Managing Device I/O Pins

4

2013.11.04

QII52013

Subscribe Send Feedback

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

Figure 4-1: Pin Planner GUI

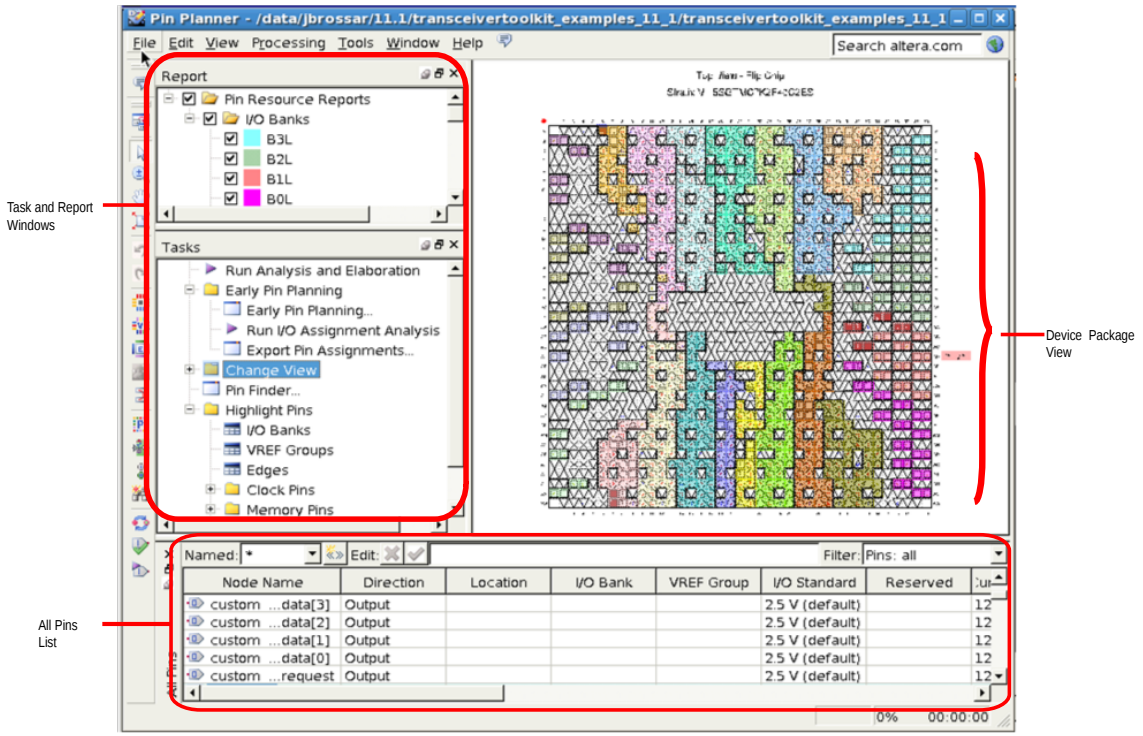


Table 4-1: Quartus II I/O Pin Planning Tools

I/O Planning Task	Click to Access
Edit, validate, or export pin assignments	**Assignments > Pin Planner**
View tailored pin planning advice	**Tools > Advisors > Pin Advisor**
Validate pin assignments against design rules	**Processing > Start I/O Assignment Analysis**

© 2013 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera 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 Altera. Altera 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.

ISO 9001:2008 Registered

I/O Planning Overview

You should plan and assign I/O pins in your design for compatibility with your target device and PCB characteristics. Plan I/O pins early to reduce design iterations and develop an accurate PCB layout sooner. You can assign expected nodes not yet defined in design files, including interface IP core signals, and then generate a top-level file. Specify interfaces for memory, high-speed I/O, device configuration, and debugging tools in your top-level file. The top-level file instantiates the next level of design hierarchy and includes interface port information.

Use the Pin Planner 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. The Pin Planner Task window provides one-click access to I/O planning steps. You can filter and search the nodes in the design. You can define custom groups of pins for assignment. Instantly locate and highlight specific pin types for assignment or evaluation, such as I/O banks, VREF groups, edges, DQ/DQS pins, hard memory interface pins, PCIe hard IP interface pins, hard processor system pins, and clock region input pins. Assign design elements, I/O standards, interface IP, and other properties to the device I/O pins by name or by drag and drop. You can then generate a top-level design file for I/O validation.

Use live I/O check to verify the legality of pin assignments in real time. Use I/O assignment analysis to run a full I/O analysis 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.

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 /O standards, voltage and power supply requirements, and available I/O pins.
2. Click **Assignments > Pin Planner**.
3. To setup a top-level HDL wrapper file that defines early port and interface information for your design, click **Early Pin Planning** in the Tasks pane.
   a. Click **Create/Import IP Core** and define the parameters of any expected interface IP core, and then assign signals to the interface IP nodes.
   b. Click **Set Up Top-Level File** and assign user nodes to device pins. User nodes become virtual pins in the top-level file and are not assigned to device pins.
   c. Click **Generate Top-Level File**. Use this file to validate I/O assignments.
4. Click **Run I/O Assignment Analysis** in the Tasks pane to validate any early assignments and generate a synthesized design netlist.
5. Assign I/O properties to match your device and PCB characteristics, including assigning logic, I/O standards, output loading, slew rate, and current strength.
6. Click **Run I/O Assignment Analysis** in the Tasks pane to validate assignments and generate a synthesized design netlist. Correct any problems reported.
7. Click **Processing > Start Compilation**. During compilation, the Quartus II software runs I/O assignment analysis and advanced I/O timing analysis.

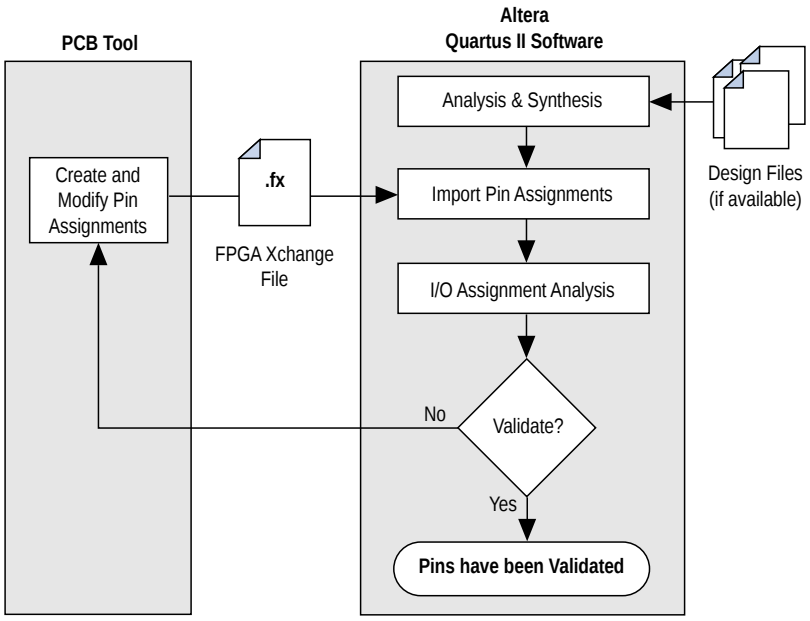
Integrating PCB Design Tools

You can integrate PCB design tools into your work flow to help correctly map pin assignments to the symbols your system circuit schematics and board layout. The Quartus II software integrates with board layout tools by allowing import and export of pin assignment information in Quartus II Settings Files (**.qsf**), Pin-Out File (**.pin**), and FPGA Xchange-Format File (**.fx**) files. You can integrate PCB tools in the the following ways:

Table 4-2: 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 DesignerCadence Allegro
Define I/O assignments in your PCB tool, and then import the assignments into the Pin Planner for validation	Mentor Graphics® I/O DesignerCadence Allegro

Figure 4-2: PCB Tool Integration



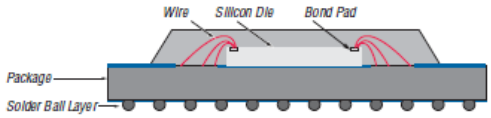
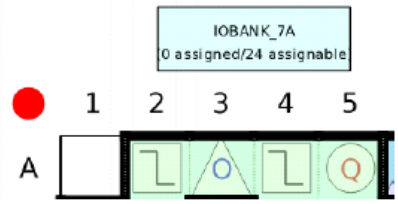
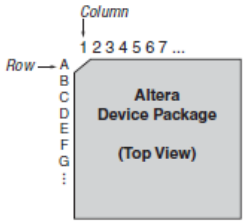
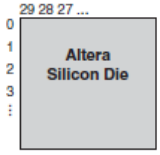
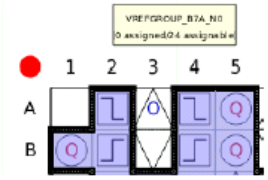
For more information about incorporating PCB design tools, refer to the *Cadence PCB Design Tools Support* and *Mentor Graphics PCB Design Tools Support* chapters in volume 2 of the *Quartus II Handbook*.

Related Information

Mentor Graphics PCB Design Tools Support

Altera Device Terms

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

Assigning I/O Pins

Use the Pin Planner to visualize, modify, and validate I/O assignments in a graphical representation of the target device. To assign I/O pins, locate the device I/O pin(s) for assignment, enter properties for the pin(s), and validate the legality of the assignment. 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 a Quartus II project, and then click **Assignments > Pin Planner**.
2. (Optional) To validate I/O pin assignments in real time, click **Processing > Enable Live I/O Check**.
3. Click **Processing > Start Analysis & Elaboration** to elaborate the design and display **All Pins** in the device view.
4. To locate or highlight pins for assignment, click **Pin Finder** or a pin type under **Highlight Pins** in the Tasks pane.
5. (Optional) To define a custom group of nodes for assignment, select one or more nodes in the **Groups** or **All Pins** list, and then click **Create Group**.
6. Enter assignments of logic, I/O standards, interface IP, and properties for device I/O pins in the **All Pins** spreadsheet, or by drag and drop into the package view.
7. 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.
8. (Optional) To create board trace model assignments, right-click an output or bidirectional pin, and then click **Board Trace Model**. For differential I/O standards, the board trace model uses a differential pin pair with two symmetrical board trace models. 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.
9. 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.

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.

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. The slew rate and drive strength apply during live I/O check and I/O assignment analysis.

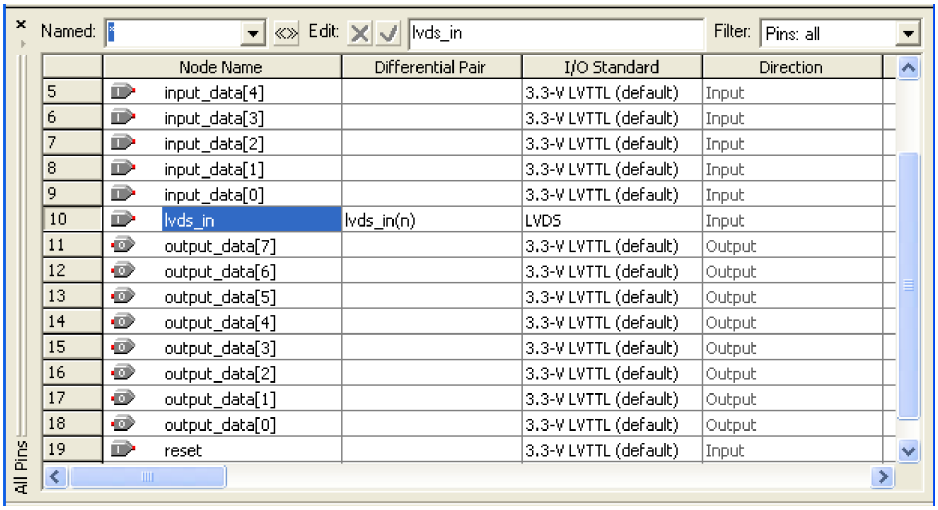
Assigning Differential Pins

When you use the Pin Planner to assign a differential I/O standard to a single-ended top-level pin in your design, it automatically recognizes the negative pin as part of the differential pin pair assignment and creates the negative pin for you. The Quartus II 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.

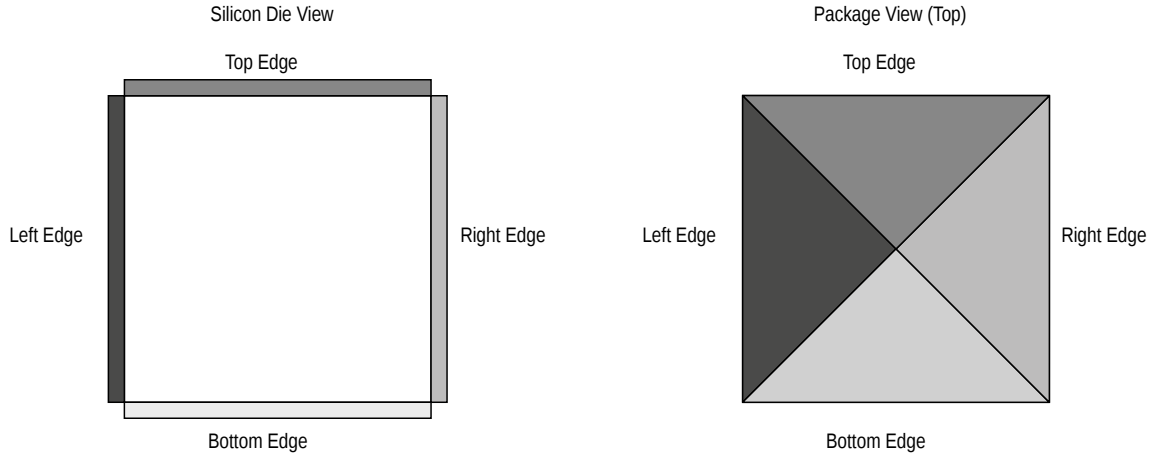
Figure 4-3: Creating a Differential Pin Pair in the Pin Planner



	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 shows Altera device package edges.

Figure 4-4: Die View and Package View of the Four Edges on an Altera Device



Overriding I/O Placement Rules on Differential Pins

Each device family has predefined I/O placement rules. The I/O placement rules ensure that noisy signals do not corrupt neighboring signals. For example, I/O placement rules define the allowed placement of single-ended I/O with respect to differential pins, or how many output and bidirectional pins can be placed within

a VREF group when using voltage referenced input standards. You can use the IO_MAXIMUM_TOGGLE_RATE assignment to override I/O placement rules on pins, such as for 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 intended.

Entering Pin Assignments with Tcl Commands

You can use Tcl scripts to apply pin assignments rather than using the GUI. Enter individual Tcl commands in the Tcl Console, or type the following to apply the assignments contained in a Tcl script:

Example 4-1: Applying Tcl Script Assignments

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

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

Example 4-2: Scripted Pin Assignment

```
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**
- **API Functions**
- **Entering Pin Assignments with Tcl Commands** on page 4-7
- **Scripting API** on page 4-25
- **Entering Pin Assignments with Tcl Commands** on page 4-7
- **Scripting API** on page 4-25

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

Using Synthesis Attributes

The Quartus II software translates synthesis attributes into standard assignments during compilation. The assignments appear in the Pin Planner. If you modify or delete these assignments in the Pin Planner and then recompile your design, the Pin Planner changes override the synthesis attributes. Quartus II synthesis supports the `chip_pin`, `useioff`, and `altera_attribute` synthesis attributes.

Use the `chip_pin` and `useioff` synthesis attributes to create pin location assignments and to assign **Fast Input Register**, **Fast Output Register,** and **Fast Output Enable Register** logic options. The following examples use the `chip_pin` and `useioff` attributes to embed location and **Fast Input Register** logic option assignments in Verilog HDL and VHDL design files.

Example 4-3: Verilog HDL Synthesis Attribute

```
input my_pin1 /* synthesis altera_attribute = "-name
FAST_INPUT_REGISTER ON; -name IO_STANDARD \"2.5 V\" " */ ;
```

Example 4-4: VHDL Synthesis Attribute

```
VHDL Example
entity my_entity is
  port(
     my_pin1: in std_logic
  );
end my_entity;

architecture rtl of my_entity is
attribute useioff : boolean;
attribute useioff of my_pin1 : signal is true;
attribute chip_pin : string;
attribute chip_pin of my_pin1 : signal is "C1";
begin -- The architecture body
end rtl;
```

Use the `altera_attribute` synthesis attribute to create other pin-related assignments in your HDL code. The `altera_attribute` attribute is understood only by Quartus II integrated synthesis and supports all types of instance assignments. The following examples use the `altera_attribute` attribute to embed **Fast Input Register** logic option assignments and I/O standard assignments in both a Verilog HDL and a VHDL design file.

Example 4-5: Verilog HDL Synthesis Attribute

```
input my_pin1 /* synthesis chip_pin = "C1" useioff = 1 */;
```

Example 4-6: VHDL Synthesis Attribute

```
entity my_entity is
  port(
    my_pin1: in std_logic
  );
end my_entity;
architecture rtl of my_entity is
begin

attribute altera_attribute : string;
attribute altera_attribute of my_pin1: signal is "-name
FAST_INPUT_REGISTER ON;
-- The architecture body
end rtl;
```

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

Importing and Exporting I/O Pin Assignments

The Quartus II 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 4-3: 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 Quartus II project with common constraints	• From Quartus II project for optimization in a PCB design tool • From Quartus II project for spreadsheet analysis or use in scripting assignments • From Quartus II project for import into another Quartus II 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

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 Quartus II software requires only the **.fx** to import pin assignments from I/O Designer.

Table 4-4: 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

File Column Name	Description
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

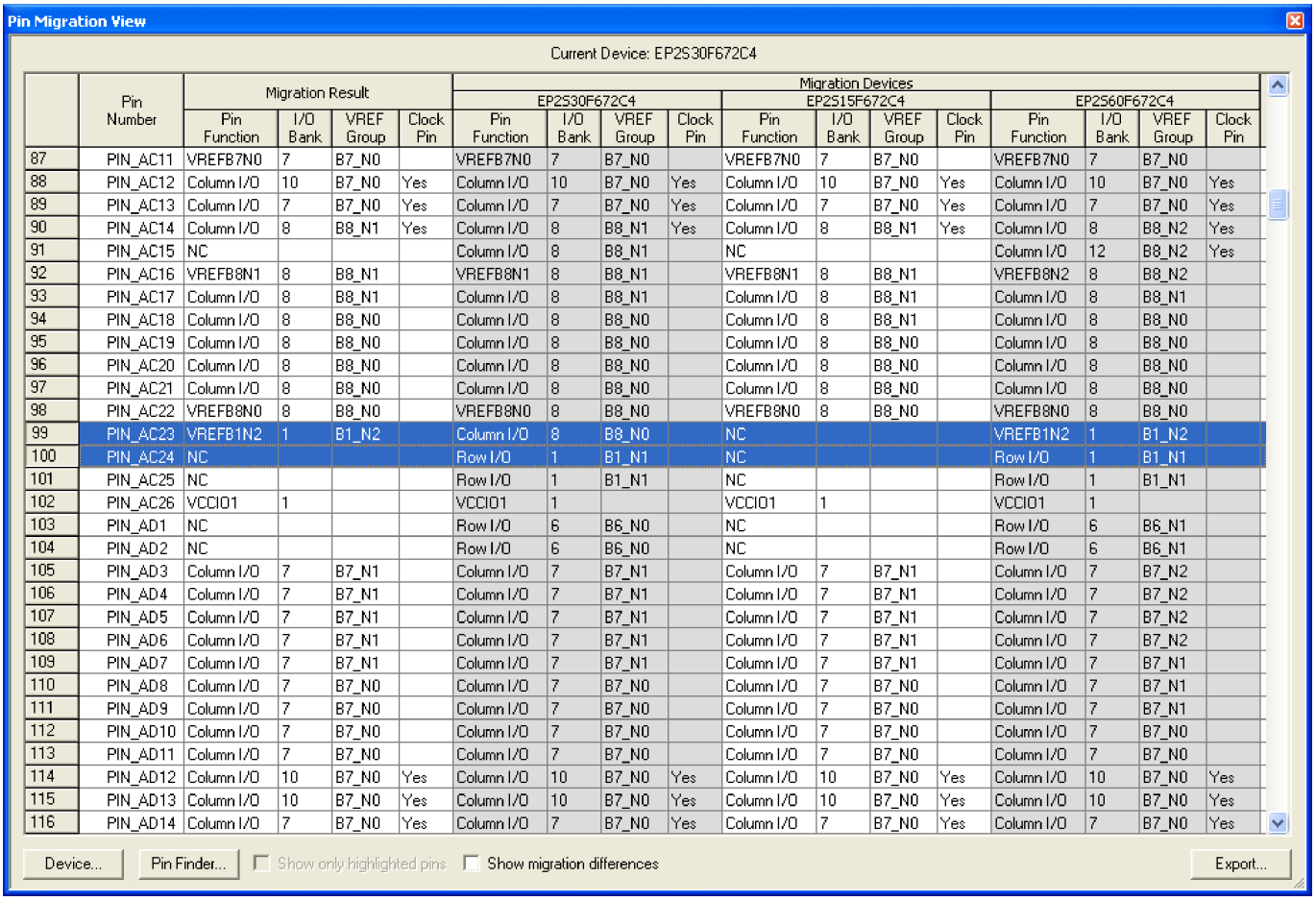
- **Pin-Out Files for Altera Devices**
- **Mentor Graphics PCB Tools Support**

Migrating Assignments to Another Target 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. Click **View > Pin Migration Window** to verify whether your pin assignments are compatible with migration to a different Altera device.

The Quartus II 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 4-5: Device Migration Compatibility (AC24 does not exist in migration device)



	Pin Number	Migration Result Pin Function	Migration Result I/O Bank	Migration Result VREF Group	Migration Result Clock Pin	EP2S30F672C4 Pin Function	EP2S30F672C4 I/O Bank	EP2S30F672C4 VREF Group	EP2S30F672C4 Clock Pin	EP2S15F672C4 Pin Function	EP2S15F672C4 I/O Bank	EP2S15F672C4 VREF Group	EP2S15F672C4 Clock Pin	EP2S60F672C4 Pin Function	EP2S60F672C4 I/O Bank	EP2S60F672C4 VREF Group	EP2S60F672C4 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 display the voltage reference. Even if you do not use that pin for a port connection in your 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 Quartus II 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 your original design because the pins are not physically connected to the smaller die.

Related Information

AN90: SameFrame PinOut Design for FineLine BGA Packages

Validating Pin Assignments

The Quartus II 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 4-5: I/O Validation Tools

I/O Validation Tool	Description	Click to Run
Live I/O Check	Verifies preliminary, basic I/O legality as you enter assignments	**Processing > Enable Live I/O Check**
I/O Assignment Analysis	Verifies I/O assignment legality of synthesized design against full set of I/O rules for the target device	**Processing > Start I/O Assignment Analysis**
Advanced I/O Timing	Fully validates I/O assignments against all I/O and timing checks during compilation	**Processing > Start Compilation**

I/O Assignment Validation Rules

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

Table 4-6: 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

Rule	Description	HDL Required?
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
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 4-7: Signal Switching Noise Rules

Rule	Description	HDL Required?
I/O bank can not 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 has to 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

Rule	Description	HDL Required?
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

Checking I/O Pin Assignments In Real-Time

Live I/O check validates I/O assignments against basic I/O buffer rules in real time. The Pin Planner immediately reports warnings or errors about assignments as you enter them. The Live I/O Check Status window displays the total number of errors and warnings. Use this analysis to quickly correct basic errors before proceeding. Run full I/O assignment analysis when you are ready to validate pin assignments against the complete set of I/O system rules.

Live I/O check validates against the following basic I/O buffer rules:

- V_{CCIO} and V_{REF} voltage compatibility rules
- Electromigration (current density) rules
- Simultaneous Switching Output (SSO) rules
- I/O property compatibility rules, such as drive strength compatibility, I/O standard compatibility, `PCI_IO` clamp diode compatibility, and I/O direction compatibility
- Illegal location assignments:
  - An I/O bank or VREF group with no available pins
  - The negative pin of a differential pair if the positive pin of the differential pair is assigned with a node name with a differential I/O standard
  - Pin locations that do not support the I/O standard assigned to the selected node name
  - For HSTL- and SSTL-type I/O standards, VREF groups of a different V_{REF} voltage than the selected node name.

Note: Live I/O check is supported only for .28nm and larger device families.

Related Information

Assigning Device I/O Pins in Pin Planner

Running 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, you could assign the **Termination** to the pin. You cannot assign drive strength or slew rate settings when a pin has an OCT assignment.

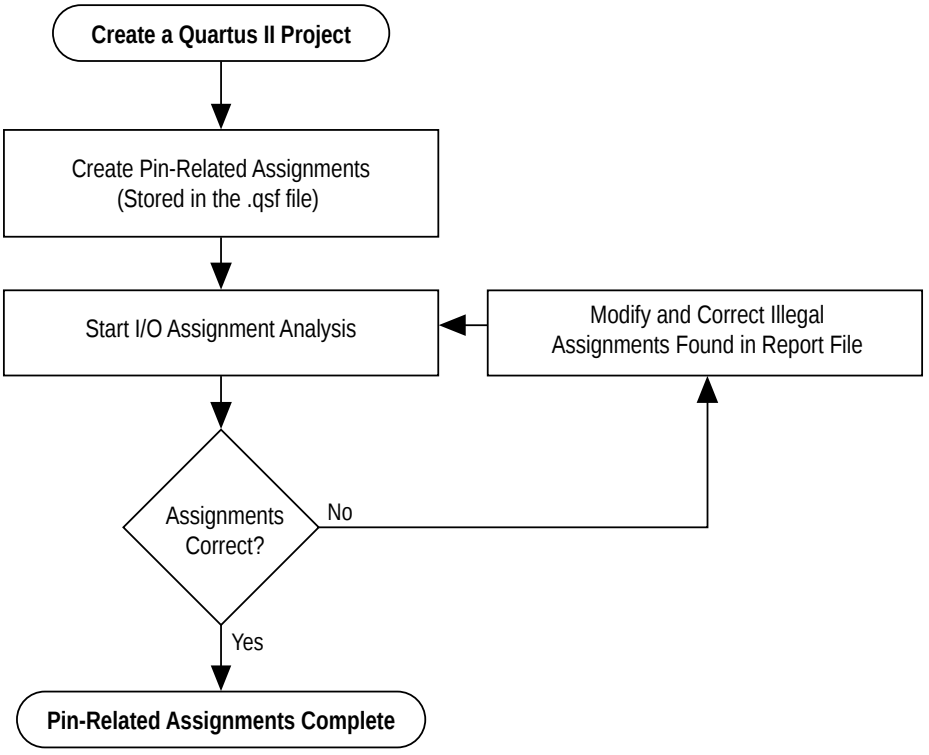
Related Information

Back-Annotating Assignments for A Project

Running 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 an 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 4-6: Assigning and Analyzing Pin-Outs without Design Files

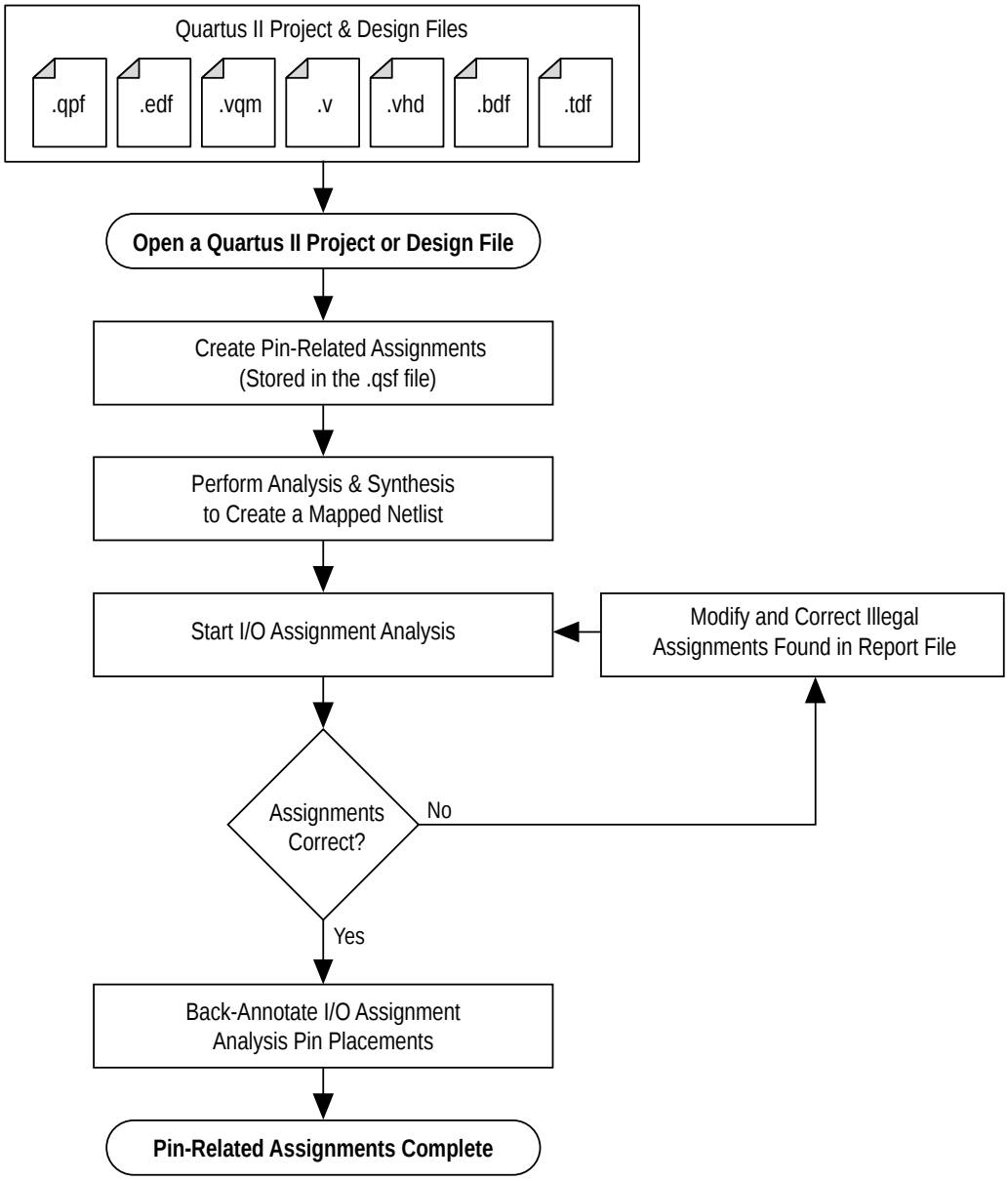


You must reserve all pins you intend to use as I/O pins, so that the Fitter can determine each pin type. After performing I/O assignment analysis, correct any errors reported by the Fitter and rerun I/O assignment analysis until all errors are corrected. A complete I/O assignment analysis requires all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

Running I/O Assignment Analysis (with Design Files)

Use I/O assignment analysis 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 designs, 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 4-7: I/O Assignment Analysis Flow



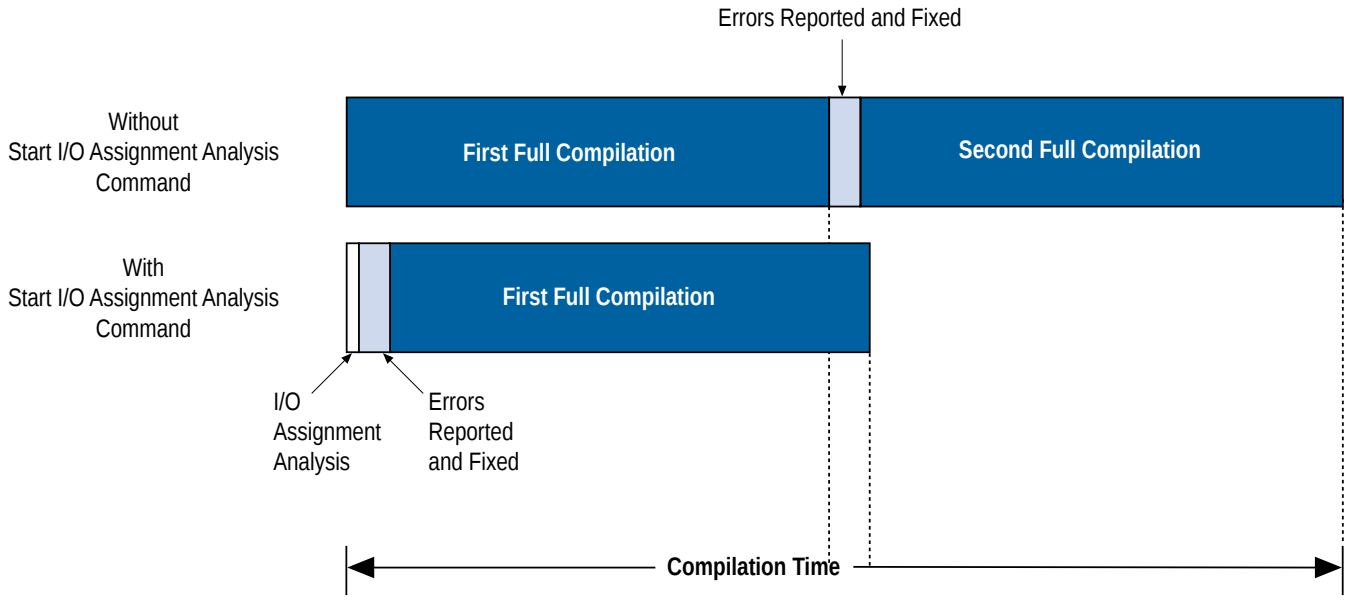
Even if I/O assignment analysis passes on incomplete design files, you may still encounter errors during full compilation. For example, you can assign a clock to a user I/O pin instead of assigning it to a dedicated clock pin, or design the clock to drive a PLL that you have not yet instantiated in the design. This occurs 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 4-8: I/O Assignment Analysis Reduces Compilation Time



Overriding Default I/O Pin Analysis

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

Each device contains a number of VREF pins, each supporting a number of 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.

This assignment is especially important for external memory interfaces. For example, consider a DDR2 interface in a Stratix II device. The device allows 30 pins in a VREF group. Each byte lane for a ×8 DDR2 interface includes one DQS pin and eight DQ pins, for a total of nine pins per byte lane. The DDR2 interface uses the **SSTL 18 Class I** VREF I/O standard. In typical interfaces, each byte lane has its own output enable. In this example, the DDR2 interface has four byte lanes. Using 30 I/O pins in a VREF group, there are three byte lanes and an extra byte lane that supports the three remaining pins. Without the **Output Enable Group** assignment, the Fitter analyzes each byte lane as an independent group driven by a unique output enable. In this worst-case scenario the three pins are inputs, and the other 27 pins are outputs violating the 20 output pin limit.

Because DDR2 DQS and DQ pins are always driven in the same direction, the analysis reports an error that is not applicable to your design. The **Output Enable Group** assignment designates the DQS and DQ pins as a single group driven by a common output enable for I/O assignment analysis. When you use the **Output Enable Group** logic option assignment, the DQS and DQ pins are checked as all input pins or all output pins and are not in violation of the I/O rules.

You can also use the **Output Enable Group** assignment to designate pins that are driven only at certain times. For example, the data mask signal in DDR2 interfaces is an output signal, but it is driven only when the DDR2 is writing (bidirectional signals are outputs). To avoid I/O assignment analysis errors, use the **Output Enable Group** logic option assignment to assign the data mask to the same value as the DQ and DQS signals.

You can also use the **Output Enable Group** to designate VREF input pins that are inactive during the time the outputs are driving. This assignment removes the VREF input pins from the VREF analysis. For example, the QVLD signal for an RLDRAM II interface is active only during a read. During a write, the QVLD pin is not active and does not count as an active VREF input pin in the VREF group. Place the QVLD pins in the same output enable group as the RLDRAM II data pins.

Related Information

The TimeQuest Timing Analyzer

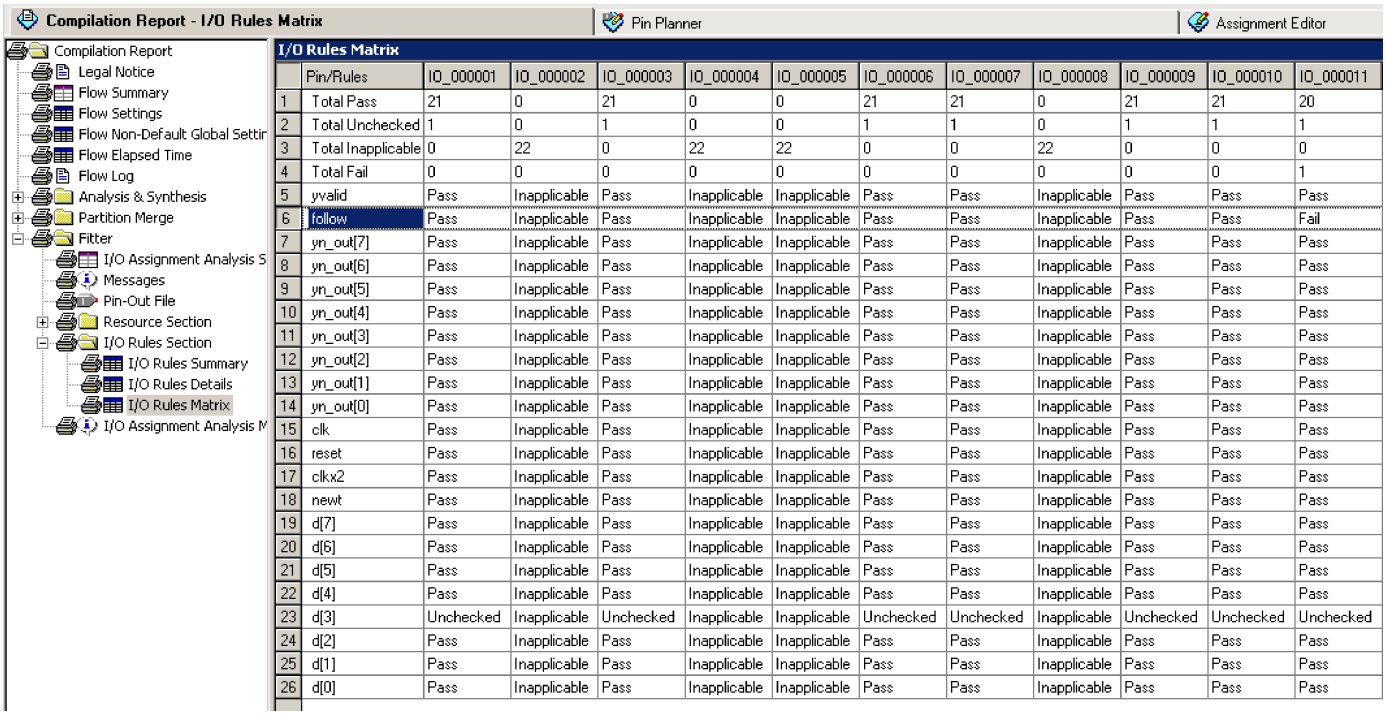
Understanding I/O Analysis Reports

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 could not be checked. A severity rating indicates the rule's importance for effective analysis. "Inapplicable" rules do not apply to the target device family.

Figure 4-9: 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

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 > TimeQuest 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 Quartus II software supports three different methods of I/O timing analysis:

Table 4-8: 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. TimeQuest 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. TimeQuest reports tCO to an I/O pin using a default or user-specified value for a capacitive load.

I/O Timing Analysis	Description
Full board routing simulation	Use Altera-provided or Quartus II software-generated IBIS or HSPICE I/O models for simulation in Mentor Graphics HyperLynx and Synopsys HSPICE.

Note: Advanced I/O timing analysis is supported only for .28nm and larger device families. For devices that support advanced I/O timing, it is the default method of I/O timing analysis. For all other devices, you must use a default or user-specified capacitive load assignment to determine t_{CO} and power measurements.

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 Quartus II software and integrating those models into HyperLynx and HSPICE simulations, refer to the*Signal Integrity Analysis with Third Party Tools* chapter in volume 2 of the *Quartus II Handbook*.

Related Information

- **Literature and Technical Documentation**
- **Altera Signal Integrity Center**
- **Signal Integrity Analysis with Third-Party Tools**

Running Advanced I/O Timing

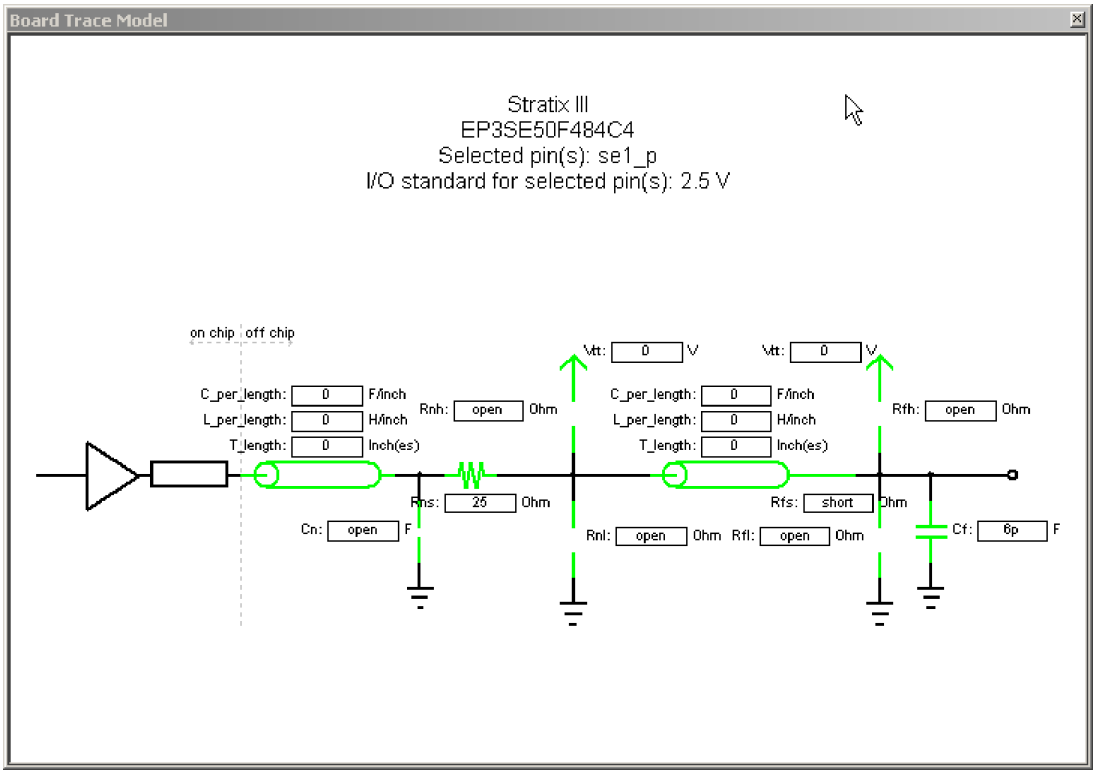
Advanced I/O timing analysis uses your board trace model and termination network specification to report accurate output buffer-to-pin timing estimates, FPGA pin and board trace signal integrity and delay values. Advanced I/O timing runs automatically for supported devices during compilation.

Understanding the Board Trace Models

The Quartus II 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.

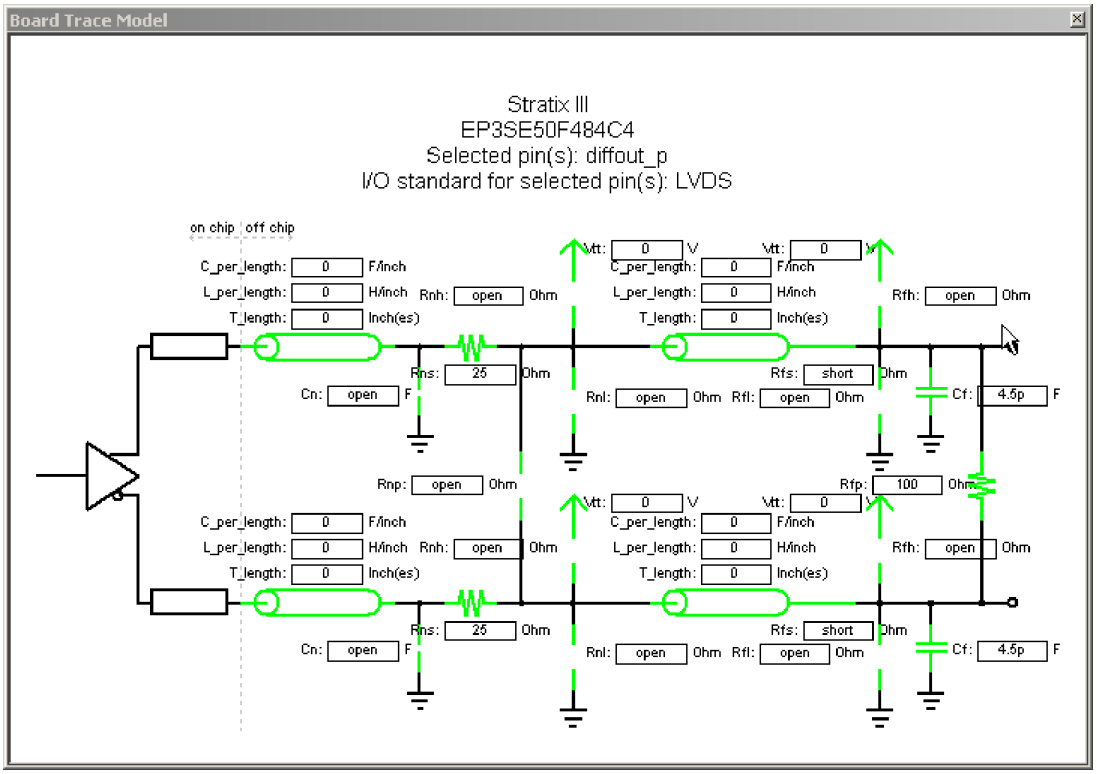
Figure 4-10: 2.5-V I/O Standard Board Trace Model



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 Quartus II software models lossless transmission lines, and does not require a transmission-line resistance value. Only distributed inductance (L) and capacitance (C) values are needed. The distributed L and C values of transmission lines must be entered on a per-inch basis, and can be obtained from the PCB vendor or manufacturer, the CAD Design tool, or a signal integrity tool, such as the Mentor Graphics Hyperlynx software.

Figure 4-11: LVDS Differential Board Trace Model



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 and the SSN Analyzer use 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** and then click **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 4-7: 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
```

Related Information

- **Using Advanced I/O Timing**
- **Board Trace Models**

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.

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.

Far-end I/O timing analysis, then 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.

Understanding Advanced I/O Timing Analysis Reports

View I/O timing analysis information in the following reports:

Table 4-9: Advanced I/O Timing Reports

I/O Timing Report	Description
TimeQuest 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 TimeQuest 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** > **TimeQuest Timing Analyzer**.

Related Information

The TimeQuest Timing Analyzer

Adjusting I/O Timing and Power with Capacitive Loading

When calculating t_{CO} and power for output and bidirectional pins, the TimeQuest analyzer and the PowerPlay 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 Quartus II 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.

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, LogicLock 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.

Analyzing Simultaneous Switching Noise

Click **Processing > Start > Start SSN Analyzer** to estimate the voltage noise for each pin in the design. The simultaneous switching noise (SSN) analysis accounts for the pin placement, I/O standard, board trace, output enable group, timing constraint, and PCB characteristics that you specify. The analysis produces a voltage noise estimate for each pin in the design. View the SSN results in the Pin Planner and adjust your I/O assignments to optimize signal integrity.

Related Information

Simultaneous Switching Noise (SSN) Analysis and Optimization

Scripting API

You can alternatively use Tcl commands to access I/O management functions, rather than using the GUI. For detailed information about specific scripting command options and Tcl API packages, type the following command at a system command prompt to view the Tcl API Help browser:

```
quartus_sh --qhelp
```

Related Information

- **Tcl Scripting**
- **Command Line Scripting**

Run I/O Assignment Analysis

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

```
execute_flow -check_ios
```

Type the following at a system command prompt:

```
quartus_fit <project name> --check_ios
```

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_map <project name>
```

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.

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`

Exclusive I/O Group

Use the following Tcl command to create an exclusive I/O group assignments:

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

Slew Rate and Current Strength

Use the following Tcl commands to create an 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

Altera Device Package Information Data Sheet

Document Revision History

The following table shows the revision history for this chapter.

Table 4-10: Document Revision History

Date	Version	Changes
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

Date	Version	Changes
July 2010	10.0.0	• Reorganized and edited the chapter • Added links to Quartus II 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 Quartus II 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

Quartus II Handbook Archive