White Paper

Performing Equivalent Timing Analysis Between the Altera Quartus II Software and Xilinx ISE

Introduction

Most hardware designers who are qualifying FPGA performance normally run “bake-off”-style software benchmark comparisons of FPGAs from different vendors to determine which vendor provides the largest margin for their timing requirements. Unfortunately, out-of-the-box design compilations do not produce equivalent or fair performance comparisons because different software tools have different default timing analysis and optimization behaviors.

The Quartus II timing analyzer, in general, calculates all possible register-to-register and complex clock structures using the worst possible assumptions. The Xilinx ISE software does not analyze many of these complex structures, and comparing the software tools on this basis unfairly penalizes Quartus II performance. The Quartus II software 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 the Quartus II software to not report these problems, changing the reported performance of the design.

This document covers the differences in timing analysis between Xilinx ISE and Altera Quartus II software and explains how to configure the tools to provide equivalent performance comparison. Adjusting settings for the Quartus II software to perform timing analysis equivalent to the timing analysis performed by ISE improves Quartus II system-level performance.

Differences in Philosophy

The Altera Quartus II software and Xilinx ISE software are fundamentally different in their constraint and timing analysis philosophies. The following sections outline major differences that affect out-of-the-box experience.

Full Analysis vs. Constrained Analysis

The Quartus II software analyzes all possible paths whether they are constrained or not. By default, the Quartus II timing analyzer analyzes everything, including gated clocks, registered clocks and combinatorial loops. The user can cut these paths. In the absence of timing constraints, the Quartus II software attempts to optimize all clock paths in the design. This accounts for worst-case scenarios and identifies all potential problems.

By default, the ISE software only analyzes constrained paths and does not optimize or report unconstrained paths. The ISE no-constraint or “Advanced” analysis (the `trce -a` option) has limited use in competitive comparisons because the design must be fully constrained in ISE to produce good results. This mode is useful in identifying all clocks in the design however. The ISE software also does not account for certain complex clock or data structures, such as registered clocks and combinatorial loops. These structures are discussed in detail in later sections.

Clock Relation

The Quartus II software assumes clock signals coming from pins and their derivative clocks are related by default. Timing constraints can be declared related or unrelated. The ISE software requires timing constraints to relate clocks with their derivativs. The ISE software provides an "Advanced" timing analysis mode (see below) for unconstrained designs but it does not assume any clock relationships. To ensure a fair comparison, the same constraint, assumed or declared, must be made in both tools.

Cross-Domain Clock Analysis

Because the Quartus II software assumes all clock pins and their derivatives are related by default, it also analyzes paths crossing multiple clock domains. The ISE software does not perform this analysis by default and requires combinatorial delay constraints for performing cross-domain analysis. Neither tool analyzes unrelated clock domains.

In general, Quartus II makes strict assumptions about any unconstrained paths and always performs analysis based on the worst possible scenarios. This makes designers aware of any potential problems. If there are no problems, users can loosen the strictness of Quartus II timing analysis by applying timing constraints. The ISE software generally analyzes timing based on relaxed assumptions unless users make tighter constraints. While this causes Quartus II default timing analysis results appear worse, Altera considers thorough and complete timing analysis to be a necessary feature.

Differences in the Quartus II and ISE Timing Analyzers

The main difference between the Quartus II and ISE timing analyzers is the types of paths or clock structures that each tool analyzes. The Quartus II software and ISE software differ significantly in their analysis of the commonly used structures described in the following sections.

(This white paper assumes that constraints exist on all clocks of interest in the example designs described in the following sections.)

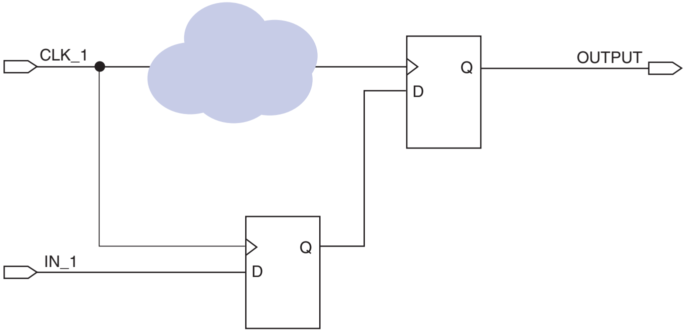
Clock Skew Analysis

Clock skew is the difference in the delay for a clock reaching 2 or more registers. Clock skew must be accounted for at all times because it can degrade register-to-register maximum frequency depending on the polarity. This effect is especially pronounced when the clock signal goes through logic and/or uses general routing resources before it reaches the registers. Figure 1 illustrates an example of a register-to-register path with some logic delay in the clock path.

Clock skew may introduce the following problems that the timing analyzer must account for:

- Reduced performance. The combinatorial data path delay is no longer the only limit on the register-to-register maximum frequency. FPGA timing analysis tools must add the difference in timing between source and destination register clocking to the minimum period analysis.
- Hold-time violation. If the clock skew is larger than the data path delay, as clock skew increases it may cause an illegal hold-time violation to the destination register, independent of frequency. In most cases, this effect is undesirable because the destination register captures an incorrect data pattern and renders the design invalid. It is essential that an FPGA timing analyzer tool detect this scenario.

Figure 1. Clock Skew Example

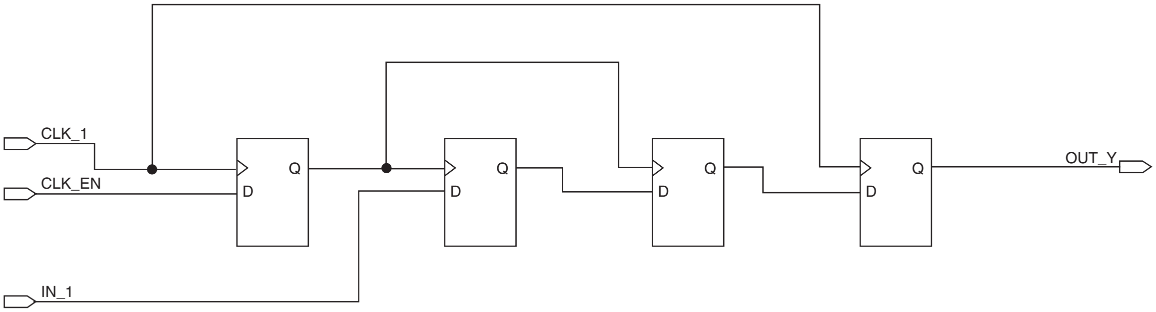


The Quartus II software accounts for all possible clock skew during timing analysis. Because of its importance, the timing analysis process is not optional. The ISE software, by contrast, does not automatically account for clock skew on non-global resources. Users must run the TRACE timing analyzer program with the -skew option to turn on the clock skew calculation. Users must check whether clock skew analysis is turned on in the ISE software for a fair comparison with the Quartus II software.

Registered-Clock Structures

Registered-clock structures are clock signals driven by the output of registers. Cascaded registered-clock structures where more than 1 registered-clocks are connected in a chain are not uncommon. These structures are generated by synthesis tools as a function of clock-enable requirements, multi-cycle requirements, or user created clock divider requirements.

Figure 2. Registered Clock Example



The Quartus II software adds the microtiming clock-to-out (t_{CO}) delay of the registered clock as part of the clock skew and reduces the register-to-register maximum frequency. This is done to account for the worst case scenario. In a design similar to that shown in Figure 2, the path between reg3 and reg4 are analyzed using reg1 as clock skew delay. The ISE software does not analyze paths using registered clocks. To analyze registered clocks, the ISE software treats the output of the registered clock as a separate domain.

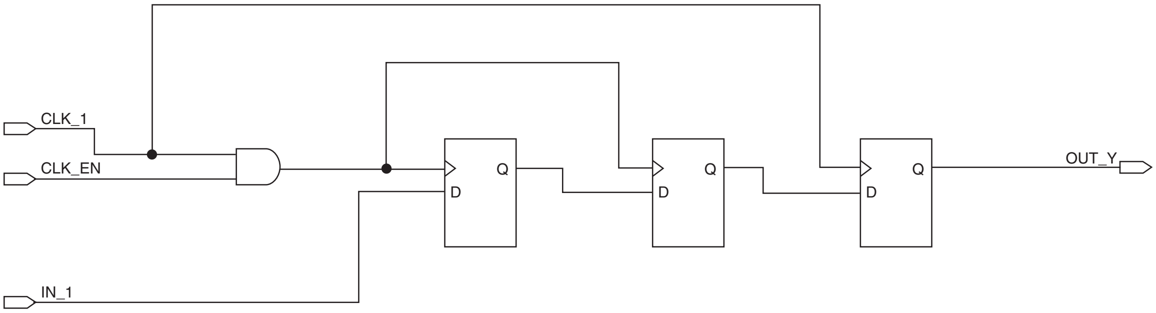
Users of the Quartus II software must assign internal registered clock signals to their own domains for equivalent timing analysis between the two tools.

Gated Clock Structures

Gated-clock structures are clock paths driven by logic. Different registers can also use clock signals from different tap points of logic, effectively making use of different signals. Gated clocks cause clock skews among the affected registers and add to the minimum timing delay. The clock skews must be accounted for during timing analysis because they reduce register-to-register maximum frequency.

If the user has constrained the design, the Xilinx ISE software only analyzes and reports constrained paths. In contrast, the Quartus II software reports all possible and worst-case gated-clock paths by default. Both tools correctly measure clock skews from gated-clock logic, if users of the ISE software properly constrain their designs. When analzying timing for the structure shown in Figure 3, the Quartus II software calculates the f_{MAX} of the clk and clk_en signals, but the ISE software reports neither unless they are constrained.

Figure 3. Gated Clock Structure Example

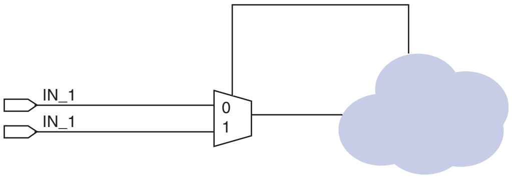


In designs with multiple inputs to logic that result in a clock signal, the Quartus II software analyzes and reports all inputs as clocks unless the user specifies otherwise. The ISE software does not report any of these signals as clocks unless directed to do so by the user.

Combinatorial Loop Structures

Combinatorial loops are logic structures designed to utilize outputs from the structure as partial input to the same structure. The total combinatorial delay from the source to the destination register is theoretically increased because of this extra logic path. The majority of combinatorial loops are false paths or 'don't care' paths however. They are most often caused by synthesis of incomplete `if-else` or `case` statements.

Figure 4. Combinatorial Loop Example

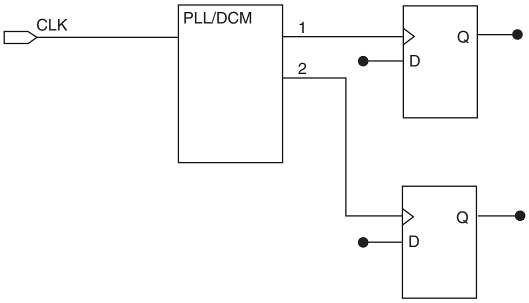


The ISE timing analyzer does not have the capability to account for combinatorial loops. The ISE software ignores combinatorial loops and a warning is issued. The Quartus II software automatically accounts for all combinatorial loops by default and adds the delays to the total register-to-register f_{MAX}. If the user wants to remove undesired combinatorial loop paths from timing analysis, the Quartus II software provides the option for timing analysis to ignore the false path.

Designs with PLL/DCM

A Xilinx DCM or Altera PLL is used to 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 DCM/PLL, especially when using multiple outputs.

Figure 5. PLL/DCM Example

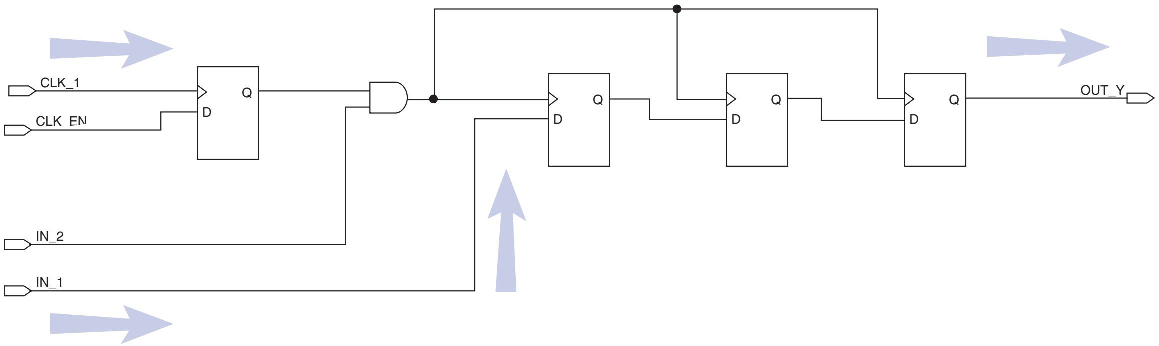


The Xilinx ISE software normally applies the timing constraints to the input of the DCM. PLL constraints are normally set in the Quartus II software using the MegaWizard Plug-In Manager for the altpll megafunction. Because the MegaWizard settings overwrite the PLL input timing constraints, the user must set the PLL timing constraint in the MegaWizard to the proper target frequency so that the place-and-route engine works to meet that frequency, and output from timing analyzer is correct.

Setup (t_{SU}) and Clock-to-Out (t_{CO}) Timing Analysis

Because the Quartus II software accounts for registered clock structures and the ISE software does not, setup (t_{SU}) and clock-to-out (t_{CO}) timing analysis results are also different between the two tools. If the input or output register has a registered-clock structure preceding it, the Quartus II software adds the microtiming parameter of the register to the external t_{SU} and t_{CO} timing measurement. The ISE software does not report t_{SU} or t_{CO} for this type of structure. For the example shown in Figure 6, the Quartus II software will report t_{CO} for reg4 through the microtiming delay of the gate and reg1. The ISE software will not report the t_{CO} value.

Figure 6. I/O Timing Issue Example



Transparent Latches

Registers are readily-available resources in FPGAs and they may be freely used in synchronous designs. Registers in FPGA designs help the designer to achieve a simpler design netlist and improved routability. Many ASIC designers, however, prefer the smaller die cost of latches. ASIC designs converted for use in FPGAs may contain a significant number of latches.

Latches can also appear in FPGA synthesis as a result of empty registers or incomplete `if-else` and `case` statements. The existence of empty registers or incomplete `if-else` and `case` in a design is poor coding practice but easily fixed.

Using the Xilinx ISE tool, the minimum period analysis may start at or end in latches because the ISE software treats latches as timing points similar to registers. The Quartus II software treats latches as logic and analyzes them as part of a look-up table (LUT) chain. Because of this difference, the register-to-register logic level in the timing analysis report is shorter as reported by the ISE software than it is as reported by Quartus II software when latches are involved.

In Figure 7, RegA and RegD are registers and LatchB and LatchC are transparent latches. The ISE software measures maximum frequency (f_{MAX}) for each path between RegA and LatchB, LatchB and LatchC and LatchC and RegD, with no combinatorial logic level between each path. The Quartus II software treats LatchB and LatchC as logic with combinatorial delays and measures f_{MAX} from RegA to RegD with 2 logic levels, causing f_{MAX} of the clock signal to appear reduced.

Figure 7. Latch Chain Example

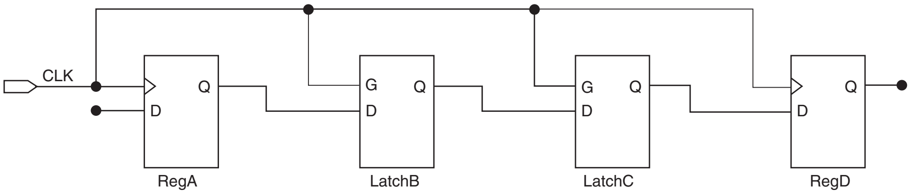


Table 1 summarizes the differences between Xilinx ISE and Altera Quartus II timing analysis.

Table 1. Differences in ISE and Quartus II Timing Analysis

Design Structure	Xilinx ISE software	Altera Quartus II Software
Clock Skew	Not analyzed by default. Option "-skew" needs to be added to TRCE timing analysis engine.	Always analyzed and cannot be turned off.
Registered Clock	Not analyzed. Output can be a separate clock domain if properly constrained.	Always analyzed as part of the clock skew but can be turned off or analyzed as a separate clock domain if constrained the same way as in the ISE software.
Gated Clock	Analyzed only when constrained.	All signals related to clock ports are assumed to be clock signals .
Combinational Loop	Cannot be analyzed. Warning reported.	Analyzed by default but can be cut off by user if desired.
Designs with DCM/PLL	Analyed if constraint assigned to input of DCM.	Constraint assigned in the PLL Megawizard.
Setup (t_{SU}) and Clock-to-Out (t_{CO})	Registered-clock preceding input or output registers not analyzed.	All worst-case structures analyzed.
Transparent Latches	Treated as timing point and included as starting or end point in f_{MAX} calculation.	Treated as combinatorial logic and added delay to the data path. Only measures register-to-register f_{MAX}.

How To Perform Equivalent Timing Analysis

To achieve a fair comparison between the Quartus II software and ISE software, adjustments must be made to make the tools analyze the same paths in a design because 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.

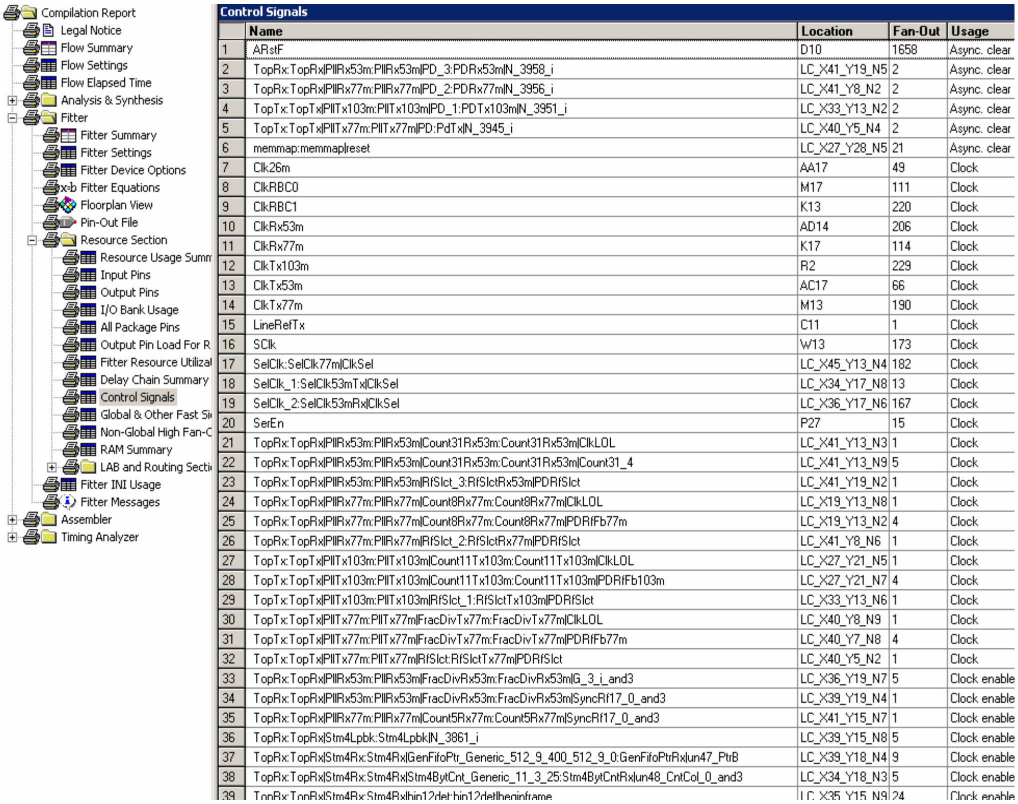
2. Because the Quartus II software reports all possible paths by default, extract the clock names from the fitter report file, as shown in Figure 8. Note that all signals that are used by registers as clock signals are listed in this table. Also, perform "Advanced" timing analysis in the ISE software (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. Once the clocks have been identified, apply 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 ISE report file. These clocks must be constrained with a default value to ensure that the Quartus II software does not automatically relate them.

Figure 8. Quartus II Fitter Report



Control Signals

	Name	Location	Fan-Out	Usage
1	ARstF	D10	1658	Async. clear
2	TopRx:TopRx|PllRx53m:PllRx53m|PD_3:PDRx53m|N_3958_i	LC_X41_Y19_N5	2	Async. clear
3	TopRx:TopRx|PllRx77m:PllRx77m|PD_2:PDRx77m|N_3956_i	LC_X41_Y8_N2	2	Async. clear
4	TopTx:TopTx|PllTx103m:PllTx103m|PD_1:PDTx103m|N_3951_i	LC_X33_Y13_N2	2	Async. clear
5	TopTx:TopTx|PllTx77m:PllTx77m|PD:PdTx|N_3945_i	LC_X40_Y5_N4	2	Async. clear
6	memmap:memmap|reset	LC_X27_Y28_N5	21	Async. clear
7	Clk26m	AA17	49	Clock
8	ClkRBC0	M17	111	Clock
9	ClkRBC1	K13	220	Clock
10	ClkRx53m	AD14	206	Clock
11	ClkRx77m	K17	114	Clock
12	ClkTx103m	R2	229	Clock
13	ClkTx53m	AC17	66	Clock
14	ClkTx77m	M13	190	Clock
15	LineRefTx	C11	1	Clock
16	SClk	W13	173	Clock
17	SelClk:SelClk77m|ClkSel	LC_X45_Y13_N4	182	Clock
18	SelClk_1:SelClk53mTx|ClkSel	LC_X34_Y17_N8	13	Clock
19	SelClk_2:SelClk53mRx|ClkSel	LC_X36_Y17_N6	167	Clock
20	SerEn	P27	15	Clock
21	TopRx:TopRx|PllRx53m:PllRx53m|Count31Rx53m:Count31Rx53m|ClkLOL	LC_X41_Y13_N3	1	Clock
22	TopRx:TopRx|PllRx53m:PllRx53m|Count31Rx53m:Count31Rx53m|Count31_4	LC_X41_Y13_N9	5	Clock
23	TopRx:TopRx|PllRx53m:PllRx53m|RfSlct_3:RfSlctRx53m|PDRfSlct	LC_X41_Y19_N2	1	Clock
24	TopRx:TopRx|PllRx77m:PllRx77m|Count8Rx77m:Count8Rx77m|ClkLOL	LC_X19_Y13_N8	1	Clock
25	TopRx:TopRx|PllRx77m:PllRx77m|Count8Rx77m:Count8Rx77m|PDRfFb77m	LC_X19_Y13_N2	4	Clock
26	TopRx:TopRx|PllRx77m:PllRx77m|RfSlct_2:RfSlctRx77m|PDRfSlct	LC_X41_Y8_N6	1	Clock
27	TopTx:TopTx|PllTx103m:PllTx103m|Count11Tx103m:Count11Tx103m|ClkLOL	LC_X27_Y21_N5	1	Clock
28	TopTx:TopTx|PllTx103m:PllTx103m|Count11Tx103m:Count11Tx103m|PDRfFb103m	LC_X27_Y21_N7	4	Clock
29	TopTx:TopTx|PllTx103m:PllTx103m|RfSlct_1:RfSlctTx103m|PDRfSlct	LC_X33_Y13_N6	1	Clock
30	TopTx:TopTx|PllTx77m:PllTx77m|FracDivTx77m:FracDivTx77m|ClkLOL	LC_X40_Y8_N9	1	Clock
31	TopTx:TopTx|PllTx77m:PllTx77m|FracDivTx77m:FracDivTx77m|PDRfFb77m	LC_X40_Y7_N8	4	Clock
32	TopTx:TopTx|PllTx77m:PllTx77m|RfSlct:RfSlctTx77m|PDRfSlct	LC_X40_Y5_N2	1	Clock
33	TopRx:TopRx|PllRx53m:PllRx53m|FracDivRx53m:FracDivRx53m|G_3_i_and3	LC_X36_Y19_N7	5	Clock enable
34	TopRx:TopRx|PllRx53m:PllRx53m|FracDivRx53m:FracDivRx53m|SyncRf17_0_and3	LC_X39_Y19_N4	1	Clock enable
35	TopRx:TopRx|PllRx77m:PllRx77m|Count5Rx77m:Count5Rx77m|SyncRf17_0_and3	LC_X41_Y15_N7	1	Clock enable
36	TopRx:TopRx|Stm4Lpbk:Stm4Lpbk|N_3861_i	LC_X39_Y15_N8	5	Clock enable
37	TopRx:TopRx|Stm4Rx:Stm4Rx|GenFifoPtr_Generic_512_9_400_512_9_0:GenFifoPtrRx|un47_PtrB	LC_X39_Y18_N4	9	Clock enable
38	TopRx:TopRx|Stm4Rx:Stm4Rx|Stm4BytCnt_Generic_11_3_25:Stm4BytCntRx|un48_CntCol_0_and3	LC_X34_Y18_N3	5	Clock enable
39	TopRx:TopRx|Stm4Rx:Stm4Rx|bin12det:bin12det|beginframe	LC_X35_Y15_N9	24	Clock enable

The following sections outline the steps needed to ensure that the tools analyze the same paths. These steps involve making constraints in the Quartus II software to match 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 Clock Skew Difference

Since the Quartus II software analyzes clock skew for both global and non-global clocks, it is important to always turn on the `–skew` option in the ISE software. This causes the ISE clock skew analysis to be equivalent to the Quartus II clock skew analysis.

Correcting Registered Clock Analysis Differences

The ISE software accounts for registered-clock structures by treating original and derived clocks as separate clock domains. In addition, constraints applied to the original clocks do not constrain paths using these registered clocks. The Quartus II software analyzes the relationships between the clocks by default, applying the f_{MAX} constraints to all registered versions of the clock, and analyzing the paths between them.

One of two approaches can be used to ensure equivalent timing analysis. One approach involves making additional constraints in the Quartus II software to break the relationships between clocks that the Quartus II software automatically infers. The other approach involves explicitly declaring relationships between clocks in the ISE software to match the default timing analysis behavior in the Quartus II software.

In the Quartus II software, use the Assignment Editor to create timing constraints for both the original and registered clock signals. For the example shown in Figure 2, apply the constraint to the clk pin and the output of reg1. Making these two clock settings breaks the relationship between the original clock and its derivative or registered version because each is treated as an independent clock. In the ISE software, use the Constraints Editor to assign timing constraints to both the original and registered clock signals to ensure analysis of the appropriate paths.

Correcting Gated Clock Analysis Differences

For gated clocks, only constrain the original clock, which is the input to the gate. For the gated-clock example shown in Figure 3, the constraint is applied to the input pin called `clk`. If that constraint is applied, both tools analyze the paths using both the original clock and its gated version as well as paths between the clocks.

To simplify the necessary changes, the user can constrain the original and gated clocks so that they are independent. This simplifies the procedure because the user does not need to understand the relationships between the original and derived clocks to make the appropriate settings. The same solution is sufficient for both registered and gated clocks. Because the ordering of constraints to the same paths is important when using the ISE software, the user must make the timing settings to the derived clock first.

Correcting Combinatorial Loop Analysis Differences

Since the ISE software does not analyze paths for combinatorial loop structures, these paths need to be explicitly cut in the Quartus II software to perform equivalent analysis. To cut these paths, first identify the combinatorial loop structures, then choose the paths in the Assignment Editor and choose “cut timing path” for each one. The Quartus II timing analyzer ignores these paths.

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. For ISE, assign a timing constraint to the input of the DCM. The ISE software applies this constraint to the output clocks and sets values according to the DCM clock frequency. In the Quartus II software, assign the input and output clock frequency parameters in the `altpll` MegaWizard, causing the Quartus II software to constrain each output clock frequency to match the settings made in the ISE software. By constraining the design in this way, both tools analyze the paths between the PLL/DCM generated clocks.

Correcting (t_{SU}) and Clock-to-Out (t_{CO}) Analysis Differences

Since the ISE software does not automatically determine the relationship between registered clocks and their original source, it does not calculate setup or clock-to-out time for any paths that use registered clock signals. To exclude the same paths in the Quartus II software, the solution is similar to the registered clock solution. Make the original and the derived clock independent by specifiying a false path between them in the Quartus II software.

Since gated clocks are handled in the same way for both tools, constraining setup and clock-to-out timing is all that is necessary.

Correcting Transparent Latch Treatment Differences

To correct transparent latch treatment differences, replace latches with registers. Registers are used in FPGA designs more often than latches because registers are more stable, more resistant to noise, and abundant in FPGA architectures. Replacing latches with registers in most cases does not change the functionality of the design.

To replace the latches with registers, here are the recommended actions:

1. If the latches appear because of empty register coding or incomplete if-else/case statements in the RTL design, complete the statements by assigning values to them.

2. If it is an ASIC design adapted for an FPGA has direct instantiations of latches, replace the latch libraries with registers.

Afterwards, recompile the design and re-run the timing analysis to obtain equivalent timing analysis between the ISE and Quartus II software packages.

Timing Analysis Correction Results

Below are some examples of equivalent timing analysis to illustrate the impact of timing analysis differences between the Quartus II software and the ISE software and how it performs after the corrections are applied to ensure equivalent timing comparison.

In each of these cases the customer has constrained all clock signals in the Xilinx ISE tool. For the Quartus II software, there are 2 compilations:

1. Default compilation, where the Quartus II software performs conservative worst-case analysis on all paths

2. 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.

Design 1

Table 2 shows Quartus II data for Design 1 after default compilation and after corrections are applied.

Table 2. Design 1 Timing Analysis Correction Example

Characteristic	Default Compilation	After Corrections Are Applied
Size	6,066 LEs	6,066 LEs
Clock 1 Maximum Frequency (f_{MAX})	63.75 MHz	91.32 MHz
Clock 2 Maximum Frequency (f_{MAX})	52.54 MHz	242.31 MHz

After the user configures the Quartus II software to perform timing analysis equivalent to the ISE software, both reported clock frequencies go up significantly. This design has many paths that go across derived clock domains and the Quartus II software analyzes them all by default, hence the lower f_{MAX}. The f_{MAX} measurement increases after the cross-clock paths are cut off from analysis. The critical path of clk2 after default compilation is 19.033 ns, of which 5.226 ns is clock skew. This skew is a result of a clock structure similar to that shown in Figure 2. After the clock relationships are broken, the critical path has two registers directly driven by clock2 with almost no clock skew.

Design 2

By default, the Quartus II software accounts for all cross-domain analysis, assumes worst-case scenarios for gated-clock signals, and analyzes all paths that exist in between. This design also contains registered clocks. After the user removes cross domain analysis and assigns an internal node clock frequency to mimic ISE timing analysis, the clock frequencies go up and 2 internal clock signals are introduced. Clocks 1, 4, and 8 are not reported because they are inputs to the same logic that produes Internal Clocks 1 and 2. Internal Clock 2 is generated by Clock 6 gated with Clock 5. Clock 6 is the pin clock and clock 5 is no longer reported as a clock because its analysis is done using internal clock 2.

Table 3. Design 2 Timing Analysis Fix Example

Characteristic	Default Compilation	After Fixes Are Applied
Size	16,362 LEs	16,362 LEs
Clock 1 Maximum Frequency (f_{MAX})	133.12 MHz	-
Clock 2 Maximum Frequency (f_{MAX})	130.23 MHz	242.31 MHz
Clock 3 Maximum Frequency (f_{MAX})	181.55 MHz	266.24 MHz
Clock 4 Maximum Frequency (f_{MAX})	133.12 MHz	-
Clock 5 Maximum Frequency (f_{MAX})	161.84 MHz	-
Clock 6 Maximum Frequency (f_{MAX})	84.79 MHz	156.25 MHz
Clock 7 Maximum Frequency (f_{MAX})	161.84 MHz	422.12 MHz
Clock 8 Maximum Frequency (f_{MAX})	133.12 MHz	-
Internal Clock 1 Frequency (f_{MAX})	-	132.8 MHz
Internal Clock 2 Frequency (f_{MAX})	-	415.11
Number of Setup/Hold Time Violations	15	0

Design 3

Design 3 contains registered clocks. Breaking the clock relationships eliminates reported clock skew. The netlist holds gated clocks and paths crossing between clock domains. After the user assigns individual clock signals, the performance of each clock signal increases, making fair performance comparison with ISE timing analysis possible (Table 4).

Table 4. Design 3 Timing Analysis Correction Example

Characteristic	Default Compilation	After Corrections Are Applied
Size	44,811 LEs	44,811LEs
Clock 1 Maximum Frequency (f_{MAX})	49.45 MHz	51.27 MHz
Clock 2 Maximum Frequency (f_{MAX})	90.95 MHz	92.91 MHz
Clock 3 Maximum Frequency (f_{MAX})	87.32 MHz	89.94 MHz

Achieving Best Results Through Quartus II Automation

The Quartus II software version 3.0 includes a feature for automating multiple compilations using different settings called the Quartus II Design Space Explorer (DSE). DSE is a Tcl/Tk utility that allows users to run multiple compilations with different seeds, physical synthesis settings and other optimization options. It is ideal for running overnight benchmark tests to potentially obtain higher performance results. This utility is recommended for benchmarking because it requires very little user effort and in general produces higher performance results.

DSE produces an average of 17% performance improvement over default push-button compilation. DSE automates all the iterations, option permutations and archive of all intermediate results and reports the best performance to the user. For more information, refer to *AN 198: Timing Closure With The Quartus II Software*.

Synthesis EDA Tools Estimation

Most EDA synthesis tools today used in FPGA design provide area and performance estimation capabilities to allow users to quickly verify their design resources without having to compile in the FPGA place and route tools, which normally take longer to complete. Although Altera worrks 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 in the place and route tool (either the Quartus II software or the ISE software) because:

1. EDA tools only provide estimates, not actual data. Actual performance and area results can vary significantly after remapping and place-and-route. The register packing performed by the Quartus II software, for example, reduces area utilization by 11%. The ISE software also produces actual performance and device utilization data that differs from that provided by EDA synthesis tools.

2. The accuracy of the estimates also depends on the existence of black boxes (containing Altera Megafunctions or Xilinx CoreGen cores). EDA synthesis tools do not synthesize the black-boxed functions and cannot perform timing estimates on them.

3. Xilinx and Altera resource estimations are not equivalent. Virtex-II, Virtex-II Pro, and Spartan-3 have slices as a unit for logic blocks. Stratix, Stratix GX and Cyclone have logic elements (LEs), which have different configurations. These two configurations are not equivalent, and the user must compile the design in either the Quartus II or ISE software to determine the actual performance of the design on the targeted device.

4. 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.

For more information, see *TB 84: Differences in Logic Utilization between Quartus II & Synplify Report Files*.

Summary

Each FPGA software tool has its own set of parameters and environment determined by the FPGA vendor to be optimum for their products. The Altera Quartus II software and the 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. Complex structures, such as gated clocks, registered clocks, combinational loops, and DLL/PLL usage are analyzed in the ISE software from the way they are analyzed in the Quartus II software.

In order to perform fair cross-vendor software benchmarking, some constraints and settings adjustments need to be made when applicable. The Quartus II timing analyzer performs complete and 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, reducing the total net delay. If Quartus II constraints are set to mimic those of the ISE software, performance improvement is seen for the Quartus II software.

The 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 feature makes it easy to perform multiple compilations with different settings to improve performance beyond default values.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com