(机器翻译成中文)

Getting started with the FPGA demo bundle for Intel FPGA

Xillybus Ltd.

www.xillybus.com

Version 3.1

本文档已由计算机自动翻译，可能会导致语言不清晰。 与原始文件相比，该文件也可能略微过时。

如果可能，请参阅英文文档。

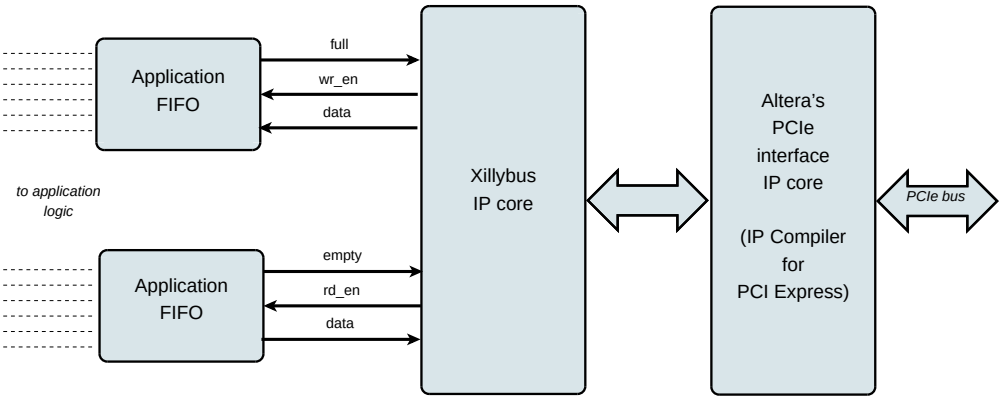
This document has been automatically translated from English by a computer, which may result in unclear language. This document may also be slightly outdated in relation to the original.

If possible, please refer to the document in English.

1

介绍

Xillybus 是基于 DMA 的端到端解决方案，用于在 FPGA 和运行 Linux 或 Microsoft Windows 的 host 之间传输数据。它为 FPGA logic 的设计者和软件的程序员提供了一个简单直观的界面。



如上图，FPGA上的application logic只需要与标准的FIFOs交互即可。

例如，将数据写入图中较低的 FIFO 会使 Xillybus IP core 感觉到数据可以在 FIFO 的另一端进行传输。很快，IP core 就从 FIFO 读取数据并发送给 host，让 userspace software 可以读取。数据传输机制对 FPGA 中的 application logic 是透明的，仅与 FIFO 交互。

另一方面，Xillybus IP core 利用 PCI Express 的 Transport Layer level 实现数据流，生成和接收 TLP 数据包。对于较低的层，它依赖于 Intel 的官方 PCIe core，它是开发工具的一部分，并且不需要额外的许可（即使使用 Quartus 的 Web/ Lite Edition）。

计算机上的应用程序与 device files 交互，其行为类似于 named pipes。 Xillybus IP core 和 driver 在 FPGAs 中的 FIFOs 和 host 上的相关 device files 之间高效且直观地传输数据。

使用XillyUSB，一个MGT transceiver用于实现一个USB 3.0接口，用于数据传输，而不是上面提到的PCIe接口。

IP core 使用在线 Web 应用程序根据客户的规格即时构建。 streams 的数量、方向和其他属性由客户定义，以实现 design 的带宽性能、同步和简单性之间的最佳平衡。按照本指南中的说明使用 demo bundle 完成准备步骤后，建议在 http://xillybus.com/custom-ip-factory 上构建和下载您的自定义 IP core。

本指南介绍如何使用 Xillybus IP core 快速设置 FPGA，Xillybus IP core 可以附加到用户提供的数据源和数据消费者，用于实际应用场景测试。 IP core 包含在 demo bundle 中，可以在网站上下载。

尽管名称如此，demo bundle 并不是一个演示套件，而是一个功能齐全的 starter design，它可以按原样执行有用的任务。

对于那些好奇的人，可以在 Xillybus host application programming guide for Linux 或 Xillybus host application programming guide for Windows的附录 A 中找到关于如何实现 Xillybus 的简要说明。

2

先决条件

2.1 硬件

Xillybus 依赖于 Intel 的 hardware IP block for PCI Express，因此可用于任何具有此组件的 Intel FPGA 设备。在支持的 FPGA 系列中：

- Arria II GX/GZ
- Cyclone IV GX
- HardCopy IV GX
- Stratix IV GX
- Arria V GX/GT/SX/ST
- Cyclone V GX/GT/SX/ST
- Stratix V GS/GX/GT
- Arria 10 GX/GT/SX
- Cyclone 10 GX
- Stratix 10 与 H-tile 或 L-tile

XillyUSB 仅支持 Cyclone 10 GX（不支持 LP 系列）。

Xillybus FPGA demo bundle 封装后可与多个板和设备一起使用，如下载页面中所列（请参阅下面的 2.2 部分）。

在对 pin placements 进行必要的更改并验证 MGT 的 reference clock 是否得到正确处理后，其他板的所有者可以在自己的硬件上运行 demo bundle。对于任何相当有经验的 FPGA 工程师来说，这应该是直截了当的。在 4.4 节中有更多相关信息。

2.2 FPGA项目

Xillybus demo bundle 可在 Xillybus 网站的下载页面下载。对于基于 PCIe 的 cores：

http://xillybus.com/pcie-download

对于 XillyUSB：

http://xillybus.com/usb-download

demo bundle包含Xillybus IP core的特定配置，用于简单测试。因此，它对于某些应用程序的性能相对较差。

可以使用 IP Core Factory Web 应用程序配置、自动构建和下载自定义 IP cores。请访问 http://xillybus.com/custom-ip-factory 以使用此工具。

任何下载的捆绑包，包括 Xillybus IP core，都可以免费使用，只要这种使用合理地匹配 “evaluation” 一词即可。这包括将 core 集成到最终用户 designs 中，运行真实数据和现场测试。 core 的使用方式没有限制，只要这种使用的唯一目的是评估其功能和对特定应用的适用性。

2.3 开发软件

下面列出了 Xillybus 的 demo bundle（以及其他涉及 Xillybus 的 designs）的 implementation 的推荐工具，具体取决于 FPGA 的系列。

Xillybus 用于 PCIe：

- 对于 series-IV 和 Arria II 的 FPGAs：Quartus 12.0 及更高版本。
- 对于 Arria 10 和 Cyclone 10：Quartus Prime 17.1 及更高版本。 Standard 和 Pro 版本均受支持。
- 对于 Stratix 10：Quartus Pro 19.2 及更高版本。
- 所有其他 FPGA 系列：Quartus II，版本 15.0 及更高版本。

XillyUSB：

- 对于 Cyclone 10：应使用 Quartus Pro 17.1 及更高版本。

该软件可以直接从 Intel 的网站 (https://www.intel.com) 下载。

请注意，Quartus 的 Web/Lite 版本免费提供，支持多个 FPGA 系列，尤其是 Cyclone 设备。

Xillybus 的 implementation 依赖于 Quartus 提供的一些 IP cores。 Quartus 的所有版本都涵盖这些 IP cores，无需任何额外许可。

2.4 使用 FPGA design 的经验

当 design 适用于出现在 demo bundles 列表中的电路板时，无需以前的 FPGA design 经验即可让 demo bundle 在 FPGA 上运行。使用其他板卡时，需要有一定的Intel FPGA工具的使用知识，尤其是pin placements和clocks的定义。

要充分利用 demo bundle，必须充分了解 logic design 技术，以及掌握 HDL 语言（Verilog 或 VHDL）。尽管如此，Xillybus demo bundle 是学习这些的一个很好的起点，因为它提供了一个简单的 starter design 来进行实验。

3

demo bundle 的 implementation

3.1 概述

Xillybus的demo bundle的implementation有三种可能的方法，获取bit stream文件（SOF）：

- 按原样使用包中的项目文件。这是最简单的方法，适用于使用出现在 demo bundles 列表中的板。
- 修改文件以匹配不同的 FPGA。这适用于与其他板和/或其他 FPGAs 一起工作时。 4.4 段中有关此内容的更多信息。
- 从头开始设置 Quartus 项目。将 demo bundle 与现有 application logic 集成时可能需要。 4.3 段中的更多细节。

在本节的其余部分中，将详细介绍第一个工作过程，这是最简单和最常选择的一个。其他两个工作程序基于第一个，差异在上文给出的段落中有详细说明。

重要的:

评估包的配置是为了简单而不是性能。对于需要持续和连续数据流的应用程序，特别是对于高带宽情况，可以获得明显更好的结果。对于这些场景，可以使用 *Web* 应用程序轻松构建和下载自定义 *IP core*。

3.2 文件大纲

该捆绑包由五个目录组成:

- core——Xillybus IP core存放在这里

- instantiation templates– 包含 core 的 instantiation templates（在 Verilog 和 VHDL 中）
- verilog– 包含 demo bundle 的项目文件和 Verilog 中的源代码（在 'src' 子目录中）
- vhdl– 包含 demo bundle 的项目文件和 VHDL 中的源代码（在 'src' 子目录中）
- pcie_core（仅限 PCIe 捆绑包）—— Intel 的 PCIe IP core 在构建捆绑包的其余部分之前在此处构建。
- quartus-essentials（仅限 XillyUSB 捆绑包）- Verilog 和 VHDL 项目通用的各种文件。

请注意，每个 demo bundle 都适用于特定的板，如下载 demo bundle 的站点网页中所列。如果使用了另一块板，或者在板上添加或移除了某些配置电阻，则必须相应地编辑 constraints 文件。

另请注意，vhdl 目录包含 Verilog 文件，但这些文件都不需要进行重大更改。

Xillybus 的 IP core 和 application logic 之间的接口发生在 xillydemo.v 文件或 xillydemo.vhd 文件中（在各自的 'src' 子目录中）。这是要编辑的文件，以便用您自己的数据试用 Xillybus。

3.3 构建 demo bundle

3.3.1 打开项目

根据您的偏好，双击 'verilog' 或 'vhdl' 子目录中的 'xillydemo.qpf' 文件。 Quartus 将使用正确的设置启动并打开项目。

如果使用 series-V（例如 Cyclone V ）或 series-10（例如 Arria 10 ）FPGA，则继续第 3.3.3 段。

否则，如下所述构建 PCIe 块。

3.3.2 构建 PCIe 块（ Arria II 和 series-IV FPGAs ）

重要的:

请注意，本段与 *series-V FPGAs* 及更高版本无关。

如果使用12.0以后的Quartus版本，需要手动编辑 megafunction variation文件。否则，MegaWizard Plug-In Manager 将拒绝接受此文件。

如果需要编辑，请使用文本编辑器打开 pcie_core/ 目录中的文件（例如 pcie_core/pcie_s4_4x.v ）。将旧 Quartus 版本的所有出现替换为使用的版本。只需更改版本号（例如，将 12.0 替换为 15.0）。

通常，与以下类似或类似的行需要修改：

```
// megafunction wizard: %IP Compiler for PCI Express v12.0%
```

和

```
// Retrieval info: <MEGACORE title="IP Compiler for PCI Express"  version="12.0"
```

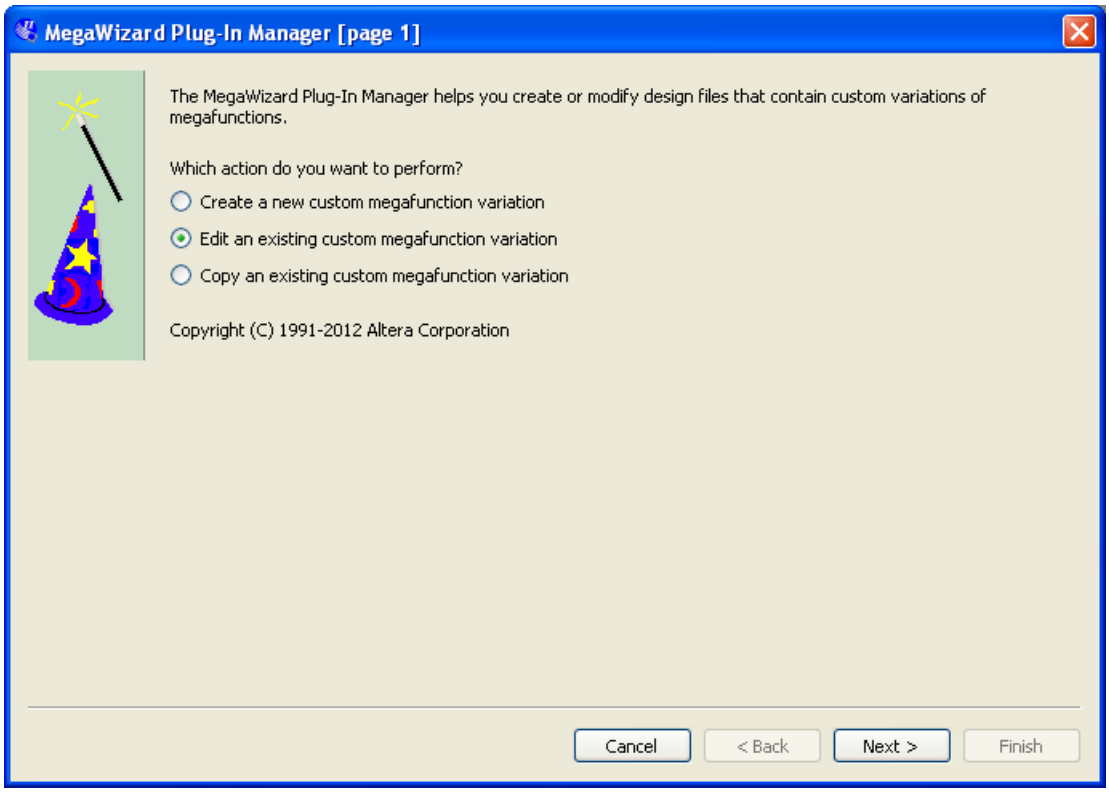
，也可能

```
// Generated using ACDS version 12.0 178
```

从 Quartus 中启动 MegaWizard Plug-In Manager：

- Quartus 14 及更高版本：打开 Command Prompt 窗口（或 Linux 中的 terminal）。确保 Quartus 的实用程序在 execution path（通常是 /some/path/quartus/bin/ ）中。键入 “qmegawiz” 以启动 MegaWizard Plug-In Manager。
- Quartus 13 及更早版本：在“工具”菜单中，选择 MegaWizard Plug-In Manager。

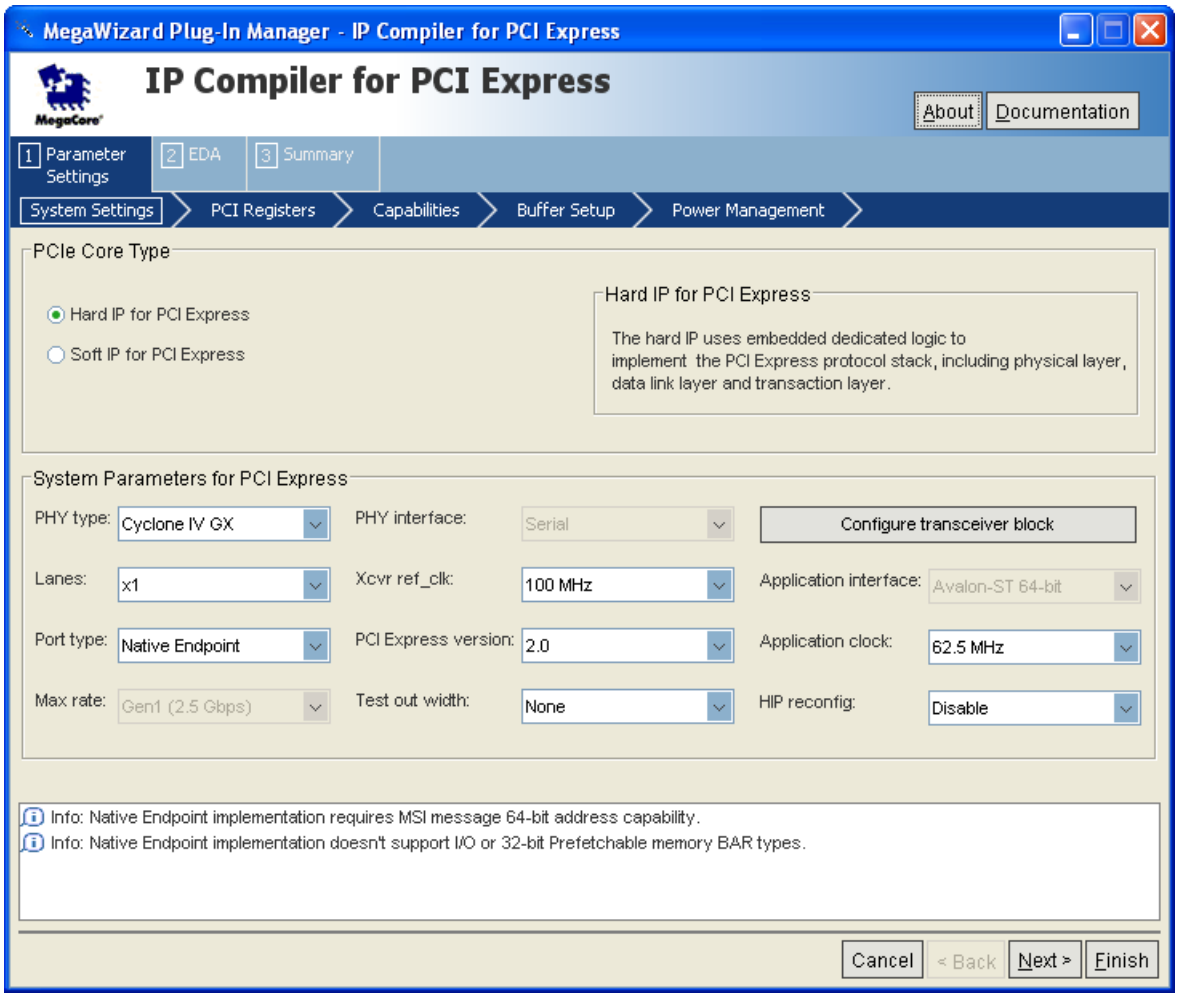
应出现以下（或类似）窗口：



选择 “Edit an existing custom megafunction variation”，然后单击下一步。

接下来，一个窗口（此处未显示）请求编辑自定义 megafunction variation 文件。在 pcie_core 目录中选择 pcie_c4_1x.v（或类似的）文件（只会出现一个合适的文件）。通常，从起点向上导航一个目录会显示要进入的目录。

在出现 “Loading MegaWizard...” 的通知后，会出现如下窗口：

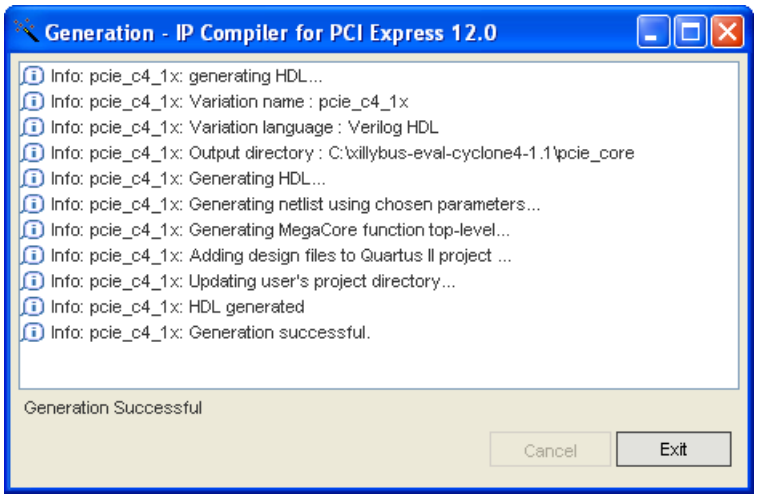


如果 Megawizard 拒绝打开文件，说 “Specify a valid MegaWizard-generated variation file”，这个问题有几种可能：

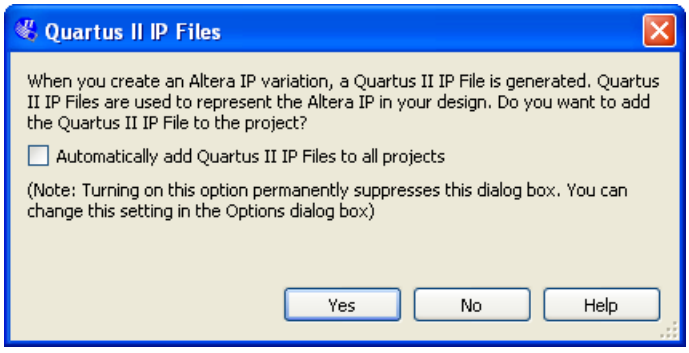
- 您正在使用最新的 FPGA（例如 Cyclone V）。在这种情况下，根本不需要构建 PCIe 块。
- 未正确编辑变体文件以将 Quartus 修订号更改为当前版本。请参阅本节的开头。
- 如果 Megawizard 拒绝一个无效的文件，编辑该文件并尝试再次加载它将无济于事。 Megawizard 程序必须退出并从命令 prompt 再次调用，否则它将继续拒绝该文件，即使它已被正确编辑。

打开的窗口及其呈现的参数因 FPGA 系列而异（特别是与 “IP Compiler for PCI Express” 不同的标题是可以的）。

不应进行任何更改。只需单击 “Finish”。如下所示的窗口显示进度（此处显示其最终状态）：



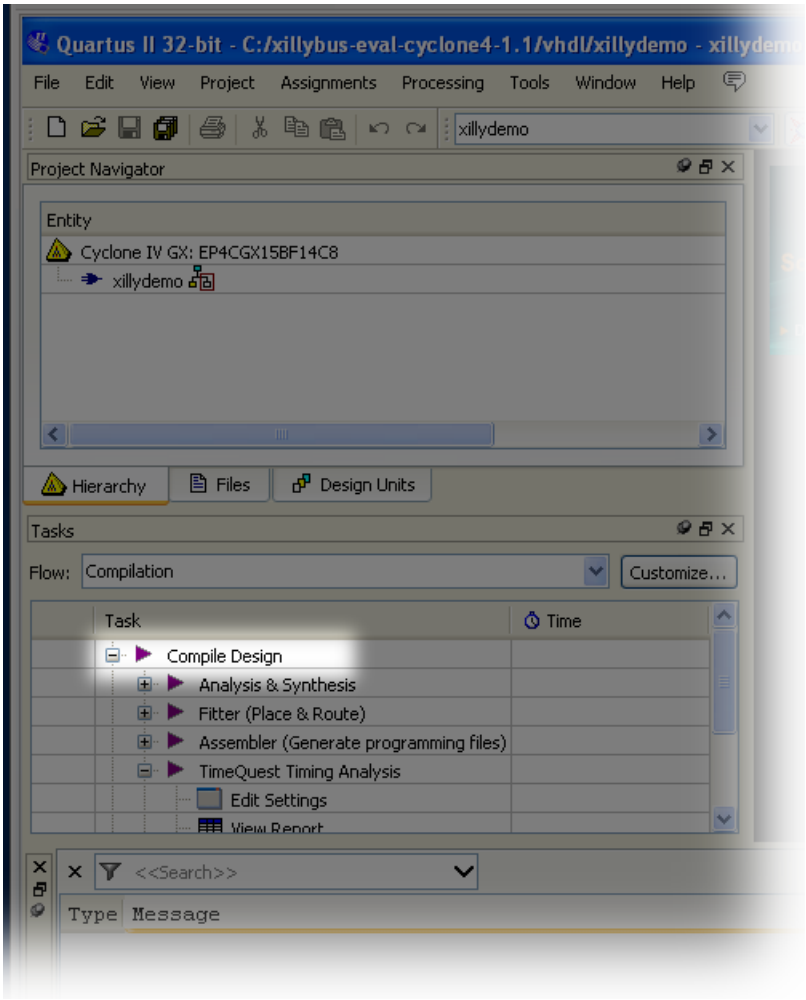
完成后，单击退出。之后可能会立即出现以下窗口，提供将 Quartus IP (QIP) 文件添加到项目中。



首选答案是 “No”，因为将 QIP 文件添加到项目中间接添加了一个不必要的 SDC 文件，其中包含无论如何都被忽略的 constraints。尽管在项目中有这个 QIP 文件是无害的，但它在 compilation 期间会导致很多 warnings，通常在 SDC 文件中每个忽略的 constraint 都会产生两个 warnings。

3.3.3 design 文件的 Compilation

在 Quartus 的主窗口中，单击 “Compile Design” 以创建 FPGA bitstream 文件。请注意，在某些 Quartus 修订版中，compilation 的默认过程可能设置为 “Rapid Recompile”，即任务窗格中仅允许 “Rapid Recompile”。如果是这种情况，请将其更改为 “Compilation”，然后继续。



该过程产生 20-50 个 warnings（取决于 QIP 文件是否包含在项目中），但应以显示 “Full Compilation was successful” 的对话框结束。

必须确认没有生成错误或 Critical Warnings（Quartus 主窗口底部的选项卡指示每种类型的消息数量）。

在该过程结束时，可以找到编程文件为 xillydemo.sof。

3.4 对 FPGA 进行编程

在早期开发阶段，建议通过 JTAG 加载 FPGA。这通常使用 USB Blaster / Intel FPGA Download Cable（板载或外部）完成。请参阅您的主板说明，了解如何通过 JTAG 加载 FPGA。

对于 XillyUSB 项目，FPGA 可以随时加载和重新加载，即使 USB 接口连接到工作计算机也是如此。

对于 PCIe 项目，FPGA 必须在计算机上电前加载 bitfile：计算机期望 PCIe 外设在上电时处于正常状态，之后可能不会容忍任何意外。

因此，只要 host 正在运行，就**不要**重新加载 FPGA。尽管 PCIe 规范要求支持 hotplugging，但主板通常不会期望 PCIe 卡消失然后重新出现。因此，某些主板可能无法正确响应。不过，在操作系统运行时重新加载 FPGA 可以在某些主板上运行。

Xillybus的driver在设计上对hotplugging反应灵敏，但是没有什么可以保证电脑的整体稳定性。这是在这个页面上讨论的：

http://xillybus.com/doc/hot-reconfiguration

如果 FPGA 加电并从 flash memory 加载并同时启动计算机，则必须确保 FPGA 加载足够快，以便在 BIOS 扫描 bus 时存在 PCIe 设备。

4

修改

4.1 与定制 logic 集成

Xillybus demo bundle 的构造便于与 application logic 集成。连接数据的地方是 xillydemo.v 或 xillydemo.vhd 文件（取决于首选语言）。为了使用 Xillybus IP core 在 host（Linux 或 Windows）和 FPGA 之间传输数据，可以忽略捆绑包中的所有其他 HDL 文件。

可以将带有自定义 logic designs 的附加 HDL 文件添加到按 3.3 段所述准备的项目中，然后通过单击 “Compile Design” 重新构建。无需重复初始部署的其他步骤，因此 logic 的开发周期相当快速和简单。

将 Xillybus IP core 连接到自定义 application logic 时，强烈建议仅通过 FIFOs 与 Xillybus IP core 交互，不要试图模仿 logic 的行为，至少在第一阶段不要。

一个例外是将存储器或 register arrays 连接到 Xillybus 时，在这种情况下，应遵循 xillydemo 模块中显示的示例。

在xillydemo模块中，FIFOs被用来执行一个data loopback。换句话说，从 host 到达的数据被发送回它。 FIFO的两边都连接到Xillybus IP core，所以core既是数据的来源，也是数据的消费者。

在实际使用场景中，只有 FIFO 的一侧连接到 Xillybus IP core。 FIFO 的另一端连接到 application logic，它提供或消耗数据。

xillydemo 模块中使用的 FIFOs 仅与两侧的一个公共 clock (scfifo) 一起工作，因为两侧均由 Xillybus 的主 clock 驱动。在实际应用中，可能需要将它们替换为具有单独的 clocks 用于读取和写入的 FIFOs。这允许使用 bus_clk 以外的 clock 驱动数据源和数据消费者。通过这样做，FIFOs 不仅充当调解器，而且还用于正确的 clock domain 交叉。

请注意，对于从 FPGA 到 host 的 streams，Xillybus IP core 需要一个普通的 FIFO 接

口（与 “show-ahead” 相反）。

以下文档与集成自定义 logic 相关：

- API 用于 logic design：Xillybus FPGA designer’s guide
- Getting started with Xillybus on a Linux host
- Getting started with Xillybus on a Windows host
- Xillybus host application programming guide for Linux
- Xillybus host application programming guide for Windows
- The guide to defining a custom Xillybus IP core

4.2 对 Intel 的 PCIe IP / Megafunction 进行更改

除非绝对必要，否则不应更改 Intel IP compiler for PCI Express Megafunction 的配置。

特别是对 receive buffers 的分配有很高的敏感性。 Xillybus IP core 根据 IP Compiler / MegaWizard 在 demo bundle 中给出的数字依赖于那些 buffers 的尺寸。如果 PCIe core 分配的 buffer 空间少于 Xillybus IP core 的预期，则在从 host 到 FPGA 的通信中可能会出现零星的数据错误。这也可能导致此方向上的通信完全停止。此类问题是数据包到达 FPGA 的结果，导致这些 buffers 的 overflow。

如果需要对 IP / Megafunction 进行任何更改，请通过 Xillybus 网站上发布的电子邮件地址寻求帮助。

4.3 包含在自定义项目中

如果需要，可以将 Xillybus IP core 包含在现有 Quartus 项目中，或从头开始创建新项目。本节与 PCIe 的 Xillybus 相关，但类似的过程适用于 XillyUSB。

如果该项目尚不存在，请启动一个新项目，并根据您首选的 HDL 语言和预期的 FPGA 进行设置。

要将 Xillybus IP core 包含在项目中：

- 将 pcie_c4_1x.v 文件（或类似文件）从 pcie_core/ 子目录复制到与项目相关的目录。
- 对于 Cyclone V、Arria V、Stratix V 和 series-10 FPGAs，还要将 pcie_core/pcie_reconfig.qsys（
  pcie_core/pcie_ip.ip）复制到同一目录中。

- 如果所选的 FPGA 需要，请按照 3.3.2 中概述的过程进行操作，以构建 Intel 的 PCIe 块。
- 应该为 Cyclone IV 添加这些文件（可能是它们的副本）：
  - pcie_c4_1x.v
  - pcie_c4_1x_core.v
  - pcie_c4_1x_serdes.v
  - pcie_c4_1x_examples/chaining_dma/pcie_c4_1x_rs_hip.v
  - ip_compiler_for_pci_express-library/altpcie_hip_pipen1b.v
  - ip_compiler_for_pci_express-library/altpcie_reconfig_3cgx.v
  - ip_compiler_for_pci_express-library/altpcie_rs_serdes.v

  对于 Stratix IV 和 Arria II，应添加以下文件：
  - pcie_s4_4x.v
  - pcie_s4_4x_core.v
  - pcie_s4_4x_serdes.v
  - pcie_s4_4x_examples/chaining_dma/pcie_s4_4x_rs_hip.v
  - ip_compiler_for_pci_express-library/altpcie_hip_pipen1b.v
  - ip_compiler_for_pci_express-library/altpcie_reconfig_4sgx.v
  - ip_compiler_for_pci_express-library/altpcie_rs_serdes.v

  对于 series-V FPGAs，不需要复制类似的文件，因为它们是由 Qsys 生成和包含的。
- 将两个 HDL 文件 xillybus.v 和 xillydemo.v(hd) 复制到两个 src/ 子目录之一中（取决于您的语言偏好）并将它们添加到项目中。
- 在任一 src/ 目录中的 SDC 文件中采用 constraints（它们可能需要稍作修改以匹配项目中 top level 信号的名称）。
- 从 core/ 目录复制 xillybus_core.qxp 或 xillybus_core.vqm 并将此文件也添加到项目中。
- 如果 xillydemo 模块不是项目的 top level module，请将其端口连接到 top level。
- 要将 Xillybus IP core 连接到自定义 application logic，请编辑 xillydemo 模块，将现有的 application logic 替换为所需的模块。

4.4 使用其他板

4.4.1 一般的

使用未出现在 demo bundles 列表中的电路板时，需要对捆绑包进行一些细微修改。

4.4.2 设备系列

demo bundle 包含一个 QXP 或 VQM 文件，它是针对某个设备系列的 synthesized 文件。它可以与该系列中的任何设备一起使用。

4.4.3 Pin placements

大多数购买的电路板都有自己的 FPGA design 示例，它显示了如何在该电路板上使用 PCIe 接口。通常最容易在目标板的 QSF 文件中找到相关的引脚分配，并将引脚名称修改为 Xillybus 的 QSF 文件中使用的名称。完成此操作后，可以替换 Xillybus 项目中使用的 QSF 文件中的相关行。

还有四根user_led线，没有任何功能意义。但如果板上有空置的 LEDs，建议连接它们，因为它们提供了一些通信状态指示。 user_led 信号假定为 active low logic（logic '0' 表示 LED 打开）。

4.4.4 仅限 PCIe：Clocking

xillydemo 模块需要这三个 clocks 中的部分或全部作为输入。所需的 clocks 是 xillydemo 模块的输入。

- pcie_refclk– 直接连接到 host 主板提供的 reference clock。这个clock的频率应该是100 MHz。当 host 断电时，以及在 PCIe 规范允许的其他情况下，此 clock 可能无法激活。根据 FPGA 的要求，clock 清洁电路可能存在于主板的 clock 和连接到 FPGA 的电路之间。

  如果向该输入端口提供不同的 clock 频率（例如由 clock 清洁器提供的 125 MHz），则需要相应地更改 IP Compiler 中的 Xcvr_ref_clk 参数以进行 PCI Express 设置。这是在调用相关的 IP Compiler / MegaWizard plug-in manager 时完成的，如 3.3 段所示。

- clk_50——驱动 gigabit transceiver 的 reconfig_clk，并且必须始终处于活动状态。因此它一定不能依赖主板的reference clock。此 clock 允许的频率范围取决于 FPGA 系列（例如 Cyclone IV 上的 37.5 MHz 到 50 MHz）。这个clock的规格可以通过在device handbook中搜索“reconfig_clk”找到相关的FPGA。

- clk_125——驱动 transceiver 所需的始终活动的 clock。这个clock的频率必须是125 MHz，不能依赖主板的reference clock。

重要的:

*pcie_refclk*不能只由板上的一些振荡器驱动。与 *host* 的 *clock* 的任何细微频率差异都会导致通信不可靠或无法将 *FPGA* 检测为 *PCIe* 设备。

在用于 Cyclone IV GX Transceiver Starter Board 的 demo bundle 中，clk_50 和 clk_125 输入由 I/O 引脚直接驱动，这些引脚分别连接到板上 50 MHz 和 125 MHz 的 clock 源。如果没有合适的 clocks 直接来自电路板，则可以使用 PLL 来生成两者，如 Intel 发布的 IP Compiler for PCI Express User Guide 的第 7 节所述。

使用 PLL 时，必须编辑 xillybus.v，将 reconfig_clk_locked 信号连接到 PLL 的锁定指示灯。这已经在 Stratix IV 的 demo bundle 上完成，因为 DE4 板不提供具有 125 MHz 频率的始终活动的 clock。

在 PCI Compiler for PCI Express 文件生成后，pcie_core 子目录中的文件中提供了使用 PLL 的示例。这可以在 pcie_c4_1x_examples/chaining_dma 目录中以 pcie_c4_1x_example_chaining_pipen1b.v 的形式找到（或与 Cyclone IV 以外的系列一起工作时的类似文件）。

4.4.5 使用 XillyUSB

XillyUSB 可用于具有 SFP+ 接口的其他板上。在这种情况下，只需将 design 的 constraints 设置为使用连接到 SFP+ 连接器的 MGT。

该板还应为 MGT 提供低 jitter 的 125 MHz reference clock。尽管 USB 规范中有要求，但不应启用 Spread Spectrum Clocking (SSC)（如果存在此类选项）：如果使用 SSC reference clock，MGT 无法正确锁定接收到的信号。

对于定制板，建议参考 sfp2usb 模块的原理图，因为 SFP+ 连接器的引脚直接连接到 FPGA 的 MGT。可以选择交换 SSRX 线，就像在 sfp2usb 模块上所做的那样。仅在简化 PCB design 时才推荐这样做。

如果需要，也可以交换 SSTX 线。这需要编辑 *_frontend.v 文件，以便反转传输位的极性，从而补偿线对交换。请注意，即使没有进行此编辑，USB 连接也很有可能会正常工作，因为 USB 规范要求 link partners 即使在极性交换的情况下也能正常工作。但是建议不要依赖它。

5

故障排除

5.1 implementation 期间的错误

有时 Intel 的工具版本之间存在细微差别。这可能会导致无法运行用于创建 SOF 文件的 implementation。

如果问题没有很快得到解决，请通过 Xillybus 网站上提供的电子邮件地址寻求帮助。请附上失败进程的输出日志，特别是围绕该工具报告的第一个错误。此外，如果在 design 中进行了自定义更改（即从 demo bundle 转移），请详细说明这些更改。还请说明使用了哪个版本的 Quartus 工具。

5.2 PCIe 硬件问题

正常情况下，host 的 BIOS 和/或操作系统正确检测到 PCIe 卡，host 的 driver 启动成功。

在大多数 PC 计算机上，BIOS 会在 boot 进程的早期简要显示检测到的外围设备列表。当成功检测到 Xillybus 接口时，列表中会出现一个带有 vendor ID 1172 和 device ID EBEB 的外设。

至于操作系统对卡的检测，请参考以下两个文档之一，以适用者为准：

- Getting started with Xillybus on a Linux host
- Getting started with Xillybus on a Windows host

检测卡失败（或电脑的boot进程出现故障）与Xillybus IP core无关，Xillybus IP core依赖Intel的PCIe IP core与bus对接。

首先，建议验证以下内容：

- bitstream 在计算机开机时已经加载到 FPGA 中（或者在开机后不久，根据 PCI-SIG 规范）。
- PCIe 的 pinouts 线，包括 reference clock 是正确的（这可以在 fitter 的报告中验证）。
- 该板为 FPGA 提供正确的 reference clock。

如果没有立即发现问题，建议尝试开发板随附的 PCIe 示例项目。这可能会显示错误的跳线设置和可能有缺陷的硬件。

如果此样本检测到卡，但 Xillybus 未检测到卡，则比较两个 designs 的 pinouts 可能会有所帮助。如果它们相等，下一步就是通过调用相关的 GUI 工具来比较 Intel 的 PCIe cores 的属性。

以下配置元素可能需要调整:

- reference clock的频率。
- base class 和 sub class（不太可能，但如果认为 class 未知，一些相对较旧的 PC 计算机在 boot 进程中失败）。
- 除基地址 register 设置、vendor ID、device ID 和中断设置外，任何其他配置不同的属性，不应更改。

如果问题仍然存在，请通过 Xillybus 网站上提供的电子邮件地址寻求帮助。