

Vivado Design Suite ユーザー ガイド

IP を使用した設計

UG896 (v2013.1) 2013 年 3 月 20 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 2012-2013 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 (v2013.1) を翻訳したもので、内容に相違が生じる場合には原文を優先します。

資料によっては英語版の更新に対応していないものがあります。

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

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
2012/07/25	2012.2	初版
2012/12/18	2012.4	2012.4 Vivado IDE の GUI に合わせてテキストおよびグラフィックをアップデート
2012/03/20	2013.1	IP インテグレーターという早期リリースの機能の使用に関する新しい章を追加。2013.1 リリースに含まれる新機能に合わせてテキストおよび画像をアップデート

目次

改訂履歴	2
第 1 章：IP 中心のデザイン フローの概要	
Vivado IP カタログ	6
Vivado IP パッケージャー	6
[Manage IP] フロー	7
フローの開始	7
IP のネットリストのシミュレーションおよび生成	10
第 2 章：IP の操作	
IP へのアクセス	13
生成前の IP の追加	13
既存 IP の追加	14
IP ステータスのレポート	15
IP カタログからの IP インスタンスの作成	15
IP のカスタマイズ	17
IP 出力ファイルの生成	19
IP インスタンスの再カスタマイズ	20
IP インスタンスの出力ファイルのリセット	20
IP のプロジェクト設定	21
IP のインスタンシエート	22
IP の合成	22
IP のシミュレーション	25
RTL デザイン内での IP のシミュレーション	25
その他のシミュレータを使用したシミュレーション	25
デザイン内での IP の制約	26
IP のサンプル デザインの使用	26
有償ライセンス IP の使用	27
既存 IP の最新バージョンへのアップグレード	29
IP 操作の Tcl コマンド	30
第 3 章：IP パッケージの基礎	
IP パッケージャーとその使用フロー	32
IP パッケージ化フロー	33
IP ユーザーのフロー	33
リポジトリの管理	34
IP カタログ	35
IP のカスタマイズと生成	35
IP パッケージの検証	36
IP パッケージャーの入力	36
入力ファイルのグループ	36
パッケージ化された IP に最低限必要なファイル セット	36
IP パッケージャーの出力	36
出力パッケージ内の IP デザイン ファイルの分類	36

出力パッケージに含まれるその他のファイル	37
IP パッケージ化の手順	37
既存の Vivado プロジェクトのパッケージ化または新規プロジェクトの作成	37
プロジェクトを IP としてパッケージ化	38
IP カタログへの新規 IP の追加	40
HDL 以外のファイルの IP パッケージへの追加	42
以前の EDK IP のインテグレーター コアへの変換	44
推奨される変換フロー	44
予約済みパラメーター	45
第 4 章：IP サブシステムの設計	
概要	46
プロジェクトの作成	46
IP インテグレーターを使用した設計	48
IP インテグレーターの図のサイズ変更	48
背景色の変更	48
カーソル動作による表示の変更	48
デザイン キャンバスへの IP モジュールの追加	50
接続	51
IP インテグレーターのブロック オートメーションとコネクション オートメーション機能	54
IP ブロックの並び替え	59
コピーと貼り付け	59
階層の作成	59
メモリ マップの作成	60
デザイン ルール チェックの実行	61
ブロック図の最上位デザインへの統合	62
付録 A：IP の特性化と Fmax マージン システムの手法	
概要	64
スタンドアロン IP の特性化手法	64
Fmax マージン システムの手法	65
ツール オプションとその他の要因	67
付録 B：その他のリソース	

IP 中心のデザイン フローの概要

Vivado™ Integrated Design Environment (IDE) には、さまざまなデザイン ソースからデザインに IP モジュールを追加できるような IP 中心のデザイン フローがあります。

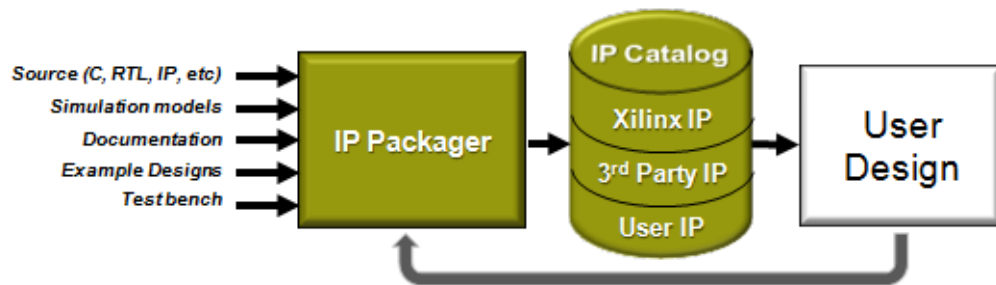


図 1-1 : IP 中心のデザイン フロー

図 1-1 に示すように、Vivado IDE には次の IP ソースを集めた IP カタログというセントラルリポジトリが含まれます。

- Vivado Design Suite IP
- System Generator for DSP デザインからのモジュール (MATLAB/Simulink アルゴリズム) および Vivado 高位合成デザイン (C/C++ アルゴリズム)
- サードパーティ IP

サードパーティから合成済み EDIF ネットリストとして提供されている IP もあります。[Add Sources] コマンドを使用すると、Vivado IDE からこれらのファイルをデザインに読み込むことができます。

注記 : 本書には、2013.1 リリースで早期アクセス機能としてライセンス付与される新しい Vivado IP インテグレーター の環境に関する情報が含まれます。ライセンスの取得については、フィールド アプリケーション エンジニア (FAE) にご連絡ください。

Vivado IP カタログ

Vivado IP カタログは、ザイリンクス IP、サードパーティ IP および企業独自の IP のセントラル リポジトリで、デザイン チーム、部署、企業内で共有して利用できます。Vivado IP カタログの主な機能は、次のとおりです。

- 開発されているエンド アプリケーションにかかわらず、構築ブロック、ウィザード、コネクティビティ、DSP、エンベデッド、AXI 基盤構造、およびビデオ IP を含むザイリンクス IP すべてに、1 つの共有リポジトリから簡単にアクセス可能
- 共有ネットワークドライブも含めた複数の物理ロケーションをサポートし、サードパーティまたは企業内で開発された IP を一貫した IP 環境で利用可能
- Vivado 統合設計環境 (IDE) または Tcl による自動スクリプト ベースのフローを使用して、IP をカスタマイズおよび生成
- インスタンス化テンプレート、シミュレーション モデル (HDL、C、または MATLAB)、および HDL サンプル デザインなど、オプションの IP 出力をオンデマンド配信
- IP を Vivado プロジェクトのインスタンス化 ソースとして直接評価可能な IP サンプル デザインを統合
- IP をデザインと共にグローバルに RTL 合成、合成可能な RTL または IP のビヘイビア シミュレーション モデルを使用可能
- 変更ログの記録と同じバージョン履歴の詳細に素早くアクセス可能。IP バージョン番号の命名規則は、メジャー番号.マイナー番号で表記されるよう統一

Vivado IP パッケージャー

図 1-1 に示すように、Vivado IDE には IP パッケージャーという独自のデザイン機能があります。IP パッケージャーは IP-XACT 規格 (IEEE-1685) に準拠しています。Vivado IDE ユーザー デザインが集められると、IP パッケージャーでデザインを再利用可能な IP モジュールに素早く変更して、Vivado IP カタログに追加して、ほかのユーザーが使用できるようにできます。

[Manage IP] フロー

Vivado IDE では、プロジェクトすべてのオーバーヘッドなしで、IP カタログで IP を検索したり、IP をカスタマイズしたり、コンフィギュレーションされた IP のリポジトリを管理したりといった使用しやすいフローが提供されています。チームで働く場合、またはプロジェクト構造外で作成した IP が多くある場合は、プロジェクト構造外で IP を作成および管理することをお勧めします。IP をプロジェクトに追加する場合、通常は前にカスタマイズした IP をこのリポジトリから参照します。ローカルのプロジェクトにはコピーしません。この方法を使用すると、リビジョン制御がシンプルになり、IP カスタマイズをほかの人と簡単に共有できるようになります。この方法は、IP を非プロジェクトのスクリプト ベースのフローで処理する場合に推奨されます。

フローの開始

[Manage IP] フローを開始するには、Vivado IDE を起動して、Getting Started ページで [Manage IP] をクリックします (図 1-2)。

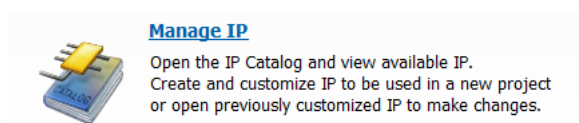


図 1-2 : [Manage IP] フローの起動

[Manage IP] をクリックすると、次のオプションを含むダイアログ ボックスが表示されます。

- [Open the IP Catalog] : IP を検索してカスタマイズできる IP カタログを起動します。
- [Open Previously Created IP] : 再カスタマイズまたは出力ファイルを生成するために、既存 IP ファイル (.xci) を開きます。
- [Recent Customized IP Locations] : 開くことのできるすべてのまたは選択した IP のある最近使用したディレクトリをリストします。

[Open the IP Catalog] を選択すると、[Manage IP Initial Settings] ダイアログ ボックス (図 2) が開き、パーツ、ターゲット言語、ターゲット シミュレーター、作成/カスタマイズした IP を保存するディレクトリを指定できます。

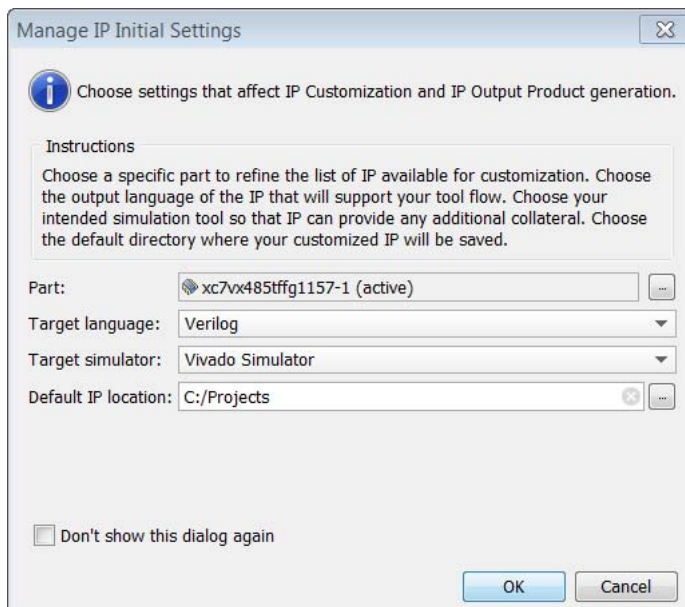


図 1-3 : セッションでの IP のデフォルト設定

情報を入力して [OK] をクリックしたら、図 1-4 のような IP カタログが表示されます。この段階で IP を選択してカスタマイズできます。カスタマイズすると IP ごとにディレクトリが作成されます。

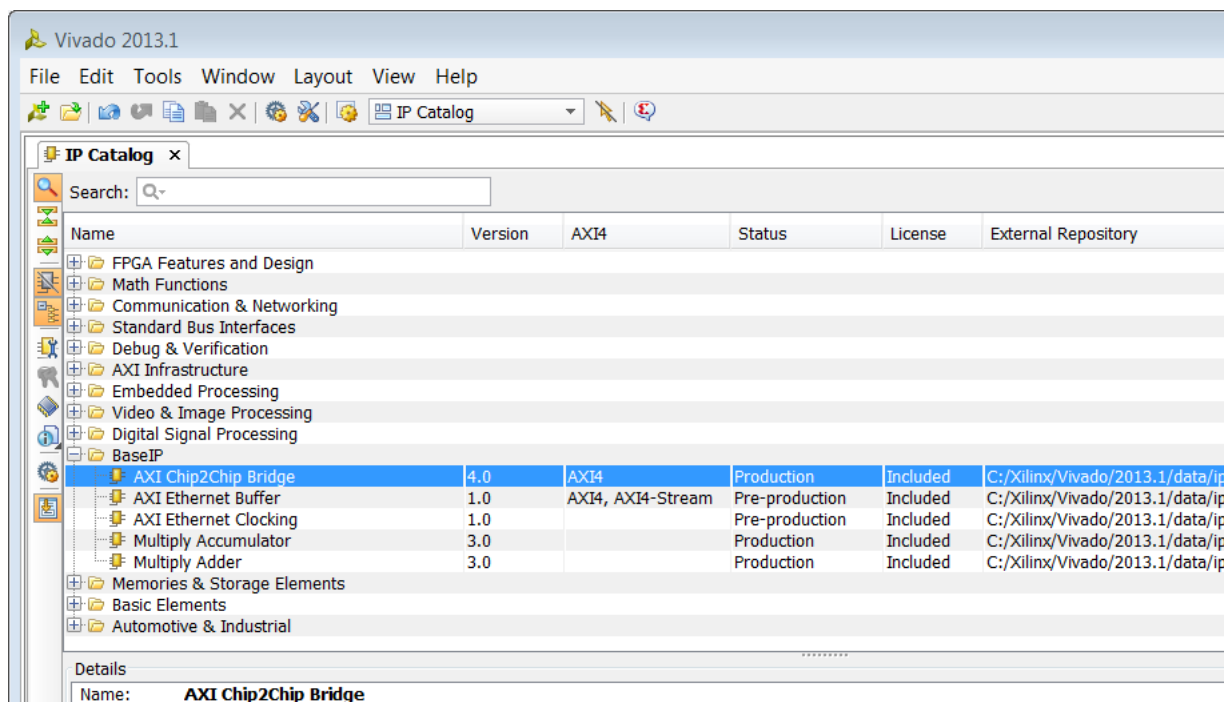


図 1-4 : IP カタログの [Manage IP] ビュー

IP 製品ガイド、変更ログ、製品ウェブ ページ、アンサー レコードなどへのアクセスを含めたすべてのカタログを開くことができます。グローバル IP 設定は、[Tools] → [Project Settings] をクリックするか、IP カタログで右クリックすると指定できます (図 1-5)。

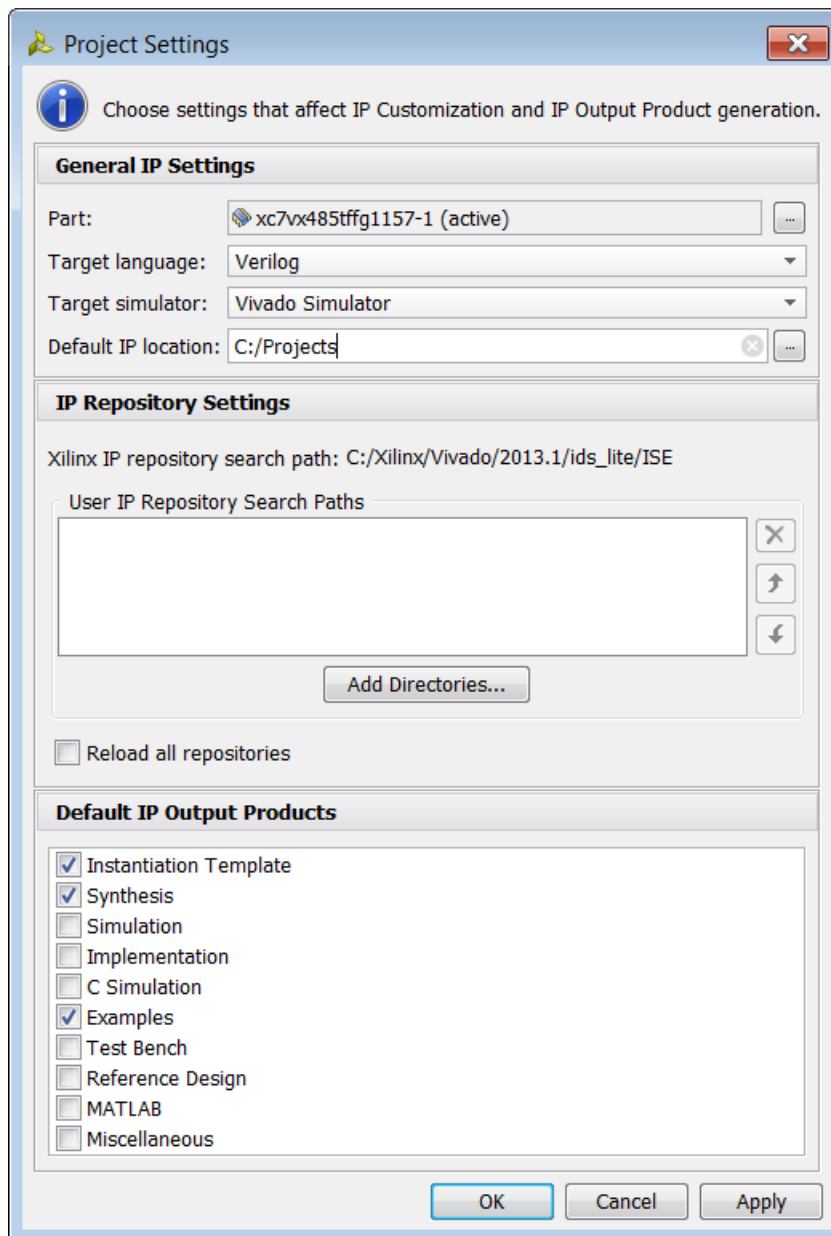


図 1-5 : [Manage IP] フローの IP 設定

これらの設定からは、IP をカスタマイズしたときに生成されるデフォルトの出力ファイルやリポジトリに対するディレクトリなどを選択できます。パーツ、ターゲット言語、ターゲット シミュレータ、ディレクトリもここから変更できます。

このフローを使用すると、複数の IP をカスタマイズおよび管理できます。IP の出力ファイルはデフォルトの IP ディレクトリに作成されたフォルダーに保存されます。ディレクトリには、XCI ファイルとその他の生成されたファイルが含まれます。既存 IP をプロジェクトに追加する際に参照されるのはこのディレクトリです。詳細は、「[IP へのアクセス](#)」

セス」を参照してください。開いたりカスタマイズした IP はすべて Vivado IDE の [Sources] ビューから表示されます (図 1-6)。

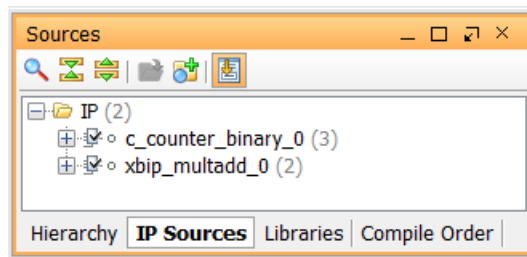


図 1-6: カスタマイズされた複数 IP の [Sources] ビュー表示

前述のとおり、カスタマイズされた IP は指定したディレクトリの別々のフォルダーにそれぞれ作成されます。図 1-7 は、IP ディレクトリが C:\Projects に設定された場合に作成された IP である xbip_multadd_0 と c_counter_binary_0 を示しています。

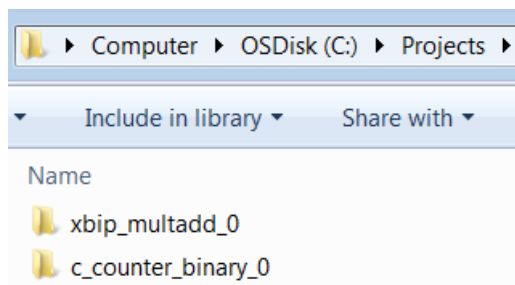


図 1-7: カスタマイズされた複数 IP の [Sources] ビュー表示

IP のネットリストのシミュレーションおよび生成

[Manage IP] フローには、Vivado IDE プロジェクトが含まれないので、シミュレーションまたは合成/インプリメンテーション run は含まれません。シミュレーションを実行するか、IP のネットリストを生成する場合は、まずプロジェクトを作成する必要があります。

これには、[File] → [Save Project As] をクリックし、プロジェクト名を指定します (図 1-8)。

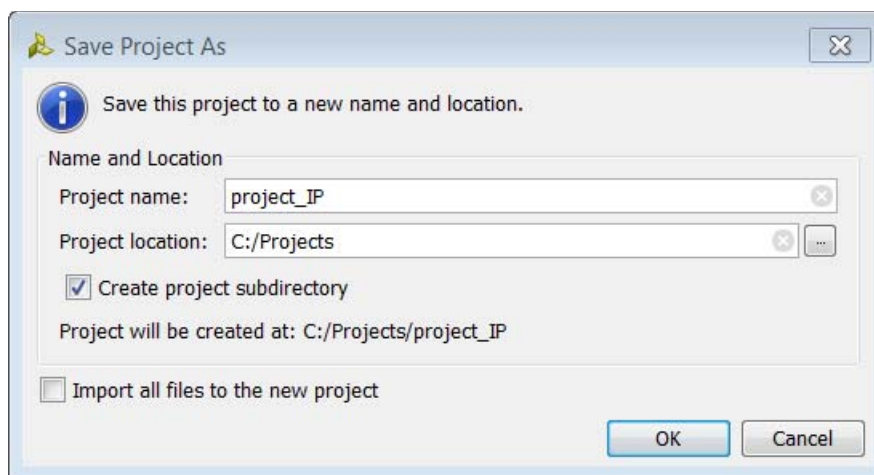


図 1-8: [Manage IP] セッションをプロジェクトに保存

RTL プロジェクトが作成され、[Sources] ビューにある IP が参照されます。デフォルトでは、IP はデフォルトの IP ディレクトリから参照されます。

[Set as Top] に IP を選択したら、「[IP のシミュレーション](#)」および「[IP の合成](#)」に説明するように IP をシミュレーションおよび合成できます。

IP の操作

RTL ベースのプロジェクトの場合、Vivado 環境外から既存の IP を追加したり、IP カタログから IP コア インスタンスを生成およびプロジェクトへ追加できます。IP ソースを追加するには、次の操作を実行します。

- 「IP へのアクセス」
- 「IP のカスタマイズ」
- 「IP 出力ファイルの生成」
- 「IP インスタンスの再カスタマイズ」
- 「IP インスタンスの出力ファイルのリセット」
- 「IP のインスタンシエート」
- 「IP の合成」
- 「IP のシミュレーション」
- 「デザイン内での IP の制約」
- 「IP のサンプル デザインの使用」
- 「有償ライセンス IP の使用」
- 「既存 IP の最新バージョンへのアップグレード」
- 「IP 操作の Tcl コマンド」

これらの操作について、次のセクションで説明します。

IP へのアクセス

生成前の IP の追加

生成前の CORE Generator™ (<core_name>.xco インスタンス ファイル) または生成前の Vivado™ IP (<core_name>.xci インスタンス ファイル) を追加するには、Add Sources ウィザードで [Add Existing IP] をオンにします (図 2-1)。生成前の IP を指定するか、生成されたソース ファイルをプロジェクトに追加するかを選択できます。

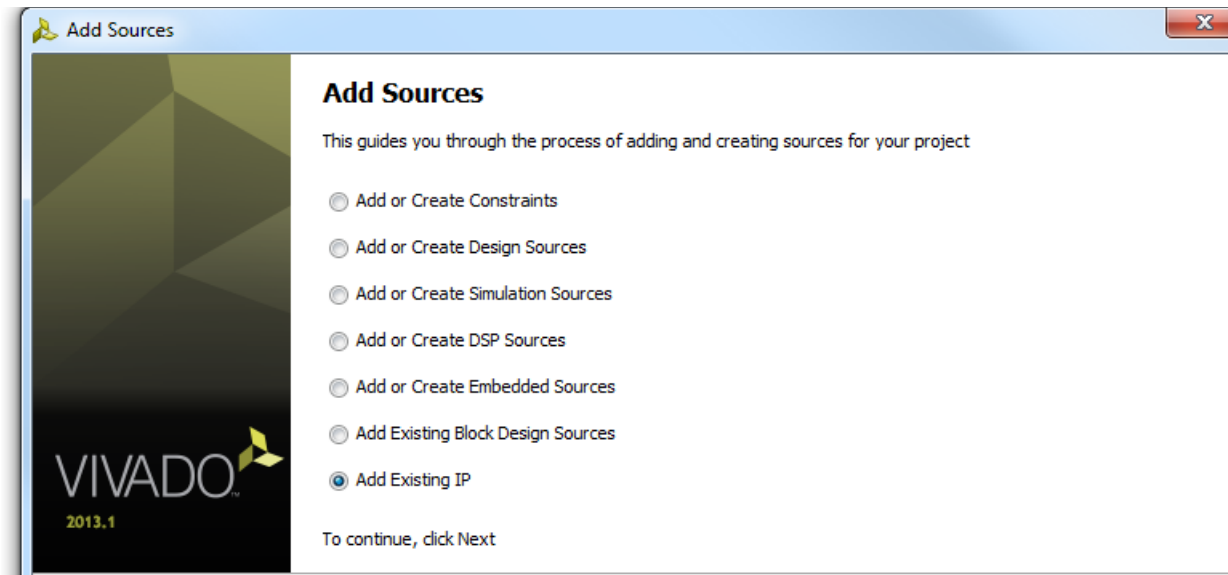


図 2-1 : 生成前の IP ソースの追加

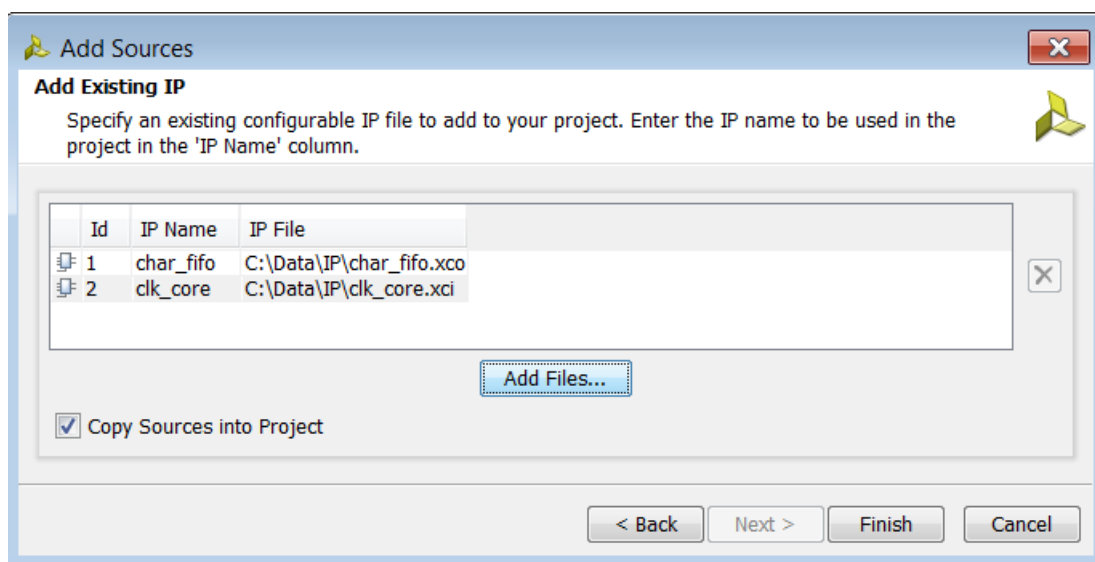


図 2-2 : Add Sources ウィザード : [Add Existing IP] ページ

既存の IP の追加は、次の例のように `import_ip Tcl` コマンドを使用しても実行できます。

```
import_ip -file C:/coregen_ip/aurora_8b10b_v7_1.xco -name aurora_8b10b_v7_1
import_ip -file C:/coregen_ip/blk_mem_gen_v6_1.xco -name blk_mem_gen_v6_1
```





注記：IP にリモートからアクセスする場合は、`read_ip` コマンドを使用する必要があります。これにより、IP はプロジェクトにはコピーされなくなります。

追加された IP コアは、[Hierarchy]、[Libraries]、[Compile Order] タブのその他のソース ファイルと一緒に、[Sources] ビューの [IP Sources] タブにそれぞれ表示されます。[Sources] ビューにはコアを構成するファイルが表示され、コアを選択すると [Source File Properties] ビューにプロパティが表示されます。

注記：IP コアの EDIF、Verilog、SystemVerilog ネットリストまたは NGC ファイルは、RTL またはネットリスト ベースのプロジェクトに追加することもできます。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』の第 2 章の「合成後プロジェクトの作成」セクションを参照してください。

既存 IP の追加

既存 IP (XCI または XCO 形式のいずれか) を追加すると、IP が Vivado IDE に読み込まれ、出力ファイル (NGC、RTL) が次のいずれかの状態で含まれます。

- 現バージョンの IP がカタログに含まれる場合 
- 現バージョンの IP がカタログに含まれ、合成ターゲットが生成されている場合 
- IP はロックされ、ターゲットが含まれている場合。IP は使用できますが、編集はできず、出力ファイルは新規に作成できません。たとえば、シミュレーション ターゲットは存在しない場合は作成できません。 
- IP はロックされ、ターゲットが含まれていない場合。 

IP は次の場合使用できません。

- IP は IP カタログに含まれる現バージョンにアップグレード可能な場合
- アップグレード オプションがない場合。この場合、IP を作成し直す必要があります。
- アップグレード オプションがなく、IP がカタログに表示されない場合。この場合、生成したターゲットを読み込まないと、IP は使用できません。

注記：Vivado IP カタログからアクセスできないバージョンの IP コアをインポートした場合、IP コアの再カスタマイズ、リセット、再生成は実行できません。

IP ステータスのレポート

プロジェクトに含まれる IP すべてのステータスは、[Tools] → [Report] → [Report IP Status] をクリックするとレポートできます (図 2-3)。レポートの名前は指定でき、オプションでテキスト ファイルを出力させることもできます。

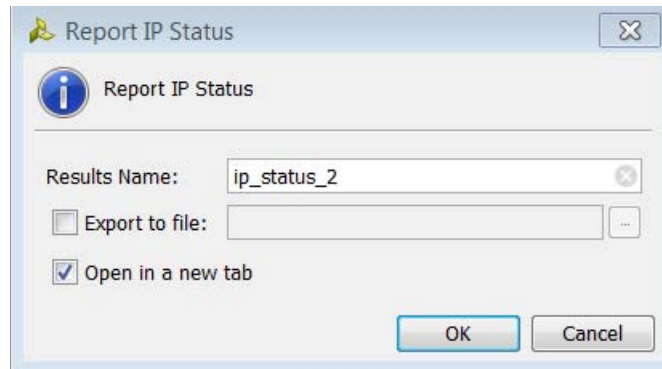


図 2-3 : IP ステータス レポートの生成

[IP Status] ビューが新しく作成され、結果が表示されます。複数の run がある場合は、このビュー内にそれぞれのタブで表示されます (図 2-4)。ここからは、一部の IP またはすべての IP を選択してアップグレードできます。IP を右クリックすると、変更ログを表示したり、IP の製品ガイドを表示できます。

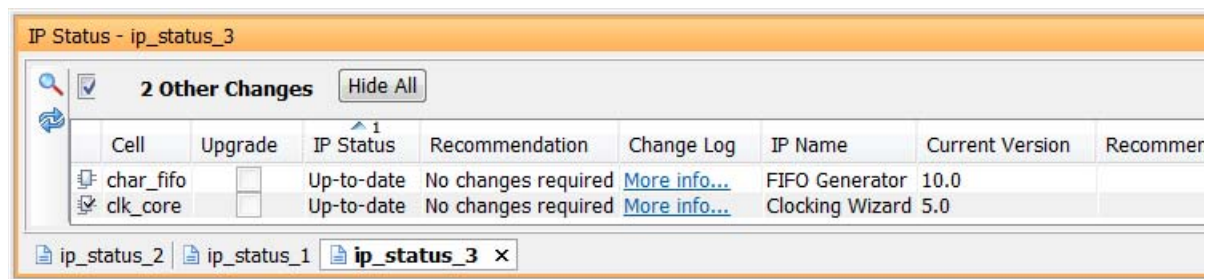


図 2-4 : [IP Status] ビュー

このレポートは、Tcl コマンドの `report_ip_status` をコマンド ラインから使用しても生成できます。-file <file_name> オプションを使用すると、テキスト ファイルを作成することもできます。

IP と一緒に古いプロジェクトを読み込むと、そのデザインの IP のステータスを確認するために [IP Status Report] が開きます。

IP カタログからの IP インスタンスの作成

IP インスタンスを作成するには、[Window] → [IP Catalog] をクリックするか、Flow Navigator で [Project Manager] → [IP Catalog] をクリックして IP カタログを開きます。IP コアはカテゴリごとにツリー形式で表示され、IP バージョン、AXI (Advanced eXtensible Interface) プロトコル準拠、ステータス、ライセンス要件などが示されます。IP コアを選択

すると、その説明がビューの下部に表示されます。カタログから IP コアを選択して、その IP に関するさまざまな情報を確認できます。

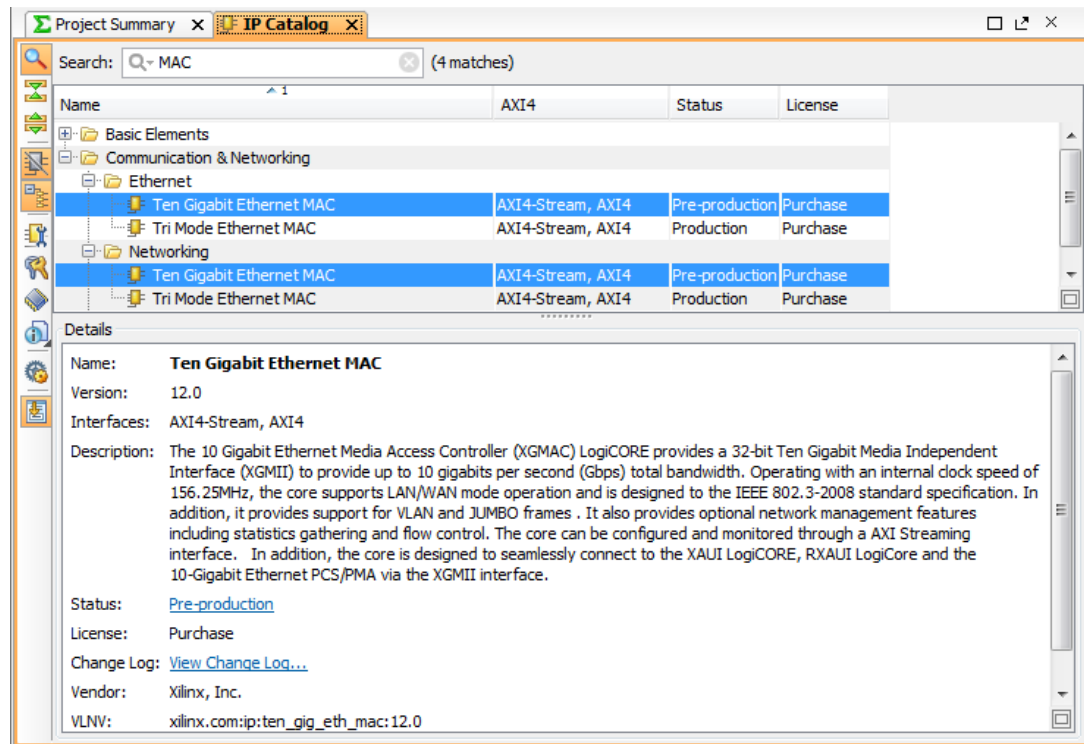


図 2-5 : [IP Catalog] ビュー

IP カタログのツールバーおよびポップアップメニューには、次のコマンド オプションがあります。

- **[Show Search]** : カタログ内で文字列を検索するための [Search] フィールドを表示します。
- **[Collapse All]/[Expand All]** : IP カタログのツリーを展開または閉じます。
- **[Hide]** : 選択したデバイス ファミリーと互換性のある IP のみを表示します。
- **[Group by Category]** : 分類および検索がしやすいように、リストをグループ分けするか、フラットにします。
- **[Customize IP]** : 選択した IP のカスタマイズ ウィンドウを開きます。
- **[License Status]** : 選択した IP のライセンス要件とステータスを表示します。
- **[Compatible Families]** : 選択した IP と互換性のあるデバイス ファミリーすべてをリストします。
- **[View Information]** : 製品ガイド、バージョン情報、ウェブページ、アンサーなど、選択した IP の資料を表示します。
- **[IP Settings]** : IP カタログ、IP 生成、IP パッケージャーを設定します。
- **[Automatically scroll to selected objects]** : 開いているビューの選択したオブジェクトにジャンプする機能をオン/オフにします。

IP に関する資料を表示するには、その IP を右クリックします (図 2-6)。[Product Guide] を選択すると、ウェブブラウザが開きます。

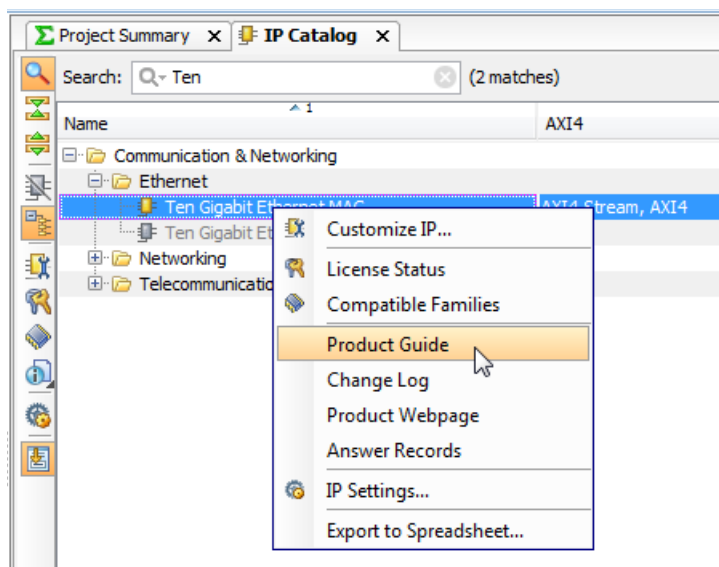


図 2-6 : 製品ガイドの表示

右クリックで [Change Log] をクリックすると、変更ログを表示することもできます。変更ログには、その IP に対する最新の変更点の概要が表示されます。または、IP カタログで IP を右クリックしても、これらの表示オプションが表示されます。

IP のカスタマイズ

IP カタログからコアを選択し、パラメーター値を指定することにより、デザイン要件に合わせて IP をカスタマイズできます。

1. [IP Catalog] ビューからカスタマイズする IP を選択します。
2. 選択した IP をダブルクリックするか、ツールバーまたはポップアップ メニューから [Customize IP] コマンドを実行します。

図 2-7 は、FIFO Generator IP の [Customize IP] ダイアログ ボックスを示しています。

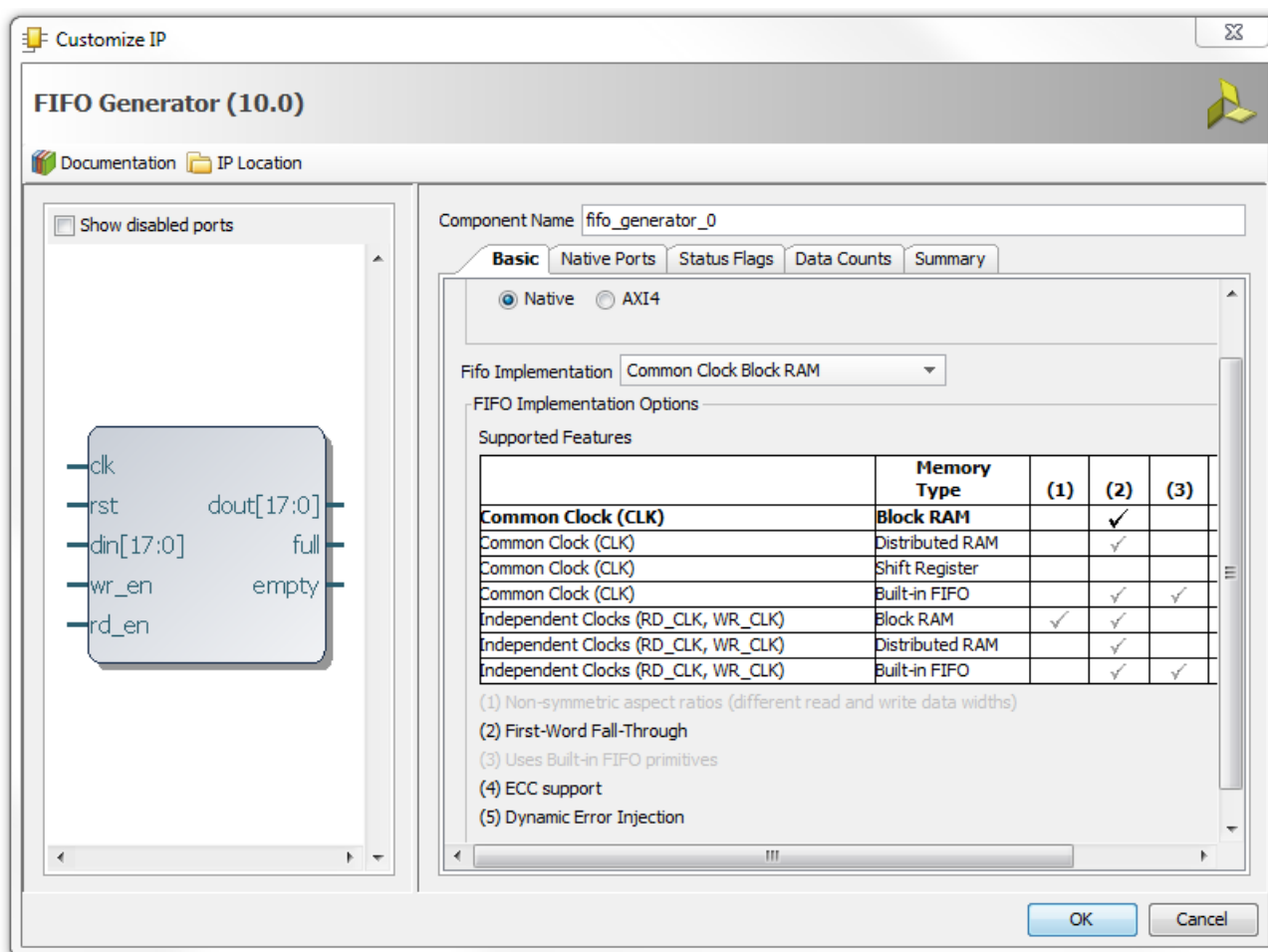


図 2-7 : [Customize IP] ダイアログ ボックス

このダイアログ ボックスに、IP をカスタマイズするためのパラメーターが表示されます。[Customize IP] ダイアログ ボックスは選択した IP コアのタイプによって異なり、パラメーターを指定するタブが 1 つまたは複数表示されます。ダイアログ ボックス上部のタブをクリックし、各ページを表示してパラメーターを設定します。[Customize IP] ダイアログ ボックスには、IP シンボルと、選択した IP によって周波数応答グラフ、リソース予測、AXI4-Stream ポート構造なども表示されます。IP シンボルでは、Vivado IDE の [Schematic] ビューと同じズーム、サイズ変更、自動フィット機能を使用できます。

[Documentation] → [Product Guide] をクリックすると、サポートされる IP コアの資料がウェブブラウザに開きます。

[Customize IP] ダイアログ ボックスでパラメーターの設定を完了したら、[OK] をクリックします。IP コアおよびインスタンス化テンプレートがデザイン ソースとしてプロジェクトに追加されます。[Sources] ビューでカスタマイズした IP コアを右クリックして [Generate Output Products] をクリックすると、IP コアの出力ファイルが作成されます。この時点では、コアは合成されません。プロジェクトに IP コアを追加して合成を実行すると、デザインのほかのソースと共に IP が自動的に合成されます。この機能により、デザインに含まれる複数の IP コアを短時間でインスタンス化できます。プロジェクトに IP を追加するたびに合成を実行する必要はありません。カスタマイズしてプロジェクトに追加した IP コアは、[Sources] ビューに表示されます。このタブでコアのツリーを展開するとコアを構成するさまざまなファイルが表示され、コアを選択すると [Source File Properties] ビューにプロパティが表示されます。

IP のカスタマイズは、create_ip Tcl コマンドでも実行できます。例：

```
create_ip -name fifo_generator -version 10.0 -vendor xilinx.com -library ip
-module_name fifo_gen
```

注記: create_ip Tcl コマンドを実行すると、ソース ファイルは作成されますが、出力ターゲットは作成されません。

[Sources] ビューで IP コアを右クリックして [Re-customize IP] をクリックすると、[Customize IP] ダイアログ ボックスが再度表示され、コアのパラメーターを変更できます。また、[Sources] ビューで IP コアを右クリックして [Upgrade IP] をクリックすると、カスタマイズされた IP をザイリンクス IP カタログの最新バージョンにアップグレードし、現在の IP コアからのカスタマイズを適用できます。

- Vivado Design Suite でサポートされる IP の詳細は、http://japan.xilinx.com/cgi-bin/search/iplocator.pl?_ProductType=Core&_SearchText=vivado を参照してください。
- 各 IP に関する情報は、<http://japan.xilinx.com/ipcenter/> または IP カタログを参照してください。AXI IP の詳細は、<http://japan.xilinx.com/ipcenter/axi4.htm> を参照してください。

IP 出力ファイルの生成

必要に応じて、サンプル デザイン、インスタンス化テンプレート、シミュレーション、合成など、IP インスタンスのさまざまな出力ファイルを生成できます。形式の中には、デフォルトで生成されるものもあります。IP カタログの [IP Settings] オプションで生成されるデフォルトの出力ファイルは変更可能です。選択した IP インスタンスのその他の出力ファイルを生成するには、右クリックして表示される [Generate Output Products] オプションを使用します。

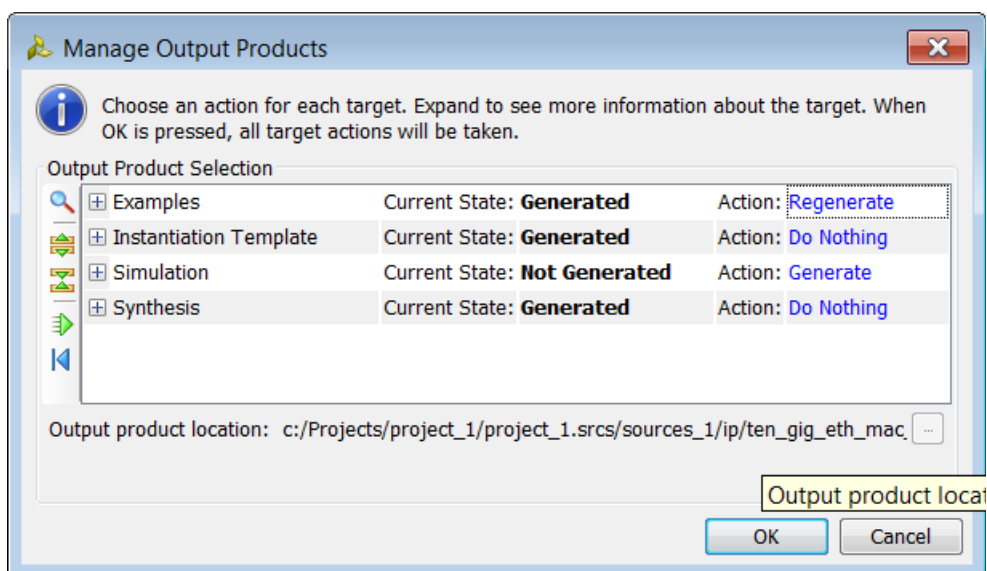


図 2-8 : IP 出力ファイルの管理

サンプルを編集したりして出力ファイルを変更した場合は、[Regenerate] をクリックして元のファイルを生成し直すことができます。

生成し直す Tcl コマンドは generate_target で、次のように使用します。

```
generate_target {instantiation_template synthesis} [get_ips fifo_gen]
```

fifo_gen は、前にモジュール名に使用したように、カスタマイズした IP の名前です。

IP インスタンスの再カスタマイズ

IP インスタンスの出力ファイルを生成したら、次のように IP インスタンスを再カスタマイズできます。

1. [Sources] ビューの [IP Sources] タブで IP を右クリックし、[Re-customize IP] コマンドをクリックします。

注記：または、IP をダブルクリックします。

2. [Re-Customize IP] ダイアログ ボックスで、そのインスタンスに関連するパラメーターを変更し、[OK] をクリックします。
-

IP インスタンスの出力ファイルのリセット

IP インスタンスの出力ファイルはいつでもリセットして、選択されている生成ファイルを削除できます。現在の IP インスタンスの出力ファイルを削除するには、[Sources] ビューの [IP Sources] タブでコアを右クリックし、[Reset Output Products] をクリックします。[Reset Output Products] コマンドを使用すると、リセットする出力ファイルを 1 つ選択するか、そのインスタンスのすべての出力ファイルをリセットできます。現在の出力ファイルのデータは削除されるので、その出力ファイルは生成し直す必要があります。

相当する Tcl コマンドは、次のとおりです。

```
reset_ip [get_ips fifo_gen]
```

IP のプロジェクト設定

図 2-9 のようにグローバル IP プロジェクト設定をすると、IP のカスタマイズの生産性が上がります。

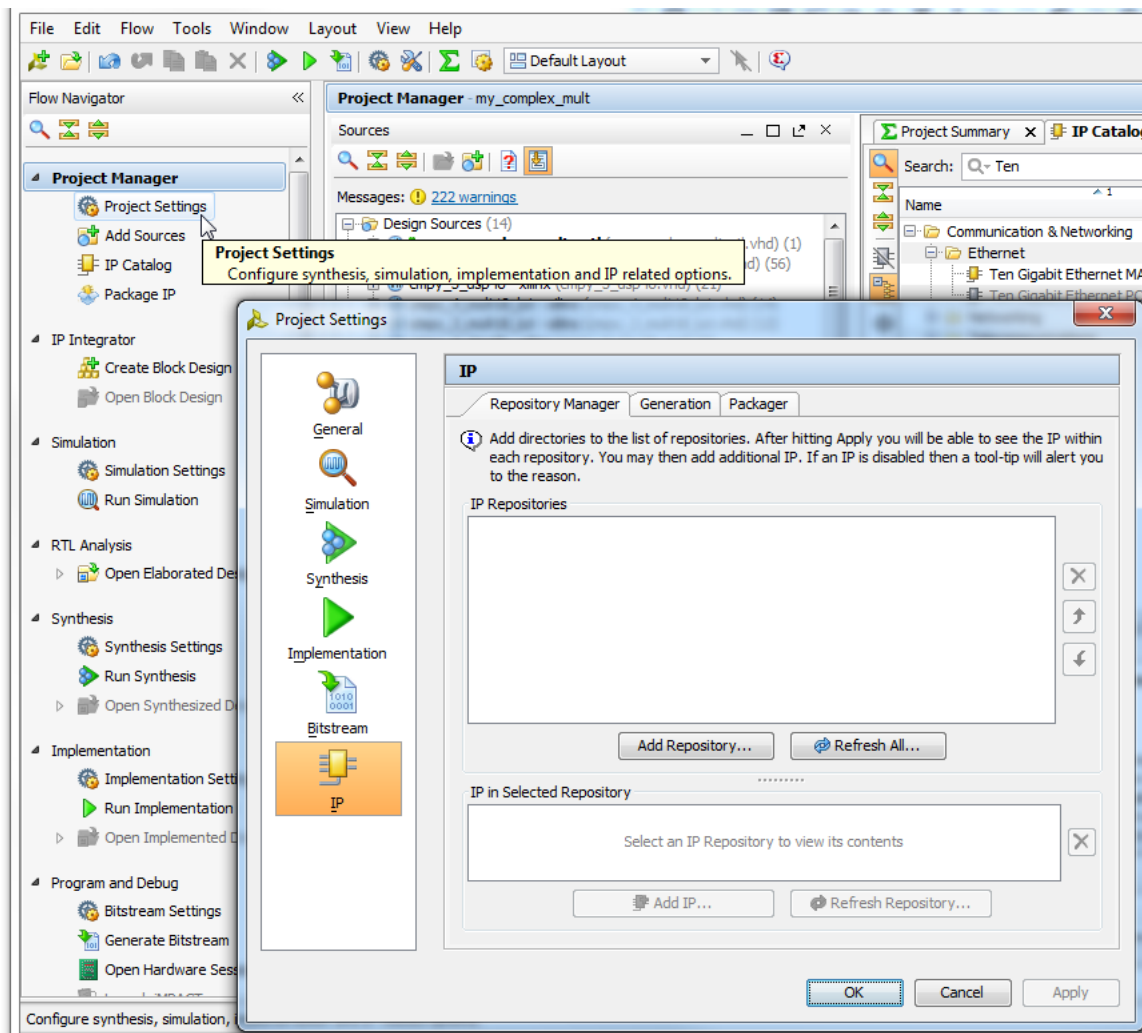


図 2-9 : IP のプロジェクト設定

デフォルトの設定には、次の 3 つの主なカテゴリが含まれます。

- **[Repository Manager]** タブ : IP リポジトリ リストに追加するディレクトリを指定します。IP は、ユーザーがパッケージするか、サードパーティから取得できます。[Apply] をクリックすると、各リポジトリに含まれる IP が表示されます。
- **[Generation]** タブ : デフォルトで生成される IP の出力ファイルを設定します。
- **[Packager]** タブ : ベンダー、ライブラリ、および分類を含めて、新しい IP をパッケージする際のデフォルト値を設定します。このタブでは、IP パッケージャーを開いたときのデフォルト ビヘイビアを設定でき、自動的にフィルターされるファイル拡張子を指定できます。

注記 : 必要であれば、IP パッケージ プロセス中に IP をパッケージする際のデフォルト値を変更できます。

IP 設定および IP カタログは、RTL プロジェクトまたは Getting Started ページから [Manage IP] リンクを使用した場合にのみ使用可能です。[Manage IP] を使用する場合、プロジェクトを作成しない限り、一部の IP 設定しか使用できません。

IP のインスタンスシート

IP インスタンスをカスタマイズしてプロジェクトに追加すると、[Sources] ビューの [IP Sources] タブに表示されます。IP インスタンスを展開すると、[Implementation Template] フォルダにインスタンス化テンプレートを含む VHO/VEO ファイルが表示されます。このインスタンス化テンプレートを RTL デザインにコピーして貼り付けることができます。

図 2-10 に、FIFO Generator コアのインスタンス化テンプレートを示します。

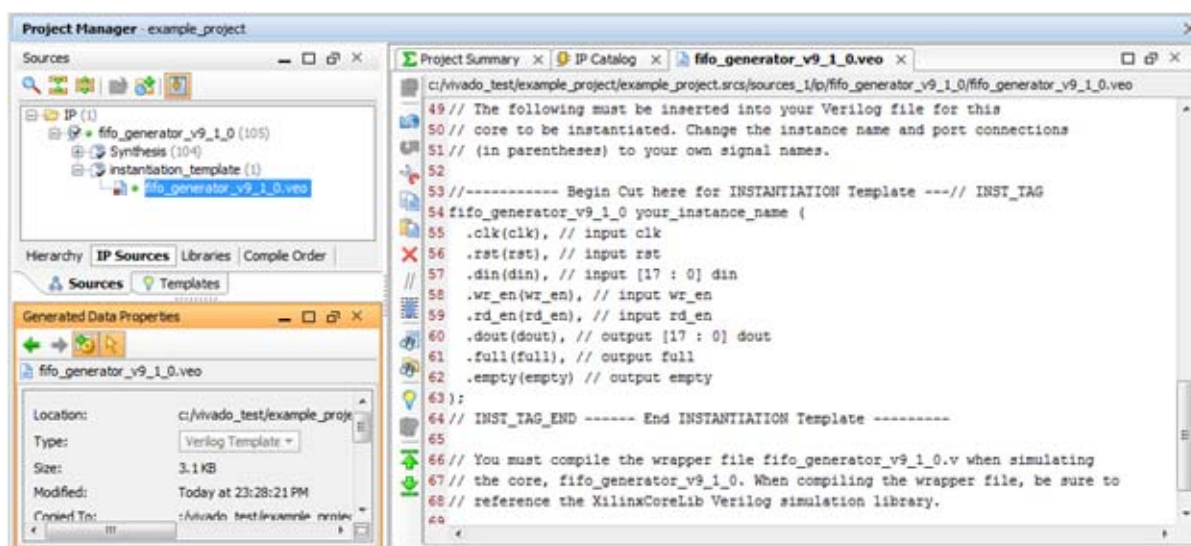


図 2-10: IP のインスタンス化テンプレート RTL コード

1. インスタンス化テンプレートをデザインで使用するには、IP コアの VEO または VHO テンプレートファイルと RTL デザインファイルを、[Sources] ビューでダブルクリックするか、右クリックして [Open File] をクリックし、テキストエディターで開きます。
2. VEO または VHO テンプレートファイルでインスタンス化テンプレートを選択してコピーし、RTL デザインの適切な位置に貼り付けます。
3. IP テンプレートをデザインに組み込むために必要な変更を加えます。
4. IP コアをデザインに正しくインスタンス化したら、IP コアをデザインの残りの部分と共に合成できます。

IP の合成

デフォルトでは、デザインを合成すると IP も一緒に合成されます。ロジックを変更するたびに IP が再合成されないようにするには、IP を前もって合成しておきます。IP を選択し、別の Out-of-Context モードの run として実行すると、

ボトムアップフローを設定できます。これには、[Sources] ビューで IP を右クリックし、[Set as Out-of-Context Module] をクリックします (図 2-11)。

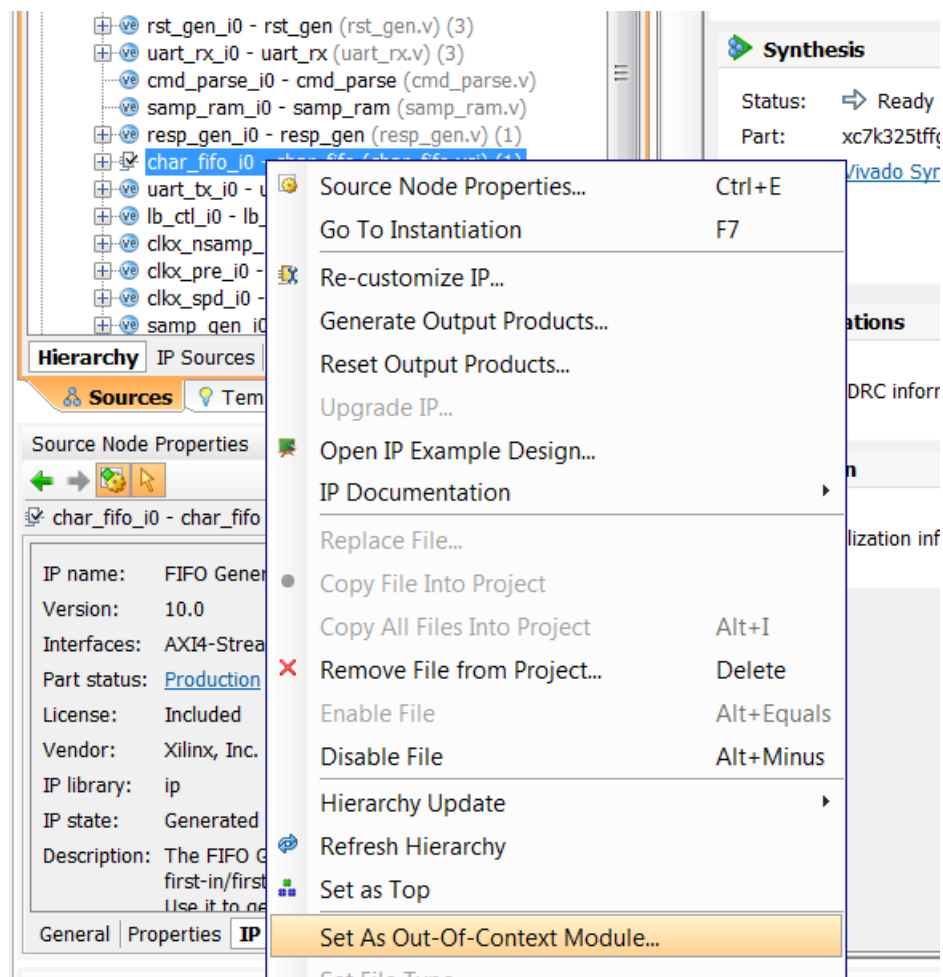


図 2-11 : IP を Out-of-Context モジュールに設定

IP を Out-of-Context モードにすると、新しいブロック ファイル セットが作成されます。この設定には、名前を付けることができます (図 2-12)。

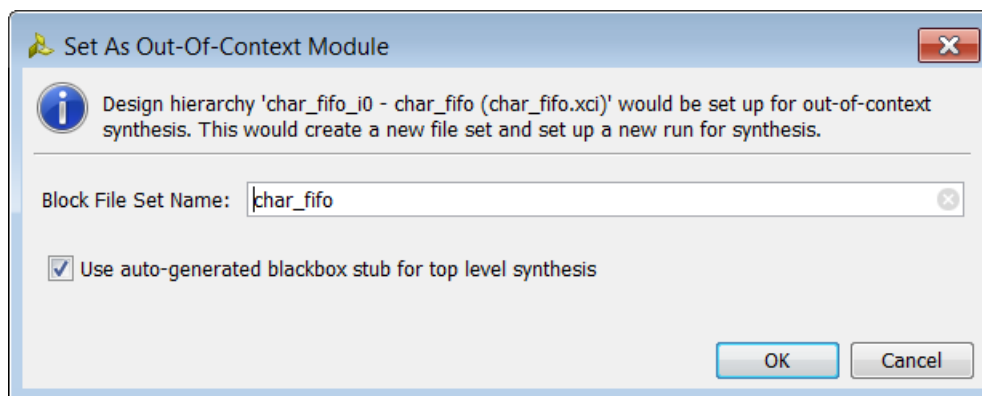


図 2-12 : ブロック ファイル の名前の設定

新しいブロック ファイル セットを作成すると、次のようになります。

- 新しいファイル セットが作成され、IP の合成ファイルがそれにコピーされます。
- 新しい合成 run が I/O バッファを挿入しないようにコンフィギュレーションされて設定されます。
- [Use auto-generated blackbox stub for top level synthesis] をオンにすると、スタブが自動的に作成され、プロジェクトの IP インスタンスがスタブと置き換えられます。

スタブ ファイルは、入力、出力、ブラック ボックス属性設定を含む下位レベルのファイルです。最上位合成を再実行すると、ボトムアップ合成のフローに挿入したスタブ ファイルが含まれ、下位レベルがブラック ボックスとしてコンパイルされます。最上位のインプリメンテーション run により下位レベルのネットリストにリンクされ、デザインが完了します。

新しい run を開始するには、[Design Runs] タブで Out-Of-Context Module Runs フォルダーから IP run を選択して、右クリックで [Launch Runs] をクリックします。デザイン チェックポイント (DCP) ファイルが run ディレクトリに作成されます。このファイルは、最上位のインプリメンテーションで使用されます。

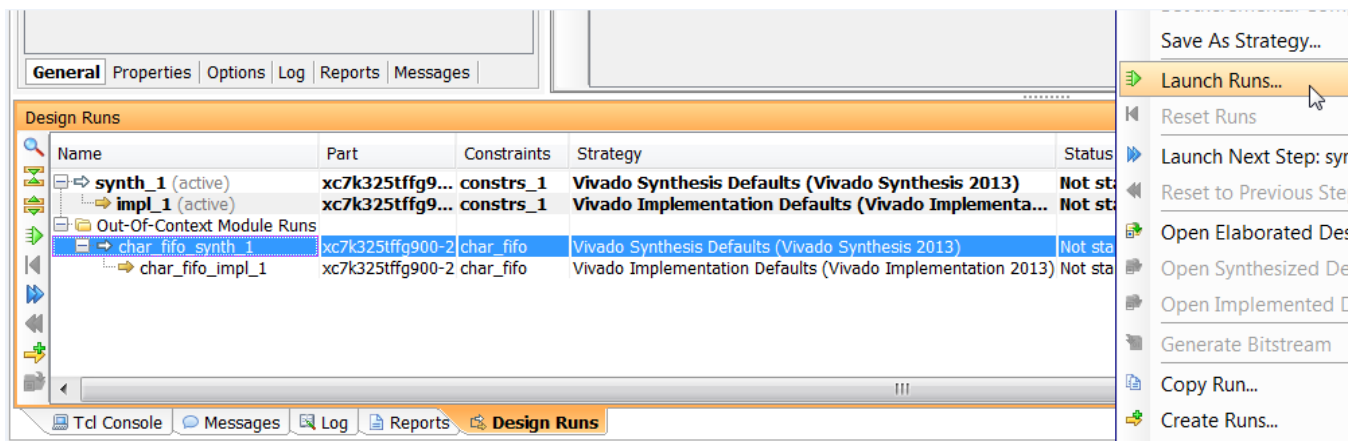


図 2-13 : 新しい Out-Of-Context run の開始

注記 : 複数の Out-Of-Context の run を並行して開始することもできます。Ctrl キーを押しながら run をクリックしていき、右クリックで [Launch Runs] を選択してください。

注記 : この段階では、最上位 run の開始前に Out-Of-Context の run すべてを手動で開始する必要があります。

IP を Out-Of-Context フローで使用しないように変更するには、[Sources] ビューで IP を右クリックし、[Unset Out-Of-Context Module] をクリックします。これにより、作成した合成 run が削除され、IP がデフォルトのファイルセットに戻され、作成されたスタブが削除されます。

このフローに使用できるのは Out-Of-Context の XDC ファイルを含む IP のみです。Out-Of-Context モードの XDC ファイルには、その IP のデフォルトのクロッキング情報が含まれます。IP は Out-Of-Context モードで生成されるので、最上位のクロッキング情報は含まれません。ほとんどの IP には、合成時にタイミング解析で使用するデフォルトのクロッキング定義が含まれます。IP に Out-Of-Context モードの XDC ファイルが含まれているかどうかは、[IP Sources] ビューの Synthesis フォルダーに <component_name>_ooc.xdc ファイルがあるかどうかで確認できます。

Out-Of-Context フローは、IP をカスタマイズした後 [Manage IP] フローを使用しても実行できます。ユーザーはプロジェクトを作成し、上記の手順に従うか、[Project Settings] の [Synthesis] ページで [More Options] に -mode out_of_context を入力する必要があります。これを設定すると、合成で I/O バッファがモジュールに挿入されないようにできます。I/O バッファは、必要に応じて後のインプリメンテーションで追加できます。

注記 : このフローの手順については、『Vivado Design Suite チュートリアル : IP を使用した設計』(UG896) を参照してください。

IP のシミュレーション

RTL デザイン内での IP のシミュレーション

Flow Navigator で [Run Simulation] をクリックすると、デザイン全体と共に IP コアがシミュレーションされます。Vivado IDE では、IP と共に供給されているシミュレーション ソースを使用して、デザイン全体の論理シミュレーションが実行されます。ビヘイビア モデル、テキスト形式の合成可能なソース、または暗号化された合成可能なソースが VHDL または Verilog シミュレーション モデルとして提供されています。シミュレーション ソースとして暗号化されたファイルが提供される IP では、選択されたシミュレータ (Vivado シミュレータまたは QuestaSim/ModelSim) のシミュレーション ソースのコンパイルは Vivado IDE で管理されます。また、プロジェクトのターゲット言語の IP シミュレーション ソースがない場合は、混合モードのシミュレーション用にプロジェクトが設定されます。

その他のシミュレータを使用したシミュレーション

サードパーティのシミュレータを使用すると、Vivado IDE に含まれるザイリンクス IP のビヘイビア シミュレーションまたはネットリスト シミュレーションのいずれかを実行できます。Vivado IDE のほとんどの IP には、主なシミュレータすべてでサポートされる業界標準の IEEE P1735 を使用して暗号化された HDL ファイルが含まれます。

サードパーティ シミュレータでビヘイビア シミュレーションを実行する前に、まずシミュレーションに必要なファイルとそれらが含まれた関連するライブラリを指定する必要があります。同様に、合成後のシミュレーションシミュレーションを実行する前には、構造シミュレーション モデル (EDIF、Verilog、VHDL) を作成する必要があります。

シミュレーションに必要な HDL ファイルの指定

シミュレーションに必要なファイルと関連するライブラリ情報のリストを作成するには、Tcl スクリプトを使用できます。IP の XCI ファイルを使用すると、シミュレーションに使用されるファイルなど、それに関連するファイルを入手できます。

次の例の場合、バイナリ カウンターの IP のカスタマイズが使用されます。次のコマンドを使用すると、この IP に関連するファイルすべてが入手できます。

```
get_files -all -of_objects [get_files c_counter_binary_0.xci]
```

暗号化された HDL ファイルを取得するには、-all オプションを使用する必要があります。

その後、Tcl のフィルター コマンドを使用してシミュレーションに必要なファイルのみをフィルターします。HDL ファイルにはすべてフィルター可能なプロパティが付いています。ファイルに関連するさまざまなプロパティを確認するには、report_property コマンドを使用します。HDL ファイルには、USED_IN_SIMULATION というプロパティが含まれます。このプロパティが true であれば、IP のシミュレーションにそのファイルが必要なファイルとして設定されています。

```
filter [get_files -all -of_objects [get_files \
c_counter_binary_0.xci]] {USED_IN_SIMULATION}
```

これで、IP シミュレーションに必要な HDL ファイルがすべてリストされます。最後に、各ファイルを関連付けるライブラリを決定します。

次は、シミュレーションに必要な HDL ファイルをそれが含まれるライブラリと共にリストする Tcl スクリプトです。この結果は、シミュレータの do ファイルに組み込まれます。

```
# Get the list of files required for simulation
set ip_files [filter [get_files -all -of_objects [get_files \
tri_mode_ethernet_mac_0.xci]] {USED_IN_SIMULATION}]
# For each of these files, get the library information
foreach file $ip_files {
    puts "[get_property LIBRARY $file] $file"
}
```

ネットリストのシミュレーション

ネットリストのシミュレーションには、暗号化されたシミュレーション ソースを含む IP の構造シミュレーション モデルを作成する必要があります。デザインを 1 つの言語でシミュレーションできるように、構造シミュレーション モデルを作成する必要がある場合もあります。構造シミュレーション モデルを作成するには、まず IP をスタンドアロンで合成する必要があります。

IP をスタンドアロンで合成して合成後のネットリストを生成する方法については、「[IP の合成](#)」を参照してください。

合成が終了したら、次の Tcl コマンドを使用して IP シミュレーション モデルを生成します。

- Verilog:write_verilog -mode funcsim <corename>.v
- VHDL:write_vhdl -mode funcsim <corename>.vhd

デザイン内での IP の制約

Vivado IDE は、IP を含むデザイン全体の XDC タイミング制約および物理制約を管理します。プロジェクトに複数回インスタンス化されている IP に含まれるデザイン エLEMENT の制約の関連付けおよび固有化は、Vivado IDE で処理されます。IP カタログの一部の IP では、カスタマイズに基づいて、IP 特定の XDC 制約が生成されます。IP 特定の XDC 制約は、デザインの合成およびインプリメンテーション中、ユーザー定義の XDC 制約が処理された後に読み込まれます。IP 特定の XDC 制約の特性を含む XDC 制約の詳細は、『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。

IP のサンプル デザインの使用

IP コアでサポートされる場合、サンプル デザインを生成できます。サンプル デザインは、「[IP 出力ファイルの生成](#)」で説明されるように、ユーザーが生成できます。これにより、RTL が生成されます。RTL には、サンプルの最上位にインスタンス化された IP が表示されます。

サンプル デザインを新規 Vivado IDE セッションで開くには、[IP Sources] タブで IP を右クリックして [Open IP Example Design] を選択します。[図 2-14](#) に示すような、新しいプロジェクトのディレクトリを指定できるポップアップ ウィンドウが表示されます。デフォルトでは、サンプル デザインは example_project というディレクトリ内のプロジェクト

ディレクトリに保存されます。IP のサンプル デザインはそれぞれ別のディレクトリに保存されます。たとえば、char_fifo という IP の場合、char_fifo_example というディレクトリに保存されます。

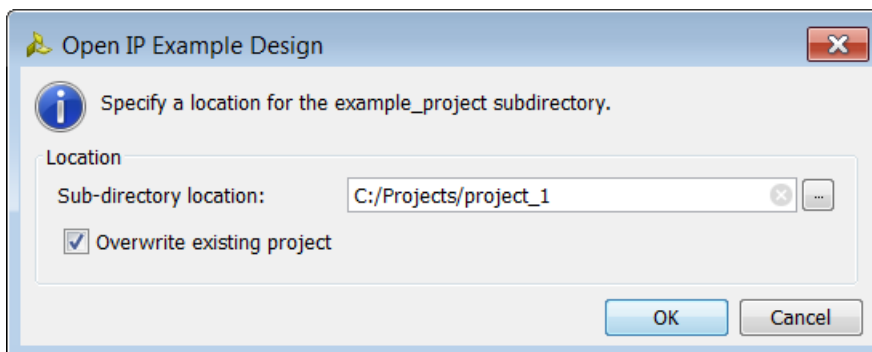


図 2-14 : IP サンプル デザインを開く

サンプル デザインがまだ生成されていない場合はまずそれが生成され、別の新しい Vivado IDE セッションで開き、[Sources] ビューに表示されます (図 2-15)。IP はサンプル XDC 制約ファイルを使用してサンプル デザインにインスタンス化されるので、IP をさらに詳しく評価できます。

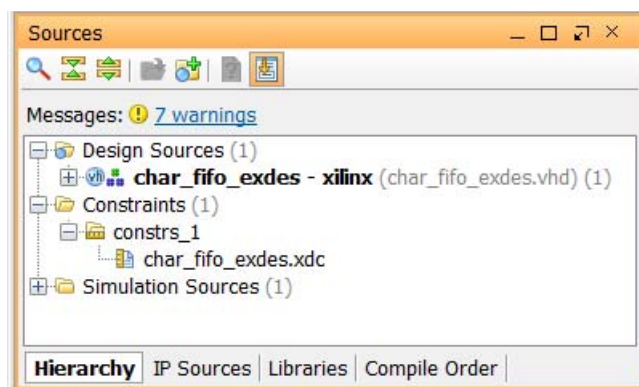


図 2-15 : IP サンプル デザイン インスタンスと制約ファイル

有償ライセンス IP の使用

Vivado IP カタログの [License] 列には、[Included] または [Purchase] と表示されます。これらは、次を意味します。

- [Included] : Vivado Design Suite ツール内で無償でライセンス付与されるザイリンクス LogiCORE™ IP コアで、ザイリンクス [エンド ユーザー ライセンス契約](#)が適用されます。
- [Purchase] : 有償のザイリンクス LogiCORE IP コアで、[コア ライセンス契約](#)が適用されます。これらのコアの評価版には、[コア評価ライセンス契約](#)が適用されます。

IP ライセンスの取得方法は、ザイリンクス ライセンス サイト http://japan.xilinx.com/ipcenter/ip_license/ip_licensing.htm を参照してください。

有償の IP では、[Customize IP] ダイアログ ボックスの [OK] ボタンは、評価ライセンスまたは購入したライセンスが検出されるまで、次の図に示すように無効になります。

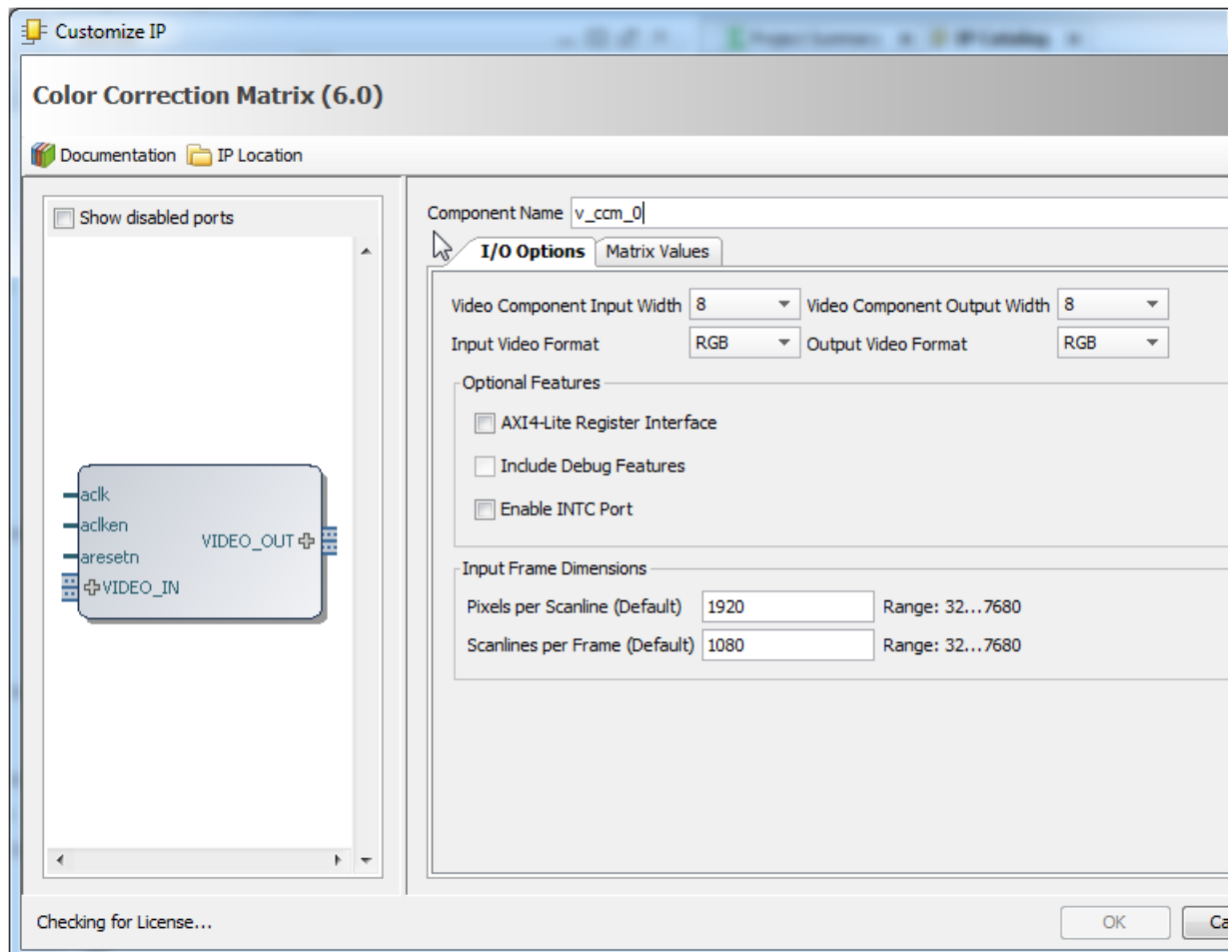


図 2-16 : 有償の IP の [OK] ボタンはライセンスを検出中は無効

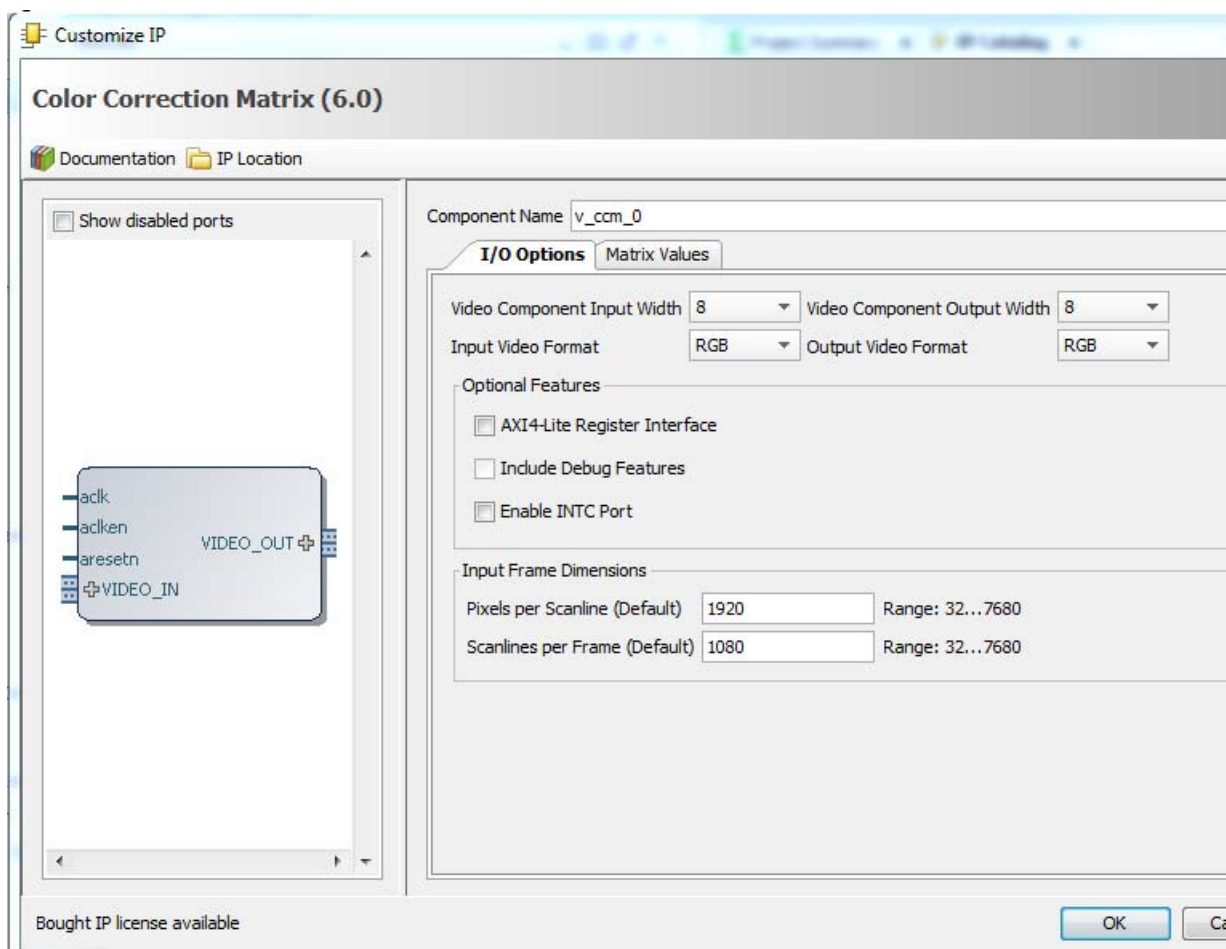


図 2-17 : 有償の IP のライセンスが検出され [OK] ボタンが有効になる

既存 IP の最新バージョンへのアップグレード

IP を最新バージョンにアップグレードするには、次の方法を使用します。

- 右クリックで[Upgrade IP] を選択します。
- 「[IP へのアクセス](#)」で説明されるように、[Report IP Status] コマンドを使用します。

自動アップグレードがサポートされる IP では、図 2-18 に示すように [Sources] ビューで IP インスタンスを右クリックして [Upgrade IP] をクリックすると、IP がアップグレードされます。

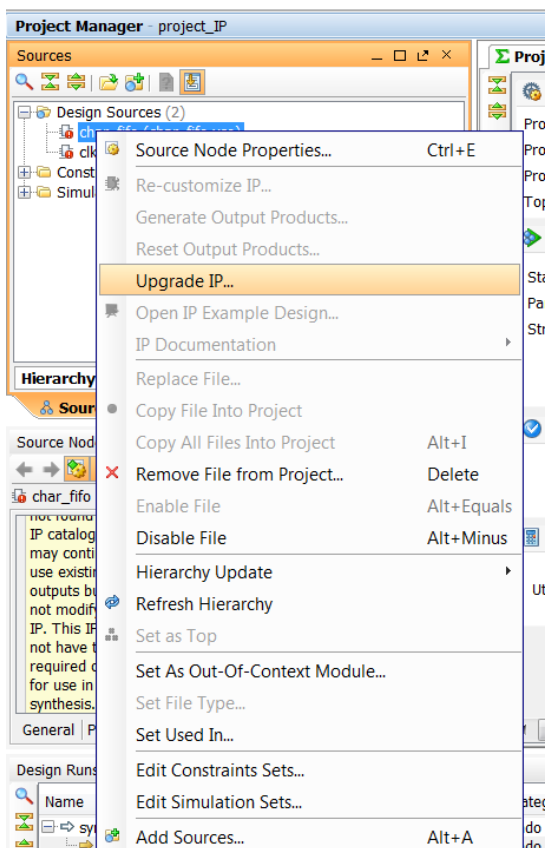


図 2-18 : IP のアップグレード

IP のアップグレードは、次の例のように Tcl コマンド `upgrade_ip` を使用しても実行できます。

```
upgrade_ip [get_ips clk_core]
```

コマンドに引数を指定しない場合は、アップグレードパスがあれば、プロジェクトに含まれるすべての IP が最新のバージョンにアップグレードされます。

IP 操作の Tcl コマンド

Vivado IP カタログは Vivado IDE に統合されており、Vivado IDE および Tcl 設計環境からスムーズにアクセスできます。バッチモード用に、IP の作成、カスタマイズ、出力ファイルの生成などの GUI で実行する各操作に対応する Tcl コマンドがあり、GUI で実行できる操作はスクリプトで自動化して実行できます。IP のパラメーターも、Tcl コンソールから直接設定できます。例：

アキュムレータ IP のインスタンスの作成：

```
create_ip -name c_accum -version 12.0 -vendor xilinx.com -library ip -module_name c_accum_0
```

入力幅および出力幅などのカスタマイズパラメーターの変更：

```
set_property -dict [list CONFIG.Input_Width {10} CONFIG.Output_Width {10}] [get_ips c_accum_0]
```

選択した出力ファイルの生成:

```
generate_target {synthesis instantiation_template simulation} [get_ips c_accum_0]
```

生成する出力ファイルのリセット:

```
reset_target all [get_ips c_accum_0]
```

サポートされる IP 関連の Tcl コマンドの詳細は、Tcl コンソールで「help -category IPFlow」と入力してください。

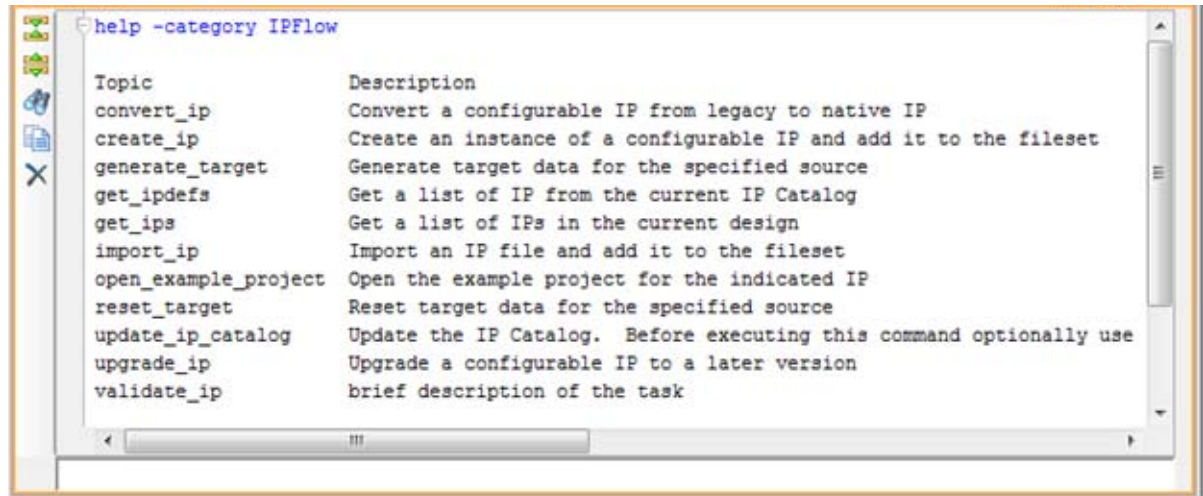


図 2-19: IP 関連の Tcl コマンドに関するヘルプを表示

IP パッケージの基礎

IP パッケージャーとその使用フロー

Vivado™ IP パッケージャーを使用すると、サードパーティの IP を Vivado IP カタログで使えるように準備できます。このように準備されたサードパーティ IP は、Vivado Design Suite のデザインにインスタンス化できます。Vivado Design Suite の IP パッケージフローを使用して IP を開発すると、ザイリンクス IP、サードパーティ IP、またはカスタム IP のいずれも Vivado Design Suite で同様に使用できます。

図 3-1 に、IP パッケージャー フローとその使用モデルを示します。IP の開発時には、IP パッケージャーを使用して IP ファイルと関連データを ZIP ファイルにパッケージ化します。この生成された ZIP ファイルを Vivado Design Suite の IP カタログにインストールすると、パラメーターを選択して IP をカスタマイズし、IP インスタンスを生成できるようになります。IP が正しくパッケージ化されたことを検証するため、各 IP モジュールに対して IP ユーザー フローを実行し、IP が使用可能な状態であるかどうかを検証することをお勧めします。

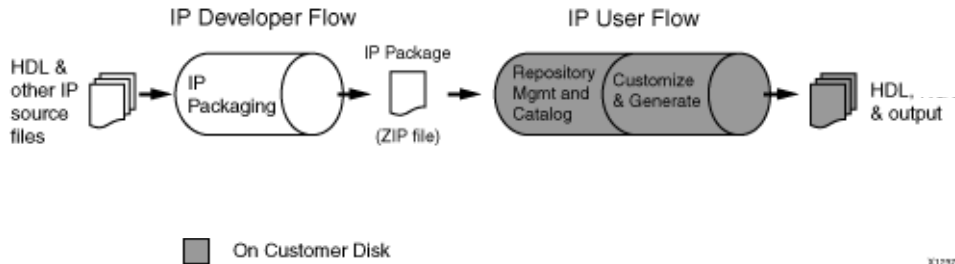


図 3-1 : IP パッケージャーおよびその使用フロー

IP パッケージ化フロー

手順 1 :IP のパッケージ化

IP パッケージャーの出力は IP-XACT コンポーネント ファイルで、ZIP にはデフォルトの GUI ファイル、レポート、再生成ファイルなどが含まれます。次のいずれかを実行できます。

- 既に生成済みの Vivado Design Suite プロジェクトからデザインをパッケージ化
- 新規 Vivado Design Suite プロジェクト ファイルを作成し、Package IP Wizard を使用して IP ソース ファイルと関連データをインポート

IP パッケージャーを起動するには、[Tools] → [Package IP] をクリックします。または、Vivado Design Suite の Tcl コマンドライン インターフェイスを使用してバッチ モードで IP パッケージャーを実行することもできます。

手順 2 :IP 配布の保護

IP ユーザーに IP を配布する際は、保護する必要があります。

IP ユーザーのフロー

手順 1 :IP カタログのアップデート

サードパーティ IP または社内で開発された IP を受信したら、Vivado IDE を起動して Vivado IP カタログに IP を追加します。

手順 2 :IP ライセンスのインストール (オプション)

オプションで、サードパーティ IP プロバイダーから FlexNet ライセンスを取得してインストールします。

手順 3 :サードパーティ IP の使用

Vivado Design Suite を使用して、サードパーティ IP コアを使用したデザインを開始します。

リポジトリの管理

Vivado IP カタログには、サードパーティの IP を追加できるビルトインのリポジトリ管理機能が含まれます。サードパーティの IP が表示されるようにするには、その IP をユーザー マシンからアクセスできるディレクトリに含めておく必要があります。また、Vivado Design Suite を起動して、IP カタログから [IP Settings] を実行して、この新しいユーザー リポジトリのディレクトリを指定して、新規 IP がカタログに含まれるようにする必要があります。

図 3-2 に示すように、リポジトリには標準ザイリンクス リポジトリとユーザー リポジトリの 2 種類があります。標準ザイリンクス リポジトリは Vivado Design Suite ツールの一部として含まれており、常にイネーブルで、ユーザーが変更することはできません。ユーザー リポジトリとは、1 つ以上の IP を含むユーザー マシンからアクセス可能なディレクトリです。ザイリンクスまたはサードパーティの IP プロバイダーは、カタログへの IP アップデートをパッチを介して配布します。

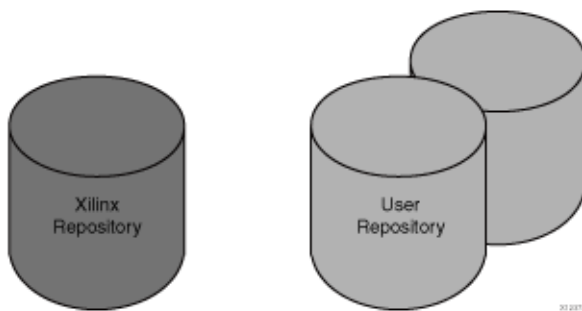


図 3-2: リポジトリ タイプ

ユーザー リポジトリは、追加または削除したり、リポジトリ間の優先順位を決めたりできます。IP は、ベンダー、ライブラリ、名前、バージョンによって分類されます。

複数のリポジトリが参照され、同じ IP が複数のロケーションにある場合は、優先順位の高いリポジトリの IP が表示されます。ザイリンクス IP リポジトリは常にイネーブルになっており、優先順位は一番低くなっています。

プロジェクトでのリポジトリ設定の変更はプロジェクトと共に保存され、そのプロジェクトをどのマシンで開いても、その変更されたリポジトリが表示されます (リポジトリパスが有効である場合)。

注記: ユーザー リポジトリを新しく作成されたプロジェクトに対して使用可能にできます。[Tools] → [Project Settings] → [IP] をクリックします。

IP カタログ

Vivado IP カタログは、IP の検索、詳細情報の確認、関連資料の表示を実行可能な統合リポジトリです。Vivado IP カタログ (図 3-3) にサードパーティ IP またはカスタマー IP を追加すると、Vivado Design Suite フローからその IP にアクセスできるようになります。

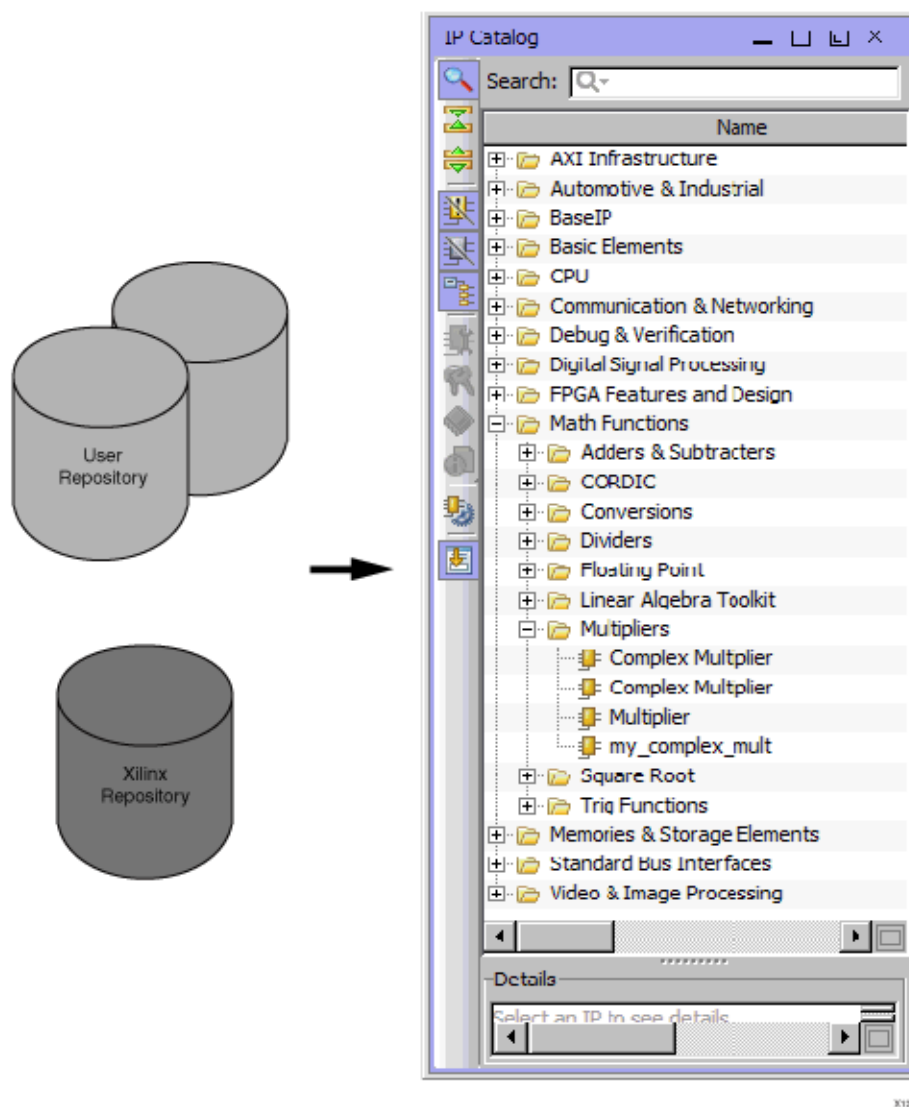


図 3-3 : リポジトリと IP カタログ

IP のカスタマイズと生成

IP のパラメーターを変更して出力を生成するには、IP カタログから [Customize IP] ダイアログ ボックスを開きます。パラメーターを指定したら、[Sources] ビューで IP を右クリックして [Generate] をクリックし、プロジェクト ディレクトリに指定の出力を生成します。出力には、カスタマイズ オプション、ネットリスト、HDL 合成およびシミュレーション ファイル、テストベンチ、サンプル デザインなどが含まれます。

IP パッケージの検証

IP パッケージャーを使用して IP をパッケージ化したら、次を確認することをお勧めします。

- Vivado Design Suite の IP リポジトリに IP を追加し、Vivado IP カタログの GUI に正しく表示されることを確認します。
- Vivado IP カタログの [Customize IP] ダイアログ ボックスを使用して IP のパラメーターを指定し、出力を生成します。
- デザインに IP をインスタンスエートし、Vivado Design Suite フローでそのデザインを実行します。
- サンプルプロジェクトが IP と一緒にパッケージ化されている場合はそれを生成し、Vivado Design Suite で問題なく動作するかどうかを検証します。

IP パッケージャーの入力

入力ファイルのグループ

IP パッケージャーでは、次を含むさまざまな入力ファイル グループがサポートされます。

- HDL 合成
- HDL シミュレーション
- 文書
- HDL テストベンチ
- サンプル デザイン
- インプリメンテーション ファイル (制約および構造ネットリスト ファイルを含む)

パッケージ化された IP に最低限必要なファイル セット

IP には、必要な数のファイル グループを指定できます。最低限必要なファイル グループのセットは特にありませんが、[IP Packager] ビューの [IP File Groups] ページに論理合成、シミュレーション、資料などの典型的なファイル グループのセットが表示されます。これらのファイル グループのいずれかが空の場合、最後の [Review and Package] ページにファイルがないことを示すメッセージが表示されます。

IP パッケージャーの出力

出力パッケージ内の IP デザイン ファイルの分類

出力パッケージのデザイン ファイルは、入力ディレクトリの構造と同じようにフォルダーに分類されます。これらは、[IP File Groups] ページで指定した論理グループに合わせて物理的には分類し直されません。

出力パッケージに含まれるその他のファイル

出力パッケージには、IP デザイン ファイルと IP-XACT XML ファイル以外にも、GUI フォルダーが含まれます。このフォルダーには、ザイリンクス ツールで IP のカスタマイズ GUI およびその他の GUI を表示するためのファイルが含まれます。

IP パッケージ化の手順

次に、Package IP Wizard を使用して IP をパッケージ化する方法を示します。次の手順では、my_complex_mult というサンプル IP デザインを使用します。詳細な手順および演習フォーマットのデザイン データについては、『Vivado Design Suite チュートリアル: IP を使用した設計』を参照してください。

既存の Vivado プロジェクトのパッケージ化または新規プロジェクトの作成

既存の Vivado プロジェクトで IP パッケージャーを起動するか、パッケージ化する IP のプロジェクトを作成します。新規プロジェクトを作成するには、次の手順に従います。

1. Vivado IDE で [New Project] をクリックします。
2. プロジェクト名に「my_complex_mult」と入力し、プロジェクト ディレクトリを確認して、[Next] をクリックします (図 3-4)。

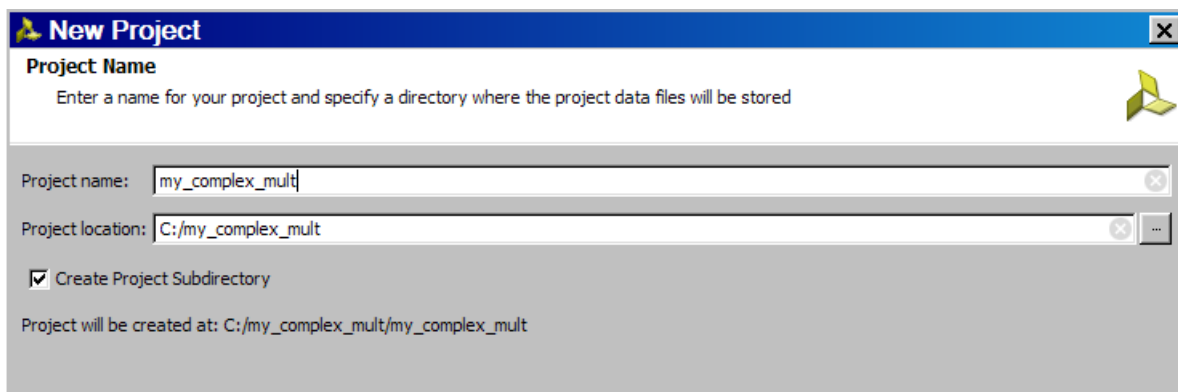


図 3-4 : [New Project] ダイアログ ボックス : [Project Name] ページ

3. [Project Type] ページで [RTL Project] がオンになっていることを確認し、[Next] をクリックします。
4. [Add Sources] ページで、次を実行します。
 - a. cmpy_v3_1、mult_gen_v1l_2、xbip_utils_v2_0 という 3 つの RTL サブディレクトリを追加します。
 - b. 3 つのサブディレクトリの [Library] 列を 図 3-5 のように変更します。
 - c. my_complex_mult_rtl.vhd という最上位の VHDL ファイルを追加します。
 - d. ディレクトリ構造を保持するために通常使用する [Copy Sources into Project] をオフにします。
 - e. ターゲット言語を [VHDL] に設定します。
 - f. [Next] をクリックします。

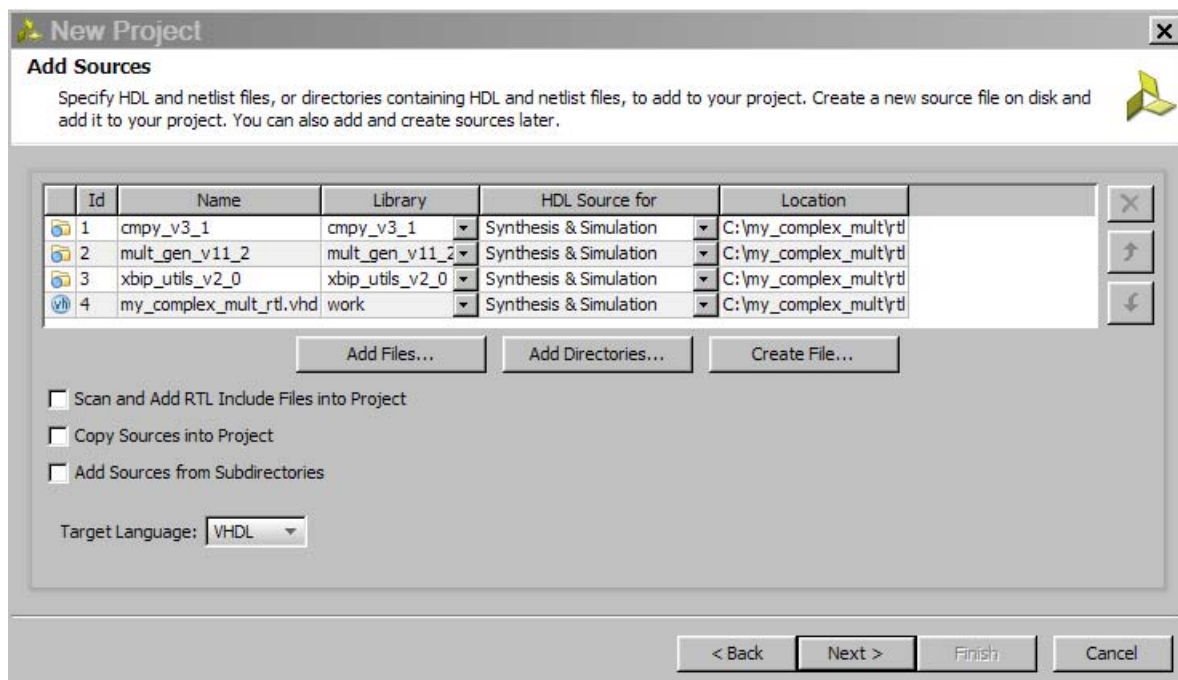


図 3-5 : [New Project] ダイアログ ボックス : [Add Sources] ページ

5. ウィザードの残りのページをクリックしていき、[Finish] をクリックします。

プロジェクトを IP としてパッケージ化

Vivado プロジェクトを IP としてパッケージ化するには、次の手順に従います。

1. Vivado プロジェクトが開いた状態で、[Tools] → [Package IP] をクリックします。
2. [Welcome to the IP Packager] ページで [Next] をクリックします。
3. [Begin IP Creation] ページで [Finish] をクリックします。

プロジェクトに関する情報が自動的に集められ、基本的な IP パッケージが準備エリアに作成されます。

4. [IP Packager Summary] ページで [OK] をクリックします。
5. 図 3-6 に示すように設定し、[Next] をクリックします。

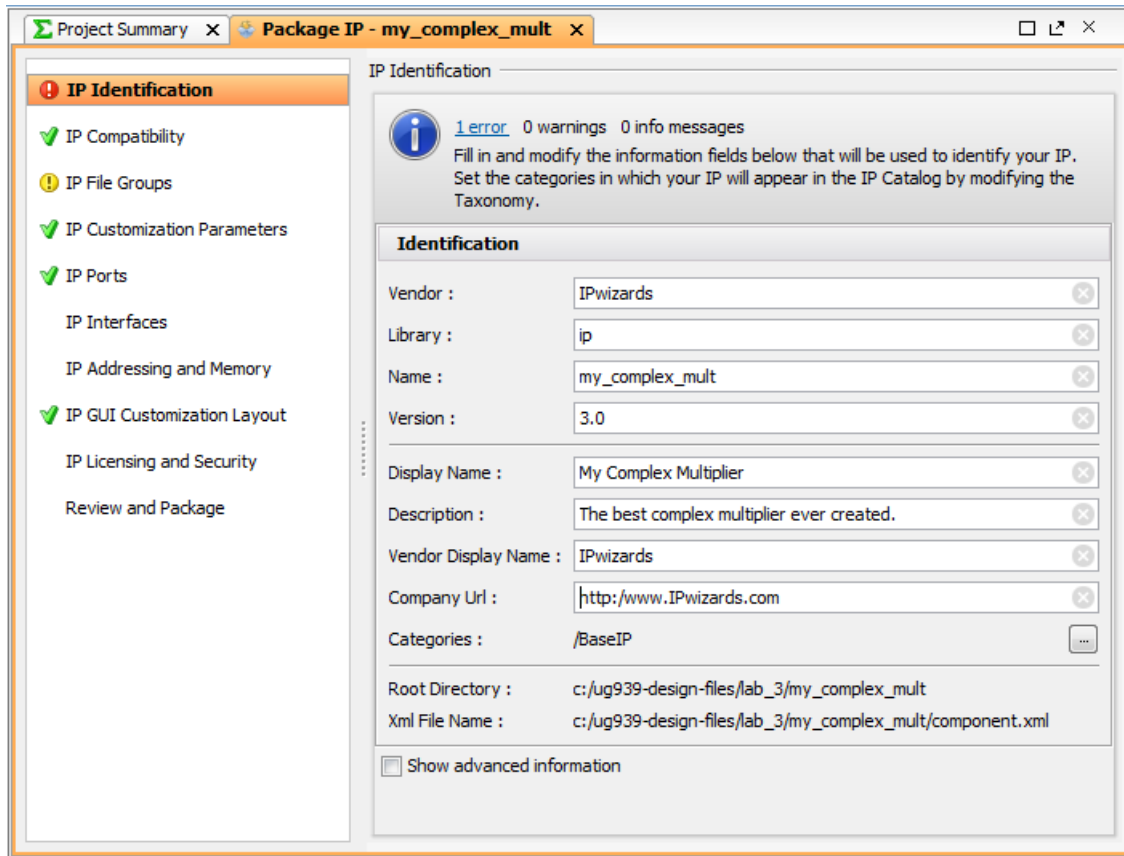


図 3-6 : [IP Identification] ページ

6. [Review and Package] をクリックします (図 3-7)。

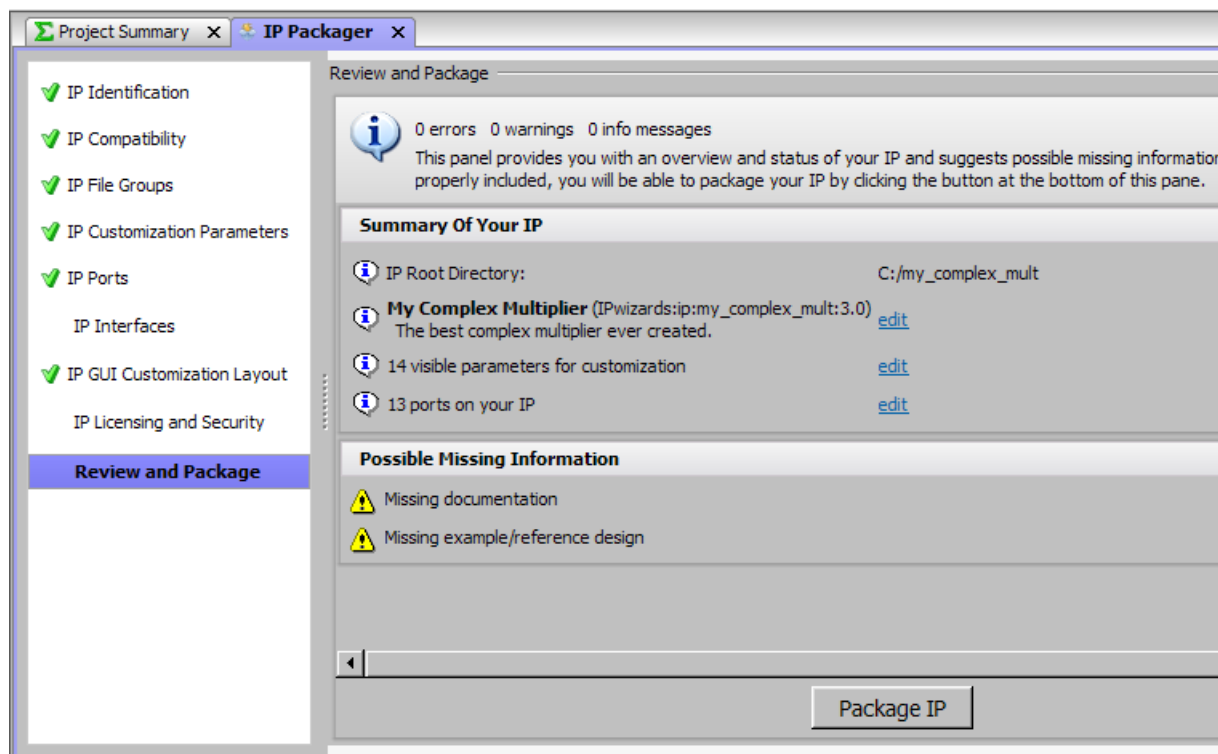


図 3-7 : [Review and Package] ページ

注記 : 不足している情報がある場合、[Possible Missing Information] にリストされます。IP 資料の追加方法はこの後「[HDL 以外のファイルの IP パッケージへの追加](#)」セクションで説明するので、現段階では無視してもかまいません。

7. [Package IP] ボタンをクリックすると、IP ユーザーに送信できる IP パッケージ (ZIP ファイル) が作成されます。
8. [Package IP] ダイアログ ボックスで次を実行します。
 - a. ZIP ファイルの名前が IPwizards_ip_my_complex_mult_3.0.zip であることを確認します (図 3-8)。
 - b. 出力ディレクトリを C:\my_complex_mult に変更します。
 - c. [OK] をクリックします。

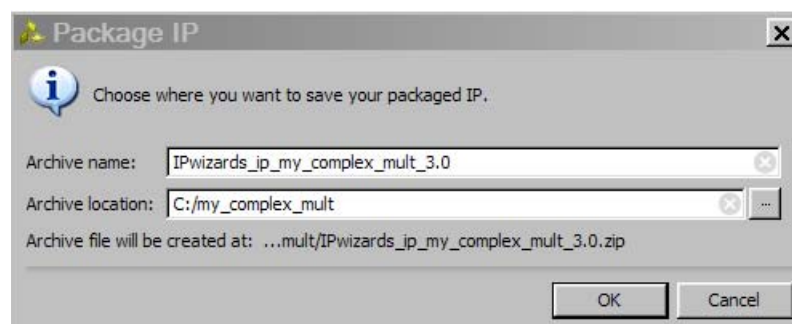


図 3-8 : [Package IP] ダイアログ ボックス

9. C:\my_complex_mult フォルダに新しい ZIP ファイルが追加されているかどうか確認します。

IP カタログへの新規 IP の追加

IP カタログにパッケージ化された IP を追加するには、次の手順に従います。

1. Flow Navigator で [Project Manager] → [IP Catalog] をクリックします。
2. [IP Catalog] で右クリックし、[IP Settings] をクリックします (図 3-9)。

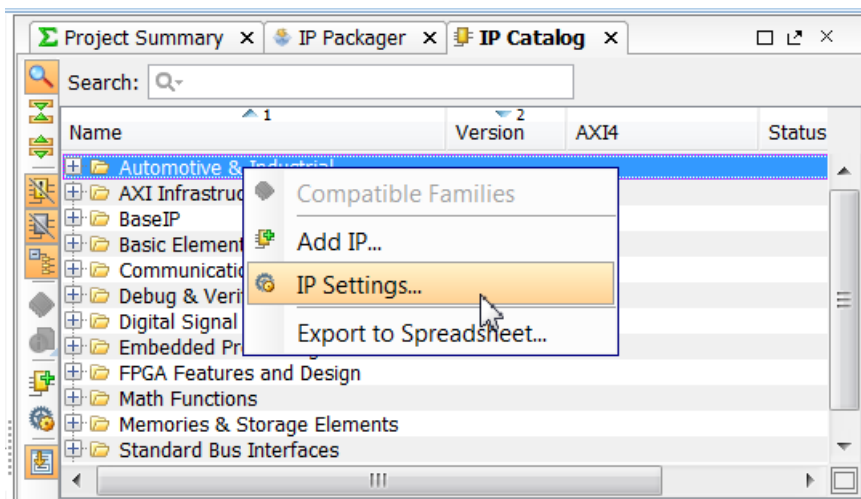


図 3-9 : [IP Settings] コマンド

3. [IP Settings] ダイアログ ボックスで [Add Directories] をクリックして IP リポジトリの検索パスに C:/third_party_ip を選択し、[Apply] をクリックします (図 3-10)。この検索パスには、IP の component.xml ファイルまたはパッケージされた ZIP ファイルが含まれている必要があります。

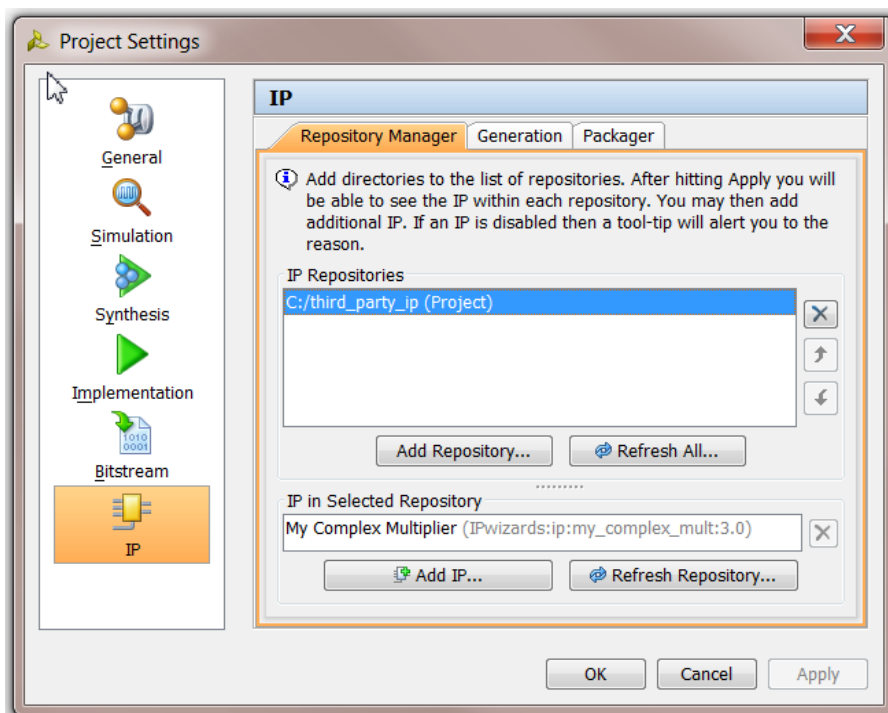


図 3-10 : IP リポジトリの検索パス

4. 追加したユーザー リポジトリを選択し、[Add IP] をクリックし、my_complex_mult の ZIP ファイルを選択して、[OK] をクリックします。
5. [IP Catalog] ビューで [My Complex] フォルダを展開し、My Complex Multiplier という名前の IP が追加されていることを確認します。入力したメタデータが [Details] エリアに表示されることも確認します (図 3-11)。

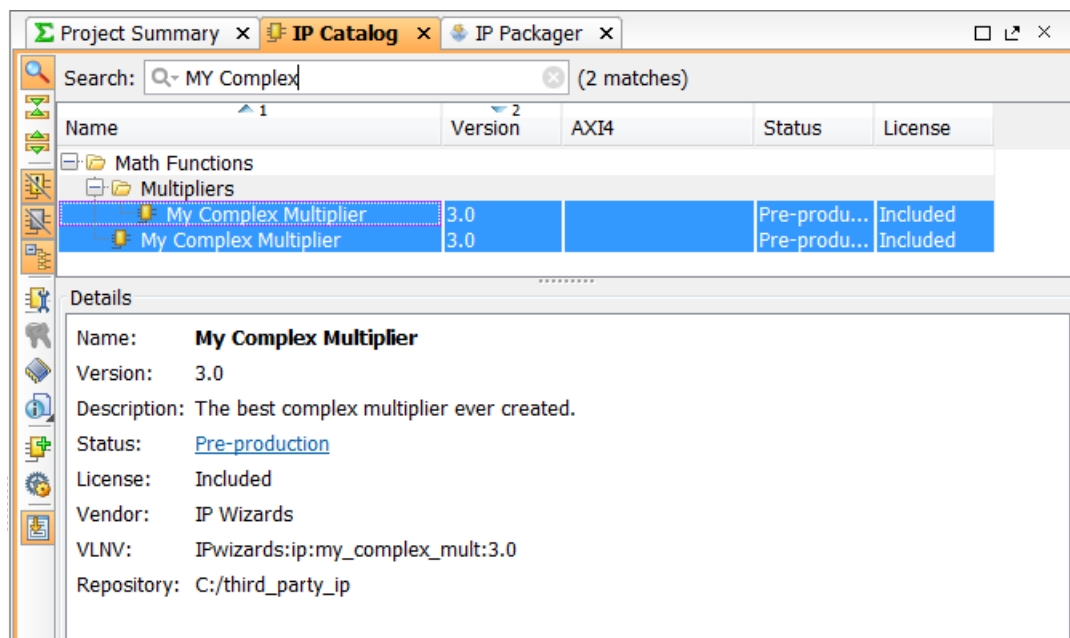


図 3-11 : IP カタログ

HDL 以外のファイルの IP パッケージへの追加

IP パッケージャーでパッケージに HDL 以外のファイルを追加するには、次の手順に従います。

1. [IP Packager] ビューの左側のペインで [IP File Groups] を選択し、右側の [Data Sheet] カテゴリを右クリックして [Add Files (Data Sheet)] をクリックします (図 3-12)。

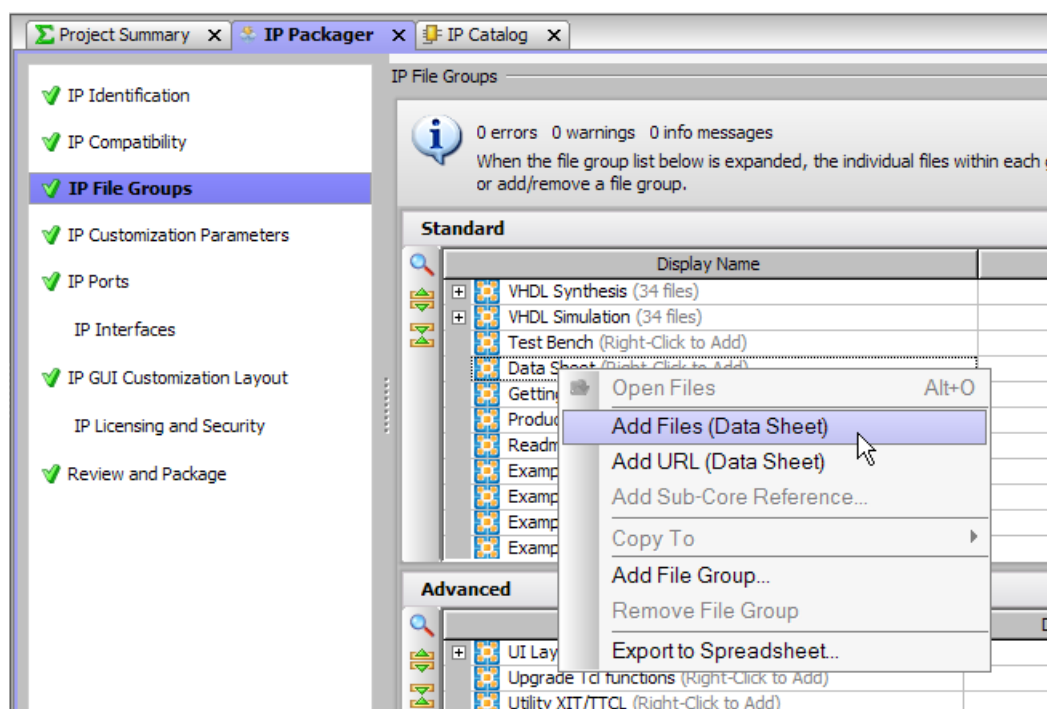


図 3-12 : パッケージへのデータシート ファイルの追加

2. [Add IP Files (Data Sheet)] ダイアログ ボックスで [Add Files] ボタンをクリックし、C:/my_complex_mult/doc ディレクトリを指定し、[Files of type] で [All Files] を選択します。2つの資料ファイルが表示されます。
3. my_complex_mult_data_sheet.pdf ファイルを選択し、[OK] をクリックします。[Add IP Files (Data Sheet)] ダイアログ ボックスで [OK] をクリックします。
4. [Data Sheet (1 file)] カテゴリを展開し、PDF ファイルがパッケージに追加されたことを確認します (図 3-13)。

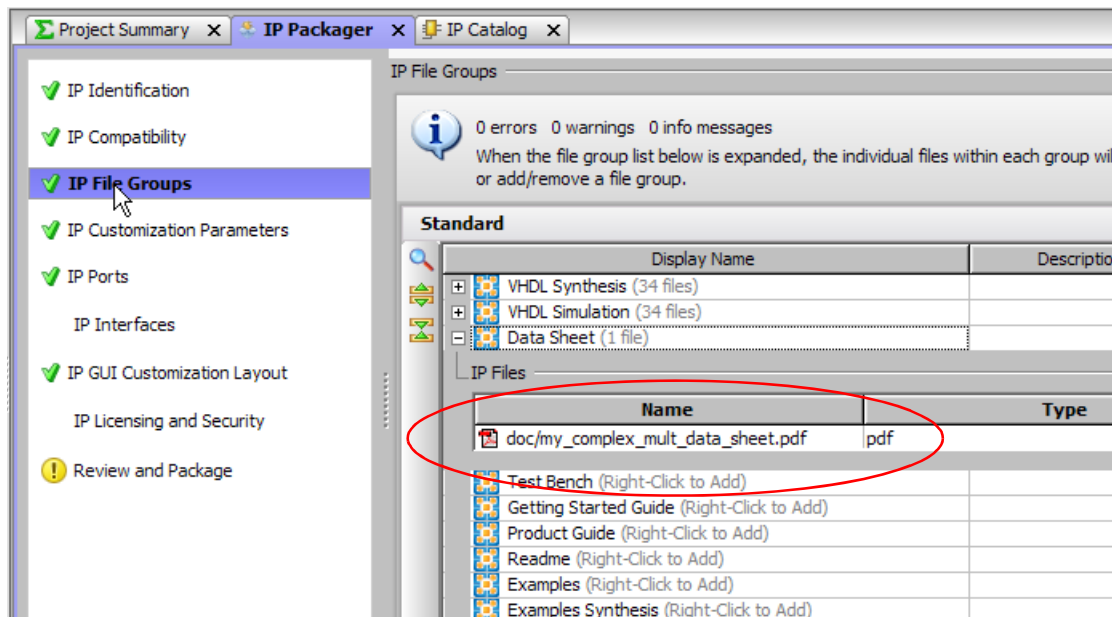


図 3-13 : データシートが追加されたことを確認

5. 手順 1 ～ 4 を繰り返して、[Readme] カテゴリに my_complex_mult_release_notes.txt というファイルを追加します。
6. [Review and Package] をクリックして [Package IP] ボタンをクリックし、追加した文書ファイルを含む IP をパッケージ化します。

以前の EDK IP のインテグレーター コアへの変換

Vivado IP パッケージには、XPS で作成した以前の EDK IP を Vivado IP インテグレーターで互換性のあるコアへ簡単に変換できるフローが含まれています。IP パッケージは以前の EDK IP ディレクトリをスキャンし、検出結果に基づいて必要なフィールドを自動生成します。ユーザーは、通常アドレス範囲の変更、コンポーネント名の付け直し、設定の確認といった最低限必要な動作を実行するだけですみます。

推奨される変換フロー

次は、既存の以前からの EDK IP を Vivado IP インテグレーターのコアに変換するのに推奨されるフローです。

1. 以前の EDK IP ディレクトリを新しいディレクトリにコピーします。
2. このディレクトリから Vivado IDE を起動します。
3. Vivado IDE の Getting Started ページから [Manage IP] をクリックします。
4. パーツおよび言語プリファレンスを選択します (これはデフォルトで表示されますが、オフにすることもできます)。
5. [Tools] → [Package IP] をクリックし、IP パッケージャー ウィザードを起動します。
6. IP パッケージャーで以前の EDK IP リポジトリが自動的にスキャンされます。
7. タブを確認し、次のように変更します。
8. [Review and Package] タブの変更を確認したら、[Archive IP] をクリックします。これにより <componenet_name>.zip が生成されるので、これをユーザー IP リポジトリにコピーします。これで、Vivado IP カタログに追加されます。

IP パッケージャーでは、次のディレクトリを使用してファイル グループを検索します。

ディレクトリ名	ファイルグループ
data/	以前の EDK IP のライブラリおよびファイル情報
src/ hdl/	合成可能なソース
example/ /examples/ ex/	サンプル ソース
simulation/ /sim	シミュレーションのみのモデル
testbench/ /test tb/	テストベンチ
cmodel c/	シミュレーションの C モデル
documents docs/ doc/	文書

src/ または hdl/ ディレクトリが検出されない場合は、見つかった HDL すべてが追加されます。data/*.pao ファイルがある場合は、それが読み込まれ、関連するライブラリ情報がそこから使用されます (これらのファイルは以前の EDK IP で使用されていたものです)。

data/*.mpd ファイルは使用されません。このファイルには、EDK 特有のメタ データが含まれますが、自動的に変換される場合はエラーが発生しやすくなります。IP パッケージャーで生成される IP 定義については、このファイルを参照してください。

ソース ディレクトリの *.ngc、*.edif、*.xdc ファイルが合成ファイル グループに追加されます。

ファイルが追加されると、システムで Vivado IDE プロジェクトが作成され、自動認識機能を使用して最上位ファイルとして設定され、パッケージプロジェクトを使用してパッケージされます。

注記: ファイルで最上位ファイルが複数定義されている場合は、予定しているのとは異なるファイルが最上位として選択されることもあります。このような場合は、Vivado IDE プロジェクトを作成し、ファイルを追加し、予定のファイルを最上位として選択して、[Package Project] を使用します。

IP がパッケージされて IP カタログに含まれるようになれば、その IP を IP カタログで右クリックし、[Edit in IP Packager] をクリックします。または、[File] → [Open] をクリックし、ファイルの種類を [IP-XACT Files] にして、必要な component.xml を参照して、IP を開き直して編集します。

XPS と Vivado Design Suite の動作は異なりますので、以前の IP を変換する際には、次に注意する必要があります。

- 以前の EDK IP にはデュアル Verilog/VHDL インプリメンテーションが含まれるので (混合言語ではなく、重複した言語特有のインプリメンテーションが 2 つあるということ)、どちらか 1 つのインプリメンテーションのみが使用されます。使用されなかった方のインプリメンテーション ファイルは手動で別のファイル グループに追加する必要があること可能性があります。
- HDL 内で XST で生成されたコアを使用すると、それらはチェックされないので、問題が発生する可能性があります。IP パッケージャー アルゴリズムでは、可能な場合はコアのリファレンスを追加しようとはしますが、IP をインスタンスエートしてインプリメンテーションまで実行し、結果を確認する必要があります。
- netlist/ のようなディレクトリの NGC ファイルは自動的に追加されません。
- ライブラリ コアを作成するには、[as library core] オプションを使用します。proc_common のようなライブラリ コアをローカルの EDK プロジェクトに含めて、ファイルを編集することがよくありますが、proc_common の場合は、[as library core] オプションを使用する必要があります (EDK のライブラリ コアは通常 data/ ディレクトリの *.pao ファイルですが、これには対応する *.mpd ファイルはありません)。
- IP パッケージャーで既にインスタンスエート済みの IP 定義を編集すると、問題になることがあります。この場合、一旦 IP を削除してから、特に IP インテグレーターで作成し直すことをお勧めします。
- IP 定義からソース ファイルを編集しても、自動的に認識されません。XST ではファイルが直接使用されますが、Vivado IDE ではこの定義からファイルがプロジェクトにコピーされます。HDL の内容が IP 定義で編集される場合は、IP の出力ファイルを常に生成し直してください。

予約済みパラメーター

次は、Vivado IDE でカスタマイズおよび生成中に自動的に設定される予約済みのパラメーターです。

- c_family
- c_component_name
- c_xdevicefamily
- c_elaboration_dir
- c_elaboration_transient_dir

IP サブシステムの設計

注記：このセクションには、2013.1 リリースで早期アクセス機能としてライセンス付与される新しい Vivado IP インテグレーター環境に関する情報が含まれます。ライセンスの取得については、フィールド アプリケーション エンジニア (FAE) にご連絡ください。

概要

Vivado IP インテグレーターでは、Vivado IP カタログからの IP をデザイン キャンパスにインスタンシエートおよびインターコネクトすることで、複雑なシステムを作成できます。デザインは、IP インテグレーターの GUI を使用したり、Tcl プログラミング インターフェイスを使用して作成できます。デザインは通常インターフェイス レベルで構築して生産性を高めることができますが、ポート レベルで詳細に操作することもできます。

インターフェイスでは機能を共有する信号がグループに分類されます。たとえば、AXI4-Lite マスターには、接続するのに必要な多くの個別の信号と複数のバスが含まれます。各信号またはバスが IP シンボルで個別に表示されると、シンボルは視覚的に大変複雑に見えます。これらの信号やバスをインターフェイスにグループ分けすると、次のような利点があります。まず、IP インテグレーター (または Tcl コマンド) で 1 つ接続をすると、マスターからスレーブへ接続されます。次に、この接続の図がシンプルな 1 つの接続として表示されます。最後に、特定のインターフェイスを検出するデザイン ルール チェック (DRC) が実行され、すべての信号が問題なく接続されているかどうかを確認されます。

IP インテグレーターを使用すると、オートメーション サービスを含めたその Tcl 拡張メカニズムにより、パラメーター伝搬などのシステム デザイン タスクを IP 別やアプリケーション ドメインごとに最適化できます。また、IP インテグレーターのダイナミック ランタイム DRC により、たとえば、IP インテグレーター デザインの IP 間の接続に互換性を持たせ、IP が適切にコンフィギュレーションされるようになります。

プロジェクトの作成

デザイン全体は IP インテグレーターを使用して作成できます。通常デザインには、HDL、IP、および IP インテグレーター ブロック図が含まれます。このセクションでは、新しい IP インテグレーター ベースのデザインを作成する方法を説明します。

Vivado IDE の GUI で [Create New Project] をクリックし、新しいプロジェクトを作成します (図 4-1)。VHDL または Verilog デザイン ファイル、カスタム IP、その他の種類のデザイン ソース ファイルをこのウィザードを使用してプロジェクトに追加します。



図 4-1 : 新規プロジェクトの作成

図 4-2 に示すように、ターゲット デバイスまたはザイリンクス ターゲット ボードを選択することもできます。

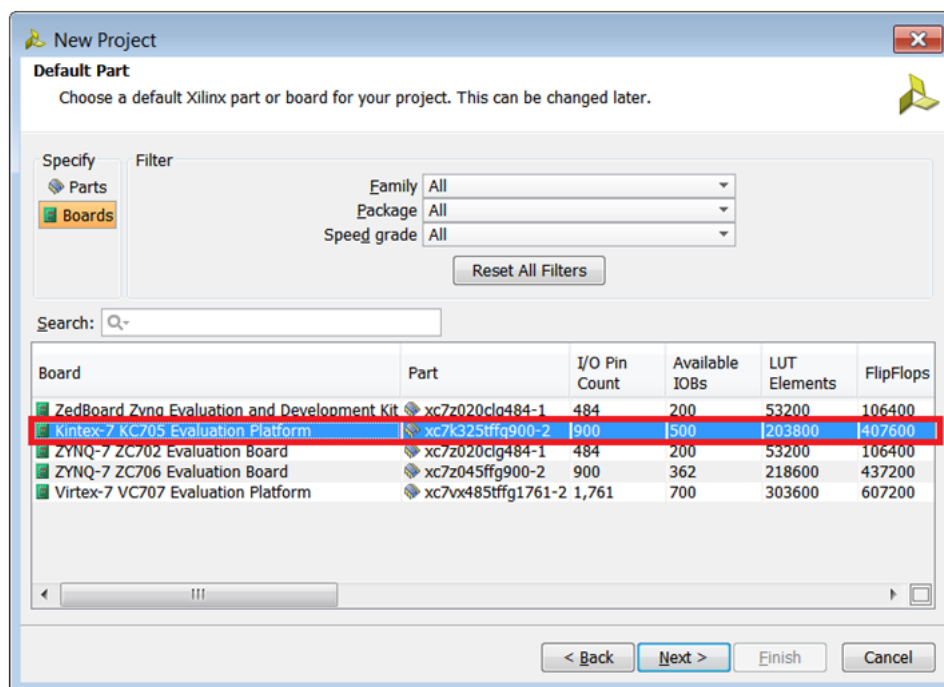


図 4-2 : 新規プロジェクトのターゲット デバイスの選択

注記 : 次の Tcl コマンドを使用しても、同じ動作を実行できます。この文書で Tcl コマンドを表示する際は、デザインに特有のパラメーターを囲むのに <> シンボルが使用されます。<> シンボルは、コマンド文字列には含まれません。

上記と同等の Tcl コマンドは、次のとおりです。

```
create_project xx <your_directory>/xx -part xc7k325tffg900-2
set_property board kc705 [current_project]
set_property target_language VHDL [current_project]
```


IP インテグレーターを使用した設計

IP インテグレーターで [Create Block Design] をクリックし、Flow Navigator で新しいブロック図を作成します。

Tcl でこれに相当するコマンドは次のとおりです。

```
create_bd_design "<your_design_name>"
```

IP インテグレーターの図のサイズ変更

デザインを作成すると、キャンバスが表示されます。このキャンバスを使用してデザインを修正します。このキャンバスの大きさは、Vivado IDE の GUI で変更可能です。この図の右上の [Float Window] ボタンをクリックすると、図を別のモニターに移動できます。図の左上の [Diagram] タブをクリックしても、図のサイズを大きくすることができます。タブをもう 1 度クリックすると、表示がデフォルトのレイアウトに戻ります。

背景色の変更

図の背景色はデフォルトの白から変更できます。図 4-3 に示すように、図の左上の [Block Diagram Options] ボタンをクリックすると、色が変更できます。

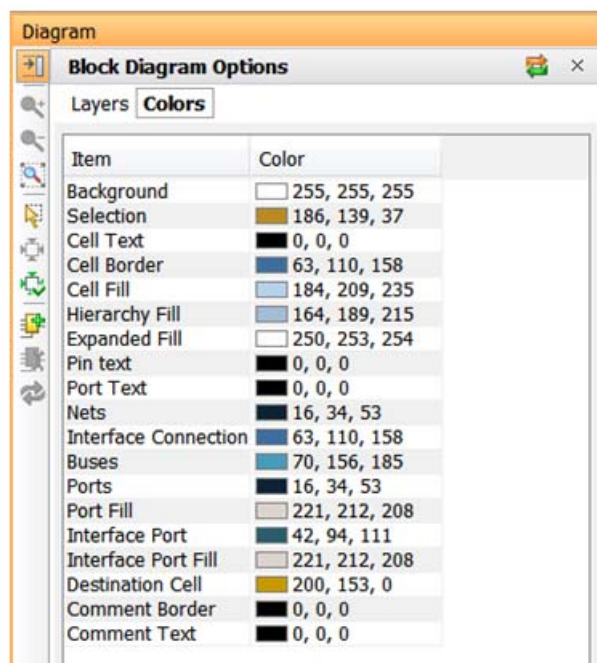


図 4-3 : IP インテグレーターの背景色の変更

IP インテグレーター図で表示されるほとんどすべてのオブジェクトの色は変更できます。たとえば、上記のように背景色を 240,240,240 に変更すると、背景色を薄い灰色にできます。[Block Diagram Options] タブを非表示にするには、右上の閉じるボタンをクリックするか、[Block Diagram Options] ボタンをもう 1 度クリックします。

カーソル動作による表示の変更

IP インテグレーター図では、カーソル動作で表示を変更できます。次の動作を試してみてください。

- 右下から左上 : [Zoom Fit] (画面に図を合わせて表示)
- 右上から左下 : [Zoom In] (拡大)
- 左下から右上 : [Zoom Out] (縮小)
- 左上から右下 : [Zoom Area] (選択したエリアを拡大)

キャンバスの左側のボタンを使用すると、次の動作を指定できます。

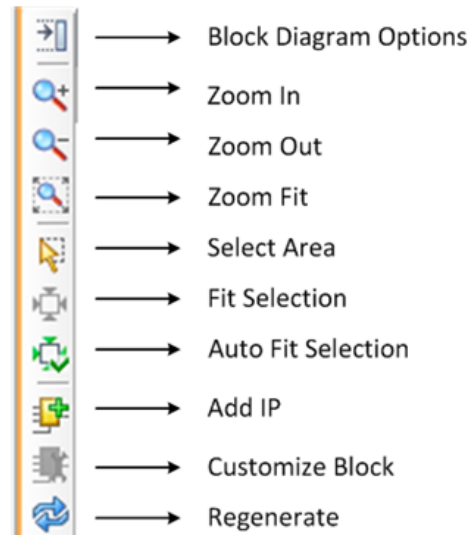


図 4-4 : IP インテグレーターの動作ボタン

デザイン キャンバスへの IP モジュールの追加

IP モジュールは、次の方法で図に追加できます。

1. 図を右クリックし、[Add IP] をクリックします。図 4-5 に示すように、検索可能な IP カタログが開きます。

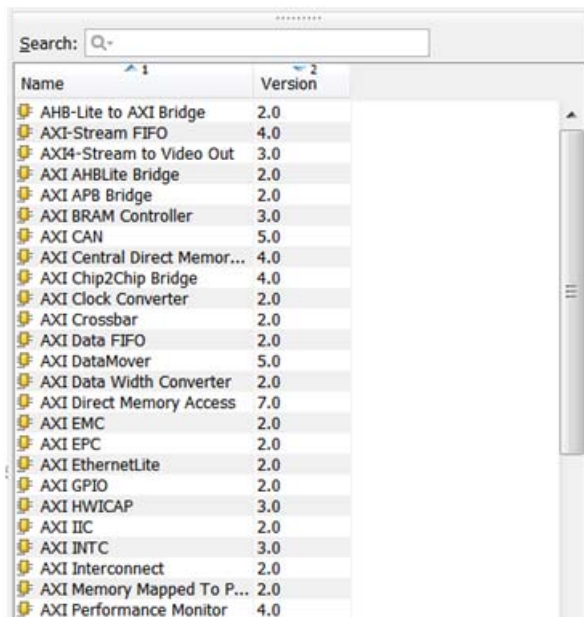


図 4-5 : Vivado IP カタログの起動

IP 名の最初の何文字かを [Search] フィールドに入力すると、それに一致する IP モジュールのみが表示されます (図 4-6)。



図 4-6 : IP カタログの [Search] フィールドの使用

2. 1 つの IP を追加するには、その IP の名前をクリックしてキーボードの Enter ボタンを押すか、その IP 名をダブルクリックします。

3. 複数の IP をキャンバスに追加するには、その IP を Ctrl を押しながらクリックしていき、Enter ボタンを押します。

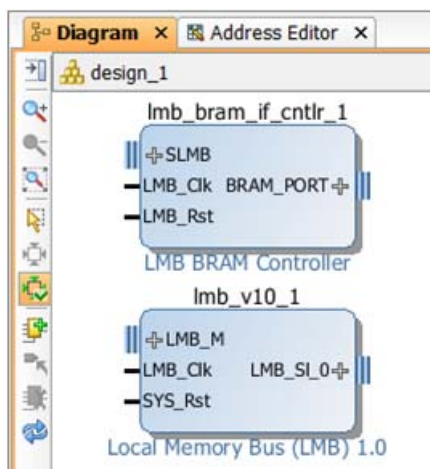


図 4-7: 複数の IP を同時に追加

4. IP は、キャンバス左側の [Add IP] ボタンをクリックしても追加できます。

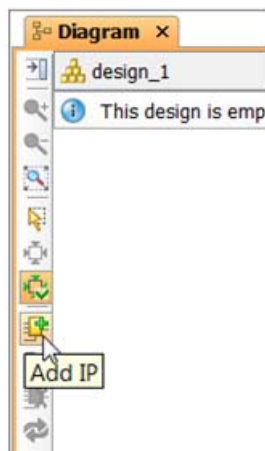


図 4-8: [Add IP] ボタンによる IP の追加

[Add IP] をクリックすると、IP はカーソルの位置の近くに配置されます。

Flow Navigator の Vivado IP カタログからも IP は表示および使用できます。モニターを 2 つ使用する場合は、1 つのモニターで IP カタログを開いておくことができます。モニターを 1 つのみ使用している場合は、IP カタログをフロートさせて、図から離しておくことができます。メインの IP カタログから IP を追加するには、IP カタログから図に選択した IP をドラッグします。

注記: IP をダブルクリックした場合は、Vivado プロジェクトには追加されますが、ブロック図には追加されません。

接続

IP インテグレーターでデザインを作成する場合、図にブロックを追加し、必要に応じてブロックをコンフィギュレーションし、インターフェイスレベルの接続または単純なネットの接続をして、インターフェイスまたは単純なポートを追加します。

IP インテグレーターでの接続は簡単です。カーソルをインターフェイスまたはピンのコネクタの近くに置くと、鉛筆の形に変わりますので、その状態で IP ブロックのインターフェイスまたはピンのコネクタをクリックして、接続するブロックへ線をドラッグします。

図 4-9 に示すように、インターフェイスレベルの接続の場合、シンボルにもっと目立つボックスが表示されます。ボックスで+マークをクリックすると、インターフェイスが展開され、関連する信号およびバスが表示されます。

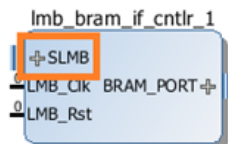


図 4-9 : シンボルの接続ボックス

信号またはバスレベルの接続は、シンボル上で細い接続線として表示されます。接続目的では、バスは各信号と同じように扱われます。図 4-10 に示すように、接続をする際は、接続が可能な接続先の近くに緑のチェックマークが表示されます。

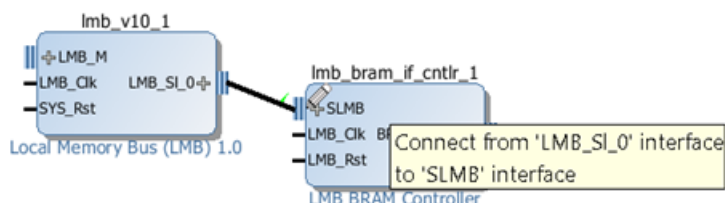


図 4-10 : シンボルの信号またはバスの接続

5. 図 4-11 に示すように、信号がインターフェイスとしてグループになっている場合、個別の信号またはバスを接続する前にまずインターフェイスを展開する必要があります。

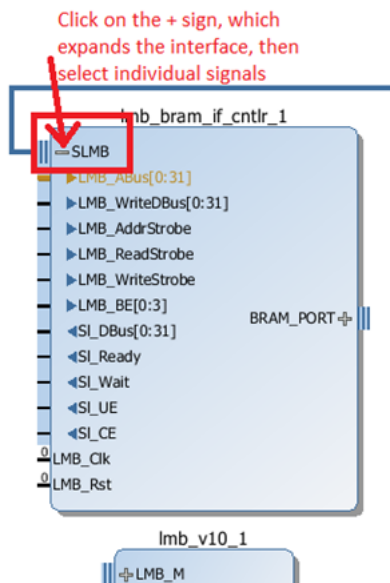


図 4-11 : 接続前のインターフェイスの展開

6. 信号およびインターフェイスを外部 I/O ポートに接続するには、次の 3 つの方法があります。
 - a. 図 4-12 に示すように、信号またはインターフェイスを図の外部ポートに接続するには、まずピン、バスまたはインターフェイス接続を選択します。次に右クリックで [Make External] を選択します。Ctrl ボタンを押

しながらピンをクリックして [Make External] をクリックすると、複数のピンを外部 I/O ポートに接続できます。

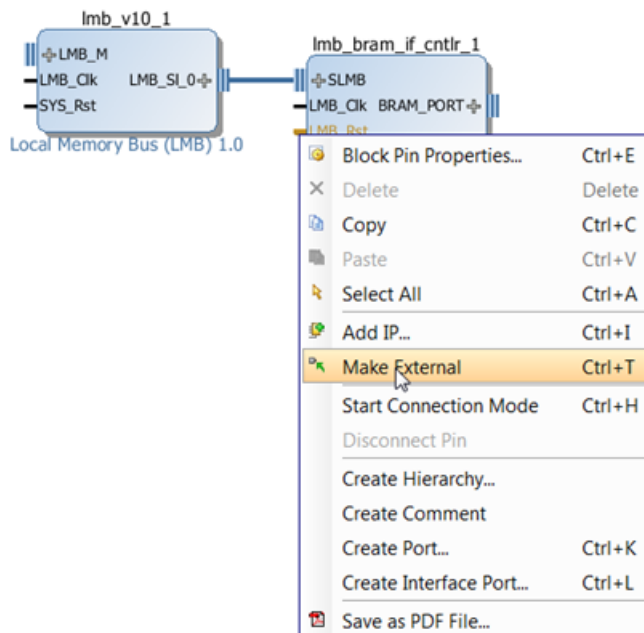


図 4-12 : 外部接続

このコマンドを使用すると、IP のピンをブロック図の I/O ポートに接続できます。IP インテグレーターでは、このように IP のポートを外部 I/O に簡単に接続できます。

- b. 2 つ目の接続方法では、右クリックで [Create Port] をクリックします (図 4-13)。この方法は、クロック、リセット、uart_txd などのインターフェイス以外の信号を接続するのに使用します。[Create Port] を使用すると、入力/出力、ビット幅、およびタイプ (clk、reset、data) などの指定がより詳細に制御できます。クロックの場合は、入力周波数を指定することすら可能です。

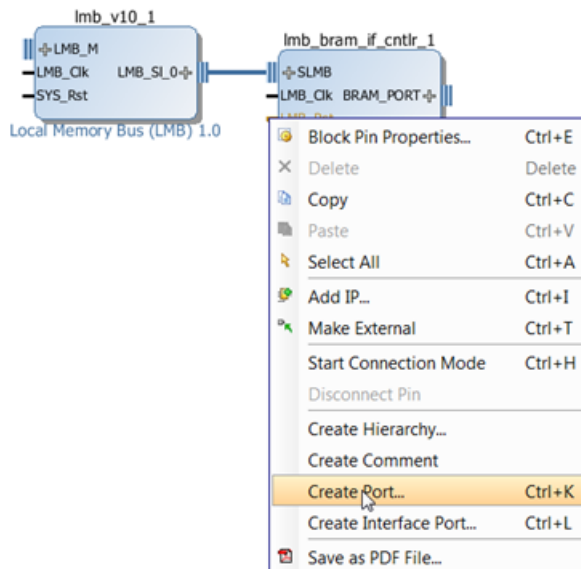


図 4-13 : ポートの作成

- c. 3つ目の接続方法では、右クリックで [Create Interface Port] をクリックします (図 4-13)。

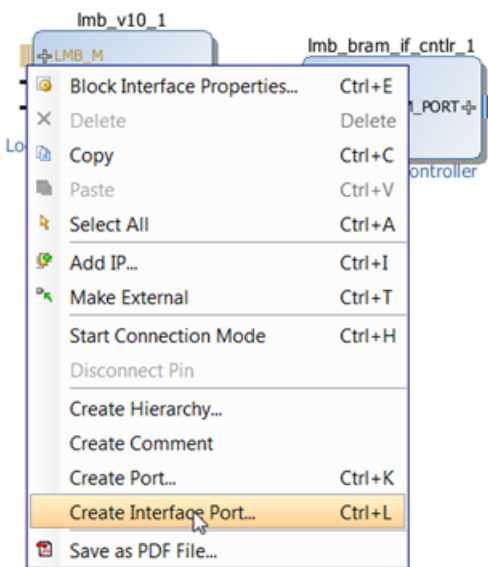


図 4-14 : インターフェイス ポートの作成

このコマンドは、共通の関数を共有する信号をグループにしたインターフェイスにポートを作成するために使用します。たとえば、S_AXI は複数のザイリンクス IP のインターフェイス ポートです。このコマンドを使用すると、インターフェイス タイプおよびモード (マスター/スレーブ) を指定する際には、より詳細に制御できます。

IP インテグレーターのブロックオートメーションとコネクションオートメーション機能

IP インテグレーターにはブロックオートメーションおよびコネクションオートメーション機能が含まれ、自動的に基本的なマイクロプロセッサシステムがまとめられたり、外部 I/O ポートにポートが接続されたりします。ブロックオートメーション機能は、Zynq プロセッシングシステム 7 や MicroBlaze プロセッサのようなマイクロプロセッサが IP インテグレーターのブロック図にインスタンス化される場合に提供されます。図 4-15 のように [Run Block Automation] をクリックすると、シンプルな MicroBlaze システムを構築するアシスタンスが表示されます。

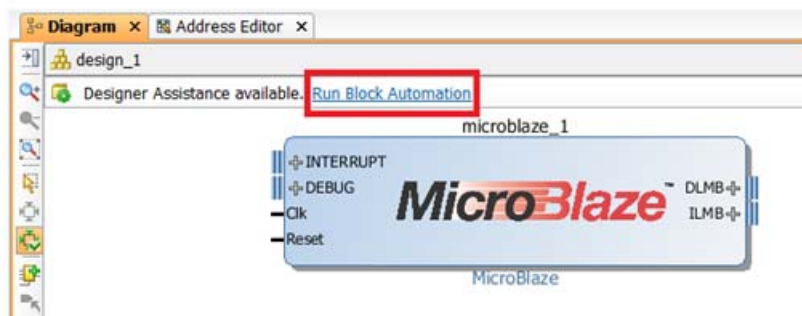


図 4-15 : ブロックオートメーション機能

[Run Block Automation] ダイアログ ボックスでは、マイクロプロセッサ システムに必要な基本的機能を入力できます。



図 4-16 : [Run Block Automation] ダイアログ ボックス

必要なオプションを指定したら、ブロック オートメーション機能により基本的なシステムが自動的に作成されます (図 4-17)。

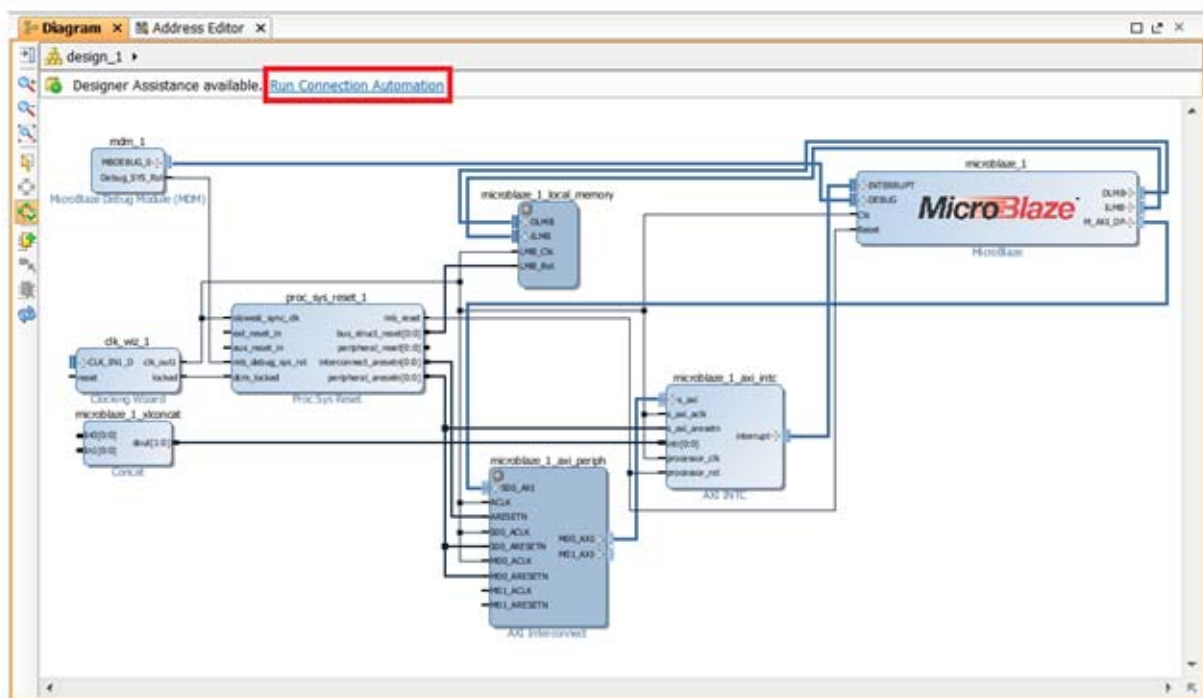


図 4-17 : ブロック オートメーション機能で作成された基本的なシステム

この場合、MicroBlaze デバッグ モジュール、ローカル メモリ バスを含む `microblaze_1_local_memory` という階層ブロック、ローカル メモリ バス コントローラー、ブロック メモリ ジェネレーター、クロッキング ウィザード、AXI インターコネクトおよび AXI 割り込みコントローラーが含まれます。デザインはこの段階では外部 I/O に接続されていないので、上の図で赤枠で囲まれている IP インテグレーターのコネクション オートメーション機能を使用できます。

[Run Connection Automation] をクリックすると、インターフェイスまたはポートを外部 I/O ポートに接続するためのアシスタンスが表示されます、

図 4-18 に示すように、コネクション オート メーションを使用可能なポート/インターフェイスがリストされます。

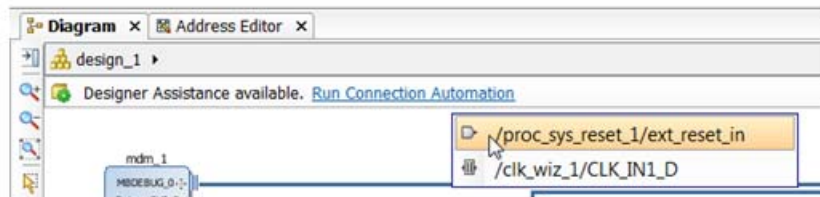


図 4-18: コネクション オート メーション機能

ザイリンクス ターゲット リファレンス プラットフォームまたは評価ボードの場合、ターゲット ボードで接続または使用される FPGA ピンはわかっていますので、その情報に基づいて、IPI コネクション オート メーション機能ではデザイン内のポートと外部ポートの接続がガイドされるようになっています。この後、IPI により最適な物理制約およびその I/O ポートに必要なその他の I/O 制約が作成されます。上記のデザインでは、Proc Sys Reset IP は外部リセットポートに、クロッキング ウィザードは外部クロックソースに接続する必要があります。上記で /proc_sys_reset_1/ext_reset_in オプションをオンにすると、図 4-19 のようなダイアログ ボックスが表示されます。

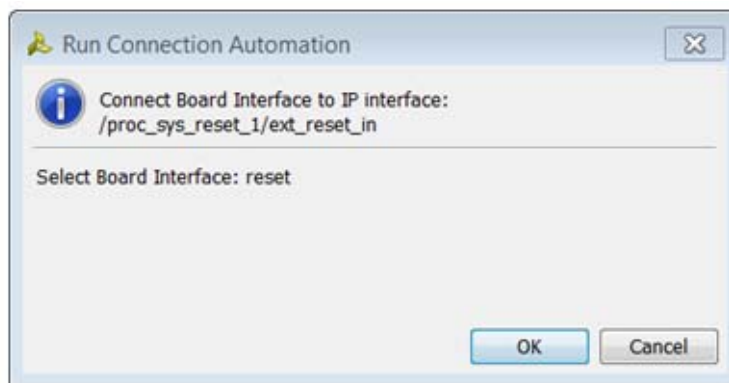


図 4-19 : [Run Connection Automation] ダイアログ ボックス

この場合、ターゲット ボードとして K705 が選択されているので、Proc Sys Reset IP の ext_reset_in は FPGA のリセットピンに接続できることが示されます。[OK] をクリックすると、ブロック図で次のように接続されます。

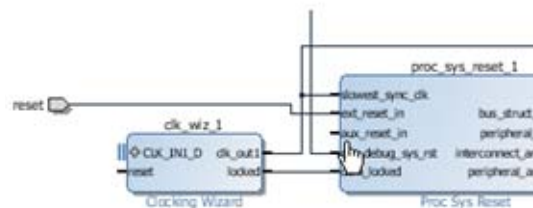
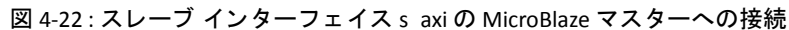


図 4-20: コネクション オート メーション機能による接続

The diagram illustrates a MicroBlaze system configuration. Key components include:

- mdm_1**: MicroBlaze Debug Module (MDM) connected to the **microblaze_1** **DEBUG** port.
- proc_sys_reset_1**: Process System Reset block connected to various reset signals of the **microblaze_1** and **axi_gpio_1**.
- microblaze_1 local memory**: Local memory block connected to the **microblaze_1** **LOCAL_MEMORY** port.
- microblaze_1**: The main MicroBlaze processor core.
- axi_gpio_1**: AXI GPIO peripheral block, highlighted with a red box, connected to the **microblaze_1** **AXI** port and the **proc_sys_reset_1** reset signals.
- axi_gpio_1** is also connected to the **axi_gpio_1** **gpio** port.

[Run Connection Automation] をクリックすると、GPIO のスレーブ AXI ポートが MicroBlaze のマスターに接続可能であることがダイアログ ボックスに表示されます。デザインにマスターが複数含まれる場合は、マスターを選択できます。



[Run Connection Automation] ダイアログ ボックスで [OK] をクリックすると、図の赤枠で囲まれた部分のように接続されます。

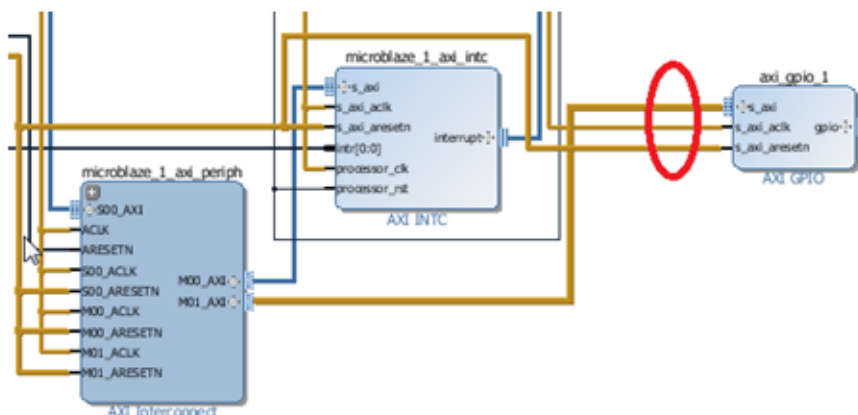


図 4-23 : マスター /スレーブの接続部分

ザイリンクス ターゲット リファレンス プラットフォームを使用する場合は、I/O ポートに関してさらに詳細なアシスタンスが表示されます。図 4-24 を参照してください。

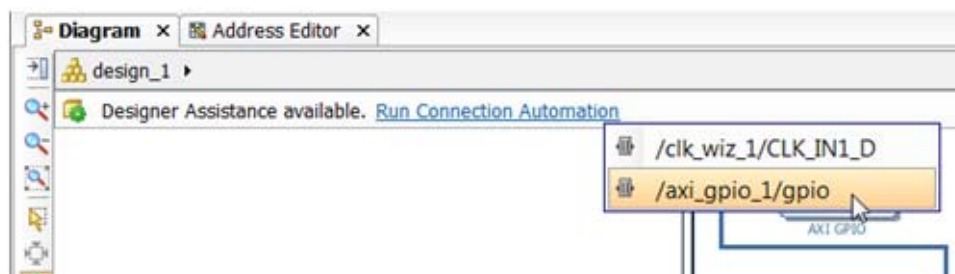


図 4-24 : AXI GPIOポートの選択

gpio ポートを選択すると、ボードに関する情報に基づいて次のようなオプションが表示されます。

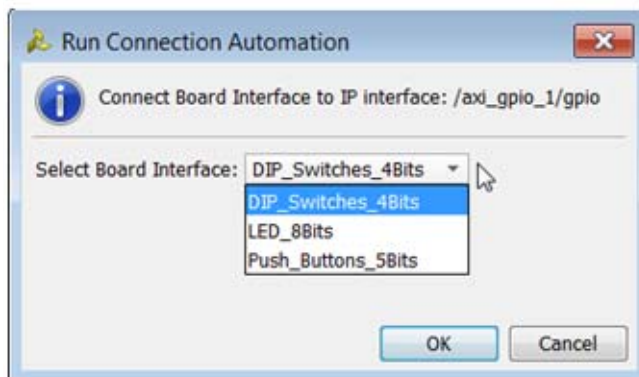


図 4-25 : ボード インターフェイスの選択

この場合、選択肢が3つ表示されます。**gpio** ポートは4ビットの DIP スイッチ、8ビットの LED、5ビットプッシュボタンのいずれかに接続できます。いずれかを選択すると、**gpio** ポートがそのボードの既存の接続に接続されます。

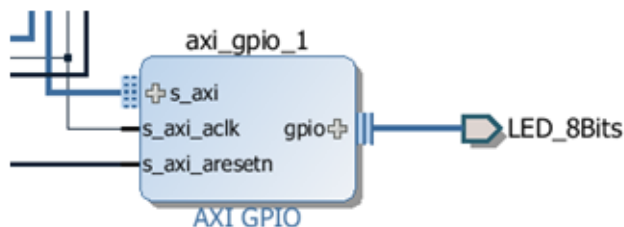


図 4-26 : ボード I/O エレメントのデザインへの接続

IP ブロックの並び替え

キャンバス上の IP ブロックは、ブロック図を見やすくするように並び替えることができます。完了した図や作業中の図を並び替える場合は、[Regenerate] ボタンをクリックします。

ブロックをドラッグするか、矢印キーを使用すると、ブロックを手動で移動できます。移動できる箇所にカーソルを移動すると、濃い灰色の縦のバーが表示されます。ブロックを移動する際は、グリッドが表示されるので、ブロックおよびピンを揃えやすくなっています。

コピーと貼り付け

図でブロックをコピーして張り付けるには、Ctrl + C と Ctrl + V を使用できます。

階層の作成

図 4-27 のように、Ctrl を押しながら IP ブロックをクリックして選択し、右クリックで [Create Hierarchy] を選択すると、図に階層ブロックを作成できます。

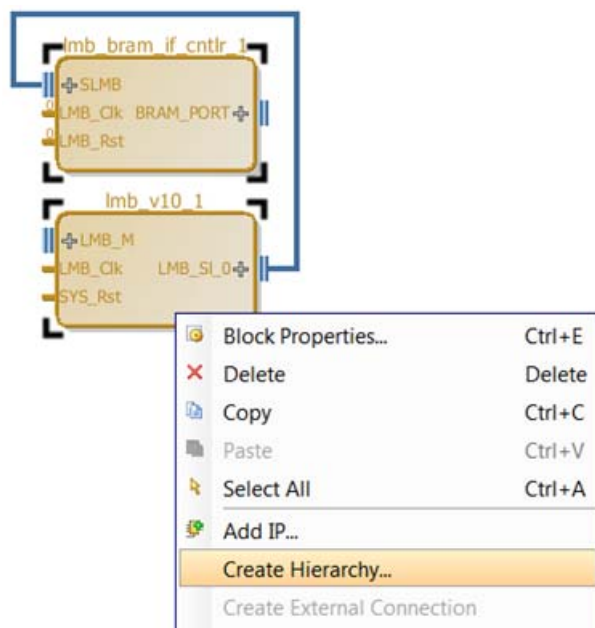


図 4-27 : 階層ブロック図の作成

選択したブロックを含んだ新しいレベルの階層が作成されます。空の階層を作っておいて、後で既存の IP ブロックをその階層ブロックにドラッグすることもできます。階層は、ブロックの左上の + マークをクリックすると展開できます。図の階層レベルは、IP インテグレーターの図の左上に表示されるエクスプローラー タイプ パス情報を使用して入れ替えることができます。[Ungroup Instances] を使用すると、IP ブロックを階層グループから解除できます。

[Create Hierarchy] をクリックすると、[Create Hierarchy] ダイアログ ボックス (図 4-28) が開き、階層名を指定できます。

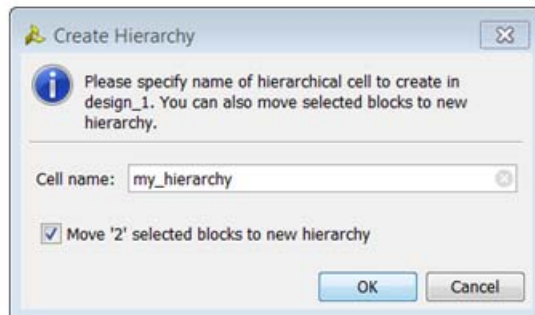


図 4-28 : [Create Hierarchy] ダイアログ ボックス

これにより、図 4-29 のように 2 つのブロックが 1 つのブロックの下にまとめられます。階層の + マークをクリックすると、その下のコンポーネントを表示できます。展開した階層で - マークをクリックすると、元のグループ表示に戻ります。

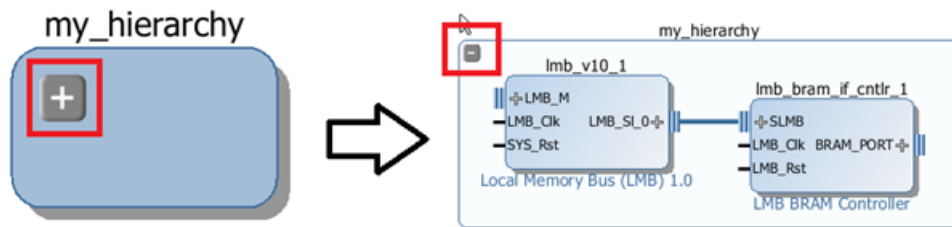


図 4-29 : 階層の展開

メモリ マップの作成

この設計のアドレス マップを生成するには、図の上部の [Address Editor] をクリックします。左下の [Auto Assign Address] をクリックしても生成できます。IP インテグレーターの図から最初にアドレスを生成せずに RTL を生成すると、自動的にアドレスが割り当てられるように選択できるプロンプトが表示されます。アドレスは [Offset Address] および [Range columns] に値を入力して手動で設定することもできます。

注記 : [Address Editor] タブは、図にバス マスターとして機能する IP ブロック (次の図の場合は MicroBlaze) が含まれる場合にのみ表示されます。

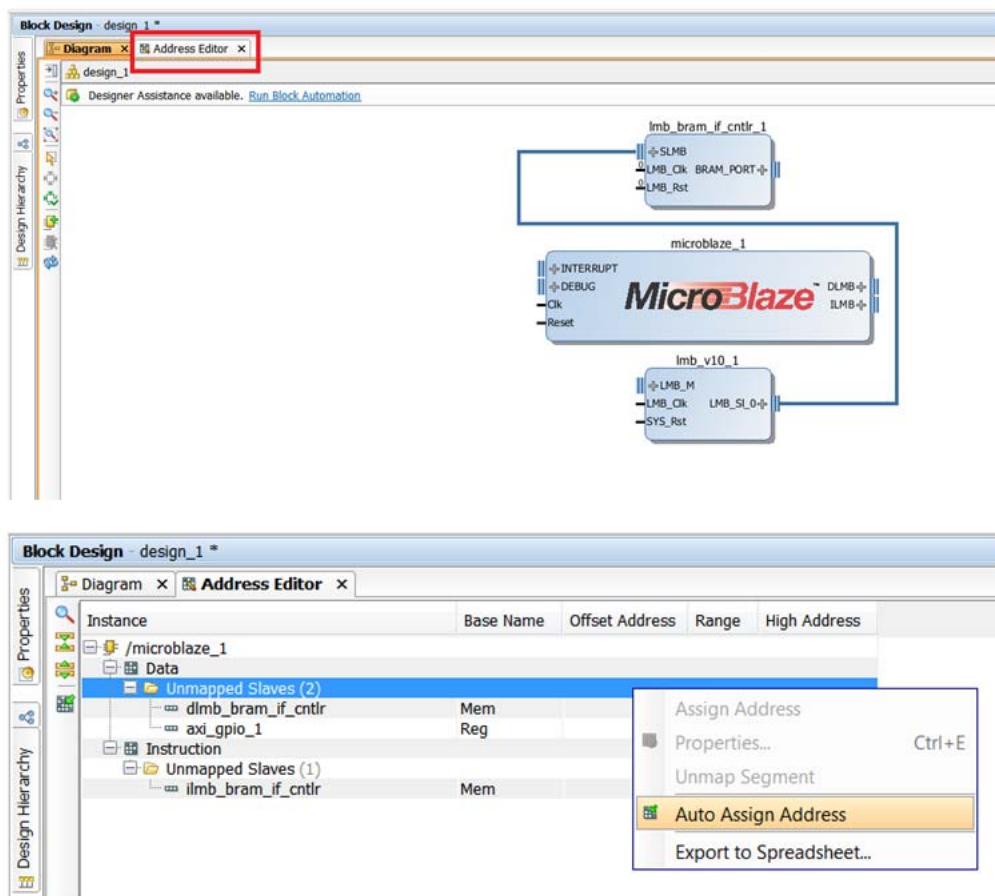


図 4-30 : [Address Editor] タブの使用

デザイン ルール チェックの実行

IP インテグレーターでは、デザインがまとめられる際にリアルタイムで基本的なデザイン ルール チェックが実行されますが、デザイン作成中に何かしらの問題になることもあります。たとえば、クロック ピンの周波数が正しく設定されていない場合などです。図 4-31 に示すように、[Validate Design] をクリックすると全体的なデザイン チェックを実行できます。

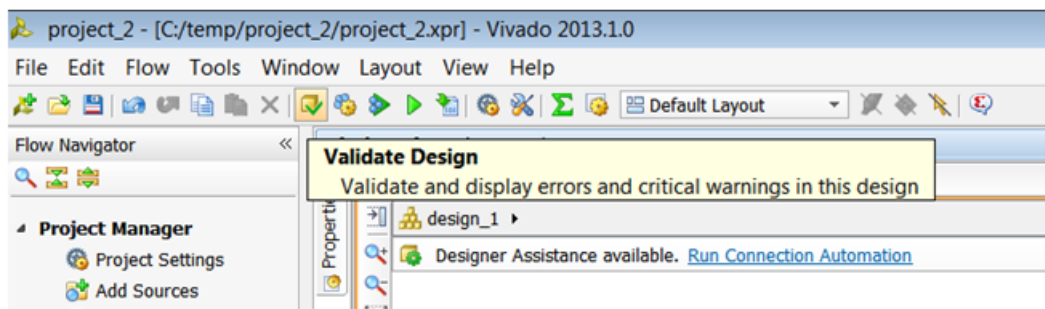


図 4-31 : デザインの検証

[Validate Design] を実行した後に警告やエラーのない場合は、図 4-32 のようなメッセージが表示されます。

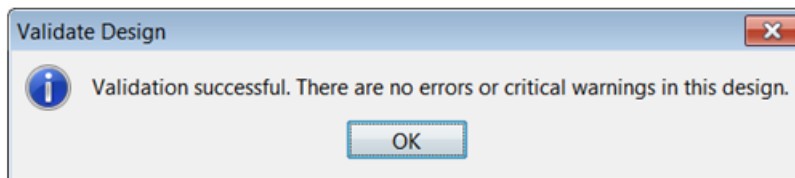


図 4-32 : 問題のなかったことを示すメッセージ

ブロック図の最上位デザインへの統合

ブロック図が完了してデザインを検証したら、次の 2 つの手順を実行してデザインを終了します。まず、出力ファイルを作成します。これにより、ソース ファイルとすべての IP に対する最適な制約が生成され、Vivado の [Sources] ビューで使用できるようになります。プロジェクト作成時に選択したターゲット言語によって、最適なファイルが生成されます。特定の IP のソース ファイルが指定したターゲット言語では生成されない場合は、それを示すメッセージがコンソールに表示されます。出力ファイルを作成するには、Vivado の [Sources] ビューでブロック図を右クリックし、[Generate Output Products] をクリックします (図 4-33)。

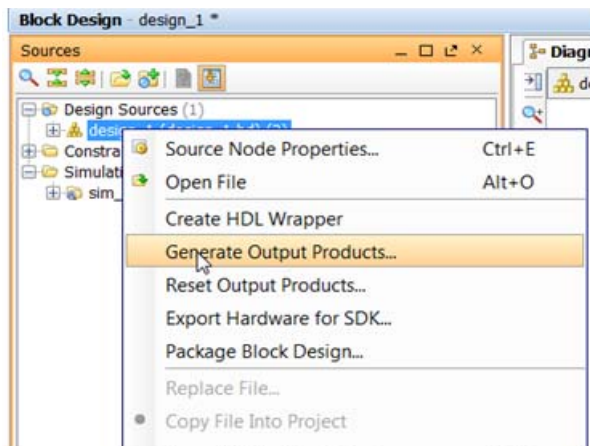


図 4-33 : 出力ファイルの生成

IP インテグレーターのブロック図はそれより上のレベルに統合したり、デザイン階層の一番上の階層にできます。IP インテグレーター デザインをより上の階層に統合するには、単にそのデザインを最上位 HDL ファイルにインスタンス化します。

ブロック図を Vivado の [Sources] ビューで選択して [Create HDL Wrapper] をクリックしても、ブロック図をより上の階層にインスタンス化できます (図 4-34)。これにより、IP インテグレーター (サブシステム) の最上位 HDL ファイルが生成されます。この段階までで、デザインはエラボレーション、合成、およびインプリメンテーションまで実行できる準備が整いました。

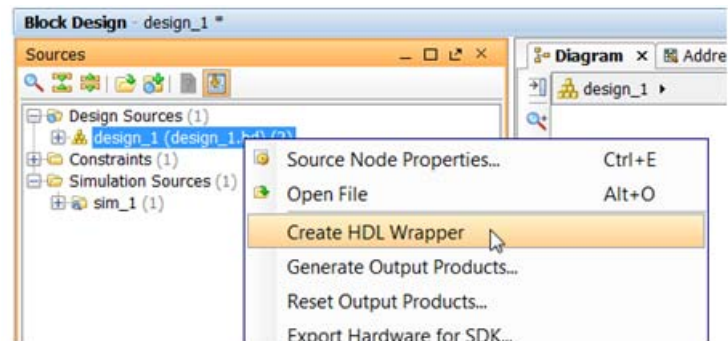


図 4-34 : HDL ラッパーの作成

IP の特性化と Fmax マージン システムの手 法

概要

この付録では、ザイリンクスで使用されるシステム デザイン内の IP 操作の最大周波数 (Fmax) の決定方法について説明します。この方法を使用すると、ザイリンクス FPGA アーキテクチャに対する現実的なパフォーマンスがレポートされるようになります。デザインの最大周波数とは、機能的な問題なしにデザイン全体を実行可能な最大周波数です。より複雑なシステムを FPGA にインプリメントするという現在のトレンドでは、IP にそれぞれ含まれる Fmax 周波数が重要な要素となります。

スタンドアロン IP の特性化手法

IP の実行速度を決定するためには、複数の要因を考慮する必要があります。最初の最も重要な要因は、IP の HDL コーディング形式です。たとえば、IP デザインをパイプライン処理してより高い周波数で実行することで、より優れたパフォーマンスを達成することができます。また、周波数に影響するリソースを共有するため、マイクロ アーキテクチャ レベルで発生するトレードオフもあります。全体のパフォーマンスを決定するもう 1 つの要因には、インプリメンテーション ツールで使用するオプション設定があります。適切なオプションを選択することで、IP 設計者は必要なデザイン目標を達成できます。デザインには、エリア (最小のリソース使用量) またはパフォーマンス (最大の達成可能周波数) 目的に制約を付けることができます。デザイン ツールのオプションによって、合成およびインプリメントをさらに詳細に制御できます。たとえば、ロジックの複製、レジスタのリタイミング、リソースの共有を選択したり、適切な制約を付けることで、より優れたパフォーマンスを達成することができます。

スタンドアロン IP の特性化の場合、ザイリンクスではデフォルトのデザイン ツール設定を使用します。図 A-1 に示すように、この手法では、スタンドアロン IP デザイン (Design Under Test) がインスタンス化され、スキャン レジスタ ラッパーにデザインをラッピングして IO の使用を制限します。ラッパー ロジックは、IO の使用を抑えるだけでなく、デザインが大型デバイスに効率的にマップされるようにもします。また、デザインが大型デバイス内に広がっても問題ないように、

IP の使用準備ができているようにもします。

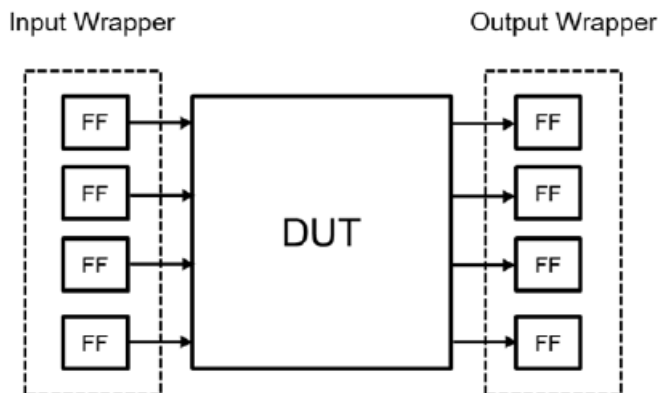


図 A-1 : スタンドアロン IP の特性化

スタンドアロン IP の特性化手法を使用すると、デフォルトのシステム制約だけを使用して IP が実行可能な最大周波数が決定できます。これはスタンドアロン IP なので、配置および配線リソースの競合するリソース制約はありません。また、IO 制約およびラッパーがないので、IO がフリップフロップで直接駆動、またはラッパー内でサンプリングされます。

注記 : スタンドアロン IP のパフォーマンス数は、ザイリンクス ツールを使用し、合成、配置、配線を含んだデザインの完全なインプリメンテーション run によって決まります。

Fmax マージン システムの手法

スタンドアロン IP の特性化では IP パフォーマンスの最大リミットが決定されますが、ユーザー システムで IP パフォーマンスを決定することも重要です。エンベデッド システムの場合、システムには通常次の項目が含まれます。

- MicroBlaze プロセッサまたは Processor System 7 (PS7)
- キャッシュ (IC および DC)
- 1 または複数レベルのインターコネクト
- メモリ コントローラー (MIG)
- Direct Memory Access (DMA) コントローラー
- オンチップ BRAM コントローラー
- ペリフェラル (UART、タイマー、GPIO、割り込みコントローラーなど)
- テスト中の IP (DUT など)

これらのコンポーネントも含めてエンベデッド IP の Fmax を決定することで、より現実的なパフォーマンス ターゲットを取得できます。また、ユーザー システムは、最大 70 ～ 80% のロジックでデバイスを埋めることができます。Fmax マージン システム手法では、基本的なエンベデッド システムに IP を加えて、デバイスの残りの部分を使用可能な LUT、BRAM、IO で埋めて、デバイスの 70% ～ 80% がフルになるようにされます。

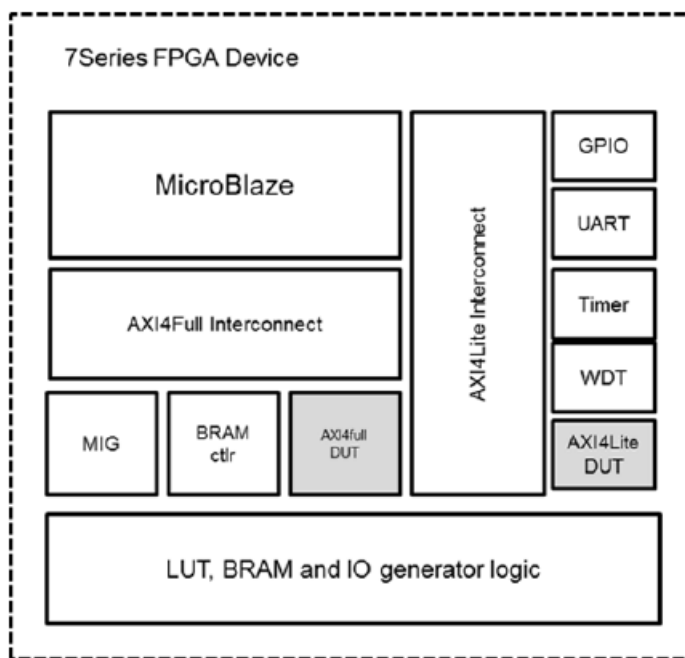


図 A-2 : 7 シリーズの Fmax マージン システム

図 A-2 に示すように、エンベデッド システムには次の 2 種類の AXI インターコネクトが含まれます。

- **AXI4-Lite** : ペリフェラル コマンドおよび制御に通常使用されます。このインターコネクトは一般的にはもっと低い周波数で実行され、エリアを最小限にするよう設計されています。
- **AXI4** : 通常、高馬力のデータ動作タイプのアプリケーションに使用されます。このインターコネクトは一般的には最適な周波数で実行され、パフォーマンスを最大にするよう設計されています。

Fmax マージン システム解析では、AXI4-Lite インターコネクトのクロック周波数は 150MHz に固定されます。AXI4 インターコネクトおよび残りのロジックは 150MHz からシステムがタイミング違反になる最大周波数 (ワースト ケース ネガティブ スラック) まで増加していきます。AXI4 が実行される最大周波数によって、システム全体の Fmax が決まります。

図 A-3 は、Zynq の Fmax マージン システムを示しています。

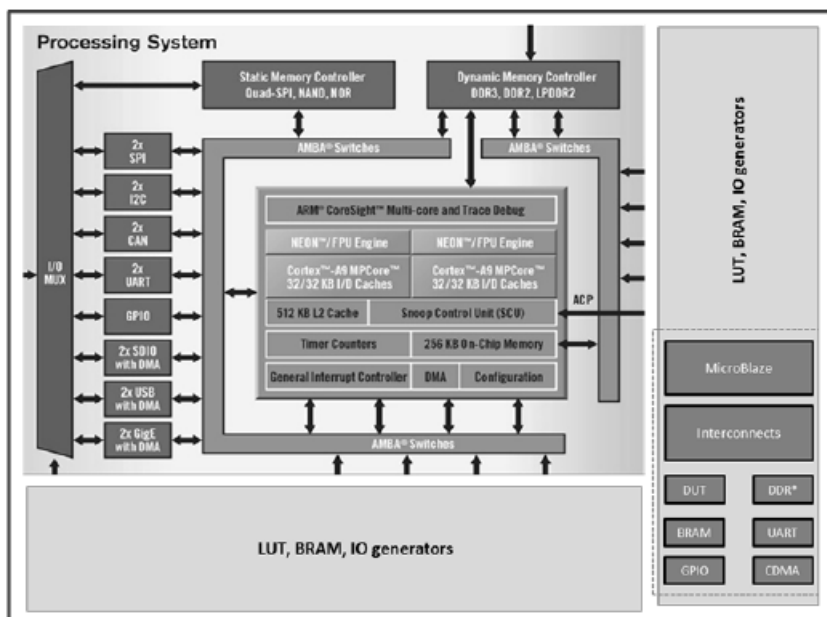


図 A-3 : Zynq の Fmax マージン システム

ツール オプションとその他の要因

ザイリンクス ツールには、デザイン パフォーマンス、リソース使用量、インプリメンテーション実行時間、メモリ フットプリント間の調整する多くのオプションおよび設定があります。あるデザインに最適な結果をもたらす設定が別のデザインでも最適な結果になるとは限りません。

Fmax マージン システム解析の目的は、クロッキング制約以外の特定の制約なしに、IP デザインをデフォルトの設定で特性化することにあります。クロッキング制約は、150MHz からシステムがタイミング違反(ワースト ケース ネガティブ スラック)になるまで徐々に増加します。この解析は、異なる FPGA アーキテクチャおよびスピード グレードを使用して実行されます。結果は IP コアの IP 製品ガイドにそれぞれレポートされています。

その他のリソース

1. 『ザイリンクス デザイン ツール : リリース ノート ガイド』(UG631)
2. 『ザイリンクス デザイン ツール : インストールおよびライセンス ガイド』(UG798)
3. 『Vivado Design Suite ユーザー ガイド : Vivado IDE の使用』(UG893)
4. 『Vivado Design Suite Tcl コマンド リファレンス ガイド』(UG835)