

Vivado Design Suite ユーザー ガイド

デザイン フローの概要

UG892 (v2012.2) 2012 年 7 月 25 日



Notice of Disclaimer

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at <http://www.xilinx.com/warranty.htm>; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: <http://www.xilinx.com/warranty.htm#critapps>.

© Copyright 2012 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, 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 (v2012.2) 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。

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。

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
2012 年 7 月 25 日	2012.2	初版

目次

改訂履歴	2
第 1 章：概要	
高レベル デザイン フロー	4
第 2 章：デザイン フローの機能	
RTL 開発および解析	6
IP の設定とインプリメンテーション	6
ロジック シミュレーション	7
I/O ピン配置	7
ロジック合成	7
ネットリストの解析および制約の定義	7
インプリメンテーション	8
インプリメンテーション結果の解析および フロアプラン	8
デバイスのプログラム、検証、およびデバッグ	8
第 3 章：基本的なデザイン フロー	
プロジェクト モードと非プロジェクトモードの理解	9
プロジェクト モードの使用	14
非プロジェクト モードの使用	39
付録 A：その他のリソース	
ザイリンクス リソース	45
ソリューション センター	45
リファレンス	45

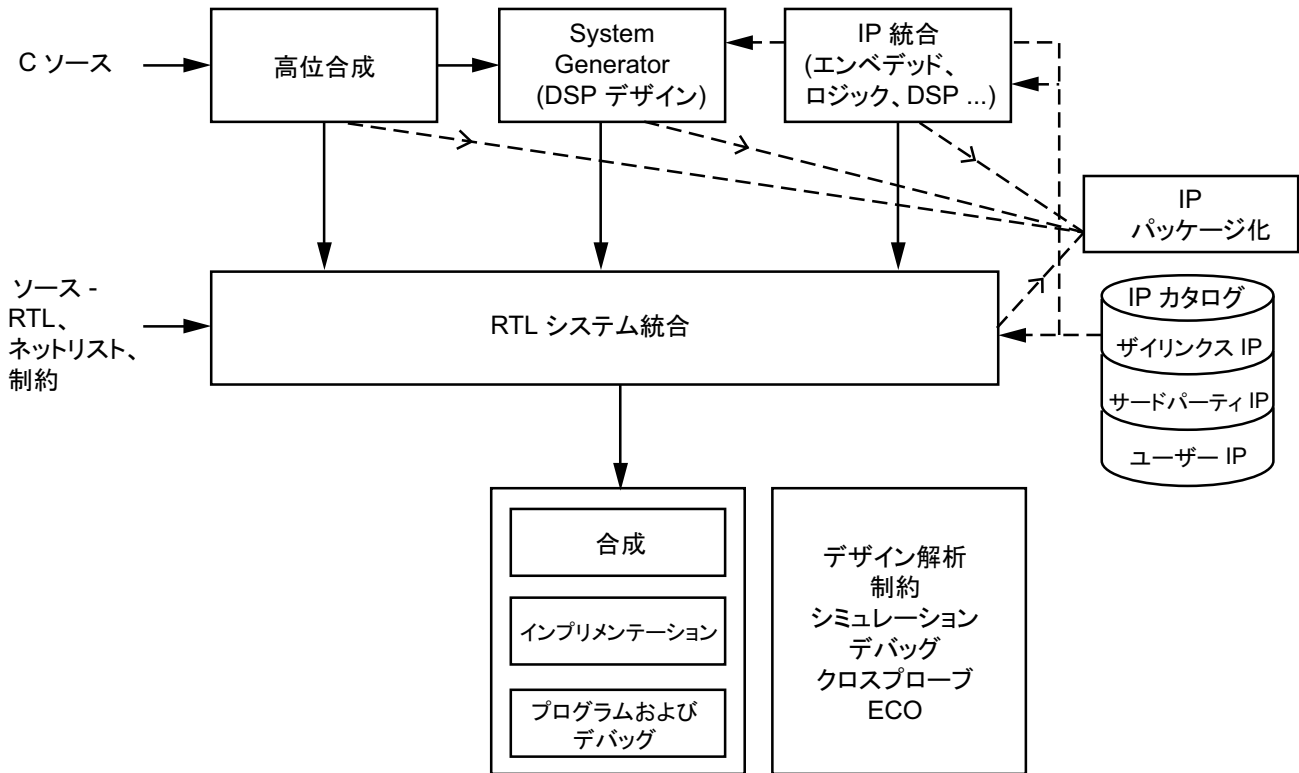
概要

高レベル デザイン フロー

Vivado™ Design Suite では、ザイリックス FPGA のデザインおよび検証に関するタスクを、さまざまな方法で達成できます。従来からの RTL からビットストリームまでの FPGA デザイン フローに加え、Vivado Design Suite では IP 中心のデザインに焦点を置いたシステム レベルの統合フローを提供しています。デザインの解析および検証は、フローの各段階で実行できます。デザイン解析機能には、ロジック シミュレーション、I/O およびクロック配置、消費電力解析、タイミング解析、デザイン ルール チェック (DRC)、デザイン ロジックおよびインプリメンテーション結果の表示、プログラムおよびデバッグなどがあります。

ソリューション全体は、Vivado 統合設計環境 (IDE) というグラフィカル ユーザー インターフェイス (GUI) に統合されています。Vivado IDE では、デザインおよび IP を作成、インプリメント、および検証するインターフェイスが提供されています。また、すべてのフローは Tcl アプリケーションプログラミング インターフェイス (API) を使用しても実行できます。Tcl コマンドは、Tcl プロンプトから入力して対話的に実行するか、Tcl スクリプトに保存して使用できます。Tcl スクリプトを使用して、デザイン解析を含むデザイン フロー全体を実行したり、フローの一部のみを実行できます。

図 1-1 に、Vivado Design Suite での高レベル デザイン フローを示します。



X12973

図 1-1 : Vivado Design Suite の高レベル デザイン フロー

デザイン フローの機能

RTL 開発および解析

Vivado™ IDE には、RTL の開発に役立つ次のような機能が含まれています。

- ソース ファイルを作成および編集するテキスト エディター
- ロジック コンストラクトの例をコピーして使用できる言語テンプレート
- 検索条件を入力してテンプレート ライブラリを検索可能な [Find in Files] 機能

エラボレートされた RTL デザインを開くと、RTL ソース ファイルがエラボレートされ、RTL ネットリストが自動的に読み込まれ、RTL 構造、構文、ロジック定義を確認できます。次の解析およびレポート機能があります。

- RTL のコンパイル検証および構文チェック
- ネットリストおよび回路図の確認
- デザイン ルール チェック (DRC)
- ビヘイビアー シミュレーション
- RTL ポート リストを使用した初期 I/O ピン配置
- 1 つのビューで RTL ソース ファイル内のインスタンス化およびロジック定義などのオブジェクトを選択すると、ほかのビューでも自動選択

詳細は、『Vivado Design Suite ユーザー ガイド : システム レベル デザイン入力』(UG895) を参照してください。

IP の設定とインプリメンテーション

Vivado Design Suite では、IP を設定、インプリメント、検証、および統合する環境が提供されています。IP は、スタンドアロンのモジュールとして、またはシステム レベル デザインの一部として設定および検証できます。IP には、エンベデッド プロセッサ、DSP デジタル信号処理モジュール、C ベースのアルゴリズム デザインなども含まれます。IP-XACT プロトコルに準拠するカスタム IP をパッケージ化して、Vivado IP カタログから使用できるようにすることも可能です。IP カタログから IP を簡単に設定、インスタンス化、および検証できます。ザイリンクス IP は AMBA AXI4 インターコネクト規格を使用しており、高速システム レベル統合が可能です。既存の IP は、デザインで RTL またはネットリストとして使用できます。また、Vivado IDE では CORE Generator™ ツールで作成されたコア (拡張子 .xco のファイル) も使用できます。詳細は、『Vivado Design Suite ユーザー ガイド : IP を使用した設計』(UG896) を参照してください。

ロジック シミュレーション

Vivado IDE に統合されている Vivado シミュレータでは、デザインのシミュレーション、波形ビューアーでの信号の追加および表示、デザインの確認およびデバッグを必要に応じて実行できます。Vivado シミュレータでは、デザインのビヘイビア シミュレーションおよびインプリメント済みデザインのタイミング シミュレーションを実行できます。詳細は、『Vivado Design Suite ユーザー ガイド：ロジック シミュレーション』(UG900) を参照してください。

I/O ピン配置

Vivado IDE には I/O ピン配置環境が含まれており、デバイス パッケージ ピンまたは内部ダイパッドに I/O ポートを正しく割り当てることができます。さまざまな表示ビューおよび表があり、パッケージおよび I/O 関連データを解析できます。詳細は、『Vivado Design Suite ユーザー ガイド：I/O およびクロックの配置』(UG899) を参照してください。

ロジック合成

Vivado Design Suite では、Vivado 合成を使用して、合成 run を設定、起動、および監視できます。合成結果が表示され、レポート ファイルにもアクセスできます。[Log] ビューから合成の警告またはエラーを選択すると、ソース ファイルで該当するロジックがハイライトされます。

複数の合成 run を同時に、または 1 つずつ順に実行できます。Linux システムでは、run をローカルまたはリモートサーバーで実行できます。複数の合成 run を実行すると、複数のネットリストが作成され、Vivado Design Suite プロジェクト内に保存されます。どのバージョンのネットリストでも、Vivado Design Suite 環境に読み込んでデバイスおよびデザインの解析を実行できます。また、I/O ピン配置、フロアプラン、およびインプリメンテーション用に制約を作成できます。デザインを合成して合成済みネットリストを作成すると、クロックとクロック ロジックを解析および配置できるようになり、包括的なデザインルール チェック (DRC) を実行できます。

詳細は、『Vivado Design Suite ユーザー ガイド：合成』(UG901) を参照してください。

ネットリストの解析および制約の定義

Vivado Design Suite では、デザインを解析し、制約を割り当てることができます。デザインの問題を早期に特定するため、インプリメンテーションの前にタイミング シミュレーション、リソース予測、接続性の解析、DRC など、タイミング解析を実行できます。デザイン データは異なるビューにさまざまな形式で表示でき、ビューのデータは連動しています。

Vivado IDE では、チップ内部および外部パッケージの対話型グラフィカル表示で、デバイス リソースを解析できます。タイミング制約および物理制約を適用し、解析できます。詳細は、『Vivado Design Suite ユーザー ガイド：デザイン解析およびタイミング クロージャ』(UG906) を参照してください。

インプリメンテーション

Vivado Design Suite では、インプリメンテーション run を設定、起動、および監視できます。インプリメンテーション run 用に異なるインプリメンテーション オプションを設定し、再利用可能なストラテジを作成できます。たとえば、短いランタイム、パフォーマンス、エリア最適化のストラテジを作成できます。インプリメンテーション run の結果はインタラクティブに表示され、レポート ファイルにも簡単にアクセスできます。

複数のインプリメンテーション run を同時に、または 1 つずつ順に実行できます。Linux では、リモート サーバーを使用できます。制約セットを作成して、さまざまな論理制約、物理制約、代替デバイスを設定して試すことができます。詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) を参照してください。

インプリメンテーション結果の解析およびフロアプラン

さまざまな run 結果を読み込み、解析、制約の設定、およびフロアプランを実行できます。インプリメント済みデザインを開くと、ネットリスト、制約、インプリメンテーション結果が表示されます。複数の run の結果を同時に開くこともできます。Vivado シミュレータを起動してタイミング シミュレーションを実行できます。タイミング解析、消費電力の予測および解析、インプリメンテーション結果およびデザイン コンフィギュレーションの確認および変更が可能です。

また、[Device] ビューで配置、配線、タイミング結果をグラフィカルに表示して解析し、配置配線にマイナーな変更を加えることができます。LUT 論理式、RAM の初期化、PLL コンフィギュレーションなど、デザイン コンフィギュレーションも変更できます。よりよく一貫したパフォーマンスを得るため、インプリメンテーションの前にデザインをフロアプランすることも可能です。

詳細は、『Vivado Design Suite ユーザー ガイド：インプリメンテーション』(UG904) および『Vivado Design Suite ユーザー ガイド：デザイン解析およびタイミング クロージャ』(UG906) を参照してください。

デバイスのプログラム、検証、およびデバッグ

完了したどのインプリメンテーション run に対しても、ビットストリーム ファイルを生成できます。ビットストリーム ファイル生成オプションも設定可能です。iMPACT を起動して、デバイスをコンフィギュレーションおよびプログラムできます。

RTL または合成済みネットリストに ILA (Integrated Logic Analyzer) や ICON (Integrated Controller) などの IP デバッグ コアをコンフィギュレーションおよびインプリメントできます。Vivado IDE で合成済みデザインを開くと、必要なブローブ信号を選択してコアにコンフィギュレーションできます。ビットストリーム ファイルが生成されているどの run に対しても、Vivado ロジック解析を実行できます。

Vivado ロジック解析は Vivado IDE から直接実行でき、配線およびデバイス リソースを解析できます。詳細は、『Vivado Design Suite ユーザー ガイド：プログラムおよびデバッグ』(UG908) を参照してください。

基本的なデザイン フロー

プロジェクト モードと非プロジェクトモードの理解

Vivado™ Design Suite では、ユーザーが使いやすい方法でツールを使用できます。すべてのユーザーがツールを同じように使用するわけではありません。GUI を使用し、ツールでデザイン プロセスおよびデザイン データが自動的に管理されるようになっているのを好むユーザーもいれば、ソースおよびデザイン プロセスを自分で管理し、スクリプトベースのコンパイル スタイル フローを好むユーザーもいます。Vivado Design Suite では、両方がサポートされます。デザイン プロセスの管理には、Vivado プロジェクトを使用します。プロジェクトを使用すると、ディスク上にディレクトリ構造が作成され、それを利用してデザイン ソース ファイル、run の結果、およびプロジェクト ステータスが管理されます。デザイン データ、プロセス、およびステータスが自動的に管理されるようにするには、プロジェクト構造が必要です。そのため、このフローをプロジェクト ベース フローまたはプロジェクト モードと呼びます。

コンパイル スタイル フローは、非プロジェクト バッチ フローまたは非プロジェクト モードと呼ばれます。このフローでは、ソースが現在の場所から読み込まれ、デザインがメモリ内でコンパイルされます。Tcl コマンドを使用し、デザイン パラメーターおよびインプリメンテーション オプションを設定して各手順を個別に実行する必要があります。デザイン プロセスのどの段階でも、Tcl を使用してデザイン チェックポイントを保存し、レポートを生成できます。各デザイン段階で、デザインの解析および制約の設定を実行するために Vivado IDE を開くことができます。メモリ内のアクティブ デザインが表示され、加えた変更は自動的にフローの次の段階に渡されます。変更は、新しい制約ファイルまたはデザイン チェックポイントとして保存できます。ただし、クロスプローブ、デザイン ステータス、IP 統合などのプロジェクト モードの機能の一部は、非プロジェクト モードでは使用できません。

次に、これらのモードについて詳細に説明します。

プロジェクト モードの概要

プロジェクト モードは、ツールとザイリンクスの推奨事項を理解する一番簡単なモードです。このモードでは、Vivado IDE でデザインおよびデザイン プロセスが自動的に管理されます。Vivado IDE の Flow Navigator (図 3-1) を使用して、合成やインプリメンテーションなどの定義済みデザイン フロー手順を実行できます。Vivado IDE でディレクトリ構造が作成され、ソース ファイル、制約、IP (Intellectual Property) データ、合成およびインプリメンテーション run の結果、レポートなどを含むデザインが自動的に管理されます。Vivado IDE では、ソース ファイルのステータス、コンフィギュレーション、デザインの状態なども管理およびレポートされます。このモードには、次の機能も含まれます。

- IP デザインを統合
- 複数の run を作成してさまざまな制約およびコマンド オプションを設定および確認
- インプリメンテーション結果から RTL ソース ファイルへのクロス プローブ
- ノンブロッキング形式の GUI 操作 (デザイン段階を実行中でも、ほかのジョブを GUI を使用して表示、解析、および同時実行可能)

注記 : Tcl コマンドを使用してプロジェクト モードを実行し、必要な場合にのみ Vivado IDE を使用することもできます。

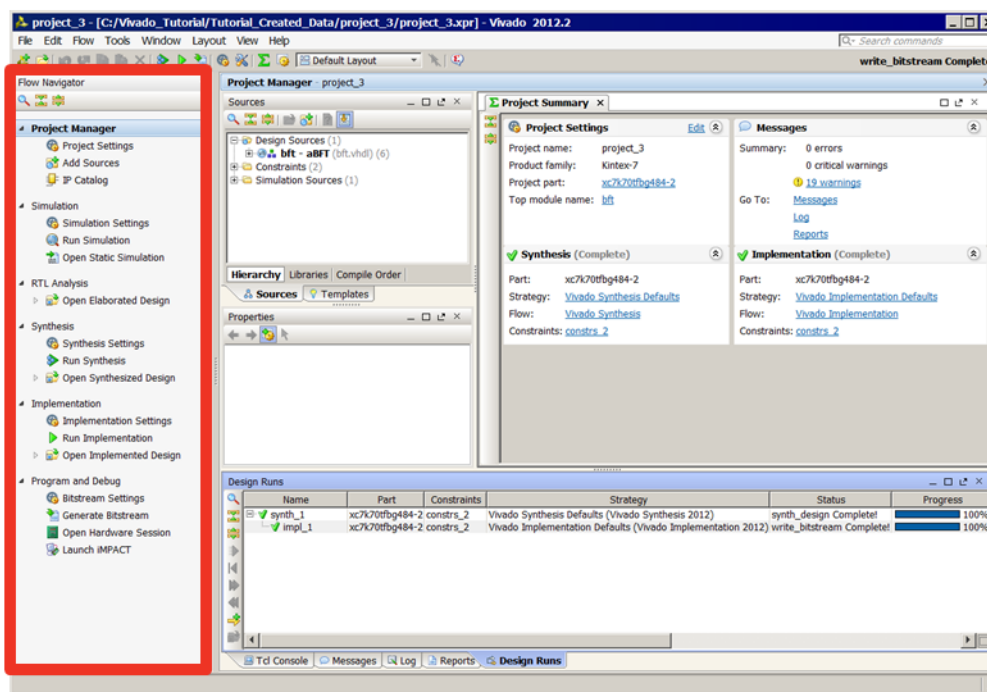


図 3-1 : Vivado IDE の Flow Navigator

プロジェクト モードの利点

プロジェクト モードを使用すると、次のような多数の利点があります。

- Vivado でデザインが自動的に管理されます。次が可能です。
 - プロジェクト ステータス、HDL ソース、制約ファイル、および IP が自動的に管理されます。
 - 合成およびインプリメンテーションの結果が生成されます。
 - インプリメンテーション結果から RTL ソース ファイルへのクロス プローブなど、高度なデザイン解析機能があります。
 - 複数の run を作成し、さまざまな制約およびコマンド オプションを設定して調べることができます。
- GUI、Tcl コマンド、および両方を組み合わせて、デザインのインプリメンテーションおよび解析機能を制御できます。
- GUI モードでは、Flow Navigator (図 3-1) から、合成やインプリメンテーションなどの定義済みデザイン フロー手順を実行したり、さまざまなデザイン解析機能を使用できます。GUI モードを使用するのが、ツールとザイリンクスの推奨事項を理解する一番簡単な方法です。

非プロジェクト モードの概要

このモードでは、Tcl コマンドを使用してメモリ内でフローを実行することで、デザインをコンパイルできます。Tcl コマンドを使用すると、柔軟なデザインの設定および実行、解析およびデバッグが可能です。Tcl コマンドは、バッチ モードで実行するか、Tcl プロンプトまたは Vivado IDE の Tcl コンソールから実行できます。

このフローを使用すると、ユーザーが各デザイン フロー段階を完全に制御できますが、ソース ファイル、レポート、中間結果 (デザイン チェックポイント) を手動で管理する必要があります。さまざまなレポートを生成し、デザイン ルール チェック (DRC) を実行し、インプリメンテーション プロセスのすべての段階のデザイン チェックポイントを書き出すことが可能です。

プロジェクト モードの場合と異なり、複数の run の設定、ソース ファイル管理、クロスプロブ、デザイン状態のレポートなどの機能は含まれません。ソース ファイルがアップデートされるたびに、デザインを手動で実行し直す必要があります。このモードでは、デフォルトではレポートや中間ファイルは作成されませんが、必要に応じて Tcl コマンドを使用してレポートやデザイン チェックポイントを作成できます。

非プロジェクト モードでは、メモリに保存されているデザインを Vivado IDE で開いて、解析したり、ネットリストや制約をアップデートできます。この場合、Vivado IDE で Flow Navigator、IP カタログ、[Sources] ビュー、メッセージやレポートなどのプロジェクト モードの機能は使用できません。また、Vivado IDE 内でソース ファイルや run を開いたり変更したりすることもできません。Tcl コマンドはブロッキング形式なので、Vivado IDE を使用するには、コマンドまたはスクリプトが終了するのを待つ必要があります。

非プロジェクト モードの利点

非プロジェクト モードを使用する主な利点は、デザイン フローの各段階をユーザーが完全に制御できることです。デザインを完全に自由に管理できますが、ユーザーが次を実行する必要があります。

- HDL ソース ファイル、制約、および IP の管理
- 依存性の管理
- 合成およびインプリメンテーションの結果の生成

このモードでは、次が可能です。

- Tcl コマンドを使用して、デザインのコンパイルからフロー全体を実行
- Tcl コマンドまたは GUI を使用して、デザイン解析およびレポート生成を実行

注記: このモードでは、インプリメンテーション結果から RTL ソース ファイルへのクロス プローブはサポートされていません。

プロジェクト モードの場合と異なり、複数の `run` の作成および管理、ソース ファイル管理、デザイン状態のレポートなどの機能は含まれません。ソース ファイルがアップデートされるたびに、デザインを手動で実行し直す必要があります。

また、デフォルトではレポートや中間ファイルは作成されません。ユーザーが手動で生成する必要があります。

プロジェクト モードと非プロジェクトモードの違い

2 つのモードの間には、機能およびコマンドに大きな違いがあります。

機能の違い

プロジェクト 構造により、Vivado IDE でデザインの履歴が管理され、デザインに関する有益な情報が保存されますが、ツール フローはやや固定されたものとなり、`run` を実行したときに標準のレポート ファイルしか生成されません。プロジェクト モードでのみ使用可能な機能の一部を、次に示します。

- ソース ファイルおよびステータスの管理
- Flow Navigator およびプロジェクト サマリ
- 統合されたメッセージおよび標準レポートの自動生成
- RTL へのクロスプローブ
- ツール設定およびデザイン コンフィギュレーションの保存
- 複数の合成 `run` およびインプリメンテーション `run` の使用
- 制約セットの使用と管理
- `run` の結果およびステータスの管理
- IP カタログを使用した IP の設定および統合

非プロジェクト モードを使用する場合は、`Tcl` コマンドを使用してすべてを個別に実行します。すべての処理はメモリ内で実行されるので、ファイルやレポートは自動的に生成されません。デザインをコンパイルするたびに、ソースの定義、ツールおよびデザイン コンフィギュレーション パラメーターの設定、すべての `Tcl` コマンドの実行、必要なレポート ファイルの生成をユーザーの責任で実行する必要があります。ディスク上にプロジェクトは作成されないため、ソース ファイルは元の場所に配置されたままになり、指定した出力のみが作成されます。このフローでは、`Tcl` API の優れた機能をすべて活用でき、デザイン プロセス全体を完全に制御できます。

表 3-1 に、プロジェクト モードと非プロジェクト モードの違いをまとめます。

表 3-1: プロジェクト モードと非プロジェクト モードの違い

フローの要素	プロジェクト モード	非プロジェクト モード
デザイン ソース ファイルの管理	自動	手動
フロー ナビゲーション	ガイド	手動
フローのカスタマイズ	制限あり	制限なし
レポート	自動	手動
GUI 操作	ノンブロッキング形式	ブロッキング形式
解析段階	デザインのみ	デザインおよびチェックポイント

コマンドの違い

プロジェクト モードの Tcl コマンドは、非プロジェクト モードで使用されるコマンドとは大きく異なります。主な違いは次のとおりです。

- プロジェクトにソースを追加するには、`add_files` Tcl コマンドを使用します。ソースをプロジェクトにコピーし、プロジェクト ディレクトリ構造内で別のバージョンのソースとして管理できます。
- `run` を作成して、合成およびインプリメンテーション プロセスと `run` ステータスを自動的に管理します。個別の合成コマンドおよびインプリメンテーション コマンドを包括するラッパー コマンド `launch_runs` が使用されます。

図 3-2 に、プロジェクト モードと非プロジェクト モードの Tcl コマンドの違いを示します。プロジェクト モードには、ほとんどの場合に Tcl コマンドが実行される GUI 操作が含まれます。Vivado IDE の Tcl コンソールに表示される Tcl コマンドは、`vivado.jou` ファイルにも保存されます。このファイルを使用して、どちらのモードでも使用できるスクリプトを開発できます。

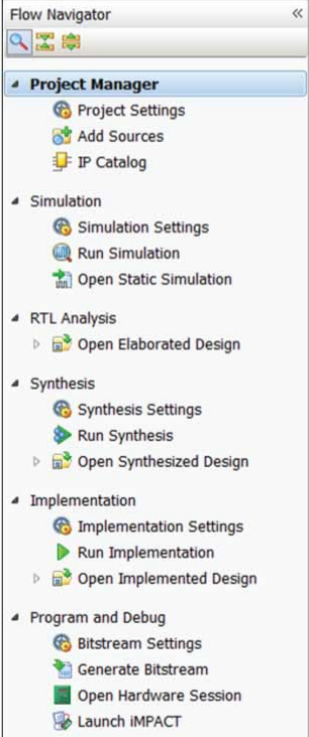
Project Mode		Non-Project Mode
GUI	Tcl Script	Tcl Script
	<pre> create_project ... add_files ... import_files ... launch_run synth_1 wait_on_run synth_1 open_run synth_1 report_timing_summary launch_run impl_1 wait_on_run impl_1 open_run impl_1 report_timing_summary launch_run impl_1 -to_step_write_bitstream wait_on_run impl_1 </pre>	<pre> read_verilog ... read_vhdl ... read_ip ... read_xdc ... read_edif ... synth_design report_timing_summary write_checkpoint opt_design place_design route_design report_timing_summary write_checkpoint write_bitstream </pre>

図 3-2: プロジェクト モードと非プロジェクトモードの Tcl コマンドの違い

プロジェクト モードの使用

プロジェクト モードは、GUI およびスクリプト ベースの両方でサポートされています。1 つのプロジェクトで IDE と Tcl コンソールを交互に使用できます。ただし、プロジェクト モードは Vivado IDE で使用するのが有利です。Vivado IDE でプロジェクトを開くか作成すると、デザインの現在の状態、run の結果、生成されたレポートおよびメッセージが表示されます。ソースの作成および変更、制約の適用、デバッグ情報の適用、ツール設定の指定、デザイン タスクの実行などが可能です。

Flow Navigator を使用すると、デザイン フロー全体を順を追って実行できます。[Generate Bitstream] をクリックすると、デザインが合成およびインプリメントされてから、ビットストリーム ファイルが生成されます。RTL エラボレーション、合成、およびインプリメンテーションの後にデザインを開き、解析および制約の定義を実行できます。デザインを開くと、ネットリストおよび制約がターゲット デバイスに対してコンパイルされ、デザインがメモリに読み込まれます。デザインがメモリに読み込まれると、さまざまな解析およびレポート機能が使用できるようになります。

デザインをさまざまな条件および視点から解析できます。制約およびデザインの変更を適用し、保存できます。詳細は、『[デザインを開く](#)』または『Vivado Design Suite ユーザー ガイド : デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

プロジェクトの作成

Create Project ウィザードを使用すると、プロジェクトを簡単に作成できます。表示される一連のページでプロジェクトを定義していきます。プロジェクトを保存する名前と場所、プロジェクト タイプ (RTL、ネットリストなど)、ターゲット パーツなどを指定します。RTL、IP、XDC/SDC 制約、シミュレーション テストベンチ、System Generator (XMP) からの DSP モジュール、Vivado 高位合成 (HLS) からのソース、Xilinx Platform Studio (XPS) からのプロセッサ モジュール、XPS からのメモリ初期化ファイル (拡張子.bmm)、デザインに関する文書など、さまざまなタイプのソースを追加できます。ソースを追加する際、ソースを元の場所から参照するか、プロジェクト ディレクトリにコピーするかを指定できます。Vivado IDE では各ファイルのタイムスタンプが監視され、ステータスがレポートされます。ファイルが変更されると、ソースまたはデザインのステータスが最新でなくなったことが示されます。詳細は、『Vivado Design Suite ユーザー ガイド : システム レベル デザイン入力』(UG895) を参照してください。

Flow Navigator の理解

Flow Navigator では、プロジェクトの設定、合成、インプリメンテーション、ビットストリームの生成などの主なデザインプロセス タスクを制御します。Flow Navigator で使用可能なコマンドおよびオプションは、デザインのステータスによって異なります。実行できない手順は淡色表示されます。図 3-3 に Flow Navigator を示します。

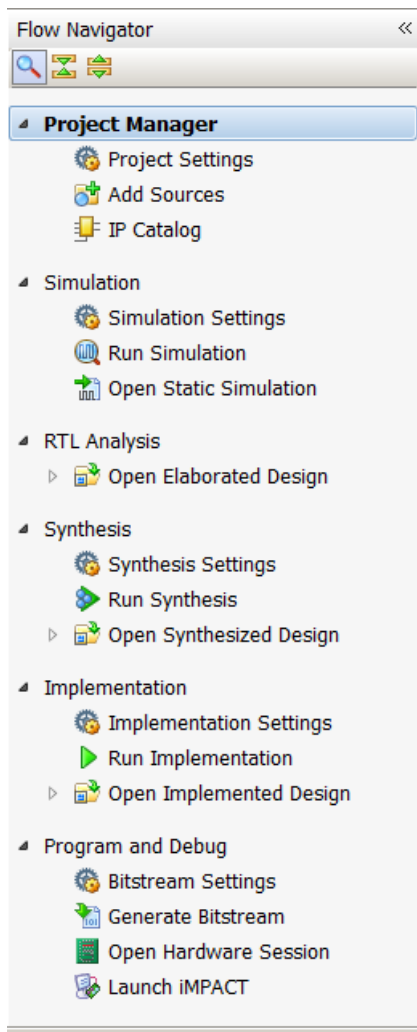


図 3-3 : Flow Navigator

これらのデザイン タスクが完了したら、デザイン結果を開き、解析したり、制約を適用したりできます。これには、Flow Navigator で [Open Elaborated Design]、[Open Synthesized Design]、または [Open Implemented Design] をクリックします。詳細は、「[デザインを開く](#)」を参照してください。

これらのデザイン表示には、デザイン フローに適したよく使用されるコマンドのセットが表示されます。Flow Navigator でこれらのコマンドをクリックすると、既に関いている場合はデザインが開き、操作が実行されます。

図 3-4 に合成に関連するコマンドを示します。

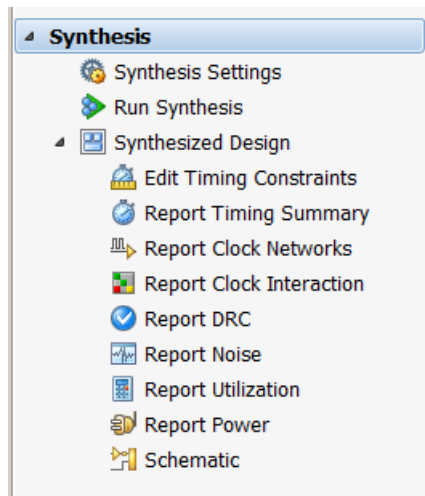
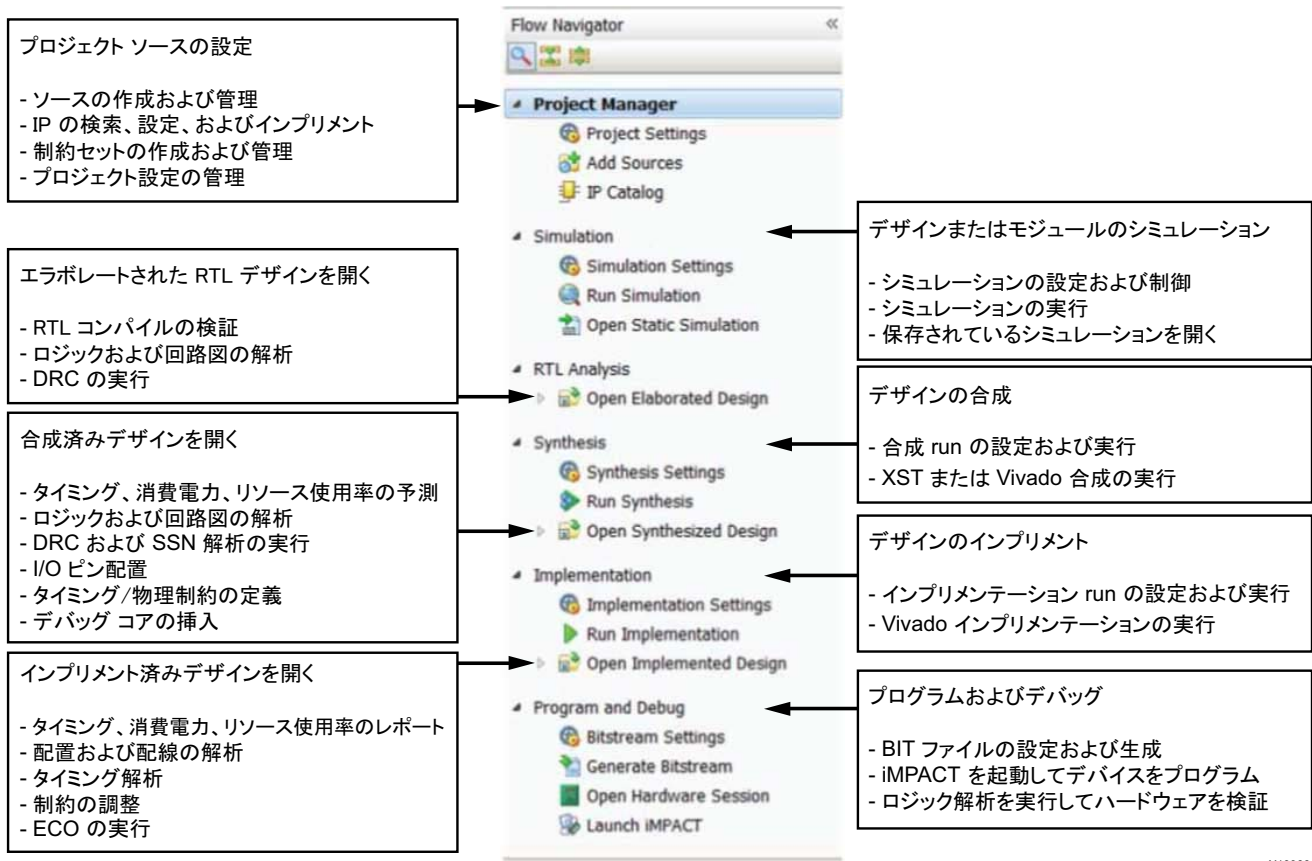


図 3-4 : Flow Navigator の [Synthesis] セクション

RTL プロジェクトでの Flow Navigator の使用

図 3-5 に、RTL ソースを入力として使用した場合のデザイン フローを示します。



X110666

図 3-5 : Flow Navigator (RTL プロジェクト)

合成済みネットリスト プロジェクトでの Flow Navigator の使用

図 3-6 に、合成済みネットリスト ベースのプロジェクトのデザイン フローを示します。

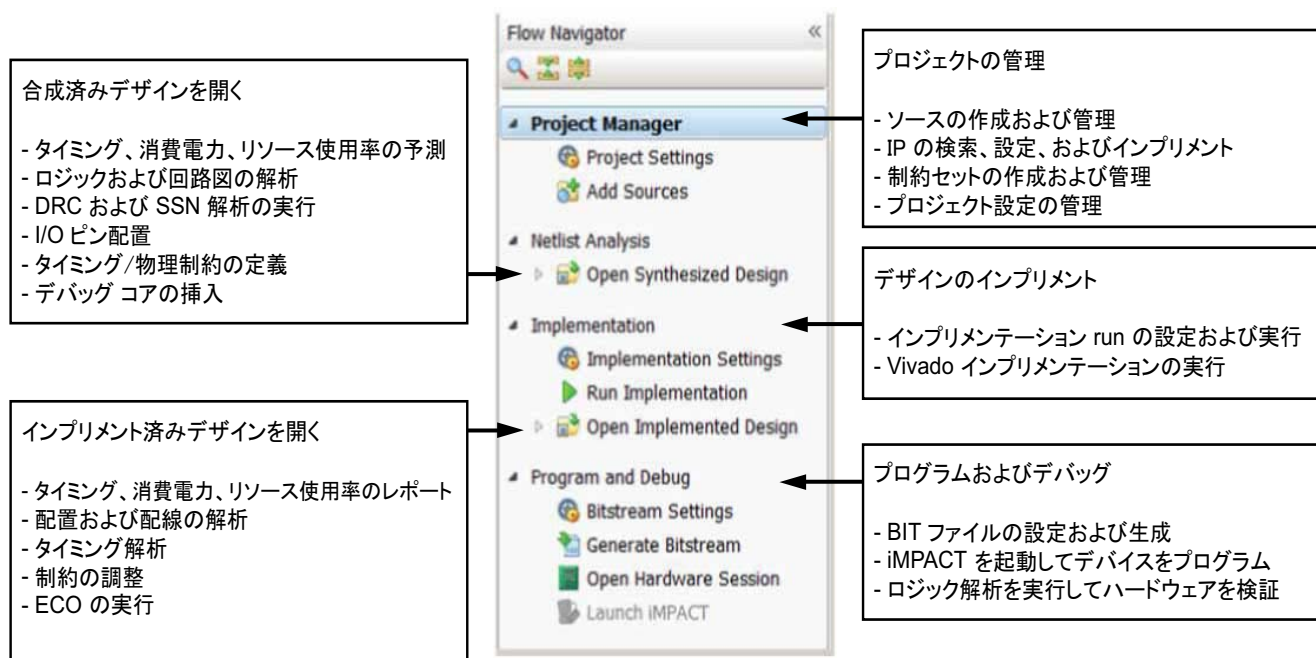


図 3-6 : Flow Navigator (合成済みネットリスト プロジェクト)

IP の設定と検証

IP カタログの使用

IP カタログには、ザイリンクス LogiCORE™ IP すべてと、[Update IP Catalog] コマンドを使用してカタログに追加されたユーザー IP およびサードパーティ IP が表示されます。カタログはカテゴリ別に分類されており、IP のタイプ、バージョン、ライセンスなどの情報が含まれています。IP カタログでは、次を実行できます。

- IP をダブルクリックして [Customize IP] ダイアログ ボックスを表示
- [Sources] ビューの IP を右クリックして [Generate] をクリックし、IP ソースを生成。インスタンシエーション テンプレートが生成され、システム レベル デザインに統合できるようになります。Vivado Design Suite IP は、ネットリストではなく RTL ソースとして作成されます。
- 合成およびインプリメンテーションを実行して、デザインのほかの部分と共に IP を合成およびインプリメント

IP の検証と再利用

スタンドアロンの IP のパフォーマンスを検証する必要がある場合があります。スタンドアロン IP の合成、シミュレーション、インプリメンテーション、および解析を実行することにより、IP のパフォーマンス要件が満たされているかを確認できます。検証した IP ネットリストがデザインのほかの部分と共に再合成されないようにすることも可能です。これには、プロジェクトの最上位モジュールを IP のモジュールに設定するか、IP のみを含む別のプロジェクトを作成してフローを実行します。コンテキスト外で合成するパラメーターを使用して、合成で I/O バッファがインスタンシエートされないようにできます。Flow Navigator のコマンドを使用して、IP をインプリメントおよび検証します。

検証したスタンドアロン IP のソースを新しいプロジェクトに追加できます。スタンドアロン IP が合成されている場合、ネットリストがデザイン ソースとして使用されるので、検証された IP ネットリストが保持されます。合成結果 (NGC または EDIF ネットリスト) のない既存のスタンドアロン IP モジュールをプロジェクトに追加すると、合成を実行したときに IP もデザインのほかの部分と共に合成されます。

IP コア生成の一部として、オプションでサンプル デザインが作成されます。最上位モジュールをサンプル IP モジュールに設定すると、デザイン プロジェクトのコンテキストでスタンドアロン IP を検証できます。サンプル デザイン ディレクトリに合成済みネットリストが存在する場合は、最上位デザインの合成中に使用されます。つまり、スタンドアロンで検証した IP ネットリストはデザインのほかの部分と共に再合成されません。

run の作成および管理

合成およびインプリメンテーション run の設定

合成およびインプリメンテーションでの処理を制御する設定が多数あります。プロジェクト モードでは、これらの設定を run ストラテジを使用して適用します。run ストラテジは、run の設定パラメーターの組み合わせを保存したものです。ザイリンクスでは、合成およびインプリメンテーションの実行用にいくつかの run ストラテジを提供しています。カスタム run 設定も適用できます。

合成の設定

合成のプロジェクト設定を変更するには、次のいずれかの操作を実行します。

- Flow Navigator で [Synthesis Settings] をクリックします。
- Flow Navigator で [Project Manager] → [Project Settings] をクリックし、左側のペインで [Synthesis] をクリックします。
- [Tools] → [Project Settings] をクリックし、左側のペインで [Synthesis] をクリックします。
- [Project Summary] ビューで合成に関するリンクをクリックします。
- [Design Runs] ビューで合成 run を右クリックし、[Change Run Settings] をクリックします。

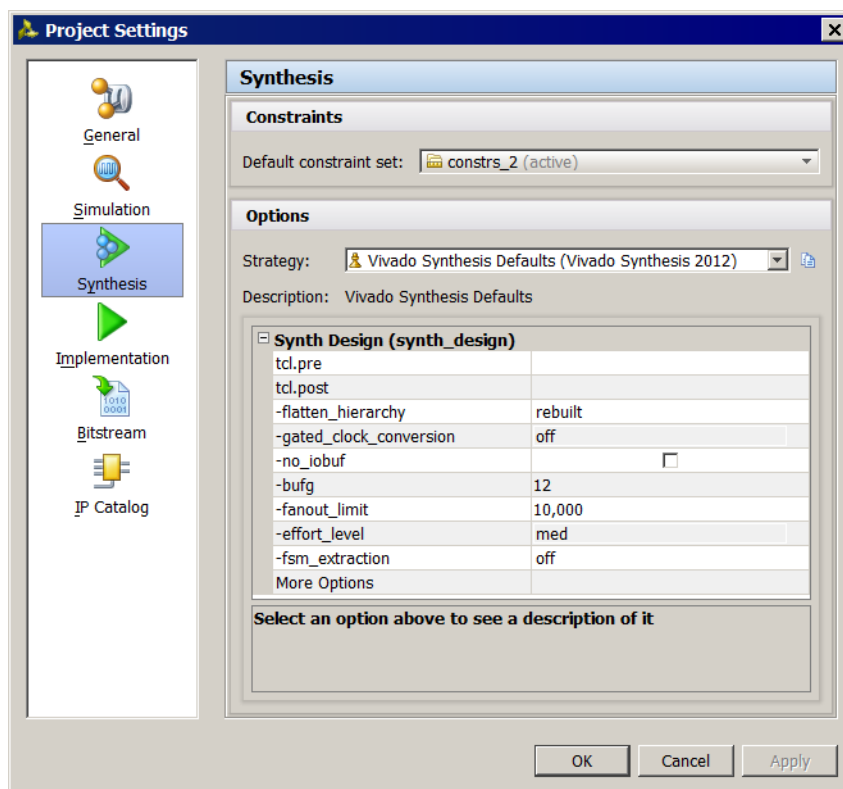


図 3-7 : [Project Settings] ダイアログ ボックスの [Synthesis] ページ

カスタム ストラテジを作成するには、必要な設定を指定してから、右クリックして [Save Strategy As] をクリックします。

Vivado 合成 run に制約セットを選択できます。Vivado 合成はタイミングドリブンであり、タイミング制約を使用して最良のパフォーマンスを達成します。合成とインプリメンテーションに異なる制約セットを設定できます。詳細は、『Vivado Design Suite ユーザー ガイド : システム レベル デザイン入力』(UG895) および『Vivado Design Suite ユーザー ガイド : 制約の使用』(UG903) を参照してください。

XST の実行

Vivado IDE では、ISE Design Suite の XST 合成ツールまたは Vivado 合成のいずれかを使用できます。XST ストラテジを使用すると、XST 合成が実行されます。どちらの結果にも Vivado インプリメンテーションが使用されます。

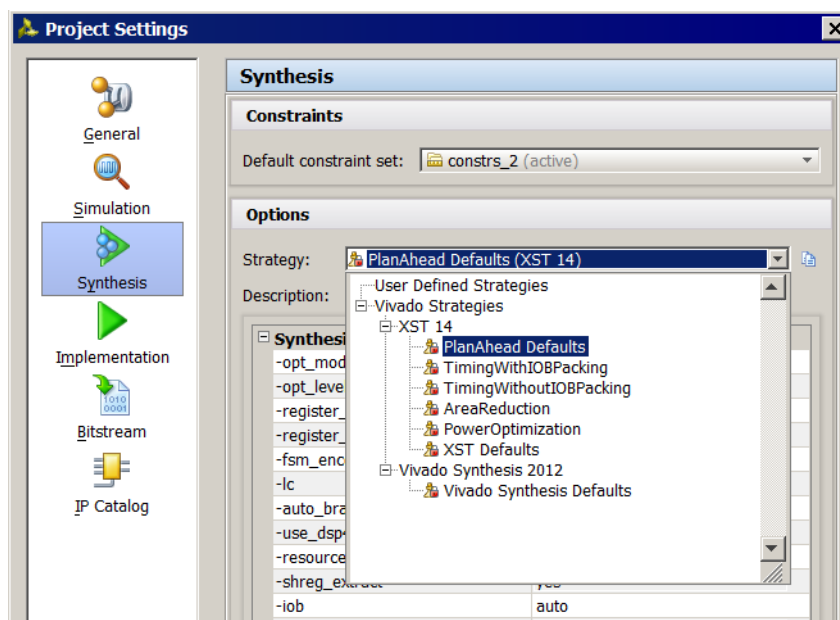


図 3-8 : XST 合成ストラテジの選択

インプリメンテーションの設定

インプリメンテーションのプロジェクト設定を変更するには、次のいずれかの操作を実行します。

- Flow Navigator で [Implementation Settings] をクリックします。
- Flow Navigator で [Project Manager] → [Project Settings] をクリックし、左側のペインで [Implementation] をクリックします。
- [Tools] → [Project Settings] をクリックし、左側のペインで [Implementation] をクリックします。
- [Project Summary] ビューでインプリメンテーションに関するリンクをクリックします。
- [Design Runs] ビューでインプリメンテーション run を右クリックし、[Change Run Settings] をクリックします。

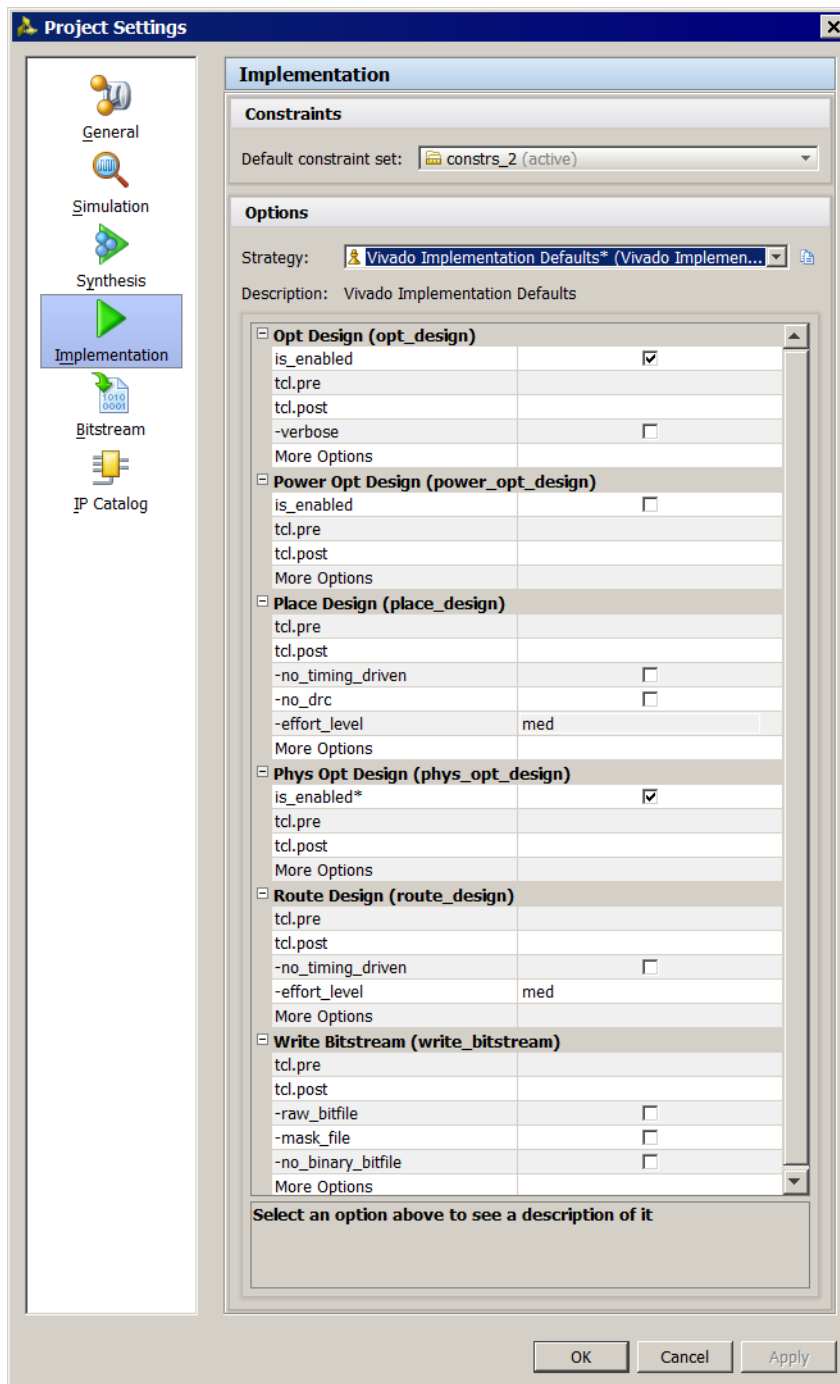


図 3-9 : [Project Settings] ダイアログ ボックスの [Implementation] ページ

[Project Settings] ダイアログ ボックス [Implementation] ページで、run に適用する制約セットとストラテジを選択します。カスタム ストラテジを作成するには、必要な設定を指定してから、右クリックして [Save Strategy As] をクリックします。

合成とインプリメンテーションに異なる制約セットを指定できます。合成またはインプリメンテーションにどの制約ファイルが使用されているかを確認するには、[Sources] ビューで制約ファイルを選択し、[Source File Properties] ビューを参照します。[Used In] セクションで run への制約の割り当てを変更できます。詳細は、『Vivado Design Suite ユーザーガイド：システム レベル デザイン入力』(UG895) および『Vivado Design Suite ユーザーガイド：制約の使用』(UG903) を参照してください。

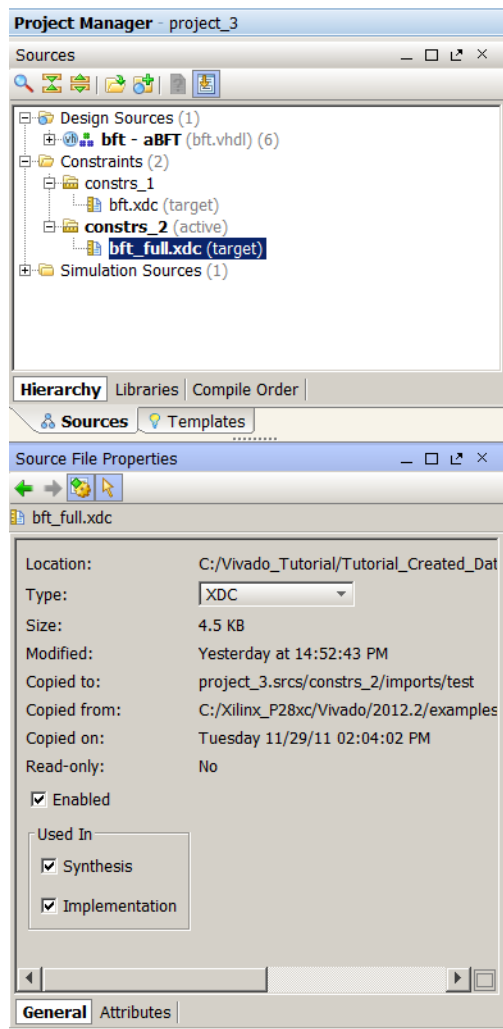


図 3-10 : 合成およびインプリメンテーション用の制約の選択

アクティブな制約セットは、[Sources] ビューで太字で表示されます。この制約セットが、開いている run またはデザインにデフォルトで使用されます。別の制約セットをアクティブにするには、制約セットを右クリックして [Make Active] をクリックします。

合成およびインプリメンテーション run の実行

[Project Settings] ダイアログ ボックスで合成およびインプリメンテーションの設定を指定したら、次のように合成 run またはインプリメンテーション run を実行できます。

1. Flow Navigator で [Run Synthesis]、[Run Implementation]、または [Generate Bitstream] をクリックします。
2. [Design Runs] ビューで run を選択し、右クリックして [Launch Runs] をクリックするか、またはツールバーの [Launch Selected Runs] ボタンをクリックします。
3. [Flow] → [Run Synthesis]、[Flow] → [Run Implementation]、または [Flow] → [Generate Bitstream] をクリックします。

複数の run の作成および管理

複数の合成 run またはインプリメンテーション run を作成して、さまざまな制約およびツール設定を試してみることができます。複数の run を作成するには、次の手順に従います。

1. Flow Navigator で [Synthesis] または [Implementation] を右クリックします。
2. [Create Synthesis Runs] または [Create Implementation Runs] をクリックします。
3. Create New Runs ウィザードで、制約セットおよびターゲット パーツを選択します。

インプリメンテーション run の作成では、複数の合成 run が存在する場合はネットリストも選択できます。異なるストラテジを指定した複数の run を作成できます。

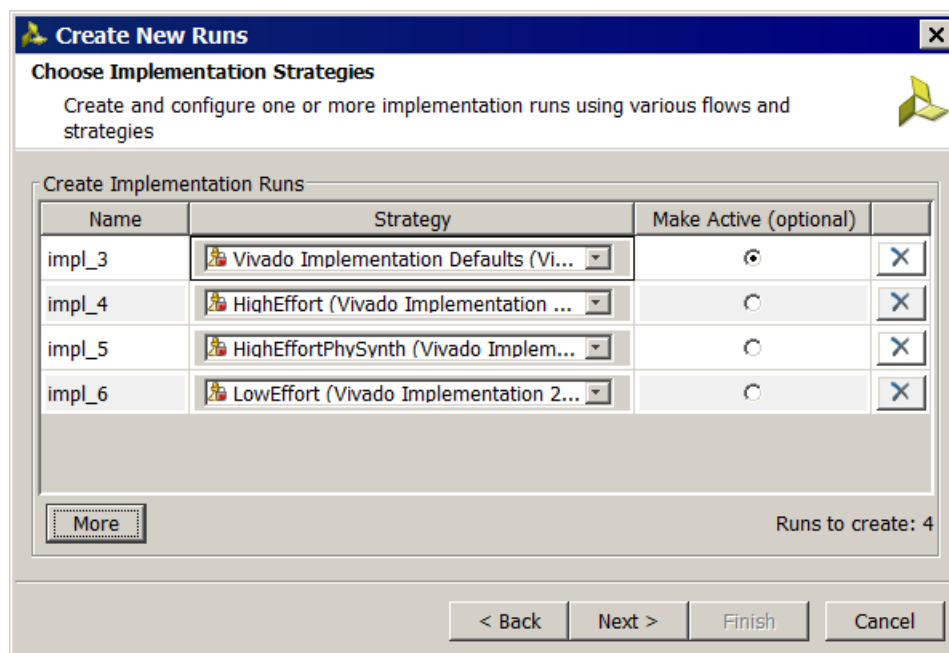


図 3-11 : Create New Runs ウィザード

run が複数ある場合、実行オプションを指定できます。選択した run を順次実行するか、複数のローカル プロセッサで同時に実行できます。Linux では、リモート ホストを設定して使用することもできます。

[Design Runs] ビューでの run の制御

run のステータスおよび情報は、[Design Runs] ビューに表示されます。run を制御する多くのコマンドは、ポップアップメニューから実行できます。

[Design Runs] ビューから複数の run を制御できます。run が複数ある場合、アクティブな run が太字で示されます。Vivado IDE には、アクティブな run の実行結果が表示されます。[Project Summary]、レポート、およびメッセージに表示されるのは、すべてアクティブ run に関する情報です。

Flow Navigator で [Open Synthesized Design] または [Open Implemented Design] をクリックすると、デフォルトでアクティブ デザインが開きます。別の run をアクティブにするには、run を右クリックして [Make Active] をクリックします。Vivado IDE に新たにアクティブになった run の情報が表示されます。合成済み run またはインプリメント済み run をダブルクリックすると、Vivado IDE にそのデザインが開きます。

Name	Part	Constraints	Strategy	Status	Progress
synth_1 (active)	xc7k70tfgb4...	constrs_2	Vivado Synthesis Defaults (Vivado Synthesis 2012)	synth_design Complete!	100%
impl_1	xc7k70tfgb484-2	constrs_2	Vivado Implementation Defaults (Vivado Implementation 2012)	Implementation Out-of-date	100%
impl_3 (active)	xc7k70tfgb4...	constrs_2	Vivado Implementation Defaults (Vivado Implementa...	Running Design Initialization...	0%
impl_4	xc7k70tfgb484-2	constrs_2	HighEffort (Vivado Implementation 2012)	Not started	0%
impl_5	xc7k70tfgb484-2	constrs_2	HighEffortPhySynth (Vivado Implementation 2012)	Not started	0%
impl_6	xc7k70tfgb484-2	constrs_2	LowEffort (Vivado Implementation 2012)	Not started	0%
synth_2	xc7k70tfgb484-2	constrs_2	Vivado Synthesis Defaults (Vivado Synthesis 2012)	Not started	0%
impl_2	xc7k70tfgb484-2	constrs_2	Vivado Implementation Defaults (Vivado Implementation 2012)	Not started	0%

図 3-12 : [Design Runs] ビューでの run の制御

メッセージの表示

プロジェクト モードを使用している場合、Vivado IDE のメッセージはすべて、[Messages] に重要度 (エラー、クリティカル警告、警告、および情報) 別に表示されます。[Messages] ビューの上部にあるチェック ボックスのオン/オフを切り替えて、表示するメッセージを選択できます。メッセージのカテゴリを展開表示すると、個々のメッセージを表示できます。多くのメッセージにはリンクが含まれており、クリックすると RTL ファイルの該当する行が開きます。

レポートの表示

launch_runs Tcl コマンドの一部として、いくつかの標準レポートが生成されます。これらのレポートは、[Reports] ビューにリストされます。レポートをダブルクリックすると、テキスト エディターで開きます。[Tcl Console] ビューで該当する Tcl コマンドを入力して、カスタム レポートを作成できます。

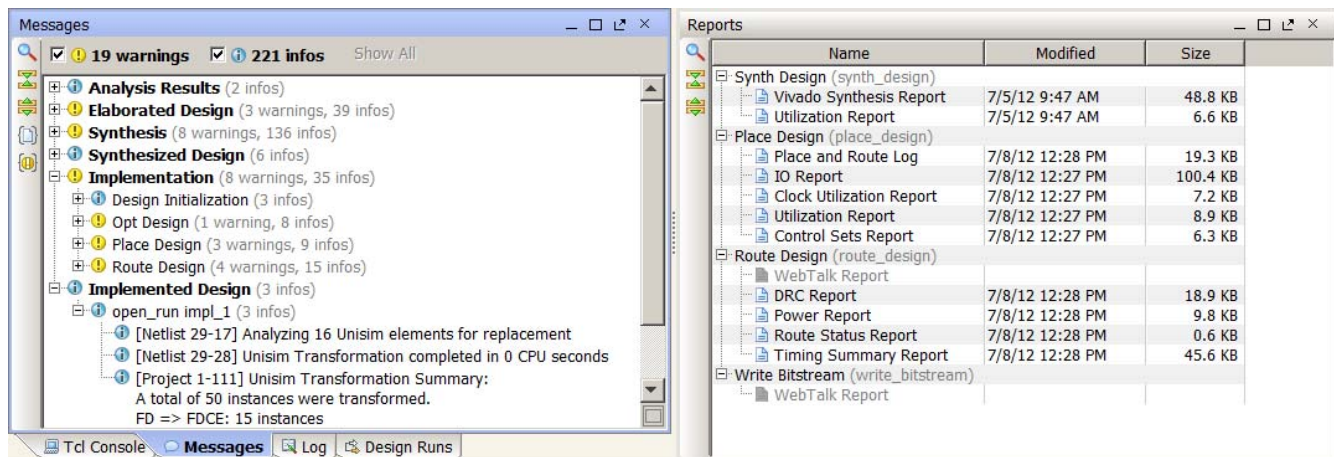


図 3-13 : [Messages] ビューと [Reports] ビュー

デザインを開く

RTL エラボレーション、合成、またはインプリメンテーションの後デザインを開いて、デザインを解析したり、制約を適用したりできます。デザインを開くと、ネットリストおよび制約がターゲット デバイスに対してコンパイルされ、デザインがメモリに読み込まれます。デザインがメモリに読み込まれると、Vivado IDE のさまざまな解析およびレポート機能が使用できるようになります。

Vivado Design Suite では、開いたり、保存したり、閉じたりできる「デザイン」が使用されます。新しいデザインを開くと、メモリ容量を確保するため、既に関いているデザインを閉じるかどうかを確認するメッセージが表示されます。既に関いているデザインを閉じる必要はなく、複数のデザインを同時に開いておくことができます。デザインが開いているときに制約を変更すると、変更を元の XDC ソース ファイルに保存するか、新しい制約セットを作成するかを選択するダイアログ ボックスが表示されます。詳細は、『Vivado Design Suite ユーザー ガイド：システム レベル デザイン入力』(UG895) および『Vivado Design Suite ユーザー ガイド：制約の使用』(UG903) を参照してください。

開いているデザインに実行可能な操作の詳細は、『Vivado Design Suite ユーザー ガイド：デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

エラボレートされた RTL デザインを開く

Vivado Design Suite でエラボレートされた RTL デザインを開くと、RTL ネットリストがコンパイルされ、ターゲット パーツに対して物理制約およびタイミング制約が適用されます。エラボレートされた RTL デザインはメモリに読み込まれ、必要に応じて解析および変更して、デザインを完了させることができます。

Flow Navigator で [Open Elaborated Design] をクリックすると、指定されている最上位モジュール名を使用して RTL デザインがコンパイルされます。基本的な RTL エラボレーションが実行され、制約が適用されて、デザインがメモリに読み込まれます。コンパイル中のメッセージおよびエラーは [Messages] ビューに表示され、メッセージのリンクをクリックすると、RTL ソース ファイルの該当するコード行が開きます。詳細は、『Vivado Design Suite ユーザー ガイド : システム レベル デザイン入力』(UG895) を参照してください。

RTL デザインでは、デザインが論理的に正しいかどうかを解析できます。ロジックが正しくコンパイルされたか、欠けているモジュールはないか、インターフェイスの不一致がないかなどをチェックできます。[Messages] ビューでメッセージのリンクをクリックすると、テキスト エディターで RTL ファイルの該当する行が開きます。[Schematic] ビューには、RTL ベースのロジック コンストラクトを使用して、RTL の接続が表示されます。RTL エラボレーションでは、FPGA テクノロジーへのマップは実行されません。[Schematic] ビューでは、ロジックの接続および階層をさまざまな方法で調べることができます。[Schematic] ビューでロジックを選択したり、テキスト エディターで RTL ファイルの特定の行を確認したりできます。詳細は、『Vivado Design Suite ユーザー ガイド : デザイン解析およびクロージャ テクニク』(UG906) を参照してください。

レジスタなど、ロジック階層の特定のロジック インスタンスに定義されている制約は、RTL エラボレーションでは解決できない場合があります。エラボレーション中に生成されるロジック名と階層は、合成中に生成されるものと一致しない可能性があります。そのため、これらの制約が定義されている場合、RTL デザインをエラボレートしたときに制約マップに関する警告またはエラー メッセージが表示されることがあります。デザインで合成を実行した場合に、同じ警告メッセージが表示されることはないはずです。

エラボレートされた RTL デザインでは、I/O ポートをインタラクティブに設定および割り当てできます。Vivado IDE の I/O 配置機能を使用して、インタラクティブに I/O ポートを割り当て、DRC を実行できます。クロックおよびロジック制約が正しく解決されるようにするため、I/O ピン配置は可能な限り合成後に実行してください。合成後には、より包括的な DRC が実行されます。詳細は、『Vivado Design Suite ユーザー ガイド : I/O およびクロックの配置』(UG899) を参照してください。

[Report DRC] コマンドをクリックすると、基本的な RTL および I/O の DRC が実行され、非同期クロック、ラッチなどのロジックの問題が特定されます。

合成済みデザインを開く

Vivado Design Suite で合成済みデザインを開くと、合成済みネットリストが開き、ターゲット パーツに対して物理制約およびタイミング制約が適用されます。合成済みデザインのエレメントがメモリに読み込まれ、必要に応じてそれらを解析および変更して、デザインを完了させることができます。変更は、制約ファイル、ネットリスト、デバッグコア、およびコンフィギュレーションに保存できます。

合成済みデザインに対して実行できるタスクは多数あります。早期のタイミング、消費電力、使用率予測を実行すると、デザインが要件を満たすことができるのかどうかを判断できます。Vivado IDE のビューを使用して、デザインをさまざまな方法で解析できます。1 つのビューでオブジェクトを選択すると、ほかのビューでも選択されます。タイミング制約を適用し、タイミング解析を実行できます。I/O ポートの物理制約、フロアプラン、デザイン コンフィギュレーションをインタラクティブに定義できます。

[Messages]、[Schematic]、[Device]、[Package]、[Find Results] などのさまざまなビューから、RTL ファイルの問題のある行にクロスプローブできます。[Schematic] ビューでは、ロジックの接続および階層をインタラクティブにさまざまな方法で調べることができます。詳細は、『Vivado Design Suite ユーザー ガイド：デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

合成済みデザインでは、I/O ポートをインタラクティブに設定および割り当てできます。Vivado IDE の I/O 配置機能を使用して、インタラクティブに I/O ポートを割り当て、DRC を実行できます。詳細は、『Vivado Design Suite ユーザー ガイド：I/O およびクロックの配置』(UG899) を参照してください。

[Report DRC] コマンドをクリックすると、包括的な DRC が実行され、ロジックの問題が特定されます。詳細は、『Vivado Design Suite ユーザー ガイド：デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

合成済みデザインでは、デバッグ コア ロジックを設定およびインプリメントできます。デバッグ用の信号を [Schematic] または [Netlist] ビューで選択できます。その後デバッグ コアをコンフィギュレーションし、デザインに挿入します。コアのロジックおよび接続は、デザインの合成がアップデートされても、できる限り保持されます。

合成済みデザインを開くには、次のいずれかを実行します。

- Flow Navigator で [Synthesis] → [Open Synthesized Design] をクリックします。
- [Design Runs] ビューで run の名前をダブルクリックします。
- [Flow] → [Open Synthesized Design] をクリックします。
- Flow Navigator で [Synthesis] → [New Synthesized Design] をクリックします。
- [Flow] → [New Synthesized Design] をクリックします。

アクティブな制約セットおよびターゲット デバイスを使用して合成済みネットリストを開くか、別の制約セットとターゲット デバイスを指定して、メモリの合成済みデザインを開くことができます。図 3-14 に示すように、次を指定して新しい合成済みデザインを定義できます。

- [Design name] : ビュー バナーに表示される名前を入力します。デザインは、現在のセッション中のみメモリに保存されます。
- [Synthesis run] : 指定した完了済み合成 run からのネットリストを使用します。このオプションは、プロジェクトに複数の合成 run がある場合にのみ表示されます。
- [Constraint set] : ネットリストに対して開く既存の制約セットを選択します。
- [Part] : ターゲット パーツを選択します。

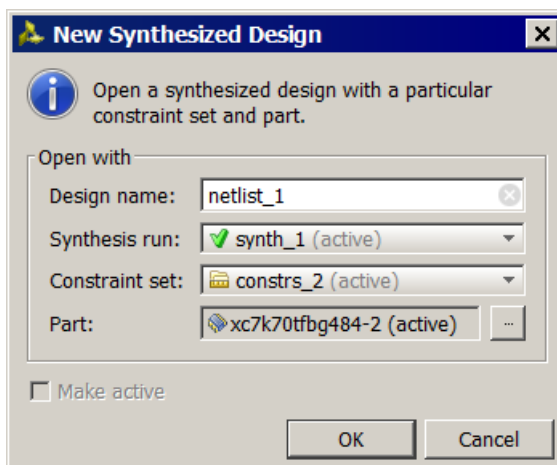


図 3-14 : 新規合成済みデザイン

図 3-15 に、合成済みデザインを開いたときの表示を示します。

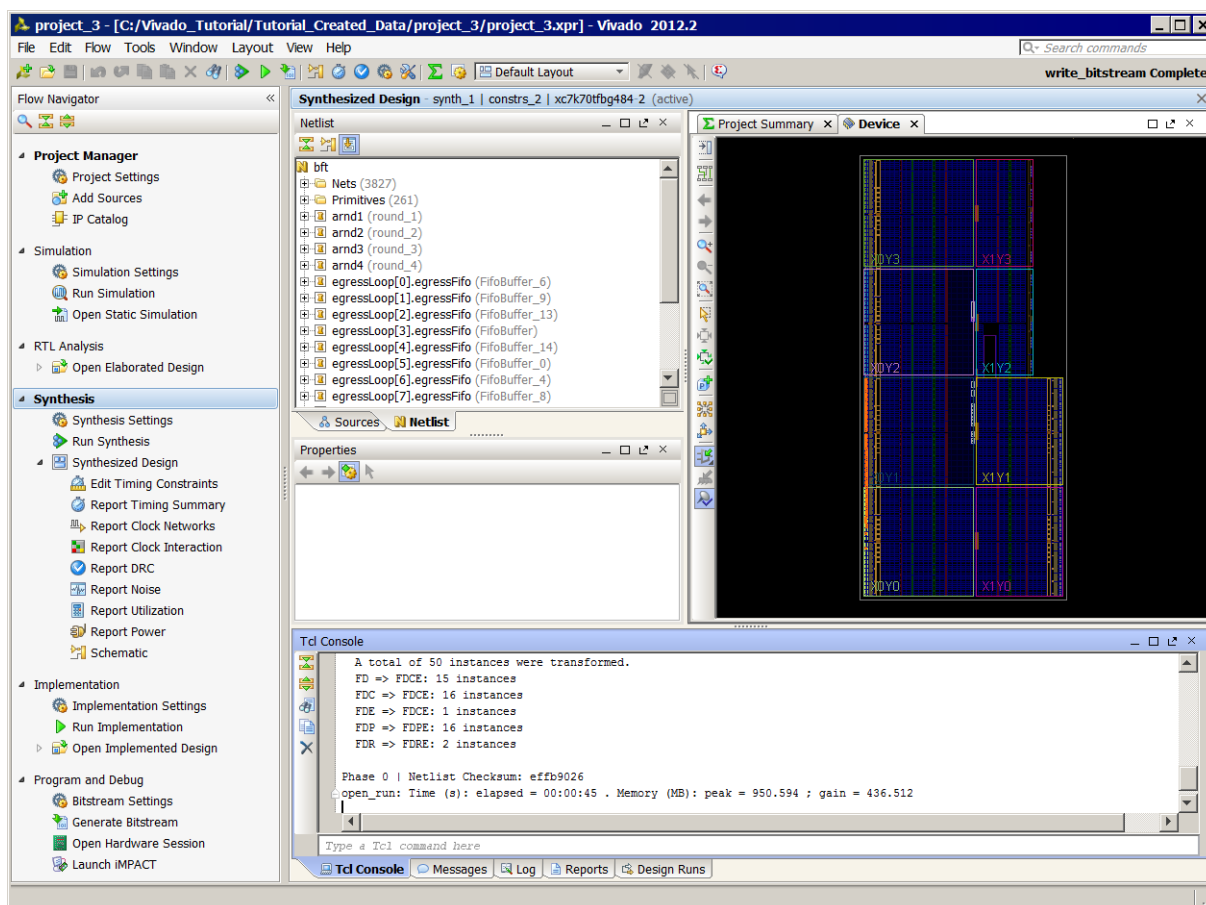


図 3-15 : 開いた合成済みデザイン

プロセスのバックグラウンドへの移動

Vivado Design Suite でデザイン ファイルおよび制約ファイルを読み込んで合成済みまたはインプリメント済みデザインを開くと、[Open Synthesized Design] ダイアログ ボックスまたは [Open Implemented Design] が開き (図 3-16)、プロセスをバックグラウンドで実行するよう指定できます。

プロセスをバックグラウンドで実行されるようにすると、バックグラウンド タスクを実行させたまま、レポートを表示したり、デザイン ファイルを開いたりといった別の操作を実行できます。この間、前の run を確認したり、レポートを表示したりして、時間を効率的に活用できます。ただし、[Tcl Console] はブロックされるので、Tcl コマンドを使用したり、開いている別のデザインに切り替えるなどの Tcl コマンドを必要とするタスクは実行できません。

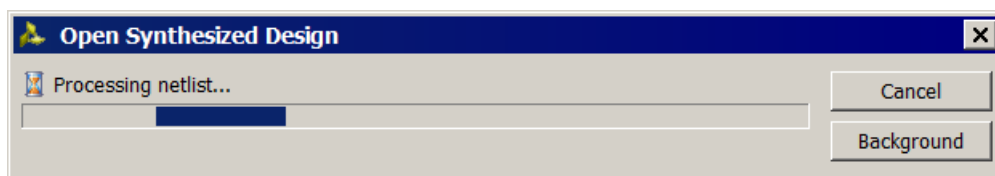


図 3-16 : [Open Synthesized Design] ダイアログ ボックス : バックグラウンド プロセス

レイアウトの使用

I/O 配置、フロアプラン、デバッグ コンフィギュレーションなどの特定のデザイン タスクを実行しやすいように、いくつかのレイアウトがデフォルトでサポートされています。レイアウトを変更すると、特定のデザイン タスクを実行するのに適したビューが表示されます。カスタム レイアウトも作成できます。

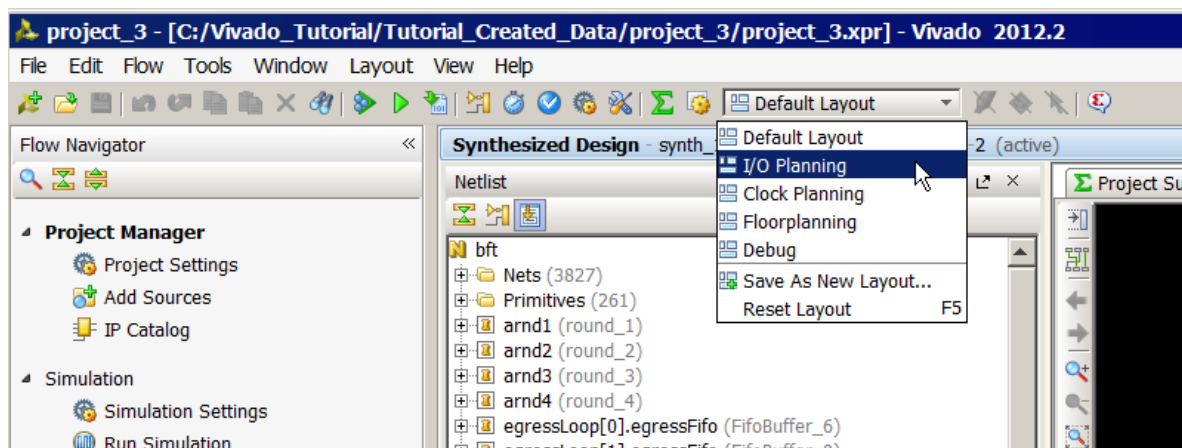


図 3-17 : レイアウトの選択

インプリメント済みデザインを開く

Vivado Design Suite でインプリメント済みデザインを開くと、インプリメント済みネットリストが開き、インプリメンテーション中に使用された物理制約およびタイミング制約が適用され、ターゲット パーツに対する配置配線結果が表示されます。インプリメント済みデザインのエレメントがメモリに読み込まれ、必要に応じてそれらを解析および変更して、デザインを完了させることができます。変更は、制約ファイル、ネットリスト、インプリメンテーション結果、およびデザイン コンフィギュレーションに保存できます。Vivado Design Suite には複数のインプリメンテーション run を設定でき、完了している run であればどれでも開くことができます。

インプリメント済みデザインを開くには、次のいずれかを実行します。

- Flow Navigator で [Implementation] → [Open Implemented Design] をクリックします。
- [Flow] → [Open Implemented Design] をクリックします。
- [Design Runs] ビューで run の名前をダブルクリックします。

インプリメント済みデザインに対してさまざまなデザイン タスクを実行できます。実際のタイミング、消費電力、使用率統計を算出することにより、デザインが要件を満たしているかどうかを判断できます。Vivado IDE のビューを使用して、デザインをさまざまな方法で解析できます。1 つのビューでオブジェクトを選択すると、ほかのビューでも選択されます。タイミング制約を変更し、タイミング解析を実行できます。フロアプランまたはデザイン コンフィギュレーション制約を適用し、今後の run で使用できるように保存できます。

[Device] ビューの [Routing Resources] をクリックして、配置または配線結果を確認できます。拡大表示していくと、より詳細な情報が表示されます。配置配線だけでなく、LUT 論理式、RAM の初期化などのデザイン コンフィギュレーションも変更できます。[Device] または [Schematic] ビューで結果を選択すると、RTL ファイルの問題の行にクロスプローブできます。[Schematic] ビューでは、ロジックの接続および階層をインタラクティブにさまざまな方法で調べることができます。詳細は、『Vivado Design Suite ユーザー ガイド: デザイン解析およびクロージャ テクニック』(UG906)を参照してください。

Flow Navigator のメニューはアクティブ run の状態によって異なるので、アクティブ run がインプリメントされていない場合は [Open Implemented Design] コマンドが淡色表示になっていることがあります。この場合、Flow Navigator で [Implementation] を右クリックして、完了したインプリメンテーション run のインプリメント済みデザインを開くことができます。

インプリメント済みデザインのデフォルト レイアウトが表示されます。図 3-18 に、インプリメント済みデザインのデフォルト レイアウトを表示します。[Device] ビューには、最後に閉じたときの状態により、配置のみが表示される場合と配線も表示される場合があります。[Device] ビューで [Routing Resources] をクリックし、配線のみを表示するか配線も表示するかを切り替えます。

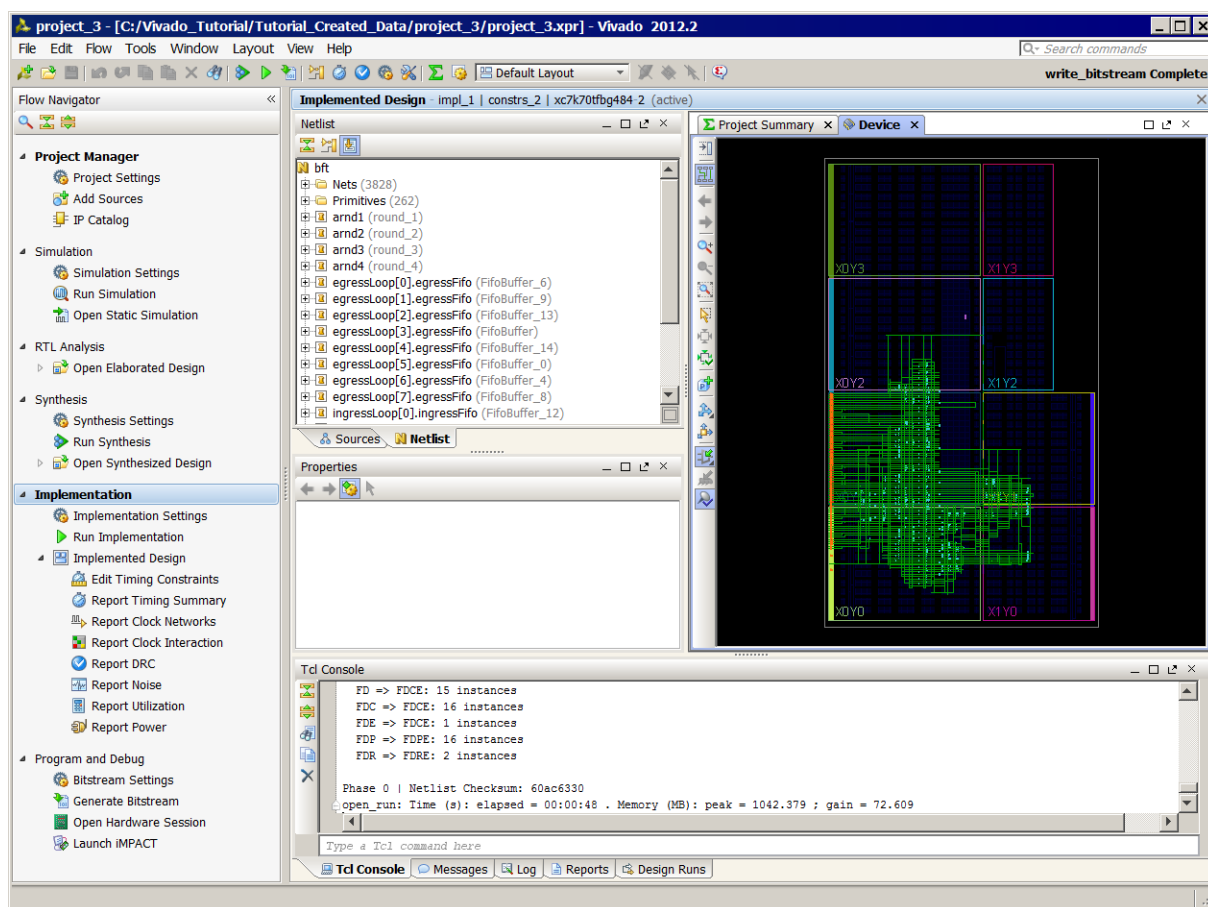


図 3-18: インプリメント済みデザイン

開いている複数のデザインの管理

Vivado Design Suite でデザインを開くと、デザインがメモリに読み込まれます。エラボレートされたデザイン、合成済みデザイン、インプリメント済みデザインを同時に複数開いて、異なるデザイン オプションおよび run の結果を表示できます。この場合、マシンのメモリ容量のみが制限となります。デザインを開くと、メモリ容量を確保するため、既に開いているデザインを閉じるかどうかを確認するメッセージが表示されます。開いているデザインは、アクティブなデザイン ビュー パナーの右端にある [X] をクリックして閉じることもできます。

アクティブになっているデザインは、Flow Navigator でハイライトされているメニューにより識別できます。これで、現在のデザインが視覚的にわかります。図 3-19 では、[Synthesis] メニューがハイライトされています。これは、現在のセッションで合成済みデザインが開いていてアクティブであることを示します。デザイン ビュー パナーでも開いているデザインを確認できます。

Flow Navigator の表示も変化して、複数のデザインが開いていることが示されます。たとえば、合成済みデザインを開くと [Open Synthesized Design] が [Synthesized Design] に変わります。

複数のデザインを開いている場合、Flow Navigator には開いているデザインに適したメニューが表示されます。図 3-19 では、[Implementation] メニューの下に [Implemented Design] コマンドが表示されているのでインプリメント済みデザインが開いていることがわかりますが、アクティブではありません。[Implemented Design] コマンドをクリックすると、インプリメント済みデザインがアクティブになり、ハイライトが [Synthesis] から [Implementation] に移動します。

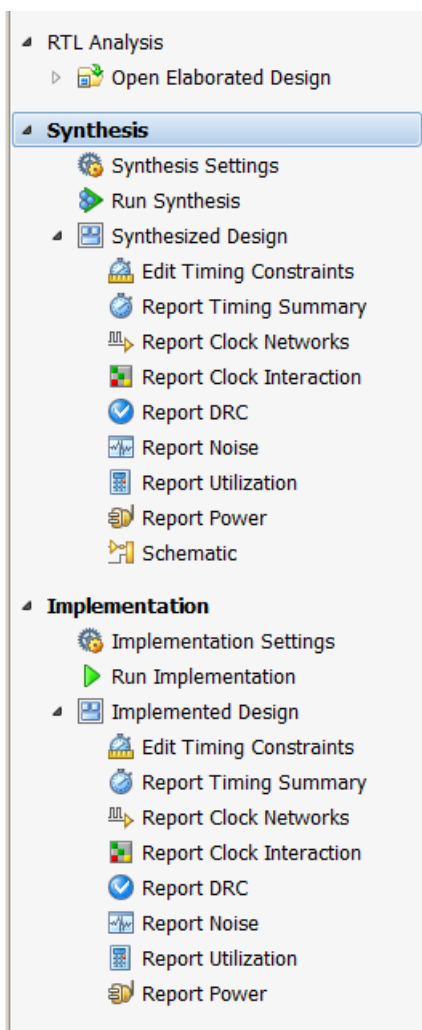


図 3-19 : 合成済みデザインとインプリメント済みデザインを両方開いている場合の Flow Navigator

Flow Navigator では、マウスの右ボタンをクリックして開くポップアップ メニューもサポートされています。Flow Navigator の [RTL Analysis]、[Synthesis]、[Implementation] を右クリックすると、ポップアップ メニューが表示されます。図 3-20 は、[Synthesis] を右クリックしたところを示しています。

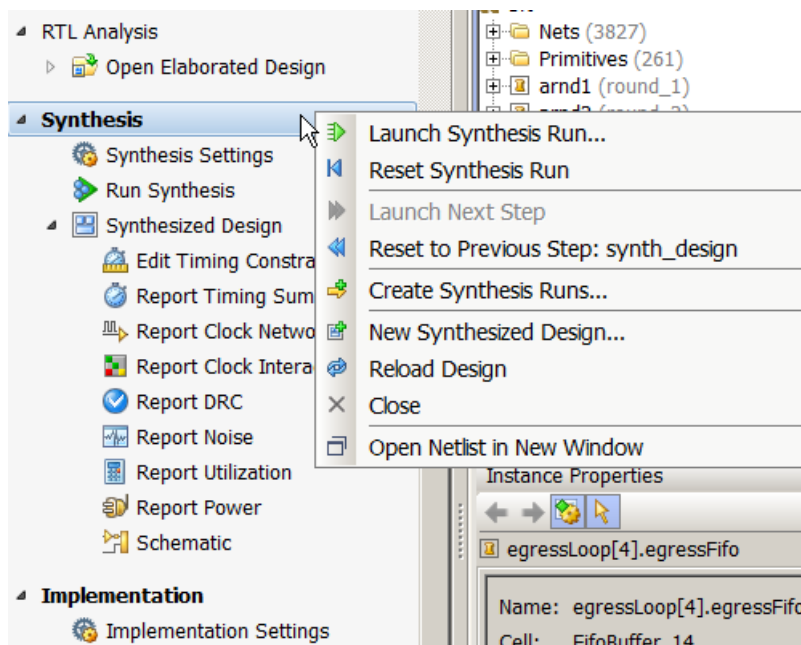


図 3-20 : Flow Navigator のポップアップ メニュー

[Open Netlist in New Window] をクリックすると、現在開いているデザインを現在のウィンドウに維持したまま、合成済みデザインを新しいウィンドウで開くことができます。このコマンドを使用すると、エラボーレートされたデザインを表示したまま、別のウィンドウで合成済みデザインまたはインプリメント済みデザインを開くことができます。どちらのウィンドウでも拡大/縮小、表示エリアを変更、デザイン エレメントを選択して、解析および開発が可能です。

デザイン ビュー バナーの使用

デザイン ビュー バナーには、現在のデザインの制約セット、ターゲット パーツ、合成またはインプリメンテーション run が表示されます。図 3-21 に、さまざまなデザイン タイプのデザイン ビュー バナーを示します。

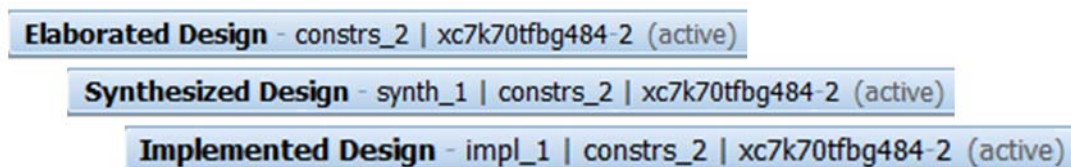


図 3-21: デザイン ビュー バナー

複数のデザインが開いている場合は、図 3-22 のようにデザイン ビュー バナーにタブが表示され、デザイン間の表示を切り替えることができます。

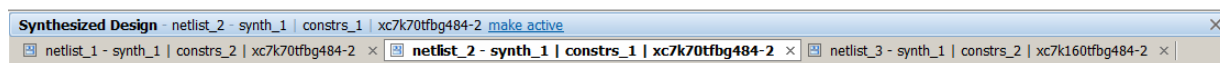


図 3-22 : 複数のデザインのタブ

最新でないデザインのアップデート

設計プロセスでは、通常ソース ファイルまたは制約の変更が必要です。これらのファイルの依存性は Vivado Design Suite で管理され、表示されているデザイン データが最新でなくなるとそれが示されます。ターゲット パーツやアクティブ制約セットなどのプロジェクト設定を変更すると、デザインが最新の状態でなくなります。

ソース ファイル、ネットリスト、またはインプリメンテーション結果がアップデートされると、開いている合成済みデザインまたはインプリメント済みデザインのデザイン ビュー パナーの右側に「Out-of-Date」と表示され、run が最新の状態ではないことが示されます (図 3-23)。デザインのどの部分が最新ではないのかを確認するには、[more info] リンクをクリックします。

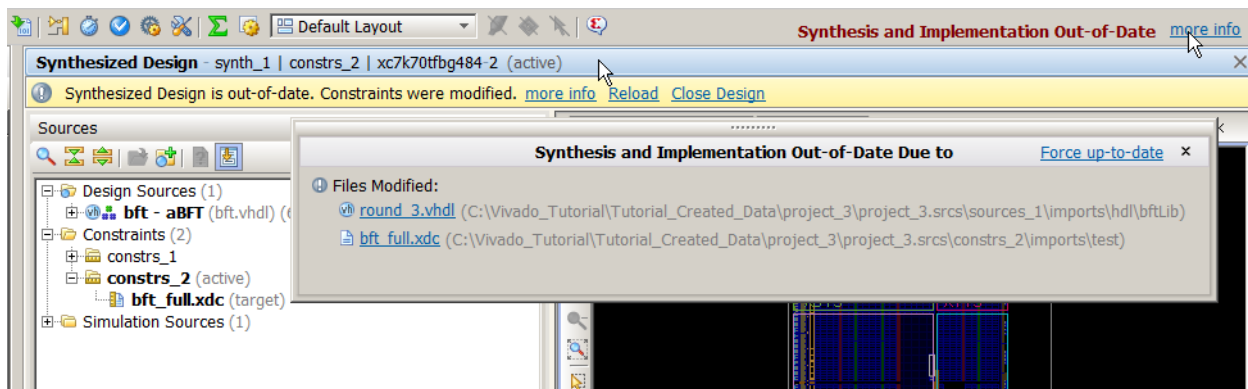


図 3-23 : デザインの再読み込みが必要であることを示すバナー

この状態を解決するには、次の 3 つの方法があります。

- 強制的に最新の状態にする : [more info] をクリックすると開く [Out-of-Date Due to] ウィンドウ (図 3-23) に表示される [Force up-to-date] リンクをクリックします。

アクティブな合成またはインプリメンテーション run の NEEDS_REFRESH プロパティがリセットされ、run が最新の状態であるとマークされます。このコマンドを Tcl で実行する場合は、次のようなコードになります。

```
set_property needs_refresh false [get_runs synth_2]
```

注記 : このコマンドは、デザインに少しだけ変更を加えており、デザインを更新したくない場合に使用します。

- RTL デザインを閉じる : 最新の状態でないデザインを閉じます。
- デザインを読み込み直す : [Reload] リンクをクリックして現在のデザインのメモリ内の表示を更新し、デザイン データへの変更を保存します。

デザインへの変更の保存

Vivado IDE では、メモリ内のデザインを変更します。保存が必要な変更には、制約の変更、ネットリストの変更、消費電力解析特性、ハードウェア コンフィギュレーション モード パラメーター、デバッグ コンフィギュレーションなどのデザイン パラメーターの変更などがあります。開いているデザインに加えた変更を保存するには、元の XDC 制約ファイルに保存するか、新しい制約セットに保存します。新しい制約セットには、変更を含むすべてのデザイン制約が含まれます。このようにすると、元の XDC ソース ファイルを変更せずに保持できます。Vivado IDE では、可能な限り元のファイル フォーマットが保持されます。追加した制約は、ファイルの最後に追加されます。既存の制約への変更は、元のファイルの場所に保持されます。

元の XDC 制約ファイルへの変更の保存

変更は元の XDC 制約ファイルに保存できます。Vivado IDE では、可能な限り元のファイル フォーマットが保持されます。追加した制約は、ファイルの最後に追加されます。既存の制約への変更は、元のファイルの場所に保持されます。

デザインに加えた変更を元の XDC 制約ファイルに保存するには、[File] → [Save Constraints] をクリックするか、またはツールバーの [Save Constraints] ボタンをクリックします。



開いているデザインに対して加えられた制約、デバッグ コアおよびコンフィギュレーション、デザイン コンフィギュレーション設定の変更がすべて保存されます。

注記：インプリメント済みデザインが開いている場合、制約の変更は現在アクティブな制約ファイルではなく、インプリメンテーションで使用されたインプリメンテーション run に関連する制約ファイルに保存されます。これにより、インプリメント済みデザインへの変更がインプリメンテーション run の適切な制約ファイルに保存されます。ただし、特定のインプリメンテーション run に使用した制約ファイルに意図していない変更が加えられる可能性があります。Vivado Design Suite では、この状況を警告するメッセージが表示され、ディスクに変更を保存する前に保存する制約ファイルを選択できます。

新しい XDC 制約ファイルへの変更の保存

デザインに加えた変更を新しい制約セットとしても保存できます。このようにすると元の制約ソース ファイルはそのまま保持し、加えた変更を保存できます。新しい制約セットには、変更を含むすべてのデザイン制約が含まれます。このようにすると、元の XDC ソース ファイルを変更せずに保持できます。新しい制約セットをアクティブにし、次の run の実行時やデザインを開いたときに適用されるようにできます。

元の制約ソース ファイルをそのまま保持して新しい制約ファイルを作成するには、[File] → [Save Constraints As] をクリックします。

デザインを閉じる

デザインを閉じると、メモリ内のデザイン数を減らし、ソースを編集できる場所が複数になるのを避けることができます。別のデザイン表示に移動する前に、デザインを閉じるようダイアログ ボックスが表示されることがあります。個々のデザインを閉じるには、メイン表示エリアのバナーの右端にある [X] をクリックします。

デザインの解析、検証、およびデバッグ

ロジック シミュレーション

Vivado Design Suite では、デザインまたは IP を検証するために複数のロジック シミュレーション オプションが提供されています。Vivado シミュレータは、完全に統合された混合モード シミュレータで、アナログ波形を表示する機能を備えています。同じシミュレーション環境をビヘイビア シミュレーション、構造シミュレーション、および Vivado ロジック解析でのハードウェア検証に使用できます。

注記: 2012.2 では、Vivado IDE からはビヘイビア シミュレーションのみを起動できます。構造シミュレーションおよびタイミング シミュレーションは、Tcl コンソールから起動できます。

開いているデザインの Verilog、VHDL ネットリスト、および SDF フォーマット ファイルを記述することにより、サードパーティ シミュレータも使用できます。ModelSim および Questa シミュレータを Vivado IDE から起動できます。

詳細は、『Vivado Design Suite ユーザー ガイド : ロジック シミュレーション』(UG900) および『Vivado Design Suite ユーザー ガイド : プログラムおよびデバッグ』(UG908) を参照してください。

タイミング解析

Vivado Design Suite では、Tcl API および SDC 制約オプションを使用した、さまざまなタイミング解析オプションが提供されています。多数の標準レポート Tcl コマンドがあり、クロック構造、ロジックの関係、デザインに適用されている制約などに関する情報を取得できます。詳細は、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) を参照するか、または Tcl コンソールのヘルプ機能を使用してください。

Vivado IDE では、タイミング解析結果をグラフィカルに設定および表示できます。タイミングをレポートするコマンドのさまざまなタイミング解析パラメーターを使用して試してみてください。クロック ネットワーク レポートおよびクロックの相互関係レポートでは、クロックのトポロジおよび関係を表示できます。スラック ヒストグラムには、デザインの全体的なタイミング パフォーマンスが表示されます。

詳細は、『Vivado Design Suite ユーザー ガイド : デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

DRC、消費電力、リソース使用率の解析

Vivado Design Suite では、Tcl API および Vivado IDE を使用した、さまざまなデザイン解析オプションが提供されています。Vivado IDE では、さまざまな方法でデザインおよびデザイン データを視覚的に確認できます。対話型環境、オブジェクトが異なるビューで相互選択される機能、およびロジック探索機能などにより、高度なデザイン解析機能が提供されています。

Vivado IDE では、消費電力、リソース使用率、DRC 解析結果をグラフィカルに設定および表示できます。デザインのどの段階でも、消費電力パラメーターを指定してすばやく消費電力を予測できます。さまざまなタイプのデバイスリソース使用率統計を解析できます。広範囲の DRC が用意されており、設定して実行できます。結果のレポートには、問題のオブジェクトへのリンクが含まれます。

詳細は、『Vivado Design Suite ユーザー ガイド : デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

多数の標準レポート Tcl コマンドがあり、デザインに関する情報を取得できます。詳細は、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) を参照するか、または Tcl コンソールのヘルプ機能を使用してください。

インプリメンテーション結果の解析

インプリメント済みデザインを開くと、配置配線のインプリメンテーション結果が表示されます。[Timing Results] ビューでタイミング パスを選択すると、[Device] ビューに配置および配線が表示されます。デザイン要件を達成するため、配置および配線を調整できます。LUT 論理式、RAM 初期化、PLL コンフィギュレーションなどのデザイン設定も実行できます。これらの変更は、インプリメント済みデザインに加えられているので、run をリセットすると変更は失われます。

詳細は、『Vivado Design Suite ユーザー ガイド : デザイン解析およびクロージャ テクニック』(UG906) を参照してください。

ハードウェアの検証およびデバッグ

Vivado IDE の Vivado ロジック解析環境には、デザインを検証およびデバッグするための多数の機能があります。RTL または合成済みネットリストに ILA (Integrated Logic Analyzer) や ICON (Integrated Controller) などのデバッグ コアをコンフィギュレーションおよびインプリメントできます。Vivado IDE で合成済みデザインを開くと、必要なプローブ信号を選択してコアにコンフィギュレーションできます。ビットストリーム ファイルが生成されているどの run に対しても、Vivado ロジック解析を起動してハードウェアを検証できます。

詳細は、『Vivado Design Suite ユーザー ガイド : プログラムおよびデバッグ』(UG908) を参照してください。

プロジェクト モードでの Tcl コマンドの使用

Tcl コマンドおよびスクリプトの使用は、モードによって異なり、作成される Tcl の run スクリプトもモードによって異なります。詳細は、「[プロジェクト モードと非プロジェクトモードの違い](#)」を参照してください。プロジェクトの作成、インプリメンテーション、レポート作成のための、プロジェクト モードの基本的な Tcl コマンドを次に示します。プロジェクトに関連するすべての Tcl コマンドの詳細は、『[Vivado Design Suite Tcl コマンド リファレンス ガイド](#)』(UG835) を参照してください。

デザイン タスクに関連する Tcl コマンドを理解するには、Vivado IDE で必要なコマンドを実行し、[Tcl Console] ビューまたは `vivado.jou` ファイルで構文を確認するのが最適な方法です。

注記：このガイドには、使用可能な Tcl コマンドすべての説明が含まれるわけではありません。詳細は、『[Vivado Design Suite Tcl コマンド リファレンス ガイド](#)』(UG835) および『[Vivado Design Suite ユーザー ガイド：Tcl スクリプト機能の使用](#)』(UG894) を参照してください。

プロジェクト モードでの基本 Tcl コマンド

表 3-2: プロジェクト モードでの基本 Tcl コマンド

Tcl コマンド	説明
<code>create_project</code>	Vivado プロジェクトを作成します。引数としてプロジェクト名、保存場所、デザインの最上位モジュール名、ターゲット パーツを指定します。
<code>add_files</code>	プロジェクトにソース ファイルを追加します。Verilog (.v)、VHDL (.vhd または .vhdl)、System Verilog (.sv)、IP (.xco または .xci)、XDC 制約 (.xdc または .sdc)、XPS からのエンベデッド プロセッサ サブシステム (.xmp)、System Generator モジュール (.mdl) などのソースを追加できます。個々のファイルを追加するか、ディレクトリ全体を指定して有効なファイルが自動的に追加されるようにすることもできます。
<code>set_property</code>	Vivado IDE でさまざまな目的で使用されます。プロジェクトでは、ソースの VHDL ライブラリ、シミュレーションのみのソース、ターゲット制約ファイル、ツール設定などを定義できます。
<code>import_files</code>	指定のファイルを現在のファイル セットにインポートし、プロジェクトに追加します。XDC ファイルを制約セットに定義するのにも使用されます。
<code>launch_runs</code> <code>launch_runs -to_step</code>	合成 (<code>synth_1</code>)、インプリメンテーション (<code>impl_1</code>)、またはビットストリーム生成 (<code>impl_1</code>) を開始します。このコマンドには、個々のインプリメンテーション コマンドと、 <code>run</code> の完了後に標準レポートを生成するコマンドが含まれます。 <code>run</code> のステータスを管理するのに使用され、すべてのインプリメンテーション コマンドを 1 つにまとめています。 <code>-to_step</code> オプションを使用すると、個々のインプリメンテーション コマンドおよびビットストリーム生成を個別に実行できます。
<code>wait_on_run</code>	<code>run</code> が完了するまで待ってからフローの次の段階を実行します。
<code>open_run</code>	合成済みデザインまたはインプリメント済みデザインをレポートまたは解析用に開きます。レポートや解析の Tcl を使用して情報を取得するには、デザインを開いておく必要があります。
<code>close_design</code>	メモリ内のデザインを閉じます。
<code>start_gui</code> <code>stop_gui</code>	メモリ内の現在のデザインで Vivado IDE を起動し、終了します。

プロジェクト モードでのその他の Tcl コマンド

表 3-3: プロジェクト モード専用の Tcl コマンド

add_files	current_project	get_ips	list_targets	reset_run
archive_project	current_run	get_projects	make_wrapper	reset_target
close_design	delete_fileset	get_runs	open_example_project	save_design
close_project	delete_run	import_as_run	open_io_design	save_design_as
convert_ip	find_top	import_files	open_project	save_project_as
create_fileset	generate_target	import_ip	open_run	set_speed_grade
create_project	get_boards	import_synplify	refresh_design	update_design
create_run	get_files	import_xise	reimport_files	update_files
create_violation	get_filesets	import_xst	remove_files	upgrade_ip
current_fileset	get_ipdefs	launch_runs	reorder_files	wait_on_run

プロジェクト モードの Tcl スクリプトの例

次は、Vivado Design Suite に含まれる BFT サンプル デザインのプロジェクト モードの Tcl スクリプトです。この例では、synth_design のような基本コマンドの多くが launch_runs コマンドの一部として実行されます。

```
#
# STEP#1:Create Project, add and configure sources and configure design
#
create_project project_bft ./project_bft -part xc7k70tfbg484-2
add_files -norecurse {./Sources/hdl/async_fifo.v ./Sources/hdl/bft.vhdl
./Sources/hdl/FifoBuffer.v}
add_files -norecurse ./Sources/hdl/bftLib
set_property library bftLib [get_files -of_objects sources_1 [glob
./Sources/hdl/bftLib/*.vhdl]]
import_files -force -norecurse
import_files -fileset constrs_1 -force -norecurse ./Sources/bft_full.xdc
set_property target_constrs_file
./project_bft/project_bft.srcs/constrs_1/imports/Sources/bft_full.xdc\
[cur-rent_fileset -constrset]
#
# STEP#2:Configure and launch Synthesis and Implementation and generate reports
#
set_property args.synth_design.flatten_hierarchy rebuilt [get_runs synth_1]
launch_runs synth_1 -jobs 4
wait_on_run synth_1
launch_runs impl_1 -jobs 4
wait_on_run impl_1
launch_runs impl_1 -to_step bitgen
wait_on_run impl_1
#
# STEP#3:Start IDE for design analysis
#
start_gui
stop_gui
```

非プロジェクト モードの使用

非プロジェクト モードでの Tcl コマンドの使用

非プロジェクト モードを使用する場合、デザインは `read_verilog`、`read_vhdl`、`read_edif`、`read_ip`、および `read_xdc` コマンドを使用してコンパイルされます。ソースはコンパイル用に順序付けられ、合成に渡されます。

注記：この文書には、使用可能な Tcl コマンドすべての説明が含まれるわけではありません。詳細は、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) および『Vivado Design Suite ユーザー ガイド：Tcl スクリプト機能の使用』(UG894) を参照してください。

非プロジェクト モードでの基本 Tcl コマンド

表 3-4: プロジェクト モードでの基本 Tcl コマンド

Tcl コマンド	説明
<code>read_verilog</code>	非プロジェクト セッション用に Verilog (.v) および System Verilog (.sv) ソース ファイルを読み込みます。
<code>read_vhdl</code>	非プロジェクト セッション用に VHDL (.vhd または .vhdl) ソース ファイルを読み込みます。
<code>read_ip</code>	非プロジェクト セッション用に既存の IP (.xco または .xci) プロジェクト ファイルを読み込みます。.xco IP プロジェクトでは .ngc ネットリストが使用されます。.xci IP では、コンパイルには RTL が使用され、存在する場合はネットリストが使用されます。
<code>read_xdc</code>	非プロジェクト セッション用に .sdc または .xdc フォーマットの制約ソース ファイルを読み込みます。
<code>set_param</code> <code>set_property</code>	デザイン コンフィギュレーション、ツール設定などを定義します。
<code>link_design</code>	セッションでネットリスト ソースを使用している場合に、デザインを合成用にコンパイルします。
<code>synth_design</code>	デザインの最上位モジュールとターゲット パーツを引数として Vivado 合成を実行します。
<code>opt_design</code>	合成済みネットリストに対して最適化を実行します。
<code>power_opt_design</code>	システム全体の消費電力を削減するためクロック ゲーティングを実行します (オプション)。
<code>place_design</code>	デザインを配置します。
<code>phys_opt_design</code>	タイミングまたは配線性を向上するため物理ロジック最適化を実行します (オプション)。
<code>route_design</code>	デザインを配線します。
<code>report_*</code>	デザイン プロセスのさまざまな段階でレポートを生成します。
<code>write_bitstream</code>	BIT ファイルを生成して DRC を実行します。
<code>write_checkpoint</code> <code>read_checkpoint</code>	フローの任意の段階でのデザインを保存します。デザイン チェックポイントには、ネットリスト、制約 (フローで実行された最適化を含む)、インプリメンテーション結果が含まれます。
<code>start_gui</code> <code>stop_gui</code>	メモリ内の現在のデザインで Vivado IDE を起動し、終了します。

非プロジェクト モードでのその他の Tcl コマンド

多くの Tcl コマンドは、レポート コマンドのようにどちらのモードでも使用できますが、プロジェクト モードまたは非プロジェクト モードのいずれかでしか使用できないものもあります。スクリプトを作成する際は、どちらかのモードでしか使用できないコマンドを混合しないように注意してください。たとえば、プロジェクト モードを使用する場合は、非プロジェクト モードでしか使用できない `synth_design` のようなコマンドを使用しないようにします。プロジェクト モードで非プロジェクト モード用のコマンドを使用すると、データベースでステータス情報がアップデイトされず、レポートが自動的に生成されなくなります。

表 3-5: 非プロジェクト モード特定の Tcl コマンド

<code>opt_design</code>	<code>power_opt_design</code>	<code>read_verilog</code>	<code>route_design</code>
<code>phys_opt_design</code>	<code>read_checkpoint</code>	<code>read_vhdl</code>	<code>write_checkpoint</code>
<code>place_design</code>	<code>read_edif</code>	<code>read_xdc</code>	

非プロジェクト モードの Tcl スクリプトの例

次は、Vivado Design Suite に含まれる BFT サンプル デザインの非プロジェクト モードの Tcl スクリプトです。デザイン チェックポイントを使用してフローのさまざまな段階でデータベースの状態を保存する方法と、さまざまなレポートを手動で生成する方法を示しています。

```
# create_bft_batch.tcl
# bft sample design
# A Vivado script that demonstrates a very simple RTL-to-bitstream batch flow
#
# NOTE:typical usage would be "vivado -mode tcl -source create_bft_batch.tcl"
#
# STEP#0: define output directory area.
#
set outputDir ./Tutorial_Created_Data/bft_output
file mkdir $outputDir
#
# STEP#1: setup design sources and constraints
#
read_vhdl -library bftLib [ glob ./Sources/hdl/bftLib/*.vhdl ]
read_vhdl ./Sources/hdl/bft.vhdl
read_verilog [ glob ./Sources/hdl/*.v ]
read_xdc ./Sources/bft_full.xdc
#
# STEP#2: run synthesis, report utilization and timing estimates, write checkpoint
design
#
synth_design -top bft -part xc7k70tfbg484-2 -flatten rebuilt
write_checkpoint -force $outputDir/post_synth
report_timing_summary -file $outputDir/post_synth_timing_summary.rpt
report_power -file $outputDir/post_synth_power.rpt
#
# STEP#3: run placement and logic optimization, report utilization and timing
estimates, write checkpoint design
#
opt_design
power_opt_design
place_design
phys_opt_design
write_checkpoint -force $outputDir/post_place
```



```
report_timing_summary -file $outputDir/post_place_timing_summary.rpt
#
# STEP#4: run router, report actual utilization and timing, write checkpoint design,
run drc, write verilog and xdc out
#
route_design
write_checkpoint -force $outputDir/post_route
report_timing_summary -file $outputDir/post_route_timing_summary.rpt
report_timing -sort_by group -max_paths 100 -path_type summary -file
$outputDir/post_route_timing.rpt
report_clock_utilization -file $outputDir/clock_util.rpt
report_utilization -file $outputDir/post_route_util.rpt
report_power -file $outputDir/post_route_power.rpt
report_drc -file $outputDir/post_imp_drc.rpt
write_verilog -force $outputDir/bft_impl_netlist.v
write_xdc -no_fixed_only -force $outputDir/bft_impl.xdc
#
# STEP#5: generate a bitstream
#
write_bitstream -force $outputDir/bft.bit
```

デザイン チェックポイントの使用

Vivado IDE では、データベースを保存および復元するためのメカニズムとして、デザイン チェックポイントを使用します。チェックポイントは、フローの特定の地点におけるデザインのスナップショットです。現在のネットリスト、制約、およびインプリメンテーション結果が、デザイン チェックポイントに保存されます。非プロジェクト モード デザイン フローでは、デザインのどの段階でもデザイン チェックポイントを保存できます。チェックポイント デザインに対しては、Tcl コマンドを使用してデザイン フローの残りの段階を実行できますが、新しいデザイン ソースで変更を加えることはできません。

Vivado Design Suite の Tcl フローには、非プロジェクト モードで保存/復元をする次のコマンドが含まれます。

- `write_checkpoint` (デザイン チェックポイントの書き出し) : デザイン フローの任意の段階でデザイン データベースのスナップショットを保存します。これにより、拡張子が `.dcp` のファイルが作成されます。
- `read_checkpoint` (デザイン チェックポイントの読み込み) : デザイン スナップショットを復元します。デザインがすべて現在のインプリメンテーション ステートで保存されるので、問題をデバッグする際に役立ちます。チェックポイント デザインに対しては、Tcl コマンドを使用してデザイン フローの残りの段階を実行できます。

これらのコマンドの詳細は、`-help` オプションを使用してコマンドを実行するか、『Vivado Design Suite Tcl コマンドリファレンス ガイド』(UG835) を参照してください。

非プロジェクト モードでの Vivado IDE の使用

アクティブ デザインで Vivado IDE を開く

Vivado Design Suite の非プロジェクト Tcl フローでも、デザイン プロセスの各段階で Vivado IDE を開くことができます。この機能を使用して、メモリ内のデザインに対して解析および操作を実行します。Vivado IDE を起動するには `start_gui` コマンド、終了するには `stop_gui` コマンドを使用します。Flow Navigator、プロジェクト サマリ、ソースへのアクセスおよび管理、`run` などのプロジェクト機能は非プロジェクト モードでは使用できませんが、解析および制約変更の機能の多くは、[Tools] メニューから使用できます。

ただし、メモリ内のアクティブ デザインに対して Vivado IDE で加えた変更は、自動的にダウンストリーム ツールに適用されません。保存機能はありません。その後の実行用に制約の変更を保存するには、[File] → [Export] → [Export Constraints] をクリックして、すべてを含む新しい XDC 制約ファイルを記述します。

非プロジェクト モードでは、メモリ内のアクティブ デザインで Vivado IDE を起動および終了するのに次のコマンドを使用します。

- `start_gui` : メモリ内のアクティブ デザインを Vivado IDE で開きます。
- `stop_gui` : Vivado IDE を終了し、Tcl プロンプトに戻ります。



重要 : Vivado IDE を閉じると Tcl シェルも閉じてしまい、メモリ内のデザインが消去されます。アクティブ デザインを保持した状態で Tcl シェルに戻るには、[Tcl Console] ビューで `stop_gui` コマンドを入力してください。

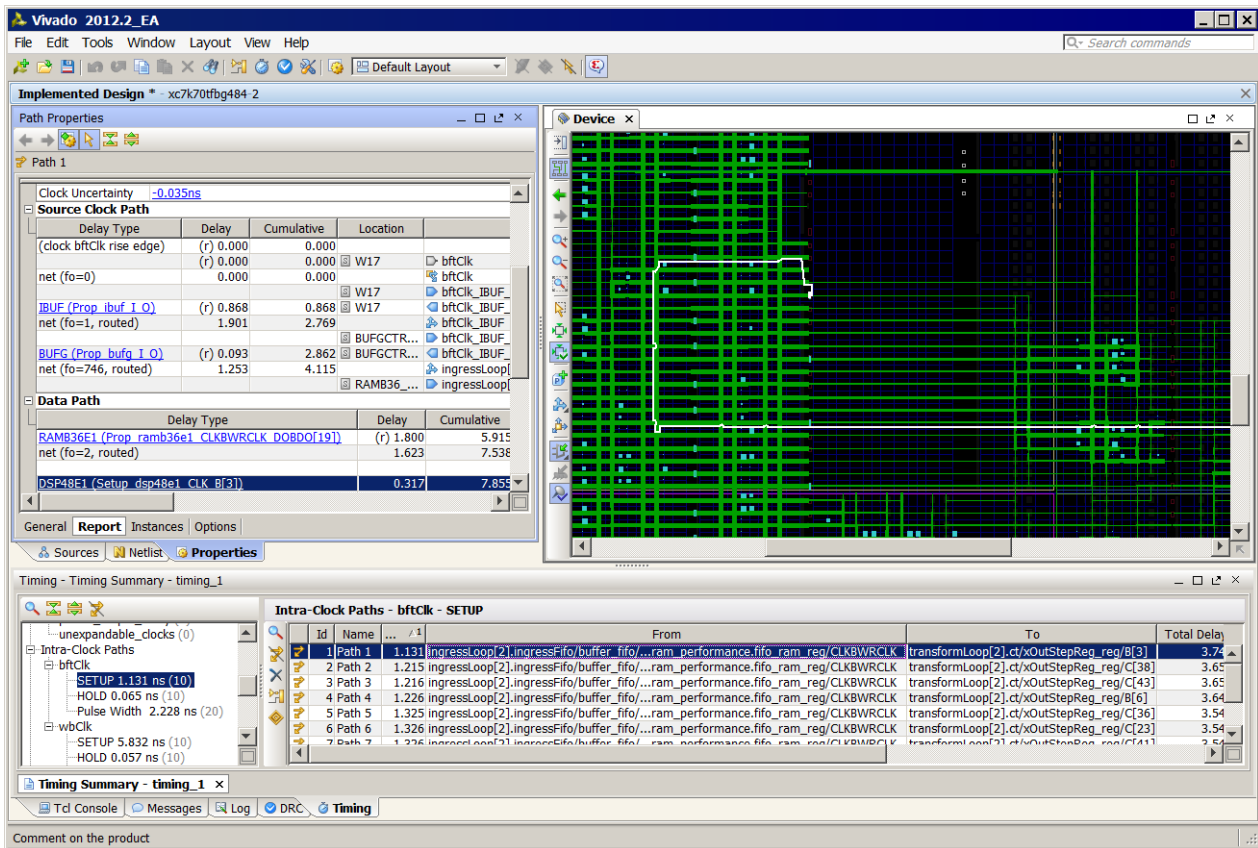


図 3-24 : アクティブ デザインで Vivado IDE を開く

アクティブ デザインへの変更の保存

メモリ内でデザインを変更するので、加えた変更は Vivado IDE Tcl セッションの残りのフローで使用されるダウンス トリーム ツールに自動的に渡されます。これは、ISE Design Suite ツールに慣れたユーザーには新しい概念です。これにより現在の実行に非常に有益であり、今後の実行用に変更を保存できます。

今後の実行用に制約を保存するには、[File] → [Export] → [Export Constraints] をクリックします。新しい制約ファイルを作成するか、元のファイルに上書きできます。

Vivado IDE でデザイン チェックポイントを開く

デザイン チェックポイントとして保存されたデザインも Vivado IDE で解析できます。Tcl コマンド (synth_design、opt_design、power_opt_design、place_design、phys_opt_design、route_design) を使用して非プロジェクト モードでデザインを実行し、任意の段階でデザインを保存して、後の Vivado IDE セッションで読み込むことができます。配線済みデザインから開始し、タイミングを解析し、タイミング問題を回避できるように配置を調整し、デザインが完全に配線されていない状態であっても、後で使用できるように保存できます。

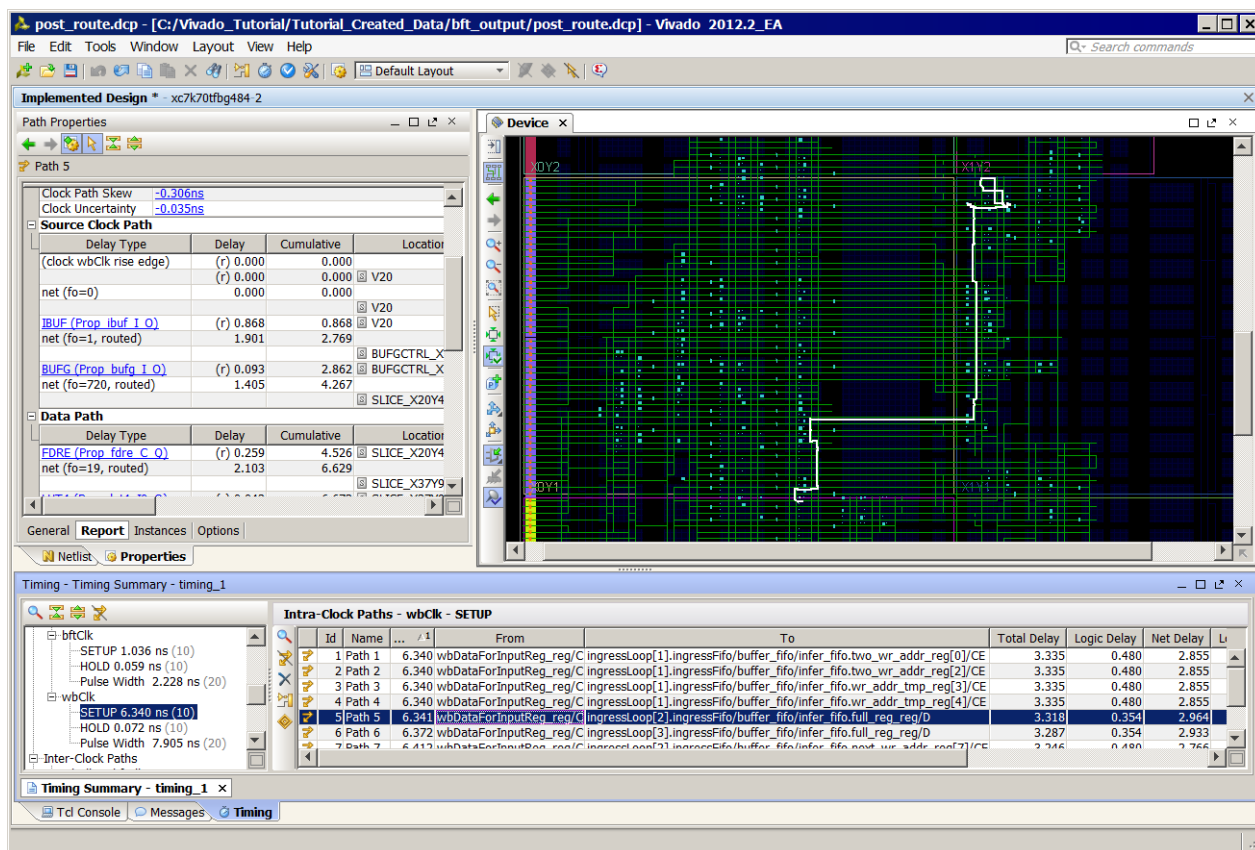


図 3-25 : Vivado IDE でデザイン チェックポイントを開く

Vivado IDE のビュー パナーに、開いているデザイン チェックポイントの名前が表示されます。

デザイン チェックポイントへの変更の保存

デザイン チェックポイントのデザインを開き、解析し、保存できます。新しいデザイン チェックポイントに変更を保存することもできます。

- [File] → [Save Checkpoint] をクリックし、現在のデザイン チェックポイントに加えた変更を保存します。
- [File] → [Write Checkpoint] をクリックし、チェックポイントの現在の状態を新しいデザイン チェックポイントとして保存します。

制約の変更は、新しい制約ファイルにエクスポートできます。[File] → [Export] → [Export Constraints] をクリックして、新しい制約ファイルを作成するか、元のファイルに上書きします。詳細は、『Vivado Design Suite ユーザー ガイド : インプリメンテーション』(UG904) を参照してください。

これらのコマンドの詳細は、-help オプションを使用してコマンドを実行するか、『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835) を参照してください。

サードパーティの合成済みネットリストの使用

Vivado Design Suite では、サードパーティの合成ツールまたは Xilinx Synthesis Technology (XST) を使用する場合など、合成済みネットリストのインプリメンテーションがサポートされます。外部合成ツールでは、Verilog または EDIF ネットリストと制約ファイルが生成されます。これらのネットリストは、プロジェクト モードまたは非プロジェクト モードのいずれかで、スタンドアロンで使用するか、RTL ファイルと混合して使用できます。

非プロジェクト モードのインプリメンテーション

ネットリストのみの非プロジェクト モードの場合、ネットリスト フローは次の点が異なります。

- `read_edif` では EDIF または NGC ネットリスト ファイルが読み込まれます。ネットリストを使用する場合、ファイルのベース名が最上位デザインの名前と同じである必要があります。
- `read_verilog` では構造型 Verilog ファイルが読み込まれます。
- `read_xdc` では、制約ファイルが読み込まれます。
- 最上位モジュールとパーツは、インプリメンテーション用にデザインを開くときに指定します。
`link_design -top my_module -part xc7vx485tffg1157-1`
- RTL/ネットリスト混合デザインの場合、`synth_design` コマンドでネットリストが自動的にリンクされるので、`link_design` コマンドを使用する必要はありません。

次は、1 つの EDIF の非プロジェクト モード スクリプトの単純な例です。

```
read_edif my_top.edf
read_xdc my_top.xdc
link_design -top my_top -part xc7vx485tffg1157-1
opt_design
# power_opt_design    ;# Optional
place_design
# phys_opt_design     ;# Optional
route_design
report_timing_summary
report_drc
write_checkpoint routed
write_bitstream
```

その他のリソース

ザイリンクス リソース

アンサー、資料、ダウンロード、フォーラムなどのサポート リソースは、次のザイリンクス サポート サイトを参照してください。

<http://japan.xilinx.com/support>

ザイリンクス資料で使用する用語集については、次を参照してください。

<http://japan.xilinx.com/company/terms.htm>

ソリューション センター

デバイス、ツール、IP のサポートについては、[ザイリンクス ソリューション センター](#)を参照してください。トピックには、デザイン アシスタンス、アドバイザリ、トラブルシュート ヒントなどが含まれます。

リファレンス

- Vivado™ Design Suite 2012.2 資料ページ
http://japan.xilinx.com/support/documentation/dt_vivado_vivado2012-2.htm