

# 钛金系列 TJ375 N1156X 开发套件 指南

TJ375N1156X-DK-UG-v1.3 2025-06 www.elitestek.com



# 目录

简介	
包装清单	
下载 Primus <sup>®</sup> 软件	
安装 Linux USB 驱动程序	4
安装 Windows USB 驱动	
开发板功能说明	5
特性	5
兼容子卡	6
概述	7
上电	
复位	
	9
eMMC	10
Serdes 接口	11
いた PHV	
但在在成哭	12
<u>一</u> 及7%品	
₽X%\$?^W93 按□	ے ہے۔ 12
I文山	ے ا 12
F1、F2、F3、F4 按口(夕功化)	
P5 按口(PCI EXPIESS <u> </u>	LO 17
J 按山(巴源)	/ ۱
」2 按山(八)羽)	
J3 排针(FML 按山电压选择)	
J4(MICro-SD下僧)	
J5 (VQPS 电源使能选择)	
J6 排针(QSE GPIO 电压选择)	
J7 排针 (MIPI GPIO 电压选择)	
J8 排针 (PCle 设备检测来源选择)	
J9、J10、J11、J12 SFP+(10G) 接口	19
J13 (SFP+ 光模块 cage)	
J14 和 J15 接口 (FMC)	22
J16 排针(JTAG)	
J18 排针 (Bank BL0 GPIO 电压选择)	
RJ1 (千兆以太网口)	
USB1接口 (USB FTDI FT4232H)	
TP1、TP2、TP3、TP4 测试点(接地)	
用户输出	
用户输入	
钛金系列TJ375 N1156X 开发板示例设计	
设置硬件	
运行示例设计	
枚举状态	
调试配置文件	
简单读写测试	
目定义设计	
佐有子例次注	70
<b>灰冬小四冬</b> 月	<b>זכ</b> דר
小//)以り入行	/ 5 דר
炕求义计到开仅似	
修订记录	39
	······································

感谢您选择 钛金系列 TJ375 N1156X 开发套件 (套件号: TJ375N1156X-DK)。有了此套件,您可充分探索 TJ375 FPGA 的功能。

钛金系列 TJ375 N1156X 开发套件为您提供 TJ375 FPGA 开发和原型设计所需的一切 资源。TJ375 FPGA 的全双工收发器支持多种协议,包括 PCIe<sup>®</sup> Gen4(数据速率高达 16 Gbps)、Ethernet 10GBase-KR 和 Ethernet SGMII。PCIe 和以太网收发器具有硬核 PCS,协 议实现和使用更容易。TJ375 还支持数据速率高达 12.5 Gbps 的 PMA Direct 模式,您可以将其 用于自定义协议。此外,我们也提供 3 个 TJ375 硬核 MIPI TX 和 3 个 TJ375 硬核 MIPI RX (每 条 Iane 速率最高分别可达 2.5 Gbps)、2 个 TJ375 软核 MIPI TX/RX,可配置成 LVDS 或其他 标准(每条 Iane 速率最高分别可达 1.5 Gbps)及 2 个硬核 LPDDR4 和 LPDDR4X 控制器。

钛金系列 TJ375 N1156X 开发板 板载 4 个 SFP+ (10G) 接口和 1 个 PCle 4.0 x 4 接口。 开发板上的 2 个高性能 512 Mbit SPI NOR Flash 芯片支持 SPI Active x1/x2/x4/x8 模式配 置模式,也可存储其他用户数据,例如 RISC-V 固件; 1 个 8 GB 容量的 eMMC 可用于储存 用户数据,您也可选择使用板上的 SD 卡连接器读写自己的 SD 卡。开发板板载一个高度集成 Ethernet RJ45 接口,支持 RGMII,速率最高 1000 Mbps。钛金系列 TJ375 N1156X 开发板 还 带有 USB 转双路 JTAG 芯片,支持 FPGA 加载和 RISC-V SOC JTAG 调试。开发板还板载 2 个 FMC (LPC) 接口,每个 FMC 接口带有 4 个高速收发器 lane,您可购买需要的 FMC 模块扩展 功能。开发板上的 4 个多功能 QSE 连接器支持 QSE 子卡。



警告:开发板在运输和使用过程中,请注意防静电。若未进行防静电处理,开发板可能会损坏。

# 包装清单

钛金系列 TJ375 N1156X 开发套件 包括:

- 钛金系列TJ375 N1156X 开发板 (已安装风扇和面板)
- USB2.0 转 Type-C 电缆
- 12V 6.25A 适配器套装
- 跳线帽

# 下载 Primus<sup>®</sup> 软件

要开发板载 TJ375 FPGA, 您需要安装 Primus<sup>®</sup> 软件。您可从 易灵思 **产品中心** 的 **开发环境** Primus 页面获取该软件, 网址为: https://www.elitestek.com/support/。

Primus<sup>®</sup> 软件包含板载器件编程工具。 您可参阅 《Primus<sup>®</sup> 软件用户指南》, 了解如何编 程器件。

**了解更多:** Primus<sup>®</sup> 文档集成在软件中,和软件一起安装(见 Help)。您也可以在官网的 技术支持 处找到此文档。

# 安装 Linux USB 驱动程序

以下是如何在 Linux 操作系统上安装 USB 驱动程序的步骤:

- 1. 将开发板与计算机断开连接。
- 2. 在终端中使用以下命令:

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

注意: 如果在执行这些命令之前,开发板已经连接到计算机,则需要先断开连接。驱动安装完成后,再重新连接。

# 安装 Windows USB 驱动

钛金系列TJ375 N1156X 开发板 上板载了一个FTDI FT4232H芯片,用于USB端口通信。

注意:如果您有两块 易灵思 开发板目 钛金系列TJ375 N1156X 开发板正在使用,您必须根据情况管理驱动程序。参阅 AN 050: Managing Windows Drivers 了解更多信息。

在 Windows 上,您可以使用 Zadig 软件安装驱动程序。从 zadig.akeo.ie下载 Zadig 软件(版本 2.7 或更高版本)。(该软件无需安装,下载后直接运行可执行文件即可。)

安装驱动程序:

- 1. 使用 USB 电缆将开发板连接到电脑并给开发板上电。
- **2.** 运行 Zadig 软件。
- 3. 选择 Options > List All Devices。
- 4. 选择 TJ375N1156X Development Kit (Interface 0)。



注意: 连接 钛金系列TJ375 N1156X 开发板 (使用 UART 接口)到电脑时, Interface 2 需要使用 FTDI 的虚拟串口驱动程序。

- 5. 在 Driver 下拉列表中,选择 libusb-win32。
- 6. 点击 Replace Driver。
- 7. 选择 TJ375N1156X Development Kit (Interface 1)。
- 8. 重复步骤 5 和 6。
- 9. 关闭 Zadig 软件。

# 开发板功能说明



钛金系列TJ375 N1156X 开发板 包含多种组件,可帮助您设计 钛金系列 TJ375 FPGA。

# 特性

- FBGA-1,156 封装的 易灵思 TJ375N1156XC4<sup>(1)</sup> FPGA
- 2 x 8 Gbit (32 Mbit x 16 DQ x 8 bank x 2 channel) LPDDR4/LPDDR4X SDRAM:
  - 支持x32数据宽度
  - 读/写速度高达 3.3 Gbps
  - 2 x 512 Mbit 兆易创新 GD25LB512MEYIGR SPI NOR Flash
    - 每个 SPI Flash都支持单线、双线和四线模式
      - 在四线模式下两个SPI Flash可支持 x8 位宽
- 8 GB eMMC
- 4 个 QSE 接口
  - 2个1.5 Gbps TX/RX 软核 MIPI 接口
  - 3个 2.5 Gbps RX 硬核 MIPI 接口
  - 3 个 2.5 Gbps TX 硬核 MIPI 接口
- 1 个 GE 接口,支持RGMII标准,符合 10 Base-T、100 Base-TX 和 1,000 Base-T IEEE 802.3 标准
- 4 个 SFP+ 接口
- PCle 4.0 x 4
- Micro-SD 卡槽
- 2 个 LPC FMC 接口 (每个 FMC 接口带有 4 个高速收发器 lane)
- 配置开发板的 USB Type-C 接口及 USB 转 JTAG 芯片
- TJ375 PLL 输入支持 25 MHz、33.3333 MHz、74.25 MHz、100 MHz 和 156.25 MHz 振荡器时钟输入
- 用户 LED 和开关:
  - 6个LED灯
  - \_ 2个按钮开关

<sup>&</sup>lt;sup>(1)</sup> FPGA 速度等级可能有所不同,视具体情况而定。

- 电源:
  - \_ 12V电源接口
  - 开发板电压种类: 0.85V、0.95V、1.1V、1.2V、1.8V、3.3V、5.0V
- Power good LED灯和 TJ375 configuration done LED灯
- 复位按钮:
  - SW2:开发板复位

# 兼容子卡

钛金系列 TJ375 N1156X 开发板 可兼容的子卡如下:

- MIPI 和 LVDS 扩展子卡
- 双路 MIPI 转 DSI 转换器子卡
- 树莓派摄像头连接子卡
- 双树莓派摄像头连接子卡
- Coral 摄像头连接子卡
- MIPI 转 HDMI2.0 子卡

# 概述

板载的 易灵思TJ375 FPGA采用 FBGA-1,156封装,fabric 部分采用易灵思 Quantum<sup>®</sup> 技术。可编程逻辑和路由结构经 Quantum<sup>®</sup> 加速后,与 I/O 接口一起组成小尺寸封装。TJ375 FPGA 还包括嵌入式存储模块和 DSP 模块。您可以使用Primus<sup>®</sup> 软件为 TJ375 FPGA创建设计,然后使用 USB 数据线将生成的配置位流文件下载到开发板。



#### 图 3: 钛金系列TJ375 N1156X 开发板 元件布局



钛金系列TJ375 N1156X 开发板 提供四个多功能 0.8mm pitch 的 QSE 连接器。这些连接器 可用于 MIPI CSI-2/DSI TX/RX 和 LVDS TX/RX 信号连接测试。

该开发板包括一个连接 FTDI 芯片的 USB Type-C 接口。FTDI模块从USB 主机处接收 TJ375 的配置位流,并在JTAG模式下写入 TJ375 FPGA。当使用JTAG模式时,您也可以使用 JTAG SPI Flash Loader Core 将配置位流写入板载的SPI NOR Flash中。此外,FTDI模块还可 为 TJ375提供一个 UART 接口。

SPI NOR Flash会存储配置位流。在主动配置模式(默认)下, TJ375 FPGA 由 SPI NOR Flash 配置。

了解更多: 有关所用组件的更多信息, 请参阅 钛金系列TJ375 N1156X 开发板 的原理图和 BOM。

# 上电

将套件中随附的 12 V 电源适配器插上电源,使用 P1J-P1M 接头连接开发板的 12 V 电源输入接头 J1,再打开 SW1 开关,开发板会正常输出 0.85 V、0.95 V、1.1 V、1.2 V、1.8 V、3.3 V、5.0 V 电压。当 3.3 V 电压上升并稳定时,LED7 power-good LED 灯会变亮,您可直接确认上电状态。

# 复位

CRESET\_N是TJ375 FPGA的复位信号。上电后,在1.8V 电压稳定之前,CRESET\_N通常会一 直保持低电平,TJ375 FPGA 会一直保持在复位状态。直到1.8V电压稳定后,CRESET\_N 信号从 低电平跳变到高电平,TJ375 FPGA才会进入正常工作模式。

注意: 您也可以通过 SW2 按钮来复位TJ375 FPGA。

CRESET\_N 有一个下拉电阻,可保证 TJ375 FPGA 在上电过程中处于复位状态。按下 SW2 时, CRESET\_N 被拉低;松开 SW2 时, CRESET\_N 被置高。因此,只需按一次 SW2 按钮即可实 现复位所需的高-低-高电平转换。

CRESET\_N 电平切换后, TJ375 FPGA 会进入配置模式并从 Flash 中读取配置位流。 配置成功完成后, FPGA 会拉高 CDONE 信号。LED1 由 CDONE 信号控制。CDONE 信号被置高时, LED1 灯会点亮。

#### FTDI复位

FT\_RST 信号是FTDI FT4232H 芯片复位信号。 当您拉低 FT\_RST 信号时,通过 FTDI FT4232H 芯片建立的所有通信都会断开,而当您置高 FT RST 信号时,所有通信又会重新连接。

# 配置

您可以使用以下配置模式配置 TJ375 FPGA:

JTAG

• SPI Active (up to x8) via JTAG Bridge

烧录闪存时必须使用 JTAG 桥,因为闪存的 SPI 信号未直接连接到 FT4232H 芯片。为您 自己的设计生成位流文件时,请确保在 Primus **Settings** 的 **EFX Pgm Options** 选项卡中选择 **Active** 选项。有关 SPI Active via JTAG Bridge 模式的说明,请参阅 烧录文件到开发板了解烧 录操作。

A

注意: 有关 JTAG SPI bridge loader 详情,请参阅 Primus 软件用户指南。

# 时钟源

六个板载有源晶振(25 MHz、33.3333 MHz、74.25 MHz、100 MHz 和 156.25 MHz) 可用于驱动 TJ375 PLL 输入引脚和时钟输入。其中,100 MHz 和 156.25 MHz 是差分晶振,分 别为 PCIe 和 SFP+ 提供参考时钟。此外,还有两个 25 MHz 的板载无源晶体,一个连接到 GE PHY,为 GE PHY 提供参考时钟,另一个连接到时钟产生器,用于为 FMC 的 SERDES 提供参考 时钟。一个 12 MHz 的板载无源晶体专用于 FT4232H 芯片。

#### 表 1: 振荡器和时钟生成器源

时钟源	TJ375 引脚名称	PLL 资源
	TR1_GPIOR_97/MIPI2_PLL_REFCLK	MIPI2 PLL
25 M山- 疟荬鸮	TL1_GPIOL_46/MIPI1_PLL_REFCLK	MIPI1 PLL
23 11172 派汤帝	TR1_GPIOR_88_PLLIN1	PLL_TR1
	BL2_GPIOL_23/MIPI0_PLL_REFCLK	MIPIO PLL
33.3333 MHz 振荡器	BR1_GPIOR_145_PLLIN1	PLL_BR1
74.25 MHz 振荡器	TR0_GPIOR_93_PLLIN1	PLL_TR0
	TL0_GPIOL_52_PLLIN1	PLL_TL0
100 MU-	TR2_GPIOR_101_PLLIN1	PLL_TR2
	BL2_GPIOL_25_PLLIN1	PLL_BL0
	Q0_REFCLK1_P/N	N/A
156.25 MHz 振荡器	Q1_REFCLK0_P/N	/



注意: 如果 TJ375 FPGA左侧使用的GCLK 资源超过 8 个 ,则 Interface Designer 会发送一条 clkmux 路由错误。 要解决此问题,请分配其中一个时钟(例如 MIPI 时钟 TX0)来使用 RCLK ,而不是 GCLK。 有关详细信息,请参阅 TJ375 数据手册中的"时钟和控制网络"章节。

# eMMC

钛金系列TJ375 N1156X 开发板 板载一个 8 GB eMMC 闪存,可用于存储用户数据。支持 HS400 模式,数据速率为 400 MBps,时钟频率为 200 MHz。信号连接到 Bank 4A 的 HSIO。

E.	2.	01/	NAC	2	「��」(二〇二〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇
不	∠.	GIAI	IVIC	- 7	ルギフロ

引脚名称	信号名称	TJ375 引脚名称
DATA0	EMMC_DATA0	4A_GPIOB_P_50
DATA1	EMMC_DATA1	4A_GPIOB_N_48
DATA2	EMMC_DATA2	4A_GPIOB_P_51
DATA3	EMMC_DATA3	4A_GPIOB_N_50
DATA4	EMMC_DATA4	4A_GPIOB_N_47
DATA5	EMMC_DATA5	4A_GPIOB_P_48
DATA6	EMMC_DATA6	4A_GPIOB_N_49
DATA7	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

# Serdes 接口

TJ375 高速收发器接口是一种多协议、全双工收发器,支持数据速率 1.25 Gbps 至 16 Gbps。

钛金系列 TJ375 N1156X 开发板 支持 10GBASE-R SFP+ 接口和 PCle4.0 接口。接口详情和 连接情况请参见下文 第12页的<mark>接口</mark>描述。

注意: 有关 Serdes 接口详情,请参阅 钛金系列TJ375 N1156X 开发板 原理图和 BOM。 有关 Serdes 接口硬件设计指导,可参阅《高速 SerDes 接口硬件设计指南》。

# 以太网 PHY

钛金系列TJ375 N1156X 开发板 提供一个以太网PHY。 以太网PHY的工作特性如下:

- 千兆以太网 (GE) PHY 工作在 RGMII to copper 模式下。
- RTL8211FICG GE PHY 的 RX0 和 RX1 管脚分别和 RXDELAY 和 TXDELAY 复用,可以上拉或者下拉调整 PHY 接收或发送 RXC、TXC 和数据之间的延迟。钛金系列TJ375 N1156X 开发板 默认 RXDELAY 和 TXDELAY 上拉, RXC 和 TXC 各增加 2ns 延迟。
- CFG\_LDO1 管脚默认上拉, CFG\_LDO0 管脚默 认下拉, CFG LDO[1:0]=2'b10, RGMII接口工作电压为1.8V。
  - GE PHY的电源如下:
    - DVDD33、AVDD33: 3.3V
    - DVDD10、AVDD10: 1.0V
    - DVDD RG: RGMII I/O 管脚电源

注意: 有关以太网 PHY详情,请参阅 钛金系列TJ375 N1156X 开发板 原理图和 BOM。



# 温度传感器

钛金系列TJ375 N1156X 开发板 提供一个 DS18B20U 温度传感器。

注意: 有关温度控制电路详情, 请参阅 钛金系列TJ375 N1156X 开发板 原理图和 BOM。

# 散热风扇

钛金系列TJ375 N1156X 开发板 提供散热风扇,降低开发板上的温度。风扇工作电压 12 V,通过 J2 接口连接到开发板。

注意:

Ĭ

#### • 风扇无保护罩,请注意勿将手指伸入转动的风扇内。

• 有关温度控制电路详情,请参阅 钛金系列TJ375 N1156X 开发板 原理图和 BOM。

# 接口

开发板上有多种接口,用于提供电源输入、信号输入输出,以及与外部器件进行通信。

### 表 3: 钛金系列TJ375 N1156X 开发板 接口

参考标号	说明
P1	用于 MIPI TX/RX、LVDS 或 GPIO 的 40 针多用途高速 QSE 接口
P2	用于 MIPI TX/RX、LVDS 或 GPIO 的 40 针多用途高速 QSE 接口
P3	用于 MIPI TX 或 GPIO 的 40 针多用途高速 QSE 接口
P4	用于 MIPI RX 或 GPIO 的 40 针多用途高速 QSE 接口
Р5	PCI Express 金手指
J1	12V 直流电源输入插孔
J2	风扇接口
J3	FMC_VADJ 电压选择
J4	Micro SD 卡插槽
J5	VQPS 电源使能选择
J6	VCCIO_QSE 电压选择
J7	VCCIO_MIPI 电压选择
J8	PCIe 设备检测来源选择
J9、J10、J11、J12	SFP+(10G) 接口
J13	SFP+ 光模块 cage
J14、J15	FMC 接口
J16	3.3 V JTAG 接口
J18	VCCIO_BL0 电压选择
RJ1	RJ-45 千兆以太网接口
USB1	USB Type-C 插座
TP1、TP2、TP3、TP4	接地测试点

# P1、P2、P3、P4 接口 (多功能)

P1、P2、P3、P4 是用于MIPI TX/RX、LVDS 或 GPIO 的多功能高速 QSE 接口连接器,支持 2 个时钟 lane 和 8 个数据 lane。

## 表 4: P1 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
1	VCC_3V3		2	HS_MIPI_TXDP0	4D_GPIOB_P_03_PLLIN0
3	VCC_5V		4	HS_MIPI_TXDN0	4D_GPIOB_N_03_CDI31
5	GND		6	GND	
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		12	GND	
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		18	GND	
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		24	GND	
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		30	GND	
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		36	GND	
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

# 表 5: P2 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
1	VCC_3V3		2	HS_MIPI_RXDP0	4D_GPIOB_P_10
3	VCC_5V		4	HS_MIPI_RXDN0	4D_GPIOB_N_10
5	GND		6	GND	
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		12	GND	
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		18	GND	
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		24	GND	
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		30	GND	
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		36	GND	
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

### 表 6: P3 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
1	VCC_3V3		2	MIPI2_TXDP0	MIPI2_TXDP0
3	VCC_5V		4	MIPI2_TXDN0	MIPI2_TXDN0
5	GND		6	GND	
7	MIPI1_TXDP0	MIPI1_TXDP0	8	MIPI2_TXDP1	MIPI2_TXDP1
9	MIPI1_TXDN0	MIPI1_TXDN0	10	MIPI2_TXDN1	MIPI2_TXDN1
11	GND		12	GND	
13	MIPI1_TXDP1	MIPI1_TXDP1	14	MIPI2_TXDP2	MIPI2_TXDP2
15	MIPI1_TXDN1	MIPI1_TXDN1	16	MIPI2_TXDN2	MIPI2_TXDN2
17	GND		18	GND	
19	MIPI1_TXDP2	MIPI1_TXDP2	20	MIPI2_TXDP3	MIPI2_TXDP3
21	MIPI1_TXDN2	MIPI1_TXDN2	22	MIPI2_TXDN3	MIPI2_TXDN3
23	GND		24	GND	
25	MIPI1_TXDP3	MIPI1_TXDP3	26	MIPI2_TXDP4	MIPI2_TXDP4
27	MIPI1_TXDN3	MIPI1_TXDN3	28	MIPI2_TXDN4	MIPI2_TXDN4
29	GND		30	GND	
31	MIPI1_TXDP4	MIPI1_TXDP4	32	QSE1_GPIO_3	TL0_GPIOL_43
33	MIPI1_TXDN4	MIPI1_TXDN4	34	QSE1_GPIO_4	TL0_GPIOL_44

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
35	GND		36	GND	
37	QSE1_GPIO_1	TL0_GPIOL_41	38	QSE1_GPIO_5	BL3_GPIOL_32_PLLIN1
39	QSE3_GPIO_2	BL3_GPIOL_27_CLK25	40	QSE1_GPIO_6	BL3_GPIOL_34_PLLIN1

### 表 7: P4 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
1	VCC_3V3		2	MIPI2_RXDP0	MIPI2_RXDP0
3	VCC_5V		4	MIPI2_RXDN0	MIPI2_RXDN0
5	GND		6	GND	
7	MIPI1_RXDP0	MIPI1_RXDP0	8	MIPI2_RXDP1	MIPI2_RXDP1
9	MIPI1_RXDN0	MIPI1_RXDN0	10	MIPI2_RXDN1	MIPI2_RXDN1
11	GND		12	GND	
13	MIPI1_RXDP1	MIPI1_RXDP1	14	MIPI2_RXDP2	MIPI2_RXDP2
15	MIPI1_RXDN1	MIPI1_RXDN1	16	MIPI2_RXDN2	MIPI2_RXDN2
17	GND		18	GND	
19	MIPI1_RXDP2	MIPI1_RXDP2	20	MIPI2_RXDP3	MIPI2_RXDP3
21	MIPI1_RXDN2	MIPI1_RXDN2	22	MIPI2_RXDN3	MIPI2_RXDN3
23	GND		24	GND	
25	MIPI1_RXDP3	MIPI1_RXDP3	26	MIPI2_RXDP4	MIPI2_RXDP4
27	MIPI1_RXDN3	MIPI1_RXDN3	28	MIPI2_RXDN4	MIPI2_RXDN4
29	GND		30	GND	
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		36	GND	
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

# P5 接口 (PCI Express 金手指)

P5 是标准 PCIe 金手指连接器,可插入金手指插槽。此连接器接入TJ375 FPGA 的 PCIe 收发器组(Quad 0),最高支持 x4 链路宽度,可满足不同带宽需求。

**注意:** 由于 钛金系列 TJ375 N1156X 开发套件 PCle 金手指连接器存在通道翻转(lane reversal)设计,当使用 PCle x1 或 x2 接口时,您需要在 **Interface Designer** 中将 PCle 控制器实例化为 x4 链路宽度,以允许在 PCle 链路训练期间进行 lane 翻转。

### 表 8: P5 引脚分配

i

信号名称	引脚名称	TJ375 引脚名称
PCIE_PRSNT1n	PRSNT1_N	
VCC_12V	+12V	
VCC_12V	+12V	
GND	GND	
	JTAG_TCK	
	JTAG_TDI	
	JTAG_TDO	
	JTAG_TMS	
	+3_3V	

信号名称	引脚名称	TJ375 引脚名称	
	+3_3V		
PCIE_EDGE_PERSTn	PERST_N	BR0_GPIOR_141_PERST_Q0_N	
GND	GND		
PCIE_EDGE_REFCLK_P	REFCLK+	Q0_REFCLK0_P	
PCIE_EDGE_REFCLK_N	REFCLK-	Q0_REFCLK0_N	
GND	GND		
PCIE_TX0_P	PEROP	Q0_TXDP3	
PCIE_TX0_N	PERON	Q0_TXDN3	
GND	GND		
	RSVD2		
GND	GND		
PCIE_TX1_P	PER1P	Q0_TXDP2	
PCIE_TX1_N	PER1N	Q0_TXDN2	
GND	GND		
GND	GND		
PCIE_TX2_P	PER2P	Q0_TXDP1	
PCIE_TX2_N	PER2N	Q0_TXDN1	
GND	GND		
GND	GND		
PCIE_TX3_P	PER3P	Q0_TXDP0	
PCIE_TX3_N	PER3N	Q0_TXDN0	
GND	GND		
	RSVD4		
VCC_12V	+12V		
VCC_12V	+12V		
VCC_12V	+12V		
GND	GND		
	SMCLK		
	SMDAT		
GND	GND		
	+3_3V		
	JTAG_TRSTN		
	+3_3VAUX		
PCIE_EDGE_WAKEn	WAKE_N	BR0_GPIOR_144_PERST_Q2_N	
	RSVD1		
GND	GND		
PCIE_RX0_P	PETOP	Q0_RXDP3	
PCIE_RX0_N	PETON	Q0_RXDN3	
GND	GND		
PCIE_PRSNT2n_x1	PRSNT2_N_X1		
GND	GND		
PCIE_RX1_P	PET1P	Q0_RXDP2	
PCIE_RX1_N	PET1N	Q0_RXDN2	
GND	GND		

信号名称	引脚名称	TJ375 引脚名称
GND	GND	
PCIE_RX2_P	PET2P	Q0_RXDP1
PCIE_RX2_N	PET2N	Q0_RXDN1
GND	GND	
GND	GND	
PCIE_RX3_P	PET3P	Q0_RXDP0
PCIE_RX3_N	PET3N	Q0_RXDN0
GND	GND	
	RSVD3	
PCIE_PRSNT2n_x4	PRSNT2_N_X4	
GND	GND	

J1 接口 (电源)

J1为12V直流电源输入插孔,其尺寸为DC5525。J1为TJ375开发板的所有DC-DC模块供电,其输入最大电流为 6.25 A。

J2 接口 (风扇)

J2 是 2 针 12 V 风扇接口。

J3 排针 (FMC 接口电压选择)

J3 是 6 针排针,用于选择 FMC 接口的电压。

表 9: J3 (FMC 接口电压选择)

连接	电压
NC	1.8V
1和2	1.5V
3 和 4	1.35V
5和6	1.2V

开发板上也有丝印标识,您也可以参考丝印来跳线设置需要的电压。

## J4 (Micro-SD卡槽)

钛金系列TJ375 N1156X 开发板 包括一个Micro-SD 卡槽 J4。 J4 连接到 bank TL 中的 GPIO 引脚。 micro-SD 最高可支持 25MB/s的数据速率。

### 表 10: J4 引脚分配

引脚名称	信号名称	TJ375 引脚名称
DATA2	SDIO_D2	TL1_GPIOL_45
CD/DATA3	SDIO_D3	TL5_GPIOL_87_PLLIN1
CMD	SDIO_CMD	TL5_GPIOL_85_PLLIN1
VDD	VCC_3V3	_
CLK	SDIO_CK	TL5_GPIOL_86
VSS	GND	_
DATA0	SDIO_D0	TL5_GPIOL_84
DATA1	SDIO_D1	TL5_GPIOL_79_CLK28
CD	TFCard_Detect	TL5_GPIOL_80_CLK29
GND_1	GND	-
GND_2	GND	-
GND_3	GND	-
GND_4	GND	-

## J5 (VQPS 电源使能选择)

J5 是 2 针排针,用于选择 VQPS 电源使能。 如果用跳线帽连接pin1和pin2,则 VQPS 使能被拉高。

# J6 排针 (QSE GPIO 电压选择)

J6 是 8 针排针,用于选择匹配 QSE 接口的 GPIO 电压。

### 表 11: J6 (QSE GPIO电压选择)

连接	电压
1 和 2	1 9\/
3 和 4	1.0V
5 和 6	2 2)/
7和8	5.5V

开发板上也有丝印标识, 您也可以参考丝印来跳线设置需要的电压。

# J7 排针 (MIPI GPIO 电压选择)

J7 是 8 针排针,用于选择匹配 MIPI 接口的 GPIO 电压。

#### 表 12: J7 (MIPI GPIO电压选择)

连接	电压
1和2	1 9\/
3 和 4	1.0V
5和6	1 2)/
7和8	1.2 V

开发板上也有丝印标识, 您也可以参考丝印来跳线设置需要的电压。

# J8 排针 (PCle 设备检测来源选择)

### J8 是 4 针排针,用于选择 PCle 设备检测来源。

### 表 13: J8 (PCIe 设备检测来源选择)

连接	设备检测引脚
1和2	PCIE_PRSNT2n_x1
3 和 4	PCIE_PRSNT2n_x4

# J9、J10、J11、J12 SFP+(10G) 接口

### J9、J10、J11、J12 是 SFP+(10G) 接口。

### 表 14: J9 引脚分配

信号名称	引脚名称	TJ375 引脚名称
VCC_3V3	VCCR	
VCC_3V3	VCCT	
SFP_A_TX_FAULT	TX_FAULT	BR1_GPIOR_152
SFP_A_TX_DISABLE	TX_DISABLE	BR1_GPIOR_153
SFP_A_RX_LOS	RX_LOS	BR1_GPIOR_154
SFP_A_RS0	RSO	BR3_GPIOR_165_PLLIN1
SFP_A_RS1	RS1	BR3_GPIOR_166
SFP_A_MOD	MOD_ABS	BR3_GPIOR_167
SFP_A_SCL	SCL	BR3_GPIOR_168
SFP_A_SDA	SDA	BR3_GPIOR_169
SFP_A_RX_P	RD_P	Q1_RXDP0
SFP_A_RX_N	RD_N	Q1_RXDN0
SFP_A_TX_P	TD_P	Q1_TXDP0
SFP_A_TX_N	TD_N	Q1_TXDN0
GND	VEET	
GND	VEER	

### 表 15: J10 引脚分配

信号名称	引脚名称	TJ375 引脚名称
VCC_3V3	VCCR	

信号名称	引脚名称	TJ375 引脚名称
VCC_3V3	VCCT	
SFP_B_TX_FAULT	TX_FAULT	BR4_GPIOR_180
SFP_B_TX_DISABLE	TX_DISABLE	BR4_GPIOR_179
SFP_B_RX_LOS	RX_LOS	BR3_GPIOR_171
SFP_B_RS0	RSO	BR4_GPIOR_175
SFP_B_RS1	RS1	BR3_GPIOR_170
SFP_B_MOD	MOD_ABS	BR4_GPIOR_176
SFP_B_SCL	SCL	BR4_GPIOR_178
SFP_B_SDA	SDA	BR4_GPIOR_177
SFP_B_RX_P	RD_P	Q1_RXDP1
SFP_B_RX_N	RD_N	Q1_RXDN1
SFP_B_TX_P	TD_P	Q1_TXDP1
SFP_B_TX_N	TD_N	Q1_TXDN1
GND	VEET	
GND	VEER	

### 表 16: J11 引脚分配

信号名称	引脚名称	TJ375 引脚名称
VCC_3V3	VCCR	
VCC_3V3	VCCT	
SFP_C_TX_FAULT	TX_FAULT	TR5_GPIOR_126
SFP_C_TX_DISABLE	TX_DISABLE	TR5_GPIOR_128
SFP_C_RX_LOS	RX_LOS	TR3_GPIOR_113
SFP_C_RS0	RSO	TR3_GPIOR_109
SFP_C_RS1	RS1	TR3_GPIOR_110
SFP_C_MOD	MOD_ABS	TR3_GPIOR_112
SFP_C_SCL	SCL	TR3_GPIOR_111
SFP_C_SDA	SDA	TR5_GPIOR_125
SFP_C_RX_P	RD_P	Q1_RXDP2
SFP_C_RX_N	RD_N	Q1_RXDN2
SFP_C_TX_P	TD_P	Q1_TXDP2
SFP_C_TX_N	TD_N	Q1_TXDN2
GND	VEET	
GND	VEER	

# 表 17: J12 引脚分配

信号名称	引脚名称	TJ375 引脚名称
VCC_3V3	VCCR	
VCC_3V3	VCCT	
SFP_D_TX_FAULT	TX_FAULT	TR3_GPIOR_114
SFP_D_TX_DISABLE	TX_DISABLE	TR3_GPIOR_115
SFP_D_RX_LOS	RX_LOS	TR5_GPIOR_132
SFP_D_RS0	RSO	TR5_GPIOR_131
SFP_D_RS1	RS1	TR5_GPIOR_129

信号名称	引脚名称	TJ375 引脚名称
SFP_D_MOD	MOD_ABS	TR5_GPIOR_133
SFP_D_SCL	SCL	TR5_GPIOR_134
SFP_D_SDA	SDA	TR3_GPIOR_116
SFP_D_RX_P	RD_P	Q1_RXDP3
SFP_D_RX_N	RD_N	Q1_RXDN3
SFP_D_TX_P	TD_P	Q1_TXDP3
SFP_D_TX_N	TD_N	Q1_TXDN3
GND	VEET	
GND	VEER	

J13 (SFP+ 光模块 cage)

J13 是 SFP+ 光模块 cage。

J14 和 J15 接口 (FMC)

J14 和 J15 是用于收发器接口的高速 FMC LPC 连接器,支持 2 路 GTX 时钟和 4 路差分时钟。收发器根据 IEEE 802.3ap-2007 规范配置为 10GBASE-KR 模式。

重要: 钛金系列 TJ375 N1156X 开发板 上的 FMC 接口支持 ANSI/VITA 57.1-2008 (建议 5.2)。

#### ANSI/VITA 57.1-2008 (Suggestion 5.2)

PRSNT\_M2C\_L 用于检测模块的存在,并控制 TDI 信号的切换,使其直接连接到 I/O 夹层模块的 TDO 信号,从而绕过该模块并保持 JTAG 链的完整性。

FMC-to-QSE 子卡和 FMC DDR3 & GPIO 子卡不兼容 FMC JTAG。因此,在将子卡插入 钛金系列 TJ375 N1156X 开发板 之前,请确保 TDI 已连接至 TDO,以实现完整的 JTAG 访问。

注意: 仅 J14A、J14D、J14E、J15A、J15D、J15E 子排针连接到 TJ375 中的 I/O 引脚。

#### 表 18: J14 和 J15 支持的子卡

1

以下子卡可用于连接至 FMC 接口的 FMC-to-QSE 适配卡。

子卡	FMC-to-QSE 适配卡上的 QSE 接口				
	J1	J2	<b>J</b> 3		
Coral 摄像头连接子卡					
树莓派摄像头连接子卡					
双树莓派摄像头连接子卡					
IMX477 摄像头连接子卡					
以太网连接子卡	~				
HDMI 连接子卡	$\checkmark$	~			
双 MIPI 转 DSI 转换子卡	$\checkmark$				
MIPI 和 LVDS 扩展子卡	$\checkmark$	~	~		

#### 表 19: J14A 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
G6	FMCA_LA0_CC_P	2C_GPIOT_P_23_ PLLIN0	G18	FMCA_LA16_P	2B_GPIOT_P_18
G7	FMCA_LA0_CC_N	2C_GPIOT_N_23	G19	FMCA_LA16_N	2B_GPIOT_N_18
D8	FMCA_LA1_CC_P	2B_GPIOT_P_10_ CLK27_P	D20	FMCA_LA17_P_T	2B_GPIOT_P_19
D9	FMCA_LA1_CC_N	2B_GPIOT_N_10_ CLK27_N	D21	FMCA_LA17_N_T	2B_GPIOT_N_19
H7	FMCA_LA2_P	2B_GPIOT_P_16	C22	FMCA_LA18_CC_P_T	2B_GPIOT_P_12_ CLK31_P
H8	FMCA_LA2_N	2B_GPIOT_N_16	C23	FMCA_LA18_CC_N_T	2B_GPIOT_N_12_ CLK31_N
G9	FMCA_LA3_P	2B_GPIOT_P_17	H22	FMCA_LA19_P_T	2B_GPIOT_P_20
G10	FMCA_LA3_N	2B_GPIOT_N_17	H23	FMCA_LA19_N_T	2B_GPIOT_N_20
H10	FMCA_LA4_P	2A_GPIOT_P_03	G21	FMCA_LA20_P_T	2B_GPIOT_P_21
H11	FMCA_LA4_N	2A_GPIOT_N_03	G22	FMCA_LA20_N_T	2B_GPIOT_N_21
D11	FMCA_LA5_P	2B_GPIOT_P_15	H25	FMCA_LA21_P_T	2B_GPIOT_P_22

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
D12	FMCA_LA5_N	2B_GPIOT_N_15	H26	FMCA_LA21_N_T	2B_GPIOT_N_22
C10	FMCA_LA6_P	2C_GPIOT_P_24_EXTFB	G24	FMCA_LA22_P	2C_GPIOT_P_28
C11	FMCA_LA6_N	2C_GPIOT_N_24	G25	FMCA_LA22_N	2C_GPIOT_N_28
H13	FMCA_LA7_P	2A_GPIOT_P_02	D23	FMCA_LA23_P	2A_GPIOT_P_04
H14	FMCA_LA7_N	2A_GPIOT_N_02	D24	FMCA_LA23_N	2A_GPIOT_N_04
G12	FMCA_LA8_P	2A_GPIOT_P_07_EXTFB	H28	FMCA_LA24_P_T	2C_GPIOT_P_26
G13	FMCA_LA8_N	2A_GPIOT_N_07	H29	FMCA_LA24_N_T	2C_GPIOT_N_26
D14	FMCA_LA9_P	2B_GPIOT_P_09_ CLK26_P	G27	FMCA_LA25_P	2C_GPIOT_P_27
D15	FMCA_LA9_N	2B_GPIOT_N_09_ CLK26_N	G28	FMCA_LA25_N	2C_GPIOT_N_27
C14	FMCA_LA10_P	2B_GPIOT_P_14_EXTFB	D26	FMCA_LA26_P	2C_GPIOT_P_25
C15	FMCA_LA10_N	2B_GPIOT_N_14	D27	FMCA_LA26_N	2C_GPIOT_N_25
H16	FMCA_LA11_P	2A_GPIOT_P_05	C26	FMCA_LA27_P	2C_GPIOT_P_30
H17	FMCA_LA11_N	2A_GPIOT_N_05	C27	FMCA_LA27_N	2C_GPIOT_N_30
G15	FMCA_LA12_P	2A_GPIOT_P_06	H31	FMCA_LA28_P	2C_GPIOT_P_31
G16	FMCA_LA12_N	2A_GPIOT_N_06	H32	FMCA_LA28_N	2C_GPIOT_N_31
D17	FMCA_LA13_P	2A_GPIOT_P_00	G30	FMCA_LA29_P	2C_GPIOT_P_29
D18	FMCA_LA13_N	2A_GPIOT_N_00	G31	FMCA_LA29_N	2C_GPIOT_N_29
C18	FMCA_LA14_P	2A_GPIOT_P_01			
C19	FMCA_LA14_N	2A_GPIOT_N_01			
H19	FMCA_LA15_P	2A_GPIOT_P_08_PLLIN0			
H20	FMCA_LA15_N	2A_GPIOT_N_08			

### 表 20: J14D 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
C2	FMCA_DP0_C2M_P	Q3_TXDP0	C6	FMCA_DP0_M2C_P	Q3_RXDP0
C3	FMCA_DP0_C2M_N	Q3_TXDN0	C7	FMCA_DP0_M2C_N	Q3_RXDN0
A22	FMCA_DP1_C2M_P	Q3_TXDP1	A2	FMCA_DP1_M2C_P	Q3_RXDP1
A23	FMCA_DP1_C2M_N	Q3_TXDN1	A3	FMCA_DP1_M2C_N	Q3_RXDN1
A26	FMCA_DP2_C2M_P	Q3_TXDP2	A6	FMCA_DP2_M2C_P	Q3_RXDP2
A27	FMCA_DP2_C2M_N	Q3_TXDN2	A7	FMCA_DP2_M2C_N	Q3_RXDN2
A30	FMCA_DP3_C2M_P	Q3_TXDP3	A10	FMCA_DP3_M2C_P	Q3_RXDP3
A31	FMCA_DP3_C2M_N	Q3_TXDN3	A11	FMCA_DP3_M2C_N	Q3_RXDN3
			D4	FMCA_GBTCLK_M2C_P	Q3_REFCLK0_P
			D5	FMCA_GBTCLK_M2C_N	Q3_REFCLK0_N

### 表 21: J14E 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
D32	VCC_3V3		E39	FMC_VADJ	VCCIO2A
D40	VCC_3V3		F40	FMC_VADJ	VCCIO2A
C39	VCC_3V3		G39	FMC_VADJ	VCCIO2A
D36	VCC_3V3		H40	FMC_VADJ	VCCIO2A
D38	VCC_3V3		F1	VCC_3V3	

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
C35	VCC_12V		D34	VCC_3V3	
C37	VCC_12V		D33	TMS_FMCA	
D1	FMCA_C2M_PG	TR0_GPIOR_89	D31	TDO_FMCA	
H2	FMCA_PRSNT	TR0_GPIOR_90	D30	TDO_FPGA	BR4_TDO
C31	FMCA_SDA	BL0_GPIOL_07	D29	TCK_FMCA	
C30	FMCA_SCL	BL0_GPIOL_08	H4	FMCA_CLK0_M2C_P	2B_GPIOT_P_13_PLLIN0
C34	FMCA_GA0	TR1_GPIOR_98	H5	FMCA_CLK0_M2C_N	2B_GPIOT_N_13
D35	FMCA_GA1	TR1_GPIOR_99	G2	FMCA_CLK1_M2C_P	2B_GPIOT_P_11_ CLK30_P
			G3	FMCA_CLK1_M2C_N	2B_GPIOT_N_11_ CLK30_N

### 表 22: J15A 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
G6	FMCB_LA0_CC_P	2E_GPIOT_P_62_PLLIN0	G18	FMCB_LA16_P	2C_GPIOT_P_35_ CLK18_P
G7	FMCB_LA0_CC_N	2E_GPIOT_N_62	G19	FMCB_LA16_N	2C_GPIOT_N_35_ CLK18_N
D8	FMCB_LA1_CC_P	2E_GPIOT_P_59_PLLIN0	D20	FMCB_LA17_P	2D_GPIOT_P_43
D9	FMCB_LA1_CC_N	2E_GPIOT_N_59	D21	FMCB_LA17_N	2D_GPIOT_N_43
H7	FMCB_LA2_P	2E_GPIOT_P_53	C22	FMCB_LA18_CC_P	2D_GPIOT_P_50_PLLIN0
H8	FMCB_LA2_N	2E_GPIOT_N_53	C23	FMCB_LA18_CC_N	2D_GPIOT_N_50
G9	FMCB_LA3_P	2E_GPIOT_P_54	H22	FMCB_LA19_P	2D_GPIOT_P_40_ CLK23_P
G10	FMCB_LA3_N	2E_GPIOT_N_54	H23	FMCB_LA19_N	2D_GPIOT_N_40_ CLK23_N
H10	FMCB_LA4_P	2C_GPIOT_P_33_ CLK16_P	G21	FMCB_LA20_P	2D_GPIOT_P_47
H11	FMCB_LA4_N	2C_GPIOT_N_33_ CLK16_N	G22	FMCB_LA20_N	2D_GPIOT_N_47
D11	FMCB_LA5_P	2C_GPIOT_P_36_ CLK19_P	H25	FMCB_LA21_P	2E_GPIOT_P_60_EXTFB
D12	FMCB_LA5_N	2C_GPIOT_N_36_ CLK19_N	H26	FMCB_LA21_N	2E_GPIOT_N_60
C10	FMCB_LA6_P	2C_GPIOT_P_34_ CLK17_P	G24	FMCB_LA22_P	2D_GPIOT_P_45
C11	FMCB_LA6_N	2C_GPIOT_N_34_ CLK17_N	G25	FMCB_LA22_N	2D_GPIOT_N_45
H13	FMCB_LA7_P	2E_GPIOT_P_51	D23	FMCB_LA23_P	2D_GPIOT_P_39_ CLK22_P
H14	FMCB_LA7_N	2E_GPIOT_N_51	D24	FMCB_LA23_N	2D_GPIOT_N_39_ CLK22_N
G12	FMCB_LA8_P	2E_GPIOT_P_56	H28	FMCB_LA24_P	2E_GPIOT_P_64_ CLK15_P
G13	FMCB_LA8_N	2E_GPIOT_N_56	H29	FMCB_LA24_N	2E_GPIOT_N_64_ CLK15_N
D14	FMCB_LA9_P	2D_GPIOT_P_49_EXTFB	G27	FMCB_LA25_P	2E_GPIOT_P_63_ CLK14_P

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
D15	FMCB_LA9_N	2D_GPIOT_N_49	G28	FMCB_LA25_N	2E_GPIOT_N_63_ CLK14_N
C14	FMCB_LA10_P	2D_GPIOT_P_46	D26	FMCB_LA26_P	2D_GPIOT_P_42
C15	FMCB_LA10_N	2D_GPIOT_N_46	D27	FMCB_LA26_N	2D_GPIOT_N_42
H16	FMCB_LA11_P	2E_GPIOT_P_55	C26	FMCB_LA27_P	2D_GPIOT_P_48
H17	FMCB_LA11_N	2E_GPIOT_N_55	C27	FMCB_LA27_N	2D_GPIOT_N_48
G15	FMCB_LA12_P	2E_GPIOT_P_58	H31	FMCB_LA28_P	2E_GPIOT_P_61_EXTFB
G16	FMCB_LA12_N	2E_GPIOT_N_58	H32	FMCB_LA28_N	2E_GPIOT_N_61
D17	FMCB_LA13_P	2D_GPIOT_P_41	G30	FMCB_LA29_P	2E_GPIOT_P_57
D18	FMCB_LA13_N	2D_GPIOT_N_41	G31	FMCB_LA29_N	2E_GPIOT_N_57
C18	FMCB_LA14_P	2D_GPIOT_P_44			
C19	FMCB_LA14_N	2D_GPIOT_N_44			
H19	FMCB_LA15_P	2E_GPIOT_P_52			
H20	FMCB_LA15_N	2E_GPIOT_N_52			

## 表 23: J15D 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
C2	FMCB_DP0_C2M_P	Q2_TXDP3	C6	FMCB_DP0_M2C_P	Q2_RXDP3
C3	FMCB_DP0_C2M_N	Q2_TXDN3	C7	FMCB_DP0_M2C_N	Q2_RXDN3
A22	FMCB_DP1_C2M_P	Q2_TXDP2	A2	FMCB_DP1_M2C_P	Q2_RXDP2
A23	FMCB_DP1_C2M_N	Q2_TXDN2	A3	FMCB_DP1_M2C_N	Q2_RXDN2
A26	FMCB_DP2_C2M_P	Q2_TXDP1	A6	FMCB_DP2_M2C_P	Q2_RXDP1
A27	FMCB_DP2_C2M_N	Q2_TXDN1	A7	FMCB_DP2_M2C_N	Q2_RXDN1
A30	FMCB_DP3_C2M_P	Q2_TXDP0	A10	FMCB_DP3_M2C_P	Q2_RXDP0
A31	FMCB_DP3_C2M_N	Q2_TXDN0	A11	FMCB_DP3_M2C_N	Q2_RXDN0
			D4	FMCB_GBTCLK_M2C_P	Q2_REFCLK0_P
			D5	FMCB_GBTCLK_M2C_N	Q2_REFCLK0_N

### 表 24: J15E 引脚分配

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
D32	VCC_3V3		E39	FMC_VADJ	VCCIO2A
D40	VCC_3V3		F40	FMC_VADJ	VCCIO2A
C39	VCC_3V3		G39	FMC_VADJ	VCCIO2A
D36	VCC_3V3		H40	FMC_VADJ	VCCIO2A
D38	VCC_3V3		F1	VCC_3V3	
C35	VCC_12V		D34	VCC_3V3	
C37	VCC_12V		D33	TMS_FMCB	
D1	FMCB_C2M_PG	TR1_GPIOR_100	D31	TDO_FT	
H2	FMCB_PRSNT	TR2_GPIOR_102	D30	TDO_FMCA	
C31	FMCB_SDA	TR2_GPIOR_103	D29	TCK_FMCB	
C30	FMCB_SCL	TR2_GPIOR_106	H4	FMCB_CLK0_M2C_P	2D_GPIOT_P_37_ CLK20_P
C34	FMCB_GA0	TR2_GPIOR_107	H5	FMCB_CLK0_M2C_N	2D_GPIOT_N_37_ CLK20_N

引脚号	信号名称	TJ375引脚名称	引脚号	信号名称	TJ375引脚名称
D35	FMCB_GA1	TR2_GPIOR_108	G2	FMCB_CLK1_M2C_P	2D_GPIOT_P_38_ CLK21_P
			G3	FMCB_CLK1_M2C_N	2D_GPIOT_N_38_ CLK21_N

# J16 排针 (JTAG)

J16 是 10 针 JTAG 接口。 您可以通过此接口访问 TJ375 JTAG 引脚。

### 表 25: J16 引脚分配

引脚号	信号名称
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

## J18 排针 (Bank BL0 GPIO 电压选择)

### J18 是 8 针排针,用于选择匹配 Bank BL0 的 GPIO 电压。

### 表 26: J18 (Bank BL0 GPIO 电压选择)

连接	电压		
1和2	1 9\/		
3 和 4	1.0V		
5 和 6	2 2)/		
7和8	5.5V		

开发板上也有丝印标识,您也可以参考丝印来跳线设置需要的电压。当 GPIO 配置为输入时,需要在软件 Interface Designer 中打开 weak pull-up。

# RJ1 (千兆以太网口)

钛金系列TJ375 N1156X 开发板 集成来自瑞昱的干兆以太网收发器(器件号: RTL8211FI, 兼容10Base-T、100Base-TX和1000Base-T IEEE 802.3标准。)该芯片支持:

- RGMII MAC接口
- 支持120米 1000Base-T CAT.5电缆
- 自动极性校正
- 低压降稳压器

RJ1接到Ethernet PHY U29。PHY 地址为 0x1, 默认配置为在 TXC 和 RXC 添加了 2 ns 延迟。

#### 表 27: U29 ETH PHY 引脚分配

信号名称	引脚名称	TJ375 引脚名称	说明	
RGMII0_TXD0	TXD0	4C_GPIOB_P_17_CLK4_P		
RGMII0_TXD1	TXD1	4C_GPIOB_N_17_CLK4_N		
RGMII0_TXD2	TXD2	4C_GPIOB_P_18_CLK5_P	数据通过 IXD[3:0] 从 MAC   发送到PHY。	
RGMII0_TXD3	TXD3	4C_GPIOB_N_18_CLK5_N		
RGMII0_TXEN	TXCTL	4C_GPIOB_N_19_ TEST_N_CLK6_N	来自 MAC 的发送控制信 号。	
RGMII0_CLK	TXCLK	4C_GPIOB_P_19_ NSTATUS_CLK6_P	不同速率下,发送参考时钟 可为 125MHz、25MHz 或 2.5MHz。	
RGMII0_RXD0	RXD0/RXDLY	4C_GPIOB_P_21		
RGMII0_RXD1	RXD1/TXDLY	4C_GPIOB_N_21		
RGMII0_RXD2	RXD2/PLLOFF	4C_GPIOB_P_22	数据通过 RXD[3:0] 从 PHY   发送到MAC。	
RGMII0_RXD3	RXD3/PHYAD0	4C_GPIOB_N_22		
RGMII0_RXCK	RXC/PHYAD1	4C_GPIOB_P_20_CLK7_P	连续接收参考时钟可为 125MHz、25MHz 或 2.5MHz。时钟来自接收数 据流。	
RGMII0_RXDV	RXCTL/PHY_AD2	4C_GPIOB_N_20_CLK7_N	发送至 MAC的接收控制信 号。	
ETH_MDC	MDC	4C_GPIOB_N_28	管理数据时钟。	
ETH_MDIO	MDIO	4C_GPIOB_P_29_ CDI11_EXTFB	管理数据输入/输出。 3.3/2.5/1.8/1.5V I/O 分别 上拉 3.3/2.5/1.8/1.5V。	
ETH_RSTN	PHYRSTB	BR0_GPIOR_142_CLK8	硬件复位。低电平有效。 一次完整的 PHY 复位要 求此引脚必须拉低至少 10ms。硬件复位后,所有 寄存器都被清除。	

信号名称	引脚名称	TJ375 引脚名称	说明
eth_int	INTB/PMEB	BR0_GPIOR_143_CLK9	此引脚支持两个功能,如果 其中一个功能未使用,则应 保持此引脚悬空。 1.中断(支持 3.3V 上拉)。 如果发生指定事件则拉低; 低电平有效。 2.电源管理事件(支持 3.3V 上拉)。 如果收到魔术包或唤醒帧或 唤醒事件则拉低;低电平有 效。 注1:此引脚的行为由电平 触发。 注2:此引脚(INTB/ PMEB)的功能可通过 Page Oxd40、Reg.22、位[5]指 定: 1:引脚 31 用作 PMEB。 0:引脚 31 用作 INTB(默 认)
LED0_10M/CFG_EXT	LED0/CFG_EXT		高电平=链路连接速率 10Mbps 闪烁=正在发送或接收。
LED1_100M/CFG_LDO0	LED1/CFG_LDO0		低电平=链路连接速率 100Mbps 闪烁=正在发送或接收。
LED2_1000M/CFG_LDO1	LED2/CFG_LDO1		高电平=链路连接速率 1000Mbps 闪烁=正在发送或接收。

# USB1接口 (USB FTDI FT4232H)

USB1是一个 Type-C USB 插座,是开发板和电脑通过 FTDI FT4232H 芯片进行通信的接 口。 连接 Type-C USB电缆来配置 TJ375 FPGA 或烧录 NOR Flash。

FTDI FT4232H模块有四条 channel, 支持以下接口:

- FTDI interface 0—— 用于RISC-V调试的JTAG
  FTDI interface 1—— 用于FPGA调试的JTAG
- FTDI interface 2-FPGA UART
- FTDI interface 3—— /

TP1、TP2、TP3、TP4 测试点(接地)

测试点TP1、TP2、TP3、TP4 是接地测试点。

# 用户输出

开发板有 6 个用户 LED 灯,连接到 TJ375 bank 4B 中的 I/O 引脚。默认情况下,连接到 这些 LED 的 TJ375 I/O 为高电平有效。要打开特定的 LED 灯,请设置对应的 I/O 管脚输出高电 平。



注意:在 Interface Designer 中添加这些GPIO时,将其配置为输出引脚。

表 28: 用户输出

参考标号	TJ375 引脚名称	有效电平
LED1	4B_GPIOB_N_41	高
LED2	4B_GPIOB_P_42	高
LED3	4B_GPIOB_N_42	高
LED4	4B_GPIOB_P_33	高
LED5	4B_GPIOB_P_34	高
LED6	4B_GPIOB_P_35	高

# 用户输入

开发板有 2 个可以用作 TJ375 FPGA 输入的按钮开关。按钮连接到 TJ375 bank 4B 中的 I/ O 引脚。 使用这些开关构建设计时,请在 Interface Designer 中打开这些引脚的 internal pullup。

用户开关默认高电平,当按下开关时,I/O引脚输入低电平。

表 29: 用户按钮

参考标号	TJ375 引脚名称	有效电平
SW3	4B_GPIOB_P_31	低
SW4	4B_GPIOB_P_32	低

# 钛金系列TJ375 N1156X 开发板示例设计

易灵思 在 钛金系列TJ375 N1156X 开发板 中预载了一个示例设计,包括以下功能:

- 在主机系统和 PCIe 设备间建立 PCIe 链路,主机系统 CPU 为 PCIe 根端口(控制器), 钛金系列 TJ375 N1156X 开发套件 为 PCIe 端点(终端设备)。钛金系列TJ375 N1156X 开发板 插入 PCIe 插槽后,主机系统和 钛金系列TJ375 N1156X 开发板 之间建立 PCIe 链路,可以进行 PCIe 操作访问。
- 简单读操作, 钛金系列TJ375 N1156X 开发板 发送数据到主机。
- 简单写操作, 主机写数据到 钛金系列TJ375 N1156X 开发板。

整体示例设计包括针对 TJ375N1156X FPGA 的 PCIe 端点设计和用于主机系统执行简单读/写操作的软件驱动程序。您可以通过 Linux 内核、C 代码或 Python 脚本执行读/写操作。

图 4: 示例设计框图概述



# 设置硬件

硬件设置需要一个具有可用 PCle 插槽,运行 Ubuntu 操作系统的工作站。硬件设置步骤见下图。





重要:在连接或拆卸电缆和子卡之前,请务必关闭开发板的电源开关。

1. 确保开发板的跳线帽如下设置:

开发板	排针	连接引脚
钛金系列TJ375 N1156X 开发板	J6、J7、J18	5 - 6 7 - 8
	J8	3 - 4

- 2. 将 钛金系列TJ375 N1156X 开发板 插入工作站的 PCle 插槽。确认开发板 没有 连接到 12V DC 电源。
- **3.** 使用 USB Type-C 电缆连接 钛金系列TJ375 N1156X 开发板 和电脑(已安装 Primus 软件)上的 USB 端口。
- 4. 打开工作站。钛金系列TJ375 N1156X 开发板 将通过 PCIe 插槽供电,无需连接 12V DC 电源。
- 5. 按下 SW2 按钮 (CRESET N)。

至此, 钛金系列TJ375 N1156X 开发板 硬件设置完成。

开发板 LED灯描述见下表。

#### 表 30: 开发板LED输出

LED灯	说明
LED1 亮起	相关设计完成
LED2 亮起	相关设计完成
LED3 亮起	相关设计完成
LED4 亮起	相关设计完成
LED5 亮起	相关设计完成
LED6 亮起	相关设计完成
LED7亮起	正常通电
LED8 亮起	FPGA 配置完成

# 运行示例设计

电脑开机后, 钛金系列TJ375 N1156X 开发板 将通电。设计将自动从板载闪存器件加载。

### 枚举状态

PCIe 枚举是检测和识别连接到 PCIe 总线设备的过程。系统扫描设备、访问其配置空间并分配必要的资源,如内存和 I/O 地址。

- 1. 在 Ubuntu 工作站上打开终端。
- 2. 使用 sudo lspci -vv -d 1f7a:0100 命令输出 PCI 类型 0 配置空间的转换描述。 类型 0 配置空间显示 PCIe 端点设备的配置设置,例如链路速度和链路宽度。
  - 图 6: 类型 0 配置空间转换描述

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-<="" th=""></perr-></tabort->
Interrupt: pin A routed to IRO 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- DI+ 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: 00000000000000 Data: 0000
Masking: 00000000 Pending: 00000000
Capabilities: [b0] MSI-X: Enable- Count=1 Masked-
Vector table: BAR=0 offset=00000000
PRA: BAR=0 offset=00000008

#### 图 7: 链路速度和宽度





**注意:** 如果在系统断电重启,SPI 闪存配置完成后,发现 PCIe 链路降级(例如无法达到 PCIe Gen 4 数据速率,或者数据总线宽度小于 x4),请尝试通过 SW2 复位按钮复位 FPGA,然后软重启 PC。

- 3. 如果转换描述未输出,请尝试以下解决方案:
  - 使用 reboot 命令重新启动 Linux 工作站。
    - 如果问题未解决,则可能表明您连接主板的插槽不支持 PCIe。尝试切换到工作 站上的另一个 PCI 插槽。
    - 如果所有 PCI 插槽均不起作用,则可能表明您的工作站不具备 PCI 功能。

### 调试配置文件

端点设计附带一个调试配置文件,用于监控 PCle 流量和状态。vio0 调试内核包含 PCle 状态和 AXI 虚拟 I/O 接口; la0 调试内核具有逻辑分析仪接口,用于监控 AXI 写通道。执行以下步骤连接 Debugger:

- 1. 启动 Primus 软件并打开 Debugger。
- 2. 选择 File > Open Debug Profile。
- 3. 选择 hardware 目录并选择 debug\_profile.json。单击 Open。
- 4. 选择 Perspectives > Debug。
- 5. 选择 JTAG USER1 并连接 Debugger。

下表描述了 vio0 中的 probe 和 source 信号。

### 表 31: Debugger vio0 选项卡

名称	类型	宽度	描述
q0_ltssm_state	Probe	6	LTSSM 状态。详情请参阅 钛金系列 PCle 控制器用户 指南中的"附录 C:LTSSM 状态编码"。
q0_link_status	Probe	2	PCle 链路状态。 2'b00: 未检测到接收端。 2'b01: 链路训练正在进行 中。 2'b10: 链路建立, DL 初始 化正在进行中。 2'b11: 链路建立, DL 初始 化已完成。
q0_p00_rate	Probe	2	收发器 quad 0 的 PIPE 链 路信号速率。 选择数据速率。 2'b00: PCle Gen1 2'b01: PCle Gen2 2'b10: PCle Gen3 2'b11: PCle Gen4
q0_cmn_ready	Probe	1	通用就绪状态。
q0_TARGET_AXI_AWSIZE	Probe	3	AXI 写传输大小。
q0_TARGET_AXI_AWADDR	TARGET_AXI_AWADDR Probe		AXI 写传输地址。
q0_TARGET_AXI_WDATA	Probe	256	AXI 写数据。
q0_TARGET_AXI_ARSIZE	Probe	3	AXI 读传输大小。
q0_TARGET_AXI_ARADDR	Probe	64	AXI 读传输地址。
q0_TARGET_AXI_RDATA	Source	256	AXI 读数据。

了解更多: 有关使用 Debugger 的完整说明, 请参阅 Primus 软件用户指南。

简单读写测试

您可使用示例设计中的文件,使用 Linux 内核模块、Python 脚本或 C 代码执行读/写测试。

#### 使用 Linux 内核模块

#### 此简单脚本在 Linux 操作系统的内核层中运行。它启用内存区域并执行单个读写操作。

注意: 在使用 C 程序或 Python 脚本读取或写入 PCIe 内存区域之前,必须加载 Linux 内核模块。

#### 此方法需要内核头文件。使用以下命令安装内核头文件:

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

要执行读/写测试,请在终端中执行以下步骤:

- 1. 更改为示例设计的 linux-kernel-mod 目录。
- 2. 使用 make all 命令编译源代码。
- **3.** 在 Debugger 的 vio0 选项卡中,将 q0\_TARGET\_AXI\_RDATA 设置为 0x12345678 或 任何 32 位值。读操作在安装内核驱动程序时使用此值。
- 4. 使用命令 sudo insmod els pcie simple rdwr.ko 安装 Linux 内核模块。
- 5. 使用命令 sudo dmesg | grep els pcie 查看内核驱动程序的输出。

\$ sudo insmod els\_pcie\_simple\_rdwr.ko \$ sudo dmesg | grep els\_pcie [ 174.141878] els\_pcie - VENDOR\_ID: 0x1f7a [ 174.141885] els\_pcie - DEVICE\_ID: 0x100 [ 174.141926] els\_pcie - read\_mem\_and\_print: 12345678 [ 174.141929] els\_pcie - write\_mem\_and\_print: 90abcdef

6. 在 Debugger 中, 观察 q0 TARGET AXI WDATA 的值从 0x0 更改为 0x90abcdef。

#### 使用 Python 脚本

此 Python 脚本使用 mmap()系统调用将 PCIe 端点应用映射到操作系统中的虚拟地址。然后,您可以在虚拟地址上执行读/写操作,以从 TJ375 PCIe 控制器发送/接收数据包。

要执行读/写测试,请在终端中执行以下步骤:

- 1. 更改为示例设计的 python-scripts 目录。
- 2. 使用此命令安装依赖项:

\$ sudo pip install -r requirements.txt

- **3.** 在 Debugger 的 vio0 选项卡中,将 q0\_TARGET\_AXI\_RDATA 设置为 0x12345678 或 任何 32 位值。Python 脚本的读操作使用此值。
- **4.** 使用命令 sudo python3 rd\_req\_pci.py 在端点上执行读操作并以小端格式显示结果。

- 5. 在 Debugger 上, 单击 la0 选项卡。
- 6. 添加触发条件以在 q0 TARGET AXI WVALID = 1时捕获波形。
- **7.** 使用命令 sudo python3 wr\_req\_pci.py 执行写操作和读操作。读操作以小端格式 输出结果。

```
$ 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: 31323334353637383930616263646566
16
7856341200000000
```

Debugger 生成显示 AXI 写操作输出的波形。写操作可以在单个 AXI 写操作或多个 AXI 写操作中完成,具体取决于工作站的型号。

#### 使用 C 代码

这个简单的 C 程序使用 sysfs 文件系统执行单个读写操作。

要执行读/写测试,请在终端中执行以下步骤:

- 1. 更改为示例设计的 c-code-sysfs 目录。
- 2. 使用 make main 命令编译源代码。
- 3. 使用命令 find /sys/bus/pci/devices/\*/ -maxdepth 1 -name vendor | xargs grep 0x1f7a 来确认开发板的完整路径名。预期的输出为 /sys/bus/pci/ devices/<BDF>/vendor:0x1f7a。

\$ find /sys/bus/pci/devices/\*/ -maxdepth 1 -name vendor | xargs grep 0x1f7a
/sys/bus/pci/devices/0000:02:00.0/vendor:0x1f7a

- **4.** 在 Debugger 的 vio0 选项卡中,将 q0\_TARGET\_AXI\_RDATA 设置为 0x12345678 或 任何 32 位值。读操作使用此值。
- 5. 使用以下命令运行 C 程序:

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

其中:

[sys file] 是 PCI 资源要操作的 sysfs 文件。 [offset] 是要操作的 PCI 内存区域中的偏移量。 [type] 是访问操作类型, [b]yte、[h]alfword、[w]ord、[d]ouble-word。 [data] 是要写入的数据。

sudo ./pcimem /sys/bus/pci/devices/<BDF>/resource0 0 w 0x10203040

以下示例使用 0000:02:00.0 BDF。

\$ sudo ./pcimem /sys/bus/pci/devices/0000:02:00.0/resource0 0 w 0x10203040
/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: 0x12345678
Written 0x10203040; readback 0x12345678

在 Debugger 中, 观察 q0 TARGET AXI WDATA 的值从 0x0 更改为 0x10203040。

# 自定义设计

钛金系列TJ375 N1156X 开发板 可让您创建和探索 TJ375 FPGA 的设计。易灵思<sup>®</sup> 会提供示例代码和设计来帮助您入门:

- 我们的支持中心(www.elitestek.com/support)会提供针对开发板的示例。
- Primus<sup>®</sup> 软件也包括示例设计,您可以使用这些设计来开始设计自己的项目。您也可以在 软件中获取详细教程,跟随教程一步步开始设计。

# 恢复示例设计

在将该开发板用于其他设计后,您可能想返回到原始的预加载示例设计。预加载的示例设计项目文件可在 技术支持 获取。要恢复示例设计,您需要将 TJ375 示例设计比特流烧录进开发板的 SPI 闪存器件。



注意: 技术支持 提供的示例设计需要 Primus 软件版本 v2024.1 Patch 4 或更高版本才能运行。

#### 图 8: SPI Flash 内容设计示例



# 示例设计文件

示例设计包括以下设计文件。

### 表 32: 设计示例文件和路径

文件或路径	说明
tj375n1156x_oob/hardware/tj375n1156x_oob.xml	示例设计项目文件。
tj375n1156x_oob/hardware/bitstream/ tj375n1156x_oob.hex	仅 FPGA 位流。

# 烧录文件到开发板

**注意:** 本节介绍在 Programmer 中 SPI Active using JTAG bridge 模式下,如何使用 **.hex** 和 **.bit** 文件烧录示例设计 FPGA 位流 文件。

在烧录文件到开发板之前,请使用 USB 电缆将 钛金系列TJ375 N1156X 开发板 连接到您的 电脑,并打开开发板电源。请参阅 第37页的表 32:设计示例文件和路径。

示例设计 zip 文件包含比特流文件, 您可快速恢复到原始设计。执行以下步骤将其下载到开 发板上:

- 1. 从 技术支持 下载文件 tj375n1156x\_oob\_v1\_0.zip。
- 2. 在 Primus 软件中打开工程
  - (tj375n1156x oob.xml)。工程目录可参考 第37页的表 32:设计示例文件和路径。
- 3. 使用 USB 电缆将 钛金系列TJ375 N1156X 开发板 连接到您的电脑。

- 4. 使用 Programmer 将比特流文件 bitstream/tj375n1156x\_oob.hex 下载到您的开发 板上,模式选择 SPI Active using JTAG Bridge (new), Starting Flash Address 设置为 0x000000。
- 5. 点击 Start Program 即可开始烧录。

)				Prog	rammer			~ ^
le Help								
🛞 🗴 📴	🖌 🔍	?	C=010					
Target								
USB Target	TJ3751	V1156X De	velopment Ki	it				
USB Info	Bus 00	1 Device 0	65: ID 0403:60	011				
Image								
Ritstroom Filo	/homo/kon	ny/project	+/ti275p1156	( oob b	ov.			
Ditstream File	/nome/ken	iny/projec	(/tj3/511150)	(_000.11	ex			
					FPGA	TJ375N1156X	Checksum BEC	006B45
Programming Mc	de							
		(					_ ]	
SPI Active using	g J I AG Bridge	e (new)					•	010-
Auto conf	gure JTAG B	ridge Imag	e					
Bitstream File	'home/ke	nny/primu	s/2024.1/pgn	n/fli/tita	nium/u00	6A0EF3.bit 🛽 🗴	FPGA TJ375	5 01
SPI Active Optic	ns				JTAG Opt	tions		
Starting Flash	Address	(	0x00000000		Device S	Select	0x006a0ef3	•
Flash Length		393	5907 Bytes	<b>P</b>	JTAG CI	ock Speed	6 MHz	•
✓ Frase Befo	re Programn	ning						
Verifi Afte	. Due energia							
Verity Afte	r Programmi	ng						
Device Configura	tion Status							
Last Updat	ed: Tue Sep 2	4 24 04:50	:10 PM   🤇 🎽					
nsole								
Tue Septembe	r 24 24 16:5	0:11 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	
Tue Septembe	r 24 24 16:5	50:11 - Bo	lid device IE	) found	: 0x006A	0EF3	F14232	
Tue Septembe	r 24 24 16:5	0:23 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	
Tue Septembe	r 24 24 16:5	0:25 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	
Tue Septembe	r 24 24 16:5	60:25 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	
Tue Septembe	r 24 24 16:5	60:25 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	
Tue Septembe	r 24 24 16:5	60:52 - Bo	ard Profile:	Generi	c Board I	Profile Using	FT4232	

6. 按下开发板上的 creset 按钮,固件开始运行。



### 表 33: 修订记录

日期	版本	说明
2025年6月	1.3	在概述中添加了开发板的机械图。
2025年4月	1.2	在FMC 连接器章节中添加了FMC J14 和 J15 支持的子卡 表格及备注。 在 使用Linux内核模块 章节中增加了运行 C 程序和 Python 脚本前需先安装 Linux 内核的必备条件说明。 添加了 第15页的P5 接口 (PCI Express 金手指)的说明。
2024年12月	1.1	更新 运行示例设计 章节部分描述。
2024年11月	1.0	首次发布。