Titanium Ti375 N1156 Development Kit User Guide

Ti375N1156-DK-UG-v1.0
October 2024
www.efinixinc.com

Contents

Introduction

Thank you for choosing the Titanium Ti375 N1156 Development Kit (part number: Ti375N1156-DK), which allows you to explore the features of the Ti375 FPGA.

The Titanium Ti375 N1156 Development Kit has everything you need to develop and prototype designs for the Ti375 FPGA. The Ti375 FPGA's full-duplex transceivers support multiple protocols including PCIe® Gen4 with data rates up to 16 Gbps, Ethernet 10GBase-KR, and Ethernet SGMII. The PCIe and Ethernet transceivers have a hardened PCS, which makes them easy to implement and use. The Ti375 also supports a PMA Direct mode with data rates up to 12.5 Gbps, which you can use for custom protocols. The Titanium Ti375 N1156 Development Board has a PCIe edge card connector so you can attach it to a motherboard. It also has SFP+ receptacles that allows data transmission of up to 10.3 Gbps through the Ethernet interfaces.

The Titanium Ti375 N1156 Development Board has two (2) 512 Mbit SPI NOR flash devices, which support SPI active (x1, x2, x4, x8 modes) to configure the FPGA and store other user data such as RISC-V software. One SPI flash device (U48) also supports SPI passive mode. A highly integrated Ethernet PHY is on-board supporting RGMII at up to 1,000 Mbps.

One key feature is the on-board USB-to-dual JTAG device that supports simultaneous debugging of the Ti375 FPGA and the RISC-V SoC. Additionally, the board has an array of on-board interfaces such as 2 FMC (HPC) and 4 QSE interfaces, which let you expand the board's connectivity.

Warning: The board can be damaged without proper anti-static handling.

What's in the Box?

The Titanium Ti375 N1156 Development Kit includes:

- Titanium Ti375 N1156 Development Board preloaded with a demonstration design
- 1 USB type-C cable
- 12 V, 6.25 A universal power adapter with 5.5 mm DC power converter
- Jumpers
- 4 screws and 4 standoffs
- Cooling fan and thermal pad

Register Your Kit

When you purchase an Efinix development kit, you also receive a license for the Efinity® software, plus 1 year of software upgrades and patches. After the first year you can request a free maintenance renewal. The Efinity® software is available for download from the Support Center.

To download the software, first register at our Support Center (**https://www.efinixinc.com/register**), then register your development kit.

Download the Efinity® Software

To develop your own designs for the Ti375 FPGA on the board, you must install the Efinity® software. You can obtain the software from the Efinix Support Center under Efinity Software (www.efinixinc.com/support/).

The Efinity® software includes tools to program the device on the board. Refer to the Efinity® Software User Guide for information about how to program the device.

Learn more: Efinity® documentation is installed with the software (see **Help** > **Documentation**) and is also available in the Support Center under Documentation (www.efinixinc.com/support/).

Installing the Linux USB Driver

The following instructions explain how to install a USB driver for Linux operating systems.

1. Disconnect your board from your computer.
2. In a terminal, use these commands:

```
> sudo <installation directory>/bin/install_usb_driver.sh
> sudo udevadm control --reload-rules
```

Note: If your board was connected to your computer before you executed these commands, you need to disconnect it, then re-connect it.

Installing the Windows USB Drivers

The Titanium Ti375 N1156 Development Board development board has an FTDI FT4232H chip to communicate with the USB port.

Note: If you have another Efinix board and are using the Titanium Ti375 N1156 Development Board, you must manage drivers accordingly. Refer to AN 050: Managing Windows Drivers for more information.

On Windows, you use software from Zadig to install drivers. Download the Zadig software (version 2.7 or later) from zadig.akeo.ie. (You do not need to install it; simply run the downloaded executable.)

To install the driver:

1. Connect the board to your computer with the appropriate cable and power it up.
2. Run the Zadig software.
3. Choose **Options > List All Devices**.
4. Select **Ti375 N1156 Development Kit (Interface 1)**.

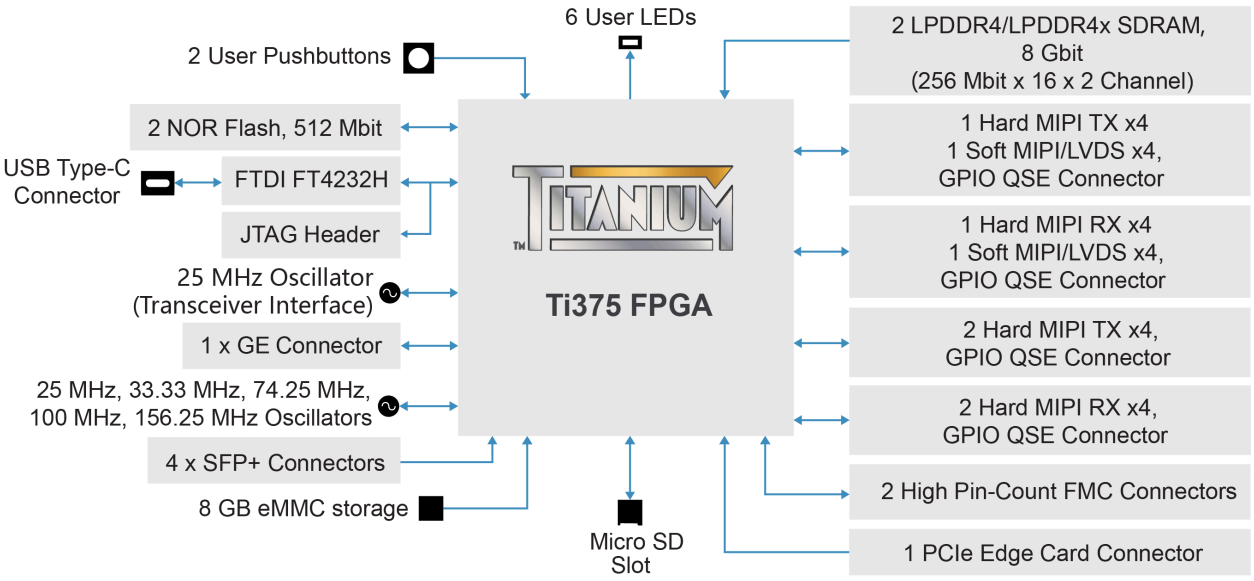
   Note: You are required to use the default driver for Interface 2 when you connect the Titanium Ti375 N1156 Development Board (which uses UART interface) to your computer.

5. Select **libusb-win32** in the **Driver** drop-down list.
6. Click **Replace Driver**.
7. Close the Zadig software.
8. Repeat steps 1–7 for interface 0, which is used for the RISC-V debugger.

Board Functional Description

The Titanium Ti375 N1156 Development Board contains a variety of components to help you build designs for the Titanium Ti375 FPGA.

Figure 1: Titanium Ti375 N1156 Development Board Block Diagram



Features

- Efinix Ti375N1156C4[1] FPGA in a 1156-ball FineLine BGA package
- Two 8 Gbit (32 Mbit x 16 DQ x 8 banks x 2 channels) LPDDR4/4x SDRAM:
  - Supports x32 data width
  - Read/write speed up to 3.3 Gbps
- Two 512 Mbit SPI NOR flash memories
  - Supports single, dual, and quad mode for each SPI flash
  - Supports x8 with both SPI flash devices in quad mode
- Eight GByte eMMC
- Four high-speed QSE connectors that support up to 8 channels (4 data lanes + 1 clock lane) TX/RX MIPI interfaces or LVDS/GPIO
  - 2 1.5 Gbps TX/RX soft MIPI interfaces
  - 3 2.5 Gbps TX hard MIPI interfaces
  - 3 2.5 Gbps RX hard MIPI interfaces
- Four SFP+ receptacles for transceiver interface
- One PCIe edge card connector
- Gigabit Ethernet interface supporting RGMII standard and complies with 10 Base-T, 100 Base-TX, and 1,000 Base-T IEEE 802.3 standards
- Micro-SD card slot
- Two FPGA mezzanine card (FMC) with high pin-count (HPC) connector
- USB type-C connector to configure the development board
- 25 MHz, 33.33 MHz, 74.25 MHz, 100 MHz and 156.25 MHz oscillators for Ti375 clock input
- User LEDs and switches:
  - 6 LEDs
  - 2 pushbutton switches
- Power:
  - 12.0 V power supply connector
  - On-board regulator sources: 0.85 V, 0.95 V, 1.1 V, 1.2 V, 1.8 V, 3.3 V, 5.0 V
- Power good and Ti375 configuration done LEDs

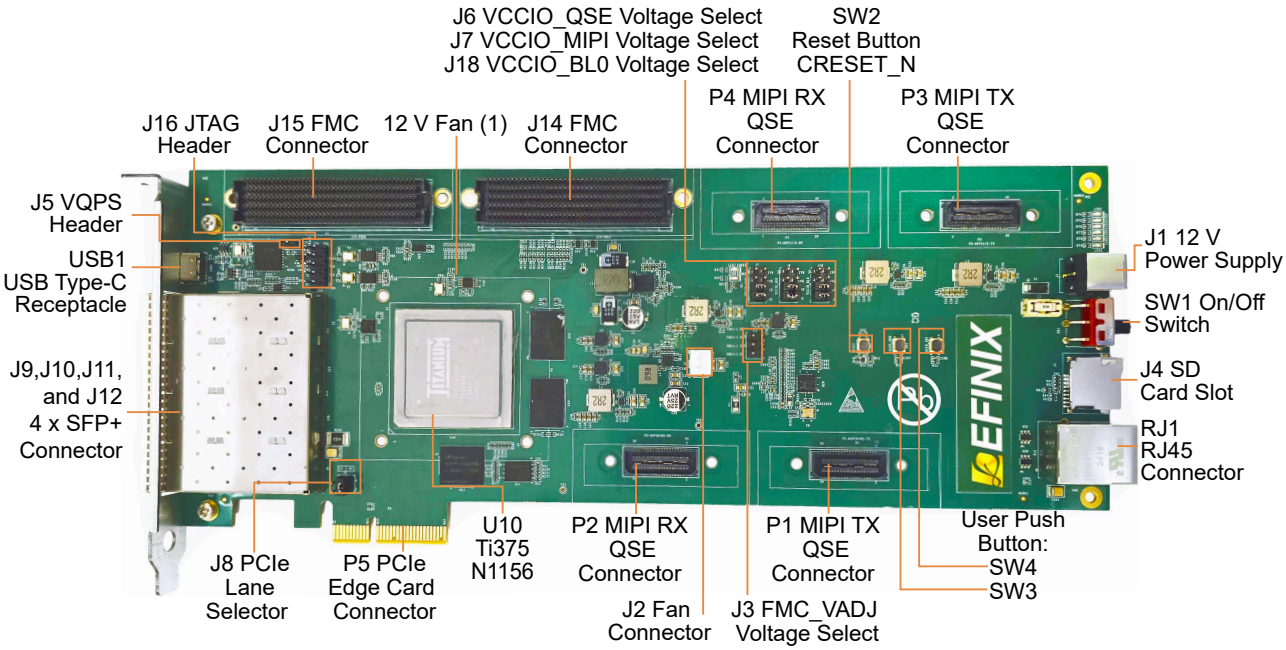
[1] The FPGA speed grade may vary and is subject to availability.

Overview

The board features the Efinix Ti375 FPGA in a 1156-ball FBGA package, which is fabricated using Efinix Quantum® technology. The Quantum®-accelerated programmable logic and routing fabric is wrapped with an I/O interface in a small footprint package. Ti375 FPGAs also include embedded memory blocks and DSP blocks. You create designs for the Ti375 FPGA in the Efinity® software, and then download the resulting configuration bitstream to the board using the USB connection.

Learn more: For more information on Ti375 FPGAs, refer to the Ti375 Data Sheet.

Figure 2: Titanium Ti375 N1156 Development Board Components



Note: (1) You can install the 12 V fan with thermal pad on the development board when needed. Both the thermal pad and the 12 V fan comes with the development kit. See the chapter on "Installing Thermal Pad and Cooling Fan" for installation instructions.

The Titanium Ti375 N1156 Development Board provides four multi-purpose 0.8 mm high-speed ground plane sockets. These sockets can be used for GPIO, MIPI CSI-2 TX/RX, and LVDS TX/RX. The board includes a USB type-C port for the FTDI interface.

The FTDI FT4232H module supports the following interfaces:

- *FTDI interface 0*—JTAG for RISC-V debugging
- *FTDI interface 1*— JTAG for FPGA and RISC-V debugging
- *FTDI interface 2*— UART Communication
- *FTDI interface 3*— PMIC Configuration/reserved

The FTDI module receives the Ti375 configuration bitstream from a USB host and writes to the Ti375 FPGA in SPI active configuration. You can write a configuration bitstream to the on-board SPI NOR flash memory through JTAG with the JTAG SPI Flash Loader Core. Additionally, it supports a UART interface to the Ti375.

The SPI NOR flash memory stores the configuration bitstream. The Ti375 FPGA uses this configuration bitstream when it is in active configuration mode (default).

Learn more: Refer to the Titanium Ti375 N1156 Development Board Schematics and BOM for more information about the components used.

Power On

To turn on the development board, turn on switch SW1. The 12 V DC power is input to the on-board regulators to generate the required 0.85 V, 0.95 V, 1,1 V, 1.2 V, 1.8 V, 3.3 V, and 5.0 V for components on the board. When these voltages are up and stable, the power-good LEDs, LED7 illuminate, giving you a visual confirmation of the status.

Reset

Note: You can manually assert the high-low-high transition with pushbutton switch SW2.

`CRESET_N` has a pull-up resistor. When you press SW2, the board drives `CRESET_N` low; when you release SW2, the board drives `CRESET_N` high. Thus, a single press of SW2 provides the required high-low-high transition.

After toggling `CRESET_N`, the Ti375 FPGA goes into configuration mode and reads the configuration bitstream from the flash memory. When configuration completes successfully, the FPGA drives the `CDONE` signal high. `CDONE` is connected to an LED (LED1), which turns on when the Ti375 FPGA enters user mode.

Configuration

Note: You need to use Efinity software version 2024.1 patch 4 or higher.

You can configure the Ti375 FPGA using the following configuration modes:

- JTAG
- SPI Active (up to x8) via JTAG Bridge

You can use the JTAG bridge mode when programming the flash. You must use the JTAG Bridge when programming the flash because the SPI active signals are not routed directly to FT4232H on the Titanium Ti375 N1156 Development Board. When generating bitstreams for you own design, ensure that you select the **Active** option in the **Bitstream Generation** tab of the Efinity **Project Editor**. Under **Programming Mode**, you can select **SPI active x1**, **SPI active x4** or **SPI active x8**. The flash devices do not support **SPI active x2**.

For PCIe applications, to meet the PCIe boot-up time requirement, Efinix recommends to use the **SPI active x4** mode and internal oscillator with a clock divider **DIV2** for the SPI clock.

The Titanium Ti375 N1156 Development Board does not support internal reconfiguration for remote updates.

Note: For more details on the JTAG SPI bridge loader, refer to **Efinity Software User Guide**.

Clock Sources

Six onboard oscillators (25 MHz, 33.3333 MHz, 74.25 MHz, 100 MHz, and 156.25 MHz) are available to drive the Ti375 PLL input pins and clock inputs. Additionally, two dedicated 25 MHz onboard oscillators serve as the Ethernet PHY and FMC-transceiver interface clock sources respectively, and one dedicated 12 MHz onboard oscillator for USB FTDI FT4232H.

Table 1: Oscillator and Clock Generator Sources

Clock Source	Signal Name	Ti375 Pin Name	PLL Resource
12 MHz oscillator for FTDI FT4232H	-	-	-
25 MHz oscillator	CLK_25M_0	TR1_GPIOR_97/MIPI2_PLL_REFCLK	MIPI2
	CLK_25M_1	TL1_GPIOL_46/MIPI1_PLL_REFCLK	MIPI1
	CLK_25M_2	TR1_GPIOR_88_PLLIN1	TR1
	CLK_25M_3	BL2_GPIOL_23/MIPI0_PLL_REFCLK	MIPI0
25 MHz oscillator for FMC-transceiver interface	-	-	-
25 MHz oscillator for Ethernet PHY	-	-	-
33.33 MHz oscillator	CLK_33.33M	BR1_GPIOR_145_PLLIN1	BR1
74.25 MHz oscillator	CLK_74.25M	TR0_GPIOR_93_PLLIN1	TR0
100 MHz oscillator	CLK_100M	TL0_GPIOL_52_PLLIN1	TL0
100 MHz oscillator	CLK_100M_P	Q0_REFCLK1_P	-
	CLK_100M_N	Q0_REFCLK1_N	-
156.25 MHz oscillator	CLK_156.25M_P	Q1_REFCLK0_P	-
	CLK_156.25M_N	Q1_REFCLK0_N	-

Note: The Efinity Interface Designer issues an unrouted clkmux input error if more than 8 GCLK resources are used on the left side of Ti375 FPGA. To solve this, assign one of the clocks, for example MIPI clock TX0, to use the RCLK instead of the GCLK. For more information, refer to the Clock and Control Network section of the Ti375 Data Sheet.

eMMC

The Titanium Ti375 N1156 Development Board provides an 8 GB eMMC 1.8 V function on board and can be used to store data. This function allows the data transfer of 400 MBps using clock frequency of 200 MHz in HS400 mode. The signals are connected to the HSIO of bank 4A.

Note: For more information on the eMMC, refer to the Titanium Ti375 N1156 Development Board Schematics and BOM.

Table 2: eMMC Pin Assignment

Pin name	Signal name	Ti375 Pin Name
DAT0	EMMC_DATA0	4A_GPIOB_P_50
DAT1	EMMC_DATA1	4A_GPIOB_N_48
DAT2	EMMC_DATA2	4A_GPIOB_P_51
DAT3	EMMC_DATA3	4A_GPIOB_N_50
DAT4	EMMC_DATA4	4A_GPIOB_N_47
DAT5	EMMC_DATA5	4A_GPIOB_P_48
DAT6	EMMC_DATA6	4A_GPIOB_N_49
DAT7	EMMC_DATA7	4A_GPIOB_P_49
RCLK/DS	EMMC_DS	4A_GPIOB_P_46_EXTFB
CMD	EMMC_CMD	4A_GPIOB_N_45
CLK	EMMC_CLK	4A_GPIOB_P_45_PLLIN0
RST_N	EMMC_RSTn	4A_GPIOB_N_46
RCLK/DS	EMMC_DS	4A_GPIOB_P_46_EXTFB

Ethernet PHY

The Titanium Ti375 N1156 Development Board provides an Ethernet PHY function. The operating characteristics of the Ethernet PHY are as follows:

- The Gigabit Ethernet (GE) PHY works in RGMII to copper mode.
- The RGMII interface of the GE PHY can configure the delay of `RXC` and `TXC` by pulling up or down the `RXDELAY` and `TXDELAY` signals to adjust the receive or transmit timing of. By default, the receiver side of the GE PHY (`RXDELAY`) has a 2 ns delay, which is added to `RXC` while `TXDELAY` is pulled down with no delay added.
- In the data trasmission of `CFG_LDO[1:0]=2bit10`, the RGMII works with a supply of 1.8 V. By default, CFG_LDO1 pin is pulled up while the CFG_LDO0 pin is pulled down.
- The power supplies for the GE PHY are as follows:
  - DVDD33, AVDD33: 3.3 V
  - DVDD10, AVDD10: 1.0 V
  - DVDD_RG: RGMII I/O Pad Power

Note: For more information on the Ethernet PHY, refer to the Titanium Ti375 N1156 Development Board Schematics and BOM

Cooling Fan

The Titanium Ti375 N1156 Development Board includes a cooling fan to lower the temperature on the development board. A 12-V fan is used to cool down the FPGA and is connected to the board through the J2 header.

Note: For more information on the temperature control circuits, refer to the **Titanium Ti375 N1156 Development Board Schematics and BOM**.

Headers

The board contains a variety of headers to provide power inputs, signal inputs, and outputs, and to communicate with external devices or boards.

Table 3: Titanium Ti375 N1156 Development Board Headers

Reference Designators	Description
P1 (QSE3)	40-pin multi-purpose high-speed QSE connector for MIPI TX
P2 (QSE2)	40-pin multi-purpose high-speed QSE connector for MIPI RX
P3 (QSE1)	40-pin multi-purpose high-speed QSE connector for MIPI TX
P4 (QSE0)	40-pin multi-purpose high-speed QSE connector for MIPI RX
P5	PCIe edge card connector
J1	12 V DC power input jack
J2	12-V fan connector
J3	FMC_VADJ power supply selector
J4	Micro SD card slot
J5	VQPS enable header
J6	VCCIO_QSE power supply selector
J7	VCCIO_MIPI power supply selector
J8	PCIe lane selector
J9, J10, J11, and J12	20-pin SFP+ receptacle for transceiver interface
J13	SFP+ (10G) interface ground
J14, J15	400-pin high-speed FMC connector for transceiver interface
J16	JTAG header
J18	VCCIO_BL0 power supply selector
RJ1	RJ-45 connector
USB1	USB Type-C receptacle (FTDI FT4232H)
TP1, TP2, TP3, TP4	Ground test point

Headers P1, P2, P3, and P4 (Multi-Purpose)

P1, P2, P3, and P4 are 40-pin multi-purpose high-speed QSE interface connectors. Each connector supports 1 clock lane and 4 data lanes:

- P1 (QSE3) for MIPI TX
- P2 (QSE2) for MIPI RX
- P3 (QSE1) for MIPI TX
- P4 (QSE0) for MIPI RX

Table 4: P1 Pin Assignment

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
1	VCC_3V3	N.C.	2	HS_MIPI_TXDP0	4D_GPIOB_P _03_PLLIN0
3	VCC_5V	N.C.	4	HS_MIPI_TXDN0	4D_GPIOB_N _03_CDI31
5	GND	N.C.	6	GND	N.C.
7	MIPI0_TXDP0	MIPI0_TXDP0	8	HS_MIPI_TXDP1	4D_GPIOB_P_04 _CDI30_EXTFB
9	MIPI0_TXDN0	MIPI0_TXDN0	10	HS_MIPI_TXDN1	4D_GPIOB_N _04_CDI29
11	GND	N.C.	12	GND	N.C.
13	MIPI0_TXDP1	MIPI0_TXDP1	14	HS_MIPI_TXDP2	4D_GPIOB_P_05 _CDI28_PLLIN0
15	MIPI0_TXDN1	MIPI0_TXDN1	16	HS_MIPI_TXDN2	4D_GPIOB_N _05_CDI27
17	GND	N.C.	18	GND	N.C.
19	MIPI0_TXDP2	MIPI0_TXDP2	20	HS_MIPI_TXDP3	4D_GPIOB_P_06 _CDI26_EXTFB
21	MIPI0_TXDN2	MIPI0_TXDN2	22	HS_MIPI_TXDN3	4D_GPIOB_N _06_CDI25
23	GND	N.C.	24	GND	N.C.
25	MIPI0_TXDP3	MIPI0_TXDP3	26	HS_MIPI_TXDP4	4D_GPIOB_P_07 _CDI24_PLLIN0
27	MIPI0_TXDN3	MIPI0_TXDN3	28	HS_MIPI_TXDN4	4D_GPIOB_N _07_CDI23
29	GND	N.C.	30	GND	N.C.
31	MIPI0_TXDP4	MIPI0_TXDP4	32	QSE3_GPIO_3	BL3_GPIOL_28
33	MIPI0_TXDN4	MIPI0_TXDN4	34	QSE3_GPIO_4	BL3_GPIOL_29
35	GND	N.C.	36	GND	N.C.
37	QSE3_GPIO_1	BL3_GPIOL _26_CLK24	38	QSE3_GPIO_5	BL3_GPIOL_30
39	QSE3_GPIO_2	BL3_GPIOL _27_CLK25	40	QSE3_GPIO_6	BL3_GPIOL_31

Table 5: P2 Pin Assignment

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
1	VCC_3V3	N.C.	2	HS_MIPI_RXDP0	4D_GPIOB_P_10
3	VCC_5V	N.C.	4	HS_MIPI_RXDN0	4D_GPIOB_N_10
5	GND	N.C.	6	GND	N.C.
7	MIPI0_RXDP0	MIPI0_RXDP0	8	HS_MIPI_RXDP1	4D_GPIOB_P_11
9	MIPI0_RXDN0	MIPI0_RXDN0	10	HS_MIPI_RXDN1	4D_GPIOB_N_11
11	GND	N.C.	12	GND	N.C.
13	MIPI0_RXDP1	MIPI0_RXDP1	14	HS_MIPI_RXDP2	4D_GPIOB_P_12
15	MIPI0_RXDN1	MIPI0_RXDN1	16	HS_MIPI_RXDN2	4D_GPIOB_N_12
17	GND	N.C.	18	GND	N.C.
19	MIPI0_RXDP2	MIPI0_RXDP2	20	HS_MIPI_RXDP3	4D_GPIOB_P_13 _CDI21_CLK0_P
21	MIPI0_RXDN2	MIPI0_RXDN2	22	HS_MIPI_RXDN3	4D_GPIOB_N_13 _CDI20_CLK0_N
23	GND	N.C.	24	GND	N.C.
25	MIPI0_RXDP3	MIPI0_RXDP3	26	HS_MIPI_RXDP4	4D_GPIOB_P_14 _CDI19_CLK1_P
27	MIPI0_RXDN3	MIPI0_RXDN3	28	HS_MIPI_RXDN4	4D_GPIOB_N_14 _CDI18_CLK1_N
29	GND	N.C.	30	GND	N.C.
31	MIPI0_RXDP4	MIPI0_RXDP4	32	QSE2_GPIO_3	BL1_GPIOL_12
33	MIPI0_RXDN4	MIPI0_RXDN4	34	QSE2_GPIO_4	BL1_GPIOL_13
35	GND	N.C.	36	GND	N.C.
37	QSE2_GPIO_1	BL1_GPIOL_10	38	QSE2_GPIO_5	BL1_GPIOL_16
39	QSE2_GPIO_2	BL1_GPIOL_11	40	QSE2_GPIO_6	BL1_GPIOL_17

Table 6: P3 Pin Assignment

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
1	VCC_3V3	N.C.	2	MIPI2_TXDP0	MIPI2_TXDP0
3	VCC_5V	N.C.	4	MIPI2_TXDN0	MIPI2_TXDN0
5	GND	N.C.	6	GND	N.C.
7	MIPI1_TXDP0	MIPI1_TXDP0	8	MIPI2_TXDP1	MIPI2_TXDP1
9	MIPI1_TXDN0	MIPI1_TXDN0	10	MIPI2_TXDN1	MIPI2_TXDN1
11	GND	N.C.	12	GND	N.C.
13	MIPI1_TXDP1	MIPI1_TXDP1	14	MIPI2_TXDP2	MIPI2_TXDP2
15	MIPI1_TXDN1	MIPI1_TXDN1	16	MIPI2_TXDN2	MIPI2_TXDN2
17	GND	N.C.	18	GND	N.C.
19	MIPI1_TXDP2	MIPI1_TXDP2	20	MIPI2_TXDP3	MIPI2_TXDP3
21	MIPI1_TXDN2	MIPI1_TXDN2	22	MIPI2_TXDN3	MIPI2_TXDN3
23	GND	N.C.	24	GND	N.C.
25	MIPI1_TXDP3	MIPI1_TXDP3	26	MIPI2_TXDP4	MIPI2_TXDP4
27	MIPI1_TXDN3	MIPI1_TXDN3	28	MIPI2_TXDN4	MIPI2_TXDN4
29	GND	N.C.	30	GND	N.C.
31	MIPI1_TXDP4	MIPI1_TXDP4	32	QSE1_GPIO_3	TL0_GPIOL_43
33	MIPI1_TXDN4	MIPI1_TXDN4	34	QSE1_GPIO_4	TL0_GPIOL_44
35	GND	N.C.	36	GND	N.C.
37	QSE1_GPIO_1	TL0_GPIOL_41	38	QSE1_GPIO_5	BL3_GPIOL_32_PLLIN1
39	QSE1_GPIO_2	TL0_GPIOL_42	40	QSE1_GPIO_6	BL3_GPIOL_34_PLLIN1

Table 7: P4 Pin Assignment

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
1	VCC_3V3	N.C.	2	MIPI2_RXDP0	MIPI2_RXDP0
3	VCC_5V	N.C.	4	MIPI2_RXDN0	MIPI2_RXDN0
5	GND	N.C.	6	GND	N.C.
7	MIPI1_RXDP0	MIPI1_RXDP0	8	MIPI2_RXDP1	MIPI2_RXDP1
9	MIPI1_RXDN0	MIPI1_RXDN0	10	MIPI2_RXDN1	MIPI2_RXDN1
11	GND	N.C.	12	GND	N.C.
13	MIPI1_RXDP1	MIPI1_RXDP1	14	MIPI2_RXDP2	MIPI2_RXDP2
15	MIPI1_RXDN1	MIPI1_RXDN1	16	MIPI2_RXDN2	MIPI2_RXDN2
17	GND	N.C.	18	GND	N.C.
19	MIPI1_RXDP2	MIPI1_RXDP2	20	MIPI2_RXDP3	MIPI2_RXDP3
21	MIPI1_RXDN2	MIPI1_RXDN2	22	MIPI2_RXDN3	MIPI2_RXDN3
23	GND	N.C.	24	GND	N.C.
25	MIPI1_RXDP3	MIPI1_RXDP3	26	MIPI2_RXDP4	MIPI2_RXDP4
27	MIPI1_RXDN3	MIPI1_RXDN3	28	MIPI2_RXDN4	MIPI2_RXDN4
29	GND	N.C.	30	GND	N.C.
31	MIPI1_RXDP4	MIPI1_RXDP4	32	QSE0_GPIO_3	TL0_GPIOL_37
33	MIPI1_RXDN4	MIPI1_RXDN4	34	QSE0_GPIO_4	TL0_GPIOL_38
35	GND	N.C.	36	GND	N.C.
37	QSE0_GPIO_1	TL0_GPIOL_35	38	QSE0_GPIO_5	TL0_GPIOL_39
39	QSE0_GPIO_2	TL0_GPIOL_36	40	QSE0_GPIO_6	TL0_GPIOL_40

Header P5 (PCIe Edge Card Connector)

This connector is a standard PCIe edge card connector that fits into an edge card socket. This connector goes to the Ti375 FPGA's PCIe transceiver bank (quad 0). It supports one (x1) or four (x4) link widths to suit different bandwidth requirements.

Table 8: P5 Pin Assignment

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
A1	PCIE_PRSNT1n	N.C.	B1	VCC_12V	N.C.
A2	VCC_12V	N.C.	B2	VCC_12V	N.C.
A3	VCC_12V	N.C.	B3	VCC_12V	N.C.
A4	GND	N.C.	B4	GND	N.C.
A5	-	-	B5	-	-
A6	-	-	B6	-	-
A7	-	-	B7	GND	N.C.
A8	-	-	B8	-	-
A9	-	-	B9	-	-
A10	-	-	B10	-	-
A11	PCIE_EDGE_PERSTn	BR0_GPIOR_141 _PERST_Q0_N	B11	PCIE_EDGE_WAKEn	BR0_GPIOR_144 _PERST_Q2_N
A12	GND	N.C.	B12	-	-
A13	PCIE_EDGE _REFCLK_P	Q0_REFCLK0_P	B13	GND	BR0_GPIOR_141 _PERST_Q0_N
A14	PCIE_EDGE _REFCLK_N	Q0_REFCLK0_N	B14	PCIE_RX0_P	Q0_RXDP3
A15	GND	N.C.	B15	PCIE_RX0_N	Q0_RXDN3
A16	PCIE_TX0_C_P	Q0_TXDP3	B16	GND	N.C.
A17	PCIE_TX0_C_N	Q0_TXDN3	B17	PCIE_PRSNT2n_x1	N.C.
A18	GND	N.C.	B18	GND	N.C.
A19	-	-	B19	PCIE_RX1_P	Q0_RXDP2
A20	GND	N.C.	B20	PCIE_RX1_N	Q0_RXDN2
A21	PCIE_TX1_C_P	Q0_TXDP2	B21	GND	N.C.
A22	PCIE_TX1_C_N	Q0_TXDN2	B22	GND	N.C.
A23	GND	N.C.	B23	PCIE_RX2_P	Q0_RXDP1
A24	GND	N.C.	B24	PCIE_RX2_N	Q0_RXDN1
A25	PCIE_TX2_C_P	Q0_TXDP1	B25	GND	N.C.
A26	PCIE_TX2_C_N	Q0_TXDN1	B26	GND	N.C.
A27	GND	N.C.	B27	PCIE_RX3_P	Q0_RXDP0
A28	GND	N.C.	B28	PCIE_RX3_N	Q0_RXDN0
A29	PCIE_TX3_C_P	Q0_TXDP0	B29	GND	N.C.
A30	PCIE_TX3_C_N	Q0_TXDN0	B30	-	-

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
A31	GND	N.C.	B31	PCIE_PRSNT2n_x4	N.C.
A32	–	–	B32	GND	N.C.

Header J1 (Power Supply)

J1 is a 12 V DC power supply input jack. J1 supplies power to regulators on the board that power the Ti375. The maximum current supply to this input jack is 6 A.

Header J2 (Fan Connector)

J2 is a 2-pin 12 V fan connector.

Table 9: J2 Pin Assignment

Pin Number	Pin Name	Description
1	GND	DC power negative
2	+12 V	DC power positive

Header J3 (FMC_VADJ Power Supply Selector)

J3 is a 6-pin header used to select the voltage for the`FMC_VADJ` signal. By default, the voltage is 1.8 V if the pins are not connected.

Table 10: J3 (FMC_VADJ Power Supply Selector)

Connection	Voltage
N.C.	1.8 V (default)
1 and 2	1.5 V
3 and 4	1.35 V
5 and 6	1.2 V

Header J4 (Micro-SD Card Slot)

Titanium Ti375 N1156 Development Board includes a Micro-SD card slot J4. J4 is connected to the GPIO pins in bank TL. The micro-SD can support data rates up to 25MB/s.

Table 11: J4 Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name
DATA2	SDIO_D2	TL1_GPIOL_45
CD/DAT3	SDIO_D3	TL5_GPIOL_87_PLLIN1
CMD	SDIO_CMD	TL5_GPIOL_85_PLLIN1
VDD	VCC_3V3	N.C.
CLK	SDIO_CK	TL5_GPIOL_86
VSS	GND	-
DAT0	SDIO_D0	TL5_GPIOL_84
DAT1	SDIO_D1	TL5_GPIOL_79_CLK28
CD	TFCard_Detect	TL5_GPIOL_80_CLK29
GND_1	GND	N.C.
GND_2	GND	N.C.
GND_3	GND	N.C.
GND_4	GND	N.C.

Header J5 (VQPS)

J5 is a 2-pin header used to enable the `VQPS`. A jumper is connect to pin 1 and pin 2 to enable the `VQPS`. The VQPS pin is used for programming the security fuses. See **AN 057: Controlling VQPS with the Efinity SVF Player**.

Header J6 (QSE GPIO Power Supply Selector)

J6 is an 8-pin header used to select the power supply for the GPIO pins of the QSE interface.

To select the desired voltage supplies, connect these pins:

- For GPIO voltage of 1.8 V, connect pin 1 and pin 2 or pin 3 and pin 4.
- For GPIO voltage of 3.3 V, connect pin 5 and pin 6 or pin 7 and pin 8.

Table 12: J6 (QSE GPIO Power Supply Selector)

Connection	Voltage
1 and 2	1.8 V
3 and 4	
5 and 6	3.3 V
7 and 8	

Header J7 (MIPI Power Supply Selector)

J7 is an 8-pin header used to select the power supply for the MIPI pins.

To select the desired voltage supplies, connect these pins:

- For MIPI voltage of 1.8 V, connect pin 1 and pin 2 or pin 3 and pin 4.
- For MIPI voltage of 1.2 V, connect pin 5 and pin 6 or pin 7 and pin 8.

Table 13: J7 (MIPI Power Supply Selector)

Connection	Voltage
1 and 2	1.8 V
3 and 4	
5 and 6	1.2 V
7 and 8	

Header J8 (PCIe Lane Selector)

J8 is a 4-pin header that selects either one or four PCI express lanes on the PCIe edge card connector, P5. Refer to the **Titanium Ti375 N1156 Development Board Schematics and BOM** for more information.

Table 14: Header J8 (PCIe Lane Selector)

Connect	Description
1 and 2	PCIE_PRSNT2n_x1; one PCI express lane
3 and 4	PCIE_PRSNT2n_x4; four PCI express lane

Header J9, J10, J11, J12 (SFP+ Transceiver Interface)

J9, J10, J11, J12 are SFP+ receptacles for the transceiver interface. For these SFP+ modules, the transceiver is configured in 10GBASE-R mode based on the IEEE802.3-2008 specifications.

Table 15: J9 Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name
VCCR	VCC_3V3	N.C.
VCCT	VCC_3V3	N.C.
TX_FAULT	SFP_A_TX_FAULT	BR1_GPIOR_152
TX_DISABLE	SFP_A_TX_DISABLE	BR1_GPIOR_153
RX_LOS	SFP_A_RX_LOS	BR1_GPIOR_154
RS0	SFP_A_RS0	BR3_GPIOR_165_PLLIN1
RS1	SFP_A_RS1	BR3_GPIOR_166
MOD_ABS	SFP_A_MOD	BR3 _GPIOR_167
SCL	SFP_A_SCL	BR3_GPIOR_168
SDA	SFP_A_SDA	BR3_GPIOR_169
RD_P	SFP_A_RX_P	Q1_RXDP0

Pin Name	Signal Name	Ti375 Pin Name
RD_N	SFP_A_RX_N	Q1_RXDN0
TD_P	SFP_A_TX_P	Q1_TXDP0
TD_N	SFP_A_TX_N	Q1_TXDN0
VEET	GND	N.C.
VEER	GND	N.C.

Table 16: J10 Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name
VCCR	VCC_3V3	N.C.
VCCT	VCC_3V3	N.C.
TX_FAULT	SFP_B_TX_FAULT	BR4_GPIOR_180
TX_DISABLE	SFP_B_TX_DISABLE	BR4_GPIOR_179
RX_LOS	SFP_B_RX_LOS	BR3_GPIOR_171
RS0	SFP_B_RS0	BR4_GPIOR_175
RS1	SFP_B_RS1	BR3_GPIOR_170
MOD_ABS	SFP_B_MOD	BR4_GPIOR_176
SCL	SFP_B_SCL	BR4_GPIOR_ 178
SDA	SFP_B_SDA	BR4_GPIOR_177
RD_P	SFP_B_RX_P	Q1_RXDP1
RD_N	SFP_B_RX_N	Q1_RXDN1
TD_P	SFP_B_TX_P	Q1_TXDP1
TD_N	SFP_B_TX_N	Q1_TXDN1
VEET	GND	N.C.
VEER	GND	N.C.

Table 17: J11 Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name
VCCR	VCC_3V3	N.C.
VCCT	VCC_3V3	N.C.
TX_FAULT	SFP_C_TX_FAULT	TR5_GPIOR_126
TX_DISABLE	SFP_C_TX_DISABLE	TR5_GPIOR_128
RX_LOS	SFP_C_RX_LOS	TR3_GPIOR_113
RS0	SFP_C_RS0	TR3_GPIOR_109
RS1	SFP_C_RS1	TR3_GPIOR_110
MOD_ABS	SFP_C_MOD	TR3_GPIOR_112
SCL	SFP_C_SCL	TR3_GPIOR_111
SDA	SFP_C_SDA	TR5_GPIOR_125
RD_P	SFP_C_RX_P	Q1_RXDP2

Pin Name	Signal Name	Ti375 Pin Name
RD_N	SFP_C_RX_N	Q1_RXDN2
TD_P	SFP_C_TX_P	Q1_TXDP2
TD_N	SFP_C_TX_N	Q1_TXDN2
VEET	GND	N.C.
VEER	GND	N.C.

Table 18: J12 Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name
VCCR	VCC_3V3	N.C,
VCCT	VCC_3V3	N.C,
TX_FAULT	SFP_D_TX_FAULT	TR3_GPIOR_114
TX_DISABLE	SFP_D_TX_DISABLE	TR3_GPIOR_115
RX_LOS	SFP_D_RX_LOS	TR5_GPIOR_132
RS0	SFP_D_RS0	TR5_GPIOR_131
RS1	SFP_D_RS1	TR5_GPIOR_129
MOD_ABS	SFP_D_MOD	TR5_GPIOR_133
SCL	SFP_D_SCL	TR5_GPIOR_134
SDA	SFP_D_SDA	TR3_GPIOR_116
RD_P	SFP_D_RX_P	Q1_RXDP3
RD_N	SFP_D_RX_N	Q1_RXDN3
TD_P	SFP_D_TX_P	Q1_TXDP3
TD_N	SFP_D_TX_N	Q1_TXDN3
VEET	GND	N.C,
VEER	GND	N.C,

Header J13 (SFP+ Interface Ground)

J13 is the SFP+ (10G) interface ground assembly for the J9, J10, J11, and J12 SFP+ connector.

Headers J14 and J15 (FMC-Transceiver)

J14 and J15 are high-speed FMC HPC connectors for the transceiver interface that support 2 GTX clocks and 4 differential clocks. The transceiver is configured in 10GBASE-KR mode based on the IEEE802.3ap-2007 specifications.

Table 19: J14 (Row A) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
A1	GND	N.C.	A2	FMCA_DP1 _M2C_P	Q3_RXDP1
A3	FMCA_DP1 _M2C_N	Q3_RXDN1	A4	GND	N.C.
A5	GND	N.C.	A6	FMCA_DP2 _M2C_P	Q3_RXDP2
A7	FMCA_DP2 _M2C_N	Q3_RXDN2	A8	GND	N.C.
A9	GND	N.C.	A10	FMCA_DP3 _M2C_P	Q3_RXDP3
A11	FMCA_DP3 _M2C_N	Q3_RXDN3	A12	GND	N.C.
A13	GND	N.C.	A14	-	-
A15	-	-	A16	GND	N.C.
A17	GND	N.C.	A18	-	-
A19	-	-	A20	GND	N.C.
A21	GND	N.C.	A22	FMCA_DP1 _C2M_P	Q3_TXDP1
A23	FMCA_DP1 _C2M_N	Q3_TXDN1	A24	GND	N.C.
A25	GND	N.C.	A26	FMCA_DP2 _C2M_P	Q3_TXDP2
A27	FMCA_DP2 _C2M_N	Q3_TXDN2	A28	GND	N.C.
A29	GND	N.C.	A30	FMCA_DP3 _C2M_P	Q3_TXDP3
A31	FMCA_DP3 _C2M_N	Q3_TXDN3	A32	GND	N.C.
A33	GND	N.C.	A34	-	-
A35	-	-	A36	GND	N.C.
A37	GND	N.C.	A38	-	-
A39	-	-	A40	GND	N.C.

Table 20: J14 (Row B) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
B1	–	–	B2	GND	N.C.
B3	GND	N.C.	B4	–	–
B5	–	-	B6	GND	N.C.
B7	GND	N.C.	B8	–	–
B9	–	–	B10	GND	N.C.
B11	GND	N.C.	B12	–	–
B13	–	–	B14	GND	N.C.
B15	GND	N.C.	B16	–	–
B17	–	–	B18	GND	N.C.
B19	GND	N.C.	B20	–	–
B21	–	–	B22	GND	N.C.
B23	GND	N.C.	B24	–	–
B25	–	–	B26	GND	N.C.
B27	GND	N.C.	B28	–	–
B29	–	–	B30	GND	N.C.
B31	GND	N.C.	B32	–	–
B33	–	–	B34	GND	N.C.
B35	GND	N.C.	B36	–	–
B37	–	–	B38	GND	N.C.
B39	GND	N.C.	B40	–	–

Table 21: J14 (Row C) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
C1	GND	N.C.	C2	FMCA_DP0_C2M_P	Q3_TXDP0
C3	FMCA_DP0_C2M_N	Q3_TXDN0	C4	GND	N.C.
C5	GND	N.C.	C6	FMCA_DP0_M2C_P	Q3_RXDP0
C7	FMCA_DP0_M2C_N	Q3_RXDN0	C8	GND	N.C.
C9	GND	N.C.	C10	FMCA_LA6_P	2C_GPIOT_P _24_EXTFB
C11	FMCA_LA6_N	2C_GPIOT_N_24	C12	GND	N.C.
C13	GND	N.C.	C14	FMCA_LA10_P	2B_GPIOT_P _14_EXTFB
C15	FMCA_LA10_N	2B_GPIOT_N_14	C16	GND	N.C.
C17	GND	N.C.	C18	FMCA_LA14_P	2A_GPIOT_P_01
C19	FMCA_LA14_N	2A_GPIOT_N_01	C20	GND	N.C.
C21	GND	N.C.	C22	FMCA_LA18_CC_P_T	2B_GPIOT_P _12_CLK31_P

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
C23	FMCA_LA18_CC_N_T	2B_GPIOT_N _12_CLK31_N	C24	GND	N.C.
C25	GND	N.C.	C26	FMCA_LA27_P	2C_GPIOT_P_30
C27	FMCA_LA27_N	2C_GPIOT_N_30	C28	GND	N.C.
C29	GND	N.C.	C30	FMCA_SCL	BL0_GPIOL_08
C31	FMCA_SDA	BL0_GPIOL_07	C32	GND	N.C.
C33	GND	N.C.	C34	FMCA_GA0	TR1_GPIOR_98
C35	VCC_12V	N.C.	C36	GND	N.C.
C37	VCC_12V	N.C.	C38	GND	N.C.
C39	VCC_3V3	N.C.	C40	GND	N.C.

Table 22: J14 (Row D) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
D1	FMCA_C2M_PG	TR0_GPIOR_89	D2	GND	N.C.
D3	GND	N.C.	D4	FMCA_GBTCLK _M2C_P	Q3_REFCLK0_P
D5	FMCA_GBTCLK _M2C_N	Q3_REFCLK0_N	D6	GND	N.C.
D7	GND	N.C.	D8	FMCA_LA1_CC_P	2B_GPIOT_P _10_CLK27_P
D9	FMCA_LA1_CC_N	2B_GPIOT_N _10_CLK27_N	D10	GND	N.C.
D11	FMCA_LA5_P	2B_GPIOT_P_15	D12	FMCA_LA5_N	2B_GPIOT_N_15
D13	GND	N.C.	D14	FMCA_LA9_P	2B_GPIOT_P_ 09_CLK26_P
D15	FMCA_LA9_N	2B_GPIOT_N_ 09_CLK26_N	D16	GND	N.C.
D17	FMCA_LA13_P	2A_GPIOT_P_00	D18	FMCA_LA13_N	2A_GPIOT_N_00
D19	GND	N.C.	D20	FMCA_LA17_P_T	2B_GPIOT_P_19
D21	FMCA_LA17_N_T	2B_GPIOT_N_19	D22	GND	N.C.
D23	FMCA_LA23_P	2A_GPIOT_P_04	D24	FMCA_LA23_N	2A_GPIOT_N_04
D25	GND	N.C.	D26	FMCA_LA26_P	2C_GPIOT_P_25
D27	FMCA_LA26_N	2C_GPIOT_N_25	D28	GND	N.C.
D29	TCK_FMCA	N.C.	D30	TDO_FPGA	BR4_TDO
D31	TDO_FMCA	N.C.	D32	VCC_3V3	N.C.
D33	TMS_FMCA	N.C.	D34	-	-
D35	FMCA_GA1	TR1_GPIOR_99	D36	VCC_3V3	N.C.
D37	GND	N.C.	D38	VCC_3V3	N.C.
D39	GND	N.C.	D40	VCC_3V3	N.C.

Table 23: J14 (Row E) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
E1	GND	N.C.	E2	–	–
E3	–	–	E4	GND	N.C.
E5	GND	N.C.	E6	–	–
E7	–	–	E8	GND	N.C.
E9	–	–	E10	–	–
E11	GND	N.C.	E12	–	–
E13	–	–	E14	GND	N.C.
E15	–	–	E16	–	–
E17	GND	N.C.	E18	–	–
E19	–	–	E20	GND	N.C.
E21	–	–	E22	–	–
E23	GND	N.C.	E24	–	–
E25	–	–	E26	GND	N.C.
E27	–	–	E28	–	–
E29	GND	N.C.	E30	–	–
E31	–	–	E32	GND	N.C.
E33	–	–	E34	–	–
E35	GND	N.C.	E36	–	–
E37	–	–	E38	GND	N.C.
E39	FMC_VADJ	VCCIO2A/2B/2C	E40	GND	N.C.

Table 24: J14 (Row F) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
F1	VCC_3V3 [(2)]	N.C.	F2	GND	N.C.
F3	GND	N.C.	F4	–	–
F5	–	–	F6	GND	N.C.
F7	–	–	F8	–	–
F9	GND	N.C.	F10	–	–
F11	–	–	F12	GND	N.C.
F13	–	–	F14	–	–
F15	GND	N.C.	F16	–	–
F17	–	–	F18	GND	N.C.
F19	–	–	F20	–	–
F21	GND	N.C.	F22	–	–

[(2)] For high pin count versions only.

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
F23	–	–	F24	GND	N.C.
F25	–	–	F26	–	–
F27	GND	N.C.	F28	–	–
F29	–	–	F30	GND	N.C.
F31	–	–	F32	–	–
F33	GND	N.C.	F34	–	–
F35	–	–	F36	GND	N.C.
F37	–	–	F38	–	–
F39	GND	N.C.	F40	FMC_VADJ	VCCIO2A/2B/2C

Table 25: J14 (Row G) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
G1	GND	N.C.	G2	FMCA_CLK1 _M2C_P	2B_GPIOT_P_ 11_CLK30_P
G3	FMCA_CLK1 _M2C_N	2B_GPIOT_N_ 11_CLK30_N	G4	GND	N.C.
G5	GND	N.C.	G6	FMCA_LA0_CC_P	2C_GPIOT_P _23_PLLIN0
G7	FMCA_LA0_CC_N	2C_GPIOT_N_23	G8	GND	N.C.
G9	FMCA_LA3_P	2B_GPIOT_P_17	G10	FMCA_LA3_N	2B_GPIOT_N_17
G11	GND	N.C.	G12	FMCA_LA8_P	2A_GPIOT_P _07_EXTFB
G13	FMCA_LA8_N	2A_GPIOT_N_07	G14	GND	N.C.
G15	FMCA_LA12_P	2A_GPIOT_P_06	G16	FMCA_LA12_N	2A_GPIOT_N_06
G17	GND	N.C.	G18	FMCA_LA16_P	2B_GPIOT_P_18
G19	FMCA_LA16_N	2B_GPIOT_N_18	G20	GND	N.C.
G21	FMCA_LA20_P_T	2B_GPIOT_P_21	G22	FMCA_LA20_N_T	2B_GPIOT_N_21
G23	GND	N.C.	G24	FMCA_LA22_P	2C_GPIOT_P_28
G25	FMCA_LA22_N	2C_GPIOT_N_28	G26	GND	N.C.
G27	FMCA_LA25_P	2C_GPIOT_P_27	G28	FMCA_LA25_N	2C_GPIOT_N_27
G29	GND	N.C.	G30	FMCA_LA29_P	2C_GPIOT_P_29
G31	FMCA_LA29_N	2C_GPIOT_N_29	G32	GND	N.C.
G33	FX10_CDI0	4B_GPIOB_N _40_CDI0	G34	FX10_CDI1	4B_GPIOB_P_ 40_CDI1_EXTFB
G35	GND	N.C.	G36	FX10_CDI4	4B_GPIOB_N _37_CDI4
G37	FX10_CDI5	4B_GPIOB_P _37_CDI5	G38	GND	N.C.
G39	FMC_VADJ	VCCIO2A/2B/2C	G40	GND	N.C.

Table 26: J14 (Row H) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
H1	–	–	H2	FMCA_PRSNT	TR0_GPIOR_90
H3	GND	N.C.	H4	FMCA_CLK0 _M2C_P	2B_GPIOT_P _13_PLLIN0
H5	FMCA_CLK0 _M2C_N	2B_GPIOT_N_13	H6	GND	N.C.
H7	FMCA_LA2_P	2B_GPIOT_P_16	H8	FMCA_LA2_N	2B_GPIOT_N_16
H9	GND	N.C.	H10	FMCA_LA4_P	2A_GPIOT_P_03
H11	FMCA_LA4_N	2A_GPIOT_N_03	H12	GND	N.C.
H13	FMCA_LA7_P	2A_GPIOT_P_02	H14	FMCA_LA7_N	2A_GPIOT_N_02
H15	GND	N.C.	H16	FMCA_LA11_P	2A_GPIOT_P_05
H17	FMCA_LA11_N	2A_GPIOT_N_05	H18	GND	N.C.
H19	FMCA_LA15_P	2A_GPIOT_P _08_PLLIN0	H20	FMCA_LA15_N	2A_GPIOT_N_08
H21	GND	N.C.	H22	FMCA_LA19_P_T	2B_GPIOT_P_20
H23	FMCA_LA19_N_T	2B_GPIOT_N_20	H24	GND	N.C.
H25	FMCA_LA21_P_T	2B_GPIOT_P_22	H26	FMCA_LA21_N_T	2B_GPIOT_N_22
H27	GND	N.C.	H28	FMCA_LA24_P_T	2C_GPIOT_P_26
H29	FMCA_LA24_N_T	2C_GPIOT_N_26	H30	GND	N.C.
H31	FMCA_LA28_P	2C_GPIOT_P_31	H32	FMCA_LA28_N	2C_GPIOT_N_31
H33	GND	N.C.	H34	FX10_CDI2	4B_GPIOB_ N_39_CDI2
H35	FX10_CDI3	4B_GPIOB_P _38_CDI3	H36	GND	N.C.
H37	FX10_CDI6	4B_GPIOB_N _36_CDI6	H38	FX10_CDI7	4B_GPIOB_P _36_CDI7
H39	GND	N.C.	H40	FMC_VADJ	VCCIO2A/2B/2C

Table 27: J14 (Row J) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
J1	GND	N.C.	J2	–	–
J3	–	–	J4	GND	N.C.
J5	GND	N.C.	J6	–	–
J7	–	–	J8	GND	N.C.
J9	–	–	J10	–	–
J11	GND	N.C.	J12	–	–
J13	–	–	J14	GND	N.C.
J15	–	–	J16	–	–
J17	GND	N.C.	J18	–	–

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
J19	–	–	J20	GND	N.C.
J21	–	–	J22	–	–
J23	GND	N.C.	J24	–	–
J25	–	–	J26	GND	N.C.
J27	–	–	J28	–	–
J29	GND	N.C.	J30	–	–
J31	–	–	J32	GND	N.C.
J33	–	–	J34	–	–
J35	GND	N.C.	J36	–	–
J37	–	–	J38	GND	N.C.
J39	–	–	J40	GND	N.C.

Table 28: J14 (Row K) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
K1	–	–	K2	GND	N.C.
K3	GND	N.C.	K4	–	–
K5	–	–	K6	GND	N.C.
K7	–	–	K8	–	–
K9	GND	N.C.	K10	–	–
K11	–	–	K12	GND	N.C.
K13	–	–	K14	–	–
K15	GND	N.C.	K16	–	–
K17	–	–	K18	GND	N.C.
K19	–	–	K20	–	–
K21	GND	N.C.	K22	–	–
K23	–	–	K24	GND	N.C.
K25	–	–	K26	–	–
K27	GND	N.C.	K28	–	–
K29	–	–	K30	GND	N.C.
K31	–	–	K32	–	–
K33	GND	N.C.	K34	–	–
K35	–	–	K36	GND	N.C.
K37	–	–	K38	–	–
K39	GND	N.C.	K40	–	–

Table 29: J15 (Row A) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
A1	GND	N.C.	A2	FMCB_DP1_M2C_P	Q2_RXDP2
A3	FMCB_DP1_M2C_N	Q2_RXDN2	A4	GND	N.C.
A5	GND	N.C.	A6	FMCB_DP2_M2C_P	Q2_RXDP1
A7	FMCB_DP2_M2C_N	Q2_RXDN1	A8	GND	N.C.
A9	GND	N.C.	A10	FMCB_DP3_M2C_P	Q2_RXDP0
A11	FMCB_DP3_M2C_N	Q2_RXDN0	A12	GND	N.C.
A13	GND	N.C.	A14	–	–
A15	–	–	A16	GND	N.C.
A17	GND	N.C.	A18	–	–
A19	–	–	A20	GND	N.C.
A21	GND	N.C.	A22	FMCB_DP1_C2M_P	Q2_TXDP2
A23	FMCB_DP1_C2M_N	Q2_TXDN2	A24	GND	N.C.
A25	GND	N.C.	A26	FMCB_DP2_C2M_P	Q2_TXDP1
A27	FMCB_DP2_C2M_N	Q2_TXDN1	A28	GND	N.C.
A29	GND	N.C.	A30	FMCB_DP3_C2M_P	Q2_TXDP0
A31	FMCB_DP3_C2M_N	Q2_TXDN0	A32	GND	N.C.
A33	GND	N.C.	A34	–	–
A35	–	–	A36	GND	N.C.
A37	GND	N.C.	A38	–	–
A39	–	–	A40	GND	N.C.

Table 30: J15 (Row B) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
B1	–	–	B2	GND	N.C.
B3	GND	N.C.	B4	–	–
B5	–	–	B6	GND	N.C.
B7	GND	N.C.	B8	–	–
B9	–	–	B10	GND	N.C.
B11	GND	N.C.	B12	–	–
B13	–	–	B14	GND	N.C.
B15	GND	N.C.	B16	–	–
B17	–	–	B18	GND	N.C.
B19	GND	N.C.	B20	–	–
B21	–	–	B22	GND	N.C.
B23	GND	N.C.	B24	–	–
B25	–	–	B26	GND	N.C.

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
B27	GND	N.C.	B28	–	–
B29	–	–	B30	GND	N.C.
B31	GND	N.C.	B32	–	–
B33	–	–	B34	GND	N.C.
B35	GND	N.C.	B36	–	–
B37	–	–	B38	GND	N.C.
B39	GND	N.C.	B40	–	–

Table 31: J15 (Row C) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
C1	GND	N.C.	C2	FMCB_DP0_C2M_P	Q2_TXDP3
C3	FMCB_DP0_C2M_N	Q2_TXDN3	C4	GND	N.C.
C5	GND	N.C.	C6	FMCB_DP0_M2C_P	Q2_RXDP3
C7	FMCB_DP0_M2C_N	Q2_RXDN3	C8	GND	N.C.
C9	GND	N.C.	C10	FMCB_LA6_P	2C_GPIOT_P_34 _CLK17_P
C11	FMCB_LA6_N	2C_GPIOT_N_34 _CLK17_N	C12	GND	N.C.
C13	GND	N.C.	C14	FMCB_LA10_P	2D_GPIOT_P_46
C15	FMCB_LA10_N	2D_GPIOT_N_46	C16	GND	N.C.
C17	GND	N.C.	C18	FMCB_LA14_P	2D_GPIOT_P_44
C19	FMCB_LA14_N	2D_GPIOT_N_44	C20	GND	N.C.
C21	GND	N.C.	C22	FMCB_LA18_CC_P	2D_GPIOT_P _50_PLLIN0
C23	FMCB_LA18_CC_N	2D_GPIOT_N_50	C24	GND	N.C.
C25	GND	N.C.	C26	FMCB_LA27_P	2D_GPIOT_P_48
C27	FMCB_LA27_N	2D_GPIOT_N_48	C28	GND	N.C.
C29	GND	N.C.	C30	FMCB_SCL	TR2_GPIOR_106
C31	FMCB_SDA	TR2_GPIOR_103	C32	GND	N.C.
C33	GND	N.C.	C34	FMCB_GA0	TR2_GPIOR_107
C35	VCC_12V	N.C.	C36	GND	N.C.
C37	VCC_12V	N.C.	C38	GND	N.C.
C39	VCC_3V3	N.C.	C40	GND	N.C.

Table 32: J15 (Row D) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
D1	FMCB_C2M_PG	TR1_GPIOR_100	D2	GND	N.C.
D3	GND	N.C.	D4	FMCB_GBTCLK _M2C_P	Q2_REFCLK0_P
D5	FMCB_GBTCLK _M2C_N	Q2_REFCLK0_N	D6	GND	N.C.
D7	GND	N.C.	D8	FMCB_LA1_CC_P	2E_GPIOT_P _59_PLLIN0
D9	FMCB_LA1_CC_N	2E_GPIOT_N_59	D10	GND	N.C.
D11	FMCB_LA5_P	2C_GPIOT_P _36_CLK19_P	D12	FMCB_LA5_N	2C_GPIOT_N _36_CLK19_N
D13	GND	N.C.	D14	FMCB_LA9_P	2D_GPIOT_P _49_EXTFB
D15	FMCB_LA9_N	2D_GPIOT_N_49	D16	GND	N.C.
D17	FMCB_LA13_P	2D_GPIOT_P_41	D18	FMCB_LA13_N	2D_GPIOT_N_41
D19	GND	N.C.	D20	FMCB_LA17_P	2D_GPIOT_P_43
D21	FMCB_LA17_N	2D_GPIOT_N_43	D22	GND	N.C.
D23	FMCB_LA23_P	2D_GPIOT_P _39_CLK22_P	D24	FMCB_LA23_N	2D_GPIOT_N _39_CLK22_N
D25	GND	N.C.	D26	FMCB_LA26_P	2D_GPIOT_P_42
D27	FMCB_LA26_N	2D_GPIOT_N_42	D28	GND	N.C.
D29	TCK_FMCB	N.C.	D30	TDO_FMCA	N.C.
D31	TDO_FT	N.C.	D32	VCC_3V3	N.C.
D33	TMS_FMCB	N.C.	D34	VCC_3V3	N.C.
D35	FMCB_GA1	TR2_GPIOR_108	D36	VCC_3V3	N.C.
D37	GND	N.C.	D38	VCC_3V3	N.C.
D39	GND	N.C.	D40	VCC_3V3	N.C.

Table 33: J15 (Row E) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
E1	GND	N.C.	E2	–	–
E3	–	–	E4	GND	N.C.
E5	GND	N.C.	E6	–	–
E7	–	–	E8	GND	N.C.
E9	–	–	E10	–	–
E11	GND	N.C.	E12	–	–
E13	–	–	E14	GND	N.C.
E15	–	–	E16	–	–

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
E17	GND	N.C.	E18	–	–
E19	–	–	E20	GND	N.C.
E21	–	–	E22	–	–
E23	GND	N.C.	E24	–	–
E25	–	–	E26	GND	N.C.
E27	–	–	E28	–	–
E29	GND	N.C.	E30	–	–
E31	–	–	E32	GND	N.C.
E33	–	–	E34	–	–
E35	GND	N.C.	E36	–	–
E37	–	–	E38	GND	N.C.
E39	FMC_VADJ	VCCIO2A/2B/2C	E40	GND	N.C.

Table 34: J15 (Row F) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
F1	VCC_3V3[3]	N.C.	F2	GND	N.C.
F3	GND	N.C.	F4	–	–
F5	–	–	F6	GND	N.C.
F7	–	–	F8	–	–
F9	GND	N.C.	F10	–	–
F11	–	–	F12	GND	N.C.
F13	–	–	F14	–	–
F15	GND	N.C.	F16	–	–
F17	–	–	F18	GND	N.C.
F19	–	–	F20	–	–
F21	GND	N.C.	F22	–	–
F23	–	–	F24	GND	N.C.
F25	–	–	F26	–	–
F27	GND	N.C.	F28	–	–
F29	–	–	F30	GND	N.C.
F31	–	–	F32	–	–
F33	GND	N.C.	F34	–	–
F35	–	–	F36	GND	N.C.
F37	–	–	F38	–	–
F39	GND	N.C.	F40	FMC_VADJ	VCCIO2A/2B/2C

[3] For high pin count versions only.

Table 35: J15 (Row G) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
G1	GND	N.C.	G2	FMCB_CLK1 _M2C_P	2B_GPIOT_P_ 11_CLK30_P
G3	FMCB_CLK1 _M2C_N	2B_GPIOT_N_ 11_CLK30_N	G4	GND	N.C.
G5	GND	N.C.	G6	FMCB_LA0_CC_P	2C_GPIOT_P _23_PLLIN0
G7	FMCB_LA0_CC_N	2C_GPIOT_N_23	G8	GND	N.C.
G9	FMCB_LA3_P	2B_GPIOT_P_17	G10	FMCB_LA3_N	2B_GPIOT_N_17
G11	GND	N.C.	G12	FMCB_LA8_P	2A_GPIOT_P _07_EXTFB
G13	FMCB_LA8_N	2A_GPIOT_N_07	G14	GND	N.C.
G15	FMCB_LA12_P	2A_GPIOT_P_06	G16	FMCB_LA12_N	2A_GPIOT_N_06
G17	GND	N.C.	G18	FMCB_LA16_P	2B_GPIOT_P_18
G19	FMCB_LA16_N	2B_GPIOT_N_18	G20	GND	N.C.
G21	FMCB_LA20_P	2B_GPIOT_P_21	G22	FMCB_LA20_N	2B_GPIOT_N_21
G23	GND	N.C.	G24	FMCB_LA22_P	2C_GPIOT_P_28
G25	FMCB_LA22_N	2C_GPIOT_N_28	G26	GND	N.C.
G27	FMCB_LA25_P	2C_GPIOT_P_27	G28	FMCB_LA25_N	2C_GPIOT_N_27
G29	GND	N.C.	G30	FMCB_LA29_P	2C_GPIOT_P_29
G31	FMCB_LA29_N	2C_GPIOT_N_29	G32	GND	N.C.
G33	–	–	G34	–	–
G35	GND	N.C.	G36	–	–
G37	–	–	G38	GND	N.C.
G39	FMC_VADJ	VCCIO2A/2B/2C	G40	GND	N.C.

Table 36: J15 (Row H) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
H1	–	–	H2	FMCB_PRSNT	TR0_GPIOR_90
H3	GND	N.C.	H4	FMCB_CLK0_M2C_P	2D_GPIOT_P _37_CLK20_P
H5	FMCB_CLK_M2C_N	2D_GPIOT_N _37_CLK20_N	H6	GND	N.C.
H7	FMCB_LA2_P	2E_GPIOT_P_53	H8	FMCB_LA2_N	2E_GPIOT_N_53
H9	GND	N.C.	H10	FMCB_LA4_P	2C_GPIOT_P _33_CLK16_P
H11	FMCB_LA4_N	2C_GPIOT_N _33_CLK16_N	H12	GND	N.C.
H13	FMCB_LA7_P	2E_GPIOT_P_51	H14	FMCB_LA7_N	2E_GPIOT_N_51

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
H15	GND	N.C.	H16	FMCB_LA11_P	2E_GPIOT_P_55
H17	FMCB_LA11_N	2E_GPIOT_N_55	H18	GND	N.C.
H19	FMCB_LA15_P	2E_GPIOT_P_52	H20	FMCB_LA15_N	2A_GPIOT_N_08
H21	GND	N.C.	H22	FMCB_LA19_P	2D_GPIOT_P _40_CLK23_P
H23	FMCB_LA19_N	2D_GPIOT_N _40_CLK23_N	H24	GND	N.C.
H25	FMCB_LA21_P	2E_GPIOT_P _60_EXTFB	H26	FMCB_LA21_N	2E_GPIOT_N_60
H27	GND	N.C.	H28	FMCB_LA24_P	2E_GPIOT_P _64_CLK15_P
H29	FMCB_LA24_N	2E_GPIOT_N _64_CLK15_N	H30	GND	N.C.
H31	FMCB_LA28_P	2E_GPIOT_P _61_EXTFB	H32	FMCB_LA28_N	2E_GPIOT_N_61
H33	GND	N.C.	H34	–	–
H35	–	–	H36	GND	N.C.
H37	–	–	H38	–	–
H39	GND	N.C.	H40	FMC_VADJ	VCCIO2A/2B/2C

Table 37: J15 (Row J) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
J1	GND	N.C.	J2	–	–
J3	–	–	J4	GND	N.C.
J5	GND	N.C.	J6	–	–
J7	–	–	J8	GND	N.C.
J9	–	–	J10	–	–
J11	GND	N.C.	J12	–	–
J13	–	–	J14	GND	N.C.
J15	–	–	J16	–	–
J17	GND	N.C.	J18	–	–
J19	–	–	J20	GND	N.C.
J21	–	–	J22	–	–
J23	GND	N.C.	J24	–	–
J25	–	–	J26	GND	N.C.
J27	–	–	J28	–	–
J29	GND	N.C.	J30	–	–
J31	–	–	J32	GND	N.C.

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
J33	–	–	J34	–	–
J35	GND	N.C.	J36	–	–
J37	–	–	J38	GND	N.C.
J39	–	–	J40	GND	N.C.

Table 38: J15 (Row K) Pin Assignments

Pin Number	Signal Name	Ti375 Pin Name	Pin Number	Signal Name	Ti375 Pin Name
K1	–	–	K2	GND	N.C.
K3	GND	N.C.	K4	–	–
K5	–	–	K6	GND	N.C.
K7	–	–	K8	–	–
K9	GND	N.C.	K10	–	–
K11	–	–	K12	GND	N.C.
K13	–	–	K14	–	–
K15	GND	N.C.	K16	–	–
K17	–	–	K18	GND	N.C.
K19	–	–	K20	–	–
K21	GND	N.C.	K22	–	–
K23	–	–	K24	GND	N.C.
K25	–	–	K26	–	–
K27	GND	N.C.	K28	–	–
K29	–	–	K30	GND	N.C.
K31	–	–	K32	–	–
K33	GND	N.C.	K34	–	–
K35	–	–	K36	GND	N.C.
K37	–	–	K38	–	–
K39	GND	N.C.	K40	–	–

Header J16 (JTAG)

J16 is a 10-pin JTAG interface. You can access the Ti375 JTAG pins through this interface.

Table 39: J16 Pin Assignment

Pin number	Signal Name
1	TDO_FT
2	VCC_3V3
3	TCK_FT
4	TDI_FT
5	TMS_FT
6	FT_RST
7	N.C.
8	CRESET_JTAG_N
9	GND
10	GND

Header J18 (Bank BL0 GPIO Power Supply Selector)

J18 is an 8-pin header used to select the GPIO voltage for Bank BL0.

Table 40: J18 (Bank BL0 GPIO Power Supply Selector)

Connection	Voltage
1 and 2	1.8V
3 and 4	
5 and 6	3.3V
7 and 8	

RJ1 (Gigabit Ethernet Port)

Titanium Ti375 N1156 Development Board provides a Gigabit Ethernet transceiver from Realtek (RTL8211FI), which is compatible with 10 Base-T, 100 Base-TX, and 1,000 Base-T IEEE 802.3 standards. The chip supports:

- RGMII MAC interface
- Supports 120 meters of 1,000 Base-T CAT.5 cable
- Automatic polarity correction
- Low drop voltage regulator

The RJ1 header is connected to Ethernet PHY (U29). The PHY address is 0x1 and the default configuration adds 2 ns delay to `TXC` and `RXC`.

Table 41: U29 ETH PHY Pin Assignment

Pin Name	Signal Name	Ti375 Pin Name	Description
TXD0	RGMII0_TXD0	4C_GPIOB_P _17_CLK4_P	Transmits data. Data is transferred from MAC to PHY via TXD[3:0].
TXD1	RGMII0_TXD1	4C_GPIOB_N _17_CLK4_N	
TXD2	RGMII0_TXD2	4C_GPIOB_P _18_CLK5_P	
TXD3	RGMII0_TXD3	4C_GPIOB_N _18_CLK5_N	
TXCTL	RGMII0_TXEN	4C_GPIOB_N_19 _TEST_N_CLK6_N	Transmits control signal from the MAC.
TXCLK	RGMII0_CLK	4C_GPIOB_P_19 _NSTATUS_CLK6_P	Transmit reference clock can be 125 MHz, 25 MHz, or 2.5 MHz at different rates.
RXD0/ RXDLY	RGMII0_RXD0	4C_GPIOB_P_21	Receives data. Data is transferred from PHY to MAC via RXD[3:0].
RXD1/ TXDLY	RGMII0_RXD1	4C_GPIOB_N_21	
RXD2/ PLLOFF	RGMII0_RXD2	4C_GPIOB_P_22	
RXD3/ PHYAD0	RGMII0_RXD3	4C_GPIOB_N_22	
RXC/ PHYAD1	RGMII0_RXCK	4C_GPIOB_P _20_CLK7_P	For continuous receive, the reference clock can be 125MHz, 25MHz, or 2.5MHz. The clock comes from the receive data stream.
RXCTL/ PHY_AD2	RGMII0_RXDV	4C_GPIOB_N _20_CLK7_N	Receives control signals transferred to MAC.
MDC	ETH_MDC	4C_GPIOB_N_28	Manages the data clock.
MDIO	ETH_MDIO	4C_GPIOB_P_29 _CDI11_EXTFB	Manages data input/output. Pull-up 3.3, 2.5, 1.8, 1.5 V respectively.
PHYRSTB	ETH_RSTN	BR0_GPIOR _142_CLK8	Hardware reset. Active low. A complete PHY reset requires this pin to be pulled low for at least 10ms. All registers are cleared after a hardware reset.

Pin Name	Signal Name	Ti375 Pin Name	Description
INTB/PMEB	ETH_INT	BR0_GPIOR _143_CLK9	This pin supports two functions. If one of the functions is not used, this pin should be left floating. • Interrupt (supports 3.3V pull-up). Set low if status changed; active low. • Power management event (supports 3.3V pull-up). Set low if received a magic packet or wake up frame; active low. **Note:** • The behavior of this pin is level-triggered. • The function of this pin (INTB/PMEB) can be specified by page 0xd40, Reg.22, Bit [5]: 1: Pin 31 is used as PMEB. 0: Pin 31 is used as INTB (default)
LED0/ CFG_EXT	LED0_10M/ CFG_EXT	N.C.	High = Link up at 10 Mbps Blinking = Transmitting or receiving.
LED1/ CFG_LDO0	LED1_100M/ CFG_LDO0	N.C.	Low = Link up at 100 Mbps Blinking = Transmitting or receiving.
LED2/ CFG_LDO1	LED2_1000M/ CFG_LDO1	N.C.	High = Link up at 1000 Mbps Blinking = Transmitting or receiving.

Header USB1 (USB FTDI FT4232H)

USB1, a type-C USB receptacle, is the interface between the board and your computer for communication through the FTDI FT4232H chip. Connect the type-C USB cable for configuring the Ti375 FPGA and NOR flash. Refer to Overview on page 7 for more details on the FTDI FT4232H module interface.

Headers TP1, TP2, TP3, and TP4 (Ground Test Point)

TP1, TP2, TP3, and TP4 are ground test points. Refer to the **Titanium Ti375 N1156 Development Board Schematics and BOM** for more information.

User Outputs

The board has 6 user LEDs that are connected to I/O pins in Ti375 bank 4B. By default, the Ti375 I/O connected to these LEDs are active high. To turn a given LED on, pull the corresponding I/O signal high.

Note: When adding these GPIO in the Efinity® Interface Designer, configure them as output pins.

Table 42: User Outputs

Reference Designator	Ti375 Pin Name	Active
LED1	4B_GPIOB_N_41	High
LED2	4B_GPIOB_P_42	High
LED3	4B_GPIOB_N_42	High
LED4	4B_GPIOB_P_33	High
LED5	4B_GPIOB_P_34	High
LED6	4B_GPIOB_P_35	High

User Inputs

The board has 2 pushbutton switches that you can use as inputs to the Ti375 FPGA. The pushbuttons are connected to I/O pins in Ti375 bank 4B. When building designs using these switches, turn on an internal pull up for these pins in the Interface Designer.

When you press the pushbutton switches the signal drives low, indicating user input.

Table 43: User Pushbuttons

Reference Designator	Ti375 Pin Name	Active
SW3	4B_GPIOB_P_31	Low
SW4	4B_GPIOB_P_32	Low

Installing Standoffs

Before using the board, attach the standoffs with the screws provided in the kit. The following table lists the standoffs and screws, required for standoffs installation.

Note:

- Always power off the development board before attaching the standoffs.
- You do not need to install standoffs if you are inserting the development board to the PCIe slot on your workstation's motherboard.

Table 44: Standoffs and Screws for Standoff Installation

Standoff			Screw		
Size	**Length**	**Qty**	**Size**	**Length**	**Qty**
M3	10 mm	4	M3	6 mm	4

Warning: You can damage the board if you over tighten the screws. Tighten all screws to a torque between 4 ± 0.5 kgf/cm and 5 ± 0.5 kgf/cm.

Heat Dissipation

You can use the Power Estimator to estimate the power consumption of your design and estimate the junction temperature based on the thermal coefficient of the package. If the junction temperature is likely to be greater than the junction temperature specified in the specification, you should consider cooling the board.

Note: You need a license to obtain the **Power Estimator** Excel file from Efinix® Support Center.

Installing Thermal Pad and Cooling Fan

Follow these steps to install the thermal pad and cooling fan:

1. Remove the adhesive film to stick the thermal pad on the development board's Ti375 FPGA.
2. Align the mounting holes on the development board with the cooling fan's PCB board spacer screws. Push each spacer screw into the mounting hole on the board until it fits.
3. Plug in the fan's power supply at the J2 header to start using the cooling fan.

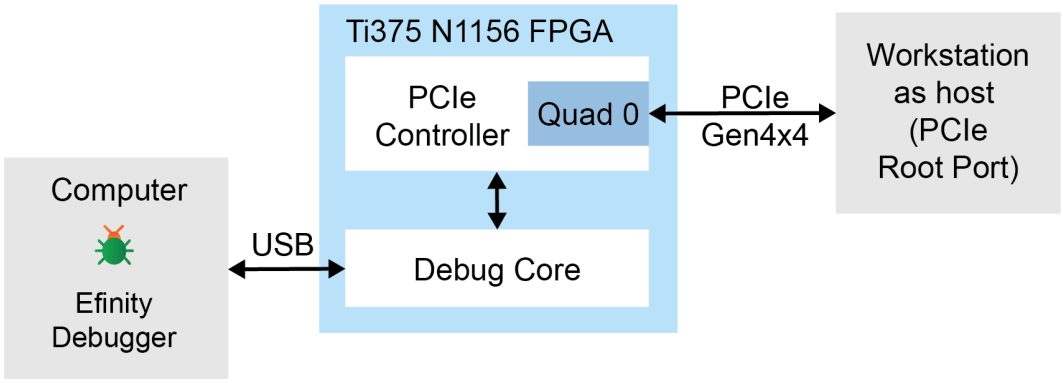
Titanium Ti375 N1156 Development Board Example Design

Efinix preloads the Titanium Ti375 N1156 Development Board with an example design that demonstrates the following functions:

- Sets up a PCIe link between a host system with a general-purpose CPU as the PCIe root port, and the Titanium Ti375 N1156 Development Kit as the PCIe end point.
- Simple read operation to send data from the end point to the host.
- Simple write operation to send data from the end point to the host.

The overall example design includes the PCIe endpoint design targeting the Ti375 N1156 FPGA and the software driver for the host system to perform simple read-and-write operations. You can perform the read-and-write operation through the Linux kernel, C code, or a Python script.

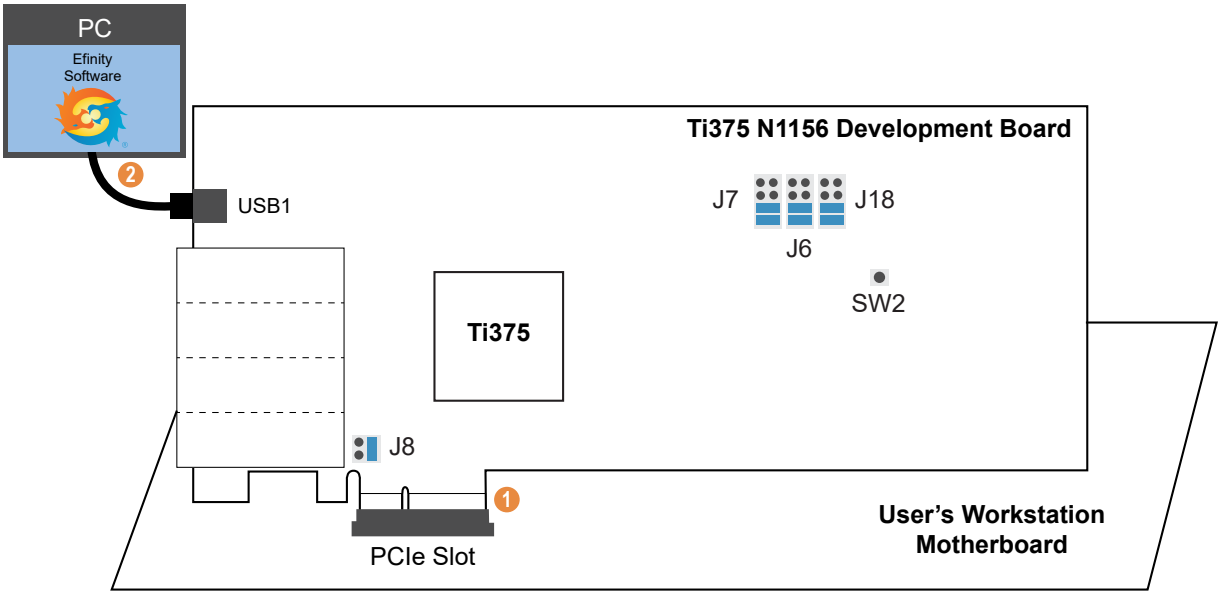
Figure 3: Example Design Block Diagram Overview



Set Up the Hardware

The hardware setup requires a workstation that runs on Ubuntu operating system and has an available PCIe slot. The following figure shows the hardware setup steps:

Figure 4: Hardware Setup



Important: Always switch off the power supply and board's power switch before attaching or detaching cables.

1. Ensure that the jumpers of the development board are set as follow:

Board	Header	Pins to Connect
Titanium Ti375 N1156 Development Board	J6, J7, J18	5 - 6 7 - 8
	J8	3 - 4

2. Insert the Titanium Ti375 N1156 Development Board into the PCIe slot on the workstation motherboard.
3. Connect a USB Type-C cable to the board and to a USB port on your computer which has the Efinity software installed.
4. Turn on the workstation.

   Note: The Titanium Ti375 N1156 Development Board is powered up through the PCIe slot, without the need to connect to a 12 V DC power supply.

5. Press the SW2 button (CRESET_N).

The development board LEDs are described in the following table.

Table 45: Development Board LED Outputs

LED	Description
LED7 turned on	Power good
LED8 turned on	FPGA configuration done

Running the Example Design

The development board powers up when the workstation is turned on. The design is automatically loaded from the on-board flash device.

Enumeration Status

PCIe enumeration is the process of detecting and identifying devices connected to the PCIe bus. The system scans for devices, accesses their configuration spaces, and assigns necessary resources like memory and I/O addresses. To check whether your board has completed the enumeration process:

1. Open a terminal on the Ubuntu workstation.
2. (Optional) Use the `update-pciids` command to fetch the latest PCI ID list and install it. This step allows your workstation to translate the device ID. (Refer to **update-pciids man page** for more information.)
3. Use the `lspci` command to get a list of the PCI devices connected to the PCI subsystem.
4. Look for `Efinix, Inc. Default ID for Titanium FPGA PCIe Interface (AXI)` or `Device 1f7a:0100` in the listing.

```
04:00.0 Ethernet Controller: Realtek Semiconductor Co., Ltd.
 RTL8111/8168/8211/8411
PCI Express Gigabit Ethernet Controller (rev 16)
02:00.0 Non-VGA unclassified device: Efinix, Inc. Default ID for Titanium
 FPGA PCIe Interface (AXI)
...
```

5. If the board's device ID is in the list, the enumeration is successful. If the device ID is not in the list, try these solutions:
   - Restart the Linux workstation with the reboot command.
   - If the problem persists, it might indicate that the slot to which you connected the board does not support PCIe. Try switching to another PCI slot on your workstation.
   - If none of the PCI slots work, it might indicate that your workstation does not have PCI capabilities.
6. Use the `sudo lspci -vv -d 1f7a:0100` command to output the translated description of the PCI Type 0 Configuration Space. The type 0 configuration space displays the configuration setting of the PCIe endpoint device, for example, the link speed and link width.

Figure 5: Using the `sudo lspci -vv -d 1f7a:0100` Command

```
$sudo lspci -vv -d 1f7a:0100
02:00.0 Non-VGA unclassified device: Efinix, Inc. Default ID for Titanium FPGA PCIe Interface
 (AXI)
 Subsystem: Efinix, Inc. Device 0000
 Control: I/O- Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B-
 DisINTx-
 Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR-
 INTx-
 Interrupt: pin A routed to IRQ 5
 Region 0: Memory at e0000000 (64-bit, prefetchable) [disabled] [size=512K]
 Expansion ROM at fcf00000 [disabled] [size=4K]
      Capabilities: [80] Power Management version 3
          Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+, D1+, D2-, D3hot+, D3cold-)
          Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
      Capabilities: [90] MSI: Enable- Count=1/1 Maskable+ 64bit+
          Address: 0000000000000000    Data: 0000
          Masking: 00000000    Pending: 00000000
      Capabilities: [b0] MSI-X: Enable- Count=1 Masked-
          Vector table: BAR=0 offset=00000000
          PBA: BAR=0 offset=00000008
```

Figure 6: Link Speed and Width

```
LinkCap: Port #0, Speed 16GT/s, Width x4, ASPM L0s L1, Exit Latency L0s <256ns, L1 <8µs
         ClockPM- Surprise- LLActRep- BwNot- ASPM0ptComp+
LinkCap: ASPM Disabled; RCB 64 bytes Disabled- CommClk-
         ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
LinkCap: Speed 16GT/s (ok); Width x4 (ok)
         TrErr- Train- SlotClk- DLActive- BWMgmt- ABWMgmt-
```

Note: If you witness a downgraded PCIe link after a system power cycling and configuration from the SPI flash memory, for example, not achieving the PCIe Gen 4 data rate or data bus width is less than x4, you need to reset the FPGA by pressing the SW2 reset button and performing a warm reboot on your computer.

Debug Profile

The endpoint design comes with a debug profile to monitor the PCIe traffic and status. The `vio0` debug core contains the PCIe status and AXI virtual I/O interface; the `la0` debug core has the logic analyzer interface to monitor the AXI write channel. Run the following steps to connect the Efinity Debugger.

1. Launch the Efinity software and open the Debugger.
2. Choose **File > Open Debug Profile**.
3. Browse to the **hardware** directory and choose **debug_profile.json**. Click **Open**.
4. Select **Perspectives > Debug**.
5. Select **JTAG > USER1** and connect the Debugger.

Table 46: Debugger vio0 Tab on page 45 describes the probe and source signals in `vio0`.

Table 46: Debugger vio0 Tab

Name	Type	Width	Description
q0_ltssm_state	Probe	6	LTSSM State. Refer to "Appendix C: LTSSM State Encoding" in the Titanium PCIe® Controller User Guide for more information.
q0_link_status	Probe	2	Status of the PCIe link. 2'b00: No receivers detected. 2'b01: Link training in progress. 2'b10: Link up, DL initialization in progress. 2'b11: Link up, DL initialization completed.
q0_p00_rate	Probe	2	PIPE link signal rate for transceiver quad 0. Selects the data rate. 2'b00: PCIe Gen1 2'b01: PCIe Gen2 2'b10: PCIe Gen3 2'b11: PCIe Gen4
q0_cmn_ready	Probe	1	Common ready.
q0_TARGET_AXI_AWSIZE	Probe	3	AXI write transfer size.
q0_TARGET_AXI_AWADDR	Probe	64	AXI write transfer address.
q0_TARGET_AXI_WDATA	Probe	256	AXI write data.
q0_TARGET_AXI_ARSIZE	Probe	3	AXI read transfer size.
q0_TARGET_AXI_ARADDR	Probe	64	AXI read transfer address.
q0_TARGET_AXI_RDATA	Source	256	AXI read data.

Learn more: Refer to the Efinity Software User Guide for complete instructions on using the Debugger.

Performing the Read/Write Test

The example design includes files to perform the read/write test using the Linux kernel module, a Python script, or C code.

Using the Linux Kernel Module

This simple script run in the Linux operating system's kernel layer. It enables the memory region and performs a single read write operation.

This method requires kernel header files. Install them with this command:

```
$ sudo apt-get install linux-headers-$(uname -r)
```

To perform the read/write test, use these steps in a terminal:

1. Change to the example design's **linux-kernel-mod** directory.
2. Use the `make all` command to compile the source code.
3. In the Efinity Debugger's vio0 tab, set `q0_TARGET_AXI_RDATA` to `0xDEADBEEF` or to any 32 bit value. The read operation uses this value when installing the kernel driver.
4. Use the command `sudo insmod efx_pcie_simple_rdwr.ko` to install the Linux kernel module.
5. Use the command `sudo dmesg | grep efx_pcie` to view the output of the kernel driver program.

```
$ sudo insmod efx_pcie_simple_rdwr.ko … $ sudo dmesg | grep efx_pcie
[ 174.141878] efx_pcie - VENDOR_ID: 0x1f7a
[ 174.141885] efx_pcie - DEVICE_ID: 0x100
[ 174.141926] efx_pcie - read_mem_and_print: deadbeef
[ 174.141929] efx_pcie - write_mem_and_print: 49333735
```

6. In the Debugger, observe that the value of `q0_TARGET_AXI_WDATA` changed from `0x0` to `0x49333735`.

Using the Python Script

This Python script uses the `mmap()` system call to map the PCIe endpoint application to a virtual address in the operting system. You can then perform the read/write operation on the virtual address to send/receive packets from Ti375 PCIe Controller.

To perform the read/write test, use these steps in a terminal:

1. Change to the example design's **python-scripts** directory.
2. Use this command to install the dependencies:

```
$ sudo pip install -r requirements.txt
```

3. In the Efinity Debugger's vio0 tab, set `q0_TARGET_AXI_RDATA` to `0xDEADBEEF` or to any 32 bit value. The Python script's read operation uses this value.
4. Use the command `sudo python3 rd_req_pci.py` perform a read operation on the endpoint and display the result in little endian format.

```
$ sudo python3 rd_req_pci.py
Target Device's Bar: [BaseAddressRegister(type='mem', addr=4156567552,
 size=4096), BaseAddressRegister(type='mem', addr=4156555264, size=8192)]
Target Device's Bar 0 addr: 0xf7c03000
Target Device's Bar 0 size: 0x1000
Reading from BAR 0:
b'\xef\xbe\xad\xde'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
b'\x00\x00\x00\x00'
```

5. On the Debugger, click the **la0** tab.
6. Add a trigger condition to capture the waveform when `q0_TARGET_AXI_WVALID = 1`.
7. Use the command `sudo python3 wr_req_pci.py` to perform a write operation and a read operation. The read operation outputs the result in little endian format.

```
$ sudo python3 wr_req_pci.py
Target Device's Bar: [BaseAddressRegister(type='mem', addr=4156567552,
 size=4096), BaseAddressRegister(type='mem', addr=4156555264, size=8192)]
Target Device's Bar 0 addr: 0xf7c03000
Target Device's Bar 0 size: 0x1000
Writing to BAR 0: 48492c5449333735
8
b'\xef\xbe\xad\xde\x00\x00\x00\x00'
```

The Debugger generates a waveform that displays the output of the AXI write operation. The write operation could be completed in a single AXI write operation or multiple AXI write operations, depending on the workstation's model.

Using the C Code

This simple C program performs a single read write operation using the `sysfs` filesystem.

To perform the read/write test, use these steps in a terminal:

1. Change to the example design's **c-code-sysfs** directory.
2. Use the `make main` command to compile the source code.
3. Use the `lspci -D` command to find the assigned Bus:Device.Function (BDF) for the board. In the following example, the BDF is `0000:02:00.0`.

```
$ lspci -D
0000:00:00.0 Host bridge: Intel Corporation 4th Gen Core Processor Dram
 controller
...
0000:02:00.0 Non-VGA unclassified device: Efinix, Inc. Default ID for
 Titanium FPGA PCIe Interface (AXI)
...
```

4. Locate the PCI files in `sysfs`. The typical location is **/sys/bus/pci/devices/<*BDF*>**. For example: **/sys/bus/pci/devices/0000:02:00.0**.
5. Use the command `cat /sys/bus/pci/devices/<*BDF*>/vendor` to confirm the board's vendor id; it should be `0x1f7a`. If the vendor id matches the expected output, note the full path name.

```
$ cat /sys/bus/pci/devices/0000:02:00.0/vendor
0x1f7a
```

6. In the Efinity Debugger's vio0 tab, set `q0_TARGET_AXI_RDATA` to `0xDEADBEEF` or to any 32 bit value. The read operation uses this value.
7. Run the C program using the command:

```
pcimem [sys file] [offset] [type [data]]
```

Where:

[sys file] is the sysfs file for the PCI resource to act on.

[offset] is the offset into the PCI memory region to act upon.

[type] is the access operation type, [b]yte, [h]alfword, [w]ord, [d]ouble-word.

[data] is the data to be written.

```
sudo ./pcimem /sys/bus/pci/devices/<BDF>/resources0 0 w 0x1234abcd
```

The following example uses the `0000:02:00.0` BDF.

```
$ sudo ./pcimem /sys/bus/pci/devices/0000:02:00.0/resource0 0 w 0x1234abcd
/sys/bus/pci/devices/0000:02:00.0/resource0 opened.
Target offset is 0x0, page size is 4096
mmap(0, 4096, 0x3, 0x1, 3, 0x0)
PCI Memory mapped to address 0x7f4bdf5a2000.
0x0000: 0xDEADBEEF
Written 0x1234ABCD; readback 0xDEADBEEF
```

In the Debugger, observe that the value of `q0_TARGET_AXI_WDATA` changed from `0x0` to `0x49333735`.

Creating Your Own Design

The Titanium Ti375 N1156 Development Board allows you to create and explore designs for the Ti375 FPGA. Efinix® provides example code and designs to help you get started:

- Our Support Center (www.efinixinc.com/support) includes examples targeting the board.
- The Efinity® software includes also example designs that you can use as a starting point for your own project and includes a step-by-step tutorial.

Restoring the Demonstration Design

After you have used the board for other designs, you may want to go back to the original pre-loaded example design. The preloaded example design project file is available in the Titanium Ti375 N1156 Development Board Demonstration Design page. To restore the example design, you need to program the board's SPI flash device with the Ti375 FPGA's example design bitstream.

Note: The example design available in the Support Center requires Efinity software v2024.1 or later.

The example design zip file includes a bitstream file to get you started quickly. Download it to the board using these steps:

1. Download the file **efx_ti375n1156_oob.zip** from the Support Center.
2. Open the project **ti375n1156_oob.xml**) in the Efinity software. The project is located in the **efx_ti375n1156_oob** directory.
3. Connect the Titanium Ti375 N1156 Development Board to your computer using a USB cable.
4. Go to Efinity Programmer to download the bitstream file, **bitstream/ti375n1156_oob.hex**, to your board using the **SPI Active using JTAG Bridge (new)** mode. Set the **Starting Flash Address** to `0x000000`.

Revision History

Table 47: Revision History

Date	Version	Description
October 2024	1.0	Initial release. (DOC-2021)