

パーシャル リコンフィギュレーション チュートリアル

PlanAhead デザイン ツール

UG743 (v 13.4) 2012 年 1 月 3 日





Xilinx is disclosing this user guide, manual, release note, and/or specification (the “Documentation”) to you solely for use in the development of designs to operate with Xilinx hardware devices. You might not reproduce, distribute, republish, download, display, post, or transmit the Documentation 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. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that might be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© Copyright 2010–2011 Xilinx Inc. All Rights Reserved. XILINX, the Xilinx logo, the Brand Window and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners. The PowerPC name and logo are registered trademarks of IBM Corp., and used under license. All other trademarks are the property of their respective owners.

本資料は英語版 (v13.4) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

日本語版は参考用としてご使用の上、最新情報につきましては、必ず最新英語版をご参照ください。

この資料に関するフィードバックおよびリンクなどの問題につきましては、jpn_trans_feedback@xilinx.com までお知らせください。いただきましたご意見を参考に早急に対応させていただきます。なお、このメール アドレスへのお問い合わせは受け付けておりません。あらかじめご了承ください。

目次

ソフトウェア要件.....	5
ハードウェア要件.....	5
チュートリアル デザイン ファイル.....	6
チュートリアル デザインの説明.....	6
チュートリアル ソフトウェアの概要.....	6
ソフトウェア ツール フロー.....	7
プロジェクト ディレクトリおよび HDL デザイン構造.....	8
手順 1 : HDL ソースからのネットリストの合成 (オプション).....	9
手順 2 : プロジェクトの作成.....	10
手順 3 : 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.....	12
手順 4 : リコンフィギャブル モジュールの追加.....	15
手順 5 : ブラック ボックス モジュールの追加 (オプション).....	18
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.....	20
手順 7 : パーティション ピンおよびリコンフィギャブル パーティション インターフェイスのタイミング.....	26
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 チェックの実行.....	28
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.....	30
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.....	34
手順 11 : コンフィギュレーションの検証.....	39
手順 12 : BIT ファイルの生成とダウンロード.....	40
まとめ.....	43

パーシャル リコンフィギュレーション チュートリアル

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 (PR) デザインを作成します。PlanAhead™ ツール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 ツールを使用します。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 などのツールも使用できますが、このチュートリアルでは説明しません。

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の知識があり、ザイリンクス ソフトウェアを使用して FPGA デザインをインプリメンテーションした経験のある方を対象にしています。

このチュートリアルでは PlanAhead ツールの機能の一部のみを紹介しており、それ以外の機能については、別の PlanAhead チュートリアルで説明します。

このチュートリアルを終了すると、PlanAhead ツール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。特に次の内容を学ぶことができます。

- リコンフィギャブル パーティション (RP) の作成
- リコンフィギャブル モジュール (RM) の追加
- リコンフィギャブル パーティションの Pblock の範囲定義
- パーシャル リコンフィギュレーション用の DRC チェックの実行
- コンフィギュレーションの作成およびコンフィギュレーション
- コンフィギュレーションの検証
- ビット ファイルの生成

ソフトウェア要件

PlanAhead ツールは、ISE® Design Suite 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 ツールが起動できるか、チュートリアル 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。

インストール手順については、http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_4/iil.pdf にある『ISE Design Suite : インストールおよびライセンス ガイド』(UG798) を参照してください。

ハードウェア要件

大型デバイスで PlanAhead ツールを使用するには、2GB 以上の RAM が推奨されます。1GB でも十分ですが、パフォーマンスに影響することがあります。

チュートリアル デザイン ファイル

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、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。リファレンス デザインのコピーは、http://japan.xilinx.com/support/documentation/dt_planahead_planahead13-4_tutorials.htm からダウンロードできます。

解凍されたデータ ディレクトリは、このチュートリアルでは <Extract_Dir> と表記しています。

チュートリアルのサンプル デザイン データは、チュートリアル実行中に変更されます。各チュートリアルを実行する前に、まず元の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 ファイルが含まれています。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、実行ディレクトリには必要なデータのみが残されています。

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 2 つのリコンフィギャブル パーティションが含まれています。

- そのうち 1 つのリコンフィギャブル パーティションには、エンベデッドされたブロック RAM が含まれています。このブロック RAM モジュールを別のブロック RAM のデータでリコンフィギュレーションすると、8 つの GPIO LED のシーケンスが変わります。
- もう 1 つのリコンフィギャブル パーティションには、エンベデッドされた I/O バッファーが含まれています。異なるステート マシンの遷移でこの 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または反時計回りに変わります。

チュートリアル ソフトウェアの概要

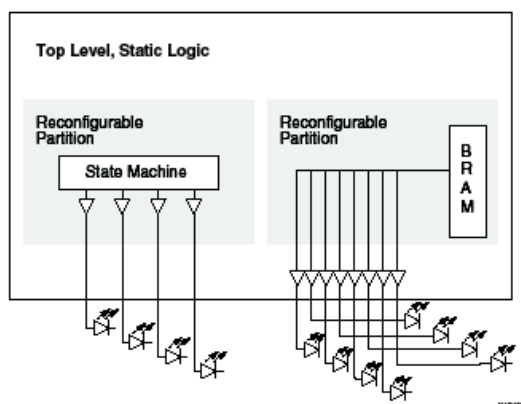


図 1 : デザインの概要 : パーシャル リコンフィギュレーション

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。このチュートリアルでは、XST (Xilinx Synthesis Technology) 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 ツールを使用します。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することができます。

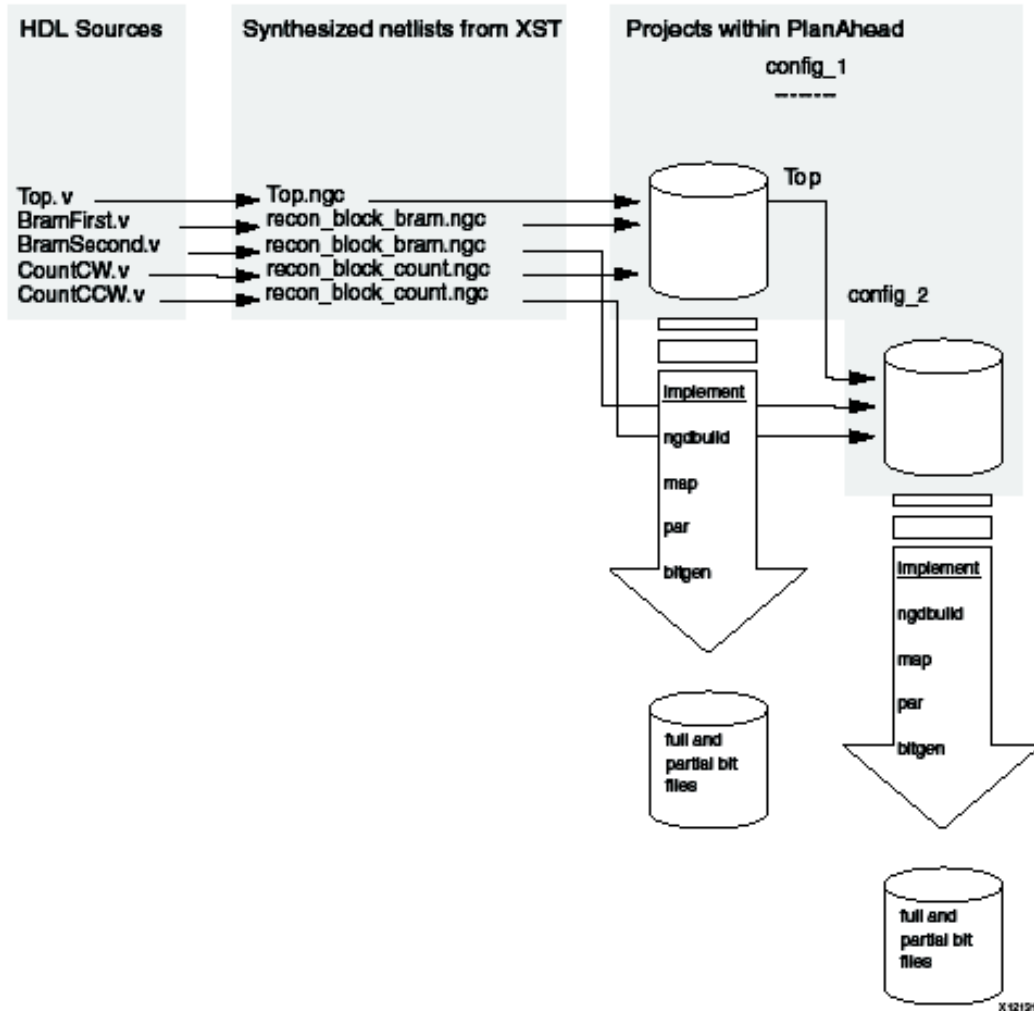


図 2 : ソフトウェア フローの概要

プロジェクト ディレクトリおよび HDL デザイン 構造

パーシャル リコンフィギャブルの FPGA デザインを正しく構築し合成するには、ブラック ボックスおよびボトムアップの合成アプローチが必要です。各リコンフィギャブル モジュールは個別のプロジェクトとして合成され、それぞれのプロジェクトでネットリストが生成されます。最上位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化するため、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ネットリストには含まれません。

チュートリアル デザインのディレクトリ構造は、次のようになっています。

〈Extract_Dir〉	
- 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)
- Synth	-> PlanAhead プロジェクトおよび結果
- 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、最上位モジュール用のディレクトリは別にありません。個々のリコンフィギャブル モジュールは /Source および /Implementation ディレクトリに含まれます。

```
Implementation
Top (最上位および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 (カウンターの時計回りバージョン)
CountCCW (カウンターの反時計回りバージョン)
```

最上位ソース ファイルを開きます。

```
〈Extract_Dir〉/Source/Top/Top.v
```

このデザインに含まれている 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。これらのブラック ボックスの基本 HDL 記述は提供されていません。

手順 1 : HDL ソースからのネットリストの合成 (オプション)

PlanAhead ツールでは、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ていないので、PlanAhead プロジェクトを作成する前に XST を使用してデザインを合成する必要があります。このチュートリアルで使用するファイルでは、XST が既に実行されています。必要に応じて NGC ファイルを使用できます。NGC ファイルを使用する場合は、手順 2 に進んでください。

XST プロジェクト ファイルは、次のディレクトリにあります。

- <Extract_Dir>/Synth/Top/Top.xst
- <Extract_Dir>/Synth/Top/Top.prj

Top.xst では、デフォルトで I/O バッファが自動挿入されるようになっています。すべてのリコンフィギュラブル モジュールの XST プロジェクト ファイルでこの属性を NO に設定する必要があります。

```
-iobuf NO
```

下位モジュールには特別な状況を除き I/O バッファを挿入することはできません。この例外は、U2_RP_Count リコンフィギュラブル パーティションで説明します。

すべてのモジュールを合成するため Tcl スクリプトを実行

すべてのモジュールを合成するには、<Extract_Dir> 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 が呼び出されソース ディレクトリの Verilog ファイルが合成されます。

- <Extract_Dir>/Source/Bram* のモジュールの場合は、リコンフィギュラブル モジュールにブロック RAM が含まれます。
- <Extract_Dir>/Source/Count* のモジュールの場合は、リコンフィギュラブル モジュールに 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 <Extract_Dir>/Synth/<module> ディレクトリに保存されています。

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、<Extract_Dir>/Synth/<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。

手順 2 : プロジェクトの作成

新規プロジェクトを作成するには、次の手順に従います。

1. PlanAhead ツールを起動します。
 - Windows の場合は、[スタート] → [プログラム] → [Xilinx ISE Design Suite 13] → [PlanAhead] → [PlanAhead] をクリックします。
 - Linux の場合は、<Extract_Dir>/planAhead ディレクトリに移動し、「planAhead」と入力します。
2. Getting Started ページの [Create New Project] というリンクをクリックします。
3. [Create a New PlanAhead Project] ページで [Next] をクリックします。
4. [Project Name] ページでプロジェクト名を設定し、プロジェクトのディレクトリを次のように入力します。
<Extract_Dir>/planAhead
5. [Next] をクリックします。
6. [Design Source] ページで [Specify synthesized (EDIF or NGC) netlist] をオンにします。
7. [Enable Partial Reconfiguration] チェックボックスをオンにします。
8. [Next] をクリックします。

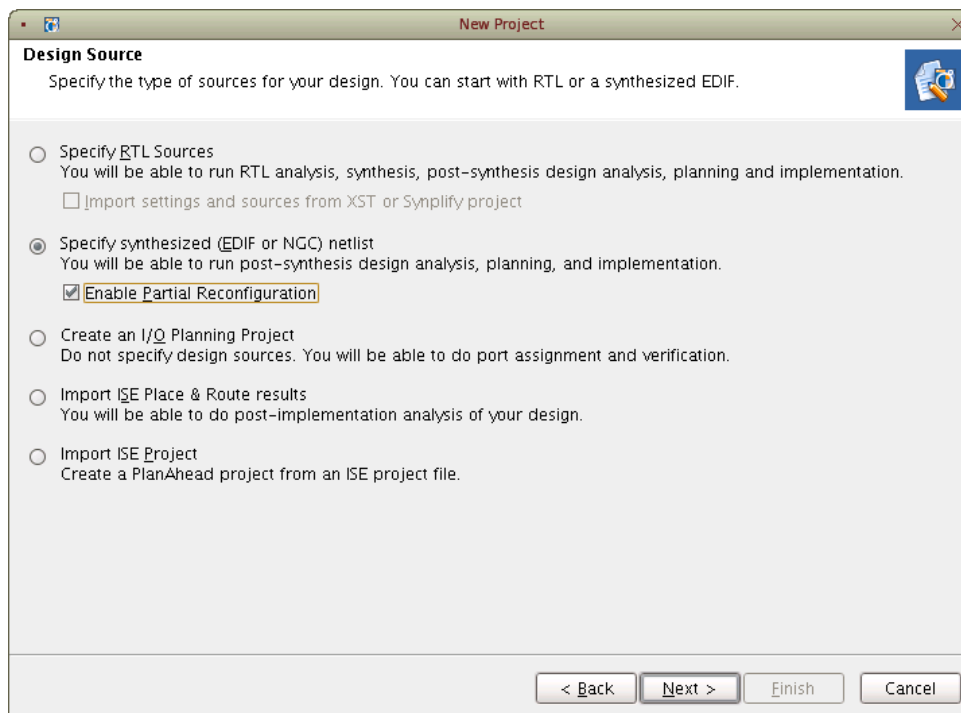


図 3 : デザイン ソースの指定

10. [Add Files] ボタンをクリックし、次の最上位ネットリストを選択します。
<Extract_Dir>/Synth/Top/Top.ngc
11. [OK] をクリックしてプロジェクトにこのファイルを追加します。[Next] をクリックします。

最上位 (またはスタティック) ロジックに関連しているファイルのみがこの時点では追加されます。下位リコンフィギュラブル モジュールのネットリストは後で追加されます。

12. [Next] をクリックします。
13. [Constraint Files] ページで [Add Files] をクリックします。
14. 次のユーザー制約ファイル (UCF) を選択します。

<Extract_Dir>/Source/UCF/top_ml605.ucf

15. [OK] をクリックします。
16. [Next] をクリックします。
- [Default Part] ページでは、ネットリストがスキャンされ、該当デバイスが自動的に選択されます。
17. xc6vlx240tff1156-1 デバイスが選択されていることを確認します。
18. [Next] をクリックします。
19. [New Project Summary] ページでプロジェクト設定を確認します。

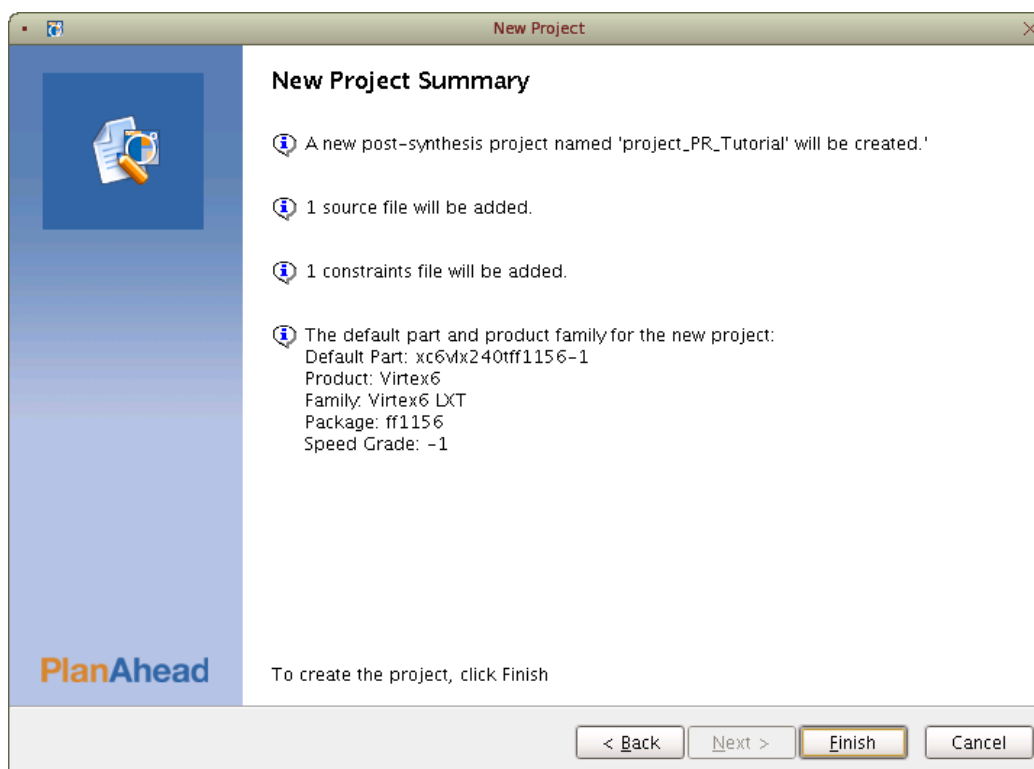


図 4 : 新規プロジェクトのサマリ

20. [Finish] をクリックします。

手順 3 : 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

ここでは次の作業を行います。

- U1_RP_Bram のリコンフィギャブル パーティションを作成
- U2_RP_Count のリコンフィギャブル パーティションを作成

U1_RP_Bram のリコンフィギャブル パーティションを作成

このチュートリアルで説明されている機能の多くは、[Netlist Design] ビューが開いている場合にのみ使用可能です。[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直した場合は、Flow Navigator の [Netlist Design] ボタンをクリックして [Netlist Design] ビューを開きます。

U1_RP_Bram のリコンフィギャブル パーティションを作成するには、次の手順に従います。

1. Flow Navigator で [Netlist Design] をクリックします。

これでネットリストがメモリに読み込まれます。リコンフィギャブル モジュールにネットリストをまだ割り当てていないため、未定義のインスタンスに関するメッセージがいくつか表示されます。

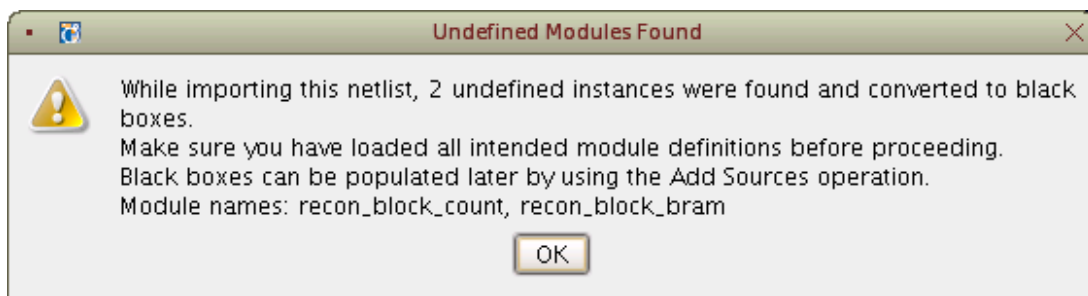


図 5 : 未定義モジュールに関する警告メッセージ

2. ブラック ボックス モジュールに関するメッセージすべてに対して [OK] をクリックします。
3. [Netlist] ビューで U1_RP_Bram をクリックします。
4. 右クリックします。
5. [Set Partition] をクリックします。

これで Set Partition ウィザードが起動します。

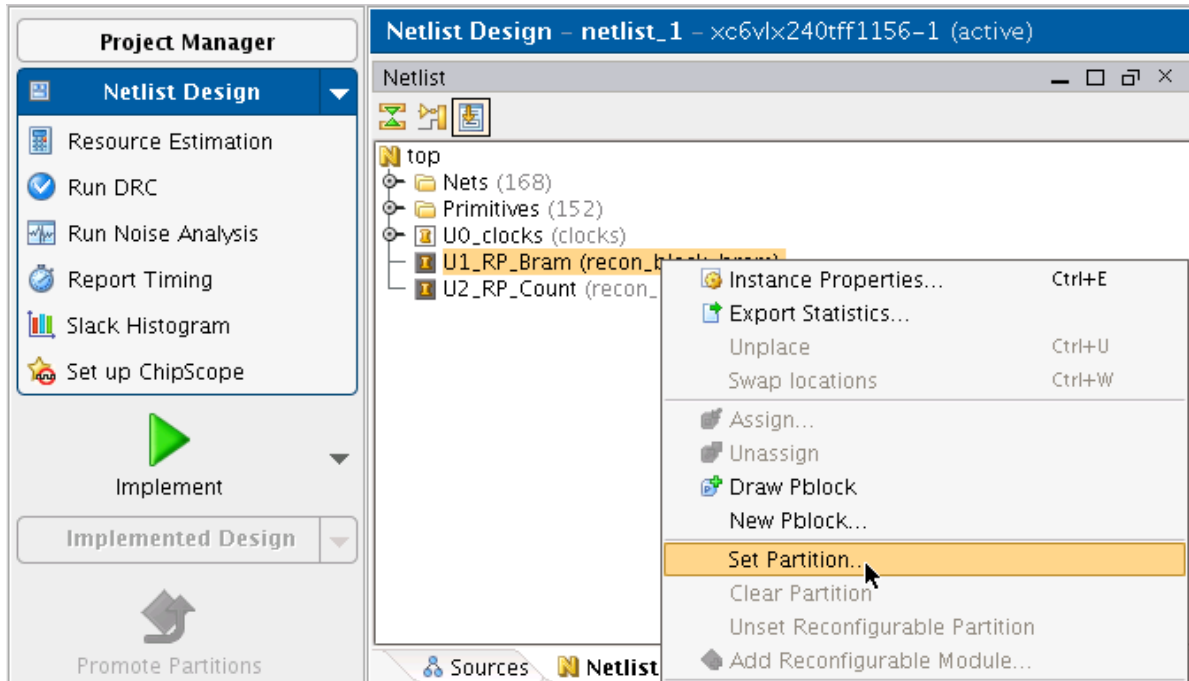


図 6 : パーティションとリコンフィギャブル モジュールの設定

6. [Next] をクリックします。
7. 選択されていない場合は [is a reconfigurable Partition] をオンにします。
8. [Next] をクリックします。
9. リコンフィギャブル モジュール名を「BramFirst」と入力します。
10. 選択されていない場合は [Netlist already available for this Reconfigurable Module] をオンにします。
11. [Next] をクリックします。
12. [Top Netlist File] に次のファイルを選択します。
 <Extract_Dir>/Synth/BramFirst/recon_block_bram.ngc
13. [OK] をクリックします。
14. [Next] をクリックします。
 リコンフィギャブル モジュールに下位ネットリストが関連付けられていない場合は、ここでネットリスト ディレクトリを追加することができます。この場合はありません。
15. [Next] をクリックしてオプションの制約ファイルの画面をスキップします。
 ここでは、モジュール レベルの制約ファイルを追加できます。このチュートリアルにはありません。
16. [Set Partition Summary] を確認します。
17. [Finish] をクリックします。

U1_RP_Bram のリコンフィギャブル パーティションがこれで作成されました。次の図にあるようにこのインスタンスの下にはリコンフィギャブル モジュールが 1 つリストされています。このインスタンスの Pblock も作成され、[Physical Constraints] ビューで確認できます。

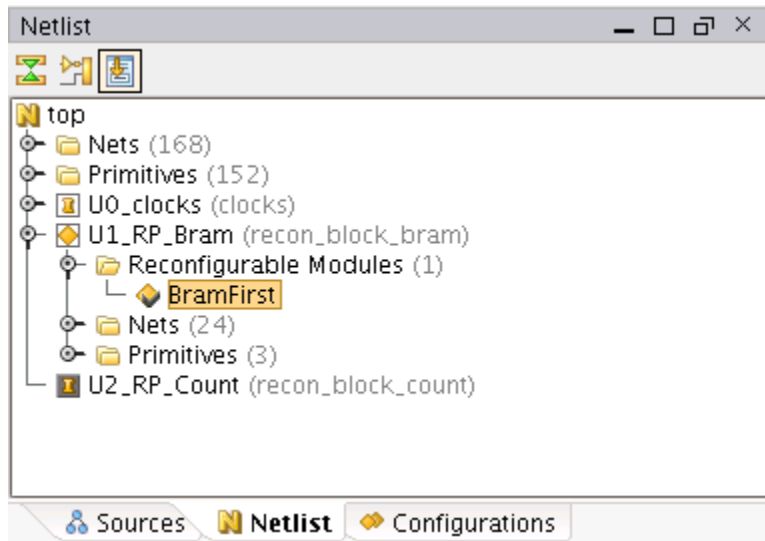


図 7 : BramFirst リコンフィギャブル モジュール

U2_RP_Count のリコンフィギャブル パーティションを作成

U2_RP_Count のリコンフィギャブル パーティションを作成するには、次の手順に従います。

1. [Netlist] ビューで U2_RP_Count をクリックします。
2. 右クリックします。
3. [Set Partition] をクリックします。
4. 選択されていない場合は [is a reconfigurable Partition] をオンにしてから [Next] をクリックします。
5. リコンフィギャブル モジュール名を「CountCW」と入力します。
6. 選択されていない場合は [Netlist already available for this Reconfigurable Module] をオンしてから [Next] をクリックします。
7. [Top Netlist File] で次のファイルを指定します。
<Extract_Dir>/Synth/CountCW/recon_block_count.ngc- 8. サマリ ページが表示されるまで [Next] をクリックします。
- 9. [Finish] をクリックしてウィザードを終了します。

これで各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 モジュールが 1 つずつ作成されました。手順 4 では、リコンフィギャブル パーティションのリコンフィギャブル モジュール追加方法について説明します。

手順 4 : リコンフィギャブル モジュールの追加

ここでは次の作業を行います。

- リコンフィギャブル モジュールを U1_RP_Bram へ追加
- リコンフィギャブル モジュールを U2_RP_Count へ追加

新しいリコンフィギャブル モジュールを追加する手順は、リコンフィギャブル パーティションの作成とそのリコンフィギャブル モジュールを追加する手順とほぼ同じです。

リコンフィギャブル モジュールを U1_RP_Bram へ追加

リコンフィギャブル モジュールを U1_RP_Bram へ追加するには、次の手順に従います。

1. [Netlist] ビューに移動します。
2. U1_RP_Bram をクリックします。
3. 右クリックします。
4. [Add Reconfigurable Module] をクリックします。

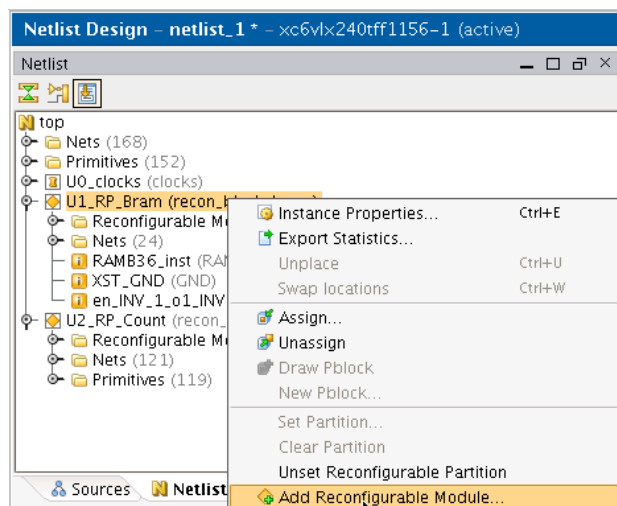


図 8 : リコンフィギャブル モジュールの追加

5. Add Reconfigurable Module ウィザードで [Next] をクリックします。
6. [Reconfigurable Module Name] に「BramSecond」と入力します。
7. [Next] をクリックします。
7. [Top Netlist File] を次のように設定します。
<Extract_Dir>/Synth/BramSecond/recon_block_bram.ngc
8. サマリ ページが表示されるまで [Next] をクリックします。
9. [Finish] をクリックしてウィザードを終了します。

リコンフィギャブル モジュールを U2_RP_Count へ追加

リコンフィギャブル モジュールを U2_RP_Count へ追加するには、次の手順に従います。

1. U2_RP_Count をクリックします。
2. 右クリックします。
3. [Add Reconfigurable Module] をクリックします。
4. [Reconfigurable Module Name] に「CountCCW」と入力します。
5. [Top Netlist File] を次のように設定します。

<Extract_Dir>/Synth/CountCCW/recon_block_count.ngc

6. ウィザードを終了します。

これで、各リコンフィギャブル パーティションに 1 つずつリコンフィギャブル モジュールが追加されました。[Netlist] ビューは、次の図のようになっているはずですが。

現在アクティブなリコンフィギャブル モジュールのネットおよびプリミティブは、各リコンフィギャブル パーティションの下にリストされています。これらには、黄色い菱形アイコンにチェック マークが付いています。

次の図では、BramFirst および CountCW がアクティブ モジュールになっています。

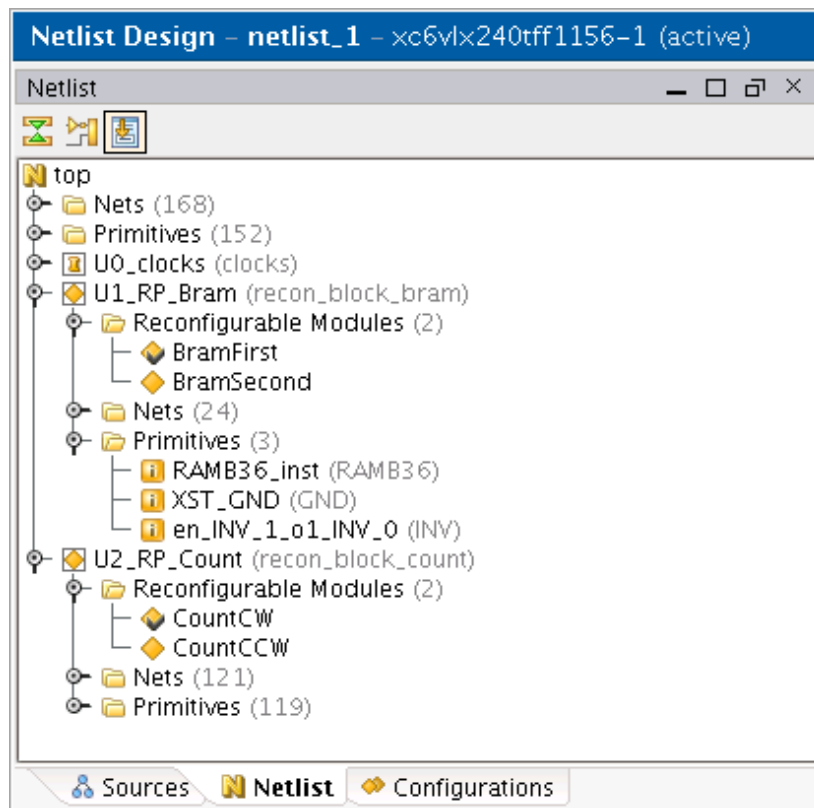


図 9 : リンコンフィギャブル モジュールを追加した後の [Netlist] ビュー

アクティブ リコンフィギャブル モジュールの変更

アクティブ リコンフィギャブル モジュールを別のリコンフィギャブル モジュールに変更するには、次の手順に従います。

1. 任意のリコンフィギャブル モジュールを選択します。
2. 右クリックします。
3. [Set as Active Reconfigurable Module] をクリックします。

リコンフィギャブル モジュールに関連付けられているプリミティブ

各リコンフィギャブル モジュールのリソースは [Netlist] ビューの [Primitives] フォルダーにあります。どのリコンフィギャブル モジュールがアクティブになっているかによりリソースは変わります。次のリソースは、現在アクティブなリコンフィギャブル モジュールで使用されます。

- RAMB36 およびインバーター : U1_RP_Bram に関連付けられているリコンフィギャブル モジュールの場合
- スライス ロジック (LUT、XORY、および FDR)、OBUF プリミティブ : U2_RP_Count に関連付けられているリコンフィギャブル モジュールの場合

OBUF および RAMB36 プリミティブは、関連付けられているリコンフィギャブル パーティションの AREA GROUP RANGE 制約に含めておく必要があります。これは手順 6 で説明します。

手順 5 : ブラック ボックス モジュールの追加 (オプション)

ブラック ボックス モジュールを作成すると、BitGen の段階で空白の BIT ファイルを生成できます。

ここでは次の作業を行います。

- ブラック ボックス モジュールを U1_RP_Bram へ追加
- ブラック ボックス モジュールを U2_RP_Count へ追加

ブラック ボックス モジュールを U1_RP_Bram へ追加

ブラック ボックス モジュールを U1_RP_Bram へ追加するには、次の手順に従います。

1. U1_RP_Bram をクリックします。
2. 右クリックします。
3. [Add Reconfigurable Module] をクリックします。
4. [Reconfigurable Module Name] に「BramBB」と入力します。
5. [Add this Reconfigurable Module as a black box without a netlist] をオンにします。
6. [Next] をクリックします。

ブラック ボックス モジュールにはネットリストや制約ファイルは関連付けられていないので、ウィザードではプロンプトは表示されません。

7. [Finish] をクリックします。

ブラック ボックス モジュールを U2_RP_Count へ追加

ブラック ボックス モジュールを U1_RP_Bram へ追加するには、次の手順に従います。

1. [Netlist] ビューで U2_RP_Count をクリックします。
2. 右クリックします。
3. [Add Reconfigurable Module] をクリックします。
4. [Reconfigurable Module Name] に「CountBB」と入力します。
5. [Add this Reconfigurable Module as a black box without a netlist] をオンにします。
6. [Finish] をクリックして、ウィザードを終了します。

これで、次の図のように [Netlist] ビューの各リコンフィギュラブル 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 つ表示されます。

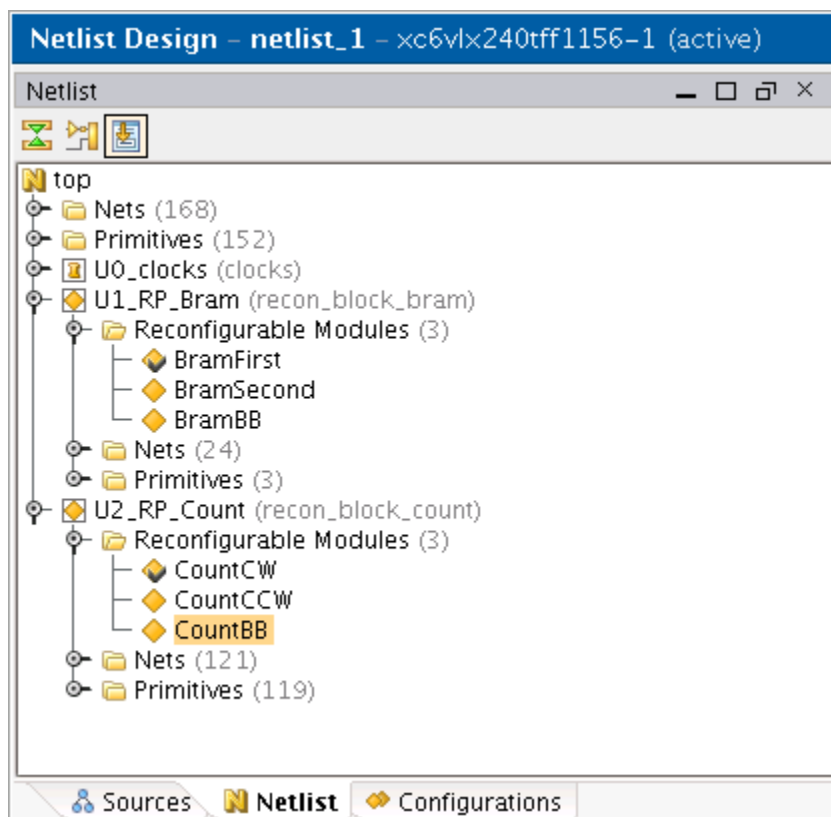


図 10 : [Netlist] ビューで表示されるブラック ボックス モジュール

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

各リコンフィギャブル パーティションには、どの物理的なリソースをリコンフィギャブル パーティションに含めるかを指定するために AREA_GROUP RANGE 制約が設定されている必要があります。この例の場合、リコンフィギャブル パーティションは U1_RP_Bram と U2_RP_Count です。

リコンフィギャブル パーティションに関連付けられている AREA_GROUP RANGE 制約に含まれていない物理的リソースは、すべてスタティック ロジックに含まれています。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です。パーシャル リコンフィギュレーション中のアクティブなロジックへの影響、およびデカップリング ロジックの使用についての詳細は、『パーシャル リコンフィギュレーション ユーザー ガイド』を参照してください。

手順 3 で説明されているように、[Set Partition] コマンドを使用してリコンフィギャブル パーティションを作成するまで AREA_GROUP RANGE 制約は作成しないでください。

pblock_U1_RP_Bram の AREA_GROUP RANGE 制約の作成

pblock_U1_RP_Bram の AREA_GROUP RANGE (範囲) 制約を作成するには、次の手順に従います。

1. [Netlist] ビューで BramFirst および CountCW がアクティブ モジュールになっていることを確認します。
アクティブ モジュールの名前の横にはチェック マークが表示されています。
2. BramFirst および CountCW がアクティブ モジュールでない場合は、次の作業を行います。
 - a) BramFirst および CountCW をクリックして選択します。
 - b) 右クリックします。
 - c) [Set as Active Reconfigurable Module] をクリックします。

PlanAhead ツールでは、アクティブなリコンフィギャブル モジュールの AREA_GROUP RANGE 制約に必要なリソースがレポートされるので、ブラック ボックスのリコンフィギャブル モジュールがアクティブになっていないようにしてください。

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ているリコンフィギャブル モジュールでさまざまなリソースが使用される場合、そのリコンフィギャブル パーティションの AREA_GROUP の RANGE (範囲) にはすべてのリコンフィギャブル モジュールで使用されるリソースを含める必要があります。

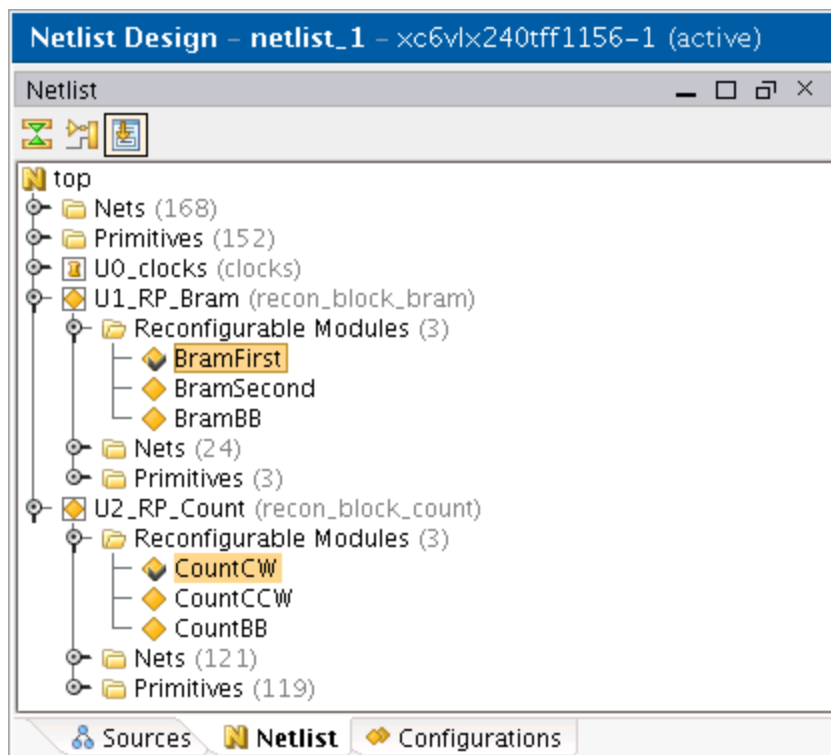



図 11 : アクティブなリコンフィギャブル モジュールの設定

3. ツールバーにあるビュー レイアウト セレクターでビューを [Design Analysis] から [Floorplanning] に変更します。
[Physical Constraints] ビューには、現在の Pblock すべてのリストが表示されます。
PlanAhead ツールでは AREA_GROUP 制約は Pblock と呼ばれます。Pblock はリコンフィギャブル パーティションとして定義されているモジュールすべてに対し自動的に作成されます。
4. [Physical Constraints] ビューで pblock_U1_RP_Bram をクリックします。
5. [Device] ビューの左側で [Set Pblock Size] ボタン  をクリックします。
このアイコンを表示させるには、Pblock を選択しておく必要があります。
6. 複数のスライス ロジックと RAMB36 (ピンクの列) を少なくとも 1 つ含める長方形を描画します。
7. 長方形を描画したら、そのエリア グループの [SLICE] と [RAMB36] を両方オンにします。

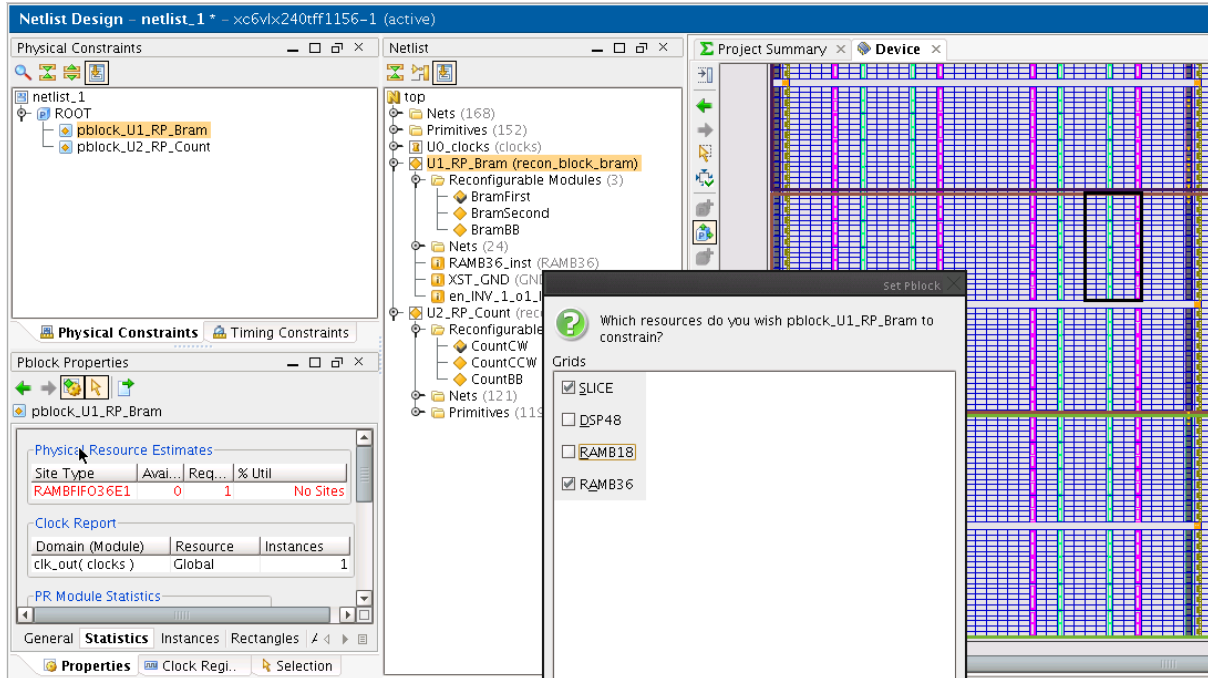


図 12 : Pblock リソース (pblock_U1_RP_Bram) の設定

8. [OK] をクリックします。

pblock_U2_RP_Count の AREA_GROUP RANGE 制約の作成

pblock_U2_RP_Count の AREA_GROUP RANGE 制約を作成するには、次の手順に従います。

1. pblock_U2_RP_Count をクリックします。
2. [Set Pblock Size] を使用し、スライス ロジックと I/O ロジックを含める長方形を描画します。

U2_RP_Count には UCF ファイルで I/O ピン配置が既に設定されているので、AREA_GROUP の RANGE には次の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。

- AD21
- AH27
- AE21
- AH28

3. 長方形を描画したら、このエリア グループの [SLICE] と [IOB] リソースをオンにします。

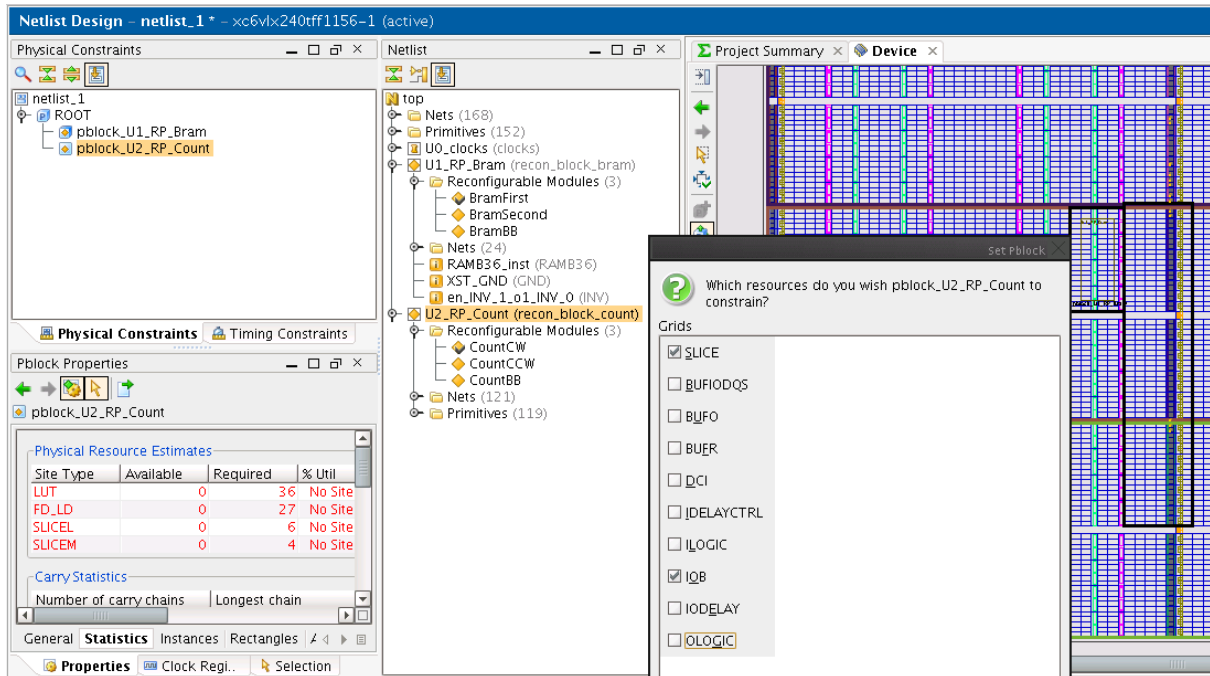


図 13 : Pblock リソース (pblock_U2_RP_Count) の設定

4. [OK] をクリックします。

リコンフィギャブル モジュールに入力または出力のフリップフロップ (または任意の入力または出力ロジック) がある場合は、ILOGIC や OLOGIC などのほかのリソースも含まれます。このチュートリアル of デザインにはそのようなリソースは含まれていません。

5. フロアプランを見直して、必要であれば調整します。

最終的なフロアプランは次の図のようになります。

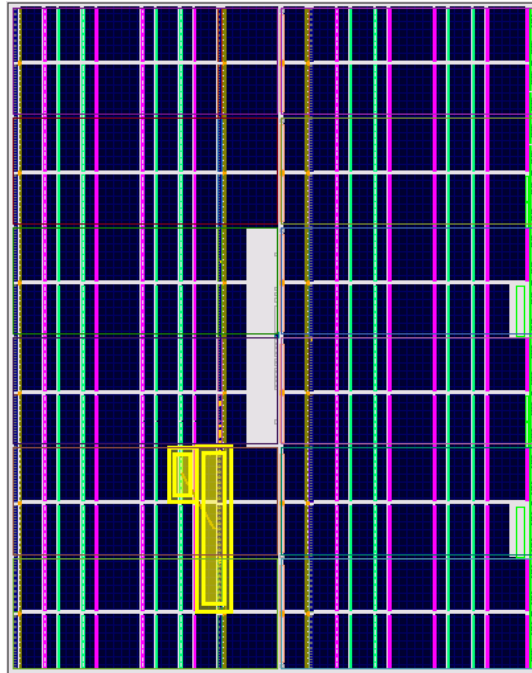


図 14 : 最終的な AREA_GROUP の範囲

6. U2_RP_Count に必要なピンが AREA_GROUP 範囲内にあることを確認します。

AD21、AH27、AE21、および AH28 ピンは、U2_RP_Count の AREA_GROUP 範囲の長方形内に含まれている必要があります。上の図を参照してください。

また、この長方形には I/O 列の右側が含まれています。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。すべてのリコンフィギャブル モジュール配線リソースは、AREA_GROUP RANGE で定義されているようにリコンフィギャブル パーティション領域内にある必要があります。
7. これらのピンが AREA_GROUP RANGE に含まれているかどうかは、[Device] ビューでこれらのピンを確認します。
 - a) [Edit] → [Find] をクリックします。
 - b) [Find] に [Sites] を選択し、[Name matches] に「AD21」を入力して [OK] をクリックします。
 - c) UCF には既に LOC 制約が含まれているので、このピンが [Device] ビューでハイライトされます。
 - d) [View] → [Fit Selection] をクリックし、ハイライトされたサイトを拡大表示します。
 - e) AH27、AE21、および AH28 ピンに対しても同じ作業を行います。
8. U1_RP_Bram に必要な RAMB36 が AREA_GROUP の RANGE に含まれていることを確認します。
 - a) [Physical Constraints] ビューで pblock_U1_RP_Bram をクリックします。
 - b) [Statistics] タブをクリックします。

次の図に含まれる RAMBFIFO36E1 の場合、必要数が使用可能な数を超えていないことが確認できます。

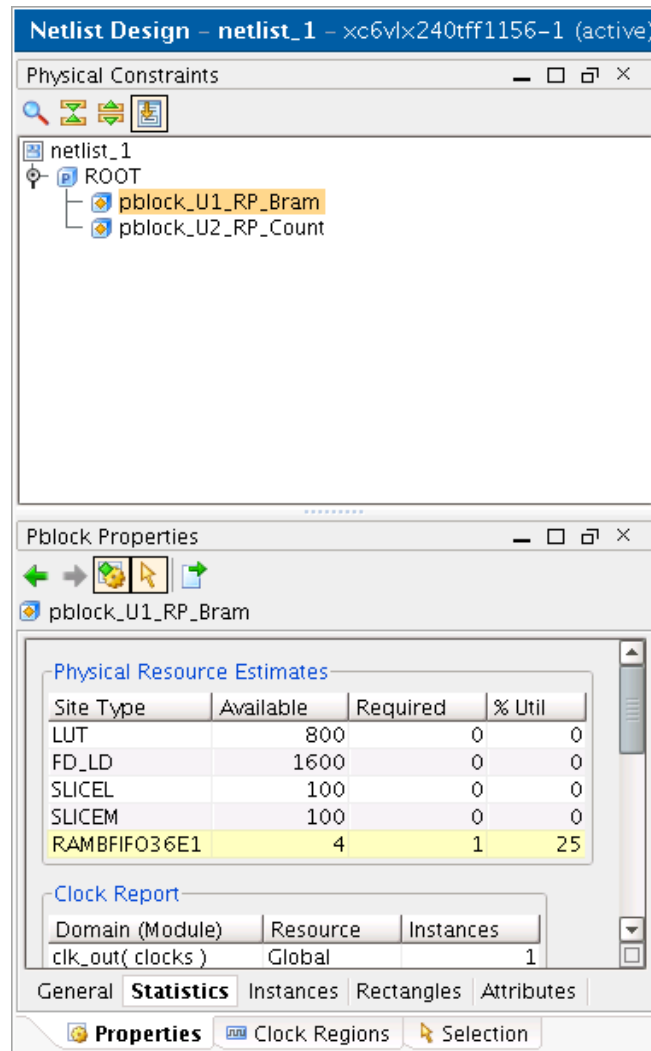


図 15 : Pblock (pblock_U1_RP_Bram) を示した [Statistics] タブ

9. [File] → [Save Design] をクリックします。
10. Flow Navigator で [Project Manager] をクリックします。
[Sources] ビューが開き、すべてのソースおよび制約ファイルが表示されます。
11. [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 AREA_GROUP 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;
```
12. [Netlist Design] ビューに戻ります。
13. ビュー レイアウト セレクターで [Design Analysis] をクリックし、PlanAhead ツールのデフォルトビューに戻ります。

手順 7 : パーティション ピンおよびリコンフィギュラブル パーティション インターフェイスのタイミング

パーシャル リコンフィギュレーション フローでは、グローバル ロジックや専用配線ではないリコンフィギュラブル パーティション インターフェイス信号すべてに対しパーティション ピンが必要です。パーティション ピンは、リコンフィギュラブル パーティションへの既知の配線接続を提供します。これらのピンは、インプリメンテーション実行時に NGDBuild により自動的に挿入されます。これ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 マクロに代わる技術です。

現段階のパーティション ピンのインプリメンテーションには、プロキシ ロジック (LUT1) が必要です。LUT1 は、リコンフィギュラブル パーティションの入力および出力パスに挿入されます。リコンフィギュラブル パーティションの両側の境界にある入力および出力には、レジスタを付けることを推奨します。レジスタを入力に付けることにより、リコンフィギュラブル パーティション インターフェイスに関連したタイミング クロージャの問題が軽減しやすくなります。

ガイドラインに従っている場合、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、タイミング要件が厳しい場合は、次の作業が必要になることがあります。

- TPSYNC 制約をパーティション ピンに設定する
- 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最小限に抑えるためスタティック ロジックに LOC 制約を追加する

各コンフィギュレーションで同じスタティック ロジック インプリメンテーションが使用されているので、最もタイミングクリティカルなリコンフィギュラブル モジュールのタイミングを一番最初に満たすようにします。このチュートリアルのリコンフィギュラブル パーティション インターフェイスは、リコンフィギュラブル モジュール間でほとんど違いはありません。リコンフィギュラブル パーティション インターフェイス タイミングを満たすには、グローバル PERIOD 制約で十分です。


PERIOD 制約の追加

PERIOD 制約を追加するには、次の手順に従います。

1. [Window] → [Timing Constraints] をクリックして [Netlist Design] ビューで [Timing Constraints] ビューを開きます。
2. [Timing Constraints] ビューで右クリックし、[New Timing Constraint] をクリックします。
3. [New Timing Constraint] の [Basic group (TNM)] カテゴリの下で次のように値を設定します。
 - [Group name] : clk_p
 - [Group type] : Net
 - [TNM type] : TNM_NET
 - [Predefined group] : 空白のまま残す
 - [Net] : clk_p
4. [OK] をクリックします。
5. もう一度 [New Timing Constraint] を開きます。

6. [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] : 5ns

7. [Group] の横にある参照アイコン () をクリックします。

8. [Group constraints type] で [User Defined] をクリックします。

9. [User defined groups] で [clk_p] を選択して、[OK] をクリックします。

10. もう一度 [OK] をクリックし、制約を作成します。

[Timing Constraint] ビューは、次の図のようになります。

このグローバル タイミング制約により、clk_p に接続されているすべての同期パスが制約されます。このタイミング制約は、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特化したものではありません。

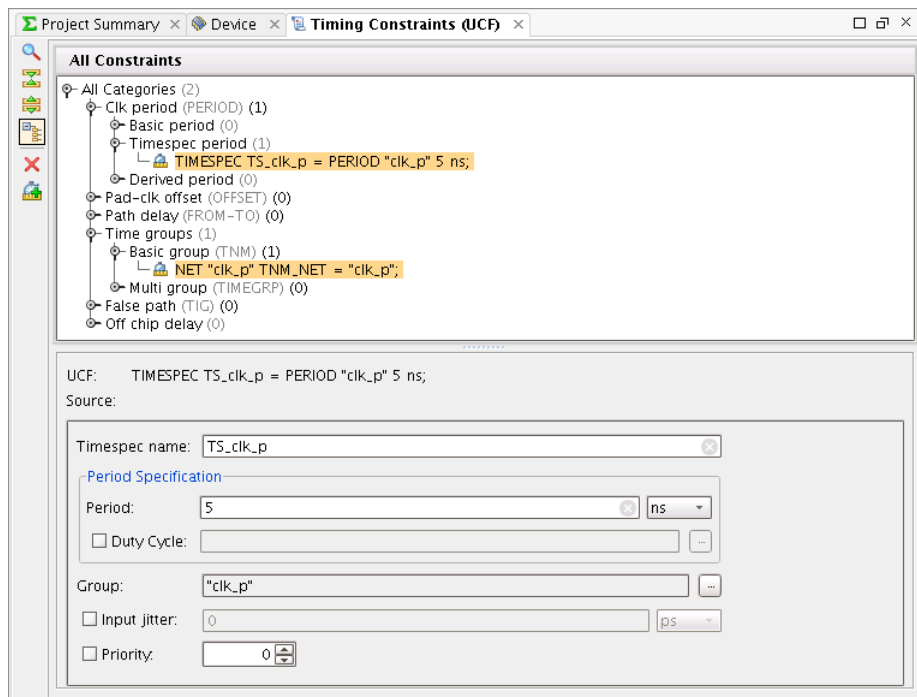


図 16 : [Timing Constraints] ビュー (pblock_U1_RP_Bram)

11. デザインを保存します。

12. 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ています。

- TIMESPEC TS_clk_p = PERIOD “clk_p” 5 ns;
- NET “clk_p” TNM_NET = “clk_p”;

前の手順から UCF ファイルが開いている場合は、変更したファイルを読み込むため [Reload] をクリックします。

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 チェックの実行

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に特化したデザイン 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 の機能に既に含まれているものもあり、[Partial Reconfig] および [Partition] の項目の下に含まれています。

コンフィギュレーションをインプリメントし、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 デザインでこれらのチェックを実行します。

通常のデザインでは、すべての PlanAhead DRC を実行することを推奨します。このチュートリアルでは、Partial Reconfig および Partition DRC のみを実行します。

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

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 を実行するには、次の手順に従います。

1. Flow Navigator で [Netlist Design] の下の [Run DRC] をクリックします。

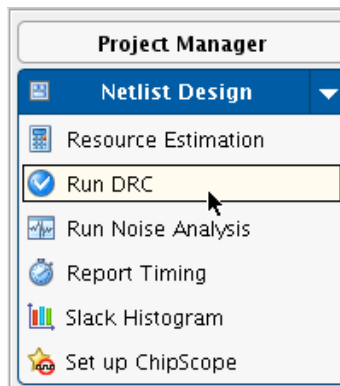


図 17 : DRC の実行

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] をクリックして選択した DRC ルールを実行します。

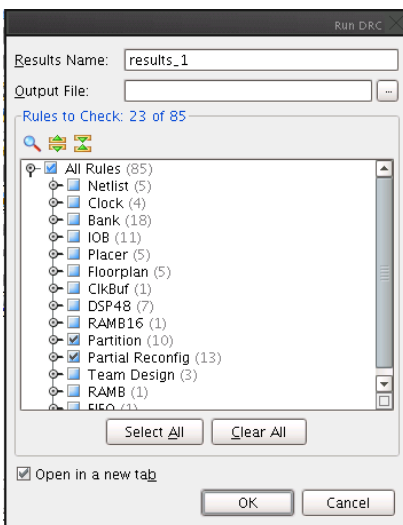


図 18 : [Partition] と [Partial Reconfig] の DRC の指定

DRC で出力されるメッセージの重要度を確認します。メッセージには、次のような重要度が設定されています。

- アドバイザリ
- 警告
- エラー
- 致命的エラー

この場合、一番重要度が高いのは警告です。

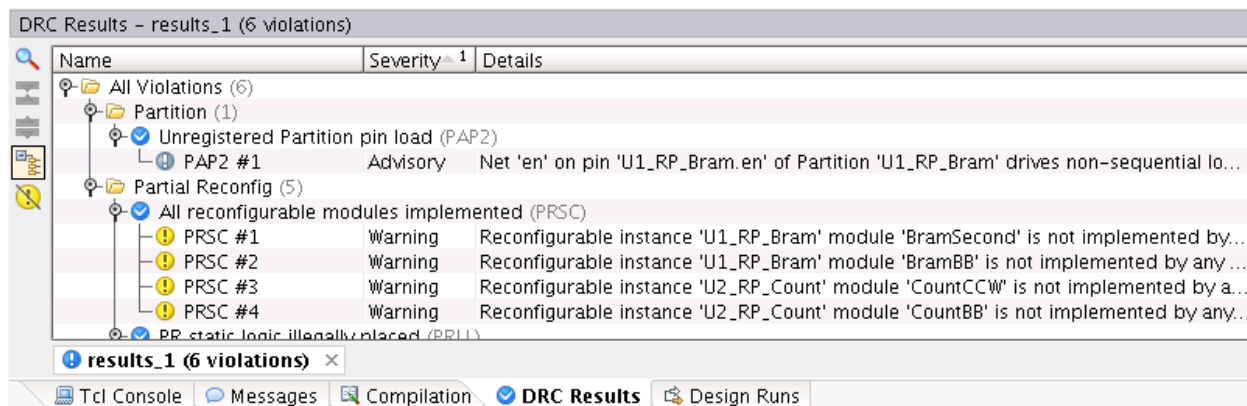


図 19 : [DRC Results] ビュー

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

各リコンフィギャブル パーティションには複数のリコンフィギャブル モジュールを関連付けることができますが、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。

スタティック ロジックとアクティブ リコンフィギャブル モジュールのセットは「コンフィギュレーション」と呼ばれ、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、さまざまな方法で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 ファイルを生成することができます。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ます。

出力ファイルは、次のようになります。

- フォーマット ファイル
 - NGD
 - NGM
 - NCD
 - PCF
- レポート ファイル

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するなど、各コンフィギュレーションで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 テクニックを使用できます。

このチュートリアルでのデザインは、次のリコンフィギャブル モジュール セットとその 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
```

つまり次のようになります。

- フル BIT ファイル config_1.bit : BramFirst および CountCW 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 config_2.bit : 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 セットでも別のフル 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
```

最後に、ブラック ボックスのリコンフィギャブル モジュールが生成されたら、追加コンフィギュレーションをインプリメントして、空白ビットストリームを生成することができます。このフル BIT ファイルにはブラック ボックス リコンフィギャブル パーティションのロジックは含まれておらず、リコンフィギャブル パーティションを空白にするには、パーシャル BIT ファイルを使用することができます。

Configuration

```
config_5 RMs:BramBB, CountBB
Bits: config_3.bit (full bit file)
config_3_U1_RP_Bram_BramBB_partial.bit
config_3_U2_RP_Count_CountBB_partial.bit
```

PlanAhead ツール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」と呼ばれます。run は各コンフィギュレーションに対して作成する必要があります。

PlanAhead ツールでは、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。各リコンフィギャブル パーティションで最初に定義されたリコンフィギャブル モジュール (このチュートリアルでは BramFirst と CountCW) がこのコンフィギュレーションに設定されています。

フル BIT ファイルでコンフィギュレーションされた FPGA デバイスには、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、すべてのリコンフィギャブル パーティションに対してブラック 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 ファイルは空白になります。

コンフィギュレーション config_1 をインプリメント

コンフィギュレーション config_1 をインプリメントするには、次の手順に従います。

1. config_1 に設定されてい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。
 - a) [Design Runs] ビュー ([Window] → [Design Runs]) で config_1 をクリックします。
 - b) [Implementation Run Properties] ビューで [Partitions] タブをクリックします。次の図を参照してください。
 - c) リストされている [Module Variants] が BramFirst と CountCW ではない場合は、変更してください。

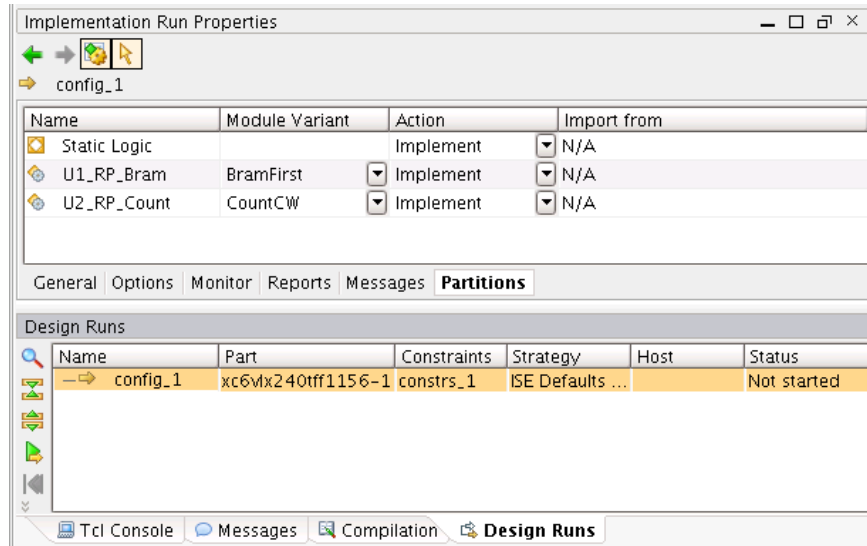


図 20 : コンフィギュレーション モジュール (config_1)

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] フィールドには NGDBuild、MAP、PAR、TRCE が実行されていることが示されます。進捗状況は、PlanAhead ツール右上のステータス バーにも表示されます。実行ステータスの詳細は [Compilation Log] ビューも参照してください。

3. config_1 コンフィギュレーションをプロモートします。Flow Navigator で [Promote Partitions] をクリックします。config_1 は問題なくインプリメントできたのでプロモートすることができます。最初のコンフィギュレーションをプロモート (その後にインポート) せずにほかのコンフィギュレーションを実行すると、コンフィギュレーション間のパーシャル BIT ファイルの互換性が失われます。

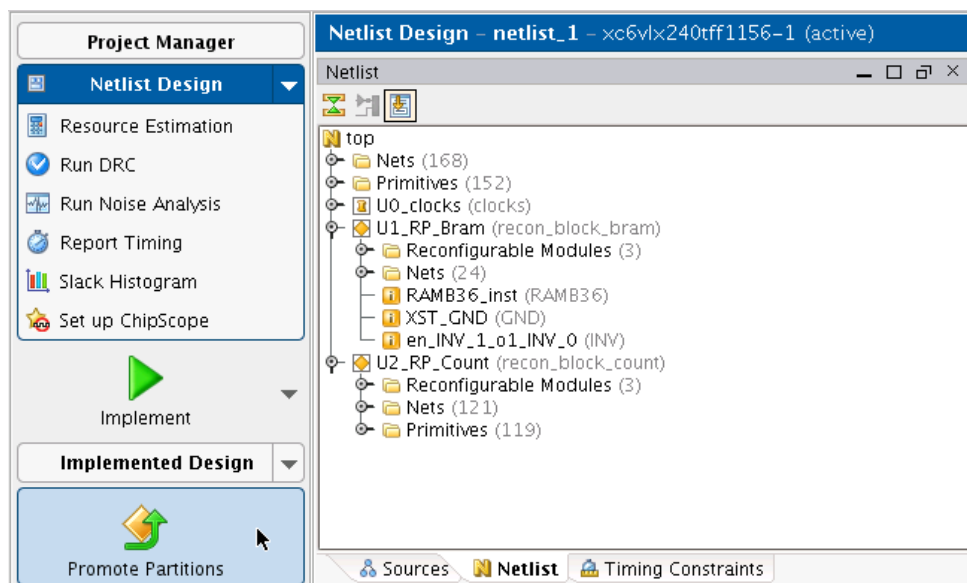


図 21 : [Promote Partitions] ボタン

- [Promote Partitions] ダイアログ ボックスで [OK] をクリックします。
[Automatically manage Partition action and import location] がオンになっていることを確認します。
- [Configurations] タブをクリックします。開いていない場合は [Window] → [Configurations] をクリックします。ステータスが [Promoted] に変更されています。

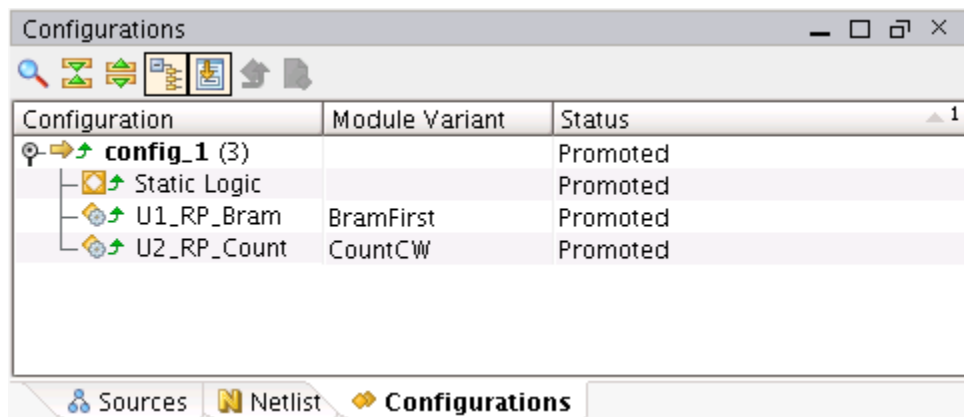


図 22 : [Promoted] ステータスの確認

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

新規プロジェクトを作成するには、次の手順に従います。

- 1 Flow Navigator の [Implement] ドロップダウン メニューから [Create New Implementation Runs] をクリックします。

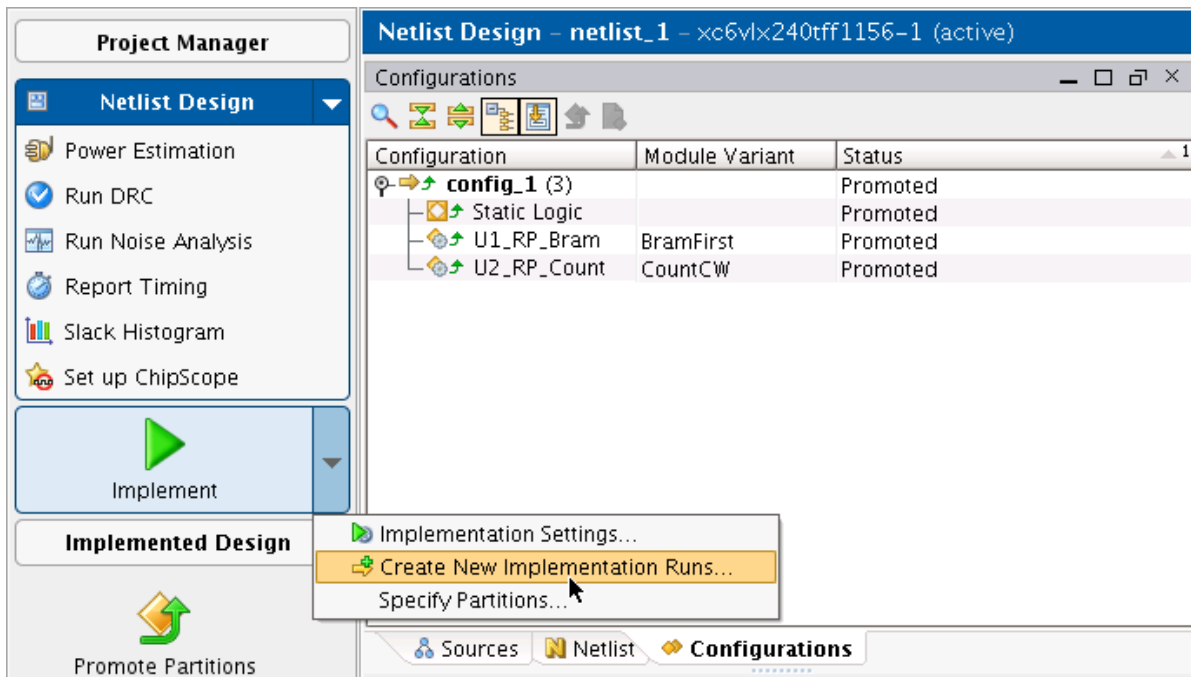


図 23 : 新規インプリメンテーション run の作成

2. Create New Runs ウィザードの最初のページで [Next] をクリックします。
3. [Set-Up Implementation Runs] ページで [Next] をクリックします。
[Choose Implementation Strategies and Reconfigurable Modules] ページでは、インプリメンテーション ストラテジを選択し、コンフィギュレーションにどのリコンフィギュラブル モジュールを使用するか定義して、複数のコンフィギュレーションを作成することができます。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。
4. 名前を変更することもできますが config_2 をそのまま使用します。
5. [Partition Action] 列の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。

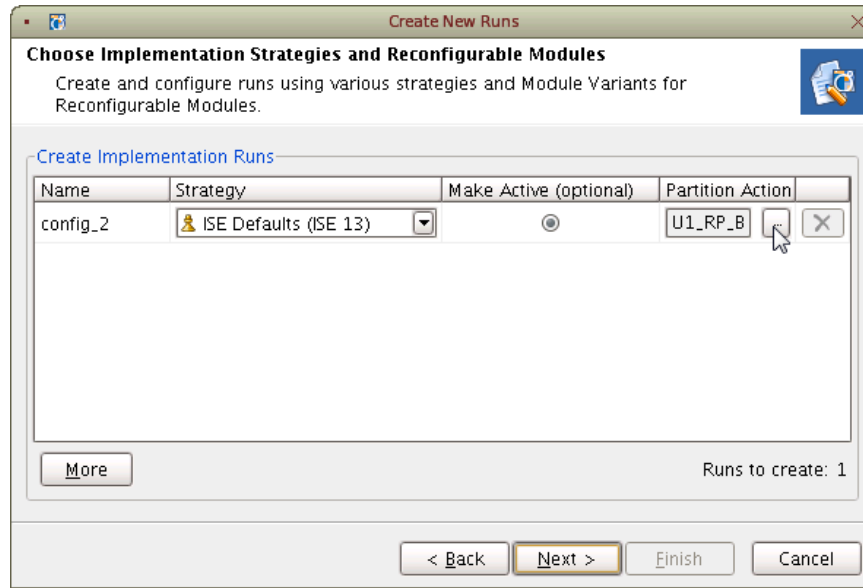


図 24 : インプリメンテーション run の作成

[Module Variant] にデフォルトで表示されるものは、プロジェクトでどのリコンフィギャブル モジュールがアクティブであるかによって変わります。この場合は BramFirst および CountCW が表示されています。これらのモジュールは既にインプリメントおよびインポートされているので、[Action] が [Import] に設定されていて、インポート ロケーションも設定されています。

6. 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 モジュールと同じになるように変更します。次の図を参照してください。

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] 列は [Implement] になります。

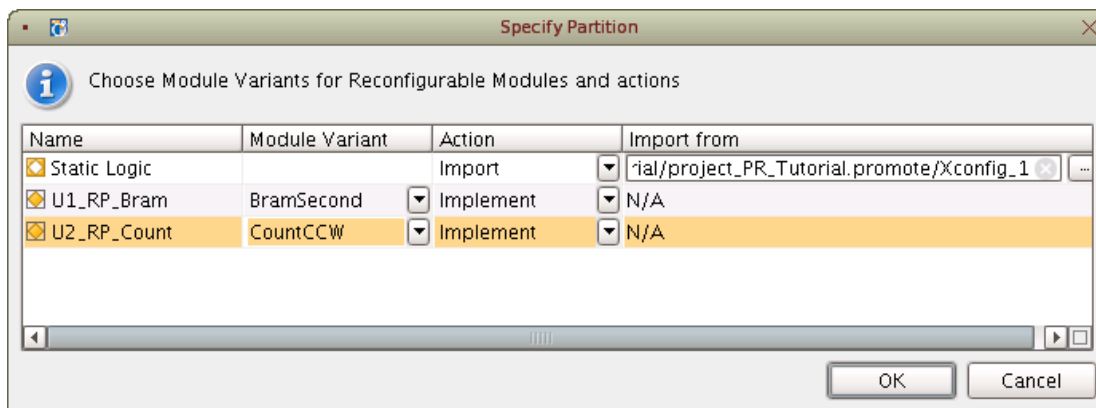


図 25 : [Specify Partition] ダイアログ ボックスの設定 (BramSecond/CountCCW)

7. [Specify Partition] ダイアログ ボックスで [OK] をクリックします。
8. オプション : ブラック ボックス モジュールを作成した場合、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。

- a) [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。
- b) [Specify Partition] ダイアログ ボックスを開きます。
- c) [Module Variants] 列に BramBB および CountBB を設定します。
- d) [OK] をクリックします。

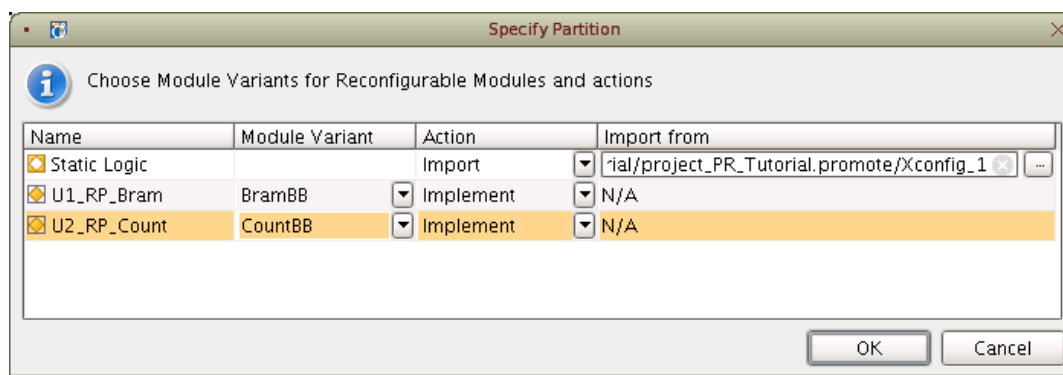


図 26 : [Specify Partition] ダイアログ ボックスの設定 (BramBB/CountBB)

9. [Next] をクリックします。
10. [Launch Options] ページで [Do not launch now] をオンにし、[Next] をクリックします。
11. [Create New Runs Summary] ページで [Finish] をクリックします。
[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。
12. [Design Runs] ビューで新しいコンフィギュレーションを選択します。
13. [Implementation Run Properties] ビューの [Partitions] タブをクリックし、[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。

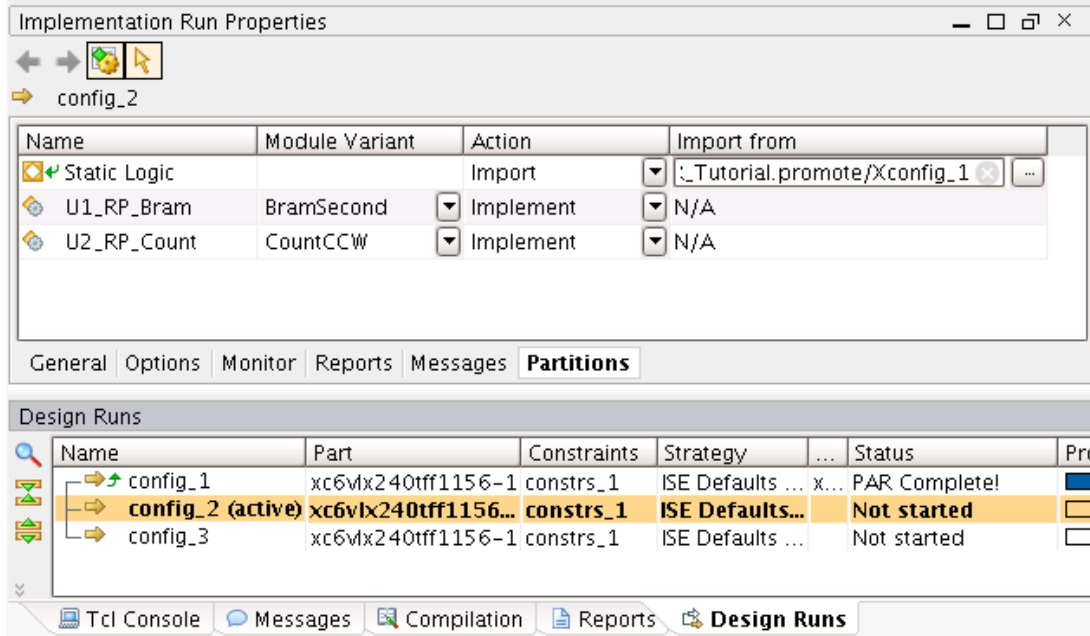


図 27 : コンフィギュレーション設定の確認

14. コンフィギュレーションを実行するには、次の手順に従います。

- a) [Design Runs] ビューで 1 つまたは複数の run を選択します。
- b) 右クリックして [Launch Runs] をクリックします。

複数の新規コンフィギュレーションを平行に実行できます (あれば複数のプロセッサ上で)。結果は相互に依存するものではなく、config_1 のスタティック ロジックにのみ依存しています。

15. [Launch Selected Runs] ダイアログ ボックスで次の作業を行います。

- a) [Launch Runs On Local Host] をオンにします。
- b) ジョブ数を選択します (使用するプロセッサ数)。
- c) [OK] をクリックします。

[OK] をクリックした後に [Promote Partitions] ダイアログ ボックスが表示されることがあります。これは既知の問題で、次のリリースの PlanAhead ツールで修正される予定です。config_1 が既にプロモートされていることがわかっている場合は、このダイアログ ボックスで [Launch Run] をクリックします。

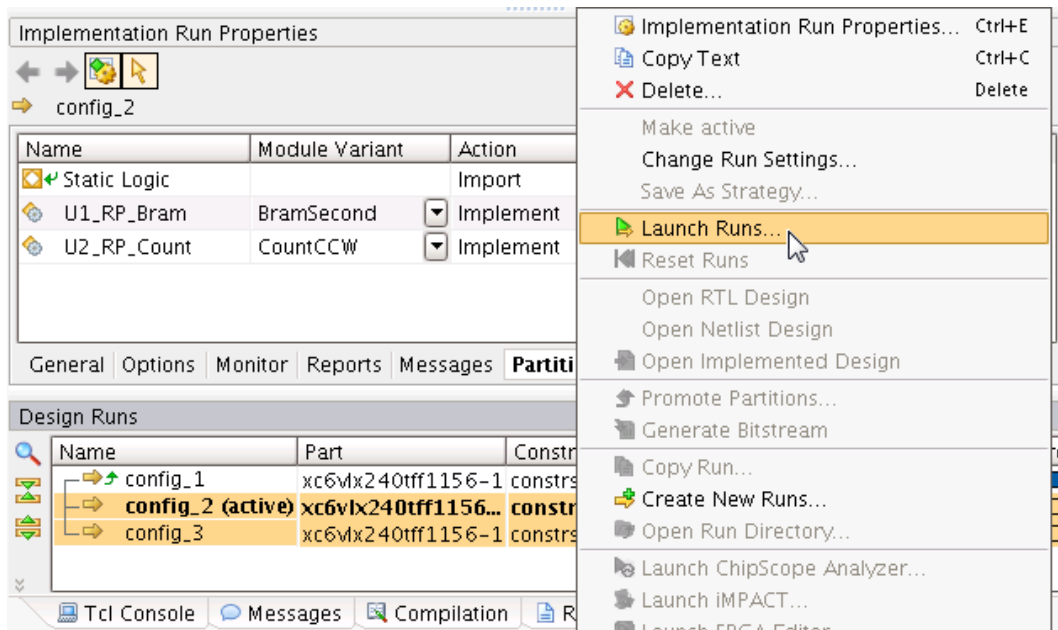


図 28 : run の実行

手順 11 : コンフィギュレーションの検証

複数のコンフィギュレーションをインプリメントしたら、すべてのコンフィギュレーションでスタティック ロジックおよびパーティション ピンが同じであることを比較して確認します。BIT ファイルに互換性があることを確認するために [Verify Configuration] を実行します。

すべてのコンフィギュレーションで検証を実行

すべてのコンフィギュレーションで検証を実行するには、次の手順に従います。

1. Flow Navigator の [Program and Debug] ドロップダウン メニューから [Verify Configuration] をクリックします。

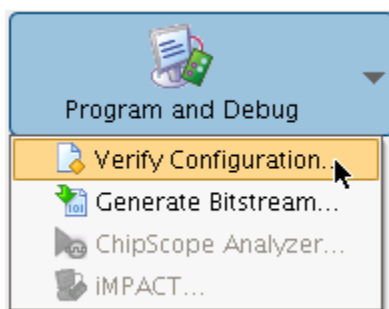


図 29 : コンフィギュレーションの検証

2. 相互検証するため、2 つ以上のコンフィギュレーションを選択します。
3. [Verification Output File] の値を確認します。

これは pr_verify のログ ファイルなので必要であればこの名前を変更します。[Verify Configuration] にエラーが発生した場合は、その原因説明にこのファイルが役立ちます。

3. [OK] をクリックします。

正しく検証チェックが終了した場合は、PlanAhead ツールに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。これで BIT ファイルを生成することができます。

4. [OK] をクリックします。

手順 12 : BIT ファイルの生成とダウンロード

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。

- フル BIT ファイル : 最初にこのファイルを使用して FPGA をプログラムできます。
- パーシャル BIT ファイル : 特定コンフィギュレーションに関連付けられた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 1 つずつ生成されます。任意のパーシャル BIT ファイルを使用して、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ことができます。

このチュートリアルでは、次のコンフィギュレーションを作成しました。

- BramFirst
- CountCW
- BramSecond
- CountCCW
- BramBB (オプション)
- CountBB (オプション)

各コンフィギュレーションで BitGen を実行すると、すべてのリコンフィギュラブル モジュールに対しパーシャル BIT ファイルが作成されます。デバイスを最初にコンフィギュレーションするのにフル BIT ファイルを使用した場合でも、パーシャル BIT ファイルを使用してそのファイルに関連付けられているパーシャル 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できます。

すべてのコンフィギュレーションの BIT ファイルを生成

すべてのコンフィギュレーションの BIT ファイルを生成するには、次の手順に従います。

1. [Design Runs] ビューで次の作業を行います。
 - a) すべてのコンフィギュレーションをハイライトします。
 - b) 右クリックします。
 - c) [Generate Bitstream] をクリックします。

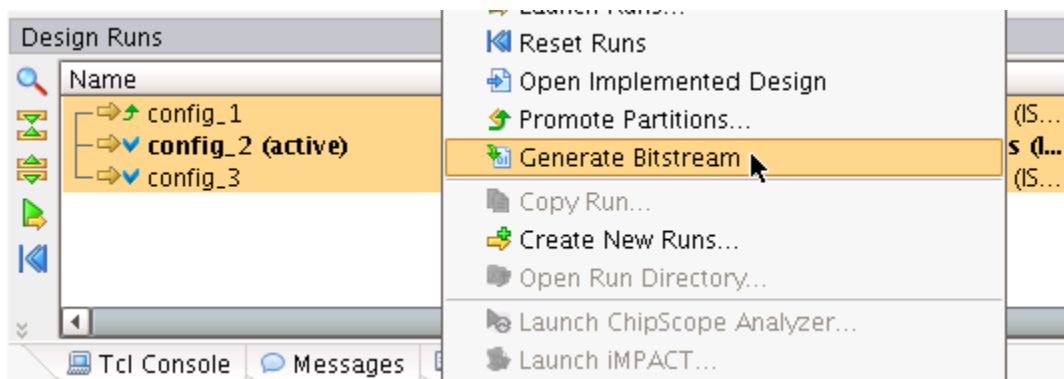


図 30 : ビットストリームの生成

2. パーシャル BIT ファイルを生成するのに特別な BitGen オプションは必要ありません。[Generate Bitstream] 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 デバイスにダウンロードされます。BIT ファイルのダウンロード、検証、およびデバッグには iMPACT を使用できます。

3. ML605 ボードと PC を USB ダウンロード ケーブルで接続します。
4. スタンドアロン モードで iMPACT を起動するには、Flow Navigator の [Program and Debug] ドロップダウン メニューで [iMPACT] をクリックします。

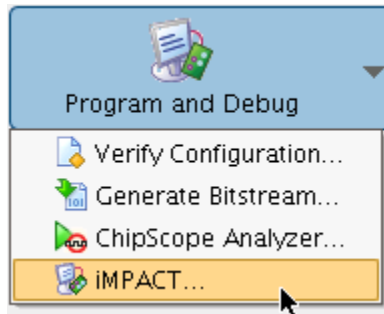


図 31 : iMPACT の起動

5. iMPACT フローは、次のようになります。
 - a) [Boundary Scan] をダブルクリックします。
 - b) [Initialize Chain] をクリックします。
6. チェーンが検出されたら、次の作業を行います。
 - a) xc6v1x240t デバイスを右クリックします。
 - b) [Assign New Configuration File] をクリックします。
 - c) 次のフル BIT ファイルを選択します。

<Extract_Dir>/PlanAhead/<project_name>/<project_name>.runs/config_1/config_1.bit

7. もう一度 xc6v1x240t デバイスを右クリックします。
8. [Program] をクリックします。

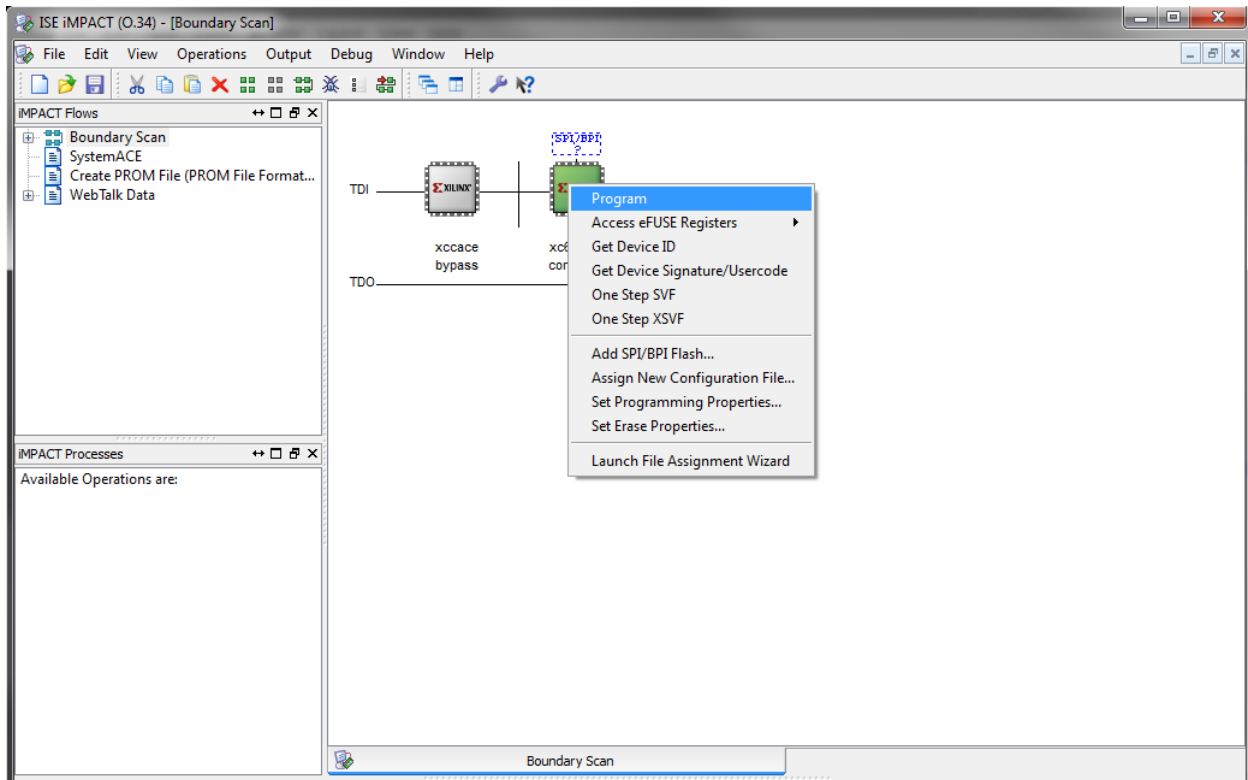


図 32 : iMPACT を使用したデバイスのコンフィギュレーション

ML605 の FPGA デバイスがフル BIT ファイルでプログラムされます。

パーシャル BIT ファイルの関連付け

パーシャル BIT ファイルを関連付けるには、次の手順に従います。

1. xc6vlx240t デバイスを右クリックします。
2. [Assign New Configuration File] をクリックします。
3. 次のフル BIT ファイルを選択します。
`<Extract_Dir>/PlanAhead/<project_name>/<project_name>.runs/config_2/config_2_U1_RP_Bram_BramSecond_partial.bit`
4. xc6vlx240t デバイスをもう一度右クリックします。
5. [Program] をクリックします。

まとめ

このチュートリアルでは、次の内容を学びました。

- PlanAhead ツールでパーシャル 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。
- 2 つのリコンフィギャブル パーティションを作成しました。
- 各リコンフィギャブル パーティションに複数のリコンフィギャブル モジュールを関連付けました。
- 各リコンフィギャブル パーティションを AREA_GROUP 制約を使用してデバイスの特定箇所に設定しました。
- デザイン全体を制約するためグローバル タイミング制約を作成しました。
- 初期コンフィギュレーションをインプリメントおよびプロモートしました。
- 初期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た追加コンフィギュレーションを作成しました。
- すべてのコンフィギュレーションが一貫していることを確認しました。
- BIT ファイルを生成しました。
- BramFirst および CountCW モジュールを含むフル BIT ファイルを ML605 ボードにダウンロードしました。
- BramSecond モジュールのパーシャル BIT ファイルを使用して U1_RP_Bram をリコンフィギュレーションしました。