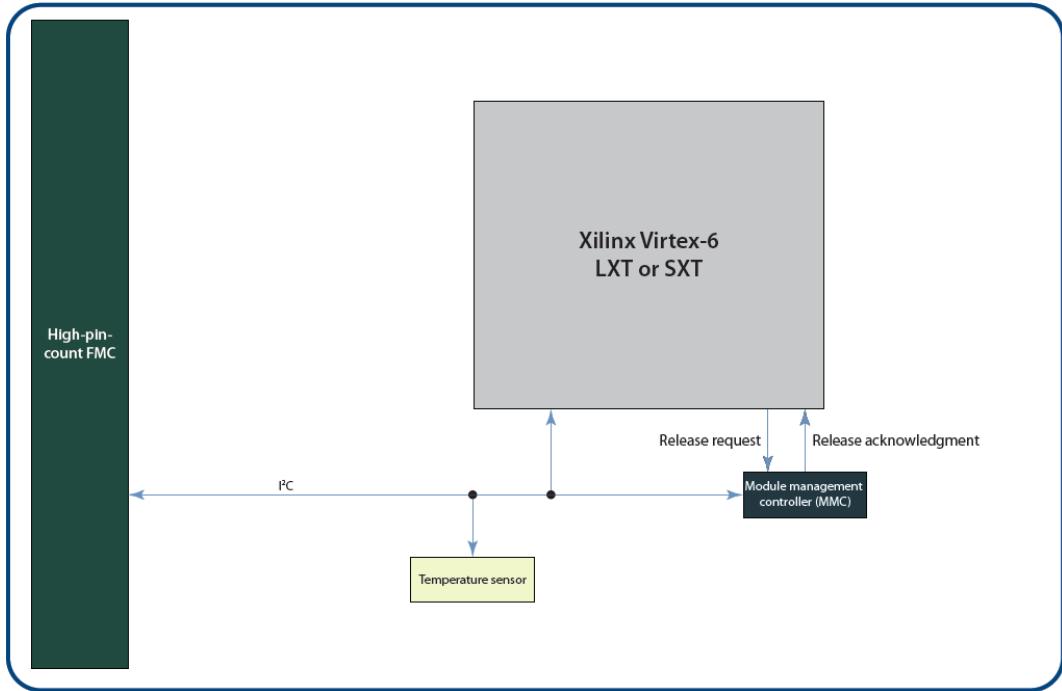


Figure 5-3 I²C bus connections

As you can see from the diagram, there are three devices connected to the Perseus I²C bus:

- Virtex-6 FPGA
- Module management controller (MMC)
- Maxim DS75S+ temperature sensor

MMC bus release

Under normal operating conditions, the MMC is the bus master and regularly polls the temperature sensor to keep its readings up-to-date. When necessary, the user design in the FPGA can take control of the bus by asserting a release I²C bus request signal. The MMC releases the bus and asserts an acknowledgement signal to let the user design know that it can now be the bus master. When the user design no longer needs the I²C bus, deassert the release I²C bus request signal to return control of the bus to the MMC. The release request and acknowledgement signals can be accessed by the user design through Perseus custom registers.

I²C address limitations of FMCs

The Perseus temperature sensor address is 1001000. This address is fixed by hardware. To avoid message collisions when using FMCs on the Perseus, it is important that no I²C device on the FMC use the temperature sensor address.