

《UltraFast 设计方法 快捷参考指南》(UG1231)

简介

AMD UltraFast 设计方法是一套推荐的最佳实践，旨在最大限度提升生产力，并减少复杂系统的设计迭代，面向系统包括嵌入式处理器子系统、模拟与数字化处理、高速连接、以及网络处理。欲知详情，请参阅《适用于 FPGA 和 SoC 的 UltraFast 设计方法指南》(UG949)。
《UltraFast 设计方法检查表》(XTP301) 包含了所有常见问题，它侧重于设计决策对下游产生的方方面面影响，介绍了通常容易忽略或未知的潜在问题，并能够帮助您轻松访问相关附属资料。该检查表已经包含在 AMD Documentation Navigator 工具 (DocNav) 内。

本快捷参考指南重点介绍了关键的设计方法步骤，有助于更快达成系统集成与设计实现，从而利用 AMD 器件与工具最大限度创造价值，同时也提供了相关附属资料的链接。本指南中涵盖的主要设计任务包括：

- 开发板和器件规划
- 设计输入和设计实现
- 顶层设计确认
- 设计分析
- 设计收敛

请参阅 AMD Documentation Navigator 工具 (DocNav) 中提供的《UltraFast 设计方法：系统级设计流程》，以获取指向所有设计中心和特定附属资料的链接。



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。

UG1231 (v2023.2) 2023 年 11 月 29 日

开发板和器件规划

PCB 设计师

检查关键接口

- 确认器件朝向与关键接口

检查 PCB 布局

- 按存储器接口与收发器检查表执行检查
- 遵循 PCB 布局建议，包括内部热处理解决方案、布局和安装
- 确保最终 FPGA 管脚分配由 FPGA 设计人员验收

审查板级原理图

- 完成 PCB 检查表审核
- 确认 PDS、去耦电容器和配置连接
- 确认配置前、配置中和配置后的 I/O 状态

制造并测试

- 使用测试 I/O 工程验证配置序列、电源和 I/O 性能

另请参阅：
[UG949：开发板和器件规划](#)
[PCB 设计检查表](#)
[存储器接口 IP 设计检查表](#)
[板级原理图设计检查表](#)
(7 系列、AMD UltraScale™ 和 AMD UltraScale+™ 器件)

FPGA/SoC 设计师

分析器件的管脚分配

- 检查收发器与绑定的 I/O 位置
- 检查 SSI 技术 I/O 管脚分配
- 确认器件朝向与关键接口

定义关键接口的 I/O 管脚

- 创建 I/O 管脚分配工程
- 定义并确认存储器控制器、GT 以及 PCIe® 技术位置
- 确立时钟骨架
- 最大限度缩短已连接的 IP 间的布局规划距离

定义最终管脚分配

- 将接口工程合并为最终 I/O 工程
- 确认 DRC 和 SSN
- 实现设计以检查时钟与 I/O 规则
- 采用最终 I/O 工程开展量产测试

估算功耗

- 使用 XPE 确定功耗预算，并使用 DELPHI 模型执行热处理解决方案
- 利用此前设计知识应用翻转率

另请参阅：
[UG949：开发板和器件规划](#)
[“功耗估算与最优化”设计中心](#)
[“I/O 管脚分配和时钟规划”设计中心](#)

设计输入和设计实现

逻辑设计师

定义理想的设计层级

- 定义相关层级，帮助开展全局布局与布局规划
- 在顶层附近插入 I/O 与时钟组件
- 在主要层级边界处添加寄存器
- 生成 IP 并审核目标器件利用率

构建并确认 RTL 子模块

- 确保设计符合 RTL 编码准则
- 在 DSP 与存储器周围添加足够的寄存器
- 只在必要时使用控制信号
- 使用综合属性控制最终逻辑映射
- 创建简单时序约束，审核估算时序与含过多逻辑层的地址路径
- 审核综合 log 日志文件、利用率报告及细化视图，找出次优映射
- 运行方法论与 RTL 检查，复查问题
- 在非关联 (OOC) 模式下实现子模块，将时钟周期收紧 5% 到 10% 并确认实现的性能
- 对比原预算审核利用率与功耗
- 开展设计仿真，确认功能性

汇编并确认顶层设计

- 综合顶层 RTL 设计并解决所有连接问题
- 复查顶层利用率与时钟设置准则
- 创建并确认顶层约束
- 迭代 RTL 与约束，解决方法论和 DRC 问题，满足时序
- 下一步即执行实现

另请参阅：
[UG949：使用 RTL 创建设计和设计实现](#)
[“采用 IP 进行设计”设计中心](#)
[“使用 IP integrator”设计中心](#)
[“逻辑综合”设计中心](#)
[“应用设计约束”设计中心](#)
[“实现”设计中心](#)

顶层约束确认

设计基线设定

- 在设计进程早期，大多数块与关键 IP 可用的情况下，确认时序收敛可行性
- 仅在以下情况下指定必要约束：
 - 使用所有 IP 约束
 - 定义符合现实的基准时钟与生成时钟
 - 定义所有时钟域交汇约束
 - 按需添加多周期路径
 - 在此阶段请勿使用 I/O 约束
- 确保路径要求的合理性
- 在流程中下列每个阶段使用 report_timing_summary 确认 WNS \approx 0.0 ns
 - 综合后
 - 布局前
 - 布线前后
- 在流程早期解决时序违例
- 在 RTL 和综合中修复 QoR 问题，实现最大成效
- 执行 report_qor_assessment 以进行确认

确认时序约束

- 运行 report_timing_summary 或 check_timing 确保所有时钟均已定义，并且所有寄存器、输入端口和输出端口均已得到约束
- 运行 report_methodology 并解决所有 TIMING* 和 XDC* 问题
- 运行 report_clock_interaction 以确保每个时钟对都采用合理的路径要求以安全方式达成了时序约束
- 运行 report_cdc 以验证所有异步时钟域交汇路径都已得到妥善约束，并使用安全的同步电路
- 运行 report_exceptions 以识别哪些时序例外存在重叠、被忽略或者低效的现象
- 确保加载设计并应用约束时，所有 Critical Warning 都已得到解决。

另请参阅：

[UG949：设计收敛](#)

- 检查设计是否正确约束

- 设计基线设定

“应用设计约束”设计中心

“时序收敛和设计分析”设计中心

设计分析和收敛

识别时序违例的根源

- 尝试运行 Intelligent Design Run 或 report_qor_suggestions 以执行自动分析并获取时序收敛建议
- 使用 report_design_analysis 查找根源
- 对于建立时间路径，检查下列原因造成的高数据路径延迟：
 - 大型单元延迟（7 系列 > 25%，UltraScale 器件 > 50%）
 - 大型信号线延迟（7 系列 > 75%，UltraScale 器件 > 50%）
- 对于保持时间路径，检查是否存在保持时间要求 > 0 ns 的问题
- 检查是否存在时钟偏差过高 (> 500 ps) 和/或时钟不确定性过高 (> 200 ps)

减少逻辑延迟

- 修改 RTL，使用并行运算符或高效运算符
- 添加流水线寄存器并使用综合重定时
- 在 DSP 或块 RAM 输出上添加寄存器
- 使用 LUT_REMAP 将最长路径中的小型 LUT 折叠
- 在 SRL 输入和/或输出上将寄存器从 SRL 拉出
- 移除 KEEP/DONT_TOUCH/MARK_DEBUG

减少信号线延迟

- 审核并调整布局规划约束
- 最优化高扇出信号线
- 查阅下列报告，如果拥塞等级 > 4，则需解决拥塞问题：
 - report_design_analysis 布局器拥塞表
 - log 日志文件中初始估算的布线器拥塞

减少时钟偏差

- 使用并行缓冲器替代级联缓冲器
- 在源于相同输入或 PLL 的同步时钟间使用 CLOCK_DELAY_GROUP
- 在异步时钟间添加时序例外

减少时钟不确定性

- 最优化 MMCM 设置
- 在 UltraScale 器件中使用 BUFGCE_DIV 进行时钟分频

另请参阅：

[UG949：设计收敛](#)

- 理解时序报告

- 识别时序违例的根源

“时序收敛和设计分析”设计中心

减少控制集

- 在控制信号中避免出现 MAX_FANOUT
- 提高综合控制集阈值
- 使用 opt_design 合并等效的控制信号

最优化高扇出信号线

- 在 RTL 中使用基于层级的寄存器复制
- 使用 opt_design -merge_equivalent_drivers -hier_fanout_limit 提升复制效率
- 将非关键高扇出信号线提升为全局时钟布线
- 使用 FORCE_MAX_FANOUT 属性强制复制

解决拥塞

- 降低器件利用率，并平衡 SLR 利用率
- 尝试运行布局器指令 AltSpreadLogic* 或 SSI_Spread*
- 使用以下命令识别拥塞的模块：report_design_analysis -complexity -congestion
- 对于拥塞的模块，使用 AlternateRoutability 块级综合策略，通过 opt_design 减少 MUXF*/CARRY*，或者使用 CELL_BLOAT_FACTOR 属性
- 针对拥塞区域内的高扇出信号线使用全局时钟设置
- 复用先前低拥塞实现中的 DSP 和块 RAM 布局

调节编译流程

- 尝试运行多项 place_design 指令，或者运行以下命令使用基于 ML 的策略：report_qor_suggestions
- 使用块级综合策略以获取最优网表
- 在布局和物理最优化期间使用以下命令对关键时钟进行过约束：set_clock_uncertainty
- 完成小幅设计修改后使用增量编译

分析并最优化功耗

- 对活动、环境和进程进行约束
- 尝试运行 power_opt 以降低功耗
- 最大程度提高块 RAM 级联的利用率

另请参阅：

[UG949：设计实现和设计收敛](#)

- 分析并解决时序违例

- 运用常用时序收敛技巧

“实现”设计中心

“时序收敛和设计分析”设计中心