5. I/O Management

Introduction

The process of managing I/Os for FPGA devices involves more than just fitting design pins into a package. The increasing complexity of I/O standards and pin placement guidelines are just some of the factors that influence pin-related assignments. Both I/O capabilities of the FPGA device and board layout guidelines influence pin location and other types of assignments. Therefore, it is necessary to begin I/O planning and PCB development even before starting the FPGA design.

Altera provides many resources for I/O planning. This chapter provides information on how to make pin assignments, how to enter I/O interface information in the Pin Planner, how to create I/O-based top-level HDL files, how to validate your pin assignments, and how to generate a valid pin-out file for use with third-party PCB tools. You can consult the device-specific pin connection guidelines available on the Altera® website for your board layout. You can also benefit from the Pin Advisors available in the Quartus II software.

To get updated information about the Altera resources available for I/O planning, refer to the I/O Management, Board Development Support, and Signal Integrity Analysis Resource Center on the Altera website.

For guidelines about PCB designs for Altera high-speed FPGAs, refer to AN 315: Guidelines for Designing High-Speed FPGA PCBs, and the Board Design Resource Center on the Altera website.

This chapter includes the following topics:

- “Understanding Altera FPGA Pin Terminology” on page 5–2
- “I/O Planning Overview” on page 5–5
- “Device Selection” on page 5–7
- “Early I/O Planning Using the Pin Planner” on page 5–7
- “Importing and Exporting Pin Assignments” on page 5–12
- “Creating Pin-Related Assignments” on page 5–13
- “Creating Pin Assignments Using the Pin Planner” on page 5–14
- “Creating Pin Assignments with Tcl” on page 5–21
- “Creating Pin Assignments with the Chip Planner” on page 5–22
- “Creating Pin Assignments in HDL” on page 5–22
- “Creating Pin Assignments with Low-Level I/O Primitives” on page 5–24
- “Validating Pin Assignments” on page 5–24
- “Validating Pin Assignments after Full Compilation” on page 5–39
- “I/O Timing Analysis” on page 5–40
Understanding Altera FPGA Pin Terminology

Altera FPGA devices are available in a variety of package types. To describe Altera FPGA pin terminology, this chapter uses a wire bond ball grid array (BGA) package in its examples. On the top surface of the silicon die, there is a ring of bond pads that connect to the silicon to the I/O pins. In a wire bond BGA package, the device is placed in the package and copper wires connect the bond pads to the solder balls of the package. Figure 5–1 shows a cross section of a wire bond BGA package.

For a list of all BGA packages available for each Altera FPGA device, refer to the Altera Device Package Information Data Sheet.

Figure 5–1. Wire Bond BGA

Package Pins

The pins of a BGA package are small solder balls arranged in a grid pattern on the bottom of the package. In the Quartus II software, the package pins are identified with pin numbers. The pin numbers are determined by pin locations using a coordinate system, with letters and numbers identifying the row and column of the pins, respectively.

The upper-most row of pins is labeled “A” and continues alphabetically as you move downward (Figure 5–2). The left-most column of pins is labeled “1” and continues with increments of 1 as you move to the right. For example, pin number “B4” represents row “B” and column “4.”

Figure 5–2. Row and Column Labeling

The letters I, O, Q, S, X, and Z are never used in pin numbers. If there are more rows than letters of the alphabet, the alphabet is repeated, prefixed with the letter “A.”

For more information about the pin numbers for your Altera device, refer to the device pin-out information available on the Altera website at www.altera.com.
Pads

Package pins are connected to pads located on the perimeter of the top metal layer of the silicon die (Figure 5–1). Each pad is identified by a pad ID, which is numbered starting at 0, incrementing by 1 in a counterclockwise direction (Figure 5–3).

To prevent signal integrity issues, the Quartus II software uses pin placement rules to validate your pin placements and pin-related assignments. It is important that you understand which pad locations your pins were assigned to, because some pin placement rules describe pad placement restrictions. For example, in certain devices, there is a restriction on the number of I/O pins supported by a VREF pad to ensure signal integrity. There are also restrictions on the number of pads between single-ended input or output pins and a differential pin. The Quartus II software performs pin placement analysis, and if pins are not placed according to pin placement rules, the design compilation fails and the Quartus II software reports an error.

For more information about pin placement guidelines, refer to the Selectable I/O Standards chapter in volume 1 of the appropriate device handbook.

I/O Banks

I/O pins are organized into I/O banks designed to facilitate various supported I/O standards. Each I/O bank is numbered and has its own voltage source pins, called VCCIO, to offer the highest I/O performance. Depending on the device and I/O standards for the pins within the I/O bank, the specified voltage of the VCCIO pin is between 1.5 V and 3.3 V. Each I/O bank can support multiple pins with different I/O standards that share the same VCCIO.

It is important to refer to the appropriate device handbook to determine the capabilities of each I/O bank. For example, the pins in the I/O banks on the left and right side of a Stratix® II device support high-speed I/O standards such as LVDS, whereas the pins on the top and bottom I/O banks support all single-ended I/O standards, including data strobe signaling (DQS) (Figure 5–4). Pins belonging to the same I/O bank must use the same VCCIO signal.
VREF Groups

A VREF group is a group of pins that includes one dedicated VREF pin as required by voltage-referenced I/O standards. A VREF group is made up of a small number of pins, as compared to the I/O bank, to maintain the signal integrity of the VREF pin. One or more VREF group(s) exist in an I/O bank. The pins in a VREF group share the same $V_{CCD}$ and VREF voltages.

For more information about I/O banks, VREF groups, and supported I/O standards, refer to the Architecture and Selectable I/O Standards chapters in the appropriate device handbook.
I/O Planning Overview

I/O planning of your FPGA design in Quartus II software can include:

- Selecting a device that meets your logic and I/O requirements, based on the device’s supported I/O standards, I/O bank structure, supply voltage requirements such as VREF and VDDQ, requirements in I/O banks, available pins for user I/O, power supply requirements, and more.

- Getting your design files ready. The design files contain the top-level ports or top-level interface information. If you do not have the design files, you can use the Early I/O Planning flow to generate a top-level HDL wrapper file.

- Importing any existing assignments from a Tcl script, .csv, or .qsf file.

- Creating, modifying, and completing all pin-related assignments that include pin location assignments, I/O standards, output loading assignments for output and bidirectional pins, slew rates, current strengths, and more.

- Validating your pin-related assignments while creating them by using the Live I/O Check feature, then running I/O assignment analysis, and finally running the Fitter with timing constraints.

- Generating a validated *.pin file for third-party PCB tools.

The method you use to create pin assignments depends on your requirements. If you have not yet designed the PCB, create and validate your I/O assignments in the Quartus II software, then export them to the PCB tool (Figure 5–5). This is the recommended design flow for creating I/O assignments for an FPGA design.
Figure 5–5. Quartus II Software I/O Planning Flow

Notes to Figure 5–5:

1. Use the Live I/O Check feature in the Pin Planner to validate pin assignments as you create them.
2. To create the FPGA Xchange file (.fx), on the Processing menu, point to Start and click EDA Netlist Writer. The .pin file is created at the <project_dir> level. The .fx file is created at the <project_dir/board/.../> level.
3. Your design files and constraints must be complete before you begin full compilation. To learn how to create I/O timing constraints, refer to the TimeQuest Timing Analyzer and Classic Timing Analyzer chapters in the Timing Analysis section in volume 3 of the Quartus II Handbook.
4. Refer to the Area and Timing Optimization chapter in volume 2 of the Quartus II Handbook.
If your PCB is partially designed, create your FPGA assignments in your PCB tool and import them into the Quartus II software for validation (Figure 5–6).

Currently, only the Mentor Graphics® I/O Designer PCB tool and the Cadence Allegro PCB tool are supported in this reverse I/O planning flow.

**Figure 5–6. I/O Planning Flow Using an FPGA Xchange File from a PCB Tool**

In the Quartus II software design flow, the most important step in I/O planning is to create, modify, complete, and validate pin-related assignments. The Quartus II software includes the Pin Planner and I/O Assignment Analysis to assist you in I/O planning.

**Device Selection**

Before you begin pin planning or I/O assignment analysis in the Quartus II software, refer to the device handbooks at www.altera.com to understand the I/O structure, supported I/O standards, available pins for user I/O, clocking schemes and options, and I/O bank structure for different devices. Then, choose an appropriate device from a supported device family for your design.

For more information on selecting a device in the Quartus II software, refer to *Setting Up and Running a Compilation* in Quartus II Help.

**Early I/O Planning Using the Pin Planner**

In a typical design methodology, you create design elements in a hardware description language (Verilog or VHDL) or in a schematic editor like the Quartus II Block editor.

Central to the design is a top-level file that instantiates the next level of hierarchy and includes port names and their direction. For example the Verilog HDL file shown below shows an example of a top level file listing input and output ports.
Early I/O Planning Using the Pin Planner

Top-level files for FPGA designs often contain interfaces for memory, high-speed I/O, device configuration, and debugging tools. Listing the ports in HDL or drawing them in the schematic can be extremely time-consuming.

You can use the Pin Planner to create a top-level design file if the design files for the entire project are not available or complete. The interfaces between your FPGA and other devices are typically determined and documented in design specifications. By adding those interfaces required to connect your FPGA with these other devices in the Pin Planner, you can plan your FPGA I/Os efficiently without design files, and generate a top-level module in Verilog HDL or VHDL. By importing and/or creating any Altera IP MegaCore® functions or Altera megafunctions in the Pin Planner, as well as creating or adding additional top-level I/O information, the generated top-level design file accurately anticipates the rest of the HDL to come.

The following sections describe the typical steps of the early I/O planning flow:

- “Create or Import a Megafunction or IP MegaCore Variation from the Pin Planner”
- “Connecting Nodes Before Creating Your Top-Level Design File” on page 5–9
- “Setting Up and Creating the Top-Level File” on page 5–10

**Create or Import a Megafunction or IP MegaCore Variation from the Pin Planner**

You can use the MegaWizard™ Plug-In Manager, from within the Pin Planner to create or import custom megafunctions and intellectual property (IP) cores from the perspective of device I/O. Adding interface information allows you to assign required pins without manually creating each pin individually in the Pin Planner.

You can create complex interfaces from within the Pin Planner. The megafunctions `ddio_in` and `ddio_out` in the schematic shown in Figure 5–7, are created by invoking the MegaWizard Plug In Manager tool from the Pin Planner. In the **Set Up Top Level File** window you can declare a port to be internal or external. Internal ports (gray) are connected later in the design.
Connecting Nodes Before Creating Your Top-Level Design File

Before you create a top-level design file, you must first connect the user ports, megafunction nodes, and IP MegaCore function nodes to each other and to the rest of the design.

For more information on connecting nodes from Megafunction variations, refer to Generating a Top-Level Design File Based on Pin Planner Megafunctions and User Nodes in Quartus II Help.

Adding User Nodes

You can add nodes in the Set Up Top Level File window. When you generate the top-level file in HDL, the new or additional nodes appear as ports in your HDL file. When you enter new node names in this window, the All Pins list and Groups list in the Pin Planner are also updated.

To make the user node connection between the reset signal and the megafunction reset input port shown in Figure 5–7, in the Node Name column, select reset, as shown in Figure 5–8.
You can create a top-level design file after you add or modify user ports, megafunction nodes, or IP MegaCore function nodes in your project with the Pin Planner. If the internal logic is incomplete, the top-level design file enables you to validate your I/O assignments and provides a base on which to build the rest of your design.

For more information on generating a top-level design file, refer to Create Top-Level Design File Dialog Box in Quartus II Help.

To generate a top-level design file, right-click in the Package View and click Create Top-Level Design File. You can also generate a top-level file on the File menu by pointing to Create/Update and clicking Create Top-Level Design File From Pin Planner. The Create Top-Level Design File dialog box appears. Enter a name and select an HDL format (Verilog HDL or VHDL). If the file already exists, you can choose to create a backup of the original file.

Example 5–2 shows a sample of an top-level HDL wrapper file representing the design in Figure 5–7.
The IP nodes you declared internal in the **Set Up Top-Level Design File** dialog box are declared virtual pins. The Pin Planner makes virtual pin assignments to internal nodes so that internal nodes are not assigned to device pins during compilation. These virtual pins are not shown in the All Pins list or Groups in the Pin Planner because they are not actual external ports of the design.

The top-level design file must be updated whenever changes are made to the design’s top-level ports, including any node changes made in the **Set Up Top-Level Design File** window.

After you generate the top-level file and compile the design, use I/O assignment analysis as described in “Using I/O Assignment Analysis to Validate Pin Assignments” on page 5–27 and continue with your design flow by modifying or creating pin assignments using the Pin Planner.
Importing and Exporting Pin Assignments

If you have existing pin assignments from a different Quartus II project or from third-party PCB tools, you can transfer these assignments between the Quartus II software and other tools in the following file formats: Tcl (.tcl) files, Comma Separated Value (.csv) files, Quartus II Settings Files (.qsf), FPGA Xchange (.fx) files, and Pin-Out (.pin) files.

Tcl Scripts and .csv Files

You can export and import pin-related assignments contained in .csv files and Tcl scripts. A .csv file consists of a row of column headings followed by rows of comma-separated data. The row of column headings in the exported file is in the same order and format as the columns displayed in the All Pins list in the Pin Planner. Do not modify the row of column headings if you plan to import the .csv file later.

When you export pin-related assignments as Tcl commands in a Tcl script, you create a script which you can later run to add these assignments as part of a scripted compilation flow.

For more information on importing and exporting pin related assignments as Tcl scripts and .csv files, refer to Assigning Pins in Quartus II Help.

For more information about Quartus II scripting support, including examples, refer to the Tcl Scripting and Command-Line Scripting chapters in volume 2 of the Quartus II Handbook.

The All Pins list in the Quartus II Pin Planner, the Pin category in the Quartus II Assignment Editor, and the device .pin files all display detailed properties about each pin of the device, in addition to the pin name and pin number. The device .pin files are available on the Altera website at www.altera.com.

Quartus II Settings Files

When you make pin assignments with the Pin Planner or the Assignment Editor in the Quartus II software, all your pin-related assignments are written to the Quartus II Settings File (.qsf). You can also export pin-related assignments to a .qsf file. The pin-related assignments, and all other design assignments, are stored as Tcl commands in the .qsf file.

For more information about .qsf files, refer to the Managing Quartus II Projects chapter in volume 2 of the Quartus II Handbook.

FPGA Xchange File

An .fx file contains device and pin-related information that allows you to transfer information between the Quartus II software and the Mentor Graphics I/O Designer software. Importing an .fx file into the I/O Designer software requires both the .fx file and *.pin file produced from Quartus II software. However, the Quartus II software requires only the .fx file to import back from the I/O Designer.
To learn more about the I/O Designer and the DxDesigner interface and support, refer to Mentor Graphics PCB Tools Support chapter in volume 2 of the Quartus II Handbook.

To import or export pin-related assignments in .qsf files and .fx files, refer to Importing and Exporting Assignments in Quartus II Help.

### .pin Files

A .pin file is an ASCII text file containing pin locations and other pin information. Use the .pin file to transfer your project’s pin information into third-party PCB tools for board development. Table 5–1 describes the columns in a .pin file.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin Name/Usage</td>
<td>The name of a design pin, ground, or power</td>
</tr>
<tr>
<td>Location</td>
<td>The pin number of the location on the device package</td>
</tr>
<tr>
<td>Dir</td>
<td>The direction of the pin</td>
</tr>
<tr>
<td>I/O Standard</td>
<td>The name of the I/O standard to which the pin is configured</td>
</tr>
<tr>
<td>Voltage</td>
<td>The voltage level that is required to be connected to this pin</td>
</tr>
<tr>
<td>I/O Bank</td>
<td>The I/O bank number that the pin belongs to</td>
</tr>
<tr>
<td>User Assignment</td>
<td>Y or N indicating if the location assignment for the design pin was user assigned (Y) or assigned by the Fitter (N)</td>
</tr>
</tbody>
</table>

For more information about Pin Name/Usage, refer to the Device Pin-Out tables for the targeted device, available on the Altera website at www.altera.com.

### Creating Pin-Related Assignments

A pin-related assignment is any assignment applied to a top-level pin. For example, a pin location assignment assigns a top-level port or node to a pin number (location) on the targeted device. Other examples of pin-related assignments include assigning an I/O standard, assigning current drive strength, or assigning a slew rate to a pin.

If when making pin assignments you do not have complete information for all the top-level pins, you can reserve certain device pins to temporarily represent your top-level design I/O pins until the I/O pins are defined in your design files. Reserved pins are intended for future use but do not currently perform a function in your design. Reserved pins require a unique pin name and pin location. Using reserved pins as place holders for future design pins increases the accuracy of I/O assignment analysis.

The Quartus II software offers many tools and features for creating reserved pins and other pin-related assignments (Table 5–2). Each tool and feature is described in more detail in the following sections.
Creating Pin Assignments Using the Pin Planner

The Pin Planner is an interface for creating and editing pin-related assignments (Figure 5–9). With the Pin Planner, you can identify I/O banks, VREF groups, and differential pin pairings to help you through the I/O planning process.

When planning your I/Os, it can be cumbersome to try to correlate pin numbers with their relative location on the package and their pin properties. The Pin Planner provides an intuitive graphical representation of the targeted device, also known as the Package View, that makes it easy to plan your I/Os, create reserved pins, and make pin location assignments. When deciding on a pin location, use the Pin Planner to gather information about available resources, as well as the functionality of each individual pin, I/O bank, and VREF group. You can assign locations to design I/O nodes by dragging and dropping each node onto a pin in the Package View.

Table 5–2. Overview of Quartus II Tools and Features to Create Pin-Related Assignments

<table>
<thead>
<tr>
<th>Feature</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin Planner</td>
<td>Make pin location assignments to one or more node names by dragging and dropping unassigned pins into the Package View</td>
</tr>
<tr>
<td></td>
<td>Edit pin location assignments for one or more node names by dragging and dropping groups of pins within the Package View</td>
</tr>
<tr>
<td></td>
<td>Visually analyze pin resources in the Package View</td>
</tr>
<tr>
<td></td>
<td>Display I/O banks and VREF groups</td>
</tr>
<tr>
<td></td>
<td>View the function of package pins using the pin legend</td>
</tr>
<tr>
<td></td>
<td>Make correct pin location decisions by referring to the Pad View window</td>
</tr>
<tr>
<td></td>
<td>Create, import, and edit megafunctions and IP MegaCore functions for early I/O planning</td>
</tr>
<tr>
<td></td>
<td>Generate a top-level wrapper file without design files based on early I/O assignments</td>
</tr>
<tr>
<td></td>
<td>Configure board trace models of selected pins for use in “board-aware” signal integrity reports generated with the Enable Advanced I/O Timing option</td>
</tr>
<tr>
<td>Tcl Scripts</td>
<td>Create any pin-related assignments for multiple pins</td>
</tr>
<tr>
<td></td>
<td>Store and reapply all pin-related assignments with Tcl scripts</td>
</tr>
<tr>
<td></td>
<td>Make assignments from the command line</td>
</tr>
<tr>
<td>Chip Planner</td>
<td>Create and change pin locations by dragging and dropping pins into the floorplan</td>
</tr>
<tr>
<td></td>
<td>Make correct pin location decisions by referring to the pad ID number and spacing</td>
</tr>
<tr>
<td></td>
<td>Display I/O banks, VREF groups, and differential pin pairing information</td>
</tr>
<tr>
<td>Synthesis Attributes</td>
<td>Embed pin-related assignments using attributes in the design files to pass assignments to the Quartus II software</td>
</tr>
</tbody>
</table>

Creating Pin Assignments Using the Pin Planner

The Pin Planner is an interface for creating and editing pin-related assignments (Figure 5–9). With the Pin Planner, you can identify I/O banks, VREF groups, and differential pin pairings to help you through the I/O planning process.

When planning your I/Os, it can be cumbersome to try to correlate pin numbers with their relative location on the package and their pin properties. The Pin Planner provides an intuitive graphical representation of the targeted device, also known as the Package View, that makes it easy to plan your I/Os, create reserved pins, and make pin location assignments. When deciding on a pin location, use the Pin Planner to gather information about available resources, as well as the functionality of each individual pin, I/O bank, and VREF group. You can assign locations to design I/O nodes by dragging and dropping each node onto a pin in the Package View.
When you select a pin in one view the Pin Planner highlights the pin in all of the other views. For example, if you select a pin in the Package View of the Pin Planner, the corresponding pad in the Pad View window is highlighted. If the pin has an assigned node name, the node name in the All Pins list and the Groups list is highlighted.

For information on how to create pin assignments using the Pin Planner, refer to Assigning Pins in Quartus II help.

**Pin Migration**

Selecting a migration device allows you to either vertically migrate to a different density, while using the same package, or to migrate between packages with different densities and ball counts. If a migration device is selected, the Pin Planner shows only pins that are available in both the current device and the migration device.

The Pin Migration View in the Pin Planner shows the pins that change function in a migration device if you select one or more migration devices for your project.
The Pin Migration View helps you identify the difference in pins that can exist between migration devices. For example, in Figure 5–10, the highlighted pin AC24 existed in the original EP2S30 device selected, but does not exist in one of the migration devices. Therefore, the migration result is a No Connect (NC). If you select your migration devices after you have successfully compiled a design and these migration devices have certain differences, an error occurs if you try to recompile your design.

Figure 5–10. Pin Migration View

There may be assignments for I/O nodes in your original design, that do not have corresponding pins in a migration device. When you select migration devices early in the design process, only the pins that exist in all migration devices are available in the Pin Planner. If you select migration devices later in your design cycle for which the pin assignment cannot be honored, an error occurs when you try to recompile.

Notice that for PIN_AC23, the Migration Result for Pin Function is not an NC but a voltage reference VREFB1N2. This is because it is an NC in one of the migration devices, but a VREFB1N2 in the other migration device. In this type of result, VREF standards have a higher priority than an NC. You might not be making use of that pin for a port connection in your design, but 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 VCC 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. In this case, to facilitate migration, you can connect these pins to \( V_{cc} \) or GND in your original design, because there the pins are not physically connected to the smaller die.

For more information about migration, refer to *AN90: SameFrame Pin-Out Design for FineLine BGA Packages*. For more information about designing for HardCopy II devices, refer to the *Quartus II Support for HardCopy Series Devices* chapter in volume 1 of the *Quartus II Handbook*. For more information about the Pin Migration view in the Pin Planner, refer to *About Device Migration* in Quartus II Help.

**Using the Pin Finder to Find Compatible Pin Locations**

As FPGA pin-counts and I/O capabilities continue to increase, it becomes more difficult to understand the capabilities of each I/O pin and to correctly assign your design I/Os. To help you address this problem, the Pin Planner highlights all pins that match the list of conditions that you enter.

For more information about the Pin Finder, refer to *Pin Finder Dialog Box* in Quartus II Help.

**SSN Visualization View**

In the Quartus II software version 9.0 and later, you can view the simultaneous switching noise (SSN) visualization in the Pin Planner by right-clicking in the Package View and selecting *Show SSN Analyzer results*. The SSN Analyzer estimates SSN for pins of your FPGA device. Refer to the Quartus II Help to find out the devices supported for SSN analysis. The integration of the SSN Analyzer and Pin Planner in the Quartus II software allows you to perform SSN analysis while planning your I/O pins.

For more information about the SSN Analyzer, refer to the *Simultaneous Switching Noise (SSN) Analysis and Optimization* chapter in volume 2 of the *Quartus II Handbook*.

**Creating Exclusive Group Assignments**

In the Quartus II software version 9.0 and later, you can create exclusive groups comprised of pins by using the following assignment:

\[ \text{set_instance_assignment} -\text{name} "\text{EXCLUSIVE_IO\_GROUP}" -\text{to} \text{pin} \]

When you create exclusive I/O group(s) in your FPGA design and use the Quartus II software to map the signals onto device pins, the Fitter does not place the I/O pins belonging to one exclusive group in an I/O bank if the pins belong to another exclusive I/O group. To understand this, consider an example in which you have a set of signals assigned exclusively to a group called *group_a*, and another set of signals assigned to *group_b*. In both exclusive groups you might have pins with different I/O standards. When you create these groups, the Quartus II software maps the pins of both groups in such a way that they are placed in different I/O banks.
Assigning Locations for Differential Pins

When you assign the top-level pins of your design as differential pins in the Pin Planner, it is important to understand the process for creating differential pins in your design. If your design has top-level pins that are single ended, you can use the Pin Planner to assign the required differential standard to those pins. The Quartus II software automatically creates the negative pin of the differential standard. For example, if you have a top-level pin defined as `lvds_in` in your design, the Quartus II software creates a `lvds_in(n)` pin when you assign a differential standard to `lvds_in`, as shown in Figure 5–11.

**Figure 5–11.** Creating a Differential Pin Pair in the Pin Planner

For the Cyclone® III, Stratix III, and Stratix IV device families, you can use low-level I/O differential primitives to define both positive and negative pins of a differential pair in your design's HDL code.

For more information about supported I/O primitives and details about their assignments, refer to the Designing with Low-Level Primitives Users Guide or Primitives in Quartus II Help.

When you use low-level I/O primitives to define various pin-related assignments for I/Os, the assignments are honored after you perform full compilation. These assignments are not shown in the Pin Planner after analysis and synthesis. After a full compilation, you can populate these assignments in the Pin Planner by back-annotating pin assignments. To back annotate your pin assignments, on the Assignments menu, click Back-Annotate Assignments.

To identify and assign differential pins using the Pin Planner, perform the following steps:

1. On the View menu, click Show Differential Pin Pair Connections.
   
   A red line connects the positive and negative pins of the differential pin pairing. The positive and negative pins are labeled in the Package View with the letters “p” and “n”, respectively (Figure 5–12).

2. Use the tool tips to identify LVDS-compatible pin locations by holding the mouse pointer over a differential pin in the Package View (Figure 5–12).
Creating Pin Assignments Using the Pin Planner

The tool tip shows the design pin name and pin number, as well as its general and special functions.

For more information about the general and special functions of pins displayed by the tool tip, refer to Pin-Out Files for Altera Devices.

3. From the All Pins list or Groups list, drag-and-drop the selected pin of the differential pair to a differential pin location in the Package View.

   Pay attention when you drag the positive or negative pin of the differential pin pair to the Package View. Connect the positive pin to the “p” pin on the device and negative pin to the “n” pin on the device. An error appears in the Quartus II software if you do not connect the differential pins properly.

   Optionally, before you drag-and-drop your pins, you can use the Pin Finder to locate pin locations that support your selected pins. When creating a query in the Pin Finder, add an assignment condition set to Unassigned and an I/O standard condition set to your differential I/O standard.

   The Quartus II software recognizes the negative pin as part of the differential pin pair assignment. The location assignment for the negative pin pair is written to the .qsf file. However, the assignment I/O standard is not entered in the .qsf file for the negative pin of the differential pair.

   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 targeted device. Single-ended pins that feed a PLL and are assigned to the negative clock pin in the targeted device cause the design to not fit.

Changing the Slew Rate and Current Drive Strength in Pin Planner

Current strength is an important property of an I/O pin. It affects the integrity of the signal going out of the device. You can set the current strengths as part of the I/O planning flow in the Pin Planner. You can also change the current strength after the compilation of your design. In the Pin Planner’s All Pins list, you can view or edit the current strengths for each of the output and bidirectional pins in the Current Strength column. If the Current Strength column is not visible in All Pins list, right-click in the All Pins list and use the Customize Columns dialog box to add the Current Strength column. The settings you make in Pin Planner are honored during Live I/O Check, I/O Assignment analysis, and full compilation.
To make changes to the current drive strength after compilation has finished, you can perform engineering change orders (ECOs). To perform ECOs, use the Resource Property Editor (RPE) tool in the Quartus II software. Performing ECO compilation does not recompile the whole design, but compiles only the changes.

For more information about ECOs, refer to the Quartus II Help or the Engineering Change Management with the Chip Planner chapter in volume 2 of the Quartus II Handbook.

The slew rate is another important property of an I/O pin that affects the outgoing signal integrity of the device. Slew rate options are not supported for all device families. To learn more about the supported families for slew rates, refer to the specific device handbooks. You can set the slew rate as part of the I/O planning flow in the Pin Planner. You can also change the slew rate setting after compiling your design.

In the Pin Planner’s All Pins list, you can view or edit the slew rate for each of the output and bidirectional pins in the Slew Rate column. If the Slew Rate column is not visible in the All Pins list, right-click in the All Pins list and use the Customize Columns dialog box to add the Slew Rate column. The settings you make in Pin Planner are honored during Live I/O Check, I/O Assignment Analysis, and full compilation. To change the slew rate after compilation has finished, you can perform an ECO by using the RPE tool in the Quartus II software. Performing ECO compilation does not perform place-and-route on the whole design, but only on the changes.

You can also set the current drive strength and slew rate settings by using the following Tcl assignments in the .qsf file:

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

For more information about the effect of I/O settings on signal integrity on the board for Stratix III devices, refer to AN 476: Impact of I/O Settings on Signal Integrity in Stratix III Devices.

**I/O Error Checking Capability**

The Pin Planner has basic pin placement checking capability, preventing pin placements that violate the fitting rules. The following checks are performed by the Pin Planner as you make pin-related assignments:

- An I/O bank or VREF group is an unassignable location if there are no available pins in the I/O bank or VREF group.
- The negative pin of a differential pair is unassignable if the positive pin of the differential pair has been assigned with a node name with a differential I/O standard.
- Dedicated input pins (for example, dedicated clock pins) are an unassignable location if you attempt to assign an output or bidirectional node name.
- Pin locations that do not support the I/O standard assigned to the selected node name become unassignable.
- All nodes in the same VREF group must have the same VREF voltage. Apply this only to HSTL- and SSTL-type I/O standards.
After creating a pin location, the Location, I/O Bank, and VREF Group columns are populated in both the All Pins list and the Group list. In the Package View, the occupied pins are filled with a dark brown color.

For more information about assignment analysis, refer to “Using I/O Assignment Analysis to Validate Pin Assignments” on page 5–27. To display live information about the warnings and errors in your pin-related assignments, enable the live I/O check feature in the Quartus II software. For more information about the live I/O check feature, refer to the section “Using the Live I/O Check Feature to Validate Pin Assignments” on page 5–25.

Displaying and Accepting Fitter Placements

The Fitter provides optimal placement to unassigned pins based on design constraints when you perform a compilation or an I/O Assignment Analysis. To display these pins, on the View menu, point to Show and click Show Fitter Placements. When you choose Show Fitter Placements, the Fitter-placed pins are shown in green in the Package View. You can create a copy of the Fitter placements in your project .qsf file using the Back-Annotate Assignments command.

Altera recommends you use the Pin Planner to create and edit pin-related assignments. However, you might find some of the other tools provided for use with the Quartus II software to be useful for working with pin-related assignments. The following sections describe these tools.

Creating Pin Assignments with Tcl

You can use Tcl to create pin-related assignments as part of a script-based compilation flow. To run a Tcl script with the Quartus II software, type the following command at a system prompt:

```
quartus_sh -t my_tcl_script.tcl
```

You can also type individual Tcl commands into the Tcl console window. To use the Tcl console, on the View menu, point to Utility Windows and click Tcl Console. In the Tcl Console window, type your Tcl commands. Example 5–3 shows a list of Tcl commands that creates pin-related assignments to the input pin address[10].

Example 5–3. Tcl Commands to Create Pin-Related Assignments

```
set_location_assignment PIN M20 -to address[10] -comment"Address pin to Second FPGA"
set_instance_assignment -name IO_STANDARD "2.5 V" -to address[10]
set_instance_assignment -name CURRENT_STRENGTH_NEW "MAXIMUM CURRENT" -to address[10]
```

When you make an assignment in the Assignment Editor or the Pin Planner, display the equivalent Tcl command in the Messages window by performing the following steps:

1. On the Tools menu, click Options. The Options dialog box appears.
2. In the Category list, select Assignment Editor. The Assignment Editor page opens.
3. Turn on Echo Tcl Commands.
4. Click OK.
For more information about using Tcl scripts to create pin-related assignments, refer to the Tcl Scripting chapter in volume 2 of the Quartus II Handbook and to the Quartus II Scripting Reference Manual.

Creating Pin Assignments with the Chip Planner

The floorplan of the device shows the pins in the same order as the pads of the device. Understanding the relative distance between a pad and related logic can help you meet your timing requirements. You can view the floorplan of the device in the Chip Planner and determine the distances between user I/O pads and V_{CC}, GND, and VREF pads to avoid signal integrity issues.

You can create a pin location assignment by selecting a pin and selecting a desired location. To do this, perform the following steps:

1. To open the Chip Planner, on the Tools menu, click Chip Planner (Floorplan & Chip Editor).
2. On the View Chip Planner, point to Utility Windows and click Node Finder. The Node Finder dialog box appears.
3. In the Filter list, select Pins: all and click List to see all the nodes in the design.
4. Select a node from the Nodes Found list and drag the selection into a pin location in the floorplan.

For more information about pin placement guidelines, refer to the Selectable I/O Standards chapter of the appropriate device handbook. For more information about supported device families in the Chip Planner, refer to the Engineering Change Management with the Chip Planner chapter in volume 2 of the Quartus II Handbook.

Creating Pin Assignments in HDL

You can use synthesis attributes or I/O primitives to embed pin-related assignments in your HDL code directly. When you analyze and synthesize your HDL code, the information in the HDL code is converted into appropriate assignments. There are two ways to specify pin related assignments using HDL:

- Using 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 setting their parameters or attributes

The following sections explain how to use synthesis attributes and I/O primitives for your top-level pins.

Synthesis Attributes

Synthesis attributes allow you to embed assignments in your HDL code. The Quartus II software reads these synthesis attributes and translates them into assignments. The Quartus II integrated synthesis supports chip_pin, useioff, and altera_attribute synthesis attributes.

For more information about integrated synthesis, refer to the Quartus II Integrated Synthesis chapter in volume 1 of the Quartus II Handbook.
For synthesis attributes support by third-party synthesis tools, contact your vendor.

**chip_pin and useioff**

You can use the chip_pin and useioff synthesis attributes to embed pin location and fast output and input register assignments, respectively. For all other assignments, including pin-related assignments, use the altera_attribute synthesis attribute as discussed in the “altera_attribute” section.

**Example 5–4** and **Example 5–5** embed a location and fast input assignment into both a Verilog HDL and VHDL design file using the chip_pin and useioff synthesis attributes.

**Example 5–4. Verilog HDL Example**

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

**Example 5–5. VHDL Example**

```vhdl
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;
```

**altera_attribute**

To create other pin-related assignments, use the altera_attribute attribute. The altera_attribute attribute is understood only by Quartus II integrated synthesis and supports all types of instance assignments. **Example 5–6** and **Example 5–7** use altera_attribute to embed the fast input register and I/O standard assignments into both a Verilog HDL and a VHDL design file.

**Example 5–6. Verilog HDL Example**

```verilog
input my_pin1 /* synthesis altera_attribute = "-name FAST_INPUT_REGISTER ON; -name IO_STANDARD \"2.5 V\" " */;
```
In Quartus II software version 8.0 and later, the pin-related assignments made using synthesis attributes are shown in Pin Planner’s All Pins list and Package views. When you modify or delete these pin assignments in the Pin Planner, you get an informational message suggesting that the pin-related assignments have been changed. If you recompile your project, your pin-related assignment in the Pin Planner, which is contained in your .qsf file, has precedence over the assignments you made using synthesis attributes in your HDL file.

For detailed information about synthesis attributes and their usage syntax, refer to the Quartus II Integrated Synthesis chapter in volume 1 of the Quartus II Handbook or the Quartus II Help.

Creating Pin Assignments with Low-Level I/O Primitives

You can make pin related assignments for your top-level nodes using low-level primitives, which allow you to create pin location assignments, and set I/O standards, current drive strengths, slew rates, and on-chip termination (OCT).

For more information about using low-level I/O primitives in your design, refer to Designing with Low-Level Primitives User Guide or the Quartus II Help.

The pin-related assignments made using primitives do not appear in the Pin Planner. During and after pin-related assignments creation, you must validate your pin-related assignments using the Enable Live I/O Check option and running I/O Assignment Analysis.

Validating Pin Assignments

The Quartus II software includes built-in I/O rules to guide you in pin placement. The Quartus II software checks your pin-related assignments against these rules during pin planning. You must validate all pin-related assignments in your design. You can enable the live I/O check feature and must use I/O Assignment Analysis to validate pin-related assignments against the predefined I/O rules encoded in the Quartus II software. To fully validate these assignments against all the I/O timing checks, you must perform full compilation.

Example 5–7. VHDL Example

```vhdl
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 the Live I/O Check Feature to Validate Pin Assignments

In Quartus II software version 7.2 and later, the live I/O check feature provides live I/O rules checking capability. When the live I/O check feature is enabled, pin-related assignment error and warning messages appear immediately in the Quartus II Messages window as you create pin-related assignments in the Pin Planner. This feature enhances your productivity by showing you warnings and errors as you create pin-related assignments, before you proceed to the next step in your design flow.

The most basic I/O rules are the I/O buffer rules. The I/O buffer rules checked by the live I/O Check feature include:

- $V_{CCIO}$ voltage compatibility rules
- $V_{REF}$ voltage compatibility rules
- Electromigration (current density rules)
- Simultaneous Switching Output (SSO) rules
- I/O properties compatibility rules such as drive strength compatibility, I/O standard compatibility, PCI_IO clamp diode compatibility, and I/O direction compatibility

An additional category of I/O rules is the set of I/O system rules. These rules can be checked only after you generate a synthesized (mapped) netlist of your design. The I/O system rules are checked when you perform I/O assignment analysis as described in “Using I/O Assignment Analysis to Validate Pin Assignments” on page 5–27.

You can enable or disable the live I/O check feature at any time. By default, the live I/O check feature is turned off.

To enable or disable the live I/O check feature in the Quartus II user interface:

1. Verify the Pin Planner tool in the Quartus II software is active.
2. In the Quartus II View menu, select Live I/O Check, or, in the Pin Planner, click on the Live I/O Check icon.

While the live I/O check feature is enabled, the Quartus II software immediately checks whether your new pin-related assignments and revisions pass the basic I/O buffer rules. The detailed messages are printed in the Messages window of the Quartus II software and shown in Package View (Figure 5–13).
The Live I/O Check Status window displays the total numbers of errors and warnings while you create and edit pin-related assignments. To open the Live I/O Check Status window, shown in Figure 5–14, in the Quartus II View menu, click **Live I/O Check Status**.

For details about a specific message, refer to the Quartus II Help.
Though the live I/O check feature checks all the basic I/O buffer rules, you must run I/O assignment analysis to validate your pin-related assignments against the complete set of I/O system rules. All rules including the basic I/O buffer rules and I/O system rules can be found in Table 5–3 on page 5–37 and Table 5–4 on page 5–38.

Using I/O Assignment Analysis to Validate Pin Assignments

This section describes a design flow that includes making and analyzing pin assignments with the Start I/O Assignment Analysis command in the Quartus II software during and after the development of your HDL design.

The Start I/O Assignment Analysis command allows you to check your I/O assignments early in the design process. With this command, you can check the legality of pin assignments before, during, or after you compile your design. If design files are available, you can use this command to perform more thorough legality checks on your design’s I/O pins and surrounding logic. These checks include proper reference voltage pin usage, valid pin location assignments, and acceptable mixed I/O standards.

The Start I/O Assignment Analysis command can be used for designs that target Stratix series, Cyclone series, and MAX® II device families.

I/O Assignment Analysis Design Flows

The I/O assignment analysis design flows depend on whether your project contains design files. The following examples show two different circumstances in which I/O Assignment Analysis can be used:

- Use the flow shown in Figure 5–15 on page 5–29 if the board layout must be complete before starting the FPGA design. This flow does not require design files and checks the legality of your pin assignments.

- With a complete design, use the flow shown in Figure 5–17 on page 5–31. This flow thoroughly checks the legality of your pin assignments against any design files provided.

Each flow involves creating pin assignments, running analysis, and reviewing the report file.
You should run the analysis each time you add or modify a pin-related assignment. You can use the **Start I/O Assignment Analysis** command frequently because it completes quickly.

The analysis checks pin assignments and surrounding logic for illegal assignments and violations of board layout rules. For example, the analysis checks whether your pin location supports the assigned I/O standard, current strength, supported VREF voltages, and whether a PCI diode is permitted.

Along with the pin-related assignments, the **Start I/O Assignment Analysis** command also checks blocks that directly feed or are fed by resources such as a PLLs, LVDS, or gigabit transceiver blocks.

### I/O Assignment Analysis without Design Files

During the early stages of developing an FPGA device, board layout engineers might request preliminary or final pin-outs. It is time consuming to manually check whether the pin-outs violate any design rules. Instead, use the **Start I/O Assignment Analysis** command to quickly perform basic checks on the legality of your pin assignments.

Without a complete design, the analysis performs limited checks and cannot guarantee that your assignments do not violate design rules.

The **I/O Assignment Analysis** command can perform limited checks on pin assignments made in a Quartus II project that has a device specified, but might not yet include any HDL design files. For example, you can create a Quartus II project with only a target device specified and create pin-related assignments based on circuit board layout considerations that are already determined. Even though the Quartus II project does not yet contain any design files, you can reserve input and output pins and make pin-related assignments for each pin using the Pin Planner or Assignment Editor. After you assign an I/O standard to each reserved pin, run the I/O Assignment Analysis to ensure that there are no I/O standard conflicts in each I/O bank. **Figure 5–15** shows the work flow for assigning and analyzing pin-outs without design files.
To assign and analyze pin-outs using the Start I/O Assignment Analysis command without design files, perform the following steps:

1. In the Quartus II software, create a project.
2. Use the Pin Planner or a Tcl script to create pin locations and related assignments. For I/O Assignment Analysis to determine the type of pin, you must reserve your I/O pins. To create a reserved pin in the Pin Planner PackageView, right-click an available pin, point to Reserve and click one of the available configurations.

   If you make pin-related assignments in Mentor Graphics I/O Designer software, you can import an .fx file into the Quartus II software.

3. To start the analysis, on the Processing menu, point to Start and click Start I/O Assignment Analysis.

   For information about using a Tcl script or command prompt to start the analysis, refer to “Scripting Support” on page 5–48.

4. View the messages in the Compilation Report window, Fitter report file (<project name>.fit.rpt), or in the Messages window.
5. Correct any errors and violations reported by the I/O Assignment Analysis. Repeat steps 1 through 5 until all of the errors are corrected.

I/O Assignment Analysis with Design Files

During a full compilation, the Quartus II software does not report illegal pin assignments until the Fitter stage. To validate pin assignments sooner, run the Start I/O Assignment Analysis command after performing analysis and synthesis and before performing a full compilation. Typically, the analysis runs quickly. Figure 5–16 shows the benefits of using the Start I/O Assignment Analysis command.
The rules that are checked by the I/O assignment analysis depend on the completeness of the design. With a complete design, the Start I/O Assignment Analysis command thoroughly checks the legality of all pin-related assignments. With a partial design, which can be just the top-level wrapper file, the Start I/O Assignment Analysis command checks the legality of those pin-related assignments for which it has enough information.

For example, you might 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 has not yet been instantiated in the design. Because the Start I/O Assignment Analysis command does not account for the logic that the pin drives, it is not able to check that only a dedicated clock input pin can drive the clock port of a PLL.

To obtain better coverage, analyze as much of the design as possible, especially logic that connects to pins. For example, if your design includes PLLs or LVDS blocks, you should include these MegaWizard Plug-In Manager-generated files in your project for analysis (Figure 5–17).
To assign and analyze pin-outs using the **Start I/O Assignment Analysis** command with design files, perform the following steps:

1. Create a project including your design files.
2. Create pin-related assignments with the Pin Planner or Assignment Editor.
   
   You can also create pin-related assignments by importing them from a `.csv` or `.fx` file, executing Tcl commands, or editing the `.qsf` file directly. On the Processing menu, point to **Start** and click **Start Analysis & Synthesis** to generate an internal mapped netlist.

   For information about using a Tcl script or the command prompt to start the analysis, refer to “Scripting Support” on page 5–48.

3. On the Processing menu, point to **Start** and click **Start I/O Assignment Analysis** to start the analysis.
4. View the messages in the Compilation Report or in the Messages window.
5. Use the Pin Planner or Assignment Editor to correct any errors and violations reported.
6. Use the **Start I/O Assignment Analysis** command until all the errors are corrected.
Inputs for I/O Assignment Analysis

The Start I/O Assignment Analysis command reads the following inputs:

- Internal mapped netlist
- .qsf file

The internal mapped netlist is used when you have a partial or complete design. The .qsf file is always used to read all pin-related assignments for analysis.

Generating a Mapped Netlist

The Start I/O Assignment Analysis command uses a mapped netlist, if available, to identify the pin type and the surrounding logic. The mapped netlist is stored internally in the Quartus II software database.

To generate a mapped netlist, on the Processing menu, point to Start and click Start Analysis & Synthesis.

To use the quartus_map executable to run analysis and synthesis, type the following command at a system command prompt:

```
quartus_map <project name>
```

Creating Pin-Related Assignments

The I/O Assignment Analysis command reads a .qsf file containing all of your pin-related assignments. These pin-related assignments include pin settings such as I/O standards, drive strength, and location assignments. The following sections highlight some of the location assignments you can make.

Reserving Pins

If you do not have any design files, you can still reserve pin locations and create pin-related assignments. Reserving pins is necessary so that the Start I/O Assignment Analysis command has information about the pin and the pin type (input, output, or bidirectional) to correctly analyze the pins.

To reserve a pin, on the Assignments menu, click Assignment Editor. In the Category list, click Pin to open the Pin assignment category. Double-click the cell in the Reserved column that corresponds to the pin that you want to reserve. Use the drop-down arrow to select from the reserved pin options (Figure 5–18).

Figure 5–18. Reserving an Input Pin with the Assignment Editor

For more information about using the Assignment Editor, refer to the Assignment Editor chapter in volume 2 of the Quartus II Handbook.
Location Assignments

You can create the following types of location assignments for your design and its reserved pins:

- Pin number
- I/O bank
- VREF group
- Edge

I/O bank, VREF group, and Edge location assignments are supported only for Stratix and Cyclone series device families.

You can assign a location to your pins using the Pin Planner or the Assignment Editor. To make a pin location assignment using the Assignment Editor, on the Assignments menu, click Assignment Editor and select the Pin category from the Category list. Type the pin name and select a location from the Location list.

It is common to place a group of pins (or bus) with compatible I/O standards in the same I/O bank or VREF group. For example, two buses with two compatible I/O standards, such as 2.5 V and SSTL-II, can be placed in the same I/O bank.

An easy way to place large buses that exceed the pins available in a particular I/O bank is to use edge location assignments. Edge location assignments improve the circuit board routing ability of large buses, because they are close together near an edge. Figure 5–19 shows Altera device package edges.

![Die View and Package View of the Four Edges on an Altera Device](image)

Suggested and Partial Placement

The Start I/O Assignment Analysis command automatically assigns suggested pin locations to unassigned pins in your design so it can perform pin legality checks. For example, if you assign an edge location to a group of LVDS pins, the I/O Assignment Analysis command assigns pin locations for each LVDS pin in the specified edge location and then performs legality checks.

To accept these suggested pin locations, on the Assignments menu, click Back-Annotate Assignments, select Pin & device assignments, and click OK. Back-annotation saves your pin and device assignments in the .qsf file.
Understanding the I/O Assignment Analysis Report and Messages

The **Start I/O Assignment Analysis** command generates detailed analysis reports and a `.pin` file. The detailed messages in the reports help you quickly understand and resolve pin assignment errors. Each message includes a related node name and a description of the problem.

To view the report file, on the Project menu, click **Compilation Report**. The Fitter section of the Compilation Report contains the following sections:

- Summary
- Settings
- Resource Section
- I/O Rules Section
- Device Options
- Advanced Fitter Data
- Pin-Out File
- Fitter Messages

The Resource Section categorizes the pins as Input Pins, Output Pins, and Bidir Pins. View the utilization of each I/O bank in your device in the I/O Bank Usage section (Figure 5–20).

**Figure 5–20.** I/O Bank Usage Summary in the I/O Assignment Analysis Report

<table>
<thead>
<tr>
<th>I/O Bank</th>
<th>Usage</th>
<th>VCCO Voltage</th>
<th>VREF Voltage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0 / 40 (0 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0 / 44 (0 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3 / 52 (6 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>7 / 42 (17 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1 / 44 (2 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1 / 40 (3 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7 / 42 (17 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>2 / 50 (4 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>1 / 40 (17 %)</td>
<td>3.3V</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1 / 40 (17 %)</td>
<td>3.3V</td>
</tr>
</tbody>
</table>

The I/O Rules Section includes detailed information about the I/O rules tested during I/O Assignment Analysis, in three sub-reports. The I/O Rules Summary report provides a quick summary of the number of I/O rules tested and how many applicable rules passed, how many failed, and how many were unchecked because of other failing rules (Figure 5–21).
Validating Pin Assignments

Figure 5–21. I/O Rules Summary Report

![I/O Rules Summary Report]

The I/O Rules Details report provides detailed information on all I/O rules. Applicable rules indicate whether they passed, failed, or could not be checked (Figure 5–22). All rules are given a level of severity from Low to Critical to indicate their level of importance for an effective analysis.

Figure 5–22. I/O Rules Details Report

![I/O Rules Details Report]

The I/O Rules Matrix shows how each I/O rule was tested on each pin in the design (Figure 5–23). Applicable rules that could be checked either pass or fail for each pin.
To find and make pin assignment adjustments on a pin that fails an I/O rule, right-click the pin name. Point to Locate and select a location where the pin exists, such as the Pin Planner. Make appropriate changes to fix the pin assignments and rerun I/O Assignment Analysis. Check the resulting I/O Rules Matrix to verify that your changes fixed the problem and allowed the failing pin assignment to pass. To rerun I/O rule analysis, on the Processing menu, point to Start and click Start I/O Assignment Analysis.

The Fitter Messages page stores all messages including errors, warnings, and information messages.

You can view the detailed messages in the Fitter Messages page in the Compilation Report and in the Processing tab in the Messages window. To open the Messages window, on the View menu, point to Utility windows and click Messages.

Use the Location box to help resolve error messages. Select from the Location list and click Locate.

Figure 5–24 shows an example of error messages reported by I/O Assignment Analysis.

You can correct the I/O Assignment Analysis failure shown for the pin in Figure 5–24 by setting the proper current drive strength for the I/O standard assigned for that pin. Current drive strength can be set in the Assignment Editor using the “Current Drive Strength” assignment.

For more information about the Assignment Editor, refer to the Assignment Editor chapter in volume 2 of the Quartus II Handbook.
The effectiveness of I/O Assignment Analysis is relative to the completeness of your pin-related assignments and design. To ensure your design functions correctly, include all pin-related assignments and as many design files as possible in your Quartus II project.

Table 5–3 on page 5–37 and Table 5–4 on page 5–38 list a subset of the I/O rule checks performed when you run I/O Assignment Analysis with and without design files.

For more detailed information about each I/O rule, refer to the appropriate device handbook.

### Table 5–3. Examples of I/O Rule Checks *(Note 1)*

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
<th>Device Families</th>
<th>HDL Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O bank capacity</td>
<td>Checks the number of pins assigned to an I/O bank against the number of pins allowed in the I/O bank.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>I/O bank $V_{ccio}$ voltage compatibility</td>
<td>Checks that no more than one $V_{ccio}$ is required for the pins assigned to the I/O bank.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>I/O bank $V_{ref}$ voltage compatibility</td>
<td>Checks that no more than one $V_{ref}$ is required for the pins assigned to the I/O bank.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>I/O standard and location conflicts</td>
<td>Checks whether the pin location supports the assigned I/O standard.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>I/O standard and signal direction conflicts</td>
<td>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.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>Differential I/O standards cannot have open drain turned on</td>
<td>Checks that open drain is turned off for all pins with a differential I/O standard.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>I/O standard and drive strength conflicts</td>
<td>Checks whether the drive strength assignments are within the specifications of the I/O standard.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>Drive strength and location conflicts</td>
<td>Checks whether the pin location supports the assigned drive strength.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>$BUSHOLD$ and location conflicts</td>
<td>Checks whether the pin location supports $BUSHOLD$. For example, dedicated clock pins do not support $BUSHOLD$.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>$WEAK_{PULLUP}$ and location conflicts</td>
<td>Checks whether the pin location supports $WEAK_{PULLUP}$ (for example, dedicated clock pins do not support $WEAK_{PULLUP}$)</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>Electromigration check</td>
<td>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.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>$PCI_{IO}$ clamp diode, location, and I/O standard conflicts</td>
<td>Checks whether the pin location along with the I/O standard assigned supports $PCI_{IO}$ clamp diode.</td>
<td>All</td>
<td>No</td>
</tr>
<tr>
<td>SERDES and I/O pin location compatibility check</td>
<td>Checks that all pins connected to a SERDES in your design are assigned to dedicated SERDES pin locations.</td>
<td>All</td>
<td>Yes</td>
</tr>
<tr>
<td>PLL and I/O pin location compatibility check</td>
<td>Checks whether pins connected to a PLL are assigned to the dedicated PLL pin locations.</td>
<td>All</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Note to Table 5–3:*

(Note 1) The supported device families are: Arria® II GX, Arria GX, Cyclone, Cyclone II, Cyclone III, HardCopy, Stratix, Stratix II, Stratix II GX, Stratix III, Stratix IV, Stratix GX, and MAX II devices.
Using Output Enable Group Logic Option Assignments with I/O Assignment Analysis

Each device has a certain number of VREF pins, and each VREF pin supports a certain number of I/O pins. Check the device pin-outs to locate the VREF pins and their associated I/O pins. A VREF pin and its supported I/O pins are called a VREF bank. The VREF pins are only used for VREF I/O standards; for example, SSTL and HSTL input pins. 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. For the Stratix II flip chip package, only 20 outputs can be present in a VREF bank when a VREF I/O standard input is present in that bank.

For interfaces that use bidirectional VREF I/O pins, the VREF restriction must be met when the pins are driving in either direction. If a set of bidirectional signals are controlled by different output enables, the I/O Assignment Analysis command treats these as independent output enables. Use the output enable group logic option assignment to treat the set of bidirectional signals as a single output enable. This is important in the case of external memory interfaces.

For example, in the case of a DDR2 interface in a Stratix II device, a Stratix II device can have 30 pins in a VREF group. Each byte lane for a ×8 DDR2 interfaces has 1 DQS pin and 8 DQ pins, for a total of 9 pins per byte lane. DDR2 uses SSTL18 as its I/O standard, which is a VREF I/O standard. In typical interfaces, each byte lane has its own output enable. In this example, the DDR2 interface has 4 byte lanes. Using 30 I/O pins in a VREF group, there are 3 byte lanes and an extra byte lane that supports the 3...
remaining pins. If you do not use the output enable group logic option assignment, the I/O Assignment Analysis command analyzes each byte lane as an independent group driven by a unique output enable. With this arrangement, the worst-case scenario is when the 3 pins are inputs, and the other 27 pins are outputs. In this case, the 27 output pins violate the 20-output pin limit.

In a DDR2 interface, all DQS and DQ pins are always driven in the same direction. Therefore, the I/O Assignment Analysis reports an error that is not applicable to your design. Assigning an output enable group logic option assignment to the DQS and DQ pins forces the I/O Assignment Analyzer to check these pins as a group driven by a common output enable. When using the output enable group logic option assignment, the DQS and DQ pins are checked as all input pins or all output pins. This does not violate the rules described in Table 5–3 on page 5–37 and Table 5–4 on page 5–38.

The value for the output enable group logic option assignment should be an integer value. All sets of signals that are driving in the same direction should be given the same integer value. You can also use the output enable group logic option assignment with 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). Therefore, an output enable group logic option assignment should assign to the data mask the same value as to the DQ and DQS signals.

Output enable groups can also be used on VREF input pins. If the VREF input pins are not active during the time the outputs are driving, add the VREF input pins to the output enable group. This procedure removes the VREF input pins from the VREF analysis. For example, the QVLD signal for RLDRAM II is only active during a read. During a write, the QVLD pin is not active and so it does not count as an active VREF input pin within the VREF group. The QVLD pins can be placed in the same output enable group as the RLDRAM II data pins.

Validating Pin Assignments after Full Compilation

If you used the Live I/O check feature during pin placements, many of the I/O assignments have been verified immediately as you made the assignment. There are some placement rules that are checked only during I/O assignment analysis and full compilation of your design. The Quartus II software validates I/O assignments at three levels. The first level checking is done with the Live I/O check feature ON. A more comprehensive level of checking is performed with I/O Assignments Analysis. The final I/O timing check is done when you fully compile your design. (To better understand I/O timing analysis, refer to “I/O Timing Analysis” on page 5–40.)

To avoid costly board re-spins, you must perform full validation with full compilation with complete design files and constraints. With timing information, the Quartus II Fitter makes intelligent placements and routing to achieve optimal timing performance in your design. Use the TimeQuest SDC editor to create timing constraints for inputs, outputs, and bidirectional pins. If you are using the Quartus II Classic Timing Analyzer, specify timing constraints on the Classic Timing Analyzer Settings page of the Settings dialog box.
For more information about the Quartus II TimeQuest Timing Analyzer and the Classic Timing Analyzer, refer to the Timing Analysis section of volume 3 of the Quartus II Handbook.

I/O Timing Analysis

Timing analysis is usually run during a full compilation of your design or with early timing estimate runs. You can also run timing analysis independently after full compilation from the Processing menu. For example, if you change the slew rates or current strengths of some I/O pins as ECOs, you do not have to recompile the entire design, but only run timing analysis to verify the timing of your design.

As part of I/O planning, especially with high-speed designs, you should take board-level signal integrity and timing into account. When adding an FPGA device with high-speed interfaces to a board design, the quality of the signal at the far end of the board route, as well as the propagation delay in getting there, is vital for proper system operation.

As part of I/O planning in your FPGA design, you must understand I/O timing results in the Quartus II software that are reported after performing timing analysis on your design. If you have all your design files complete and have completed full compilation, all the timing checks related to I/O timing are covered during timing analysis of your design. Static timing analysis is performed when you compile your design in the Quartus II software. You must understand I/O timing and what factors affect I/O timing paths in your design. One important factor that counts greatly in I/O timing results is how accurately you specify the output loads at the output and bidirectional pins in your FPGA design. Incomplete I/O constraints can affect your I/O timing results.

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

- I/O timing using a default or user-specified capacitive load with no signal integrity analysis (default)

  The Quartus II TimeQuest Timing Analyzer and the Quartus II Classic Timing Analyzer create timing reports that measure $t_{CO}$ to an I/O pin using a default or user-specified value for a capacitive load.

- The Quartus II software Enable Advanced I/O Timing option utilizing a user-defined board trace model to produce enhanced timing reports from accurate, “board-aware” simulation models

  The Quartus II software Enable Advanced I/O Timing option enables you to configure a complete board trace model for each I/O standard or pin used in your design. With Enable Advanced I/O Timing turned on, the Quartus II TimeQuest Timing Analyzer uses the results of simulations of the I/O buffer, package, and board trace model to generate more accurate I/O delays and extra reports to give insight into signal behavior at the system level. You can use these advanced timing reports as a guide to make changes to your I/O assignments and board design to improve timing and signal integrity.
Full board routing simulation in third-party tools using Altera-provided or Quartus II software generated IBIS or HSPICE I/O models

The creation of simulation model files for use by third-party board simulation tools is achieved with the IBIS and HSPICE Writers. The IBIS and HSPICE Writers in the Quartus II software can export accurate simulation models for use in applications such as Mentor Graphics HyperLynx and Synopsys HSPICE.

This section describes the first and second methods.

I/O timing using a default or user-specified capacitive load is not supported for Cyclone III, Stratix III, and Stratix IV devices. Use the Enable Advanced I/O Timing option for Cyclone II, Stratix III, and Stratix IV devices.

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.

I/O Timing and Power with Capacitive Loading

When calculating \( t_{\text{CO}} \) and power for output and bidirectional pins, the Quartus II TimeQuest Timing Analyzer and the PowerPlay Power Analyzer use a bulk capacitive load. This is the default method for these pins. You can adjust the value of the capacitive load per I/O standard to get \( t_{\text{CO}} \) and power measurements that more accurately reflect the behavior of the output or bidirectional net on your PCB. Input pins ignore this setting. To adjust the value of the capacitive load, on the Assignments menu, click Device. Click Device & Pin Options and click the Capacitive Loading tab (Figure 5–25).

Figure 5–25. Capacitive Tab of the Device and Pin Options Dialog Box
All of the available I/O standards for your selected device are listed with their default loading values in picofarads (pF). Adjust the loading values as desired for the I/O standards used in your design. Power and t\text{co} measurements in the Compilation Report are adjusted based on the settings.

You can also adjust the load on any individual pin in the Groups list or All Pins list in the Pin Planner by adding the Output Pin Load column. Right-click anywhere in either list and select Customize Columns. Select Output Pin Load from the list of available custom columns and add it to the list of visible columns. You can customize the load for individual pins or multiple pins with different I/O standards.

For more information about capacitive loading, the devices that support it, and how t\text{co} and power are adjusted based on the setting, refer to the Quartus II Help.

**Enabling and Configuring Advanced I/O Timing**

With the Quartus II software Enable Advanced I/O Timing option turned on, you can expand upon the basic timing and power measurements made with the Capacitive Loading settings. The Enable Advanced I/O Timing option gives you the ability to fully define not only the capacitive load, but also 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 as well as customize the model for specific pins using a graphical interface.

When the Enable Advanced I/O Timing option is turned on, the board trace model replaces the Capacitive Loading tab settings because the load is included in the model. For timing measurements, the entire board trace model is taken into account when calculating I/O delays. For power measurements, an effective capacitive load is used based on the sum of the capacitive elements in the model. This includes the Near capacitance, Far capacitance, and Transmission line distributed capacitance elements of the model.

For Cyclone III and Stratix IV devices, advanced I/O timing is the only way to measure I/O timing. Advanced I/O timing is supported for Stratix II devices also. All other devices use capacitive loading for I/O t\text{co} and power measurements. Check the Altera website at www.altera.com to determine which devices are supported in newer versions of the Quartus II software. For Cyclone III and Stratix III devices, the Enable Advanced I/O Timing option is turned on by default and is always performed when you run the Quartus II TimeQuest timing analyzer.

Before you configure a board trace model for advanced I/O timing, you must turn on Enable Advanced I/O Timing if your selected device supports it. All devices in each supported family work with advanced I/O timing. If the Settings dialog box is not currently open, on the Assignments menu, click Settings. In the Category list, click the “+” icon to expand Timing Analysis Settings. Select TimeQuest Timing Analyzer. The TimeQuest Timing Analysis page appears. Turn on Enable Advanced I/O Timing.
**Define Overall Board Trace Models**

You can now define an overall board trace model for each I/O standard in your design. This is the default model for all pins that use a particular I/O standard. After configuring the overall board trace model, customize the model for specific pins using the Board Trace Model view in the Pin Planner.

With the Settings dialog box open, in the Category list, click Device. Click Device & Pin Options and click the Board Trace Model tab (Figure 5–26).

**Figure 5–26. Board Trace Model Tab of the Device and Pin Options Dialog Box**

![Board Trace Model Tab of the Device and Pin Options Dialog Box](image)

You can still click the Capacitive Loading tab. However, because you can configure all capacitive loading settings as part of the board trace model, the tab indicates that you must use the settings in the Board Trace Model tab.

All of the I/O standards available to the device are listed. Select any I/O standard from the list. The Board trace model list displays the names and values of all configurable components of the board trace for the selected I/O standard. Components of the model are initially set to short, open, or a numeric value depending on the component. The default settings for components in the model for each I/O standard are device-specific and match the default test model used for calculating delay when the Enable Advanced I/O Timing option is turned off. In this way, default delay measurements are the same whether or not the Enable Advanced I/O Timing option is used.

For information about the default models used for measuring I/O delay, refer to the DC & Switching Characteristics chapter in the relevant device handbook.
All of the component values listed in Figure 5–26 are adjustable. For differential I/O standards, the component values you set are used for both the positive and negative signals of a differential pair. An additional component, Far differential resistance, is also included. To reset individual settings to their defaults, leave the setting blank. If you want all the settings for an I/O standard to revert to their original settings, click Reset. Click OK to close the Device & Pin Options dialog box. Click OK again to close the Settings dialog box.

Any component value changes made in the Board Trace Model tab for a particular I/O standard are reflected in the Board Trace Model view in the Pin Planner of all pins assigned with the same I/O standard (described in “Customize the Board Trace Model in the Pin Planner”). However, custom component value changes made to selected pins in the Board Trace Model view in the Pin Planner take priority and are not affected by changes made to an I/O standard in the Board Trace Model tab.

**Customize the Board Trace Model in the Pin Planner**

In addition to the views available in the Package View in the Pin Planner, you can also view a graphical representation of the board trace model you have configured using the Board Trace Model view. To open the Board Trace Model view, right-click on an output or bidirectional pin in the Groups list, All Pins list, or Package View and click **Board Trace Model**. The Board Trace Model view opens in a floating window (Figure 5–27).

**Figure 5–27. Board Trace Model View**

For differential signals, the Board Trace Model view displays the routing and components for both the positive and negative signals of the differential pair (Figure 5–28).
Any changes made to the Board Trace Model view for a differential signal pair must be performed on the positive signal of the pair. The settings must match between the positive and negative signals of a differential pair, so the changes are automatically reflected in the settings for the negative signal.

Double-click a component value to edit it. For numerical values, use standard unit prefixes such as \( p \), \( n \), and \( k \) to represent pico, nano, and kilo, respectively. To short a series component or have an open circuit for a parallel component, double-click the component value and select **short** or **open** from the list.

All the assignments for board trace models you specify in the schematic are saved to the Quartus II Settings File (.qsf). You can also enter these Tcl assignment commands in the .qsf to specify the board trace parameters for an output or bidirectional pin. The examples in Example 5–8 use Tcl assignments to specify board trace models.

**Example 5-8. Specifying Board Trace Models**

```tcl
## 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 sel_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 sel_p
```

For more details about configuring component values for a board trace model, including a complete list of the supported unit prefixes and setting the values using Tcl scripts, refer to the Quartus II Help.
To view a display of the model for a particular pin, in the Package View, Groups list, or All Pins list, click on the pin. This changes the Board Trace Model view to display the model of the pin. To select multiple pins that share the same I/O standard, open the Board Trace Model view and edit the model for all of the selected pins. If an input pin or multiple pins with different I/O standards are selected, the Board Trace Model view window indicates that it cannot display the model for the selected pin or pins.

The components in the Board Trace Model view correspond to the components listed in the Board Trace Model tab directly and the settings match initially. You can click and edit any value in the Board Trace Model view to customize the model for the selected pin or pins. Changes made in the Board Trace Model view do not affect the settings in the Board Trace Model tab.

Configuring Board Trace Models

The Quartus II software provides board trace model templates for various I/O standards where you can fill in various parameters. Figure 5–27 shows the template for a 2.5-V I/O standard. This model consists of near-end and far-end board component parameters. Each parameter is configurable in the Board Trace Model tab of the Device & Pin Options window.

To configure board trace models for the pins in your design, define the model for each I/O standard in the Board Trace Model tab. With the overall model defined, use the Board Trace Model view in the Pin Planner to customize individual pins as required. These customizations take priority over the global settings in the Board Trace Model tab on a per-pin and per-model component basis.

Modeling of the near-end of the board trace includes the elements which are close to the FPGA and modeling of the far-end includes the elements which are at the receiver end of the link, closer to the receiving device. The topology represented in the Quartus II board trace model is conceptual and does not necessarily match the board trace component for component. For example near-end model parameters can represent FPGA-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. The same circuit can be analyzed with near-end modeling of the entire board, including memory component termination, and far-end modeling of the actual memory component.

The far-end capacitance (Cf) shown in Figure 5–28 represents the external-device or multiple-device capacitive load. If you have multiple devices on the far-end, you need to find the equivalent capacitance at the far-end, taking into account all receiver capacitances. The far end capacitance Cf is can be the sum of all the receiver capacitances. Specifications for external device capacitance values can be found in the datasheet for the receiving device or devices.

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 like Mentor Graphics Hyperlynx.
Near-End vs Far-End Timing Analysis

With advanced I/O timing analysis, you have the option of selecting a near-end or far-end point for your I/O timing. This option can be selected on the I/O Timing tab as shown in Figure 5–26. With near-end timing, the timing is analyzed to the FPGA pin, ending at the vertical dashed line shown in Figure 5–27 separating the FPGA I/O pin and off-chip components.

By default, advanced I/O timing analysis analyzes output I/O timing to the FPGA pin. When you use the near-end option, you can use a set_output_delay SDC timing constraint to account for the delay across the board. However, when a far-end I/O timing endpoint is chosen, then advanced I/O timing analysis analyzes timing 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 while performing timing analysis.

Create Signal Integrity Result Reports

After you have turned on Enable Advanced I/O Timing and configured board trace models for the pins you want to analyze, compile your project or run the Quartus II TimeQuest Timing Analyzer after a full compilation. The Enable Advanced I/O Timing option creates signal integrity subreports under TimeQuest Timing Analyzer in the Compilation Report window.

The Board Trace Model Assignments report (Figure 5–29) summarizes the board trace model component settings for each output and bidirectional signal.

The Signal Integrity Metrics subfolder contains detailed reports listing all of the metrics calculated by the Enable Advanced I/O Timing option (Figure 5–30).
The Slow- and Fast-Corner Signal Integrity Metrics reports are generated by the **Enable Advanced I/O Timing** option. They list, in tabular format, all of the signal integrity metrics calculated by the **Enable Advanced I/O Timing** option, based on the board trace model settings for each output or bidirectional pin. The reports contain many metrics, including measurements at both the FPGA and at the far-end load of board delay, steady state voltages, and rise and fall times.

The Slow- or Fast-Corner Signal Integrity Metrics reports are generated depending on the **Timing Netlist** option in the Quartus II TimeQuest Timing Analyzer. To select whether to create a slow- or a fast-corner report, in the TimeQuest Timing Analyzer on the Netlist menu, click **Create Timing Netlist**. Under **Delay model**, select **Slow corner** or **Fast corner** to create reports of that type.

For complete descriptions of all of the metrics calculated when the **Enable Advanced I/O Timing** option is turned on and diagrams illustrating the metrics on output waveforms, refer to **Signal Integrity Metrics** in Quartus II Help. 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.

---

**Incorporating PCB Design Tools**

Signal and pin assignments are initially made by the FPGA or ASIC designer and it is up to the board designer to transfer these assignments to the symbols used in their system circuit schematics and board layout correctly. As the board design progresses, pin reassignments might be requested or required to optimize the layout. These reassignments must in turn be relayed to the FPGA designer, so that the new assignments can be validated with the I/O Assignment Analyzer and processed through an updated place-and-route of the FPGA.

The Quartus II software interacts with board layout tools by importing and exporting pin information files, including the `.qsf`, `.pin`, and `.fx` files.

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**.

---

**Scripting Support**

A Tcl script allows you to run procedures and determine settings described in this chapter. You can also run some of these procedures at a command prompt.

For detailed information about specific scripting command options and Tcl API packages, type the following command at a system command prompt to run the Quartus II Command-Line and Tcl API Help browser:

```
quartus_sh --qhelp
```

For more information about Quartus II scripting support, including examples, refer to the **Tcl Scripting** and **Command-Line Scripting** chapters in volume 2 of the **Quartus II Handbook**.
Running the I/O Assignment Analysis

You can run I/O Assignment Analysis with a Tcl command or with a command run at a command prompt. For more information about running the I/O Assignment Analysis, refer to “Understanding the I/O Assignment Analysis Report and Messages” on page 5–34.

Enter the following in a Tcl console or script:

```
execute_flow -check_ios
```

Type the following at a (non-Tcl) system command prompt:

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

Generating a Mapped Netlist

You can generate a mapped netlist with a Tcl command or with a command-line command. For more information about generating a mapped netlist, refer to “Generating a Mapped Netlist” on page 5–32.

Enter the following in the Tcl console or in a 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>
```

Reserving Pins

Use the following Tcl command to reserve a pin:

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

For more information about reserving pins, refer to “Reserving Pins” on page 5–32.

Valid values are:

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

Include the quotes when specifying the value.

Location Assignments

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

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

For more information about location assignments, refer to “Location Assignments” on page 5–33.
Valid locations are pin location names, such as `PIN_A3`. The Stratix and Cyclone series of devices 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` up to `IOBANK_n`, in which `n` is the number of I/O banks in a particular device.

For more information on I/O banks in your device, refer to the appropriate device handbook.

**Conclusion**

The Quartus II software provides many tools and features to help you with the I/O planning process. The I/O assignment analysis process offers the ability to validate pin assignments in all design stages, even before the development of the design. The ability to import and export assignments between the Quartus II software and other PCB tools also enables you to make iterative changes efficiently. Finally, the ability to enter a board trace model and create advanced timing reports based on how I/O signals are routed on a board truly makes the Quartus II software “board-aware.”

**Referenced Documents**

The following documents were referenced in this chapter:

- **AN 90: SameFrame Pin-Out Design for FineLine BGA Packages**
- **AN 315: Guidelines for Designing High-Speed FPGA PCBs**
- **AN 366: Understanding I/O Output Timing in Altera Devices**
- **AN 476: Impact of I/O Settings on Signal Integrity in Stratix III Devices**
- **Altera Device Package Information Datasheet**
- **Analyzing and Optimizing the Design Floorplan** chapter in volume 2 of the Quartus II Handbook
- **Assignment Editor** chapter in volume 2 of the Quartus II Handbook
- **Cadence PCB Design Tools Support** chapter in volume 2 of the Quartus II Handbook
- **Command-Line Scripting** chapter in volume 2 of the Quartus II Handbook
- **Data (DQ) and Data Strobe (DQS) Megafunction User Guide (ALTDQ and ALTDQS)**
- **DC & Switching Characteristics** chapter in volume 1 of the Stratix II Device Handbook
- **Engineering Change Management with the Chip Planner** chapter in volume 2 of the Quartus II Handbook
- **Managing Quartus II Projects** chapter in volume 2 of the Quartus II Handbook
- **Mentor Graphics PCB Design Tools Support** chapter in volume 2 of the Quartus II Handbook
- **Quartus II Integrated Synthesis** chapter in volume 1 of the Quartus II Handbook
- **Quartus II Scripting Reference Manual**
- **Quartus II Support for HardCopy Series Devices** chapter in volume 1 of the Quartus II Handbook
Document Revision History

Table 5–5 shows the revision history for this chapter.

<table>
<thead>
<tr>
<th>Date and Document Version</th>
<th>Changes Made</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2009 v9.1.0</td>
<td>Reorganized entire chapter to include links to Quartus II help for</td>
<td>Updated for the Quartus II software version 9.1 release.</td>
</tr>
<tr>
<td></td>
<td>procedural information previously included in the chapter.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added documentation on near-end and far-end advanced I/O timing</td>
<td></td>
</tr>
<tr>
<td>March 2009 v9.0.0</td>
<td>Updated “Pad View Window” on page 5–20.</td>
<td>Updated for the Quartus II software version 9.0 release.</td>
</tr>
<tr>
<td></td>
<td>Added new figures:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Figure 5–15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Figure 5–16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added new section “SSN Visualization View” on page 5–17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added new section “Creating Exclusive Group Assignments” on page 5–17</td>
<td></td>
</tr>
<tr>
<td>November 2008 v8.1.0</td>
<td>Changed to 8½” x 11” page size.</td>
<td>Updated for the Quartus II software version 8.1 release.</td>
</tr>
<tr>
<td></td>
<td>Reorganized chapter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Updated the following sections:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“I/O Planning Overview” on page 5–5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Early I/O Planning Using the Pin Planner” on page 5–8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Create or Import a Megafunction or IP MegaCore Variation from the Pin Planner” on page 5–9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Configure Nodes” on page 5–10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“I/O Analysis for Designs with Pins Only” on page 5–14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Using the Pin Planner” on page 5–20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Validating Pin Assignments after Full Compilation” on page 5–76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“Accepting Fitter Placements—Back-Annotation Assignments” on page 5–76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“I/O Timing Analysis” on page 5–77</td>
<td></td>
</tr>
</tbody>
</table>
Table 5–5. Document Revision History (Part 2 of 2)

<table>
<thead>
<tr>
<th>Date and Document Version</th>
<th>Changes Made</th>
<th>Summary of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>May 2008 v8.0.0</td>
<td>■ Reorganized chapter</td>
<td>Updated for the Quartus II software version 8.0 release.</td>
</tr>
<tr>
<td></td>
<td>■ Updated links</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Updated the following sections:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Importing and Exporting Pin Assignments” on page 5–16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “I/O Planning Overview” on page 5–5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Creating Pin-Related Assignments” on page 5–19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Using Hardware Description Language (HDL)” on page 5–54</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Validating Pin Assignments” on page 5–56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “I/O Timing Analysis” on page 5–76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “.pin File” on page 5–18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Import a Megafuction or IP MegaCore Variation from the Pin Planner” on page 5–13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Using the Live I/O Check Feature to Validate Pin Assignments” on page 5–57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Pin Migration View” on page 5–30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “altera_attribute” on page 5–55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Assigning a Location for Differential Pins” on page 5–37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Synthesis Attributes” on page 5–55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Added the following sections:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Swapping Pin Locations” on page 5–42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Using Low-Level I/O Primitives” on page 5–56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Advanced I/O Timing in the Quartus II Software” on page 5–78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>→ “Enabling and Configuring Advanced I/O Timing” on page 5–78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ Updated figures to reflect updates to the Quartus II software</td>
<td></td>
</tr>
</tbody>
</table>

For previous versions of the Quartus II Handbook, refer to the Quartus II Handbook Archive.