
Performing Equivalent Timing Analysis Between Altera TimeQuest and Xilinx Trace

Introduction

Most hardware designers who are qualifying FPGA performance normally run software benchmark comparisons of FPGAs from different vendors to determine which vendor provides the largest margin for their timing requirements. To produce equivalent or fair performance comparisons, designers must consider that different software tools have different default timing analysis and optimization behaviors.

Altera's Quartus® II TimeQuest timing analyzer generally calculates all possible register-to-register and complex clock structures using the most conservative assumptions by default. In contrast, Xilinx's Trace timing analyzer does not analyze many of these complex structures as extensively, so comparing the software tools on this basis unfairly penalizes Quartus II performance. TimeQuest makes the most stringent assumptions, allowing users to see possible problems by default. This can cause users comparing the two tools to perceive, initially, that Quartus II performance is inferior, which is not the case in actual practice. For instance, if designers determine that the reported problems are not significant, they can adjust TimeQuest not to report these problems, changing the reported performance of the design.

This document covers the differences in timing analysis between Altera's TimeQuest and Xilinx's Trace, and explains how to configure the tools to provide equivalent performance comparison. Quartus II software's system-level performance report can be more favorable when the settings for TimeQuest are adjusted to perform timing analysis equivalent to the timing analysis performed by Trace.

• For a more in-depth discussion on benchmarking a single design, see Altera's white paper, *Guidance for Accurately Benchmarking FPGAs*. It is assumed the reader is familiar with using TimeQuest, which is documented in the *TimeQuest Timing Analyzer* chapter of the *Quartus II Handbook*. It is suggested that Altera Classic Timing Analyzer users read the *Switching to the TimeQuest Timing Analyzer* handbook chapter to understand the benefits of switching to TimeQuest.

Differences in the TimeQuest and Trace Timing Analyzers

Altera's Quartus II software package offers TimeQuest timing analyzer support for Altera® device families that include Stratix® series FPGAs, HardCopy® II series structured ASICs, Cyclone® series FPGAs, and MAX® II CPLDs. TimeQuest is an easy-to-use, second-generation, ASIC-strength static timing analyzer that supports the industry-standard Synopsys Design Constraints (SDC) format. Xilinx's ISE software package offers Trace as its timing analyzer. TimeQuest and Trace are fundamentally different in their constraint and timing analysis philosophies. The following sections outline major differences that affect the out-of-the-box experience.

Cross-Domain Clock Analysis

Because TimeQuest assumes all clocks identified are related by default, it also analyzes paths crossing multiple clock domains. Trace does not perform this analysis by default in its "advanced" analysis setting (the `trce -a` option), and instead requires additional constraints for performing cross-domain analysis. Neither tool analyzes unconstrained clock domains.

In general, TimeQuest makes worst-case assumptions about clocks without relationship settings. This is essential for ensuring a design's correct behavior. If there are no problems, users can relax TimeQuest assumptions with timing settings. Trace generally analyzes timing based on relaxed assumptions, unless users make specific relationship settings. While this causes TimeQuest's default results to appear worse, this is the product of thorough and complete timing analysis. Trace users run the risk of missing legitimate crossings requiring synchronization during timing analysis, and therefore run the risk of not having a working design. [Table 1](#) compares the default method of how TimeQuest and Trace identify clocks, and relates the clocks for both the default behavior and when timing constraints are applied.

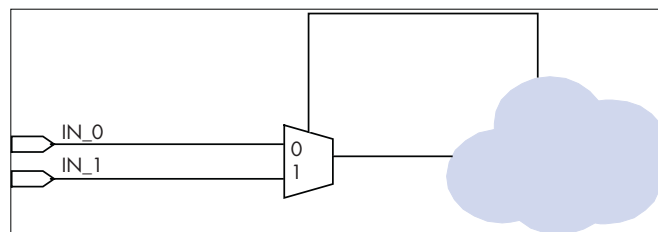
Table 1. Comparison of Clocks Analyzed and Their Relationships in Xilinx Trace and Altera TimeQuest Timing Analyzer

	Xilinx Trace Timing Analyzer	Altera TimeQuest Timing Analyzer
Clocks Analyzed by Default	Analyzes clock networks that directly feed clock pins of registers as clocks.	Analyzes clocks as being the most direct port, register, or phase-locked loop (PLL) from which clock pins of registers are derived.
Default Clock Relationships	Does not analyze register-to-register paths that have a different source and destination clock.	Relates all identified clocks .
Clocks Analyzed With Timing Constraints	Analyzes only constrained clocks.	Analyzes only constrained clocks.
Clock Relationships With Timing Constraints	Only analyzes paths with source and destination clocks that are the same or have timing settings relating them.	Relates all identified clocks, unless there are timing settings to cut relationships. (set_disable_timing)

Combinational Loop Structures

Combinational loops (Figure 1) are logic structures designed to utilize outputs from the structure as partial input to the same structure. The total combinational delay from the source to the destination register is theoretically increased because of this extra logic path.

Figure 1. Combinational Loop Example

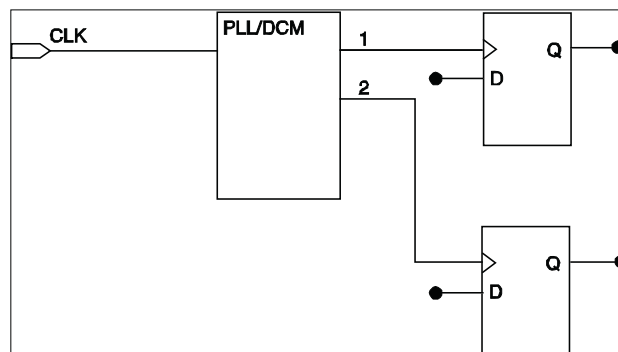


The Trace timing analyzer does not have the capability to account for all combinational loops. The ISE software issues a warning that some paths taken through connections that close logic loops may not be analyzed. TimeQuest automatically accounts for all combinational loops by default and adds the delays to the total register-to-register f_{MAX} . If the user wants to remove undesired combinational loop paths from timing analysis, TimeQuest provides the ability to break the loop using the set_disable_timing command.

Designs With PLL or DCM

As shown in Figure 2, a Xilinx digital clock manager (DCM) or Altera PLL can provide signal de-skew or clock synthesis, such as clock multiplication, division, or phase shifting. Clock synthesis affects the timing constraints placed on an FPGA because of different clock rates, clock relationships, or phases. There are necessary differences in constraining and reporting designs that use DCMs or PLLs, especially when using multiple outputs.

Figure 2. PLL or DCM Example



Timing constraints to a DCM or a PLL are set by applying them to their respective input clocks. TimeQuest recommends also using the `derive_pll_clocks` SDC setting to obtain PLL parameters specified from using the MegaWizard® Plug-In Manager for the `altpll` megafunction. These PLL parameters help determine the constraints for the output clocks of the PLL. It is important to have the same parameters in DCMs and PLLs such that the relationships between output clocks and input clocks are set the same way for both vendors. When applying constraints on the inputs of DCMs and PLLs, verify that their respective output clocks have equivalent constraints resulting from the parameters.

Table 2 summarizes the timing analysis differences between Xilinx Trace and Altera TimeQuest.

Table 2. Timing Analysis Differences in Xilinx Trace and Altera TimeQuest Timing Analyzer

Design Structure	Xilinx Trace Timing Analyzer	Altera TimeQuest Timing Analyzer
Default Cross-Domain Clock Analysis	Register-to-register paths clocked by different clocks are not analyzed. (Exception: clocks from the same DCM)	Identified clocks are related and all paths of registers fed by them are analyzed.
Combinational Loop	Loops are not guaranteed to be analyzed. Warning reported.	The longest and shortest paths of all loops are analyzed for all inputs to all outputs. Optionally, such paths can be ignored.
Designs With DCMs/PLLs	Analyzed if the constraint is applied to the input of a DCM.	Analyzed when the <code>derive_pll_clocks</code> SDC setting is used (done by default) to obtain PLL clock settings made in the PLL function.(1)

Note:

(1) To use the PLL input frequency setting made in the MegaWizard Plug-In Manager for the `altpll` function as the PLL input clock constraint, use `derive_pll_clocks -create_base_clocks` option.

How To Perform Equivalent Timing Analysis

To achieve a fair comparison between TimeQuest and Trace, users must adjust the tools so that they analyze the same paths in a design, since the tools are based on different philosophies for optimizing performance and performing timing analysis. The user must constrain all relevant clocks in both tools. Constraining all clocks as separate domains is the approach taken for the purposes of this document, although a subset of clocks may also be constrained, as appropriate.

To gather information on clocks in the design, perform the following steps:

1. Perform an initial compilation using default values for both the ISE software and the Quartus II software.
2. Because the Quartus II software reports all possible paths by default, extract the clock names from the fitter report file by clicking on the usage column of the Control Signals table (see Figure 3). Note that all signals used by registers as clock signals are listed in this table. Also, perform “advanced” timing analysis in Trace (`trce -a`) and extract the clock names from the report file.
3. Verify and correlate that the clock names used in the Quartus II software and ISE initial report file are the same.
4. After identifying the clocks, apply period or f_{MAX} constraints to the clocks in both tools.

There may be clocks listed in the fitter report file that do not appear in the Trace report file. These clocks must be constrained with a default value to ensure that TimeQuest does not ignore them.

Figure 3. Quartus II Fitter Report

Compilation Report - Control Signals				
Control Signals				
	Name	Location	Fan-Out	Usage
1	TopRx:TopRxPIIRx53m:PIRx53mPD_3PDRx53mIun1_PDRst_x	LCCOMB_X15_Y2_N16	2	Async. clear
2	TopRx:TopRxPIIRx77m:PIRx77mPD_2PDRx77mIun1_PDRst_x	LCCOMB_X15_Y2_N22	2	Async. clear
3	TopTx:TopTxPIITx103m:PIITx103mPD_1PDTx103mIun1_PDRst_x	LCCOMB_X12_Y7_N24	2	Async. clear
4	TopTx:TopTxPIITx77m:PIITx77mPDtIun1_PDRst_x	LCCOMB_X21_Y1_N0	2	Async. clear
5	mpint:MPIntReset_i	LCCOMB_X20_Y7_N28	21	Async. clear
6	ARstF	PIN_87	1815	Async. clear.
7	Clk26m	PIN_138	40	Clock
8	ClkRBC0	PIN_28	143	Clock
9	ClkRBC1	PIN_24	215	Clock
10	ClkRx53m	PIN_132	204	Clock
11	ClkRx77m	PIN_130	113	Clock
12	ClkTx103m	PIN_23	308	Clock
13	ClkTx53m	PIN_129	64	Clock
14	ClkTx77m	PIN_27	189	Clock
15	LineRelTx	PIN_164	1	Clock
16	SClk	PIN_131	173	Clock
17	SelClk:SelClk77mClkSel_x	LCCOMB_X15_Y8_N0	219	Clock
18	SelClk_1:SelClk53mClkSel_x	LCCOMB_X18_Y12_N6	13	Clock
19	SelClk_2:SelClk53mRxClkSel_x	LCCOMB_X21_Y11_N30	167	Clock
20	SerEn	PIN_92	14	Clock
21	TopRx:TopRxPIIRx53m:PIRx53mCount31Rx53m:Count31Rx53mClkL0L_Z	LCFF_X14_Y1_N9	1	Clock
22	TopRx:TopRxPIIRx53m:PIRx53mCount31Rx53m:Count31Rx53mPDRIFb53m	LCFF_X15_Y1_N5	5	Clock
23	TopRx:TopRxPIIRx53m:PIRx53mRfISlct_3RfISlctRx53mPDRfISlct_x	LCCOMB_X15_Y2_N28	1	Clock
24	TopRx:TopRxPIIRx77m:PIRx77mCount8Rx77m:Count8Rx77mClkL0L_Z	LCFF_X12_Y1_N17	1	Clock
25	TopRx:TopRxPIIRx77m:PIRx77mCount8Rx77m:Count8Rx77mPDRfIFb77m_Z	LCFF_X13_Y1_N25	4	Clock
26	TopRx:TopRxPIIRx77m:PIRx77mRfISlct_2RfISlctRx77mPDRfISlct_x	LCCOMB_X15_Y2_N26	2	Clock
27	TopTx:TopTxPIITx103m:PIITx103mCount11Tx103m:Count11Tx103mClkL0L_Z	LCFF_X13_Y12_N13	1	Clock
28	TopTx:TopTxPIITx103m:PIITx103mCount11Tx103m:Count11Tx103mPDRfIFb103m_Z	LCFF_X9_Y7_N23	4	Clock
29	TopTx:TopTxPIITx103m:PIITx103mRfISlct_1RfISlctTx103mPDRfISlct_x	LCCOMB_X12_Y7_N26	2	Clock
30	TopTx:TopTxPIITx77m:PIITx77mFracDivTx77m:FracDivTx77mClkL0L_Z	LCFF_X19_Y1_N25	1	Clock
31	TopTx:TopTxPIITx77m:PIITx77mFracDivTx77m:FracDivTx77mPDRfIFb77m_Z	LCFF_X20_Y1_N21	4	Clock
32	TopTx:TopTxPIITx77m:PIITx77mRfISlct:RfISlctTx77mPDRfISlct_x	LCCOMB_X21_Y1_N20	2	Clock
33	G_517	LCCOMB_X25_Y10_N4	8	Clock enable
34	TopRx:TopRxPIIRx53m:PIRx53mFracDivRx53m:FracDivRx53mN_84_i	LCCOMB_X14_Y3_N28	1	Clock enable
35	TopRx:TopRxPIIRx77m:PIRx77mCount5Rx77m:Count5Rx77mSyncRf20	LCCOMB_X6_Y1_N12	1	Clock enable
36	TopRx:TopRxStm4Rx:Stm4RxGenFifoPtr_Generic_512s_9s_400s_512s_9s_0s:GenFifoPtrRxIN_379_i	LCCOMB_X12_Y6_N28	9	Clock enable
37	TopRx:TopRxStm4Rx:Stm4RxIpbip12det:bp12detbeginframe	LCCOMB_X14_Y6_N18	24	Clock enable
38	TopRx:TopRxStm4Rx:Stm4RxIstm4ovhdext:stm4ovhdextIcidloadsync_0_a2	LCCOMB_X14_Y9_N20	38	Clock enable
39	TopRx:TopRxStm4Rx:Stm4RxIstm4ovhdext:stm4ovhdextIcidword_reg_1_sqmuxa~_DUP_COMB_4	LCCOMB_X17_Y9_N4	1	Clock enable

The following sections outline the steps needed to ensure that the tools analyze the same paths. These steps involve making the constraints in the Quartus II software match the default ISE behavior and equivalent settings in the ISE software. The effectiveness of these recommendations may vary with the characteristics of the design used to measure benchmarks. An intimate knowledge of the design structure is necessary to reduce inequivalency.

Correcting Cross-Domain Clock Analysis Differences

Since the ISE software does not analyze paths between different clocks, these paths need to be explicitly ignored in TimeQuest software to perform equivalent analysis. To ignore these paths:

1. Identify the clocks.
2. Use TimeQuest's set_clock_groups SDC constraint for each clock domain to unrelate them from the other clock domains.

Correcting Combinational Loop Analysis Differences

Since the ISE software does not analyze all paths for combinational loop structures, paths that are not analyzed by Trace need to be explicitly cut in the Quartus II software to perform equivalent analysis. To disable these paths so that TimeQuest will ignore them:

1. Identify the paths in the combinational loop structures.
2. Create “set_disable_timing” SDC settings for each.

Correcting Analysis for Designs With PLL/DCM

To perform equivalent timing analysis for clocks generated by DCMs in ISE and PLLs in the Quartus II software, perform the following procedure. By constraining the design in this way, both tools will analyze the paths between the PLL/DCM generated clocks.

1. Assign a timing constraint to the input of the DCM. The ISE software applies constraints to the output clocks and sets values according to the DCM clock frequency parameters.
2. In TimeQuest, assign the same constraint to the input of the PLL and apply the derive_pll_clocks SDC setting.
3. Verify that the input and output clock frequency parameters in the altpll MegaWizard causes the Quartus II software to constrain each output clock frequency to match the settings made in the ISE software.

Timing Analysis Correction Results

Below is an example of before and after results of configuring the Quartus II software to perform equivalent timing analysis to the ISE software. In this example, the customer has constrained all clock signals in the Xilinx ISE tool. For the Quartus II software, there are two compilations:

- Default compilation, where the Quartus II software performs conservative, worst-case analysis on all paths
- Modified timing analysis, where the corrections described in the previous section of this document are applied to achieve a fair comparison of the two tools

Table 3 shows Quartus II data for a design example after default compilation and after corrections are applied.

Table 3. Timing Analysis Correction Example

Characteristic	Default Compilation	After Corrections Are Applied
Size	8,739 logic elements (LEs)	8,739 LEs
Clock1 Worst-Case Slack	-2.353 ns	2.383 ns
Clock2 Worst-Case Slack	-12.366 ns	2.540 ns

After the user configures the Quartus II software to perform timing analysis equivalent to that performed by the ISE software, both reported worst-case slacks are positive. This design has many paths that go across clock domains, and the Quartus II software analyzes them all by default, hence the negative worst-case slack. The worst-case slack improves after the cross-clock paths between two pin clocks are cut off from analysis. The critical path of Clock2 after default compilation is 19.033 ns, of which 5.226 ns is clock skew. After the clock relationships are broken, Clock2's critical path has two registers directly driven by Clock2 with almost no clock skew.

Differences in Timing Analyzer Accuracy

TimeQuest meets industry standards static timing analysis with the many analysis capabilities and features it supports. For its timing calculations, a fundamental advantage is that TimeQuest analyzes hold constraints as well as setup constraints. These are necessary to perform a complete analysis on any clock relationship and provide significantly more confidence than analyzing setup times alone. Without this analysis, designs can and do fail hold times.

Furthermore, the Quartus II software's fitter queries TimeQuest's timing analysis to make well informed decisions during its optimizations. This section describes features in TimeQuest that deliver accurate and optimal results during timing analysis.

Recovery/Removal Analysis

Recovery time is the minimum length of time during which the de-assertion of an asynchronous control signal must be stable before the next active clock edge. Removal time is the minimum length of time the de-assertion of an asynchronous control signal must be stable after the active clock edge. Recovery and removal analysis checks whether recovery and removal times are met in the design.

TimeQuest performs recovery and removal analysis by default, including analysis of asynchronous reset paths, as part of its comprehensive static timing analysis. Recovery and removal times are optimized by the fitter and summarized in TimeQuest's report. In Trace, analysis of asynchronous control signals is disabled by default. Trace users need to recognize on their own that their asynchronous control signals are not guaranteed unless they enable Trace to analyze those paths.

Multicorner Timing Analysis

Multicorner timing analysis verifies that the timing constraints specified for a design meet a range of a device's operating conditions (i.e., process, voltage, and temperature (PVT)). By examining different corners of PVT, multicorner timing analysis provides more accurate timing results than just using a single timing model for a selected speed grade. This analysis becomes more important as devices geometries become smaller than 90 nm.

The TimeQuest timing analyzer performs multicorner timing analysis in its timing report. It analyzes the design with the available slow-corner timing models of selected devices (e.g., a slow model at 0°C and another slow model at 85°C in Stratix III and Cyclone III FPGAs). It also performs another pass of the same analysis with the fast-corner timing model. This allows the designer to examine the worst-case timing results across the multiple corners. Quartus II software's fitter can automatically optimize to meet timing at multiple timing corners. Trace only examines one slow corner of the selected speed grade, and offers an option for their users to perform a reanalysis with its fast corner of the place-and-route results compiled with the slow corner.

Relative Minimum/Maximum Timing Analysis

The purpose of supplying relative minimum and maximum delays in timing models is to capture a range of delays due to sources of uncertainty. The main source of uncertainty is due to PVT variation.

Both Altera and Xilinx perform worst-case analysis with their relative minimum and maximum delays. This can be seen by different delays used for setup and hold analysis of the path, where the selection of either the minimum or maximum delay appropriately calculates the worst-case timing result.⁽¹⁾

Rise/Fall Analysis

Rising and falling signals have different delays through a circuit element. Instead of using only the slowest delay, rise/fall analysis examines the combinations of rise (R) and fall (F) transitions which make up four components of a timing model delay (RR, RF, FR, and FF, where each represents the time it takes for the first transition at a circuit element's input to produce the second transition at the element's output.) This analysis offers higher precision in timing results. The TimeQuest timing analyzer performs rise/fall analysis, while this feature is not available in the timing analysis offered in Xilinx's ISE 9.2i SP3.

Synthesis EDA Tools Estimation

Most EDA synthesis tools used today in FPGA design provide area and performance estimation capabilities to allow users to verify their design resources quickly without having to compile through the whole FPGA place-and-route flow, which can take a long time to complete. Although Altera works very closely with EDA vendors, there are some inherent inaccuracies that make this comparison method unsuitable for performing resource and performance benchmark tests.

To obtain accurate comparisons, the design *must be* compiled using the FPGA vendor's place-and-route tool (Quartus II software or the ISE software), because:

- EDA tools only provide estimates, not actual results based on placement and routing. Actual performance and area results can vary significantly after remapping, placement, and routing, primarily because analysis on a placed-and-routed design is based on physical information not available to the third-party tool. Having register packing performed by Quartus II software, for example, reduces area utilization by 11 percent compared to not packing registers. The ISE software also produces actual performance and device utilization data that differs from that provided by EDA synthesis tools.
- The accuracy of the estimates depends on the existence of black boxes (containing Altera megafunctions or Xilinx CoreGen cores). EDA synthesis tools do not synthesize black box functions and cannot perform timing estimations on them. To alleviate this issue, Altera offers clearbox methods to provide EDA-synthesis-tool timing models for black box functions. Similarly, Xilinx offers EDFs that contain timing information about their black box functions.
- Most FPGA EDA synthesis tools provide system f_{MAX} estimates instead of core f_{MAX} . The Quartus II software and the ISE software report core frequency. TimeQuest provides signoff quality analysis, which allows interoperability due to the use of industry-standard SDC. If designers use PrimeTime, TimeQuest's SDC makes porting the constraints into PrimeTime easier than from ISE.

■ For more information, see *TB 84: Differences in Logic Utilization Between Quartus II and Synplify Report Files*.

Summary

Each FPGA software tool has its own set of parameters and environment determined by the FPGA vendor to be optimal for their products. Altera Quartus II software and Xilinx ISE software have different default environments, causing out-of-the-box software benchmarking to occasionally produce non-equivalent comparisons, especially in large-density designs. Combinational loops and DCM/PLL usage are analyzed differently by ISE software and by Quartus II software.

In order to perform fair cross-vendor software benchmarking, some constraints and settings adjustments need to be made where applicable. The Quartus II timing analyzer performs complete and more thorough analysis of all permutations of paths, and assumes the worst-possible case when reporting timing analysis. The ISE software, by contrast, omits certain structures during timing analysis, reporting potentially inaccurate higher performance. When Quartus II constraints are set to mimic those of the ISE software, the performance reported by the Quartus II software tends to improve as well.

More timing analysis accuracy features are built into TimeQuest. In addition to setup constraints, hold constraints can be specified and TimeQuest uses them in its calculations. Recovery/removal, multicorners, relative minimum/maximum, and rise/fall analysis deliver accurate and optimal results.

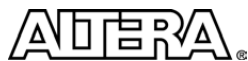
Quartus II software and ISE software must be used to completely compile a design for accurate timing analysis because EDA synthesis tools alone do not provide a complete picture of timing in the device. To facilitate the performance of multiple full compilations, the Quartus II Design Space Explorer utility makes it easy to perform multiple compilations with different settings to improve performance beyond default values.

Further Information

1. Note that the different available channels of the same type of delay represent another source of uncertainty. Each input of a digital signal processing (DSP) block has a different delay, and those inputs are encompassed between the relative minimum and maximum DSP input delay models.
- *Guidance for Accurately Benchmarking FPGAs:*
www.altera.com/literature/wp/wp-01040.pdf
 - *TimeQuest Timing Analyzer* chapter of the *Quartus II Handbook:*
www.altera.com/literature/hb/qts/qts_qii53018.pdf
 - *Switching to the TimeQuest Timing Analyzer* chapter of the *Quartus II Handbook:*
www.altera.com/literature/hb/qts/qts_qii53019.pdf
 - *TB 84: Differences in Logic Utilization Between Quartus II and Synplify Report Files:*
www.altera.com/literature/tb/tb84.pdf

Acknowledgements

- Carolyn Lam-Leventis, Sr. Software Engineer, Software and System Engineering, Altera Corporation
- Kalen Brunham, Member of Technical Staff, Software and System Engineering, Altera Corporation
- Alexander Grbic, Manager, Software and System Engineering, Altera Corporation



101 Innovation Drive
San Jose, CA 95134
www.altera.com

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.