Vivado Design Suite 用户指南

采用 IP 进行设计

UG896 (v2020.1) 2020 年 7 月 13 日

条款中英文版本如有歧义,概以英文文本为准。





修订历史

下表列出了本文档的修订历史。

章节	修订综述
2020年7月	13 日 v2020.1
常规更新	编辑更新。



目录

第1章:围绕 IP 的设计流程

引言	5
IP 术语	7
IP 封装器	7
IP integrator	8
使用版本控制和源代码控制	8
使用加密	8

第2章: IP 基础知识

引言	9
使用 IP 工程设置	9
使用 IP 目录	15
创建 IP 自定义	22
例化 IP	29
了解工程中的 IP 状态	30
管理 IP 约束	31
设置目标时钟周期	34
IP 的综合选项	36
仿真 IP	38
升级 IP	43
了解多层 IP	46
处理调试 IP	48
使用核容器	49

第 3章:使用"管理 IP"工程

引言	
托管 IP 功能	
使用管理 IP 流程	

第4章:使用 IP 设计示例

引言		58
打开设计示例		58
检验独立 IP	•	59

第 5章:将赛灵思 IP 与第三方综合工具配合使用

引言	61
第三方综合流程	61

第 6章:用于常见 IP 操作的 Tcl 命令

引言		64
在设计流程中使用 IP Tcl 命令	۶	64

E XILINX_®

用于常见 IP 操作的 Tcl 命令 IP 流程命令示例	
附录 A: 判定 IP 锁定原因 	
 附录 B: IP 文件和目录结构 引言 IP 生成的目录和文件 与 IP 关联的文件 使用 COE 文件 	
附录 C: 对 IP 使用平台开发板流程 _{引言}	82
附录 D: 编辑或覆盖 IP 源文件 引言 覆盖 IP 约束 编辑 IP 源文件 编辑子系统 IP.	87 87 87 89 90
 附录 E: 附加资源与法律声明 赛灵思资源 解决方案中心 Documentation Navigator 与设计中心 参考资料 请阅读:重要法律提示 	92 92 92 92 92 92 92 95



第 1章

围绕 IP 的设计流程

引言

赛灵思 Vivado® Design Suite 可提供围绕 IP 的设计流程,支持您将来自各种设计的 IP 模块添加到自己的设计中。此环境的核心是可扩展的 IP 目录 (IP catalog),其中包含赛灵思提供的即插即用 IP。 IP catalog 可通过添加以下内容来加以 扩展:

- 来自 System Generator for DSP 设计的模块 (源自 Simulink[®] 算法的 MATLAB[®])。
- Vivado 高层次综合 (HLS) 设计 (C/C++ 算法)
- 第三方 IP
- 使用 Vivado IP 封装器工具封装为 IP 的设计

图 1-1 显示了围绕 IP 的设计流程。







X14070-030917

图 1-1: 围绕 IP 的设计流程

注释:在某些情况下,第三方供应商可提供 IP 作为已综合的 EDIF 网表。您可使用"Add Sources"命令将这些文件加载 到 Vivado 设计中。

在设计中使用 IP 的方法包括:

- 使用"托管 IP (Managed IP)"流程来自定义 IP 并生成输出文件 (包括综合后的设计检查点 (DCP)),这样即可保留 自定义设置以供在当前和未来版本中使用。如需了解更多信息,请参阅第 3 章"使用"管理 IP"工程"。
- 在工程模式或非工程模式下使用 IP 的方法是引用已创建的赛灵思核实例 (XCI) 文件,推荐在处理由多名团队成员 协作完成的大型工程时采用此方法。



- 从工程访问 IP 目录,以自定义 IP 并将其添加到设计中。 IP 文件可存储在工程本地,或者对于小型团队规模的工程,建议将其存储在工程外部。
- 源文件的添加方法是右键单击 IP integrator 画布并将 RTL 模块添加到设计图中,这样即可提供"画布上的 RTL (RTL on Canvas)"。请访问此链接并参阅《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》(UG994) [参照 32] 以获取有关模块参考的更多信息。
- 在非工程脚本流程中创建并自定义 IP,并生成输出文件(包括生成 DCP)。请访问此链接并参阅《Vivado Design Suite 用户指南:设计流程简介》(UG892) [参照 15] 以获取有关非工程模式的更多信息。



建议: 请始终使用 XCI 文件来引用 IP。不建议通过工程模式或非工程模式流程直接读取 IP DCP 文件。虽然早于 2017.1 的版本中 DCP 曾包含约束,但在后续 Vivado 版本中, DCP 不含约束、不提供其它输出文件 (IP 可提供此类输出文件),也无法提供其它可能需要的输出文件 (如 ELF 或 COE 文件和 Tcl 脚本)。

《Vivado Design Suite 教程:采用 IP 进行设计》(UG939) [参照 29] 提供了有关如何在 Vivado 中使用赛灵思 IP 的指示信息。



培训: 赛灵思可提供培训课程,以帮助用户进一步了解有关本文档中提出的概念。请使用下列链接来浏览相关课程: FPGA 设计要领和嵌入式系统软件设计。

IP 术语

Vivado IDE 使用下列术语来描述 IP、其存储位置及其表述方式。

- IP 定义: 有关 IP 的 IP-XACT 特性的描述。
- IP 自定义:根据 IP 定义来自定义 IP,从而生成 XCI 文件。XCI 文件用于存储用户指定的配置。
- IP 位置: 包含当前工程中的一项或多项自定义 IP 的目录。
- IP 存储库:包含添加到赛灵思 IP 目录中的 IP 定义集合的统一视图。
- IP 目录 (IP catalog): IP catalog 支持浏览赛灵思即插即用型 IP 以及第三方供应商提供的其它 IP-XACT 兼容的 IP。 也包括您封装为 IP 的设计。如需了解更多信息,请参阅第 2 章"IP 基础知识"。
- 输出文件:根据 IP 自定义生成的文件。包括 HDL、约束和仿真目标。输出文件生成期间, Vivado 工具会将 IP 自定义存储在 XCI 文件中,并使用此 XCI 文件来生成可供在综合与仿真期间使用的文件。
- 全局综合:用于将 IP 随顶层用户逻辑一起进行综合。
- **非关联 (OOC) 设计流程:** OOC 设计流程用于为生成的输出文件创建独立的综合设计运行。此默认流程会创建设计 检查点文件 (DCP) 和赛灵思设计约束文件 (_ooc.xdc)。如需了解更多信息,请参阅"非关联流程"。
- 层级 IP 和子系统 IP: 这些术语可互换使用, 且均用于描述 IP, 表示在块设计或 RTL 流程中以包含多个 IP 的层级 拓扑结构方式所构建的子系统。
- 子核 IP: 术语"子核 IP"表示在不属于层级(子系统) IP 的另一个 IP 中所使用的 IP。此 IP 可源于 Vivado IP catalog、用户定义的 IP、第三方 IP 或 IP 核库。

IP 封装器

Vivado IP 封装器支持您创建即插即用型 IP,以添加到可扩展的 Vivado IP catalog。IP 封装器向导基于对应 IP-XACT 的 IEEE 标准(IEEE 1685 标准)《在工具流程中封装、集成和复用 IP 的标准结构》 [参照 1]。



装配完成 Vivado Design Suite 用户设计后, IP 封装器支持您将此设计转变为可复用的 IP 模块,以便您将其添加到 Vivado IP catalog 中,并可供他人用于设计工作。您可在基于工程模式的设计或基于非工程模式的设计中使用封装的 IP。如需了解更多信息,请参阅下列参考文档:

- 《Vivado Design Suite 用户指南: 创建和封装定制 IP》 (UG1118) [参照 34], 以获取有关使用封装功能的更多信息。
- 《Vivado Design Suite 教程: 创建和封装定制 IP》 (UG1119) [参照 35],其中提供了含设计解决方案的实践,并为 您演示了如何使用封装功能。

IP integrator

Vivado Design Suite IP integrator 工具支持您通过在设计画布上将来自 Vivado IP catalog 的 IP 核与模块参考加以例化 并互连,从而创建复杂的子系统设计。如需了解更多信息,请参阅《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》(UG994) [参照 32]。

使用版本控制和源代码控制

Vivado Design Suite 按设计可配合任何版本控制系统一起使用。对于 IP 设计,您在使用版本控制系统来管理设计源代码时需要考量一些利弊取舍。这些利弊取舍主要影响运行时间与所管理的文件数量。如需了解有关如何将 Vivado Design Suite 与版本和源代码控制系统配合使用的信息,请访问此链接并参阅《Vivado Design Suite 用户指南:设计流程简介》 (UG892) [参照 15]。

使用加密

赛灵思根据《IEEE 建议的电子设计知识产权 (IP) 加密与管理实践》(IEEE P1735 标准) [参照 2] 对 IP HDL 文件进行加密。



视频: 请参阅 《Vivado Design Suite QuickTake 视频: 在 Vivado Design Suite 中使用 IP 加密工具》,以获取有关加密 的更多信息。

此外,请访问此链接以参阅《Vivado Design Suite 用户指南: 创建和封装 IP》 (UG1118) [参照 34]。

china.xilinx.com



第2章

IP 基础知识

引言

本章描述了采用 IP 进行设计的功能特性。

注释:对于含蓝色链接的 Tcl 命令,您可直接访问《Vivado Design Suite Tcl 命令参考指南》 (UG835) [参照 13] 以获取 有关该命令的更多信息。您还可在 Vivado IDE 中使用 <command_name> -help。

要使用赛灵思 IP, 首先需自定义 IP 以供在 RTL 设计中使用。您可使用 Vivado[®] Design Suite 通过如下所述多种方式来 创建 IP 自定义:

- 直接在工程中自定义来自 IP catalog 的 IP。
- 使用"管理 IP (Manage IP)"工程流程来创建独立的 IP 自定义,以便在当前工程和其它工程中使用。如需了解更多信息,请参阅第 3 章"使用"管理 IP"工程"。
- 在工程模式或非工程模式下使用 Tcl 脚本创建 IP 自定义。
- 添加或创建块设计 (BD)

创建自定义后,可直接生成输出文件,或者也可以延后生成输出文件。

- 在工程模式下,如果输出文件不存在,那么 Vivado 工具会在执行综合或仿真前自动生成所需的输出文件。默认情况下,以非关联 (OOC) 流程生成输出文件用于综合。如需了解更多信息,请参阅"非关联流程"。
- 在非工程模式下,您必须在执行综合或仿真前手动生成输出文件。

要在设计中使用 IP 自定义,您必须在自己的项层设计的 HDL 代码中例化 IP。 IP 输出文件具有自动生成的 VHDL 和 Verilog 例化模板。如需了解更多信息,请参阅"使用 IP 工程设置"。

0

视频:请参阅 《Vivado Design Suite QuickTake 视频: Vivado IDE 入门》,以获取有关使用 Vivado IDE 的更多信息。

使用 IP 工程设置

在"Manage IP"工程或 RTL 工程中使用 IP 时,您可使用"设置 (Settings)"对话框中的 IP 分类来配置特定于 IP 的设置。其中包含如下选项:

- "IP": 支持您指定使用的"核容器 (Core Containers)"、自动生成仿真脚本、创建升级日志、设置 IP 输出文件默认 位置以及开启 IP 高速缓存。
 - 。 "Repository": 添加 IP 存储库并指定要包含在 IP catalog 中的 IP。
 - 。 "Packager": 设置封装 IP 时, IP 封装器所使用的默认行为。如需了解更多信息,请参阅《Vivado Design Suite 用户指南: 创建和封装定制 IP》 (UG1118) [参照 34]。



注释处理 RTL 工程或使用"Getting Started"页面上的"Manage IP"选项时,可使用工程的"IP 设置 (IP Settings)" 以及 Vivado IP catalog。使用"Manage IP"工程时,您可看到可用的 IP 设置。

IP 设置

IP 设置支持您为 IP 指定各种工程专用的选项,如图 2-1 所示。对于您创建的每项工程,您必须再次访问工程设置才能更改设置。

Q~	IP >
Project Settings	Specify various settings associated to IP.
General	
Simulation	Core Containers
Elaboration	Use Core Containers for IP
Synthesis	
Implementation	Simulation
Bitstream	Use Precompiled IP simulation libraries
/ IF	Automatically generate simulation scripts for IP
Tool Settings	
Project	Upgrade IP
IP Defaults	Generate log file
Source File	
WebTalk	Default IP Location
Help	Location that IP added to the project will have output products
> Text Editor	and customization stored.
3rd Party Simulators	IP location: 6 <local project="" to=""></local>
> Colors	IP Cache
Selection Rules Shortcuts	Out of Context per IP Synthesis needs to be used to take advantage of IP Caching.
> Strategies	Cache scope Local ~
> Window Behavior	Cache location: C:/Temp/project 4/project 4 cache/in
	Clear Cache
~	

图 2-1: 一般 IP 设置对话框

IP 设置包含以下选项:

- "Core Containers": 勾选"Use Core Containers for IP"即可使用核容器功能,该功能支持您将 IP 和所有生成的 输出文件都包含在单一压缩二进制文件中,该文件扩展名为 XCIX。如需了解更多信息,请参阅"使用核容器"。
- "Simulation":默认已勾选"Use Precompiled IP simulation libraries"选项和"Automatically generate simulation scripts for IP"选项的复选框。

Vivado 可为所有赛灵思 IP 静态文件提供预编译库,以便配合 Vivado 仿真器一起使用。创建仿真脚本时,这些脚本 会引用这些预编译库。如果您使用第三方仿真器,则必须根据《Vivado Design Suite 用户指南:逻辑仿真》(UG900) [参照 20] 中所述方式创建这些库。

"Automatically Generate Simulation Scripts for IP"选项可为每个 IP 自动生成仿真脚本。Vivado 工具将脚本置于 <project name>.ip_user_files 目录中。如需了解更多信息,请参阅"仿真 IP"。要禁用仿真脚本创建功能,请取消勾选该选项。

- "Upgrade IP":默认已勾选"Generate log file"。这样在升级 IP 时会创建 ip_upgrade.log 文件。升级其它 IP 时,会将其添加到此日志文件顶部。此日志文件存储在工程目录的根目录下 (即工程 XPR 文件所在位置)。如需 了解更多信息,请参阅"升级 IP",其中还包含有关如何使用 Tcl 命令指定日志文件名称和位置的信息。要禁用日志 文件创建功能,请取消选中"Upgrade IP"复选框。
- "Default IP Location":您可使用该选项来设置创建和存储 IP 和输出文件的位置。默认情况下,Vivado 工具将 IP 存储在工程目录结构的.src/sources_1/ip 目录下的 RTL 工程中。
 - 。 使用版本控制系统时,建议您将自己的 IP 存储在工程外部,就像其它源文件一样。

注释: 自定义 IP 时, "IP Location"可用于设置 IP 及其输出文件的存储位置。如设置默认 IP 位置,则此设置将在 多个 Vivado 会话间保持不变。

- "IP Cache": 支持您定义 Vivado 为工程使用 IP 高速缓存的方式。高速缓存选项包括:
 - 。 "Cache scope":选项包括"disabled"、"local"(默认设置)或"remote"。对于"local",高速缓存目录为工程本地目录,并且此位置无法更改。"remote"选项对应您指定的目录。
 - 。 "Cache location": 浏览并选择高速缓存目录。对于"local",高速缓存目录为工程本地目录 (project_name.ip_cache)并且此位置无法更改。对于"remote"选项,它对应您指定的目录。
 - 。 "Cache name": 一连串唯一的随机字符和数字, 用于引用生成的 IP。
 - "Clear Cache":在 Tcl 控制台 (Tcl console) 上发出以下 config_ip_cache 命令以从磁盘删除高速缓存文件: config_ip_cache -clear_output_repo

如需获取 IP 高速缓存的详细描述,请参阅"设置 IP 高速缓存"。



重要提示:非工程流程不支持 IP 高速缓存。

设置 IP 高速缓存

为了在工程模式下使用默认 OOC 流程加速生成 IP 的综合输出文件,默认启用"IP Cache"选项。启用高速缓存后,使用 默认 OOC 流程生成 IP 以进行综合时,Vivado 工具会创建综合输出文件 (例如, DCP 和存根文件)以及高速缓存条 目。

高速缓存条目由磁盘上的2个目录组成:

- <cache ID>: 包含 XCI、 DCP、 sim_netlist 和存根文件。
- <cache ID>.logs: 包含综合日志 (runme.log)。

填充高速缓存后,当您使用相同属性创建新的自定义 IP 时,生成期间不会对此 IP 再次进行综合。而是改为由 Vivado 引用高速缓存并复制综合输出。此 IP 会引用已综合的 IP,因此对于另一个采用相同自定义的 IP,无需再次运行 OOC 综合。

启用高速缓存后生成 IP 时, Vivado 工具会像无缓存命中一样照常创建设计运行。但如果存在缓存命中,则综合结果 (DCP 和存根文件)将从高速缓存目录复制到 IP 目录并重命名。这样就不会创建设计运行,但会添加一个条目以报告 缓存命中,如下图所示。



Tcl Console Messages Log	Reports De	sign Runs ×						? _ 🗆	Ľ
Q, 素 ♠ I∢ ≪	▶ ≫ +	%							
Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	F
✓ ✓ synth_1 (active)	constrs_1	synth_design Complete!							
✓ impl_1	constrs_1	route_design Complete!	1.448	0.000	0.062	0.000	0.000	0.411	
👻 🖹 Out-of-Context Module Runs									
🗸 axi_data_fifo_0_synth_1	axi_data_fifo_0	synth_design Complete!							
🗸 axi_data_fifo_0_1		Using cached IP results							
<									\rightarrow

图 2-2:设计运行状态

生成后将会显示一个对话框以告知您是否发生缓存命中。

以下是 INFO 消息示例, Vivado 工具会在 Tcl console 中生成此类消息,并将其存储在 vivado.log 中:

INFO: [IP_Flow 19-4993] Using cached IP synthesis design for IP fifo_0, cache-ID =
aa71c47ae9ccd380; cache size = 0.383 MB.

此处显示了所使用的高速缓存 ID 以及 IP 高速缓存的当前大小。

要发生缓存命中, IP 自定义方式必须完全相同并且采用相同器件和语言设置。生成 IP 后,此 IP 并不会引用高速缓存位置。IP 的所有输出文件都存储在工程本地位置。仅在 IP 输出文件生成期间才引用高速缓存。



注意! IP 高速缓存会随存在的 IP 数量增加而不断增大。

要管理 IP 高速缓存,请使用 config_ip_cache Tcl 命令。此命令可用于列出高速缓存内容以及高速缓存大小(以 KB 为单位)。如需了解更多信息,请参阅《Vivado Design Suite Tcl 命令参考指南》(UG835) [参照 13]。

为新工程配置 IP 高速缓存

默认情况下,所有新工程均已启用"IP Cache",并配置为存储在工程本地,但对于多人/团队,可能将其设置为指向相同高速缓存位置更好。这样可以减少使用的磁盘空间,因为这样每个用户就无需重复生成与其它用户所使用的相同IP。

要将新创建的工程配置为禁用高速缓存,或者指定远程位置作为默认位置,可使用 Vivado_init.tcl 并添加参数以 控制 project.defaultIPCacheSetting。请访问此链接,并参阅《Vivado Design Suite 用户指南:使用 Tcl 脚 本》 (UG894) [参照 17] 以获取有关 Vivado_init.tcl 文件和初始化脚本的更多信息。

建议:使用共享高速缓存时,请创建 IP 高速缓存专用的目录,并指向该目录。这样将在该目录下创建 IP 高速缓存条目,并且不会创建父目录。如果设置其它远程位置并且该位置包含众多其它子目录,则会减慢工程创建速度,因为 Vivado 工具会扫描这些目录中是否包含高速缓存条目。

在 Vivado init.tcl 中设置高速缓存的示例

以下是有关如何使用 Vivado_init.tcl 文件禁用 IP 高速缓存的示例:

set_param project.defaultIPCacheSetting none

以下是有关将 IP 高速缓存设置到远程位置的示例:

Linux: set_param project.defaultIPCacheSetting /wrk/staff/smith/ip_cache/

china.xilinx.com



• Windows: set_param project.defaultIPCacheSetting c:/<project_dir>/ip_cache/



下图显示了"存储库 (Repository)"选项。

Q	IP > Repository
Project Settings General Simulation Elaboration	Add directories to the list of repositories. You may then add additional IP to a selected repository. If an IP is disabled then a tool-tip will alert you to the reason.
Synthesis	IP Repositories
Implementation	+ $ +$ $+$
Bitstream	
~ IP	•
Repository	No content
Packager	
Tool Settings	
Project	Refresh All
IP Defaults	~

图 2-3: IP 存储库选项

要更改存储库 (Repository) 设置,请转至"Flow Navigator > Settings",然后单击"IP > Repository"。下列操作可用 于管理 IP (自定义封装 IP 或第三方 IP)的位置:

- 单击"Add"按钮 以指定 IP 定义的位置。"存储库管理器 (Repository Manager)"会在 IP 定义的指定路径内逐层 搜索。指定目录后即可列出 IP。添加 IP 存储库时,会显示一个对话框,其中包含存储库中的 IP 列表。包含灰色图 标的 IP 表示已禁用的 IP,黄色图标则表示已启用的 IP。
- "Remove" 按钮可用于移除存储库列表。 IP catalog 会显示存储库已不存在。
- 您可使用上下方向键来更改存储库的搜索顺序。

要使用每个存储库中的 IP 来更新 IP catalog 中的内容,请单击"Apply",然后单击"OK"。

提示: 使用"IP Settings"对话框的"Repository Manager"的另一种方法是使用 Vivado IP catalog 中的右键单击菜单,将存储库直接添加到 IP catalog 中。



使用封装器设置

下图显示了"封装器 (Packager)"设置。

Project Settings	IP > Packager Specify settings related to IP Packager.	
General Simulation	Default Values	
Elaboration Synthesis	The followir the IP Pack	ng values will be automatically applied after finishing ager Wizard.
Implementation Bitetream	Vendor:	xilinx.com 📀
✓ IP	Library:	user 🙁
Repository	Category:	/UserIP
Packager	IP locatio <u>n</u> :	/ip_repo
Tool Settings Proiect	Automatic Beha	avior
Project IP Defaults Source File Display WebTalk Help > Text Editor 3rd Party Simulators > Colors Selection Rules Shortcuts > Strategies > Window Behavior	After Packa Create Add Close Edit IP in IP Close Edit IP in IP Edit IP in IP in IP Edit IP in	ate archive of IP IP to the IP Catalog of the current project se IP Packager window Packager ete project after packaging to Filter on Add Directory file extensions that will be automatically filtered when bry to a File Group.
		No content

图 2-4: IP 设置: 封装器

请参阅《Vivado Design Suite 用户指南:创建和封装定制 IP》 (UG1118) [参照 34] 以获取有关封装 IP 和封装器设置的 更多信息。



使用 IP 目录

Vivado IP catalog 的主要功能包括:

- 通过单一公用存储库即可连续轻松访问赛灵思 IP,包括构建块、向导、连接情况、DSP、嵌入、AXI 基础架构以及 视频 IP,与开发的终端应用无关。
- 支持在单一 IP 存储库中设置多个物理位置,包括共享网络驱动器,以便用户或企业对第三方 IP 或内部开发的 IP 应用统一的 IP 部署环境。
- 获取赛灵思提供的 IP, 此类 IP 经过严格的测试之后才会包含到 IP 目录中。
- 支持使用 Vivado IDE 或基于自动化脚本的流程 (使用 Tcl) 来使用 IP 自定义和生成功能。
- 按需交付 IP 输出文件,如例化模板和仿真模型(HDL、C或 MATLAB® 软件)。
- IP 示例设计,可提供直接评估 IP 是否适合作为 Vivado Design Suite 工程中的例化源的功能。
- 获取"更改日志 (Change Log)"中所记录的版本历史纪录详细信息。通过 <Major#.Minor#.Rev#> 编号方案可统 IP 版本号。

如需了解更多信息,请访问赛灵思网站上的"赛灵思 IP 版本"页面,并参阅《Vivado IP 版本》[参照 37]。

- 。 以下是有关 IP 版本处理方法的响应措施建议:
 - Major#: 您需要执行更改。
 - Minor#: 您可能需要执行更改。
 - Rev#: 无需执行更改。
- "目录 (Catalog)"筛选选项支持您按"受支持的输出文件 (Supported Output Products)"、"受支持的接口 (Supported Interfaces)"、"许可 (Licensing)"、"供应商 (Provider)"和"状态 (Status)"进行过滤。
- 按分类法 (Taxonomy) 或存储库 (Repository) 对 IP 进行分组。

report_property Tcl 命令可用于列出可用于 IP 定义的属性,如下所示:

report_property -all [get_ipdefs <IP VLNV>]

此信息同样显示在 IP catalog 中。以下显示了此命令的示例:

report_property -all [get_ipdefs xilinx.com:ip:fifo_generator:13.1]

您还可查询特定属性,如下所示:

get_property supported_families [get_ipdefs xilinx.com:ip:fifo_generator:13.1]

china.xilinx.com

如需获取有关支持 Vivado Design Suite 的 IP 的信息,请访问赛灵思网站并参阅《IP 文档》[参照 38]。如需获取有关特定 IP 的信息,请参阅《IP 中心》[参照 39] 或者查看 IP catalog。

下图显示了默认 Vivado IP catalog 视图,其中列出了可用的 IP 类别 (核)。



Project Summary × IP Catalog ×			
Cores Interfaces			
≍ ≑ ≇ •€ <i>⊮</i> <i>₽</i> ⊕ ■	Q,-		
Name	∧¹ AXI4	Status Li	cense VLNV
Alliance Partners			
> 🗁 Automotive & Industrial			
> 🗁 AXI Infrastructure			
> 🗁 AXIS Infrastructure			
> 🗁 BaselP			
> 🗁 Basic Elements			
> 🖹 Communication & Networking			
> 🗁 Debug & Verification			
> 🖹 Digital Signal Processing			
> 🗁 Embedded Processing			
> 📄 FPGA Features and Design			
> 🗎 HMC Host Controller			
> 🗎 Math Functions			
> 🗁 Memories & Storage Elements			
> 🗁 Partial Reconfiguration			
> 🗁 SDAccel DSA Infrastructure			
> 🗁 Standard Bus Interfaces			
> 🗁 Test NOC			
> 📮 Video & Image Processing			
> 🔄 Video Connectivity			

Details

Select an IP or Interface or Repository to see details

图 2-5: 赛灵思 IP 目录 - 核

china.xilinx.com

IP catalog 中包含 2 个视图,分别是:

- "Cores":由赛灵思、合作伙伴和客户存储库所提供的 IP。
- "Interfaces":可用接口列表,如图 2-6 所示。

Diagram × Address E	Editor × IP Catalog ×	2013
Cores Interfaces		
± \$ ⊘ €	QT	
Name	^ 1 Version ∨ 2 Vendor VLNV	External Repository
👻 🚞 Vivado Repository		
> 🚍 Advanced		
> 🚍 AXI		
> 📄 IO Interfaces		
> 🚍 Signal		
> 🗅 User		
Details		
	Select an IP or Interface or Repository to see detail	s

图 2-6: 赛灵思 IP 目录 - 接口

"接口 (Interfaces)"按 IP 封装时在 Vivado 存储库中存储内容的方式进行分类。

IP 筛选

IP catalog 包含 3 类 IP 可供您筛选和搜索。图 2-7 显示了 IP catalog 筛选选项,支持您按类别对 IP catalog 进行筛选。选择**"Settings"**按钮即可打开这些筛选选项。

第2章: IP 基础知识



Diagram × IP Catalog ×					? 🗆 🖒
Cores Interfaces					
Q ¥ ♦ ₽ ₽ ⊕					۵
Search: Q.				Q 🔮 🌲 246 of 246	с×
Name	^ 1 AXI4	Status	L	✓	^
Vivado Repository			^	C Simulation	
> 🚍 Alliance Partners				🕑 Examples	
> 🚍 Audio Connectivity & Processing			- 1	🕑 Test Bench	
> 🚍 Automotive & Industrial				✓ ✓ Supported Interfaces	
> 🚍 AXI Infrastructure			- 1	🕑 Other	
> 🚍 AXIS Infrastructure				AXI4 / AXI4-Lite	
> 🚍 BaselP				AXI4-Stream	
> 🚍 Basic Elements				V V Licensing	
> 🚍 Communication & Networking				Purchase	
> 🚍 Debug & Verification				✓ Included	
> 🚍 Digital Signal Processing				V Provider	
> 🚍 Embedded Processing			~	Viliov	
< -			>	A VIII IX	~
Details					
	Select an IP or Interfa	ce or Repository to	o see det	ails	
	serve anni or interna	to or nepository to		Sector Sec	
1					

图 2-7: IP catalog 筛选选项

取消勾选筛选选项即可过滤掉工程不需要的 IP。



下图显示了名称中包含"filter"一词的 IP 的搜索结果以及文件夹包含"filter"一词的 IP 的搜索结果。

Project Summary × IP Catalog ×				? 🗆 🖸
Cores Interfaces				
🛣 � 🌮 ▪⊑ 𝒫 @ 🔳 Q- Filter	8			ø
Name	∧ ¹ AXI4	Status	License	VLNV
✓ ☐ Alliance Partners				
~ 🖨 OmniTek				
🁎 Noise Filter	AXI4, AXI4-Stream	Production	Purchase	omnitek.tv:ip:on
👻 🚍 Digital Signal Processing				
✓ ➡ Filters				
P CIC Compiler	AXI4-Stream	Production	Included	xilinx.com:ip:cic
DUC/DDC Compiler	AXI4-Stream	Production	Included	xilinx.com:ip:du
P FIR Compiler	AXI4-Stream	Production	Included	xilinx.com:ip:fir_
✓				
Color Filter Array Interpolation	AXI4, AXI4-Stream	Production	Purchase	xilinx.com:ip:v_(
👎 Noise Filter	AXI4, AXI4-Stream	Production	Purchase	omnitek.tv:ip:on
<				
Details				

图 2-8: 按"Filter"一词执行筛选搜索

IP catalog 选项

IP catalog 选项支持您展开、折叠和搜索 IP catalog 内容。其中图标与 Vivado IDE 中的其它窗口的图标一致。

在 IP catalog 中执行搜索时,包含在与搜索模式相匹配的目录或文件夹中的 IP 都会按分类法结构显示,即使它与您的 搜索模式不完全匹配也是如此。

下表显示了表示 IP catalog 信息的符号。

表 2-1: IP 目录 IP 信息

选项	描述
择	隐藏不兼容 IP
	按使用方法分组。单击此按钮即可打开"Group by"选项,您可在其中选择"Taxonomy" 和/或"Repository"。
	Group by X
•[Taxonomy
-	Repository
	"按存储库分组 (Group by Repository)"支持您根据赛灵思 IP 或用户 IP 来对 IP 进行分组。"按分类法分组 (Group by Taxonomy)"支持您按功能或类别对 IP 进行分组。



表 2-1: IP 目录 IP 信息 (续)

选项	描述
¥	自定义 IP。打开"自定义 IP (Customize IP)"对话框以显示最近选中的 IP。请参阅"创建 IP 自定义"。
Þ	许可状态。如需了解更多信息,请参阅"使用付费许可 IP"。
٥	显示兼容的产品系列。显示与当前选中的 IP 兼容的器件系列。
0	查看产品指南、更改日志、产品网页和答复记录。
¢	设置 IP 目录、IP 生成、存储库管理器和 IP 封装器。此命令可用于打开"设置 (Settings)" 对话框以显示当前工程,并显示其 IP 设置。如需了解更多信息,请参阅"使用 IP 工程设置"。
ŧ	可向选定的合作伙伴采购的 IP。请参阅"合作伙伴 IP",以获取更多信息。

合作伙伴 IP

Vivado IP catalog 包含可向选定的合作伙伴采购的 IP。这些 IP 以蓝色磁盘标记高亮。

选择合作伙伴 IP 时,会打开一个对话框并为您提供相应的 IP 采购链接。"Customize IP"选项将保持灰显,直至您完成 IP 采购和安装为止。

使用付费许可 IP

Vivado IP catalog 会在 IP catalog 的"许可 (License)"列下显示"Included"或"Purchase",并提供有关许可状态的字段。 以下定义适用于赛灵思提供的 IP:

- "License Status": IP 许可状态包括"Full" (也称为"Purchased")、"Simulation"或"Eval"。
 - 。 "Included": 赛灵思《最终用户许可协议》[参照 1] 适用于赛灵思 Vivado Design Suite 软件工具中许可的赛 灵思 LogiCORE™ IP 核 (无需额外费用)。
 - 。 "Purchase":《核许可协议》[参照 2] 适用于付费赛灵思 LogiCORE IP,《核评估许可协议》[参照 3] 则适用于 付费赛灵思 LogiCORE IP 的评估。
- "License Type": 许可类型包括"Floating"或"Node-Locked"。
 - 。 基于证书的网络浮动许可证和基于激活的服务器许可证属于锁定到运行 FLEX 许可证服务器守护程序的许可服 务器主机的许可证。许可证按每个独立用户进行检出。
 - 。 单机版许可证或客户端许可证属于锁定到具体机器的许可证,或者对于基于证书的许可证,则属于锁定到特定硬件锁的许可证。只要您不更换自己的硬盘,即可依靠磁盘序列号(卷ID)来识别单机版许可证对应的节点。
- 其它许可证级别包括: "Design_Linking"、"Hardware_Evaluation"和"Full"(Purchased)。

如需了解有关如何获取 IP 许可的更多信息,请参阅赛灵思 IP 许可网站 [参照 1]。



请访问此链接并参阅《Vivado Design Suite 用户指南:版本说明、安装和许可》(UG973) [参照 31] 以获取有关许可的 更多信息。



 \diamondsuit

视频:请参阅以下内容以获取有关许可的更多信息:《Vivado QuickTake 视频: Vivado 激活浮动许可证生成》和《Vivado QuickTake 视频: Vivado 许可与激活简介》。

对于付费 IP, "Customize IP"对话框上的"OK"按钮将保持禁用状态,直至找到评估许可证或付费许可证为止,如下图 所示。

	uiteb la Dafaulta					
	which to Defaults					
Show disabled ports	Component Name v_ccm_0					
	I/O Options Matrix Values					
	Video Component Input Width 8	¥	Video Compo	onent Output Width	8	~
	Input Video Format RC	B v	Output Video	Format	RGB	v
	Optional Features					
= + video_in ack video_out + = ack ackcn video_out + =	AXI4-Lite Register Interface	9				
	Input Frame Dimensions					
	Pixels per Scanline (Default)	1920	0	[32 - 7680]		
	Scanlines per Frame (Default)	1080	0	[32 - 7680]		

图 2-9: 对于付费 IP, 找到许可证后即启用"OK"按钮

重要提示:在实现期间, Vivado 工具会确认针对 IP 是否存在许可证;因此,您必须重新生成 LogiCORE IP 核,这样核 网表才能接收最新许可证状态。



创建 IP 自定义

0

提示: 在文本框中输入或修改数据时,如果使用了某个可编辑的值,那么文本为黑色且背景为白色。如果使用了不可 编辑的值,那么文本为黑色且背景为灰色。如果未使用值或者使用的值不适用,那么文本(包括位于其前后的标签) 为灰色。

要使用 IP catalog 创建 IP 自定义,请双击 IP 将其选中,然后在工具栏中选择"Customize IP"命令,或者右键单击并选中此命令。

"Customize IP"对话框会显示可供您用于自定义此 IP 的各种参数。此界面因所选 IP 而异,可包含 1 个或多个选项卡以 便您在其中输入值。

并且, "Customize IP"对话框包含 IP 符号和选项卡,用于为特定 IP 设置配置选项。Vivado IDE 会将这些配置选项写入 <ip_name>.xci 文件,并将其作为属性存储在 IP 对象上。



注意! Windows 操作系统对于路径长度存在总计 260 个字符的限制,这可能影响 Vivado 工具。为避免此问题,创建 工程、定义 IP 或托管 IP 工程以及创建块设计时,请尽可能使用较短的名称和目录位置。在工程外部存储 IP 时请牢记此 信息。

IP 符号支持缩放、调整大小和自动适应选项,这些选项与 Vivado IDE 中的原理图视图画布一致。

图 2-10 显示了对应 FIFO Generator IP 的"Customize IP"界面。



ocumentation 📄 IP Location 🤇	Switch to Defaults		
] Show disabled ports	Component Name	axi_data_fifo_0	0
	PROTOCOL	AXI4 🗸	
	READ_WRITE Mode	READ WRITE V	
	Address Width	32 💿	[1 - 64]
	Data Width	32 🗸	
	ID Width	0 ~	
	User signal widths		
	AWUSER_WIDTH	H 0 6	0 - 1024]
	ARUSER_WIDTH	0	[0 - 1024]
+ S_AXI	WUSER_WIDTH	0	[0 - 1024]
aresetn	RUSER_WIDTH	0	[0 - 1024]
	BUSER_WIDTH	0 6	0 - 1024]
	Write FIFO Options		
	WRITE FIFO DEP	тн 0 🗸	
	WRITE FIFO DEL	AY OFF ~	
	Read FIFO Options		
	READ FIFO DEPT	гн о 🗸	
	READ FIFO DELA	Y OFF V	

图 2-10: FIFO Generator 对话框

在 IP 对话框中可设置下列选项:

- "Documentation > Product Guide": 打开选定 IP 的产品指南。
- 。 "IP Location": 指定磁盘上用于存储 IP 的位置。此位置只能在基于 RTL 的工程中根据每个 IP 进行调整。该功能在"Managed IP"工程中并未提供,因为"Managed IP"会构建存储库。
- 。 "Switch to Defaults":显示1个窗口,询问您是否要将所有配置选项复位到默认起点。

请根据您的设计需要对 IP 进行自定义, 然后单击"OK"。

建议: 赛灵思建议指定数字值时, 使用十六进制以加速处理。

适用于 IP 的 Tcl 命令

以下是适用于 IP 的 Tcl 命令示例。



IP 自定义创建示例

您还可使用 create_ip Tcl 命令来创建 IP 自定义。例如:

create_ip -name fifo_generator -version 12.0 -vendor xilinx.com -library ip\

-module_name fifo_gen

您必须指定 -vlnv 或者同时指定 -vendor、 -library、 -name 和 version。

注释:执行 create_ip Tcl 命令即可创建 IP 自定义文件 (XCI) 作为 IP 配置,并且也可以创建例化模板和 BOM (XML) 文件,但不会创建任何其它输出文件。 IP 默认配置是使用 create_ip 命令设置的。

IP 属性设置示例

要定义不同 IP 配置设置,请使用 set_property 命令。例如:

set_property CONFIG.Input_Data_Width 12 [get_ips fifo_gen]

IP 属性报告示例

要获取可用于 IP 的属性列表,请使用 report_property 命令。例如:

report_property CONFIG.* [get_ips <ip_name>]

配置属性以 CONFIG 开头。

IP 自定义属性查询示例

要判定 IP 自定义属性已设置为默认值还是由用户设置,请参阅以下示例:

```
# Find the read data count width.
get_property CONFIG.Read_Data_Count_Width [get_ips char_fifo]
10
# Determine the source of CONFIG.Read_Data_Count_Width property.
# See that this is the default value
get_property CONFIG.Read_Data_Count_Width.value_src [get_ips char_fifo]
default
# Get the output data width.
get_property CONFIG.Output_Data_Width [get_ips char_fifo]
8
```

Determine the source of CONFIG.Output_Data_Width property. # See that this is set by the user. get_property CONFIG.Output_Data_Width.value_src [get_ips char_fifo] user

生成输出文件

完成 IP 自定义后,会打开"生成输出文件 (Generate Output Products)"对话框。在"预览 (Preview)"区域中会列出 IP 交付 的输出文件。单击"Generate" 即可为 IP 创建 XCI 和 DCP,并创建更改日志、行为仿真模型和例化模板;否则,请单击 "Skip"。

这样即可选择多项 IP 自定义,并一次性生成所有输出文件,包括针对 IP DCP 文件启动并行综合运行。

 \bigcirc

提示: 在工程模式下工作时,会根据需要先自动生成输出文件,然后再执行顶层设计综合。这包括所有指定的 DCP 文件。此外,对于 IP 核,始终会为其生成 XCI 文件和例化模板,即使不生成其它输出文件也是如此。



默认情况下, Vivado Design Suite 会为综合后的 IP DCP 生成 OOC 运行。在"Generate Output Products"对话框中,选择以下任一选项:

- 。 "Global Synthesis":指示工具对当前设计执行自上而下的综合。选中该选项时会移除所有 OOC 运行文件。
- 。 "Out-of-Context Settings": 支持您添加 OOC 设置描述。

您还可指定同时启动的 OOC 综合运行数。默认情况下,指定 1 项作业,并且设计运行按顺序启动。在"Number of Jobs"选项中指定更大的数值,即可指定可并行运行的最大设计运行数。

更改这些设置后,您可生成输出文件或者延后生成输出文件。无论您当前是否生成输出文件,都会保留 DCP 生成的首选项设置。

使用 Tcl 命令来复位并生成目标 IP

复位和重新生成输出文件所需的 Tcl 命令如下所示:

reset_target:

```
reset_target all
```

```
[get_files /project_1/project_1.srcs/sources_1/ip/<core_name>.xci]
```

generate_target:

```
generate_target all
```

[get_files project_1/project_1.srcs/sources_1/ip/<core_name>.xci]

检验生成的输出文件

"IP Sources"窗口会显示为工程中所有 IP 生成的输出文件。默认情况下, IP 的输出文件将写入位于 <project_name>.srcs/sources_1/ip/<ip_name> 的本地工程目录; 但如果您从 IP catalog 自定义 IP, 那么 IP 位置可指定为位于本地工程目录外部的位置。

生成综合输出文件后, Vivado IDE 会创建并启动设计运行以生成 OOC DCP。

默认情况下, Vivado IDE 会在生成过程中为大部分 Vivado Design Suite IP 自动创建综合后的设计检查点 (DCP) 文件。

执行顶层设计综合时,将随关联 DCP 文件一起,将 IP 标记为非关联流程,并将其作为黑盒来处理,因为它正在通过 OOC 流程执行综合。

在处理综合运行时, OOC 相关文件会显示为缺失。

如果您选择使用"Global Synthesis",并且不生成 DCP,那么 Vivado IDE 不会生成结构仿真网表和存根文件。

如果您不生成输出文件,那么例化模板是唯一生成的文件 (如下图所示,位于图中未显示的 XCI 和 BOM 文件旁)。



图 2-11: 执行 IP 自定义并跳过生成输出文件



如下图所示,生成输出文件时,"IP Sources"窗口会列出未加密的文件。



图 2-12:执行 IP 自定义并生成输出文件,包括 OOC DCP

这些文件包括:例化模板、综合和仿真目标、 XDC 约束、更改日志及其它输出文件。

默认情况下, Vivado IDE 会随结构仿真网表(<ip_name>_sim_netlist.v 或 <ip_name>_sim_netlist.vhdl) 一起创建 OOC DCP,并创建存根文件 (*_stub.v/*_stub.vhdl) 以配合第三方综合工具一起用于为 IP 推断黑盒。

注释: 在早于 2015.3 的 Vivado Design Suite 版本中, 仿真文件为 named *_funcsim.v和 *_funcsim.vhdl。

注释: "IP Sources"中并不显示生成的所有 IP 输出文件。加密文件不显示,未置于综合文件组中的 XDC 文件同样不予显示。要查看这些文件,请查看"设计源 (Design Sources)"视图或"编译器顺序 (Compiler Order)"视图。您还可使用 report_compile_order 命令。

手动生成输出文件

您可随时通过选择"IP Sources"中的 IP、右键单击并选择"Generate Output Products"来手动生成输出文件。

生成输出文件的默认流程是通过非关联 (OOC) 流程为 IP 创建并启动综合设计运行。这样即可在对设计顶层进行综合时 推断 IP 黑盒。



重要提示:实现阶段可通过从 IP 的 DCP 中提取网表来对黑盒进行解析。

注释: 如果不想为 IP 生成 OOC DCP 文件,或者如果想要使用脚本化流程,则可将 GENERATE_SYNTH_CHECKPOINT 属性设置为 FALSE,这样生成输出文件时就不会创建检查点。

用于在 IP 上禁用 OOC 选项的 Tcl 命令

set_property GENERATE_SYNTH_CHECKPOINT FALSE [get_ips <ip_name>]



建议: 赛灵思建议您使用默认 OOC 来缩短用于设计综合的运行时间。如果使用 OOC 流程,则无需每次都在开发期间运行综合时对 IP 进行综合。



将现有 IP 添加到工程中

您可使用"Add Sources"选项来添加先前在 CORE Generator[™] 工具中创建的 IP (<ip_name>.xco 文件)或 Vivado IP (<ip_name>.xci 文件或 <ip_name>.xcix 文件)。您可将 IP 和任何生成的输出文件从其当前位置引用到工程中,或者也可以将 IP 或任何生成的输出文件复制到工程中。

通过 IP 自定义即可在其它设计中使用现有 IP,或者可使用"Managed IP"工程来对现有 IP 进行自定义以便在诸多其它设计中使用。"Managed IP"工程可创建 XCI 文件用于 IP 自定义,也可以生成任何必需的输出文件。

请参阅第 3 章"使用"管理 IP"工程",以获取有关使用"Managed IP"流程创建 IP 的详细信息。

添加的 IP 和任何输出文件会显示在"源文件 (Sources)"窗口的"IP 源文件 (IP Sources)"选项卡中,并且随其它源文件一起显示在"层级 (Hierarchy)"视图、"库 (Libraries)"视图和"编译顺序 (Compile Order)"视图中。

您可在"IP Sources"中选择 IP,并在"源文件属性 (Source File Property)"窗口中查看其属性。



重要提示: 在 Vivado Design Suite 中针对 UltraScale[™] 器件不支持 NGC 格式文件。赛灵思建议您使用具有本机输出文件的 Vivado Design Suite IP 自定义工具重新生成 IP。或者,您可使用 NGC2EDIF 命令将 NGC 文件移植到 EDIF 格式。 请访问此链接并参阅《ISE 到 Vivado Design Suite 移植指南》(UG911) [参照 25] 以获取有关为 Vivado 移植文件的更 多信息。

将现有 IP 添加或导入到工程中时,此 IP 的现有输出文件将被引用或复制到工程中;但不会引用或复制设计运行。

要为 IP 创建设计运行,有 2 种方法可选:为 IP 重新生成输出文件,或者在 Tcl 控制台 (Tcl console) 中输入 create_ip_run 命令:

create_ip_run -force [get_ips <ip_name>]

这样设计运行会报告称尚未启动。添加的 IP 还需例化到顶层设计中,如"例化 IP"中所述。

使用 Tcl 命令添加现有 IP

以下显示了用于将现有 IP 及其生成的输出文件添加到工程中的 Tcl 命令:

- import_files 用于添加现有 IP: import_files <ip_filename>
- read_ip 用于远程访问 IP: read_ip <ip_filename>

解决 IP 重复

如果输入的 IP 与现有 IP 同名,那么 Repository Manager 会发出警告。在此情况下您可执行的操作如下所示:

- 。 在 IP 目录中复查重复的 IP。
- 。 复查描述中的目录。
- 。 如果需要,请移除 IP 或者移除包含此 IP 的存储库。

创建存储器 IP 自定义

"存储器 IP (Memory IP)"可为赛灵思器件和 IP 创建内存控制器。"Memory IP"可为选定的 FPGA 创建完整的自定义 RTL 源代码、管脚输出以及设计约束,并创建用于实现和仿真的脚本文件。

在 7 系列器件中,存储器 IP 被称为 Memory Interface Generator (MIG)。在 UltraScale™和 UltraScale+器件中,已弃用 此术语。请参阅《UltraScale 架构 FPGA 存储器 IP LogiCORE IP 产品指南》 (PG150) [参照 5] 和《Zynq-7000 SoC 和 7 系列 FPGA 存储器接口解决方案》 (UG586) [参照 12] 以获取更多信息。



对于 Zynq[®] UltraScale+ MPSoC 处理器上的存储器 IP, Vivado 工具会启动管脚分配工程,并支持您为该器件设置相应 的管脚。请访问此链接并参阅《Vivado Design Suite 用户指南: I/O 管脚分配和时钟规划》(UG899) [参照 19] 中的相应 内容。



视频: 请参阅 《Vivado Design Suite QuickTake 视频:利用 UltraScale 存储器 IP 进行设计》以获取有关如何使用存储器 IP 的指示信息。

此外,请访问"解决方案中心"以获取有关存储器 IP 的更多信息。赛灵思 Document Navigator 中的"采用 IP 进行设计"设计中心提供了存储器 IP 的相关视频以及文档的链接。

重新自定义现有 IP

您可在 RTL 工程或"Manage IP"工程中重新自定义现有 IP。要为 IP 打开"IP 自定义 (IP customization)"对话框,请双击 IP 或者在"IP Sources"视图中右键单击 IP 并从上下文菜单中选择"Re-customize IP"。

更改 IP 配置设置,然后单击"OK"。这样会打开"生成输出文件 (Generate Output Products)"对话框,以便您重新生成输出文件。或者,您可以复查当前设置,然后单击"Cancel"以保留设置不变。

当您对 IP 配置执行更改并生成输出文件后,现有输出文件将复位,并且设计运行 (如果已启用)也将复位并再次启动。

建议: 赛灵思建议您始终为 IP 生成输出文件,包括综合后的 DCP;但如果您在自定义或重新自定义 IP 后不生成输出文件,那么 Vivado Design Suite 会根据需要自动生成输出文件,例如,在综合项层设计时。

复制 IP

您可复制现有 IP 自定义以便将其用作为新 IP 的起点。如果您已自定义 IP 并且只需对新 IP 进行少量或简单的自定义更改,那么此方法很有用。

要复制 IP,请在"IP Sources"中选中 IP,右键单击并选择"Copy IP"。然后为副本提供目标名称和位置,如下图所示。

🥕 Сору IP	×
Enter a new name and destination to copy the selected IP.	4
Destination IP name: ila_0_1	8
Destination IP Location: oject_8.srcs/sources_1/ip OK	el

图 2-13: "复制 IP (Copy IP)"对话框

将复制的 IP 保存到工程目录结构 (默认位置为 <project_name>.src/sources_1/ip/),或者指定位于当前工程 之外的其它位置以存储复制的 IP。处理"Manage IP"工程时,默认位置与 /manage_ip_project 目录所在位置相同。 然后,您可通过双击 IP 或者右键单击并从"IP Sources"选项卡中选择"Re-customize IP"来重新自定义已复制的 IP 自定 义。这样会打开"复制的 IP 自定义 (copied IP customization)"窗口,其中包含来自源 IP 的自定义设置。现在,您即可对 其进行编辑。

E XILINX_®



注意! IP 可引用其它子 IP 版本,包括已锁定的旧版本或其它新版本 IP。在此类情况下,综合工具可能会生成错误或逻辑错误 (logic bug),因为有文件包含相同模块名称但内容不同。请升级子 IP 或对非关联的 IP 进行综合。

用于复制 IP 的 Tcl 命令示例

您可使用 copy_ip 命令来创建 IP 自定义的副本:

copy_ip -name newFIFo [get_ips char_fifo]

此示例会创建 char_fifo IP 的副本、将其命名为 newFIFO,并将其添加到工程中。由于未使用 -dir 选项指定任何 目录,因此创建的 IP 位于工程目录结构内。

例化 IP

Vivado 工具会在 IP 自定义后创建例化模板,与是否已生成输出文件无关。例化模板显示在 Sources > IP Sources > Instantiation Template 目录中。

创建 IP 自定义后,它会打开 IP 例化模板并将相关代码从模板复制到您的代码中。Vivado 工具会生成 VHDL 和 Verilog 例化模板以供您选择并复制粘贴到自己的 RTL 设计中。

如果您要在自己的设计中使用例化模板,请执行以下步骤:

- 1. 在"Sources"视图中双击 IP 自定义的例化模板文件以将其打开,或者使用"Open Files"命令选中该文件。
- 2. 根据例化模板文本中的指示,高亮例化模板的注释之间的内容,并复制此段代码。
- 3. 打开设计 HDL 文件,以便您在其中于设计顶层或设计层级内例化此 IP。
- 4. 将复制的模板粘贴到您所选的位置。
- 5. 编辑此 HDL 以根据需要将模板集成到您的设计中:例如,更改端口连接,并为例化提供唯一名称。

将此 IP 例化到设计中后,即可在设计层级中正确列示此 IP。将 IP 自定义正确例化到设计中之后,即可将此 IP 与设计 其余部分一起进行综合,如果使用的是 OOC 流程,则可将其综合为黑盒,或者如果使用全局综合,则可将 IP 与设计 顶层一起进行综合。如需了解更多详情,请参阅"IP 的综合选项"。

提示: 右键单击 IP 展开 IP 层级时,即可看到"加密 IP 源文件 (Encrypted IP source)"图标 🧬。此源文件的内容无法查 看。请访问此链接并参阅《Vivado Design Suite 用户指南: 创建和封装定制 IP》 (UG1118) [参照 34] 以获取有关"在 Vivado 中加密 IP"的信息。



重要提示:可以创建重复的 IP。如有重复 IP,请参阅"解决 IP 重复"。

报告 IP 状态

您可使用"Reports > Report IP Status"来查看工程中所有 IP 的报告。新选项卡"IP Status"可显示报告结果。

状态筛选

您可右键单击源文件以显示选项列表,以便筛选"IP 状态 (IP Status)"报告中显示的信息。选项如下:

- 重要更改 (Major change)
- 次要更改 (Minor change)
- 修订版更改 (Revision change)



- 器件更改 (Part change)
- 最新状态 (Up-to-date)
- 其它 (Other): 此类别包含处于其它状态的 IP。示例如下:
 - 。 未找到 IP 定义 (IP definition not found)
 - 。 只读 XCI、 XML 或 XPR 文件 (Read-only XCI, XML, or XPR file)
 - 。 用户管理的 IP (User-managed IP)
 - 。 禁用的组件 (Disabled component)
 - 。 许可证不兼容 (Incompatible license)
 - 。 XCI 或 XML 文件不兼容 (Incompatible XCI or XML file)
 - 。 已弃用的流程 (Deprecated flow)
 - 。 因子 IP 被锁定而导致锁定 (Locked due to child IP being locked)

查看更改日志

每个 IP 均可提供一份更改日志。更改日志可提供有关每个版本的 IP 更改的信息。您可通过以下选项来访问更改日志:

- 在"Sources"视图中的"IP Sources"部分中,选中 IP,然后右键单击并选择"IP Documentation > View Change Log"。
- 在"源文件属性 (Source Files Properties)"视图中,选中 IP 然后向下滚动并选中"View Change Log"。您可选择查 看完整日志链接以打开完整更改日志,如图 2-14 中所示。



图 2-14: 查看更改日志以获取更多信息链接

使用 Tcl 命令报告 IP 状态

您还可使用 report_ip_status Tcl 命令生成此信息:

report_ip_status

了解工程中的 IP 状态

根据目录中的当前版本以及您是否已生成输出文件,工程中显示的 IP 可能处于多种状态。



重要提示:如果已导入的 IP 核的版本无法从 Vivado IP catalog 访问,则不启用 IP 重新自定义、重新设置和重新生成功能。



添加 XCI 或 XCO 格式的现有 IP (如果存在)时,还会一并添加输出文件 (例如, HDL 文件)。

下表显示了表示 Vivado IDE IP 状态的按钮。

表 2-2:	工程中的	IP 状态
--------	------	-------

按钮	描述
	RTL 工程中将以 OOC 流程进行综合的 IP。请参阅"非关联流程"。
<u>_</u> =	IP 目录中需随工程一起进行综合(全局综合)的自定义 IP。请参阅"全局综合流程"。
轷	非受管 IP。用户已将 IP 的 IS_MANAGED 属性更改为 false 并且已承担 IP 管理职责。此 IP 同样已变 为锁定状态。目的是便于用户对未加密的 HDL 源代码或约束进行修改。请参阅附录 D:编辑或覆盖 IP 源文件。
4	IP 已锁定,其输出文件可在流程中使用,但不可重新自定义或重新生成。已锁定的 IP 如不含输出文件,则不可在流程中使用。要使用已锁定但不含输出文件的 IP,请提供原始输出文件,或者升级至最新版本。如需了解更多信息,请参阅附录 A: 判定 IP 锁定原因。

管理 IP 约束

Vivado IDE 可为整个设计以及赛灵思 IP 管理用户定义的 XDC 时序约束和物理约束。它可负责处理在工程内多次例化的 赛灵思 IP 的关联与统一。

IP 目录中的大部分 IP 均可根据用户自定义提供特定于 IP 的 XDC 约束。由 IP 提供的约束使用默认综合设置加以最优化。

建议: 请勿为任何 IP 设计运行更改这些设置,否则应用约束时可能会遇到问题。要获取所有权以便对 IP 进行约束,请 禁用随 IP 交付的 SDC 文件。如果不得不为 IP OOC 运行更改综合设置,可以在 Tcl 控制台中使用以下 set_property 命 令: set_property <synthesis_option> <value> [get_runs <ip_name>_synth_1]。

用于更改综合运行属性的 Tcl 命令示例

set_property STEPS.SYNTH.DESIGN.ARGS.FSM_EXTRACTION sequential /
[get_runs <ip_name>_synth_1]

设计综合和实现期间, Vivado Design Suite 会根据约束文件先处理 IP 交付的 XDC 约束, 然后再处理用户定义的约束。



注意!如果通过 OOC 流程对任何 IP 进行综合,那么顶层综合运行会为这些 IP 推断黑盒。因此,用户将无法在顶层综 合约束内引用 IP 内部的任何对象 (例如,管脚、信号线、单元等)。实现期间,来自 IP DCP 的网表将与顶层设计文件 综合时生成的网表相链接,而 Vivado Design Suite 则会对 IP 黑盒进行解析。已生成的可供实现期间使用的 IP XDC 输 出文件将随任何用户约束一起应用。



约束文件处理顺序

默认情况下, IP XDC 约束的 PROCESSING_ORDER 值为 EARLY,用户约束则标记为 NORMAL。因此,后处理的约束即可覆盖先处理的相同对象上的约束。

IP XDC 文件处理顺序如下所示:

- 。 用户 XDC 设置为 EARLY
- 。 IP XDC 设置为 EARLY
- 。 用户 XDC 设置为 NORMAL (默认值)
- 。 IP XDC 设置为 LATE
- 。 用户 XDC 设置为 LATE

通过使用此方法,即可调整 XDC 文件与 IP XDC 的处理先后顺序。

欲知详情,请参阅下列文档:

- 。 《Vivado Design Suite 用户指南: 创建和封装定制 IP》 (UG1118) [参照 34]
- 。 此链接指向《Vivado Design Suite 用户指南:如何使用约束》(UG903) [参照 21]

Vivado IP 可生成多个 XDC 约束文件。默认情况下, IP 约束的处理顺序先于用户约束的可能原因如下:

- 。 IP 可能生成最终用户约束必需的时钟。
- 。 如果 IP 提供物理约束,那么最终用户可根据需要覆盖这些约束。

以下是用于报告约束编译顺序的 report_compile_order Tcl 命令示例:

report_compile_order -constraints

此命令可提供约束处理顺序,用于用户逻辑的综合与实现,并且可提供用于每个 IP 综合运行的约束细分,用于生成 IP DCP。

IP 提供的部分约束与来自最终用户或其它 IP 的时钟对象之间存在依赖关系。这些约束通过单独的 XDC 文件提供,并且 其处理顺序晚于最终用户约束。

通常, IP 提供的核 XDC 文件可包含时钟创建命令以及无外部时钟依赖关系的命令。约束文件名为 <ip_name>.xdc, 并且被称为核 XDC 文件。

IP 还可包含其它 XDC 文件,其中可能包含存在时钟依赖关系的命令。

由于顶层时钟可能来自其它约束或者来自具有依赖关系的其它 IP,因此需首先定义此类时钟的任何约束都应置于 <ip_name>_clocks.xdc 内。默认情况下, Vivado IDE 会先处理用户约束和其它 IP 核 XDC 文件,然后再处理 <ip_name>_clocks.xdc 文件。

大部分 IP 同样可提供 OOC XDC 文件 (<ip_name>_OOC.xdc)。此文件包含 IP 输入时钟的默认顶层定义。仅限在使用 推荐的默认流程创建 DCP 时(IP 通过 OOC 流程综合到顶层设计)才可使用此文件。当 Vivado Design Suite 通过 OOC 流程对顶层设计的 IP 进行综合时,由最终用户或其它 IP 创建的时钟不可用;因此,此文件是提供用于 IP 综合的时钟 定义所必需的。

利用 IP 对用户逻辑进行实现期间无需 <ip_name>_ooc.xdc,因为应用约束前所有网表都链接在一起。此时,用户创建的时钟或 IP 创建的时钟可供需时钟的任何 IP 使用。

部分 IP 可提供更多 XDC 文件。这可能是因为这些 IP 提供的约束仅用于综合或仅用于实现。要获取 IP 可提供的 XDC 文件的列表,请参阅附录 B: IP 文件和目录结构。

部分 IP 支持 Vivado 开发板流程,请访问此链接,并参阅《Vivado Design Suite 用户指南:系统级设计输入》(UG895) [参照 18] 以获取详情。



如果您创建的工程以平台开发板而非目标器件为目标,那么此开发板可在 IP 自定义期间供您使用,以便您指定连接到 IP 时要使用的板上连接。这样会生成 <ip_name>_board.xdc 文件,其中包含 PACKAGE_PIN、 IOSTANDARD 和其 它物理约束。

如需了解有关 XDC 约束的更多详细信息,请访问此链接以参阅《Vivado Design Suite 用户指南:如何使用约束》(UG903) [参照 21]。

为工程处理完部分约束后,这些约束会成为工程属性。如需了解有关属性的更多信息,请参阅《Vivado Design Suite 属性参考指南》 (UG912) [参照 26]。

视频: 请参阅《Vivado Design Suite QuickTake 视频: IP 约束概述》以观看有关 IP 流程期间如何使用以下约束的演示。

以下段落简要描述了 Vivado Design Suite 在处理 IP 时创建的部分约束文件。

dont_touch.xdc 约束

Vivado 工具在 IP 综合期间使用 dont_touch.xdc 在 IP 顶层设置 DONT_TOUCH 属性。这样即可避免移除接口端口。

使用 OOC (默认流程)进行 IP 综合时可在综合日志文件中看到此约束文件的处理过程,或者使用最终用户 RTL 进行 IP 综合时可在全局综合日志文件中看到此约束文件的处理过程。

in context.xdc 约束

默认情况下,顶层综合期间 IP 作为黑盒来处理,因为它是采用 OOC 流程来进行综合的。创建 IP DCP 时,在以下情况下,会在 DCP 文件中创建并存储 <ip_name>_in_context.xdc 文件:

- 。 IP 生成可在 IP 边界引用的时钟
- 。 IP 包含任何 I/O 缓存实例

综合用户逻辑时, <ip_name>_in_context.xdc 文件 (如果存在)的处理顺序先于最终用户约束。实现期间无需 此文件,因为此 IP 不再为黑盒。

如果创建时钟,则会将其布局在 IP 黑盒单元的边界管脚处。时钟类型可包括:

- 。 位于 IP 输入端口上的基准时钟 (例如, Clocking Wizard IP)
- 。 位于 IP 输出端口上的基准时钟
- 。 位于 IP 输出端口上的生成时钟,其中主时钟为输入时钟(例如, Clocking Wizard IP)

仅当 IP 包含输入缓存时,才会在 IP 的输入端口上创建时钟。默认情况下, Clocking Wizard 采用此方式进行配置。

该时钟会在顶层用户逻辑综合期间传输至顶层端口。

如果用户约束必须引用 IP 生成的时钟,则应通过引用生成时钟的 IP 的管脚以间接方式来执行,如以下 get_clocks 命令 所示:

get_clocks -of_objects [get_pins <IP_clock_pin>]

如果存在 I/O 缓存,则对于 I/O 缓存的接口管脚, IO_BUFFER_TYPE 属性设置为 NONE。设置该属性可防止在顶层综合期间插入额外的 I/O 缓存。



设置目标时钟周期

默认情况下, IP 采用 OOC 流程独立于设计其余部分进行综合。

- 用户逻辑综合期间,此 IP 被视为黑盒。
- 实现期间, IP 网表与其它 IP 网表和用户网表相链接。

由于 IP 与顶层分离并单独进行综合,因此 Vivado 工具会在 OOC 完成后创建 get_runs <ip_name>_ooc.xdc 以便 为 IP 提供时钟定义,如"IP 的综合选项"中所述。

如果您不指定目标周期,那么 IP 会使用默认时钟周期。当单独用于 IP 的周期与顶层综合时的周期不同时,可能导致出现如下警告:

(1) [Timing 38-316] Clock period '10.000' specified during out-of-context synthesis of instance 'char_fifo_i0' at clock pin 'rd_clk' is different from the actual clock period '6.000', this can result in different synthesis results.

此警告信息会告知您,如果使用的是全局综合,则 IP 综合时所使用的时钟周期不同于 OOC IP 综合时所使用的时钟周期。

IP 具有多种时钟选项:

- 在自定义 GUI 中, IP 可能具有多种选项可用于设置 OOC 综合期间所使用的目标频率/周期。您可使用 Tcl 控制台 来查询和设置 IP 的配置选项。
- IP 可能包含"Clocks"选项卡,用于为 IP 设置目标频率。
- IP 可能将目标频率/周期设置与其它设置混合在一起。通常,其中包含用于解释设置的工具提示。
- 未提供用于自定义目标时钟频率或周期的 GUI 选项的 IP 必须依靠其它时钟源来设置频率。

IP 的时钟端口具有与之关联的属性 (以 FREQ_HZ 结尾)。更改这些属性会导致为 IP 生成输出文件时, IP 的 _ooc.xdc 文件使用这些值。

使用 Tcl 命令设置目标时钟周期

要自定义 IP,请在 IP 对象上设置唯一的属性。使用 Vivado IDE 配置 IP 时, Vivado IDE 会自动发出对应的 Tcl 命令。

如果 IP 未提供任何 GUI 方法来为 OOC 综合下的 IP 设置目标周期,那么您可手动设置此目标周期。

IP 自定义更改是永久性行为,除非再次使用 Tcl (或使用适用的 IP 自定义 GUI)进行更改;复位和重新生成 IP 时,将 保留您的更改。

以下示例显示了为 FIFO Generator IP 设置目标时钟的步骤,此 IP 名为 char_fifo,并在波形生成器 (Wave Generator) 设计示例中使用。此 IP 已自定义为针对读写端口和本机接口使用公共端口。

1. report_property 命令可用于报告此 IP 可用的属性。在 Tcl 控制台中输入以下命令:

report_property [get_ips char_fifo]

- 2. 在输出中,您可看到针对此 IP 存在 5 个以 FREQ_HZ 结尾的属性:
 - 。 CONFIG.core_clk.FREQ_HZ: 使用公共时钟 (此示例)时适用该属性
 - 。 CONFIG.read_clk.FREQ_HZ: 使用独立时钟时适用该属性
 - 。 CONFIG.write_clk.FREQ_HZ: 使用独立时钟时适用该属性
 - 。 CONFIG.slave_aclk.FREQ_HZ: 使用 AXI 时适用该属性
 - 。 CONFIG.master_aclk.FREQ_HZ: 使用 AXI 时适用该属性



仅限第一项输出才适用于使用公共时钟的本机接口。默认情况下, CONFIG.core_clk.FREQ_HZ 设置为 100000000 或 100MHz。

如果已生成 IP, 可查看 char_fifo_ooc.xdc 文件中的以下行:

create_clock -period 10 -name clk [get_ports clk]

周期为 10 对应于 CONFIG.core_clk.FREQ_HZ 值 100MHz。

对于此示例,如果期望的时钟频率为 250MHZ,请按如下方式设置:

3. 在 Tcl 控制台中输入以下 Tcl 命令:

set_property CONFIG.core_clk.FREQ_HZ 250000000 [get_ips char_fifo]

4. 设置属性后,生成 IP。这将导致 OOC 运行 (如果存在)复位并重新运行。

运行完成后,请查看 char_fifo_ooc.xdc 文件。其中会显示:

create_clock -period 4 -name clk [get_ports clk]

现在,执行 IP 综合时就会使用期望的时钟周期/频率。

判定时控约束和解释时控消息

Vivado 可包含层级约束、顶层用户约束和 IP 提供的约束。这些约束可能存在某些依赖关系,必须满足这些依赖关系后才能正确生效。比如,时钟创建约束。

- IP 可能创建其它 IP 或顶层设计所必需的时钟。
- IP 可能需要存在时钟才能正常运行,并避免生成严重警告。

如果未提供必要的时钟约束,那么设计顶层的 IP 会发出 CRITICAL WARNING,如"有关时控的严重警告和警告示例"中所述。

如需了解有关处理含时控要求的设计的更多信息,请访问此链接并参阅《Vivado Design Suite 用户指南:如何使用约束》(UG903) [参照 21]。

视频:请参阅《Vivado Design Suite QuickTake 视频:创建基本时钟约束》和《Vivado Design Suite QuickTake 视频:
 创建生成时钟约束》,以获取更多信息。

处理 IP 时钟依赖关系的 Tcl 命令示例

以下示例演示了使用 set_max_delay 命令的 IP 约束,此 IP 具有与顶层时钟的依赖关系,该顶层时钟由 ref_clk 端口 上的 IP 提供。

set_max_delay -from [get_cells data_reg] -to [get_cells synchro_stage0_reg]\
-datapath only [get property PERIOD [get clocks -of objects [get ports ref clk]]]

《Vivado Design Suite 用户指南:如何使用约束》(UG903) [参照 21] (可通过此链接访问)描述了在 IP 单元实例上,如何将 get_ports 命令转换为 get_pins 命令。根据设计中 IP 的连接方式,该时钟可能源自用户提供的时钟或其它 IP:

- 如果时钟是由其它 IP 提供的,则会提供该时钟并且不会生成严重警告。
- 如果此时钟是在最终用户逻辑中提供的,那么不创建任何时钟对象 (使用 create_clock 命令或 create_generated_clock 命令) 的情况下会生成严重警告。

用于时控的 Tcl 命令示例

在设计中查找未正确生成的时钟的方法之一是使用 report_clock_networks Tcl 命令:



report_clock_networks

此命令会为设计生成时钟报告,包括已约束的时钟和未约束的时钟。您可使用此报告来判定连接到您的 IP 的时钟模块 是否缺少时钟定义。

其它实用命令包括:

report_clocks

此命令会返回一个表格,其中显示了设计中的所有时钟,包括当前已综合或已实现的设计中已传输的时钟、生成时 钟、自动生成的时钟、虚拟时钟和反相时钟。

report_compile_order

report_compile_order -constraints

此命令会显示设计用于综合和实现的 XDC 文件及其处理顺序。如果创建时钟的 IP XDC 来自需要时钟的 IP XDC,那 么此命令可用于对关系进行分类。

通常只需向顶层 XDC 时序约束文件添加约束即可解决 IP 缺失时钟的问题。如果处理的 XPS 设计中某些可用于创建时钟的 IP 不含任何 XDC 文件 (例如,串行收发器),则可能发生此状况。

有关时控的严重警告和警告示例

以下示例显示了设计未能找到 IP 核所需时钟约束时返回的警告。

CRITICAL WARNING: [Vivado 12-259] No clocks specified, please specify clocks using -clock, -fall_clock, -rise_clock options

[C:/Design/v_tc.xdc:1]INFO: [Vivado 12-1399] There are no top level ports directly connected to pins of cell 'system/v_tc', returning the pins matched for query '[get_ports s_axi_aclk]' of cell 'system/v_tc'.

[C:/Design/v_tc.xdc:1]Resolution: The get_ports call is being converted to a get_pins call as there is no direct connection to a top level port. This could be due to the insertion of IO Buffers between the top level terminal and cell pin. If the goal is to apply constraints that will migrate to top level ports it is required that IO Buffers manually be instanced.

CRITICAL WARNING: [Vivado 12-1387] No valid object(s) found for set_max_delay constraint with option 'from'.

 $[C:/Design/v_tc.xdc:1]$ Resolution: Check if the specified object(s) exists in the current design. If it does, ensure that the correct design hierarchy was specified for the object.

IP 的综合选项

为 IP 生成输出文件时,默认行为是输出通过 OOC 流程综合的设计检查点 (DCP)。或者,您可以选择将 IP 随顶层用户逻辑一起进行综合,这称为"全局综合"。

在上述任一流程中, Vivado IDE 都会为 IP 生成 HDL 和 XDC 文件,并在综合期间和实现期间使用这些文件,如下图所示。




图 2-15: 非关联 (OOC) 综合流程和全局综合流程

全局综合流程

处理 Vivado Design Suite IP 时,可禁用生成非关联 (OOC) DCP,改为使用"Global Synthesis"选项随项层设计进行 IP RTL 综合。

选择全局综合流程时,Vivado 工具会将 IP 与用户 HDL 一起进行综合。对用户 HDL 执行的任何更改也会导致重新对 IP 进行综合。

实现期间, Vivado 工具会将已生成的可供实现期间使用的任何 IP XDC 输出文件与任何用户约束一起使用。





建议:请始终使用 XCI 文件来引用 IP。不建议通过工程模式或非工程模式流程直接读取 IP DCP 文件。虽然 DCP 确实包含约束,但它不提供 IP 可交付的其它可能需要的输出文件,如 ELF 文件、 COE 文件或 Tcl 脚本。

非关联流程

建议:建议生成 IP 时使用 OOC 流程,这也是 Vivado Design Suite 中的默认行为。OOC 流程可缩短完整工程的运行时间,并避免执行工程运行时对 IP 进行重新综合。

在 OOC 流程中, Vivado 工具会将 IP 作为独立模块进行综合并生成 OOC 设计检查点 (DCP)。Vivado 工具进行 IP 综合 时还会生成并使用一个特殊 OOC 流程,即赛灵思设计约束 (XDC) 输出文件 _ooc.xdc,此文件可提供默认输入时钟定 义。生成的 DCP 是一个容器文件,其中包含网表和约束。_ooc.xdc 文件属于 IP 定义的一部分。

综合顶层设计时,将随 DCP 文件提供 HDL 存根模块,导致为 IP 执行黑盒推断。并且在对整个设计进行综合期间,DCP 还会提供 XDC 文件 _in_context.xdc 以定义 IP 可能输出的任何时钟,供顶层设计使用。如需了解更多信息,请参阅"管理 IP 约束"。



注意!如果通过 OOC 流程对任何 IP 进行综合,那么顶层综合运行会为这些 IP 推断黑盒。因此,用户将无法在顶层综合约束内引用 IP 内部的任何对象 (例如,管脚、信号线、单元等)。

实现期间,来自 IP DCP 的网表将与顶层设计文件综合时生成的网表相链接,而 Vivado Design Suite 则会对 IP 黑盒进行 解析。已生成的可供实现期间使用的 IP XDC 输出文件将随任何用户约束一起应用。

采用 OOC 流程作为默认流程主要是因为它具备 2 大优势:

- 它可缩短综合运行时间,因为它仅根据 IP 自定义或版本更改的需要来执行综合,而不是在顶层设计过程中执行重新综合。
- 它可生成 <ip_name>_sim_netlist.v 或 <ip_name>_sim_netlist.vhdl 结构仿真网表。如使用单一语言 仿真器且 IP 不提供该语言的行为 HDL,则可在仿真期间使用这些文件。请参阅"仿真 IP"以获取更多信息。

注释: 在早于 2015.3 的 Vivado Design Suite 版本中, 仿真文件为 named *_funcsim.v和 *_funcsim.vhdl。

仿真 IP

在设计流程中,使用仿真是验证设计的功能和性能的重要且必要的步骤。为支持此步骤,Vivado IP 目录中的 IP 可提供 仿真模型以供您包含在整体设计的仿真中。

为 IP 提供的仿真模型包括:

- 定制行为仿真模型
- 明文或加密可综合 RTL 源代码,用于仿真
- 结构仿真模型

注释:部分 IP (例如, FIR Compiler IP) 可提供 IP 级测试激励文件,以供您直接用于 IP 仿真。请参阅"为 IP 使用测试 激励文件"以获取更多信息。



提示:通常用于对赛灵思器件进行仿真的第三方仿真器作为选项集成在 Vivado® 集成设计环境 (IDE) 中。请访问此链接并参阅《Vivado Design Suite 用户指南:逻辑仿真》 (UG900) [参照 20] 以获取有关使用第三方仿真器的更多信息。相关文件位于 ip_user_files 目录中。



提供 IP 仿真模型

大部分赛灵思 IP 仅提供单一语言的 RTL 源代码 (Verilog 或 VHDL),如果您没有 IP 支持的语言的相应许可,那么可以为"语言锁定"的仿真器禁用仿真。

为进行设计仿真并包含 IP, Vivado 工具使用工程属性"Simulator language"设置来确保 IP 的相应仿真模型可供使用。 当前工程的 SIMULATOR_LANGUAGE 属性支持您将自己的仿真器所支持的语言告知 Vivado 工具。值包括 "Verilog"、"VHDL"和"Mixed"。请在基于"Manage IP"工程的流程和非工程流程中设置该属性。

部分 IP 可为 VHDL 提供仿真文件,部分 IP 则为 Verilog 提供仿真文件。当仿真器语言设置为"Mixed"时,不同 IP 可向 仿真器发送对应两种语言的同一个模块。

Vivado 仿真器为混合语言仿真器,可处理采用 VHDL 和 Verilog 语言的仿真模型。如果使用第三方仿真器并且只具有对应单一语言的许可证,请将"Simulator language"更改为与您的许可证匹配的仿真器语言。

如果 IP 不提供行为模型或者所提供的模型与所选的已许可的仿真器语言不匹配,那么 Vivado 工具会自动生成结构仿真 网表 (<ip_name>_sim_netlist.v 或 <ip_name>_sim_netlist.vhdl)以支持仿真。

注释: 在早于 2015.3 的 Vivado Design Suite 版本中, 仿真文件为 named *_funcsim.v 和 *_funcsim.vhdl。

建议: 生成 IP 输出文件时,请启用综合后设计检查点 (DCP) 选项以确保 Vivado IDE 可提供 IP 的结构仿真网表。如需了 解更多信息,请参阅"生成输出文件"。

注释:部分赛灵思 IP 使用 Vivado 高层次综合 (HLS) 工具来生成 RTL。这些 IP 需要运行综合后才能生成这些 RTL 文件。从 Vivado 内部启动仿真时,将根据需要自动为 IP 启动非关联综合运行或全局综合运行。

建议:为了使 IP 目录中仿真所需的所有文件均可置于版本控制系统中,建议首先运行综合。

如果仿真器语言未设置为"Mixed",可能需要使用默认 OOC 综合生成 IP。如果您使用的 IP 不提供采用指定仿真语言的 RTL,那么必须创建_sim_netlist.v或_sim_netlist.vhdl 文件以供仿真。仅限在 OOC 综合流程中才能创建这些文件。如果可用仿真文件与"仿真语言 (Simulation Language)"设置之间存在不匹配,将显示以下消息。

Simulation	n Model Incompatibility
•	The Project's Simulator Language setting and the behavioral simulation model language of the following IP are incompatible. The behavioral simulation may fail to elaborate. thigh_speed_selectio_wiz_0.xci trimicroblaze_0.xci traui_0.xci fv rxaui_0.xci furn on Out-of-context settings for the IP listed above to automatically use a structural simulation netlist model instead.
	Are you sure you want to continue generation?

图 2-16: "仿真模型不兼容"对话框



验证 IP

对使用 AXI IP 的设计执行仿真时,"验证 IP (Verification IP)"很有用。如需了解更多信息,请参阅下列参考文档:

- 《AXI Verification LogiCORE IP 产品指南》 (PG267) [参照 10]
- 《AXI4-Stream Verification LogiCORE IP 产品指南》 (PG277) [参照 11]
- 《Zynq-7000 SoC Verification IP 数据手册》 (DS940) [参照 1]
- 《Zynq MPSoC UltraScale Verification IP 数据手册》 (DS941) [参照 2]



重要提示: AXI Verification IP 是以 SystemVerilog 编写的,并支持使用随机化。并非所有第三方仿真器都支持 SystemVerilog 和随机化。请参阅《Vivado Design Suite 用户指南:版本说明、安装和许可》(UG973)[参照 31] 以获取有关 AXI VIP 的第三方兼容性的信息。



视频:此外,以下视频有助于您了解如何使用 Zynq-7000 VIP 进行仿真:《Vivado Design Suite QuickTake 视频:如何使用 Zynq-7000 Verification IP 通过仿真来进行验证和调试》。



您可例化 IP。如需为接口提供额外 Verification IP 支持,请使用以下任一 Tcl 命令:

set_property CONFIG.INSERT_VIP 1 [get_bd_intf_pin <path_to_interface>]

set_property CONFIG.<interface_name>.INSERT_VIP 1 [get_ips <ipname>]

适用于仿真的 Tcl 命令

要指定仿真器语言,请在 Tcl 控制台中输入以下命令:

set_property SIMULATOR_LANGUAGE <language_option> [current_project] 下表显示了仿真语言属性、语言和在其中应用属性的仿真模型。

表 2-3: 仿真器语言属性

仿真模型	语言	仿真模型
	Mixed	提供的采用指定 SIMULATOR_LANGUAGE 的 行为仿真模型。
IP 可提供 VHDL 和 Verilog 行为模型	Verilog	Verilog 行为模型。
	VHDL	VHDL 行为模型。
	Mixed	Verilog 行为模型。
IP 仅提供 Verilog 行为模型	Verilog	Verilog 行为模型。
	VHDL	从 IP DCP 生成的 VHDL 仿真网表。
	Mixed	VHDL 行为模型。
IP 仅提供 VHDL 行为模型	Verilog	从 IP DCP 生成的 Verilog 仿真网表。
	VHDL	VHDL 行为模型。
IP 不提供任何行为模型	Mixed/Verilog/VHDL	从 DCP 生成的采用指定 SIMULATOR_LANGUAGE 的结构仿真网表。

注释: (如果可用)行为仿真 (Behavioral Simulation) 模型始终优先于结构仿真网表 (Structural Simulation netlist)。 Vivado 不提供仿真模型选择。

注释:当 IP 同时支持 Verilog 和 VHDL 时,工程属性 SIMULATOR_LANGUAGE 的设置可用于判定所提供的仿真模型。

为 IP 使用测试激励文件

IP 目录中的许多 IP 还可提供测试激励文件用于单独进行 IP 仿真。如果 IP 可提供测试激励文件,那么在"生成输出文件 (Generate Output Product)"对话框中可看到此测试激励文件列为输出文件,如图 2-17 中所示。

china.xilinx.com



🍋 Generate Output Products	×
The following output products will be generated.	-
Preview	
Q 素 €	
Synthesized Checkpoint (dcp)	
Structural Simulation	
C Simulation	
🗇 Test Bench	
🗅 Change Log	
Synthesis Options	
O <u>G</u> lobal	
Out of context per IP	
Run Settings	
Number of jobs: 6 🗸	
Apply Generate Skip]

图 2-17: 测试激励文件输出文件

要使用 IP 提供的测试激励文件,请在"Sources"窗口的"Simulation Sources"部分的层级中找到 IP 并展开 IP 层级。 "Sources"窗口的"层级 (Hierarchy)"选项卡中的文件如下图所示。

✓ ₽ ir_compiler_0 (20)
> 📄 Instantiation Template (2)
> 🔄 Synthesis (4)
> 📄 Simulation (3)
> 🗁 C Simulation (4)
> 📄 Test Bench (1)
> 📄 Change Log (1)
fir_compiler_0.dcp
fir_compiler_0_sim_netlist.vhd
fir_compiler_0_sim_netlist.v
fir_compiler_0_stub.vhdl
fir_compiler_0_stub.v

图 2-18: 展开 IP 层级

注释: 文件为红色, 因为 OOC 综合尚未运行。

单击"OK"以展开层级。查找并选中名为 tb_<ip_name> 的 IP 测试激励文件,然后右键单击并选中"Set as Top"。



在"Flow Navigator"中,选择"Run Simulation"或者在 Tcl 控制台中使用 launch_simulation 命令在新的设计顶层 (即 IP 的仿真测试激励文件) 启动仿真器。

注释:您可单击"Settings",然后浏览至"Simulation"部分以查看当前仿真设置。您可在此处查看仿真顶层模块名称,此名称应与您设置的 IP 测试激励文件相匹配。您还可在此处更改仿真器设置,这样会影响"Run Simulation"按钮的行为。

升级 IP



注意! 升级 IP 时,将移除所有先前生成的输出文件,包括 DCP 和所有关联的设计运行。作为预防措施,请先将工程归档,然后再升级 IP。

每个 Vivado Design Suite 版本仅提供单一 IP 版本。 Vivado Design Suite 的新版本和补丁可能在 Vivado IP 目录中包含 现有工程中使用的更新版本的 IP。在此情况下,当前工程中的 IP 将被锁定,必须升级才能使用最新版本的 IP。



视频:请参阅《Vivado Design Suite QuickTake 视频:管理 Vivado IP 版本升级》,以查看 IP 升级演示。

迁移到新版本 Vivado Design Suite 前,请执行以下一项或多项操作:

• 为工程中的 IP 生成所有输出文件,包括 DCP。这样您即可根据需要在新版本 Vivado Design Suite 中使用旧版本的 IP。

注释: 您无法为不属于当前版本的 IP 生成输出文件。

- 如果使用"Manage IP"工程,请复制整个"Manage IP"工程位置作为备份。
- 将包含 IP 的设计工程归档。
- 升级 IP 前,请查看"report_ip_status"窗口以获取有关更改的信息。



重要提示:对于不同 Vivado Design Suite 版本之间 IP 存在重大版本更改的 IP 而言,这至关重要,因为这些 IP 通常需 要更改 RTL。

从先前版本的 Vivado 升级工程时,每个 IP 都有 1 个可用升级,您可升级此 IP。下图显示了打开的对话框,此对话框会 询问您是否要禁用"核容器 (Core Container)"功能并继续操作。



?	Core Container is a new format for IP that simplifies the directory structure by storing each IP as a single file on disk. It is possible to revert to
	the old format at any time. For more information on using Core Containers,
	Management and Revision Control section of the UltraFast Design Methodology Guide for the Vivado Design Suite.
	Do you wish to convert your existing IP to Core Containers at this time?
ptio	ns
C	<u>Convert IP to Core Container and Set as Default in Project</u>
۲	Continue with Core Container <u>D</u> isabled
D	on't show this dialog again
	ок

图 2-19: 工程升级与核容器选项对话框

建议:请选择"Report IP Status"以查看哪个 IP 可升级,并查看更改日志。下图显示了运行"Report IP Status"命令时 打开的"IP 状态 (IP Status)"窗口。

下图显示了"IP 状态 (IP Status)"选项卡,其中显示了建议升级 IP 的消息。

	Jes Log Reports	Design Runs I	Status ×			
C	Minor Change (*	1) Hide All				
1	IP Status ^1	Recommendation	Change Log	IP Name	Current Version	Recommended Version
1	IP minor version change	Upgrade IP	More info	Clocking Wizard	5.3 (Rev. 2)	5.4
	C V V	C ✓ Minor Change (* ✓ IP Status ^1 ✓ IP minor version change	C ✓ Minor Change (1) Hide All ✓ IP Status ^1 Recommendation ✓ IP minor version change Upgrade IP	C ✓ Minor Change (1) Hide All ✓ IP Status ^1 Recommendation Change Log ✓ IP minor version change Upgrade IP More info	C ✓ Minor Change (1) Hide All ✓ IP Status ^1 Recommendation Change Log IP Name ✓ IP minor version change Upgrade IP More info Clocking Wizard	C ✓ Minor Change (1) Hide All ✓ IP Status ^1 Recommendation Change Log IP Name Current Version ✓ IP minor version change Upgrade IP More info Clocking Wizard 5.3 (Rev. 2)



单击 IP 旁的"Upgrade IP"选项。此报告可提供有关工程中的所有 IP 的信息。单击任意带下划线的蓝色文本即可提供更多详细信息。这些信息包括:

- 。 "IP Status": 如果 IP 处于最新状态或者如果存在次要或重要版本更改,则显示此项。如果存在器件更改,也 会显示此项。
- **。** "Recommendation":列出建议操作。
- 。 "Change Log":选择"更多信息 (More info)"链接即可查看 IP 的更改日志。更改日志可提供有关 IP 的最新版本的信息。建议您先复查更改日志,然后再升级 IP。
- 。 "IP Name": IP 的名称。
- "Current Version":提供 IP 的当前版本。
- **"Recommended Version"**:提供建议的 IP 升级目标版本。

重要版本更改可能需要修改连接到 IP 的 RTL。"IP Status"窗口包含:

。 "IP Name": IP 目录中所示的 IP 名称。



- 。 "Current Version": 提供 IP 的当前版本。
- **"Recommended Version"**:提供建议的 IP 升级目标版本。
- 。 "License":显示 IP 许可证状态。
- "Current Part":设计中使用的器件。

您可选中"Source File"旁的复选框以选择升级 IP。勾选此列中的复选框将选中有升级可用的所有 IP。单击"Upgrade Selected"以升级所选 IP。

默认情况下,升级 IP 会导致 ip_upgrade.log 文件中存储的信息一并升级,此文件与 XPR 文件位于同一工程目录内。

选择性升级 IP

对于先前版本的 Vivado 中已生成的 IP,支持选择性升级 IP。非升级 IP 的 DCP 以 LOCKED 状态导入,并且不可修改。 此功能支持您与块设计进行交互(即使并未升级所有 IP 也是如此),并且支持您生成比特流,甚至可为部分或全部锁 定 IP 生成比特流(假定锁定 IP 的 OOC DCP 均可用)。

注释: "IP 参数 (IP Parameter)"传输受到限制,请访问此链接并参阅《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》 (UG994) [参照 32] 以获取有关选择性升级块设计中的 IP 的信息。

选择性升级 IP 时, "Report IP Status"窗口如图 2-21 所示:

Tcl Console	ħ	lessa	ges	Log	Reports	Design Runs I	P Status ×			
Q I	¢	C		Mine	or Change (1	I) Hide All				
Source File		$\overline{\mathbf{v}}$	IP S	tatus	~1	Recommendation	Change Log	IP Name	Current Version	Recommended Version
Cik_core		1	IP mi	inor vers	ion change	Upgrade IP	More info	Clocking Wizard	5.3 (Rev. 2)	5.4
							Uc	ograde Selected		

图 2-21: 含选择性升级的 IP 的 IP 状态报告

如需了解有关为新版本更新设计的更多信息,请访问此链接以参阅《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》(UG994) [参照 32]。如需了解有关升级块设计 (BD) 中的 IP 的信息,请访问此链接并参阅《Vivado Design Suite 用户指南:创建和封装定制 IP》(UG1118) [参照 34] 以获取有关编辑已封装的 BD 的信息。

升级设计中的 IP 会创建更新日志。这些日志位于"Sources"窗口中的 IP/Upgrade Log 文件夹下。

此日志包含有关 IP 升级的信息,例如:

- 升级是否成功
- 是否需要用户干预:如果已升级的 IP 需要用户干预,那么此日志会列出遇到的问题,例如参数已不存在。
- IP 的原始版本和修订版
- IP 的升级后版本和修订版
- 其它相应信息:
 - 。 升级脚本执行的更改的描述。此类信息的示例包括: Renamed parameter DATA_W to C_DATA_WIDTH 和 Set parameter BUFFER_LENGTH to value 32。
 - 。 自定义 IP 期间发出的警告,例如,升级期间添加、更改和移除的端口。
 - 。 与升级关联的警告;升级的版本相关的一般问题或者当前参数化相关的具体问题。



。 IP 自定义期间发出的任何警告。

使用 Tcl 命令升级 IP

您可使用 upgrade_ip 命令升级所有指定 IP。例如,要升级设计中的所有 IP,请在 Tcl 控制台中输入 upgrade_ip,如下所示:

upgrade_ip

这样将升级 IP, 但不创建任何升级日志。添加 -log 选项可指定升级日志。



注意! 请勿使用 upgrade_ip [get_ips -all]; 它可能导致 Vivado 出现问题。 -all 选项会返回子核 IP。升级父核 IP 期间可能会移除这些子核 IP,导致无法引用 Tcl 对象。

要升级 IP,并为此 IP 创建日志文件,请在 Tcl 控制台中输入以下命令:

upgrade_ip [get_ips cfifo] -log c:/prj/IP/cfifo_upgrade.log

并非每次 IP 升级都会覆盖升级日志。已升级的最新 IP 将在文件顶部添加升级信息。

了解多层 IP

部分 IP 旨在使用其它 IP 作为设计源。根据父 IP 的创建方式,针对子 IP 可能采用 OOC 流程运行综合。以下是含父子关系的子系统 IP 的类型:

- 引用其它 IP 作为文件库的 IP 或者位于顶层以下的 IP (子核引用):如需了解有关子核引用的信息,请参阅 《Vivado Design Suite 用户指南:创建和封装定制 IP》(UG1118) [参照 34],此类引用仅含单一 OOC 综合运行。在 "IP Sources"中不显示任何子核 IP,仅显示单一 XCI。
- 与子 IP 的 XCI 文件封装在一起的 IP (静态 IP):静态 IP 即与其它 IP 的 XCI 文件封装在一起的 IP。
- 动态创建子 IP 和 HDL 的 IP (动态 IP): 动态 IP 仅含单一 OOC 综合运行,因为所有 IP 都结合在一起进行综合。 与静态 IP 相似,在"IP Sources"中会显示多个 XCI。
- 使用 IP integrator 技术执行 IP 动态创建和互连的 IP (子系统 IP): 子系统 IP 即 IP integrator 技术所创建的 IP。例 如,查看 10G Ethernet Subsystem IP 的综合日志时可以看到针对子 IP 执行黑盒推断。这类似于综合期间用户设计 中的 IP 默认流程。

在"IP Sources"中可以看到子 IP 的 XCI,输出文件位于父 IP 的 Synthesis 文件夹中,如下图所示。

china.xilinx.com





图 2-22: 使用 XCI 文件的子系统 IP 的输出文件

按 IP 执行 OOC 综合期间,完成子 IP 的综合后,会将其链接在一起以便为 IP 创建组合 DCP,如下图所示。这样子系统 IP 看起来与其它 IP 并无两样。

		? _	Пы×
Constraints	Status	Progress	Strategy
	Submodule Runs Complete	I 100%	
axi_10g_ethernet_0	synth_design Completel	I 100%	Vivado Syr
	Submodule Runs Complete	100%	
bd_efdb_0_xmac_0	synth_design Complete!	I 100%	Vivado Syr
bd_efdb_0_xpcs_0	synth_design Complete!	100%	Vivado Syr
bd_efdb_0_prtad_driver_0	synth_design Complete!	I 100%	Vivado Syr
bd_efdb_0_dcm_locked_driver_0	synth_design Completel	100%	Vivado Syr
	Constraints axi_10g_ethernet_0 bd_efdb_0_xmac_0 bd_efdb_0_xpcs_0 bd_ofdb_0_prtad_driver_0 bd_efdb_0_dcm_locked_driver_0	Constraints Status submodule Runs Complete synth_design Complete! axi_10g_ethernet_0 synth_design Complete! bd_efdb_0_xmac_0 synth_design Complete! bd_efdb_0_xpcs_0 synth_design Complete! bd_efdb_0_ptrad_driver_0 synth_design Complete! bd_efdb_0_dcm_locked_driver_0 synth_design Complete!	Constraints Status Progress Constraints Submodule Runs Complete 100% axi_10g_ethernet_0 synth_design Complete! 100% bd_efdb_0_xmac_0 synth_design Complete! 100% bd_efdb_0_xmac_0 synth_design Complete! 100% bd_efdb_0_prtad_driver_0 synth_design Complete! 100% bd_efdb_0_cm_locked_driver_0 synth_design Complete! 100%

图 2-23: 子系统 IP 的输出文件



提示:根据块设计,子系统 IP 的主要优势在于,生成输出文件时,子 IP OOC 运行会并行启动。



"生成输出文件 (Generate Output Products)"对话框支持您在"Run Settings"字段中指定并行运行数量。

此外,如果启用"IP 高速缓存 (IP Caching)",层级 IP 可包含子 IP 的缓存命中。这样即可显著加快生成速度。

处理调试 IP

Vivado Design Suite 包含支持您对器件中实现后的设计执行系统内编程和调试的功能。在系统内调试设计的好处是能够在以系统速度运行的实际系统环境内对实现后时序准确的设计进行调试。

您可以使用 Vivado Lab Edition 工具来测试和验证通过 JTAG 连接附加到赛灵思开发板上的 IP 功能。可用调试 IP 核包括:

• "Vivado Integrated Logic Analyzer": Integrated Logic Analyzer (ILA) 也称为 Vivado Logic Analyzer, 它支持您 对 FPGA 上实现后的设计执行系统内调试。

如需监控设计中的信号,可使用此功能。另外,您还可以使用此功能触发硬件事件并以系统级速度采集数据。您可在自己的 RTL 代码中例化 ILA 核,或者在综合后在 Vivado 设计流程中插入该核。

• "Vivado Virtual I/O Analyzer": Virtual Input/Output (VIO) 调试功能也称为 Vivado Serial I/O Analyzer,可实时 监控并驱动内部 FPGA 信号。如果无法通过物理方式访问目标硬件,则可使用此调试功能来驱动并监控真实硬件 上存在的信号。

此调试核必须在 RTL 代码中例化;因此您需要明确了解要驱动的信号线。

• "IBERT Serial Analyzer":集成式误码率测试器 (IBERT) Serial Analyzer 可启用系统内串行 I/O 验证和调试。这样 您即可在基于 FPGA 的系统内对自己的高速串行 I/O 链路进行测量和最优化。

LogiCORE IBERT Serial Analyzer 可用于解决各种系统内调试和验证问题,从简单的时控和连接问题到复杂的裕度分析和通道最优化问题都无一例外。

对接收到的信号应用接收器均衡后,即可使用该核来测量信号质量。这样可确保在发射到接收通道中的最优点执行测量,从而确保获取真实准确的数据。

可生成设计示例,用于对 IBERT 核进行任意自定义设置。完成自定义并生成核实例后,右键单击生成的核并选择对 应该核的"Open IP Example Design"功能。

• "JTAG to AXI": JTAG-to-AXI 调试功能可生成 AXI 传输事务,以便与硬件中运行的系统内各 AXI4 和 AXI4-Lite 从 核进行交互。

该核可用于生成 AXI 传输事务和调试,并在运行时间驱动 FPGA 内部的 AXI 信号。您还可在无处理器的 IP 设计中使用该核。 IP 目录在"Debug"类别下列出了该核。

如需了解更多信息,请参阅下列参考文档:

- 。 《IBERT 7 系列 GTX 收发器 LogiCORE IP 产品指南》 (PG132) [参照 3]
- 。 《IBERT 7 系列 GTP 收发器 LogiCORE IP 产品指南》 (PG133) [参照 4]
- 。 《IBERT 7 系列 GTH 收发器 LogiCORE IP 产品指南》 (PG152) [参照 6]
- 。 《Integrated Logic Analyzer LogiCORE IP 产品指南》 (PG172) [参照 8]
- 。 《JTag to AXI LogiCore IP 产品指南》 (UG174) [参照 9]
- 。 《Vivado Design Suite 用户指南:编程和调试》(UG908) [参照 23]
- 。 《Vivado Design Suite 教程:编程和调试》(UG936) [参照 27]
- 。 《UltraFAST 设计方法指南 (适用于 Vivado Design Suite)》 (UG949) [参照 30]



调试流程

Vivado 工具提供了多种方法用于在设计中添加调试探针。您需要判定最适合您的设计要求的流程。可用调试流程如下:

• "HDL instantiation debug probing flow":此流程包括将调试 IP 核明确添加到您的 HDL 设计中并将 HDL 源代 码中的信号附加到 ILA 调试探针中。

请访问此链接并参阅《Vivado Design Suite 用户指南:编程和调试》(UG908) [参照 23] 以获取有关此流程的更多 信息。

此流程的优缺点如下:

- 。 优点:提供在 HDL 设计层级进行探测的功能。
- 。 缺点:
 - 您必须在设计中修改自己的 HDL 源代码,手动添加和移除调试信号线和 IP。
 - 生成、例化和连接调试核时很容易出错。
- "Netlist insertion debug probing flow" (推荐):此流程包括将已综合的网表中的信号明确附加到 ILA 调试核 实例:
 - 。 使用 MARK_DEBUG 属性来标记 RTL 源代码中要调试的信号。
 - 。 使用 MARK_DEBUG 右键单击菜单选项从已综合的设计网表中选择要调试的信号线。

网表插入流程使用 Set up Debug wizard 来指导您完成添加调试核和探测设计信号的整个过程。

- 。 优点:
 - 灵活性最高,有良好的预测能力。
 - 便于在各个不同设计层级 (如 HDL、综合设计、系统设计)进行探测。
 - 无需修改 HDL 源代码。
 - 缺点:无法用于 IBERT 或 JTAG-to-AXI Master 核。
- **"Tcl-based netlist insertion flow"**:使用 set_property Tcl 命令在调试信号线上设置 MARK_DEBUG 属性,然 后使用以下 Tcl 命令来向已综合的设计添加调试核和探针:
 - create_debug_core
 - create_debug_port
 - connect_debug_port

使用核容器

"核容器 (Core Container)"功能通过提供 IP 的单一文件表示法来帮助简化版本控制系统的使用。

注释:不推荐采用二进制文件进行版本控制。建议将 XCI 置于核容器之外,这样您即可比较版本控制环境内的 IP,查看是否存在任何变动。

set_property coreContainer.alwaysCreateXCl 1 [current_project]



视频:如需了解更多信息,请参阅《Vivado Design Suite QuickTake 视频:使用面向 IP 的 Core Container 功能》。

此可选功能支持您选择将 IP 和所有生成的输出文件都包含在单一压缩二进制文件(扩展名为 XCIX)内。此扩展名类 似于用于 IP 自定义文件的 XCI 文件,其作用方式也是相似的。



注释: 在此流程中,针对"Upgrade IP", IS_MANAGED 不得设置为 true。要升级 IP,请使用以下 命令:

[get_ips {<ip_name>}]

添加或读取 IP 时,指定 XCI 文件,如果已启用核容器,则请添加或读取 XCIX 文件。

为现有 IP 启用核容器功能时,XCIX 文件会替换 IP 目录和输出文件。为 IP 禁用核容器时,XCIX 文件将还原为包含所有输出文件 (包括 XCI 文件)的 IP 目录。

为 IP 启用核容器会更改在磁盘上显示此 IP 实例的方式;在 Vivado 内部显示的 IP 则保持不变。

在 Vivado 的"Sources"视图中显示的 2 个 IP 相同,两者均列出输出文件,并且所有输出文件均可从 Vivado IDE 内部打 开以供查看。

在磁盘上存在 clk_core 文件夹以及对应 char_fifo IP 的 XCIX 文件。当 IP 使用 Core Container 时, Vivado 工具会 从该 XCIX 文件读取综合和实现期间所需的 IP 源文件。这些文件不会解压至临时目录,而是直接从二进制文件中读取。



重要提示:重申,以单一文件来表示 IP 可简化版本控制。

仅使用 XCIX 文件可为您提供与使用 XCI 文件相同的功能,例如:

- 。 报告 IP 状态和详情 (Report IP Status)
- 。 有新版本可用时升级 IP,以及能够根据需要使用旧版本的 IP
- 。 能够重新自定义 IP (前提是使用最新版本的 IP)
- 。 能够复位和重新生成 IP (前提是使用最新版本的 IP)

无论是否使用核容器功能,在工程中所显示的 IP 视觉效果并无变化。"IP Sources"选项卡会显示针对 IP 已生成的所有文件,并且您仍然可以打开这些文件进行查看。并且,无论您是否使用核容器, IP 相关的 Tcl 命令都保留不变。

就像所有 IP 一样,某些文件存储在 ip_user_files 目录中以便于使用。如需了解更多详情,请参阅"用于核容器的 IP 用户文件 (ip_user_files)"。

启用和禁用核容器

默认情况下禁用"核容器 (Core Container)"功能。要使新创建的 IP 使用 Core Container 功能,请转至"Settings > IP", 并勾选"Use Core Containers for IPs"。



视频: 请参阅 《Vivado Design Suite QuickTake 视频: 使用面向 IP 的 Core Container 功能》以获取更多信息。

注释: 7 系列 Memory Interface IP 不支持 Core Container 功能。此外,位于 IP integrator 块设计内部的 IP 也无法使用 Core Container 功能。

如果适用 Core Container, 该功能的启用方法如下:

- 要为所有 IP 使用 Core Container 格式,请选择"Settings > IP",然后勾选"Use Core Container"选项。
- 如果希望现有 IP 使用 Core Container 格式,请从"IP Sources"中选中相应的 IP,右键单击并选择"Enable Core Container"。

要为使用 Core Container 的 IP 禁用该功能,请从"IP Sources"选项卡中选中相应的 IP, 然后选择"Disable Core Container"。

将 Vivado 工程从旧版本升级到最新版本时,如果检测到 IP,则"启用核容器 (Enable Core Container)"对话框会打开并 提示您选择"Convert IP to Core Container and Set as Default in Project"选项即可使用 Core Container 功能。



利用核容器进行仿真

启用 Core Container 后, Vivado 工具会在生成 IP 期间将 IP 的仿真相关文件存储在 XCIX 文件外部以便于用户使用。

如果只有 XCIX 可供使用,则会使用 export_ip_user_files Tcl 命令解压这些文件。如需了解更多信息,请访问此 链接以参阅 《Vivado Design Suite 用户指南:逻辑仿真》 (UG900) [参照 20]。

使用 Core Container 的 IP 的行为仿真文件存储在 sim 目录中,此目录位于以下两处位置的其中一处:

- 使用工程时,仿真文件存储在: <project_directory>\<project name>.ip_user_files\ip\<ip_name>\
- 使用"Managed IP"工程创建 IP 时, 仿真文件存储在: <managed_ip_project directory>\ip_user_files\ip\<ip_name>\

对于第三方仿真器,如果 IP 仅提供单一语言的行为仿真且该语言不受支持,则会在以上列出的位置中提供下列功能仿 真文件:

- 。 <ip_name>_sim_netlist.vhdl

注释: 在早于 2015.3 的 Vivado Design Suite 版本中, 仿真文件为 named *_funcsim.v 和 *_funcsim.vhdl。

如需获取有关生成的 IP 仿真相关文件及其位置的更多详细信息,请参阅"仿真 IP"。

核容器的支持文件

就像仿真文件一样,在生成 IP 期间还会解压使用 Core Container 的 IP 的其它支持文件。这些文件包括:

- 上述"利用核容器进行仿真"部分中所述的仿真文件。
- 对应 Verilog、 System Verilog 和 VHDL 的例化模板 (.veo 和 .vho)
- 在第三方综合工具中用于为 IP 推断黑盒的存根文件 (*_stub.v 和 *_stub.vhdl)

这些支持文件位于以下两处位置的其中一处:使用工程时,这些文件位于 <project_directory>\<project name>.ip_user_files\ip\<ip_name>\。使用"Managed IP"工程时,这些文件位于: <managed_IP_project_directory>\ip_user_files\ip\<ip_name>

用于导出支持文件的 Tcl 命令

生成 IP 期间,无论是否使用 Core Container 功能,都会自动将支持文件置于前述"利用核容器进行仿真"和"核容器的支持文件"部分中所述的位置中。

如果仅有 XCIX 文件并且希望导出支持文件,请在 Tcl 控制台中输入:

export_ip_user_files -of_objects [get_ips <ip_name>]

如果省略该选项,那么无论您当前是否使用 Core Container, Vivado 工具都会再次导出设计中的所有 IP 文件。在此情况下,将从 XCIX 文件中提取 Core Container 的所有文件。对应基于 XCI 的 IP,将从 IP 目录复制这些文件。

china.xilinx.com

E XILINX_®

用于核容器的 IP 用户文件 (ip_user_files)

生成 IP 输出文件期间,部分文件将被自动复制到名为 ip_user_files 的特殊目录中以便于使用。使用 Core Container 功能时,这很有用(请参阅"使用核容器")。IP 支持文件存储在 ip_user_files 目录下便于取用的位置。

此目录结构允许您使用 Core Container 功能访问 IP 的例化模板和仿真文件,而无需从二进制容器手动解压文件。无论 您是否使用 Core Container 功能, IP 支持文件都存储在 ip_user_files 目录中。

创建 IP 自定义 (XCI) 文件时, Vivado IDE 会创建 1 个目录,其名称与包含 IP 定义和输出文件的 IP 的名称相同。附录 B: IP 文件和目录结构对这些文件进行了描述。

如果您选择使用 Core Container 功能,那么 Vivado IDE 会创建 XCIX 二进制文件,其中包含 IP 的所有文件(请参阅"使用核容器"以获取详细信息)。

此目录位置取决于 IP 是在 RTL 工程中创建的还是在"Managed IP"工程中创建的,如下所述:

- 。 RTL 工程: <directory to project>/<project name>/ip_user_files/
- 。 "Managed IP"工程: <managed_ip_project_directory>/ip_user_files/

在 ip_user_files 目录中包含多个文件夹,具体所含文件夹取决于工程中包含的内容 (IP、块设计等)。

以下是可能存在的每个目录的简要描述。在本章节中以及附录 B: IP 文件和目录结构中都对每个目录进行了更加详细的讲解。

- bd: 对应工程中每个 IP integrator 块设计 (BD) 包含 1 个子文件夹。这些子文件夹包含所使用的 IP 的支持文件。
- 。 ip: 包含工程中存在或者在"Managed IP"工程中创建的每个 IP 自定义 (XCI/XCIX) 文件所特有的文件。
- ipstatic: 包含来自工程中所有 IP/BD 的公用 IP 静态文件。
- mem_init_files: 如有任何 IP 提供数据文件,就会存在此目录。
- sim_scripts: 默认情况下,针对存在的每个 IP 和每个块设计都会创建对应选定操作系统的所有受支持的 仿真器的脚本。

无论您是否使用 Core Container 功能, Vivado Design Suite 都会创建这些文件和目录。根据是否使用该功能,这些文件存在于 XCIX 二进制文件中或 IP 目录中。

要将 IP/BD 文件手动导出到 ip_user_files 目录中,可在 Tcl 控制台中使用 export_ip_user_files 命令。复位 或生成 IP 或 BD 时,此命令会自动运行。

bd 目录的内容

只要工程包含 1 个或多个 IP integrator 块设计,就会存在 bd 目录。每个 BD 都有一个专用子文件夹,其中包含所使用 的 IP 的支持文件。

存在的3个目录分别是:

- 。 hdl: 块设计的仿真顶层文件。
- 。 ip: BD 中的每个 IP 都有 1 个目录,其中包含仿真文件。
- ipshared: BD 中存在的 IP 之间公用的仿真文件。

如果生成期间按 BD 选择了非关联 (OOC) 流程,那么在 bd 目录中相应的块设计子文件夹下将包含存根文件。

ip 目录的内容

ip 目录包含工程中存在的 IP 的支持文件。这些文件置于根据 IP 名称命名的子目录中。支持文件包括:

• 名为 sim 的子文件夹中存在的仿真文件 (仅限 Core Container - 请参阅"利用核容器进行仿真")。



- 对应 Verilog 和 VHDL 的例化模板文件 (.veo 和 .vho)
- 在第三方综合工具中用于为 IP 推断黑盒的存根文件 (*_stub.v 和 *_stub.vhdl)

支持文件也位于 IP 目录中。为方便使用并与使用 Core Container 功能的 IP 保持一致,用户可能需要的文件副本也置于 ip_user_files 目录中。

ipstatic 目录的内容

有许多 IP 共享用于仿真的文件,这些文件无需随每次自定义进行更改。ipstatic 目录包含适用于工程中的所有 IP 和 BD 的共享文件。针对仿真创建的脚本会根据需要引用此目录中的这些文件。 IP 提供的动态仿真文件位于 IP 自定义目 录中。使用 Core Container 功能时,动态仿真文件位于 ip 目录中。请参阅"ip 目录的内容"。

mem_init_files 目录的内容

部分 IP 会提供数据文件。这些文件以 DATA 属性来标记。这些文件存储在 mem_init_files 目录中。可能存在的文件标记为 data,包括存储器初始化文件 (MIF) 和文本文件 (TXT)。

sim_scripts 目录的内容

针对每个 IP 和 BD 仿真都会创建脚本。默认情况下, Vivado Design Suite 会为生成 IP 的操作系统所支持的所有仿真器 生成脚本。

对于 Microsoft Windows, 这包括:

- Vivado 仿真器
- Mentor Graphics ModelSim 仿真器
- Mentor Graphics Questa Advanced 仿真器
- Riviera-PRO 仿真器
- Active-HDL 仿真器 (仅限 Windows)

对于 Linux,还包括下列仿真器:

- Synopsys Verilog Compiler Simulator (VCS)
- Cadence Incisive Enterprise Simulator (IES)

要控制脚本生成,请参阅"IP 设置"。生成的脚本会从 IP 自定义目录中引用仿真文件。对于使用 Core Container 功能的 IP,这些脚本会引用"sim_scripts 目录的内容"目录中的仿真文件。对于块设计中的 IP,这些脚本会引用"bd 目录的内容"中的仿真文件。

用于导出 IP 支持文件的 Tcl 命令

生成 IP 期间,无论是否使用 Core Container 功能,都会自动将支持文件置于指定位置。请参阅"用于导出支持文件的 Tcl 命令"。



第3章

使用"管理 IP"工程

引言

Vivado® 集成设计环境 (IDE) 可提供下列机制:

- 。 浏览 IP 目录中的 IP
- 。 自定义 IP
- 。 管理自定义 IP 的集中存储位置。

Vivado IDE 可以创建特殊工程用于管理指定 IP 的自定义和输出文件,此工程称为"管理 IP (Manage IP)"工程。在 "Manage IP"工程中,您可查看 IP 目录 (IP catalog)、自定义 IP 和生成输出文件。IP 自定义 (XCI) 文件和生成的输出文件 存储在位于"Manage IP"工程外部的专用目录中。"Manage IP"工程负责管理 IP 设计运行,以便生成综合后的设计检查 点 (DCP) 文件和其它输出文件。自定义 IP (包括生成的所有输出文件)可在多项设计中作为预配置的 IP 来使用。请参 阅"将现有 IP 添加到工程中"。

以团队协作方式开展工作时或者如果设计使用多个赛灵思 IP,那么请在 Vivado 工程结构外部某处位置创建并维护您的 自定义 IP。此方法便于版本控制,并且便于与他人共享自定义 IP。这也是在基于脚本的非工程流程中处理 IP 时推荐使 用的方法。

托管 IP 功能

在 Vivado IDE 中使用"Manage IP"流程时,可利用下列特性:

- 。 简单的 IP 工程界面
- 。 直接访问赛灵思 IP 目录
- 。 能够自定义多个 IP
- 。 针对每个 IP 自定义文件 (包括所有相关 IP 文件)提供独立专用目录
- 。 可选择生成或者跳过生成设计检查点 (DCP) 文件。单一 DCP 文件包含 IP 的网表和约束,并且创建此文件包含 在默认流程中。



视频: 另请参阅:《Vivado Design Suite QuickTake 视频: 在 Vivado 中配置和管理可复用 IP》和《Vivado Design Suite QuickTake 视频: 使用设计检查点》。



重要提示: AXI Peripheral IP 不适用于"Managed IP"工程。如果您尝试使用此 IP, Vivado GUI 将发出错误。如需 AXI Peripheral IP,请使用常规 Vivado 工程。



使用管理 IP 流程

要使用"管理 IP (Manage IP)"流程,请调用 Vivado IDE,在"Getting Started"页面中,选中"Manage IP"。



图 3-1:调用管理 IP 流程

打开新 IP 或现有 IP 位置。

- 。 "New IP Location": 打开位于指定位置的新 IP 工程以便浏览 IP 目录并自定义 IP,包括生成输出文件。
- 。 "Open IP Location": 支持您浏览到现有位置以便从中打开 IP。
- 。 "Recent Projects": "Getting Started"页面右侧列出了最近使用的"Manage IP"工程的打开位置。

选中"New IP Location"选项时,"Create a New Customized IP Location"菜单会显示向导以指导您完成创建和管理新的自定义 IP 位置的过程,如下图所示。



图 3-2: 创建新的自定义 IP 位置

如果指定用于创建新的"Manage IP"工程的位置已包含工程,则会打开 1 个对话框 (请参阅下图)。



🍋 New	/ IP Location	
?	An IP project already exists at this location. Would you like to open the existing project?	
	Open Cancel	

图 3-3: 新建 IP 位置对话框

您可以选择打开现有工程或者取消,然后单击"Next"。

这样会打开"管理 IP 设置 (Manage IP Settings)"对话框,如下图所示。

t options for creating ar	d generating IP.	1
Part:	xc7z100ffg900-2 (active)	
Target language:	Verilog	~
Target simulator:	Vivado Simulator	~
Simulator language:	Mixed	~
IP location:	C:/Xilinx/lab_1	•

图 3-4: 管理 IP 设置对话框

管理 IP 设置

要管理 IP 设置,请输入以下信息,然后单击"Finish":

- 。 "Part":选择处于活动状态的器件。针对 IP 生成的所有输出文件都基于指定器件。
- 。 "Target language":将目标语言设置为设计的顶层模块的语言。
- 。 "Target Simulator": 指定要使用的仿真器,可选 Vivado 仿真器或下列任一第三方仿真器。
- 。 "Simulator language": 根据可用于仿真器的许可证,选项包括"VHDL"、"Verilog"、"SystemVerilog"或 "Mixed"。对于 Vivado 仿真器,默认选项为"Mixed"。
- 。 "IP location": Vivado IDE 创建 managed_ip_project 目录的位置。

注释: IP 位置可被称为 IP 存储库。



Vivado IDE 会打开"Managed IP"工程,这样您即可选择和自定义 IP。您可以访问整个 IP 目录,包括 IP 产品指南、更改 日志、产品网页和答复记录。

完成自定义 IP 后,会显示"Sources"窗口和"Properties"窗口以提供工程中创建的 IP 的相关信息。

每个 IP 自定义都包含 1 个创建于指定管理 IP 位置下的目录。此目录包含赛灵思定制接口 (XCI) 文件和任何生成的输出 文件。下图显示了可供您在其中自定义和管理多个 IP 的"Manage IP"工程窗口。

Manage IP - [C:\Xilinx\Workspace\iprepo] - Vivado	2017.1							
File Edit Tools Window Layout View Help	Q- Quick Access							
							🔚 Default Layou	nt 🗸
PROJECT MANAGER - xcku040-ffva1156-2-e								2 X
Sources ? _ D	IP Catalog						?	
Q ≚ ≑ +	Cores Interfaces							
✓ E IP (3) → E → aut 40a, attempt 0.704)		🎤 🖉 🐵 🔳 🔍						۰
> The birds speed selection with 0 (18)	Name			^ 1 AXI4	Status	License	VLNV	
> 🖓 🖬 mgn_speed_selectio_wiz_0 (10)	🗸 🚍 Vivado Repositor	У						<u>^</u>
	> 🕒 Alliance Partn	ers						
	> 🗁 Automotive &	Industrial						
	> 🗁 AXI Infrastruct	ure						
	> 🖨 AXIS Infrastru	cture						
Hierarchy IP Sources	🗸 🖨 BaselP							
Course File Development	P Multiply Ad	der			Production	Included	xilinx.com:ip:xb	pip_multad
Source File Properties ? _ U	⊇ × 👎 oddr				Production	Included	xilinx.com:ip:o	ddr:1.0
₱ high_speed_selectio_wiz_0.xci	Utility Buffe	r			Production	Included	xilinx.com:ip:ut	til_ds_buf::
Chabled Location: C:/Xilinx/Workspace/iprepo/high_speed_sel Type: IP Part xcku/040-fiva1156-2-e Ceneral Properties IP	Details	s	select an IP or	Interface or Reposit	tory to see details			
Tcl Console Messages Log Design Runs X							2	B
Name	Constraints	Status	Progress	Strategy		Part	Г	Description
		-		57			-	
> 🖌 axi_10g_ethernet_0		Submodule Runs Complete	1 00%					
✓ rxaui_0_synth_1	rxaui_0	synth_design Complete!	1 00%	Vivado Synthesis I	Defaults (Vivado Synthesis 2017)	xcku040-	ffva1156-2-e V	/ivado Synthe
high_speed_selectio_wiz_0_synth_1	high_speed_selectio_wiz_0	synth_design Complete!	100%	Vivado Synthesis I	Defaults (Vivado Synthesis 2017)	xcku040-	ffva1156-2-e V	/ivado Synthe
< c				Completin	g preparation of all runs for			> Cancel

图 3-5: 包含 3 个 IP 的"Manage IP"工程窗口



注意! 创建新的 AXI4 外设时,无法通过 AXI4 VIP 和 JTAG 接口执行验证。要使用此外设验证,必须使用该外设创建 Vivado 工程。



第 4 章

使用 IP 设计示例

引言

许多赛灵思 IP 都提供了设计工程示例。设计工程示例由顶层逻辑和与所创建的 IP 自定义进行交互的约束组成。这些设计示例通常随附有助于设计仿真的测试激励文件示例。

?

提示: 您无需使用最新 IP 来将现有设计示例升级至新版本,只需在新版本中利用该 IP 创建新的设计示例即可。这样可确保设计示例经调整后可支持最新版本的 IP。

打开设计示例

要在标准工程或"Manage IP"工程中打开 IP 的设计工程示例,请选择"IP Sources"选项卡中的 IP 自定义,然后从上下文 菜单中选择"Open IP Example Design"。

这样会打开"Open IP Example Project"对话框,以便您指定位置,如图 4-1 中所示。此工程名为 <ip_name_ex>。



重要提示:请勿将设计示例存储在标准工程或"Manage IP"工程下的 IP 目录中。赛灵思建议将整个 IP 目录置于版本控制中,并建议您不要将工程置于版本控制中。否则启用和禁用核容器时也会导致出现问题。

在 IP integrator 块设计中,选择"IP Sources"中的 IP,或者从块设计直接访问 IP。

下图显示了"打开 IP 工程示例 (Open IP Example Project)"对话框。

Open IP Example Design		_
Specify a location where the example proje	ct directory "char_fifo_ex	will be placed.
Location		
Put example project directory here:	C:/Xilinx	
✓ Overwrite existing example project	t	
		K Cancel

图 4-1: 打开 IP 设计示例对话框



"新工程摘要 (New Project Summary)"对话框可供您复查自己的选择,并且还会随设计示例打开新的 Vivado IDE 会话, 如下图所示。



图 4-2: 含约束文件的 IP 设计实例示例

在设计示例中使用 XDC 约束文件示例来对此 IP 进行例化,以便进一步评估此 IP。

用于打开工程的 Tcl 命令

您也可以选择使用 open_example_project Tcl 命令来打开工程:

```
open_example_project [get_ips <ip_name>]
```

检验独立 IP

使用已综合的 IP 时,实现完成后,会打开"Implementation Completed"对话框,其中提供了"打开已实现的设计 (Open Implemented Design)"、"生成比特流 (Generate Bitstream)"或"查看报告 (View Reports)"的选项,如下图所示。

Implementation Completed				
Implementation successfully completed.				
Open Implemented Design				
◯ <u>G</u> enerate Bitstream				
◯ <u>V</u> iew Reports				
Don't show this dialog again				
OK]			

图 4-3:打开已实现的 IP



执行时序分析时,由于时钟尚未布线,且使用的是理想时钟,因此结果并不准确。执行保持时间分析时,这最为明显,因为布线器无法修复保持时间违例。

部分 IP 的 <ip_name>_ooc.xdc 文件中包含 HD.CLK_SRC 属性,该属性可提供时钟缓存位置以及斜率 (SLEW) 定时 器模型,用于提升实现后时序分析的准确性。



重要提示:已实现的 IP 仅用于分析,在实现项层设计期间,不使用也不保留这些结果。如需了解有关模块复用和使用 IP 的已实现版本的信息,请参阅《Vivado Design Suite 用户指南: 层级设计》(UG905) [参照 22]。



第5章

将赛灵思 IP 与第三方综合工具配合使用

引言

赛灵思支持将第三方综合工具创建的网表用于用户逻辑。使用赛灵思 IP 时,唯一受支持的综合工具是 Vivado® 综合工 具。 Vivado Design Suite 会在逻辑综合期间为赛灵思 IP 推断黑盒。如前述章节中所述,将提供 1 个文件用于推断黑 盒。 Vivado Design Suite 会在实现期间解析黑盒。仅支持采用 Vivado 综合工具执行赛灵思 IP 综合,包括 IP 核以及 IP 可提供的任意设计文件示例。



视频:请参阅《Vivado Design Suite QuickTake 视频:配合使用第三方综合工具和 IP》以获取更多信息。



重要提示:赛灵思根据《IEEE 建议的电子设计知识产权 (IP)加密与管理实践》(IEEE P1735标准)对 IP HDL 文件进行加密。因此,仅当使用 Vivado 综合时, IP HDL 文件才可用。您可将第三方综合工具用于最终用户逻辑,并生成可供 Vivado 实现使用的网表。支持使用第三方仿真工具来利用已加密的 RTL 执行行为仿真。

第三方综合流程

使用 Synopsys® Synplify Pro 或 Mentor® Graphics Precision 网表对包含赛灵思 IP 的设计进行综合时,建议的流程是使用"Manage IP"流程创建和自定义 IP,并为 IP 生成输出文件,包括每个 IP 的综合设计检查点 (DCP) 文件。

生成 DCP 文件时,会创建存根文件以配合下列第三方综合工具用于推断黑盒: <ip_name>_stub.v 和 <ip_name>_stub.vhdl。

Verilog 或 VHDL 存根文件可添加到工程中以供第三方综合工具使用。 Verilog 或 VHDL 存根文件会在综合期间推断黑 盒,还会阻止综合工具添加 I/O 缓存。

<ip_name>_stub.v和 <ip_name>_stub.vhdl 包含综合指令,如果 IP 连接到顶层端口,那么这些指令可阻止第 三方综合工具为 IP 推断 I/O 缓存。您可根据需要更改这些指令,以配合第三方综合工具使用。

第三方综合工具可为顶层设计生成网表。

注释: 请参阅《Vivado Design Suite 用户指南:系统级设计输入》 (UG895) [参照 18] 以获取有关网表工程的更多信息。

创建 Vivado 网表工程以对顶层设计进行布局布线,并为器件生成比特流。

您还可通过以下步骤来为设计创建 RTL 工程、将来自第三方综合工具的 EDIF 网表封装在封装器中,以及实现设计:

- 1. 围绕由第三方综合工具所生成的 EDIF 网表创建 HDL 封装器。
- 2. 选择"Sources"窗口的"Hierarchy"选项卡。
- 3. 右键单击并选择"Hierarchy Update",然后选中"No Update, Manual Compile Order"选项。



- 4. 将以下内容添加到 Vivado 网表工程中:
 - 。 来自第三方综合工具的网表
 - 。 用户层顶层设计约束
 - 。 对应 IP 的 XCI 文件 (每个 IP 1 个 XCI 文件)

使用对应 IP 的 XCI 文件时,实现期间将自动使用 IP DCP 中的网表以及 XDC 输出文件。

5. 实现设计。

如果 IP 的 DCP 中不存在 I/O 缓存, Vivado 实现会添加所有必需的 I/O 缓存。

 \bigcirc

建议:在工程模式或非工程模式下引用赛灵思 IP 时,请使用 IP XCI 文件,而不是直接使用 DCP 文件。虽然 DCP 确实 包含约束,但其解析与最终用户约束无关联。使用 XCI 会导致在组合所有网表 (最终用户和 IP)后应用 IP 的 XDC 输 出文件。此外,随后 IP XDC 中的任何 Tcl 脚本的评估都将与最终用户约束和网表相关联。

用于工程模式下第三方综合的 Tcl 脚本示例

Create a project on disk
create_project <name> -part <part>

configure as a netlist project
set_property design_mode "GateLvl" [current_fileset]

Add in the netlist from third-party synthesis tool add_files top.edif

Add in XCI files for the IP add_files {ip1.xci ip2.xci ip3.xci}

Add in top level constraints: this might include XDC files from the third-party
synthesis tool
add_files top.xdc
Launch implementation
launch_run impl_1 -to write_bitstream

用于非工程模式下第三方综合的 Tcl 脚本示例

Set target part set_part <part> # Read the netlist from third-party synthesis tool read_edif top.edif # Read in the IP XCIs read_ip ip1.xci read_ip ip2.xci # read in top level constraints read_xdc top.xdc # Implement the design link_design -top <top> opt_design place_design phys_opt_design

route_design



write_bitstream -file <name>

注释:请确保在读入 IP 时,从原先生成 IP 的输出文件的位置读取 XCI 文件,或者也可以读入 XCI 文件,然后使用 synth_ip 命令生成 IP。



第6章

用于常见 IP 操作的 Tcl 命令

引言

本章涵盖了用于常用 IP 操作的 Tcl 命令。

如需了解有关使用 Tcl 和 Tcl 脚本编制的更多信息,请参阅:

- 。 《Vivado Design Suite 用户指南:如何使用 Tcl 脚本》 (UG894) [参照 17]
- 。 《Vivado Design Suite Tcl 命令参考指南》 (UG835) [参照 13]

如需获取演示如何在 Vivado 工具中使用 Tcl 的分步教程,请参阅《Vivado Design Suite 教程:设计流程简介》(UG888) [参照 14]。



重要提示: 使用 Tcl 命令关联文件时,请确保此文件的路径为绝对路径而不是相对路径。

重要提示:在非工程模式下使用 HLS IP 时,请务必先运行 compile_c 命令,然后再运行综合。

在设计流程中使用 IP Tcl 命令

通常,工程模式流程与非工程模式流程之间常用的 IP Tcl 命令是一致的,仅在设置用于 IP 创建和综合的器件方面存在 些许例外。下表所列的 Tcl 命令按其在设计中的使用顺序排序。

表 6-1: 按设计	使用顺序排序的 IP Tcl 命令	Ŷ
------------	-------------------	---

操作	工程模式命令	非工程模式命令
设置用于 IP 创建的 器件	不适用。器件属于工程设置。	 # Set target part set_part <part></part> 注释:在内存中而不是在磁盘上创建工程。包含器件选项的命 令 (例如, synth_design),然后使用指定器件。
创建 IP 自定义	<pre>create_ip <ip_name></ip_name></pre>	create_ip <ip_name></ip_name>





表 6-1: 按设计使用顺序排序的 IP Tcl 命令 (续)

操作	工程模式命令	非工程模式命令
升级 IP	<pre>upgrade_ip <ip_name></ip_name></pre>	upgrade_ip <ip_name></ip_name>
		 注意!请勿使用 upgrade_ip [get_ips -all]; 它可能导致 Vivado 出现问题。 -all 选项会返回子核 IP。升级父核 IP 期间可能会移除这些子核 IP,导致无法引用 Tcl 对象。
配置 IP 自定义	<pre>set_property \ CONFIG.Input_Data_Width 8 \ [get_ips <ip_name>]</ip_name></pre>	<pre>set_property \ CONFIG.Input_Data_Width 8 \ [get_ips <ip_name>]</ip_name></pre>
创建目标时钟周期	如果支持,请使用 IP 自定义 GUI。如果不支持,请使用 Tcl 命令,如非工程模式中所示。	set_property \ CONFIG. <clock_name>.FREQ_HZ \ <#>[get_ips char_fifo] 请参阅"使用 Tcl 命令设置目标时钟周期"。</clock_name>
生成输出文件	generate_target all \ [get_ips <ip_name>] 注释: (可选)您可指定要生成的目标。</ip_name>	generate_target all \ [get_ips <ip_name>] 注释: (可选)您可指定要生成的目标。</ip_name>
综合 IP 以创建 OOC DCP	<pre>create_ip_run \ [get_ips <ip_name>] launch_runs <ip_name>_synth_1</ip_name></ip_name></pre>	<pre>synth_ip [get_ips <ip_name>]</ip_name></pre>
读取 IP	<pre>将 IP 随任意输出文件一起复制到工程中: import_files <ip_name>.xci 将 IP 随任意输出文件一起添加到工程中,并从 指定位置引用。使用以下任一命令: add_files <ip_name>.xci read_ip <ip_name>.xci</ip_name></ip_name></ip_name></pre>	 读取 IP 以及任意生成的输出文件。使用以下任一命令: add_files <ip_name>.xci read_ip <ip_name>.xci</ip_name></ip_name> 注释:不同于工程模式流程,不会自动生成输出文件。您必须使用 generate_target 命令生成这些输出文件。如果使用 synth_ip 命令来为 IP 生成 DCP,则无需首先 生成输出目标;这些目标会自动生成。
文件查询	get_files -of_objects \ [get_ips <ip_name>]</ip_name>	get_files -of_objects \ [get_ips <ip_name>]</ip_name>
仿真	请参阅"仿真 IP"。	请参阅"提供 IP 仿真模型"。
调试	请参阅"调试流程"。	
IP 定义		report_property -all [get_ipdefs <ipvlnv></ipvlnv>
IP 创建	write_ip_tcl 从 Tcl 脚本写出 IP。	write_ip_tcl <ip_name> 从 Tcl 脚本写出 IP。</ip_name>

用于常见 IP 操作的 Tcl 命令

在 Vivado IDE 中, Vivado IP catalog 可从 Vivado IDE 和 Tcl 设计环境访问。

为满足首选批量脚本编制模式的最终用户的需求, Vivado IDE 中执行的每项 IP 目录操作 (例如, IP 创建、重新自定 义、输出文件生成等)都会向 vivado.log 文件回显 1 条等效的 Tcl 命令;因此您在 Vivado IDE 中可执行的任何操作 均可同样通过脚本来执行。



Vivado IP catalog 支持从集成 Vivado IDE Tcl 控制台直接访问 IP 参数自定义功能,因此您可在 Tcl 控制台中直接设置各 IP 参数。

以下是使用 Tcl 命令执行的常用 IP 操作示例:

创建累加器 IP 的自定义:

create_ip -name c_accum -vendor xilinx.com -library ip \
-module_name c_accum_0

更改自定义参数,例如,输入和输出宽度:

set_property -dict [list CONFIG.Input_Width {10} CONFIG.Output_Width {10}] [get_ips c_accum_0]

生成选择性的输出文件:

generate_target {synthesis instantiation_template simulation} \
[get_ips c_accum_0]

复位生成的任意输出文件:

reset_target all [get_ips c_accum_0]

您可以通过使用 list_property 或 report_property 命令并引用创建的 IP 来使用 Tcl 脚本列出可供 IP 使用的用户配置参数。以下是适用于 IP 的实用 Tcl 命令。

返回可通过 Tcl 脚本处理的对象列表:

list_property

返回文本报告,提供每个参数的当前值、其类型及其它参数:

report_property

获取适用于 IP 的所有属性的列表:

list_property [get_ips fifo_generator_0]

返回仅包含自定义参数的列表 (按字母顺序排序):

lsearch -all -inline [list_property [get_ips fifo_generator_0]] CONFIG.*

创建报告,列出 IP 的所有属性 (包括配置参数):

report_property [get_ips fifo_generator_0]

移除设计运行:

delete_ip_run

导出仿真:

export_simulation

将文件从核容器解压到磁盘:

extract_files

在 Vivado IDE 中为 IP 移除设计运行并为 IP 复位输出文件时,会发出 2 条 Tcl 命令: reset_target 和 delete_ip_run。 将约束写入 XDC 文件:

write_xdc <file>



此命令会根据约束设置将约束写入文件,写入顺序与约束执行顺序相同。请参阅第 2 章中的"管理 IP 约束"以获取有关设置 IP 约束的更多信息。另请参阅附录 D中的"覆盖 IP 约束"。请参阅《Vivado Design Suite 用户指南:用户约束》 (UG903) [参照 21] 以获取更多信息。



视频:请参阅《Vivado Design Suite QuickTake 视频: IP 约束概述》以观看有关 IP 流程期间约束使用方式的演示。

视频:

视频:请参阅《Vivado Design Suite QuickTake 视频: Vivado 中的 Tcl 脚本和约束文件》以获取更多信息。

为所有 RTL IP 创建单一 Tcl 脚本,或者为含自定义的各 RTL IP 单独创建 Tcl 脚本:

write_ip_tcl

如需获取有关受支持的 IP Tcl 命令类型的更多信息,请参阅 Tcl 控制台中的 help -category IPFlow。

注释:《Vivado Design Suite 教程:采用 IP 进行设计》(UG939) [参照 29] 包含的实践中涵盖了适用于 IP 的工程模式流程和非工程模式流程的脚本编制相关信息。其中还包含生成输出文件以及选择性升级 IP 的示例。

IP 流程命令示例

本章节提供了部分常用操作的 Tcl 脚本示例。

用于创建 IP 的命令

create_ip 命令用于创建 IP 自定义。



建议:如"使用管理 IP 流程"中所述执行此操作。通过"Manage IP"流程创建 IP 时,后续可以在工程模式和非工程模式下使用此 IP。

以下脚本显示了如何创建"Manage IP"工程、创建和自定义 IP 以及生成 DCP:

Create a Manage IP project create_project <managed_ip_project> ./managed_ip_project -part <part> -ip # Set the simulator language (Mixed, VHDL, Verilog) set_property simulator_language Mixed [current_project] # Target language for instantiation template and wrapper (Verilog, VHDL) set_property target_language Verilog [current_project] # Create an IP customization create_ip -name c_accum -vendor xilinx.com -library ip -module_name c_accum_0 # configure the parameters for the IP customization set_property -dict {CONFIG.Input_Width 10 CONFIG.Output_Width 10} [get_ips c_accum_0] # Create a synthesis design run for the IP create_ip_run [get_ips c_accum_0] # Launch the synthesis run for the IP # Because this is a project, the output products are generated automatically launch_run c_accum_0_synth_1



重要提示:赛灵思建议采用基于工程的流程。基于工程的流程可在 Vivado IDE 中运行,也可使用 Tcl 命令运行。

查询 IP 自定义文件

用于获取文件以便进行源文件控制的 Tcl 脚本

此脚本示例演示了如何获取给定 IP 自定义的所有文件。您可以使用该脚本获取文件列表以配合源文件控制系统一起使用。

Create a project in memory, no project directory
created on disk
create_project -in_memory -part <part>

read an IP customization
read_ip <ip_name>.xci

Generate all the output products
generate_target all [get_ips <ip_name>]

Create a DCP for the IP
synth_ip [get_ips <ip_name>]

Query all the files for this IP
get_files -all -of_objects [get_files <ip_name>.xci]

注释: 请参阅表 6-1 以获取有关这些 Tcl 命令的详细说明及其适用的时机。

查询按顺序排列的源文件列表

创建定制脚本时,可以使用以下任一 Tcl 命令:

仅适用于 IP: 综合

get_files -compile_order sources -used_in synthesis \
-of_objects [get_files <ip_name>.xci]

仅适用于 IP: 仿真:

get_files -compile_order sources -used_in simulation \
-of_objects [get_files <ip_name>.xci]

适用于顶层设计:包含用于综合的 IP:

get_files -compile_order sources -used_in synthesis

适用于顶层设计:包含用于仿真的 IP:

get_files -compile_order sources -used_in simulation



脚本编制示例

实现 IP 设计示例

创建工程以在 IP 设计示例上运行实现。

```
# Create a project
create_project <name> <dir> -part <part>
# Create an IP customization and a DCP
# This will also generate all the output products
create_ip ...
create_ip_run [get_ips <ip>.xci]
launch_runs <ip>_synth_1
wait_on_run <ip>_synth_1
# Open the example design for the IP
# This will use the IP DCP generated
open_example_project -force -dir <project_location> -in_process [get_ips <ip>]
launch_runs synth_1
wait_on_run synth_1
launch_runs impl_1
wait_on_run impl_1 -to write_bitstream
open_run impl_1
# produce some reports
report_timing_summary ...
report_utilization ...
```

非工程综合

在非工程流程中综合并实现设计时,某个 IP 可能包含生成的 OOC DCP,某个 IP 可能与用户逻辑一起进行综合。

读取 IP XCI 文件时,将使用存在的所有输出文件 (包括 OOC DCP),无需生成这些文件。

如果尚未为 IP 生成输出文件,则必须生成这些输出文件 (或者使用 synth_ip 命令创建 DCP,此命令同样可用于生成输出文件)。

如果选择为 IP 使用全局综合 (请参阅第 2 章中的"IP 的综合选项"),那么必须禁用检查点支持,然后生成输出文件。 以下 Tcl 脚本可为此操作提供模板:

#create an in memory project to provide the part to use for IP creation and for #running synthesis

set_part <part>

read in sources
read_verilog top.v

Read in an existing IP customization
or create an IP from scratch
create_ip ... or read_ip ip1.xci

```
# Generate a DCP for the IP
# will generate output products if needed
synth_ip [get_ips ip1]
```

XILINX®

```
# Read in an existing IP customization
# or create an IP from scratch
# create_ip ... or read_ip ip2.xci
# Set IP to use global synthesis (no DCP generated)
set_property generate_synth_checkpoint false [get_files ip2.xci]
# Need to generate output products for IP
generate_target all [get_ips ip2]
# synthesis the complete design
synth_design -top top
# run implementation
opt_design
place_design
# write the bitstream
write_bitstream -file top
```

仿真 IP 设计示例

创建工程以在 IP 设计示例上运行仿真。

```
#create the project
create_project <name> <dir> -part <part>
# create IP and a synthesis run
create_ip_run [get_ips <ip_name>]
#launch runs
launch_runs <ip>_synth_1
wait_on_run <ip>_synth_1
#open the example project
open_example_project -force -dir <project_location> -in_process [get_ips <ip>]
#launch_simulation
<launch_simulation> | <target_simulator>
```

综合和仿真 IP

如果 IP 未提供设计示例,但已提供测试激励文件,则可仅对 IP 执行仿真。

#create the project create_project <name> <dir> -part <part> # create_ip ... or add_files ip.xci # create an IP design run create_ip_run [get_ips <ip_name>] #launch IP synthesis run launch_run <ip>_synth_1 wait_on_run <ip>_synth_1 # Setting up simulation test bench

```
set_property top <tb> [current_fileset -simset]
```



Launch simulation
<launch_simulation> | <target_simulator>



视频:请参阅《Vivado Design Suite QuickTake 视频: Vivado 中的 Tcl 脚本和约束文件》以获取更多信息。



附录 A

判定 IP 锁定原因

引言

Vivado® IP 核会因各种原因而变为锁定状态。Vivado 集成设计环境 (IDE) 提供了 IP 状态报告用于提供相关原因和建议。 下表列出了 IP 锁定消息与建议。请参阅"报告 IP 状态"以获取有关"IP 状态报告"的更多信息。



重要提示:使用赛灵思提供的补丁时,您可能会注意到由于补丁中 IP 定义发生变化而导致 IP 锁定。


表 A-1: IP 锁定原因与建议

原因简述	原因详情	简述	详细建议
IP 文件只读	<ip_name> 包含只读文件 <ipxmlfile>。 IP 处于写保护状态。</ipxmlfile></ip_name>	请检查文化和工程权限	复查导致 IP 变为只读的工程和文件系统权限。
	<ip_name> 包含只读文件 <ipxcifile> 且功能受限。用于更 改此 IP 配置的命令处于禁用状态。</ipxcifile></ip_name>	时位旦天日仲工任汉族	请参阅"编辑 IP 源文件"以获取更多信息。
共享输出目录	<ip_name> 与其它 IP 共享公用输 出目录。赛灵思建议将每个 IP 置于 其自己专用的目录内。</ip_name>	移动 IP	 使用 remove_files 命令手动从工程中移除此 IP (请参阅 《Vivado Design Suite Tcl 命令参考指南》(UG835) [参照 20])。 使用 import_files 命令将其导入工程中专用目录内。
未找到 IP 定义	在 IP 目录中未找到 <ip_name> 的 IP 定义 <current_ipdef> (己自 定 义 为 软 件 版 本 <sw_version>)。</sw_version></current_ipdef></ip_name>	将 IP 定义添加到目录中	请查询 IP 目录,以查找替换 IP。请参阅"使用 IP 目录"以获取更多信息。
IP 主要版本更改	IP 目录中的 <ip_name> 的 IP 定义 <current_ipdef>(己自定义为软 件版本 <sw_version>)包含更新 的主要版本。</sw_version></current_ipdef></ip_name>	升级 IP	目标 IP 定义 <target_ipdef> 需要执行主要版本更改。请先复查 升级 IP 对设计的影响,然后再执行升级。请参阅"升级 IP"以获取更多 信息。</target_ipdef>
IP 次要版本更改	IP 目录中的 <ip_name> 的 IP 定义 <current_ipdef>(己自定义为软 件版本 <swversion>) 包含更新的 次要版本。</swversion></current_ipdef></ip_name>	升级 IP	目标 IP 定义 <target_ipdef> 需要执行次要版本更改。请先复查 更改日志,然后再升级 IP。请参阅"升级 IP"以获取更多信息。</target_ipdef>
IP 修订版更改	IP 目录中的 <ip_name> 的 IP 定义 <current_ipdef>(己自定义为软 件版本 <sw_version>)包含不同 修订。</sw_version></current_ipdef></ip_name>	升级 IP	目标 IP 定义 <target_ipdef> 需要执行修订版更改。请先复查更改日志,然后再升级 IP。请参阅"升级 IP"以获取更多信息。</target_ipdef>
检测到不兼容的 IP 数据	存储库中的 IP 数据与当前实例不兼容(尽管版本号和修订版号相同)。 通常如果使用的是当前正在开发中的 IP,则可能发生此情况。您无法查 看自定义,也无法生成输出,直至其 完成更新为止。	升级 IP	升级 IP。请参阅"升级 IP"以获取更多信息。
IP 不支持器件	IP <ip_name> 不支持当前工程器件 <current_part>。但是,器件差 异可能导致发生未定义的行为。</current_part></ip_name>	不支持 升级 IP	目标 IP 定义 <target_ipdef> 不支持当前工程器件 <current_part>。请选择受支持的工程器件,然后再升级 IP。 请参阅附录 C: 对 IP 使用平台开发板流程以获取更多信息。</current_part></target_ipdef>

E XILINX_®

表 A-1: IP 锁定原因与建议 (续)

原因简述	原因详情	简述	详细建议
未找到 IP 许可证	IP <ip_name> 需要一个或多个必需的许可证,但未找到有效的许可证。</ip_name>	升级 IP 未获许可	目标 IP 定义 <target_ipdef> 需有效许可证。请先获取有效许可证, 然后再升级 IP。请参阅"使用付费许可 IP"以获取更多信息。</target_ipdef>
	但是,许可证检查点可能在某些工具 流程中阻止使用此 IP。	检查 IP 许可证	IP <ip_name> 需有效许可证。请获取有效许可证,或者复查您的许可环境。请参阅"使用付费许可 IP"以获取更多信息。</ip_name>
IP 开发板更改 ⁽¹⁾	此 IP 包含特定于开发板的输出。当前工程开发板 <current_board> 与用于自定义 IP <ip_name> 的开发 板 <original_board> 不匹配。</original_board></ip_name></current_board>	重新选择目标 IP	对当前工程器件或开发板使用升级流程来更改工程器件或者重新选择目标 IP。请参阅"升级 IP"以获取更多信息。
IP 器件更改	当前工程器件 <current_part> 与用于自定义 IP <ip_name> 的器件 <original_part> 不匹配。</original_part></ip_name></current_part>		对当前工程器件或开发板使用升级流程来更改工程器件或者重新选择目标 IP。请参阅"升级 IP"以获取更多信息。
IP 包含已锁定的子核	IP <ip_name> 包含一个或多个已锁 定的子核。</ip_name>	升级父 IP	升级父 IP <parentname>。请参阅"编辑 IP 源文件"以获取更多信息。</parentname>
	未找到 IP 定义	在 IP 目录中未找到 IP 定 义	将 IP 定义添加至目录中,或者查询 IP 目录以查找替换 IP。 请参阅"使用 IP 目录"以获取更多信息。
	只读 XCI/BOM/工程	XCI、XML 或 XPR 文件为 只读,因此 IP 处于写保护 状态。	复查导致 IP 变为只读的工程和文件系统权限。 请参阅"编辑 IP 源文件"以获取更多信息。
	用户管理的 IP	此 IP 己配置为用户管理的 IP。在此模式下,用户负 责管理所有 IP 文件。	如果这与期望不符,请将此 IP 重新配置为系统管理的 IP (请参阅"编 辑 IP 源文件"中所述的 is_managed 属性)。
其它	USER_LOCKED 属性	此 IP 已被用户锁定。在此 模式下,用户负责管理所 有 IP 文件。	在 IP 上移除 USER_LOCKED 属性。请参阅"编辑 IP 源文件"以获取更多 信息。
	组件被禁用	不支持升级 IP。	目标 IP 定义 <target_ipdef> 不支持当前工程器件 <current_part>。请选择受支持的工程器件,然后再升级 IP。请 参阅附录 C:对 IP 使用平台开发板流程以获取更多信息。</current_part></target_ipdef>
	许可证不兼容	此 IP 实例需要一个或多个 必需的许可证,但未找到 有效的许可证。	请先获取有效许可证,然后再升级 IP。请参阅"使用付费许可 IP"。

E XILINX_®

表 A-1: IP 锁定原因与建议 (续)

原因简述	原因详情	简述	详细建议
	XCI/BOM 不兼容	升级 IP	目录中的 IP 数据与当前 IP 实例不兼容 (尽管版本号和修订版号相同)。 您需要先更新此 IP, 然后才能查看自定义和生成输出。 请参阅"升级 IP"以获取更多信息。
—————————————————————————————————————	流程已弃用	升级 IP	此 IP 实例支持 Vivado 生成,但当前生成时使用的是 CORE™ Generator 工具。请参阅"升级 IP"以获取更多信息。
	因子 IP 被锁定而导致锁定	此 IP 实例包含一个或多个 已锁定的子核。	请在 IP 上运行升级或者使用当前已锁定的子 IP 的更新版本来重新封装该组件。请参阅"编辑或覆盖 IP 源文件"以获取更多信息。

1. 当 IP 核由于器件或开发板更改而导致被锁定时,此 IP 核升级后需复查端口。根据所选器件,部分 IP 可能存在端口差异。例如,对于 QSGMII IP,从 7 系列 FPGA 更新到 UltraScale® 平台后,调试端口名称和功能会发生更改。您必须执行 RTL 更改以避免综合和/或实现期间遇到错误。请参阅相应的 IP 产品指南以了解详情。



附录 B

IP 文件和目录结构

引言

使用 IP 目录自定义 IP 时,可在工程中直接操作,也可以使用"Managed IP"流程, Vivado® 集成设计环境 (IDE) 会为每个 IP 创建专用目录。

创建 IP 自定义后,会生成专用目录,其中包含赛灵思核实例 (XCI) 文件、例化模板、 BOM 文件以及所有已生成的输出 文件。在此 IP 目录中有多个其它目录。每个 IP 提供的文件并无统一的组织结构,但针对每个 IP 会创建一些公用文件。

IP 生成的目录和文件

下表列出了 IP 生成的目标目录和文件,也称为输出文件。



建议: 赛灵思建议您使用 Tcl 命令访问相关文件列表,而不是使用文件和目录结构视图。例如,您可使用 get_files Tcl 命令,如第 6 章中的"查询 IP 自定义文件"中所示。如需了解更多信息,请参阅《Vivado Design Suite Tcl 命令参考指 南》(UG835) [参照 13]。

表 B-1: IP 输出文件

目录名称、文件名称或文件类型	描述
/doc	包含 <core_name>_changelog.txt 文件,提供有关对每个版本的 IP 所 执行的更改的信息。</core_name>
/sim	包含 IP 的仿真源文件。此目录并非对于所有 IP 都存在。
/synth	包含 IP 的可综合源文件。此目录对于不支持综合的 IP 不存在,例如仅用于仿 真的 Verification IP。
<ip_name>.xci</ip_name>	包含 IP 自定义信息。您可从该文件生成输出文件。如果针对目录中的某个 IP 存在升级路径,您可从该文件升级到最新版本。
<ip_name>.xcix</ip_name>	核容器 (Core Container) 文件,列出设计中 IP 间的所有公用元素。
<ip_name>.xml</ip_name>	IP 物料清单 (BOM) 文件,用于保留 IP 当前状态的记录,包括生成的文件、计算参数以及接口信息。
<ip_name>.veo vho</ip_name>	Verilog (VEO) 或 VHDL (VHO) 例化模板。您将使用其中任一文件在自己的设 计内例化 IP。
<ip_name>.dcp*</ip_name>	综合后设计检查点文件,包含综合后的网表和已处理的 XDC 约束。
	建议: 赛灵思建议您不要直接引用 IP DCP 文件;而应改为使用 XCI 文件,此 文件可根据需要引入 DCP。
<ip_name>_stub.[v vhdl]*</ip_name>	模块 (Verilog) 和组件 (VHDL), 配合第三方综合工具用于为 IP 推断黑盒。





表 B-1: IP 输出文件 (续)

目录名称、文件名称或文件类型	描述
<ip_name>_funcsim.[v vhdl]*</ip_name>	综合后结构仿真网表文件 (仅适用于低于 2015.3 版本的 Vivado)。
<ip_name>_sim_netlist</ip_name>	综合后结构仿真网表文件 (适用于 2015.3 版本的 Vivado)。
<ip_name>.xdc</ip_name>	时序约束和/或物理约束。这些文件并非对于所有 IP 都存在,其位置因 IP 而异。
<ip_name>_in_context.xdc</ip_name>	请参阅"判定时控约束和解释时控消息"以获取更多信息。
dont_buffer.xdc	已弃用的文件。该功能包含在 <ip_name>_in_context.xdc 内。</ip_name>
<ip_name>_clocks.xdc</ip_name>	含时钟依赖关系的约束。这些文件并非对于所有 IP 都存在,其位置因 IP 而异。
<ip_name>_board.xdc</ip_name>	平台开发板流程中使用的约束。这些文件并非对于所有 IP 都存在,其位置因 IP 而异。
<ip_name>_ooc.xdc</ip_name>	以非关联流程综合 IP 时使用的默认时钟定义。
对应 IP 的加密 HDL	用于 IP 综合和仿真的文件。这些文件并非对于所有 IP 都存在,其位置因 IP 而异。

 仅当使用非关联流程执行综合 (默认)时,才会创建 DCP、_stub 和 *_funcsim 或 *_sim_netlist 文件。如需了解更多详情,请参阅 第 2 章中的"IP 的综合选项"。

注释: 虽然设计示例并非输出文件,但针对 IP 一般都会生成这些文件。仅当采用以下任一方法打开设计示例时,设计示例文件才可用:

- 。 在 Tcl 控制台中, 使用 open_example_project 命令。
- 。 在 Vivado IDE 中使用"Open IP Example Design"菜单命令。

如需了解更多信息,请参阅第 4 章"使用 IP 设计示例"。

与 IP 关联的文件

下表列出了可能与 IP 关联的其它类型的文件。

表 B-2: 与 IP 关联的文件

文件类型	描述
Name.coe	系数文件 (COE)。ASCII 文本文件,含单一基数报头后接多个矢量。基数可以是 2、10 或 16。每个矢量都必须以分号作为结尾。
Name.mif	存储器初始化文件 (MIF)。ASCII 文本文件, Vivado IDE 会将 COE 文件转换为此格式的文件。
Name.bmm	块存储器管理器文件。
Name.csv	逗号分隔版本的电子数据表文件。
Name.elf	可执行和可链接格式的文件,供 MicroBlaze [™] 处理器使用。

注释: 仅限部分 IP 使用此类文件。如果使用工程,用户应将其作为源文件添加到工程中。这些文件通常是使用 IP 的自定义 GUI 中提供的配置属性来设置的。欲知详情,请参阅相应 IP 的产品指南。



使用 COE 文件

在某些情况下,部分参数值使用 COE (COEfficient) 文件传递至 Vivado IP catalog; COE 文件是含单一基数报头后接多个 矢量的 ASCII 文本文件。基数可以是 2、 10 或 16。每个矢量都必须以分号作为结尾。

Vivado 工具会读取 COE 文件,并在生成核时写出 1 个或多个 MIF 文件。该核的 VHDL 和 Verilog 行为仿真模型依赖于 这些 MIF 文件。

重要提示:您必须先升级所有 IP,然后再添加 COE 文件。此外, COE 文件与 XCI 文件应置于相同目录中。

注释:如果 IP 不再使用某个 COE 文件,请移除此文件。不移除旧 COE 文件可能导致新关联的 COE 和旧的 COE 被一并 传递至综合。此外,如果从磁盘上移除旧 COE 文件而不从工程中将其移除,那么综合期间可能会发生错误。

COE 文件语法

以下语法显示了 COE 文件的一般形式:

Keyword =Value ; Optional Comment
Keyword =Value ; Optional Comment
<Radix_Keyword> =Value ; Optional Comment
<Data_Keyword> =Data_Value1, Data_Value2, Data_Value3;

下表描述了用于为数据指定基数值的 COE 文件关键字。关键字不区分大小写。如需了解有关 IP 所需的特定关键字的信息,请参阅对应 IP 的产品指南。

表 B-3: 对应基数值的 COE 文件关键字

关键字	描述
RADIX	供非存储器核用于指示用于指定筛选工具系数的基数。
MEMORY_INITIALIZATION_RADIX	供存储器初始化值用于指定所使用的基数。

下表描述了数据值的 COE 文件关键字。关键字不区分大小写。

表 B-4: 对应数据值的 COE 文件关键字

关键字	描述
COEFDATA	供筛选工具用于指示后续数据包含筛选工具的系数。
MEMORY_INITIALIZATION_VECTOR	供块存储器和分布式存储器使用。
PATTERN	供 Bit Correlator COE 文件使用。
BRANCH_LENGTH_VECTOR	在 Interleaver COE 文件中使用。

注释: 分号后的所有文本均作为注释来处理, 因此将被忽略。

在 COE 文件中必须指定下列任一关键字作为最后一个关键字:

- COEFDATA
- MEMORY_INITIALIZATION_VECTOR

位于上述任一关键字之后的任何其它关键字都将被忽略。





COE 文件示例

以下是各种赛灵思 IP 的 COE 文件示例。

Virtex Bit Correlator COE 文件示例

双端口块存储器 COE 文件示例

```
******** Example of Dual Port Block Memory .COE file *********
; Sample memory initialization file for Dual Port Block Memory,
; v3.0 or later.
;
; This .COE file specifies the contents for a block memory
; of depth=16, and width=4. In this case, values are specified
; in hexadecimal format.
memory_initialization_radix=2;
memory_initialization_vector=
1111,
1111,
1111,
1111,
1111.
0000,
0101,
0011,
0000,
1111,
1111,
1111,
1111,
1111,
1111,
1111;
```



单端口块存储器 COE 文件示例

```
******
              ******* Example of Single Port Block Memory .COE file ********
; Sample memory initialization file for Single Port Block Memory,
; v3.0 or later.
; This .COE file specifies initialization values for a block
; memory of depth=16, and width=8. In this case, values are
; specified in hexadecimal format.
memory_initialization_radix=16;
memory_initialization_vector=
ff,
ab.
f0,
11,
11,
00,
01,
aa,
bb,
CC,
dd,
ef,
ee,
ff,
00,
```

ff;

Distributed Memory COE 文件示例

```
; Sample memory initialization file for Distributed Memory v2.0 and
; later.
;
; This .COE file is NOT compatible with v1.0 of Distributed Memory Core.
; The example specifies initialization values for a memory of depth= 32,
; and width=16. In this case, values are specified in hexadecimal
; format.
memory_initialization_radix = 16;
memory_initialization_vector = 23f4 0721 11ff ABe1 0001 1 0A 0
23f4 0721 11ff ABe1 0001 1 0A 0
23f4 721 11ff ABe1 0001 1 A 0
23f4 721 11ff ABe1 0001 1 A 0;
****** Example of Distributed Arithmetic FIR Filter .COE file ***
; Example of a Distributed Arithmetic (DA) FIR Filter .COE file
; with hex coefficients, 8 symmetrical taps, and 12-bit
; coefficients.
;
```



; Compatible with all versions of the Distributed Arithmetic ; FIR Filter which supports Virtex and Spartan Radix = 16; CoefData= 346, EDA, 0D6, F91, F91, 0D6, EDA, 346;

MIF 文件描述

COE 文件提供了一种高级方法用于指定初始内存内容。生成核时, Vivado 工具会将 COE 文件转换为 MIF 文件,其中保存了用于在核与仿真模型中初始化内存的实际二进制数据。

MIF 文件对应每个内存位置包含一行文本。此文件中的第一行对应于地址 0,第二行对应于地址 1,以此类推。每一行上的文本都必须为对应内存地址的二进制格式初始化值 (MSB 优先),内存位宽的每个位对应 1 个二进制数值。

注释:对于 HDL 仿真, MIF 文件必须驻留在仿真目录中。



附录 C

对 IP 使用平台开发板流程

引言

部分 IP 支持 Vivado® Design Suite 的"平台开发板流程 (Platform Board Flow)"功能,该功能支持您在自定义 IP 时选择 开发板接口。通过使用该功能,即可通过输入其它 XDC 约束来定义管脚分配并输入 IOSTANDARDS 作为目标开发板上 实现的接口信号来自动为 IP 创建物理约束。

如图 C-1 所示,创建新工程时,可以选择开发板作为默认器件。

Filter/ Preview										
Ve <u>n</u> dor:	All	~								
Display Name:	All	~								
Board Rev	Lotoot									
Board Rey.	Reset All Filters	•								
Bearch: Q-		~								
Display Name			Vendor	Board Rev	Part	I/O Pin Count	File Version	Available IOBs	LUT Elements	FlipFlops
📓 ZedBoard Zynq E	Evaluation and Development	Kit	em.avnet.com	đ	xc7z020clg484-1	484	1.3	200	53200	106400
Artix-7 AC701 Eva	aluation Platform		xilinx.com	1.1	xc7a200tfbg676-2	676	1.3	400	134600	269200
Kintex-7 KC705 B	Evaluation Platform		xillinx.com	1.1	xc7k325tffg900-2	900	1.5	500	203800	407600
Kintex-UltraScale	e KCU105 Evaluation Platform	n .	xilinx.com	1.0	xcku040-ffva1156-2-e	1,156	1.2	520	242400	484800
Kintex UltraScale	+ KCU116 Evaluation Platfo	rm	xillinx.com	1.0	xcku5p-ftvb676-2-e	676	1.0	280	216960	433920
Kintex UltraScale	KCU1500 Acceleration Dev	elopment Board	xilinx.com	1.0	xcku115-flvb2104-2-e	2,104	1.0	702	663360	1326720
Virtex-7 VC707 E	valuation Platform		xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.3	700	303600	607200
Virtex-7 VC709 E	valuation Platform		xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.8	850	433200	866400
Virtex-UltraScale	VCU108 Evaluation Platform	1	xilinx.com	1.0	xcvu095-ffva2104-2-e	2,104	1.2	832	537600	1075200
Virtex-UltraScale	VCU110 Evaluation Platform	n	xillinx.com	1.0	xcvu190-flgc2104-2-e	2,104	1.1	416	1074240	2148480
ZYNO-7 ZC702 E	Evaluation Board		xilinx.com	1.0	xc7z020dg484-1	484	1.3	200	53200	106400
ZYNQ-7 ZC706 E	Evaluation Board		xilinx.com	1.1	@ xc7z045ffg900-2	900	1.4	362	218600	437200
Zynq UltraScale+	ZCU 102 Evaluation Board		xilinx.com	1.0	xczu9eg-ffvb1156-2-i	1,156	3.0	328	274080	548160
0										

图 C-1: 选择开发板作为默认器件



选中其中列出的任一开发板即可在支持平台开发板流程的 IP 核的 IP 自定义对话框中启用"Board"选项卡,如图 C-2 所示。

Select: 📵 Parts	Boards		-				
Filter/ Preview				while the second	12 See		
Ve <u>n</u> dor:	All 🗸		1.4.4.4. 1.1.1.4.			a a	
Display <u>N</u> ame:	All					ter Marca	
Roard Boyr	Latast					To the	
board Key.							
	Reset All Filters			20 (V. Longen) 2. Det Killigen Termin Prover Michael 1. Det Killigen Termin Prover Michael 1. Det Killigen Termin Prover Michael 1. Det Killigen 1. Det	E Contraction (1997)		
Search: Q-	~						
						1000 ALL	Available
Display Name		Vendor	Board Rev	Part	I/O Pin Count	File Version	IOBs
ZedBoard Zyng Evaluation and Development Kit		em.avnet.com	d	xc7z020clg484-1	484	1.3	200 ^
Artix-7 AC701 Ev	aluation Platform	xilinx.com	1.1	xc7a200tfbg676-2	676	1.3	400
Kintex-7 KC705	Evaluation Platform	xilinx.com	1.1	xc7k325tffg900-2	900	1.5	500
Kintex-UltraScale KCU105 Evaluation Platform		xilinx.com	1.0	xcku040-ffva1156-2-e	1,156	1.2	520
Kintex-UltraScal	e+ KCU116 Evaluation Platform	xilinx.com	b	xcku5p-ffvb676-2-e	676	1.0	280
Kintex UltraScal	e KCU1500 Acceleration Development Board	xilinx.com	1.0	xcku115-flvb2104-2-e	2,104	1.0	702
Virtex-7 VC707 E	valuation Platform	xilinx.com	1.1	xc7vx485tffg1761-2	1,761	1.3	700
Virtex-7 VC709 E	Evaluation Platform	xilinx.com	1.0	xc7vx690tffg1761-2	1,761	1.8	850
Virtex-UltraScale	VCU108 Evaluation Platform	xilinx.com	1.0	xcvu095-ffva2104-2-e	2,104	1.2	832
Virtex-UltraScale	VCU110 Evaluation Platform	xilinx.com	1.0	xcvu190-flgc2104-2-e	2,104	1.1	416
ZYNQ-7 ZC702	Evaluation Board	xilinx.com	1.0	@ xc7z020clg484-1	484	1.3	200
ZYNQ-7 ZC706	Evaluation Board	xilinx.com	1.1	@ xc7z045ffg900-2	900	1.4	362
📓 Zynq UltraScale	+ ZCU102 Evaluation Board	xilinx.com	1.0	😟 xczu9eg-ffvb1156-2-i	1,156	3.0	328
: <	· · · · · · · · · · · · · · · · · · ·						>
Board Connectors			Target Cor	nections			
FMC_HPC							~
FMC_LPC							~

图 C-2:开发板摘要信息



选中其中列出的任一开发板即可在自定义期间显示 1 个对应支持平台开发板流程的 IP 的新选项卡,如下图所示。

Documentation 🔚 IP Location 🤇	Switch to Defaults			
Show disabled ports	Component Name	axi_gpio_0		6
	Board IP Configuration			
	Associate IP interfa			
	IP Interface		Board Interface	
	GPIO		Custom	•
	GPIO2		Custom	•
s_axi_acik GPID +	Clear Board P	arameters		

图 C-3: 受支持的 IP 自定义中显示的开发板类型

"Board"选项卡支持您将 IP 核上定义的接口与目标开发板上实现的接口相关联。图 C-4 显示您可将 IP 接口关联到其中一个已关联的开发板接口。



🍋 Customize IP		
AXI GPIO (2.0)		4
🚯 Documentation 📄 IP Location C	Switch to Defaults	
Show disabled ports	Component Name axi_gpio_0	8
	Board IP Configuration	tinterface
	IP Interface	Board Interface
	GPIO	Custom
	GPI02	Custom
s_axi_aclk GPID +	Clear Board Parameters	dip switches 4bits Icd 7bits Ied 8bits push buttons 5bits rotary switch
	Enable Interrupt	OK Cancel

图 C-4: 将 IP 接口与开发板接口相关联

当 Vivado IDE 在"IP Sources"视图中生成 IP 输出文件时,可以看到其中会列出 <IP_Name>_board.xdc 文件。

此文件包含物理约束,用于将 IP 的端口分配到封装管脚,这些封装管脚连接到相关的开发板连接器或器件(例如,USB 端口、LED、按钮或开关)。

图 C-5 显示了将"GPI0"接口连接到开发板"Icd7bits"接口并将"GPIO2"接口连接到开发板"Custom"接口时为 GPIO IP 创建的 XDC 约束。



Customize IP				×	
AXI GPIO (2.0)				4	
Documentation P Location	C Switch to Defaults				
Show disabled ports	Component Name axi_gpio_0		0		
	Board IP Config	uration			
	Associate IP interface with board interface				
	IP Interface		Board Interface		
	GPIO		Icd 7bits		
	GPIO2		Custom	•	
	Clear Board P	arameters			
	Enable Interrupt				
			ок	Cancel	

图 C-5:已选中开发板接口

通过使用 Vivado Design Suite 平台开发板流程,您即可将 IP 接口信号快速连接到目标开发板,从而在开发板上加速完成器件实现。

如果已为工程选定目标开发板,那么支持 Vivado 平台开发板流程的任何 IP 都会在 IP 自定义对话框中显示"Board"选项 卡,如图 C-4 中所示。

- 请访问此链接以参阅《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》(UG994) [参照 32] 以获 取有关使用平台开发板的更多信息。
- 请访问此链接以参阅《Vivado Design Suite 用户指南:系统级设计输入》(UG895) [参照 18] 以获取有关开发板接口文件和创建您自己的开发板文件的信息。



附录 D

编辑或覆盖 IP 源文件

引言

有时,您可能需要修改或覆盖 IP 提供的未加密源文件,包括 XDC 文件和 HDL 文件。如非必要,请勿执行此操作。修改 IP 源文件可能导致 IP 无法正常运行。

重要提示:如果您判定确实必须修改任何 IP 源文件,除非谨遵本附录中提供的指导信息进行操作,否则请勿直接修改 磁盘上的源文件。直接修改磁盘上的源文件可能导致所作更改被删除,因为在此流程期间可能会对 IP 进行复位或重新 生成。

如需修改 IP RTL 源文件,请参阅"编辑 IP 源文件"。

要修改 IP 提供的 XDC 命令,可覆盖顶层 XDC 或 Tcl 文件中的 XDC,或者编辑 IP 源文件。

注释:这些选项适用于 RTL 工程中的大部分 IP。您无法修改包含层级 IP 的 IP。

☆

重要提示:请务必先禁用"Core Container"功能,然后再编辑 IP 源文件。

覆盖 IP 约束

IP 利用随附的约束来进行验证。但在某些情况下,可能需要更改 IP 提供的约束(例如,LOC 或 PACKAGE_PIN 属性等对应的物理约束)才能满足设计目标。

您可使用"编辑 IP 源文件"中所述方法来编辑 IP XDC。或者,您可以通过提供含所需命令的顶层用户 XDC 或 Tcl 文件来 覆盖 IP XDC 命令。

•

视频:请参阅 《Vivado Design Suite QuickTake 视频:使用约束集》以获取有关约束的更多信息。

根据要覆盖的约束种类,您可以使用 XDC 文件或 Tcl 文件 (请访问此链接以参阅《Vivado Design Suite 用户指南:如何使用约束》 (UG903) [参照 21])。

建议:强烈建议您不要修改任何 IP 时序约束,但可修改_ooc.xdc 以便设置以非关联流程进行 IP 综合的目标频率。

由于默认情况下以非关联流程进行 IP 综合,因此仅限在实现阶段才能执行物理约束的覆盖操作。单独执行 IP 综合期间 将忽略物理约束;因此导致 Tcl 或 XDC 文件仅限用于实现。

如需覆盖 IP 时序约束,请谨遵"编辑 IP 源文件"中所述流程进行操作。这样即可确保在以非关联流程进行 IP 综合期间以及在顶层实现期间均可应用您所做的更改。



XDC 命令按顺序从最后一条命令开始进行处理。对于时序约束,此操作并不总能成功;如果 IP 路径设置为包含伪路径 例外,并且您后续对该路径应用 max_delay 约束,则会保留此伪路径,因为其优先级更高(请访问此链接并参阅 《Vivado Design Suite:用户约束》(UG903) [参照 20] 以获取更多详情)。要执行这些级别的更改,您必须修改 IP 所提供的 XDC。

在 XDC 中禁止执行某些操作和命令,此类情况下需使用 Tcl 文件。例如,在 BUFG_GT 单元上更改 LOC 属性。布局器无 法在已被占用的 site 位置上进行实例布局。您必须首先清除当前设置,然后设置新的 LOC。此操作通过 reset_property 命令完成,该命令并非 XDC 命令,并且必须置于 Tcl 文件中。复位 LOC 属性后,请设置新的值。

作用域限定约束

IP 提供的 XDC 文件作用域限定于使用 XDC 文件上的下列 2 个属性的 IP 实例:

- 。 SCOPED_TO_REF: 指定 XDC 文件应用到的模块。
- 。 SCOPED_TO_CELLS: 指定模块中 XDC 文件应用到的单元。

如需了解有关这些属性的更多信息,请参阅《Vivado Design Suite 用户指南:如何使用约束》(UG903) [参照 21]。



在顶层覆盖 IP 约束时,有 2 种方法可供选择:

- 将层级指定为 IP 的特定单元。如果 IP 存在多个实例,请执行以下任一操作:
 - 。 使用通配符
 - 。 复制每个 IP 的约束
- 使用 IP 所使用的 SCOPED_TO_REF 和 SCOPED_TO_CELLS 属性,按 IP 单元为层级顶层的方式来编写约束(推荐方法)。

要查找 SCOPED_TO_REF 和 SCOPED_TO_CELLS 值,可使用 report_compile_order -constraints 命令。查看 IP 文件集的综合或实现部分。

Constraint evaluation order for 'synthesis' with fileset 'pcie3_X8_XOY1':						
Index	File Name	Used_In	Scoped_To_Ref	Scoped_To_Cells		
1	pcie3_X8_X0Y1_ooc.xdc	Synth & Impl	pcie3_X8_X0Y1	inst		
2	pcie3_X8_XOY1_gt.xdc	Synth & Impl	pcie3_X8_X0Y1_gt	inst		
3	pcie3_X8_X0Y1-PCIE_X0Y1.xdc	Synth & Impl	pcie3_X8_X0Y1	inst		

图 D-1: 综合文件集

SCOPED_TO_REF 通常为 IP 自定义名称。 SCOPED_TO_CELLS 通常为 inst (在 Verilog 中) 或 U0 (在 VHDL 中)。

建议: 赛灵思建议您创建新的 XDC 或 Tcl 文件,添加覆盖 IP XDC 所需的所有 XDC/Tcl 命令,并设置 SCOPED_TO_REF 和 SCOPED_TO_CELLS 属性以与 1 所列结果相匹配。

完整流程如下:

- 1. 创建新的 XDC 或 Tcl 文件,并将其添加到处于活动状态的约束集中。
- 2. 添加覆盖新文件中的 IP XDC 所需的所有 XDC 或 Tcl 命令。
- 3. 使用 set_property 命令以设置 SCOPED_TO_REF 和 SCOPED_TO_CELLS 属性:

set_property SCOPED_TO_REF <REF> [get_files <new XDC/Tcl file>]
set_property SCOPED_TO_CELLS <CELL> [get_files <new XDC/Tcl file>]

 将此 XDC/Tcl 文件标记为仅限在实现中使用: set_property USED_IN IMPLEMENTATION [get_files <net XDC/Tcl file]

编辑 IP 源文件

要准备编辑 IP, 请执行以下操作:

1. 如果尚未自定义 IP,请完成自定义,并生成所有输出文件(包括 DCP)。如果不想针对 IP 使用默认 OOC 流程,请 禁用 DCP 创建。

建议: 赛灵思强烈建议您使用默认流程。

2. 生成输出文件 (包括适用的 DCP) 后,请在 IP 的 XCI 文件上使用以下 Tcl 命令将 IS_MANAGED 属性设置为 false:

set_property IS_LOCKED true [get_files <IP_Name>.xci]

对于复杂的子系统 IP, 会显示以下错误消息:



ERROR: [IP_Flow 19-3666] The is_managed property cannot be directly modified for hierarchical IP.



注意!将 IS_MANAGED 属性设置为 false 后,此 IP 将由用户管理。您无法将该属性重新切换回 true,因为合理范围内用户编辑的源文件被覆盖的风险很高。

3. 出现此错误时,请阅读"编辑子系统 IP"并遵循其中步骤进行操作。

将 IS_MANAGED 属性设置为 false 会导致 IS_LOCKED 属性变为 TRUE。"IP Sources"窗口中的 IP 图标会变为 ■ 表示此 IP 并非由 Vivado 管理,而是已改由用户管理。

在"Report IP Status"命令的输出窗口中,可以看到此 IP 由用户管理,并且您可修改未加密的 HDL 文件和 XDC 文件。

- 4. 完成所需的编辑。
- 5. 按如下所述步骤重新创建 IP 输出文件,包括 DCP:
 - a. 复位 IP OOC 运行。此操作必须使用 Tcl 控制台来操作。查看"Out-of-Context Module Runs"文件夹中的"Design Runs"选项卡,找到 IS_MANAGED 属性已设为 FALSE 的 IP, 其名称为 <IP_Name>_synth_1。在 Tcl 控制台中 执行以下命令以复位运行:

reset_run <ip_name>_synth_1

b. 使用以下命令重新启动运行:

launch_run <ip_name>_synth_1

这样会使用您修改的任意 IP 的 HDL 或约束。

运行完成后,您即可照常使用 IP。

建议:通过引用 XCI 文件 (推荐),您即可使用 IP 源文件来进行仿真,并使用 DCP 进行顶层文件的综合以及实现。

编辑子系统 IP

部分复杂的子系统 IP 不允许更改 IS_MANAGED 属性。 7 系列和 UltraScale™ 器件系列即存在此状况。

子系统 IP 是否允许更改 IS_MANAGED 属性取决于特定 IP 的特定自定义选项。



注意!编辑此类 IP 的 RTL 文件存在风险。 可以通过更改来使子核连接失效。应谨慎考量对这些 IP HDL 源文件执行的更 改。

- 1. 对于综合选项,请确保已使用 OOC 为 IP 完全生成 IP 输出文件。对于存在的 IP,需存在现有设计运行。
- 2. 将 IS_MANAGED 属性设置为 false,这将把 IS_LOCKED 属性设为 true。对于不允许更改 IS_MANAGED 属性的复杂 子系统 IP,请将 IS_LOCKED 属性设置为 true。
- 3. 这将使 IP 改为由用户管理。



注意! 将 IS_MANAGED 属性设置为 false 后,此 IP 将由用户管理。您无法将该属性重新切换回 true,因为合理范围内用户编辑的源文件被覆盖的风险很高。

- 4. 找到需要编辑的 IP RTL 文件并按需执行更改。您必须完成以下任一操作:
 - 。 在文本编辑器部分中使用"Tools > Options > General"切换到其它编辑器。
 - 。 使用您所选的文本编辑器直接在磁盘上编辑文件。



- 5. 按如下所述步骤重新创建 IP 输出文件,包括 DCP:
 - a. 使用 Tcl 控制台复位 IP OOC 运行。
 - b. 查看"Out-of-Context Module Runs"文件夹中的"Design Runs"选项卡,找到要重新综合的 IP,其名称为 <IP_Name>_synth_1。
 - c. 在 Tcl 控制台中执行以下操作以复位运行: reset_run <ip_name>_synth_1
 - d. 在 Tcl 控制台中使用以下命令重新启动运行:
 launch_run <ip_name>_synth_1

这样会使用您修改的任意 IP 的 HDL 或约束。

运行完成后,您即可照常使用 IP。

注释:由于子系统 IP 不允许更改 IS_MANAGED 属性,因此无法直观查看您所做更改的结果。用户自行负责保留 IP 修改记录。



附录 E

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。

解决方案中心

如需了解设计周期各阶段有关器件、软件工具和 IP 等的技术支持,请参阅赛灵思解决方案中心。相关专题包括设计辅助、建议和故障排除提示等。如需获取有关存储器 IP 的信息,请参阅赛灵思存储器接口解决方案中心。

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。打开 DocNav 的方法:

- 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 (Xilinx Design Hubs) 提供了根据设计任务和其它话题整理的文档链接,您可以使用这些链接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"视图。
- 在赛灵思网站上,请参阅设计中心页面。

注释:如需了解有关 Documentation Navigator 的更多信息,请参阅赛灵思网站上的 Documentation Navigotor。

参考资料

赛灵思网站

1. 最终用户许可协议

EXILINX®

- 2. 核许可协议
- 3. 核评估许可协议

Vivado Design Suite 技术文档

本指南中引用的文档如下:

- 1. 《Zynq-7000 SoC Verification IP 数据手册》 (DS940)
- 2. 《Zynq MPSoC UltraScale Verification IP 数据手册》 (DS941)
- 3. 《IBERT 7 Series GTX Transceivers LogiCORE IP 产品指南》 (PG132)
- 4. 《IBERT 7 Series GTP Transceivers LogiCORE IP 产品指南》 (PG133)
- 5. 《UltraScale Architecture-Based FPGAs Memory IP LogiCORE IP 产品指南》 (PG150)
- 6. 《IBERT 7 Series GTH Transceivers LogiCORE IP 产品指南》 (PG152)
- 7. 《Virtual Input/Output LogiCORE IP 产品指南》 (PG159)
- 8. 《Integrated Logic Analyzer LogiCORE IP 产品指南》 (PG172)
- 9. 《JTAG to AXI LogicCORE IP 产品指南》 (PG174)
- 10. 《AXI Verification LogiCORE IP 产品指南》 (PG267)
- 11. 《AXI4-Stream Verification IP LogiCORE IP 产品指南》 (PG277)
- 12. 《Zynq-7000 SoC 与 7 系列 FPGA 存储器接口解决方案》 (UG586)
- 13. 《Vivado Design Suite Tcl 命令参考指南》 (UG835)
- 14. 《Vivado Design Suite 教程:设计流程简介》 (UG888)
- 15. 《Vivado Design Suite 用户指南:设计流程简介》 (UG892)
- 16. 《Vivado Design Suite 用户指南:如何使用 Vivado IDE》 (UG893)
- 17. 《Vivado Design Suite 用户指南:如何使用 Tcl 脚本》 (UG894)
- 18. 《Vivado Design Suite 用户指南:系统级设计输入》 (UG895)
- 19. 《Vivado Design Suite 用户指南: I/O 管脚分配和时钟规划》 (UG899)
- 20. 《Vivado Design Suite 用户指南:逻辑仿真》 (UG900)
- 21. 《Vivado Design Suite 用户指南:如何使用约束》 (UG903)
- 22. 《Vivado Design Suite 用户指南: 层级设计》 (UG905)
- 23. 《Vivado Design Suite 用户指南:编程和调试》 (UG908)
- 24. 《Vivado Design Suite 用户指南:入门指南》 (UG910)
- 25. 《ISE 到 Vivado Design Suite 移植指南》 (UG911)
- 26. 《Vivado Design Suite 属性参考指南》 (UG912)
- 27. 《Vivado Design Suite 教程:编程和调试》(UG936)
- 28. 《Vivado Design Suite 教程:逻辑仿真》 (UG937)
- 29. 《Vivado Design Suite 教程:采用 IP 进行设计》 (UG939)
- 30. 《UltraFast 设计方法指南 (适用于 Vivado Design Suite)》 (UG949)
- 31. 《Vivado Design Suite 用户指南:版本说明、安装和许可》(UG973)
- 32. 《Vivado Design Suite 用户指南:采用 IP integrator 设计 IP 子系统》 (UG994)
- 33. 《Vivado AXI 参考指南》 (UG1037)
- 34. 《Vivado Design Suite 用户指南: 创建和封装定制 IP》 (UG1118)
- 35. 《Vivado Design Suite 教程: 创建和封装定制 IP》 (UG1119)



- 36. Vivado Design Suite 技术文档
- 37. Vivado IP 版本
- 38. IP 文档
- 39. IP 中心

标准和第三方文档

- 1. IP-XACT《在工具流程中封装、集成和复用 IP 的标准结构》(IEEE 1685 标准)
- 2. 《IEEE 推荐的电子产品设计 IP 加密与管理实践》(IEEE P1735 标准)

培训资料

赛灵思提供多种多样的培训课程和 QuickTake 视频,可帮助用户进一步了解有关本文档中提出的概念。使用以下链接获 取相关培训资料:

- 1. FPGA 设计要领
- 2. 嵌入式系统软件设计



Vivado QuickTake 视频

- 1. Vivado Design Suite QuickTake 视频:管理 Vivado IP 版本升级
- 2. Vivado Design Suite QuickTake 视频: Vivado 许可与激活简介
- 3. Vivado Design Suite QuickTake 视频:在 Vivado 中创建 AXI 外设
- 4. Vivado Design Suite QuickTake 视频: 在 Vivado 中配置和管理可复用 IP
- 5. Vivado Design Suite QuickTake 视频:设计约束概述
- 6. Vivado Design Suite QuickTake 视频:全局时序约束
- 7. Vivado Design Suite QuickTake 视频:管理工程源文件
- 8. Vivado Design Suite QuickTake 视频:将 UCF 约束移植到 XDC
- 9. Vivado Design Suite QuickTake 视频: Vivado 中的 Tcl 脚本和约束文件
- 10. Vivado Design Suite QuickTake 视频: 使用面向 IP 的 Core Container 功能
- 11. Vivado Design Suite QuickTake 视频:如何配合使用第三方综合工具和 IP
- 12. Vivado Design Suite QuickTake 视频: Vivado 激活和浮动许可证生成
- 13. Vivado Design Suite QuickTake 视频: 使用约束集
- 14. Vivado Design Suite QuickTake 视频: 使用设计检查点
- 15. Vivado Design Suite QuickTake 视频: 使用 UltraScale 存储器 IP 进行设计
- 16. Vivado Design Suite QuickTake 视频: Vivado IDE 入门
- 17. Vivado Design Suite QuickTake 视频:如何使用 Zynq-7000 Verification IP 通过仿真来进行验证和调试
- 18. Vivado Design Suite QuickTake 视频教程

请阅读: 重要法律提示

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 者赔偿是何种类或性质,也不论是基于合同、侵权、过失或是其它责任认定原理,即便该损失或赔偿可以合理预见或赛 灵思事前被告知有发生该损失或赔偿的可能。赛灵思无义务纠正资料中包含的任何错误,也无义务对资料或产品说明书 发生的更新进行通知。未经赛灵思公司的事先书面许可,贵司/您不得复制、修改、分发或公开展示本资料。部分产品受 赛灵思有限保证条款的约束,请参阅赛灵思销售条款:https://china.xilinx.com/legal.htm#tos; IP核可能受赛灵思向贵司/ 您签发的许可证中所包含的保证与支持条款的约束。赛灵思产品并非为故障安全保护目的而设计,也不具备此故障安全 保护功能,不能用于任何需要专门故障安全保护性能的用途。如果把赛灵思产品应用于此类特殊用途,贵司/您将自行承 担风险和责任。请参阅赛灵思销售条款:https://china.xilinx.com/legal.htm#tos。

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。

© 2012-2020 年赛灵思公司版权所有。Xilinx、赛灵思标识、Artix、ISE、Kintex、Spartan、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 的 商 标。"AMBA"、"AMBA Designer"、"Arm"、"ARM1176JZ-SV"、"CoreSight"、"Cortex"、"PrimeCell"、"Mali"和"MPCore"为 Arm Limited 在欧盟 及其它国家的注册商标。"MATLAB"和"Simulink"均为 The MathWorks, Inc. 拥有的注册商标。"PCI"、"PCIe"和"PCI Express"均为 PCI-SIG 拥有的商标,且经授权使用。所有其它商标均为各自所有方所属财产。