XILINX®

XAPP869 (v1.0) 2007 年 10 月 4 日

# 使用用于 PCI Express 设计的集成端点模块实现点到点连接

作者：Sunita Jain 和 Guru Prasanna

## 摘要

本应用指南提供了一个关于如何使用 Virtex™-5 LXT FPGA 中用于 PCI Express® 设计的集成端点模块实现点到点 （FPGA 到 FPGA）高速串行包传输功能的参考设计。

该设计使用了两个用于实现 PCI Express 功能的集成端点模块，其中一个模块 （主控模块）具有配置另一个模块 （从属模块）的功能。数据可以在这两个集成端点模块之间双向 （即在全双工模式下）流动。该设计的用户接口是通过本地链路 (LL) 接口提供的。

该设计能够在 x1、 x2、 x4 和 x8 通道配置下操作。

## 硬件要求

该设计可以在 ML523 （Virtex-5 RocketIO™ 特性化平台）或 ML505 （Virtex-5 RocketIO 评估平台）上进行演示。此外还需要以下硬件：

- JTAG 电缆或平台 USB 电缆
- SMA 连接器电缆
- ML523 板的超级时钟模块

## 软件要求

软件要求包括：

- ISE™ 软件， v9.2i SP2 (9.2.02i) 或更高版本
- Modelsim 6.1e
- ChipScope™ Pro 分析器， v9.1i 或更高版本

## PCI Express 标准

PCI Express 标准是一个高性能的通用互连架构，可用于多种计算和通信平台。它是一种基于包的点到点串行接口，支持每方向每通道 2.5 Gb/s 的原始带宽。其可靠且基于信用度的内置流量控制可消除因接收缓冲器溢出而导致的丢包问题，而重试功能则可确保数据传输的高度可靠性。

## 简介

该参考设计使用 Virtex-5 LXT FPGA 中用于 PCI Express 设计的集成端点模块提供了一个成本低且非常可靠的高速点到点或芯片到芯片连接解决方案。图 1 是该设计的完整框图。

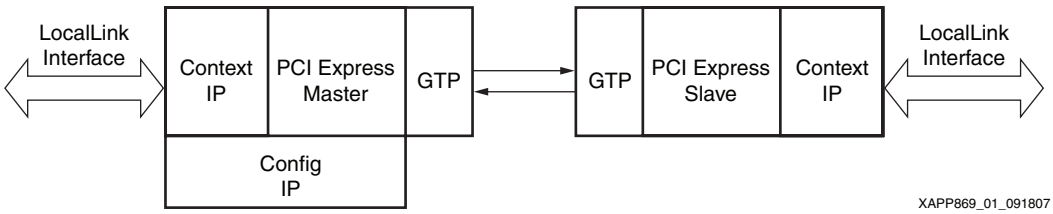


*图 1：* **框图**

© 2007 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

该设计使用了两个集成端点模块 （每端一个）。其中一端作为主控端，可以在本地自行配置。另一端作为从属端，由主控端通过 PCI Express 链路进行配置。进行配置后，该设计便已准备好可以在全双工模式下进行数据传输操作了。

设计的用户接口符合本地链路协议的要求。在传输期间，用户提供的本地链路帧在前往集成端点模块进行传输之前，会先被转换成传输层包 (TLP)。在接收期间，来自集成端点模块的 TLP 会被转换成本地链路帧，而数据以及其他控制信号将通过本地链路接口被传送给用户。

## 设计特性和功能

- 用户接口的本地链路成帧接口
  - 64 位数据总线宽度与 3 位提醒总线
  - 包化接口，带用于标记包的帧起始 (SOF) 和帧结束 (EOF)
- 传输和接收方向用户接口的包中断特性
- 帧错误检测支持
- 多通道配置支持：x1、 x2、 x4 和 x8
- 每通道 1.62 Gb/s 或更高的吞吐量
- 自动初始化、恢复与通道维护，对用户应用是透明的
- Virtex-5 LXT 器件中的集成端点模块
  - 集成端点模块会将三个或六个 Block RAM 用于其内部 TX、 RX 与 RETRY 缓冲器
  - 通过集成端点模块的所有数据传输都是以存储器写 TLP 的形式进行的
  - 利用集成端点模块中的内置包重试功能，可以在链路层进行纠错
- 利用集成端点模块中的自动协商功能，可以在某些通道不能工作时使设计使用较低的带宽继续工作
- Virtex-5 器件中的 GTP 收发器可提供高速收发功能
  - 物理层中使用的 8B/10B 编码和解码
  - 时钟补偿
  - 自动时钟数据恢复
  - 使用多达 8 个收发器
- 配置 IP 块可配置两个集成端点模块
- Context IP 逻辑可执行 TLP 生成 / 解码
- Virtex-5 器件中的内置 FIFO 用于在 FPGA 中进行缓存

## 可配置参数

表 1 列出了生成设计时的用户可配置参数。

*表 1：* **参数列表**

| 参数 | 允许的值 | 说明 |
|---|---|---|
| Board | ML523 （默认）<br>ML505 | 用户可以选择 ML523 （XC5VLX110T-1-FF1136 器件）或 ML505 （XC5VLX50T-1-FF1136 器件）板。ML505 板只提供用于 GTP_X0Y4 (GTP1) 的 SMA 连接器。因此， ML505 板仅支持 x1 通道配置。 |
| Device | XC5VLX110T-1-FF1136 | 用户输入目标器件。必须按左侧所示输入完整的器件名称。 |
| Number of Lanes | 1 | 用户选择设计操作的通道数量。仅在 ML523 上支持多通道设计。 |
| | 2 | |
| | 4 （默认） | |
| | 8 | |
| Silicon Sample Version | PS （默认） | 用户选择芯片版本：PS 表示生产器件， ES 表示工程样本版本。该设计支持将 ES 样本用于与在用户约束文件 (UCF) 中为每个通道宽度指定的 GTP 对应的 XC5VLX110T 器件。有关如何使用 ES 芯片版本的不同 GTP 位置或器件的详细信息，请参阅 “生成设计”。 |
| | ES | |
| Transceiver Locations | 在 UCF 中设置 | 用户可以在 UCF 中设置 GTP 位置。将一个通道用于 ML523 板时，总是会使用 GTP0。目前不支持在 ML523 中为单通道设计选择 GTP1。对于 ML505 板， GTP 位置固定在 GTP_X0Y4 (GTP1) 处。使 GTP 位置远离集成端点模块可能会影响时序。 |
| Design Option | 1 （默认） | 用户根据以下内容选择设计选项：用户可提供的 Block RAM 的数量、要求的吞吐量以及用户帧大小 （以字节为单位）。有关更多详细信息，请参阅表 2。 |
| | 2 | |

表 2 详细介绍了允许的设计选项。

*表 2：* **设计选项详细信息**

| 设计选项 | 使用的 Block RAM 的数量 | 可实现的最高吞吐量 （Gb/s/ 通道） | 说明 |
|---|---|---|---|
| 1 | 3 | 1.62 | 此选项可使大小 ≥ 512 字节的用户帧达到单通道 1.62 Gb/s 的吞吐量，还会将集成端点模块的最大有效载荷大小 (MPS) 设置为 512 字节。 |
| 2 | 6 | 1.74 | 此选项可使大小 ≥ 1024 字节的用户帧达到每通道 1.74 Gb/s 的吞吐量，而大小 ≥ 512 字节的用户帧则可达到每通道 1.72 Gb/s 的吞吐量。设计选项 2 会将 MPS 设置为 1024 字节。 |

## 用户接口

用户接口符合本地链路规范，并支持本地链路中的可选中断信号。表 3、表 4 与表 5 描述了系统接口端口、本地链路传输和本地链路接收端口。

*表 3：* **系统接口端口**

| 端口 | 方向 | 说明 |
|---|---|---|
| RST_N | 输入 | 全局复位 — 复位集成端点模块和 FPGA 逻辑。 |
| GTPRESET_N | 输入 | GTP 复位 — 复位 FPGA 的整个收发器部分。 |
| USRCLK | 输出 | 用户时钟 — 作为输出提供并与 FPGA 逻辑配合使用，x1/x2/x4 设计为 125 MHz， x8 设计为 250 MHz。用户本地链路必须在此时钟下运行。 |
| CORECLK | 输出 | 核心时钟 — 这是集成端点模块中的 250 MHz 核心时钟。 |
| LINK_READY | 输出 | 链路就绪 — 表明链路已准备好可以进行数据传输了。链路建好并完成配置后触发。 |
| LINK_UP | 输出 | 链路建好 — 表明两个集成端点模块之间的链路调训已成功，并且链路已建好。 |
| LINK_WIDTH[3:0] | 输出 | 链路宽度 — 表明协商的链路宽度：<br>0001：单通道<br>0010：双通道<br>0100：四通道<br>1000：八通道 |
| CLOCK_LOCK | 输出 | 时钟锁定 — 表明 PLL（锁相环）已实现时钟锁定。 |
| SYSTEM_ERROR | 输出 | 系统错误 — 此输出表明链路不再可靠，并且系统需要复位。 |
| L0RXMAC_LINK_ERROR[1:0] | 输出 | 链路错误 — 触发 Bit 1 表明接收器出错。触发 Bit 0 表明链路调训出错。 |

*表 4：* **用户接口传输端口**

| 端口 | 方向 | 说明 |
|---|---|---|
| LLUI_TX_DATA[63:0] | 输入 | 传输来自用户的数据 — 将传输来自用户应用的数据。 |
| LLUI_TX_SRC_RDY_N | 输入 | 源就绪 — 触发后表明位于 LLUI_TX_DATA 输入上的数据有效。 |
| LLUI_TX_DST_RDY_N | 输出 | 目的地就绪 — 触发后表明 Context IP 已准备好可以接收来自用户应用的数据了。 |
| LLUI_TX_SOF_N | 输入 | 帧起始 — 目前位于 LLUI_TX_DATA 端口上的数据是新帧的第一个字节。 |
| LLUI_TX_EOF_N | 输入 | 帧结束 — 目前位于 LLUI_TX_DATA 端口上的数据是目前帧的最后一个字节。 |

*表 4：* **用户接口传输端口 （续）**

| 端口 | 方向 | 说明 |
|---|---|---|
| LLUI_TX_REM[2:0] | 输入 | 提醒总线 — 主要用于以字的形式表明帧最后一个字节的位置，随 LLUI_TX_EOF_N 一同触发。<br>以下显示了映射到数据的 REM 区段：<br>000：一个数据字节 [63:56] 有效<br>001：两个数据字节 [63:48] 有效<br>010：三个数据字节 [63:40] 有效<br>011：四个数据字节 [63:32] 有效<br>100：五个数据字节 [63:24] 有效<br>101：六个数据字节 [63:16] 有效<br>110：七个数据字节 [63:8] 有效<br>111：所有数据字节 [63:0] 都有效<br>数据字节是根据 LLUI_TX_DATA[63:0] 定义的。 |
| LLUI_TX_SRC_DSC_N | 输入 | 源中断 — 表明目前正在传输的帧被取消。应随 LLUI_TX_EOF_N、LLUI_TX_SRC_RDY_N 和 LLUI_TX_DST_RDY_N 一同触发，以表明传输结束。 |
| LLUI_TX_DST_DSC_N | 输出 | 目的地中断 — 从系统发出，表明目前帧已中断。系统错误或远端触发复位都可能会导致目的地中断。 |
| FRAME_ERROR | 输出 | 帧错误 — 表明出现了本地链路帧错误。<br>以下情况被视为成帧错误，会导致触发 FRAME_ERROR：<br>• 两个 SOF，没有插入 EOF<br>• 两个 EOF，没有插入 SOF<br>• SRC_DSC_N，没有 EOF |

*表 5：* **用户接口接收端口**

| 端口 | 方向 | 说明 |
|---|---|---|
| LLUI_RX_DATA[63:0] | 输出 | 收到发给用户的数据 — 用户应用收到数据。 |
| LLUI_RX_SRC_RDY_N | 输出 | 源就绪 — 触发后表明 Context IP 已将有效数据置于 LLUI_RX_DATA 上。 |
| LLUI_RX_DST_RDY_N | 输入 | 目的地就绪 — 触发后表明用户应用已准备好可以接收来自 Context IP 的数据了。 |
| LLUI_RX_SOF_N | 输出 | 帧起始 — 目前位于 LLUI_RX_DATA 端口上的数据是新帧的第一个字节。 |
| LLUI_RX_EOF_N | 输出 | 帧结束 — 目前位于 LLUI_RX_DATA 端口上的数据是目前帧的最后一个字节。 |

*表 5：* **用户接口接收端口 （续）**

| 端口 | 方向 | 说明 |
|---|---|---|
| LLUI_RX_REM[2:0] | 输出 | 提醒总线 — 主要用于以字的形式表明帧最后一个字节的位置，随 LLUI_RX_EOF_N 一同触发。<br>以下显示了映射到数据的 REM 区段：<br>000：一个数据字节 [63:56] 有效<br>001：两个数据字节 [63:48] 有效<br>010：三个数据字节 [63:40] 有效<br>011：四个数据字节 [63:32] 有效<br>100：五个数据字节 [63:24] 有效<br>101：六个数据字节 [63:16] 有效<br>110：七个数据字节 [63:8] 有效<br>111：所有数据字节 [63:0] 都有效<br>数据字节是根据 LLUI_RX_DATA[63:0] 定义的。 |
| LLUI_RX_SRC_DSC_N | 输出 | 源中断 — 表明目前正在传输的帧被取消。<br>随 LLUI_RX_EOF_N、LLUI_RX_SRC_RDY_N 和 LLUI_RX_DST_RDY_N 一同触发，以表明传输结束。 |
| LLUI_RX_DST_DSC_N | 输入 | 目的地中断 — 来自用户，表明目前帧被取消。在下一周期 Context IP 会根据本地链路协议触发 LLUI_RX_EOF_N。 |

# 架构

整个设计已被划分为三个主要架构模块：配置 IP、 Context IP 传输与 Context IP 接收。

## 配置 IP

配置 IP 块负责在本地配置主控端，并通过 PCI Express 协议链路配置从属端。配置完成后，会触发 LINK_READY，以表明链路已准备好可以进行数据传输了。配置期间，将在主控端和从属端设置最大有效载荷大小。

配置 IP 会根据选择的用户设计选项设置集成端点模块的最大有效载荷大小 (MPS)，如果是设计选项 1，则设置为 512 字节，如果是设计选项 2，则设置为 1024 字节。

## Context IP 传输

Context IP 传输模块采用存储转发技术原理，通过协议链路以存储器写 TLP 的形式传输用户帧。存储器写 TLP 标头中的 TAG 区段未经过定义，可以包含任何值，因此可用于存储帧起始、帧结束、帧中断等标记，这些标记有助于在收到结束信号时重建本地链路帧。

传输器逻辑可将从用户处传来的数据存储在 FIFO 中。FIFO 的容量等于为选定设计选项设置的 MPS。长度计数器会对有效载荷的长度进行计数，有效载荷的长度用于建立 TLP 标头中的长度区段。计数器在达到最高计数值 （MPS = 512 时为 511， MPS = 1024 时为 1023）后会归零。帧的长度会存储在长度缓冲器内。

Virtex-5 器件中的内置 FIFO 用于进行存储。如果数据传输之间的链路断开 （LINK_UP 信号变为低），则当前存储在缓冲器内的所有数据都将丢失。由于必须重新进行配置，因此可以将数据传输之间的链路断开视为复位。

图 2 为传输逻辑的概图。

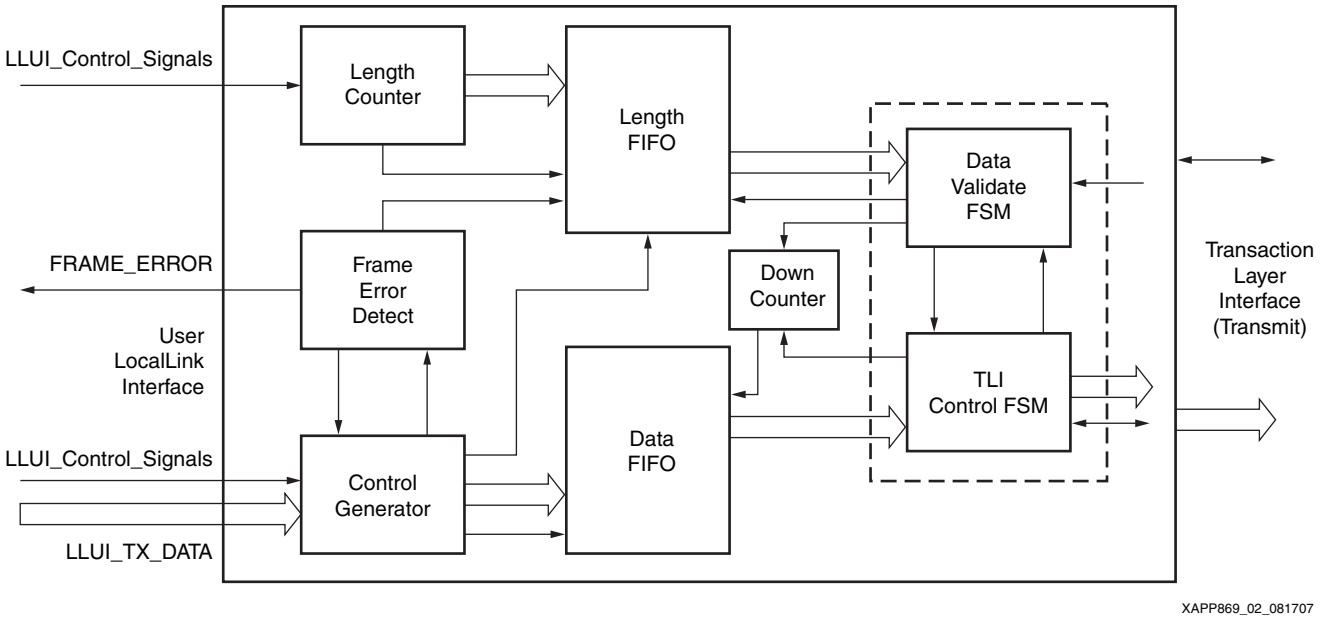


*图 2：* **Context IP 传输框图**

如果用户帧大小小于在集成端点模块中设置的 MPS，整个帧将作为单个 TLP 进行传输，如图 3 所示。前缀为 “llui_” 的信号表示用户本地链路信号，而前缀为 “tli_” 的信号表示位于传输层接口传输侧的信号。

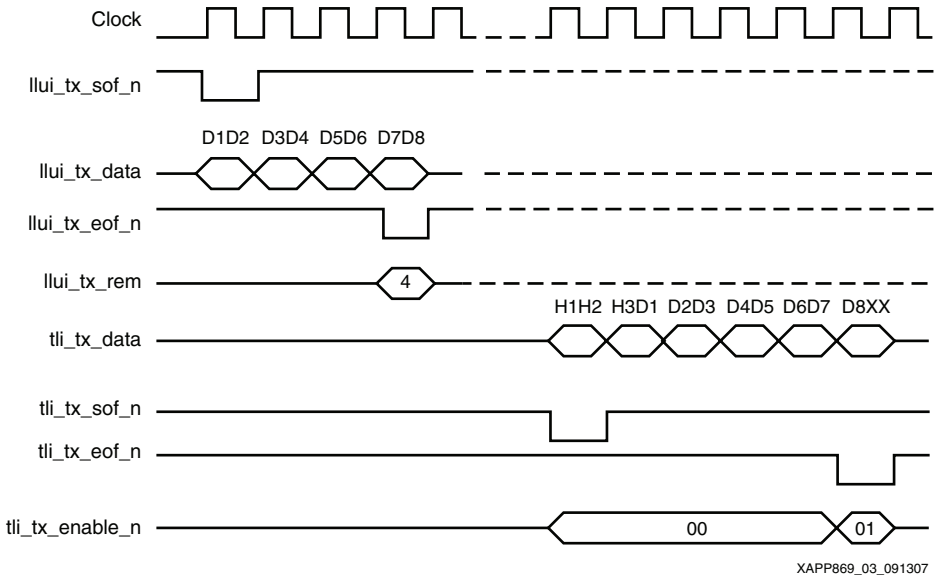


*图 3：* **通过传输层传输接口传输帧**

通过解除 LLUI_TX_SRC_RDY_N 输入， Context IP 即会准备好可以接收帧了，此时用户可以选择不传输帧。

## 用户帧大小大于 MPS 时的分段方案

当用户帧大小大于 MPS 时，在传输帧之前会先将其分段，分段后每段的大小等于 MPS 字节数。例如，在设计选项 1 中， MPS 等于 512 字节。如果用户帧大小大于 512 字节，则数据 FIFO 从用户处收到 512 字节后会变满，同时 LLUI_TX_DST_RDY_N 信号会解除。长度计数器达到终止计数点 （在此情况下为 511）时会触发一个给长度 FIFO 的写信号。缓存在此计数点之前的用户段将作为一个 TLP （TLP 长度 = 512 字节）通过传输层接口 (TLI) 进行发送。缓冲器中的数据通过 TLI 传输后，会触发 LLUI_TX_DST_RDY_N 信号，这表明数据 FIFO 已准备好 可以接收来自用户的更多数据了。因此，当用户帧大小大于 MPS 时，数据会以多个 TLP 的形式通过 TLI 进行传输。

## 中断特性行为

通过在传输端触发 LLUI_TX_SRC_DSC_N 信号，用户可以中断本地链路帧。触发该信号时应同时触发 LLUI_TX_EOF_N，如图 4 所示。如果未触发 LLUI_TX_EOF_N, FRAME_ERROR，帧会中断。

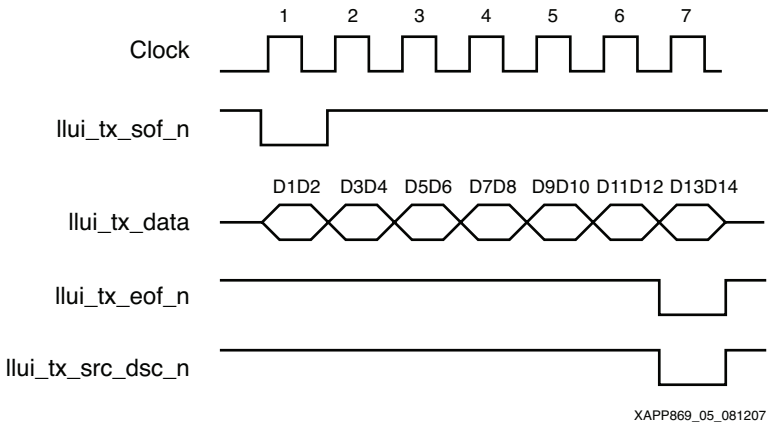


*图 4：* **在本地链路传输时中断**

以下针对各种情况对用户的中断情况进行了说明。

### 用户帧小于或等于 MPS：已中断

当小于或等于 MPS 的用户帧已中断时， Context IP 传输逻辑会丢弃与该帧对应的数据。在此情况下不会传输任何 TLP。

### 用户帧大于 MPS：已中断

#### *在第二段期间 （在第一个 512 字节组之后）收到源中断*

在此情况下， MPS 等于 512 字节，并且用户会在发送 600 字节后中断该帧。 Context IP 会缓存前 512 字节，并针对第一段启动通过 TLI 进行 TLP 传输。通过 TLI 传输第一段的数据时，下一段的数据会缓存在数据 FIFO 中。当用户在第二段期间触发 LLUI_TX_SRC_DSC_N 时，正在通过 TLI 传输的 TLP 将被中断，并且两个 FIFO 都会被清空。在 FIFO 清空期间，LLUI_TX_DST_RDY_N 会被解除。在此情况下不会传输任何 TLP。

### *通过 TLI 传输某段后收到源中断*

在此中断情况下，用户可以在来自用户的段已通过 TLI 发送后触发 LLUI_TX_SRC_DSC_N。例如，用户可以在第三段期间触发 LLUI_TX_SRC_DSC_N。这表明第一段已通过 TLI 发送，来自用户的第二段正在数据缓冲器中，并且正在通过 TLI 进行传输。缓存第三段期间，如果用户中断该帧，则通过 TLI 传输第二段会被中断，并且 FIFO 会被 Context IP 逻辑清空。此外，还会通过 TLI 发送一个双字 (DW) TLP （一个长度等于 1DW [32 位 ] 的 TLP）以及 TAG 区段中的中断标记位集。该 TLP 会告知接收端先前收到 TLP 的帧已中断。在接收端，接收 1-DW TLP 以及中断标记位集会被翻译为 LLUI_RX_SRC_DSC_N，如图 5 所示。

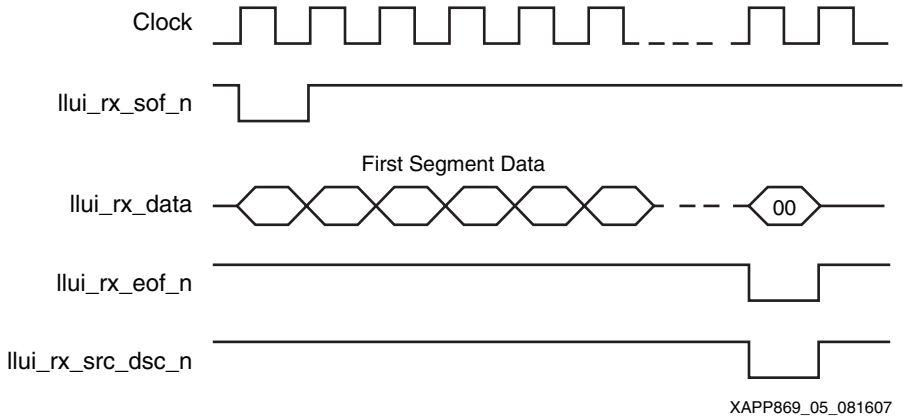


*图 5：* **在本地链路接收时中断**

因此，通过触发 LLUI_RX_SRC_DSC_N，可中断属于已中断帧且通过 TLI 传输的段。用户应丢弃收到 LLUI_RX_SRC_DSC_N 的帧。

## Context IP 接收

该模块可接收来自传输层接收接口的 TLP，然后提取其中的有效载荷。从 TLP 的 TAG 区段提取的标记位将被写入到接收端 FIFO 的数据奇偶位输入区段。

提取的数据以及相应的本地链路成帧信号会被发送给接收端的用户。请参阅图 6。

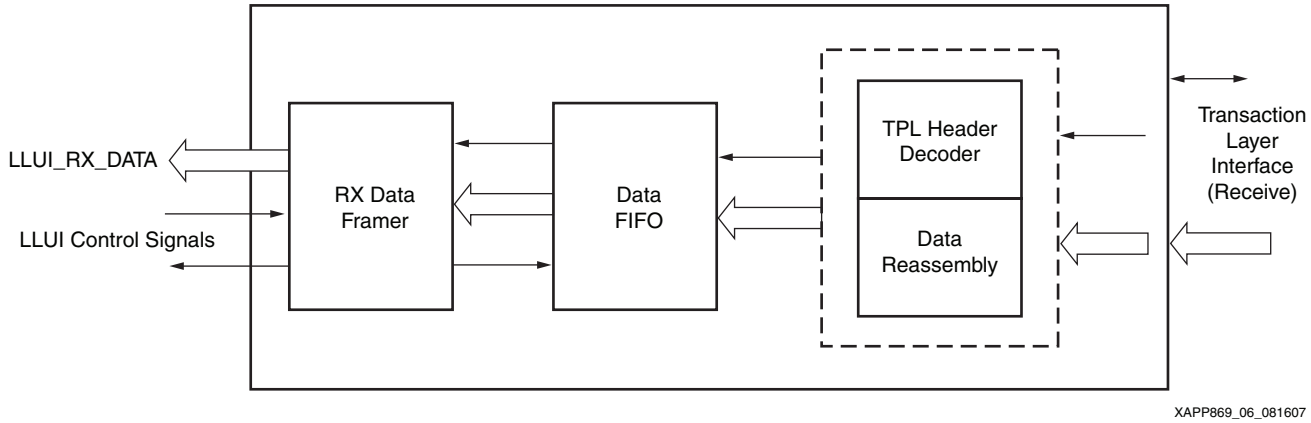


*图 6：* **Context IP 接收模块**

图 7 显示了在 TLI 接收接口收到的 TLP （带 “tli_rx*” 的信号）如何翻译成用户接收接口处的本地链路帧 （带 “llui_rx*” 的信号）。

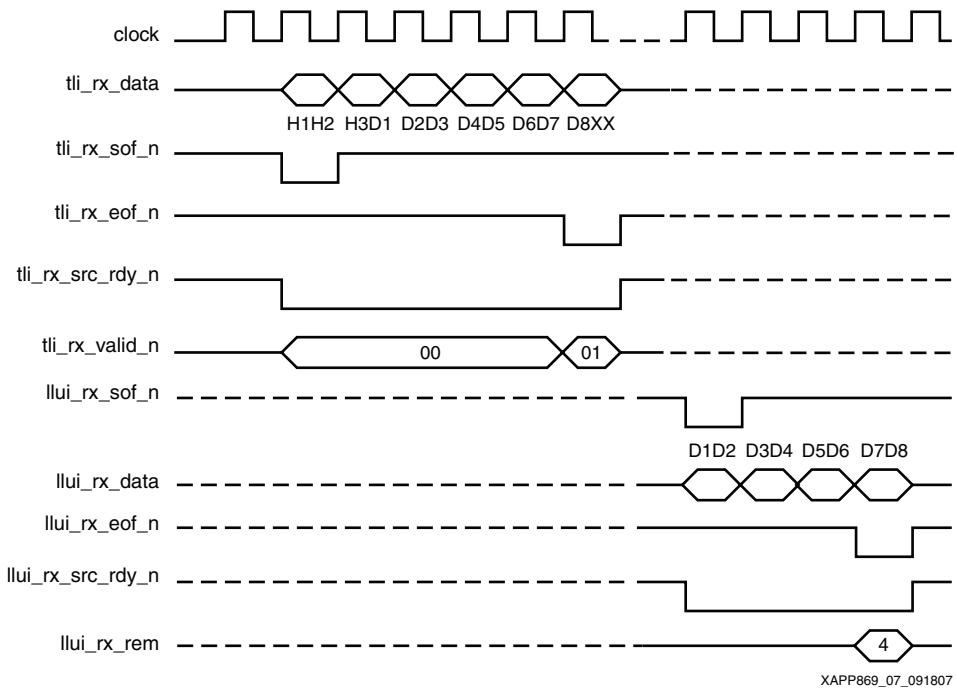


*图 7：* **通过传输层接收接口传输帧**

## 接收端的目的地中断

接收端的用户也可以通过触发 LLUI_RX_DST_DSC_N 中断帧。在下一周期 Context IP 接收逻辑将根据本地链路协议触发 LLUI_RX_EOF_N，并且已中断帧内的其他数据会被丢弃。请参阅图 8。

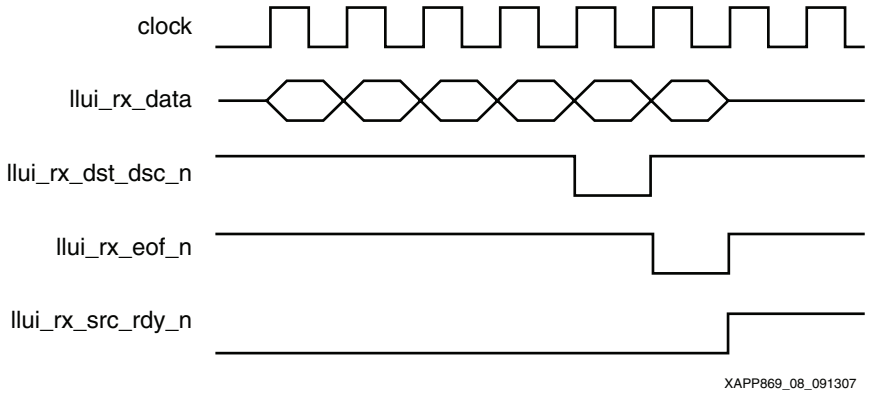


*图 8：* **LLUI_RX_DST_DSC_N 行为**

## 自动协商

利用自动协商，可以在较高的通道宽度设计出现链路电缆断开时使设计在较低的通道配置中恢复。

这意味着 x8 设计能够在 x1、x2 或 x4 配置中恢复，而 x4 设计能够在 x1 或 x2 模式中恢复，具体取决于断开的电缆属于哪个通道。

利用自动协商功能，可以在某些通道不能工作时使设计使用较低的带宽继续工作。

## 时钟要求

用于 PCI Express 设计的集成端点模块工作时需要的内核时钟频率和线速分别为 250 MHz 和 2.5 Gb/s。

集成端点模块还会提供一个时钟，以便针对各种通道通过 TLI 传输数据。所有 x8 设计必须在 250 MHz 的时钟下运行 TLI，才能维持全带宽。对于 x1、 x2、 x4 通道配置， TLI 可以在 125 MHz 的较低频率下运行。

对于 x1、x2、x4 配置，此参考设计会在 125 MHz 的 USRCLK 下运行 TLI；对于 x8 配置，则会在 250 MHz 的 USRCLK 下运行 TLI。同一时钟 (USRCLK) 会作为输出信号提供。用户需要在提供的 USRCLK 下运行本地链路。

要使设计正常工作，用户需要输入一个 100 MHz 的差分参考时钟。必须根据 UCF 将差分时钟输入与 GTP 时钟输入相连。

## 设计范例

该设计的数据传输功能将通过一个设计范例进行说明。本地链路帧生成器和帧检查器用于生成流量和检查收到的数据。

帧生成器具有定性特征，可为本地链路用户帧提供渐增的有效载荷大小和各种提醒值。帧检查器会检查收到的数据，并在发现收到的数据有错误时标记错误输出。请参阅图 9。

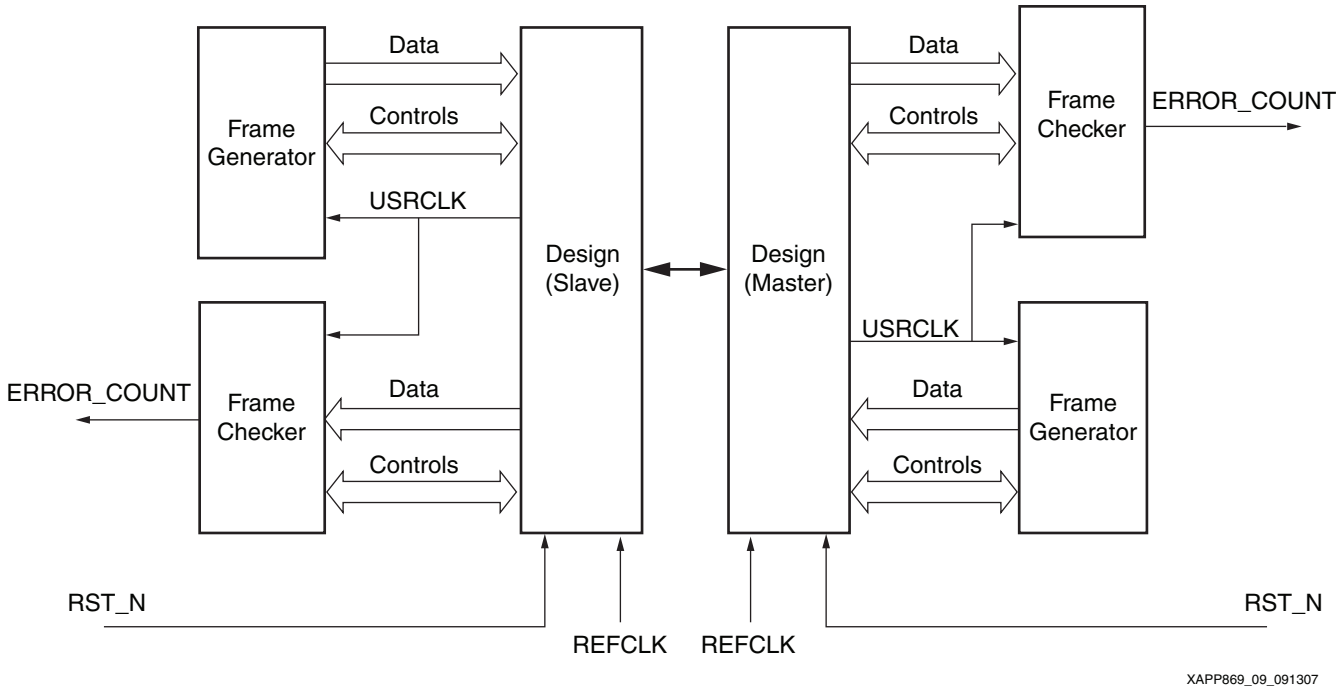


*图 9：* **测试数据传输设计范例**

## 设计目录结构

图 10 显示了此参考设计的设计目录结构。

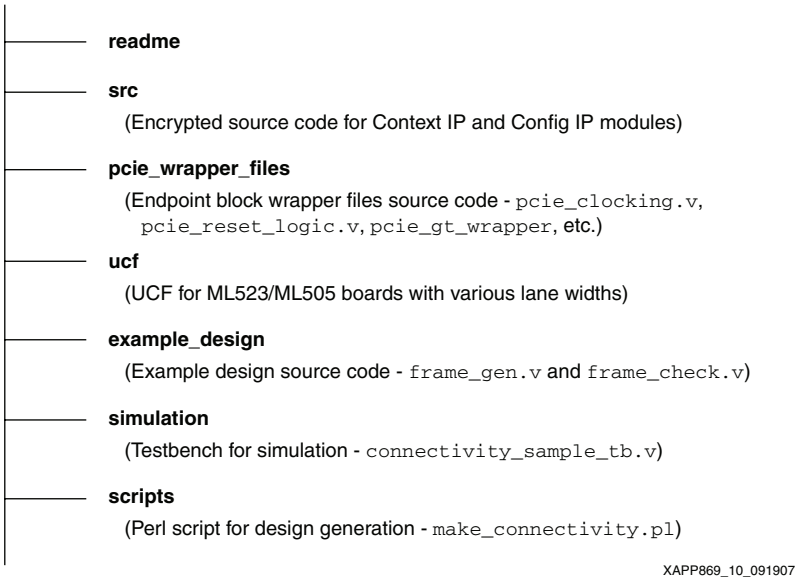


*图 10：* **设计目录结构**

图 11 显示了设计生成后的设计目录结构 （有关详细信息，请参阅 “生成设计”）。它显示生成的新文件。

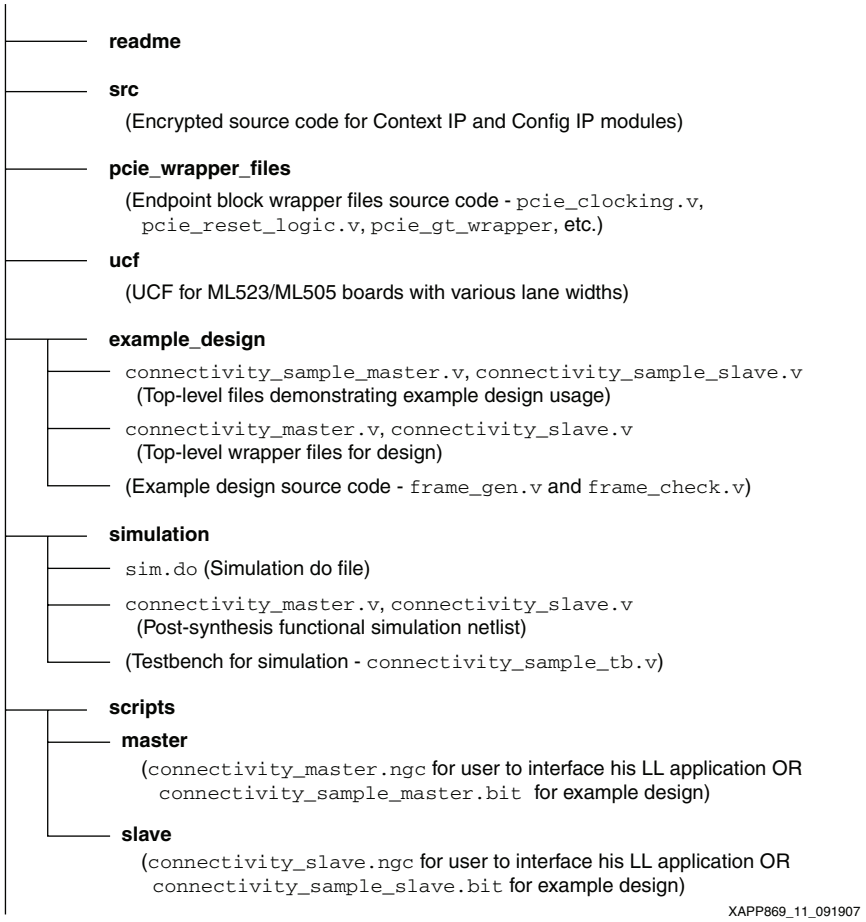


*图 11：* **设计生成后的设计目录结构**

## 生成设计

设计源文件是以加密方式进行提交的。集成端点模块所需的封装文件 （例如时钟模块、复位逻辑、 GT 封装等）是作为源代码进行提交的。设计随附的 perl 脚本会根据选择的参数为用户生成综合后网表。 perl 脚本的使用方法如下所示 （它需要根据用户安装的 ISE 软件设置 Xilinx 环境）：

```
xilperl make_connectivity.pl -design_option=2 -lanes=2 -board=ML523 -example -ps
```

板、设计选项、通道数量等参数可通过命令行接口输入。

perl 脚本参数：

- design_option：设置为 1 时，此参数会将集成端点模块的 MPS 值设置为 512 字节，这会将三个 Block RAM 用于 PCI Express 协议中的各种传输、接收和重试缓冲器。设置为 2 时，此参数会将 MPS 设置为 1024 字节，并使用六个 Block RAM。此参数的默认值设置为 1。
- lanes：此参数用于选择设计的通道数量。允许的值为 1、 2、 4 和 8。此参数的默认值设置为 4。
- board：此参数用于选择用于实现的目标板。允许的选项为 ML505 （仅支持 x1）和 ML523 （支持所有通道配置）。默认值设置为支持 ML523 板。
- example：此切换器用于为 ChipScope Pro 分析器随附的设计范例生成比特流，并用于观察本地链路帧。会生成两个比特流：一个用于主控端，另一个用于从属端。
- device：此参数用于目标器件。默认值设置为 XC5VLX110T-1-FF1136 （对于 ML523 ）和 XC5VLX50T-1-FF1136 （对于 ML505）。
- es/ps：此切换器用于选择工程样本 (ES) 芯片版本或生产器件。 ES 支持是针对位于以下位置的 XC5VLX110T 器件提供的：在 UCF 中针对各种通道配置设置的 GTP 位置。

要针对 ES 版本的不同 GTP 位置或 ES 版本的不同器件创建设计，用户必须针对特定 GTP 位置或器件生成 pcie_gt_wrapper.v 文件，并且不得使用参考设计提供的 pcie_gt_wrapper.v 文件。需要执行以下步骤：

1. 使用作为 CORE Generator™ 软件 (IP) 提供的 LogiCORE™ IP *Endpoint Block Wrapper for PCI Express 生成包含所需 GTP 位置和器件的设计*。
2. 使用生成的 IP 中的 pcie_gt_wrapper.v 文件。
3. 打开 make_connectivity.pl 脚本，并将变量 “$gt_wrapper” 的路径和文件名设置为前面生成的 pcie_gt_wrapper.v 文件。现在，该脚本会使用用户设置的 pcie_gt_wrapper 文件。
4. 输入*完整的器件名称*，并更改 UCF 以反映准确的 GTP 位置。运行脚本并生成设计。

如果需要，用户可以按步骤 1–4 中所述，从用于 PCI Express 设计的端点模块封装中获取最新的 pcie_gt_wrapper.v 文件 （以便在 GTP 属性改变时用于生产器件），并包括该文件而非参考设计随附的 gt_wrapper 文件。

用户还可以使用用于 PCI Express 设计的端点模块封装中的复位逻辑和时钟模块，而非使用在 pcie_wrapper_files 文件夹中提供的模块。为此，用户必须在 make_connectivity.pl 脚本中更新这些文件的路径。

不过，建议用户不要修改封装文件中的任何参数值。

如果没有 -example 选项，会使用给定选项在以下位置生成设计网表 （connectivity_master.ngc 和 connectivity_slave.ngc）：scripts 目录中的 master 和 slave 文件夹内。为了方便在用户应用中使用，该设计针对顶模块提供了两个封装文件。文件名分别为 connectivity_master.v 和 connectivity_slave.v，位于 example_design 文件夹内。用户可以使用这两个文件，并可将其应用连接至提供的本地链路端口。

此外，还针对 ML505 （XC5VLX50T-1-FF1136 器件）和 ML523 （XC5VLX110T-1-FF1136 器件）板提供了 UCF。

`perl` 脚本还可以生成综合后功能仿真 `Verilog` 文件。

要运行仿真，必须使用在 `simulation` 目录中设置的 Modelsim (v6.1e) 环境运行以下内容：

```
vsim -do sim.do &
```

## 资源利用量

表 6 和表 7 显示了主控和从属设计的资源利用量。

这些设计已在带 XC5VLX110T 器件的 ML523 板和带 XC5VLX50T 器件的 ML505 板上进行测试。

*表 6：* **设计的资源利用量**

| 通道数量 | 主控资源利用量 LUT/FIFO | 从属资源利用量 LUT/FIFO |
|---|---|---|
| 1 | 589/656 | 551/661 |
| 2 | 591/691 | 553/728 |
| 4 | 596/761 | 558/862 |
| 8 | 608/901 | 569/1130 |

*表 7：* **包含设计范例的资源利用量**

| 通道数量 | 主控资源利用量 LUT/FIFO | 从属资源利用量 LUT/FIFO |
|---|---|---|
| 1 | 672/831 | 636/837 |
| 2 | 674/866 | 638/904 |
| 4 | 679/936 | 643/1038 |
| 8 | 716/1074 | 680/1304 |

设计利用的其他资源包括：

- 用于 PCI Express 设计的集成端点模块 （每端一个）
- PLL （每端一个）
- Block RAM
  - 对于设计选项 1，每端 3 个
  - 对于设计选项 2，每端 6 个
- Virtex-5 器件中的内置 FIFO
  - 每端 3 个
    - 2 个，大小为 512 x 72 (FIFO36_72)
    - 1 个，大小为 512 x 36 (FIFO18_36)
- 收发器 (1–8 个 )，具体取决于用户选择的通道数量

# 性能

## 资源利用量与通道宽度

图 12 显示了资源利用量与通道宽度之间的关系。随着通道宽度增加，资源利用量仅略有增加。

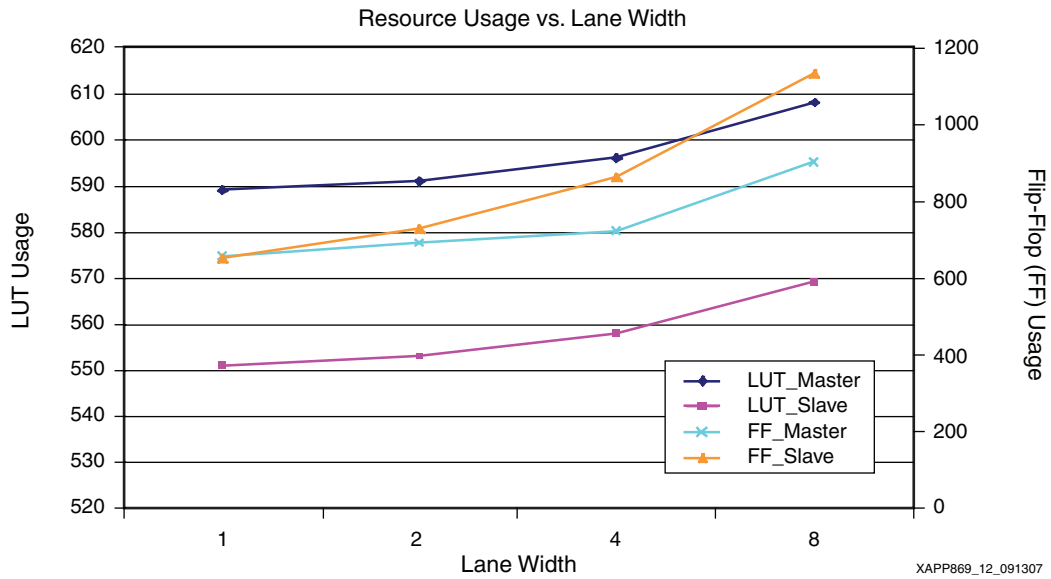


*图 12：* **资源利用量与通道宽度**

## 吞吐量与帧大小

图 13 显示了使用不同 Block RAM 时吞吐量变化情况 （8B/10B 以上除外）与帧大小 （以字节为单位）之间的关系。

选择设计选项 1 会将集成端点模块的 MPS 设置为 512 字节，这会将三个 Block RAM 用于 PCI Express 协议实现中的各种传输、接收和重试缓冲器。选择设计选项 2 会将 MPS 设置为 1024 字节，这会使用六个 Block RAM。

吞吐量结果是针对相同的情况进行描述的。

对于设计选项 1，任何大小大于或等于 512 字节的用户帧都会提供每通道 1.62 Gb/s 的恒定吞吐量。

对于设计选项 2，任何大小大于或等于 1024 字节的用户帧都会提供每通道 1.74 Gb/s 的恒定吞吐量。在此情况下，大小为 512 字节的帧可提供每通道 1.72 Gb/s 的吞吐量。

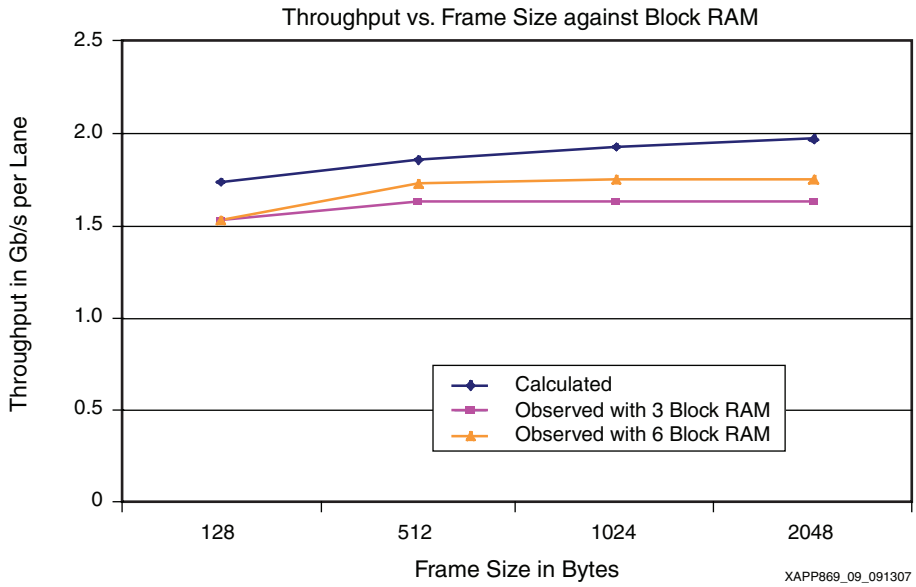


*图 13：* **吞吐量与帧大小**

## 参考设计

该参考设计可在 Xilinx 网站上找到，网址为：
http://www.xilinx.com/cn/bvdocs/appnotes/xapp869.zip

## 结论

此参考设计展示了使用用于 PCI Express 设计的集成端点模块实现数据传输的基本方法。

为了实现具有高度数据可靠性且成本非常低的高速连接解决方案，该设计使用了两个集成端点模块 （每端一个）。

集成端点模块中的内置包重试功能可确保在链路层纠错，从而可确保高度的数据可靠性。该设计使用最少的资源 （约 600 LUT）实现了 FPGA 中的分段和重组逻辑。

利用 PCI Express 协议中的自动协商功能，可以在某些通道不能工作时使设计使用较低的带宽继续工作。

该设计的用户接口是通过 64 位的本地链路成帧接口提供的。该参考设计还支持额外的本地链路特性，即 “中断”。随着通道数量增加，资源利用量仅略有增加。

## 参考内容

1. UG197，*《用于 PCI Express 设计的 Virtex-5 集成端点模块用户指南》*
2. UG196，*《Virtex-5 RocketIO GTP 收发器用户指南》*
3. UG350，*《用于 PCI Express 设计的 Virtex-5 LogiCORE 端点模块用户指南》*
4. UG190，*《Virtex-5 用户指南》*
5. SP006，*《本地链路接口规范 v2.0》*

## 修订历史

下表说明此技术文档的修订历史：

| 日期 | 版本 | 修订说明 |
|---|---|---|
| 10/04/07 | 1.0 | Xilinx 最初版本。 |

## Notice of Disclaimer

Xilinx is disclosing this Application Note to you “AS-IS” with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.