Benchmarking Using the Quartus University Interface Program (QUIP)

Version 1.1

February 6, 2008

Table of Contents:

1. Overview

This document is intended for developers working with the Quartus University Interface Program (QUIP). It describes the best way to extract various quality metrics from the Quartus II software, and how to ensure that Quartus II is attempting to optimize the metric(s) in which you are interested. Only by (1) extracting the appropriate quality metric and (2) ensuring that the settings used to run Quartus II have made this metric an optimization goal can you meaningfully compare CAD algorithms or flows.

The original version of this document was written for version 5.0 of the Quartus II software. It was updated for version 7.2 of the Quartus II software, and will apply to other versions with minor changes. The document describes how to make settings using the UI. If you prefer to use Tcl scripts or directly edit the <project>.qsf file, simply compare the <project>.qsf file before and after making a settings change in the GUI to see the appropriate Tcl command or qsf file setting.

We have only described details on the most common comparisons researchers make when evaluating CAD tools. If you have any questions about how to make a fair comparison for these or other areas, feel free to send an email to quip@altera.com for advice.

2. Recommendations for specific metrics

Comparisons of different CAD flows usually involve running several designs through each and then using the value of certain metrics for each design to determine the advantage of one flow over another. The most common of these metrics are discussed below, along with notes on their use so as to create fair comparisons.

2.1 Performance

The following recommendations apply when you want to compare flows for circuit performance using the Quartus II software.

1) For performance metrics to be meaningful, the Quartus II software must be told to optimize the appropriate timing constraints. Quartus II will not optimize the timing of any circuitry without a timing constraint. In detail, one should:

   1.1) Set impossible timing constraints.

   When optimizing for Fmax (the most common performance metric), the easiest way to ensure this is to set an impossibly high global Fmax requirement. This requirement will apply to all clocks in the design. For other metrics (eg. Tsu, Tco) there are similar constraints that can be set. An impossible constraint should be used because otherwise, the Quartus II software will stop optimizing the metric once a design's performance exceeds it, resulting in slower performance numbers than the Quartus II software can actually achieve.

   To set an impossible global Fmax requirement:

   1.1.1) There are several ways to set a requirement of 1000 MHz ("i.e., one thousand Mhz") depending on the timing analyzer tool that you use. In

Assignments->Settings->Timing Analysis Settings you can either choose to “Use TimeQuest Timing Analyzer during compilation” or to “Use the Classic Timing Analyzer during compilation” by selecting the corresponding option. We suggest that you use TimeQuest for MAX II and all Stratix and Cyclone device families since it performs a rise/fall analysis that requires less guard banding. For all older device families such as APEX20KE you have to use the Classic Timing Analyzer. If you choose to use TimeQuest, you need to specify your constraints in an .sdc file. This .sdc file can contain a single line such as

derive clocks –period “1000MHz”

and needs to be added to your Quartus Project in Assignments->Settings->Timing Analysis Settings->TimeQuest Timing Analyzer. If you choose to use the Classic Timing Analyzer you can create such a constraint by setting Assignments->Settings->Timing Analysis Settings->Classic Timing Analyzer Settings->Default required fmax to 1000MHz.

1.2) Ensure fitter timing optimization is turned on (this is the default).

You can check this setting by making sure that Assignments->Settings->Fitter Settings->Fitter effort is set to “Standard Fit”.

2) When comparing designs that contain multiple clocks, the Quartus II software will optimize all the clocks if you use a global Fmax timing assignment. For such designs, comparing just one of the clocks (often the slowest) is not the best practice, because that is not the only clock being optimized.

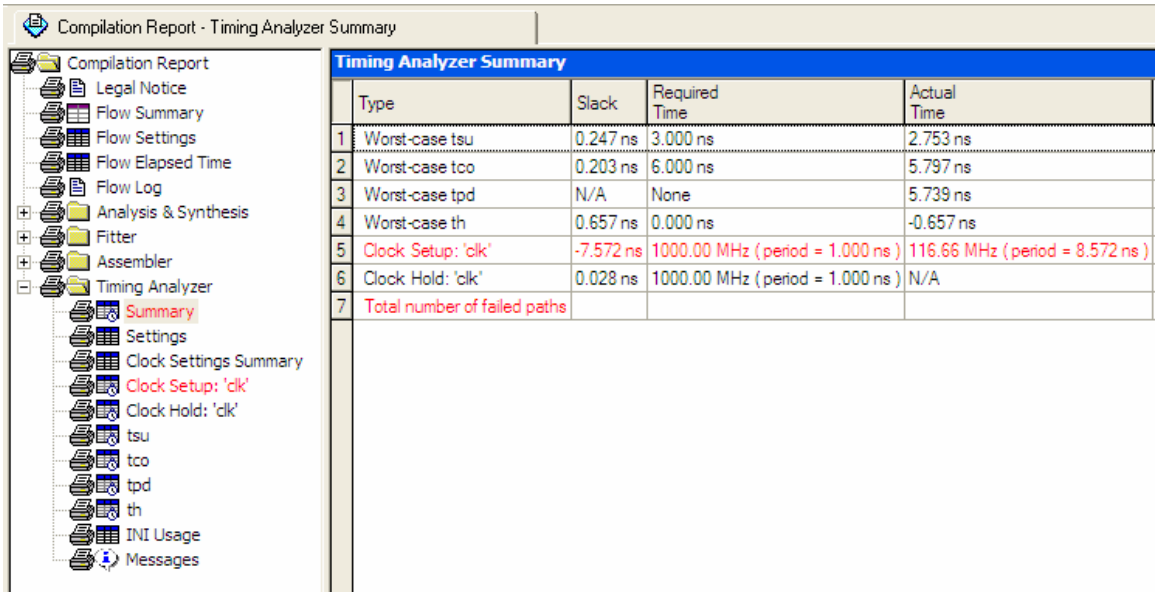
To avoid this, you can take the average Fmax of all the clocks to reflect the performance of such designs. A geometric average fairly (evenly) weights all the clocks, regardless of speed, so it is preferable to an arithmetic average.

Alternatively, you can set a timing constraint only on the slowest clock in the design, and measure the speed of only that clock if that is desired.

3) When comparing synthesis CAD flows where circuit area is not considered, the settings that make the Quartus II synthesis algorithms focus on timing optimization, even at the expense of area, should be set.

3.1) Set Assignments->Settings->Analysis & Synthesis Settings->Optimization Technique to “Speed”.

4) Depending on the timing analysis tool that is used you can get information on clock performance from the TimeQuest Timing Analyzer|Clocks section of the compilation report or from the Timing Analyzer|Summary section of the Compilation Report. As already mentioned above, we suggest that you use TimeQuest for MAX II and all Stratix and Cyclone device families since it performs a rise/fall analysis that requires less guard banding. For all older device families such as APEX20KE you have to use the Classic Timing Analyzer. **Figure 1** shows an example of what this summary looks like for the Classic Timing Analyzer.



	Type	Slack	Required Time	Actual Time
1	Worst-case tsu	0.247 ns	3.000 ns	2.753 ns
2	Worst-case tco	0.203 ns	6.000 ns	5.797 ns
3	Worst-case tpd	N/A	None	5.739 ns
4	Worst-case th	0.657 ns	0.000 ns	-0.657 ns
5	Clock Setup: 'clk'	-7.572 ns	1000.00 MHz (period = 1.000 ns)	116.66 MHz (period = 8.572 ns)
6	Clock Hold: 'clk'	0.028 ns	1000.00 MHz (period = 1.000 ns)	N/A
7	Total number of failed paths			

Figure 1 - Fmax Results in Classic Timing Analyzer Report

2.2 Density (Area)

The following recommendations apply when you want to compare flows for circuit density or area using the Quartus II software.

1) When using the Stratix, Cyclone, Cyclone II or Cyclone III families, the LE count reported by either Analysis & Synthesis or Fitting reflects the area of a design. For Stratix II, Stratix II GX, and Stratix III, the ALUT count should be used. Since register packing occurs after fitting, the most accurate area numbers are available only after fitting.

   The LE count and ALUT count values are given in the Fitter|Resource Section|Resource Usage Summary section of the Compilation Report. An example for Stratix II is shown in **Figure 2**, and the total logic used by the design is given by the circled “ALUTs Used” entry.

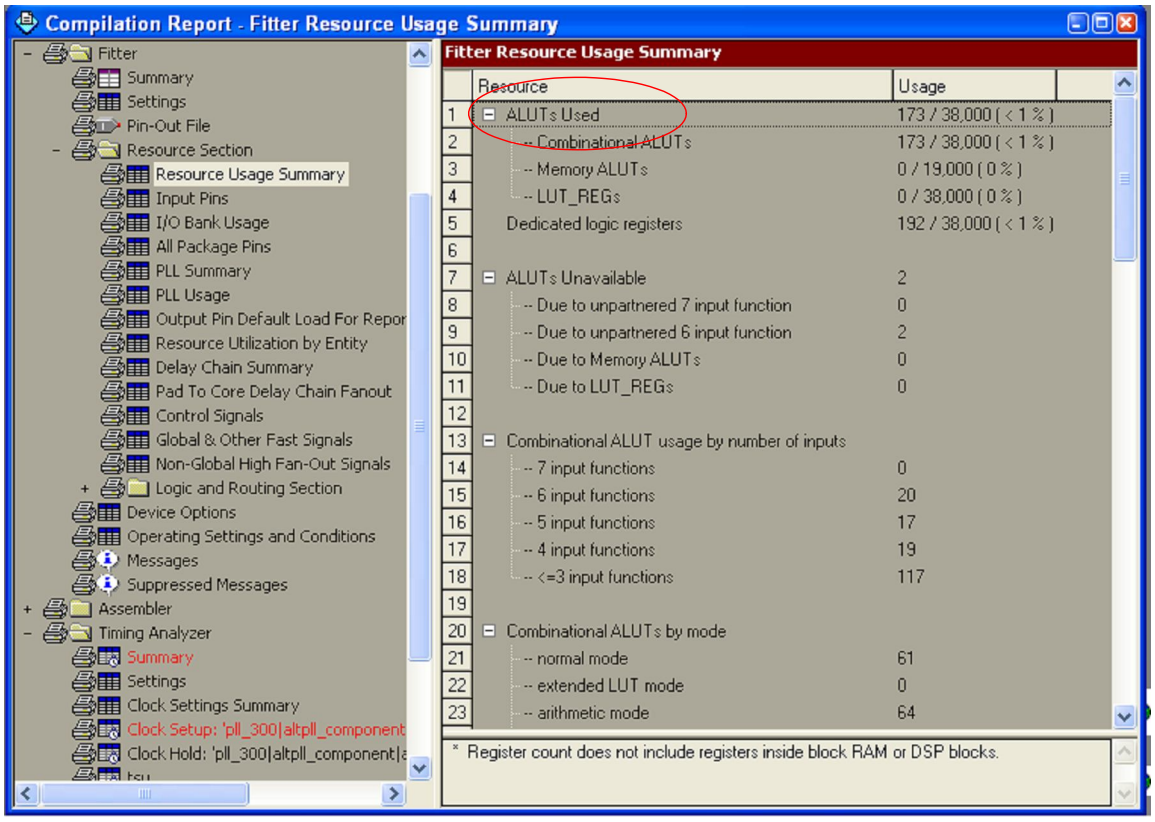


Figure 2 - ALUT count in Fitter report

2) When comparing flows where timing performance is not included in the comparison, the settings that make the Quartus II software focus on area optimization should be set. For a complete list, refer to the Resource Optimization Advisor in Quartus. Some of the important settings include:

   2.1) Tell synthesis to focus on area, even at the expense of speed, by setting Assignments->Settings->Analysis & Synthesis Settings->Optimization Technique to “Area”.

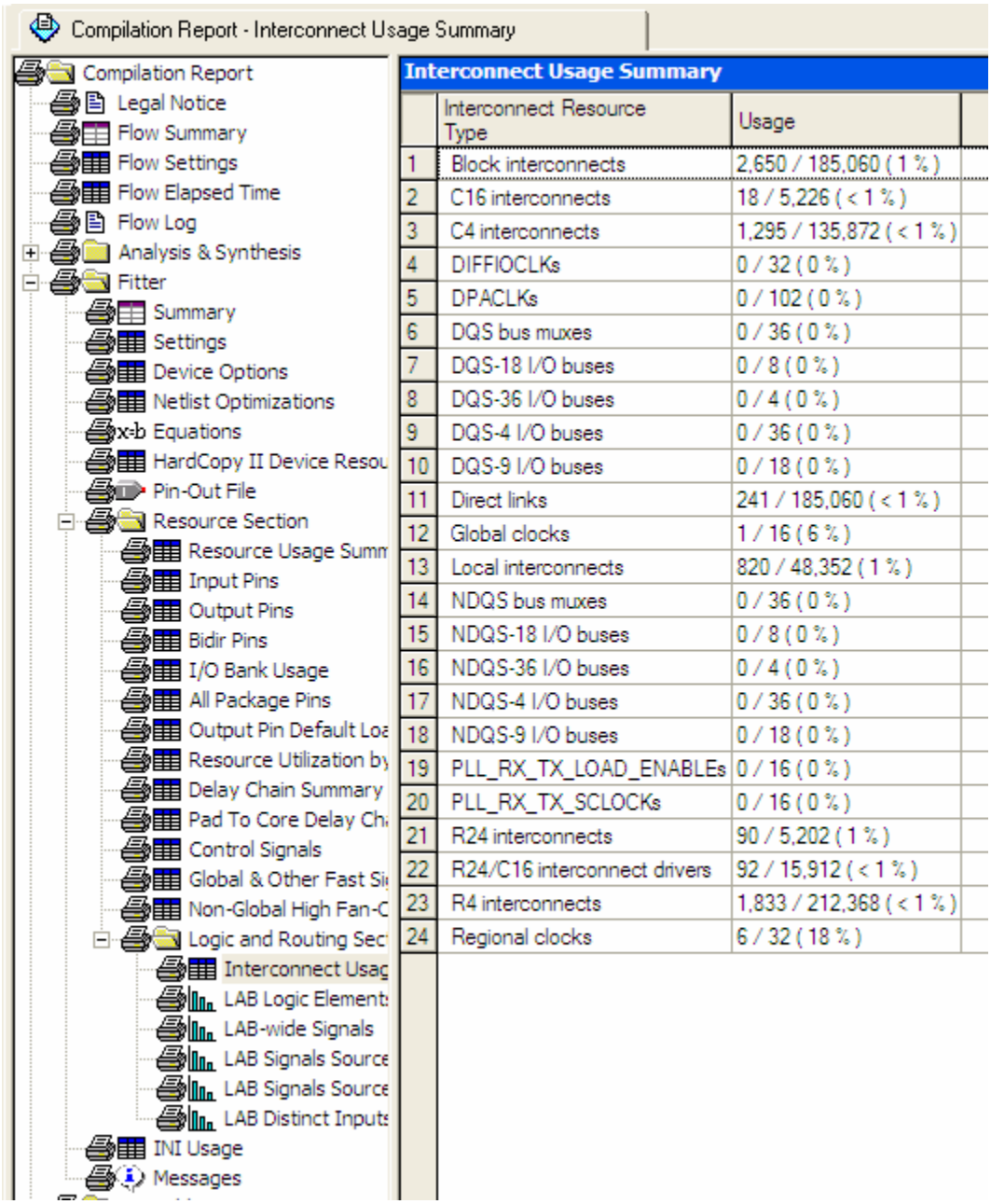
   2.2) Turn on aggressive register packing. This can be done by setting Assignments->Settings->Fitter Settings->More Settings->Auto Packed Registers – Stratix II/II GX/III Cyclone II/III Arria GX to Minimize Area with Chains.

2.3 Routed Wirelength

The following recommendations apply when you want to compare flows for wirelength using the Quartus II software.

1) When measuring wirelength, it is best to use the final routed values.

2) The routed wirelength of a design can be computed by referring to the Fitter Report|Resource Section|Logic and Routing Section|Interconnect Usage Summary section of the compilation report file, either in the GUI or in the <project>.fit.rpt file. **Figure 3** shows an example of what this section of the report looks like.



Interconnect Usage Summary

	Interconnect Resource Type	Usage
1	Block interconnects	2,650 / 185,060 (1 %)
2	C16 interconnects	18 / 5,226 (< 1 %)
3	C4 interconnects	1,295 / 135,872 (< 1 %)
4	DIFFIOCLKs	0 / 32 (0 %)
5	DPACLKs	0 / 102 (0 %)
6	DQS bus muxes	0 / 36 (0 %)
7	DQS-18 I/O buses	0 / 8 (0 %)
8	DQS-36 I/O buses	0 / 4 (0 %)
9	DQS-4 I/O buses	0 / 36 (0 %)
10	DQS-9 I/O buses	0 / 18 (0 %)
11	Direct links	241 / 185,060 (< 1 %)
12	Global clocks	1 / 16 (6 %)
13	Local interconnects	820 / 48,352 (1 %)
14	NDQS bus muxes	0 / 36 (0 %)
15	NDQS-18 I/O buses	0 / 8 (0 %)
16	NDQS-36 I/O buses	0 / 4 (0 %)
17	NDQS-4 I/O buses	0 / 36 (0 %)
18	NDQS-9 I/O buses	0 / 18 (0 %)
19	PLL_RX_TX_LOAD_ENABLEs	0 / 16 (0 %)
20	PLL_RX_TX_SCLOCKs	0 / 16 (0 %)
21	R24 interconnects	90 / 5,202 (1 %)
22	R24/C16 interconnect drivers	92 / 15,912 (< 1 %)
23	R4 interconnects	1,833 / 212,368 (< 1 %)
24	Regional clocks	6 / 32 (18 %)

Figure 3 - Routing Summary in Fitter Report

3) To ensure that Quartus is optimizing wirelength to its full ability, set the fitter effort level to standard. This can be done by setting Assignments->Settings->Fitter Settings->Fitter Effort to “Standard Fit”.

2.4 Compile Time

The following recommendations apply when you want to compare flows for compile time using the Quartus II software.

1) If comparing the compile time of Quartus without comparing other metrics (such as performance or wirelength), set the fitter effort level to auto. This makes sure that

the Quartus II software is only optimizing as much as is required to achieve a fit, thereby minimizing compile time. It is done by setting Assignments->Settings->Fitter Settings->Fitter Effort to “Auto Fit”.