Intel® Quartus® Prime Pro Edition User Guide

PCB Design Tools

Updated for Intel® Quartus® Prime Design Suite: 18.1
1. Signal Integrity Analysis with Third-Party Tools

1.1. Signal Integrity Analysis with Third-Party Tools
1.1.1. Signal Integrity Simulations with HSPICE and IBIS Models
1.2. I/O Model Selection: IBIS or HSPICE
1.3. FPGA to Board Signal Integrity Analysis Flow
1.3.1. Create I/O and Board Trace Model Assignments
1.3.2. Output File Generation
1.3.3. Customize the Output Files
1.3.4. Set Up and Run Simulations in Third-Party Tools
1.3.5. Interpret Simulation Results
1.4. Simulation with IBIS Models
1.4.1. Elements of an IBIS Model
1.4.2. Creating Accurate IBIS Models
1.4.3. Design Simulation Using the Mentor Graphics HyperLynx Software
1.4.4. Configuring LineSim to Use Intel IBIS Models
1.4.5. Integrating Intel IBIS Models into LineSim Simulations
1.4.6. Running and Interpreting LineSim Simulations
1.5. Simulation with HSPICE Models
1.5.1. Supported Devices and Signaling
1.5.2. Accessing HSPICE Simulation Kits
1.5.3. The Double Counting Problem in HSPICE Simulations
1.5.4. HSPICE Writer Tool Flow
1.5.5. Running an HSPICE Simulation
1.5.6. Interpreting the Results of an Output Simulation
1.5.7. Interpreting the Results of an Input Simulation
1.5.8. Viewing and Interpreting Tabular Simulation Results
1.5.9. Viewing Graphical Simulation Results
1.5.10. Making Design Adjustments Based on HSPICE Simulations
1.5.11. Sample Input for I/O HSPICE Simulation Deck
1.5.12. Sample Output for I/O HSPICE Simulation Deck
1.5.13. Advanced Topics
1.6. Signal Integrity Analysis with Third-Party Tools Document Revision History

2. Reviewing Printed Circuit Board Schematics with the Intel Quartus Prime Software

2.1. Reviewing Intel Quartus Prime Software Settings
2.1.1. Device and Pins Options Dialog Box Settings
2.2. Reviewing Device Pin-Out Information in the Fitter Report
2.3. Reviewing Compilation Error and Warning Messages
2.4. Using Additional Intel Quartus Prime Software Features
2.5. Using Additional Intel Quartus Prime Software Tools
2.5.1. Pin Planner
2.6. Reviewing Printed Circuit Board Schematics with the Intel Quartus Prime Software Revision History


3.1. Integrating with DxDesigner
3.1.1. DxDesigner Project Settings
3.1.2. Creating Schematic Symbols in DxDesigner
3.2. Mentor Graphics PCB Design Tools Support Revision History .................................. 51

4. Cadence Board Design Tools Support ......................................................................... 52
   4.1. Cadence PCB Design Tools Support ................................................................... 52
   4.2. Product Comparison ......................................................................................... 53
   4.3. FPGA-to-PCB Design Flow ............................................................................. 53
      4.3.1. Integrating Intel FPGA Designs ................................................................. 55
   4.4. Setting Up the Intel Quartus Prime Software ...................................................... 56
      4.4.1. Generating a .pin File ............................................................................. 56
   4.5. FPGA-to-Board Integration with the Cadence Allegro Design Entry HDL Software ................................................................................................................. 56
      4.5.1. Creating Symbols ..................................................................................... 57
      4.5.2. Instantiating the Symbol in the Cadence Allegro Design Entry HDL Software ................................................................................................................. 62
   4.6. FPGA-to-Board Integration with Cadence Allegro Design Entry CIS Software ................................................................................................................. 63
      4.6.1. Creating a Cadence Allegro Design Entry CIS Project ................................. 64
      4.6.2. Generating a Part ..................................................................................... 64
      4.6.3. Generating Schematic Symbol ................................................................. 65
      4.6.4. Splitting a Part ......................................................................................... 65
      4.6.5. Instantiating a Symbol in a Design Entry CIS Schematic ............................. 67
      4.6.6. Intel Libraries for the Cadence Allegro Design Entry CIS Software ............. 67
   4.7. Cadence Board Design Tools Support Revision History ..................................... 69

   A. Intel Quartus Prime Pro Edition User Guides ...................................................... 71
1. Signal Integrity Analysis with Third-Party Tools

1.1. Signal Integrity Analysis with Third-Party Tools

With the ever-increasing operating speed of interfaces in traditional FPGA design, the timing and signal integrity margins between the FPGA and other devices on the board must be within specification and tolerance before a single PCB is built.

If the board trace is designed poorly or the route is too heavily loaded, noise in the signal can cause data corruption, while overshoot and undershoot can potentially damage input buffers over time.

As FPGA devices are used in high-speed applications, signal integrity and timing margin between the FPGA and other devices on the printed circuit board (PCB) are important aspects to consider to ensure proper system operation. To avoid time-consuming redesigns and expensive board respins, the topology and routing of critical signals must be simulated. The high-speed interfaces available on current FPGA devices must be modeled accurately and integrated into timing models and board-level signal integrity simulations. The tools used in the design of an FPGA and its integration into a PCB must be “board-aware”—able to take into account properties of the board routing and the connected devices on the board.

The Intel® Quartus® Prime software provides methodologies, resources, and tools to ensure good signal integrity and timing margin between Intel FPGA devices and other components on the board. Three types of analysis are possible with the Intel Quartus Prime software:

- I/O timing with a default or user-specified capacitive load and no signal integrity analysis (default)
- The Intel Quartus Prime Enable Advanced I/O Timing option utilizing a user-defined board trace model to produce enhanced timing reports from accurate “board-aware” simulation models
- Full board routing simulation in third-party tools using Intel-provided or generated Input/Output Buffer Information Specification (IBIS) or HSPICE I/O models

I/O timing using a specified capacitive test load requires no special configuration other than setting the size of the load. I/O timing reports from the Intel Quartus Prime Timing Analyzer or the Intel Quartus Prime Classic Timing Analyzer are generated based only on point-to-point delays within the I/O buffer and assume the presence of the capacitive test load with no other details about the board specified. The default size of the load is based on the I/O standard selected for the pin. Timing is measured to the FPGA pin with no signal integrity analysis details.

The Enable Advanced I/O Timing option expands the details in I/O timing reports by taking board topology and termination components into account. A complete point-to-point board trace model is defined and accounted for in the timing analysis. This ability to define a board trace model is an example of how the Intel Quartus Prime software is “board-aware.”
In this case, timing and signal integrity metrics between the I/O buffer and the defined far end load are analyzed and reported in enhanced reports generated by the Intel Quartus Prime Timing Analyzer.

The information about signal integrity in this chapter refers to board-level signal integrity based on I/O buffer configuration and board parameters, not simultaneous switching noise (SSN), also known as ground bounce or $V_{CC}$ sag. SSN is a product of multiple output drivers switching at the same time, causing an overall drop in the voltage of the chip’s power supply. This can cause temporary glitches in the specified level of ground or $V_{CC}$ for the device.

This chapter is intended for FPGA and board designers and includes details about the concepts and steps involved in getting designs simulated and how to adjust designs to improve board-level timing and signal integrity. Also included is information about how to create accurate models from the Intel Quartus Prime software and how to use those models in simulation software.

The information in this chapter is meant for those who are familiar with the Intel Quartus Prime software and basic concepts of signal integrity and the design techniques and components in good PCB design. Finally, you should know how to set up simulations and use your selected third-party simulation tool.

### 1.1.1. Signal Integrity Simulations with HSPICE and IBIS Models

The Intel Quartus Prime software can export accurate HSPICE models with the built-in HSPICE Writer. You can run signal integrity simulations with these complete HSPICE models in Synopsys* HSPICE. IBIS models of the FPGA I/O buffers are also created easily with the Intel Quartus Prime IBIS Writer.

You can run signal integrity simulations with these complete HSPICE models in Synopsys HSPICE.

You can integrate IBIS models into any third-party simulation tool that supports them, such as the Mentor Graphics* HyperLynx* software. With the ability to create industry-standard model definition files quickly, you can build accurate simulations that can provide data to help improve board-level signal integrity.

The I/O’s IBIS and HSPICE model creation available in the Intel Quartus Prime software can help prevent problems before a costly board respin is required. In general, creating and running accurate simulations is difficult and time consuming. The tools in the Intel Quartus Prime software automate the I/O model setup and creation process by configuring the models specifically for your design. With these tools, you can set up and run accurate simulations quickly and acquire data that helps guide your FPGA and board design.

For a more information about SSN and ways to prevent it, refer to AN 315: Guidelines for Designing High-Speed FPGA PCBs.

For information about basic signal integrity concepts and signal integrity details pertaining to Intel FPGA devices, visit the Intel Signal & Power Integrity Center.

**Related Information**

1.2. I/O Model Selection: IBIS or HSPICE

The Intel Quartus Prime software can export two different types of I/O models that are useful for different simulation situations, IBIS models and HSPICE models.

IBIS models define the behavior of input or output buffers through voltage-current (V-I) and voltage-time (V-t) data tables. HSPICE models, or decks, include complete physical descriptions of the transistors and parasitic capacitances that make up an I/O buffer along with all the parameter settings that you require to run a simulation.

The Intel Quartus Prime software generates HSPICE decks, and adds preconfigured I/O standard, voltage, and pin loading settings for each pin in your design.

The choice of I/O model type is based on many factors.

Table 1. IBIS and HSPICE Model Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>IBIS Model</th>
<th>HSPICE Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Buffer Description</td>
<td>Behavioral— I/O buffers are described by voltage-current and voltage-time tables in typical, minimum, and maximum supply voltage cases.</td>
<td>Physical— I/O buffers and all components in a circuit are described by their physical properties, such as transistor characteristics and parasitic capacitances, as well as their connections to one another.</td>
</tr>
<tr>
<td>Model Customization</td>
<td>Simple and limited—The model completely describes the I/O buffer and does not usually have to be customized.</td>
<td>Fully customizable—Unless connected to an arbitrary board description, the description of the board trace model must be customized in the model file. All parameters of the simulation are also adjustable.</td>
</tr>
<tr>
<td>Simulation Set Up and Run Time</td>
<td>Fast—Simulations run quickly after set up correctly.</td>
<td>Slow—Simulations take time to set up and take longer to run and complete.</td>
</tr>
<tr>
<td>Simulation Accuracy</td>
<td>Good—For most simulations, accuracy is sufficient to make useful adjustments to the FPGA or board design to improve signal integrity.</td>
<td>Excellent—Simulations are highly accurate, making HSPICE simulation almost a requirement for any high-speed design where signal integrity and timing margins are tight.</td>
</tr>
<tr>
<td>Third-Party Tool Support</td>
<td>Excellent—Almost all third-party board simulation tools support IBIS.</td>
<td>Good—Most third-party tools that support SPICE support HSPICE. However, Synopsys HSPICE is required for simulations of Intel's encrypted HSPICE models.</td>
</tr>
</tbody>
</table>

For more information about IBIS files created by the Intel Quartus Prime IBIS Writer and IBIS files in general, as well as links to websites with detailed information, refer to AN 283: Simulating Intel Devices with IBIS Models.

Related Information
AN 283: Simulating Intel Devices with IBIS Models

1.3. FPGA to Board Signal Integrity Analysis Flow

Board signal integrity analysis can take place at any point in the FPGA design process and is often performed before and after board layout. If it is performed early in the process as part of a pre-PCB layout analysis, the models used for simulations can be more generic.

These models can be changed as much as required to see how adjustments improve timing or signal integrity and help with the design and routing of the PCB. Simulations and the resulting changes made at this stage allow you to analyze “what if” scenarios to plan and implement your design better. To assist with early board signal integrity
analysis, you can download generic IBIS model files for each device family and obtain HSPICE buffer simulation kits from the “Board Level Tools” section of the EDA Tool Support Resource Center.

Typically, if board signal integrity analysis is performed late in the design, it is used for a post-layout verification. The inputs and outputs of the FPGA are defined, and required board routing topologies and constraints are known. Simulations can help you find problems that might still exist in the FPGA or board design before fabrication and assembly. In either case, a simple process flow illustrates how to create accurate IBIS and HSPICE models from a design in the Intel Quartus Prime software and transfer them to third-party simulation tools.

Your design depends on the type of model, IBIS or HSPICE, that you use for your simulations. When you understand the steps in the analysis flow, refer to the section of this chapter that corresponds to the model type you are using.
Related Information

EDA Tool Support Resource Center
For more information, generic IBIS model files for each device family, and to obtain HSPICE buffer simulation kits.
1.3.1. Create I/O and Board Trace Model Assignments

You can configure a board trace model for output signals or for bidirectional signals in output mode. You can then automatically transfer its description to HSPICE decks generated by the HSPICE Writer. This helps improve simulation accuracy.

To configure a board trace model, in the Settings dialog box, in the Timing Analyzer page, turn on the Enable Advanced I/O Timing option and configure the board trace model assignment settings for each I/O standard used in your design. You can add series or parallel termination, specify the transmission line length, and set the value of the far-end capacitive load. You can configure these parameters either in the Board Trace Model view of the Pin Planner, or click Assignments ➤ Device ➤ Device and Pin Options.

The Intel Quartus Prime software can generate IBIS models and HSPICE decks without having to configure a board trace model with the Enable Advanced I/O Timing option. In fact, IBIS models ignore any board trace model settings other than the far-end capacitive load. If any load value is set other than the default, the delay given by IBIS models generated by the IBIS Writer cannot be used to account correctly for the double counting problem. The load value mismatch between the IBIS delay and the $t_{CO}$ measurement of the Intel Quartus Prime software prevents the delays from being safely added together. Warning messages displayed when the EDA Netlist Writer runs indicate when this mismatch occurs.

1.3.2. Output File Generation

IBIS and HSPICE model files are not generated by the Intel Quartus Prime software by default. To generate or update the files automatically during each project compilation, select the type of file to generate and a location where to save the file in the project settings.

The IBIS and HSPICE Writers in the Intel Quartus Prime software are run as part of the EDA Netlist Writer during normal project compilation. If either writer is turned on in the project settings, IBIS or HSPICE files are created and stored in the specified location. For IBIS, a single file is generated containing information about all assigned pins. HSPICE file generation creates separate files for each assigned pin. You can run the EDA Netlist Writer separately from a full compilation in the Intel Quartus Prime software or at the command line.

1.3.3. Customize the Output Files

The files generated by either the IBIS or HSPICE Writer are text files that you can edit and customize easily for design or experimentation purposes.

IBIS files downloaded from the Intel website must be customized with the correct RLC values for the specific device package you have selected for your design. IBIS files generated by the IBIS Writer do not require this customization because they are configured automatically with the RLC values for your selected device. HSPICE decks require modification to include a detailed description of your board. With Enable Advanced I/O Timing turned on and a board trace model defined in the Intel Quartus Prime software, generated HSPICE decks automatically include that model's parameters. However, Intel recommends that you replace that model with a more detailed model that describes your board design more accurately. A default simulation included in the generated HSPICE decks measures delay between the FPGA and the...
far-end device. You can make additions or adjustments to the default simulation in the generated files to change the parameters of the default simulation or to perform additional measurements.

1.3.4. Set Up and Run Simulations in Third-Party Tools

When you have generated the files, you can use them to perform simulations in your selected simulation tool.

With IBIS models, you can apply them to input, output, or bidirectional buffer entities and quickly set up and run simulations. For HSPICE decks, the simulation parameters are included in the files. Open the files in Synopsys HSPICE and run simulations for each pin as required.

With HSPICE decks generated from the HSPICE Writer, the double counting problem is accounted for, which ensures that your simulations are accurate. Simulations that involve IBIS models created with anything other than the default loading settings in the Intel Quartus Prime software must take the change in the size of the load between the IBIS delay and the Intel Quartus Prime tCO measurement into account. Warning messages during compilation alert you to this change.

1.3.5. Interpret Simulation Results

If you encounter timing or signal integrity issues with your high-speed signals after running simulations, you can make adjustments to I/O assignment settings in the Intel Quartus Prime software.

You can adjust drive strength or I/O standard, or make changes to the board routing or topology. After regenerating models in the Intel Quartus Prime software based on the changes you have made, rerun the simulations to check whether your changes corrected the problem.

1.4. Simulation with IBIS Models

IBIS models provide a way to run accurate signal integrity simulations quickly. IBIS models describe the behavior of I/O buffers with voltage-current and voltage-time data curves.

Because of their behavioral nature, IBIS models do not have to include any information about the internal circuit design of the I/O buffer. Most component manufacturers, including Intel, provide IBIS models for free download and use in signal integrity analysis simulation tools. You can download generic device family IBIS models from the Intel website for early design simulation or use the IBIS Writer to create custom IBIS models for your existing design.

1.4.1. Elements of an IBIS Model

An IBIS model file (\*ibs) is a text file that describes the behavior of an I/O buffer across minimum, typical, and maximum temperature and voltage ranges with a specified test load.

The tables and values specified in the IBIS file describe five basic elements of the I/O buffer.
The following elements correspond to each numbered block.

1. **Pulldown**—A voltage-current table describes the current when the buffer is driven low based on a pull-down voltage range of \(-V_{CC}\) to \(2V_{CC}\).

2. **Pullup**—A voltage-current table describes the current when the buffer is driven high based on a pull-up voltage range of \(-V_{CC}\) to \(V_{CC}\).

3. **Ground and Power Clamps**—Voltage-current tables describe the current when clamping diodes for electrostatic discharge (ESD) are present. The ground clamp voltage range is \(-V_{CC}\) to \(V_{CC}\), and the power clamp voltage range is \(-V_{CC}\) to ground.

4. **Ramp and Rising/Falling Waveform**—A voltage-time (dv/dt) ratio describes the rise and fall time of the buffer during a logic transition. Optional rising and falling waveform tables can be added to more accurately describe the characteristics of the rising and falling transitions.

5. **Total Output Capacitance and Package RLC**—The total output capacitance includes the parasitic capacitances of the output pad, clamp diodes (if present), and input transistors. The package RLC is device package-specific and defines the resistance, inductance, and capacitance of the bond wire and pin of the I/O.

**Related Information**

AN 283: Simulating Intel Devices with IBIS Models
For more information about IBIS models and Intel-specific features, including links to the official IBIS specification.

### 1.4.2. Creating Accurate IBIS Models

There are two methods to obtain Intel device IBIS files for your board-level signal integrity simulations. You can download generic IBIS models from the Intel website. You can also use the IBIS writer in the Intel Quartus Prime software to create design-specific models.

The IBIS file generated by the Intel Quartus Prime software contains models of both input and output termination, and is supported for IBIS model versions of 4.2 and later. Arria® V, Cyclone® V, and Stratix® V device families allow the use of bidirectional I/O with dynamic on-chip termination (OCT).

Dynamic OCT is used where a signal uses a series on-chip termination during output operation and a parallel on-chip termination during input operation. Typically this is used in Altera External Memory Interface IP.
The Intel Quartus Prime IBIS dynamic OCT IBIS model names end in g50c_r50c. For example: sstl15i_ctnio_g50c_r50c.

In the simulation tool, the IBIS model is attached to a buffer.
- When the buffer is assigned as an output, use the series termination r50c.
- When the buffer is assigned as an input, use the parallel termination g50c.

1.4.2.1. Download IBIS Models

Intel provides IBIS models for almost all FPGA and FPGA configuration devices. You can use the IBIS models from the website to perform early simulations of the I/O buffers you expect to use in your design as part of a pre-layout analysis.

Downloaded IBIS models have the RLC package values set to one particular device in each device family.

The .ibs file can be customized for your device package and can be used for any simulation. IBIS models downloaded and used for simulations in this manner are generic. They describe only a certain set of models listed for each device on the Intel IBIS Models page of the Intel website. To create customized models for your design, use the IBIS Writer as described in the next section.

To simulate your design with the model accurately, you must adjust the RLC values in the IBIS model file to match the values for your particular device package by performing the following steps:

1. Download and expand the ZIP file (.zip) of the IBIS model for the device family you are using for your design. The .zip file contains the .ibs file along with an IBIS model user guide and a model data correlation report.
2. Download the Package RLC Values spreadsheet for the same device family.
3. Open the spreadsheet and locate the row that describes the device package used in your design.
4. From the package’s I/O row, copy the minimum, maximum, and typical values of resistance, inductance, and capacitance for your device package.
5. Open the .ibs file in a text editor and locate the [Package] section of the file.
6. Overwrite the listed values copied with the values from the spreadsheet and save the file.

Related Information
Intel IBIS Models
For information about whether models for your selected device are available.

1.4.2.2. Generate Custom IBIS Models with the IBIS Writer

If you have started your FPGA design and have created custom I/O assignments, you can use the Intel Quartus Prime IBIS Writer to create custom IBIS models to accurately reflect your assignments.

Examples of custom assignments include drive strength settings or the enabling of clamping diodes for ESD protection. IBIS models created with the IBIS Writer take I/O assignment settings into account.
If the **Enable Advanced I/O Timing** option is turned off, the generated `.ibs` files are based on the load value setting for each I/O standard on the **Capacitive Loading** page of the **Device and Pin Options** dialog box in the **Device** dialog box. With the **Enable Advanced I/O Timing** option turned on, IBIS models use an effective capacitive load based on settings found in the board trace model on the **Board Trace Model** page in the **Device and Pin Options** dialog box or the **Board Trace Model** view in the Pin Planner. The effective capacitive load is based on the sum of the **Near capacitance**, **Transmission line distributed capacitance**, and the **Far capacitance** settings in the board trace model. Resistance values and transmission line inductance values are ignored.

**Note:** If you made any changes from the default load settings, the delay in the generated IBIS model cannot safely be added to the Intel Quartus Prime $t_{CO}$ measurement to account for the double counting problem. This is because the load values between the two delay measurements do not match. When this happens, the Intel Quartus Prime software displays warning messages when the EDA Netlist Writer runs to remind you about the load value mismatch.

**Related Information**
- Intel IBIS models
- Generating IBIS Output Files with the Intel Quartus Prime Software
  - In Intel Quartus Prime Help
- AN 283: Simulating Intel Devices with IBIS Models

### 1.4.3. Design Simulation Using the Mentor Graphics HyperLynx Software

You must integrate IBIS models downloaded from the Intel website or created with the Intel Quartus Prime IBIS Writer into board design simulations to accurately model timing and signal integrity.

The HyperLynx software from Mentor Graphics is one of the most popular tools for design simulation. The HyperLynx software makes it easy to integrate IBIS models into simulations.

The HyperLynx software is a PCB analysis and simulation tool for high-speed designs, consisting of two products, LineSim and BoardSim.

**LineSim** is an early simulation tool. Before any board routing takes place, you use LineSim to simulate “what if” scenarios that assist in creating routing rules and defining board parameters.

**BoardSim** is a post-layout tool that you use to analyze existing board routing. You select one or more nets from a board layout file and BoardSim simulates those nets in a manner similar to LineSim. With board and routing parameters, and surrounding signal routing known, highly accurate simulations of the final fabricated PCB are possible.

This section focuses on LineSim. Because the process of creating and running simulations is very similar for both LineSim and BoardSim, the details of IBIS model use in LineSim applies to simulations in BoardSim.

You configure simulations in LineSim using a schematic GUI to create connections and topologies between I/O buffers, route trace segments, and termination components. LineSim provides two methods for creating routing schematics: cell-based and free-form. Cell-based schematics are based on fixed cells consisting of typical placements...
of buffers, trace impedances, and components. Parts of the grid-based cells are filled with the desired objects to create the topology. A topology in a cell-based schematic is limited by the available connections within and between the cells.

A more robust and expandable way to create a circuit schematic for simulation is to use the free-form schematic format in LineSim. The free-form schematic format makes it easy to place parts into any configuration and edit them as required. This section describes the use of IBIS models with free-form schematics, but the process is nearly identical for cell-based schematics.

**Figure 3. HyperLynx LineSim Free-Form Schematic Editor**

When you use HyperLynx software to perform simulations, you typically perform the following steps:

1. Create a new LineSim free-form schematic document and set up the board stackup for your PCB using the Stackup Editor. In this editor, specify board layer properties including layer thickness, dielectric constant, and trace width.

2. Create a circuit schematic for the net you want to simulate. The schematic represents all the parts of the routed net including source and destination I/O buffers, termination components, transmission line segments, and representations of impedance discontinuities such as vias or connectors.

3. Assign IBIS models to the source and destination I/O buffers to represent their behavior during operation.

4. Attach probes from the digital oscilloscope that is built in to LineSim to points in the circuit that you want to monitor during simulation. Typically, at least one probe is attached to the pin of a destination I/O buffer. For differential signals, you can attach a differential probe to both the positive and negative pins at the destination.
5. Configure and run the simulation. You can simulate a rising or falling edge and test the circuit under different drive strength conditions.

6. Interpret the results and make adjustments. Based on the waveforms captured in the digital oscilloscope, you can adjust anything in the circuit schematic to correct any signal integrity issues, such as overshoot or ringing. If necessary, you can make I/O assignment changes in the Intel Quartus Prime software, regenerate the IBIS file with the IBIS Writer, and apply the updated IBIS model to the buffers in your HyperLynx software schematic.

7. Repeat the simulations and circuit adjustments until you are satisfied with the results.

8. When the operation of the net meets your design requirements, implement changes to your I/O assignments in the Intel Quartus Prime software and optionally adjust your board routing constraints, component values, and placement to match the simulation.

For more information about HyperLynx software, including schematic creation, simulation setup, model usage, product support, licensing, and training, refer to the Mentor Graphics webpage.

Related Information
www.mentor.com

1.4.4. Configuring LineSim to Use Intel IBIS Models

You must configure LineSim to find and use the downloaded or generated IBIS models for your design. To do this, add the location of your .ibs file or files to the LineSim Model Library search path. Then you apply a selected model to a buffer in your schematic.

To add the Intel Quartus Prime software’s default IBIS model location, `<project directory>/board/ibis`, to the HyperLynx LineSim model library search path, perform the following steps in LineSim:

1. From the Options menu, click **Directories**. The **Set Directories** dialog box appears. The **Model-library file path(s)** list displays the order in which LineSim searches file directories for model files.
Figure 4.  **LineSim Set Directories Dialog Box**

2. Click **Edit**. A dialog box appears where you can add directories and adjust the order in which LineSim searches them.

Figure 5.  **LineSim Select Directories Dialog Box**

3. Click **Add**

4. Browse to the default IBIS model location, `<project directory>/board/ibis`. Click **OK**.

5. Click **Up** to move the IBIS model directory to the top of the list. Click **Generate Model Index** to update LineSim’s model database with the models found in the added directory.

6. Click **OK**. The IBIS model directory for your project is added to the top of the Model-library file path(s) list.

7. To close the **Set Directories** dialog box, click **OK**.
1.4.5. Integrating Intel IBIS Models into LineSim Simulations

When the location for IBIS files has been set, you can assign the downloaded or generated IBIS models to the buffers in your schematic. To do this, perform the following steps:

1. Double-click a buffer symbol in your schematic to open the Assign Models dialog box. You can also click Assign Models from the buffer symbol’s right-click menu.

Figure 6. LineSim Assign Model Dialog Box

2. The pin of the buffer symbol you selected should be highlighted in the Pins list. If you want to assign a model to a different symbol or pin, select it from the list.

3. Click Select. The Select IC Model dialog box appears.
4. To filter the list of available libraries to display only IBIS models, select .IBS. Scroll through the Libraries list, and click the name of the library for your design. By default, this is `<project name>.ibs`.

5. The device for your design should be selected as the only item in the Devices list. If not, select your device from the list.

6. From the Signal list, select the name of the signal you want to simulate. You can also choose to select by device pin number.

7. Click OK. The Assign Models dialog box displays the selected .ibs file and signal.

8. If applicable to the signal you chose, adjust the buffer settings as required for the simulation.

9. Select and configure other buffer pins from the Pins list in the same manner.

10. Click OK when all I/O models are assigned.

1.4.6. Running and Interpreting LineSim Simulations

You can run any simulation and make adjustments to the I/O assignments or simulation parameters as required.

For example, if you see too much overshoot in the simulated signal at the destination buffer after running a simulation, you can adjust the drive strength I/O assignment setting to a lower value. Regenerate the .ibs file, and run the simulation again to verify whether the change fixes the problem.
If you see a discontinuity or other anomalies at the destination, such as slow rise and fall times, adjust the termination scheme or termination component values. After making these changes, rerun the simulation to check whether your adjustments solved the problem. In this case, it is not necessary to regenerate the .ibs file.

For more information about board-level signal integrity, and to learn about ways to improve it with simple changes to your design, visit the Intel FPGA Signal & Power Integrity Support Center.

**Related Information**

Intel Signal & Power Integrity Center
1.5. Simulation with HSPICE Models

HSPICE decks are used to perform highly accurate simulations by describing the physical properties of all aspects of a circuit precisely. HSPICE decks describe I/O buffers, board components, and all the connections between them, as well as defining the parameters of the simulation to be run.

By their nature, HSPICE decks are highly customizable and require a detailed description of the circuit under simulation. For devices that support advanced I/O timing, when Enable Advanced I/O Timing is turned on, the HSPICE decks generated by the Intel Quartus Prime HSPICE Writer automatically include board components and topology defined in the Board Trace Model. Configure the board components and topology in the Pin Planner or in the Board Trace Model tab of the Device and Pin Options dialog box. All HSPICE decks generated by the Intel Quartus Prime software include compensation for the double count problem. You can simulate with the default simulation parameters built in to the generated HSPICE decks or make adjustments to customize your simulation.

Related Information
The Double Counting Problem in HSPICE Simulations on page 21

1.5.1. Supported Devices and Signaling

The HSPICE Writer in the Intel Quartus Prime software supports Arria, Cyclone, and Stratix devices for the creation of a board trace model in the Intel Quartus Prime software for automatic inclusion in an HSPICE deck.

The HSPICE files include the board trace description you create in the Board Trace Model view in the Pin Planner or the Board Trace Model tab in the Device and Pin Options dialog box.

Note: Note that for Intel Arria 10 devices, you may need to download the Encrypted HSPICE model from the Intel website.

Related Information
• I/O Management
  For information about how to use the Enable Advanced I/O Timing option and configure board trace models for the I/O standards used in your design.
• SPICE Models for Intel Devices
  For more information about the Encrypted HSPICE model.

1.5.2. Accessing HSPICE Simulation Kits

You can access the available HSPICE models with the Intel Quartus Prime software’s HSPICE Writer tool and also at the Spice Models for Intel Devices web page.

The Intel Quartus Prime software HSPICE Writer tool removes many common sources of user error from the I/O simulation process. The HSPICE Writer tool automatically creates preconfigured I/O simulation spice decks that only require the addition of a user board model. All the difficult tasks required to configure the I/O modes and interpret the timing results are handled automatically by the HSPICE Writer tool.
1. Signal Integrity Analysis with Third-Party Tools

1.5.3. The Double Counting Problem in HSPICE Simulations

Simulating I/Os using accurate models is extremely helpful for finding and fixing FPGA I/O timing and board signal integrity issues before any boards are built. However, the usefulness of such simulations is directly related to the accuracy of the models used and whether the simulations are set up and performed correctly.

To ensure accuracy in models and simulations created for FPGA output signals you must consider the timing hand-off between t\textsubscript{CO} timing in the Intel Quartus Prime software and simulation-based board delay. If this hand-off is not handled correctly, the calculated delay could either count some of the delay twice or even miss counting some of the delay entirely.

1.5.3.1. Defining the Double Counting Problem

The double counting problem is inherent to the difference between the method to analyze output timing in the Intel Quartus Prime software versus the method HSPICE models use. The timing analyzer tools in the Intel Quartus Prime software measure delay timing for an output signal from the core logic of the FPGA design through the output buffer, ending at the FPGA pin with a default capacitive load or a specified value for the I/O standard you selected. This measurement is the t\textsubscript{CO} timing variable.

Figure 10. Double Counting Problem

HSPICE models for board simulation measure t\textsubscript{PD} (propagation delay) from an arbitrary reference point in the output buffer, through the device pin, out along the board routing, and ending at the signal destination. If you add these two delays, the delay between the output buffer and the device pin appears twice in the calculation. A model or simulation that does not account for this double count creates overly pessimistic simulation results, because the double-counted delay can limit I/O performance artificially.
One approach to fix the problem is subtracting the overlap between $t_{CO}$ and $t_{PD}$ to account for the double count. However, this adjustment is not accurate, because each measurement considers a different load.

*Note:* Input signals do not exhibit this problem, because the HSPICE models for inputs stop at the FPGA pin instead of at the input buffer. In this case, adding the delays together produces an accurate measurement of delay timing.

### 1.5.3.2. The Solution to Double Counting

To adjust the measurements to account for the double-counting, the delay between the arbitrary point in the output buffer selected by the HSPICE model and the FPGA pin must be subtracted from either $t_{CO}$ or $t_{PD}$ before adding the results together. The subtracted delay must also be based on a common load between the two measurements. This is done by repeating the HSPICE model measurement, but with the same load used by the Intel Quartus Prime software for the $t_{CO}$ measurement.

**Figure 11.** Common Test Loads Used for Output Timing

With $t_{TESTLOAD}$ known, the total delay is calculated for the output signal from the FPGA logic to the signal destination on the board, accounting for the double count.

$$t_{delay} = t_{CO} + (t_{PD} - t_{TESTLOAD})$$

The preconfigured simulation files generated by the HSPICE Writer in the Intel Quartus Prime software are designed to account for the double-counting problem based on this calculation automatically.
1.5.4. HSPICE Writer Tool Flow

This section includes information to help you get started using the Intel Quartus Prime software HSPICE Writer tool. The information in this section assumes you have a basic knowledge of the standard Intel Quartus Prime software design flow, such as project and assignment creation, compilation, and timing analysis.

1.5.4.1. Applying I/O Assignments

The first step in the HSPICE Writer tool flow is to configure the I/O standards and modes for each of the pins in your design properly. In the Intel Quartus Prime software, these settings are represented by assignments that map I/O settings, such as pin selection, and I/O standard and drive strength, to corresponding signals in your design.

The Intel Quartus Prime software provides multiple methods for creating these assignments:

- Using the Pin Planner
- Using the assignment editor
- Manually editing the .qsf file
- By making assignments in a scripted Intel Quartus Prime flow using Tcl

1.5.4.2. Enabling HSPICE Writer

You must enable the HSPICE Writer in the Settings dialog box of the Intel Quartus Prime software to generate the HSPICE decks from the Intel Quartus Prime software.

Figure 12. EDA Tool Settings: Board Level Options Dialog Box
1.5.4.3. Enabling HSPICE Writer Using Assignments

You can also use HSPICE Writer in conjunction with a scripted Tcl flow. To enable HSPICE Writer during a full compile, include the following lines in your Tcl script.

Enable HSPICE Writer

```tcl
set_global_assignment -name EDA_BOARD_DESIGN_SIGNAL_INTEGRITY_TOOL "HSPICE (Signal Integrity)"
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT HSPICE -section_id eda_board_design_signal_integrity
set_global_assignment -name EDA_NETLIST_WRITER_OUTPUT_DIR <output_directory> -section_id eda_board_design_signal_integrity
```

As with command-line invocation, specifying the output directory is optional. If not specified, the output directory defaults to `board/hspice`.

1.5.4.4. Naming Conventions for HSPICE Files

HSPICE Writer automatically generates simulation files and names them using the following naming convention: `<device>_<pin #>_<pin_name>_<in/out>.sp`.

For bidirectional pins, two spice decks are produced; one with the I/O buffer configured as an input, and the other with the I/O buffer configured as an output.

The Intel Quartus Prime software supports alphanumeric pin names that contain the underscore ( `_`) and dash ( `-`) characters. Any illegal characters used in file names are converted automatically to underscores.

Related Information
- Sample Output for I/O HSPICE Simulation Deck on page 34
- Sample Input for I/O HSPICE Simulation Deck on page 30

1.5.4.5. Invoking HSPICE Writer

After HSPICE Writer is enabled, the HSPICE simulation files are generated automatically each time the project is completely compiled. The Intel Quartus Prime software also provides an option to generate a new set of simulation files without having to recompile manually. In the Processing menu, click Start EDA Netlist Writer to generate new simulation files automatically.

*Note:* You must perform both Analysis & Synthesis and Fitting on a design before invoking the HSPICE Writer tool.

1.5.4.6. Invoking HSPICE Writer from the Command Line

If you use a script-based flow to compile your project, you can create HSPICE model files by including the following commands in your Tcl script (.tcl file).

Create HSPICE Model Files

```tcl
set_global_assignment -name EDA_BOARD_DESIGN_SIGNAL_INTEGRITY_TOOL "HSPICE (Signal Integrity)"
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT HSPICE -section_id eda_board_design_signal_integrity
set_global_assignment -name EDA_NETLIST_WRITER_OUTPUT_DIR <output_directory> -section_id eda_board_design_signal_integrity
```
The `<output_directory>` option specifies the location where HSPICE model files are saved. By default, the `<project directory>/board/hspice` directory is used.

**Invoke HSPICE Writer**

To invoke the HSPICE Writer tool through the command line, type:

```bash
quartus_eda.exe <project_name> --board_signal_integrity=on --format=HSPICE \
--output_directory=<output_directory>
```

`<output_directory>` specifies the location where the tool writes the generated spice decks, relative to the design directory. This is an optional parameter and defaults to `board/hspice`.

### 1.5.4.7. Customizing Automatically Generated HSPICE Decks

HSPICE models generated by the HSPICE Writer can be used for simulation as generated.

A default board description is included, and a default simulation is set up to measure rise and fall delays for both input and output simulations, which compensates for the double counting problem. However, Intel recommends that you customize the board description to more accurately represent your routing and termination scheme.

The sample board trace loading in the generated HSPICE model files must be replaced by your actual trace model before you can run a correct simulation. To do this, open the generated HSPICE model files for all pins you want to simulate and locate the following section.

#### Sample Board Trace Section

```plaintext
* I/O Board Trace and Termination Description
* - Replace this with your board trace and termination description
```

You must replace the example load with a load that matches the design of your PCB board. This includes a trace model, termination resistors, and, for output simulations, a receiver model. The spice circuit node that represents the pin of the FPGA package is called `pin`. The node that represents the far pin of the external device is called `load-in` (for output SPICE decks) and `source-in` (for input SPICE decks).

For an input simulation, you must also modify the stimulus portion of the spice file. The section of the file that must be modified is indicated in the following comment block.

#### Sample Source Stimulus Section

```plaintext
* Sample source stimulus placeholder
* - Replace this with your I/O driver model
```

Replace the sample stimulus model with a model for the device that drives the FPGA.

### 1.5.5. Running an HSPICE Simulation

Because simulation parameters are configured directly in the HSPICE model files, running a simulation requires only that you open an HSPICE file in the HSPICE user interface and start the simulation.
Click **Open** and browse to the location of the HSPICE model files generated by the Intel Quartus Prime HSPICE Writer. The default location for HSPICE model files is `<project directory>/board/hspice`. Select the `.sp` file generated by the HSPICE Writer for the signal you want to simulate. Click **OK**.

To run the simulation, click **Simulate**. The status of the simulation is displayed in the window and saved in an `.lis` file with the same name as the `.sp` file when the simulation is complete. Check the `.lis` file if an error occurs during the simulation requiring a change in the `.sp` file to fix.

### 1.5.6. Interpreting the Results of an Output Simulation

By default, the automatically generated output simulation spice decks are set up to measure three delays for both rising and falling transitions. Two of the measurements, `tpd_rise` and `tpd_fall`, measure the double-counting corrected delay from the FPGA pin to the load pin. To determine the complete clock-edge to load-pin delay, add these numbers to the Intel Quartus Prime software reported default loading `tCO` delay.

The remaining four measurements, `tpd_uncomp_rise`, `tpd_uncomp_fall`, `t_dblcnt_rise`, and `t_dblcnt_fall`, are required for the double-counting compensation process and are not required for further timing usage.

**Related Information**

Simulation Analysis on page 34

### 1.5.7. Interpreting the Results of an Input Simulation

By default, the automatically generated input simulation SPICE decks are set up to measure delays from the source's driver pin to the FPGA's input pin for both rising and falling transitions.

The propagation delay is reported by HSPICE measure statements as `tpd_rise` and `tpd_fall`. To determine the complete source driver pin-to-FPGA register delay, add these numbers to the Intel Quartus Prime software reported `TH` and `TSU` input timing numbers.
1.5.8. Viewing and Interpreting Tabular Simulation Results

The .lis file stores the collected simulation data in tabular form. The default simulation configured by the HSPICE Writer produces delay measurements for rising and falling transitions on both input and output simulations.

These measurements are found in the .lis file and named tpd_rise and tpd_fall. For output simulations, these values are already adjusted for the double count. To determine the complete delay from the FPGA logic to the load pin, add either of these measurements to the Intel Quartus Prime tCO delay. For input simulations, add either of these measurements to the Intel Quartus Prime tSU and tH delay values to calculate the complete delay from the far end stimulus to the FPGA logic. Other values found in the .lis file, such as tpd_uncomp_rise, tpd_uncomp_fall, t_dblcnt_rise, and t_dblcnt_fall, are parts of the double count compensation calculation. These values are not necessary for further analysis.

1.5.9. Viewing Graphical Simulation Results

You can view the results of the simulation quickly as a graphical waveform display using the AvanWaves viewer included with HSPICE. With the default simulation configured by the HSPICE Writer, you can view the simulated waveforms at both the source and destination in input and output simulations.

To see the waveforms for the simulation, in the HSPICE user interface window, click AvanWaves. The AvanWaves viewer opens and displays the Results Browser.

**Figure 14.** HSPICE AvanWaves Results Browser

The Results Browser lets you select which waveform to view quickly in the main viewing window. If multiple simulations are run on the same signal, the list at the top of the Results Browser displays the results of each simulation. Click the simulation...
description to select which simulation to view. By default, the descriptions are derived from the first line of the HSPICE file, so the description might appear as a line of asterisks.

Select the type of waveform to view, by performing the following steps:

1. To see the source and destination waveforms with the default simulation, from the **Types** list, select **Voltages**.

2. On the **Curves** list, double-click the waveform you want to view. The waveform appears in the main viewing window.

You can zoom in and out and adjust the view as desired.

**Figure 15. AvanWaves Waveform Viewer**

1.5.10. Making Design Adjustments Based on HSPICE Simulations

Based on the results of your simulations, you can make adjustments to the I/O assignments or simulation parameters if required. For example, after you run a simulation and see overshoot or ringing in the simulated signal at the destination buffer, you can adjust the drive strength I/O assignment setting to a lower value. Regenerate the HSPICE deck, and run the simulation again to verify that the change fixed the problem.
If there is a discontinuity or any other anomalies at the destination, adjust the board description in the Intel Quartus Prime Board Trace Model, or in the generated HSPICE model files to change the termination scheme or adjust termination component values. After making these changes, regenerate the HSPICE files if necessary, and rerun the simulation to verify whether your adjustments solved the problem.
For more information about board-level signal integrity and to learn about ways to improve it with simple changes to your FPGA design, visit the Intel Signal & Power Integrity Center

**Related Information**

Intel Signal & Power Integrity Center

### 1.5.11. Sample Input for I/O HSPICE Simulation Deck

The following sections examine a typical HSPICE simulation spice deck for an I/O of type input. Each section presents the simulation file one block at a time.

#### 1.5.11.1. Header Comment

The first block of an input simulation spice deck is the header comment. The purpose of this block is to provide an easily readable summary of how the simulation file has been automatically configured by the Intel Quartus Prime software.

This block has two main components: The first component summarizes the I/O configuration relevant information such as device, speed grade, and so on. The second component specifies the exact test condition that the Intel Quartus Prime software assumes for the given I/O standard.
Sample Header Comment Block

* Intel Quartus Prime HSPICE Writer I/O Simulation Deck *

* This spice simulation deck was automatically generated by Quartus for the following IO settings:

* Device: EP2S60F1020C3
* Speed Grade: C3
* Pin: AA4 (out96)
* Bank: IO Bank 6 (Row I/O)
* I/O Standard: LVTTL, 12mA
* OCT: Off

* Intel Quartus Prime’s default I/O timing delays assume the following slow corner simulation conditions.

* Specified Test Conditions For Intel Quartus Prime Tco
* Temperature: 85C (Slowest Temperature Corner)
* Transistor Model: TT (Typical Transistor Corner)
* Vccn: 3.135V (Vccn_min = Nominal - 5%)
* Vccpd: 2.97V (Vccpd_min = Nominal - 10%)
* Load: No Load
* Vtt: 1.5675V (Voltage reference is Vccn/2)

* Note: The I/O transistors are specified to operate at least as fast as the TT transistor corner, actual production devices can be as fast as the FF corner. Any simulations for hold times should be conducted using the fast process corner with the following simulation conditions.

* Temperature: 0C (Fastest Commercial Temperature Corner **)
* Transistor Model: FF (Fastest Transistor Corner)
* Vccn: 1.98V (Vccn_hold = Nominal + 10%)
* Vccpd: 3.63V (Vccpd_hold = Nominal + 10%)
* Vtt: 0.95V (Vtt_hold = Vccn/2 - 40mV)
* Vcc: 1.25V (Vcc_hold = Maximum Recommended)
* Package Model: Short-circuit from pad to pin (no parasitics)

* Warnings:

1.5.11.2. Simulation Conditions

The simulation conditions block loads the appropriate process corner models for the transistors. This condition is automatically set up for the slow timing corner and is modified only if other simulation corners are desired.

Simulation Conditions Block

* Process Settings

.options brief
.inc 'sii_tt.inc' * TT process corner

1.5.11.3. Simulation Options

The simulation options block configures the simulation temperature and configures HSPICE with typical simulation options.

Simulation Options Block

* Simulation Options

.options brief=0
.options badchr co=132 scale=1e-6 acct ingold=2 nomod dv=1.0
1.5.11.4. Constant Definition

The constant definition block of the simulation file instantiates the voltage sources that control the configuration modes of the I/O buffer.

**Constant Definition Block**

```
* Constant Definition
voeb       oeb       0     vc  * Set to 0 to enable buffer output
vopdrain   opdrain   0     0   * Set to vc to enable open drain
vrambh     rambh     0     0   * Set to vc to enable bus hold
vrpullup   rpullup   0     0   * Set to vc to enable weak pullup
vpcdp5     rpcdp5    0     rp5 * Set the IO standard
vpcdp4     rpcdp4    0     rp4
vpcdp3     rpcdp3    0     rp3
vpcdp2     rpcdp2    0     rp2
vpcdp1     rpcdp1    0     rp1
vpcdp0     rpcdp0    0     rp0
vpcdn4     rpcdn4    0     rn4
vpcdn3     rpcdn3    0     rn3
vpcdn2     rpcdn2    0     rn2
vpcdn1     rpcdn1    0     rn1
vpcdn0     rpcdn0    0     rn0
vdin din       0     0
```

Where:
- Voltage source `voeb` controls the output enable of the buffer and is set to disabled for inputs.
- `vopdrain` controls the open drain mode for the I/O.
- `vrambh` controls the bus hold circuitry in the I/O.
- `vrpullup` controls the weak pullup.
- The next 11 voltages sources control the I/O standard of the buffer and are configured through a later library call.
- `vdin` is not used on input pins because it is the data pin for the output buffer.

1.5.11.5. Buffer Netlist

The buffer netlist block of the simulation spice deck loads all the load models required for the corresponding input pin.

**Buffer Netlist Block**

```
* IO Buffer Netlist
.include 'vio_buffer.inc'
```

1.5.11.6. Drive Strength

The drive strength block of the simulation SPICE deck loads the configuration bits necessary to configure the I/O into the proper I/O standard and drive strengths.
Although these settings are not relevant to an input buffer, they are provided to allow the SPICE deck to be modifiable to support bidirectional simulations.

**Drive Strength Block**

* Drive Strength Settings

`.lib 'drive_select_hio.lib' 3p3ttl_12ma`

### 1.5.11.7. I/O Buffer Instantiation

The I/O buffer instantiation block of the simulation SPICE deck instantiates the necessary power supplies and I/O model components that are necessary to simulate the given I/O.

**I/O Buffer Instantiation**

* Supply Voltages Settings

```plaintext
.param vcn=3.135
.param vpd=2.97
.param vc=1.15
```

* Instantiate Power Supplies

```plaintext
vvcc vcc 0 vc * FPGA core voltage
vvss vss 0 0 * FPGA core ground
vvccn vccn 0 vcn * IO supply voltage
vvssn vssn 0 0 * IO ground
vvccpd vccpd 0 vpd * Pre-drive supply voltage
```

* Instantiate I/O Buffer

```plaintext
xvio_buf din oeb opdrain die rambh
+ rpcdn4 rpcdn3 rpcdn2 rpcdn1 rpcdn0
+ rpcdp5 rpcdp4 rpcdp3 rpcdp2 rpcdp1 rpcdp0
+ rpullup vccn vccpd vcpad0 vio_buf
```

* Internal Loading on Pad

* - No loading on this pad due to differential buffer/support

* circuitry

* I/O Buffer Package Model

* - Single-ended I/O standard on a Row I/O

`.lib 'lib/package.lib' hio
xpkg die pin hio_pkg`

### 1.5.11.8. Board Trace and Termination

The board trace and termination block of the simulation SPICE deck is provided only as an example. Replace this block with your own board trace and termination models.

**Board Trace and Termination Block**

* I/O Board Trace and Termination Description

* - Replace this with your board trace and termination description

```plaintext
wtline pin vssn load vssn N=1 L=1 RLGMODEL=tlinemodel
.MODEL tlinemodel W MODELTYPE=RLGC N=1 Lo=7.13n Co=2.85p
Rterm2 load vssn 1x
```
1.5.11.9. Stimulus Model

The stimulus model block of the simulation spice deck is provided only as a place holder example. Replace this block with your own stimulus model. Options for this include an IBIS or HSPICE model, among others.

**Stimulus Model Block**

* Sample source stimulus placeholder
  * - Replace this with your I/O driver model

```
Vsource source 0 pulse(0 vcn 0s 0.4ns 0.4ns 8.5ns 17.4ns)
```

1.5.11.10. Simulation Analysis

The simulation analysis block of the simulation file is configured to measure the propagation delay from the source to the FPGA pin. Both the source and end point of the delay are referenced against the 50% $V_{CCN}$ crossing point of the waveform.

**Simulation Analysis Block**

* Simulation Analysis Setup

```
.print tran v(source) v(pin)
.tran 0.020ns 17ns

* Measure the propagation delay from the source pin to the pin referenced against the 50% voltage threshold crossing point

.measure TRAN tpd_rise TRIG v(source) val='vcn*0.5' rise=1
  + TARG v(pin) val ='vcn*0.5' rise=1

.measure TRAN tpd_fall TRIG v(source) val='vcn*0.5' fall=1
  + TARG v(pin) val ='vcn*0.5' fall=1
```

1.5.12. Sample Output for I/O HSPICE Simulation Deck

A typical HSPICE simulation SPICE deck for an I/O-type output has several sections. Each section presents the simulation file one block at a time.

1.5.12.1. Header Comment

The first block of an output simulation SPICE deck is the header comment. The purpose of this block is to provide a readable summary of how the simulation file has been automatically configured by the Intel Quartus Prime software.

This block has two main components:

- The first component summarizes the I/O configuration relevant information such as device, speed grade, and so on.
- The second component specifies the exact test condition that the Intel Quartus Prime software assumes when generating $t_{CO}$ delay numbers. This information is used as part of the double-counting correction circuitry contained in the simulation file.

The SPICE decks are preconfigured to calculate the slow process corner delay but can also be used to simulate the fast process corner as well. The fast corner conditions are listed in the header under the notes section.
The final section of the header comment lists any warning messages that you must consider when you use the SPICE decks.

Header Comment Block

* Intel Quartus Prime HSPICE Writer I/O Simulation Deck
* This spice simulation deck was automatically generated by
* Intel Quartus Prime for the following IO settings:
  * Device: EP2S60F1020C3
  * Speed Grade: C3
  * Pin: AA4 (out96)
  * Bank: IO Bank 6 (Row I/O)
  * I/O Standard: LVTTL, 12mA
  * OCT: Off

* Quartus’ default I/O timing delays assume the following slow corner simulation conditions.
* Specified Test Conditions For Intel Quartus Prime Tco
  * Temperature: 85C (Slowest Temperature Corner)
  * Transistor Model: TT (Typical Transistor Corner)
  * Vccn: 3.135V (Vccn_min = Nominal - 5%)  
  * Vccpd: 2.97V (Vccpd_min = Nominal - 10%)
  * Load: No Load
  * Vtt: 1.5675V (Voltage reference is Vccn/2)
  * For C3 devices, the TT transistor corner provides an approximation for worst case timing. However, for functionality simulations, it is recommended that the SS corner be simulated as well.

* Note: The I/O transistors are specified to operate at least as fast as the TT transistor corner, actual production devices can be as fast as the FF corner. Any simulations for hold times should be conducted using the fast process corner with the following simulation conditions.
  * Temperature: 0C (Fastest Commercial Temperature Corner **)
  * Transistor Model: FF (Fastest Transistor Corner)
  * Vccn: 1.98V (Vccn_hold = Nominal + 10%)
  * Vccpd: 3.63V (Vccpd_hold = Nominal + 10%)
  * Vtt: 0.95V (Vtt_hold = Vccn/2 - 40mV)
  * Vcc: 1.25V (Vcc_hold = Maximum Recommended)
  * Package Model: Short-circuit from pad to pin

1.5.12.2. Simulation Conditions

The simulation conditions block loads the appropriate process corner models for the transistors. This condition is automatically set up for the slow timing corner and must be modified only if other simulation corners are desired.

Simulation Conditions Block

* Process Settings
  .options brief
  .inc `sii_tt.inc` * typical-typical process corner

Note: Two separate corners cannot be simulated at the same time. Instead, simulate the base case using the Quartus corner as one simulation and then perform a second simulation using the desired customer corner. The results of the two simulations can be manually added together.
1. Signal Integrity Analysis with Third-Party Tools

1.5.12.3. Simulation Options

The simulation options block configures the simulation temperature and configures HSPICE with typical simulation options.

**Simulation Options Block**

```
* Simulation Options
.options brief=0
.options badchr co=132 scale=1e-6 acct ingold=2 nomod dv=1.0
+        dcstep=1 absv=1e-3 absi=1e-8 probe csdf=2 accurate=1
+        converge=1
.temp 85
```

**Note:** For a detailed description of these options, consult your HSPICE manual.

1.5.12.4. Constant Definition

The constant definition block of the output simulation SPICE deck instantiates the voltage sources that controls the configuration modes of the I/O buffer.

**Constant Definition Block**

```
* Constant Definition
voeb       oeb       0     0 * Set to 0 to enable buffer output
vopdrain   opdrain   0     0 * Set to vc to enable open drain
vrambh     rambh     0     0 * Set to vc to enable bus hold
vropullup   rpullup   0     0 * Set to vc to enable weak pullup
vpci       rpci      0     0 * Set to vc to enable pci mode
vpcdp4     rpcdp4    0     rp4  * These control bits set the IO standard
vpcdp3     rpcdp3    0     rp3
vpcdp2     rpcdp2    0     rp2
vpcdp1     rpcdp1    0     rp1
vpcdp0     rpcdp0    0     rp0
vpcdn4     rpcdn4    0     rn4
vpcdn3     rpcdn3    0     rn3
vpcdn2     rpcdn2    0     rn2
vpcdn1     rpcdn1    0     rn1
vpcdn0     rpcdn0    0     rn0
vdin       din       0     pulse(0 vc 0s 0.2ns 0.2ns 8.5ns 17.4ns)
```

**Where:**

- **Voltage source** voeb controls the output enable of the buffer.
- vopdrain controls the open drain mode for the I/O.
- vrambh controls the bus hold circuitry in the I/O.
- vropullup controls the weak pullup.
- vpci controls the PCI clamp.
- The next ten voltage sources control the I/O standard of the buffer and are configured through a later library call.
- vdin is connected to the data input of the I/O buffer.
- The edge rate of the input stimulus is automatically set to the correct value by the Intel Quartus Prime software.
1.5.12.5. I/O Buffer Netlist

The I/O buffer netlist block loads all of the models required for the corresponding pin. These include a model for the I/O output buffer, as well as any loads that might be present on the pin.

I/O Buffer Netlist Block

```
*IO Buffer Netlist
.include 'hio_buffer.inc'
.include 'lvds_input_load.inc'
.include 'lvds_oct_load.inc'
```

1.5.12.6. Drive Strength

The drive strength block of the simulation spice deck loads the configuration bits for configuring the I/O to the proper I/O standard and drive strength. These options are set by the HSPICE Writer tool and are not changed for expected use.

Drive Strength Block

```
* Drive Strength Settings
.lib 'drive_select_hio.lib' 3p3ttl_12ma
```

1.5.12.7. Slew Rate and Delay Chain

Stratix and Cyclone devices have sections for configuring the slew rate and delay chain settings.

Slew Rate and Delay Chain Settings

```
* Programmable Output Delay Control Settings
.lib 'lib/output_delay_control.lib' no_delay
* Programmable Slew Rate Control Settings
.lib 'lib/slew_rate_control.lib' slow_slow
```

1.5.12.8. I/O Buffer Instantiation

The I/O buffer instantiation block of the output simulation spice deck instantiates the necessary power supplies and I/O model components that are necessary to simulate the given I/O.

I/O Buffer Instantiation Block

```
* I/O Buffer Instantiation

* Supply Voltages Settings
.param vcn=3.135
.param vpd=2.97
.param vc=1.15

* Instantiate Power Supplies
vvcc vcc 0 vc  * FPGA core voltage
vvss vss 0 0  * FPGA core ground
vvccn vccn 0 vcn  * IO supply voltage
vvssn vssn 0 0  * IO ground
vvccpd vccpd 0 vpd  * Pre-drive supply voltage
```
1.5.12.9. Board and Trace Termination

The board trace and termination block of the simulation SPICE deck is provided only as an example. Replace this block with your specific board loading models.

**Board Trace and Termination Block**

* I/O Board Trace And Termination Description
  * - Replace this with your board trace and termination description

wtline pin vssn load vssn N=1 L=1 RLGC
.MODEL tlinemodel W MODELTYPE=RLGC N=1 L=7.13n C=2.85p
Rterm2 load vssn 1x

1.5.12.10. Double-Counting Compensation Circuitry

The double-counting compensation circuitry block of the simulation SPICE deck instantiates a second I/O buffer that is used to measure double-counting. The buffer is configured identically to the user I/O buffer but is connected to the Intel Quartus Prime software test load. The simulated delay of this second buffer can be interpreted as the amount of double-counting between the Intel Quartus Prime software and HSPICE Writer simulated results.

As the amount of double-counting is constant for a given I/O standard on a given pin, consider separating the double-counting circuitry from the simulation file. In doing so, you can perform any number of I/O simulations while referencing the delay only once.

**(Part of )Double-Counting Compensation Circuitry Block**

* Double Counting Compensation Circuitry
  *
  * The following circuit is designed to calculate the amount of double counting between Intel Quartus Prime and the HSPICE models. If you have not changed the default simulation temperature or transistor corner this spice deck automatically compensates the double counting. In the event you wish to simulate an IO at a different temperature or transistor corner you need to remove this section of code and manually account for double counting. A description of Intel’s recommended procedure for this process can be found in the Intel Quartus Prime HSPICE Writer AppNote.

* Supply Voltages Settings
  .param vcn_tl=3.135
  .param vpd_tl=2.97
1. Signal Integrity Analysis with Third-Party Tools

1.5.12.11. Simulation Analysis

The simulation analysis block is set up to measure double-counting corrected delays. This is accomplished by measuring the uncompensated delay of the I/O buffer when connected to the user load, and when subtracting the simulated amount of double-counting from the test load I/O buffer.

Simulation Analysis Block

* Simulation Analysis Setup

*Print out the voltage waveform at both the pin and far end load
.print tran v(pin) v(load)
.tran 0.020ns 17ns

* Measure the propagation delay to the load pin. This value
* includes some double counting with Intel Quartus Prime's Tco
.measure TRAN tpd uncomp_rise TRIG v(din) val='vc*0.5' rise=1+ TARG v(load)
  val='vcn*0.5' rise=1
.measure TRAN tpd uncomp_fall TRIG v(din) val='vc*0.5' fall=1
 + TARG v(load) val='vcn*0.5' fall=1

* The test load buffer can calculate the amount of double counting
.measure TRAN tdblcnt_rise TRIG v(din) val='vc*0.5' rise=1
 + TARG v(pin_tl) val='vcn_tl*0.5' rise=1
.measure TRAN tdblcnt_fall TRIG v(din) val='vc*0.5' fall=1
 + TARG v(pin_tl) val='vcn_tl*0.5' fall=1

* Calculate the true propagation delay by subtraction
.measure TRAN tpd_rise PARAM='tpd uncomp_rise-t dblcnt_rise'
.measure TRAN tpd fall PARAM='tpd uncomp fall-t dblcnt fall'
1.5.13. Advanced Topics

The information in this section describes some of the more advanced topics and methods employed when setting up and running HSPICE simulation files.

1.5.13.1. PVT Simulations

The automatically generated HSPICE simulation files are set up to simulate the slow process corner using low voltage, high temperature, and slow transistors. To ensure a fully robust link, Intel recommends that you run simulations over all process corners.

To perform process, voltage, and temperature (PVT) simulations, manually modify the spice decks in a two step process:

1. Remove the double-counting compensation circuitry from the simulation file. This is required as the amount of double-counting is dependent upon how the Intel Quartus Prime software calculates delays and is not based on which PVT corner is being simulated. By default, the Intel Quartus Prime software provides timing numbers using the slow process corner.

2. Select the proper corner for the PVT simulation by setting the correct HSPICE temperature, changing the supply voltage sources, and loading the correct transistor models.

A more detailed description of HSPICE process corners can be found in the family-specific HSPICE model documentation.

Related Information

Accessing HSPICE Simulation Kits on page 20

1.5.13.2. Hold Time Analysis

Intel recommends performing worst-case hold time analysis using the fast corner models, which use fast transistors, high voltage, and low temperature. This involves modifying the SPICE decks to select the correct temperature option, change the supply voltage sources, and load the correct fast transistor models. The values of these parameters are located in the header comment section of the corresponding simulation deck files.

For a truly worst-case analysis, combine the HSPICE Writer hold time analysis results with the Intel Quartus Prime software fast timing model. This requires that you change the double-counting compensation circuitry in the simulations files to also simulate the fast process corners, as this is what the Intel Quartus Prime software uses for the fast timing model.

Note: This method of hold time analysis is recommended only for globally synchronous buses. Do not apply this method of hold-time analysis to source synchronous buses. This is because the source synchronous clocking scheme is designed to cancel out some of the PVT timing effects. If this is not taken into account, the timing results are not accurate. Proper source synchronous timing analysis is beyond the scope of this document.

1.5.13.3. I/O Voltage Variations

Use each of the FPGA family datasheets to verify the recommended operating conditions for supply voltages. For current FPGA families, the maximum recommended voltage corresponds to the fast corner, while the minimum recommended voltage
corresponds to the slow corner. These voltage recommendations are specified at the power pins of the FPGA and are not necessarily the same voltage that are seen by the I/O buffers due to package IR drops.

The automatically generated HSPICE simulation files model this IR effect pessimistically by including a 50-mV IR drop on the $V_{CCPD}$ supply when a high drive strength standard is being used.

1.5.13.4. Correlation Report

Correlation reports for the HSPICE I/O models are located in the family-specific HSPICE I/O buffer simulation kits.

**Related Information**

[Accessing HSPICE Simulation Kits on page 20](#)

### 1.6. Signal Integrity Analysis with Third-Party Tools Document Revision History

**Table 2. Document Revision History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017.11.06</td>
<td>17.1.0</td>
<td>- Reorganized chapter introduction.</td>
</tr>
<tr>
<td>2016.10.31</td>
<td>16.1.0</td>
<td>- Implemented Intel rebranding.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Corrected statement about timing simulation and double counting.</td>
</tr>
<tr>
<td>2015.11.02</td>
<td>15.1.0</td>
<td>- Changed instances of Quartus II to Intel Quartus Prime.</td>
</tr>
<tr>
<td>June 2014</td>
<td>14.0.0</td>
<td>Updated format.</td>
</tr>
<tr>
<td>December 2010</td>
<td>10.0.1</td>
<td>Template update.</td>
</tr>
<tr>
<td>July 2010</td>
<td>10.0.0</td>
<td>Updated device support.</td>
</tr>
<tr>
<td>November 2009</td>
<td>9.1.0</td>
<td>No change to content.</td>
</tr>
<tr>
<td>March 2009</td>
<td>9.0.0</td>
<td>- Was volume 3, chapter 12 in the 8.1.0 release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- No change to content.</td>
</tr>
<tr>
<td>November 2008</td>
<td>8.1.0</td>
<td>- Changed to 8-1/2 x 11 page size.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added information for Stratix III devices.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Input signals for Cyclone III devices are supported.</td>
</tr>
<tr>
<td>May 2008</td>
<td>8.0.0</td>
<td>- Updated &quot;Introduction&quot; on page 12–1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated Figure 12–1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated Figure 12–3.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated Figure 12–13.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated &quot;Output File Generation&quot; on page 12–6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated &quot;Simulation with HSPICE Models&quot; on page 12–17.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated &quot;Invoking HSPICE Writer from the Command Line&quot; on page 12–22.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added &quot;Sample Output for I/O HSPICE Simulation Deck&quot; on page 12–33.</td>
</tr>
</tbody>
</table>

continued...
<table>
<thead>
<tr>
<th>Date</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Updated &quot;Correlation Report&quot; on page 12–41.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added hyperlinks to referenced documents and websites throughout the chapter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Made minor editorial updates.</td>
</tr>
</tbody>
</table>
2. Reviewing Printed Circuit Board Schematics with the Intel Quartus Prime Software

Intel FPGAs and CPLDs offer a multitude of configurable options to allow you to implement a custom application-specific circuit on your PCB.

Your Intel Quartus Prime project provides important information specific to your programmable logic design, which you can use in conjunction with the device literature available on Altera's website to ensure that you implement the correct board-level connections in your schematic.

Refer to the Settings dialog box options, the Fitter report, and Messages window when creating and reviewing your PCB schematic. The Intel Quartus Prime software also provides the Pin Planner to assist you during your PCB schematic review process.

Related Information
Schematic Review Worksheets

2.1. Reviewing Intel Quartus Prime Software Settings

Review these settings in the Intel Quartus Prime software to help you review your PCB schematic.

The Device dialog box in the Intel Quartus Prime software allows you to specify device-specific assignments and settings. You can use the Device dialog box to specify general project-wide options, including specific device and pin options, which help you to implement correct board-level connections in your PCB schematic.

The Device dialog box provides project-specific device information, including the target device and any migration devices you specify. Using migration devices can impact the number of available user I/O pins and internal resources, as well as require connection of some user I/O pins to power/ground pins to support migration.

If you want to use vertical migration, which allows you to use different devices with the same package, you can specify your list of migration devices in the Migration Devices dialog box. The Fitter places the pins in your design based on your targeted migration devices, and allows you to use only I/O pins that are common to all the migration devices.

If a migration device has pins that are power or ground, but the pins are also user I/O pins on a different device in the migration path, the Fitter ensures that these pins are not used as user I/O pins. You must ensure that these pins are connected to the appropriate plane on the PCB.

If you are migrating from a smaller device with NC (no-connect) pins to a larger device with power or ground pins in the same package, you can safely connect the NC pins to power or ground pins to facilitate successful migration.
2.1.1. Device and Pins Options Dialog Box Settings

You can set device and pin options and verify important design-specific data in the Device and Pin Options dialog box, including options found on the General, Configuration, Unused Pin, Dual-Purpose Pins, and Voltage pages.

2.1.1.1. Configuration Settings

The Configuration page of the Device and Pin Options dialog box specifies the configuration scheme and configuration device for the target device. Use the Configuration page settings to verify the configuration scheme with the MSEL pin settings used on your PCB schematic and the I/O voltage of the configuration scheme.

Your specific configuration settings may impact the availability of some dual-purpose I/O pins in user mode.

Related Information
Dual-Purpose Pins Settings on page 44

2.1.1.2. Unused Pin Settings

The Unused Pin page specifies the behavior of all unused pins in your design. Use the Unused Pin page to ensure that unused pin settings are compatible with your PCB.

For example, if you reserve all unused pins as outputs driving ground, you must ensure that you do not connect unused I/O pins to VCC pins on your PCB. Connecting unused I/O pins to VCC pins may result in contention that could lead to higher than expected current draw and possible device overstress.

The Reserve all unused pins list shows available unused pin state options for the target device. The default state for each pin is the recommended setting for each device family.

When you reserve a pin as output driving ground, the Fitter connects a ground signal to the output pin internally. You should connect the output pin to the ground plane on your PCB, although you are not required to do so. Connecting the output driving ground to the ground plane is known as creating a virtual ground pin, which helps to minimize simultaneous switching noise (SSN) and ground bounce effects.

2.1.1.3. Dual-Purpose Pins Settings

The Dual-Purpose Pins page specifies how configuration pins should be used after device configuration completes. You can set the function of the dual-purpose pins by selecting a value for a specific pin in the Dual-purpose pins list. Pin functions should match your PCB schematic. The available options on the Dual-Purpose Pins page may differ depending on the selected configuration mode.
2.1.1.4. Voltage Settings

The Voltage page specifies the default VCCIO I/O bank voltage and the default I/O bank voltage for the pins on the target device. VCCIO I/O bank voltage settings made in the Voltage page are overridden by I/O standard assignments made on I/O pins in their respective banks.

Ensure that the settings in the Voltage page match the settings in your PCB schematic, especially if the target device includes transceivers.

The Voltage page settings requirements differ depending on the settings of the transceiver instances in the design. Refer to the Fitter report for the required settings, and verify that the voltage settings are correctly set up for your PCB schematic.

After verifying your settings in the Device and Settings dialog boxes, you can verify your device pin-out with the Fitter report.

Related Information
Reviewing Device Pin-Out Information in the Fitter Report on page 45

2.1.1.5. Error Detection CRC Settings

The Error Detection CRC page specifies error detection cyclic redundancy check (CRC) use for the target device. When Enable error detection CRC is turned on, the device checks the validity of the programming data in the devices. Any changes made in the data while the device is in operation generates an error.

Turning on the Enable open drain on CRC error pin option allows the CRC ERROR pin to be set as an open-drain pin in some devices, which decouples the voltage level of the CRC ERROR pin from VCCIO voltage. You must connect a pull-up resistor to the CRC ERROR pin on your PCB if you turn on this option.

In addition to settings in the Device dialog box, you should verify settings in the Voltage page of the Settings dialog box.

Related Information
Device and Pin Options Dialog Box
In Intel Quartus Prime Help

2.2. Reviewing Device Pin-Out Information in the Fitter Report

After you compile your design, you can use the reports in the Resource section of the Fitter report to check your device pin-out in detail.

The Input Pins, Output Pins, and Bidirectional Pins reports identify all the user I/O pins in your design and the features enabled for each I/O pin. For example, you can find use of weak internal pull-ups, PCI clamp diodes, and on-chip termination (OCT) pin assignments in these sections of the Fitter report. You can check the pin assignments reported in the Input Pins, Output Pins, and Bidirectional Pins reports against your PCB schematic to determine whether your PCB requires external components.

These reports also identify whether you made pin assignments or if the Fitter automatically placed the pins. If the Fitter changed your pin assignments, you should make these changes user assignments because the location of pin assignments made by the Fitter may change with subsequent compilations.
Figure 18. **Resource Section Report**

Open the **Compilation Report** tab with Ctrl+R, In the **Plan Stage** folder under **Fitter** open the Input Pins report. The following figure shows the pins the Fitter chose for the OCT external calibration resistor connections (RUP/RDN) and the name of the associated termination block in the Input Pins report. You should make these types of assignments user assignments.

![Input Pins Report](image)

The I/O Bank Usage report provides a high-level overview of the VCCIO and VREF requirements for your design, based on your I/O assignments. Verify that the requirements in this report match the settings in your PCB schematic. All unused I/O banks, and all banks with I/O pins with undefined I/O standards, default the VCCIO voltage to the voltage defined in the **Voltage** page of the **Device and Pin Options** dialog box.

The All Package Pins report lists all the pins on your device, including unused pins, dedicated pins and power/ground pins. You can use this report to verify pin characteristics, such as the location, name, usage, direction, I/O standard and voltage for each pin with the pin information in your PCB schematic. In particular, you should verify the recommended voltage levels at which you connect unused dedicated inputs and I/O and power pins, especially if you selected a migration device. Use the All Package Pins report to verify that you connected all the device voltage rails to the voltages reported.

Errors commonly reported include connecting the incorrect voltage to the predriver supply (VCCPD) pin in a specific bank, or leaving dedicated clock input pins floating. Unused input pins that should be connected to ground are designated as **GND+** in the **Pin Name/Usage** column in the All Package Pins report.

You can also use the All Package Pins report to check transceiver-specific pin connections and verify that they match the PCB schematic. Unused transceiver pins have the following requirements, based on the pin designation in the Fitter report:

- **GXB_GND**—Unused GXB receiver or dedicated reference clock pin. This pin must be connected to GXB_GND through a 10k Ohm resistor.
- **GXB_NC**—Unused GXB transmitter or dedicated clock output pin. This pin must be disconnected.
Some transceiver power supply rails have dual voltage capabilities, such as VCCA_L/R and VCCH_L/R, that depend on the settings you created for the ALTGX parameter editor. Because these user-defined settings overwrite the default settings, you should use the All Package Pins report to verify that these power pins on the device symbol in the PCB schematics are connected to the voltage required by the transceiver. An incorrect connection may cause the transceiver to function not as expected.

If your design includes a memory interface, the DQS Summary report provides an overview of each DQ pin group. You can use this report to quickly confirm that the correct DQ/DQS pins are grouped together.

Finally, the Fitter Device Options report summarizes some of the settings made in the Device and Pin Options dialog box. Verify that these settings match your PCB schematics.

### 2.3. Reviewing Compilation Error and Warning Messages

If your project does not compile without error or warning messages, you should resolve the issues identified by the Compiler before signing off on your pin-out or PCB schematic. Error messages often indicate illegal or unsupported use of the device resources and IP.

Additionally, you should cross-reference fitting and timing analysis warnings with the design implementation. Timing may be constrained due to nonideal pin placement. You should investigate if you can reassign pins to different locations to prevent fitting and timing analysis warnings. Ensure that you review each warning and consider its potential impact on the design.

### 2.4. Using Additional Intel Quartus Prime Software Features

You can generate IBIS files, which contain models specific to your design and selected I/O standards and options, with the Intel Quartus Prime software.

Because board-level simulation is important to verify, you should check for potential signal integrity issues. You can turn on the Board-Level Signal Integrity feature in the EDA Tool Settings page of the Settings dialog box.

Additionally, using advanced I/O timing allows you to enter physical PCB information to accurately model the load seen by an output pin. This feature facilitates accurate I/O timing analysis.

**Related Information**

- Signal Integrity Analysis with Third-Party Tools on page 4
- Managing Device I/O Pins

### 2.5. Using Additional Intel Quartus Prime Software Tools

Use the Pin Planner to assist you with reviewing your PCB schematics.
2.5.1. Pin Planner

The Intel Quartus Prime Pin Planner helps you visualize, plan, and assign device I/O pins in a graphical view of the target device package. You can quickly locate various I/O pins and assign them design elements or other properties to ensure compatibility with your PCB layout.

You can use the Pin Planner to verify the location of clock inputs, and whether they have been placed on dedicated clock input pins, which is recommended when your design uses PLLs.

You can also use the Pin Planner to verify the placement of dedicated SERDES pins. SERDES receiver inputs can be placed only on DIFFIO_RX pins, while SERDES transmitter outputs can be placed only on DIFFIO_TX pins.

The Pin Planner gives a visual indication of signal-to-signal proximity in the Pad View window, and also provides information about differential pin pair placement, such as the placement of pseudo-differential signals.

Related Information
Managing Device I/O Pins

2.6. Reviewing Printed Circuit Board Schematics with the Intel Quartus Prime Software Revision History

Table 3. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.09.24</td>
<td>18.1</td>
<td>• First release as part of the stand-alone Intel Quartus Prime Standard Edition User Guide</td>
</tr>
<tr>
<td>2016.10.31</td>
<td>16.1</td>
<td>• Implemented Intel rebranding.</td>
</tr>
<tr>
<td>2015.11.02</td>
<td>15.1</td>
<td>• Changed instances of Quartus II to Intel Quartus Prime.</td>
</tr>
<tr>
<td>June 2014</td>
<td>14.0</td>
<td>• Template update.</td>
</tr>
<tr>
<td>November 2012</td>
<td>12.1</td>
<td>• Minor update of Pin Planner description for task and report windows.</td>
</tr>
<tr>
<td>June 2012</td>
<td>12.0</td>
<td>• Removed survey link.</td>
</tr>
<tr>
<td>November 2011</td>
<td>10.0</td>
<td>• Template update.</td>
</tr>
<tr>
<td>December 2010</td>
<td>10.0</td>
<td>• Changed to new document template. No change to content.</td>
</tr>
<tr>
<td>July 2010</td>
<td>10.0</td>
<td>• Initial release.</td>
</tr>
</tbody>
</table>

You can integrate the Mentor Graphics DxDesigner® PCB design tool into the Intel Quartus Prime design flow.

With today’s large, high-pin-count and high-speed FPGA devices, good and correct PCB design practices are essential to ensure correct system operation. The PCB design takes place concurrently with the design and programming of the FPGA. The FPGA or ASIC designer initially creates signal and pin assignments, and the board designer must correctly transfer these assignments to the symbols in their system circuit schematics and board layout. As the board design progresses, Intel recommends reassigning pins to optimize the PCB layout. Ensure that you inform the FPGA designer of the pin reassignments so that the new assignments are included in an updated placement and routing of the design.

This chapter covers the following topics:

- Mentor Graphics and Intel Quartus Prime software integration flow
- Generating supporting files
- Creating DxDesigner symbols from the Intel Quartus Prime output files

This chapter is intended for board design and layout engineers who want to start the FPGA board integration while the FPGA is still in the design phase. Alternatively, the board designer can plan the FPGA pin-out and routing requirements in the Mentor Graphics tools and pass the information back to the Intel Quartus Prime software for placement and routing. Part librarians can also benefit from this chapter by learning how to use output from the Intel Quartus Prime software to create new library parts and symbols.

The procedures in this chapter require the following software:

- The Intel Quartus Prime software version 5.1 or later
- DxDesigner software version 2004 or later

Note: To obtain and license the Mentor Graphics tools and for product information, support, and training, refer to the Mentor Graphics website.

3.1. Integrating with DxDesigner

You can integrate the Mentor Graphics DxDesigner schematic capture tool into the Intel Quartus Prime design flow. Use DxDesigner to create flat circuit schematics or to create hierarchical schematics that facilitate design reuse and a team-based design for all PCB types.
If you use DxDesigner without the I/O Designer software, the design flow is one-way, using only the .pin generated by the Intel Quartus Prime software. You can only make signal and pin assignment changes in the Intel Quartus Prime software. You cannot back-annotate changes made in a board layout tool or in a DxDesigner symbol to the Intel Quartus Prime software.

Figure 19. DxDesigner-only Flow

3.1.1. DxDesigner Project Settings

DxDesigner new projects automatically create FPGA symbols by default. To enable the DxBoardLink flow design configuration when creating a new DxDesigner project, follow these steps:

1. Start the DxDesigner software.
2. Click File ➤ New, and then click the Project tab.
3. Click More. Turn on DxBoardLink. To enable the DxBoardLink Flow design configuration for an existing project, click Design Configurations in the Design Configuration toolbar and turn on DxBoardLink.

3.1.2. Creating Schematic Symbols in DxDesigner

You can create schematic symbols in the DxDesigner software manually or with the Symbol wizard. The DxDesigner Symbol wizard is similar to the I/O Designer Symbol wizard, but with fewer fracturing options. The DxDesigner Symbol wizard creates, fractures, and edits FPGA symbols based on the specified Intel FPGA device. To create a symbol with the Symbol wizard, follow these steps;

1. Start the DxDesigner software.
2. Click Symbol Wizard in the toolbar.
3. Type the new symbol name in the name field and click OK.
4. Specify creation of a new symbol or modification of an existing symbol. To modify an existing symbol, specify the library path or alias, and select the existing symbol. To create a new symbol, select DxBoardLink for the symbol source. The DxDesigner block type defaults to Module because the FPGA design does not have an underlying DxDesigner schematic. Choose whether or not to fracture the symbol. Click Next.
5. Type a name for the symbol, an overall part name for all the symbol fractures, and a library name for the new library created for this symbol. By default, the part and library names are the same as the symbol name. Click **Next**.

6. Specify the appearance of the generated symbol in your DxDesigner project schematic. After making your selections. Click **Next**.

7. In the **FPGA vendor** list, select **Intel Quartus**. In the **Pin-Out file to import** field, select the `.pin` from your Intel Quartus Prime project directory. You can also specify Fracturing Scheme, Bus pin, and Power pin options. Click **Next**.

8. Select to create or modify symbol attributes for use in the DxDesigner software. Click **Next**.

9. On the **Pin Settings** page, make any final adjustments to pin and label location and information. Each tabbed spreadsheet represents a fracture of your symbol. Click **Save Symbol**.

After creating the symbol, you can examine and place any fracture of the symbol in your schematic. You can locate separate files of all the fractures you created in the library you specified or created in the `/sym` directory in your DxDesigner project. You can add the symbols to your schematics or you can manually edit the symbols or with the Symbol wizard.

### 3.2. Mentor Graphics PCB Design Tools Support Revision History

**Table 4. Document Revision History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.11.04</td>
<td>18.1</td>
<td>Removed references to unsupported <code>.fx</code> files and related tools.</td>
</tr>
<tr>
<td>2016.10.31</td>
<td>16.1</td>
<td>• Implemented Intel rebranding.</td>
</tr>
<tr>
<td>2015.11.02</td>
<td>15.1</td>
<td>• Changed instances of Quartus II to Intel Quartus Prime.</td>
</tr>
<tr>
<td>2014.06.30</td>
<td>14.0</td>
<td>• Replaced MegaWizard Plug-In Manager information with IP Catalog.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added standard information about upgrading IP cores.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added standard installation and licensing information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Removed outdated device support level information. IP core device support is now available in IP Catalog and parameter editor.</td>
</tr>
<tr>
<td>June 2012</td>
<td>12.0</td>
<td>• Removed survey link.</td>
</tr>
<tr>
<td>December 2010</td>
<td>10.1</td>
<td>• Changed to new document template.</td>
</tr>
</tbody>
</table>
4. Cadence Board Design Tools Support

4.1. Cadence PCB Design Tools Support

The Intel Quartus Prime software interacts with the following software to provide a complete FPGA-to-board integration design workflow: the Cadence Allegro Design Entry HDL software and the Cadence Allegro Design Entry CIS (Component Information System) software (also known as OrCAD Capture CIS). The information is useful for board design and layout engineers who want to begin the FPGA board integration process while the FPGA is still in the design phase. Part librarians can also benefit by learning the method to use output from the Intel Quartus Prime software to create new library parts and symbols.

With today’s large, high-pin-count and high-speed FPGA devices, good PCB design practices are important to ensure the correct operation of your system. The PCB design takes place concurrently with the design and programming of the FPGA. An FPGA or ASIC designer initially creates the signal and pin assignments and the board designer must transfer these assignments to the symbols used in their system circuit schematics and board layout correctly. As the board design progresses, you must perform pin reassignments to optimize the layout. You must communicate pin reassignments to the FPGA designer to ensure the new assignments are processed through the FPGA with updated placement and routing.

You require the following software:

- The Intel Quartus Prime software version 5.1 or later
- The Cadence Allegro Design Entry HDL software or the Cadence Allegro Design Entry CIS software version 15.2 or later
- The OrCAD Capture software with the optional CIS option version 10.3 or later (optional)

**Note:** These programs are very similar because the Cadence Allegro Design Entry CIS software is based on the OrCAD Capture software. Any procedural information can also apply to the OrCAD Capture software unless otherwise noted.

**Related Information**

- [www.cadence.com](http://www.cadence.com)  
  For more information about obtaining and licensing the Cadence tools and for product information, support, and training
- [www.orcad.com](http://www.orcad.com)  
  For more information about the OrCAD Capture software and the CIS option
- [www.ema-eda.com](http://www.ema-eda.com)  
  For more information about Cadence and OrCAD support and training.
4.2. Product Comparison

Table 5. Cadence and OrCAD Product Comparison

<table>
<thead>
<tr>
<th>Description</th>
<th>Cadence Allegro Design Entry HDL</th>
<th>Cadence Allegro Design Entry CIS</th>
<th>OrCAD Capture CIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Former Name</td>
<td>Concept HDL Expert</td>
<td>Capture CIS Studio</td>
<td>—</td>
</tr>
<tr>
<td>History</td>
<td>More commonly known by its former name, Cadence renamed all board design tools in 2004 under the Allegro name.</td>
<td>Based directly on OrCAD Capture CIS, the Cadence Allegro Design Entry CIS software is still developed by OrCAD but sold and marketed by Cadence. EMA provides support and training.</td>
<td>The basis for Design Entry CIS is still developed by OrCAD for continued use by existing OrCAD customers. EMA provides support and training for all OrCAD products.</td>
</tr>
<tr>
<td>Vendor Design Flow</td>
<td>Cadence Allegro 600 series, formerly known as the Expert Series, for high-end, high-speed design.</td>
<td>Cadence Allegro 200 series, formerly known as the Studio Series, for small- to medium-level design.</td>
<td>—</td>
</tr>
</tbody>
</table>

Related Information
- [www.cadence.com](http://www.cadence.com)
- [www.ema-eda.com](http://www.ema-eda.com)

4.3. FPGA-to-PCB Design Flow

You can create a design flow integrating an Intel FPGA design from the Intel Quartus Prime software through a circuit schematic in the Cadence Allegro Design Entry HDL software or the Cadence Allegro Design Entry CIS software.
Figure 20. Design Flow with the Cadence Allegro Design Entry HDL Software

Start FPGA Design

Create or Change Pin Assignments

Run I/O Assignment Analysis

Intel Quartus Prime Software

Run Full Compilation

Part Developer

Start PCB Design (Allegro Design Entry HDL)

Create or Open a Project

Project Manager

Import or Update Pin Assignments

Create or Update FPGA Symbol

Edit Symbol

Design Entry HDL

Instantiate Symbol in Schematic

Forward to Board Layout Tool

Board Layout Tool

Layout and Route FPGA

End
To create FPGA symbols using the Cadence Allegro PCB Librarian Part Developer tool, you must obtain the Cadence PCB Librarian Expert license. You can update symbols with changes made to the FPGA design using any of these tools.

4.3.1. Integrating Intel FPGA Designs

To integrate an Intel FPGA design starting in the Intel Quartus Prime software through to a circuit schematic in the Cadence Allegro Design Entry HDL software or the Cadence Allegro Design Entry CIS software, follow these steps:

1. In the Intel Quartus Prime software, compile your design to generate a Pin-Out File (.pin) to transfer the assignments to the Cadence software.

2. If you are using the Cadence Allegro Design Entry HDL software for your schematic design, follow these steps:
   a. Open an existing project or create a new project in the Cadence Allegro Project Manager tool.
   b. Construct a new symbol or update an existing symbol using the Cadence Allegro PCB Librarian Part Developer tool.
   c. With the Cadence Allegro PCB Librarian Part Developer tool, edit your symbol or fracture it into smaller parts (optional).
   d. Instantiate the symbol in your Cadence Allegro Design Entry HDL software schematic and transfer the design to your board layout tool.

3. If you are using the Cadence Allegro Design Entry CIS software for your schematic design, follow these steps:
a. Generate a new part in a new or existing Cadence Allegro Design Entry CIS project, referencing the .pin output file from the Intel Quartus Prime software. You can also update an existing symbol with a new .pin.

b. Split the symbol into smaller parts as necessary.

c. Instantiate the symbol in your Cadence Allegro Design Entry CIS schematic and transfer the design to your board layout tool.

4.4. Setting Up the Intel Quartus Prime Software

You can transfer pin and signal assignments from the Intel Quartus Prime software to the Cadence design tools by generating the Intel Quartus Prime project .pin. The .pin is an output file generated by the Intel Quartus Prime Fitter containing pin assignment information. You can use the Intel Quartus Prime Pin Planner to set and change the assignments in the .pin and then transfer the assignments to the Cadence design tools. You cannot, however, import pin assignment changes from the Cadence design tools into the Intel Quartus Prime software with the .pin.

The .pin lists all used and unused pins on your selected Intel device. The .pin also provides the following basic information fields for each assigned pin on the device:

- Pin signal name and usage
- Pin number
- Signal direction
- I/O standard
- Voltage
- I/O bank
- User or Fitter-assigned

4.4.1. Generating a .pin File

To generate a .pin, follow these steps:

1. Compile your design.
2. Locate the .pin in your Intel Quartus Prime project directory with the name <project name>.pin.

4.5. FPGA-to-Board Integration with the Cadence Allegro Design Entry HDL Software

The Cadence Allegro Design Entry HDL software is a schematic capture tool and is part of the Cadence 600 series design flow. Use the Cadence Allegro Design Entry HDL software to create flat circuit schematics for all types of PCB design. The Cadence Allegro Design Entry HDL software can also create hierarchical schematics to facilitate design reuse and team-based design. With the Cadence Allegro Design Entry HDL software, the design flow from FPGA-to-board is one-way, using only the .pin generated by the Intel Quartus Prime software. You can only make signal and pin assignment changes in the Intel Quartus Prime software and these changes reflect as updated symbols in a Cadence Allegro Design Entry HDL project.
For more information about the design flow with the Cadence Allegro Design Entry HDL software, refer to Figure 20 on page 54.

**Note:** Routing or pin assignment changes made in a board layout tool or a Cadence Allegro Design Entry HDL software symbol cannot be back-annotated to the Intel Quartus Prime software.

**Related Information**

[www.cadence.com](http://www.cadence.com)

Provides information about the Cadence Allegro Design Entry HDL software and the Cadence Allegro PCB Librarian Part Developer tool, including licensing, support, usage, training, and product updates.

### 4.5.1. Creating Symbols

In addition to circuit simulation, circuit board schematic creation is one of the first tasks required when designing a new PCB. Schematics must understand how the PCB works, and to generate a netlist for a board layout tool for board design and routing. The Cadence Allegro PCB Librarian Part Developer tool allows you to create schematic symbols based on FPGA designs exported from the Intel Quartus Prime software.

You can create symbols for the Cadence Allegro Design Entry HDL project with the Cadence Allegro PCB Librarian Part Developer tool, which is available in the Cadence Allegro Project Manager tool. Intel recommends using the Cadence Allegro PCB Librarian Part Developer tool to import FPGA designs into the Cadence Allegro Design Entry HDL software.

You must obtain a PCB Librarian Expert license from Cadence to run the Cadence Allegro PCB Librarian Part Developer tool. The Cadence Allegro PCB Librarian Part Developer tool provides a GUI with many options for creating, editing, fracturing, and updating symbols. If you do not use the Cadence Allegro PCB Librarian Part Developer tool, you must create and edit symbols manually in the Symbol Schematic View in the Cadence Allegro Design Entry HDL software.

**Note:** If you do not have a PCB Librarian Expert license, you can automatically create FPGA symbols using the programmable IC (PIC) design flow found in the Cadence Allegro Project Manager tool.

Before creating a symbol from an FPGA design, you must open a Cadence Allegro Design Entry HDL project with the Cadence Allegro Project Manager tool. If you do not have an existing Cadence Allegro Design Entry HDL project, you can create one with the Cadence Allegro Design Entry HDL software. The Cadence Allegro Design Entry HDL project directory with the name <project name>.cpm contains your Cadence Allegro Design Entry HDL projects.

While the Cadence Allegro PCB Librarian Part Developer tool refers to symbol fractures as slots, the other tools use different names to refer to symbol fractures.

**Table 6. Symbol Fracture Naming Conventions**

<table>
<thead>
<tr>
<th></th>
<th>Cadence Allegro PCB Librarian Part Developer Tool</th>
<th>Cadence Allegro Design Entry HDL Software</th>
<th>Cadence Allegro Design Entry CIS Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>During symbol generation</td>
<td>Slots</td>
<td>—</td>
<td>Sections</td>
</tr>
<tr>
<td>During symbol schematic instantiation</td>
<td>—</td>
<td>Versions</td>
<td>Parts</td>
</tr>
</tbody>
</table>
4.5.1.1. Cadence Allegro PCB Librarian Part Developer Tool

You can create, fracture, and edit schematic symbols for your designs using the Cadence Allegro PCB Librarian Part Developer tool. Symbols designed in the Cadence Allegro PCB Librarian Part Developer tool can be split or fractured into several functional blocks called slots, allowing multiple smaller part fractures to exist on the same schematic page or across multiple pages.

4.5.1.1.1. Cadence Allegro PCB Librarian Part Developer Tool in the Design Flow

To run the Cadence Allegro PCB Librarian Part Developer tool, you must open a Cadence Allegro Design Entry HDL project in the Cadence Allegro Project Manager tool. To open the Cadence Allegro PCB Librarian Part Developer tool, on the Flows menu, click Library Management, and then click Part Developer.

Related Information
FPGA-to-PCB Design Flow on page 53

4.5.1.1.2. Import and Export Wizard

After starting the Cadence Allegro PCB Librarian Part Developer tool, use the Import and Export wizard to import your pin assignments from the Intel Quartus Prime software.

Note: Intel recommends using your PCB Librarian Expert license file. To point to your PCB Librarian Expert license file, on the File menu, click Change Product and then select the correct product license.
To access the Import and Export wizard, follow these steps:

1. On the File menu, click **Import and Export**.

2. Select **Import ECO-FPGA**, and then click **Next**.

3. In the **Select Source** page of the **Import and Export** wizard, specify the following settings:
   a. In the **Vendor** list, select **Altera**.
   b. In the **PnR Tool** list, select **quartusII**.
   c. In the **PR File** box, browse to select the **.pin** in your Intel Quartus Prime project directory.
   d. Click **Simulation Options** to select simulation input files.
   e. Click **Next**.

4. In the **Select Destination** dialog box, specify the following settings:
   a. Under **Select Component**, click **Generate Custom Component** to create a new component in a library, or click **Use standard component** to base your symbol on an existing component.

   **Note:** Intel recommends creating a new component if you previously created a generic component for an FPGA device. Generic components can cause some problems with your design. When you create a new component, you can place your pin and signal assignments from the Intel Quartus Prime software on this component and reuse the component as a base when you have a new FPGA design.

   b. In the **Library** list, select an existing library. You can select from the cells in the selected library. Each cell represents all the symbol versions and part fractures for a particular part. In the **Cell** list, select the existing cell to use as a base for your part.

   c. In the **Destination Library** list, select a destination library for the component. Click **Next**.

   d. Review and edit the assignments you import into the Cadence Allegro PCB Librarian Part Developer tool based on the data in the **.pin** and then click **Finish**. The location of each pin is not included in the **Preview of Import Data** page of the **Import and Export** wizard, but input pins are on the left side of the created symbol, output pins on the right, power pins on the top, and ground pins on the bottom.

### 4.5.1.1.3. Editing and Fracturing Symbol

After creating your new symbol in the Cadence Allegro PCB Librarian Part Developer tool, you can edit the symbol graphics, fracture the symbol into multiple slots, and add or change package or symbol properties.

The Part Developer Symbol Editor contains many graphical tools to edit the graphics of a particular symbol. To edit the symbol graphics, select the symbol in the cell hierarchy. The **Symbol Pins** tab appears. You can edit the preview graphic of the symbol in the **Symbol Pins** tab.
Fracturing a Cadence Allegro PCB Librarian Part Developer package into separate symbol slots is useful for FPGA designs. A single symbol for most FPGA packages might be too large for a single schematic page. Splitting the part into separate slots allows you to organize parts of the symbol by function, creating cleaner circuit schematics. For example, you can create one slot for an I/O symbol, a second slot for a JTAG symbol, and a third slot for a power/ground symbol.

Figure 22. Splitting a Symbol into Multiple Slots

To fracture a part into separate slots, or to modify the slot locations of pins on parts fractured in the Cadence Allegro PCB Librarian Part Developer tool, follow these steps:

1. Start the Cadence Allegro Design Project Manager.
3. Click Part Developer.
4. Click the name of the package you want to change in the cell hierarchy.
5. Click Functions/Slots. If you are not creating new slots but want to change the slot location of some pins, proceed to Step 6. If you are creating new slots, click Add. A dialog box appears, allowing you to add extra symbol slots. Set the number of extra slots you want to add to the existing symbol, not the total number of desired slots for the part. Click OK.
6. Click Distribute Pins. Specify the slot location for each pin. Use the checkboxes in each column to move pins from one slot to another. Click OK.
7. After distributing the pins, click the Package Pin tab and click Generate Symbol(s).
8. Select whether to create a new symbol or modify an existing symbol in each slot. Click OK.

The newly generated or modified slot symbols appear as separate symbols in the cell hierarchy. Each of these symbols can be edited individually.
Caution: TheCadence Allegro PCB Librarian Part Developer tool allows you to remap pin assignments in the **Package Pin** tab of the main Cadence Allegro PCB Librarian Part Developer window. If signals remap to different pins in the Cadence Allegro PCB Librarian Part Developer tool, the changes reflect only in regenerated symbols for use in your schematics. You cannot transfer pin assignment changes to the Intel Quartus Prime software from the Cadence Allegro PCB Librarian Part Developer tool, which creates a potential mismatch of the schematic symbols and assignments in the FPGA design. If pin assignment changes are necessary, make the changes in the Intel Quartus Prime Pin Planner instead of the Cadence Allegro PCB Librarian Part Developer tool, and update the symbol as described in the following sections.

For more information about creating, editing, and organizing component symbols with the Cadence Allegro PCB Librarian Part Developer tool, refer to the Part Developer Help.

4.5.1.1.4. Updating FPGA Symbols

As the design process continues, you must make logic changes in the Intel Quartus Prime software, placing signals on different pins after recompiling the design, or use the Intel Quartus Prime Pin Planner to make changes manually. The board designer can request such changes to improve the board routing and layout. To ensure signals connect to the correct pins on the FPGA, you must carry forward these types of changes to the circuit schematic and board layout tools. Updating the `.pin` in the Intel Quartus Prime software facilitates this flow.

**Figure 23. Updating the FPGA Symbol in the Design Flow**
To update the symbol using the Cadence Allegro PCB Librarian Part Developer tool after updating the .pin, follow these steps:

1. On the File menu, click **Import and Export**. The Import and Export wizard appears.
2. In the list of actions to perform, select **Import ECO - FPGA**. Click **Next**. The Select Source dialog box appears.
3. Select the updated source of the FPGA assignment information. In the **Vendor** list, select **Altera**. In the **PnR Tool** list, select **quartusII**. In the **PR File** field, click browse to specify the updated .pin in your Intel Quartus Prime project directory. Click **Next**. The Select Destination window appears.
4. Select the source component and a destination cell for the updated symbol. To create a new component based on the updated pin assignment data, select **Generate Custom Component**. Selecting **Generate Custom Component** replaces the cell listed under the **Specify Library and Cell** name header with a new, nonfractured cell. You can preserve these edits by selecting **Use standard component and select the existing library and cell**. Select the destination library for the component and click **Next**. The Preview of Import Data dialog box appears.
5. Make any additional changes to your symbol. Click **Next**. A list of ECO messages appears summarizing the changes made to the cell. To accept the changes and update the cell, click **Finish**.
6. The main Cadence Allegro PCB Librarian Part Developer window appears. You can edit, fracture, and generate the updated symbols as usual from the main Cadence Allegro PCB Librarian Part Developer window.

**Note:** If the Cadence Allegro PCB Librarian Part Developer tool is not set up to point to your PCB Librarian Expert license file, an error message appears in red at the bottom of the message text window of the Part Developer when you select the **Import and Export** command. To point to your PCB Librarian Expert license, on the File menu, click **Change Product**, and select the correct product license.

**Related Information**
FPGA-to-PCB Design Flow on page 53

### 4.5.2. Instantiating the Symbol in the Cadence Allegro Design Entry HDL Software

To instantiate the symbol in your Cadence Allegro Design Entry HDL schematic after saving the new symbol in the Cadence Allegro PCB Librarian Part Developer tool, follow these steps:

1. In the Cadence Allegro Project Manager tool, switch to the board design flow.
2. On the Flows menu, click **Board Design**.
3. To start the Cadence Allegro Design Entry HDL software, click **Design Entry**.
4. To add the newly created symbol to your schematic, on the Component menu, click **Add**. The **Add Component** dialog box appears.
5. Select the new symbol library location, and select the name of the cell you created from the list of cells.
The symbol attaches to your cursor for placement in the schematic. To fracture the symbol into slots, right-click the symbol and choose Version to select one of the slots for placement in the schematic.

**Related Information**

www.cadence.com
Provides more information about the Cadence Allegro Design Entry HDL software, including licensing, support, usage, training, and product updates.

### 4.6. FPGA-to-Board Integration with Cadence Allegro Design Entry CIS Software

The Cadence Allegro Design Entry CIS software is a schematic capture tool (part of the Cadence 200 series design flow based on OrCAD Capture CIS). Use the Cadence Allegro Design Entry CIS software to create flat circuit schematics for all types of PCB design. You can also create hierarchical schematics to facilitate design reuse and team-based design using the Cadence Allegro Design Entry CIS software. With the Cadence Allegro Design Entry CIS software, the design flow from FPGA-to-board is unidirectional using only the `.pin` generated by the Intel Quartus Prime software. You can only make signal and pin assignment changes in the Intel Quartus Prime software. These changes reflect as updated symbols in a Cadence Allegro Design Entry CIS schematic project.

**Figure 24. Design Flow with the Cadence Allegro Design Entry CIS Software**

- **Start FPGA Design**
  - Create or Change Pin Assignments
  - Run I/O Assignment Analysis
  - Run Full Compilation
  - `.pin`

- **Start PCB Design (Allegro Design Entry CIS)**
  - Create or Open Project
  - Generate or Update Schematic
  - Edit or Fracture Symbol
  - Instantiate Symbol in Schematic
  - Forward to Board Development Tool
  - Board Layout Tool
    - Layout and Route FPGA
    - End

**Note:** Routing or pin assignment changes made in a board layout tool or a Cadence Allegro Design Entry CIS symbol cannot be back-annotated to the Intel Quartus Prime software.
4.6.1. Creating a Cadence Allegro Design Entry CIS Project

The Cadence Allegro Design Entry CIS software has built-in support for creating schematic symbols using pin assignment information imported from the Intel Quartus Prime software.

To create a new project in the Cadence Allegro Design Entry CIS software, follow these steps:

1. On the File menu, point to **New** and click **Project**. The New Project wizard starts.

   When you create a new project, you can select the PC Board wizard, the Programmable Logic wizard, or a blank schematic.

2. Select the PC Board wizard to create a project where you can select which part libraries to use, or select a blank schematic.

   The Programmable Logic wizard only builds an FPGA logic design in the Cadence Allegro Design Entry CIS software.

   Your new project is in the specified location and consists of the following files:
   - OrCAD Capture Project File (**.opj**)
   - Schematic Design File (**.dsn**)

4.6.2. Generating a Part

After you create a new project or open an existing project in the Cadence Allegro Design Entry CIS software, you can generate a new schematic symbol based on your Intel Quartus Prime FPGA design. You can also update an existing symbol. The Cadence Allegro Design Entry CIS software stores component symbols in OrCAD Library File (**.olb**). When you place a symbol in a library attached to a project, it is immediately available for instantiation in the project schematic.

You can add symbols to an existing library or you can create a new library specifically for the symbols generated from your FPGA designs. To create a new library, follow these steps:

1. On the File menu, point to **New** and click **Library** in the Cadence Allegro Design Entry CIS software to create a default library named **library1.olb**. This library appears in the **Library** folder in the Project Manager window of the Cadence Allegro Design Entry CIS software.

2. To specify a desired name and location for the library, right-click the new library and select **Save As**. Saving the new library creates the library file.
4.6.3. Generating Schematic Symbol

You can now create a new symbol to represent your FPGA design in your schematic.

To generate a schematic symbol, follow these steps:
1. Start the Cadence Allegro Design Entry CIS software.
2. On the Tools menu, click **Generate Part**. The **Generate Part** dialog box appears.
3. To specify the .pin from your Intel Quartus Prime design, in the **Netlist/source file type** field, click **Browse**.
4. In the **Netlist/source file type** list, select **Altera Pin File**
5. Type the new part name.
6. Specify the **Destination part library** for the symbol. Failing to select an existing library for the part creates a new library with a default name that matches the name of your Cadence Allegro Design Entry CIS project.
7. To create a new symbol for this design, select **Create new part**. If you updated your .pin in the Intel Quartus Prime software and want to transfer any assignment changes to an existing symbol, select **Update pins on existing part in library**.
8. Select any other desired options and set **Implementation type** to <none>. The symbol is for a primitive library part based only on the .pin and does not require special implementation. Click **OK**.
9. Review the Undo warning and click **Yes** to complete the symbol generation.

You can locate the generated symbol in the selected library or in a new library found in the **Outputs** folder of the design in the Project Manager window. Double-click the name of the new symbol to see its graphical representation and edit it manually using the tools available in the Cadence Allegro Design Entry CIS software.

**Note:** For more information about creating and editing symbols in the Cadence Allegro Design Entry CIS software, refer to the Help in the software.

4.6.4. Splitting a Part

After saving a new symbol in a project library, you can fracture the symbol into multiple parts called sections. Fracturing a part into separate sections is useful for FPGA designs. A single symbol for most FPGA packages might be too large for a single schematic page. Splitting the part into separate sections allows you to organize parts of the symbol by function, creating cleaner circuit schematics. For example, you can create one slot for an I/O symbol, a second slot for a JTAG symbol, and a third slot for a power/ground symbol.
Figure 25. Splitting a Symbol into Multiple Sections

- This diagram represents a Cyclone device with JTAG or passive serial (PS) mode configuration option settings. Symbols created for other devices or other configuration modes might have different sets of configuration pins, but can be fractured in a similar manner.
- The power/ground section shows only a representation of power and ground pins because the device contains a high number of power and ground pins.

Note: Although symbol generation in the Design Entry CIS software refers to symbol fractures as sections, other tools use different names to refer to symbol fractures.

To split a part into sections, select the part in its library in the Project Manager window of the Cadence Allegro Design Entry CIS software. On the Tools menu, click Split Part or right-click the part and choose Split Part. The Split Part Section Input Spreadsheet appears.

Figure 26. Split Part Section Input Spreadsheet
Each row in the spreadsheet represents a pin in the symbol. The **Section** column indicates the section of the symbol to which each pin is assigned. You can locate all pins in a new symbol in section 1. You can change the values in the **Section** column to assign pins to various sections of the symbol. You can also specify the side of a section on the location of the pin by changing the values in the **Location** column. When you are ready, click **Split**. A new symbol appears in the same library as the original with the name `<original part name>_Split1`.

View and edit each section individually. To view the new sections of the part, double-click the part. The Part Symbol Editor window appears and the first section of the part displays for editing. On the View menu, click **Package** to view thumbnails of all the part sections. To edit the section of the symbol, double-click the thumbnail.

For more information about splitting parts into sections and editing symbol sections in the Cadence Allegro Design Entry CIS software, refer to the Help in the software.

### 4.6.5. Instantiating a Symbol in a Design Entry CIS Schematic

After saving a new symbol in a library in your Cadence Allegro Design Entry CIS project, you can instantiate the new symbol on a page in your schematic. Open a schematic page in the Project Manager window of the Cadence Allegro Design Entry CIS software. To add the new symbol to your schematic on the schematic page, on the Place menu, click **Part**. The **Place Part** dialog box appears.

#### Figure 27. **Place Part Dialog Box**

Select the new symbol library location and the newly created part name. If you select a part that is split into sections, you can select the section to place from the **Part** menu. Click **OK**. The symbol attaches to your cursor for placement in the schematic. To place the symbol, click the schematic page.

For more information about using the Cadence Allegro Design Entry CIS software, refer to the Help in the software.

### 4.6.6. Intel Libraries for the Cadence Allegro Design Entry CIS Software

Intel provides downloadable `.olb` for many of its device packages. You can add these libraries to your Cadence Allegro Design Entry CIS project and update the symbols with the pin assignments contained in the `.pin` generated by the Intel Quartus Prime software. You can use the downloaded library symbols as a base for creating custom
schematic symbols with your pin assignments that you can edit or fracture. This method increases productivity by reducing the amount of time it takes to create and edit a new symbol.

4.6.6.1. Using the Intel-provided Libraries with your Cadence Allegro Design Entry CIS Project

To use the Intel-provided libraries with your Cadence Allegro Design Entry CIS project, follow these steps:

1. Download the library of your target device from the Download Center page found through the Support page on the Intel website.

2. Create a copy of the appropriate .olb to maintain the original symbols. Place the copy in a convenient location, such as your Cadence Allegro Design Entry CIS project directory.

3. In the Project Manager window of the Cadence Allegro Design Entry CIS software, click once on the Library folder to select it. On the Edit menu, click Project or right-click the Library folder and choose Add File to select the copy of the downloaded .olb and add it to your project. You can locate the new library in the list of part libraries for your project.


5. In the Netlist/source file field, click Browse to specify the .pin in your Intel Quartus Prime design.

6. From the Netlist/source file type list, select Altera Pin File.

7. For Part name, type the name of the target device the same as it appears in the downloaded library file. For example, if you are using a device from the CYCLONE06.OLB library, type the part name to match one of the devices in this library such as ep1c6f256. You can rename the symbol in the Project Manager window after updating the part.

8. Set the Destination part library to the copy of the downloaded library you added to the project.

9. Select Update pins on existing part in library. Click OK.

10. Click Yes.

The symbol is updated with your pin assignments. Double-click the symbol in the Project Manager window to view and edit the symbol. On the View menu, click Package if you want to view and edit other sections of the symbol. If the symbol in the downloaded library is fractured into sections, you can edit each section but you cannot further fracture the part. You can generate a new part without using the downloaded part library if you require additional sections.

For more information about creating, editing, and fracturing symbols in the Cadence Allegro Design Entry CIS software, refer to the Help in the software.
## 4.7. Cadence Board Design Tools Support Revision History

### Table 7. Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.11.04</td>
<td>18.1</td>
<td>• Added &quot;Intel&quot; to &quot;Intel Quartus Prime&quot; software reference in a figure.</td>
</tr>
<tr>
<td>2019.07.15</td>
<td>18.1</td>
<td>• Retitled Cadence chapter to &quot;Cadence Board Tools Support.&quot; • Restructured steps in &quot;Integrating Intel FPGA Designs.&quot; • Added document archive.</td>
</tr>
<tr>
<td>2018.09.24</td>
<td>18.1</td>
<td>• Document title renamed • Other minor edits</td>
</tr>
<tr>
<td>2018.05.07</td>
<td>18.0</td>
<td>• First release as part of the stand-alone <em>PCB Design Tools User Guide</em></td>
</tr>
<tr>
<td>2016.10.31</td>
<td>16.1</td>
<td>• Implemented Intel rebranding.</td>
</tr>
<tr>
<td>2015.11.02</td>
<td>15.1</td>
<td>• Changed instances of <em>Quartus II</em> to <em>Intel Quartus Prime.</em></td>
</tr>
<tr>
<td>June 2014</td>
<td>14.0</td>
<td>Converted to DITA format.</td>
</tr>
<tr>
<td>June 2012</td>
<td>12.0</td>
<td>Removed survey link.</td>
</tr>
<tr>
<td>November 2011</td>
<td>10.0</td>
<td>Template update.</td>
</tr>
<tr>
<td>December 2010</td>
<td>10.0</td>
<td>Template update.</td>
</tr>
<tr>
<td>July 2010</td>
<td>10.0</td>
<td>• General style editing. • Removed Referenced Document Section. • Added a link to Help in &quot;Performing Simultaneous Switching Noise (SSN) Analysis of Your FPGA&quot; on page 9–5.</td>
</tr>
<tr>
<td>November 2009</td>
<td>9.1</td>
<td>• Added &quot;Performing Simultaneous Switching Noise (SSN) Analysis of Your FPGA&quot; on page 9–5. • General style editing. • Edited Figure 9–4 on page 9–10 and Figure 9–8 on page 9–16.</td>
</tr>
<tr>
<td>March 2009</td>
<td>9.0</td>
<td>• Chapter 9 was previously Chapter 7 in the 8.1 software release. • No change to content.</td>
</tr>
<tr>
<td>November 2008</td>
<td>8.1</td>
<td>Changed to 8-1/2 x 11 page size.</td>
</tr>
<tr>
<td>May 2008</td>
<td>8.0</td>
<td>Updated references.</td>
</tr>
</tbody>
</table>

If an software version is not listed, the user guide for the previous software version applies.

<table>
<thead>
<tr>
<th>Intel Quartus Prime Software Version</th>
<th>User Guide</th>
</tr>
</thead>
</table>
A. Intel Quartus Prime Pro Edition User Guides

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

Related Information

• Intel Quartus Prime Pro Edition User Guide: Getting Started
  Introduces the basic features, files, and design flow of the Intel Quartus Prime Pro Edition software, including managing Intel Quartus Prime Pro Edition projects and IP, initial design planning considerations, and project migration from previous software versions.

  Describes creating and optimizing systems using Platform Designer, a system integration tool that simplifies integrating customized IP cores in your project. Platform Designer automatically generates interconnect logic to connect intellectual property (IP) functions and subsystems.

• Intel Quartus Prime Pro Edition User Guide: Design Recommendations
  Describes best design practices for designing FPGAs with the Intel Quartus Prime Pro Edition software. HDL coding styles and synchronous design practices can significantly impact design performance. Following recommended HDL coding styles ensures that Intel Quartus Prime Pro Edition synthesis optimally implements your design in hardware.

• Intel Quartus Prime Pro Edition User Guide: Design Compilation
  Describes set up, running, and optimization for all stages of the Intel Quartus Prime Pro Edition Compiler. The Compiler synthesizes, places, and routes your design before generating a device programming file.

• Intel Quartus Prime Pro Edition User Guide: Design Optimization
  Describes Intel Quartus Prime Pro Edition settings, tools, and techniques that you can use to achieve the highest design performance in Intel FPGAs. Techniques include optimizing the design netlist, addressing critical chains that limit retiming and timing closure, optimizing device resource usage, device floorplanning, and implementing engineering change orders (ECOs).

  Describes operation of the Intel Quartus Prime Pro Edition Programmer, which allows you to configure Intel FPGA devices, and program CPLD and configuration devices, via connection with an Intel FPGA download cable.

• Intel Quartus Prime Pro Edition User Guide: Block-Based Design
  Describes block-based design flows, also known as modular or hierarchical design flows. These advanced flows enable preservation of design blocks (or logic that comprises a hierarchical design instance) within a project, and reuse of design blocks in other projects.
• Intel Quartus Prime Pro Edition User Guide: Partial Reconfiguration
  Describes Partial Reconfiguration, an advanced design flow that allows you to reconfigure a portion of the FPGA dynamically, while the remaining FPGA design continues to function. Define multiple personas for a particular design region, without impacting operation in other areas.

• Intel Quartus Prime Pro Edition User Guide: Third-party Simulation
  Describes RTL- and gate-level design simulation support for third-party simulation tools by Aldec*, Cadence*, Mentor Graphics, and Synopsys that allow you to verify design behavior before device programming. Includes simulator support, simulation flows, and simulating Intel FPGA IP.

• Intel Quartus Prime Pro Edition User Guide: Third-party Synthesis
  Describes support for optional synthesis of your design in third-party synthesis tools by Mentor Graphics, and Synopsys. Includes design flow steps, generated file descriptions, and synthesis guidelines.

• Intel Quartus Prime Pro Edition User Guide: Third-party Logic Equivalence Checking Tools
  Describes support for optional logic equivalence checking (LEC) of your design in third-party LEC tools by OneSpin*.

• Intel Quartus Prime Pro Edition User Guide: Debug Tools
  Describes a portfolio of Intel Quartus Prime Pro Edition in-system design debugging tools for real-time verification of your design. These tools provide visibility by routing (or "tapping") signals in your design to debugging logic. These tools include System Console, Signal Tap logic analyzer, system debugging toolkits, In-System Memory Content Editor, and In-System Sources and Probes Editor.

  Explains basic static timing analysis principals and use of the Intel Quartus Prime Pro Edition Timing Analyzer, a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in your design using an industry-standard constraint, analysis, and reporting methodology.

  Describes the Intel Quartus Prime Pro Edition Power Analysis tools that allow accurate estimation of device power consumption. Estimate the power consumption of a device to develop power budgets and design power supplies, voltage regulators, heat sink, and cooling systems.

• Intel Quartus Prime Pro Edition User Guide: Design Constraints
  Describes timing and logic constraints that influence how the Compiler implements your design, such as pin assignments, device options, logic options, and timing constraints. Use the Interface Planner to prototype interface implementations, plan clocks, and quickly define a legal device floorplan. Use the Pin Planner to visualize, modify, and validate all I/O assignments in a graphical representation of the target device.

  Describes support for optional third-party PCB design tools by Mentor Graphics and Cadence*. Also includes information about signal integrity analysis and simulations with HSPICE and IBIS Models.

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