PlanAhead ソフトウェア チュートリアル

デザイン解析およびフロアプラン

UG 676 (v 12.1) 2010 年 5 月 3 日

XILINX®

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 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

デザイン解析およびフロアプラン 7
ザイリンクス ISE および PlanAhead ソフトウェア 8
ザイリンクス コマンド ライン ツール 8
ハードウェア要件 8
PlanAhead のマニュアルと情報 8
チュートリアルの目標 8
チュートリアルの手順 9
デバイス リソースとクロック領域の表示　手順 1 10
論理ネットリストの階層表示　手順 2 14
デザイン リソース統計の表示　手順 3 17
デザイン ルール チェック (DRC) の実行　手順 4 20
タイミング解析の実行　手順 5 23
回路図でのロジックの表示　手順 6 29
デザイン制約の確認　手順 7 35
デザインのインプリメンテーション　手順 8 39
タイミング結果の解析　手順 9 42
モジュール レベルの配置のハイライト　手順 10 46
接続の確認　手順 11 48
配置制約の使用　手順 12 50
階層接続の表示　手順 13 53
検索機能を使用したクロック ドメインの表示　手順 14 59
タイミング クリティカルな階層のフロアプラン　手順 15 61
まとめ 64

PlanAhead ソフトウェア チュートリアル

デザイン解析およびフロアプラン

はじめに

この演習では、高度な FPGA を設計するのに必要なザイリンクスの PlanAhead™ ソフトウェアの機能や利点を簡単に紹介します。このチュートリアルでは、より短い時間で効率の良いデザインを作成するための主な手順を説明します。このチュートリアルでは、次を実行します。

- インプリメンテーション前処理と解析
- インプリメンテーション機能の確認
- インプリメンテーション結果のフロアプラン

メモ：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。その他の機能については、別の PlanAhead チュートリアルで説明します。

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

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**

インストール プロセスが実行されると、上記のディレクトリに ZIP ファイルが抽出されます。この ZIP ファイルを書き込み権のあるディレクトリに解凍しておいてください。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。

メモ：ZIP ファイルをデフォルトの <*ISE_install_Dir*>/PlanAhead/testcases/PlanAhead_Tutorial ディレクトリでそのまま解凍すると、最初に表示される Getting Started ページの **[Open an Example Project]** のリンクをクリックするだけで、チュートリアルのデザインを開くことができます。チュートリアルを実行すると、サンプル デザイン データは変更されます。

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。このチュートリアルでは、インプリメンテーション プロセスをすべて実行する必要があります。

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。フロアプランによるパフォーマンス改善は、通常大型のデザインの方がわかりやすいので、このデザインではあまり改善されていないように見えないかもしれません。

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、ISE 12.1 ソフトウェア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_1/irn.pdf

このチュートリアルを実行するには、ザイリンクス ISE の配置配線ソフトウェアを使用できるようにしておく必要があります。チュートリアルの結果は、ほかの ISE リリースとは異なる可能性があります。

ザイリンクス コマンド ライン ツール

このチュートリアルを実行するには、NGDBuild、MAP、PAR、TRACE、XDL などのザイリンクス コマンド ライン ツールを使用できるようにしておく必要があります。このチュートリアルには、 前もって ISE で実行された結果が既に含まれています。

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 XC6VLX75T デザインを使用し、1 度に開くことができるプラン数を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でることもあります。

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_1/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_1/Floorplanning_Methodology_Guide.pdf

PlanAhead を使用する利点やほかのチュートリアルなどについて説明するビデオ 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。

チュートリアルの目標

このチュートリアルは、PlanAhead ソフトウェアのさまざまな解析、フロアプラン、インプリメンテーションの機能を説明するためのもので、次のトピックが含まれます。

- デバイス使用率の統計を解析してターゲット デバイスを最適なデバイスに変更
- デザイン ルール チェック (DRC) を実行して、インプリメンテーション エラーになる可能性のある制約の競合を素早く解消
- [Netlist]、[Logic Hierarchy]、[Schematic] ビューでロジックを確認
- タイミング パフォーマンスを素早く概算することで、デザインの実現可能性を評価し、潜在的に問題のあるエリアを識別
- デザインに含まれる制約を表示、作成、変更
- デザイン階層接続およびデータ フローを解析し、クリティカルなロジック接続およびクロック領域を識別
- タイミング クリティカルなロジックをフロアプランしてタイミングを改善

このチュートリアルを終了したら、PlanAhead ソフトウェアのプロセスや機能に注目し、これらの機能をユーザー デザインでどのように使用するか決定してください。

チュートリアルの手順

チュートリアルは各手順に分けられ、それぞれで大まかな手順が説明された後、細かい手順が説明されていますので、スキル レベルに合った方の手順を参照してください。

まず、最初に PlanAhead ソフトウェアのインプリメンテーション前のデザイン解析機能について説明します。これらの機能を使用すると、デザインの潜在的な問題を早期に検出したり、代替デバイスを探したり、フロアプランしたりできます。

このチュートリアルは、次の手順で構成されています。

手順 1 : デバイス リソースとクロック領域の表示

手順 2 : 論理ネットリストの階層表示

手順 3 : デザイン リソース統計の表示

手順 4 : デザイン ルール チェック (DRC) の実行

手順 5 : タイミング解析の実行

手順 6 : 回路図でのロジックの表示

手順 7 : デザイン制約の確認

手順 8 : デザインのインプリメンテーション

手順 9 : タイミング結果の解析

手順 10 : モジュール レベルの配置のハイライト

手順 11 : 接続の確認

手順 12 : 配置制約の使用

手順 13 : 階層接続の表示

手順 14 : 検索機能を使用したクロックドメインの表示

手順 15 : タイミング クリティカルな階層のフロアプラン

大まかな手順でわからない場合はその後の詳細な手順を参照してください。既に手順を理解している場合は、その部分は飛ばして次の手順に進んでください。

デバイス リソースとクロック領域の表示　　手順 1

1-1. PlanAhead ソフトウェアを起動します。

1-1-1. CPU (Synthesized) プロジェクトを top_full 制約セットを使用して開きます。

- Windows の場合、Xilinx PlanAhead 12 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 12.1] → [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 環境が開きます (図 1)。

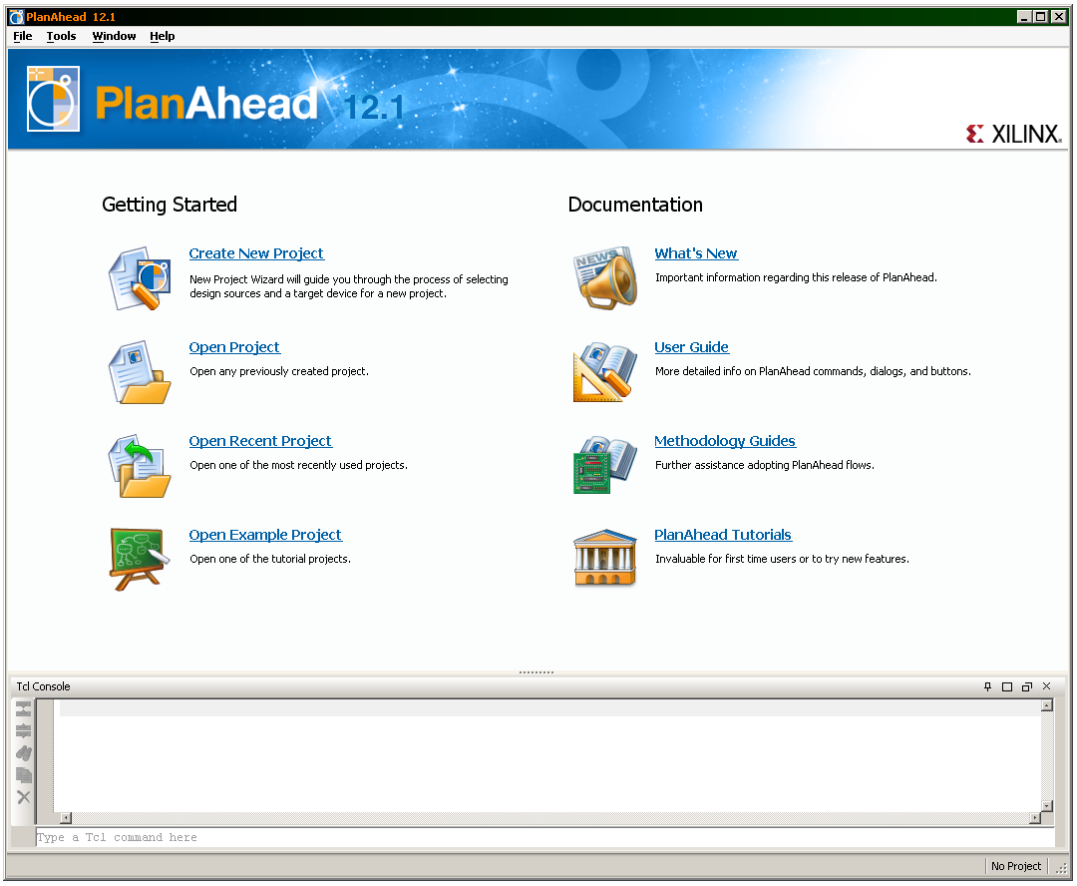


図 1 : PlanAhead Getting Started ページ

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る PlanAhead の Getting Started ページが表示されます。

1-1-2. [File] → [Open Example Project] → [CPU (Synthesized)] をクリックします。

PlanAhead 環境が開きます (図 2)。

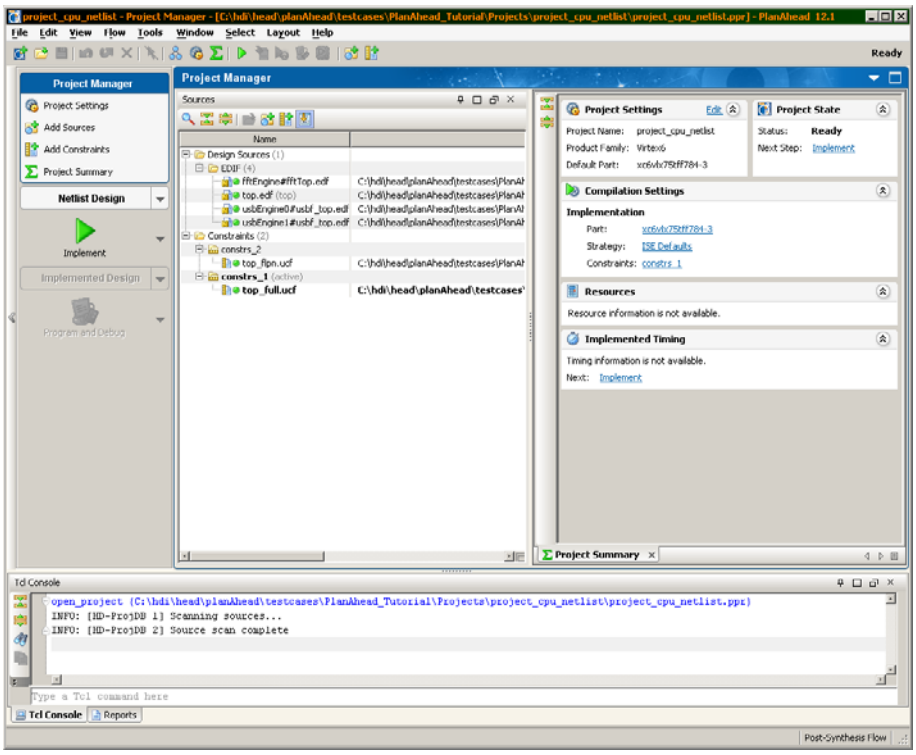


図 2 : PlanAhead 環境

1-1-3.　PlanAhead のメイン ウィンドウで [Constraints] → [constrs_1] → [top_full.ucf] がアクティブになっていることを確認します (図 3)。

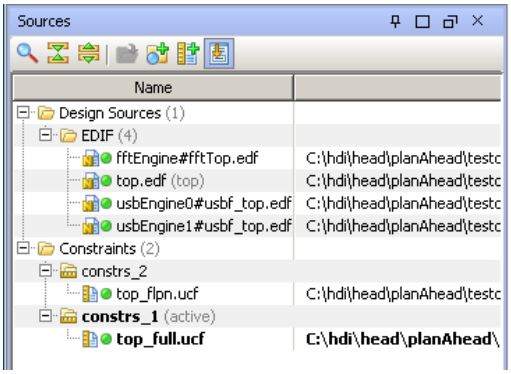


図 3 :Constrs_1 が選択された [Sources] ビュー

1-1-4.　Flow Navigator で [Netlist Design] をクリックしてデザインと制約を開きます (図 4)。

図 4 : [Netlist Design] ボタン

1-1-5.　次のようにネットリストと top_full.ucf が開きます (図 5)。

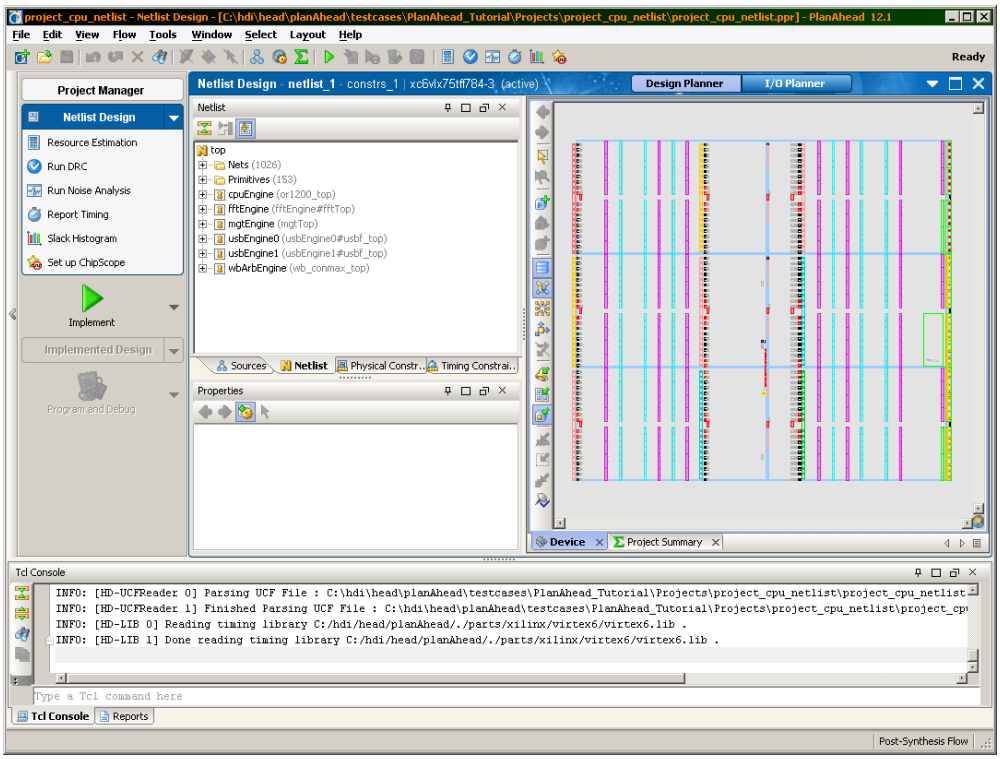


図 5 : [Netlist Design] ビューの CPU ネットリスト

1-1-6.　次のように、[Device] ビューで [Zoom Area] を使用してさまざまなデバイス リソースを確認します。

デバイスの左上をクリックし、右下に向かってドラッグします。描画した箇所が拡大されます。上記の手順を繰り返し、デバイス リソースが確認できる大きさまで拡大します (図 6)。

図 6 : 拡大した [Device] ビューでの CLB とロジック ゲート サイトの表示

1-1-7.　さまざまなデバイス リソースの上にカーソルを置いて、ツールチップを表示します。 次に注意してください。

- SLICE 座標 は PlanAhead メイン ウィンドウ一番下のステータス バーに表示される
- カーソルの位置が同じサイト内にある場合は、表示されるツール ヒントも同じ

- I/O ポートの位置と I/O バッファの割り当ては I/O バンク内に表示される
- 縦長の赤いタイルは RAMB36 サイトを示し、2 つの RAMB18 または 1 つの FIFO を含むことも可能
- 縦長の青いタイルは 2 つの DSP48 のサイトを示す
- 青い正方形にはそれぞれ 2 つの SLICE を含む CLB を示す

1-2. サイトのどれかをクリックすると、[Site Properties] ビューにそのサイトに関する情報が表示されます。

1-2-1. [Site Properties] のタイトル バーをダブルクリックし、ウィンドウを最大化します。または [Maximize] ボタン ( ) をクリックします。

1-2-2. [Site Properties] のタイトル バーをダブルクリックし、ウィンドウを元に戻します。または [Restore] ボタン ( ) をクリックします。

1-2-3. その他のサイト タイプをクリックして、[Site Property] を表示してみます。

1-2-4. [Zoom Fit] ボタンをクリックし、[Device] ビューにデバイス全体を表示させます。マウスをクリックし、左上に向かってドラッグするか、Ctrl キーを押しながら G キーを押します。

[Device] ビューにクロック領域が表示され、フロアプランをしやすくなります。[Clock Region] ビューには、デバイスのクロック領域がすべて表示されます。[Clock Region Properties] ビューで、インプリメンテーション前にクロックに潜在的な競合がないかどうかを確認します。クロック ドメインの表示については、このチュートリアルの後半で説明します。

1-3. [Clock Regions] ビューを開きます。

1-3-1. [Device] ビューで次の手順を使用してクロック領域を表示します。

- [Windows] → [Clock Regions] をクリックします。
- 表中のクロック領域の 1 つをクリックします。

選択したクロック 領域が [Device] ビューでハイライトされます (図 7)。

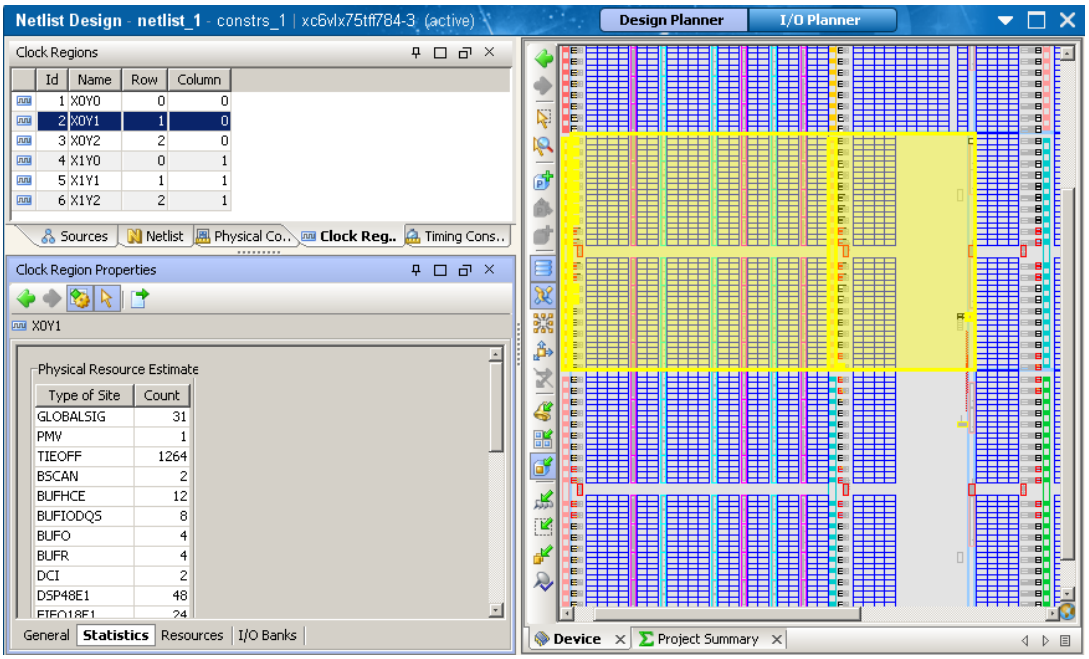


図 7 : クロック領域プロパティの表示

1-3-2. [Clock Region Properties] ビューの **[Properties]** ビューをクリックします。

1-3-3. **[Statistics]** タブをクリックし、スクロール ダウンしてロジックの内容を確認します (図 7 の一番下)。

1-4. [Resources] タブをクリックし、ロジックの内容を確認します

1-4-1. [Resources] リストから **BUFR** の 1 つをクリックすると、それが [Device] ビューでもハイライトされます。

1-4-2. 右クリックで **[Mark]** を選択してそのサイトをマークします。

1-4-3. [Fit Selection] ツールバー ボタン ( ) をクリックし、選択したオブジェクトが表示されるようにします。

1-4-4. [Unmark All] ツールバー ボタン ( ) をクリックし、マークを取ります。

1-4-5. [Zoom Fit] ボタンをクリックし、[Device] ビューにデバイス全体を表示させます。

1-4-6. [Clock Region Properties] ビューの **[I/O Banks]** タブをクリックし、そのクロック領域に関連する I/O バンクを確認します。

1-4-7. [Clock Region Properties] ビューで I/O バンクの 1 つをクリックすると、それが [Device] ビューでもハイライトされます。

1-4-8. **[I/O Planner]** ボタン (I/O Planner) をクリックします。

メモ :このビューは、ピン配置がしやすいように変更され、[Package] ビューが開くようになりました。I/O バンクは、[Device] ビューと [Package] ビューの両方でハイライトされます。

1-4-9. **[Design Planner]** ボタン (Design Planner) をクリックします。

論理ネットリストの階層表示　手順 2

2-1. [Netlist] ビューでデザイン階層を確認します。

2-1-1. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

2-1-2. cpuEngine の横にあるプラス記号 (+) をクリックしてモジュールを展開します。

[Netlist] ビューは次のように表示されるはずです (図 8)。

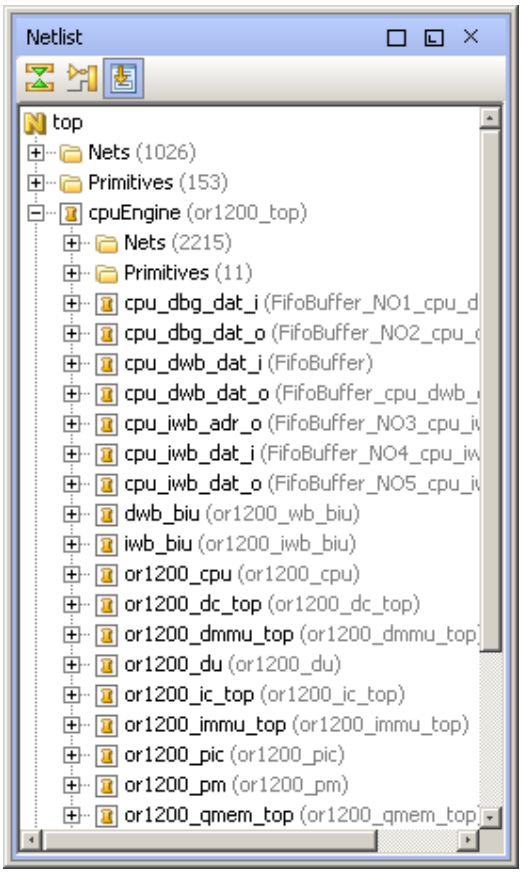


図 8 : [Netlist] ビューの展開表示

メモ : [Netlist] ビューの [Primitives] フォルダには、各モジュールの最上位レベルのインスタンスが含まれます。

2-1-3. [Primitives] フォルダを展開表示します。

2-1-4. サブ モジュールではなく、cpuEngine レベルにインスタンスがあります。

2-1-5. [Nets] フォルダを展開表示します。ネットはモジュール内にあります。

2-1-6. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

2-2. ネットリスト モジュールを選択し、ロジックがデザイン階層のどこにあるかを確認します。

2-2-1. [Netlist] ビューで usbEngine0 モジュールを展開表示します。

2-2-2. **u4** モジュールを選択します。

2-2-3. 右クリックし、[Show Hierarchy] を選択します。

2-2-4. ワークスペースに [Hierarchy] ビューが開きます (図 9)。

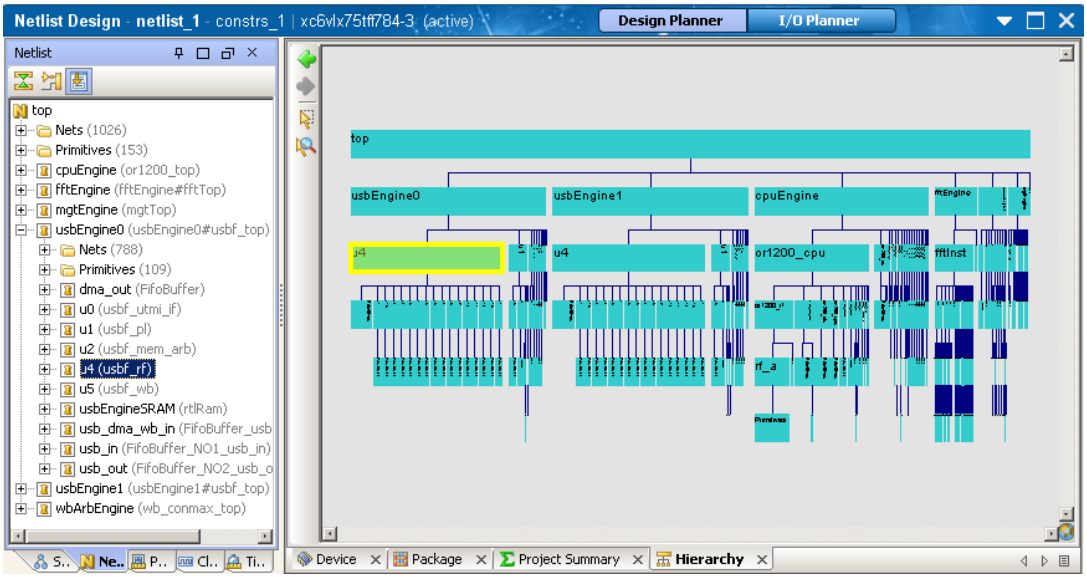


図 9 : インスタンスの階層表示

[Hierarchy] ビューには、モジュールの階層関係や相対的なサイズが表示されます。選択したロジック モジュールが [Hierarchy] ビューに表示され、フロアプラン前にモジュールの位置および相対的なサイズを視覚的に確認できます。この表示から直接モジュールを選択してフロアプランができます。[Hierarchy] ビューには、ほかのビューで選択したロジックもハイライトされます。

2-2-5.　階層表示でモジュールを選択すると、[Netlist] ビューでも選択されます。

2-2-6.　[Unselect All] ツールバー ボタン ( ) をクリックするか、F12 キーを押します。

デザイン リソース統計の表示　　手順 3

PlanAhead ソフトウェアの使用率解析を実行すると、デザインに対して最適なデバイスを決定するのに役立つデザイン統計が表示できます。モジュール間でロジック リソースがどのように分配されているか確認することもできます。複数のデバイス タイプを試すことができるので、最適な使用率やパフォーマンスを決定しやすくなっています。

3-1. ROOT レベルのデザイン リソースの概算を確認します。

3-1-1. Flow Navigator で **[Netlist Design]** → **[Resource Estimation]** をクリックし、[Resource Estimation] ボタン (Resource Estimation) をクリックします。

[Resource Estimation] がワークスペースに開きます。

3-1-2. **[Resource Estimation]** タブをダブルクリックし、ウィンドウを最大化します。[Resource Estimation] タブを右クリックし、**[Maximize]** を選択しても最大化できます。[Resource Estimation] ビューに階層ごとのリソース使用率が表示されます (図 10)。

ワークスペースのタブの右上には、最大化ボタンやその他のウィンドウ操作ボタンが表示されません。これらは、右クリックで使用できるようになっています。

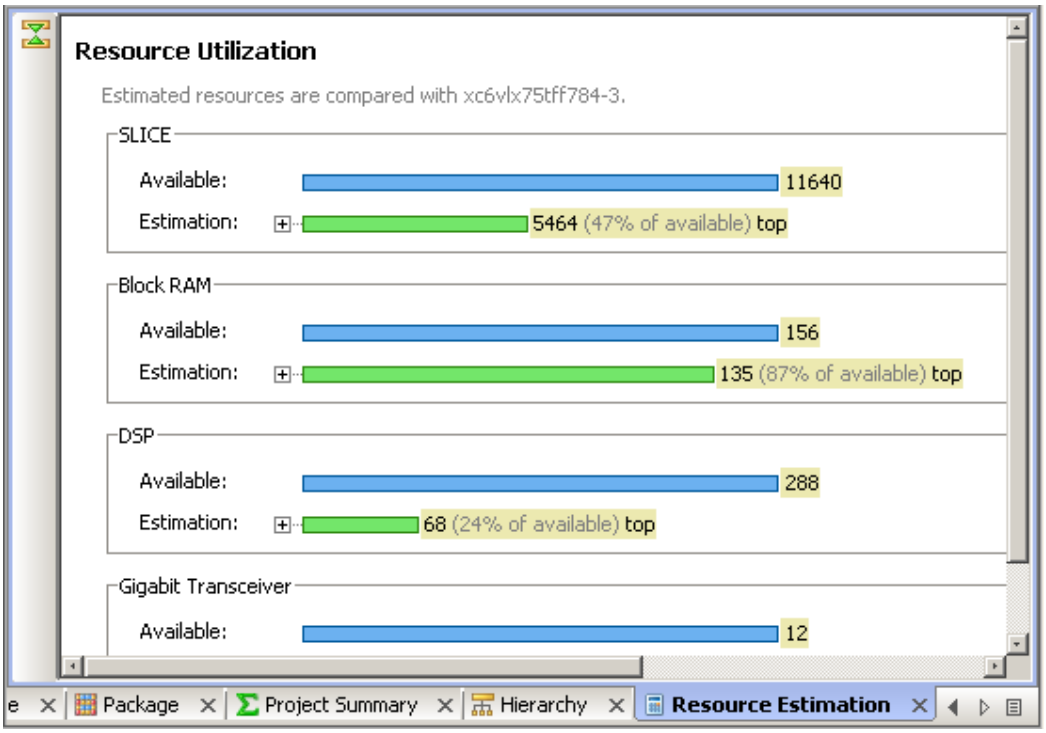


図 10 : [Resource Estimation] ビュー

3-1-3. [Resource Estimation] ビューで [Block RAM] → [Estimation] の横の ⊞ をクリックして展開表示します。ブロック RAM を主に消費しているのは、usbEngine0 および usbEngine1 の 2 つであるのがわかります。

3-1-4. その他のリソースも展開表示してみます。

3-1-5. [Block RAM Estimation] セクションから usbEngine1 をクリックします。

3-1-6. [**Hierarchy**] タブをクリックします。

メモ : 階層ブラウザでも usbEngine1 が選択されます。1 つビューで選択したものは、ほかのビューでも選択されます。

3-1-7. Hierarchy の X ボタンをクリックして [Hierarchy] ビューを閉じます。

3-1-8. Resource Estimation の X ボタンをクリックして [Resource Estimation] ビューを閉じます。

3-1-9. [Layout] → [Load Layout] → [PlanAhead Default] をクリックし、デフォルトのレイアウトに戻します。

3-2. 左上の [Physical Constraints] ビューをクリックします。

3-2-1. [**Physical Constraints**] タブをクリックします。

3-2-2. ROOT デザインを選択します (図 11)。

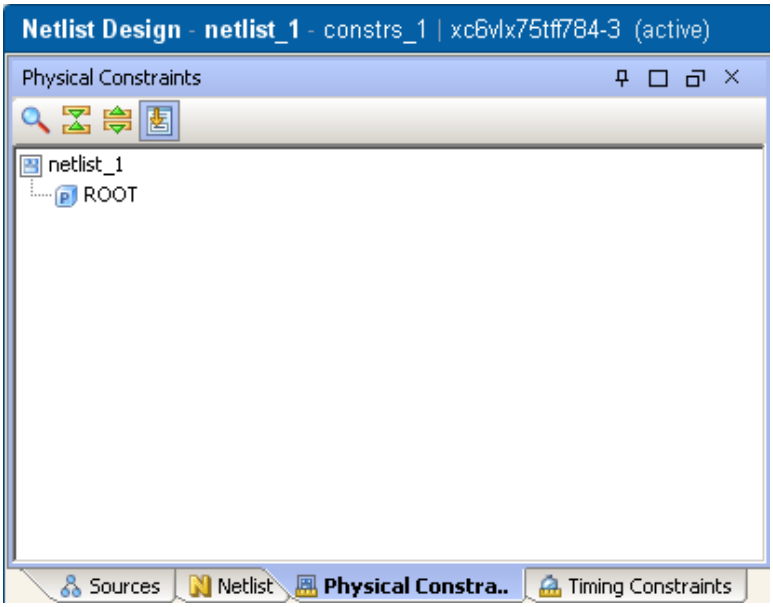


図 11 : [Physical Constraints] ビュー

3-2-3. [**Pblock Properties**] ビューをクリックします。

3-2-4. **[Statistics]** タブがオンになっていない場合はクリックします。

3-2-5. [Pblock Properties] ビューの [Physical Resources Estimates] フィールドを確認します (図 12)。

3-2-6. [**Pblock Properties**] ビューを最大化します。

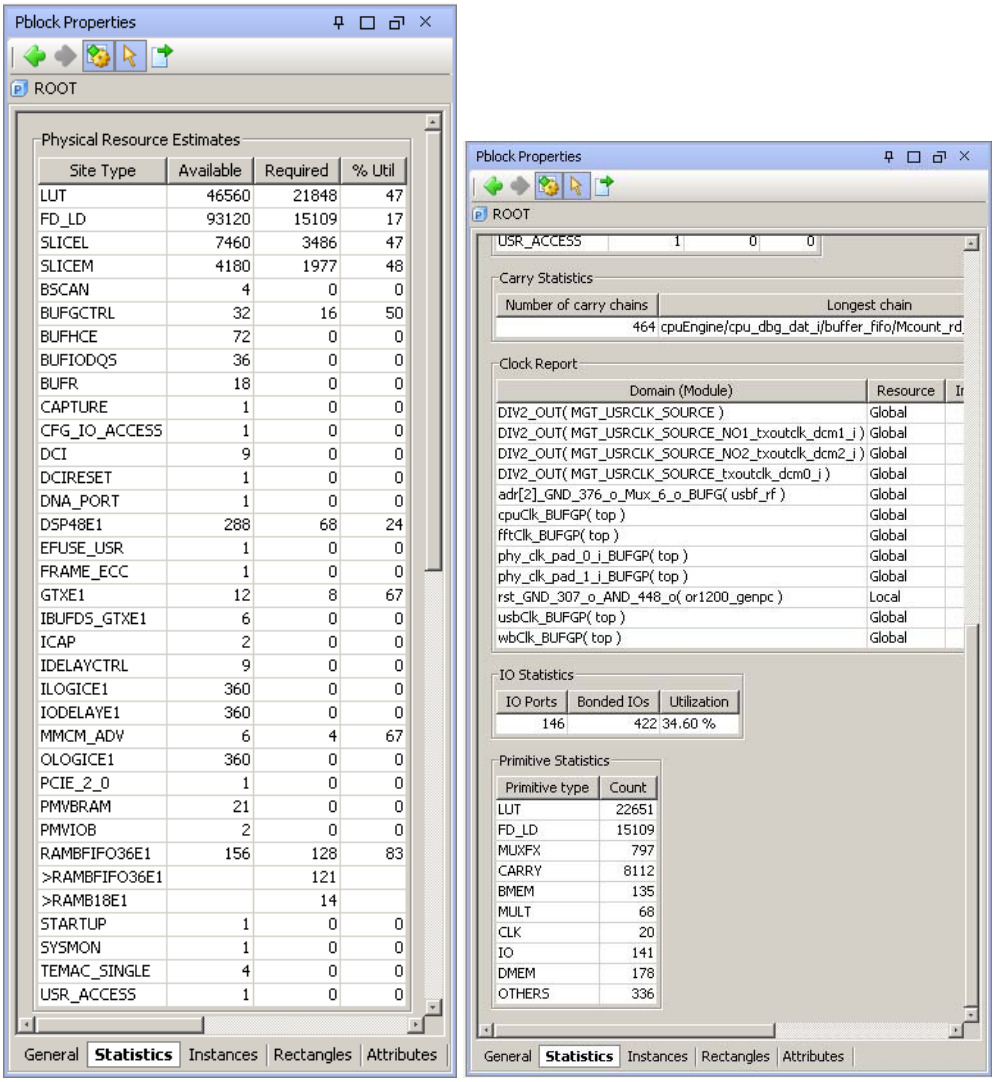


図 12 : デザイン リソース統計の表示

3-2-7.　[Pblock Properties] ビューのデザイン統計をスクロール ダウンします。

メモ : [Statistics] タブには、ロジック エレメント、キャリー チェーン数と最長チェーンの長さ、クロック レポート、I/O 使用率、プリミティブ インスタンスとインターフェイス ネット数などが種類別に表示されています。デザインに RPM が含まれる場合は、RPM の数と最大サイズなどの情報も表示されます。

3-2-8.　[**Pblock Properties**] ビューを元の大きさに戻します。

デザイン ルール チェック (DRC) の実行　　手順 4

インプリメンテーション前にデザイン ルール チェックの実行をお勧めします。

4-1. デザイン ルール チェックを実行します。

4-1-1. [Tools] → [Run DRC] をクリックします。

[Run DRC] ダイアログ ボックスが表示されます (図 13)。

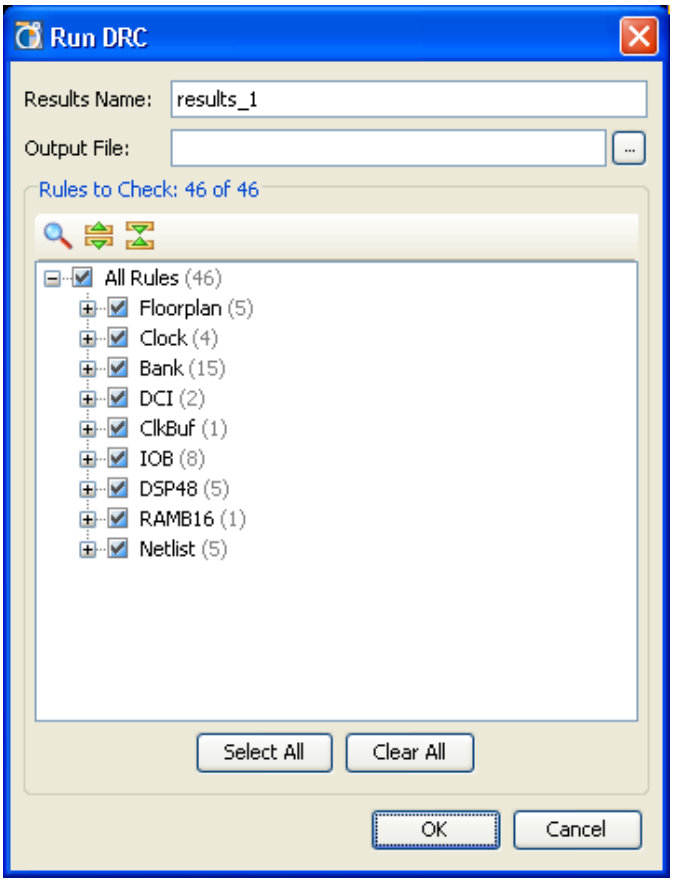


図 13 : DRC の実行

4-1-2. [OK] をクリックすると、すべてのルールがチェックされます。

[DRC Results] ビューに DSP48 に関する警告が複数表示されます (図 14)。

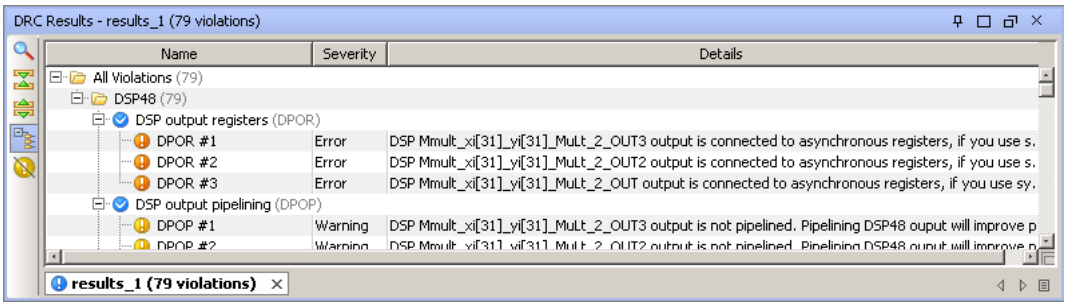


図 14 : デザイン ルール チェックの結果

エラー、警告、情報メッセージが [DRC Results] ビューに表示されます。エラーは赤いアイコン、警告はオレンジのアイコン、情報メッセージは黄色のアイコンで表示されます。DRC でエラーがレポートされても、インプリメンテーションは問題なく終了することあります。

4-1-3. [DRC Results] ビューで **DPOP #1** 警告をクリックします。

[Violation Properties] ビューが開き、違反が表示されます。

4-1-4. [Violations Properties] ビューで Mmult_xi[31]_yi[31]_MuLt_2_OUT3 をクリックします (図 15)。

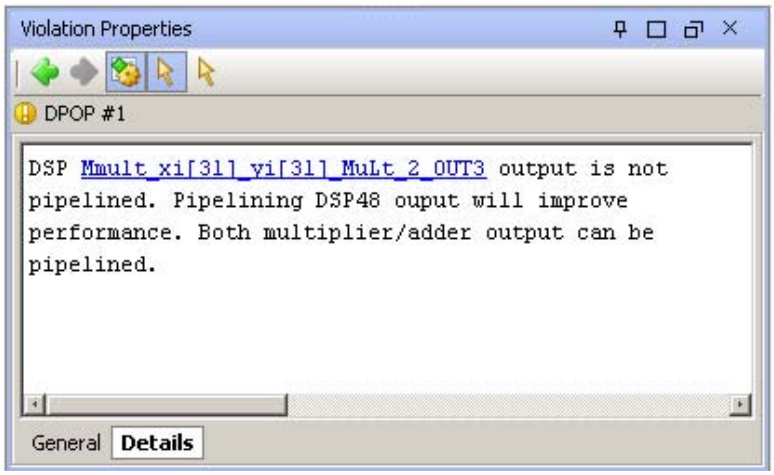


図 15 : [Violations Properties] ビュー

4-1-5. **[Netlist]** ビューをクリックします。

Mmult_xi[31]_yi[31]_MuLt_2_OUT3 が選択されています (図 16)。

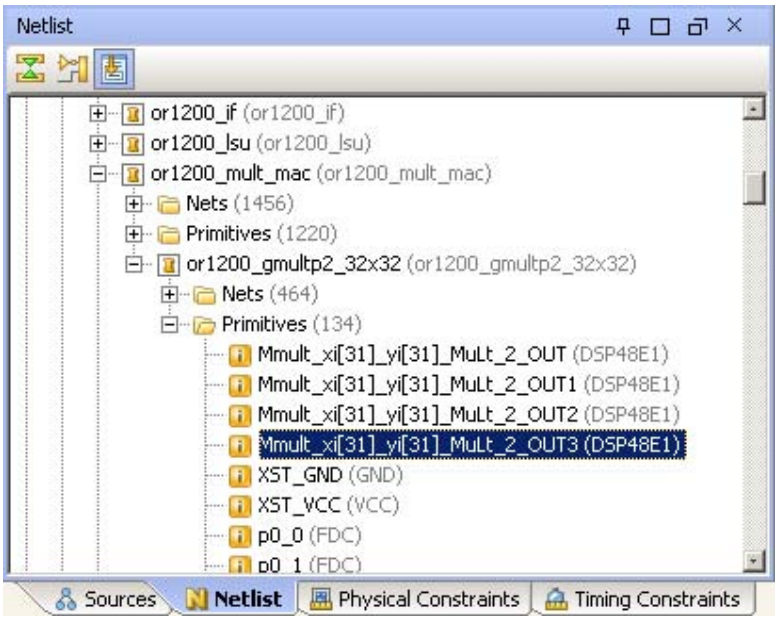


図 16 : [Netlist] ビューのインスタンス

4-1-6.　[DRC Results] ビューで違反リストをスクロール ダウンしてみます。

4-1-7.　X ボタンをクリックして [DRC Results] ビューを閉じます。

タイミング解析の実行　　手順 5

デザインのタイミング制約の実現可能性を指定するため、概算済み配線遅延を含めた早期スタティック タイミング概算を実行します。

5-1. スラック ヒストグラムでタイミング エンド ポイントを解析します。

スラックに基づいてエンドポイントが分類され、ヒストグラムに表示されます。

5-1-1. **[Tools] → [Slack Histogram]** をクリックします。

5-1-2. [Number of bins] を 20 に変更します。

[Generate Slack Histgram for Endpoints] ダイアログ ボックスは、図 17 のようになります。

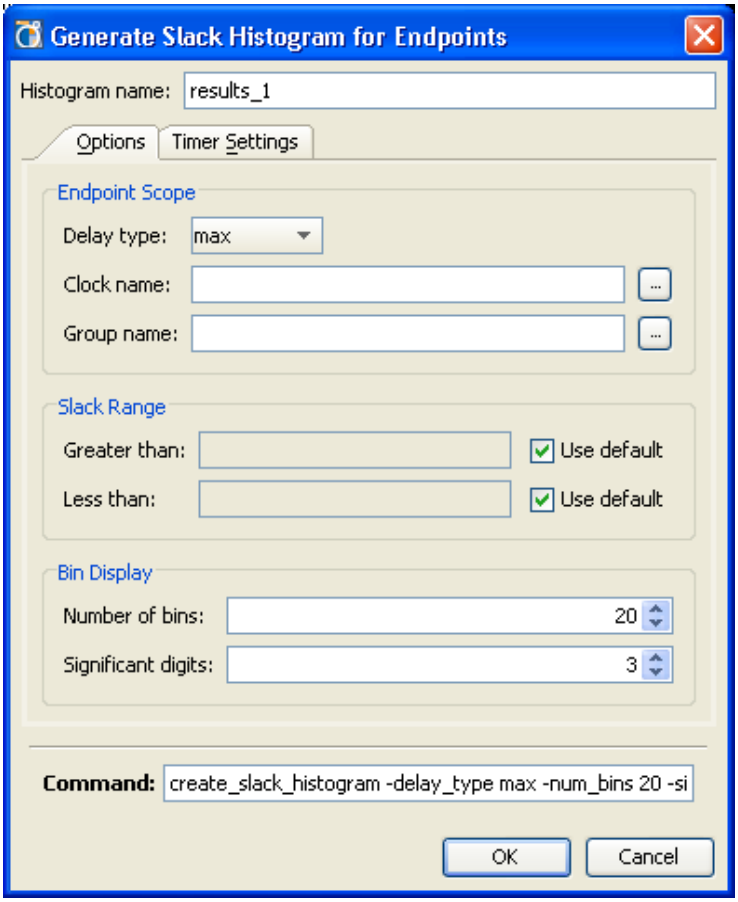


図 17 : [Generate Slack Histgram for Endpoints] ダイアログ ボックス

5-2. [Timer Settings] タブをクリックします。

5-2-1. 設定を確認してください。

メモ : [Timer Settings] タブでは、配線遅延の処理方法を指定できます。タイマでは配線遅延を概算したり、インターコネクト配線遅延がないものとして処理できます。

5-2-2. **[OK]** をクリックします。

ヒストグラムにエンドポイントが棒グラフで分類されて表示されます (図 18)。

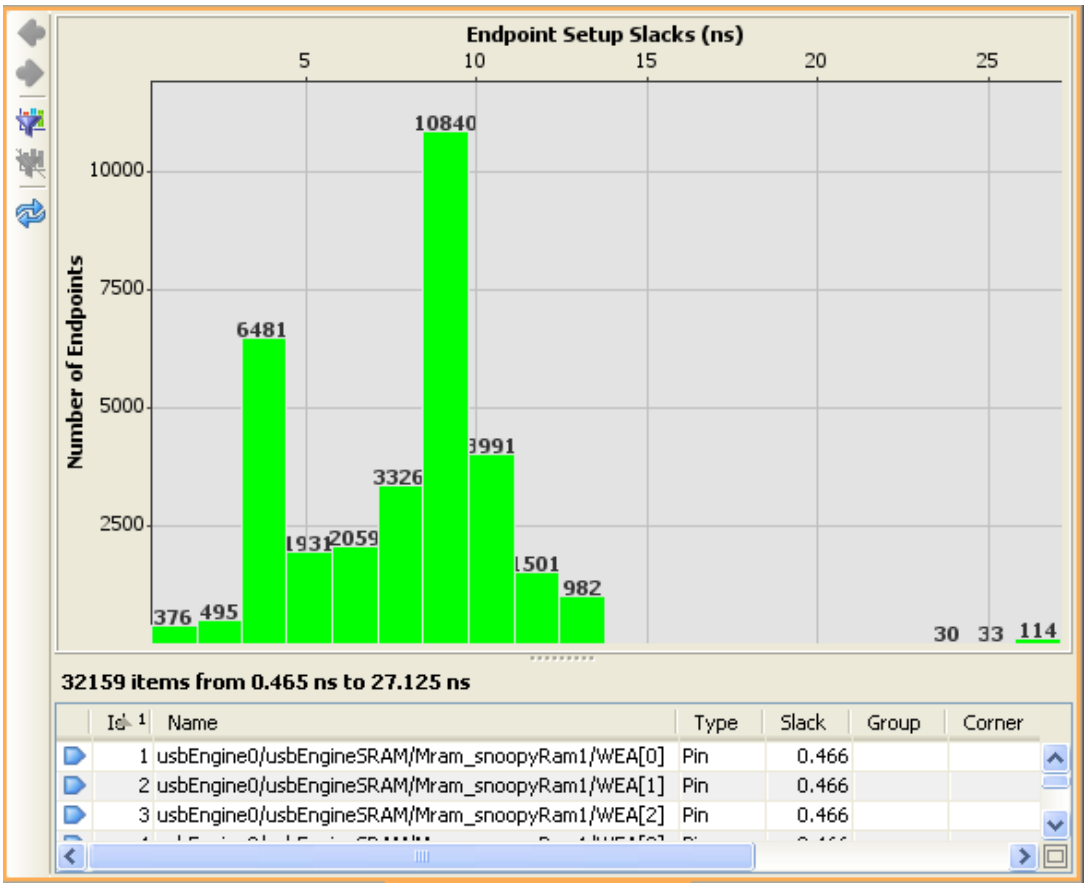


図 18 : スタック ヒストグラム

5-2-3. 棒グラフのさまざまな棒部分をクリックしてみます。

メモ :棒部分をクリックすると、下部のビューのエンドポイントも選択されます。スラック範囲もアップデートされます。

5-2-4. タイミング マージンの一番少ない一番左の棒部分をクリックします。

5-2-5. エンドポイントを確認してください。

メモ : ほとんどのタイミング エンドポイントは 2 つの usbEngine ブロックのブロック RAM であることがわかります。

5-2-6. フィルタされたヒストグラム レポートからインスタンスを選択します。[Netlist] ビューでもそれが選択されます。

5-2-7. Histogram - results_1 の X ボタンをクリックしてヒストグラムを閉じます。

5-3. [Timing Results] ビューの結果と情報を解析します。

5-3-1.　**[Tools] → [Report Timing]** をクリックします。

[Report Timing] ダイアログ ボックスの [Target] タブでは、タイミングのスタートポイントとエンドポイントを指定できます。

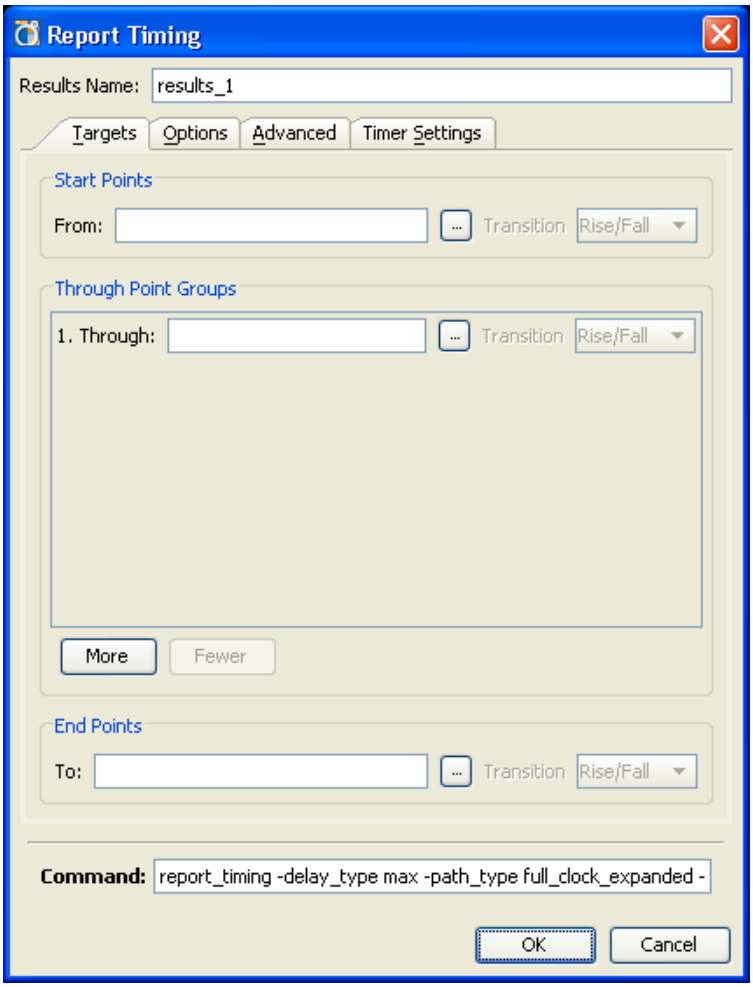


図 19 : タイミング解析の実行

5-4.　[Options] タブをクリックします (図 20)。

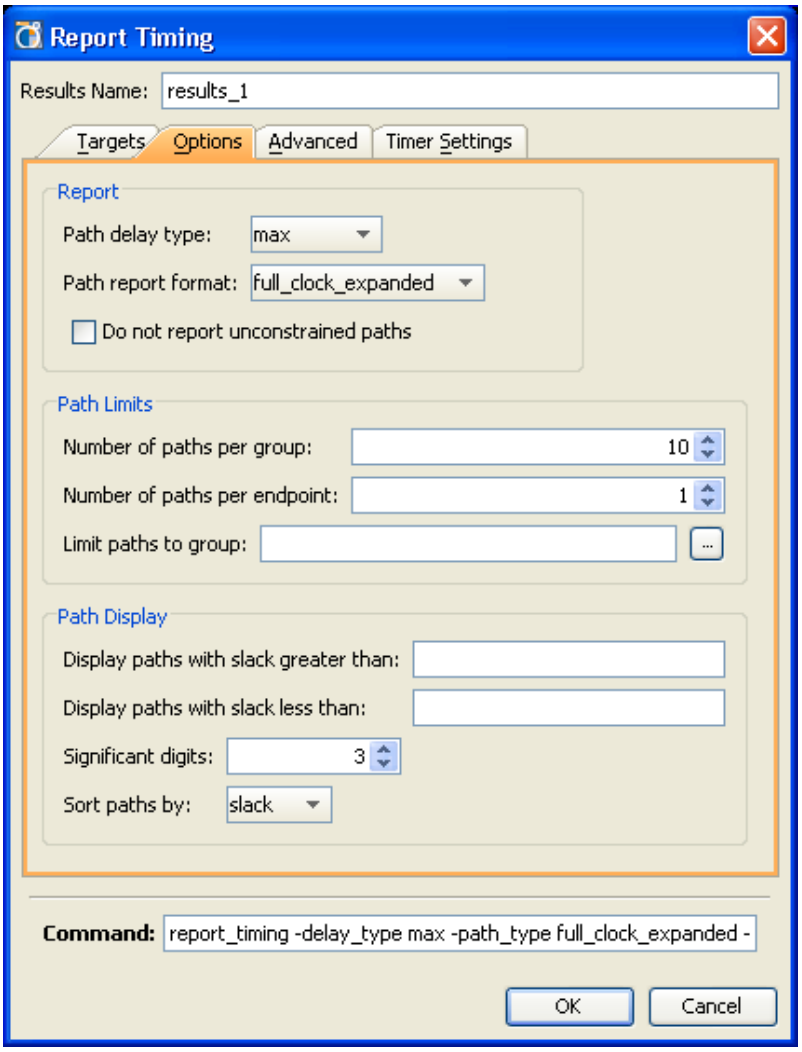


図 20 : [Report Timing] ダイアログ ボックスの [Options] タブ

5-4-1.　[Number of paths per group] フィールドに 30 と入力します。

5-4-2.　**[Advanced]** タブと **[Timer Settings]** タブもクリックして確認します。ここでは何も変更しません。

5-4-3.　**[OK]** をクリックして解析を実行します。

[Timing Results] ビューが開きます (図 21)。

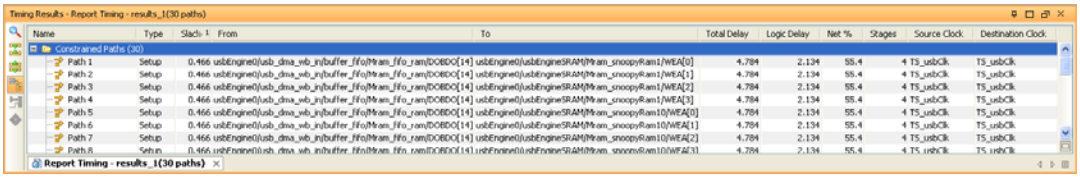


図 21 : タイミング結果の解析

レポートには、パスがリストされます。パス タイプ、スラック、ソース、デスティネーション、遅延合計、ロジック遅延、ネット遅延の割合 (%)、ロジックのステージ、開始クロック、デスティネーション クロックなどが表示されます。エラーのあったパスは赤色で表示されます。

5-4-4. [Timing Results] ビューを最大化します。

5-4-5. パスのリストをスクロールダウンします。

メモ : usbEngine モジュールの 1 つに From と To の値があります。

5-4-6. [To] 列のヘッダを 2 回クリックし、ソース別にリストを分類します。これで、[To] 列の値が逆順序で分類されます。

メモ : 表で表示される場合は、すべてこの方法で分類できます。

- 列ヘッダをもう 1 度クリックすと、反対の順序で分類されます。
- 2 つ目の分類をするには、Ctrl キーを押しつつ、別の列ヘッダをクリックします。

5-4-7. [Timing Results] ビューを元の大きさに戻します。

5-4-8. リストの一番上の **Path 1** をクリックします。

5-5.　[Path Proprties] ビューを最大化します (図 22)。

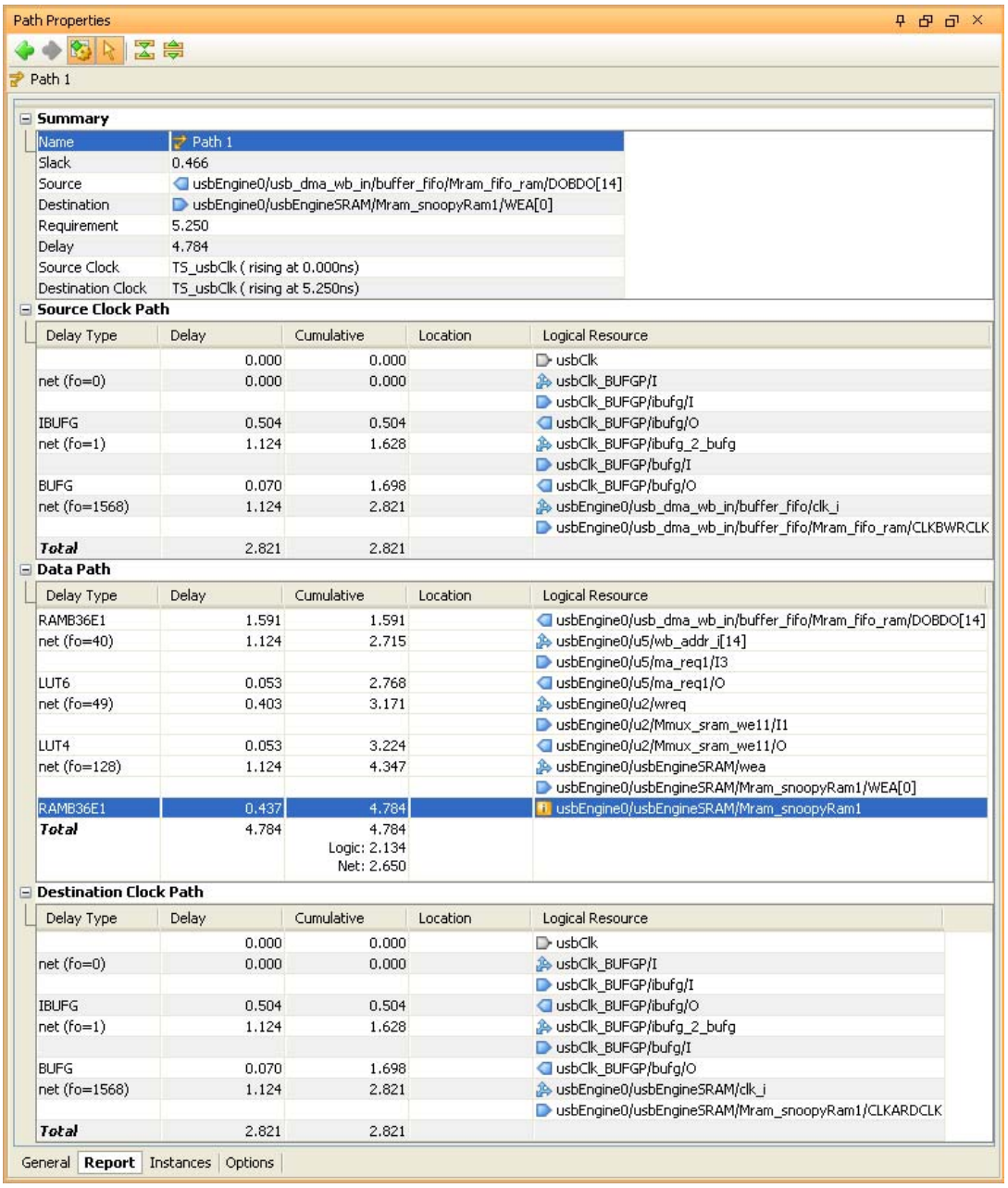


図 22 : パス プロパティの確認

ソース クロック、ディスティネーション クロック、クリティカル ロジックのパスが表示されます。パスに含まれるロジック オブジェクトすべてとそれに関連する遅延が [Path Properties] レポートに表示されます。レポートのリンクをクリックすると、そのロジック オブジェクトが別のビューで選択されます。

5-5-1.　[Path Properties] ビューで [Restore] ボタン ( ) をクリックし、表示を元の位置に戻します。

回路図でのロジックの表示　　手順 6

[Schematic] ビューで表示や削除などのさまざまな回路図ロジックのコマンドを試してみます。たとえば、回路図からロジックを選択して削除します。

6-1. 回路図でタイミング パスを表示します。

6-1-1. [Timing Results] ビューで **Path 1** を選択します。

6-1-2. 右クリックし、**[Schematic]** を選択します。

6-1-3. [Unselect All] ツールバー ボタン ( ) をクリックするか、**F12** キーを押します。

[Schematic] ビューにパス ロジックが表示されます (図 23)。

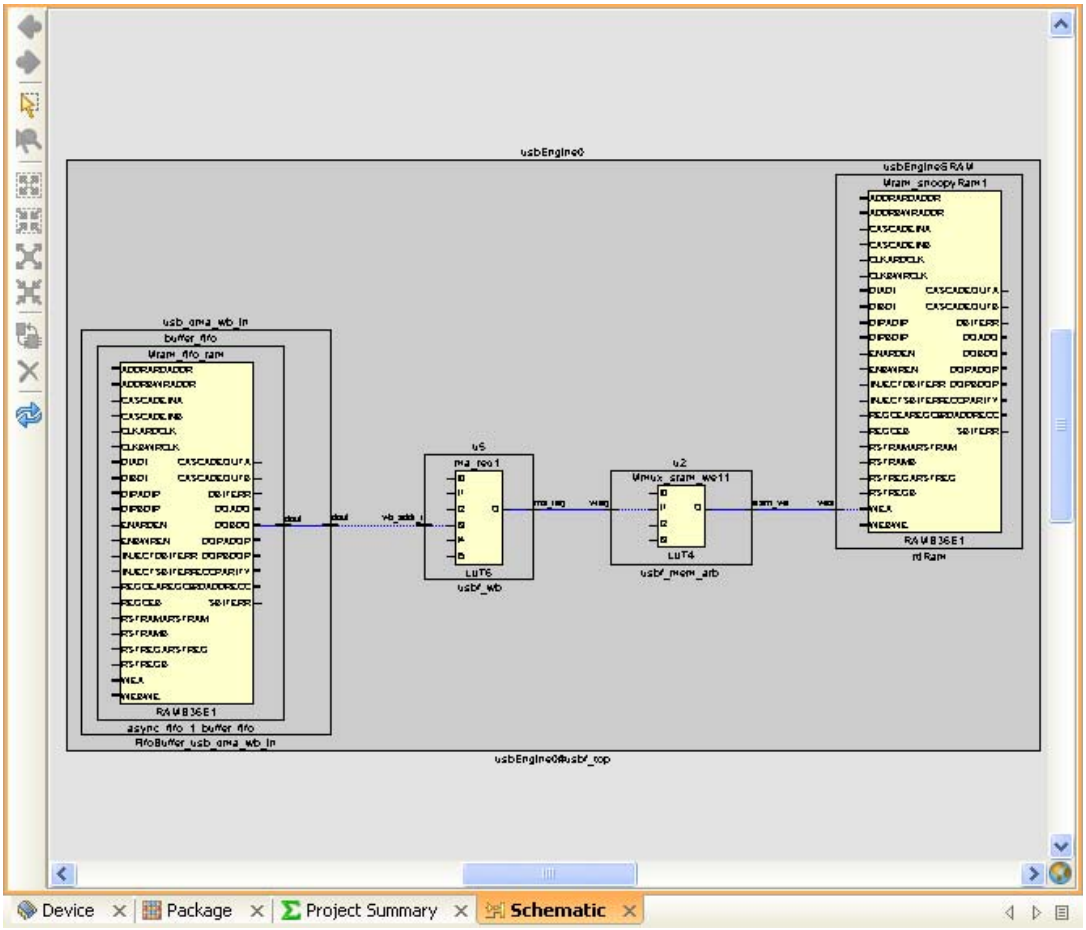


図 23 : [Schematic] でのタイミング パス ロジックの表示

さまざまなレベルの階層が長方形で示されます。これにより、クリティカル パス ロジックに関連するロジック モジュールが判別しやすくなります。

6-1-4. [Schematic] タブの X ボタンをクリックして、回路図を閉じます。

6-1-5. Shift キーを押して [Timing Results] ビューの [From] の値が **usbEngine0/…** で始まるパスをすべて選択します。

6-1-6. 右クリックし、**[Schematic]** を選択します。

選択したパスそれぞれに関連するロジックがすべて [Schematic] ビューで選択されます(図 24)。

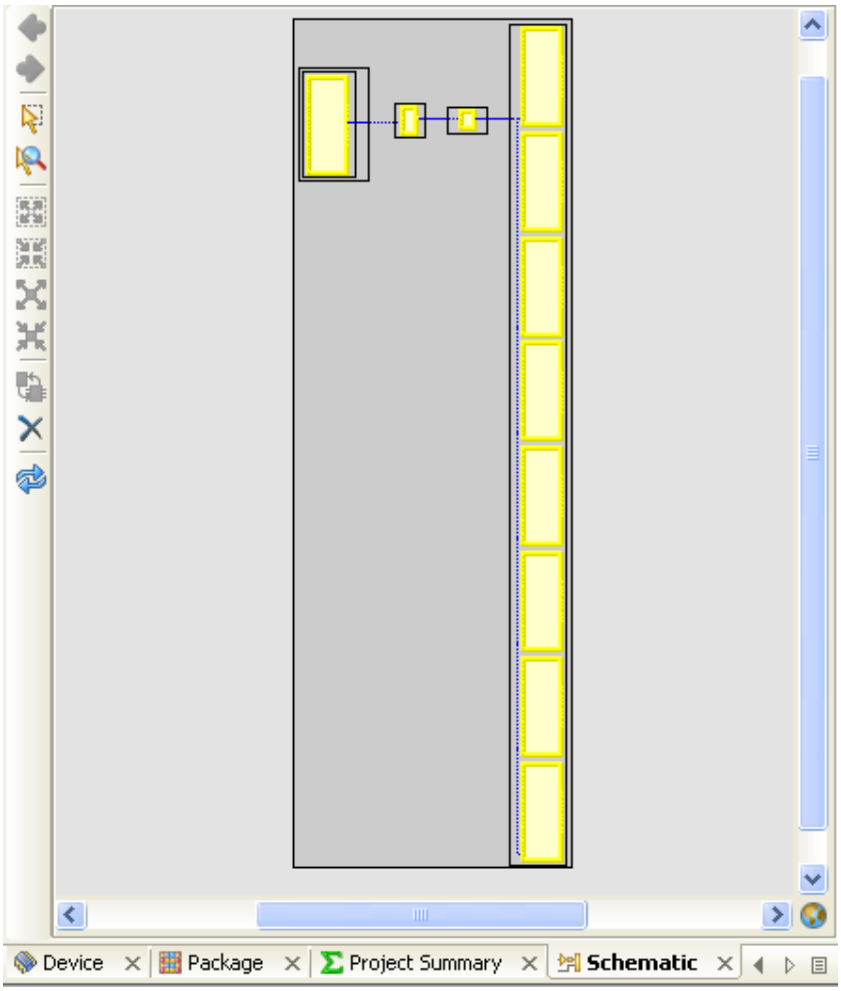


図 24 : 回路図での複数のタイミング パスの表示

6-1-7. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

6-1-8. 上記のパスが選択された状態で [Schematic] ビューで右クリックし、**[Select Primitive Parents]** をクリックします。

そのパス ロジックを含むモジュール 4 つが [Netlist] ビューでハイライトされます。

6-2. 右クリックし、[Show Hierarchy] をクリックします。

6-2-1. usbEngine0 の左側の [Zoom Area] を使用してクリティカルな階層を表示します。

ヒント：[Zoom Area] をマウス動作で実行にするには、ドライバの左上端のすぐ外側にカーソルを移動し、クリックしてドライバの右下端にドラッグしてから、マウスを放します。必要なだけこの動作を繰り返し、[Hierarchy] ビューでハイライトされたパス ロジックを含む 4 つのモジュールを確認します (図 25)。

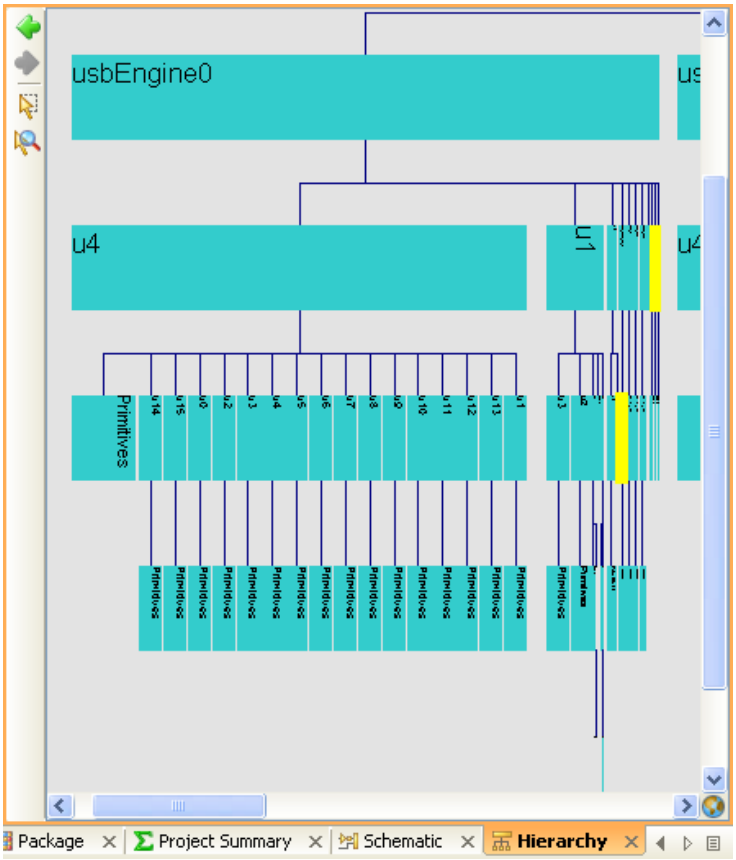


図 25 : [Hierarchy] ビューでの選択したロジックの表示

6-2-2. X ボタンをクリックして [Hierarchy] ビューを閉じます。

6-2-3. マウス動作で [Zoom Area] を実行し、[Schematic] ビューの左にあるハイライトされたモジュール内の 1 つのインスタンスを拡大表示します (図 26)。

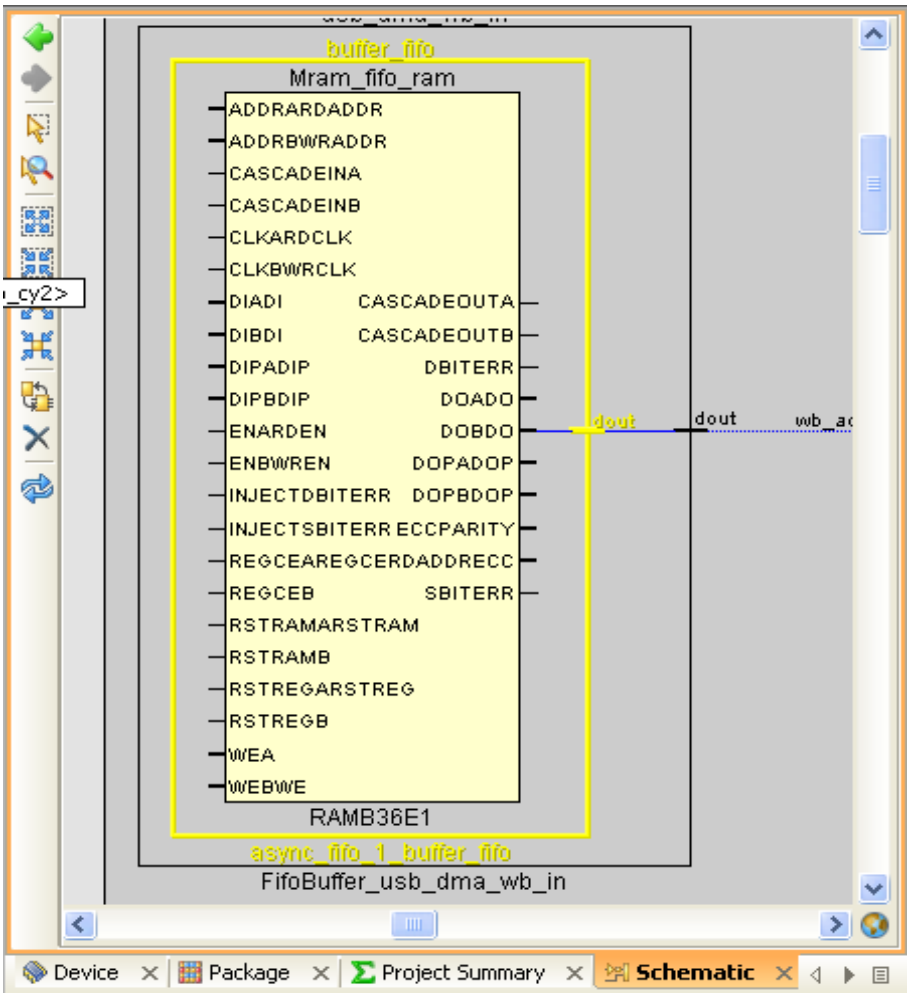


図 26 : クリティカル パス ソース

6-2-4. **RAMB36E1** を選択します。

6-2-5. [Collapse Outside] ツールバー ボタン ( ) をクリックします。

6-2-6. [Unselect All] ツールバー ボタン ( ) をクリックするか、F12 キーを押します。

6-2-7. **ADDRARDADDR** ピンをダブルクリックすると、それに接続されるロジックが展開表示されます (図 27)。

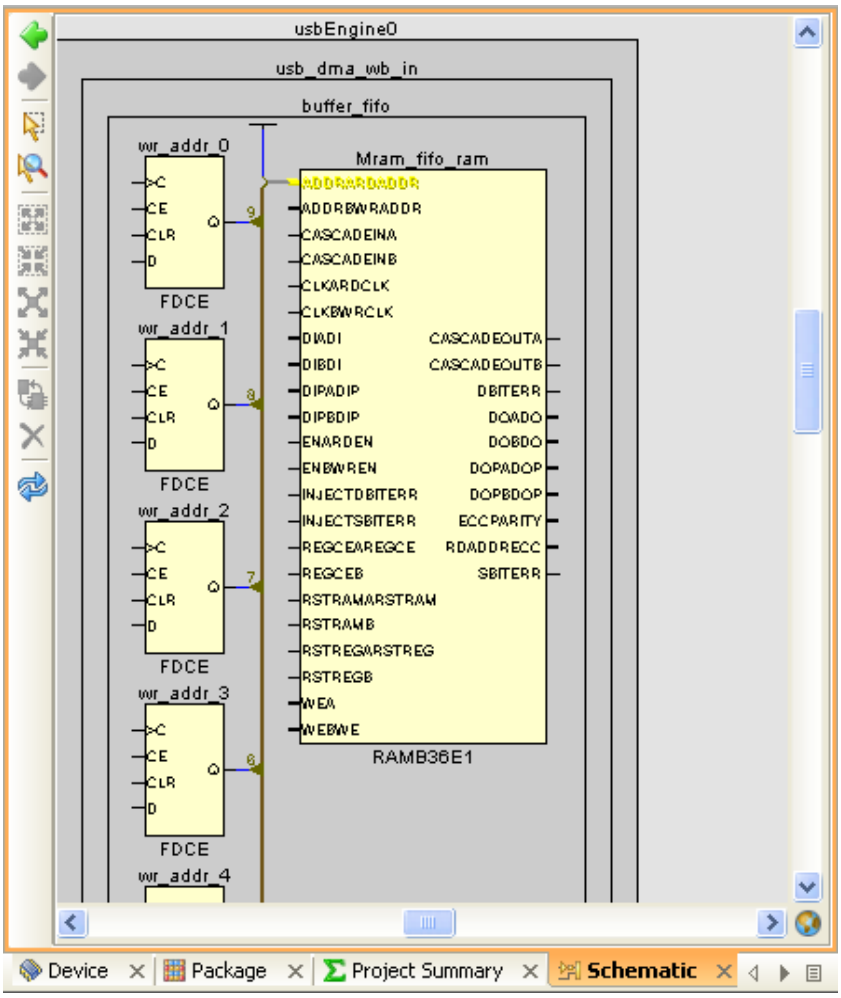


図 27 : [Schematic] ビューでのロジックの展開表示

6-2-8. wr_addr_0 をダブルクリックすると、接続されるロジックが回路図に追加されます。

6-2-9. [Schematic] ビューで [Previous schematic] ボタン ( ) をクリックします。

[Previous schematic] および [Next schematic] ツールバー ボタンをクリックすると、[Schematic] ビューでさまざまな表示を確認できます。このボタンにより、さまざまな回路図レベルを切り替えて表示できます。

メモ : **[Edit]** → **[Undo]** コマンドは、回路図ロジックをインタラクティブに展開表示している間は使用できません。

6-2-10. RAMB36E1 インスタンスを含む小型の buffer_fifo モジュールの階層ボックス (長方形) をクリックします。

6-2-11. [Schematic] ビューで [Expand Inside] ボタン ( ) をクリックします。

6-2-12. [Zoom Fit] をクリックします (図 28)。

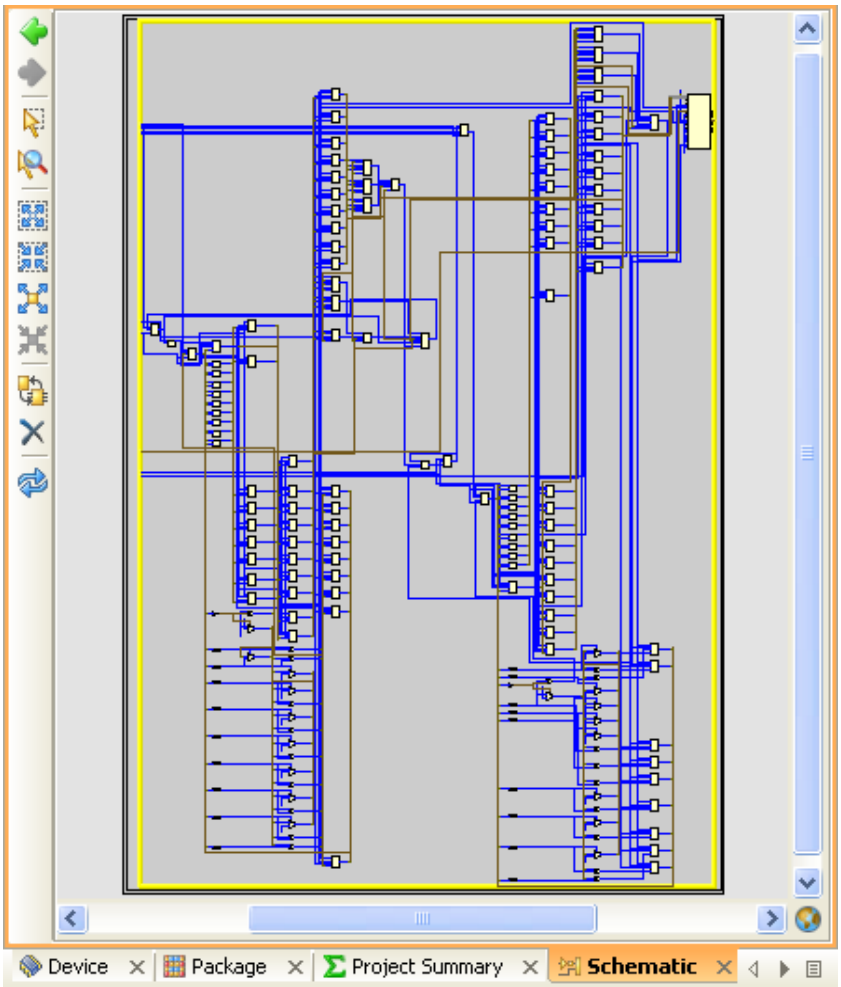


図 28 : 回路図モジュール内のロジックの展開表示

6-2-13. [Select Area] ツールバー ボタン ( ) をクリックし、[Schematic] ビューでロジックのいくつかを含めるように、長方形を描画します。

6-2-14. [Remove selected elements] ボタン ( ) をクリックします。選択したロジックが表示されなくなります。

6-2-15. Schematic の X ボタンをクリックして [Schematic] ビューを閉じます。

6-2-16. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

メモ : このチュートリアルでは一部の [Schematic] ビュー コマンドとロジック削除のオプションのみを説明しています。

デザイン制約の確認　　手順 7

7-1. TS_cpuClk 制約のプロパティでターゲットの 13.75 ns を 12.75 ns に変更します。

7-1-1. [Netlist] ビューの隣の **[Timing Constraints]** ビューをクリックします (図 29)。

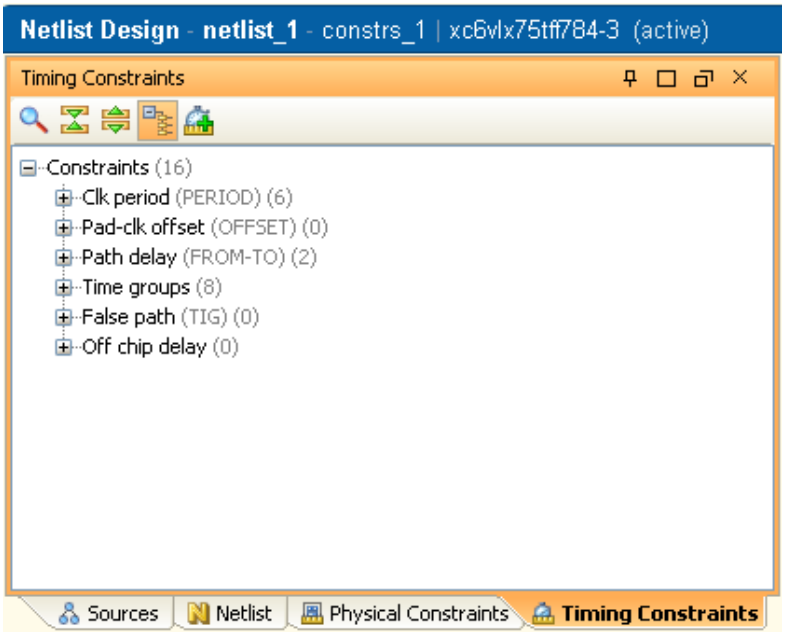


図 29 : [Timing Constraints] ビュー

7-1-2. [Expand All] ツールバー ボタン ( ) をクリックし、タイミング制約を展開表示します。

7-1-3. TIMESPEC TS_cpuClk 制約を選択します (図 30)。[Constraints Properties] ビューが表示されます。

メモ : TIMESPEC は、Clk period と Timespec period の下位に含まれます。デフォルトでは、タイプ別に制約が表示されます。

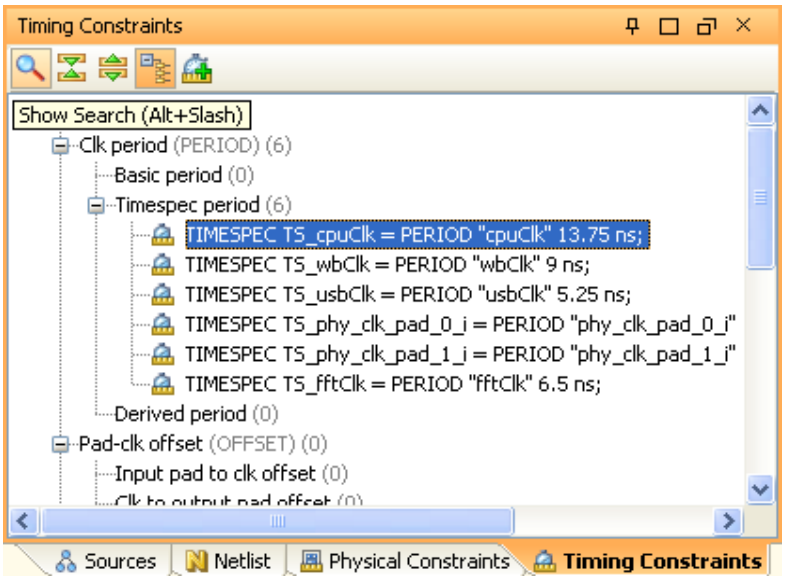


図 30 : タイミング制約の展開表示

7-1-4. [Period] フィールドの値を **12.75** に変更します (図 31)。

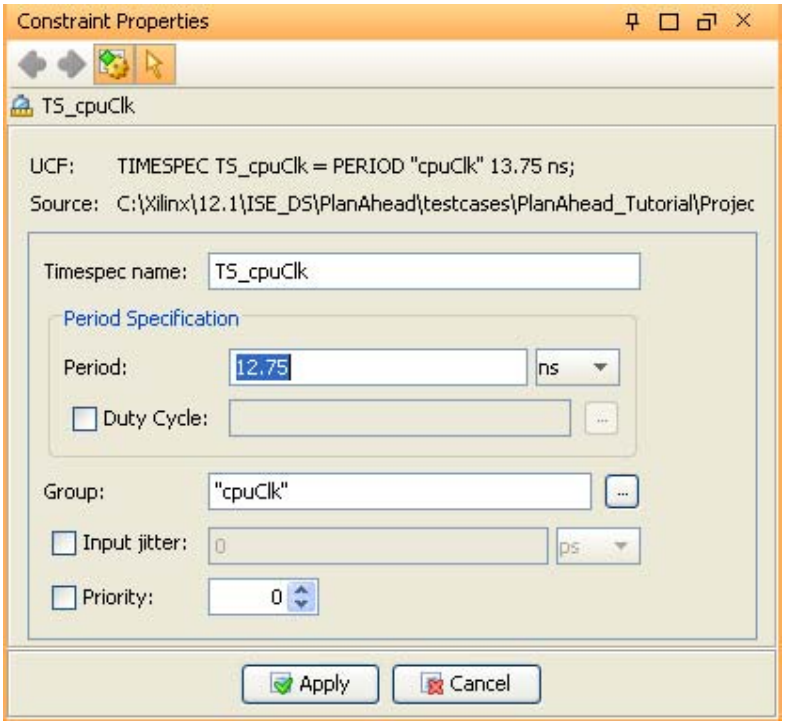
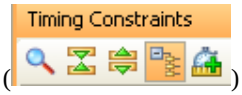


図 31 : TS_cpuClk を変更します。

7-1-5. **[Cancel]** ボタンをクリックし、変更をキャンセルします。デザインは、より速い制約で実行される必要はありません。

7-1-6. [Timing Constraints] ビューで [Group by type] ボタン () をクリックしてオフにし、カテゴリ別ではなく、デザインのタイミング制約をすべてリストします (図 32)。

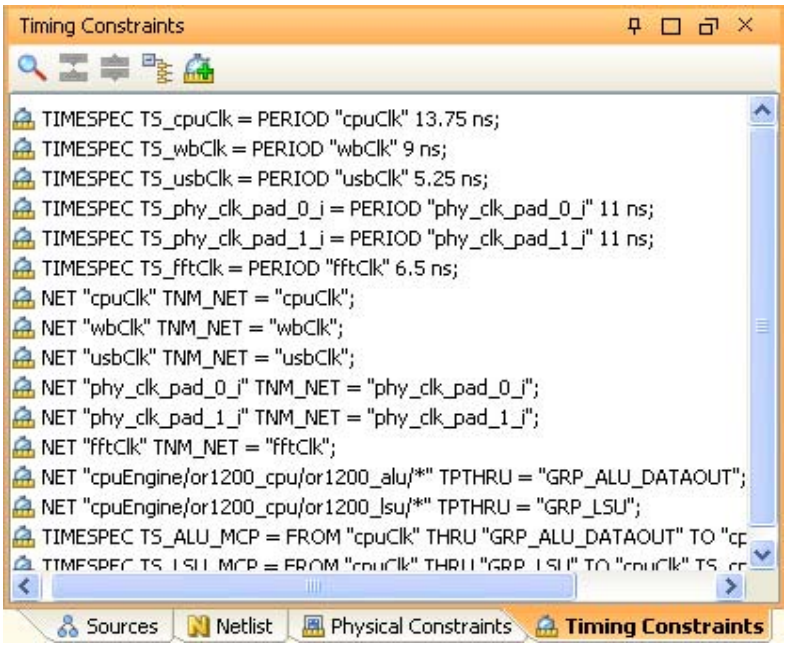


図 32 : 定義済みタイミング制約すべてのリスト

メモ : デザインからタイミング制約を削除するには、その制約を右クリックし、**[Delete]** を選択します。

7-2. 新しいタイミング制約を作成します。

7-2-1. [New Timing Constraint] ボタン ( ) をクリックし、[New Timing Constraint] ダイアログ ボックスを開きます (図 33)。

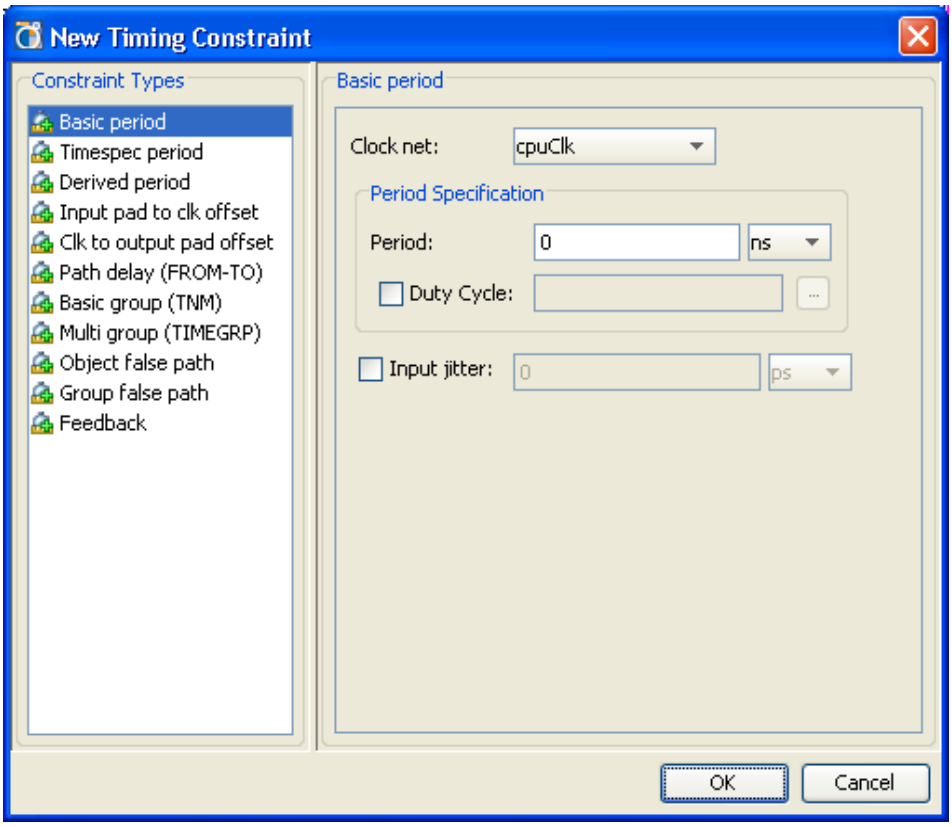


図 33 : [New Timing Constraint] ダイアログ ボックス

7-2-2. [Constraint Types] フィールドを確認してください。

7-2-3. **[Cancel]** をクリックし、[New Timing Constraint] ダイアログ ボックスを閉じます。

7-2-4. [Netlist Design] ビューを X ボタンをクリックして閉じます。メッセージが表示されたら、保存しないで閉じます。

これでツールが Project Manager に表示されるようになります。

デザインのインプリメンテーション　手順 8

8-1. インプリメンテーション ツールの動作を確認します。

8-1-1. [Sources] ビューが表示されていない場合は、**[Window]** → **[Sources]** をクリックして表示させます。

8-1-2. [Sources] ビューで [Expand All] ( ) をクリックしてすべてのソース ファイルを表示します (図 34)。

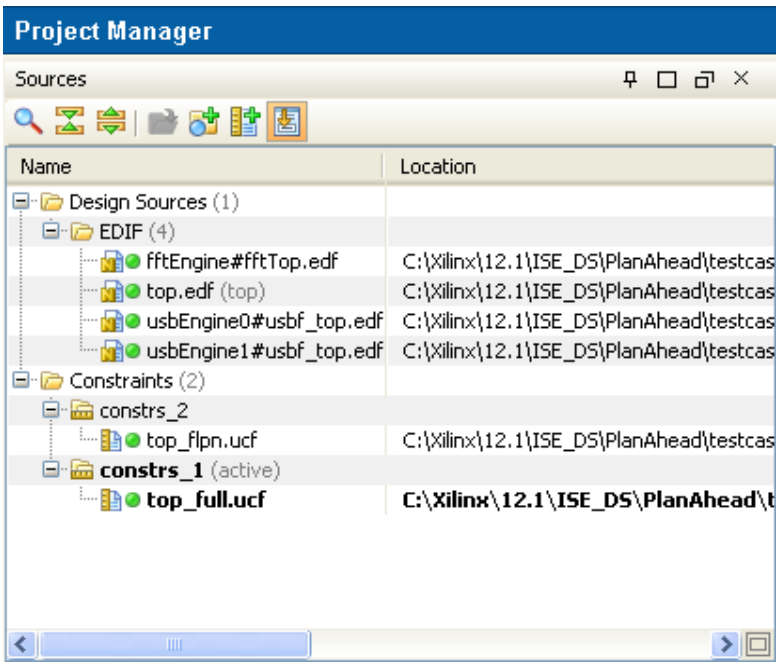


図 34 : [Sources] ビューの展開表示

インプリメンテーション ツールを実行する場合、PlanAhead はデフォルトで選択されている制約ファイルを使用します。top_full.ucf を使用する consts_1 が選択されているかどうか確認してください。

8-1-3. Flow Navigator で **[Implement]** ボタン (図 35) をクリックしてインプリメンテーションを開始します。

図 35 : インプリメンテーションの開始ボタン

NGDBuild、MAP、PAR、TRACE、および XDLISE といった ISE の配置配線ツールが起動されます。この処理は高速マシンで約 20 分かかります。

インプリメンテーション実行中は、ツールに触らないようにしてください。右上のステータス バーに進捗状況が表示されます (図 36)。

図 36 : インプリメンテーションの進捗状況を示すバー

インプリメンテーションが終了したら、[Implementation Completed] ダイアログ ボックスが表示されます (図 37)。

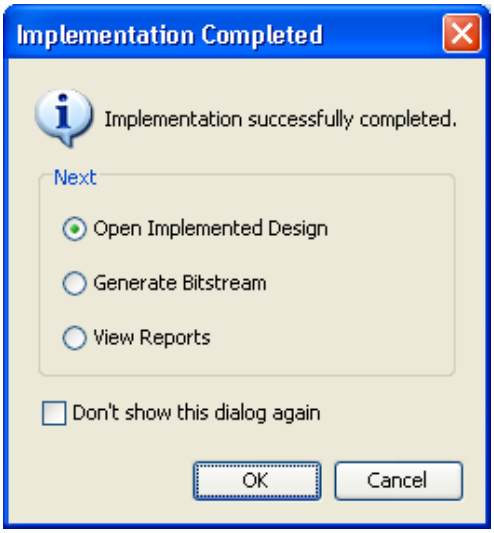


図 37 : [Implementation Completed] ダイアログ ボックス

8-1-4. [**Open Implemented Desig**] をオンにします。

8-1-5. **[OK]** をクリックします。

配置配線ファイルが読み込まれ、各インスタンスがどこに配置されたか、どの TRCE がタイミングについてレポートしているのかが表示されます (図 38)。

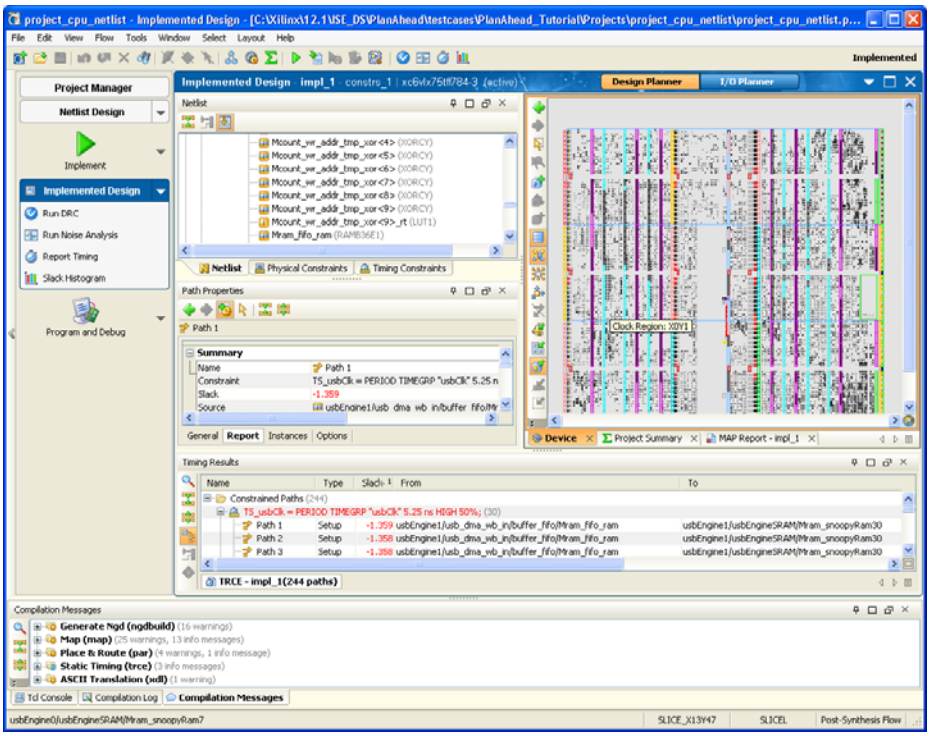


図 38 : インプリメント済みのデザイン

8-1-6. **[Compilation Messages]** のタイトル バーをダブルクリックし、ビューを最大化します。

8-1-7. [Place and Route (par)] の隣の + マークをクリックし、この段階で生成されているメッセージを確認します (図 39)。

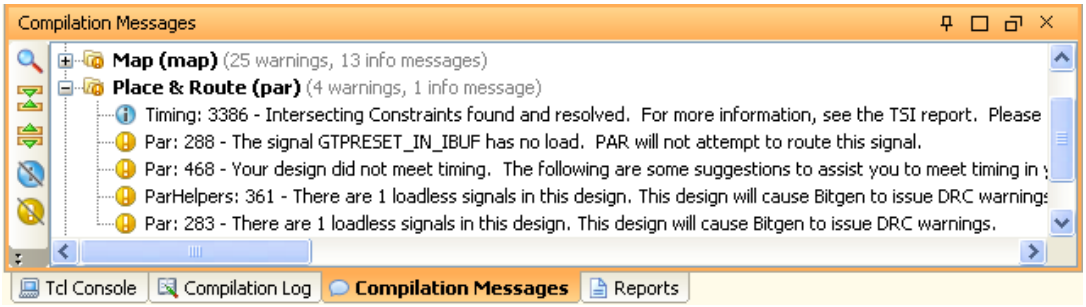


図 39 : [Compilation Messages] タブ

メモ : [Compilation Messages] には、インプリメンテーション ツールで生成されたすべての情報、警告、エラー メッセージが表示されます。青い円に斜線の入ったアイコンをクリックすると情報メッセージ、黄色い円に斜線の入ったアイコンをクリックすると警告メッセージが非表示にできます。

8-1-8. **[Compilation Messages]** のタイトル バーをダブルクリックし、ビューを元に戻します。

タイミング結果の解析　　手順 9

インプリメンテーションからのタイミング結果を解析すると、フロアプランに役立ちます。インポートされた TRCE レポート データを含む [Timing Results] ビューでパスを分類し、選択できます。

9-1. インプリメンテーション タイミング結果を確認します。

9-1-1. **[Implemented Design – impl_1]** の一番上の青いバーをダブルクリックし、ビューを最大化します。

9-1-2. [Timing Results] ビューで **Path 1** を選択します (図 40**Error! No bookmark name given.**)。

9-1-3. 右クリックで **[Mark]** をクリックします。

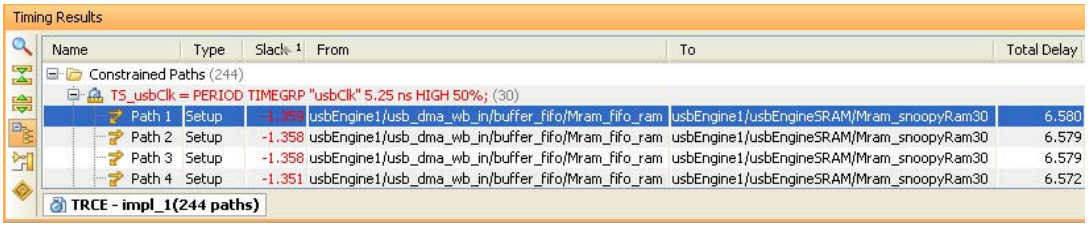


図 40 : 最もクリティカルなタイミング パスの選択

インポートされたタイミング結果では、タイミング パスが制約別に分類されています。[Timing Results] ビューのパスを選択すると、[Path Properties] ビューにそのタイミング パスの詳細が表示されます。

配置情報がインポートされているので、そのパスは [Device] ビューでもハイライトされます。これにより、どのフロアプラン手順を使用してタイミングを達成するのがわかりやすくなります。

[Mark] コマンドを実行すると、赤と緑の星マークでそれぞれタイミング パスのスタート ポイントとエンドポイントが表示されます。

図 41 : タイミング パスの配置

9-1-4. [Select] → [Unmark Mark] をクリックしてマークを削除します。

9-1-5. **Shift** を押しながら TS_usbClk のエラーのあったパスをすべて選択します。ほかのタイミング制約からのパスは選択しないでください。

9-1-6. 右クリックで **[Schematic]** をクリックします。

[Schematic] ビューに選択したパス上にあるインスタンスがすべて表示されます (図 42)。

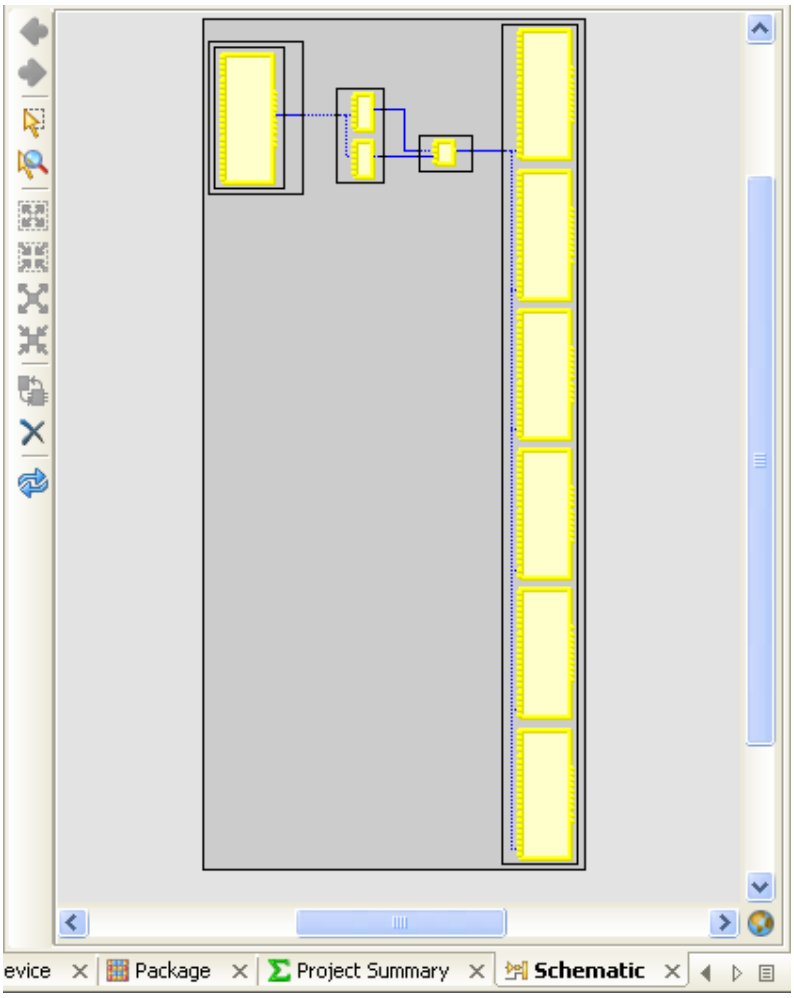


図 42 : 回路図での問題のあったタイミング パスの確認

9-1-7. [Netlist] ビューで [Collapse All] ボタンをクリックします。

9-1-8. [Schematic] ビューで右クリックして **[Select Primitive Parents]** をクリックし、選択したパス上にあるインスタンスすべてを含む一番小さい親モジュールを選択します (図 43)。

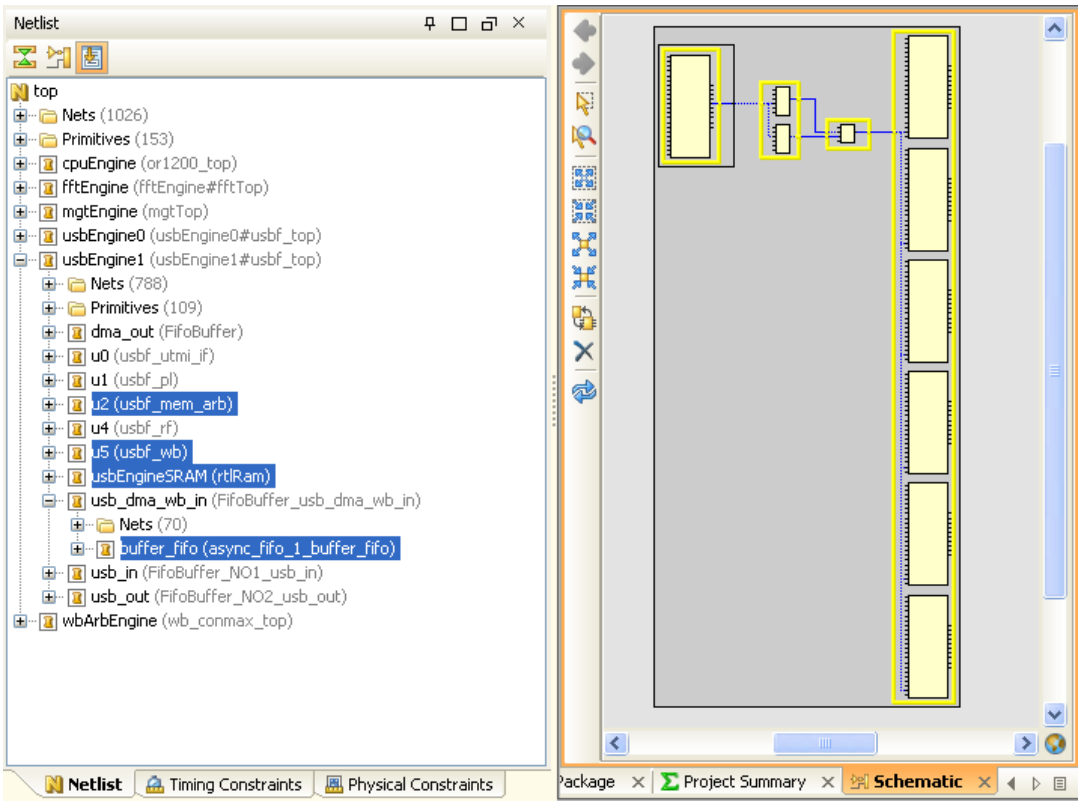


図 43 : パス オブジェクトの親モジュールの選択

[Netlist] ビューで対応するロジック モジュールが選択されます。

9-2. [Device] ビューを表示します。

9-2-1. [Device] ビューで [Hide All Timing Paths] ボタン ( ) をクリックします。

9-2-2. [Schematic] ビューを X ボタン Schematic をクリックして閉じます。

モジュール レベルの配置のハイライト　　手順 10

フロアプラン ストラテジは、前のインプリメンテーション結果を検証することで決定できます。フロアプランなしでロジックがどのようにインプリメントされたかを理解することで、モジュールの配置を解析し、Pblock の位置を決定できます。

10-1. クリティカル タイミング パス モジュールをハイライトして、配置を見やすくします。

10-1-1. [Netlist view] ビューで右クリックをし、**[Highlight Primitives]** → **[Cycle colors]** を選択します。

10-1-2. [Device] タブをクリックし、ハイライトを表示します。

各モジュールのプリミティブは別の色で表示されます (図 44)。

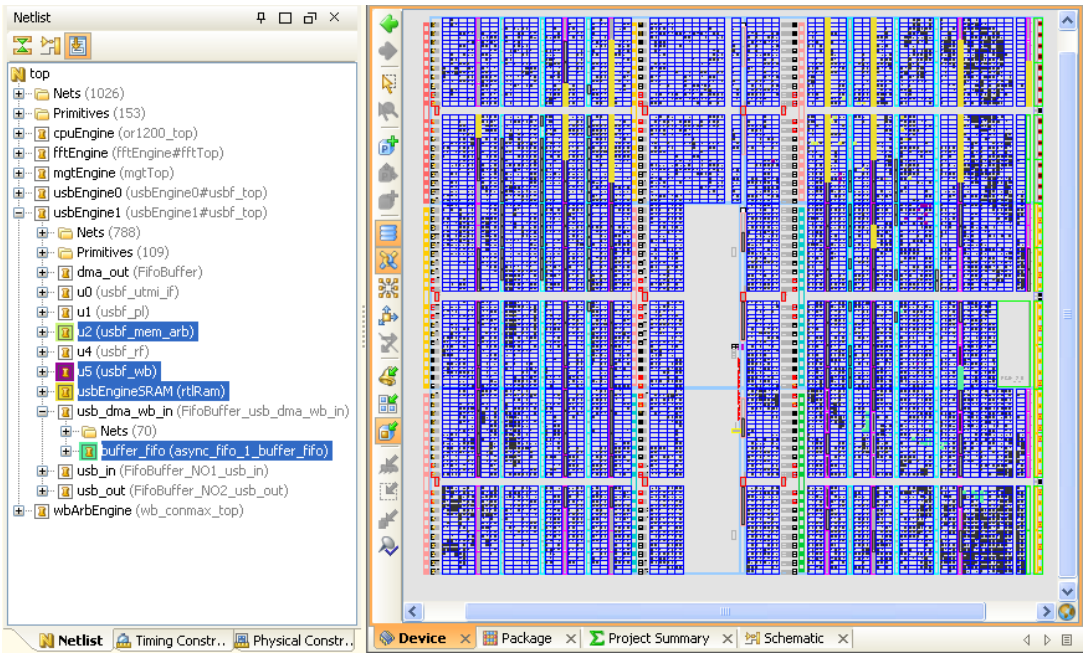


図 44 : モジュールの配置のハイライト

プリミティブが広い範囲に広がっているのがわかります。スクロールしたり、拡大レベルを変更したりしてください。多くのインスタンスがブロック RAM です。これらは、フロアプランによりタイミングが改善される可能性があります。

10-1-3. [Netlist] ビューで [Collapse All] ボタン ( ) をクリックします。

10-1-4. usbEngine1 モジュールを選択します。

10-1-5. 右クリックで [**Highlight Primitives**] → [**Color 1**] を選択するか、目立つ色を使用します。

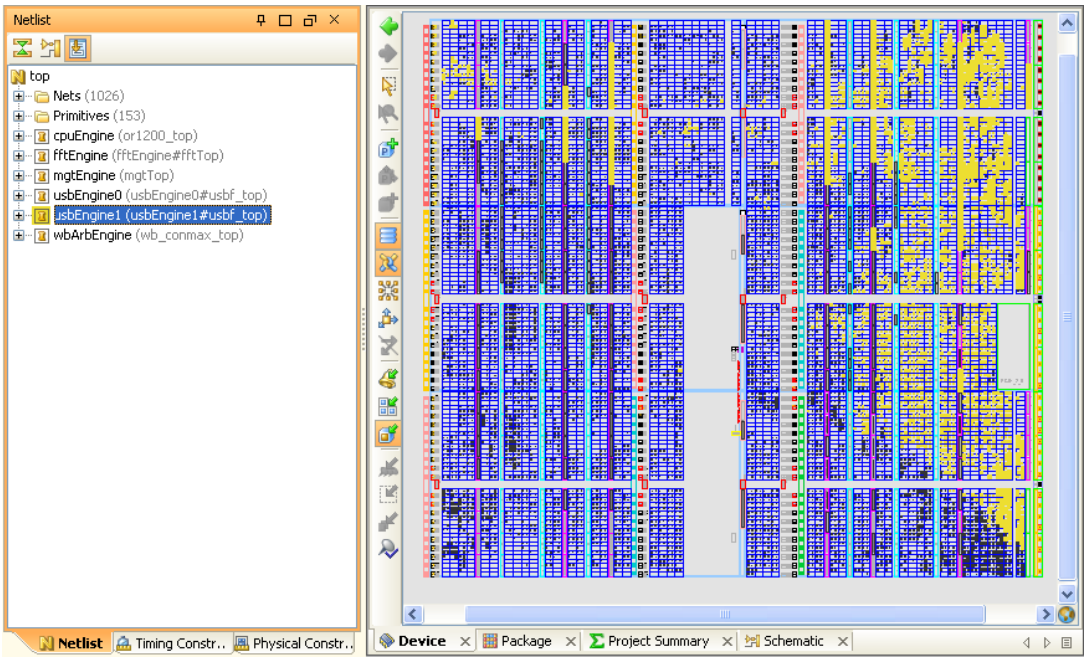


図 45 : USB ネットリスト モジュールのハイライト

モジュールの配置が広い範囲に広がっていることがわかります (図 45)。スライス ロジックはブロック RAM の配置に近くありません。 usbEngine1 および usbEngine0 はフロアプランするのに良い例です。フロアプランについては、次の手順で説明します。

10-1-6. メイン ツールバーで [Unhighlight All] ボタン ( ) をクリックします。

10-1-7. [Device] ビュー ツールバーの [Hide All Timing Paths] ボタン ( ) をクリックします。

接続の確認　　手順 11

PlanAhead には、ロジックの拡張、選択、ハイライトなどの機能があります。これらの機能を使用し、モジュールをフロアプランしやすいようにします。ロジックがデバイス中に広がっている場合は、まとめておかないとフロアプランしにくくなります。

配線の混線やタイミング競合は、ロジックがクリティカル エリアに移動されないように、クリティカル ロジック エリアの外にロジックをフロアプランすると回避できます。

11-1. I/O 接続を表示します。

11-1-1. [Device] ビューで [Hide/Show IO Nets] ボタン ( ) をクリックします。

緑のラインは、配置済みのロジックから I/O ピンへの接続を示します (図 46)。

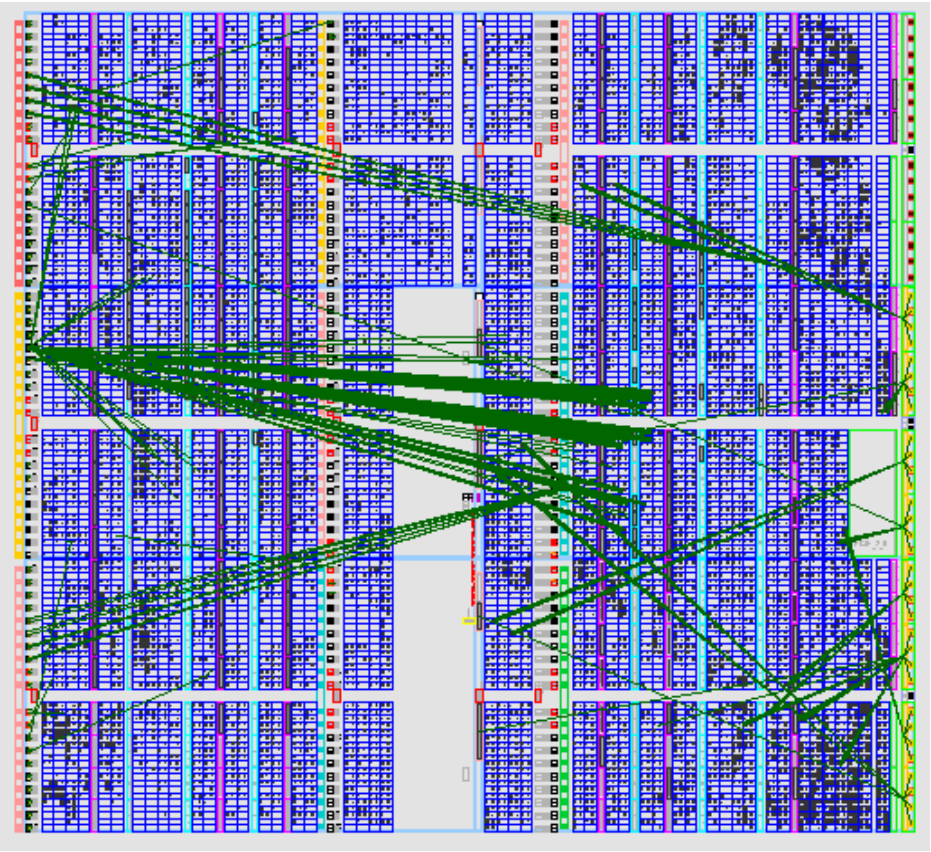


図 46 : I/O 接続を表示した [Device] ビュー

チップの左下の I/O ラインがチップの右側に交差しているのが表示されます。

11-1-2. バンク 14 (チップの左下の I/O バンク) の一番上の部分に接続される I/O ネットの 1 つをクリックします。

11-1-3. [Netlist] ビューを確認します。

このネットは、usbEngine0/dma_out/buffer_fifo/DataOut_pad_o バスの一部です。

11-1-4. [Netlist] ビューで [Collapse All] ボタン ( ) をクリックします。

11-1-5. [Netlist] ビューで usbEngine0 および usbEngine1 を選択します。

11-1-6. 右クリックで [Show Connectivity] を選択します。

usbEngine0 および usbEngine1 モジュールから残りのデザインへ接続されたインターフェイス ネットが黄色で表示されます (図 47)。

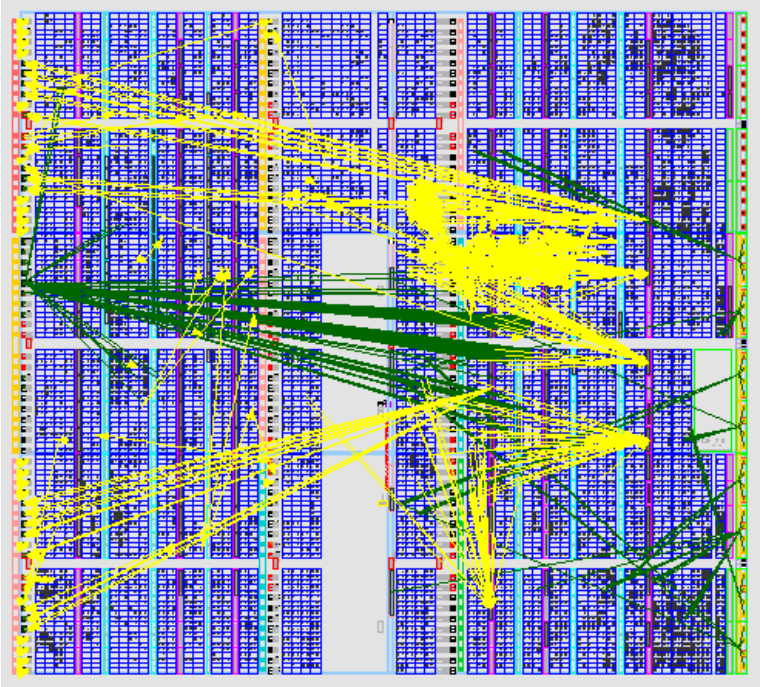


図 47 : [Show Connectivity] コマンドの使用

11-1-7. 右クリックで **[Show Connectivity]** をもう 1 度選択し、インターフェイス ネットから接続されるロジック オブジェクトをすべて表示します。

11-1-8. 右クリックで **[Show Connectivity]** をもう 1 度選択し、これらの選択したロジック オブジェクトから広がったネットすべてをハイライトします。

[Show Connectivity] コマンドは、ソース ネットまたはロジック オブジェクトからのロジックの広がりをハイライトまたは選択するのに使用できます。

11-1-9. [Unselect All] ツールバー ボタン ( ) をクリックするか、**F12** キーを押します。

配置制約の使用　手順 12

PlanAhead では、配置したオブジェクトを処理できます。プリミティブ タイプ別に配置制約を表示または削除できます。

12-1. 配置されたインスタンスを表示します。

12-1-1. [Edit] → [Find] をクリックします。

12-1-2. [Primitive] フィールドを **[Block RA]** に変更します (図 48)。

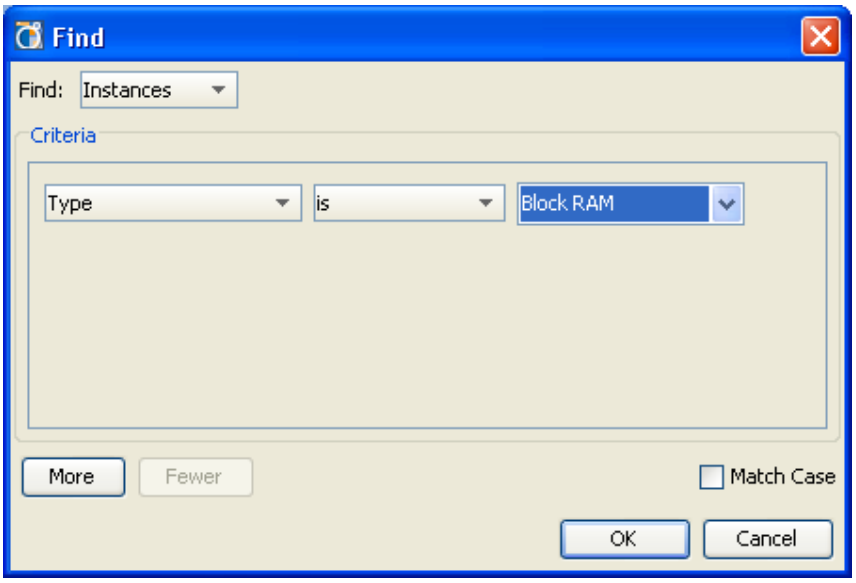


図 48 : [Find] ダイアログ ボックス

12-1-3. **[OK]** をクリックします。

12-1-4. [Find Results] ビューで [Block Ram] を選択します。

12-1-5. **Shift** キーを使うか、**Ctrl + A** を使用してブロック RAM すべてを選択します。

[Device] ビューですべてのブロック RAM が選択されます (図 49)。

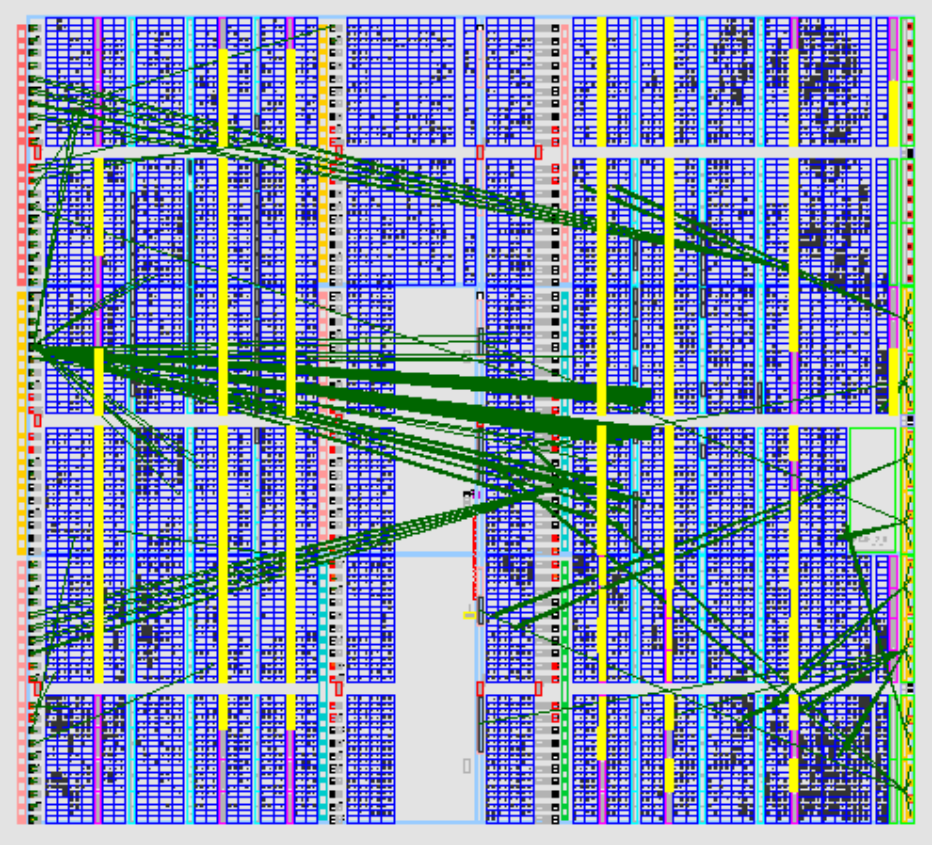


図 49 : ブロック RAM が選択された [Device] ビュー

12-1-6. ブロック RAM はデバイス全体に広がっているので、この配置を維持する理由はありません。

12-2. [Clear Placement Constraints] コマンドを使用して配置制約をすべて削除します。

[Clear Placement Constraints] ダイアログ ボックスでは、配置制約を選択して削除できます。I/O およびクロック関連のリソースは、通常変更しないので、ファブリック ロジックから分離します。これにより、ISE ソフトウェアからインポートされた配置制約が素早く削除できます。ロジックのタイプ別フィルタもあるので、タイプ別に配置制約を選択して削除することもできます。

12-2-1. [**Tools**] → [**Clear Placement**] をクリックします。

12-2-2. [Clear Placement Constraints] ダイアログ ボックスで [**Instance placement**] をオンにします (図 50)。

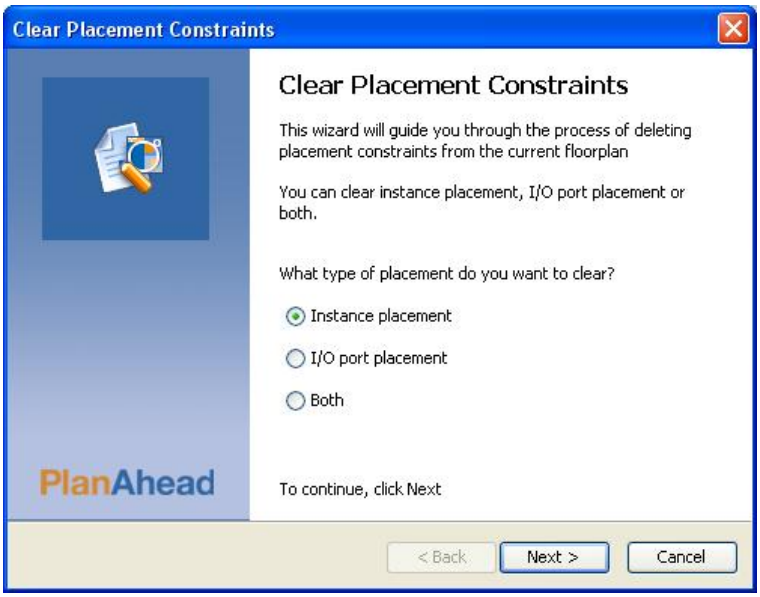


図 50：配置制約の削除

12-2-3. **[Next]** をクリックし、[Unplace Instances] ページに進みます。

12-2-4. [**Unplace All instances**] をオンにします。

12-2-5. **[Next]** をクリックし、[Instance Types to Unplace] ページに進みます。

12-2-6. **[Default]** をクリックしてデフォルト選択を使用します。

コンフィギュレーション可能なフィルタにより、ロジックの LOC 制約をタイプ別に削除するか維持するかが選択できます。

12-3. [Next] をクリックし、[Fixed Placement] ページに進みます。

12-3-1. [**Keep 17 Fixed Instances**] をオンにします。

12-3-2. **[Next]** をクリックし、[Clear Placement Summary] ページに進みます。

12-3-3. サマリを確認したら、**[Finish]** をクリックします。

12-3-4. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

階層接続の表示　　手順 13

ネットリストのモジュール間の接続を確認すると役立つことがあります。このセクションでは、さまざまなツールを使用してデザイン接続を表示します。ネットリスト階層は Pblock という物理階層に分割され、チップに配置されます。Pblock は AREA_GROUP 制約として制約ファイルに保存されます。

13-1. [Auto-create Pblocks] コマンドを使用してデザインの最上位レベルを分割します。

13-1-1. [**Tools**] → [**Auto-create Pblocks**] をクリックします。

このダイアログ ボックスでは、作成する Pblock の最大数を定義したり、最小の Pblock サイズを指定したりできます。

作成するように指定した合計数より多くのモジュールがある場合は、最大のモジュールを使用して Pblock が作成されます。

13-1-2. 6 つのモジュールが選択された状態のまま **[OK]** をクリックします。

[Physical Hierarchy] ビューで最上位レベルの Pblock を確認します (図 51)。[Netlist] ビューでは、6 つのモジュールの隣のアイコンが から へ変更されています。 これは、そのインスタンスが Plock に配置されたことを示します。

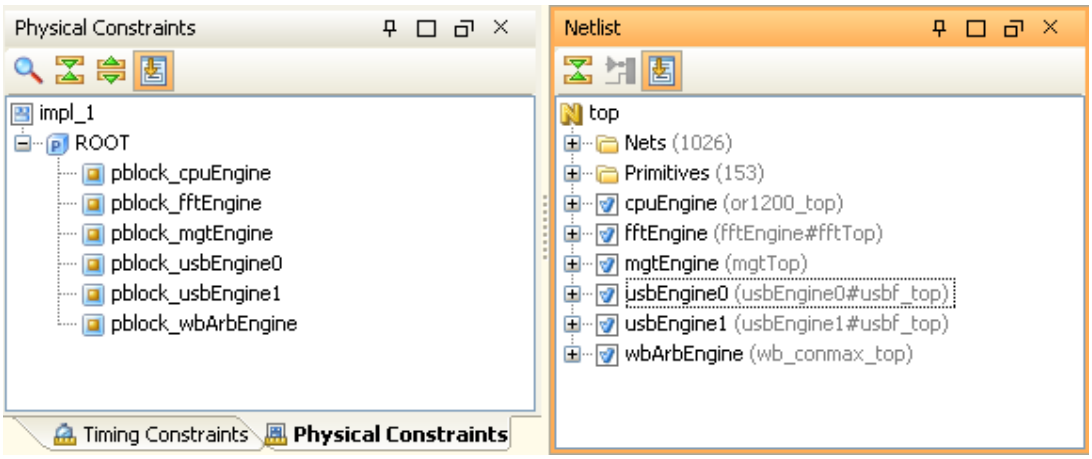


図 51 : 最上位レベルのパーティション

13-2. [Place Pblocks] コマンドを使用して新規に作成した Pblock を配置します。

13-2-1. [**Tools**] → [**Place Pblocks**] をクリックします。

[Place Pblocks] ダイアログ ボックスでは、配置する Pblock を選択したり、Pblock のターゲット SLICE の使用率サイズを変更したりできます。

メモ : [Place Pblocks] コマンドは、選択した Pblock を素早く作成するために使用します。Pblock のサイズは、SLICE ロジックに基づいてのみ決定されます。その他 SLICE 以外の範囲は、考慮されません。このため、[Place Pblocks] コマンドを使用して作成された Pblock は修正しないと、インプリメンテーションで問題が発生する可能性があります。

13-2-2. **[OK]** をクリックし、Pblock を配置します (図 52)。

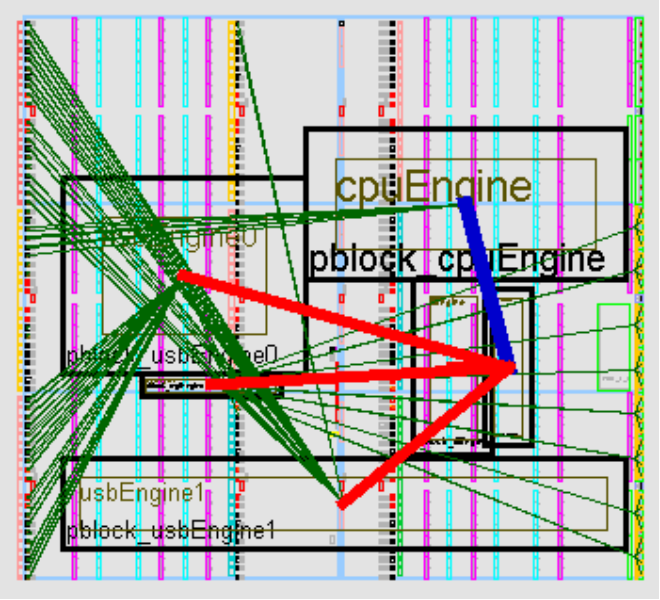


図 52 : 最上位レベルの Pblock の配置

[Place Pblocks] コマンドによる Pblock の配置は、実行するたびに異なることがあるので、図の配置とは異なっている可能性があります。

13-2-3. 接続が表示されていない場合は、[Device] ビューで [Show/Hide I/O Nets] ツールバー ボタン ( ) と [Show/Hide Bundle Nets] ツールバー ボタン ( ) がオンになっていることを確認してください。

ヒント :このボタンを使用した表示の切り替え方法は、チュートリアル全体で使用できます。

バンドル ネットや I/O の接続線を表示すると、デザインの接続がわかりやすくなり、Pblock を移動して絡まった接続をほどくことができます。これにより、デザイン全体のデータ フローや潜在的な配線の混雑状態が早期にわかるようになります。

13-3. 接続とリソースに基づいて Pblock の配置およびサイズを調整してください。

13-3-1. 必要であれば、Pblock の配置を調整して、接続の絡まりをほどきます (図 53)。

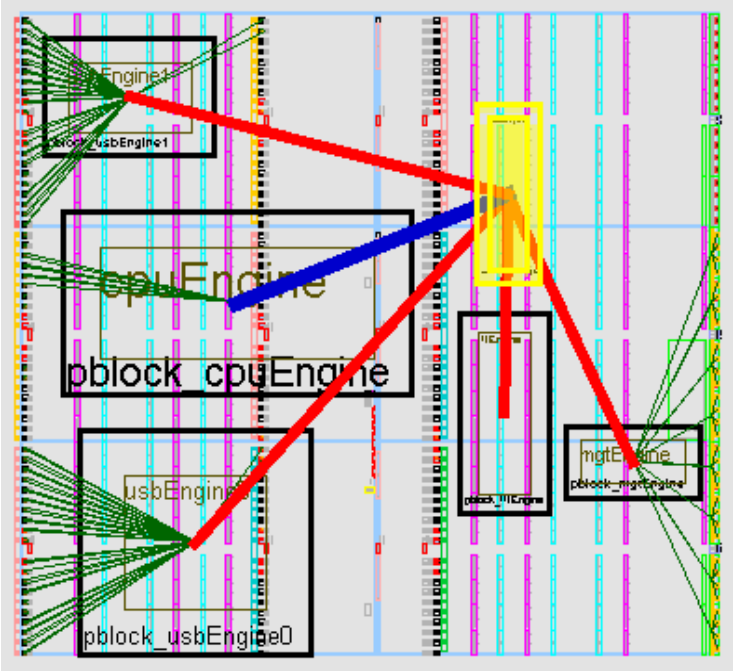


図 53 : Pblock の形を変更して階層接続を表示

ヒント : [Device] ビューで [Set Pblock Size] ツールバー ボタン ( ) をクリックし、デバイスのいずれかの場所に長方形を再描画します。長方形を移動したり、再描画して、ロジック接続がはっきり表示されるようにします。この段階では、リソース要件を満たすように Pblock の大きさを変更する必要はありません。

13-3-2. [Set Pblock] ダイアログ ボックスで Pblock 長方形内のグリッド タイプをすべて使用するかどうか尋ねられたら、**[OK]** をクリックします。

usbEngine1 および usbEngine0 への接続を確認してください。これらの 2 つのブロックはタイミングに問題があります。I/O 接続を確認し、この 2 つのブロックをどこに配置するか検討します。

13-3-3. Pblock を選択し、[Pblock Properties] ビューで **[Statistics]** タブをクリックします (図 54)。

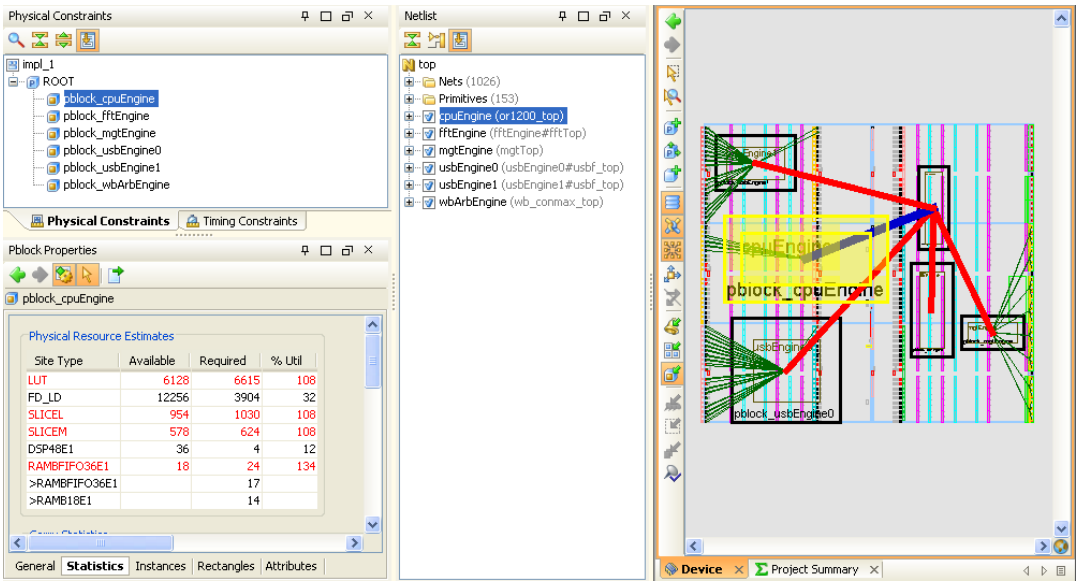


図 54 : リソース概算に基づいた Pblock のサイズ変更

Pblock の [Physical Resources Estimates] フィールドを確認します。Pblock のサイズを変更する際はこれらのリソースを使用し、Pblock の長方形内に適切なリソースが存在するかどうか確認し、割り当てられるロジックと矛盾のないようにします。

13-3-4. [Pblock Property Statistics] のリストをスクロール ダウンします。

フロアプランを作成する場合、I/O ネット、バンドル ネット、インスタンスなどのオブジェクトを選択できないようにすると、Pblock を選択しやすくなります。

13-4. 表示オプションを設定し、オブジェクトが選択できないようにします。

13-4-1. **[Tools]** → **[Options]** をクリックします。

13-4-2. [PlanAhead Options] ダイアログ ボックスで左側の **[Themes]** をクリックし、カラー テーマ オプションを表示します。

13-4-3. ダイアログ ボックスで **[Device]** タブをクリックします。

13-4-4. [Object Type] 列の [Assigned Instance] と [IO Net] の [Select] ボックスをオフにします (図 55)。

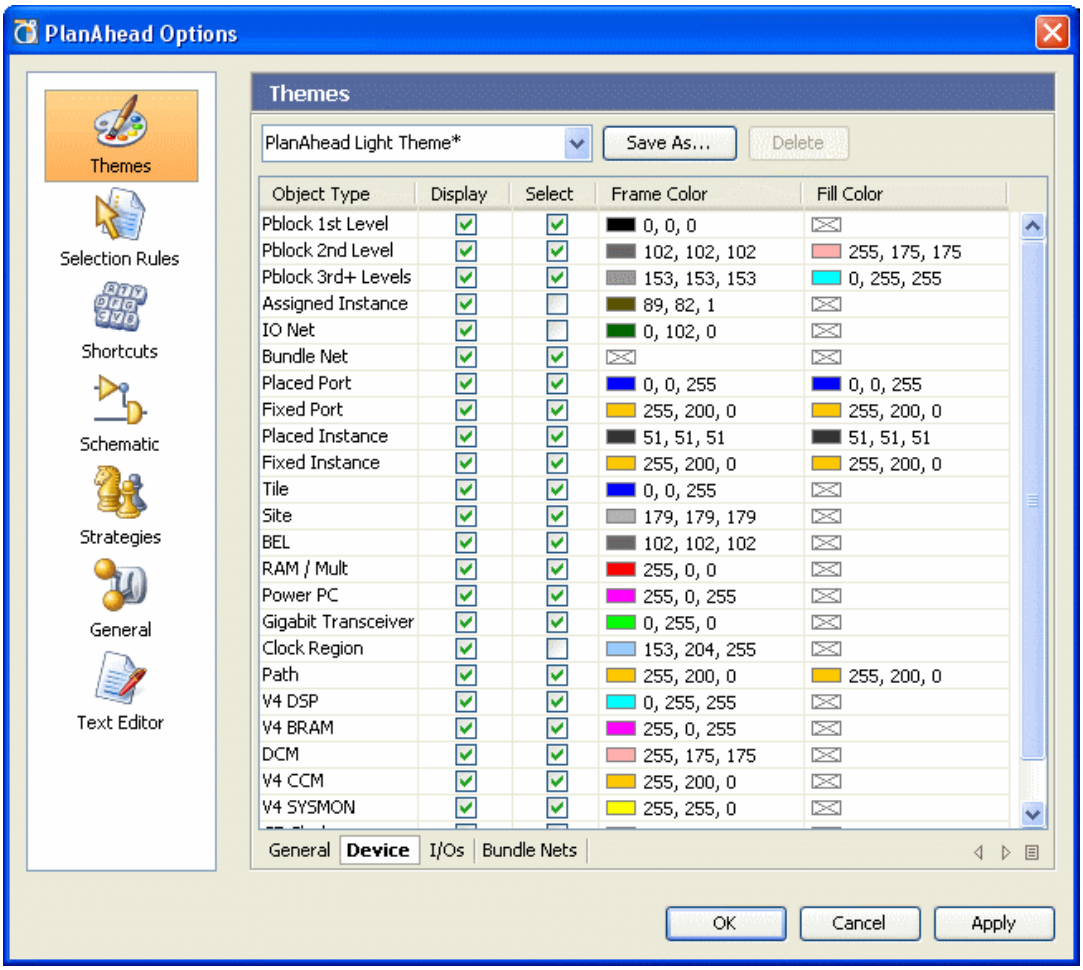


図 55 : 表示オプションの設定

13-4-5. **[Save As]** ボタンをクリックし、何か名前を入力して **[OK]** をクリックします。

このようにカスタム ビュー オプションを保存すると、今後同じ設定を使用できるようになります。

13-4-6. [PlanAhead Dark Theme] または [PlanAhead Light Theme] にテーマを戻します。

13-4-7. **[OK]** をクリックします。

PlanAhead では、Pblock とバンドル ネットの接続を表示できます。バンドル ネットの色と線の幅がその束に含まれる信号の数を示します。色、ネット カウント、ライン幅などの設定は、[PlanAhead Options] ダイアログ ボックスの [Bundle Nets] タブから設定できます。

13-5. [Bundle Net Properties] ビューを確認します。

13-5-1. [Device] ビューで色のついたバンドル ネットの 1 つを選択します。

13-5-2. [Bundle Net Properties] ビューが表示されます。

13-5-3. **[Properties]** タブをクリックすると、2 つのモジュール間のバンドルに含まれるネットすべてがリストされます (図 56)。

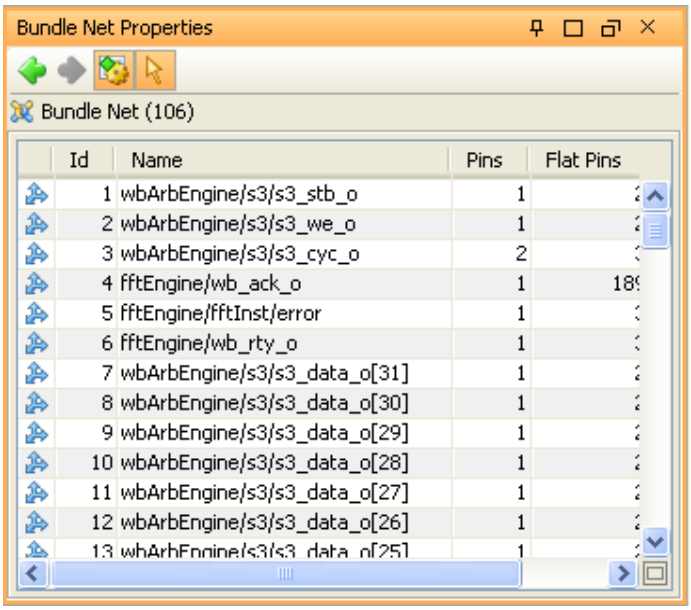


図 56 : バンドル ネット プロパティの表示

検索機能を使用したクロックドメインの表示　　手順 14

フロアプランが効率的かどうかは、異なるクロック領域に同期エレメントが適切に配置されたかどうかによって決まることが多くあります。クロック領域をハイライトして接続を表示すると、クロック領域に対して Pblock を適切に配置できます。

14-1. グローバル クロック ネットを選択し、マークします。

14-1-1. **[Edit]** → **[Find]** をクリックします。

14-1-2. [Find] ダイアログ ボックスで次のオプションを設定します。

- [Find] : **Nets**
- 最初のフィールド : **Type**
- 2 つ目のフィールド : **is**
- 3 つ目のフィールド : **Global Clock**

14-1-3. **[Unique Nets Only]** がオンになっていることを確認し、**[OK]** をクリックします。

14-1-4. [**Pins**] 列ヘッダを 2 回クリックして [Find Results] ビューを並び替えます。

14-1-5. [Find Results] ビューで usbClk_BUFGP を選択します。

2 つの usbEngine Pblock で I/O に接続されたネットがハイライトされます (図 57)。

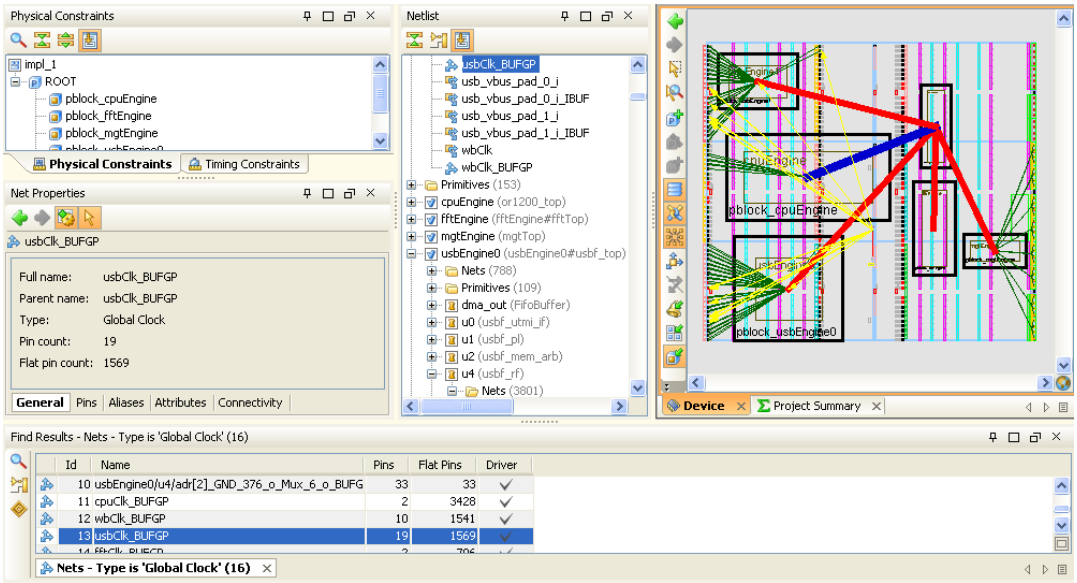


図 57 : クロック ネットのディスティネーションの表示

14-2. 回路図を使用し、グローバル クロック ネットがプリミティブに対してどのように広がっているかを確認します。

14-2-1. グローバル クロック ネットが [Find Results] タブで選択された状態で、右クリックし、**[Schematic]** を選択します。

回路図にレジスタ グループへ接続された usbClk_BUFGP ネットが表示されます。

14-2-2. 回路図の上部の BUFGP (usbClk_BUFGP) を拡大します (図 58)。

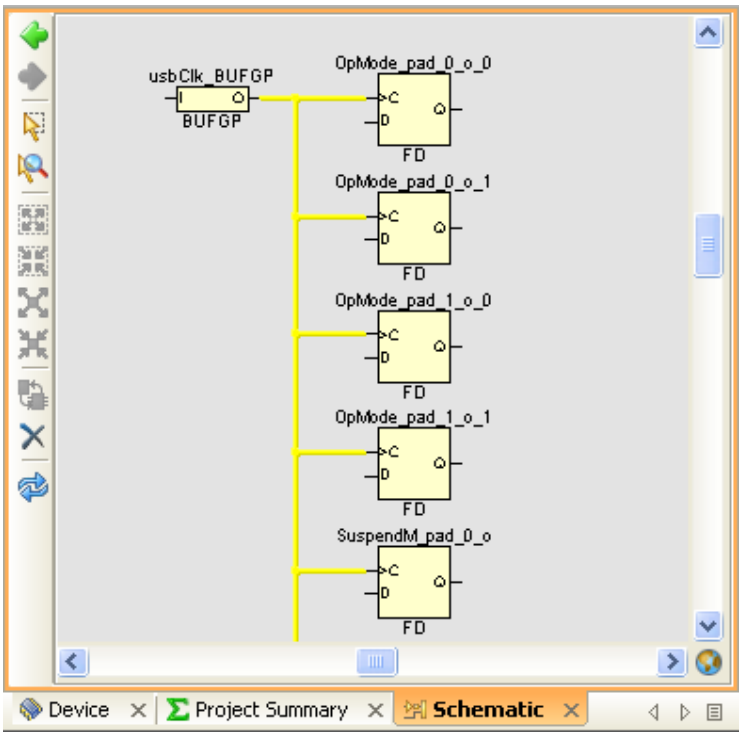


図 58 : 回路図でのクロック信号の表示

14-2-3. [Schematic] ビューを閉じます。

14-2-4. [Physical Constraints] ビューで **Shift** キーを押しながら 6 つの Pblock すべてを選択します。

14-2-5. [Delete] ツールバー ボタン ( ) をクリックします。

14-2-6. 確認するメッセージが表示されたら **[OK]** をクリックして削除します。

タイミング クリティカルな階層のフロアプラン　手順 15

タイミング クリティカルな階層をフロアプランすると、タイミング パフォーマンスが改善できます。前の手順では、usbEngine1 および usbEngine0 にタイミング クリティカルなロジックが含まれることがわかりました。usbEngine1 および usbEngine0 は、左端に移動できます。接続を表示するために多くの Pblock が作成されています。すべての Pblock をインプリメンテーションまで実行すると、タイミングに悪影響がでますので、タイミング クリティカルな階層のみフロアプランします。

15-1. タイミング クリティカルな階層の Pblock を配置します。

15-1-1. **[Device]** タブをクリックします。

15-1-2. [Device] ビューで必要であれば ボタンを使用して I/O ネットを表示します。

15-1-3. [Netlist] ビューで [Collapse All] ツールバー ボタン ( ) をクリックします。

15-1-4. [Netlist] ビューで usbEngine0 を選択します。

15-1-5. [Device] ビューのツールバーで [Draw Pblock] ボタン ( ) をクリックします。

カーソルが十字に変わります。

15-1-6. [Device] ビューの左下の端に長方形を描画します。

15-1-7. [Pblock Properties] を確認します。

15-1-8. [**Statistics**] タブをクリックします。

インプリメンテーションで問題がないように、Pblock のサイズを変更します。

15-1-9. ブロック RAM の使用率が 100%、スライス使用率が 60% になるように Pblock のサイズを変更します。真ん中のグレー部分の列にかからないように Pblock を配置します (図 59)。

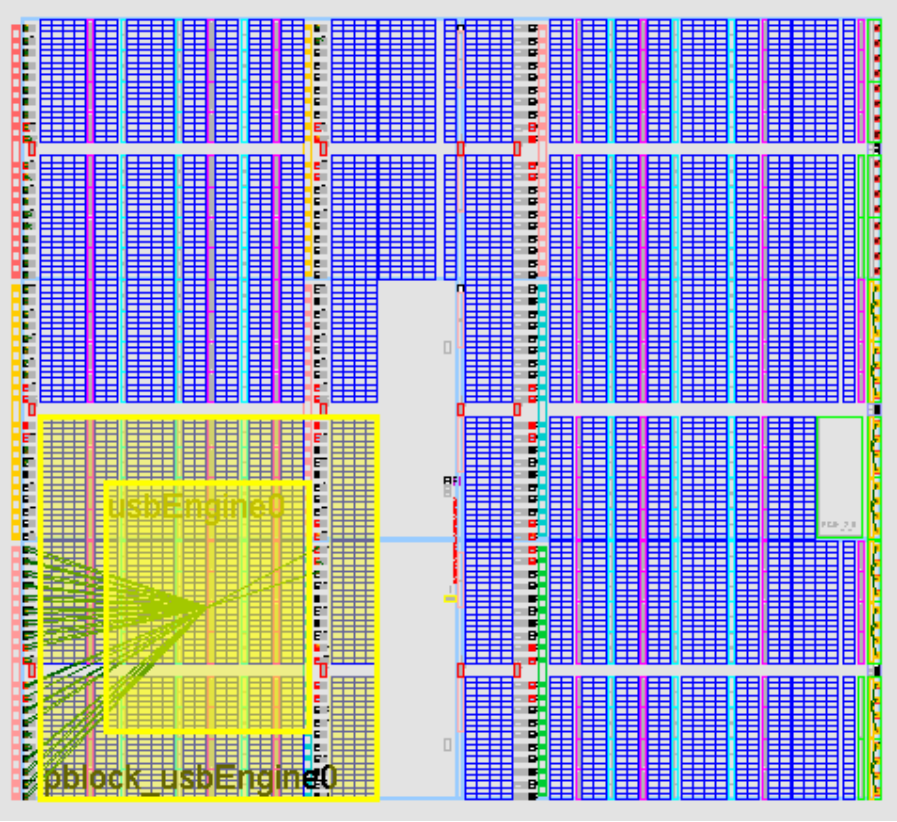


図 59 : usbEngine0 の配置

15-1-10. usbEngine1 を選択し、左上に同様の Pblock を作成します。

15-1-11. Pblock 同士が重ならないようにしてください。最終的な配置は、次の図 60 のようになります。

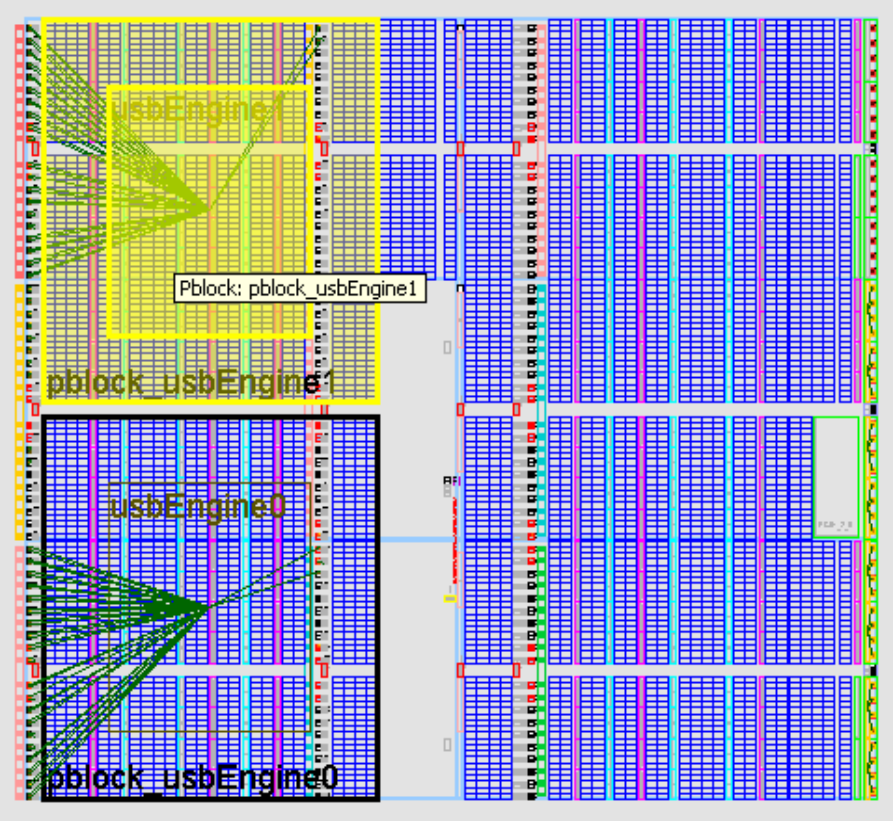


図 60 : タイミング クリティカルなブロック

15-1-12. [Implemented Design] ビューを X ボタンをクリックして閉じます。

15-1-13. **[OK]** をクリックし、**[Yes]** をクリックして保存します。

ツールは Project Manager に戻ります。

15-1-14. ソース ウィンドウで top_full.ucf を開き、この UCF ファイルの一番下までスクロールし、AREA_GROUP の行が新しく追加されていることを確認します。これでインプリメンテーションを実行できるようになりました。

まとめ

このチュートリアルでは、PlanAhead ソフトウェアを使用して、インプリメンテーション ツールを実行する前に合成済みのデザインとターゲット デバイスを確認および解析しました。これにより、潜在的な問題やエラーを、インプリメンテーション中ではなく、デザイン サイクルの早期段階で発見できます。また、デザイン リソースの概算、デザイン ルール違反、タイミング概算、制約および接続などをグラフィカルに表示することで、デザインを理解し、潜在的な問題のエリアを発見しやすくもなっています。

インプリメンテーションを実行し、その結果をインポートしてタイミング結果を検証しました。この後、回路図でクリティカル パスのオブジェクトを解析し、これらのパス オブジェクトの親モジュールを選択しました。モジュール配置をハイライトし、[Show Connectivity] コマンドを使用してモジュールの接続を表示し、ISE ソフトウェアで割り当てられた配置制約を削除しました。デザインを解析し、フロアプランを作成してタイミングを改善しました。