FPGA design has evolved from small programmable circuits to designs that compete with multimillion-gate ASICs. At the same time, the I/O counts on FPGAs and logic density requirements of designs have increased exponentially. The higher-speed interfaces in FPGAs, including high-speed serial interfaces and memory interfaces, require careful interface design on the PCB. Designers must address the timing and signal integrity requirements of these interfaces early in the design cycle. Simultaneous switching noise (SSN) often leads to the degradation of signal integrity by causing signal distortion, thereby reducing the noise margin of a system.

Today’s complex FPGA system design is incomplete without addressing the integrity of signals coming in and out of the FPGA. Altera recommends that you perform SSN analysis early in your FPGA design and prior to the layout of your PCB with complete SSN analysis of your FPGA in the Quartus® II software. This chapter describes the Quartus II SSN Analyzer tool and covers the following topics:

- “Definitions”
- “Understanding SSN” on page 5–2
- “SSN Estimation Tools” on page 5–5
- “SSN Analysis Overview” on page 5–5
- “Optimizing Your Design for SSN Analysis” on page 5–8
- “Performing SSN Analysis and Viewing Results” on page 5–15
- “Decreasing Processing Time for SSN Analysis” on page 5–17

**Definitions**

The terminology used in this chapter includes the following terms:

**Aggressor:** An output or bidirectional signal that contributes to the noise for a victim I/O pin

**PDN:** Power distribution network

**QH:** Quiet high signal level on a pin

**QHN:** Quiet high noise on a pin, measured in volts

**QL:** Quiet low signal level on a pin

**QLN:** Quiet low noise on a pin, measured in volts

**SI:** Signal integrity (a superset of SSN, covering all noise sources)

**SSN:** Simultaneous switching noise
SSO: Simultaneous switching output (which are either the output or bidirectional pins)

Victim: An input, output, or bidirectional pin that is analyzed during SSN analysis. During SSN analysis, each pin is analyzed as a victim. If a pin is an output or bidirectional pin, the same pin acts as an aggressor signal for other pins.

Understanding SSN

SSN is defined as a noise voltage induced onto a single victim I/O pin on a device due to the switching behavior of other aggressor I/O pins on the device. SSN can be divided into two types of noise: voltage noise and timing noise.

Figure 5–1 shows a system with three pins. Two of the pins (A and C) are switching, while one pin (B) is quiet. If the pins are driven in isolation, the voltage waveforms at the output of the buffers appear without noise interference, as shown by the solid curves at the left of the figure. However, when the pins are switched simultaneously, the noise generated by pins A and C switching is injected onto the other pins, manifesting itself as a voltage noise on pin B and timing noise on pins A and C, as shown by the dotted curves in the figure.

Voltage noise is measured as the worst-case change in voltage of a signal due to SSN. When a signal is QH, it is measured as the change in voltage toward 0 V. When a signal is QL, it is measured as the change in voltage toward $V_{CC}$.

In the Quartus II software, only voltage noise is analyzed. Voltage noise can be caused by SSOs under two worst-case conditions:

- The victim pin is high and the aggressor pins (SSOs) are switching from low to high
- The victim pin is low and the aggressor pins (SSOs) are switching from high to low
For outputs, the noise is computed at the far-end receiver for pin B (refer to Figure 5–2).

**Figure 5–2. Quiet High Output Noise Estimation**

![Figure 5–2. Quiet High Output Noise Estimation](image)

For inputs, the noise is computed at the FPGA bumps as shown in for pin D (refer to Figure 5–3).

**Figure 5–3. Quiet Low Input Noise Estimation**

![Figure 5–3. Quiet Low Input Noise Estimation](image)

SSN can occur in any system, but the induced noise does not always result in failures. Voltage functional errors are caused by SSN on quiet victim pins only when the voltage values on the quiet pins change by a large enough voltage that the logic listening to that signal reads a change in the logic value. For QH signals, a voltage functional error occurs when noise events cause the voltage to fall below $V_{IH}$. Similarly, for QL signals, a voltage functional error occurs when noise events cause the voltage to rise above $V_{IL}$ (refer to Figure 5–4). Because $V_{IH}$ and $V_{IL}$ are different for different I/O standards, and because signals have different quiet voltage values, the absolute amount of SSN, measured in volts, cannot be used to determine if a voltage
failure occurs. Instead, to quantify whether an SSN event will cause a voltage error, the Quartus II software uses the amount of noise as a percent of signal margin when reporting noise margins in SSN analysis (refer to \textbf{Figure 5–4}).

\textbf{Figure 5–4. Reporting Noise Margins}

\begin{figure}[h!]
\centering
\includegraphics[width=0.8\textwidth]{noise_margins.png}
\caption{Reporting Noise Margins}
\end{figure}

\textbf{Figure 5–4} shows four noise events, two on QH signals and two on QL signals. The two noise events on the right-side of the figure consume 50 percent of the signal margin and do not cause voltage functional errors. However, the two noise events on the left side of the figure consume 100 percent of the signal margin and can cause a voltage functional error.

\textbf{Figure 5–5} illustrates a synchronous voltage noise event that does not result in a voltage functional error. Noise or glitches caused by aggressor signals are synchronously related to the victim pin outside of the sampling window of a receiver. The noise or glitches affect the switching time of a victim pin, but are not considered an input threshold violation failure.

\textbf{Figure 5–5. Synchronous Voltage Noise}

\begin{figure}[h!]
\centering
\includegraphics[width=0.8\textwidth]{synchronous_noise.png}
\caption{Synchronous Voltage Noise}
\end{figure}

For more information about the design factors that affect the noise margins during SSN analysis in the Quartus II software, refer to “\textit{SSN Analysis Overview}”.
SSN Estimation Tools

Addressing SSN early in your FPGA design and PCB layout can help you avoid costly board respins and lost time, both of which can impact your time-to-market. Altera provides many tools for SSN analysis and estimation, including the following tools:

- SSN characterization reports
- An early SSN estimation (ESE) tool
- The SSN Analyzer in the Quartus II software

For more information about the SSN characterization reports and the ESE tool, including device support information, refer to the Signal Integrity Center page of the Altera website.

For more information about the devices for which you can run the SSN Analyzer, refer to About the SSN Analyzer in Quartus II Help.

The ESE tool is useful for preliminary SSN analysis of your FPGA design; for more accurate results, however, you must use the SSN Analyzer in the Quartus II software. Table 5–1 compares some of the differences between the ESE tool and the SSN Analyzer.

<table>
<thead>
<tr>
<th>ESE Tool</th>
<th>SSN Analyzer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is not integrated with the Quartus II software.</td>
<td>Integrated with the Quartus II software, allowing you to perform preliminary SSN analysis while making I/O assignment changes in the Quartus II software.</td>
</tr>
<tr>
<td>QL and QH levels are computed assuming a worst-case pattern of I/O placements.</td>
<td>QL and QH levels are computed based on the I/O placements in your design.</td>
</tr>
<tr>
<td>No support for entering board information.</td>
<td>Supports board trace models and board layer information, resulting in a more accurate SSN analysis.</td>
</tr>
<tr>
<td>No graphical representation.</td>
<td>Integrated with the Quartus II Pin Planner, in which an SSN map shows the QL and QH levels on victim pins.</td>
</tr>
<tr>
<td>Good for doing an early SSN estimate. Does not require you to use the Quartus II software.</td>
<td>Requires you to create a Quartus II software project and provide the top-level port information.</td>
</tr>
</tbody>
</table>

SSN Analysis Overview

You can run the SSN Analyzer at different stages in your design cycle to obtain SSN results. The accuracy of the results depends on the completeness of your design information. Altera recommends that you start SSN analysis early in the design cycle to obtain preliminary results and make adjustments to your I/O assignments, and iterate through the design cycle to finally perform a fully constrained SSN analysis with complete information about your board.

Figure 5–6 shows the flows for both early pin-out and final pin-out SSN analysis. The early pin-out flow assumes conservative design rules initially, and then lets you analyze the design and iteratively apply tighter design rules until SSN analysis indicates your design meets SSN constraints. You must define pass criteria for SSN analysis as a percentage of signal margin in both the early pin-out flow and the final
pin-out flow. The pass criteria you define is specific to your design requirements. For example, a pass criterion you might define is a condition that verifies you have sufficient SSN margins in your design. You may require that the acceptable voltage noise on a pin must be below 70% of the voltage level for that pin. The pass criteria for the early-pin out flow may be higher than the final pin-out flow criteria, so that you do not spend too much time optimizing the on-FPGA portions of your design when the SSN metrics for the design may improve after the design is fully specified.

**Figure 5–6. Pin-Out Analysis (1)**

### Performing Early Pin-Out SSN Analysis

In the early stages of your design cycle, before you create pin location for your design, use the early pin-out flow (refer to Figure 5–6) to obtain preliminary SSN analysis results. In order to obtain useful SSN results, you must define the top-level ports of your design, but your design files do not have to be complete.

#### Performing Early Pin-Out SSN Analysis with the ESE Tool

If you know the I/O standards and signaling standards for your design, you can use the ESE tool to perform an initial SSN evaluation.

---

For more information about the ESE tool, refer to the Signal Integrity Center page of the Altera website.
Performing Early Pin-Out SSN Analysis with the SSN Analyzer

If you have complete information for the top-level ports of your design, you can use the SSN Analyzer to perform an initial SSN evaluation. Use the following steps to perform early pin-out SSN analysis:

1. Create a project in the Quartus II software.
2. Specify your top-level design information either in schematic form or in HDL code.
4. Create I/O assignments, such as I/O standard assignments, for the top-level ports in your design.
   - Do not create pin location assignments. The Fitter automatically creates optimized pin location assignments.
5. If you do not have completed design files and timing constraints, run I/O assignment analysis.
   - During I/O assignment analysis, the Fitter places all the unplaced pins on the device, and checks all the I/O placement rules.
6. Run the SSN Analyzer.

For more information about creating and managing projects, refer to the Managing Quartus II Projects chapter in volume 2 of the Quartus II Handbook. For more about generating a top-level design file in the Quartus II software and I/O assignment analysis, refer to the I/O Management chapter in volume 2 of the Quartus II Handbook.

In the early stages of your design cycle, you may not have complete board information, such as board trace parameters, layer information, and the signal breakout layers. If you run the SSN Analyzer without this specific information, it uses default board trace models and board layer information for SSN analysis, and as a result the SSN Analyzer confidence level is low. If the noise amounts are larger than the pass criteria for early pin-out SSN analysis, verify whether the SSN noise violations are true failures or false failures. For example, sometimes the SSN Analyzer can determine whether pins are switching synchronously and use that information to filter false positives; however, it may not be able to determine all the synchronous groups. You can improve the SSN analysis results by adjusting your I/O assignments and other design settings. After you optimize your design such that it meets the pass criteria for the early pin-out flow, you can then begin to design your PCB.

For more information, refer to “Optimizing Your Design for SSN Analysis”.
Performing Final Pin-Out SSN Analysis

You perform final pin-out SSN analysis after you place all the pins in your design, or the Fitter places them for you, and you have complete information about the board trace models and PCB layers. Even if your design achieves sufficient SSN results during early pin-out SSN analysis, you should run SSN analysis with the complete PCB information to ensure that SSN does not cause failures in your final design. You must specify the board parameters in the Quartus II software, including the PCB layer thicknesses, the signal breakout layers, and the board trace models, before you can run SSN analysis on your final assignments.

For more information, refer to “Optimizing Your Design for SSN Analysis”.

If the SSN analysis results meet the pass criteria for final pin-out SSN analysis, SSN analysis is complete. If the SSN analysis results do not meet the pass criteria, you must further optimize your design by changing the board and design parameters and then rerun the SSN Analyzer. If the design still does not meet the pass criteria, reduce the pass criteria for early pin-out SSN analysis, and restart the process. By reducing the pass criteria for early pin-out SSN analysis, you place a greater emphasis on reducing SSN through I/O settings and I/O placement. Changing the drive strength and slew rate of output and bidirectional pins, as well as adjusting the placement of different SSOs, can affect SSN results. Adjusting I/O settings and placement allows the design to meet the pass criteria for final pin-out SSN analysis after you specify the actual PCB board parameters.

Design Factors Affecting SSN Results

There are many factors that affect the SSN levels in your design. The two main factors are the drive strength and slew rate settings of the output and bidirectional pins in your design.

For more information about the factors that contribute to SSN voltage noise in your FPGA design and managing SSN in your system, refer to AN 472: Stratix II GX SSN Design Guidelines, AN 508: Cyclone III Simultaneous Switching Noise (SSN) Design Guidelines, and the Signal Integrity Center page of the Altera website.

Optimizing Your Design for SSN Analysis

The SSN Analyzer gives you flexibility to precisely define your system to obtain accurate SSN results. The SSN Analyzer produces a voltage noise estimate for each input, output, and bidirectional pin in the design. It allows you to estimate the SSN levels, comprised of QLN and QHN levels, for your FPGA pins. Performing SSN analysis helps you optimize your design for SSN during compilation.

Because the SSN Analyzer is integrated into the Quartus II software, it can automatically set up a system topology that matches your design. The SSN Analyzer accounts for different I/O standards and slew rate settings for each buffer in the design and models different board traces for each signal. Also, it correctly models the state of the unused pins in the design. The SSN Analyzer leverages any custom board trace assignments you set up for use by the advanced I/O timing feature.
The SSN Analyzer also models the package and vias in the design. Models for the different packages that Altera devices support are integrated into the Quartus II software. In the Quartus II software, you can specify different layers on which signals break out, each with its own thickness, and then specify which signal breaks out on which layer.

Figure 5–7 shows the circuit topology the SSN Analyzer automatically constructs. After constructing the circuit topology, the SSN Analyzer uses a simulation-based methodology to determine the SSN for each victim pin in the design.

**Figure 5–7. Circuit Topology for SSN Analysis**

---

**Optimizing Pin Placements for Signal Integrity**

You can take advantage of a built-in SSN optimization feature in the Quartus II software with the **SSN Optimization** logic option.

The I/O placements in your design may be affected when you use this option. Setting this option to **Normal compilation** does not affect the $f_{\text{MAX}}$ of your design during compilation, however setting this option to **Extra effort** level may impact your design $f_{\text{MAX}}$.

In order to use the **SSN Optimization** logic option, Altera recommends that you do not create location assignments for your pins; instead, let the Fitter place the pins during compilation so that it places the pins to meet the timing performance of your design. To display the Fitter-placed pins use the Show Fitter Placements feature in the Pin Planner. To accept these suggested pin locations, you must back-annotate your pin assignments.
Figure 5–8 shows the results of turning on the **SSN Optimization** logic option for a design. The image on the left shows the placement of the pins without the SSN Optimization logic option, and the image on the right shows the adjustments the Fitter made to pin placements to reduce the amount of SSN in the design when the SSN Optimization logic option is turned on.

**Figure 5–8. SSN Analysis Results Before and After Using the SSN Optimization Logic Option**

For more information about creating project-wide logic option assignments, refer to *Setting Up and Running the Fitter* in Quartus II Help. For more information about the Show Fitter Placements feature, refer to *Show Commands* in Quartus II Help. For more information about back-annotating assignments, refer to *Back-Annotating Assignments for A Project* in Quartus II Help.

**Specifying Board Trace Model Settings**

The SSN Analyzer uses circuit models to determine voltage noise during SSN analysis. The circuit topology (refer to Figure 5–7) is incomplete without board trace information and PCB layer information. You must describe the board trace and PCB layer parameters in your design to accurately compute the SSN in your FPGA device. However, if you do not specify some or all of the board trace parameters and PCB layer information, the SSN Analyzer uses default parameters during SSN analysis. When you use the default parameters, the SSN confidence level is low.

For more information about the default parameters used by the SSN Analyzer and SSN confidence levels, refer to “Confidence Metric Details Report” on page 5–16.

The board trace models required for the SSN Analyzer include the board trace termination resistors, pin loads (capacitance), and transmission line parameters. You can define the board circuit models, which are also known as board trace models, in the Quartus II software. The board trace model settings are shared with the models used during advanced I/O timing.

For more information about defining board trace models and advanced I/O timing, refer to the *I/O Management* chapter in volume 2 of the *Quartus II Handbook*. 
You can define an overall board trace model for each I/O standard in your design; this overall board trace model is the default model for all pins that use a particular I/O standard. After configuring the overall board trace model, you can customize the model for specific pins. The parameters you specify for the board trace model are also used in during advanced I/O timing analysis with the TimeQuest Timing Analyzer. If you already specified the board trace models as part of your advanced I/O timing assignments, the same parameters are used during SSN analysis.

For more information about defining a board trace model for your entire design, refer to Using Advanced I/O Timing in Quartus II Help. For more information about configuring component values for a board trace model, including a complete list of the supported unit prefixes and setting the values with Tcl scripts, refer to Board Trace Model in Quartus II Help.

All the assignments for board trace models you specify are saved to the .qsf. You can also use Tcl commands to create board trace model assignments. Example 5–1 shows Tcl commands for specifying transmission line parameters.

**Example 5–1. Specifying Board Trace Models**

```
set_instance_assignment -name BOARD_MODEL_TLINE_L_PER_LENGTH "3.041E-7" -to e[0]
set_instance_assignment -name BOARD_MODEL_TLINE_LENGTH 0.1391 -to e[0]
set_instance_assignment -name BOARD_MODEL_TLINE_C_PER_LENGTH "1.463E-10" -to e[0]
```

The best way to calculate transmission line parameters is to use a two-dimensional solver to estimate the inductance per inch and capacitance per inch for the transmission line. The termination resistor topology information can be obtained from the PCB schematics. The near-end and far-end pin load (capacitance) values can be obtained from the PCB schematic and other device data sheets. For example, if you know that an FPGA pin is driving a DIMM, you can obtain the far-end loading information in the data sheet for your target device.

For more information, refer to the Device Family Data Sheet in the appropriate device handbook available on the Literature and Technical Documentation page of the Altera website.

**Defining PCB Layers and PCB Layer Thickness**

Every PCB is fabricated using a number of layers. To remove some of the pessimism from your SSN results, Altera recommends that you create assignments describing your PCB layers in the Quartus II software. You can specify the number of layers on your PCB, and their thickness. The PCB layer information is used only during SSN analysis and is not used in other processes run by the Quartus II software. If a custom PCB breakout region is not described you can select the default thickness, which directs the SSN Analyzer to use a single-layer PCB breakout region during SSN analysis.

For more information about specifying PCB layer information, refer to Running the SSN Analyzer in Quartus II Help.
All the assignments you create for the PCB layers are saved to the .qsf. You can also use Tcl commands to create PCB layer assignments. You can create any number of PCB layers, however, the layers must be consecutive. Example 5–2 shows Tcl commands for specifying PCB layer assignments.

Example 5–2. Specifying PCB Layer Assignments

```tcl
set_global_assignment -name PCB_LAYER_THICKNESS 0.00099822M -section_id 1
set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 2
set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 3
```

Figure 5–9 shows the layout cross-section of a PCB in the Cadence Allegro PCB tool. The cross-section shows the stackup information of a PCB, which tells you the number of layers used in your PCB. The PCB shown in this example consists of various signal and circuit layers on which FPGA pins are routed, as well as the power and ground layers.

Figure 5–9. Snapshot of Stackup of a PCB Shown in the Allegro Board Design Environment

<table>
<thead>
<tr>
<th>Sublayer Name</th>
<th>Type</th>
<th>Material</th>
<th>Thickness (MIL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SURFACE</td>
<td>ALY</td>
<td>1.8</td>
</tr>
<tr>
<td>2</td>
<td>TOP</td>
<td>CONDUCTOR</td>
<td>PLATED COPPER FOIL</td>
</tr>
<tr>
<td>3</td>
<td>L2_PWR</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>4</td>
<td>L3_SIG</td>
<td>PLANE</td>
<td>COOPER</td>
</tr>
<tr>
<td>5</td>
<td>L4_SIG</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>6</td>
<td>L5_SIG</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>7</td>
<td>L6_SIG</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>8</td>
<td>L7_PWR</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>9</td>
<td>L8_SIG</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>10</td>
<td>L9_SIG</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>11</td>
<td>L10_GND</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>12</td>
<td>L11_GND</td>
<td>PLANE</td>
<td>COOPER</td>
</tr>
<tr>
<td>13</td>
<td>L12_GND</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>14</td>
<td>L13_GND</td>
<td>PLANE</td>
<td>COOPER</td>
</tr>
<tr>
<td>15</td>
<td>L14_GND</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
<tr>
<td>16</td>
<td>L15_GND</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>17</td>
<td>L16_GND</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>18</td>
<td>L17_GND</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>19</td>
<td>L18_GND</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>20</td>
<td>L19_GND</td>
<td>CONDUCTOR</td>
<td>FR-4</td>
</tr>
<tr>
<td>21</td>
<td>L20_GND</td>
<td>DIELECTRIC</td>
<td>FR-4</td>
</tr>
</tbody>
</table>

In this example, each of the four signal layers are a different thickness, with the depths shown in the Thickness (MIL) column. The layer thickness for each signal layer is computed as follows:

- Signal Layer 1 is the L4-SIGNAL, at thickness (1.9+3.6+1.2+3+1.2+4=) 14.9 mils
- Signal Layer 2 is the L5-SIGNAL, at thickness (0.6+6=) 6.6 mils
- Signal Layer 3 is the L8-SIGNAL, at thickness (0.6+4+1.2+3+1.2+4=) 14 mils
- Signal Layer 4 is the L9-SIGNAL, at thickness (0.6+6=) 6.6 mils
Figure 5–10 shows the results in the Quartus II software after you enter these PCB signal layers and thickness assignments.

Figure 5–10. PCB Layers Specified in the Quartus II Software

### Specifying Signal Breakout Layers

Each user I/O pin in your FPGA device can break out at different layers on your PCB. In the Pin Planner, you can specify on which layers the I/O pins in your design break out. The breakout layer information is used only during SSN analysis and is not used in other processes run by the Quartus II software. To assign a pin to PCB layer, follow these steps:

1. On the Assignments menu, click **Pin Planner**.
2. If necessary, perform Analysis & Elaboration, Analysis & Synthesis, or fully compile the design to populate the Pin Planner with the node names in the design.
3. Right-click anywhere in the **All Pins** or **Groups** list, and then click **Customize Columns**.
4. Select the **PCB layer** column and move it from the **Available columns** list to the **Show these columns in this order** list.
5. Click **OK**.
6. In the **PCB layer** column, specify the PCB layer to which you want to connect the signal.
7. On the File menu, click **Save Project** to save the changes.

When you create PCB breakout layer assignments in the Pin Planner, you can assign the pin to any layer, even if you did not yet define the PCB layer.
Creating I/O Assignments

I/O assignments are required in FPGA design and are also used during SSN analysis to estimate voltage noise. Each input, output, or bidirectional signal in your design is assigned a physical pin location on the device using pin location assignments. Each signal has a physical I/O buffer that has a specific I/O standard, pin location, drive strength, and slew rate. The SSN Analyzer supports most I/O standards in a device family, such as the LVTTL and LVCMOS I/O standards.

The SSN Analyzer does not support differential I/O standards, such as the LVDS I/O standard and its variations, because differential I/O standards contribute a small amount of SSN.

For more information about supported I/O standards, refer to the appropriate device handbook available on the Literature and Technical Documentation page of the Altera website.

For more information about creating and managing I/O assignments, refer to the I/O Management chapter in volume 2 of the Quartus II Handbook.

Decreasing Pessimism in SSN Analysis

In the absence of specific timing information, the SSN Analyzer analyzes your design under worst-case conditions. Worst-case conditions include all pins acting as aggressor signals on all possible victim pins and all aggressor pins switching with the worst possible timing relationship. The results of SSN analysis under worst-case conditions are very pessimistic. You can improve the results of SSN Analysis by creating group assignments for specific types of pins. Use the following group assignments to decrease the pessimism in SSN analysis results:

- Assign pins to an output enable group—All pins in an output enable group must be either all input pins or all output pins. If all the pins in a group are always either all inputs or all outputs, it is impossible for an output pin in the group to cause SSN noise on an input pin in the group. You can assign pins to an output enable group with the Output Enable Group logic option.

- Assign pins to a synchronous group—I/O pins that are part of a synchronous group (signals that switch at the same time) may cause SSN, but do not result in any failures because the noise glitch occurs during the switching period of the signal. The noise, therefore, does not occur in the sampling window of that signal. You can assign pins to an output enable group with the Synchronous Group logic option. For example, in your design you have a bus with 32 pins that all belong to the same group. In a real operation, the bus switches at the same time, so any voltage noise induced by a pin on its groupmates does not matter, because it does not fall in the sampling window. If you do not assign the bus to a synchronous group, the other 31 pins can act as aggressors for the first pin in that group, leading to higher QL and QH noise levels during SSN analysis.

In some cases, the SSN Analyzer can detect the grouping for bidirectional pins by looking at the output enable signal of the bidirectional pins. However, Altera recommends that you explicitly specify the bidirectional groups and output groups in your design.
For more information about creating logic option assignments, refer to Assigning Device I/O Pins in Quartus II Help.

Excluding Pins as Aggressor Signals

The SSN Analyzer uses the following conditions to exclude pins as aggressor signals for a specific victim pin:

- A pin that is a complement of the victim pin. For example, any pin that is assigned a differential I/O standard cannot be an aggressor pin.

- A programming pin or JTAG pin because these pins are not active in user mode.

- Pins that have the same output enable signal as a bidirectional victim pin that the SSN Analyzer analyses as an input pin. Pins with the same output enable signal also act as input pins and therefore cannot be aggressor pins at the same time. For information about grouping bidirectional pins, refer to “Performing SSN Analysis and Viewing Results”.

- Pins in the same synchronous group as a victim output pin. For information about grouping output pins, refer to “Performing SSN Analysis and Viewing Results”.

- A pin assigned the I/O Maximum Toggle Rate logic option with a frequency setting of zero. The SSN Analyzer does not consider pins with this setting as aggressor pins.

For more information about creating pin assignments with the Pin Planner, refer to Assigning Device I/O Pins in Quartus II Help.

Performing SSN Analysis and Viewing Results

You can perform SSN analysis either on your entire design, or you can limit the analysis to specific I/O banks.

If you know the problem area for SSN is within one I/O bank and you are changing pin assignments only in that bank, you can run SSN analysis for just that one I/O bank to reduce analysis time.

For more information, refer to Running the SSN Analyzer in Quartus II Help.

For more information about I/O bank numbering, refer to the appropriate device handbook available on the Literature and Technical Documentation page of the Altera website.

Understanding the SSN Reports

When SSN analysis is complete, you can view detailed analysis reports. The detailed messages in the reports help you understand and resolve SSN problems.

The SSN Analyzer section of the Compilation report contains information generated during SSN analysis, including the following reports:

- Summary
- Output Pins
- Input Pins
- Unanalyzed Pins
- Confidence Metric Details

**Summary Report**

The Summary report summarizes the SSN Analyzer status and rates the SSN Analyzer confidence level as low, medium, or high. The confidence level depends on the completeness of your board trace model assignments. The more assignments you complete, the higher the confidence level. However, the confidence level does not always contribute to the accuracy of the QL and QH noise levels on a victim pin. The accuracy of QH and QL noise levels depends the accuracy of your board trace model assignments.

**Output Pins and Input Pins Reports**

The Output Pins report lists all of the output pins and bidirectional pins that are treated as output pins during SSN analysis. The Input Pins report lists all of the input pins and bidirectional pins that are treated as inputs during SSN analysis. Both reports list the location assignments for the pins treated as SSN outputs or inputs during SSN analysis, the QL and QH noise in volts, and what percentage the QL and QH margins are for the I/O standard used for that signal. The QH and QL noise margins that fall in the critical range (> 90%) are shown in red. The QH and QL noise margins that fall in the range of 70% to 90% are shown in gray.

**Unanalyzed Pins Report**

Not all pins are analyzed for SSN analysis. The following pins are not analyzed and are reported in the Unanalyzed Pins report:

- Pins assigned the LVDS I/O standard or any LVDS variations, such as the mini-LVDS I/O standard
- Pins created in the migration flow that cover power and supply pins in other packages
- The negative terminals of pseudo-differential I/O standards; the noise on differential standards is reported as the differential noise and is reported on the positive terminal

**Confidence Metric Details Report**

The Confidence Metric Details Report lists the values used during SSN Analysis for unspecified I/O, board, and PCB assignments.

**Viewing SSN Analysis Results in the Pin Planner**

After SSN analysis completes, you can analyze the results in the Pin Planner. In the Pin Planner you can identify the SSN hotspots in your device, as well as the QL and QH noise levels. The QL and QH results for each pin are displayed with a different color that represents whether the pin is below the warning threshold, below the critical threshold, or above the critical threshold. This color representation is also referred to as the SSN map of your FPGA device.
When you view the SSN map, you can customize which details to display, including input pins, output pins, QH signals, QL signals, and noise levels. You can also adjust the threshold levels for QH and QL noise voltages. Adjusting the threshold levels in the Pin Planner does not change the threshold levels reported during SSN analysis and does not change the data in any of the SSN reports.

You can also change I/O assignments and board trace information and rerun the SSN Analyzer to view the SSN analysis results based on those modified settings.

For more information, refer to Show SSN Analyzer Results and Running the SSN Analyzer in Quartus II Help.

Decreasing Processing Time for SSN Analysis

FPGA designs are getting larger in density, logic, and I/O count. The time it takes to complete SSN analysis and other Quartus II software processes affects your development time. Faster processing times can reduce your design cycle time. Use the following guidelines to reduce processing time:

- Direct the Quartus II software to use more than one processor for parallel executables, including the SSN Analyzer
- Perform SSN analysis after I/O assignment analysis if your design files and constraints are complete, and you are interested in generating the SSN results early in the design process and want to adjust I/O placements to see if you can obtain better results
- Perform SSN analysis after fitting if you want to view preliminary SSN results that do not take into account complete I/O assignment and I/O timing results
- Perform engineering change orders (ECOs) on your design, rather than recompiling the entire design, if you want to rerun SSN analysis after changing I/O assignments

For more information about using parallel processors, refer to Setting Up and Running Analysis and Synthesis and Compilation Process Settings Page in Quartus II Help. For more information about performing I/O assignment analysis, refer to Assigning Device I/O Pins in Quartus II Help. For more information about running the Fitter, refer to Setting Up and Running the Fitter in Quartus II Help.

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

Scripting Support

A Tcl script allows you to run procedures and determine settings described in this chapter. You can also run some of these procedures at a command prompt. The Quartus II software provides several packages to compile your design and create I/O assignments for analysis and fitting. You can create a custom Tcl script that maps the design and runs SSN analysis on your design.
For detailed information about specific scripting command options and Tcl API packages, type the following command at a system command prompt to run the Quartus II Command-Line and Tcl API Help browser:

```
quartus_sh --qhelp
```

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

### Optimizing Pin Placements for Signal Integrity

You can create an assignment that directs the Fitter to optimize pin placements for signal integrity with a Tcl command.

The following Tcl command directs the Fitter to optimize pin placement for signal integrity without affecting design f\textsubscript{MAX}:

```
set_global_assignment -name OPTIMIZE_SIGNAL_INTEGRITY "Normal Compilation"
```

For more information, refer to “Optimizing Pin Placements for Signal Integrity” on page 5–9.

### Defining PCB Layers and PCB Layer Thickness

You can create PCB layer and thickness assignments with a Tcl command. shows Tcl commands for specifying PCB layer assignments.

#### Example 5–3. Specifying PCB Layer Assignments

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00099822M -section_id 1</td>
<td>PCB Layer 1 Thickness: 0.00099822M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 2</td>
<td>PCB Layer 2 Thickness: 0.00034036M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 3</td>
<td>PCB Layer 3 Thickness: 0.00034036M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00055372M -section_id 4</td>
<td>PCB Layer 4 Thickness: 0.00055372M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 5</td>
<td>PCB Layer 5 Thickness: 0.00034036M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00034036M -section_id 6</td>
<td>PCB Layer 6 Thickness: 0.00034036M</td>
</tr>
<tr>
<td>set_global_assignment -name PCB_LAYER_THICKNESS 0.00082042M -section_id 7</td>
<td>PCB Layer 7 Thickness: 0.00082042M</td>
</tr>
</tbody>
</table>

These Tcl commands specify that there are seven PCB layers in the design, each with a different thickness. In each assignment, the letter \( \text{M} \) indicates the unit of measurement is millimeters. When you specify PCB layer assignments with Tcl commands, you must list the layers in consecutive order. For example, you would receive an error during SSN Analysis if your Tcl commands created the following assignments:

```
set_global_assignment -name PCB_LAYER_THICKNESS 0.00099822M -section_id 1
set_global_assignment -name PCB_LAYER_THICKNESS 0.00082042M -section_id 7
```

To create assignments with the unit of measurement in mils, refer to the syntax in the following Tcl commands. These Tcl commands specify the same settings as shown in Figure 5–10 on page 5–13.

```
set_global_assignment -name PCB_LAYER_THICKNESS 14.9MIL -section_id 1
set_global_assignment -name PCB_LAYER_THICKNESS 6.6MIL -section_id 2
set_global_assignment -name PCB_LAYER_THICKNESS 14MIL -section_id 3
set_global_assignment -name PCB_LAYER_THICKNESS 6.6MIL -section_id 4
```
For more information, refer to “Defining PCB Layers and PCB Layer Thickness” on page 5–11.

**Specifying Signal Breakout Layers**

You can create signal breakout layer assignments with a Tcl command. Example 5–4 shows Tcl commands for specifying signal breakout layer assignments:

<table>
<thead>
<tr>
<th>Example 5–4. Specifying Signal Breakout Layer Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>set_instance_assignment -name PCB_LAYER 10 -to e[2]</td>
</tr>
<tr>
<td>set_instance_assignment -name PCB_LAYER 3 -to e[3]</td>
</tr>
</tbody>
</table>

When you create PCB breakout layer assignments with Tcl commands, if you do not specify a PCB layer, or if you specify a PCB layer that does not exist, the SSN Analyzer breaks out the signal at the bottommost PCB layer.

If you create a PCB layer breakout assignment to a layer that does not exist, the SSN Analyzer will generate a warning message.

For more information, refer to “Specifying Signal Breakout Layers” on page 5–13.

**Decreasing Pessimism in SSN Analysis**

You can create output enable group and synchronous group assignments to help decrease pessimism during SSN Analysis with a Tcl command.

The following Tcl command assigns the bidirectional bus `DATAINOUT` to an output enable group:

```tcl
set_instance_assignment -name OUTPUT_ENABLE_GROUP 1 -to DATAINOUT
```

The following Tcl command assigns the bus `PCI_ADD_io` to a synchronous group:

```tcl
set_instance_assignment -name SYNCHRONOUS_GROUP 1 -to PCI_AD_io
```

For more information, refer to “Decreasing Pessimism in SSN Analysis” on page 5–14.

**Performing SSN Analysis**

You can perform SSN analysis with a command-line command. Use the `quartus_si` package that is provided with the Quartus II software.

Type the following command at a system command prompt to start the SSN Analyzer:

```
quartus_si <project name> r
```

To analyze just one I/O bank, type the following command at a system command prompt:

```
quartus_si <project revision> --bank = bank id r
```

For example, to run analyze the I/O bank 2A type the following command:

```
quartus_si counter --bank=2A r
```
For more information, refer to “Performing SSN Analysis and Viewing Results” on page 5–15.

For more information about the `quartus_si` package, type `quartus_si -h` at a system command prompt.

**Conclusion**

To assist you with SSN Analysis, you can use the fast and accurate SSN Analyzer to help you estimate the SSN performance of your FPGA both early in the design cycle and when your PCB is complete. The SSN methodology discussed in this chapter gives you the tools you need to ensure your FPGA design meets your SSN requirements.

**Document Revision History**

Table 5–2 shows the revision history for this chapter.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2012</td>
<td>12.0.0</td>
<td>Removed survey link.</td>
</tr>
<tr>
<td>November 2011</td>
<td>10.0.2</td>
<td>Template update</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>Reorganized and edited the chapter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added links to Quartus II Help for procedural information previously included in the chapter</td>
</tr>
<tr>
<td>November 2009</td>
<td>9.1.0</td>
<td>Added “Figure 6–9 shows the layout cross-section of a PCB in the Cadence Allegro PCB tool. The cross-section shows the stackup information of a PCB, which tells you the number of layers used in your PCB. The PCB shown in this example consists of various signal and circuit layers on which FPGA pins are routed, as well as the power and ground layers.” on page 6–12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated for the Quartus II software 9.1 release</td>
</tr>
<tr>
<td>March 2009</td>
<td>9.0.0</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

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