Synplicity®

Simply Better Results

Synplicity® FPGA Synthesis

Synplify Premier Quick Start Guide for Altera

July 2008

Synplicity, Inc.
600 West California Avenue
Sunnyvale, CA 94086, USA
(U.S.) +1 408 215-6000 direct
(U.S.) +1 408 990-0263 fax
www.synplicity.com

Preface

Disclaimer of Warranty

Synplicity, Inc. makes no representations or warranties, either expressed or implied, by or with respect to anything in this manual, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose of for any indirect, special or consequential damages.

Copyright Notice

Trademarks

Restricted Rights Legend

Printed in the U.S.A
July 2008

Synplicity Software License Agreement

Important! READ CAREFULLY BEFORE PROCEEDING

BY INDICATING YOUR ACCEPTANCE OF THE TERMS OF THIS AGREEMENT, YOU (“LICENSEE”) ARE REPRESENTING THAT YOU HAVE THE RIGHT AND AUTHORITY TO LEGALLY BIND YOURSELF OR YOUR COMPANY, AS APPLICABLE, AND CONSENTING TO BE LEGALLY BOUND BY ALL OF THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS DO NOT INSTALL OR USE THE SOFTWARE, AND RETURN THE SOFTWARE TO THE LOCATION OF PURCHASE FOR A REFUND. This is a legal agreement governing use of the software program provided by Synplicity, Inc. (“Synplicity”) to you (the “SOFTWARE”). The term “SOFTWARE” also includes related documentation (whether in print or electronic form), any authorization keys, authorization codes, and license files, and any updates or upgrades of the SOFTWARE provided by Synplicity, but does not include certain “open source” software licensed by third party licensors and made available to you by Synplicity under the terms of such third party licensor’s license (such as software licensed under the General Public License (GPL)) (“Third Party Software”). If Licensee is a participant in the University Program or has been granted an Evaluation License or Subscription License, then some of the following terms and conditions may not apply (refer to the sections entitled, respectively, **Evaluation License** and **Subscription License**, below).

License. Synplicity grants to Licensee a non-exclusive right to install the SOFTWARE and to use or authorize use of the SOFTWARE by up to the number of nodes for which Licensee has a license and for which Licensee has the security key(s) or authorization code(s) provided by Synplicity or its agents for the purpose of creating and modifying Designs (as defined below). If Licensee has obtained the SOFTWARE under a node-locked license, then a “node” refers to a specific machine, and the SOFTWARE may be installed only on the number of “nodes” or machines authorized, must be used only on the machine(s) on which it is installed, and may be accessed only by users who are physically present at that node or machine. A node-locked license may only be used by one user at a time running one instance of the software at a time. If Licensee has obtained the SOFTWARE under a “floating” license, then a “node” refers to a concurrent user or session, and the SOFTWARE may be used concurrently by up to the number of users or sessions indicated. All SOFTWARE must be used within the country for which the systems were licensed and at Licensee's Site (contained within a one kilometer radius); however, if Licensee has a floating license then remote use is permitted by employees who work at the site but are temporarily telecommuting to that same site from less than 50 miles away (for example, an employee who works at a home office on occasion), but the maximum number of concurrent sessions or nodes still applies. In addition, Synplicity grants to Licensee a non-exclusive license to copy and distribute internally the documentation portion of the SOFTWARE in support of its license to use the program portion of the SOFTWARE. For purposes of this Agreement the “Licensee’s Site” means the location of the server on which the SOFTWARE resides, or when a server is not required, the location of the client computer for which the license was issued.

Evaluation License. If Licensee has obtained the SOFTWARE pursuant to an evaluation license, then, in addition to all other terms and conditions herein, the following restrictions apply: (a) the license to the SOFTWARE terminates after 20 days (unless otherwise agreed to in writing by Synplicity); and (b) Licensee may use the SOFTWARE only for the sole purpose of internal testing and evaluation to determine whether Licensee wishes to license the SOFTWARE on a commercial basis. Licensee shall not use the SOFTWARE to design any integrated circuits for production or pre-production purposes or any other commercial use including, but not limited to, for the benefit of Licensee's customers. If Licensee breaches any of the foregoing restrictions, then Licensee shall pay to Synplicity a license fee equal to Synplicity's perpetual list price plus maintenance for the commercial version of the SOFTWARE.

Subscription (Time-Based) License. If Licensee has obtained a Subscription License to the SOFTWARE, the, in addition to all other terms and conditions herein, the following restrictions apply: (a) Licensee is authorized to use the SOFTWARE only for a limited time (which time is indicated on the quotation or in the purchase confirmation documents); (b) Licensee's right to use the SOFTWARE terminates on the date the subscription term expires as set forth in the quotation or the purchase confirmation documents, unless Licensee has renewed the license by paying the applicable fees.

Project Based License. If Licensee has obtained a Project-Based License to the SOFTWARE, in addition to all other terms and conditions herein, the terms of Exhibit A will apply.

Copy Restrictions. This SOFTWARE is protected by United States copyright laws and international treaty provisions and Licensee may copy the SOFTWARE only as follows: (i) to directly support authorized use under the license, and (ii) in order to make a copy of the SOFTWARE for backup purposes. Copies must include all copyright and trademark notices.

Use Restrictions. This SOFTWARE is licensed to Licensee for internal use only. Licensee shall not (and shall not allow any third party to): (i) decompile, disassemble, reverse engineer or attempt to reconstruct, identify or discover any source code, underlying ideas, underlying user interface techniques or algorithms of the SOFTWARE by any means whatever, or disclose any of the foregoing; (ii) provide, lease, lend, or use the SOFTWARE for timesharing or service bureau purposes, on an application service provider basis, or otherwise circumvent the internal use restrictions; (iii) modify, incorporate into or with other software, or create a derivative work of any part of the SOFTWARE; (iv) disclose the results of any benchmarking of the SOFTWARE, or use such results for its own competing software development activities, without the prior written permission of Synplicity; or (v) attempt to circumvent any user limits, maximum gate count limits or other license, timing or use restrictions that are built into the SOFTWARE.

Transfer Restrictions/No Assignment. The SOFTWARE may only be used under this license at the designated locations and designated equipment as set forth in the license grant above, and may not be moved to other locations or equipment or otherwise transferred without the prior written consent of Synplicity. Any permitted transfer of the SOFTWARE will require that Licensee executes a "Software Authorization Transfer Agreement" provided by Synplicity.   Further, Licensee shall not sublicense, or assign this Agreement or any of the rights or licenses granted under this Agreement, without the prior written consent of Synplicity.

Security. Licensee agrees to take all appropriate measures to safeguard the SOFTWARE and prevent unauthorized access or use thereof. Suggested ways to accomplish this include: (i) implementation of firewalls and other security applications, (ii) use of FLEXlm options file that restricts access to the SOFTWARE to identified users; (iii) maintaining and storing license information in paper format only; (iv) changing TCP port numbers every three (3) months; and (v) communicating to all authorized users that use of the SOFTWARE is subject to the restrictions set forth in this Agreement.

Ownership of the SOFTWARE. Synplicity retains all right, title, and interest in the SOFTWARE (including all copies), and all worldwide intellectual property rights therein. Synplicity reserves all rights not expressly granted to Licensee. This license is not a sale of the original SOFTWARE or of any copy.

Ownership of Design Techniques. “Design” means the representation of an electronic circuit or device(s), derived or created by Licensee through the use of the SOFTWARE in its various formats, including, but not limited to, equations, truth tables, schematic diagrams, textual descriptions, hardware description languages, and netlists. “Design Techniques” means the data, circuit and logic elements, libraries, algorithms, search strategies, rule bases, techniques and technical information incorporated in the SOFTWARE and employed in the process of creating Designs. Synplicity retains all right, title and interest in and to Design Techniques incorporated in the SOFTWARE, including all intellectual property rights embodied therein, provided that to the extent any Design Techniques are included as part of or embedded within Licensee’s Designs, Synplicity grants Licensee a personal, non exclusive, nontransferable license to reproduce the Design Techniques and distribute such Design Techniques solely as incorporated into Licensee’s Designs and not on a standalone basis. Additionally, Licensee acknowledges that Synplicity has an unrestricted, royalty-free right to incorporate any Design Techniques disclosed by Licensee into its software, documentation and other products, and to sublicense third parties to use those incorporated design techniques.

Protection of Confidential Information. “Confidential Information” means (i) the SOFTWARE, in object and source code form, and any related technology, idea, algorithm or information contained therein, including without limitation Design Techniques, and any trade secrets related to any of the foregoing; (ii) either party's product plans, Designs, costs, prices and names; non-published financial information; marketing plans; business opportunities; personnel; research; development or know-how; (iii) any information designated by the disclosing party as confidential in writing or, if disclosed orally, designated as confidential at the time of disclosure and reduced to writing and designated as confidential in writing within thirty (30) days; and (iv) the terms and conditions of this Agreement; provided, however that “Confidential Information” will not include information that: (a) is or becomes generally known or available by publication, commercial use or otherwise through no fault of the receiving party; (b) is known and has been reduced to tangible form by the receiving party at the time of disclosure and is not subject to restriction; (c) is independently developed by the receiving party without use of the disclosing party's Confidential Information; (d) is lawfully obtained from a third party who has the right to make such disclosure; and (e) is released for publication by the disclosing party in writing.

Each party will protect the other's Confidential Information from unauthorized dissemination and use with the same degree of care that each such party uses to protect its own like information. Neither party will use the other's Confidential Information for purposes other than those necessary to directly further the purposes of this Agreement. Neither party will disclose to third parties the other's Confidential Information without the prior written consent of the other party.

Open Source Software. The SOFTWARE may be delivered with software that is subject to open source licensing terms (“Open Source Software”) which are available at http://www.synplicity.com/products/license_agreement.html. If the Open Source Software license also requires source code to be made available, Licensee may reference http://www.synplicity.com/products/opensource.html for information on how to obtain such source code. Licensee agrees that all Open Source Software shall be and shall remain subject to the terms and conditions under which it is provided. The Open Source Software is provided “AS IS,” WITHOUT ANY WARRANTY OF ANY KIND, AND SYNPLICITY FURTHER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO OPEN SOURCE SOFTWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. NEITHER SYNPLICITY NOR THE LICENSORS OF OPEN SOURCE SOFTWARE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AN ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE ECLIPSE SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Copyrights to the Open Source Software are held by the copyright holders indicated in the copyright notices in the corresponding source files.

Termination. Synplicity may terminate this Agreement immediately if Licensee breaches any provision, including without limitation, failure by Licensee to implement adequate security measures as set forth above. Upon notice of termination by Synplicity, all rights granted to Licensee under this Agreement will immediately terminate, and Licensee shall cease using the SOFTWARE and return or destroy all copies (and partial copies) of the SOFTWARE and documentation.

Limited Warranty and Disclaimer. Synplicity warrants that the program portion of the SOFTWARE will perform substantially in accordance with the accompanying documentation for a period of 90 days from the date of receipt. Synplicity's entire liability and Licensee's exclusive remedy for a breach of the preceding limited warranty shall be, at Synplicity's option, either (a) return of the license fee, or (b) providing a fix, patch, workaround, or replacement of the SOFTWARE. In either case, Licensee must return the SOFTWARE to Synplicity with a copy of the purchase receipt or similar document. Replacements are warranted for the remainder of the original warranty period or 30 days, whichever is longer. Some states/jurisdictions do not allow limitations, so the above limitation may not apply. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES OR CONDITIONS, EITHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, ARE MADE BY SYNPLICITY OR ITS LICENSORS WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING DOCUMENTATION, AND SYNPLICITY EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS NOT EXPRESSLY STATED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE. SYNPLICITY AND ITS LICENSORS DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET LICENSEE'S REQUIREMENTS, BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS IN THE PROGRAM WILL BE CORRECTED. Licensee assumes the entire risk as to the results and performance of the SOFTWARE. Some states/jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply.

Limitation of Liability. IN NO EVENT SHALL SYNPLICITY OR ITS LICENSORS OR THEIR AGENTS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTIONS, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF SYNPLICITY AND/OR ITS LICENSORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. FURTHER, IN NO EVENT SHALL SYNPLICITY'S LICENSORS BE LIABLE FOR ANY DIRECT DAMAGES ARISING OUT OF LICENSEE'S USE OF THE SOFTWARE. IN NO EVENT WILL SYNPLICITY OR ITS LICENSORS BE LIABLE TO LICENSEE FOR DAMAGES IN AN AMOUNT GREATER THAN THE FEES PAID FOR THE USE OF THE SOFTWARE. Some states/jurisdictions do not allow the limitation or exclusion of incidental or consequential damages, so the above limitations or exclusions may not apply.

Intellectual Property Right Infringement. Synplicity will defend or, at its option, settle any claim or action brought against Licensee to the extent it is based on a third party claim that the SOFTWARE as used within the scope of this Agreement infringes or violates any US patent, copyright, trade secret or trademark of any third party, and Synplicity will indemnify and hold Licensee harmless from and against any damages, costs and fees reasonably incurred that are attributable to such claim or action; provided that Licensee provides Synplicity with (i) prompt written notification of the claim or action; (ii) sole control and authority over the defense or settlement thereof (including all negotiations); and (iii) at Synplicity's expense, all available information, assistance and authority to settle and/or defend any such claim or action. Synplicity's obligations under this subsection do not apply to the extent that (i) such claim or action would have been avoided but for modifications of the SOFTWARE, or portions thereof, other than modifications made by Synplicity after delivery to Licensee; (ii) such claim or action would have been avoided but for the combination or use of the SOFTWARE, or portions thereof, with other products, processes or materials not supplied or specified in writing by Synplicity; (iii) Licensee continues allegedly infringing activity after being notified thereof or after being informed of modifications that would have avoided the alleged infringement; or (iv) Licensee's use of the SOFTWARE is not strictly in accordance with the terms of this Agreement. Licensee will be liable for all damages, costs, expenses, settlements and attorneys' fees related to any claim of infringement arising as a result of (i)-(iv) above.

If the SOFTWARE becomes or, in the reasonable opinion of Synplicity is likely to become, the subject of an infringement claim or action, Synplicity may, at Synplicity's option and at no charge to Licensee, (a) obtain a license so Licensee may continue use of the SOFTWARE; (b) modify the SOFTWARE to avoid the infringement; (c) replace the SOFTARE with a compatible, functionally equivalent, and non-infringing product, or (d) if Synplicity determines that options (a), (b), and (c) are not commercially reasonable, then Synplicity shall have the right to terminate the licenses granted hereunder and refund to Licensee the amount paid for the SOFTWARE, as depreciated on a straight-line 5-year basis, or such other shorter period applicable to Subscription Licenses.

THE FOREGOING PROVISIONS OF THIS SECTION STATE THE ENTIRE AND SOLE LIABILITY AND OBLIGATIONS OF SYNPLICTY, AND THE EXCLUSIVE REMEDY OF LICENSEE, WITH RESPECT TO ANY ACTUAL OR ALLEGED INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHTS BY THE SOFTWARE (INCLUDING DESIGN TECHNIQUES) AND DOCUMENTATION.

Export. Licensee warrants that it is not prohibited from receiving the SOFTWARE under U.S. export laws; that it is not a national of a country subject to U.S. trade sanctions; that it will not use the SOFTWARE in a location that is the subject of U.S. trade sanctions that would cover the SOFTWARE; and that to its knowledge it is not on the U.S. Department of Commerce's table of deny orders or otherwise prohibited from obtaining goods of this sort from the United States.

Miscellaneous. This Agreement is the entire agreement between Licensee and Synplicity with respect to the license to the SOFTWARE, and supersedes any previous oral or written communications or documents (including, if you are obtaining an update, any agreement that may have been included with the initial version of the Software). This Agreement is governed by the laws of the State of California, USA excluding its conflicts of laws principals. This Agreement will not be governed by the U. N. Convention on Contracts for the International Sale of Goods and will not be governed by any statute based on or derived from the Uniform Computer Information Transactions Act (UCITA). If any provision, or portion thereof, of this Agreement is found to be invalid or unenforceable, it will be enforced to the extent permissible and the remainder of this Agreement will remain in full force and effect. Failure to prosecute a party's rights with respect to a default hereunder will not constitute a waiver of the right to enforce rights with respect to the same or any other breach.

Government Users. If the SOFTWARE is licensed to the United States government or any agency thereof, then the SOFTWARE and any accompanying documentation will be deemed to be “commercial computer software” and “commercial computer software documentation”, respectively, pursuant to DFAR Section 227.7202 and FAR Section 12.212, as applicable. Any use, reproduction, release, performance, display or disclosure of the SOFTWARE and accompanying documentation by the U.S. Government will be governed solely by the terms of this Agreement and are prohibited except to the extent expressly permitted by the terms of this Agreement.

March 2008

CHAPTER 1

Synplify Premier Tool for Altera Devices

This document describes the design flow for the Synplify Premier tool using Altera technologies. Topics include:

Synplify Premier Overview

- Physical Synthesis
- Prerequisites
- Supported Altera Devices
- Supported Altera Place and Route Tools

Physical Synthesis

The Synplify Premier product is a physical synthesis timing closure solution that provides more accurate timing correlation and faster timing closure than could be achieved through previous design methodologies. This tool offers a push-button, graph-based design flow for improving overall device performance while simultaneously delivering tight correlation between pre-route timing estimates and final post place-and-route results. The essence of the graph-based approach is that preexisting wires, switches and placement sites used for routing an FPGA can be represented as a detailed routing resource graph. The notion of distance then changes to a measure of delay and availability of wires. Graph-based physical synthesis technology merges optimization and placement to generate a fully placed and physically optimized netlist, providing rapid timing closure and increased timing improvement. You can enable the retiming feature to provide an additional performance boost.

The Synplify Premier tool supports these flows:

- Logic Synthesis Validation Phase – synthesis validation phase to ensure that the design has realistic constraints and can successfully complete synthesis and place and route.
- Graph-based Physical Synthesis – push-button, single pass flow that merges optimization and placement to generate a fully placed, physically optimized netlist.
- Graph-based Physical Synthesis with Design Planner – single pass flow which includes a design plan physical constraint file to guide global placement.
- Design-plan Based Physical Synthesis – flow that requires manual placement of the critical path to improve the design. This flow is for older technologies and requires the Synplify Premier tool with Design Planner option.

See *Supported Altera Devices,* on page 1-4 and *Supported Altera Place and Route Tools,* on page 1-4 to determine the flows available for your Synplify Premier version.

See *Flow Overview,* on page 1-5 for more details on the flows.

Prerequisites

- Include the entire design – black boxes cannot be present. However, Altera LPMs (Library of Parameterized Modules) or Megafunctions are supported. (See, *Using Altera LPMs or Megafunctions in Synthesis,* on page 4-44 in the *User Guide* if more information is needed.)
- Use the appropriate methodology defined for Altera IPs or Nios II cores in the design. For more information in the *User Guide*, see:
  - *Including Altera MegaCore IP Using an IP Package,* on page 4-37
  - *Including Altera MegaCore IP Using a Greybox Netlist,* on page 4-41
  - *Including Altera Processor Cores Generated in SOPC Builder,* on page 4-32.
- Assign realistic, accurate timing constraints. Do not over-constrain the tool. (See *Improve Performance,* on page 1-33 for tips.)
- Use the top-down design methodology. (A bottom-up flow is not supported.)
- Do not use the MultiPoint Synthesis flow with graph-based physical synthesis.
- Install the recommended version of the Altera Quartus II place-and-route tool.

Supported Altera Devices

Synplify Premier physical synthesis is supported for the following Altera technologies:

Physical Optimization Flows	Altera Devices
Graph-based Physical Synthesis	Stratix II Stratix II GX Stratix III
Graph-based Physical Synthesis with Design Planner [1]	Stratix II Stratix II GX Stratix III
Design Plan based Physical Synthesis[1]	Cyclone Cyclone II Stratix Stratix GX

[1] Requires the Synplify Premier tool with the Design Plan option.

Supported Altera Place and Route Tools

Consult the release notes for the most current information on supported Quartus versions. (From the Synplify Premier tool: Help->Online Documents->release_notes.pdf->*Third Party Tool Versions*)

Flow Overview

This section describes the flows for physical synthesis. The flow you use depends on your Synplify Premier tool and your target technology. See *Supported Altera Devices*, on page 1-4 to determine the flows available for your Synplify Premier version.

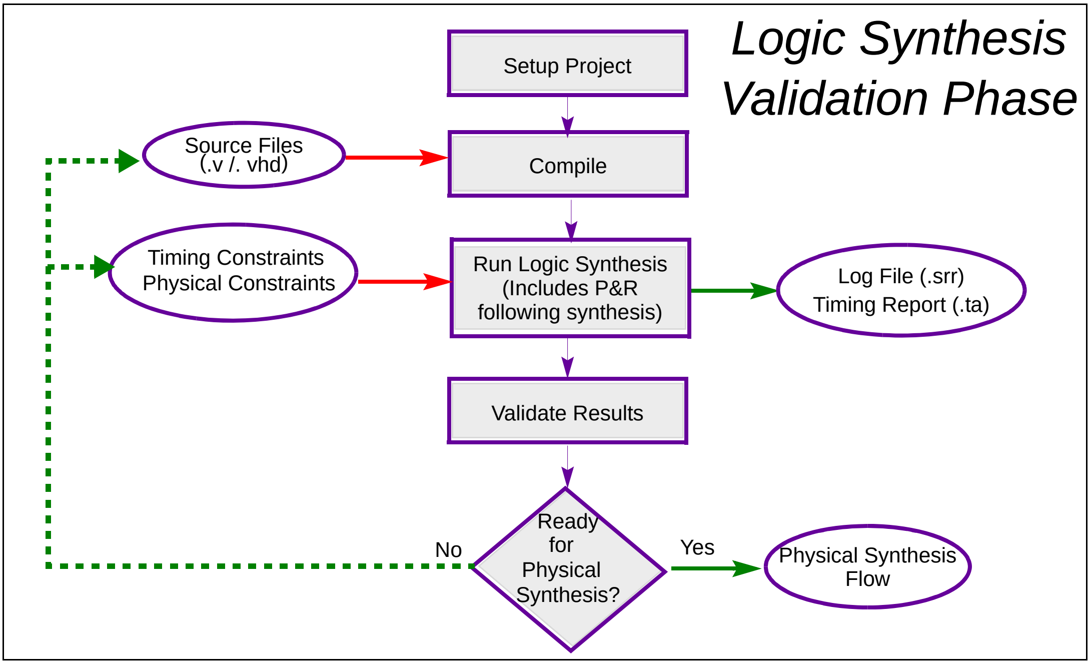
- Logic Synthesis Validation Phase
- Graph-based Physical Synthesis
- Graph-based Physical Synthesis with Design Planner
- Design-plan Based Physical Synthesis

Logic Synthesis Validation Phase

A prerequisite to physical synthesis is to first run the design through logic synthesis to ensure the design:

- Can successfully complete synthesis.
- Can successfully complete place and route.
- Has been assigned accurate, realistic constraints.

The figure below shows the flow for logic synthesis validation phase.


Logic Synthesis Validation Phase
Setup Project
Source Files (.v /. vhd)
Compile
Timing Constraints Physical Constraints
Run Logic Synthesis (Includes P&R following synthesis)
Log File (.srr) Timing Report (.ta)
Validate Results
No
Ready for Physical Synthesis?
Yes
Physical Synthesis Flow

See Also

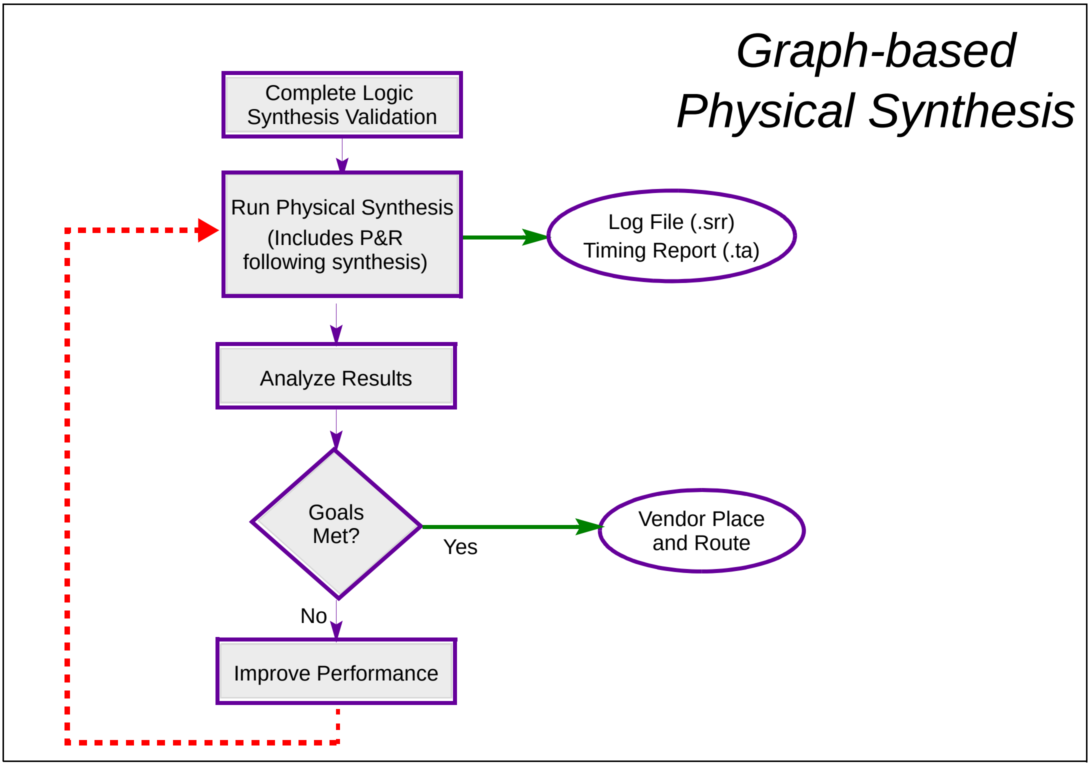
- Steps 1 through 3 in *Physical Synthesis Task Summary*, on page 1-10 for a summary of tasks required to complete this flow.
- *Set up Project*, on page 1-11 to start the physical synthesis design flow.

Graph-based Physical Synthesis

This is a single-pass, push-button physical synthesis flow that merges design optimization and placement to generate a fully-placed, physically-optimized netlist, providing rapid timing closure and increased timing improvement. Synthesis and placement are integrated by performing concurrent placement and optimization based on timing constraints and device technology. The output netlist contains placement information. Graph-based physical synthesis also simplifies the process for critical path timing improvements.

This flow can only be used with the Stratix II, Stratix II GX, and Stratix III Altera technologies.

The figure below shows the flow for graph-based physical synthesis.


Graph-based Physical Synthesis
Complete Logic Synthesis Validation
Run Physical Synthesis (Includes P&R following synthesis)
Log File (.srr) Timing Report (.ta)
Analyze Results
Goals Met?
Yes
Vendor Place and Route
No
Improve Performance

For a flow that includes added performance improvement, see Graph-based Physical Synthesis with Design Planner, below.

See Also

- *Physical Synthesis Task Summary*, on page 1-10 for a summary of tasks required to complete this flow.
- *Run Physical Synthesis*, on page 1-27 to start the physical synthesis design flow.

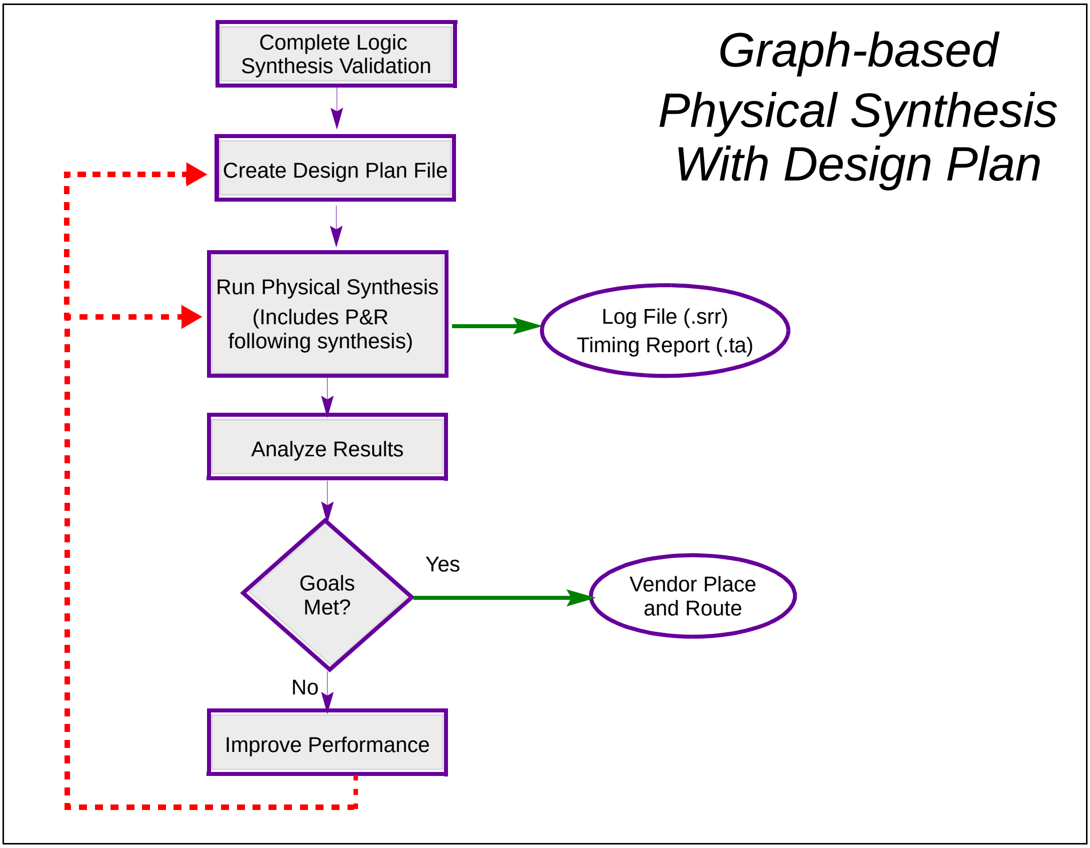
Graph-based Physical Synthesis with Design Planner

This flow is available when you have the Design Plan option with the Synplify Premier tool and can only be used with the Altera Stratix II, Stratix II GX, Stratix III technologies.

This flow is similar to the push-button graph-based physical synthesis flow described in the previous topic, except a design plan file is included to guide global placement. Use this flow to improve performance.

Placement constraints are generated when you assign RTL logic from the RTL view (HDL Analyst) to ports or regions in the Design Plan view (Design Planner). These regions constrain logic to the areas you specify on the device. During optimizations, the regions guide global placement and subsequently influence physical optimizations.

The figure below shows the flow for graph-based physical synthesis using a design plan file.


Graph-based Physical Synthesis With Design Plan
Complete Logic Synthesis Validation
Create Design Plan File
Run Physical Synthesis (Includes P&R following synthesis)
Log File (.srr) Timing Report (.ta)
Analyze Results
Goals Met?
Yes
Vendor Place and Route
No
Improve Performance

Design-plan Based Physical Synthesis

This flow is for Synplify Premier users that have the Design Planner option and use any supported Altera technology.

The flow requires using Design Planner to manually create physical constraints by assigning critical path logic to a specific location on the die to improve performance. Constraints are saved to the design plan file and added to the project to complete physical synthesis for the design. See *Design-Plan Physical Synthesis Flow*, on page 1-35 for details on using this flow.

Physical Synthesis Task Summary

Here is a summary of the tasks for the physical synthesis design flows.

1. Set up Project
   - Create Project
   - Setup Timing and Physical Constraints
   - Set Implementation Options
   - Create Place and Route Implementation
2. Run Logic Synthesis
3. Validate Logic Synthesis Results
4. Create Design Plan File (optional)
5. Run Physical Synthesis
6. Analyze Physical Synthesis Results
7. Improve Performance and Rerun Physical Synthesis, as required

These are the overall tasks to complete physical synthesis.

The remaining sections of this chapter provide details on how to complete these tasks.

If your technology supports only the design-plan based flow, a different set of tasks than those in listed above is required. See *Design-Plan Physical Synthesis Flow*, on page 1-35 for information.

See Also

- *Logic Synthesis Validation Phase*, on page 1-6
- *Graph-based Physical Synthesis*, on page 1-7
- *Graph-based Physical Synthesis with Design Planner*, on page 1-8

Set up Project

Project setup is the first phase of the physical synthesis design process. The project file (`.prj`) is a collection of input files and optimization switches required to synthesize your design. This section contains details on how to setup the file.

First, here are some guidelines to consider before setting up your project:

- Make sure the design is properly constrained. (See *Improve Performance*, on page 1-33 for tips.)
- Make sure to specify IO pin location constraints for all pins in the design for physical synthesis.
- Make sure to include any IO constraints from the Quartus settings file (`.qsf`) as necessary. (Currently, you can successfully translate the IO constraints and IO standards to `.sdc` format – see *Translate Altera QSF Physical Constraints*, on page 1-40.)
- Depending on your target Altera technology, a design plan file (`.sfp`) for physical synthesis is optional. However, to use an `.sfp` file requires the separately-licensed Synplify Premier Design Planner option.
- If you are using one of the graph-based physical synthesis flows, make sure you select a target technology that is supported. See *Supported Altera Devices*, on page 1-4.

The following tasks are required to set up your project for physical synthesis:

- Create Project
- Setup Timing and Physical Constraints
- Set Implementation Options
- Create Place and Route Implementation

See the following subsections for details.

Create Project

To create a project file for physical synthesis:

1. Bring up the Synplify Premier tool.
2. Click on Open Project, then New Project.
3. Click the Add File button and add the following design files:
   - `.v` and/or .vhd (HDL source files).
   - `.sfp` (optional design plan file. You can use this file only if your tool includes the Design Planner option. See *Design Planner,* on page 1-34 for details).
4. Save the project file.

For details on creating a project file, see:

- *Setting Up HDL Source Files,* on page 3-2 of the *User Guide*
- *Setting Up Project Files,* on page 6-2 of the *User Guide*

Setup Timing and Physical Constraints

Constraints for physical synthesis include timing constraints and physical constraints for your design. Timing constraints are used to specify performance goals and describe the design environment. Altera physical constraints, as well as all constraints from the Quartus settings file (QSF) should also be included in the project for physical synthesis. All of these constraints must be read from a single `.sdc` constraint file. To specify timing constraints for the Synplify Premier tool and to translate your Altera constraints (qsf2sdc) to use for physical synthesis, see *Constraints Setup,* on page 1-38.

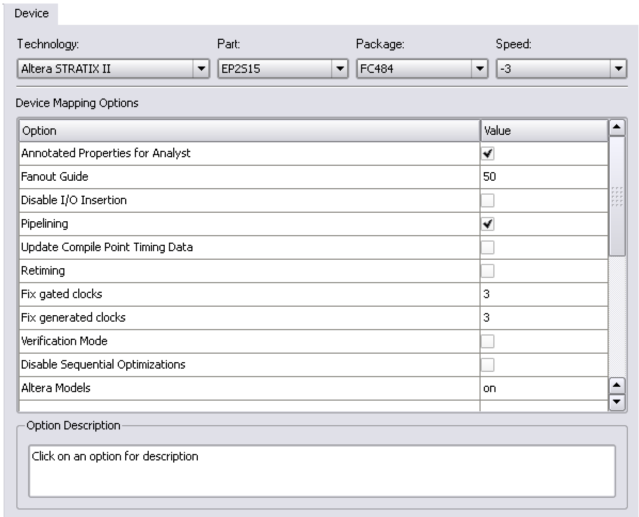
After you have completed the constraint file, add it to the project.

Set Implementation Options

Specify the implementation options for synthesis.

1. Bring up the Implementations Options dialog box (Implementation Options button).

2. In the Device panel, set options for:
   - Technology, part, speed, and package
   - Device mapping options
   - Make sure the Disable I/O Insertion option is not checked. Synplify Premier physical synthesis requires that I/O insertion is enabled.



Device

Technology: Altera STRATIX II

Part: EP2S15

Package: FC484

Speed: -3

Device Mapping Options

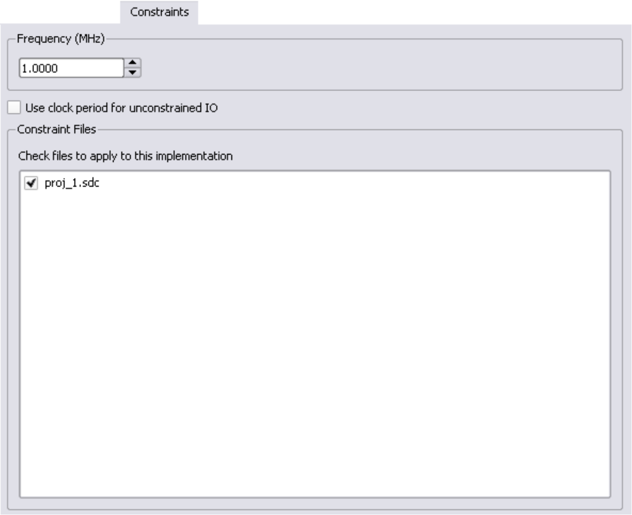
Option	Value
Annotated Properties for Analyst	☑
Fanout Guide	50
Disable I/O Insertion	□
Pipelining	☑
Update Compile Point Timing Data	□
Retiming	□
Fix gated clocks	3
Fix generated clocks	3
Verification Mode	□
Disable Sequential Optimizations	□
Altera Models	on

Option Description

Click on an option for description

3. In the Options panel, set the optimization switches for synthesis. You can use the default switches for physical synthesis, which include the following: FSM Compiler, Resource Sharing, and Pipelining. For descriptions of all of the optimization switches, see *Setting Optimization Options*, on page 6-17 of the *User Guide*.

4. In the Constraints panel:
   - Set an overall target frequency for the design. See *Specifying Global Frequency and Constraint Files*, on page 6-19 of the *User Guide* for information.
   - Make sure the constraint files that you want to use for synthesis are selected.



Constraints

Frequency (MHz)

1.0000

□ Use clock period for unconstrained IO

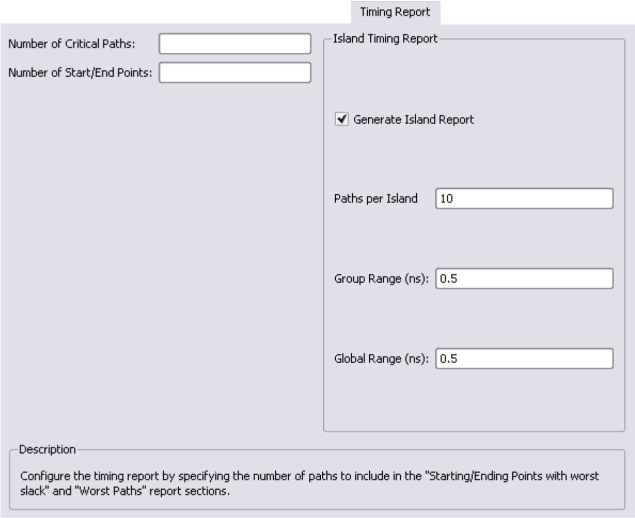
Constraint Files

Check files to apply to this implementation

☑ proj_1.sdc

5. In the Implementation Results panel, specify the output results directory and output file options. See *Specifying Result Options*, on page 6-21 of the *User Guide* for details.

6. In the Timing Report panel specify:
   - Number of critical paths and start/end points to display in the timing report.
   - Island timing report; parameters to use for the report. For details, see *Generating the Island Timing Report Automatically*, on page 14-14 of the *User Guide*.



Timing Report

Number of Critical Paths:

Number of Start/End Points:

Island Timing Report

☑ Generate Island Report

Paths per Island 10

Group Range (ns): 0.5

Global Range (ns): 0.5

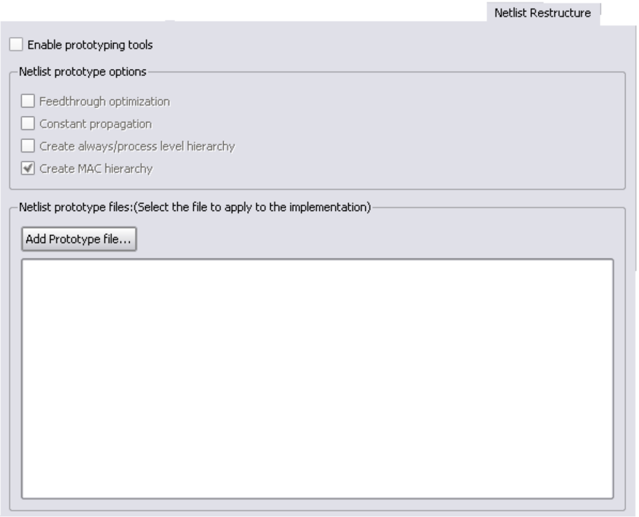
Description

Configure the timing report by specifying the number of paths to include in the "Starting/Ending Points with worst slack" and "Worst Paths" report sections.

You can also change timing parameters and rerun the timing analyzer after synthesis. See *Using the Island Timing Analyst*, on page 14-12 of the *User Guide*.

7. In the Verilog/VHDL panel, specify the desired HDL options. See *Setting Verilog and VHDL Options*, on page 6-24 of the *User Guide*.

8. Specify options, as appropriate, in the Netlist Restructure panel for:
   - Any necessary netlist optimizations.
   - Netlist restructure file (`.nrf`) for which bit slicing or zippering might have been performed.



Netlist Restructure

□ Enable prototyping tools

Netlist prototype options

□ Feedthrough optimization

□ Constant propagation

□ Create always/process level hierarchy

☑ Create MAC hierarchy

Netlist prototype files:(Select the file to apply to the implementation)

See *Setting Synplify Premier Netlist Restructuring Optimizations, on page 7-3* of the *User Guide* for descriptions of these switches.

9. Click OK to apply the implementation options.

10. Save the project file.

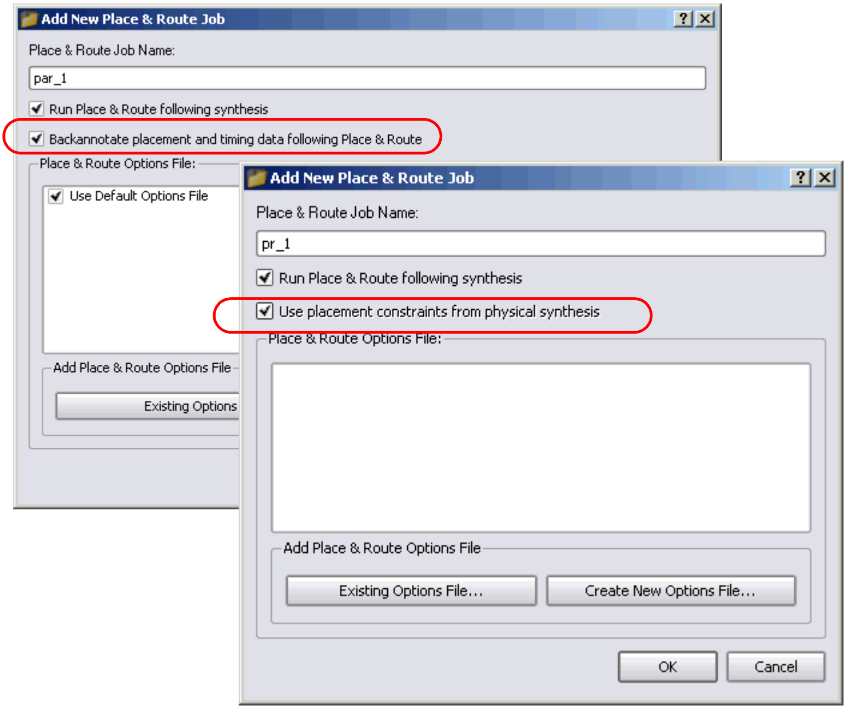
Create Place and Route Implementation

You can set up an implementation to run Altera Quartus place and route after synthesis completes. To do this:

1. Make sure you are using the correct Quartus version for your tool. See *Supported Altera Place and Route Tools*, on page 1-4 for information.

2. Set the QUARTUS_ROOTDIR and PATH environment variables to point to a valid installation of the place and route tool.

3. From the Project view, click on the Add P&R Implementation button.



Add New Place & Route Job

Place & Route Job Name: par_1

☑ Run Place & Route following synthesis

☑ Backannotate placement and timing data following Place & Route

Place & Route Options File:

☑ Use Default Options File

Add New Place & Route Job

Place & Route Job Name: pr_1

☑ Run Place & Route following synthesis

☑ Use placement constraints from physical synthesis

Place & Route Options File:

4. Specify the Place & Route Job Name. Default is par_*n*.

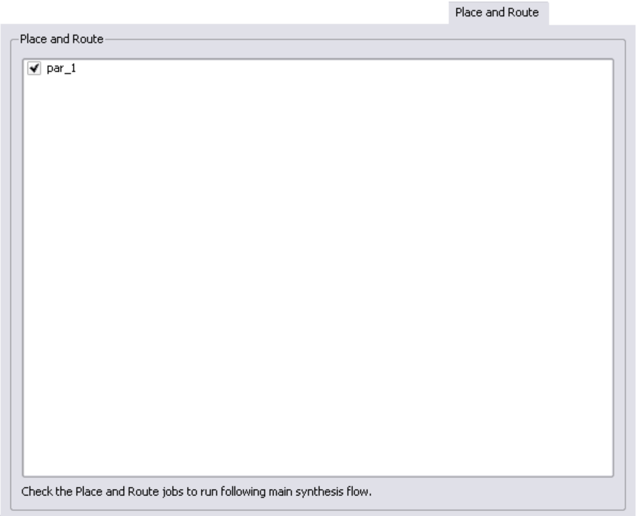
5. Make sure the Run Place and Route following synthesis switch is enabled.

6. Specify the place-and-route options file. The tool automatically uses default options located in

   <*install_directory*>\lib\altera\altera_par.tcl

You can change or override the default options. See *Specifying Altera Place-and-Route Options, on page 1-18* for details.

7. You can choose to:
   - Backannotate placement and timing data following Place & Route for certain Altera technologies. See *Backannotating Place-and-Route Data, on page 1-21* for details.
   - Use placement constraints from physical synthesis for Altera Stratix III, Stratix II GX, and Stratix II devices. See *Forward Annotating Physical Constraints, on page 1-22* for details.
8. Enable the P&R implementation, if not already done, to use (Implementation Options->Place and Route tab).



☑ par_1

Check the Place and Route jobs to run following main synthesis flow.

9. Save the project file.

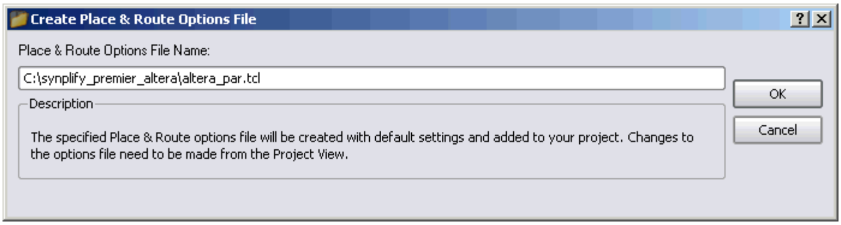
Specifying Altera Place-and-Route Options

This section shows you how to customize your Altera place-and-route run by specifying a place-and-route options file or .tcl script. You can use either the default file or create a custom file.

1. To use the default place-and-route options, click the Add P&R Implementation button in the Project view and select Use Default Options File in the dialog box. Click OK.

The software uses the options in the altera_par.tcl file which is located in the installation directory.

2. To use an existing options file (.tcl script):
   - Click the Add P&R Implementation button in the Project view.
   - Click Existing Options File. Select the file name in the next dialog box, and click Open.
   - Return to the Add New Place & Route Job dialog box and make sure the correct options file is selected. Click OK.
3. To create a new place-and-route options file:
   - Click the Add P&R Implementation button in the Project view. In the dialog box, click Create New Options File. Specify the file name in the next dialog box, and click OK.

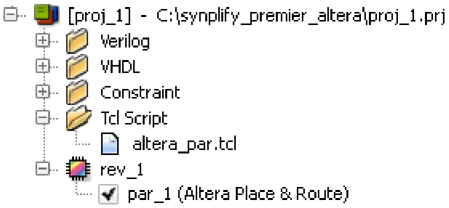


Place & Route Options File Name: C:\synplify_premier_altera\altera_par.tcl

Description

The specified Place & Route options file will be created with default settings and added to your project. Changes to the options file need to be made from the Project View.

A text window opens with the default options file. This file is automatically added to the project.



```
[proj_1] - C:\synplify_premier_altera\proj_1.prj
    Verilog
    VHDL
    Constraint
    Tcl Script
        altera_par.tcl
    rev_1
        par_1 (Altera Place & Route)
```

   - Edit the default options to customize this options file. For more information about the contents of this file, see *Customizing the Altera Place-and-Route Options File*, on page 1-20.
   - Save the file.
   - Return to the Add New Place & Route Job dialog box, and make sure the options file you created is selected.

- Select Run Place & Route following synthesis. Click OK.

  The software uses the options file to place and route the design after synthesis.

4. View the results.

   - Select the P&R implementation in the Project view. The result files are displayed in the Implementation Results view.
   - View the log file quartus.log for information about the run.

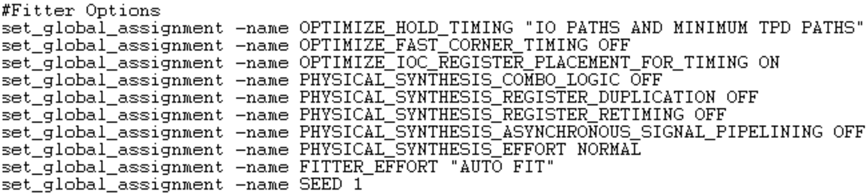
Customizing the Altera Place-and-Route Options File

To customize the Altera place-and-route options file, you can edit the default options file (altera_par.tcl). This file contains the options for the following place-and-route processes:

- Fitter Options
- Timing Analyzer Options
- Analysis & Synthesis Options

Fitter Options

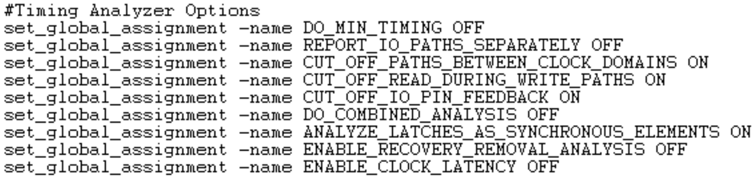
Edit the following default fitter options for the Quartus process shown below.



```
#Fitter Options
set_global_assignment -name OPTIMIZE_HOLD_TIMING "IO PATHS AND MINIMUM TPD PATHS"
set_global_assignment -name OPTIMIZE_FAST_CORNER_TIMING OFF
set_global_assignment -name OPTIMIZE_IOC_REGISTER_PLACEMENT_FOR_TIMING ON
set_global_assignment -name PHYSICAL_SYNTHESIS_COMBO_LOGIC OFF
set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION OFF
set_global_assignment -name PHYSICAL_SYNTHESIS_REGISTER_RETIMING OFF
set_global_assignment -name PHYSICAL_SYNTHESIS_ASYNCHRONOUS_SIGNAL_PIPELINING OFF
set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT NORMAL
set_global_assignment -name FITTER_EFFORT "AUTO FIT"
set_global_assignment -name SEED 1
```

Timing Analyzer Options

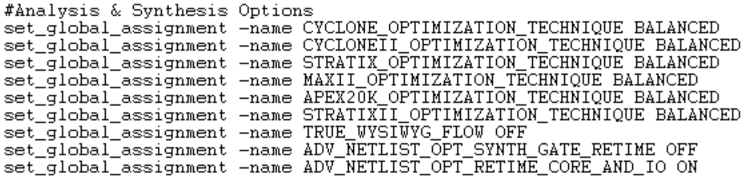
Edit the following default timing analyzer options for the Quartus process shown below.



```
#Timing Analyzer Options
set_global_assignment -name DO_MIN_TIMING OFF
set_global_assignment -name REPORT_IO_PATHS_SEPARATELY OFF
set_global_assignment -name CUT_OFF_PATHS_BETWEEN_CLOCK_DOMAINS ON
set_global_assignment -name CUT_OFF_READ_DURING_WRITE_PATHS ON
set_global_assignment -name CUT_OFF_IO_PIN_FEEDBACK ON
set_global_assignment -name DO_COMBINED_ANALYSIS OFF
set_global_assignment -name ANALYZE_LATCHES_AS_SYNCHRONOUS_ELEMENTS ON
set_global_assignment -name ENABLE_RECOVERY_REMOVAL_ANALYSIS OFF
set_global_assignment -name ENABLE_CLOCK_LATENCY OFF
```

Analysis & Synthesis Options

Edit the following default analysis and synthesis options for the Quartus process shown below.



```
#Analysis & Synthesis Options
set_global_assignment -name CYCLONE_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name CYCLONEII_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name STRATIX_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name MAXII_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name APEX20K_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name STRATIXII_OPTIMIZATION_TECHNIQUE BALANCED
set_global_assignment -name TRUE_WYSIWYG_FLOW OFF
set_global_assignment -name ADV_NETLIST_OPT_SYNTH_GATE_RETIME OFF
set_global_assignment -name ADV_NETLIST_OPT_RETIME_CORE_AND_IO ON
```

Backannotating Place-and-Route Data

You can also choose to backannotate place-and-route data which provides accurate timing and placement information during physical synthesis. However, this option is only applicable for certain Altera technologies.

Use the following procedure to backannotate place-and-route data.

To create a new place-and-route implementation:

1. Click on the Add P&R Implementation button from the Project view or right-click and select Add Place & Route Job from the popup menu.
2. On the Add New Place & Route Job dialog box, enable the Backannotate placement and timing data following Place & Route option.

For an existing place-and-route implementation:

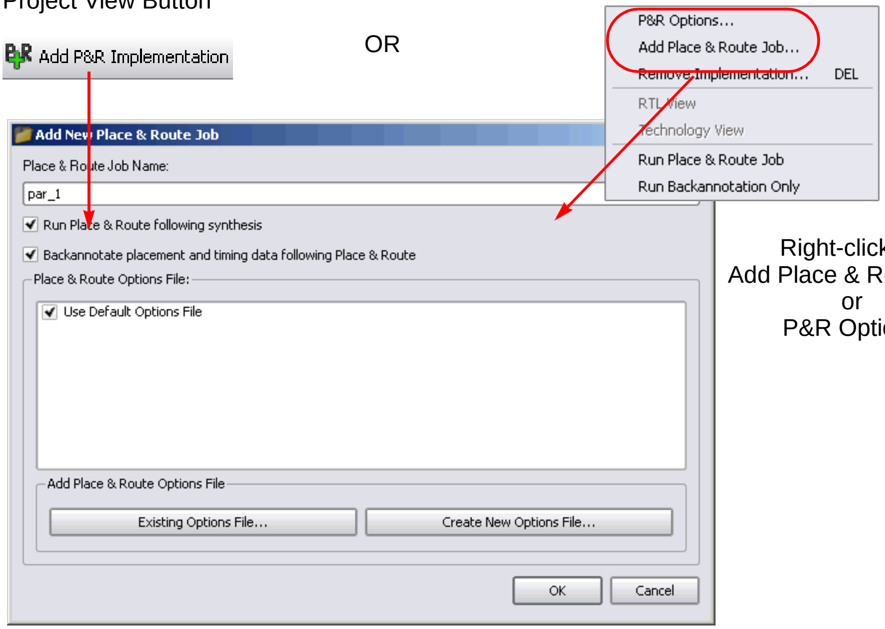
1. In the Project view, select the place-and-route implementation, then right-click and select Place & Route Options from the popup menu.
2. Enable the Backannotate placement and timing data following Place & Route option from the popup dialog box.

Note: Back-annotation should not be performed on designs containing IP cores.

Project View Button

OR

Implementation Pop-up Menu



Place & Route Job Name: par_1

☑ Run Place & Route following synthesis

☑ Backannotate placement and timing data following Place & Route

Place & Route Options File:

☑ Use Default Options File

Add Place & Route Options File

Right-click on
Add Place & Route Job
or
P&R Options

Forward Annotating Physical Constraints

As an alternative flow, you can choose to forward annotate physical constraint from the Synplify Premier physical synthesis tool, or else, let the Quartus II place-and-route tool determine how to handle the physical constraints. However, this option is only available for Altera Stratix III, Stratix II GX, or Stratix II devices. To do this, use the following procedure.

To create a new place-and-route implementation:

1. Click on the Add P&R Implementation button from the Project view or right-click and select Add Place & Route Job from the popup menu.

2. On the Add New Place & Route Job dialog box, enable or disable the Use placement constraints from physical synthesis option. By default, this option is enabled.

Note: The Physical Synthesis switch must be enabled to use this option. When this option is disabled, physical synthesis optimizations are performed but placement constraints will not be forward annotated.

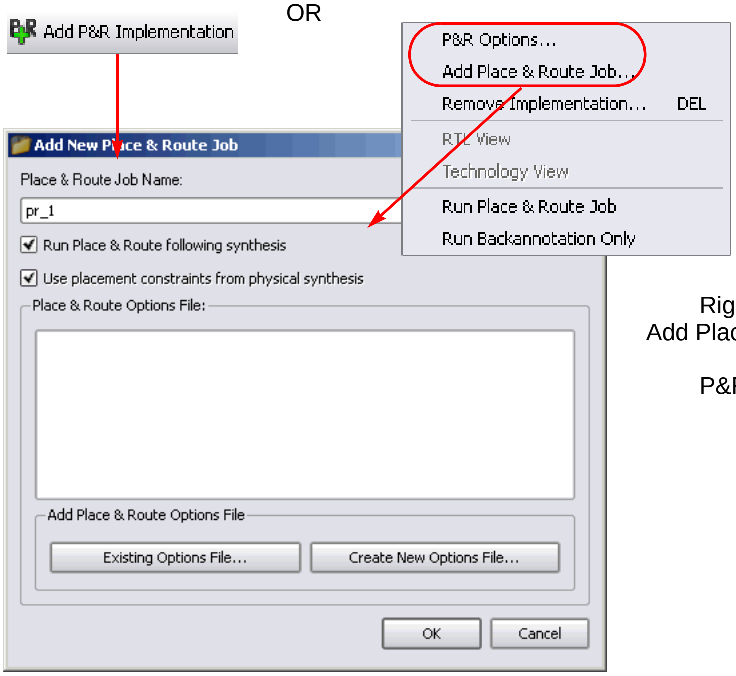
For an existing place-and-route implementation:

1. In the Project view, select the place-and-route implementation, then right-click and select Place & Route Options from the popup menu.
2. Enable or disable the Use placement constraints from physical synthesis option from the popup dialog box. By default, this option is enabled.

Project View Button

OR

Implementation Pop-up Menu



Place & Route Job Name: pr_1

☑ Run Place & Route following synthesis

☑ Use placement constraints from physical synthesis

Right-click on
Add Place and Route Job
or
P&R Options

Run Logic Synthesis

If this is the first time you are running synthesis on the design, run in logic synthesis mode. This means the Physical Synthesis switch is disabled. The initial synthesis run is to determine if there are any problems that need to be addressed before going on to the physical synthesis stage.

1. Before running logic synthesis, the following phases must be complete:
   - Create the project – *Create Project*, on page 1-12.
   - Specify constraints – *Setup Timing and Physical Constraints*, on page 1-12.
   - Set implementation options – *Set Implementation Options*, on page 1-12.
   - Setup for place and route – *Create Place and Route Implementation*, on page 1-17.
2. Disable the Physical Synthesis switch:
   - Located in Project view

     or
   - From Implementation Options->Options.
3. Click the Run button.

   The Synplify Premier tool goes through Compiling and Mapping phases. When physical synthesis completes, the place and route implementation that was set up in the project file is also run. When the job completes, Done! (or Warnings!) displays in the Project view. Output results files are shown in the right pane of the Project view. Double-click on the files to display them.
4. Go on to the next phase, *Validate Logic Synthesis Results*, on page 1-25.

Validate Logic Synthesis Results

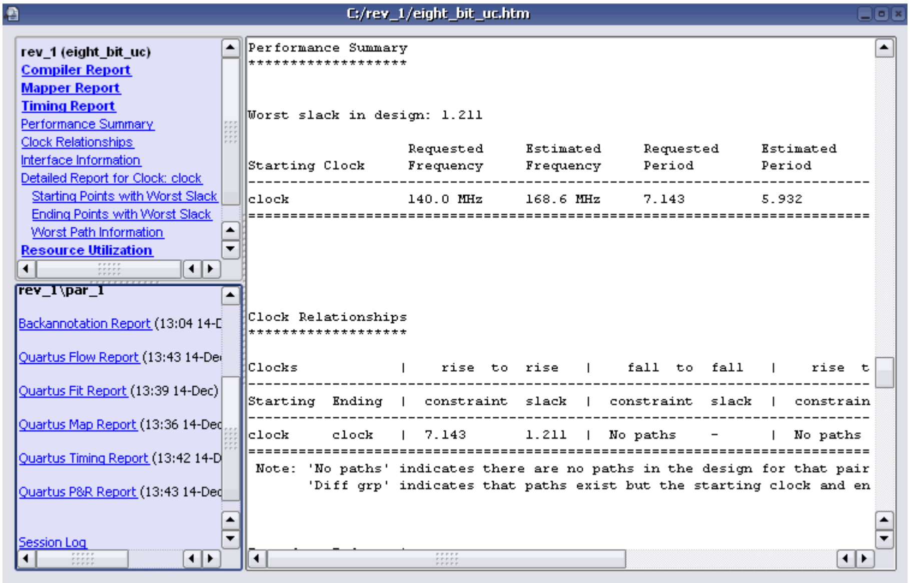
Check initial logic synthesis results. Topics in this section include:

- Using the Log File
- Validate Results

Using the Log File

The log file contains default timing and area reports. This section provides the steps required to validate your results.

Click the View Log button in the Project view to display the log file in either text (`.srr`) or HTML (`.htm`) format.


C:/rev_1/eight_bit_uc.htm
rev_1 (eight_bit_uc)
Compiler Report
Mapper Report
Timing Report
Performance Summary
Clock Relationships
Interface Information
Detailed Report for Clock: clock
Starting Points with Worst Slack
Ending Points with Worst Slack
Worst Path Information
Resource Utilization
rev_1\par_1
Backannotation Report (13:04 14-D
Quartus Flow Report (13:43 14-De
Quartus Fit Report (13:39 14-Dec)
Quartus Map Report (13:36 14-Dec
Quartus Timing Report (13:42 14-D
Quartus P&R Report (13:43 14-Dec
Session Log

```
Performance Summary
*******************


Worst slack in design: 1.211

                   Requested     Estimated     Requested     Estimated
Starting Clock     Frequency     Frequency     Period        Period
---------------------------------------------------------------------------
clock              140.0 MHz     168.6 MHz     7.143         5.932
===========================================================================




Clock Relationships
*******************

Clocks            |    rise  to  rise   |    fall  to  fall   |    rise  t
---------------------------------------------------------------------------
Starting  Ending  |  constraint  slack  |  constraint  slack  |  constrain
---------------------------------------------------------------------------
clock     clock   |  7.143       1.211  |  No paths    -      |  No paths
===========================================================================
 Note: 'No paths' indicates there are no paths in the design for that pair
       'Diff grp' indicates that paths exist but the starting clock and en
```

Validate Results

Use the following guidelines to validate your results:

1. Was the logic synthesis run successful? (Physical Synthesis switch disabled; successful logic synthesis and place-and-route?)

2. Did you use the correct PAR Quartus version for your tool? (See the Release Notes, Help->Online Documents->release_notes.pdf->*Third Party Tool Versions*).

3. Are there black boxes in the design? Search the synthesis `.srr` log file for black box.

   A design that contains black boxes errors out in the tool and should be eliminated from the design.

4. Are there any combinational feedback loops? Search the synthesis `.srr` log file for:

   ```
   Found combinational loop
   ```

   Combinational loops cause random timing analysis results that invalidate any comparison and should be eliminated from the design.

5. Are the clock constraints correct? Check the Clock Relationships table in the `.srr` log file.

6. Are the forward annotated timing constraints (TCL) consistent with the post place-and-route timing constraints?

7. Are the false and multi-cycle paths constraints correctly defined in the `.sdc` file? Ensure that the back-annotation timing report (`.srr` log file in the PAR directory) matches the `.tan.rpt` file.

8. Are the clocks routed on global resources? Check the Clock Path Skew numbers in the `.tan.rpt` file. Clocks routed on general routing resources usually result in large skews. Because the tool does not take clock skew into account, large skews can degrade the quality of results (QoR) and result in poor timing correlation.

See Also

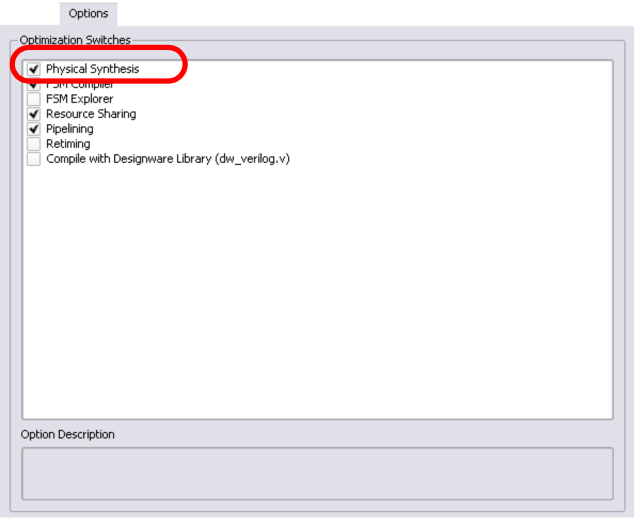
- *Analyze Physical Synthesis Results*, on page 1-30

Run Physical Synthesis

Once you complete the logic synthesis validation phase (see *Logic Synthesis Validation Phase*, on page 1-6 for details), you are ready to run physical synthesis. The project you created for logic synthesis (*Set up Project*, on page 1-11) requires these additional steps:

1. Enable the Physical Synthesis switch, located either:
   - in the Project view

     or
   - Implementation Option ->Options tab



Options

Optimization Switches

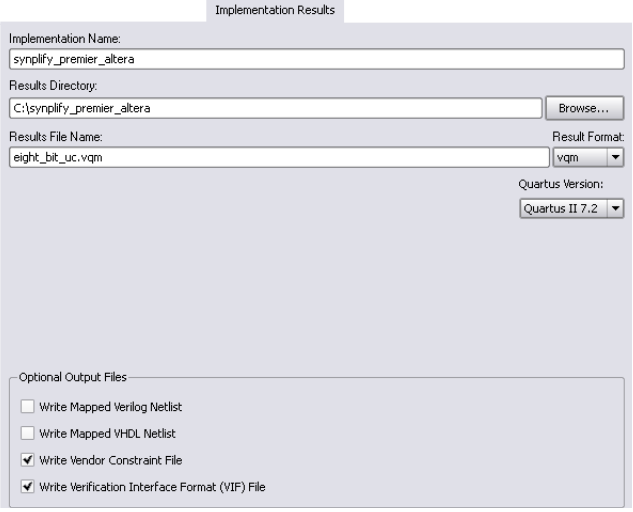
☑ Physical Synthesis
☑ FSM Compiler
☐ FSM Explorer
☑ Resource Sharing
☑ Pipelining
☐ Retiming
☐ Compile with Designware Library (dw_verilog.v)

Option Description

2. If you want a different directory for your physical synthesis results, click on the Implementation Results tab and specify the result options.

   When your design contains LPMs or Megafunctions, make sure that you select the appropriate Quartus Version from the pull-down menu on this panel so graph-based physical synthesis implements these functions correctly.

   See *Using Altera LPMs or Megafunctions in Synthesis*, on page 4-44 and *Supported Altera Place and Route Tools*, on page 1-4 of the *User Guide* for more information.



Implementation Results

Implementation Name: synplify_premier_altera

Results Directory: C:\synplify_premier_altera

Results File Name: eight_bit_uc.vqm

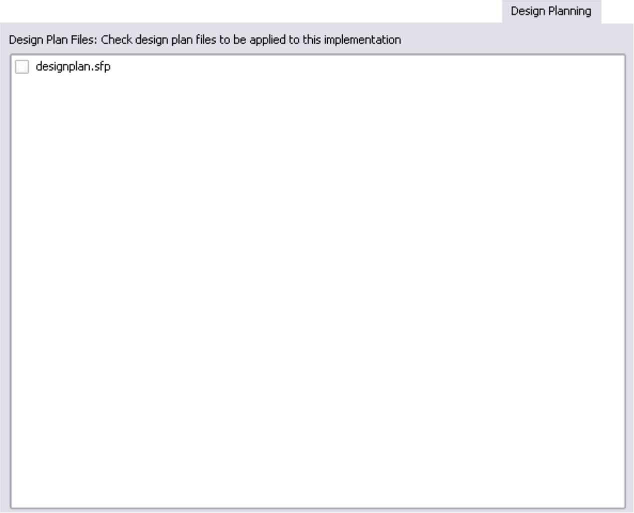
Result Format: vqm

Quartus Version: Quartus II 7.2

Optional Output Files

- ☐ Write Mapped Verilog Netlist
- ☐ Write Mapped VHDL Netlist
- ☑ Write Vendor Constraint File
- ☑ Write Verification Interface Format (VIF) File

3. If you are using Design Planner, click on the Design Planning tab and enable the desired design plan file (`.sfp`) if needed.



Design Planning

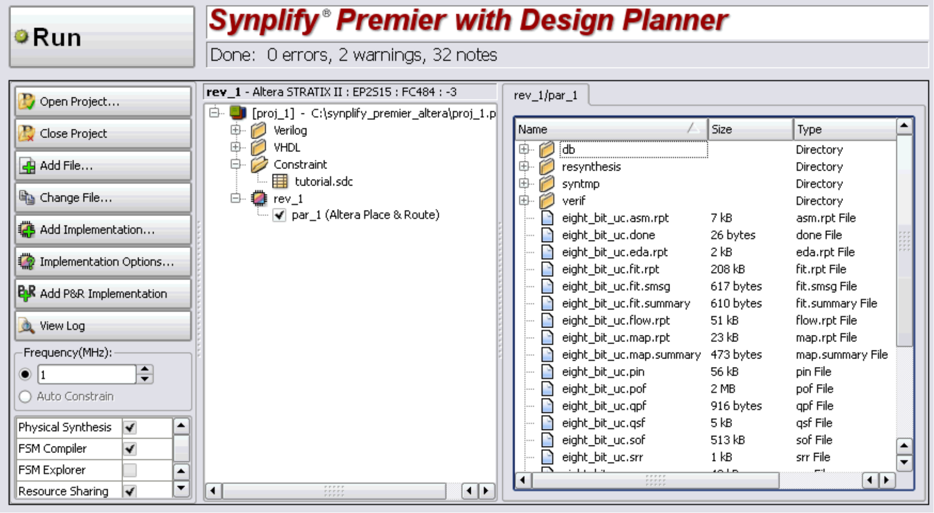
Design Plan Files: Check design plan files to be applied to this implementation

- ☐ designplan.sfp

Note: You do not need to select a design plan file to run graph-based physical synthesis. However, if you are using a graph-based flow and want to use a design plan file, see *Create Design Plan File, on* page 1-42.
For older Altera technologies, you must create a design plan (`.sfp`) to run physical synthesis. See *Design Planner, on* page 1-34 for more information.

4. Click OK to apply the implementation options.
5. Make sure the place-and-route implementation is enabled, Implementation Options->Place and Route tab.
6. Click Run in the Project view.

   Optimizations are performed on the design using placement-aware synthesis. Synthesis and placement are integrated by performing concurrent placement and optimization based on timing constraints and device technology.



Done: 0 errors, 2 warnings, 32 notes

rev_1 - Altera STRATIX II : EP2S15 : FC484 : -3

```
[proj_1] - C:\synplify_premier_altera\proj_1.p
   Verilog
   VHDL
   Constraint
      tutorial.sdc
   rev_1
      ☑ par_1 (Altera Place & Route)
```

Frequency(MHz): ● 1
○ Auto Constrain

Physical Synthesis	☑
FSM Compiler	☑
FSM Explorer	□
Resource Sharing	☑

Name	Size	Type
db		Directory
resynthesis		Directory
syntmp		Directory
verif		Directory
eight_bit_uc.asm.rpt	7 kB	asm.rpt File
eight_bit_uc.done	26 bytes	done File
eight_bit_uc.eda.rpt	2 kB	eda.rpt File
eight_bit_uc.fit.rpt	208 kB	fit.rpt File
eight_bit_uc.fit.smsg	617 bytes	fit.smsg File
eight_bit_uc.fit.summary	610 bytes	fit.summary File
eight_bit_uc.flow.rpt	51 kB	flow.rpt File
eight_bit_uc.map.rpt	23 kB	map.rpt File
eight_bit_uc.map.summary	473 bytes	map.summary File
eight_bit_uc.pin	56 kB	pin File
eight_bit_uc.pof	2 MB	pof File
eight_bit_uc.qpf	916 bytes	qpf File
eight_bit_uc.qsf	5 kB	qsf File
eight_bit_uc.sof	513 kB	sof File
eight_bit_uc.srr	1 kB	srr File

7. Analyze results. See *Analyze Physical Synthesis Results*, next for details.

Analyze Physical Synthesis Results

To determine if your design has met performance goals, use the Synplify Premier analysis tools, which include:

- Log file (`.srr` or `.htm`), includes the default timing report
- HDL Analyst:
  - RTL View (⊕)
  - Technology View ()
- Physical Analyst ( )
- Timing Report ()

Use these tools to analyze the critical path(s) with negative slack and identify potential solutions to improve performance. Use Design Planner to create physical constraints to also aid in improving performance ().

Here are some guidelines for analyzing results:

1. Are start and end points being constrained by the proper clocks?
   The timing report is the primary tool for checking this. You can also trace the clock network using HDL Analyst Technology view.
2. Is the critical path a multi-cycle path or false path?
   Use the timing report and HDL Analyst tool to get best view of the design's timing.
3. Can pipelining be used to close timing?
   Use the HDL Analyst tool; also, see *Pipelining,* on page 9-5 of the *User Guide* for details on this feature.
4. If the path is inside a state machine, is the FSM being fully optimized?
   Use the HDL Analyst. Open the RTL view and push down into the state machine module to display the FSM viewer.
5. Look at the timing report that provides the % breakdown of delay for each path (Do a find on "Total path delay"). Are the net delays contributing to the highest percentage on the critical path?
   Use the Physical Analyst to analyze the instance placement of the critical path.

6. Can performance be improved using a physical constraints file? Use Physical Analyst and Design Planner to determine if constraining logic to specific regions can provide improved performance.

For more details and guidelines on improving design performance, see *Improve Performance,* on page 1-33.

See Also

- *Using the Log File,* on page 1-25
- *Validate Results,* on page 1-26

HDL Analyst

The RTL and Technology views provide schematics to analyze the design.

Display the RTL schematic for a compiled design (compile phase complete only). Select HDL Analyst->RTL->Hierarchical View or ->Flattened View.

Display the Technology schematic for a synthesized design (technology mapping complete). Select HDL Analyst ->Technology->Hierarchical View, or ->Flattened View.

For an overview of using the HDL Analyst views, see:

- *Basic Operations on Schematic Objects,* on page 6-13 of the *User Guide*
- *Exploring Design Hierarchy,* on page 6-21 of the *User Guide*
- *Finding Schematic Objects,* on page 6-13 of the *User Guide*
- *HDL Analyst Views and Commands,* on page 6-2 of the *User Guide*

Stand-alone Timing Analyst

You can run the stand-alone timing analyzer to produce a timing report (`.ta`) that displays more or less information than the default timing report in the log file (`.srr`). Use the stand-alone timing analyzer for your more specific report requirements.

- Select Analysis->Timing Analyst.
- Fill in the parameters for the report (see *Timing Report Generation Parameters*, on page 3-92 for details on completing the fields).
- Click Generate to run the report.

For more information, see *Using the Stand-alone Timing Analyst*, on page 14-6 of the *User Guide*.

Physical Analyst

The Physical Analyst provides a visual display of the device and design placement. Select HDL Analyst->Physical Analyst. The Physical Analyst view can display instances and nets. For complete details, see Chapter 13, *Analyzing Designs in Physical Analyst*.

You can also use the Physical Analyst tool to analyze performance during the RTL physical synthesis phase. For more information, see *Intermediate File for Debugging RTL Physical Synthesis*, on page 1-33.

Improve Performance

The Synplify Premier tool is timing-driven; optimizations depend on timing constraints and are applied until all constraints are met. Therefore, it is very important that you adequately apply timing constraints and not over-constrain the tool. This section includes guidelines for applying constraints.

- Verify constraints consistency between synthesis and P&R:
  - Clock constraints
  - Clock-to-clock constraints
  - IO delays
  - IO standard, drive, slew and pull-up/pull-down
  - Multi-cycle and false paths
  - Max-delay paths
  - DCM parameters
  - Register packing into IOB
  - SYN_LOC on IO pins and pad types
  - Placement constraints on instances
- Ensure the final physical synthesis slack is negative, but no more than 10-15% of the clock constraint.

Intermediate File for Debugging RTL Physical Synthesis

Graph-based physical synthesis generates an intermediate file that you can display in the Technology View and use for debugging.

The `preplace.srm` intermediate file captures the netlist after RTL physical synthesis and immediately before global placement, showing the same results as would be obtained from logic synthesis. This file is written to the physical synthesis implementation results directory.

To display the `preplace.srm` file in the Technology View, double-click or right-click on the file in the Project view, then select Open. You can also find the corresponding timing slack for all the clocks in the design in the Pre-placement Timing Snapshot section of the log file for the critical path reflected in the `preplace.srm` file.

Design Planner

If you have the Synplify Premier tool with the Design Planner option you can specify placement constraints to guide global placement. Physical constraints, specified in a Design Plan file (`.sfp`), constrain logic to specified regions on the device. The `.sfp` file also serves as a guide for initial placement for the following objects types:

- - I/Os
- - RAMs
- - ROMs
- - DSPs
- - clock pins

Note: For Altera technologies that support graph-based physical synthesis, the `.sfp` file is not required because you can run physical synthesis in a single-pass flow without the need for a `.sfp` file.

See *Supported Altera Devices,* on page 1-4 to determine the physical synthesis flows that can be used with the specific Altera technologies and consider using the design planner option if:

- You are using a Altera technology that does not support graph-based physical synthesis thus requiring the design-plan based physical synthesis flow. In a design-plan based flow, effective manual placement of the critical path is required to improve the design using placement-based optimization, register replication for high fanouts, and register tunneling across region boundaries. See *Design-Plan Physical Synthesis Flow,* on page 1-35 for details.
- You are using a Altera technology that supports graph-based physical synthesis, but you want to apply physical constraints to guide global placement.

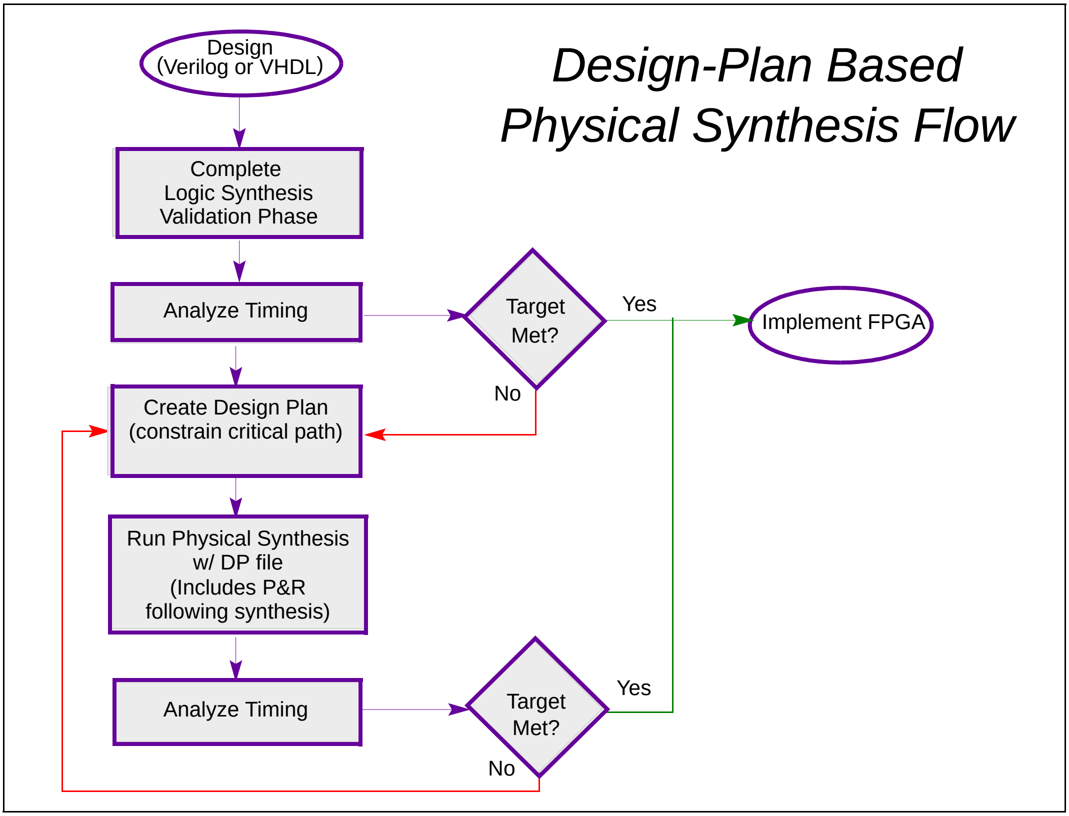
Design-Plan Physical Synthesis Flow

Use this flow if you have the Synplify Premier tool with the Design Planner option and are using any of the following Altera technologies:

– Cyclone
– Cyclone II
– Stratix
– Stratix GX

For all other Altera technologies, see *Supported Altera Devices*, on page 1-4 for the flow to use with your technology.

The figure below shows the design-plan based flow; accompanying task descriptions follow the flow diagram.


Design-Plan Based Physical Synthesis Flow
Design (Verilog or VHDL)
Complete Logic Synthesis Validation Phase
Analyze Timing
Target Met?
Yes
No
Implement FPGA
Create Design Plan (constrain critical path)
Run Physical Synthesis w/ DP file (Includes P&R following synthesis)
Analyze Timing
Target Met?
Yes
No

Design Tasks

These tasks reflect the flow diagram above:

1. Synthesize the design in logic synthesis mode—using timing constraints and no physical constraints. This phase is to determine if the design can successfully complete synthesis and if timing performance enhancements are needed. The logic synthesis validation phase includes running the netlist through place-and-route after synthesis completes. For details on how to complete this phase, see:

   – *Logic Synthesis Validation Phase,* on page 1-6

   – *Set up Project,* on page 1-11

2. Analyze timing results. See *Validate Logic Synthesis Results,* on page 1-25 for details.

   If timing goals are met, you are done. Otherwise, go to the next step.

3. Determine the critical paths from PAR; these are the candidates for logic assignments to regions.

4. Bring up the Design Planner ( ) and:

   – Create regions for the critical paths and interactively assign the critical paths to regions of the chip. See *Working with Regions,* on page 10-19 of the *User Guide* for details.

   – Obtain a size estimation for each RTL block in the design. See *Checking Utilization,* on page 10-29 of the *User Guide* for details.

   – For multiple clocks, assign critical logic associated with each clock domain (that does not meet design requirements) to a unique region to avoid resource contention.

   – If you have any black boxes in your design, assign them to a region. Designate this region as an IP block, so that the Synplify Premier software can instantiate the black box in the `.vqm` file. However, you must provide the content for the black box so that the place-and-route tool can run successfully.

   You can also bring up Physical Analyst to view the design and critical path placement.

   Consult the following sections of the *User Guide* for more information on how to complete the Design Plan file (`.sfp`).

  - *Creating and Using a Design Plan File for Physical Synthesis*, on page 10-8
  - *Working with Regions*, on page 10-19
  - *Assigning Pins and Clocks*, on page 10-9

5. Save the design plan file (`.sfp`) and add it to your project.

6. Run physical synthesis. Use the same project file that you created in step 1 above. This time enable the Physical Synthesis switch and include the physical constraints file (`.sfp`). This phase also includes running the netlist through place-and-route after synthesis completes.

7. Analyze the timing in the Synplify Premier tool. Use the log file and graphical analysis tools. See *Analyze Physical Synthesis Results*, on page 1-30 for details.

   If the target is met, you can continue to the next design phase. If not, you should re-evaluate timing and placement. Perhaps there is a new critical path or the one that is already assigned to regions needs tweaking. See *Improve Performance*, on page 1-33 for more suggestions.

Additional Topics

This section includes the following additional topics that are related to running physical synthesis:

- Constraints Setup
- Create Design Plan File
- Performance Results Comparison
- Running Multiple Implementations

Constraints Setup

This section provides information on defining timing and physical constraints and attributes for synthesis. Topics include:

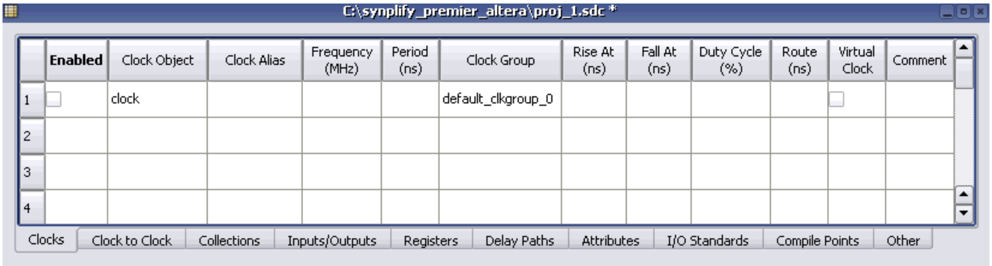
- Create the Timing Constraint File (.sdc)
- Translate Altera QSF Physical Constraints
- Run Constraint Check

Create the Timing Constraint File (.sdc)

Use timing constraints to specify performance goals for the design and describe the environment. You can specify constraints in a text file using any text editor, however it is easier to use the SCOPE GUI (Synplicity synthesis constraint optimization environment), a spreadsheet-like interface that automatically formats the constraints you define. To create the constraints file:

1. Create the project file (see *Create Project,* on page 1-12).
2. Compile the design (Run->Compile Only). This can be done after adding the source files to the project.
3. Open an .sdc file:
   - Click the New Constraint file (SCOPE) icon in the toolbar. (▦)
   - Click OK to Initialize Constraints.

The SCOPE window opens with the design objects initialized, such as clock frequency, ports, input/outputs. This means you can use the pull-downs in the SCOPE columns to fill in object names and types when specifying constraints.



	Enabled	Clock Object	Clock Alias	Frequency (MHz)	Period (ns)	Clock Group	Rise At (ns)	Fall At (ns)	Duty Cycle (%)	Route (ns)	Virtual Clock	Comment
1	☐	clock				default_clkgroup_0					☐	
2												
3												
4												

Specify the desired constraints for the design. See *Using the SCOPE UI,* on page 5-2 of the *User Guide* for descriptions of the constraints and complete details on creating constraint files.

4. Save the file; click Yes in the dialog box to add the file to your project.

5. Save the project file.

See Also

- *Translate Altera QSF Physical Constraints*, next, for information on how to complete the `.sdc` file by adding the QSF constraints for physical synthesis.

Translate Altera QSF Physical Constraints

A unified set of timing and physical constraints is required to run physical synthesis. Earlier, you created Synplify Premier timing constraints using the SCOPE editor. This section describes how to use the qsf2sdc utility to translate constraints in your Altera QSF to .sdc format. This ensures that QSF timing and physical constraints are also honored during physical synthesis.

You can run the utility from any shell window; qsf2sdc is located in the bin directory where your software is installed:

```
<install_dir>/bin
```

To convert QSF constraints:

1. Run qsf2sdc using the following syntax:

   ```
   <install_dir>/bin/qsf2sdc -iqsf <constraints_file>.qsf
        -osdc <constraints_file>.sdc
        [-oqsf <residual_constraints_file>.qsf] [-all]
        [-silent]
   ```

2. After translating the constraints, edit the new .sdc file.

   The translator converts the most common timing and physical constraints. However, because of the diversity and complexity of QSF format, the resulting .sdc file requires manual intervention. To do this:

   - Visually inspect the translated file.

     The original QSF commands are written as comments in the new .sdc file so that you can validate the translated constraints. Constraints which were successfully translated are specified as Supported. However, constraints which were unsuccessfully translated are specified as Unsupported.

     Use the -silent option to suppress all the #Supported and #Unsupported messages in the .sdc file.

   - Manually edit the SDC file to complete the translation of constraints, as necessary.
   - Optionally, use the -all option to convert any instances with location assignments. By default, only pin location assignments and IO standards are automatically converted.

   For reference information on qsf2sdc, *see Altera qsf2sdc Utility, on page 12-33* (*Reference Manual*).

3. To run physical synthesis, timing and physical constraints can be combined into one `.sdc` file. Include the timing constraints (created in *Create the Timing Constraint File (.sdc)*, on page 1-38) into the `.sdc` file containing the translated physical constraints. Make sure that all of the following types of constraints are combined into the `.sdc` file:

   – Timing Constraints:
   Clock
   Clock-to-clock
   IO delays
   IO standard, drive, slew and pull-up/pull-down
   Multi-cycle and false paths
   Max-delay paths
   DCM parameters

   – Physical Constraints [1]:
   SYN_LOC on IO pins and pad types

4. Also, include any synthesis attributes, from logic synthesis, such as syn_ramstyle, into the `.sdc` file.

5. Remove all successfully-translated constraints from the original `.qsf` file.

6. After translating and editing, the resulting `.qsf` might contain a few constraints that cannot be translated. If there are any QSF commands in the residual constraints file, add the `.qsf` file to your project. The file must have the same base name as the `.vqm` netlist so that the Altera place-and-route tool can source the file.

7. Save the project file.

Run Constraint Check

After you set up your project, you can check your constraints and their syntax. Do the following:

1. Make sure you target a technology that supports this feature.

2. Generate a constraint file, then select Run->Constraint Check.

[1] Physical constraints applied to invariant objects (such as registers, instantiated macros and modules) can be safely translated to SDC constraints. Please use the Design Planner™ tool for advanced physical constraints.

This command generates a report that checks the syntax and applicability of the timing constraints in the `.sdc` file(s) for your project. The report is written to the *`project_name`*`_cck.rpt` file.

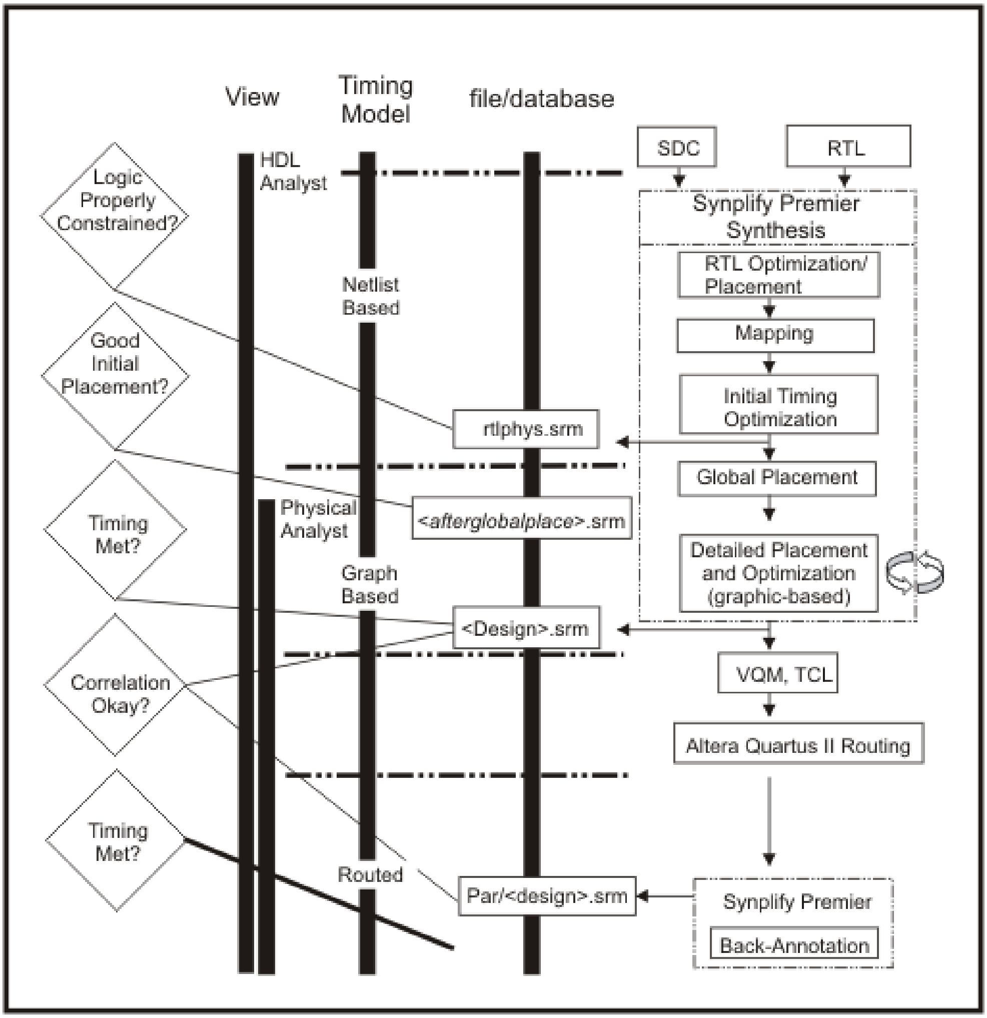
Create Design Plan File

Use Design Planner to interactively assign RTL modules, paths or components to regions on the device.

1. See *Creating Design Planner Regions for Altera Designs*, on page 10-33 of the *User Guide* for complete details on creating the design plan file.
2. After you have completed the design plan file (`.sfp`), add it to the project and enable the file in the Implementation Options ->Design Planning tab.
3. See *Run Physical Synthesis*, on page 1-27 when you are ready to run physical synthesis. To do this, you must have already completed the initial phases of physical synthesis:
   - *Set up Project*, on page 1-11
   - *Run Logic Synthesis*, on page 1-24
   - *Validate Logic Synthesis Results*, on page 1-25

Performance Results Comparison

Synplify Premier physical synthesis provides a timing closure solution that yields more accurate timing correlation and faster timing closure for your design. This section provides details on how to analyze results from logic synthesis, physical synthesis and place and route to show more accurate timing correlation between physical synthesis and final place and route results. The diagram below provides an overview of how to use the Synplify Premier tool and its features to analyze performance.



1. Determine performance improvement between logic synthesis and physical synthesis by comparing, for each clock:
   - The maximum frequency reported in the `.tan.rpt` file with the results from logic synthesis (log file).
   - The maximum frequency reported in the `.tan.rpt` file with the physical synthesis results (log file).
2. Determine the timing correlation by comparing, for the critical clocks:
   - The estimated performance in the `.srr` file.
   - The actual performance in the `.tan.rpt` file.
3. Determine the productivity gain by comparing:
   - The sum of logic synthesis runtime (reported in `.srr` file) with place-and-route runtime (reported in quartus`.log` file).
   - The physical synthesis runtime alone (reported in `.srr` file).

Running Multiple Implementations

You can create multiple implementations of the same design so that you can compare the results of each implementation and place-and-route run. This lets you experiment with different settings for the same design with different place-and-route options. Implementations are revisions of your design within the context of the Synplify Premier software and do not replace external source code control software and processes.

For the Graph-based physical synthesis with a design plan flow (Stratix II, Stratix II GX, and Stratix III only), you can run the first pass using the Synplify Premier software without a design plan file (`.sfp`) to synthesize the design. Placement and routing runs automatically. Then, create a new implementation and apply a design plan for Design plan-based physical synthesis.

See *Working with Multiple Implementations*, on page 6-10 of the *User Guide* for more information.