Intel® Quartus® Prime Pro Edition User Guide

设计约束

针对 Intel® Quartus® Prime 设计套件的更新：**19.3**

本翻译版本仅供参考，如果本翻译版本与其英文版本存在差异，则以英文版本为准。某些翻译版本尚未更新对应到最新的英文版本，请参考英文版本以获取最新信息。

在线版本
发送反馈
UG-20142
ID: **683143**
版本: **2019.10.16**

内容

1. 限制设计 4
1.1. 在 GUI 中指定设计约束设计 4
1.1.1. 全局约束和分配 4
1.1.2. 节点，实体和实例级约束 5
1.1.3. Intel Quartus Prime GUI 的组件之间的探测 8
1.1.4. 在 GUI 中指定时序约束 9
1.2. 使用 Tcl 脚本约束设计 10
1.2.1. 创建一个工程并应用约束 11
1.2.2. 分配一个管脚 11
1.2.3. 生成 Intel Quartus Prime 设置文件 12
1.2.4. Synopsys Design Constraint (.sdc)文件 13
1.2.5. Tcl-only 脚本流程 13
1.3. 一个完全迭代的脚本流程 15
1.4. 约束设计修订历史 15

2. 接口规划 17
2.1. 接口规划概述 18
2.2. 使用 Interface Planner 19
2.2.1. 步骤 1：设置和综合工程 22
2.2.2. 步骤 2：初始化 Interface Planner 22
2.2.3. 步骤 3：使用工程分配更新规划 22
2.2.4. 步骤 4：规划外围布局 23
2.2.5. 步骤 5：报告布局数据 27
2.2.6. 步骤 6：验证和导出规划约束 27
2.3. Interface Planner 用户接口控制 28
2.3.1. 流程控制(Flow Controls) 28
2.3.2. Home 选项卡控制 29
2.3.3. Assignments 选项卡控制 29
2.3.4. Plan 选项卡控制 29
2.3.5. Reports 选项卡控制 31
2.4. Interface Planner 报告 32
2.4.1. 报告汇总 32
2.4.2. 报告管脚 33
2.4.3. 报告 HSSI 通道 35
2.4.4. 报告时钟 35
2.4.5. 报告外围位置 36
2.4.6. 报告单元连接性 37
2.4.7. Report Instance Assignments(报告实例分配) 37
2.5. 接口规划修订历史 38

3. 管理器件 I/O 管脚 40
3.1. I/O 规划概述 41
3.1.1. 基本 I/O 规划流程 41
3.1.2. 集成 PCB 设计工具 41
3.1.3. Intel 器件术语 42

3.2. 分配 I/O 管脚.....43
3.2.1. 分配给独占管脚组(Assigning to Exclusive Pin Groups).....44
3.2.2. 分配摆率和驱动强度.....44
3.2.3. 分配差分管脚.....44
3.2.4. 使用 Tcl 命令输入管脚分配.....46
3.2.5. 在 HDL 代码中输入管脚分配.....46
3.3. 导入和导出 I/O 管脚分配.....46
3.3.1. PCB 工具的导入和导出.....47
3.3.2. 移植分配到另一个目标器件.....47
3.4. 验证管脚分配.....48
3.4.1. I/O 分配验证规则.....49
3.4.2. I/O 分配分析(I/O Assignment Analysis).....50
3.4.3. 了解 I/O 分析报告.....52
3.5. 验证 I/O 时序.....53
3.5.1. 运行高级 I/O 时序.....54
3.5.2. 通过容性负载调整 I/O 时序和功耗.....56
3.6. 查看布线和时序延迟.....56
3.7. 脚本编写 API.....57
3.7.1. 生成映射的网表.....57
3.7.2. 保留管脚.....57
3.7.3. 设置位置.....57
3.7.4. Exclusive I/O Group.....58
3.7.5. 摆率和驱动强度.....58
3.8. 管理器件 I/O 管脚修订历史.....58

A. Intel Quartus Prime Pro Edition 用户指南.....60

1. 限制设计

您指定的设计约束，分配和逻辑选项会影响 Intel® Quartus® Prime Compiler 如何实现您的设计。Compiler 尝试以一种满足您的约束的方式对逻辑进行综合和布局。此外，设计约束也影响 Timing Analyzer 和 Power Analyzer 如何进行综合，布局和布线。

您可以在 GUI 中通过脚本或者直接在存储约束的文件中指定设计约束。 Intel Quartus Prime 软件在以下文件中保留在 GUI 中指定的约束：

- Intel Quartus Prime Settings file (`<project_directory>/<revision_name>.qsf`)—包含工程的当前修订版本的工程范围(project-wide)和实例级别(instance-level)分配(Tcl 语法)。一个工程的每个修订版本都有一个`.qsf` 文件。
- Synopsys* Design Constraints 文件(`<project_directory>/<revision_name>.sdc`)—Timing Analyzer 使用行业标准 Synopsys Design Constraint 格式，并将这些约束存储在`.sdc` 文件中。

通过将`.qsf` 文件和`.sdc` 文件的语法与程序化 Tcl 相结合，可以在几种不同的设置下自动执行迭代，更改约束并重新编译。

相关链接

- Intel Quartus Prime Pro Edition 设置文件参考手册
  提供关于 Intel Quartus Prime 软件中所有设置和约束的信息。
- Tcl Scripting
  Intel Quartus Prime Pro Edition 用户指南：脚本
- Command Line Scripting
  Intel Quartus Prime Pro Edition 用户指南：脚本

1.1. 在 GUI 中指定设计约束设计

Intel Quartus Prime 软件提供了可帮助您手动实现工程的工具。这些工具也能支持设计可视化，预填充的参数和窗口交叉探测，加速设计浏览和调试。

当在 Intel Quartus Prime 软件中创建或者更新一个约束时，**Messages** 窗口的 **System** 选项卡显示等效的 Tcl 命令。将这些命令用作将来的脚本化设计定义和编译的参考。

1.1.1. 全局约束和分配

全局约束和工程设置影响整个 Intel Quartus Prime 工程和设计中的所有可应用的逻辑。在早期工程开发中经常定义全局约束；例如，当运行 New Project Wizard 时。 Intel Quartus Prime 软件将全局约束存储在`.qsf` 文件中(one for each project revision)。

英特尔公司。保留所有权利。英特尔、英特尔徽标和其他英特尔标志是英特尔公司或其子公司的商标。依照英特尔的标准保证条例，英特尔保证其 FPGA 和半导体产品的性能符合当前规格，但保留随时更改任何产品和服务的权利，恕不另行通知。英特尔概不承担因应用或使用本文中描述的任何信息、产品或服务而产生的任何责任和义务，除非得到英特尔书面上的明确同意。建议英特尔客户在信赖任何已发布的信息之前以及下单订购产品或服务之前，应先获取最新版本的器件规格。
*其他的名称和品牌可能是其他所有者的资产。

ISO 9001:2015 Registered

表 1.　设置全局约束的 Intel Quartus Prime 工具

Setting Type	New Project Wizard	Device Dialog Box	Settings Dialog Box
Project-wide	X	X	X
Synthesis	X	X	X
Fitter	X	X	X
Simulation			X
Third-party Tools			X
IP Settings			X

相关链接

Managing Project Settings
In *Intel Quartus Prime Pro Edition User Guide: Getting Started*

1.1.2. 节点，实体和实例级约束

节点，实体和实例级约束应用于设计层次结构的子集。这些约束优先于影响设计层次结构相同部分的任何全局分配。

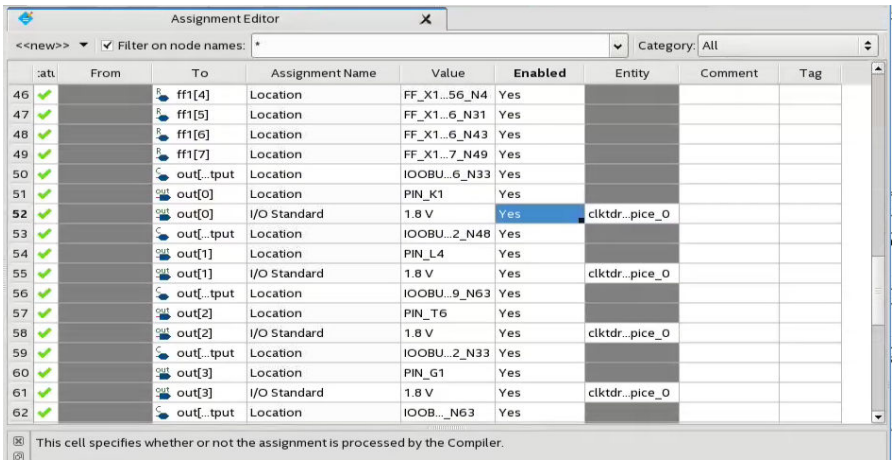
表 2.　设置节点，实体和实例级约束的 Intel Quartus Prime Pro Edition 工具

Assignment Type	Assignment Editor	Interface Planner	Chip Planner	Pin Planner
Pin	X	X		X
Location	X	X	X	
Routing	X		X	
Simulation	X		X	X

1.1.2.1. 在 Assignment Editor 中指定特定实例约束

Intel Quartus Prime Assignment Editor (**Assignments** > **Assignment Editor**)提供一个类似电子表格的界面，用于对所有特定实例设置和约束进行分配。为帮助探索设计，Assignment Editor 使您能够按节点名称和类别对 assignment 进行过滤。

图 1.　Intel Quartus Prime Assignment Editor



	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
46		ff1[4]	Location	FF_X1...56_N4	Yes			
47		ff1[5]	Location	FF_X1...6_N31	Yes			
48		ff1[6]	Location	FF_X1...6_N43	Yes			
49		ff1[7]	Location	FF_X1...7_N49	Yes			
50		out[...tput	Location	IOOBU...6_N33	Yes			
51		out[0]	Location	PIN_K1	Yes			
52		out[0]	I/O Standard	1.8 V	Yes	clktdr...pice_0		
53		out[...tput	Location	IOOBU...2_N48	Yes			
54		out[1]	Location	PIN_L4	Yes			
55		out[1]	I/O Standard	1.8 V	Yes	clktdr...pice_0		
56		out[...tput	Location	IOOBU...9_N63	Yes			
57		out[2]	Location	PIN_T6	Yes			
58		out[2]	I/O Standard	1.8 V	Yes	clktdr...pice_0		
59		out[...tput	Location	IOOBU...2_N33	Yes			
60		out[3]	Location	PIN_G1	Yes			
61		out[3]	I/O Standard	1.8 V	Yes	clktdr...pice_0		
62		out[...tput	Location	IOOB..._N63	Yes			

使用 Assignment Editor:

- 添加，编辑或者删除所选节点的分配
- 显示关于特定分配的信息
- 使能或者禁用单个分配
- 对分配添加注释

此外，您可以将分配导出到一个 Comma-Separated Value File (`.csv`)中。

1.1.2.2. 指定多维总线约束

Intel Quartus Prime Pro Edition 软件通常仅支持 1 维和 2 维总线名称用于指定约束。 Intel Quartus Prime Pro Edition 19.3 版本及后续版本可支持多维总线名称，以实现更有效的约束。

例如，您可以指定以下分配，将一个约束应用到 `reg [31:0] r [0:2][4:5]`三维总线中的所有比特:

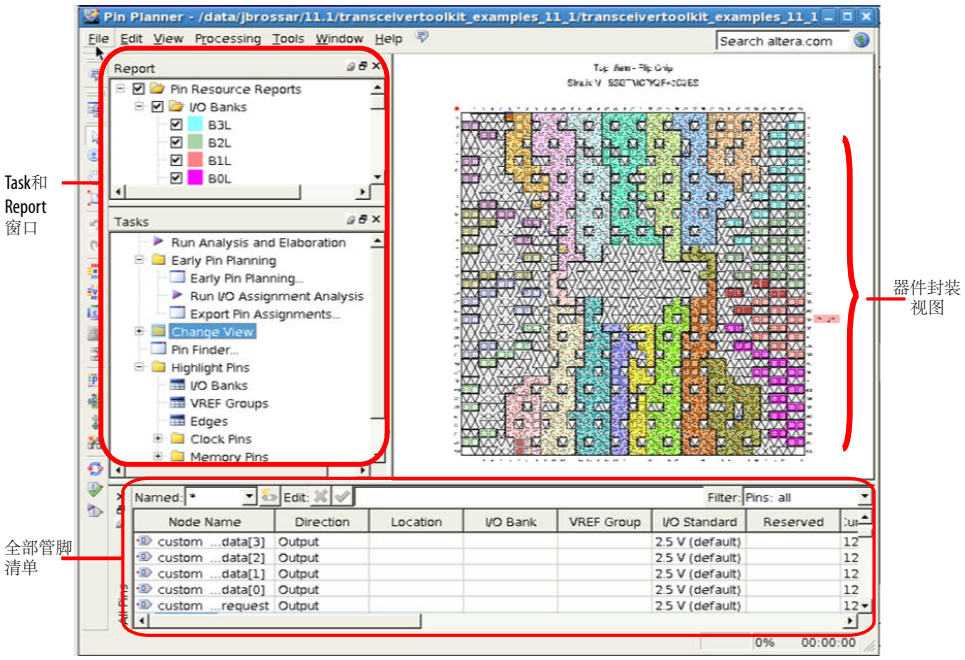
```
set_instance_assignment -name PRESERVE_REGISTER ON -to r
```

此约束然后应用到所有比特 `r: [0][4][31], r[0][4][30], … , r[1][5][0]`。

1.1.2.3. 在 Pin Planner 中指定 I/O 约束

Intel Quartus Prime Pin Planner 使您能够将设计单元分配给 I/O 管脚。您也可以对设计中未定义的 IP 接口或者用户节点进行布局和分配。

图 2. Pin Planner GUI



相关链接

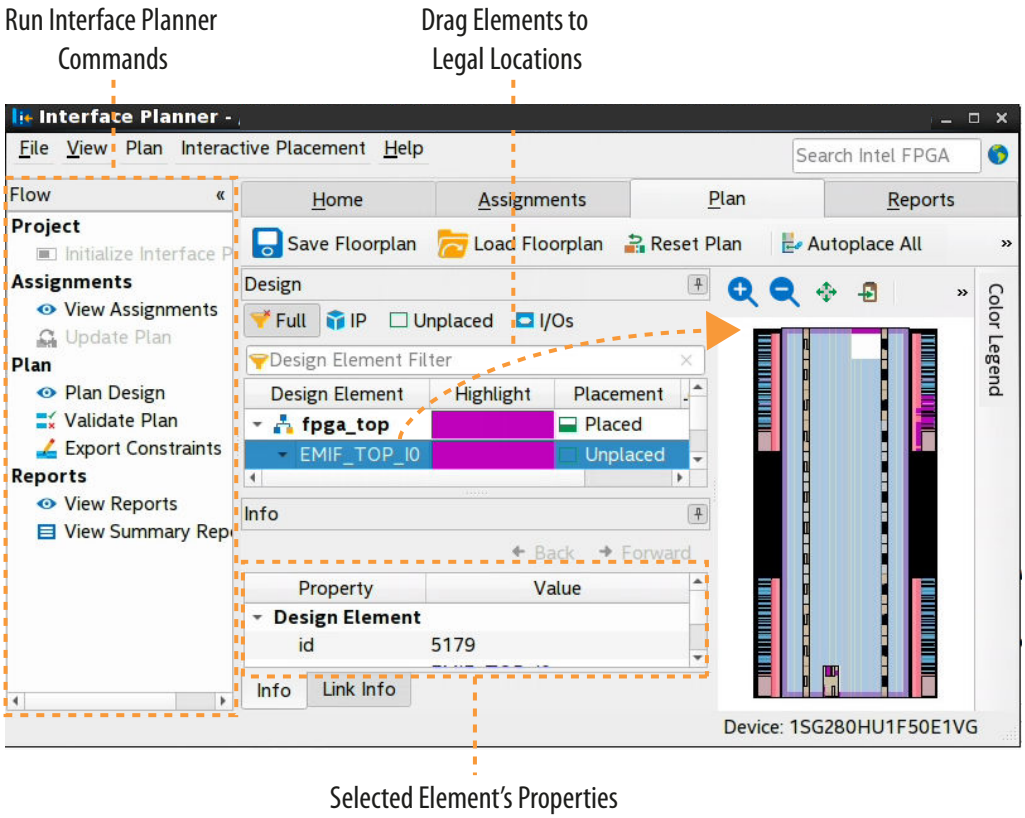
管理器件 I/O 管脚 (第 40 页)

1.1.2.4. 在 Interface Planner 中的规划接口约束

Interface Planner 简化了物理实现的准确约束的规划。使用 Interface Planner 对接口实现进行原型设计，规划时钟，并快速定义一个合法的器件平面图。

Interface Planner 与 Intel Quartus Prime Fitter 进行动态交互，准确地验证布局合法性。您可以使用交互式报表来评估不同的平面图，从而无需迭代编译即可准确地规划最佳实现。Fitter 验证可确保接口规划与实际实现结果之间的最高关联性。您可以在最终的实现中将接口规划约束应用到您的工程。

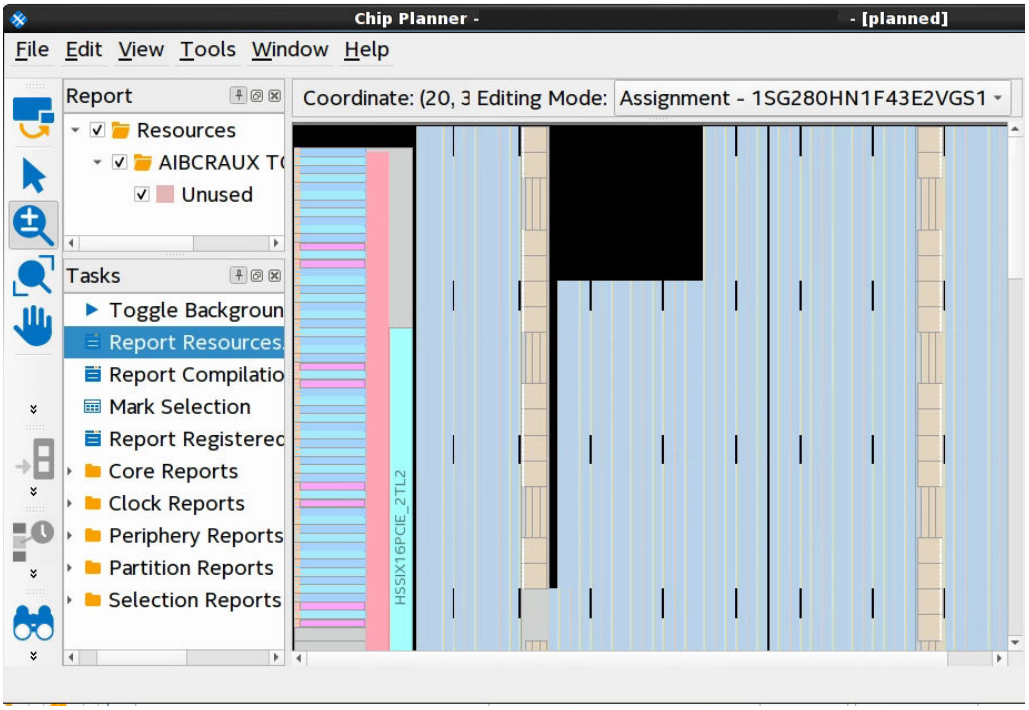
图 3. Interface Planner GUI



1.1.2.5. 使用 Chip Planner 调整约束

通过使用 Chip Planner，您能够以器件平面图的图形表示形式来调整对器件资源(例如：管脚，逻辑单元和 LAB)的现有分配。通过拖放到 **Logic Lock Regions Window** 中的 Logic Lock 区域，您也可以查看公式和布线信息以及降级分配(demote assignment)。

图 4. **Chip Planner GUI**



相关链接

Design Floorplan Analysis in the Chip Planner
In *Intel Quartus Prime Pro Edition User Guide: Design Optimization*

1.1.2.6. 使用 Design Partition Planner 对设计进行约束

Design Partition Planner 使您能够查看设计连接性和层次结构，并可以帮助您创建有效的设计分区。

此外，Design Partition Planner 还使您能够通过逐个分区地对故障路径进行隔离和解决来优化设计性能。

相关链接

Creating Partitions and Logic Lock Regions with the Design Partition Planner and the Chip Planner
In *Intel Quartus Prime Pro Edition User Guide: Design Optimization*

1.1.3. Intel Quartus Prime GUI 的组件之间的探测

Intel Quartus Prime 软件使您能够在窗口和源文件中定位节点和实例。

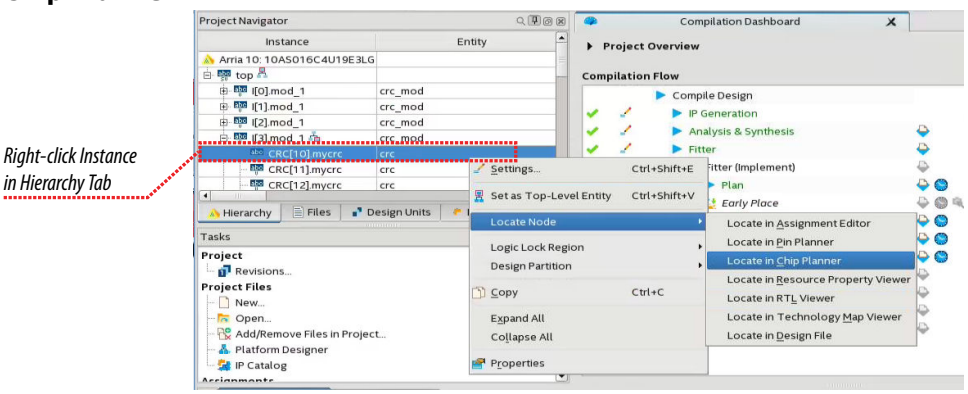
当在 Project Navigator，Assignment Editor，Chip Planner 或者 Pin Planner 中，并想要在其他 Intel Quartus Prime 工具中显示一个特定资源时：

1. 右键点击想要显示的资源。
2. 点击 **Locate Node**，然后点击任何菜单选项。

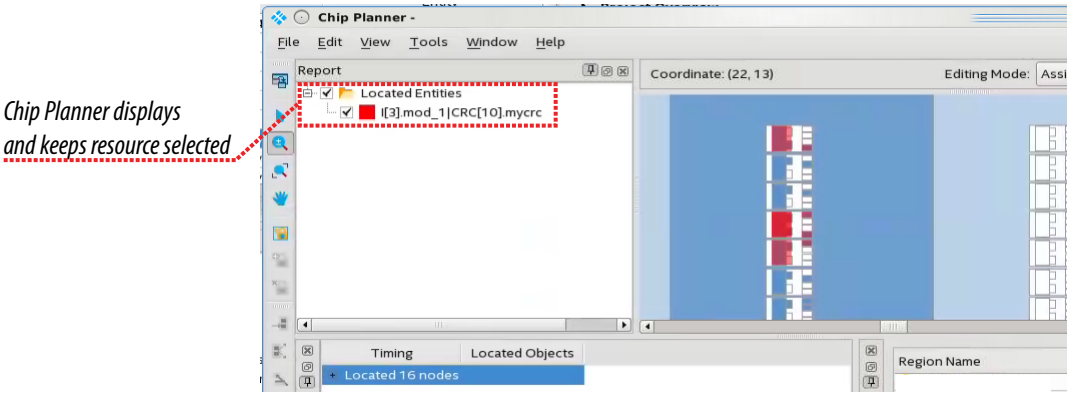
对应的窗口打开—或者显示在前景中(如果已打开)—并显示所单击的单元。

实例-1: 定位一个在 Project Navigator 中选择的资源

在 **Hierarchy** 选项卡的 **Entity** 列表中，右键点击任何对象，然后点击 **Locate** > **Locate in Chip Planner**。



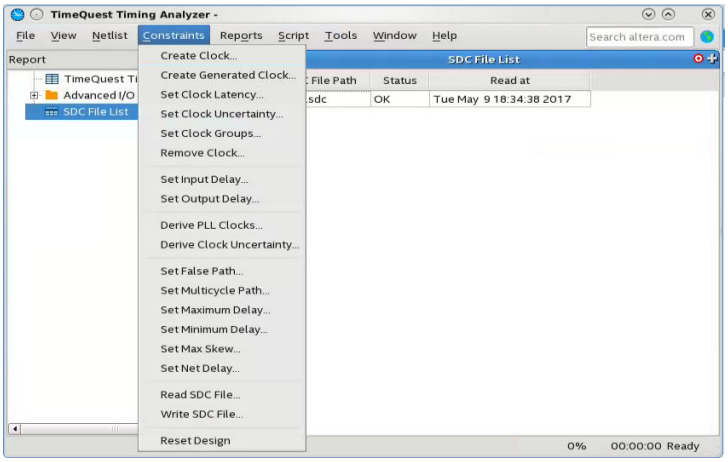
Chip Planner 打开并显示所选择的实例。



1.1.4. 在 GUI 中指定时序约束

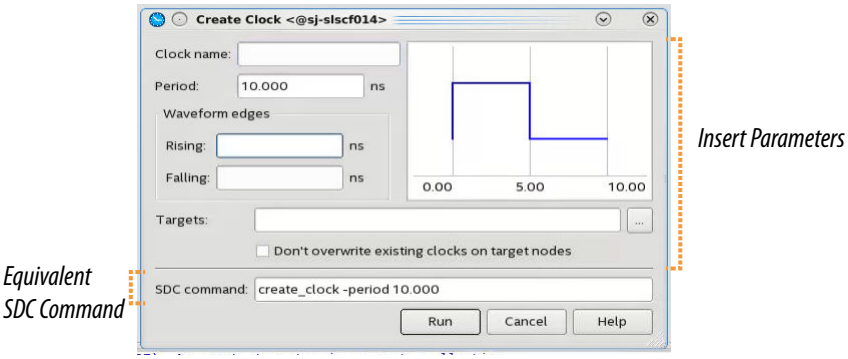
您可以在 Timing Analyzer GUI 中指定时序约束。点击 Timing Analyzer 中的 Constraints 菜单，指定能够应用于工程的时序约束。

图 5. Timing Analyzer 中的 Constraint 菜单



当您在 GUI 中指定一个约束时，对话框显示等效的 SDC 命令语法。

实例-2: Create Clock 对话框



单独的时序分配(timing assignment)会覆盖整个工程范围的要求。

- 为避免报告不正确或不相关的时序违规，可以将时序异常分配给节点和路径。
- Timing Analyzer 支持点对点时序约束，通配符以在创建约束时标识特定节点，以及分配组以对节点组进行单独约束。

相关链接

Using Timing Constraints
提供了所有 Constraints 菜单命令的描述

1.2. 使用 Tcl 脚本约束设计

您可以使用 `.sdc` 和 `.qsf` 设置文件执行所有设计分配。要将这些文件集成到编译和优化流程中，请使用 Tcl 脚本。即使 `.sdc` 和 `.qsf` 文件是使用 Tcl 语法编写的，但它们本身也是不可执行的。

当使用 Intel Quartus Prime Tcl package 时，您的脚本可以打开工程，进行分配，编译设计，并将编译结果与已知目标和基准进行比较。此外，这样的脚本可以通过修改约束并重新编译设计来使迭代设计过程自动化。

1.2.1. 创建一个工程并应用约束

命令行可执行文件包括通用全局工程设置和命令的选项。您可以使用 Tcl 脚本来应用约束，例如管脚位置和时序分配。您可以编写一个 Tcl 约束文件，或者通过点击 **Project** > **Generate Tcl File for Project** 对现有工程生成一个 Tcl 文件。

此示例使用一个 Tcl 脚本创建一个工程，并使用< Intel Quartus Prime *installation directory*> `/qdesigns/fir_filter/`目录中的教学设计文件应用工程约束。

```
project_new filtref -overwrite
# Assign family, device, and top-level file
set_global_assignment -name FAMILY "Arria 10"
set_global_assignment -name DEVICE <Device>
set_global_assignment -name VERILOG_FILE filtref.v
# Assign pins
set_location_assignment -to clk Pin_28
set_location_assignment -to clkx2 Pin_29
set_location_assignment -to d[0] Pin_139
set_location_assignment -to d[1] Pin_140
#
project_close
```

将此脚本保存在一个名为 `setup_proj.tcl` 的文件中，并在命令提示符下键入示例中所示的命令，以创建设计，应用约束，编译设计以及执行快速角和慢角时序分析。时序分析结果保存在两个文件中：`filtref_sta_1.rpt` 和 `filtref_sta_2.rpt` 。

```
quartus_sh -t setup_proj.tcl
quartus_syn filtref
quartus_fit filtref
quartus_asm filtref
quartus_sta filtref --model=fast --export_settings=off
mv filtref_sta.rpt filtref_sta_1.rpt
quartus_sta filtref --export_settings=off
mv filtref_sta.rpt filtref_sta_2.rpt
```

键入以下命令以创建设计，应用约束并编译设计，而无需执行时序分析：

```
quartus_sh -t setup_proj.tcl
quartus_sh --flow compile filtref
```

`quartus_sh --flow compile` 命令执行一个完整编译，等效于单击工具栏中的 **Start Compilation** 按钮。

1.2.2. 分配一个管脚

要将一个信号分配给一个管脚或者器件位置，请使用此示例中显示的 Tcl 命令：

```
set_location_assignment -to <signal name> <location>
```

有效的位置是管脚位置名称。某些器件系列也支持边沿和 I/O bank 位置。边沿位置是 `EDGE_BOTTOM`、`EDGE_LEFT`、`EDGE_TOP` 和 `EDGE_RIGHT`。I/O bank 位置包括 `IOBANK_1` 到 `IOBANK_n`，其中 n 是器件中 I/O bank 的数量。

1.2.3. 生成 Intel Quartus Prime 设置文件

Intel Quartus Prime 软件使您能够从修订版生成.qsf 文件。您可以将这些约束嵌入脚本编写的编译流程中，甚至可以创建.qsf 文件以进行设计优化。

要从 Intel Quartus Prime 软件生成一个.qsf 文件，请点击 **Assignments** > **Export Assignments**。

要以可读形式组织.qsf，请点击 **Project** > **Organize Intel Quartus Prime Settings File**。

实例-3: 已组织的.qsf 文件

此示例显示.qsf 文件如何表征一个设计修订版。set_global_assignment 命令进行所有的全局约束和软件设置，set_location_assignment 将设计中的每个 I/O 节点约束成器件上的一个物理管脚。

```
# Project-Wide Assignments
# ========================
set_global_assignment -name SYSTEMVERILOG_FILE top.sv
set_global_assignment -name SYSTEMVERILOG_FILE blinking_led.sv
set_global_assignment -name SDC_FILE blinking_led.sdc
set_global_assignment -name SDC_FILE jtag.sdc
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name LAST_QUARTUS_VERSION "17.1.0 Pro Edition"
set_global_assignment -name TEXT_FILE blinking_led_generated.txt
# Pin & Location Assignments
# ==========================
set_location_assignment PIN_AN18 -to clock
set_location_assignment PIN_AR23 -to led_zero_on
set_location_assignment PIN_AM21 -to led_two_on
set_location_assignment PIN_AR22 -to led_one_on
set_location_assignment PIN_AL20 -to led_three_on
# Analysis & Synthesis Assignments
# ================================
set_global_assignment -name FAMILY "Arria 10"
set_global_assignment -name TOP_LEVEL_ENTITY top
# Fitter Assignments
# ==================
set_global_assignment -name DEVICE 10AS066N3F40E2SG
# -----------------
# start ENTITY(top)
# Fitter Assignments
# ==================
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_zero_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_one_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_two_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_three_on
set_instance_assignment -name SLEW_RATE 1 -to led_zero_on
set_instance_assignment -name SLEW_RATE 1 -to led_one_on
set_instance_assignment -name SLEW_RATE 1 -to led_two_on
set_instance_assignment -name SLEW_RATE 1 -to led_three_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to clock
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_zero_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_one_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_two_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_three_on
# end ENTITY(top)
# ---------------
```

相关链接

Intel Quartus Prime Pro Edition 设置文件参考手册
提供关于 Intel Quartus Prime 软件中所有设置和约束的信息。

1.2.4. Synopsys Design Constraint (.sdc)文件

Intel Quartus Prime 软件在`.sdc`文件中保存时序约束(使用 Tcl 语法)。您可以将这些约束嵌入脚本编写的编译流程中，甚至可以创建`.sdc`文件集以进行设计优化。

实例-4: .sdc 文件

以下示例显示一个小型设计的时序约束。

```
## PROGRAM "Quartus Prime"
## VERSION "Version 17.1.0 Internal Build 91 05/07/2017 SJ Pro Edition"
## DATE    "Wed May 10 14:22:08 2017"
##
## DEVICE  "10AX115R4F40I3SG"
##
#**************************************************************
# Time Information
#**************************************************************
set_time_format -unit ns -decimal_places 3
#**************************************************************
# Create Clock
#**************************************************************
create_clock -name {clk_in} -period 10.000 -waveform { 0.000 5.000 } [get_ports
{clk_in}]
#**************************************************************
# Create Generated Clock
#**************************************************************
derive_pll_clocks
#**************************************************************
# Set Clock Uncertainty
#**************************************************************
derive_clock_uncertainty
#**************************************************************
# Set Input Delay
#**************************************************************
set_input_delay -add_delay  -clock [get_clocks {clk_in}]  1.500 [get_ports
{async_rst}]
set_input_delay -add_delay  -clock [get_clocks {clk_in}]  1.200 [get_ports
{data_in}]
#**************************************************************
# Set Output Delay
#**************************************************************
set_output_delay -add_delay  -clock [get_clocks {clk_in}]  2.000 [get_ports
{data_out}]
#**************************************************************
# Set Multicycle Path
#**************************************************************
set_multicycle_path -setup -end -from [get_keepers *] -to [get_keepers {reg2}] 2
```

相关链接

Constraining and Analyzing with Tcl Commands
In *Intel Quartus Prime Pro Edition User Guide: Timing Analyzer*

1.2.5. Tcl-only 脚本流程

作为`.sdc`和`.qsf`文件的替代，您可以在 Tcl 脚本内执行所有设计分配和时序约束。在这种情况下，自动执行编译和自定义结果报告的脚本也包含设计约束。

您可以将设计的内容导出到一个程序可执行的 Tcl (`.tcl`)文件，然后在尝试其他约束条件后使用生成的脚本还原设置。

要将约束导出为可执行的 Tcl 脚本，请单击 **Project** > **Generate Tcl File for Project**。

实例-5: blinking_led_generated.tcl 文件

```
# Quartus Prime: Generate Tcl File for Project
# File: blinking_led_generated.tcl
# Generated on: Wed May 10 10:14:44 2017
# Load Quartus Prime Tcl Project package

package require ::quartus::project
set need_to_close_project 0
set make_assignments 1
# Check that the right project is open
if {[is_project_open]} {
  if {[string compare $quartus(project) "blinking_led"]} {
puts "Project blinking_led is not open"
set make_assignments 0
  }
} else {
  # Only open if not already open
  if {[project_exists blinking_led]} {
project_open -revision blinking_led blinking_led
  } else {
project_new -revision blinking_led blinking_led
  }
  set need_to_close_project 1
}

# Make assignments
if {$make_assignments} {
set_global_assignment -name SYSTEMVERILOG_FILE top.sv
set_global_assignment -name SYSTEMVERILOG_FILE blinking_led.sv
set_global_assignment -name SDC_FILE blinking_led.sdc
set_global_assignment -name SDC_FILE jtag.sdc
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY output_files
set_global_assignment -name LAST_QUARTUS_VERSION "17.1.0 Pro Edition"
set_global_assignment -name TEXT_FILE blinking_led_generated.txt
set_global_assignment -name FAMILY "Arria 10"
set_global_assignment -name TOP_LEVEL_ENTITY top
set_global_assignment -name DEVICE 10AS066N3F40E2SG
set_location_assignment PIN_AN18 -to clock
set_location_assignment PIN_AR23 -to led_zero_on
set_location_assignment PIN_AM21 -to led_two_on
set_location_assignment PIN_AR22 -to led_one_on
set_location_assignment PIN_AL20 -to led_three_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_zero_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_one_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_two_on
set_instance_assignment -name IO_STANDARD "1.8 V" -to led_three_on
set_instance_assignment -name SLEW_RATE 1 -to led_zero_on
set_instance_assignment -name SLEW_RATE 1 -to led_one_on
set_instance_assignment -name SLEW_RATE 1 -to led_two_on
set_instance_assignment -name SLEW_RATE 1 -to led_three_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to clock
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_zero_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_one_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_two_on
set_instance_assignment -name CURRENT_STRENGTH_NEW 12MA -to led_three_on
# Commit assignments
export_assignments
# Close project
if {$need_to_close_project} {
project_close
}
}
```

示例：

- 打开工程
- 分配约束
- 将分配写入到 QSF 文件中
- 关闭工程

1.2.5.1. Tcl-only Timing Analysis

为避免使用单独的文件来保持时序约束，请将`.sdc`文件复制并粘贴到可执行的时序分析脚本中。

1.3. 一个完全迭代的脚本流程

Intel Quartus Prime Tcl API 中的**::quartus::flow** Tcl package 使您能够修改设计约束并在迭代流程中进行重新编译。

相关链接

- ::quartus::flow
  Intel Quartus Prime Help 中
- Command Line Scripting
  In *Intel Quartus Prime Pro Edition 用户指南：脚本*

1.4. 约束设计修订历史

文档版本	Intel Quartus Prime 版本	修订内容
2019.10.16	19.3.0	• 添加了"Specifying Multi-Dimensional Bus Constraints"主题。 • 更新了"Create a Project and Apply Constraints"中的示例。
2019.08.21	18.1.0	纠正了"Tcl-only Script Flows"主题中的次要拼写错误。
2019.01.04	18.1.0	• 阐明了"Constraining Designs"主题中的.sdc and .qsf 文件的默认位置。 • 增添了"Plan Interface Constraints with Interface Planner"主题。 • 增添了"Constrain Designs with the Pin Planner"和"Constrain Designs with the Chip Planner"的屏幕截图。 • 增添了两个新的"Assigning a Pin"和"Creating a Project and Applying Constraints"主题，显示 Tcl 示例。 • 在 Timing Analyzer UG 中添加了 Using Timing Constraints 主题的链接，解释了所有命令。
2017.11.06	17.1.0	• 将 Constraining Designs with the GUI 主题重命名为 Constraining Designs with Quartus Prime Tools。 • 将 Global Constraints 主题重命名为 Global Constraints and Assignments。 • 添加了表：Quartus Prime Tools to Set Global Constraints。 • 删除了主题：Common Types of Global Constraints。 • 删除了主题：Settings That Direct Compilation and Analysis Flows。 • 更新了主题：Node, Entity and Instance-Level Constraints。 • 添加了表：Quartus Prime Tools to Set Node, Entity and Instance Level Constraints。 • 添加了主题：Assignment Editor。 • 更新了主题：Constraining Designs with the Pin Planner。 • 更新了主题：Constraining Designs with the Chip Planner。

继续...

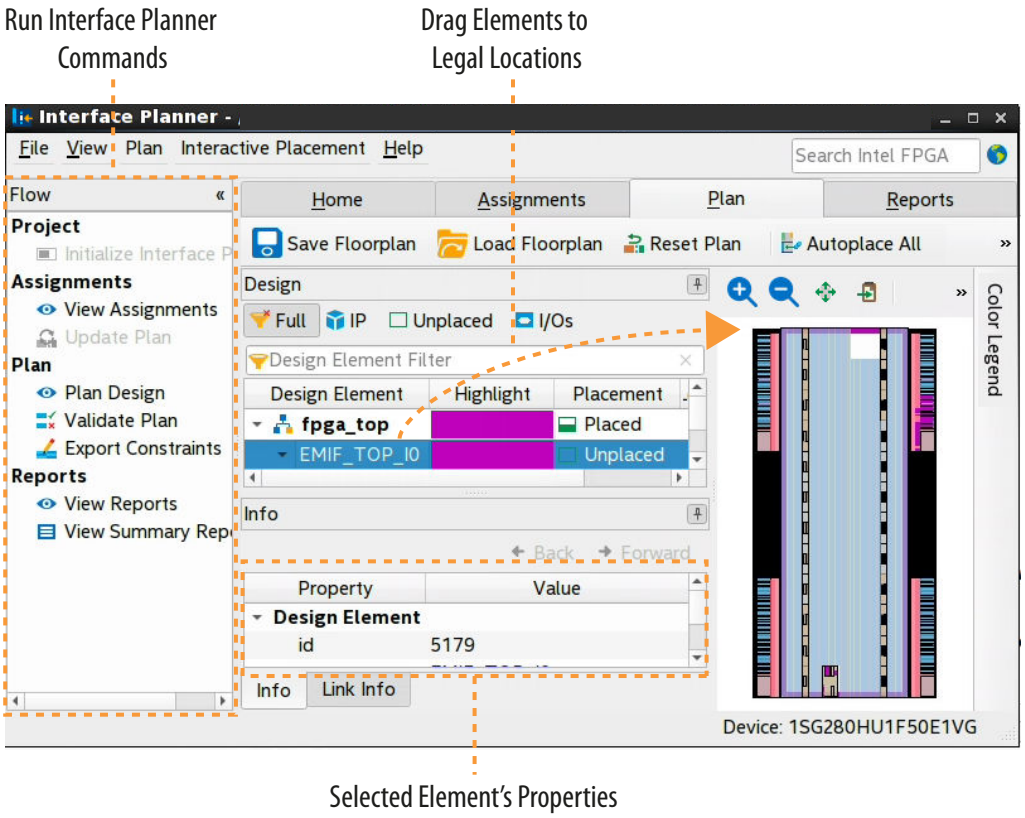
文档版本	Intel Quartus Prime 版本	修订内容
		• 添加了主题：Constraining designs with the Design Partition Planner。 • 更新了主题：Probing Between Components of the Quartus Prime GUI。 • 增添了示例：Locate a Resource Selected in the Project Navigator。 • 更新了主题：SDC and the Timing Analyzer， 并重命名为 Specifying Individual Timing Constraints。 • 添加了图：Constraint Menu in Timing Analyzer。 • 增添了示例：Create Clock Dialog Box。 • 更新了主题：Constraining Designs with Tcl，并重命名为 Constraining Designs with Tcl Scripts。 • 更新了主题：Quartus Prime Settings Files and Tcl，并重命名为 Generating Quartus Prime Settings Files。 • 增添了示例：blinking_led.qsf File。 • 更新了主题：Timing Analysis with Synopsys Design Constraints and Tcl，并重命名为 Timing Analysis with .sdc Files and Tcl Scripts。 • 添加了示例：.sdc File with Timing Constraints。 • 增添了主题：Tcl-only Script Flows。 • 更新了主题：A Fully Iterative Scripted Flow。
2017.05.08	17.0.0	• 删除了对不推荐使用的 Fitter Effort 逻辑选项的引用。
2016.10.31	16.1.0	• 进行了 Intel 重命名。
2015.11.02	15.1.0	• 将 *Quartus II* 更改成 *Intel Quartus Prime*。
2014 年 6 月	14.0.0	格式更新。
2012 年 11 月	12.1.0	更新了任务和报告窗口的 Pin Planner 描述。
2012 年 6 月	12.0.0	删除了调查链接。
2011 年 11 月	10.0.2	模板更新。
2010 年 12 月	10.0.1	模板更新。
2010 年 7 月	10.0.0	重写章节以更广泛地涵盖所有设计约束方法。删除了程序步骤和用户接口详细信息，并替换为 Quartus II Help 的链接。
2009 年 11 月	9.1.0	• 添加了两个注释。 • 少量文本编辑
2009 年 3 月	9.0.0	• 修改并重新整理了整章。 • 添加了“Probing to Source Design Files and Other Quartus Windows”部分。 • 添加了节点类型图标的描述(Table1 – 3)。 • 添加了通配符说明。
2008 年 11 月	8.1.0	更改成 8½” × 11”页面大小。无内容变更。
2008 年 5 月	8.0.0	更新了 Quartus II 8.0 修订版和日期。

2. 接口规划

接口规划—接口物理约束的可行性分析—是高级 FPGA 设计的基本早期步骤。外围布局可能是一个涉及很多变量的复杂过程。 Intel Quartus Prime Interface Planner 简化了物理实现的准确约束的规划。使用 Interface Planner 对接口实现进行原型设计，规划时钟，并快速定义一个合法的器件平面图。

Interface Planner 与 Intel Quartus Prime Fitter 进行动态交互，准确地验证布局合法性。您可以使用交互式报表来评估不同的平面图，从而无需迭代编译即可准确地规划最佳实现。Fitter 验证可确保接口规划与实际实现结果之间的最高关联性。您可以在最终的实现中将接口规划约束应用到您的工程。

图 6. Interface Planner GUI



英特尔公司。保留所有权利。英特尔、英特尔徽标和其他英特尔标志是英特尔公司或其子公司的商标。依照英特尔的标准保证条例，英特尔保证其 FPGA 和半导体产品的性能符合当前规格，但保留随时更改任何产品和服务的权利，恕不另行通知。英特尔概不承担因应用或使用本文中描述的任何信息、产品或服务而产生的任何责任和义务，除非得到英特尔书面上的明确同意。建议英特尔客户在信赖任何已发布的信息之前以及下单订购产品或服务之前，应先获取最新版本的器件规格。
*其他的名称和品牌可能是其他所有者的资产。

ISO 9001:2015 Registered

Interface Planner 特性

Interface Planner 提供以下支持：

- 在设计过程的任何阶段，规划合法的设计外围平面图。
- 分析和修改时钟网络方案以指导高扇出信号的布局。
- 使用 Fitter 自动评估所有布局合法性。
- 保存并重新加载各种平面图文件。
- GUI 和 Tcl 命令行操作。

注意: Intel 现在将 BluePrint 称为 Interface Planner。

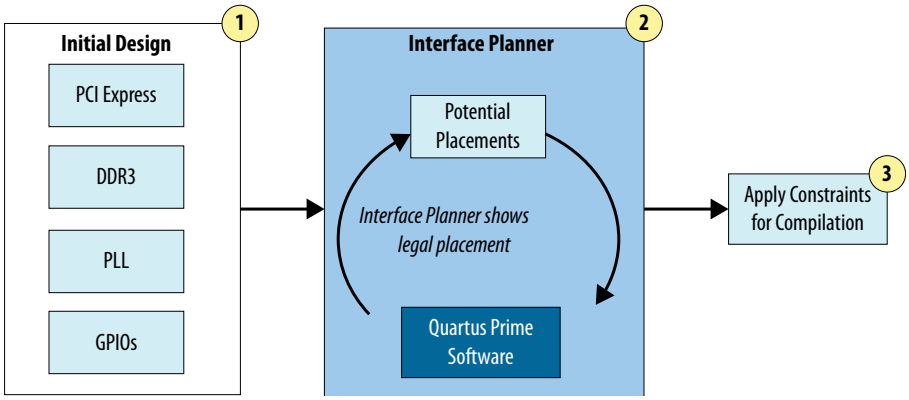
相关链接

Video Demo: Using Interface Planner to Place DDR-3 and PCI Express Gen3

2.1. 接口规划概述

设计综合后，使用 Interface Planner 快速定义一个合法的器件平面图。

图 7. Interface Planner 简化合法布局



Intel FPGA 包含内核和外围器件位置。器件内核位置是自适应查找表(ALUT)，内核触发器，RAM 和数字信号处理器(DSP)。器件外围位置包括 I/O 单元，锁相环(PLL)，时钟缓冲器和硬核处理器系统(HPS)。

Intel FPGA 在器件外围包含了很多硅功能，例如：硬核 PCI Express® IP core，高速收发器，硬核存储器接口电路和嵌入式处理器。这些外围单元之间的相互作用可能很复杂。Interface Planner 简化了这种复杂性，并使您能够快速可视化和布局 I/O 接口和外围单元，例如：

- I/O 单元
- LVDS 接口
- PLL
- 时钟
- 硬接口 IP Core

- 高速收发器
- 硬盘存储器接口 IP Core
- 嵌入式处理器

初始化之后，Interface Planner 显示工程的逻辑层次结构，后综合(post-synthesis)设计单元和 Fitter 创建的设计单元，以及目标器件位置。GUI 支持在平面图中布局设计单元的多种方法。在平面图中布局单元时，Fitter 实时验证合法性以确保与最终实现的准确关联。

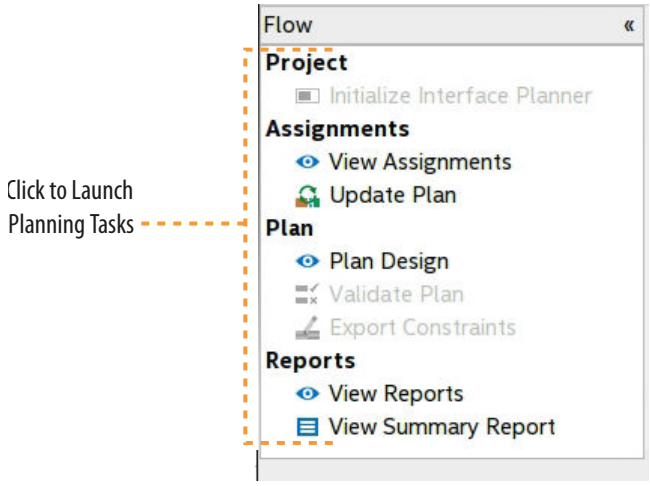
相关链接

管理器件 I/O 管脚 (第 40 页)

2.2. 使用 Interface Planner

Interface Planner 的用户界面可指导您完成设计规划步骤。使用 Interface Planner 的 **Flow** 控制来按顺序执行流程的主要初始化，规划和验证功能。

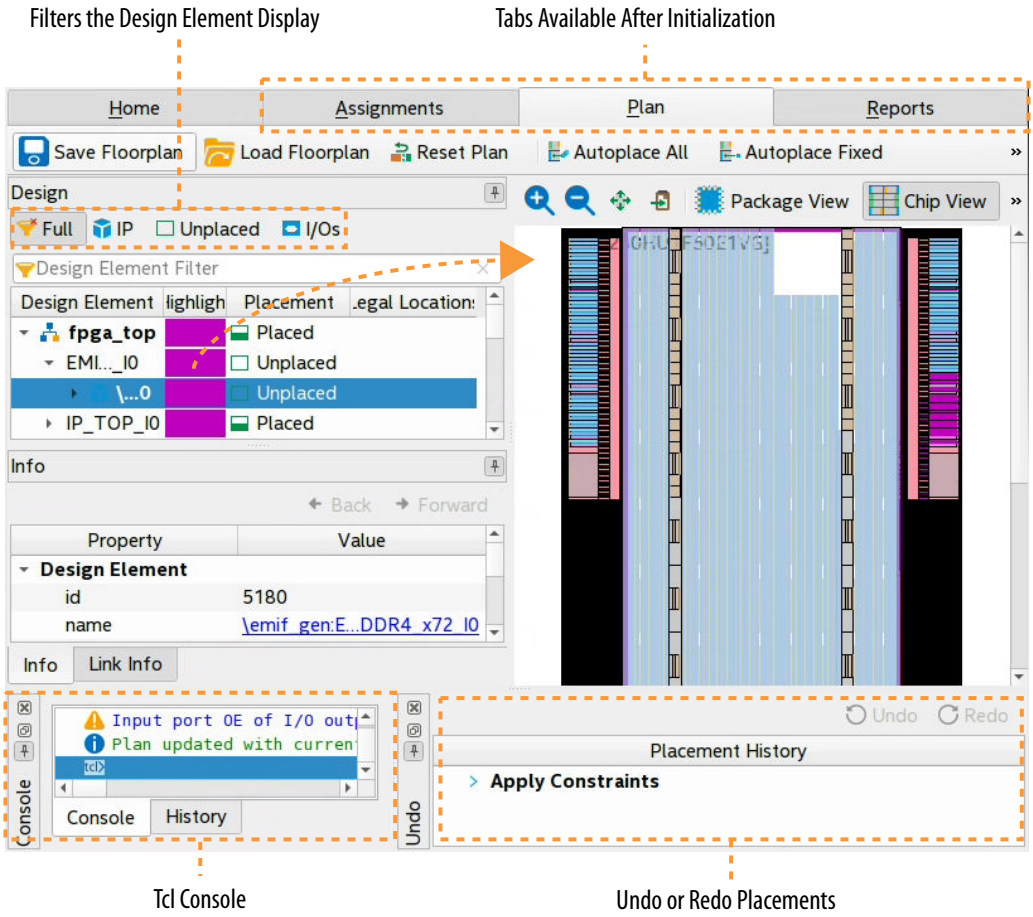
图 8. Interface Planner 流程控制



当运行 **Flow** 控制中的每个步骤时，下游命令和 **Assignments**，**Plan** 和 **Reports** 选项卡变为可用。Interface Planner 仅允许您在完成流程中的所有必备步骤之后运行命令。

Initialize Interface Planner 之后，系统会提示您确认在规划开始之前所做的所有工程分配。在 **Assignments** 选项卡上禁止或者使能的任何导入的功能分配来解决任何冲突并评估不同的实现。

图 9. **Interface Planner GUI**



使用工程分配 **Update Plan**(更新规划)后，即可在 **Plan** 选项卡上的目标器件 **Chip View** 或者 **Package View** 上布局设计单元。在 **Plan** 选项卡上布局设计单元时，Fitter 实时验证布局合法性。一旦布局完成并通过验证，可以作为 Tcl 脚本导出约束，用于工程中的应用程序。

注意: 通过生成的 Tcl 脚本将定义的 Interface Planner 约束导出并获取之前，这些约束不会应用到工程中。

图 10. Interface Planner Chip View

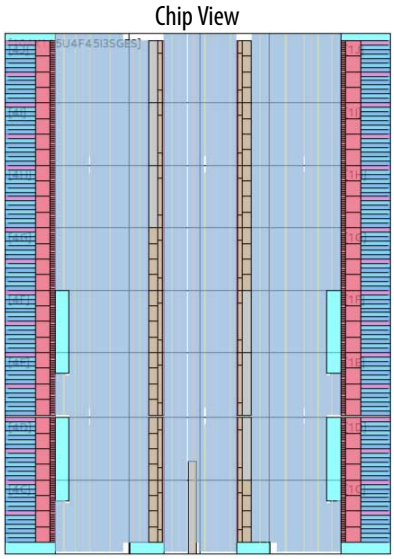
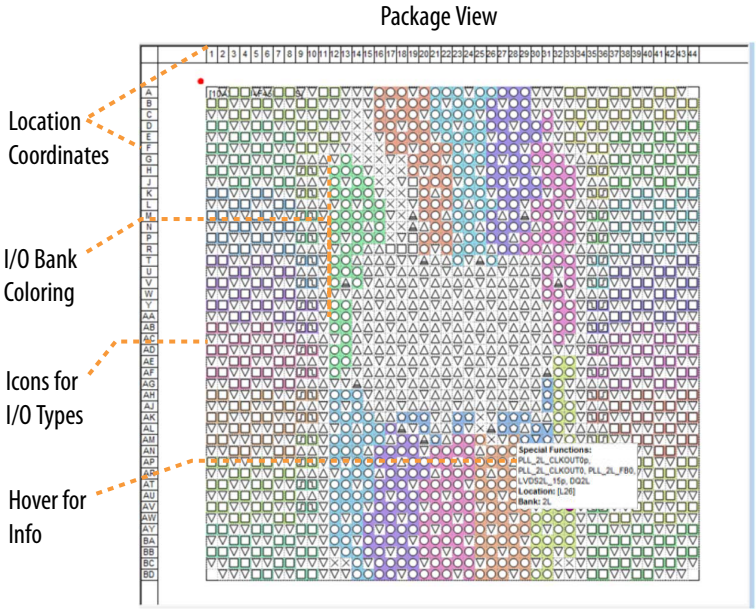


图 11. Interface Planner Package View



以下主题详细描述了这些接口规划流程步骤：

步骤 1：设置和综合工程 (第 22 页)
步骤 2：初始化 Interface Planner (第 22 页)
步骤 3：使用工程分配更新规划 (第 22 页)
步骤 4：规划外围布局 (第 23 页)
步骤 5：报告布局数据 (第 27 页)
步骤 6：验证和导出规划约束 (第 27 页)

2.2.1. 步骤 1：设置和综合工程

Interface Planner 需要至少一个部分完整的，综合的 Intel Quartus Prime 工程作为输入。您也可以使用 Interface Planner 对布局进行调整，以实现一个完全完整的设计工程。

请按照以下步骤设置工程并运行综合：

1. 请完成至少以下步骤：
   - 完全定义已知的器件外围接口。
   - 例化所有已知的接口 IP 内核。
   - 声明所有通用 I/O。
   - 对所有通用 I/O 定义 I/O 标准，电压，驱动强度和摆率。
   - 定义内核时钟(可选的，但是建议的)。
   - 将外围 IP 的所有接口连接到虚拟管脚或测试逻辑。此技术在外壳设计中的所有接口上都创建了环回，有助于确保外围接口在综合优化后仍然存在。
2. 要综合设计，请点击 **Processing** > **Start** > **Start Analysis & Synthesis**。在运行 Interface Planner 之前必须至少运行 Analysis & Synthesis。

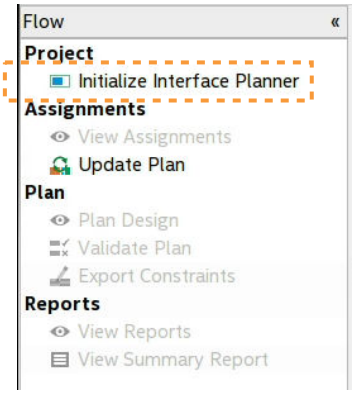
2.2.2. 步骤 2：初始化 Interface Planner

初始化 Interface Planner 将加载用于综合快照的编译数据，并使能 **View Assignments** 命令和 **Assignments** 选项卡以调解工程分配。

初始化 Interface Planner：

1. 点击 **Tools** > **Interface Planner** 。Interface Planner 打开，显示 **Home** 选项卡。
2. 在 **Flow** 控制中，点击 **Initialize Interface Planner** 。初始化后，Fitter 在您进行更改时会动态地验证接口规划。

图 12. Interface Planner Home Tab



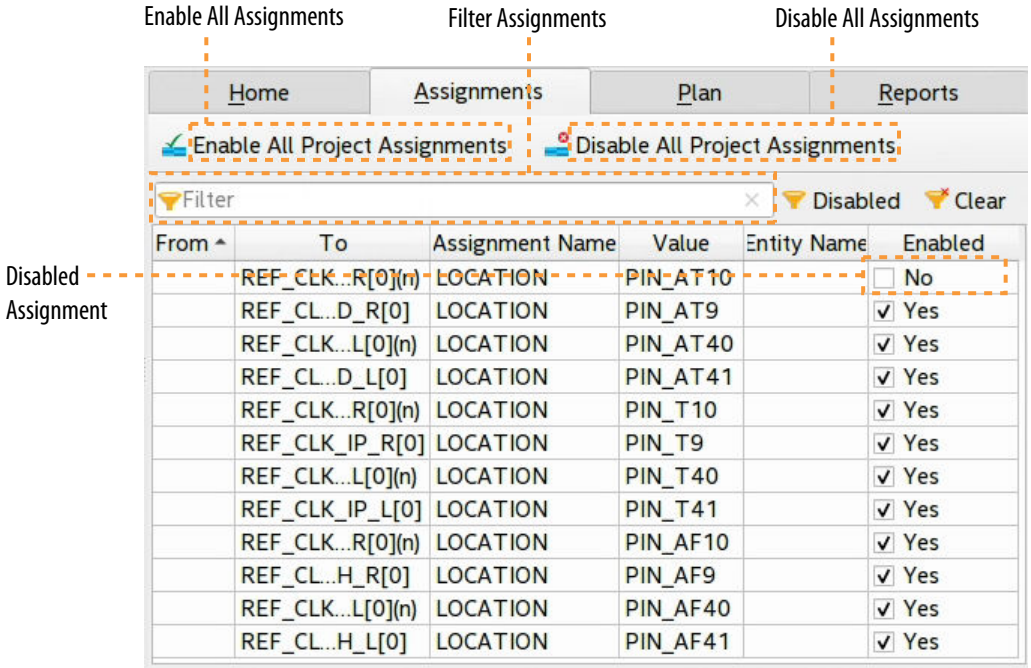
2.2.3. 步骤 3：使用工程分配更新规划

在 Interface Planner 中进行外围规划之前，必须调解所有冲突的导入工程分配，并使用想要保留在规划中的分配来 **Update Plan**(更新规划)。

请按照以下步骤检查导入的工程分配并调解所有冲突：

1. 在 **Flow** 控制上，点击 **View Assignments**。
2. 在 **Assignments** 选项卡上，使能或禁用特定的工程分配或一组工程分配，以解决任何冲突或尝试使用不同的设置。您可以按分配名称或状态过滤分配列表。
3. 解决所有冲突之后，在 **Flow** 控制上单击 **Update Plan**，将使能的工程分配应用于接口规划。

图 13. Interface Planner (Assignments Tab)



相关链接

- Home 选项卡控制 (第 29 页)
- Assignments 选项卡控制 (第 29 页)

2.2.4. 步骤 4：规划外围布局

在 **Flow** 控制上点击 **Plan Design**，在设计外围中的合法位置对 IP core 和其他设计单位进行交互式地布局。**Plan** 选项卡显示工程设计单元的列表以及目标器件体系结构的图形抽象。

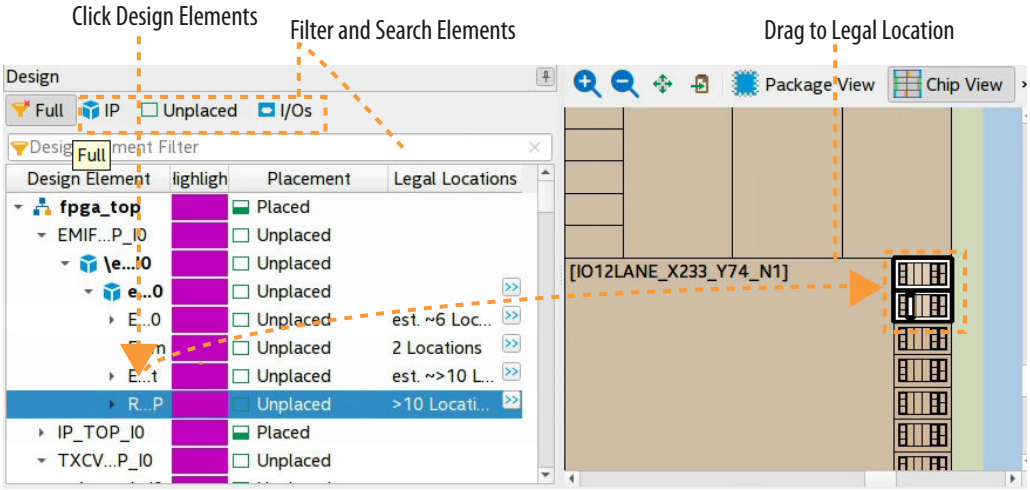
为提高效率，请按以下顺序在 Interface Planner 中布局设计单元：

1. 对具有已知特定位置要求的所有 I/O 管脚或者单元进行布局，例如 PLL。
2. 对于所有已知的外围接口 IP 进行布局。
3. (可选)对所有剩余的未布局的单元进行布局。

使用以下控制在 Interface Planner 平面图中布局设计单元：

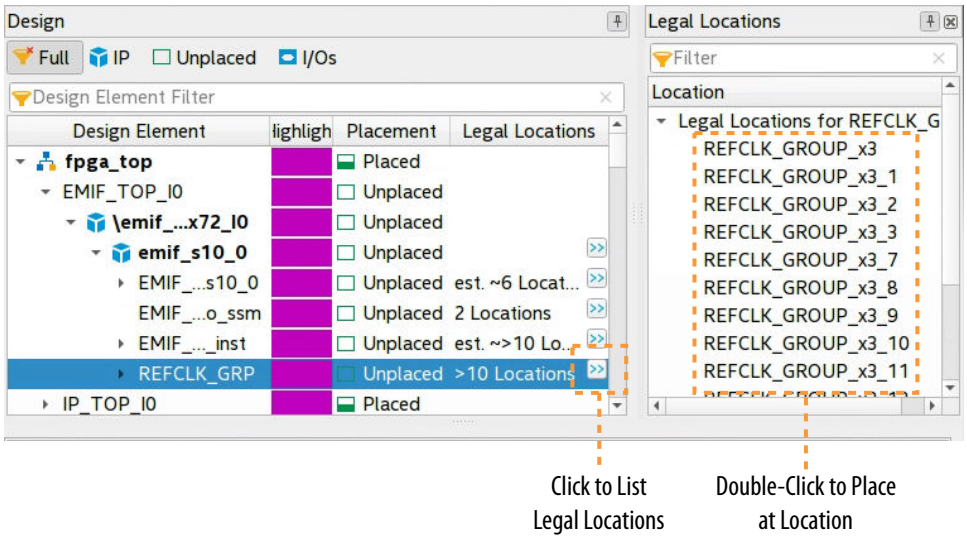
1. 在 **Design Element** 列表中定位想要布局的设计单元。您可以按名称，IP，布局状态，I/O 和其他条件来搜索和过滤列表。
2. 要自定义设计单元颜色编码定义，请单击 **Highlight** 列。

图 14. **Interface Planner (Plan Tab)**



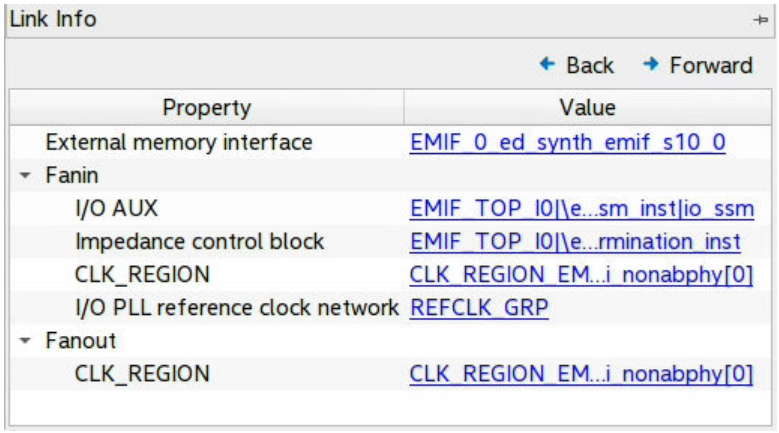
3. 使用以下任何方法在平面图中布局设计单元：
   - 将 **Design Elements** 列表中的单元拖放到 **Chip** 或者 **Package** 视图中的可用器件资源上。使用 Ctrl+Click 在 **Chip** 或者 **Package** 视图上进行拖动并平移。由于 Interface Planner 计算合法位置，因此拖动时可能会有一些延迟。
   - 要使 Interface Planner 能够在一个合法位置对一个未布局的设计单元进行布局，请右键点击并选择 **Autoplace Selected**。您必须对所有未布局的时钟使用 **Autoplace Selected**。
   - 点击 **Design Elements** 旁边的 >> 按钮，显示一列 **Legal Locations**(合法位置)。点击列表中的任何合法位置以在平面图中高显此位置。双击列表中的任何位置以在此位置布局单元。

图 15. 列出合法位置



4. 如果要回到规划变更的前一步或者下一步，请点击 **Undo** 或者 **Redo** 按钮。
5. 要可视化并遍历设计连接性(例如，查看参考时钟管脚和 PLL 的驱动目标单元)，请选择任何设计单元，然后单击 **Link Info** 选项卡。单击 **Back** 和 **Forward** 按钮以遍历设计连接。
6. 要生成一个显示 Fitter 所要求的布局位置的报告，请选择一个设计单元并点击 **Report Placeability of Selected Element**。

图 16. 遍历连接性的 **Link Info** 选项卡



注意: 直到将生成的接口规划约束应用到您的工程中，在 Interface Planner 中所作的变更才会应用到 Intel Quartus Prime 工程。

相关链接

Plan 选项卡控制 (第 29 页)

2.2.4.1. 布局时钟网络

Interface Planner 使您能够可视化和布局时钟网络。对于 Intel Arria® 10 和 Intel Cyclone® 10 GX 器件，您可以在 **Plan** 选项卡中定位，高亮显示和编辑时钟单元的类型。

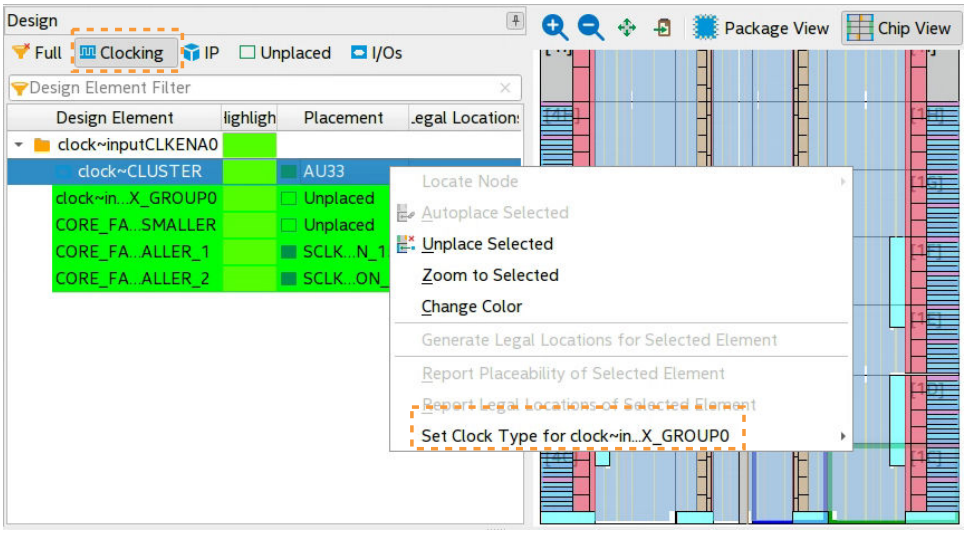
注意: Intel Stratix® 10 器件系列不支持 Interface Planner 中的 **Clocking** 过滤器。对于 Intel Stratix 10 设计，请使用 **Autoplace Selected** 命令对所有未布局的时钟单元进行布局。

Interface Planner 生成一个 Clocks 报告，使用器件中的低偏斜布线网络(时钟网络)对信号进行详细描述。

要在设计中标识和布局时钟单元，请点击 **Plan** 选项卡中的 **Clocking** 过滤器。通过在 **Design Element Filter** 中输入搜索文本可以进一步优化列表。Interface Planner 将时钟网络表示为以下时钟网络单元分组:

- Clock source
- Clock mux
- Clock region

图 17. 时钟设计单元



通过拖动到指定位置或者使用 **Report Legal Locations of Selected Element** 或 **Autoplace Selected** 命令对整个时钟组或者单个时钟单元进行布局。布局之后，将鼠标悬停在 **Design Element** 列表中的项目上以高亮显示布局。时钟单元的布局会影响相关内核和外围单元的布局。

您可以编辑时钟设计单元的时钟类型。时钟类型会影响相关内核和外围单元的布局。右键点击任何时钟单元来指定以下其中一种时钟类型：

- **Not Set**
- **Locally Routed**
- **Global**
- **Large Periphery**
- **Periphery**
- **Regional**

2.2.4.2. 保存和加载平面图

您可以保存 Interface Planner 平面图的状态以用于接下来的 Interface Planner 部分。Interface Planner 以 Interface Planner Floorplan Format (`.plan`)保存规划。您可以在 Interface Planner 中加载一个`.plan` 文件来重新打开平面图。

1. 要保存 Interface Planner 平面图，请点击 **File** > **Save Floorplan** 并指定一个文件名。
2. 要加载一个 Interface Planner 平面图，请点击 **File** > **Load Floorplan** 并找到`.plan` 文件。

注意: `.plan` 文件仅用于 Interface Planner 中，不直接用于 Intel Quartus Prime 软件中。如果尝试加载一个与当前 Interface Planner 工程不关联的`.plan` 文件，那么 Interface Planner 会生成一个错误消息。

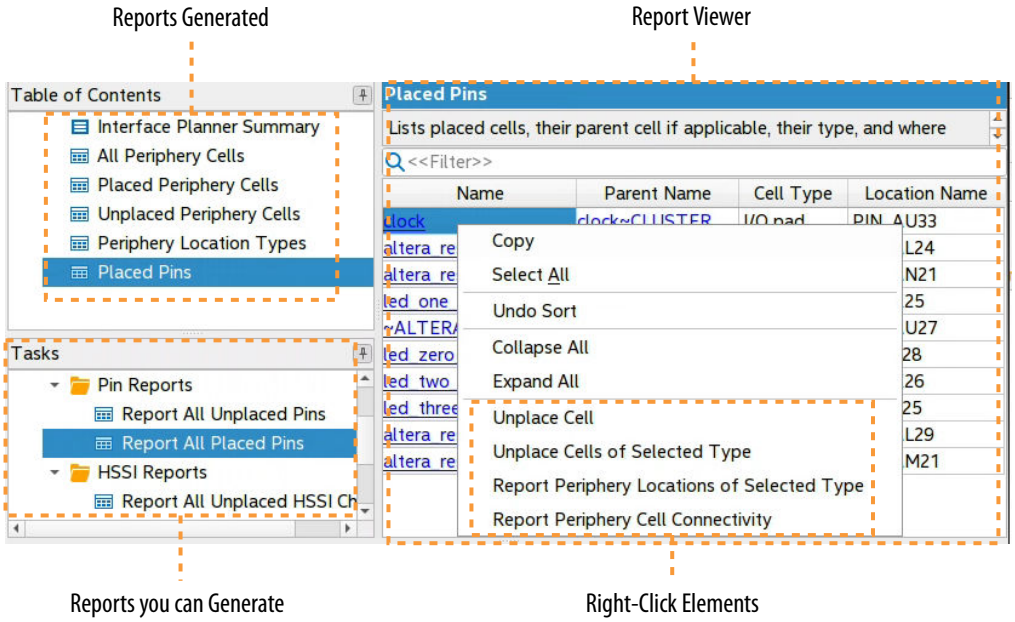
2.2.5. 步骤 5：报告布局数据

生成 Interface Planner 布局和连接性报告可帮助定位单元并就设计中接口和单元的布局做出最佳决策。点击 **Flow** 控制上的 **View Reports**，打开 **Reports** 选项卡，您可以从此选项卡生成一系列报告。

按照以下步骤报告 Interface Planner 布局数据：

1. 在 **Flow** 控制中，点击 **View Reports**。报告列表出现在 **Tasks** 窗格中。
2. 在 **Tasks** 窗格中，双击报告名称，在 **Table of Contents** 窗格中生成报告。
3. 在报告中选择设计单元，然后点击 **Place**，**Unplace**，或者报告有关所选单元或位置的详细数据。

图 18. Reports 选项卡



相关链接

- Reports 选项卡控制 (第 31 页)
- Interface Planner 报告 (第 32 页)

2.2.6. 步骤 6：验证和导出规划约束

您必须先验证接口规划，然后才能将规划约束作为生成的 Tcl 脚本导出到工程中。验证必须确认 Fitter 可以布局所有剩余的未布局的设计单元，然后才能生成脚本。当您对接口规划感到满意时，请按照以下步骤验证接口规划并将其应用于 Intel Quartus Prime 工程：

1. 在 **Flow** 控制中，点击 **Validate Plan**。Fitter 确认所有剩余的未布局的设计单元的布局。在导出约束前必须纠正所有错误。
2. 验证后，点击 **Export Constraints** 生成一个 Tcl 脚本，将规划应用到设计中。输出 Tcl 文件包含将接口规划应用到 Intel Quartus Prime 工程的指令。
3. 关闭 Interface Planner。
4. 要将导出的接口规划约束应用到 Intel Quartus Prime 工程中，请点击 **Tools** > **Tcl Scripts**，然后选择 `<project name>.pdp_assignments.tcl` 脚本文件。
5. 点击 **Run**。脚本运行，将 Interface Planner 约束应用到工程。或者，从工程目录运行脚本：

```
quartus_sh -t <assignments_file>.tcl
```

6. 要运行综合并应用接口规划，请点击 **Start** > **Start Analysis & Synthesis**。
7. 通过查看 Compilation Report 来确认规划的实现。

2.3. Interface Planner 用户接口控制

Interface Planner 用户接口包括用于规划设计平台的以下控制。

流程控制(Flow Controls) (第 28 页)
Home 选项卡控制 (第 29 页)
Assignments 选项卡控制 (第 29 页)
Plan 选项卡控制 (第 29 页)
Reports 选项卡控制 (第 31 页)

2.3.1. 流程控制(Flow Controls)

Flow 控制面板提供从 Interface Planner 中的任何地方对通用 Interface Planner 命令进行立即访问。 **Flow** 按典型接口规划流程显示。

表 3. 流程控制(Flow Controls)

命令	说明
Open Project	使您能够在 Interface Planner 中选择并打开一个 Intel Quartus Prime 工程。**Open Project** 命令仅在独立模式下使用时才需要。
Initialize Interface Planner	加载综合网表，启动 Fitter 验证引擎，从 Intel Quartus Prime 工程导入分配。
View Assignments	打开 **Assignments** 选项卡，查看和调解 Interface Planner 从工程中导入的冲突分配。使能或者禁用特定的工程分配来解决任何冲突。
Update Plan	将使能的工程分配应用到您的接口规划中。在更新规划之前，您不能在 **Plan** 选项卡上执行外围规划。
Plan Design	打开 **Plan** 选项卡，在接口规划中布局逻辑。
Export Constraints	将接口规划保存为 Tcl 脚本文件，用于工程中的其他应用。此命令只有在成功运行 **Validate Plan** 之后才可用。
Validate plan	验证接口规划中的所有约束是否与剩余的全部未布局的设计单元兼容。然后，您可以直接找到并解决所有报告的验证错误的来源。在运行 **Write Plan File** 之前必须成功地验证规划。
View Reports	打开 **Reports** 选项卡，过滤数据并查找实体和位置。

2.3.2. Home 选项卡控制

Interface Planner **Home** 选项卡包含在 Interface Planner 中打开工程的控制。当 Interface Planner 处于独立模式时仅需要 **Home** 选项卡。

表 4. Home 选项卡控制

命令	说明
Recent Projects	提供对最近打开的 Intel Quartus Prime 工程的快速访问。一个命名的图块(tile)代表每个工程。单击图块以显示关于工程的详细信息(**Details**)。双击图块以在 Interface Planner 中打开工程。
Browse	使您能够在 Interface Planner 中定位并打开一个 Intel Quartus Prime 工程。Interface Planner 需要工程的综合网表才能运行。
Details	提供工程和文件的详细信息，例如文件路径，修订版和 Intel Quartus Prime 工程的创建日期。您可以选择一个特定的工程修订版本。

2.3.3. Assignments 选项卡控制

Assignments 选项卡包含用于解决与工程分配潜在冲突的控制。点击 **View Assignments** 以显示 **Assignments** 选项卡。

您可以使能或禁用特定分配或类分配，直到解决所有潜在的冲突。对所有工程分配的状态感到满意之后，请单击 **Update Plan** 以使用使能的工程分配来更新接口规划。Interface Planner 报告对于任何剩余的分配冲突报告一个错误。

表 5. Assignments 选项卡控制

命令	说明
Filter field	支持对分配目标创建通配符表达式。**Enabled** 和 **Disabled** 按钮仅过滤列表中的使能或禁用分配。
Enable All Project Assignments	使能接口规划中的所有导入的工程分配。
Disable All Project Assignments	禁用规划中的所有导入的工程分配。
Clear	从 **Assignments** 列表中清除所有过滤器。

2.3.4. Plan 选项卡控制

Plan 选项卡包含以下控制，可帮助您在接口布局中定位和布局逻辑。点击 **Plan Design** 以显示 **Plan** 选项卡。

直到将所生成的 Interface Planner 约束脚本添加到工程中，接口规划中的布局或者未布局才会应用到 Intel Quartus Prime 工程中。

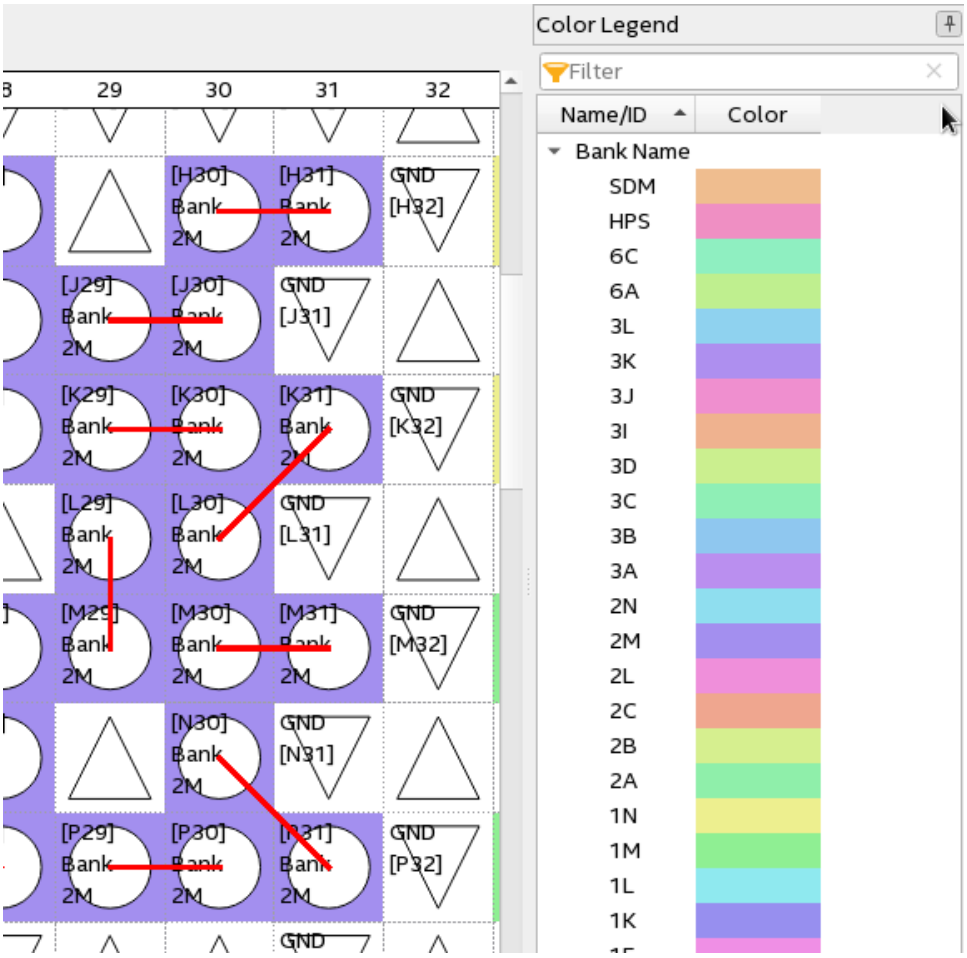
表 6. Plan 选项卡控制

命令	说明
>>	列出布局的合法位置。
Locate Node	显示一列 Intel Quartus Prime Pro Edition 工具，其中所选的设计单元在层次数据库中被引用。如果 **Locate Node** 命令被禁止用于 **Design Elements** 列表中的特定单元，那么因为此单元没有在设计中表示为一个单元。

继续...

命令	说明
Autoplace All	尝试在接口规划中的合法位置布局所有未布局的设计单元。
Autoplace Fixed	尝试将只有一个合法位置的所有未布局的设计单元布局到接口规划中。
Unplace All	对接口规划中的的所有已布局的设计单元进行布局解除。
Right-click > **Auto-place selected element**	尝试在接口规划中的合法位置布局已选的设计单元以及其所有子设计单元。
Chip View	显示目标器件芯片。放大以显示芯片详细信息。
Package View	显示目标器件封装。放大以显示芯片详细信息。
Show I/O Banks	在 **Plan** 选项卡中选择并颜色编码 I/O bank。
Show Differential Pin Pair Connections	在一对差分管脚之间显示一条红色连接线。Package View 使用字母 p 和 n 分别标记正管脚和负管脚。
Show PCIe Hard IP Interface Pins	在 **Plan** 选项卡中选择并颜色编码 PCIe Hard IP 接口。要访问此命令，右键点击 **Plan** 选项卡封装视图，然后选择 **x1 Lanes**，**x2 Lanes**，**x4 Lanes**，**x8 Lanes** 或者 **by 16 Lanes**。使能后，在 **Color Legend** 中查看颜色编码。
Show DQ/DQS Pins	在 **Plan** 选项卡中选择并颜色编码 PCIe Hard IP 接口。要访问此命令，右键点击 **Plan** 选项卡封装视图，然后选择 **x4 Mode**，**x8/x9 Mode**，**x16/x16 Mode** 或者 **x32/x36 Mode**。使能后，在 **Color Legend** 中查看颜色编码。
Right-click > **Report Placeability of Selected Element**	显示 **Reports** 选项卡上的详细信息，以相应的顺序显示所选单元在接口规划中的合法位置。
Copy Current View	将当前接口规划复制到剪贴板，以粘贴到其他文件中，例如文字处理或演示文稿文件。
Reset Plan	对所有已布局的设计单元解除布局，并从接口规划中删除应用的工程分配。将所有工程分配复位成使能的状态。您必须先布局设计单元，然后在运行 **Update Plan**。此命令仅应用于接口规划，在应用 Interface Planner 脚本之前不会影响 Intel Quartus Prime 工程分配。
Load Floorplan	使您能够选择并加载 Interface Planner Floorplan Format (`.plan`)文件。通过点击 **Save Floorplan** 以此格式保存 Interface Planner 平面图文件。
Save Floorplan	可将 Interface Planner 平面图保存为`.plan` 文件。
Undo/Redo buttons	**Undo** 按钮将还原在 **Plan** 选项卡中所作的最后更改。**Redo** 重新实现最后一次撤销。当规划变更时，使用这些命令进行前进和后退操作。

图 19. 差分管脚对颜色编码



2.3.5. Reports 选项卡控制

Interface Planner **Reports** 选项卡包含以下 **Task** 窗格控制来帮助您过滤数据和查找实体和位置。

表 7. **Reports** 选项卡控制

命令	说明
Create all Summary Reports	创建以下汇总报告： • **Interface Planner Summary** • **All Periphery Cells** • **Placed/Unplaced Periphery Cells** • **Periphery Location Types**
Report All Placed/Unplaced Pins	报告接口规划中的所有布局的(**Report All Placed Pins**)或者未布局的(**Report All Unplaced Pins**)管脚的名称，父级(如果有)和类型。Placed Pins 报告包括布局位置名称。Unplaced Pins 报告包括潜在布局位置的数量。右键点击任何单元来布局，解除布局，或者报告连接性或者位置信息。

继续...

命令	说明
Report All Placed/Unplaced HSSI Channels	报告接口规划中的所有布局的(**Report All Placed HSSI Channels**)或者未布局的(**Report All Unplaced HSSI Channels**)通道的名称，父级(如果有)和类型。Placed HSSI Channels 报告包括布局位置名称。Unplaced HSSI Channels 报告包括潜在布局位置的数量。右键点击任何单元来布局，解除布局，或者报告连接性或者位置信息。
Right-click > Report Placed/ Unplaced Periphery Cells of Selected Type	报告与所选类型匹配的布局的(**Report Placed Periphery Cells of Selected Type**)或者未布局的(**Report Unplaced Periphery Cells of Selected Type**)单元的名称，父级(如果有)和类型。布局的单元报告包括布局位置名称。未布局的单元报告包括潜在布局位置的数量。右键点击任何单元来布局，解除布局，或者报告连接性或者位置信息。
Right-click > Report Periphery Locations of Selected Type	报告所选类型器件中的所有位置，以及此位置是否支持合并。
Right-click > Report Periphery Cell Connectivity	报告源端口和类型，目的端口和与所选单元的连接类型。右键点击任何单元以报告单个单元连接性。
Right-click > Place/Unplace Cell	将单元布局在接口规划中的所选位置。同样，您也可以右键点击任何单元，然后点击 **Place Cell of Selected Type** 或者 **Unplace Cell of Selected Type** 对相同类型的多个单元进行布局或者解除布局。
Right-click > Report Cell Locations for Custom Placement	在 Legal Location 报告中报告接口规划中所选单元的首选合法位置。右键点击后立即在一个位置布局单元，或者报告所选类型的所有外围位置。
Remove Invalid Reports	删除过时的 Interface Planner 报告，因更改接口规划而使这些报告无效。
Report Instance Assignments	显示接口规划中所有导入的工程分配。您可以从规划中删除这些分配。

2.4. Interface Planner 报告

使用 Interface Planner 报告来定位单元并对设计中特定的接口和单元分配合适的布局位置。Interface Planner 报告提供详细可行的反馈，以帮助您快速实现最佳设计规划。您可以直接从 Interface Planner 报告访问布局和其他报告功能。Interface Planner 生成以下报告，这些报告提供详细的规划信息：

报告汇总 (第 32 页)
报告管脚 (第 33 页)
报告 HSSI 通道 (第 35 页)
报告时钟 (第 35 页)
报告外围位置 (第 36 页)
报告单元连接性 (第 37 页)
Report Instance Assignments(报告实例分配) (第 37 页)

2.4.1. 报告汇总

点击 **Reports** 选项卡上的 **Create all Summary Reports** 来生成关于接口规划中的外围单元的汇总报告。右键点击任何单元类型以报告布局信息，未布局信息，连接信息或者位置信息。

图 20. 汇总报告

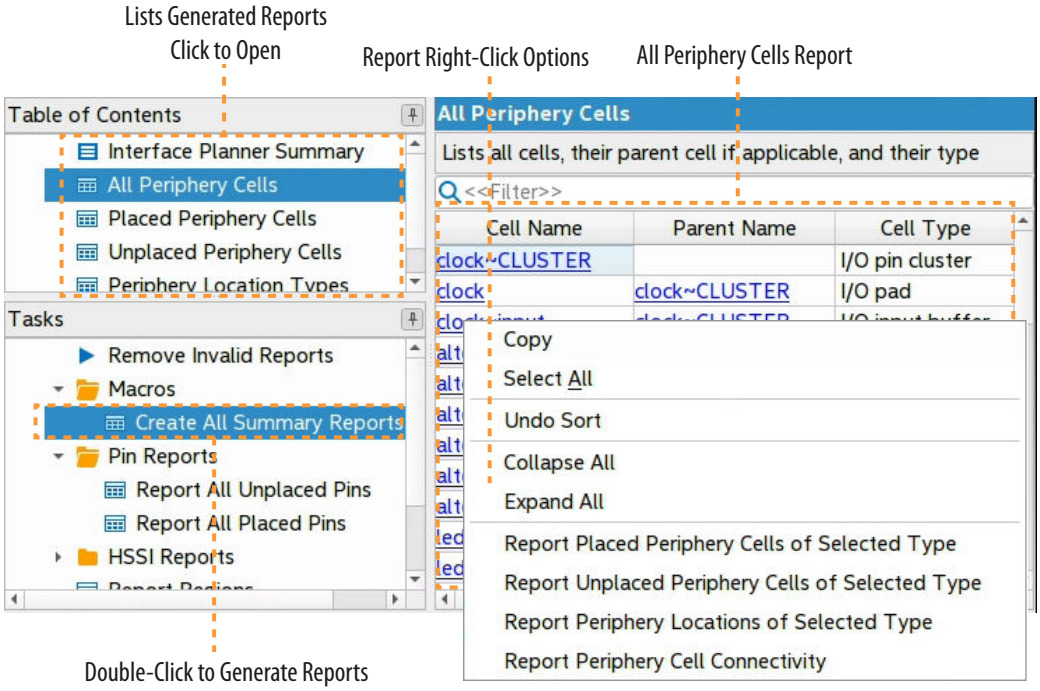


表 8. 报告汇总

命令	说明
Create all Summary Reports	创建以下汇总报告： • **Interface Planner Summary**—报告软件版本和外围和顶层外围单元的总数。 • **All Periphery Cells**—报告设计中所有外围单元的名称，父级和类型。 • **Placed/Unplaced Periphery Cells**—报告接口规划中所有布局的和未布局的外围单元的名称，父级和类型。 • **Periphery Location Types**—报告目标器件中可用的每种类型的外围位置的数量以及设计所要求的数量。

2.4.2. 报告管脚

生成关于设计中 I/O 管脚的报告。右键点击任何单元类型以报告布局信息，未布局信息，连接信息或者位置信息。

表 9. 报告管脚命令

命令	说明
Report All Placed Pins	生成 Placed Pins 报告。此报告列出了接口规划中所有布局的管脚的名称，父级和位置。
Report All Unplaced Pins	生成 Unplaced Pins 报告。此报告列出了接口规划中所有未布局的管脚的潜在布局的名称，父级，类型和数量。

图 21. Placed Pins 报告

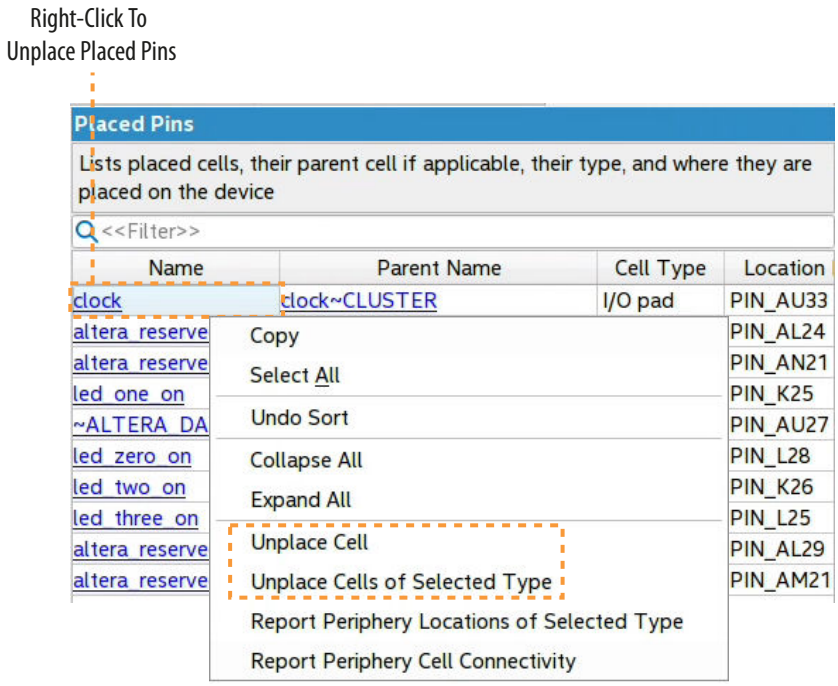
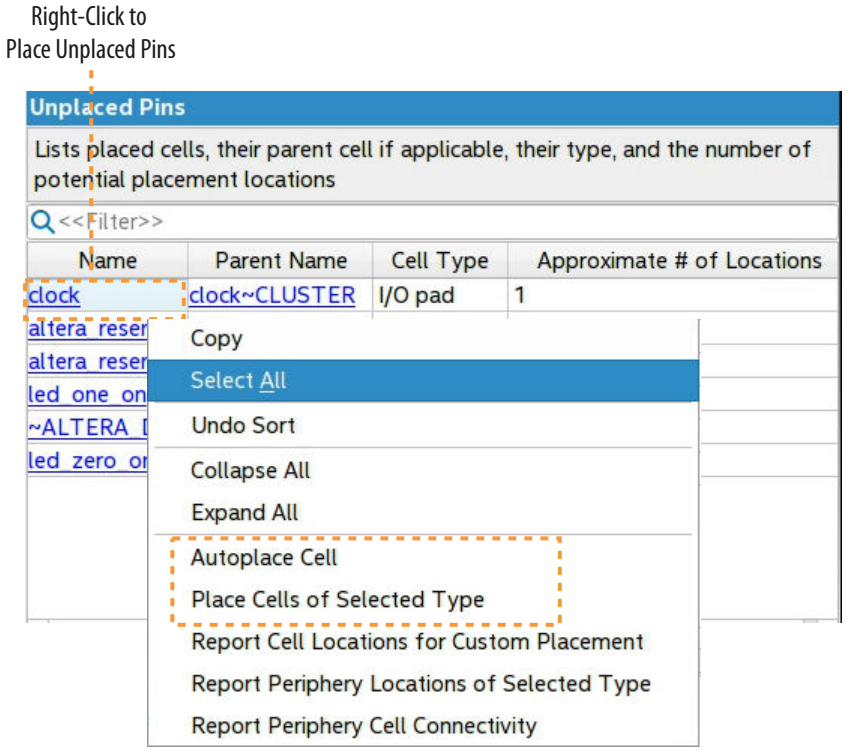


图 22. Unplaced Pins 报告



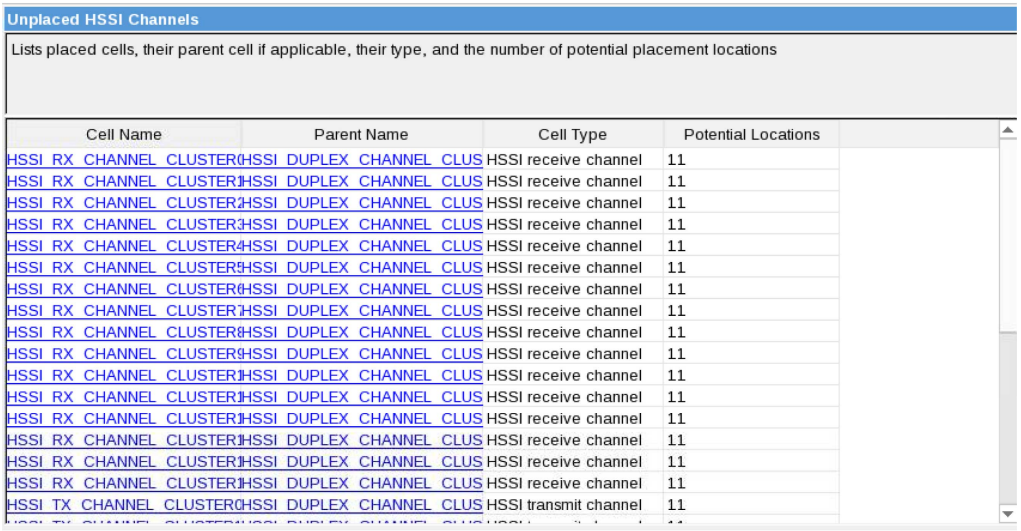
2.4.3. 报告 HSSI 通道

生成关于接口规划中 HSSI 通道的报告。右键点击任何单元类型以报告布局信息，未布局信息，连接信息或者位置信息。

表 10. 报告通道命令

命令	说明
Report All Placed HSSI Channels	生成 Placed HSSI Channels 报告。此报告列出了接口规划中所有布局的 HSSI RX/TX 通道的名称，父级和位置。
Report All Unplaced HSSI Channels	生成 Unplaced HSSI Channel 报告。此报告列出了接口规划中所有未布局的 HSSI RX/TX 通道的名称，父级和位置。

图 23. Unplaced HSSI Channels 报告



Unplaced HSSI Channels

Lists placed cells, their parent cell if applicable, their type, and the number of potential placement locations

Cell Name	Parent Name	Cell Type	Potential Locations
HSSI_RX_CHANNEL_CLUSTER0	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER2	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER3	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER4	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER5	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER6	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER7	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER8	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER9	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_RX_CHANNEL_CLUSTER1	HSSI_DUPLEX_CHANNEL_CLUS	HSSI receive channel	11
HSSI_TX_CHANNEL_CLUSTER0	HSSI_DUPLEX_CHANNEL_CLUS	HSSI transmit channel	11

2.4.4. 报告时钟

生成显示规划中时钟网络的报告。使用此报告分析时钟网络方案并确保特定器件区域由高扇出信号驱动。

表 11. 报告时钟命令

命令	说明
Report Clocks	生成 Global 和其他 Fast Signals 报告。

图 24. 时钟报告

Clocks

Shows the signals that are using low-skew routing networks (clock networks) in the device.
If applicable, also shows any signals that were considered for automatic clock network promotion, but were not promoted.

<<Filter>>

	Source	Source Location	Fan-Out	Signal Type	Promotion Type	Global Buffer	Clock Region
1	clock	PIN_AU33	33	Global...omoted	Automatic	clock~inputCLKEN	Spine Clock Region 7 to

2.4.5. 报告外围位置

生成显示接口布局中外围单元的状态的报告。

表 12. 报告外围位置命令

命令	说明
Right-click > Report Placed Periphery Cells of Selected Type	可从 **All Periphery Cells** 报告进行访问。此命令报告与已选类型匹配的已选布局的外围单元的名称，父级(如果有)，类型和位置。右键点击任何单元来布局，解串布局或者报告连接性或者位置信息。
Right-click > Report Unplaced Periphery Cells of Selected Type	可从 **All Periphery Cells** 报告进行访问。此命令报告与已选类型匹配的已选的未布局的外围单元的名称，父级(如果有)，类型和适合位置的数量。右键点击任何单元来布局，解串布局或者报告连接性或者位置信息。
Right-click > Report Periphery Locations of Selected Type	报告所选类型器件中的所有位置，以及此位置是否支持合并。

图 25. **Placed Periphery Cells** 报告

Placed Periphery Cells

Lists placed cells, their parent cell if applicable, their type, and where they are placed on the

<<Filter>>

Name	Parent Name	Cell Type	Location Name
clock~CLUSTER		I/O pin cluster	AU33
clock	clock~CLUSTER	I/O pad	PIN_AU33
clock~input	clock~CLUSTER	I/O input buffer	IOIBUF_X78_Y125_N32
altera_reserved_tms~CLUS		I/O pin cluster	AL24
altera_reserved_tms	altera_reserved_tms~	I/O pad	PIN_AL24
altera_reserved_tms~input	altera_reserved_tms~	I/O input buffer	IOIBUF_X115_Y0_N65
altera_reserved_tdi~CLUST		I/O pin cluster	AN21
altera_reserved_tdi	altera_reserved_tdi~(	I/O pad	PIN_AN21
altera_reserved_tdi~input	altera_reserved_tdi~(	I/O input buffer	IOIBUF_X115_Y0_N47
led_one_on~CLUSTER		I/O pin cluster	K25
led_one_on	led_one_on~CLUSTE	I/O pad	PIN_K25
led_one_on~output	led_one_on~CLUSTE	I/O output buffer	IOOBUF_X142_Y205_N18

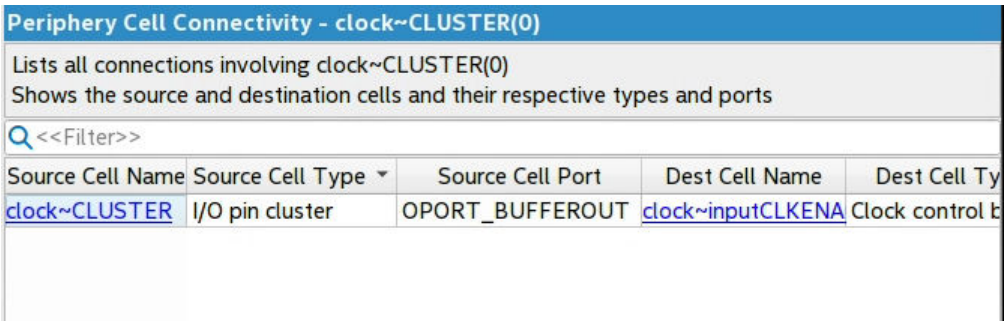
2.4.6. 报告单元连接性

生成显示接口规划中所有单元之间的连接的报告。

表 13. 报告单元连接性命令

命令	说明
Right-click > Report Periphery Cell Connectivity	右键点击报告中的任何 **Cell Name** 来报告外围单元连接性 **Report Periphery Cell Connectivity**。此报告列出源和目的端口和与所选单元的连接类型。右键点击任何单元以报告与此单元的所有连接。

图 26. Periphery Cell Connectivity 报告



2.4.7. Report Instance Assignments(报告实例分配)

点击 **Report Instance Assignments** 以显示接口规划中所有导入的工程分配。您可以从规划中删除这些分配。

表 14. 报告实例分配命令

命令	说明
Report Instance Assignments	报告设计中所有使能的实例分配。右键点击任何单元来删除分配或者删除同一类型的所有分配。

图 27. **Instance Assignments** 报告

Instance Assignments

<<Filter>>

ID	Status	From	To	Assignment Name	Va
0	Enabled		u_blinking_led	PLACE_REGION	X57 Y6 X61 Y
1	Enabled		u_top_counter	PLACE_REGION	X64 Y6 X68 Y
2	Enabled		u_blinking_led	ROUTE_REGION	X56 Y5 X62 Y
3	Enabled		u_top_counter	ROUTE_REGION	X63 Y5 X69 Y
4	Enabled		u_blinking_led	RESERVE_PLACE_REGION	ON
5	Enabled		u_top_counter	RESERVE_PLACE_REGION	ON
6	Enabled		u_blinking_led	CORE_ONLY_PLACE_REGION	ON
7	Enabled		u_top_counter	CORE_ONLY_PLACE_REGION	ON
8	Enabled		u_blinking_led	REGION_NAME	u_blinking_lec
9	Enabled		u_top_counter	REGION_NAME	u_top_counte
10	Enabled		clock	LOCATION	PIN_AU33
11	Enabled		led_one_on	LOCATION	PIN_K25
12	Enabled		led_three_on	LOCATION	PIN_L25

2.5. 接口规划修订历史

本文档的修订历史如下:

文档版本	Intel Quartus Prime 版本	修订内容
2019.04.01	19.1.0	• 更新了"Plan Tab Controls"，描述了 I/O bank，差分管脚对，DQ/DQS 管脚和 PCIe hard IP 管脚的新颜色编码控制。 • 更新了最新用户接口的截屏图和程序步骤。
2018.05.07	18.0.0	• Design Constraints User Guide: Intel Quartus Prime Pro Edition 中的初始发布。 • 更新了 *Step 2: Initialize Interface Planner*，移除了关闭 Intel Quartus Prime 的要求。 • 更新了 *Step 4: Plan Periphery Placement*，描述了何时禁用 **Locate Node** 命令。
2017.11.06	17.1.0	• 删除了对 Intel Stratix 10 的 **Clocking** 支持。Intel Stratix 10 时钟必须使用 **Autoplace Selected**。 • 将 BluePrint 重命名为 Interface Planner。 • 将 *BluePrint Design Planning* 章节重命名为 *Interface Planning*。
2016.10.31	16.1.0	• 进行了 Intel 重命名。
2016.05.03	16.0.0	• 添加了 Plan Clock Networks 主题。 • 添加了 Saving and Loading Floorplans 主题。 • 添加了 Undo/Redo 命令描述。 • 添加了流程控制描述。 • 添加了关于平移功能(panning feature)的注释。 • 更新了最新 GUI 的所有屏幕截图。

继续...

文档版本	Intel Quartus Prime 版本	修订内容
2015.11.02	15.1.0	• 将 *Quartus II* 修改成 *Quartus Prime*。 • 将内容整合到 Quartus Prime Handbook 中。 • 添加了新动态报告的描述。 • 添加了 Package View 描述。 • 添加了 GUI 控制参考。
2015.05.04	15.0.0	Molson 上文档的第二个 beta 版本。添加了关于以下主题的信息： • Overview information • Reset Plan command • Legal Assignments list and prompt • Tcl console
2014.12.15	14.1.	Molson 上的第一个 beta 版本的文档。

相关链接

Altera 文档存档

关于 *Intel Quartus Prime Handbook* 的早前版本，请搜索 Altera 文档存档。

3. 管理器件 I/O 管脚

本章介绍了目标器件中 I/O 管脚的有效规划和分配。在设计阶段的初期，请考虑 I/O 标准，管脚布局规则和 PCB 特性。

图 28. Pin Planner GUI

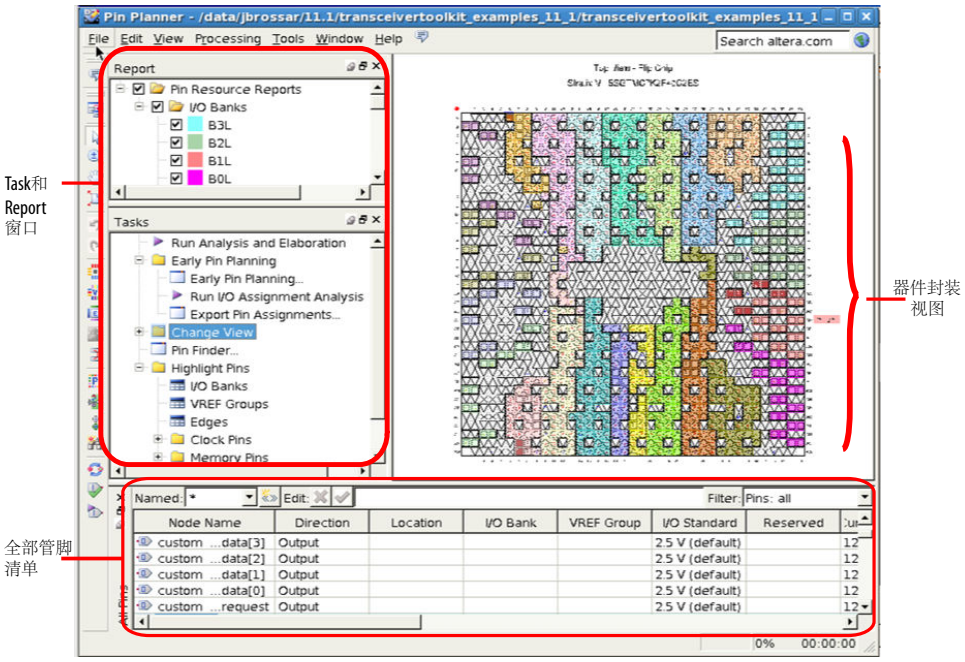


表 15. Intel Quartus Prime I/O 管脚规划工具

I/O 规划任务	点击访问
规划接口和器件外设	**Tools** > **Interface Planner**
编辑，验证或者导出管脚分配	**Assignments** > **Pin Planner**

关于 Intel Arria 10 SoC 器件的特殊管脚分配特性的详细信息，请参考 *Intel Arria 10 Hard Processor System Technical Reference Manual* 中的 *Instantiating the HPS Component*。

相关链接

- 接口规划概述 (第 18 页)
- Instantiating the HPS Component
  In *Intel Arria 10 Hard Processor System Technical Reference Manual*

英特尔公司。保留所有权利。英特尔、英特尔徽标和其他英特尔标志是英特尔公司或其子公司的商标。依照英特尔的标准保证条例，英特尔保证其 FPGA 和半导体产品的性能符合当前规格，但保留随时更改任何产品和服务的权利，恕不另行通知。英特尔概不承担因应用或使用本文中描述的任何信息、产品或服务而产生的任何责任和义务，除非得到英特尔书面上的明确同意。建议英特尔客户在信赖任何已发布的信息之前以及下单订购产品或服务之前，应先获取最新版本的器件规格。
*其他的名称和品牌可能是其他所有者的资产。

ISO 9001:2015 Registered

3.1. I/O 规划概述

在 FPGA 设计上，I/O 规划包括创建与管脚相关的分配，并根据管脚布局指南进行验证。此过程可确保成功匹配目标器件。当在工程的初始阶段规划和分配 I/O 管脚时，您要设计与目标器件和 PCB 特征的兼容性。因此，您的设计过程将减少迭代次数，并且可以更快地开发出准确的 PCB 布局。

您甚至可以在定义设计文件之前就规划 I/O 管脚。分配设计文件(包括接口 IP 内核信号)中尚未定义的预期节点，然后生成一个顶层文件。顶层文件例化设计层次结构的下一层，并包括接口端口信息，例如存储器，高速 I/O，器件配置和调试工具。

通过名称或拖动到单元来对器件 I/O 管脚分配设计单元，I/O 标准，接口 IP 和其他属性。然后，您可以生成用于 I/O 验证的顶层设计文件。

使用 I/O 分配验证可根据 VCCIO，VREF，电迁移(电流密度)，同时开关输出(SSO)，驱动强度，I/O 标准，PCI_IO 钳位二极管和 I/O 管脚方向兼容性规则对 I/O 管脚进行全面分析。

Intel Quartus Prime 软件提供 Pin Planner 工具，查看，分配和验证器件 I/O 管脚逻辑和属性。或者，您可以在 Tcl 脚本中或者直接在 HDL 代码中输入 I/O assignment。

3.1.1. 基本 I/O 规划流程

以下步骤描述了分配和验证 I/O 管脚的基本流程：

1. 点击 **Assignments** > **Device**，然后选择一个满足您的逻辑，性能和 I/O 要求的目标器件。考虑并指定 I/O 标准，电压和电源要求以及可用的 I/O 管脚。
2. 点击 **Assignments** > **Pin Planner**。
3. 分配 I/O 属性以匹配器件和 PCB 特征，包括分配逻辑，I/O 标准，输出负载，摆率和电流强度。
4. 点击 **Tasks** 窗格中的 **Run I/O Assignment Analysis** 以验证分配并生成综合的设计网表。更正报告的所有问题。
5. 点击 **Processing** > **Start Compilation**。编译期间， Intel Quartus Prime 软件运行 I/O assignment 分析。

3.1.2. 集成 PCB 设计工具

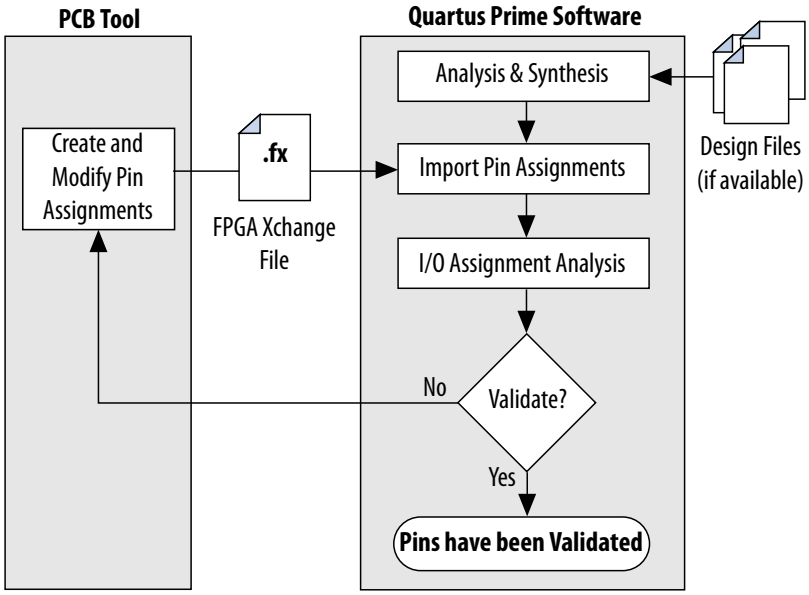
您可以将 PCB 设计工具集成到工作流程中，以将管脚分配映射到系统电路原理图和电路板布局中的符号。

Intel Quartus Prime 软件通过支持 Intel Quartus Prime Settings Files (`.qsf`)，Pin-Out File (`.pin`)和 FPGA Xchange-Format File (`.fx`)文件中的管脚分配信息的导入和导入将电路板布局工具集成在一起。

表 16. 集成 PCB 设计工具

PCB 工具集成	支持的 PCB 工具
在 Pin Planner 中定义和验证 I/O 分配，然后将分配导出到 PCB 工具进行验证	Mentor Graphics* I/O Designer Cadence Allegro
在 PCB 工具中定义 I/O 分配，然后将分配导入 Pin Planner 进行验证	Mentor Graphics I/O Designer Cadence Allegro

图 29. PCB 工具集成

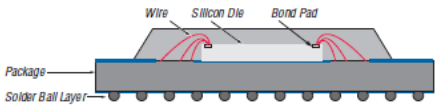
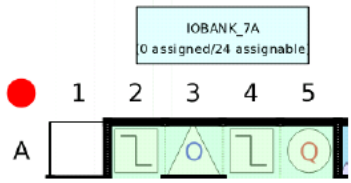
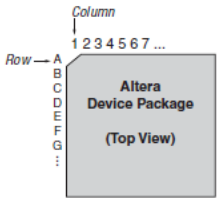
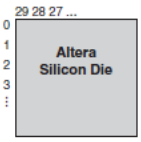
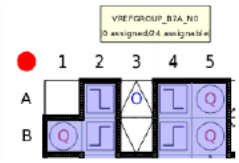


相关链接

Cadence PCB Design Tools Support
In *Intel Quartus Prime Pro Edition User Guide: PCB Design Tools*

3.1.3. Intel 器件术语

以下术语描述了 Intel 器件和 I/O 结构:

Terms	Description	Diagram
Device Package (BGA example)	Ceramic or plastic heat sink surface mounted with FPGA die and I/O pins or solder balls. In a wire bond BGA example, copper wires connect the bond pads to the solder balls of the package. Click **View** > **Show** > **Package Top** or **View** > **Show** > **Package Bottom** in Pin Planner	
I/O Bank	I/O pins are grouped in I/O banks for assignment of I/O standards. Each numbered bank has its own voltage source pins, called VCCIO pins, for high I/O performance. The specified VCCIO pin voltage is between 1.5 V and 3.3 V. Each bank supports multiple pins with different I/O standards. All pins in a bank must use the same VCCIO signal. Click **View** > **Show** > **I/O Banks** in Pin Planner.	
I/O Pin	A wire lead or small solder ball on the package bottom or periphery. Each pin has an alphanumeric row and column number. I, O, Q, S, X, and Z are never used. The alphabet is repeated and prefixed with the letter A when exceeded. All I/O pins display by default.	
Pad	I/O pins are connected to pads located on the perimeter of the top metal layer of the silicon die. Each pad is numbered with an ID starting at 0, and increments by one in a counterclockwise direction around the device. Click **View** > **Pad View** in Pin Planner.	
VREF Pin Group	A group of pins including one dedicated VREF pin required by voltage-referenced I/O standards. A VREF group contains a smaller number of pins than an I/O bank. This maintains the signal integrity of the VREF pin. One or more VREF groups exist in an I/O bank. The pins in a VREF group share the same VCCIO and VREF voltages. Click **View** > **Show** > **Show VREF Groups** in Pin Planner..	

3.2. 分配 I/O 管脚

使用 Pin Planner，以目标器件的图像表示形式对 I/O assignment 进行可视化，修改和验证。通过保留特定器件管脚以适应未定义的但期望的 I/O，可以提高 I/O assignment 分析的准确性。

请按照以下步骤在 Pin Planner 中分配 I/O 管脚：

1. 打开一个 Intel Quartus Prime 工程，然后点击 **Assignments** > **Pin Planner**。
2. 点击 **Processing** > **Start Analysis & Elaboration** 对设计进行详述，并在器件视图中显示 **All Pins**。
3. 要查找或者高显要分配的管脚，请点击 **Pin Finder**，或者在 **Tasks** 窗格中的 **Highlight Pins** 下选择一个管脚类型。

4. (可选)要定义一组定制的节点进行分配，请在 **Groups** 或者 **All Pins** 列表中选择一个或者多个节点，然后点击 **Create Group**。
5. 在 **All Pins** 表格中输入器件 I/O 管脚的逻辑分配，I/O 标准，接口 IP 和属性，或者拖动到封装视图中。
6. 要对差分管脚对分配属性，请点击 **Show Differential Pin Pair Connections**。在正(p)和负(n)差分管脚之间出现一条红色连线。
7. (可选)创建板级走线模型分配:
   a. 右键单击一个输出或双向管脚，然后单击 **Board Trace Model**。对于差分 I/O 标准，板级走线模型使用一个差分管脚对以及两个对称板级走线模型。
   b. 指定差分管脚对的正端上的板级走线参数。此分配适用于差分管脚对负端上的对应值。
8. 要运行完整的 I/O assignment 分析，请点击 **Run I/O Assignment Analysis**。Fitter 报告分析结果。在设计综合之前，仅分析保留的管脚。

3.2.1. 分配给独占管脚组(Assigning to Exclusive Pin Groups)

您可以对独占分配指定管脚组。当将管脚分配给 **Exclusive I/O Group** 时，Fitter 不会将信号与任何其他独占 I/O 组布局在同一 I/O 组中。例如，如果您有一组专门分配给 `group_a` 的信号，而另一组分配给 `group_b` 的信号，那么 Fitter 确保将每个组布局在不同的 I/O bank 中。

3.2.2. 分配摆率和驱动强度

您可以指定器件管脚摆率和驱动强度。这些属性会影响管脚的输出信号完整性。使用 **Slew Rate** 或者 **Slow Slew Rate** assignment 可以通过 **Current Strength** assignment 来调整一个管脚的驱动强度。

注意: 摆率和驱动强度应用在 I/O assignment 分析期间。

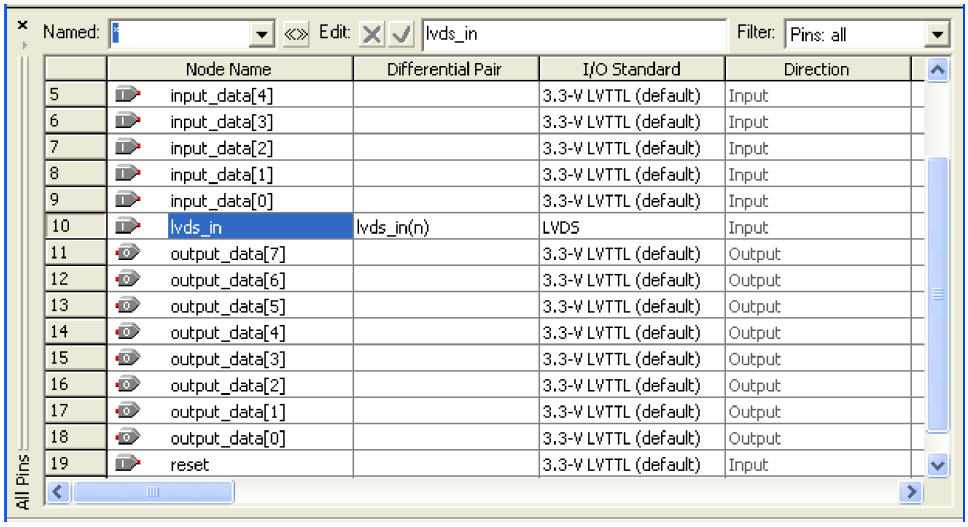
3.2.3. 分配差分管脚

在设计中为一个单端顶层管脚分配一个差分 I/O 标准时，Pin Planner 会自动将负管脚识别为差分管脚对分配的一部分，并为您创建负管脚。 Intel Quartus Prime 软件将负管脚的位置分配写入到 `.qsf` 中；但是，对于差分对的负管脚，没有将 I/O 标准分配添加到 `.qsf` 中。

以下示例显示了一个包括 `lvds_in` 顶层管脚的设计，此管脚分配了一个差分 I/O 标准。Pin Planner 自动创建差分管脚 `lvds_in(n)` 以完成差分管脚对。

注意: 如果您有一个驱动 PLL 的单端时钟，那么仅将此管脚分配给目标器件中差分对的正时钟管脚。驱动 PLL 并分配给负时钟管脚器件的单端管脚会导致设计不适用。

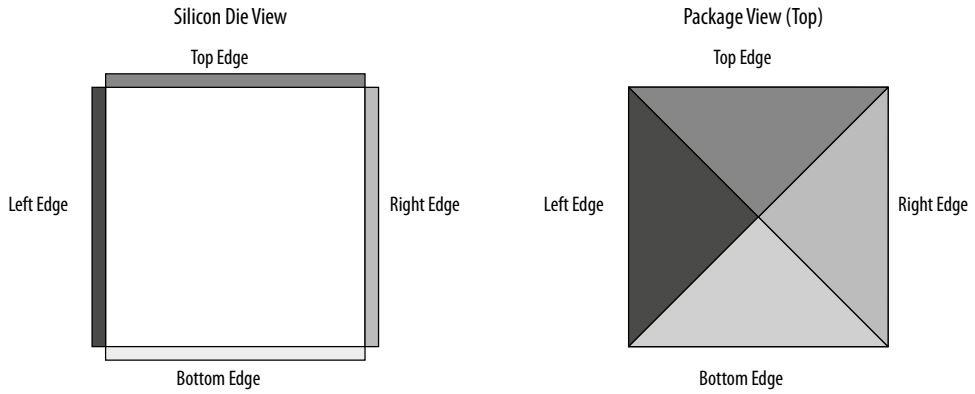
图 **30.** 在 **Pin Planner** 中创建差分管脚对



	Node Name	Differential Pair	I/O Standard	Direction
5	input_data[4]		3.3-V LVTTL (default)	Input
6	input_data[3]		3.3-V LVTTL (default)	Input
7	input_data[2]		3.3-V LVTTL (default)	Input
8	input_data[1]		3.3-V LVTTL (default)	Input
9	input_data[0]		3.3-V LVTTL (default)	Input
10	lvds_in	lvds_in(n)	LVDS	Input
11	output_data[7]		3.3-V LVTTL (default)	Output
12	output_data[6]		3.3-V LVTTL (default)	Output
13	output_data[5]		3.3-V LVTTL (default)	Output
14	output_data[4]		3.3-V LVTTL (default)	Output
15	output_data[3]		3.3-V LVTTL (default)	Output
16	output_data[2]		3.3-V LVTTL (default)	Output
17	output_data[1]		3.3-V LVTTL (default)	Output
18	output_data[0]		3.3-V LVTTL (default)	Output
19	reset		3.3-V LVTTL (default)	Input

如果设计包含的大型总线超出了特定 I/O bank 中可用的管脚，那么可以使用边缘位置分配来布局总线。边缘位置分配提高了大型总线的电路板布线能力，因为它们在边缘附近彼此靠近。下图显示了 Intel 器件封装边缘。

图 **31.** **Intel** 器件上四个边的管芯视图和封装视图



3.2.3.1. 覆盖差分管脚上的 I/O 布局规则

I/O 布局规则可确保噪声信号不会破坏相邻信号。每个器件系列都有预定义的 I/O 布局规则。

I/O 布局规则定义了相对于差分管脚支持的单端 I/O 布局，或者使用参考电压的输入标准时，您可以在 VREF 组中布局多少个输出和双向管脚。

使用 **IO_MAXIMUM_TOGGLE_RATE** assignment 来覆盖管脚上的 I/O 布局，例如常规设计活动中不会切换的系统复位管脚。对此 assignment 设置一个 0 MHz 值会导致 Fitter 在整个器件操作过程中标识 DC 状态上的管脚。Fitter 将分配的管脚从布局规则分析中排除。请勿将 0 MHz 的 **IO_MAXIMUM_TOGGLE_RATE** 分配给任何活动的切换管脚，否则设计可能无法正常运行。

3.2.4. 使用 Tcl 命令输入管脚分配

您可以使用 Tcl 脚本来应用管脚分配，通过在 Tcl Console 中输入单个 Tcl 命令，或者创建一个 `.tcl` 脚本并在命令行中输入以下命令：

实例-6: 应用 Tcl 脚本分配

```
quartus_sh -t <my_tcl_script>.tcl
```

实例-7: 脚本化管脚分配

以下示例使用 `set_location_assignment` 和 `set_instance_assignment` Tcl 命令将一个管脚分配到一个特定位置，I/O 标准和驱动强度。

```
set_location_assignment PIN_M20 -to address[10]
set_instance_assignment -name IO_STANDARD "2.5 V" -to address[10]
set_instance_assignment -name
        CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to address[10]
```

相关链接

Tcl Scripting
In *Intel Quartus Prime Pro Edition 用户指南：脚本*

3.2.5. 在 HDL 代码中输入管脚分配

您可以使用综合属性或低级 I/O 原语将 I/O 管脚分配直接嵌入在 HDL 代码中。分析和综合 HDL 代码时，信息将转换为相应的 I/O 管脚分配。您可以使用以下两种方法之一来使用 HDL 代码指定与管脚相关的分配：

- 为顶层管脚的信号名称分配综合属性
- 使用低级 I/O 原语(例如 ALT_BUF_IN)指定输入，输出和差分缓冲器，以及设置参数或属性

3.2.5.1. 使用低级 I/O 原语

您也可以使用低级 I/O 原语输入 I/O 管脚分配。您可以分配管脚位置，I/O 标准，驱动强度，摆率和片上端接(OCT)值分配。您还可以使用低级差分 I/O 原语在设计的 HDL 代码中定义差分对的正管脚和负管脚。

在执行完整的编译并向后注释管脚分配(**Assignments > Back Annotate Assignments**)之后，基于基元的分配才会出现在 Pin Planner 中。

相关链接

Designing with Low Level Primitives User Guide

3.3. 导入和导出 I/O 管脚分配

Intel Quartus Prime 软件支持工程之间的 I/O 管脚分配的传输，或者在第三方 PCB 工具中进行分析。您可以通过以下方式导入或导出 I/O 管脚分配：

表 17. 导入和导出 I/O 管脚分配

	Import Assignments	Export Assignments
Scenario	• 在早期管脚规划期间或在 PCB 工具中进行优化之后，从 PCB 设计工具或电子表格导入到 Pin Planner 中 • 从另一个具有通用约束的 Intel Quartus Prime 工程	• 从 Intel Quartus Prime 工程，以实现 PCB 设计工具中的优化 • 从 Intel Quartus Prime 工程，进行电子表格分析或者在脚本化分配中使用 • 从 Intel Quartus Prime 工程，导入到另一个具有类似约束的 Intel Quartus Prime 工程中
Command	**Assignments** > **Import Assignments**	**Assignments** > **Export Assignments**
File formats	`.qsf`, `.esf`, `.acf`, `.csv`, `.txt`,`.sdc`	`.pin`,`.fx`,`.csv`, `.tcl`, `.qsf`
Notes	N/A	导出的`.csv` 文件保留列和行顺序和格式。如果导入`.csv` 文件，请不要修改列标题的行

3.3.1. PCB 工具的导入和导出

Pin Planner 支持使用 PCB 工具导入和导出分配。您可以将有效的分配导出为一个**.pin** 文件，用于在其他支持的 PCB 工具中进行分析。您也可以从支持的 PCB 工具导入优化的分配。**.pin** 文件包含管脚名称，数量和详细属性。

Mentor Graphics I/O Designer 要求您生成并导入**.fx** 以及**.pin** 文件来传输分配。然而， Intel Quartus Prime 软件仅要求**.fx** 从 I/O Designer 导入管脚分配。

表 18. .pin 文件的内容

文件列名	说明
Pin Name/Usage	设计管脚的名称，或者管脚是 GND 还是 V_{CC} 管脚
Location	器件封装上位置的管脚号
Dir	管脚的方向
I/O Standard	管脚被配置成的 I/O 标准的名称
Voltage	需要连接到管脚的电压电平
I/O Bank	管脚属于的 I/O bank
User Assignment	Y 或者 N 表示设计管脚的位置分配是用户分配的(Y)还是 Fitter 分配的(N)

相关链接

- Pin-Out Files for Intel Devices
- PCB Design Tools Support
  In *Intel Quartus Prime Pro Edition User Guide: PCB Design Tools*

3.3.2. 移植分配到另一个目标器件

点击 **View** > **Pin Migration Window** 来验证管脚分配是否与移植到一个不同的 Intel 器件相兼容。

您可以将兼容的管脚分配从一个目标器件移植到另一个目标器件。您可以移植到一个不同密度的相同器件封装。您还可以在不同密度和管脚数的器件封装之间进行移植。

Intel Quartus Prime 软件会忽略无效的分配，并在编译过程中生成错误消息。在评估移植兼容性之后，修改所有不兼容的分配，然后单击 **Export** 以将分配导出到另一个工程中。

图 32. 器件移植兼容性(AC24 不存在于移植器件中)

Pin Migration View

Current Device: EP2S30F672C4

	Pin Number	Migration Result				Migration Devices											
						EP2S30F672C4				EP2S15F672C4				EP2S60F672C4			
		Pin Function	I/O Bank	VREF Group	Clock Pin	Pin Function	I/O Bank	VREF Group	Clock Pin	Pin Function	I/O Bank	VREF Group	Clock Pin	Pin Function	I/O Bank	VREF Group	Clock Pin
87	PIN_AC11	VREFB7N0	7	B7_N0		VREFB7N0	7	B7_N0		VREFB7N0	7	B7_N0		VREFB7N0	7	B7_N0	
88	PIN_AC12	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes
89	PIN_AC13	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes
90	PIN_AC14	Column I/O	8	B8_N1	Yes	Column I/O	8	B8_N1	Yes	Column I/O	8	B8_N1	Yes	Column I/O	8	B8_N2	Yes
91	PIN_AC15	NC				Column I/O	8	B8_N1		NC				Column I/O	12	B8_N2	Yes
92	PIN_AC16	VREFB8N1	8	B8_N1		VREFB8N1	8	B8_N1		VREFB8N1	8	B8_N1		VREFB8N2	8	B8_N2	
93	PIN_AC17	Column I/O	8	B8_N1		Column I/O	8	B8_N1		Column I/O	8	B8_N1		Column I/O	8	B8_N1	
94	PIN_AC18	Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N1		Column I/O	8	B8_N0	
95	PIN_AC19	Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0	
96	PIN_AC20	Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0	
97	PIN_AC21	Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0		Column I/O	8	B8_N0	
98	PIN_AC22	VREFB8N0	8	B8_N0		VREFB8N0	8	B8_N0		VREFB8N0	8	B8_N0		VREFB8N0	8	B8_N0	
99	PIN_AC23	VREFB1N2	1	B1_N2		Column I/O	8	B8_N0		NC				VREFB1N2	1	B1_N2	
100	PIN_AC24	NC				Row I/O	1	B1_N1		NC				Row I/O	1	B1_N1	
101	PIN_AC25	NC				Row I/O	1	B1_N1		NC				Row I/O	1	B1_N1	
102	PIN_AC26	VCCIO1	1			VCCIO1	1			VCCIO1	1			VCCIO1	1		
103	PIN_AD1	NC				Row I/O	6	B6_N0		NC				Row I/O	6	B6_N1	
104	PIN_AD2	NC				Row I/O	6	B6_N0		NC				Row I/O	6	B6_N1	
105	PIN_AD3	Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N2	
106	PIN_AD4	Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N2	
107	PIN_AD5	Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N2	
108	PIN_AD6	Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N2	
109	PIN_AD7	Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1		Column I/O	7	B7_N1	
110	PIN_AD8	Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N1	
111	PIN_AD9	Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N1	
112	PIN_AD10	Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0	
113	PIN_AD11	Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0		Column I/O	7	B7_N0	
114	PIN_AD12	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes
115	PIN_AD13	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes	Column I/O	10	B7_N0	Yes
116	PIN_AD14	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes	Column I/O	7	B7_N0	Yes

Device... Pin Finder... Show only highlighted pins Show migration differences Export...

突出显示的 PIN_AC23 的管脚功能的移植结果不是 NC，而是电压参考 VREFB1N2，即使管脚是移植器件中的 NC。VREF 标准具有比 NC 高的优先级，因此移植结果显示电压参考。即使在设计中不使用此管脚进行端口连接，也必须将 VREF 标准用于 I/O 标准，而 I/O 标准要求在移植器件的实际板级上使用此管脚。

如果其中一个移植器件包含用于连接到 V_{CC} 或 GND 的管脚，并且这些相同的管脚是移植路径中另一器件上的 I/O 管脚，那么 Intel Quartus Prime 软件确保这些管脚不用于 I/O。确保这些管脚连接到正确的 PCB 平面。

在同一封装中的两个器件之间移植时，未连接到较小晶片的管脚可能旨在连接到较大晶片上的 V_{CC} 或 GND。为了便于移植，您可以在原始设计中将这些管脚连接到 V_{CC} 或 GND，因为这些管脚实际上并未连接到较小晶片。

相关链接

AN90: SameFrame PinOut Design for FineLine BGA Packages

3.4. 验证管脚分配

Intel Quartus Prime 软件会根据目标器件的预定义 I/O 规则来验证 I/O 管脚分配。您可以使用以下工具在整个管脚规划过程中验证 I/O 管脚分配：

表 19. I/O 验证工具

I/O 验证工具	说明	点击运行
Advanced I/O Timing	根据编译期间所有 I/O 和时序检查来全面验证 I/O 分配	**Processing** > **Start Compilation**

3.4.1. I/O 分配验证规则

I/O Assignment Analysis 将根据以下规则验证分配:

表 20. I/O 规则检查的示例

规则	说明	是否需要 HDL?
I/O bank 性能	根据 I/O bank 中所允许的管脚数来检查分配给一个 I/O bank 的管脚数。	否
I/O bank VCCIO 电压兼容性	检查不超过一个 VCCIO 被要求用于分配给 I/O bank 的管脚。	否
I/O bank VREF 电压兼容性	检查不超过一个 VREF 被要求用于分配给 I/O bank 的管脚。	否
I/O 标准和位置冲突	检查管脚位置是否支持分配的 I/O 标准。	否
I/O 标准和信号方向冲突	检查管脚位置是否支持分配的 I/O 标准和方向。例如，特定管脚位置上的某些 I/O 标准只能支持输出管脚。	否
差分 I/O 标准不能开启开漏(open drain)	检查具有差分 I/O 标准的所有管脚的开漏是否都已关闭。	否
I/O 标准和驱动强度冲突	检查驱动强度分配是否在 I/O 标准的规格之内。	否
驱动强度和位置冲突	检查管脚位置是否支持分配的驱动强度。	否
BUSHOLD 和位置冲突	检查管脚位置是否支持 BUSHOLD。例如，专用时钟管脚不支持 BUSHOLD。	否
WEAK_PULLUP 和位置冲突	检查管脚位置是否支持 WEAK_PULLUP (例如，专用时钟管脚不支持 WEAK_PULLUP)。	否
电迁移检查	检查连续焊盘(consecutive pads)的综合驱动强度是否超过一定极限。例如，一个 Stratix II 器件上的连续 10 个焊盘的总电流驱动不能超过 200 mA。	否
PCI_IO 钳位二极管，位置和 I/O 标准冲突	检查管脚位置以及分配的 I/O 标准是否支持 PCI_IO 钳位二极管。	否
SERDES 和 I/O 管脚位置兼容性检查	检查设计中连接到 SERDES 的所有管脚都已分配给专用 SERDES 管脚位置。	是
PLL 和 I/O 管脚位置兼容性检查	检查连接到一个 PLL 的管脚是否分配给专用的 PLL 管脚位置。	是

表 21. 信号切换噪声规则

规则	说明	是否需要 HDL?
当存在 DPA 时，I/O bank 不能有单端 I/O	检查没有单端 I/O 管脚存在于相同的 I/O bank 中(作为 DPA)。	否
PLL I/O bank 不同时支持单端 I/O 和差分信号	检查存在差分信号时没有单端 I/O 管脚存在于 PLL I/O Bank 中。	否
单端输出要求与差分 I/O 管脚相距一定距离	检查单端输出是否与差分 I/O 管脚相距一定距离。	否
单端输出必须与 VREF pad 相距一定距离	检查单端输出管脚是否与 VREF pad 相距一定距离。	否
单端输入要求与差分 I/O 管脚相距一定距离	检查单端输入是否与差分 I/O 管脚相距一定距离。	否
当使用 VREF 时，VREFGROUP 中有过多的输出或者双向管脚	检查当使用 VREF 时 VREFGROUP 中是否有超过一定数量的输出或者双向管脚数量。	否
VREFGROUP 中有过多的输出	检查 VREFGROUP 中是否有过多的输出。	否

3.4.2. I/O 分配分析(I/O Assignment Analysis)

I/O 分配分析可根据整套 I/O 系统和电路板布局规则来验证 I/O 分配。完整的 I/O 分配分析可验证直接馈送或由 PLL，LVDS 或千兆位收发器模块之类的资源馈送的模块。此外，checker 还验证了正确使用 VREF 管脚，管脚位置以及可接受的混合 I/O 标准的合法性。

在早期管脚规划期间运行 I/O 分配分析，以在编译之前验证初始保留的管脚分配。定义设计文件后，请运行 I/O 分配分析以对综合网表执行更彻底的合法性检查。每当您修改 I/O 分配时，都要运行 I/O 分配分析。

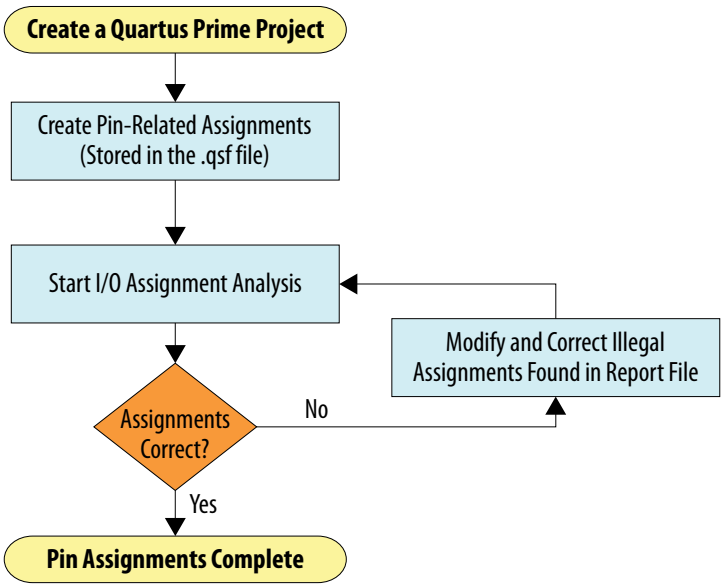
Fitter 分配管脚以满足您的约束。例如，如果将一个边沿位置分配给一组 LVDS 管脚，那么 Fitter 会为指定边沿位置中的每个 LVDS 管脚分配管脚位置，然后执行合法性检查。要显示 Fitter 布局的管脚，请在 Pin Planner 中点击 **Show Fitter Placements**。要接受建议的管脚位置，必须对管脚分配进行反注释。

查看 I/O Assignment Warnings 报告以查看并解决所有分配警告。例如，某些设计管脚有未定义的驱动强度或者摆率。Fitter 将未定义的单端输出和双向管脚标识为非校准的 OCT。要解决此警告，将 **Current Strength**，**Slew Rate** 或者 **Slow Slew Rate** 分配给报告的管脚。或者，将 **Termination** 分配 给此管脚。当一个管脚有 OCT 分配时，您不能分配驱动强度或者摆率。

3.4.2.1. 没有设计文件的早期 I/O 分配分析

您可以在定义 HDL 设计文件之前执行基本的 I/O 合法性检查。此技术可产生初步的电路板布局。例如，您可以指定目标器件并输入与 PCB 特性相对应的管脚分配。您可以为每个管脚保留并分配 I/O 标准，然后运行 I/O 分配分析以确保每个 I/O bank 中都没有 I/O 标准冲突。

图 33. 没有设计文件，分配和分析管脚(pin-out)

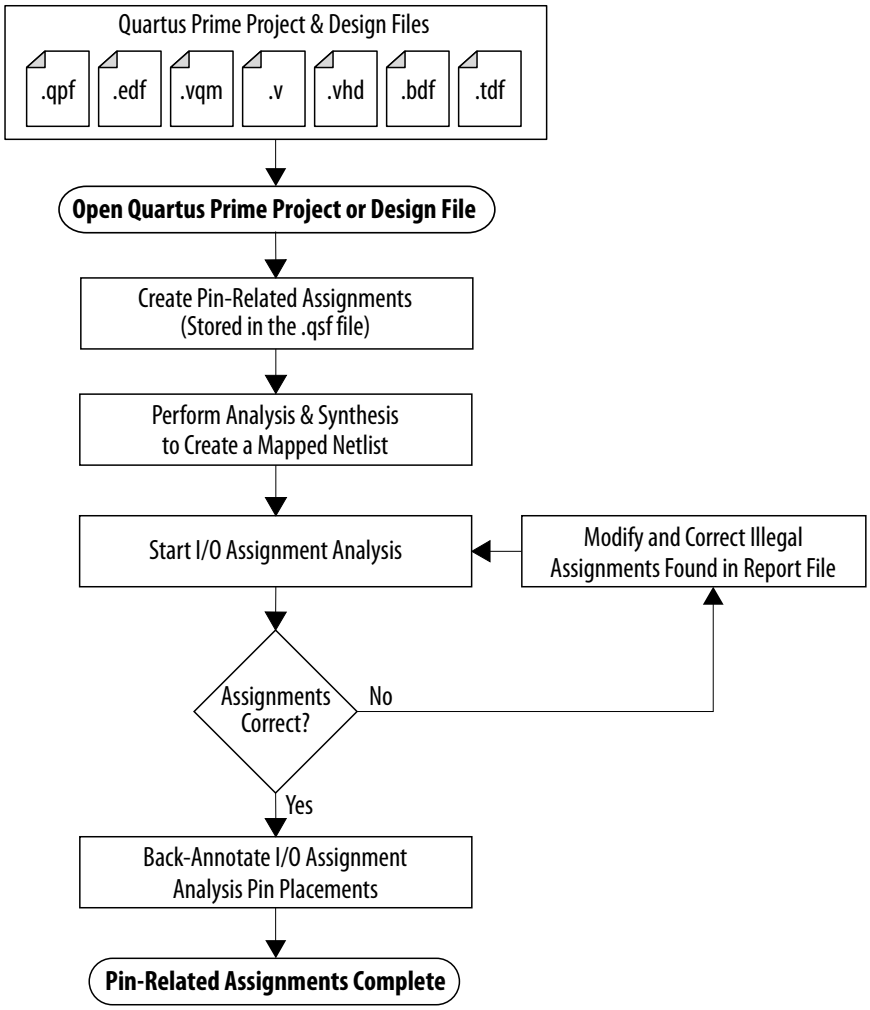


您必须保留打算用作 I/O 管脚的所有管脚，以便 Fitter 可以确定每种管脚类型。执行 I/O 分配分析后，更正 Fitter 报告的所有错误，然后重新运行 I/O 分配分析，直到纠正所有错误。一个完整的 I/O 分配分析需要所有设计文件。

3.4.2.2. 使用设计文件的 I/O 分配分析

I/O 分配分析使您能够在完全定义 HDL 设计文件之后执行完整的 I/O 合法性检查。当您在完整的设计上运行 I/O 分配分析时，此工具将根据所有 I/O 规则验证所有 I/O 管脚分配。在部分设计上运行 I/O 分配分析时，此工具仅检查设计中已定义部分的合法性。下图显示了使用设计文件分析管脚(pin-out)分配的工作流程。

图 34. I/O 分配分析流程

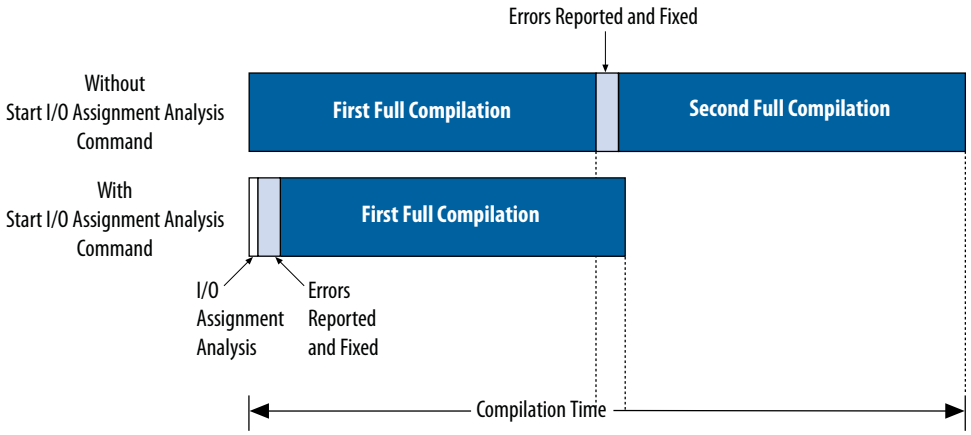


即使 I/O 分配分析在不完整的设计文件上通过，您也可能在完全编译期间遇到错误。例如，您可以将时钟分配给用户 I/O 管脚，而不是分配给专用时钟管脚，或者设计时钟来驱动尚未在设计中例化的 PLL。发生此问题的原因是，I/O 分配分析未考虑管脚驱动的逻辑，并且未验证只有专用时钟输入才能驱动 PLL 时钟端口。

为了获得更好的覆盖范围，需要尽可能多地分析设计，尤其是连接到管脚的逻辑。例如，如果您的设计包括 PLL 或 LVDS 模块，那么在进行全面分析之前定义这些文件。执行 I/O 分配分析后，更正 Fitter 报告的所有错误，然后重新运行 I/O 分配分析，直到纠正所有错误。

下图显示了在运行完整编译之前执行 I/O 分配分析的编译时间优势。

图 35. I/O 分配分析减少编译时间



3.4.2.3. 覆盖默认的 I/O 管脚分析

您可以覆盖默认的管脚 I/O 分析，以满足 I/O 规则异常，例如分析 VREF 或无效管脚。

每个器件都包含 VREF 引脚，每个管脚都支持一个或多个 I/O 管脚。VREF 管脚及其 I/O 管脚组成一个 VREF bank。VREF 管脚通常是使用 VREF I/O 标准分配的输入，例如 HSTL 和 SSTL 类型 I/O 标准。相反，VREF 输出不需要 VREF 管脚。当参考电压输入存在于 VREF bank 时，此 VREF bank 中只能存在一定数量的输出。I/O 分配分析将由不同输出使能控制的双向信号视为独立输出使能。

对双向信号分配 **Output Enable Group** 选项，将信号作为单一输出使能组进行分析，操作步骤如下：

1. 要在 Pin Planner 中访问此分分配，请右键点击 **All pins** 列表，然后点击 **Customize Columns**。
2. 在 **Available columns** 下，将 **Output Enable Group** 添加到 **Show these columns in this order**。此列出现在 **All Pins** 列表中。
3. 为在相同方向上驱动的所有信号集的 **Output Enable Group** 分配输入相同的整数值。

相关链接

Using the Timing Analyzer
In *Intel Quartus Prime Pro Edition User Guide: Timing Analyzer*

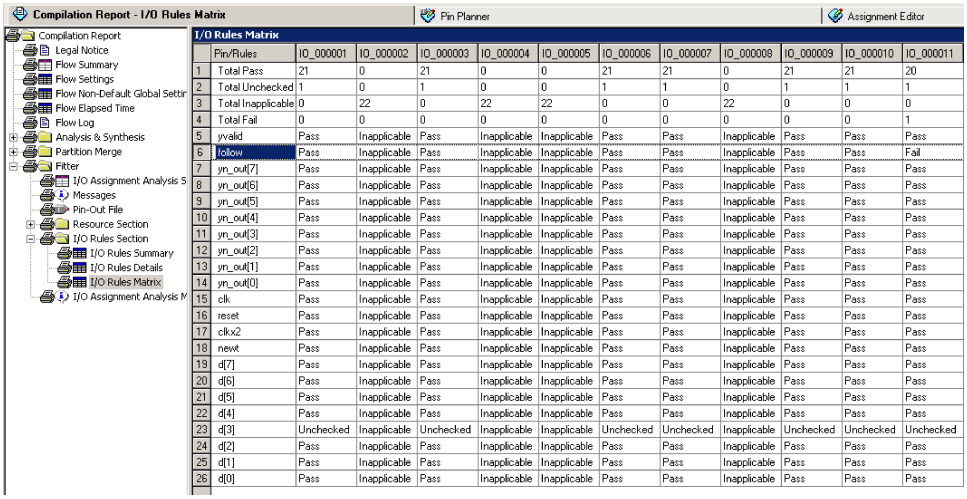
3.4.3. 了解 I/O 分析报告

详细的 I/O 分配分析报告包括受影响的管脚名称和问题描述。Compilation 报告的 Fitter 部分包含在 I/O 分配分析过程中生成的信息，包括以下报告：

- I/O Assignment Warnings—列出为每个管脚生成的警告消息
- Resource Section—量化各种管脚类型和 I/O bank 的使用
- I/O Rules Section—列出有关测试的 I/O 规则的汇总，详细信息和矩阵信息

Status 列指示规则是否通过，失败或未检查。严重等级表明规则对于有效分析的重要性。“Inapplicable”规则不应用于目标器件系列。

图 **36.** **I/O 规则矩阵**



	Pin/Rules	IO_000001	IO_000002	IO_000003	IO_000004	IO_000005	IO_000006	IO_000007	IO_000008	IO_000009	IO_000010	IO_000011
1	Total Pass	21	0	21	0	0	21	21	0	21	21	20
2	Total Unchecked	1	0	1	0	0	1	1	0	1	1	1
3	Total Inapplicable	0	22	0	22	22	0	0	22	0	0	0
4	Total Fail	0	0	0	0	0	0	0	0	0	0	1
5	yvalid	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
6	follow	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Fail
7	yn_out[7]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
8	yn_out[6]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
9	yn_out[5]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
10	yn_out[4]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
11	yn_out[3]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
12	yn_out[2]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
13	yn_out[1]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
14	yn_out[0]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
15	clk	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
16	reset	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
17	clkx2	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
18	newt	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
19	d[7]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
20	d[6]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
21	d[5]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
22	d[4]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
23	d[3]	Unchecked	Inapplicable	Unchecked	Inapplicable	Inapplicable	Unchecked	Unchecked	Inapplicable	Unchecked	Unchecked	Unchecked
24	d[2]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
25	d[1]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass
26	d[0]	Pass	Inapplicable	Pass	Inapplicable	Inapplicable	Pass	Pass	Inapplicable	Pass	Pass	Pass

3.5. 验证 I/O 时序

分配 I/O 管脚时，必须验证板级信号的完整性和 I/O 时序。高速接口操作需要在电路板布线的远端提供高质量信号和低传播延迟。点击 **Tools** > **Timing Analyzer** 以确认分配 I/O 管脚后的时序。

例如，如果使用 ECO 更改某些 I/O 管脚的摆率或驱动强度，那么无需重新编译设计即可验证时序。您必须了解 I/O 时序以及哪些因素会影响设计中的 I/O 时序路径。输出和双向管脚的输出负载规范的准确性会影响 I/O 时序结果。

Intel Quartus Prime 软件支持三种不同的 I/O 时序分析方法：

表 **22.** **I/O 时序分析方法**

I/O 时序分析	说明
高级 I/O 时序分析	通过电路板走线模型来分析 I/O 时序，以报告准确的，“可感知电路板(board-aware)”的仿真模型。为每个 I/O 标准或管脚配置完整的电路板走线模型。Timing Analyzer 应用 I/O 缓冲器，封装和板级走线模型的仿真结果来生成准确的 I/O 延迟和系统级信号信息。使用此信息可以提高时序和信号完整性。
I/O 时序分析	在没有信号完整性分析的情况下，使用默认或指定的容性负载来分析 I/O 时序。Timing Analyzer 使用电容负载的默认值或用户指定的值将 tCO 报告给 I/O 管脚。
全板级布线仿真	使用 Intel 提供的或者 Intel Quartus Prime 软件生成的 IBIS 或者 HSPICE I/O 模型在 Mentor Graphics HyperLynx*和 Synopsys HSPIC 中进行仿真。

关于高级 I/O 时序支持的更多信息，请参考目标器件的相应器件手册。关于板级信号完整性的更多信息以及有关如何在高速设计中提高信号完整性的技巧，请访问 Signal Integrity and Power Integrity – Support Center 网站。

关于使用 Intel Quartus Prime 软件创建 IBIS 和 HSPICE 模型以及将这些模型集成到 HyperLynx 和 HSPICE 仿真中的信息，请参考 *Signal Integrity Analysis with Third Party Tools* 章节。

相关链接

- Literature and Technical Documentation

- Signal Integrity and Power Integrity – Support Center

3.5.1. 运行高级 I/O 时序

高级 I/O 时序分析使用电路板走线模型和终端网络规范来报告准确的输出缓冲器到管脚时序评估，FPGA 管脚和电路板走线信号的完整性和延迟值。在编译过程中，支持的器件会自动运行高级 I/O 时序。

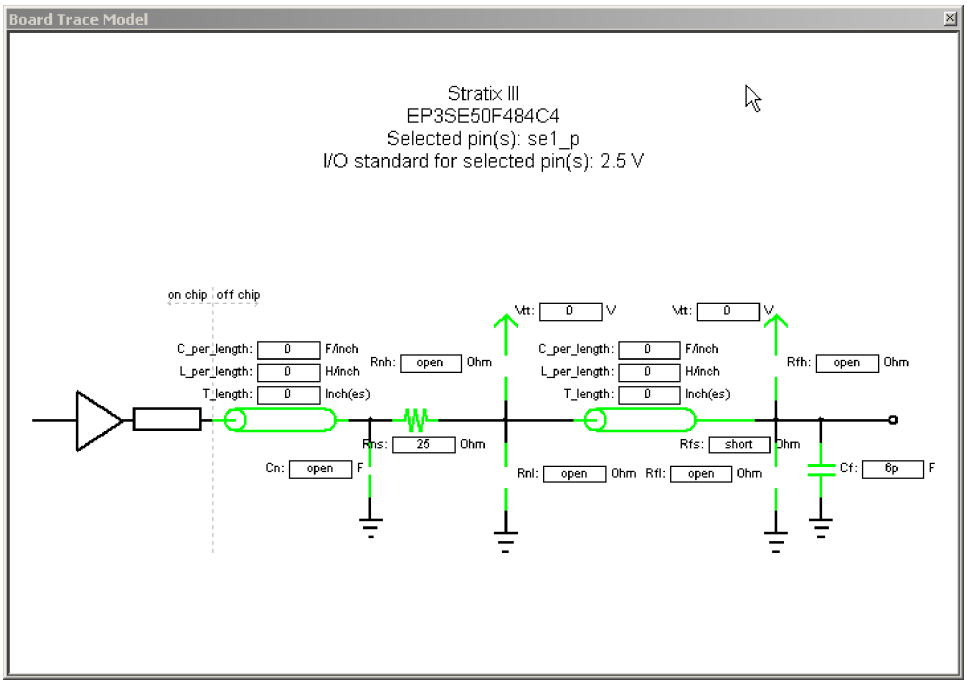
3.5.1.1. 电路板走线模型

Intel Quartus Prime 软件提供用于各种 I/O 标准的电路板走线模型模板。

下图显示了 **2.5 V** I/O 标准的模板。此模型由近端和远端板组件参数组成。

近端电路板走线建模包括靠近器件的单元。远端建模包括位于链路的接收器端，更靠近接收器件的单元。电路板走线模型拓扑是概念性的，不一定与每个组件的实际电路板走线都匹配。例如，近端模型参数可以代表器件端的离散端接和突围走线。远端建模可以代表到离散外部存储器组件以及远端终端网络的大部分电路板走线。您可以使用整个电路板的近端建模来分析同一电路，包括存储器组件的端接以及实际存储器组件的远端建模。

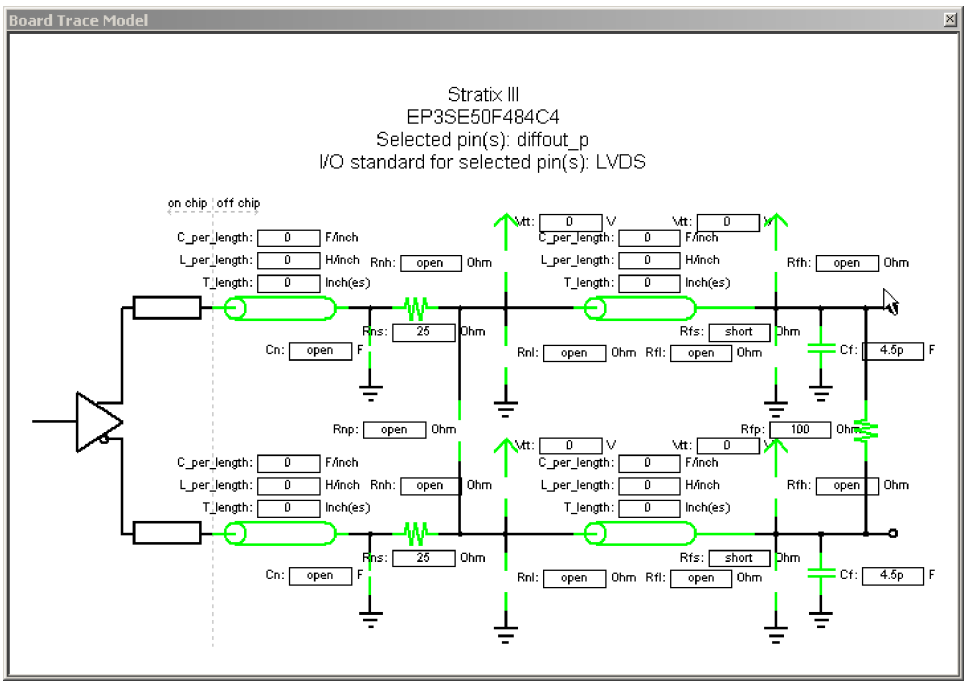
图 37. 2.5-V I/O 标准电路板走线模型



下图显示了 **LVDS** I O 标准的模板。远端电容(Cf)表示外部器件或多器件电容负载。如果远端有多个器件，那么必须考虑所有接收器电容，才能在远端找到等效电容。远端电容可以是所有接收器电容的总和。

传输线的 Intel Quartus Prime 软件模型不考虑传输线电阻(无损模型)。您只需以每英寸来指定分布电感(L)和电容(C)值，可从 PCB 供应商或制造商，CAD 设计工具或信号完整性工具(例如 Mentor Graphics HyperLynx 软件)获得。

图 **38.** **LVDS** 差分电路板走线模型



3.5.1.2. 定义电路板走线模型

电路板走线模型将电路板走线和端接网络描述为一组电容，电阻和电感参数。

Advanced I/O Timing 使用此模型来仿真从输出缓冲器到电路板走线远端的输出信号。您可以在输出模式下的任何输出管脚或双向管脚的电路板布线中定义电容性负载，任何端接组件和走线阻抗。您可以为每个 I/O 标准或特定的管脚配置整体的电路板走线模型。为设计中的每个 I/O 标准定义整体的电路板走线模型。对于使用 I/O 标准的所有管脚，请使用此模型。您可以使用 Pin Planner 中的 **Board Trace Model** 窗口为特定管脚自定义模型。

1. 点击 **Assignments** > **Device** > **Device and Pin Options**。
2. 点击 **Board Trace Model**，对每个 I/O 标准定义电路板走线模型值。
3. 点击 **I/O Timing**，然后定义电路板走线的近端和远端上的默认 I/O 时序选项。
4. 点击 **Assignments** > **Pin Planner**，将电路板走线模型值分配给单个管脚。

实例-8: 指定电路板走线模型

```
## setting the near end series resistance model of sel_p output pin to 25 ohms
set_instance_assignment -name BOARD_MODEL_NEAR_SERIES_R 25 -to se1_p
## Setting the far end capacitance model for sel_p output signal to 6 picofarads
set_instance_assignment -name BOARD_MODEL_FAR_C 6P -to se1_p
```

3.5.1.3. 修改电路板走线模型

要修改电路板走线模型，请点击 Pin Planner 中的 **View** > **Board Trace Model**。

您可以在电路板走线模型的图形表示内修改任何电路板走线模型参数。

Board Trace Model 窗口显示差分信号对中正信号和负信号的布线和组件。仅修改差分信号对的正信号，因为此设置会自动应用于负信号。使用标准单位前缀(例如 *p*，*n* 和 *k*)分别表示 pico，nano 和 kg。使用 **short** 或者 **open** 值可为并联组件指定短路或开路。

3.5.1.4. 指定近端与远端 I/O 时序分析

您可以对 I/O 时序分析选择一个近端点或远端点。近端时序分析扩展到器件管脚。您可以在近端分析期间应用 `set_output_delay` 约束，以解决整体延迟问题。

通过远端 I/O 时序分析，高级 I/O 时序分析扩展到了电路板走线远端的外部器件输入。无论您选择近端还是远端时序端点，时序分析期间都将考虑电路板走线模型。

3.5.1.5. 高级 I/O 时序分析报告

以下报告显示了高级 I/O 时序分析信息：

表 23. 高级 I/O 时序报告

I/O 时序报告	说明
Timing Analyzer Report	报告信号完整性和电路板延迟数据。
Board Trace Model Assignments report	总结了每个输出和双向信号的电路板走线模型组件设置。
Signal Integrity Metrics report	包含在高级 I/O 时序分析期间根据每个输出或双向管脚的电路板走线模型设置计算得出的所有信号完整性指标。包括在 FPGA 管脚和远端电路板延迟，稳态电压以及上升和下降时间的测量。

注意: 默认情况下，Timing Analyze 会生成 Slow-Corner Signal Integrity Metrics 报告。要生成一个 Fast-Corner Signal Integrity Metrics 报告，您必须通过点击 **Tools** > **Timing Analyzer** 对延迟模型进行更改。

相关链接

Using the Timing Analyzer
In *Intel Quartus Prime Pro Edition User Guide: Timing Analyzer*

3.5.2. 通过容性负载调整 I/O 时序和功耗

在计算输出和双向管脚的 t_{CO} 和功耗时，Timing Analyzer 和 Power Analyzer 使用大容性负载。您可以调整每个 I/O 标准的容性负载值，以获得更精确的 t_{CO} 和功耗测量值，以反映 PCB 上输出或双向网络的行为。 Intel Quartus Prime 软件会忽略输入管脚上的电容性负载设置。您可以按照整个 I/O 标准以皮法(pF)调整电容性负载设置。编译期间，Complier 会根据您的设置测量功耗和 t_{CO} 测量值。您还可以使用 **Output Pin Load** 逻辑选项来调整单个管脚上的电容负载。

3.6. 查看布线和时序延迟

右键单击任何节点，然后单击 **Locate > Locate in Chip Planner**，可视化并调整用户 I/O pad 与 V_{CC}，GND 和 V_{REF} pad 之间的 I/O 时序延迟和布线。Chip Planner 以图形方式显示逻辑位置，Logic Lock 区域，相对资源使用情况，详细的布线信息，扇入和扇出，寄存器路径以及高速收发器通道。您可以查看物理时序评估，布线拥塞和时钟区域。使用 Chip Planner 来更改资源之间的连接，并对逻辑单元和 I / O 原子放置进行编译后更改。当您在 Pin Planner 中选择项目时，相应的项目在 Pin Planner 中突出显示。

3.7. 脚本编写 API

Intel Quartus Prime 软件使您能够通过 Tcl 命令而不是通过 GUI 对 I/O 管理功能进行访问。关于脚本命令选项和 Tcl API 软件包的详细信息，请在系统命令提示符下键入以下命令，以查看 Tcl API Help browser：

```
quartus_sh --qhelp
```

相关链接

- Tcl Scripting
  Intel Quartus Prime Pro Edition 用户指南：脚本
- Command Line Scripting
  Intel Quartus Prime Pro Edition 用户指南：脚本

3.7.1. 生成映射的网表

在 Tcl 控制台或 Tcl 脚本中输入以下命令：

`execute_module -tool map`

`execute_module` 命令位于流程包中。

在系统命令符上输入下列命令：

```
quartus_syn <project name>
```

3.7.2. 保留管脚

使用以下 Tcl 命令来保留管脚：

`set_instance_assignment -name RESERVE_PIN <value> -to <signal name>`

使用以下有效的保留管脚值之一：

- `"AS BIDIRECTIONAL"`
- `"AS INPUT TRI STATED"`
- `"AS OUTPUT DRIVING AN UNSPECIFIED SIGNAL"`
- `"AS OUTPUT DRIVING GROUND"`
- `"AS SIGNALPROBE OUTPUT"`

注意: 指定保留的管脚值时，必须包含引号。

3.7.3. 设置位置

使用以下 Tcl 命令将信号分配给管脚或者器件位置：

```
set_location_assignment <location> -to <signal name>
```

有效位置是管脚位置，I/O bank 位置或者边沿位置。管脚位置包含管脚名称，例如：`PIN_A3`。I/O bank 位置包含 `IOBANK_1` 至 `IOBANK_` *n*，其中 *n* 是器件中的 I/O bank 的数量。

使用以下有效边沿位置值之一：

- EDGE_BOTTOM
- EDGE_LEFT
- EDGE_TOP
- EDGE_RIGHT

3.7.4. Exclusive I/O Group

以下 Tcl 命令创建一个 exclusive I/O group assignment：

```
set_instance_assignment -name "EXCLUSIVE_IO_GROUP" -to pin
```

3.7.5. 摆率和驱动强度

使用以下 Tcl 命令创建摆率和驱动强度分配：

```
set_instance_assignment -name CURRENT_STRENGTH_NEW 8MA -to e[0]
set_instance_assignment -name SLEW_RATE 2 -to e[0]
```

相关链接

Package Information Datasheet for Mature Altera Devices

3.8. 管理器件 I/O 管脚修订历史

下表显示了此章节的修订历史。

文档版本	Intel Quartus Prime 版本	修订内容
2018.05.07	18.0.0	• 首次发布，作为独立的 *Design Constraints User Guide* 的一部分
2017.11.06	17.1.0	• 修订了主题：I/O Planning Overview。 • 修订了主题：Basic I/O Planning Flow with the Pin Planner，重新命名为 Basic I/O Planning Flow with the Pin Planner。
2017.05.08	17.0.0	• 重命名命令：Run I/O Assignment Analysis to Start Fitter (Plan)。
2016.10.31	16.1.0	• 进行了 Intel 重命名。
2015.11.02	15.1.0	• 从 Quartus Prime Pro Edition 手册中删除了早期的管脚规划和 Live I/O Check 支持 • 将 *Quartus II* 修改成 *Quartus Prime*。
2014.12.15	14.1.0	• 更新了 Live I/O 检查器件支持，仅包括有限的器件系列。
2014.08.30	14.0a10.0	• 添加了有关 Arria 10 SoC 器件的特殊管脚分配功能信息的链接。
2014.06.30	14.0.0	• 将 MegaWizard Plug-In Manager 信息替换为 IP Catalog。
2013 年 11 月	13.1.0	• 重组并转换为 DITA。
2013 年 5 月	13.0.0	• 添加了有关覆盖 I/O 布局规则的信息。
2012 年 11 月	12.1.0	• 更新了新任务和报告窗口的 Pin Planner 描述。
2012 年 6 月	12.0.0	• 删除了调查链接。

继续...

文档版本	Intel Quartus Prime 版本	修订内容
2011 年 11 月	11.1.0	• 少量更新和更正。 • 更新了文档模板。
2010 年 12 月	10.0.1	模板更新
2010 年 7 月	10.0.0	• 重新组织和编辑章节 • 对于之前在章节中包含的程序信息将链接添加到 Help • 添加了有关在 I/O Rules Matrix Report 报告中标记为不适用的规则的信息 • 添加了有关将摆率和驱动强度设置分配给管脚以修复 I/O 分配警告的信息
2009 年 11 月	9.1.0	• 重新组织了整章，以包含指向本章先前包含的过程信息的 Help 链接。 • 添加了有关近端和远端高级 I/O 时序的文档
2009 年 3 月	9.0.0	• 更新了“Pad View Window” • 添加了新图: • 图 5–15 • 图 5–16 • 添加了新部分“Viewing Simultaneous Switching Noise (SSN) Results” • 添加了新部分“Creating Exclusive I/O Group Assignments”

相关链接

文档存档

关于 *Intel Quartus Prime Handbook* 的早前版本，请搜索文档存档。

A. Intel Quartus Prime Pro Edition 用户指南

请参阅以下用户指南获得关于 Intel Quartus Prime Pro Edition FPGA 设计流程中所有阶段的综合性信息。

相关链接

- Intel Quartus Prime Pro Edition 用户指南：入门
  介绍 Intel Quartus Prime Pro Edition 软件的基本功能，文件和设计流程，包括管理 Intel Quartus Prime Pro Edition 工程和 IP，初始设计布局考量以及从软件先前版本进行工程移植。
- Intel Quartus Prime Pro Edition 用户指南：平台设计程序
  说明使用 Platform Designer 创建和优化系统，该系统集成工具可简化工程中自定义 IP 核聚合。Platform Designer 自动生成互连逻辑以连接知识产权（IP）功能和子系统。
- Intel Quartus Prime Pro Edition 用户指南：设计建议
  介绍使用 Intel Quartus Prime Pro Edition 软件进行 FPGA 设计时的最佳设计实践。HDL 代码样式和同步设计时间可显著影响设计性能。以下建议的 HDL 代码样式可确保 Intel Quartus Prime Pro Edition 将设计在硬件中最佳实现。
- Intel Quartus Prime Pro Edition 用户指南：设计编译
  说明了 Intel Quartus Prime Pro Edition Compiler 从建立，运行到优化的全部阶段。生成器件编程文件之前，Compiler 对设计进行综合，布局和布线。
- Intel Quartus Prime Pro Edition 用户指南：设计优化
  介绍 Intel Quartus Prime Pro Edition 可用于实现 Intel FPGA 中最高设计性能的设置，工具和技术。技术包括优化设计网表，解决限制重定时和时序收敛的关键链，优化器件资源使用，器件布局规划以及实施工程变更单(ECO)。
- Intel Quartus Prime Pro Edition 用户指南：Programmer
  说明 Intel Quartus Prime Pro Edition Programmer 的运行，并通过连接 Intel FPGA 下载电缆配置 Intel FPGA 器件，编程 CPLD 和配置器件。
- Intel Quartus Prime Pro Edition 用户指南：基于块的设计
  说明基于块的设计流程，亦称为模块化或分层式设计流程。这些高级流程可将设计块（或是包含分层型设计实例的逻辑）保留在工程中，并在其他工程中重复使用设计块。
- Intel Quartus Prime Pro Edition 用户指南：局部重新配置
  介绍 Partial Reconfiguration 高级设计流程，其支持动态重配置 FPGA 某一部分的同时其余 FPGA 设计继续运行。将部分设计区域定义为多重角色时，并不影响其他区域中的操作。
- Intel Quartus Prime Pro Edition 用户指南：第三方仿真
  说明通过 Aldec*，Cadence*， Mentor Graphics 和 Synopsys 为第三方仿真工具提供 RTL-和门级设计仿真支持，从而允许在器件编程之前验证设计行为。包括仿真器支持，仿真流程和仿真 Intel FPGA IP。
- Intel Quartus Prime Pro Edition 用户指南：第三方综合
  说明通过 Mentor Graphics 和 Synopsys ，第三方综合工具为设计中选择性综合部分提供支持。包括设计流程步骤，生成的文件说明和综合指导。

英特尔公司。保留所有权利。英特尔、英特尔徽标和其他英特尔标志是英特尔公司或其子公司的商标。依照英特尔的标准保证条例，英特尔保证其 FPGA 和半导体产品的性能符合当前规格，但保留随时更改任何产品和服务的权利，恕不另行通知。英特尔概不承担因应用或使用本文中描述的任何信息、产品或服务而产生的任何责任和义务，除非得到英特尔书面上的明确同意。建议英特尔客户在信赖任何已发布的信息之前以及下单订购产品或服务之前，应先获取最新版本的器件规格。
*其他的名称和品牌可能是其他所有者的资产。

ISO 9001:2015 Registered

- Intel Quartus Prime Pro Edition 用户指南：第三方逻辑等效检查工具
  对 OneSpin*的第三方 LEC 工具中设计的可选逻辑等效性检查(LEC)的支持进行了描述。
- Intel Quartus Prime Pro Edition 用户指南：调试工具
  介绍为设计进行实时验证的 Intel Quartus Prime Pro Edition 在系统设计调试工具文件夹。这些工具通过将设计中的信号路由选择（或“分接”）到调试逻辑来提供可视性。具体工具包括，System Console，Signal Tap logic analyzer，Transceiver Toolkit，In-System Memory Content Editor 和 In-System Sources and Probes Editor。
- Intel Quartus Prime Pro Edition 用户指南：Timing Analyzer
  解释基本静态时序分析原则和 Intel Quartus Prime Pro Edition Timing Analyzer 的使用。其作为功能强大的 ASIC 式时序分析工具，通过使用行业标准的约束，分析和报告方法验证设计中所有逻辑的时序性能。
- Intel Quartus Prime Pro Edition 用户指南：功耗分析和优化
  说明 Intel Quartus Prime Pro Edition Power Analysis 工具支持准确估算器件功耗。估算器件功耗以开发功率预算和设计电源，稳压器，散热器和冷却系统。
- Intel Quartus Prime Pro Edition 用户指南：设计约束
  说明影响 Compiler 如何实现设计的时序和逻辑，例如，管脚约束，器件选项，逻辑选项和时序约束。使用 Interface Planner 原型开发接口实现，规划时钟并迅速定义合法器件平面图。使用 Pin Planner 在目标器件的图形呈现中可视化，修改和验证所有 I/O 约束。
- Intel Quartus Prime Pro Edition 用户指南：PCB 设计工具
  说明通过 Mentor Graphics 和 Cadence*实现对可选第三方 PCB 设计工具的支持。还包括信号集成分析和使用 HSPICE 和 IBIS 模型进行仿真的信息。
- Intel Quartus Prime Pro Edition 用户指南：脚本
  说明使用 Tcl 和命令行编制脚本进行 Intel Quartus Prime Pro Edition 软件控制并广泛执行各种功能，例如管理工程，指定约束，运行编译或时序分析，以及生成报告。