Versal Adaptive SoC Transceivers Wizard v1.1

LogiCORE IP 产品指南

Vivado Design Suite

PG331 (v1.1) 2023 年 10 月 24 日

目录

第 1 章

简介

AMD Versal™ Adaptive SoC GTY, GTYP, and GTM Transceivers Wizard IP 解决方案旨在帮助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

该向导的关键功能特性如下所示：

- 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。
- 定制 IP 的设计输入是通过 IP integrator (IPI) 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

AMD LogiCORE™ IP 相关信息表	
核规格	
支持的器件系列	AMD Versal™ 自适应 SoC
支持的用户接口	不适用
资源	· Versal 自适应 SoC GTM 寄存器映射：“性能与资源使用情况”网页 · Versal 自适应 SoC GTY 和 GTYP 寄存器映射：“性能与资源使用情况”网页
随核提供	
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
经过测试的设计流程	
设计输入	AMD Vivado™ Design Suite
仿真	如需了解有关受支持的仿真器的信息，请参阅《Vivado Design Suite 用户指南：版本说明、安装和许可》(UG973)。
综合	Vivado 综合
支持	
版本说明和已知问题	主答复记录：75716
所有 Vivado IP 更改日志	Vivado IP 主更改日志：72775
支持网页	

注释：

1. 如需获取受支持的器件的完整列表，请参阅 AMD Vivado™ IP 目录。
2. 如需获取受支持的工具版本，请参阅《Vivado Design Suite 用户指南：版本说明、安装和许可》(UG973)。

第 2 章

概述

按设计进程浏览内容

AMD 自适应计算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。您可以在设计中心页面上访问 AMD Versal™ 自适应 SoC 设计进程。您还可以使用设计流程助手来更深入了解设计流程，并找到特定于预期设计需求的内容。

- 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 AMD 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

根据最终用户许可条款，本 AMD LogiCORE™ IP 模块随附 AMD Vivado™ Design Suite 免费提供。

有关其他 AMD LogiCORE™ IP 模块的信息，请访问知识产权页面。如需了解有关其他 AMD LogiCORE IP 模块和工具的定价和可用性信息，请联系您当地的销售代表。

第 3 章

产品规格

AMD Versal™ Adaptive SoC Transceivers Wizard 解决方案包含 2 个核：

- Versal Adaptive SoC 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，需要例化多个 Transceivers Wizard。
  - GT Quad base IP：在 AMD Vivado™ IP 目录中包含 GT Quad base IP，它可作为 Versal Adaptive SoC Transceivers Wizard (`gt_quad_base`) 以供使用。它用于例化、配置和连接 `GT*_QUAD` 原语，并提供仿真支持，以展示各种 GT 四通道功能特性。您可配置 GT Quad base IP 以共享多个协议 IP，每个协议 IP 都支持多种线速率；还可创建设计示例并进行仿真，以了解多个协议 IP 之间的动态线速率切换和 GT 四通道共享。如需获取此 IP 的端口列表和定义，请参阅《Versal 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal 自适应 SoC GTM 收发器架构手册》(AM017)。
- Versal Adaptive SoC Transceivers Bridge：这是参考父级 IP (Bridge IP)，用于配置 Transceivers Wizard 参数。
  - Bridge IP：在 IP integrator 画布中包含 Versal Adaptive SoC 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 有两种工作模式：

    “Default Mode”（默认模式）：此模式用于在 FPGA 开发板上进行独立的 Transceivers Wizard 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 参数后，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 Adaptive SoC Transceiver Bridge IP (`gt_bridge_ip`)，并在 GUI 中配置其参数（包括通道数量），完成后单击“Block Automation”（块自动化设置）。

    注释： `gt_quad_base` 参数继承自 `gt_bridge_ip`。

    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**

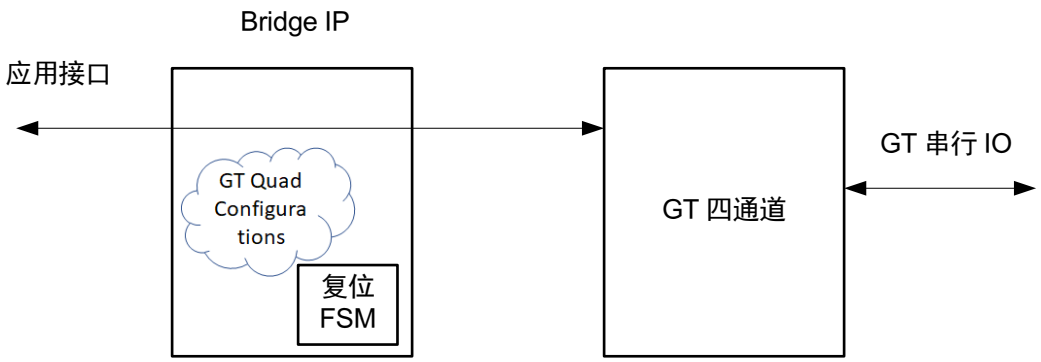


图 2：**单个 Bridge IP - GT 四通道设计**

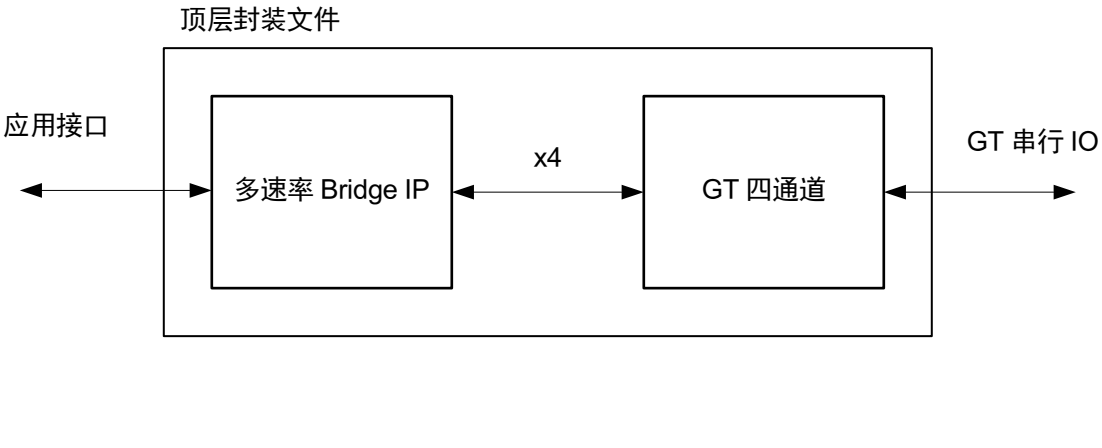
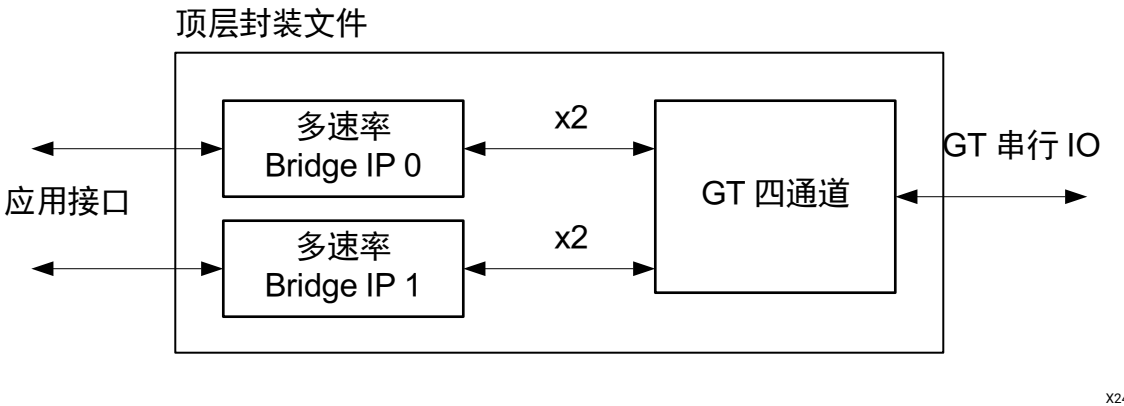


图 3：**多个 Bridge IP 共享 GT 四通道**



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设配置
  - 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。 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。 此信号连接到复位控制器帮助程序块的 gtwiz_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。 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。 此信号连接到复位控制器帮助程序块的 gtwiz_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。 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。 此信号连接到复位控制器帮助程序块的 gtwiz_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。 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。 此信号连接到复位控制器帮助程序块的 gtwiz_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。 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。 此信号连接到复位控制器帮助程序块的 gtwiz_reset_tx_datapath_in 信号。
tx_resetdone_out	输出	1	apb3clk	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。
rx_resetdone_out	输出	1	apb3clk	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 gpio_enable 端口用法。

表 1：**直通模式下的 Bridge IP 端口** (续)

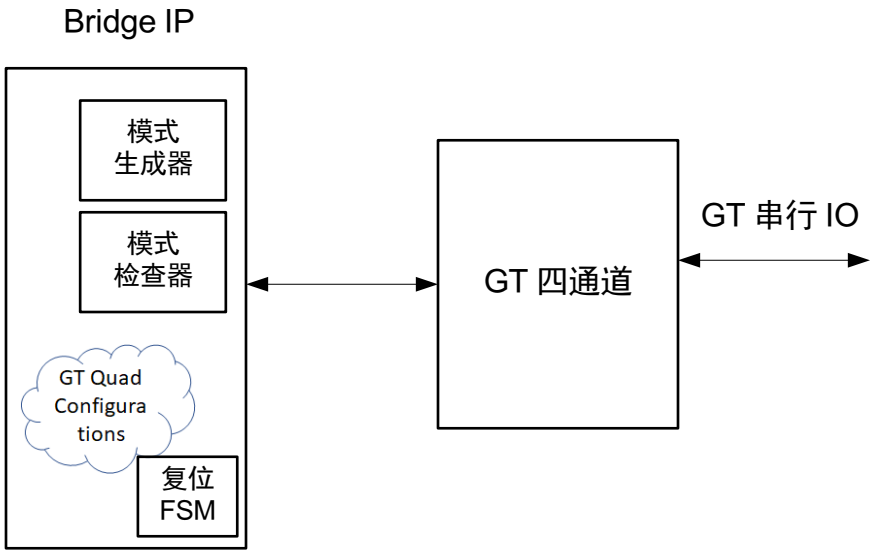
名称	方向	宽度	时钟域	描述
gpo_in	输入	1	apb3clk	与 gpio_enable 搭配使用。在块自动化设置期间，会自动连接此信号。不使用此信号时，需要驱动它。
gpi_out	输出	1	apb3clk	与 gpio_enable 搭配使用。在块自动化设置期间，会自动连接此信号。
rate_sel[3:0]	输入	4	apb3clk	此输入速率配置端口用于控制 GT 四通道的 ch*_txrate[7:0] 端口和 ch*_rxrate[7:0] 端口。输入宽度为 4 位，对应于可在 Bridge/GT Quad IP 输入级别采用的配置输入的数量 (16)。
link_status_out	输出	1	apb3clk	这是一个无关紧要的信号，启用直通模式时可忽略此信号。禁用直通模式选项时，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。
gt_tx_usrclk	输入	1	-	连接到收发器原语上的 TXUSRCLK。
gt_rx_usrclk	输入	1	-	连接到收发器原语上的 RXUSRCLK。
txusrclk_out	输出	1	-	要用于用户逻辑的发射器时钟输出。 在内部绑定到 Bridge IP 的 gt_txusrclk 输入。
rxusrclk_out	输出	1	-	要用于用户逻辑的接收器时钟输出。 在内部绑定到 Bridge IP 的 gt_rxusrclk 输入。
tx_clr_out	输出	1	apb3clk	此高电平有效信号扇出到 BUFG_GT 或 MBUFG_GT 原语的 CLR 端口。
rx_clr_out	输出	1	apb3clk	此高电平有效信号扇出到 BUFG_GT 或 MBUFG_GT 原语的 CLR 端口。
tx_clrb_leaf_out	输出	1	apb3clk	此低电平有效信号扇出到 MBUFG_GT 原语的 CLRB_LEAF 端口。
rx_clrb_leaf_out	输出	1	apb3clk	此低电平有效信号扇出到 MBUFG_GT 原语的 CLRB_LEAF 端口。

注释：

1. 可以忽略 gt_lcpll_lock、gt_rpll_lock、ch_phystatus_in 和 ilo_resetdone 输入。
2. gt_ilo_reset、gt_pll_reset、rpll_lock_out、lcpll_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_lcplllock、hsclk1_lcplllock、hsclk0_rplllock 和 hsclk1_rplllock 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。如需了解有关复位序列在内部的发生方式的更多信息，请参阅《Versal 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal 自适应 SoC GTM 收发器架构手册》(AM017)。

如果在直通模式下未配置 Bridge IP，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**



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

表 2：**Quad IP 中的通道绑定端口** (续)

名称	方向	宽度	时钟域	描述
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更多信息，请参阅《Versal 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal 自适应 SoC GTM 收发器架构手册》(AM017)。

复位控制器帮助程序块

您必须为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 `gtwiz_reset_clk_freerun_in` 时钟。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	IN	1	异步	自由运行的时钟，用于复位收发器原语。应在器件配置之前进行切换。
gtwiz_reset_all_in	IN	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	IN	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	IN	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	IN	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。

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_rx_datapath_in	IN	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	IN	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	IN	1	异步	对于 GTY 和 GTYP：由收发器通道原语生成的所有 TXPMARESETDONE 信号的逻辑 AND。 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	IN	1	异步	对于 GTY 和 GTYP：由收发器通道原语生成的所有 RXPMARESETDONE 信号的逻辑 AND。 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	IN	1	异步	由收发器通道原语生成的所有 MSTTXRESETDONE 信号的逻辑 AND。
mst_tx_resetdone	IN	1	异步	由收发器通道原语生成的所有 MSTRXRESETDONE 信号的逻辑 AND。
mst_tx_reset	OUT	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	OUT	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	OUT	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	OUT	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	OUT	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	OUT	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	OUT	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。

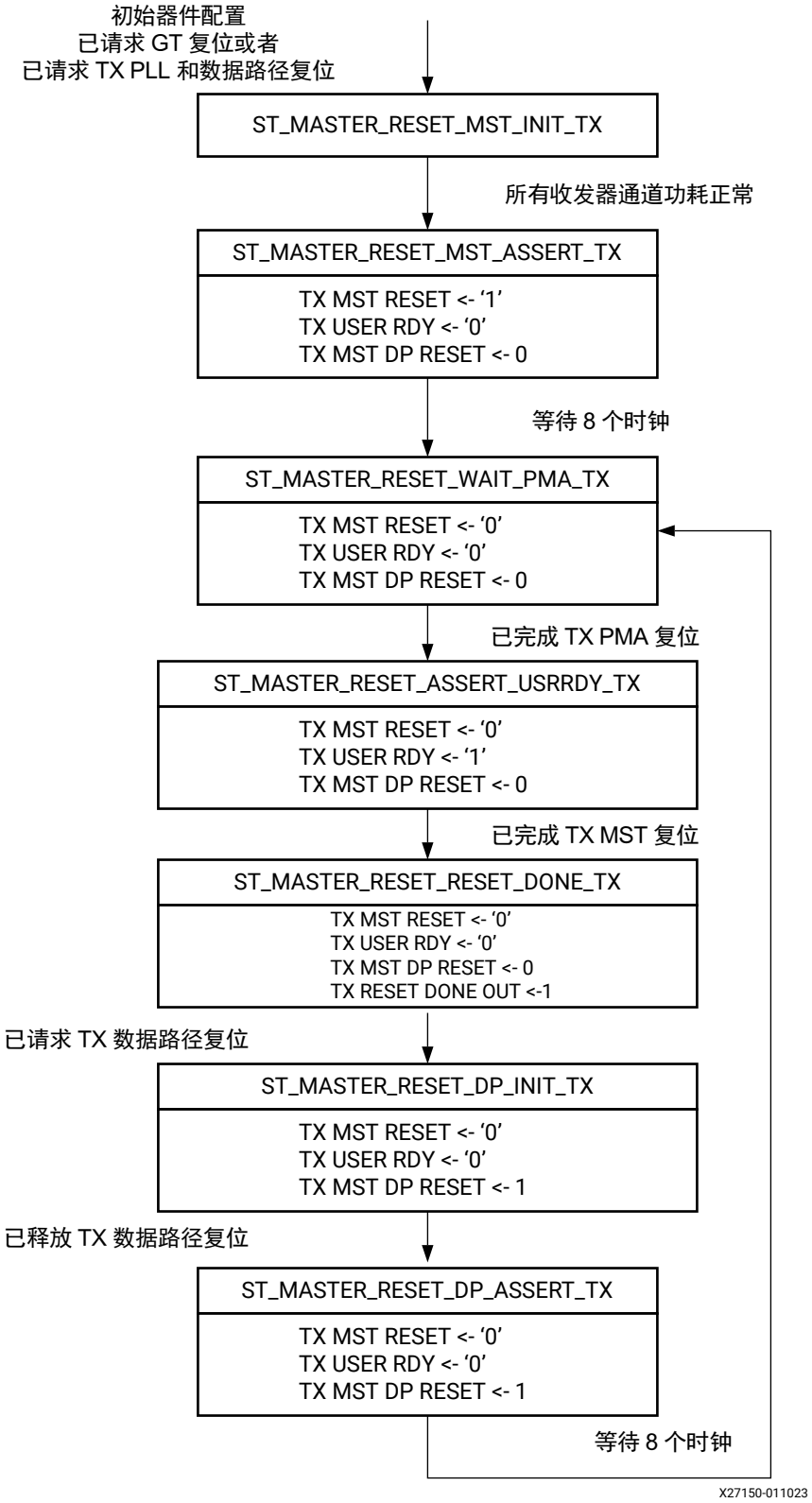
表 3：**复位控制器帮助程序块端口** (续)

名称	方向	宽度	时钟域	描述
gtwiz_reset_rx_done_out	OUT	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。

帮助程序块遵循控制器复位序列，并包含 2 个状态机。

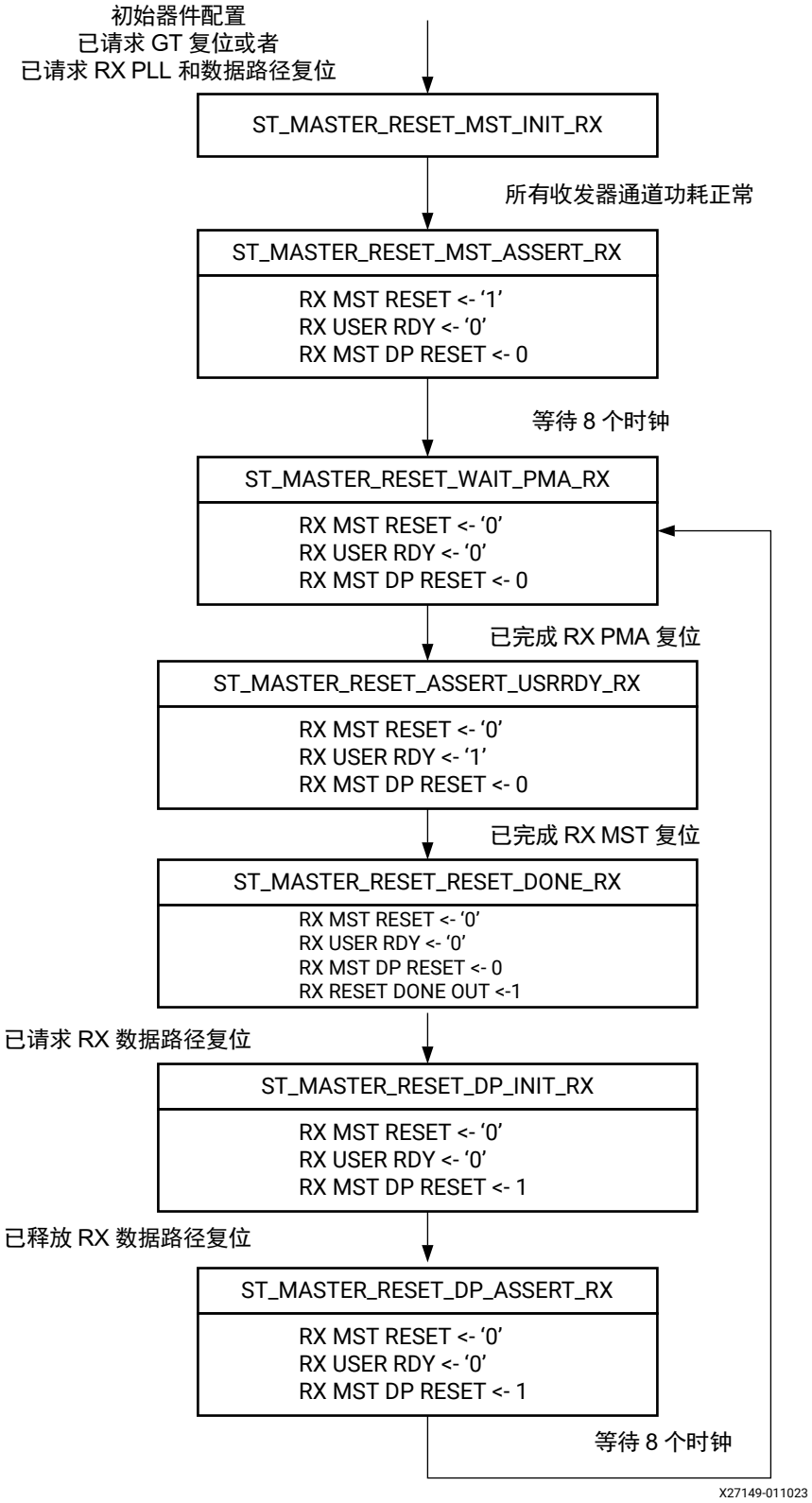
- 发射器复位状态机：用于复位所有收发器原语的发射器 PLL 和/或发射器数据路径，并指示其完成状态。

图 5：**发射器复位状态机**


初始器件配置
已请求 GT 复位或者
已请求 TX PLL 和数据路径复位
ST_MASTER_RESET_MST_INIT_TX
所有收发器通道功耗正常
ST_MASTER_RESET_MST_ASSERT_TX
TX MST RESET <- '1'
TX USER RDY <- '0'
TX MST DP RESET <- 0
等待 8 个时钟
ST_MASTER_RESET_WAIT_PMA_TX
TX MST RESET <- '0'
TX USER RDY <- '0'
TX MST DP RESET <- 0
已完成 TX PMA 复位
ST_MASTER_RESET_ASSERT_USRRDY_TX
TX MST RESET <- '0'
TX USER RDY <- '1'
TX MST DP RESET <- 0
已完成 TX MST 复位
ST_MASTER_RESET_RESET_DONE_TX
TX MST RESET <- '0'
TX USER RDY <- '0'
TX MST DP RESET <- 0
TX RESET DONE OUT <-1
已请求 TX 数据路径复位
ST_MASTER_RESET_DP_INIT_TX
TX MST RESET <- '0'
TX USER RDY <- '0'
TX MST DP RESET <- 1
已释放 TX 数据路径复位
ST_MASTER_RESET_DP_ASSERT_TX
TX MST RESET <- '0'
TX USER RDY <- '0'
TX MST DP RESET <- 1
等待 8 个时钟
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**


初始器件配置
已请求 GT 复位或者
已请求 RX PLL 和数据路径复位
ST_MASTER_RESET_MST_INIT_RX
所有收发器通道功耗正常
ST_MASTER_RESET_MST_ASSERT_RX
RX MST RESET <- '1'
RX USER RDY <- '0'
RX MST DP RESET <- 0
等待 8 个时钟
ST_MASTER_RESET_WAIT_PMA_RX
RX MST RESET <- '0'
RX USER RDY <- '0'
RX MST DP RESET <- 0
已完成 RX PMA 复位
ST_MASTER_RESET_ASSERT_USRRDY_RX
RX MST RESET <- '0'
RX USER RDY <- '1'
RX MST DP RESET <- 0
已完成 RX MST 复位
ST_MASTER_RESET_RESET_DONE_RX
RX MST RESET <- '0'
RX USER RDY <- '0'
RX MST DP RESET <- 0
RX RESET DONE OUT <-1
已请求 RX 数据路径复位
ST_MASTER_RESET_DP_INIT_RX
RX MST RESET <- '0'
RX USER RDY <- '0'
RX MST DP RESET <- 1
已释放 RX 数据路径复位
ST_MASTER_RESET_DP_ASSERT_RX
RX MST RESET <- '0'
RX USER RDY <- '0'
RX MST DP RESET <- 1
等待 8 个时钟
X27149-011023

gtwiz_reset_all_in 输入会启动发射器和接收器状态机。发射器和接收器复位状态机相互独立。每个状态机都可根据需要直接通过用户界面启动，或者通过 gtwiz_reset_all_in 输入控制启动操作。

激活 gtwiz_reset_all_in 信号时，会同时启动发射器和接收器状态机。如果通过设置通道配置以使接收器传入数据依赖于发射器数据，则接收器应在 gtwiz_reset_all_in 完成后执行单独的数据路径复位，或按顺序先使用 reset_tx_pll_and_datapath，而后 reset_rx_datapath。

复位状态机

发射器和接收器复位状态机各有两个入口点：

- 其中一个入口点会先复位关联的 PLL，然后复位数据路径。
- 另一个入口点只会复位数据路径。

对 gtwiz_reset_tx_pll_and_datapath_in 输入进行脉冲时，发射器复位状态机会先启动 PLL 复位，然后启动发射器数据路径复位。由用于为发射器数据路径进行时钟设置的 IP 实例所例化的所有 PLL（LCPLL 或 RPLL 类型）都将复位，以响应此输入。在所有这些 PLL 锁定后，所有收发器原语的发射器可编程分频器和数据路径都将复位。如果不需要 PLL 复位，那么对 gtwiz_reset_tx_datapath_in 输入进行脉冲时，将启动仅限发射器数据路径的复位。无论使用哪一个复位入口点，在完成所有收发器原语的发射器复位序列后，gtwiz_reset_tx_done_out 指示器都会与自由运行的 gtwiz_reset_clk_freerun_in 时钟保持同步断言有效。同样，对 gtwiz_reset_rx_pll_and_datapath_in 输入进行脉冲时，接收器复位状态机会先启动 PLL 复位，然后启动接收器数据路径复位。由用于为接收器数据路径进行时钟设置的 IP 实例所例化的所有 PLL（LCPLL 或 RPLL 类型）都将复位，以响应此输入。在所有这些 PLL 锁定后，所有收发器原语的接收器数据路径都将复位。如果不需要 PLL 复位，则当对 gtwiz_reset_rx_datapath_in 输入进行脉冲时，将启动仅限接收器数据路径的复位。无论使用哪一个复位入口点，在完成所有收发器原语的接收器复位序列后，gtwiz_reset_rx_done_out 指示器都会与自由运行的 gtwiz_reset_clk_freerun_in 时钟保持同步断言有效。

重要提示！

独立的发射器和接收器复位状态机十分简单实用。但是，由于可在发射器数据路径和接收器数据路径之间共享 PLL，因此必须理解在使用 gtwiz_reset_tx_pll_and_datapath_in 和 gtwiz_reset_rx_pll_and_datapath_in 输入时潜在的系统影响。例如，如果发射器和接收器的数据路径都由 LCPLL 资源进行时钟设置，则这两个输入中的任何一个断言有效都会复位每个收发器四通道的共享 LCPLL，从而导致另一个数据方向上可能出现意外的链路中断。应谨慎使用这些输入，与其他核实例共享 PLL 资源时尤其如此。可使用 gtwiz_reset_all_in 输入以避免此类冗余的 PLL 复位序列。

复位排序和其他服务

发射器和接收器复位状态机会实现《AMD Versal™ 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002) 或《Versal 自适应 SoC GTM 收发器架构手册》(AM017) 中指定的相关主复位序列。复位控制器帮助程序块的收发器接口连接到收发器原语。根据器件配置，在报告收发器电源处于正常状态之前，不应将任何复位帮助程序块复位输入断言有效。复位控制器帮助程序块在内部将所有 PLL 和数据路径资源保持复位状态，直至 GT 四通道的 GTPOWERGOOD 转为“High”（高电平）为止，然后通过发射器和接收器状态机进行一次转换来复位所有收发器资源。因此，在尝试任何类型的后续复位之前，应等待 GT 四通道 IP 上的 gtpowergood 端口初始断言有效，或者等待 gtwiz_reset_tx_done_out 和 gtwiz_reset_tx_done_out 初始断言有效。

第 4 章

设计流程步骤

本章节描述了该核的自定义和生成方式、该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 AMD 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 AMD 工具在 AMD Vivado™ Design Suite 中自定义和生成该核的信息。

如果要在 Vivado IP integrator 中自定义和生成该核，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。

如 第 3 章：产品规格 中所述，AMD Versal™ Adaptive SoC Transceivers Wizard 解决方案包含 2 个核：

1. Versal Adaptive SoC 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 Adaptive SoC 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。AMD 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该 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。

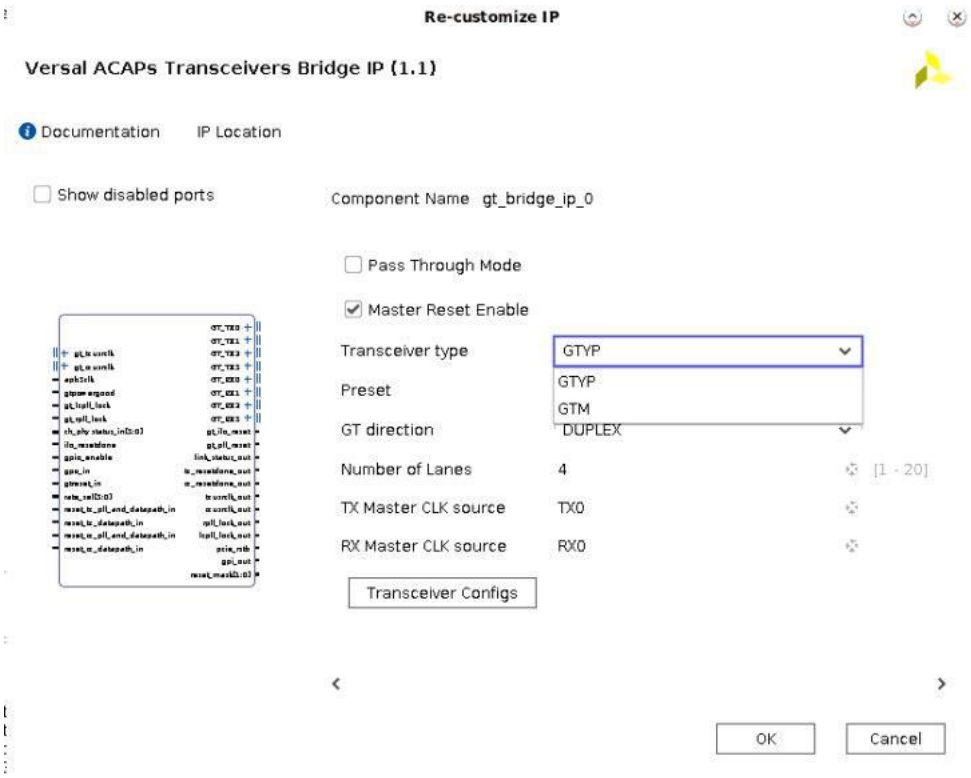
适用于定制 IP 的 IP integrator 设计输入

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 中。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 中添加 `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他参数（`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**

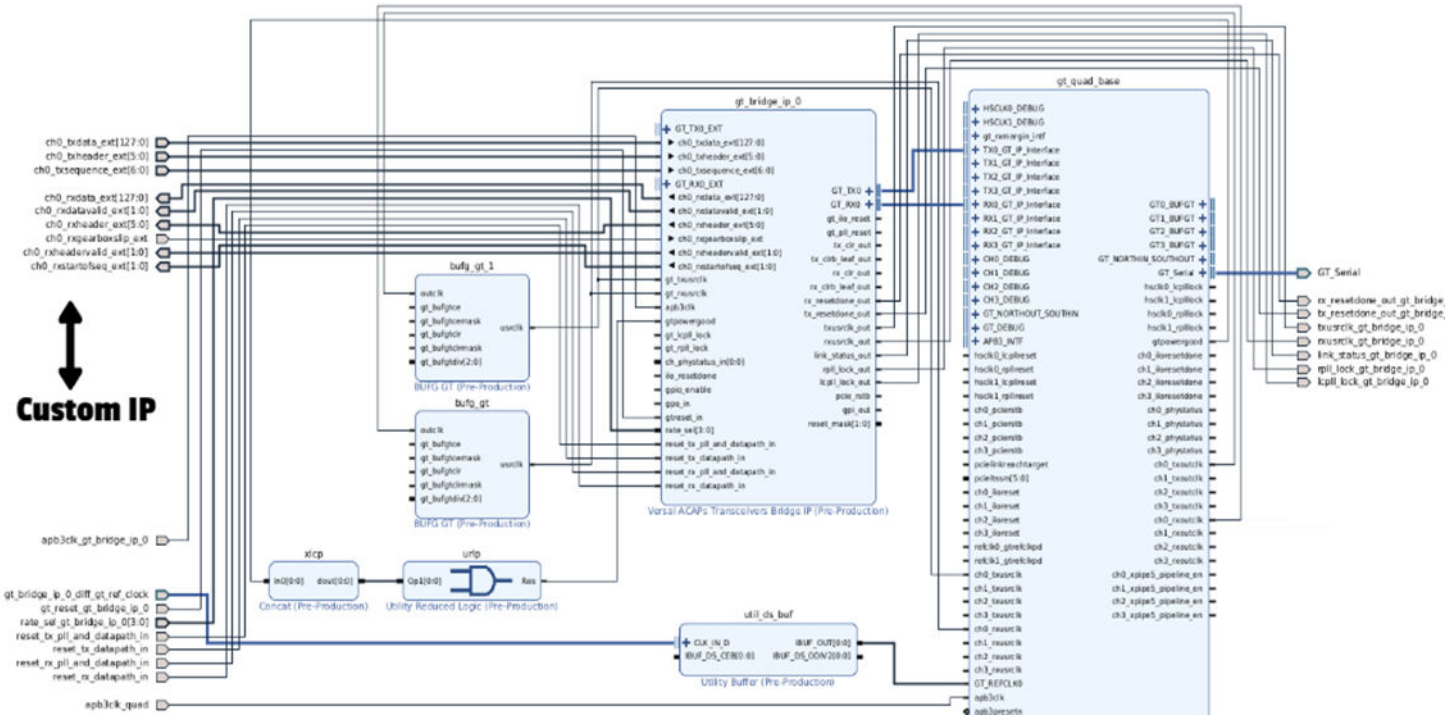


- “Component Name”（组件名称）：在“Component Name”字段中设置生成的 IP 的名称。默认名称是 `gt_bridge_ip_0`。
- “Pass Through Mode”（直通模式）：选择该选项即可显示用户接口的必需 GT 并行接口。
- “Master Reset Enable”（主复位启用）：在 GT 四通道中启用主复位逻辑。默认情况下，已启用该选项。请勿禁用该选项。
- “Preset”（预设）：支持标准协议预设。选择预设即可加载子 GUI 中对应的收发器设置。在此“Preset”窗口中可加载多线速率预设。它将填充对应子 GUI 中的多个单线速率预设。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 56Gbps 时，四通道上添加的通道数量将作为活动的 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）：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。

下图展示了在直通模式下使用块自动化设置为 Bridge IP 生成的设计。

图 8：**直通模式下的 Bridge IP 连接**

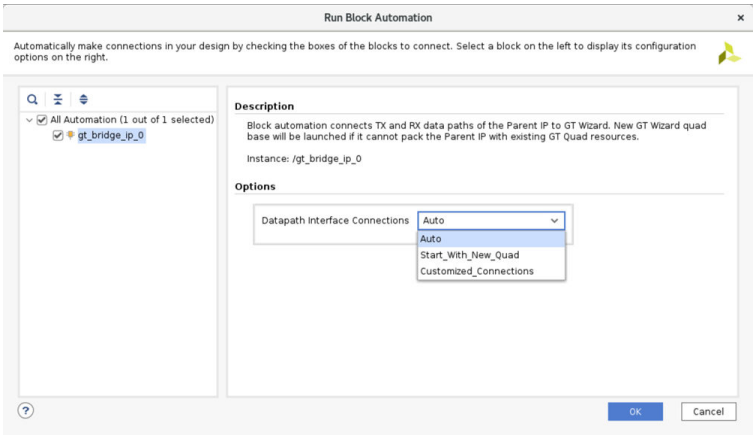


创建设计后，将 Bridge IP 上定制 IP 所需的所有基本端口都布局在块设计的外部。在块设计上创建 HDL 封装文件，并将 Bridge IP 信号传输至定制 IP。

块自动化设置流程

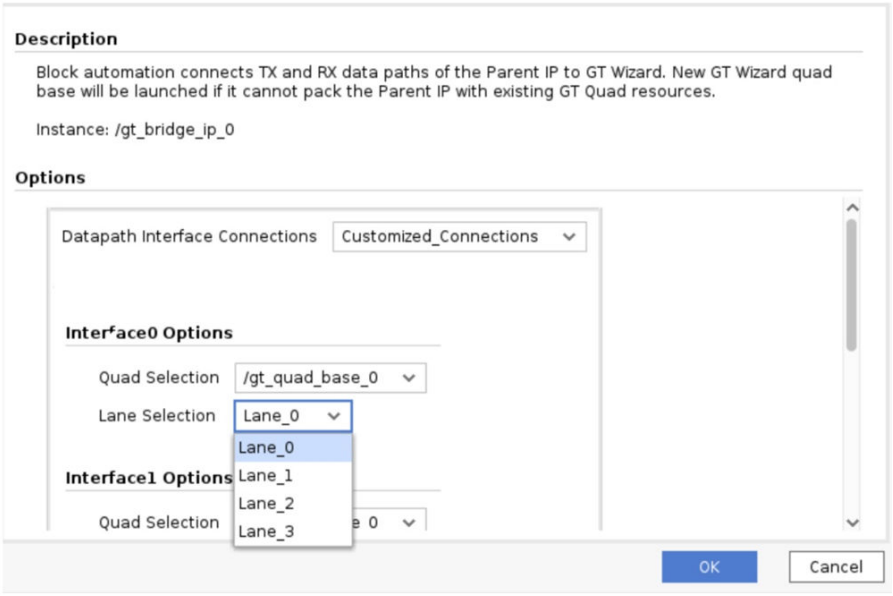
在 IP integrator 画布中添加 Bridge IP 时，会在顶部显示绿色功能区。其中包含块自动化设置链接，用于自动设置部分数据路径、时钟和复位连接。配置 Bridge IP 并单击“Block Automation”（块自动化设置）后，会打开“Block Automation”GUI。选中“Auto”（自动）即表示您允许“Block Automation”选择 GT 四通道资源的最优用法。该工具根据已知可用时钟资源以及 REFCLK 和 PLL 共享规则来制定决策。选中“Start_With_New_Quad”（以新四通道开始）后，“Block Automation”会例化新的 GT 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、时钟和复位连接。

图 9：**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。您必须先在画布中手动添加 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。大多数 GT 父级 IP 应当都已完成移植以支持此功能特性。请参阅相应的 GT 父级 IP 产品指南，以获取有关此高级功能特性用法的进一步详细信息。“Customized_Connections”GUI 选项如下图所示。

图 10：**“Customized_Connections”GUI 选项**



注释： “Block Automation”（块自动化设置）不应将块设计容器 (BDC) 内的四通道纳入考量范围。

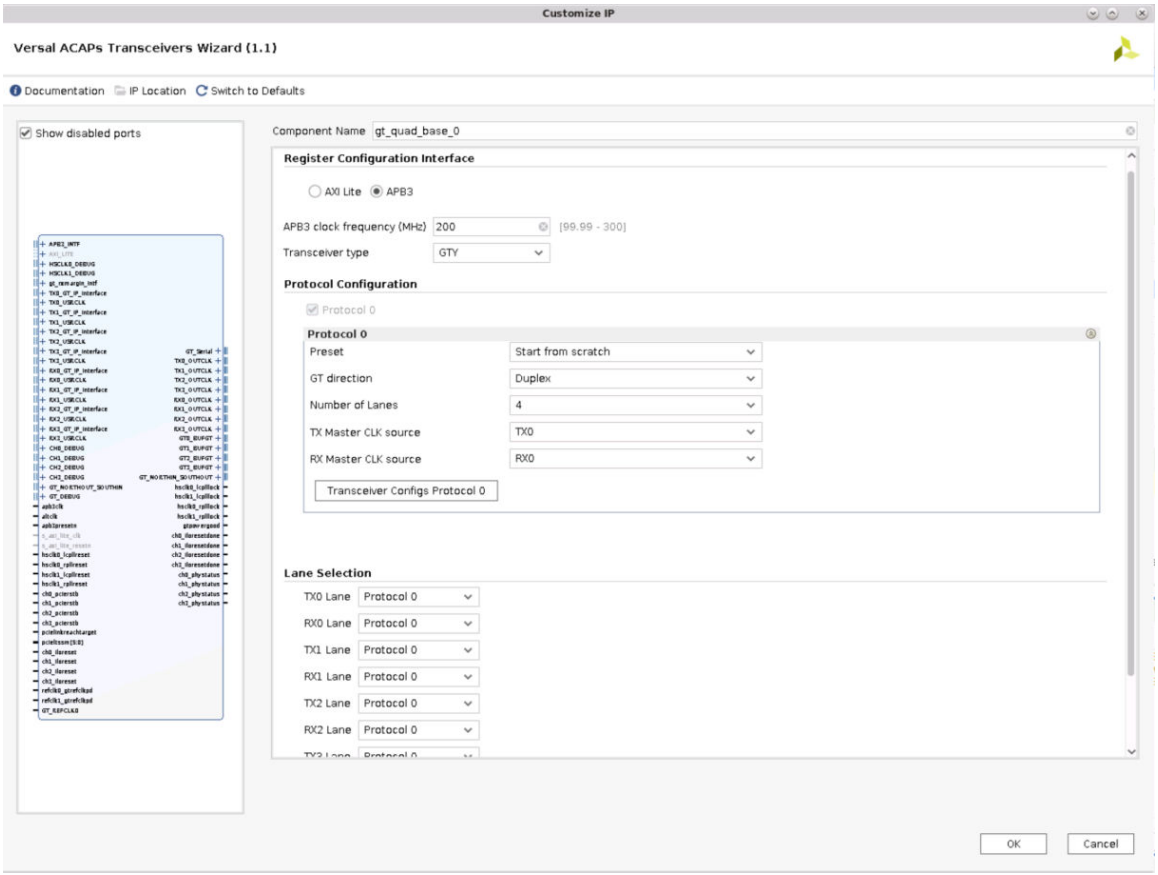
Versal Adaptive SoC Transceivers Wizard

AMD Versal™ Adaptive SoC Transceivers Wizard 包含在 IP 目录中以供使用。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：

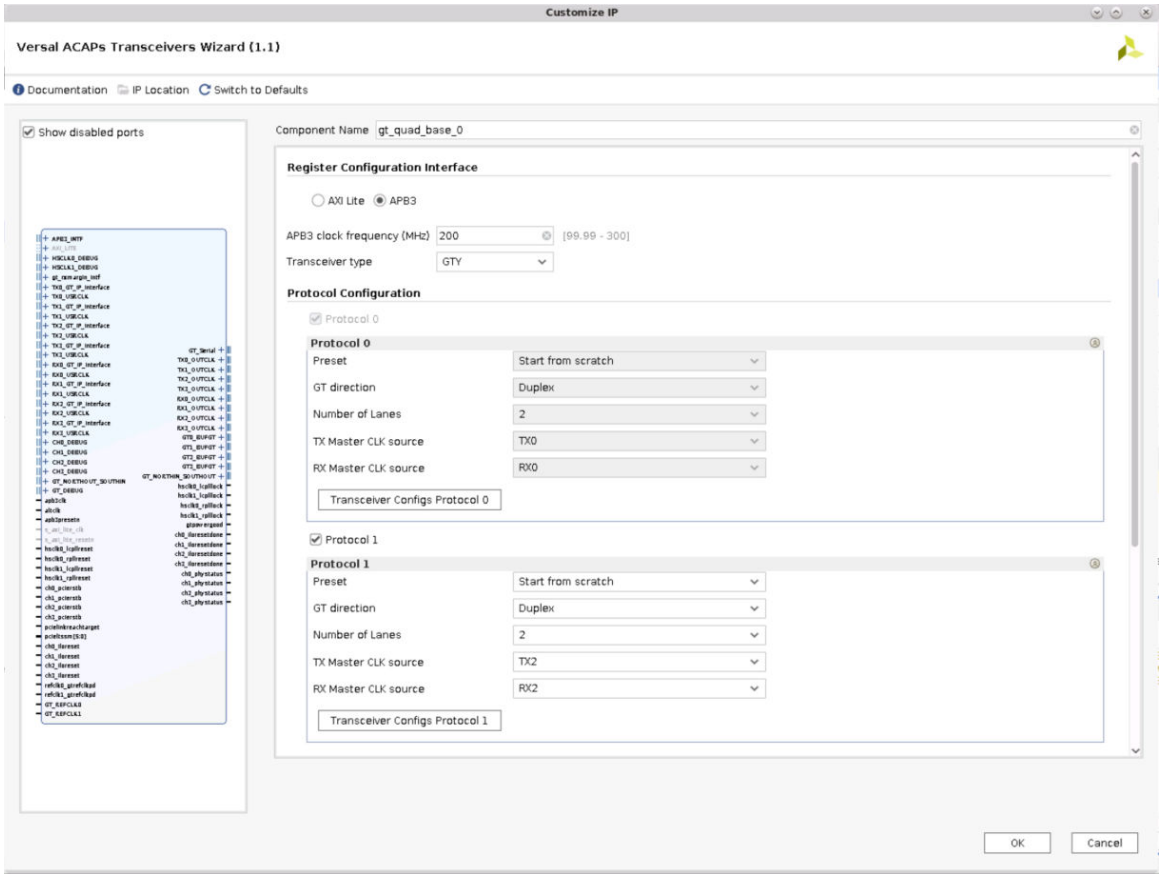
图 11： **“Main GUI Configuration”选项卡**



- “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`。
- “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。此外，可输入所选接口的工作频率。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 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002) 和《Versal 自适应 SoC GTM 收发器架构手册》(AM017) 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）。两个协议选项如下图所示。这两个协议选项均为双工，每个选项都会对 2 条通道进行编程。

图 12：**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，因此最多支持 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 56 Gbps 时，四通道上添加的通道数量将作为活动的 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。

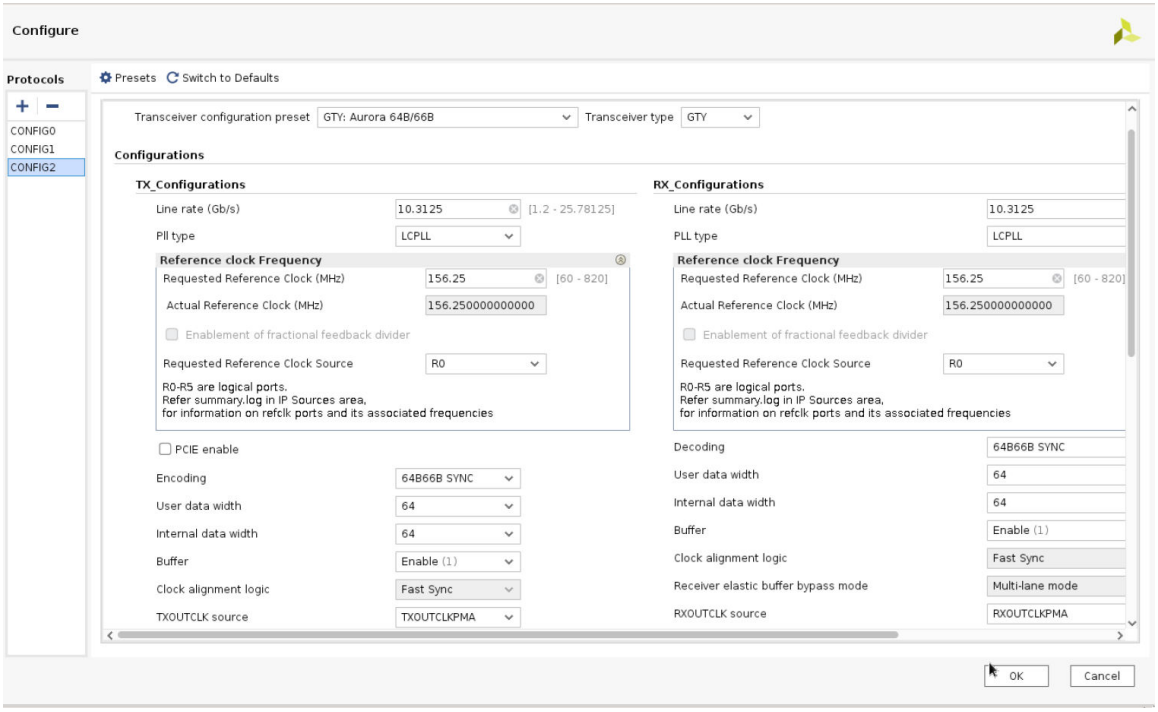
- “Transceiver Configuration Protocol 0”（收发器配置协议 0）：该选项会打开弹出式子 GUI，您可在其中配置收发器的发射器和接收器设置。

- “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 13： **“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 选项的删除采用递减式操作。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。然而，核生成规则基于主 GUI 中输入的 GT 方向（在此例中为 Simplex TX）。

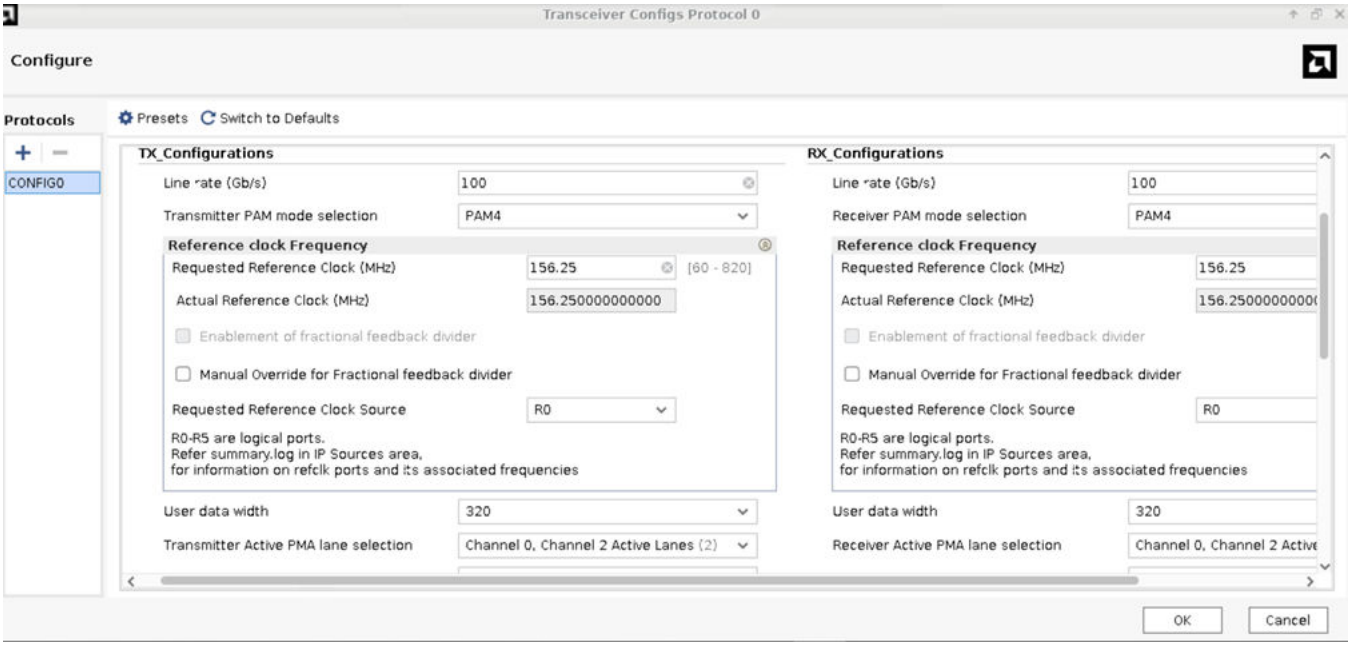
- “Configurations”（配置）：GT 收发器和接收器设置（例如，线速率、GT 类型、参考时钟频率）均可供选择。
- “Reference Clock Frequency”（参考时钟频率）：其中包含用户参考时钟频率输入。实际参考时钟频率是根据用户参考时钟输入计算得出的。
- “Requested Reference Clock Source”（请求的参考时钟源）：应按顺序提供 6 个逻辑参考时钟输入，从 R0 到 R5。基于用户的各种线速率选择，该向导会分配已排序的参考时钟端口 (GTREFCLK0-5) 和频率以供在这些时钟端口上进行驱动。“IP Sources”（IP 源文件）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 14：**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      |
+---------------+-------------------------------+----------+
```

GTME5 的“Sub GUI Configuration”选项卡如下所示。

图 15：**GTME5 的“Sub GUI Configuration”选项卡**



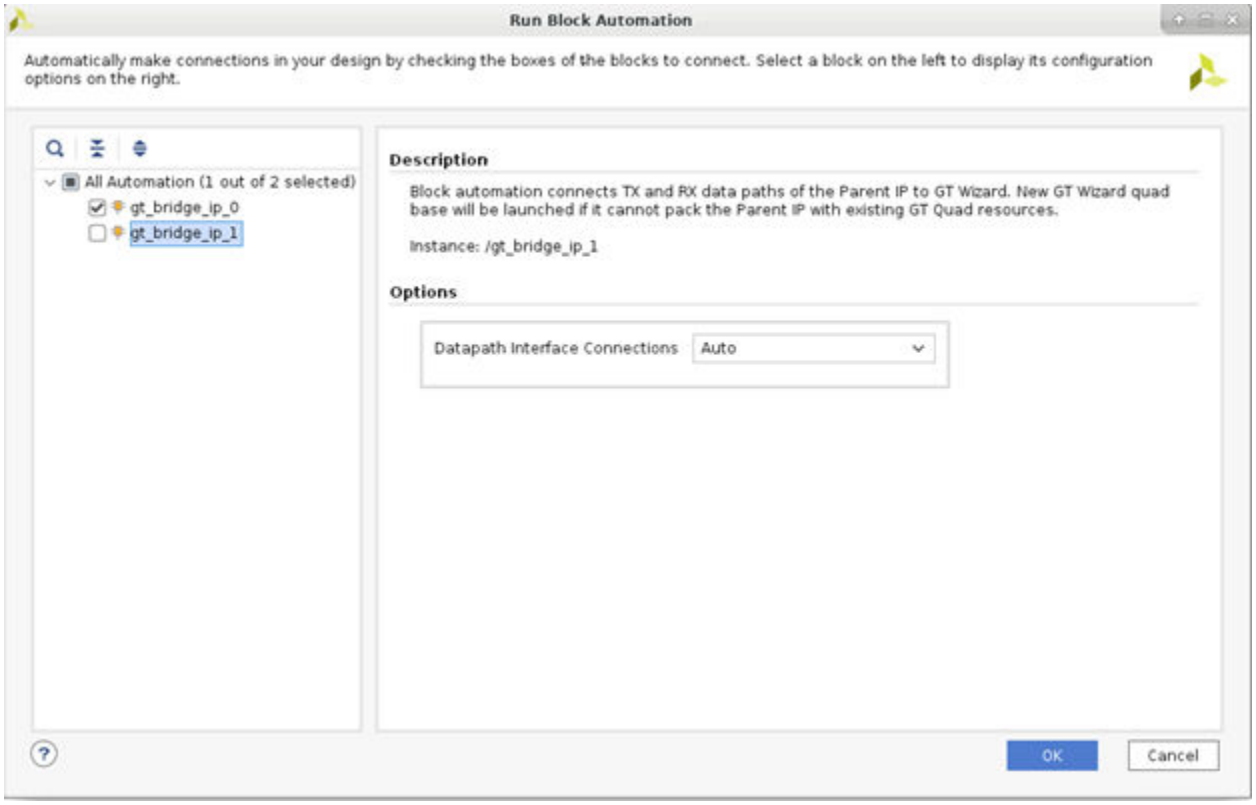
- “Active PMA Lane Selection”（活动 PMA 通道选择）：如果 GTME5 配置所需用户数据宽度为 320/512，那么在双通道内仅使用一条 PMA 通道。该 GUI 选项允许用户选择四通道中所需的活动 PMA 通道。

注释：您可在四通道 IP 目录、*_rules_output.vh 和 *_log.txt 中的 *.gen 目录下找到有关速率变更进程中所涉及的具体属性的信息。

共享多个 Bridge IP 的 GT 四通道

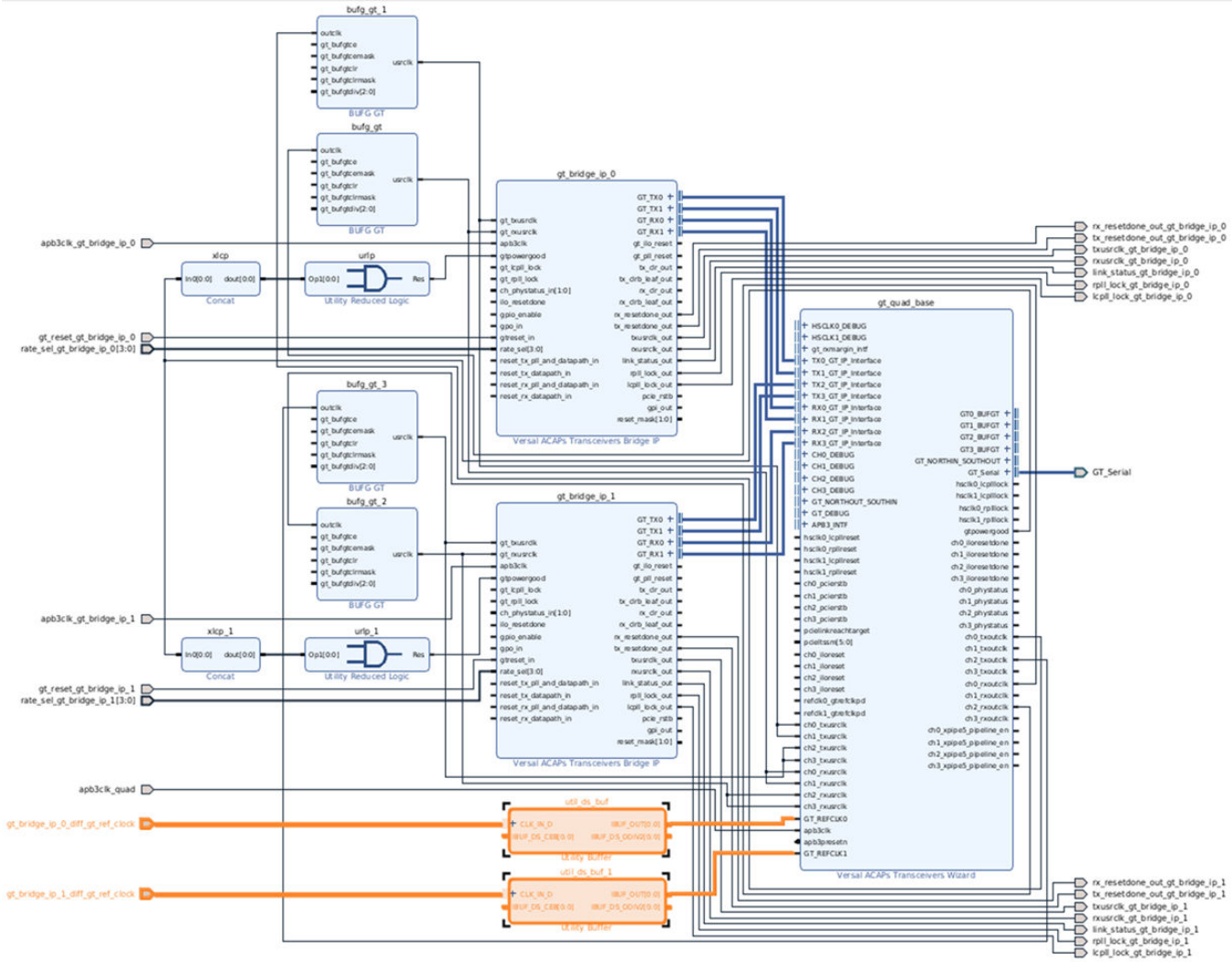
根据 GT 四通道的通道数量、PLL 要求和其他配置，多个 Bridge IP 可共享一个 GT 四通道。例如，两个 x2 通道 Bridge IP 可共享一个 GT 四通道。为达成此结果，您可将 2 个 Bridge IP 放入 IP integrator 画布中，配置 Bridge IP 窗口，然后单击“Block Automation”（块自动化设置）。这样会打开“Block Automation”窗口，如下图所示。此图显示了 2 个 Bridge IP 的“Block Automation”窗口。可对这 2 个 Bridge IP 同时执行或逐个执行块自动化设置。

图 16：**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 17：**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。GT 参考时钟汇总操作将读取 IP integrator 中的 GT 四通道的接口属性，并提供设计中的参考时钟连接的汇总信息。

- 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或拆分 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 18：**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 17：完全连接的 Bridge IP 中所示。

图 19：**示例 1 GT 参考时钟汇总信息表**

```
================================== GT_REFCLOCK Summary Table ==================================

+-------+--------------------------+--------+----------------+---------------+---------+
| S.No. | GT REFCLOCK 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：下图显示了共享 2 个 GT 四通道的 4 个 x2 IP 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 20：**多个 Bridge IP 共享 GT 四通道**

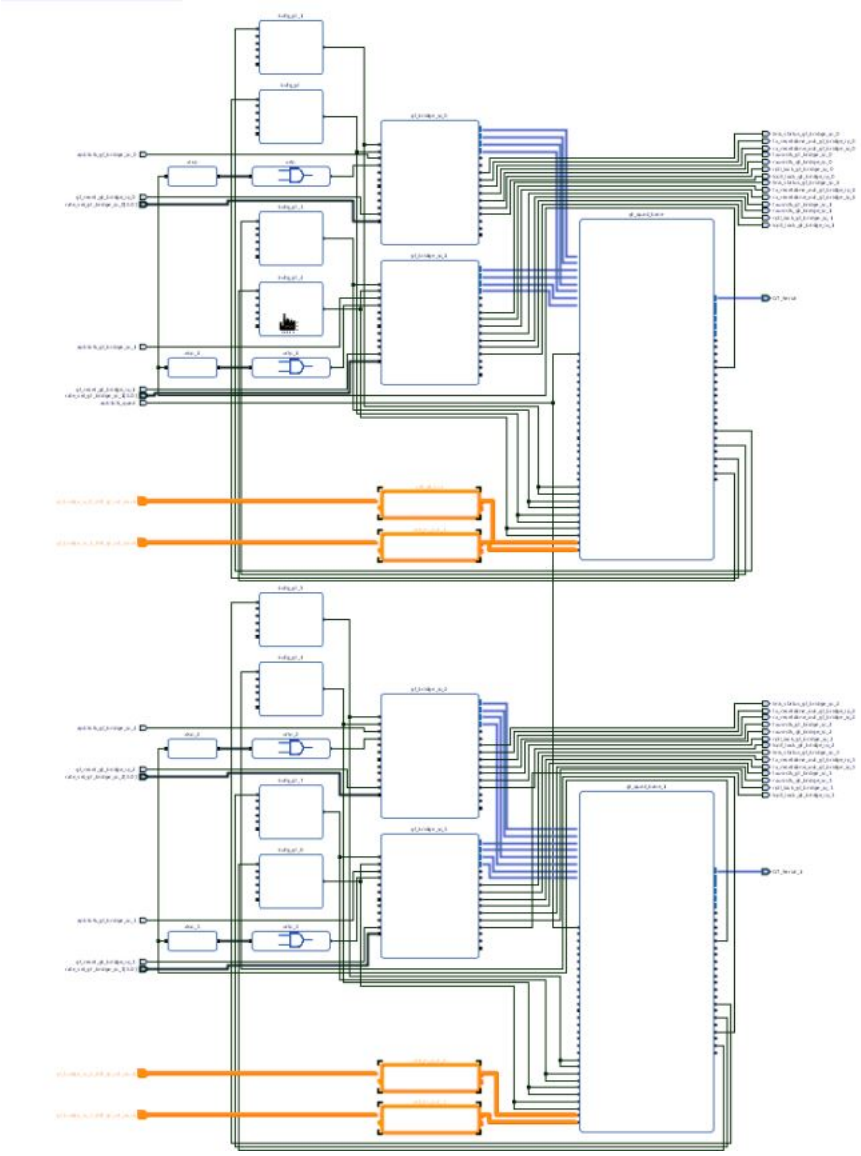


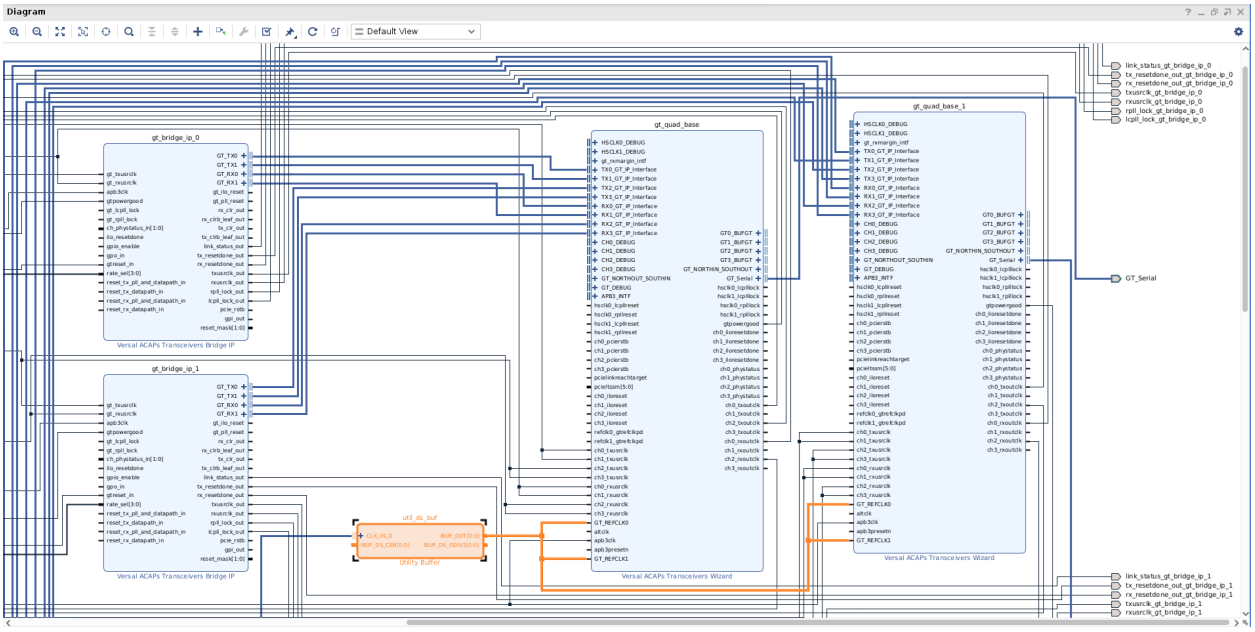
图 21：**示例 2 GT 参考时钟汇总信息表**

GT_REFCLOCK Summary Table

S.No.	GT REFCLOCK 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 22：**共享 GT 四通道的多个 Bridge IP 的修改后参考时钟连接**



AMD IP - GT 四通道集成

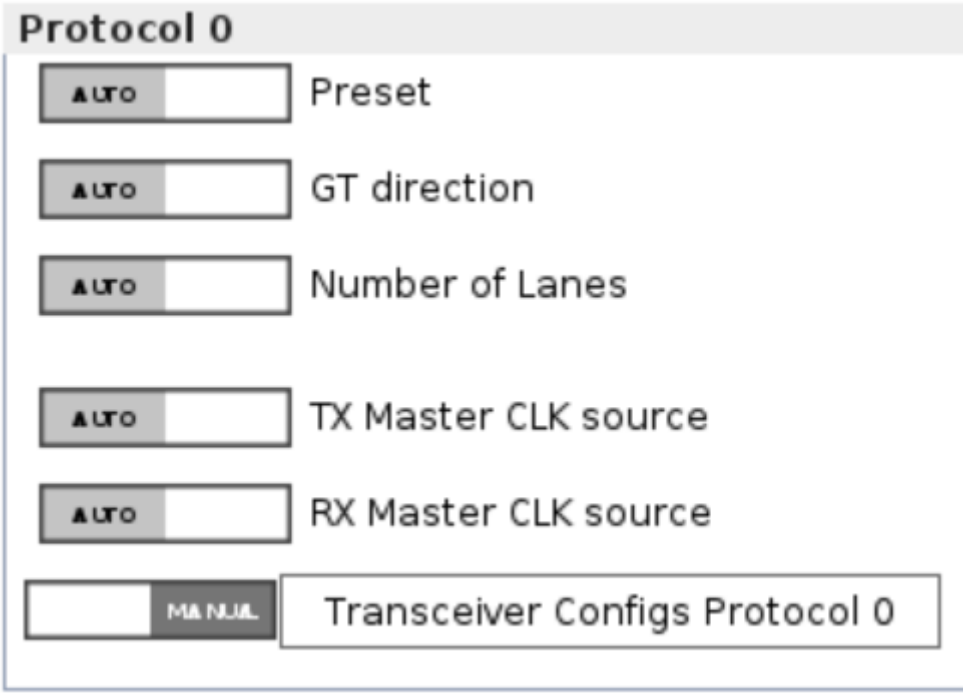
AMD 基于 GT 的 IP（如 Aurora、PCIe 和 MRMAC）可在 IP integrator 中提供块自动化设置，以支持将多个父级 AMD IP 无缝连接到 GT 四通道。IP 块自动化设置可例化 GT 四通道，并创建基本数据路径 USRCLK 和 GT REFCLK 连接。

要使用块自动化设置来连接多个 Aurora IP，请执行以下步骤：

1. 在 IP integrator 画布中，使用“Add IP”（添加 IP）选项添加“Aurora64B66B” IP。
2. 为“Aurora64B66B” IP 配置通道数量、线速率等参数。
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，重复步骤 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”（收发器向导）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）等参数以及其他高级设置进行微调，如下图所示。切换为手动模式后，无论对父级 IP 配置进行任何更改，而后执行确认设计步骤，都不再会将任何 GT 四通道参数从父级 IP 传输至 GT 四通道。仅当确有必要并将所有父级 IP 参数都传输至 GT 四通道后，才能执行手动更改。

图 23： **“Transceiver Wizard”中的“Auto”到“Manual”选项切换**



AMD IP 和定制 IP 共享 GT 四通道

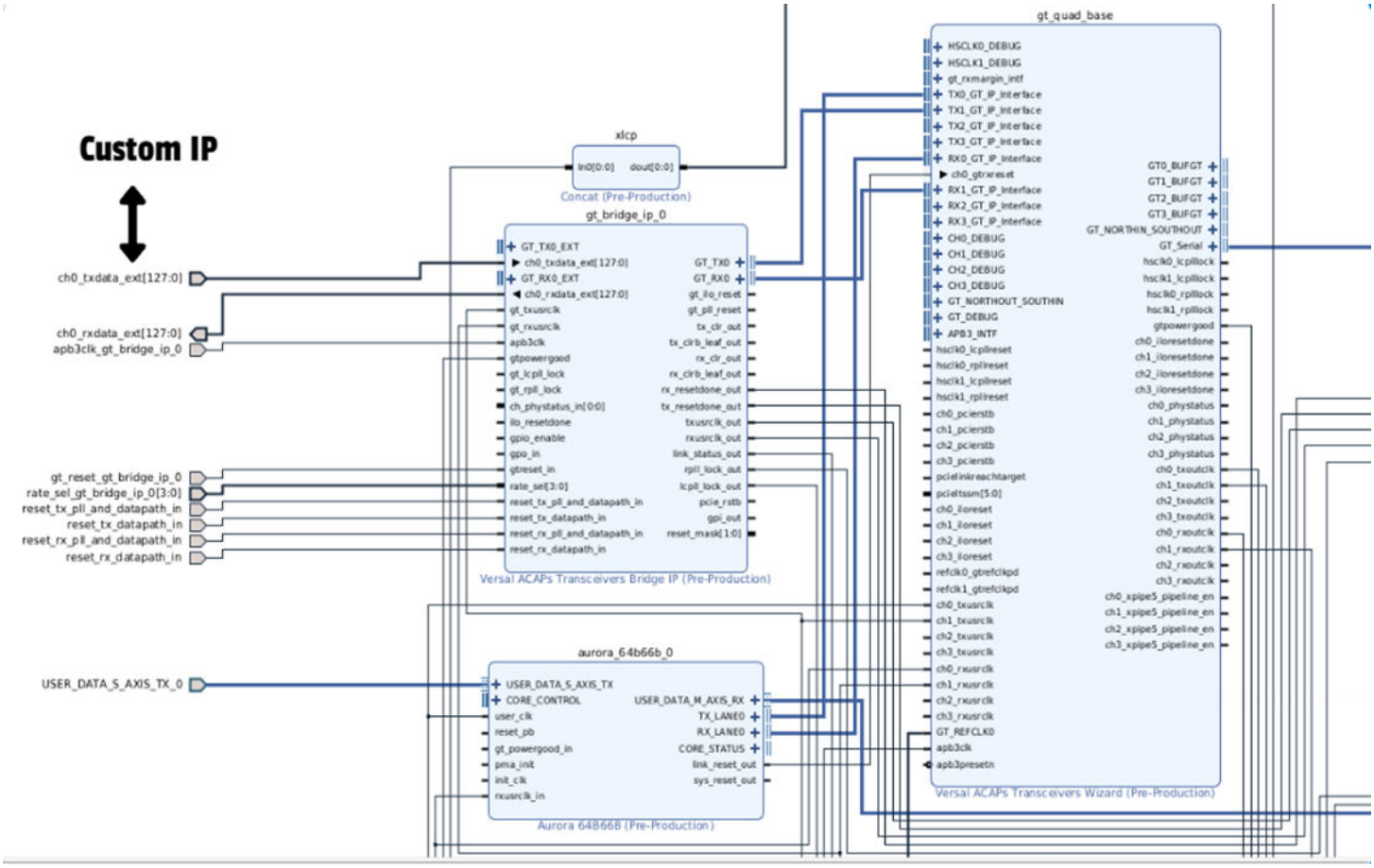
通过使用 IP integrator 功能特性，可在 AMD IP 和定制 IP 之间无缝共享 GT Quad IP。

例如，要在 Aurora IP 和定制 IP 之间共享 GT 四通道，请执行以下步骤：

1. 在 IP integrator 画布中使用“Add IP”（添加 IP）选项添加 Aurora64B66B IP。
2. 为 Aurora64B66B IP 配置通道数量、线速率等参数。
3. 使用 Run_Block_Automation 功能特性将 Aurora IP 连接到 GT Quad base IP。根据 Aurora IP 配置，配置 GT 四通道。
4. 要与定制 IP 共享同一个四通道，须在 IP integrator 画布中添加 Bridge IP。
5. 在直通模式下配置 GT Bridge IP。
6. 在 GT Bridge IP 中输入对应于定制 IP 的 GT 四通道配置。
7. 使用 GT Bridge IP 的 Run_Block_Automation 功能特性，分配定制 IP 通道位置，与 Aurora IP 共享 GT 四通道。
8. 将 GT Bridge IP 的所有基本端口都布局在块设计的外部，以便与定制 IP 对接，并为块设计创建 HDL 封装文件。
9. 将生成的 HDL 封装文件与定制 IP 集成，以便创建系统设计。

下图显示了在 Aurora IP 和 GT Bridge IP 之间共享 GT 四通道的块设计快照。

图 24：**定制 IP 和 Aurora IP 共享同一个 GT 四通道**

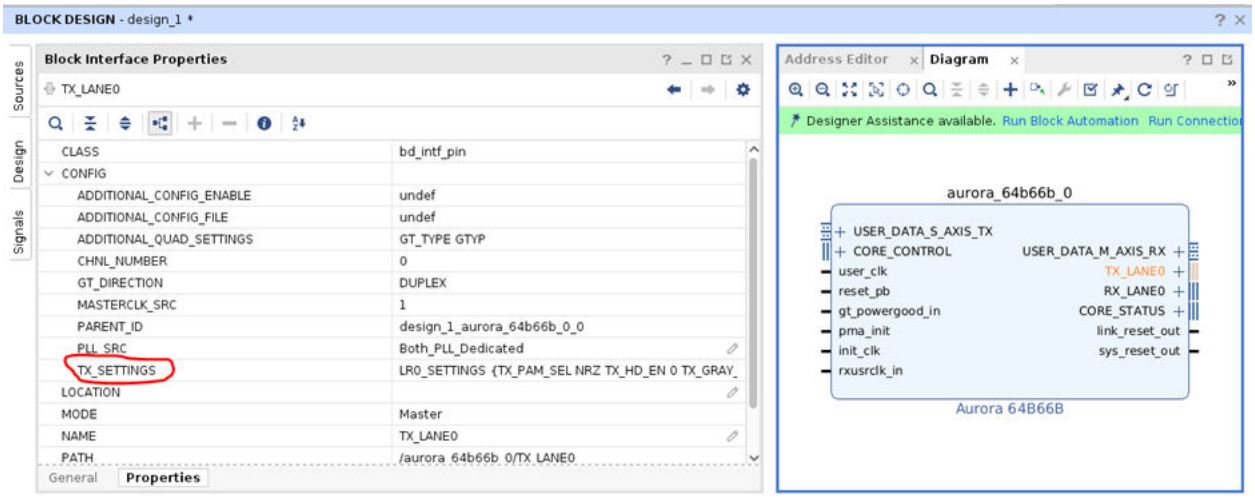


IP integrator 中针对 GT 四通道用户的通用准则

以下是 IP integrator 中针对 GT 四通道用户的一些通用准则：

- 在直通模式下，IP integrator 中基于 GT 四通道的设计的设计输入为 AMD IP 或 GT Bridge IP。直通模式下的 GT Bridge IP 专用于定制 IP，以支持在 IP integrator 中使用 GT 四通道。对于当前 IP，建议使用 AMD IP 或 GT Bridge IP 来配置 GT Quad base IP。配置 GT Quad base IP 后，用户即可修改高级设置
- 父级 IP 在 IP integrator 的接口上托管 GT 配置。发射器配置设置是 TX 接口的一部分，而接收器配置设置是 RX 接口的一部分。发射器配置显示在“Block Interface Properties”（块接口属性）窗口的 TX_SETTINGS 中，如下图所示。根据在 IP integrator 中与 GT 四通道建立的连接数，GT 四通道可自动推断配置。确认设计后，配置设置将从父级 IP 传输至 GT 四通道。

图 25：**块设计中的 Aurora 64B66B 接口属性**



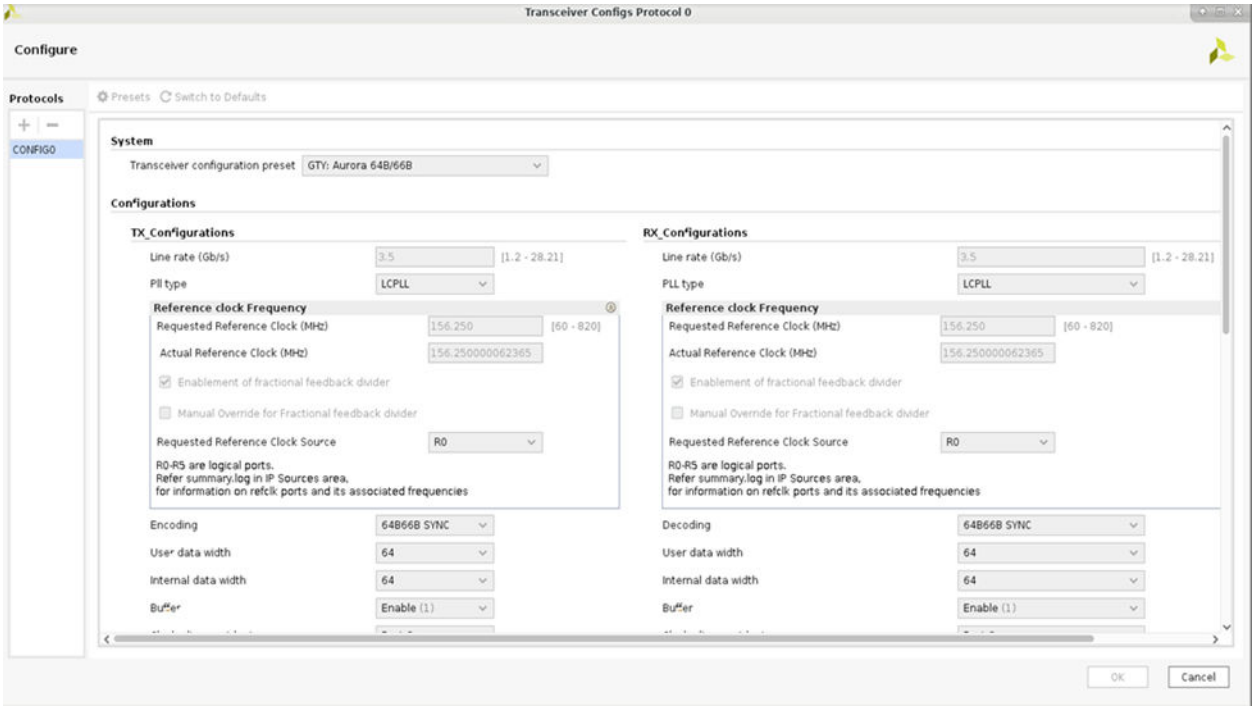
以下是 Aurora IP 的 TX_SETTINGS 示例。

图 26：**TX 接口的四通道设置快照**

```
LR0_SETTINGS {TX_PAM_SEL NRZ TX_GRAY_BYP true TX_GRAY_LITTLEENDIAN true TX_PRECODE_BYP true TX_PRECODE_LITTLEENDIAN false TX_LINE_RATE 3.5 TX_PLL_TYPE LCPLL TX_REFCLK_FREQUENCY 156.250 TX_ACTUAL_REFCLK_FREQUENCY 156.250000062365 TX_FRACN_ENABLED true TX_FRACN_NUMERATOR 0 TX_REFCLK_SOURCE R0 TX_DATA_ENCODING 64B66B_SYNC TX_USER_DATA_WIDTH 64 TX_INT_DATA_WIDTH 64 TX_BUFFER_MODE 1 TX_BUFFER_BYPASS_MODE Fast_Sync TX_PIPM_ENABLE false TX_OUTCLK_SOURCE TXOUTCLKPMA TXPROGDIV_FREQ_ENABLE false TXPROGDIV_FREQ_SOURCE LCPLL TXPROGDIV_FREQ_VAL 322.265625 TX_DIFF_SWING_EMPH_MODE CUSTOM TX_64B66B_SCRAMBLER false TX_64B66B_ENCODER false TX_64B66B_CRC false TX_RATE_GROUP A TX_LANE_DESKEW_HDMI_ENABLE false TX_BUFFER_RESET_ON_RATE_CHANGE_ENABLE GT_TYPE GTY}
```

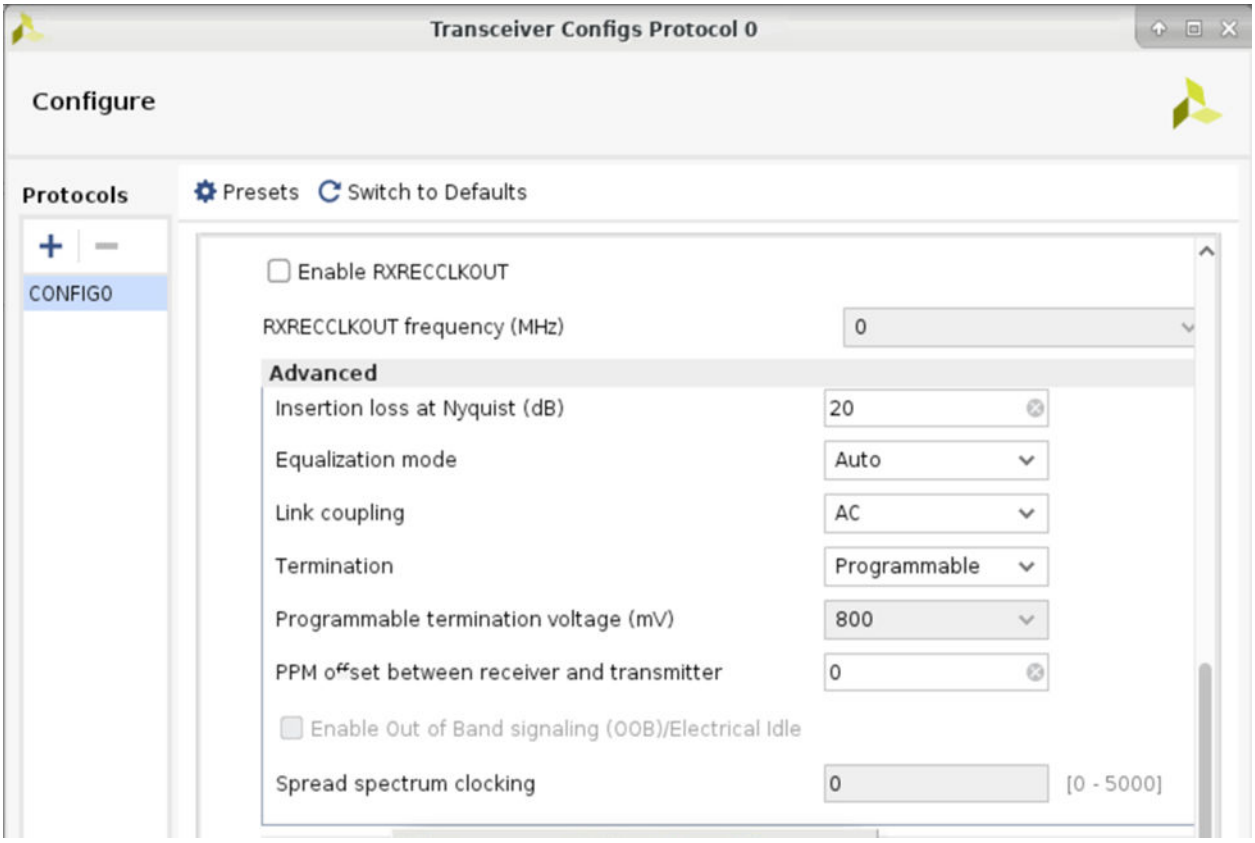
同样，Aurora IP 上的 RX 接口具有 GT 四通道的 RX SETTINGS。下图显示了将参数从 Aurora IP 传输至 GT Quad IP 后的 GT 四通道配置。

图 27：**对应于 Aurora 配置的 GT 四通道配置**



- 强烈建议不要在 IP integrator 中显式配置 GT 四通道（将 AUTO 模式更改为 MANUAL 模式）。可使用 GT 四通道的自动推断功能，根据设计中的连接数来传输配置。
- 要微调插入损耗和均衡等参数以及其他高级设置，可将“Transceiver Configs Protocol”（收发器配置协议）的“AUTO”（自动）选项更改为“MANUAL”（手动）模式，并编辑 GT 四通道配置，如下图所示。

图 28：**高级接收器设置**



- 将 GT 四通道配置更改为手动模式可能会导致 write_bd_tcl 故障。如需了解更多信息，请参阅答复记录 000034832。
- 切换为手动模式后，无论对父级 IP 配置进行任何更改，而后执行确认设计步骤，都不再会将任何 GT 四通道设置从父级 IP 传输至 GT 四通道。

更改高级接收器设置

1. 使用父级 IP 和 GT 四通道之间的所有连接来创建设计。
2. 确认设计，以确保 GT 四通道自动配置为父级 IP 配置。
3. 将“Transceiver Configs Protocol”（收发器配置协议）的模式从“Auto”（自动）更改为“Manual”（手动），并更新所需的高级接收器设置。完成此操作后，建议不要将新的父级 IP 连接到 GT 四通道接口。
4. 如果要从“Manual”切换回“Auto”，建议将所有参数切换为“Auto”并重新确认设计。完成此操作后，GT 四通道中手动配置的所有配置都将由传输自父级 IP 接口的配置所覆盖。

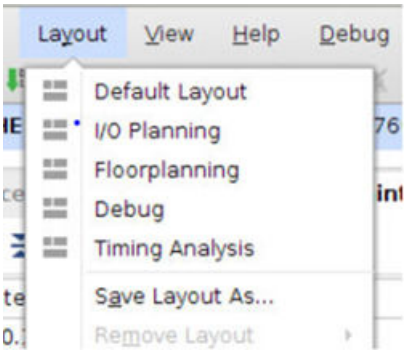
串行 I/O 调试

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：烧录和调试》(UG908) 中的“串行 I/O 硬件调试流程”章节。

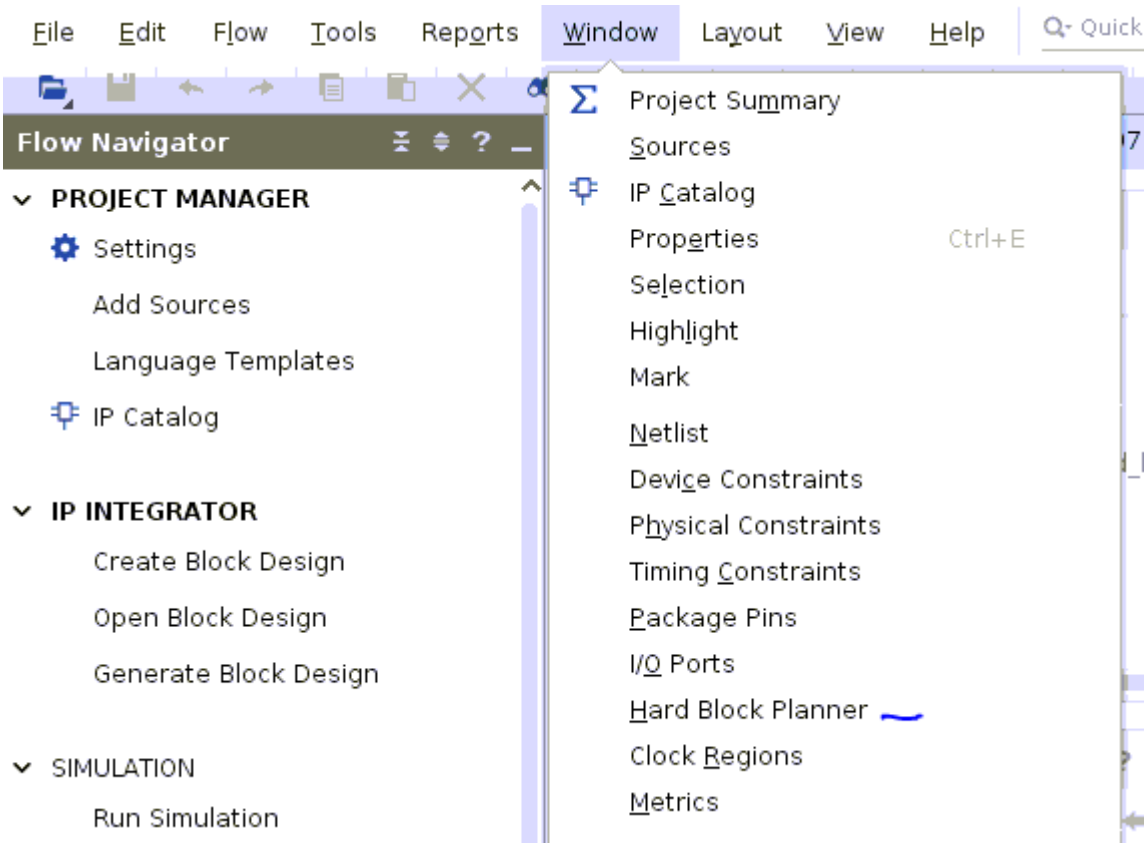
设计的综合和实现

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，然后单击“Validate Design”（确认设计）或者按 F6 键。
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，单击 Vivado 中的“Run Synthesis”（运行综合）以进行设计综合。
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“Window Hard Block Planner”（窗口硬核块规划器）窗格，如下图所示：



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。

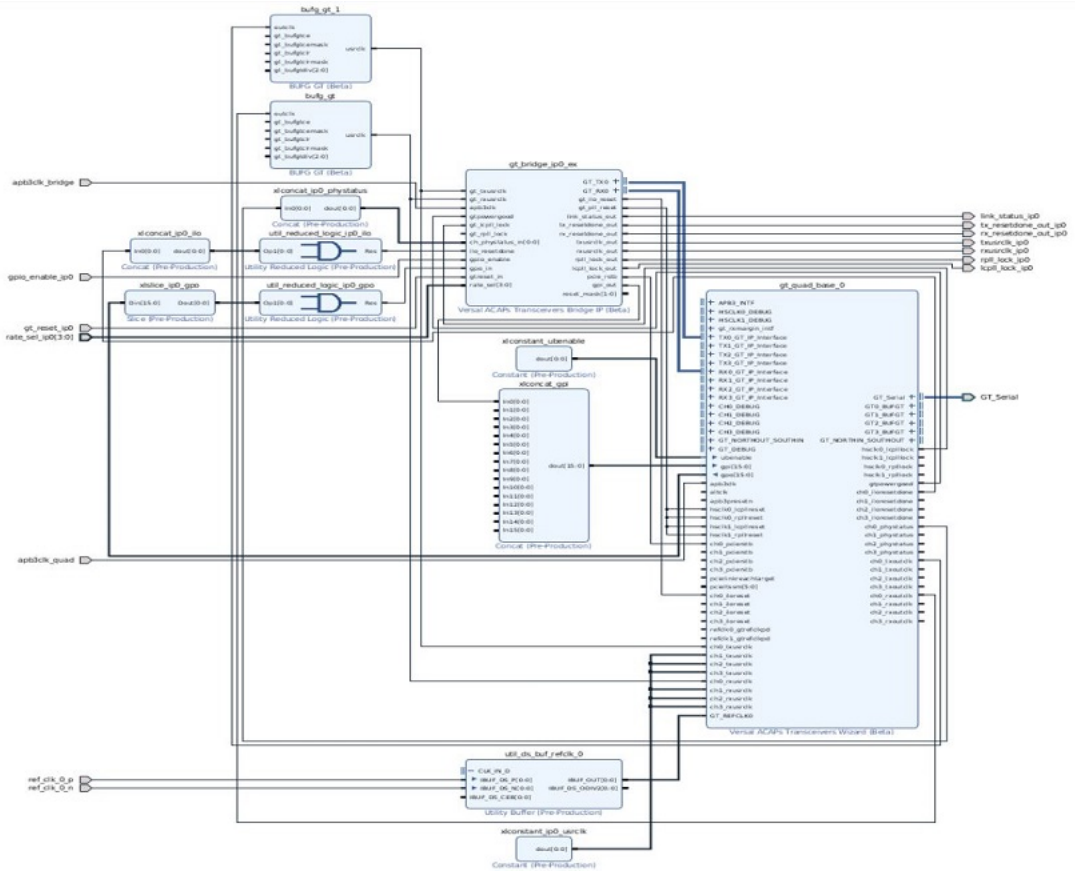
6. 分配完所有 I/O 端口后，单击“Run Implementation”（运行实现）即可实现设计。

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 AMD 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 29：**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。

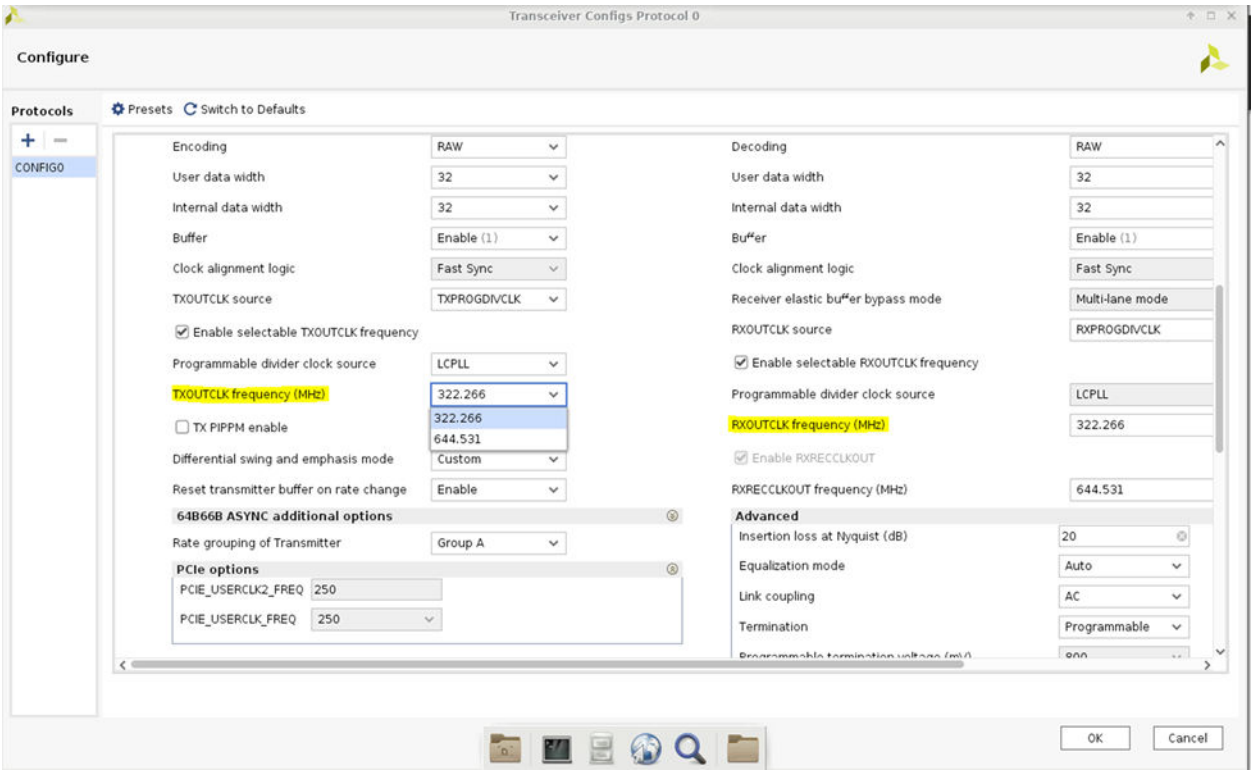
Bridge IP 具有复位控制器帮助程序块，可简化串行收发器原语的复位和初始化操作。

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。此过程非常简单，但您需要了解以下限制：

- 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。仅当发射器和接收器配置为采用相同的线速率和相同的数据编码时，才能正确检查数据完整性。不使用任何速率调整方案。如果系统中配置的发射器和接收器线速率或数据编码彼此不同，那么可以对两个适当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 的接收器的线速率和数据编码相匹配，反之亦然。
- 对于多四通道设计，不提供设计示例。
- 如果用户修改了 GUI 所显示的 `outclk` 频率值，此设计示例并不保证能够实现此类配置的数据完整性。这是因为需要对设计示例内从 `outclk` 到 `usrclk` 的路径中 BUFG GT 分频器的值进行相应调整。

图 30：**收发器配置示例**



对于此配置，`OUTCLK` 频率的默认值为 322.266 MHz。如果将 OUTCLK 频率配置为 644.531 MHz，则用户需要相应调整 BUFG GT 分频器值，以便 BUFG GT 能够将 322.266 MHz 作为 `USRCLK` 频率值应用于 GT 四通道。

设计示例仿真

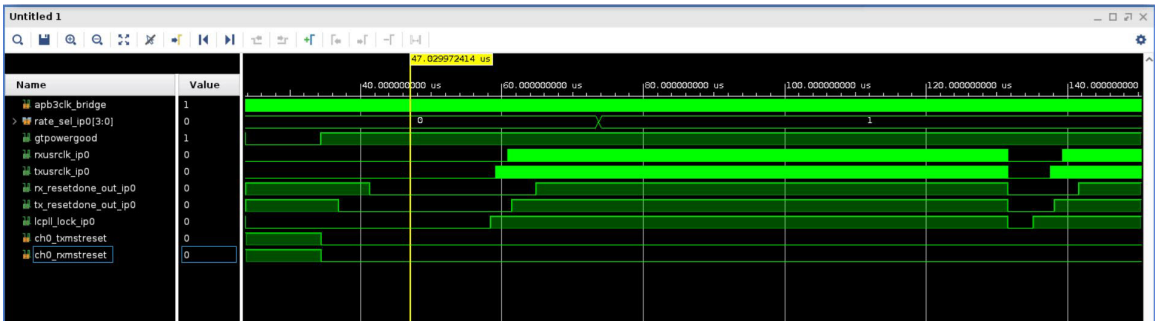
要对 Wizard IP 核的实例进行仿真，请首先打开其设计示例。

在工程示例中，转至 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）以启动行为仿真。

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) 中的实验课 6。

设计示例仿真测试激励文件提供了必要的自由运行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的其余部分。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。在多线速率配置中，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他“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 31：**显示速率更改的仿真波形**



速率变更应以 0 来驱动，直至初始 tx/rx_resetdone_out_ip0 断言有效为止。

`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: simulation 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
```

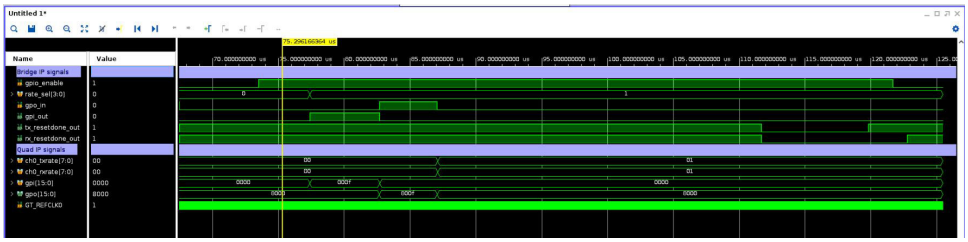
最后，会在转录文本中打印以下消息，表明测试成功。

```
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，假设某个 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。在此类情况下，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 后，Bridge IP 会生成要传输给 GT 四通道的 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 32：**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 AMD 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。AMD 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。

利用 AMD 自适应计算解决方案获取帮助

为了帮助您在使用该核时完成设计和调试进程，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。社区论坛还可供会员学习、参与、分享和提出与 AMD 自适应计算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技术支持网页上找到，也可以通过 AMD 自适应计算 Documentation Navigator 来获取。要下载 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。

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 AMD 自适应计算产品的所有已知问题。我们每天都会创建和维护答复记录，确保您可以获取最准确的信息。

您可以通过支持网页（主页）上的“Search Support”（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 Adaptive SoC Transceivers Wizard 的主答复记录

- 有关 Versal 自适应 SoC GTY/GTYP 的答复记录 75716。
- 有关 Versal 自适应 SoC GTM 的答复记录 000034042。

技术支持

AMD 自适应计算在社区论坛上为此 AMD 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 AMD 自适应计算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，请导航至社区论坛。

调试工具

有许多工具可用于解决 Versal Adaptive SoC 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

AMD 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 AMD 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：烧录和调试》(UG908)。

附录 B

附加资源与法律声明

查找其他文档

文档门户

AMD 自适应计算文档门户是旨在使用您的网页浏览器提供健全的文档搜索和导航的在线工具。要访问文档门户，请转至 https://docs.amd.com。

注释：单击链接将打开英语版本，但您可从下拉列表中选择简体中文版本（如可用）。请注意，简体中文版本可能比英语版本旧。

Documentation Navigator

Documentation Navigator (DocNav) 是预安装的工具，支持访问 AMD 自适应计算文档、视频和支持资源，您可在其中通过筛选和搜索来查找信息。要打开 DocNav，请执行以下操作：

- 在 AMD Vivado™ IDE 中，单击“Help” → “Documentation and Tutorials”。
- 在 Windows 上，单击“Start”（开始）按钮并选中“Xilinx Design Tools” → “DocNav”。
- 在 Linux 命令提示中输入 `docnav`。

注释：如需了解有关 DocNav 的更多信息，请参阅《Documentation Navigator 用户指南》(UG968)。

注释：您无法从 DocNav 访问简体中文版本。请使用设计中心网页。

设计中心

AMD 设计中心提供了根据设计任务和其他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。
- 转至设计中心网页。

支持资源

如需获取答复记录、技术文档、下载以及论坛等支持资源，请访问技术支持。

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：烧录和调试》(UG908)
7. 《Versal 自适应 SoC GTY 和 GTYP 收发器架构手册》(AM002)
8. 《Vivado Design Suite 教程：逻辑仿真》(UG937)
9. 《Versal 自适应 SoC 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
2023 年 10 月 24 日 1.1 版	
· 块自动化设置流程 · “Sub GUI Configuration”选项卡 · 设计示例仿真	已更新。
2023 年 5 月 22 日 1.1 版	
· 复位状态机 · 复位排序和其他服务 · AMD IP 和定制 IP 共享 GT 四通道 · IP integrator 中针对 GT 四通道用户的通用准则 · 更改高级接收器设置	新增章节。
· 复位控制器帮助程序块 · 适用于定制 IP 的 IP integrator 设计输入 · GT 参考时钟汇总与最优化 · 设计示例的限制	已更新。
2022 年 10 月 19 日 1.1 版	
复位控制器帮助程序块	在 第 5 章：设计示例 中新增章节
“Sub GUI Configuration”选项卡	更新 第 4 章：设计流程步骤 中的章节
GT 参考时钟汇总与最优化	
2022 年 4 月 29 日 1.1 版	
第 3 章：产品规格	更新《Versal 自适应 SoC GTM 收发器架构手册》(AM017) 链接

章节	修订综述
2022 年 4 月 26 日 1.1 版	
· 第 3 章：产品规格 · 适用于定制 IP 的 IP integrator 设计输入 · “Sub GUI Configuration”选项卡 · 设计示例仿真	更新章节。
2021 年 6 月 16 日 1.1 版	
常规更新	更新 GTYP 和 GTM 的参考
2020 年 11 月 23 日 1.1 版	
常规更新	· 更新 第 3 章：产品规格 · 更新 自定义和生成核 · 更新 共享多个 Bridge IP 的 GT 四通道 · 新增 AMD IP - GT 四通道集成 · 更新 第 5 章：设计示例
2020 年 7 月 14 日 1.0 版	
初始版本	不适用

请阅读：重要法律声明

本文档所示信息仅做参考，其中可能包含不准确的技术信息、疏漏和印刷错误。受诸多原因影响，此处所含信息可能发生更改，也可能无法准确呈现，这些原因包括但不限于产品和路线图变更、组件和主板版本更改、新增模型和/或产品发布、不同制造商之间存在的产品差异、软件更改、BIOS 刷新、固件升级等。任何计算机系统均存在安全性漏洞风险，无法彻底阻止也无法缓解这类风险。AMD 没有任何义务来更新或者以任何其他方式纠正或修改这些信息。但 AMD 保留随时修改这些信息和更改文档内容的权利，AMD 没有任何义务将此类修改或更改通知任何人。此处信息“按原样”提供。AMD 对于本文档内容不作任何陈述或保证，并且对于这些信息中可能出现的任何不准确、错误或疏漏问题不承担任何责任。对于有关任何暗含的非侵权、适销性及适合特定用途的保证，AMD 特此声明不承担任何责任。无论在任何情况下，对于任何人因使用此处包含的任何信息而形成的依赖或者引发的任何直接、间接、特殊或其他后果性损害，AMD 概不负责，即使 AMD 已明确获悉存在发生此类损害的可能性也是如此。

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