

# 10G/25G High Speed Ethernet Subsystem v4.1

## 产品指南

Vivado Design Suite

PG210 (v4.1) 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

<b>第 1 章：简介</b>	4
功能特性	4
IP 相关信息	5
<b>第 2 章：概述</b>	6
按设计进程浏览内容	6
子系统概述	6
功能特性总结	6
应用	9
许可和订购	9
<b>第 3 章：产品规格</b>	12
标准	14
性能和资源使用情况	14
时延	14
端口描述 - MAC+PCS 变体	15
端口描述 - PCS 变体	38
端口描述 - 10G Ethernet MAC (64 位) 变体	42
寄存器空间	54
<b>第 4 章：利用子系统进行设计</b>	110
时钟设置	110
复位	115
LogiCORE 设计示例时钟设置和复位	118
针对 IEEE 1588v2 标准的支持	122
RS-FEC 支持	130
以太网数据路径奇偶校验	132
802.1cm 抢占功能	133
状态/控制接口	137
暂停处理	138
自动协商	142
链路训练	144
<b>第 5 章：设计流程步骤</b>	148
自定义和生成子系统	148
约束子系统	156
仿真	157
综合与实现	159

<b>第 6 章：设计示例.....</b>	160
概述.....	160
设计示例层级（设计示例中包含 GT）.....	163
用户接口.....	167
核 xci 顶层端口列表.....	169
双工操作模式.....	211
运行时可切换.....	212
共享逻辑实现.....	213
AXI4-Lite 接口实现.....	215
IEEE 第 108 条 (RS-FEC) 集成.....	219
PTP 1588 Timer Syncer 块.....	220
<b>第 7 章：批处理模式测试激励文件.....</b>	232
<b>附录 A：升级.....</b>	233
从 v2.3 到 v2.4 的更改.....	233
从 v2.3（2017 年 10 月 4 日）到 v2.3（2017 年 12 月 20 日）的更改.....	234
从 v2.2 到 v2.3 的更改.....	235
从 v2.1 到 v2.2 的更改.....	236
从 v2.0 到 v2.1 的更改.....	237
从 v2.0（2016 年 10 月 5 日）到 v2.0（2016 年 11 月 30 日）的更改.....	238
从传统 XGEMAC 进行移植.....	241
<b>附录 B：调试.....</b>	252
在 Xilinx.com 上寻求帮助.....	252
调试工具.....	253
仿真调试.....	254
硬件调试.....	256
协议接口调试.....	258
<b>附录 C：附加资源与法律声明.....</b>	262
赛灵思资源.....	262
Documentation Navigator 与设计中心.....	262
参考资料.....	262
修订历史.....	263
请阅读：重要法律声明.....	270

# 简介

赛灵思 10G/25G High Speed Ethernet Subsystem 可按 25G 以太网联盟指定的方式实现含物理编码子层 (PCS) 的 25G 以太网媒体访问控制器 (MAC)。用户可使用 MAC 和物理编码子层/物理介质连接 (PCS/PMA)，或者也可以单独使用 PCS/PMA。支持传统 10 Gb/s 操作。

---

## 功能特性

- 专为满足 IEEE 802.3 第 49 条、IEEE 802.3 by 和 25G 以太网联盟所指定的 10/25 Gb/s 以太网运行要求而设计
- 包含完整的以太网 MAC 和 PCS/PMA 功能或者独立 PCS/PMA（适用于 25 Gb/s 运行速率）
- 包含完整的以太网 MAC 和 PCS/PMA 功能、独立 MAC 或独立 PCS/PMA（适用于 10 Gb/s 运行速率）。包含独立 64 位以太网 MAC
- 以数据包为导向的简单用户接口
- 全方位的统计数据收集
- 对应所有主要功能指示符的状态信号
- 随顶层封装一起交付，包括功能性收发器封装、IP 网表、测试脚本样本以及 Vivado® Design Suite 工具编译脚本
- BASE-R PCS 子层，运行速率为 10.3125 Gb/s 或 25.78125 Gb/s
- 可选第 74 条 BASE-KR FEC 子层
- 可选自动协商/链路训练
- 可选第 108 条 25G 里所码
- 前向纠错 (RS-FEC) 子层（“Soft RS-FEC TX, Hard RS-FEC RX” 选项，用于通过利用 UltraScale+™ 器件中的 CMAC 块中存在的嵌入式 100G RS-FEC 功能来降低逻辑利用率）。
- 定制前导码模式
- 可选 IEEE 1588 单步和双步时间戳
- 运行时可切换（可在 10G 与 25G 间切换）
- 可选付费型时间敏感网络 (TSN) 功能，根据 IEEE 802.1 CM 标准而设计
  - 支持高速流量与低优先级流量穿插
  - 支持帧抢占

# IP 相关信息

相关信息表	
子系统规格	
支持的器件系列 <sup>1</sup>	Versal® ACAP Zynq® UltraScale+™ RFSoC Zynq® UltraScale+™ MPSoC Virtex® UltraScale+™ Kintex® UltraScale+™ Virtex® UltraScale™ Kintex UltraScale Artix® UltraScale+
支持的用户接口	AXI4-Stream 适用于 MAC 变体 XGMII 或 25GMII 适用于仅限 PCS 变体
资源	<a href="#">性能和资源使用情况网页</a>
随子系统提供	
设计文件	加密寄存器传输级 (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	Verilog
支持的软件驱动程序	<a href="#">Linux</a>
经过测试的设计流程 <sup>2</sup>	
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	Synopsys 或 Vivado 综合
支持	
版本说明和已知问题	主答复记录： <a href="#">64710</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：**

- 如需获取受支持的器件的完整列表，请参阅 Vivado IP 目录。
- 如需了解受支持的第三方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](#)。

**注释：**要访问 25G 规范，请访问 [25G 以太网联盟](#)网站。

## 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](https://www.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：
  - 端口描述 - MAC+PCS 变体
  - 端口描述 - PCS 变体
  - 端口描述 - 10G Ethernet MAC（64 位）变体
  - 寄存器空间
  - 时钟设置
  - 复位
  - 自定义和生成子系统
  - 第 6 章：设计示例

### 子系统概述

本文档详述了 25G 以太网联盟所定义的 10G/25G Ethernet Subsystem 的功能特性。PCS 功能符合 IEEE 802.3 标准 2015 第 49 条 “Physical Coding Sublayer (PCS) for 64B/66B, type 10GBASE-R” 的定义。对于 25G 运行，通过提升时钟频率来提供以 25.78125 Gb/s 速率运行的串行接口，以便充分利用最新的高速串行收发器。此低时延设计专为 UltraScale™ 架构器件经过最优化。

### 功能特性总结

请参阅下表以了解 LogiCORE IP 子系统的不同变体选项的兼容性。

### 25G 支持的功能特性

- 完整的以太网 MAC 和 PCS 功能

- 根据 25G 以太网联盟的 Schedule 3 设计
- 统计数据和诊断
- 66 位串行器/解串器 (SerDes) 接口，使用启用异步变速箱 (Asynchronous Gearbox) 的赛灵思 GTY 收发器
- 暂停处理，包括 IEEE802.3 标准的附录 31D（基于优先级的流量控制）
- 低时延
- 定制前导码和可调整的帧间间距
- 可配置为 10.3125 Gb/s 运行速率（第 49 条）
- 可选第 73 条自动协商
- 可选第 72.6.10 条链路训练
- 可选第 74 条 FEC - 简化循环代码 (2112, 2080)
- 前向纠错 (RS-FEC) 子层（“Soft RS-FEC TX, Hard RS-FEC RX” 选项，用于通过利用 UltraScale+ 器件中的 CMAC 块中存在的嵌入式 100G RS-FEC 功能来降低逻辑利用率）。
- 可选第 108 条 RS FEC（仅限 25 Gb/s 运行速率）
- 仅限 PCS 版本（含 25GMII 接口）
- 64 位 AXI4-Stream 接口
- 可选 AXI4-Lite 控制接口和状态接口
- 支持 802.1CM (802.3br/802.1bu) 抢占功能（适用于 MAC+PCS/PMA 64 位 Base-R）

## 10G 支持的功能特性

- 完整的 MAC 和 PCS 功能
- 基于 IEEE 802.3 第 49 条的 Base-KR 模式
- 暂停处理
- 可选 64 位或 32 位 AXI4-Stream 用户接口
- 可选独立 MAC，含 64 位 AXI4-Stream 接口和 XGMII 管脚输出
- 可选第 73 条自动协商
- 可选第 72.6.10 条链路训练
- 可选第 74 条 FEC - 简化循环代码 (2112, 2080)
- 仅限 PCS 版本，含 XGMII 接口
- 可选 AXI4-Lite 控制接口和状态接口
- 统计数据和诊断
- 66 位 SerDes 接口
- 定制前导码和可调整的帧间间距
- 支持 802.1CM (802.3br/802.1bu) 抢占功能（适用于 MAC+PCS/PMA 64 位 Base-R）

## 10G/25G 运行时可切换 IP 特性

- 完整的 MAC 和 PCS 功能特性

- 基于 IEEE 802.3 第 49 条的 Base-KR 模式
- 统计数据和诊断
- 使用赛灵思 GTY 收发器的 66 位 SerDes 接口
- 定制前导码和可调整的帧间间距
- 暂停处理
- 64 位 AXI4-Stream 接口
- 可选第 73 条自动协商
- 可选第 72 条链路训练
- 可选第 74 条 FEC

表 1：特性兼容性矩阵 (GTY)

变体	用户接口	MAC	PCS	暂停处理	自动协商和链路训练 <sup>3</sup>	第 74 条 FEC	第 108 条 RS-FEC	IEEE 1588 时间戳
10G MAC (含 PCS)	64 位 AXI4-Stream	X	X	X	X	X		X
低时延 10G MAC (含 PCS)	64 位 AXI4-Stream	X	X					
低时延 10G MAC (含 PCS)	32 位 AXI4-Stream	X	X		X <sup>2</sup>			
低时延 25G MAC (含 PCS)	64 位 AXI4-Stream	X	X					
25G MAC (含 PCS)	64 位 AXI4-Stream	X	X	X	X	X	X	X
运行时可切换 10G/25G MAC+PCS	64 位 AXI4-Stream	X	X	X	X	X	X	X
10G 仅限 PCS	XGMII		X		X	X		X <sup>1</sup>
25G 仅限 PCS	25GMII		X		X	X	X	X <sup>1</sup>
运行时可切换 10/25G 仅限 PCS	XGMII/25GMII		X		X	X	X	X <sup>1</sup>
10G 仅限 MAC	64 位 AXI4-Stream	X		X				
抢占 (802.1CM) 仅限 10G/25G	64 位 AXI4-Stream	X	X					X

## 注释：

1. 仅限 PCS 配置仅支持双步时间戳。
2. 仅限自动协商逻辑。
3. Versal 器件系列不支持自动协商和链路训练。

表 2：特性兼容性矩阵 (GTM)

变体	用户接口	MAC	PCS	暂停处理	自动协商和链路训练 <sup>2</sup>	第 74 条 FEC	第 108 条 RS-FEC	IEEE 1588 时间戳
10G MAC (含 PCS)	64 位 AXI4-Stream	X	X	X		X		X
低时延 10G MAC (含 PCS)	64 位 AXI4-Stream	X	X					

表 2：特性兼容性矩阵 (GTM) (续)

变体	用户接口	MAC	PCS	暂停处理	自动协商和链路训练 <sup>2</sup>	第 74 条 FEC	第 108 条 RS-FEC	IEEE 1588 时间戳
低时延 25G MAC (含 PCS)	64 位 AXI4-Stream	X	X					
25G MAC (含 PCS)	64 位 AXI4-Stream	X	X	X		X	X	X
10G 仅限 PCS	XGMII		X			X		X <sup>1</sup>
25G 仅限 PCS	25GMII		X			X	X	X <sup>1</sup>
抢占(802.CM) 仅限 10G/25G	64 位 AXI4-Stream	X	X					X

注释：

1. 仅限 PCS 配置仅支持双步时间戳。
2. Versal 器件系列不支持自动协商和链路训练。

## 应用

IEEE 802.3 标准支持局域网 (LAN) 应用采用多种不同以太网速度，该标准最新添加了 25 Gb/s 速度。互连器件能否以 25 Gb/s 以太网速率运行对于下一代数据中心网络尤其重要，这主要体现在：

- 为了跟上不断增加的 CPU 和存储带宽需求，机架式或刀片式服务器需支持来自其网络接口卡 (NIC) 或板载 LAN (LOM) 网络端口的聚合吞吐量超过 10 Gb/s (单通道) 或 20 Gb/s (双通道)；
- 由于端点带宽增加，架顶式 (TOR) 或刀片式交换机的上行链路需要从 40 Gb/s (四通道) 升级到 100 Gb/s (四通道)，同时最好保留同样的每通道分接能力；
- 由于预计将采用 100GBASE-CR4/KR4/SR4/LR4，因此 SerDes 和布线技术均已在开发和部署中，以支持每个物理通道、双绞线或光纤 25 Gb/s 的传输速率。

## 许可和订购

### 许可证检查器

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。

## 许可证类型

### 10G/25G Ethernet PCS/PMA (10G/25G BASE-R)

根据[赛灵思最终用户许可条款](#)，此赛灵思 IP 模块随附赛灵思 Vivado Design Suite 免费提供。如需获取有关此模块及其它赛灵思 IP 模块的信息，请访问[赛灵思知识产权](#)页面。有关其它赛灵思 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](#)。如需了解更多信息，请访问[10G/25G Ethernet Subsystem 页面](#)。

### 独立 10G/25G Ethernet MAC and PCS/PMA (10G/25G MAC + 10G/25G BASE-R/KR)、10G/25G BASE-KR 和 10G TSN IEEE802.1CM

**注释：**10G/25G Ethernet MAC + BASE-R, 10GBASE-KR/25GBASE-KR and 10G TSN IP 功能特性需要单独付费许可。

根据[赛灵思核许可协议](#)的条款，这些赛灵思 IP 模块随附产品提供。该模块作为 Vivado Design Suite 的一部分提供。必须购买一个或多个核许可证才能获取仿真和硬件中的所有核功能的完整访问权限。有关定价和可用性的信息，请联系您[当地的赛灵思销售代表](#)。如需了解更多信息，请访问[10G/25G Ethernet Subsystem 页面](#)。

## 订购信息

要购买其中任何 IP 核，请联系您当地的[赛灵思销售代表](#)，并提供下表中相应的器件编号：

表 3：订购信息

描述	器件编号	许可证密钥
<ul style="list-style-type: none"><li>10G/25G Ethernet MAC + BASE-R PCS/PMA (64 位)</li><li>10E MAC + BASE-R PCS/PMA (32 位)</li></ul>	EF-DI-25GEMAC-PROJ <sup>1</sup> EF-DI-25GEMAC-SITE <sup>1</sup>  <b>注释：</b> 这些器件编号不包含由 FEC 和 AN/LT 组成的 BASE-KR/CR/SR 功能。要包含对该功能的支持，请订购 EF-DI-25GBASE-KR-PROJ 或 EF-DI-25GBASE-KR-SITE。	xxv_eth_mac_pcs x_eth_mac
<ul style="list-style-type: none"><li>25GBASE-KR PCS/PMA (CL108 RS-FEC、CL74 FE 和 AN)<sup>2</sup></li><li>10GBASE-KR PCS/PMA (CL74 FEC 和 AN) 仅限独立 PCS/PMA</li></ul>	EF-DI-25GBASE-KR-PROJ <sup>2,3</sup> EF-DI-25GBASE-KR-SITE <sup>2,3</sup>  <b>注释：</b> 10G/25G Ethernet MAC 单独售卖。要包含赛灵思 MAC，还请订购 EF-DI-25GEMAC-PROJ 或 EF-DI-25GEMAC-SITE。	xxv_eth_basekr ieee802d3_25g_rs_fec_full ieee802d3_25g_rs_fec_only <sup>4</sup>
包含第 1 和第 2 行中的所有配置，并且包含 10G/25G Ethernet MAC + BASE-KR PCS/PMA (64 位)。	EF-DI-25GEMAC-PROJ <sup>1</sup> EF-DI-25GEMAC-SITE <sup>1</sup> 和 EF-DI-25GBASE-KR-PROJ <sup>2,3</sup> EF-DI-25GBASE-KR -SITE <sup>2,3</sup>	xxv_eth_mac_pcs x_eth_mac xxv_eth_basekr ieee802d3_25g_rs_fec_full ieee802d3_25g_rs_fec_only <sup>4</sup>
<ul style="list-style-type: none"><li>25G/10GBASE-R 仅限 PCS/PMA</li><li>10GBASE-R 仅限 PCS/PMA</li><li>可切换 25G 和 10G BASE-R PCS/PMA</li></ul>	随附于 Vivado 设计工具。无需购买。	无许可证密钥

表 3：订购信息 (续)

描述	器件编号	许可证密钥
<ul style="list-style-type: none"><li>· 25G/10G Ethernet MAC+PCS/PMA</li><li>· 10G/25G Ethernet TSN 802.1CM（抢占）功能特性</li></ul>	EF-DI-25G-TSN-802-1-CM-PRO  <b>注释：</b> 捆绑中包含 Ethernet MAC IP。Vivado 中当前仅支持 10GE TSN。有关 25G 咨询，请发送电子邮件至 <a href="mailto:ethernet_mgmt@xilinx.com">ethernet_mgmt@xilinx.com</a> 。	xxv_tsn_802d1cm xxv_eth_mac_pcs x_eth_mac

**注释：**

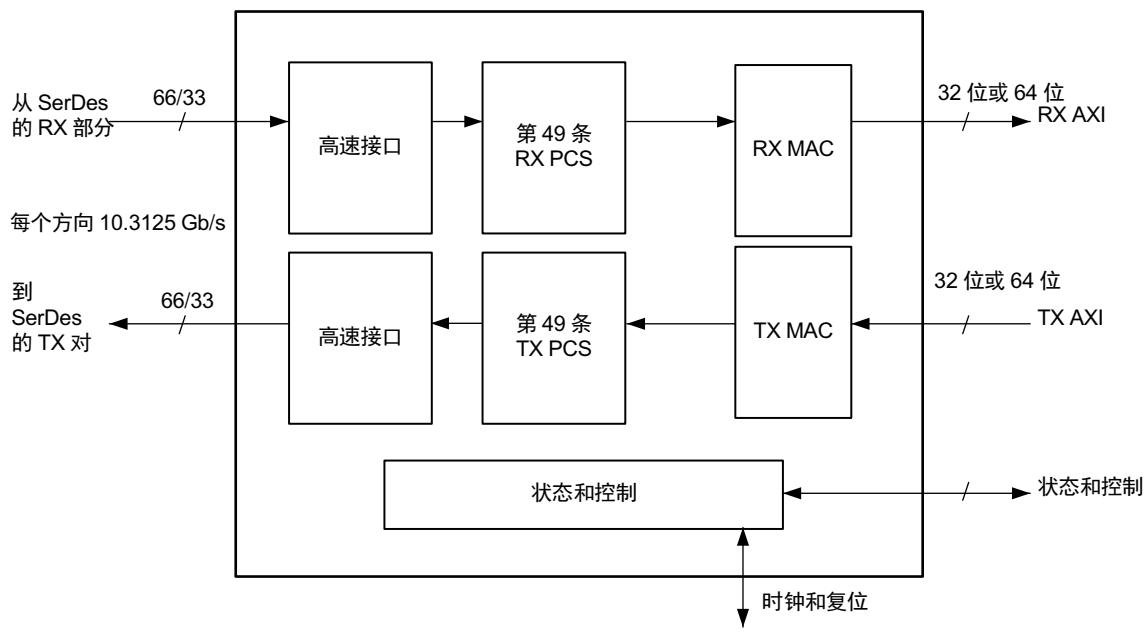
1. 包含对适用于 7 系列和 UltraScale™ 器件的旧 [10 Gigabit Ethernet Media Access Controller - 10GEMAC](#)（密钥名称：ten\_gig\_eth\_mac）的访问权。
2. 用于 25GBASE-CR、25GBASE-KR 或 25GBASE-SR 应用。
3. 包含对适用于 7 系列和 UltraScale 器件的 [10 Gigabit Ethernet PCS/PMA with FEC/Auto-Negotiation - 10GBASE-KR](#)（密钥名称：ten\_gig\_eth\_pcs\_pma\_basekr）的访问权。
4. iee802d3\_25g\_rs\_fec\_only 许可证密钥仅为定制应用提供转码旁路模式支持。如需了解更多信息，请参阅《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》([PG217](#))（需注册）。如需申请访问权，请发送电子邮件至 [ethernet\\_mgmt@xilinx.com](mailto:ethernet_mgmt@xilinx.com)。如不需要定制独立仅限 FEC 实现（无 MAC 和/或 PCS），请忽略 Vivado 中的 iee802d3\_25g\_rs\_fec\_only 许可证密钥警告信息。

## 产品规格

下图分别显示了 10G/25G High Speed Ethernet Subsystem 的各模块框图。

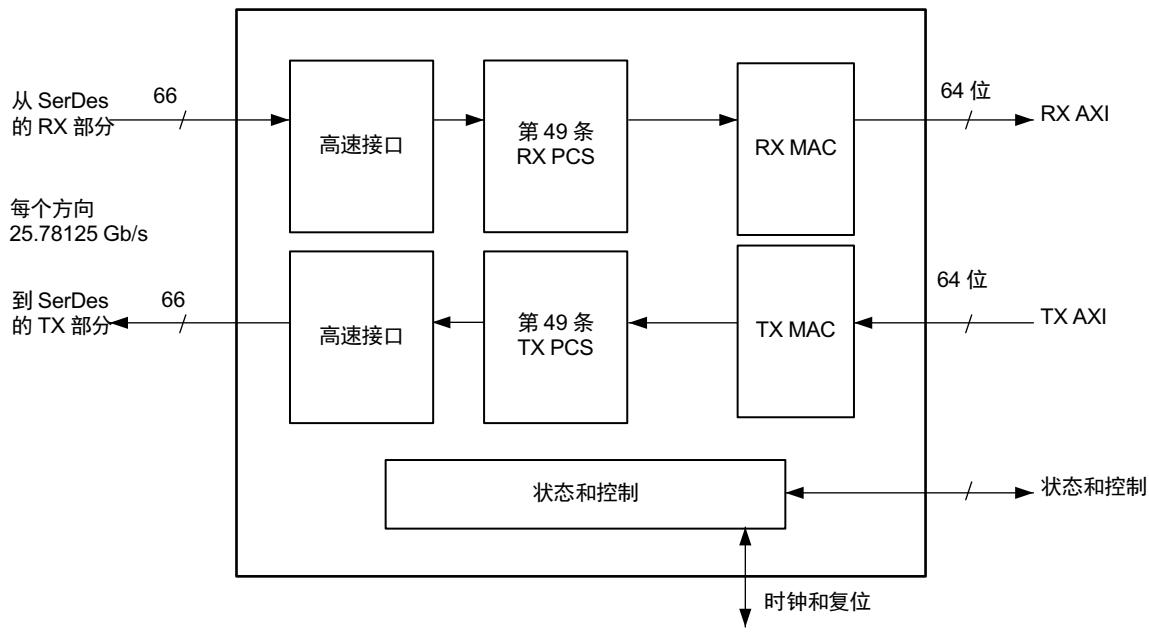
注释：GT 收发器未予显示。

图 1：10 Gb/s 核模块框图



X15162-101122

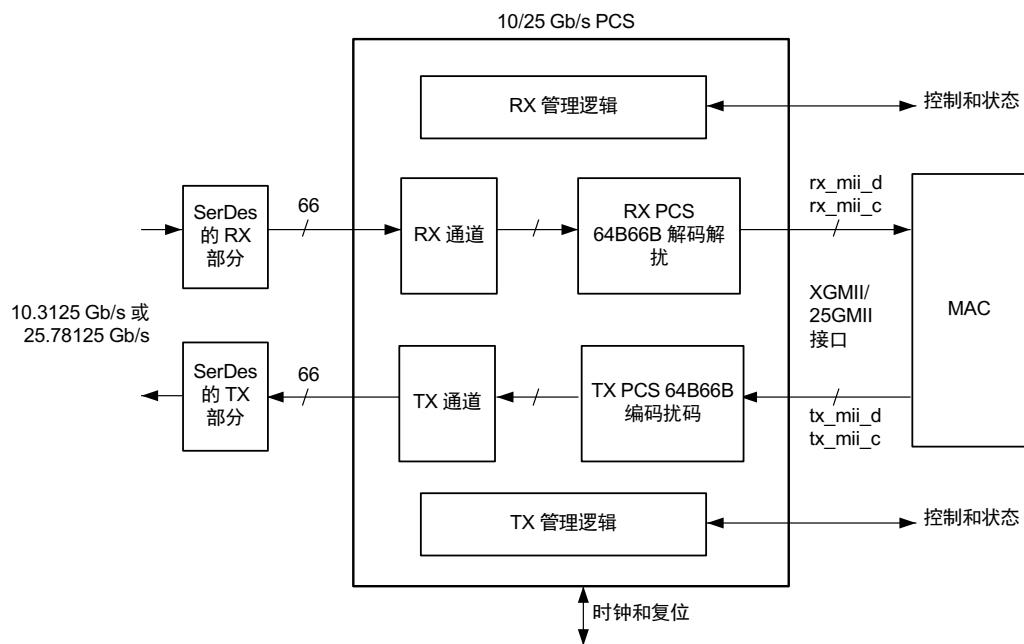
图2：25 Gb/s核模块框图



X25530-101122

核的仅限 PCS 变体也同样可用。其模块框图如下图所示。

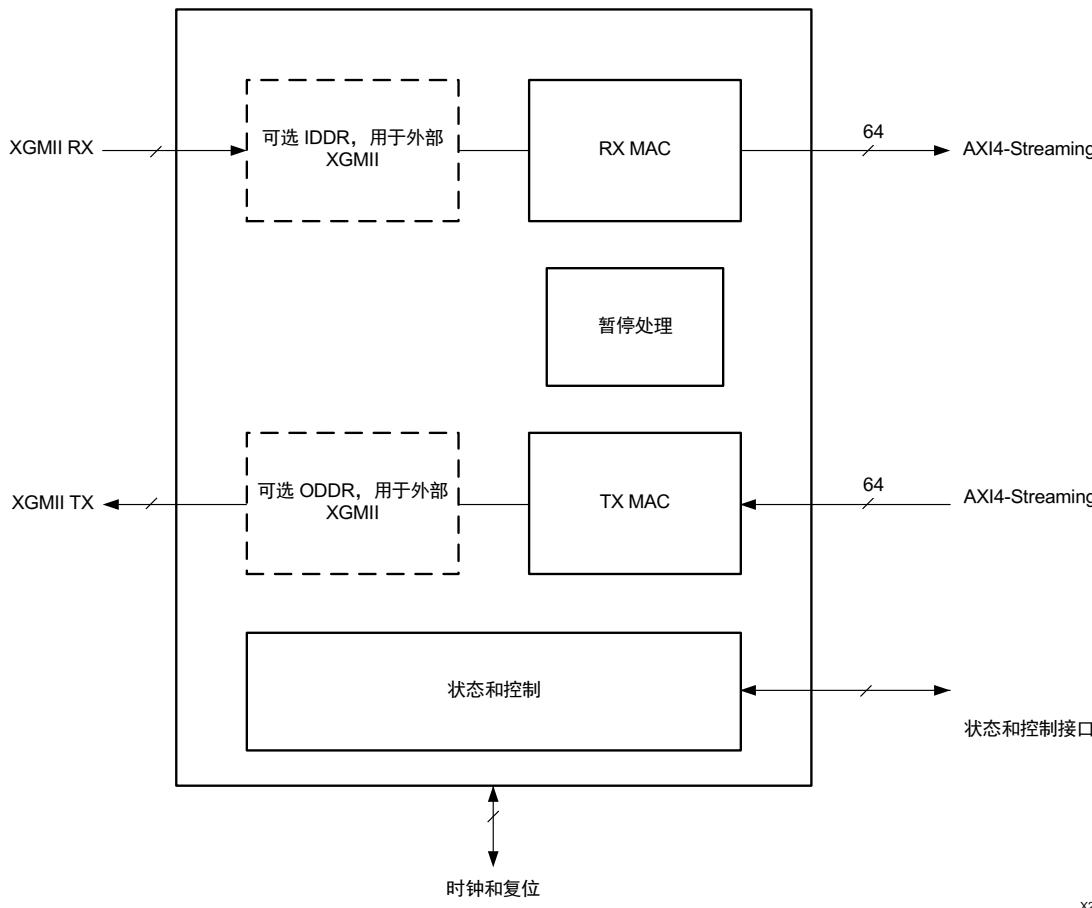
图3：仅限 PCS 的核变体的模块框图



X15163-101122

此外，您可选择生成 64 位独立版本 MAC，用于 10 Gb/s 运行。其模块框图如下图所示。

图 4：64 位独立版本 MAC，用于 10 Gb/s 运行



## 标准

10G/25G Ethernet Subsystem 是根据 25G 和 50G 以太网联盟和 IEEE 802.3 标准中指定的标准进行设计的。

## 性能和资源使用情况

如需获取有关性能与资源使用情况的完整详情，请访问[性能与资源使用情况网页](#)。

## 时延

下表提供了针对 10G/25G Ethernet Subsystem 测量所得的时延信息。此信息由默认配置下核的 RX 和 TX 时延组合而成，不包括收发器中的时延。

表 4：时延

核	总时延 (ns)	TX 时延 (ns)	RX 时延 (ns)	用户总线宽度 (位)	核时钟频率 (MHz)
10GE MAC + PCS	188.82	67.2	121.62	64	156.25
25GE MAC + PCS	80.41	29.58	50.83	64	390.625
10G PCS (BASE-KR)	332.8	172.8	160	64	156.25
25G PCS	138.25	71.68	66.57	64	390.625
10G PCS (BASE-R)	179.2	19.2	160	64	156.25
10GE MAC + PCS	54.4	28.8	25.6	32	312.5
10GE PCS	48	19.2	28.8	32	312.5

注释：

- 这些数值包含 RX 和 TX 互连结构逻辑，但不包含 GT。

## 端口描述 - MAC+PCS 变体

下表列出了含集成 MAC 和 PCS 的 10G/25G Ethernet Subsystem 的端口。可在 `wrapper.v` 层级中找到这些信号。这些端口适用于 64 位集成 MAC+PCS（适用于 25 Gb/s 和 10 Gb/s 线速率）和低时延 32 位集成 MAC + PCS（适用于 10 Gb/s 线速率）。包含 AXI 寄存器接口时，其中部分端口使用寄存器（而非宽边总线）来访问。

## 收发器接口

下表显示了 10G/25G Ethernet Subsystem 的收发器 I/O 端口。请参阅“时钟设置”以获取有关每个时钟域的详细信息。

表 5：收发器 I/O

名称	I/O	描述	时钟域
gt_tx_reset	输入	复位，适用于千兆位收发器 (GT) TX。	异步
gt_rx_reset	输入	GT RX 复位。	异步
ctl_gt_reset_all	输入	高电平有效异步复位，适用于收发器启动有限状态机 (FSM)。请注意，此信号还会启动整个 10G/25G Ethernet Subsystem 的复位序列。	异步
refclk_n0	输入	SerDes 的差分参考时钟输入，负相。	请参阅 <a href="#">时钟设置</a> 。
refclk_p0	输入	SerDes 的差分参考时钟输入，正相。	请参阅 <a href="#">时钟设置</a> 。
rx_serdes_data_n0	输入	来自数据线的串行数据；差分信号的负相	请参阅 <a href="#">时钟设置</a> 。
rx_serdes_data_p0	输入	来自数据线的串行数据；差分信号的正相	请参阅 <a href="#">时钟设置</a> 。
tx_serdes_data_n0	输出	到数据线的串行数据；差分信号的负相。	请参阅 <a href="#">时钟设置</a> 。
tx_serdes_data_p0	输出	到数据线的串行数据；差分信号的正相。	请参阅 <a href="#">时钟设置</a> 。
tx_serdes_clk	输出	此信号存在时，与 tx_clk_out 相同。	请参阅 <a href="#">时钟设置</a> 。

## AXI4-Stream 接口

10G/25G High Speed Ethernet Subsystem IP 核为 10G 运行速率的系统提供了 32 位和 64 位 AXI4-Stream 接口选项，为 25G 运行速率的系统则仅提供 64 位选项。对于 10G 系统，提供了 32 位和 64 位接口。对于 25G 系统，则仅提供 64 位接口。

### AXI4-Stream 时钟和复位

表 6：AXI4-Stream 接口时钟信号和复位信号

名称	I/O	描述	时钟域
rx_clk_out	输出	rx_serdes_clk。GT 与核之间的时钟 RX 接口。在低时延缓冲器旁路模式下，该时钟还会对 AXI4-Stream RX 接口进行时钟设置。	请参阅 <a href="#">时钟设置</a> 。
tx_clk_out	输出	对 TX AXI4-Stream 接口和完整 TX 数据路径进行时钟设置。	请参阅 <a href="#">时钟设置</a> 。
rx_reset	输入	RX 电路的复位。此信号为高电平有效 (1 = 复位)，并且必须保持高电平，直至 clk 达到稳定状态为止。该核可处理将 rx_reset 输入同步到核中相应的时钟域的操作。	异步
tx_reset	输入	TX 电路的复位。此信号为高电平有效 (1 = 复位)，并且必须保持高电平，直至 clk 达到稳定状态为止。该核可处理将 tx_reset 输入同步到核中相应的时钟域的操作。	异步
rx_core_clk	输入	rx_core_clk 信号用于对 AXI4-Stream 接收接口进行时钟设置。不含 FIFO 时，此信号必须由 rx_clk_out 驱动。包含 FIFO 时，rx_core_clk 可由 tx_clk_out、rx_clk_out 或相同频率的其它异步时钟驱动。	rx_core_clk

### AXI4-Stream 发射接口

下表显示了 AXI4-Stream 发射接口信号。

表 7：AXI4-Stream 发射接口信号

信号	I/O	描述	时钟域
tx_axis_tdata[63 or 31:0]	输入	AXI4-Stream 数据。32 位和 64 位接口均可使用。总线宽度取决于选择的是 64 位接口还是 32 位接口。	tx_clk_out
tx_axis_tkeep[7:0 or 3:0]	输入	AXI4-Stream 数据控制。总线宽度取决于选择的是 64 位接口还是 32 位接口。	tx_clk_out
tx_axis_tvalid	输入	AXI4-Stream 数据有效输入	tx_clk_out
tx_axis_tuser	输入	AXI4-Stream 用户边带接口。 <ul style="list-style-type: none"><li>· 1 指示包无效</li><li>· 0 指示包有效</li></ul>	tx_clk_out
tx_axis_tlast	输入	AXI4-Stream 信号，指示以太网包结束。	tx_clk_out
tx parityin[7:0]	输入	AXI4-Stream 用户生成的奇偶校验。遵循与 tx_axis_tkeep 相同的数据通道映射。	tx_clk_out
tx_axis_tready	输出	AXI4-Stream 确认信号，用于指示开始数据传输。	tx_clk_out

### 数据通道映射 - TX

对于发射数据 tx\_axis\_tdata，端口按逻辑分割为通道 0 到通道 3（对应 32 位接口）或通道 0 到通道 7（对应 64 位接口），且 tx\_axis\_tkeep 数据字的对应位用于在 tx\_axis\_tdata 上发出有效数据信号。

表 8: tx\_axis\_tdata 通道 - 32 位

通道/ tx_axis_tkeep	tx_axis_tdata[31:0]
0	7:0
1	15:8
2	23:16
3	31:24

表 9: tx\_axis\_tdata 通道 - 64 位

通道/ tx_axis_tkeep	tx_axis_tdata[63:0]
0	7:0
1	15:8
2	23:16
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

## 正常发射

下列 32 位变体和 64 位变体的图示中分别显示了对应的正常帧传输的时序。当客户端发射帧时，它会断言 tx\_axis\_tvalid 有效并将 tx\_axis\_tdata 和 tx\_axis\_tkeep 中的数据和控制布局在同一个时钟周期内。核接受此数据时（通过 tx\_axis\_tready 断言有效来指示），客户端必须提供下一个周期的数据。如果核未断言 tx\_axis\_tready 有效，则客户端必须保持当前有效数据值，直至其断言有效为止。通过 tx\_axis\_tlast 断言有效并保持 1 个周期来向核指示包结束。正确设置 tx\_axis\_tkeep 的位，以指示最终数据传输中的有效字节数。通过同时断言 tx\_axis\_tuser 有效来指示数据包无效。

在 tx\_axis\_tlast 断言无效后，所有数据与控制都被视为无效，直至下一次 tx\_axis\_tvalid 断言有效为止。

图 5: 正常帧传输 - 32 位

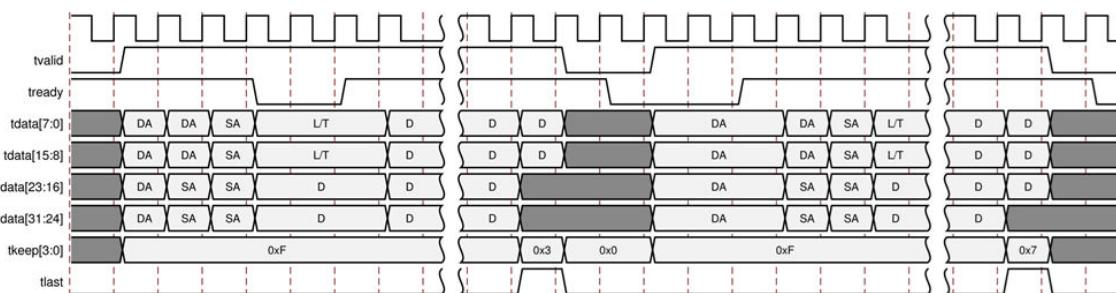
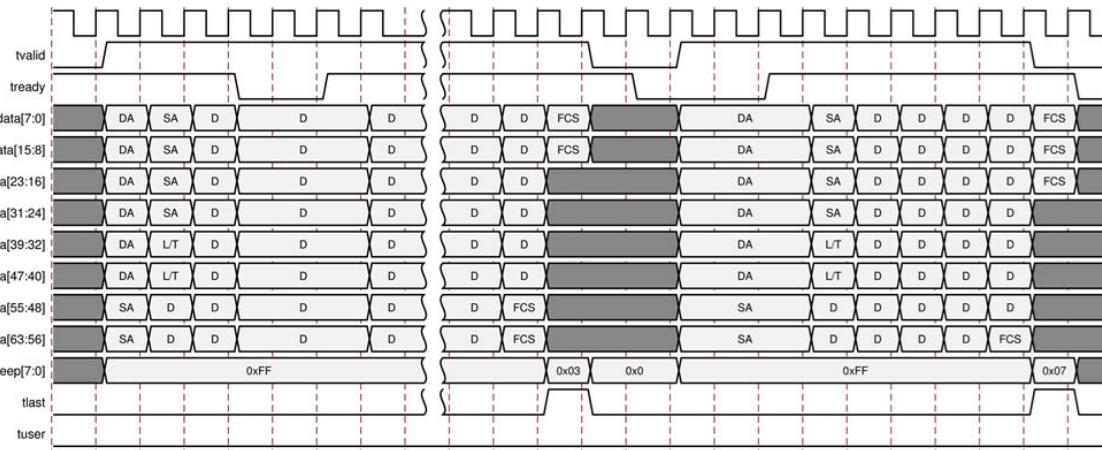


图6：正常帧传输 - 64位



### 不间断连续传输

在 AXI4-Stream 发射接口上之所以能够连续传输数据，原因在于 `tx_axis_tvalid` 信号能够连续保持高电平，并且包边界只能通过断言 `tx_axis_tlast` 信号有效（表示以太网包结束）来定义。不过，核可以根据需要通过断言 `tx_axis_tready` 确认信号无效来对客户端数据进行节流。请参阅以下两张图。在核断言 `tx_axis_tready` 确认信号无效时，客户端数据逻辑可以用有效的数据来更新 AXI4-Stream 接口。不过，在 `valid` 断言有效并且新数据置于 AXI4-Stream 上之后，它应保留不动，直到核断言 `tx_axis_tready` 信号有效为止。

图7：发射客户端接口上的不间断连续传输 - 32位

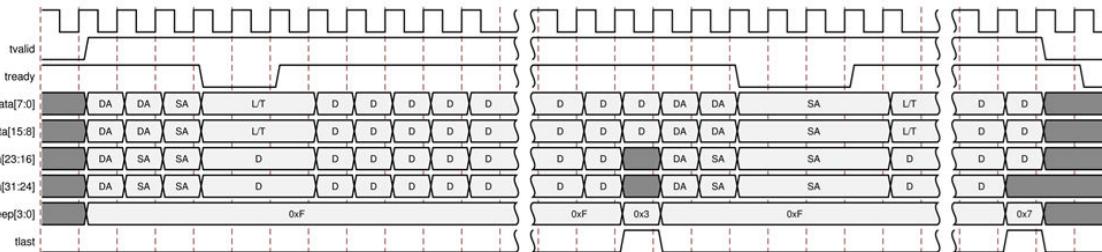
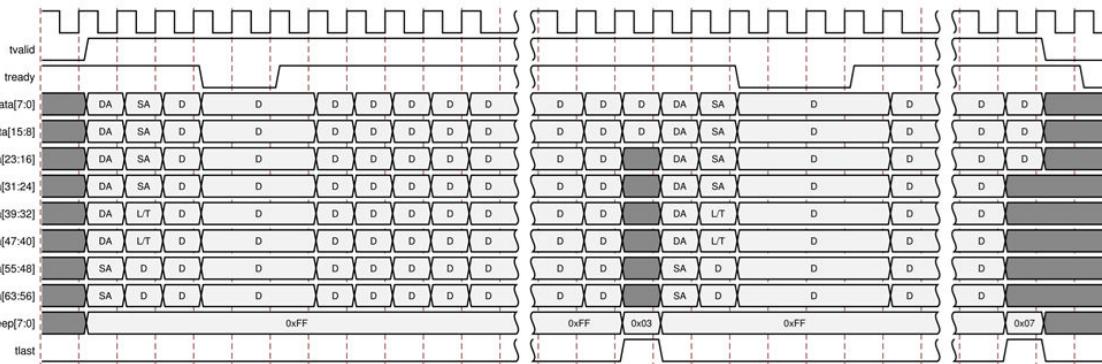


图8：发射客户端接口上的不间断连续传输 - 64位



## 中止发射

客户端接口数据包传输中止称为欠载 (underrun)。如果 AXI 发射客户端接口中的 FIFO 在帧完成之前清空，则可能发生这种情况。

通过以下任一方法即可向核指明发生发射中止：

- 显式错误，由于 `tx_axis_tuser` 断言为高电平有效且 `tx_axis_tlast` 为高电平而导致帧传输中止。
- 隐式欠载，由于 `tx_axis_tvalid` 断言无效且 `tx_axis_tlast` 未断言有效而导致帧传输中止。

在帧发射期间如果发生以上任一情况，则该核会在数据串流中插入错误代码，以将当前帧标记为错误帧。由客户端负责根据需要对异常中止的帧进行重新排队并发射。

## AXI4-Stream 接收接口

表 10：AXI4-Stream 接收接口信号

信号	I/O	描述	时钟域
<code>rx_axis_tdata[63 or 31:0]</code>	输出	AXI4-Stream 数据，连接到上层。总线宽度取决于选择的是 64 位或 32 位。	<code>rx_core_clk</code>
<code>rx_axis_tkeep[7 or 3:0]</code>	输出	AXI4-Stream 数据控制，连接到上层。总线宽度取决于选择的是 64 位或 32 位。	<code>rx_core_clk</code>
<code>rx_axis_tvalid</code>	输出	AXI4-Stream 数据有效	<code>rx_core_clk</code>
<code>rx_axis_tuser</code>	输出	AXI4-Stream 用户边带接口。 1 指示接收到的包无效。 0 指示接收到的数据包有效。	<code>rx_core_clk</code>
<code>rx_axis_tlast</code>	输出	AXI4-Stream 信号，指示包结束。	<code>rx_core_clk</code>
<code>rx parityout[7:0]</code>	输出	AXI4-Stream 核生成的奇偶校验。遵循与 <code>rx_axis_tkeep</code> 相同的数据通道映射。	<code>rx_core_clk</code>

### 数据通道映射 - RX

对于接收数据 `rx_axis_tdata`，端口按逻辑分割为通道 0 到通道 3（对应 32 位接口）或通道 0 到通道 7（对应 64 位接口），且 `rx_axis_tkeep` 数据字的对应位用于在 `rx_axis_tdata` 上发出有效数据信号。

表 11：`rx_axis_tdata` 通道 - 32 位

Lane/ <code>rx_axis_tkeep</code>	<code>rx_axis_tdata[31:0]</code> 位
0	7:0
1	15:8
2	23:16
3	31:24

表 12：`rx_axis_tkeep` 通道 - 64 位

通道/ <code>rx_axis_tkeep</code>	<code>rx_axis_tdata Bits</code>
0	7:0
1	15:8
2	23:16

表 12: rx\_axis\_tkeep 通道 - 64 位 (续)

通道/ rx_axis_tkeep	rx_axis_tdata Bits
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

## 正常帧接收

客户端必须准备好随时接受数据；由于核中没有缓冲机制，因此接收客户端不允许出现时延。

在帧接收期间，`rx_axis_tvalid` 断言有效，以指示有效的帧数据正通过 `rx_axis_tdata` 传输到客户端。所有 `rx_axis_tkeep` 位均设为 1，表示整个帧中的所有字节始终有效，但在 `rx_axis_tlast` 断言有效时该帧进行最终传输期间除外。在帧数据的最终传输期间，`rx_axis_tkeep` 位使用上述映射来指示帧的最终有效字节。最终传输的有效字节始终来自于 `rx_axis_tdata[7:0]` (`rx_axis_tkeep[0]`)，因为以太网帧数据是连续的并且先接收最低有效字节。

仅在完成所有帧校验后，随传输的最终字节一起，`rx_axis_tlast` 断言有效并且 `rx_axis_tuser` 断言无效。这发生在接收到帧校验序列 (FCS) 字段后。核使 `rx_axis_tuser` 信号保持处于断言无效状态，以指示已成功接收到帧，并且客户端应对帧进行分析。这也表示包结束，以 `rx_axis_tlast` 断言有效并保持 1 个周期来表示。

图 9: 正常帧接收 - 32 位

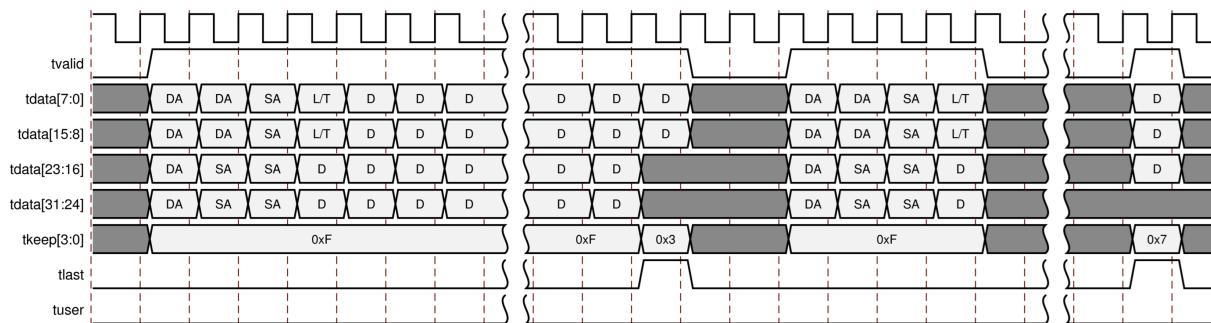
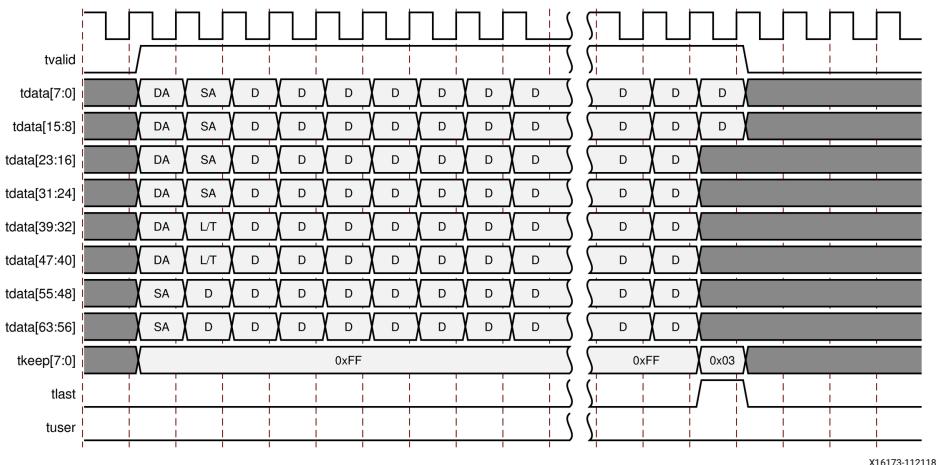


图10：正常帧接收 - 64位



X16173-112118

### 帧接收时出错

帧接收不成功可能存在如下形式：超短帧，或者含错误 FCS 的帧。在此情况下，接收到的帧为坏帧，并且在此帧结束时会向客户端断言 rx\_axis\_tuser 信号有效。随后，客户端负责丢弃已为此帧传输的数据。

以下条件会导致 rx\_axis\_tlast 断言有效并且 rx\_axis\_tuser = 1 (表示 bad\_frame)：

- 发生 FCS 错误。
- 数据包长度小于 64 字节（大小过小或者碎片帧）。
- 接收到的帧的长度超过已编程的最大传输单元 (MTU) 大小。
- 接收到的任何控制帧的长度都并非恰好为最小帧长度。
- XGMII 数据串流包含错误代码。

图11：帧接收时出错 - 32位

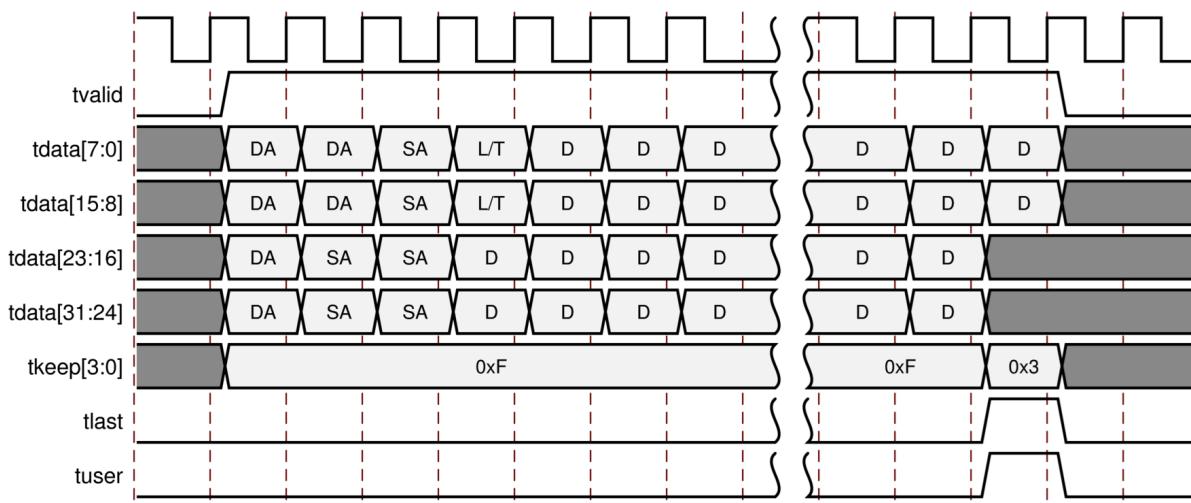
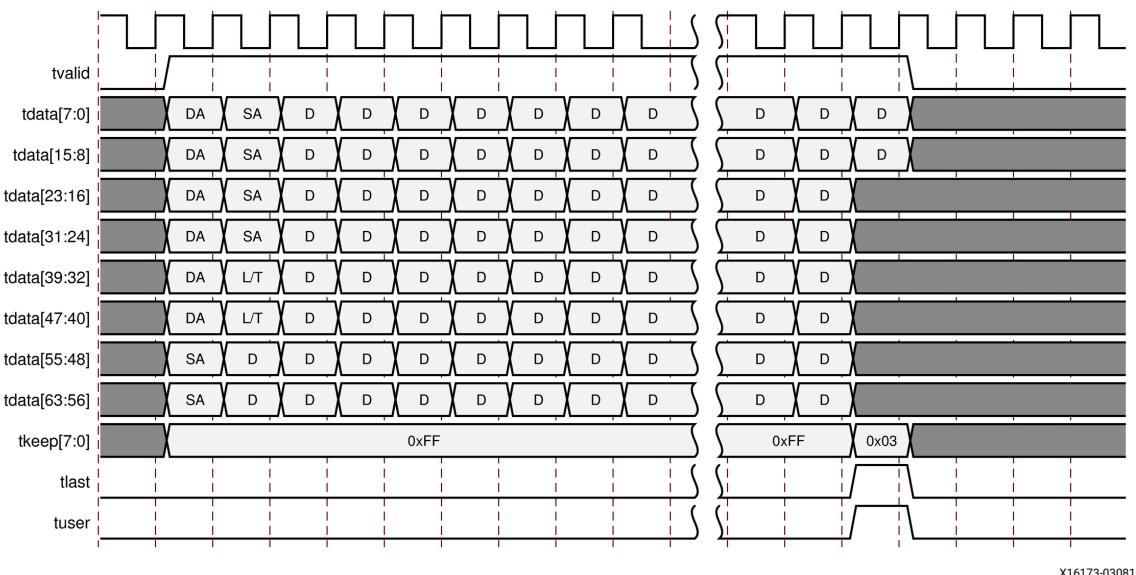


图12：帧接收时出错 - 64位



X16173-030816

## AXI4-Stream 控制端口和状态端口

表13：AXI4-Stream 接口 - TX 路径控制信号和状态信号

名称	I/O	描述	时钟域
ctl_tx_custom_preamble_enable	输入	此信号断言有效时，支持使用 tx_preamblein 作为自定义前导码，而不是插入标准前导码。	tx_clk_out
tx_preamblein [55:0]	输入	此信号为定制前导码，它应为独立输入端口，而不是与数据保持一致。 包起始期间，此信号应有效。	tx_clk_out
ctl_tx_ipg_value[3:0]	输入	此信号为可选。ctl_tx_ipg_value 用于定义在 AXI4-Stream 包之间插入的目标平均最小包间间隔 (IPG，以字节为单位)。有效值范围为 8 到 12。ctl_tx_ipg_value 可编程为范围介于 0 到 7 之间的值，但在此情况下，它解释为 8 (最小有效值)。	tx_clk_out
ctl_tx_enable	输入	TX 使能 (启用)。此信号采样为 1 时用于启用数据发射。当此信号采样为 0 时，核仅发射空闲信号。仅当数据发射到的接收器 (即其它器件中的接收器) 完全同步并且已准备好接收数据 (即，其它器件当前未发送远端故障状况) 后，此输入才应设置为 1。否则，可能发生数据丢失。如果发射数据包时此信号设置为 0，那么当前数据包发射完成后，核将停止发射任何其它数据包。	tx_clk_out
ctl_tx_send_rfi	输入	发射远端故障指示 (RFI) 代码字。如果此输入采样为 1，那么 TX 路径仅发射远端故障 (Remote Fault) 代码字。此输入应设置为 1，直至 RX 路径已完全同步并且已准备好接受来自链路伙伴的数据为止。	tx_clk_out
ctl_tx_send_lfi	输入	发射本地故障指示 (LFI) 代码字。优先于远端故障指示 (RFI)。	tx_clk_out
ctl_tx_send_idle	输入	发射空闲代码字。如果此输入采样为 1，那么 TX 路径仅发射空闲 (Idle) 代码字。当伙伴器件正在发送 RFI 代码字时，此输入应设置为 1。	tx_clk_out

表 13：AXI4-Stream 接口 - TX 路径控制信号和状态信号（续）

名称	I/O	描述	时钟域
ctl_tx_fcs_ins_enable	输入	启用由 TX 核执行 FCS 插入。如果该位设为 0，该核不会将 FCS 添加到包中。如果该位设为 1，那么该核会计算 FCS 并将 FCS 添加到包中。在不同数据包之间不得动态更改此输入。	tx_clk_out
ctl_tx_ignore_fcs	输入	启用由 TX 核在 AXI4-Stream 接口上执行 FCS 纠错。仅当 ctl_tx_fcs_ins_enable 为低电平时，此输入才有效。如果此输入为低电平，并且发射的数据包中含有无效 FCS，则不会将其作为有效包丢弃。如果此输入为高电平，则含无效 FCS 的包将被作为有效包丢弃。  注释：报告的统计数据中显示不含 FCS 错误。	tx_clk_out
ctl_tx_parity_err_response	输入	TX 核发出奇偶校验错误响应。如果该位设为 0，那么检测到任何奇偶校验错误时，该核都不会执行任何操作。如果该位设为 1，那么该核会给传出的 FCS 添加 stamp 标记（即按位反转）并断言 stat_tx_bad_fcs 有效。当检测到奇偶校验错误时，stat_tx_bad_parity 输出即断言有效，与 ctl_tx_parity_err_response 配置无关。	tx_clk_out

表 14：AXI4-Stream 接口 - RX 路径控制信号和状态信号

名称	I/O	描述	时钟域
rx_preambleout [55:0]	输出	这是前导码，原先在前版本中与数据保持一致，当前版本中则改为独立输出。	rx_core_clk
ctl_rx_enable	输入	RX 使能（启用）。在正常操作期间，此输入必须设置为 1。当此输入设置为 0 时，RX 完成接收当前包（如果有）之后，就会停止接收包并阻止 PCS 继续对传入数据进行解码。在此模式下不报告任何统计数据，并且 AXI4-Stream 接口处于空闲状态。	rx_clk_out
ctl_rx_check_preamble	输入	此输入断言有效时，会使 MAC 检查接收到的帧的前导码。	rx_clk_out
ctl_rx_check_sfd	输入	此输入断言有效时，会使 MAC 检查接收到的帧的帧起始定界符。	rx_clk_out
ctl_rx_force_resync	输入	RX 强制再同步输入。此信号用于强制 RX 路径进行复位和再同步。值为 1 即强制执行复位操作。值为 0 允许正常操作。请注意，此输入通常应为低电平（Low）并且应只能对此输入进行脉冲（单周期最小脉冲）。	rx_clk_out
ctl_rx_delete_fcs	输入	启用由 RX 核执行 FCS 移除。如果此位设置为 0，则该核不会移除传入包的 FCS。如果此位设置为 1，则该核将会删除接收包的 FCS。请注意，针对长度小于等于 8 字节的包，不删除 FCS。仅当对应的复位输入断言有效时，才应更改此输入。	rx_clk_out
ctl_rx_ignore_fcs	输入	启用由 RX 核在 AXI4-Stream 接口上执行 FCS 纠错。如果此位设置为 0，对于接收到的含 FCS 错误的包，将随在上一次传输期间（rx_axis_tuser 和 rx_axis_tlast 均采样为 1）断言有效的 rx_axis_tuser 管脚一起发送。如果此位设置为 1，则该核不会在 AXI4-Stream 接口处标记 FCS 错误。  注释：报告的统计数据中显示此数据包有效。但 stat_rx_bad_fcs 信号则会报告错误。	rx_clk_out

表 14: AXI4-Stream 接口 - RX 路径控制信号和状态信号 (续)

名称	I/O	描述	时钟域
ctl_rx_max_packet_len[14:0]	输入	长度超过该值的任何包都将被视为大小过大。如果包的大小大于该值，则该包将被截位至该值，并且 rx_axis_tuser 信号随 rx_axis_tlast 信号一起断言有效。小于 4 字节的包将被丢弃。 ctl_rx_max_packet_len[14] 保留并且必须设置为 0。	rx_clk_out
ctl_rx_min_packet_len[7:0]	输入	长度小于该值的任何包都将被视为大小过小。如果包的大小小于该值，则在 rx_axis_tlast 断言有效的周期内，rx_axis_tuser 信号将断言有效。小于 4 字节的包将被丢弃。 ctl_rx_min_packet_len[7:0] 值应 $\geq 64B$ 。	rx_clk_out
stat_rx_framing_err[1:0]	输出	RX 同步报头位成帧错误属于总线，用于指示接收到的同步报头错误数量。仅当 stat_rx_framing_err_valid 设置为 1 时，该总线的值才有效。这些值可随时更新，且旨在作为递增值以供同步报头错误计数器使用。	rx_clk_out
stat_rx_framing_err_valid	输出	stat_rx_framing_err 的有效性指示符。当此信号采样为 1 时，stat_rx_framing_err 上的值有效。	rx_clk_out
stat_rx_local_fault	输出	当 stat_rx_internal_local_fault 或 stat_rx_received_local_fault 断言有效时，此输出为高电平 (High)。此输出对电平敏感。	rx_clk_out
stat_rx_status	输出	此信号表示链路的当前状态。	rx_clk_out
stat_rx_block_lock	输出	块锁定状态。值为 1 表示已根据第 49.2.14 条和 MDIO 寄存器 3.32.0 的定义达成块锁定。此输出对电平敏感。	rx_clk_out
stat_rx_remote_fault	输出	远端故障指示状态。如果该位采样为 1，则表示检测到远端故障状况。如果该位采样为 0，则表示不存在远端故障状况。此输出对电平敏感。	rx_clk_out
stat_rx_bad_fcs[1:0]	输出	无效 FCS 指示符。该总线上的值用于指示在单一周期内，接收到的包带有无效 FCS，但不带有含 stomp 标记的 FCS。含 stomp 标记的 FCS 定义为期望的有效 FCS 的按位反转。此输出将脉冲 1 个时钟周期，以指示错误状况。请注意，脉冲可在不间断的连续周期内发生。	rx_clk_out
stat_rx_stomped_fcs[1:0]	输出	含 stomp 标记的 FCS 指示符。该总线上的值用于指示接收到的包带有含 stomp 标记的 FCS。含 stomp 标记的 FCS 定义为期望的有效 FCS 的按位反转。此输出将脉冲 1 个时钟周期，以指示含 stomp 标记。请注意，脉冲可在连续周期内发生。	rx_clk_out
stat_rx_truncated	输出	包截位指示符。值为 1 表示当前进行中的包因长度超过 ctl_rx_max_packet_len[14:0] 而被截位。此输出将脉冲 1 个时钟周期，以指示截位状况。请注意，脉冲可在连续周期内发生。	rx_clk_out
stat_rx_internal_local_fault	输出	当由于以下任一操作导致生成内部本地故障时，此信号转为高电平 (High)：测试模式生成或误码率过高。请注意，只要故障状况仍然存在，此信号就会保持高电平 (High)。	rx_clk_out
stat_rx_received_local_fault	输出	当接收到来自链路伙伴的本地故障字数足以触发 IEEE 故障状态机所指定的故障条件时，此信号转为高电平 (High)。只要故障状况仍然存在，此信号就会保持高电平 (High)。	rx_clk_out
stat_rx_hi_ber	输出	误码率 (BER) 过高指示符。设置为 1 时，BER 过高（根据 IEEE 802.3 标准的定义）。对应于第 49.2.14 条中所定义的 MDIO 寄存器位 3.32.1。此输出对电平敏感。	rx_clk_out
ctl_rx_custom_preamble_enable	输入	当此信号断言有效时，会导致 AXI4-Stream 上存在的数据包边带在线路上显示为前导码。	rx_clk_out

## 杂项状态信号和控制信号

下表显示了杂项 I/O 状态信号和控制信号。

表 15：杂项状态端口和控制端口

名称	I/O	描述	时钟域
dclk	输入	动态重配置端口 (DRP) 时钟输入。通过在 Vivado® 集成设计环境 (IDE) 的“GT Selection and Configuration”选项卡中的“GT DRP Clock”字段中提供值，设置必需的频率。这必须是自由运行的输入时钟。	请参阅 <a href="#">时钟设置</a> 。
stat_rx_valid_ctrl_code	输出	表示已接收到含有效控制代码的 PCS 块。	rx_clk_out
stat_rx_got_signal_os	输出	Signal OS 指示。如果该位采样为 1，则表示接收到“Signal OS”字。请注意，以太网网络中不应接收到 Signal OS。	rx_clk_out
ctl_rx_process_lfi	输入	此输入设置为 1 时，RX 核会预测从收发器传入的 LF 控制代码，并对其进行处理。此信号设置为 0 时，RX 核会忽略从收发器传入的 LF 控制代码。	rx_clk_out
ctl_rx_test_pattern	输入	针对 RX 核启用测试模式检查。值为 1 即表示根据第 49 条中的定义启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。检查扰码空闲模式。	rx_clk_out
ctl_tx_test_pattern	输入	针对 TX 核启用扰码空闲测试模式生成。值为 1 即表示根据第 49 条中的定义启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.7。	tx_clk_out
stat_rx_test_pattern_mismatch	输出	测试模式不匹配数递增值。任一周期内的非 0 值均表示 RX 核中在测试模式下发生的不匹配次数。仅当 ctl_rx_test_pattern 设置为 1 时，此输出才有效。此输出可用于根据第 45 条的定义生成 MDIO 寄存器。此输出将脉冲 1 个时钟周期。  <b>注释：</b> 当启用“802cm Preemption”或者核类型为 MAC +PCS/PMA 32 位类型时，将存于此信号。	rx_clk_out
ctl_rx_data_pattern_select	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。	rx_clk_out
ctl_rx_test_pattern_enable	输入	针对 RX 核启用测试模式。值为 1 表示启用测试模式。 对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。具有第二优先级。	rx_clk_out
ctl_tx_data_pattern_select	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。	tx_clk_out
ctl_tx_test_pattern_enable	输入	针对 TX 核启用测试模式生成。值为 1 表示启用测试模式。 对应于第 45 条中所定义的 MDIO 寄存器位 3.42.3。具有第二优先级。	tx_clk_out
ctl_tx_test_pattern_seed_a[57:0]	输入	对应于第 45 条中所定义的 MDIO 寄存器 3.34 到 3.37。	tx_clk_out
ctl_tx_test_pattern_seed_b[57:0]	输入	对应于第 45 条中所定义的 MDIO 寄存器 3.38 到 3.41。	tx_clk_out
ctl_tx_test_pattern_select	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.1。	tx_clk_out

## 统计数据接口端口

下表显示了统计数据接口 I/O 端口。

表 16：统计数据接口 - RX 路径

名称	I/O	描述	时钟域
stat_rx_total_bytes[3:0]	输出	此信号为递增值，对应已收到的字节总数。	rx_clk_out

表 16：统计数据接口 - RX 路径 (续)

名称	I/O	描述	时钟域
stat_rx_total_packets[1:0]	输出	此信号为递增值，对应已收到的数据包总数。	rx_clk_out
stat_rx_total_good_bytes[13:0]	输出	此信号为递增值，对应已收到的有效字节总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。	rx_clk_out
stat_rx_total_good_packets	输出	此信号为递增值，对应已收到的有效数据包总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。	rx_clk_out
stat_rx_packet_bad_fcs	输出	此信号为递增值，对应含帧校验序列 (FCS) 错误的数据包（含 64 到 <code>ctl_rx_max_packet_len</code> 个字节）数。	rx_clk_out
stat_rx_packet_64_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 64 字节）数。	rx_clk_out
stat_rx_packet_65_127_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 65 到 127 字节）数。	rx_clk_out
stat_rx_packet_128_255_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 128 到 255 字节）数。	rx_clk_out
stat_rx_packet_256_511_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 256 到 511 字节）数。	rx_clk_out
stat_rx_packet_512_1023_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 512 到 1,023 字节）数。	rx_clk_out
stat_rx_packet_1024_1518_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 1,024 到 1,518 字节）数。	rx_clk_out
stat_rx_packet_1519_1522_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 1519 到 1522 字节）数。	rx_clk_out
stat_rx_packet_1523_1548_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 1,523 到 1,548 字节）数。	rx_clk_out
stat_rx_packet_1549_2047_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 1,549 到 2,047 字节）数。	rx_clk_out
stat_rx_packet_2048_4095_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 2,048 到 4,095 字节）数。	rx_clk_out
stat_rx_packet_4096_8191_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 4,096 到 8,191 字节）数。	rx_clk_out
stat_rx_packet_8192_9215_bytes	输出	此信号为递增值，对应已收到的有效和无效数据包（含 8,192 到 9,215 字节）数。	rx_clk_out
stat_rx_packet_small	输出	此信号为递增值，对应所含数据小于 64 字节的所有包数。小于 4 字节的包将被丢弃。	rx_clk_out
stat_rx_packet_large	输出	此信号为递增值，对应所含数据超过 9,215 字节的所有包数。	rx_clk_out
stat_rx_unicast	输出	此信号为递增值，对应有效单播数据包数。	rx_clk_out
stat_rx_multicast	输出	此信号为递增值，对应有效多播数据包数。	rx_clk_out
stat_rx_broadcast	输出	此信号为递增值，对应有效广播数据包数。	rx_clk_out
stat_rx_oversize	输出	此信号为递增值，对应含有效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。	rx_clk_out
stat_rx_toolong	输出	此信号为递增值，对应含有效和无效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。	rx_clk_out
stat_rx_undersize	输出	此信号为递增值，对应含有效 FCS 且长度小于 <code>ctl_rx_min_packet_len</code> 的数据包数。	rx_clk_out
stat_rx_fragment	输出	此信号为递增值，对应含无效 FCS 且长度小于 <code>ctl_rx_min_packet_len</code> 的数据包数。	rx_clk_out
stat_rx_vlan	输出	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。	rx_clk_out

表 16：统计数据接口 - RX 路径 (续)

名称	I/O	描述	时钟域
stat_rx_inrangeerr	输出	此信号为递增值，对应含“Length”（长度）字段错误但具有有效 FCS 的数据包数。	rx_clk_out
stat_rx_jabber	输出	此信号为递增值，对应含无效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。	rx_clk_out
stat_rx_pause	输出	此信号为递增值，对应含有效 FCS 的 802.3x MAC 暂停包数。	rx_clk_out
stat_rx_user_pause	输出	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。	rx_clk_out
stat_rx_bad_code	输出	此信号为递增值，对应 64B/66B 编码违例数。此信号用于指示 RX PCS 接收状态机处于 IEEE 802.3 标准所指定的 RX_E 状态。此输出可用于根据第 45 条的定义生成 MDIO 寄存器。	rx_clk_out
stat_rx_bad_sfd	输出	对应无效 SFD 的递增值。此信号用于指示接收到以太网数据包之前是否已接收到有效的 SFD。值为 1 表示接收到的 SFD 无效。  <b>注释：</b> 检测到无效的 SFD 时， <code>stat_rx_bad_sfd</code> 信号即断言有效，与 <code>ctl_rx_check_sfd</code> 信号的设置无关。	rx_clk_out
stat_rx_bad_preamble	输出	对应无效前导码的递增值。此信号用于指示接收到以太网数据包之前是否已接收到有效的前导码。值为 1 表示接收到的前导码无效。  <b>注释：</b> 检测到无效的前导码时， <code>stat_rx_bad_preamble</code> 信号即断言有效，与 <code>ctl_rx_check_preamble</code> 信号的设置无关。	rx_clk_out

表 17：统计数据接口 - TX 路径

名称	I/O	描述	时钟域
stat_tx_total_bytes[3:0]	输出	此信号为递增值，对应已发射的字节总数。 指定 32 位 AXI4-Stream 选项时， <code>stat_tx_total_bytes</code> 的信号宽度将为 [2:0]。	tx_clk_out
stat_tx_total_packets	输出	此信号为递增值，对应已发射的数据包总数。	tx_clk_out
stat_tx_total_good_bytes[13:0]	输出	此信号为递增值，对应已发射的有效字节总数。仅当完全发射不含错误的数据包时，该值才为非 0 值。	tx_clk_out
stat_tx_total_good_packets	输出	此信号为递增值，对应已发射的有效数据包总数。	tx_clk_out
stat_tx_bad_fcs	输出	此信号为递增值，对应具有 FCS 错误的数据包（大于 64 字节）数。	tx_clk_out
stat_tx_packet_64_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 64 字节）数。	tx_clk_out
stat_tx_packet_65_127_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 65 到 127 字节）数。	tx_clk_out
stat_tx_packet_128_255_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 128 到 255 字节）数。	tx_clk_out
stat_tx_packet_256_511_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 256 到 511 字节）数。	tx_clk_out
stat_tx_packet_512_1023_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 512 到 1,023 字节）数。	tx_clk_out

表 17：统计数据接口 - TX 路径 (续)

名称	I/O	描述	时钟域
stat_tx_packet_1024_1518_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 1,024 到 1,518 字节）数。	tx_clk_out
stat_tx_packet_1519_1522_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 1,519 到 1,522 字节）数。	tx_clk_out
stat_tx_packet_1523_1548_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 1,523 到 1,548 字节）数。	tx_clk_out
stat_tx_packet_1549_2047_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 1,549 到 2,047 字节）数。	tx_clk_out
stat_tx_packet_2048_4095_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 2,048 到 4,095 字节）数。	tx_clk_out
stat_tx_packet_4096_8191_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 4,096 到 8,191 字节）数。	tx_clk_out
stat_tx_packet_8192_9215_bytes	输出	此信号为递增值，对应已发射的有效和无效数据包（含 8,192 到 9,215 字节）数。	tx_clk_out
stat_tx_packet_small	输出	此信号为递增值，对应所含数据小于 64 字节的所有包数。	tx_clk_out
stat_tx_packet_large	输出	此信号为递增值，对应所含数据超过 9,215 字节的所有包数。	tx_clk_out
stat_tx_unicast	输出	此信号为递增值，对应有效单播数据包数。	tx_clk_out
stat_tx_multicast	输出	此信号为递增值，对应有效多播数据包数。	tx_clk_out
stat_tx_broadcast	输出	此信号为递增值，对应有效广播数据包数。	tx_clk_out
stat_tx_vlan	输出	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。	tx_clk_out
stat_tx_pause	输出	此信号为递增值，对应含有效 FCS 的 802.3x MAC 暂停包数。	tx_clk_out
stat_tx_user_pause	输出	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。	tx_clk_out
stat_tx_frame_error	输出	此信号为递增值，对应具有如下特征的数据包数：此类数据包已设置 tx_axis_tuser 以表示包结束 (EOP) 异常中止或者帧异常中止（方法是断言 tvalid 无效，但不含 tlast）。	tx_clk_out
stat_tx_bad_parity	输出	如果在任意时钟周期内，TX 奇偶校验检查逻辑计算所得的用户生成的奇偶校验结果不正确，那么此信号将递增。	tx_clk_out

## 暂停接口

下表显示了暂停接口 I/O 端口。

表 18：暂停接口 - 控制端口

名称	I/O	描述	时钟域
ctl_rx_pause_enable[8:0]	输入	RX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。请注意，此信号仅影响 RX 用户接口，不影响暂停处理逻辑。	rx_clk_out
ctl_tx_pause_enable[8:0]	输入	TX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。此信号用于对暂停包的发射进行门控。	tx_clk_out

表 19：暂停接口 - RX 路径

名称	I/O	描述	时钟域
ctl_rx_enable_gcp	输入	值为 1 即表示启用全局控制包处理。	rx_clk_out
ctl_rx_check_mcast_gcp	输入	值为 1 即表示启用全局控制多播目标地址处理。	rx_clk_out
ctl_rx_check_unicast_gcp	输入	值为 1 即表示启用全局控制单播目标地址处理。	rx_clk_out
ctl_rx_pause_da_unicast[47:0]	输入	暂停处理的单播目标地址。	rx_clk_out
ctl_rx_check_sa_gcp	输入	值为 1 即表示启用全局控制源地址处理。	rx_clk_out
ctl_rx_pause_sa[47:0]	输入	暂停处理的源地址。	rx_clk_out
ctl_rx_check_etype_gcp	输入	值为 1 即表示启用全局控制以太类型 (Ethertype) 处理。	rx_clk_out
ctl_rx_check_opcode_gcp	输入	值为 1 即表示启用全局控制操作代码 (opcode) 处理。	rx_clk_out
ctl_rx_opcode_min_gcp[15:0]	输入	最小全局控制操作代码值。	rx_clk_out
ctl_rx_opcode_max_gcp[15:0]	输入	最大全局控制操作代码值。	rx_clk_out
ctl_rx_etype_gcp[15:0]	输入	全局控制处理的以太类型 (Ethertype) 字段。	rx_clk_out
ctl_rx_enable_pcp	输入	值为 1 即表示启用优先控制包处理。	rx_clk_out
ctl_rx_check_mcast_pcp	输入	值为 1 即表示启用优先控制多播目标地址处理。	rx_clk_out
ctl_rx_check_unicast_pcp	输入	值为 1 即表示启用优先控制单播目标地址处理。	rx_clk_out
ctl_rx_pause_da_mcast[47:0]	输入	暂停处理的多播目标地址。	rx_clk_out
ctl_rx_check_sa_pcp	输入	值为 1 即表示启用优先控制源地址处理。	rx_clk_out
ctl_rx_check_etype_pcp	输入	值为 1 即表示启用优先控制以太类型 (Ethertype) 处理。	rx_clk_out
ctl_rx_etype_pcp[15:0]	输入	优先控制处理的以太类型 (Ethertype) 字段。	rx_clk_out
ctl_rx_check_opcode_pcp	输入	值为 1 即表示启用优先控制操作代码 (opcode) 处理。	rx_clk_out
ctl_rx_opcode_min_pcp[15:0]	输入	最小优先级控制操作代码值。	rx_clk_out
ctl_rx_opcode_max_pcp[15:0]	输入	最大优先级控制操作代码值。	rx_clk_out
ctl_rx_enable_gpp	输入	值为 1 即表示启用全局暂停包处理。	rx_clk_out
ctl_rx_check_mcast_gpp	输入	值为 1 即表示启用全局暂停多播目标地址处理。	rx_clk_out
ctl_rx_check_unicast_gpp	输入	值为 1 即表示启用全局暂停单播目标地址处理。	rx_clk_out
ctl_rx_check_sa_gpp	输入	值为 1 即表示启用全局暂停源地址处理。	rx_clk_out
ctl_rx_check_etype_gpp	输入	值为 1 即表示启用全局暂停以太类型 (Ethertype) 处理。	rx_clk_out
ctl_rx_etype_gpp[15:0]	输入	全局暂停处理的以太类型 (Ethertype) 字段。	rx_clk_out
ctl_rx_check_opcode_gpp	输入	值为 1 即表示启用全局暂停操作代码 (opcode) 处理。	rx_clk_out
ctl_rx_opcode_gpp[15:0]	输入	全局暂停操作代码值。	rx_clk_out
ctl_rx_enable_ppp	输入	值为 1 即表示启用优先暂停包处理。	rx_clk_out
ctl_rx_check_mcast_ppp	输入	值为 1 即表示启用优先暂停多播目标地址处理。	rx_clk_out
ctl_rx_check_unicast_ppp	输入	值为 1 即表示启用优先暂停单播目标地址处理。	rx_clk_out
ctl_rx_check_sa_ppp	输入	值为 1 即表示启用优先暂停源地址处理。	rx_clk_out
ctl_rx_check_etype_ppp	输入	值为 1 即表示启用优先暂停以太类型 (Ethertype) 处理。	rx_clk_out
ctl_rx_etype_ppp[15:0]	输入	优先暂停处理的以太类型 (Ethertype) 字段。	rx_clk_out

表 19：暂停接口 - RX 路径 (续)

名称	I/O	描述	时钟域
ctl_rx_check_opcode_ppp	输入	值为 1 即表示启用优先暂停操作代码 (opcode) 处理。	rx_clk_out
ctl_rx_opcode_ppp[15:0]	输入	优先暂停操作代码值。	rx_clk_out
stat_rx_pause_req[8:0]	输出	暂停请求信号。当 RX 收到有效的暂停帧时，它会将该总线的对应位设置为 1，其值保持为 1 不变，直至完成暂停包的处理为止。	rx_clk_out
ctl_rx_pause_ack[8:0]	输入	暂停确认信号。该总线用于确认收到来自用户逻辑的暂停帧。	rx_clk_out
ctl_rx_check_ack	输入	等待确认。如果此输入设置为 1，那么该核会使用 ctl_rx_pause_ack[8:0] 总线进行暂停处理。如果此输入设置为 0，则不使用 ctl_rx_pause_ack[8:0]。	rx_clk_out
ctl_rx_forward_control	输入	值为 1 表示该核将转发控制包。值为 0 会导致核删除控制包。	rx_clk_out
stat_rx_pause_valid[8:0]	输出	用于指示已收到暂停包，并且 stat_rx_pause_quanta[8:0][15:0] 总线上的关联暂停量有效，且必须将其用于暂停处理。如果收到的是 802.3x MAC 暂停包，则 bit[8] 设置为 1。	rx_clk_out
stat_rx_pause_quanta[8:0][15:0]	输出	这 9 个总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x MAC 暂停包，那么此暂停量将置于值 [8] 内。	rx_clk_out

表 20：暂停接口 - TX 路径

名称	I/O	描述	时钟域
ctl_tx_pause_req[8:0]	输入	如果该总线的某个位设置为 1，那么核会使用 ctl_tx_pause_quanta[8:0][15:0] 总线上关联的暂停量值来发射暂停包。如果 bit[8] 设置为 1，那么将发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。	tx_clk_out
ctl_tx_pause_quanta[8:0][15:0]	输入	这 9 个总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。	tx_clk_out
ctl_tx_pause_refresh_timer [8:0][15:0]	输入	这 9 个总线用于设置在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。	tx_clk_out
ctl_tx_da_gpp[47:0]	输入	发射全局暂停包的目标地址。	tx_clk_out
ctl_tx_sa_gpp[47:0]	输入	发射全局暂停包的源地址。	tx_clk_out
ctl_tx_ether_type_gpp[15:0]	输入	发射全局暂停包的以太类型 (EtherType)。	tx_clk_out
ctl_tx_opcode_gpp[15:0]	输入	发射全局暂停包的操作代码。	tx_clk_out
ctl_tx_da_ppp[47:0]	输入	发射优先暂停包的目标地址。	tx_clk_out
ctl_tx_sa_ppp[47:0]	输入	发射优先暂停包的源地址。	tx_clk_out
ctl_tx_ether_type_ppp[15:0]	输入	发射优先暂停包的以太类型 (EtherType)。	tx_clk_out
ctl_tx_opcode_ppp[15:0]	输入	发射优先暂停包的操作代码。	tx_clk_out

表 20：暂停接口 - TX 路径 (续)

名称	I/O	描述	时钟域
ctl_tx_resend_pause	输入	重新发射暂停的暂停包。当此输入采样为 1 时，所有挂起的暂停包都将尽快（即，在当前进行中的包完成后）重新发射，并且重新发射计数器将复位。此输入应脉冲至 1，并且每次保持 1 个周期不变。	tx_clk_out
stat_tx_pause_valid[8:0]	输出	如果该总线的某个位设置为 1，则表示核已发射暂停包。如果 bit[8] 设置为 1，则表示已发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。	tx_clk_out

## 自动协商端口

下表显示了用于自动协商的其它端口。可在 `*wrapper.v` 层级文件中找到这些信号。

表 21：用于自动协商的其它端口

端口名称	I/O	描述	时钟域
an_clk	输入	自动协商电路的输入时钟。在相应版本的 README 自述文件中标示了所需的频率。此信号应为自由运行的时钟。Vivado 集成设计环境 (IDE) 会将此时钟设置为与 GUI 中输入的 GT drp CLK 设置相同的频率。	请参阅 <a href="#">时钟设置</a> 。
an_reset	输入	异步高电平有效复位。	异步
ctl_autoneg_enable	输入	自动协商的使能信号。	an_clk
ctl_autoneg_bypass	输入	输入信号，用于禁用自动协商并绕过自动协商功能。如果断言此输入有效，将关闭自动协商，但 PCS 将连接到输出以允许执行操作。	an_clk
ctl_an_nonce_seed[7:0]	输入	8 位种子，用于初始化随机数 (nonce) 字段多项式生成器。针对自动协商器的每个实例，此输入应始终设置为唯一的非零 (0) 值。	an_clk
ctl_an_pseudo_sel	输入	选择位 49 随机位生成器的多项式生成器。如果此输入为 1，则多项式为 $x^7+x^6+1$ 。如果此输入为 0，则多项式为 $x^7+x^3+1$ 。	an_clk
ctl_restart_negotiation	输入	此输入用于触发重新启动自动协商，与当前电路状态无关。	an_clk
ctl_an_local_fault	输入	此输入信号用于设置发射链路代码字的 remote_fault 位。	an_clk
用于暂停功能广播的信号			
ctl_an_pause	输入	此输入信号用于设置发射链路代码字的 PAUSE 位 (C0)。如果该核不支持暂停，则此信号可能不存在。	an_clk
ctl_an_asmdir	输入	此输入信号用于设置发射链路代码字的 ASMDIR 位 (C1)。如果该核不支持暂停，则此信号可能不存在。	an_clk

表 21：用于自动协商的其它端口 (续)

端口名称	I/O	描述	时钟域
<b>功能信号输入</b>			
ctl_an_ability_1000base_kx	输入	这些输入用于识别发射链路代码字中播发到链路伙伴的以太网协议功能。值为 1 表示接口支持此协议并播发此信息。	an_clk
ctl_an_ability_100gbase_cr10	输入		an_clk
ctl_an_ability_100gbase_cr4	输入		an_clk
ctl_an_ability_100gbase_kp4	输入		an_clk
ctl_an_ability_100gbase_kr4	输入		an_clk
ctl_an_ability_10gbase_kr	输入		an_clk
ctl_an_ability_10gbase_kx4	输入		an_clk
ctl_an_ability_25gbase_cr	输入		an_clk
ctl_an_ability_25gbase_cr1	输入		an_clk
ctl_an_ability_25gbase_kr	输入		an_clk
ctl_an_ability_25gbase_kr1	输入		an_clk
ctl_an_ability_40gbase_cr4	输入		an_clk
ctl_an_ability_40gbase_kr4	输入		an_clk
ctl_an_ability_50gbase_cr2	输入		an_clk
ctl_an_ability_50gbase_kr2	输入		an_clk
ctl_an_ability_2_5gbase_kx	输入	训练模式选择： 0=>25BASE-CR/KR, 50GBASE-CR2/KR2, 100GBASE-CR10/KR10	an_clk
ctl_an_ability_5gbase_kr	输入	1=>100GBASE-CR4/KR4	an_clk
ctl_an_ability_50gbase_krcr	输入	2=>50GBASE-CR/KR, 100GBASE-CR2/KR2, 200GBASE-CR4/KR4	an_clk
ctl_an_ability_100gbase_kr2cr2	输入		an_clk
stat_an_lp_ability_200gbase_kr4cr4	输入		an_clk
ctl_lt_polynomial_select [1:0]	输入		an_clk
ctl_an_fec_request	输入	用于在发射链路代码字中控制第 74 条 FEC 请求位。如果该 IP 核不支持第 74 条 FEC，则此信号可能不存在。	an_clk
ctl_an_fec_ability_override	输入	用于在发射链路代码字中控制第 74 条 FEC 功能位。如果设置此输入，则清除发射链路代码字中的 FEC 功能位。如果该 IP 核不支持第 74 条 FEC，则此信号可能不存在。	an_clk
ctl_an_cl91_fec_ability	输入	该位用于指示第 91 条 FEC 的功能。	an_clk
ctl_an_cl91_fec_request	输入	该位用于请求第 91 条 FEC。	an_clk

表 21：用于自动协商的其它端口 (续)

端口名称	I/O	描述	时钟域
stat_an_link_cntl_1000base_kx[1:0]	输出	来自各种以太网协议的自动协商控制器的链路控制输出。设置如下： 00: DISABLE; PCS 断开连接 01: SCAN_FOR_CARRIER; RX 已连接到 PCS 11: ENABLE; PCS 已连接，可执行任务模式操作 10: 不使用	an_clk
stat_an_link_cntl_100gbase_cr10[1:0]	输出		an_clk
stat_an_link_cntl_100gbase_cr4[1:0]	输出		an_clk
stat_an_link_cntl_100gbase_kp4[1:0]	输出		an_clk
stat_an_link_cntl_100gbase_kr4[1:0]	输出		an_clk
stat_an_link_cntl_10gbase_kr[1:0]	输出		an_clk
stat_an_link_cntl_10gbase_kx4[1:0]	输出		an_clk
stat_an_link_cntl_25gbase_cr[1:0]	输出		an_clk
stat_an_link_cntl_25gbase_cr1[1:0]	输出		an_clk
stat_an_link_cntl_25gbase_kr[1:0]	输出		an_clk
stat_an_link_cntl_25gbase_kr1[1:0]	输出		an_clk
stat_an_link_cntl_40gbase_cr4[1:0]	输出		an_clk
stat_an_link_cntl_40gbase_kr4[1:0]	输出		an_clk
stat_an_link_cntl_50gbase_cr2[1:0]	输出		an_clk
stat_an_link_cntl_50gbase_kr2[1:0]	输出		an_clk
stat_an_link_cntl_2_5gbase_kx[1:0]	输出	来自各种以太网协议的自动协商控制器的链路控制输出。设置如下：	
stat_an_link_cntl_5gbase_kr[1:0]	输出	00: DISABLE; PCS 断开连接 01: SCAN_FOR_CARRIER; RX 已连接到 PCS 11: ENABLE; PCS 已连接，可执行任务模式操作 10: 不使用	
stat_an_link_cntl_50gbase_krcr[1:0]	输出		
stat_an_link_cntl_100gbase_kr2cr2[1:0]	输出		
stat_an_link_cntl_200gbase_kr4cr4[1:0]	输出		
stat_an_fec_enable	输出	用于支持在链路上使用第 74 条 FEC。	an_clk
stat_an_rs_fec_enable	输出	用于支持在链路上使用第 91 条 FEC。	an_clk
stat_an_tx_pause_enable	输出	用于支持在发射路径中进行站对站（全局）暂停包生成，以控制接收路径中的数据流。	an_clk
stat_an_rx_pause_enable	输出	用于支持在接收路径中进行站对站（全局）暂停包解释，以控制来自发射器的数据流。	an_clk
stat_an_autoneg_complete	输出	指示自动协商已完成，并且已接收到来自 PCS 的 RX 链路状态。	an_clk
stat_an_parallel_detection_fault	输出	指示自动协商期间发生并行检测故障。	an_clk
stat_an_lp_ability_1000base_kx	输出	这些信号用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，这些信号全部变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_ability_100gbase_cr10	输出		an_clk
stat_an_lp_ability_100gbase_cr4	输出		an_clk
stat_an_lp_ability_100gbase_kp4	输出		an_clk
stat_an_lp_ability_100gbase_kr4	输出		an_clk
stat_an_lp_ability_10gbase_kr	输出		an_clk
stat_an_lp_ability_10gbase_kx4	输出		an_clk
stat_an_lp_ability_25gbase_cr	输出		an_clk
stat_an_lp_ability_25gbase_kr	输出		an_clk
stat_an_lp_ability_40gbase_cr4	输出		an_clk
stat_an_lp_ability_40gbase_kr4	输出		an_clk

表 21：用于自动协商的其它端口 (续)

端口名称	I/O	描述	时钟域
stat_an_lp_ability_2_5gbase_kx	输出	这些信号用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，这些信号全部变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_ability_5gbase_kr	输出		an_clk
stat_an_lp_ability_50gbase_krcr	输出		an_clk
stat_an_lp_ability_100gbase_kr2cr2	输出		an_clk
stat_an_lp_ability_200gbase_kr4cr4	输出		an_clk
stat_an_lp_ability_25gbase_cr1	输出	用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Extended_Ability_Valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_ability_25gbase_kr1	输出	用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Extended_Ability_Valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_ability_50gbase_cr2	输出	用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Extended_Ability_Valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_ability_50gbase_kr2	输出	用于指示从链路伙伴播发的协议。当 stat_AN_Lp_Extended_Ability_Valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。	an_clk
stat_an_lp_pause	输出	此信号用于指示来自链路伙伴的接收链路代码字的 PAUSE 位 (C0) 的播发值。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，此信号即变为有效。	an_clk
stat_an_lp_asm_dir	输出	此信号用于指示来自链路伙伴的接收链路代码字的 ASMDIR 位 (C1) 的播发值。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，此信号即变为有效。	an_clk
stat_an_lp_fec_ability	输出	此信号用于指示来自链路伙伴的接收链路代码字的 FEC 功能位的播发值。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，此信号即变为有效。	an_clk
stat_an_lp_fec_request	输出	此信号用于指示来自链路伙伴的接收链路代码字的 FEC 请求位的播发值。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，此信号即变为有效。	an_clk
stat_an_lp_autoneg_able	输出	此输出信号用于指示链路伙伴能够执行自动协商。当 stat_AN_Lp_Ability_Valid 输出信号断言有效时，此信号即变为有效。	an_clk
stat_an_lp_ability_valid	输出	此信号用于指示何时所有链路伙伴播发将变为有效。	an_clk
an_loc_np_data[47:0]	输入	本地下一页代码字。这是设置 loc_np 输入的情况下所使用的 48 位代码字。在此数据字段中，位 NP、ACK 和 T 以及位元位置 15、14、12 和 11 将不会随下一页代码字一起传输。这些位是在 Auto-Negotiation Intellectual Property Core (ANIPC) 中生成的。但位元位置 13 处的消息协议位 MP 将进行传输。	an_clk
an_lp_np_data[47:0]	输出	链路伙伴下一页数据信号。此 48 位代码字由 ANIPC 驱动，包含来自远端链路伙伴的 48 位下一页代码字。	an_clk
ctl_an_loc_np	输入	本地下一页 (Local Next Page) 信号指示符。如果此位为 1，则 ANIPC 会将 loc_np_data 输入处的下一页数据字传输到远端链路伙伴。如果此位为 0，则 ANIPC 不会启动下一页协议。如果链路伙伴包含多个待发送的下一页信号，并且 loc_np 位已清零，那么 ANIPC 将传输空消息页面。	an_clk

表 21：用于自动协商的其它端口 (续)

端口名称	I/O	描述	时钟域
ctl_an_lp_np_ack	输入	链路伙伴下一页确认信号。此信号用于告知 ANIPC，本地主机已读取位于 lp_np_data 输出管脚处的下一页数据（来自远端链路伙伴）。当此信号变为高电平时，表示 ANIPC 确认已接收发送到远端链路伙伴的下一页代码字，并启动下一个代码字的传输。在此期间，ANIPC 会移除 lp_np 信号，直至新的下一页信息可用为止。	an_clk
stat_an_loc_np_ack	输出	此信号可指示用于接收 loc_np_data 输入管脚上显示的本地下一页数据的本地主机。当 ANIPC 在 loc_np_data 输入管脚上对下一页数据进行采样时，此信号将脉冲至高电平并保持 1 个时钟周期。当本地主机检测到此信号为高电平时，它必须将位于 loc_np_data 输入管脚处的 48 位下一页代码字替换为要发送的下一个 48 位代码字。如果本地主机没有其它下一页要发送，则它必须将 loc_np 输入清零。	an_clk
stat_an_lp_np	输出	链路伙伴下一页信号。此信号用于指示在 lp_np_data 输出管脚处存在来自远端链路伙伴的有效 48 位下一页代码字。当 lp_np_ack 输入信号被驱动至高电平时，此信号将被驱动至低电平，表示本地主机已读取下一页数据。它将保持低电平，直至在 lp_np_data 输出管脚上下一个代码字变为可用为止，届时 lp_np 输出将被重新驱动至高电平。	an_clk
stat_an_lp_ability_extended_fec[1:0]	输出	此输出用于指示 Schedule 3 中所定义的 FEC 扩展功能。	an_clk
stat_an_lp_extended_ability_valid	输出	当该位为 1 时，表示检测到的扩展功能有效。	an_clk
stat_an_lp_rf	输出	此位用于指示链路伙伴远端故障。	an_clk
stat_an_start_tx_disable	输出	当 ctl_autoneg_enable 为高电平且 ctl_autoneg_bypass 为低电平时，此 stat_an_start_tx_disable 信号会在自动协商的 TX_DISABLE 阶段开始时，在 1 个时钟周期内保持高电平循环。即，当自动协商进入 TX_DISABLE 状态时，此输出将在 1 个时钟周期内保持循环高电平。这样即可有效标示自动协商开始。	an_clk
stat_an_start_an_good_check	输出	当 ctl_autoneg_enable 为高电平且 ctl_autoneg_bypass 为低电平时，此 stat_an_start_an_good_check 信号会在自动协商的 AN_GOOD_CHECK 阶段开始时保持高电平循环（持续 1 个时钟周期）。即，当自动协商进入 AN_GOOD_CHECK 状态时，此输出将在 1 个时钟周期内保持循环高电平。这样即可有效标示链路训练开始。但如果未启用链路训练，即，如果 ctl_lt_training_enable 输入为低电平，那么 stat_an_start_an_good_check 输出可有效标示任务模式操作开始。	an_clk

## 链路训练端口

下表显示了链路训练端口。

表 22：链路训练端口

端口名称	I/O	描述	时钟域
ctl_lt_training_enable	输入	启用链路训练。禁用链路训练时，所有 PCS 通道都以任务模式进行操作。	tx_serdes_clk
ctl_lt_restart_training	输入	此信号用于触发重新启动链路训练，与当前状态无关。	tx_serdes_clk
ctl_lt_rx_trained	输入	此信号断言有效即可指示接收器有限脉冲响应 (FIR) 滤波器系数已全部设置完成，并且训练的接收器部分已完成。	tx_serdes_clk
stat_lt_signal_detect	输出	此信号用于指示相应的链路训练状态机已进入 SEND_DATA 状态，在此状态下可恢复正常 PCS 操作。	tx_serdes_clk

表 22：链路训练端口 (续)

端口名称	I/O	描述	时钟域
stat_lt_training	输出	此信号用于指示何时相应的链路训练状态机正在执行链路训练。	tx_serdes_clk
stat_lt_training_fail	输出	此信号在链路训练期间断言有效，前提是对应的链路训练状态机在训练周期期间检测到超时。	tx_serdes_clk
stat_lt_frame_lock	输出	当链路训练已开始后，将针对每个物理介质相关 (PMD) 通道断言这些信号有效，前提是对应的链路训练接收器能够与链路伙伴建立帧同步。	rx_serdes_clk
stat_lt_preset_from_rx	输出	此信号用于反映从控制块中的链路伙伴接收到的预置控制位的值。	rx_serdes_clk
stat_lt_initialize_from_rx	输出	此信号用于反映从控制块中的链路伙伴接收到的初始化控制位的值。	rx_serdes_clk
stat_lt_k_p1_from_rx0[1:0]	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k+1 系数的更新控制位。	rx_serdes_clk
stat_lt_k0_from_rx0[1:0]	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k0 系数的更新控制位。	rx_serdes_clk
stat_lt_k_m1_from_rx0[1:0]	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k-1 系数的更新控制位。	rx_serdes_clk
stat_lt_stat_p1_from_rx0[1:0]	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k+1 系数的更新状态位。	rx_serdes_clk
stat_lt_stat0_from_rx0[1:0]	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k0 系数的更新状态位。	rx_serdes_clk
stat_lt_stat_m1_from_rx0[1:0]	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k-1 系数的更新状态位。	rx_serdes_clk
ctl_lt_pseudo_seed0[10:0]	输入	此 11 位信号用于生成训练模式生成器的种子。	tx_serdes_clk
ctl_lt_preset_to_tx	输入	此信号用于设置预置位的值，该位用于发射到训练帧的控制块中的链路伙伴。	tx_serdes_clk
ctl_lt_initialize_to_tx	输入	此信号用于设置初始位的值，该位用于发射到训练帧的控制块中的链路伙伴。	tx_serdes_clk
ctl_lt_k_p1_to_tx0[1:0]	输入	此 2 位字段用于设置 k+1 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。	tx_serdes_clk
ctl_lt_k0_to_tx0[1:0]	输入	此 2 位字段用于设置 k0 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。	tx_serdes_clk
ctl_lt_k_m1_to_tx0[1:0]	输入	此 2 位字段用于设置 k-1 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。	tx_serdes_clk
ctl_lt_stat_p1_to_tx0[1:0]	输入	此 2 位字段用于设置 k+1 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。	tx_serdes_clk
ctl_lt_stat0_to_tx0[1:0]	输入	此 2 位字段用于设置 k0 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。	tx_serdes_clk
ctl_lt_stat_m1_to_tx0[1:0]	输入	此 2 位字段用于设置 k-1 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。	tx_serdes_clk
stat_lt_rx_sof[1:1:0]	输出	此输出将在 1 个 RX SerDes 时钟周期内保持高电平，以指示链路训练帧开始。	rx_serdes_clk

## IEEE 802.3 第 74 条 FEC 接口

下表显示了 IEEE 802.3 第 74 条 FEC 控制信号、状态信号和统计数据信号。

表 23：IEEE 802.3 第 74 条 FEC 接口控制信号、状态信号和统计数据信号

信号	I/O	时钟	描述
ctl_fec_tx_enable	输入	tx_serdes_clk	此信号断言有效即可对发射的数据启用第 74 条 FEC 编码
ctl_fec_rx_enable	输入	rx_serdes_clk	此信号断言有效即可对接收的数据启用第 74 条 FEC 解码
ctl_fec_enable_error_to_pcs	输入	rx_serdes_clk	第 74 条 FEC 启用 PCS 错误
stat_fec_inc_correct_count[3:0]	输出	rx_serdes_clk	此信号将每隔约 32 个代码字断言有效，前提是同时断言 ctl_rx_fec_enable 有效，且 FEC 解码器已检测到并已纠正对应帧中的位元错误。
stat_fec_inc_cant_correct_count[3:0]	输出	rx_serdes_clk	此信号将每隔约 32 个代码字断言有效，前提是 ctl_rx_fec_enable 断言有效，并且 FEC 解码器已检测到该位。
stat_fec_lock_error[3:0]	输出	rx_serdes_clk	此信号断言有效的前提是 FEC 解码器经过约 5 ms 后仍无法检测到帧边界。检测到帧边界后，即清除此信号。
stat_fec_rx_lock[3:0]	输出	rx_serdes_clk	此信号断言有效的前提是同时断言 ctl_fec_rx_enable 有效，且 FEC 解码器可检测到帧边界。

## IEEE 802.3 第 108 条 RS-FEC 接口

下表显示了 IEEE 802.3 第 108 条 RS-FEC 控制信号、状态信号和统计数据信号。

表 24：IEEE 802.3 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号

信号	I/O	时钟	描述
ctl_rx_rsfec_enable_correction	输入	rx_serdes_clk	等效于 MDIO 寄存器 1.200.0 0：解码器执行错误检测，但不执行纠错（请参阅 IEEE 802.3by 第 91.5.3.3 条）。 1：解码器还执行纠错。
ctl_rx_rsfec_enable_indication	输入	rx_serdes_clk	等效于 MDIO 寄存器 1.200.1 0：绕过错误指示功能（请参阅 IEEE 802.3by 标准第 91.5.3.3 条）。 1：解码器向 PCS 子层发出错误指示。
ctl_rsfec_enable	输入	rx_serdes_clk	启用 RS-FEC 功能。 10G/25G Ethernet IP 子系统的部分变体可能包含单独的 TX 和 RX 使能信号。
ctl_rsfec_ieee_error_indication_mode	输入	rx_serdes_clk	此信号表示该核遵循 IEEE RS-FEC 规范 1：核遵循 IEEE RS-FEC 规范。 0：如果 ctl_rx_rsfec_enable_correction 和 ctl_rx_rsfec_enable_indication 均设为 0，则绕过 RS 解码器。
ctl_rsfec_consortium_25g	输入	rx_serdes_clk	此信号在 IEEE 第 108 条与 25G 以太网联盟模式之间进行切换 1 = 25G 以太网联盟规范模式。 0 = IEEE 802.3by 模式。 10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 联盟信号。
stat_rx_rsfec_hi_ser	输出	rx_serdes_clk	表示符号错误过多。 如果含 8192 个代码字的窗口内的 RS-FEC 符号错误数量超过阈值 417，则此项设置为 1。 否则，设为 0。

表 24: IEEE 802.3 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号 (续)

信号	I/O	时钟	描述
stat_rx_rsfec_lane_alignment_status	输出	rx_serdes_clk	值为 1 表示 RX RS-FEC 块在来自收发器的数据上已达成对齐。
stat_rx_rsfec_corrected_cw_inc	输出	rx_serdes_clk	此信号为递增值，对应已纠正的错误数。
stat_rx_rsfec_uncorrected_cw_inc	输出	rx_serdes_clk	此信号为递增值，对应未纠正的错误数。
stat_rx_rsfec_err_count0_inc[2:0]	输出	rx_serdes_clk	此信号为递增值，对应已检测到的错误数。
stat_tx_rsfec_lane_alignment_status	输出	tx_serdes_clk	值为 1 表示 TX RS-FEC 块在传入 PCS 数据上已达成对齐。

## 端口描述 - PCS 变体

本节将展示 10G/25G PCS 核端口。这些是在提供仅限 PCS 选项时可用的端口。其中不提供 FCS 功能。PCS 不包含“暂停”和“流量控制”端口。系统接口为 XGMII/25GMII。下表显示了 PCS 变体 I/O 端口。

表 25: PCS 变体 I/O 端口

名称	I/O	时钟域	描述
stat_tx_local_fault	输出	tx_mii_clk	值为 1 表示发射编码器状态机当前处于 TX_INIT 状态。此输出对电平敏感。
ctl_rx_prbs31_test_pattern_enable	输入	rx_clk_out	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.5。具有第一优先级。
ctl_rx_test_pattern_enable	输入	rx_clk_out	针对 RX 核启用测试模式。值为 1 表示启用测试模式。 对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。具有第二优先级。
ctl_rx_data_pattern_select	输入	rx_clk_out	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。
ctl_rx_test_pattern	输入	rx_clk_out	对 RX 核启用测试模式以接收扰码空闲模式。具有第三优先级。
ctl_tx_prbs31_test_pattern_enable	输入	tx_mii_clk	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.4。具有第一优先级。
ctl_tx_test_pattern_enable	输入	tx_mii_clk	针对 TX 核启用测试模式生成。值为 1 表示启用测试模式。 对应于第 45 条中所定义的 MDIO 寄存器位 3.42.3。具有第二优先级。
ctl_tx_test_pattern_select	输入	tx_mii_clk	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.1。
ctl_tx_data_pattern_select	输入	tx_mii_clk	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。
ctl_tx_test_pattern_seed_a[57:0]	输入	tx_mii_clk	对应于第 45 条中所定义的 MDIO 寄存器 3.34 到 3.37。
ctl_tx_test_pattern_seed_b[57:0]	输入	tx_mii_clk	对应于第 45 条中所定义的 MDIO 寄存器 3.38 到 3.41。
ctl_tx_test_pattern	输入	tx_mii_clk	针对 TX 核启用扰码空闲测试模式生成。值为 1 表示启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.7。具有第三优先级。
stat_tx_fifo_error	输出	tx_mii_clk	发射时钟补偿 FIFO 错误指示符。值为 1 表示时钟补偿 FIFO 发生下溢或上溢。 如果在任意时钟周期内此输出采样为 1，那么对应端口必须复位才能恢复正常运行。

表 25：PCS 变体 I/O 端口 (续)

名称	I/O	时钟域	描述
stat_rx_fifo_error	输出	rx_clk_out	接收时钟补偿 FIFO 错误指示符。值为 1 表示时钟补偿 FIFO 发生下溢或上溢。仅当恢复的时钟与局部参考时钟之间的 PPM 差超过 $\pm 200$ ppm 时，才会发生此状况。 如果在任意时钟周期内此输出采样为 1，那么对应端口必须复位才能恢复正常运行。
stat_rx_local_fault	输出	rx_clk_out	值为 1 表示接收解码器状态机处于 RX_INIT 状态。 此输出对电平敏感。
stat_rx_hi_ber	输出	rx_clk_out	误码率 (BER) 过高指示符。设置为 1 时，BER 过高（根据 802.3 的定义）。 对应于第 45 条中所定义的 MDIO 寄存器位 3.32.1。 此输出对电平敏感。
stat_rx_block_lock	输出	rx_clk_out	每个 PCS 通道的块锁定状态。值为 1 表示对应通道已达到第 49 条中定义的块锁定状态。 对应于第 45 条中所定义的 MDIO 寄存器位 3.50.7:0 和 3.51.11:0。 此输出对电平敏感。
stat_rx_error	输出	rx_clk_out	测试模式不匹配数递增值。任一周期内的非 0 值均表示 RX 核中在测试模式下发生的不匹配次数。 仅当 ctl_rx_test_pattern 设置为 1 时，此输出才有效。 此输出可用于根据第 45 条的定义生成 MDIO 寄存器 3.43.15:0。 此输出将脉冲 1 个时钟周期。
stat_rx_valid_ctrl_code	输出	rx_clk_out	表示已接收到含有效控制代码的 PCS 块。
stat_rx_error_valid	输出	rx_clk_out	增量有效指示符。如果在任意时钟周期内此信号为 1，stat_rx_error_valid[0:0] 的值有效。
stat_rx_bad_code	输出	rx_clk_out	此信号为递增值，对应 64B/66B 编码违例数。此信号指示接收到的包含无效块的 64b/66b 字的数量，或者指示是否检测到错误的 64b/66b 块序列。 此输出可用于根据第 45 条的定义生成 MDIO 寄存器 3.33:7:0。
stat_rx_bad_code_valid	输出	rx_clk_out	增量有效指示符。如果在任意时钟周期内此信号为 1，则 stat_rx_bad_code[0:0] 的值有效。
stat_rx_framing_err	输出	rx_clk_out	此信号为增量值，对应检测到的错误同步报头位的数量。该总线的值仅在对应 stat_rx_framing_err_valid 为 1 的周期内有效。
stat_rx_framing_err_valid	输出	rx_clk_out	增量有效指示符。如果在任意时钟周期内此信号为 1，stat_rx_framing_err[0:0] 的值有效。

## 收发器接口端口

下表显示了收发器 I/O 端口。

表 26：收发器 I/O

名称	I/O	时钟域	描述
GT_reset	输入	异步	收发器启动 FSM 的高电平有效复位。请注意，此信号还会启动整个 10G/25G Ethernet IP 核的复位序列。
refclk_n0	输入	请参阅时钟设置。	SerDes 的差分参考时钟输入，负相。
refclk_p0	输入	请参阅时钟设置。	SerDes 的差分参考时钟输入，负相。

表 26：收发器 I/O (续)

名称	I/O	时钟域	描述
rx_serdes_data_n0	输入	请参阅 <a href="#">时钟设置</a> 。	来自数据线的串行数据；差分信号的负相。
rx_serdes_data_p0	输入	请参阅 <a href="#">时钟设置</a> 。	来自数据线的串行数据；差分信号的正相。
tx_serdes_data_n0	输出	请参阅 <a href="#">时钟设置</a> 。	到数据线的串行数据；差分信号的负相。
tx_serdes_data_p0	输出	请参阅 <a href="#">时钟设置</a> 。	到数据线的串行数据；差分信号的正相。
tx_serdes_clkout	输出	请参阅 <a href="#">时钟设置</a> 。	此信号存在时，与 tx_clk_out 相同。

## XGMII/25GMII 接口端口

下表显示了 XGMII/25GMII I/O 端口。

表 27：XGMII/25GMII 接口端口

名称	I/O	时钟域	描述
rx_mii_d[63:0]	输出	rx_mii_clk	接收 XGMII/25GMII 数据总线。
rx_mii_c[7:0]	输出	rx_mii_clk	XGMII/25GMII 接收控制总线。
rx_mii_clk	输入	请参阅 <a href="#">时钟设置</a> 。	接收 XGMII/25GMII 时钟输入。
tx_mii_d[63:0]	输入	tx_mii_clk	发射 XGMII/25GMII 数据总线。
tx_mii_c[7:0]	输入	tx_mii_clk	发射 XGMII/25GMII 控制总线。
rx_clk_out	输出	请参阅 <a href="#">时钟设置</a> 。	这是 RX PCS 统计数据的参考时钟。
tx_clk_out (或 tx_mii_clk)	输出	请参阅 <a href="#">时钟设置</a> 。	此输出用于对 TX MII 总线进行时钟设置。在此信号的上升沿上对数据进行时钟设置。
rx_mii_reset	输入	异步	RX MII 接口的复位输入。
tx_mii_reset	输入	异步	TX MII 接口的复位输入。

## 杂项状态端口和控制端口

下表显示了杂项状态端口和控制端口。

表 28：杂项状态端口和控制端口

名称	I/O	时钟域	描述
dclk	输入	请参阅 <a href="#">时钟设置</a> 。	动态重配置端口 (DRP) 时钟输入。通过在 Vivado® IDE 的“GT Selection and Configuration”(GT 选择和配置) 选项卡中的“GT DRP Clock”(GT DRP 时钟) 字段中提供值，设置必需的频率。这必须是自由运行的输入时钟。

## IEEE 802.3 第 74 条 FEC 接口

下表显示了 IEEE 802.3 第 74 条 FEC 控制信号、状态信号和统计数据信号。

表 29：IEEE 802.3 第 74 条 FEC 接口控制信号、状态信号和统计数据信号

信号	I/O	时钟	描述
ctl_fec_tx_enable	输入	tx_serdes_clk	此信号断言有效即可对发射的数据启用第 74 条 FEC 编码。
ctl_fec_rx_enable	输入	rx_serdes_clk	此信号断言有效即可对接收的数据启用第 74 条 FEC 解码。
ctl_fec_enable_error_to_pcs	输入	rx_serdes_clk	第 74 条 FEC 启用 PCS 错误。
stat_fec_inc_correct_count[3:0]	输出	rx_serdes_clk	此信号将每隔约 32 个代码字断言有效，前提是同时断言 <code>ctl_rx_fec_enable</code> 有效，且 FEC 解码器已检测到并已纠正对应帧中的位元错误。
stat_fec_inc_cant_correct_count[3:0]	输出	rx_serdes_clk	此信号将每隔约 32 个代码字断言有效，前提是 <code>ctl_rx_fec_enable</code> 断言有效，并且 FEC 解码器已检测到该位。
stat_fec_lock_error[3:0]	输出	rx_serdes_clk	此信号断言有效的前提是 FEC 解码器经过约 5 ms 后仍无法检测到帧边界。检测到帧边界后，即清除此信号。
stat_fec_rx_lock[3:0]	输出	rx_serdes_clk	此信号断言有效的前提是同时断言 <code>ctl_fec_rx_enable</code> 有效，且 FEC 解码器可检测到帧边界。

## IEEE 802.3 第 108 条 RS-FEC 接口

下表显示了 IEEE 802.3 第 108 条 RS-FEC 控制信号、状态信号和统计数据信号。

表 30：IEEE 802.3 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号

信号	I/O	时钟	描述
ctl_rx_rsfec_enable_correction	输入	rx_serdes_clk	等效于 MDIO 寄存器 1.200.0 <ul style="list-style-type: none"><li>· 0：解码器执行错误检测，但不执行纠错（请参阅 IEEE 802.3by 第 91.5.3.3 条）。</li><li>· 1：解码器还会执行纠错。</li></ul>
ctl_rx_rsfec_enable_indication	输入	rx_serdes_clk	等效于 MDIO 寄存器 1.200.1 <ul style="list-style-type: none"><li>· 0：绕过错误指示功能（请参阅 IEEE 802.3by 标准第 91.5.3.3 条）。</li><li>· 1：解码器向 PCS 子层发出错误指示</li></ul>
ctl_rsfec_enable	输入	rx_serdes_clk	启用 RS-FEC 功能。 <b>注释：</b> 10G/25G Ethernet IP 子系统的部分变体可能包含单独的 TX 和 RX 使能信号。
ctl_rsfec_ieee_error_indication_mod_e	输入	rx_serdes_clk	此信号表示该核遵循 IEEE RS-FEC 规范。 <ul style="list-style-type: none"><li>· 1：核遵循 IEEE RS-FEC 规范。</li><li>· 0：如果 <code>ctl_rx_rsfec_enable_correction</code> 和 <code>ctl_rx_rsfec_enable_indication</code> 均设为 0，则绕过 RS 解码器。</li></ul>

表 30：IEEE 802.3 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号 (续)

信号	I/O	时钟	描述
ctl_rsfec_consortium_25g	输入	rx_serdes_clk	<p>此信号在 IEEE 第 108 条与 25G 以太网联盟模式之间进行切换</p> <ul style="list-style-type: none"> <li>· 1 = 25G 以太网联盟规范模式；</li> <li>· 0 = IEEE 802.3by 模式</li> </ul> <p><b>注释：</b>10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 联盟信号。</p>
stat_rx_rsfec_hi_ser	输出	rx_serdes_clk	<p>表示符号错误过多。</p> <p>如果含 8192 个代码字的窗口内的 RS-FEC 符号错误数量超过阈值 417，则此项设置为 1。</p> <p>否则，设为 0</p>
stat_rx_rsfec_lane_alignment_status	输出	rx_serdes_clk	值为 1 表示 RX RS-FEC 块在来自收发器的数据上已达成对齐。
stat_rx_rsfec_corrected_cw_inc	输出	rx_serdes_clk	此信号为递增值，对应已纠正的错误数
stat_rx_rsfec_uncorrected_cw_inc	输出	rx_serdes_clk	此信号为递增值，对应未纠正的错误数
stat_rx_rsfec_err_count0_inc[2:0]	输出	rx_serdes_clk	此信号为递增值，对应已检测到的错误数
stat_tx_rsfec_lane_alignment_status	输出	tx_serdes_clk	值为 1 表示 TX RS-FEC 块在传入 PCS 数据上已达成对齐。

## 端口描述 - 10G Ethernet MAC (64 位) 变体

### MII 接口

此接口用于连接到物理层，在其中作为独立器件，或者在 Ethernet MAC 核旁的 FPGA 内实现。下表显示了与该接口关联的端口。

表 31：MII 接口

名称	I/O	时钟域	描述
rx_mac_mii_d[63:0]/rx_mii_d	输入	rx_mii_clk	从 PHY 接收数据
rx_mac_mii_c[7:0]/rx_mii_c	输入	rx_mii_clk	从 PHY 接收控制
rx_mac_mii_clk/rx_mii_clk	输入		接收从 PHY 连接的时钟
rx_mac_mii_reset/rx_mii_reset	输入	rx_mii_clk	复位从 PHY 接收的信号
tx_mac_mii_d[63:0]/tx_mii_d	输出	tx_mii_clk	将数据发射到 PHY
tx_mac_mii_c[7:0]/tx_mii_c	输出	tx_mii_clk	将控制发射到 PHY
tx_mac_mii_clk(tx_mii_clk)	输出		XGMII 输出时钟，发送到外部 PHY
tx_mac_mii_reset(tx_mii_reset)	输出	tx_mii_clk	复位信号，发送到外部 PHY

### AXI4-Stream 接口

AXI4-Stream 接口时钟和复位信号如下表所示。

表 32：AXI4-Stream - 时钟和复位

名称	I/O	时钟域	描述
rx_reset	输入	异步	RX 电路的复位。此信号为高电平有效 (1 = 复位)，并且必须保持高电平，直至 clk 时钟达到稳定状态为止。该核可处理将 rx_reset 输入同步到核中相应的时钟域的操作。
tx_reset	输入	异步	TX 电路的复位。此信号为高电平有效 (1 = 复位)，并且必须保持高电平，直至 clk 时钟达到稳定状态为止。该核可处理将 tx_reset 输入同步到核中相应的时钟域的操作。
clk	输入		10G/25G High Speed Ethernet Subsystem 与用户端逻辑之间的所有信号均同步到此信号的上升沿。

## AXI4-Stream 接口 - TX

下表显示了 AXI4-Stream 发射接口信号。

表 33：AXI4-Stream 发射接口信号

名称	I/O	时钟域	描述
tx_axis_tdata[63:0]	输入	clk	AXI4-Stream 数据
tx_axis_tkeep[7:0]	输入	clk	AXI4-Stream 数据控制。
tx_axis_tlast	输入	clk	AXI4-Stream 信号，指示包结束。
tx_axis_tvalid	输入	clk	AXI4-Stream 数据有效。
tx_axis_tuser	输入	clk	AXI4-Stream 用户边带接口信号。 1 指示包无效 0 指示包有效
tx_axis_tready	输出	clk	AXI4-Stream 确认信号，用于指示开始数据传输
tx_parityin[7:0]	输入	clk	AXI4-Stream 用户生成的奇偶校验。遵循与 tx_axis_tkeep 相同的数据通道映射。

## 数据通道映射 - TX

对于发射数据 tx\_axis\_tdata[63:0]，端口按逻辑分割为通道 0 到通道 7。请参阅下表。

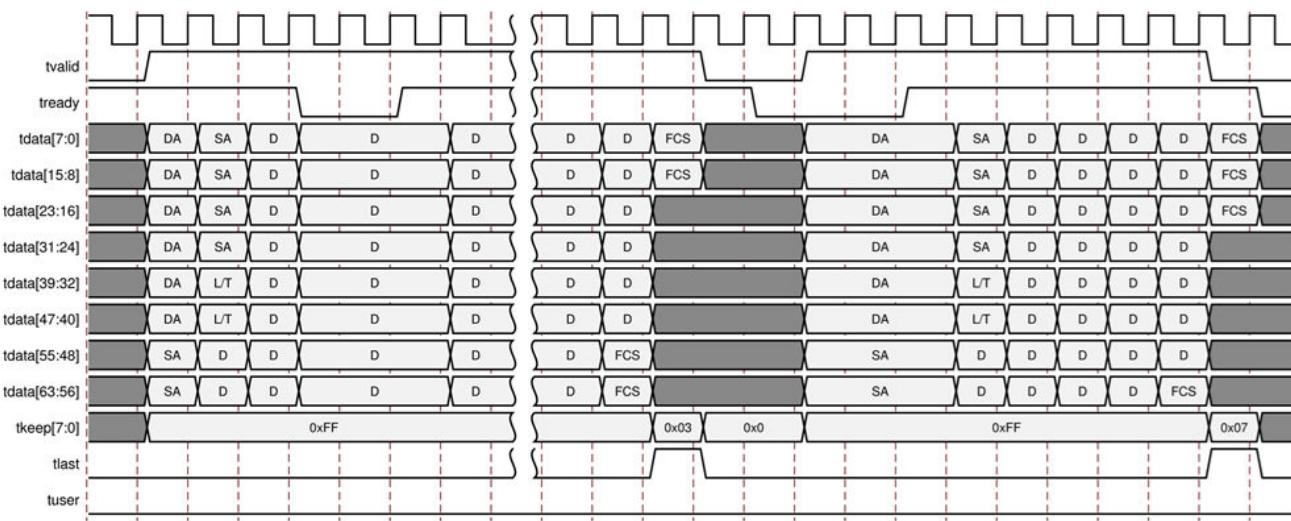
表 34：tx\_axis\_tdata 通道

通道/tx_axis_tkeep	tx_axis_tdata[63:0] 位
0	7:0
1	15:8
2	23:16
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

## 正常发射

正常帧传输的时序如下图所示。当客户端发射帧时，它会断言 `tx_axis_tvalid` 有效并将 `tx_axis_tdata` 和 `tx_axis_tkeep` 中的数据和控制布局在同一个时钟周期内。核接受此数据时（通过 `tx_axis_tready` 断言有效来指示），客户端必须提供下一个周期的数据。如果核未断言 `tx_axis_tready` 有效，则客户端必须保持当前有效数据值，直至其断言有效为止。通过 `tx_axis_tlast` 断言有效并保持 1 个周期来向核指示包结束。正确设置 `tx_axis_tkeep` 的位，以指示最终数据传输中的有效字节数。通过同时断言 `tx_axis_tuser` 有效来指示数据包无效。在 `tx_axis_tlast` 断言无效后，所有数据与控制都被视为无效，直至下一次 `tx_axis_tvalid` 断言有效为止。

图 13：正常帧传输 - 64 位



## 中止发射

客户端接口数据包传输中止称为欠载 (underrun)。如果 AXI 发射客户端接口中的 FIFO 在帧完成之前清空，则可能发生这种情况。

通过以下任一方法即可向核指明发生发射中止：

- 显式错误，由于 `tx_axis_tuser` 断言为高电平无效且 `tx_axis_tlast` 为高电平而导致帧传输中止。请参阅 [AXI4-Stream 接口](#)。
- 隐式欠载，由于 `tx_axis_tvalid` 断言无效且 `tx_axis_tlast` 未断言有效而导致帧传输中止。

## AXI4-Stream 接口 - RX

下表显示了 AXI4-Stream 接收接口信号。

表 35：AXI4-Stream 接收接口信号

名称	I/O	时钟域	描述
<code>rx_axis_tdata[63:0]</code>	输出	clk	AXI4-Stream 数据，连接到上层
<code>rx_axis_tkeep[7:0]</code>	输出	clk	AXI4-Stream 数据控制，连接到上层
<code>rx_axis_tlast</code>	输出	clk	AXI4-Stream 信号，指示包结束
<code>rx_axis_tvalid</code>	输出	clk	AXI4-Stream 数据有效

表 35：AXI4-Stream 接收接口信号（续）

名称	I/O	时钟域	描述
rx_axis_tuser	输出	clk	AXI4-Stream 用户边带接口 1 指示接收到的包无效 0 指示接收到的数据包有效
rx_parityout[7:0]	输出	clk	AXI4-Stream 核生成的奇偶校验。遵循与 rx_axis_tkeep 相同的数据通道映射。

## 数据通道映射 - RX

对于接收数据 `rx_axis_tdata[63:0]`，端口按逻辑划分为通道 0 到通道 7。请参阅下表。

表 36：`rx_axis_tdata` 通道

Lane/ <code>rx_axis_tkeep</code>	<code>rx_axis_tdata[63:0]</code> 位
0	7:0
1	15:8
2	23:16
3	31:24
4	39:32
5	47:40
6	55:48
7	63:56

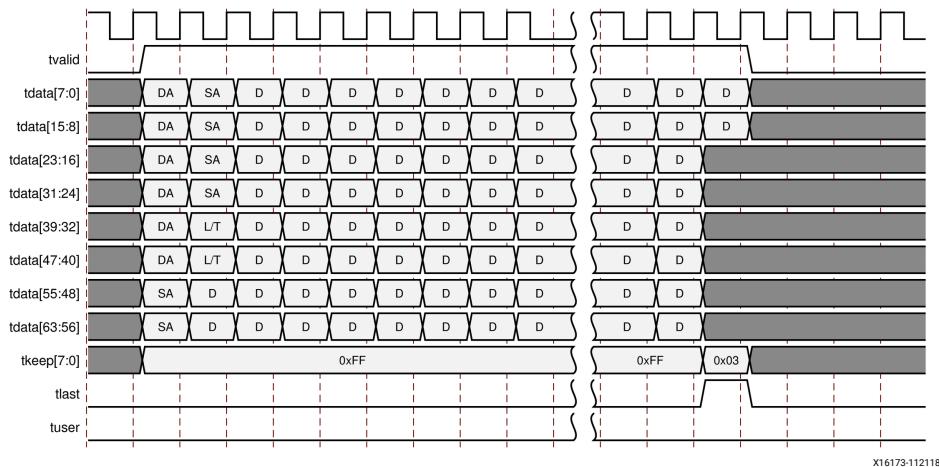
## 正常帧接收

正常入站帧传输的时序如下图所示。客户端必须准备好随时接受数据；由于核中没有缓冲机制，因此接收客户端不允许出现时延。

在帧接收期间，`rx_axis_tvalid` 断言有效，以指示有效的帧数据正通过 `rx_axis_tdata` 传输到客户端。所有 `rx_axis_tkeep` 位均设为 1，表示整个帧中的所有字节始终有效，但在 `rx_axis_tlast` 断言有效时该帧进行最终传输期间除外。在帧数据的最终传输期间，`rx_axis_tkeep` 位使用上述映射来指示帧的最终有效字节。最终传输的有效字节始终来自于 `rx_axis_tdata[7:0]` (`rx_axis_tkeep[0]`)，因为以太网帧数据是连续的并且先接收最低有效字节。

仅在完成所有帧校验后，随传输的最终字节一起，`rx_axis_tlast` 断言有效并且 `rx_axis_tuser` 断言无效。这发生在接收到帧校验序列 (FCS) 字段后。核使 `rx_axis_tuser` 信号保持处于断言无效状态，以指示已成功接收到帧，并且客户端应对帧进行分析。这也表示包结束，以 `rx_axis_tlast` 断言有效并保持 1 个周期来表示。

图 14：正常帧接收 - 64 位



X16173-112118

### 帧接收时出错

下图显示了对应 32 位和 64 位的帧接收不成功的情况（例如，超短帧或含错误 FCS 的帧）。在此情况下，接收到的帧为坏帧，并且在此帧结束时会向客户端断言 `rx_axis_tuser` 信号有效。随后，客户端负责丢弃已为此帧传输的数据。

以下条件会导致 `rx_axis_tlast` 断言有效并且 `rx_axis_tuser = 1`（表示坏帧）：

- 发生 FCS 错误
- 数据包长度小于 64 字节（大小过小或者碎片帧）
- 接收到的帧的长度超过已编程的最大传输单元 (MTU) 大小
- 接收到的任何控制帧的长度都并非恰好为最小帧长度
- XGMII 数据串流包含错误代码

## AXI4-Stream 控制端口和状态端口 - TX

表 37：AXI4-Stream 接口 - TX 路径控制信号和状态信号

名称	I/O	时钟域	描述
ctl_tx_enable	输入	clk	TX 使能（启用）。 当此信号采样为 1 时，用于启用数据传输。 当此信号采样为 0 时，核仅发射空闲 (IDLE) 信号。 仅当接收此输入所发送的数据的接收器已完全同步并且准备好接收数据后，此输入才应设置为 1。（即，链路伙伴上的接收器当前未在发送远端故障状况。） 否则，可能发生数据丢失。如果发射数据包时此信号设置为 0，那么当前数据包发射完成后，核将停止发射任何其它数据包。
ctl_tx_custom_preamble_enable	输入	clk	当此信号断言有效时，会启用使用 <code>tx_preamblein</code> 作为定制前导码，而不是插入标准前导码。
tx_preamblein[55:0]	输入	clk	此信号为定制前导码，它应为独立输入端口，而不是与数据保持一致。包起始期间，此信号应有效。

表 37：AXI4-Stream 接口 - TX 路径控制信号和状态信号（续）

名称	I/O	时钟域	描述
ctl_tx_ipg_value[3:0]	输入	clk	ctl_tx_ipg_value 用于定义在数据包之间插入的目标平均最小包间间隔 (IPG，以字节为单位)。典型值为 12。ctl_tx_ipg_value 还可编程为范围介于 0 到 7 之间的值，但在此情况下，它解释为表示“最小 IPG”，因此仅插入终止 (Terminate) 代码字 IPG；在此情况下从不添加空闲 (Idle)，因此发射随机大小的数据包时，生成的 IPG 平均约为 4 字节。此信号为可选。
ctl_tx_fcs_ins_enable	输入	clk	启用由 TX 核执行 FCS 插入。 如果设为 0，该核不会将 FCS 添加到包中。 如果设为 1，那么该核会计算 FCS 并将 FCS 添加到包中。 在不同数据包之间不得动态更改此输入
ctl_tx_send_lfi	输入	clk	发射本地故障指示 (LFI) 代码字。优先于远端故障指示 (RFI)。
ctl_tx_send_rfi	输入	clk	发射远端故障指示 (RFI) 代码字。 如果采样 1，那么 TX 路径仅发射 RFI 代码字。 此输入应设置为 1，直至 RX 路径已完全同步并且已准备好接受来自链路伙伴的数据为止。
ctl_tx_send_idle	输入	clk	发射 IDLE 代码字。 如果采样 1，那么 TX 路径仅发射 IDLE 代码字。 当伙伴发送 RFI 代码字时，此输入应设置为 1。
ctl_tx_ignore_fcs	输入	clk	启用由 TX 核在 AXI4-Stream 接口上执行 FCS 纠错。 仅当 ctl_tx_fcs_ins_enable 为低电平时，此输入才有效。 如果设为 0，并且发射的数据包中含有无效 FCS，则不会将其作为有效包进行分箱。 如果设为 1，则将把含无效 FCS 的包作为有效包进行分箱。 在 stat_tx_bad_fcs 和 stomped_fcs 信号上会标记此错误，并且此数据包将作为已接收的包进行发射。 <b>注释：</b> 报告的统计数据中显示不含 FCS 错误。
ctl_tx_parity_err_response	输入	clk	TX 核发出奇偶校验错误响应。如果该位设为 0，那么检测到任何奇偶校验错误时，该核都不会执行任何操作。如果该位设为 1，那么该核会给传出的 FCS 添加 stomp 标记（即按位反转）并断言 stat_tx_bad_fcs 有效。

## AXI4-Stream 控制端口和状态端口 - RX

表 38：AXI4-Stream 接口 - RX 路径控制信号和状态信号

名称	I/O	时钟域	描述
ctl_rx_enable	输入	clk	RX 使能（启用）。在正常操作期间，此输入必须设置为 1。当此输入设置为 0 时，RX 完成接收当前包（如果有）之后，就会停止接收包并阻止 PCS 继续对传入数据进行解码。在此模式下不报告任何统计数据，并且 AXI4-Stream 接口处于空闲状态。
ctl_rx_custom_preamble_enable	输入	clk	当此信号断言有效时，会导致 AXI4-Stream 上存在的数据包边带在线路上显示为前导码。
rx_preambleout[55:0]	输出	clk	这是前导码，当前版本中为独立输出，而非与数据保持一致。

表 38：AXI4-Stream 接口 - RX 路径控制信号和状态信号 (续)

名称	I/O	时钟域	描述
ctl_rx_delete_fcs	输入	clk	启用由 RX 核执行 FCS 移除。 如果设置为 0，则该核不会移除传入包的 FCS。 如果设置为 1，则该核将会删除接收包的 FCS。 针对长度小于 8 字节的包，不删除 FCS。 仅当对应的复位输入断言有效时，才应更改此输入。
ctl_rx_ignore_fcs	输入	clk	启用由 RX 核在 AXI4-Stream 接口上执行 FCS 纠错。 如果设置为 0，则接收到的含 FCS 错误的包将标示为错误帧（当 rx_axis_tlast=1 时，rx_axis_tuser=1） 如果设置为 1，则该核不会在 AXI4-Stream 接口处标记 FCS 错误。 报告的统计数据中显示此数据包有效。但 stat_rx_bad_fcs 信号则会报告错误。
ctl_rx_max_packet_len[14:0]	输入	clk	长度超过该值的任何包都将被视为大小过大。如果包的大小大于该值，则将被截位至该值，并且 rx_axis_tuser 信号随 rx_axis_tlast 信号一起断言有效。 ctl_rx_max_packet_len[14] 保留并且必须设置为 0。
ctl_rx_min_packet_len[7:0]	输入	clk	长度小于该值的任何包都将被视为大小过小。如果包的大小小于该值，则 rx_axis_tuser 信号随 rx_axis_tlast 信号一起断言有效。小于 4 字节的包将被丢弃。
ctl_rx_check_sfd	输入	clk	此输入断言有效时，会使 MAC 检查接收到的帧的帧起始定界符 (SFD)。
ctl_rx_check_preamble	输入	clk	此输入断言有效时，会使 MAC 检查接收到的帧的前导码。
stat_rx_local_fault	输出	clk	当 stat_rx_internal_local_fault 或 stat_rx_received_local_fault 断言有效时，此输出为高电平 (High)。此输出对电平敏感。
stat_rx_remote_fault	输出	clk	远端故障指示状态。如果该位采样为 1，则表示检测到远端故障状况。如果该位采样为 0，则表示不存在远端故障状况。此输出对电平敏感。

## 杂项状态信号和控制信号

下表显示了杂项状态信号和控制信号。

表 39：杂项状态信号和控制信号

名称	I/O	时钟域	描述
ctl_rx_process_lfi	输入	clk	此输入设置为 1 时，RX 核会预测从收发器传入的 LF 控制代码，并对其进行处理。 此信号设置为 0 时，RX 核会忽略从收发器传入的 LF 控制代码。
stat_tx_gmii_fifo_unf <sup>1</sup>	输出	clk	TX FIFO 下溢
stat_tx_gmii_fifo_ovf <sup>1</sup>	输出	clk	TX FIFO 上溢

注释：

- 仅在 10G 仅限 MAC 变体中可用。

## 统计数据接口端口

下表显示了 RX 和 TX 路径各自的统计数据接口端口。

表 40：统计数据接口端口 - RX

名称	I/O	时钟域	描述
stat_rx_bad_code	输出	clk	此信号为递增值，对应 64B/66B 编码违例数。此信号用于指示 RX PCS 接收状态机处于 IEEE 802.3 标准所指定的 RX_E 状态。此输出可用于根据第 45 条的定义生成 MDIO 寄存器。
stat_rx_total_packets[1:0]	输出	clk	此信号为递增值，对应已收到的数据包总数。
stat_rx_total_good_packets	输出	clk	此信号为递增值，对应已收到的有效数据包总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。
stat_rx_total_bytes[3:0]	输出	clk	此信号为递增值，对应已收到的字节总数。
stat_rx_total_good_bytes[13:0]	输出	clk	此信号为递增值，对应已收到的有效字节总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。
stat_rx_packet_small	输出	clk	此信号为递增值，对应所含数据小于 64 字节的所有包数。小于 4 字节的包将被丢弃。
stat_rx_jabber	输出	clk	此信号为递增值，对应含无效 FCS 且长度超过 ctl_rx_max_packet_len 的数据包数。
stat_rx_packet_large	输出	clk	此信号为递增值，对应所含数据超过 9,215 字节的所有包数。
stat_rx_oversize	输出	clk	此信号为递增值，对应含有效 FCS 且长度超过 ctl_rx_max_packet_len 的数据包数。
stat_rx_undersize	输出	clk	此信号为递增值，对应含有效 FCS 且长度小于 ctl_rx_min_packet_len 的数据包数。
stat_rx_toolong	输出	clk	此信号为递增值，对应含有效和无效 FCS 且长度超过 ctl_rx_max_packet_len 的数据包数。
stat_rx_fragment	输出	clk	此信号为递增值，对应含无效 FCS 且长度小于 ctl_rx_min_packet_len 的数据包数。
stat_rx_packet_64_bytes	输出	clk	此信号为递增值，对应已收到的有效和无效数据包（含 64 字节）数。
stat_rx_packet_65_127_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 65 到 127 字节）数。
stat_rx_packet_128_255_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 128 到 255 字节）数。
stat_rx_packet_256_511_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 256 到 511 字节）数。
stat_rx_packet_512_1023_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 512 到 1023 字节）数。
stat_rx_packet_1024_1518_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 1024 到 1518 字节）数。
stat_rx_packet_1519_1522_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 1519 到 1522 字节）数。
stat_rx_packet_1523_1548_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 1523 到 1548 字节）数。
stat_rx_packet_1549_2047_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 1549 到 2047 字节）数。
stat_rx_packet_2048_4095_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 2048 到 4095 字节）数。
stat_rx_packet_4096_8191_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 4096 到 8191 字节）数。
stat_rx_packet_8192_9215_bytes	输出	clk	此信号为递增值，对应已接收的有效和无效数据包（含 8192 到 9215 字节）数。

表 40：统计数据接口端口 - RX (续)

名称	I/O	时钟域	描述
stat_rx_bad_fcs [1:0]	输出	clk	当此信号为正值时，表示错误检测逻辑已在接收到的包中发现接收的 CRC32 值与期望的值不匹配。 当检测到 CRC32 错误时，接收到的包将被标记为包含错误，并随上一次传输（即 rx_axis_tlast 断言有效的周期）期间断言有效的 rx_axis_tuser 一起发送，但 ctl_rx_ignore_fcs 断言有效的情况下除外。对于检测到的每个 CRC32 错误，此信号都将断言有效并保持 1 个时钟周期。
stat_rx_packet_bad_fcs	输出	clk	此信号为递增值，对应含帧校验序列 (FCS) 错误的数据包（含 64 到 ctl_rx_max_packet_len 个字节）数。
stat_rx_stomped_fcs [1:0]	输出	clk	含 stomp 标记的 FCS 指示符。该总线上的值用于指示接收到的数据包带有含 stomp 标记的 FCS。含 stomp 标记的 FCS 定义为期望的有效 FCS 的按位反转。此输出将脉冲 1 个时钟周期，以指示含 stomp 标记。请注意，脉冲可在连续周期内发生。
stat_rx_unicast	输出	clk	此信号为递增值，对应有效单播数据包数
stat_rx_multicast	输出	clk	此信号为递增值，对应有效多播数据包数
stat_rx_broadcast	输出	clk	此信号为递增值，对应有效广播数据包数
stat_rx_vlan	输出	clk	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。
stat_rx_pause	输出	clk	此信号为递增值，对应含有效 FCS 的 802.3x MAC 暂停包数。
stat_rx_user_pause	输出	clk	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。
stat_rx_inrangeerr	输出	clk	此信号为递增值，对应含“Length”（长度）字段错误但具有有效 FCS 的数据包数。
stat_rx_bad_preamble	输出	clk	此信号为递增值，对应接收到的含错误前导码的数据包数。此信号用于指示接收到以太网数据包之前是否已接收到有效的前导码。值为 1 表示接收到的前导码无效。 检测到无效的前导码时，stat_rx_bad_preamble 信号即断言有效，与 ctl_rx_check_preamble 信号的设置无关。
stat_rx_bad_sfd	输出	clk	此信号为递增值，对应已接收到的含错误 SFD 的数据包数。此信号用于指示接收到以太网数据包之前是否已接收到有效的 SFD。值为 1 表示接收到的 SFD 无效。 检测到无效的 SFD 时，stat_rx_bad_sfd 信号即断言有效，与 ctl_rx_check_sfd 信号的设置无关。
stat_rx_got_signal_os	输出	clk	Signal OS 指示。如果该位采样为 1，则表示接收到“Signal OS”字。 <b>注释：</b> 在以太网网络中不应接收到 Signal OS。
stat_rx_truncated	输出	clk	包截位指示符。值为 1 表示当前进行中的包因长度超过 ctl_rx_max_packet_len[14:0] 而被截位。此输出将脉冲 1 个时钟周期，以指示截位状况。脉冲可在连续周期内发生。

表 41：统计数据接口 - TX 路径

名称	I/O	时钟域	描述
stat_tx_total_packets	输出	clk	此信号为递增值，对应已发射的数据包总数。
stat_tx_total_bytes[2:0]	输出	clk	此信号为递增值，对应已发射的字节总数。
stat_tx_total_good_packets	输出	clk	此信号为递增值，对应已发射的有效数据包总数。

表 41：统计数据接口 - TX 路径 (续)

名称	I/O	时钟域	描述
stat_tx_total_good_bytes[13:0]	输出	clk	此信号为递增值，对应已发射的有效字节总数。仅当完全发射不含错误的数据包时，该信号才为非 0 值。
stat_tx_packet_64_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 64 字节）数。
stat_tx_packet_65_127_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 65 到 127 字节）数。
stat_tx_packet_128_255_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 128 到 255 字节）数。
stat_tx_packet_256_511_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 256 到 511 字节）数。
stat_tx_packet_512_1023_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 512 到 1023 字节）数。
stat_tx_packet_1024_1518_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 1024 到 1518 字节）数。
stat_tx_packet_1519_1522_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 1519 到 1522 字节）数。
stat_tx_packet_1523_1548_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 1523 到 1548 字节）数。
stat_tx_packet_1549_2047_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 1549 到 2047 字节）数。
stat_tx_packet_2048_4095_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 2048 到 4095 字节）数。
stat_tx_packet_4096_8191_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 4096 到 8191 字节）数。
stat_tx_packet_8192_9215_bytes	输出	clk	此信号为递增值，对应已发射的有效和无效数据包（含 8192 到 9215 字节）数。
stat_tx_packet_small	输出	clk	此信号为递增值，对应所含数据小于 64 字节的所有包数。
stat_tx_packet_large	输出	clk	此信号为递增值，对应所含数据超过 9,215 字节的所有包数。
stat_tx_unicast	输出	clk	此信号为递增值，对应有效单播数据包数。
stat_tx_multicast	输出	clk	此信号为递增值，对应有效多播数据包数。
stat_tx_broadcast	输出	clk	此信号为递增值，对应有效广播数据包数。
stat_tx_vlan	输出	clk	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。
stat_tx_pause	输出	clk	此信号为递增值，对应含有效 FCS 的 802.3x MAC 暂停包数
stat_tx_user_pause	输出	clk	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。
stat_tx_bad_fcs	输出	clk	此信号为递增值，对应具有 FCS 错误的数据包（大于 64 字节）数。
stat_tx_frame_error	输出	clk	此信号为递增值，对应具有如下特征的数据包数：此类数据包已设置 tx_axis_tuser 以表示包结束 (EOP) 异常中止或者帧异常中止（方法是断言 tvalid 无效，但不含 tllast）。
stat_tx_local_fault	输出	clk	值为 1 表示接收解码器状态机当前处于 TX_INIT 状态。此输出对电平敏感。
stat_tx_bad_parity	输出	clk	如果在任意时钟周期内，TX 奇偶校验检查逻辑计算所得的用户生成的奇偶校验结果不正确，那么此信号将递增。

## 暂停接口

下表显示了暂停接口 I/O 端口。

表 42：暂停接口 - 控制端口

名称	I/O	时钟域	描述
ctl_rx_pause_enable[8:0]	输入	clk	RX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。 此信号仅影响 RX 用户接口，不影响暂停处理逻辑
ctl_tx_pause_enable[8:0]	输入	clk	TX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。此信号用于对暂停包的发射进行门控。

表 43：暂停接口 - TX 路径

名称	I/O	时钟域	描述
ctl_tx_pause_req[8:0]	输入	clk	如果该总线的某个位设置为 1，那么核会使用 ctl_tx_pause_quanta[8:0][15:0] 总线上关联的暂停量值来发射暂停包。如果 bit[8] 设置为 1，那么将发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。
ctl_tx_resend_pause	输入	clk	重新发射暂停的暂停包。当此输入采样为 1 时，所有挂起的暂停包都将尽快（即，在当前进行中的包完成后）重新发射，并且重新发射计数器将复位。此输入应脉冲至 1，并且每次保持 1 个周期不变。
ctl_tx_pause_quanta[8:0][15:0]	输入	clk	这 9 个总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于基于优先级的暂停操作。
ctl_tx_pause_refresh_timer[8:0][15:0]	输入	clk	这 9 个总线用于设置在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。
ctl_tx_da_gpp[47:0]	输入	clk	发射全局暂停包的目标地址。
ctl_tx_sa_gpp[47:0]	输入	clk	发射全局暂停包的源地址。
ctl_tx_ether_type_gpp[15:0]	输入	clk	发射全局暂停包的以太类型 (EtherType)。
ctl_tx_opcode_gpp[15:0]	输入	clk	发射全局暂停包的操作代码。
ctl_tx_da_ppp[47:0]	输入	clk	发射优先暂停包的目标地址。
ctl_tx_sa_ppp[47:0]	输入	clk	发射优先暂停包的源地址。
ctl_tx_ether_type_ppp[15:0]	输入	clk	发射优先暂停包的以太类型 (EtherType)。
ctl_tx_opcode_ppp[15:0]	输入	clk	发射优先暂停包的操作代码。
stat_tx_pause_valid[8:0]	输出	clk	如果该总线的某个位设置为 1，则表示核已发射暂停包。如果 bit[8] 设置为 1，则表示已发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。

表 44：暂停接口 - RX

名称	I/O	时钟域	描述
ctl_rx_pause_ack[8:0]	输入	clk	暂停确认信号。该总线用于确认收到来自用户逻辑的暂停帧。
ctl_rx_check_ack	输入	clk	等待确认。如果此输入设置为 1，那么该核会使用 ctl_rx_pause_ack[8:0] 总线进行暂停处理。如果此输入设置为 0，则不使用 ctl_rx_pause_ack[8:0]。
ctl_rx_enable_gcp	输入	clk	值为 1 即表示启用全局控制包处理。
ctl_rx_check_mcast_gcp	输入	clk	值为 1 即表示启用全局控制多播目标地址处理。
ctl_rx_check_unicast_gcp	输入	clk	值为 1 即表示启用全局控制单播目标地址处理。
ctl_rx_pause_da_unicast[47:0]	输入	clk	暂停处理的单播目标地址。
ctl_rx_check_sa_gcp	输入	clk	值为 1 即表示启用全局控制源地址处理。
ctl_rx_pause_sa[47:0]	输入	clk	暂停处理的源地址。
ctl_rx_check_etype_gcp	输入	clk	值为 1 即表示启用全局控制以太类型 (Ethertype) 处理。
ctl_rx_etype_gcp[15:0]	输入	clk	以太类型 (Ethertype) 字段，用于全局控制处理
ctl_rx_check_opcode_gcp	输入	clk	值为 1 即表示启用全局控制操作代码 (opcode) 处理。
ctl_rx_opcode_min_gcp[15:0]	输入	clk	最小全局控制操作代码值
ctl_rx_opcode_max_gcp[15:0]	输入	clk	最大全局控制操作代码值
ctl_rx_enable_pcp	输入	clk	值为 1 即表示启用优先控制包处理
ctl_rx_check_mcast_pcp	输入	clk	值为 1 即表示启用优先控制多播目标地址处理
ctl_rx_check_unicast_pcp	输入	clk	值为 1 即表示启用优先控制单播目标地址处理
ctl_rx_pause_da_mcast[47:0]	输入	clk	暂停处理的多播目标地址。
ctl_rx_check_sa_pcp	输入	clk	值为 1 即表示启用优先控制源地址处理
ctl_rx_check_etype_pcp	输入	clk	值为 1 即表示启用优先控制以太类型 (Ethertype) 处理
ctl_rx_etype_pcp[15:0]	输入	clk	以太类型 (Ethertype) 字段，用于优先控制处理
ctl_rx_check_opcode_pcp	输入	clk	值为 1 即表示启用优先控制操作代码 (opcode) 处理。
ctl_rx_opcode_min_pcp[15:0]	输入	clk	最小优先控制操作代码值
ctl_rx_opcode_max_pcp[15:0]	输入	clk	最大优先控制操作代码值
ctl_rx_enable_gpp	输入	clk	值为 1 即表示启用全局暂停包处理
ctl_rx_check_mcast_gpp	输入	clk	值为 1 即表示启用全局暂停多播目标地址处理
ctl_rx_check_unicast_gpp	输入	clk	值为 1 即表示启用全局暂停单播目标地址处理
ctl_rx_check_sa_gpp	输入	clk	值为 1 即表示启用全局暂停源地址处理。
ctl_rx_check_etype_gpp	输入	clk	值为 1 即表示启用全局暂停以太类型 (Ethertype) 处理
ctl_rx_etype_gpp[15:0]	输入	clk	以太类型 (Ethertype) 字段，用于全局暂停处理
ctl_rx_check_opcode_gpp	输入	clk	值为 1 即表示启用全局暂停操作代码 (opcode) 处理。
ctl_rx_opcode_gpp[15:0]	输入	clk	全局暂停操作代码值。
ctl_rx_enable_ppp	输入	clk	值为 1 即表示启用优先暂停包处理
ctl_rx_check_mcast_ppp	输入	clk	值为 1 即表示启用优先暂停多播目标地址处理
ctl_rx_check_unicast_ppp	输入	clk	值为 1 即表示启用优先暂停单播目标地址处理
ctl_rx_check_sa_ppp	输入	clk	值为 1 即表示启用优先暂停源地址处理
ctl_rx_check_etype_ppp	输入	clk	值为 1 即表示启用优先暂停以太类型 (Ethertype) 处理
ctl_rx_etype_ppp[15:0]	输入	clk	以太类型 (Ethertype) 字段，用于优先暂停处理

表 44：暂停接口 - RX (续)

名称	I/O	时钟域	描述
ctl_rx_check_opcode_ppp	输入	clk	值为 1 即表示启用优先暂停操作代码 (opcode) 处理
ctl_rx_opcode_ppp[15:0]	输入	clk	优先暂停操作代码值
ctl_rx_forward_control	输入	clk	值为 1 表示该核将转发控制包。 值为 0 会导致核删除控制包。
stat_rx_pause_valid [8:0]	输出	clk	用于指示已收到暂停包，并且 stat_rx_pause_quanta[8:0] [15:0] 总线上的关联暂停量有效，且必须将其用于暂停处理。如果收到的是 802.3x MAC 暂停包，则 bit[8] 设置为 1。
stat_rx_pause_quanta[8:0] [15:0]	输出	clk	这 9 个总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_req [8:0]	输出	clk	暂停请求信号。当 RX 收到有效的暂停帧时，它会将该总线的对应位设置为 1，其值保持为 1 不变，直至完成暂停包的处理为止。

## 寄存器空间

用户可选择将 10G/25G Ethernet Subsystem 配置为搭配 AXI4-Lite 寄存器以便访问配置信号和状态信号。

### AXI4-Lite 端口

下表描述了 AXI 处理器接口的端口列表。

表 45：AXI 端口

信号	I/O	描述
s_axi_aclk	输入	AXI4-Lite 时钟。范围在 10 MHz 和 300 MHz 之间
s_axi_arresetn	输入	异步低电平有效复位
s_axi_awaddr[31:0]	输入	写入地址总线
s_axi_awvalid	输入	写入地址有效
s_axi_awready	输出	写入地址确认
s_axi_wdata[31:0]	输入	写入数据总线
s_axi_wstrb[3:0]	输入	数据总线字节通道的选通信号
s_axi_wvalid	输出	写入数据有效
s_axi_wready	输出	写入数据确认
s_axi_bresp[1:0]	输出	写入传输事务响应
s_axi_bvalid	输出	写入响应有效
s_axi_bready	输入	写入响应确认
s_axi_araddr[31:0]	输入	读取地址总线
s_axi_arvalid	输入	读取地址有效

表 45：AXI 端口 (续)

信号	I/O	描述
s_axi_arready	输出	读取地址确认
s_axi_rdata[31:0]	输出	读取数据输出
s_axi_rresp[1:0]	输出	读取数据响应
s_axi_rvalid	输出	读取数据/响应有效
s_axi_rready	输入	读取数据确认
pm_tick	输入	读取统计数据计数器的顶层信号；需要将 MODE_REG[30]（即， tick_reg_mode_sel）设置为 0。

如需获取有关 AXI4 总线运行的其它信息，请参阅《赛灵思 AXI 存储器映射协议 1.8 版》。

如前所述，可使用顶层信号 pm\_tick（而不是配置寄存器 TICK\_REG）来读取统计数据计数器。在此情况下，配置寄存器 MODE\_REG 的位 30（即， tick\_reg\_mode\_sel）应设置为 0。如果 tick\_reg\_mode\_sel 设置为 1，则使用 tick\_reg 来读取统计数据计数器。

## 配置和状态寄存器映射

### 配置寄存器映射 10G/25G Ethernet Subsystem

配置空间可为软件提供配置各种 IP 核用例的功能。部分配置为可选，分配的寄存器在特定变体中可能不存在，在此情况下，相应的寄存器被视为保留 (RESERVED)。

表 46：配置寄存器映射

十六进制地址	寄存器名称	注释
0x0000	GT_RESET_REG: 0000	
0x0004	RESET_REG: 0004	
0x0008	MODE_REG: 0008	
0x000C	CONFIGURATION_TX_REG1: 000C	
0x0014	CONFIGURATION_RX_REG1: 0014	
0x0018	CONFIGURATION_RX_MTU: 0018	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x001C	CONFIGURATION_VL_LENGTH_REG: 001C	
0x0020	TICK_REG: 0020	
0x0024	CONFIGURATION_REVISION_REG: 0024	
0x0028	CONFIGURATION_TX_TEST_PAT_SEED_A_LSB: 0028	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x002C	CONFIGURATION_TX_TEST_PAT_SEED_A_MSB: 002C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0030	CONFIGURATION_TX_TEST_PAT_SEED_B_LSB: 0030	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0034	CONFIGURATION_TX_TEST_PAT_SEED_B_MSB: 0034	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0038	CONFIGURATION_1588_REG: 0038	仅限 MAC+PCS 变体
0x0040	CONFIGURATION_TX_FLOW_CONTROL_REG1: 0040	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0044	CONFIGURATION_TX_FLOW_CONTROL_REFRESH_REG1: 0044	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0048	CONFIGURATION_TX_FLOW_CONTROL_REFRESH_REG2: 0048	仅适用于 MAC+PCS 变体和仅限 MAC 变体

表 46：配置寄存器映射 (续)

十六进制地址	寄存器名称	注释
0x004C	CONFIGURATION_TX_FLOW_CONTROL_REFRESH_REG3: 004C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0050	CONFIGURATION_TX_FLOW_CONTROL_REFRESH_REG4: 0050	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0054	CONFIGURATION_TX_FLOW_CONTROL_REFRESH_REG5: 0054	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0058	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG1: 0058	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x005C	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG2: 005C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0060	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG3: 0060	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0064	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG4: 0064	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0068	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG5: 0068	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x006C	CONFIGURATION_TX_FLOW_CONTROL_PPP_ETYPE_OP_REG: 006C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0070	CONFIGURATION_TX_FLOW_CONTROL_GPP_ETYPE_OP_REG: 0070	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0074	CONFIGURATION_TX_FLOW_CONTROL_GPP_DA_REG_LSB: 0074	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0078	CONFIGURATION_TX_FLOW_CONTROL_GPP_DA_REG_MSB: 0078	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x007C	CONFIGURATION_TX_FLOW_CONTROL_GPP_SA_REG_LSB: 007C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0080	CONFIGURATION_TX_FLOW_CONTROL_GPP_SA_REG_MSB: 0080	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0084	CONFIGURATION_TX_FLOW_CONTROL_PPP_DA_REG_LSB: 0084	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0088	CONFIGURATION_TX_FLOW_CONTROL_PPP_DA_REG_MSB: 0088	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x008C	CONFIGURATION_TX_FLOW_CONTROL_PPP_SA_REG_LSB: 008C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0090	CONFIGURATION_TX_FLOW_CONTROL_PPP_SA_REG_MSB: 0090	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0094	CONFIGURATION_RX_FLOW_CONTROL_REG1: 0094	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0098	CONFIGURATION_RX_FLOW_CONTROL_REG2: 0098	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x009C	CONFIGURATION_RX_FLOW_CONTROL_PPP_ETYPE_OP_REG: 009C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00A0	CONFIGURATION_RX_FLOW_CONTROL_GPP_ETYPE_OP_REG: 00A0	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00A4	CONFIGURATION_RX_FLOW_CONTROL_GCP_PCP_TYPE_REG: 00A4	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00A8	CONFIGURATION_RX_FLOW_CONTROL_PCP_OP_REG: 00A8	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00AC	CONFIGURATION_RX_FLOW_CONTROL_GCP_OP_REG: 00AC	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00B0	CONFIGURATION_RX_FLOW_CONTROL_DA_REG1_LSB: 00B0	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00B4	CONFIGURATION_RX_FLOW_CONTROL_DA_REG1_MSB: 00B4	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00B8	CONFIGURATION_RX_FLOW_CONTROL_DA_REG2_LSB: 00B8	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00BC	CONFIGURATION_RX_FLOW_CONTROL_DA_REG2_MSB: 00BC	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00C0	CONFIGURATION_RX_FLOW_CONTROL_SA_REG1_LSB: 00C0	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00C4	CONFIGURATION_RX_FLOW_CONTROL_SA_REG1_MSB: 00C4	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x00D0	CONFIGURATION_RSFEC_REG: 00D0	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x00D4	CONFIGURATION_FEC_REG: 00D4	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x00E0	CONFIGURATION_AN_CONTROL_REG1: 00E0	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x00E4	CONFIGURATION_AN_CONTROL_REG2: 00E4	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x00F8	CONFIGURATION_AN_ABILITY: 00F8	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0100	CONFIGURATION_LT_CONTROL_REG1: 0100	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0104	CONFIGURATION_LT_TRAINED_REG: 0104	仅适用于 MAC+PCS 变体和仅限 PCS 变体

表 46：配置寄存器映射 (续)

十六进制地址	寄存器名称	注释
0x0108	CONFIGURATION_LT_PRESET_REG: 0108	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x010C	CONFIGURATION_LT_INIT_REG: 010C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0110	CONFIGURATION_LT_SEED_REG0: 0110	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0130	CONFIGURATION_LT_COEFFICIENT_REG0: 0130	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0134	USER_REG_0: 0134	
0x0138	SWITCH_CORE_SPEED_REG: 0138	
0x013C	CONFIGURATION_1588_32BIT_REG: 0x013C	仅适用于 MAC+PCS/PMA 32 位变体 1588 变体
0x0140	TX_CONFIGURATION_1588_REG: 0x0140	仅适用于 MAC+PCS/PMA 32 位变体 1588 变体
0x0144	RX_CONFIGURATION_1588_REG: 0x0144	仅适用于 MAC+PCS/PMA 32 位变体 1588 变体
0x019C	CONFIGURATION_TSN_REG: 0x019C	仅当启用“Preemption”（抢占）功能时才可用。
0x014C	VERSAL_CHANNEL_NUM_REG: 0x014C	仅适用于 MAC+PCS/PMA 32 位（前提是针对 Versal® 平台启用时间戳）。
0x0154	GT_WIZ_CONTROL_REG	仅限 Versal 器件

## 10G/25G Ethernet Subsystem 的状态寄存器映射

状态寄存器可用于指示系统运行状况。这些寄存器为只读，并且读取操作会清除相应的寄存器。

按以下条件清除状态寄存器：

- 应用 `s_axi_aresetn` 会同时清除 TX 和 RX 状态寄存器
- 读取特定的状态寄存器时（读取时清除）
- 应用 `rx_reset` 仅清除 RX 状态寄存器
- 应用 `tx_reset` 仅清除 TX 状态寄存器

表 47：状态寄存器映射

十六进制地址	寄存器名称	注释
0x0400	STAT_TX_STATUS_REG1: 0400	
0x0404	STAT_RX_STATUS_REG1: 0404	
0x0408	STAT_STATUS_REG1: 0408	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x040C	STAT_RX_BLOCK_LOCK_REG: 040C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x043C	STAT_RX_RSFEC_STATUS_REG: 043C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0448	STAT_RX_FEC_STATUS_REG: 0448	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x044C	STAT_TX_RSFEC_STATUS_REG: 044C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0450	STAT_TX_FLOW_CONTROL_REG1: 0450	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0454	STAT_RX_FLOW_CONTROL_REG1: 0454	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0458	STAT_AN_STATUS: 0458	仅适用于 MAC+PCS 变体和仅限 PCS 变体

表 47：状态寄存器映射 (续)

十六进制地址	寄存器名称	注释
0x045C	STAT_AN_ABILITY: 045C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0460	STAT_AN_LINK_CTL: 0460	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x09F0	STAT_AN_LINK_CTL2: 09F0	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0464	STAT_LT_STATUS_REG1: 0464	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0468	STAT_LT_STATUS_REG2: 0468	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x046C	STAT_LT_STATUS_REG3: 046C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0470	STAT_LT_STATUS_REG4: 0470	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0474	STAT_LT_COEFFICIENT0_REG: 0474	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0494	STAT_RX_VALID_CTRL_CODE: 0494	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0498	STAT_CORE_SPEED_REG: 0498	
0x049C	STAT_TSN_REG: 0x049C	仅当启用 802.1cm Preemption 功能时才适用
0x04A0	STAT_GT_WIZ_REG	

## 统计数据计数器

统计数据计数器提供流量分类和错误计数的直方图。根据 MODE\_REG[30] (`tick_reg_mode_sel`) 的值，可通过将 `pm_tick` 设为 1 或者向 TICK\_REG 寄存器写入 1 来读取这些计数器。当 `MODE_REG[30] = 0` 时，使用 `pm_tick` 信号，当 `MODE_REG[30] = 1` (1 为默认值) 时，则使用 TICK\_REG。

这些计数器采用一个内部累加器。写入 TICK\_REG 寄存器会导致将累加的计数推送到可读的 STAT\_\*\_MSB/LSB 寄存器，同时将累加器清零。随后，即可读取 STAT\_\*\_MSB/LSB 寄存器。这样统计数据计数器中存储的所有值都会按相同时间间隔显示快照。

**注释：**这些可读的 STAT\_\*\_MSB/LSB 寄存器不可复位。这可能导致在复位之后且在 TICK\_REG 写入之前存在未知的数据。

STAT\_CYCLE\_COUNT\_MSB/LSB 寄存器包含 TICK\_REG 写入操作间隔的 RX 核时钟周期的计数。这样即可便于提供基于时间间隔的统计数据。

将根据以下条件来清除统计数据计数器寄存器：

- 应用 `s_axi_aresetn` 同时清除 TX 和 RX 统计数据计数器寄存器
- 应用 `pm_tick` 同时清除 TX 和 RX 统计数据计数器寄存器
- 应用 `rx_reset` 将只清除 RX 统计数据计数器寄存器
- 应用 `tx_reset` 将只清除 TX 统计数据计数器寄存器

表 48：统计数据计数器

十六进制地址	寄存器名称	注释
0x0500	STATUS_CYCLE_COUNT_LSB: 0500	
0x0504	STATUS_CYCLE_COUNT_MSB: 0504	
0x0648	STAT_RX_FRAMING_ERR LSB: 0648	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x064C	STAT_RX_FRAMING_ERR MSB: 064C	仅适用于 MAC+PCS 变体和仅限 PCS 变体

表 48：统计数据计数器 (续)

十六进制地址	寄存器名称	注释
0x0660	STAT_RX_BAD_CODE_LSB: 0660	
0x0664	STAT_RX_BAD_CODE_MSB: 0664	
0x0668	STAT_RX_ERROR_LSB: 0668	仅限 PCS 变体
0x066C	STAT_RX_ERROR_MSB: 066C	仅限 PCS 变体
0x0670	STAT_RX_RSFEC_CORRECTED_CW_INC_LSB: 0670	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0674	STAT_RX_RSFEC_CORRECTED_CW_INC_MSB: 0674	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0678	STAT_RX_RSFEC_UNCORRECTED_CW_INC_LSB: 0678	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x067C	STAT_RX_RSFEC_UNCORRECTED_CW_INC_MSB: 067C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0680	STAT_RX_RSFEC_ERR_COUNT0_INC_LSB: 0680	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0684	STAT_RX_RSFEC_ERR_COUNT0_INC_MSB: 0684	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x06A0	STAT_TX_FRAME_ERROR_LSB: 06A0	
0x06A4	STAT_TX_FRAME_ERROR_MSB: 06A4	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0700	STAT_TX_TOTAL_PACKETS_LSB: 0700	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0704	STAT_TX_TOTAL_PACKETS_MSB: 0704	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0708	STAT_TX_TOTAL_GOOD_PACKETS_LSB: 0708	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x070C	STAT_TX_TOTAL_GOOD_PACKETS_MSB: 070C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0710	STAT_TX_TOTAL_BYTES_LSB: 0710	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0714	STAT_TX_TOTAL_BYTES_MSB: 0714	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0718	STAT_TX_TOTAL_GOOD_BYTES_LSB: 0718	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0720	STAT_TX_PACKET_64_BYTES_LSB: 0720	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0724	STAT_TX_PACKET_64_BYTES_MSB: 0724	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0728	STAT_TX_PACKET_65_127_BYTES_LSB: 0728	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x072C	STAT_TX_PACKET_65_127_BYTES_MSB: 072C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0730	STAT_TX_PACKET_128_255_BYTES_LSB: 0730	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0734	STAT_TX_PACKET_128_255_BYTES_MSB: 0734	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0738	STAT_TX_PACKET_256_511_BYTES_LSB: 0738	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x073C	STAT_TX_PACKET_256_511_BYTES_MSB: 073C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0740	STAT_TX_PACKET_512_1023_BYTES_LSB: 0740	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0744	STAT_TX_PACKET_512_1023_BYTES_MSB: 0744	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0748	STAT_TX_PACKET_1024_1518_BYTES_LSB: 0748	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x074C	STAT_TX_PACKET_1024_1518_BYTES_MSB: 074C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0750	STAT_TX_PACKET_1519_1522_BYTES_LSB: 0750	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0754	STAT_TX_PACKET_1519_1522_BYTES_MSB: 0754	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0758	STAT_TX_PACKET_1523_1548_BYTES_LSB: 0758	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x075C	STAT_TX_PACKET_1523_1548_BYTES_MSB: 075C	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0760	STAT_TX_PACKET_1549_2047_BYTES_LSB: 0760	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0764	STAT_TX_PACKET_1549_2047_BYTES_MSB: 0764	仅适用于 MAC+PCS 变体和仅限 MAC 变体
0x0768	STAT_TX_PACKET_2048_4095_BYTES_LSB: 0768	仅适用于 MAC+PCS 变体和仅限 MAC 变体

表48：统计数据计数器(续)

十六进制地址	寄存器名称	注释
0x076C	STAT_TX_PACKET_2048_4095_BYTES_MSB: 076C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0770	STAT_TX_PACKET_4096_8191_BYTES_LSB: 0770	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0774	STAT_TX_PACKET_4096_8191_BYTES_MSB: 0774	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0778	STAT_TX_PACKET_8192_9215_BYTES_LSB: 0778	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x077C	STAT_TX_PACKET_8192_9215_BYTES_MSB: 077C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0780	STAT_TX_PACKET_LARGE_LSB: 0780	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0784	STAT_TX_PACKET_LARGE_MSB: 0784	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0788	STAT_TX_PACKET_SMALL_LSB: 0788	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x078C	STAT_TX_PACKET_SMALL_MSB: 078C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07B8	STAT_TX_BAD_FCS_LSB: 07B8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07BC	STAT_TX_BAD_FCS_MSB: 07BC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07D0	STAT_TX_UNICAST_LSB: 07D0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07D4	STAT_TX_UNICAST_MSB: 07D4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07D8	STAT_TX_MULTICAST_LSB: 07D8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07DC	STAT_TX_MULTICAST_MSB: 07DC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07E0	STAT_TX_BROADCAST_LSB: 07E0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07E4	STAT_TX_BROADCAST_MSB: 07E4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07E8	STAT_TX_VLAN_LSB: 07E8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07EC	STAT_TX_VLAN_MSB: 07EC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07F0	STAT_TX_PAUSE_LSB: 07F0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07F4	STAT_TX_PAUSE_MSB: 07F4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07F8	STAT_TX_USER_PAUSE_LSB: 07F8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x07FC	STAT_TX_USER_PAUSE_MSB: 07FC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0808	STAT_RX_TOTAL_PACKETS_LSB: 0808	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x080C	STAT_RX_TOTAL_PACKETS_MSB: 080C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0810	STAT_RX_TOTAL_GOOD_PACKETS_LSB: 0810	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0814	STAT_RX_TOTAL_GOOD_PACKETS_MSB: 0814	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0818	STAT_RX_TOTAL_BYTES_LSB: 0818	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x081C	STAT_RX_TOTAL_BYTES_MSB: 081C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0820	STAT_RX_TOTAL_GOOD_BYTES_LSB: 0820	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0824	STAT_RX_TOTAL_GOOD_BYTES_MSB: 0824	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0828	STAT_RX_PACKET_64_BYTES_LSB: 0828	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x082C	STAT_RX_PACKET_64_BYTES_MSB: 082C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0830	STAT_RX_PACKET_65_127_BYTES_LSB: 0830	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0834	STAT_RX_PACKET_65_127_BYTES_MSB: 0834	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0838	STAT_RX_PACKET_128_255_BYTES_LSB: 0838	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x083C	STAT_RX_PACKET_128_255_BYTES_MSB: 083C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0840	STAT_RX_PACKET_256_511_BYTES_LSB: 0840	仅适用于 MAC+ PCS 变体和仅限 MAC 变体

表 48：统计数据计数器 (续)

十六进制地址	寄存器名称	注释
0x0844	STAT_RX_PACKET_256_511_BYTES_MSB: 0844	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0848	STAT_RX_PACKET_512_1023_BYTES_LSB: 0848	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x084C	STAT_RX_PACKET_512_1023_BYTES_MSB: 084C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0850	STAT_RX_PACKET_1024_1518_BYTES_LSB: 0850	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0854	STAT_RX_PACKET_1024_1518_BYTES_MSB: 0854	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0858	STAT_RX_PACKET_1519_1522_BYTES_LSB: 0858	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x085C	STAT_RX_PACKET_1519_1522_BYTES_MSB: 085C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0860	STAT_RX_PACKET_1523_1548_BYTES_LSB: 0860	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0864	STAT_RX_PACKET_1523_1548_BYTES_MSB: 0864	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0868	STAT_RX_PACKET_1549_2047_BYTES_LSB: 0868	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x086C	STAT_RX_PACKET_1549_2047_BYTES_MSB: 086C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0870	STAT_RX_PACKET_2048_4095_BYTES_LSB: 0870	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0874	STAT_RX_PACKET_2048_4095_BYTES_MSB: 0874	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0878	STAT_RX_PACKET_4096_8191_BYTES_LSB: 0878	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x087C	STAT_RX_PACKET_4096_8191_BYTES_MSB: 087C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0880	STAT_RX_PACKET_8192_9215_BYTES_LSB: 0880	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0884	STAT_RX_PACKET_8192_9215_BYTES_MSB: 0884	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0888	STAT_RX_PACKET_LARGE_LSB: 0888	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x088C	STAT_RX_PACKET_LARGE_MSB: 088C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0890	STAT_RX_PACKET_SMALL_LSB: 0890	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0894	STAT_RX_PACKET_SMALL_MSB: 0894	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0898	STAT_RX_UNDERSIZE_LSB: 0898	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x089C	STAT_RX_UNDERSIZE_MSB: 089C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08A0	STAT_RX_FRAGMENT_LSB: 08A0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08A4	STAT_RX_FRAGMENT_MSB: 08A4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08A8	STAT_RX_OVERSIZE_LSB: 08A8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08AC	STAT_RX_OVERSIZE_MSB: 08AC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08B0	STAT_RX_TOOLONG_LSB: 08B0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08B4	STAT_RX_TOOLONG_MSB: 08B4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08B8	STAT_RX_JABBER_LSB: 08B8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08BC	STAT_RX_JABBER_MSB: 08BC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08C0	STAT_RX_BAD_FCS_LSB: 08C0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08C4	STAT_RX_BAD_FCS_MSB: 08C4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08C8	STAT_RX_PACKET_BAD_FCS_LSB: 08C8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08CC	STAT_RX_PACKET_BAD_FCS_MSB: 08CC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08D0	STAT_RX_STOMPED_FCS_LSB: 08D0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08D4	STAT_RX_STOMPED_FCS_MSB: 08D4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08D8	STAT_RX_UNICAST_LSB: 08D8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体

表 48：统计数据计数器 (续)

十六进制地址	寄存器名称	注释
0x08DC	STAT_RX_UNICAST_MSB: 08DC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08E0	STAT_RX_MULTICAST_LSB: 08E0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08E4	STAT_RX_MULTICAST_MSB: 08E4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08E8	STAT_RX_BROADCAST_LSB: 08E8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08EC	STAT_RX_BROADCAST_MSB: 08EC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08F0	STAT_RX_VLAN_LSB: 08F0	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08F4	STAT_RX_VLAN_MSB: 08F4	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08F8	STAT_RX_PAUSE_LSB: 08F8	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x08FC	STAT_RX_PAUSE_MSB: 08FC	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0900	STAT_RX_USER_PAUSE_LSB: 0900	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0904	STAT_RX_USER_PAUSE_MSB: 0904	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0908	STAT_RX_INRANGEERR_LSB: 0908	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x090C	STAT_RX_INRANGEERR_MSB: 090C	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0910	STAT_RX_TRUNCATED_LSB: 0910	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0914	STAT_RX_TRUNCATED_MSB: 0914	仅适用于 MAC+ PCS 变体和仅限 MAC 变体
0x0918	STAT_RX_TEST_PATTERN_MISMATCH_LSB: 0918	仅限 MAC+PCS 变体
0x091C	STAT_RX_TEST_PATTERN_MISMATCH_MSB: 91C	仅限 MAC+PCS 变体
0x0920	STAT_FEC_INC_CORRECT_COUNT_LSB: 0920	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0924	STAT_FEC_INC_CORRECT_COUNT_MSB: 0924	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0928	STAT_FEC_INC_CANT_CORRECT_COUNT_LSB: 0928	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x092C	STAT_FEC_INC_CANT_CORRECT_COUNT_MSB: 092C	仅适用于 MAC+PCS 变体和仅限 PCS 变体
0x0980	STAT_TX_MM_STATUS_LSB: 0x0980	含 TSN 的 MAC+PCS
0x0984	STAT_TX_MM_STATUS_MSB: 0x0984	含 TSN 的 MAC+PCS
0x0988	STAT_TX_MM_STATUS_LSB: 0x0988	含 TSN 的 MAC+PCS
0x098C	STAT_TX_MM_FRAGMENT_MSB: 0x098C	含 TSN 的 MAC+PCS
0x0990	STAT_TX_MM_HOLD_LSB: 0x0990	含 TSN 的 MAC+PCS
0x0994	STAT_TX_MM_HOLD_MSB: 0x0994	含 TSN 的 MAC+PCS
0x0998	STAT_RX_MM_ASSEMBLY_ERROR_LSB: 0x0998	含 TSN 的 MAC+PCS
0x099C	STAT_RX_MM_ASSEMBLY_ERROR_MSB: 0x099C	含 TSN 的 MAC+PCS
0x09A0	STAT_RX_MM_FRAME_SMD_ERROR_LSB: 0x09A0	含 TSN 的 MAC+PCS
0x09A4	STAT_RX_MM_FRAME_SMD_ERROR_MSB: 0x09A4	含 TSN 的 MAC+PCS
0x09A8	STAT_RX_MM_FRAME_ASSEMBLY_OK_LSB: 0x09A8	含 TSN 的 MAC+PCS
0x09AC	STAT_RX_MM_FRAME_ASSEMBLY_OK_MSB: 0x09AC	含 TSN 的 MAC+PCS
0x09B0	STAT_RX_MM_FRAGMENT_LSB: 0x09B0	含 TSN 的 MAC+PCS
0x09B4	STAT_RX_MM_FRAGMENT_MSB: 0x09B4	含 TSN 的 MAC+PCS

这些统计数据计数器将可用于 MAC+PCS/PMA 64 位变体（不启用 TX/RX 流量控制 GUI 选项）。对于 MAC+PCS/PMA 32 位、Ethernet MAC 64 位和 Preemption 变体，仅当 GUI 中启用 TX/RX 流量控制时，这些计数器才会更新。

## 寄存器描述

本节包含配置寄存器的描述。如果位元字段中所描述的功能在 IP 核中不存在，那么位元字段将还原为“RESERVED”（保留）。

### 10G/25G Ethernet Subsystem 的配置寄存器

下表定义了配置寄存器的位赋值。

寄存器或寄存器中的位元字段可供访问，用于执行读写 (RW)、只写 (WO) 或只读 (RO) 操作。显示的默认值均为十进制值，并在完成 s\_axi\_aresetn 后生效。

如需了解每个信号的描述，请参阅 [端口描述 - MAC+PCS 变体](#)。

#### GT\_RESET\_REG: 0000

表 49: GT\_RESET\_REG: 0000

位	默认	类型	信号
0	0	RW	ctl_gt_reset_all 在写入寄存器上，这是清除信号。
1	0	RW	ctl_gt_rx_reset
2	0	RW	ctl_gt_tx_reset

#### RESET\_REG: 0004

表 50: RESET\_REG: 0004

位	默认	类型	信号
0	0	RW	rx_serdes_reset
28	0	RW	ctl_an_reset  注释：在写入寄存器上，这是清除信号。  仅当选中“Auto Negotiation/Link Training Logic”（自动协商/链路训练逻辑）功能特性时，此信号才可用。
29	0	RW	tx_serdes_reset
30	0	RW	rx_reset
31	0	RW	tx_reset

#### MODE\_REG: 0008

表 51: MODE\_REG: 0008

位	默认	类型	信号
0	1	RW	en_wr_slverr_indication
1	1	RW	en_rd_slverr_indication
30	1	RW	tick_reg_mode_sel

表 51: MODE\_REG: 0008 (续)

位	默认	类型	信号
31	0	RW	ctl_local_loopback

注释：

1. mode\_reg 寄存器的第 0 位和第 1 位支持灵活禁用和启用 slv 误差。用户可向这些位写入“0”来禁止报告 slv 错误。

## CONFIGURATION\_TX\_REG1: 000C

表 52: CONFIGURATION\_TX\_REG1: 000C

位	默认	类型	信号
0	1	RW	ctl_tx_enable <sup>1</sup>
1	1	RW	ctl_tx_fcs_ins_enable <sup>1</sup>
2	0	RW	ctl_tx_ignore_fcs <sup>1</sup>
3	0	RW	ctl_tx_send_lfi <sup>1</sup>
4	0	RW	ctl_tx_send_rf1 <sup>1</sup>
5	0	RW	ctl_tx_send_idle <sup>1</sup>
13:10	12	RW	ctl_tx_ipg_value <sup>1</sup>
14	0	RW	ctl_tx_test_pattern
15	0	RW	ctl_tx_test_pattern_enable
16	0	RW	ctl_tx_test_pattern_select
17	0	RW	ctl_tx_data_pattern_select
18	0	RW	ctl_tx_custom_preamble_enable <sup>1</sup>
23	0	RW	ctl_tx_prbs31_test_pattern_enable <sup>2</sup>

注释：

1. 仅限 MAC+PCS 变体。
2. 仅限 PCS 变体。

## CONFIGURATION\_RX\_REG1: 0014

表 53: CONFIGURATION\_RX\_REG1: 0014

位	默认	类型	信号
0	1	RW	ctl_rx_enable <sup>1</sup>
1	1	RW	ctl_rx_delete_fcs <sup>1</sup>
2	0	RW	ctl_rx_ignore_fcs <sup>1</sup>
3	0	RW	ctl_rx_process_lfi <sup>1</sup>
4	1	RW	ctl_rx_check_sfd <sup>1</sup>
5	1	RW	ctl_rx_check_preamble <sup>1</sup>
6	0	RW	ctl_rx_force_resync <sup>1</sup>
7	0	RW	ctl_rx_test_pattern
8	0	RW	ctl_rx_test_pattern_enable
9	0	RW	ctl_rx_data_pattern_select

表 53: CONFIGURATION\_RX\_REG1: 0014 (续)

位	默认	类型	信号
10	-	-	保留
11	0	RW	ctl_rx_custom_preamble_enable
12	0	RW	ctl_rx_prbs31_test_pattern_enable <sup>2</sup>

注释:

1. 仅限 MAC+PCS 变体
2. 仅限 PCS 变体

## CONFIGURATION\_RX\_MTU: 0018

表 54: CONFIGURATION\_RX\_MTU: 0018

位	默认	类型	信号
7:0	64	RW	ctl_rx_min_packet_len
30:16	9600	RW	ctl_rx_max_packet_len

## CONFIGURATION\_VL\_LENGTH\_REG: 001C

表 55: CONFIGURATION\_VL\_LENGTH\_REG: 001C

位	默认	类型	信号
15:0	20479	RW	ctl_tx_vl_length_minus1
31:16	20479	RW	ctl_rx_vl_length_minus1

## TICK\_REG: 0020

表 56: TICK\_REG: 0020

位	默认	类型	信号
0	0	WO	tick_reg 在写入寄存器上，这是清除信号。

## CONFIGURATION\_REVISION\_REG: 0024

表 57: CONFIGURATION\_REVISION\_REG: 0024

位	默认	类型	信号
7:0	3	RO	major_rev
15:8	0	RO	minor_rev
31:24	0	RO	patch_rev

**CONFIGURATION\_TX\_TEST\_PAT\_SEED\_A\_LSB: 0028**

表 58: CONFIGURATION\_TX\_TEST\_PAT\_SEED\_A\_LSB: 0028

位	默认	类型	信号
31:0	0	RW	ctl_tx_test_pattern_seed_a[31:0]

**CONFIGURATION\_TX\_TEST\_PAT\_SEED\_A\_MSB: 002C**

表 59: CONFIGURATION\_TX\_TEST\_PAT\_SEED\_A\_MSB: 002C

位	默认	类型	信号
25:0	0	RW	ctl_tx_test_pattern_seed_a[57:32]

**CONFIGURATION\_TX\_TEST\_PAT\_SEED\_B\_LSB: 0030**

表 60: CONFIGURATION\_TX\_TEST\_PAT\_SEED\_B\_LSB: 0030

位	默认	类型	信号
31:0	0	RW	ctl_tx_test_pattern_seed_b[31:0]

**CONFIGURATION\_TX\_TEST\_PAT\_SEED\_B\_MSB: 0034**

表 61: CONFIGURATION\_TX\_TEST\_PAT\_SEED\_B\_MSB: 0034

位	默认	类型	信号
25:0	0	RW	ctl_tx_test_pattern_seed_b[57:32]

**CONFIGURATION\_1588\_REG: 0038**

表 62: CONFIGURATION\_1588\_REG: 0038

位	默认	类型	信号
0	0	RW	ctl_tx_ptp_1step_enable
2	0	RW	ctl_ptp_transpclk_mode
26:16	0	RW	ctl_tx_ptp_latency_adjust

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REG1: 0040**

表 63: CONFIGURATION\_TX\_FLOW\_CONTROL\_REG1: 0040

位	默认	类型	信号
8:0	0	RW	ctl_tx_pause_enable

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG1: 0044**

表 64: CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG1: 0044

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_refresh_timer0
31:16	0	RW	ctl_tx_pause_refresh_timer1

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG2: 0048**

表 65: CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG2: 0048

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_refresh_timer2
31:16	0	RW	ctl_tx_pause_refresh_timer3

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG3: 004C**

表 66: CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG3: 004C

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_refresh_timer4
31:16	0	RW	ctl_tx_pause_refresh_timer5

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG4: 0050**

表 67: CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG4: 0050

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_refresh_timer6
31:16	0	RW	ctl_tx_pause_refresh_timer7

**CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG5: 0054**

表 68: CONFIGURATION\_TX\_FLOW\_CONTROL\_REFRESH\_REG5: 0054

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_refresh_timer8

**CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG1: 0058**

表 69: CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG1: 0058

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_quanta0
31:16	0	RW	ctl_tx_pause_quanta1

**CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG2: 005C**

表 70: CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG2: 005C

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_quanta2
31:16	0	RW	ctl_tx_pause_quanta3

**CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG3: 0060**

表 71: CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG3: 0060

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_quanta4
31:16	0	RW	ctl_tx_pause_quanta5

**CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG4: 0064**

表 72: CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG4: 0064

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_quanta6
31:16	0	RW	ctl_tx_pause_quanta7

**CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG5: 0068**

表 73: CONFIGURATION\_TX\_FLOW\_CONTROL\_QUANTA\_REG5: 0068

位	默认	类型	信号
15:0	0	RW	ctl_tx_pause_quanta8

**CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_ETYPE\_OP\_REG: 006C**

表 74: CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_ETYPE\_OP\_REG: 006C

位	默认	类型	信号
15:0	34824	RW	ctl_tx_etherstype_ppp
31:16	257	RW	ctl_tx_opcode_ppp

**CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_ETYPE\_OP\_REG: 0070**

表 75: CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_ETYPE\_OP\_REG: 0070

位	默认	类型	信号
15:0	34824	RW	ctl_tx_etherstype_gpp
31:16	1	RW	ctl_tx_opcode_gpp

**CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_DA\_REG\_LSB: 0074**

表 76: CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_DA\_REG\_LSB: 0074

位	默认	类型	信号
31:0	0	RW	ctl_tx_da_gpp[31:0]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_DA\_REG\_MSB: 0078**

表 77: CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_DA\_REG\_MSB: 0078

位	默认	类型	信号
15:0	0	RW	ctl_tx_da_gpp[47:32]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_SA\_REG\_LSB: 007C**

表 78: CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_SA\_REG\_LSB: 007C

位	默认	类型	信号
31:0	0	RW	ctl_tx_sa_gpp[31:0]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_SA\_REG\_MSB: 0080**

表 79: CONFIGURATION\_TX\_FLOW\_CONTROL\_GPP\_SA\_REG\_MSB: 0080

位	默认	类型	信号
15:0	0	RW	ctl_tx_sa_gpp[47:32]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_DA\_REG\_LSB: 0084**

表 80: CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_DA\_REG\_LSB: 0084

位	默认	类型	信号
31:0	0	RW	ctl_tx_da_ppp[31:0]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_DA\_REG\_MSB: 0088**

表 81: CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_DA\_REG\_MSB: 0088

位	默认	类型	信号
15:0	0	RW	ctl_tx_da_ppp[47:32]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_SA\_REG\_LSB: 008C**

表 82: CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_SA\_REG\_LSB: 008C

位	默认	类型	信号
31:0	0	RW	ctl_tx_sa_ppp[31:0]

**CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_SA\_REG\_MSB: 0090**

表 83: CONFIGURATION\_TX\_FLOW\_CONTROL\_PPP\_SA\_REG\_MSB: 0090

位	默认	类型	信号
15:0	0	RW	ctl_tx_sa_ppp[47:32]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_REG1: 0094**

表 84: CONFIGURATION\_RX\_FLOW\_CONTROL\_REG1: 0094

位	默认	类型	信号
8:0	0	RW	ctl_rx_pause_enable
9	0	RW	ctl_rx_forward_control
10	0	RW	ctl_rx_enable_gcp
11	0	RW	ctl_rx_enable_pcp
12	0	RW	ctl_rx_enable_gpp
13	0	RW	ctl_rx_enable_ppp
14	0	RW	ctl_rx_check_ack

**CONFIGURATION\_RX\_FLOW\_CONTROL\_REG2: 0098**

表 85: CONFIGURATION\_RX\_FLOW\_CONTROL\_REG2: 0098

位	默认	类型	信号
0	0	RW	ctl_rx_check_mcast_gcp
1	0	RW	ctl_rx_check_unicast_gcp
2	0	RW	ctl_rx_check_sa_gcp
3	0	RW	ctl_rx_check_etype_gcp
4	0	RW	ctl_rx_check_opcode_gcp
5	0	RW	ctl_rx_check_mcast_pcp
6	0	RW	ctl_rx_check_unicast_pcp
7	0	RW	ctl_rx_check_sa_pcp
8	0	RW	ctl_rx_check_etype_pcp
9	0	RW	ctl_rx_check_opcode_pcp
10	0	RW	ctl_rx_check_mcast_gpp
11	0	RW	ctl_rx_check_unicast_gpp
12	0	RW	ctl_rx_check_sa_gpp

表 85: CONFIGURATION\_RX\_FLOW\_CONTROL\_REG2: 0098 (续)

位	默认	类型	信号
13	0	RW	ctl_rx_check_etype_gpp
14	0	RW	ctl_rx_check_opcode_gpp
15	0	RW	ctl_rx_check_mcast_ppp
16	0	RW	ctl_rx_check_unicast_ppp
17	0	RW	ctl_rx_check_sa_ppp
18	0	RW	ctl_rx_check_etype_ppp
19	0	RW	ctl_rx_check_opcode_ppp

### CONFIGURATION\_RX\_FLOW\_CONTROL\_PPPETYPE\_OP\_REG: 009C

表 86: CONFIGURATION\_RX\_FLOW\_CONTROL\_PPPETYPE\_OP\_REG: 009C

位	默认	类型	信号
15:0	34824	RW	ctl_rx_etype_ppp
31:16	257	RW	ctl_rx_opcode_ppp

### CONFIGURATION\_RX\_FLOW\_CONTROL\_GPPETYPE\_OP\_REG: 00A0

表 87: CONFIGURATION\_RX\_FLOW\_CONTROL\_GPPETYPE\_OP\_REG: 00A0

位	默认	类型	信号
15:0	34824	RW	ctl_rx_etype_gpp
31:16	1	RW	ctl_rx_opcode_gpp

### CONFIGURATION\_RX\_FLOW\_CONTROL\_GCP\_PCP\_TYPE\_REG: 00A4

表 88: CONFIGURATION\_RX\_FLOW\_CONTROL\_GCP\_PCP\_TYPE\_REG: 00A4

位	默认	类型	信号
15:0	34824	RW	ctl_rx_etype_gcp
31:16	34824	RW	ctl_rx_etype_pcp

### CONFIGURATION\_RX\_FLOW\_CONTROL\_PCP\_OP\_REG: 00A8

表 89: CONFIGURATION\_RX\_FLOW\_CONTROL\_PCP\_OP\_REG: 00A8

位	默认	类型	信号
15:0	257	RW	ctl_rx_opcode_min_pcp
31:16	257	RW	ctl_rx_opcode_max_pcp

**CONFIGURATION\_RX\_FLOW\_CONTROL\_GCP\_OP\_REG: 00AC**

表 90: CONFIGURATION\_RX\_FLOW\_CONTROL\_GCP\_OP\_REG: 00AC

位	默认	类型	信号
15:0	1	RW	ctl_rx_opcode_min_gcp
31:16	6	RW	ctl_rx_opcode_max_gcp

**CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG1\_LSB: 00B0**

表 91: CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG1\_LSB: 00B0

位	默认	类型	信号
31:0	0	RW	ctl_rx_pause_da_unicast[31:0]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG1\_MSB: 00B4**

表 92: CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG1\_MSB: 00B4

位	默认	类型	信号
15:0	0	RW	ctl_rx_pause_da_unicast[47:32]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG2\_LSB: 00B8**

表 93: CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG2\_LSB: 00B8

位	默认	类型	信号
31:0	0	RW	ctl_rx_pause_da_multicast[31:0]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG2\_MSB: 00BC**

表 94: CONFIGURATION\_RX\_FLOW\_CONTROL\_DA\_REG2\_MSB: 00BC

位	默认	类型	信号
15:0	0	RW	ctl_rx_pause_da_multicast[47:32]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_SA\_REG1\_LSB: 00C0**

表 95: CONFIGURATION\_RX\_FLOW\_CONTROL\_SA\_REG1\_LSB: 00C0

位	默认	类型	信号
31:0	0	RW	ctl_rx_pause_sa[31:0]

**CONFIGURATION\_RX\_FLOW\_CONTROL\_SA\_REG1\_MSB: 00C4**

表 96: CONFIGURATION\_RX\_FLOW\_CONTROL\_SA\_REG1\_MSB: 00C4

位	默认	类型	信号
15:0	0	RW	ctl_rx_pause_sa[47:32]

**CONFIGURATION\_RSFEC\_REG: 00D0**

表 97: CONFIGURATION\_RSFEC\_REG: 00D0

位	默认	类型	信号
0	0	RW	ctl_rsfec_enable
1	0	RW	ctl_rsfec_consortium_25g
2	0	RW	ctl_rx_rsfec_enable_indication
3	0	RW	ctl_rx_rsfec_enable_correction
5	0	RW	ctl_rsfec_ieee_error_indication_mode

**CONFIGURATION\_FEC\_REG: 00D4**

表 98: CONFIGURATION\_FEC\_REG: 00D4

位	默认	类型	信号
0	0	RW	ctl_fec_rx_enable
1	0	RW	ctl_fec_tx_enable
2	0	RW	ctl_fec_enable_error_to_pcs

**CONFIGURATION\_AN\_CONTROL\_REG1: 00E0**

表 99: CONFIGURATION\_AN\_CONTROL\_REG1: 00E0

位	默认	类型	信号
0	0	RW	ctl_autoneg_enable
1	1	RW	ctl_autoneg_bypass <sup>1</sup>
9:2	0	RW	ctl_an_nonce_seed
10	0	RW	ctl_an_pseudo_sel
11	0	RW	ctl_restart_negotiation
12	0	RW	ctl_an_local_fault

**注释:**

- 对于仿真，复位期间写入的 `ctl_autoneg_bypass` 值为 1。要采用启用 ANLT 的配置进行测试，请以 `ctl_autoneg_enable` 设为 1 且 `ctl_autoneg_bypass` 设为 0 来写入寄存器。

**CONFIGURATION\_AN\_CONTROL\_REG2: 00E4**

表100: CONFIGURATION\_AN\_CONTROL\_REG2: 00E4

位	默认	类型	信号
0	0	RW	ctl_an_pause
1	0	RW	ctl_an_asmdir
16	0	RW	ctl_an_fec_10g_request
17	0	RW	ctl_an_fec_ability_override
18	0	RW	ctl_an_cl91_fec_request
19	0	RW	ctl_an_cl91_fec_ability
20	0	RW	ctl_an_fec_25g_rs_request
21	0	RW	ctl_an_fec_25g_baser_request

**CONFIGURATION\_AN\_ABILITY: 00F8**

表101: CONFIGURATION\_AN\_ABILITY: 00F8

位	默认	类型	信号
0	0	RW	ctl_an_ability_1000base_kx
1	0	RW	ctl_an_ability_10gbbase_kx4
2	0	RW	ctl_an_ability_10gbase_kr
3	0	RW	ctl_an_ability_40gbase_kr4
4	0	RW	ctl_an_ability_40gbase_cr4
5	0	RW	ctl_an_ability_100gbase_cr10
6	0	RW	ctl_an_ability_100gbase_kp4
7	0	RW	ctl_an_ability_100gbase_kr4
8	0	RW	ctl_an_ability_100gbase_cr4
9	0	RW	ctl_an_ability_25gbase_krkr_s
10	0	RW	ctl_an_ability_25gbase_krkr
11	0	RW	ctl_an_ability_2_5gbase_kx
12	0	RW	ctl_an_ability_5gbase_kr
13	0	RW	ctl_an_ability_50gbase_krkr
14	0	RW	ctl_an_ability_100gbase_kr2cr2
15	0	RW	ctl_an_ability_200gbase_kr4cr4
16	0	RW	ctl_an_ability_25gbase_kr1
17	0	RW	ctl_an_ability_25gbase_cr1
18	0	RW	ctl_an_ability_50gbase_kr2
19	0	RW	ctl_an_ability_50gbase_cr2

**CONFIGURATION\_LT\_CONTROL\_REG1: 0100**

表 102: CONFIGURATION\_LT\_CONTROL\_REG1: 0100

位	默认	类型	信号
0	0	RW	ctl_lt_training_enable
1	0	RW	ctl_lt_restart_training

**CONFIGURATION\_LT\_TRAINED\_REG: 0104**

表 103: CONFIGURATION\_LT\_TRAINED\_REG: 0104

位	默认	类型	信号
0	0	RW	ctl_lt_rx_trained

**CONFIGURATION\_LT\_PRESET\_REG: 0108**

表 104: CONFIGURATION\_LT\_PRESET\_REG: 0108

位	默认	类型	信号
0	0	RW	ctl_lt_preset_to_tx

**CONFIGURATION\_LT\_INIT\_REG: 010C**

表 105: CONFIGURATION\_LT\_INIT\_REG: 010C

位	默认	类型	信号
0	0	RW	ctl_lt_initialize_to_tx

**CONFIGURATION\_LT\_SEED\_REG0: 0110**

表 106: CONFIGURATION\_LT\_SEED\_REG0: 0110

位	默认	类型	信号
10:0	0	RW	ctl_lt_pseudo_seed0

**CONFIGURATION\_LT\_COEFFICIENT\_REG0: 0130**

表 107: CONFIGURATION\_LT\_COEFFICIENT\_REG0: 0130

位	默认	类型	信号
1:0	0	RW	ctl_lt_k_p1_to_tx0
3:2	0	RW	ctl_lt_k0_to_tx0
5:4	0	RW	ctl_lt_k_m1_to_tx0
7:6	0	RW	ctl_lt_stat_p1_to_tx0
9:8	0	RW	ctl_lt_stat0_to_tx0

表 107: CONFIGURATION\_LT\_COEFFICIENT\_REG0: 0130 (续)

位	默认	类型	信号
11:10	0	RW	ctl_lt_stat_m1_to_tx0

### USER\_REG\_0: 0134

表 108: USER\_REG\_0: 0134

位	默认	类型	信号
31:0	0	RW	user_reg0

### SWITCH\_CORE\_SPEED\_REG: 0138

表 109: SWITCH\_CORE\_SPEED\_REG: 0138

位	默认	类型	信号
0	0	RW	axi_ctl_core_mode_switch

仅适用于“Runtime Switch”模式。写入 1 即可启用 10G 与 25G 之间的模式切换。在写入寄存器上，这是清除信号。该值用作为执行 GT DRP 操作的传输调试模块的输入。

### CONFIGURATION\_1588\_32BIT\_REG: 0x013C

表 110: CONFIGURATION\_1588\_32BIT\_REG: 0x013C

位	默认	类型	信号
0	0	RW	ctl_tx_lat_adj_enb
1	0	RW	ctl_rx_lat_adj_enb
2	0	RW	ctl_tx_timestamp_adj_enb
3	0	RW	ctl_rx_timestamp_adj_enb
4	0	RW	ctl_core_speed

### TX\_CONFIGURATION\_1588\_REG: 0x0140

表 111: TX\_CONFIGURATION\_1588\_REG: 0x0140

位	默认	类型	信号
31:0	0	RW	ctl_tx_latency

### RX\_CONFIGURATION\_1588\_REG: 0x0144

表 112: RX\_CONFIGURATION\_1588\_REG: 0x0144

位	默认	类型	信号
31:0	0	RW	ctl_rx_latency

**CONFIGURATION\_TSN\_REG: 0x019C**

表 113: CONFIGURATION\_TSN\_REG: 0x019C

位	默认	类型	信号
0	1	RW	ctl_en_preempt
1	0	RW	ctl_hold_request
2	0	RW	ctl_disable_verify
3	0	RW	ctl_restart_verify
5:4	2'b00	RW	ctl_addfrag_size[1:0]
15:8	8'h01	RW	ctl_verify_time[7:0]
19:16	4'h3	RW	ctl_verify_limit[3:0]

**VERSAL\_CHANNEL\_NUM\_REG: 0x014C**

表 114: VERSAL\_CHANNEL\_NUM\_REG: 0x014C

位	默认	类型	信号
1:0	2'b0	RW	ctl_gtye5_chan_num

**GT\_WIZ\_CONTROL\_REG: 0154**

注释：仅适用于 Versal 器件。

表 115: GT\_WIZ\_CONTROL\_REG: 0154

位	默认	类型	信号
2:0	3'b000	RW	gtwiz_loopback
7:3	5'b00000	RW	gtwiz_txprecursor
12:8	5'b00000	RW	gtwiz_txpostprecursor
19:13	7'b0000000	RW	gtwiz_txmainprecursor
20	1'b0	RW	gtwiz_rxcdrhold

注释：

- 提及的位描述（对应 gtwiz\_txprecursor、gtwiz\_txpostcursor、gtwiz\_txmaincursor 和 gtwiz\_rxcdrhold）仅对非 GTM Versal 器件有效。
- 对于 Versal GTM 器件，有效的位描述为：寄存器的位 [8:3] 对应 gtwiz\_txprecursor，位 [14:9] 对应 gtwiz\_txpostcursor，位 [21:15] 对应 gtwiz\_txmaincursor，位 [22] 对应 gtwiz\_rxcdrhold。
- 如需了解更多详情，请参阅《Versal ACAP GTY 和 GTYP 收发器架构手册》(AM002)。

**10G/25G Ethernet Subsystem 的状态寄存器**

下表定义了状态寄存器的位赋值。

某些位属于粘滞位，即一旦其值置于高电平或低电平后即加以锁存。此操作以类型 LH（已锁存至高电平）或 LL（已锁存至低电平）来指示。

如需了解每个信号的描述，请参阅 [端口描述 - MAC+PCS 变体](#)。

## STAT\_TX\_STATUS\_REG1: 0400

表 116: STAT\_TX\_STATUS\_REG1: 0400

位	默认	类型	信号
0	0	RO LH	stat_tx_local_fault
1 <sup>1</sup>	0	RO LH	stat_tx_gmii_fifo_ovf_1h_r_out
2 <sup>1</sup>	0	RO LH	stat_tx_gmii_fifo_unf_1h_r_out
7 <sup>2</sup>	0	RO LH	stat_tx_bad_parity

注释:

1. 位 1 和 2 适用于 10G 仅限 MAC 变体。
2. 位 7 仅适用于数据路径奇偶校验功能。

## STAT\_RX\_STATUS\_REG1: 0404

表 117: STAT\_RX\_STATUS\_REG1: 0404

位	默认	类型	信号
0	0	RO LL	stat_rx_status
4	0	RO LH	stat_rx_hi_ber
5	0	RO LH	stat_rx_remote_fault <sup>1</sup>
6	0	RO LH	stat_rx_local_fault
7	0	RO LH	stat_rx_internal_local_fault <sup>1</sup>
8	0	RO LH	stat_rx_received_local_fault <sup>1</sup>
9	0	RO LH	stat_rx_bad_preamble <sup>1</sup>
10	0	RO LH	stat_rx_bad_sfd <sup>1</sup>
11	0	RO LH	stat_rx_got_signal_os <sup>1</sup>

注释:

1. 仅限 MAC+PCS 变体。

## STAT\_STATUS\_REG1: 0408

表 118: STAT\_STATUS\_REG1: 0408

位	默认	类型	信号
0	0	RO LH	stat_tx_fifo_error <sup>1</sup>
4	0	RO LH	stat_tx_ptp_fifo_read_error
5	0	RO LH	stat_tx_ptp_fifo_write_error
16	0	RO LH	stat_rx_fifo_error <sup>1</sup>

注释:

1. 仅限 PCS 变体。

**STAT\_RX\_BLOCK\_LOCK\_REG: 040C**

表 119: STAT\_RX\_BLOCK\_LOCK\_REG: 040C

位	默认	类型	信号
0	1	RO LL	stat_rx_block_lock

**STAT\_RX\_RSFEC\_STATUS\_REG: 043C**

表 120: STAT\_RX\_RSFEC\_STATUS\_REG: 043C

位	默认	类型	信号
0	1	RO LL	stat_rx_rsfec_lane_alignment_status
1	1	RO LL	stat_rx_rsfec_hi_ser

**STAT\_RX\_FEC\_STATUS\_REG: 0448**

表 121: STAT\_RX\_FEC\_STATUS\_REG: 0448

位	默认	类型	信号
0	1	RO LL	stat_fec_rx_lock
16	1	RO LL	stat_fec_lock_error

**STAT\_TX\_RSFEC\_STATUS\_REG: 044C**

表 122: STAT\_TX\_RSFEC\_STATUS\_REG: 044C

位	默认	类型	信号
0	1	RO LL	stat_tx_rsfec_lane_alignment_status

**STAT\_TX\_FLOW\_CONTROL\_REG1: 0450**

表 123: STAT\_TX\_FLOW\_CONTROL\_REG1: 0450

位	默认	类型	信号
8:0	0	RO LH	stat_tx_pause_valid

**STAT\_RX\_FLOW\_CONTROL\_REG1: 0454**

表 124: STAT\_RX\_FLOW\_CONTROL\_REG1: 0454

位	默认	类型	信号
8:0	0	RO LH	stat_rx_pause_req
17:9	0	RO LH	stat_rx_pause_valid

**STAT\_AN\_STATUS: 0458**

表 125: STAT\_AN\_STATUS: 0458

位	默认	类型	信号
0	0	RO	stat_an_fec_enable
1	0	RO	stat_an_rs_fec_enable
2	0	RO	stat_an_autoneg_complete
3	0	RO	stat_an_parallel_detection_fault
4	0	RO	stat_an_tx_pause_enable
5	0	RO	stat_an_rx_pause_enable
6	0	RO LH	stat_an_lp_ability_valid
7	0	RO	stat_an_lp_autoneg_able
8	0	RO	stat_an_lp_pause
9	0	RO	stat_an_lp_asm_dir
10	0	RO	stat_an_lp_rf
11	0	RO	stat_an_lp_fec_10g_ability
12	0	RO	stat_an_lp_fec_10g_request
13	0	RO LH	stat_an_lp_extended_ability_valid
17:14	0	RO	stat_an_lp_ability_extended_fec
18	0	RO	stat_an_lp_fec_25g_rs_request
19	0	RO	stat_an_lp_fec_25g_baser_request

**STAT\_AN\_ABILITY: 045C**

表 126: STAT\_AN\_ABILITY: 045C

位	默认	类型	信号
0	0	RO	stat_an_lp_ability_1000base_kx
1	0	RO	stat_an_lp_ability_10gbase_kx4
2	0	RO	stat_an_lp_ability_10gbase_kr
3	0	RO	stat_an_lp_ability_40gbase_kr4
4	0	RO	stat_an_lp_ability_40gbase_cr4
5	0	RO	stat_an_lp_ability_100gbase_cr10
6	0	RO	stat_an_lp_ability_100gbase_kp4
7	0	RO	stat_an_lp_ability_100gbase_kr4
8	0	RO	stat_an_lp_ability_100gbase_cr4
9	0	RO	stat_an_lp_ability_25gbase_kr_cr_s
10	0	RO	stat_an_lp_ability_25gbase_kr_cr
11	0	RO	stat_an_lp_ability_2_5gbase_kx
12	0	RO	stat_an_lp_ability_5gbase_kr
13	0	RO	stat_an_lp_ability_50gbase_kr_cr
14	0	RO	stat_an_lp_ability_100gbase_kr2_cr2

表 126: STAT\_AN\_ABILITY: 045C (续)

位	默认	类型	信号
15	0	RO	stat_an_lp_ability_200gbase_kr4cr4
16	0	RO	stat_an_lp_ability_25gbase_kr1
17	0	RO	stat_an_lp_ability_25gbase_cr1
18	0	RO	stat_an_lp_ability_50gbase_kr2
19	0	RO	stat_an_lp_ability_50gbase_cr2

## STAT\_AN\_LINK\_CTL: 0460

表 127: STAT\_AN\_LINK\_CTL: 0460

位	默认	类型	信号
1:0	0	RO	stat_an_link_cntl_1000base_kx
3:2	0	RO	stat_an_link_cntl_10gbase_kx4
5:4	0	RO	stat_an_link_cntl_10gbase_kr
7:6	0	RO	stat_an_link_cntl_40gbase_kr4
9:8	0	RO	stat_an_link_cntl_40gbase_cr4
11:10	0	RO	stat_an_link_cntl_100gbase_cr10
13:12	0	RO	stat_an_link_cntl_100gbase_kp4
15:14	0	RO	stat_an_link_cntl_100gbase_kr4
17:16	0	RO	stat_an_link_cntl_100gbase_cr4
19:18	0	RO	stat_an_link_cntl_25gbase_kr_cr_s
21:20	0	RO	stat_an_link_cntl_25gbase_kr_cr
23:22	0	RO	stat_an_link_cntl_2_5gbase_kx
25:24	0	RO	stat_an_link_cntl_5gbase_kr
27:26	0	RO	stat_an_link_cntl_50gbase_kr_cr
29:28	0	RO	stat_an_link_cntl_100gbase_kr2_cr2
31:30	0	RO	stat_an_link_cntl_200gbase_kr4_cr4

## STAT\_AN\_LINK\_CTL2: 09F0

表 128: STAT\_AN\_LINK\_CTL2: 09F0

位	默认	类型	信号
1:0	0	RO	stat_an_link_cntl_25gbase_kr1
3:2	0	RO	stat_an_link_cntl_25gbase_cr1
5:4	0	RO	stat_an_link_cntl_50gbase_kr2
7:6	0	RO	stat_an_link_cntl_50gbase_cr2

**STAT\_LT\_STATUS\_REG1: 0464**

表 129: STAT\_LT\_STATUS\_REG1: 0464

位	默认	类型	信号
0	0	RO	stat_lt_initialize_from_rx
16	0	RO	stat_lt_preset_from_rx

**STAT\_LT\_STATUS\_REG2: 0468**

表 130: STAT\_LT\_STATUS\_REG2: 0468

位	默认	类型	信号
0	0	RO	stat_lt_training
16	0	RO	stat_lt_frame_lock

**STAT\_LT\_STATUS\_REG3: 046C**

表 131: STAT\_LT\_STATUS\_REG3: 046C

位	默认	类型	信号
0	0	RO	stat_lt_signal_detect
16	0	RO	stat_lt_training_fail

**STAT\_LT\_STATUS\_REG4: 0470**

表 132: STAT\_LT\_STATUS\_REG4: 0470

位	默认	类型	信号
0	0	RO LH	stat_lt_rx_sof

**STAT\_LT\_COEFFICIENT0\_REG: 0474**

表 133: STAT\_LT\_COEFFICIENT0\_REG: 0474

位	默认	类型	信号
1:0	0	RO	stat_lt_k_p1_from_rx0
3:2	0	RO	stat_lt_k0_from_rx0
5:4	0	RO	stat_lt_k_m1_from_rx0
7:6	0	RO	stat_lt_stat_p1_from_rx0
9:8	0	RO	stat_lt_stat0_from_rx0
11:10	0	RO	stat_lt_stat_m1_from_rx0

### STAT\_RX\_VALID\_CTRL\_CODE: 0494

表 134: STAT\_RX\_VALID\_CTRL\_CODE: 0494

位	默认	类型	信号
0	0	RO LH	stat_rx_valid_ctrl_code

### STAT\_CORE\_SPEED\_REG: 0498

表 135: STAT\_CORE\_SPEED\_REG: 0498

位	默认	类型	信号
0	GUI 已配置	RO	stat_core_speed
1	GUI 已配置	RO	runtime_switchable

注释:

- 此寄存器仅适用于 64 位 AXI4-Stream 数据路径接口。

00 - 独立 25G

01 - 独立 10G

10 - 运行时可切换 25G

11 - 运行时可切换 10G

### STAT\_TSN\_REG: 0x049C

表 136: STAT\_TSN\_REG: 0x049C

位	默认	类型	信号
1:0	0	RO	stat_tx_mm_verify

### STAT\_GT\_WIZ\_REG: 04A0

表 137: STAT\_GT\_WIZ\_REG: 04A0

位	默认	类型	信号
0	0	RO	gtwiz_reset_tx_done
1	0	RO	gtwiz_reset_rx_done

注释:

- 此寄存器表示 GT 已解复位，恢复的 TX/RX 时钟处于稳定状态。

- 用户需首先读取 0x04A0 寄存器 (STAT\_GT\_WIZ\_REG)，并等待该值变为 2'b11。然后才能执行后续操作，如读取块锁定寄存器 (0x040C)。

## 统计数据计数器

下表定义了统计数据计数器的位赋值。

计数器为 48 位，需 2 个 32 位地址空间，其中包含指定的 MSB 和 LSB。所有计数器的默认值均为 0。当 tick\_reg (或 pm\_tick，如已选中) 读取计数器时，会将其清零，但包含计数器的寄存器仍保留其值。每个计数器饱和值均为 FFFFFFFFFFFFFF (十六进制)。

如需了解每个信号的描述，请参阅 [端口描述 - MAC+PCS 变体](#)。

### **STATUS\_CYCLE\_COUNT\_LSB: 0500**

表 138: STATUS\_CYCLE\_COUNT\_LSB: 0500

位	默认	类型	信号
31:0	0	RO HIST	stat_cycle_count[31:0]

### **STATUS\_CYCLE\_COUNT\_MSB: 0504**

表 139: STATUS\_CYCLE\_COUNT\_MSB: 0504

位	默认	类型	信号
15:0	0	RO HIST	stat_cycle_count[47:32]

### **STAT\_RX\_FRAMING\_ERR\_LSB: 0648**

表 140: STAT\_RX\_FRAMING\_ERR\_LSB: 0648

位	默认	类型	信号
31:0	0	HIST	stat_rx_framing_err_count[31:0]

### **STAT\_RX\_FRAMING\_ERR\_MSB: 064C**

表 141: STAT\_RX\_FRAMING\_ERR\_MSB: 064C

位	默认	类型	信号
15:0	0	HIST	stat_rx_framing_err_count[48:132]

### **STAT\_RX\_BAD\_CODE\_LSB: 0660**

表 142: STAT\_RX\_BAD\_CODE\_LSB: 0660

位	默认	类型	信号
31:0	0	HIST	stat_rx_bad_code_count[31:0]

### **STAT\_RX\_BAD\_CODE\_MSB: 0664**

表 143: STAT\_RX\_BAD\_CODE\_MSB: 0664

位	默认	类型	信号
15:0	0	HIST	stat_rx_bad_code_count[47:32]

**STAT\_RX\_ERROR\_LSB: 0668**

表 144: STAT\_RX\_ERROR\_LSB: 0668

位	默认	类型	信号
31:0	0	HIST	stat_rx_error_count[31:0]

**STAT\_RX\_ERROR\_MSB: 066C**

表 145: STAT\_RX\_ERROR\_MSB: 066C

位	默认	类型	信号
15:0	0	HIST	stat_rx_error_count[47:32]

**STAT\_RX\_RSFEC\_CORRECTED\_CW\_INC\_LSB: 0670**

表 146: STAT\_RX\_RSFEC\_CORRECTED\_CW\_INC\_LSB: 0670

位	默认	类型	信号
31:0	0	HIST	stat_rx_rsfec_corrected_cw_inc_count[31:0]

**STAT\_RX\_RSFEC\_CORRECTED\_CW\_INC\_MSB: 0674**

表 147: STAT\_RX\_RSFEC\_CORRECTED\_CW\_INC\_MSB: 0674

位	默认	类型	信号
15:0	0	HIST	stat_rx_rsfec_corrected_cw_inc_count[47:32]

**STAT\_RX\_RSFEC\_UNCORRECTED\_CW\_INC\_LSB: 0678**

表 148: STAT\_RX\_RSFEC\_UNCORRECTED\_CW\_INC\_LSB: 0678

位	默认	类型	信号
31:0	0	HIST	stat_rx_rsfec_uncorrected_cw_inc_count[31:0]

**STAT\_RX\_RSFEC\_UNCORRECTED\_CW\_INC\_MSB: 067C**

表 149: STAT\_RX\_RSFEC\_UNCORRECTED\_CW\_INC\_MSB: 067C

位	默认	类型	信号
15:0	0	HIST	stat_rx_rsfec_uncorrected_cw_inc_count[47:32]

**STAT\_RX\_RSFEC\_ERR\_COUNT0\_INC\_LSB: 0680**

表 150: STAT\_RX\_RSFEC\_ERR\_COUNT0\_INC\_LSB: 0680

位	默认	类型	信号
31:0	0	HIST	stat_rx_rsfec_err_count_inc_count[31:0]

**STAT\_RX\_RSFEC\_ERR\_COUNT0\_INC\_MSB: 0684**

表 151: STAT\_RX\_RSFEC\_ERR\_COUNT0\_INC\_MSB: 0684

位	默认	类型	信号
15:0	0	HIST	stat_rx_rsfec_err_count_inc_count[47:32]

**STAT\_TX\_FRAME\_ERROR\_LSB: 06A0**

表 152: STAT\_TX\_FRAME\_ERROR\_LSB: 06A0

位	默认	类型	信号
31:0	0	HIST	stat_tx_frame_error_count[31:0]

**STAT\_TX\_FRAME\_ERROR\_MSB: 06A4**

表 153: STAT\_TX\_FRAME\_ERROR\_MSB: 06A4

位	默认	类型	信号
15:0	0	HIST	stat_tx_frame_error_count[47:32]

**STAT\_TX\_TOTAL\_PACKETS\_LSB: 0700**

表 154: STAT\_TX\_TOTAL\_PACKETS\_LSB: 0700

位	默认	类型	信号
31:0	0	HIST	stat_tx_total_packets_count[31:0]

**STAT\_TX\_TOTAL\_PACKETS\_MSB: 0704**

表 155: STAT\_TX\_TOTAL\_PACKETS\_MSB: 0704

位	默认	类型	信号
15:0	0	HIST	stat_tx_total_packets_count[47:32]

**STAT\_TX\_TOTAL\_GOOD\_PACKETS\_LSB: 0708**

表 156: STAT\_TX\_TOTAL\_GOOD\_PACKETS\_LSB: 0708

位	默认	类型	信号
31:0	0	HIST	stat_tx_total_good_packets_count[31:0]

**STAT\_TX\_TOTAL\_GOOD\_PACKETS\_MSB: 070C**

表 157: STAT\_TX\_TOTAL\_GOOD\_PACKETS\_MSB: 070C

位	默认	类型	信号
15:0	0	HIST	stat_tx_total_good_packets_count[47:32]

**STAT\_TX\_TOTAL\_BYTES\_LSB: 0710**

表 158: STAT\_TX\_TOTAL\_BYTES\_LSB: 0710

位	默认	类型	信号
31:0	0	HIST	stat_tx_total_bytes_count[31:0]

**STAT\_TX\_TOTAL\_BYTES\_MSB: 0714**

表 159: STAT\_TX\_TOTAL\_BYTES\_MSB: 0714

位	默认	类型	信号
15:0	0	HIST	stat_tx_total_bytes_count[47:32]

**STAT\_TX\_TOTAL\_GOOD\_BYTES\_LSB: 0718**

表 160: STAT\_TX\_TOTAL\_GOOD\_BYTES\_LSB: 0718

位	默认	类型	信号
31:0	0	HIST	stat_tx_total_good_bytes_count[31:0]

**STAT\_TX\_TOTAL\_GOOD\_BYTES\_MSB: 071C**

表 161: STAT\_TX\_TOTAL\_GOOD\_BYTES\_MSB: 071C

位	默认	类型	信号
15:0	0	HIST	stat_tx_total_good_bytes_count[47:32]

**STAT\_TX\_PACKET\_64\_BYTES\_LSB: 0720**

表 162: STAT\_TX\_PACKET\_64\_BYTES\_LSB: 0720

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_64_bytes_count[31:0]

**STAT\_TX\_PACKET\_64\_BYTES\_MSB: 0724**

表 163: STAT\_TX\_PACKET\_64\_BYTES\_MSB: 0724

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_64_bytes_count[47:32]

**STAT\_TX\_PACKET\_65\_127\_BYTES\_LSB: 0728**

表 164: STAT\_TX\_PACKET\_65\_127\_BYTES\_LSB: 0728

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_65_127_bytes_count[31:0]

**STAT\_TX\_PACKET\_65\_127\_BYTES\_MSB: 072C**

表 165: STAT\_TX\_PACKET\_65\_127\_BYTES\_MSB: 072C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_65_127_bytes_count[47:32]

**STAT\_TX\_PACKET\_128\_255\_BYTES\_LSB: 0730**

表 166: STAT\_TX\_PACKET\_128\_255\_BYTES\_LSB: 0730

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_128_255_bytes_count[31:0]

**STAT\_TX\_PACKET\_128\_255\_BYTES\_MSB: 0734**

表 167: STAT\_TX\_PACKET\_128\_255\_BYTES\_MSB: 0734

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_128_255_bytes_count[47:32]

**STAT\_TX\_PACKET\_256\_511\_BYTES\_LSB: 0738**

表 168: STAT\_TX\_PACKET\_256\_511\_BYTES\_LSB: 0738

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_256_511_bytes_count[31:0]

**STAT\_TX\_PACKET\_256\_511\_BYTES\_MSB: 073C**

表 169: STAT\_TX\_PACKET\_256\_511\_BYTES\_MSB: 073C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_256_511_bytes_count[47:32]

**STAT\_TX\_PACKET\_512\_1023\_BYTES\_LSB: 0740**

表 170: STAT\_TX\_PACKET\_512\_1023\_BYTES\_LSB: 0740

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_512_1023_bytes_count[31:0]

**STAT\_TX\_PACKET\_512\_1023\_BYTES\_MSB: 0744**

表 171: STAT\_TX\_PACKET\_512\_1023\_BYTES\_MSB: 0744

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_512_1023_bytes_count[47:32]

**STAT\_TX\_PACKET\_1024\_1518\_BYTES\_LSB: 0748**

表 172: STAT\_TX\_PACKET\_1024\_1518\_BYTES\_LSB: 0748

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_1024_1518_bytes_count[31:0]

**STAT\_TX\_PACKET\_1024\_1518\_BYTES\_MSB: 074C**

表 173: STAT\_TX\_PACKET\_1024\_1518\_BYTES\_MSB: 074C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_1024_1518_bytes_count[47:32]

**STAT\_TX\_PACKET\_1519\_1522\_BYTES\_LSB: 0750**

表 174: STAT\_TX\_PACKET\_1519\_1522\_BYTES\_LSB: 0750

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_1519_1522_bytes_count[31:0]

**STAT\_TX\_PACKET\_1519\_1522\_BYTES\_MSB: 0754**

表 175: STAT\_TX\_PACKET\_1519\_1522\_BYTES\_MSB: 0754

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_1519_1522_bytes_count[47:32]

**STAT\_TX\_PACKET\_1523\_1548\_BYTES\_LSB: 0758**

表 176: STAT\_TX\_PACKET\_1523\_1548\_BYTES\_LSB: 0758

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_1523_1548_bytes_count[31:0]

**STAT\_TX\_PACKET\_1523\_1548\_BYTES\_MSB: 075C**

表 177: STAT\_TX\_PACKET\_1523\_1548\_BYTES\_MSB: 075C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_1523_1548_bytes_count[47:32]

**STAT\_TX\_PACKET\_1549\_2047\_BYTES\_LSB: 0760**

表 178: STAT\_TX\_PACKET\_1549\_2047\_BYTES\_LSB: 0760

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_1549_2047_bytes_count[31:0]

**STAT\_TX\_PACKET\_1549\_2047\_BYTES\_MSB: 0764**

表 179: STAT\_TX\_PACKET\_1549\_2047\_BYTES\_MSB: 0764

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_1549_2047_bytes_count[47:32]

**STAT\_TX\_PACKET\_2048\_4095\_BYTES\_LSB: 0768**

表 180: STAT\_TX\_PACKET\_2048\_4095\_BYTES\_LSB: 0768

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_2048_4095_bytes_count[31:0]

**STAT\_TX\_PACKET\_2048\_4095\_BYTES\_MSB: 076C**

表 181: STAT\_TX\_PACKET\_2048\_4095\_BYTES\_MSB: 076C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_2048_4095_bytes_count[47:32]

**STAT\_TX\_PACKET\_4096\_8191\_BYTES\_LSB: 0770**

表 182: STAT\_TX\_PACKET\_4096\_8191\_BYTES\_LSB: 0770

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_4096_8191_bytes_count[31:0]

**STAT\_TX\_PACKET\_4096\_8191\_BYTES\_MSB: 0774**

表 183: STAT\_TX\_PACKET\_4096\_8191\_BYTES\_MSB: 0774

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_4096_8191_bytes_count[47:32]

**STAT\_TX\_PACKET\_8192\_9215\_BYTES\_LSB: 0778**

表 184: STAT\_TX\_PACKET\_8192\_9215\_BYTES\_LSB: 0778

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_8192_9215_bytes_count[31:0]

**STAT\_TX\_PACKET\_8192\_9215\_BYTES\_MSB: 077C**

表 185: STAT\_TX\_PACKET\_8192\_9215\_BYTES\_MSB: 077C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_8192_9215_bytes_count[47:32]

**STAT\_TX\_PACKET\_LARGE\_LSB: 0780**

表 186: STAT\_TX\_PACKET\_LARGE\_LSB: 0780

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_large_count[31:0]

**STAT\_TX\_PACKET\_LARGE\_MSB: 0784**

表 187: STAT\_TX\_PACKET\_LARGE\_MSB: 0784

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_large_count[47:32]

**STAT\_TX\_PACKET\_SMALL\_LSB: 0788**

表 188: STAT\_TX\_PACKET\_SMALL\_LSB: 0788

位	默认	类型	信号
31:0	0	HIST	stat_tx_packet_small_count[31:0]

**STAT\_TX\_PACKET\_SMALL\_MSB: 078C**

表 189: STAT\_TX\_PACKET\_SMALL\_MSB: 078C

位	默认	类型	信号
15:0	0	HIST	stat_tx_packet_small_count[47:32]

**STAT\_TX\_BAD\_FCS\_LSB: 07B8**

表 190: STAT\_TX\_BAD\_FCS\_LSB: 07B8

位	默认	类型	信号
31:0	0	HIST	stat_tx_bad_fcs_count[31:0]

**STAT\_TX\_BAD\_FCS\_MSB: 07BC**

表 191: STAT\_TX\_BAD\_FCS\_MSB: 07BC

位	默认	类型	信号
15:0	0	HIST	stat_tx_bad_fcs_count[47:32]

**STAT\_TX\_UNICAST\_LSB: 07D0**

表 192: STAT\_TX\_UNICAST\_LSB: 07D0

位	默认	类型	信号
31:0	0	HIST	stat_tx_unicast_count[31:0]

**STAT\_TX\_UNICAST\_MSB: 07D4**

表 193: STAT\_TX\_UNICAST\_MSB: 07D4

位	默认	类型	信号
15:0	0	HIST	stat_tx_unicast_count[47:32]

**STAT\_TX\_MULTICAST\_LSB: 07D8**

表 194: STAT\_TX\_MULTICAST\_LSB: 07D8

位	默认	类型	信号
31:0	0	HIST	stat_tx_multicast_count[31:0]

**STAT\_TX\_MULTICAST\_MSB: 07DC**

表 195: STAT\_TX\_MULTICAST\_MSB: 07DC

位	默认	类型	信号
15:0	0	HIST	stat_tx_multicast_count[47:32]

**STAT\_TX\_BROADCAST\_LSB: 07E0**

表 196: STAT\_TX\_BROADCAST\_LSB: 07E0

位	默认	类型	信号
31:0	0	HIST	stat_tx_broadcast_count[31:0]

**STAT\_TX\_BROADCAST\_MSB: 07E4**

表 197: STAT\_TX\_BROADCAST\_MSB: 07E4

位	默认	类型	信号
15:0	0	HIST	stat_tx_broadcast_count[47:32]

**STAT\_TX\_VLAN\_LSB: 07E8**

表 198: STAT\_TX\_VLAN\_LSB: 07E8

位	默认	类型	信号
31:0	0	HIST	stat_tx_vlan_count[31:0]

**STAT\_TX\_VLAN\_MSB: 07EC**

表 199: STAT\_TX\_VLAN\_MSB: 07EC

位	默认	类型	信号
15:0	0	HIST	stat_tx_vlan_count[47:32]

**STAT\_TX\_PAUSE\_LSB: 07F0**

表 200: STAT\_TX\_PAUSE\_LSB: 07F0

位	默认	类型	信号
31:0	0	HIST	stat_tx_pause_count[31:0]

**STAT\_TX\_PAUSE\_MSB: 07F4**

表 201: STAT\_TX\_PAUSE\_MSB: 07F4

位	默认	类型	信号
15:0	0	HIST	stat_tx_pause_count[47:32]

**STAT\_TX\_USER\_PAUSE\_LSB: 07F8**

表 202: STAT\_TX\_USER\_PAUSE\_LSB: 07F8

位	默认	类型	信号
31:0	0	HIST	stat_tx_user_pause_count[31:0]

**STAT\_TX\_USER\_PAUSE\_MSB: 07FC**

表 203: STAT\_TX\_USER\_PAUSE\_MSB: 07FC

位	默认	类型	信号
15:0	0	HIST	stat_tx_user_pause_count[47:32]

**STAT\_RX\_TOTAL\_PACKETS\_LSB: 0808**

表 204: STAT\_RX\_TOTAL\_PACKETS\_LSB: 0808

位	默认	类型	信号
31:0	0	HIST	stat_rx_total_packets_count[31:0]

**STAT\_RX\_TOTAL\_PACKETS\_MSB: 080C**

表 205: STAT\_RX\_TOTAL\_PACKETS\_MSB: 080C

位	默认	类型	信号
15:0	0	HIST	stat_rx_total_packets_count[47:32]

**STAT\_RX\_TOTAL\_GOOD\_PACKETS\_LSB: 0810**

表 206: STAT\_RX\_TOTAL\_GOOD\_PACKETS\_LSB: 0810

位	默认	类型	信号
31:0	0	HIST	stat_rx_total_good_packets_count[31:0]

**STAT\_RX\_TOTAL\_GOOD\_PACKETS\_MSB: 0814**

表 207: STAT\_RX\_TOTAL\_GOOD\_PACKETS\_MSB: 0814

位	默认	类型	信号
15:0	0	HIST	stat_rx_total_good_packets_count[47:32]

**STAT\_RX\_TOTAL\_BYTES\_LSB: 0818**

表 208: STAT\_RX\_TOTAL\_BYTES\_LSB: 0818

位	默认	类型	信号
31:0	0	HIST	stat_rx_total_bytes_count[31:0]

**STAT\_RX\_TOTAL\_BYTES\_MSB: 081C**

位	默认	类型	信号
15:0	0	HIST	stat_rx_total_bytes_count[47:32]

**STAT\_RX\_TOTAL\_GOOD\_BYTES\_LSB: 0820**

表 209: STAT\_RX\_TOTAL\_GOOD\_BYTES\_LSB: 0820

位	默认	类型	信号
31:0	0	HIST	stat_rx_total_good_bytes_count[31:0]

**STAT\_RX\_TOTAL\_GOOD\_BYTES\_MSB: 0824**

表 210: STAT\_RX\_TOTAL\_GOOD\_BYTES\_MSB: 0824

位	默认	类型	信号
15:0	0	HIST	stat_rx_total_good_bytes_count[47:32]

**STAT\_RX\_PACKET\_64\_BYTES\_LSB: 0828**

表 211: STAT\_RX\_PACKET\_64\_BYTES\_LSB: 0828

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_64_bytes_count[31:0]

**STAT\_RX\_PACKET\_64\_BYTES\_MSB: 082C**

表 212: STAT\_RX\_PACKET\_64\_BYTES\_MSB: 082C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_64_bytes_count[47:32]

**STAT\_RX\_PACKET\_65\_127\_BYTES\_LSB: 0830**

表 213: STAT\_RX\_PACKET\_65\_127\_BYTES\_LSB: 0830

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_65_127_bytes_count[31:0]

**STAT\_RX\_PACKET\_65\_127\_BYTES\_MSB: 0834**

表 214: STAT\_RX\_PACKET\_65\_127\_BYTES\_MSB: 0834

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_65_127_bytes_count[47:32]

**STAT\_RX\_PACKET\_128\_255\_BYTES\_LSB: 0838**

表 215: STAT\_RX\_PACKET\_128\_255\_BYTES\_LSB: 0838

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_128_255_bytes_count[31:0]

**STAT\_RX\_PACKET\_128\_255\_BYTES\_MSB: 083C**

表 216: STAT\_RX\_PACKET\_128\_255\_BYTES\_MSB: 083C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_128_255_bytes_count[47:32]

**STAT\_RX\_PACKET\_256\_511\_BYTES\_LSB: 0840**

表 217: STAT\_RX\_PACKET\_256\_511\_BYTES\_LSB: 0840

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_256_511_bytes_count[31:0]

**STAT\_RX\_PACKET\_256\_511\_BYTES\_MSB: 0844**

表 218: STAT\_RX\_PACKET\_256\_511\_BYTES\_MSB: 0844

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_256_511_bytes_count[47:32]

**STAT\_RX\_PACKET\_512\_1023\_BYTES\_LSB: 0848**

表 219: STAT\_RX\_PACKET\_512\_1023\_BYTES\_LSB: 0848

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_512_1023_bytes_count[31:0]

**STAT\_RX\_PACKET\_512\_1023\_BYTES\_MSB: 084C**

表 220: STAT\_RX\_PACKET\_512\_1023\_BYTES\_MSB: 084C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_512_1023_bytes_count[47:32]

**STAT\_RX\_PACKET\_1024\_1518\_BYTES\_LSB: 0850**

表 221: STAT\_RX\_PACKET\_1024\_1518\_BYTES\_LSB: 0850

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_1024_1518_bytes_count[31:0]

**STAT\_RX\_PACKET\_1024\_1518\_BYTES\_MSB: 0854**

表 222: STAT\_RX\_PACKET\_1024\_1518\_BYTES\_MSB: 0854

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_1024_1518_bytes_count[47:32]

**STAT\_RX\_PACKET\_1519\_1522\_BYTES\_LSB: 0858**

表 223: STAT\_RX\_PACKET\_1519\_1522\_BYTES\_LSB: 0858

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_1519_1522_bytes_count[31:0]

**STAT\_RX\_PACKET\_1519\_1522\_BYTES\_MSB: 085C**

表 224: STAT\_RX\_PACKET\_1519\_1522\_BYTES\_MSB: 085C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_1519_1522_bytes_count[47:32]

**STAT\_RX\_PACKET\_1523\_1548\_BYTES\_LSB: 0860**

表 225: STAT\_RX\_PACKET\_1523\_1548\_BYTES\_LSB: 0860

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_1523_1548_bytes_count[31:0]

**STAT\_RX\_PACKET\_1523\_1548\_BYTES\_MSB: 0864**

表 226: STAT\_RX\_PACKET\_1523\_1548\_BYTES\_MSB: 0864

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_1523_1548_bytes_count[47:32]

**STAT\_RX\_PACKET\_1549\_2047\_BYTES\_LSB: 0868**

表 227: STAT\_RX\_PACKET\_1549\_2047\_BYTES\_LSB: 0868

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_1549_2047_bytes_count[31:0]

**STAT\_RX\_PACKET\_1549\_2047\_BYTES\_MSB: 086C**

表 228: STAT\_RX\_PACKET\_1549\_2047\_BYTES\_MSB: 086C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_1549_2047_bytes_count[47:32]

**STAT\_RX\_PACKET\_2048\_4095\_BYTES\_LSB: 0870**

表 229: STAT\_RX\_PACKET\_2048\_4095\_BYTES\_LSB: 0870

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_2048_4095_bytes_count[31:0]

**STAT\_RX\_PACKET\_2048\_4095\_BYTES\_MSB: 0874**

表 230: STAT\_RX\_PACKET\_2048\_4095\_BYTES\_MSB: 0874

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_2048_4095_bytes_count[47:32]

**STAT\_RX\_PACKET\_4096\_8191\_BYTES\_LSB: 0878**

表 231: STAT\_RX\_PACKET\_4096\_8191\_BYTES\_LSB: 0878

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_4096_8191_bytes_count[31:0]

**STAT\_RX\_PACKET\_4096\_8191\_BYTES\_MSB: 087C**

表 232: STAT\_RX\_PACKET\_4096\_8191\_BYTES\_MSB: 087C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_4096_8191_bytes_count[47:32]

**STAT\_RX\_PACKET\_8192\_9215\_BYTES\_LSB: 0880**

表 233: STAT\_RX\_PACKET\_8192\_9215\_BYTES\_LSB: 0880

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_8192_9215_bytes_count[31:0]

**STAT\_RX\_PACKET\_8192\_9215\_BYTES\_MSB: 0884**

表 234: STAT\_RX\_PACKET\_8192\_9215\_BYTES\_MSB: 0884

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_8192_9215_bytes_count[47:32]

**STAT\_RX\_PACKET\_LARGE\_LSB: 0888**

表 235: STAT\_RX\_PACKET\_LARGE\_LSB: 0888

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_large_count[31:0]

**STAT\_RX\_PACKET\_LARGE\_MSB: 088C**

表 236: STAT\_RX\_PACKET\_LARGE\_MSB: 088C

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_large_count[47:32]

**STAT\_RX\_PACKET\_SMALL\_LSB: 0890**

表 237: STAT\_RX\_PACKET\_SMALL\_LSB: 0890

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_small_count[31:0]

**STAT\_RX\_PACKET\_SMALL\_MSB: 0894**

表 238: STAT\_RX\_PACKET\_SMALL\_MSB: 0894

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_small_count[47:32]

**STAT\_RX\_UNDERSIZE\_LSB: 0898**

表 239: STAT\_RX\_UNDERSIZE\_LSB: 0898

位	默认	类型	信号
31:0	0	HIST	stat_rx_undersize_count[31:0]

**STAT\_RX\_UNDERSIZE\_MSB: 089C**

表 240: STAT\_RX\_UNDERSIZE\_MSB: 089C

位	默认	类型	信号
15:0	0	HIST	stat_rx_undersize_count[47:32]

**STAT\_RX\_FRAGMENT\_LSB: 08A0**

表 241: STAT\_RX\_FRAGMENT\_LSB: 08A0

位	默认	类型	信号
31:0	0	HIST	stat_rx_fragment_count[31:0]

**STAT\_RX\_FRAGMENT\_MSB: 08A4**

表 242: STAT\_RX\_FRAGMENT\_MSB: 08A4

位	默认	类型	信号
15:0	0	HIST	stat_rx_fragment_count[47:32]

**STAT\_RX\_OVERSIZE\_LSB: 08A8**

表 243: STAT\_RX\_OVERSIZE\_LSB: 08A8

位	默认	类型	信号
31:0	0	HIST	stat_rx_oversize_count[31:0]

**STAT\_RX\_OVERSIZE\_MSB: 08AC**

表 244: STAT\_RX\_OVERSIZE\_MSB: 08AC

位	默认	类型	信号
15:0	0	HIST	stat_rx_oversize_count[47:32]

**STAT\_RX\_TOOLONG\_LSB: 08B0**

表 245: STAT\_RX\_TOOLONG\_LSB: 08B0

位	默认	类型	信号
31:0	0	HIST	stat_rx_toolong_count[31:0]

**STAT\_RX\_TOOLONG\_MSB: 08B4**

表 246: STAT\_RX\_TOOLONG\_MSB: 08B4

位	默认	类型	信号
15:0	0	HIST	stat_rx_toolong_count[47:32]

**STAT\_RX\_JABBER\_LSB: 08B8**

表 247: STAT\_RX\_JABBER\_LSB: 08B8

位	默认	类型	信号
31:0	0	HIST	stat_rx_jabber_count[31:0]

**STAT\_RX\_JABBER\_MSB: 08BC**

表 248: STAT\_RX\_JABBER\_MSB: 08BC

位	默认	类型	信号
15:0	0	HIST	stat_rx_jabber_count[47:32]

**STAT\_RX\_BAD\_FCS\_LSB: 08C0**

表 249: STAT\_RX\_BAD\_FCS\_LSB: 08C0

位	默认	类型	信号
31:0	0	HIST	stat_rx_bad_fcs_count[31:0]

**STAT\_RX\_BAD\_FCS\_MSB: 08C4**

表 250: STAT\_RX\_BAD\_FCS\_MSB: 08C4

位	默认	类型	信号
15:0	0	HIST	stat_rx_bad_fcs_count[47:32]

**STAT\_RX\_PACKET\_BAD\_FCS\_LSB: 08C8**

表 251: STAT\_RX\_PACKET\_BAD\_FCS\_LSB: 08C8

位	默认	类型	信号
31:0	0	HIST	stat_rx_packet_bad_fcs_count[31:0]

**STAT\_RX\_PACKET\_BAD\_FCS\_MSB: 08CC**

表 252: STAT\_RX\_PACKET\_BAD\_FCS\_MSB: 08CC

位	默认	类型	信号
15:0	0	HIST	stat_rx_packet_bad_fcs_count[47:32]

**STAT\_RX\_STOMPED\_FCS\_LSB: 08D0**

表 253: STAT\_RX\_STOMPED\_FCS\_LSB: 08D0

位	默认	类型	信号
31:0	0	HIST	stat_rx_stomped_fcs_count[31:0]

**STAT\_RX\_STOMPED\_FCS\_MSB: 08D4**

表 254: STAT\_RX\_STOMPED\_FCS\_MSB: 08D4

位	默认	类型	信号
15:0	0	HIST	stat_rx_stomped_fcs_count[47:32]

**STAT\_RX\_UNICAST\_LSB: 08D8**

表 255: STAT\_RX\_UNICAST\_LSB: 08D8

位	默认	类型	信号
31:0	0	HIST	stat_rx_unicast_count[31:0]

**STAT\_RX\_UNICAST\_MSB: 08DC**

表 256: STAT\_RX\_UNICAST\_MSB: 08DC

位	默认	类型	信号
15:0	0	HIST	stat_rx_unicast_count[47:32]

**STAT\_RX\_MULTICAST\_LSB: 08E0**

表 257: STAT\_RX\_MULTICAST\_LSB: 08E0

位	默认	类型	信号
31:0	0	HIST	stat_rx_multicast_count[31:0]

**STAT\_RX\_MULTICAST\_MSB: 08E4**

表 258: STAT\_RX\_MULTICAST\_MSB: 08E4

位	默认	类型	信号
15:0	0	HIST	stat_rx_multicast_count[47:32]

**STAT\_RX\_BROADCAST\_LSB: 08E8**

表 259: STAT\_RX\_BROADCAST\_LSB: 08E8

位	默认	类型	信号
31:0	0	HIST	stat_rx_broadcast_count[31:0]

**STAT\_RX\_BROADCAST\_MSB: 08EC**

表 260: STAT\_RX\_BROADCAST\_MSB: 08EC

位	默认	类型	信号
15:0	0	HIST	stat_rx_broadcast_count[47:32]

**STAT\_RX\_VLAN\_LSB: 08F0**

表 261: STAT\_RX\_VLAN\_LSB: 08F0

位	默认	类型	信号
31:0	0	HIST	stat_rx_vlan_count[31:0]

**STAT\_RX\_VLAN\_MSB: 08F4**

表 262: STAT\_RX\_VLAN\_MSB: 08F4

位	默认	类型	信号
15:0	0	HIST	stat_rx_vlan_count[47:32]

**STAT\_RX\_PAUSE\_LSB: 08F8**

表 263: STAT\_RX\_PAUSE\_LSB: 08F8

位	默认	类型	信号
31:0	0	HIST	stat_rx_pause_count[31:0]

**STAT\_RX\_PAUSE\_MSB: 08FC**

表 264: STAT\_RX\_PAUSE\_MSB: 08FC

位	默认	类型	信号
15:0	0	HIST	stat_rx_pause_count[47:32]

**STAT\_RX\_USER\_PAUSE\_LSB: 0900**

表 265: STAT\_RX\_USER\_PAUSE\_LSB: 0900

位	默认	类型	信号
31:0	0	HIST	stat_rx_user_pause_count[31:0]

**STAT\_RX\_USER\_PAUSE\_MSB: 0904**

表 266: STAT\_RX\_USER\_PAUSE\_MSB: 0904

位	默认	类型	信号
15:0	0	HIST	stat_rx_user_pause_count[47:32]

**STAT\_RX\_INRANGEERR\_LSB: 0908**

表 267: STAT\_RX\_INRANGEERR\_LSB: 0908

位	默认	类型	信号
31:0	0	HIST	stat_rx_inrangeerr_count[31:0]

**STAT\_RX\_INRANGEERR\_MSB: 090C**

表 268: STAT\_RX\_INRANGEERR\_MSB: 090C

位	默认	类型	信号
15:0	0	HIST	stat_rx_inrangeerr_count[47:32]

**STAT\_RX\_TRUNCATED\_LSB: 0910**

表 269: STAT\_RX\_TRUNCATED\_LSB: 0910

位	默认	类型	信号
31:0	0	HIST	stat_rx_truncated_count[31:0]

**STAT\_RX\_TRUNCATED\_MSB: 0914**

表 270: STAT\_RX\_TRUNCATED\_MSB: 0914

位	默认	类型	信号
15:0	0	HIST	stat_rx_truncated_count[47:32]

**STAT\_RX\_TEST\_PATTERN\_MISMATCH\_LSB: 0918**

表 271: STAT\_RX\_TEST\_PATTERN\_MISMATCH\_LSB: 0918

位	默认	类型	信号
31:0	0	HIST	stat_rx_test_pattern_mismatch_count[31:0]

**STAT\_RX\_TEST\_PATTERN\_MISMATCH\_MSB: 091C**

表 272: STAT\_RX\_TEST\_PATTERN\_MISMATCH\_MSB: 091C

位	默认	类型	信号
15:0	0	HIST	stat_rx_test_pattern_mismatch_count[47:32]

**STAT\_FEC\_INC\_CORRECT\_COUNT\_LSB: 0920**

表 273: STAT\_FEC\_INC\_CORRECT\_COUNT\_LSB: 0920

位	默认	类型	信号
31:0	0	HIST	stat_fec_inc_correct_count_count[31:0]

**STAT\_FEC\_INC\_CORRECT\_COUNT\_MSB: 0924**

表 274: STAT\_FEC\_INC\_CORRECT\_COUNT\_MSB: 0924

位	默认	类型	信号
15:0	0	HIST	stat_fec_inc_correct_count_count[47:32]

**STAT\_FEC\_INC\_CANT\_CORRECT\_COUNT\_LSB: 0928**

表 275: STAT\_FEC\_INC\_CANT\_CORRECT\_COUNT\_LSB: 0928

位	默认	类型	信号
31:0	0	HIST	stat_fec_inc_cant_correct_count_count[31:0]

**STAT\_FEC\_INC\_CANT\_CORRECT\_COUNT\_MSB: 092C**

表 276: STAT\_FEC\_INC\_CANT\_CORRECT\_COUNT\_MSB: 092C

位	默认	类型	信号
15:0	0	HIST	stat_fec_inc_cant_correct_count_count[47:32]

**STAT\_TX\_MM\_STATUS\_LSB: 0x0980**

表 277: STAT\_TX\_MM\_STATUS\_LSB: 0x0980

位	默认	类型	信号
31:0	0	HIST	stat_tx_mm_status[31:0]

**STAT\_TX\_MM\_STATUS\_MSB: 0x0984**

表 278: STAT\_TX\_MM\_STATUS\_MSB: 0x0984

位	默认	类型	信号
15:0	0	HIST	stat_tx_mm_status[47:32]

**STAT\_TX\_MM\_STATUS\_LSB: 0x0988**

表 279: STAT\_TX\_MM\_STATUS\_LSB: 0x0988

位	默认	类型	信号
31:0	0	HIST	stat_tx_mm_fragment[31:0]

**STAT\_TX\_MM\_FRAGMENT\_MSB: 0x098C**

表 280: STAT\_TX\_MM\_FRAGMENT\_MSB: 0x098C

位	默认	类型	信号
15:0	0	HIST	stat_tx_mm_fragment[47:32]

**STAT\_TX\_MM\_HOLD\_LSB: 0x0990**

表 281: STAT\_TX\_MM\_HOLD\_LSB: 0x0990

位	默认	类型	信号
31:0	0	HIST	stat_tx_mm_hold[31:0]

**STAT\_TX\_MM\_HOLD\_MSB: 0x0994**

表 282: STAT\_TX\_MM\_HOLD\_MSB: 0x0994

位	默认	类型	信号
15:0	0	HIST	stat_tx_mm_hold[47:32]

**STAT\_RX\_MM\_ASSEMBLY\_ERROR\_LSB: 0x0998**

表 283: STAT\_RX\_MM\_ASSEMBLY\_ERROR\_LSB: 0x0998

位	默认	类型	信号
31:0	0	HIST	stat_rx_mm_assembly_error[31:0]

**STAT\_RX\_MM\_ASSEMBLY\_ERROR\_MSB: 0x099C**

表 284: STAT\_RX\_MM\_ASSEMBLY\_ERROR\_MSB: 0x099C

位	默认	类型	信号
15:0	0	HIST	stat_rx_mm_assembly_error[47:32]

**STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_LSB: 0x09A0**

表 285: STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_LSB: 0x09A0

位	默认	类型	信号
31:0	0	HIST	stat_rx_mm_frame_smd_error[31:0]

**STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_MSB: 0x09A4**

表 286: STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_MSB: 0x09A4

位	默认	类型	信号
15:0	0	HIST	stat_rx_mm_frame_smd_error[47:32]

**STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_LSB: 0x09A8**

表 287: STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_LSB: 0x09A8

位	默认	类型	信号
31:0	0	HIST	stat_rx_mm_frame_assembly_ok[31:0]

**STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_MSB: 0x09AC**

表 288: STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_MSB: 0x09AC

位	默认	类型	信号
15:0	0	HIST	stat_rx_mm_frame_assembly_ok[47:32]

**STAT\_RX\_MM\_FRAGMENT\_LSB: 0x09B0**

表 289: STAT\_RX\_MM\_FRAGMENT\_LSB: 0x09B0

位	默认	类型	信号
31:0	0	HIST	stat_rx_mm_fragment[31:0]

**STAT\_RX\_MM\_FRAGMENT\_MSB: 0x09B4**

表 290: STAT\_RX\_MM\_FRAGMENT\_MSB: 0x09B4

位	默认	类型	信号
15:0	0	HIST	stat_rx_mm_fragment[47:32]

# 利用子系统进行设计

本章包含有关利用子系统来协助简化设计的指导信息和其它信息。

## 时钟设置

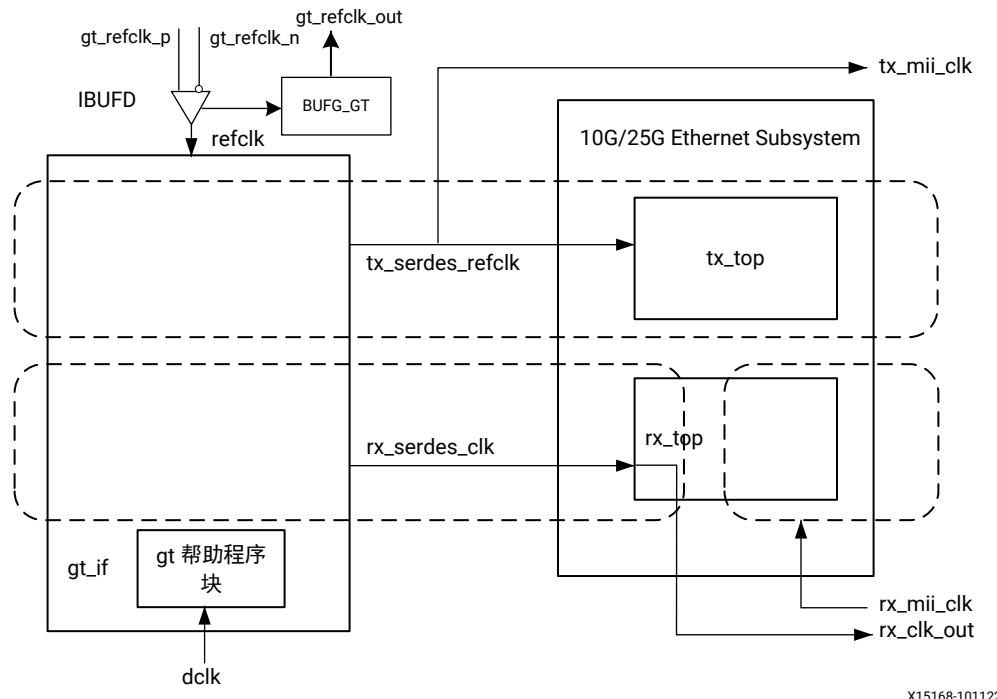
本节描述了组件支持封装文件层级上的所有 10G/25G 配置的时钟设置。其中包含 7 种完全不同的时钟架构（按功能和选项区分）：

- PCS/PMA 64 位时钟设置
- PCS/PMA 32 位时钟设置
- 含 PCS/PMA 时钟设置的 10G/25G MAC
- 10G 仅限 MAC 的时钟设置
- 含 PCS/PMA 时钟设置的低时延 10G/25G MAC
- 低时延 32 位 10 Gb/s MAC（含 PCS）
- 自动协商和链路训练时钟

### PCS/PMA 64 位时钟设置

下图展示了 10G/25G PCS 的时钟架构。数据路径中有 3 个时钟域，如下图中虚线所示。

图 15：PCS/PMA 时钟设置



- refclk\_p0、refclk\_n0 和 tx\_serdes\_refclk: refclk 差分对是必需的，用作为 FPGA 的输入。设计示例包含缓冲器，用于将该时钟转换为单端信号 refclk，此信号用作为 GT 块的参考时钟。tx\_serdes\_refclk 直接衍生自 refclk。请注意，必须选择 refclk，以确保 tx\_mii\_clk 满足 802.3 的要求，即范围应分别在 390.625 MHz 的 100 ppm 内（针对 25G）和 156.25 MHz 的 100 ppm 内（针对 10G）。
- tx\_mii\_clk: tx\_mii\_clk 是输出，与 tx\_serdes\_refclk 相同。整个 TX 路径由此时钟驱动。您必须将 TX 路径 mii 总线同步到此时钟输出。所有 TX 控制信号和状态信号均以此时钟为参考。
- rx\_serdes\_clk: rx\_serdes\_clk 衍生自 GT 块中的输入数据串流。输入数据串流由此时钟域中的 RX 核进行处理。
- rx\_clk\_out: 提供的 rx\_clk\_out 输出信号可供 RX 核所处理的 RX 控制信号和状态信号作为参考。其频率与 rx\_serdes\_clk 相同。
- rx\_mii\_clk: rx\_mii\_clk 输入必须同步到 RX XGMII/25GMII 数据总线。此时钟和 RX XGMII/25GMII 总线必须在所需频率（即 390.625 MHz (25G) 和 156.25 MHz (10G)）的 100 ppm 内。
- dclk: dclk 信号必须为稳定且便于使用的时钟。它用作为 GT 帮助程序块（用于启动 GT 本身）的参考频率。在设计示例中，典型值为 75 MHz，该值直接衍生自 VCU107 评估板上可用的 300 MHz 时钟。请注意，GT 帮助程序块必须已知实际频率才能正常运行。

## PCS/PMA 32 位时钟设置

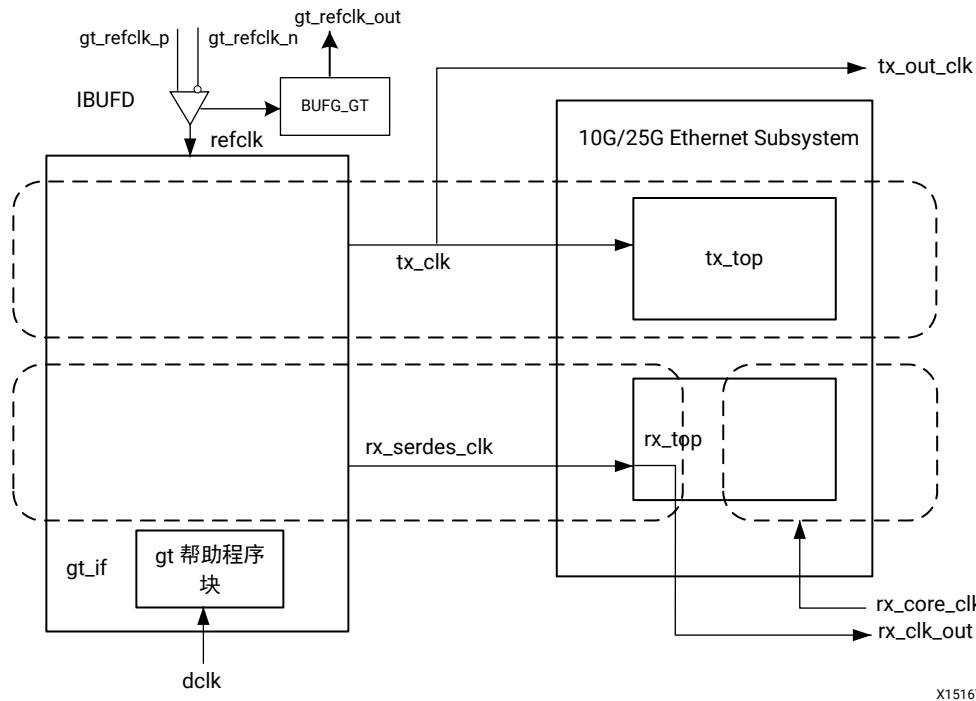
- refclk\_p0、refclk\_n0 和 rx\_serdes\_refclk: refclk 差分对是必需的，用作为 FPGA 的输入。设计示例包含缓冲器，用于将该时钟转换为单端信号 refclk，此信号用作为 GT 块的参考时钟。请注意，必须选择 refclk，以确保 tx\_clk\_out 满足 IEEE 802.3 的要求，即范围应在 312.5 MHz 的 100 ppm 内（针对 10G）。
- tx\_clk\_out: tx\_clk\_out 是输出，您必须将 TX 路径 mii 总线同步到此时钟输出。所有 TX 控制信号和状态信号均以此时钟为参考。

- rx\_serdes\_clk: rx\_serdes\_clk 衍生自 GT 块中的输入数据串流。输入数据串流由此时钟域中的 RX 核进行处理。
- rx\_clk\_out: 提供的 rx\_clk\_out 输出信号可供 RX 核所处理的 RX 控制信号和状态信号作为参考。其频率与 rx\_serdes\_clk 相同。
- dclk: dclk 信号必须为稳定且便于使用的时钟。它用作为 GT 帮助程序块（用于启动 GT 本身）的参考频率。在设计示例中，典型值为 100 MHz，该值直接衍生自 VCU107 评估板上可用的 300 MHz 时钟。请注意，GT 帮助程序块必须已知实际频率才能正常运行。

## 含 PCS/PMA 时钟设置的 10G/25G MAC

下图展示了含 PCS/PMA 时钟设置的 10/25G MAC 的时钟架构。此版本的子系统在 RX 中包含 FIFO。数据路径中有 3 个时钟域，如下图中虚线所示。

图 16: 含 PCS/PMA 时钟设置的 10G/25G MAC



- refclk\_p0、refclk\_n0 和 tx\_serdes\_refclk: refclk 差分对是必需的，用作为 FPGA 的输入。设计示例包含缓冲器，用于将该时钟转换为单端信号 refclk，此信号用作为 GT 块的参考时钟。tx\_serdes\_refclk 直接衍生自 refclk。请注意，必须选择 refclk，以确保 tx\_serdes\_refclk 满足 802.3 的要求，即范围应分别在 390.625 MHz 的 100 ppm 内（针对 25G）、156.25 MHz 的 100 ppm 内（针对 64 位 10G）和 312.5 MHz 的 100 ppm 内（针对 32 位 10G）。
- tx\_clk\_out: 此时钟用作为时钟设置数据输入 TX AXI4-Stream 接口，它还用作为 TX 控制信号和状态信号的参考时钟。其频率与 tx\_serdes\_refclk 相同。
- rx\_clk\_out: 提供的 rx\_clk\_out 输出信号可供 RX 核所处理的 RX 控制信号和状态信号作为参考。其频率与 rx\_serdes\_clk 相同。

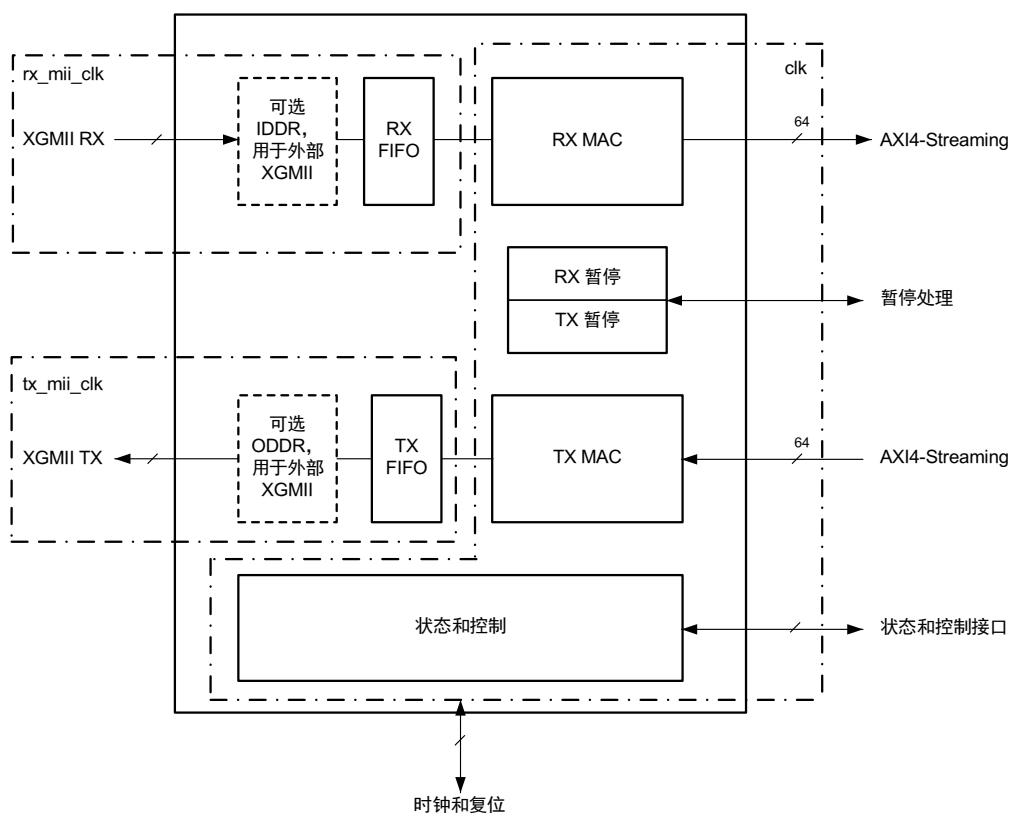
- rx\_clk: rx\_clk 可用，因为 rx\_core\_clk 是 RX 核的输入时钟。您必须从设计示例驱动此时钟。应以与 tx\_clk 相同的频率来驱动 rx\_core\_clk。启用 FIFO 时，系统端数据路径的首选运行模式是将 tx\_clk\_out 连接到 rx\_core\_clk。通过这种方式进行连接时，RX AXI4-Stream 接口和 TX AXI4-Stream 接口都位于同一时钟域中。禁用 FIFO 时，rx\_core\_clk 必须由 rx\_clk\_out/rx\_serdes\_clk 来驱动。
- dclk: dclk 信号必须为稳定且便于使用的时钟。它用作为 GT 帮助程序块（用于启动 GT 本身）的参考频率。在设计示例中，典型值为 75 MHz，该值直接衍生自 VCU107 评估板上可用的 300 MHz 时钟。

注释：GT 帮助程序块必须已知实际频率才能正常运行。

## 10G 仅限 MAC 的时钟设置

10G 仅限 MAC 的时钟架构如 [时钟设置](#) 所示。其中有 3 个时钟域，如虚线部分所示。

图 17：10G 仅限 MAC 配置的时钟架构



X20135-101122

- rx\_mii\_clk: rx\_mii\_clk 可在内部驱动，也可在外部驱动。必须选择该时钟以满足 IEEE 802.3 要求的 156.25 MHz  $\pm$  100 ppm（针对 10 Gb/s 运行）。
- tx\_mii\_clk: tx\_mii\_clk 可在内部驱动，也可在外部驱动。必须选择该时钟以满足 IEEE 802.3 要求的 156.25 MHz  $\pm$  100 ppm（针对 10 Gb/s 运行）。
- clk: clk 时钟用于驱动所有内部 RX 和 TX 核逻辑，包括 AXI4-Stream 接口以及控制信号和状态信号。clk 时钟运行频率应大于或等于 156.25 MHz。

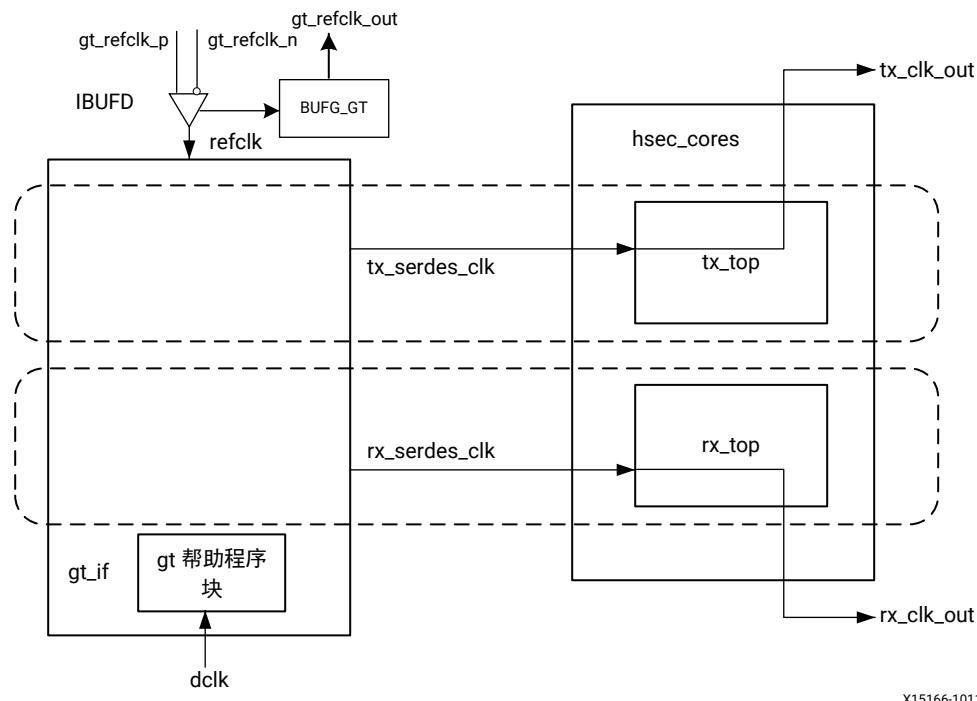
## 低时延 32 位 10 Gb/s MAC（含 PCS）

此配置的时钟架构与其对应 64 位配置的时钟架构的唯一区别在于 AXI4-Stream 接口的时钟现在是 312.5 MHz。请参阅有关 [时钟设置](#) 的章节，以获取有关时钟架构的更多详细信息。

## 含 PCS/PMA 时钟设置的低时延 10G/25G MAC

含 PCS/PMA 时钟设置的低时延 10G/25G MAC 的时钟架构如下图所示。低时延是通过省略 RX FIFO 来实现的，这导致时钟排布产生变化。在数据路径中有 2 个时钟域，如下图虚线所示。

图 18：含 PCS/PMA 时钟设置的低时延 10G/25G MAC



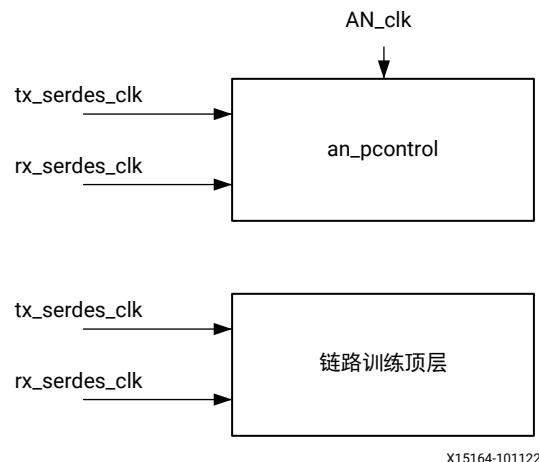
X15166-101122

- **refclk\_p0、refclk\_n0 和 tx\_serdes\_refclk**: **refclk** 差分对是必需的，用作为 FPGA 的输入。设计示例包含缓冲器，用于将该时钟转换为单端信号 **refclk**，此信号用作为 GT 块的参考时钟。**tx\_serdes\_refclk** 直接衍生自 **refclk**。请注意，必须选择 **refclk** 以确保 **tx\_serdes\_refclk** 满足 802.3 的要求，即范围应分别在 390.625 MHz 的 100 ppm 内（针对 25G）和 156.25 MHz 的 100 ppm 内（针对 10G）。
- **tx\_clk\_out**: 此时钟用作为时钟设置数据输入 TX AXI4-Stream 接口，它还用作为 TX 控制信号和状态信号的参考时钟。其频率与 **tx\_serdes\_refclk** 相同。由于不存在 TX FIFO，因此您必须立即响应 **tx\_axis\_tready** 信号。
- **rx\_clk\_out**: 提供的 **rx\_clk\_out** 输出信号可供 RX 核所处理的 RX 控制信号和状态信号作为参考。其频率与 **rx\_serdes\_clk** 相同。由于不存在 RX FIFO，因此该时钟还用于驱动 RX AXI4-Stream 接口。按此排布方式，**rx\_clk\_out** 与 **tx\_clk\_out** 的频率不同，且彼此间不存在定义的相位关系。
- **dclk**: **dclk** 信号必须为稳定且便于使用的时钟。它用作为 GT 帮助程序块（用于启动 GT 本身）的参考频率。在设计示例中，典型值为 75 MHz，该值直接衍生自 VCU107 评估板上可用的 300 MHz 时钟。请注意，GT 帮助程序块必须已知实际频率才能正常运行。

## 自动协商和链路训练时钟

下图显示了自动协商和链路训练块的时钟架构。请注意，除非选择了 BASE-KR 特性，否则不包含这些块。自动协商和链路训练块与 MAC 和 PCS 分开运作，因此处于不同的时钟域。

图 19：自动协商和链路训练时钟



- `rx_serdes_clk`: `rx_serdes_clk` 驱动用于自动协商和链路训练的 RX 线一侧的逻辑。
- `tx_serdes_clk`: `tx_serdes_clk` 驱动用于自动协商和链路训练的 TX 线一侧的逻辑。DME 帧在此时钟域上生成。
- `AN_clk`: `AN_clk` 驱动自动协商状态机。所有功能信号都位于此时钟域。`AN_clk` 可以是任何便于使用的频率。在设计示例中，`AN_clk` 连接到 `dclk` 输入，其典型频率为 75 MHz。自动协商状态机必须已知 `AN_clk` 频率，因为这是所有定时器的参考频率。

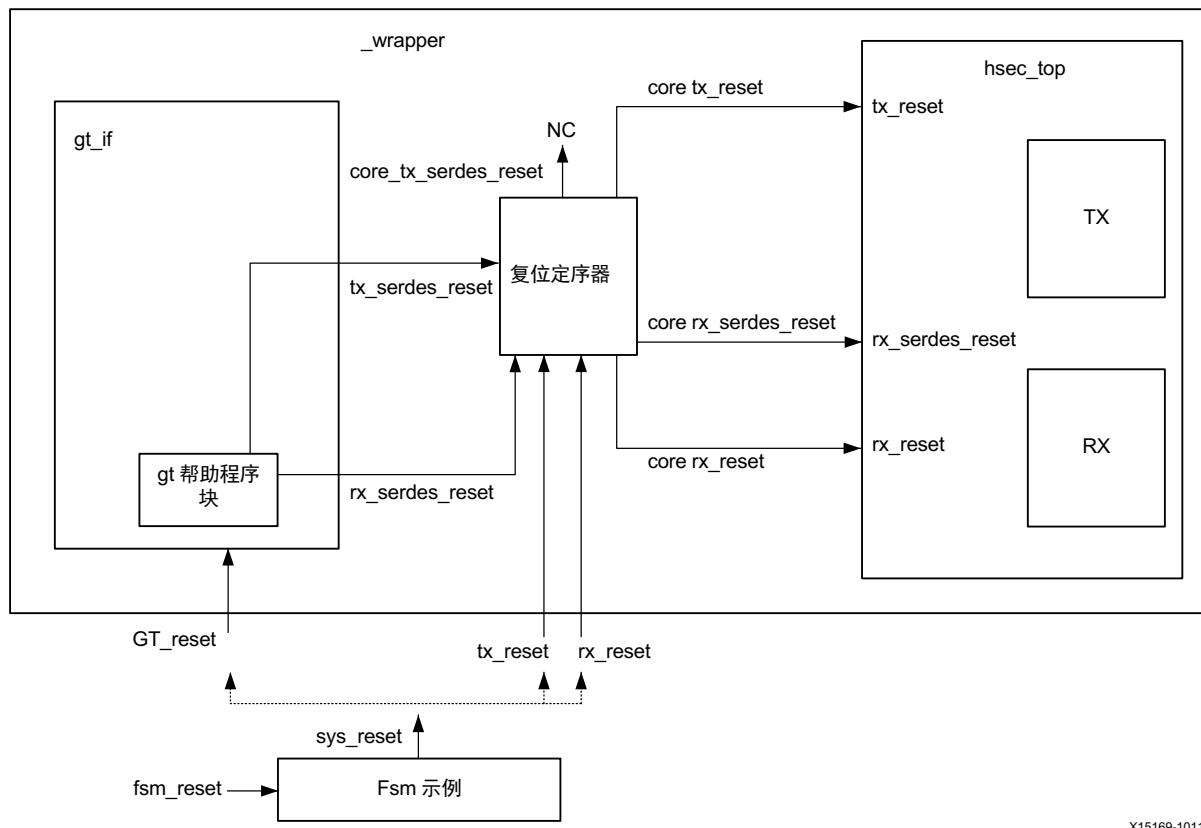
Vivado 集成设计环境 (IDE) 会将此时钟设置为与 GUI 中输入的 GT DRP CLK 设置相同的频率。

---

## 复位

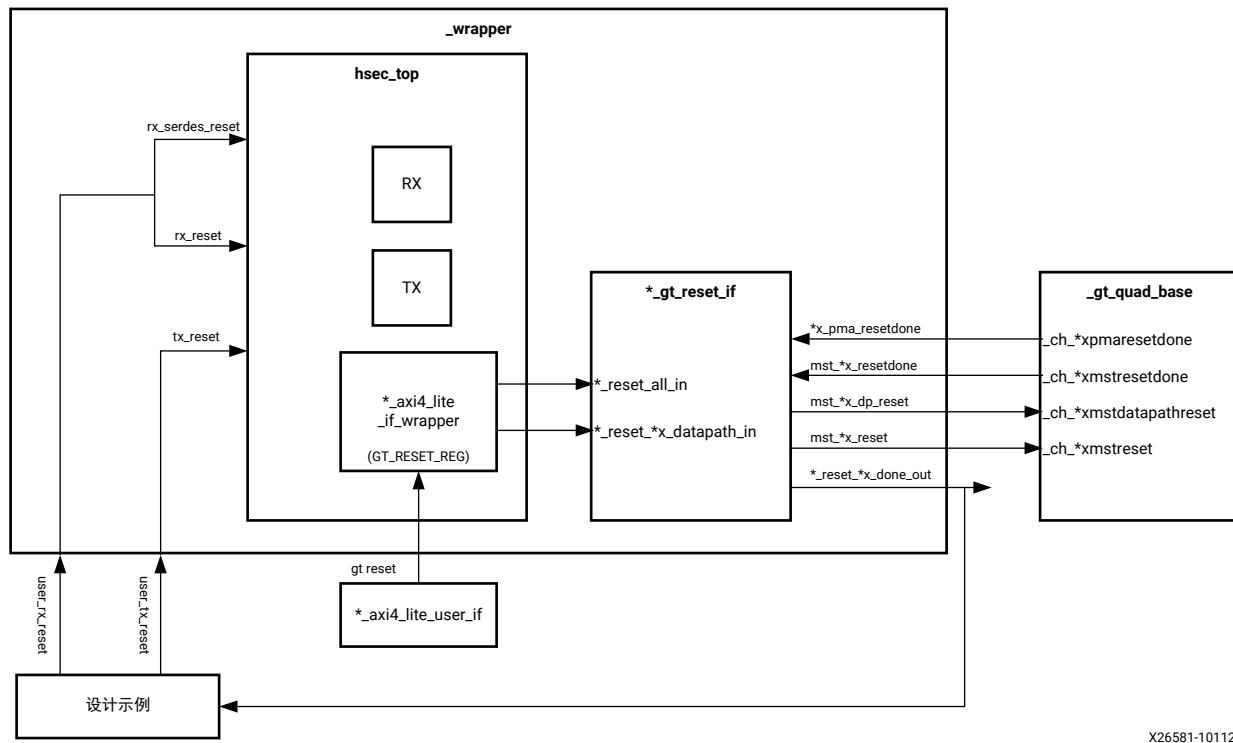
下图显示了在组件支持封装文件层级上为非 Versal 器件实现的 10G/25G Ethernet Subsystem MAC（含 PCS/PMA）的复位结构。澄清：未显示时钟。

图 20：复位结构 (UltraScale/UltraScale+)



对于 Versal 器件，下图显示了在组件支持封装文件层级上为 10G/25G Ethernet Subsystem MAC（含 PCS/PMA）实现的复位结构。

图 21：复位结构 (Versal ACAP)



## 组件支持层级复位

在设计示例中，使用单一复位来对整个封装器层级进行复位。通过使用外部激励 `fsm_reset`，`example_fsm` 块会发出 `sys_reset` 信号，此信号连接至 3 个 `_wrapper` 复位。因此，设计示例显示全部 3 个封装器复位均可同时释放，并且随后执行纠正操作。

## 封装器复位

层级的 `_wrapper` 层假定为您在自己的设计中例化的对象。有 3 个复位需要处理，如下所示：

- GT\_reset
- tx\_reset
- rx\_reset

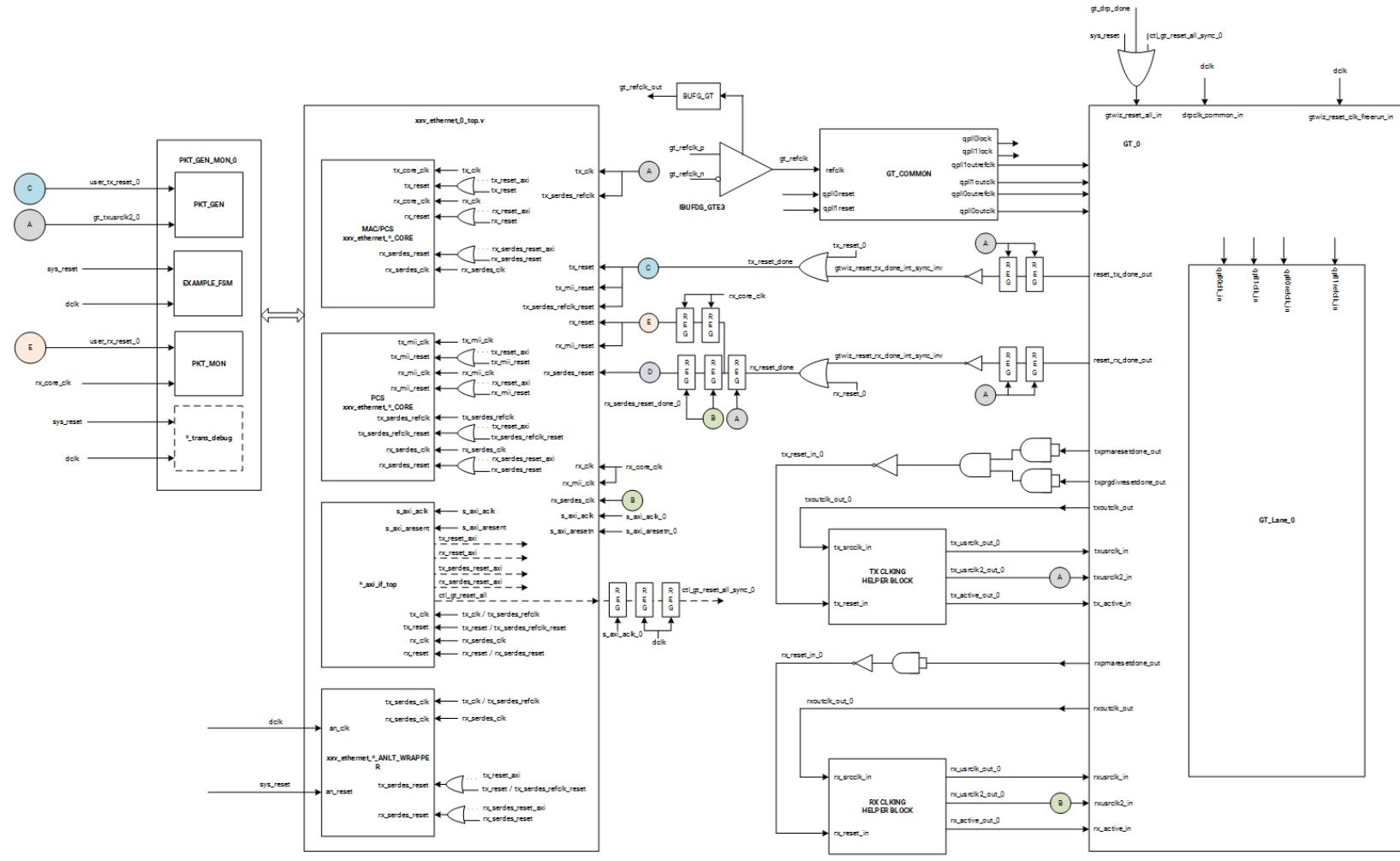
复位信号的时序由复位定序器块来处理。

- GT\_reset：`GT_reset` 是异步高电平有效复位输入，连接到 GT。GT 的内部复位由 GT 帮助程序块来处理。
- tx\_reset：`tx_reset` 是异步高电平有效复位，适用于 10G/25G Ethernet Subsystem 的 TX 路径逻辑。虽然此复位连接到设计示例中的 GT 复位，但它可随时断言有效以单独复位 TX 路径，而不会干扰 RX 路径。
- rx\_reset：`rx_reset` 是异步高电平有效复位，适用于 10G/25G Ethernet Subsystem 的 RX 路径逻辑。虽然此复位连接到设计示例中的 GT 复位，但它可随时断言有效以单独复位 RX 路径，而不会干扰 TX 路径。

## LogiCORE 设计示例时钟设置和复位

## 单核 - 异步时钟模式的详细图示 (UltraScale/UltraScale+)

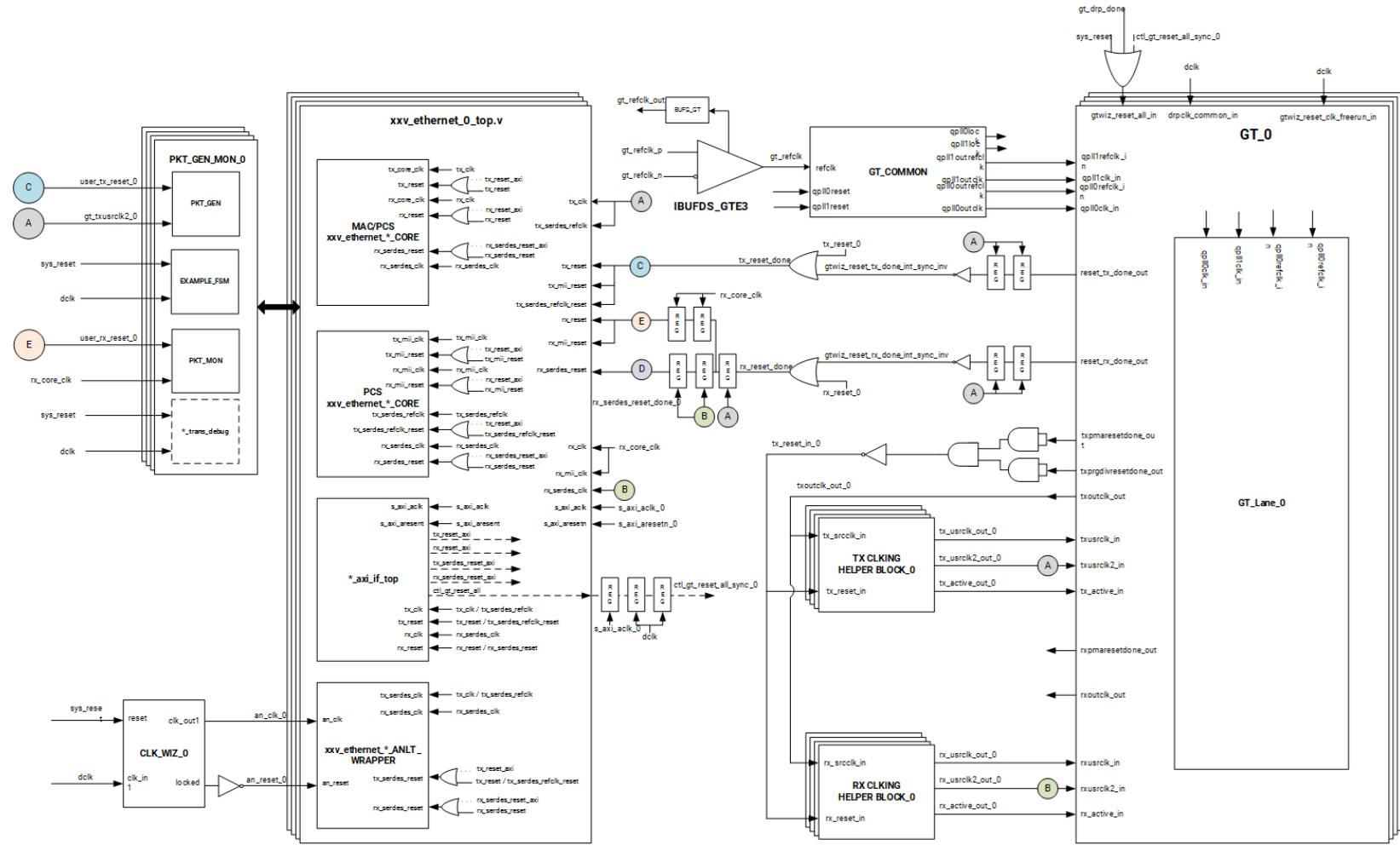
图 22：单核 - 异步时钟模式的详细图示



XILINX-111111

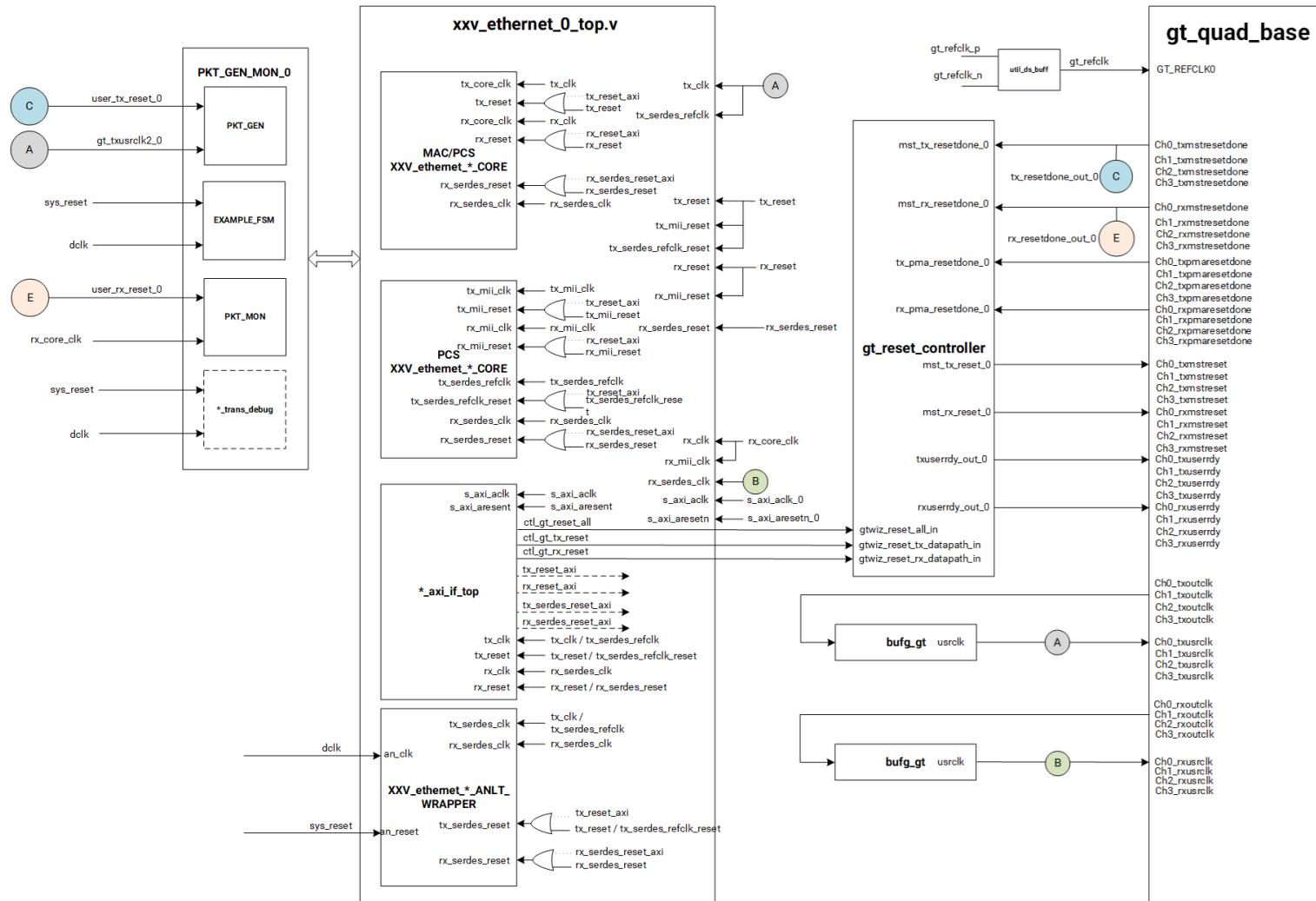
## 多核 - 异步时钟模式的详细图示 (UltraScale/UltraScale+)

图 23：多核 - 异步时钟模式的详细图示



## 单核详细图示 (Versal)

图 24：单核详细图示



X23975-120720

# 针对 IEEE 1588v2 标准的支持

## 概述

本节详述了包含 MAC 层时，10G/25G Ethernet Subsystem 的数据包时间戳功能。时间戳选项必须在从 IP 目录生成子系统时或者对 IP 核进行异步排序时指定。该功能特性可提供单步和双步 IEEE 1588v2 功能。

如果您所选的 IEEE PTP 1588v2 工作模式为“Two Step”，那么将仅填充双步相关端口，该核将仅执行双步时间戳功能。如果您所选的 IEEE PTP 1588v2 工作模式为“One Step”，那么将填充与单步和双步相关的所有端口，并且该核的单步和双步功能都将可用。

以太网帧在入口和出口均生成时间戳。该选项可用于实现所有种类的 IEEE 1588v2 时钟，包括：“Ordinary Clock”（普通时钟）、“Transparent Clock”（透明时钟）和“Boundary Clock”（边界时钟）。它还可用于在系统的入口端口和出口端口处为数据包生成通用时间戳。虽然此功能可用于各种数据包时间戳应用，但本节其余部分假定您将同时实现 IEEE 1588v2 精确时间协议 (PTP)。

IEEE 1588v2 定义了用于在整个网络中执行时序同步的协议。每个 1588 网络都具有单一主时钟参考，通常它是通过最佳主时钟算法来选定的。该主时钟会定期对其系统定时器参考计数器进行采样，并使用定义的数据包格式在整个网络中发射采样所得的时间值。该定时器应在 1588 时序包开始发射时进行采样（生成 1 个时间戳）。因此，要在整个网络上实现高精度同步，就需要精确的时间戳。如果将此采样所得定时器值（时间戳）放入触发时间戳的包内，这就被称为单步操作。或者，也可将此时间戳值置于后续数据包内，称为双步操作。

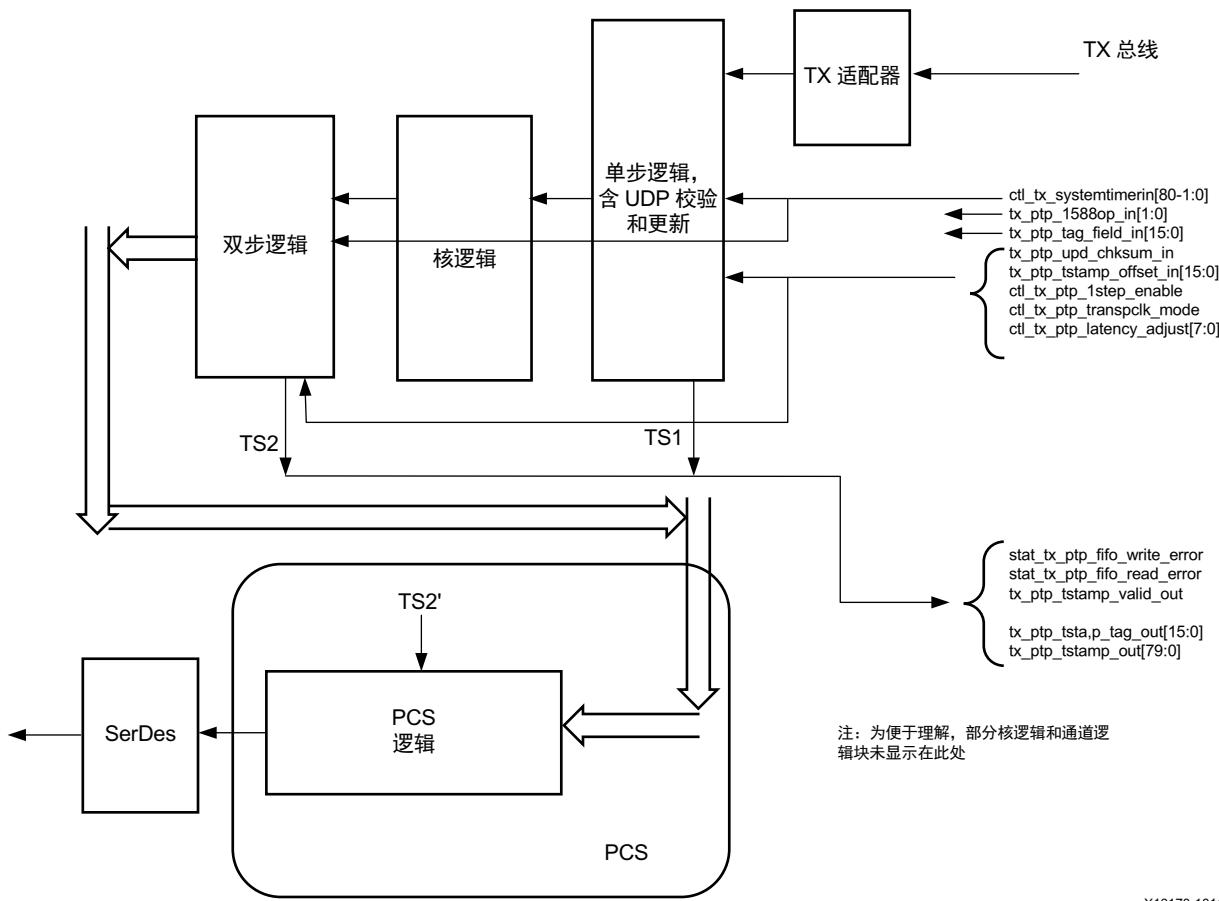
网络上的其它时序从器件会从网络时序主时钟接收到这些时序参考包，并尝试据此同步自己的本地定时器参考时钟。此机制要求在接收到 1588 时序包时，这些以太网端口同样生成时间戳（对其自己的本地定时器进行采样）。有关 1588 操作方式的进一步解释不在本文档的讨论范围内。本文档当前描述的是子系统的 1588 硬件时间戳功能特性。

提供给子系统的 1588 定时器以及从该定时器生成的后续时间戳可通过 2 种格式提供，在子系统生成时选择所用格式。

- 当日时间 (ToD) 格式：IEEE 1588-2008 格式由无符号的 48 位秒字段和 32 位纳秒字段组成。
- 纠正字段 (Correction Field) 格式：IEEE 1588-2008 数字格式包含 1 个 64 位有符号字段，该字段表示纳秒数乘以 216（请参阅 IEEE 1588 第 13.3.2.7 条）。此定时器应从 0 开始向上计数（全量程）直至 264 -1 为止，然后再回绕。

## 出口

图 25：出口



X16170-101122

如前图所示，根据所需的是单步操作还是双步操作，时间戳逻辑存在于两处不同位置。单步操作需用户数据报协议 (UDP) 校验和及 FCS，因此使用 FCS 核逻辑。

TS 参考定义如下：

- TS1：选中单步操作时的输出时间戳信号。
- TS2：选中双步操作时的输出时间戳信号。
- TS2'：2 个时间戳纠正到该平面。

对于 TS2，始终应用纠正以使其可供 TS2' 平面作为参考。TS1 可根据 `ctl_tx_ptp_latency_adjust[10:0]` 信号的值来选择是否应用 TS2' 纠正。

基于速率和时钟模式（普通或透明），`ctl_tx_ptp_latency_adjust[10:0]` 信号的建议默认值如下：

- 25G 普通时钟 = 370
- 25G 透明时钟 = 449
- 10G 普通时钟 = 925

- 10G 透明时钟 = 1132

在发射端，“Command”字段由客户端随发送的帧（以供发射之用）并行提供给子系统。这表示根据每个帧来确定要执行的 1588 功能（即，无操作、单步操作或双步操作），并且对于单步帧，这还可用于指示是否存在需更新的 UDP 校验和字段。

如果使用 ToD 格式，那么对于单步操作和双步操作，完整捕获的 80 位 ToD 时间戳将使用 [端口描述](#) 中定义的附加端口返回给客户端逻辑。

如果使用“Correction Field”（纠正字段）格式，那么对于单步双步操作，完整捕获的 64 位时间戳将使用 [端口描述](#) 中定义的附加端口返回给客户端逻辑（根据表中定义，上位数据设置为 0）。

如果使用 ToD 格式，那么对于单步操作，完整捕获的 80 位 ToD 时间戳将插入帧中。如果使用“Correction Field”格式，那么对于单步操作，捕获的 64 位时间戳将与帧中包含的现有“Correction Field”求和，求和所得结果将覆盖到帧中的原“Correction Field”中。针对单步时间戳支持的帧类型包括：

- 原始以太网 (Raw Ethernet)
- UDP/IPv4
- UDP/IPv6 对于单步 UDP 帧类型，将根据 IETF RFC 1624 来更新 UDP 校验和。对于所有单步帧，将在完成所有帧修改之后计算“Ethernet Frame Check Sequence (FCS)”（以太网帧校验序列）字段。对于双步发射操作，支持所有“Precision Time Protocol (PTP)”（精确时间协议）帧类型。

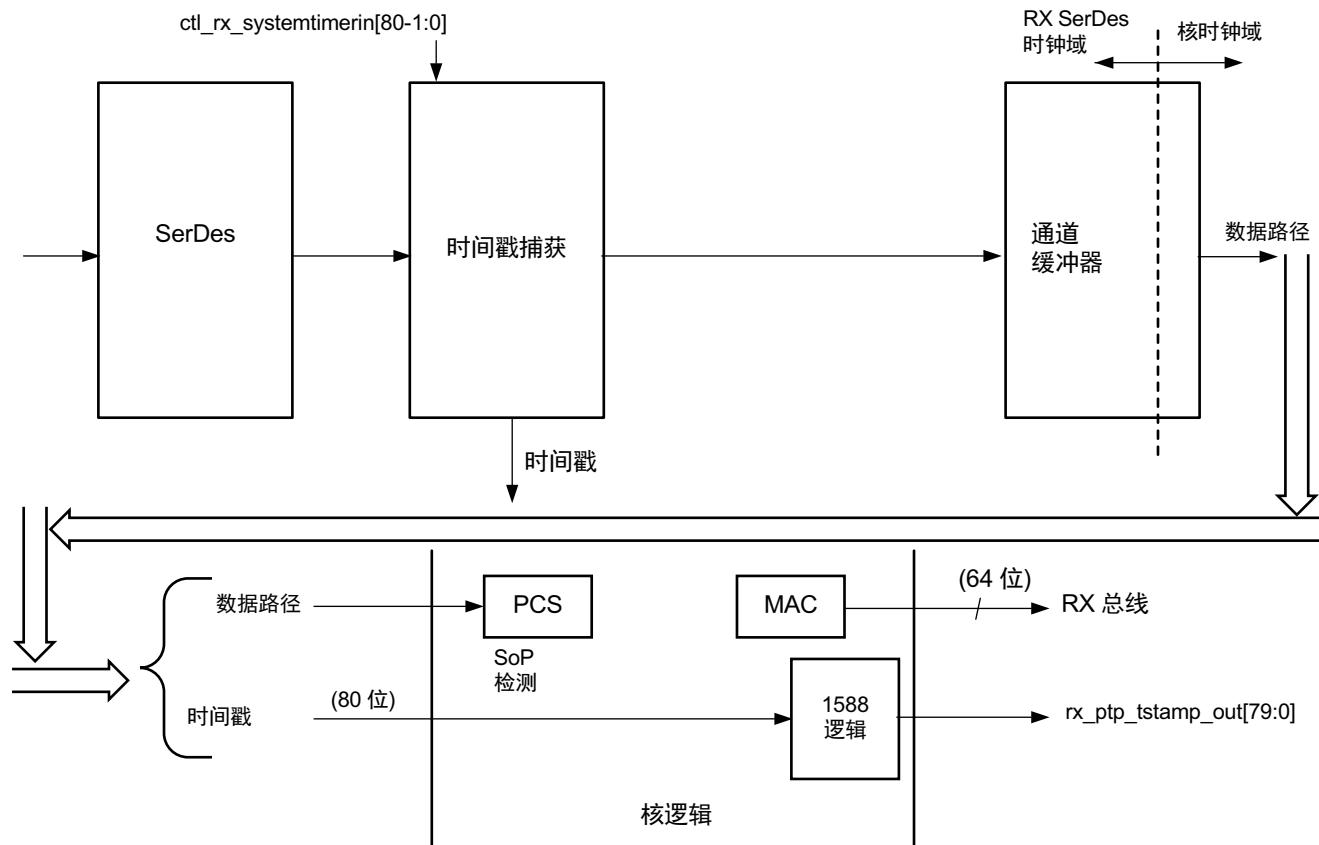
### 逐帧设置时间戳操作

出口时间戳功能的操作模式由 1588 命令端口上的设置判定。命令端口中所含信息用于指示以下任一结果：

- 无操作：该帧并非 PTP 帧，不应执行任何时间戳操作。
- 需执行双步操作并在“Command”字段中提供 1 个标签值（用户序列 ID）；对于该帧，应设置时间戳，并且所设时间戳可供客户端逻辑使用，所提供的标签值则可供该帧使用。附加 MAC 发射器端口可提供此功能。
- 需执行单步操作。
  - 对于 ToD 定时器和时间戳格式，将在命令端口中提供时间戳偏移；应对该帧设置时间戳，并且此时间戳应按提供的偏移（字节数）插入帧中。
  - 对于“Correction Field”格式，将在命令端口中提供“Correction Field”偏移值；应对该帧设置时间戳，捕获的 64 位时间戳应与该帧中包含的现有“Correction Field”求和，求和所得结果将覆盖到帧中的原“Correction Field”中。对于帧修改后的单步操作，还应更新/重新计算帧的循环冗余校验 (CRC) 值。对于 UDP IPv4 格式和 IPv6 PTP 格式的帧，需更新/重新计算该帧的报头中的校验和值。
- 对于单步 UDP 帧类型，将根据 IETF RFC 1624 来更新 UDP 校验和。
  - 如果使用“ToD”格式，为了能正确执行此更新功能，计算发送（以供发射）的帧的原校验和值时，针对时间戳数据应使用 0 值。使用“Correction Field”格式时，则不适用此特殊限制。
  - 如果使用“Correction Field”格式，则适用另一种限制；1588 PTP 帧报头中的“UDP Checksum”（UDP 校验和）字段与“Correction Field”之差为固定间隔（以字节为单位），以支持 1588 PTP 帧定义。此要求用于最大限度缩短通过 MAC 的时延，因为校验和与纠正字段都必须完全包含在 MAC 流水线内才能正确更新校验和。此特殊限制不适用于“ToD”格式，因为计算所得原时间戳数据为 0 值；因此，校验和与时间戳位置可分别位于同一帧中的不同位置。

## 入口

图 26：入口



X23834-101122

入口逻辑不会通过解析入口数据包来搜索 1588 (PTP) 帧。而是改为提取接收到的每个帧的时间戳，并将该值输出到用户逻辑。该功能始终启用，但如果无需此功能，则可忽略时间戳输出。

先过滤 PCS 解码器，然后再过滤时间戳，这样即可仅保留对应于包起始 (SoP) 的时间戳。这些 80 位时间戳在系统侧输出。在 SoP 周期内，此时间戳有效。

以太网帧的 1588 时间戳的准确性取决于多个因素，其中部分因素不在此 IP 核的适用范围内：

时间戳误差 = (10/25G IP 核 1588 时间戳误差) + (外部时间戳采样误差) + (因系统时钟抖动、收发器不确定性和其它隐私而导致的误差)

10/25G IP 核 1588 时间戳误差：

- 普通时钟模式：+/- 1 ns（由 ToD 格式的粒度所致）
- 透明时钟模式：+/- 1 SerDes 时钟位时间

影响时间戳准确性的因素：

- 此 IP 核要求用户对系统定时器输入进行采样，并将其重定时至指定的时钟域（TX SerDes 或 RX SerDes）。简单的采样电路将引发 +/- 1 SerDes 时钟误差。如果客户实现定制同步设计，则可提升准确性。
- 在此 IP 核内部不执行外部延迟（例如，收发器）补偿。这些延迟可以作为偏移来判定和添加。
- 对于具有 RSFEC 的 IP 核，时序层假定位于 RSFEC 外部，因此 RX 反相函数可以消除因 TX 上的转码与校验和插入而引发的任何变化。欲知详情，请参阅 802.3-2018 第 90.7 条。
- 对于具有 RSFEC 的 MAC+PCS IP 核，该核会对由于 AM 插入导致的速率调整所产生的影响加以补偿。
- 64b IP 核会对半块启动（含 S4 的 64B/66B 块）进行补偿。

## 端口描述

下表描述了包含数据包时间戳功能的情况下存在的其它信号。

表 291：1588v2 端口列表和描述

名称	I/O	描述	时钟域
<b>IEEE 1588 接口 - TX 路径</b>			
ctl_tx_systemtimerin[80-1:0]	输入	适用于 TX 的系统定时器输入。在正常时钟模式下，由 32 个 LSB 承载纳秒 (nsec)，由 48 个 MSB 承载秒 (second)。在透明时钟模式下，位 63 预期为零，位 62:16 的单位为纳秒，位 15:0 的单位为小数纳秒。请参阅 IEEE 1588v2 以了解相关表示法的定义。 此输入必须位于 TX SerDes 时钟域内。	tx_serdes_clk
tx_ptp_tstamp_valid_out	输出	此位用于表示 TX 系统接口上当前呈现的有效时间戳。	tx_clk_out
tx_ptp_tstamp_tag_out[15:0]	输出	对应于 tx_ptp_tag_field_in[15:0] 的标签输出	tx_clk_out
tx_ptp_tstamp_out[80-1:0]	输出	已发射的包 SOP 的时间戳，对应于它通过捕获平面的时间。用于双步 1588 操作。时间格式与定时器输入相同。	tx_clk_out
tx_ptp_1588op_in[1:0]	输入	此信号应在数据包的首个周期内有效。对于 PCS 核，第一个周期对应于数据包的第一个数据字。  2'b00 - No operation：表示无操作，不记录时间戳，不修改帧。 2'b01 - 1-step：表示单步，应记录时间戳并将其插入帧。 2'b10 - 2-step：表示双步，应记录时间戳并使用对应双步操作的额外端口将该时间戳返回到客户端。帧本身无需修改。 2'b11 - Reserved：表示保留，相当于“无操作”。	tx_clk_out
ctl_tx_ptp_1step_enable	输入	设为 1 时，该位即启用单步操作。	tx_clk_out
tx_ptp_upd_chksum_in		请参阅 <a href="#">IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号</a> 。	tx_clk_out
tx_ptp_chksum_offset_in		请参阅 <a href="#">IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号</a> 。	tx_clk_out
tx_ptp_tstamp_offset_in_*		请参阅 <a href="#">IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号</a> 。	tx_clk_out
ctl_ptp_transpclk_mode	输入	设为 1 时，此输入会将时间戳逻辑置于透明时钟模式下。在此模式下，系统定时器输入将解释为纠正值。TX 将根据 IEEE 1588v2 中定义的进程，把该纠正值添加到 TX 时间戳。纠正值的符号位假定为 0（正时间）。  这样对应的传入 PTP 包纠正字段应已按适当的 RX 时间戳完成调整。	tx_clk_out

表 291：1588v2 端口列表和描述 (续)

名称	I/O	描述	时钟域
tx_ptp_tag_field_in[15:0]	输入	该字段的使用取决于 1588 操作。此信号应在数据包的首个周期内有效。 <ul style="list-style-type: none"><li>· 对于 “No operation” , 将忽略该字段。</li><li>· 对于 “1-step” 和 “2-step” , 此字段将成为标签字段。该标签值将通过使用双步操作的额外端口, 与对应当前帧的时间戳一起返回至客户端。该标签值可供软件用于确保该时间戳能够与其发送的 PTP 帧 (用于发射) 相匹配。</li></ul>	tx_clk_out
ctl_tx_latency_*	输入	这是核 (包括 GT) 的 TX 路径的静态时延。MSB 16 位表示以 ns 为单位的延迟, LSB 16 位则表示亚纳秒值。时延采用二进制 Q16.16 格式。  <b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。	tx_clk_out
ctl_tx_lat_adj_enb_*	输入	当此信号处于使能 (启用) 状态时, TX 路径上的延迟计算会将 ctl_tx_latency_0 寄存器所提供的值一并纳入考量。  <b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。	tx_clk_out
ctl_tx_ptp_latency_adjust[10:0]	输入	该总线可用于调整与双步时间戳有关的单步 TX 时间戳。位 [10:3] 的单位为纳秒, 而位 [2:0] 的单位则为小数纳秒。	tx_clk_out
stat_tx_ptp_fifo_write_error	输出	发射 PTP FIFO 写入错误。此状态值为 1 表示 PTP Tag 写入期间发生错误。需 TX 路径复位才能清除此错误。	tx_clk_out
stat_tx_ptp_fifo_read_error	输出	发射 PTP FIFO 读取错误。此状态值为 1 表示 PTP Tag 读取期间发生错误。需 TX 路径复位才能清除此错误。	tx_clk_out
ctl_tx_timestamp_adj_enb_* (默认值: 1)	输入	当此信号处于使能 (启用) 状态时, TX 路径上的延迟计算会将 GT DRP 读取的 TX 变速箱 FIFO 时延值一并纳入考量。  <b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。	tx_clk_out
tx_period_ns_*	输出	请参阅 <a href="#">IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号</a> 。	tx_clk_out
<b>IEEE 1588 接口 - RX 路径</b>			
ctl_rx_systemtimerin[80-1:0]	输入	适用于 RX 的系统定时器输入。时间格式与 TX 相同。此输入必须与 RX SerDes 位于相同时钟域内。  <b>注释：</b> 此信号在 1G 模式下无效。	rx_serdes_clk
rx_ptp_tstamp_out[80-1:0]	输出	已接收的包 SOP 的时间戳, 对应于它通过捕获平面的时间。请注意, 此信号将在数据包的首个周期内有效。用于双步 1588 操作。	rx_core_clk
rx_ptp_tstamp_valid_out_*	输出	此位用于表示 RX 系统接口上当前呈现的有效时间戳。	rx_core_clk
rx_ptp_pcslane_out	输出	该总线用于指示检测到对应时间戳的 SOP 所在的 PCS 通道。请注意, 只要针对一个分段断言 SOP 有效, 则从断言有效的时钟周期开始, 此信号即生效。	rx_serdes_clk
rx_lane_aligner_fill_0	输出	此输出用于指示 PCS lane0 的对齐缓冲器的填充级别。此信息可供 PTP 应用配合 rx_ptp_pcslane_out_* 信号一起使用, 以便调整即将到达的 SOP 的通道偏差。单位为 SerDes 时钟周期数。	
rx_lane_aligner_fill_1	输出	此输出用于指示 PCS lane1 的对齐缓冲器的填充级别。	
rx_lane_aligner_fill_2	输出	此输出用于指示 PCS lane2 的对齐缓冲器的填充级别。	

表 291：1588v2 端口列表和描述 (续)

名称	I/O	描述	时钟域
rx_lane_aligner_fill_3	输出	此输出用于指示 PCS lane3 的对齐缓冲器的填充级别。	
ctl_rx_latency_*	输入	<b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。这是核（包括 GT）的 RX 路径的静态时延。MSB 16 位表示以 ns 为单位的延迟，LSB 16 位则表示亚纳秒值。时延采用二进制 Q16.16 格式。	rx_clk_out
ctl_rx_lat_adj_enb_* (默认值：1)	输入	当此信号处于使能（启用）状态时，RX 路径上的延迟计算会将 ctl_rx_latency_0 寄存器所提供的值一并纳入考量。 <b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。	rx_clk_out
ctl_rx_timestamp_adj_enb_* (默认值：1)	输入	当此信号处于使能（启用）状态时，RX 路径上的延迟计算会将 GT DRP 读取的 RX 变速箱 FIFO 时延值一并纳入考量。 <b>注释：</b> 此信号仅适用于 MAC+PCS/PMA 32 位版本和 PCS/PMA 32 位版本。	rx_clk_out
rx_period_ns_*	输出	请参阅 <a href="#">IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号</a> 。	rx_clk_out

## IEEE 1588v2 PTP 功能描述

10G/25G High Speed Ethernet Subsystem 的 IEEE 1588v2 功能在硬件级别为出入双向提供准确的以太网帧时间戳。

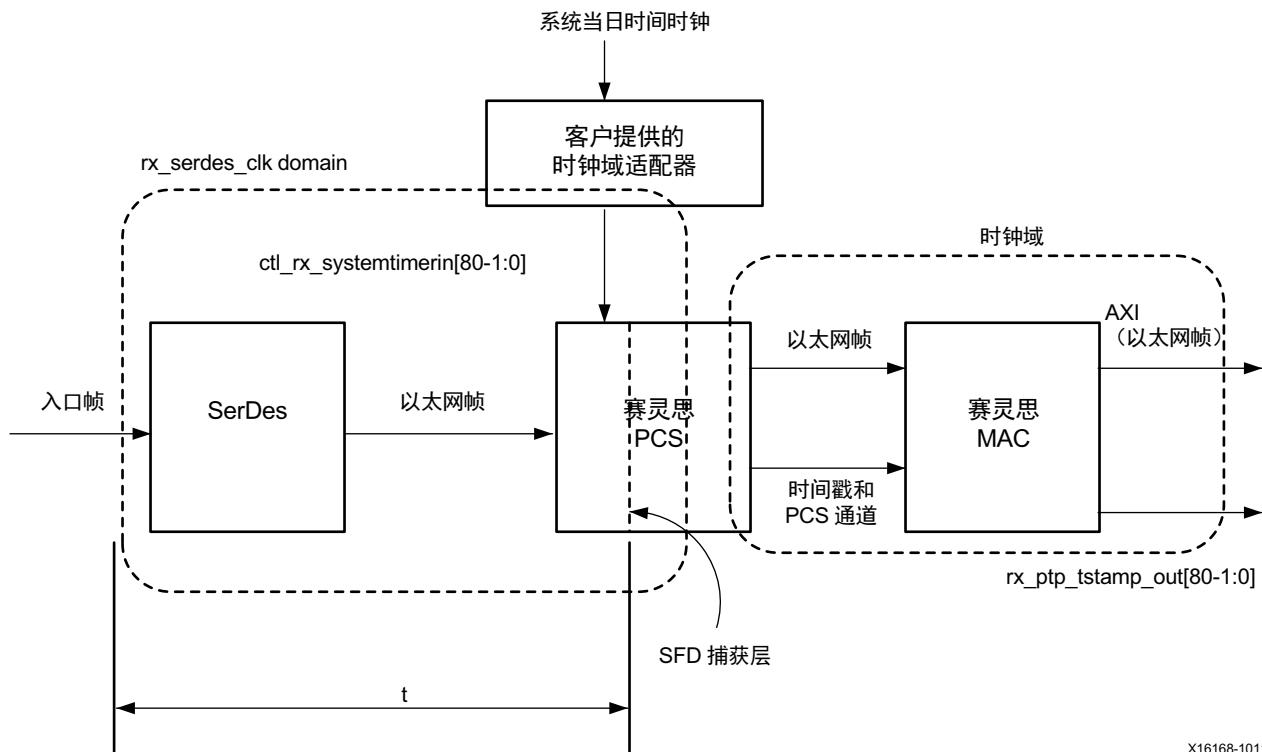
时间戳根据上述输入时钟源来捕获。不过，此时间源必须与 SerDes 位于同一个时钟域中。用户提供的外部电路可能需对此进行重定时。

所有入口帧都会获得时间戳。您需要解读收到的帧，并确定特定帧是否含有 PTP 信息（通过其以太类型来判断）以及判断此时间戳需保留还是丢弃。如果出口帧被标记为 PTP 帧，则为其设置时间戳。出口帧的时间戳与其用户提供的标签相匹配。

传入帧的时间戳则在包起始的时钟周期内显示在用户接口上。然后，您可以根据需要将时间戳添加到包上。

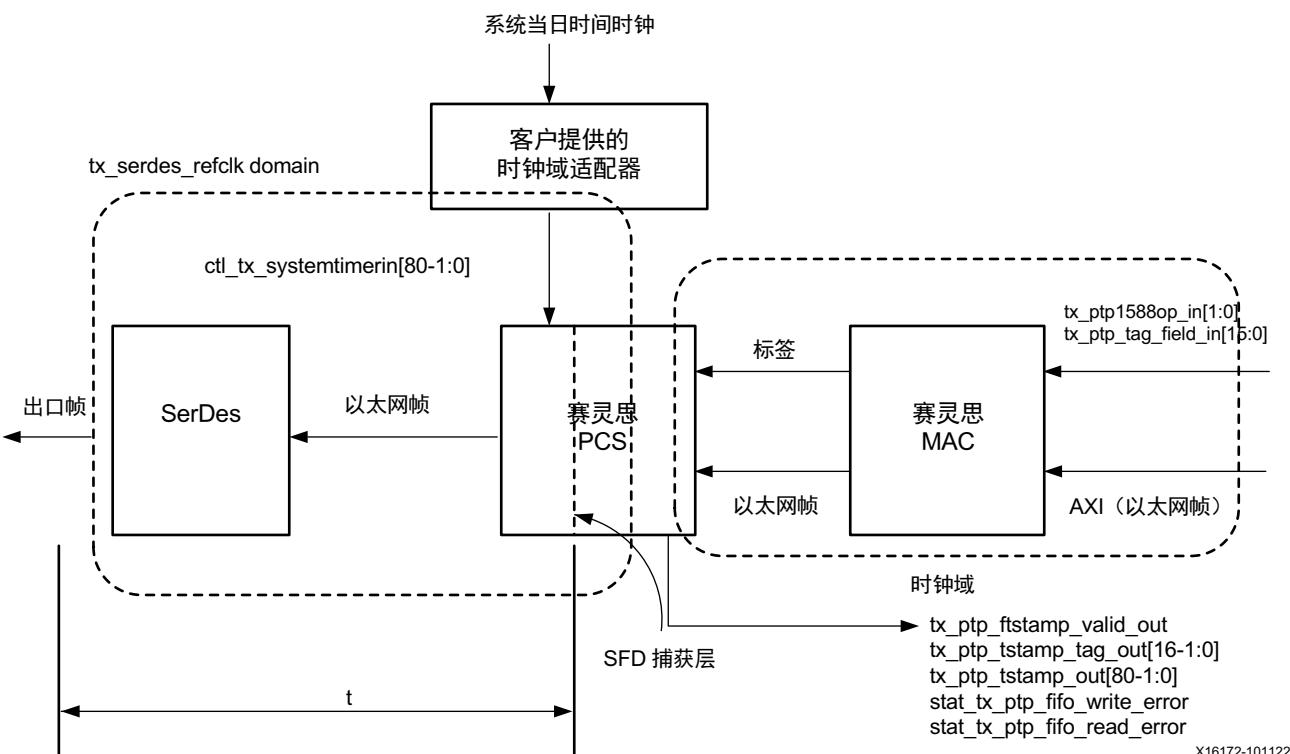
按照定义，时间戳的捕获时间与 SOP 通过 10G/25G High Speed Ethernet Subsystem 中捕获平面的时间一致，如下图所示。

图 27：接收



X16168-101122

图 28：发射



X16172-101122

## 性能

在典型应用中，入口和出口的捕获时间差对于判定绝对时间至关重要。PTP 算法可使用非对称信息来提升精度。

1588v2 功能要求所有时钟频率都已知才能执行内部计算。时钟频率应在订购 PTP IP 核时指定，以便时间戳纠正功能正常工作。

在典型的应用中，PTP 算法（或 servo，不属于此 IP）将移除一段时间内（许多包样本）的抖动。这种抖动最好要尽可能小，以最大程度减少收敛时间和从时钟漂移。

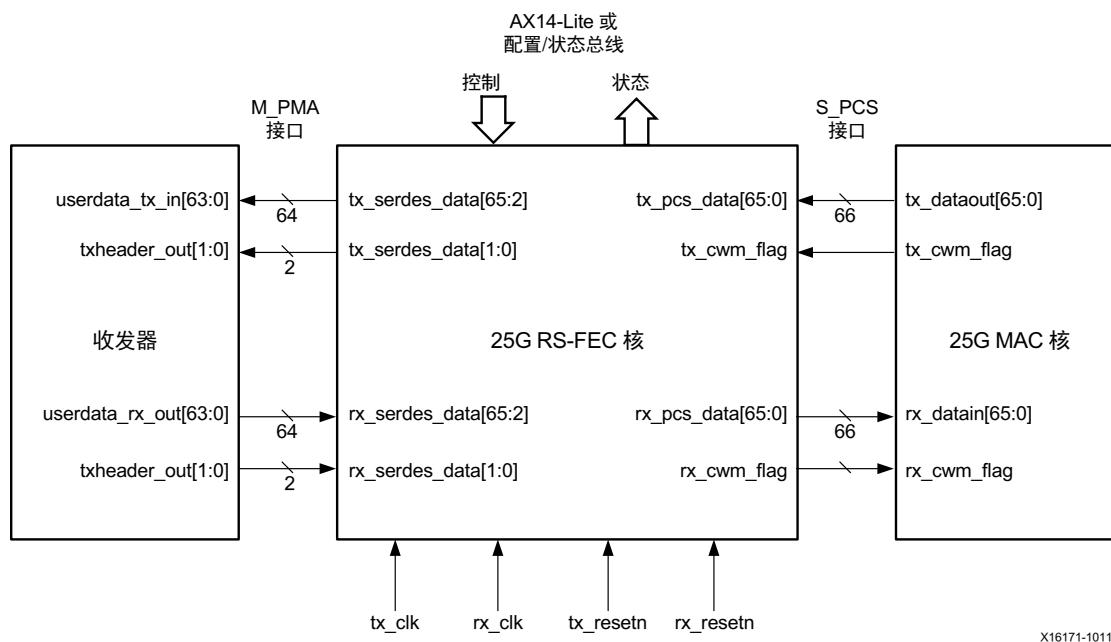
## RS-FEC 支持

### 概述

本节描述了 10G/25G Ethernet Subsystem 的可选 RS-FEC 功能。RS-FEC 选项必须在从 IP 目录生成子系统时或者对 IP 核进行异步排序时指定。

请参阅下图，RS-FEC 核时钟和复位与收发器信号等效，收发器复位处于高电平有效状态。RS-FEC 块位于 PCS 和 PMA 之间，如下图所示。

图 29：RS-FEC 模块框图



RS-FEC 内部详细信息不在本文档讨论范围内。请参阅 IEEE 802.3 第 108 条和 25G 以太网联盟的 Schedule 3。

如需了解更多信息，请参阅《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》(PG217) (需注册)。

## 端口描述

表 292：RS-FEC 端口列表和描述

端口	方向	描述	时钟域
<b>RS-FEC 控制信号</b>			
ctl_rsfec_ieee_error_indication_mode	输入	当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 <ul style="list-style-type: none"><li>· 1：核遵循 IEEE RS-FEC 规范。</li><li>· 0：如果 ctl_rx_rsfec_enable_correction 和 ctl_rx_rsfec_enable_indication 均设为 0，则绕过 RS 解码器。</li></ul>	rx_serdes_clk
ctl_rsfec_consortium_25g	输入	在 IEEE 第 108 条与 25G 以太网联盟模式之间进行切换。 当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 <ul style="list-style-type: none"><li>· 1 = 25G 以太网联盟规范模式。</li><li>· 0 = IEEE 802.3by 模式。</li></ul> <p>请注意，10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 联盟信号。</p>	rx_serdes_clk
ctl_rsfec_enable	输入	当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 启用 RS-FEC 功能。请注意，10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 使能信号。	rx_serdes_clk
ctl_rx_rsfec_enable_correction	输入	当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 等效于 MDIO 寄存器 1.200.0 <ul style="list-style-type: none"><li>· 0：解码器执行错误检测，但不执行纠错（请参阅 IEEE 802.3802.3by 第 91.5.3.3 条）。</li><li>· 1：解码器还会执行纠错。</li></ul>	rx_serdes_clk
ctl_rx_rsfec_enable_indication	输入	当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 等效于 MDIO 寄存器 1.200.1 <ul style="list-style-type: none"><li>· 0：绕过错误指示功能（请参阅 IEEE 802.3by 标准第 91.5.3.3 条）。</li><li>· 1：解码器向 PCS 子层发出错误指示。</li></ul>	rx_serdes_clk
ctl_rx_vl_length_minus1[15:0]	输入	通常设为 20,479 (4FFF 十六进制)。常规值等于 $(16,383 \times 5 - 4) = 81,916$ 。	
ctl_rx_vl_marker_id0[63:0]	输入	等于 IEEE 802.3 第 82 条中对应 40G 以太网定义的 RX PCS 通道 0 对齐标记。	
ctl_rx_vl_marker_id1[63:0]	输入	等于 PCS 通道 1 对齐标记。	
ctl_rx_vl_marker_id2[63:0]	输入	等于 PCS 通道 2 对齐标记。	
ctl_rx_vl_marker_id3[63:0]	输入	等于 PCS 通道 3 对齐标记。	
ctl_tx_vl_length_minus1[15:0]	输入	通常设为 20479 (十进制)。常规值等于 $(16,383 \times 5 - 4) = 81,916$ 。	
ctl_tx_vl_marker_id0[63:0]	输入	等于 IEEE 802.3 第 82 条中对应 40G 以太网定义的 TX PCS 通道 0 对齐标记。	
ctl_tx_vl_marker_id1[63:0]	输入	等于 PCS 通道 1 对齐标记。	

表 292：RS-FEC 端口列表和描述 (续)

端口	方向	描述	时钟域
ctl_tx_vl_marker_id2[63:0]	输入	等于 PCS 通道 2 对齐标记。	
ctl_tx_vl_marker_id3[63:0]	输入	等于 PCS 通道 3 对齐标记	
<b>RS-FEC 状态信号</b>			
stat_rx_rsfec_corrected_cw_inc	输出	此信号为递增值，对应已纠正的错误数。	rx_serdes_clk
stat_rx_rsfec_uncorrected_cw_inc	输出	此信号为递增值，对应未纠正的错误数。	rx_serdes_clk
stat_rx_rsfec_err_count_inc[2:0]	输出	此信号为递增值，对应已检测到的错误数。	rx_serdes_clk
stat_rx_rsfec_hi_ser	输出	如果含 8192 个代码字的窗口内的 RS-FEC 符号错误数量超过阈 K = 417，则此项设置为 1，否则设为 0。	rx_serdes_clk
stat_rx_rsfec_lane_alignment_status	输出	值为 1 表示 RX RS-FEC 块在来自收发器的数据上已达成对齐。	rx_serdes_clk
stat_tx_rsfec_lane_alignment_status	输出	值为 1 表示 TX RS-FEC 块在传入 PCS 数据上已达成对齐。	rx_serdes_clk

## RS-FEC 功能描述

根据 IEEE 802.3 第 108 条或 25G 以太网联盟的 Schedule 3，10G/25G 子系统的 RS-FEC 功能特性可提供纠错功能。

该功能需按 IEEE 802.3 表 82-2 中的定义，插入 PCS 对齐标记。针对对齐标记以及对齐标记之间的代码字值都提供了输入。

可通过使能信号绕过 RS-FEC 功能。这将绕过 RS-FEC 功能并将 PCS 直接连接到收发器，好处是可降低时延。请参阅《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》([PG217](#)) (需注册)，以获取各旁路模式（定义如下）下最新时延性能数据：

- FEC Bypass Correction：解码器执行错误检测，但不纠错（请参阅 IEEE 802.3by 标准第 108.5.3.2 条）。在此模式下可降低时延（请参阅《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》([PG217](#)) (需注册) 中的时延图示）。
- FEC Bypass Indication：在此模式下，执行数据纠错，但无错误指示。在此模式下会生成额外信号 rx\_hi\_ser 用于降低无法检测到包中错误的可能性。RS 解码器会计算在 8192 个代码字的非重叠块中检测到的符号错误的数量（请参阅 IEEE 802.3by 标准第 108.5.3.2 条）。在此模式下可降低时延。
- Decoder Bypass：当纠错旁路和错误指示旁路均处于高位 (High) 时，可通过将“IEEE Error indication”设置为“Low”来绕过 RS 解码器。

---

## 以太网数据路径奇偶校验

数据路径奇偶校验 (Datapath Parity) 功能在该核的 AXI4-Stream 接口与 Ethernet FCS 逻辑之间驻留的数据路径逻辑上提供了软核错误检测。

对于此数据路径的每个字节，都提供了 1 个单一位元 (single bit) 以反映该字节计算所得的奇偶校验结果。



**重要提示！** 此 IP 核通过实现偶校验使数据路径内包含奇偶校验位的所有 1 的总和为偶数（即，0、2、4...）。

在该核的数据路径的发射方向和接收方向中都会实现数据路径奇偶校验。在这两个方向上，仅在有效的 AXI4-Stream 字节上才将奇偶校验位视为有效。具体来说，AXI4-Stream 字节在 TKEEP[n]、TVALID 和 TREADY 全部断言有效的周期内才被视为有效。与接收数据路径内的 AXI4-Stream 无效字节关联的奇偶校验位应予以忽略。与发射方向内的 AXI4-Stream 无效字节关联的奇偶校验位将不予检查。

数据路径奇偶校验功能添加了单一统计数据 `stat_tx_bad_parity` 用于识别是否检测到软核错误。此统计数据基于时钟周期，因此它可针对单一数据包多次断言有效。并且由于它基于时钟周期，因此它只能识别已检测到至少一个软核错误。如果在单一时钟周期内检测到 2 个软核错误，那么将仅标示 1 个软核错误。根据配置，存在奇偶校验错误还可能触发 `stat_tx_bad_fcs` 错误断言有效。

对于支持奇偶校验的变体，接收数据路径无需配置，因为 AXI4-Stream 接口上存在的所有数据包都将生成奇偶校验。在发射方向上，下表描述了影响奇偶校验逻辑行为的配置位：

表 293：影响奇偶校验逻辑行为的配置位

<code>ctl_tx_fcs_ins_enable</code>	<code>ctl_tx_parity_err_response</code>	是否为以太网 FCS 添加 stomp 标记的行为	<code>stat_tx_bad_parity</code> 行为	<code>stat_tx_bad_fcs</code> 行为
0	0	核将不会为 FCS 添加 stomp 标记，但用户逻辑可能已为 FCS 添加 stomp 标记。	如果检测到奇偶校验错误， <code>stat_tx_bad_parity</code> 将断言有效。 <sup>1</sup>	如果传入包中用户生成的 FCS 不正确， <code>stat_tx_bad_fcs</code> 将断言有效。奇偶校验错误无论是否存在都不会导致 <code>stat_tx_bad_fcs</code> 断言有效。
0	1	核将不会为 FCS 添加 stomp 标记，但用户逻辑可能已为 FCS 添加 stomp 标记。	同上。 <sup>1</sup>	同上
1	0	不支持	同上。 <sup>1</sup>	奇偶校验错误无论是否存在都不会导致 <code>stat_tx_bad_fcs</code> 断言有效。
1	1	是，前提是检测到奇偶校验错误。	同上。 <sup>1</sup>	如果检测到奇偶校验错误， <code>stat_tx_bad_fcs</code> 将断言有效。 <sup>2</sup>

注释：

- 满足如下所述限制时，`stat_tx_bad_parity` 将不会断言有效：在欠载包（即，数据包同时包含 `tx_axis_tuser` 和 `tlast`，或者包含 `de-assertstvalid` 但不含 `tlast`）的最后一个时钟周期内接收到的奇偶校验错误，或者在发生 LBUS FIFO 下溢事件时。
- 满足如下所述限制时，`stat_tx_bad_fcs` 将不会断言有效：仅在欠载包（即，数据包同时包含 `tx_axis_tuser` 和 `tlast`，或者包含 `de-assertstvalid` 但不含 `tlast`）的最后一个时钟周期内接收到的奇偶校验错误，或者在发生 LBUS FIFO 下溢事件时。在最后一个时钟周期之前接收到的奇偶校验错误仍将导致 `stat_tx_bad_fcs` 断言有效。

## 802.1cm 抢占功能

### 功能特性

基于《IEEE 局域网和城域网标准 - 适用于前传的时间敏感网络》（[IEEE 802.1CM-2018 标准](#)）的可选付费型时间敏感网络 (TSN) 功能。

- 支持帧抢占

- 支持高速流量与低优先级流量穿插

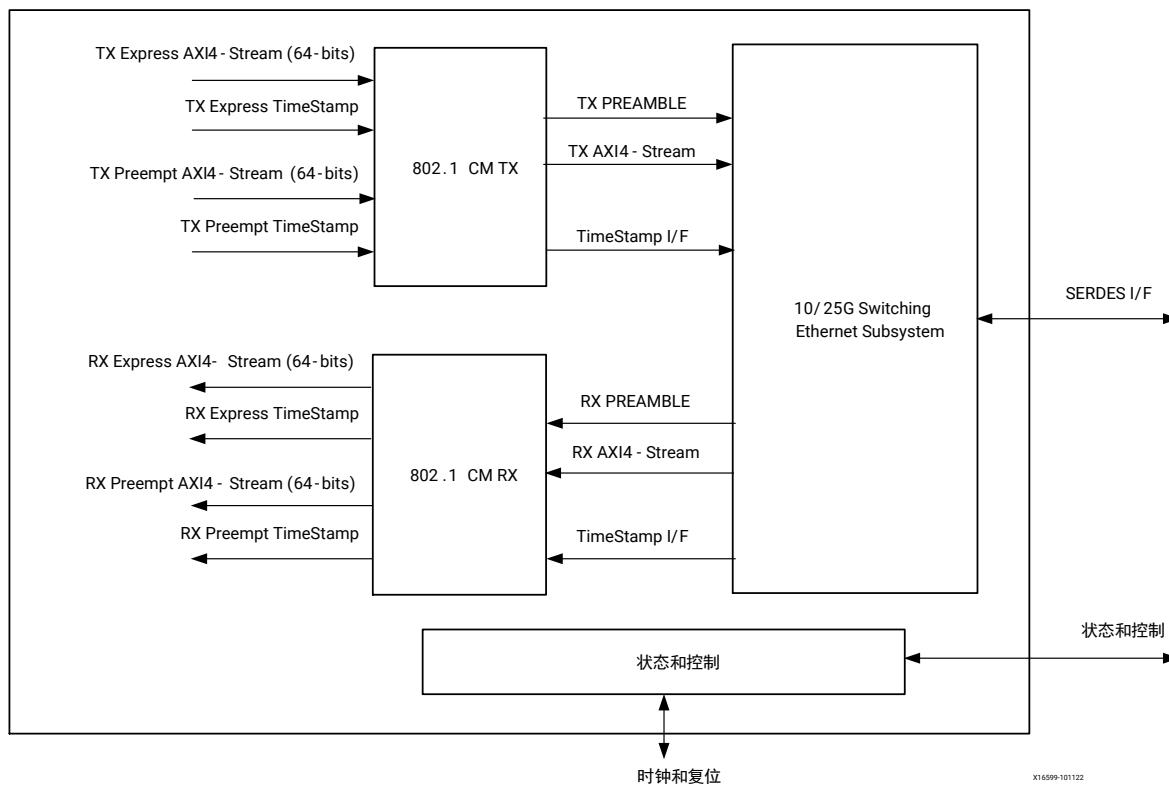
## 概述

帧抢占和高速流量穿插符合 IEEE 802.1 CM 标准的定义。10G/25G Ethernet Subsystem 包含基于 IEEE 802.1 CM 的可选 TSN 功能特性。

## 产品规格

下图显示了含可选 TSN 功能特性的 10/25G Ethernet IP 子系统的模块框图：

图 30：含可选 TSN 的 10/25G Ethernet IP 子系统



**注释：**启用“802.1cm Preemption”（802.1cm 抢占）功能特性时，仅在 Express 接口上支持 PTP 时间戳。

## AXI4-Stream 发射接口

该核包含 2 个 AXI4-Stream 接口，如果生成的核包含可选 TSN 功能特性，则这 2 个接口可用于高速流量 (`tx_axis_e_*`) 和抢占流量 (`tx_axis_p_*`)。如需了解详细信息，请参阅 [AXI4-Stream 发射接口](#) 中的信号相同。

**注释：**对于这些信号所适用的描述和规则与 [AXI4-Stream 发射接口](#) 中的信号相同。

在核生成期间，可以选择在抢占接口上插入 FIFO。插入此 FIFO 时，入口帧将缓冲，仅当此 FIFO 中可用的帧完全无错时，才会在 AXI4-Stream 接口上提供该帧以供使用。

## 帧发射

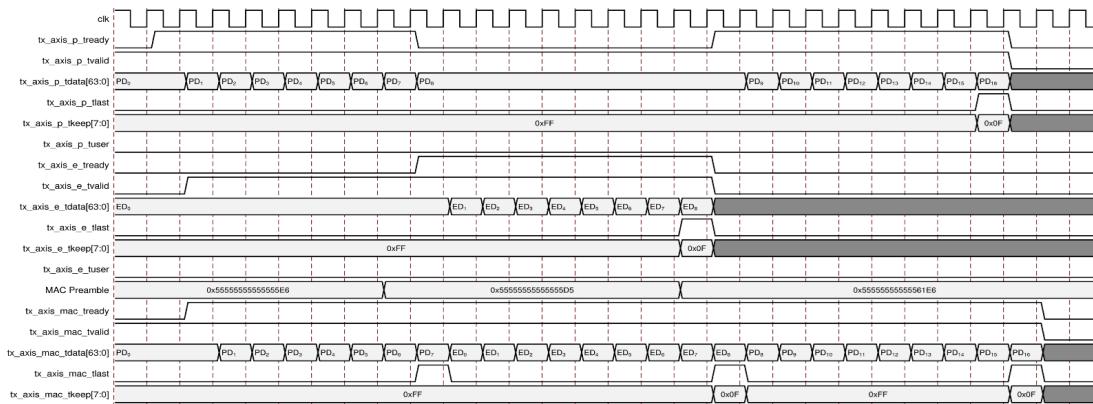
如果禁用抢占，则核会基于先入先出的原则来服务高速流量和抢占流量。如果两个接口同时提供帧，则首先服务高速流量，然后服务抢占流量。如需获取有关提供帧以供发射的详细信息，请参阅“AXI4-Stream 接口”章节。

启用抢占前，必须首先确定链路伙伴的抢占功能。判定依据是根据 IEEE 802.3 标准的第 79.3.7 条中所述方式执行“其它以太网功能 TLV”交换。核假定您仅在判定链路伙伴同样支持抢占后才会启用抢占。如果通过断言 `ctl_en_preempt` 有效来启用抢占，那么该核会首先验证抢占操作，前提是 `ctl_disable_verify` 已断言无效。仅当成功完成验证后，才会激活抢占。您也可以禁用验证，在此情况下，该核不会尝试启动验证进程，而是直接激活抢占。抢占激活后，该核会按如下方式来服务帧发射请求：

- 如果高速接口处于不活动状态，则发射抢占接口上提供的帧。
- 如果高速接口和抢占接口同时请求帧发射，则发射高速帧。
- 如果在发射抢占流量时，高速接口提供帧，那么根据 IEEE 802.3 br-2106 标准中指定的如下公式，抢占流量将被抢占： $pAllow * (eTx + hold) * preemptableFragSize * MIN\_REMAIN$
- 高速帧完成发射后，如果队列中没有其它高速帧在等待发射，那么该核会恢复发射抢占帧。

如需获取有关抢占流量与高速流量穿插方式的详细信息，请参阅 IEEE 802.1 CM 标准。以下时序图演示了抢占和穿插方式。如需获取有关抢占流量与高速流量穿插方式的详细信息，请参阅 [IEEE 802.1CM-2018 标准](#)。

图 31：抢占和穿插



**注释：**在上图中为保持紧凑，并未呈现帧间间隔(IFG)。但存在 IFG 并导致帧间 `tx_axis_mac_tready` 断言无效。

### 相关信息

[AXI4-Stream 接口](#)

## AXI4-Stream 接收接口

[AXI4-Stream 接收接口](#) 中显示 2 个 AXI4-Stream 接口，如果生成的核包含可选 TSN 功能特性，则这 2 个接口可用于高速流量 (`tx_axis_e_*`) 和抢占流量 (`tx_axis_p_*`)。

**注释：**对于这些信号所适用的描述和规则与 [AXI4-Stream 接收接口](#) 中的信号相同。

## 帧接收

入口帧可采用抢占 (preempt) 或高速 (express) 类型。核会根据判定的类型将入口帧分别置于相应的高速 AXI4-Stream 接口或抢占 AXI4-Stream 接口上。

高速流量将连续传输，由于核在此接口上不具有任何缓冲机制，您必须准备好随时接受高速帧。

由于抢占流量的性质，帧能以一组碎片的形式到达，后续由核进行汇编。在核生成期间，可以选择在抢占接口上插入 FIFO。插入此 FIFO 时，将缓冲该抢占帧的所有碎片，仅当汇编进程成功完成后，该帧才会在 AXI4-Stream 接口上可用。如果不插入 FIFO，那么 AXI4-Stream 接口上的 *tvalid* 可能在不同碎片之间出现波动。核通过断言 *tlast* 有效来指示碎片 *tlast* 汇编进程完成；如果汇编进程不成功，则会断言 *tuser* 有效。

下图显示了不插入 FIFO 时 AXI4-Stream 接口上的抢占帧碎片的呈现方式。

图 32：抢占帧碎片：不插入 FIFO 时

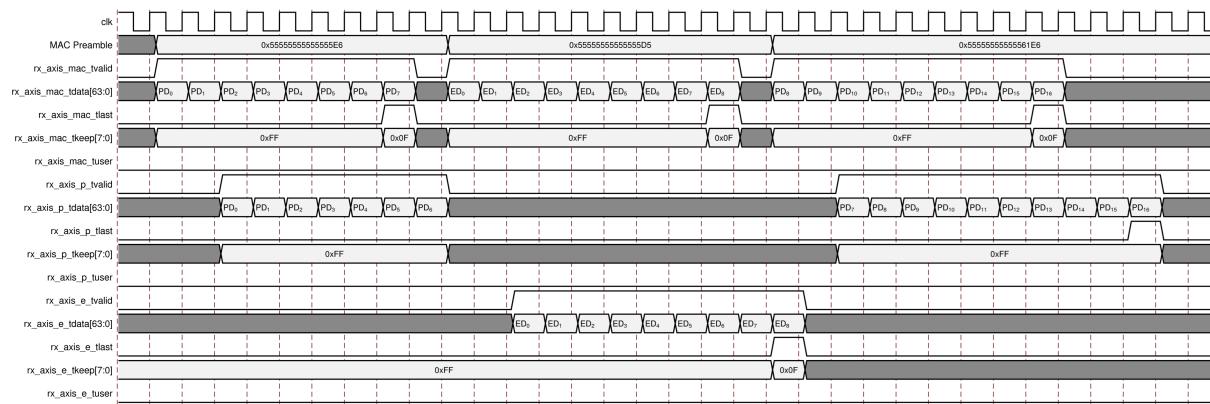


表 294：控制端口和状态端口

名称	I/O	描述	时钟域
<b>控制</b>			
ctl_en_preempt	输入	断言有效时，允许抢占。 首次断言有效时，如果 <i>ctl_disable_verify</i> = 1'b0 且 <i>stat_tx_mm_verified[1:0]</i> = 2'b00，则会触发验证。	<i>tx_clk_out</i>
ctl_hold_request	输入	如果断言有效，则阻止抢占流量。	<i>tx_clk_out</i>
ctl_disable_verify	输入	如果断言有效，则阻止验证进程。	<i>tx_clk_out</i>
ctl_restart_verify	输入	如果 <i>ctl_disable_verify</i> = 1'b0，那么 0 到 1 转换将触发验证。	<i>tx_clk_out</i>
ctl_addfrag_size[1:0]	输入	启用抢占前的剩余碎片大小： 2'b00 = 64 字节 2'b01 = 128 字节 2'b10 = 192 字节 2'b11 = 256 字节	<i>tx_clk_out</i>
ctl_verify_time[7:0]	输入	验证超时值（以毫秒为单位）。整数范围 1-128。默认值为 1 ms。	<i>tx_clk_out</i>
ctl_verify_limit[3:0]	输入	核尝试验证的次数。整数范围 1-15。默认值为 3。	<i>tx_clk_out</i>
p_frame_len_0	输入	表示 TX 抢占 AXI-S 接口上呈现的帧长度（以字节为单位）。在抢占帧 SOP 时应有效。	<i>tx_clk_out</i>

表 294：控制端口和状态端口（续）

名称	I/O	描述	时钟域
状态			
stat_tx_mm_verify[1:0] (无计数器)	输出	表示验证状态。 [0] - 断言有效时，表示验证完成。 [1] - 断言有效时，表示验证成功。 仅当 <code>ctl_en_preempt = 1'b1</code> 且 <code>ctl_disable_verify = 1'b0</code> 时，该状态矢量中包含的值才有效	tx_clk_out
stat_tx_mm_status	输出	当发射可抢占的数据包（初始碎片或完整包）时，此信号断言有效。	tx_clk_out
stat_tx_mm_fragment	输出	当发射可抢占包的持续碎片时，此信号断言有效。	tx_clk_out
stat_tx_mm_hold	输出	当 <code>ctl_hold_request</code> 从 1'b0 转换为 1'b1 时，此信号断言有效。	tx_clk_out
stat_rx_mm_assembly_error	输出	碎片汇编期间，检测到错误时，此信号断言有效。	rx_core_clk
stat_rx_mm_frame_smd_error	输出	此信号在以下情况下断言有效：当帧碎片因 SMD 值错误而遭拒绝，或者帧碎片到达时包含 SMD-C（因当前未处理任何帧）。	rx_core_clk
stat_rx_mm_frame_assembly_ok	输出	当所有可抢占帧碎片均已汇编完成并呈现时，此信号断言有效。	rx_core_clk
stat_rx_mm_fragment	输出	当接收到碎片帧时，此信号断言有效。	rx_core_clk
rx_p_frm_drop_0	输出	当数据包汇编 FIFO 因已满而无法继续处理抢占碎片汇编时，会断言此信号有效。发生此情况时，接收到的抢占帧的所有碎片都将被丢弃。	rx_core_clk
rx_p_frm_drop_count_0	输出	此类事件的数量。	rx_core_clk

**注释：**

- 高速和抢占接口的时间戳端口描述与 1588v2 端口描述类似。欲知详情，请参阅 [端口描述](#)。

## 状态/控制接口

状态/控制接口允许您设置 10G/25G Ethernet Subsystem 核配置并监控其状态。本节对部分状态信号和控制信号进行了详细描述。

### **stat\_rx\_framing\_err 和 stat\_rx\_framing\_err\_valid**

这些信号用于保留同步报头错误记录。这组总线用于保留同步报头错误记录。`stat_rx_framing_err` 输出用于指示已接收到的同步报头错误数量，当对应 `stat_rx_framing_err_valid` 采样为 1 时，此输出合格（即，仅限在此情况下其值才有效）。

### **stat\_rx\_block\_lock**

该位表示此接口已达成 IEEE 802.3 标准所定义的同步报头锁定。值为 1 表示已达成块锁定。

## stat\_rx\_local\_fault

当 stat\_rx\_internal\_local\_fault 或 stat\_rx\_received\_local\_fault 断言有效时，此输出为高电平 (High)。此输出对电平敏感。

## RX 错误状态

该核系统提供了一些状态信号，用于指示 64 位/66 位代码字和序列违例以及 CRC32 校验故障。所有信号都与 clk 上升沿同步。以下是每个信号的详细描述。

### stat\_rx\_bad\_fcs[1:0]

当此信号为正值时，表示错误检测逻辑已在接收到的包中发现接收的 CRC32 值与期望的值不匹配。当检测到 CRC32 错误时，接收到的包将被标记为包含错误，并随在上一次传输（即断言 rx\_eopout 有效的周期）期间断言有效的 rx\_errout 一起发送，但断言 ctl\_rx\_ignore\_fcs 有效的情况下除外。对于检测到的每个 CRC32 错误，此信号都将断言有效并保持 1 个时钟周期。

### stat\_rx\_bad\_code

此信号用于指示 RX PCS 接收状态机处于 IEEE 802.3 标准所定义的 RX\_E 状态的周期数。

## 暂停处理

10G/25G Ethernet Subsystem 提供了一整套暂停包终止和生成机制。TX 和 RX 具有独立接口用于处理暂停信息，如本节中所述。

### TX 暂停生成

您可以使用 ctl\_tx\_pause\_req[8:0] 和 ctl\_tx\_pause\_enable[8:0] 输入总线请求发射暂停包。位 [8] 对应于全局暂停包，位 [7:0] 对应于优先暂停包。

此总线的每个位都必须保持稳定状态达至少 16 个周期，随后才能执行下一次转换。



**注意！** 同时请求全局和优先暂停包会导致不可预测的行为，必须避免。

暂停包的内容可使用下列输入管脚来确定。

全局暂停包：

```
ctl_tx_da_gpp[47:0]
ctl_tx_sa_gpp[47:0]
ctl_tx_ether_type_gpp[15:0]
ctl_tx_opcode_gpp[15:0]
ctl_tx_pause_quanta8[15:0]
```

优先暂停包：

```
ctl_tx_da_ppp[47:0]
ctl_tx_sa_ppp[47:0]
ctl_tx_ether_type_ppp[15:0]
ctl_tx_opcode_ppp[15:0]
ctl_tx_pause_quanta0[15:0]
ctl_tx_pause_quanta1[15:0]
ctl_tx_pause_quanta2[15:0]
ctl_tx_pause_quanta3[15:0]
ctl_tx_pause_quanta4[15:0]
ctl_tx_pause_quanta5[15:0]
ctl_tx_pause_quanta6[15:0]
ctl_tx_pause_quanta7[15:0]
```

10G/25G Ethernet Subsystem 会自动计算 FCS 并将其添加到包中。对于优先暂停包，10G/25G Ethernet Subsystem 还会根据请求的优先级来自动生成启用矢量。

要请求暂停包，必须将 `ctl_tx_pause_req[8:0]` 和 `ctl_tx_pause_enable[8:0]` 总线的对应位设置为 1，并在暂停请求期间使其保持为 1 不变（即，如果这些输入设置为 0，则取消所有挂起的暂停包）。通过发出另一个暂停包并将其中包含的对应暂停量设为 0，即可取消暂停。10G/25G Ethernet Subsystem 将在当前进行中的包完成后立即发射暂停包。



**重要提示！** 此总线的每个位都必须保持稳定状态达至少 16 个周期，随后才能执行下一次转换。

为重新发射暂停包，10G/25G Ethernet Subsystem 包含总计 9 个独立定时器；对应每个优先暂停包 1 个定时器，对应全局暂停包 1 个定时器。这些定时器将加载对应输入总线的值。发射暂停包后，对应定时器将加载 `ctl_tx_pause_refresh_timer[8:0]` 输入总线的对应值。当定时器超时后，一旦当前进行中的包完成后就会立即发射对应该优先级的另一个包（或全局包）。此外，您可将定时器手动强制设为 0，因此，可针对某一时钟周期将 `ctl_tx_resend_pause` 输入强制设置为 1 来强制重新发射。

为了减少暂停包执行优先模式操作的次数，只要任一定时器超时，该暂停包对应的定时器就会被视为“已超时”。此外，等待当前进行中的包完成的过程中，只要任何新定时器发生超时，或者收到任何新请求，都将合并为单一暂停帧。例如，如有 2 个定时器正在倒计时，并且您针对第 3 个优先暂停包发送请求，则前 2 个定时器将被强制超时，并且发射当前进行中的包（如有）后就会立即发送一个对应全部 3 个优先级的暂停包。同样，如果前 2 个定时器中的任何一个定时器超时且未收到其它请求，则这 2 个定时器都会被强制超时，并且发射当前进行中的包（如有）后就会立即发送一个对应这 2 个优先级的暂停包。

您可通过将 `ctl_tx_pause_req[8:0]` 或 `ctl_tx_pause_enable[8:0]` 的相应位设置为 0 来停止生成暂停包。

## RX 暂停终止

10G/25G Ethernet Subsystem 去除了全局暂停帧和优先暂停帧，并提供了一个简单的握手接口以允许用户响应暂停包。

### 暂停包判定

暂停包判定分 3 个步骤：

1. 执行检查，确认包属于全局控制包还是优先控制包。  
仅当 `ctl_rx_forward_control` 设为 1 时才会将通过步骤 1 的包转发给您。
2. 如果通过步骤 1，则会检查该包以判定它是否属于全局暂停包。
3. 如果未通过步骤 2，则会检查该包以判定它是否属于优先暂停包。

对于步骤1，以下伪代码显示了检查功能：

```
assign da_match_gcp = (!ctl_rx_check_mcast_gcp && !ctl_rx_check_unicast_gcp)
|| ((DA
== ctl_rx_pause_da_unicast) && ctl_rx_check_unicast_gcp) || ((DA ==
48'h0180c2000001) &&
ctl_rx_check_mcast_gcp);
assign sa_match_gcp = !ctl_rx_check_sa_gcp || (SA == ctl_rx_pause_sa);
assign etype_match_gcp = !ctl_rx_check_etype_gcp || (ETYPE ==
ctl_rx_etype_gcp);
assign opcode_match_gcp = !ctl_rx_check_opcode_gcp || ((OPCODE >=
ctl_rx_opcode_min_gcp) && (OPCODE <= ctl_rx_opcode_max_gcp));
assign global_control_packet = da_match_gcp && sa_match_gcp &&
etype_match_gcp &&
opcode_match_gcp && ctl_rx_enable_gcp;
assign da_match_pcp = (!ctl_rx_check_mcast_pcp && !ctl_rx_check_unicast_pcp)
|| ((DA
== ctl_rx_pause_da_unicast) && ctl_rx_check_unicast_pcp) || ((DA ==
ctl_rx_pause_da_mcast) && ctl_rx_check_mcast_pcp);
assign sa_match_pcp = !ctl_rx_check_sa_pcp || (SA == ctl_rx_pause_sa);
assign etype_match_pcp = !ctl_rx_check_etype_pcp || (ETYPE ==
ctl_rx_etype_pcp);
assign opcode_match_pcp = !ctl_rx_check_opcode_pcp || ((OPCODE >=
ctl_rx_opcode_min_pcp) && (OPCODE <= ctl_rx_opcode_max_pcp));
assign priority_control_packet = da_match_pcp && sa_match_pcp &&
etype_match_pcp &&
opcode_match_pcp && ctl_rx_enable_pcp;
assign control_packet = global_control_packet || priority_control_packet;
```

其中 DA 表示目标地址，SA 表示源地址，OPCODE 为操作代码 (opcode)，而 ETYPE 则是从传入的包提取的“ethertype/length”（以太类型/长度）字段。

对于步骤2，以下伪代码显示了检查功能：

```
assign da_match_gpp = (!ctl_rx_check_mcast_gpp && !ctl_rx_check_unicast_gpp)
|| ((DA
== ctl_rx_pause_da_unicast) && ctl_rx_check_unicast_gpp) || ((DA ==
48'h0180c2000001) &&
ctl_rx_check_mcast_gpp);
assign sa_match_gpp = !ctl_rx_check_sa_gpp || (SA == ctl_rx_pause_sa);
assign etype_match_gpp = !ctl_rx_check_etype_gpp || (ETYPE ==
ctl_rx_etype_gpp);
assign opcode_match_gpp = !ctl_rx_check_opcode_gpp || (OPCODE ==
ctl_rx_opcode_gpp);
assign global_pause_packet = da_match_gpp && sa_match_gpp &&
etype_match_gpp &&
opcode_match_gpp && ctl_rx_enable_gpp;
```

其中 DA 表示目标地址，SA 表示源地址，OPCODE 为操作代码，而 ETYPE 则是从传入的包提取的“ethertype/length”字段。

对于步骤3，以下伪代码显示了检查功能：

```
assign da_match_ppp = (!ctl_rx_check_mcast_ppp && !ctl_rx_check_unicast_ppp)
|| ((DA
== ctl_rx_pause_da_unicast) && ctl_rx_check_unicast_ppp) || ((DA ==
ctl_rx_pause_da_mcast) && ctl_rx_check_mcast_ppp);
assign sa_match_ppp = !ctl_rx_check_sa_ppp || (SA == ctl_rx_pause_sa);
assign etype_match_ppp = !ctl_rx_check_etype_ppp || (ETYPE ==
ctl_rx_etype_ppp);
```

```
assign opcode_match_ppp = !ctl_rx_check_opcode_ppp || (OPCODE ==  
ctl_rx_opcode_ppp);  
assign priority_pause_packet = da_match_ppp && sa_match_ppp &&  
etype_match_ppp &&  
opcode_match_ppp && ctl_rx_enable_ppp;
```

其中 DA 表示目标地址，SA 表示源地址，OPCODE 为操作代码，而 ETYPY 则是从传入的包提取的“ethertype/length”字段。

## 用户接口

将使用简单的握手协议来告知您，已使用 `ctl_rx_pause_enable[8:0]`、`stat_rx_pause_req[8:0]` 和 `ctl_rx_pause_ack[8:0]` 总线接收到暂停包。对于这些总线，位 [8] 对应于全局暂停包，位 [7:0] 对应于优先暂停包。

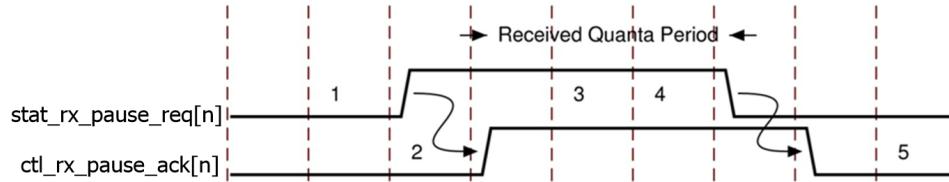
收到暂停包时，会发生以下步骤：

1. 如果 `ctl_rx_pause_enable[8:0]` 的对应位为 0，那么将忽略暂停量并且该核保持处于步骤 1 不变。否则，`stat_rx_pause_req[8:0]` 总线的对应位将设置为 1，并且接收到的暂停量将加载到定时器中。  
如果当暂停处理处于步骤 2 或后续阶段时，`ctl_rx_pause_enable[8:0]` 的任一位设置为 0（即禁用），那么该核会照常完成这些步骤，然后返回步骤 1。
2. 如果 `ctl_rx_check_ack` 输入为 1，该核会等待您将 `ctl_rx_pause_ack[8:0]` 总线的相应位设置为 1。
3. 当您将 `ctl_rx_pause_ack[8:0]` 的适当位设置为 1 之后，或者如果 `ctl_rx_check_ack` 为 0，那么该核会开始对定时器进行倒计时。
4. 当定时器超时时，该核会将 `stat_rx_pause_req[8:0]` 的相应位重新设置为 0。
5. 如果 `ctl_rx_check_ack` 输入为 1，则当您将 `ctl_rx_pause_ack[8:0]` 的相应位重新设为 0 时，此操作即告完成。

如果您不将 `ctl_rx_pause_ack[8:0]` 的相应位重新设为 0，那么该核会在 32 个时钟周期后将此操作视为完成。

下图中演示了上述步骤，每个步骤均显示在波形上。

图 33：RX 暂停接口示例



如果步骤 2 到 5 期间任意时间收到新的暂停包，那么定时器将加载新获取的暂停量值，并且此进程继续运行。

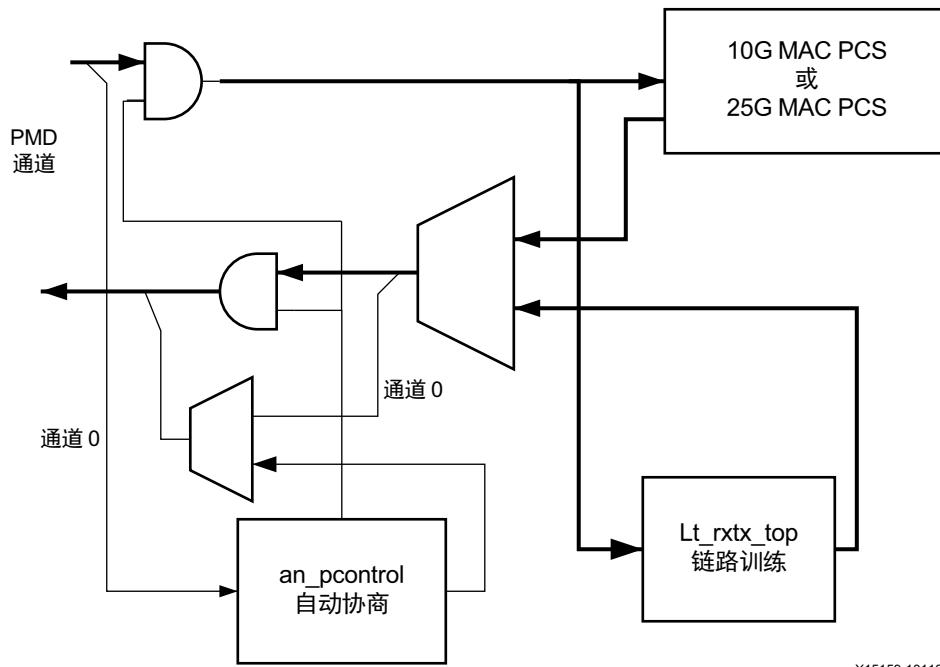
**注释：**在“Pause”定时器中指定的时间段内，发射器 MAC 在接收到来自接收器的暂停帧时不应发射帧/包。

# 自动协商

## 自动协商

下图显示了含自动协商 (AN) 和链路训练 (LT) 的 10G/25G Ethernet Subsystem 模块框图。

图 34：含自动协商和链路训练的核



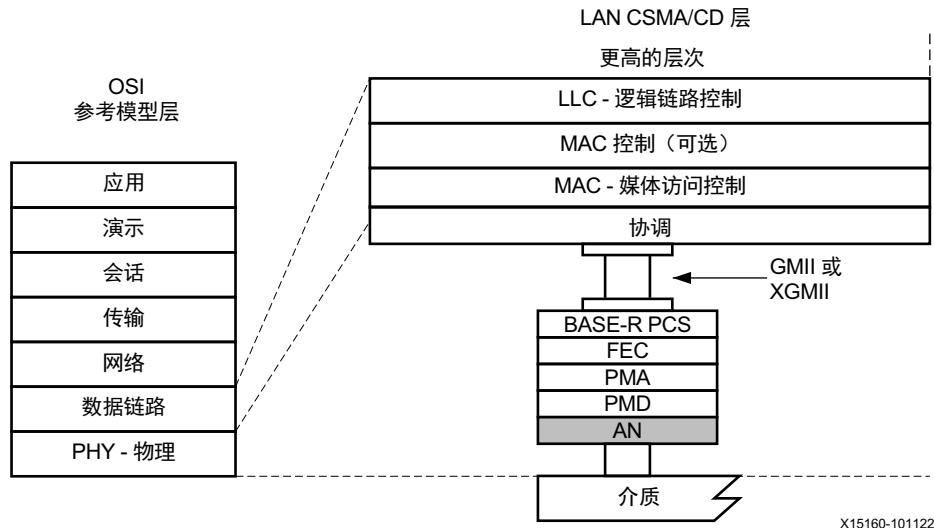
“Auto-Negotiation”（自动协商）功能允许以太网器件将其拥有的运行模式播发至位于背板以太网链路远端的其它器件，并检测其它器件可能正在播发的对应运行模式。此自动协商功能的目标是提供一条途径，以便在 2 个器件之间交换信息，并自动配置这些器件以充分利用其功能。它的另一个目标是支持数字信号检测以确保器件连接到链路伙伴，而不是检测因串扰而产生的信号。当自动协商完成后，将根据可用操作模式报告相应的功能。

如果链路两端都支持链路训练功能，则会在自动协商完成后执行链路训练。通常由于数字信号遍历背板可能导致产生频率相关损失，因而需要执行链路训练。该核所包含的链路训练块的主要功能是在背板链路的基础上提供寄存器信息和训练顺序，随后，收发器所包含的接收端电路将对这些信息进行分析。链路训练块的另一项功能是将训练反馈从接收器传递到对应发射器，以便根据需要对其均衡器电路（作为收发器的一部分）进行调整。在此核中不包含决策制定算法。当自动协商和链路训练完成后，数据路径将切换至任务模式 (PCS)，如前图所示。

## 概述

下图显示了 OSI 参考模型中自动协商功能的位置。

图 35：OSI 模型中的自动协商功能



Auto-Negotiation Intellectual Property Core (ANIPC) 满足 IEEE 802.3-2015 标准 802.3-2015 第 73 条中指定的要求，包括 IEEE P802.3ba 和 802.3ap 标准中指定的修订内的相关要求。ANIPC 核的功能列在第 73 条中，具体来说，这些功能列在第 73.10.4 节“状态图”图 73-11 的“仲裁状态图”中。

正常任务模式操作期间，通过将链路控制输出设置为 11（二进制），收发器输入和输出的位工作频率通常为 10.3125 或 25.78125 Gb/s。但自动协商期间通道上所使用的双曼彻斯特编码 (Dual Manchester Encoding, DME) 比特率与任务模式操作期间所使用的比特率不同。为满足此要求，ANIPC 核使用过采样和过驱动来使 156.25 Mb/s 自动协商速度 (DME 时钟频率 312.5 MHz) 与任务模式 10.3125 或 25.78125 Gb/s 物理线速相匹配。

## 功能描述

### autoneg\_enable

当 autoneg\_enable 输入信号设为 1 时，自动协商在下列情况下自动开始执行：在上电时、丢失载波信号时，或者 restart\_negotiation 输入信号从 0 到 1 循环时。所有“功能”输入信号以及 2 个输入信号 (PAUSE 和 ASM\_DIR) 均绑定到低电平或高电平，以便指示硬件功能。针对自动协商器的每个实例，nonce\_seed[7:0] 输入都必须设置为唯一非零值。此操作的重要意义在于，保证在上电时不会发生死锁。如果连接在一起的 2 个链路伙伴在其 nonce\_seed[7:0] 输入设置为相同值的情况下尝试自动协商，那么自动协商将连续不断失败。pseudo\_sel 输入是一种任意选择方式，用于选择随机位生成器的多项式，此生成器即自动协商期间所用 DME 页面的位元位置 49 中使用的随机位生成器。此输入的任何选择均有效，并且不会导致有害行为。

### 链路控制

当自动协商开始时，将根据各链路的对应功能 (Ability) 输入的处理方法来激活各链路控制信号。因此，ANIPC 硬件将监控对应链路状态信号，以便确认已连接的各链路状态。如果特定链路未使用，则对应链路控制输出处于未连接状态，且对应链路状态输入应绑定到低电平。在此期间，ANIPC 硬件会与链路伙伴建立通信链路，并使用此链路来协商连接的功能。

### 自动协商完成

当自动协商完成时，`autoneg_complete` 输出信号将断言有效。此外，如果要使用前向纠错硬件，则会断言 `an_fec_enable` 输出信号有效；如果允许发射器硬件生成 PAUSE 控制包，则会断言 `tx_pause_en` 输出信号有效，如果允许接收器硬件检测 PAUSE 控制包，并且所选链路的输出链路控制设置为其任务模式值 11（二进制），则会断言 `rx_pause_en` 输出信号有效。



**重要提示！** 接收到来自 PCS 的 `rx_status` 之后，才会断言 `autoneg_complete` 信号有效。这意味着，如果包含链路训练，则完成链路训练并且 `rx_status` 为高电平后才会断言 `autoneg_complete` 输出信号有效。

## 链路训练

当自动协商集中到背板或者铜工艺后，才会执行链路训练。工艺选择也可以采用手动输入或并行检测方式来完成。由于频率相关损失可能导致需要进行链路培训，而当数字信号遍历背板或铜缆时则可能会发生此类损失。链路训练核的主要功能是在背板链路的基础上提供寄存器信息和训练顺序，随后，该核范围外的接收端电路将对这些信息进行分析。

该核的另一项功能是将训练反馈从接收器传递到对应发射器，以便根据需要对其均衡器电路（不包含在此核中）进行调整。组成该核的 2 条电路分别是链路训练接收块和链路训练发射块。

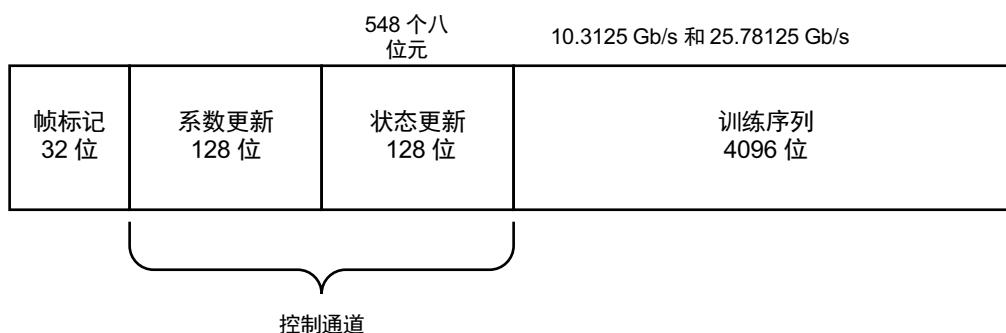


**重要提示！** 负责发射器预加重抽头的调整工作的逻辑必须在此 IP 核外部提供。

### 发射

链路训练发射块会构造 1 个 4,384 位帧，其中包含帧定界符、控制通道和链路训练顺序。其格式如下图所示。

图 36：链路训练帧结构



X15161-101122

赛灵思建议，在发射状态机发射控制通道位的过程中，不要通过链路训练算法来更改这些位，否则接收到的位可能会有错误，从而可能导致 DME 错误。当 `tx_SOF` 断言有效时开始发射，至少达到 288 位时间（或者约 30 ns）后停止发射。

虽然系数和状态的线速率包含 128 位时间，但这 2 个字段的实际信号处理速率将减去 8 的倍数。因此，DME 时钟速率为线速率的四分之一。

## 帧标记

帧标记包含 16 个连续的 1 后接 16 个连续的 0。在该帧剩余部分中，不会重复此模式。

## 系数和状态

由于这 2 个字段的 DME 信号处理速率减去 8 的倍数，因此每个系数和状态发射都包含  $128/8 = 16$  位，每个位均按 15:0 的顺序编号。以下几个表按这些位的发射顺序（从位 15 开始到位 0 结束）定义这些位。

表 295：系数和更新字段位定义

位	名称	描述
15:14	保留	作为 0 发射，接收时忽略。
13	Preset	1 = 预置系数 0 = 正常操作
12	Initialize	1 = 初始化系数 0 = 正常操作
11:6	保留	作为 0 发射，接收时忽略。
5:4	Coefficient (+1) update	1 1 = 保留 0 1 = 递增 1 0 = 递减 0 0 = 保持
3:2	Coefficient (0) update	1 1 = 保留 0 1 = 递增 1 0 = 递减 0 0 = 保持
1:0	Coefficient (-1) update	1 1 = 保留 0 1 = 递增 1 0 = 递减 0 0 = 保持

表 296：状态报告字段位定义

位	名称	描述
15	Receiver ready	1 = 局部接收器已判定训练完成，并且已准备好接收数据。 0 = 局部接收器正在请求训练继续。
14:6	保留	作为 0 发射，接收时忽略。
5:4	Coefficient (+1) update	1 1 = 最大值 0 1 = 已更新 1 0 = 最小值 0 0 = 未更新 (not_updated)
3:2	Coefficient (0) update	1 1 = 最大值 0 1 = 已更新 1 0 = 最小值 0 0 = 未更新 (not_updated)

表 296：状态报告字段位定义 (续)

位	名称	描述
1:0	Coefficient (-1) update	1 1 = 最大值 0 1 = 已更新 1 0 = 最小值 0 0 = 未更新 (not_updated)

IEEE 802.3 标准第 72 条中定义了每个位的功能。其用途是在处理链路训练期间传递发射均衡器的调整方式。[自动协商端口](#)中定义了对应的信号名称。

## 训练顺序

训练顺序由伪随机位元序列 (PRBS) 组成，此序列包含 4,094 位后接 2 个 0，共计 4,096 位。PRBS 以 10.3125 或 25.78125 Gb/s 的线速率进行发射。PRBS 生成器可接收外部来源的 11 位种子。加载初始种子后，PRBS 生成器会继续运行，无需进一步干预。PRBS 生成器通过对应于如下多项式的电路来实现：

$$G(x) = 1 + x^9 + x^{11}$$

## 接收

接收块用于实现 IEEE 802.3 标准第 72 条的图 72-4 中所示的帧对齐状态图。

## 帧锁定状态机

帧锁定状态机用于搜索帧标记，帧标记包含 16 个连续的 1 后接 16 个连续的 0。在 IEEE 802.3 标准第 72 条的图 72-4 中详细指定了该功能。达成帧锁定时，`frame_lock` 的值设置为 TRUE。

## 已接收的数据

接收器用于输出控制通道，其中包含 [发射](#) 中定义的位定义和 [端口描述 - MAC+PCS 变体](#) 中定义的信号名称。

如果接收特定 DME 帧期间发生 DME 错误，则不会更新控制通道输出，但会保留最后一个接收到的有效 DME 帧的值并在接收到下一个有效 DME 帧时进行更新。

**注释：**虽然子系统原生支持训练协议 (Training Protocol)，但并不存在任何逻辑用于根据接收到的信号质量分析来控制远端发射器调整。因为通过广泛测试发现，并无此必要。但在子系统上提供了训练接口，以支持访问所有子系统寄存器和收发器上的 DRP 端口。如果需要，您可使用此接口来实现自己的 10GBASE-KR 训练算法。

## 使用 AXI4-Lite 接口进行自动协商和链路训练的开发板测试步骤

- 根据要播发的核配置或功能，启用功能 (Abilities) 寄存器 CONFIGURATION\_AN\_ABILITY (0x00F8)。例如，将值 0x1E06 写入地址 0x00F8。
- 读取 CONFIGURATION\_AN\_CONTROL\_REG1 寄存器 (0x00E0)，根据需求，可启用或绕过自动协商。如已启用自动协商，则需将随机数种子值写入相同寄存器。例如，将值 0x16D 写入地址 0x00E0。
- 在 CONFIGURATION\_LT\_CONTROL\_REG1 (0x0100) 寄存器中设置链路训练使能信号，以启用“Link Training”(链路训练) 选项。例如，将值 0x1 写入地址 0x0100。
- 将 1 写入 CONFIGURATION\_LT\_TRAINED\_REG 寄存器 (0x0104) 使 `ctl_lt_rx_trained` 转至高位。这表示接收器有限脉冲响应 (FIR) 滤波器系数已全部设置完成，并且训练的接收器部分已完成。

5. 将部分种子值写入 CONFIGURATION\_LT\_SEED\_REG0 寄存器 (0x0110)。例如，将值 0x0605 写入地址 0x0110。
6. 将部分系数值（用于占位符逻辑）写入 CONFIGURATION\_LT\_COEFFICIENT\_REG0 寄存器 (0x0130)。例如，将值 0x540 写入地址 0x0130。
7. 向 `ctl_an_reset` (0x0004 地址的位 28) 发出 `sys_reset` 或写入 `1'b1`，以便自动协商块将更新后的随机数种子值纳入考量范围。
8. 保持读取 `stat_an_autoneg_complete` (地址 0x0458 的位 2)，它可指示自动协商和链路训练是否成功完成。

如果链路伙伴发送下一页信号，那么必须设置 `ctl_an_lo_np_ack` 信号。此端口可绑定到高电平。

# 设计流程步骤

本章节描述了子系统的自定义和生成方式、子系统的约束方式以及此 IP 子系统的仿真、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](#))

---

## 自定义和生成子系统

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 中自定义和生成子系统，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。

您可以使用以下步骤通过指定与核关联的各种参数值来对设计中要使用的 IP 进行自定义：

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](#))。

**注释：**本章中的附图是 Vivado 集成设计环境 (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。

### “Configuration” 选项卡

“Configuration”（配置）选项卡提供了基本核配置选项。所有字段中的默认值均已预填充。

图37：“Configuration”选项卡 (Versal)

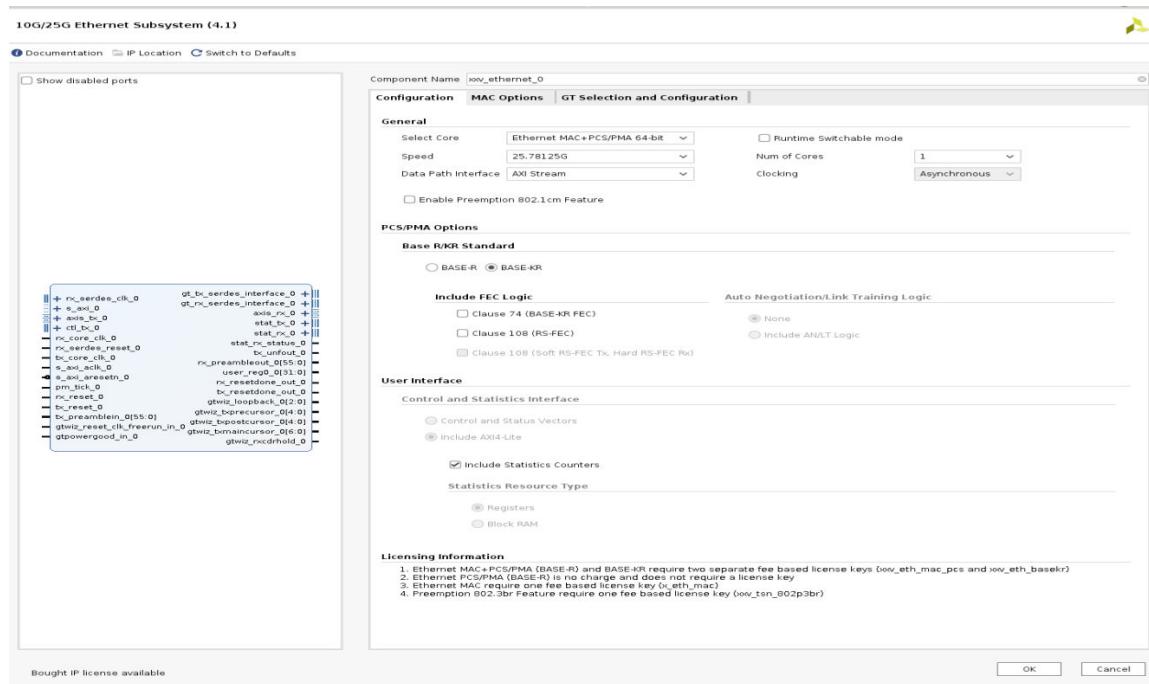


图38：“Configuration”选项卡 (UltraScale/UltraScale+)

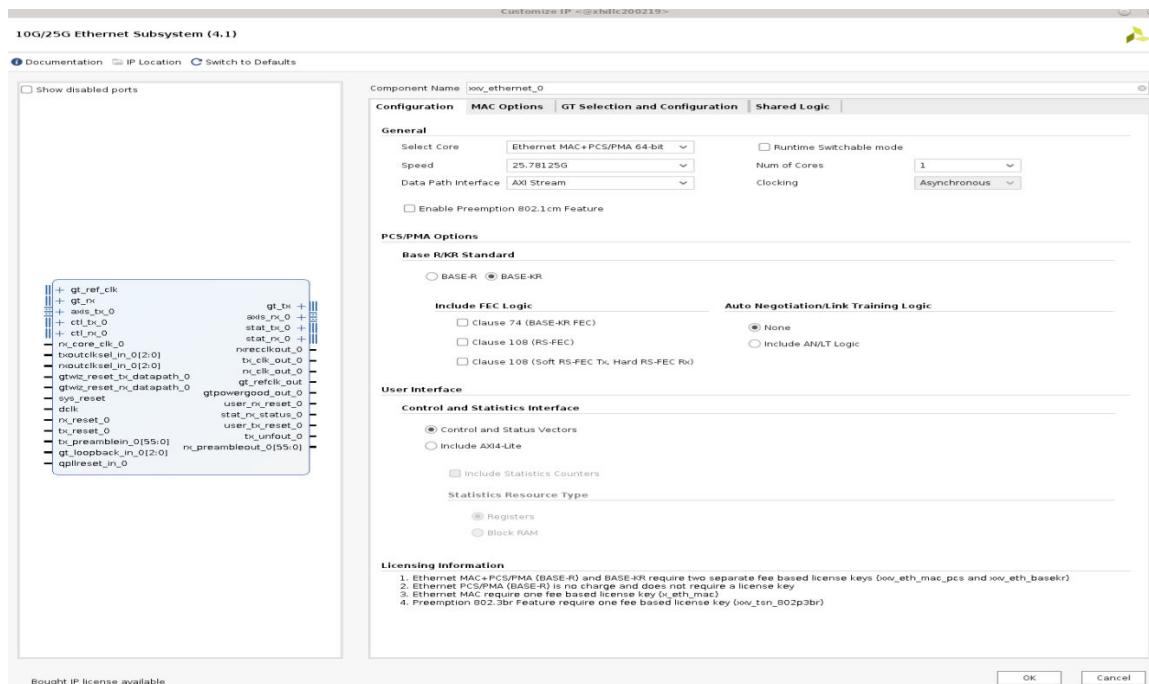


表 297：配置选项

选项	值	默认
<b>General</b>		
Select Core	Ethernet MAC 64-bit Ethernet PCS/PMA 32-bit Ethernet PCS/PMA 64-bit Ethernet MAC+PCS/PMA 32-bit Ethernet MAC+PCS/PMA 64-bit	Ethernet MAC+PCS/PMA 64-bit
Speed	25.7812G 10.3125G	25.7812G
Runtime Switchable Mode	0 和 1	0
Num of Cores <sup>9</sup>	1 2 3 4	1
Clocking	Asynchronous	Asynchronous
Data Path Interface	AXI4-Stream <sup>1</sup> Media Independent Interface (MII) <sup>2</sup>	AXI4-Stream
<b>PCS/PMA 32-bit</b>		
Enable Preemption 802.3br Feature	0 和 1	0
<b>PCS/PMA Options</b>		
Base-R Base-KR	Base-R Base-KR	Base-KR
<b>Include FEC Logic</b>		
Clause 74 (BASE-KR FEC) <sup>3 6</sup>	0 和 1	0
Clause 108 (RS-FEC) <sup>4</sup>	0 和 1	0
Clause 108 (Soft RS-FEC Tx, Hard RS-FEC Rx) <sup>8</sup>	0 和 1	0
<b>Auto-Negotiation/Link Training Logic</b>		
Auto-Negotiation/Link Training Logic <sup>10</sup>	无 Include AN/LT Logic	无
<b>Control and Statistics Interface</b>		
Control and Statistics interface	Control and Status Vectors Include AXI4-Lite	Control and Status Vectors
Include Statistics Counters <sup>5</sup>	0 和 1	1

表 297：配置选项（续）

选项	值	默认
Statistics Resource Type <sup>7</sup>	Registers, Block RAM	Register

**注释：**

1. AXI4-Stream 接口是对 Ethernet MAC+PCS/PMA 和独立 Ethernet MAC 核可见的唯一选项。
2. MII 接口是对以太网 PCS/PMA 核可见的唯一选项。
3. 针对 Base-R 不支持 “Clause 74 (BASE-KR FEC)” 逻辑。
4. 针对 Base-R 10G 速度不支持 “Clause 108 (RS-FEC)”。
5. 仅当启用 “Include Statistics Counters”（包含统计数据计数器）选项时，在寄存器映射中 “Statistics Counters”（统计数据计数器）才可用。否则，“Statistics Counters” 将不可用。
6. “Clause 74 (BASE-KR FEC)” 和 “Clause 108 (RS-FEC)” 在 Vivado IDE 中均为可选，但在正常操作期间每次只能使用相应的控制信号启用其中之一。
7. “Statistics Resource Type block RAM”（统计数据资源类型块 RAM）选项将在后续版本中提供。
8. “Soft RS-FEC TX, Hard RS-FEC RX” 选项，用于通过利用 UltraScale+ 器件中的 CMAC 块中存在的嵌入式 100G RS-FEC 功能来降低逻辑利用率。
9. 仅限针对 Versal® 器件，“Number of Cores” 才为 1，不支持多核，因为 GT 将始终位于设计示例内/核外部。
10. Versal 器件系列不支持自动协商和链路训练。

## “MAC Options” 选项卡

“MAC Options” 选项卡提供额外的核配置选项。

图 39：“MAC Options” 选项卡 (Versal)

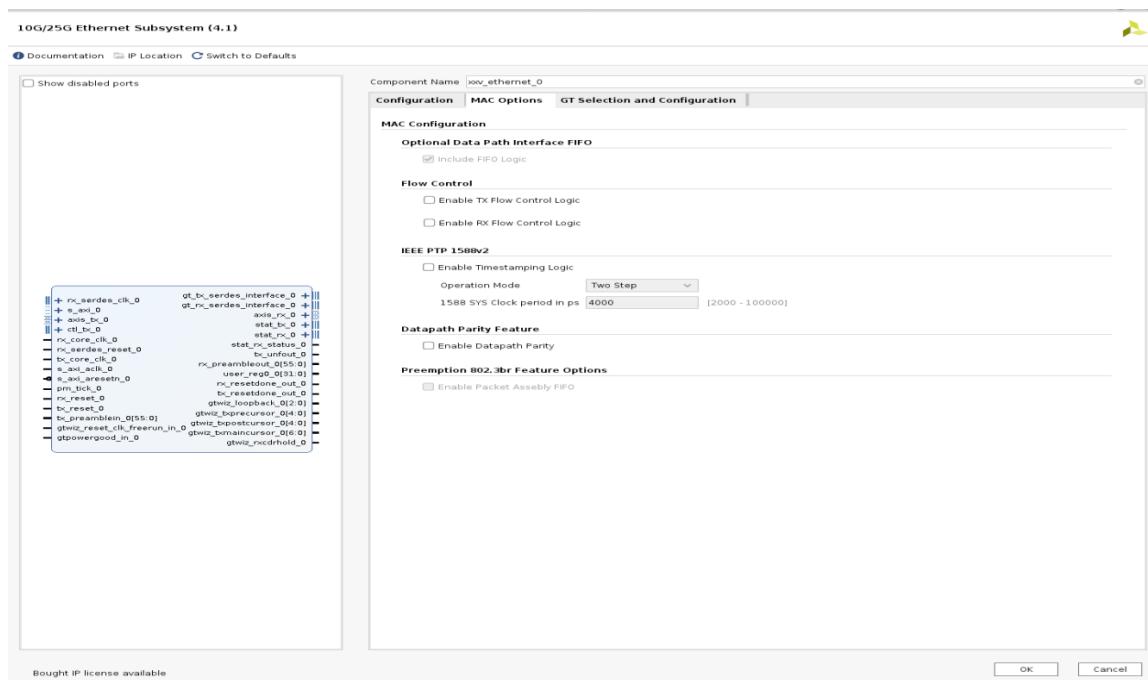


图 40：“MAC Options” 选项卡 (UltraScale/UltraScale+)

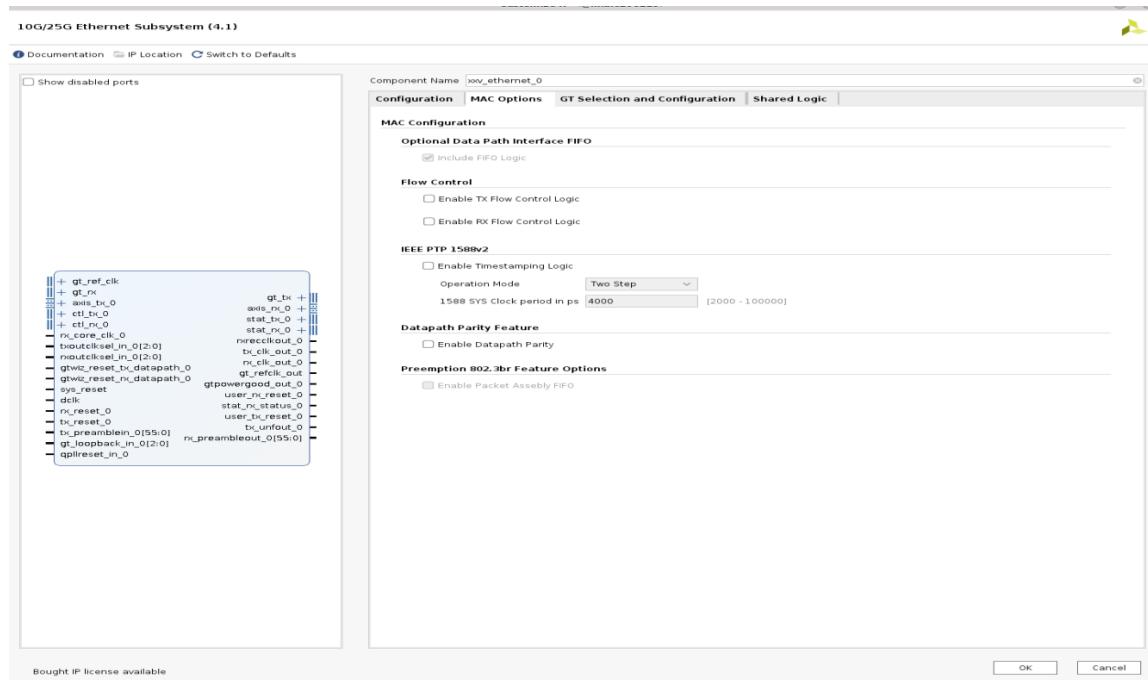


表 298：MAC 选项

选项	值	默认
可选数据路径接口 FIFO		
Include FIFO Logic <sup>1</sup>	勾选或不勾选	勾选
<b>Flow Control</b>		
Enable TX Flow Control Logic	勾选或不勾选	不勾选
Enable RX Flow Control Logic	勾选或不勾选	不勾选
IEEE PTP 1588v2		
Enable Timestamping Logic	勾选或不勾选	不勾选
Operation Mode	One Step Two Step	Two Step
1588 SYS Clock period in ps	2000 - 100000	4000 <sup>2</sup>
Enable Datapath Parity	0 和 1	0
Enable Packet Assembly FIFO	0 和 1	0

**注释：**

- 仅当 Base-R/BaseKR 标准设为 Base-R 时，该选项才可用，如果不勾选该选项，则核将处于低时延模式。
- 4000（针对 Versal 器件），10000（针对非 Versal 器件）。

## “GT Selection and Configuration” 选项卡

“GT Selection and Configuration” 选项卡支持您配置核的串行收发器功能。

图 41：“GT Selection and Configuration” 选项卡 (Versal)

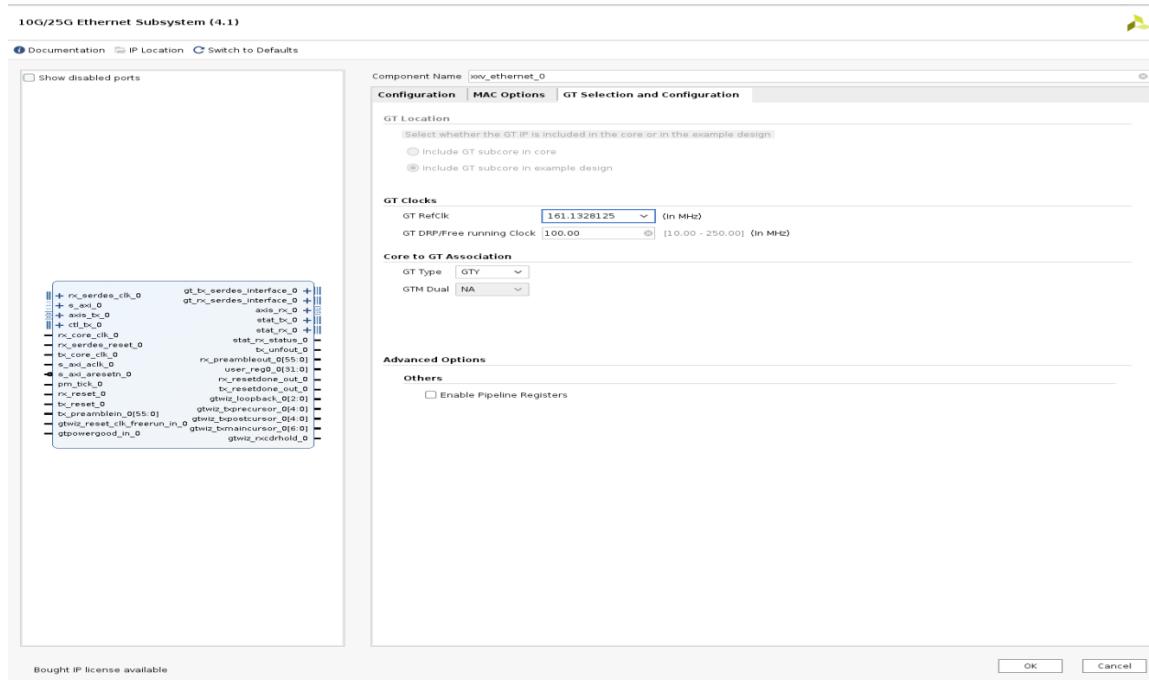


图 42：“GT Selection and Configuration” 选项卡 (UltraScale/UltraScale+)

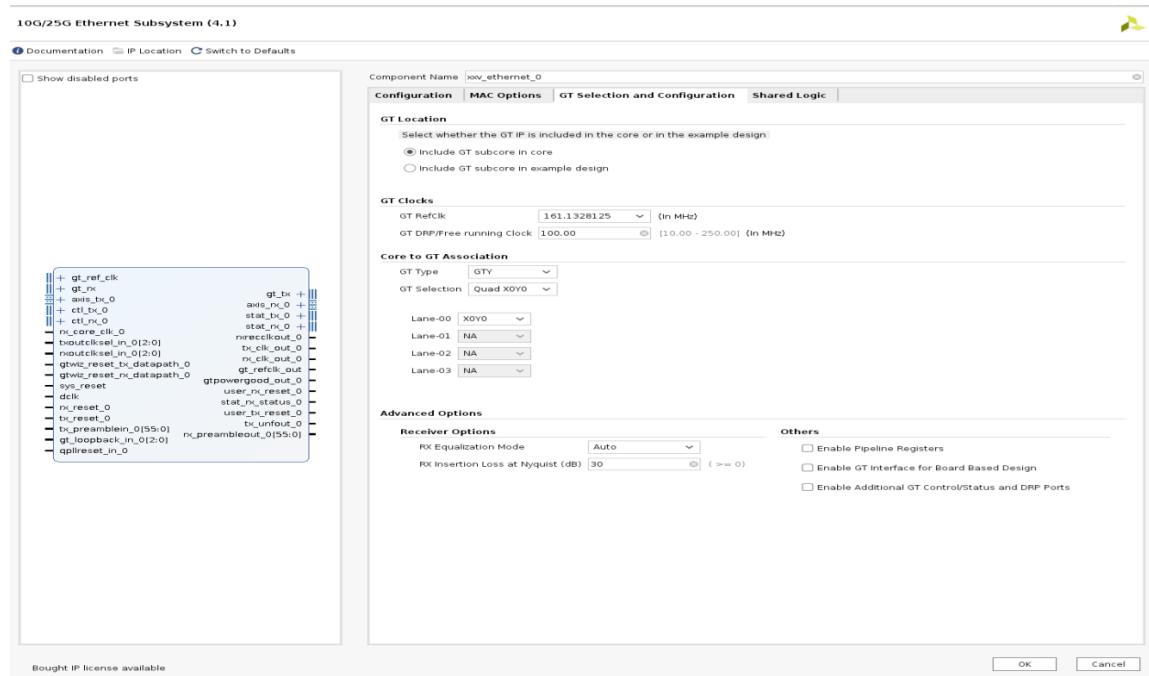


表 299：GT 时钟选项

选项	值	默认
<b>GT Location</b>		
选择将 GT IP 包含在核中还是包含在设计示例中	Include GT subcore in core Include GT subcore in example design	Include GT subcore in core
<b>GT Clocks<sup>2</sup></b>		
GT RefClk (MHz) <sup>1</sup>	161.1328125 195.3125 156.25 201.4160156 257.8125 322.265625 312.5 103.125 128.90625 206.25 309.375	161.1328125
GT DRP Clock (MHz)	10 - 250 MHz	100.00
<b>Core to GT Association</b>		
GT Type	GTY GTH GTM GTYP	GTY
GT Selection	基于器件/封装四通道组的选项。 例如： Quad X0Y1 Quad X0Y2 Quad X0Y3 ...	Quad X0Y1
Lane-00 到 Lane-03	根据器件/封装自动填充。 例如，如果 Num of Core = 4 且 GT Selection = Quad X0Y1，则 4 条通道分别为： X0Y4 X0Y5 X0Y6 X0Y7	
RX Equalization Mode	Auto LPM DFE	Auto
RX Insertion Loss at Nyquist (dB)	取决于 GT Wizard	30
<b>其它</b>		
Enable Pipeline Registers	勾选或不勾选	不勾选
Enable GT Interface for Board Based Design <sup>3</sup>	勾选或不勾选	不勾选

表 299：GT 时钟选项（续）

选项	值	默认
Enable Additional GT Control/Status and DRP Ports	勾选或不勾选	不勾选

**注释：**

1. 此列表提供用于默认配置的频率。如需获取不同速度所支持的时钟频率的完整列表，请参阅最新版工具中的 Vivado IDE。
2. 以下是支持各器件的各 GT 时钟：
  - a. 10G Versal 器件支持 128.90625、156.25、161.1328125、206.25、257.8125、312.5 和 322.265625。
  - b. 25G Versal 器件支持 161.1328125、103.125、128.90625、156.25、206.25、257.8125、309.375、312.5 和 322.265625。
  - c. 10G 非 Versal 器件支持 156.25、103.125、128.90625、161.1328125、206.25、257.8125、309.375、312.5 和 322.265625。
  - d. 25G 非 Versal 器件支持 161.1328125、156.25、195.3125、201.4160156、257.8125、312.5 和 322.265625。
  - e. 运行时切换功能支持的 `gt_refclk` 频率包括 128.90625、206.25、257.8125、322.265625、156.25、161.1328125 和 312.5（针对 Versal 器件），支持的 `gt_refclk` 频率包括 161.1328125、156.25 和 312.5（针对非 Versal 器件）。
3. 选中该选项时，根据 GT 标准接口，所有 GT 端口都将作为单一总线接口来使用。该选项将用于基于开发板的设计，也适合基于 IP integrator 的自动连接流程。仅当 IP 核内部存在 GT 子核时，该选项才可用。

## “Shared Logic” 选项卡

“Shared Logic” 选项卡可让您在核或设计示例中使用共享逻辑。

**注释：** “Shared Logic”（共享逻辑）选项卡仅可用于 UltraScale/UltraScale+ 器件。

图 43： “Shared Logic” 选项卡 (UltraScale/UltraScale+)

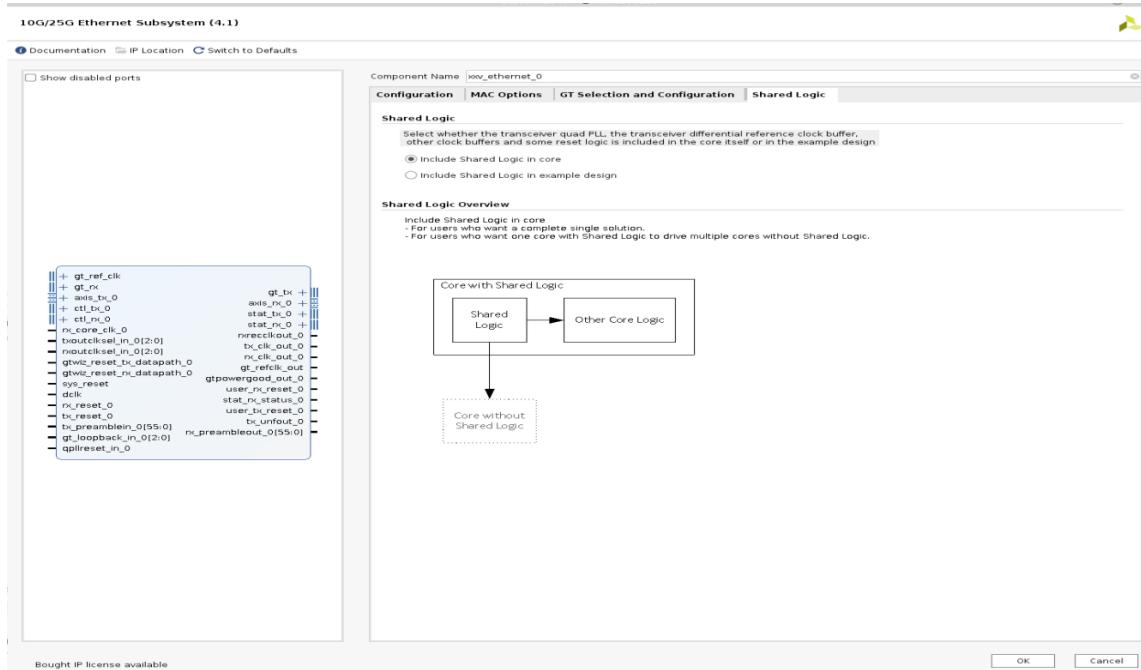


表 300：“Shared Logic”（共享逻辑）选项

选项	默认
Include Shared Logic in core	Include Shared Logic in core
Include Shared Logic in example design	

## 输出生成

如需了解更多详情，请参阅《Vivado Design Suite 用户指南：采用 IP 进行设计》([UG896](#))。

---

# 约束子系统

本节包含与 Vivado Design Suite 中核的约束有关的信息。

### 所需约束

10G/25G High Speed Ethernet Subsystem 要求时序规范和其它物理实现约束能够满足指定的性能要求。这些约束通过赛灵思器件约束(XDC)文件来提供。生成的 XDC 中的管脚分配和层级名称对应于所提供的 10G/25G High Speed Ethernet Subsystem 的设计示例。

为了达成一致的实现结果，通过赛灵思设计工具运行设计时，必须使用包含这些未经修改的原始约束的 XDC。如需获取有关 XDC 特定约束的定义及其使用方式的更多详细信息，请参阅《Vivado Design Suite 用户指南：使用约束》([UG903](#))。

10G/25G High Speed Ethernet Subsystem 中提供的约束已通过实现得到验证，并提供一致结果。约束可修改，但前提是充分了解每个约束的影响。

### 器件、封装和速度等级选择

本节不适用于此 IP 子系统。

### 时钟频率

本节不适用于此子系统。

### 时钟管理

本节不适用于此 IP 子系统。

### 时钟布局

本节不适用于此子系统。

### bank 分配

本节不适用于此 IP 子系统。

## 收发器布局

本节不适用于此 IP 子系统。

## I/O 标准与布局

本节不适用于此 IP 子系统。

---

# 仿真

有关 Vivado 仿真组件的全面信息，以及与如何使用支持的第三方工具相关的信息，请参阅《Vivado Design Suite 用户指南：逻辑仿真》(UG900)。

## 仿真加速

设计示例包含等待定时器。``define SIM_SPEED_UP` 可用于通过缩短这些等待时间来缩短仿真时间。`SIM_SPEED_UP` 仅在运行 RTL 仿真时才可用。它在使用综合后或实现后网表运行仿真时不可用。

### VCS

使用 vlogan 选项：`+define+SIM_SPEED_UP`。

### ModelSim

使用 vlog 选项：`+define+SIM_SPEED_UP`。

### Vivado 仿真器

使用 xvlog 选项：`-d SIM_SPEED_UP`。

### Questa Advanced Simulator

使用 vlog 选项：`+define+SIM_SPEED_UP`。

### Riviera Pro 仿真器

使用 vlog 选项：`+define+SIM_SPEED_UP`。

### Xcelium Parallel Simulator

使用 xmvlog 选项：`+define+SIM_SPEED_UP`。

**注释：**该核必须采用满量程 (fs) 分辨率进行仿真。

### 启用 RS-FEC 的配置仿真

为加速仿真，可应用 `SIM_SPEED_UP` 并在“Settings”（设置）窗口中取消选中“Use Precompiled IP simulation libraries”（使用预编译的 IP 仿真库）复选框，如下图所示。如不执行此操作，仿真可能长时间运行并导致超时错误。

图 44：“Use Precompiled IP Simulation Libraries” 已禁用

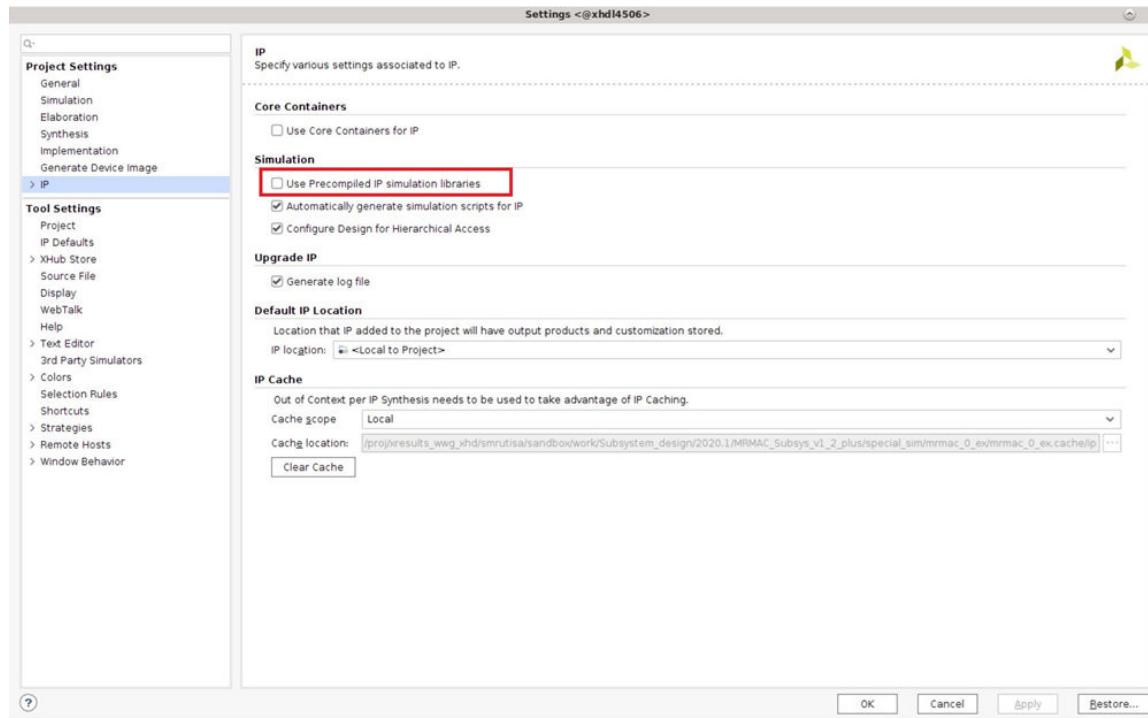
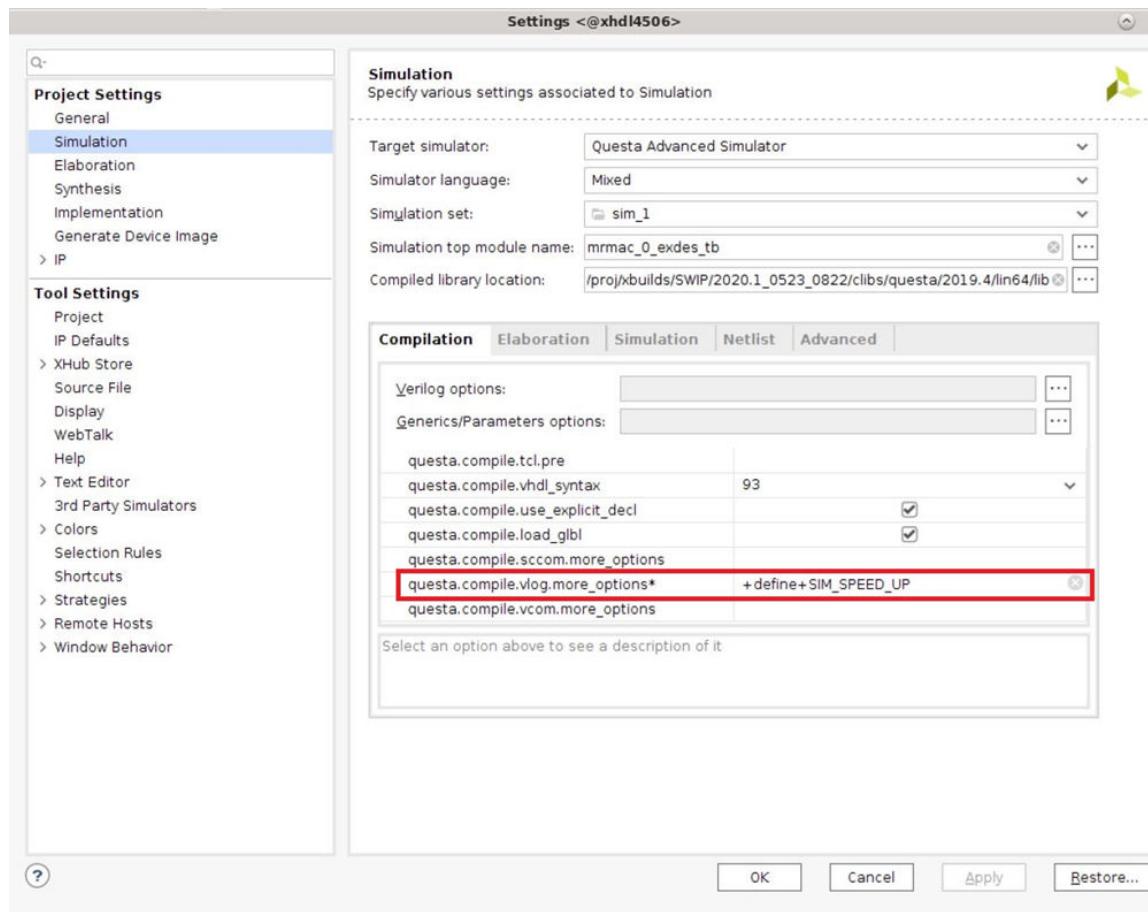


图 45: SIM\_SPEED\_UP 已启用



## 综合与实现

如需了解有关综合和实现方面的详情，请参阅《Vivado Design Suite 用户指南：采用 IP 进行设计》([UG896](#))。

# 设计示例

本章包含有关使用 Vivado 集成设计环境 (IDE) 时，Vivado® Design Suite 中提供的设计示例的信息。

---

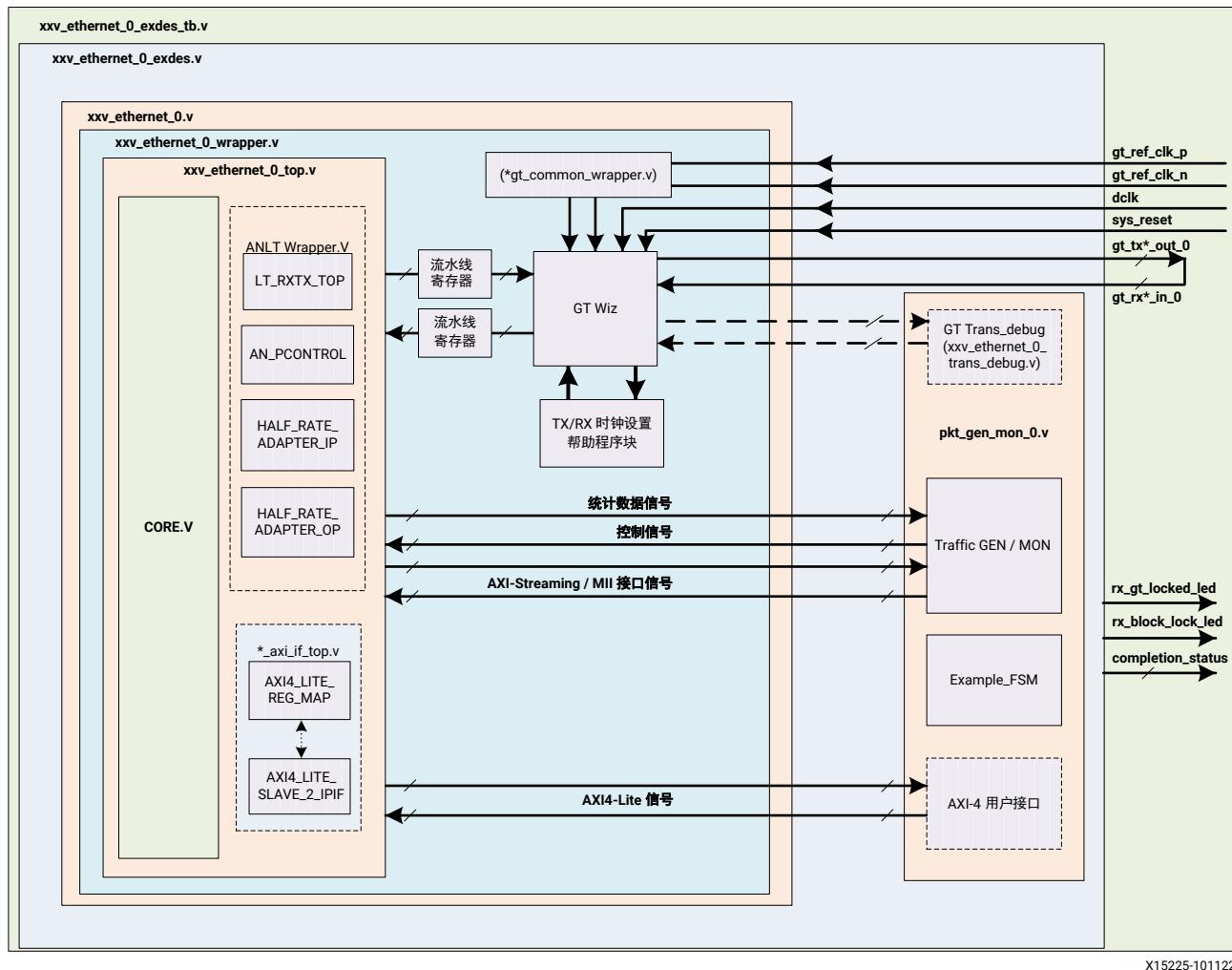
## 概述

下图显示了当 GT（串行收发器）位于 IP 核内时，xxv\_ethernet\_0 单核配置设计示例的各个模块的例化及其层级。  
(串行收发器将始终包含在 Versal® ACAP 设计示例内)。

同步寄存器和流水线寄存器用于在核与 GT 之间进行数据同步。

时钟帮助程序块用于为该核生成所需时钟频率。

图 46：单核设计示例层级



X15225-101122

以下用户接口可用于不同配置。

- MAC/PCS 配置：
  - AXI4-Stream，用于数据路径接口
  - AXI4-Lite，用于控制接口和统计数据接口
- PCS 配置：
  - MII，用于数据路径接口
  - AXI4-Lite，用于控制接口和统计数据接口

**xxv\_ethernet\_0\_pkt\_gen\_mon** 模块用于生成数据包以进行完整性测试。包生成和检查由 FSM 模块来控制。

可选模块如下所述：

- xxv\_ethernet\_0\_trans\_debug：在下列情况下，设计示例中存在此模块：在 Vivado IDE 中从“[GT Selection and Configuration”选项卡](#) 启用“Additional GT Control and Status Ports”（其它 GT 控制和状态端口）复选框、在“GT Selection and Configuration”（GT 选择和配置）选项卡下启用“Include GT subcore in example design”（在设计示例中包含 GT 子核）选项，或者在“Configuration”（配置）选项卡下启用“Runtime Switchable mode”（运行时可切换模式）选项。此模块从 xxv\_ethernet 核中引出所有 GT 通道 DRP 端口以及收发器模块的部分控制端口和状态端口。
- “Retiming registers”（重定时寄存器）：在“[GT Selection and Configuration”选项卡](#) 中选中“Enable Retiming Register”（启用重定时寄存器）选项时，将在核与 GT 之间包含一个单阶流水线寄存器以简化时序，其中使用 gt\_txusrcclk2 表示 TX 路径，使用 gt\_rxusrcclk2 表示 RX 路径。但默认情况下，针对 GT 与核之间的信号完成二阶寄存。
- “TX / RX Sync register”（TX / RX 同步寄存器）：TX 同步寄存器用于对从核到 GT 之间的 tx\_clk 相关数据进行双重同步。RX 同步寄存器则用于对从 GT 到核之间的 rx\_serdes\_clk 相关数据进行双重同步。

**注释：**对于 Runtime Switchable，如果在 Vivado IDE 中选中自动协商/链路训练，那么在开关期间将仅以 25G 数据速率执行 AN 操作，并在任务模式下执行 LT 操作。

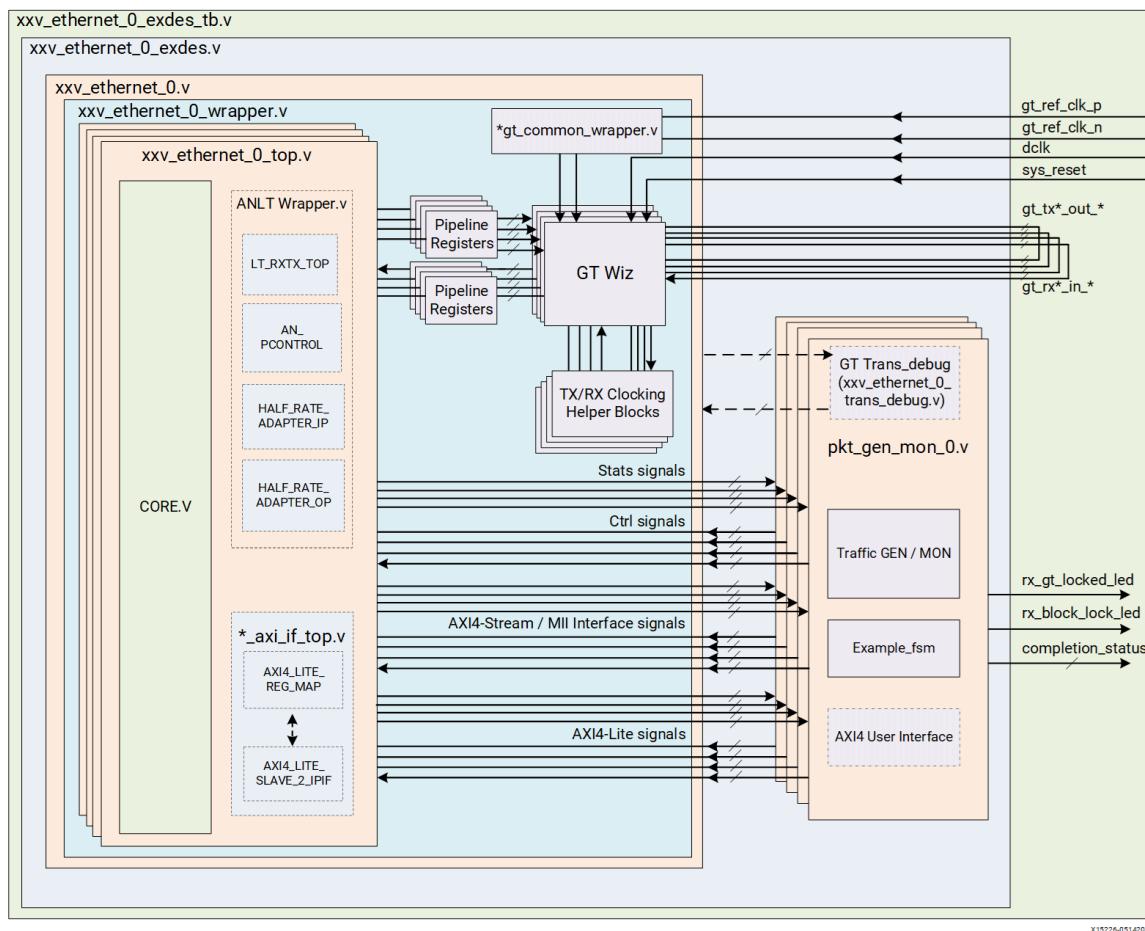
**注释：**如果在 Vivado IDE 中选中自动协商/链路训练，并且核数  $\geq 3$ ，则必须为 anlt\_wrapper 应用 Pblock 约束。Pblock 应布局到选定的收发器 (GT) 附近，并且大小应足以满足 anlt\_wrapper 的使用需求。如需了解更多信息，请参阅 example\_top.xdc。以下 xcvu095-ffva2104-2-e 器件示例中，已选中 4 个核且收发器为 x0y4 到 x0y7。

示例：

```
create_pblock pblock_ANLT
add_cells_to_pblock [get_pblocks pblock_ANLT] [get_cells -quiet [list DUT/
inst/
i_*_top_0/i_*_*_ANLT_WRAPPER DUT/inst/i_*_top_1/i_*_*_ANLT_WRAPPER DUT/inst/
i_*_*_top_2/i_*_*_ANLT_WRAPPER DUT/inst/i_*_top_3/
i_*_ANLT_WRAPPER]] resize_pblock
[get_pblocks pblock_ANLT] -add {SLICE_X0Y5:SLICE_X40Y180}
```

下图显示了 xxv\_ethernet\_0 多核配置设计示例的各个模块的例化及其层级。

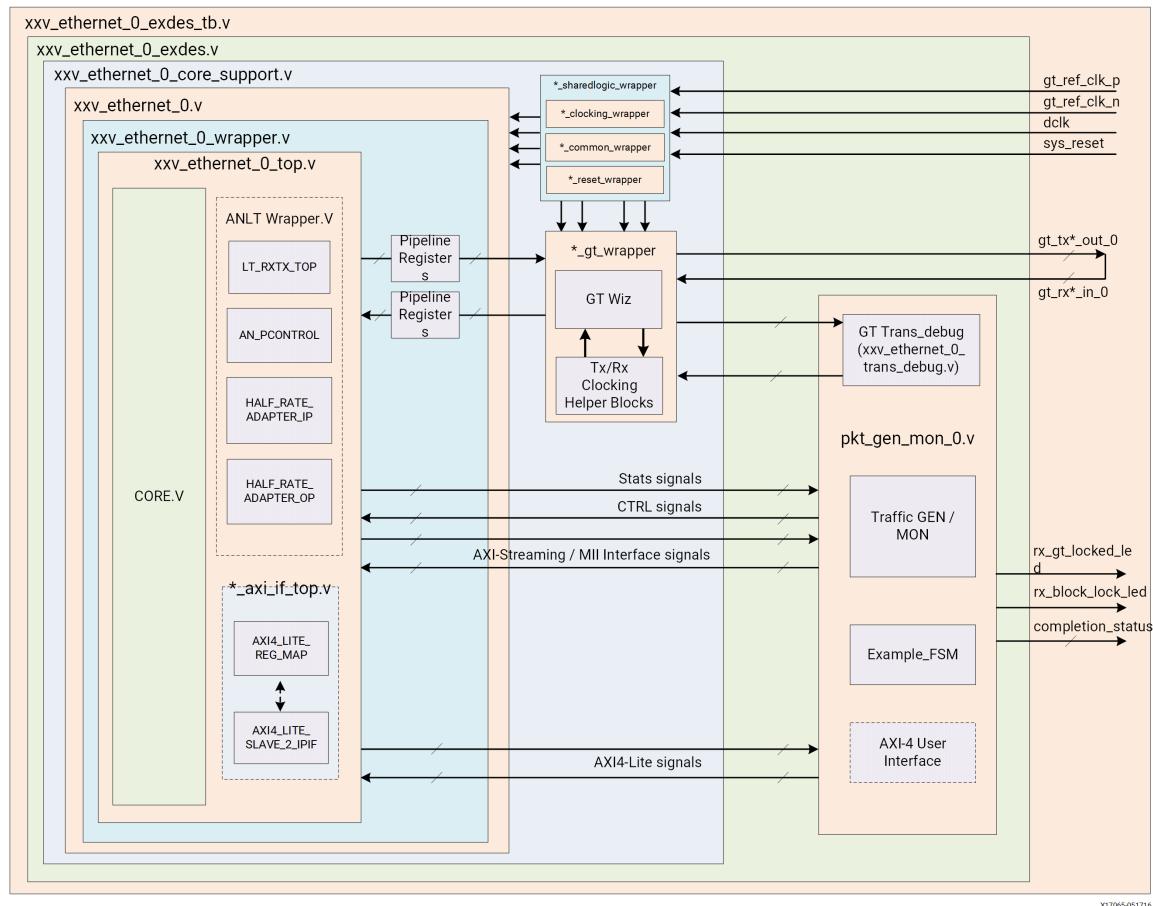
图 47：多核设计示例层级



## 设计示例层级（设计示例中包含 GT）

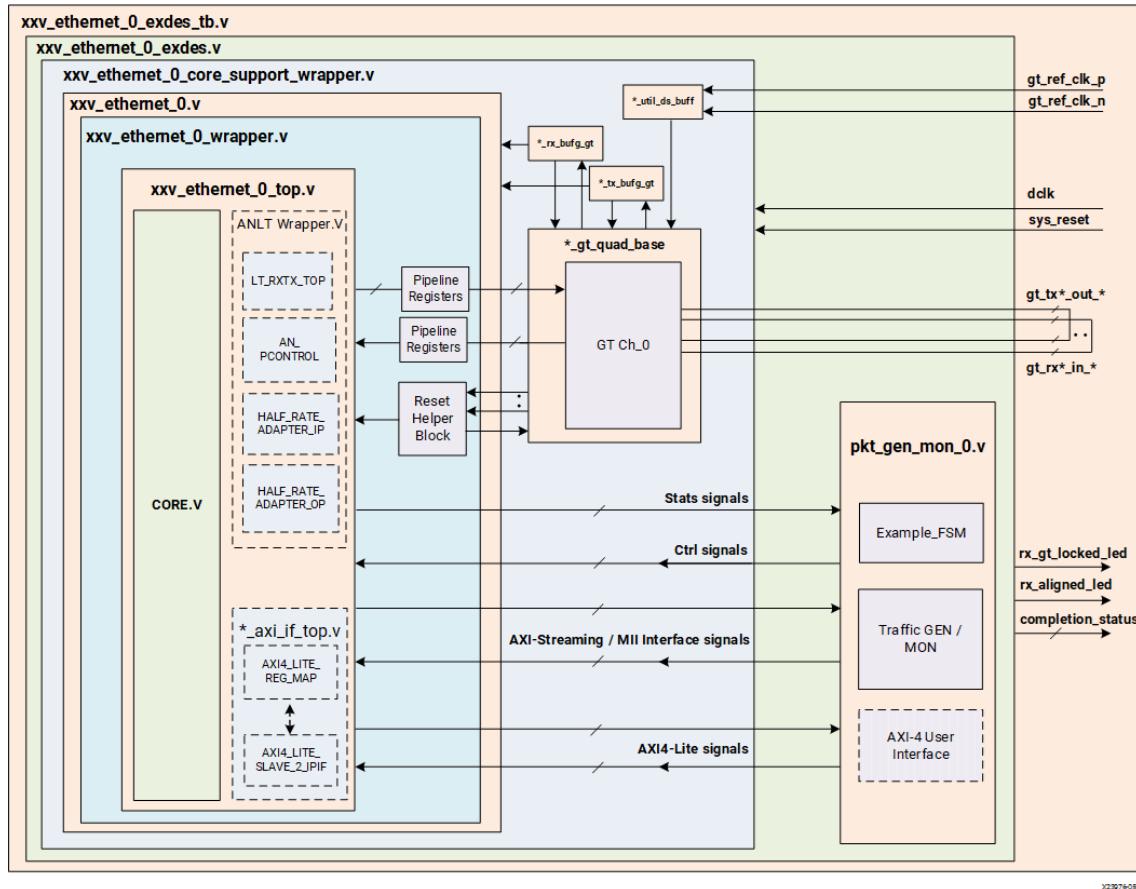
将 10/25G Ethernet 子系统添加到 Vivado IP integrator 中之后，Run Block Automation IP 核与 GT（串行收发器）将根据核配置与部分帮助程序块相连。在 10/25G Ethernet IP 内部有复位接口 IP，用于将 TX/RX mstreset 释放到 Versal 器件 GT、检查 TX/RX mstresetdone 状态并将 GT 排序复位。

图 48：设计示例层级中含 GT 的单核



X17065-051716

图 49：设计示例层级中含 GT 的单核 (Versal ACAP)



上图显示了当 GT（串行收发器）位于 IP 核外部（即，位于设计示例中）时，xxv\_ethernet\_0 设计的单核配置的各模块的例化及其层级。在“GT Selection and Configuration”（GT 选择和配置）选项卡下选中“Include GT subcore in example design”（在设计示例中包含 GT 子核）选项时即可生成此层级设计示例。

在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项时，在层级中即可显示 `xxv_ethernet_0_core_support.v`。

可用于 MAC/PCS 配置和 PCS 配置的用户接口与“概述”主题中所提及的配置相同。

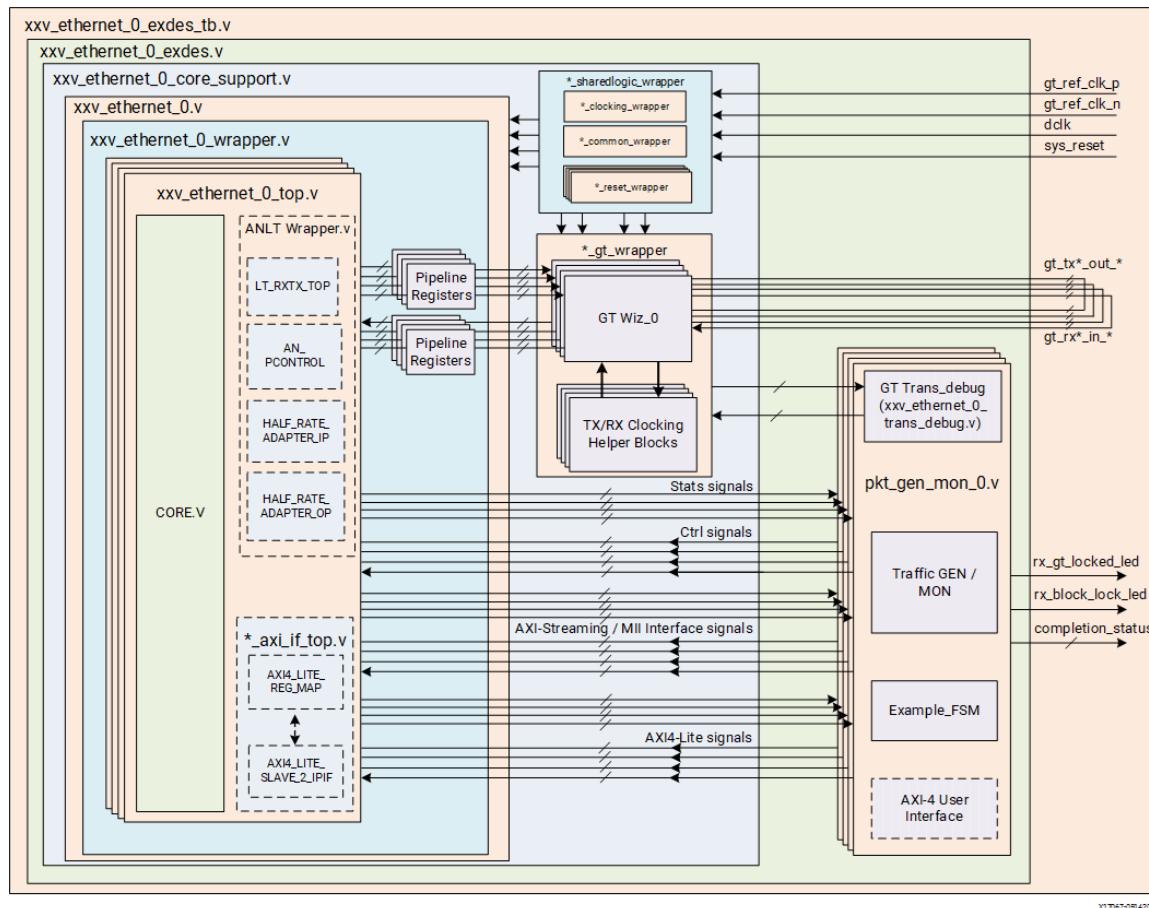
`xxv_ethernet_0.v` 模块会例化必要的同步寄存器/重定时流水线寄存器，以在核与 GT 之间实现数据同步。

`xxv_ethernet_0_pkt_gen_mon` 模块用于生成数据包以进行完整性测试。包生成和检查由有限状态机 (FSM) 模块来控制。可选模块的描述如下：

- `xxv_ethernet_0_sharedlogic_wrapper`: 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项或者在“Shared Logic”选项卡下选中“Include Shared Logic in the Example Design”时，在设计示例中即可显示此模块。此模块包含可在位于 IP 核外部的多个 IP 核与设计之间共享的所有模块。
- `xxv_ethernet_0_gt_wrapper`: 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项时，在设计示例中即可显示此模块。此模块包含 GT 的例化以及各种帮助程序块。时钟帮助程序块用于为该核生成所需时钟频率。

下图显示了当 GT 位于设计示例中时，xxv\_ethernet\_0 设计示例的多核配置的各模块的例化及其层级。

图 50：按设计示例层级划分的含 GT 的多核

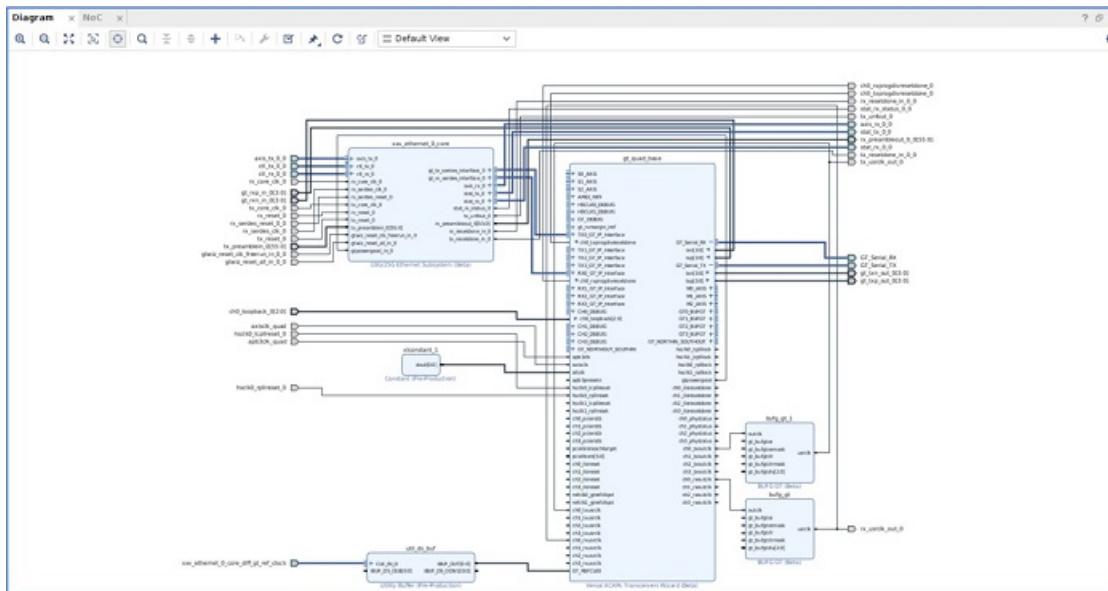


对于 Versal 平台，`gt_quad_base`（适用于 Versal 器件的 GT Wizard）将仅包含在设计示例中，而 10/25G Ethernet Subsystem IP 和 GT（串行收发器）IP 将在块设计中使用 IP integrator 相连。

下图显示了块设计，其中 IP integrator 中已连接 10/25G 以太网设计示例。如需了解有关 IP integrator 的更多信息，请参阅《Vivado Design Suite 教程：采用 IP integrator 设计 IP 子系统》([UG995](#))。

**注释：**将 10/25G Ethernet 子系统添加到 Vivado IP integrator 中之后，Run Block Automation IP 核与 GT（串行收发器）将根据核配置与部分帮助程序块相连。在 10/25G Ethernet IP 内部有复位接口 IP，用于将 TX/RX mstreset 释放到 Versal 器件 GT、检查 TX/RX mstresetdone 状态并将 GT 排序复位。

图 51：10/25G 以太网块设计



## 用户接口

提供的通用 I/O (GPIO) 用于控制设计示例。用户输入和输出端口如下表中所述。

表 301：用户 I/O 端口

名称	大小	I/O	描述
sys_reset	1	输入	用于核的复位。
gt_ref_clk_p	1	输入	差分输入时钟，连接到 GT。此时钟频率应与 Vivado IDE 的“GT Selection and Configuration”选项卡中所提及的 GT RefClk 频率相等。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”时，此端口可用。
gt_ref_clk_n	1	输入	差分输入时钟，连接到 GT。此时钟频率应与 Vivado IDE 的“GT Selection and Configuration”选项卡中所提及的 GT RefClk 频率相等。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”时，此端口可用。

表 301：用户 I/O 端口（续）

名称	大小	I/O	描述
gtm_gtrefclk	1	输入	单端输入时钟，连接到 GT。此时钟频率应与 Vivado IDE 的“GT Selection and Configuration”选项卡中所提及的 GT Refclk 频率相等。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in Example Design”选项时，此端口可供非 Versal GTM 器件使用。
dclk	1	输入	稳定/自由运行的输入时钟，连接到 GT。此时钟频率应与 Vivado IDE 的“GT Selection and Configuration”选项卡中所提及的 GT DRP 时钟频率相等。
rx_gt_locked_led_0	1	输出	指示 GT 处于已锁定状态。
rx_block_lock_led_0	1	输出	表示已达成 RX 块锁定。
restart_tx_rx_0	1	输入	当包生成器和包监控器处于空闲状态时，此信号用于重新启动包生成和接收以便执行数据完整性测试。
completion_status	5	输出	此信号表示测试状态/结果。 <ul style="list-style-type: none"><li>· 5'd0: 测试未运行。</li><li>· 5'd1: 25GE/10GE 核测试已成功完成，并已通过。</li><li>· 5'd2: 没有任何通道上存在块锁定。</li><li>· 5'd3: 并非所有通道都已实现块锁定。</li><li>· 5'd4: 部分通道实现块锁定后丢失块锁定。</li><li>· 5'd5: 没有任何通道上存在通道同步。</li><li>· 5'd6: 并非所有通道都已实现同步。</li><li>· 5'd7: 部分通道实现同步后丢失同步。</li><li>· 5'd8: 未实现任何对齐状态或 rx_status。</li><li>· 5'd9: 实现对齐状态和 rx_status 后，其中之一丢失。</li><li>· 5'd10: TX 超时。</li><li>· 5'd11: 未发送任何 TX 数据。</li><li>· 5'd12: 接收到的包数量与发送的包数量不相等。</li><li>· 5'd13: 接收到的字节总数与发送的字节总数不相等。</li><li>· 5'd14: 检测到协议错误。</li><li>· 5'd15: 在接收到的包中检测到位元错误。</li><li>· 5'd31: 测试滞留在复位状态。</li></ul>
mode_change_*	1	输入	仅当在 Vivado IDE 中选中“Runtime Switchable”（运行时可切换）时，此端口才可用，且用于切换核速度。
core_speed_*	1	输出	此信号用于指示核的处理速度： 1'b1 = 10G 和 1'b0 = 25G
send_continuous_pkts_*	1	输入	此端口可用于发送连续包以供开发板确认。 <ul style="list-style-type: none"><li>· 1'b0: 发送固定的 20 个包以供仿真。</li><li>· 1'b1: 发送连续包以供开发板验证。</li></ul> <b>注释：</b> 当 send_continuous_pkts 设为 1 时，生成器将继续发送包。仅当此信号将复位为 0 时，仿真才能完成。
stat_reg_compare	1	输出	指示 TX 和 RX 统计数据寄存器比较状态。 <ul style="list-style-type: none"><li>· 1'b1: 指示 TX 和 RX 统计数据均匹配。</li><li>· 1'b0: 指示 TX 与 RX 统计数据之间是否存在任何不匹配。</li></ul> 在“General”（常规）选项卡下选中“Include AXI4-Lite”（包含 AXI4-Lite）选项时，此输出可用。

表 301：用户 I/O 端口（续）

名称	大小	I/O	描述
ts_clk	1	输入	这是系统定时器时钟输入端口。 <b>注释：</b> 在 GUI 选项卡-2 下选中“Enable Timestamping Logic”时，此端口可用。
ptp_results_*	1	输出	定时器比较信号输出至监控器，并将工具限制为对 PTP 设计进行最优化。 <b>注释：</b> 在 GUI 选项卡-2 下选中“Enable Timestamping Logic”时，此端口可用。

## 核 xci 顶层端口列表

以下列出了启用所有功能的核 xci 的顶层端口列表：

在下表中，星号 (\*) 表示核编号（值为 0 到 3）。

示例：port\_name\_\*

- port\_name\_0：表示第一个核
- port\_name\_1：表示第二个核（核数指定为  $\geq 2$  时显示）
- port\_name\_2：表示第三个核（核数指定为  $\geq 3$  时显示）
- port\_name\_3：表示第四个核（核数指定为 =4 时显示）

## 公共时钟/复位信号

表 302：公共时钟/复位信号

名称	大小	I/O	描述
sys_reset	1	输入	用于核的复位。 <b>注释：</b> <ol style="list-style-type: none"><li>1. 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并且核类型并非 Ethernet MAC 64 位时，此端口可用。</li><li>2. 在“GT Selection and Configuration”选项卡下针对设计示例选中“Include GT subcore”选项时，此端口可用。核类型为 Ethernet MAC+PCS/PMA 32 位或 PCS/PMA 32 位，且启用时间戳。</li></ol>

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
dclk	1	输入	<p>稳定输入时钟，连接到 GT。</p> <p><b>注释：</b></p> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并且核类型并非 Ethernet MAC 64 位时，此端口可用。</li><li>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项时，此端口可用。核类型为 Ethernet MAC+PCS/PMA 32 位或 PCS/PMA 32 位，且启用时间戳。</li></ol>
sys_reset_0	1	输入	<p>用于核的复位。</p> <p><b>注释：</b></p> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include GT subcore in core”选项，且核类型并非 Ethernet MAC 64 位时，此端口可用。</li><li>在“GT Selection and Configuration”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include GT subcore in example design”选项，且核类型为 Ethernet MAC+PCS/PMA 32 位或 PCS/PMA 32 位，同时启用时间戳时，此端口可用。</li></ol>
dclk_0	1	输入	<p>稳定输入时钟，连接到 GT。</p> <p><b>注释：</b></p> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include GT subcore in core”选项，且核类型并非 Ethernet MAC 64 位时，此端口可用。</li><li>在“GT Selection and Configuration”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include GT subcore in example design”选项，且核类型为 Ethernet MAC+PCS/PMA 32 位或 PCS/PMA 32 位，同时启用时间戳时，此端口可用。</li></ol>
clk_322	1	输入	<p>作为基于 CMAC 的硬核 RS-FEC 块的稳定输入时钟。</p> <p><b>注释：</b>在“Shared Logic”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include Shared Logic in Example Design”选项时，此端口可供非 Versal GTM 器件使用。您必须向该核提供此时钟信号（频率为 322.265625 MHz）。</p>
locked_out_322	1	输入	<p>作为基于 CMAC 的硬核 RS-FEC 块的稳定复位。</p> <p><b>注释：</b>在“Shared Logic”选项卡下启用“Soft RS-FEC TX”选项和“Hard RS-FEC RX”选项并选中“Include Shared Logic in Example Design”选项时，此端口可供使用。您必须向核 w.r.t. clk_322 信号提供该信号。</p>
gt_refclk_p	1	输入	<p>差分输入时钟，连接到 GT。</p> <p><b>注释：</b>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”选项时，此端口可用。</p>
gt_refclk_n	1	输入	<p>差分输入时钟，连接到 GT。</p> <p><b>注释：</b>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”选项时，此端口可用。</p>

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
qpll0clk_in	2/4	输入	QPLL0 时钟输入。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”，且所选“GT Type”并非 GTM 时，此端口可用。  端口宽度：50G 单核为 2 位，40G 单核/50G 双核为 4 位。
qpll0refclk_in	2/4	输入	QPLL0 参考时钟输入。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。  端口宽度：50G 单核为 2 位，40G 单核/50G 双核为 4 位。
qpll1clk_in	2/4	输入	QPLL1 时钟输入。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。  端口宽度：50G 单核为 2 位，40G 单核/50G 双核为 4 位。
qpll1refclk_in	2/4	输入	QPLL1 参考时钟输入。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。  端口宽度：50G 单核为 2 位，40G 单核/50G 双核为 4 位。
gtwiz_reset_qpll0lock_in	1	输入	QPLL0 锁定复位输入，连接到 GT。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。
gtwiz_reset_qpll0reset_out	1	输出	QPLL0 锁定复位输出，来自 GT。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。
gtwiz_reset_qpll1lock_in	1	输入	QPLL1 锁定复位输入，连接到 GT。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
gtwiz_reset_qpll1reset_out	1	输出	QPLL1 锁定复位输出，来自 GT。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”并非 GTM 时，此端口可用。
tx_clk_out_*	1	输出	来自 GT 的 TX 用户时钟输出。  <b>注释：</b> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA 32/64-bit”且选中“Include GT subcore in core”选项时，此端口可用。</li><li>在“Selection and Configuration”选项卡下，所选“Select Core”为“Ethernet MAC+PCS/PMA 32/64-bit”、“GT type”为 GTM 并选中“Include GT subcore in example design”选项时，此端口可用。</li></ol>
tx_mii_clk_*	1	输出	来自 GT 的 TX mii 时钟输出。  <b>注释：</b> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下，所选“Select Core”为“Ethernet PCS/PMA 32/64-bit”，并选中“Include GT subcore in core”选项时，此端口可用。</li><li>在“Selection and Configuration”选项卡下，所选“Select Core”为“Ethernet PCS/PMA 32/64-bit”、“GT type”为 GTM 并选中“Include GT subcore in example design”选项时，此端口可用。</li></ol>
rx_clk_out_*	1	输出	来自 GT 的 RX 用户时钟输出。  <b>注释：</b> <ol style="list-style-type: none"><li>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。</li><li>在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型为 GTM 时，此端口可用。</li></ol>
rx_serdes_clk_*	1	输入	RX serdes 时钟输入，连接到核。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
rx_serdes_reset_*	1	输入	RX serdes 复位输入，连接到核  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项时，此端口可用。
rxrecclkout_*	1	输出	来自 GT 的 RX 恢复时钟输出。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
tx_core_clk_*	1	输入	来自 GT 封装器的 TX 核时钟输入。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
rx_core_clk_*	1	输入	RX 核时钟输入，连接到核。
tx_reset_*	1	输入	TX 复位输入，连接到核。
user_tx_reset_*	1	输出	对应用户逻辑的 TX 复位输出。 <b>注释：</b> user_tx_reset 可用于复位用户逻辑。当 GT resetdone 转至低位，并且时钟变为不稳定时，IP 会将其断言有效。在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”选项时，此端口可用。
gt_reset_tx_done_out_*	1	输出	来自 GT 的 TX 复位已完成信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
rx_reset_*	1	输入	RX 复位输入，连接到核。
user_rx_reset_*	1	输出	对应用户逻辑的 RX 复位输出。 <b>注释：</b> user_rx_reset 可用于复位用户逻辑。当 GT resetdone 转至低位，并且时钟变为不稳定时，IP 会将其断言有效。在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in core”选项时，此端口可用。
gt_reset_rx_done_out_*	1	输出	来自 GT 的 RX 复位已完成信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gtwiz_reset_all_in*	1	输入	来自用户的 gt_reset_all 信号。 <b>注释：</b> 仅限 Versal 器件。在“Configuration”选项卡下选中“Control and Statistics Interface”时，此端口可用。
ctl_gt_reset_all_*	1	输出	来自 AXI4-Lite 寄存器映射的 gt_reset_all 信号。 <b>注释：</b> 在“Configuration”选项卡下选中“Include AXI4-Lite”并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gtwiz_tx_datapath_reset_in_*	1	输入	来自用户的 gt_tx_reset 信号。 <b>注释：</b> 仅限 Versal 器件。在“Configuration”选项卡下选中“Control and Statistics Interface”选项时，此端口可用。

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
ctl_gt_tx_reset_*	1	输出	来自 AXI4-Lite 寄存器映射的 gt_tx_reset 信号。  <b>注释：</b> 在“Configuration”选项卡下选中“Include AXI4-Lite”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gtwiz_rx_datapath_reset_in_*	1	输入	来自用户的 gt_rx_reset 信号。  <b>注释：</b> 仅限 Versal 器件。在“Configuration”选项卡下选中“Control and Statistics Interface”时，此端口可用。
ctl_gt_rx_reset_*	1	输出	来自 AXI4-Lite 寄存器映射的 gt_rx_reset 信号。  <b>注释：</b> 在“Configuration”选项卡下选中“Include AXI4-Lite”并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gt_reset_all_in_*	1	输入	来自共享逻辑封装器的 reset_wrapper 的 gt_reset_all 信号。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gt_tx_reset_in_*	1	输入	来自共享逻辑封装器的 reset_wrapper 的 gt_tx_reset_in 信号。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gt_rx_reset_in_*	1	输入	来自共享逻辑封装器的 reset_wrapper 的 gt_rx_reset_in 信号。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gt_refclk_out	1	输出	用于指示 GT_refclk 输出。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可用。
gtpowergood_out_*	1	输出	请参阅《UltraScale 架构 GTH 收发器用户指南》(UG576) 或《UltraScale 架构 GTY 收发器用户指南》(UG578) 以获取端口描述。
TXOUTCLKSEL_IN_*	3	输入	此端口用于选择 gtwizard TX 输出时钟的时钟源 此端口根据预置以 3'b101 驱动，对于 GTM GT 类型，将不呈现此端口。
RXOUTCLKSEL_IN_*	3	输入	此端口用于选择 gtwizard RX 输出时钟的时钟源 此端口根据预置以 3'b101 驱动，对于 GTM GT 类型，将不呈现此端口。
gtm_txusrclk2_*	1	输入	TX 时钟输入，连接到核。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型为 GTM 时，此端口可用。
gtm_rxusrclk2_*	1	输入	RX 时钟输入，连接到核。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型为 GTM 时，此端口可用。

表 302：公共时钟/复位信号 (续)

名称	大小	I/O	描述
gtm_txusrclk_out2_*	1	输出	表示核中包含 gt 时，来自该核的 TX 时钟输出。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”为 GTM 时，此端口可用。
gtm_rxusrclk_out2_*	1	输出	表示核中包含 gt 时，来自该核的 RX 时钟输出。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”为 GTM 时，此端口可用。
gtm_rxprgdivresetdone_*	1	输出	表示核中包含 gt 时，来自该核的 RX 复位输出。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项、在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项，且所选“GT Type”为 GTM 时，此端口可用。
txoutclk_out_*	1	输入	TX 时钟输入，连接到核。 <b>注释：</b> 在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可供 GTM 器件使用。您必须向该核提供此频率（针对 10G 提供的频率为 156.25 MHz，针对 25G 提供的频率为 390.625 MHz）。
rxoutclk_out_*	1	输入	RX 时钟输入，连接到核。 <b>注释：</b> 在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可供 GTM 器件使用。您必须向该核提供此频率（针对 10G 提供的频率为 156.25 MHz，针对 25G 提供的频率为 390.625 MHz）。
tx_locked_*	1	输入	TX resetdone 输入，连接到核 <b>注释：</b> 在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可供 GTM 器件使用。您必须向核 w.r.t. txoutclk_out_* 时钟提供此输入。
rx_locked_*	1	输入	X resetdone 输入，连接到核。 <b>注释：</b> 在“Shared Logic”选项卡下选中“Include Shared Logic in example design”选项时，此端口可供 GTM 器件使用。您必须向核 w.r.t. rxoutclk_out_* 时钟提供此输入。

## 常用收发器接口端口

表 303：常用收发器接口端口

名称	大小	I/O	描述
gt_loopback_in_*	3	输入	GT 环回输入信号。 请参阅 GT 用户指南。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。
gt_loopback_out_*	1	输出	GT 环回输出信号（来自 AXI4-Stream 寄存器映射）。 请参阅 GT 用户指南。  <b>注释：</b> 在“Configuration”选项卡下选中“Include AXI4-Lite”并在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项时，此端口可用。
gt_txp_out	1	输出	差分串行 GT TX 输出  <b>注释：</b> 对于基于开发板的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。  <b>注释：</b> 对于基于器件的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项和“Enable GT Interface for Board Based Design”选项时，此端口可用。
gt_txn_out	1	输出	差分串行 GT TX 输出。  <b>注释：</b> 对于基于开发板的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。  <b>注释：</b> 对于基于器件的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项和“Enable GT Interface for Board Based Design”选项时，此端口可用。
gt_rxn_in	1	输入	差分串行 GT RX 输入。  <b>注释：</b> 对于基于开发板的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。  <b>注释：</b> 对于基于器件的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项和“Enable GT Interface for Board Based Design”选项时，此端口可用。
gt_rxp_in	1	输入	差分串行 GT RX 输入。  <b>注释：</b> 对于基于开发板的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。  <b>注释：</b> 对于基于器件的设计，在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项和“Enable GT Interface for Board Based Design”选项时，此端口可用。

表 303：常用收发器接口端口 (续)

名称	大小	I/O	描述
gt_rxp_in_0	1	输入	差分串行 GT RX 输入（针对通道 0）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。
gt_rxn_in_0	1	输入	差分串行 GT RX 输入（针对通道 0）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。
gt_rxp_in_1	1	输入	差分串行 GT RX 输入（针对通道 1）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 1，并选中“Include GT subcore in core”选项时，此端口可用。
gt_rxn_in_1	1	输入	差分串行 GT RX 输入（针对通道 1）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 1，并选中“Include GT subcore in core”选项时，此端口可用。
gt_rxp_in_2	1	输入	差分串行 GT RX 输入（针对通道 2）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 2，并选中“Include GT subcore in core”选项时，此端口可用。
gt_rxn_in_2	1	输入	差分串行 GT RX 输入（针对通道 2）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 2，并选中“Include GT subcore in core”选项时，此端口可用。
gt_rxp_in_3	1	输入	差分串行 GT RX 输入（针对通道 3）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 3，并选中“Include GT subcore in core”选项时，此端口可用。
gt_rxn_in_3	1	输入	差分串行 GT RX 输入（针对通道 3）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 3，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txp_out_0	1	输出	差分串行 GT TX 输出（针对通道 0）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。
gt_txn_out_0	1	输出	差分串行 GT TX 输出（针对通道 0）。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项时，此端口可用。

表 303：常用收发器接口端口 (续)

名称	大小	I/O	描述
gt_txp_out_1	1	输出	差分串行 GT TX 输出（针对通道 1）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 1，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txn_out_1	1	输出	差分串行 GT TX 输出（针对通道 1）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 1，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txp_out_2	1	输出	差分串行 GT TX 输出（针对通道 2）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 2，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txn_out_2	1	输出	差分串行 GT TX 输出（针对通道 2）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 2，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txp_out_3	1	输出	差分串行 GT TX 输出（针对通道 3）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 3，并选中“Include GT subcore in core”选项时，此端口可用。
gt_txn_out_3	1	输出	差分串行 GT TX 输出（针对通道 3）。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下“Num of Cores”大于 3，并选中“Include GT subcore in core”选项时，此端口可用。
gtwiz_loopback_*	3	输出	GT 环回输出信号（来自 AXI4-Lite 寄存器映射）。（仅限 Versal 器件）。请参阅相应的 GT 用户指南。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并在“Configuration”选项卡下选中 AXI4-Lite 接口时，此端口可用。  <b>注释：</b> 您必须在开发板设计中手动连接此信号。
gtwiz_tx_rate_*	8	输出	可从 AXI4-Lite 寄存器映射中选择 GT TX 线速率。（仅限 Versal 器件）。请参阅相应的 GT 用户指南。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并在“Configuration”选项卡下选中 AXI4-Lite 接口时，此端口可用。
gtwiz_rx_rate_*	8	输出	可从 AXI4-Lite 寄存器映射中选择 GT TX 线速率。（仅限 Versal 器件）。请参阅相应的 GT 用户指南。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并在“Configuration”选项卡下选中 AXI4-Lite 接口时，此端口可用。

表 303：常用收发器接口端口 (续)

名称	大小	I/O	描述
rxgearboxslip_in_*	1	输出	从核到 GT 的 Rxgearboxslip 信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
temperature	10	输入	如需了解更多信息，请参阅《Virtex UltraScale+ FPGA GTM Transceivers Wizard LogiCORE IP 产品指南》(PG315)。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in core”选项并且 GT 类型为 GTM 时，此端口可用。
rxdatavalid_out_*	2	输入	从 GT 到核的 RX 数据有效信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
rx_serdes_data_out_*	32/64/128	输入	从 GT 到核的 RX 数据信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。 对于 10G 配置，数据宽度为 32/64 位，对于 25G 配置，则为 128 位。
rxdata_out_*	256	输入	从 GT 到核的 RX 数据信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型为 GTM 时，此端口可用。
rxheader_out_*	6	输入	从 GT 到核的 RX 报头信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
rxheadervalid_out_*	2	输入	从 GT 到核的 RX 报头有效信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。
tx_serdes_data_in_*	32/64/128	输出	从核到 GT 的 TX 数据信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型并非 GTM 时，此端口可用。 对于 10G 配置，数据宽度为 32/64 位，对于 25G 配置，则为 128 位。
txdata_in_*	256	输出	从核到 GT 的 TX 数据信号。 <b>注释：</b> 在“GT Selection and Configuration”选项卡下选中“Include GT subcore in example design”选项并且 GT 类型为 GTM 时，此端口可用。

表 303：常用收发器接口端口 (续)

名称	大小	I/O	描述
mst_tx_resetdone_*	1	输入	从 GT 到核的 TX 主控制器复位完成信号，用于指示 lane0 状态。 (仅限 Versal 器件)。
mst_rx_resetdone_*	1	输入	从 GT 到核的 RX 主控制器复位完成信号，用于指示 lane0 状态。 (仅限 Versal 器件)。
tx_pma_resetdone_*	1	输入	从 GT 到核的 TX PMA 复位完成信号，用于指示 lane0 状态。 (仅限 Versal 器件)。
rx_pma_resetdone_*	1	输入	从 GT 到核的 RX PMA 复位完成信号，用于指示 lane0 状态。 (仅限 Versal 器件)。
mst_tx_reset_*	1	输出	从核到 Lane0 GT 的 TX 主控制器复位输出信号。 (仅限 Versal 器件)。
mst_rx_reset_*	1	输出	从核到 Lane0 GT 的 RX 主控制器复位输出信号。 (仅限 Versal 器件)。
txuserrdy_out_*	1	输出	从核 (复位接口 IP) 到 Lane0 GT 的 TX 用户就绪输出信号。 (仅限 Versal 器件)。
mst_tx_dp_reset_*	1	输出	从 GT 复位 IP 到核的 TX 复位输出信号 (仅限 Versal 器件)。
mst_rx_dp_reset_*	1	输出	从 GT 复位 IP 到核的 RX 复位输出信号 (仅限 Versal 器件)。
rxuserrdy_out_*	1	输出	从核 (复位接口 IP) 到 Lane0 GT 的 RX 用户就绪输出信号。 (仅限 Versal 器件)。
tx_resetdone_out_*	1	输出	从核到设计示例的 TX 用户就绪输出信号。 (仅限 Versal 器件)。
rx_resetdone_out_*	1	输出	从核到设计示例的 RX 用户就绪输出信号。 (仅限 Versal 器件)。
txheader_in_*	6	输出	从核到 GT 的 TX 报头信号。  注释：选中“GT Selection and Configuration”选项卡中的“Include GT subcore in example design”选项时，此端口可用。

## 收发器核和状态调试端口

在“GT Selection and Configuration”(GT 选择和配置)选项卡下选中“Include GT subcore in core”(在核中包含 GT 子核)选项，或者在“GT Selection and Configuration”选项卡下选中“Enable Additional GT Control/Status and DRP Ports”(启用其它 GT 控制/状态和 DRP 端口)时，下表中的端口可用。

如需获取端口描述，请参阅《UltraScale FPGA Transceivers Wizard LogiCORE IP 产品指南》(PG182)、《UltraScale 架构 GTH 收发器用户指南》(UG576) 和《UltraScale 架构 GTY 收发器用户指南》(UG578)。

表 304：收发器核和状态调试端口

名称	大小	I/O
gt_dmonitorout_*	16	输出
gt_eyescandataerror_*	1	输出
gt_eyescanreset_*	1	输入
gt_eyescantrigger_*	1	输入
gt_pcsrcsvdin_*	16	输入
gt_rxbufreset_*	1	输入
gt_rxbufstatus_*	3	输出
gt_rxcdrhold_*	1	输入

表 304：收发器核和状态调试端口 (续)

名称	大小	I/O
gt_rxcommadeten_*	1	输入
gt_rxdfeghold_*	1	输入
gt_rxdfelpmreset_*	1	输入
gt_rxlatclk_*	1	输入
gt_rxlpmen_*	1	输入
gt_rxpcreset_*	1	输入
gt_rxpmareset_*	1	输入
gt_rxpolarity_*	1	输入
gt_rxprbscntreset_*	1	输入
gt_rxprbslocked_*	1	输出
gt_rxprbserr_*	1	输入
gt_rxprbssel_*	4	输入
gt_rxrate_*	3	输入
gt_rxslide_in_*	1	输入
gt_rxstartofseq_*	2	输出
gt_txresetdone_*	1	输出
gt_txbufstatus_*	2	输出
gt_txdiffctrl_*	5	输入
gt_txinhibit_*	1	输入
gt_txlatclk_*	1	输入
gt_txmaincursor_*	7	输入
gt_txpcreset_*	1	输入
gt_txpmareset_*	1	输入
gt_txpolarity_*	1	输入
gt_txpostcursor_*	5	输入
gt_txprbsforceerr_*	1	输入
gt_txprbssel_*	4	输入
gt_txprecursor_*	5	输入
gtwiz_reset_tx_datapath_*	1	输入
gtwiz_reset_rx_datapath_*	1	输入
gt_drpclk_* <sup>1</sup>	1	输入
gt_drpdo_* <sup>1</sup>	16	输出
gt_drprdy_* <sup>1</sup>	1	输出
gt_drpren_* <sup>1</sup>	1	输入
gt_drpwe_* <sup>1</sup>	1	输入
gt_drpaddr_* <sup>1</sup>	10	输入
gt_drpdi_* <sup>1</sup>	16	输入

表 304：收发器核和状态调试端口 (续)

名称	大小	I/O
gt_drprst_*	1	输入

**注释：**

- 此端口仅可用于非 Versal 器件。

## AXI4-Lite 接口端口

在“Configuration”（配置）选项卡下选中“Include AXI4-Lite”（包含 AXI4-Lite）时，下表中的端口可用。

表 305：AXI4-Lite 接口端口

名称	大小	I/O	描述
s_axi_aclk_*	1	输入	AXI 时钟信号
s_axi_arresetn_*	1	输入	AXI 复位信号
pm_tick_*	1	输入	PM 时钟节拍用户输入
s_axi_awaddr_*	32	输入	AXI 写入地址
s_axi_awvalid_*	1	输入	AXI 写入地址有效
s_axi_awready_*	1	输出	AXI 写入地址就绪
s_axi_wdata_*	32	输入	AXI 写入数据
s_axi_wstrb_*	4	输入	AXI 写入选通。此信号用于指示保存有效数据的字节通道。
s_axi_wvalid_*	1	输入	AXI 写入数据有效。此信号用于指示有效的写入数据和写入选通均可使用。
s_axi_wready_*	1	输出	AXI 写入数据就绪
s_axi_bresp_*	2	输出	AXI 写入响应。此信号用于指示写入传输事务的状态。 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
s_axi_bvalid_*	1	输出	AXI 写入响应有效。此信号用于指示该通道正在发出有效的写入响应信号。
s_axi_bready_*	1	输入	AXI 写入响应就绪。
s_axi_araddr_*	32	输入	AXI 读取地址
s_axi_arvalid_*	1	输入	AXI 读取地址有效
s_axi_arready_*	1	输出	AXI 读取地址就绪
s_axi_rdata_*	32	输出	AXI 读取数据（由从接口发出）
s_axi_rresp_*	2	输出	AXI 读取响应。此信号用于指示读取传输的状态。 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
s_axi_rvalid_*	1	输出	AXI 读取数据有效
s_axi_rready_*	1	输入	AXI 读取就绪。此信号用于指示用户/主控制器可接受读取数据和响应信息。

## AXI4-Stream 用户接口信号

在“Configuration”（配置）选项卡下选中“Ethernet MAC + PCS/PMA-32/64-bit”（以太网 MAC + PCS/PMA 32/64 位）选项时，本节下所述端口可用。

表 306：AXI4-Stream 用户接口信号

名称	大小	I/O	描述
tx_unfout_*	1	输出	来自核的 TX 数据路径下溢信号。如果 tx_unfout_* 采样为 1，则表示已发生违例，即当前包已受损。只要下溢状况仍然存在，就会发射错误控制块信号。  用户逻辑负责将完整的包输入核，并避免 TX 数据路径接口欠载。  注释：当此信号采样为 1 时，您必须应用 tx_reset/sys_reset 才能将子系统从下溢问题恢复。tx_reset 仅复位 TX 路径，sys_reset 则会恢复整个系统。
tx_axis_tready_*	1	输出	来自核的 TX 路径就绪信号。
tx_axis_tvalid_*	1	输入	发射 AXI4-Stream 数据有效。
tx_axis_tdata_*	64/32	输入	发射 AXI4-Stream 数据总线。
tx_axis_tlast_*	1	输入	发射 AXI4-Stream tlast。
tx_axis_tkeep_*	8/4	输入	发射 AXI4-Stream tkeep。
tx_axis_tuser_*	1	输入	发射 AXI4-Stream tuser。
tx_preamblein_*	56	输入	发射 AXI4-Stream 前导码。
tx parityin_*	8	输入	发射 AXI4-Stream 数据路径奇偶校验。
rx_axis_tvalid_*	1	输出	接收 AXI4-Stream 数据有效。
rx_axis_tdata_*	64/32	输出	接收 AXI4-Stream 数据总线。
rx_axis_tlast_*	1	输入	接收 AXI4-Stream tlast。
rx_axis_tkeep_*	8/4	输入	接收 AXI4-Stream tkeep。
rx_axis_tuser_*	1	输入	接收 AXI4-Stream tuser。
rx_preamblein_*	56	输入	接收 AXI4-Stream 前导码。
rx parityout_*	8	输出	接收 AXI4-Stream 数据路径奇偶校验。

## MII 用户接口信号

在“Configuration”（配置）选项卡下选中“Ethernet MAC+PCS/PMA-32/64-bit”（以太网 MAC+PCS/PMA 32/64 位）选项时，本节下所述端口可用。

表 307：MII 用户接口信号

名称	大小	I/O	描述
tx_mii_d_*	64	输入	发射 XGMII/25GMII 数据总线。
tx_mii_c_*	8	输入	XGMII/25GMII 控制总线。
rx_mii_d_*	64	输出	接收 XGMII/25GMII 数据总线。
rx_mii_c_*	8	输出	XGMII/25GMII 接收控制总线。

## TX 路径控制信号、状态信号和统计数据信号

表 308：TX 路径控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_tx_enable_*	1	输入	<p>TX 使能（启用）。此信号采样为 1 时用于启用数据发射。当此信号采样为 0 时，核仅发射空闲信号。仅当数据发射到的接收器（即其它器件中的接收器）完全对齐并且已准备好接收数据（即，其它器件当前未发送远端故障状况）后，此输入才应设置为 1。</p> <p>否则，可能发生数据丢失。如果发射数据包时此信号设置为 0，那么当前数据包发射完成后，核将停止发射任何其它数据包。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_tx_send_rfi_*	1	输入	<p>发射远端故障指示（RFI）代码字。如果此输入采样为 1，那么 TX 路径仅发射远端故障（Remote Fault）代码字。此输入应设置为 1，直至 RX 路径已完全对齐并且已准备好接受来自链路伙伴的数据为止。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
ctl_tx_send_lfi_*	1	输入	<p>发射本地故障指示（LFI）代码字。优先于 RFI。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
ctl_tx_send_idle_*	1	输入	<p>发射空闲代码字。如果此输入采样为 1，那么 TX 路径仅发射空闲（Idle）代码字。当伙伴器件正在发送远端故障指示（RFI）代码字时，此输入应设置为 1。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
ctl_tx_fcs_ins_enable_*	1	输入	<p>启用由 TX 核执行 FCS 插入。如果该位设为 0，该核不会将 FCS 添加到包中。如果该位设为 1，那么该 HSEC 核会计算 FCS 并将 FCS 添加到包中。在不同数据包之间不得动态更改此输入。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_tx_ignore_fcs_*	1	输入	<p>启用由 TX 核在该接口上执行 FCS 纠错。仅当 <code>ctl_tx_fcs_ins_enable</code> 为低电平时，此输入才有效。如果此输入为低电平，并且发射的数据包中含有无效 FCS，则不会将其作为有效包丢弃。如果此输入为高电平，则含无效 FCS 的包将被作为有效包丢弃。</p> <p>在 <code>stat_tx_bad_fcs</code> 和 <code>stomped_fcs</code> 信号上会标记此错误，并且此数据包将作为已接收的包进行发射。</p> <p>报告的统计数据中显示不含 FCS 错误。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>

表 308：TX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_tx_test_pattern_*	1	输入	针对 TX 核启用扰码空闲测试模式生成。值为 1 即表示根据第 49 条中的定义启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.7。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_test_pattern_enable_*	1	输入	针对 TX 核启用测试模式。值为 1 表示启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。具有第二优先级。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_test_pattern_select_*	1	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.1。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_data_pattern_select_*	1	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_test_pattern_seed_a_*	58	输入	对应于第 45 条中所定义的 MDIO 寄存器 3.34 到 3.37。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_test_pattern_seed_b_*	58	输入	对应于第 45 条中所定义的 MDIO 寄存器 3.38 到 3.41。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_tx_prbs31_test_pattern_enable_*	1	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.4。具有第一优先级。  <b>注释：</b> 在“GT Selection and Configuration”选项卡下未选中“Include AXI4-Lite”，并且在“Configuration”选项卡下所选“Select Core”为“PCS/PMA 64-bit”(PCS/PMA 64 位)时，此端口可用。
ctl_tx_ipg_value_*	4	输入	此信号为可选。ctl_tx_ipg_value 用于定义在 rx_serdess_clk 包之间插入的目标平均最小包间隔 (IPG，以字节为单位)。典型值为 12。ctl_tx_ipg_value 还可编程为范围介于 0 到 7 之间的值，但在此情况下，它解释为表示“最小 IPG”，因此仅插入终止 (Terminate) 代码字 IPG；在此情况下从不添加空闲 (Idle)，因此发射随机大小的数据包时，生成的 IPG 平均约为 4 字节。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”(包含 AXI4-Lite)、所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”(以太网 MAC+PCS/PMA 32/64 位)，并在“MAC Options”(MAC 选项)选项卡下启用“Include FIFO Logic”(包含 FIFO 逻辑)时，此端口可用。

表 308：TX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_tx_custom_preamble_enable_*	1	输入	当此信号断言有效时，会启用使用 tx_preamblein 作为定制前导码，而不是插入标准前导码。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”、所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”，并在“MAC Options”选项卡下禁用“Include FIFO Logic”，或者所选“Select Core”为“Ethernet MAC”时，此端口可用。
ctl_tx_parity_err_response_*	1	输入	TX 核发出奇偶校验错误响应。如果该位设为 0，那么检测到任何奇偶校验错误时，该核都不会执行任何操作。如果该位设为 1，那么该核会给传出的 FCS 添加 stomp 标记（即按位反转）并断言 stat_tx_bad_fcs 有效。
stat_tx_bad_parity_*	1	输出	如果在任意时钟周期内，TX 奇偶校验检查逻辑计算所得的用户生成的奇偶校验结果不正确，那么此信号将递增。
stat_tx_local_fault_*	1	输出	值为 1 表示接收解码器状态机当前处于 TX_INIT 状态。此输出对电平敏感。
stat_tx_fifo_error_*	1	输出	指示 TX FIFO 何时进入下溢或上溢状态。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet PCS/PMA”时，此端口可用。
stat_tx_total_bytes_*	5	输出	此信号为递增值，对应已发射的字节总数。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_tx_total_packets_*	1	输出	此信号为递增值，对应已发射的数据包总数。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_tx_total_good_bytes_*	14	输出	此信号为递增值，对应已发射的有效字节总数。仅当完全发射不含错误的数据包时，该值才为非 0 值。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_tx_total_good_packets_*	1	输出	此信号为递增值，对应已发射的有效数据包总数。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_tx_bad_fcs_*	1	输出	此信号为递增值，对应具有 FCS 错误的数据包（大于 64 字节）数。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_tx_packet_64_bytes_*	1	输出	此信号为递增值，对应已发射的有效和无效数据包（含 64 字节）数。  <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。

表 308：TX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_tx_packet_65_127_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 65 到 127 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_128_255_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 128 到 255 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_256_511_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 256 到 511 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_512_1023_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 512 到 1,023 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_1024_1518_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 1,024 到 1,518 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_1519_1522_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 1,519 到 1,522 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_1523_1548_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 1,523 到 1,548 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_1549_2047_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 1,549 到 2,047 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>

表 308：TX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_tx_packet_2048_4095_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 2,048 到 4,095 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_4096_8191_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 4,096 到 8,191 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_8192_9215_bytes_*	1	输出	<p>此信号为递增值，对应已发射的有效和无效数据包（含 8,192 到 9,215 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_small_*	1	输出	<p>此信号为递增值，对应所含数据小于 64 字节的所有包数。小于 64 字节的包将不予发射。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_packet_large_*	1	输出	<p>此信号为递增值，对应所含数据超过 9,215 字节的所有包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_tx_frame_error_*	1	输出	<p>此信号为递增值，对应具有如下特征的数据包数：此类数据包已设置 tx_axis_tuser 以表示包结束 (EOP) 异常中止或者帧异常中止（方法是断言 tvalid 无效，但不含 tlast）。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>

## RX 路径控制信号、状态信号和统计数据信号

表 309：RX 路径控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_rx_enable_*	1	输入	<p>RX 使能（启用）。在正常操作期间，此输入必须设置为 1。当此输入设置为 0 时，RX 完成接收当前包（如果有）之后，就会停止接收包并阻止 PCS 继续对传入数据进行解码。在此模式下不报告任何统计数据，并且用户接口处于空闲状态。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>

表 309：RX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_rx_check_preamble_*	1	输入	<p>此输入断言有效时，会使 Ethernet MAC 检查接收到的帧的前导码。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_rx_check_sfd_*	1	输入	<p>此输入断言有效时，会使 Ethernet MAC 检查接收到的帧的帧起始定界符。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_rx_force_resync_*	1	输入	<p>RX 强制再同步输入。此信号用于强制 RX 路径进行复位、再同步和重新对齐。值为 1 即强制执行复位操作。值为 0 允许正常操作。</p> <p><b>注释：</b>此输入通常应为低电平 (Low) 并且仅限用于强制重新对齐时才应进行脉冲 (单周期最小脉冲)。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_rx_delete_fcs_*	1	输入	<p>启用由 RX 核执行 FCS 移除。如果将此位设置为 0，则 HSEC 核不会移除传入包的 FCS。如果此位设置为 1，则 HSEC 核将会删除接收包的 FCS。对于长度小于等于 8 字节的包，不删除 FCS。仅当对应的复位输入断言有效时，才应更改此输入。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_rx_ignore_fcs_*	1	输入	<p>启用由 RX 核在用户接口上执行 FCS 纠错。如果此位设置为 0，对于接收到的含 FCS 错误的包，将随在上一次传输期间断言有效 (tuser 和 tlast 均采样为 1) 的 tuser 管脚一起发送。如果此位设置为 1，则 HSEC 核不会在用户接口处标记 FCS 错误。</p> <p><b>注释：</b>报告的统计数据中显示此数据包有效。但 stat_rx_bad_fcs 信号会报告错误。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>
ctl_rx_max_packet_len_*	15	输入	<p>长度超过该值的任何包都将被视为大小过大。如果包的大小大于该值，则该包将被截位至该值，并且 rx_errout 信号随 rx_eopout 信号一起断言有效。小于 4 字节的包将被丢弃。该总线允许的值范围为 64 到 16,383。 ctl_rx_max_packet_len[14] 保留并且必须设置为 0。</p> <p><b>注释：</b>在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。</p>

表 309：RX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_rx_min_packet_len_*	8	输入	长度小于该值的任何包都将被视为大小过小。如果包的大小小于该值，则在 rx_eofout 断言有效的周期内，rx_errout 信号将断言有效。小于 4 字节的包将被丢弃。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。
ctl_rx_process_lfi_*	1	输入	此输入设置为 1 时，RX 核会预测从收发器传入的 LF 控制代码，并对其进行处理。此信号设置为 0 时，RX 核会忽略从收发器传入的 LF 控制代码。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或者“Ethernet MAC”时，此端口可用。
ctl_rx_test_pattern_*	1	输入	针对 RX 核启用测试模式检查。值为 1 即表示根据第 49 条中的定义启用测试模式。对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。检查扰码空闲模式。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_rx_data_pattern_select_*			对应于第 45 条中所定义的 MDIO 寄存器位 3.42.0。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”、“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”并禁用“Include FIFO Logic”时，此端口可用。
ctl_rx_test_pattern_enable_*			针对 RX 核启用测试模式。值为 1 表示启用测试模式。 对应于第 45 条中所定义的 MDIO 寄存器位 3.42.2。具有第二优先级。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”、“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”并禁用“Include FIFO Logic”时，此端口可用。
ctl_rx_prbs31_test_pattern_enable_*	1	输入	对应于第 45 条中所定义的 MDIO 寄存器位 3.42.1。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”，并且所选“Select Core”为“PCS/PMA 64-bit”时，此端口可用。
ctl_rx_custom_preamble_enable_*	1	输入	此信号断言有效时，会导致在 rx_preambleout 上呈现前导码。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”、所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”，并且禁用“Include FIFO Logic”，或者所选“Select Core”为“Ethernet MAC”时，此端口可用。
stat_rx_block_lock_*	4	输出	每个 PCS 通道的块锁定状态。值为 1 表示对应通道已达到第 82 条中定义的块锁定状态。对应于第 82.3 条中所定义的 MDIO 寄存器位 3.50.7:0 和 3.51.11:0。此输出对电平敏感。
stat_rx_framing_err_valid_*	1	输出	stat_rx_framing_err 的有效性指示符。当设为 1 时，表示 stat_rx_framing_err_0 有效。
stat_rx_framing_err_*	3	输出	RX 同步报头位成帧错误。每个 PCS 通道都有 1 个 4 位总线，用于指示针对该 PCS 通道接收到的同步报头错误的数量。仅当对应 stat_rx_framing_err_valid 设置为 1 时，该总线的值才有效。这些总线上的值可随时更新，并且这些值旨在作为递增值以供同步报头错误计数器使用。

表 309：RX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_rx_hi_ber_*	1	输出	误码率 (BER) 过高指示符。设置为 1 时，BER 过高（根据 IEEE 802.3-2015 标准的定义）。对应于第 82.3 条中所定义的 MDIO 寄存器位 3.32.1。 此输出对电平敏感。
stat_rx_bad_code_*	2	输出	此信号为递增值，对应 64B/66B 编码违例数。此信号用于指示 RX PCS 接收状态机处于 IEEE 802.3-2015 标准所指定的 RX_E 状态。此输出可用于根据第 82.3 条的定义生成 MDIO 寄存器 3.33:7:0。
stat_rx_bad_code_valid_*	1	输出	指示 stat_rx_bad_code 何时有效。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_error_valid_*	1	输出	指示 stat_rx_error 何时有效。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_error_*	8	输出	测试模式不匹配数递增值。任一周期内的非 0 值均表示 RX 核中在测试模式下发生了不匹配。仅当 ctl_rx_test_pattern 置位时，此输出才有效。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_fifo_error_*	1	输出	指示 RX FIFO 何时进入下溢或上溢状态。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_total_packets_*	2	输出	此信号为递增值，对应已收到的数据包总数。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet PCS/PMA”或“Ethernet MAC”时，此端口可用。
stat_rx_total_good_packets_*	1	输出	此信号为递增值，对应已收到的有效数据包总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_rx_total_bytes_*	6	输出	此信号为递增值，对应已收到的字节总数。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_rx_total_good_bytes_*	14	输出	此信号为递增值，对应已收到的有效字节总数。仅当完全收到不含错误的数据包时，该值才为非 0 值。 <b>注释：</b> 在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。

表 309：RX 路径控制信号、状态信号和统计数据信号（续）

名称	大小	I/O	描述
stat_rx_packet_small_*	2	输出	<p>此信号为递增值，对应所含数据小于 64 字节的所有包数。小于 4 字节的包将被丢弃。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_jabber_*	1	输出	<p>此信号为递增值，对应含无效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_large_*	1	输出	<p>此信号为递增值，对应所含数据超过 9,215 字节的所有包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_oversize_*	1	输出	<p>此信号为递增值，对应含有效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_undersize_*	2	输出	<p>此信号为递增值，对应含有效 FCS 且长度小于 <code>stat_rx_min_packet_len</code> 的数据包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_toolong_*	1	输出	<p>此信号为递增值，对应含有效和无效 FCS 且长度超过 <code>ctl_rx_max_packet_len</code> 的数据包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_fragment_*	2	输出	<p>此信号为递增值，对应含无效 FCS 且长度小于 <code>stat_rx_min_packet_len</code> 的数据包数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_64_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 64 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>

表 309：RX 路径控制信号、状态信号和统计数据信号（续）

名称	大小	I/O	描述
stat_rx_packet_65_127_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 65 到 127 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_128_255_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 128 到 255 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_256_511_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 256 到 511 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_512_1023_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 512 到 1,023 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_1024_1518_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 1,024 到 1,518 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_1519_1522_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 1,519 到 1,522 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_1523_1548_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 1,523 到 1,548 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_1549_2047_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 1,549 到 2,047 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>

表 309：RX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_rx_packet_2048_4095_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 2,048 到 4,095 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_4096_8191_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 4,096 到 8,191 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_8192_9215_bytes_*	1	输出	<p>此信号为递增值，对应已收到的有效和无效数据包（含 8,192 到 9,215 字节）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_bad_fcs_* <sup>1</sup>	2	输出	<p>无效 FCS 指示符。该总线上的值用于指示接收到的包带有无效 FCS，但不带有含 stomp 标记的 FCS。含 stomp 标记的 FCS 定义为期望的有效 FCS 的按位反转。此输出将脉冲 1 个时钟周期，以指示错误状况。脉冲可在不间断的连续周期内发生。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_packet_bad_fcs_* <sup>1</sup>	1	输出	<p>此信号为递增值，对应具有 FCS 错误的数据包（所含字节数为 64 到 ctl_rx_max_packet_len 之间）数。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_stomped_fcs_*	2	输出	<p>含 stomp 标记的 FCS 指示符。该总线上的值用于指示接收到的包带有含 stomp 标记的 FCS。含 stomp 标记的 FCS 定义为期望的有效 FCS 的按位反转。此输出将脉冲 1 个时钟周期，以指示含 stomp 标记。脉冲可在不间断的连续周期内发生。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>
stat_rx_bad_preamble_*	1	输出	<p>对应无效前导码的递增值。此信号用于指示接收到以太网数据包之前是否已接收到有效的前导码。值为 1 表示接收到的前导码无效。</p> <p>检测到无效的前导码时，stat_rx_bad_preamble 信号即断言有效，与 ctl_rx_check_preamble 信号的设置无关。</p> <p><b>注释：</b>在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。</p>

表 309：RX 路径控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_rx_bad_sfd_*	1	输出	对应无效 SFD 的递增值。此信号用于指示接收到以太网数据包之前是否已接收到有效的 SFD。值为 1 表示接收到的 SFD 无效。 检测到无效的 SFD 时，stat_rx_bad_sfd 信号即断言有效，与 ctl_rx_check_sfd 信号的设置无关。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_rx_got_signal_os_*	1	输出	Signal OS 指示。如果该位采样为 1，则表示接收到“Signal OS”字。在以太网网络中不应接收到 Signal OS。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_rx_test_pattern_mismatch_*	2	输出	测试模式不匹配数递增值。任一周期内的非 0 值均表示 RX 核中在测试模式下发生的不匹配次数。仅当 ctl_rx_test_pattern 设置为 1 时，此输出才有效。此输出可用于根据第 82.3 条的定义生成 MDIO 寄存器 3.43.15:0。此输出将脉冲 1 个时钟周期。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_truncated_*	1	输出	包截位指示符。值为 1 表示当前进行中的包因长度超过 ctl_rx_max_packet_len[14:0] 而被截位。此输出将脉冲 1 个时钟周期，以指示截位状况。脉冲可在不间断的连续周期内发生。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”或“Ethernet MAC”时，此端口可用。
stat_rx_local_fault_*	1	输出	当 stat_rx_internal_local_fault 或 stat_rx_received_local_fault 断言有效时，此输出为高电平 (High)。此输出对电平敏感。
stat_rx_remote_fault_*	1	输出	远端故障指示状态。如果该位采样为 1，则表示检测到远端故障状况。如果该位采样为 0，则表示不存在远端故障状况。此输出对电平敏感。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_internal_local_fault_*	1	输出	当由于以下任一操作导致生成内部本地故障时，此信号将转为高电平 (High)：测试模式生成、通道对齐错误或误码率过高。只要故障状况仍然存在，此信号就会保持高电平 (High)。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_received_local_fault_*	1	输出	当接收到来自链路伙伴的本地故障字数足以触发 IEEE 故障状态机所指定的故障条件时，此信号将转为高电平 (High)。只要故障状况仍然存在，此信号就会保持高电平 (High)。  注释：在“Configuration”选项卡下所选“Select Core”为“Ethernet MAC+PCS/PMA-32/64-bit”时，此端口可用。
stat_rx_valid_ctrl_code_*	1	输出	表示已接收到含有效控制代码的 PCS 块。

表 309：RX 路径控制信号、状态信号和统计数据信号（续）

名称	大小	I/O	描述
stat_rx_status	1	输出	表示链路状态。

**注释：**

1. stat\_rx\_bad\_fcs 在单个时钟周期内可以递增两次，前提是含错误 FCS 的包后紧接一个超短包。  
stat\_rx\_packet\_bad\_fcs 不会对此类超短包进行计数，因此后一个信号每个时钟只能递增一次。

## TX 暂停接口控制信号、状态信号和统计数据信号

在“MAC Options”（MAC 选项）选项卡下选中“Enable TX Flow Control Logic”（启用 TX 流量控制逻辑）并且所选“Select Core”（核选择）为 Ethernet MAC+PCS/PMA 64/32 位时，本节下所述端口可用。

表 310：TX 暂停接口控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_tx_pause_req_*	9	输入	如果该总线的某个位设置为 1，那么核会使用 ctl_tx_pause_quanta[8:0][15:0] 总线上关联的暂停量值来发射暂停包。如果 bit[8] 设置为 1，那么将发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。
ctl_tx_pause_enable_*	9	输入	TX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。此信号用于对暂停包的发射进行门控。 <sup>1</sup>
ctl_tx_resend_pause_*	1	输入	重新发射暂挂的暂停包。当此输入采样为 1 时，所有挂起的暂停包都将尽快（即，在当前进行中的包完成后）重新发射，并且重新发射计数器将复位。此输入应脉冲至 1，并且每次保持 1 个周期不变。
ctl_tx_pause_quanta0_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta1_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta2_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta3_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta4_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta5_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta6_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta7_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_quanta8_*	16	输入	这些总线用于指示基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级要发射的暂停量。ctl_tx_pause_quanta[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer0_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>

表 310：TX 暂停接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_tx_pause_refresh_timer1_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer2_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer3_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer4_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer5_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer6_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer7_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_pause_refresh_timer8_*	16	输入	此总线用于设置基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中每个优先级，暂停包的重新发射时间。ctl_tx_pause_refresh_timer[8] 的值用于全局暂停操作。所有其它值均用于优先暂停操作。 <sup>1</sup>
ctl_tx_da_gpp_*	48	输入	发射全局暂停包的目标地址。 <sup>1</sup>
ctl_tx_sa_gpp_*	48	输入	发射全局暂停包的源地址。 <sup>1</sup>
ctl_tx_etherstype_gpp_*	16	输入	发射全局暂停包的以太类型 (Etherstype)。 <sup>1</sup>
ctl_tx_opcode_gpp_*	16	输入	发射全局暂停包的操作代码。 <sup>1</sup>
ctl_tx_da_ppp_*	48	输入	发射优先暂停包的目标地址。 <sup>1</sup>
ctl_tx_sa_ppp_*	48	输入	发射优先暂停包的源地址。 <sup>1</sup>
ctl_tx_etherstype_ppp_*	16	输入	发射优先暂停包的以太类型 (Etherstype)。 <sup>1</sup>
ctl_tx_opcode_ppp_*	16	输入	发射优先暂停包的操作代码。 <sup>1</sup>
stat_tx_pause_valid_*	9	输出	如果该总线的某个位设置为 1，则表示 HSEC 核已发射暂停包。如果 bit[8] 设置为 1，则表示已发射全局暂停包。所有其它位设置为 1 都会导致发射优先暂停包。
stat_tx_unicast_* <sup>2</sup>	1	输出	此信号为递增值，对应有效单播数据包数。
stat_tx_multicast_*	1	输出	此信号为递增值，对应有效多播数据包数。
stat_tx_broadcast_*	1	输出	此信号为递增值，对应有效广播数据包数。
stat_tx_vlan_*	1	输出	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。
stat_tx_pause_*	1	输出	此信号为递增值，对应含有效 FCS 的 802.3x Ethernet MAC 暂停包数。
stat_tx_user_pause_*	1	输出	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。

**注释：**

- 在“Configuration”（配置）选项卡下未选中“Include AXI4-Lite”（包含 AXI4-Lite）时，此端口可用。
- 当启用 RX 流量控制、启用“Preemption”（抢占）或者核类型为 MAC+PCS/PMA 32 位时，此端口可用。

## RX 暂停接口控制信号、状态信号和统计数据信号

在“MAC Options”（MAC 选项）选项卡下选中“Enable RX Flow Control Logic”（启用 RX 流量控制逻辑）并且所选“Select Core”为“Ethernet MAC+PCS/PMA 64/32-bit”时，本节下所述端口可用。

表 311：RX 暂停接口控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_rx_forward_control_*	1	输入	值为 1 表示该核将控制包转发给您。值为 0 会导致核删除控制包。 <sup>1</sup>
ctl_rx_pause_ack_*	9	输入	暂停确认信号。该总线用于确认收到来自用户逻辑的暂停帧。
ctl_rx_check_ack_*	1	输入	等待确认。如果此输入设置为 1，那么该核会使用 ctl_rx_pause_ack[8:0] 总线进行暂停处理。如果此输入设置为 0，则不使用 ctl_rx_pause_ack[8:0]。 <sup>1</sup>
ctl_rx_pause_enable_*	9	输入	RX 暂停使能信号。此输入用于启用对应优先级的暂停量的处理。 <sup>1</sup>
ctl_rx_enable_gcp_*	1	输入	值为 1 即表示启用全局控制包处理。 <sup>1</sup>
ctl_rx_check_mcast_gcp_*	1	输入	值为 1 即表示启用全局控制多播目标地址处理。 <sup>1</sup>
ctl_rx_check_unicast_gcp_*	1	输入	值为 1 即表示启用全局控制单播目标地址处理。 <sup>1</sup>
ctl_rx_pause_da_unicast_*	48	输入	单播目标地址，用于暂停处理。 <sup>1</sup>
ctl_rx_check_sa_gcp_*	1	输入	值为 1 即表示启用全局控制源地址处理。 <sup>1</sup>
ctl_rx_pause_sa_*	48	输入	源地址，用于暂停处理。 <sup>1</sup>
ctl_rx_check_etype_gcp_*	1	输入	值为 1 即表示启用全局控制以太类型 (Ethertype) 处理。 <sup>1</sup>
ctl_rx_etype_gcp_*	16	输入	以太类型 (Ethertype) 字段，用于全局控制处理。 <sup>1</sup>
ctl_rx_check_opcode_gcp_*	1	输入	值为 1 即表示启用全局控制操作代码 (opcode) 处理。 <sup>1</sup>
ctl_rx_opcode_min_gcp_*	16	输入	最小全局控制操作代码值。 <sup>1</sup>
ctl_rx_opcode_max_gcp_*	16	输入	最大全局控制操作代码值。 <sup>1</sup>
ctl_rx_enable_pcp_*	1	输入	值为 1 即表示启用优先控制包处理。 <sup>1</sup>
ctl_rx_check_mcast_pcp_*	1	输入	值为 1 即表示启用优先控制多播目标地址处理。 <sup>1</sup>
ctl_rx_check_unicast_pcp_*	1	输入	值为 1 即表示启用优先控制单播目标地址处理。 <sup>1</sup>
ctl_rx_pause_da_mcast_*	48	输入	多播目标地址，用于暂停处理。 <sup>1</sup>
ctl_rx_check_sa_pcp_*	1	输入	值为 1 即表示启用优先控制源地址处理。 <sup>1</sup>
ctl_rx_check_etype_pcp_*	1	输入	值为 1 即表示启用优先控制以太类型 (Ethertype) 处理。 <sup>1</sup>
ctl_rx_etype_pcp_*	16	输入	优先控制处理的以太类型 (Ethertype) 字段。 <sup>1</sup>
ctl_rx_check_opcode_pcp_*	1	输入	值为 1 即表示启用优先控制操作代码 (opcode) 处理。 <sup>1</sup>
ctl_rx_opcode_min_pcp_*	16	输入	最小优先级控制操作代码值。 <sup>1</sup>
ctl_rx_opcode_max_pcp_*	16	输入	最大优先级控制操作代码值。 <sup>1</sup>
ctl_rx_enable_gpp_*	1	输入	值为 1 即表示启用全局暂停包处理。 <sup>1</sup>
ctl_rx_check_mcast_gpp_*	1	输入	值为 1 即表示启用全局暂停多播目标地址处理。 <sup>1</sup>
ctl_rx_check_unicast_gpp_*	1	输入	值为 1 即表示启用全局暂停单播目标地址处理。 <sup>1</sup>
ctl_rx_check_sa_gpp_*	1	输入	值为 1 即表示启用全局暂停源地址处理。 <sup>1</sup>
ctl_rx_check_etype_gpp_*	1	输入	值为 1 即表示启用全局暂停以太类型 (Ethertype) 处理。 <sup>1</sup>
ctl_rx_etype_gpp_*	16	输入	以太类型 (Ethertype) 字段，用于全局暂停处理。 <sup>1</sup>
ctl_rx_check_opcode_gpp_*	1	输入	值为 1 即表示启用全局暂停操作代码 (opcode) 处理。 <sup>1</sup>
ctl_rx_opcode_gpp_*	16	输入	全局暂停操作代码值。 <sup>1</sup>
ctl_rx_enable_ppp_*	1	输入	值为 1 即表示启用优先暂停包处理。 <sup>1</sup>

表 311：RX 暂停接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_rx_check_mcast_ppp_*	1	输入	值为 1 即表示启用优先暂停多播目标地址处理。 <sup>1</sup>
ctl_rx_check_unicast_ppp_*	1	输入	值为 1 即表示启用优先暂停单播目标地址处理。 <sup>1</sup>
ctl_rx_check_sa_ppp_*	1	输入	值为 1 即表示启用优先暂停源地址处理。 <sup>1</sup>
ctl_rx_check_etype_ppp_*	1	输入	值为 1 即表示启用优先暂停以太类型 (Ethertype) 处理。 <sup>1</sup>
ctl_rx_etype_ppp_*	16	输入	以太类型 (Ethertype) 字段，用于优先暂停处理。 <sup>1</sup>
ctl_rx_check_opcode_ppp_*	1	输入	值为 1 即表示启用优先暂停操作代码 (opcode) 处理。 <sup>1</sup>
ctl_rx_opcode_ppp_*	16	输入	优先暂停操作代码值。 <sup>1</sup>
stat_rx_unicast_* <sup>2</sup>	1	输出	此信号为递增值，对应有效单播数据包数。
stat_rx_multicast_* <sup>2</sup>	1	输出	此信号为递增值，对应有效多播数据包数。
stat_rx_broadcast_* <sup>2</sup>	1	输出	此信号为递增值，对应有效广播数据包数。
stat_rx_vlan_* <sup>2</sup>	1	输出	此信号为递增值，对应含有效 802.1Q 标签的 VLAN 数据包数。
stat_rx_pause_*	1	输出	此信号为递增值，对应含有效 FCS 的 802.3x Ethernet MAC 暂停包数。
stat_rx_user_pause_*	1	输出	此信号为递增值，对应含有效 FCS 的基于优先级的暂停包数。
stat_rx_inrangeerr_* <sup>2</sup>	1	输出	此信号为递增值，对应含“Length”（长度）字段错误但具有有效 FCS 的数据包数。
stat_rx_pause_valid_*	9	输出	该总线用于指示已收到暂停包，并且 stat_rx_pause_quanta[8:0][15:0] 总线上的关联暂停量有效，且必须将其用于暂停处理。如果收到的是 802.3x Ethernet MAC 暂停包，则 bit[8] 设置为 1。
stat_rx_pause_quanta0_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta1_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta2_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta3_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta4_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta5_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta6_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta7_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。
stat_rx_pause_quanta8_*	16	输出	这些总线用于指示在基于优先级的暂停操作和全局暂停操作中，针对 8 个优先级中的每个优先级所接收到的暂停量。如果收到的是 802.3x Ethernet MAC 暂停包，那么此暂停量将置于值 [8] 内。

表 311：RX 暂停接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_rx_pause_req_*	9	输出	暂停请求信号。当 RX 收到有效的暂停帧时，它会将该总线的对应位设置为 1，其值保持为 1 不变，直至完成暂停包的处理为止。

**注释：**

- 在“Configuration”（配置）选项卡下未选中“Include AXI4-Lite”（包含 AXI4-Lite）时，此端口可用。
- 当启用 RX 流量控制、启用“Preemption”（抢占）或者核类型为 MAC+PCS/PMA 32 位时，此端口可用。

## IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号

表 312：IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_tx_systemtimerin_*	80	输入	适用于 TX 的系统定时器输入。 在正常时钟模式下，时间格式遵循 IEEE 1588 格式，以 48 位表示秒，32 位表示纳秒。 在透明时钟模式下，位 63 预期为零，位 62:16 的单位为纳秒，位 15:0 的单位为小数纳秒。请参阅 IEEE 1588v2 以了解相关表示法的定义。 此输入必须位于 TX 时钟域内。
ctl_rx_systemtimerin_*	80	输入	适用于 RX 的系统定时器输入。 在正常时钟模式下，时间格式遵循 IEEE 1588 格式，以 48 位表示秒，32 位表示纳秒。 在透明时钟模式下，位 63 预期为零，位 62:16 的单位为纳秒，位 15:0 的单位为小数纳秒。请参阅 IEEE 1588v2 以了解相关表示法的定义。 此输入必须与通道 0 RX SerDes 位于相同时钟域内。
ctl_tx_ptp_1step_enable_*	1	输入	设为 1 时，该位即启用单步操作。
ctl_tx_ptp_latency_adjust_*	11	输入	该总线可用于调整与双步时间戳有关的单步 TX 时间戳。位 [10:3] 的单位为纳秒，而位 [2:0] 的单位则为小数纳秒。 <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_ptp_transpclk_mode_*	1	输入	设为 1 时，此输入会将时间戳逻辑置于透明时钟模式下。在此模式下，系统定时器输入将解释为纠正值。TX 将根据 IEEE 1588v2 中定义的进程，把该纠正值添加到 TX 时间戳。纠正值的符号位假定为 0（正时间）。 这样对应的传入 PTP 包纠正字段应已按适当的 RX 时间戳完成调整。 <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
stat_tx_ptp_fifo_read_error_*	1	输出	发射 PTP FIFO 写入错误。此状态值为 1 表示 PTP Tag 写入期间发生错误。需 TX 路径复位才能清除此错误。
stat_tx_ptp_fifo_write_error_*	1	输出	发射 PTP FIFO 读取错误。此状态值为 1 表示 PTP Tag 读取期间发生错误。需 TX 路径复位才能清除此错误。

表 312：IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号（续）

名称	大小	I/O	描述
tx_period_ns_*	64	输出	启用时间戳时，此输出信号表示 tx_clk 的工作时间周期，其精度格式为 2 <sup>-48</sup> 。  注释： 1. 对于 10G 线速率： <ul style="list-style-type: none"><li>对于 MAC+PCS/PMA 32 位和 PCS/PMA 32 位核，tx_period_ns 的值为 'h0003333333333333。</li><li>对于 MAC+PCS/PMA 64 位和 PCS/PMA 64 位核，tx_period_ns 的值为 'h0006666666666666。</li></ul> 2. 对于 25G 线速率： <ul style="list-style-type: none"><li>tx_period_ns 的值为 'h00028F5C28F5C28F</li></ul>
rx_period_ns_*	64	输出	启用时间戳时，此输出信号表示 rx_clk 的工作时间周期，其精度格式为 2 <sup>-48</sup> 。  注释： 1. 对于 10G 线速率： <ul style="list-style-type: none"><li>对于 MAC+PCS/PMA 32 位和 PCS/PMA 32 位核，rx_period_ns 的值为 'h0003333333333333。</li><li>对于 MAC+PCS/PMA 64 位和 PCS/PMA 64 位核，rx_period_ns 的值为 'h0006666666666666。</li></ul> 2. 对于 25G 线速率： <ul style="list-style-type: none"><li>rx_period_ns 的值为 'h00028F5C28F5C28F</li></ul>
tx_ptp_1588op_in_*	2	输入	2'b00 - No operation：表示无操作，不记录时间戳，不修改帧。 2'b01 - 1-step：表示单步，应记录时间戳并将其插入帧。 2'b10 - 2-step：表示双步，应记录时间戳并使用对应双步操作的额外端口将该时间戳返回到客户端。帧本身无需修改。 2'b11 - Reserved：表示保留，相当于“无操作”。
tx_ptp_tag_field_in_*	16	输入	该字段的使用取决于 1588 操作 <ul style="list-style-type: none"><li>对于“No operation”，将忽略该字段。</li><li>对于“1-step”和“2-step”，此字段将成为标签字段。该标签值将通过使用双步操作的额外端口，与对应当前帧的时间戳一起返回至客户端。该标签值可供软件用于确保该时间戳能够与其发送的 PTP 帧（用于发射）相匹配。</li></ul>
tx_ptp_tstamp_valid_out_*	1	输出	此位用于表示 TX 上当前呈现的有效时间戳。
tx_ptp_tstamp_tag_out_*	16	输出	对应于 tx_ptp_tag_field_in[15:0] 的标签输出。
tx_ptp_tstamp_out_*	80	输出	已发射的包 SOP 的时间戳，对应于它通过捕获平面的时间。 该总线中包含的位的表示法与定时器输入相同。
rx_ptp_tstamp_valid_out_*	1	输出	此位用于表示 RX 上当前呈现的有效时间戳。 仅当核为 Ethernet MAC+PCS/PMA-32/64 位时，才显示此信号。
rx_ptp_tstamp_out_*	80	输出	已接收的包 SOP 的时间戳，对应于它通过捕获平面的时间。请注意，只要针对一个分段断言 SOP 有效，则从断言有效的时钟周期开始，此信号即生效。 该总线中包含的位的表示法与定时器输入相同。
tx_ptp_e_rxtstamp_*	64	输入	TX PTP RX 时间戳。
tx_ptp_p_rxtstamp_*	64	输入	TX PTP RX 时间戳。
rx_ptp_e_tstamp_*	80	输出	已接收的包 SOP 的时间戳，对应于它通过捕获平面的时间。请注意，只要针对一个分段断言 SOP 有效，则从断言有效的时钟周期开始，此信号即生效。该总线中包含的位的表示法与定时器输入相同。

表 312：IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号（续）

名称	大小	I/O	描述
rx_ptp_p_tstamp_*	80	输出	已接收的包 SOP 的时间戳，对应于它通过捕获平面的时间。请注意，只要针对一个分段断言 SOP 有效，则从断言有效的时钟周期开始，此信号即生效。该总线中包含的位的表示法与定时器输入相同。
tx_ptp_e_tstamp_valid_out_*	1	输出	此位用于表示 TX 上呈现的有效时间戳。
tx_ptp_p_tstamp_valid_out_*	1	输出	保留。
tx_ptp_e_tstamp_tag_out_*	16	输出	对应于 tx_ptp_tag_field_in[15:0] 的标签输出。
tx_ptp_p_tstamp_tag_out_*	16	输出	保留。
tx_ptp_upd_chksum_in_*	1	输入	TX 更新校验和值。  注释：当“MAC options”选项卡中所选“PTP Operation mode”为“one-step”时，此端口可用。
tx_ptp_tstamp_offset_in_*	16	输入	TX PTP 时间戳偏移。  注释：当“MAC options”选项卡中所选“PTP Operation mode”为“one-step”时，此端口可用。仅支持偶数值。
tx_ptp_chksum_offset_in_*	16	输入	TX PTP 校验和偏移。  注释：当“MAC options”选项卡中所选“PTP Operation mode”为“one-step”时，此端口可用。仅支持偶数值。
tx_ptp_rxtstamp_in_*	64	输入	TX PTP RX 时间戳。  注释：当“MAC options”选项卡中所选“PTP Operation mode”为“one-step”时，此端口可用。

## AN 和 LT 接口控制信号、状态信号和统计数据信号

在“Configuration”选项卡下选中“Include AN/LT Logic”时，本节下所述端口可用。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
an_clk_*	1	输入	自动协商电路的输入时钟。 此信号应为自由运行的时钟。
an_reset_*	1	输入	与 an_clk 域对应的异步高电平有效复位。
an_loc_np_data_*	48	输入	本地下一页代码字。这是设置“loc_np”输入的情况下所使用的 48 位代码字。在此数据字段中，位 NP、ACK 和 T 以及位元位置 15、14、12 和 11 将不会随下一页代码字一起传输。这些位是在 AN IP 中生成的。但位元位置 13 处的消息协议位 MP 将进行传输。
an_lp_np_data_*	48	输出	链路伙伴下一页数据信号。此 48 位代码字由 AN IP 驱动，包含来自远端链路伙伴的 48 位下一页代码字。
lt_tx_sof_*	4	输出	这是链路训练信号，在每个训练帧开始后的 1 个 tx_serdes_clk 周期内，此信号将断言有效。此信号是为需要计算训练帧数量或者需要将事件同步至训练帧的输出的应用而提供的。
ctl_autoneg_enable_*	1	输入	自动协商的使能信号。  注释：在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_autoneg_bypass_*	1	输入	输入信号，用于禁用自动协商并绕过自动协商功能。如果断言此输入有效，将关闭自动协商，但 PCS 将连接到输出以允许执行操作。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_nonce_seed_*	8	输入	8 位种子，用于初始化随机数 (nonce) 字段多项式生成器。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_pseudo_sel_*	1	输入	选择位 49 随机位生成器的多项式生成器。如果此输入为 1，则多项式为 $x^7+x^6+1$ 。如果此输入为 0，则多项式为 $x^7+x^3+1$ 。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_restart_negotiation_*	1	输入	此输入用于触发重新启动自动协商，与当前电路状态无关。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_local_fault_*	1	输入	此输入信号用于设置发射链路代码字的 remote_fault 位。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_pause_*	1	输入	此输入信号用于设置发射链路代码字的 PAUSE 位 (C0)。如果该核不支持暂停，则此信号可能不存在。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_asmdir_*	1	输入	此输入信号用于设置发射链路代码字的 ASMDIR 位 (C1)。如果该核不支持暂停，则此信号可能不存在。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_fec_10g_request_*	1	输入	此信号用于告知链路伙伴，本地工作站正在请求在 10 Gb/s 通道协议上使用第 74 条 FEC。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_fec_25g_rs_request_*	1	输入	此信号用于告知链路伙伴，本地工作站正在请求在 25 Gb/s 通道协议上使用 rs FEC (第 91 或 108 条)。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_fec_25g_baser_request_*	1	输入	此信号表示 BASE-R FEC 请求。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_an_fec_ability_override_*	1	输入	用于在发射链路代码字中设置第 74 条 FEC 功能位。如果设置此输入，则清除发射链路代码字中的 FEC 功能位。如果该 IP 核不支持第 74 条 FEC，则此信号可能存在。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_loc_np_*	1	输入	本地下一页 (Local Next Page) 信号指示符。如果此位为 1，则 AN IP 会将 loc_np_data 输入处的下一页代码字传输到远端链路伙伴。如果此位为 0，则 AN IP 不会启动下一页协议。如果链路伙伴包含多个待发送的下一页信号，并且“loc_np”位已清零，那么 AN IP 将传输空消息页面。
ctl_an_lp_np_ack_*	1	输入	链路伙伴下一页确认信号。此信号用于告知 AN IP，本地主机已读取位于 lp_np_data 输出管脚处的下一页数据（来自远端链路伙伴）。当此信号变为高电平时，表示 AN IP 确认已接收发送到远端链路伙伴的下一页代码字，并启动下一个代码字的传输。在此期间，AN IP 会移除 lp_np 信号，直至新的下一页信息可用为止。
ctl_an_cl91_fec_request_*	1	输入	该位用于请求第 91 条 FEC。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_cl91_fec_ability_*	1	输入	该位用于设置第 91 条 FEC 功能。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_an_ability_1000base_kx_*	1	输入	这些输入用于识别发射链路代码字中播发到链路伙伴的以太网协议功能。值为 1 表示接口支持此协议并播发此信息。
ctl_an_ability_10gbase_kx4_*	1	输入	
ctl_an_ability_10gbase_kr_*	1	输入	
ctl_an_ability_40gbase_kr4_*	1	输入	
ctl_an_ability_40gbase_cr4_*	1	输入	
ctl_an_ability_100gbase_cr10_*	1	输入	
ctl_an_ability_100gbase_kp4_*	1	输入	
ctl_an_ability_100gbase_kr4_*	1	输入	
ctl_an_ability_100gbase_cr4_*	1	输入	
ctl_an_ability_25gbase_krcr_s_*	1	输入	
ctl_an_ability_25gbase_krcr_*	1	输入	
ctl_an_ability_25gbase_kr1_*	1	输入	
ctl_an_ability_25gbase_cr1_*	1	输入	
ctl_an_ability_50gbase_kr2_*	1	输入	
ctl_an_ability_50gbase_cr2_*	1	输入	
ctl_lt_polynomial_select	2	输入	这些输入用于识别发射链路代码字中播发到链路伙伴的以太网协议功能。值为 1 表示接口支持此协议并播发此信息。
ctl_an_ability_2_5gbase_kx	1	输入	
ctl_an_ability_5gbase_kr	1	输入	
ctl_an_ability_50gbase_krcr	1	输入	
ctl_an_ability_100gbase_kr2cr2	1	输入	
stat_an_lp_ability_200gbase_kr4cr4	1	输入	

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_lt_training_enable_*	1	输入	启用链路训练。禁用链路训练时，所有 PCS 通道都以任务模式进行操作。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_restart_training_*	1	输入	此信号用于触发重新启动链路训练，与当前状态无关。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_rx_trained_*	4	输入	此信号断言有效即可指示接收器 FIR 滤波器系数已全部设置完成，并且训练的接收器部分已完成。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_preset_to_tx_*	4	输入	此信号用于设置预置位的值，该位用于发射到训练帧的控制块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_initialize_to_tx_*	4	输入	此信号用于设置初始位的值，该位用于发射到训练帧的控制块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_pseudo_seed0_*	11	输入	此 11 位信号用于生成训练模式生成器的种子。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_k_p1_to_tx0_*	2	输入	此 2 位字段用于设置 k+1 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_k0_to_tx0_*	2	输入	此 2 位字段用于设置 k0 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_k_m1_to_tx0_*	2	输入	此 2 位字段用于设置 k-1 系数更新字段的值，该字段用于发送到训练帧的控制块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_stat_p1_to_tx0_*	2	输入	此 2 位字段用于设置 k+1 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。  <b>注释：</b> 在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_lt_stat0_to_tx0_*	2	输入	此 2 位字段用于设置 k0 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。  注释：在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
ctl_lt_stat_m1_to_tx0_*	2	输入	此 2 位字段用于设置 k-1 系数更新状态的值，该状态用于发送到训练帧的状态块中的链路伙伴。  注释：在“Configuration”选项卡下未选中“Include AXI4-Lite”时，此端口可用。
stat_an_link_cntl_1000base_kx_*	2	输出	来自各种以太网协议的自动协商控制器的链路控制输出。设置如下： 00: DISABLE; PCS 断开连接； 01: SCAN_FOR_CARRIER; RX 已连接到 PCS； 11: ENABLE; PCS 已连接，可执行任务模式操作。 10: 不使用
stat_an_link_cntl_10gbase_kx4_*	2	输出	
stat_an_link_cntl_10gbase_kr_*	2	输出	
stat_an_link_cntl_40gbase_kr4_*	2	输出	
stat_an_link_cntl_40gbase_cr4_*	2	输出	
stat_an_link_cntl_100gbase_cr10_*	2	输出	
stat_an_link_cntl_100gbase_kp4_*	2	输出	
stat_an_link_cntl_100gbase_kr4_*	2	输出	
stat_an_link_cntl_100gbase_cr4_*	2	输出	
stat_an_link_cntl_25gbase_krcr_s_*	2	输出	
stat_an_link_cntl_25gbase_krcr_*	2	输出	
stat_an_link_cntl_25gbase_kr1_*	2	输出	
stat_an_link_cntl_25gbase_cr1_*	2	输出	
stat_an_link_cntl_50gbase_kr2_*	2	输出	
stat_an_link_cntl_50gbase_cr2_*	2	输出	
stat_an_link_cntl_2_5gbase_kx	2	输出	来自各种以太网协议的自动协商控制器的链路控制输出。设置如下： 00: DISABLE; PCS 断开连接； 01: SCAN_FOR_CARRIER; RX 已连接到 PCS； 11: ENABLE; PCS 已连接，可执行任务模式操作。 10: 不使用
stat_an_link_cntl_5gbase_kr	2	输出	
stat_an_link_cntl_50gbase_krcr	2	输出	
stat_an_link_cntl_100gbase_kr2cr2	2	输出	
stat_an_link_cntl_200gbase_kr4cr4	2	输出	
stat_an_fec_enable_*	1	输出	用于支持在链路上使用第 74 条 FEC。
stat_an_tx_pause_enable_*	1	输出	用于支持在发射路径中进行站对站（全局）暂停包生成，以控制接收路径中的数据流。
stat_an_rx_pause_enable_*	1	输出	用于支持在接收路径中进行站对站（全局）暂停包解释，以控制来自发射器的数据流。
stat_an_autoneg_complete_*	1	输出	指示自动协商已完成，并且已接收到来自 PCS 的 rx 链路状态。
stat_an_parallel_detection_fault_*	1	输出	指示自动协商期间发生并行检测故障。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_an_lp_ability_1000base_kx_*	1	输出	
stat_an_lp_ability_10gbase_kx4_*	1	输出	
stat_an_lp_ability_10gbase_kr_*	1	输出	
stat_an_lp_ability_40gbase_kr4_*	1	输出	
stat_an_lp_ability_40gbase_cr4_*	1	输出	
stat_an_lp_ability_100gbase_cr10_*	1	输出	
stat_an_lp_ability_100gbase_kp4_*	1	输出	
stat_an_lp_ability_100gbase_kr4_*	1	输出	
stat_an_lp_ability_100gbase_cr4_*	1	输出	
stat_an_lp_ability_25gbase_krcr_s_*	1	输出	
stat_an_lp_ability_25gbase_krcr_*	1	输出	
stat_an_lp_ability_2_5gbase_kx	1	输出	这些信号用于指示从链路伙伴播发的协议。当 stat_an_lp_ability_valid 输出信号断言有效时，这些信号全部变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_lp_ability_5gbase_kr	1	输出	
stat_an_lp_ability_50gbase_krcr	1	输出	
stat_an_lp_ability_100gbase_kr2cr2	1	输出	
stat_an_lp_ability_200gbase_kr4cr4	1	输出	
stat_an_lp_ability_25gbase_cr1_*	1	输出	用于指示从链路伙伴播发的协议。当 stat_AN_lp_Extended_Ability_Valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_rxcdrhold_*	1	输出	用于指示 rx cdr hold 信号。
stat_an_lp_pause_*	1	输出	此信号用于指示来自链路伙伴的接收链路代码字的 PAUSE 位 (C0) 的播发值。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_asm_dir_*	1	输出	此信号用于指示来自链路伙伴的接收链路代码字的 ASMDIR 位 (C1) 的播发值。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_rf_*	1	输出	此位用于指示链路伙伴远端故障。
stat_an_lp_fec_10g_ability_*	1	输出	此信号用于指示与链路伙伴正在播发的 10Gb/s 通道协议相关联的第 74 条 FEC 功能。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_fec_10g_request_*	1	输出	此信号用于指示链路伙伴正在请求在 10 Gb/s 通道协议上使用第 74 条 FEC。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_fec_25g_rs_request_*	1	输出	此信号用于指示链路伙伴正在请求在 25 Gb/s 通道协议上使用第 91 条 (或第 108 条) rs FEC。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_fec_25g_baser_request_*	1	输出	此信号指示链路伙伴正在请求在 25 Gb/s 通道 base-r 协议上使用第 74 条 FEC。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_autoneg_able_*	1	输出	此输出信号用于指示链路伙伴能够执行自动协商。当 stat_an_lp_ability_valid 输出信号断言有效时，此信号即变为有效。
stat_an_lp_ability_valid_*	1	输出	此信号用于指示何时所有链路伙伴播发将变为有效。

表 313：AN 和 LT 接口控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
stat_an_loc_np_ack_*	1	输出	此信号可指示用于接收 loc_np_data 输入管脚上显示的本地下一页数据的本地主机。 当 AN IP 在 loc_np_data 输入管脚上对下一页数据进行采样时，此信号将脉冲至高电平并保持 1 个时钟周期。当本地主机检测到此信号为高电平时，它必须将位于“loc_np_data”输入管脚处的 48 位下一页代码字替换为要发送的下一个 48 位代码字。如果本地主机没有其它下一页要发送，则必须将 loc_np 输入清零。
stat_an_lp_np_*	1	输出	链路伙伴下一页信号。此信号用于指示在 lp_np_data 输出管脚处存在来自远端链路伙伴的有效的 48 位下一页代码字。当 lp_np_ack 输入信号被驱动至高电平时，此信号将被驱动至低电平，表示本地主机已读取下一页数据。它将保持低电平，直至在 lp_np_data 输出管脚上下一个代码字变为可用为止，届时 lp_np 输出将被重新驱动至高电平。
stat_an_lp_ability_25gbase_kr1_*	1	输出	用于指示从链路伙伴播发的协议。当 stat_an_lp_extended_ability_valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_link_cntl_25gbase_cr1_*	1	输出	用于指示从链路伙伴播发的协议。当 stat_an_lp_extended_ability_valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_lp_ability_50gbase_kr2_*	1	输出	用于指示从链路伙伴播发的协议。当 stat_an_lp_extended_ability_valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_lp_ability_50gbase_cr2_*	1	输出	用于指示从链路伙伴播发的协议。当 stat_an_lp_extended_ability_valid 输出信号断言有效时，此信号即变为有效。值为 1 表示此协议已作为链路伙伴支持的协议进行播发。
stat_an_lp_ability_extended_fec_*	4	输出	此输出用于指示 FEC 扩展功能。
stat_an_rs_fec_enable_*	1	输出	用于支持在链路上使用第 91 条 FEC。
stat_an_lp_extended_ability_valid_*	1	输出	当该位为 1 时，表示检测到的扩展功能有效。
stat_lt_signal_detect_*	4	输出	此信号用于指示相应的链路训练状态机已进入 SEND_DATA 状态，在此状态下可恢复正常 PCS 操作。
stat_lt_training_*	4	输出	此信号用于指示何时相应的链路训练状态机正在执行链路训练。
stat_lt_training_fail_*	4	输出	此信号在链路训练期间断言有效，前提是对应的链路训练状态机在训练周期期间检测到超时。
stat_lt_rx_sof_*	4	输出	此输出将在 1 个 RX SerDes 时钟周期内保持高电平，以指示链路训练帧开始。
stat_lt_frame_lock_*	4	输出	链路训练开始后，将针对每个 PMD 通道断言这些信号有效，前提是对应的链路训练接收器能够与链路伙伴建立帧同步。
stat_lt_preset_from_rx_*	4	输出	此信号用于反映从控制块中的链路伙伴接收到的预置控制位的值。
stat_lt_initialize_from_rx_*	4	输出	此信号用于反映从控制块中的链路伙伴接收到的初始化控制位的值。
stat_lt_k_p1_from_rx0_*	2	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k+1 系数的更新控制位。
stat_lt_k0_from_rx0_*	2	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k0 系数的更新控制位。
stat_lt_k_m1_from_rx0_*	2	输出	此 2 位字段用于指示从控制块中的链路伙伴接收到的 k-1 系数的更新控制位。
stat_lt_stat_p1_from_rx0_*	2	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k+1 系数的更新状态位。
stat_lt_stat0_from_rx0_*	2	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k0 系数的更新状态位。
stat_lt_stat_m1_from_rx0_*	2	输出	此 2 位字段用于指示从状态块中的链路伙伴接收到的 k-1 系数的更新状态位。

## 第 74 条 FEC 接口控制信号、状态信号和统计数据信号

在“Configuration”（配置）选项卡下选中“Clause 74 (BASE-KRFEC)”时，下表中的端口可用。

表 314：第 74 条 FEC 接口控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_fec_tx_enable_*	1	输入	此信号断言有效即可对发射的数据启用第 74 条 FEC 编码。
ctl_fec_rx_enable_*	1	输入	此信号断言有效即可对接收的数据启用第 74 条 FEC 解码。
ctl_fec_enable_error_to_pcs_*	1	输入	第 74 条 FEC 启用 PCS 错误。
stat_fec_inc_correct_count_*	4	输出	此信号将每隔约 32 个代码字断言有效，前提是同时断言 <code>ctl_rx_fec_enable</code> 有效，且 FEC 解码器已检测到并已纠正对应帧中的位元错误。
stat_fec_inc_cant_correct_count_*	4	输出	此信号将每隔约 32 个代码字断言有效，前提是 <code>ctl_rx_fec_enable</code> 断言有效，并且 FEC 解码器已检测到该位
stat_fec_lock_error_*	4	输出	<code>stat_fec_lock_error_*</code> 断言有效，前提是 FEC 解码器经过约 5 ms 后仍无法检测到帧边界。检测到帧边界后，即清除此信号。
stat_fec_rx_lock_*	4	输出	此信号断言有效的前提是同时断言 <code>ctl_fec_rx_enable</code> 有效，且 FEC 解码器可检测到帧边界。

## IEEE 第 108 条 (RS-FEC) 控制信号、状态信号和统计数据信号

在“Configuration”（配置）选项卡下选中“IEEE Clause 108 (RS-FEC)”时，本节下所述端口可用。

表 315：IEEE 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号

名称	大小	I/O	描述
ctl_rx_rsfec_enable_correction_*	1	输入	当 <code>rx_resetn</code> 已断言低电平有效 ( <code>~rx_serdes_reset</code> ) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 等效于 MDIO 寄存器 1.200.0 <ul style="list-style-type: none"><li>· 0：解码器执行错误检测，但不执行纠错（请参阅 IEEE 802.3by 第 91.5.3.3 条）。</li><li>· 1：解码器还会执行纠错。</li></ul>
ctl_rx_rsfec_enable_indication_*	1	输入	当 <code>rx_resetn</code> 已断言低电平有效 ( <code>~rx_serdes_reset</code> ) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 等效于 MDIO 寄存器 1.200.1 <ul style="list-style-type: none"><li>· 0：绕过错误指示功能（请参阅 IEEE 802.3by 标准第 91.5.3.3 条）。</li><li>· 1：解码器向 PCS 子层发出错误指示。</li></ul>
ctl_rsfec_enable_*	1	输入	当 <code>rx_resetn</code> 已断言低电平有效 ( <code>~rx_serdes_reset</code> ) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 启用 RS-FEC 功能。 <b>注释：</b> 10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 使能信号。
ctl_rsfec_ieee_error_indication_mode_*	1	输入	当 <code>rx_resetn</code> 已断言低电平有效 ( <code>~rx_serdes_reset</code> ) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。 <ul style="list-style-type: none"><li>· 1：核遵循 IEEE RS-FEC 规范。</li><li>· 0：如果 <code>ctl_rx_rsfec_enable_correction</code> 和 <code>ctl_rx_rsfec_enable_indication</code> 均设为 0，则绕过 RS 解码器。</li></ul>

表 315：IEEE 第 108 条 (RS/FEC) 控制信号、状态信号和统计数据信号 (续)

名称	大小	I/O	描述
ctl_rsfec_consortium_25g_*	1	输入	<p>在 IEEE 第 108 条与 25G 以太网联盟模式之间进行切换。</p> <p>当 rx_resetn 已断言低电平有效 (~rx_serdes_reset) 后，该位上的设置即生效。复位时，在首个周期内将对新的值进行采样。</p> <ul style="list-style-type: none"> <li>· 1 = 25G 以太网联盟规范模式。</li> <li>· 0 = IEEE 802.3by 模式。</li> </ul> <p><b>注释：</b>10G/25G 子系统的部分变体可能包含独立的 RX 和 TX 联盟信号。</p>
stat_rx_rsfec_hi_ser_*	1	输出	如果含 8192 个代码字的窗口内的 RS-FEC 符号错误数量超过阈 K = 417，则此项设置为 1，否则设为 0。
stat_rx_rsfec_lane_alignment_status_*	1	输出	值为 1 表示 RX RS-FEC 块在来自收发器的数据上已达成对齐。
stat_rx_rsfec_corrected_cw_inc_*	1	输出	此信号为递增值，对应已纠正的错误数。
stat_rx_rsfec_uncorrected_cw_inc_*	1	输出	此信号为递增值，对应未纠正的错误数。
stat_rx_rsfec_err_count0_inc_*	3	输出	此信号为递增值，对应已检测到的错误数。
stat_tx_rsfec_lane_alignment_status_*	1	输出	值为 1 表示 TX RS-FEC 块在传入 PCS 数据上已达成对齐。

## PLL 和 SYS 时钟

表 316：选中“RUNTIME SWITCH”功能时的 PLL 和 SYS 时钟选择线路

名称	大小	I/O	描述
gt_drp_done_*	1	输入	<p>指示 GT DRP 操作完成。用于复位 GT 模块。</p> <p><b>注释：</b>在“Configuration”（配置）选项卡下选中“Runtime Switchable Mode”（运行时可切换模式）时，此端口可用。</p>
txpllclksel_in_*	1	输入	<p>TX PLL 时钟选择线路。</p> <p><b>注释：</b>在“Configuration”（配置）选项卡下选中“Runtime Switchable Mode”（运行时可切换模式）时，此端口可用。</p>
rxpllclksel_in_*	1	输入	<p>RX PLL 时钟选择线路。</p> <p>在“Configuration”选项卡下选中“Runtime Switchable Mode”时，此端口可用。</p>
txsysclksel_in_*	1	输入	<p>选择 TX PLL 时钟源以驱动 txoutclk。</p> <p><b>注释：</b>在“Configuration”选项卡下选中“Runtime Switchable Mode”时，此端口可用。</p>
rxsysclksel_in_*	1	输入	<p>选择 RX PLL 时钟源以驱动 rxoutclk</p> <p><b>注释：</b>在“Configuration”选项卡下选中“Runtime Switchable Mode”时，此端口可用。</p>
rxafecfoken_*	1	输入	仅适用于 UltraScale+ 器件
rxdfecfokfcnum_*	1	输入	仅适用于 UltraScale+ 器件

表 316：选中“RUNTIME SWITCH”功能时的 PLL 和 SYS 时钟选择线路（续）

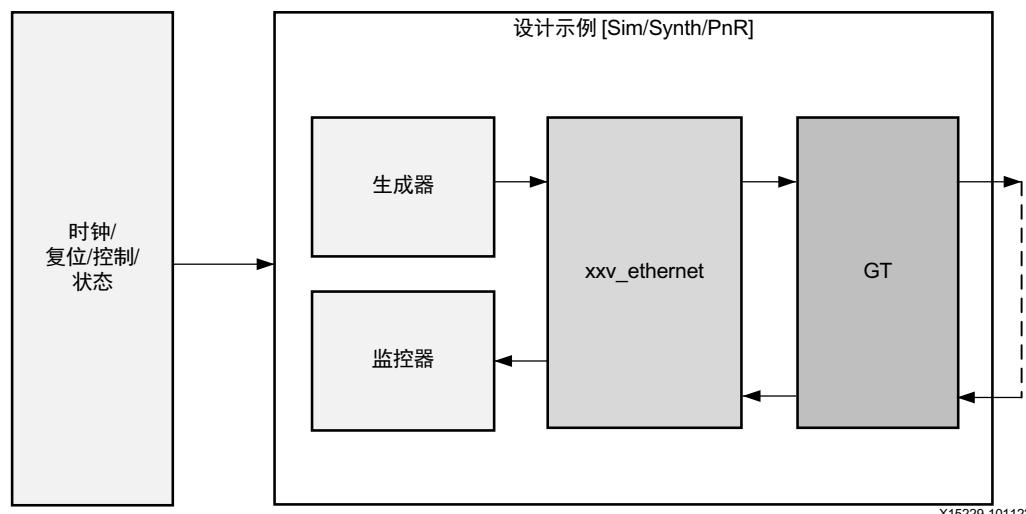
名称	大小	I/O	描述
speed_*	1	输入	此信号用于指示核的处理速度： 1'b1 = 10G 和 1'b0 = 25G  注释：在“Configuration”（配置）选项卡下选中“Runtime Switchable Mode”（运行时可切换模式）时，此端口可用。
anlt_done_*	1	输出	指示自动协商和链路训练完成。  注释：在“Configuration”（配置）选项卡下选中“Runtime Switchable Mode”（运行时可切换模式）时，此端口可用。
axi_ctl_core_mode_switch_*	1	输出	在“Configuration”选项卡下选中“Include AXI4-Lite”（包含 AXI4-Lite）并将 1 写入 0x0138 自清除寄存器以启动 GT DRP 操作时，此信号可用于在 10G 和 25G 之间切换线速率。
user_reg0_*	32	输出	来自 AXI4-Lite 寄存器映射 user_reg0 寄存器的用户定义的信号。  注释：在“Configuration”选项卡下选中“Include AXI4-Lite interface”（包含 AXI4-Lite 接口）时，此端口可用。
gt_drp_grant_*	1	输出	此信号表示 GT DRP 接口处于繁忙状态还是空闲状态。 1 表示 GT DRP 接口已准备就绪，可供读写访问。 0 表示 GT DRP 接口正忙于处理其它传输事务，无法进行读写访问。

## 双工操作模式

在此操作模式下，核的发射器和接收器都处于有效状态，在 GT 输出接口上提供环回功能，即输出作为输入进行回馈。包生成和监控模块在此模式下同样处于有效状态。生成器模块负责生成所需数量的包，并利用可用的数据接口发送到核。监控模块检查来自接收器的包。

下图显示了双工操作模式。

图 52：双工操作模式



## 运行时可切换

通过此配置可随时在 10G 和 25G 之间灵活切换线速率。要激活此功能，请在“Configuration”（配置）选项卡下选中“Runtime Switchable mode”（运行时可切换模式）选项复选框。选中此选项时，可使用 ch0\_txrate 和 ch0\_rxrate 端口来更改速度。在内部，GT IP 将执行必要的 DRP 写入操作来达到所需的速度。

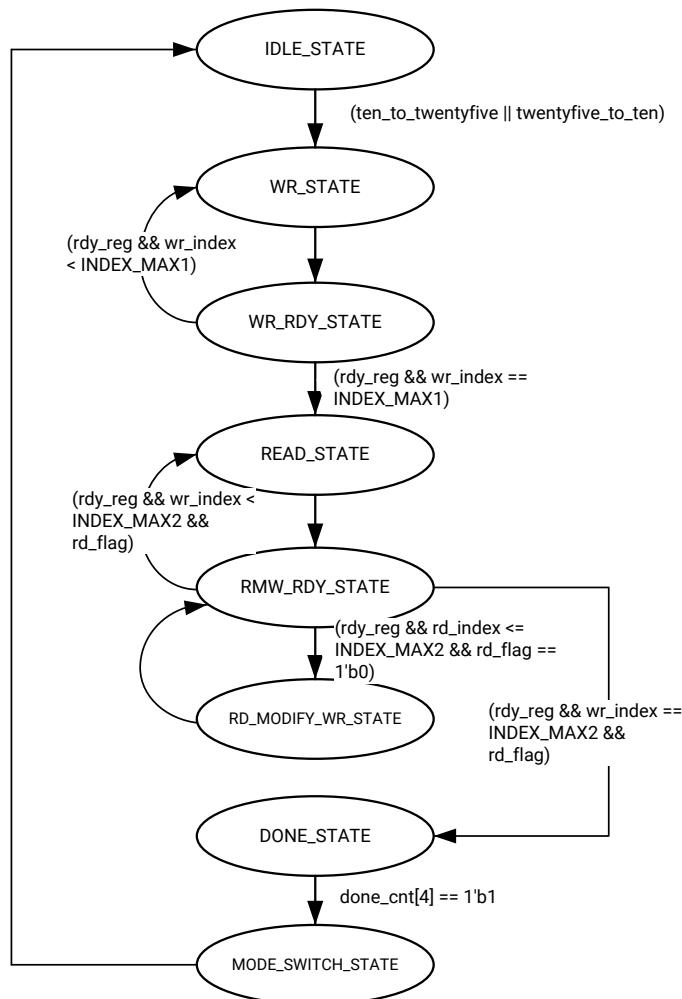
选中此选项时，设计示例的 \*\_pkt\_gen\_mon.v 模块内将显示 \*\_trans\_debug 模块。此 \*\_trans\_debug 模块负责执行所有 GT DRP 写入操作以切换收发器模式，即从 10G 切换至 25G 或从 25G 切换至 10G。将 mode\_change\_\* 输入信号设置为高电平并保持 2 个时钟周期，然后再将其设置为低电平时，它会为特定的核针对 GT 通道启动 DRP 写入操作，然后将该核复位。

**注释：**对于切换操作，您应将 trans\_debug 模块复制到自己的设计中。

仅针对该通道执行 DRP 写入操作。对于线速率 25G，公用端口固定为 QPLL0，线速率 10G 则固定为 QPLL1。

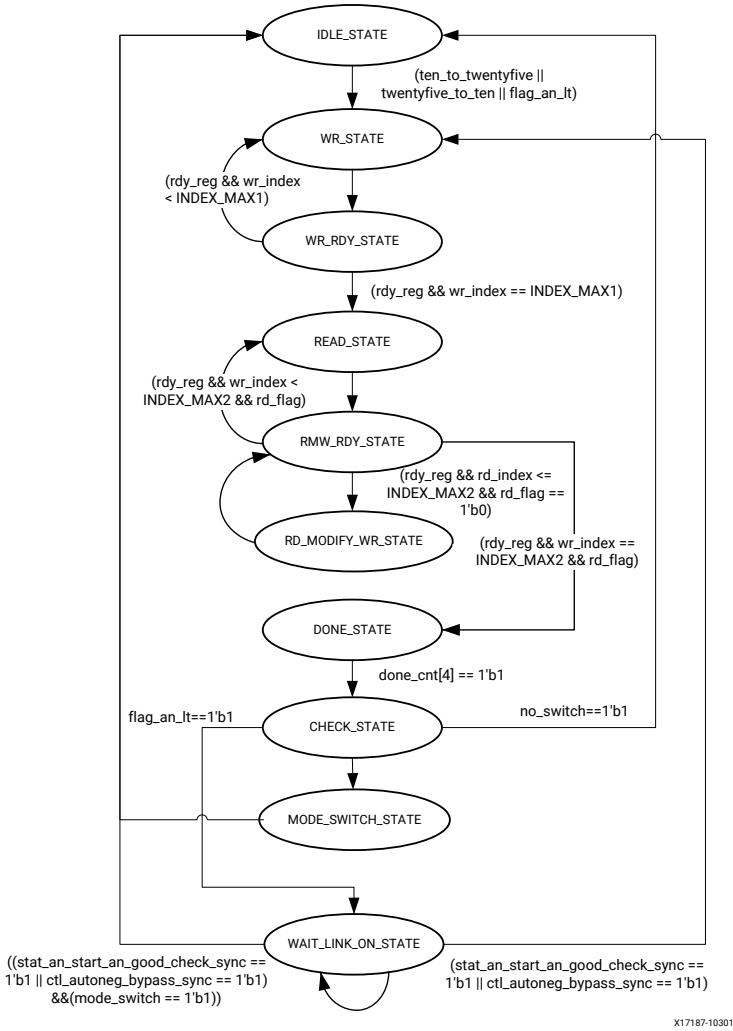
下图显示了在此进程中发生的状态：

图 53：不包含 AN/LT 时的运行时可切换 DRP 操作状态转换图



X17068-051716

图 54：包含 AN/LT 时的运行时可切换 DRP 操作状态转换图



## 共享逻辑实现

共享逻辑包含所有可共享模块，这些模块显示为核的一部分或者显示在设计示例中。

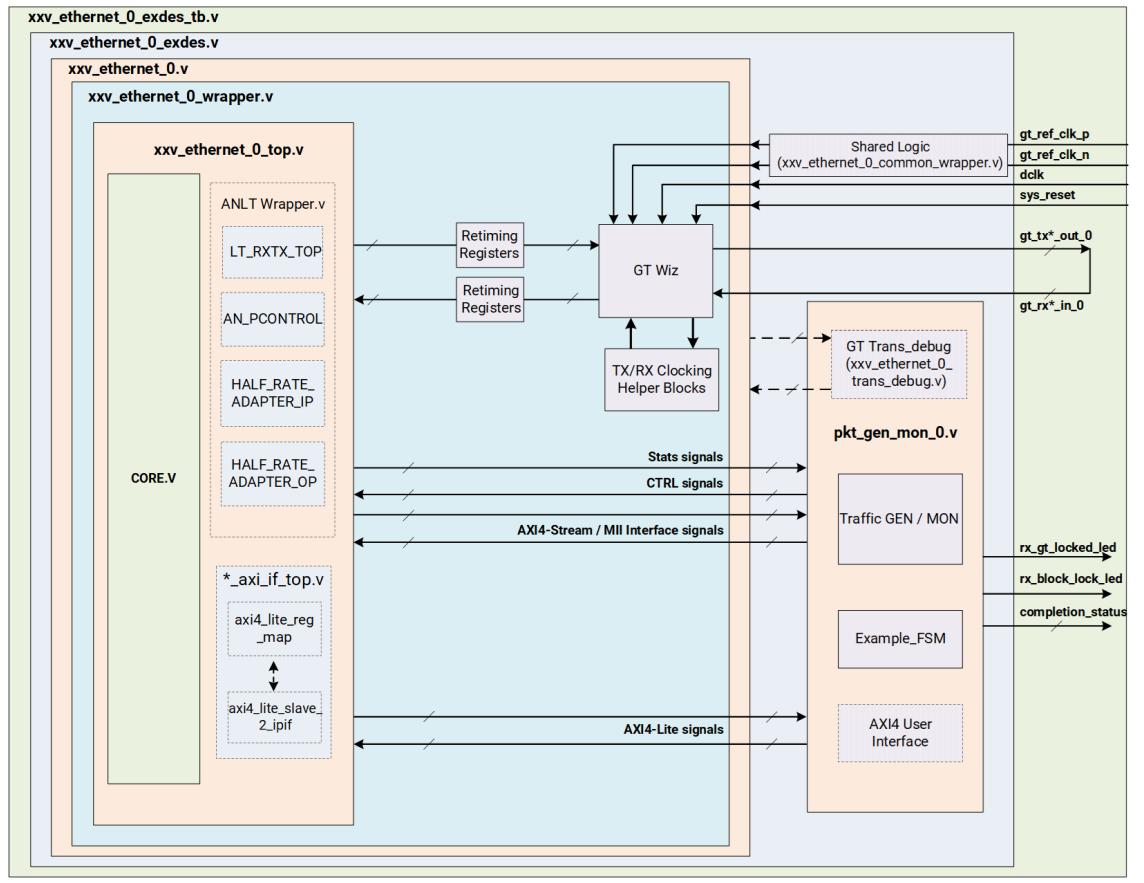
默认情况下，GT 公用模块、复位逻辑和时钟模块显示在 IP 核内部。满足以下条件时，这些模块将布局在核外部，以便与其它设计共享。

- 在“GT Selection and Configuration”（GT 选择和配置）选项卡下选中“Include GT subcore in example design”（在设计示例中包含 GT 子核）选项时。
- 在“Shared Logic”（共享逻辑）选项卡下选中“Include Shared Logic in Example Design”（在设计示例中包含共享逻辑）选项时。

选择设计示例中的共享逻辑时，将在 `xxv_ether*_exdes.v` 和 DUT（即，`xxv_ether*_*.v`）之间例化新的 `xxv_ether*_core_support.v` 模块。此模块将包含可在多个设计间共享的所有子模块。

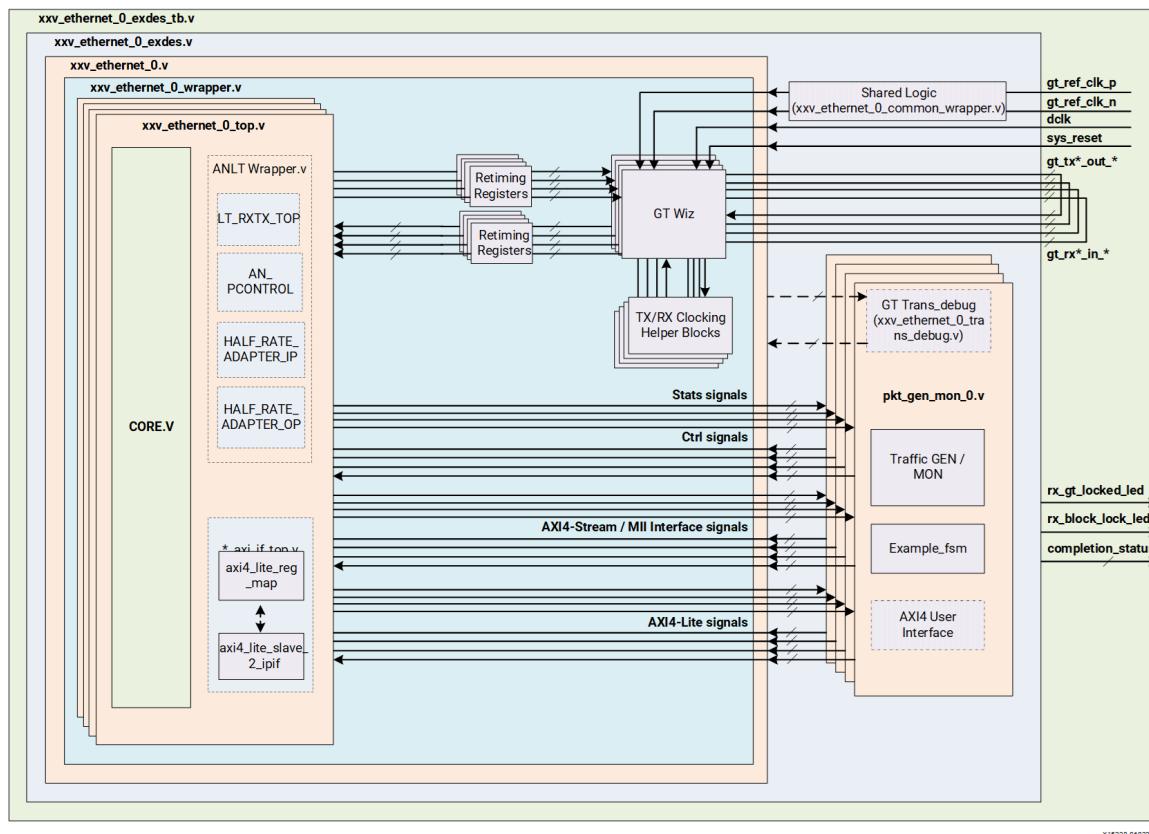
下图显示单核设计示例内部例化共享逻辑时的实现过程。

图 55：含共享逻辑实现的单核设计示例层级



下图显示了多核设计示例内部例化共享逻辑时的实现过程。

图 56：含共享逻辑实现的多核设计示例层级



下列模块是共享逻辑封装器的一部分。

- \*\_clocking\_wrapper：此模块包含可与其它设计共享的所有时钟资源。
- \*\_common\_wrapper：此模块包含可与其它设计共享的 GT 公用模块。
- \*\_reset\_wrapper：此模块包含适用于选定的 Vivado IDE 配置的所有复位逻辑。

## AXI4-Lite 接口实现

为了例化 AXI4-Lite 接口以访问 xxv\_ethernet\_0 核的控制寄存器和状态寄存器，请勾选 Vivado IDE 的“Configuration”选项卡中的“Include AXI4-Lite”（包含 AXI4-Lite）复选框。该选项可启用 xxv\_ethernet\_0\_axi\_if\_top 模块（其中包含 xxv\_ethernet\_0\_pif\_registers 以及 xxv\_ethernet\_0\_slave\_2\_ipif 模块）。您可从 xxv\_ethernet\_0\_pkt\_gen\_mon 模块访问 AXI4-Lite 接口逻辑寄存器（控制寄存器、状态寄存器和统计数据寄存器）。

此模式会启用以下功能：

- 您可通过 AXI4-Lite 接口配置该核的所有控制 (CTL) 端口。此操作的执行方式是将含所需数据的一组地址位置写入寄存器映射接口。
- 您可通过 AXI4-Lite 接口从该核访问所有状态寄存器和统计数据寄存器。此操作的执行方式是通过寄存器映射读取状态寄存器和统计数据寄存器的地址位置。

## .h 报头文件

在 Vivado Design Suite 中启用“Include AXI4-Lite”（包含 AXI4-Lite）的情况下生成 IP 时，以头文件 (.h) 格式提供 AXI4 寄存器信息，例如，寄存器地址、寄存器名称（含位的位置）、掩码值、访问类型及其默认值，并且，在工程路径的 header\_files 文件夹下可找到此头文件。

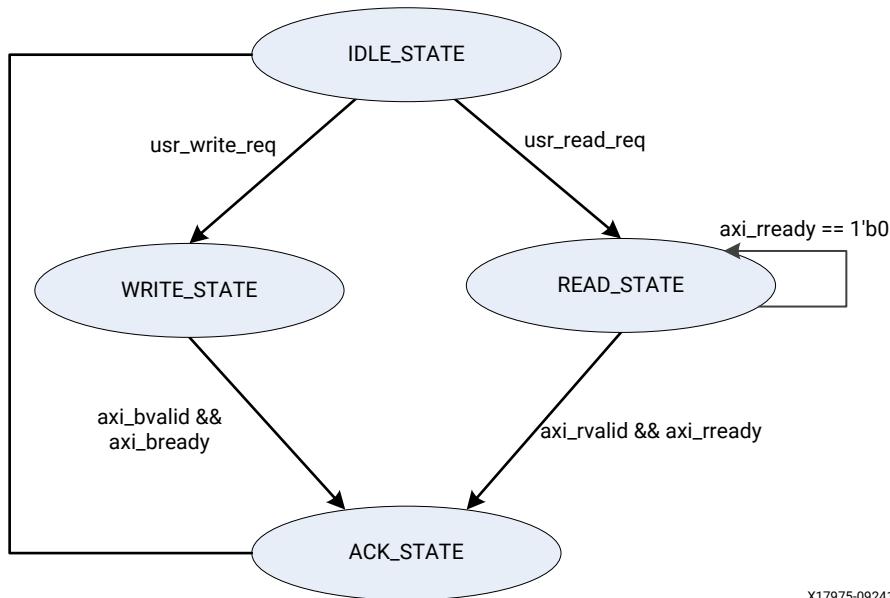
## AXI4 接口用户逻辑

以下章节提供了 AXI4-Lite 接口状态机控制和端口。

### 用户状态机

通过 AXI4-Lite 从模块接口执行的读写操作由状态机来控制，如下所示：

图 57：AXI4-Lite 接口的用户状态机



X17975-092416

每个状态的功能描述如下：

- **IDLE\_STATE**：默认情况下，FSM 处于 IDLE\_STATE 状态。当 user\_read\_req 信号变为高电平 (High) 时，它会转至 READ\_STATE 状态，或者如果 user\_write\_req 信号为 High，那么它会转为 WRITE\_STATE 状态，否则它将保持 IDLE\_STATE 状态。
- **WRITE\_STATE**：在此状态下，您提供 S\_AXI\_AWVALID、S\_AXI\_AWADDR、S\_AXI\_WVALID、S\_AXI\_WDATA 和 S\_AXI\_WSTRB 以通过 AXI 写入寄存器映射。当来自 AXI 从接口的 S\_AXI\_BVALID 和 S\_AXI\_BREADY 均为 High 时，它会转至 ACK\_STATE。如果在任意违规地址中发生任何写入操作，S\_AXI\_BRESP[1:0] 会指示 2'b10 以断言写入错误信号有效。
- **READ\_STATE**：在此状态下，您提供 S\_AXI\_ARVALID 和 S\_AXI\_ARADDR 以通过 AXI 从寄存器映射读取。当 S\_AXI\_RVALID 和 S\_AXI\_RREADY 均为 High 时，它会转至 ACK\_STATE。如果在任意违规地址中发生任何读取操作，S\_AXI\_RRESP[1:0] 会指示 2'b10 以断言读取错误信号有效。
- **ACK\_STATE**：状态将转至 IDLE\_STATE。

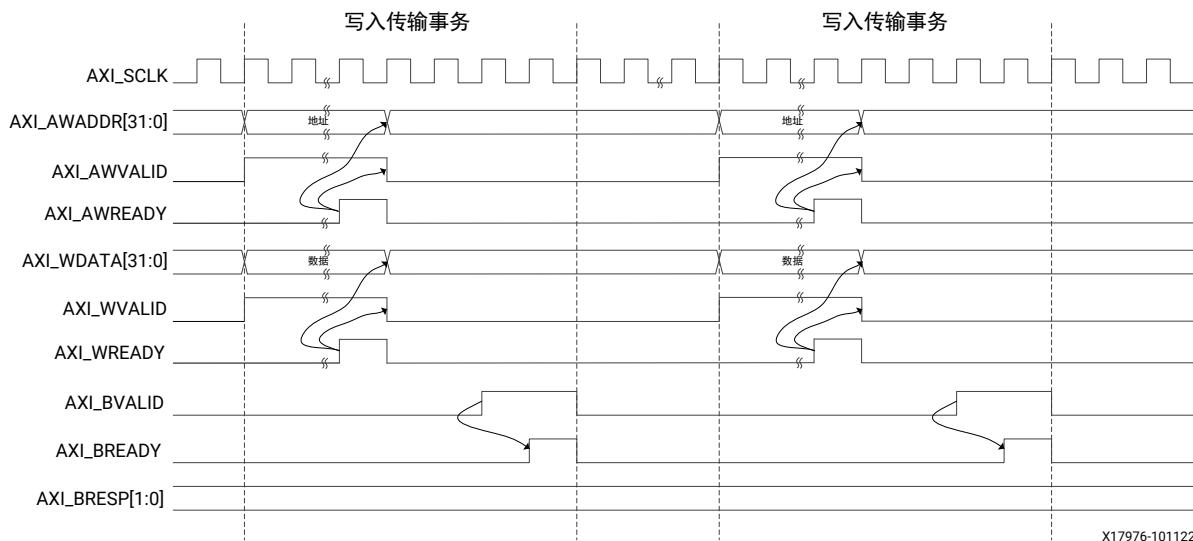
## AXI 用户接口端口

表 317：AXI 用户接口端口

名称	大小	I/O	描述
S_AXI_ACLK	1	输入	AXI 时钟信号
S_AXI_ARESETN	1	输入	AXI 低电平有效同步复位
S_AXI_PM_TICK	1	输入	PM 时钟节拍用户输入
S_AXI_AWADDR	32	输入	AXI 写入地址
S_AXI_AWVALID	1	输入	AXI 写入地址有效
S_AXI_AWREADY	1	输出	AXI 写入地址就绪
S_AXI_WDATA	32	输入	AXI 写入数据
S_AXI_WSTRB	4	输入	AXI 写入选通。此信号用于指示保存有效数据的字节通道。
S_AXI_WVALID	1	输入	AXI 写入数据有效。此信号用于指示有效的写入数据和写入选通均可供使用。
S_AXI_WREADY	1	输出	AXI 写入数据就绪
S_AXI_BRESP	2	输出	AXI 写入响应。此信号用于指示写入传输事务的状态。 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
S_AXI_BVALID	1	输出	AXI 写入响应有效。此信号用于指示该通道正在发出有效的写入响应信号。
S_AXI_BREADY	1	输入	AXI 写入响应就绪。
S_AXI_ARADDR	32	输入	AXI 读取地址
S_AXI_ARVALID	1	输入	AXI 读取地址有效
S_AXI_ARREADY	1	输出	AXI 读取地址就绪
S_AXI_RDATA	32	输出	AXI 读取数据（由从接口发出）
S_AXI_RRESP	2	输出	AXI 读取响应。此信号用于指示读取传输的状态。 'b00 = OKAY 'b01 = EXOKAY 'b10 = SLVERR 'b11 = DECERR
S_AXI_RVALID	1	输出	AXI 读取数据有效
S_AXI_RREADY	1	输入	AXI 读取就绪。此信号用于指示用户/主控制器可接受读取数据和响应信息。

### 有效的写入传输事务

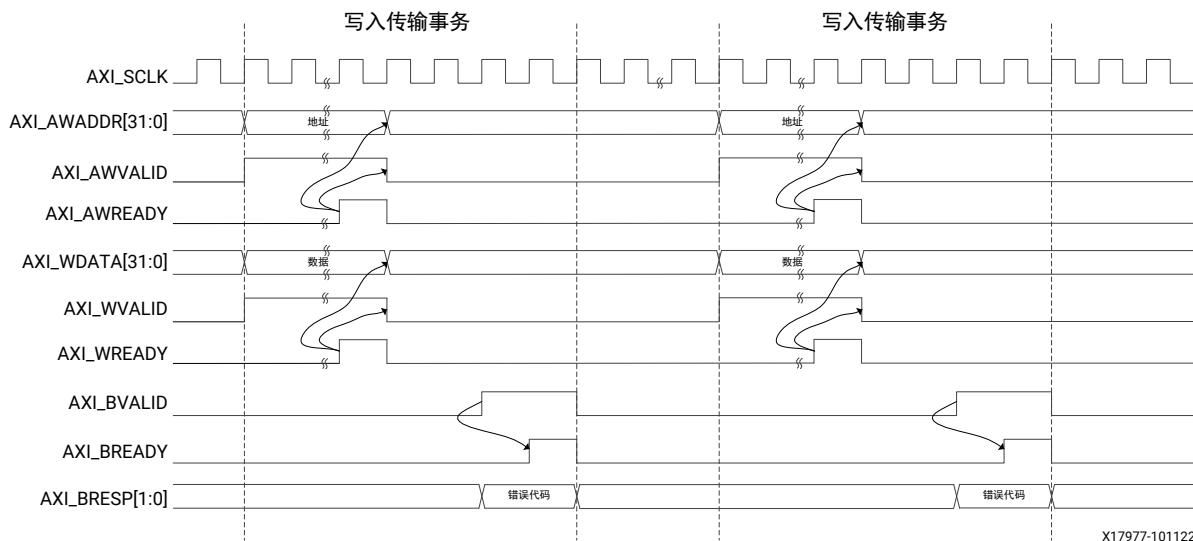
图 58：AXI4-Lite 用户侧写入传输事务



X17976-101122

### 无效的写入传输事务

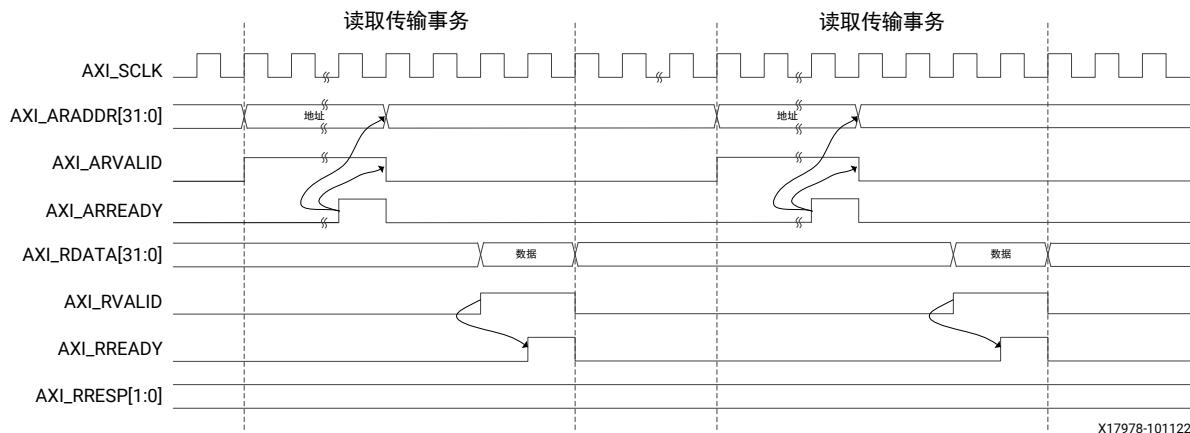
图 59：AXI4-Lite 用户侧写入传输事务（含无效的写入地址）



X17977-101122

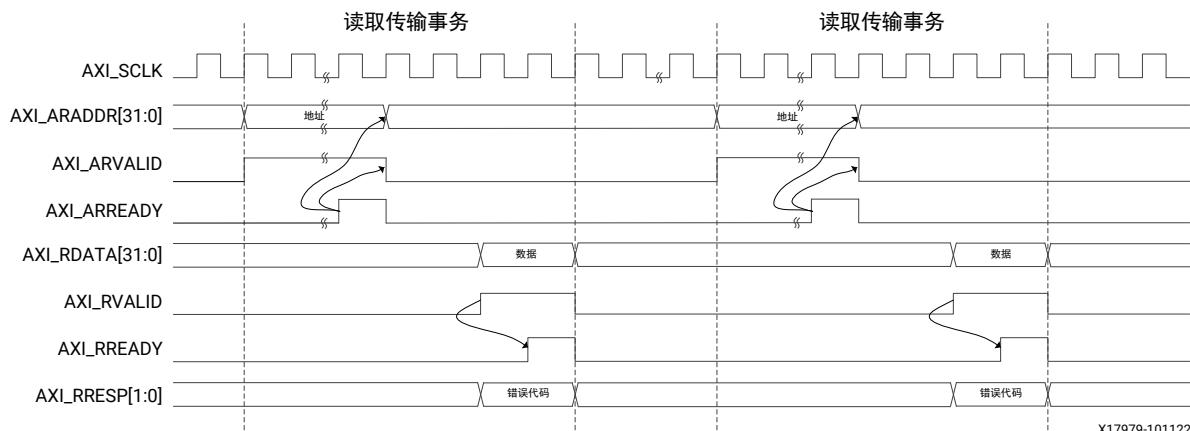
### 有效的读取传输事务

图 60：AXI4-Lite 用户侧读取传输事务



### 无效的读取传输事务

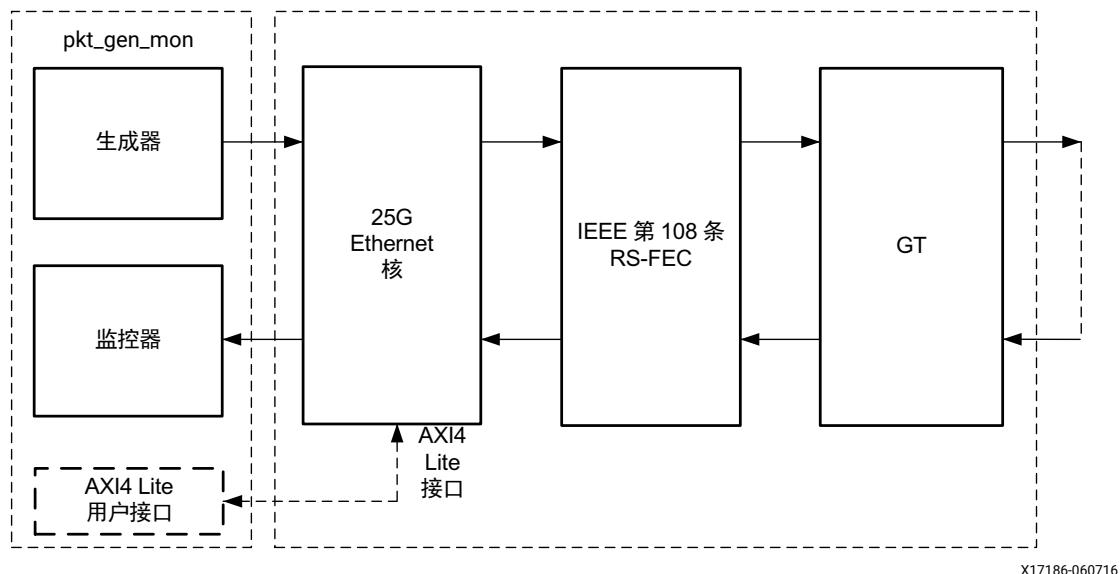
图 61：AXI4-Lite 用户侧读取传输事务（含无效的读取地址）



## IEEE 第 108 条 (RS-FEC) 集成

如果想要在 25G Ethernet IP 和 GT 之间包含 IEEE 第 108 条 RS-FEC 软核 IP（用于纠错），则必须在“Configuration”（配置）选项卡下选中“Include Clause 108 (RS-FEC)”（包含第 108 条 (RS-FEC)）复选框。该选项仅可用于 25G 线速率。

图 62：25G 和 GT 之间的 RS-FEC 集成



此功能启用 25G 核和 GT 之间例化的 IEEE 第 108 条 RS-FEC 软核 IP 组件。来自 25G 核的 TX SerDes 线将输入到 RS-FEC 软核 IP 以用于前向纠错编码。然后，将来自 RS-FEC 模块的输出馈送到 GT。同样，来自 GT 的 RX SerDes 线将馈送到 RS-FEC 模块以用于纠错解码，然后馈送到 25G 核。

请参阅《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》(PG217) (需注册) 以了解适用于 LogiCORE IP 核的 IEEE 第 108 条里德·所罗门前向纠错码 (Reed-Solomon Forward Error Correction) 及其功能。

## PTP 1588 Timer Syncer 块

### 简介

PTP 1588 Timer Syncer 块可为设计中的所有以太网端口提供参考时间。它可实现 IEEE1588 实时时钟定时器，并且可在局部端口的时钟域中精确重建 1588 定时器。

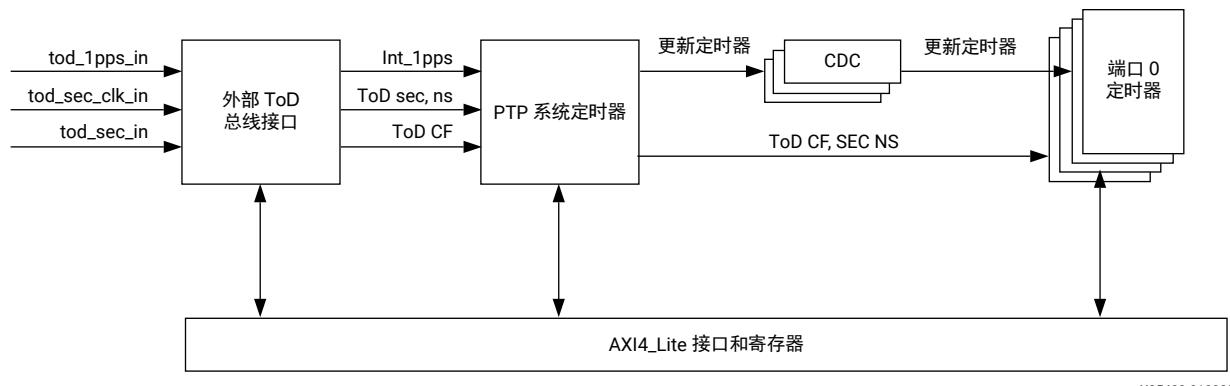
### 功能特性

- 可配置为主系统定时器、主系统和端口定时器，也可配置为端口定时器。
- 支持 ToD (秒 - 纳秒) 和纠正时间 (CF) 格式。
- 实现外部 ToD 总线接口逻辑，用于同步到高精度时钟源。
- 使系统 ToD 与端口定时器的时钟域交汇。
- AXI4-Lite I/F 用于配置和监控。

## 核概述

下图用于标识 Timer Syncer IP 的主要功能块。在此图中，为便于演示，仅显示端口定时器的 4 个实例。用户可在生成核时选择端口实例的数量。该核支持最多 16 个端口定时器实例。

图 63：Timer Syncer IP 功能模块框图



X25482-010823

Timer Syncer IP 包含实现各种 ToD 拓扑结构和应用所需的所有功能和接口。此外，该 IP 还可通过软件或硬件器件来控制。

系统定时器 (System Timer) 用于维护自由运行的时钟 ( $ts\_clk$ ) 上的时间，并提供相应的机制，以将定时器值与其它各端口定时器进行同步，其中每个端口定时器都能在其各自的独立时钟 ( $phy\_clk$ ) 上进行时钟设置。

System Timer IP 可按以下 2 种格式来提供定时器值：时间戳格式（或 ToD 格式）以及纠正字段 (Correction Field, CF) 格式。时间戳格式由 80 位组成，其中包含无符号的正秒和纳秒值字段，如 {seconds[47:0], nanoseconds[31:0]}。纠正字段 (CF) 格式则为有符号的 64b 纳秒值乘以  $2^{+16}$ 。

## 功能特性

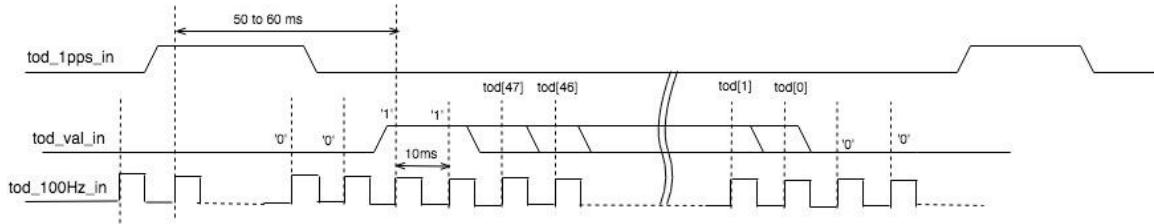
- 可配置为主系统定时器、主系统和端口定时器，也可配置为端口定时器。
- 支持 ToD（秒 - 纳秒）和纠正时间 (CF) 格式。
- 实现外部 ToD 总线接口逻辑，用于同步到高精度时钟源。
- 使系统 ToD 与端口定时器的时钟域交汇。
- AXI4-Lite I/F 用于配置和监控。

## 外部 ToD 接口子块

此模块为可选，仅当您在生成 IP 的过程中选中“Enable Ext ToD Bus I/F”（启用外部 ToD 总线接口）选项时，此模块才会出现。其功能是将参考秒值加载到 PTP 系统定时器 (PTP System Timer) 上，并将系统定时器同步到外部 1PPS 源。其目的是与高精度时钟源/时序器件进行连接。

此模块的输入为：1PPS 脉冲串行 1 位数据总线，此总线可承载参考 ToD 秒数时间（由串行时钟输入进行时钟设置）。串行时钟应小于或等于定时器同步 (Timer Sync) 块的自由运行时钟 ( $ts\_clk$ )。加载参考 48 位秒的过程如下图所示。

图 64: ToD 值（秒）接口



外部 ToD 接口块用于将串行输入 ToD 秒数值存储在用于保存的寄存器中。在 1PPS 信号的下一个接收上升沿处，该寄存器的值将递增 +1 秒，且此结果将提供给 PTP 系统定时器子块。此 1PPS 信号的上升沿将在外部 ToD 接口块内重定时至 `ts_clk` 时钟，然后将输出以供 PTP 系统定时器块使用。

## 功能特性

- 可配置为主系统定时器、主系统和端口定时器，也可配置为端口定时器。
- 支持 ToD（秒 - 纳秒）和纠正时间 (CF) 格式。
- 实现外部 ToD 总线接口逻辑，用于同步到高精度时钟源。
- 使系统 ToD 与端口定时器的时钟域交汇。
- AXI4-Lite I/F 用于配置和监控。

## PTP 端口定时器子块

这些子块包含各端口的 TX 和 RX 定时器，这些定时器通常由各自的 TX 和 RX PHY 时钟 (`tx_phy_clk` 和 `rx_phy_clk`) 来进行时钟设置。Timer Synchronizer IP 支持在生成时启用最多 16 个端口定时器。

这些端口定时器用于维护 ToD 内的时间（48 位秒和 30 位纳秒）以及连续时间/纠正字段（63 位 CF）格式，并提供输出（同步到 `tx_phy_clk` 和 `rx_phy_clk`）以供赛灵思的以太网 IP 使用。

每个端口的 TX 和 RX 时钟源都应与定时器的 `ts_clk` 主时钟保持相对异步关系。端口定时器包含跨 PTP 系统定时器的负载脉冲和定时器更新值的域同步逻辑。

由于 PTP 系统定时器已初始化，或者已同步至高精度参考时钟，因此它将转而把自己更新后的定时器值推送到与之相连的所有端口定时器。初始化后，Timer Syncer IP 即可配置为使端口定时器与 PTP 系统定时器的主 ToD 值保持持续同步，或者可独立控制端口定时器。

## 功能特性

- 可配置为主系统定时器、主系统和端口定时器，也可配置为端口定时器。
- 支持 ToD（秒 - 纳秒）和纠正时间 (CF) 格式。
- 实现外部 ToD 总线接口逻辑，用于同步到高精度时钟源。
- 使系统 ToD 与端口定时器的时钟域交汇。
- AXI4-Lite I/F 用于配置和监控。

## 端口描述

### Timer Syncer IP 接口端口

以下是 Timer Syncer IP 顶层端口。

#### 时钟与复位

表 318：时钟与复位

信号	方向	时钟域	描述
ts_clk	输入	不适用	自由运行的时钟，用于对系统定时器的计数器进行时钟设置
ts_RST	输入	ts_clk	系统定时器复位高电平有效
tod_intr	输出	ts_clk	发生 1-PPS 事件时，中断断言有效

#### 外部 ToD 总线接口

表 319：外部 ToD 总线接口

信号	方向	时钟域	描述
tod_1pps_in	输入	不适用	外部总线 1-PPS 输入
tod_sec_clk_in	输入	不适用	外部总线时钟，用于 tod_sec_in
tod_sec_in	输入	tod_sec_clk_in	外部总线秒串行数据
tod_1pps_out	输出	ts_clk	1-PPS 输出 当系统定时器的纳秒字段滚动时，此信号断言有效
tod_sec_clk_out	输入	不适用	外部总线时钟的回显
tod_sec_out	输入	tod_sec_clk_in	内部 ToD 秒的串行输出

#### 各端口的端口定时器接口

m 表示端口号 ( $0 \leq m \leq 15$ )。

表 320：各端口的端口定时器接口

信号	方向	时钟域	描述
tx_phy_clk_m	输入	不适用	端口 TX PHY 时钟
rx_phy_clk_m	输入	不适用	端口 RX PHY 时钟
tx_phy_RST_m	输入	tx_phy_clk_m	端口 TX 复位
rx_phy_RST_m	输入	rx_phy_clk_m	端口 RX 复位
tx_tod_sec_m[47:0]	输出	tx_phy_clk_m	端口 TX 定时器秒字段
tx_tod_ns_m[31:0]	输出	tx_phy_clk_m	端口 TX 定时器纳秒字段
tx_tod_corr_m[63:0]	输出	tx_phy_clk_m	端口 TX 定时器 CF 字段
rx_tod_sec_m[47:0]	输出	rx_phy_clk_m	端口 RX 定时器秒字段
rx_tod_ns_m[31:0]	输出	rx_phy_clk_m	端口 RX 定时器纳秒字段

表 320：各端口的端口定时器接口 (续)

信号	方向	时钟域	描述
rx_tod_corr_m[63:0]	输出	rx_phy_clk_m	端口 RX 定时器 CF 字段

## AXI4-Lite 接口

表 321：AXI4-Lite 接口

信号	方向	时钟域	描述
s_axi_aclk	输入	不适用	AXI4-Lite 接口时钟
s_axi_aresetn	输入	s_axi_aclk	AXI4-Lite 接口复位
s_axi_awaddr[31:0]	输入	s_axi_aclk	写入地址
s_axi_awvalid	输入	s_axi_aclk	写入地址有效
s_axi_awready	输出	s_axi_aclk	写入地址就绪
s_axi_wdata[31:0]	输入	s_axi_aclk	写入数据
s_axi_wstrb[3:0]	输入	s_axi_aclk	写入数据字节有效。绑定到 4'hF
s_axi_wvalid	输入	s_axi_aclk	写入有效
s_axi_wready	输出	s_axi_aclk	写入就绪
s_axi_bresp	输出	s_axi_aclk	写入响应
s_axi_bvalid	输出	s_axi_aclk	写入响应有效
s_axi_bready	输入	s_axi_aclk	写入响应就绪
s_axi_araddr[31:0]	输入	s_axi_aclk	读取地址
s_axi_arvalid	输入	s_axi_aclk	读取地址有效
s_axi_arready	输出	s_axi_aclk	读取地址就绪
s_axi_rdata[31:0]	输出	s_axi_aclk	读取数据
s_axi_rresp	输出	s_axi_aclk	读取响应
s_axi_rvalid	输出	s_axi_aclk	读取数据有效
s_axi_rready	输入	s_axi_aclk	读取就绪

## 内部子块端口

以下各表包含 IP 子块上存在的选定端口，用于辅助集成和调试。

## 外部 ToD 总线端口

表 322：外部 ToD 总线端口

信号	方向	时钟域	描述
ext_tod_bus_sec	输入	ts_clk	自由运行的时钟，用于对系统定时器的计数器进行时钟设置
ext_tod_bus_ns	输入	ts_clk	系统定时器复位高电平有效
ext_tod_bus_1pps	输出	ts_clk	发生 1PPS 事件时，中断断言有效

## PTP 系统定时器端口

表 323：PTP 系统定时器端口

信号	方向	时钟域	描述
sys_tod_sec[48:0]	输出	ts_clk	系统定时器 ToD 秒字段
sys_tod_ns[31:0]	输出	ts_clk	系统定时器 ToD 纳秒字段
sys_tod_corr[63:0]	输出	ts_clk	系统定时器 ToD CF 格式
update_timer	输出	ts_clk	同步更新脉冲至端口定时器块。 当主定时器由外部 ToD 接口或者 通过寄存器更新来进行同步时，此 信号断言有效。
sys_timer_1pps_out	输出	ts_clk	1-PPS 输出至外部 ToD 总线块 当系统定时器的纳秒字段滚动时， 此端口断言有效。

## 端口定时器端口

表 324：端口定时器端口

信号	方向	时钟域	描述
update_port_timer	输入	ts_clk	由系统定时器驱动的同步更新脉冲
sys_tod_sec_in[48:0]	输入	ts_clk	系统定时器秒字段 仅当定时器格式为 ToD 或双格式 时，该信号才存在
sys_tod_ns_in[31:0]	输入	ts_clk	系统定时器纳秒字段 仅当定时器格式为 ToD 或双格式 时，该信号才存在
sys_tod_corr_in[63:0]	输入	ts_clk	系统定时器 CF 字段。 仅当定时器格式为 CF 或双格式 时，该信号才存在

## PTP 系统定时器子块

这是主定时器或主 ToD 定时器，由自由运行的时钟 ( $ts\_clk$ ) 进行时钟设置。系统定时器用于维护 ToD 内的时间（48 位秒和 30 位纳秒）以及连续/纠正字段（63 位 CF）格式。

提供的寄存器用于初始化定时器秒和纳秒计数器值，或者回读值的快照。针对秒值和纳秒值提供了一组偏移寄存器，这些值被添加到 ToD 定时器值中，然后再输出到端口定时器。

初始化后，PTP 系统定时器内部 ToD 计数器可选择同步到外部器件，方法是使用外部 ToD 接口块的输出 1PPS 信号，或者使用软件控制通过寄存器操作来执行同步。

1PPS 输出用于指示何时 PTP 系统定时器的 ToD ns 字段从 999\_999\_999 ns 滚动至 1 秒。并且，还可从快照寄存器读取系统定时器的值。

将系统定时器的 ToD 计数器传输到端口定时器的过程是可配置的，可通过外部总线的 1PPS 脉冲或者通过写入 `TOD_SW_LOAD` 寄存器来触发。触发传输后，PTP 系统定时器会提供负载脉冲输出（同步到  $ts\_clk$  域），并将其内部定时器的值置于输出总线上。

## 地址映射

所有控制寄存器和状态寄存器均为存储器映射寄存器。上电或复位后，您可随时重新配置核的参数默认值。

### 定时器寄存器类型

表 325：定时器寄存器类型

访问类型	描述
RO	只读 可读取的寄存器；写入无效
RW	读取和写入 可读取且可写入的寄存器
RW1T	读取和写入。设为“1”表示触发 写入“1”即触发。写入“0”则忽略。读取返回“0”。
RW1C	读取，写入“1”表示清除 写入“1”会将寄存器中的对应位元位置设为0（清零）；写入“0”则使对应位保持不变。 例如，它可用于确认中断状态。
WO	只写 可写入的寄存器，其值不可读（返回“0”）

## 定时器寄存器映射

表 326：定时器寄存器映射

偏移	寄存器名称	访问	描述
0x0000	TOD_CONFIG	RW	<p>主配置</p> <p>[0] - 启用系统定时器 该字段为“0”会禁用系统定时器 IP</p> <p>[1] - 启用外部 ToD 总线 写入“1”即可启用 ToD 总线信号。“overwrite mode”（覆盖模式）字段可用于进一步定义信号的使用方式 仅当生成的核具有外部 ToD 总线接口支持时，此模式才存在</p> <p>[3:2] - Overwrite Mode:</p> <ul style="list-style-type: none"><li>· 0x0 - 当发生来自外部 ToD 总线的 1-PPS 事件时，不覆盖系统定时器计数器</li><li>· 0x1 - 当发生来自外部 ToD 总线的 1-PPS 事件时，以外部 ToD 总线秒数输入来覆盖系统定时器计数器</li><li>· 0x2 - 当发生来自外部 ToD 总线的 1-PPS 事件时，以软件 TOD_SW_SEC_0/1 寄存器中存储的值覆盖系统定时器计数器</li><li>· 0x3 - 保留</li></ul> <p>仅当生成的核具有外部 ToD 总线接口支持时，以上模式才存在</p> <p>[4] - 启用 Sys_timer 自动刷新。写入 1 即可启用 System Timer 块以最新 ToD 和 CTIME（如已启用）值来自动刷新端口定时器的功能</p> <p>[5] - 在外部 1PPS 上启用定时器快照</p> <p>[15:6] - 保留</p> <p>[16 到 31] - 启用端口 TX 和 RX 定时器</p> <ul style="list-style-type: none"><li>· [16] = 启用端口 0 TX 和 RX</li><li>· [17] = 启用端口 1 TX 和 RX</li><li>· [18] = 启用端口 2 TX 和 RX</li><li>· [19] = 启用端口 3 TX 和 RX</li><li>· [20] = 启用端口 4 TX 和 RX</li><li>· [21] = 启用端口 5 TX 和 RX</li><li>· [22] = 启用端口 6 TX 和 RX</li><li>· [23] = 启用端口 7 TX 和 RX</li><li>· [24] = 启用端口 8 TX 和 RX</li><li>· [25] = 启用端口 9 TX 和 RX</li><li>· [26] = 启用端口 10 TX 和 RX</li><li>· [27] = 启用端口 11 TX 和 RX</li><li>· [28] = 启用端口 12 TX 和 RX</li><li>· [29] = 启用端口 13 TX 和 RX</li><li>· [30] = 启用端口 14 TX 和 RX</li><li>· [31] = 启用端口 15 TX 和 RX</li></ul> <p>端口数上限取决于核生成时启用的端口数。</p>
0x0004	TOD_SNAPSHOT	RW1T	<p>[0] - 对所有定时器生成快照 写入 1'b1 将对所有计数器（系统、外部 ToD 总线和所有已启用的端口）生成快照</p> <p>[31:1] - 保留</p>

表 326：定时器寄存器映射 (续)

偏移	寄存器名称	访问	描述
0x0008	TOD_INTR_ENABLE	RW	中断使能寄存器 [0] - 1-PPS 中断 (主 RTC 秒字段) [15:1] - 保留 [16] - 1-PPS 中断 (外部 1pps 输入) [31:17] - 保留
0x000C	TOD_INTR_STATUS	RW1C	中断清除寄存器 [0] - 1-PPS 中断 (主 RTC 秒字段) [15:1] - 保留 [16] - 1-PPS 中断 (外部 1pps 输入) [31:17] - 保留
0x0010	TOD_SW_SEC_0	RW	[31:0] - 覆盖主定时器的秒字段位 [31:0]
0x0014	TOD_SW_SEC_1	RW	[15:0] - 覆盖主定时器的秒字段位 [47:32] [31:16] - 保留
0x0018	TOD_SW_NS	RW	[29:0] - 覆盖主定时器的秒字段位 [31:30] - 保留
0x001C	TOD_SW_LOAD	RW1T	[0] - 写入 “1” 即可初始化大量系统定时器的 ToD 值 (来自 TOD_SW_SEC_0/1、TOD_SW_NS 和 TOD_SW_CTIME_0/1 寄存器) [1] - 写入 “1” 即可初始化大量系统定时器的 ToD 偏移值 (来自 TOD_SEC_SYS_OFFSET_0/1 和 TOD_NS_SYS_OFFSET_0 寄存器)  <b>注释：</b> 偏移是由逻辑在系统定时器输出到端口定时器前添加的。因此，系统定时器 ToD 回读不反映偏移。 [31:2] - 保留
0x0020	TOD_SW_CTIME_0	RW	[31:0] - 覆盖主定时器的 CF 字段位 [31:0]
0x0024	TOD_SW_CTIME_1	RW	[30:0] - 覆盖主定时器的秒字段位 [63:32] [31] - 保留
0x0028	TOD_SEC_SYS_OFFSET_0	RW	{TOD_SEC_SYS_OFFSET_1[15:0], TOD_SEC_SYS_OFFSET_0[31:0]} - 表示系统定时器 48b 秒字段有符号偏移值
0x002C	TOD_SEC_SYS_OFFSET_1	RW	TOD_NS_SYS_OFFSET_0[29:0] - 表示系统定时器 30b 纳秒字段有符号偏移值
0x0030	TOD_NS_SYS_OFFSET_0	RW	针对 TOD_SEC_SYS_OFFSET，有符号位解释如下 如果 [47] = 1'b1，则从系统定时器减去 如果 [47] = 1'b0，则与系统定时器相加 需在寄存器 0x001C 上应用触发器位 [1]
0x0100	TOD_SYS_SEC_0	RO	[31:0] - 生成系统定时器的秒字段 [31:0] 的快照
0x0104	TOD_SYS_SEC_1	RO	[15:0] - 生成系统定时器的秒字段 [47:32] 的快照 [31:16] - 保留
0x0108	TOD_SYS_NS	RO	[29:0] - 生成系统定时器的纳秒字段 [29:0] 的快照 [31:30] - 保留
0x010C	TOD_SYS_OFFSET	RW	[31:0] - 有符号偏移，将在外部 ToD 总线上发生 1PPS 事件时，应用于从外部 ToD 总线加载的秒值，并添加到 0 纳秒字段。 有符号值以 $2^{16}$ ns 来表示 仅当生成的核具有外部 ToD 总线支持时，此模式才存在
0x0110	TOD_SYS_CTIME_0	RO	[31:0] - 生成系统定时器的 CF 字段 [31:0] 的快照
0x0114	TOD_SYS_CTIME_1	RO	[30:0] - 生成系统定时器的 CF 字段 [63:32] 的快照 [31] - 保留

表 326：定时器寄存器映射 (续)

偏移	寄存器名称	访问	描述
0x0120	TODBUS_SEC_0	RO	外部 ToD 总线的秒字段 [31:0] 的当前值
0x0124	TODBUS_SEC_1	RO	[15:0] - 外部 ToD 总线的秒字段 [47:32] 的当前值 [31:16] - 保留
0x012C	TODBUS_SYS_DIFF	RO	[31:0] - 对外部 ToD 总线采集的值与系统定时器的内部 ToD 值（以有符号单位 $2^{-16}$ ns 来表示）进行每秒一次比较
0x0034	TOD_SYS_PERIOD_0	RW	系统定时器 TS 时钟周期，以 $2^{-48}$ ns 来表示。 例如，3.2 ns 表示为： TOD_SYS_PERIOD_0[31:0] = 0x3333_3333
0x0038	TOD_SYS_PERIOD_1	RW	TOD_SYS_PERIOD_1[23:0] = 0x0003_3333 TOD_SYS_PERIOD_1[31:24] reserved 写入 TOD_SYS_PERIOD_1 寄存器将把更新后的周期值“落实”到系统定时器内。

## 端口定时器寄存器

端口 1 到 15 所罗列的偏移处都存在对应的端口定时器（0 到 15），其中每个端口定时器都采用重复的端口定时器寄存器设置。

表 327：端口定时器寄存器

偏移	寄存器名称	访问	描述
端口 0 寄存器：0x0200 到 0x023F			
0x0200	TX0_CTIME_0	RW	[31:0] - 生成端口 0 TX 定时器的 CF 字段 [31:0] 的快照
0x0204	TX0_CTIME_1	RW	[30:0] - 生成端口 0 TX 定时器的 CF 字段 [63:32] 的快照 [31] - 保留
0x0208	TX0_PERIOD_0	RW	端口 0 TX 时钟周期，以 $2^{-48}$ ns 来表示 例如，3.2 ns 表示为
0x020C	TX0_PERIOD_1	RW	TX0_PERIOD_0[31:0] = 0x3333_3333 TX0_PERIOD_1[23:0] = 0x0003_3333 TX0_PERIOD_1[31:24] reserved
0x0210	TX0_SYS_OFFSET	RW	[31:0] - 应用于端口 0 TX 定时器的 ToD 输出的有符号偏移，以 $2^{-16}$ ns 来表示。
0x0214	TX0_NS_SNAP	RO	[29:0] - 生成端口 0 TX 定时器的纳秒字段 [29:0] 的快照 [31:30] - 保留
0x0218	TX0_SEC_0_SNAP	RO	[31:0] - 生成端口 0 TX 定时器的秒字段 [31:0] 的快照
0x021C	TX0_SEC_1_SNAP	RO	[15:0] - 生成端口 0 TX 定时器的秒字段 [47:32] 的快照 [31:16] - 保留
0x0220	RX0_CTIME_0	RW	[31:0] - 生成端口 0 RX 定时器的 CF 字段 [31:0] 的快照
0x0224	RX0_CTIME_1	RW	[30:0] - 生成端口 0 RX 定时器的 CF 字段 [63:32] 的快照 [31] - 保留

表 327：端口定时器寄存器（续）

偏移	寄存器名称	访问	描述
0x0228	RX0_PERIOD_0	RW	端口 0 RX 时钟周期，以 $2^{-48}$ ns 来表示 例如，3.2 ns 表示为 $RX0\_PERIOD\_0[31:0] = 0x3333_3333$
0x022C	RX0_PERIOD_1	RW	$RX0\_PERIOD\_1[23:0] = 0x0003_3333$ $RX0\_PERIOD\_1[31:24]$ reserved
0x0230	RX0_SYS_OFFSET	RW	[31:0] - 应用于端口 0 RX 定时器的 ToD 输出的有符号偏移，以 $2^{-16}$ ns 来表示。
0x0234	RX0_NS_SNAP	RO	[29:0] - 生成端口 0 RX 定时器的纳秒字段 [29:0] 的快照 [31:30] - 保留
0x0238	RX0_SEC_0_SNAP	RO	[31:0] - 生成端口 0 RX 定时器的秒字段 [31:0] 的快照
0x023C	RX0_SEC_1_SNAP	RO	[15:0] - 生成端口 0 RX 定时器的秒字段 [47:32] 的快照 [31:16] - 保留
0x0240	CORE_TX0_PERIOD_0	RO	核中配置的 Port0 TX 时钟周期，以 $2^{-48}$ ns 来表示。
0x0244	CORE_TX0_PERIOD_1	RO	例如，3.2 ns 表示为： $CORE\_TX0\_PERIOD\_0 = 0x3333_3333$ $CORE\_TX0\_PERIOD\_1 = 0x0003_3333$ $CORE\_TX0\_PERIOD\_1[31:24]$ reserved
0x0248	CORE_RX0_PERIOD_0	RO	核中配置的 Port0 RX 时钟周期，以 $2^{-48}$ ns 来表示。
0x024C	CORE_RX0_PERIOD_1	RO	例如，3.2 ns 表示为： $CORE\_RX0\_PERIOD\_0 = 0x3333_3333$ $CORE\_RX0\_PERIOD\_1 = 0x0003_3333$ $CORE\_RX0\_PERIOD\_1[31:24]$ reserved
0x0250	PORT0_SEC_SYS_OFFSET_0		PORT0_SEC_SYS_OFFSET_1[15:0], PORT0_SEC_SYS_OFFSET_0[31:0]} - 表示端口定时器 48b 秒字段有符号偏移值。
0x0254	PORT0_SEC_OFFSET_1		PORT0_NS_SYS_OFFSET_0[29:0] - 表示端口定时器 30b 纳秒字段有符号偏移值。
0x0258	PORT0_NS_OFFSET_0		针对 PORT0_SEC_SYS_OFFSET，有符号位解释如下： 如果 [47] = 1'b1，则从端口定时器减去 如果 [47] = 1'b0，则与端口定时器相加 这些寄存器用于应用大型端口特定的偏移。 要触发应用此端口的偏移，需写入 PORT0_NS_SYS_OFFSET_0。
0x025C - 0x027F	Reserved	不适用	保留
端口 1 寄存器：0x0280 到 0x02FF			
端口 2 寄存器：0x0300 到 0x037F			
端口 3 寄存器：0x0380 到 0x03FF			
端口 4 寄存器：0x0400 到 0x047F			
端口 5 寄存器：0x0480 到 0x04FF			
端口 6 寄存器：0x0500 到 0x057F			
端口 7 寄存器：0x0580 到 0x05FF			

表 327：端口定时器寄存器 (续)

偏移	寄存器名称	访问	描述
端口 8 寄存器：0x0600 到 0x067F			
端口 9 寄存器：0x0680 到 0x06FF			
端口 10 寄存器：0x0700 到 0x077F			
端口 11 寄存器：0x0780 到 0x07FF			
端口 12 寄存器：0x0800 到 0x087F			
端口 13 寄存器：0x0880 到 0x08FF			
端口 14 寄存器：0x0900 到 0x097F			
端口 15 寄存器：0x0980 到 0x09FF			

## 软件驱动程序初始化

建议软件驱动程序遵循下列步骤按顺序正确执行核初始化。

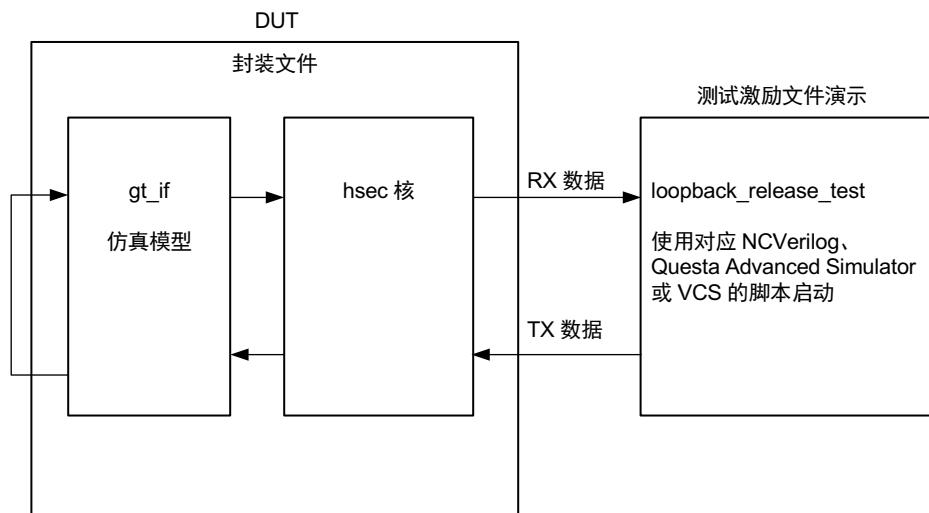
1. 初始化设计的其它元素以使包括 `ts_clk` 和 `tx/rx_phy_clk*` 在内的时钟存在并保持稳定。为这些时钟域释放复位，例如，`ts_rst` 或 `tx/rx_phy_rst*`。
2. 配置 `TOD_CONFIG` 寄存器：
  - a. 在“Timer”（定时器）模式或“Timer Syncer”（定时器同步器）模式下，设置位 [0] 以启用系统定时器。
  - b. 如果要使用外部 ToD 总线（1PPS 同步和可选串行秒输入），则请将位 [1] 设置为 1。
  - c. 请将模式位字段 [3:2] 设置为与系统中使用的目标同步方法相匹配的设置。例如，如果系统使用连接到外部 ToD 总线（1PPS 输入和秒值串行输入）的外部器件，那么模式字段应设置为 0x1。
  - d. 最后，启用“Port Timer”（端口定时器）使能位字段 [19:4] 的相应的位（设置为 1）即可启用端口 TX 定时器和 RX 定时器。
3. 如果系统定时器的初始值将由软件通过寄存器编程来设置（外部 ToD 总线接口的替代方法），那么请将相应的初始值写入软件加载寄存器，例如，`TOD_SW_SEC_0/1` 和 `TOD_SW_NS`, `TOD_SW_CTIME_0/1`。并通过写入 `TOD_SEC_SYS_OFFSET_0/1` 和 `TOD_NS_SYS_OFFSET_0` 寄存器来配置要加载的任何静态偏移。
4. 如果在第 2 步中更新了软件寄存器，那么向 `TOD_SW_LOAD[1:0]` 的相应位写入 1 即可触发系统定时器以加载软件值。
5. 对端口 TX/RX 定时器进行编程。这包括对周期值以及相应值的所有必要的静态偏移进行编程，方法是写入端口定时器的寄存器：`TX<M>_PERIOD_0/1`、`RX<M>_PERIOD_0/1` 和 `TX<M>/RX<M>_SYS_OFFSET`。
6. 将输入 `CORE_CONFIGURATION` 配置为“1”并提供系统定时器时钟 `RESYNC_CLK_PERIOD`（以 ps 为单位）来启用混合同步器模块。

针对设计中存在的所有端口 (M) 继续执行步骤 4。

# 批处理模式测试激励文件

10G/25G Ethernet Subsystem 的每个批处理模式版本都包含测试激励文件演示，用于在完整的子系统上执行环回测试。为便于您使用，其中提供了相应脚本，可用于从多种业界标准的仿真器启动测试激励文件。测试程序会测试数据路径以检查能否接收到发射的帧。用于子系统的寄存器传输级 (RTL) 仿真模型也包含在内。您必须根据所使用的 Vivado® Design Suite 版本中最新的仿真环境设置来为收发器仿真模型提供正确的路径。

图 65：测试激励文件



X15170-101122

# 升级

本附录包含有关升级至 Vivado® Design Suite 中更新版本的 IP 核的信息。

其中提供的脚本可用于将设计升级到更新版本的 Vivado Design Suite。此脚本可在 `/compile/xilinx/upgrade_IP` 目录下找到。请运行以下脚本以执行以下升级：

- 用于将收发器封装升级到最新版本。
- 用于使用最新收发器仿真模型，如不升级此模型，可能导致仿真错误。



**提示：**在运行此脚本前，请为原始设计保存副本，以便在您需要还原到原版本时使用。

## 从 v2.3 到 v2.4 的更改

### 新增端口

- `tx_parityin[7:0]`
- `rx_parityout[7:0]`
- `ctl_tx_parity_err_response`
- `stat_tx_bad_parity`

### 新增寄存器

#### 配置寄存器和状态寄存器

- `CONFIGURATION_TSN_REG`
- `STAT_TSN_REG`

#### 统计数据计数器

- `STAT_TX_MM_STATUS_LSB: 0x0980`
- `STAT_TX_MM_STATUS_MSB: 0x0984`
- `STAT_TX_MM_STATUS_LSB: 0x0988`
- `STAT_TX_MM_FRAGMENT_MSB: 0x098C`
- `STAT_TX_MM_HOLD_LSB: 0x0990`
- `STAT_TX_MM_HOLD_MSB: 0x0994`

- STAT\_RX\_MM\_ASSEMBLY\_ERROR\_LSB: 0x0998
- STAT\_RX\_MM\_ASSEMBLY\_ERROR\_MSB: 0x099C
- STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_LSB: 0x09A0
- STAT\_RX\_MM\_FRAME\_SMD\_ERROR\_MSB: 0x09A4
- STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_LSB: 0x09A8
- STAT\_RX\_MM\_FRAME\_ASSEMBLY\_OK\_MSB: 0x09AC
- STAT\_RX\_MM\_FRAGMENT\_LSB: 0x09B0
- STAT\_RX\_MM\_FRAGMENT\_MSB: 0x09B4

## 新增端口

### 控制端口和状态端口

- ctl\_en\_preempt
- ctl\_hold\_request
- ctl\_disable\_verify
- ctl\_restart\_verify
- ctl\_addfrag\_size
- ctl\_verify\_time
- ctl\_verify\_limi
- stat\_tx\_mm\_verify
- stat\_tx\_mm\_status
- stat\_tx\_mm\_fragment
- stat\_tx\_mm\_hold
- stat\_rx\_mm\_assembly\_error
- stat\_rx\_mm\_frame\_smd\_error
- stat\_rx\_mm\_frame\_assembly\_ok
- stat\_rx\_mm\_fragment

---

## 从 v2.3 (2017 年 10 月 4 日) 到 v2.3 (2017 年 12 月 20 日) 的更改

## 新增端口

- ctl\_tx\_latency\_\*
- ctl\_tx\_lat\_adj\_enb\_\*

- ctl\_rx\_latency\_0
- ctl\_rx\_lat\_adj\_enb\_0
- ctl\_rx\_timestamp\_adj\_enb\_0
- ctl\_rx\_timestamp\_adj\_enb\_0

## 端口更改

- 更新 rx\_clk\_out、rx\_core\_clk 和 tx\_clk\_out 的描述。
- 更新 send\_continuous\_pkts\_\* 的描述
- 针对 ctl\_tx\_test\_pattern\_\* 将 3.42.3 更改为 3.42.7。

## 移除端口

- rx\_ptp\_pcslane\_out
- rx\_lane\_aligner\_fill\_0
- rx\_lane\_aligner\_fill\_1
- rx\_lane\_aligner\_fill\_2
- rx\_lane\_aligner\_fill\_3
- tx\_ptp\_rxtstamp\_in\_\*

---

## 从 v2.2 到 v2.3 的更改

### 新增端口

- rx\_core\_clk
- tx\_ptp\_upd\_chksum\_in
- tx\_ptp\_pcslane\_out
- tx\_ptp\_chksum\_offset\_in
- rx\_ptp\_pcslane\_out
- rx\_lane\_aligner\_fill\_0
- rx\_lane\_aligner\_fill\_1
- rx\_lane\_aligner\_fill\_2
- rx\_lane\_aligner\_fill\_3
- send\_continuous\_pkts\_\*
- user\_reg0\_\*

## 更新端口

- stat\_rx\_status
- user\_reg1\_\*

## 移除端口

- tx\_ptp\_rxtstamp\_in

## 端口名称更改

- mode\_change\_0 改为 mode\_change\_\*
- core\_speed\_0 改为 core\_speed\_\*

## 寄存器

- 已删除 USER\_REG1。
- 已将 CORE\_SPEED\_REG 名称更改为 SWITCH\_CORE\_SPEED\_REG。
- 在第 2 章的“统计数据计数器”部分中添加有关可读 STAT\_\*\_MSB/LSB 寄存器的注释。
- 添加 STAT\_CORE\_SPEED\_REG。另添加适用于新寄存器的 STAT\_CORE\_SPEED\_REG: 0498。
- 已将 USER\_REG\_0: 0184 更新为 USER\_REG\_0: 0134。
- 添加下列寄存器：CONFIGURATION\_1588\_REG、TX\_CONFIGURATION\_1588\_REG 和 RX\_CONFIGURATION\_1588\_REG

---

# 从 v2.1 到 v2.2 的更改

## 端口

新增 IEEE 802.3 第 74 条 FEC 接口和 IEEE 802.3 第 108 条 RS-FEC 接口章节，其中包含新的表格，用于提供端口描述。

### 新增端口

新增 gtpowergood\_out\_\*

### 更新端口描述

- dclk
- ctl\_gt\_reset\_all
- tick\_reg
- rx\_clk
- gt\_ref\_clk\_p、gt\_ref\_clk\_n 和 dclk

- an\_clk\_\*、stat\_fec\_lock\_error\_\* 和 ctl\_rsfec\_enable\_\*
- 

## 从 v2.0 到 v2.1 的更改

### 适用于 10 Gb/s 运行速率的新变体

- 64 位仅限 MAC 配置
- 32 位低时延 MAC 和 PCS 配置
- 重新设计了自动协商和链路训练功能特性，降低了利用率；所有更改都对用户透明呈现

### 针对 25 Gb/s 运行的功能特性更新

重新设计了自动协商和链路训练功能特性，降低了利用率；所有更改都对用户透明呈现。

## 端口

### 新增端口

- stat\_rx\_status
- axi\_ctl\_core\_mode\_switch\_\*
- rx\_mii\_clk
- tx\_mii\_clk
- clk

### 删除端口

tx\_ptp\_pcslane\_out

### 端口更改

- 添加 32 位 tx\_axis\_tdata\_\* 和 rx\_axis\_tdata\_\*
- 更新诸多信号的总线大小
- 更新 tx\_axis\_tdata\_\* 和 rx\_axis\_tdata\_\* 的总线大小。
- 添加 32/64 位 tx\_clk\_out\_\* 和 tx\_mii\_out\_\*。
- 添加 32 位 rx\_serdes\_data\_out\_\* 和 tx\_serdes\_data\_in\_\*。
- 在 [核 xci 顶层端口列表](#) 的大部分信号中，将“以太网 MAC+PCS/PMA”替换为“以太网 MAC+PCS/PMA 32/64 位”。
- 在 [核 xci 顶层端口列表](#) 的大部分信号中插入“或以太网 MAC”。
- 更新下列信号的描述信息：
  - o rx\_axis\_tdata[63 or 31:0]

- rx\_axis\_tkeep[7 or 3:0]
- ctl\_rx\_ignore\_fcs
- ctl\_rx\_max\_packet\_len[14:0]
- ctl\_rx\_min\_packet\_len[7:0]
- stat\_rx\_undersize
- stat\_rx\_fragment
- stat\_tx\_total\_bytes[3:0]
- stat\_tx\_frame\_error
- ctl\_tx\_pause\_quanta[8:0][15:0]
- ctl\_tx\_pause\_refresh\_timer[8:0][15:0]
- ctl\_an\_nonce\_seed[7:0]
- rx\_mii\_clk
- tx\_mii\_d\_\*
- tx\_mii\_c\_\*
- rx\_mii\_d\_\*
- rx\_mii\_c\_\*

### 新增寄存器

- USER\_REG\_1: 0188
- CORE\_SPEED\_REG: 018C

### 寄存器更改

- 为 CORE\_SPEED\_REG: 0180 新增位 1。
- 为 STAT\_TX\_STATUS\_REG1: 0400 的位 1 和 2 新增注释。
- 修改 配置寄存器映射 10G/25G Ethernet Subsystem、10G/25G Ethernet Subsystem 的状态寄存器映射 和 统计数据计数器 中的大部分寄存器的注释。

---

## 从 v2.0 (2016 年 10 月 5 日) 到 v2.0 (2016 年 11 月 30 日) 的更改

### 新增端口

- gtwiz\_reset\_qpll1lock\_in
- gtwiz\_reset\_qpll1reset\_out
- gt\_refclk\_out

- tx\_unfout\_\*
- stat\_rx\_valid\_ctrl\_code\_\*
- ctl\_tx\_ptp\_1step\_enable\_\*
- ctl\_tx\_ptp\_latency\_adjust\_\*
- ctl\_ptp\_transpclk\_mode\_\*
- tx\_ptp\_upd\_chksum\_in\_\*
- tx\_ptp\_tstamp\_offset\_in\_\*
- tx\_ptp\_chksum\_offset\_in\_\*
- tx\_ptp\_rxtstamp\_in\_\*
- ctl\_an\_fec\_25g\_baser\_request\_\*
- stat\_an\_lp\_ability\_25gbase\_crl\_\*
- stat\_an\_rxcdrhold\_\*
- ctl\_fec\_enable\_error\_to\_pcs\_\*
- ctl\_rx\_rsfec\_enable\_correction\_\*
- ctl\_rx\_rsfec\_enable\_indication\_\*
- ctl\_rsfec\_enable\_\*
- ctl\_rsfec\_ieee\_error\_indication\_mode\_\*
- ctl\_rsfec\_consortium\_25g\_\*
- stat\_rx\_rsfec\_hi\_ser\_\*
- stat\_rx\_rsfec\_lane\_alignment\_status\_\*
- stat\_rx\_rsfec\_corrected\_cw\_inc\_\*
- stat\_rx\_rsfec\_uncorrected\_cw\_inc\_\*
- stat\_rx\_rsfec\_err\_count0\_inc\_\*
- stat\_tx\_rsfec\_lane\_alignment\_status\_\*
- gt\_drp\_done\_\*
- speed\_\*
- anlt\_done\_\*

### 移除端口

- stat\_rx\_framing\_err\_valid\_1\_\*
- stat\_rx\_framing\_err\_1\_\*
- stat\_rx\_framing\_err\_valid\_2\_\*
- stat\_rx\_framing\_err\_2\_\*
- stat\_rx\_framing\_err\_valid\_3\_\*
- stat\_rx\_framing\_err\_3\_\*
- stat\_rx\_vl\_demuxed\_\*

- stat\_rx\_vl\_number\_0\_\*
- stat\_rx\_vl\_number\_1\_\*
- stat\_rx\_vl\_number\_2\_\*
- stat\_rx\_vl\_number\_3\_\*
- stat\_rx\_synced\_\*
- stat\_rx\_synced\_err\_\*
- stat\_rx\_mf\_len\_err\_\*
- stat\_rx\_mf\_repeat\_err\_\*
- stat\_rx\_mf\_err\_\*
- stat\_rx\_misaligned\_\*
- stat\_rx\_aligned\_err\_\*
- stat\_rx\_bip\_err\_0\_\*
- stat\_rx\_bip\_err\_1\_\*
- stat\_rx\_bip\_err\_2\_\*
- stat\_rx\_bip\_err\_3\_\*
- stat\_rx\_aligned\_\*
- stat\_rx\_status\_\*
- tx\_ptp\_pcslane\_out\_\*
- rx\_lane\_aligner\_fill\_0\_\*
- rx\_lane\_aligner\_fill\_1\_\*
- rx\_lane\_aligner\_fill\_2\_\*
- rx\_lane\_aligner\_fill\_3\_\*
- ctl\_lt\_pseudo\_seed1\_\*
- ctl\_lt\_k\_p1\_to\_tx1\_\*
- ctl\_lt\_k0\_to\_tx1\_\*
- ctl\_lt\_stat\_p1\_to\_tx1\_\*
- ctl\_lt\_stat0\_to\_tx1\_\*
- ctl\_lt\_stat\_m1\_to\_tx1\_\*
- ctl\_lt\_pseudo\_seed2\_\*
- ctl\_lt\_k\_p1\_to\_tx2\_\*
- ctl\_lt\_k0\_to\_tx2\_\*
- ctl\_lt\_k\_m1\_to\_tx2\_\*
- ctl\_lt\_k\_m1\_to\_tx2\_\*
- ctl\_lt\_stat0\_to\_tx2\_\*
- ctl\_lt\_stat\_m1\_to\_tx2\_\*
- ctl\_lt\_pseudo\_seed3\_\*

- ctl\_lt\_k\_p1\_to\_tx3\_\*
- ctl\_lt\_k0\_to\_tx3\_\*
- ctl\_lt\_k\_m1\_to\_tx3\_\*
- ctl\_lt\_stat\_p1\_to\_tx3\_\*
- Ctl\_lt\_stat0\_to\_tx3\_\*
- ctl\_lt\_stat\_m1\_to\_tx3\_\*
- stat\_an\_start\_tx\_disable\_\*
- stat\_an\_start\_an\_good\_check\_\*
- stat\_lt\_k\_p1\_from\_rx1\_\*
- stat\_lt\_k0\_from\_rx1\_\*
- stat\_lt\_k\_m1\_from\_rx1\_\*
- stat\_lt\_stat\_p1\_from\_rx1\_\*
- stat\_lt\_stat0\_from\_rx1\_\*
- stat\_lt\_stat\_m1\_from\_rx1\_\*
- stat\_lt\_k\_p1\_from\_rx2\_\*
- Stat\_lt\_k0\_from\_rx2\_\*
- stat\_lt\_k\_m1\_from\_rx2\_\*
- stat\_lt\_stat\_p1\_from\_rx2\_\*
- stat\_lt\_stat0\_from\_rx2\_\*
- stat\_lt\_stat\_m1\_from\_rx2\_\*
- stat\_lt\_k\_p1\_from\_rx3\_\*
- stat\_lt\_k0\_from\_rx3\_\*
- stat\_lt\_k\_m1\_from\_rx3\_\*
- stat\_lt\_stat\_p1\_from\_rx3\_\*
- stat\_lt\_stat0\_from\_rx3\_\*
- stat\_lt\_stat\_m1\_from\_rx3\_\*

---

## 从传统 XGEMAC 进行移植

本节包含有关从旧版本的赛灵思 10G EMAC IP 升级到全新 10G/25G High Speed Ethernet Subsystem IP 核的信息。



**建议：**对于所有新设计，赛灵思建议您使用最新版本的 10G/25G High Speed Ethernet Subsystem IP 核。

赛灵思 10G Ethernet MAC 和 10G/25G High Speed Ethernet Subsystem 核都是根据以太网 IEEE 802.3 标准来设计的。新旧版本间的部分功能特性的设计和/或处理方式存在显著差异。信号和参数名称以及对应的 AXI 寄存器同样存在差异。请注意，本节仅概括了旧 10G EMAC IP 中的功能特性，并将其与新的 10G/25G High Speed Ethernet Subsystem IP 核进行比较。要获取新功能特性或者新 IP 不包含的功能特性的列表，请参阅 [第 3 章：产品规格](#)。

在新的 10G/25G High Speed Ethernet Subsystem IP 中，下列功能特性有所改变。

- 填充：10G/25G High Speed Ethernet Subsystem IP 未添加“Pad”字段。您必须提供满足 IP 核的最短长度要求的数据包。如果将 IP 核配置为计算 FCS 并将其添加到数据包 (`ctl_tx_fcs_ins_enable = 1`) 中，那么最短数据包长度为 60 字节。如果在 IP 核外部计算并添加 FCS (`ctl_tx_fcs_ins_enable = 0`)，那么最短数据包长度为 64 字节。
- IFG 扩展：可使用 `ctl_tx_ipg_value[2:0]` 参数将帧间间隔 (IFG/IPG) 扩展至 12B。
- 赤字空闲计数 (DIC)：10G/25G High Speed Ethernet IP 始终启用 DIC。
- 管理数据输入/输出 (MDIO) 主控制器：10G/25G High Speed Ethernet Subsystem IP 不提供 MDIO 主控。在状态信号内提供相应的 MDIO 寄存器的内容。
- 故障处理：必须为 TX 故障设计不同的用户逻辑。默认情况下，如果接收到 LF/RF，旧 10G EMAC TX 会发射 RF 或者将数据包设为空闲并丢弃。此处提供了禁用故障发射的选项。10G/25G High Speed Ethernet Subsystem IP 要求您控制是否发射 LF/RF。您还必须提供故障状态信号。
- VLAN：新 10G/25G Ethernet Subsystem IP 不提供 VLAN 专用功能特性。但您可通过设置相应的 `ctl_rx_max_packet_len` 属性来支持标准 VLAN 帧 (1522 B)，并设计用户逻辑来处理任意数量的堆叠 VLAN 标签。
- 启用链路训练但不启用自动协商：在旧 10G IP 上，原先始终启用链路训练（请参阅下图），但 10G/25G High Speed Ethernet Subsystem IP 在执行自动协商后才执行链路训练，因此两者都必须启用。
- 链路训练转换：旧 10-BaseKR 子系统原先包含支持通过远端器件对其进行训练的逻辑（无需用户交互）。在 10G/25G High Speed Ethernet Subsystem IP 中，此功能不可用，必须设计相应的用户逻辑以支持此功能。
- 独立 MAC，含 64 位内部 XGMII 接口用于连接到 XAUI/RXAUI：在 10G/25G High Speed Ethernet Subsystem v2.1 和更高版本中，现已提供含 64 位内部 XGMII 接口的独立 MAC。
- 连接到外部 PHY 的外部 XGMII DDR 接口：当前在 10G/25G High Speed Ethernet IP v2.1 和更高版本中，提供了连接到外部 PHY 的外部 XGMII DDR 接口选项作为 64 位独立 MAC 的一部分。
- 暂停接口：接收到全局暂停帧时，10G/25G High Speed Ethernet Subsystem IP 不会暂停 TX 数据包传输，这交由用户逻辑来判断。

## 端口/信号比较

信号与参数的名称和功能之间存在显著的差异。由于两者间难以一对一进行比较，下表基于功能及其在 2 个 IP 核中的表现形式来描述这两者的差异。

### 发射器配置

传统 XGEMAC 针对所有 TX 配置使用的是 `mac_tx_configuration_vector[368:0]`，而 10G/25G High Speed Ethernet IP 则通过部署多种信号来实现相同目的。两者比较结果如下表所示。

表 328：发射器配置比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
传统暂停刷新值	<code>mac_tx_configuration_vector[367:352]</code>	
TX 优先级 7 暂停量刷新值	<code>mac_tx_configuration_vector[351:336]</code>	<code>ctl_tx_pause_refresh_timer7[15:0]</code>
TX 优先级 7 暂停量	<code>mac_tx_configuration_vector[335:320]</code>	<code>ctl_tx_pause_quanta7[15:0]</code>
TX 优先级 6 暂停量刷新值	<code>mac_tx_configuration_vector[319:304]</code>	<code>ctl_tx_pause_refresh_timer6[15:0]</code>
TX 优先级 6 暂停量	<code>mac_tx_configuration_vector[303:288]</code>	<code>ctl_tx_pause_quanta6[15:0]</code>

表 328: 发射器配置比较 (续)

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
TX 优先级 5 暂停量刷新值	mac_tx_configuration_vector[287:272]	ctl_tx_pause_refresh_timer5[15:0]
TX 优先级 5 暂停量	mac_tx_configuration_vector[271:256]	ctl_tx_pause_quanta5[15:0]
TX 优先级 4 暂停量刷新值	mac_tx_configuration_vector[255:240]	ctl_tx_pause_refresh_timer4[15:0]
TX 优先级 4 暂停量	mac_tx_configuration_vector[239:224]	ctl_tx_pause_quanta4[15:0]
TX 优先级 3 暂停量刷新值	mac_tx_configuration_vector[223:208]	ctl_tx_pause_refresh_timer3[15:0]
TX 优先级 3 暂停量	mac_tx_configuration_vector[207:192]	ctl_tx_pause_quanta3[15:0]
TX 优先级 2 暂停量刷新值	mac_tx_configuration_vector[191:176]	ctl_tx_pause_refresh_timer2[15:0]
TX 优先级 2 暂停量	mac_tx_configuration_vector[175:160]	ctl_tx_pause_quanta2[15:0]
TX 优先级 1 暂停量刷新值	mac_tx_configuration_vector[159:144]	ctl_tx_pause_refresh_timer1[15:0]
TX 优先级 1 暂停量	mac_tx_configuration_vector[143:128]	ctl_tx_pause_quanta1[15:0]
TX 优先级 0 暂停量刷新值	mac_tx_configuration_vector[127:112]	ctl_tx_pause_refresh_timer0[15:0]
TX 优先级 0 暂停量	mac_tx_configuration_vector[111:96]	ctl_tx_pause_quanta0[15:0]
启用 TX 优先级 7 流量控制	mac_tx_configuration_vector[95]	ctl_tx_pause_enable[7]
启用 TX 优先级 6 流量控制	mac_tx_configuration_vector[94]	ctl_tx_pause_enable[6]
启用 TX 优先级 5 流量控制	mac_tx_configuration_vector[93]	ctl_tx_pause_enable[5]
启用 TX 优先级 4 流量控制	mac_tx_configuration_vector[92]	ctl_tx_pause_enable[4]
启用 TX 优先级 3 流量控制	mac_tx_configuration_vector[91]	ctl_tx_pause_enable[3]
启用 TX 优先级 2 流量控制	mac_tx_configuration_vector[90]	ctl_tx_pause_enable[2]
启用 TX 优先级 1 流量控制	mac_tx_configuration_vector[89]	ctl_tx_pause_enable[1]
启用 TX 优先级 0 流量控制	mac_tx_configuration_vector[88]	ctl_tx_pause_enable[0]
启用自动 XON	mac_tx_configuration_vector[81]	
启用优先级流量控制	mac_tx_configuration_vector[80]	
TX 暂停帧源地址	mac_tx_configuration_vector[79:32]	ctl_tx_pause_da[47:0]
TX MTU 大小	mac_tx_configuration_vector[30:16]	非必需，因为 TX 可处理提供给 IP 的任意大小的帧。
启用 TX MTU	mac_tx_configuration_vector[14]	不适用，因为无需设置 TX。无需设置 MTU。
启用赤字空闲计数	mac_tx_configuration_vector[10]	不支持
TX LAN/WAN 模式	mac_tx_configuration_vector[9]	该功能当前不可用。10G/25G High Speed Ethernet IP 始终根据 IEEE 802.3 规范维持平均 IPG。 但是，您可将用户逻辑设计为使用 ctl_send_idle 插入空闲，并使用 ctl_tx_ipg_value[3:0] 来增大 IPG 值以满足 OC-192 SONET 要求。
启用 TX IFG 调整	mac_tx_configuration_vector[8]	ctl_tx_ipg_value 用于设置定制 IPG/IFG 的相 应值
启用 TX 保留前导码	mac_tx_configuration_vector[7]	ctl_tx_custom_preamble_enable 可启用定 制前导码的使用。Tx_preamblein 可提供定 制前导码，rx_preambleout 包含前导码字段 (来自接收到的帧)。
启用 TX 流量控制	mac_tx_configuration_vector[5]	ctl_tx_pause_enable[8:0]
启用 TX 巨型帧	mac_tx_configuration_vector[4]	非必需
启用 TX 频带内 FCS	mac_tx_configuration_vector[3]	ctl_tx_fcs_ins_enable

表 328: 发射器配置比较 (续)

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
启用 TX VLAN	mac_tx_configuration_vector[2]	非必需
TX 使能 (启用)	mac_tx_configuration_vector[1]	ctl_tx_enable
TX 复位	mac_tx_configuration_vector[0]	tx_reset

### 接收器配置

传统 XGEMAC 针对所有 RX 配置使用的是 rx\_configuration\_vector[95:0]，而 10G/25G High Speed Ethernet IP 核则通过部署多种信号来实现相同目的。两者比较结果如下表所示。

表 329: 接收器配置比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
启用 RX 优先级 7-0 流量控制	rx_configuration_vector[95:88]	ctl_rx_pause_enable[8:0]
启用优先级流量控制	rx_configuration_vector[80]	
RX 暂停帧 SA	rx_configuration_vector[79:32]	ctl_rx_pause_sa[47:0]
RX MTU 大小	rx_configuration_vector[30:16]	使用 ctl_rx_max_packet_len 和 ctl_rx_min_packet_len 信号来置位。
启用 RX MTU	rx_configuration_vector[14]	非必需。
协调子层故障抑制	rx_configuration_vector[10]	将用户逻辑设计为在接收到 RFI 时将 ctl_tx_send_idle 置位。
禁用控制帧长度检查	rx_configuration_vector[9]	不可用
禁用 RX 长度/类型错误	rx_configuration_vector[8]	在 10/25G High Speed Ethernet IP 核上无法禁用长度/类型错误。
启用 RX 保留前导码	rx_configuration_vector[7]	ctl_rx_custom_preamble_enable
启用 RX 流量控制	rx_configuration_vector[5]	ctl_rx_pause_enable[8:0]
启用 RX 巨型帧	rx_configuration_vector[4]	使用 ctl_rx_max_packet_len 信号来置位。
启用 RX 频带内 FCS	rx_configuration_vector[3]	ctl_rx_delete_fcs
启用 RX VLAN	rx_configuration_vector[2]	交由用户逻辑来实现此功能。
RX 使能 (启用)	rx_configuration_vector[1]	ctl_rx_enable
RX 复位	rx_configuration_vector[0]	rx_reset

### 状态矢量

表 330: 状态矢量比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
远端故障 RX	status_vector[1]	该功能不可用。
本地故障 RX	status_vector[0]	该功能不可用。

### TX 统计数据

传统 XGEMAC 针对所有 TX 统计数据使用的是 tx\_statistics\_vector[25:0]，而 10G/25G High Speed Ethernet IP 核则通过部署多种信号来实现相同目的。两者比较结果如下表所示。

表 331: TX 统计数据比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet MAC
发射的 PFC 帧	tx_statistics_vector[26]	stat_tx_user_pause
发射的暂停帧	tx_statistics_vector[25]	stat_tx_pause
有效字节	tx_statistics_vector[24:21]	stat_tx_total_good_bytes
VLAN 帧	tx_statistics_vector[20]	stat_tx_vlan
帧长度计数	tx_statistics_vector[19:5]	stat_tx_packet_* 信号, 这些信号同样可用于数据包直方图。
控制帧	tx_statistics_vector[4]	
欠载帧	tx_statistics_vector[3]	
多播帧	tx_statistics_vector[2]	stat_tx_multicast
广播帧	tx_statistics_vector[1]	stat_tx_broadcast
成功帧	tx_statistics_vector[0]	stat_tx_total_good_packets
TX 统计数据有效	tx_statistics_valid	非必需

## RX 统计数据

传统 XGEMAC 针对所有 RX 统计数据使用的是 `rx_statistics_vector[30:0]`, 而 10G/25G High Speed Ethernet IP 核则通过部署多种信号来实现相同目的。两者比较结果如下表所示。

表 332: RX 统计数据比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet MAC
PFC 帧	rx_statistics_vector[30]	stat_rx_user_pause
长度/类型超出范围	rx_statistics_vector[29]	stat_rx_inrangeerr
操作代码错误	rx_statistics_vector[28]	
流量控制帧	rx_statistics_vector[27]	stat_rx_pause
有效字节	rx_statistics_vector[26:23]	stat_rx_total_bytes[3:0]
VLAN 帧	rx_statistics_vector[22]	stat_rx_vlan
出界	rx_statistics_vector[21]	stat_rx_toolong
控制帧	rx_statistics_vector[20]	
帧长度计数	rx_statistics_vector [19:5]	stat_rx_total_good_bytes [13:0]
多播帧	rx_statistics_vector [4]	stat_rx_multicast
广播帧	rx_statistics_vector [3]	stat_rx_broadcast
FCS 错误	rx_statistics_vector [2]	stat_rx_packet_bad_fcs
坏帧	rx_statistics_vector [1]	stat_rx_total_good_packets 采样为 0
好帧	rx_statistics_vector [0]	stat_rx_total_good_packets 采样为 1
RX 统计数据有效	rx_statistics_valid	非必需

## 寄存器空间

传统 XGEMAC 和 10/25G High Speed Ethernet IP 核均提供 AXI 用户接口。虽然这 2 个 IP 核的大部分配置参数都相似, 但寄存器和寄存器的存储器映射不同。以下章节展示了配置、统计数据和其它寄存器之间的比较。

**TX 配置寄存器**

表 333: TX 配置寄存器比较

功能特性	传统 XGEMAC	10/25G HSEC
TX MTU 大小	@0x418: 发射器 MTU 配置字 [14:0]	非必需, 因为 TX 可处理提供给 IP 的任意大小的帧。
启用 TX MTU	@0x418: 发射器 MTU 配置字 [16]	不适用, 因为无需设置 TX MTU。
启用赤字空闲计数	@0x408: 发射器配置字 [24]	不支持
TX LAN/WAN 模式	@0x408: 发射器配置字 [26]	该功能当前不可用。10G/25G High Speed Ethernet IP 子系统始终根据 IEEE 802.3 规范维持平均 IPG。 但是, 您可将用户逻辑设计为使用 @0x000C: CONFIGURATION_TX_REG1[5] ctl_send_idle 插入空闲, 并使用 @0x000C: CONFIGURATION_TX_REG1[13:10] ctl_tx_ipg_value[3:0] 来增大 IPG 值以满足 OC-192 SONET 要求。
启用 TX IFG 调整	@0x408: 发射器配置字 [25]	@0x000C: CONFIGURATION_TX_REG1[13:10] ctl_tx_ipg_value 用于设置定制 IPG/IFG 的相应值。
启用 TX 保留前导码	@0x408: 发射器配置字 [23]	@0x000C: CONFIGURATION_TX_REG1[18] ctl_tx_custom_preamble_enable 可启用定制前导码的使用。Tx_preamblein 可提供定制前导码, rx_preambleout 包含前导码字段 (来自接收到的帧)。
启用 TX 流量控制	@0x40C: 流量控制配置寄存器 [30]	ctl_tx_pause_enable[8:0]
启用 TX 巨型帧	@0x408: 发射器配置字 [30]	非必需。
启用 TX 频带内 FCS	@0x408: 发射器配置字 [29]	@0x000C: CONFIGURATION_TX_REG1[1] ctl_tx_fcs_ins_enable
启用 TX VLAN	@0x408: 发射器配置字 [27]	非必需。
TX 使能 (启用)	@0x408: 发射器配置字 [28]	@0x000C: CONFIGURATION_TX_REG1[0] ctl_tx_enable
TX 复位	@0x408: 发射器配置字 [31]	@0x0004: RESET_REG[31] tx_reset

**RX 配置**

表 334: RX 配置比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP 核
启用 RX 优先级 7-0 流量控制	位 95:88	@0x0094: CONFIGURATION_RX_FLOW_CONTROL_REG1 [7:0]
启用优先级流量控制	位 80	不支持传统 PFC
RX 暂停帧 SA	@0x404: 接收器配置字 1[47:32], @0x400: 接收器配置字 0	@0x00C4: CONFIGURATION_RX_FLOW_CONTROL_SA_REG1_MSB[15:0], @0x00C0: CONFIGURATION_RX_FLOW_CONTROL_SA_REG1_LSB
RX MTU 大小	@0x414: 接收器 MTU 配置字 [14:0]	使用 @0x0018: CONFIGURATION_RX_MTU[30:16] ctl_rx_max_packet_len 和 @0x0018: CONFIGURATION_RX_MTU[7:0] ctl_rx_min_packet_len 信号来置位
启用 RX MTU	@0x414: 接收器 MTU 配置字 [16]	非必需。
协调子层故障抑制	@0x410: 协调子层配置字 [27]	将用户逻辑设计为在接收到 RFI 时将 @0x000C: CONFIGURATION_TX_REG1[5] ctl_tx_send_idle 置位

表 334: RX 配置比较 (续)

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP 核
禁用控制帧长度检查	@0x404: 接收器配置字 1[24]	不可用
禁用 RX 长度/类型错误	@0x404: 接收器配置字 1[25]	在 10/25G High Speed Ethernet Subsystem 上无法禁用长度/类型错误。
启用 RX 保留前导码	@0x404: 接收器配置字 1[26]	@0x0014: CONFIGURATION_RX_REG1[11] ctl_rx_custom_preamble_enable
启用 RX 流量控制	@0x40C: 流量控制配置寄存器 [29]	@0x0094: CONFIGURATION_RX_FLOW_CONTROL[8:0] ctl_rx_pause_enable[8:0]
启用 RX 巨型帧	@0x404: 接收器配置字 1[30]	使用 ctl_rx_max_packet_len 信号来置位
启用 RX 频带内 FCS	@0x404: 接收器配置字 1[29]	@0x0014: CONFIGURATION_RX_REG1[1] ctl_rx_delete_fcs
启用 RX VLAN	@0x404: 接收器配置字 1[27]	交由用户逻辑来实现此功能
RX 使能 (启用)	@0x404: 接收器配置字 1[28]	@0x0014: CONFIGURATION_RX_REG1[0] ctl_rx_enable
RX 复位	@0x404: 接收器配置字 1[31]	@0x0004: RESET_REG[30] rx_reset

## 状态矢量

表 335: 状态矢量比较

功能特性	传统 XGEMAC	10/25G High Speed Ethernet IP
远端故障 RX	@0x410: 协调子层配置字 [29]	该功能不可用。
本地故障 RX	@0x410: 协调子层配置字 [28]	该功能不可用。

## 统计数据计数器

下表列出了不同的统计数据计数器及其地址。

表 336: 统计数据计数器

地址 (十六进制)	寄存器	寄存器	地址 (十六进制)
0x200	已接收的字节数 (LSW)	STAT_RX_TOTAL_BYTES_LSB	0x0808
0x204	已接收的字节数 (MSW)	STAT_RX_TOTAL_BYTES_MSB	0x080C
0x208	已发射的字节数 (LSW)	STAT_TX_TOTAL_BYTES_LSB	0x0710
0x20C	已发射的字节数 (MSW)	STAT_TX_TOTAL_BYTES_MSB	0x0714
0x210	已接收的大小过小的帧数 (LSW)	STAT_RX_UNDERSIZE_LSB	0x0898
0x214	已接收的大小过小的帧数 (MSW)	STAT_RX_UNDERSIZE_MSB	0x089C
0x218	已接收的碎片帧数 (LSW)	STAT_RX_FRAGMENT_LSB	0x08A0
0x21C	已接收的碎片帧数 (MSW)	STAT_RX_FRAGMENT_MSB	0x08A4
0x220	已正常接收的 64 字节帧数 (LSW)	STAT_RX_PACKET_64_BYTES_LSB	0x0828
0x224	已正常接收的 64 字节帧数 (MSW)	STAT_RX_PACKET_64_BYTES_MSB	0x082C

表 336: 统计数据计数器 (续)

地址 (十六进制)	寄存器	寄存器	地址 (十六进制)
0x228	已正常接收的 65 - 127 字节帧数 (LSW)	STAT_RX_PACKET_65_127_BYTES LSB	0x0830
0x22C	已正常接收的 65 - 127 字节帧数 (MSW)	STAT_RX_PACKET_65_127_BYTES MSB	0x0834
0x230	已正常接收的 128 - 255 字节帧数 (LSW)	STAT_RX_PACKET_128_255_BYTES LSB	0x0838
0x234	已正常接收的 128 - 255 字节帧数 (MSW)	STAT_RX_PACKET_128_255_BYTES MSB	0x083C
0x238	已正常接收的 256 - 511 字节帧数 (LSW)	STAT_RX_PACKET_256_511_BYTES LSB	0x0840
0x23C	已正常接收的 256 - 511 字节帧数 (MSW)	STAT_RX_PACKET_256_511_BYTES MSB	0x0844
0x240	已正常接收的 512 - 1023 字节帧数 (LSW)	STAT_RX_PACKET_512_1023_BYTES LSB	0x0848
0x244	已正常接收的 512 - 1023 字节帧数 (MSW)	STAT_RX_PACKET_512_1023_BYTES MSB	0x084C
0x248	已正常接收的 1024 - MaxFrameSize (最大帧大小) 字节帧数 (LSW)	STAT_RX_PACKET_1024_1518_BYTES LSB	0x0850
		STAT_RX_PACKET_1519_1522_BYTES LSB	0x0858
		STAT_RX_PACKET_1523_1548_BYTES LSB	0x0860
		STAT_RX_PACKET_1549_2047_BYTES LSB	0x0868
		STAT_RX_PACKET_2048_4095_BYTES LSB	0x0870
		STAT_RX_PACKET_4096_8191_BYTES LSB	0x0878
		STAT_RX_PACKET_8192_9215_BYTES LSB	0x0880
		STAT_RX_PACKET_LARGE LSB	0x0888
0x24C	已正常接收的 1024 - MaxFrameSize (最大帧大小) 字节帧数 (MSW)	STAT_RX_PACKET_1024_1518_BYTES MSB	0x0854
		STAT_RX_PACKET_1519_1522_BYTES MSB	0x085C
		STAT_RX_PACKET_1523_1548_BYTES MSB	0x0864
		STAT_RX_PACKET_1549_2047_BYTES MSB	0x086C
		STAT_RX_PACKET_2048_4095_BYTES MSB	0x0874
		STAT_RX_PACKET_4096_8191_BYTES MSB	0x087C
		STAT_RX_PACKET_8192_9215_BYTES MSB	0x0884
		STAT_RX_PACKET_LARGE MSB	0x088C
0x250	已正常接收的大小过大的 64 字节帧数 (LSW)	STAT_RX_OVERSIZE LSB	0x08A8
0x254	已正常接收的大小过大的 64 字节帧数 (MSW)	STAT_RX_OVERSIZE MSB	0x08AC
0x258	已正常发射的 64 字节帧数 (LSW)	STAT_TX_PACKET_64_BYTES LSB	0x0720
0x25C	已正常发射的 64 字节帧数 (MSW)	STAT_TX_PACKET_64_BYTES MSB	0x0724
0x260	已正常发射的 65 - 127 字节帧数 (LSW)	STAT_TX_PACKET_65_127_BYTES LSB	0x0728
0x264	已正常发射的 65 - 127 字节帧数 (MSW)	STAT_TX_PACKET_65_127_BYTES MSB	0x072C
0x268	已正常发射的 128 - 255 字节帧数 (LSW)	STAT_TX_PACKET_128_255_BYTES LSB	0x0730
0x26C	已正常发射的 128 - 255 字节帧数 (MSW)	STAT_TX_PACKET_128_255_BYTES MSB	0x0734
0x270	已正常发射的 256 - 511 字节帧数 (LSW)	STAT_TX_PACKET_256_511_BYTES LSB	0x0738
0x274	已正常发射的 256 - 511 字节帧数 (MSW)	STAT_TX_PACKET_256_511_BYTES MSB	0x073C
0x278	已正常发射的 512 - 1023 字节帧数 (LSW)	STAT_TX_PACKET_512_1023_BYTES LSB	0x0740
0x27C	已正常发射的 512 - 1023 字节帧数 (MSW)	STAT_TX_PACKET_512_1023_BYTES MSB	0x0744

表 336: 统计数据计数器 (续)

地址 (十六进制)	寄存器	寄存器	地址 (十六进制)
0x280	已正常发射的 1024 - MaxFrameSize (最大帧大小) 字节帧数 (LSW)	STAT_TX_PACKET_1024_1518_BYTES LSB	0x0748
		STAT_TX_PACKET_1519_1522_BYTES LSB	0x0750
		STAT_TX_PACKET_1523_1548_BYTES LSB	0x0758
		STAT_TX_PACKET_1549_2047_BYTES LSB	0x0760
		STAT_TX_PACKET_2048_4095_BYTES LSB	0x0768
		STAT_TX_PACKET_4096_8191_BYTES LSB	0x0770
		STAT_TX_PACKET_8192_9215_BYTES LSB	0x0778
		STAT_TX_PACKET_LARGE LSB	0x0780
0x284	已正常发射的 1024 - MaxFrameSize (最大帧大小) 字节帧数 (MSW)	STAT_TX_PACKET_1024_1518_BYTES MSB	0x074C
		STAT_TX_PACKET_1519_1522_BYTES MSB	0x0754
		STAT_TX_PACKET_1523_1548_BYTES MSB	0x075C
		STAT_TX_PACKET_1549_2047_BYTES MSB	0x0764
		STAT_TX_PACKET_2048_4095_BYTES MSB	0x076C
		STAT_TX_PACKET_4096_8191_BYTES MSB	0x0774
		STAT_TX_PACKET_8192_9215_BYTES MSB	0x077C
		STAT_TX_PACKET_LARGE MSB	0x0784
0x288	已正常发射的大小过大的 64 字节帧数 (LSW)	不适用	
0x28C	已正常发射的大小过大的 64 字节帧数 (MSW)		
0x290	已正常接收的帧数 (LSW)	STAT_RX_TOTAL_GOOD_PACKETS LSB	0x0810
0x294	已正常接收的帧数 (MSW)	STAT_RX_TOTAL_GOOD_PACKETS MSB	0x0814
0x298	帧校验序列错误 (LSW)	STAT_RX_PACKET_BAD_FCS LSB	0x08C8
0x29C	帧校验序列错误 (MSW)	STAT_RX_PACKET_BAD_FCS MSB	0x08CC
0x2A0	已正常接收的广播帧数 (LSW)	STAT_RX_BROADCAST LSB	0x8E8
0x2A4	已正常接收的广播帧数 (MSW)	STAT_RX_BROADCAST MSB	0x8EC
0x2A8	已正常接收的多播帧数 (LSW)	STAT_RX_MULTICAST LSB	0x08E0
0x2AC	已正常接收的多播帧数 (MSW)	STAT_RX_MULTICAST MSB	0x08E4
0x2B0	已正常接收的控制帧数 (LSW)		
0x2B4	已正常接收的控制帧数 (MSW)		
0x2B8	长度/类型超出范围 (LSW)	STAT_RX_INRANGEERR LSB	0x0908
0x2BC	长度/类型超出范围 (MSW)	STAT_RX_INRANGEERR MSB	0x090C
0x2C0	已正常接收的含 VLAN 标签的帧数 (LSW)	STAT_RX_VLAN LSB	0x08F0
0x2C4	已正常接收的含 VLAN 标签的帧数 (MSW)	STAT_RX_VLAN MSB	0x08F4
0x2C8	已正常接收的 PAUSE (暂停) 帧数 (LSW)	STAT_RX_PAUSE LSB	0x08F8
0x2CC	已正常接收的 PAUSE (暂停) 帧数 (MSW)	STAT_RX_PAUSE MSB	0x08FC
0x2D0	已接收的含不受支持的操作代码的控制帧数 (LSW)		
0x2D4	已接收的含不受支持的操作代码的控制帧数 (MSW)		
0x2D8	已正常发射的帧数 (LSW)	STAT_TX_TOTAL_GOOD_PACKETS LSB	0x0708

表 336: 统计数据计数器 (续)

地址 (十六进制)	寄存器	寄存器	地址 (十六进制)
0x2DC	已正常发射的帧数 (MSW)	STAT_TX_TOTAL_GOOD_PACKETS_MSB	0x070C
0x2E0	已正常发射的广播帧数 (LSW)	STAT_TX_BROADCAST_LSB	0x07E0
0x2E4	已正常发射的广播帧数 (MSW)	STAT_TX_BROADCAST_MSB	0x07E4
0x2E8	已正常发射的多播帧数 (LSW)	STAT_TX_MULTICAST_LSB	0x07D8
0x2EC	已正常发射的多播帧数 (MSW)	STAT_TX_MULTICAST_MSB	0x07DC
0x2F0	欠载错误数 (LSW)		
0x2F4	欠载错误数 (MSW)		
0x2F8	已正常发射的控制帧数 (LSW)		
0x2FC	已正常发射的控制帧数 (MSW)		
0x300	已正常发射的含 VLAN 标签的帧数 (LSW)	STAT_TX_VLAN_LSB	0x07E8
0x304	已正常发射的含 VLAN 标签的帧数 (MSW)	STAT_TX_VLAN_MSB	0x07EC
0x308	已正常发射的 PAUSE (暂停) 帧数 (LSW)	STAT_TX_PAUSE_LSB	0x07F0
0x30C	已正常发射的 PAUSE (暂停) 帧数 (MSW)	STAT_TX_PAUSE_MSB	0x07F4

## 暂停处理

以下章节概述了基于优先级的流量控制的配置。

表 337: 暂停处理

地址 (十六进制)	寄存器	寄存器	地址 (十六进制)
0x480	优先级 0 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG1 [15:0]	0x0058
0x484	优先级 1 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG1 [31:16]	0x0058
0x488	优先级 2 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG2 [15:0]	0x005C
0x48C	优先级 3 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG2 [31:16]	0x005C
0x490	优先级 4 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG3 [15:0]	0x0060
0x494	优先级 5 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG3 [31:16]	0x0060
0x498	优先级 6 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG4 [15:0]	0x0064
0x49C	优先级 7 暂停量寄存器	CONFIGURATION_TX_FLOW_CONTROL_QUANTA_REG4 [31:16]	0x0064
0x4A0	传统暂停刷新寄存器	不支持传统 PFC	

## MDIO 控制寄存器

10G/25G High Speed Ethernet IP 不提供 MDIO 站主控，因此不含任何 MDIO 控制寄存器。

## 中断寄存器

通常，MDIO 操作完成后会生成中断以表示操作完成；由于没有 MDIO 主控，因此没有中断寄存器。

## PCS/PMA MDIO 寄存器映射

同样，由于未提供任何 MDIO 接口，因此没有 MDIO 寄存器可用于 PCS/PMA 接口。

### AXI4-Stream 接口

10G/25G High Speed Ethernet IP 子系统可提供 64 位和 32 位 AXI4-Stream 接口用作为数据路径，就像传统 10G Ethernet IP 子系统一样。请注意，在 RX 接口上使用 tuser 位时存在如下差异。下表对 TX 和 RX 上的 tuser 信号定义进行了比较。

表 338: TX 和 RX 中的 tuser 信号定义比较

信号	传统 XGEMAC	10/25G High Speed Ethernet IP
RX AXI4-Stream tuser	m_axis_rx_tuser AXI4-Stream 用户边带接口。 0 指示接收到的包无效。 1 指示接收到的数据包有效。	rx_axis_tuser AXI4-Stream 用户边带接口。 1 指示接收到的包无效。 0 指示接收到的数据包有效。
TX AXI4-Stream tuser	s_axis_tx_tuser AXI4-Stream 用户信号，用于表示显式欠载	tx_axis_tuser AXI4-Stream 用户边带接口。 1 指示接收到的包无效。 0 指示接收到的数据包有效。

# 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。



**提示：**如果 IP 生成因错误而停止，则可能存在许可证问题。如需了解更多详情，请参阅 [许可证检查器](#)。

## 在 Xilinx.com 上寻求帮助

为了帮助您在使用子系统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

本产品指南是与该子系统相关的主要文档。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。

请参阅[赛灵思以太网 IP 解决方案中心](#)。

### 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”框找到对应该子系统的答复记录。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。

## 该核的主答复记录

AR: [64710](#)

## 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 10G/25G Ethernet Subsystem 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](#))。

## 参考板

各种赛灵思开发板均支持 10G/25G Ethernet Subsystem。这些开发板可用于进行原型设计并确定核可与系统通信。

- UltraScale FPGA 评估板：
  - VCU107
  - VCU108

# 仿真调试

## 仿真器许可证可用性

如果仿真器无法启动，则表示您可能没有有效许可证。请确保许可证处于最新状态。原因也可能是您的企业拥有可用于某一其它仿真器的许可证，因此，请尝试提供的所有脚本。

## 仿真缓慢

在某些情况下，仿真可能看上去运行较为缓慢。如果仿真太慢而难以接受，以下建议或有助于提升运行时间性能。

- 使用含更多内存、运行速度更快的计算机。
- 使用平台负载共享工具 (LSF)（如果可在您的组织中使用）。
- 绕过赛灵思收发器（可能需要客户自行创建自己的测试激励文件）。
- 减少发送的数据包的数量。
- 如果使用设计示例，请参阅 [仿真加速](#) 以缩短设计示例中的等待时间。

## 仿真失败且未完成

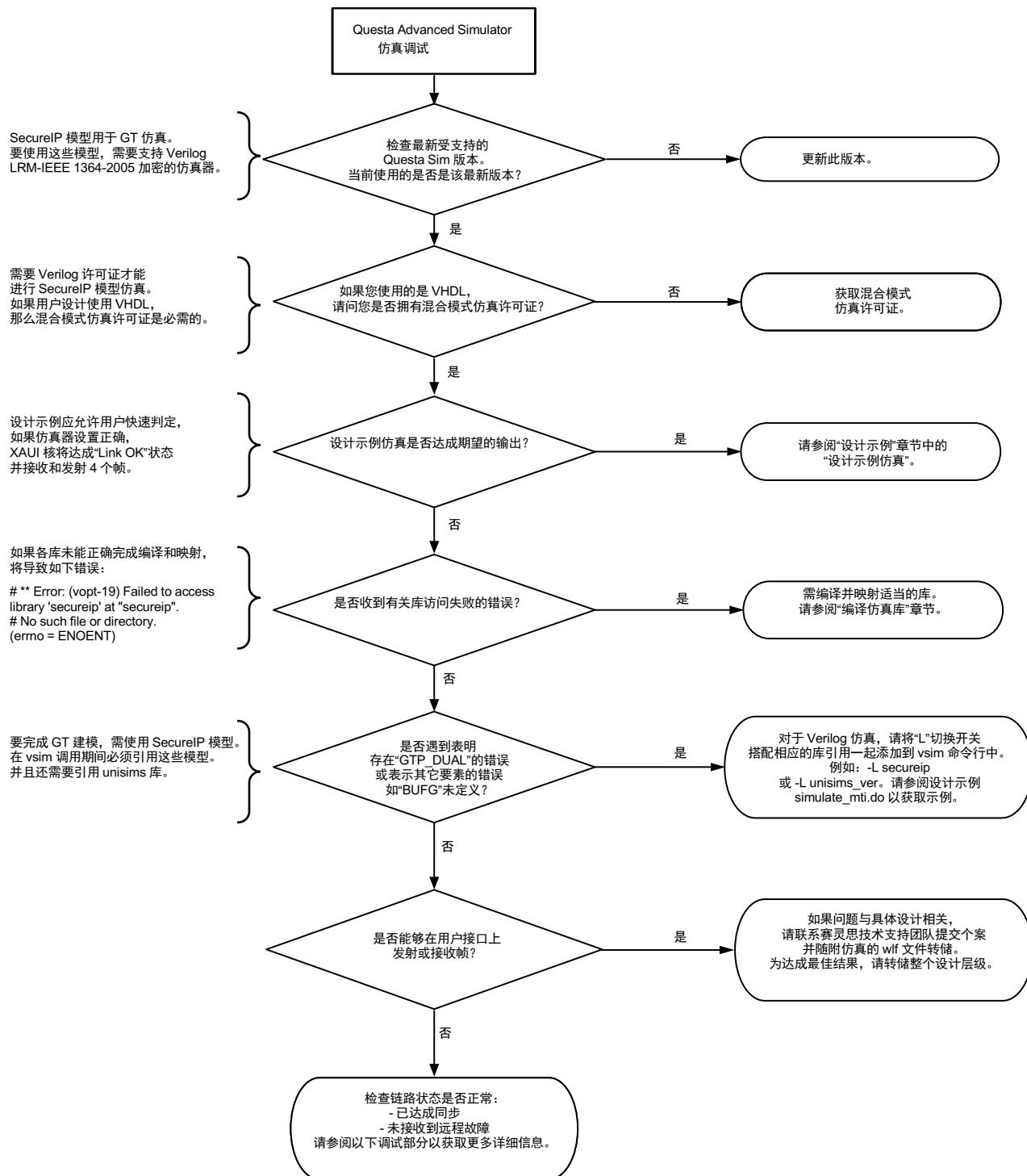
如果仿真样本在成功完成前即失败并挂起，那么可能发生了超时。请确保针对仿真超时所设置的值足以容纳仿真等待时间段，例如，在通道对齐阶段期间的时间段。

## 仿真完成但失败

如果仿真样本可完成但失败，请联系赛灵思技术支持。每个版本在交货前都经过测试，通常测试都能成功完成。请参阅仿真样本 log 日志文件，以确认仿真行为是否符合期望的行为。

Questa® Advanced Simulator 的仿真调试流程如下图所示。类似的方法可用于其它仿真器。

图 66: Mentor Graphics Questa Advanced Simulator 仿真调试流程



X25615-101122

# 硬件调试

硬件问题各不相同，可能是链路初始化问题，也可能是测试数小时后才会显现的问题。本节提供了常见问题的调试步骤。Vivado 调试功能是可用于硬件调试的宝贵资源。可以通过调试功能来探测以下各个部分中提到的信号名称，以便对特定问题进行调试。

## 常规检查

确保核的所有时序约束都已正确整合在设计示例中，并且在实现期间已满足所有约束条件。

- 在布局布线后的时序仿真中是否能够正常工作？如果在硬件中发现问题，但在时序仿真中没有出现问题，则可能表示存在 PCB 问题。确保所有时钟源均处于活动状态且无任何错误。
- 如果在设计中使用混合模式时钟管理器 (MMCM)，请通过监控 LOCKED 端口来确保所有 MMCM 都已被锁定。
- 如果您的输出为 0，请检查您的许可。

## 时序

在硬件中尝试实现 IP 前，请确保根据 Vivado 工具满足时序要求。

## 收发器专用检查

- 确保 txn/txp 和 rxn/rxp 线的极性没有反转。如果反转，可使用收发器的 TXPOLARITY 和 RXPOLARITY 端口进行修复。
- 确认收发器没有保持在复位状态或仍在初始化中。收发器的 RESETDONE 输出指示收发器于何时就绪。
- 将收发器置于并行或串行近端环回模式。
- 如果在收发器串行环回模式时操作正常，但在通过光缆执行环回时不正常，则可能表明光学模块存在故障。
- 如果核在收发器并行环回模式下操作正常，但在串行环回模式时不正常，这可能表明收发器有问题。
- 轻微的误码率可以通过调整收发器的发射器预加重和差分摆幅控制属性来解决。

## 以太网专用检查

在 10G/25G High Speed Ethernet Subsystem 的首次硬件测试期间，会发生几个常见的问题。这些问题应按如下所述方式来进行检查。

其中假定 10G/25G Ethernet Subsystem 已成功通过硬件中要实现的所有仿真测试。这是执行任何种类的硬件调试的先决条件。

常用调试顺序如下所述：

1. 清理信号完整性。
2. 确保 SerDes 可实现时钟数据恢复 (CDR) 锁定。
3. 检查 10G/25G Ethernet Subsystem IP 是否都已实现字同步。
4. 下一步，继续执行接口和协议调试。

## 信号完整性

如果首次执行开发板初始化，并且 10G/25G Ethernet Subsystem 似乎尚未实现字同步，那么很可能是信号完整性问题。必须先解决信号完整性问题，然后才能执行任何其它调试操作。

信号完整性应与 10G/25G Ethernet Subsystem 分开，单独进行调试。应执行以下过程。请注意，假定 PCB 本身是根据所需的走线阻抗和走线长度（包括 IEEE 802.3 规范中规定的偏差要求）设计和制造的。

- 收发器设置
- 检查噪声
- 误码率测试

如需帮助进行收发器和信号完整性调试，请联系赛灵思技术支持。

## N/P 交换

如果差分信号的正负信号进行交换，则在此通道上数据无法正确接收数据。您应验证链路中每个差分对的极性是否正确。

## 时钟设置和复位

请参阅 [时钟设置](#) 和 [复位](#) 以了解相关要求。

请确保 10G/25G Ethernet Subsystem 和赛灵思收发器参考时钟的时钟频率都与订购该子系统时所申请的配置相匹配。核时钟具有与之关联的最小频率。最大核时钟频率则由时序约束来确定。最小核时钟频率衍生自所需的以太网带宽加上为时钟容限、时钟漂移和时钟抖动所保留的裕度。

调试期间首先需要验证的是确保复位保持断言有效，直至时钟稳定为止。时钟频率必须稳定并且不含毛刺，随后才能使 10G/25G Ethernet Subsystem 解复位。此要求对 SerDes 时钟与核时钟都适用。

如果后续在时钟上检测到任何不稳定现象，则 10G/25G Ethernet Subsystem 必须复位。此类不稳定现象的示例之一是 CDR 锁定丢失。用户逻辑应判定需要复位的所有外部条件（例如，时钟毛刺、CDR 锁定丢失或电源毛刺等）。

当串行数据变为有效后，GT 需要 GTRXRESET 以确保对数据施加正确的 CDR 锁定。在上电、复位或重新连接链路伙伴后，这是必需的。在核级别，为了避免链路 TX 侧发生中断，可使用 `gtwiz_reset_rx_datapath` 触发复位。可通过对光学模块信号丢失使用信号检测或反相（如果可用）来触发复位。如果信号检测或信号丢失不可用，则可添加超时逻辑以监控对齐是否尚未完成，并发出 `gtwiz_reset_rx_datapath` 复位。

除非该子系统已复位，否则无法执行配置更改。配置更改的示例包括更改最大数据包长度设置。检查端口列表上特定信号的描述，以判定此要求是否适用于要更改的参数。

## RX 调试

请参阅端口列表章节以获取可用于 RX 调试的诊断信号的描述。

### **stat\_rx\_block\_lock**

此信号用于指示接收器已检测到并锁定到由 01 或 10 控制或数据报头所定义的字边界。这是确保 10G/25G Ethernet Subsystem IP 正常工作的第一步。



**注意!** 在无信号输入的某些情况下，SerDes 接收器呈现稳定交替模式 1010101...这可能导致错误的块锁定，但仍表明接收器已检测到此模式。

#### **stat\_rx\_bad\_fcs**

无效 FCS 指示在接收到的包内存在位元错误。发生 FCS 错误的原因可能是存在任意数量的数据包损坏（例如，线路上存在噪声）。

#### **stat\_rx\_local\_fault**

可能在本地生成或接收到本地故障指示。本地故障的部分原因可能包括：

- 块锁定未完成
- 误码率过高
- 上溢或下溢

#### **环回检查**

如果根据 IEEE 802.3 标准正确传输了以太网包，则不应出现 RX 错误。但必须首先验证接收到的信号的信号完整性。

为帮助调试，选中“Include AXI4-Lite enable”（包含 AXI4-Lite 使能）选项时，可使用 `ctl_local_loopback` 信号执行局部环回；否则，对于非 AXI4-Lite 配置，可使用 `gt_loopback_in` 信号执行局部环回。这样会将 TX SerDes 连接到 RX SerDes，从而有效地避免了潜在的信号完整性问题。收发器将布局到“PMA 环回”内，如需了解详细描述，请参阅收发器产品指南。通过这种方式，接收到的数据可与发射的包进行对比检查，以验证逻辑是否正常运行。

## **TX 调试**

如果在链路伙伴处接收到无效包，但在长期空闲的序列中并未出现错误，这可能表明 TX AXI 接口输入上存在格式错误的包。请检查这些包是否存在欠载。当 `tx_axis_tlast` 为高电平时，断言 `tx_axis_tuser` 为高电平有效即可表明存在显式欠载。当 `tx_axis_tvalid` 断言无效，且 `tx_axis_tlast` 未断言有效时，隐式欠载会导致帧传输中止。欠载的包发射时将包含错误块 (0xFE)。

---

## **协议接口调试**

要使 10G/25G Ethernet Subsystem 实现无错数据传输，应遵循 802.3 规范进行操作。请注意，应始终首先确保信号完整性，然后再继续进行协议调试。

### **诊断信号**

有许多错误指示可用于检查是否存在协议违例。请仔细阅读每项指示的描述，查看它是否适用于特定调试问题。

以下是建议的调试顺序：

1. 请确保已实现字同步。
2. 确保不存在解扰器错误。
3. 消除 CRC32 错误（如果有）。
4. 确保正确遵循协议进行操作。

- 确保发送数据包时不存在上溢或下溢状况。

## 统计数据计数器

实现无错通信后，可监控统计数据指示符以确保流量特性符合期望。请注意，部分信号仅为选通，即计数器不包含在子系统中。这样即可对计数器大小进行自定义。（可选）计数器可搭配 AXI 接口使用。

## 调试自动协商和链路训练

### 自动协商

要启用自动协商，请执行以下操作：

- ctl\_autoneg\_enable = 1
- ctl\_autoneg\_bypass = 0

设置 `ctl_an_*` 以播发所期望的自动协商设置。

使用控制和状态接口时，设计示例会将 `ctl_an_*` 值绑定到有效设置。如果使用的是寄存器接口，请参阅 [使用 AXI4-Lite 接口进行自动协商和链路训练的开发板测试步骤](#) 以了解寄存器顺序。

### 链路训练

要启用链路训练，请将 `ctl_lt_training_enable` 设置为 1。

- 这样核不会实际执行任何训练。它仅提供第 72.6.10 条所需的控制协议。训练算法由用户负责。
- 该核不会监控 RX 眼，也不会向链路伙伴 TX 发送任何预置、初始化或系数控制请求。建议将 `ctl_lt_rx_trained` 设为 1。设置 `ctl_lt_rx_trained` 即可告知链路伙伴，您的 RX 训练已完成，并且您将不会发送任何其它预置、初始化或系数更改。
- 该核不会对任何 GT TX 振幅或系数控制设置进行调整以响应从链路伙伴接收到的任何训练消息。设计示例链路训练 Place\_Holder 逻辑可指示何时已达到最大限制。这样应可允许链路训练成功完成。

### 随机数

`nonce_seed` 必须设置为非零值。

- 如果要连接同一开发板上含相同随机数种子的 2 个端口，那么复位释放时间必须不同。
- 如果 `nonce_seed` 发生更改，则需通过 `an_reset` 来加载新的值。这包括使用 AXI4-Lite 寄存器更改 `nonce_seed`。

### 下一页信号

如果链路伙伴发送下一页信号，`ctl_an_loc_np_ack` 必须设置为高电平，以确认下一页信号并允许自动协商完成操作。接收到下一页信号或始终绑定到高电平后，此控制信号即可设置为高电平。

### 有关阶段和状态信号的详细信息

- 自动协商启动时处于 TX 禁用状态，不显示任何数据，这样可以确保两端链路均处于中断状态。  
`stat_an_stat_tx_disable` 信号将进行切换并保持 1 个周期，以指示此阶段开始。

2. TX 禁用状态后, 将交换自动协商信息。在此阶段中, `stat_an_rxcdrhold` 保持高电平。`stat_an_lp_autoneg_able` 和 `stat_an_lp_ability_valid` 信号将切换至高电平并保持 1 个时钟周期, 以指示何时 `stat_an_lp*` 信息有效。
3. 链路训练开始时, `stat_an_start_an_good_check` 信号将切换至高电平并保持 1 个时钟周期。`stat_an_rxcdrhold` 信号将断言无效并切换 `gtwiz_reset_rx_datapath`。链路训练开始后, 将采用 500 毫秒定时器来执行训练和块锁定、以任务模式建立连接并完成正常 PCS 操作, 否则自动协商将重新启动。当链路训练块完成帧同步后, `stat_lt_frame_lock` 信号会变为高电平, `stat_lt_rx_sof` 将进行切换。在训练帧边界处, `stat_lt_rx_sof` 信号将继续切换至高电平, 并保持 1 个时钟周期。
4. 当链路训练完成时, `stat_lt_signal_detect` 信号将断言有效, 指示开始正常 PCS 操作。
5. 当块锁定、同步并对齐 (对应多通道核), `stat_rx_status` 和 `stat_rx_valid_ctrl_code` (`stat_rx_valid_ctrl_code` 仅适用于单通道 10G/25G 核) 转至高电平时, `an_autoneg_complete` 信号会转至高电平。
6. `an_autoneg_complete` 信号必须在 500 ms 超时范围内转至高电平, 否则自动协商会重新启动。如果 `stat_rx_status` 在任何时间返回低电平, 那么自动协商会重新启动。

## 仿真与环回

自动协商 TX 禁用状态需耗时 50 毫秒才能完成仿真。使用不含预编译的 IP 库的 SIM\_SPEED\_UP 选项可缩短等待时间。请参阅答复记录 [73518](#) 以获取有关关闭预编译库的更多信息。

自动协商无法在环回过程中完成, 因为自动协商要求从链路伙伴接收到的随机数值必须与发送给链路伙伴的随机数值不同。

可添加到 ILA 以供调试的起始信号列表如下所示:

- `sys_reset`
- `an_reset`
- `ctl_an_*`
- `ctl_lt_*`
- `stat_an_start_tx_disable`
- `stat_an_rx cdrhold`
- `stat_an_lp_autoneg_able`
- `stat_an_lp_ability_valid`
- `stat_an_start_an_good_check`
- `stat_lt_frame_lock`
- `stat_lt_signal_detect`
- `stat_lt_link_training`
- `stat_lt_link_training_fail`
- `stat_rx_block_lock`
- `stat_rx_synced` (仅在多通道核上可用)
- `stat_rx_aligned` (仅在多通道核上可用)
- `stat_rx_valid_ctrl_code` (仅在 10G/25G 核上可用)
- `stat_rx_status`
- `stat_rx_bad_code`
- `stat_rx_hi_ber`

如果使用的线速率支持第 74 条法尔码 FEC, 则此信号列表如下:

- `stat_fec_inc_cant_correct_count`
- `stat_fec_lock_error`

- stat\_fec\_rx\_lock
- stat\_fec\_inc\_correct\_count
- ctl\_an\_fec\_10g\_request
- ctl\_fec\_rx\_enable
- ctl\_fec\_tx\_enable
- stat\_an\_fec\_enable
- stat\_an\_lp\_fec\_10g\_ability
- stat\_an\_lp\_fec\_10g\_request

如果使用的线速率支持 RS-FEC，则此信号列表如下：

- ctl\_tx\_rsfec\_enable
- ctl\_rx\_rsfec\_enable
- stat\_rx\_rsfec\_am\_lock
- stat\_an\_rs\_fec\_enable

**注释：**如果链路伙伴发送下一页信号，那么必须设置 `ctl_an_lo_np_ack` 信号。此端口可绑定到高电平。

## 使用 AXI4-Lite 接口调试自动协商和链路训练

要使用 AXI4-Lite 接口调试自动协商和链路训练，请参阅 [使用 AXI4-Lite 接口进行自动协商和链路训练的开发板测试步骤](#)。

# 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，可配合本产品指南一起使用：

1. 《25/50G 千兆以太网联盟工作计划表 3 (v1.6)》(<https://25gethernet.org/>)
2. 《IEEE 以太网标准》(IEEE 802.3-2015 标准)
3. 《IEEE 以太网标准 - 修订 2：以 25 Gb/s 运行时的媒体访问控制参数、物理层和管理参数》(IEEE 802.3by 标准)
4. 《Vivado Design Suite 用户指南：采用 IP integrator 设计 IP 子系统》(UG994)
5. 《Vivado Design Suite 用户指南：采用 IP 进行设计》(UG896)

6. 《Vivado Design Suite 用户指南：入门指南》(UG910)
7. 《Vivado Design Suite 用户指南：逻辑仿真》(UG900)
8. 《Vivado Design Suite 用户指南：编程和调试》(UG908)
9. 《Vivado Design Suite 用户指南：实现》(UG904)
10. 《Vivado Design Suite：AXI 参考指南》(UG1037)
11. 《IEEE 1588 网络测量和控制系统的精密时钟同步协议标准》(IEEE 1588)
12. 《IEEE 局域网和城域网标准 - 适用于前传的时间敏感网络》(IEEE 802.1CM-2018 标准)
13. 《25G IEEE 802.3by Reed-Solomon Forward Error Correction LogiCORE IP 产品指南》(PG217) (需注册)
14. 《UltraScale 架构 GTH 收发器用户指南》(UG576)
15. 《UltraScale 架构 GTY 收发器用户指南》(UG578)
16. 《Vivado Design Suite 用户指南：使用约束》(UG903)
17. 《Vivado Design Suite 教程：采用 IP integrator 设计 IP 子系统》(UG995)

---

## 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 4.1 版</b>	
端口定时器寄存器	通过提供 CORE_CONFIGURATION 和 RESYNC_CLK_PERIOD 值，添加针对混合定时器模块的支持。
<b>2022 年 5 月 13 日 4.1 版</b>	
复位	添加 Versal 器件复位图示。
入口	更新 1588 时间戳精度信息。
产品规格	添加有关 PTP 时间戳支持的注释。
10G/25G Ethernet Subsystem 的配置寄存器	将 reg 0154 重命名为 GT_WIZ_CONTROL_REG。
公共时钟/复位信号	新增端口。
GT_WIZ_CONTROL_REG: 0154	添加位 3 到位 22。
<b>2021 年 10 月 27 日 4.0 版</b>	
常规更新	更新 Versal® GTM 支持。
仿真加速	新增仿真器。
RX 路径控制信号、状态信号和统计数据信号	更新表格。
MODE_REG: 0008	更新表格。
<b>2021 年 8 月 2 日 4.0 版</b>	
定时器寄存器映射	新增寄存器。
端口定时器寄存器	
“GT Selection and Configuration” 选项卡	添加新的 GUI 选项 “Enable GT Interface for Board Based Design”。
IEEE 1588 TX/RX 接口控制信号、状态信号和统计数据信号	新增端口 tx_period_ns 和 rx_period_ns。

章节	修订综述
<b>2020 年 12 月 16 日 3.3 版</b>	
PTP 1588 Timer Syncer 块	新增设计示例。
自定义和生成子系统	更新图示
LogiCORE 设计示例时钟设置和复位	更新单核详细图示 (Versal)
第 4 章：利用子系统进行设计	端口描述
<b>2020 年 9 月 1 日 3.2 版</b>	
不适用	更新 Versal ACAP 支持
<b>2020 年 7 月 14 日 3.2 版</b>	
不适用	对格式稍作更改。
<b>2020 年 6 月 3 日 3.2 版</b>	
RESET_REG: 0004	添加位 28
MODE_REG: 0008	添加位 0 和 1
STAT_RX_STATUS_REG1: 0404	添加位 0
STAT_GT_WIZ_REG: 04A0	新增寄存器
LogiCORE 设计示例时钟设置和复位	更新图示
使用 AXI4-Lite 接口进行自动协商和链路训练的开发板测试步骤	澄清步骤
自定义和生成子系统	更新 Vivado IDE
仿真加速	新增启用 RS-FEC 的配置仿真
收发器核和状态调试端口	gt_ch_* changed to gt_*
调试自动协商和链路训练	新增调试主题
<b>2019 年 10 月 30 日 3.1 版</b>	
	新增 PCS/PMA 32 位时钟设置。 新增启用 AXI4-Lite 情况下的 ANLT 开发板测试步骤。
<b>2019 年 5 月 22 日 3.0 版</b>	
	新增 AXI-4 统计数据计数器启用支持。 新增 GT Rx 接收器选项。
<b>2018 年 12 月 5 日 2.5 版</b>	
	更新 TSN 功能的许可表。 更新 ctl_rx_min_pakcet_length 的描述。 新增运行时切换配置的注释。
<b>2018 年 6 月 6 日 2.4 版</b>	
	更新 10G TSN 许可证密钥信息。 在“设计示例”章节中新增 stat_reg_compare 端口。 重命名 MAC+PCS/PMA 32 位 1588 寄存器。 为表 3-4 新增表格注释。

章节	修订综述
2018 年 4 月 4 日 2.4 版	
	<p>在“IP 相关信息”的“功能特性”部分中新增时间敏感网络(TSN)这一付费功能。</p> <p>在表 1-2 中新增 25G/10G Ethernet MAC+PCS/PMA 描述。</p> <p>在第 1 章“25G 支持的功能特性”和“10G 支持的功能特性”部分中添加对 802.1cm 抢占(preemption)功能的支持。</p> <p>更新第 1 章中的许可信息。</p> <p>在表 2-12 中添加有关 <code>ctl_local_loopback</code> 描述的注释。</p> <p>在表 2-96 中添加有关 <code>ctl_autoneg_bypass</code> 信号的注释。</p> <p>在表 2-110 中针对 <code>stat_tx_bad_parity</code> 信号添加位 7。</p> <p>在表 2-128 中为 <code>runtime_switchable</code> 信号的位 1 添加注释。</p> <p>在第 3 章的“入口”部分中添加有关 1588 时间戳精度的选项。</p> <p>在第 3 章中新增“以太网数据路径奇偶校验”章节和 802.1cm 抢占功能特性章节。</p> <p>更新第 3 章中“以太网数据路径奇偶校验”的第二段中的描述。</p> <p>更新图 4-1、4-2、4-3 和 4-4</p> <p>在表 4-1 中添加“Enable Preemption 802.3br”选项。</p> <p>在表 4-2 中添加“Enable Datapath Parity”选项和“Enable Packet Assembly FIFO”选项。</p> <p>在表 3-1 中更新有关默认频率的注释。</p> <p>在第 4 章的“必要约束”部分中新增内容。</p> <p>在第 5 章的“AXI4-Lite 接口实现”部分中新增 .h 头文件小节。</p> <p>在第 6 章中更新图 6-1。</p> <p>在附录 C 的“参考资料”部分中添加 Vivado Design Suite 用户指南：如何适用约束(UG903)。</p> <p>如需了解有关寄存器和端口的添加信息，请参阅附录 A 中的“从 v2.3 到 v2.4 的更改”。</p>
2017 年 12 月 20 日 2.3 版	
常规更新	<p>更新第 108 条 RS-FEC 以支持表 1-1 中的下列条目：运行时可切换 10G/25G MAC+PCS 和运行时可切换 10/25G 仅限 PCS。</p> <p>在表 3-1 中，将第一列的列名从“本地故障指示”更改为“名称”。</p> <p>如需了解有关端口和寄存器的添加、移除和更改信息，请参阅附录 A 中的“从 v2.3 (10/04/2017) 到 v2.3 (12/20/2017) 的更改”。</p> <p>更新图 3-2。</p> <p>在 <code>refclk_p0</code>、<code>refclk_n0</code> 和 <code>tx_serdess_refclk</code> 描述中添加对应 32 位 10G 的 312.5 MHz。</p> <p>在 2-13、2-37 和 5-2 中为下列信号添加有关无效前导码的注释：<code>stat_rx_bad_sfd</code>、<code>stat_rx_bad_preamble</code>、<code>stat_rx_bad_preamble_*</code> 和 <code>stat_rx_bad_sfd_*</code>。</p>

章节	修订综述
<b>2017 年 10 月 4 日 2.3 版</b>	
	<p>在表 1-1 中添加有关自动协商的注释。</p> <p>在第 2 章的“不间断连续传输”部分中添加有关客户端数据逻辑的文本。</p> <p>在第 3 章的“性能”部分中移除表 3-2 和支持文本。</p> <p>在第 2 章的“统计数据计数器”部分中添加有关可读 STAT_*_MSB/LSB 寄存器的注释。</p> <p>在表 2-4 和表 2-7 中添加“时钟域”列，并更新表 2-11 中的时钟域值。</p> <p>在第 2 章的“状态寄存器”部分开头处添加有关运行时切换模式的文本。</p> <p>在第 3 章的“出口”部分中修改普通时钟和透明时钟数值。</p> <p>在表 3-1 中为 tx_ptp_tstamp_out[80-1:0] 和 rx_ptp_tstamp_out[80-1:0] 添加双步 1588 操作相关文本。</p> <p>更新法律声明和关于与汽车相关用途的免责声明。</p> <p>更新第 4 章中的截屏。</p> <p>在第 4 章的“仿真加速”部分中添加有关 SIM_SPEED_UP 的文本。</p> <p>对于表 5-2 中的 axi_ctl_core_mode_switch_*, 将其值更改为 0x0138</p> <p>添加、更改并移除部分端口和寄存器。欲知详情，请参阅“从 v2.2 到 v2.3 的更改”。</p>
<b>2017 年 6 月 7 日 2.2 版</b>	
	<p>更新“订购信息”部分。</p> <p>更新“IP 相关信息”表中的“支持的用户接口”行和“支持的软件驱动程序”行。</p> <p>请参阅“从 v2.1 到 v2.2 的更改”，以了解新变体、功能特性更新以及端口添加和更新的相关信息。</p> <p>更新表 1-1、表 1-2 和表 2-1。</p> <p>更新第 4 章中的截屏。</p> <p>从表 4-1 中移除“AN/LT 时钟”选项。</p>

章节	修订综述
<b>2017 年 4 月 5 日 2.1 版</b>	
	<p>将“IP 相关信息”表中的“支持的用户接口”更改为 AXI4-Stream。</p> <p>更新功能特性列表，添加新的功能特性。</p> <p>在第 1 章中添加“10G/25G 运行时可切换 IP 功能特性”部分和“特性兼容性矩阵”表。</p> <p>在第 1 章的“许可和订购信息”中添加“订购信息”部分。在新增的表 1-2 中为独立 64 位 MAC 添加新的许可证密钥。</p> <p>将图 2-1 标题更改为 25 Gb/s 核模块框图。</p> <p>新增“图 2-2：10 Gb/s 核模块框图”和“图 2-4：适用于 10 Gb/s 操作的 64 位独立版本 MAC”。</p> <p>在全文各小节的 AXI4-Stream 接口标题中新增 32 位信息，并新增适用于 32 位操作的时序图。</p> <p>新增对应于 64 位 10G MAC 服务产品的章节。</p> <p>更新 64 位 MAC+PCS 变体，以包含全新的 32 位低时延 10G MAC + PCS 变体。</p> <p>将“端口描述”名称更改为“端口描述 - MAC+PCS 变体”</p> <p>新增“端口描述 - 10G Ethernet MAC (64 位) 变体”章节及其各小节。</p> <p>在表 2-5 中新增对应时延的行。</p> <p>更新表 2-39 到表 2-41 中的大部分注释。</p> <p>在第 3 章中添加“低时延 32 位 10 Gb/s MAC (含 PCS)”和“仅限 10G MAC 时钟设置”部分。</p> <p>更新图 3-7 和 3-8 中的 IBUFDS_GTE3。</p> <p>移除图 3-6 和图 3-8 (同步时钟模式)</p> <p>在表 4-1 中更新“选择核”和“时钟设置”选项。</p> <p>更新图 4-1 到图 4-4。</p> <p>在所有实例中将“Ethernet MAC+PCS/PMA”替换为“Ethernet MAC+PCS/PMA-32 位/64 位”，并在表 5-2 中插入“或 Ethernet MAC”。</p> <p>更新移植指南以包含 AXI4-Stream 接口</p> <p>已通篇将 LGMII 更改为 XGMII/25GMII。</p> <p>已通篇将 XXVGMI 更改为 25GMII</p> <p>如需了解端口和寄存器更改，请参阅附录 A “移植和更新”。</p>
<b>2016 年 11 月 30 日 2.0 版</b>	
	<p>在第 2 章的“配置寄存器映射”部分中，已将“tx_reset 和 rx_reset”更改为“s_axi_aresetn”，将“高电平有效”更改为“低电平有效”。</p> <p>在第 2 章的“状态寄存器映射”部分的第一段中添加有关清除状态寄存器的文本。</p> <p>在第 2 章的“统计数据寄存器”部分的第一段中添加有关清除统计数据计数器的文本。</p> <p>为表 4-1 中的“配置选项”更新表格注释 3 和 4 并添加表格注释 5。</p> <p>在第 5 章的“概述”部分中添加有关自动协商/链路训练的注释。</p> <p>在表 5-2 核 xci 顶层端口列表中添加许多端口，并删除许多端口。欲知详情，请参阅“移植和更新”附录。</p> <p>在表 2 核 xci 顶层端口列表的所有描述中移除文本“GT 选择和”。</p>

章节	修订综述
<b>2016 年 10 月 5 日 2.0 版</b>	
	<p>在附录 A “移植和更新” 中添加有关从传统 10G EMAC 移植的信息。</p> <p>通篇添加对 tick_reg_mode_sel 的引用。</p> <p>更新下列图示：3-6、3-7、3-8、3-9、4-1、4-2、4-3 和 4-4</p> <p>在表 3-1 中更新 tx_ptp_1588op_in[1:0] 和 rx_ptp_tstamp_out[80:1:0] 的描述。</p> <p>添加单步操作支持。</p> <p>在表 3-3 中更新多项端口描述。</p> <p>在表 4-1 的 “Select Core” 选项中添加 “Ethernet MAC” 值。</p> <p>在表 4-1 中将 “Include FEC Logic” 选项替换为 “Clause 74 (BASE-KR FEC)” 。</p> <p>在表 4-1 中添加 “Clause 108 (RS-FEC)” 选项。</p> <p>在第 4 章的 “仿真” 部分中添加新的 “仿真加速” 小节。</p> <p>在表 5-2 中更新 ctl_rx_rate_10g_25gn_* 的描述。</p> <p>在表 5-2 中添加 rx_ptp_tstamp_valid_out_*。</p> <p>在第 5 章的 “AXI4-Lite 接口实现” 部分下添加多个小节。</p> <p>在 “调试” 附录的 “仿真缓慢” 部分中添加步骤 5。</p> <p>在 “调试” 附录的 “时钟设置和复位” 部分中添加有关 GTRXRESET 的新段落。</p> <p>在第 5 章中添加 “核 xci 顶层端口列表” 部分。</p> <p>更新对 IEEE 2015 的引用（替代 2012）。</p>
<b>2016 年 6 月 8 日 1.3 版</b>	
	<p>通篇将 10 Gb/s 更改为 10.3125 Gb/s</p> <p>更新图示：2-2、3-16、4-1、4-2、4-3、4-4、5-1、5-2、5-3、5-4 和 5-5</p> <p>通篇添加 XGMII 到 XVGMII。</p> <p>通篇将 XXVMII 更改为 XVGMI。</p> <p>通篇添加有关支持 MAC+PCS 的地址的注释。</p> <p>在表 2-24 中添加十六进制地址和链路。</p> <p>在表 2-25、2-28 和 2-86 中添加多个位元</p> <p>为 STAT_TX_RSFEC_STATUS_REG: 044C、STAT_RX_ERROR LSB: 0668、STAT_RX_ERROR MSB: 066C、STAT_RX_RSFEC_ERR_COUNT0_INC_LSB: 0680 和 STAT_RX_RSFEC_ERR_COUNT0_INC_MSB: 0684 添加新的寄存器表</p> <p>移除第 3 章中的 “通用设计指南” 部分。</p> <p>在表 3-1 中添加 tx_ptp_rxtstamp_in</p> <p>通篇已将 “HSEC” 更改为 “10G/25G High Speed Ethernet Subsystem” 。</p> <p>在表 4-1 中添加 “控制和统计数据接口” 部分</p> <p>在表 4-3 中添加 “GT 位置” 部分，并更新 “其它” 部分中的选项</p>

章节	修订综述
<b>2016 年 6 月 8 日 1.3 版</b>	
	<p>在表 4-3 “GT 时钟选项” 中更新部分描述 在第 5 章 “设计示例” 中更新概述 更新可选模块的描述。 在第 5 章中添加 “设计示例层级（设计示例中包含 GT）” 、 “运行时可切换” 和 “IEEE 第 108 条 (RS-FEC) 集成” 部分。 完全修改第 5 章的 “共享逻辑实现” 部分。 添加共享逻辑封装器中所含模块的描述。 完全修改附录 B “调试” 中的 “仿真调试” 章节。 通篇将 802.3-2012 更改为 802.3-2015。 通篇添加单步操作。 在表 3-3 中更新多项端口描述。 在第 4 章中新增 “仿真加速” 部分。 在表 5-2 中新增端口 rx_ptp_tstamp_valid_out_*。 在 “调试” 附录的 “仿真缓慢” 部分中添加步骤 5。 在 “调试” 附录的 “时钟设置和复位” 部分中添加有关 GTRXRESET 的段落。</p>
<b>2016 年 4 月 6 日 1.2 版</b>	
	<p>添加 UltraScale+ 支持。 新增包含 RSFEC、1588 单步支持和双步支持的章节。 新增 “IEEE 1588 时间戳” 章节。 为 2 个 AXI4-Stream 接口添加 rx_preambleout [55:0]。 为 AXI4-Stream 接口添加 tx_preamblein [55:0]。 在配置、状态和计数器寄存器映射中添加寄存器。 将定制前导码从频带内更改为频带外。 在 “统计数据计数器” 章节中添加有关 pm_tick 和 TIC_REG 的文本 更改 tx_axis_tuser 和 rx_axis_tuser 信号的极性。 更新图 3-13 和图 3-14。 从表 4-2 中移除 “VLane 调整模式” 。 移除 LBUS 资料。 在表 C-4 中，添加 ctl_tx_ipg_value[3:0]。</p>
<b>2015 年 12 月 2 日 1.1 版</b>	
	更新性能和资源使用情况数据链接。
<b>2015 年 11 月 18 日 1.1 版</b>	
	<p>添加指向网上的性能和资源使用情况数据的链接。 添加 stat_rx_valid_ctrl_code、 ctl_tx_custom_preamble_enable 和 ctl_rx_custom_preamble_enable 信号。 更新 tx_axis_tuser 信号描述。 在 “AXI4-Stream 发射接口” 章节中更新 “正常发射” 和 “中止发射” 信息。 在 “设计流程步骤” 章节中添加 Vivado IDE 选项详细信息。 在 “设计示例” 章节中新增信息。</p>
<b>2015 年 9 月 30 日 1.0 版</b>	
常规更新	初始版本。

# 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时参考。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>。

## 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

© 2015-2022 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。“AMBA”、“AMBA Designer”、“Arm”、“ARM1176JZ-S”、“CoreSight”、“Cortex”、“PrimeCell”、“Mali” 和 “MPCore” 为 Arm Limited 在欧盟及其它国家或地区的注册商标。所有其它商标均为各自所有方所属财产。