階層デザイン手法ガイド

UG748 (v12.2) 2010 年 7 月 23 日

XILINX®

本資料は英語版 (v.12.2) 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。

目次

このガイドについて

このガイドでは、ザイリンクス ソフトウェアでのパーティションおよびデザイン保持の手法について説明します。デザイン保持は階層デザイン フローの 1 つです。

このガイドには、次の章が含まれています。

- 第 1 章「パーティションの概要」：ザイリンクス ソフトウェアの階層デザイン フローの基本ブロックであるパーティションの概要を説明します。
- 第 2 章「設計に関する考慮事項」：デザインの論理および物理レイアウト、HDL コーディングのガイドライン、フロアプランの必要性など、パーティションおよび階層デザイン フローを使用して設計する際のさまざまな注意事項を示します。
- 第 3 章「合成パーティションフロー」：デザインでパーティションを使用する際のインクリメンタル合成およびボトムアップ合成について説明し、サポートされる合成ツールを示します。
- 第 4 章「コマンド ラインでのパーティション フロー」：ISE® コマンド ライン ツールでの合成後のデザインフローにおけるパーティションの設定および使用方法を示します。
- 第 5 章「PlanAhead でのパーティション フロー」：PlanAhead™ ソフトウェア ツールでの合成後のデザインフローにおけるパーティションの設定および使用方法を示します。
- 第 6 章「デザイン保持フロー」：ISE コマンド ライン ツールおよび PlanAhead ソフトウェアを使用した RTL フローおよびネットリスト フローでデザイン保持を使用する方法を示します。

このガイドには、次の付録が含まれています。

- 付録 A「パーティションのデバッグ」：デザイン保持フローで発生するエラーのデバッグおよびトラブルシューティング方法を示します。既知の問題およびソリューションは、アンサー #35019 を参照してください。このアンサーへのリンクは、「その他のリソース」にあります。

メモ：このガイドを参照する前に、PlanAhead ソフトウェアの基本的な機能に慣れておくことをお勧めします。PlanAhead ソフトウェアの詳細は、次を参照してください。

- 『PlanAhead ユーザー ガイド』(UG632)

  http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/PlanAhead_UserGuide.pdf
- PlanAhead チュートリアル

  http://japan.xilinx.com/support/documentation/dt_planahead_planahead12-2_tutorials.htm

階層デザインの概要

階層デザイン (HD) は、デザインの論理階層を利用したデザイン手法で、フラット デザイン フローでの制限を克服します。フラット フローではデザイン全体を一度に最適化できるという利点がありますが、階層デザインではフラット フローでは使用できない機能を使用できます。階層デザインを使用すると、合成およびインプリメンテーションの実行時間および実行回数を削減でき、タイミング要件を満たして保持するのが困難であったり、一貫したインプリメンテーション結果が得られないといった問題を解決できます。階層デザインでは、デザインを小型のロジック ブロックに分割できるので、主要な機能を個別に開発できます。これらの機能は FPGA の設計では新しいものであり、新しい設計フローを使用することが可能となります。

階層デザイン フロー

現在のところ、ソフトウェアでは階層デザイン フローの次の機能がサポートされています。

- **デザイン保持** : デザインを小型のモジュールに分割し、個々のモジュールが完成しタイミングが満たされたら、その結果を保持します。
- **パーシャル リコンフィギュレーション** : FPGA のユーザー定義の部分を動作中にリコンフィギュレーションし、デザインのほかの部分に影響を与えずにロジックを変更、消費電力を削減、リソースを節約することが可能です。
- SCC (Single Chip Cryptography) : デザインの厳しいセキュリティ要件を、複数の FPGA に適用するのではなく、1 つの FPGA に適用します。これは、パーシャル リコンフィギュレーション ソリューションおよび場合によってデザイン保持を利用したフローを使用することにより達成します。

今後のザイリンクス ISE Design Suite ソフトウェアには、次の階層デザイン フローも含まれる予定です。

- チーム デザイン : デザインのコード、インプリメンテーション、検証をチーム メンバーに分割する方法です。各チーム メンバーのブロックを最終的なインプリメンテーションにインポートすることにより、最終的なデザインを作成します。この際、各ブロックの配置配線は保持されます。
- IP 再利用 : ザイリンクス IP、サードパーティ IP、またはユーザー IP を、検証済みの配置配線結果を使用してデザインにインポートします。IP のインプリメント、タイミング検証、ファンクション検証を再実行する必要はありません。

その他のリソース

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。

SCC フローの詳細は、SCC サイト http://japan.xilinx.com/member/crypto/index.htm を参照してください。この情報およびマニュアルを入手するには、まず Web サイトで登録をする必要があります。

PlanAhead ソフトウェアの詳細は、次の資料を参照してください。

- 『PlanAhead ユーザー ガイド』(UG632)
  http://japan.xilinx.com/support/documentation/sw_manuals/xilinx12_2/PlanAhead_UserGuide.pdf
- PlanAhead チュートリアル
  http://japan.xilinx.com/support/documentation/dt_planahead_planahead12-2_tutorials.htm

PlanAhead に関する一般的な情報、オンライン デモ、ホワイト ペーパーは、次を参照してください。

http://japan.xilinx.com/planahead

既知の問題およびその回避方法については、アンサー #35019 を参照してください。

http://japan.xilinx.com/support/answers/35019.htm

XILINX®

第 1 章

パーティションの概要

この章には、次のセクションが含まれています。

階層デザイン フローでは、デザインを小型のブロックに分割します。これらのブロックは「パーティション」と呼ばれ、ザイリンクス ソフトウェアの階層デザイン フローの基本ブロックとなります。複雑なデザインを小型の作業しやすい部分に分割するため、パーティションを使用して階層の境界を定義します。パーティションにより階層モジュールの周囲に境界が作成され、デザインのほかの部分と分離されます。インプリメントおよびエクスポートしたパーティションは、単純な切り取り/貼り付け操作によりデザインに挿入することにより、そのモジュール インスタンスの配置配線結果を保持します。

パーティションの定義および制御は、`xpartition.pxml` (PXML) ファイルで指定します。ツールを実行すると、このファイルが読み込まれます。PXML ファイルは、PlanAhead™ ソフトウェア グラフィカル ユーザー インターフェイスを使用して作成するか、提供されている PXML テンプレートを使用するかまたは使用せずに手動で作成できます。XPML ファイルの作成については、第 4 章「コマンド ラインでのパーティション フロー」を参照してください。

12.x リリースでは、次の FPGA アーキテクチャでパーティションがサポートされています。

- Spartan®-3 プラットフォーム
- Spartan-3E プラットフォーム
- Spartan-3L プラットフォーム
- Spartan-3A プラットフォーム
- Spartan-3A DSP プラットフォーム
- Spartan-6 ファミリ
- Virtex®-4 ファミリ
- Virtex-5 ファミリ
- Virtex-6 ファミリ

パーティションのステート

パーティションは、そのステートによってインプリメントまたはインポートできます。パーティションを初めて ISE® Design Suite インプリメンテーション ツール (NGDBuild、MAP、PAR など) で実行する場合、パーティションのステートを implement に設定する必要があります。インプリメンテーションが完了したら、パーティションをエクスポートし、今後の実行で結果をインポートできます。ただし、エクスポートされた結果は、内部ロジックおよびパーティション インターフェイスが変更されていない場合にのみ今後のインプリメンテーションで使用可能です。

パーティション モジュールに変更を加えた場合は、そのパーティションの配置配線をアップデートする必要があります。変更されたパーティションを再インプリメントし、変更されなかったパーティションは以前の実行結果をインポートできます。パーティションをアップデートする必要のある変更は、次のとおりです。

- HDL コードの変更、またはパーティションに関連したネットリストが変更される変更
- AREA_GROUP、LOC 制約など、パーティションに関連する物理制約の変更
- デバイス、パッケージ、スピード グレードなど、ターゲット アーキテクチャの変更
- パーティションに接続された ChipScope™ Analyzer コアの追加および接続の変更

エクスポートされたパーティションのアップデートが必要な場合は、パーティションのステートを正しく管理するため、PXML ファイルで State 属性を変更します。パーティションのステートを正しく管理しないと、インプリメンテーション ツールでエラーが発生します。

パーティションを再インプリメントする必要のない変更は、次のとおりです。

- TIMESPEC など、ロジックの物理的なロケーションに影響しない制約の変更
- par –xe など、インプリメンテーション オプションの変更

パーティションの保持レベル

パーティションの主な目的は、以前の結果をインポートすることにより実行結果を保持することです。保持レベルは指定できます。デフォルトでは、パーティションをインポートしたときに配置配線が 100% 保持されます。

このデフォルト設定を、配置結果のみ (配線は保持されない)、合成結果のみ (配置配線は保持されない) を保持するよう変更できます。パーティションをインポートすると、保持レベルにかかわらず、配置配線を含むすべてのインプリメンテーション情報もインポートされます。保持レベルは、インプリメンテーション ツールで結果を向上するためにインポートされた配置配線をどれだけ変更できるかを指定するものです。一部のパーティションで保持レベルを緩和するとデバイス リソースが解放され、インプリメンテーション ツールでほかのパーティションをより柔軟に配置配線できるようになります。保持レベルはパーティションごとに設定でき、インポートされたパーティションにのみ適用されます。

タイミング クリティカルなパーティション モジュールでタイミングが満たされ、変更する予定がない場合は (IP コアなど)、配置までを保持するのが適切です。パーティションがタイミング クリティカルでなく、タイミングが満たされていない場合は、保持レベルを緩和してツールでソリューションがより柔軟に検索されるようにします。

パーティションを使用する目的が検証時間の短縮である場合は、常に保持レベルを配線に設定してください。デザインのほかの部分のタイミングを満たすため、または配線を完了するために保持レベルを変更する必要がある場合は、パーティションを再検証する必要があります。

パーティションをフロアプランすると、配置レベルを緩和する必要がなくなることがあります。

インポート先ディレクトリ

パーティションをインポートする際は、エクスポートする結果の場所を指定する必要があります。インプリメントされたデザインをエクスポートすると、デザインに含まれるすべてのパーティションが自動的にエクスポートされます。可能な限り、1 つのエクスポート ディレクトリを使用し、そこからインポートすることをお勧めします。パーティションをデザインにインポートすると、関連するデザインがメモリで開きます。複数のデザイン実行 (複数の場所) からのパーティションをインポートすると、すべてのデザインがメモリで開くため、メモリの使用量が増加し、インプリメンテーション ツールのランタイムも長くなります。これにより配線の競合が発生する可能性が高くなり、一部のネットの再配線が必要となる場合があります。

パーティションを使用する状況の判断

パーティションは、パーティションが必要なモジュールにのみ使用してください。パーティションを過剰に使用すると、ランタイムおよびパフォーマンスが悪化する可能性があります。モジュールが独立したファンクション ブロックではなく、個別の階層を持たない場合、ほかのブロックとのグローバル最適化が有益な場合は、フラット最適化によりよい結果が得られます。

パーティションを適切に使用するには、第 2 章「設計に関する考慮事項」のガイドラインに従ってください。

パーティションの使用に適しているのは、次のものです。

- DSP モジュールまたは EDK システムなどのファンクション ブロック
- 高パフォーマンス コア
- デバイスに一緒に配置する必要のあるロジックを含むインスタンス
- デザイン ガイドラインに従ったモジュール

パーティション使用の利点と欠点

階層フローを使用するのには複数の利点がありますが、欠点もあります。パーティションを使用する主な欠点は、パーティション階層の境界が最適化に影響するということです。パーティションの境界を越えた最適化は実行されないので、その点を考慮してデザインを設計しないと、パーティションを追加することによりタイミング、リソース使用量、ランタイムに大きく影響します。注意深く設計しても、最適化および配置に関するほかの制限もあるので、リソース使用量が増加し、タイミングが悪化する場合があります。適切に設計されたデザインではこれらの影響は最小限に抑えられますが、これらの制限を念頭に置いておくことは重要です。

パーティションの最適化への影響およびこれらの影響を最小限に抑える設計方法については、次の章を参照してください。

XILINX®

第 2 章

設計に関する考慮事項

階層デザイン フローを使用するかどうかは、タイミング クロージャや結果に一貫性がないなどの問題が発生してからではなく、設計の初期段階で決定します。階層デザイン フローの利点を最大限に活用するには、最初に考慮すべき事項が多数あります。考慮事項には、デザインの論理および物理レイアウト、HDL コーディング ガイドライン、フロアプラン制約の仕様などがあります。これらについて詳細に説明します。

この章には、次のセクションが含まれています。

最適化の制限

デザインでパーティションを使用すると、最適化が制限されることに注意してください。このセクションでは、パーティションを使用することによる最適化の制限をリストします。デザインにパーティションを 1 つ追加しただけで、すべてのインスタンスがパーティションの一部になります。パーティションとして指定されていないインスタンスは、最上位パーティションの一部となります。次に、最適化の制限をリストし、各制限がデザインのロジックに与える影響の例、およびこれらの影響を回避する方法または最小限に抑える方法を示します。

パーティションの境界を越えた最適化は実行されない

この制限には、親パーティションと子パーティションの間、子パーティション間の最適化も含まれます。この制限は、フラット デザインと比較して、タイミングとリソース使用量に影響します。

組み合わせロジックを駆動する組み合わせロジックでロジック間にパーティションの境界があると、このロジックは最適化されません。

2 つのパーティションに共通のロジックがあっても、リソースは共有されません。4 つのパーティションに供給される制御バスに対して 4 つのパーティションすべてに同じデコード ロジックがある場合、このロジックが 4 つ作成されます。フラット フローでは、このロジックは共有されます。同様に、複数のパーティションにこのバスのレジスタがある場合、複数のレジスタ セットが推論されます。フラット フローでは、1 つのレジスタ セットに最適化され、供給されます。

パーティションの入力に供給される定数

最適化で削除されることを目的としてパーティションの入力を定数値に固定している場合、この定数はパーティションの境界を越えてパーティションに挿入できないので、最適化は実行されません。

この状態は、コアまたはモジュールの特定の機能をイネーブルまたはディスエーブルにするために定数を使用している場合に発生します。モジュールのロジックをポートを介して制御することはお勧めしません。このような方法はパーティションでは機能しません。パラメータ/属性を使用するか、パッケージ ファイルを含めてください。

パーティションの未接続の出力は最適化されない

パーティションの出力が何も駆動していない場合、ソース ロジックはフラット フローでのように最適化されません。

1 つのパーティションのロジックを別のパーティションのロジックにパックできない

この制限は、フリップフロップと LUT の比率が大きく異なる場合に影響します。

パーティション内の組み合わせロジックが最終的にフリップフロップに達する出力を駆動している場合、LUT はフリップフロップと共にパックできません。

パーティションを使用するインスタンスの制限

一部のインスタンスでは、パーティションはサポートされません。次のような制限があります。

- 個別の HDL ファイルで定義されていないモジュール/エンティティのインスタンスには、パーティションを使用できません。
- インスタンス名が変更する可能性のあるインスタンスには、パーティションを使用できません (インスタンス名がパラメータまたはジェネレート文に基づいている場合)。

デザインの構造

デザインをフラット フローを使用してインプリメントすると、合成ツールおよびインプリメンテーション ツールでデザイン全体をスピードおよびエリアを優先して最適化できます。デザインは階層の境界を越えて最適化されるので、デザインの論理レイアウトはそれほど重要ではありません。デザインを階層フローでインプリメントする場合、ロジックが分離されるので、パーティションが最適化の壁となり、デザインに大きく影響します。図 1 のようなデザインがあるとします。

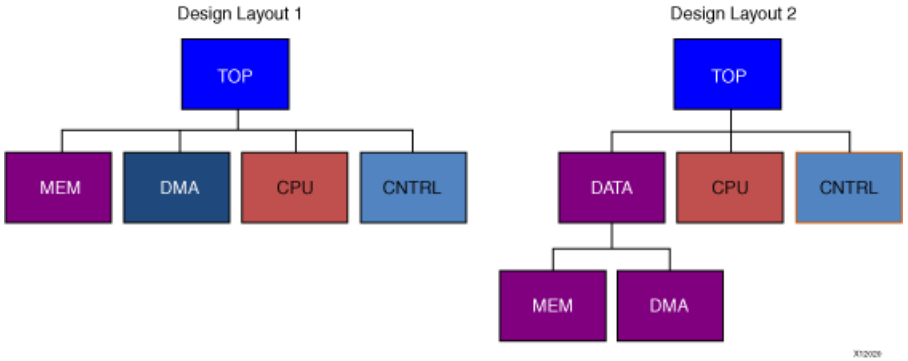

図 1 : デザイン階層の例

左側のデザイン レイアウトには、同じ階層レベルに MEM および DMA というモジュールがあります。TOP の下にあるすべてのモジュールにパーティションを追加すると、MEM と DMA の間では最適化は実行されません。これらの 2 つのモジュールに関連するロジックが多数ある場合、フラット フローでは最適化が実行されますが、階層デザイン フローでは最適化は実行されず、リソース使用量が増加し、よいタイミング結果が得られない可能性があります。デザイン階層を右側のレイアウトのように変更し、共有するロジックを含むモジュールを 1 つのパーティション DATA の下にグループ化すると、フラット フローと同様に MEM と DMA の間で最適化が実行されます。

HDL に関する注意事項

階層デザイン フローを使用するかどうかは、デザインの構築、モジュール インターフェイスの定義、およびモジュール コードの記述の前に決定する必要があります。階層デザイン フローは、フラット フローでタイミング問題が発生した後に使用するタイミング クロージャ手法ではありません。階層デザイン フローの利点を活かすため、コーディングに関するガイドラインおよび推奨事項があります。

パーティションを使用するとモジュールが分離され、ソフトウェアで境界を越えた最適化は実行されなくなります。この問題を回避してパフォーマンスを向上するには、次のガイドラインに従う必要があります。

- 入力ポートおよび出力ポートにレジスタを付けます。
- パーティションの内外にあるロジックを駆動するネットを管理します。
- ファンアウトの大きいネットを管理します。
- パーティションの入力として定数を使用しないようにします。
- 入力ポートおよび出力ポートを未接続のままのしないようにします。

入力ポートおよび出力ポートにレジスタを付ける

入力および出力にレジスタを付けることは非常に重要です。パーティションの境界を越えた最適化は実行されないので、パーティションの入力および出力でタイミングの問題が発生しやすくなります。入力および出力にレジスタを付けると、ツールでパーティション内のパスに焦点を置いた処理が可能となり、モジュールのタイミングを確実に保持できます。パーティションの境界を横切るネットは、ネットに接続されているパーティションすべてをインポートしない限り保持されません。そのため、タイミング クリティカル ネットがこの境界を横切る場合、インポートした場合でもパーティションでタイミング違反が発生する可能性があります。このネットにレジスタが付いていれば、パーティション内でタイミング クリティカルな変更が発生するのを回避できます。

パーティション内外のネットを管理する

パーティション内で、1 つのネットを内部ネットおよび出力ネットの両方として使用しないでください。ネットをパーティション内および出力ポートとして使用する必要がある場合は、ネットのソースを複製し、1 つのネットを内部ネットとして、もう 1 つのネットを出力ポートとして使用します。

ファンアウトの大きいネットを管理する

パーティション モジュールの出力のファンアウトも考慮する必要があります。パーティションの出力のファンアウトが大きく、デザインの複数のエリアに接続される場合、ドライバを複製する必要がある場合もあります。フラット フローでは、この複製は自動的に実行されますが、階層デザイン フローでは手動で複製する必要があります。

パーティションの入力として定数を使用しない

IP コアなどのネットリスト モジュールには、ロジックの不要な部分がマップの最適化で削除されることを前提として設計されているものがあります。このような IP では、コアの特定のロジックをイネーブルまたはディスエーブルにするために、入力を定数に接続します。これにより、IP をネットリストとして提供すると同時に、多少のカスタマイズも可能になります。ただし、パーティションでは境界を越えた最適化が実行されないので、この方法は機能しません。ポートに定数を接続することにより最適化でロジックが削除されることを前提とした EDIF/NGC コアにパーティションを直接追加すると、ロジックの最適化は実行されず、最適な結果は得られません。このように動作するコアにパーティションを追加する必要がある場合は、EDIF/NGC コアを含む HDL ラッパを追加し、ラッパにパーティションを配置します。HDL ラッパのポート リストには、IP コアをデザインの残りの部分に接続するのに必要な I/O のみを含めるようにし、定数の割り当てはラッパ ロジック内に残します。パーティションを IP コアの上のレベルで定義することにより、マップで定数をコアにトレースすることが可能となり、必要な最適化が実行されます。

ポートを未接続のままにしない

未接続のネットでも、同じような最適化の問題が発生します。パーティションの出力を未接続のままにすると、ソースのないこのネットに接続されたドライバがマップの最適化で削除されません。このロジックを HDL コードで削除できない場合は、パーティションを含むラッパを作成するとロジックがマップの最適化で削除されます。

同様に、ロジックで駆動されておらず、パーティションの外部のロジックに接続されいてるパーティションの出力も、エラーの原因となります。この場合、インプリメンテーション ツールでパーティションの出力が駆動されていないことを検出できず、部分的なネットが配線されます。これは無効であり、BitGen で DRC エラーが発生します。不要なパーティション ポートをコードから削除し、パーティションを再インプリメントしてください。

インポートの制限

すべての配線情報を保持する必要はありません。パーティションの配線が再配線される可能性があるのは、次のような場合です。

- インポートされたパーティションの保持レベルが、配線ではなく配置または合成に設定されている。
- 下位パーティションの I/O バッファがインポートされ、接続されているパッドがインプリメントされる親パーティションに含まれている。ただし、この配線は専用であるため、厳密にいえば保持されませんが、変更されません。
- 下位パーティションのフリップフロップがインポートされ、インプリメントされる親パーティションに含まれている I/O バッファに接続されている。配線は専用であり、フリップフロップが I/O ロジックにパックされれば変更されません。フリップフロップをパーティションの境界を越えて引き込み、I/O ロジックにパックするため、IOB=FORCE または IOB=TRUE 制約を設定する必要があります。フリップフロップがスライスにパックされている場合、配線は保持されず、タイミングが満たされるとは限りません。IOB=FORCE を使用すると、フリップフロップが I/O ロジックに正しくパックされなかった場合にエラーが表示され、IOB=TRUE を使用すると警告が表示されます。
- 下位パーティションの LUT がインポートされ、インプリメントされる親パーティションに含まれている I/O バッファに接続されている。LUT はスライス ロジックにパックされる必要があります。配線は専用ではなく、タイミングが満たされるとは限りません。
- デザインに PWR/GND ネットが含まれ、最上位パーティションがインプリメントされる。PWR/GND ネットは常にインプリメントされ、子パーティションに含まれていても、最上位パーティションと共にインポートされます。

パーティションのフロアプラン

フロアプランとは、制約を使用してデザインの配置を制御する手法で、このセクションでは AREA_GROUP 制約を使用することを指します。

パーティション デザインで AREA_GROUP 制約を使用するのに制限はありませんが、CLB 境界にスライス範囲を作成することをお勧めします。これにより、配置および配線のリソースを最大限に利用できます。スライス範囲が CLB 境界上にあるかどうかを検証するには、制約の XY 座標を確認します。XY 座標が X0Y0 ～ X3Y9 のように偶数で始まり奇数で終わっていれば、範囲は CLB 境界上にあります。AREA_GROUP 制約は PlanAhead™ ソフトウェアを使用して作成できます。この場合、制約は CLB 境界上に自動的に設定されます。CLB または FPGA のその他のブロックの詳細は、デバイスのデータシートを参照してください。

http://japan.xilinx.com/support/documentation/data_sheets.htm

フロアプランをパーティションで使用すると、パーティションに関連するすべてのロジックをデバイスの 1 つのエリアに保持できるという利点があります。フロアプランで各パーティションを配置配線する領域を作成すると、インポートの際に配線の競合が発生する可能性を最小限に抑えることができます。また、後で追加するロジック用に FPGA のその他の部分を予約するのにも便利な方法です。

パーティションは PlanAhead ツールでサポートされていますが、コマンド ライン ツールを使用してパーティション デザインを実行することも可能です。この場合、PlanAhead を使用してパーティションの設定、デザインのフロアプランを実行し、ISE® コマンド ライン フローを使用して `xpartition.pxml` ファイルを作成できます。サポートされるフローは、第 4 章「コマンド ラインでのパーティション フロー」を参照してください。

デザイン保持

デザイン保持パーティションでは AREA_GROUP 制約は必要ありませんが、デザインによってはフロアプランによりランタイムとタイミング結果が向上します。また、AREA_GROUP 制約を使用すると、インポートの際に配線の競合が発生する可能性を最小限に抑えることができます。

XILINX®

第 3 章

合成パーティションフロー

この章には、次のセクションが含まれています。

合成パーティション フローの概要

階層デザイン フローでは、1 つのエリアの変更により別のエリアの合成結果が変更されないようにするため、各パーティションを個別に合成する必要があります。これには、インクリメンタル合成手法またはボトムアップ合成手法を使用します。インクリメンタル合成手法は、ほとんどのサードパーティ合成ツールでサポートされており、RTL モジュールをパーティションとして指定できます。合成を実行すると、各パーティションが個別に合成され、1 つのモジュールの HDL を変更しても別のモジュールには影響しません。再合成するモジュールまたはインスタンスは、HDL または制約の変更に基づいてツールにより判断されます。

サポートされているインクリメンタル合成フローは、次のとおりです。

- Synopsys 社 Synplify Pro/Premier (コンパイル ポイントを使用)
- Mentor Graphics 社 Precision (HDL の属性を使用してパーティションを指定)

もう 1 つの手法は、ボトムアップ合成手法です。このフローでは、各パーティションに個別の合成プロジェクトおよびネットリストがあります。HDL コードおよび合成制約の変更に基づいて、再合成する必要のある合成プロジェクトをユーザーが判断できます。最上位パーティションは下位モジュールにブラック ボックスを使用して合成され、下位モジュールは I/O およびクロック バッファを推論せずに合成されます。この方法は、Xilinx® Synthesis Technology (XST) およびサードパーティの合成ツールでサポートされます。.

ベンダー特定のインクリメンタル合成フローの利点は、次のとおりです。

- 各パーティションに対して個別の合成プロジェクト ファイルを作成する必要はありません。
- フラット合成フローからの移行が簡単です。
- 合成ツールにより、HDL コードおよびタイミング制約の変更に基づいて再合成が必要なモジュールが判断されます。

ボトムアップ フローの利点は、次のとおりです。

- 各パーティションに個別の合成プロジェクトがあるので、合成中、複数の設計者が同じデザインを作業できます。
- 再合成するインスタンスを完全に制御できます。再合成するプロジェクトをユーザーが決定するので、再合成されたインスタンスを特定するのが簡単です。各ネットリストに個別のタイム スタンプがあります。

Synplify Pro/Premier の使用

Synplify および Synplify Pro/Premier でパーティションを使用するには、基本的なボトムアップ合成フローと、コンパイル ポイントを使用したブロック ベース (インクリメンタル) 合成フローの 2 つがあります。

Synplify ボトムアップ フロー

ボトムアップ フローでは、各インスタンスに対応する Synplify プロジェクト ファイルがあります。下位プロジェクト ファイルに IOB またはグローバル クロック バッファを推論しないようにしてください。

プロジェクト ファイルで I/O 挿入をオフにするには、次の構文を使用します。

```
set_option disable_io_insertion 1
```

クロック バッファの使用をオフにするには、syn_noclockbuf 属性を使用します。次の構文を Synplify 制約 (SDC) ファイルに追加します。

```
define_attribute { clock_port } syn_noclockbuf 0
define_global_attribute  syn_noclockbuf 0
```

syn_noclockbuf 属性は、Verilog および VHDL コードに直接追加することも可能です。追加の例は、Synplify のマニュアルを参照してください。

Synplify Pro/Premier のインクリメンタル合成フロー

インクリメンタル合成フローは、Synplify Pro および Synplify Premier で使用できます。このフローでは、コンパイル ポイントを使用してデザインを小型の合成ユニットに分割します。locked モードを使用すると、コンパイル ポイントを越えたロジックの移動は実行されません。soft モードおよび hard モードでは境界を越えた最適化が許容されますが、サポートされていません。パーティションのネットリストが変更された場合、パーティションの再インプリメントが必要になります。インポートされたパーティションがネットリストと一致しない場合、NGDBuild でエラーが発生します。

次に、SDC ファイルのコンパイル ポイントの例を示します。

```
define_compile_point {v:controller} -type {locked} -cpfile {}
```

SDC ファイルでコンパイル ポイントを作成および変更するには、Synplify を使用してください。合成レポート ファイルに、各コンパイル ポイントのステータスが示されます。

```
Summary of Compile Points

Name              Status        Reason
-------------------------------------------------
controller        Unchanged     -
elevator_car      Remapped      Design changed
express_car       Remapped      Design changed
top               Remapped      Design changed
=================================================
```

図 2 : Synplify 合成レポートのパーティションを設定するコンパイル ポイントを示す部分の例

合成が終了したら、次の 3 つの方法でザイリンクス インプリメンテーション ツールを実行します。

- ISE コマンド ライン フロー : 単純な Tcl コマンドを実行することにより、Synplify でザイリンクス インプリメンテーション ツールで使用するアップデートされた PXML ファイルを作成します。詳細は、第 4 章「コマンドラインでのパーティション フロー」を参照してください。
- PlanAhead フロー : 合成ネットリストがインポートされます。オプションで、Synplify で生成された PXML ファイルもインポートされます。パーティションを、PlanAhead™ ソフトウェアで手動で再定義することも可能です。詳細は、第 5 章「PlanAhead でのパーティション フロー」を参照してください。
- Synplify コックピット : Synplify コックピットからザイリンクス インプリメンテーション ツールを実行します。詳細は、次のサイトから Synplify Pro/Premier の資料を参照してください。

  http://www.synopsys.co.jp/

Precision の使用

Mentor Graphics 社の Precision 合成ツールでも、階層デザイン フローがサポートされます。最もよく使用されるフローは、パーティション ベースのインクリメンタル フローです。このフローでは、属性を使用してパーティションを指定します。パーティションは、モジュールまたはインスタンスに設定できます。

```
Verilog Module:
module my_block( input clk; ...) /* synthesis incr_partition */;
Verilog Instance:
my_block my_block_inst( .clk(clk), ... .data_out(data_out) ); // synthesis attribute
my_block_inst incr_partition true
VHDL Module:
entity my_block is port( clk: in std_logic; ...); attribute incr_partition :
boolean; attribute incr_partition of my_block : entity is true; end entity my_block;
VHDL Instance:
component my_block is port( ... end component; ... attribute incr_partition :
boolean; attribute incr_partition of my_block_inst : label is true; ...
my_block_inst
```

図 3 : Precision 合成レポートのパーティションを設定する属性を示す部分の例

合成レポートには、パーティションのステートに基づいてパーティションが最適化されたかどうかが示されます。

```
[16027]: Incremental: Skipping Optimize for <...>.fifo_16_64.rtl_unfold_0
[16027]: Incremental: Skipping Optimize for <...>.fir_filter.rtl_unfold_0
[15002]: Optimizing design <...>.fsm.rtl_unfold_0
[16027]: Incremental: Skipping Optimize for <...>.fir_top.rtl
```

図 4 : 合成レポートの例

合成が終了したら、次の 3 つの方法でザイリンクス インプリメンテーション ツールを実行します。

- **ISE コマンド ライン フロー** : Precision で [Place & Route] をクリックし、ザイリンクス インプリメンテーション ツールで使用するアップデートされた PXML ファイルを作成します。詳細は、第 4 章「コマンド ラインでのパーティション フロー」を参照してください。
- **PlanAhead フロー** : 合成ネットリストがインポートされます。デザイン パーティションは PlanAhead 内で定義します。詳細は、第 5 章「PlanAhead でのパーティション フロー」を参照してください。
- **Precision** : Precision の [Place & Route] を使用してザイリンクス インプリメンテーション ツールを実行します。ISE ツールが自動的に起動します。このフローの詳細は、次の Mentor Graphics 社の SupportNet サイトからアプリケーション ノートを参照してください。

  http://supportnet.mentor.com/japan/index.cfm

XST の使用

XST では、パーティションの使用にボトムアップ合成フローがサポートされています。

パーティションとして使用される各インスタンスには、最上位パーティションも含め、個別のプロジェクト ファイルが必要です。最上位パーティションに IOB およびグローバル クロック バッファが含まれている場合は、下位パーティションには含めないでください。

I/O が推論されないようにするには、XST ファイルで次のオプションを指定します。

```
-iobuf NO
```

下位パーティションに IOB を使用することは可能ですが、その場合は最上位パーティションに余分な IOB が推論されないようにする必要があります。最上位パーティション全体で IOB の挿入をオフにするには `-iobuf` オプションを使用し、個別の信号で IOB の挿入をオフにするには信号名に `BUFFER_TYPE=NONE` 属性を設定します。`BUFFER_TYPE=NONE` 属性は、下位パーティションにグローバル バッファ (BUFG) がインスタンシエートされていて、最上位パーティションで BUFG が推論されないようにするために使用できます。

BUFFER_TYPE 属性の設定方法、XST をコマンド ライン モードで実行する方法の詳細は、『XST ユーザー ガイド』を参照してください。

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

XILINX®

第 4 章

コマンド ラインでのパーティション フロー

ISE® Design Suite のコマンド ライン フローでは、合成後のデザインにパーティションを使用します。このデザイン フローはコマンド ラインで実行し、ザイリンクスの NGDBuild、MAP、および PAR インプリメンテーション ツールを使用します。

インプリメンテーション ツールを実行すると、現在の作業ディレクトリで `xpartition.pxml` ファイルが検索されます。現在の作業ディレクトリは、NGDBuild、MAP、および PAR を実行したディレクトリです。インプリメンテーション ツールは、このファイルを使用して、パーティションの定義されている部分およびパーティションのステートを確認します。

この章には、次のセクションが含まれています。

PXML ファイルの作成

パーティションの定義は、xpartition.pxml ファイルに含まれます。PXML ファイルでは大文字と小文字が区別され、必ず `xpartition.pxml` という名前をつける必要があります。下位パーティションと共に、デザインの最上位モジュールをパーティションとして定義する必要があります。子パーティションまたはネストされたパーティションもサポートされています。

PXML ファイルは、テキスト エディタを使用して手動で作成するか、PlanAhead™ ソフトウェアなどのグラフィカル ユーザー インターフェイス (GUI) を使用して作成できます。PXML ファイルが現在の作業ディレクトリに存在していれば、インプリメンテーション ツールで自動的に検出されます。

手動で PXML ファイルを作成する際の参考に、`xpartition.pxml` ファイルのテンプレートが提供されています。テンプレート ファイルは、次のディレクトリにあります。

```
<Xilinx_12_directory>/PlanAhead/testcases/xpartition.pxml
```

次に、PXML ファイルの例を示します。

```
<?xml version="1.0" encoding="UTF-8" ?>


<Project FileVersion="1.2" Name="Example" ProjectVersion="2.0">
  <Partition Name="/top" State="implement" ImportLocation="NONE">
    <Partition Name="/top/module_A" State="import"
ImportLocation="/home/user/Example/export" Preserve="routing">
    </Partition>
    <Partition Name="/top/module_B" State="import"
ImportLocation="../export" Preserve="routing">
    </Partition>
    <Partition Name="/top/module_C" State="implement"
ImportLocation="../export" Preserve="placement">
    </Partition>
  </Partition>
</Project>
```

図 5 : xpartition.pxml ファイルの例

次の表に、上記の PXML ファイルの例で使用されている属性とその値を説明します。

表 1 : Project を定義するための PXML 属性

属性名	値	説明
FileVersion	1.2	ツールで使用されます。この値は変更しないでください。
Name	プロジェクト名	プロジェクト名を指定します。
ProjectVersion	2.0	ツールで使用されます。この値は変更しないでください。

表 2 : Partition を定義するための PXML 属性

属性名	値	説明
Name	パーティション名	パーティションを適用するモジュールの階層インスタンス名を指定します。
State	implement	パーティションは再インプリメントされます。
	import	パーティションはインポートされ、Preserve で設定されたレベルでインプリメンテーションが保持されます。
ImportLocation	パス	インポート先を指定します。State が import に設定されていない場合は無視されます。State が import に設定されている場合、パスを相対パスまたは絶対パスで指定できますが、指定した場所に export ディレクトリが含まれている必要があります。この属性が無視される場合、ディレクトリを設定したりこの属性を削除する代わりに、NONE キーワードを設定できます。
Preserve	routing	コンポーネントの配置配線が 100% 保持されます。最上位パーティションのデフォルト設定です。
	placement	配置が保持されます。配線は変更される場合があります。
	synthesis	配置および配線が変更される場合があります。
	inherit	親パーティションの設定値が使用されます。最上位パーティション以外のすべてのパーティションのデフォルト設定です。

PXML ファイルを手動で作成した場合やサードパーティの合成ツールで生成した場合は、「インプリメンテーションの実行」セクションに進んでください。PlanAhead を使用して PXML ファイルを作成する方法は、第 5 章「PlanAhead でのパーティション フロー」の「PXML のエクスポート」を参照してください。

インプリメンテーションの実行

`xpartition.pxml` ファイルを作成したら、ザイリンクス インプリメンテーション ツールをコマンド ラインまたはバッチ モードで実行できます。基本的なスクリプトは、次のようになります。

```
ngdbuild -uc design.ucf design.edn design.ngd
map -w design.ngd -o design_map.ncd design.pcf
par -w design_map.ncd design.ncd design.pcf
```

メモ : NGDBuild コマンドで指定するネットリストには、フラット合成の出力は指定できません。合成プロジェクトを設定および実行する際に、パーティションに設定する必要があります。合成フローの詳細は、第 3 章「合成パーティションフロー」を参照してください。

`xpartition.pxml` ファイルがツールで認識され、パーティションのステートが正しく設定されていることを確認するには、インプリメンテーション ツールのレポートを参照します。たとえば、NGDBuild レポート ファイル (BLD) を見ると、レポートの最後の方に、次の図に示すようなパーティション インプリメンテーション ステータスを示すセクションがあります。

```
Partition Implementation Status
-------------------------------
  Preserved Partitions:

  Implemented Partitions:

    Partition "/top":
Attribute STATE set to IMPLEMENT.

     Partition "/top/Control_Module":
Attribute STATE set to IMPLEMENT.

     Partition "/top/Express_Car":
Attribute STATE set to IMPLEMENT.

     Partition "/top/Main_Car":
Attribute STATE set to IMPLEMENT.

     Partition "/top/Tracking_Module":
Attribute STATE set to IMPLEMENT.
```

図 6 : NGDBuild レポートのパーティション ステータスを示す部分の例

MAP レポートおよび PAR レポートにも、同様のパーティション セクションがあります。

パーティションでサポートされていないインプリメンテーション オプション

パーティションでは、次のインプリメンテーション オプションはサポートされていません。

- -glob_opt (グローバル最適化)
- -smartguide (SmartGuide™ テクノロジ)
- -power (消費電力の最適化)

メモ : インプリメンテーション ツールの動作に影響を与えるザイリンクス環境変数 (XIL_*) も多数あり、ISE のリリースによってその影響は異なります。ザイリンクスの階層デザイン フローは、環境変数に対してテストされていないので、ツールを実行する前に特別なザイリンクス環境変数をすべて削除してください。

パーティションのエクスポート

満足するインプリメンテーション結果が得られたら、パーティションをエクスポートします。パーティションをエクスポートするには、implementation ディレクトリにあるすべてのファイルをエクスポート ディレクトリにコピーします。implementation ディレクトリの完全なコピーのサイズが大きすぎる場合は、スクリプトを使用して、次に示す必須ファイルと重要なオプションのファイルのみをコピーすることも可能です。

必須ファイル

必須ファイルは、*prev*.* ファイルと xpartition.pxml ファイルです。*prev*.* ファイルには、パーティションをインポートするのに必要なインプリメンテーション データが含まれています。PXML ファイルは、パーティションをインポートするのが適切であるかどうかを確認するために必要です。たとえば、PXML ファイルにインポートするパーティションが含まれていない場合、エラーが生成されます。

オプションのファイル

NGDBuild (.bld)、MAP (.mrp および .map)、PAR (.par)、TRACE (.twr/.twx) で生成されるポート ファイルおよび UCF ファイルはオプションですが、重要なファイルです。これらのファイルには、デザインをインプリメントしたときに使用されたオプションが記録されています。

エクスポート ディレクトリのファイルは変更しないでください。これらのファイルは、ソース ファイルのようなものと考えることができます。

デザインに複数のパーティションがある場合、すべてがエクスポートされます。特定のパーティションがエクスポートされるというよりは、デザイン全体がエクスポートされると考える方がわかりやすいです。ランタイムを短縮するため、エクスポートされるデータをすべて 1 つのディレクトリに配置することをお勧めします。これにより、パーティションをインポートする際にメモリに読み込む必要のあるデザイン ファイルの数を制限できます。

パーティションのステートを import に変更

パーティションをエクスポートしたら、xpartition.pxml ファイルでパーティションのステートをアップデートする必要があります。これには、テキスト エディタを使用できます。エクスポートしたパーティションをインポートする場合は、パーティションのステートを import に設定し、ImportLocation 属性をエクスポートされたパーティションの場所に設定します。相対パス (../export など) または絶対パス (/home/user/Example/export など) のどちらでも設定できます。.

デザインでの反復作業

必要に応じてデザインを変更し、再合成します。合成でエクスポートされたパーティションが変更された場合は、xpartition.pxml ファイルでそのパーティションのステートを import から implement に手動で変更する必要があります。インポート先のパーティションに完全に一致しないパーティションをインポートしようとすると、NGDBuild で次のようなエラー メッセージが表示されます。

```
ERROR:NgdBuild:1037 - The logic for imported partition '/top/Main_Car' using
previous implementation './export/top_prev_built.ngd' has been modified.
```

このエラー メッセージは、ソースが変更されたパーティションを再インプリメントおよびエクスポートしていない場合に表示されます。パーティションを再インプリメントし、エクスポートしてから、デザインにインポートする必要があります。HDL にコメントを追加した場合にも、ネットリストで多少のロジックの変更や名前の変更が発生する可能性があり、パーティションを再インプリメントする必要があることに注意してください。

保持レベルを routing 以外に設定してパーティションをインポートし、PAR により配置および配線が変更された場合、次の実行でその結果が使用されるようパーティションをエクスポートする必要があります。

SmartXplorer の使用

SmartXplorer は、パーティションを含むデザインでもフラット デザイン フローと同様に使用できます。タイミング要件が厳しいパーティションが終了した時点で、SmartXplorer を実行してタイミングを満たすソリューションを検索できます。その後、タイミング クリティカルなモジュールの SmartXplorer での結果をインポートし、変更される可能性のあるデザインのほかの部分のインプリメンテーションには通常のフローを使用します。

SmartXplorer の実行結果からパーティションをインポートする場合は、PXML の ImportLocation として SmartXplorer で作成されたディレクトリを指定する必要があります。このディレクトリへのパスは、相対パス (`../run2` など) または絶対パス (`/home/user/Example/run2` など) のどちらでも設定できます。

SmartXplorer には、最上位ネットリストまたは NGD ファイルを入力します。NGD を指定する場合は、次の点に注意する必要があります。

- NGD ファイルは、MAP および PAR で使用されるのと同じ `xpartition.pxml` ファイルを使用して作成する必要があります。NGDBuild を実行したときに PXML ファイルが存在しなかい場合は、MAP および PAR で PXML ファイルが無視されます。
- SmartXplorer では MAP および PAR が実行され、NGD ファイルは 1 レベル上に作成されます。つまり、SmartXplorer の結果をインポートするには、1 レベル上にある *prev_built.ngd ファイルを SmartXplorer で作成された適切な実行ディレクトリにコピーする必要があります。

SmartXplorer の詳細は、『コマンド ライン ツール ユーザー ガイド』(UG688) を参照してください。

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

パーティションの削除

パーティションを削除するには、`xpartition.pxml` ファイルでそのパーティションの参照を削除するか、親パーティションのステートを implement に設定します。すべてのパーティションを削除してフラット フローを実行するには、implementation ディレクトリの `xpartition.pxml` ファイルを削除するか、名前を変更します。パーティションを再度設定するには、implementation ディレクトリに `xpartition.pxml` ファイルを再度追加します。入力ネットリストが変更されておらず、エクスポート ディレクトリが存在していれば、パーティションをインポートできます。

XILINX®

第 5 章

PlanAhead でのパーティション フロー

この章では、PlanAhead™ ソフトウェアでのパーティション フローの概要を説明します。現在のところ、PlanAhead ではネットリスト プロジェクトでのみパーティションがサポートされており、HDL ベースの PlanAhead プロジェクトを作成してパーティション フローを実行することはできません。そのため、合成は PlanAhead の環境外で実行する必要があります。

パーティションは、個別のネットリストを持つ階層 (ボトムアップ フロー) またはインクリメンタル合成フローで作成されたネットリストに設定する必要があります。合成プロジェクトをパーティション用に設定する方法は、第 3 章「合成パーティションフロー」を参照してください。

メモ：PlanAhead でフラット ネットリスト プロジェクトを作成し、プロジェクトの階層にパーティションを追加することも可能ですが、合成で生成されるネットリストは 1 つだけなので、デザインの 1 つの部分を変更すると、ほかの部分にも影響します。フラットなグローバル最適化合成フローはサポートされていません。

この章には、次のセクションが含まれています。

- 新規プロジェクトの作成
- パーティションの作成
- PXML のインポート
- PXML のエクスポート
- パーティションのフロアプラン
- パーティション デザインのインプリメント
- パーティションのプロモート
- パーティション ステートの管理
- 保持レベルの管理
- デザイン実行の管理

新規プロジェクトの作成

PlanAhead ソフトウェアを起動したら、Getting Started ページで [Create New Project] リンクをクリックします。開いた New Project ウィザードの指示に従がって、新規 PlanAhead プロジェクトを作成します。[Design Source] ページで [Specify synthesized (EDIF or NGC) netlist] をオンにし、デザイン ソースとして合成済みネットリストを指定します。パーティションでは、RTL ソースのプロジェクトはサポートされていません。

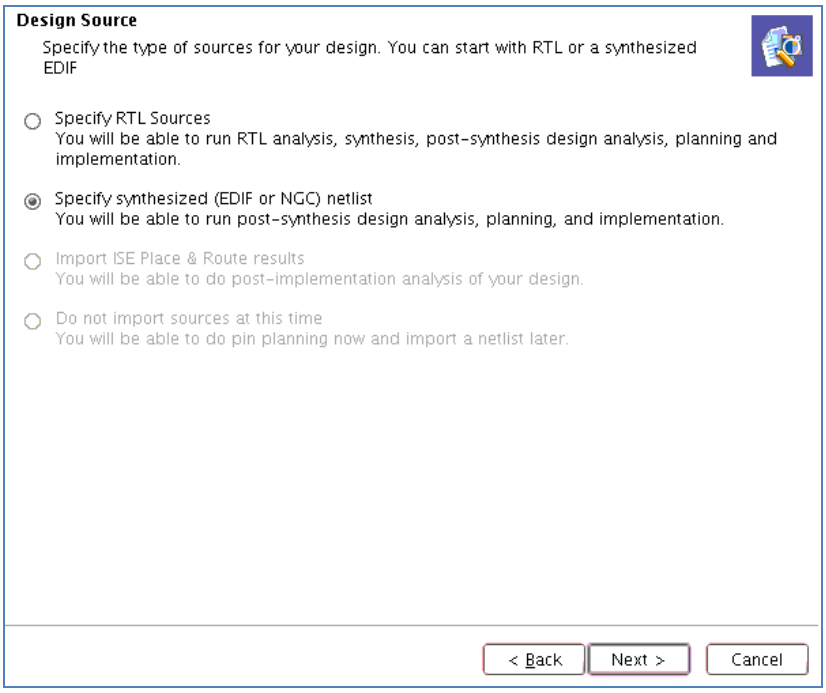

図 7 : デザイン ソースとして合成済みネットリストを選択

ウィザードでの設定を続行し、最上位ネットリスト、下位ネットリストの場所、および UCF 制約ファイルを指定します。すべてを指定したら、[Finish] をクリックして PlanAhead で プロジェクトを開きます。

パーティションの作成

PlanAhead でパーティションを作成するには、次の手順に従います。

1. 左側にある Flow Navigator で [Netlist Design] をクリックしてネットリストを読み込みます。
2. ネットリストが開いたら、[Netlist] ビューをクリックします。
3. パーティションを設定するモジュール インスタンスを選択して右クリックし、[Set Partition] をクリックします (図 8)。

メモ : このコマンドは、パーティションに設定して設計および合成したインスタンスにのみ実行可能です。

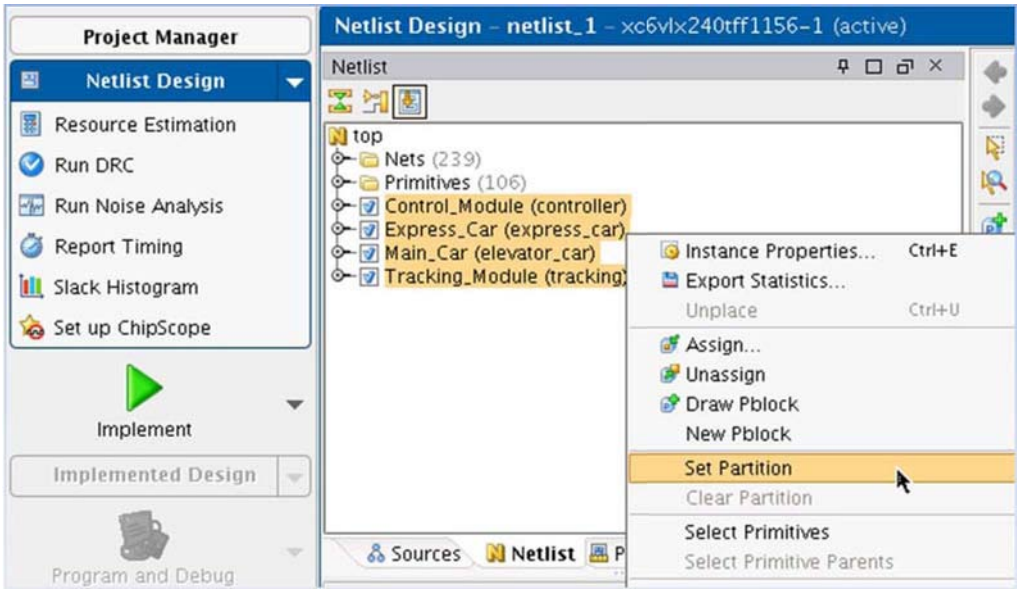

図 8 : PlanAhead でのパーティション設定

PXML のインポート

Synplify のコンパイル ポイント フローを使用する場合、PXML ファイルは Synplify で生成されます。このファイルの値を PlanAhead にインポートし、パーティションを定義できます。ただし、プロジェクトで既にパーティションを設定している場合はインポートできず、このオプションは淡色表示されます。

Synplify で生成された `xpartition.pxml` ファイルをインポートするには、次の手順に従います。

1. PlanAhead で新規ネットリスト プロジェクトを作成します。
2. Flow Navigator の [Netlist Design] をクリックしてネットリストをメモリに読み込みます。
3. [Netlist] ビューをクリックしてデザイン階層を確認します。
4. [Netlist] ビューを右クリックし、[Import Partition Settings (pxml) file] をクリックします。

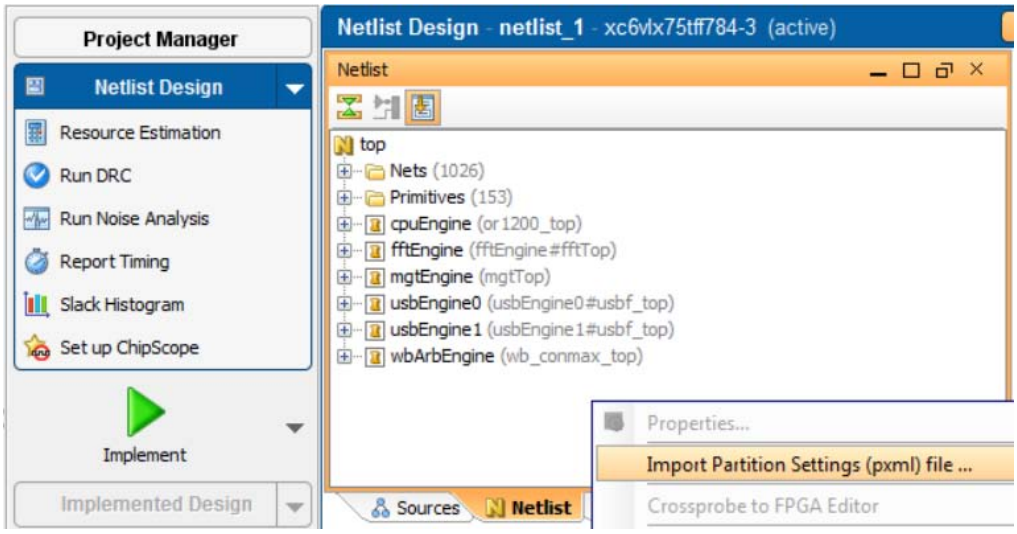

図 9 : パーティション設定 (PXML) ファイルのインポート

PXML のエクスポート

PlanAhead を使用して `xpartition.pxml` ファイルを作成するには、次の手順に従います。

1. PlanAhead ソフトウェアで新規ネットリスト プロジェクトを作成し、最上位ネットリストおよびパーティションを定義する下位ネットリストを指定します。
2. Flow Navigator の [Netlist Design] をクリックしてネットリストをメモリに読み込みます。
3. [Netlist] ビューをクリックします。
4. パーティションを設定するモジュール インスタンスを選択して右クリックし、[Set Partition] をクリックします (図 10)。

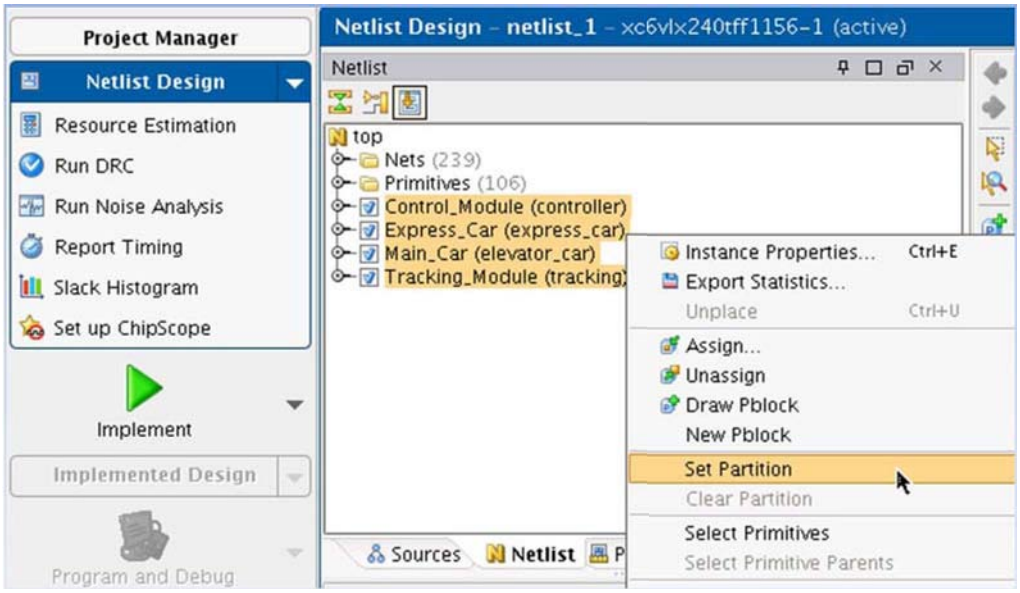

図 10 : PlanAhead でのパーティション設定

5. [Implement] ボタンの右側の矢印をクリックし、[Implementation Settings] をクリックします (図 11)。

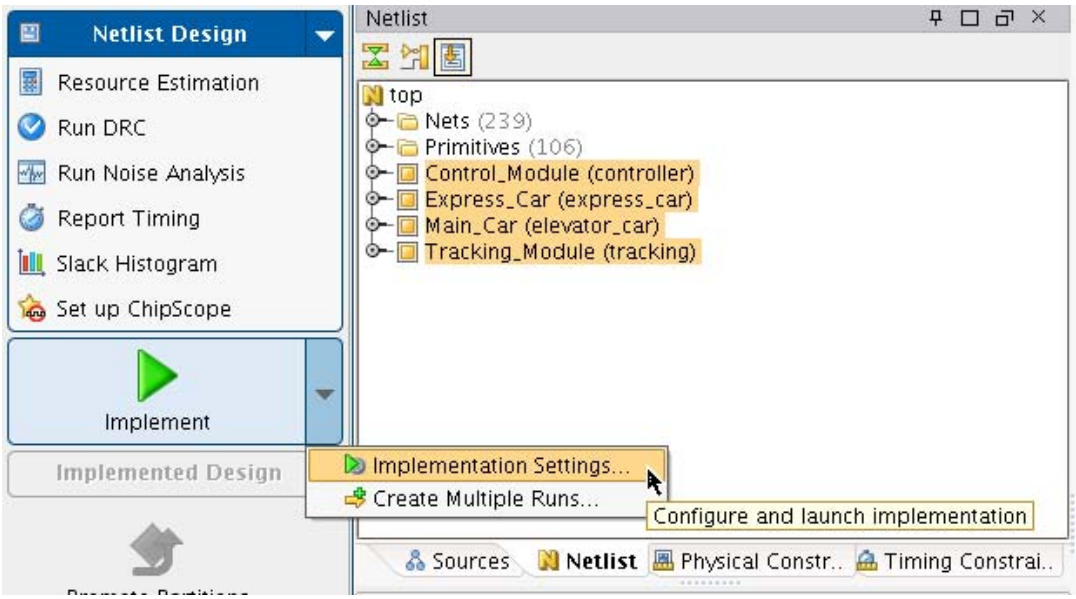

図 11 : インプリメンテーション設定

6. [Implementation Settings] ダイアログ ボックスで、[Launch Options] フィールドの右側にあるボタンをクリックします (図 12)。

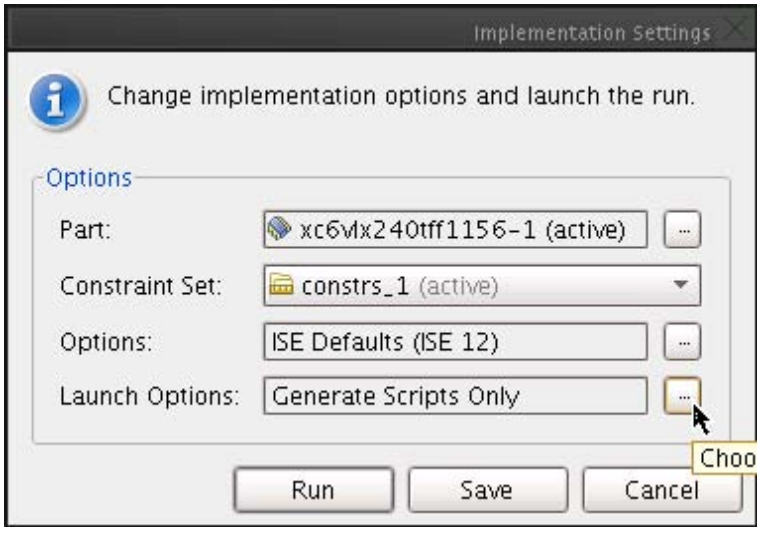

図 12 : 起動オプションを開く

7. [Specify Launch Options] ダイアログ ボックスで、[Generate scripts only] をオンにして [OK] をクリックします。
8. [Run] をクリックします。

PlanAhead からさまざまな階層デザイン フローを実行する手順は、第 6 章の「PlanAhead フロー」を参照してください。

PlanAhead の使用法の詳細は、『PlanAhead ユーザー ガイド』(UG632) を参照してください。

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

`xpartition.pxml` ファイルは、`<project_name>.runs/impl_1` ディレクトリに生成されます。このファイルを、ISE コマンド ライン フローを実行するディレクトリにコピーします。

パーティションのフロアプラン

パーティションをフロアプランするかどうかを判断する方法については、第 1 章の「パーティションのフロアプラン」を参照してください。ここに示す単純な例では、4 つのパーティションをフロアプランします。

[Netlist] ビューでパーティションを 1 つずつ選択すると、各パーティションに対して Pblock を作成できます。PlanAhead の Pblock は、AREA_GROUP 制約を定義します。

Pblock を作成するには、次の手順に従います。

1. パーティション ネットリストを右クリックします。
2. [Draw Pblock] をクリックします。
3. [Device] ビューで、各パーティションを表す矩形を描きます。

すべてのパーティションに対して Pblock を作成すると、図 13 のようになります。

PlanAhead でモジュールをフロアプランする方法の詳細は、ザイリンクス Web サイトから『PlanAhead チュートリアル : デザイン解析とフロアプラン』(UG676) を参照してください。

http://japan.xilinx.com/support/documentation/dt_planahead_planahead12-2_tutorials.htm

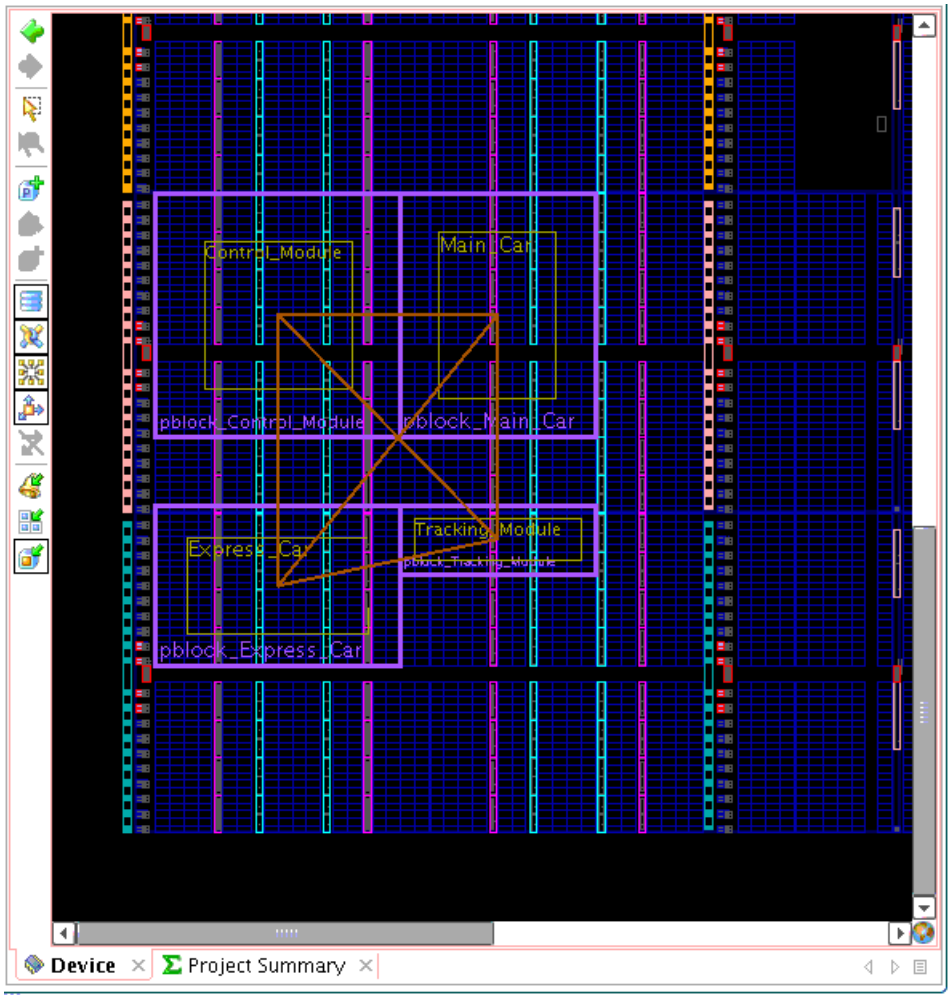

図 13 : PlanAhead でのパーティション設定

パーティション デザインのインプリメント

パーティションを設定したデザインをインプリメントするには、Flow Navigator で [Implement] ボタンをクリックします。

- ツールの使用に慣れていない場合や最小限の設定で実行する場合は、緑色の [Implement] ボタンをクリックします。このボタンをクリックすると、デフォルトの ISE 設定でインプリメンテーション ツール (NGDBuild、MAP、PAR、および TRACE) が実行されます。インプリメンテーション オプションを変更してインプリメンテーション ツールを実行するには、[Implenet] ボタンの右側にある矢印をクリックしてプルダウン メニューから [Implementation Settings] をクリックし、必要な変更を加えます。
- プルダウン メニューから [Create Multiple Runs] をクリックすると、異なるインプリメンテーション オプションを使用して複数のインプリメンテーション実行を作成できます。これは、SmartXplorer の実行に似ています。

インプリメンテーションの結果は、[Compilation Log] ビュー、PlanAhead 右上のステータス バー、または [Window] → [Design Suns] をクリックして [Design Runs] ビューで確認できます。

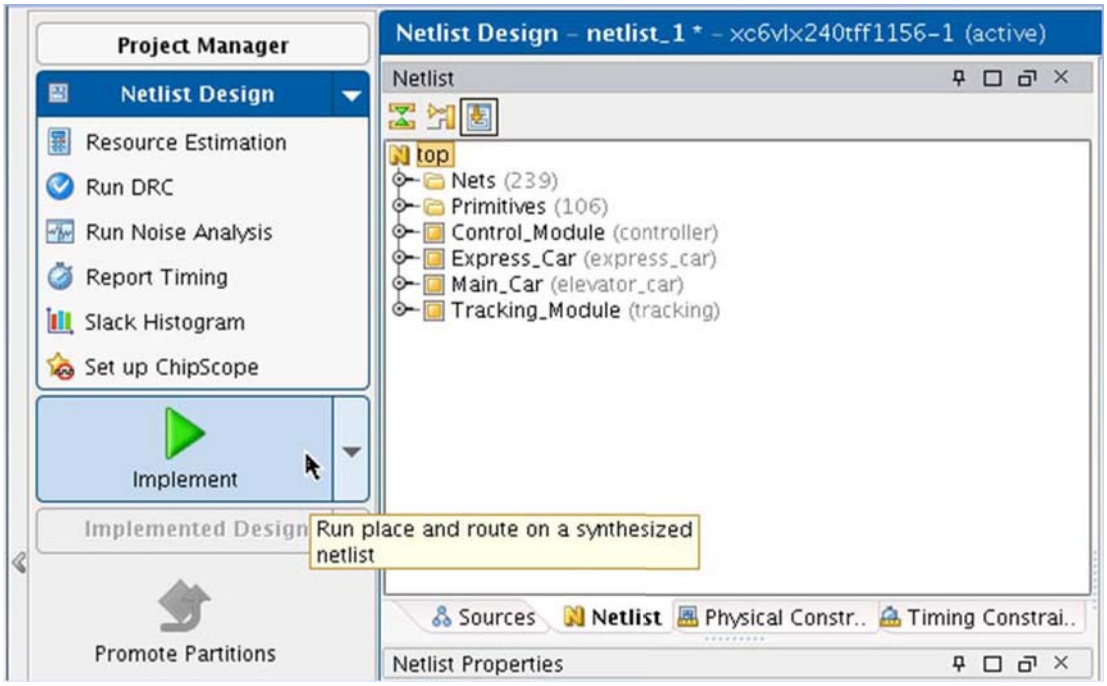

図 14 : [Implement] ボタン

インプリメンテーションが完了したら、[Implemented Design] をクリックすることにより結果を読み込むことができます。パーティションが適切に定義されており、インプリメンテーション ツールで使用されているかどうかを確認するには、インプリメンテーション ログ ファイルを表示します。

パーティションのプロモート

パーティションがインプリメントされたら、今後のインポート実行用にプロモートする必要があります。PlanAhead でパーティションをプロモートするのは、コマンド ライン フローでパーティションをエクスポートするのと同じです。アクティブな実行からパーティションをプロモートするには、Flow Navigator で [Promote Partitions] ボタンをクリックします (図 15) 。

メモ : 実行結果をプロモートしないと、実行をリセットしたときに現在のパーティションが失われます。

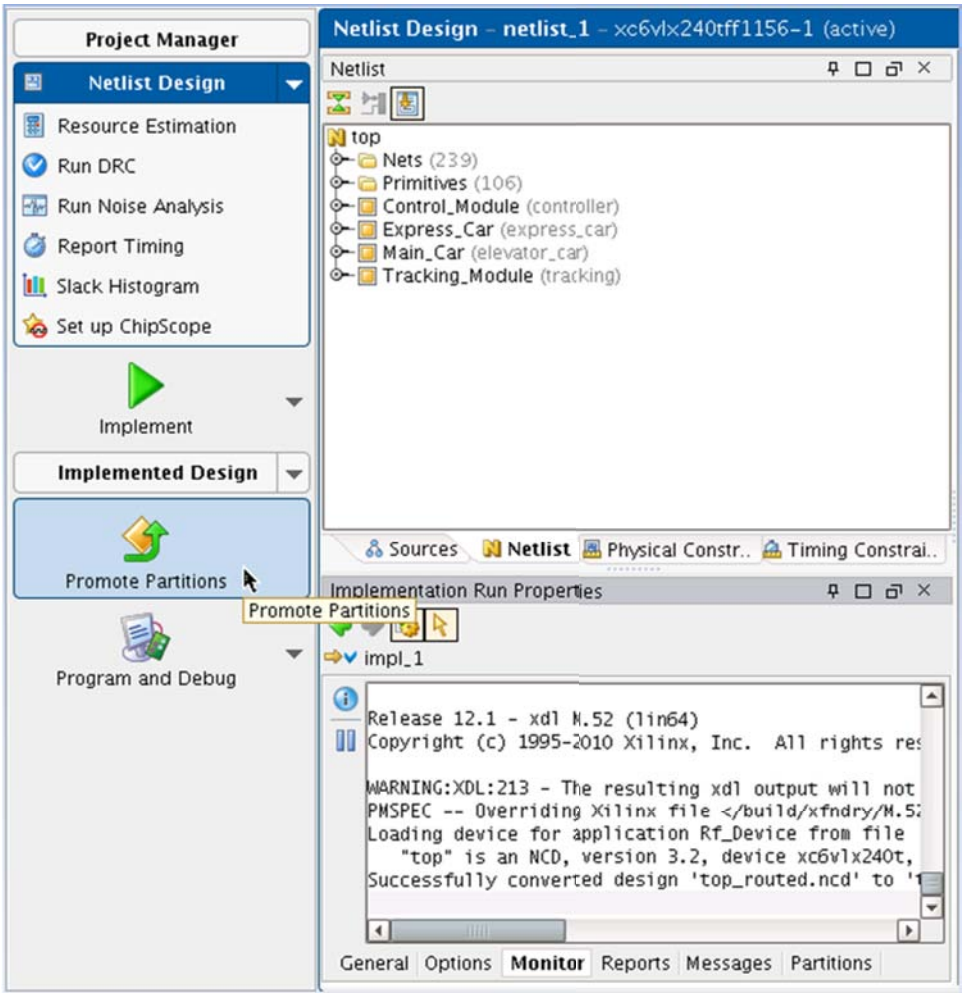

図 15 : パーティションのプロモート

結果をプロモートしたら、現在の実行のリセットおよび再実行を必要なだけ繰り返します。プロモートされた結果は、パーティションが変更されなければ常にインポートされます。新しい結果をプロモートすると、現在プロモートされているデータが上書きされます。複数の実行およびプロモートされたデータの管理については、次のザイリンクス Web サイトから『PlanAhead ソフトウェア チュートリアル : 予測可能な結果に対する保存デザインの利用』(UG747) を参照してください。

http://japan.xilinx.com/support/documentation/dt_planahead_planahead12-2_tutorials.htm

パーティション ステートの管理

プロモートしたパーティションのステートは、次の実行用に自動的に import にアップデートされます。プロモートしていないパーティションでもインポートできますが、これらのパーティションは PlanAhead では自動的に管理されません。これらのパーティションは、PlanAhead の [Implementation Run Properties] ビューの [Partitions] タブで手動で管理します (図 16)。

ソース ネットリストがアップデートされたり、物理制約が変更されたりしてパーティションのアップデートが必要になった場合は、パーティションのステートを implement に変更する必要があります。このステートの変更は、PlanAhead では自動的に実行されません。ステートを正しく変更しないと、次のような NGDBuild エラーが発生します。

```
ERROR:NgdBuild:1037 - The logic for imported Partition '/top/Express_Car'
   using previous implementation
   '../../project_1.promote/Ximpl_1/top_prev_built.ngd'
has been modified.This situation can occur when the source for the
Partition was modified but the Partition was not re-implemented
   and exported.
You must re-implement and export the Partition before it can be imported
into this design.
```

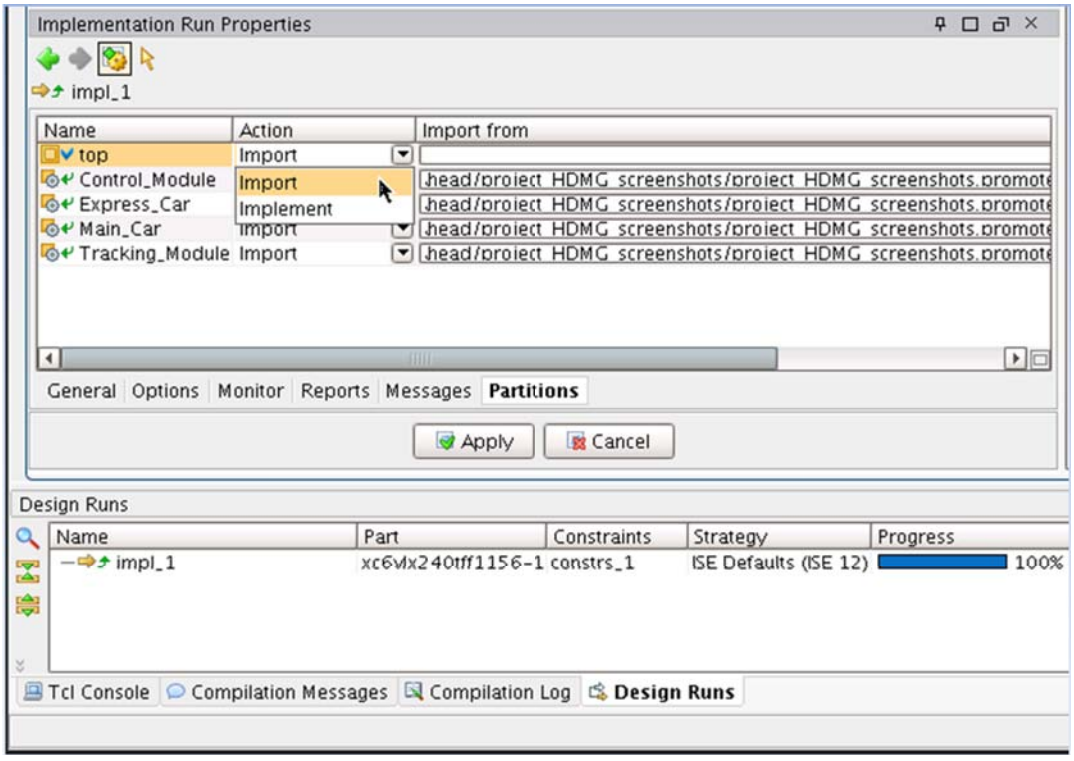

図 16 : [Implementation Run Properties] ビューでのパーティション ステートの変更

保持レベルの管理

保持レベルは、デフォルトでは PlanAhead に表示されません。保持レベルをデフォルト値から変更する場合は、[Implementation Run Properties] ビューの [Partitions] タブにフィールドを追加します。[Preservation] フィールドを表示するには、次の手順に従います。

1. [Design Runs] ビューをクリックします。このビューが表示されていない場合は、[Window] → [Design Runs] をクリックします。
2. インプリメンテーション実行を選択します。デフォルトでは impl_1 が選択されています。
3. [Implementation Run Properties] ビューで [Partitions] タブをクリックします。
4. 表のヘッダを右クリックし、[Preservation] をクリックしてこの列を表に追加します (図 17)。
5. 必要に応じて値を変更します。

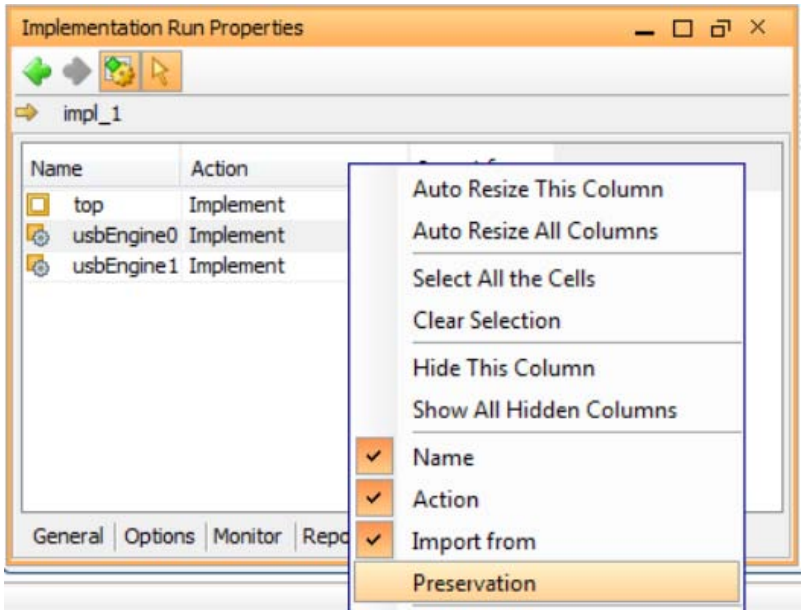

図 17 : PlanAhead に保持レベル属性を表示

デザイン実行の管理

デザインに必要な変更を加え、パーティション ステートを正しく設定したら、デザインを再インプリメントできます。PlanAhead でのデザイン保持に推奨されるのは、1 つのデザイン実行 (impl_1) を保持し、必要に応じてプロモートおよび再インプリメントを繰り返すフローです。

この章の例では、Flow Navigator で [Implement] ボタンをクリックしてデザインを再インプリメントできます。これにより実行データがリセットされ、インプリメンテーション ツールが起動され、[Implementation Run Properties] ビューの [Partitions] タブでの定義に応じてパーティションがインポートされます。インプリメンテーションが完了して結果をプロモートすると、以前にプロモートされていた結果が上書きされるので、1 つのプロモート ディレクトリのみにデザイン パーティションの結果が含まれます。プロモート ディレクトリが 1 つのみであれば管理は簡単であり、第 1 章の「インポート先ディレクトリ」で説明したような複数の場所からインポートすることによる煩雑さを回避できます。

XILINX®

第 6 章

デザイン保持フロー

デザイン保持の目的は、タイミング クロージャ段階でのインプリメンテーションの実行回数を減らすことです。デザインの一部でタイミングが満たされたら、そのインプリメンテーション結果 (配置および配線) を次の実行で使用します。これにより、完成し、タイミングを満たしている部分は、デザインのほかの部分が変更されても影響を受けません。

デザイン保持では、パーティションを使用してモジュール インスタンスの以前のインプリメンテーション結果を保持します。パーティションを適切に使用することにより、デザインの実行回数が削減され、変更されていないインスタンスを再検証する必要がないので、タイミング クロージャ段階の時間を短縮できます。デザインの一部を保持した場合、どのモジュールがインプリメントされ、どのモジュールが保持されるかによって、インプリメンテーションのランタイムは異なります。インプリメントされるモジュールのタイミング要件が厳しい場合、ランタイムは長くなります。インプリメントされるモジュールのタイミングを簡単に満たすことができ、クリティカル パスが保持されるモジュールに含まれている場合は、ランタイムは短くなります。

この章では、ISE® Design Suite コマンド ライン ツールおよび PlanAhead™ ソフトウェアを使用して、RTL フローおよびネットリスト フローでデザイン保持を使用する方法を示します。この章には、次のセクションが含まれています。

- コマンド ライン フロー
- PlanAhead フロー

コマンド ライン フロー

コマンド ラインからデザイン保持フローを実行する際は、どのモジュールがアップデートされているか、どのパーティションをインプリメントし、どのパーティションをインポートするのか、ユーザーが管理する必要があります。これらを編集するには、テキスト エディタまたは XML エディタで xpartition.pxml ファイルを変更します。このフローを実行するのに使用する必要のある特別なインプリメンテーション オプションはありませんが、PXML ファイルをインプリメンテーションを実行するディレクトリに配置する必要があります。PXML ファイルの作成および管理の詳細は、このガイドの第 4 章「コマンド ラインでのパーティション フロー」を参照してください。

デザイン保持フローでは、デザイン サイクルの初期段階でデザインのパーティションを定義する必要があります。デザインの初期のコード記述が完了したら、インクリメンタル合成フローまたはボトムアップ合成フローを使用して合成できます。サポートされる合成ツールおよびフローの詳細は、このガイドの第 3 章「合成パーティションフロー」を参照してください。

デザインを合成したら、PXML ファイルですべてのパーティションのステートを implement に設定してデザインをインプリメントします。パーティションでタイミングが満たされたら、その結果を今後の実行用にエクスポートします。

バグの修正や機能の向上のためにデザインを変更したら、変更したパーティションをすべて再インプリメントする必要があります。ネットリスト レベルで変更されていないパーティションは、インポートして配置配線情報を保持します。

インポートしたパーティションのネットリストがインポート先ディレクトリで指定されているデザインと一致しない場合は、NGDBuild でエラーが発生します。この場合、パーティションのステートを implement に変更する必要があります。パーティション モジュールが完了し、タイミングが満たされていれば、デザインのほかの部分を変更している間、継続してインポートできます。

PlanAhead フロー

このセクションでは、デザインを保持するために PlanAhead を使用する方法を説明します。デザイン保持フローは、パーティション ベースのフローです。ソフトウェアの基本的な実行方法は、第 5 章「PlanAhead でのパーティション フロー」で説明しました。より詳細なチュートリアルは、次のザイリンクス Web サイトから『PlanAhead ソフトウェア チュートリアル : 予測可能な結果に対する保存デザインの利用』(UG747) を参照してください。

http://japan.xilinx.com/support/documentation/dt_planahead_planahead12-2_tutorials.htm

PlanAhead でのデザイン保持フローの手順は、次のとおりです。

1. デザインでボトムアップ合成またはインクリメンタル合成を実行します。
2. 手順 1 の合成結果を使用してネットリスト ベースの PlanAhead プロジェクトを作成し、存在する場合は UCF 制約ファイルを読み込みます。
3. PlanAhead で 1 つずつパーティションを定義するか、既存の `xpartition.pxml` ファイルをインポートします。
4. 必要なタイミング制約 (PERIOD、OFFSET IN/OUT など) およびフロアプラン制約 (I/O ピン LOC、AREA_GROUP など) を作成します。
5. すべてのパーティションのステートを implement に設定してデザインをインプリメントします。
6. 適切なスタティックおよびダイナミック タイミング結果が得られたインプリメント済みパーティションをプロモートします。
7. プロモートしたパーティションをインポートし、合成およびインプリメンテーションを繰り返します。

XILINX®

付録 A

パーティションのデバッグ

インプリメンテーション エラー

インポートしようとするとエラーが発生する

次のようなエラー メッセージが表示される場合、xpartition.pxml ファイルがインポート先ディレクトリにコピーされていません。

```
ERROR:HierarchicalDesignC:154 - Import project "./import" is not an XPartition
project.
ERROR:HierarchicalDesignC:143 - Error importing Partition "/top/modulea".
ERROR:HierarchicalDesignC:142 - Error found in XPartition file data.
```

インポートしたときにデザインのタイミングが満たされない

デザインのパーティションされている部分でタイミングが満たされない場合、インポートされたパーティションの保持レベルを synthesis に緩和すると、インポートされたパーティションの配置および配線を必要に応じて変更でき、ツールでより柔軟な処理が可能となるため、タイミングが満たされる可能性が高くなります。タイミングが満たされたらパーティションをエクスポートし、保持レベルを必要に応じて placement または routing に変更します。

保持レベルを緩和してもデザインのタイミングが満たされない場合は、パーティションのステートを import から implement に変更するか、クリティカル パスを含むパーティションを削除します。

また、標準フラット フローのタイミング クロージャ手法も適用できます。

デザインを配置できない

デザインのパーティション数を多くし、インポートされるロジックの割合を高くすると、インプリメントされるロジックで使用可能なリソースが少なくなり、デザインを配置できないことがあります。

この場合、次のようなメッセージが表示されます。

```
WARNING:Place:1178 - 4 sites were unavailable during the placement phase because
they were already used by routing within preserved Partitions.If the Placer is
not able to find a successful solution, the following suggestions may help the
Placer find a solution.


1) To allow the Placer to use these sites, back off the preservation level on
one or more of the following Partitions.The following Partitions contain
unavailable sites:

Partition /top/Control_Module:3 (preserve=routing)

Partition /top/Express_Car:1 (preserve=routing)


If the problem persists, then careful floorplanning can minimize the number of
sites occupied by routing.
```

これらのサイトを使用できないのは、インポートされた配線に含まれるピンによりコンポーネントが使用不可能になっているからです。配線ツールでインポートされた配線を移動することができれば、これらのコンポーネントにロジックを配置できます。これには、使用不可能なサイトの多いパーティションで保持レベルを placement または synthesis に変更します。これでは十分でない場合は、パーティションを再インプリメントする必要があります。

最上位パーティションのステートを implement に設定すると、配置ツールで有効な配置を見つけるのに十分なリソースが使用可能になる可能性があります。これでは十分でない場合は、そのほかのパーティションのステートも implement に変更してみてください。PlanAhead™ ソフトウェアで配置配線情報を読み込み、各パーティションをハイライトして FPGA のどこに位置しているかを確認すると、どのパーティションのステートを implement に変更したらよいのかを判断するのに役立ちます。

パーティションの数が多すぎると、デバイスの使用率に悪影響を与える場合があります。デザインのクリティカルでない部分のモジュールからパーティションを削除すると、その他のロジックで使用できるリソースが増え、ツールでタイミングを満たすためのソリューションをより柔軟に見つけ出すことができます。

また、適切なフロアプランを使用すると、最終的な配置を見つけるのに役立つ場合があります。すべてのロジックをチップ全体のどこにでも配置可能な場合、配置ツールでインプリメントされる新しいロジック用のリソースを見つけるのが困難な場合があります。フロアプランでは、各パーティションをデザインの特定の部分に割り当てます。AREA_GROUP 範囲を追加または変更すると、エクスポートされたパーティション データのアップデートが必要となるので、次の実行でパーティションのステートを implement に設定する必要があります。

デザインを配線できない

インポートしたパーティションを含むデザインを配線できない場合は、保持レベルを placement に変更して PAR を再実行するのが最も簡単な方法です。MAP を再実行する必要はありません。

フロアプランも、配置の問題を解決するのに役立つのと同様の理由で、配線の問題を解決するのに役立つ場合があります。

保持レベルを変更したり、フロアプランを適用しても配線が完了しない場合は、配置されたデザインを解析すると、ステートを変更する必要のあるパーティションを判断できます。PlanAhead を使用すると、配置配線情報を読み込み、各パーティションをハイライトして FPGA のどこに位置しているかを確認できます。

パーティションによりスライスの使用量が増加する

パーティションを使用すると、最適化およびパッキングの制限により、スライスの使用量が多少増加することが予測されますが、これは通常数パーセントか、理想的には無視できる程度である必要があります。

場合によっては、スライスの使用量が大幅に増加します。1 つの原因は、LUT とフリップフロップの比率が大きく異なっていることです。パーティションが使用されていないデザインでは、モジュール A のコンポーネントの一部をモジュール B のコンポーネントと結合できる場合がありますが、パーティションを使用するとパッキングが制限され、パーティション A のコンポーネントをパーティション B のコンポーネントと共にパックすることはできません。このため、LUT とフリップフロップのバランスが取られていないロジックで空のスライスが使用されることになり、スライスの使用量が増加します。

デザインでこの問題が発生しているかどうかを確認するには、マップ レポート (`.mrp`) でエリア グループおよびパーティション サマリを参照してください。次に、この例を示します。

```
Partition "/top/moduleA":
  State=implement
  Slice Logic Utilization:
    Number of Slice Registers:           4,318 (4,318)
    Number of Slice LUTs:                4,337 (4,337)
      Number used as logic:              3,717 (3,717)
      Number used as Memory:               620 (620)
  Slice Logic Distribution:
    Number of occupied Slices:           1,750 (1,750)
    Number of LUT Flip Flop pairs used:  5,249 (5,249)
      Number with an unused Flip Flop:     931 out of  5,249   17%
      Number with an unused LUT:         1,508 out of  5,249   28%
      Number of fully used LUT-FF pairs: 2,810 out of  5,249   53%
  Number of Block RAM/FIFOs:               3 (3)
  Number of BUFDS:                           1 (1)
  Number of BUFR:                              1 (1)
  Number of GTX_DUAL:                  2 (2)
  Number of PLL_ADV:                     1 (1)
```

図 18 : マップ レポートのパーティション リソース サマリ

この例では、使用量に対して 2 つの値が示されています。1 つ目の値は個々のパーティションのリソース使用量で、かっこに含まれている 2 つ目の値はそのパーティションと子パーティションすべての値です。このことは MRP レポートのパーティション サマリ セクションの最初に記述されており、見逃されがちです。次に、この部分の記述を示します。

```
Partition Resource Summary:
---------------------------
Resources are reported for each Partition followed in parenthesis by resources
for the Partition plus all of its descendents.
```

BitGen の DRC エラー

デザインで PAR の実行が問題なく完了しても、BitGen で次のような DRC エラーが発生することがあります。

```
ERROR:PhysDesignRules:10 - The network <signal_OBUF> is completely unrouted
```

これは、パーティションの駆動されていない出力が、親パーティションのロジックに接続されていることが原因です。フラット フローのように親パーティションのロジックをマップで最適化できないため、部分的に配線された信号がデザインに含まれることになり、BitGen で DRC エラーが発生します。

この問題は HDL レベルで修正してください。HDL を修正した場合、修正されたパーティションを再インプリメントする必要があります。

ChipScope のサポート

ChipScope™ ツールでパーティションはサポートされていますが、パーティションで ChipScope コアを変更した場合はパーティション ステートを implement に変更する必要があります。