

PLUNIFY

METHOD FOR SCALING DISTRIBUTED TIMING CLOSURE

CASE STUDY: OPENRISE 1200 PROCESSOR

Version 1.0

October 26, 2011

PLUNIFY

TABLE OF CONTENTS

INTRODUCTION.....	3
Background.....	3
Purpose	3
Example Design and Approach.....	3
Results and Analysis	3
OVERVIEW: SCALABLE DISTRIBUTED TIMING CLOSURE (SDTC).....	5
Prerequisites.....	5
Methodology	5
Required Tools and Hardware	5
OPENRISC 1200 PROCESSOR CORE	6
Logical View	6
Hardware Architecture.....	6
OR1200 Design Constraints.....	6
TIMING VIOLATION AND CLOSURE STEPS.....	7
Initial Implementation Results.....	7
run SDTC (Refer to Section 2.2 for description).....	7
RESULTS.....	8
Timing Analysis Results	8
Power Analysis Results.....	9
Runtime Comparison.....	9
Individual Iteration Settings.....	10
Implementation Tradeoffs	11
CONCLUSION AND NEXT STEPS.....	11
APPENDIX A: REFERENCES.....	12
APPENDIX B: KEY TERMS.....	13
APPENDIX C: OR1200 TIMING ANALYSIS TABLE.....	14
APPENDIX D: OR1200 POWER ANALYSIS TABLE	15
APPENDIX E: OR1200 DSE RUNTIME RESULTS.....	16

INTRODUCTION

BACKGROUND

Timing closure is an essential yet resource-consuming step in the FPGA design. In general here are common scenarios:

1. Timing violations are relatively minor, so making changes to tool parameters alone are likely to lead to a timing solution.
2. Timing slack is comparatively large, so changes at earlier stages of the design are required, for example, to the RTL or to the constraints.

In these cases, a significant number of synthesis, place-and-route and timing analysis iterations need to be run and then analyzed, incurring valuable time and resources. Design teams are very often limited by the lack of software licenses and servers, which in turn result in inefficient, serialized workflows.

PURPOSE

This study demonstrates a scalable distributed approach to timing closure, which customizes cloud computing resources specifically for chip design workflows. Using Plunify's design exploration techniques to perform automated what-if analysis on different design parameters, FPGA (Field Programmable Gate Array) designers can achieve timing closure and get more optimized designs in a shorter amount of time.

EXAMPLE DESIGN AND APPROACH

Using the Altera Quartus II software for synthesis and place-and-route, an OpenRISC 1200 (OR1200) 32-bit processor core was implemented on an Altera Stratix III FPGA. The OR1200 design initially fails to meet its timing requirements.

Subsequently, using an approach termed, “Scalable Distributed Timing Closure” (SDTC), sets of different timing parameters were calculated in a focused, random manner. The resulting sets of parameters were then run in parallel to achieve design closure, targeting the best performance at the fastest possible turnaround time.

RESULTS AND ANALYSIS

The amount of data increases in conjunction with the availability of results from multiple parallel runs. Current tools and methods for examining FPGA compilation results can be enhanced to higher levels of abstraction.

For example, instead of manually trying out several design iterations and compiling the timing and utilization statistics into a table for comparison later, automate these steps right from the beginning. This approach is described in subsequent sections.

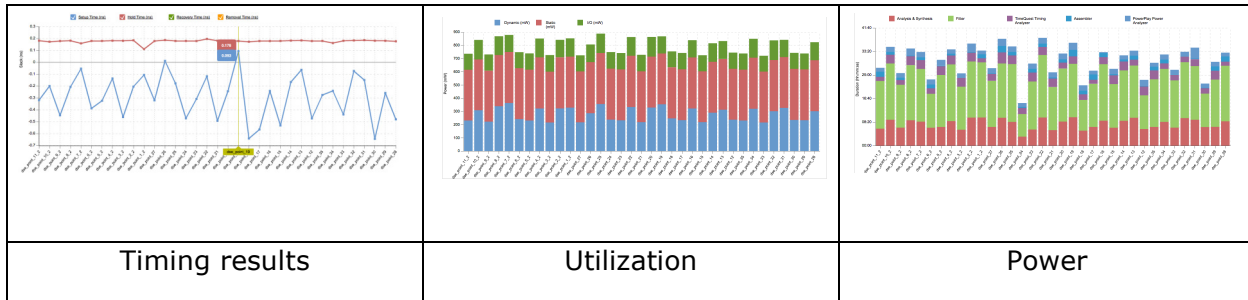


TABLE 1: EXAMPLES OF RESULT ANALYTICS

OVERVIEW: SCALABLE DISTRIBUTED TIMING CLOSURE (SDTC)

The underlying infrastructure for SDTC is a scalable cloud computing network pre-initialized with all the components for FPGA design – software tools, servers, Intellectual Property (IP) modules and design scripts. This can be considered an on-demand FPGA design resource, managed and controlled by an internal team or by a solutions provider like Plunify.

PREREQUISITES

Basic working knowledge of the command-line shell environment in FPGA and EDA (Electronic Design Automation) tools is helpful towards understanding how to run SDTC.

Such tools usually come with a scripting interface, for example, a TCL interpreter in the case of Altera Quartus II.

METHODOLOGY

Design tools provide hundreds of parameters that influence results like attainable clock frequencies, logic utilization and power consumption. Based on knowledge of the impact of timing-related parameters, these settings were grouped into categories of different optimization goals, desired effort levels and coverage.

For the OR1200 example design, 35 sets of timing parameters were calculated, with a focus on total runtime and achieving timing closure. Each set was sent to the managed cloud for parallel implementation. The results from each iteration were then analyzed to pick out the passing sets.

REQUIRED TOOLS AND HARDWARE

To authenticate usage, encrypt and load design files into a managed cloud resource, a desktop client consisting of TCL scripts and Java applications is required and used alongside an existing Quartus II software tool installation.

The Quartus II software provides the necessary TCL interpreter and Java Runtime Environment for packaging the FPGA design and executing the desktop client to encrypt and transmit the design.

Used in this case study:

- Windows 7 PC
Others: Windows XP, Windows Vista, Linux
- Altera Quartus II 10.0 SP1
Web and Subscription Editions available from <http://www.altera.com>
- Plunify FPGAAccel 1.0 desktop client
Downloadable from <http://www.plunify.com/downloads>

OPENRISC 1200 PROCESSOR CORE

The OpenRISC 1200 Processor (OR1200) is a 32-bit RISC processor with a scalar 5-stage integer pipeline, virtual Memory Management Units (MMUs) and basic DSP capabilities.

It supports two 1-way direct-mapped memory caches (8KB data, instruction caches), each with a block size of 16-bytes. Supplementary facilities include a real-time debug unit, a high resolution tick timer, a programmable interrupt controller and power management support. [see Appendix A].

A block diagram of the OR1200 core is shown in Figure 1.

LOGICAL VIEW

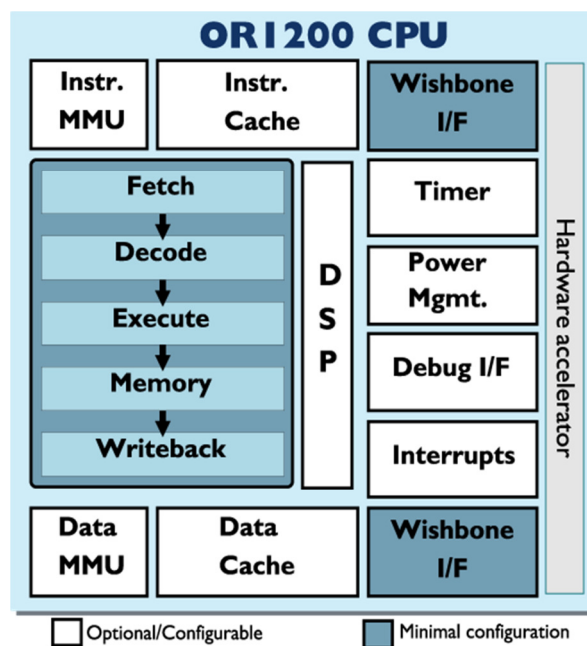


FIGURE 1. OPENRISC 1200 BLOCK DIAGRAM (SOURCE: OPENCORES.ORG)

HARDWARE ARCHITECTURE

Please refer to http://opencores.org/svnget.or1k?file=/trunk/docs/openisc_arch.pdf for more details on the OR1200.

OR1200 DESIGN CONSTRAINTS

The input clock, TS_clk_i, was constrained at 6ns.

TIMING VIOLATION AND CLOSURE STEPS

INITIAL IMPLEMENTATION RESULTS

Utilization (%)	Logic utilization: 14% Combinational ALUTs: 9% Dedicated logic registers: 5% Total pins: 77% Total block memory bits: 3% DSP block 18-bit elements: 2%
Worst Setup Time Slack	(Slow 1100mV 85C Model) TS_clk_i: -0.519ns
Maximum Frequency	(Slow 1100mV 85C Model) TS_clk_i: 153.4MHz

TABLE 2. IMPLEMENTATION STATISTICS

There is a setup timing violation of 0.519ns.

RUN SDTC (REFER TO SECTION 2.2 FOR DESCRIPTION)

From within a Quartus shell environment, navigate to the project directory and invoke a TCL script that carries out SDTC as described in Section 2.2.

```
% cd or1200_top  
% quartus_sh cloudcompile.tcl -project or1200_top -op closure -closuretype dse -src
```

```
2011.10.18 08:50:23 | File encrypting: /root/or1200/or1200_top.qar  
2011.10.18 08:50:44 | File has been uploaded to: /or1200_top/or1200_top.qar  
2011.10.18 08:50:44 | Upload COMPLETED!  
2011.10.18 08:50:44 | --- Job Id: 2065 submitted to Plunify.  
2011.10.18 08:50:44 | Job submitted with return code = 1  
2011.10.18 08:50:44 | Job submission COMPLETED!
```

FIGURE 2. SDTC REQUEST SUBMISSION TO MANAGED CLOUD

Number of iterations synthesized and placed-and-routed in parallel: 35

RESULTS

The outputs and values of the 35 SDTC iterations are then analyzed and examined in a summarized format.

TIMING ANALYSIS RESULTS

Performed using the Quartus II TimeQuest Timing Analyzer tool, static timing analysis reveals which iterations achieved positive Setup Time values. Figure 3 below shows the Multi-corner Worst Case Timing comparison across all iterations. Only two, *dse_point_26* and *dse_point_19*, have positive slack values. The best timing result was obtained in *dse_point_19* - Setup Slack = 0.093ns.

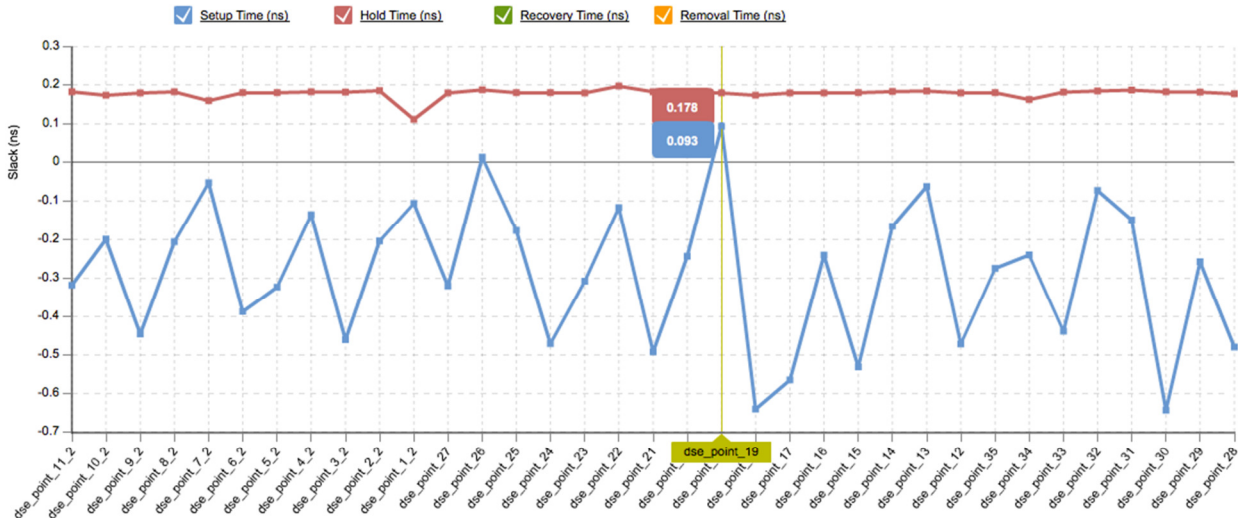


FIGURE 3. OPENRISC 1200 TIMING ANALYSIS RESULTS

The following figure shows the SDTC-generated settings used for *dse_point_19*.

```
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: 4
```

FIGURE 4. TIMING SETTINGS FOR DSE_POINT_19.

POWER ANALYSIS RESULTS

The Quartus II PowerPlay Power Analysis tool was used to estimate device power consumption. Figure 5 shows the distribution with power consumption divided into three types - dynamic power in blue, static power in red and IO power in green.

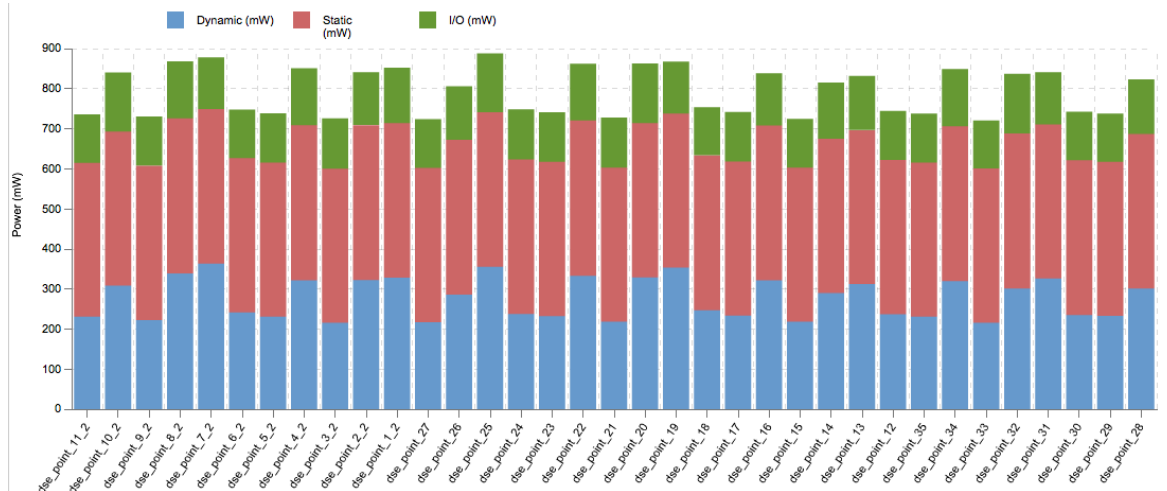


FIGURE 5. POWER ANALYSIS RESULTS

As shown in the graph, static (standby) power constituted the majority of the power consumption, which is pretty constant across all iterations. Dynamic power varies between 215mW to 363mW. The fastest performing iteration, *dse_point_19*, consumes 353.13mW of dynamic power and in total, 867.57mW, slightly lower than the highest total power consumption of 888.23mW.

RUNTIME COMPARISON

Figure 6 presents a comparison of how long each iteration took to run. Results are divided into different compilation stages, with the majority of the runtime taken up by synthesis (Analysis and Synthesis) and place-and-route (Fitter) stages.

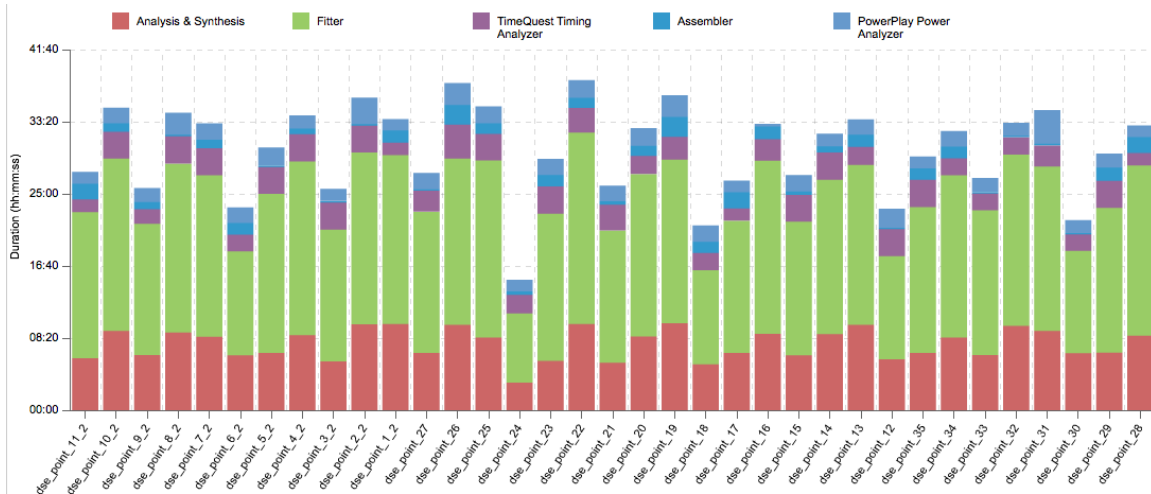


FIGURE 6. RUNTIME DIFFERENCES ACROSS DIFFERENT ITERATIONS

Based on the above chart, dse_point_24 has the fastest turnaround time of 15 minutes; however Figure 3 in an previous section shows that it misses timing requirements by -0.47ns. The iteration with the best timing results, dse_point_19, took 36 minutes to complete.

It is important to note that all of these 35 iterations were run in parallel. Thus, the maximum runtime to obtain all the results was just 38 minutes. The worst-case runtime to perform the same experiment serially can be calculated as the sum of all the individual runtimes which is equal to 17 hours and 34 minutes.

Runtime differences across different iterations are attributed to the settings used. High optimization effort levels and more aggressive place-and-rout settings will most likely translate to longer runtimes.

INDIVIDUAL ITERATION SETTINGS

To determine the relative impact (positive or negative) of the settings in each iteration, a passing case like dse_point_19 and a failing case such as dse_point_13 were compared side-by-side.

As Table 2 shows, the SEED value is the difference between timing failure and not achieving timing closure.

dse_point_13	dse_point_19
Worst Setup Slack: -0.064ns	Worst Setup Slack: 0.093ns
<pre> 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: 3 </pre>	<pre> 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: 4 </pre>

Table 3. Compare settings across two iterations.

IMPLEMENTATION TRADEOFFS

Timing closure does come at a price sometimes, for example when the FPGA software explores ways to pack logic more densely to shorten critical paths. In this case, overall logic utilization increased by 1%.

CONCLUSION AND NEXT STEPS

A scalable distributed approach to meeting clock frequency targets in FPGA designs was attempted on an OpenRISC 1200 processor core design. The OR1200 design initially failed setup time requirements, but through the described SDTC method, 35 sets of timing parameters were calculated and run in parallel in a cloud computing network optimized for FPGA design.

Results show that timing closure was successfully achieved through SDTC, almost 30X faster than if the iterations were run one after another. SDTC can be further improved to calculate parameter sets which are optimized for specific designs or design goals. A similar approach can be used for area and power closure.

Existing approaches to achieving timing closure is often constrained by the lack of compute resources, which in turn leads to serialized, inefficient workflows. With a secure, robust, on-demand cloud computing network, and a scalable distributed processing algorithm, designers can parallelize timing closure iterations in a targeted manner to achieve clock frequency targets at a fraction of the time required by traditional workflows.

APPENDIX A: REFERENCES

- [1] OR1200 OpenRISC processor, <http://opencores.org/openrisc.or1200>
- [2] OpenRISC 1200, http://en.wikipedia.org/wiki/OpenRISC_1200

APPENDIX B: KEY TERMS

The following table provides definitions for terms relevant to this document.

Term	Definition
Timing Closure	<p>Process by which an FPGA or an ASIC design is modified to meet its timing requirements. FPGA or EDA tools handle most of the modifications based on designer directives.</p> <p>Also used for the goal that is achieved, when such a design has reached the end of the flow and its timing requirements are satisfied.</p>
Cloud Computing	Cloud computing provides computation, software, data access, and storage services in a scalable, on-demand manner that do not require end-user knowledge of the physical location and configuration of the system that delivers the services.

APPENDIX C: OR1200 TIMING ANALYSIS TABLE

Multi Corner Analysis Summary: Worst Timing Results

Number of Parallel Runs: 35

Number of Runs that Pass Timing: 2

Best Results: dse_point_19

Revision	Setup Time (ns)	Hold Time (ns)	Recovery Time (ns)	Removal Time (ns)	Pulse Width	TNS Setup
dse_point_11_2	-0.3190	0.1810	N/A	N/A	2.1120	-8.24
dse_point_10_2	-0.2010	0.1720	N/A	N/A	2.1130	-2.35
dse_point_9_2	-0.4470	0.1780	N/A	N/A	2.1120	-37.96
dse_point_8_2	-0.2070	0.1810	N/A	N/A	2.1120	-1.53
dse_point_7_2	-0.0540	0.1580	N/A	N/A	2.1090	-0.05
dse_point_6_2	-0.3880	0.1790	N/A	N/A	2.1090	-72.97
dse_point_5_2	-0.3240	0.1790	N/A	N/A	2.1120	-26.62
dse_point_4_2	-0.1370	0.1810	N/A	N/A	2.1090	-1.22
dse_point_3_2	-0.4620	0.1800	N/A	N/A	2.1120	-96.69
dse_point_2_2	-0.2050	0.1840	N/A	N/A	2.1120	-2.39
dse_point_1_2	-0.1070	0.1100	N/A	N/A	2.1120	-0.78
dse_point_27	-0.3210	0.1780	N/A	N/A	2.1120	-26.63
dse_point_26	0.0110	0.1860	N/A	N/A	2.1120	0.00
dse_point_25	-0.1780	0.1790	N/A	N/A	2.1090	-0.75
dse_point_24	-0.4720	0.1790	N/A	N/A	2.1090	-169.46
dse_point_23	-0.3090	0.1780	N/A	N/A	2.1090	-26.39
dse_point_22	-0.1180	0.1960	N/A	N/A	2.1120	-0.79
dse_point_21	-0.4930	0.1810	N/A	N/A	2.1120	-94.92
dse_point_20	-0.2450	0.1800	N/A	N/A	2.1110	-2.58
dse_point_19	0.0930	0.1780	N/A	N/A	2.1120	0.00
dse_point_18	-0.6420	0.1720	N/A	N/A	2.1110	-140.84
dse_point_17	-0.5660	0.1780	N/A	N/A	2.1090	-52.62
dse_point_16	-0.2420	0.1780	N/A	N/A	2.1110	-8.95
dse_point_15	-0.5320	0.1790	N/A	N/A	2.1120	-106.62
dse_point_14	-0.1670	0.1820	N/A	N/A	2.1120	-4.62
dse_point_13	-0.0640	0.1830	N/A	N/A	2.1090	-0.12
dse_point_12	-0.4730	0.1780	N/A	N/A	2.1120	-46.40
dse_point_35	-0.2760	0.1790	N/A	N/A	2.1120	-20.32
dse_point_34	-0.2410	0.1610	N/A	N/A	2.1120	-1.20
dse_point_33	-0.4400	0.1800	N/A	N/A	2.1120	-82.23
dse_point_32	-0.0740	0.1830	N/A	N/A	2.1120	-0.20
dse_point_31	-0.1500	0.1850	N/A	N/A	2.1090	-0.73
dse_point_30	-0.6450	0.1810	N/A	N/A	2.1120	-151.36
dse_point_29	-0.2590	0.1800	N/A	N/A	2.1120	-10.55
dse_point_28	-0.4810	0.1760	N/A	N/A	2.1110	-53.97

APPENDIX D: OR1200 POWER ANALYSIS TABLE

Number of Parallel Runs: 35

Highest Power Consumption: 888.23mW

Lowest Power Consumption: 720.86mW

Power Consumption of Run with Best Timing: 867.57mW (dse_point_19)

Revision	Power Models	Dynamic (mW)	Static (mW)	I/O (mW)	Total (mW)
dse_point_1_2	Final	328.43	385.88	138.30	852.61
dse_point_10_2	Final	308.29	384.67	147.84	840.80
dse_point_11_2	Final	230.94	384.30	121.22	736.47
dse_point_12	Final	236.72	385.72	122.81	745.26
dse_point_13	Final	312.66	384.22	134.78	831.66
dse_point_14	Final	289.91	385.21	140.23	815.36
dse_point_15	Final	218.67	384.55	121.52	724.74
dse_point_16	Final	321.44	387.10	130.29	838.82
dse_point_17	Final	233.89	384.44	124.08	742.40
dse_point_18	Final	247.15	386.76	119.97	753.88
dse_point_19	Final	353.13	384.60	129.84	867.57
dse_point_2_2	Final	322.58	386.73	132.05	841.35
dse_point_20	Final	328.71	385.61	149.14	863.47
dse_point_21	Final	218.26	384.99	125.34	728.59
dse_point_22	Final	333.24	387.86	141.52	862.62
dse_point_23	Final	232.41	385.08	123.72	741.20
dse_point_24	Final	237.34	385.89	124.82	748.05
dse_point_25	Final	355.59	385.72	146.91	888.23
dse_point_26	Final	285.79	386.49	133.90	806.17
dse_point_27	Final	217.33	385.26	121.63	724.23
dse_point_28	Final	301.21	385.71	136.49	823.41
dse_point_29	Final	232.87	384.84	120.25	737.97
dse_point_3_2	Final	215.85	384.56	125.48	725.89
dse_point_30	Final	234.81	386.82	121.27	742.89
dse_point_31	Final	326.31	384.40	130.17	840.89
dse_point_32	Final	301.47	387.01	148.80	837.28
dse_point_33	Final	215.96	385.03	119.87	720.86
dse_point_34	Final	319.86	386.50	142.78	849.14
dse_point_35	Final	230.71	384.71	122.69	738.12
dse_point_4_2	Final	321.70	387.52	142.54	851.76
dse_point_5_2	Final	231.16	384.32	122.99	738.47
dse_point_6_2	Final	241.72	385.23	120.96	747.91
dse_point_7_2	Final	363.58	385.67	129.07	878.32
dse_point_8_2	Final	338.84	387.33	142.17	868.35
dse_point_9_2	Final	222.59	385.48	122.67	730.74

APPENDIX E: OR1200 DSE RUNTIME RESULTS

Number of Parallel Runs: **35**

Longest Design Compilation Time: **37 minutes, 49 seconds (dse_point_26)**

(All times expressed in hh:mm:ss format)

Revision	Analysis & Synthesis	Fitter	TimeQuest Timing Analyzer	Assembler	PowerPlay Power Analyzer	Total
dse_point_1_2	0:09:57	0:19:32	0:01:26	0:01:26	0:01:19	0:33:40
dse_point_10_2	0:09:10	0:19:55	0:03:05	0:00:59	0:01:48	0:34:57
dse_point_11_2	0:06:01	0:16:53	0:01:28	0:01:49	0:01:23	0:27:34
dse_point_12	0:05:53	0:11:55	0:03:08	0:00:10	0:02:12	0:23:18
dse_point_13	0:09:52	0:18:29	0:02:05	0:01:25	0:01:46	0:33:37
dse_point_14	0:08:48	0:17:50	0:03:09	0:00:43	0:01:29	0:31:59
dse_point_15	0:06:20	0:15:28	0:03:06	0:00:24	0:01:55	0:27:13
dse_point_16	0:08:49	0:20:01	0:02:30	0:01:28	0:00:18	0:33:06
dse_point_17	0:06:38	0:15:16	0:01:26	0:01:52	0:01:21	0:26:33
dse_point_18	0:05:18	0:10:53	0:01:59	0:01:18	0:01:54	0:21:22
dse_point_19	0:10:04	0:18:54	0:02:38	0:02:18	0:02:31	0:36:25
dse_point_2_2	0:09:55	0:19:53	0:03:06	0:00:11	0:03:04	0:36:09
dse_point_20	0:08:32	0:18:46	0:02:06	0:01:10	0:02:03	0:32:37
dse_point_21	0:05:30	0:15:16	0:03:00	0:00:24	0:01:48	0:25:58
dse_point_22	0:09:57	0:22:08	0:02:52	0:01:10	0:02:03	0:38:10
dse_point_23	0:05:42	0:17:01	0:03:09	0:01:19	0:01:53	0:29:04
dse_point_24	0:03:11	0:08:00	0:02:09	0:00:24	0:01:21	0:15:05
dse_point_25	0:08:24	0:20:29	0:03:04	0:01:13	0:01:57	0:35:07
dse_point_26	0:09:53	0:19:12	0:03:55	0:02:17	0:02:32	0:37:49
dse_point_27	0:06:38	0:16:19	0:02:25	0:00:11	0:01:55	0:27:28
dse_point_28	0:08:37	0:19:40	0:01:27	0:01:50	0:01:21	0:32:55
dse_point_29	0:06:39	0:16:43	0:03:08	0:01:32	0:01:38	0:29:40
dse_point_3_2	0:05:38	0:15:14	0:03:08	0:00:10	0:01:24	0:25:34
dse_point_30	0:06:35	0:11:50	0:01:54	0:00:10	0:01:31	0:22:00
dse_point_31	0:09:10	0:19:00	0:02:26	0:00:12	0:03:53	0:34:41
dse_point_32	0:09:45	0:19:48	0:02:00	0:00:11	0:01:30	0:33:14
dse_point_33	0:06:23	0:16:43	0:01:57	0:00:10	0:01:38	0:26:51
dse_point_34	0:08:24	0:18:45	0:01:58	0:01:21	0:01:49	0:32:17
dse_point_35	0:06:38	0:16:51	0:03:09	0:01:19	0:01:22	0:29:19
dse_point_4_2	0:08:40	0:20:05	0:03:09	0:00:39	0:01:33	0:34:06
dse_point_5_2	0:06:37	0:18:23	0:03:07	0:00:10	0:02:05	0:30:22
dse_point_6_2	0:06:20	0:12:01	0:01:56	0:01:22	0:01:49	0:23:28
dse_point_7_2	0:08:30	0:18:39	0:03:07	0:01:00	0:01:55	0:33:11
dse_point_8_2	0:08:59	0:19:30	0:03:11	0:00:11	0:02:32	0:34:23
dse_point_9_2	0:06:22	0:15:10	0:01:42	0:00:49	0:01:37	0:25:40

For additional information, please contact sales.ap@plunify.com (Asia Pacific) or sales.us@plunify.com (North America/Europe/North Africa/Middle East)

Revision History

Version	Date	Description
1.0	October 2011	Initial release

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 are 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 agreements, and this document is not part of, nor does it modify, any agreement between Plunify and its customers.

