Intel® Quartus® Prime Pro Edition User Guide

Design Constraints

Updated for Intel® Quartus® Prime Design Suite: **18.1**

Subscribe

Send Feedback

UG-20142 | 2019.01.10

Latest document on the web: PDF | HTML

Contents

1. Constraining Designs

The design constraints, assignments, and logic options that you specify influence how the Intel® Quartus® Prime Compiler implements your design. The Compiler attempts to synthesize and place logic in a manner than meets your constraints. In addition, design constraints also have an impact on how the Timing Analyzer and the Power Analyzer influence synthesis, placement, and routing.

You can specify design constraints in the GUI, with scripts, or directly in the files that store the constraints. The Intel Quartus Prime software preserves the constraints that you specify in the GUI in the following files:

- Intel Quartus Prime Settings file (`<project_directory>/<revision_name>.qsf`)—contains project-wide and instance-level assignments for the current revision of the project, in Tcl syntax. Each revision of a project has a separate `.qsf` file.
- Synopsys* Design Constraints file (`<project_directory>/<revision_name>.sdc`)—the Timing Analyzer uses industry-standard Synopsys Design Constraint format and stores those constraints in `.sdc` files.

By combining the syntax of the `.qsf` files and the `.sdc` files with procedural Tcl, you can automate iterations over several different settings, changing constraints and recompiling.

Related Information

- Intel Quartus Prime Pro Edition Settings File Reference Manual
  For information about all settings and constraints in the Intel Quartus Prime software.
- Tcl Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*
- Command Line Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*

1.1. Specifying Design Constraints Designs in the GUI

Intel Quartus Prime software provides tools that help you manually implement your project. These tools can also support design visualization, pre-filled parameters, and window cross probing, facilitating design exploration and debugging.

When you create or update a constraint in the Intel Quartus Prime software, the **System** tab of the **Messages** window displays the equivalent Tcl command. Utilize these commands as references for future scripted design definition and compilation.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

1.1.1. Global Constraints and Assignments

Global constraints and project settings affect the entire Intel Quartus Prime project and all the applicable logic in the design. You often define global constraints in early project development; for example, when running the New Project Wizard. Intel Quartus Prime software stores global constraints in `.qsf` files, one for each project revision.

Table 1. Intel Quartus Prime Tools to Set Global Constraints

Assignment Type	Example	New Project Wizard	Device Dialog Box	Settings Dialog Box
Project-wide	Project files	X		X
Synthesis	• Device Family • Top-level Entity	X	X	X
Fitter	• Device • Fitter Effort • IO Standard		X	X
Simulation	Vector input source			X
Third-party Tools	External Logic Analyzer			X
IP Settings	Maximum Platform Designer Memory Usage			X

Related Information

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

1.1.2. Node, Entity, and Instance-Level Constraints

Node, entity, and instance-level constraints apply to a subset of the design hierarchy. These constraints take precedence over any global assignment that affects the same sections of the design hierarchy.

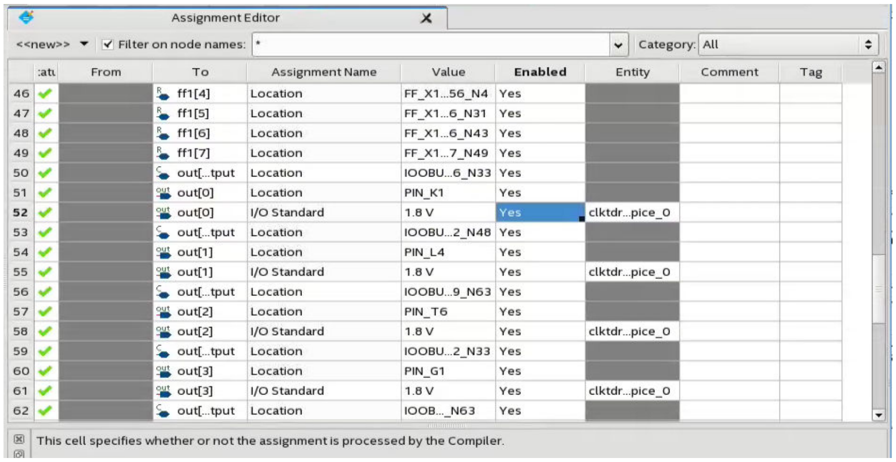
Table 2. Intel Quartus Prime Pro Edition Tools to Set Node, Entity and Instance Level Constraints

Assignment Type	Example	Assignment Editor	Interface Planner	Chip Planner	Pin Planner
Pin	Project files	X	X		X
Location	• Device Family • Top-level Entity	X	X	X	
Routing	• Device • Fitter Effort • IO Standard	X		X	
Simulation	Vector input source	X		X	X

1.1.2.1. Specify Instance-Specific Constraints in Assignment Editor

Intel Quartus Prime Assignment Editor (**Assignments ➤ Assignment Editor**) provides a spreadsheet-like interface for assigning all instance-specific settings and constraints. To help you explore your design, the Assignment Editor allows you to filter assignments by node name or category.

Figure 1. Intel Quartus Prime Assignment Editor



<<new>> ☑ Filter on node names: * Category: All

	:atu	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			

This cell specifies whether or not the assignment is processed by the Compiler.

Use the Assignment Editor to:

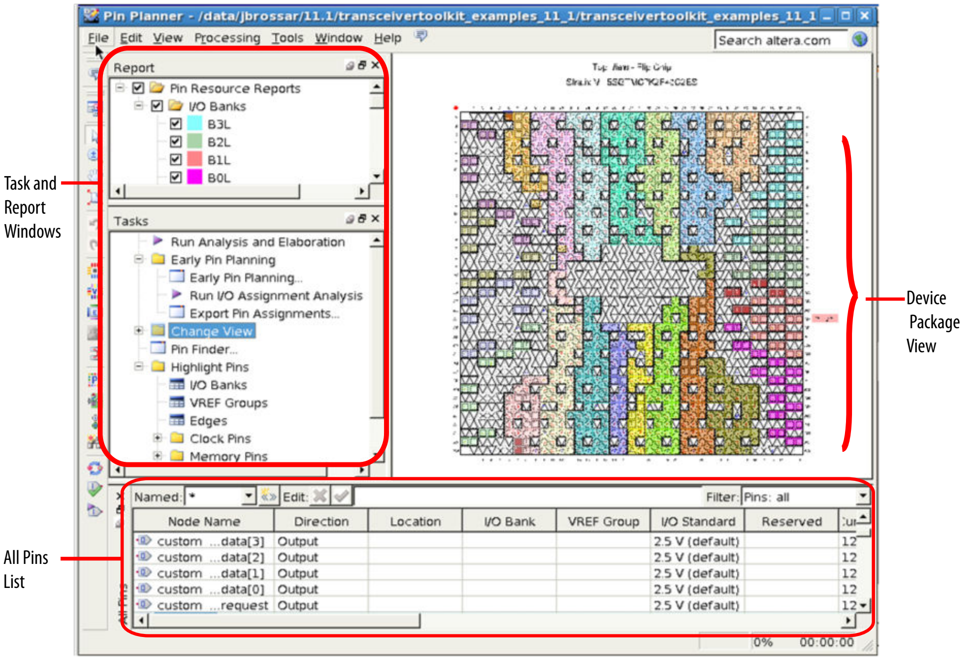
- Add, edit, or delete assignments for selected nodes
- Display information about specific assignments
- Enable or disable individual assignments
- Add comments to an assignment

Additionally, you can export assignments to a Comma-Separated Value File (`.csv`).

1.1.2.2. Specify I/O Constraints in Pin Planner

Intel Quartus Prime Pin Planner allows you to assign design elements to I/O pins. You can also plan and assign IP interface or user nodes not yet defined in the design.

Figure 2. Pin Planner GUI



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Cur
custom ...data[3]	Output				2.5 V (default)		12
custom ...data[2]	Output				2.5 V (default)		12
custom ...data[1]	Output				2.5 V (default)		12
custom ...data[0]	Output				2.5 V (default)		12
custom ...request	Output				2.5 V (default)		12

Related Information

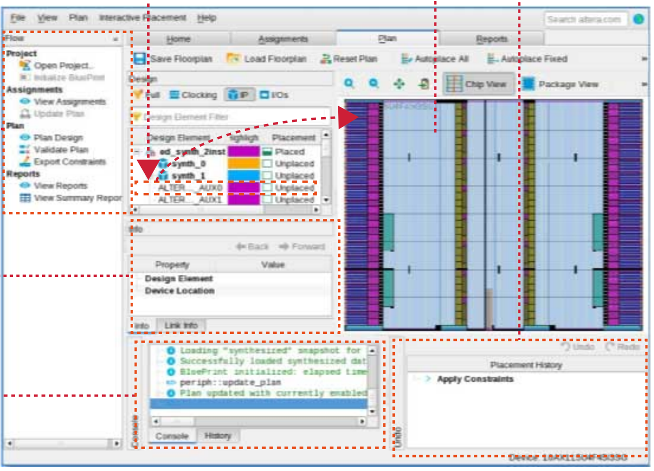
Managing Device I/O Pins on page 41

1.1.2.3. Plan Interface Constraints in Interface Planner

The Interface Planner simplifies the planning of accurate constraints for physical implementation. Use Interface Planner to prototype interface implementations, plan clocks, and rapidly define a legal device floorplan.

Interface Planner interacts dynamically with the Intel Quartus Prime Fitter to accurately verify placement legality while you plan. You can evaluate different floorplans, using interactive reports to accurately plan the best implementation without iterative compilation. Fitter verification ensures the highest correlation between your interface plan and actual implementation results. You can apply the interface plan constraints to your project with high confidence in the final implementation.

Figure 3. Interface Planner GUI



Design Element	Highlight	Placement
ed_synth_2inst	🟪	Placed
synth_0	🟧	Unplaced
synth_1	🟦	Unplaced
ALTER..._AUX0	🟪	Unplaced
ALTER..._AUX1	🟪	Unplaced

Property	Value
Design Element	
Device Location	

```
Loading "synthesized" snapshot for
Successfully loaded synthesized dat
BluePrint initialized: elapsed time
periph::update_plan
Plan updated with currently enabled
```

Drag or click Design Elements to place in legal locations

Undo/Redo Placement History

Ctrl+Click to pan

Flow controls provide quick access to planning functions

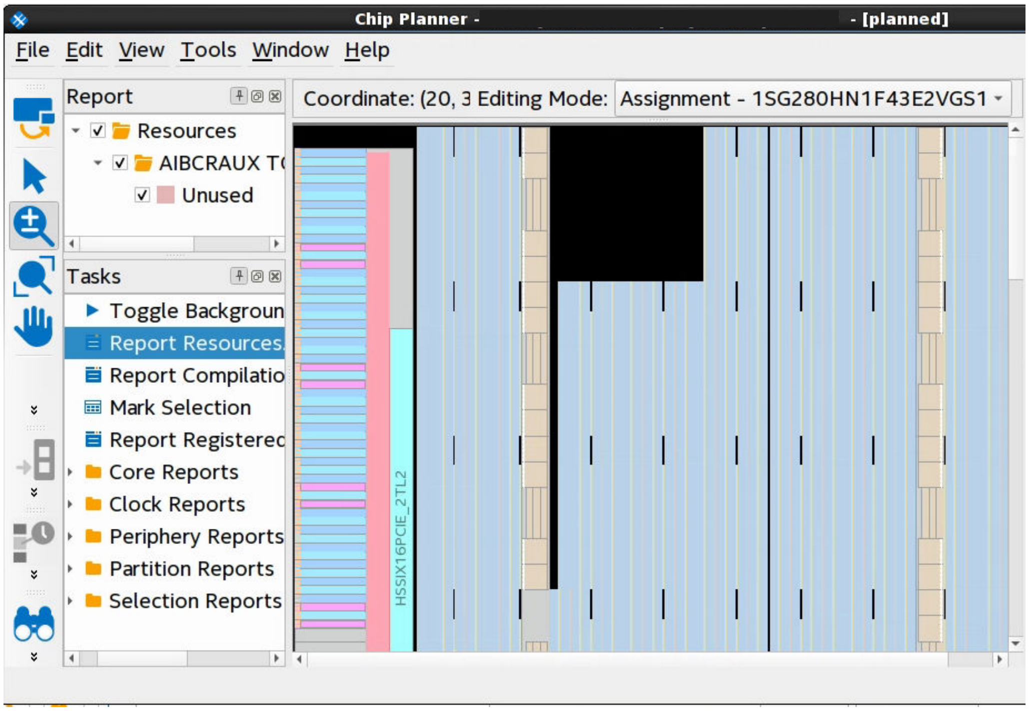
Selected element's properties

Tcl Console window

1.1.2.4. Adjust Constraints with the Chip Planner

With the Chip Planner you can adjust existing assignments to device resources, such as pins, logic cells, and LABs in a graphical representation of the device floorplan. You can also view equations and routing information and demote assignments by dragging and dropping to Logic Lock regions in the **Logic Lock Regions Window**.

Figure 4. Chip Planner GUI



Related Information

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

1.1.2.5. Constraining Designs with the Design Partition Planner

The Design Partition Planner allows you to view design connectivity and hierarchy and can assist you in creating effective design partitions.

Additionally, the Design Partition Planner allows you to optimize design performance by isolating and resolving failing paths on a partition-by-partition basis.

Related Information

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. Probing Between Components of the Intel Quartus Prime GUI

Intel Quartus Prime software allows you to locate nodes and instances across windows and source files.

When you are in the Project Navigator, Assignment Editor, Chip Planner, or Pin Planner, and want to display a given resource in other Intel Quartus Prime tool:

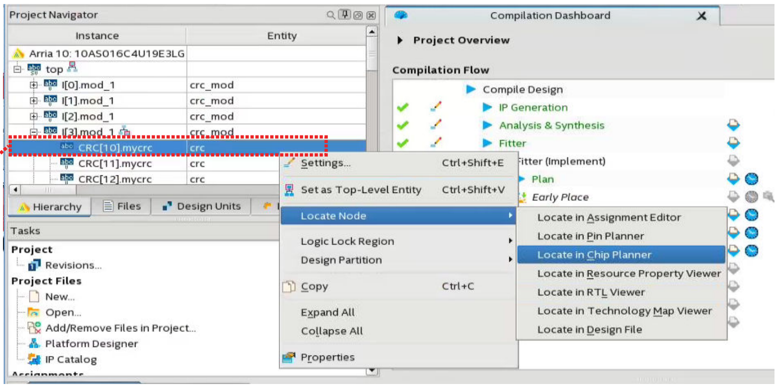
1. Right-click the resource you want to display.
2. Click **Locate Node**, and then click one of the menu options.

The corresponding window opens—or appears in the foreground if it is already open—and shows the element you clicked.

Example 1. Locate a Resource Selected in the Project Navigator

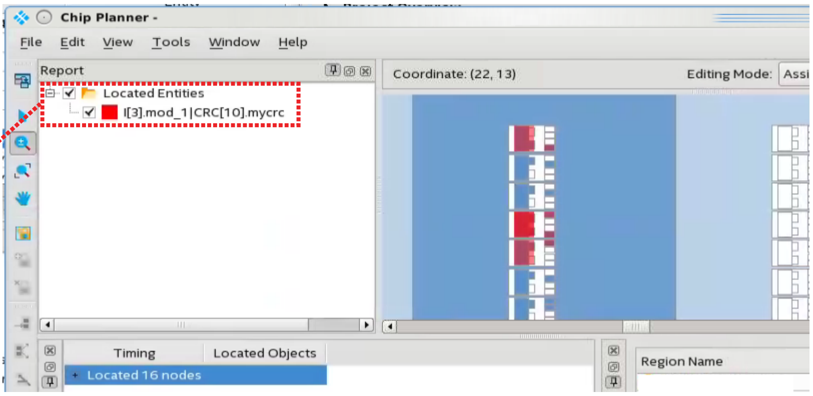
In the **Entity** list of the **Hierarchy** tab, right-click one object, and click **Locate ➤ Locate in Chip Planner**.

Right-click Instance in Hierarchy Tab



The Chip Planner opens and displays the instance you selected.

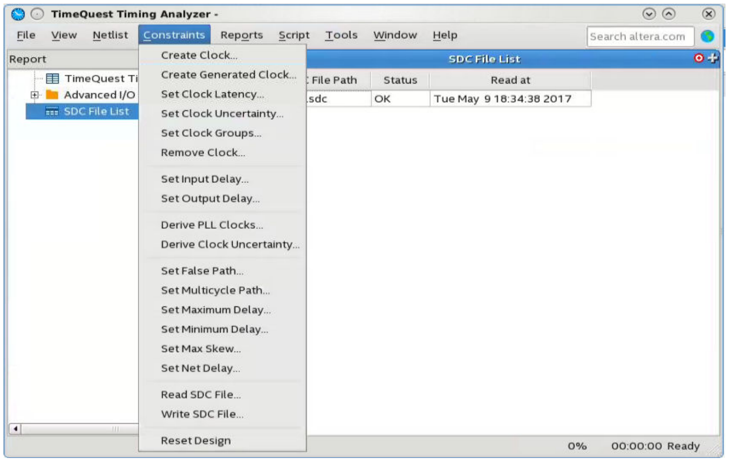
Chip Planner displays and keeps resource selected



1.1.4. Specifying Timing Constraints in the GUI

You can specify timing constraints in the Timing Analyzer GUI. Click the Constraints menu in the Timing Analyzer to specify timing constraints that you can apply to your project.

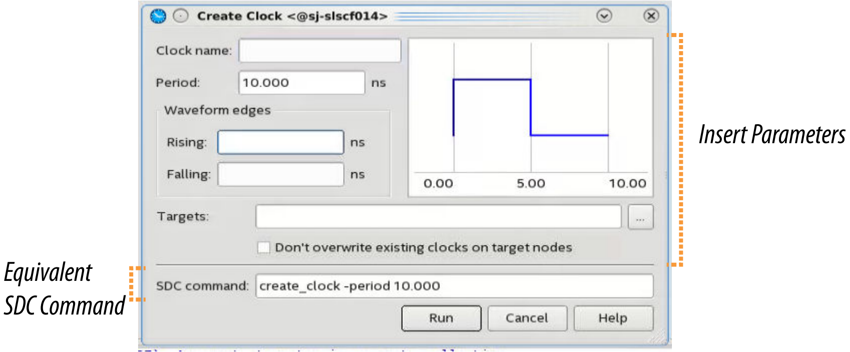
Figure 5. Constraint menu in Timing Analyzer



File Path	Status	Read at
.sdc	OK	Tue May 9 18:34:38 2017

When you specify a constraint in the GUI, the dialog box displays the equivalent SDC command syntax.

Example 2. Create Clock Dialog Box



Clock name:

Period: 10.000 ns

Rising: ns

Falling: ns

Targets:

□ Don't overwrite existing clocks on target nodes

SDC command: create_clock -period 10.000

Individual timing assignments override project-wide requirements.

- To avoid reporting incorrect or irrelevant timing violations, you can assign timing exceptions to nodes and paths.
- The Timing Analyzer supports point-to-point timing constraints, wildcards to identify specific nodes when making constraints, and assignment groups to make individual constraints to groups of nodes.

Related Information

Using Timing Constraints
For descriptions of all Constraints menu commands

1.2. Constraining Designs with Tcl Scripts

You can perform all your design assignments using `.sdc` and `.qsf` setting files. To integrate these files in compilation and optimization flows, use Tcl scripts. Even though `.sdc` and `.qsf` files are written in Tcl syntax, they are not executable by themselves.

When you use Intel Quartus Prime Tcl packages, your scripts can open projects, make the assignments, compile the design, and compare compilation results against known goals and benchmarks. Furthermore, such a script can automate the iterative design process by modifying constraints and recompiling the design.

1.2.1. Create a Project and Apply Constraints

The command-line executables include options for common global project settings and commands. You can use a Tcl script to apply constraints such as pin locations and timing assignments. You can write a Tcl constraint file, or generate one for an existing project by clicking **Project ➤ Generate Tcl File for Project**.

The example creates a project with a Tcl script and applies project constraints using the tutorial design files in the <Intel Quartus Prime *installation directory*>/`qdesigns/fir_filter/` directory.

```
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 BDF_FILE filtref.bdf
# 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
```

Save the script in a file called `setup_proj.tcl` and type the commands illustrated in the example at a command prompt to create the design, apply constraints, compile the design, and perform fast-corner and slow-corner timing analysis. Timing analysis results are saved in two files, `filtref_sta_1.rpt` and `filtref_sta_2.rpt`.

```
quartus_sh -t setup_proj.tcl
quartus_map 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
```

Type the following commands to create the design, apply constraints, and compile the design, without performing timing analysis:

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

The `quartus_sh --flow compile` command performs a full compilation, and is equivalent to clicking the **Start Compilation** button in the toolbar.

1.2.2. Assigning a Pin

To assign a signal to a pin or device location, use the Tcl command shown in this example:

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

Valid locations are pin location names. Some device families also support edge and I/O bank locations. Edge locations are EDGE_BOTTOM, EDGE_LEFT, EDGE_TOP, and EDGE_RIGHT. I/O bank locations include IOBANK_1 to IOBANK_n, where n is the number of I/O banks in a device.

1.2.3. Generating Intel Quartus Prime Settings Files

Intel Quartus Prime software allows you to generate .qsf files from your revision. You can embed these constraints in a scripted compilation flow, and even create sets of .qsf files for design optimization.

To generate a .qsf file from the Intel Quartus Prime software, click **Assignments ➤ Export Assignments**.

To organize the .qsf in a human readable form, **Project ➤ Organize Intel Quartus Prime Settings File**.

Example 3. Organized .qsf File

This example shows how .qsf files characterize a design revision. The set_global_assignment command makes all global constraints and software settings and set_location_assignment constrains each I/O node in the design to a physical pin on the device.

```
# 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)
# ---------------
```

Related Information

Intel Quartus Prime Pro Edition Settings File Reference Manual
For information about all settings and constraints in the Intel Quartus Prime software.

1.2.4. Synopsys Design Constraint (.sdc) Files

Intel Quartus Prime software keeps timing constraints in `.sdc` files, which use Tcl syntax. You can embed these constraints in a scripted compilation flow, and even create sets of `.sdc` files for timing optimization.

Example 4. .sdc File

The example shows the timing constrains of a small design.

```
## 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
```

Related Information

Constraining and Analyzing with Tcl Commands

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

1.2.5. Tcl-only Script Flows

As an alternative to `.sdc` and `.qsf` files, you can perform all design assignments and timing constraints inside the Tcl scripts. In this case, the script that automates compilation and custom results reporting also contains the design constraints.

You can export a design's contents to a procedural, executable Tcl (`.tcl`) file, and then use the generated script to restore settings after experimenting with other constraints.

To export your constraints as an executable Tcl script, click **Project ➤ Generate Tcl File for Project**.

Example 5. bliking_led_generated.tcl File

```
# 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
}
}
```

The example:

- Opens the project
- Assigns Constraints
- Writes assignments to QSF file
- Closes project

1.2.5.1. Tcl-only Timing Analysis

To avoid using a separated file to keep your timing constraints, copy and paste the `.sdc` file into your executable timing analysis script.

1.3. A Fully Iterative Scripted Flow

The **::quartus::flow** Tcl package in the Intel Quartus Prime Tcl API allows you to modify design constraints and recompile in an iterative flow.

Related Information

- ::quartus::flow
  In *Intel Quartus Prime Help*
- Command Line Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*

1.4. Constraining Designs Revision History

Document Version	Intel Quartus Prime Version	Changes
2019.01.04	18.1.0	• Clarified default location of .sdc and .qsf files in "Constraining Designs" topic. • Added "Plan Interface Constraints with Interface Planner" topic. • Added screenshots to "Constrain Designs with the Pin Planner" and "Constrain Designs with the Chip Planner." • Added two new "Assigning a Pin" and "Creating a Project and Applying Constraints" topics showing Tcl examples. • Added link to Using Timing Constraints topic in Timing Analyzer UG that explains all of the commands
2017.11.06	17.1.0	• Renamed topic: Constraining Designs with the GUI to Constraining Designs with Quartus Prime Tools. • Renamed topic: Global Constraints to Global Constraints and Assignments. • Added table: Quartus Prime Tools to Set Global Constraints. • Removed topic: Common Types of Global Constraints. • Removed topic: Settings That Direct Compilation and Analysis Flows. • Updated topic: Node, Entity and Instance-Level Constraints. • Added table: Quartus Prime Tools to Set Node, Entity and Instance Level Constraints. • Added topic: Assignment Editor. • Updated topic: Constraining Designs with the Pin Planner. • Updated topic: Constraining Designs with the Chip Planner. • Added topic: Constraining designs with the Design Partition Planner. • Updated topic: Probing Between Components of the Quartus Prime GUI. • Added example: Locate a Resource Selected in the Project Navigator. • Updated topic: SDC and the Timing Analyzer, and renamed to Specifying Individual Timing Constraints. • Added figure: Constraint Menu in Timing Analyzer. • Added example: Create Clock Dialog Box. • Updated topic: Constraining Designs with Tcl, and renamed to Constraining Designs with Tcl Scripts • Updated topic: Quartus Prime Settings Files and Tcl , and renamed to Generating Quartus Prime Settings Files. • Added example: blinking_led.qsf File. • Updated topic: Timing Analysis with Synopsys Design Constraints and Tcl, and renamed to Timing Analysis with .sdc Files and Tcl Scripts. • Added example: .sdc File with Timing Constraints. • Added topic: Tcl-only Script Flows. • Updated topic: A Fully Iterative Scripted Flow.
2017.05.08	17.0.0	• Removed references to deprecated Fitter Effort logic option.
2016.10.31	16.1.0	• Implemented Intel rebranding.
2015.11.02	15.1.0	• Changed instances of *Quartus II* to *Intel Quartus Prime*.
June 2014	14.0.0	Formatting updates.
November 2012	12.1.0	Update Pin Planner description for task and report windows.
June 2012	12.0.0	Removed survey link.
November 2011	10.0.2	Template update.
December 2010	10.0.1	Template update.

continued...

Document Version	Intel Quartus Prime Version	Changes
July 2010	10.0.0	Rewrote chapter to more broadly cover all design constraint methods. Removed procedural steps and user interface details, and replaced with links to Quartus II Help.
November 2009	9.1.0	• Added two notes. • Minor text edits.
March 2009	9.0.0	• Revised and reorganized the entire chapter. • Added section “Probing to Source Design Files and Other Quartus Windows” on page1–2. • Added description of node type icons (Table1–3). • Added explanation of wildcard characters.
November 2008	8.1.0	Changed to 8½” × 11” page size. No change to content.
May 2008	8.0.0	Updated Quartus II software 8.0 revision and date.

Related Information

Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the documentation archives.

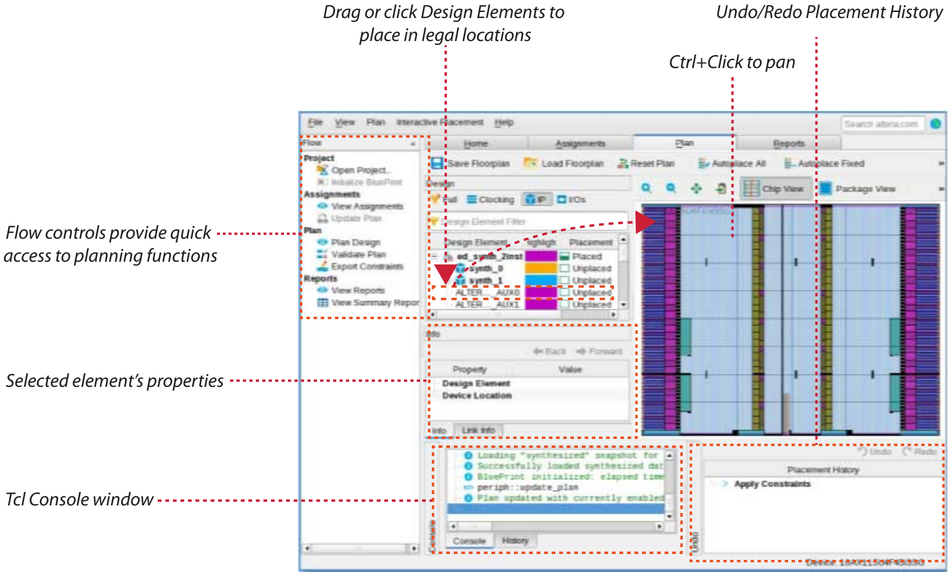
2. Interface Planning

Interface planning—the feasibility analysis of interface physical constraints—is a fundamental early step in advanced FPGA design. Periphery placement can be a complex process involving many variables. The Interface Planner simplifies the planning of accurate constraints for physical implementation. Use Interface Planner to prototype interface implementations, plan clocks, and rapidly define a legal device floorplan.

Note: Intel now refers to BluePrint as Interface Planner.

Interface Planner interacts dynamically with the Intel Quartus Prime Fitter to accurately verify placement legality while you plan. You can evaluate different floorplans, using interactive reports to accurately plan the best implementation without iterative compilation. Fitter verification ensures the highest correlation between your interface plan and actual implementation results. You can apply the interface plan constraints to your project with high confidence in the final implementation.

Figure 6. Interface Planner GUI



Drag or click Design Elements to place in legal locations

Undo/Redo Placement History

Ctrl+Click to pan

Flow controls provide quick access to planning functions

Selected element's properties

Tcl Console window

Design Element	Highligh	Placement
ed_synth_2inst	🟪	Placed
synth_0	🟧	□ Unplaced
synth_1	🟦	□ Unplaced
ALTER..._AUX0	🟪	□ Unplaced
ALTER..._AUX1	🟪	□ Unplaced

Property	Value
Design Element	
Device Location	

```
Loading "synthesized" snapshot for
Successfully loaded synthesized dat
BluePrint initialized: elapsed time
periph::update_plan
Plan updated with currently enabled
```

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

Interface Planner Features

Interface Planner provides support for the following:

- Plan legal design periphery floorplans at any stage of the design process.
- Analyze and modify clock network scenarios to direct the placement of high fan-out signals.
- Automatically evaluate all placement legality using the Fitter.
- Save and reload various floorplan files.
- GUI and Tcl command-line operation.
- Intel Arria® 10, Intel Cyclone® 10 GX, and Intel Stratix® 10 device families.

Related Information

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

2.1. Interface Planning Overview

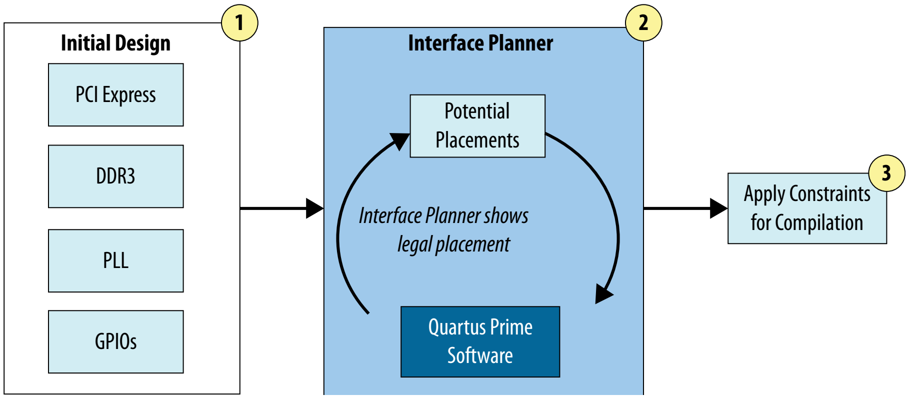
After design synthesis, use Interface Planner to rapidly define a legal device floorplan.

Interface planning involves

- Setting up your project
- Initializing the Interface Planner
- Updating project assignments
- Planning periphery placement
- Exporting your plan constraints to your Intel Quartus Prime project.

Note: The plan constraints that you define in Interface Planner do not automatically apply to your Intel Quartus Prime project until you export the Interface Planner constraints to your project via a generated Tcl script.

Figure 7. Interface Planner Streamlines Legal Placement



Intel FPGAs contain core and periphery device locations. The device core locations are adaptive look-up tables (ALUTs), core flip-flops, RAMs, and digital signal processors (DSPs). Device periphery locations include I/O elements, phase-locked loops (PLLs), clock buffers, and hard processor systems (HPS).

Intel's advanced FPGAs contain many silicon features in the periphery, such as hard PCI Express® IP cores, high speed transceivers, hard memory interface circuitry, and embedded processors. Interactions among these periphery atoms can be complex. Interface Planner simplifies this complexity and allows you to quickly visualize and place I/O interface and periphery elements, such as:

- I/O elements
- LVDS interfaces
- PLLs
- Clocks
- Hard interface IP Cores
- High-Speed Transceivers
- Hard Memory Interface IP Cores
- Embedded Processors

After initialization, Interface Planner displays your project's logical hierarchy, post-synthesis design elements, and Fitter-created design elements, alongside a view of target device locations. The GUI supports a variety of methods for placing design elements in the floorplan. As you place elements in the floorplan, the Fitter verifies legality in real time to ensure accurate correlation with the final implementation.

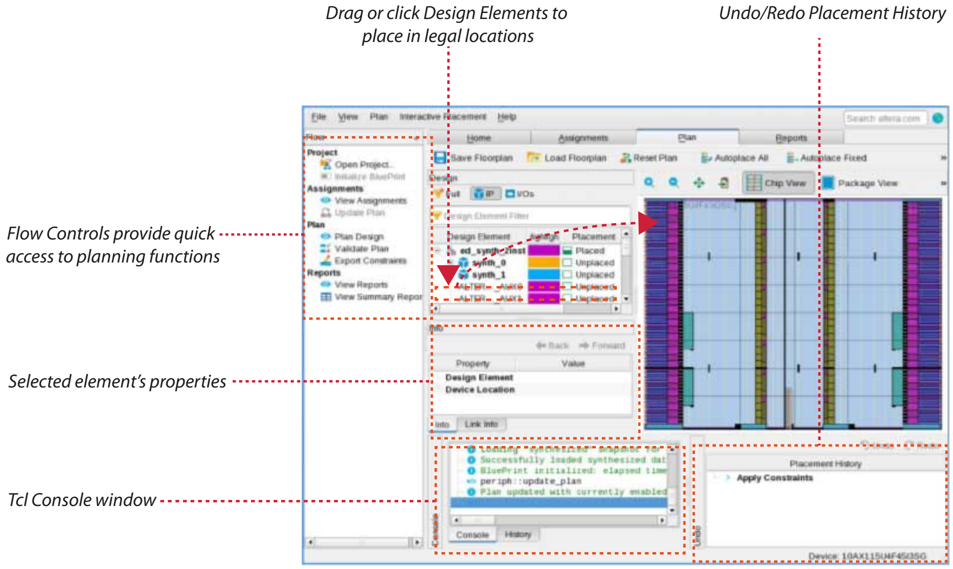
Related Information

Managing Device I/O Pins on page 41

2.2. Using Interface Planner

Interface Planner's user interface guides you through the design planning steps. You proceed through each Interface Planner tab as you successively complete the setup, initialization, placement, and reporting steps. Interface Planner allows you to move forward only after you complete the prerequisite steps.

Figure 8. Interface Planner GUI



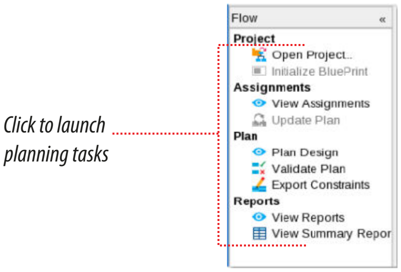
```
Loading "synthesized" snapshot for
Successfully loaded synthesized dat
BluePrint initialized: elapsed time
periph::update_plan
Plan updated with currently enabled
```

Table 3. Interface Planner User Interface Reference

Use this Interface Planner Tab	For This Task
Home	Load a project and initialize Interface Planner. Use of the **Home** tab is only required when using Interface Planner in standalone mode.
Assignments	Resolve (disable) any conflicting assignments from your synthesized project. Click **Update Plan** to apply your enabled project assignments to the interface plan.
Plan	Interactively plan placement with real-time validation and feedback in a graphical floorplan of the device.
Reports	Generate detailed, interactive reports to inform implementation decisions.
Tcl Console window	View and enter Tcl commands that control Interface Planner user.
Undo/Redo Placement History	Display the history of placement operations in Interface Planner. The **Undo** button reverts the last change made in the **Plan** tab. **Redo** re-implements the last **Undo** operation. Use these buttons to step forward and backward though your plan changes.

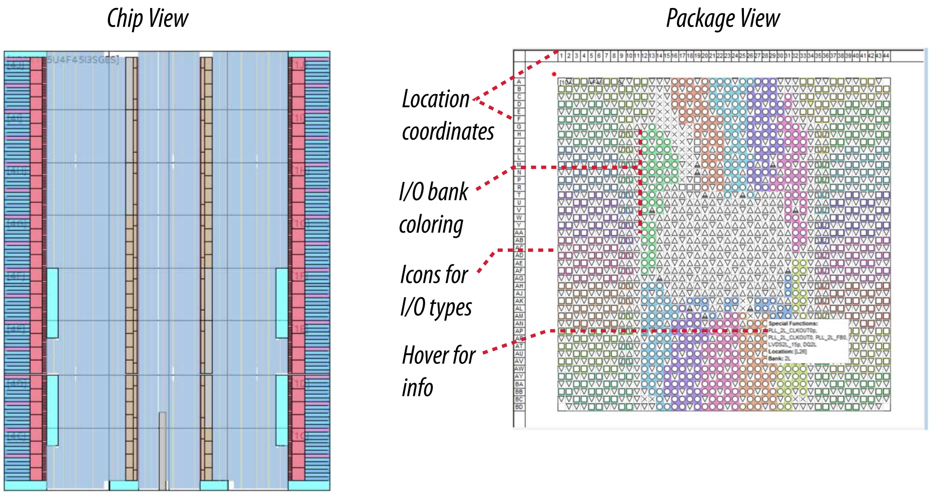
Interface Planner's **Flow** control provides immediate access to the main tasks in the interface planning flow. The **Flow** controls appear in order of a typical interface planning flow. Click the **Flow** controls to complete the setup and initialization tasks in order.

Figure 9. Interface Planner Flow Control



Interface Planner accounts for any imported project assignments. You can disable or enable imported project assignments to resolve any conflicts and evaluate different implementations. After setup and initialization, you can place design elements onto the **Chip View** or **Package View** of the target device. During placement, Interface Planner provides real-time, accurate feedback about legal placement options verified by the Fitter.

Figure 10. Interface Planner Chip and Package Views



After you are satisfied with the interface plan for important interfaces, you validate the plan to confirm that the Fitter can place all remaining logic. On validation, you can apply the interface plan to your project by sourcing a Tcl file that Interface Planner generates.

Step 1: Setup the Project on page 24

Step 2: Initialize Interface Planner on page 24

Step 3: Update Plan with Project Assignments on page 25

Step 4: Plan Periphery Placement on page 25

Step 5: Report Placement Data on page 30

Step 6: Validate and Export Plan Constraints on page 31

2.2.1. Step 1: Setup the Project

Interface Planner requires at least a partially complete, synthesized Intel Quartus Prime project as input. You can also use Interface Planner to place a fully complete design project. Before planning in Interface Planner, prepare the Intel Quartus Prime project in the following ways:

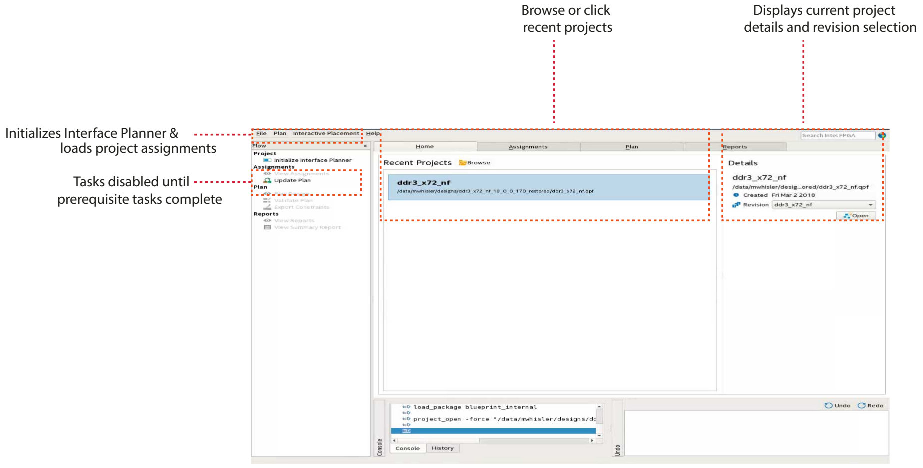
- Fully define known device periphery interfaces.
- Instantiate all known interface IP cores.
- Declare all general purpose I/Os.
- Define the I/O standard, voltage, drive strength, and slew rate for all general purpose I/Os.
- Define the core clocking (optional, but recommended).
- Connect all interfaces of the periphery IP to virtual pins or test logic. This technique creates loop backs on any interfaces in the shell design, helping to ensure that periphery interfaces persist after synthesis optimization.

2.2.2. Step 2: Initialize Interface Planner

To open a project in Interface Planner:

1. Click **Tools ➤ Interface Planner**. The Interface Planner opens, displaying the **Home** tab.
2. Click **Initialize Interface Planner** on the **Flow** control to start the Fitter validation and import any project assignments. After initialization, the Fitter dynamically validates your interface plan as you make changes.

Figure 11. Interface Planner Home Tab



Browse or click recent projects

Displays current project details and revision selection

Initializes Interface Planner & loads project assignments

Tasks disabled until prerequisite tasks complete

```
tcl> load_package blueprint_internal
tcl>
tcl> project_open -force "/data/mwhisler/designs/dc
tcl>
tcl>
```

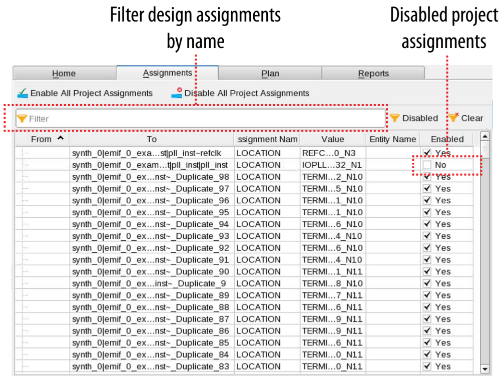
Note: The changes that you make in Interface Planner do not apply to your Quartus Prime project until you export and apply the interface plan constraints to your project.

2.2.3. Step 3: Update Plan with Project Assignments

Before periphery planning in Interface Planner, you must reconcile conflicting imported project assignments and **Update Plan** with the assignments.

1. Click **View Assignments** on the **Flow** control to review imported project assignments and reconcile any conflicts. You can filter the list of assignments by assignment name or status. Enable or disable specific or groups of project assignments to resolve any conflicts.
2. Click **Plan ➤ Reset Plan** if you want to return all project assignments to the enabled state.
3. After resolving all conflicts, click **Update Plan** on the **Flow** control to apply the enabled project assignments to your interface plan.

Figure 12. Interface Planner (Assignments Tab)


Filter design assignments by name
Disabled project assignments
Home
Assignments
Plan
Reports
Enable All Project Assignments
Disable All Project Assignments
Filter
Disabled
Clear

From	To	ssignment Nam	Value	Entity Name	Enabled
	synth_0|emif_0_exa…st|pll_inst~refclk	LOCATION	REFC…0_N3		☑ Yes
	synth_0|emif_0_exam…t|pll_inst|pll_inst	LOCATION	IOPLL…32_N1		□ No
	synth_0|emif_0_ex…nst~_Duplicate_98	LOCATION	TERMI…2_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_97	LOCATION	TERMI…5_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_96	LOCATION	TERMI…1_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_95	LOCATION	TERMI…1_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_94	LOCATION	TERMI…6_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_93	LOCATION	TERMI…4_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_92	LOCATION	TERMI…6_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_91	LOCATION	TERMI…4_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_90	LOCATION	TERMI…1_N11		☑ Yes
	synth_0|emif_0_ex…inst~_Duplicate_9	LOCATION	TERMI…8_N10		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_89	LOCATION	TERMI…7_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_88	LOCATION	TERMI…6_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_87	LOCATION	TERMI…9_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_86	LOCATION	TERMI…9_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_85	LOCATION	TERMI…6_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_84	LOCATION	TERMI…0_N11		☑ Yes
	synth_0|emif_0_ex…nst~_Duplicate_83	LOCATION	TERMI…0_N11		☑ Yes

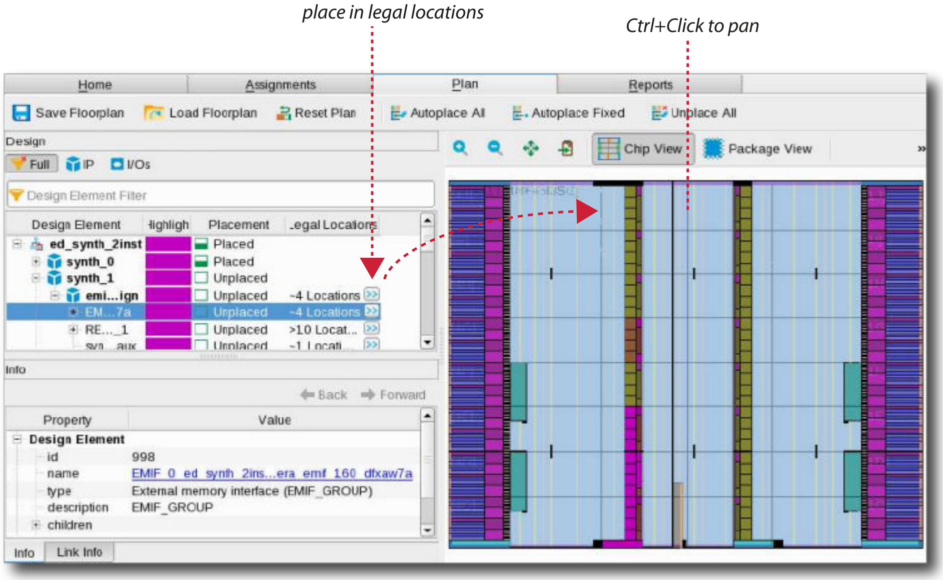
Related Information

- Home Tab Controls on page 32
- Assignments Tab Controls on page 32

2.2.4. Step 4: Plan Periphery Placement

Click **Plan Design** on the **Flow** control to interactively place IP cores and other design elements in legal locations in the device periphery. The **Plan** tab displays a list of your project's design elements, alongside a graphical abstraction of the target device architecture.

Figure 13. Interface Planner (Plan Tab)



Drag or click Design Elements to place in legal locations

Ctrl+Click to pan

Design Element	ighligh	Placement	egal Locations
ed_synth_2inst	🟪	Placed	
synth_0	🟪	Placed	
synth_1	🟪	□ Unplaced	
emi...ign	🟪	□ Unplaced	~4 Locations
EM...7a	🟪	Unplaced	~4 Locations
RE..._1	🟪	□ Unplaced	>10 Locat...
syn...aux	🟪	□ Unplaced	~1 Locati...

Property	Value
Design Element	
id	998
name	EMIF_0_ed_synth_2ins...era_emif_160_dfxaw7a
type	External memory interface (EMIF_GROUP)
description	EMIF_GROUP
children	

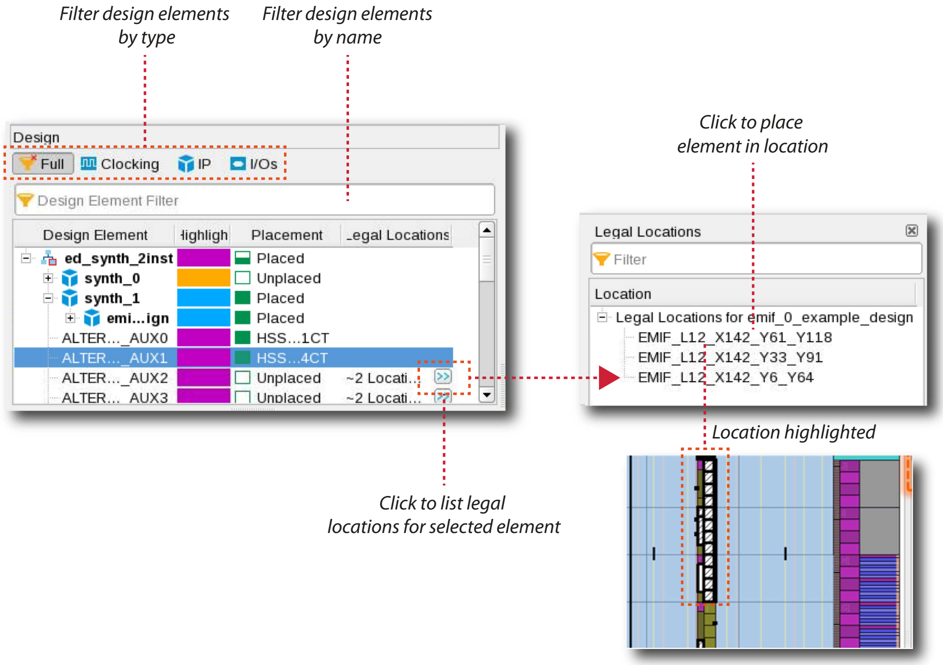
Use the following controls to place design elements in the Interface Planner floorplan:

- Click the >> button next to any design element to display a list of **Legal Locations**. Click any legal location in the list to highlight the location in the floorplan. Double-click any legal location in the list to place in the floorplan.
- Alternatively, drag elements from the **Design Elements** list and drop them onto available device resources in the **Chip** or **Package** view.

  Note: The drag-and-drop method can experience delays while calculating a list of legal locations.
- Right-click and select **Autoplace Selected** to have Interface Planner set placement for an unplaced design element. All unplaced clocks require that you use **Autoplace Selected**.
- Use the **Undo** and **Redo** buttons to step forward and backward though your plan changes.
- Use Ctrl+Click to drag and pan across the **Chip** or **Package** views.
- Use the **Link Info** tab to visualize and traverse design connectivity. For example, you can view the reference clock pin and driven destination cells of a PLL. Often these links have significant impact on placement flexibility because of dedicated routing paths for periphery features. Use the Back and Forward buttons to traverse design connectivity.
- If the **Locate Node** command is disabled for a specific element in the **Design Elements** list, it is because the element doesn't possess a design instance name, and is not included in the hierarchical database.

Note: The **Clocking** controls are not supported for the Intel Stratix 10 device family. You must use the **Autoplace Selected** control to place unplaced clock design elements in designs that target Intel Stratix 10 devices.

Figure 14. Listing Legal Locations



Filter design elements by type

Filter design elements by name

Click to place element in location

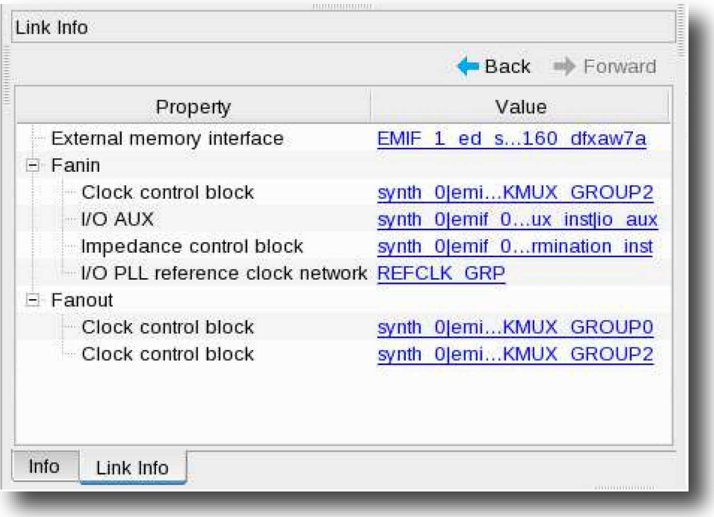
Location highlighted

Click to list legal locations for selected element

Design Element	ighligh	Placement	egal Locations
ed_synth_2inst	🟪	Placed	
synth_0	🟧	Unplaced	
synth_1	🟦	Placed	
emi...ign	🟦	Placed	
ALTER..._AUX0	🟪	HSS...1CT	
ALTER..._AUX1	🟪	HSS...4CT	
ALTER..._AUX2	🟪	Unplaced	~2 Locati...
ALTER..._AUX3	🟪	Unplaced	~2 Locati...

Location
- Legal Locations for emif_0_example_design
  - EMIF_L12_X142_Y61_Y118
  - EMIF_L12_X142_Y33_Y91
  - EMIF_L12_X142_Y6_Y64

Figure 15. Link Info Tab for Traversing Connectivity



Property	Value
External memory interface	EMIF_1_ed_s...160_dfxaw7a
Fanin	
Clock control block	synth_0|emi...KMUX_GROUP2
I/O AUX	synth_0|emif_0...ux_inst|io_aux
Impedance control block	synth_0|emif_0...rmination_inst
I/O PLL reference clock network	REFCLK_GRP
Fanout	
Clock control block	synth_0|emi...KMUX_GROUP0
Clock control block	synth_0|emi...KMUX_GROUP2

You can apply filters to locate specific design elements or target specific device resources for placement. Device resources are color coded by type, allowing quick access to any structure. Click the **Highlight** column to customize the color definitions. Place elements in the following order for efficiency:

1. Place all I/O pins or elements, such as PLLs, that have known, specific location requirements.
2. Place all known periphery interface IP.
3. (Optional) Place all remaining unplaced cells (**Autoplace Fixed** or right-click any individual unplaced element to **Autoplace Selected Element**).

Select any physical design element and click **Report Placeability of Selected Element** to generate a report that shows the placement locations the Fitter prefers. Right-click reported locations to directly place cells in the location. Show error messages for unsupported locations to help understand the device constraints. You can **Export** the report content for analysis in other tools.

Note: Changes made in Interface Planner do not apply to your Quartus Prime project until you apply the generated interface plan constraints script to your project.

Related Information

Plan Tab Controls on page 33

2.2.4.1. Plan Clock Networks

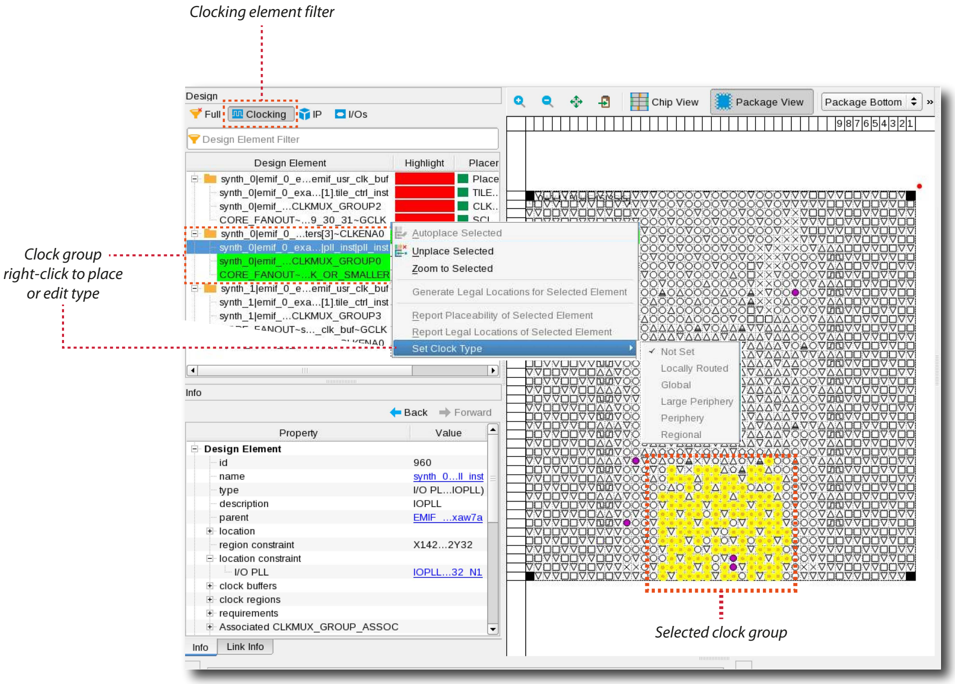
Interface Planner allows you to visualize and plan clock networks. You can locate, highlight, place, and edit the type of clocking design elements in the **Plan** tab. Interface Planner generates a Clocks report that details the signals using low-skew routing networks (clock networks) in the device.

Note: The Intel Stratix 10 device family does not support **Clocking** controls. You must use the **Autoplace Selected** control to place unplaced clock design elements in designs that target Intel Stratix 10 devices.

To identify and place clocking elements in your design, click the **Clocking** filter in the **Plan** tab. You can refine the list further by entering search text in the **Design Element Filter**. Interface Planner represents clock networks as groupings of the following clock network elements:

- Clock source
- Clock mux
- Clock region

Figure 16. Clocking Design Elements


Clocking element filter
Clock group right-click to place or edit type
Selected clock group
Design
Full
Clocking
IP
I/Os
Design Element Filter
Design Element
Highlight
Placer
synth_0|emif_0_e...emif_usr_clk_buf
synth_0|emif_0_exa...[1].tile_ctrl_inst
synth_0|emif_...CLKMUX_GROUP2
CORE_FANOUT~...9_30_31~GCLK
synth_0|emif_0_...ters[3]~CLKENA0
synth_0|emif_0_exa...|pll_inst|pll_inst
synth_0|emif_...CLKMUX_GROUP0
CORE_FANOUT~...K_OR_SMALLER
synth_1|emif_0_e...emif_usr_clk_buf
synth_1|emif_0_exa...[1].tile_ctrl_inst
synth_1|emif_...CLKMUX_GROUP3
CORE_FANOUT~s..._clk_buf~GCLK
Place
TILE..
CLK..
SCL
Autoplace Selected
Unplace Selected
Zoom to Selected
Generate Legal Locations for Selected Element
Report Placeability of Selected Element
Report Legal Locations of Selected Element
Set Clock Type
✓ Not Set
Locally Routed
Global
Large Periphery
Periphery
Regional
Chip View
Package View
Package Bottom
9 8 7 6 5 4 3 2 1
Info
Back
Forward
Info
Link Info

Property	Value
Design Element	
id	960
name	synth_0...ll_inst
type	I/O PL...IOPLL)
description	IOPLL
parent	EMIF_...xaw7a
location	
region constraint	X142...2Y32
location constraint	
I/O PLL	IOPLL...32_N1
clock buffers	
clock regions	
requirements	
Associated CLKMUX_GROUP_ASSOC	

You can place an entire clock group or individual clock elements by dragging into the location, or using the **Report Legal Locations of Selected Element** or the **Autoplace Selected** commands. After placement, hover the cursor over the item in the **Design Element** list to highlight the placement. The placement of clock elements impacts the placement of dependent core and periphery elements.

You can edit the clock type for clocking design elements. The clock type impacts the placement of dependent core and periphery elements. Right-click any clock element to specify one of the following clock types:

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

2.2.4.2. Saving & Loading Floorplans

You can save the state of your Interface Planner floorplan for use in subsequent Interface Planner sessions. Interface Planner saves your plan in Interface Planner Floorplan Format (`.plan`). You can load a `.plan` file in Interface Planner to reopen the floorplan.

1. To save an Interface Planner floorplan, click **File ➤ Save Floorplan** and specify a file name.
2. To load a Interface Planner floorplan, click **File ➤ Load Floorplan** and browse for the `.plan` file.

Note: `.plan` files are for use only in Interface Planner and are not for use directly in the Intel Quartus Prime software. Interface Planner generates an error if you attempt to load a `.plan` file that is not associated with the current Interface Planner project.

2.2.5. Step 5: Report Placement Data

Generate Interface Planner placement and connectivity reports to help locate cells and make the best decisions about placement for the interfaces and elements in your design. Click **View Reports** on the **Flow** control to open the **Reports** tab from which you can generate a range of reports.

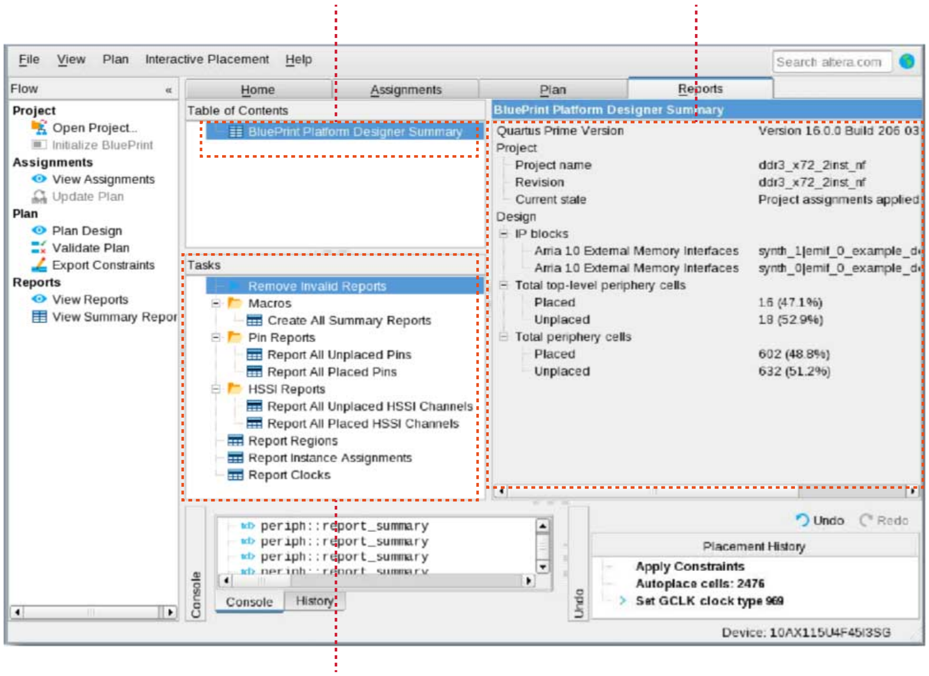
Click any report in the **Tasks** list to generate the report. The Interface Planner reports are dynamic, allowing you to select elements in the report and place, unplace, or report detailed data about the selected elements or locations.

Select any element in the **Plan** tab to **Report Placeability of Selected Element**, or **Report Legal Locations of Selected Element**. The generated reports appear on the **Reports** tab. This document describes each report in detail.

Figure 17. Reports Tab

Click to view generated reports

Report viewer



Flow
- Project
  - Open Project...
  - Initialize BluePrint
- Assignments
  - View Assignments
  - Update Plan
- Plan
  - Plan Design
  - Validate Plan
  - Export Constraints
- Reports
  - View Reports
  - View Summary Repor

Table of Contents
- BluePrint Platform Designer Summary

Tasks
- Remove Invalid Reports
- Macros
  - Create All Summary Reports
- Pin Reports
  - Report All Unplaced Pins
  - Report All Placed Pins
- HSSI Reports
  - Report All Unplaced HSSI Channels
  - Report All Placed HSSI Channels
- Report Regions
- Report Instance Assignments
- Report Clocks

BluePrint Platform Designer Summary

Quartus Prime Version	Version 16.0.0 Build 206 03
Project	
Project name	ddr3_x72_2inst_nf
Revision	ddr3_x72_2inst_nf
Current state	Project assignments applied
Design	
IP blocks	
Arria 10 External Memory Interfaces	synth_1|emif_0_example_de
Arria 10 External Memory Interfaces	synth_0|emif_0_example_de
Total top-level periphery cells	
Placed	16 (47.1%)
Unplaced	18 (52.9%)
Total periphery cells	
Placed	602 (48.8%)
Unplaced	632 (51.2%)

```
periph::report_summary
periph::report_summary
periph::report_summary
periph::report_summary
```

Placement History
- Apply Constraints
- Autoplace cells: 2476
- Set GCLK clock type 969

Click items to generate reports

Related Information

- Reports Tab Controls on page 34
- Interface Planner Reports on page 35

2.2.6. Step 6: Validate and Export Plan Constraints

You must validate your interface plan before exporting the plan constraints to your project as a generated Tcl script. Validation must confirm that the Fitter can place all remaining unplaced design elements. When you are satisfied with your interface plan, follow these steps to validate and apply the interface plan to your Intel Quartus Prime project:

1. In the **Flow** control, click **Validate Plan**. The Fitter confirms placement of all remaining unplaced design elements. You must correct any errors before you can export constraints.
2. After validation, click **Export Constraints** to generate a Tcl script that applies the plan to your project. The output Tcl file contains instructions to apply the interface plan to your Intel Quartus Prime project. Close Interface Planner and open the Intel Quartus Prime software.
3. To apply the exported interface plan constraints to your Intel Quartus Prime project, click **Tools ➤ Tcl Scripts** and select the `<`*`project name`*`>.pdp_assignments.tcl` script file.
4. Click **Run**. The script runs, applying the Interface Planner constraints to the project.
   Alternatively, you can run the script from the project directory:

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

5. In the Intel Quartus Prime software, click **Start ➤ Start Analysis & Synthesis** to synthesize and apply the interface plan in your project.

2.3. Interface Planner User Interface Controls

The Interface Planner user interface includes the following controls for planning your design platform.

Flow Controls on page 31

Home Tab Controls on page 32

Assignments Tab Controls on page 32

Plan Tab Controls on page 33

Reports Tab Controls on page 34

2.3.1. Flow Controls

The **Flow** control panel provides immediate access to common Interface Planner commands from anywhere within Interface Planner. The **Flow** controls appear in order of a typical interface planning flow.

Table 4. Flow Controls

Command	Description
Open Project	Allows you to select and open an Intel Quartus Prime project in Interface Planner. Use of **Open Project** command is only required when using Interface Planner in standalone mode.
Initialize Interface Planner	Loads the synthesis netlist, starts the Fitter verification engine, and imports assignments from your Intel Quartus Prime project.
View Assignments	Opens the **Assignments** tab, which allows you to review and reconcile any conflicting assignments that Interface Planner imports from your project. Enable or disable specific project assignments to resolve any conflicts.
Update Plan	Applies the enabled project assignments to your interface plan. You cannot perform periphery planning on the **Plan** tab until you update the plan.
Plan Design	Opens the **Plan** tab for placing logic in the interface plan.
Export Constraints	Saves your interface plan as a Tcl script file for subsequent application in your project. This command is available only after successfully running **Validate Plan**.
Validate plan	Verifies that all constraints in the interface plan are compatible with placement of all remaining unplaced design elements. You can then directly locate and resolve the source of any reported validation errors. You must successfully validate the plan before running **Write Plan File**.
View Reports	Opens the **Reports** tab for filtering data and finding entities and locations.

2.3.2. Home Tab Controls

The Interface Planner **Home** tab contains controls for opening projects in Interface Planner. You only need the **Home** tab when Interface Planner is in standalone mode.

Table 5. Home Tab Controls

Command	Description
Recent Projects	Provides quick access to recently opened Intel Quartus Prime projects. A named tile represents each project. Click the tile to display **Details** about the project. Double-click the tile to open the project in Interface Planner.
Browse	Allows you to locate and open an Intel Quartus Prime project in Interface Planner. Interface Planner requires the project's synthesized netlist for operation.
Details	Provides project and file details such as the file path, revision, and creation date of the Intel Quartus Prime project. You can select a specific project revision.

2.3.3. Assignments Tab Controls

The **Assignments** tab contains controls for resolving potential conflicts with project assignments. Click **View Assignments** to display the **Assignments** tab.

You can enable or disable specific or classes of assignments until you resolve all potential conflicts. After you are satisfied with the status of all project assignments, click **Update Plan** to update your interface plan with the enabled project assignments. Interface Planner reports an error for any remaining assignment conflicts.

Table 6. Assignments Tab Controls

Command	Description
Filter field	Supports creation of wildcard expressions for assignment targets. **Enabled** and **Disabled** buttons filter only enabled or disabled assignments in the list.
Enable All Project Assignments	Enables all imported project assignments in your interface plan.
Disable All Project Assignments	Disables all imported project assignments in the plan.
Clear	Clears any filter from the **Assignments** list.

2.3.4. Plan Tab Controls

The **Plan** tab contains the following controls to help you place logic in the interface plan. Click **Plan Design** to display the **Plan** tab.

Placement or unplacement in the interface plan does not apply to your Intel Quartus Prime project until you add the generated Interface Planner constraints script to your project.

Table 7. Plan Tab Controls

Command	Description
>>	Lists legal locations for placement.
Locate Node	Display a list of Intel Quartus Prime Pro Edition tools where the selected design element is referrenced in the hierarchical database. If the **Locate Node** command is disabled for a specific element in the **Design Elements** list, it is because that element is not represented as an element in the design.
Autoplace All	Attempts to place all unplaced design elements in legal locations in the interface plan.
Autoplace Fixed	Attempts to place all unplaced design elements that have only one legal location into the interface plan.
Unplace All	Unplaces all placed design elements in the interface plan.
Right-click ➤ Auto-place selected element	Attempts to place the selected design element and all its children in a legal location in the interface plan.
Chip View	Displays the target device chip. Zoom in to display chip details.
Package View	Displays the target device package. Zoom in to display chip details.
Right-click ➤ Report Placeability of Selected Element	Displays detailed information on the **Reports** tab, showing legal locations in the interface plan for the selected cell in order of suitability for fitting.
Copy Current View	Copies the current interface plan to the clipboard for pasting into other files, such as word processing or presentation files.
Reset Plan	Unplaces all placed design elements and removes applied project assignments from the interface plan. Resets all project assignments to the enabled state. You must subsequently run **Update Plan** prior to placing design elements. This command only applies to your interface plan and does not impact your Intel Quartus Prime project assignments until you apply the Interface Planner script.

continued...

Command	Description
Load Floorplan	Allows you to select and load an Interface Planner Floorplan Format (`.plan`) file. You can save Interface Planner floorplan files in the format by clicking **Save Floorplan**.
Save Floorplan	Allows you to save your Interface Planner floorplan as a `.plan` file.
Undo/Redo buttons	The **Undo** button reverts the last change made in the **Plan** tab. **Redo** re-implements the last undo. Use these commands to step forward and backward though your plan changes.

2.3.5. Reports Tab Controls

The Interface Planner **Reports** tab contains the following **Task** pane controls to help you filter data and find entities and locations.

Table 8. Reports Tab Controls

Command	Description
Create all Summary Reports	Creates the following summary reports: • **Interface Planner Summary** • **All Periphery Cells** • **Placed/Unplaced Periphery Cells** • **Periphery Location Types**.
Report All Placed/Unplaced Pins	Reports the name, parent (if any), and type of all placed (**Report All Placed Pins**) or unplaced (**Report All Unplaced Pins**) pins in the interface plan, respectively. The Placed Pins report includes the placement location name. The Unplaced Pins report includes the number of potential placement locations. Right-click any cell to place, unplace, or report connectivity or location information.
Report All Placed/Unplaced HSSI Channels	Reports the name, parent (if any), and type of all placed (**Report All Placed HSSI Channels**) or unplaced (**Report All Unplaced HSSI Channels**) channels in the interface plan, respectively. The Placed HSSI Channels report includes the placement location name. The Unplaced HSSI Channels report includes the number of potential placement locations. Right-click any cell to place, unplace, or report connectivity or location information.
Right-click ➤ Report Placed/Unplaced Periphery Cells of Selected Type	Reports the name, parent (if any), and type of placed (**Report Placed Periphery Cells of Selected Type**) or unplaced (**Report Unplaced Periphery Cells of Selected Type**) cells matching the selected type. The placed cells report includes the placement location name. The unplaced cells report includes the number of potential placement locations. Right-click any cell to place, unplace, or report connectivity or location information.
Right-click ➤ Report Periphery Locations of Selected Type	Reports all locations in the device of the selected type, and whether the location supports merging.
Right-click ➤ Report Periphery Cell Connectivity	Reports the source port and type, destination port and type, of connections to the selected cell. Right-click any cell to report the individual cell connectivity.
Right-click ➤ Place/Unplace Cell	Places the cell in the selected location of the interface plan. Similarly, you can right-click any cell and then click **Place Cell of Selected Type** or **Unplace Cell of Selected Type** to place or unplace multiple cells of the same type.
Right-click ➤ Report Cell Locations for Custom Placement	Reports the preferred legal locations for the selected cell in the interface plan in the Legal Location report. Right-click to immediately place the cell in a location or report all periphery location of the selected type.
Remove Invalid Reports	Removes outdated Interface Planner reports that you invalidate by changes to the interface plan.
Report Instance Assignments	Shows all imported project assignments in the interface plan. You can delete these assignments from the plan.

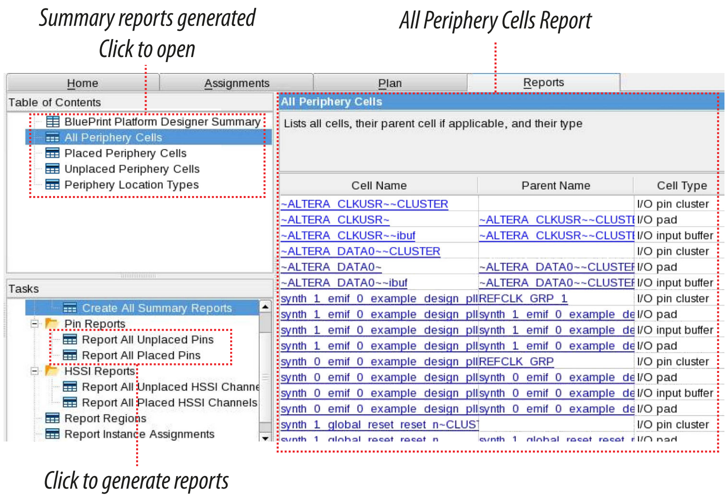
2.4. Interface Planner Reports

Use Interface Planner reports to locate cells and assign suitable placement locations for specific interfaces and elements in your design. Interface Planner reports provide detailed, actionable feedback to help you quickly implement the best plan for your design. You can access placement and further reporting functions directly from Interface Planner reports. Interface Planner generates the following reports that provide detailed planning information:

2.4.1. Report Summary

Click **Create all Summary Reports** on the **Reports** tab to generate summary reports about periphery cells in the interface plan. Right-click any cell type to report placed, unplaced, connectivity, or location information.

Figure 18. Summary Reports



All Periphery Cells

Lists all cells, their parent cell if applicable, and their type

Cell Name	Parent Name	Cell Type
~ALTERA_CLKUSR~~CLUSTER		I/O pin cluster
~ALTERA_CLKUSR~	~ALTERA_CLKUSR~~CLUSTE	I/O pad
~ALTERA_CLKUSR~~ibuf	~ALTERA_CLKUSR~~CLUSTE	I/O input buffer
~ALTERA_DATA0~~CLUSTER		I/O pin cluster
~ALTERA_DATA0~	~ALTERA_DATA0~~CLUSTEF	I/O pad
~ALTERA_DATA0~~ibuf	~ALTERA_DATA0~~CLUSTEF	I/O input buffer
synth_1_emif_0_example_design_pll	REFCLK_GRP_1	I/O pin cluster
synth_1_emif_0_example_design_pll	synth_1_emif_0_example_de	I/O pad
synth_1_emif_0_example_design_pll	synth_1_emif_0_example_de	I/O input buffer
synth_1_emif_0_example_design_pll	synth_1_emif_0_example_de	I/O pad
synth_0_emif_0_example_design_pll	REFCLK_GRP	I/O pin cluster
synth_0_emif_0_example_design_pll	synth_0_emif_0_example_de	I/O pad
synth_0_emif_0_example_design_pll	synth_0_emif_0_example_de	I/O input buffer
synth_0_emif_0_example_design_pll	synth_0_emif_0_example_de	I/O pad
synth_1_global_reset_reset_n~CLUS		I/O pin cluster
synth_1_global_reset_reset_n	synth_1_global_reset_reset_r	I/O pad

Table 9. Report Summary

Command	Description
Create all Summary Reports	Creates the following summary reports:

Command	Description
	• **Interface Planner Summary**—reports software version and total number of periphery and top-level periphery cells. • **All Periphery Cells**— reports the name, parent, and type of all periphery cells in the design. • **Placed/Unplaced Periphery Cells**—reports the name, parent, and type of all placed and unplaced periphery cells in the interface plan. • **Periphery Location Types**—reports the number of each type of periphery location available in the target device and the number required by your design.

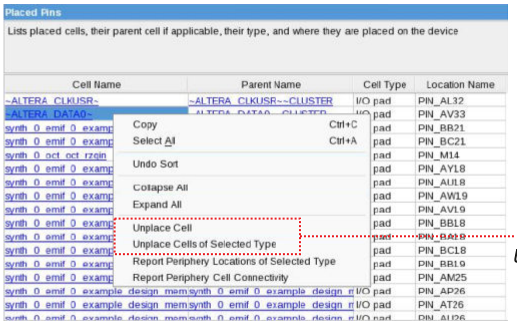
2.4.2. Report Pins

Generate reports about I/O pins in the design. Right-click any cell type to place, unplace, or report connectivity or location information.

Table 10. Report Pin Commands

Command	Description
Report All Placed Pins	Generates the Placed Pins report. This report lists the name, parent, type, and location of all placed pins in the interface plan.
Report All Unplaced Pins	Generates the Unplaced Pins report. This report lists the name, parent, type, and the number of potential placements for all unplaced pins in the interface plan.

Figure 19. Placed Pins Report



Placed Pins

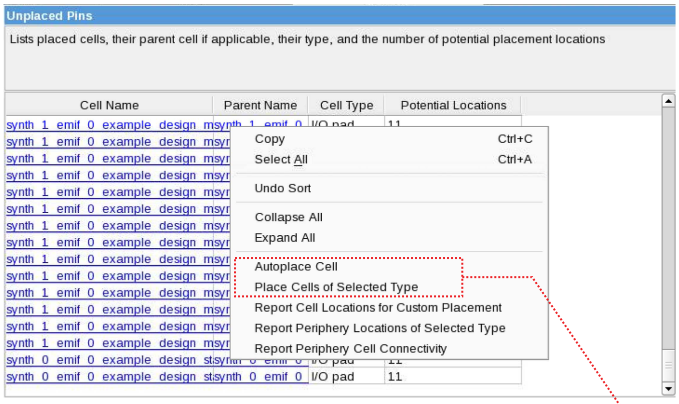
Lists placed cells, their parent cell if applicable, their type, and where they are placed on the device

Cell Name	Parent Name	Cell Type	Location Name
~ALTERA_CLKUSR~	~ALTERA_CLKUSR~~CLUSTER	I/O pad	PIN_AL32
~ALTERA_DATA0~	~ALTERA_DATA0~~CLUSTER	I/O pad	PIN_AV33
synth_0_emif_0_examp		pad	PIN_BB21
synth_0_emif_0_examp		pad	PIN_BC21
synth_0_oct_oct_rzqin		pad	PIN_M14
synth_0_emif_0_examp		pad	PIN_AY18
synth_0_emif_0_examp		pad	PIN_AU18
synth_0_emif_0_examp		pad	PIN_AW19
synth_0_emif_0_examp		pad	PIN_AV19
synth_0_emif_0_examp		pad	PIN_BB18
synth_0_emif_0_examp		pad	PIN_BA18
synth_0_emif_0_examp		pad	PIN_BC18
synth_0_emif_0_examp		pad	PIN_BB19
synth_0_emif_0_examp		pad	PIN_AM25
synth_0_emif_0_example_design_mem	synth_0_emif_0_example_design_m	I/O pad	PIN_AP26
synth_0_emif_0_example_design_mem	synth_0_emif_0_example_design_m	I/O pad	PIN_AT26
synth_0_emif_0_example_design_mem	synth_0_emif_0_example_design_m	I/O pad	PIN_AU26

Copy Ctrl+C
Select All Ctrl+A
Undo Sort
Collapse All
Expand All
Unplace Cell
Unplace Cells of Selected Type
Report Periphery Locations of Selected Type
Report Periphery Cell Connectivity

Right-click to unplace placed pins

Figure 20. Unplaced Pins Report



Unplaced Pins

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
synth_1_emif_0_example_design_m	synth_1_emif_0	I/O pad	11
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_1_emif_0_example_design_m	sy		
synth_0_emif_0_example_design_st	synth_0_emif_0	I/O pad	11
synth_0_emif_0_example_design_st	synth_0_emif_0	I/O pad	11

Copy Ctrl+C
Select All Ctrl+A
Undo Sort
Collapse All
Expand All
Autoplace Cell
Place Cells of Selected Type
Report Cell Locations for Custom Placement
Report Periphery Locations of Selected Type
Report Periphery Cell Connectivity

Right-click to place unplaced pins

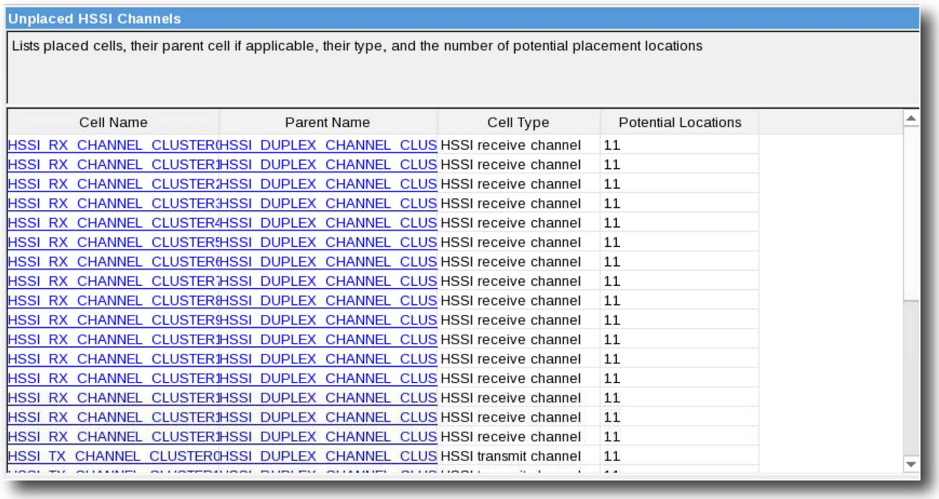
2.4.3. Report HSSI Channels

Generate reports about HSSI channels in the interface plan. Right-click any cell type to place, unplace, or report connectivity or location information.

Table 11. Report Channel Commands

Command	Description
Report All Placed HSSI Channels	Generates the Placed HSSI Channels report. This report lists the name, parent, type, and location of all placed HSSI RX/TX channels in the interface plan.
Report All Unplaced HSSI Channels	Generates the Unplaced HSSI Channels report. This report lists the name, parent, type, and location of all unplaced HSSI RX/TX channels in the interface plan.

Figure 21. Unplaced HSSI Channels Report



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

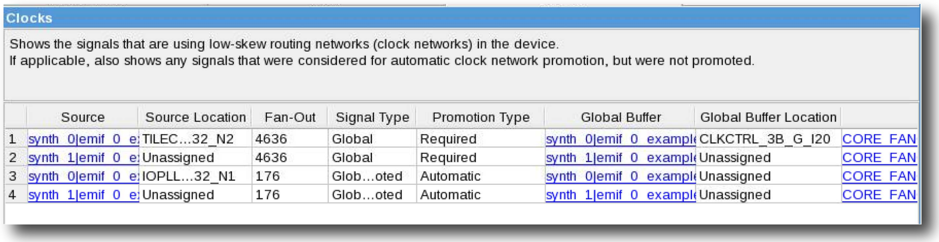
2.4.4. Report Clocks

Generate reports showing clock networks in the plan. Use this report to analyze clock network scenarios and ensure that specific device regions are fed by high fan-out signals.

Table 12. Report Clocks Commands

Command	Description
Report Clocks	Generates the Global and other Fast Signals report.

Figure 22. Clocks Report



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.

	Source	Source Location	Fan-Out	Signal Type	Promotion Type	Global Buffer	Global Buffer Location	
1	synth_0|emif_0_e:	TILEC...32_N2	4636	Global	Required	synth_0|emif_0_exampl	CLKCTRL_3B_G_I20	CORE_FAN
2	synth_1|emif_0_e:	Unassigned	4636	Global	Required	synth_1|emif_0_exampl	Unassigned	CORE_FAN
3	synth_0|emif_0_e:	IOPLL...32_N1	176	Glob...oted	Automatic	synth_0|emif_0_exampl	Unassigned	CORE_FAN
4	synth_1|emif_0_e:	Unassigned	176	Glob...oted	Automatic	synth_1|emif_0_exampl	Unassigned	CORE_FAN

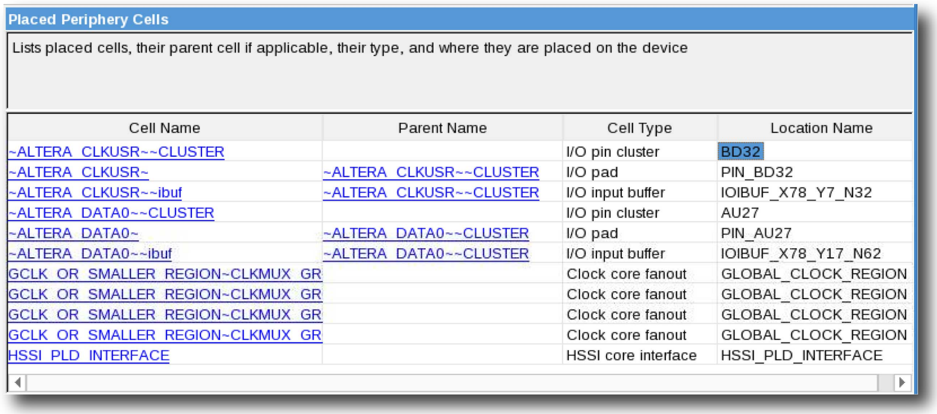
2.4.5. Report Periphery Locations

Generate reports that show the status of periphery cells in the interface plan.

Table 13. Report Periphery Locations Commands

Command	Description
Right-click ➤ Report Placed Periphery Cells of Selected Type	Accessible from the **All Periphery Cells** report. This command reports the name, parent (if any), type, and location of the selected placed periphery cells matching the selected type. Right-click any cell to place, unplace, or report connectivity or location information.
Right-click ➤ Report Unplaced Periphery Cells of Selected Type	Accessible from the **All Periphery Cells** report. This command reports the name, parent (if any), type, and number of suitable locations for the selected unplaced periphery cells matching the selected type. Right-click any cell to place, unplace, or report connectivity or location information.
Right-click ➤ Report Periphery Locations of Selected Type	Reports all locations in the device of the selected type, and whether the location supports merging.

Figure 23. Periphery Locations Report



Placed Periphery Cells

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

Cell Name	Parent Name	Cell Type	Location Name
~ALTERA_CLKUSR~~CLUSTER		I/O pin cluster	BD32
~ALTERA_CLKUSR~	~ALTERA_CLKUSR~~CLUSTER	I/O pad	PIN_BD32
~ALTERA_CLKUSR~~ibuf	~ALTERA_CLKUSR~~CLUSTER	I/O input buffer	IOIBUF_X78_Y7_N32
~ALTERA_DATA0~~CLUSTER		I/O pin cluster	AU27
~ALTERA_DATA0~	~ALTERA_DATA0~~CLUSTER	I/O pad	PIN_AU27
~ALTERA_DATA0~~ibuf	~ALTERA_DATA0~~CLUSTER	I/O input buffer	IOIBUF_X78_Y17_N62
GCLK_OR_SMALLER_REGION~CLKMUX_GR		Clock core fanout	GLOBAL_CLOCK_REGION
GCLK_OR_SMALLER_REGION~CLKMUX_GR		Clock core fanout	GLOBAL_CLOCK_REGION
GCLK_OR_SMALLER_REGION~CLKMUX_GR		Clock core fanout	GLOBAL_CLOCK_REGION
GCLK_OR_SMALLER_REGION~CLKMUX_GR		Clock core fanout	GLOBAL_CLOCK_REGION
HSSI_PLD_INTERFACE		HSSI core interface	HSSI_PLD_INTERFACE

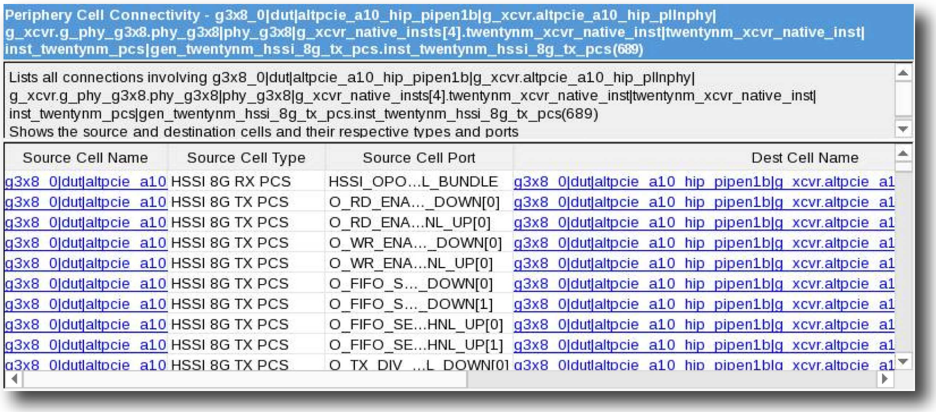
2.4.6. Report Cell Connectivity

Generate reports showing the connections between all cells in the interface plan.

Table 14. Report Cell Connectivity Command

Command	Description
Right-click ➤ Report Periphery Cell Connectivity	Right-click any **Cell Name** in the reports to **Report Periphery Cell Connectivity**. The report lists the source and destination ports and type of connections to the selected cell. Right-click any cell to report all connections to the cell.

Figure 24. Cell Connectivity Report



Periphery Cell Connectivity - g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a10_hip_pllnphy|g_xcvr.g_phy_g3x8.phy_g3x8|phy_g3x8|g_xcvr_native_insts[4].twentynm_xcvr_native_inst|twentynm_xcvr_native_inst|inst_twentynm_pcs|gen_twentynm_hssi_8g_tx_pcs.inst_twentynm_hssi_8g_tx_pcs(689)

Lists all connections involving g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a10_hip_pllnphy|g_xcvr.g_phy_g3x8.phy_g3x8|phy_g3x8|g_xcvr_native_insts[4].twentynm_xcvr_native_inst|twentynm_xcvr_native_inst|inst_twentynm_pcs|gen_twentynm_hssi_8g_tx_pcs.inst_twentynm_hssi_8g_tx_pcs(689)
Shows the source and destination cells and their respective types and ports

Source Cell Name	Source Cell Type	Source Cell Port	Dest Cell Name
g3x8_0|dut|altpcie_a10	HSSI 8G RX PCS	HSSI_OPO...L_BUNDLE	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_RD_ENA..._DOWN[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_RD_ENA...NL_UP[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_WR_ENA..._DOWN[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_WR_ENA...NL_UP[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_FIFO_S..._DOWN[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_FIFO_S..._DOWN[1]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_FIFO_SE...HNL_UP[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_FIFO_SE...HNL_UP[1]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1
g3x8_0|dut|altpcie_a10	HSSI 8G TX PCS	O_TX_DIV_...L_DOWN[0]	g3x8_0|dut|altpcie_a10_hip_pipen1b|g_xcvr.altpcie_a1

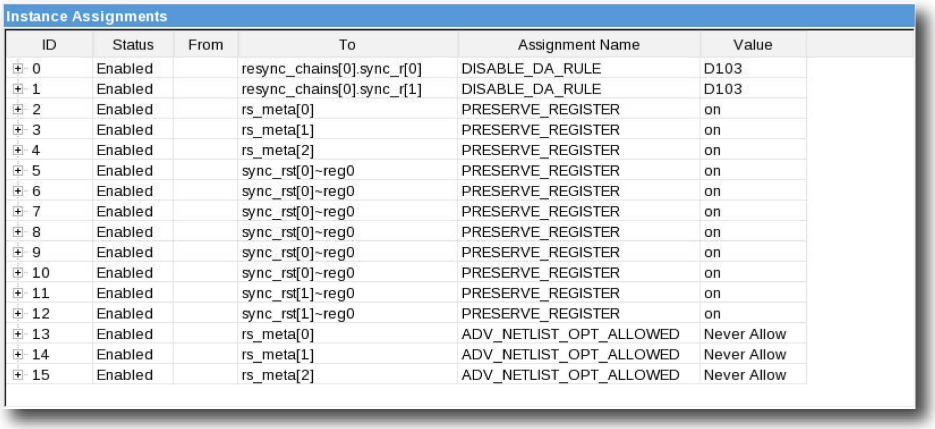
2.4.7. Report Instance Assignments

Click **Report Instance Assignments** to show all imported project assignments in the interface plan. You can delete these assignments from the plan.

Table 15. Report Instance Assignments Command

Command	Description
Report Instance Assignments	Reports all enabled instance assignments in your design. Right-click any cell to delete the assignment or to delete all assignments of the same type.

Figure 25. Instance Assignments Report



Instance Assignments

ID	Status	From	To	Assignment Name	Value
⊞ 0	Enabled		resync_chains[0].sync_r[0]	DISABLE_DA_RULE	D103
⊞ 1	Enabled		resync_chains[0].sync_r[1]	DISABLE_DA_RULE	D103
⊞ 2	Enabled		rs_meta[0]	PRESERVE_REGISTER	on
⊞ 3	Enabled		rs_meta[1]	PRESERVE_REGISTER	on
⊞ 4	Enabled		rs_meta[2]	PRESERVE_REGISTER	on
⊞ 5	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 6	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 7	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 8	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 9	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 10	Enabled		sync_rst[0]~reg0	PRESERVE_REGISTER	on
⊞ 11	Enabled		sync_rst[1]~reg0	PRESERVE_REGISTER	on
⊞ 12	Enabled		sync_rst[1]~reg0	PRESERVE_REGISTER	on
⊞ 13	Enabled		rs_meta[0]	ADV_NETLIST_OPT_ALLOWED	Never Allow
⊞ 14	Enabled		rs_meta[1]	ADV_NETLIST_OPT_ALLOWED	Never Allow
⊞ 15	Enabled		rs_meta[2]	ADV_NETLIST_OPT_ALLOWED	Never Allow

2.5. Interface Planner Revision History

This document has the following revision history:

Document Version	Intel Quartus Prime Version	Changes
2018.05.07	18.0.0	• Initial release in Design Constraints User Guide: Intel Quartus Prime Pro Edition. • Updated *Step 2: Initialze Interface Planner* to remove the requirement to close Intel Quartus Prime. • Updated *Step 4: Plan Periphery Placement* to describe when the **Locate Node** command is disabled.
2017.11.06	17.1.0	• Removed support for the **Clocking** feature for Intel Stratix 10. Intel Stratix 10 clocks must use **Autoplace Selected**. • Renamed BluePrint to Interface Planner. • Renamed chapter from *BluePrint Design Planning* to *Interface Planning.*
2016.10.31	16.1.0	• Implemented Intel rebranding.
2016.05.03	16.0.0	• Added Plan Clock Networks topic. • Added Saving and Loading Floorplans topic. • Added Undo/Redo command descriptions. • Added Flow control description. • Added note about panning feature. • Updated all screenshots for latest GUI.
2015.11.02	15.1.0	• Changed instances of *Quartus II* to *Quartus Prime*. • Integration of content into Quartus Prime Handbook. • Added descriptions of new dynamic reports. • Added Package View description. • Added GUI controls reference.
2015.05.04	15.0.0	Second beta release of document on Molson. Added information about the following subjects: • Overview information • Reset Plan command • Legal Assignments list and prompt • Tcl console
2014.12.15	14.1.	First beta release of document on Molson.

Related Information

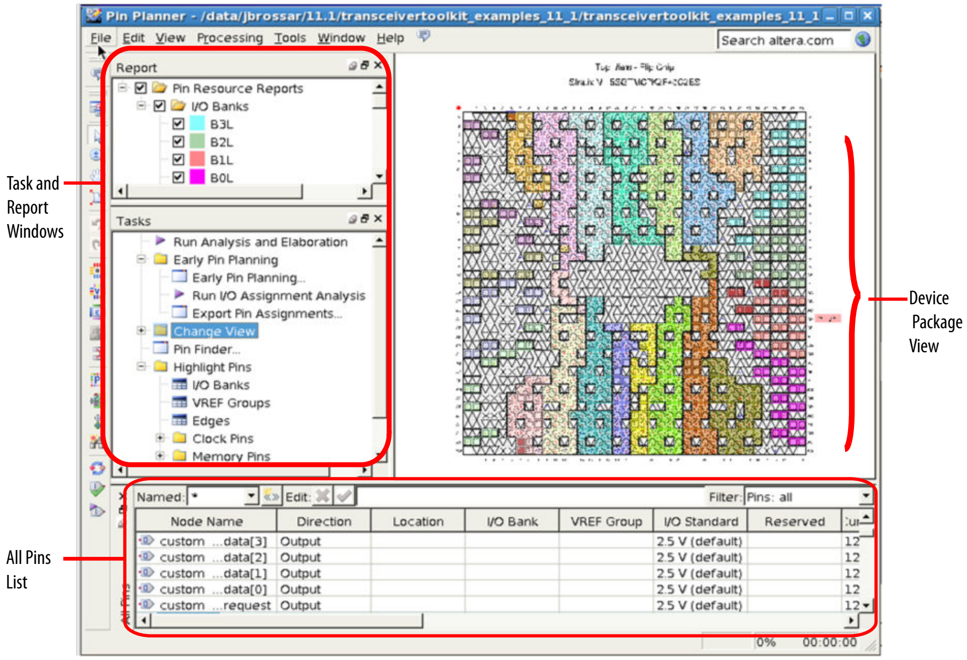
Altera Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the Altera documentation archives.

3. Managing Device I/O Pins

This chapter describes efficient planning and assignment of I/O pins in your target device. Consider I/O standards, pin placement rules, and your PCB characteristics early in the design phase.

Figure 26. Pin Planner GUI



Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	:ur
custom ...data[3]	Output				2.5 V (default)		12
custom ...data[2]	Output				2.5 V (default)		12
custom ...data[1]	Output				2.5 V (default)		12
custom ...data[0]	Output				2.5 V (default)		12
custom ...request	Output				2.5 V (default)		12

Table 16. Intel Quartus Prime I/O Pin Planning Tools

I/O Planning Task	Click to Access
Plan interfaces and device periphery	**Tools ➤ Interface Planner**
Edit, validate, or export pin assignments	**Assignments ➤ Pin Planner**

For more information about special pin assignment features for the Intel Arria 10 SoC devices, refer to *Instantiating the HPS Component* in the *Intel Arria 10 Hard Processor System Technical Reference Manual*.

Related Information

- Interface Planning Overview on page 20
- Instantiating the HPS Component
  In *Intel Arria 10 Hard Processor System Technical Reference Manual*

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2015 Registered

3.1. I/O Planning Overview

On FPGA design, I/O planning includes creating pin-related assignments and validating them against pin placement guidelines. This process ensures a successful fit in your target device. When you plan and assign I/O pins in the initial stages of your project, you design for compatibility with your target device and PCB characteristics. As a result, your design process goes through fewer iterations, and you develop an accurate PCB layout sooner.

You can plan your I/O pins even before defining design files. Assign expected nodes not yet defined in design files, including interface IP core signals, and then generate a top-level file. The top-level file instantiates the next level of design hierarchy and includes interface port information like memory, high-speed I/O, device configuration, and debugging tools.

Assign design elements, I/O standards, interface IP, and other properties to the device I/O pins by name or by dragging to cells. You can then generate a top-level design file for I/O validation.

Use I/O assignment validation to fully analyze I/O pins against VCCIO, VREF, electromigration (current density), Simultaneous Switching Output (SSO), drive strength, I/O standard, PCI_IO clamp diode, and I/O pin direction compatibility rules.

Intel Quartus Prime software provides the Pin Planner tool to view, assign, and validate device I/O pin logic and properties. Alternatively, you can enter I/O assignments in a Tcl script, or directly in HDL code.

3.1.1. Basic I/O Planning Flow

The following steps describe the basic flow for assigning and verifying I/O pin assignments:

1. Click **Assignments ➤ Device** and select a target device that meets your logic, performance, and I/O requirements. Consider and specify I/O standards, voltage and power supply requirements, and available I/O pins.
2. Click **Assignments ➤ Pin Planner**.
3. Assign I/O properties to match your device and PCB characteristics, including assigning logic, I/O standards, output loading, slew rate, and current strength.
4. Click **Run I/O Assignment Analysis** in the **Tasks** pane to validate assignments and generate a synthesized design netlist. Correct any problems reported.
5. Click **Processing ➤ Start Compilation**. During compilation, the Intel Quartus Prime software runs I/O assignment analysis.

3.1.2. Integrating PCB Design Tools

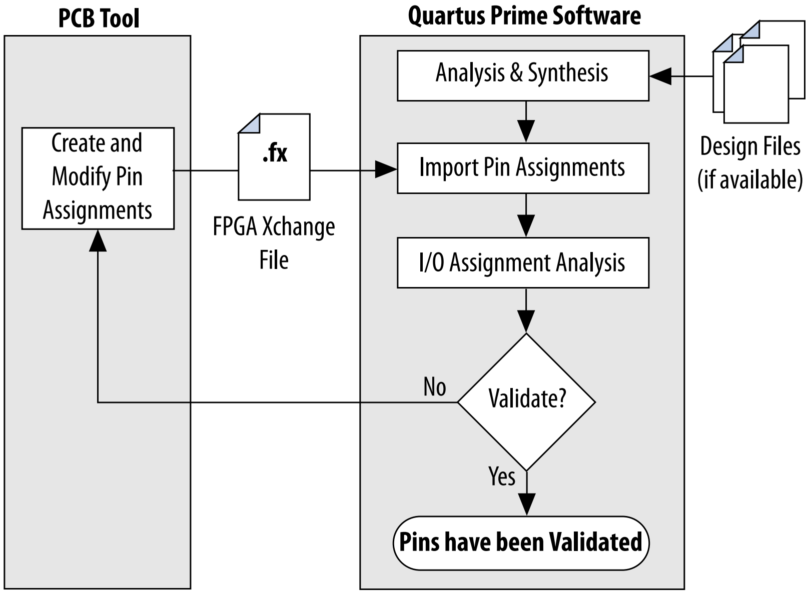
You can integrate PCB design tools into your work flow to map pin assignments to symbols in your system circuit schematics and board layout.

The Intel Quartus Prime software integrates with board layout tools by allowing import and export of pin assignment information in Intel Quartus Prime Settings Files (`.qsf`), Pin-Out File (`.pin`), and FPGA Xchange-Format File (`.fx`) files.

Table 17. Integrating PCB Design Tools

PCB Tool Integration	Supported PCB Tool
Define and validate I/O assignments in the Pin Planner, and then export the assignments to the PCB tool for validation	Mentor Graphics* I/O Designer Cadence Allegro
Define I/O assignments in your PCB tool, and then import the assignments into the Pin Planner for validation	Mentor Graphics I/O Designer Cadence Allegro

Figure 27. PCB Tool Integration

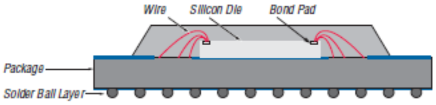
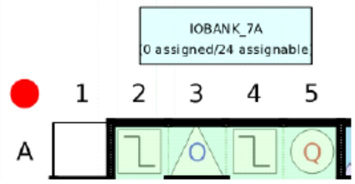
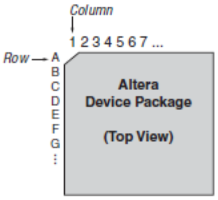
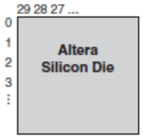
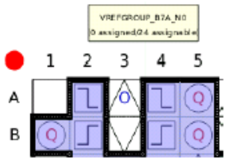


Related Information

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

3.1.3. Intel Device Terms

The following terms describe Intel device and I/O structures:

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	 Wire Silicon Die Bond Pad Package Solder Ball Layer
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.	 IOBANK_7A 0 assigned/24 assignable 1 2 3 4 5 A O Q
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.	 Column 1 2 3 4 5 6 7 ... Row A B C D E F G Altera Device Package (Top View)
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..	 VREFGROUP_B7A_N0 0 assigned/24 assignable 1 2 3 4 5 A B O Q

3.2. Assigning I/O Pins

Use the Pin Planner to visualize, modify, and validate I/O assignments in a graphical representation of the target device. You can increase the accuracy of I/O assignment analysis by reserving specific device pins to accommodate undefined but expected I/O.

To assign I/O pins in the Pin Planner, follow these steps:

1. Open an Intel Quartus Prime project, and then click **Assignments ➤ Pin Planner**.
2. Click **Processing ➤ Start Analysis & Elaboration** to elaborate the design and display **All Pins** in the device view.
3. To locate or highlight pins for assignment, click **Pin Finder** or a pin type under **Highlight Pins** in the **Tasks** pane.
4. (Optional) To define a custom group of nodes for assignment, select one or more nodes in the **Groups** or **All Pins** list, and click **Create Group**.
5. Enter assignments of logic, I/O standards, interface IP, and properties for device I/O pins in the **All Pins** spreadsheet, or by dragging into the package view.
6. To assign properties to differential pin pairs, click **Show Differential Pin Pair Connections**. A red connection line appears between positive (p) and negative (n) differential pins.
7. (Optional) To create board trace model assignments:
   a. Right-click an output or bidirectional pin, and click **Board Trace Model**. For differential I/O standards, the board trace model uses a differential pin pair with two symmetrical board trace models.
   b. Specify board trace parameters on the positive end of the differential pin pair. The assignment applies to the corresponding value on the negative end of the differential pin pair.
8. To run a full I/O assignment analysis, click **Run I/O Assignment Analysis**. The Fitter reports analysis results. Only reserved pins are analyzed prior to design synthesis.

3.2.1. Assigning to Exclusive Pin Groups

You can designate groups of pins for exclusive assignment. When you assign pins to an **Exclusive I/O Group**, the Fitter does not place the signals in the same I/O bank with any other exclusive I/O group. For example, if you have a set of signals assigned exclusively to `group_a`, and another set of signals assigned to `group_b`, the Fitter ensures placement of each group in different I/O banks.

3.2.2. Assigning Slew Rate and Drive Strength

You can designate the device pin slew rate and drive strength. These properties affect the pin's outgoing signal integrity. Use either the **Slew Rate** or **Slow Slew Rate** assignment to adjust the drive strength of a pin with the **Current Strength** assignment.

Note: The slew rate and drive strength apply during I/O assignment analysis.

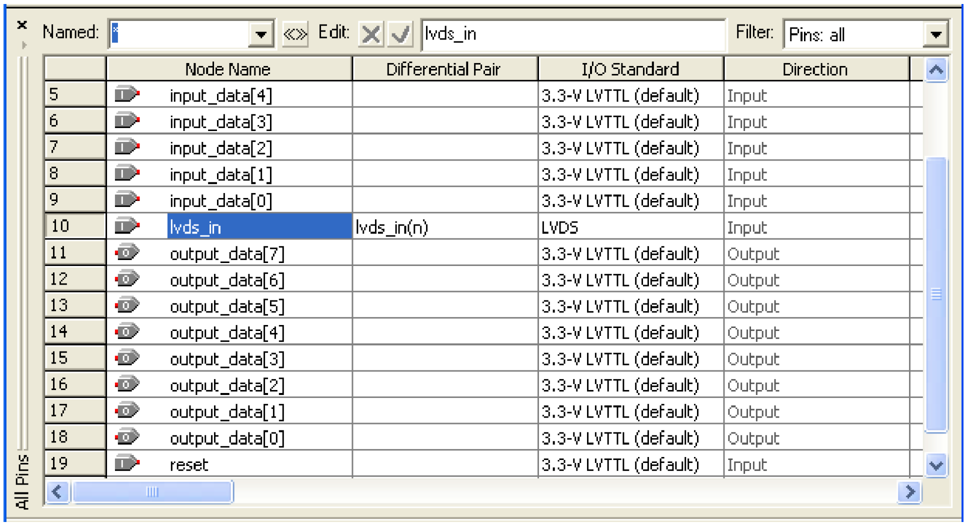
3.2.3. Assigning Differential Pins

When you assign a differential I/O standard to a single-ended top-level pin in your design, the Pin Planner automatically recognizes the negative pin as part of the differential pin pair assignment and creates the negative pin for you. The Intel Quartus Prime software writes the location assignment for the negative pin to the `.qsf`; however, the I/O standard assignment is not added to the `.qsf` for the negative pin of the differential pair.

The following example shows a design with `lvds_in` top-level pin, to which you assign a differential I/O standard. The Pin Planner automatically creates the differential pin, `lvds_in(n)` to complete the differential pin pair.

Note: If you have a single-ended clock that feeds a PLL, assign the pin only to the positive clock pin of a differential pair in the target device. Single-ended pins that feed a PLL and are assigned to the negative clock pin device cause the design to not fit.

Figure 28. Creating a Differential Pin Pair in the Pin Planner

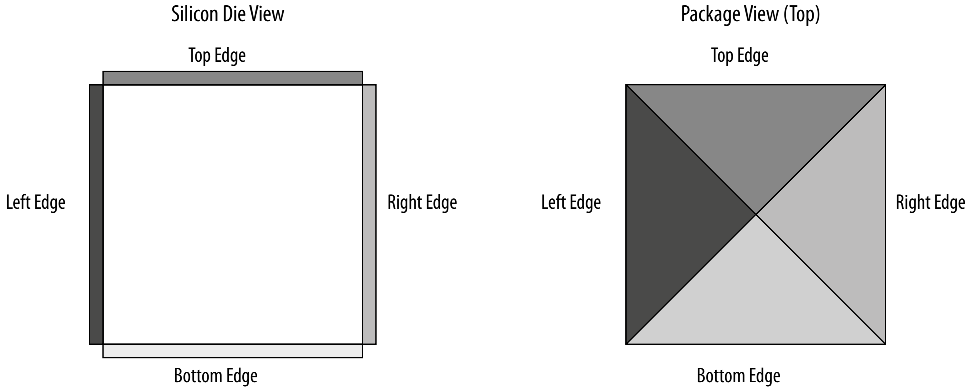


Named: * Edit: lvds_in Filter: Pins: all

	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

If your design contains a large bus that exceeds the pins available in a particular I/O bank, you can use edge location assignments to place the bus. Edge location assignments improve the circuit board routing ability of large buses, because they are close together near an edge. The following figure shows Intel device package edges.

Figure 29. Die View and Package View of the Four Edges on an Intel Device



3.2.3.1. Overriding I/O Placement Rules on Differential Pins

I/O placement rules ensure that noisy signals do not corrupt neighboring signals. Each device family has predefined I/O placement rules.

I/O placement rules define, for example, the allowed placement of single-ended I/O with respect to differential pins, or how many output and bidirectional pins you can place within a VREF group when using voltage referenced input standards.

Use the **IO_MAXIMUM_TOGGLE_RATE** assignment to override I/O placement rules on pins, such as system reset pins that do not switch during normal design activity. Setting a value of 0 MHz for this assignment causes the Fitter to recognize the pin at a DC state throughout device operation. The Fitter excludes the assigned pin from placement rule analysis. Do not assign an **IO_MAXIMUM_TOGGLE_RATE** of 0 MHz to any actively switching pin, or your design may not function as you intend.

3.2.4. Entering Pin Assignments with Tcl Commands

You can apply pin assignments with Tcl scripts, by either entering individual Tcl commands in the Tcl Console, or creating a `.tcl` script and the typing the following in the command line:

Example 6. Applying Tcl Script Assignments

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

Example 7. Scripted Pin Assignment

The following example uses `set_location_assignment` and `set_instance_assignment` Tcl commands to assign a pin to a specific location, I/O standard, and drive strength.

```
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]
```

Related Information

Tcl Scripting
In *Intel Quartus Prime Pro Edition User Guide: Scripting*

3.2.5. Entering Pin Assignments in HDL Code

You can use synthesis attributes or low-level I/O primitives to embed I/O pin assignments directly in your HDL code. When you analyze and synthesize the HDL code, the information is converted into the appropriate I/O pin assignments. You can use either of the following methods to specify pin-related assignments with HDL code:

- Assigning synthesis attributes for signal names that are top-level pins
- Using low-level I/O primitives, such as ALT_BUF_IN, to specify input, output, and differential buffers, and for setting parameters or attributes

3.2.5.1. Using Low-Level I/O Primitives

You can alternatively enter I/O pin assignments using low-level I/O primitives. You can assign pin locations, I/O standards, drive strengths, slew rates, and on-chip termination (OCT) value assignments. You can also use low-level differential I/O primitives to define both positive and negative pins of a differential pair in the HDL code for your design.

Primitive-based assignments do not appear in the Pin Planner until after you perform a full compilation and back-annotate pin assignments (**Assignments > Back Annotate Assignments**).

Related Information

Designing with Low Level Primitives User Guide

3.3. Importing and Exporting I/O Pin Assignments

The Intel Quartus Prime software supports transfer of I/O pin assignments across projects, or for analysis in third-party PCB tools. You can import or export I/O pin assignments in the following ways:

Table 18. Importing and Exporting I/O Pin Assignments

	Import Assignments	Export Assignments
Scenario	• From your PCB design tool or spreadsheet into Pin Planner during early pin planning or after optimization in PCB tool • From another Intel Quartus Prime project with common constraints	• From Intel Quartus Prime project for optimization in a PCB design tool • From Intel Quartus Prime project for spreadsheet analysis or use in scripting assignments • From Intel Quartus Prime project for import into another Intel Quartus Prime project with similar constraints
Command	**Assignments ➤ Import Assignments**	**Assignments ➤ Export Assignments**
File formats	`.qsf`, `.esf`, `.acf`, `.csv`, `.txt`,`.sdc`	`.pin`,`.fx`,`.csv`, `.tcl`, `.qsf`
Notes	N/A	Exported `.csv` files retain column and row order and format. Do not modify the row of column headings if importing the `.csv` file

3.3.1. Importing and Exporting for PCB Tools

The Pin Planner supports import and export of assignments with PCB tools. You can export valid assignments as a **.pin** file for analysis in other supported PCB tools. You can also import optimized assignment from supported PCB tools. The **.pin** file contains pin name, number, and detailed properties.

Mentor Graphics I/O Designer requires you to generate and import both an **.fx** and a **.pin** file to transfer assignments. However, the Intel Quartus Prime software requires only the **.fx** to import pin assignments from I/O Designer.

Table 19. Contents of .pin File

File Column Name	Description
Pin Name/Usage	The name of the design pin, or whether the pin is GND or V_{CC} pin
Location	The pin number of the location on the device package
Dir	The direction of the pin
I/O Standard	The name of the I/O standard to which the pin is configured
Voltage	The voltage level that is required to be connected to the pin
I/O Bank	The I/O bank to which the pin belongs
User Assignment	Y or N indicating if the location assignment for the design pin was user assigned (Y) or assigned by the Fitter (N)

Related Information

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

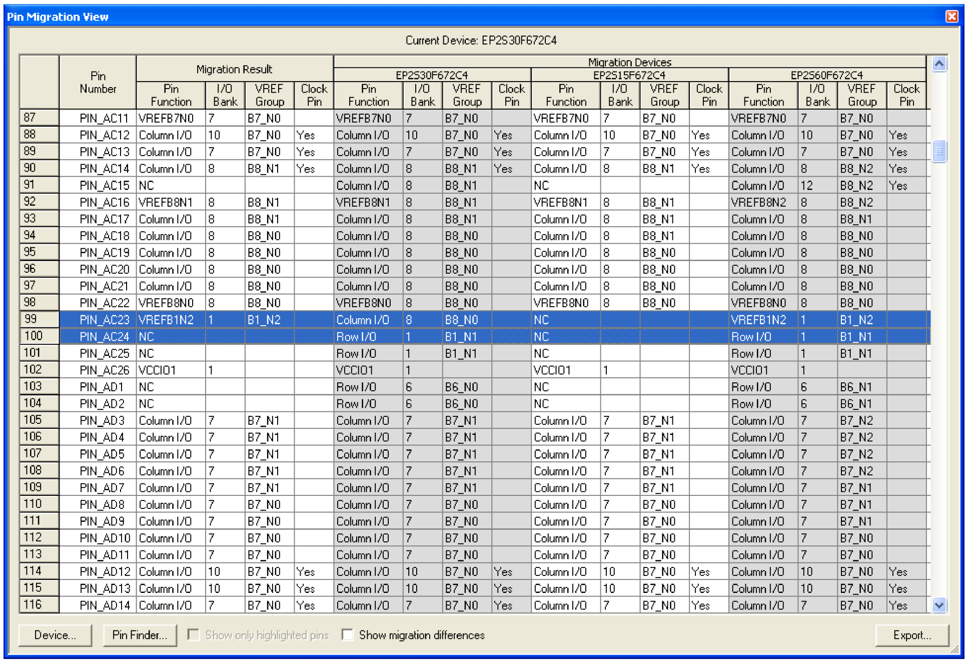
3.3.2. Migrating Assignments to Another Target Device

Click **View ➤ Pin Migration Window** to verify whether pin assignments are compatible with migration to a different Intel device.

You can migrate compatible pin assignments from one target device to another. You can migrate to a different density and the same device package. You can also migrate between device packages with different densities and pin counts.

The Intel Quartus Prime software ignores invalid assignments and generates an error message during compilation. After evaluating migration compatibility, modify any incompatible assignments, and then click **Export** to export the assignments to another project.

Figure 30. Device Migration Compatibility (AC24 does not exist in migration device)



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

The migration result for the pin function of highlighted PIN_AC23 is not an NC but a voltage reference VREFB1N2 even though the pin is an NC in the migration device. VREF standards have a higher priority than an NC, thus the migration result displays the voltage reference. Even if you do not use that pin for a port connection in the design, you must use the VREF standard for I/O standards that require it on the actual board for the migration device.

If one of the migration devices has pins intended for connection to V_{CC} or GND and these same pins are I/O pins on a different device in the migration path, the Intel Quartus Prime software ensures these pins are not used for I/O. Ensure that these pins are connected to the correct PCB plane.

When migrating between two devices in the same package, pins that are not connected to the smaller die may be intended to connect to V_{CC} or GND on the larger die. To facilitate migration, you can connect these pins to V_{CC} or GND in the original design because the pins are not physically connected to the smaller die.

Related Information

AN90: SameFrame PinOut Design for FineLine BGA Packages

3.4. Validating Pin Assignments

The Intel Quartus Prime software validates I/O pin assignments against predefined I/O rules for your target device. You can use the following tools to validate your I/O pin assignments throughout the pin planning process:

Table 20. I/O Validation Tools

I/O Validation Tool	Description	Click to Run
Advanced I/O Timing	Fully validates I/O assignments against all I/O and timing checks during compilation	**Processing ➤ Start Compilation**

3.4.1. I/O Assignment Validation Rules

I/O Assignment Analysis validates your assignments against the following rules:

Table 21. Examples of I/O Rule Checks

Rule	Description	HDL Required?
I/O bank capacity	Checks the number of pins assigned to an I/O bank against the number of pins allowed in the I/O bank.	No
I/O bank VCCIO voltage compatibility	Checks that no more than one VCCIO is required for the pins assigned to the I/O bank.	No
I/O bank VREF voltage compatibility	Checks that no more than one VREF is required for the pins assigned to the I/O bank.	No
I/O standard and location conflicts	Checks whether the pin location supports the assigned I/O standard.	No
I/O standard and signal direction conflicts	Checks whether the pin location supports the assigned I/O standard and direction. For example, certain I/O standards on a particular pin location can only support output pins.	No
Differential I/O standards cannot have open drain turned on	Checks that open drain is turned off for all pins with a differential I/O standard.	No
I/O standard and drive strength conflicts	Checks whether the drive strength assignments are within the specifications of the I/O standard.	No
Drive strength and location conflicts	Checks whether the pin location supports the assigned drive strength.	No
BUSHOLD and location conflicts	Checks whether the pin location supports BUSHOLD. For example, dedicated clock pins do not support BUSHOLD.	No

continued...

Rule	Description	HDL Required?
WEAK_PULLUP and location conflicts	Checks whether the pin location supports WEAK_PULLUP (for example, dedicated clock pins do not support WEAK_PULLUP).	No
Electromigration check	Checks whether combined drive strength of consecutive pads exceeds a certain limit. For example, the total current drive for 10 consecutive pads on a Stratix II device cannot exceed 200 mA.	No
PCI_IO clamp diode, location, and I/O standard conflicts	Checks whether the pin location along with the I/O standard assigned supports PCI_IO clamp diode.	No
SERDES and I/O pin location compatibility check	Checks that all pins connected to a SERDES in your design are assigned to dedicated SERDES pin locations.	Yes
PLL and I/O pin location compatibility check	Checks whether pins connected to a PLL are assigned to the dedicated PLL pin locations.	Yes

Table 22. Signal Switching Noise Rules

Rule	Description	HDL Required?
I/O bank cannot have single-ended I/O when DPA exists	Checks that no single-ended I/O pin exists in the same I/O bank as a DPA.	No
A PLL I/O bank does not support both a single-ended I/O and a differential signal simultaneously	Checks that there are no single-ended I/O pins present in the PLL I/O Bank when a differential signal exists.	No
Single-ended output is required to be a certain distance away from a differential I/O pin	Checks whether single-ended output pins are a certain distance away from a differential I/O pin.	No
Single-ended output must be a certain distance away from a VREF pad	Checks whether single-ended output pins are a certain distance away from a VREF pad.	No
Single-ended input is required to be a certain distance away from a differential I/O pin	Checks whether single-ended input pins are a certain distance away from a differential I/O pin.	No
Too many outputs or bidirectional pins in a VREFGROUP when a VREF is used	Checks that there are no more than a certain number of outputs or bidirectional pins in a VREFGROUP when a VREF is used.	No
Too many outputs in a VREFGROUP	Checks whether too many outputs are in a VREFGROUP.	No

3.4.2. I/O Assignment Analysis

I/O assignment analysis validates I/O assignments against the complete set of I/O system and board layout rules. Full I/O assignment analysis validates blocks that directly feed or are fed by resources such as a PLL, LVDS, or gigabit transceiver blocks. In addition, the checker validates the legality of proper VREF pin use, pin locations, and acceptable mixed I/O standards

Run I/O assignment analysis during early pin planning to validate initial reserved pin assignments before compilation. Once you define design files, run I/O assignment analysis to perform more thorough legality checks with respect to the synthesized netlist. Run I/O assignment analysis whenever you modify I/O assignments.

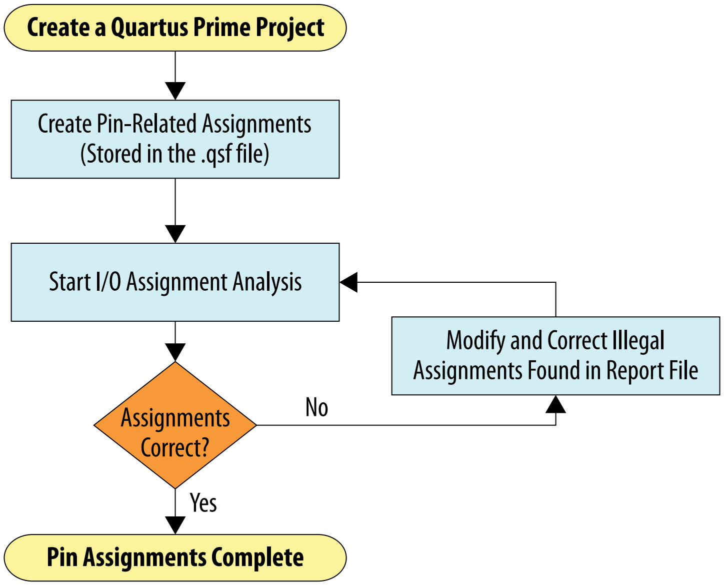
The Fitter assigns pins to accommodate your constraints. For example, if you assign an edge location to a group of LVDS pins, the Fitter assigns pin locations for each LVDS pin in the specified edge location and then performs legality checks. To display the Fitter-placed pins, click **Show Fitter Placements** in the Pin Planner. To accept these suggested pin locations, you must back-annotate your pin assignments.

View the I/O Assignment Warnings report to view and resolve all assignment warnings. For example, a warning that some design pins have undefined drive strength or slew rate. The Fitter recognizes undefined, single-ended output and bidirectional pins as non-calibrated OCT. To resolve the warning, assign the **Current Strength**, **Slew Rate** or **Slow Slew Rate** for the reported pin. Alternatively, can assign the **Termination** to the pin. You cannot assign drive strength or slew rate settings when a pin has an OCT assignment.

3.4.2.1. Early I/O Assignment Analysis Without Design Files

You can perform basic I/O legality checks before defining HDL design files. This technique produces a preliminary board layout. For example, you can specify a target device and enter pin assignments that correspond to PCB characteristics. You can reserve and assign I/O standards to each pin, and then run I/O assignment analysis to ensure that there are no I/O standard conflicts in each I/O bank.

Figure 31. Assigning and Analyzing Pin-Outs without Design Files


Create a Quartus Prime Project
Create Pin-Related Assignments (Stored in the .qsf file)
Start I/O Assignment Analysis
Modify and Correct Illegal Assignments Found in Report File
Assignments Correct?
No
Yes
Pin Assignments Complete

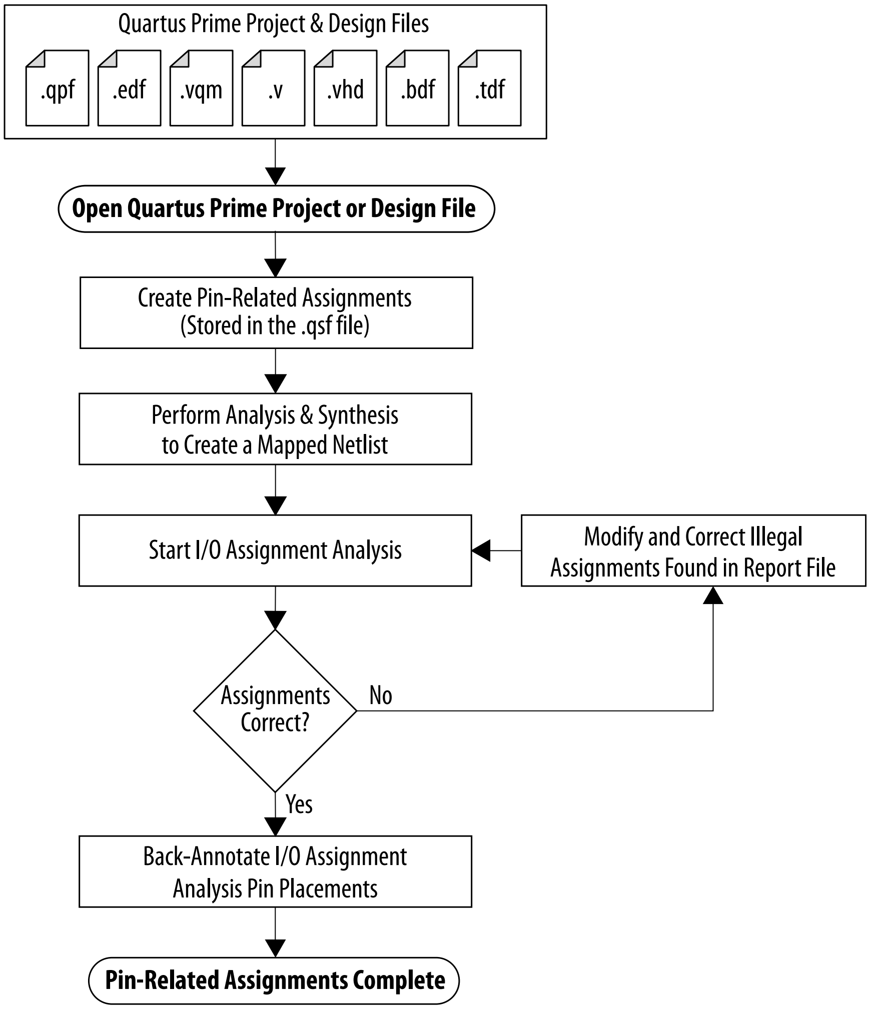
You must reserve all pins you intend to use as I/O pins, so that the Fitter can determine each pin type. After performing I/O assignment analysis, correct any errors reported by the Fitter and rerun I/O assignment analysis until all errors are corrected. A complete I/O assignment analysis requires all design files.

3.4.2.2. I/O Assignment Analysis With Design Files

I/O assignment analysis allows you to perform full I/O legality checks after fully defining HDL design files. When you run I/O assignment analysis on a complete design, the tool verifies all I/O pin assignments against all I/O rules. When you run

I/O assignment analysis on a partial design, the tool checks legality only for defined portions of the design. The following figure shows the work flow for analyzing pin-outs with design files.

Figure 32. I/O Assignment Analysis Flow

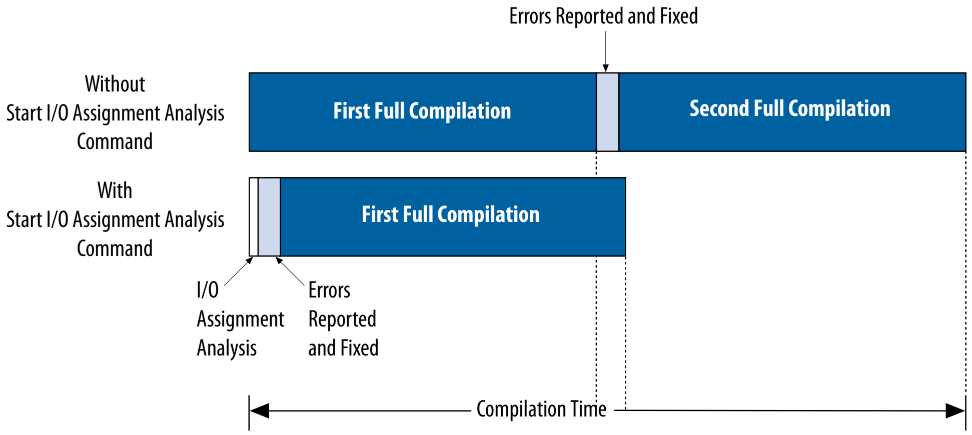

Quartus Prime Project & Design Files
.qpf
.edf
.vqm
.v
.vhd
.bdf
.tdf
Open Quartus Prime Project or Design File
Create Pin-Related Assignments (Stored in the .qsf file)
Perform Analysis & Synthesis to Create a Mapped Netlist
Start I/O Assignment Analysis
Modify and Correct Illegal Assignments Found in Report File
Assignments Correct?
No
Yes
Back-Annotate I/O Assignment Analysis Pin Placements
Pin-Related Assignments Complete

Even if I/O assignment analysis passes on incomplete design files, you may still encounter errors during full compilation. For example, you can assign a clock to a user I/O pin instead of assigning to a dedicated clock pin, or design the clock to drive a PLL that you have not yet instantiated in the design. This issues occur because I/O assignment analysis does not account for the logic that the pin drives and does not verify that only dedicated clock inputs can drive the a PLL clock port.

To obtain better coverage, analyze as much of the design as possible over time, especially logic that connects to pins. For example, if your design includes PLLs or LVDS blocks, define these files prior to full analysis. After performing I/O assignment analysis, correct any errors reported by the Fitter and rerun I/O assignment analysis until all errors are corrected.

The following figure shows the compilation time benefit of performing I/O assignment analysis before running a full compilation.

Figure 33. I/O Assignment Analysis Reduces Compilation Time



3.4.2.3. Overriding Default I/O Pin Analysis

You can override the default I/O analysis of pins to accommodate I/O rule exceptions, such as for analyzing VREF or inactive pins.

Each device contains VREF pins, each supporting one or more I/O pins. A VREF pin and its I/O pins comprise a VREF bank. The VREF pins are typically assigned inputs with VREF I/O standards, such as HSTL- and SSTL-type I/O standards. Conversely, VREF outputs do not require the VREF pin. When a voltage-referenced input is present in a VREF bank, only a certain number of outputs can be present in that VREF bank. I/O assignment analysis treats bidirectional signals controlled by different output enables as independent output enables.

To assign the **Output Enable Group** option to bidirectional signals to analyze the signals as a single output enable group, follow these steps:

1. To access this assignment in the Pin Planner, right-click the **All pins** list and click **Customize Columns**.
2. Under **Available columns**, add **Output Enable Group** to **Show these columns in this order**. The column appears in the **All Pins** list.
3. Enter the same integer value for the **Output Enable Group** assignment for all sets of signals that are driving in the same direction.

Related Information

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

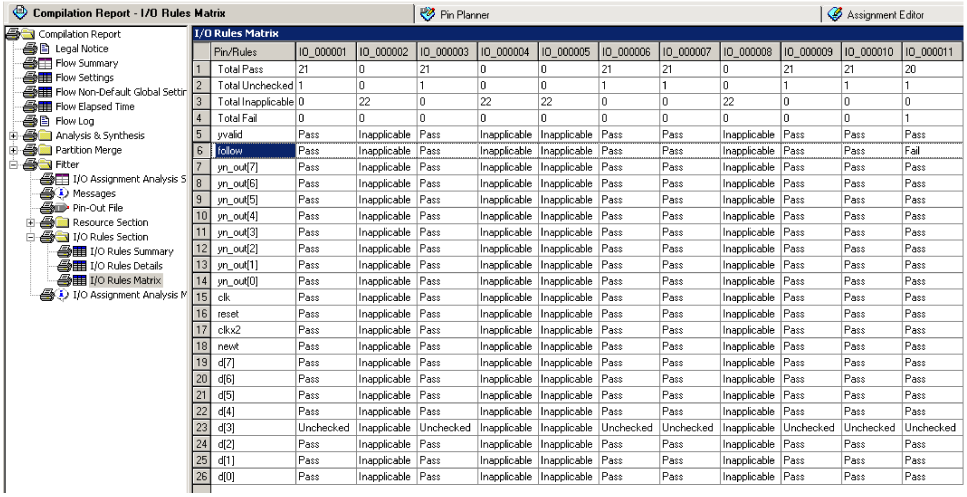
3.4.3. Understanding I/O Analysis Reports

The detailed I/O assignment analysis reports include the affected pin name and a problem description. The Fitter section of the Compilation report contains information generated during I/O assignment analysis, including the following reports:

- I/O Assignment Warnings—lists warnings generated for each pin
- Resource Section—quantifies use of various pin types and I/O banks
- I/O Rules Section—lists summary, details, and matrix information about the I/O rules tested

The **Status** column indicates whether rules passed, failed, or were not checked. A severity rating indicates the rule’s importance for effective analysis. “Inapplicable” rules do not apply to the target device family.

Figure 34. I/O Rules Matrix



I/O Rules Matrix

	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. Verifying I/O Timing

You must verify board-level signal integrity and I/O timing when assigning I/O pins. High-speed interface operation requires a quality signal and low propagation delay at the far end of the board route. Click **Tools ➤ Timing Analyzer** to confirm timing after making I/O pin assignments.

For example, if you change the slew rates or drive strengths of some I/O pins with ECOs, you can verify timing without recompiling the design. You must understand I/O timing and what factors affect I/O timing paths in your design. The accuracy of the output load specification of the output and bidirectional pins affects the I/O timing results.

The Intel Quartus Prime software supports three different methods of I/O timing analysis:

Table 23. I/O Timing Analysis Methods

I/O Timing Analysis	Description
Advanced I/O timing analysis	Analyze I/O timing with your board trace model to report accurate, "board-aware" simulation models. Configures a complete board trace model for each I/O standard or pin. Timing Analyzer applies simulation results of the I/O buffer, package, and board trace model to generate accurate I/O delays and system level signal information. Use this information to improve timing and signal integrity.
I/O timing analysis	Analyze I/O timing with default or specified capacitive load without signal integrity analysis. Timing Analyzer reports tCO to an I/O pin using a default or user-specified value for a capacitive load.
Full board routing simulation	Use Intel-provided or Intel Quartus Prime software-generated IBIS or HSPICE I/O models for simulation in Mentor Graphics HyperLynx* and Synopsys HSPICE.

For more information about advanced I/O timing support, refer to the appropriate device handbook for your target device. For more information about board-level signal integrity and tips on how to improve signal integrity in your high-speed designs, refer to the Altera Signal Integrity Center page of the Altera website.

For information about creating IBIS and HSPICE models with the Intel Quartus Prime software and integrating those models into HyperLynx and HSPICE simulations, refer to the *Signal Integrity Analysis with Third Party Tools* chapter.

Related Information

- Literature and Technical Documentation
- Intel Signal & Power Integrity Center

3.5.1. Running Advanced I/O Timing

Advanced I/O timing analysis uses your board trace model and termination network specification to report accurate output buffer-to-pin timing estimates, FPGA pin and board trace signal integrity and delay values. Advanced I/O timing runs automatically for supported devices during compilation.

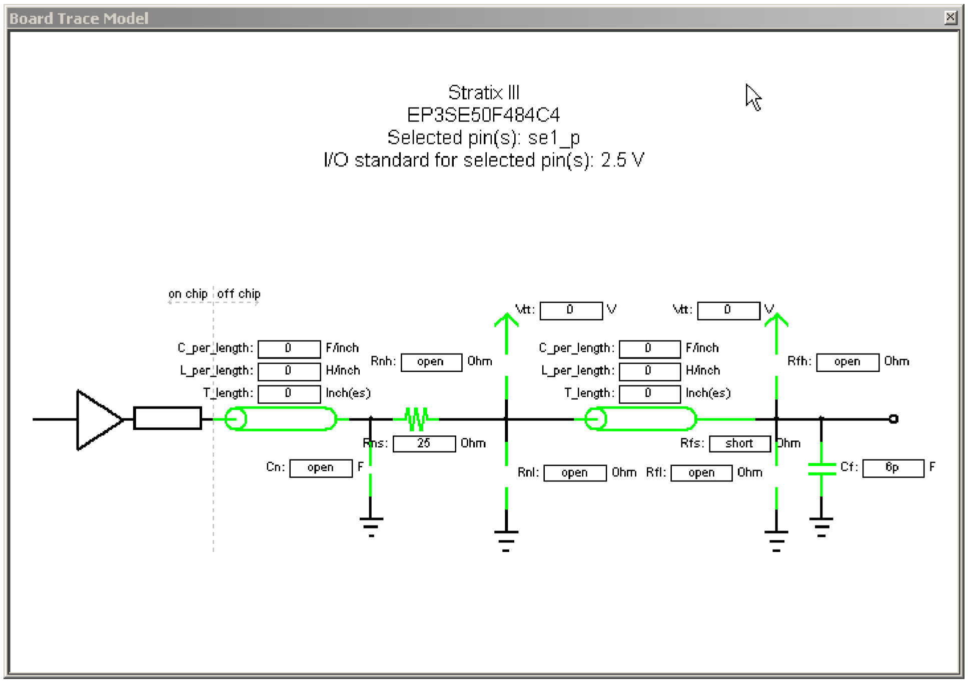
3.5.1.1. Board Trace Models

The Intel Quartus Prime software provides board trace model templates for various I/O standards.

The following figure shows the template for a **2.5 V** I/O standard. This model consists of near-end and far-end board component parameters.

Near-end board trace modeling includes the elements which are close to the device. Far-end modeling includes the elements which are at the receiver end of the link, closer to the receiving device. Board trace model topology is conceptual and does not necessarily match the actual board trace for every component. For example, near-end model parameters can represent device-end discrete termination and breakout traces. Far-end modeling can represent the bulk of the board trace to discrete external memory components, and the far end termination network. You can analyze the same circuit with near-end modeling of the entire board, including memory component termination, and far-end modeling of the actual memory component.

Figure 35. 2.5-V I/O Standard Board Trace Model



Board Trace Model

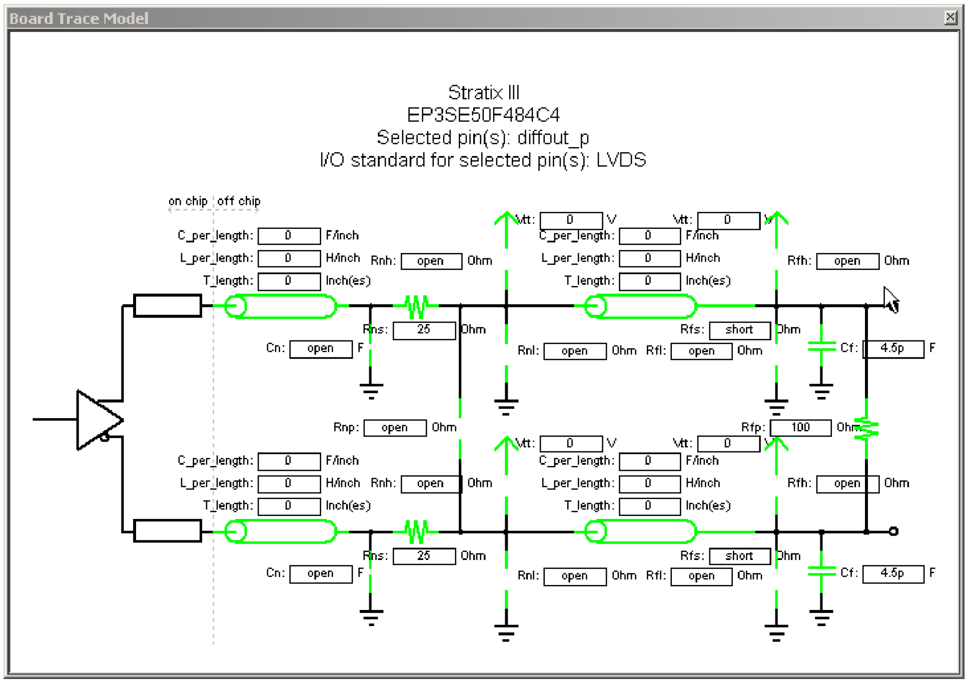
Stratix III
EP3SE50F484C4
Selected pin(s): se1_p
I/O standard for selected pin(s): 2.5 V

C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Cn: open F
Rns: 25 Ohm
Rnh: open Ohm
Vtt: 0 V
Rnl: open Ohm
C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Rfs: short Ohm
Rfl: open Ohm
Vtt: 0 V
Rfh: open Ohm
Cf: 6p F

The following figure shows the template for the **LVDS** I/O standard. The far-end capacitance (Cf) represents the external-device or multiple-device capacitive load. If you have multiple devices on the far-end, you must find the equivalent capacitance at the far-end, taking into account all receiver capacitances. The far-end capacitance can be the sum of all the receiver capacitances.

The Intel Quartus Prime software models of transmission lines do not consider transmission-line resistance (lossless models). You only need to specify distributed inductance (L) and capacitance (C) values on a per-inch basis, which you can obtain from the PCB vendor or manufacturer, the CAD Design tool, or a signal integrity tool, such as the Mentor Graphics HyperLynx software.

Figure 36. LVDS Differential Board Trace Model



Stratix III
EP3SE50F484C4
Selected pin(s): diffout_p
I/O standard for selected pin(s): LVDS

C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Rnh: open Ohm
Rns: 25 Ohm
Cn: open F
Vtt: 0 V
C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Rnl: open Ohm
Vtt: 0 V
Rfh: open Ohm
Rfs: short Ohm
Rfl: open Ohm
Cf: 4.5p F
Rnp: open Ohm
Rfp: 100 Ohm
C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Rnh: open Ohm
Rns: 25 Ohm
Cn: open F
Vtt: 0 V
C_per_length: 0 F/inch
L_per_length: 0 H/inch
T_length: 0 Inch(es)
Rnl: open Ohm
Vtt: 0 V
Rfh: open Ohm
Rfs: short Ohm
Rfl: open Ohm
Cf: 4.5p F

3.5.1.2. Defining the Board Trace Model

The board trace model describes a board trace and termination network as a set of capacitive, resistive, and inductive parameters.

Advanced I/O Timing uses the model to simulate the output signal from the output buffer to the far end of the board trace. You can define the capacitive load, any termination components, and trace impedances in the board routing for any output pin or bidirectional pin in output mode. You can configure an overall board trace model for each I/O standard or for specific pins. Define an overall board trace model for each I/O standard in your design. Use that model for all pins that use the I/O standard. You can customize the model for specific pins using the **Board Trace Model** window in the Pin Planner.

1. Click **Assignments ➤ Device ➤ Device and Pin Options**.
2. Click **Board Trace Model** and define board trace model values for each I/O standard.
3. Click **I/O Timing** and define default I/O timing options at board trace near and far ends.
4. Click **Assignments ➤ Pin Planner** and assign board trace model values to individual pins.

Example 8. Specifying Board Trace Model

```
## 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. Modifying the Board Trace Model

To modify the board trace model, click **View ➤ Board Trace Model** in the Pin Planner.

You can modify any of the board trace model parameters within a graphical representation of the board trace model.

The **Board Trace Model** window displays the routing and components for positive and negative signals in a differential signal pair. Only modify the positive signal of the pair, as the setting automatically applies to the negative signal. Use standard unit prefixes such as *p*, *n*, and *k* to represent pico, nano, and kilo, respectively. Use the **short** or **open** value to designate a short or open circuit for a parallel component.

3.5.1.4. Specifying Near-End vs Far-End I/O Timing Analysis

You can select a near-end or far-end point for I/O timing analysis. Near-end timing analysis extends to the device pin. You can apply the `set_output_delay` constraint during near-end analysis to account for the delay across the board.

With far-end I/O timing analysis, the advanced I/O timing analysis extends to the external device input, at the far-end of the board trace. Whether you choose a near-end or far-end timing endpoint, the board trace models are taken into account during timing analysis.

3.5.1.5. Advanced I/O Timing Analysis Reports

The following reports show advanced I/O timing analysis information:

Table 24. Advanced I/O Timing Reports

I/O Timing Report	Description
Timing Analyzer Report	Reports signal integrity and board delay data.
Board Trace Model Assignments report	Summarizes the board trace model component settings for each output and bidirectional signal.
Signal Integrity Metrics report	Contains all the signal integrity metrics calculated during advanced I/O timing analysis based on the board trace model settings for each output or bidirectional pin. Includes measurements at both the FPGA pin and at the far-end load of board delay, steady state voltages, and rise and fall times.

Note: By default, the Timing Analyzer generates the Slow-Corner Signal Integrity Metrics report. To generate a Fast-Corner Signal Integrity Metrics report you must change the delay model by clicking **Tools ➤ Timing Analyzer**.

Related Information

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

3.5.2. Adjusting I/O Timing and Power with Capacitive Loading

When calculating t_{CO} and power for output and bidirectional pins, the Timing Analyzer and the Power Analyzer use a bulk capacitive load. You can adjust the value of the capacitive load per I/O standard to obtain more precise t_{CO} and power measurements, reflecting the behavior of the output or bidirectional net on your PCB. The Intel Quartus Prime software ignores capacitive load settings on input pins. You can adjust the capacitive load settings per I/O standard, in picofarads (pF), for your entire design. During compilation, the Compiler measures power and t_{CO} measurements based on your settings. You can also adjust the capacitive load on an individual pin with the **Output Pin Load** logic option.

3.6. Viewing Routing and Timing Delays

Right-click any node and click **Locate > Locate in Chip Planner** to visualize and adjust I/O timing delays and routing between user I/O pads and V_{CC}, GND, and V_{REF} pads. The Chip Planner graphically displays logic placement, Logic Lock regions, relative resource usage, detailed routing information, fan-in and fan-out, register paths, and high-speed transceiver channels. You can view physical timing estimates, routing congestion, and clock regions. Use the Chip Planner to change connections between resources and make post-compilation changes to logic cell and I/O atom placement. When you select items in the Pin Planner, the corresponding item is highlighted in Chip Planner.

3.7. Scripting API

The Intel Quartus Prime software allows you to access I/O management functions through Tcl commands, rather than with the GUI. For detailed information about scripting command options and Tcl API packages, type the following at a system command prompt to view the Tcl API Help browser:

```
quartus_sh --qhelp
```

Related Information

- Tcl Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*
- Command Line Scripting
  In *Intel Quartus Prime Pro Edition User Guide: Scripting*

3.7.1. Generate Mapped Netlist

Enter the following in the Tcl console or in a Tcl script:

```
execute_module -tool map
```

The `execute_module` command is in the flow package.

Type the following at a system command prompt:

```
quartus_syn <project name>
```

3.7.2. Reserve Pins

Use the following Tcl command to reserve a pin:

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

Use one of the following valid reserved pin values:

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

Note: You must include the quotation marks when specifying the reserved pin value.

3.7.3. Set Location

Use the following Tcl command to assign a signal to a pin or device location:

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

Valid locations are pin locations, I/O bank locations, or edge locations. Pin locations include pin names, such as `PIN_A3`. I/O bank locations include `IOBANK_1` up to `IOBANK_` *n*, where *n* is the number of I/O banks in the device.

Use one of the following valid edge location values:

- `EDGE_BOTTOM`
- `EDGE_LEFT`
- `EDGE_TOP`
- `EDGE_RIGHT`

3.7.4. Exclusive I/O Group

The following Tcl command creates an exclusive I/O group assignment:

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

3.7.5. Slew Rate and Current Strength

Use the following Tcl commands to create a slew rate and drive strength assignments:

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

Related Information

Package Information Datasheet for Mature Altera Devices

3.8. Managing Device I/O Pins Revision History

The following table shows the revision history for this chapter:

Document Version	Intel Quartus Prime Version	Changes
2018.05.07	18.0.0	• First release as part of the stand-alone *Design Constraints User Guide*
2017.11.06	17.1.0	• Revised topic: I/O Planning Overview. • Revised topic: Basic I/O Planning Flow with the Pin Planner and renamed to Basic I/O Planning Flow with the Pin Planner.
2017.05.08	17.0.0	• Renamed command: Run I/O Assignment Analysis to Start Fitter (Plan).
2016.10.31	16.1.0	• Implemented Intel rebranding.
2015.11.02	15.1.0	• Removed early pin planning and Live I/O Check support from Quartus Prime Pro Edition handbook • Changed instances of *Quartus II* to *Quartus Prime*.
2014.12.15	14.1.0	• Updated Live I/O check device support to include only limited device families.
2014.08.30	14.0a10.0	• Added link to information about special pin assignment features for Arria 10 SoC devices.
2014.06.30	14.0.0	• Replaced MegaWizard Plug-In Manager information with IP Catalog.
November 2013	13.1.0	• Reorganization and conversion to DITA.
May 2013	13.0.0	• Added information about overriding I/O placement rules.
November 2012	12.1.0	• Updated Pin Planner description for new task and report windows.
June 2012	12.0.0	• Removed survey link.
November 2011	11.1.0	• Minor updates and corrections. • Updated the document template.
December 2010	10.0.1	Template update
July 2010	10.0.0	• Reorganized and edited the chapter • Added links to Help for procedural information previously included in the chapter • Added information on rules marked Inapplicable in the I/O Rules Matrix Report • Added information on assigning slew rate and drive strength settings to pins to fix I/O assignment warnings
November 2009	9.1.0	• Reorganized entire chapter to include links to Help for procedural information previously included in the chapter • Added documentation on near-end and far-end advanced I/O timing
March 2009	9.0.0	• Updated “Pad View Window” on page 5–20 • Added new figures: • Figure 5–15 • Figure 5–16 • Added new section “Viewing Simultaneous Switching Noise (SSN) Results” on page 5–17 • Added new section “Creating Exclusive I/O Group Assignments” on page 5–18

Related Information

Documentation Archive

For previous versions of the *Intel Quartus Prime Handbook*, search the documentation archives.

A. Intel Quartus Prime Pro Edition User Guides

Refer to the following user guides for comprehensive information on all phases of the Intel Quartus Prime Pro Edition FPGA design flow.

Related Information

- Intel Quartus Prime Pro Edition User Guide: Getting Started
  Introduces the basic features, files, and design flow of the Intel Quartus Prime Pro Edition software, including managing Intel Quartus Prime Pro Edition projects and IP, initial design planning considerations, and project migration from previous software versions.
- Intel Quartus Prime Pro Edition User Guide: Platform Designer
  Describes creating and optimizing systems using Platform Designer, a system integration tool that simplifies integrating customized IP cores in your project. Platform Designer automatically generates interconnect logic to connect intellectual property (IP) functions and subsystems.
- Intel Quartus Prime Pro Edition User Guide: Design Recommendations
  Describes best design practices for designing FPGAs with the Intel Quartus Prime Pro Edition software. HDL coding styles and synchronous design practices can significantly impact design performance. Following recommended HDL coding styles ensures that Intel Quartus Prime Pro Edition synthesis optimally implements your design in hardware.
- Intel Quartus Prime Pro Edition User Guide: Design Compilation
  Describes set up, running, and optimization for all stages of the Intel Quartus Prime Pro Edition Compiler. The Compiler synthesizes, places, and routes your design before generating a device programming file.
- Intel Quartus Prime Pro Edition User Guide: Design Optimization
  Describes Intel Quartus Prime Pro Edition settings, tools, and techniques that you can use to achieve the highest design performance in Intel FPGAs. Techniques include optimizing the design netlist, addressing critical chains that limit retiming and timing closure, and optimization of device resource usage.
- Intel Quartus Prime Pro Edition User Guide: Programmer
  Describes operation of the Intel Quartus Prime Pro Edition Programmer, which allows you to configure Intel FPGA devices, and program CPLD and configuration devices, via connection with an Intel FPGA download cable.
- Intel Quartus Prime Pro Edition User Guide: Block-Based Design
  Describes block-based design flows, also known as modular or hierarchical design flows. These advanced flows enable preservation of design blocks (or logic that comprises a hierarchical design instance) within a project, and reuse of design blocks in other projects.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

- Intel Quartus Prime Pro Edition User Guide: Partial Reconfiguration
  Describes Partial Reconfiguration, an advanced design flow that allows you to reconfigure a portion of the FPGA dynamically, while the remaining FPGA design continues to function. Define multiple personas for a particular design region, without impacting operation in other areas.
- Intel Quartus Prime Pro Edition User Guide: Third-party Simulation
  Describes RTL- and gate-level design simulation support for third-party simulation tools by Aldec*, Cadence*, Mentor Graphics, and Synopsys that allow you to verify design behavior before device programming. Includes simulator support, simulation flows, and simulating Intel FPGA IP.
- Intel Quartus Prime Pro Edition User Guide: Third-party Synthesis
  Describes support for optional synthesis of your design in third-party synthesis tools by Mentor Graphics, and Synopsys. Includes design flow steps, generated file descriptions, and synthesis guidelines.
- Intel Quartus Prime Pro Edition User Guide: Third-party Logic Equivalence Checking Tools
  Describes support for optional logic equivalence checking (LEC) of your design in third-party LEC tools by OneSpin*. Describes how to verify the logic equivalence between compilation netlists.
- Intel Quartus Prime Pro Edition User Guide: Debug Tools
  Describes a portfolio of Intel Quartus Prime Pro Edition in-system design debugging tools for real-time verification of your design. These tools provide visibility by routing (or “tapping”) signals in your design to debugging logic. These tools include System Console, Signal Tap logic analyzer, Transceiver Toolkit, In-System Memory Content Editor, and In-System Sources and Probes Editor.
- Intel Quartus Prime Pro Edition User Guide: Timing Analyzer
  Explains basic static timing analysis principals and use of the Intel Quartus Prime Pro Edition Timing Analyzer, a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in your design using an industry-standard constraint, analysis, and reporting methodology.
- Intel Quartus Prime Pro Edition User Guide: Power Analysis and Optimization
  Describes the Intel Quartus Prime Pro Edition Power Analysis tools that allow accurate estimation of device power consumption. Estimate the power consumption of a device to develop power budgets and design power supplies, voltage regulators, heat sink, and cooling systems.
- Intel Quartus Prime Pro Edition User Guide: Design Constraints
  Describes timing and logic constraints that influence how the Compiler implements your design, such as pin assignments, device options, logic options, and timing constraints. Use the Interface Planner to prototype interface implementations, plan clocks, and quickly define a legal device floorplan. Use the Pin Planner to visualize, modify, and validate all I/O assignments in a graphical representation of the target device.
- Intel Quartus Prime Pro Edition User Guide: PCB Design Tools
  Describes support for optional third-party PCB design tools by Mentor Graphics and Cadence*. Also includes information about signal integrity analysis and simulations with HSPICE and IBIS Models.

- Intel Quartus Prime Pro Edition User Guide: Scripting
  Describes use of Tcl and command line scripts to control the Intel Quartus Prime Pro Edition software and to perform a wide range of functions, such as managing projects, specifying constraints, running compilation or timing analysis, or generating reports.