7. Managing Metastability with the Quartus II Software

QII51018-9.0.0

Introduction

This chapter describes the industry-leading analysis, reporting, and optimization features that can help you manage metastability in Altera® devices. You can use the Quartus® II software to analyze the average mean time between failures (MTBF) due to metastability when a design synchronizes asynchronous signals, and optimize the design to improve the metastability MTBF. This chapter explains how to take advantage of these features in the Quartus II software, and provides guidelines to help you reduce the chance of metastability effects due to signal synchronization.

What is metastability? As is well known in digital design, all registers have defined signal timing requirements that allow the register to correctly capture data at the inputs and produce the output signal. To ensure reliable operation, the input to the register must be stable for a minimum time before the clock edge (register setup time or t_{SU}) and a minimum time after the clock edge (register hold time or t_H). The register output will then be available after a specified clock-to-output delay (t_{CO}). If the data violates a register's setup and/or hold time requirements, the output of the register may go into a metastable state. In this state, 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 metastability does not occur. Metastability problems commonly occur when a signal is transferred between circuitry in unrelated or asynchronous clock domains, because the designer cannot guarantee that the signal will meet setup and hold time requirements in this case.

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 given the context of your entire system and the fact 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.

For more information about metastability due to signal synchronization, its effects in FPGAs, and how MTBF is calculated, refer to the *Understanding Metastability in FPGAs* white paper. Your overall device MTBF is also affected by other FPGA failure mechanisms that you cannot control with your design. For information about Altera device reliability, refer to the *Reliability Report*.

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 TimeQuest Timing Analyzer, and for select device families. In Quartus II software version 9.0, typical MTBF values are generated for designs using Arria® II GX, Stratix® IV, Stratix III, and Cyclone® III devices. Additional worst-case MTBF values are reported for the fully characterized Stratix III devices.

This chapter contains the following topics:

- "Metastability Analysis in the Quartus II Software" on page 7–2, including the important first step "Identifying Synchronizers for Metastability Analysis"
- "Metastability and MTBF Reporting" on page 7–6
- "MTBF Optimization" on page 7–9
- "Reducing Metastability Effects" on page 7–11, including guidelines to ensure complete and accurate metastability analysis and some suggestions to follow if the Quartus II metastability reports calculate an unacceptable MTBF value

For details about device and version support for the metastability features in the Quartus II software, refer to Quartus II Help.

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 Altera recommends using a standard of three registers for better metastability protection.

The TimeQuest 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 you require longer synchronizer chains in your design.

The first step to enable metastability MTBF analysis and optimization in the Quartus II software is to identify which registers are part of a synchronization register chain.

This section contains the following subsections:

- "What is a Synchronization Register Chain?"
- "Identifying Synchronizers for Metastability Analysis" on page 7–4
- "How Timing Constraints Affect Synchronizer Chain Identification and Metastability Analysis" on page 7–6

For information about the reports generated by the TimeQuest Timing Analyzer, refer to "Metastability and MTBF Reporting" on page 7–6. For more information about optimizing the MTBF, refer to "MTBF Optimization" on page 7–9.

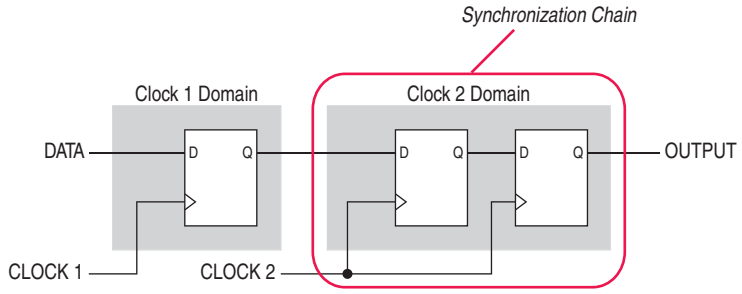
What is a Synchronization Register Chain?

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

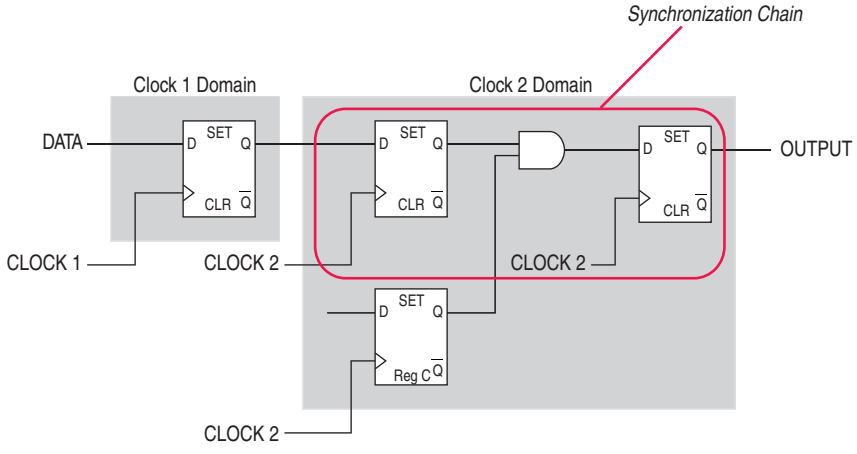
The length of the synchronization register chain is the number of registers in the synchronizing clock domain that meet the above requirements. Figure 7–1 shows a sample synchronization chain of length two, assuming the `OUTPUT` signal fans out to more than one register destination.

Figure 7–1. Sample Synchronization Register Chain



The path between synchronization registers can contain combinational logic, as long as all registers of the synchronization chain are in the same clock domain. Figure 7–2 shows an example of a synchronization register chain that includes logic between the registers.

Figure 7–2. Sample Synchronization Register Chain Containing Logic



The Quartus II software uses the design timing constraints to determine which connections are asynchronous signal transfers, as described in "How Timing Constraints Affect Synchronizer Chain Identification and Metastability Analysis" on page 7–6.

Synchronization registers allow time for a potentially metastable signal to resolve to a known value before the signal is used in the rest of the design. The timing slack available in the synchronizer register-to-register paths provides time for a metastable signal to settle, and is known as the available settling time. The available settling time for a synchronization chain 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.

Identifying Synchronizers for Metastability Analysis

The first step to enable metastability MTBF analysis and optimization in the Quartus II software is to identify which registers are part of a synchronization register chain with the **Synchronizer Identification** option.

You can apply synchronizer identification settings globally, to a design entity, or to specific registers of a synchronization chain. You can use the global options in "Using the Global Synchronizer Identification Setting" on page 7–4 to automatically list or analyze possible synchronizers. You should review this list of possible synchronizers, and identify the confirmed synchronization chains with specific registers assignments as described in "Refining Synchronizer Identification Using the Instance-Specific Assignment" on page 7–5.

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

Using the Global Synchronizer Identification Setting

To set the global **Synchronizer Identification** option, on the Assignments menu, click **Settings**. Under **Timing Analysis Settings**, click on the **TimeQuest Timing Analyzer** page and select the appropriate **Synchronizer Identification** setting under **Metastability Analysis**: **Off**, **Auto**, or **Forced If Asynchronous**. To apply the global assignment with Tcl, use the following command:

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

Use the following guidelines to choose the global setting:

- The default global **Off** setting means that no synchronization registers are automatically analyzed.
- Use the global **Auto** setting to generate a list of likely synchronization chains in your design, based on the software's automatic synchronizer detection criteria.
  - With this setting, any chain synchronizing an asynchronous signal with more than one register is listed as a likely synchronizer if the chain does not contain logic.
  - MTBF is not reported or optimized for automatically detected register chains.

- You can use the global **Forced If Asynchronous** setting to report and optimize MTBF for all asynchronous signal transfers in the design.
  - This setting forces synchronization register identification and MTBF analysis if the software detects any asynchronous signal transfer, even if there is combinational logic or only one register in the synchronization register chain.
  - This setting is likely to identify some registers that were not designed as synchronizers, and thus might report an MTBF that is too conservative. For example, asynchronous reset signals to registers and SignalTap® II Embedded Logic Analyzer signals might generate metastability reports.
  - You can turn off synchronizer identification for specific registers using the instance-specific option, as described below.

Refining Synchronizer Identification Using the Instance-Specific Assignment

Altera recommends applying the **Synchronizer Identification** option **Forced If Asynchronous** to each register in a synchronization chain. Instance-specific assignments ensure that the software analyzes an accurate set of synchronizers to report an accurate MTBF. If you use the global **Auto Synchronizer Identification** setting to detect likely synchronizers, use specific assignments to identify the true synchronizers for MTBF analysis.

Use the Assignment Editor on the Assignments menu to set the **Synchronizer Identification** option on a register or design entity to **Auto**, **Forced If Asynchronous**, **Forced**, or **Off**. To apply the assignment with Tcl, use the following command:

```
set_instance_assignment -name SYNCHRONIZER IDENTIFICATION <AUTO|"FORCED
IF ASYNCHRONOUS"|FORCED|OFF> -to <register name>
```

Use the following guidelines to choose the register-specific settings:

- Enable MTBF analysis and optimization by identifying each synchronizer register with the **Forced If Asynchronous** setting
  - Use the register chains listed by the global **Auto Synchronizer Identification** option to help you identify these registers.
  - There may be additional synchronization chains in your design that are not detected by the **Auto** setting because they contain logic or only one register.
- If you have a specific register or register chain in your design that the software detects as synchronous but you want to be analyzed and optimized for metastability like an asynchronous signal, apply the **Forced** setting to the first synchronization register in the chain.
  - Use this setting as a manual “override” for a signal that the TimeQuest Timing Analyzer does not report as asynchronous, such as a virtual pin associated with a virtual clock.
  - This **Forced** setting is not available globally in the **Settings** dialog box because making this setting globally would incorrectly identify every design register as a synchronizer.

- If some register chains are misidentified as synchronizers when you apply a global or entity level **Forced If Asynchronous** setting, you can disable metastability analysis for specific registers. To do so, use the Assignment Editor to set **Synchronizer Identification** to **Off** for the first synchronization register in these register chains.

How Timing Constraints Affect Synchronizer Chain Identification and Metastability Analysis

The TimeQuest 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, it is important for your design to be 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 TimeQuest 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, you should apply a `set_input_delay` constraint to the input port; otherwise, the input register is reported as a synchronization register. If you constrain a synchronous input port with a `set_max_delay` constraint for a setup (t_{SU}) requirement, this does not prevent synchronizer identification because the constraint does not associate the input port with a clock domain.

Instead, use the following syntax 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 t_{SU} and t_H 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 as described in the previous section.

Metastability and MTBF Reporting

The Quartus II software reports the metastability analysis results in the Compilation Report and TimeQuest Timing Analyzer reports as described in "Metastability Report". The MTBF calculation uses timing and structural information about the design, silicon characteristics, and operating conditions, along with the data toggle rate described in "Synchronizer Data Toggle Rate in MTBF Calculation" on page 7–9.

For more information about how metastability MTBF is calculated, refer to the *Understanding Metastability in FPGAs* white paper.

If you change the **Synchronizer Identification** settings, you can generate new metastability reports by rerunning the TimeQuest Timing Analyzer. However, you should rerun the Fitter first so that the registers identified with the new setting can be optimized for metastability MTBF. For information about metastability optimization, refer to "MTBF Optimization" on page 7–9.

Metastability Report

The Metastability Report provides a summary of the metastability analysis results. In addition to the MTBF Summary and Synchronizer Summary reports, the TimeQuest Timing Analyzer tool reports additional statistics in a report for each synchronizer chain. This section provides more information about the reports.

To view the MTBF Summary and Synchronizer Summary reports, open the Metastability Report in the **TimeQuest Timing Analyzer** section of the Compilation Report. If the software performs multicorner timing analysis, expand the timing analysis results for one of the timing corners, and then select the Metastability Report for those operating conditions.

To view the additional synchronizer Statistics in the TimeQuest report, open the TimeQuest Timing Analyzer from the Tools menu, and double-click **Report Metastability** in the **Tasks** list (or use the `report_metastability` command). You can generate the reports with Tcl commands in addition to the TimeQuest Timing Analyzer user interface; refer to "Scripting Support" on page 7–13 for details.

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 as described in "Identifying Synchronizers for Metastability Analysis" on page 7–4.

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, as described in "How Timing Constraints Affect Synchronizer Chain Identification and Metastability Analysis" on page 7–6.

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.

The MTBF Summary Report reports the **Typical MTBF of Design** for supported devices 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 TimeQuest 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. In the **Settings** dialog box, under **Timing Analysis Settings**, click the **TimeQuest Timing Analyzer** page, and turn on **Enable multicorner timing analysis during compilation**.

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.

Finally, 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

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. To see more detail about each synchronizer, refer to the TimeQuest statistics report described in the following section.

Synchronizer Chain Statistics Report in the TimeQuest Timing Analyzer

The TimeQuest Timing Analyzer provides an additional report for each synchronizer chain. The **Chain Summary** tab matches the Synchronizer Summary information described in the previous section, while the **Statistics** tab adds more details. The Statistics list whether the **Method of Synchronizer Identification** was **User Specified** (with the **Forced if Asynchronous** or **Forced Synchronizer Identification** setting) or **Automatic** (with the **Auto** setting). The **Number of Synchronization Registers in Chain** is also reported. This report provides more details about the parameters that

affect the MTBF calculation: the **Available Settling Time** for the chain and the **Data Toggle Rate Used in MTBF Calculation** (for information about the toggle rate, see "Synchronizer Data Toggle Rate in MTBF Calculation" on page 7–9). There is also additional detail to help you identify where this chain is in your design: the **Source Clock** and **Asynchronous Source** node of the signal, the **Synchronization Clock** in the destination clock domain, and the node names of the **Synchronization Registers** in the chain.

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 the approximate rate at which the data changes, and would like to obtain a more accurate MTBF, use the **Synchronizer Toggle Rate** assignment in the Assignment Editor. Set the data toggle rate, in number of transitions per second, on the first register of a synchronization chain. The TimeQuest Timing Analyzer takes the specified rate into account when computing the MTBF of that particular register chain. You can also apply this assignment to an entity or the entire design. Because a **Synchronizer Toggle Rate** assignment of 0 indicates that the data signal never toggles, the affected synchronization chain will not be reported since it does not affect the reliability of the design. To apply the assignment with Tcl, use the following command:

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

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, as described in "Identifying Synchronizers for Metastability Analysis" on page 7–4. 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 described in the next section.

In addition, the Fitter optimizes identified synchronizers for improved MTBF, by placing and routing the registers to increase the output setup slacks of synchronization registers. Adding slack in the synchronizer chain increases the available settling time for a potentially metastable signal, which improves the chance that the signal will resolve 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 described in the next section.

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:

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

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 described in "Metastability Report" on page 7–7. As discussed in the "Introduction", you should 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.

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

To enable the Quartus II metastability features, make sure that the TimeQuest 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, the MTBF can not be calculated. If the clock domain constraints are set up incorrectly, the signal transfers between circuitry in unrelated or asynchronous clock domains may not be identified correctly.

Use industry-standard system-centric I/O timing constraints instead of using FPGA-centric timing constraints. As described in "How Timing Constraints Affect Synchronizer Chain Identification and Metastability Analysis" on page 7–6, 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.

Force the Identification of Synchronization Registers

Use the guidelines in "Identifying Synchronizers for Metastability Analysis" on page 7–4 to ensure the software reports and optimizes the appropriate register chains.

In summary, 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 **TimeQuest Timing Analyzer** page of the **Settings** dialog box to **Auto** to generate a list of all the possible synchronization chains in your design.

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** as described in "Synchronizer Data Toggle Rate in MTBF Calculation" on page 7–9.

Optimize Metastability During Fitting

Ensure that the **Optimize Design for Metastability** setting described in "MTBF Optimization" on page 7–9 is turned on.

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, as described in "Synchronization Register Chain Length" on page 7–10. 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.

Set Fitter Effort to Standard Fit instead of Auto Fit

If your design MTBF is too low after following the previous guidelines in this section, 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 Fitter effort during your design compilation, setting the **Fitter effort** to **Standard Fit** might improve the design's MTBF results. On the Assignments menu, click **Settings**. On the **Fitter Settings** page, set **Fitter effort** to **Standard Fit**.

If Possible, 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 Altera recommends using a standard of three registers for better metastability protection. However, using chains of length two or even three 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. Of course, adding a synchronization stage does introduce an additional stage of latency on the signal.

If you use the Altera FIFO megafunction with separate read and write clocks to cross clock domains, increase the metastability protection (and latency) for better MTBF. In the MegaWizard™ Plug-In Manager for the DCFIFO function, choose the **Best metastability protection, best fmax, unsynchronized clocks** option to add 3 or more synchronization stages. You can increase the number of stages to more than 3 using the **How many sync stages?** setting.

If Possible, 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 ↵
```

The *Quartus II Scripting Reference Manual* includes the same information in PDF form.

For more information about Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*. Refer to the *Quartus II Settings File Reference Manual* for information about all settings and constraints in the Quartus II software. For more information about command-line scripting, refer to the *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

Identifying Synchronizers for Metastability Analysis

To apply the global Synchronizer Identification assignment described on page "Identifying Synchronizers for Metastability Analysis" on page 7–4, use the following command:

```
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:

```
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" on page 7–9, use the following command:

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

report_metastability TimeQuest and Tcl Command

If you use a command-line or scripting flow, you can generate the metastability analysis reports described in "Metastability Report" on page 7–7 outside of the Quartus II and TimeQuest user interfaces. Table 7–1 describes the options for the `report_metastability` TimeQuest and Tcl command.

Table 7–1. report_metastabilty Command Options

Option	Description
`-append`	If output is sent to a file, this option appends the result to that file. Otherwise, the file is overwritten.
`-file <name>`	Sends the results to an ASCII or HTML file. The extension specified in the file name determines the file type—either ***.txt** or ***.html**.
`-panel_name <name>`	Sends the results to the panel and specifies the name of the new panel.
`-stdout`	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.

MTBF Optimization

To ensure that metastability optimization described on page "MTBF Optimization" on page 7–9 is turned on (or to turn it off), use the following command:

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

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" on page 7–10, 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, use the following command:

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

Conclusion

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 and follow the guidelines in this chapter as required to make your design more robust with respect to metastability.

Referenced Documents

This chapter references the following documents:

- *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*
- *Quartus II Scripting Reference Manual*
- *Quartus II Settings File Reference Manual*
- *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*
- *Understanding Metastability in FPGAs* white paper

Document Revision History

Table 7–2 shows the revision history for this chapter.

Table 7–2. Document Revision History

Date and Version	Changes Made	Summary of Changes
March 2009 v9.0.0	Initial release.	—

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