

PlanAhead ソフトウェア チュートリアル

RTL デザインと CORE Generator を使用した IP の生成

UG 675 (v 12.2) 2010 年 7 月 23 日





Xilinx is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with Xilinx FPGAs. Except as stated herein, none of the Design may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Any unauthorized use of the Design may violate copyright laws, trademark laws, the laws of privacy and publicity, and communications regulations and statutes.

Xilinx does not assume any liability arising out of the application or use of the Design; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Design. Xilinx reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED “AS IS” WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XILINX, OR ITS AGENTS OR EMPLOYEES. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XILINX IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XILINX HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XILINX WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications”) Xilinx specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

© 2010 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners.

Demo Design License

© 2010 Xilinx, Inc.

This Design is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this design file; if not, see: <http://www.gnu.org/licenses/>



PlanAhead™ ソースコードには、次のプログラムのソースコードが使用されています。

Centerpoint XML

- The initial developer of the original code is CenterPoint – Connective Software
- Software Engineering GmbH. portions created by CenterPoint – Connective Software
- Software Engineering GmbH. are Copyright© 1998-2000 CenterPoint - Connective Software Engineering GmbH. All Rights Reserved. Source code for CenterPoint is available at <http://www.cpointc.com/XML/>

NLView Schematic Engine

- Copyright© Concept Engineering.

Static Timing Engine by Parallax Software Inc.

- Copyright© Parallax Software Inc.

Java Two Standard Edition

- Includes portions of software from RSA Security, Inc. and some portions licensed from IBM are available at <http://oss.software.ibm.com/icu4j/>
- Powered By JIDE – <http://www.jidesoft.com>

The BSD License for the JGoodies Looks

Copyright© 2001-2010 JGoodies Karsten Lentzsch. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of JGoodies Karsten Lentzsch nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Free IP Core License

This is the Entire License for all of our Free IP Cores.

Copyright (C) 2000-2003, ASICs World Services, LTD. AUTHORS

All rights reserved.

Redistribution and use in source, netlist, binary and silicon forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of ASICs World Services, the Authors and/or the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

目次

RTL デザインと CORE Generator を使用した IP の生成	7
はじめに.....	7
サンプル デザイン データ	7
ザイリンクス ISE および PlanAhead ソフトウェア	8
ハードウェア要件.....	8
PlanAhead のマニュアルと情報	8
チュートリアルの説明	8
チュートリアルの目標	9
チュートリアルの手順	9
新規 RTL プロジェクトの作成 手順 1.....	10
[Sources] ビューと RTL Editor の使用 手順 2	15
RTL デザインのエラボレーションと解析 手順 3	21
リソース使用率と電力の概算 手順 4.....	28
RTL デザイン ルール チェックの実行 手順 5.....	32
ザイリンクス IP カタログからの IP の選択 手順 6.....	34
IP のカスタマイズおよびインスタンス化 手順 7	36
IP の生成 手順 8.....	41
まとめ	42

PlanAhead ソフトウェア チュートリアル

RTL デザインと CORE Generator を使用した IP の生成

はじめに

このチュートリアルでは、RTL 開発と解析環境の概要について説明します。

- RTL Editor を使用して RTL ソースをインポート
- エラボーレーションを実行して RTL をコンパイル
- さまざまな RTL 解析機能を使用してコンパイル済み RTL デザインを確認。これには、次の内容が含まれます。
- RTL 回路図を使用した RTL ロジック階層の解析
- 消費電力を使用して RTL リソースを概算
- RTL DRC を実行
- ザイリンクス IP カタログの確認、デザインに含まれる IP コアのカスタマイズとインプリメントの手順を確認

PlanAhead™ の解析機能の詳細は、ほかのチュートリアルで紹介しています。すべてのコマンド オプションについて説明されているわけではありせん。このチュートリアルでは、ISE® Design Suite の PlanAhead ソフトウェア製品に含まれる機能を使用しています。

サンプル デザイン データ

このチュートリアルでは、PlanAhead ソフトウェアをインストールすると含まれるサンプル デザイン データを使用します。サンプル デザイン データは、次のディレクトリにあります。

`<ISE_install_Dir>/PlanAhead/testcases/PlanAhead_Tutorial.zip`

書き込み権のあるディレクトリに ZIP ファイルを保存し、抽出します。チュートリアルでは、解凍ファイルのディレクトリを `<Install_Dir>` と記述しています。

チュートリアルのサンプル データは、チュートリアルを実行中に変更されます。各チュートリアルを実行する前に、まず元の PlanAhead_Tutorial データのコピーを取っておいてください。サンプル デザインの詳細は、「チュートリアルの説明」セクションを参照してください。

ザイリンクス ISE および PlanAhead ソフトウェア

PlanAhead ソフトウェアは、デフォルトで ISE Design Suite をインストールするとインストールされます。チュートリアルを始める前に、PlanAhead が起動できるか、サンプル デザイン データがインストールされているかを確認してください。ソフトウェアのインストール方法および詳細は、次のザイリンクス サイトから『ISE Design Suite 12 : インストール、ライセンス、リリース ノート』を参照してください。

http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/irn.pdf

ハードウェア要件

ターゲット デバイスが大規模の場合、2GB 以上の RAM 容量が必要です。このチュートリアルでは、小型のデザインを使用し、1 度に開くことができるデザインの数制限していますので、1GB で十分ですが、パフォーマンスに影響のすることもあります。

PlanAhead のマニュアルと情報

PlanAhead ソフトウェアの詳細については、次のマニュアルを参照してください。

- 『PlanAhead ユーザー ガイド』(UG632) – PlanAhead ソフトウェアに関する詳細情報
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/PlanAhead_UserGuide.pdf
- 『フロアプラン手法ガイド』(UG633) – フロアプランのヒント情報
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/Floorplanning_Methodology_Guide.pdf
- 『階層デザイン手法ガイド』(UG748) – PlanAhead の階層デザインの概要
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/Hierarchical_Design_Methodology_Guide.pdf
- ビデオ デモなど、PlanAhead のその他の情報については、<http://www.xilinx.com/planahead> を参照してください。

チュートリアルの説明

このチュートリアルで使用される小型のサンプル デザインには、Verilog と VHDL などの RTL デザイン ソースのセットが含まれます。VHDL ソースは、複数の VHDL ライブラリからのものです。このチュートリアルで使われるデザインには、次が含まれます。

- RISC プロセッサ
- 疑似 FFT
- ギガビットトランシーバ
- USB ポート モジュール 2 つ
- xc6vlx75tff784-3 デバイス

ハードウェア リソースやチュートリアルにかかる時間、データ サイズを節約するために、小型のデザインを使用しています。

チュートリアルに関する質問および問題は、ザイリックス テクニカル サポート (ホットライン) までご連絡ください。

チュートリアルの目標

このチュートリアルでは、PlanAhead ソフトウェアを使用した RTL 開発と解析プロセスについて説明します。

チュートリアルの手順

- 手順 1 新規 RTL プロジェクトの作成
- 手順 2 [Sources] ビューと RTL Editor の使用
- 手順 3 RTL デザインのエラボレーションと解析
- 手順 4 リソース使用率と電力の概算
- 手順 5 RTL デザイン ルール チェック (DRC) の実行
- 手順 6 ザイリックス IP カタログからの IP の選択
- 手順 7 IP のカスタマイズおよびインスタンス化
- 手順 8 IP の生成

新規 RTL プロジェクトの作成

手順 1

PlanAhead ソフトウェアでは、使用されるデザイン フローの段階によってさまざまなタイプのプロジェクトを作成できます。RTL ソースは、開発、解析、合成、インプリメンテーション、ビット ファイル生成などのプロジェクトを作成するために使用できます。

1-1. ソフトウェアを起動します。

- Windows の場合、Xilinx PlanAhead 12.2 のデスクトップ アイコンをダブルクリックするか、[スタート] → [プログラム] → [Xilinx ISE Design Suite 12.2] → [PlanAhead] → [PlanAhead] をクリックします。
- Linux の場合は、<Install_Dir>/PlanAhead_Tutorial/Tutorial_Created_Data ディレクトリに移動し、**planAhead** と入力します。

PlanAhead の Getting Started ページが開きます。

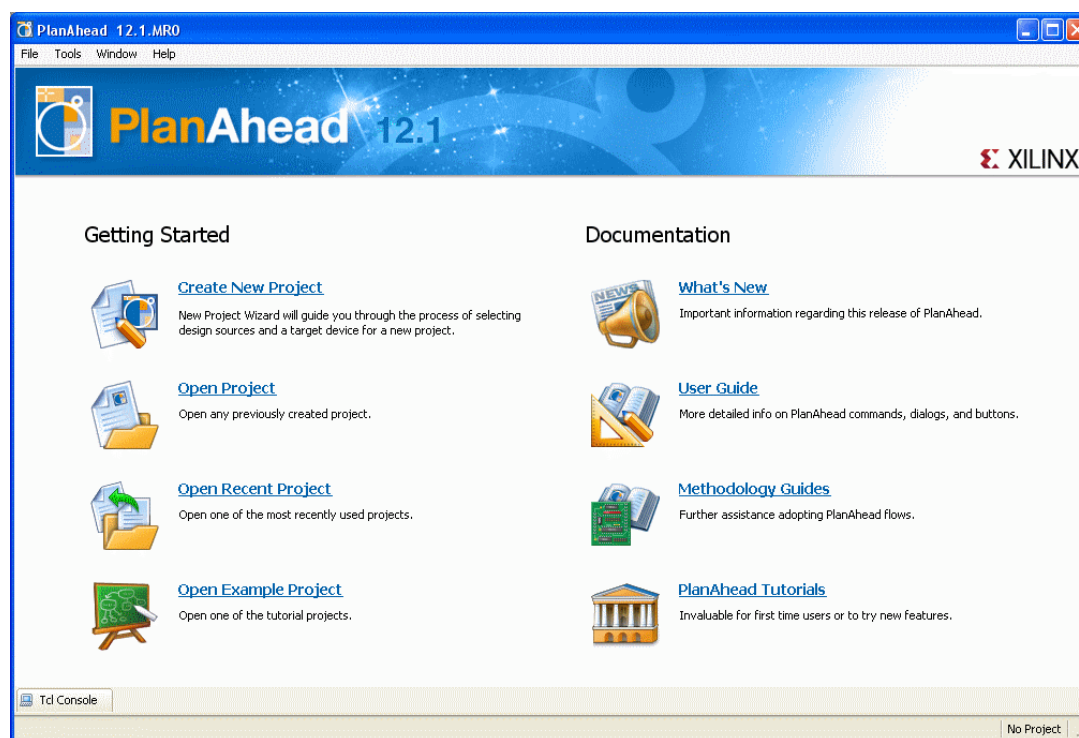


図 1 : PlanAhead の Getting Started ページ

PlanAhead の Getting Started ページには、プロジェクトを開いたり、作成したり、ドキュメントを確認するリンクが含まれます。

1-2. <Install_Dir>\PlanAhead_Tutorial\Sources\hdl ディレクトリの RTL ソース ファイルを使用して project_rtl という RTL プロジェクトを新規に作成します。

1-2-1. Getting Started ページの **Create New Project** というリンクをクリックします。

1-2-2. [Create a New PlanAhead Project] ページで **[Next]** をクリックします。

[New Project] ダイアログ ボックスの [Project Name] ページが開きます。

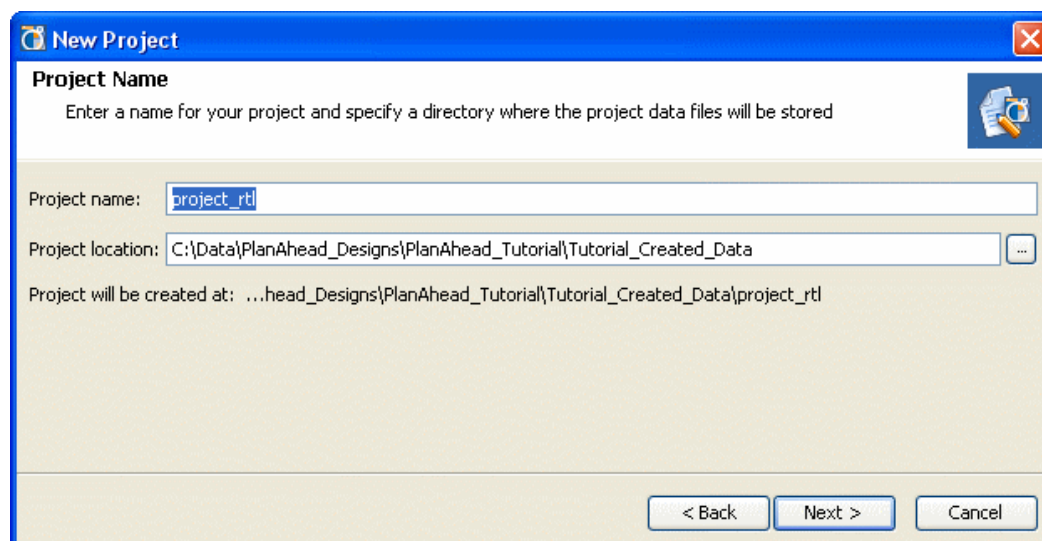


図 2 : [Project Name] ページ

1-2-3. 参照ボタンで次のフォルダを指定します。<Install_Dir>\PlanAhead_Tutorial\Tutorial_Created_Data.

1-2-4. プロジェクト名はデフォルトの **project_rtl** のままにし、**[Next]** をクリックします。

[Design Source] ページが表示されます。

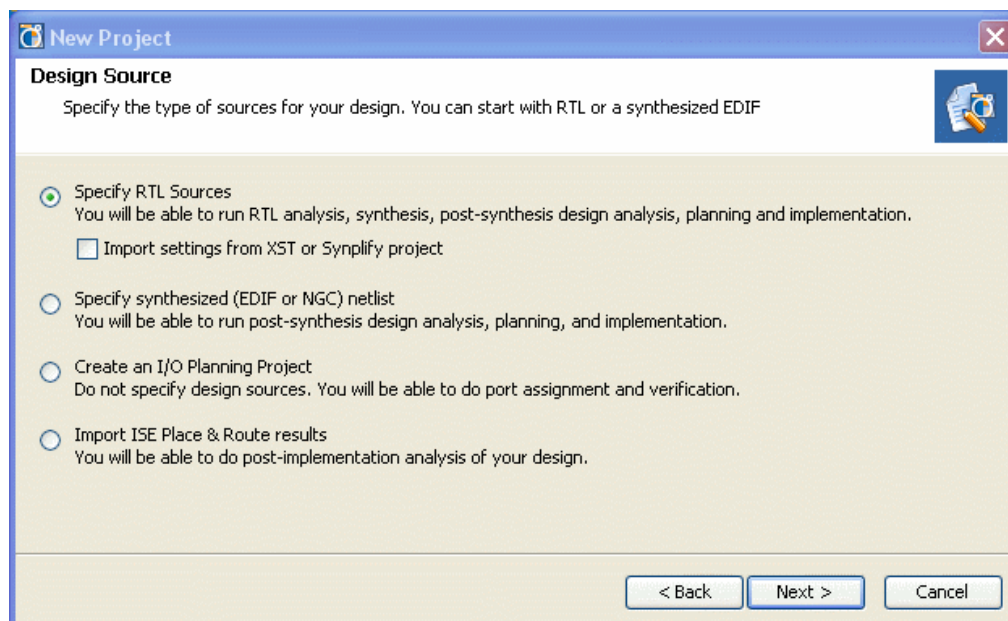


図 3 : インポートする RTL ソースの選択

1-2-5. [Specify RTL Sources] をオンにし、[Next] をクリックします。

[Add Sources] ページが表示されます。

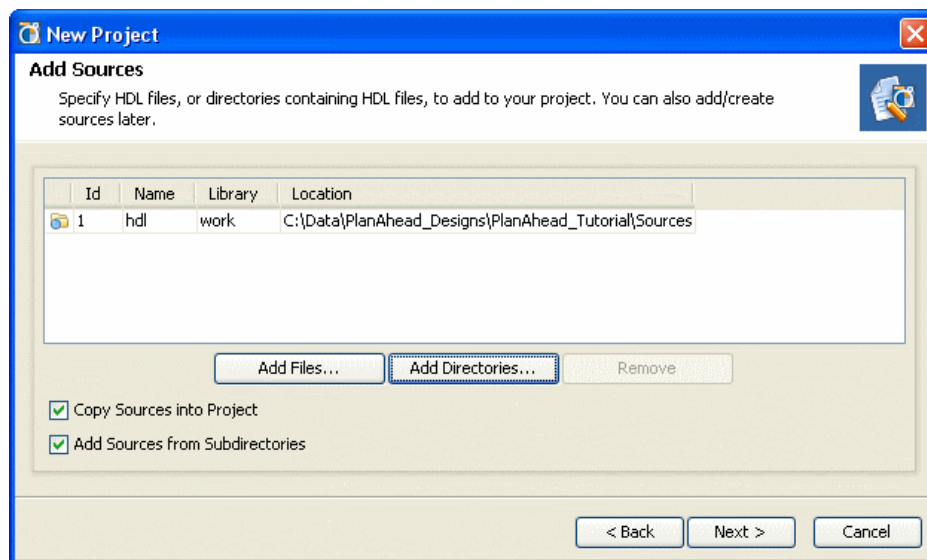


図 4 : プロジェクトに追加するソースの選択

1-3. ディレクトリおよびファイルを追加します。

1-3-1. [Add Directories] ボタンをクリックし、次のディレクトリを選択します。

<Install_Dir>/PlanAhead_Tutorial/Sources/hdl

1-3-2. [Copy Sources into Project] と [Add Sources from Subdirectories] がオンになっていることを確認します。

1-3-3. 次の図 4 のように設定したら、[Next] をクリックします。

[Constraints Files] ページが表示されます。

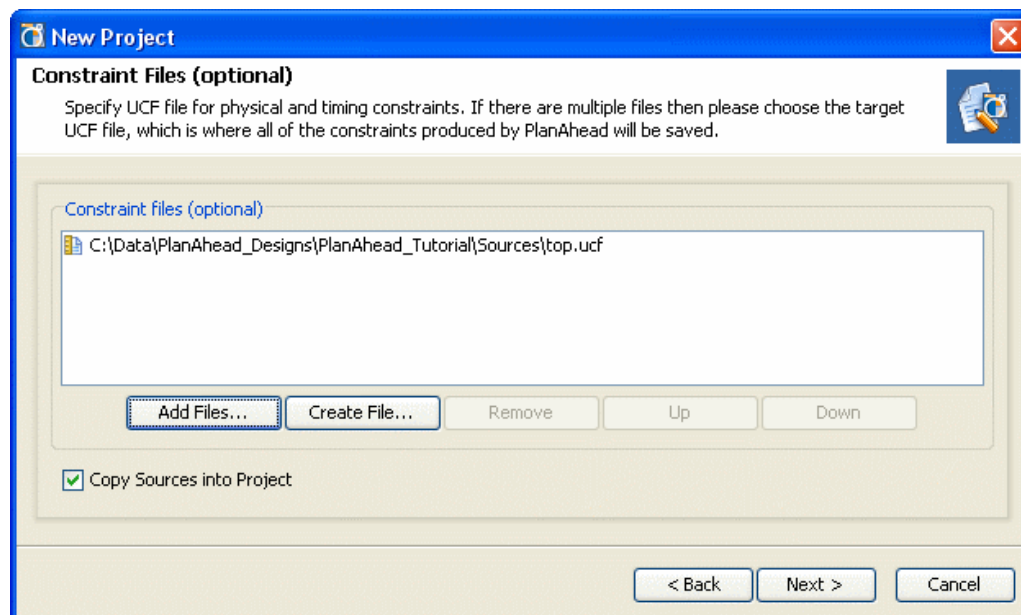


図 5 : [New Project] ダイアログ ボックスの [Constraint Files] ページ

1-4. 制約ファイルを追加します。

1-4-1. [Add Files] ボタンをクリックし、次のディレクトリを選択します。

<Install_Dir>\PlanAhead_Tutorial\Sources\top.ucf

1-4-2. [Copy Sources into Project] をオンにします。

1-4-3. [Next] をクリックします。

[Default Part] ページが表示されます

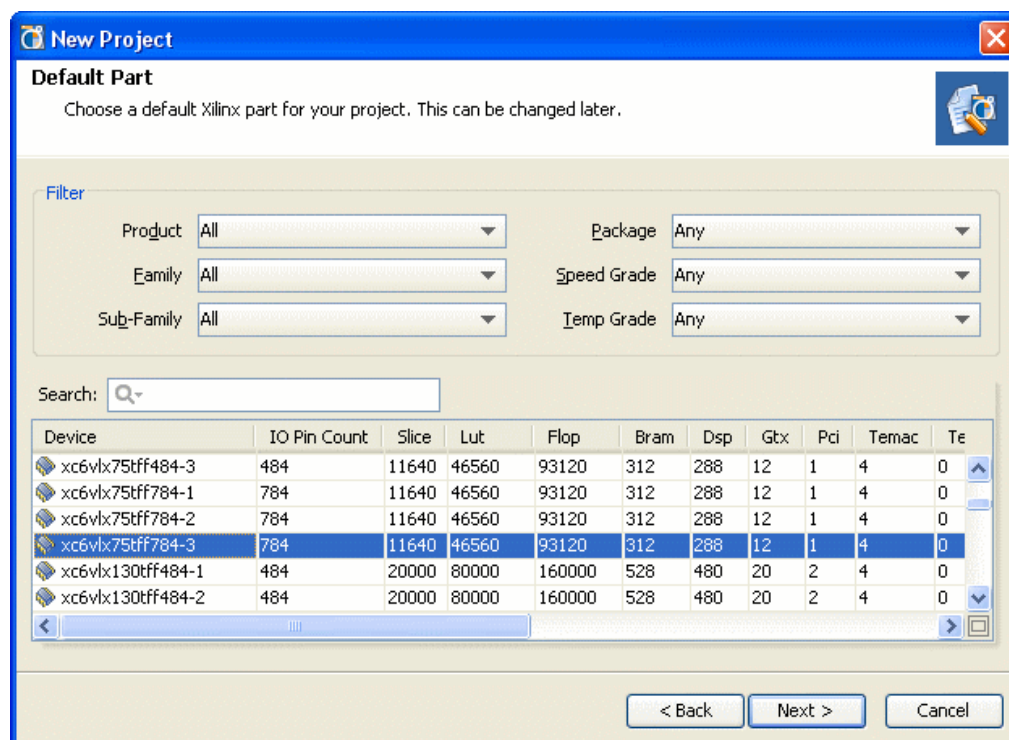


図 6 : ファミリーおよびデフォルト パーツの選択

1-5. デフォルト パーツを選択します。

1-5-1. xc6vlx75tff784-3 デバイスを選択し、[Next] をクリックします。

1-5-2. サマリを確認したら、[Finish] をクリックします。

PlanAhead 環境が開きます。

[Sources] ビューと RTL Editor の使用

手順 2

PlanAhead ソフトウェアには、Verilog、VHDL、NGC 形式のコアなどさまざまなファイル形式のデザイン ソースを追加できます。これらのファイルは、[Sources] ビューに分類されて表示されます。RTL ソースの作成または開発には、RTL Editor を使用します。

2-1. [Sources] ビューとプロジェクト サマリを確認します。

2-1-1. [Project Summary] の情報を確認します。デザインの進捗状況に応じて、さらに情報が追加されていきます。

2-1-2. [Sources] ビューを確認します。

2-1-3. [VHDL] フォルダの横のマイナスサインをクリックして、展開表示します (図 7)。

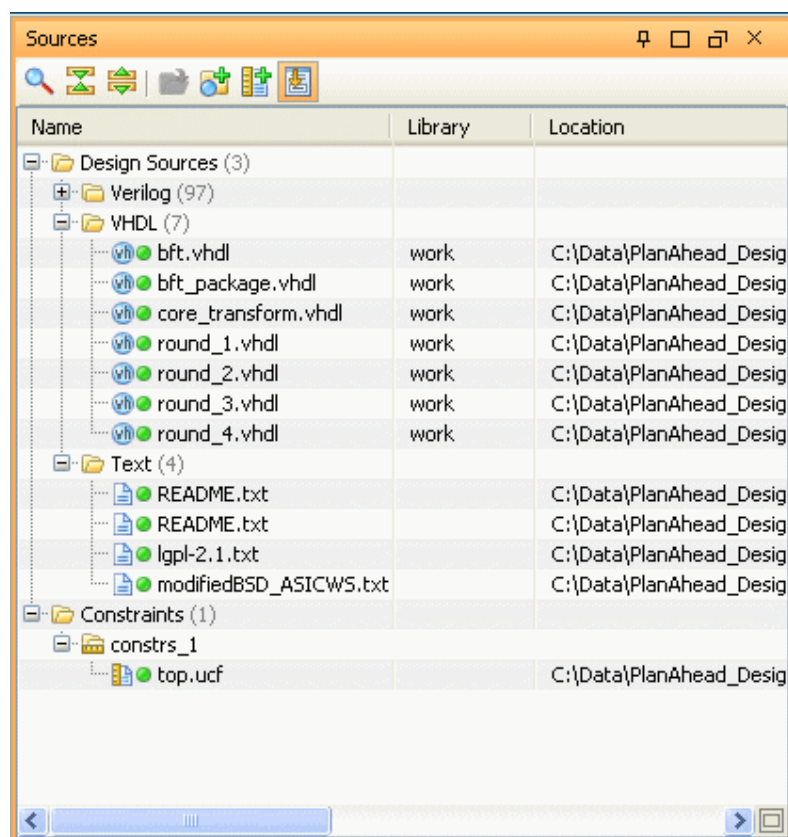


図 7: ソースの種類別表示

ソースはファイルの種類別に表示されます。[Library] および [Location] 列からは、ソース ファイルのライブラリ名とディレクトリ情報が確認できます。

2-2. 選択した VHDL ソースの VHDL ライブラリを bftLib に設定します。

2-2-1. Shift キーを使用して bft.vhdl ファイル以外の VHDL ソース ファイルをすべて選択します (図 8)。

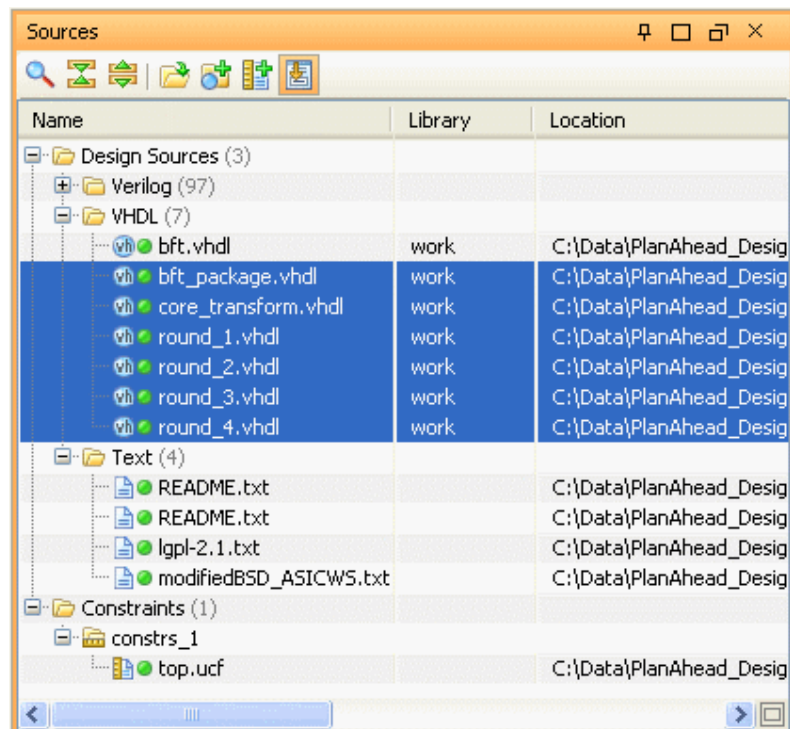


図 8 : VHDL ソースを選択して VHDL ライブラリを設定

2-2-2. [Sources] ビューでアイテムを選択したままで右クリックし、[Set Library] をクリックします。

2-2-3. [Specify Library] ダイアログ ボックスで **bftLib** と入力し、[OK] をクリックします。

これで、選択したファイルに対して **bftLib** という VHDL ライブラリが設定されました (図 9)。

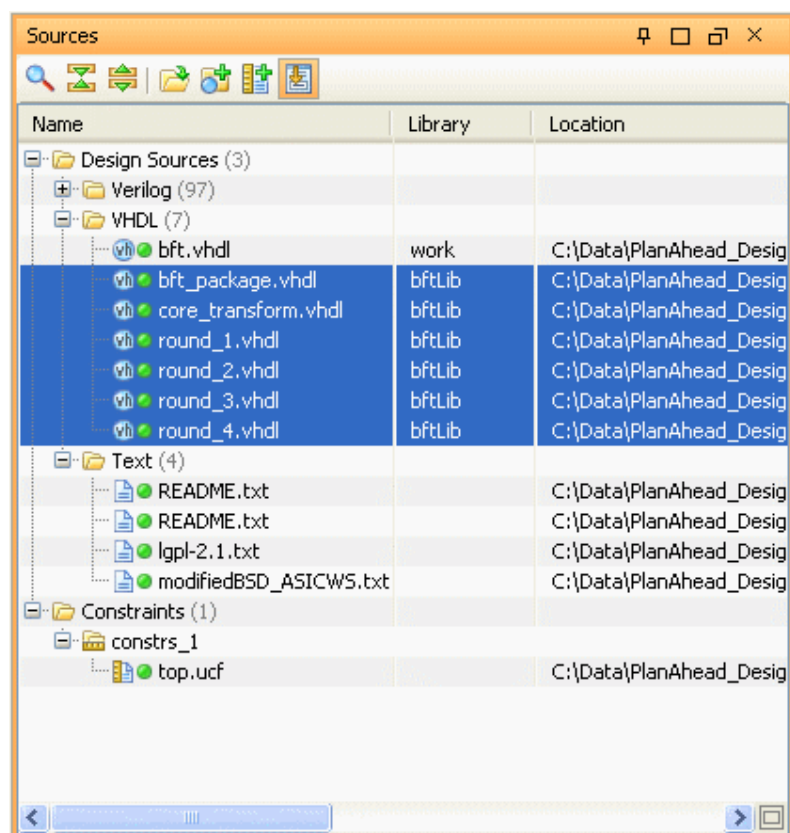


図 9 : VHDL ライブラリの設定

2-3. [Sources] ビューのコマンドを確認します。

2-3-1. [Sources] ビューで VHDL ソースの 1 つを選択します。

2-3-2. 右クリックし、[Sources] ビューのポップアップ メニューでどういうコマンドが表示されるか確認してみてください。表示をオフにするには Esc キーを押します。

2-4. RTL Editor を使用します。

2-4-1. [Sources] ビューで VHDL ソース ファイルの 1 つをダブルクリックし、RTL Editor で開きます。

2-4-2. 次のウィンドウは、[Find in Files] ポップアップ メニュー コマンドを実行すると開きます。

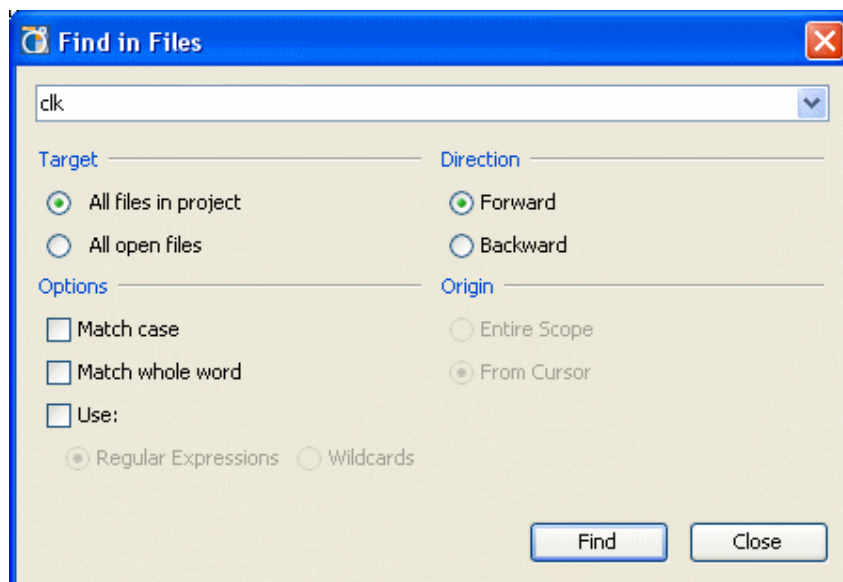


図 10 : ファイル内の検索

2-4-3. **clk** と入力し、[Find] をクリックします。

[Find in Files] ビューが PlanAhead 環境の下部にあるメッセージ エリアに表示されます。

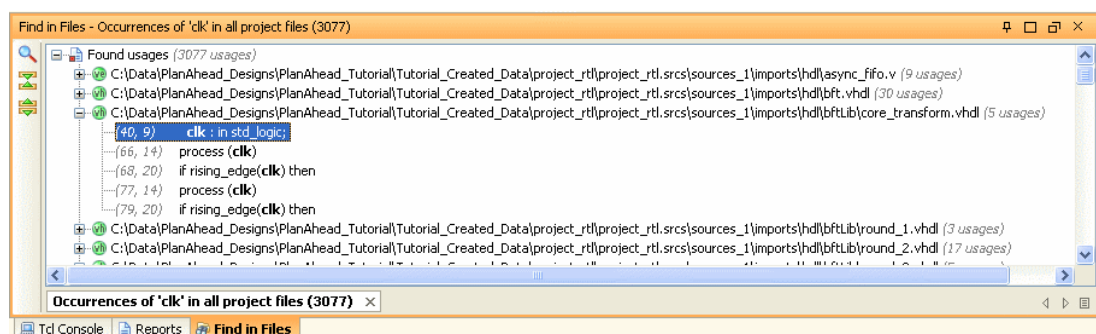


図 11 : 検出結果の表示

2-4-4. [Find in Files] ビューでディレクトリを展開し、clk を含む文の 1 つを選択すると、そのファイルが RTL Editor で表示されます。

2-4-5. [Find in Files] ビューの X ボタンをクリックして、ビューを閉じます。

2-4-6. RTL Editor で開いた RTL タブも X ボタンをクリックしてそれぞれ閉じます。

2-5. RTL ソース ファイルを新規作成し、テンプレートをインポートします。

PlanAhead では、Verilog または VHDL ソース ファイルを新規に作成できます。ザイリンクスの提供する標準テンプレートを使用すると、それを元にさまざまなロジックおよびコード構造などを記述できます。

2-5-1. Project Manager の下の Flow Navigator で **[Add/Create Sources]** をクリックします。

2-5-2. [Add/Create Sources] ダイアログ ボックスで **[Create File]** ボタンをクリックします。

[Create Source File] ダイアログ ボックスが開きます (図 12)。

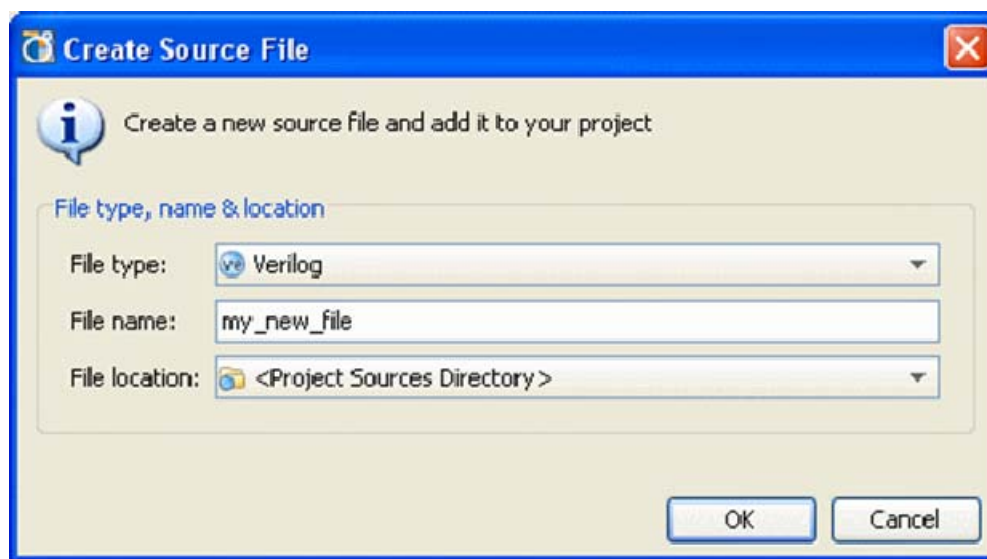


図 12 : [Create Source File] ダイアログ ボックス

2-5-3. [File location] を `<Install_dir>/PlanAhead_Tutorial/Tutorial_Created_Data` に設定します。

2-5-4. [File name] ボックスに **my_new_file** と入力します。

2-5-5. **[OK]** をクリックします。

2-5-6. [Add/Create Sources] ダイアログ ボックスで **[OK]** をクリックします。

新しい白紙のファイルが [Sources] ビューの [Verilog] フォルダにリストされるようになります。

2-5-7. [Sources] ビューで **my_new_file.v** ファイルをダブルクリックし、RTL Editor で開きます。

2-5-8. RTL Editor で **[Insert Template]** ポップアップ メニュー コマンドを選択して、[Insert Template] ダイアログ ボックスを開きます (図 13)。

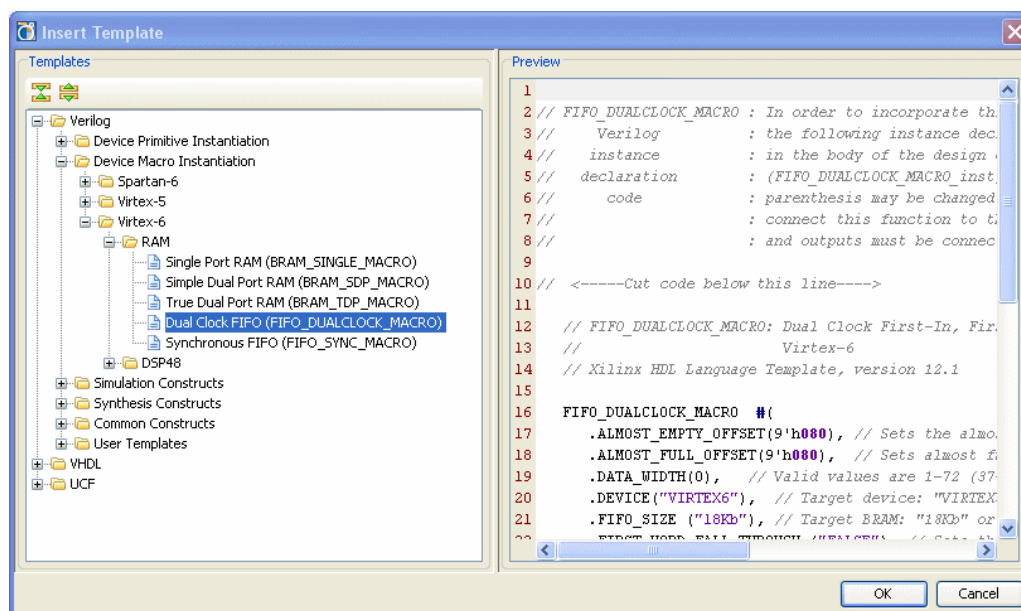


図 13 : [Insert Template] ダイアログ ボックス

2-5-9. Verilog フォルダで使用可能なテンプレートタイプを確認し、1 つ選んで、[OK] をクリックします。

テンプレートテキストが新規ソース ファイルに挿入されます。

2-5-10. RTL Editor の新規ソース ファイルを X ボタンをクリックして閉じます。

2-5-11. [Save Text Editor Changes] ダイアログ ボックスで [No] をクリックします。

RTL デザインのエラボレーションと解析

手順 3

PlanAhead にはプロジェクトの RTL ソース ファイルをコンパイルする RTL エラボレーション機能が含まれています。コンパイル エラーや警告メッセージが表示され、クリックすると RTL コードの該当部分を選択されます。RTL ロジック階層を展開すると、解析機能が使用できます。エラボレーションが終了すると、すべての RTL ビューでロジック オブジェクトの選択が連動するようになります。Flow Navigator から RTL デザインを開くと、RTL デザインが自動的にエラボレーションされ、Design Planner と I/O Planner のビュー レイアウトが表示されます。

- [RTL Netlist] および [Hierarchy] ビューには、デザインのロジック階層が表示されます。
- [RTL Schematic] ビューもインタラクティブに動作します。
- [Find] コマンドでは、RTL ロジック オブジェクトを検索できます。
- [Instance Properties] ビューには、リソース概算を含む選択したロジックのインスタンス化に関する情報が表示されます。
- RTL DRC を実行すると、問題になりそうなエリアがハイライトされ、電力またはパフォーマンスが改善しやすくなります。

3-1. 最上位レベル モジュールに top を使用して、RTL デザインをエラボレーションし、開きます。

3-1-1. Flow Navigator で [RTL Design] ボタン (図 14) をクリックします。

3-1-2. [Top Module] ダイアログ ボックスの [Top Module Name] に **top** と入力し、[OK] をクリックしてエラボレーションを開始します。

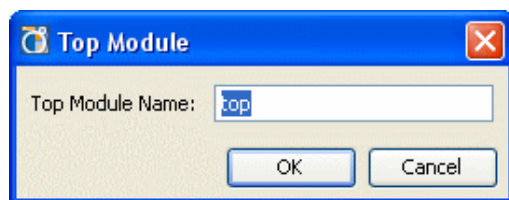


図 14 : [Top Module] ダイアログ ボックス

[Elaboration Messages] ビューが開きます (図 15)。

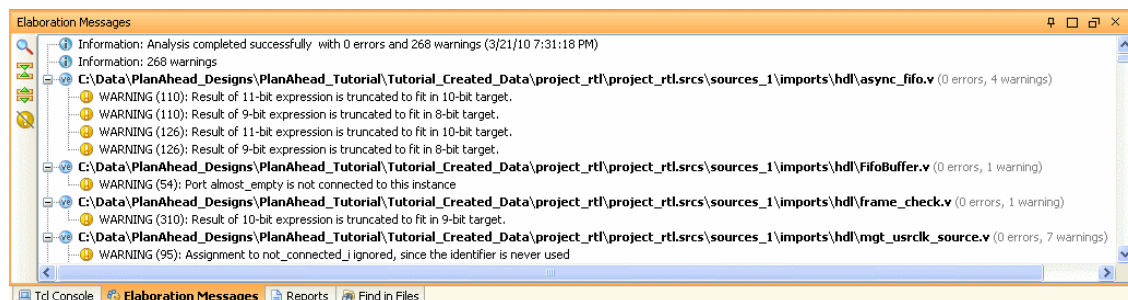


図 15 : [Elaboration Messages] の表示

3-2. ささまざまなエラボーレーション警告を確認します。

- 3-2-1. [Elaboration Messages] ビューで [Hide Warning Messages] ボタン () をクリックします。

このデザインの場合、エラーはありませんが、エラー メッセージがフィルタされて表示されます。

- 3-2-2. [Elaboration Messages] ビューで [Hide Warning Messages] ボタンをもう一度クリックして警告メッセージを表示します。

- 3-2-3. [Elaboration Messages] ビューの警告メッセージの 1 つをクリックします。RTL ファイルの該当する行が RTL Editor で表示されます。必要な場合は、そのソースファイルが開きます。

- 3-2-4. [Elaboration Messages] のビューの右上の X ボタンをクリックして、このビューを閉じます。

- 3-2-5. 開いている RTL ファイルの X ボタンをそれぞれクリックして、RTL Editor を閉じます。

3-3. RTL ロジック階層を確認します。

- 3-3-1. [RTL Netlist] ビューで **usbEngine0** インスタンスの横の + マークをクリックして展開表示します。

- 3-3-2. **usbEngine0/u0** インスタンスを選択します。

- 3-3-3. 右クリックし、[Show Definition] をクリックします。

usbg_utmi_if モジュール インスタンスエーションを含む RTL ファイルが RTL Editor で開きます (図 16)。

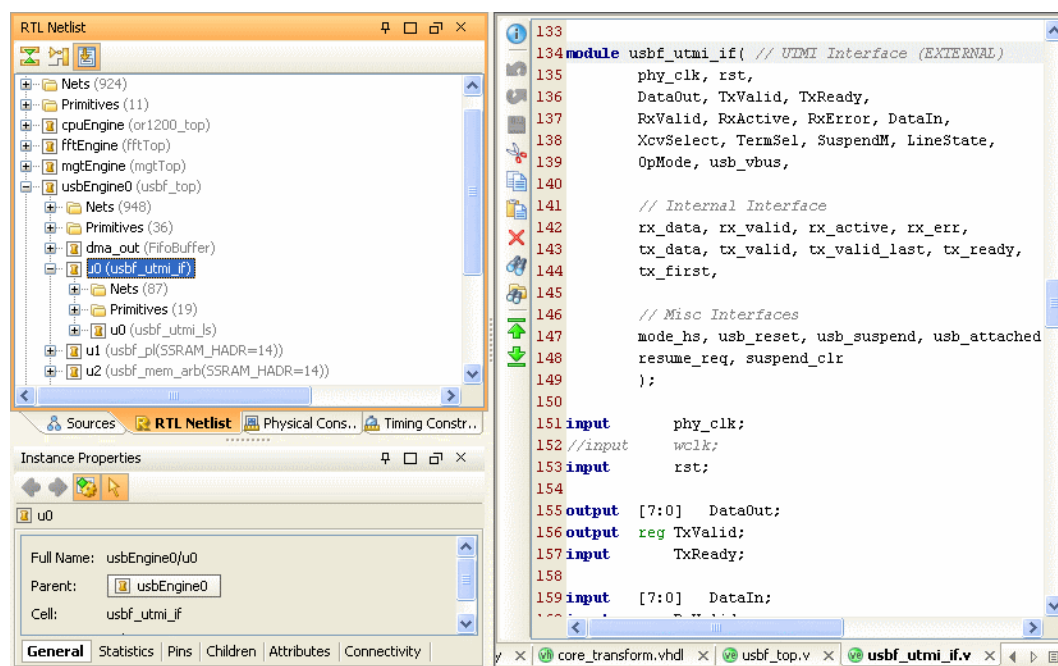


図 16 : RTL ロジック階層の表示

- 3-3-4. [RTL Netlist] ビューで右クリックし、[Show Source] をクリックし、usb_utmi_if コードを含む RTL の行が RTL Editor で開いていることを確認します。
- 3-3-5. [RTL] ビューで右クリックし、[Show Hierarchy] をクリックします。

[RTL Hierarchy] ビューが開き、選択したモジュールがハイライトされます。モジュールを示す長方形のサイズは、含まれるロジック数に対応しており、大きいモジュールがどれかわかりやすくなっています (図 17)。

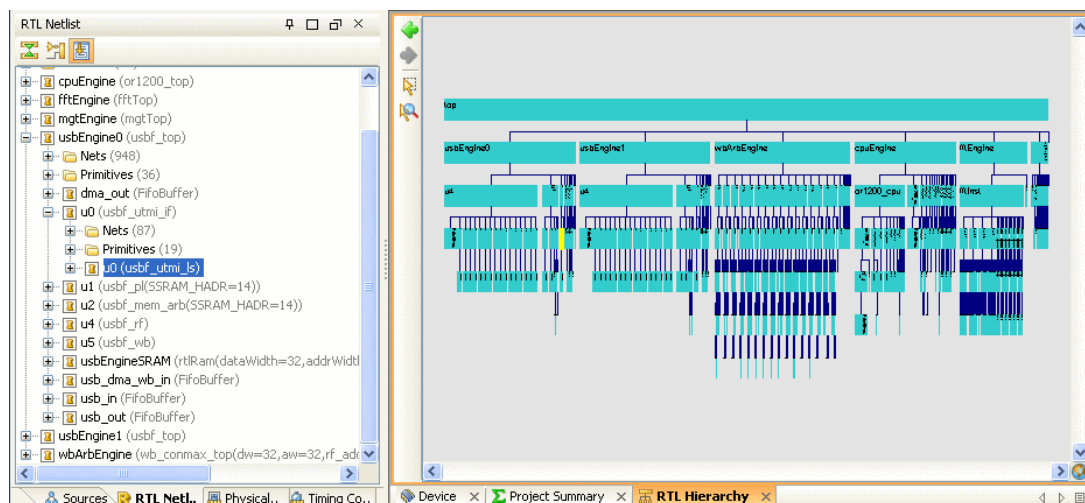


図 17 : [RTL Hierarchy] ビューのモジュールの表示

- 3-3-6. [RTL Hierarchy] ビューの X ボタンをクリックして、ビューを閉じます。
- 3-3-7. RTL Editor で X ボタンをクリックして、開いている RTF ファイルをすべて閉じます。
- 3-4. RTL 回路図を確認します。
 - 3-4-1. [RTL Netlist] ビューで **usbEngine0/u0/u0** インスタンス (前の選択の下レベルにあり) をクリックします。
 - 3-4-2. 右クリックし、[**Schematic**] をクリックします。

3-5. [RTL Schematic] ビューに表示されるモジュールを確認します。

3-5-1. u0 モジュール内で **LineState** ピンをダブルクリックし、ロジックを展開表示します (図 18)。

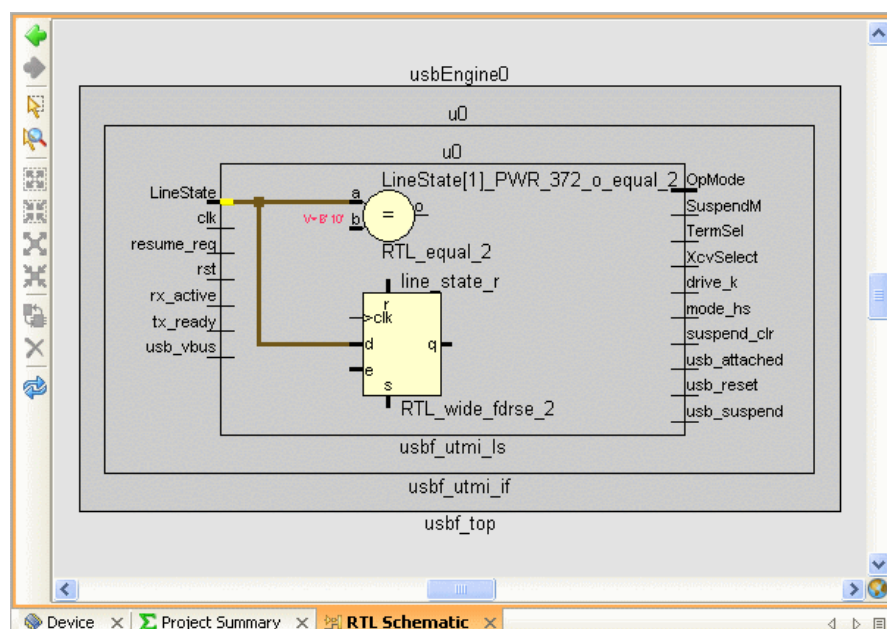


図 18 : [RTL Schematic] ビュー

3-5-2. u0 モジュールの外側で **LineState** ピンをダブルクリックし、ロジックを展開表示します (図 19)。

3-5-3. [RTL Schematic] ビューで [Zoom Fit] をクリックします (ビュー内でクリックして、右下から左上にカーソルをドラッグした場合も同じ表示になります)。

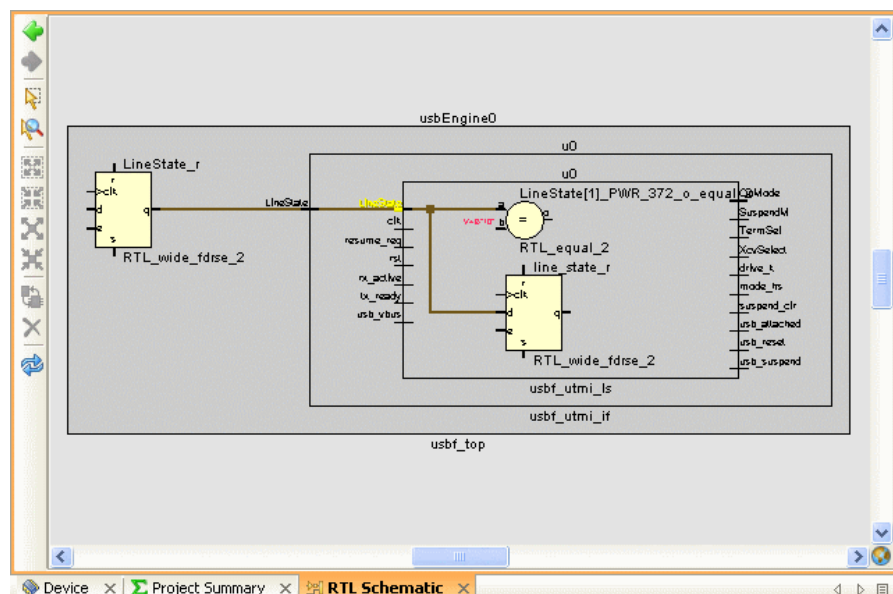


図 19 : [RTL Schematic] ビューでロジックの展開

回路図の展開表示の詳細については、PlanAhead チュートリアル「デザイン解析およびフロアプラン」(UG676)を参照してください。

- 3-5-4. [RTL Schematic] ビューの左側で RTL_wide_fdrse_2 インスタンスを選択します。
- 3-5-5. [RTL Schematic] ビューで右クリックし、[Show Source] をクリックし、RTL ファイルにロジック定義が含まれているのを確認します。
- 3-5-6. [RTL Editor] および [RTL Schematic] ビューを閉じます。
- 3-5-7. [RTL Netlist] ビューで [Collapse All] ボタン () をクリックします。
- 3-6. [Find] コマンドで RTL ブロック RAM ロジックを検索します。
- 3-6-1. [Edit] → [Find] をクリックするか、[Find] ボタン () をクリックして [Find] ダイアログ ボックスを開きます (図 20)。

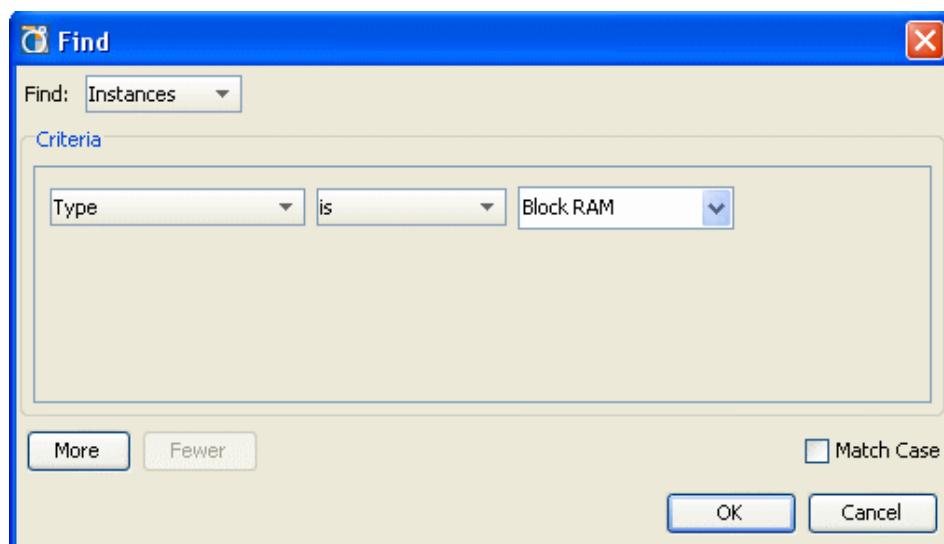


図 20 :[Find] ダイアログ ボックスを使用した RTL ロジックの検索

3-6-2. [Find] フィルタ オプションを確認します。

3-6-3. [Criteria] で [Type]、[is]、[Block RAM] を選択し、[OK] をクリックして [Find Results] ビューを開きます (図 21)。

Find Results - Instances - Type is 'Block RAM' (36)

	Id	Name	Cell	Pins
	1	cpuEngine/or1200_immu_top/or1200_immu_tlb/itlb_mr_ram/ramb16_s18	RAMB16_S18	50
	2	cpuEngine/or1200_immu_top/or1200_immu_tlb/itlb_tr_ram/ramb16_s36	RAMB16_S36	85
	3	cpuEngine/or1200_ic_top/or1200_ic_ram/ic_ram0/ramb16_s9_0	RAMB16_S9	33
	4	cpuEngine/or1200_ic_top/or1200_ic_ram/ic_ram0/ramb16_s9_1	RAMB16_S9	33
	5	cpuEngine/or1200_ic_top/or1200_ic_ram/ic_ram0/ramb16_s9_2	RAMB16_S9	33
	6	cpuEngine/or1200_ic_top/or1200_ic_ram/ic_ram0/ramb16_s9_3	RAMB16_S9	33
	7	cpuEngine/or1200_dmmu_top/or1200_dmmu_tlb/dtlb_mr_ram/ramb16_s18	RAMB16_S18	50

Instances - Type is 'Block RAM' (36) x

図 21 :RTL ブロック RAM の検索結果

3-6-4. 検索結果が [Find Results] ビューに表示されたら、[Find Results] ビューを閉じます。

リソース使用率と電力の概算

手順 4

4-1. リソース概算のオプションを確認します。

4-1-1. Flow Navigator で **[Resource Estimation]** コマンドをクリックします。

4-1-2. [Resource Estimation] ビューが表示されます (図 22)。

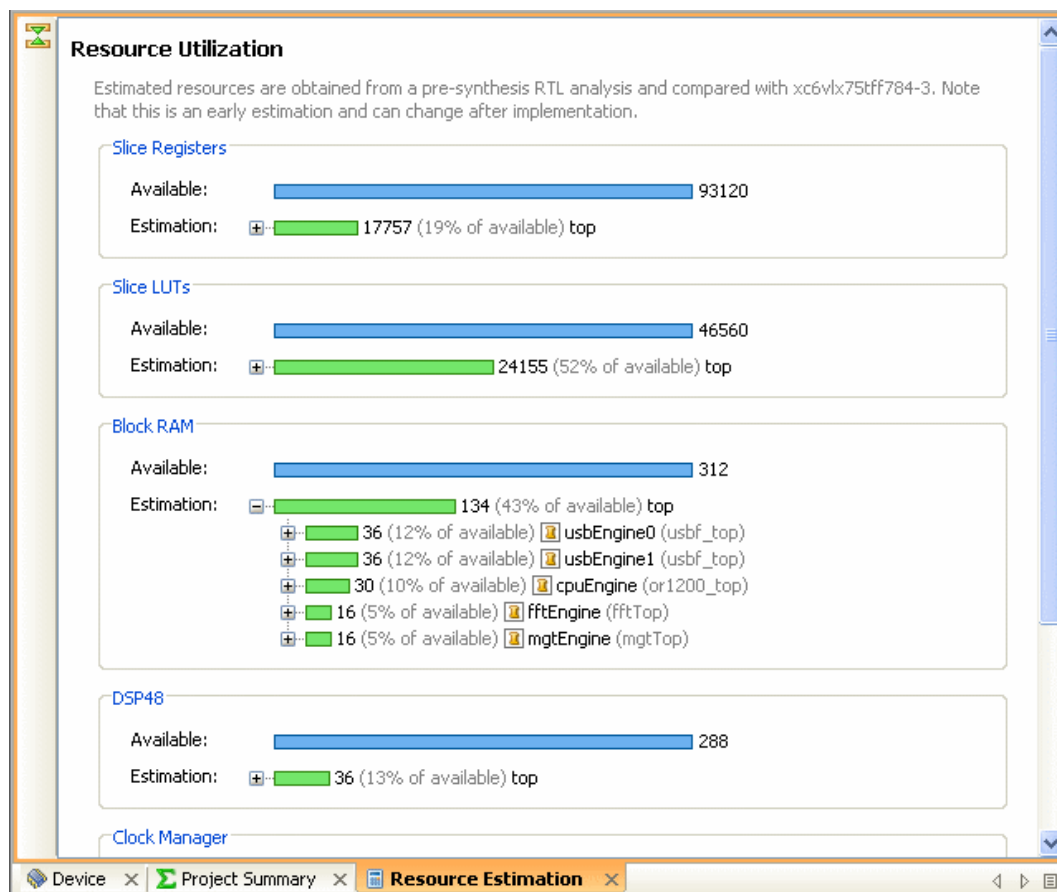


図 22 : RTL リソース概算の表示

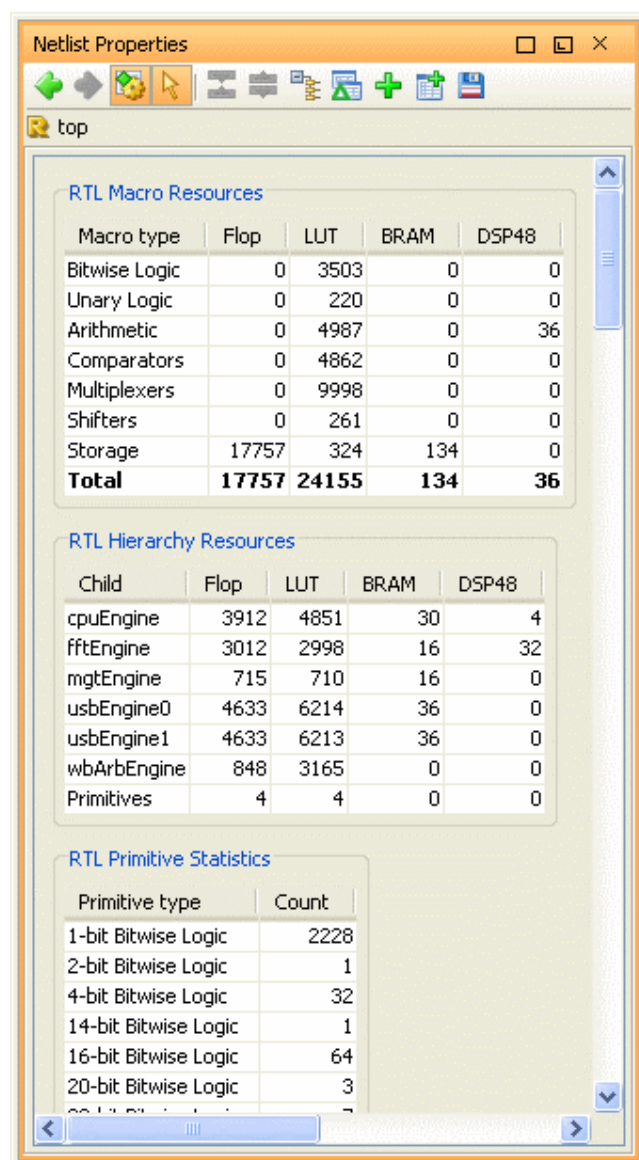
4-1-3. [Block RAM] フィールドの [Estimation] ツリーを展開し、その下のレポートを確認します。

4-1-4. [Resource Estimation] の X マークをクリックし、[Resource Estimation] ビューを閉じます。

4-2. RTL インスタンスのリソース概算を確認します。

4-2-1. [RTL Netlist] ビューで **top** をクリックすると、[Netlist Properties] ビューに RTL マクロリソースが表示されます (図 23)。

4-2-2. [Netlist Properties] ビューが表示されあいは、右クリックで **[Netlist Properties]** を選択します。



The screenshot shows the 'Netlist Properties' window with the 'top' instance selected. It displays three sections of resource usage data:

RTL Macro Resources				
Macro type	Flop	LUT	BRAM	DSP48
Bitwise Logic	0	3503	0	0
Unary Logic	0	220	0	0
Arithmetic	0	4987	0	36
Comparators	0	4862	0	0
Multiplexers	0	9998	0	0
Shifters	0	261	0	0
Storage	17757	324	134	0
Total	17757	24155	134	36

RTL Hierarchy Resources				
Child	Flop	LUT	BRAM	DSP48
cpuEngine	3912	4851	30	4
fftEngine	3012	2998	16	32
mgtEngine	715	710	16	0
usbEngine0	4633	6214	36	0
usbEngine1	4633	6213	36	0
wbArbEngine	848	3165	0	0
Primitives	4	4	0	0

RTL Primitive Statistics	
Primitive type	Count
1-bit Bitwise Logic	2228
2-bit Bitwise Logic	1
4-bit Bitwise Logic	32
14-bit Bitwise Logic	1
16-bit Bitwise Logic	64
20-bit Bitwise Logic	3

図 23 : RTL リソース概算の表示

4-2-3. [Netlist Properties] をスクロール ダウンし、RTL 階層リソース、RTL メモリリソース、RTL プリミティブ統計、ネットバウンダリ統計、クロックレポートなどの情報を確認します。

4-2-4. [RTL Netlist] ビューでその他のモジュールのどれかを選択して、概算を確認してみます。

4-3. RTL の消費電力の概算を出します。

4-3-1. Flow Navigator で **[Power Estimation]** をクリックします。

4-3-2. [Power Estimation] ダイアログ ボックスが開きます (図 24)。

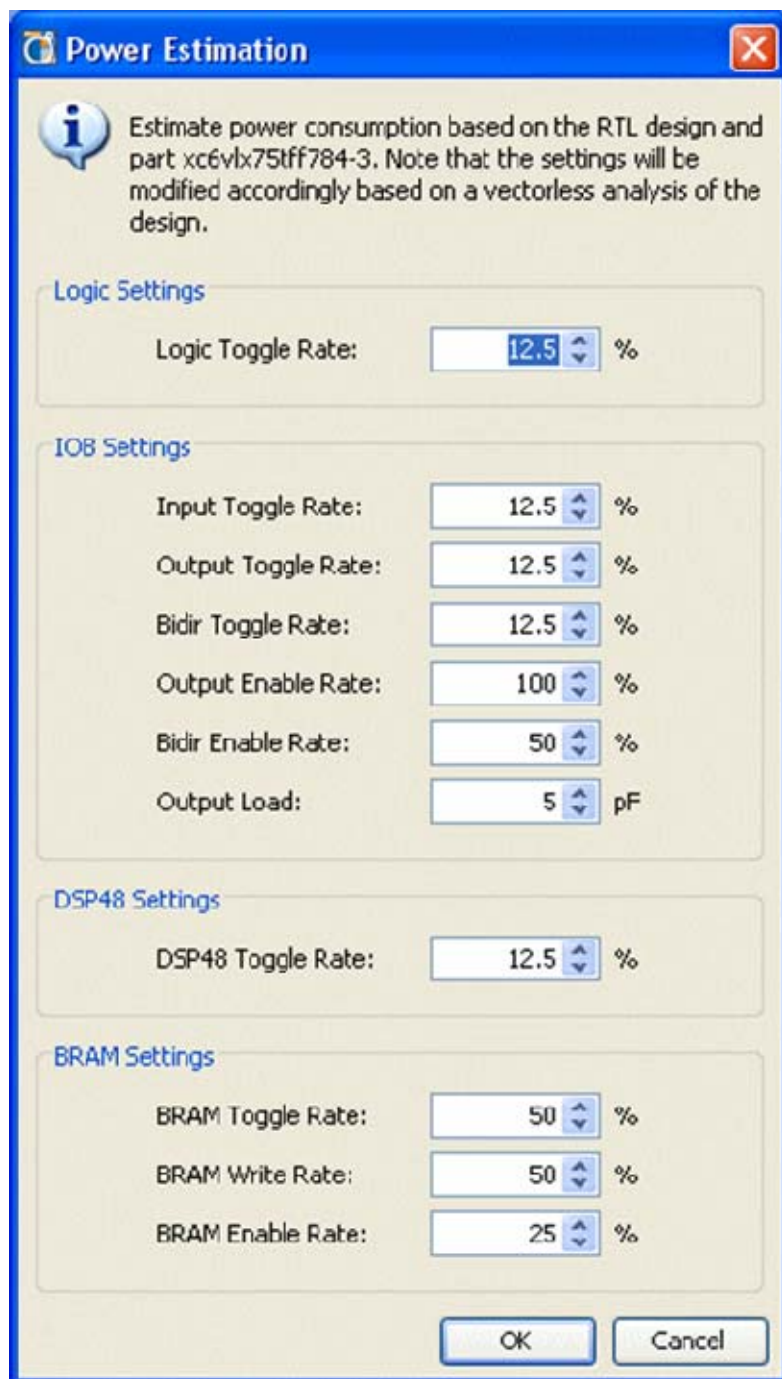


図 24 : 電力概算の設定

4-3-3. デフォルトの設定のまま、**[OK]** をクリックします。

4-3-4. [Power Estimation] ビューが開きます (図 25)。

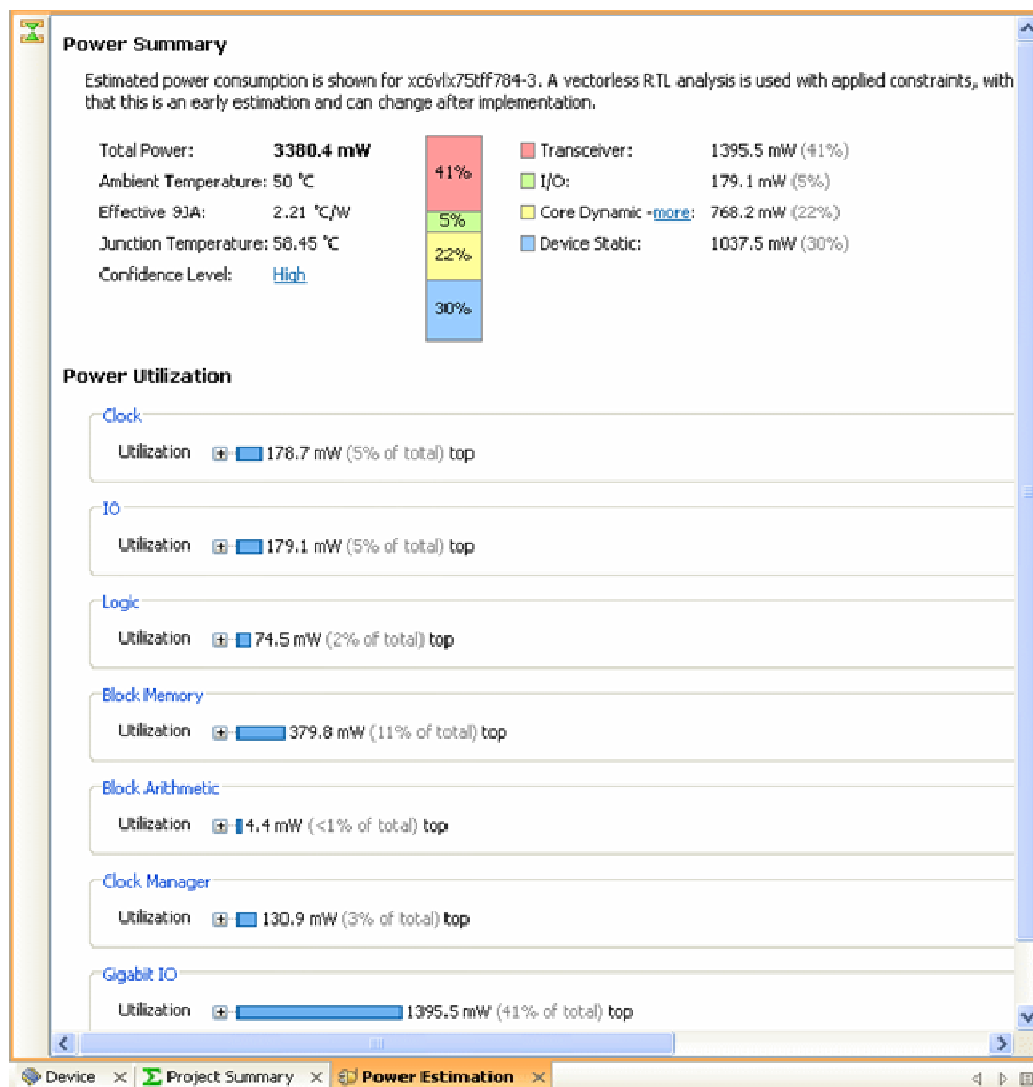


図 25 : RTL デザインの消費電力の概算

4-3-5. スクロールダウンし、[Utilization] ツリーを展開すると、さまざまなリソースの種類が表示されます。

4-3-6. [Power Estimation] ビューを閉じます。

RTL デザイン ルール チェックの実行

手順 5

PlanAhead には RTL デザインで実行可能なデザイン ルール チェック (DRC) が複数含まれます。中には、LINT スタイルの RTL チェックで、電力やパフォーマンスの改善に対する提案が表示されるものもあります。また、RTL デザインの基本的な I/O バンクや電圧規則をチェックするものもあります。デザインが合成されたら、さらに多くのロジック デザイン、I/O およびクロックの DRC が使用できるようになります。

5-1. DRC を実行します。

5-1-1. Flow Navigator または [Tools] メニューから [Run DRC] をクリックします。

5-1-2. [Run DRC] ダイアログボックスが表示されます。RTL 規則を展開表示して確認し (図 26)、[OK] をクリックします。

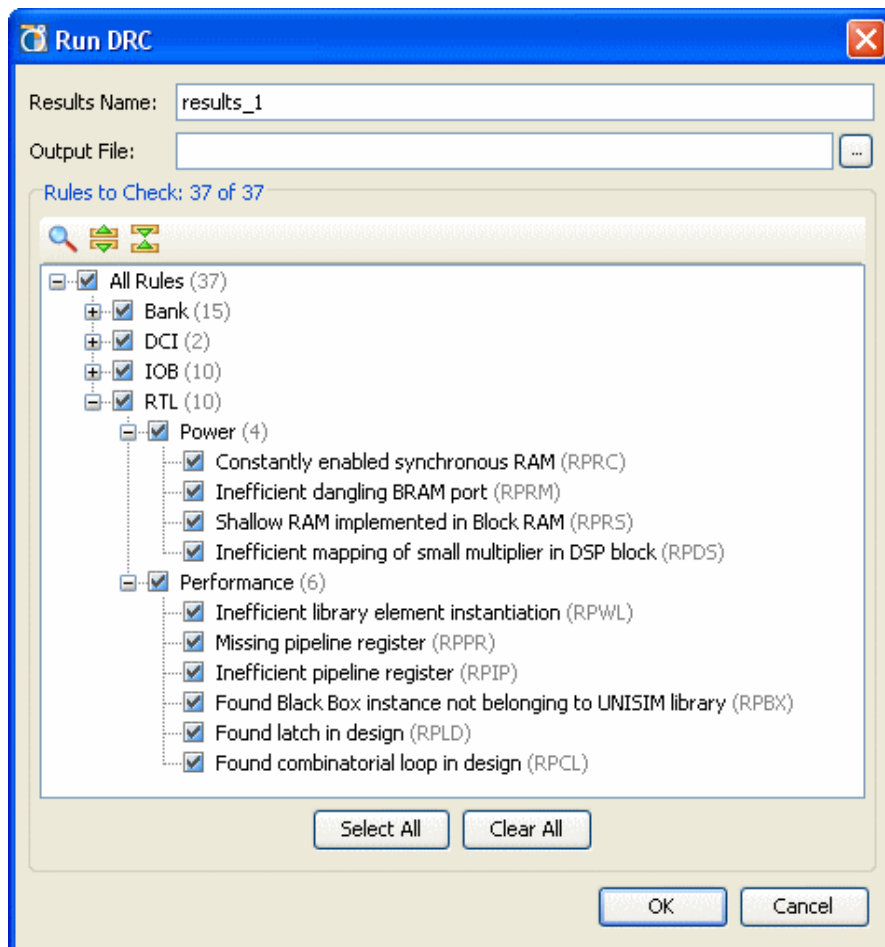


図 26 : RTL DRC の実行

[DRC Results] ビューが開きます (図 27)。

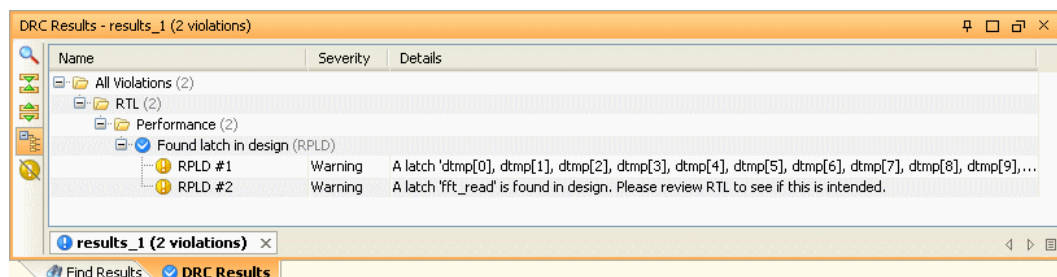


図 27 : RTL DRC 結果の表示

[RTL Results] ビューには、次の色でエラー、警告、情報メッセージがそれぞれ表示されます。

- エラーは赤いアイコン
- 警告はオレンジのアイコン
- 情報は黄色のアイコン

5-1-3. リストの **RPLD #1** ラッチの警告をクリックします。

[Violation Properties] ビューに違反に関する情報と該当するロジック オブジェクトを選択するリンクが表示されます。

5-1-4. [Violations Properties] ビューで **dtmp[0]** のリンクをクリックすると、[RTL Netlist] ビューでそのオブジェクトがハイライトされます。

5-1-5. [RTL Netlist] ビューで右クリックし、[Show Source] をクリックし (または **F7** キーを押し)、RTL Editor を開きます。

5-1-6. [DRC Results] ビューと開いている RTL Editor をすべて閉じます。

5-1-7. 開いている RTL デザインの X ボタンをそれぞれクリックして閉じ、それを確認するダイアログ ボックスで [OK] をクリックします。

ザイリンクス IP カタログからの IP の選択

手順 6

PlanAhead は CORE Generator ソフトウェア ツールと連動しており、CORE Generator では検索やフィルタ機能を持つ IP カタログで必要な IP を簡単に見つけることができます。IP を見つけたら、それを PlanAhead から直接カスタマイズ、インスタンシエート、インプリメントできます。IP カタログには、Project Manager と RTL デザイン環境のどちらからでもアクセスできます。

6-1. IP カタログを開いて、検索オプションを確認します。

6-1-1. Flow Navigator で **[IP Catalog]** をクリックします。

6-1-2. IP カテゴリを展開表示してみます。

6-1-3. IP を選択して、使用可能なツールバー ボタンやポップアップ メニューを確認してみてください (図 28)。

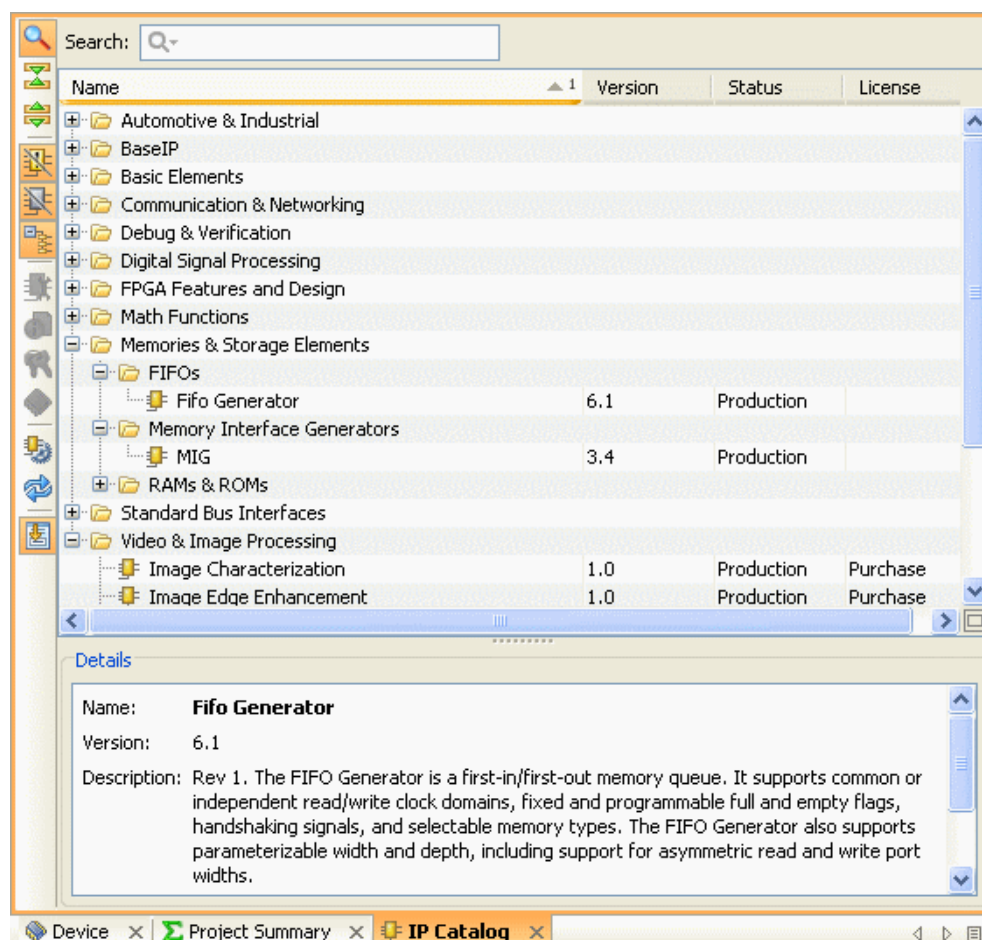





図 28 : IP カタログの検索

- 6-1-4. 一番下の [Details] フィールドには選択した IP の詳細が表示されます。デフォルトでは、選択したデバイスに使用可能な IP のみが表示されます。


メモ : すべての IP を表示するには、[Hide non production IP] () と [Hide incompatible IP] () ツールバー ボタンを切り替えます。平坦化した IP のリストを表示するには、[Group by Category] ツールバー ボタンで切り替えます。

- 6-1-5. 一番上の [Search] フィールドに **fir** と入力します。
- 6-1-6. [FIR Compiler IP] を選択し、[Data Sheet] ボタン () をクリックします。
- 6-1-7. しばらくするとデータシートが表示されます。確認したら、PDF を閉じます。
- 6-1-8. [Search] フィールドをクリアにしたら、カタログリストを展開表示します。

IP のカスタマイズおよびインスタンス化

手順 7

7-1. 単純な加算器 IP をカスタマイズします。

7-1-1. [Hide incompatible IP] ツールバー ボタン () をクリックし、IP フォルダの [Math Functions] を表示します。

7-1-2. [Math Functions] → [Adders & Subtractors] フォルダを展開表示します。

7-1-3. [Adder Subtractor] をダブルクリックすると、[Customize IP] コマンドが実行されます。

これにより、CORE Generator ツールが起動され、選択した IP のカスタマイズ インターフェイスが表示されます。表示されるインターフェイスは IP によって異なります (図 29)。

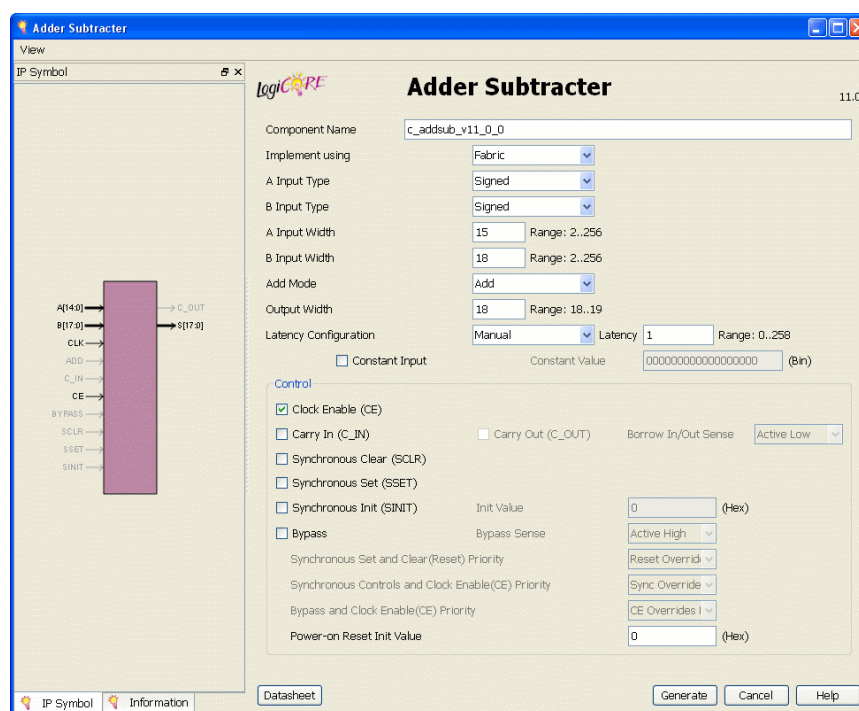


図 29 : CORE Generator を使用した IP のカスタマイズ

7-1-4. [B Input Width] フィールドを 18 にします。

7-1-5. [Generate] をクリックします。

[Generate] ボタンをクリックしたときの動作は、PlanAhead から CORE Generator を起動した場合と、CORE Generator をスタンドアロンで実行した場合で異なります。スタンドアロン モードの場合、CORE Generator は自動的に XST を起動して IP コアを合成します。PlanAhead から起動した場合、合成は自動的に実行されないため、合成を実行する前に RTL でコアをインスタンス化したりコンフィギュレーションした

ることができます。IP の合成はいつでも実行できます。デザインの合成を実行する場合は、まず IP が自動的に合成されます。

7-2. 加算器 IP をインスタンス化します。

7-2-1. [Sources] ビューで [Collapse All] ボタン () をクリックして、IP フォルダを展開表示します。

7-2-2. IP コアの **c_addsub_v11_0_0** を展開します。

7-2-3. **c_addsub_v11_0_0.veo** ファイルをダブルクリックすると、RTL Editor でそのインスタンス化テンプレートが表示されます (図 30)。

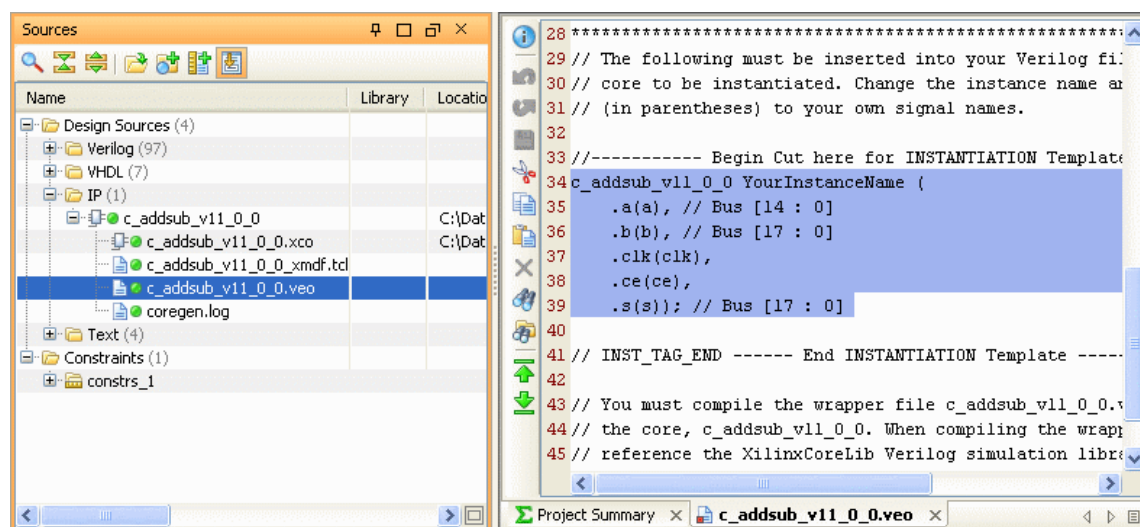


図 30 : インスタンス化テンプレートの表示

7-2-4. 上の図のように RTL Editor でテキストを選択し、[Copy] ボタンをクリックします。

7-2-5. [Sources] ビューで [Verilog] フォルダを展開表示します。

7-2-6. スクロールダウンして top.v ファイルをダブルクリックし、そのファイルを RTL Editor で開きます。ファイルの一番下の endmodule テキストの手前までスクロールダウンします。

7-2-7. endmodule のすぐ上の行を選択し、[Paste] ボタンをクリックします。

7-2-8. テンプレートの *YourInstanceName* を my_adder に変更します (図 31)。

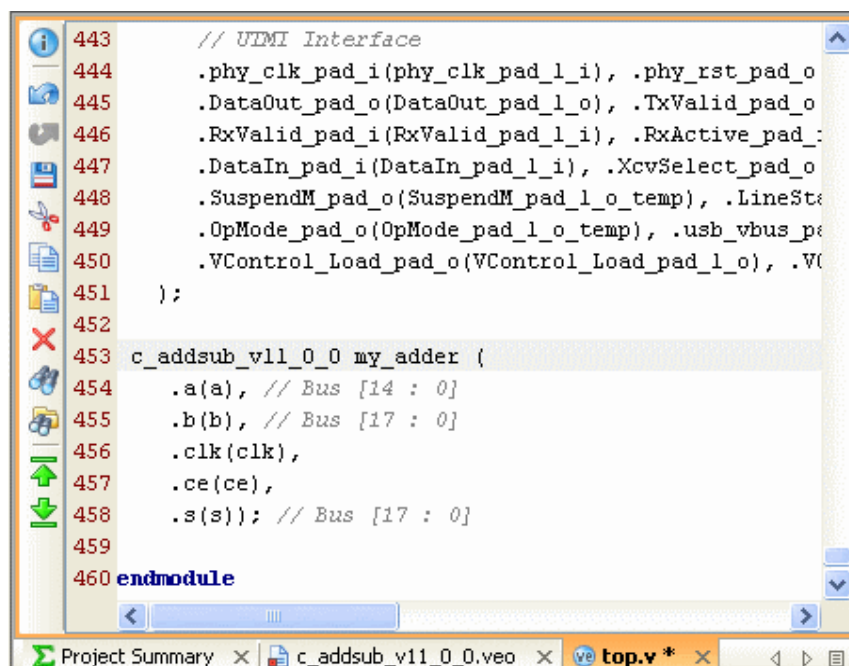
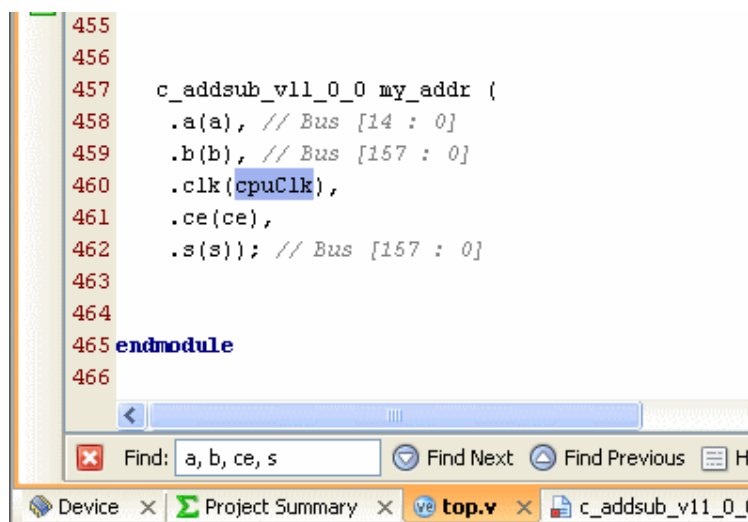


図 31 : IP のインスタンス化

7-2-9. `.clk` のポート定義を既存の `cpuClk` クロック信号を使用するように変更します (図 32)。

図 32 : `cpuClk` クロック信号を使用するように `clk` 信号を変更

7-2-10. `top.v` ファイルの一番上までスクロールし、モジュールポート定義に IP ポート (`a`, `b`, `ce`, `s`) を追加します (図 33)。

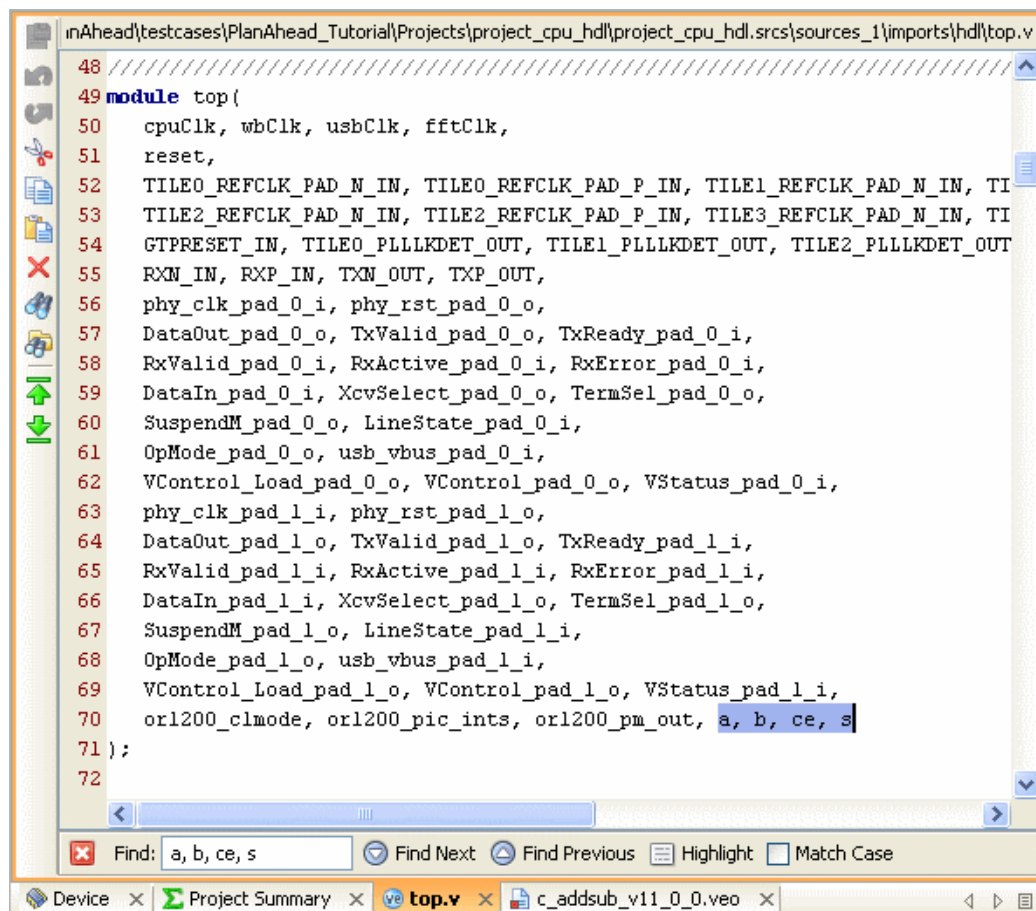


図 33 : 最上位レベル モジュールのポート リストへ IP ポートを追加

7-2-11. top.v ファイルで新しいポートを定義します。次のテキストを追加します (図 34)。

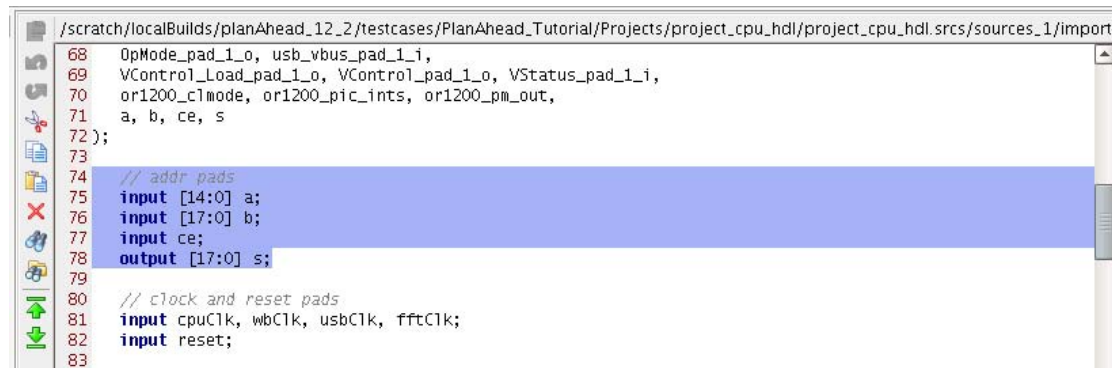


図 34 : top.v ファイルでの IP ポートの定義

7-2-12. top.v ファイルは、図 34 と同じようになるはずです。

7-2-13. タブで X ボタンをクリックして top.v ファイルを閉じ、変更を保存するかどうか尋ねられたら **[Yes]** をクリックします。

7-2-14. タブの X ボタンをクリックして VEO テンプレートファイルを閉じます。

7-2-15. タブの X ボタンをクリックして IP カタログを閉じます。

IP の生成

手順 8

8-1. IP を生成し、回路図でそのロジックを確認します。

8-1-1. [Sources] ビューで `c_addsub_Vxx_x.xco` ファイルを右クリックし、[Generate IP] を選択します。IP が合成されるまでお待ちください。

IP 生成後は、RTL デザインをアップデートする必要があります。

8-1-2. 黄色のバナーの [Reload] リンクをクリックします。

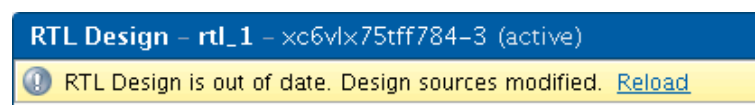


図 35 : RTL デザインの再読み込み

8-1-3. [RTL Netlist] ビューで `my_adder` モジュールを展開表示して、選択します。

メモ: [RTL Netlist] ビューでブラック ボックスのアイコンが表示されたら、[Elaboration Messages] ビューを確認し、ファイルに戻ってエラーをチェックします。IP が [RTL Netlist] ビューに表示されるまで [Elaborate] を再実行します。

8-1-4. ツールバーのポップアップ メニューから [Schematic] を選択します。

8-1-5. [Schematic] ビューでインスタンスをダブルクリックして、ロジックを展開表示します。

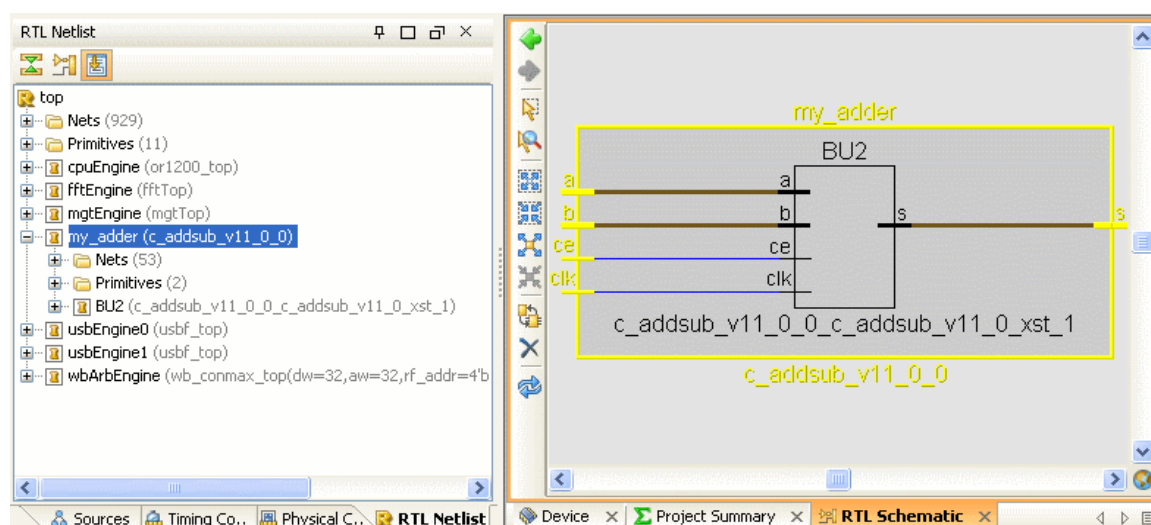


図 36 : 回路図での IP ロジックの解析

8-1-6. [Schematic] ビューを閉じます。

8-1-7. [File] → [Exit] をクリックします。変更を保存するかどうか尋ねるメッセージが表示されたら [No] をクリックし、[OK] をクリックして PlanAhead を閉じます。

まとめ

このチュートリアルでは、小型の RTL プロジェクトを使用して、PlanAhead の RTL 開発と解析環境について説明しました。ここでは、まず RTL プロジェクトを作成し、RTL ソースと RTL Editor を確認しました。RTL デザインをエラボーレーションし、解析機能を確認しました。解析機能には、RTL ロジック階層、RTL 回路図、ロジックタイプの検索、RTL リソース概算および RTL DRC の実行などが含まれます。そのあと、ザイリンクスの IP カタログを確認し、小型の加算器 IP コアをカスタマイズし、インスタンス化してインプリメントしました。