

# Versal ACAP Transceivers Wizard v1.1

## LogiCORE IP 产品指南

Vivado Design Suite

PG331 (v1.1) 2022 年 10 月 19 日

本文档为英语文档的翻译版本，若译文与英语原文存在歧义、差异、不一致或冲突，概以英语文档为准。译文可能并未反映最新英语版本的内容，故仅供参考，请参阅最新版本的英语文档获取最新信息。

赛灵思矢志不渝地为员工、客户与合作伙伴打造有归属感的包容性环境。为此，我们正从产品和相关宣传资料中删除非包容性语言。我们已发起内部倡议，以删除任何排斥性语言或者可能固化历史偏见的语言，包括我们的软件和 IP 中嵌入的术语。虽然在此期间，您仍可能在我们的旧产品中发现非包容性语言，但请确信，我们正致力于践行革新使命以期与不断演变的行业标准保持一致。如需了解更多信息，请参阅此[链接](#)。



## 目录

|                                    |    |
|------------------------------------|----|
| 第 1 章：简介.....                      | 3  |
| 功能特性.....                          | 3  |
| IP 相关信息.....                       | 4  |
| 第 2 章：概述.....                      | 5  |
| 按设计进程浏览内容.....                     | 5  |
| 许可和订购.....                         | 5  |
| 第 3 章：产品规格.....                    | 6  |
| 复位控制器帮助程序块.....                    | 11 |
| 第 4 章：设计流程步骤.....                  | 17 |
| 自定义和生成核.....                       | 17 |
| 共享多个 Bridge IP 的 GT 四通道.....       | 24 |
| 赛灵思 IP - GT 四通道集成.....             | 28 |
| 串行 I/O 调试.....                     | 29 |
| 设计的综合和实现.....                      | 29 |
| 第 5 章：设计示例.....                    | 31 |
| 设计示例的限制.....                       | 32 |
| 设计示例仿真.....                        | 32 |
| 附录 A：调试.....                       | 34 |
| 在 Xilinx.com 上寻求帮助.....            | 34 |
| 调试工具.....                          | 35 |
| 附录 B：附加资源与法律声明.....                | 36 |
| 赛灵思资源.....                         | 36 |
| Documentation Navigator 与设计中心..... | 36 |
| 参考资料.....                          | 36 |
| 修订历史.....                          | 37 |
| 请阅读：重要法律声明.....                    | 37 |

## 简介

Versal® ACAP GTY, GTYP, and GTM Transceivers Wizard IP 解决方案有助于配置一个或多个串行收发器。您可从头开始输入自己的要求并生成有效的配置。此外，此向导还有助于生成设计示例，用于进行简单仿真。

---

## 功能特性

- 简单且直观的功能特性选择流程。
- 一次性 GUI 输入即可启用多线速率选项，无需 GT 寄存器重新编程即可实现无缝动态速率切换。
- 设计输入是通过针对定制 IP 的 IP integrator 来完成的。
- 可综合的设计示例包含可配置的伪随机二进制序列 (PRBS) 数据生成器、检查器和链接状态指示器逻辑，用于快速展示仿真中的核与收发器功能。

## IP 相关信息

| LogiCORE™ IP 相关信息表        |   |
|---------------------------|---|
| <b>核规格</b>                |   |
| 支持的器件系列                   | Versal® ACAP  |
| 支持的用户接口                   | 不适用   |
| <b>随核提供</b>               |   |
| 设计文件                      | RTL   |
| 设计示例                      | Verilog   |
| 测试激励文件                    | Verilog   |
| 约束文件                      | 赛灵思设计约束 (XDC)   |
| 仿真模型                      | 含 SecureIP 收发器仿真的源 HDL                                |
| 支持的软件驱动程序                 | 不适用   |
| <b>经过测试的设计流程</b>          |   |
| 设计输入                      | Vivado® Design Suite                                  |
| 仿真                        | 如需了解受支持的仿真器的相关信息，请参阅 <a href="#">赛灵思设计工具：版本说明指南</a> 。 |
| 综合                        | Vivado 综合   |
| <b>支持</b>                 |   |
| 版本说明和已知问题                 | 主答复记录： <a href="#">75716</a>                          |
| 所有 Vivado IP 更改日志         | Vivado IP 主更改日志： <a href="#">72775</a>                |
| <a href="#">赛灵思技术支持网页</a> |   |

**注释：**

1. 如需获取受支持的器件的完整列表，请参阅 Vivado® IP 目录。
2. 如需了解受支持的工具版本的相关信息，请参阅[赛灵思设计工具：版本说明指南](#)。

# 概述

---

## 按设计进程浏览内容

赛灵思文档按一组标准设计进程进行组织，以便帮助您查找当前开发任务相关的内容。所有 Versal® ACAP 设计进程的对应[设计中心](#)和[设计流程助手](#)资料均可在 [Xilinx.com](#) 网站上找到。本文档涵盖了以下设计进程：

- 硬件、IP 和平台开发：为硬件平台创建 PL IP 块、创建 PL 内核、功能仿真以及评估 Vivado® 时序收敛、资源使用情况和功耗收敛。还涉及为系统集成开发硬件平台。本文档中适用于此设计进程的主题包括：
  - [自定义和生成核](#)
  - [第 5 章：设计示例](#)

## 许可和订购

---

根据[赛灵思最终用户许可条款](#)，此赛灵思 LogiCORE™ IP 模块随附于赛灵思 Vivado® Design Suite 免费提供。

**注释：**要验证是否需要许可证，请查看 IP 目录的“许可证”列。“已含”意味着许可证随附 Vivado® Design Suite 提供；“购买”意味着必须购买许可才能使用核。

有关其它赛灵思 LogiCORE™ IP 模块的信息，请访问[赛灵思知识产权](#)页面。如需获取有关其它赛灵思 LogiCORE IP 模块和工具的定价和可用性信息，请联系您[当地的赛灵思销售代表](#)。

## 产品规格

Versal® ACAP Transceivers Wizard (Versal ACAP 收发器向导) 解决方案包含 2 个核:

- Versal ACAP Transceivers Wizard: 围绕 GT\*\_QUAD 原语的封装文件。其中包含单个 GT 四通道 (gt\_quad\_base IP)。对于多通道 (>4 条通道) 设计, 将例化多个 Transceivers Wizard。
  - GT Quad base IP: 在 Vivado® IP 目录中包含 GT Quad base IP, 它可作为 Versal ACAP Transceivers Wizard (gt\_quad\_base) 以供使用。它用于例化、配置和连接 GT\*\_QUAD 原语。其中提供了仿真支持, 以展示各种 GT 四通道功能特性。您可通过配置 GT Quad base IP 来共享多个协议 IP, 每个协议 IP 都支持多种线速率。您可通过创建设计示例并仿真来了解多个协议 IP 之间共享的动态线速率切换和 GT 四通道。如需获取此 IP 的端口列表和定义, 请参阅《Versal ACAP GTY 和 GTPY 收发器架构手册》(AM002) 和《Versal ACAP GTM 收发器架构手册》(AM017)。
- Versal ACAP Transceivers Bridge: 参考父级 IP (Bridge IP), 用于配置 Transceivers Wizard 参数。
  - Bridge IP: 在 IP integrator 画布中包含 Versal ACAP Transceivers Bridge IP (gt\_bridge\_ip) 以供使用。通过每个 Bridge IP 只能创建一个定制设计输入。Bridge IP 是可配置的封装文件, 通过此文件即可配置 GT Quad base IP (gt\_quad\_base)。

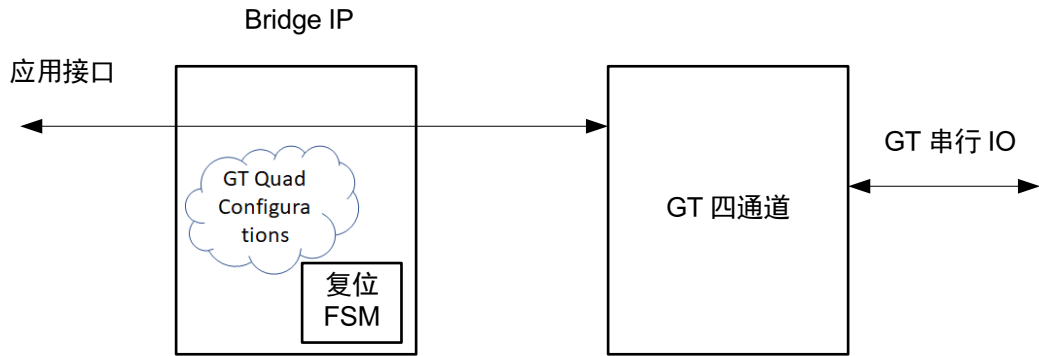
Bridge IP 具有两种工作模式:

“Default Mode” (默认模式): 此模式用于在 FPGA 开发板上进行独立收发器向导仿真和确认。在此模式下, Bridge IP 会基于已编程的配置来封装数据生成器、检查器逻辑和复位逻辑。

“Pass Through Mode” (直通模式): 将 Bridge IP 配置为直通模式时, 它会向应用接口公开 GT 四通道的所有相关信号。在配置所有 GT 参数时, 它会例化、配置和连接单个或多个 GT Quad base IP (gt\_quad\_base)。

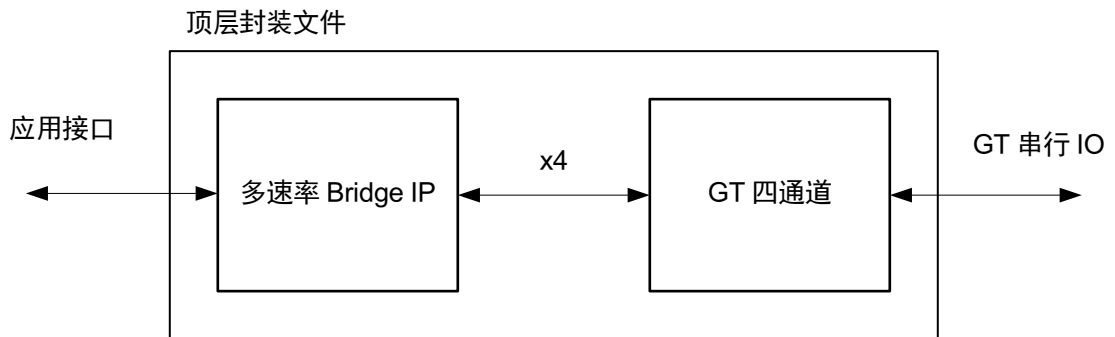
您可在 IP integrator 中添加 Versal ACAP Transceiver Bridge IP (gt\_bridge\_ip), 并在 GUI 中配置其参数 (gt\_quad\_base 参数继承自 gt\_bridge\_ip), 配置的参数包括通道数量, 完成后单击 “Block Automation” (块自动化设置)。它会基于 gt\_bridge\_ip 中配置的通道数量来例化多个 gt\_quad\_base, 并建立所有必要的连接。以下各图显示了含 Bridge IP 和 GT Quad base IP 的系统, 这些 IP 可在 IP integrator 画布内生成。每个 Bridge IP 均可配置为在 GUI 中启用多线速率选项。根据 Bridge IP 需求, 每个 GT Quad base IP 均可共享多个 Bridge IP。如需了解更多信息, 请参阅 [适用于定制 IP 的 IP integrator 设计输入](#)。

图 1：直通模式下的 Bridge IP



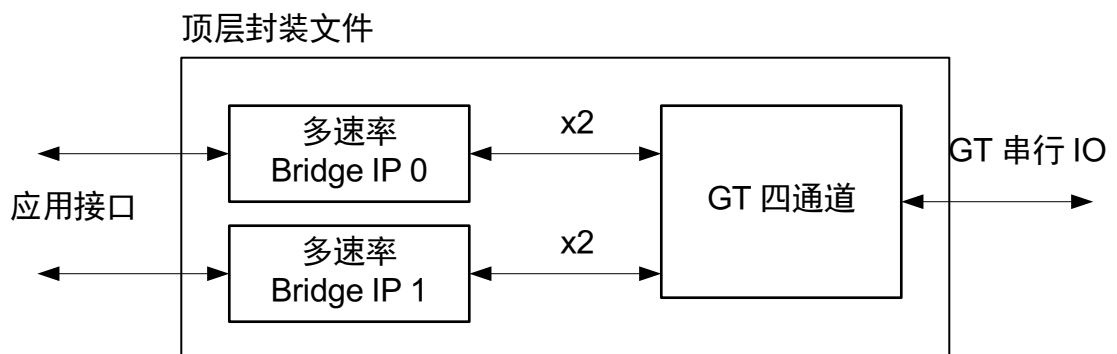
X24828-011023

图 2：单个 Bridge IP - GT 四通道设计



X24829-010923

图 3：多个 Bridge IP 共享 GT 四通道



X24802-010923

Bridge IP 中支持的功能特性如下所示：

- 单工和双工配置。
  - 适用于通用标准的预配置
  - 用于精调的完整参数控制
  - 用于连接单个或多个 GT 四通道的块自动化设置支持
- 为动态线速率更改提供原生支持，无需重新编程或构建 APB3 控制器。每条通道每个方向配置最多 16 种线速率

直通模式下的 Bridge IP 端口如下所示：

表 1：直通模式下的 Bridge IP 端口

| 名称                           | 方向 | 宽度      | 时钟域      | 描述  |
|------------------------------|----|---------|----------|---|
| gtreset_in                   | 输入 | 1       | 异步       | 此用户信号用于复位收发器原语的锁相环 (PLL) 和有效数据方向。<br>高电平有效异步脉冲持续保持至少一个 apb3clk 周期，这样即可初始化此进程。<br>此信号连接到复位控制器帮助程序块的 gt wiz_reset_all_in 信号。              |
| reset_tx_pll_and_datapath_in | 输入 | 1       | 异步       | 此用户信号用于复位收发器原语的发射数据方向和关联的 PLL。<br>高电平有效异步脉冲持续保持至少一个 apb3clk 周期，这样即可初始化此进程。<br>此信号连接到复位控制器帮助程序块的 gt wiz_reset_tx_pll_and_datapath_in 信号。 |
| reset_rx_pll_and_datapath_in | 输入 | 1       | 异步       | 此用户信号用于复位收发器原语的接收数据方向和关联的 PLL。<br>高电平有效异步脉冲持续保持至少一个 apb3clk 周期，这样即可初始化此进程。<br>此信号连接到复位控制器帮助程序块的 gt wiz_reset_rx_pll_and_datapath_in 信号。 |
| reset_tx_datapath_in         | 输入 | 1       | 异步       | 此用户信号用于复位收发器原语的发射数据方向。<br>高电平有效异步脉冲持续保持至少一个 apb3clk 周期，这样即可初始化此进程。<br>此信号连接到复位控制器帮助程序块的 gt wiz_reset_tx_datapath_in 信号。                 |
| reset_rx_datapath_in         | 输入 | 1       | 异步       | 此用户信号用于复位收发器原语的接收数据方向。<br>高电平有效异步脉冲持续保持至少一个 apb3clk 周期，这样即可初始化此进程。<br>此信号连接到复位控制器帮助程序块的 gt wiz_reset_rx_datapath_in 信号。                 |
| txresetdone_out              | 输出 | 1       | apb3clk  | 此高电平有效指示表示收发器原语的发射器复位序列已完成。   |
| rxresetdone_out              | 输出 | 1       | apb3clk  | 此高电平有效指示表示收发器原语的接收器复位序列已完成。   |
| ch*_txdata_ext               | 输入 | 128/256 | TXUSRCLK | 此用户接口可供通过收发器通道发射的数据使用（针对 GTYE5/GTYP 为 128 位，针对 GTME5 则为 256 位）。   |



表 1：直通模式下的 Bridge IP 端口 (续)

| 名称                      | 方向 | 宽度      | 时钟域      | 描述   |
|-------------------------|----|---------|----------|--|
| ch*_rxdata_ext          | 输出 | 128/256 | RXUSRCLK | 此用户接口可供通过收发器通道接收的数据使用（针对 GTYE5/GTYP 为 128 位，针对 GTME5 则为 256 位）。  |
| ch*_rxgearboxslip_ext   | 输入 | 1       | RXUSRCLK | 连接到收发器原语上的 RXGEARBOXSLIP。  |
| ch*_txheader_ext        | 输入 | 6       | TXUSRCLK | 连接到收发器原语上的 TXHEADER。   |
| ch*_txsequence_ext      | 输入 | 7       | TXUSRCLK | 连接到收发器原语上的 TXSEQUENCE。   |
| ch*_rxstartofseq_ext    | 输出 | 2       | RXUSRCLK | 连接到收发器原语上的 RXSTARTOFSEQ。   |
| ch*_rxheader_ext        | 输出 | 6       | RXUSRCLK | 连接到收发器原语上的 RXHEADER。   |
| ch*_rxheadervalid_ext   | 输出 | 2       | RXUSRCLK | 连接到收发器原语上的 RXHEADERVALID。  |
| ch*_rxdatavalid_ext     | 输出 | 2       | RXUSRCLK | 连接到收发器原语上的 RXDATAVALID。  |
| ch*_txctrl0_ext         | 输入 | 16      | TXUSRCLK | 连接到收发器原语上的 TXCTRL0。  |
| ch*_txctrl1_ext         | 输入 | 16      | TXUSRCLK | 连接到收发器原语上的 TXCTRL1。  |
| ch*_txctrl2_ext         | 输入 | 16      | TXUSRCLK | 连接到收发器原语上的 TXCTRL2。  |
| ch*_rxctrl0_ext         | 输出 | 16      | RXUSRCLK | 连接到收发器原语上的 RXCTRL0。  |
| ch*_rxctrl1_ext         | 输出 | 16      | RXUSRCLK | 连接到收发器原语上的 RXCTRL1。  |
| ch*_rxctrl2_ext         | 输出 | 8       | RXUSRCLK | 连接到收发器原语上的 RXCTRL2。  |
| ch*_rxctrl3_ext         | 输出 | 8       | RXUSRCLK | 连接到收发器原语上的 RXCTRL3。  |
| ch*_rxchbondi_ext       | 输入 | 5       | RXUSRCLK | 连接到收发器原语上的 RXCHBONDI。  |
| ch*_rxchanbondseq_ext   | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXCHANBONDSEQ。  |
| ch*_rxchanisaligned_ext | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXCHANISALIGNED。  |
| ch*_rxchanrealign_ext   | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXCHANREALIGN。  |
| ch*_rxchbondo_ext       | 输出 | 5       | RXUSRCLK | 连接到收发器原语上的 RXCHBONDO。  |
| ch*_txbufstatus_ext     | 输出 | 2       | TXUSRCLK | 连接到收发器原语上的 TXBUFSTATUS。  |
| ch*_rxbufstatus_ext     | 输出 | 3       | RXUSRCLK | 连接到收发器原语上的 RXBUFSTATUS。  |
| ch*_rxcommadet_ext      | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXCOMMADET。   |
| ch*_rxbyteisaligned_ext | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXBYTEISALIGNED。  |
| ch*_rxbyterealign_ext   | 输出 | 1       | RXUSRCLK | 连接到收发器原语上的 RXBYTEREALIGN。  |
| gpio_enable             | 输入 | 1       | apb3clk  | 在线速率切换期间当 GT REFCLK 值发生更改时，此用户信号断言有效。它应保持断言有效直至 *resetdone 信号变为高电平为止。如需了解更多信息，请参阅第 5 章的“gpio_enable 端口用法”部分。 |

表 1: 直通模式下的 Bridge IP 端口 (续)

| 名称              | 方向 | 宽度 | 时钟域     | 描述   |
|-----------------|----|----|---------|--|
| gpo_in          | 输入 | 1  | apb3clk | 此信号搭配 <code>gpio_enable</code> 一起使用。在块自动化设置期间，会自动连接此信号。不使用时，不需要驱动它。  |
| gpi_out         | 输出 | 1  | apb3clk | 此信号搭配 <code>gpio_enable</code> 一起使用。在块自动化设置期间，会自动连接此信号。  |
| rate_sel[3:0]   | 输入 | 4  | apb3clk | 此输入速率配置端口用于控制 GT 四通道的 <code>ch*_txrate[7:0]</code> 端口和 <code>ch*_rxrate[7:0]</code> 端口。此处输入宽度仍为 4 位，对应于 Bridge/GT Quad IP 输入级别可取的配置输入的数量 (16)。 |
| Link_status_out | 输出 | 1  | apb3clk | 这是一个无关紧要信号，启用直通模式时可忽略此信号。禁用直通模式时，此信号用于指示设计示例仿真或确认期间的链路正常运行状态。当 TX 与 RX 的 PRBS 模式相匹配时，此信号即处于高电平有效状态。  |

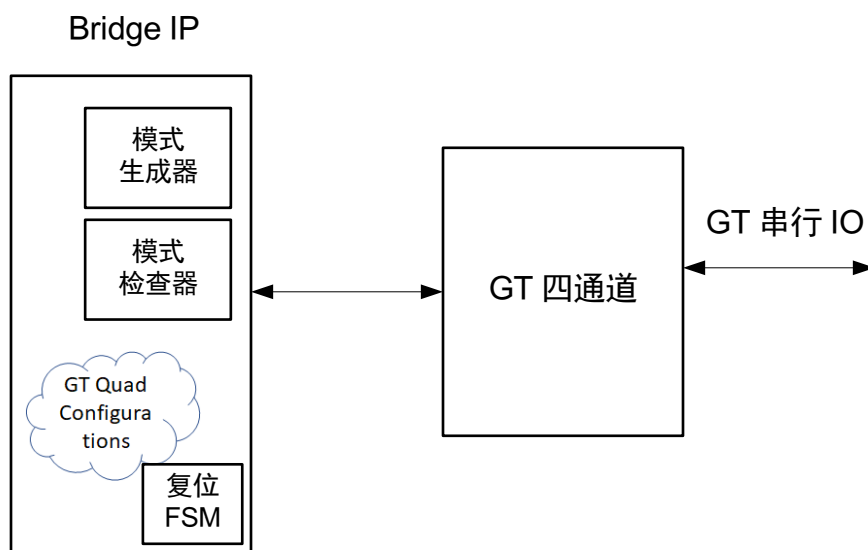
1. `gt_lcp11_lock`、`gt_rp11_lock`、`ch_phystatus_in` 和 `ilo_resetdone` 输入均为无关紧要的信号，且可忽略。

2. `gt_ilo_reset`、`gt_pll_reset`、`rp11_lock_out`、`lcp11_lock_out`、`pcie_rstb` 和 `reset_mask` 输出值不予使用，且可忽略。建议使用来自相应 `gt_quad_base` IP 的 `hsclk0_lcp11lock`、`hsclk1_lcp11lock`、`hsclk0_rp11lock` 和 `hsclk1_rp11lock` 信号进行监控。

3. 默认模式下的 Bridge IP 包含表 1: 直通模式下的 Bridge IP 端口 中列出的同一组端口 (`gpio_enable` 除外)。这些信号是在 Bridge IP 与 GT Quad 之间处理的，在 Bridge IP 的应用接口上不可用。如需获取有关复位序列在内部的发生方式的详细参考信息，请参阅《Versal ACAP GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal ACAP GTM 收发器架构手册》(AM017)。

在直通模式下不配置 Bridge IP 时，它包含模式生成器和检查器，用于支持各种编码/解码选项。这主要用于仿真。在此情况下，应用接口不可供使用。如需了解设计示例仿真，请参阅 第 5 章：设计示例。

图 4: 默认模式下的 Bridge IP



X24801-011023

### 通道绑定端口

通道绑定端口在 GT Bridge IP 中不可用，但您可使用 CHANNEL\_BONDING 参数在 Quad（四通道）级别启用这些端口。

生成 IP 核期间，请在 dict 中包含以下 Tcl 命令作为核生成的一部分：

```
set CHANNEL_BONDING true
```

Quad IP 中的通道绑定端口如下所示

表 2: Quad IP 中的通道绑定端口

| 名称                    | 方向 | 宽度 | 时钟域     | 描述   |
|-----------------------|----|----|---------|--|
| ch*_rxchanbond_en     | 输入 | 1  | apb3clk | 更新 CH*_RX_ELASTIC_BUF_CFG7 寄存器的 EB8B10B_CHAN_BOND_EN 位     |
| ch*_rxchanbond_master | 输入 | 1  | apb3clk | 更新 CH*_RX_ELASTIC_BUF_CFG7 寄存器的 EB8B10B_CHAN_BOND_MASTER 位 |
| ch*_rxchanbond_slave  | 输入 | 1  | apb3clk | 更新 CH*_RX_ELASTIC_BUF_CFG7 寄存器的 EB8B10B_CHAN_BOND_SLAVE 位  |
| ch*_rxchanbond_level  | 输入 | 3  | apb3clk | 更新 CH*_RX_ELASTIC_BUF_CFG7 寄存器的 EB8B10B_CHAN_BOND_LEVEL 位  |
| ch*_rxchanbond_busy   | 输出 | 1  | apb3clk | 指示将通道绑定端口值更新到寄存器位的状态                                       |

如需了解有关主复位序列的详细信息，请参阅《Versal ACAP GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal ACAP GTM 收发器架构手册》(AM017)。

## 复位控制器帮助程序块

您必须为“Reset Controller”（复位控制器）帮助程序块提供在 IP 自定义期间已指定的自由运行的时钟 `gtwiz_reset_clk_freerun_in`。每个 GT Bridge IP 核的实例都随附单个帮助程序块实例。

表 3: 复位控制器帮助程序块端口

| 名称                         | 方向 | 宽度 | 时钟域 | 描述                              |
|----------------------------|----|----|-----|---------------------------------|
| gtwiz_reset_clk_freerun_in | 输入 | 1  | 异步  | 自由运行的时钟，用于复位收发器原语。必须在器件配置前进行切换。 |

表 3：复位控制器帮助程序块端口 (续)

| 名称                                 | 方向 | 宽度 | 时钟域 | 描述  |
|------------------------------------|----|----|-----|---|
| gtwiz_reset_all_in                 | 输入 | 1  | 异步  | 此信号用于复位收发器原语的锁相环 (PLL) 和有效数据方向。高电平有效异步脉冲下降沿持续保持至少一个 gtwiz_reset_clk_freerun_in 周期，这样即可初始化此进程。               |
| gtwiz_reset_tx_pll_and_datapath_in | 输入 | 1  | 异步  | 此信号用于复位收发器原语的发射数据方向和关联的 PLL。高电平有效异步脉冲持续保持至少一个 gtwiz_reset_clk_freerun_in 周期，这样即可初始化此进程。                     |
| gtwiz_reset_tx_datapath_in         | 输入 | 1  | 异步  | 此信号用于复位收发器原语的发射数据方向。高电平有效异步脉冲持续保持至少一个 gtwiz_reset_clk_freerun_in 周期，这样即可初始化此进程。                             |
| gtwiz_reset_rx_pll_and_datapath_in | 输入 | 1  | 异步  | 此信号用于复位收发器原语的接收数据方向和关联的 PLL。高电平有效异步脉冲持续保持至少一个 gtwiz_reset_clk_freerun_in 周期，这样即可初始化此进程。                     |
| gtwiz_reset_rx_datapath_in         | 输入 | 1  | 异步  | 此信号用于复位收发器原语的接收数据方向。高电平有效异步脉冲持续保持至少一个 gtwiz_reset_clk_freerun_in 周期，这样即可初始化此进程。                             |
| gtpowergood_in                     | 输入 | 1  | 异步  | 连接到收发器通道逻辑生成的 GTPowerGood 信号。   |
| gtwiz_reset_userclk_tx_active_in   | 输入 | 1  | 异步  | 对于 GTY 和 GTPY：由收发器通道原语生成的所有 TXPMARESETDONE 信号的逻辑 AND。<br>对于 GTM：由收发器通道原语生成的所有 TXPROGDIVRESETDONE 信号的逻辑 AND。 |
| gtwiz_reset_userclk_rx_active_in   | 输入 | 1  | 异步  | 对于 GTY 和 GTPY：由收发器通道原语生成的所有 RXPMARESETDONE 信号的逻辑 AND。<br>对于 GTM：由收发器通道原语生成的所有 RXPROGDIVRESETDONE 信号的逻辑 AND。 |
| mst_tx_resetdone                   | 输入 | 1  | 异步  | 由收发器通道原语生成的所有 MSTTXRESETDONE 信号的逻辑 AND。   |
| mst_rx_resetdone                   | 输入 | 1  | 异步  | 由收发器通道原语生成的所有 MSTRXRESETDONE 信号的逻辑 AND。   |

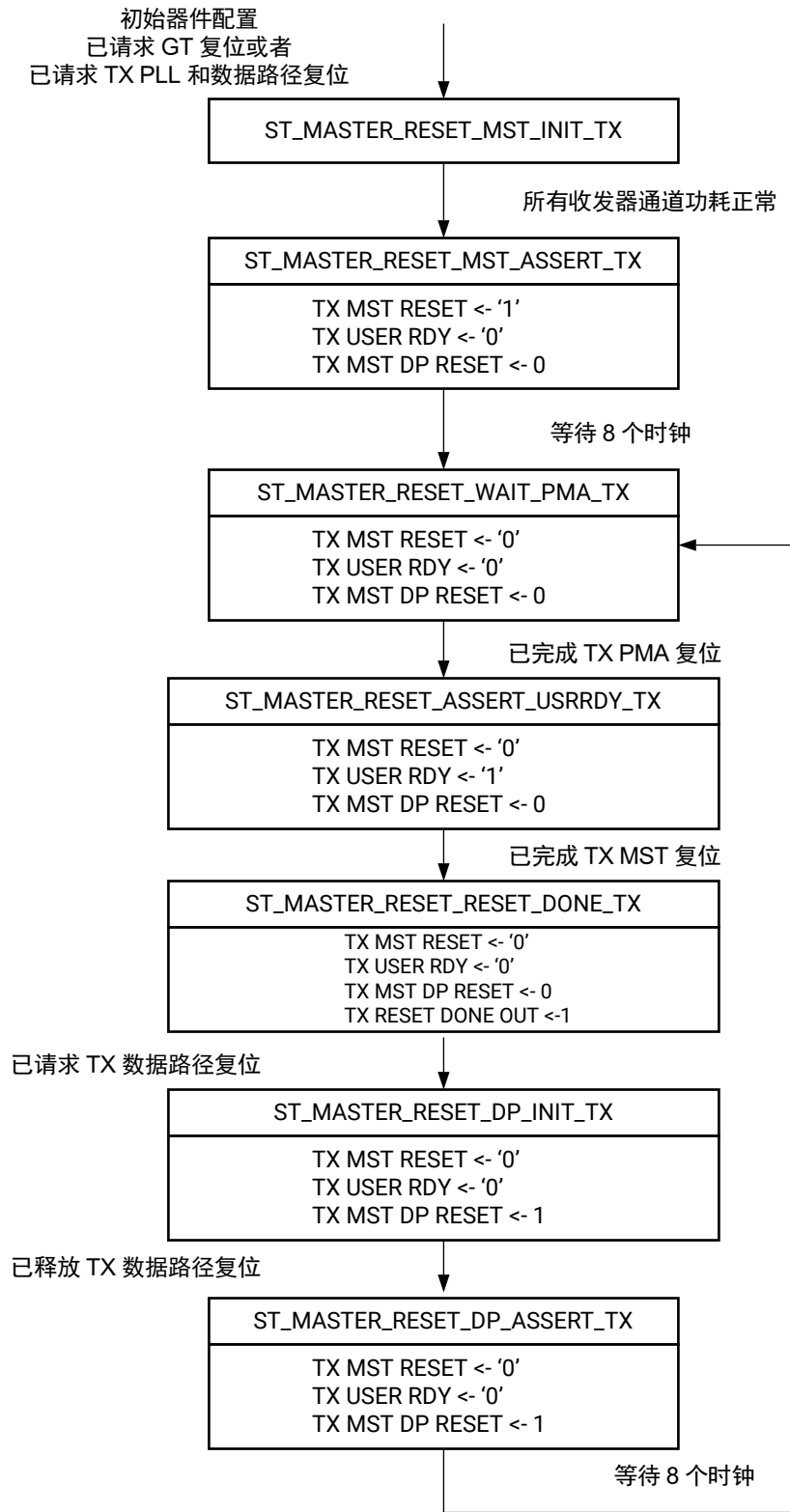
表 3：复位控制器帮助程序块端口 (续)

| 名称                      | 方向 | 宽度 | 时钟域                        | 描述  |
|-------------------------|----|----|----------------------------|---|
| mst_tx_reset            | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到收发器通道的 TXMSTRESET 端口。           |
| mst_rx_reset            | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到收发器通道的 RXMSTRESET 端口。           |
| mst_tx_dp_reset         | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到收发器通道的 TXMSTDATAPATHRESET 端口。   |
| mst_rx_dp_reset         | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到收发器通道的 RXMSTDATAPATHRESET 端口。   |
| txuserrdy_out           | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到所有收发器通道原语的 TXUSERRDY 端口。        |
| rxuserrdy_out           | 输出 | 1  | gtwiz_reset_clk_freerun_in | 高电平有效信号扇出到所有收发器通道原语的 RXUSERRDY 端口。        |
| gtwiz_reset_tx_done_out | 输出 | 1  | gtwiz_reset_clk_freerun_in | 此高电平有效指示表示由复位控制器帮助程序块发起的收发器原语的发射器复位序列已完成。 |
| gtwiz_reset_rx_done_out | 输出 | 1  | gtwiz_reset_clk_freerun_in | 此高电平有效指示表示由复位控制器帮助程序块发起的收发器原语的接收器复位序列已完成。 |

帮助程序块遵循主复位序列，并包含 2 个状态机。

- 发射器复位状态机：用于复位所有收发器原语的发射器 PLL 和/或发射器数据路径，并指示其完成状态。

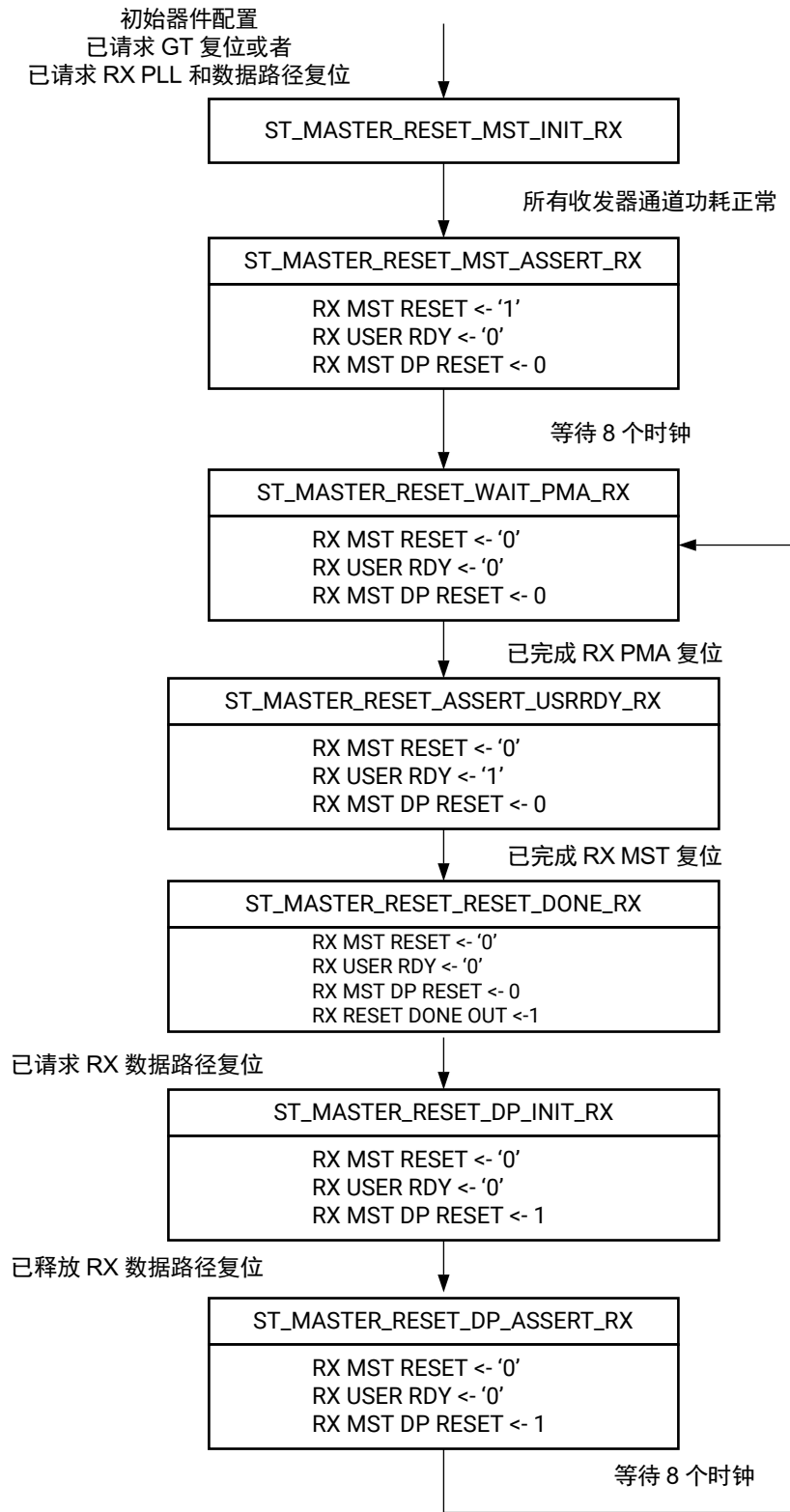
图 5：发射器复位状态机



X27150-011023

- 接收器复位状态机：用于复位所有收发器原语的接收器 PLL 和/或接收器数据路径，并指示其完成状态。

图 6：接收器复位状态机



X27149-011023



# 设计流程步骤

本章节描述了核的自定义和生成方式、核的约束方式以及此 IP 核的仿真、综合与实现的具体步骤。如需获取有关标准 Vivado® 设计流程以及有关 IP integrator 的详细信息，请参阅以下 Vivado Design Suite 用户指南：

- 《Vivado Design Suite 用户指南：采用 IP integrator 设计 IP 子系统》(UG994)
- 《Vivado Design Suite 用户指南：采用 IP 进行设计》(UG896)
- 《Vivado Design Suite 用户指南：入门指南》(UG910)
- 《Vivado Design Suite 用户指南：逻辑仿真》(UG900)

## 自定义和生成核

本节包含有关如何使用赛灵思工具在 Vivado® Design Suite 中自定义和生成核的信息。

如果要在 Vivado IP integrator 中自定义和生成核，请参阅《Vivado Design Suite 用户指南：采用 IP integrator 设计 IP 子系统》(UG994) 了解详情。确认或生成设计时，IP integrator 可能会自动计算某些配置值。要查看配置值是否会更改，请参阅本章中的参数说明。要查看参数值，请在 Tcl 控制台中运行 `validate_bd_design` 命令。

正如 [第 3 章：产品规格](#) 中所提到的，Versal® ACAP Transceivers Wizard 解决方案包含 2 个核：

1. Versal ACAP Transceivers Bridge - 这是参考父级 IP (Bridge IP)，用于配置 Transceivers Wizard 参数。如需了解更多信息，请参阅“适用于定制 IP 的 IP integrator 设计输入”。
2. Versal ACAP Transceivers Wizard - 围绕 GT\*\_QUAD 原语的封装文件。其中包含单个 GT 四通道 (GT quad base IP)。对于多通道 (>4 条通道) 设计，将例化多个 Transceivers Wizard。请参阅以下“IP integrator 设计输入”部分，了解为定制设计输入而建议采用的设计输入。赛灵思 GT 父级 IP 将支持块自动化设置，以获取所需的连接。

您可以遵循以下步骤通过指定与 IP 核关联的各种参数值来自定义设计中使用的 IP：

1. 从 IP 目录选择 IP。
2. 双击所选 IP，或者从工具栏或右键单击菜单中选择“Customize IP”（自定义 IP）命令。

欲知详情，请参阅《Vivado Design Suite 用户指南：采用 IP 进行设计》(UG896) 和《Vivado Design Suite 用户指南：入门指南》(UG910)。

本章中的附图是 Vivado IDE 的插图。此处展示的布局可能与当前版本中的布局有所不同。

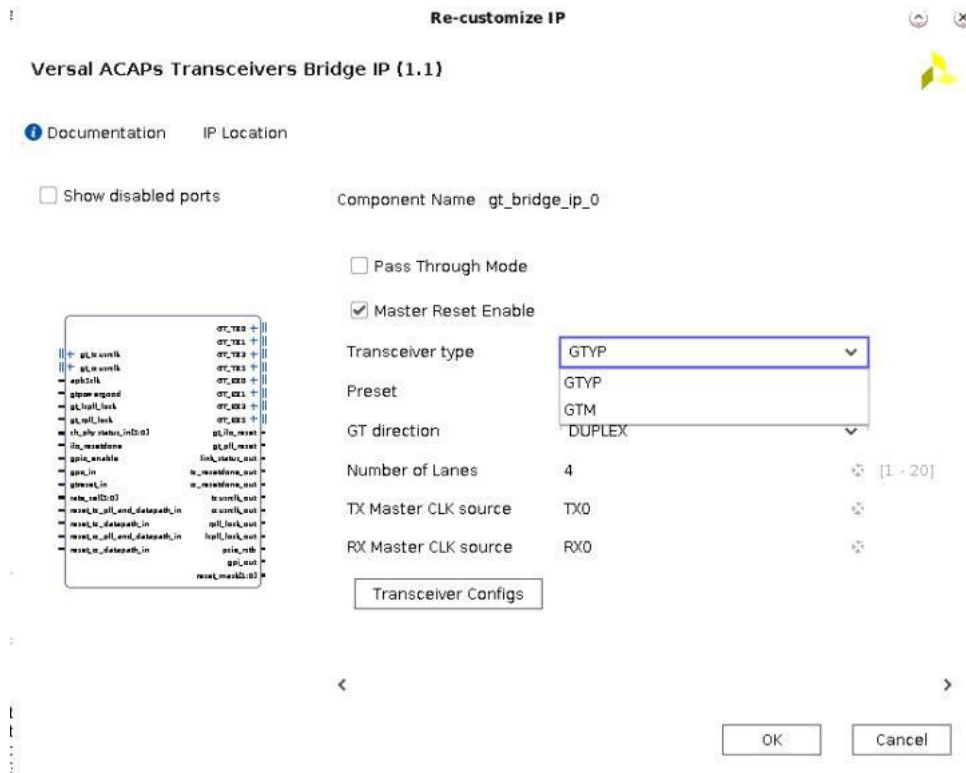
## 适用于定制 IP 的 IP integrator 设计输入

定制 IP 的设计输入是通过 Bridge IP (`gt_bridge_ip`) 来完成的。定制 IP 无需封装在 IP integrator 内。您需要在 Bridge IP 内输入自己的要求，并使用块自动化设置来生成单个或多个 GT 四通道设计。生成顶层文件，并将其与定制 IP 相连。Bridge IP GUI 输入是 GT Wizard (`gt_quad_base`) 的副本。因此，系统生成期间，Bridge IP GUI 中输入的值会自动填充到 GT 四通道内。

您可在 IP integrator 内添加 `gt_bridge_ip`。双击 IP 符号打开 Bridge IP GUI。在 GUI 中选中“Pass Through Mode”（直通模式）启用该选项。这样即可显示用户接口的必需 GT 并行接口。配置包括通道数量在内的其它参数（`gt_quad_base` 参数可通过 `gt_bridge_ip` GUI 进行编程）。

- “Main GUI Configuration” 选项卡：“Main GUI Configuration”（主 GUI 配置）选项卡提供了各种自定义选项，可用于收发器预设选择、方向、主时钟源和通道数量。“Main GUI”（主 GUI）选项卡如下图所示：

图 7：Bridge IP GUI



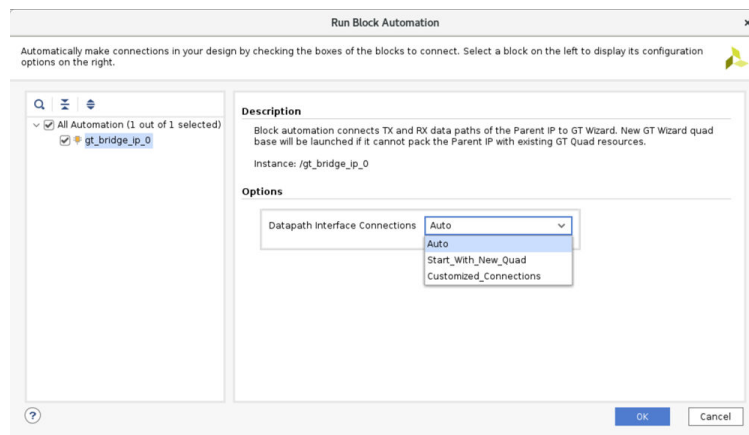
- “Component Name”（组件名称）：在“Component Name”字段中设置生成的 IP 的名称。默认名称是 `gt_bridge_ip_0`。
- “Pass Through Mode”（直通模式）：选择该选项即可显示用户接口的必需 GT 并行接口。
- “Master Reset Enable”（主复位启用）：在 GT 四通道中启用主复位逻辑。默认启用此项。您不应禁用该选项。
- “Preset”（预设）：支持标准协议预设。选择预设即可加载子 GUI 中对应的收发器设置。在此“Preset”窗口中可加载多线速率预设。它将填充对应子 GUI 中的多个单线速率预设。例如，在预设列表中会添加多线速率预设（含后缀 MLR）。
- “GT Direction”（GT 方向）：在下拉列表中选择给定的 GT 方向选项。
- “Number of Lanes”（通道数）：选择所需的 GT 通道数量。如果 GTME5 配置所需用户数据宽度为 320/512，那么当线速率大于 56Gb/s 时，四通道上添加的通道数量作为活动的 PCS 通道数来处理，因为两种 PCS 通道都会得到使用。
- “TX RX Master Clock Source”（TX RX 主时钟源）：在下拉列表中给定的主时钟源选项中选择所需的选项。
- “Transceiver Configs”（收发器配置）：该选项会打开弹出式子 GUI，您可在其中配置收发器的发射器和接收器设置。

如需了解更多信息，请参阅 [“Sub GUI Configuration” 选项卡](#)。

## 块自动化设置流程

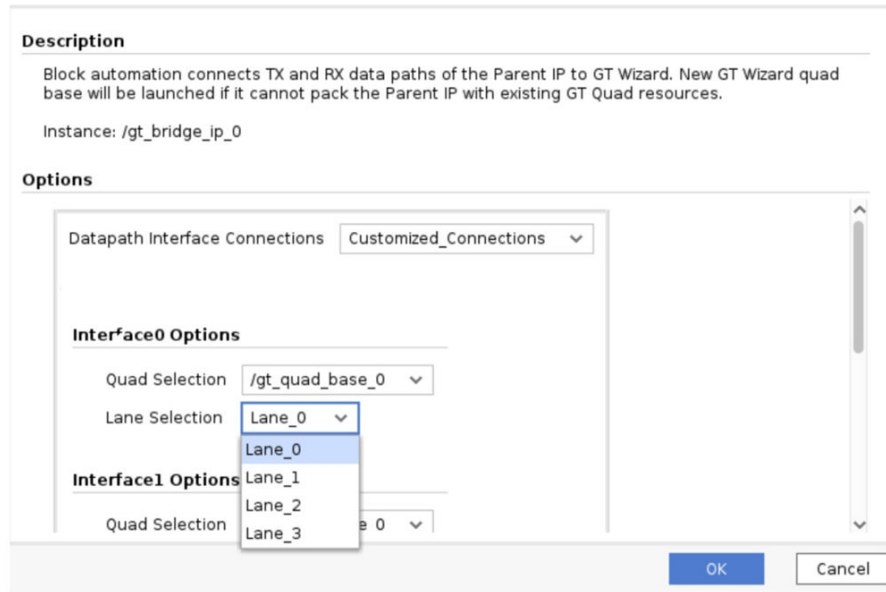
在 IP integrator 画布中添加 Bridge IP 时，会在顶部显示绿色功能区。其中包含块自动化设置链接，用于自动设置部分数据路径、时钟和复位连接。配置 Bridge IP 后单击“Block Automation”（块自动化设置）时，会打开“Block Automation” GUI。选中“Auto”（自动）即表示您允许“Block Automation”选择 GT 四通道资源的最优用法。该工具根据已知可用时钟资源以及 REFCLK 和 PLL 共享规则来制定决策。选中“Start\_With\_New\_Quad”（以新四通道开始）后，“Block Automation”会例化新的 GT 四通道，并建立数据路径、时钟和复位连接。在全新 IP integrator 画布上，“Auto”选项与“Start\_With\_New\_Quad”选项的行为方式相似。它会基于 `gt_bridge_ip` 中配置的通道数量来例化多个 `gt_quad_base`，并建立数据路径、时钟和复位的连接。

图 8: Bridge IP 的“Block Automation” GUI



选中“Customized\_Connections”（自定义连接）后，GUI 会展开并显示系统中已有的所有可用且有效的通道。您可在 GUI 中选择相应的“GT Quad Selection”（GT 四通道选择）选项和“Lane Selection”（通道选择）选项来将 Bridge IP 的 GT 接口连接到 GT Quad base IP 的任意通道。请注意，在“Customized\_Connections”选项中，如果通道不可用，就不会自动例化新的 GT Quad base IP。您必须先画布中手动添加 GT Quad base IP，然后再单击“Customized\_Connections”。例如，在空画布中，您必须手动例化 Bridge IP 和 GT Quad base IP，这样“Customized\_Connections”才能生效。大部分 GT 父级 IP 都应已完成移植并可支持此操作，请参阅相应的 GT 父级 IP 产品指南，获取有关此高级功能特性的用法的进一步详细信息。“Customized\_Connections” GUI 选项如下图所示：

图 9: “Customized\_Connections” GUI 选项



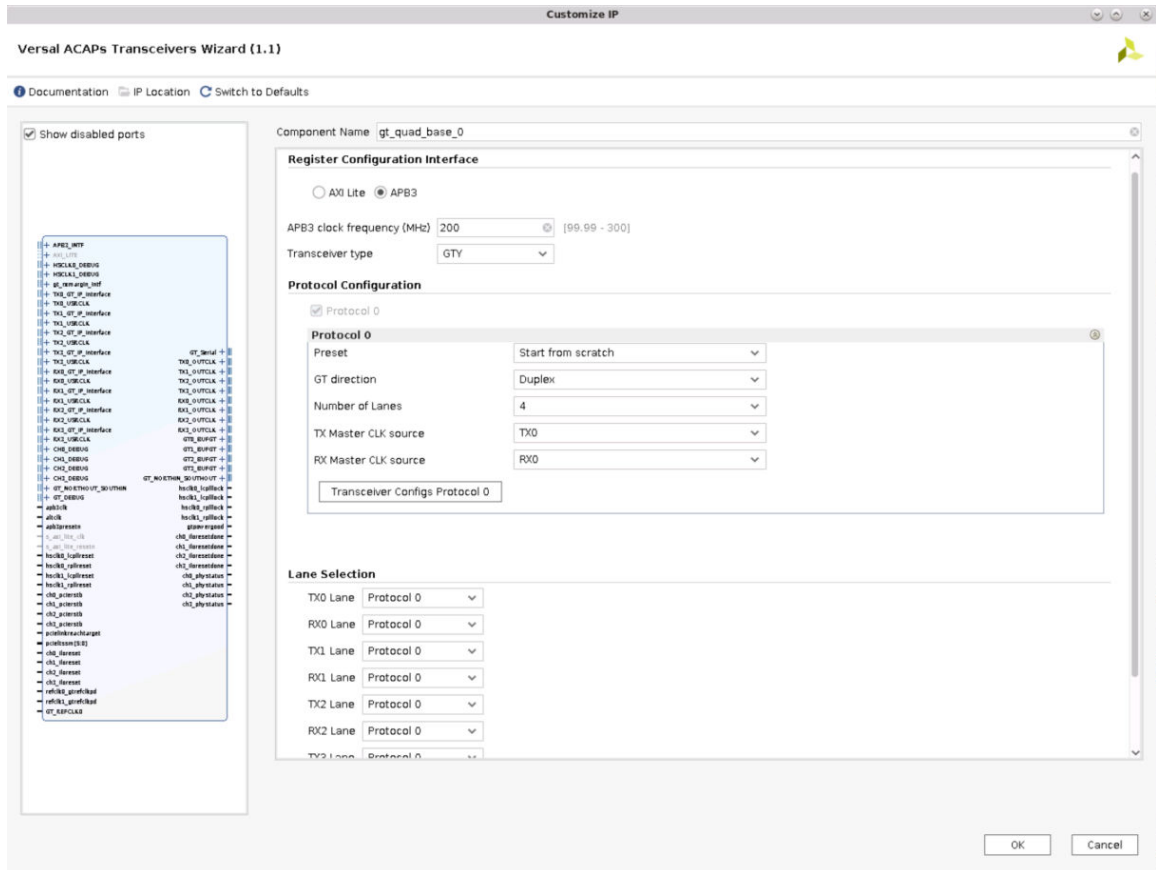
## Versal ACAP Transceivers Wizard

Versal® ACAP Transceivers Wizard 包含在 IP 目录中以便使用。它封装有单一 GT\*\_QUAD 原语。其中为一个 GT 四通道提供了仿真支持。您可通过配置 GT Quad base IP 来共享多个协议 IP，每个协议 IP 都支持多种线速率。您可通过创建设计示例并仿真来了解多个协议 IP 之间共享的动态线速率切换和 GT 四通道。

### “Main GUI Configuration” 选项卡

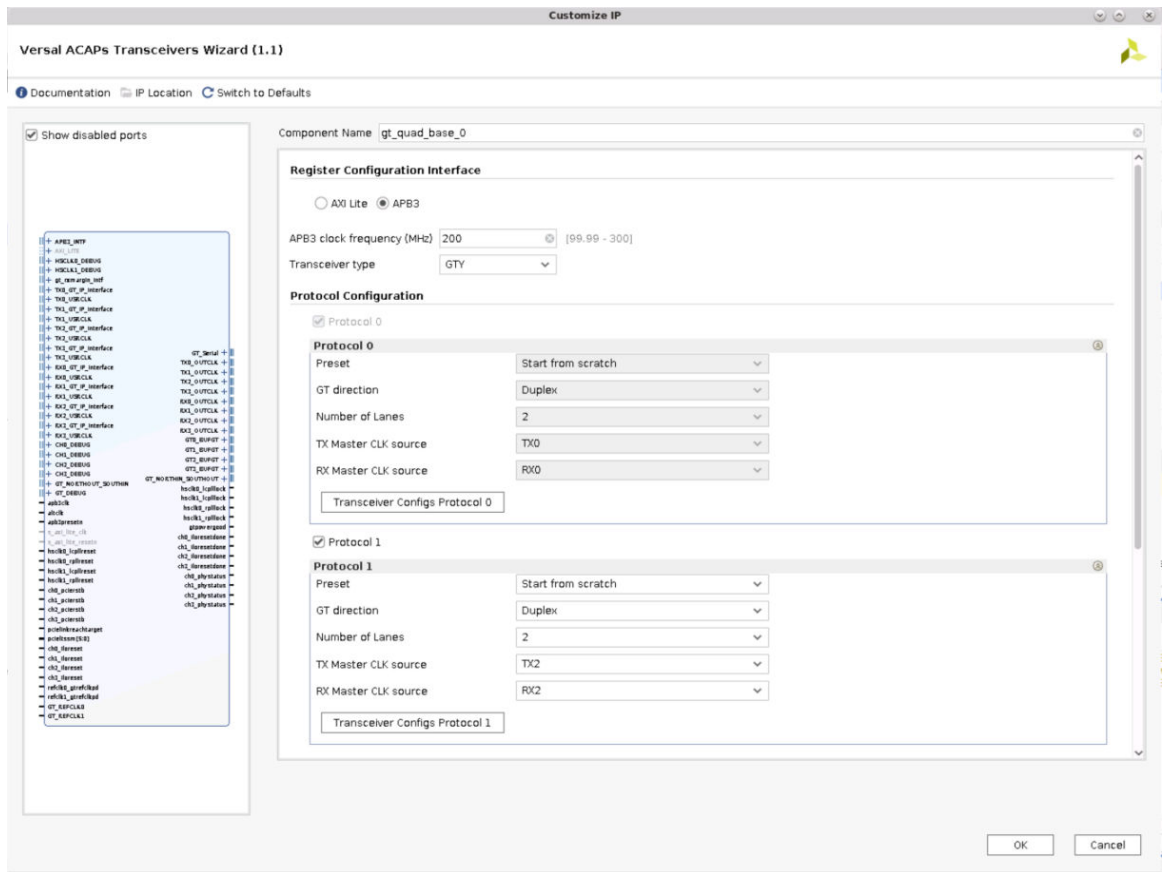
“Main GUI Configuration”（主 GUI 配置）选项卡提供了各种自定义选项，可用于收发器预设选择、方向、主时钟源和通道数量。“Main GUI”（主 GUI）选项卡如下图所示：

图 10：“Main GUI Configuration” 选项卡



- “Component Name”（组件名称）：在“Component Name”字段中设置生成的 IP 的名称。默认名称是 `gt_quad_base_0`。
- “Register Configuration Interface”（寄存器配置接口）：针对寄存器配置，可选择 AXI4-Lite 或 APB3 接口。并可输入所选择接口的工作频率。APB3 是 32 位数据总线寻址，而 AXI4-Lite 则是基于字节的 8 位总线。在《Versal ACAP GTY 和 GTYP 收发器架构手册》(AM002) 中，寄存器配置寻址基于 APB3。因此，如果您使用 AXI4-Lite，则需将寻址乘以 4。基于所选寄存器配置接口，自由运行的时钟必须驱动 `apb3clk` 信号或 `s_axi_lite_clk` 信号。
- “Transceiver type”（收发器类型）：选择要配置的串行收发器类型。可用选项仅限于所选器件中存在的收发器类型。
- “Protocol 0”（协议 0）：作为四通道双工设计，默认启用该选项。基于未使用的通道数量，启用其余协议选项（1、2、3 等）。下图显示了两种协议选项，这两种协议选项均为双工，并且针对每个选项都会对 2 条通道进行编程：

图 11：多协议编程选项



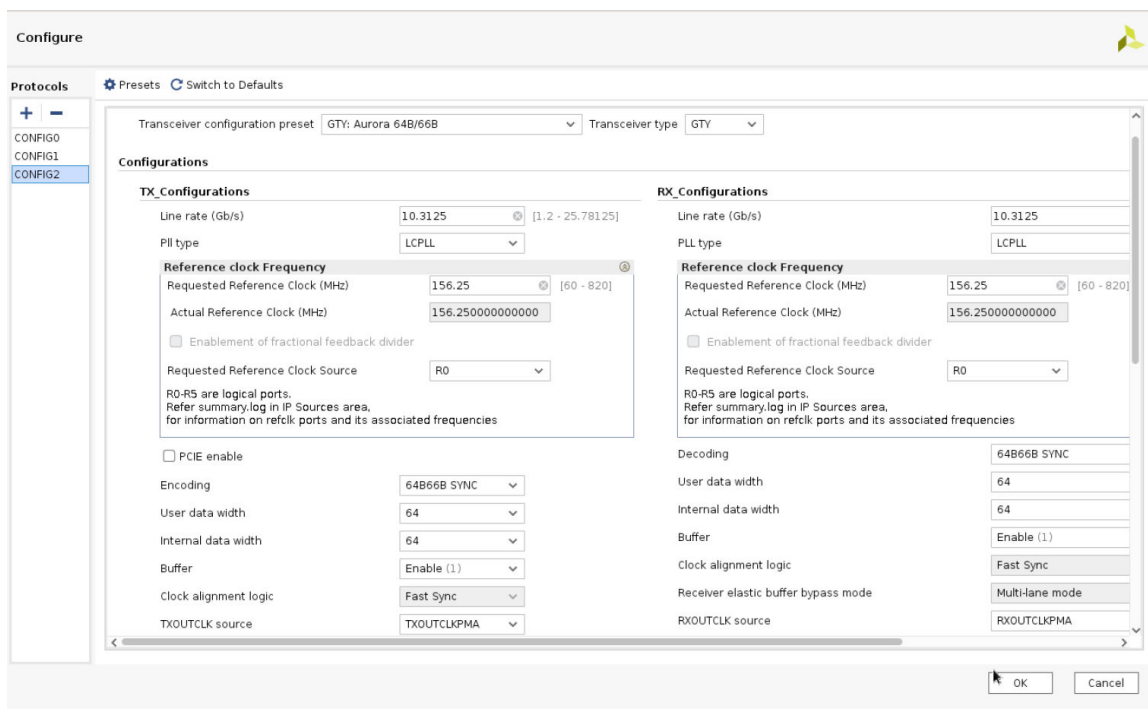
启用协议选项 1 时，协议选项 0 配置为灰显。您需要按顺序配置协议选项。如果您在启用协议选项 1 之后要修改协议选项 0，则需要禁用协议选项 1，然后再自定义协议选项 0。

- “Preset”（预设）：支持标准协议预设。选择预设将加载子 GUI 中的对应收发器设置。在此“Preset”窗口中可加载多线速率预设。它将填充对应子 GUI 中的多个单线速率预设。例如，在预设列表中会添加多线速率预设（含后缀 MLR）。
  - “GT Direction”（GT 方向）：在下拉列表中选择给定的 GT 方向选项。
  - “TX RX Master Clock Source”（TX RX 主时钟源）：在下拉列表中给定的主时钟源选项中选择所需的选项。
  - “Number of Lanes”（通道数）：选择通道数量。由于它是 GT 四通道原语的封装文件，因此支持最多 4 条通道。
- 注释：**如果 GTME5 配置所需用户数据宽度为 320/512，那么当线速率大于 56Gb/s 时，四通道上添加的通道数量作为活动的 PCS 通道数来处理，因为两种 PCS 通道都会得到使用。在给定 GTME5 Dual 中，仅有 1 条活动 PMA 通道可用于这些速率。请参阅配置协议 GUI 选项，了解活动 PMA 通道的选择。
- “Transceiver Configuration Protocol 0”（收发器配置协议 0）：该选项会打开弹出式子 GUI，您可在其中配置收发器的发射器和接收器设置。
  - “Lane Selection”（通道选择）：基于“Number of Lanes”值选择物理通道映射。

## “Sub GUI Configuration” 选项卡

“Sub GUI Configuration”（子 GUI 配置）选项卡提供了各种自定义选项，可用于选择基础收发器功能特性以及发射器和接收器设置。每个子 GUI 配置都对应于一个目标收发器配置，在“Protocols”（协议）面板中称之为 CONFIG<0/1...>。通过控制向导上的 `ch*_txrate[7:0]` 端口或 `ch*_rxrate[7:0]` 端口即可选择目标配置。多线速率设置需要多个子 GUI 配置。该选项卡如下图所示：

图 12：“Sub GUI Configuration” 选项卡



- “Protocol”（协议）：单击“+”按钮即可添加新的 CONFIGx。每个 CONFIG 都对应于一个子 GUI。
- “Transceiver configuration preset”（收发器配置预设）：可从预设列表加载标准协议预设。

CONFIG 选项的添加采用递增式操作。每次单击“+”按钮都将添加下一个 CONFIG 选项（例如，CONFIG 1、CONFIG 2 以此类推）。同样，CONFIG 选项的删除是递减式操作。每次单击“-”按钮都将删除前一个 CONFIG 选项（例如，CONFIG N、CONFIG N-1 以此类推）。不允许随机 CONFIG 选项。如果用户更改子 GUI 中的预设值，同时所选 GT 方向为“Simplex”（单工），如，“Simplex TX”（单工发射），那么子 GUI 会同时显示 TX 和 RX 配置字段。用户可以自定义 TX 配置字段，但还需在 RX 配置字段中输入有效的值才能生成核。但核生成规则基于主 GUI（在此例中，即其 Simplex TX）中输入的 GT 方向。

- “Configurations”（配置）：GT 收发器和接收器设置（例如，线速率、GT 类型、参考时钟频率）均可供选择。
- “Reference Clock Frequency”（参考时钟频率）：其中包含用户参考时钟频率输入。实际参考时钟频率是根据用户参考时钟输入计算得出的。
- “Requested Reference Clock Source”（请求的参考时钟源）：应按顺序提供 6 个逻辑参考时钟输入，从 R0 到 R5。基于用户的各种线速率选择，该向导会分配已排序的参考时钟端口 (GTREFCLK0-5) 和频率以供在这些时钟端口上进行驱动。“IP Sources”（IP 源文件）区域中的 `summary.log` 可提供时钟端口和频率分配以及 CONFIG 汇总信息表。下图提供了 `summary.log` 样本：

图 13: 汇总日志样本

```

-----GTREFCLK0 -----
GTREFCLK0 is (R0 of PROT0) a single frequency port
Drive 156.250000 MHz on GTREFCLK0

-----GTREFCLK1 -----
GTREFCLK1 is (R0 of PROT1) a single frequency port
Drive 125.000000 MHz on GTREFCLK1

-----Table Summary-----
+-----+-----+-----+
| Protocol Info | Parameters | CONFIG 0 |
+-----+-----+-----+
| Protocol 0 | TX LINE RATE (Gb/s) | 10.3125 |
| DUPLEX | TX PLL TYPE | LCPLL |
| 2 lanes | TX REF CLK FREQ (MHz) | 156.25 |
| | Actual TX REF CLK FREQ (MHz) | 156.25 |
| | RX LINE RATE (Gb/s) | 10.3125 |
| | RX PLL TYPE | LCPLL |
| | RX REF CLK FREQ (MHz) | 156.25 |
| | Actual RX REF CLK FREQ (MHz) | 156.25 |
| Protocol 1 | TX LINE RATE (Gb/s) | 10.3125 |
| DUPLEX | TX PLL TYPE | LCPLL |
| 2 lanes | TX REF CLK FREQ (MHz) | 125 |
| | Actual TX REF CLK FREQ (MHz) | 125 |
| | RX LINE RATE (Gb/s) | 10.3125 |
| | RX PLL TYPE | LCPLL |
| | RX REF CLK FREQ (MHz) | 125 |
| | Actual RX REF CLK FREQ (MHz) | 125 |
+-----+-----+-----+

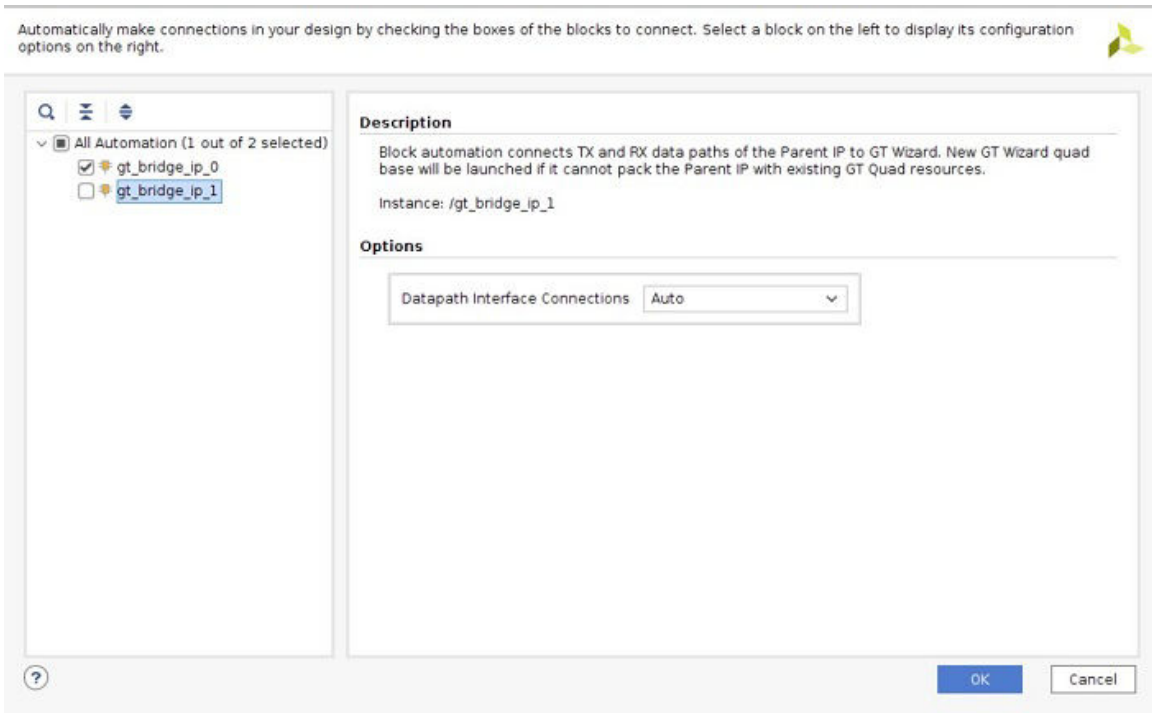
```

## 共享多个 Bridge IP 的 GT 四通道

根据 GT 四通道的通道数量、PLL 要求和其它配置，多个 Bridge IP 可共享一个 GT 四通道。例如，两个 x2 通道 Bridge IP 可共享一个 GT 四通道。为达成此结果，您可将 2 个 Bridge IP 放入 IP integrator 画布中、配置 Bridge IP GUI，然后单击“Block Automation”（块自动化设置）。这将打开“Block Automation” GUI，如下图所示。此图显示了 2 个 Bridge IP 的“Block Automation” GUI。您可以选择为这 2 个 Bridge IP 同时执行块自动化设置，也可以选择逐一执行设置。

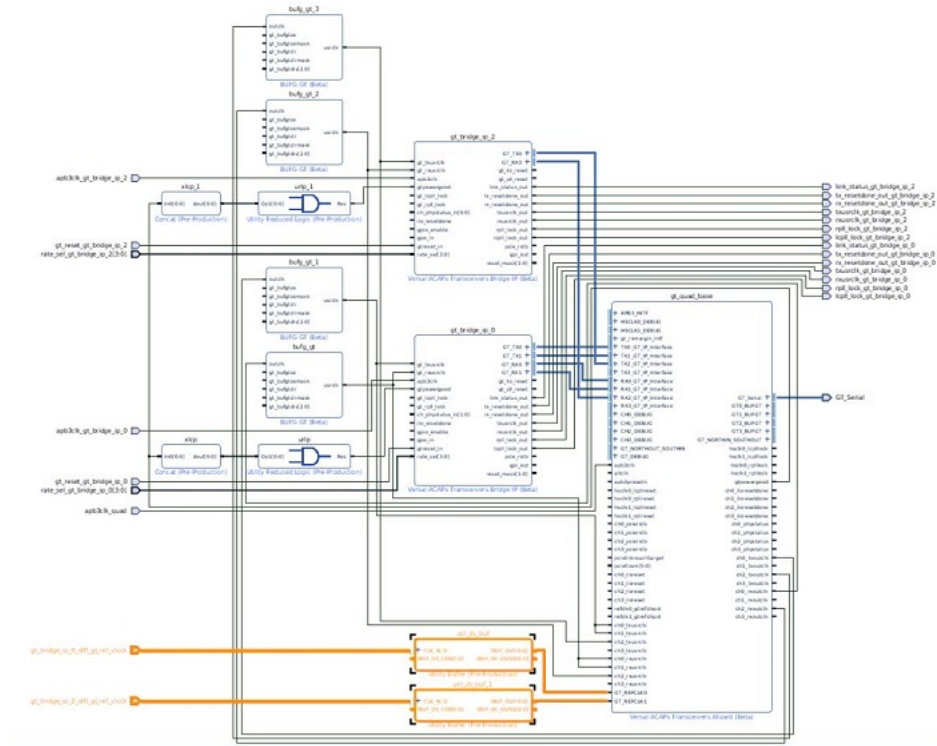


图 14：2 个 Bridge IP 的块自动化设置



下图显示了共享单个 GT 四通道的多个完全连接的 Bridge IP。

图 15：完全连接的 Bridge IP



## GT 参考时钟汇总与最优化

作为块自动化设置的一部分，在简单设计的 GT 四通道之间，GT 参考时钟 (GTREFCLK) 会缩短。

- 示例 1：对于使用 2 个 GT 四通道的 x8 设计，来自这 2 个 GT 四通道的 GTREFCLK 都会缩短，并连接到单个 IBUFDS\_GT。对于使用含多个 GT 四通道的多个 Bridge IP 的复杂设计，系统设计师可以基于其频率信息、开发板上的四通道布局和时钟可用性来缩短和拆分 GTREFCLK。为了帮助系统设计师制定出明智的决策，在 IP integrator 的块设计画布中为整个系统提供了 GTREFCLK 汇总信息。请注意，对于具有多个块设计的 Vivado 工程，必须为每个给定的块设计执行一次该命令。REFCLK 汇总信息是根据 BD 生成的。如果系统具有多个 BD，那么必须单独生成 REFCLK 汇总信息。您可在 Tcl 控制台中输入以下命令来获取该表。

```
xilinx::designutils::report_gt_refclk_summary
```

执行命令时，会在以下路径中生成 <BD\_name>\_gt\_refclk\_summary.txt

图 16：GT 参考时钟汇总信息文件位置



它会报告此设计中的 GT 参考时钟、其频率及其源文件，如 图 15：完全连接的 Bridge IP 中所示。

图 17：示例 1 GT 参考时钟汇总信息表

```
===== GT_REFCLK Summary Table =====
```

| S.No. | GT REFCLK Name             | Freq   | ParentIP        | REFCLK Source  | GT Type |
|-------|----------------------------|--------|-----------------|----------------|---------|
| 1     | /gt_quad_base/GT_REFCLK0   | 156.25 | /gt_bridge_ip_0 | /util_ds_buf   | GTYP    |
| 2     | /gt_quad_base_1/GT_REFCLK0 | 156.25 | /gt_bridge_ip_1 | /util_ds_buf_1 | GTYP    |

- 示例 2：此图 显示了共享 2 个 GT 四通道的 x2 IP 的 4 个实例。在此例中，设计中有 4 个 GTREFCLK 可用。由于这 4 个实例均为同一个 IP，因此所有 IP 的 GTREFCLK 频率都相同。如果 GT 四通道布局位置彼此相邻并使用单个输入管脚，那么系统设计师可能可以相同 GTREFCLK 来驱动这些实例。此图 显示了此设计的 GTREFCLK 汇总信息：

图 18: 多个 Bridge IP 共享 GT 四通道

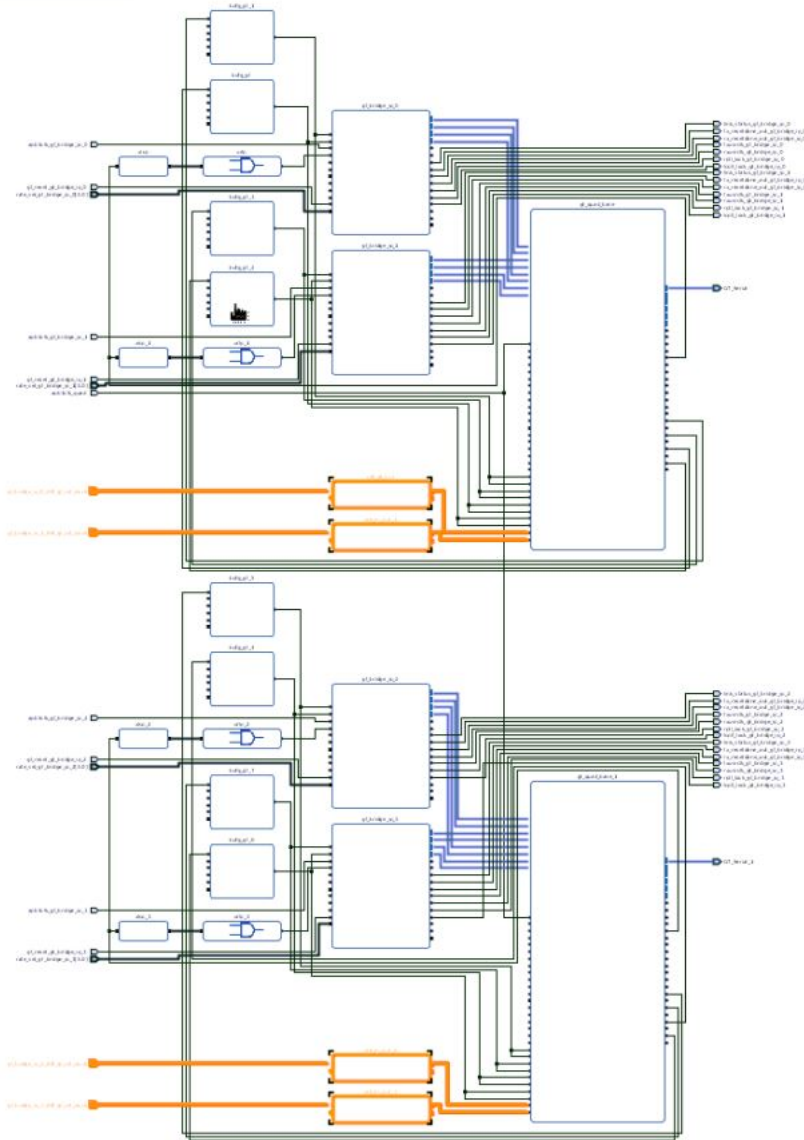


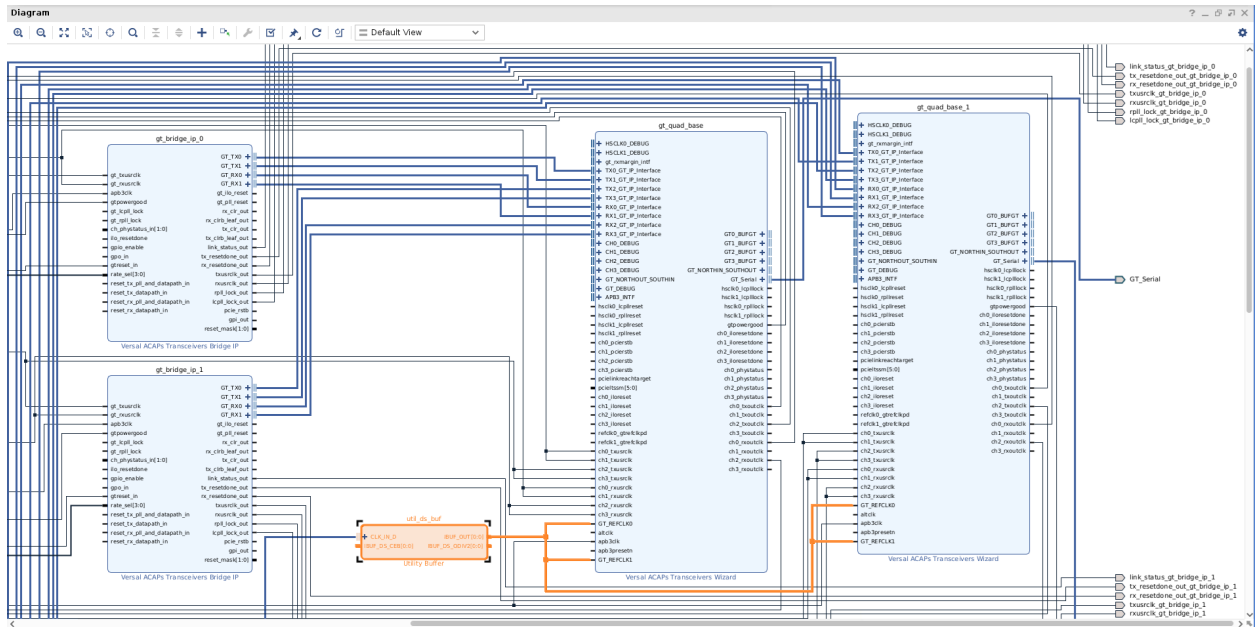
图 19: 示例 2 GT 参考时钟汇总信息表

===== GT\_REFCLK Summary Table =====

| S.No. | GT REFCLK Name             | Freq   | ParentIP        | REFCLK Source  | GT Type |
|-------|----------------------------|--------|-----------------|----------------|---------|
| 1     | /gt_quad_base/GT_REFCLK0   | 156.25 | /gt_bridge_ip_0 | /util_ds_buf   | GTy     |
| 2     | /gt_quad_base/GT_REFCLK1   | 156.25 | /gt_bridge_ip_1 | /util_ds_buf_1 | GTy     |
| 3     | /gt_quad_base_1/GT_REFCLK0 | 156.25 | /gt_bridge_ip_2 | /util_ds_buf_2 | GTy     |
| 4     | /gt_quad_base_1/GT_REFCLK1 | 156.25 | /gt_bridge_ip_3 | /util_ds_buf_3 | GTy     |

基于汇总信息表，可通过单个实用工具缓冲器来连接各四通道上的 GT REFCLK，如下图所示。

图 20：共享 GT 四通道的多个 Bridge IP 的修改后参考时钟连接



## 赛灵思 IP - GT 四通道集成

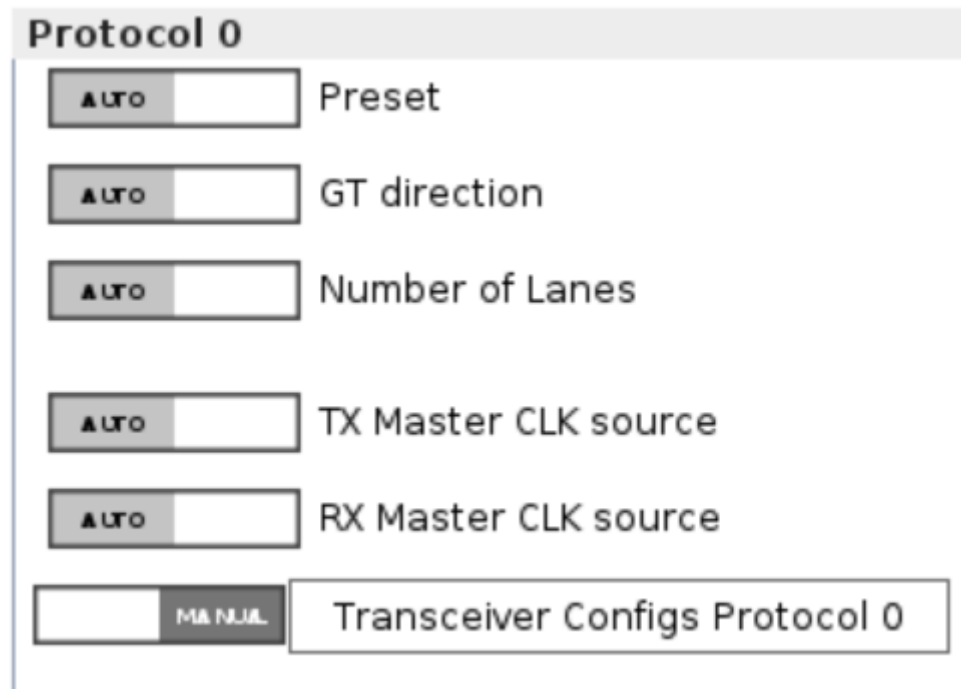
基于赛灵思 GT 的 IP (如 Aurora、PCIe 和 MRMAC) 可在 IP integrator 中提供块自动化设置, 以支持您将多个父级赛灵思 IP 无缝连接到 GT 四通道。IP 块自动化设置可例化 GT 四通道, 并创建基本数据路径 USRCLK 和 GT REFCLK 连接。

要使用块自动化设置来连接多个 Aurora IP, 请执行以下步骤:

1. 使用 IP integrator 画布中的“Add IP” (添加 IP) 选项添加“Aurora64B66B” IP。
2. 按通道数量、线速率等配置“Aurora64B66B” IP。
3. 单击“Run\_Block\_Automation”。在“Block Automation” (块自动化设置) GUI 中, 选择下列选项之一: “Auto” (自动)、“Start\_with\_New\_Quad” (以新四通道开始) 或“Customized\_Connections” (自定义连接)。如需了解有关块自动化选项的详细信息, 请参阅 [块自动化设置流程](#)。
4. 根据您的系统需求, 执行步骤 2 和 3 以添加更多“Aurora64B66B” IP 实例。

确认设计时, 会将 GT 四通道参数从相连的 IP 传输至这些 GT 四通道。这样即可在“Transceiver Wizard” GUI 中将所有 GT 四通道参数标记为“Auto” (自动)。但您可在“Transceiver Configs” (收发器配置) 中将“Auto”选项更改为“Manual” (手动), 以便对“insertion loss” (插入损耗)、“drive strength” (驱动强度)、“equalization” (均衡) 等参数以及其它高级设置进行微调, 如下图所示。切换为“Manual”模式后, 无论对父级 IP 配置进行任何更改, 而后执行确认设计步骤, 都不会再将任何 GT 四通道参数从父级 IP 传输至 GT 四通道。因此, 仅限确有必要并将所有父级 IP 参数都传输到 GT 四通道后, 才能执行手动更改。

图 21: “Transceiver Wizard” 中的 “Auto” 到 “Manual” 选项切换



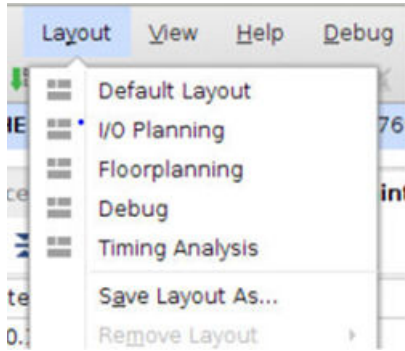
## 串行 I/O 调试

含 GT 的所有设计均可供硬件管理器中的 IBERT 运行时软件访问。创建 GT 设计期间，无需指定 IBERT 专用设置。如需了解设计输入步骤，请参阅 [自定义和生成核](#)。如需了解更多信息，请参阅《Vivado Design Suite 用户指南：编程和调试》(UG908) 中的“串行 I/O 硬件调试流程”章节。

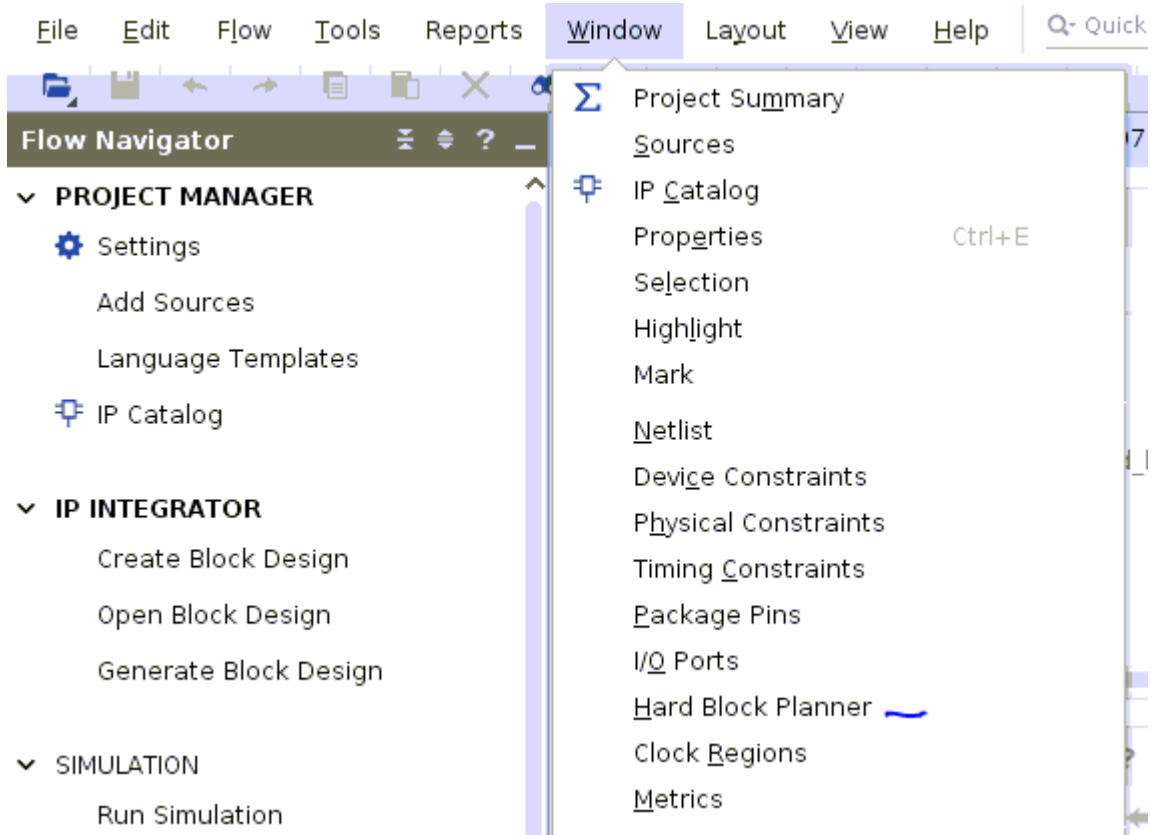
## 设计的综合和实现

创建系统后，执行以下步骤：

1. 确认设计正确性。右键单击 IP integrator 画布，并选择“Validate Design”（确认设计）或者按功能键 F6。
2. 在顶层 xdc 上添加 REFCLK create\_clock 约束。
3. 确认设计并生成顶层文件后，您即可在 Vivado 中单击“Run Synthesis”（运行综合）来进行设计综合。
4. 为 GT 和 refclk 管脚布局打开“synthesized design” → “Layout” → “I/O Planning”（已综合的设计 > 布局 > I/O 管脚分配），如下图所示：



或者，导航至如下“Window Hard Block Planner”（窗口硬核块规划器）：



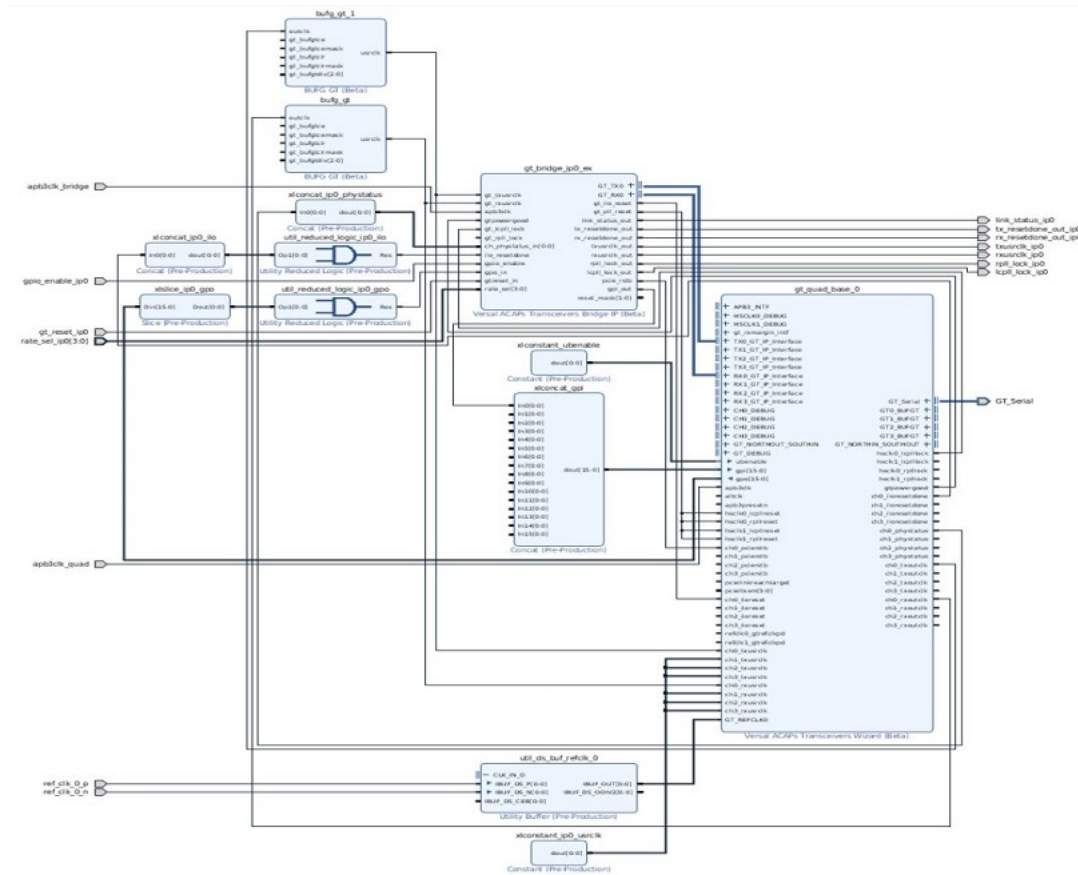
5. 打开“Package Pins”（封装管脚）选项卡，并在对应 MGT bank 中提供 GT 四通道和 GT 参考时钟位置。
6. 分配完所有 I/O 端口后，单击“Run Implementation”（运行实现）即可实现设计。

**注释：**如需了解有关通用 IO 管脚分配和时钟规划准则的更多信息，请参阅《Vivado Design Suite 用户指南：I/O 管脚分配和时钟规划》(UG899)。

### 设计示例

您可生成设计示例，用于对 Transceivers Wizard IP 核进行任意自定义设置。自定义并生成核实例后，您可为该实例选择“Open IP Example Design Vivado® Integrated Design Environment (IDE)”（打开 IP 设计示例 Vivado 集成设计环境）选项。这样向导设计示例会打开一个单独的 Vivado 工程作为顶层模块。此设计示例会例化自定义的核。设计示例会在后台调用 IP integrator 并为给定配置添加 gt\_bridge\_ip 和生成 GT 四通道设计，如下图所示。

图 22: 核设计示例



Wizard IP 设计示例的用途是：

- 通过基于 PRBS 生成器和检查器使用链接状态指示器，提供在仿真中运行的自定义核实例的简单演示。
- 为将自定义核集成到系统（包括参考时钟缓冲器）提供起点。

对于每条收发器通道，gt\_bridge\_ip 都包含可配置的 PRBS 生成器和检查器模块，支持简单的数据完整性测试，并可报告所生成的链接状态。设计示例同样可综合。

**注释：**如果不使用 IO Planner 选择位置，那么您就需要在 IP xdc 中为选定器件提供相应的 GT 四通道和 GTREFCLK 位置。

**注释：**对于 Bridge IP 不提供设计示例。

Bridge IP 具有复位控制器帮助块，可简化串行收发器原语的复位和初始化进程。

---

## 设计示例的限制

我们推荐采用设计示例在您自己的系统环境之外对 Wizard IP 核进行仿真或实现。它非常简单，并且您应知晓它存在的下列限制：

- 设计示例不会实现特定协议来生成或检查数据。从根本上来说，它会生成并检查原始 PRBS 数据。
- 使用提供的测试激励文件来仿真设计示例时，每个收发器通道都会从串行数据发射器环回到接收器。因此，仅当发射器和接收器配置为相同线速率并使用相同数据编码时，才能正确检查数据完整性。不使用任何速率调整方案。如果在您的系统内配置的发射器和接收器线速率或数据编码彼此不同，那么您可以对两个相应自定义的核实例进行交叉耦合，并在硬件或者您自己的测试激励文件中检查数据完整性。在此类设置中，核实例 A 的发射器与核实例 B 的接收器线速率与编码相匹配，且反之亦然。
- 对于多四通道设计，不提供设计示例。

---

## 设计示例仿真

要对 Wizard IP 核的实例进行仿真，请首先打开其设计示例。

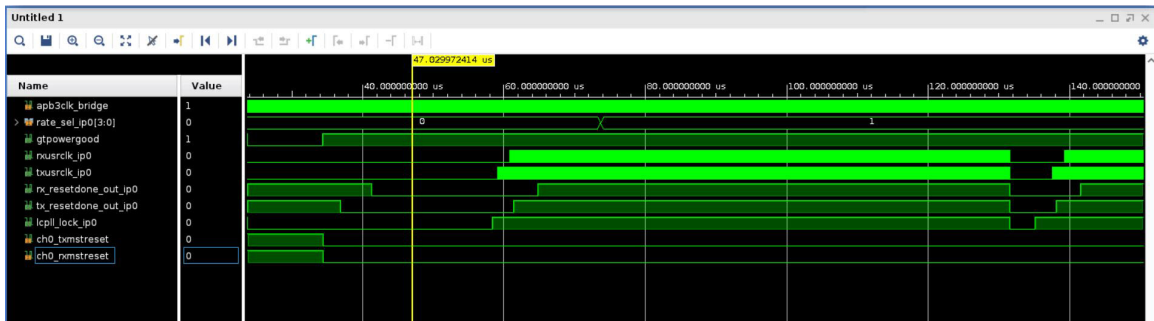
在工程示例中，单击 Vivado 集成设计环境 (IDE) 中的“Run” → “Simulation” → “Run Behavioral Simulation”（运行 > 仿真 > 运行行为仿真）来启动行为仿真。

如需了解 GTME5 整数端口相关仿真更新，请参阅《Vivado Design Suite 教程：逻辑仿真》(UG937) 中的实验 6。

设计示例仿真测试激励文件提供了必要的自由运行的时钟和收发器参考时钟信号，并对设计示例逻辑和复位控制器帮助程序块输入端口提供了 reset all（全部复位）脉冲。此激励足以允许帮助程序块启动系统其余部分。稍后，收发器 PLL 将达成锁定，允许复位控制器帮助程序块有限状态机完成整个复位序列。完成复位序列后，您即可开始观察到激励模块示例发射数据。片刻后，检查模块示例就会开始搜索数据对齐并检查数据完整性，结果将供链接状态逻辑用于驱动链接状态指示器。对于多线速率配置，设计示例会切换为下一个线速率，并再次尝试达成锁定和其它“Reset Done”（复位完成）信号。您可观察到当 \*rate\_sel[3:0] 端口发生切换时，当 \*resetdone 断言高位有效之后，其对应 \*outclk 值会发生更改。



图 23：显示速率更改的仿真波形



tx/rx\_resetdone\_out\_ip0 的默认值为“high”（高电平），直至 gtpowergood 断言高位有效，当 ch\*\_tx/rxmstreset 应用于四通道时，则转至“low”（低电平）。最后，它会测量期望的用户时钟频率，以确保达成期望的线速率。每次速率更改后，都会打印下列消息：

```
Time : 46545 ps PROTO CONFIG2: stauation completed
PROTO RATE CHANGE INITIATED
PROTO CONFIG3::::: PLL LOCKED
PROTO CONFIG3::::: TX RESET DONE asserted
PROTO CONFIG3:::::POST PLL LOCK TXUSRCLK Frequency = 156.25
PROTO CONFIG3::::: TX_USER CLOCK IS PROPER TX_USER_CLOCK POST PLL LOCK is 156.25 and EXPECTED is [154.688:157.812]
PROTO CONFIG3::::: RX RESET DONE asserted
PROTO CONFIG3:::::POST PLL LOCK RXUSRCLK Frequency = 156.25
PROTO CONFIG3::::: RX_USER CLOCK IS PROPER RX_USER_CLOCK POST PLL LOCK is 156.25 and EXPECTED is [154.688:157.812]
run: Time (s): CPU = 00:00:18 ; elapsed = 00:01:48 . Memory (MB): peak = 7054.980 ; gain = 0.000 ; free physical = 46634 ; free virtual = 115817
```

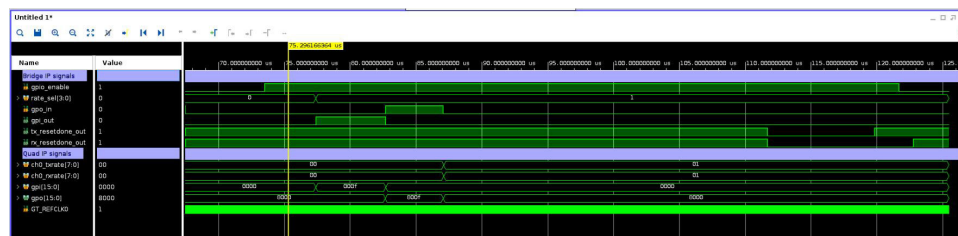
最后，会在转录文本中打印以下消息，并判定测试成功。

```
PROTO POST CONFIG3 LINK STABLE TXUSRCLK Frequency = 156.25
PROTO CONFIG3::::: POST LINK STABLE RXUSRCLK Frequency = 156.25
Time : 172150 ps PROTO CONFIG3: simulation completed
Time : 172150 ps PASS: simulation completed
** Test completed successfully
```

## gpio\_enable 端口用法

当多线速率 IP 从某一个线速率切换为另一个线速率（其中含新的参考时钟值，但参考时钟源相同）时，即可使用 gpio\_enable 端口。例如，某个 IP 配置有 2 个线速率，第一个配置含 line\_rate0 和 REFCLK0，第二个配置含 line\_rate1 和 REFCLK1。在此例中，当某个 IP 从 line\_rate0 切换为 line\_rate1 时，参考时钟值也会随之更改。在此情况下，您需要断言 gpio\_enable 信号有效、切换至新的参考时钟频率，并确保它保持稳定，然后切换 \*rate\_sel port。切换 \*rate\_sel port 时，Bridge IP 会向 GT 四通道生成 gpi 信号，并处理 gpi/gpo 排序，无需用户干预。当 \*resetdone 断言高电平有效后，gpio\_enable 即可断言无效。下图显示了在此类情况下的 gpio\_enable 用法：

图 24：对应线速率和 REFCLK 频率更改的仿真波形



## 调试

本附录包含有关赛灵思支持网站和调试工具上可用资源的详细信息。

如果 IP 需要许可证密钥，则必须验证密钥。Vivado® 设计工具设置有多个许可证检查点，旨在通过此流程对获得许可的 IP 进行门控。如果许可证检查成功，则可以继续生成 IP。否则，系统会因错误而停止生成。许可证检查点由以下工具强制执行：

- Vivado 综合
- Vivado 实现
- write\_bitstream (Tcl 命令)



**重要提示！** 在检查点处忽略 IP 许可证级别。测试会确认是否存在有效的许可证。它不检查 IP 许可证级别。

## 在 Xilinx.com 上寻求帮助

为了帮助您在使用核时完成设计和调试进程，[赛灵思技术支持网页](#)上提供了大量关键资源，如产品文档、版本说明、答复记录、已知问题相关信息以及如何获取进一步产品支持的链接。[赛灵思社区论坛](#)还可供会员学习、参与、分享和提出与赛灵思解决方案相关的问题。

### 文档

本产品指南是与该核相关的主要文档。本指南以及有助于设计进程的所有产品相关文档都可以在[赛灵思技术支持网页](#)上找到，也可以通过赛灵思 Documentation Navigator 来获取。要下载赛灵思 Documentation Navigator，请访问[下载页面](#)。如需了解此工具和可用功能的详细信息，请在安装后打开联机帮助。

### 解决方案中心

如需获取设计周期各阶段内与器件、软件工具和 IP 等相关的技术支持，请参阅[赛灵思解决方案中心](#)。相关专题包括设计辅助、设计咨询和故障排除提示等。

以下列出了 Versal ACAP Transceivers Wizard 核专用的解决方案中心。

- [赛灵思高速串行 I/O 解决方案中心](#)
- [赛灵思 HSSIO 解决方案中心 - 设计辅助串行收发器调试](#)
- [串行收发器板块](#)

## 答复记录

答复记录包括有关常见问题的信息、有关如何解决这些问题的实用信息以及有关赛灵思产品的所有已知问题。我们每天都会创建和维护答复记录，确保用户可以获取最准确的信息。

您可以通过[赛灵思技术支持网页](#)（主页）上的“搜索支持”框找到对应该核的答复记录。要最大程度扩展搜索结果范围，请使用关键字，例如：

- 产品名称
- 工具消息
- 所遇到问题的摘要

返回结果后，可以使用过滤器搜索来进一步定位结果。

## Versal ACAP Transceivers Wizard 的主答复记录

答复记录 [75716](#)。

## 技术支持

赛灵思在[赛灵思社区论坛](#)上为此 LogiCORE™ IP 产品提供技术支持，前提是用户按产品文档中所述方式使用该产品。如果您执行以下任何操作，则赛灵思无法保证产品时序和功能的正常运行，也无法保证提供相应支持：

- 在文档中未定义的器件中实现解决方案。
- 超出产品文档中允许的范围自定义解决方案。
- 更改设计中任何标记有“DO NOT MODIFY”的部分。

如需提问，请导航至[赛灵思社区论坛](#)。

---

## 调试工具

有许多工具可用于解决 Versal ACAP Transceivers Wizard 设计问题。至关重要的是要了解哪些工具可用于调试各种情况。

## Vivado Design Suite 调试功能

Vivado® Design Suite 调试功能可以将逻辑分析器和虚拟 I/O 核直接插入到您的设计中。调试功能还支持您设置触发条件，以便在硬件中捕获应用和集成块端口信号。随后，您便可对捕获的信号进行分析。Vivado IDE 中的这个功能用来对在赛灵思器件中运行的设计进行逻辑调试和确认。

Vivado 逻辑分析器用于与下列逻辑调试 LogiCORE IP 核交互：

- ILA 2.0（及更高版本）
- VIO 2.0（及更高版本）

请参阅《Vivado Design Suite 用户指南：编程和调试》([UG908](#))。

# 附加资源与法律声明

## 赛灵思资源

如需获取答复记录、技术文档、下载以及论坛等支持性资源，请参阅[赛灵思技术支持](#)。

## Documentation Navigator 与设计中心

赛灵思 Documentation Navigator (DocNav) 提供了访问赛灵思文档、视频和支持资源的渠道，您可以在其中筛选搜索信息。要打开 DocNav，请执行以下操作：

- 在 Vivado® IDE 中，单击“Help” → “Documentation and Tutorials”。
- 在 Windows 中，单击“Start” → “All Programs” → “Xilinx Design Tools” → “DocNav”。
- 在 Linux 命令提示中输入 `docnav`。

赛灵思设计中心提供了根据设计任务和其它主题整理的文档链接，可供您用于了解关键概念以及常见问题解答。要访问设计中心，请执行以下操作：

- 在 DocNav 中，单击“Design Hubs View”选项卡。
- 在赛灵思网站上，查看[设计中心](#)页面。

**注释：**如需了解有关 DocNav 的更多信息，请参阅赛灵思网站上的 [Documentation Navigator](#) 页面。

## 参考资料

以下技术文档是非常实用的补充资料，可配合本指南一起使用：

1. 《Vivado Design Suite 用户指南：采用 IP integrator 设计 IP 子系统》(UG994)
2. 《Vivado Design Suite 用户指南：采用 IP 进行设计》(UG896)
3. 《Vivado Design Suite 用户指南：入门指南》(UG910)
4. 《Vivado Design Suite 用户指南：逻辑仿真》(UG900)
5. 《Vivado Design Suite 用户指南：I/O 管脚分配和时钟规划》(UG899)
6. 《Vivado Design Suite 用户指南：编程和调试》(UG908)

7. 《Versal ACAP GTY 和 GTYP 收发器架构手册》(AM002)
8. 《Vivado Design Suite 教程：逻辑仿真》(UG937)
9. 《Versal ACAP GTM 收发器架构手册》(AM017)

## 修订历史

下表列出了本文档的修订历史。

| 章节   | 修订综述   |
|--|--|
| <b>2022 年 10 月 19 日 1.1 版</b>  |  |
| <a href="#">复位控制器帮助程序块</a>   | 在 <a href="#">第 5 章：设计示例</a> 中新增章节   |
| <a href="#">“Sub GUI Configuration” 选项卡</a>  | 更新 <a href="#">第 4 章：设计流程步骤</a> 中的章节   |
| <a href="#">GT 参考时钟汇总与最优化</a>  |  |
| <b>2022 年 4 月 29 日 1.1 版</b>   |  |
| <a href="#">第 3 章：产品规格</a>   | 更新《Versal ACAP GTM 收发器架构手册》(AM017) 链接  |
| <b>2022 年 4 月 26 日 1.1 版</b>   |  |
| <ul style="list-style-type: none"> <li>· <a href="#">第 3 章：产品规格</a></li> <li>· <a href="#">适用于定制 IP 的 IP integrator 设计输入</a></li> <li>· <a href="#">“Sub GUI Configuration” 选项卡</a></li> <li>· <a href="#">设计示例仿真</a></li> </ul> | 更新章节。  |
| <b>2021 年 6 月 16 日 1.1 版</b>   |  |
| 常规更新   | 更新 GTYP 和 GTM 的参考  |
| <b>2020 年 11 月 23 日 1.1 版</b>  |  |
| 常规更新   | <ul style="list-style-type: none"> <li>· 更新 <a href="#">第 3 章：产品规格</a></li> <li>· 更新 <a href="#">自定义和生成核</a></li> <li>· 更新 <a href="#">共享多个 Bridge IP 的 GT 四通道</a></li> <li>· 新增 <a href="#">赛灵思 IP - GT 四通道集成</a></li> <li>· 更新 <a href="#">第 5 章：设计示例</a></li> </ul> |
| <b>2020 年 7 月 14 日 1.0 版</b>   |  |
| 初始版本   | 不适用  |

## 请阅读：重要法律声明

本文向贵司/您所提供的信息（下称“资料”）仅在对赛灵思产品进行选择和使用参考。在适用法律允许的最大范围内：(1) 资料均按“现状”提供，且不保证不存在任何瑕疵，赛灵思在此声明对资料及其状况不作任何保证或担保，无论是明示、暗示还是法定的保证，包括但不限于对适销性、非侵权性或任何特定用途的适用性的保证；且 (2) 赛灵思对任何因资料发生的或与资料有关的（含对资料的使用）任何损失或赔偿（包括任何直接、间接、特殊、附带或连带损失或赔偿，如数据、利润、商誉的损失或任何因第三方行为造成的任何类型的损失或赔偿），均不承担责任，不论该等损失或者赔偿是何种类或性质，也不论是基于合同、侵权、过失或是其它责任认定原理，即便该损失或赔偿可以合理预见或赛灵思事前被告知有发生该损失或赔偿的可能。赛灵思无义务纠正资料中包含的任何错误，也无义务对资料或产品说

明书发生的更新进行通知。未经赛灵思公司的事先书面许可，贵司/您不得复制、修改、分发或公开展示本资料。部分产品受赛灵思有限保证条款的约束，请参阅赛灵思销售条款：<https://china.xilinx.com/legal.htm#tos>；IP 核可能受赛灵思向贵司/您签发的许可证中所包含的保证与支持条款的约束。赛灵思产品并非为故障安全保护目的而设计，也不具备此故障安全保护功能，不能用于任何需要专门故障安全保护性能的用途。如果把赛灵思产品应用于此类特殊用途，贵司/您将自行承担风险和责任。请参阅赛灵思销售条款：<https://china.xilinx.com/legal.htm#tos>。

本文档包含一些初步信息，该等信息可能会在不另行通知的情况下有所变化。此处提供的信息与当前尚未公开发售的产品和/或服务有关，并且仅供参考，并非旨在用作为对此处引用的产品和/或服务进行销售或尝试性商业化。

### 关于与汽车相关用途的免责声明

如将汽车产品（部件编号中含“XA”字样）用于部署安全气囊或用于影响车辆控制的应用（“安全应用”），除非有符合 ISO 26262 汽车安全标准的安全概念或冗余特性（“安全设计”），否则不在质保范围内。客户应在使用或分销任何包含产品的系统之前为了安全的目的全面地测试此类系统。在未采用安全设计的条件下将产品用于安全应用的所有风险，由客户自行承担，并且仅在适用的法律法规对产品责任另有规定的情况下，适用该等法律法规的规定。

### 版权声明

© 2020-2022 年 Advanced Micro Devices, Inc. 版权所有。Xilinx、赛灵思徽标、Alveo、Artix、Kintex、Kria、Spartan、Versal、Vitis、Virtex、Vivado、Zynq 及本文提到的其它指定品牌均为赛灵思在美国及其它国家或地区的商标。“PCI”、“PCIe”和“PCI Express”均为 PCI-SIG 拥有的商标，且经授权使用。所有其它商标均为各自所有方所属财产。