

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

UG743 (v 13.1) 2011 年 3 月 1 日



The information disclosed to you hereunder (the “Information”) is provided “AS-IS” with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. **XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.**

© Copyright 2011 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

本資料は英語版 (v 13.1) 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。

改訂履歴

次の表に、この文書の改訂履歴を示します。

日付	バージョン	改訂内容
2011 年 3 月 1 日	13.1	ISE 13.1 リリース用に改訂

目次

改訂履歴.....	2
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	
概要	5
チュートリアル目標.....	5
はじめに.....	5
チュートリアル手順.....	9
手順 1：HDL ソースからのネットリストの合成 (オプション)	10
手順 2：プロジェクトの作成.....	11
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...	13
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).....	17
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.....	19
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	24
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	26
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.....	28
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	32
手順 11：コンフィギュレーションの検証.....	38
手順 12：ビット ファイルの生成とダウンロード.....	39
まとめ	41
付録 A：その他のリソース	
ザイリンクス リソース	43
パーシャル リコンフィギュレーション資料.....	43
PlanAhead 資料.....	43

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

概要

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™ 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が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、[付録 A「その他のリソース」](#)に示す『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 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。[付録 A「その他のリソース」](#)の PlanAhead のチュートリアルのリンク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 PR 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、PR 専用の 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、コンフィギュレーションを検証し、ハードウェアでパーシャル 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。

はじめに

ソフトウェア要件

PlanAhead ソフトウェアは、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 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。

インストール方法およびその詳細は、[付録 A「その他のリソース」](#)に示される『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。

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用の FlexLM ライセンスを取得する必要があります。30 日間限定の無料ライセンスを取得する場合は、ザイリンクスの FAE (フィールド アプリケーション エンジニア) にご連絡いただくか、<http://japan.xilinx.com/getproduct> から取得してください。オプションで ML605 ボードと USB ダウンロード ケーブルを使用してハードウェアでテストすることもできます。

ハードウェア要件

大規模デバイスで PlanAhead ソフトウェアを使用するには、2GB 以上の RAM が推奨されます。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。

チュートリアル デザイン ファイルのディレクトリ

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_planahead13-1_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> と記述します。

チュートリアルのサンプル データは、チュートリアルを実行中に変更されます。各チュートリアルを実行する前に、まず元の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。

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ュラブル 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ュラ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

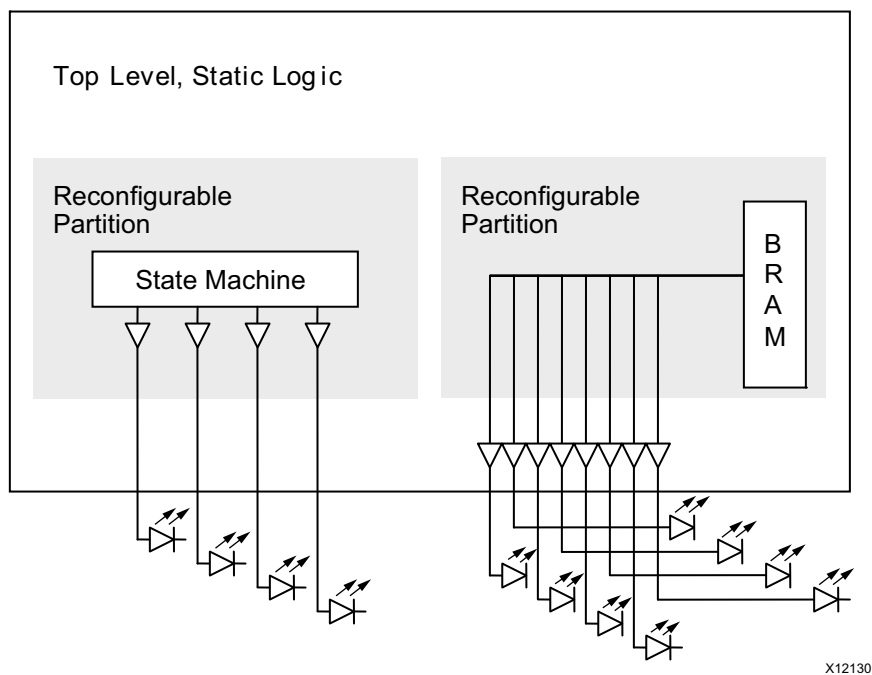
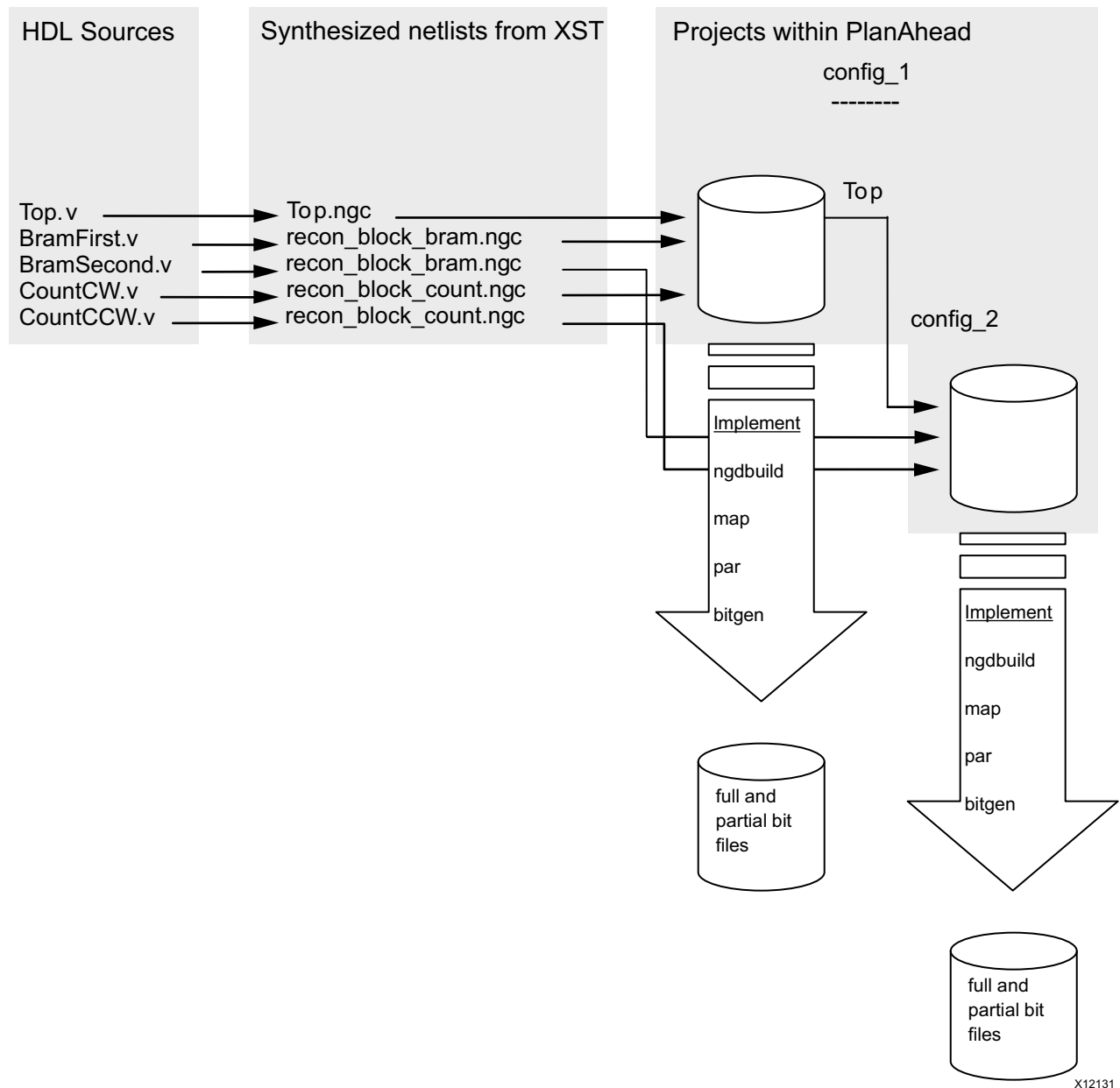


図 1：デザインの概要 - パーシャル リコンフィギュレーション

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 (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** 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。



X12131

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のディレクトリ構造は次のようになります。

```
<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 (RM) 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 RM 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 `<Extract_Dir>/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

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 (オプション)」
- 「手順 2: 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と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 チェックの実行」
- 「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: コンフィギュレーションの検証」
- 「手順 12: ビット ファイルの生成とダウンロード」

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 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ため、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。この NCG ファイルを使用する場合は、「[手順 2 : プロジェクトの作成](#)」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>/_bram_led/ImplementationSynth/Top/Top.xst
- <Extract_Dir>/ImplementationSynth/Top/Top.prj

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ュラブル 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 リコンフィギュラブル パーティションのような特殊な場合を除く)。

<Extract_Dir>/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

1. <Extract_Dir>/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。

- <Extract_Dir>/Source/Bram* のモジュールの場合は、RM 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、RM 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>/ImplementationSynth/<module> ディレクトリに保存されています。

メモ : 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>/ImplementationSynth/<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

PlanAhead を起動し、新規プロジェクトを作成

1. PlanAhead ソフトウェアを起動します。
 - Windows の場合、Xilinx PlanAhead 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 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。

PlanAhead の Getting Started ページが開きます。

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。
3. [Next] をクリックします。
[Project Name] ページが開きます。
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。
5. [Specify synthesized (EDIF or NGC) netlist] をオンにし [Set PR Project] をオンにした後 [Next] をクリックします (図 3)。

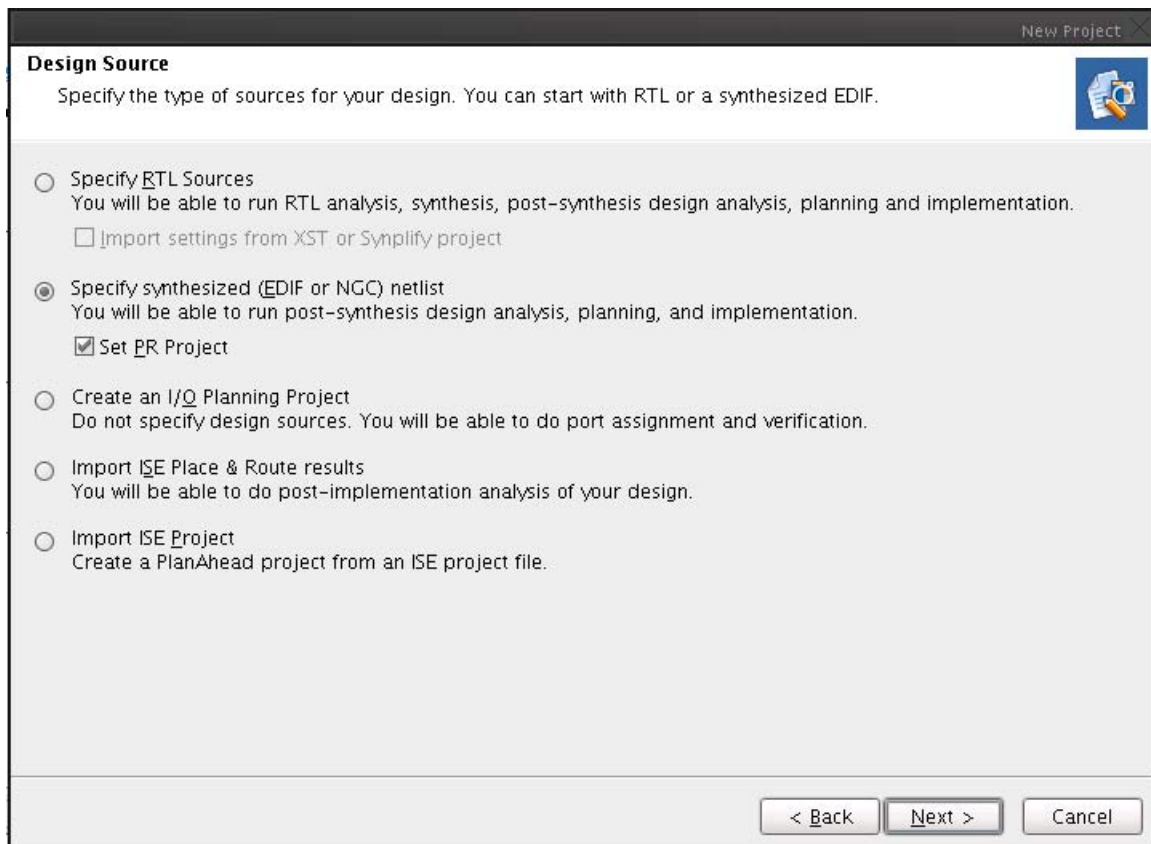


図 3：デザイン ソースの指定

6. [Top Netlist File] で <Extract_Dir>/ImplementationSynth/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 PR プロジェクトで使用されるべきです。リコンフィギュラブル モジュール (RM) 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。

7. [Constraint Files] ページで [Add Files] ボタンをクリックし、次のユーザー制約ファイル (UCF) を指定します。

<Extract_Dir>/Source/UCF/top_ml605.ucf

8. [OK] をクリックし、[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、最適なパーツが自動的に選択されます。

9. 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。

10. [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。

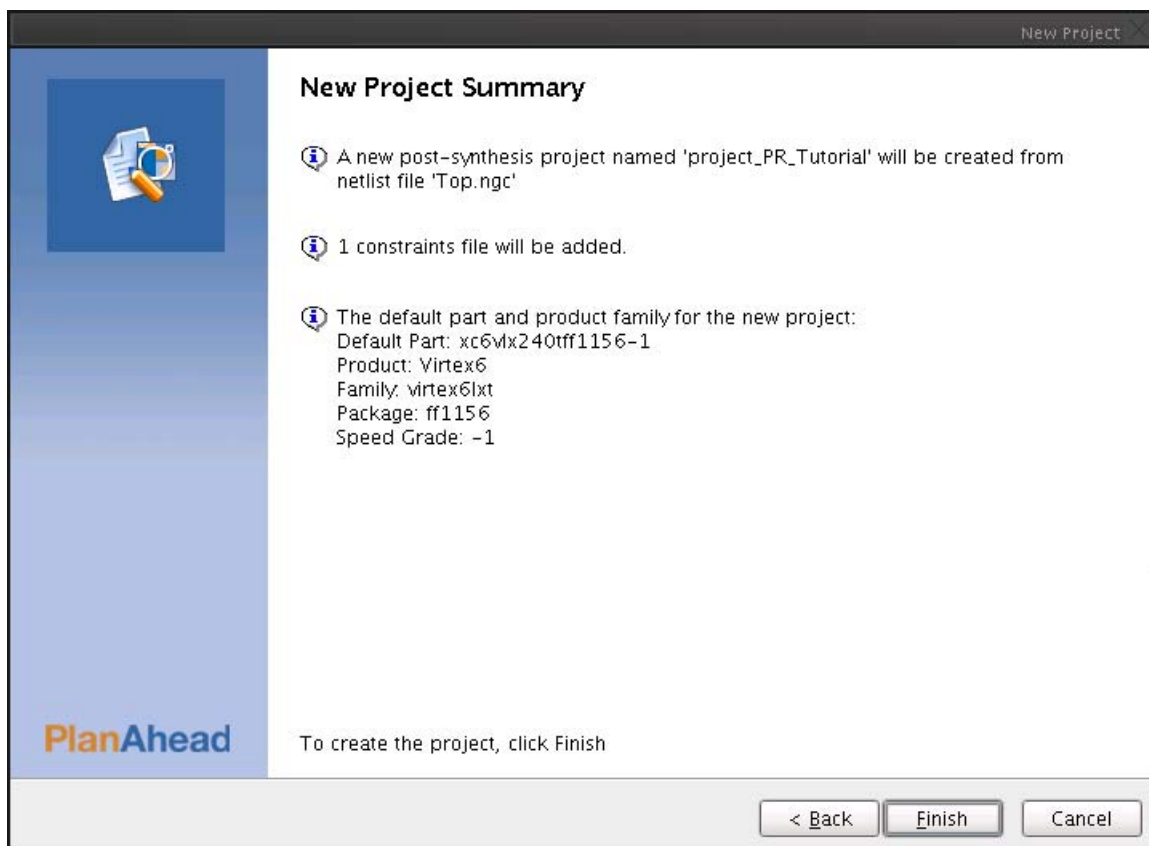


図 4 : 新規プロジェクトのサマリ

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

U1_RP_Bram のリコンフィギャブル パーティション (RP) を作成

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。
リコンフィギャブル モジュール (RM) にはまだネットリストを割り当てていないので、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。
2. [OK] をクリックします。

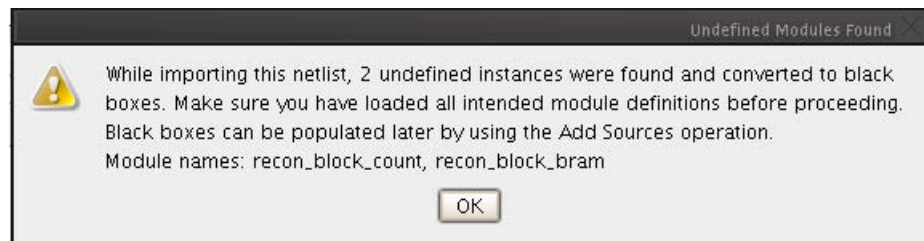


図 5：定義されていないモジュールがあることを示す警告メッセージ

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なおした場合は、Flow Navigator の [Netlist Design] ボタンをクリックして [Netlist Design] ビューを開き直す必要があります。

3. [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。

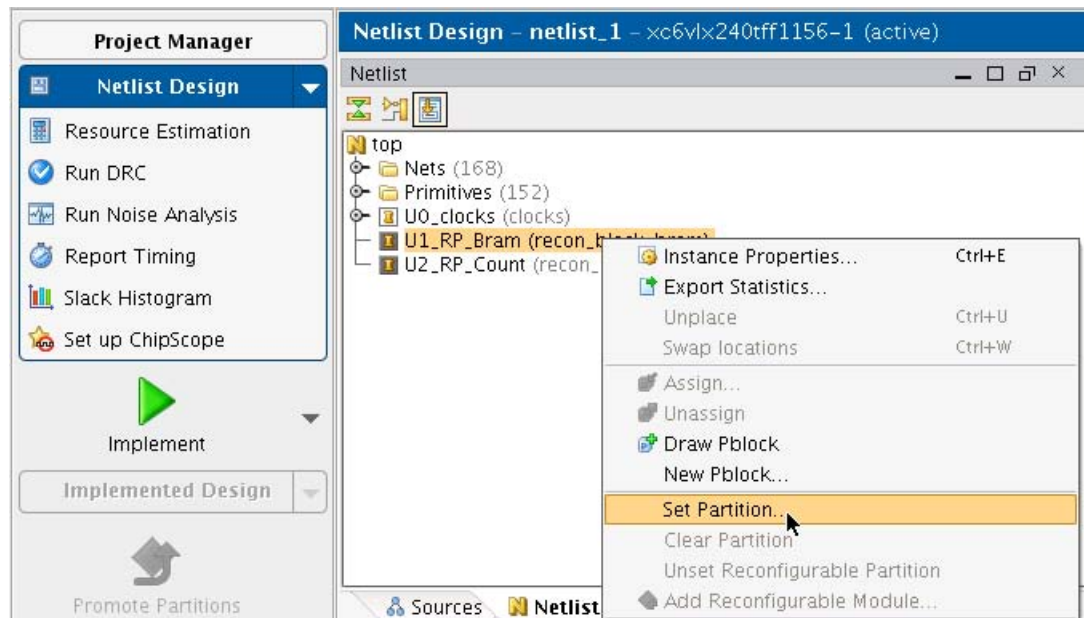




図 6：パーティションとリコンフィギャブル モジュールの設定

4. 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] をオンにします。[Next] をクリックします。
5. [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。
6. [Top Netlist File] で <Extract_Dir>/SynthImplementation/BramFirst/recon_block_bram.ngc を参照します。
7. [開く] をクリックし、[Next] をクリックします。
メモ：RM 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。
8. [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、この場合はファイルがありません。
9. [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 (図 7)。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。

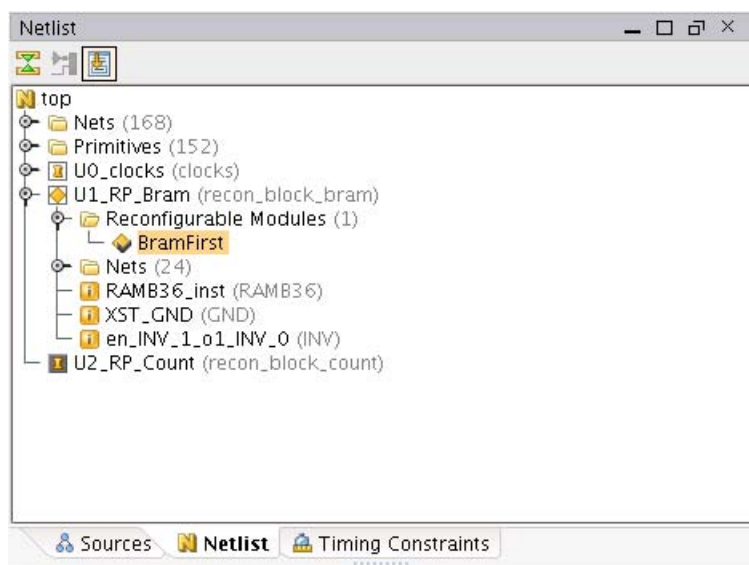


図 7：リコンフィギャブル パーティションのアイコン

U2_RP_Count の RP を作成

1. U2_RP_Count を選択して右クリックし、[Set Partition] を選択します。
2. リコンフィギャブル パーティションを設定する前に、[Yes] をクリックし、ネットリスト デザインを保存します。
3. 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 <Extract_Dir>/implementationSynth/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になります。次の手順では、RP に RM を追加する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

RP U1_RP_Bram にリコンフィギャブル モジュール (RM) を追加

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。

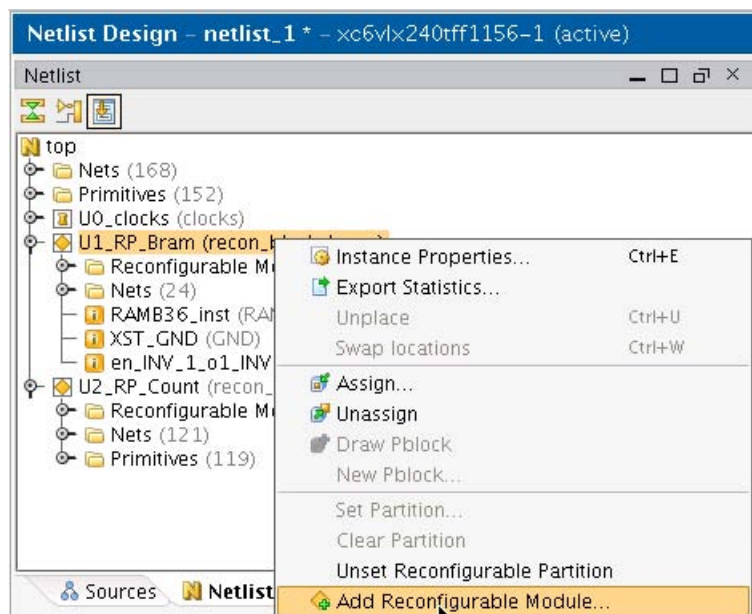


図 8：リコンフィギャブル モジュールの追加

2. 最初のページで [Next] をクリックします。
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. [Top Netlist File] で <Extract_Dir>/ImplementationSynth/BramSecond/recon_block_bram.ngc を参照します。
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。
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。
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。

U2_RP_Count に RM を追加

1. U2_RP_Count を選択して右クリックし、[Add Reconfigurable Module] を選択します。

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 <Extract_Dir>/implementationSynth/CountCW/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 (RM) 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 RM 用のもので、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 RM 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 RM 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 : 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。

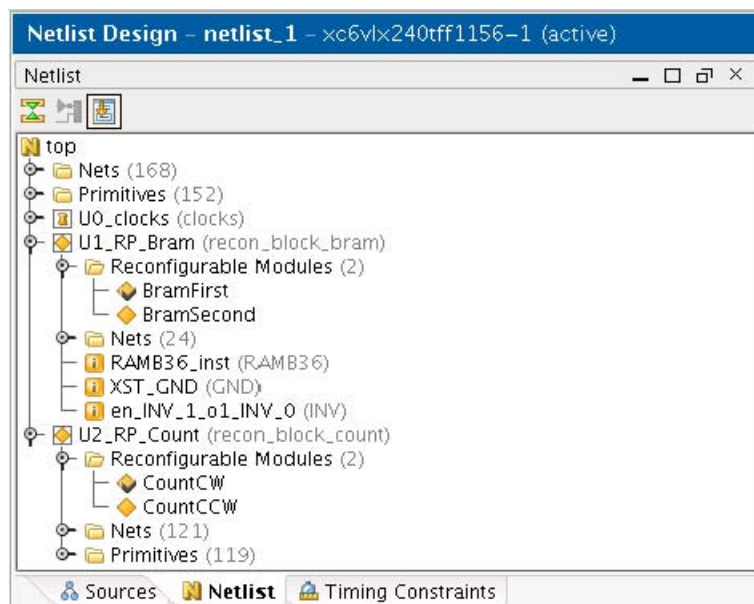


図 9 : リコンフィギャブル モジュールを追加した後の [Netlist] ビュー

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、[付録 A「その他のリソース」](#)に示す『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。

ブラック ボックス モジュールを U1_RP_Bram へ追加

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。Add Reconfigurable Module ウィザードが表示されます。
2. 最初のページで [Next] をクリックします。
3. 新しい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。
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。
4. [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。

ブラック ボックス モジュールを U2_RP_Count へ追加

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。Add Reconfigurable Module ウィザードが表示されます。
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] をオン

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 (RP) 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)。RM のネットまたはプリミティブは表示されなくなります。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。

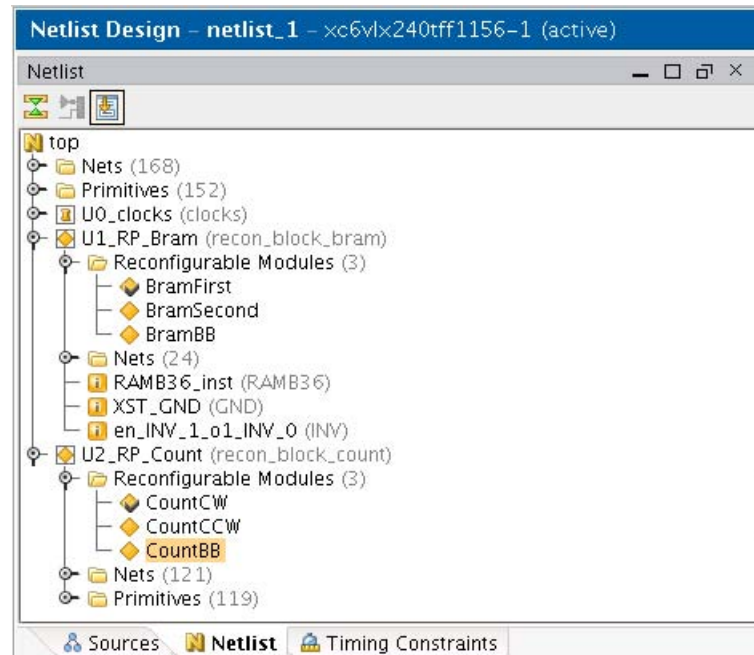


図 10 : ブラック ボックス モジュールを含む [Netlist] ビュー

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 (RP) 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。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 ロジックはパシシャル 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、RP が「[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](#)」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。

pblock_U1_RP_Bram の AREA_GROUP 範囲を作成

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。

PlanAhead では、アクティブなリコンフィギャブル モジュール (RM) の AREA_GROUP 範囲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 RM は必ず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 (RP) に関連付けられた異なる RM がさまざまなリソースを使用する場合、その RP の AREA_GROUP 範囲には RM 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。

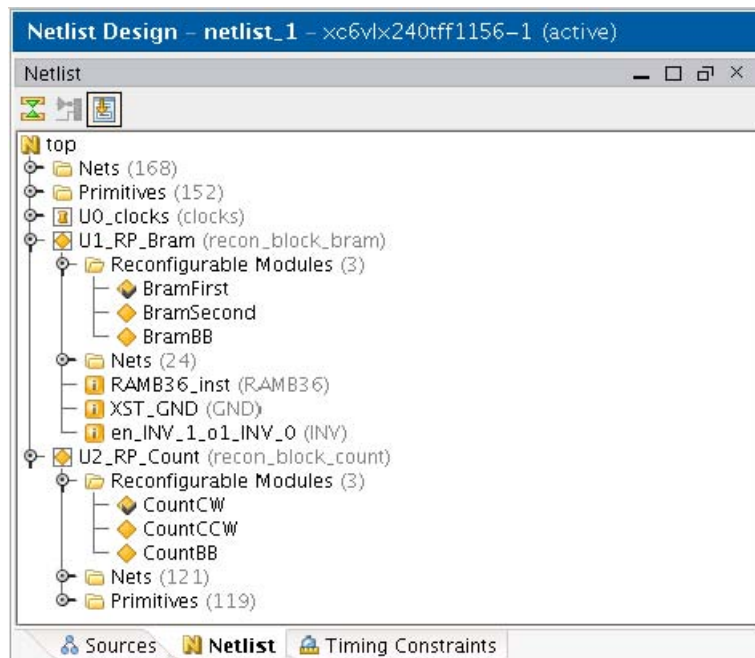



図 11：アクティブなリコンフィギャブル モジュールの設定

2. レイアウト メニュー ボタンから [Floorplanning] をクリックし、レイアウト ビューを [Design Analysis] から [Floorplanning] に変更します。
3. [Physical Constraints] ビューをクリックし、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。

4. [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。
5. [Device] ビューの左側で [Set Pblock Size] ボタン  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. 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。
7. 長方形を描画したら、[図 12](#) のように [SLICE] と [RAMB36] をオンにします。
8. [OK] をクリックします。

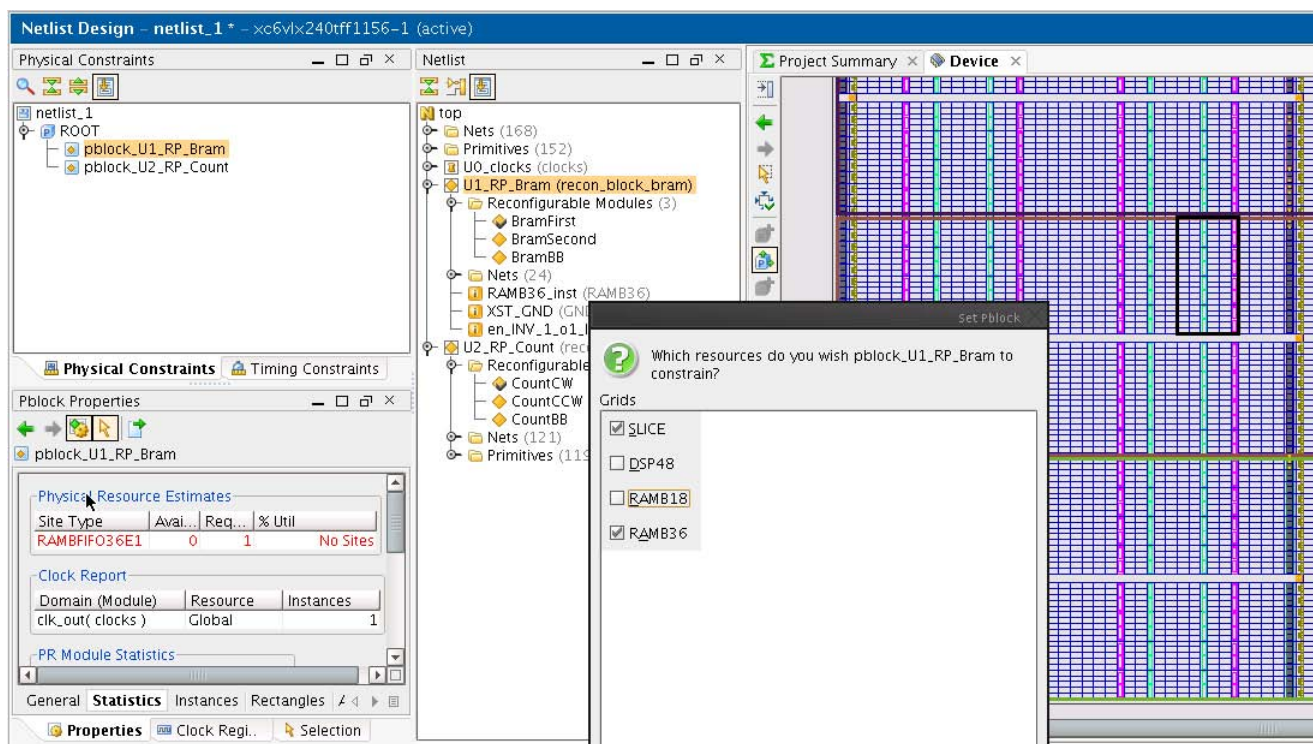


図 12 : Pblock リソース (pblock_U1_RP_Bram) の設定

pblock_U2_RP_Count の AREA_GROUP 範囲を作成

1. pblock_U2_RP_Count を選択します。
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 ピン配置が既に含まれるので、AREA_GROUP 範囲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、図 13 のように [SLICE] と [IOB] リソースをオンにします。
4. [OK] をクリックします。

RM 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して含めることができます。このデザインには、そのようなリソースは含まれません。

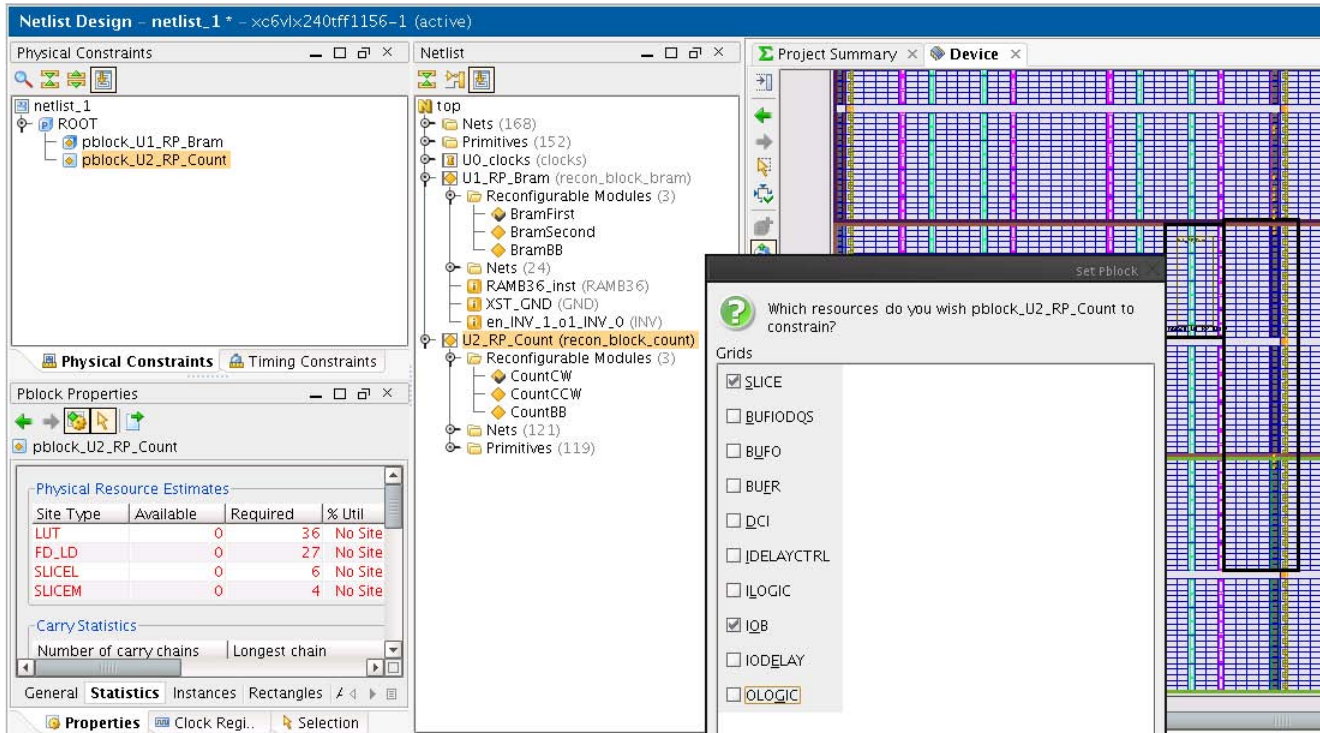


図 13：Pblock リソース (pblock_U2_RP_Count) の設定

5. フロアプランを確認し、必要な変更を加えます。
結果のフロアプランは、図 14 のようになるはずです。

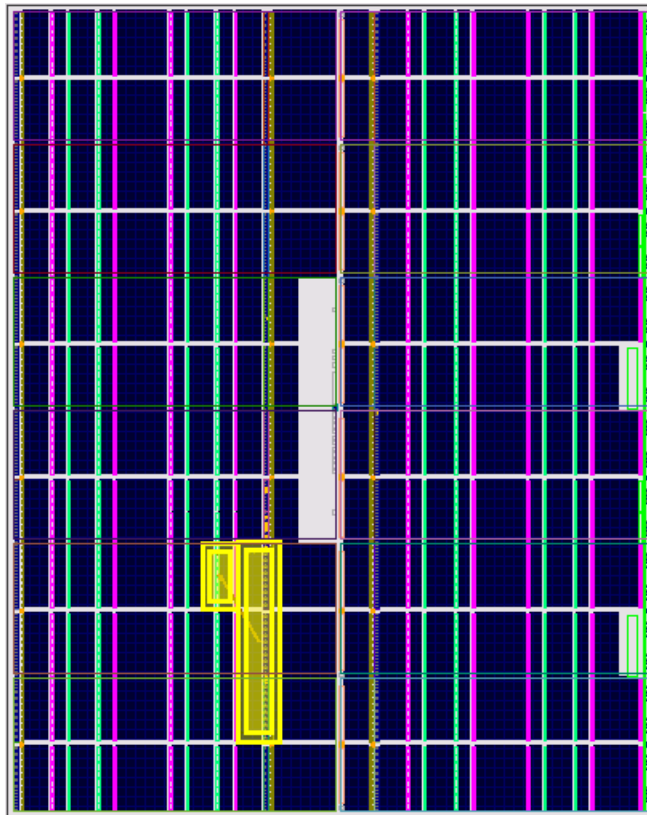


図 14 : 最終的な AREA_GROUP 範囲

6. 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 ピンは、図 14 のように U2_RP_Count 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、リコンフィギュラブル 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、リコンフィギュラブル 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] をクリックし (または F9 を押し)、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。

7. 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、[Statistics] をクリックします (図 13)。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。

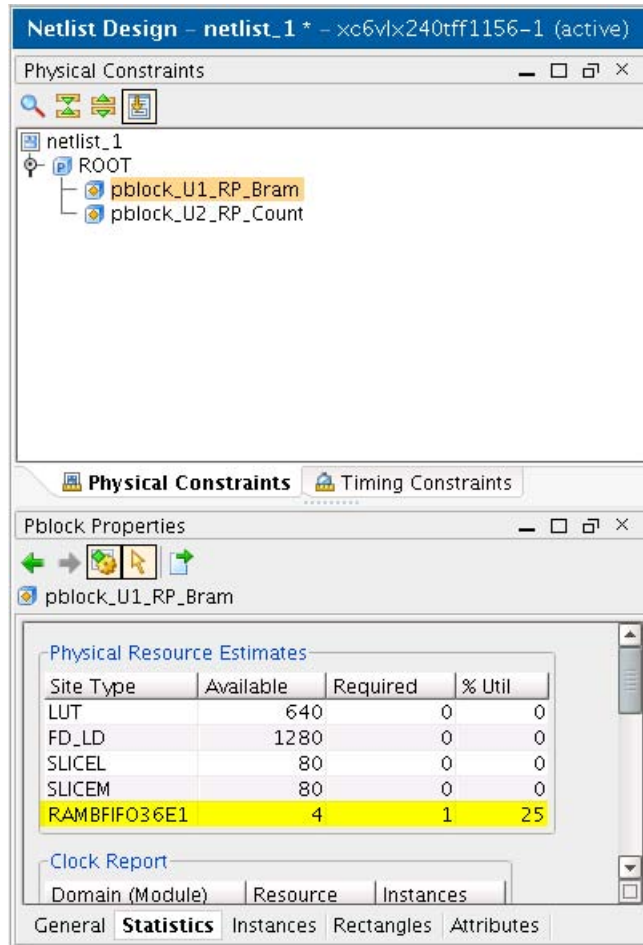


図 15 : Pblock (pblock_U1_RP_Bram) の統計

前の手順で作成した AREA_GROUP 範囲制約は PlanAhead で表示できます。

8. まず [File] → [Save Design] をクリックしてデザインを保存します。

9. 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。

10. [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;
```

11. レイアウト メニューから [Design Analysis] を選択し、PlanAhead をデフォルト ビューに戻します。


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

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 フローのパス マクロに代わるテクノロジーです

現在、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。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ることをお勧めします。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最小限に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、[付録 A「その他のリソース」](#)に示す『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は RP インターフェイスは、さまざまな RM 間で非常に類似しているので、グローバル PERIOD 制約で十分に RP インターフェイスのタイミングを満たすことができます。

PERIOD 制約をデザインに追加

1. [Netlist Design] ビューで [Timing Constraints] タブをクリックします。
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。
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
4. [OK] をクリックし、制約を追加します。
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
 - 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

6. [OK] をクリックし、制約を追加します。[Timing Constraint] ビューは図 16 のようになっています。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。

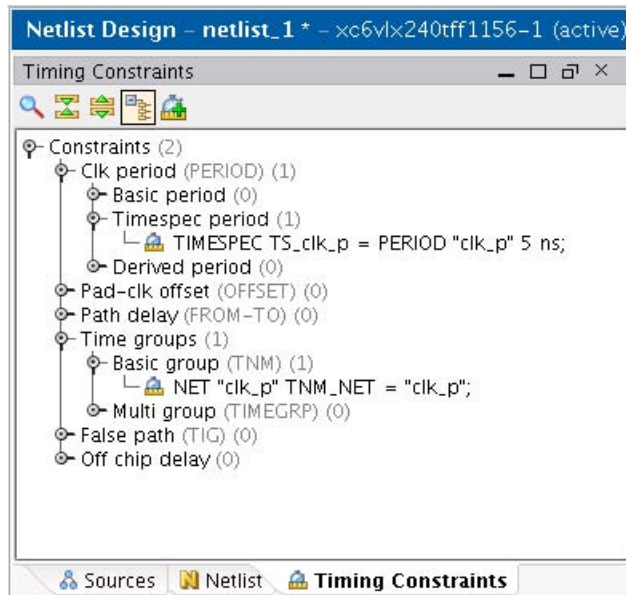


図 16 : [Timing Constraints] ビュー (pblock_U1_RP_Bram)

7. [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";
```

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 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。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、PR 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。

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

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

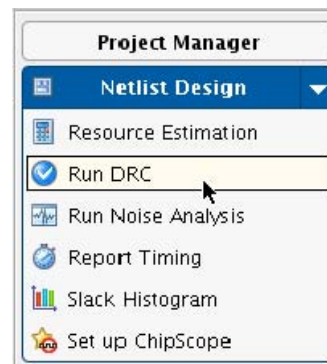


図 17 : [Run DRC] ボタン

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。

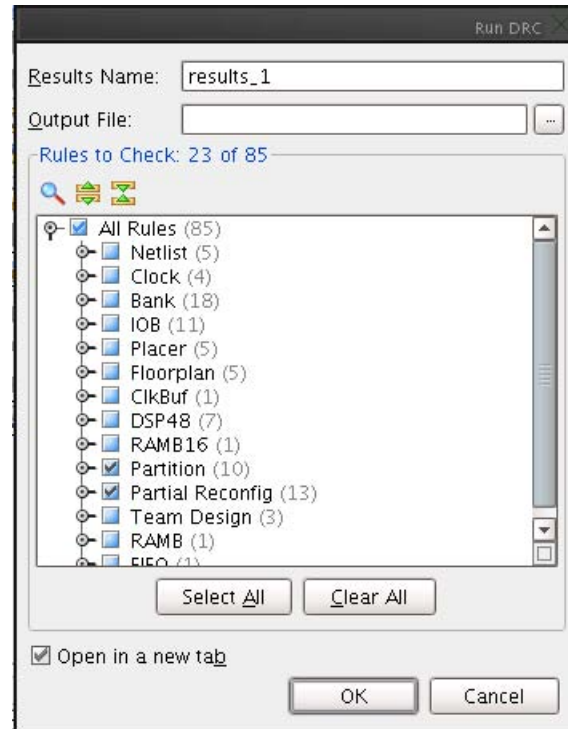


図 18 : [Partition] と [Partial Reconfig] の DRC の指定

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)。

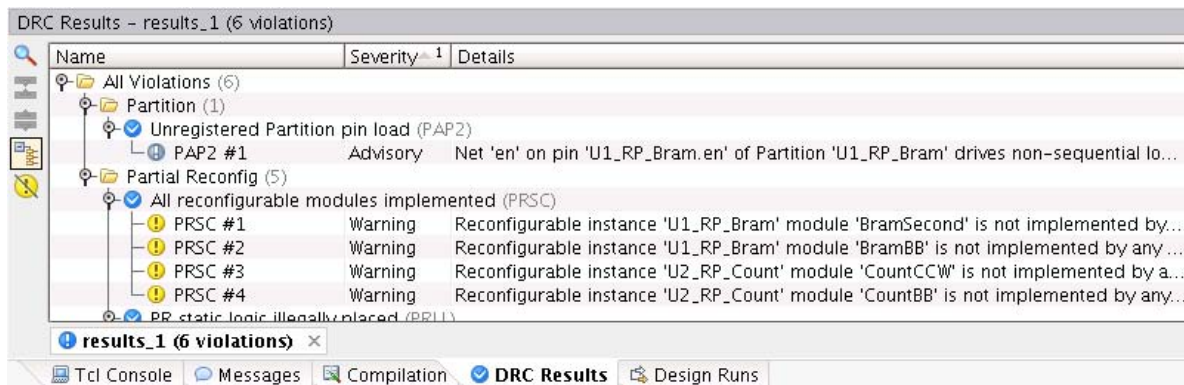


図 19 : [DRC Results] ビュー

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 ファイルといった形式で出力されます。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でのデザインは、次の RM 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
```

まとめ :

- 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 モジュールが含まれます。
- 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 RM は、プロジェクトに RM が追加された順序によって異

なります。これは、各 RP に定義された最初の RM 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にインプリメントされた RM 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があるシステムの場合、すべての RP 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。

config_1 コンフィギュレーションをインプリメント

1. config_1 に設定される RM ramFirst 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 を選択し、
 - b. [Implementation Run Properties] ビューで [Partitions] タブをクリックします (図 20)。
 - c. 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。

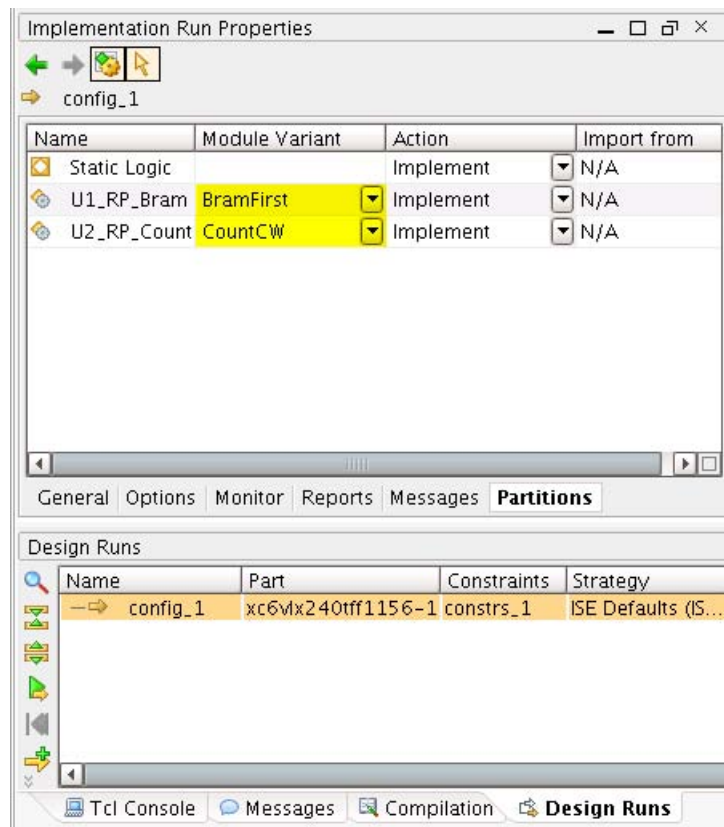


図 20：コンフィギュレーション モジュール (config_1)

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 NGDBuild、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。
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。プロモートするコンフィギュレーションは 1 つだけです。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、このプロモートした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するようにはほかのすべてのコンフィギュレーションで設定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 ロジックになります。

複数コンフィギュレーション間の互換性は、「[手順 11 : コンフィギュレーションの検証](#)」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するとチェックできます。

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

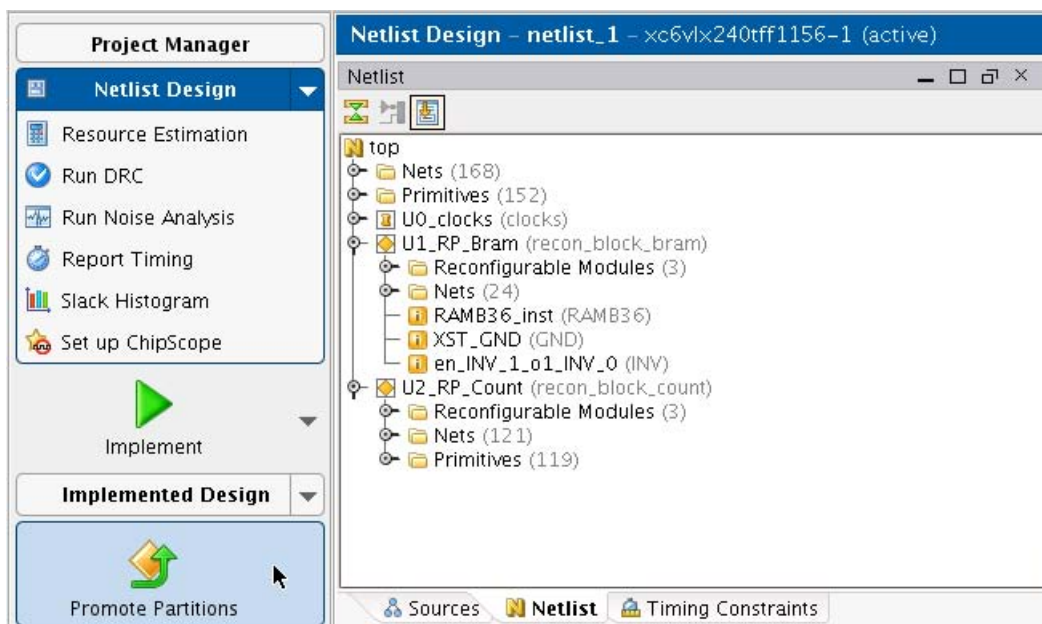


図 21 : [Promote Partitions] ボタン

5. [Promote Partitions] ダイアログ ボックスで [OK] をクリックします。
6. [Configurations] タブをクリックし (開いていない場合は [Window] → [Configurations] をクリック)、ステータスが [Promoted] に変更されていることを確認してください (図 22)。

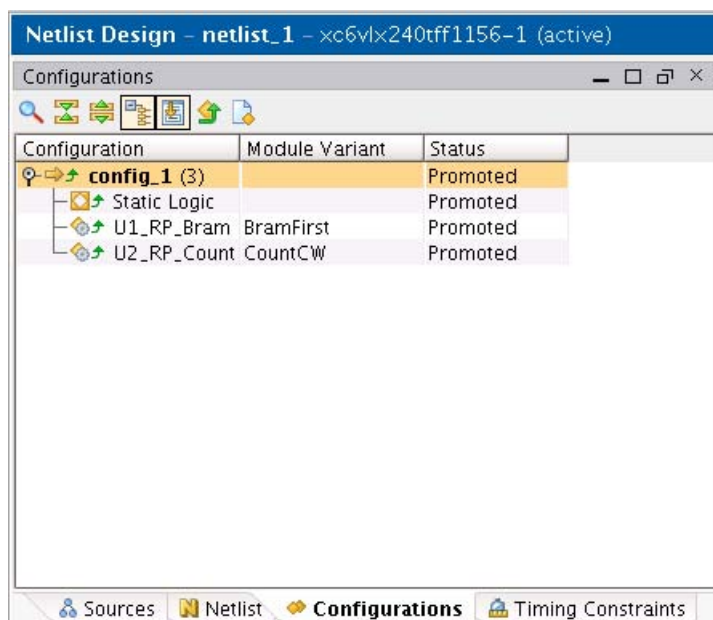


図 22 : [Promoted] ステータス

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

新しいコンフィギュレーションの作成

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

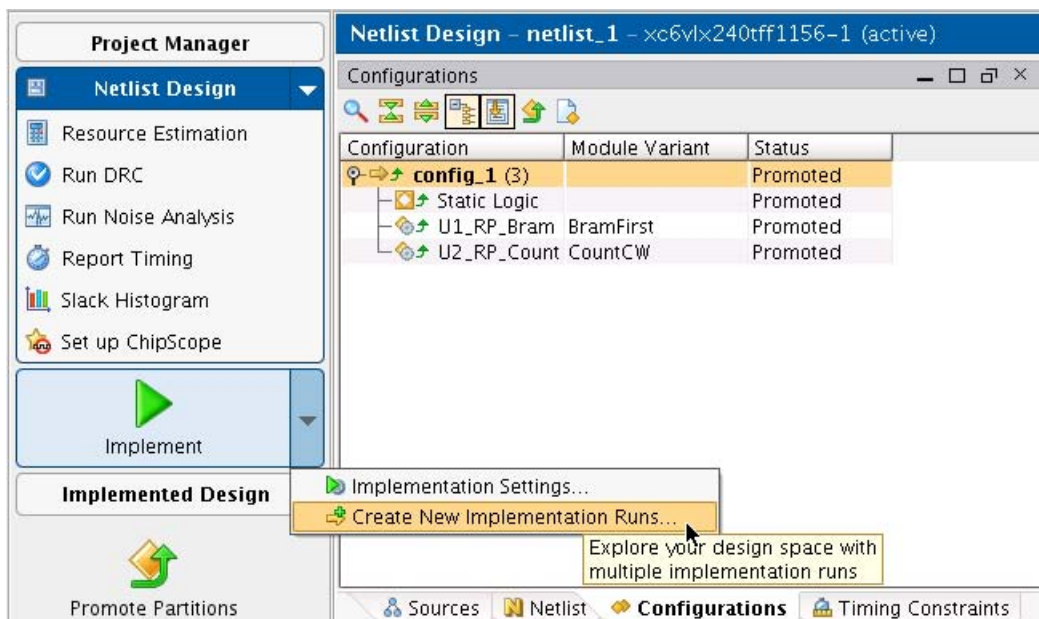



図 23 : 新規インプリメンテーション run の作成

Create Multiple Runs ウィザードで新しいコンフィギュレーションを作成します。

2. 最初のページで [Next] をクリックします。
3. [Setup Implementation Run] 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 (RM) 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。
4. 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。
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)。

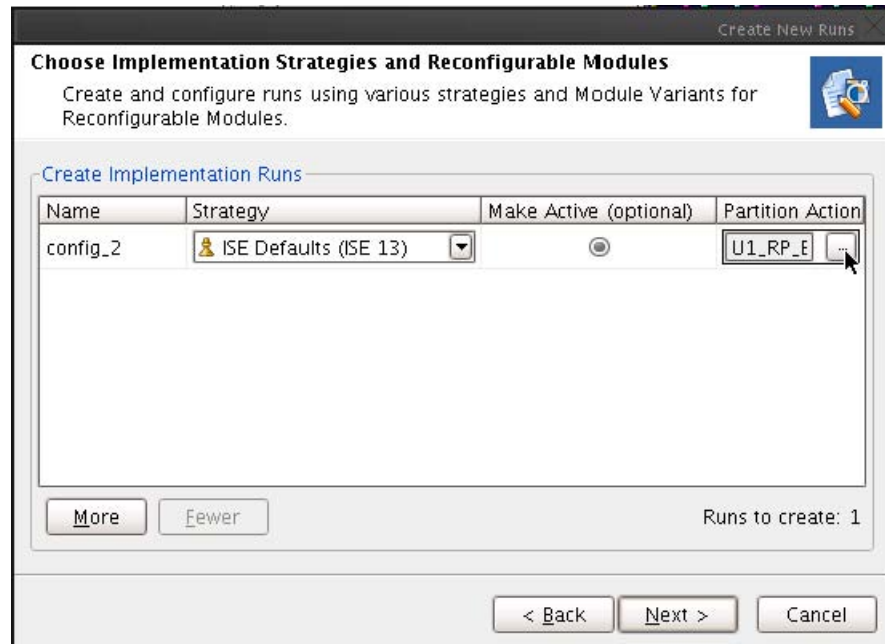


図 24 : インプリメンテーション run の作成

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 RM 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。

6. BramSecond および CountCCW RM 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 RM 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。

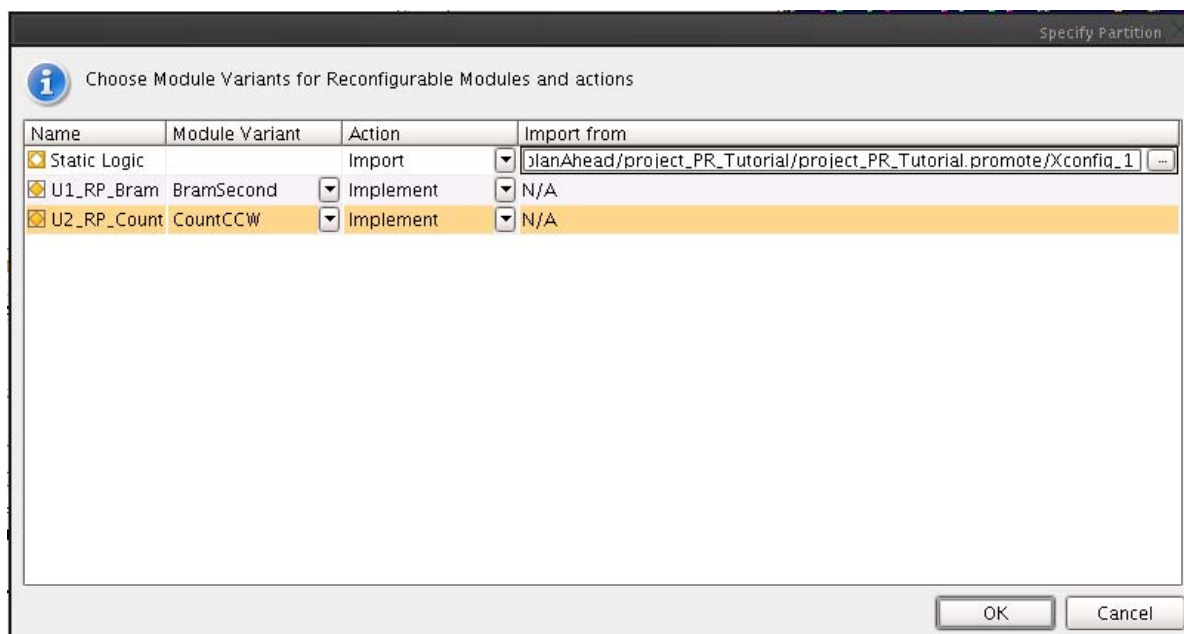


図 25 : [Specify Partition] ダイアログ ボックス (BramSecond/CountCCW)

7. [Promote Partitions] 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。
 - 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. [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。

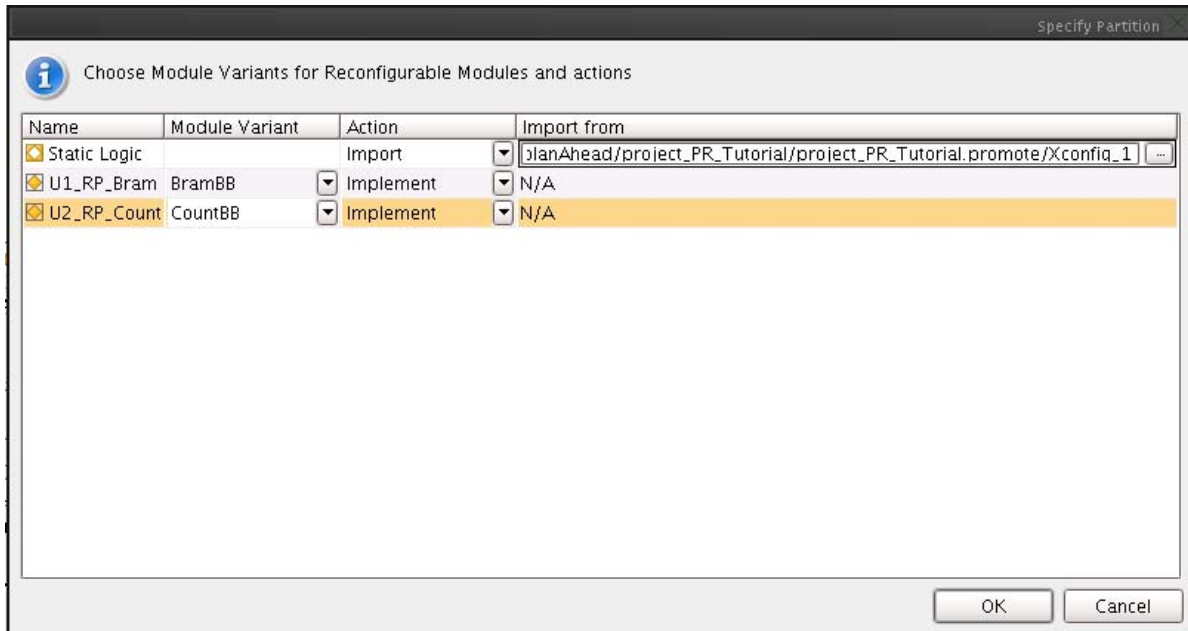


図 26 : [Specify Partition] ダイアログ ボックス (BramBB/CountBB)

9. [Choose Implementation Strategies and Reconfigurable Modules] ページで [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。
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。
11. [Create New Runs Summary] ページで [Finish] をクリックしウィザードを終了します。
[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、
13. [Implementation Run Properties] ビューの [Partitions] タブをクリックし (図 27)、[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。

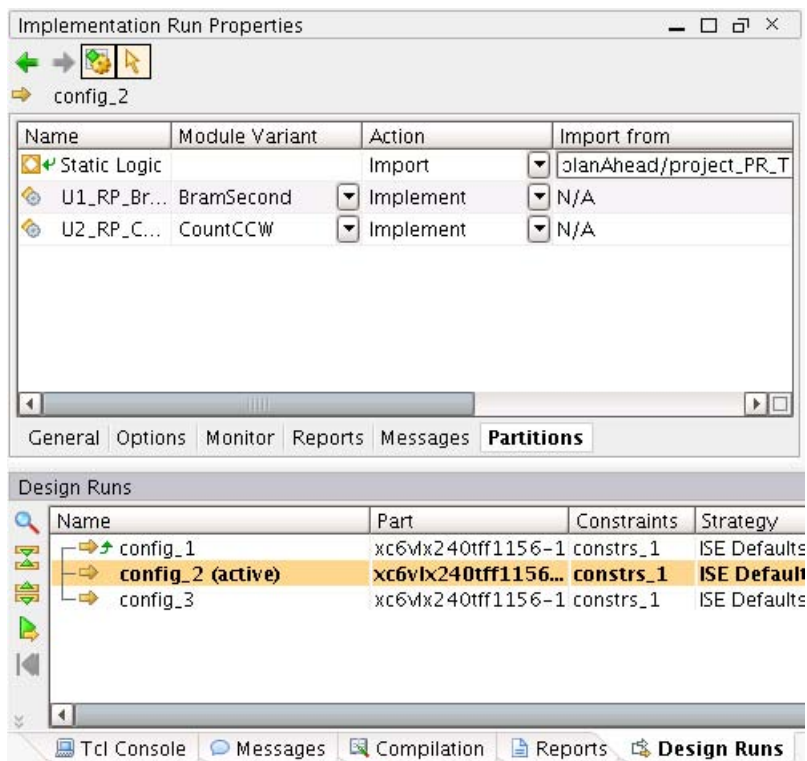


図 27 : コンフィギュレーション設定の確認

14. 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)。

15. [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。

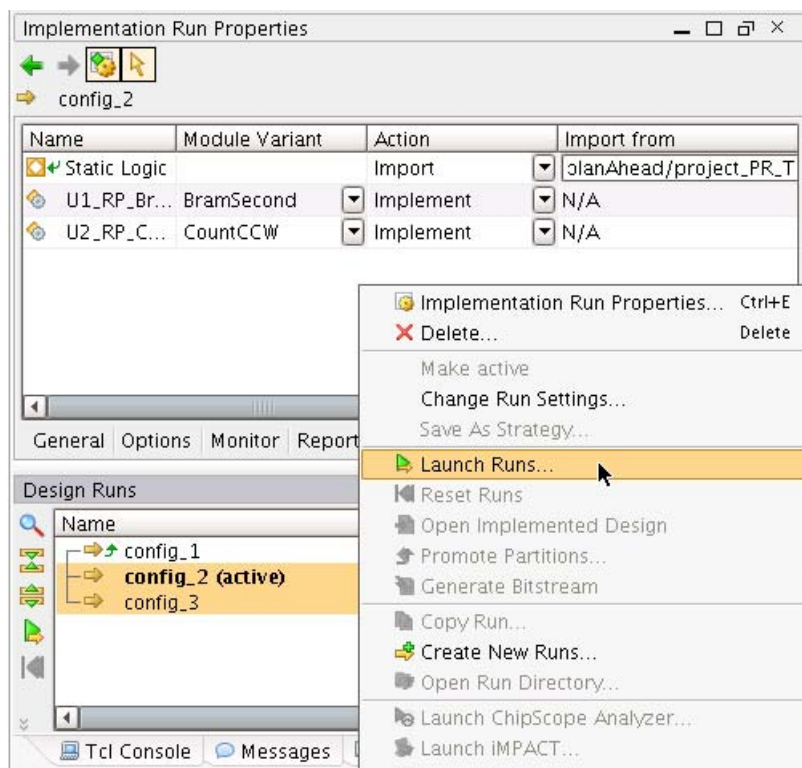


図 28 : run の実行

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。

すべてのコンフィギュレーションで検証を実行

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)。

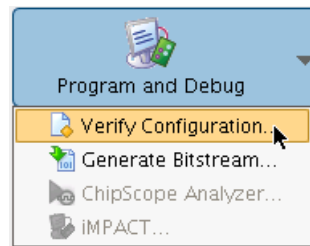


図 29 : コンフィギュレーションの検証

2. 相互に検証するため、2 つ以上のコンフィギュレーションを選択し、[OK] をクリックします。
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ようになったことを意味します。
3. [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、<Extract_Dir>/PlanAhead/project_1/project_1.runs/pr_verify.log に保存されます。

手順 12 : ビット 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。

- 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可能なフル 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さまざまなモジュールのロジックを含む RP 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。どのパーシャル BIT ファイルでも関連する PR 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。

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 RM 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 PR 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。

すべてのコンフィギュレーションの BIT ファイルを生成

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。右クリックし、[Generate Bitstream] をクリックします (図 30)。

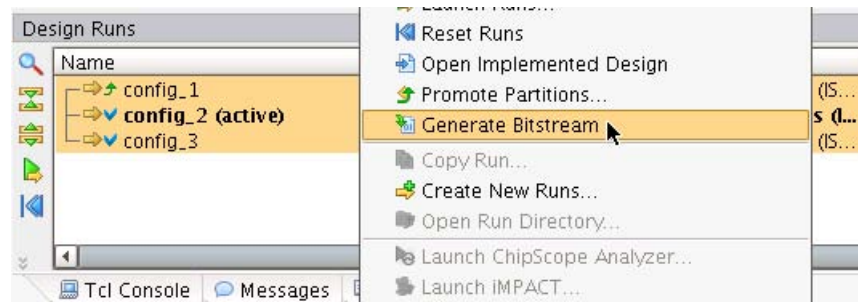


図 30 : BitGen の実行

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。BIT ファイルのダウンロード、検証、デバッグには、iMPACT ソフトウェア ツールを使用できます。

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

4. iMPACT をスタンドアロン モードで起動するには、Flow Navigator の [Program and Debug] ドロップダウン リストから [iMPACT] をクリックします (図 31)。

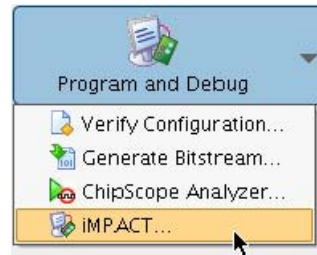



図 31 : iMPACT の起動

5. iMPACT のウィンドウで [Boundary Scan] をダブルクリックし、[Initialize Chain] ボタン  をクリックします。
6. 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て [Assign New Configuration File] をクリックし、次のフル BIT ファイルを指定します。
- `<Extract_Dir>/PlanAhead/<project_name>/<project_name>.runs/config_1/config_1.bit`
7. xc6vlx240t を再び右クリックし、[Program] をクリックします (図 32)。

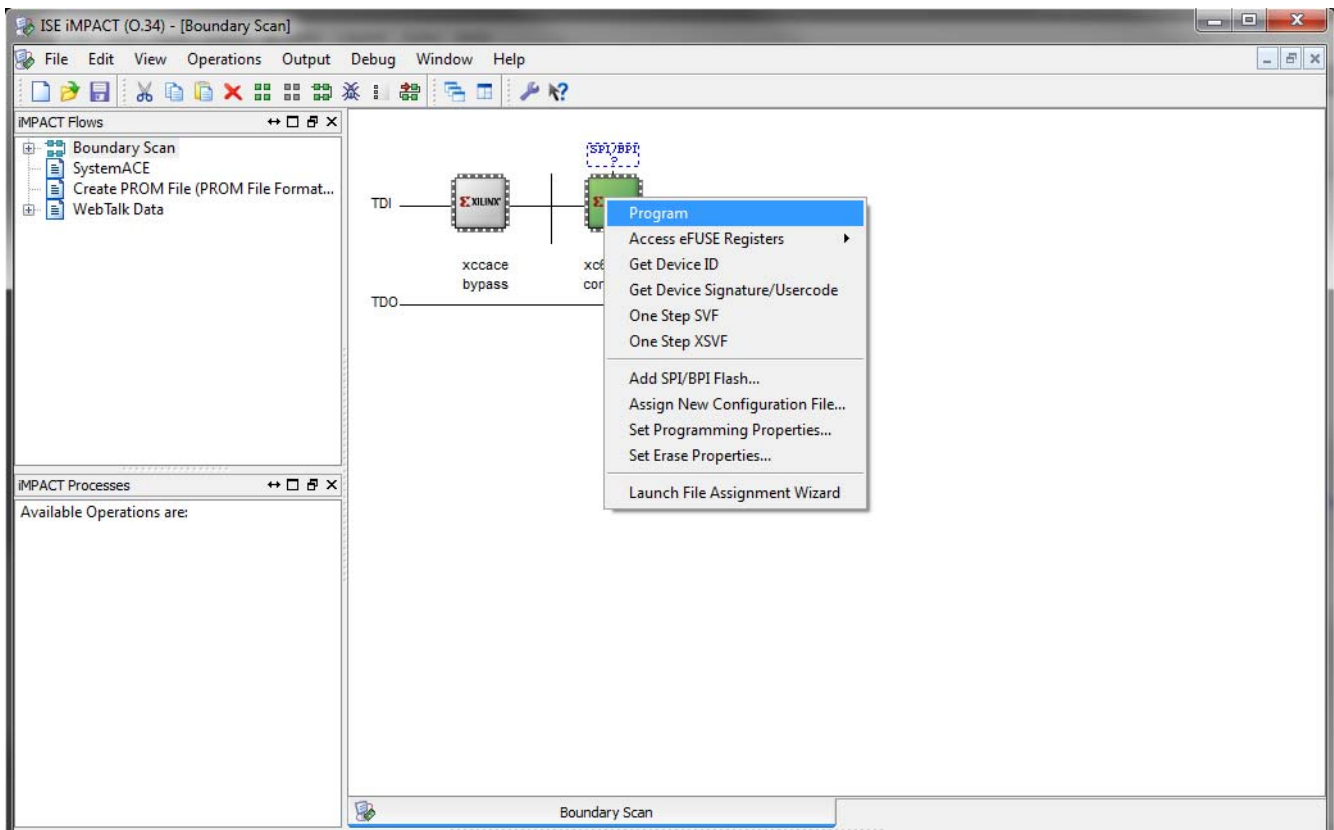


図 32 : iMPACT を使用したデバイスのコンフィギュレーション

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

パーシャル BIT ファイルの関連付け

1. xc6vlx240t デバイスを右クリックして [Assign New Configuration File] をクリックし、次のパーシャル BIT ファイルを指定します。

```
<Extract_Dir>/PlanAhead/<project_name>/<project_name>.runs/config_2/  
config_2_U1_RP_Bram_BramSecond_partial.bit.
```

2. xc6vlx240t 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。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検証し、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。

その他のリソース

ザイリンクス リソース

- 『ISE Design Suite : インストールおよびライセンス ガイド』(UG798) :
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/iil.pdf
- 『ISE Design Suite 13 : リリース ノート ガイド』(UG631) :
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/irn.pdf
- ザイリンクス資料 :
<http://japan.xilinx.com/support/documentation.htm>
- ザイリンクス用語集 :
http://japan.xilinx.com/support/documentation/sw_manuals/glossary.pdf
- ザイリンクス サポート :
<http://japan.xilinx.com/support>
- ビデオ デモ :
http://japan.xilinx.com/products/design_resources/design_tool/resources/index.htm

パーシャル リコンフィギュレーション資料

- 『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) :
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/ug702.pdf
- チュートリアル :
 - 『プロセッサ ペリフェラルのパーシャル リコンフィギュレーション』(UG744) :
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/PlanAhead_Tutorial_Reconfigurable_Peripheral.pdf

PlanAhead 資料

- 『PlanAhead ユーザー ガイド』(UG632) :
http://japan.xilinx.com/support/documentation/sw_manuals/xilinx13_1/PlanAhead_UserGuide.pdf
- PlanAhead 手法ガイド :
http://japan.xilinx.com/support/documentation/dt_planahead_planahead13-1_userguides.htm
- PlanAhead チュートリアル :
http://japan.xilinx.com/support/documentation/dt_planahead_planahead13-1_tutorials.htm

