You can use the Quartus® II software to analyze the average mean time between failures (MTBF) due to metastability caused by synchronization of asynchronous signals, and optimize the design to improve the metastability MTBF.

All registers in digital devices, such as FPGAs, have defined signal-timing requirements that allow each register to correctly capture data at its input ports and produce an output signal. To ensure reliable operation, the input to a register must be stable for a minimum amount of time before the clock edge (register setup time or $t_{SU}$) and a minimum amount of time after the clock edge (register hold time or $t_{H}$). The register output is available after a specified clock-to-output delay ($t_{CO}$).

If the data violates the setup or hold time requirements, the output of the register might go into a metastable state. In a metastable state, the voltage at the register output hovers at a value between the high and low states, which means the output transition to a defined high or low state is delayed beyond the specified $t_{CO}$. Different destination registers might capture different values for the metastable signal, which can cause the system to fail.

In synchronous systems, the input signals must always meet the register timing requirements, so that metastability does not occur. Metastability problems commonly occur when a signal is transferred between circuitry in unrelated or asynchronous clock domains, because the signal can arrive at any time relative to the destination clock.

The MTBF due to metastability is an estimate of the average time between instances when metastability could cause a design failure. A high MTBF (such as hundreds or thousands of years between metastability failures) indicates a more robust design. You should determine an acceptable target MTBF in the context of your entire system and taking in account that MTBF calculations are statistical estimates.

The metastability MTBF for a specific signal transfer, or all the transfers in a design, can be calculated using information about the design and the device characteristics. Improving the metastability MTBF for your design reduces the chance that signal transfers could cause metastability problems in your device.

The Quartus II software provides analysis, optimization, and reporting features to help manage metastability in Altera designs. These metastability features are supported only for designs constrained with the Quartus II Timing Analyzer. Both typical and worst-case MTBF values are generated for select device families.

Related Information

- **Understanding Metastability in FPGAs**
  For more information about metastability due to signal synchronization, its effects in FPGAs, and how MTBF is calculated
Metastability Analysis in the Quartus II Software

When a signal transfers between circuitry in unrelated or asynchronous clock domains, the first register in the new clock domain acts as a synchronization register.

To minimize the failures due to metastability in asynchronous signal transfers, circuit designers typically use a sequence of registers (a synchronization register chain or synchronizer) in the destination clock domain to resynchronize the signal to the new clock domain and allow additional time for a potentially metastable signal to resolve to a known value. Designers commonly use two registers to synchronize a new signal, but a standard of three registers provides better metastability protection.

The timing analyzer can analyze and report the MTBF for each identified synchronizer that meets its timing requirements, and can generate an estimate of the overall design MTBF. The software uses this information to optimize the design MTBF, and you can use this information to determine whether your design requires longer synchronizer chains.

Related Information

- Metastability and MTBF Reporting on page 13-4
- MTBF Optimization on page 13-7

Synchronization Register Chains

A synchronization register chain, or synchronizer, is defined as a sequence of registers that meets the following requirements:

- The registers in the chain are all clocked by the same clock or phase-related clocks.
- The first register in the chain is driven asynchronously or from an unrelated clock domain.
- Each register fans out to only one register, except the last register in the chain.

The length of the synchronization register chain is the number of registers in the synchronizing clock domain that meet the above requirements. The figure shows a sample two-register synchronization chain.

Figure 13-1: Sample Synchronization Register Chain
The path between synchronization registers can contain combinational logic as long as all registers of the synchronization register chain are in the same clock domain. The figure shows an example of a synchronization register chain that includes logic between the registers.

**Figure 13-2: Sample Synchronization Register Chain Containing Logic**

The timing slack available in the register-to-register paths of the synchronizer allows a metastable signal to settle, and is referred to as the available settling time. The available settling time in the MTBF calculation for a synchronizer is the sum of the output timing slacks for each register in the chain. Adding available settling time with additional synchronization registers improves the metastability MTBF.

**Related Information**

*How Timing Constraints Affect Synchronizer Identification and Metastability Analysis* on page 13-3

### Identifying Synchronizers for Metastability Analysis

The first step in enabling metastability MTBF analysis and optimization in the Quartus II software is to identify which registers are part of a synchronization register chain. You can apply synchronizer identification settings globally to automatically list possible synchronizers with the *Synchronizer identification* option on the *Timing Analyzer* page in the *Settings* dialog box.

Synchronization chains are already identified within most Altera intellectual property (IP) cores.

**Related Information**

*Identifying Synchronizers for Metastability*

For more information about how to enable metastability MTBF analysis and optimization in the Quartus II software, and more detailed descriptions of the synchronizer identification settings

### How Timing Constraints Affect Synchronizer Identification and Metastability Analysis

The timing analyzer can analyze metastability MTBF only if a synchronization chain meets its timing requirements. The metastability failure rate depends on the timing slack available in the synchronizer’s register-to-register connections, because that slack is the available settling time for a potential metastable signal. Therefore, you must ensure that your design is correctly constrained with the real application frequency requirements to get an accurate MTBF report.

In addition, the *Auto* and *Forced If Asynchronous* synchronizer identification options use timing constraints to automatically detect the synchronizer chains in the design. These options check for signal transfers between
circuitry in unrelated or asynchronous clock domains, so clock domains must be related correctly with timing constraints.

The timing analyzer views input ports as asynchronous signals unless they are associated correctly with a clock domain. If an input port fans out to registers that are not acting as synchronization registers, apply a set_input_delay constraint to the input port; otherwise, the input register might be reported as a synchronization register. Constraining a synchronous input port with a set_max_delay constraint for a setup (tsU) requirement does not prevent synchronizer identification because the constraint does not associate the input port with a clock domain.

Instead, use the following command to specify an input setup requirement associated with a clock:

```
set_input_delay -max -clock <clock name> <latch – launch – tsu requirement> <input port name>
```

Registers that are at the end of false paths are also considered synchronization registers because false paths are not timing-analyzed. Because there are no timing requirements for these paths, the signal may change at any point, which may violate the tsU and th of the register. Therefore, these registers are identified as synchronization registers. If these registers are not used for synchronization, you can turn off synchronizer identification and analysis. To do so, set Synchronizer Identification to Off for the first synchronization register in these register chains.

**Metastability and MTBF Reporting**

The Quartus II software reports the metastability analysis results in the Compilation Report and Timing Analyzer reports.

The MTBF calculation uses timing and structural information about the design, silicon characteristics, and operating conditions, along with the data toggle rate.

If you change the Synchronizer Identification settings, you can generate new metastability reports by rerunning the timing analyzer. However, you should rerun the Fitter first so that the registers identified with the new setting can be optimized for metastability MTBF.

**Related Information**

- [Metastability Reports](#) on page 13-4
- [MTBF Optimization](#) on page 13-7
- [Synchronizer Data Toggle Rate in MTBF Calculation](#) on page 13-6
- [Understanding Metastability in FPGAs](#)

**Metastability Reports**

Metastability reports provide summaries of the metastability analysis results. In addition to the MTBF Summary and Synchronizer Summary reports, the Timing Analyzer tool reports additional statistics in a report for each synchronizer chain.

**Note:** If the design uses only the Auto Synchronizer Identification setting, the reports list likely synchronizers but do not report MTBF. To obtain an MTBF for each register chain, force identification of synchronization registers.
**Note:** If the synchronizer chain does not meet its timing requirements, the reports list identified synchronizers but do not report MTBF. To obtain MTBF calculations, ensure that the design is properly constrained and that the synchronizer meets its timing requirements.

**Related Information**
- Identifying Synchronizers for Metastability Analysis on page 13-3
- How Timing Constraints Affect Synchronizer Identification and Metastability Analysis on page 13-3
- Viewing Metastability Reports

**MTBF Summary Report**
The MTBF Summary reports an estimate of the overall robustness of cross-clock domain and asynchronous transfers in the design. This estimate uses the MTBF results of all synchronization chains in the design to calculate an MTBF for the entire design.

**Typical and Worst-Case MTBF of Design**
The MTBF Summary Report shows the **Typical MTBF of Design** and the **Worst-Case MTBF of Design** for supported fully-characterized devices. The typical MTBF result assumes typical conditions, defined as nominal silicon characteristics for the selected device speed grade, as well as nominal operating conditions. The worst case MTBF result uses the worst case silicon characteristics for the selected device speed grade.

When you analyze multiple timing corners in the timing analyzer, the MTBF calculation may vary because of changes in the operating conditions, and the timing slack or available metastability settling time. Altera recommends running multi-corner timing analysis to ensure that you analyze the worst MTBF results, because the worst timing corner for MTBF does not necessarily match the worst corner for timing performance.

**Related Information**
Timing Analyzer page

**Synchronizer Chains**
The MTBF Summary report also lists the **Number of Synchronizer Chains Found** and the length of the **Shortest Synchronizer Chain**, which can help you identify whether the report is based on accurate information.

If the number of synchronizer chains found is different from what you expect, or if the length of the shortest synchronizer chain is less than you expect, you might have to add or change **Synchronizer Identification** settings for the design. The report also provides the **Worst Case Available Settling Time**, defined as the available settling time for the synchronizer with the worst MTBF.

You can use the reported **Fraction of Chains for which MTBFs Could Not Be Calculated** to determine whether a high proportion of chains are missing in the metastability analysis. A fraction of 1, for example, means that MTBF could not be calculated for any chains in the design. MTBF is not calculated if you have not identified the chain with the appropriate **Synchronizer identification** option, or if paths are not timing-analyzed and therefore have no valid slack for metastability analysis. You might have to correct your timing constraints to enable complete analysis of the applicable register chains.
Increasing Available Settling Time

The MTBF Summary report specifies how an increase of 100ps in available settling time increases the MTBF values. If your MTBF is not satisfactory, this metric can help you determine how much extra slack would be required in your synchronizer chain to allow you to reach the desired design MTBF.

Synchronizer Summary Report

The Synchronizer Summary lists the synchronization register chains detected in the design depending on the Synchronizer Identification setting.

The Source Node is the register or input port that is the source of the asynchronous transfer. The Synchronization Node is the first register of the synchronization chain. The Source Clock is the clock domain of the source node, and the Synchronization Clock is the clock domain of the synchronizer chain.

This summary reports the calculated Worst-Case MTBF, if available, and the Typical MTBF, for each appropriately identified synchronization register chain that meets its timing requirement.

Related Information

Synchronizer Chain Statistics Report in the Timing Analyzer on page 13-6

Synchronizer Chain Statistics Report in the Timing Analyzer

The timing analyzer provides an additional report for each synchronizer chain.

The Chain Summary tab matches the Synchronizer Summary information described in Synchronizer Summary Report, while the Statistics tab adds more details, including whether the Method of Synchronizer Identification was User Specified (with the Forced if Asynchronous or Forced settings for the Synchronizer Identification setting), or Automatic (with the Auto setting). The Number of Synchronization Registers in Chain report provides information about the parameters that affect the MTBF calculation, including the Available Settling Time for the chain and the Data Toggle Rate Used in MTBF Calculation.

The following information is also included to help you locate the chain in your design:

- Source Clock and Asynchronous Source node of the signal.
- Synchronization Clock in the destination clock domain.
- Node names of the Synchronization Registers in the chain.

Related Information

Synchronizer Data Toggle Rate in MTBF Calculation on page 13-6

Synchronizer Data Toggle Rate in MTBF Calculation

The MTBF calculations assume the data being synchronized is switching at a toggle rate of 12.5% of the source clock frequency. That is, the arriving data is assumed to switch once every eight source clock cycles.

If multiple clocks apply, the highest frequency is used. If no source clocks can be determined, the data rate is taken as 12.5% of the synchronization clock frequency.

If you know an approximate rate at which the data changes, specify it with the Synchronizer Toggle Rate assignment in the Assignment Editor. You can also apply this assignment to an entity or the entire design. Set the data toggle rate, in number of transitions per second, on the first register of a synchronization chain. The timing analyzer takes the specified rate into account when computing the MTBF of that particular register chain. If a data signal never toggles and does not affect the reliability of the design, you can set the
**Synchronizer Toggle Rate** to 0 for the synchronization chain so the MTBF is not reported. To apply the assignment with Tcl, use the following command:

```tcl
set_instance_assignment -name SYNCHRONIZER_TOGGLE_RATE <toggle rate in transitions/second> -to <register name>
```

In addition to **Synchronizer Toggle Rate**, there are two other assignments associated with toggle rates, which are not used for metastability MTBF calculations. The I/O Maximum Toggle Rate is only used for pins, and specifies the worst-case toggle rates used for signal integrity purposes. The Power Toggle Rate assignment is used to specify the expected time-averaged toggle rate, and is used by the PowerPlay Power Analyzer to estimate time-averaged power consumption.

## MTBF Optimization

In addition to reporting synchronization register chains and MTBF values found in the design, the Quartus II software can also protect these registers from optimizations that might negatively impact MTBF and can optimize the register placement and routing if the MTBF is too low.

Synchronization register chains must first be explicitly identified as synchronizers. Altera recommends that you set **Synchronizer Identification** to **Forced If Asynchronous** for all registers that are part of a synchronizer chain.

Optimization algorithms, such as register duplication and logic retiming in physical synthesis, are not performed on identified synchronization registers. The Fitter protects the number of synchronization registers specified by the **Synchronizer Register Chain Length** option.

In addition, the Fitter optimizes identified synchronizers for improved MTBF by placing and routing the registers to increase their output setup slack values. Adding slack in the synchronizer chain increases the available settling time for a potentially metastable signal, which improves the chance that the signal resolves to a known value, and exponentially increases the design MTBF. The Fitter optimizes the number of synchronization registers specified by the **Synchronizer Register Chain Length** option.

Metastability optimization is **on** by default. To view or change the option, on the Assignments menu, click **Settings**. Under **Fitter Settings**, click **More Settings**. From the **More Settings** dialog box, you can turn on or off the **Optimize Design for Metastability** option. To turn the optimization on or off with Tcl, use the following command:

```tcl
set_global_assignment -name OPTIMIZE_FOR_METASTABILITY <ON|OFF>
```

**Related Information**

- **Identifying Synchronizers for Metastability Analysis** on page 13-3

**Synchronization Register Chain Length**

The **Synchronization Register Chain Length** option specifies how many registers should be protected from optimizations that might reduce MTBF for each register chain, and controls how many registers should be optimized to increase MTBF with the **Optimize Design for Metastability** option.

For example, if the **Synchronization Register Chain Length** option is set to 2, optimizations such as register duplication or logic retiming are prevented from being performed on the first two registers in all identified synchronization chains. The first two registers are also optimized to improve MTBF when the **Optimize Design for Metastability** option is turned on.
The default setting for the **Synchronization Register Chain Length** option is 2. The first register of a synchronization chain is always protected from operations that might reduce MTBF, but you should set the protection length to protect more of the synchronizer chain. Altera recommends that you set this option to the maximum length of synchronization chains you have in your design so that all synchronization registers are preserved and optimized for MTBF.

To change the global **Synchronization Register Chain Length** option, on the Assignments menu, click **Settings**. Under **Analysis & Synthesis Settings**, click **More Settings**. From the **More Settings** dialog box, you can set the **Synchronization Register Chain Length**.

You can also set the **Synchronization Register Chain Length** on a node or an entity in the Assignment Editor. You can set this value on the first register in a synchronization chain to specify how many registers to protect and optimize in this chain. This individual setting is useful if you want to protect and optimize extra registers that you have created in a specific synchronization chain that has low MTBF, or optimize less registers for MTBF in a specific chain where the maximum frequency or timing performance is not being met. To make the global setting with Tcl, use the following command:

```
set_global_assignment -name SYNCHRONIZATION_REGISTER_CHAIN_LENGTH <number of registers>
```

To apply the assignment to a design instance or the first register in a specific chain with Tcl, use the following command:

```
set_instance_assignment -name SYNCHRONIZATION_REGISTER_CHAIN_LENGTH <number of registers> -to <register or instance name>
```

### Reducing Metastability Effects

You can check your design’s metastability MTBF in the Metastability Summary report, and determine an acceptable target MTBF given the context of your entire system and the fact that MTBF calculations are statistical estimates. A high metastability MTBF (such as hundreds or thousands of years between metastability failures) indicates a more robust design.

This section provides guidelines to ensure complete and accurate metastability analysis, and some suggestions to follow if the Quartus II metastability reports calculate an unacceptable MTBF value. The Timing Optimization Advisor (available from the Tools menu) gives similar suggestions in the Metastability Optimization section.

**Related Information**

- [Metastability Reports](#) on page 13-4

### Apply Complete System-Centric Timing Constraints for the Timing Analyzer

To enable the Quartus II metastability features, make sure that the timing analyzer is used for timing analysis. Ensure that the design is fully timing constrained and that it meets its timing requirements. If the synchronization chain does not meet its timing requirements, MTBF cannot be calculated. If the clock domain constraints are set up incorrectly, the signal transfers between circuitry in unrelated or asynchronous clock domains might be identified incorrectly.

Use industry-standard system-centric I/O timing constraints instead of using FPGA-centric timing constraints.
You should use `set_input_delay` constraints in place of `set_max_delay` constraints to associate each input port with a clock domain to help eliminate false positives during synchronization register identification.

**Related Information**
How Timing Constraints Affect Synchronizer Identification and Metastability Analysis on page 13-3

---

### Force the Identification of Synchronization Registers

Use the guidelines in **Identifying Synchronizers for Metastability Analysis** to ensure the software reports and optimizes the appropriate register chains.

You should identify synchronization registers with the **Synchronizer Identification** set to **Forced If Asynchronous** in the Assignment Editor. If there are any registers that the software detects as synchronous but you want to be analyzed for metastability, apply the **Forced** setting to the first synchronizing register. Set **Synchronizer Identification** to **Off** for registers that are not synchronizers for asynchronous signals or unrelated clock domains.

To help you find the synchronizers in your design, you can set the global **Synchronizer Identification** setting on the **Timing Analyzer** page of the **Settings** dialog box to **Auto** to generate a list of all the possible synchronization chains in your design.

**Related Information**
Identifying Synchronizers for Metastability Analysis on page 13-3

---

### Set the Synchronizer Data Toggle Rate

The MTBF calculations assume the data being synchronized is switching at a toggle rate of 12.5% of the source clock frequency.

To obtain a more accurate MTBF for a specific chain or all chains in your design, set the **Synchronizer Toggle Rate**.

**Related Information**
Synchronizer Data Toggle Rate in MTBF Calculation on page 13-6

---

### Optimize Metastability During Fitting

Ensure that the **Optimize Design for Metastability** setting is turned on.

**Related Information**
MTBF Optimization on page 13-7

---

### Increase the Length of Synchronizers to Protect and Optimize

Increase the **Synchronizer Chain Length** parameter to the maximum length of synchronization chains in your design. If you have synchronization chains longer than 2 identified in your design, you can protect the entire synchronization chain from operations that might reduce MTBF and allow metastability optimizations to improve the MTBF.

**Related Information**
Synchronization Register Chain Length on page 13-7
**Set Fitter Effort to Standard Fit instead of Auto Fit**

If your design MTBF is too low after following the other guidelines, you can try increasing the Fitter effort to perform more metastability optimization. The default **Auto Fit** setting reduces the Fitter’s effort after meeting the design’s timing and routing requirements to reduce compilation time.

This effort reduction can result in less metastability optimization if the timing requirements are easy to meet. If **Auto Fit** reduces the Fitter’s effort during your design compilation, setting the Fitter effort to **Standard Fit** might improve the design’s MTBF results. In the **Settings** dialog box, on the **Fitter Settings** page, set **Fitter effort** to **Standard Fit**.

**Increase the Number of Stages Used in Synchronizers**

Designers commonly use two registers in a synchronization chain to minimize the occurrence of metastable events, and a standard of three registers provides better metastability protection. However, synchronization chains with two or even three registers may not be enough to produce a high enough MTBF when the design runs at high clock and data frequencies.

If a synchronization chain is reported to have a low MTBF, consider adding an additional register stage to your synchronization chain. This additional stage increases the settling time of the synchronization chain, allowing more opportunity for the signal to resolve to a known state during a metastable event. Additional settling time increases the MTBF of the chain and improves the robustness of your design. However, adding a synchronization stage introduces an additional stage of latency on the signal.

If you use the Altera FIFO IP core with separate read and write clocks to cross clock domains, increase the metastability protection (and latency) for better MTBF. In the DCFIFO parameter editor, choose the **Best metastability protection, best fmax, unsynchronized clocks** option to add three or more synchronization stages. You can increase the number of stages to more than three using the **How many sync stages?** setting.

**Select a Faster Speed Grade Device**

The design MTBF depends on process parameters of the device used. Faster devices are less susceptible to metastability issues. If the design MTBF falls significantly below the target MTBF, switching to a faster speed grade can improve the MTBF substantially.

**Scripting Support**

You can run procedures and make settings described in this chapter in a Tcl script. You can also run some procedures at a command prompt. For detailed information about scripting command options, refer to the Quartus II Command-Line and Tcl API Help browser.

To run the Help browser, type the following command at the command prompt:

```
quartus_sh --qhelp r
```

**Related Information**

- **Tcl Scripting**
  For more information about Tcl scripting

- **Quartus II Settings File Reference Manual**
  For more information about settings and constraints in the Quartus II software
• **Command-Line Scripting**
  For more information about command-line scripting

• **About Quartus II Scripting**
  For more information about command-line scripting

## Identifying Synchronizers for Metastability Analysis

To apply the global Synchronizer Identification assignment, use the following command:

```plaintext
set_global_assignment -name SYNCHRONIZER_IDENTIFICATION <OFF|AUTO|"FORCED IF ASYNCHRONOUS">
```

To apply the **Synchronizer Identification** assignment to a specific register or instance, use the following command:

```plaintext
set_instance_assignment -name SYNCHRONIZER_IDENTIFICATION <AUTO|"FORCED IF ASYNCHRONOUS"|FORCED|OFF> -to <register or instance name>
```

## Synchronizer Data Toggle Rate in MTBF Calculation

To specify a toggle rate for MTBF calculations as described on page **Synchronizer Data Toggle Rate in MTBF Calculation**, use the following command:

```plaintext
set_instance_assignment -name SYNCHRONIZER_TOGGLE_RATE <toggle rate in transitions/second> -to <register name>
```

### Related Information

**Synchronizer Data Toggle Rate in MTBF Calculation** on page 13-6

## report_metastability Data Toggle Rate in MTBF Calculation

If you use a command-line or scripting flow, you can generate the metastability analysis reports described in **Metastability Reports** outside of the Quartus II and user interfaces.

The table describes the options for the `report_metastability` and Tcl command.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-append</code></td>
<td>If output is sent to a file, this option appends the result to that file. Otherwise, the file is overwritten.</td>
</tr>
<tr>
<td><code>-file &lt;name&gt;</code></td>
<td>Sends the results to an ASCII or HTML file. The extension specified in the file name determines the file type — either <code>*.txt</code> or <code>*.html</code>.</td>
</tr>
<tr>
<td><code>-panel_name &lt;name&gt;</code></td>
<td>Sends the results to the panel and specifies the name of the new panel.</td>
</tr>
<tr>
<td><code>-stdout</code></td>
<td>Indicates the report be sent to the standard output, via messages. This option is required only if you have selected another output format, such as a file, and would also like to receive messages.</td>
</tr>
</tbody>
</table>

### Related Information

**Metastability Reports** on page 13-4
MTBF Optimization

To ensure that metastability optimization described on page MTBF Optimization is turned on (or to turn it off), use the following command:

```plaintext
set_global_assignment -name OPTIMIZE_FOR_METASTABILITY <ON|OFF>
```

Related Information
MTBF Optimization on page 13-7

Synchronization Register Chain Length

To globally set the number of registers in a synchronization chain to be protected and optimized as described on page Synchronization Register Chain Length, use the following command:

```plaintext
set_global_assignment -name SYNCHRONIZATION_REGISTER_CHAIN_LENGTH <number of registers>
```

To apply the assignment to a design instance or the first register in a specific chain, use the following command:

```plaintext
set_instance_assignment -name SYNCHRONIZATION_REGISTER_CHAIN_LENGTH <number of registers> -to <register or instance name>
```

Related Information
Synchronization Register Chain Length on page 13-7

Managing Metastability

Altera’s Quartus II software provides industry-leading analysis and optimization features to help you manage metastability in your FPGA designs. Set up your Quartus II project with the appropriate constraints and settings to enable the software to analyze, report, and optimize the design MTBF. Take advantage of these features in the Quartus II software to make your design more robust with respect to metastability.

Document Revision History

Table 13-2: Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2014</td>
<td>14.0.0</td>
<td>Updated formatting.</td>
</tr>
<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>Changed to new document template.</td>
</tr>
<tr>
<td>July 2010</td>
<td>10.0.0</td>
<td>Technical edit.</td>
</tr>
<tr>
<td>November 2009</td>
<td>9.1.0</td>
<td>Clarified description of synchronizer identification settings.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Minor changes to text and figures throughout document.</td>
</tr>
</tbody>
</table>

Altera Corporation
Changes

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 2009</td>
<td>9.0.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>

Related Information

Quartus II Handbook Archive
For previous versions of the Quartus II Handbook