

PLUNIFY

FPGAACCEL CLIENT

Version 1.0

USER GUIDE



Notices

© 2009-2011 Plunify. This document is provided for informational purposes only. It represents Plunify's current product offerings as of the date of issue of this document, which is subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of Plunify's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from Plunify, its affiliates, suppliers or licensors. The responsibilities and liabilities of Plunify to its customers are controlled by Plunify's agreements, and this document is not part of, nor does it modify, any agreement between Plunify and its customers.

CONTENTS

OVERVIEW	4
INTENDED AUDIENCE.....	4
INTRODUCTION	4
ACRONYMS USED IN THIS GUIDE	4
REFERENCE MATERIAL.....	5
PRODUCT OVERVIEW	5
REQUIREMENTS	6
INSTALLATION AND SETUP	8
INSTALLING FPGAACCEL.....	8
PLUNIFY ACCOUNT SETUP.....	9
COMPILE DESIGN	10
REQUIRED ARGUMENTS.....	10
COMMAND-LINE OPTIONS.....	11
RUN EXAMPLES.....	12
JOB OUTPUT.....	14
VIEW RESULTS.....	15
DOWNLOAD RESULTS	15
USE FPGAACCEL WEB.....	15
SUPPORT	21
HOW TO GET SUPPORT	21

Intended Audience

This guide is designed to assist hardware designers in using Plunify's FPGAAccel™ Client product.

Introduction

The Plunify FPGAAccel™ Client is a desktop software tool that enables secure, efficient offloading of FPGA design tasks to Plunify's cloud-based FPGA design platform. Used for speeding up compute-intensive tasks like synthesis and place-and-route, the FPGAAccel Client consists of Tcl scripts and a Java application, which are invoked from the user's existing software tools. This Client handles user authentication to Plunify's managed cloud, encryption of design files and secure transmission of data.

Acronyms used in this guide

Table 1. Acronyms explained

Terms/Abbreviation	Explanation
API	Application Programming Interface
GUI	Graphical User Interface
JDK	Java Development Kit
JRE	Java 2 Runtime Environment
FPGAAccel	Product to accelerate FPGA design
EDAextend	Cloud Platform that extends EDA capabilities
AES	Advanced Encryption Standard
SSL	Secure Sockets Layer
Tcl	Tool Command Language
FPGA	Field Programmable Gate Array

Reference Material

Quartus™ II Software

Quartus II refers to a software suite of FPGA design tools produced by Altera® Corporation. Please refer to the Altera website on how to obtain Quartus II software.

<http://www.altera.com/products/software/products/quartus2>

Quartus II Documentation and User Guides

Please refer to the Altera website for all related documentation and resources.

<http://www.altera.com/support/software/sof-index.html>

Product Overview

Plunify provides an online platform, **EDAxtend™**, that harnesses the power of cloud computing and customizes it specifically for chip design workflows. EDAxtend enables design companies to access scalable computing resources and perform compute-intensive processes like simulation, regression testing and design exploration on-demand. The optimized timing closure flow maximizes parallel execution of design iterations to enable users to perform tasks much faster than before. The use of cloud computing reduces IT overheads and maintenance while providing a secure and easy-to-use computing environment with no disruption to current workflows.

BENEFITS

For Tool/IP Vendors

- **Enablement:** Bring cloud capabilities for existing and next-generation software tools with minimal time and effort.
- **Higher Customer Adoption:** Allow users access to tools at any time, from anywhere in a tested and secured environment.
- **Limits Software Piracy:** Convert users into customers, streamline release and patch activities and provide features that do not require end-user knowledge of the physical location and configuration of the system that delivers the services.

For End Users

- **Turnkey Solution:** Focus on designs and not making the software tools work. Software setup, server provisioning, updates and patches are done automatically.
- **Parallelize Design Flows:** Automation of what-if analysis using different design configuration, constraints and design optimization, allowing designers to obtain the most optimized designs in a faster period of time.
- **Mobility:** Access design tools at any time, from anywhere in a tested and secured environment.
- **Lower Initial Costs:** Reduce up-front investment on IT infrastructure overheads by making use of computing resources only when needed on a pay-as-you-go basis.
- **Ease of use:** Client scripts integrate directly with existing software tools.

Requirements

The components listed below are needed to run the FPGAAccel Client.

Table 2. FPGAAccel Client Requirements

Operating System
Windows® (32-, 64-bit):
Windows XP SP3
Windows Vista Professional SP1
Windows 7 Professional
Linux (32-, 64-bit):
Red Hat Enterprise 4.0, 5.0
SUSE Linux Enterprise 11
CentOS 4.0, 5.0
Ubuntu 8.04, 9.04, 10.04

FPGA Design Software

Altera®Quartus™ II ver. 9.1 onwards

Java Runtime Environment

JRE Version 1.5, latest version preferred.

Note: Quartus II tools come pre-installed with JRE. You can configure the cfg files to point to this installation.

32 bit :

- Windows:
<Quartus Installation Folder>/quartus/bin/jre/bin
- Linux
<Quartus Installation Folder>/quartus/linux/jre/bin

64 bit :

- Windows:
<Quartus Installation Folder>/quartus/bin64/jre64/bin
- Linux
<Quartus Installation Folder>/quartus/linux64/jre64/bin

INSTALLATION AND SETUP

The FPGAAccel Client is available for Windows and Linux environments and can be downloaded from the Plunify website at <http://www.plunify.com/>

EXAMPLE SETUP:

- An appropriate Windows or Linux operating system with Internet Connection and configuration of the minimum hardware and software requirements.
- Sign-up for a Plunify account at <http://www.plunify.com/en/register.php> After registration, an activation e-mail will be sent to the user. Upon successful activation, a user will be able to launch FPGA design and verification tasks via FPGAAccel.

Installing FPGAAccel

1. Download the installer from www.plunify.com/en/portal.php

Windows:

- Run the installer and follow the onscreen instructions. The installation process is expected to take less than two minutes, assuming all prerequisites are in place. A "readme.txt" document will pop up at the end.

Linux

- Extract the contents of the *fpgaaccel.v0001.tar.gz* file into a target directory. e.g. /usr/local/bin/Plunify
- Edit or Create file *plunify.ini* to specify your FPGAAccel and Java Install directory as shown in [Figure 1](#).

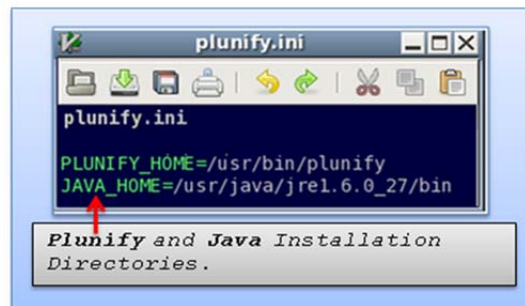


Figure 1. Tool Setup Initialization file for Linux

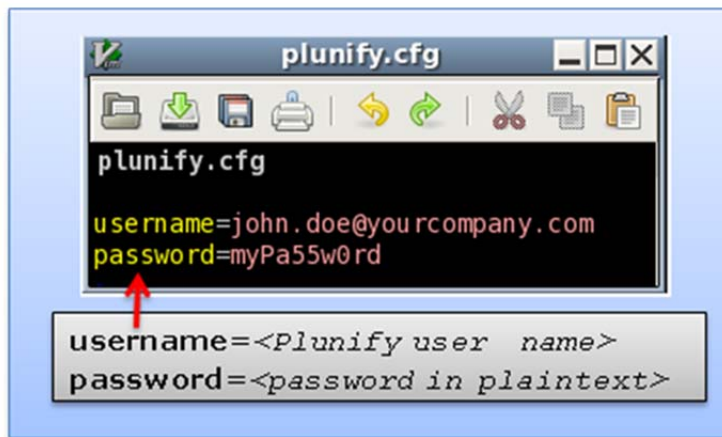
Plunify Account Setup

Create/Modify the Plunify configuration file **plunify.cfg** to add your Plunify user name and password as shown in [Figure 2](#). This file is to authenticate you to FPGAAccel. Default location for this file:

Windows: C:\Plunify\plunify.cfg

Linux: /usr/local/bin/plunify/plunify.cfg

Figure 2. Sample configuration file with user name and password



Note: Please use strong passwords for your account and to keep your configuration file in a secure location. If you believe your account has been compromised, change your password immediately via the Plunify website.

COMPILE DESIGN

Using the Quartus II command-line executable (*quartus_sh* or *quartus_sh.exe*) and FPGAAccel, offload each design to the Plunify cloud with a script command.

Each submitted design is termed a “job”.

The *cloudcompile.tcl* script used for this operation can be found at <INSTALLDIR>/tcl/quartus

Default installation directory locations:

- **Windows**
C:\Plunify\tcl\quartus\cloudcompile.tcl
- **Linux**
/usr/local/bin/plunify/tcl/quartus/cloudcompile.tcl

To submit a design, execute the following command from within a Quartus II project directory:

- **Windows**
quartus.exe -t <INSTALLDIR>\tcl\quartus\cloudcompile.tcl <REQUIRED ARGUMENTS> [OPTIONS]
- **Linux**
quartus_sh -t <INSTALLDIR>/tcl/quartus/cloudcompile.tcl <REQUIRED ARGUMENTS> [OPTIONS]

Required Arguments

Table 3. Mandatory Inputs for Cloudcompile.tcl

Name	Description
-project <project name>	Name of project to work on

Command-line options

Table 1. Command-line options

Options	Description
<p>-op <operation></p>	<p>Operation to perform:</p> <ul style="list-style-type: none"> • compile - synthesis, place-and-route, timing analysis, power estimation, programming file generation. • closure - compile multiple revisions/iterations in parallel to achieve timing closure. • download - retrieve results for a particular job <p>Default: compile</p>
<p>-cfg <file></p>	<p>Custom FPGAAccel Client configuration file</p> <p>Default: <INSTALLDIR>/<i>plunify.cfg</i></p>
<p>-closuretype <timing closure approach></p>	<p>Approach to generating timing parameters:</p> <ul style="list-style-type: none"> • default - use Plunify's custom algorithms [1] • dse – use Quartus II Design Explorer <p>Applicable only when -op is set to closure</p> <p>Default: Plunify's custom algorithms</p>
<p>-goal <optimization goal></p>	<p>Optimization goal options for timing closure:</p> <ul style="list-style-type: none"> • speed • area • power <p>Applicable only when -op is set to closure</p> <p>Default: speed</p>
<p>-ini <file></p>	<p>Custom FPGAAccel Client initialization file</p> <p>Default: <INSTALLDIR>/<i>plunify.ini</i></p>

-jobid <job identifier>	Job identifier as given in job completion notification email Required if -op is set to download
-numservers <integer>	Number of servers to use Default: Number of revisions
-revision <revision name>	Specific revision to work on Applicable only when -op is set to compile Default: Current revision
-seeds <comma-, space-delimited or hyphenated ranges of integers>	Random seeds to sweep as part of the timing closure process used by Quartus II. Applicable only when -op is set to closure and -closuretype is dse. Default: "2 3 4 5 6"
-src	Use full RTL for compilation Default: Synthesizes project locally and send binary netlist

Run Examples

Here are examples that show how to execute various operations:

Notes:

- All commands are entered from within the Quartus II project directory where the project .QPF and .QSF files are located.
- A Quartus II project named **foo** is used in the following examples.
- The location of the cloudcompile.tcl script is stored in a shell variable called **\$dir**
- Windows users use **quartus_sh.exe** instead of **quartus_sh**

1. Compile project

Compile on Plunify Cloud Platform

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op compile -src
```

2. Compile after synthesis

Runs synthesis on local machine first but runs place-and-route, static timing analysis, estimate power consumption, generate programming file on Plunify cloud platform

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op compile
```

3. Compile a revision called *bar* for a project called *foo*

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -revision bar -op compile -src
```

4. Compile with user-specified configuration and initialization files

To support different users

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -cfg ~/myPlunify.cfg -ini ~/myPlunify.ini -op compile -src
```

5. Attempt timing closure using Plunify's timing algorithms ^[1]

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op closure -src
```

6. Attempt timing closure using Quartus II Design Explorer

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op closure -src -closuretype dse
```

7. Compile all available project revisions

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op closure -src -allrevs
```

8. Downloading results

<Job ID> is obtained from job output window or from the job completion notification email

```
quartus_sh -t $dir/cloudcompile.tcl -project foo -op download -jobid  
<JobID>
```

Job Output

If a design is successfully sent to the cloud, the following output can be seen below in [Figure 3](#). It shows, **2126** is the “Job ID” that uniquely identifies a design that is run in the cloud.

This “Job ID” can be used subsequently to download results (See Section 8: View Results).

Figure 3. Design submission output with Job ID

```
Info: Archiving project or1200_top as or1200_top.qar ...  
Info: done.  
Info: Preparing or1200_top revision or1200_top for processing...  
Info: done.  
Info: Sending job request...  
Upload | 2011.11.04 11:02:50 | Log file created at ./  
Upload | 2011.11.04 11:02:50 | Plunify Client Mode  
Upload | 2011.11.04 11:02:57 | --- File plunify.job submitted to Plunify.  
Upload | 2011.11.04 11:02:57 | Pre-Uploading files to Plunify - Step 1: COMPLETED..  
Upload | 2011.11.04 11:02:57 | Pre-Uploading files to Plunify - Step 2: COMPLETED..  
Upload | 2011.11.04 11:03:01 | Pre-Uploading files to Plunify - Step 3: COMPLETED..  
Upload | 2011.11.04 11:03:01 | Now Ready to Upload  
Upload | 2011.11.04 11:03:01 | File encrypting: D:/Plunify/FPGAMiner/or1200_top.qar  
Upload | 2011.11.04 11:03:32 | File has been uploaded to: /or1200_top/or1200_top.qar  
Upload | 2011.11.04 11:03:32 | Upload COMPLETED!  
Upload | 2011.11.04 11:03:33 | --- Job Id: 2126 submitted to Plunify.  
Upload | 2011.11.04 11:03:33 | Job submitted with return code = 1  
Upload | 2011.11.04 11:03:33 | Job submission COMPLETED!  
Info: ... done.
```

VIEW RESULTS

When the job has completed, notification is sent to the user's registered email address with a "Job ID".

There are two ways to view your results, either download all your results or view them online.

Download Results

To download generated binaries and report files, first obtain your "Job ID" from the job submission output message shown in [Figure 3](#) or from the job completion email. An example is shown below.

Assumptions:

- Project Name: *foo*
- Job ID: *2065*

Run the following command from within the Quartus II project directory:

```
quartus_sh -t /usr/local/bin/plunify/cloudcompile.tcl -
project foo -op download -jobid 2065
```

All generated files will be downloaded into a new folder called, "2065" in the same directory.

Use FPGAAccel Web

FPGAAccel Web is an online interface where users can view and analyze collated results in a graphical manner. It is available to all registered users. To view your results, first, log onto <http://www.plunify.com/en/login.php> and click the Results tab ([Figure 4](#)).

Figure 4. Results panel in FPGAAccel web interface



The Results tab provides an easy way to see if a job has completed and which files were generated as a result. This section is especially helpful when analyzing and comparing multiple design iterations after compiling several revisions or attempting timing closure. [Figure 5](#) below shows the initial Results tab view. Three sub-tabs are available:

- **Complete** - Shows jobs that have already completed.
- **Running** - Shows jobs that are currently running.
- **Fail/Error** - Shows jobs that failed to complete or encountered some errors.

Under **Complete** tab, the *Type* column indicates the type of job. These include:

- **VRF** - Verification. Normal design compilation
- **CLO** - Closure. Achieve timing closure.
- **SIM** - Simulation. Functional simulation.

Figure 5: Main Results view

The screenshot shows the 'Results' interface with three tabs: 'Complete', 'Running', and 'Fail / Error'. The 'Complete' tab is active, displaying a table with the following data:

JobID	Job Source	Job Status	Type	Ended On	Duration	Results
J2074	Home/or1200_top/	Completed	CLO	20 Oct 2011 6:04 pm	1hr 50m	View
J2072	Home/or1200_top/	Completed	CLO	20 Oct 2011 4:44 pm	43m 40s	View

“VRF” JOB REPORTS

Click on the “**View**” hyperlink on a completed VRF job as shown above in [Figure 5](#).

A new sub-page, as shown in [Figure 6](#), will show the detailed summary of the design. [Figure 6](#) shows the initial view where the target device, device vendor and Pass/Fail status for each compilation stage are displayed. Each stage is individually tracked to aid debugging.

Figure 6. Initial Results Summary View

The screenshot shows the 'Initial Results Summary View' with a navigation bar containing: Results, Assessment, Summary, Logic Utilization, IO Pin Utilization, Max Frequency, Total Power, and Publish. The main content area displays the following information:

Device	Vendor	Synthesis	Implementation	Prog File
EP1AGX20CF780C6	Altera			

Assessment and Summary

These tabs present overviews of device result statistics across multiple jobs or a single job that targeted multiple devices. For instance, to see how different design parameters might affect a design, results are ranked according to various utilization ranges (Figures 7A, 7B).

Figure 7A. Logic Utilization below 45% and above 65%

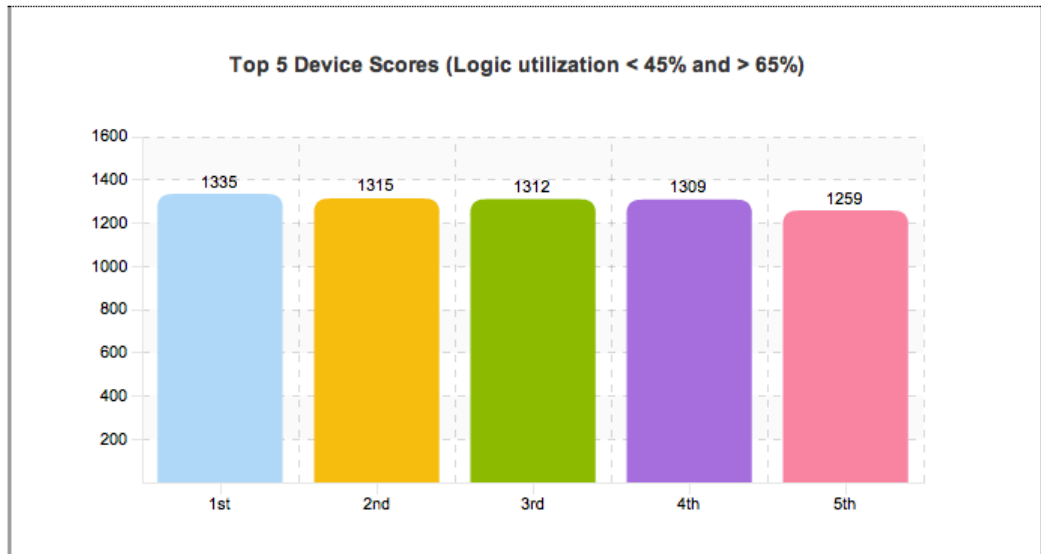
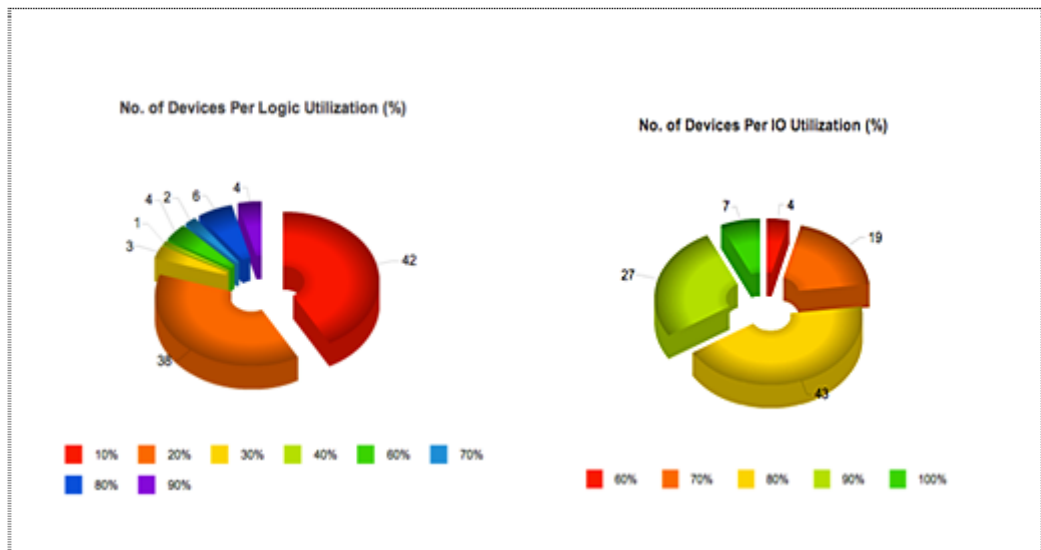


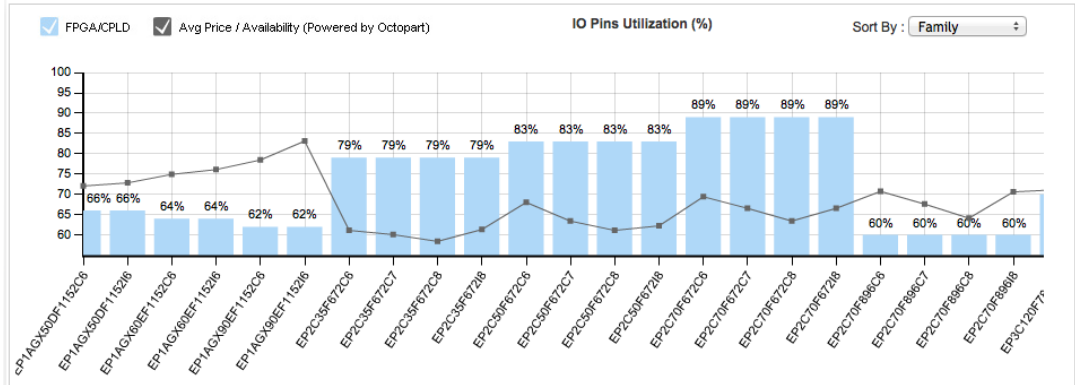
Figure 7B. Logic / IO Utilization across devices



Logic, IO Pin Utilization, Max Frequency and Total Power

These tabs give a high-level view of how the results differ across iterations and devices. The IO Pin Utilization view is shown as an example below (Figure 8).

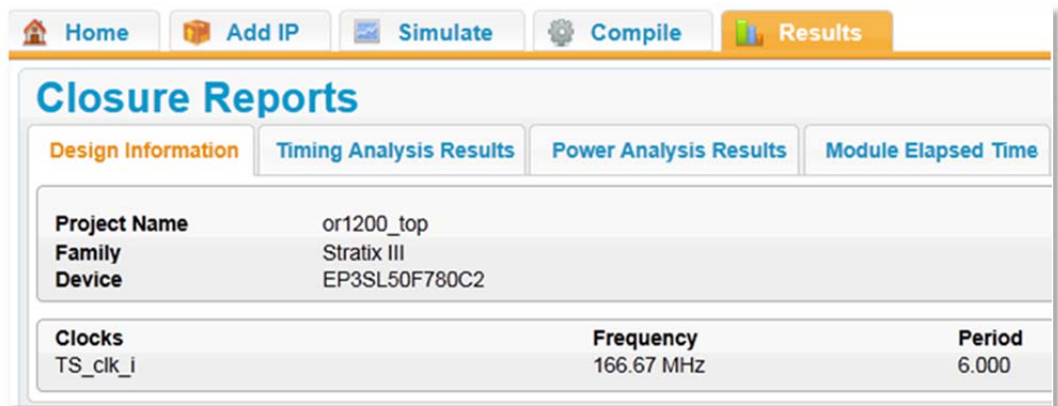
Figure 8. IO Pins Utilization and Prices across devices



“CLO” CLOSURE/COMPILE REPORTS

For “CLO” type jobs, after clicking on the “View” link, a new sub-page as shown in Figure 9, will show timing closure details for all sets of timing parameters attempted.

Figure 9. Basic information about the design



There are four views:

- **Design Information**
- **Timing Analysis Results**
- **Power Analysis Results**
- **Module Elapsed Time**

The charts below compare Timing ([Figure 10A](#)), Power ([Figure 10B](#)) and Runtime ([Figure 10C](#)) results obtained from different timing closure iterations of the design.

Exact values for each iteration can be viewed by holding the mouse cursor over each data point or bar.

Figure 10a. Timing Analysis



Figure 10b. Power Analysis



Figure 10c. Module Elapsed Time



Additionally, the same results are tabulated below each chart. Holding the mouse cursor over the **i** icon beside each revision name causes the exact timing parameters to pop up (Figure 10D).

Figure 10d. Settings

Multi Corner Analysis Summary: Worst Timing Results

Revision					Pulse Width
dse_point_11_2					2.112
dse_point_10_2					2.113
dse_point_9_2					2.112
dse_point_8_2					2.112
dse_point_7_2					2.109
dse_point_6_2	-0.366	0.179	N/A	N/A	2.109
dse_point_5_2	-0.324	0.179	N/A	N/A	2.112

MAX_CORE_JUNCTION_TEMP: 85
 MIN_CORE_JUNCTION_TEMP: 0
 NOMINAL_CORE_SUPPLY_VOLTAGE: 1.1V
 PARTITION_FITTER_PRESERVATION_LEVEL: PLACEMENT_AND_ROUTING
 PARTITION_NETLIST_TYPE: SOURCE
 PHYSICAL_SYNTHESIS_COMBO_LOGIC: On
 PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION: On
 PHYSICAL_SYNTHESIS_REGISTER_RETIMING: On
 PRE_MAPPING_RESYNTHESIS: On
 ROUTER_LCELL_INSERTION_AND_LOGIC_DUPLICATION: On
 ROUTER_TIMING_OPTIMIZATION_LEVEL: MAXIMUM
 SEED: 2

How to get support

The easiest way to get support is to use the Plunify support website

- For technical resources, download them at <http://www.plunify.com/en/portal.php>
- If you have a technical issue, log a support ticket at <http://www.plunify.com/en/support.php> or contact us at help@plunify.com with the details of your technical issue.
- There is also a FAQ page, <http://www.plunify.com/en/faqs.php>, which addresses some of the common questions.

Revision History

Version	Date	Description
1.0	November 2011	Initial release