

PlanAhead ソフトウェア チュートリアル

パーシャル リコンフィギュレーション フローの概要

UG 743 (v 12.1) 2010 年 5 月 3 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 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

パーシャル リコンフィギュレーション フローの概要	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
チュートリアルの目標	8
チュートリアル デザインの説明	9
チュートリアルの概要	9
ソフトウェア ツール フロー.....	10
プロジェクト ディレクトリおよび HDL デザインの構造	11
チュートリアルの手順	12
HDL ソースからのネットリストの合成 (オプション) 手順 1	13
プロジェクトを開く 手順 2	14
パーシャル リコンフィギュレーションの作成とリコンフィギャブル モジュールの追加 手順 3	17
リコンフィギャブル モジュールの追加 手順 4	20
ブラック ボックス モジュールの追加 (オプション) 手順 5	22
リコンフィギャブル パーティションのフロアプラン 手順 6	24
パーティション ピンとリコンフィギャブル パーティションのインターフェイスタイミング 手順 7	30
パーシャル リコンフィギュレーションのデザイン ルール チェック 手順 8	32
コンフィギュレーションのインプリメントとプロモート 手順 9	34
追加コンフィギュレーションの作成とインプリメント 手順 10	38
パーシャル リコンフィギュレーション検証の実行 手順 11	43
ビット ファイルの生成とダウンロード 手順 12	44
まとめ	47

PlanAhead ソフトウェア チュートリアル

パーシャル リコンフィギュレーション フローの概要

はじめに

このチュートリアルでは、HDL 合成から BIT ファイルの生成およびダウンロードまでの単純なパーシャル リコンフィギュレーション デザインを作成します。PlanAhead™ ソフトウェアでデザインをインプリメントおよび解析するには、ザイリンクス ソフトウェア ツールが使用されます。パーシャル リコンフィギュレーション デザインには、CORE Generator™ および ChipScope™ Pro などのその他のツールも使用できますが、このチュートリアルでは説明されません。このチュートリアルは、パーシャル リコンフィギュレーションおよびザイリンクス ソフトウェアを使用した FPGA デザインのインプリメンテーションを学ぶ必要のある場合に使用します。パーシャル リコンフィギュレーションの詳細は、『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) を参照してください。

メモ : このチュートリアルでは、ISE® Design Suite バージョン 12 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 UG743_design_files.zip リファレンス デザインを使用します。このファイルは、解凍しておいてください。リファレンス デザインのコピーは、<http://japan.xilinx.com/tools/partial-reconfiguration> からダウンロードできます。

パーシャル リコンフィギュレーションの機能を使用するには、パーシャル リコンフィギュレーションの FlexLM ライセンスを取得する必要があります。30 日間無料のライセンスを取得するには、ザイリンクスのフィールド アプリケーション エンジニア (FAE) にご連絡いただくか、ザイリンクス Web サイト <http://japan.xilinx.com/getproduct> にアクセスする必要があります。

ハードウェアのテスト用に、オプションで ML605 ボードおよび USB ダウンロード ケーブルを使用できます。

各チュートリアルを実行する前に、まず元のサンプル デザイン 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には、既にインプリメント済みのプロジェクト ファイルが含まれます。サンプル データのディレクトリでは、容量を削減するために、インプリメンテーション ファイルの一部が削除され、必要なデータのみが残されています。

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 Design Suite 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、リファレンス デザインが解凍されているかどう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

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、小型のデザインが使用されますので、1GB で十分ですが、パフォーマンスに影響のこともあります。

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

『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_1/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://japan.xilinx.com/planahead> を参照してください。

パーシャル リコンフィギュレーションの詳細は、<http://japan.xilinx.com/tools/partial-reconfiguration> から『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) 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を終了すると、PlanAhead ソフトウェアを使用してパーシャル リコンフィギュレーション プロジェクトを設定、実行および管理できるようになります。具体的には、リコンフィギュラブル パーティションを作成し、リコンフィギュラブル モジュールを追加し、そのリコンフィギュラブル パーティションの Pblock 範囲を定義し、パーシャル リコンフィギュレーション用の DRC チェックを実行し、コンフィギュレーションを作成およびインプリ

メントし、パーシャル リコンフィギュレーション 検証を実行し、ハードウェアでのパーシャル リコンフィギュレーションに必要な BIT ファイルを生成します。

チュートリアル デザインの説明

このチュートリアルの FPGA デザインは、<http://japan.xilinx.com/ml605> に記述されているザイリンクス ML605 プロトタイプ ボードをターゲットにしています。ターゲット デバイスは Virtex[®]-6 xc6vlx240tff1156-1 です。FPGA デバイスは読み込まれるリコンフィギャブル モジュールによって異なる順番で LED を駆動します。デザインにはリコンフィギャブル パーティションが 2 つ含まれ、そのうち 1 つにはエンベデッド ブロック RAM、もう 1 つはエンベデッド I/O バッファが使用されています。異なるブロック RAM データを含むブロック RAM モジュールをリコンフィギュレーションすると、8 個の GPIO LED の LED の順序が変更されます。異なるステートマシン遷移を含む I/O モジュールをリコンフィギュレーションすると、4 個の LED の回転方向が時計回りか反時計回りに変わります。

チュートリアルの概要

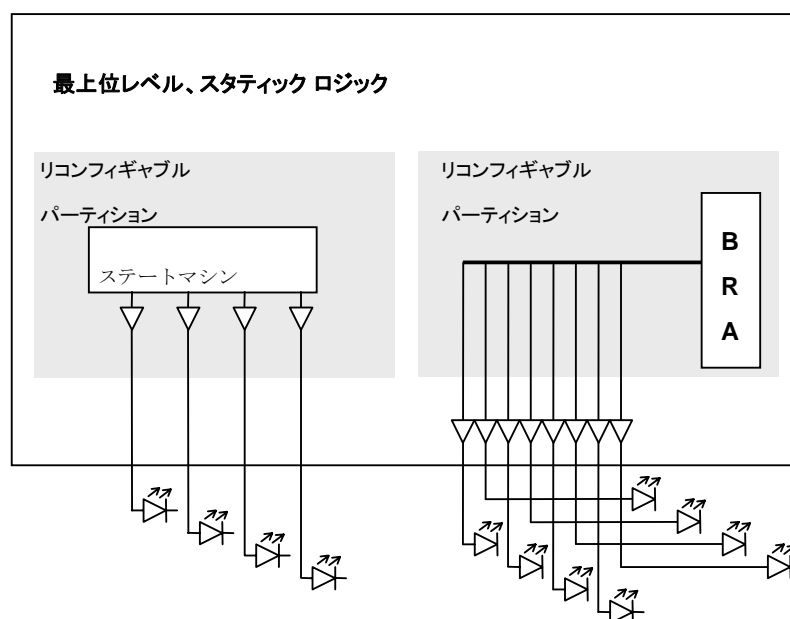


図 1 : デザインの概要 - パーシャル リコンフィギュレーション

ソフトウェア ツール フロー

パーシャル リコンフィギュレーションではトップダウン インプリメンテーションを使用してボトムアップ合成方法を使用します。このようなデザインおよびこのチュートリアルでは、XST を使用してデザインを合成し、インプリメンテーションに PlanAhead を使用します。その他のツールおよび手法を使用しても、問題なくパーシャル リコンフィギュレーション デザインをインプリメントできます。

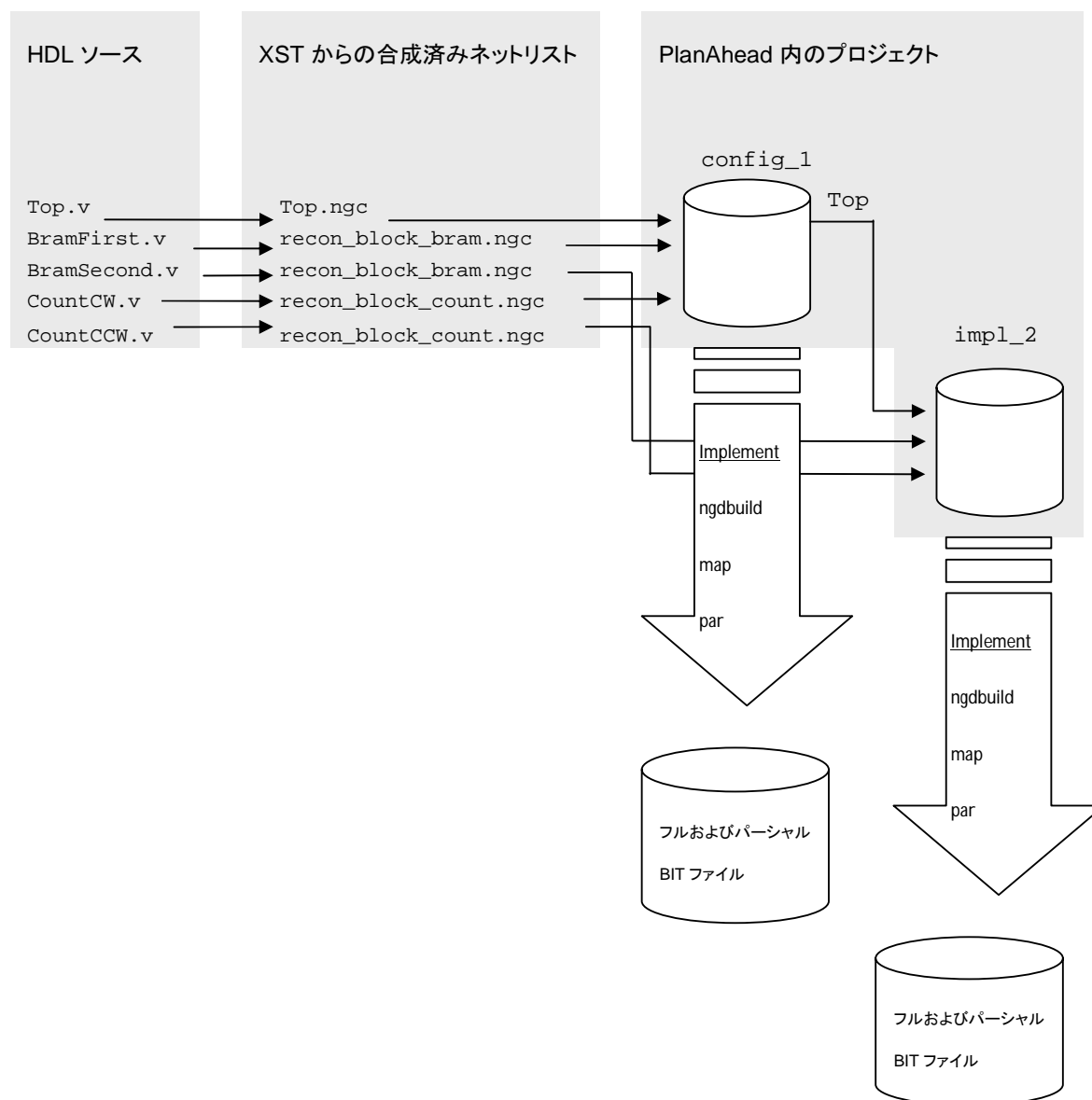


図 2 : ソフトウェア フローの概要

プロジェクト ディレクトリおよび HDL デザインの構造

部分的にリコンフィギャブルな FPGA デザインを構築して合成するには、ブラック ボックスとボトムアップ合成方法を使用する必要があります。リコンフィギャブル モジュールはそれぞれ個別のプロジェクトとして合成され、ネットリストが個別に生成されます。最上位レベルのデザインはリコンフィギャブル モジュールをブラック ボックスとしてインスタンスシートするので、リコンフィギャブル モジュールのネットリストは最上位レベルのネットリストに含まれません。

このチュートリアル of デレクトリ構造は次のようになります。

xpr_bram_led

- Implementation → 合成結果 (スクリプト手法が使用される場合はインプリメンテーション結果も)
- PlanAhead → PlanAhead プロジェクトおよび結果
- Source → HDL ソース ファイルと制約ファイル
- Tools → コマンド ライン フローの Tcl スクリプト (このチュートリアルでは説明なし)

リコンフィギャブル モジュールはそれぞれ別に合成されるので、最上位レベル モジュール用のディレクトリが別にあります。個別のリコンフィギャブル モジュールは Source および Implementation ディレクトリ内に含まれます。

Implementation

Top (最上位レベルとすべてのスタティック ロジック)

BramFirst (BRAM リコンフィギャブル モジュールの最初のバージョン)

BramSecond (BRAM リコンフィギャブル モジュールの 2 つ目のバージョン)

CountCW (カウンタの時計回りバージョン)

CountCCW (カウンタの反時計回りバージョン)

最上位レベルのソース ファイル xpr_bram_led/Source/Top/Top.v を開きます。このデザインの 2 つのリコンフィギャブル モジュール (recon_block_bram および recon_block_count) は HDL でブラック ボックスとして宣言されています。これらのブロックに対する下位の HDL 記述は提供されていません。

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。

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: HDL ソースからのネットリストの合成 (NGC が提供されるのでオプション)

手順 2: PlanAhead プロジェクトの作成

手順 3: リコンフィギャブル パーティションの作成およびリコンフィギャブル モジュールの追加

手順 4: リコンフィギャブル モジュールの追加

手順 5: ブラック ボックス モジュールの追加 (オプション)

手順 6: リコンフィギャブル パーティションのフロアプラン

手順 7: パーティション ピンとリコンフィギャブル パーティションのインターフェイス タイミング

手順 8: パーシャル リコンフィギュレーションのデザイン ルール チェック

手順 9: コンフィギュレーションのインプリメントとプロモート

手順 10: 追加コンフィギュレーションの作成とインプリメント

手順 11: パーシャル リコンフィギュレーション検証の実行

手順 12: BIT ファイルの生成とダウンロード

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。

HDLソースからのネットリストの合成(オプション)

手順1

PlanAhead ではパーシャル リコンフィギュレーション フローの HDL プロジェクトがサポートされないので、PlanAhead プロジェクトを作成する前に XST を使用して合成をしておく必要があります。このチュートリアルに含まれるファイルでは、XST を既に実行済みです。また、含まれる NCG ファイルを使用することもできます。NGC ファイルを使用する場合は、手順 2 に進んでください。

このデザインに対しては、XST プロジェクト ファイルが既に作成されています。XST の合成には、xpr_bram_led/Implementation/Top/Top.xst と xpr_bram_led/Implementation/Top/Top.prj の 2 つのプロジェクトファイルが必要です。

Top.xst では、I/O バッファが自動挿入されるようになっています。これはデフォルトです。

```
-iobuf YES
```

すべてのリコンフィギャブル モジュールの XST プロジェクト ファイルでは、この属性は NO に設定する必要があります。これは、下位レベルのモジュールには I/O バッファを挿入できないからです (後で説明される U2_RP_Count リコンフィギャブル パーティションのような特殊な場合を除く)。

1-1 xpr_bram_led/Tools ディレクトリで Tcl スクリプトを実行してすべてのモジュールを合成します。

1-1-1 xpr_bram_led/Implementation ディレクトリから次のコマンドを実行します。

```
tclsh ../Tools/xpartition.tcl ../Tools/data_synth.tcl
```

このスクリプトにより、XST が呼び出され、Source ディレクトリの Verilog ファイルが合成されます。

メモ : xpr_bram_led/Source/Bram* モジュールの場合は、リコンフィギャブル モジュールにブロック RAM が、xpr_bram_led/Source/Count* モジュールの場合は、I/O バッファが含まれます。XST で生成された NGC ネットリスト ファイルは xpr_bram_led/Implementation/<module> ディレクトリに保存されています。

1-1-2 合成ツールに Synplify Pro を使用する場合は、上記を実行前に data_synth.tcl ファイルを次のように修正してください。

```
SYNTH_TOOL "synplify_pro" \
```

このオプションを設定すると、xpr_bram_led/Implementation/<module> ディレクトリの Synplify Pro プロジェクトファイルが使用されます。

プロジェクトを開く

手順2

2-1 PlanAhead を起動し、新規プロジェクトを作成します。

2-1-1 PlanAhead ソフトウェアを起動します。

- Windows の場合、Xilinx PlanAhead 12.1 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 の Getting Started ページが開きます。

2-1-2 [Create New Project] のリンクをクリックします。

[Create a New PlanAhead Project] ページが開きます。

2-1-3 [Next] をクリックします。

[Project Name] ページが開きます。

2-1-4 プロジェクト名とディレクトリを選択し、[Next] をクリックします。

2-1-5 [Specify synthesized (EDIF or NGC) netlist] をオンにし [Set PR Project] をオンにしたら [Next] をクリックします (図 3)。

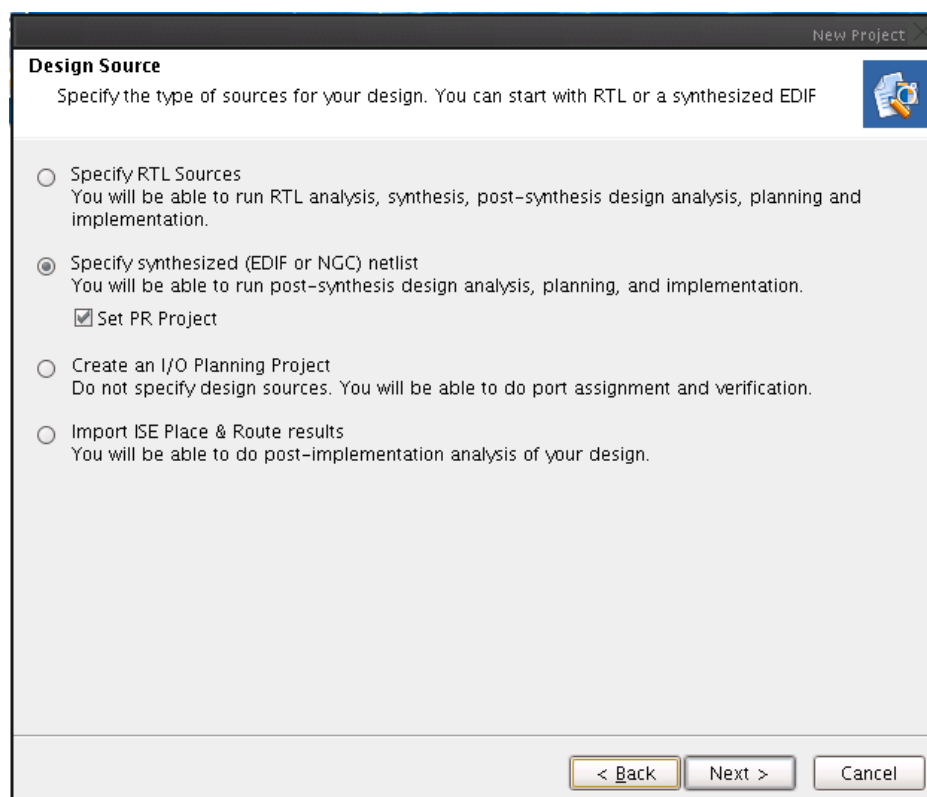


図 3 : デザイン ソースの指定

- 2-1-6 [Top Netlist File] で `xpr_bram_led/Implementation/Top/Top.ngc` を参照し、**[開く]** をクリックし、**[Next]** をクリックします。オプションのネットリスト ディレクトリは設定しないでください。

メモ : オプションのネットリスト ディレクトリは、スタティック ロジックに関連する下位レベルのネットリストがある場合にのみパーシャル リコンフィギュレーション プロジェクトで使用されるべきです。リコンフィギャブル モジュールに関連する下位レベルのネットリストは後で追加します。

- 2-1-7 [Constraint Files] ページで **[Add Files]** ボタンをクリックし、`xpr_bram_led/Source/UCF/top_ml605.ucf` のユーザー制約ファイル (UCF) を指定します。**[Next]** をクリックします。
- 2-1-8 [Default Part] ページでは、ネットリストがスキャンされ、最適なパーツが自動的に選択されます。**xc6vlx240tff1156-1** デバイスが選択されていることを確認し、**[Next]** をクリックします。
- 2-1-9 [New Project Summary] ページで図 4 のようにプロジェクトが設定されていることを確認したら、**[Finish]** をクリックします。

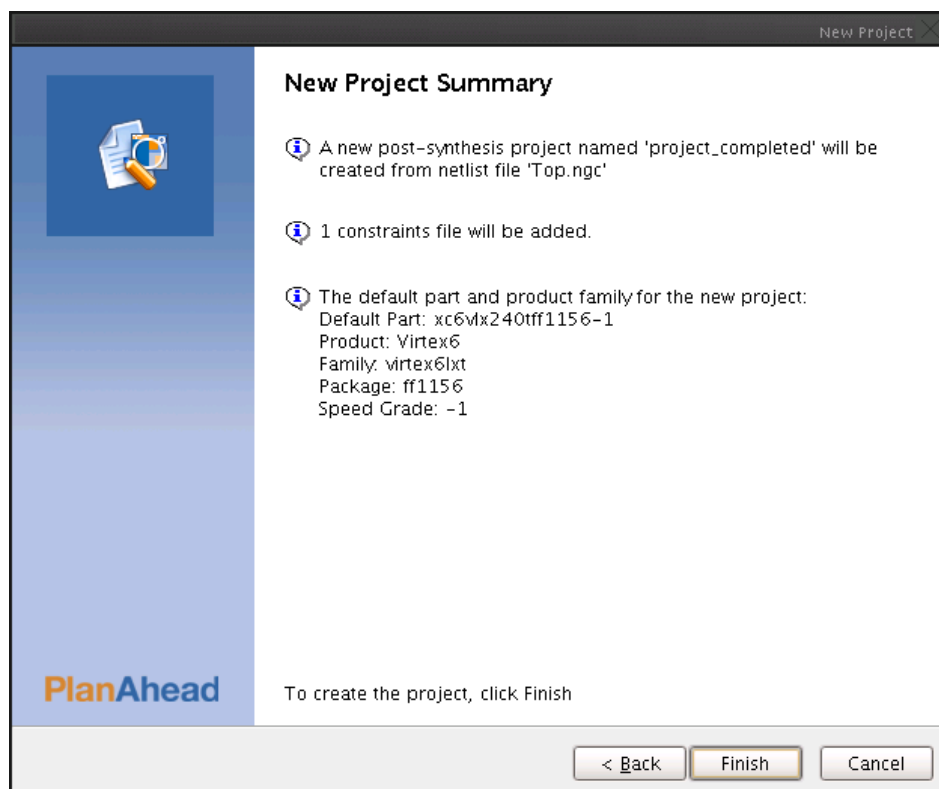


図 4 : 新規プロジェクトのサマリ

パーシャルリコンフィギュレーションの作成とリコンフィギャブル モジュールの追加手順3

3-1 U1_RP_Bram のリコンフィギャブル パーティションを作成します。

3-1-1 Flow Navigator の [Netlist Design] ボタンをクリックしてネットリストをメモリに読み込みます。

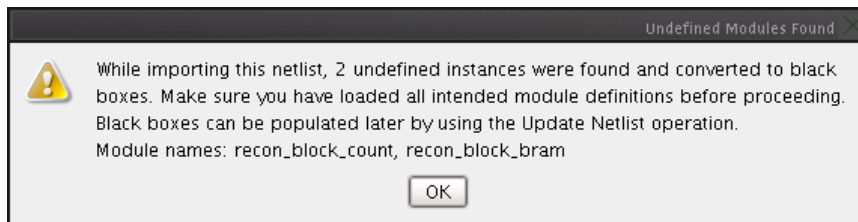


図 5 : 定義されていないモジュールがあることを示す警告メッセージ

上記の定義されていないインスタンスがあることを示すメッセージが表示されます。これは、リコンフィギャブルモジュールに割り当てられたネットリストがまだないためです。[OK] をクリックします。

メモ : このチュートリアルで記述されるウィンドウやツールの多くは、[Netlist Design] ビューが開いている場合のみ使用可能です。[Netlist Design] を閉じた場合、またはプロジェクトを一度閉じてから開きなおした場合は、上記の [Netlist Design] ボタンをクリックして [Netlist Design] ビューを開きなおす必要があります。

3-1-2 [Netlist] ビューで U1_RP_Bram を選択して右クリックし、[Set Partition] を選択します。これにより Set Partition ウィザードが起動されます。

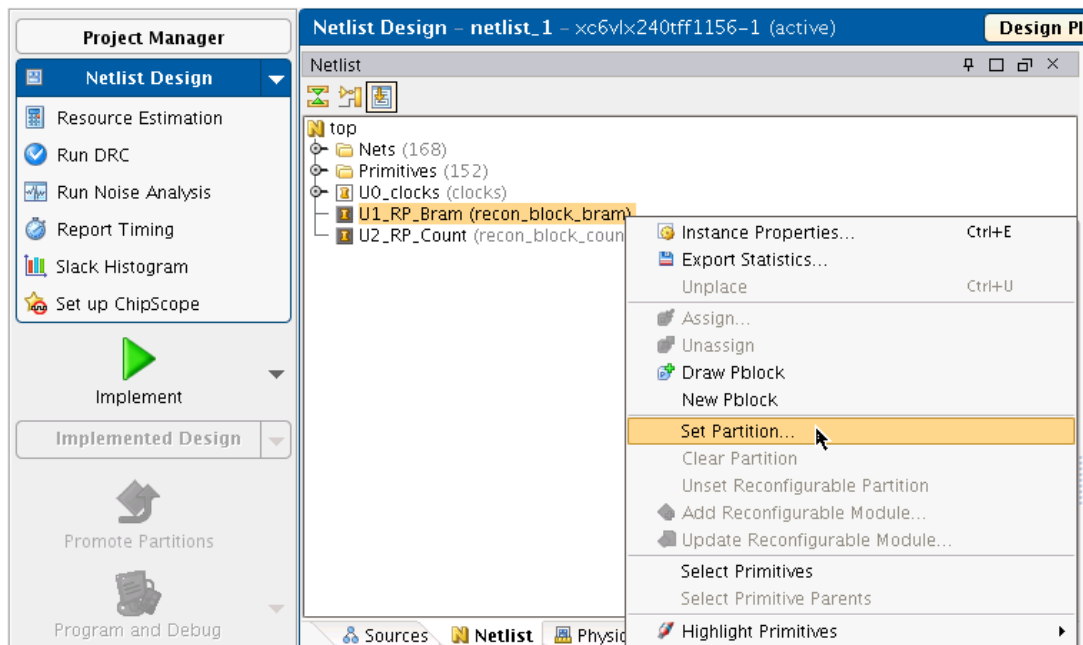


図 6 : リコンフィギャブル モジュールのパーティション設定

3-1-3 ウィザードの最初のページで [Next] をクリックし、選択されていない場合は **[is a reconfigurable Partition]** をオンにし、[Next] をクリックします。

3-1-4 [Reconfigurable Module Name] に **BramFirst** という名前を付け、選択されていない場合は **[Netlist already available for this Reconfigurable Module]** を選択し、[Next] をクリックします。



3-1-5 [Top Netlist File] で **xpr_bram_led/Implementation/BramFirst/recon_block_bram.ngc** を参照します。

3-1-6 [開く] をクリックし、[Next] をクリックします。

リコンフィギャブル モジュールに関連する下位レベルのネットリストがある場合、オプションのネットリスト ディレクトリはここで追加できますが、この場合はありません。

3-1-7 [Next] をクリックしてオプションの制約ファイルの画面を飛ばします。ここでモジュールレベルの制約ファイルを追加できますが、この場合はありません。

3-1-8 [Set Partition Summary] ページを確認したら **[Finish]** をクリックしてウィザードを終了します。

これで、U1_RP_Bram のリコンフィギャブル パーティションが作成されました。[Netlist] ビューのアイコンは  から  に変更され、U1_RP_Bram の下にリコンフィギャブル モジュールが 1 つ表示されます。このインスタンスは、[Physical Constraints] ビューで Pblock として表示されるようになっています。

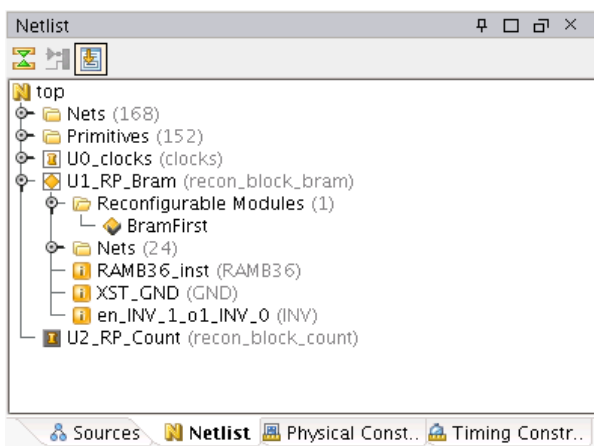


図 7 : リコンフィギャブル パーティションのアイコン

3-2 3-1 の手順に従って U2_RP_Count のリコンフィギャブル パーティションを作成します。

3-2-1 U2_RP_Count を選択して右クリックし、[Set Partition] を選択します。

3-2-2 次の設定で Set Partition ウィザードを終了します。

- 選択されていない場合は **[is a reconfigurable Partition]** をオン
- リコンフィギャブル モジュール名を **CountCW** に指定
- 選択されていない場合は **[Netlist already available for this Reconfigurable Module]** をオン
- [Top Netlist File] を **xpr_bram_led/implementation/CountCW/recon_block_count.ngc** に設定

これにより、リコンフィギャブル パーティションそれぞれに 1 つのリコンフィギュレーション モジュールが含まれることとなります。次の手順では、リコンフィギャブル パーティションにリコンフィギャブル モジュールを追加する方法について説明します。

リコンフィギャブル モジュールの追加

手順4

4-1 RP U1_RP_Bram リコンフィギャブル パーティションにリコンフィギャブル モジュールを追加します。

4-1-1 [Netlist] ビューで U1_RP_Bram を選択して右クリックし、[Add Reconfigurable Module] を選択します。これにより Add Reconfigurable Module ウィザードが起動されます。

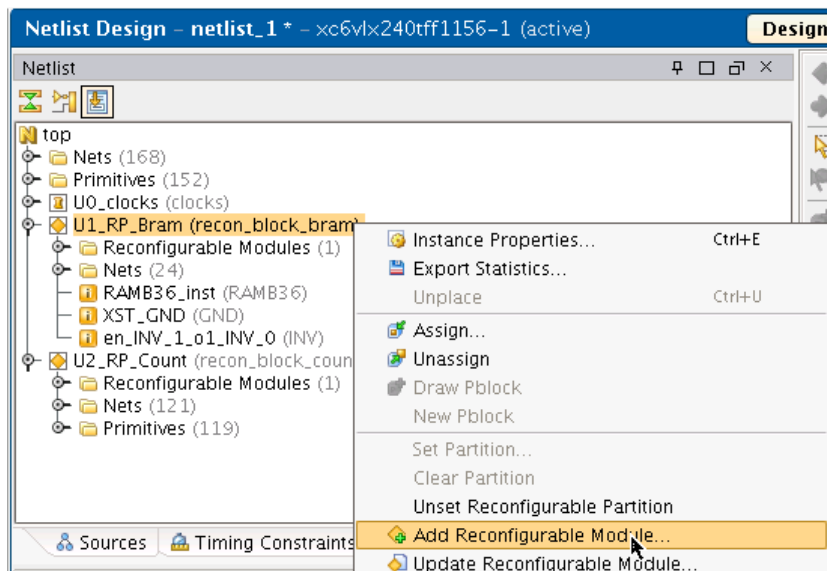


図 8 : リコンフィギャブル モジュールの追加

4-1-2 最初のページで [Next] をクリックします。

4-1-3 [Reconfigurable Module] に **BramSecond** と入力し、選択されていない場合は [Netlist already available for this Reconfigurable Module] をオンにし、[Next] をクリックします。

4-1-4 [Top Netlist File] で **xpr_bram_led/Implementation/BramSecond/recon_block_bram.ngc** を参照します。

4-1-5 オプションのネットリスト ディレクトリに追加する下位レベルのネットリストはないので、[Next] をクリックして続行します。

4-1-6 [Next] をクリックしてオプションのモジュール レベル制約ファイルの画面を飛ばします。


4-1-7 [Add Reconfigurable Module Summary] ページを確認したら [Finish] をクリックしてウィザードを終了します。

5-2 5-1 の手順に従って U2_RP_Count にリコンフィギャブル モジュールを追加します。

5-2-1 U2_RP_Count を右クリックし、[Add Reconfigurable Module] を選択します。

5-2-2 次の設定で Add Reconfigurable Module ウィザードを終了します。

- リコンフィギャブル モジュール名を **CountCCW** に指定
- 選択されていない場合は [Netlist already available for this Reconfigurable Module] をオン
- [Top Netlist File] を xpr_bram_led/Implementation/CountCCW/recon_block_count.ngc に設定

この段階で、リコンフィギャブル モジュールが 1 つ各リコンフィギャブル パーティションに追加され、[Netlist] ビューは図 9 のように表示されているはずです。各リコンフィギャブル パーティションの下にはネットおよびプリミティブがリストされます。これらのネットおよびプリミティブは現在アクティブになっているリコンフィギャブル モジュール特有のもので、チェックマークの付いた黄色のひし形アイコン  が表示されます。図 9 では、BramSecond および CountCCW がアクティブ モジュールです。アクティブにするリコンフィギャブル モジュールを変更する場合は、それを右クリックして [Set as Active Reconfigurable Module] を選択します。

少し時間をとって、さまざまなリコンフィギャブル モジュールに関連するプリミティブを確認してください。U1_RP_Bram に関連するリコンフィギャブル モジュールの RAMB36、CountCW/CountCCW の Primitives フォルダの下には、スライス ロジック (LUT、XORY、FDR) があるほか、OBUF プリミティブがあります。OBUF プリミティブは、手順 6 の「リコンフィギャブル パーティションのフロアプラン」でリコンフィギャブル パーティションのエリアグループ範囲に含める必要があるため、特に重要です。

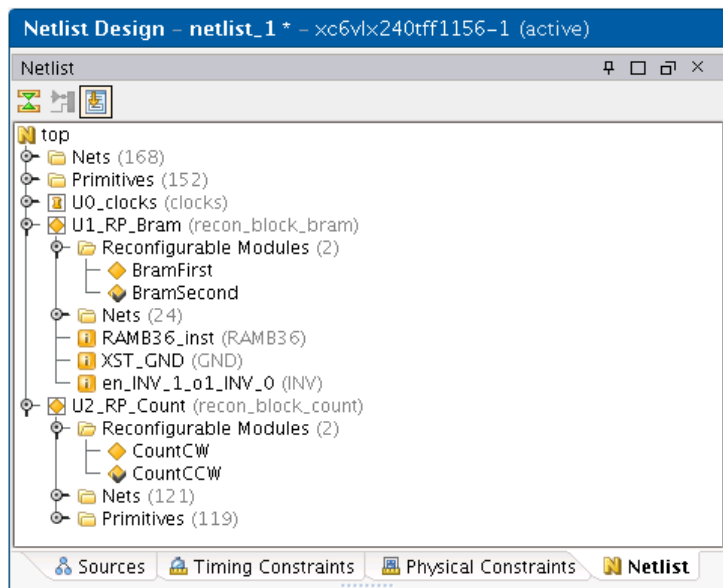


図 9 : リコンフィギャブル モジュールを追加した後の [Netlist] ビュー

ブラックボックスモジュールの追加(オプション)

手順5

ブラック ボックス モジュールはすべてのパーシャル リコンフィギュレーション デザインに必要なわけではないので、この手順はオプションです。ブラック ボックス モジュールを作成すると、BitGen 段階で空白の BIT ファイルを生成できます。この BIT ファイルの使用については、<http://japan.xilinx.com/tools/partial-reconfiguration> から『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702)を参照してください。

5-1 ブラック ボックス モジュールを U1_RP_Bram へ追加します。

5-1-1 [Netlist] ビューで U1_RP_Bram を右クリックし、[Add Reconfigurable Module] を選択します。これにより Add Reconfigurable Module ウィザードが起動されます。

5-1-2 最初のページで [Next] をクリックします。

5-1-3 リコンフィギャブル モジュールに **BramBB** という名前を付け、[Add this Reconfigurable module as a black box without a netlist] を選択し、[Next] をクリックします。

5-1-4 ブラック ボックス モジュールに関連するネットリストや制約ファイルはないので、ウィザードではこの情報について尋ねるウィンドウは表示されません。

[Add Reconfigurable Module Summary] ページを確認したら [Finish] をクリックしてウィザードを終了します。

5-2 5-1 の手順に従って U2_RP_Count にブラック ボックス モジュールを追加します。

5-2-1 [Netlist] ビューで U1_RP_Bram を右クリックし、[Add Reconfigurable Module] を選択します。これにより Add Reconfigurable Module ウィザードが起動されます。

5-2-2 次の設定で Add Reconfigurable Module ウィザードを終了します。

- リコンフィギャブル モジュール名を **CountBB** に指定
- [Add this Reconfigurable module as a black box without a netlist] を選択

このオプションの手順を終了すると、[Netlist] ビューの各リコンフィギャブル パーティションの下にブラック ボックス モジュールが 1 つずつ表示されます (図 10)。リコンフィギャブル モジュールのネットまたはプリミティブは表示されなくなっています。これはブラック ボックス リコンフィギャブル モジュールが現在アクティブで、このモジュールに関連するロジックまたはネットがないためです。

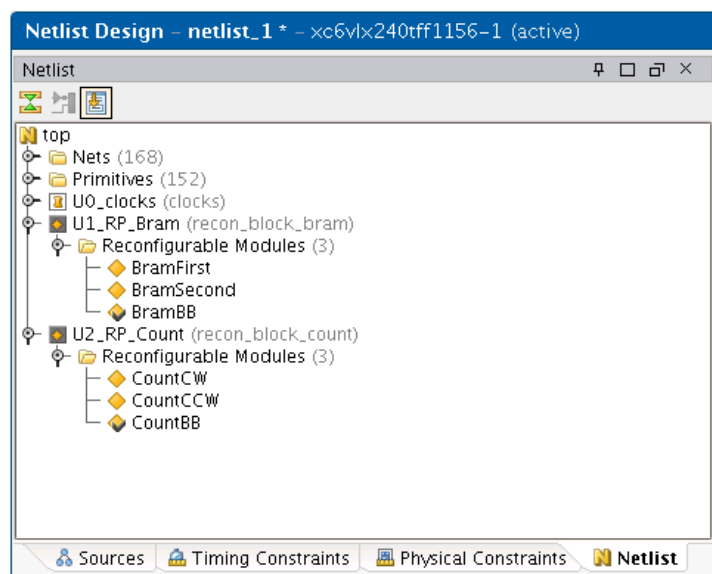


図 10 : ブラック ボックス モジュールを含む [Netlist] ビュー

リコンフィギャブルパーティションのフロアプラン

手順6

サンプル デザインのリコンフィギャブル パーティションの U1_RP_Bram および U2_RP_Count には、どの物理リソースがリコンフィギャブル パーティションの一部であるか指定する AREA_GROUP 範囲制約を含める必要があります。

リコンフィギャブル パーティションに関連する AREA_GROUP 範囲制約の一部ではない物理制約は、すべてスタティック ロジックの一部です (スタティック ロジックはパーシャル リコンフィギュレーションの影響を受けないので、リコンフィギュレーション プロセス中も動作可能な状態のままになります)。AREA_GROUP 範囲制約は、リコンフィギャブル パーティションが手順 3 の「リコンフィギャブル パーティションの作成とリコンフィギャブル モジュールの追加」で説明された [Set Partition] コマンドを使用して作成されるまで、作成するべきではありません。

6-1 pblock_U1_RP_Bram の AREA_GROUP 範囲を作成します。

6-1-1 [Netlist] ビューで BramFirst と CountCW をそれぞれ右クリックし、どちらでも **[Set as Active Reconfigurable Module]** を選択して、アクティブなリコンフィギャブル モジュールにします。

特定のリコンフィギャブル モジュールに必要なリソースはどのリコンフィギャブル モジュールがアクティブかによって異なるので、ブラック ボックスリコンフィギャブル モジュールをアクティブにしてしまうと、AREA_GROUP 範囲に必要な正しいリソースがレポートされません。1 つのリコンフィギャブル パーティションに関連付けられた異なるリコンフィギャブル モジュールがさまざまなリソースを使用する場合、リコンフィギャブル パーティションの AREA_GROUP 範囲にはリコンフィギャブル モジュールで使用されるリソースの上位集合を含める必要があります。

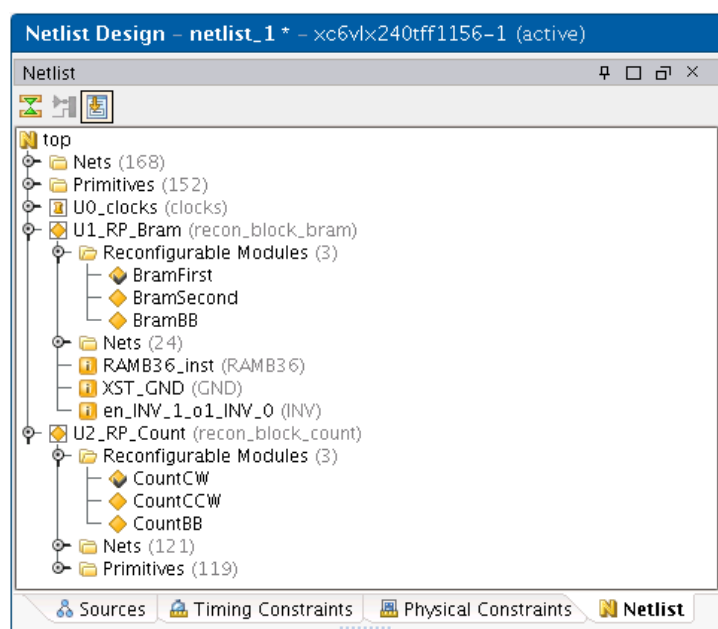



図 11 : アクティブなリコンフィギャブル モジュールの設定

6-1-2 [Physical Constraints] タブをクリックし、現在の Pblock すべてのリストを表示します。PlanAhead では AREA_GROUP 制約が Pblock と呼ばれ、リコンフィギャブル パーティションとして定義されたモジュールの Pblock が自動的に作成されます。[Physical Constraints] ビューでリコンフィギャブル パーティションの **pblock_U1_RP_Bram** を選択します。

6-1-3 [Device] ビューの左側で [Set Pblock Size] ボタン ( **Set Pblock Size**) をクリックします。**pblock_U1_RP_Bram** を右クリックし、[Set Pblock Size] を選択しても同じ操作を実行できます。

6-1-4 複数のスライス ロジックと少なくとも RAMB36 (ピンクの列) を 1 つ含めるような長方形を描画します。

6-1-5 長方形を描画したら、図 12 のように [SLICE] と [RAMB36] をオンにし、[OK] をクリックします。

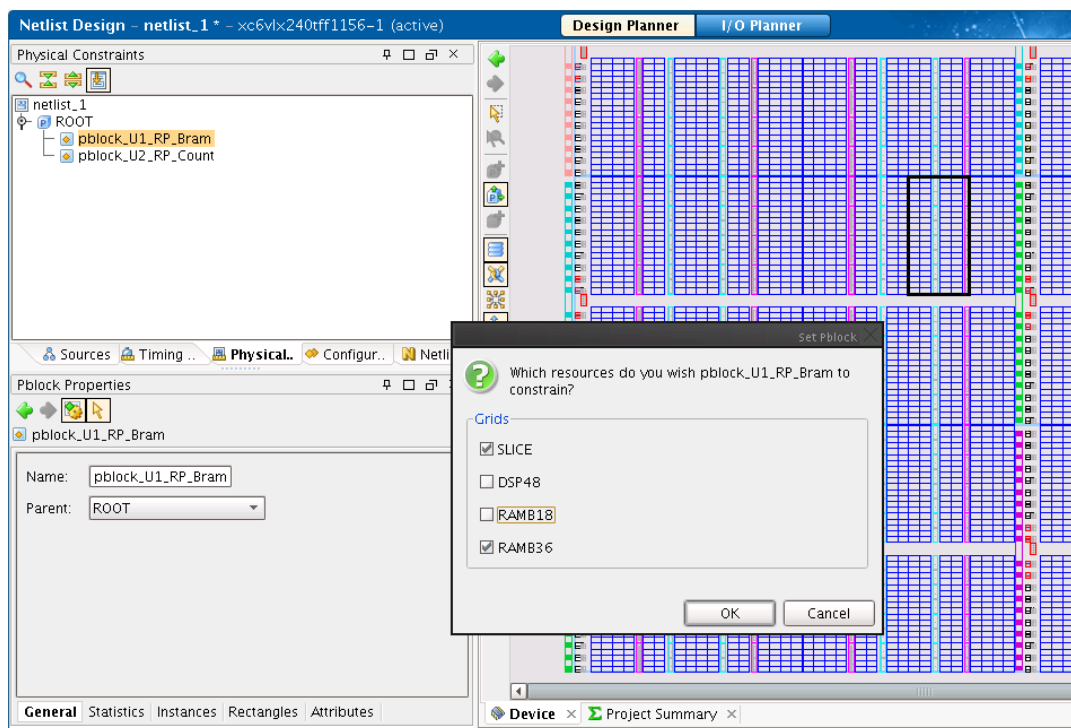


図 12 : Pblock リソース (pblock_U1_RP_Bram) の設定

6-2 6-1 の手順に従って pblock_U2_RP_Count の AREA_GROUP 範囲を作成します。

- 6-2-1 pblock_U2_RP_Count を選択し、[Set Pblock Size] を実行するか、スライス ロジックと I/O ロジックを含めるような長方形を描画します。U2_RP_Count には UCF ファイルで I/O 配置が既に含まれるので、AREA_GROUP 範囲には AD21、AH27、AE21、AH28 ピンを含める必要があります。これらのピンはフロアプランで確認できます。
- 6-2-2 長方形を描画したら、図 13 のように [SLICE] と [IOB] リソースをオンにし、[OK] をクリックします (図 3)。リコンフィギュラブル モジュールに入力または出力フリップフロップ (またはその他の入力/出力ロジック) がある場合は、ILOGIC および OLOGIC のようなその他のロジックもオンにできます。このデザインには、そのようなリソースは含まれません。

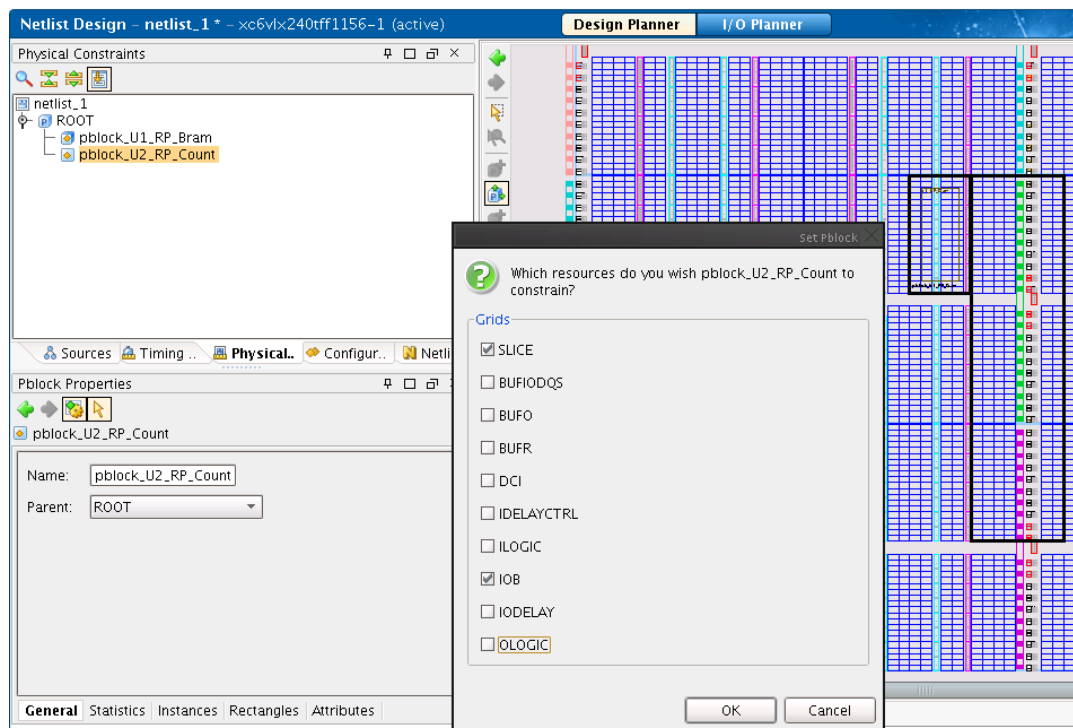


図 13 : Pblock リソース (pblock_U2_RP_Count) の設定

フロアプランを確認し、必要な変更を加えます。

6-2-3 結果のフロアプランは、図 14 のようになるはずです。

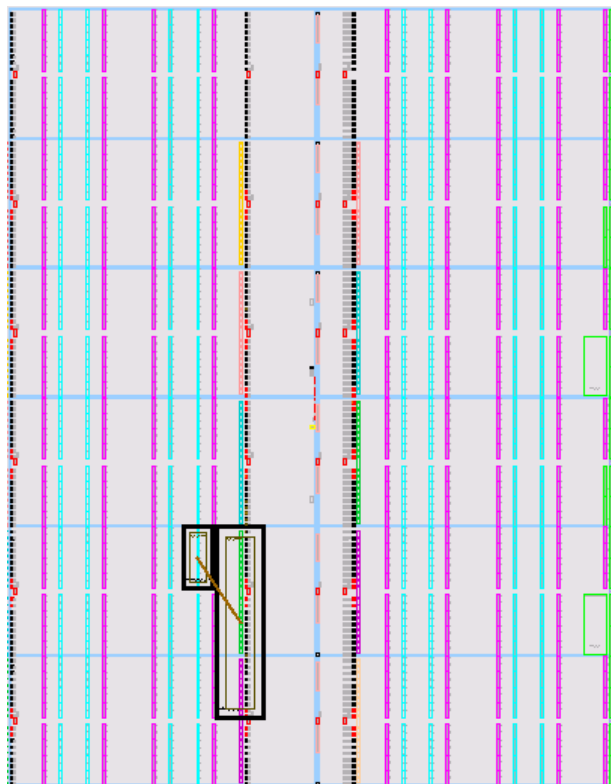


図 14 : 最終的な AREA_GROUP 範囲

6-2-4 U2_RP_Count に必要なピンが AREA_GROUP 範囲内にあるかどうかを確認します。AD21、AH27、AE21 および AH28 ピンが図 14 のように の AREA_GROUP 範囲の長方形内に含まれる必要があります。また、この長方形が I/O 列の右側を含めていることに注意してください。この I/O 列の右側には I/O 配線リソースがあるので、AREA_GROUP 範囲に含めないと、配線でインプリメンテーション エラーになります。これは、リコンフィギャブル モジュールの配線リソースが、すべて AREA_GROUP 範囲で定義されるように、リコンフィギャブル パーティション内に含まれる必要があるからです。

[Device] ビューでこれらのピンが AREA_GROUP 範囲内にあるかどうかを確認するには、[Edit] → [Find] をクリックし、AD21 を名前に含むサイトを検索します。UCF には既に LOC 制約が含まれるので、このピンが [Device] ビューでハイライトされます。[View] → [Fit Selection] (F9) をクリックし、ハイライトされたサイトを拡大します。同じ手順で AH27、AE21、AH28 ピンも検索できます。

6-2-5 U1_RP_Bram に必要な RAMB36 が AREA_GROUP 範囲内にあるかどうかを確認します。[Physical Constraints] ビューで pblock_U1_RP_Bram をクリックし、[Pblock Properties] ビューで [Statistics] を確認します (図 15)。RAMBFIFO36E1 の使用可能な数が必要な数を超過していることがわかります。

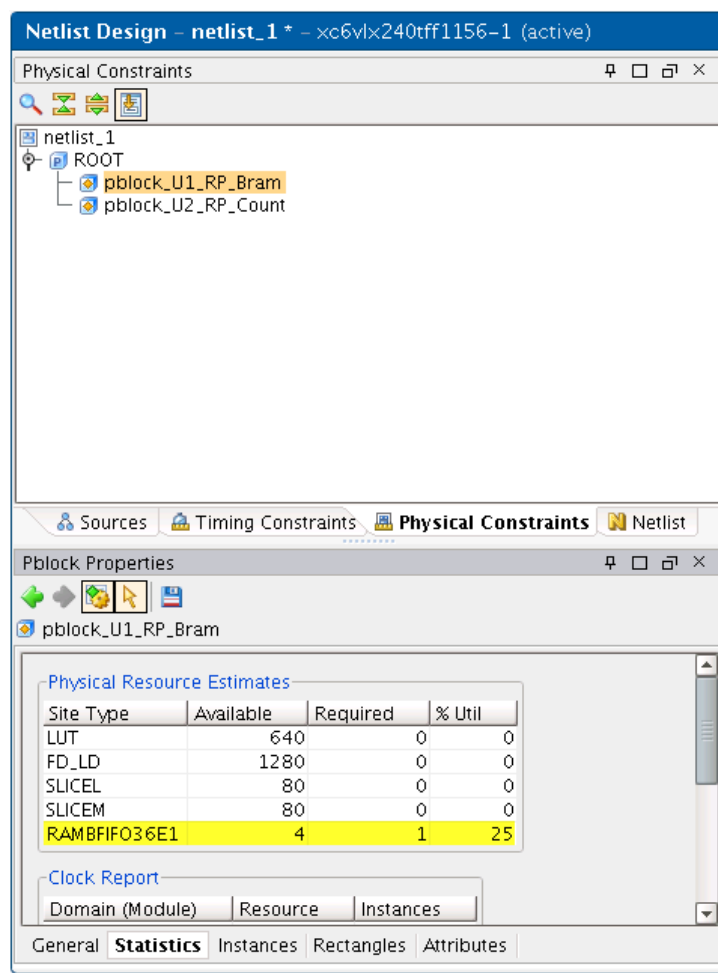


図 15 : Pblock (pblock_U1_RP_Bram) の統計

- 6-1-6 前の手順で作成した AREA_GROUP 範囲制約は PlanAhead で表示できます。まず [File] → [Save Design] をクリックしてデザインを保存します。
- 6-1-7 Flow Navigator で [Project Manager] をクリックします。
これにより [Sources] ビューが起動され、デザインソースおよび制約がすべて表示されます。
- 6-1-8 [Constraints] の下で UCF ファイルを探し、ダブルクリックしてファイルを開きます。
次のようなエリア グループ制約が表示されるはずです。

```
INST "U1_RP_Bram" AREA_GROUP = "pblock_U1_RP_Bram";

AREA_GROUP "pblock_U1_RP_Bram" RANGE=SLICE_X48Y60:SLICE_X55Y79;

AREA_GROUP "pblock_U1_RP_Bram" RANGE=RAMB36_X3Y12:RAMB36_X3Y15;

INST "U2_RP_Count" AREA_GROUP = "pblock_U2_RP_Count";

AREA_GROUP "pblock_U2_RP_Count" RANGE=SLICE_X56Y20:SLICE_X67Y79;

AREA_GROUP "pblock_U2_RP_Count" RANGE=IOB_X1Y20:IOB_X1Y79;
```

パーティションピンとリコンフィギャブルパーティションのインターフェイスタイミング手順7

パーティション ピンは、グローバル ロジックでも専用配線でもないすべてのリコンフィギャブル パーティションのインターフェイス信号のパーシャル リコンフィギュレーション フローで必要です。パーティションはリコンフィギャブル パーティションへ既知の配線を提供し、インプリメンテーションが実行されると NGDBuild で自動的に挿入されます (これらは以前のバージョンのパーシャル リコンフィギュレーション フローのバス マクロに代わるテクノロジーです)。

パーティション ピンの現在のインプリメンテーションには、プロキシ ロジックの LUT1 が必要です。LUT1 は、リコンフィギャブル パーティションの入力および出力パスに挿入されますので、リコンフィギャブル パーティションの両側の境界でこれらの入力および出力にレジスタを付けてください。これにより、リコンフィギャブル パーティション インターフェイスに関連するタイミング クロージャ問題を最小限に抑えることができます。これらのガイドラインに従うと、このインターフェイスに制約を付けるには単純な PERIOD 制約で十分なことがほとんどです。ただし、タイミング要件の厳しい場合、パーティション ピンに TPSYNC 制約を作成するか、スタティック ロジックに LOC 制約を追加して、スタティック ロジックとパーティション ピン間の配線遅延を抑える必要があります。パーティション ピンへの TPSYNC 制約の追加については、<http://japan.xilinx.com/tools/partial-reconfiguration> から『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) を参照してください。

コンフィギュレーションすべてで同じスタティック ロジックのインプリメンテーションが使用されるので、まず最もタイミング クリティカルなリコンフィギャブル モジュールのタイミングを満たすことが重要です。このチュートリアルのリコンフィギャブル パーティション インターフェイスは、さまざまなリコンフィギャブル モジュール間で非常に類似しているため、グローバル PERIOD 制約で十分にリコンフィギャブル インターフェイス タイミングを満たすことができます。

7-1 PERIOD を追加してデザイン (リコンフィギャブル パーティション インターフェイスを含む) に制約を付けます。

7-1-1 [Netlist Design] ビューで [Timing Constraints] タブをクリックします。

7-1-2 このタブで右クリックして [New Timing Constraint] を選択するか、上部の [New Timing Constraint] ボタン (🏠) をクリックします。

7-1-3 [Basic group (TNM)] グループ カテゴリをクリックし、次の値を設定します。

- [Group name] : **clk_p**
- [Group type] : **Net**
- [TNM type] : **TNM_NET**
- [Predefined group] : 空白
- [Net] : clk_p

7-1-4 [OK] をクリックし、制約を追加します。

7-1-5 もう 1 度 [New Timing Constraint] を実行します。[Timespec period] グループ カテゴリをクリックし、次の値を設定します。

- [TimeSpec name] : **TS_clk_p**
- [Period] : 5 ns
- [Group] : 参照ボタンをクリックし [Group constraints type] で User defined を、[User defined groups] で clk_p を設定

7-1-6 [OK] をクリックし、制約を追加します。[Timing Constraint] ビューは図 16 のようになっているはずです。これが clk_p に接続される同期パスすべてを制約するグローバル タイミング制約です。これはリコンフィギュラブル パーティション用のタイミング制約ではありません。

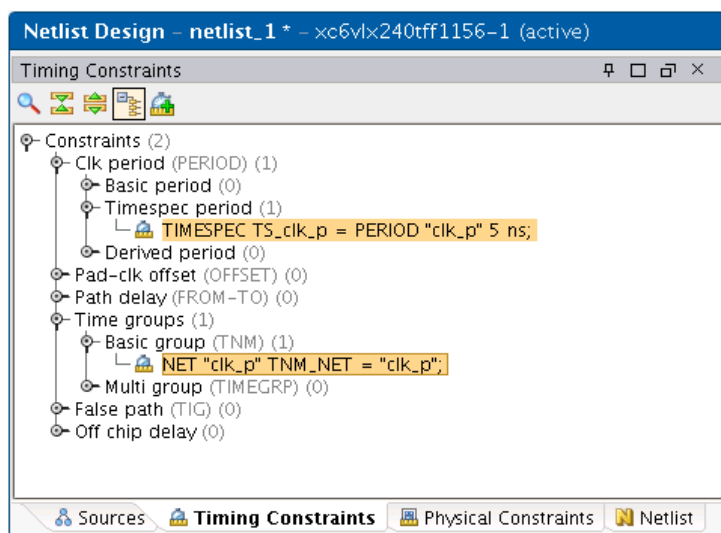


図 16 : [Timing Constraints] ビュー (pblock_U1_RP_Bram)

7-1-7 [File] → [Save Design] をクリックしてデザインを保存し、Project Manager に戻って UCF を確認します。UCF 制約ファイルには次の制約が含まれるはずです。

```
TIMESPEC TS_clk_p = PERIOD "clk_p" 5 ns;
```

```
NET "clk_p" TNM_NET = "clk_p";
```

パーシャルリコンフィギュレーションのデザイン ルールチェック

手順8

有効なデザインをインプリメントするには、パーシャル リコンフィギュレーション特有のデザイン ルールに従う必要があります。これらのルールの中には、PlanAhead DRC に含まれているものもあります。これらのチェックは、コンフィギュレーションをインプリメントして BIT ファイルを生成する前に、パーシャル リコンフィギュレーション デザインで実行する必要があります。

通常のデザインでは、すべての PlanAhead DRC を実行することをお勧めします。このチュートリアルでは、パーシャル リコンフィギュレーションとパーティションの DRC のみを実行します。

8-1 パーシャル リコンフィギュレーションとパーティションの DRC を実行します。

8-1-1 Flow Navigator で [Netlist Design] の下の [Run DRC] ボタンをクリックします (図 17)。

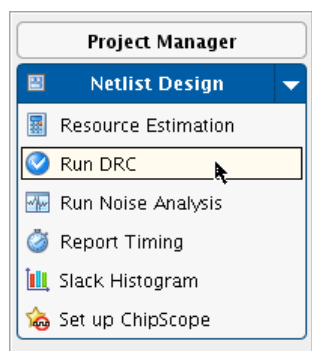


図 17 : [Run DRC] ボタン

8-1-2 [Run DRC] ダイアログ ボックスで [Partition] と [Partial Reconfig] ルールをオンにし、[OK] をクリックします。

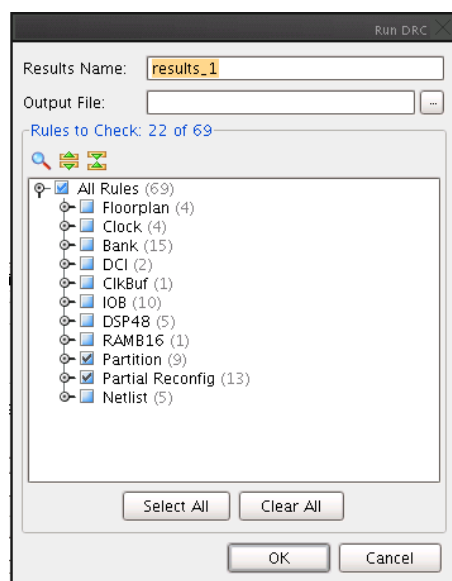


図 18 : [Partition] と [Partial Reconfig] の DRC の指定

8-1-3 [DRC Results] ビュー [Details] 列でメッセージを確認します。[Severity] 列に注目してください。Advisory (アドバイザリ)、Warning (警告)、Error (エラー)、Fatal (致命エラー) のいずれかが表示されます。この場合、一番危険度の高いのは Warning です (図 19)。

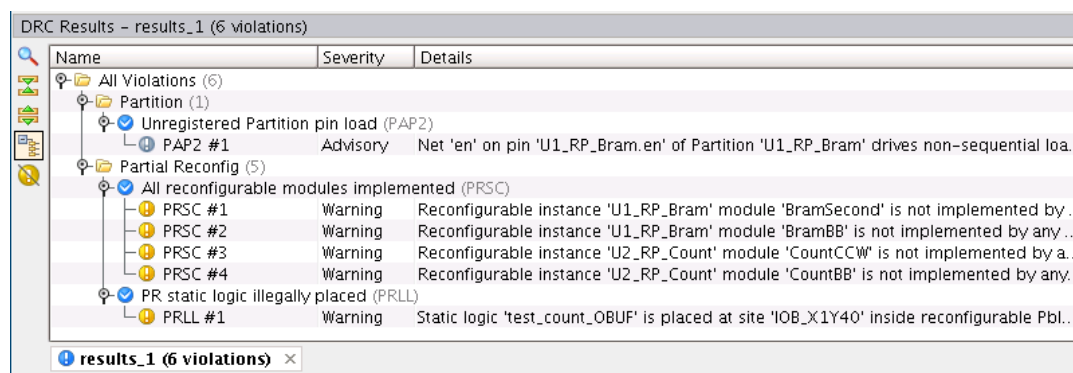


図 19 : [DRC Results] ビュー

コンフィギュレーションのインプリメントとプロモート

手順9

各リコンフィギャブル パーティションにはそれに関連するリコンフィギャブル モジュールが複数含まれますが、リコンフィギャブル パーティションには 1 度に 1 つのリコンフィギャブル モジュールしかインプリメントできません。

アクティブなリコンフィギャブル モジュールとスタティック ロジックを合わせたものは、「コンフィギュレーション」と呼ばれ、これだけで完全なデザインです。1 つのパーシャル リコンフィギュレーション プロジェクトに対して複数のコンフィギュレーションがあるので、さまざまなリコンフィギャブル モジュールをインプリメントでき、フルおよびパーシャル BIT ファイルが生成できます。各コンフィギュレーションのインプリメンテーションは独立して実行され、NGD、NGM、NCD、PCF およびレポート ファイルといったファイルが出力されます。特定コンフィギュレーションの NCD を FPGA Editor で開いたり、ゲート レベルのシミュレーションを実行したり、各コンフィギュレーションにはそれぞれザイリンクスのソフトウェア ツールおよびデバッグ機能を使用できます。

このチュートリアルのデザインは、次のリコンフィギャブル モジュール セットとその結果の BIT ファイルを使用した 2 つのコンフィギュレーションのみで完全にインプリメントできます。

Configuration

```
config_1          RMs:  BramFirst, CountCW
                  Bits: config_1.bit (full bit file)
                   config_1_U1_RP_Bram_BramFirst_partial.bit
                   config_1_U2_RP_Count_CountCW_partial.bit

config_2          RMs:  BramSecond, CountCCW
                  Bits: config_2.bit (full bit file)
                   config_2_U1_RP_Bram_BramSecond_partial.bit
                   config_2_U2_RP_Count_CountCCW_partial.bit
```

フル BIT ファイルの config_1.bit には BramFirst および CountCW リコンフィギャブル モジュールが、config_2.bit には BramSecond と CountCCW リコンフィギャブル モジュールが含まれます。

その他 2 つのコンフィギュレーション セットでも独自のフル BIT ファイルの生成が可能ですが、これらは前にインプリメントされたモジュールを再利用するので、パーシャル BIT ファイルは上記のコンフィギュレーションで生成されたパーシャル BIT ファイルとまったく同じになります。

```
Configuration
```

```
-----
```

```
config_3          RMs:  BramFirst, CountCCW

                  Bits: config_3.bit (full bit file)

                        config_3_U1_RP_Bram_BramFirst_partial.bit

                        config_3_U2_RP_Count _CountCCW_partial.bit

config_4          RMs:  BramSecond, CountCW

                  Bits: config_4.bit (full bit file)

                        config_4_U1_RP_Bram_BramSecond_partial.bit

                        config_4_U2_RP_Count_CountCW_partial.bit
```

PlanAhead ソフトウェアでのコンフィギュレーションのインプリメンテーションは、「run」といいます。run はコンフィギュレーションごとに作成する必要があります。

PlanAhead では、プロジェクトを作成するとコンフィギュレーションが自動的に作成されます。このコンフィギュレーション用に設定されるリコンフィギャブル モジュールは、プロジェクトにリコンフィギャブル モジュールが追加された順序によって異なります。これは、各リコンフィギャブル パーティションに定義された最初のリコンフィギャブル モジュールがこのコンフィギュレーション (チュートリアルでは **BramFirst** と **CountCW**) 用に設定されるからです。

フル BIT ファイルでコンフィギュレーションされた FPGA には、コンフィギュレーションでインプリメントされたリコンフィギャブル モジュールが含まれます。フル BIT ファイルを読み込んだ後にスタティック ロジックのみが動作する必要のあるシステムの場合、リコンフィギャブル パーティションすべてに対してブラック ボックスを含めたコンフィギュレーションをインプリメントします。この場合、出力されるパーシャル BIT ファイルは効率的なように空白になります。

9-1 config_1 コンフィギュレーションをインプリメントします。

- 9-1-1 config_1 に設定されるリコンフィギャブル モジュールが **BramFirst** と **CountCW** であることを確認します。
- [Design Runs] ビュー ([**Window**] → [**Design Runs**]) で **config_1** を選択し、[Implementation Run Properties] ビューで [Partitions] タブをクリックします (図 20)。リストされる [Module Variants] が **BramFirst** と **CountCW** でない場合は、変更してください。

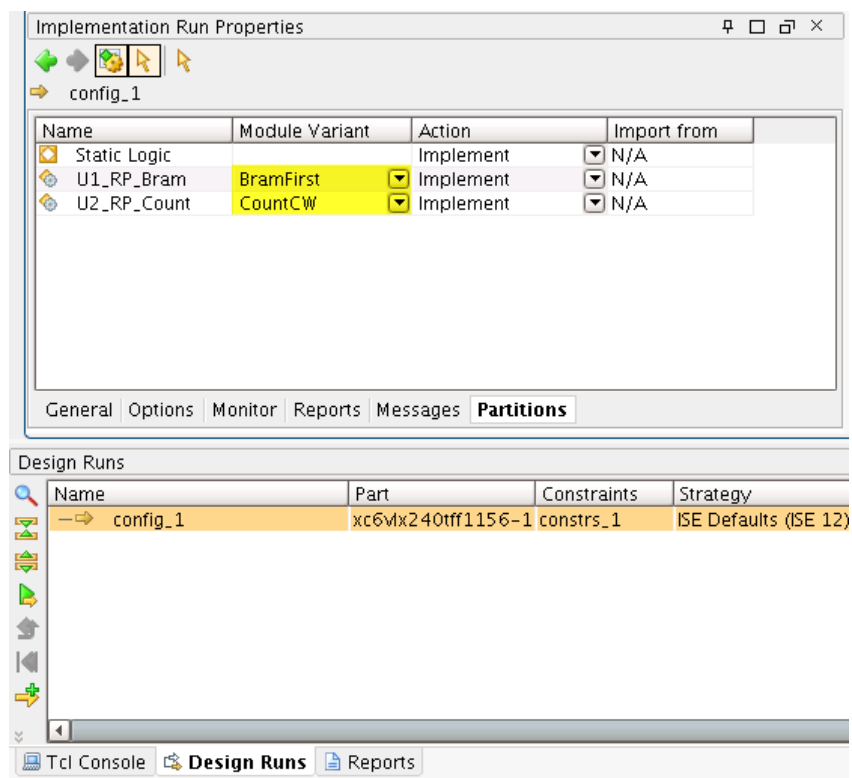


図 20 : コンフィギュレーション モジュール (config_1)

9-1-2 Flow Navigator で **[Implement]** をクリックし、インプリメンテーションを実行します。

[Design Runs] ビューの [Status] フィールドに NGDDBuild、MAP、PAR、TRCE の何かが実行されているかが表示されます。進捗状況は PlanAhead 右上のステータス バーにも表示されます。[Compilation Log] ビューにはさらに詳細が表示されます。

9-1-3 config_1 コンフィギュレーションをプロモートします。

config_1 は問題なくインプリメントできたので、次はプロモートできます。最初のコンフィギュレーションをプロモートしなくてもほかのコンフィギュレーションは実行できますが、コンフィギュレーション間のパーシャル BIT ファイル同士が競合してしまいます。複数のコンフィギュレーション間で互換性のあるパーシャル BIT ファイルを出力するには、プロモートするコンフィギュレーションを 1 つ選び、ほかのすべてのコンフィギュレーションはこのプロモートされたパーティションからスタティック ロジックをインポートする必要があります。これにより、すべてのコンフィギュレーション間で一貫したプロキシ ロジックが維持できます。

複数コンフィギュレーション間の互換性は、手順 11 の「パーシャル リコンフィギュレーション検証の実行」の方法に従って、パーシャル リコンフィギュレーション検証を使用してチェックできます。

9-1-4 Flow Navigator で **[Promote Partitions]** ボタンをクリックします (図 21)。

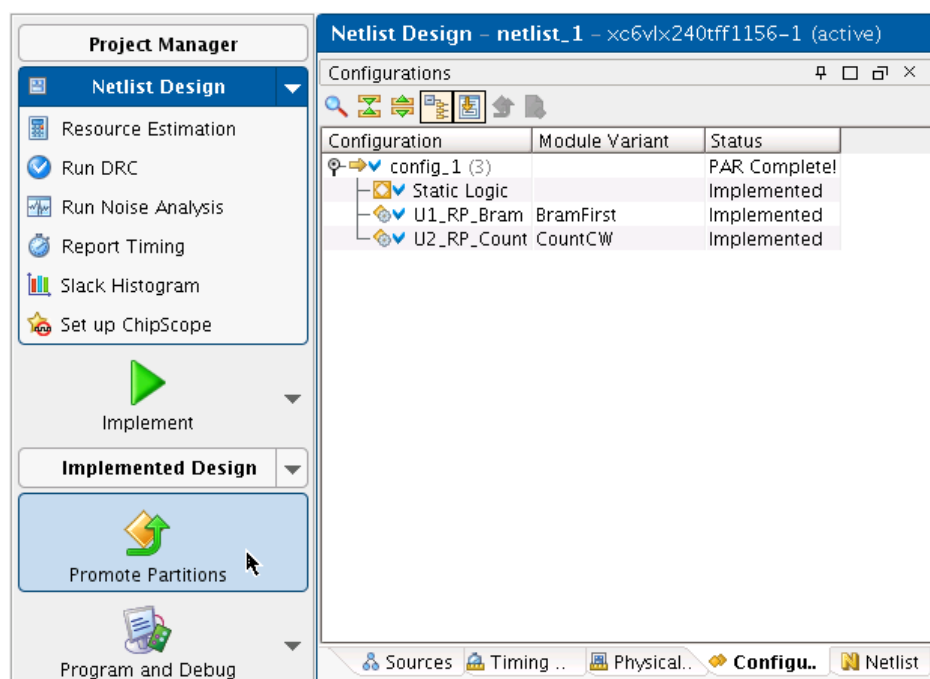


図 21 : [Promote Partitions] ボタン

- 9-1-5 [Promote Partitions] ダイアログ ボックスで [OK] をクリックします。
- 9-1-6 [Netlist Design] ビューの [Configurations] タブをクリックし、ステータスが [Implemented] から [Promoted] に変更されていることを確認してください (図 22)。

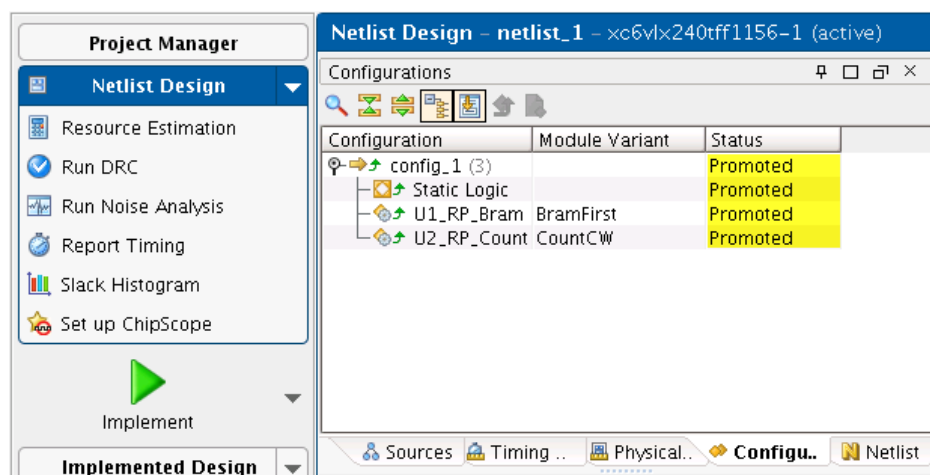


図 22 : [Promoted] ステータス

追加コンフィギュレーションの作成とインプリメント

手順 10

10-1 新しくコンフィギュレーションを作成します。

10-1-1 Flow Navigator の [Implement] ドロップダウン リストから **[Create Multiple Runs]** をクリックします (図 23)。

Create Multiple Runs ウィザードが開きます。

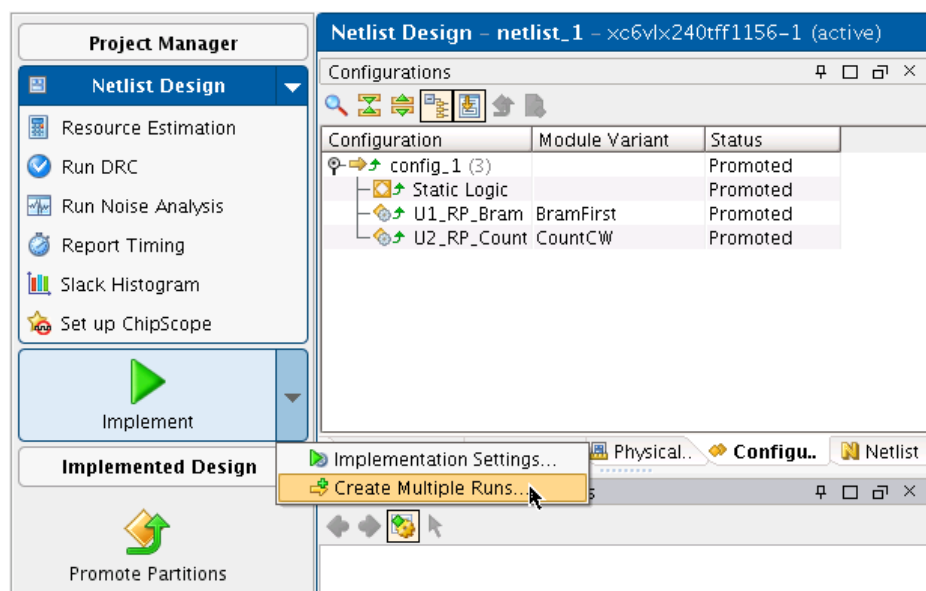


図 23 : Create Multiple Runs ウィザードの起動

10-1-2 新しくコンフィギュレーションを作成します。

10-1-3 最初のページで **[Next]** をクリックします。

10-1-4 **[Setup Implementation Run]** ページで **[Next]** をクリックします。

10-1-5 **[Choose Implementation Strategies and Reconfigurable Modules]** ページでは、インプリメンテーション戦略を選択し、コンフィギュレーションにどのリコンフィギャブル モジュールを使用するか定義して、複数コンフィギュレーションを作成できます。既に `config_2` という新しいコンフィギュレーションが表示されています。名前を付け替えることもできますが、このチュートリアルでは `config_2` をそのまま使用します。

10-1-6 **[Partition Action]** 列の参照ボタンをクリックし、**[Specify Partition]** ダイアログ ボックスを開きます (図 24)。

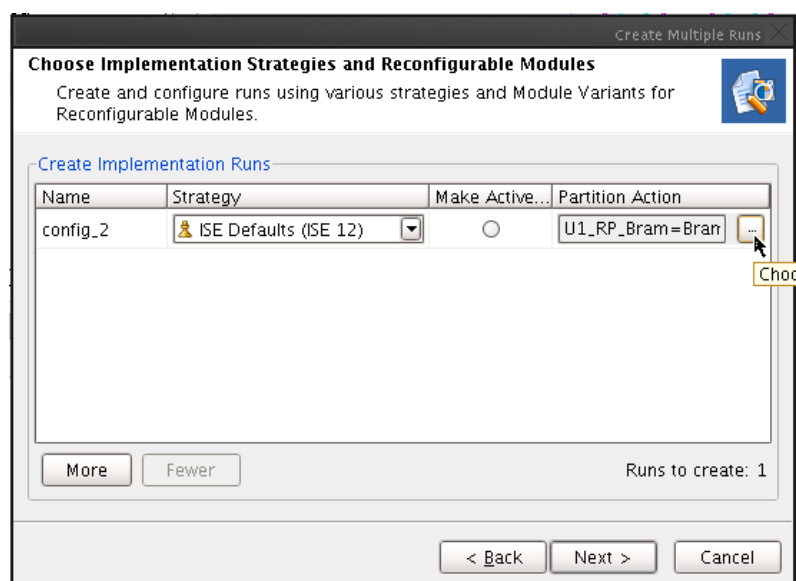


図 24 : [Choose Implementation Strategies and Reconfigurable Modules] ページ

10-1-7 デフォルトの [Module Variant] は、プロジェクトで現在アクティブになっているリコンフィギャブル モジュールによって異なります。この場合、BramFirst と CountCW がアクティブです。これらは既にインプリメントされインポートされているので、[Import] に設定され、インポート ディレクトリも設定されています。

10-1-8 BramSecond および CountCCW リコンフィギャブル モジュールをインプリメントするコンフィギュレーションを作成するため、[Module Variant] 列をこれらのリコンフィギャブル モジュールと同じように変更します (図 25)。

これらのリコンフィギャブル モジュールはまだインプリメント (またはプロモート) されていないので、[Action] フィールドは [Implement] に変更します。

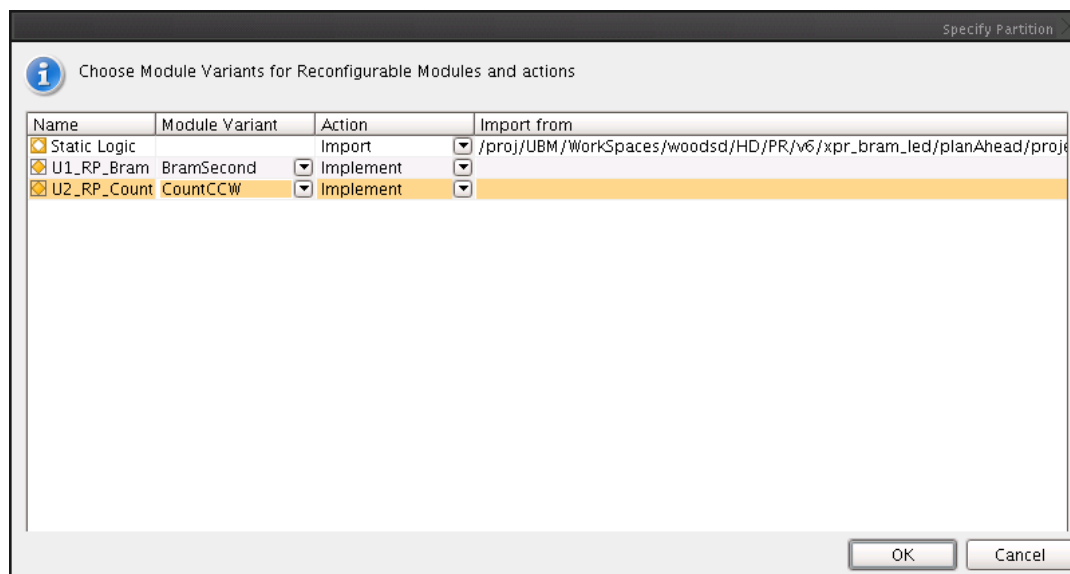


図 25 : [Specify Partition] ダイアログ ボックス (BramSecond/CountCCW)

10-1-9 [Specify Partition] ダイアログ ボックスで **[OK]** をクリックします。

10-1-10 **オプション** : ブラック ボックス モジュールを作成するオプションの手順を実行した場合、それらのモジュールをインプリメントするコンフィギュレーションを追加で作成します。[Choose Implementation Strategies and Reconfigurable Modules] ページで **[More]** ボタンをクリックし、**config_3** を追加します。[Partition Action] 列の参照ボタンをクリックし、[Module Variants] を **BramBB** と **CountBB** に設定します。

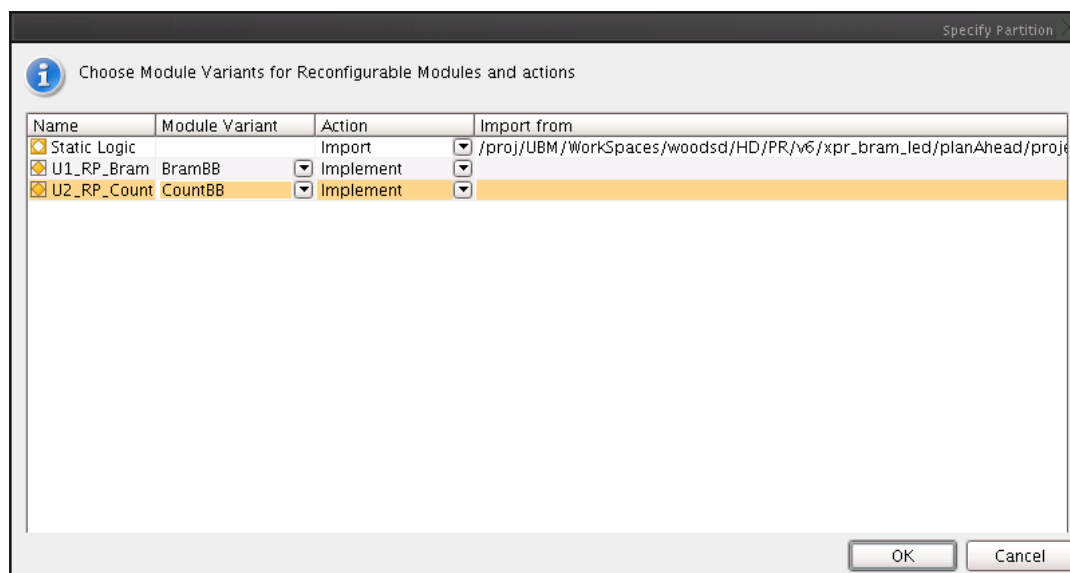


図 26 : [Specify Partition] ダイアログ ボックス (BramBB/CountBB)

10-1-11 [Choose Implementation Strategies and Reconfigurable Modules] ページで **[Next]** をクリックします。

10-1-12 [Launch Options] ページで **[Do not launch now]** をオンにし、**[Next]** をクリックします。コンフィギュレーションはここから起動することもできますが、ここでは新しいコンフィギュレーションを次に続く手順で作成していきます。

10-1-13 [Create Multiple Runs Summary] ページで **[Finish]** をクリックしウィザードを終了します。

10-1-14 [Design Runs] ビューにウィザードで作成した新しいコンフィギュレーションが表示されます。[Design Runs] ビューで新しいコンフィギュレーションを選択し、[Implementation Run Properties] ビューの [Partitions] タブをクリックして [Module Variants] および [Action] 列を確認します。[Static Logic] が [Import] に設定されているので、config_1 からのプロモートされた結果がインポートされます。

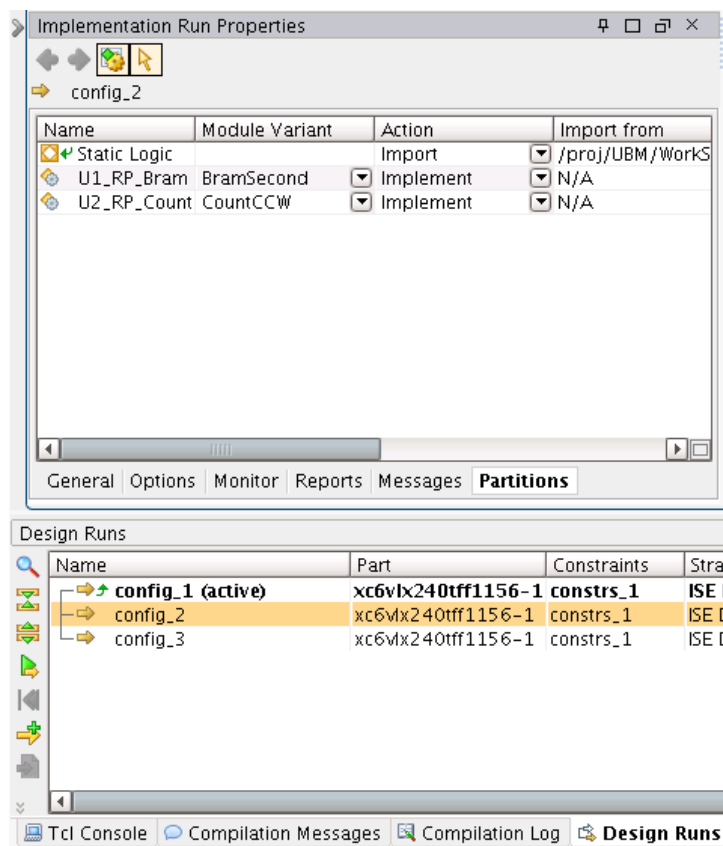


図 27 : コンフィギュレーション設定の確認

10-1-15 コンフィギュレーションを右クリックし、**[Launch Runs]** を選択します。新しいコンフィギュレーションが複数ある場合、config_1 の場合とは違って結果が互いに依存しないので、それらを一緒に (使用可能な場合は複数プロセッサで) 起動することもできます (図 28)。

10-1-16 [Launch Selected Runs] ダイアログ ボックスで **[Launch runs on Local Host]** をオンにし、適切なジョブ数 (使用するプロセッサ数) を選択し、**[OK]** をクリックします。

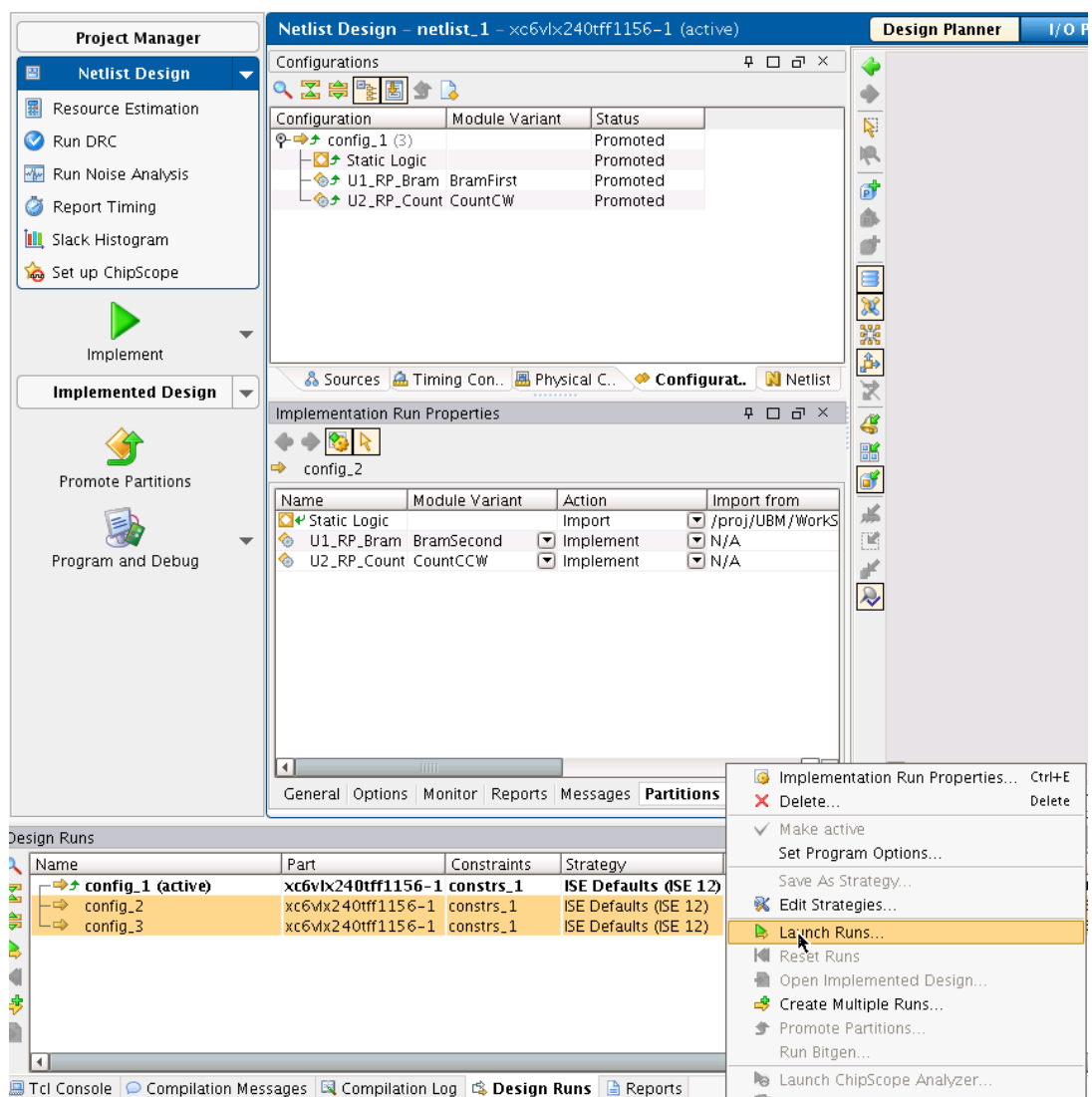


図 28 : [Launch Selected Runs] ダイアログ ボックス

パーシャルリコンフィギュレーション検証の実行

手順 11

複数のコンフィギュレーションがインプリメントされたら、それらを比較して、スタティック ロジックおよびパーティション ピンがすべてのコンフィギュレーションで一貫しているかどうかを検証してください。これは、BIT ファイルに互換性があるかどうかを確認するために必要なチェックです。

11-1 すべてのコンフィギュレーションでパーシャル リコンフィギュレーション検証を実行します。

11-1-1 Flow Navigator の [Program and Debug] ドロップダウンリストから **[Verify Configuration]** をクリックします (図 29)。

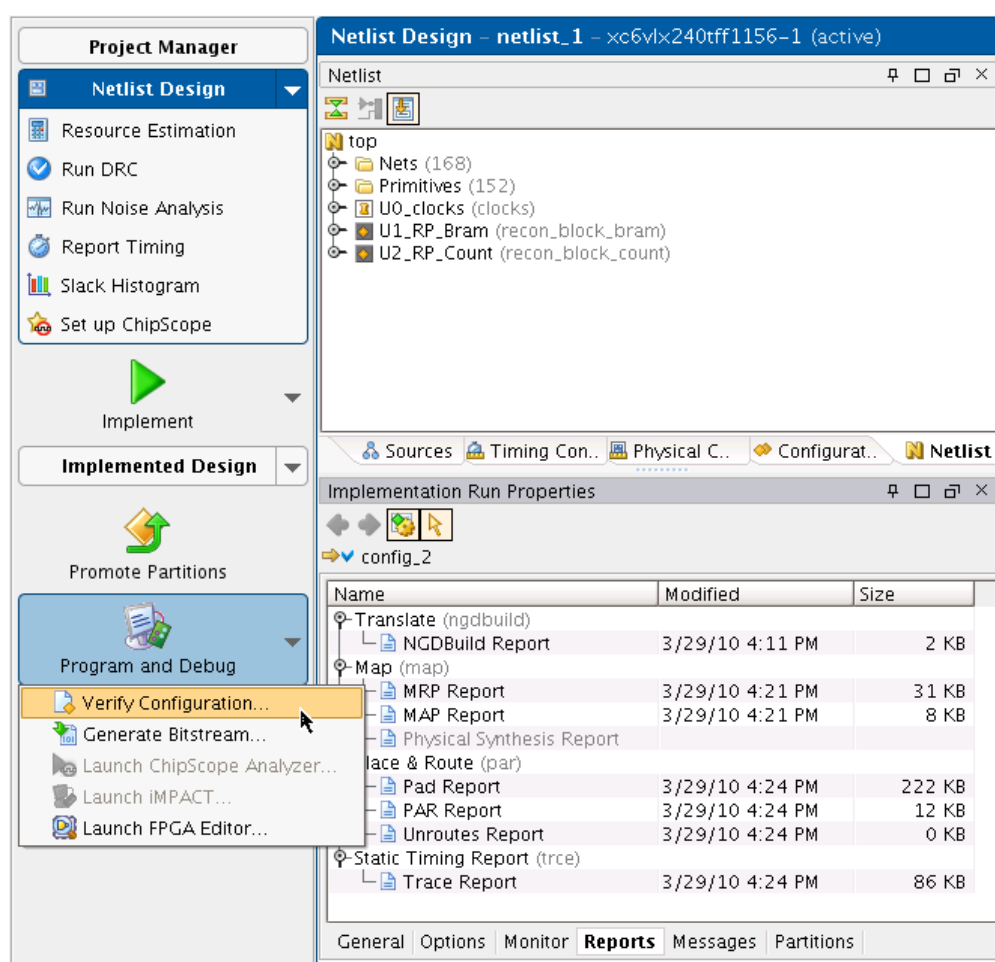


図 29 : コンフィギュレーション設定の確認

11-1-2 少なくとも 2 つコンフィギュレーションを選択し (すべて選択してもかまいません)、**[OK]** をクリックし、パーシャルリコンフィギュレーション検証を実行します。

検証チェックがパスしたら、エラーがなかったことを示すメッセージが表示されます。これは、BIT ファイルを生成できることを意味します。[OK] をクリックしてメッセージ ボックスを閉じます。

詳細なレポートが PlanAhead で開き、xpr_bram_led/PlanAhead/project_1/project_1.runs/pr_verify.log に保存されます。

ビットファイルの生成とダウンロード

手順 12

コンフィギュレーションごとに複数の BIT ファイルが生成されます。生成されるのは、電源投入時に FPGA をプログラムするのに使用されるフル BIT ファイルが 1 つ、特定コンフィギュレーションに関連した個別モジュールのロジックを含むリコンフィギュラブル パーティションそれぞれに対してパーシャル BIT ファイルが 1 つずつです。このチュートリアルでは、BramFirst、CountCW、BramSecond、CountCCW (オプションで BramBB および CountBB) を使用してコンフィギュレーションを作成しました。各コンフィギュレーションの BIT ファイルを生成することで、これらすべてのリコンフィギュラブル モジュールのパーシャル BIT ファイルが生成されました。これらのパーシャル BIT ファイルはいずれも、デバイスを最初にコンフィギュレーションするためのフル BIT ファイルに関係なく、関連するパーシャル リコンフィギュレーション領域をリコンフィギュレーションするために使用できます。

12-1 すべてのコンフィギュレーションの BIT ファイルを生成します。

12-1-1 [Design Runs] ビューですべてのコンフィギュレーションを選択します。ビューで右クリックし、[Run Bitgen] をクリックします (図 30)。

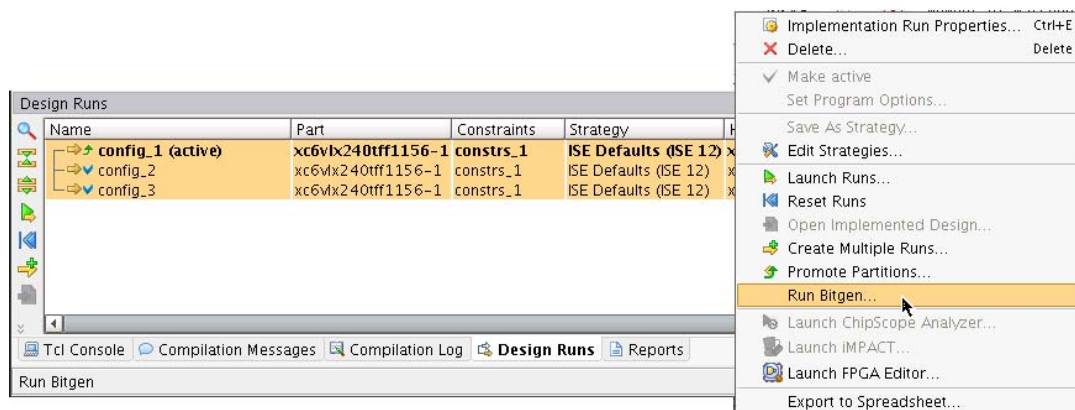


図 30 : BitGen の実行

12-1-2 パーシャル BIT ファイルを生成するために特に設定する必要のある BitGen オプションはありませんので、ダイアログ ボックスで [OK] をクリックし、BitGen を起動します。

パーシャル BIT ファイルはフル BIT ファイルと同じように FPGA にダウンロードされます。ラボ環境では、BIT ファイルのダウンロード、検証、デバッグ目的に iMPACT ソフトウェア ツールを使用できます。

12-1-3 USB ダウンロード ケーブルを ML605 と PC に接続します。

12-1-4 iMPACT をスタンドアロン モードで起動します。

12-1-5 Flow Navigator の [Program and Debug] ドロップダウンリストから **[Launch iMPACT]** をクリックします (図 31)。

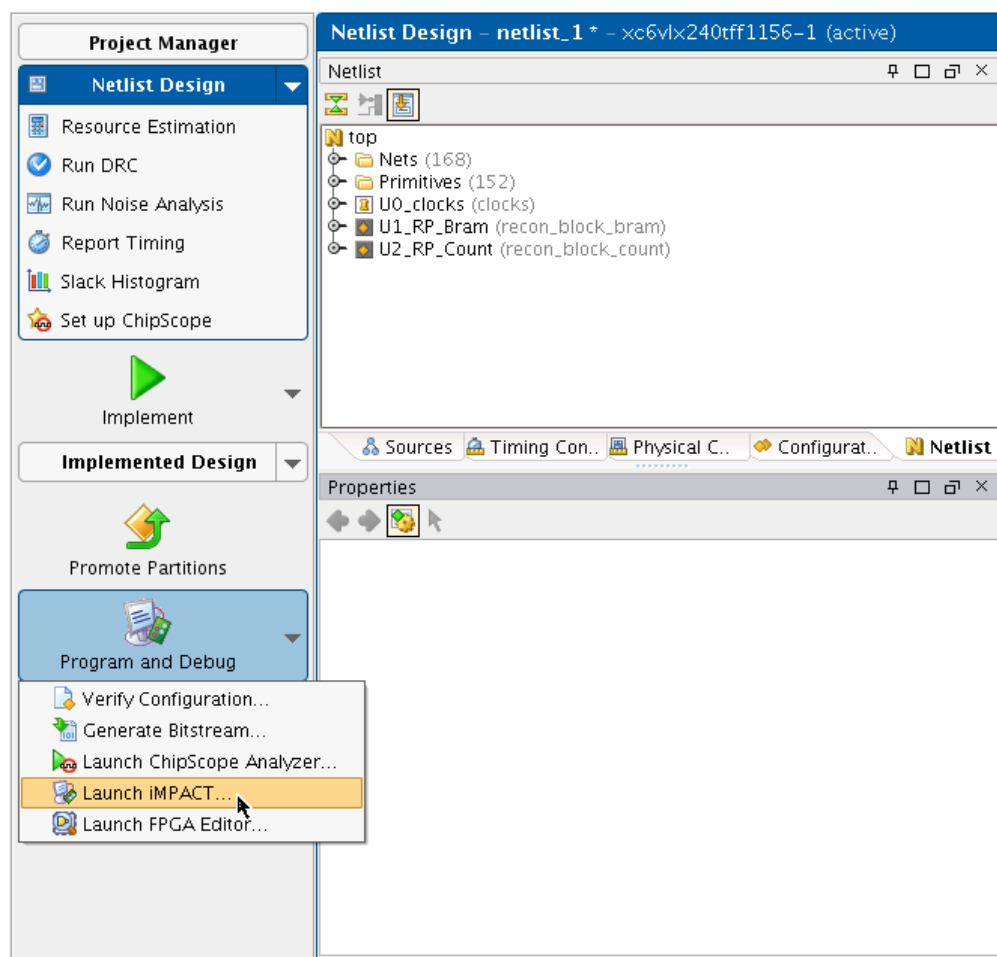



図 31 : iMPACT の起動

12-1-6 iMPACT のウィンドウで **[Boundary Scan]** をダブルクリックし、**[Initialize Chain]** ボタン ( **Initialize Chain**) をクリックします。

12-1-7 チェーンが問題なく認識されたら、xc6vlx240t デバイスを右クリックし、フル BIT ファイルの `xpr_bram_led/PlanAhead/<project_name>/<project_name>.runs/config_1/config_1.bit` を指定します。

12-1-8 xc6vlx240t を再び右クリックし、**[Program]** をクリックします (図 32)。

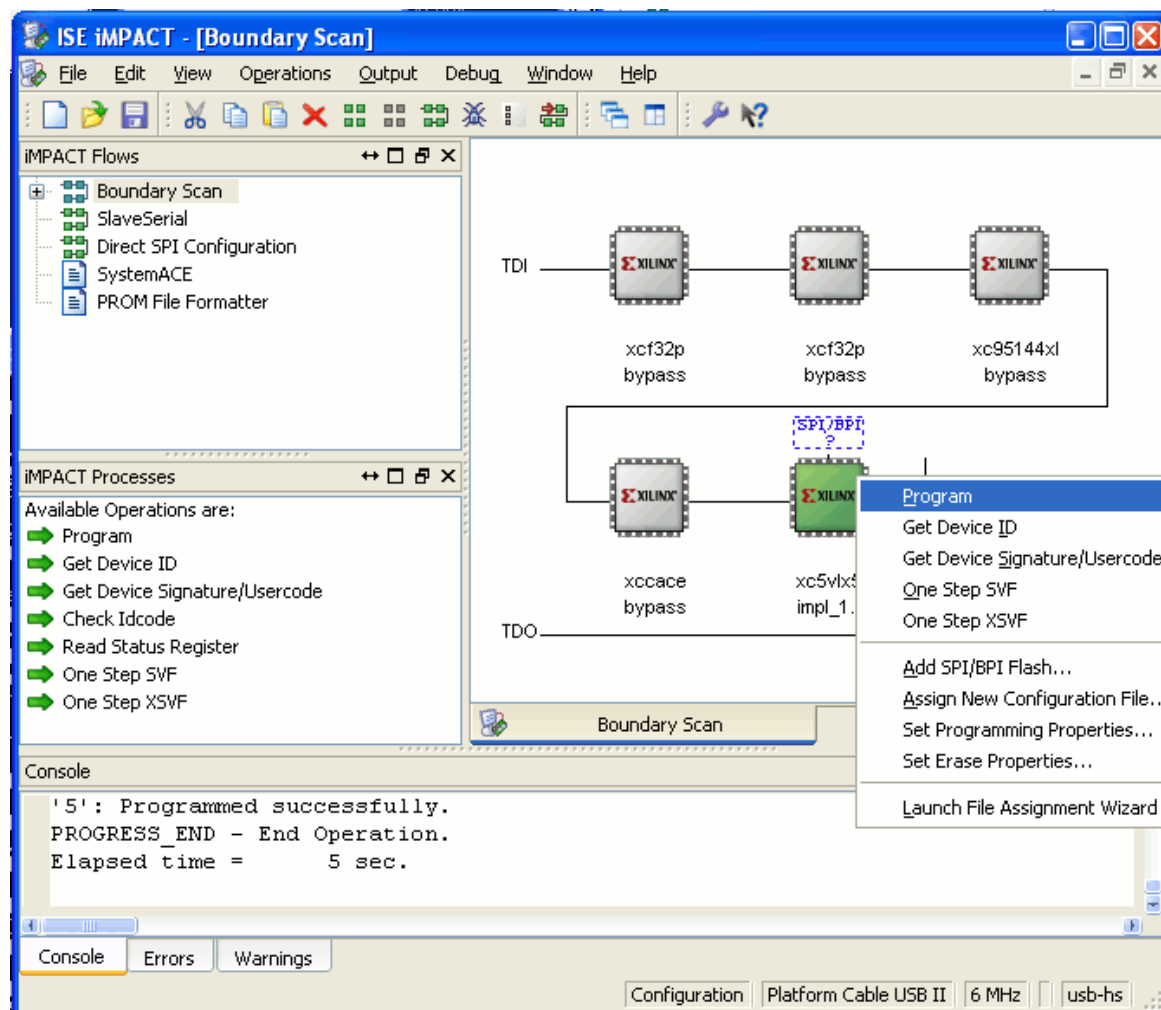


図 32 : iMPACT を使用したデバイスのコンフィギュレーション

ML605 の FPGA がフル BIT ファイルを使用してプログラムされました。コンフィギュレーションには数秒かかります。

12-2 パーシャル BIT ファイルを関連付けます。

12-2-1 xc6vlx240t デバイスを右クリックし、パーシャル BIT ファイル

xpr_bram_led/PlanAhead/<project_name>/<project_name>.runs/config_2/config_2_U1_RP_Bram_BramSecond_partial.bit を指定します。

12-2-2 xc6vlx240t デバイスを右クリックし、[Program] をクリックします。

パーシャル BIT ファイルは非常に小さいので、FPGA のパーシャル リコンフィギュレーションはすぐに終了します。

まとめ

このチュートリアルでは、PlanAhead のパーシャル リコンフィギュレーション プロジェクトを作成しました。この後、リコンフィギュラブル パーティションを 2 つ作成し、それぞれに複数のリコンフィギュラブル モジュールを関連付けました。AREA_GROUP 制約を使用して各リコンフィギュラブル パーティションをデバイスのエリアに制約付け、グローバル タイミング制約を作成してデザイン全体に制約を付けました。最初のコンフィギュレーションをインプリメントしてプロモートし、そのコンフィギュレーションからスタティック ロジックをインポートしてその他のコンフィギュレーションを追加しました。パーシャル リコンフィギュレーション検証を実行し、すべてのコンフィギュレーションに競合がないかどうか確認し、BIT ファイルを生成しました。最後に BramFirst および CountCW モジュールを含む ML605 ボードにフル BIT ファイルをダウンロードし、BramSecond モジュールのパーシャル BIT ファイル3を使用して U1_RP_Bram RP をリコンフィギュレーションしました。