

Vivado Design Suite ユーザー ガイド

IP インテグレーターを使用した IP サブシステム
の設計

UG994 (v2013.2) 2013 年 6 月 19 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 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 (v2013.2) 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。

日付	変更点
2013/06/19	初版

目次

IP インテグレーターを使用した IP サブシステムの設計.....	1
IP サブシステムの設計.....	4
概要.....	4
プロジェクトの作成.....	4
IP インテグレーターを使用した設計.....	7
ハードウェア定義の SDK へのエクスポート.....	29
ブロック図のパッケージ.....	31
ELF ファイルのエンベデッド デザインへの追加および関連付け.....	34

IP サブシステムの設計

概要

Vivado IP インテグレーター機能を使用すると、Vivado IP カタログからの IP をデザイン キャンバス上でインスタンス化およびインターコネクトして、複雑なデザインを作成できます。デザインは、IP インテグレーター キャンバスの GUI を使用してインタラクティブに作成できるほか、Tcl プログラミング インターフェイスを使用しても作成できます。デザインは通常インターフェイス レベルで構築できますが（生産性向上目的）、ポート レベルでも操作できます（デザイン操作の精度向上目的）。

インターフェイスは、よくあるファンクションを共有する信号のグループで、たとえば、AXI4-Lite マスターには多くの信号と複数のバスが含まれ、これらはすべて接続に必要です。各信号またはバスが IP シンボル上で個別に表示されると、シンボルが複雑に見えます。これらの信号およびバスをインターフェイスにまとめると、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 つ接続すると、マスターからスレーブへの接続になり、この接続のグラフィック表示は単純な 1 つの接続になります。最後に、特定インターフェイスを認識するデザイン ルール チェック (DRC) が実行され、必要な信号すべてが正し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利点は、パラメーター伝搬などのシステム デザイン タスクが IP ごとまたはアプリケーションドメインごとに最適化できるように、自動化サービスの Tcl 展開機能が備わっている点にあります。IP インテグレーターではダイナミックなランタイム DRC を実行することで、IP インテグレーター デザインの IP 間の接続が互換性を持ち、IP 自体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が、典型的なデザインには 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 ベースのデザインの作成方法の概要を示します。

次の図に示すように、Vivado の GUI で [Create New Project] をクリックして、新規プロジェクトを作成します。このウィザードで 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 ファイルを追加します。

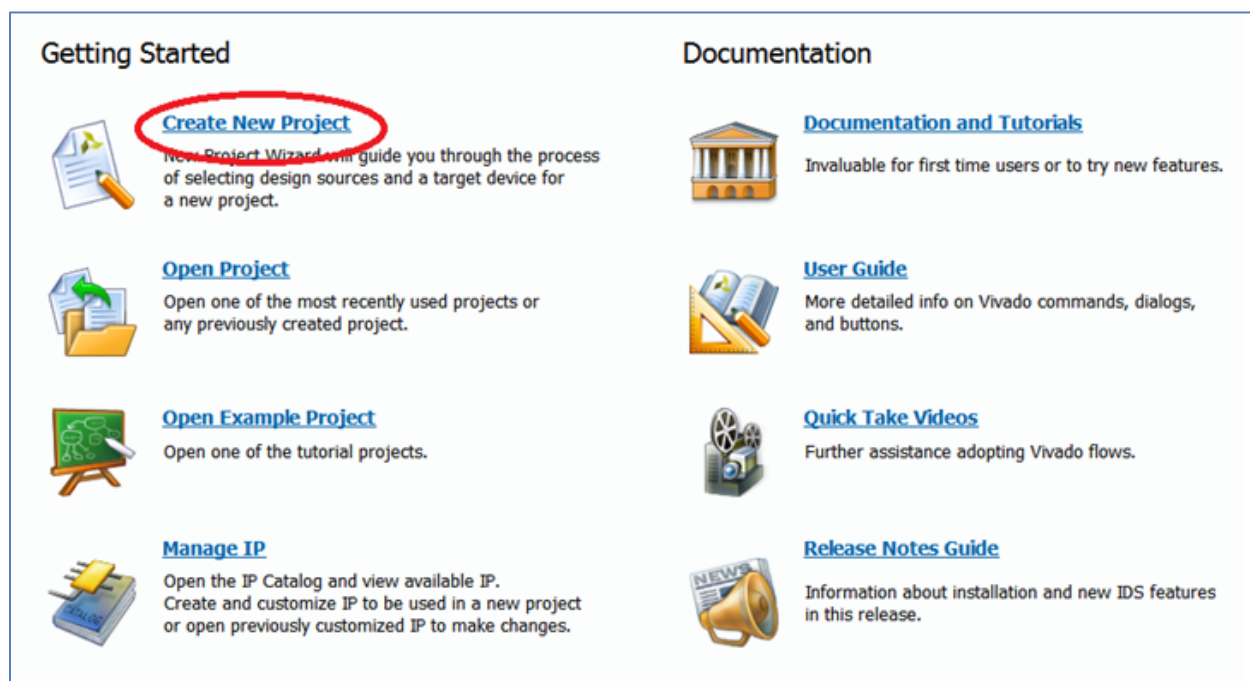


図 1 : 新規プロジェクトの作成

次の図に示すように、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。Vivado では、複数バージョンのザイリンクス ターゲット ボードがサポートされますので、ターゲット ハードウェアの選択には注意してください。

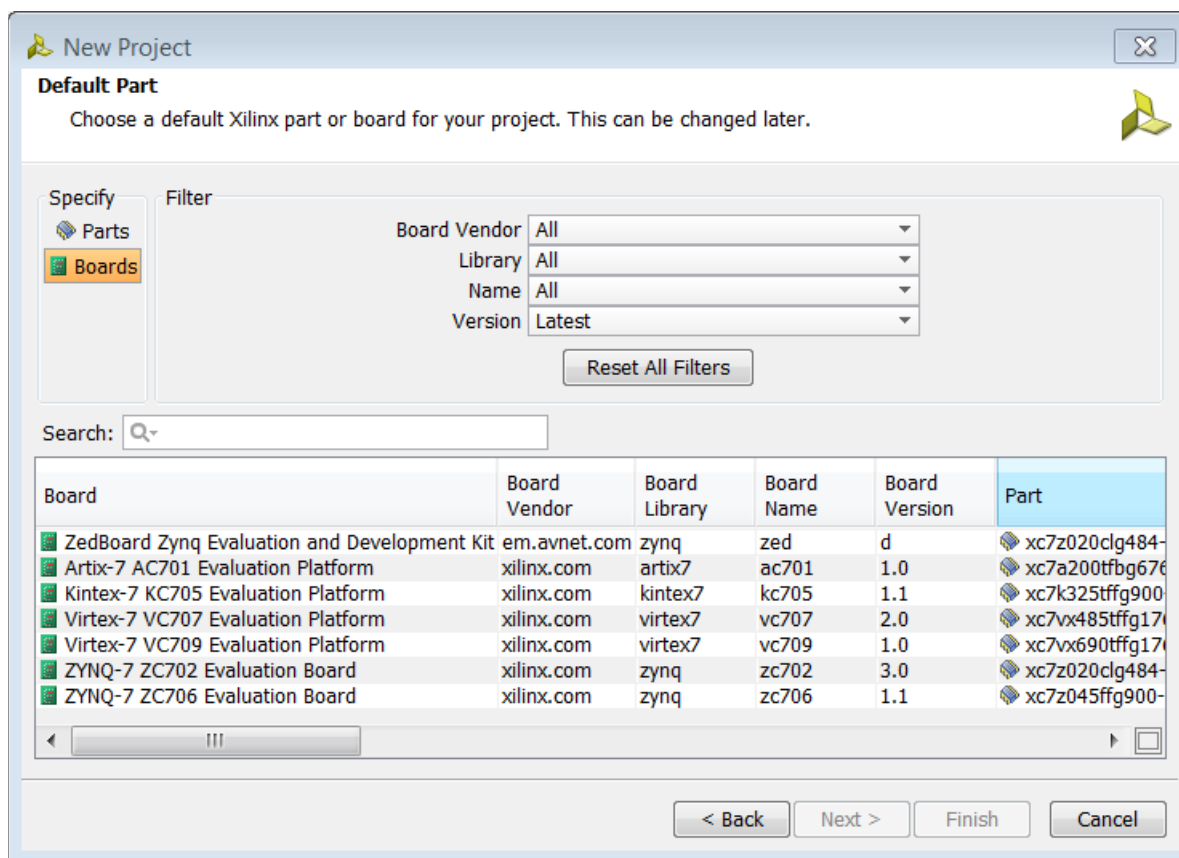


図 2 : 新規プロジェクト ターゲット デバイスの選択

注記： 次のコマンドを使用すると同じ操作が実行できます。この文書の Tcl コマンド例では、`<>` はユーザー デザイン特有のパラメーターを囲んだものを示します。`<>` シンボル自体は、コマンド文字列には含めないでください。

これは、Tcl コマンドでは次のようになります。

```
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 Integrator] の下の [Create Block Design] をクリックし、Flow Navigator で新規ブロック図を作成します。

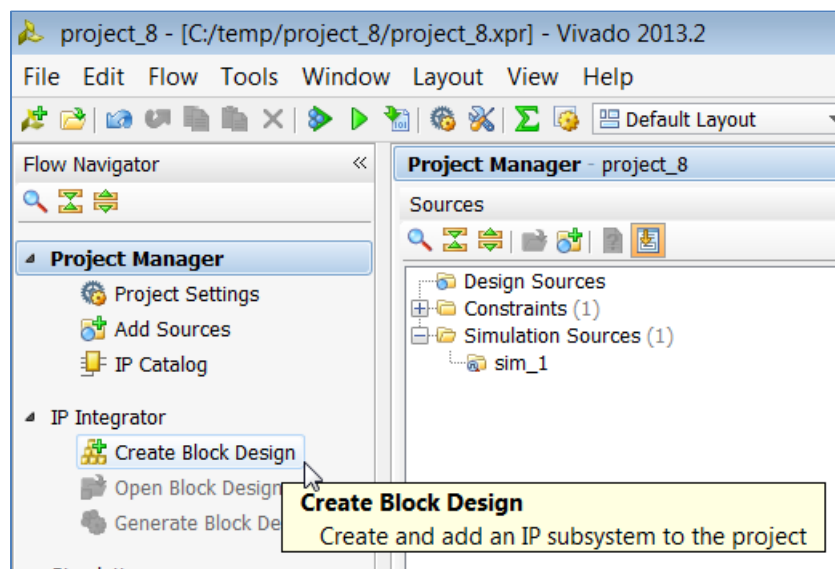


図 3 : ブロック図の作成

これは、Tcl コマンドでは次のようになります。

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

IP インテグレーター図のサイズ変更

デザインを作成すると、デザインを接続するのに使用するキャンバスが表示されます。このキャンバスの大きさは、Vivado IDE の GUI で画面の大きさを変えると必要なだけ変更できます。図の右上の [Float Window] ボタンをクリックすると、図を別の画面に移動できます。図の左上の [Diagram] タブをダブルクリックすると、図のサイズを拡大することもできます。タブをもう 1 度ダブルクリックすると、表示がデフォルト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することができます。次の図に示すように、図の左上の [Block Diagram Options] ボタンをクリックすると、色を変更できます。

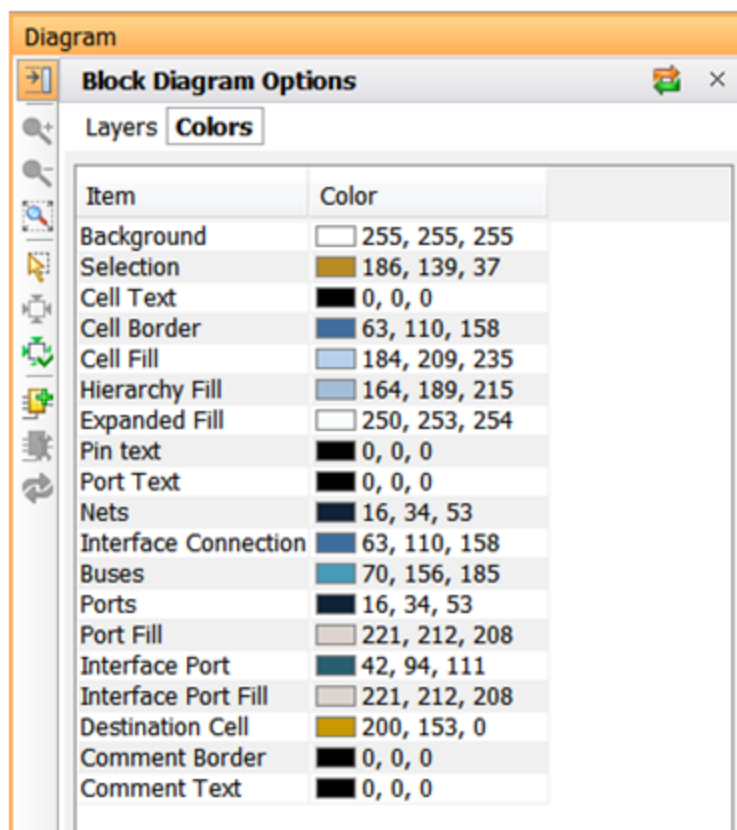


図 4 : IP インテグレーターの背景色の変更

IP インテグレーター図に表示されるほとんどすべてのオブジェクトの色を変更できます。たとえば、背景色を上記のように 240,240,240 に変更すると、明るいグレーにできます。[Block Diagram Options] を非表示にするには、右上の X ボタンをクリックするか、[Block Diagram Options] ボタンをもう 1 回クリックします。

マウス ストロークと左ボタン パネルの使用

[Zoom Fit] : 右下から左上へ

[Zoom In] : 右上から左下へ

[Zoom Out] : 左下から右上へ

[Zoom Area] : 左上から右下へ

キャンバス左側のボタンを使用すると、特定の動作を実行できます。

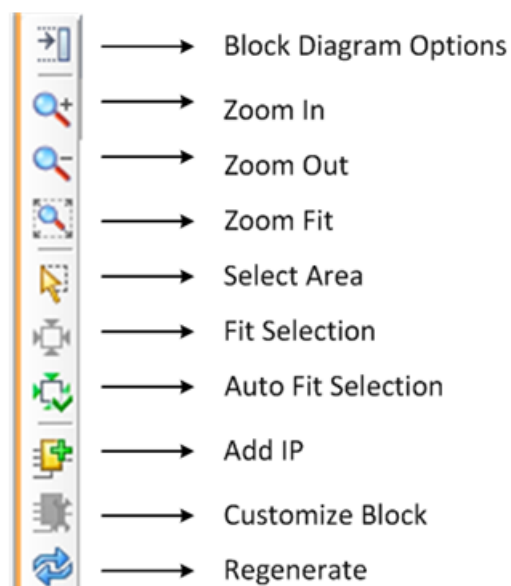


図 5 : IP インテグレーターのボタン

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へ追加できます。

1. 図を右クリックし、[Add IP] をクリックします。検索可能な IP カタログが開きます。

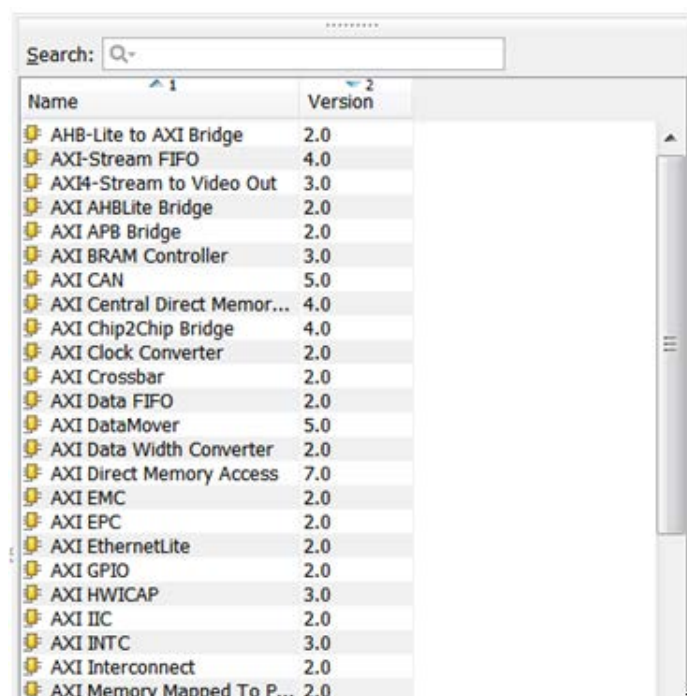


図 6 : Vivado IP カタログの起動

検索フィルターに IP 名の最初の何文字かを入力すると、それに一致する IP モジュールのみが表示されます。

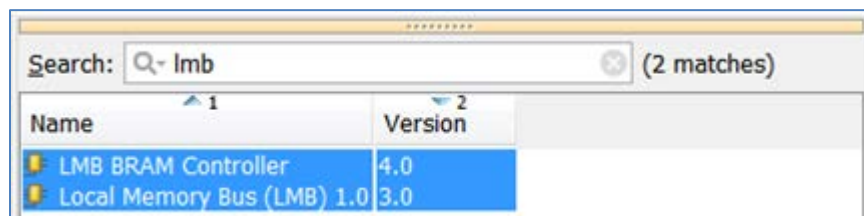


図 7 : IP カタログの検索フィルターの使用

2. 1 つの IP を追加するには、その IP 名をクリックしてキーボードの Enter キーを押すか、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、必要な IP を Ctrl キーを押しながらクリックしていき、Enter キーを押します。

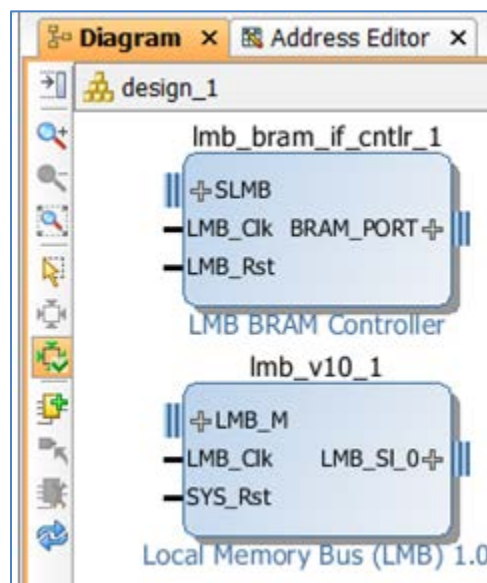


図 8 : 複数 IP の同時追加

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。

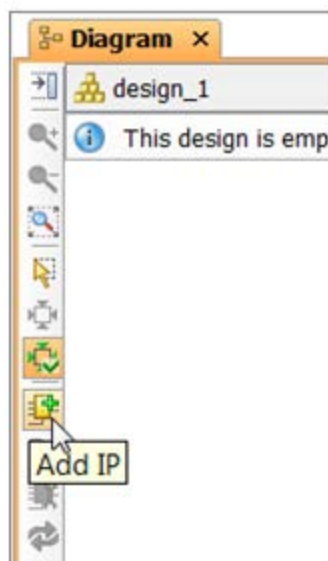


図 9 : [Add IP] ボタンによる IP の追加

5. IP は、IP インテグレーターのキャンパスの上にある Add IP リンクをクリックしても追加できます。

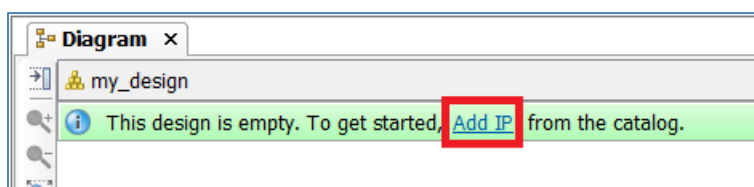


図 10 :Add IP リンクによる IP の追加

IP は、[Add IP] コマンドを実行したときのカーソル位置の近くに配置されます。

Flow Navigator の Vivado IP カタログのエントリも表示および使用できます。デュアル モニターを使用している場合は、IP カタログをご自身のモニターで開くことができます。シングル モニターを使用している場合は、IP カタログをフロートさせて、図から離すことができます。メイン IP カタログから IP を追加する場合は、IP カタログから IP を選択して図にドラッグ アンド ドロップします。

注記 : IP をダブルクリックすると 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したら、図にブロックを追加して、必要に応じてブロックをコンフィギュレーションして、インターフェイス レベルの接続または単純なネット接続を実行して、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、シンプルです。カーソルを IP ブロックのインターフェイスまたはピン コネクタの近くに移動すると、カーソルの形がペン形に変わります。IP ブ

ロックのインターフェイスまたはピン コネクタをクリックして、マウス ボタンを押したまま接続先のブロックまでドラッグします。

次の図に示すように、インターフェイス レベルの接続の場合、もっとわかりやすい接続ボックスで表示されます。

ブロックの + マークをクリックすると、その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。

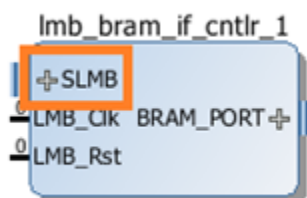


図 11 : シンボルの接続ボックス

信号レベルまたはバス レベルの接続は、シンボル上の細い接続線で表示されます。バスは、接続目的に各信号と同じように処理されます。次の図に示すように、接続をする際には、接続が可能であることを示す緑のチェック マークが接続先に表示されます。

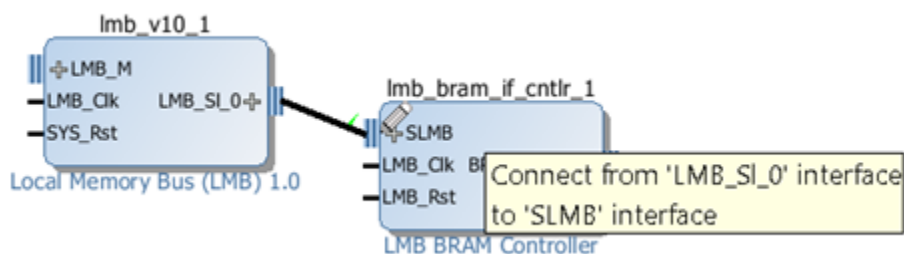


図 12 : シンボル上の信号接続またはバス接続

- 次の図に示すように、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は、各信号またはバスの接続をする前にインターフェイスをまず展開する必要があります。

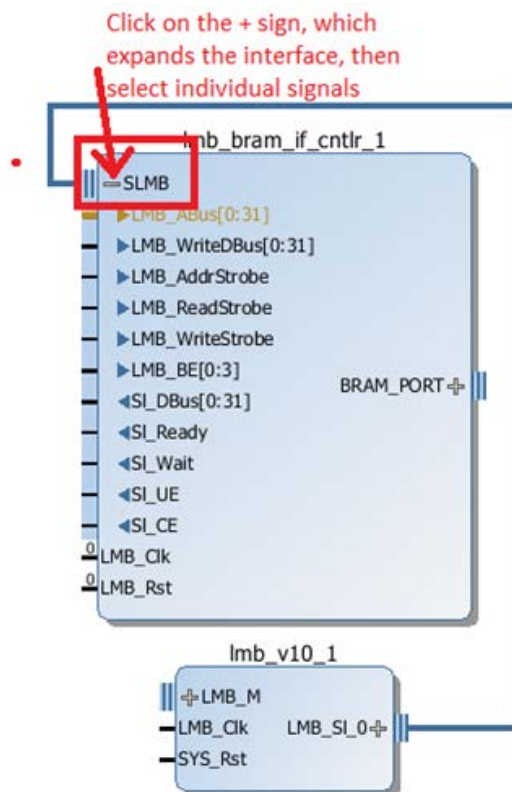


図 13 : 接続前のインターフェースの展開

7. 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. 次の図に示すように、信号またはインターフェイスを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の接続を選択し、右クリックで [Make External] をクリックします。Ctrl を押しながら複数のピンをクリックして、一度に [Make External] を実行することもできます。

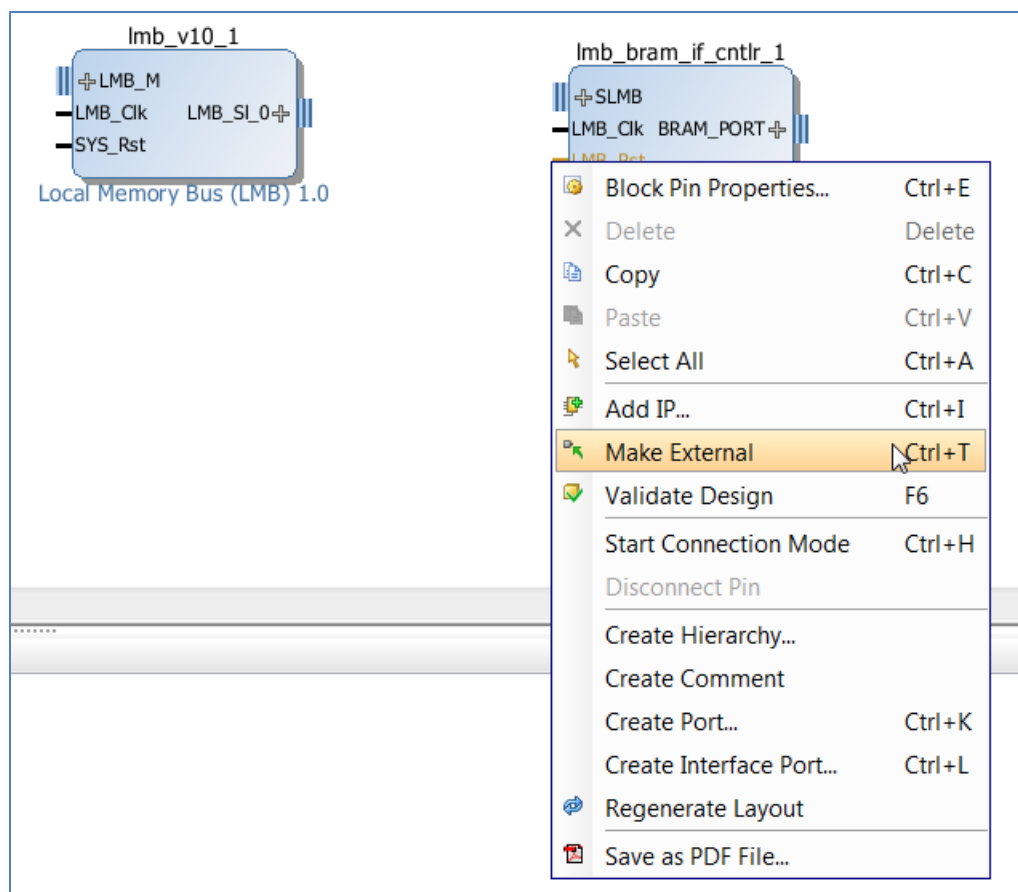


図 14 : 外部への接続

このコマンドは、IP のピンをブロック図の I/O ポートに接続するために使用します。IP インテグレーターでは、単に IP のポートが外部 I/O に接続されます。

- b. 外部に接続する 2 つ目の方法は、次の図のように右クリックしてから [Create Port] をクリックする方法です。この機能は、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際に使用します。[Create Port] を使用すると、入力/出力、ビット幅、およびタイプなどの指定をより詳細に制御できます。クロックの場合は、入力周波数ですら指定できます。

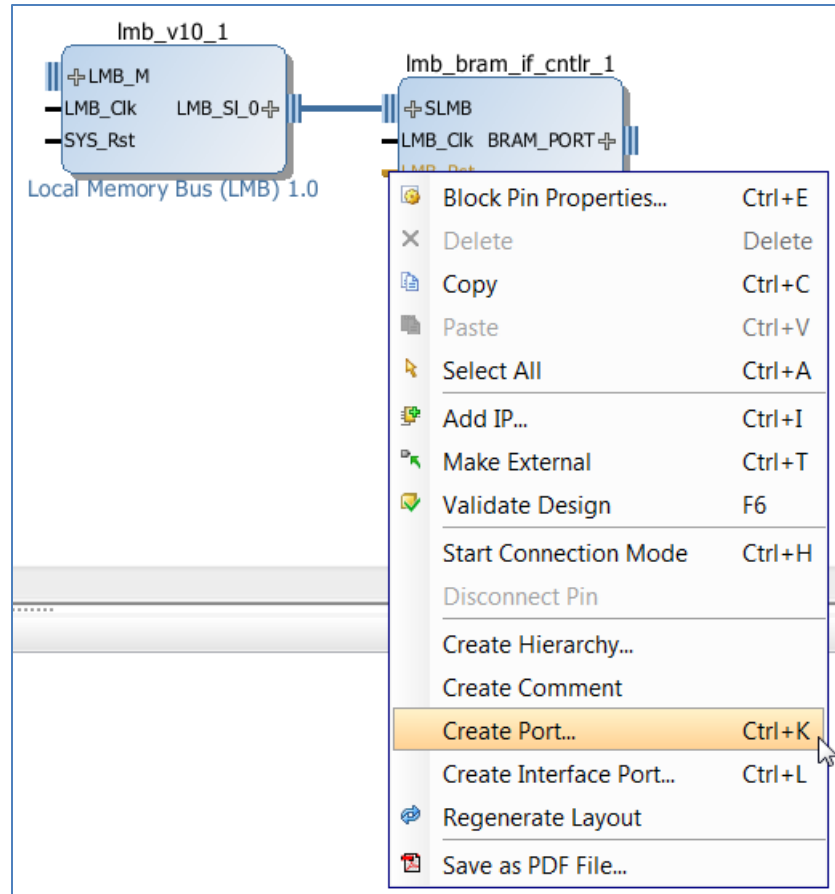


図 15 : ポートの作成

- c. 外部に接続する 3 つ目の方法は、次の図のように右クリックしてから [Create Interface Port] をクリックする方法です。

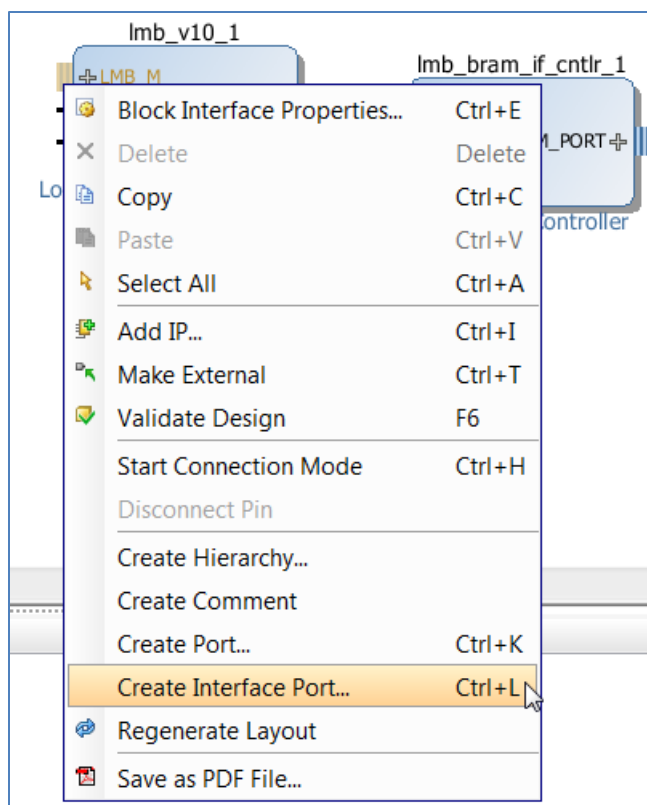


図 16 : インターフェイス ポートの作成

このコマンドは、よく使用されるファンクション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されます。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) の指定を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およびコネクションオートメーション機能

IP インテグレーターの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を使用すると、マイクロプロセッサシステムを含めたり、外部 I/O ポートへポートを接続したりする際のプロセスが自動化されます。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 または MicroBlaze プロセッサなどの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と提供されます。次の図のように [Run Block Automation] をクリックすると、単純な MicroBlaze が接続されます。

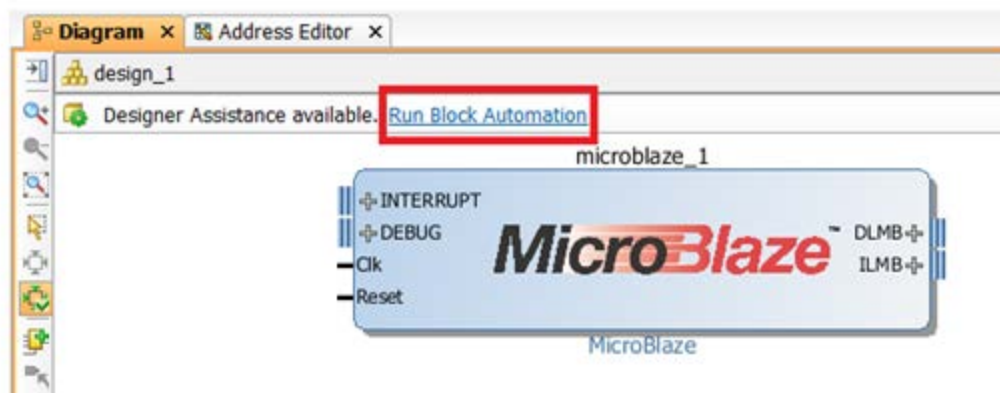


図 17 : [Run Block Automation] 機能

[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な機能に関する入力ができます。

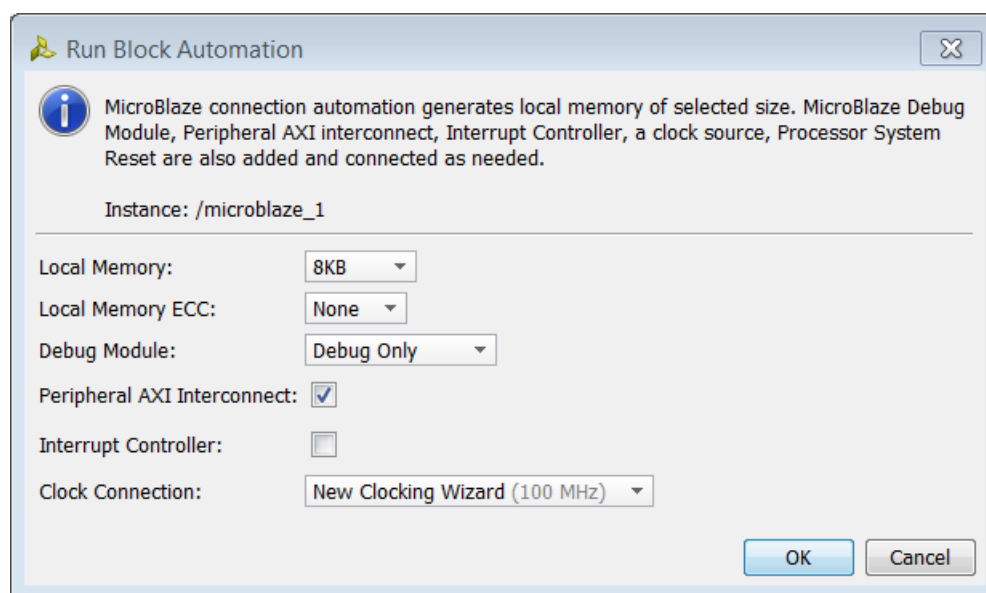


図 18 : [Run Block Automation] ダイアログ ボックス

必要なオプションを指定したら、次の図のようにブロック 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。

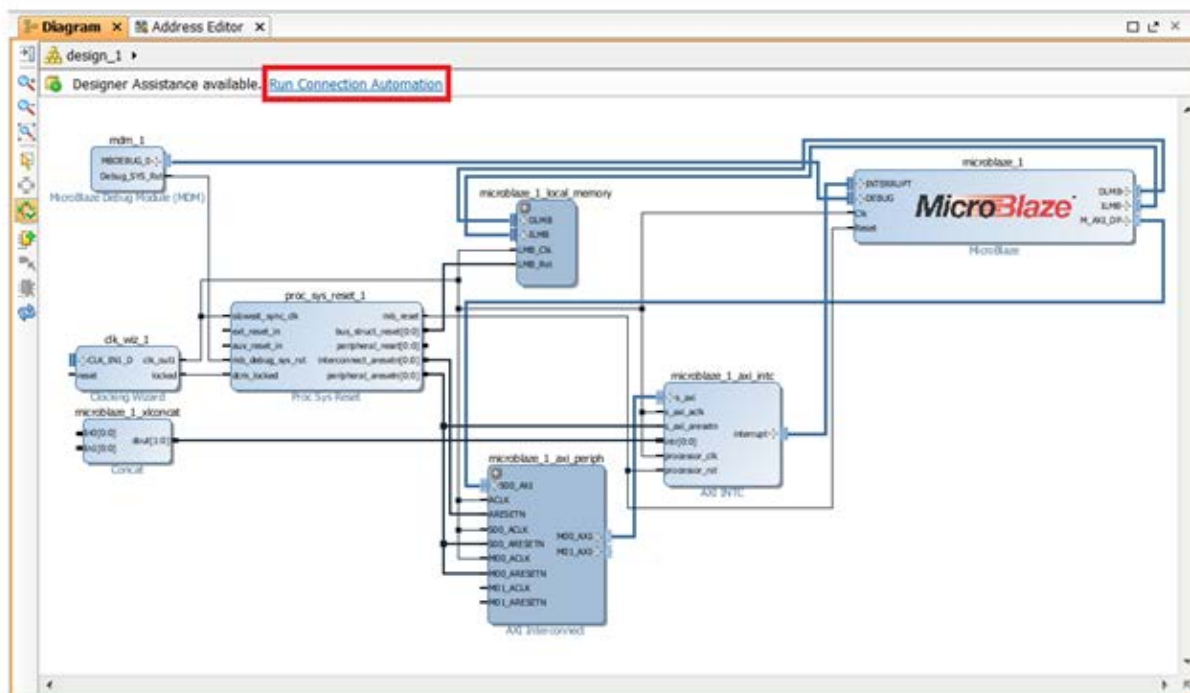


図 19 : ブロック オートメーション機能で作成された基本的なシステム

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、およびブロック メモリ ジェネレーター、Clocking Wizard、AXI インターコネクトおよび AXI 割り込みコントローラーなどを含む基本的な MicroBlaze システムが作成されます。デザインはこの段階ではまだ外部 I/O ポートに接続されていないので、IP インテグレーターには上記の図でハイライトされているようにコネクション オートメーション機能が提供されます。

[Run Connection Automation] をクリックすると、インターフェイスおよびポートを外部 I/O ポートに接続するためのアシスタンスが実行されます。

コネクション オートメーション機能を使用できるポート/インターフェイスは、次の図のようにリストされます。

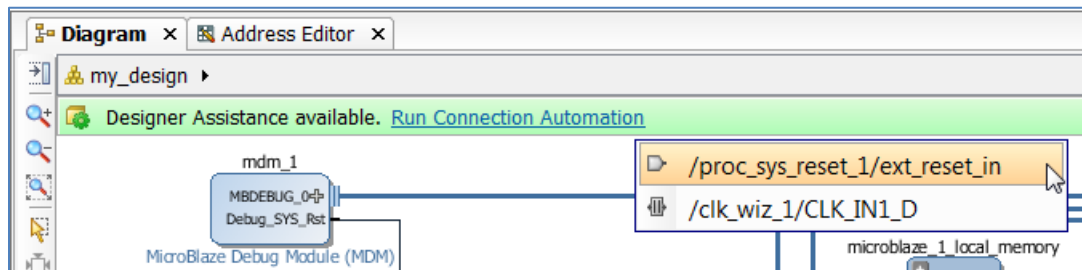


図 20 : コネクション オートメーションを使用可能なポートおよびインターフェイスのリスト

ザイリンクスの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/使用される FPGA ピンはデザインで認識されるので、その情報に基づいて 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はデザイン内のポートを外部ポートに接続します。IP インテグレーターは、この後最適な物理制約および該当する I/O ポートに必要なその他の I/O 制約を作成します。上記のデザインの場合、Proc Sys Reset IP を外部リセット ポートに接続し、Clocking Wizard を外部クロック 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 オプションを選択した場合は、次の図のようなダイアログ ボックスが開きます。

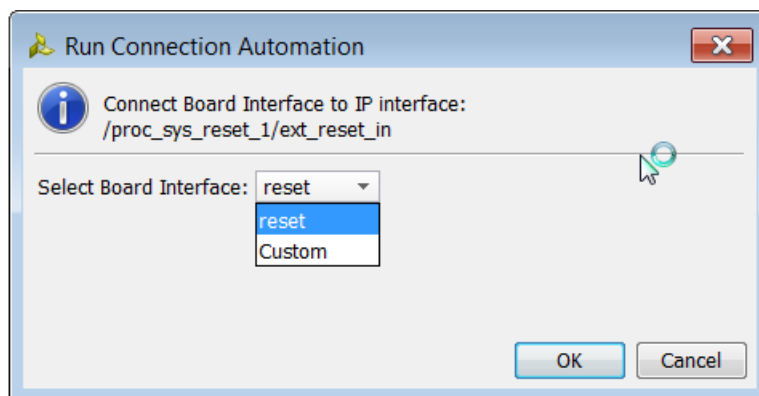


図 21 : [Run Connection Automation] ダイアログ ボックス

ターゲット ボード、この場合は KC705 に既存のリセット ピンを選択するか、カスタム リセット ピンを指定します。指定したら、そのリセット ピンを Proc Sys Rst IP の ext_reset_in ピンに接続します。

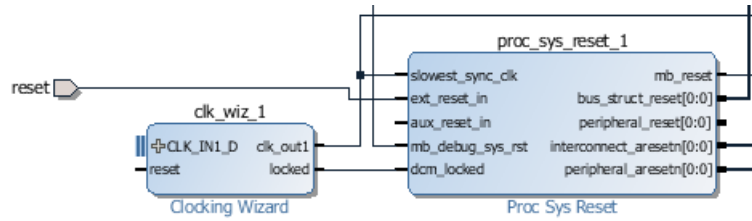


図 22 : ボード リセット ピンへのリセット ポートの接続

ここまでで、別の IP をコネクション オートメーション機能が使用可能な IP インテグレーター デザインにインスタンスエートしたとします。たとえば、AXI GPIO IP をデザインにインスタンスエートしたとします。これで [Run Connection Automation] をクリックすると、AXI インターコネクトを介して AXI GPIO の s_axi ポートを MicroBlaze プロセッサに接続できることが表示されます。

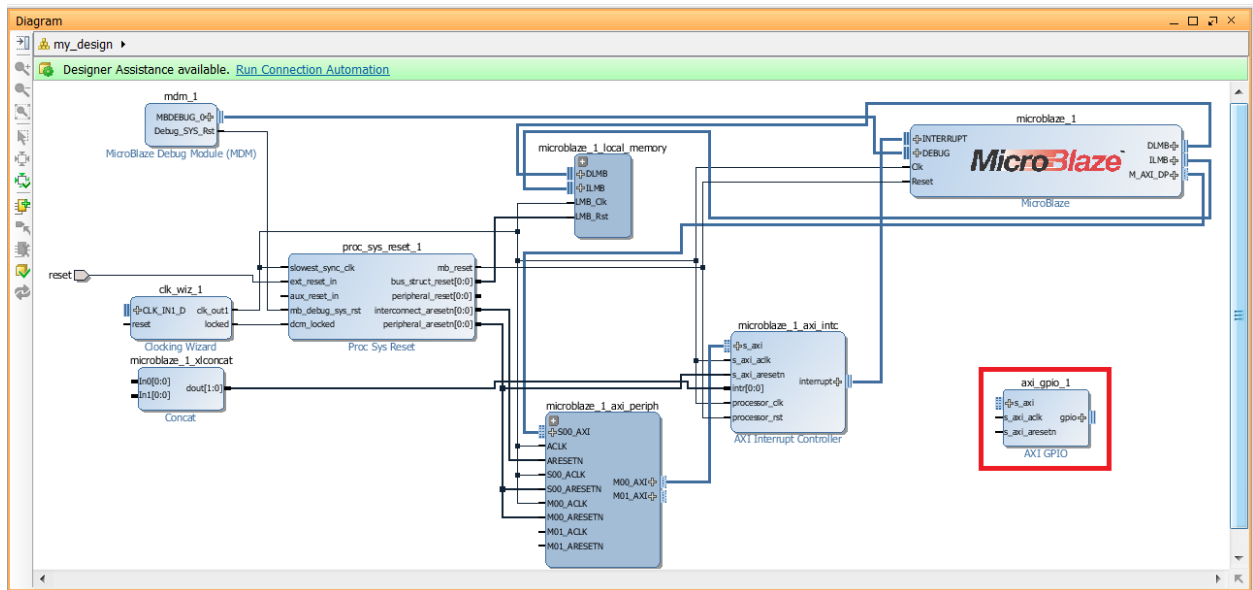


図 23 : コネクション オートメーションを使用した潜在的な接続の表示

[Run Connection Automation] をクリックすると、GPIO のスレーブ AXI ポートを MicroBlaze マスターに接続可能なことを示す次のようなポップアップ ウィンドウが表示されます。デザインにマスターが複数含まれる場合は、いずれかを選択できます。

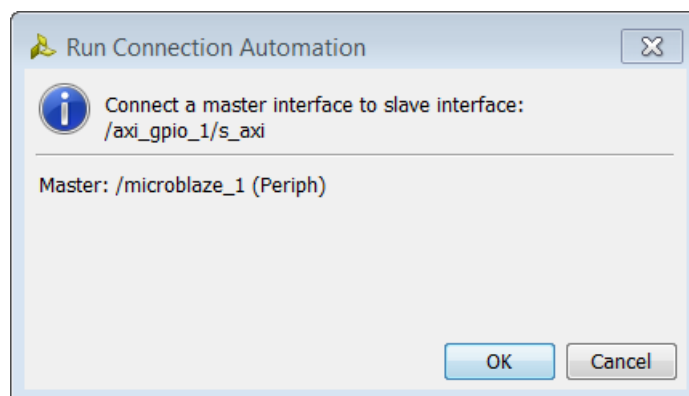


図 24 : s_axi スレーブ インターフェイスの MicroBlaze マスターへの接続

[Run Connection Automation] ダイアログ ボックスで [OK] をクリックすると、接続が実行され、次の図のようにハイライトされます。

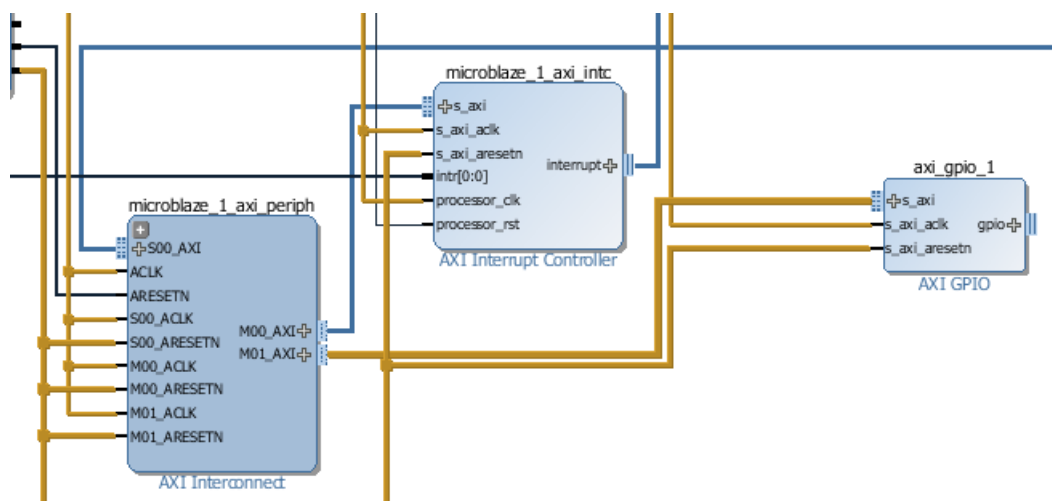


図 25 : マスター/スレーブ接続

ザイリンクス ターゲット リファレンス プラットフォームが使用された場合は、I/O ポートの接続に関してより詳細なアシスタンスが表示されます。次の図を参照してください。

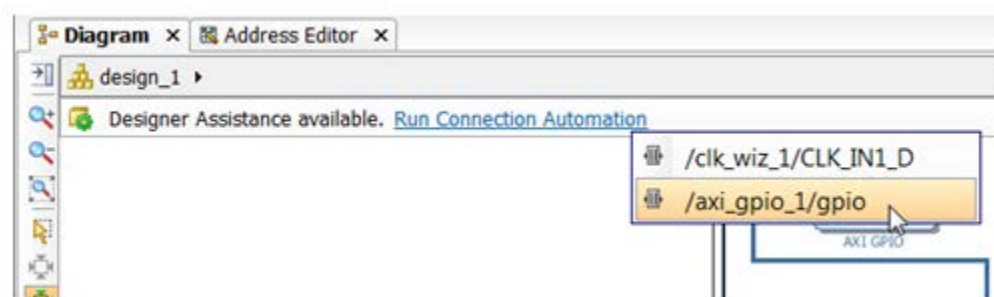


図 26 : ザイリンクス ターゲット リファレンス プラットフォームで使用可能な詳細なアシスタンス

gpio ポートをクリックすると、ボードに関して使用可能な情報に基づいて、次のようなオプションが表示されます。

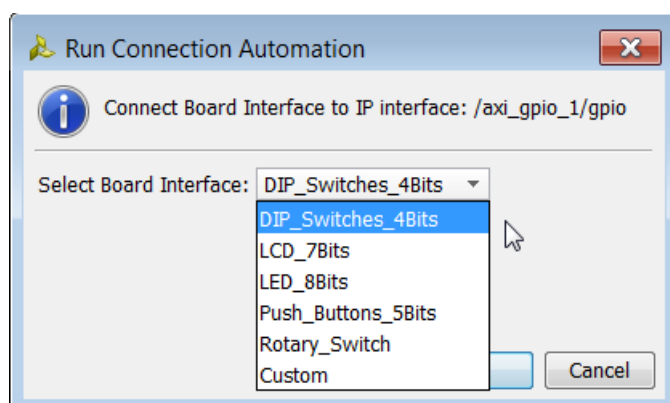


図 27 : ポート選択後に表示されるオプション

この場合、6 つの選択肢が表示されます。gpio ポートは、4 ビットの Dip Switch、7 ビットの LCD、8 ビットの LED、5 ビットのプッシュ ボタン、ボードの Rotary Switch、カスタム インターフェイスのいずれかに接続できます。いずれかを選択すると、gpio ポートがボードの既存の接続に接続されます。

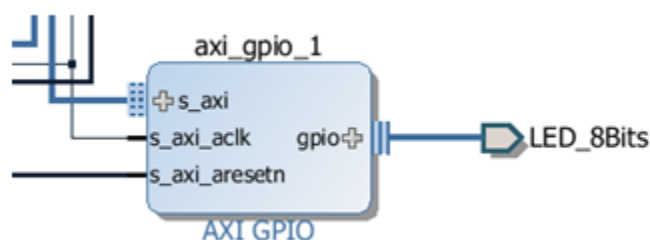


図 28 : ボード I/O エLEMENT のデザインへの接続

IP ブロックの並べ替え

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ために並べ替えることができます。完了した図または作業中の図を並べ替えるには、[Regenerate] ボタンをクリックします。

ブロックをクリックしてドラッグしたり、矢印キーを使用すると、ブロックを手動で移動できます。図では、ブロックを移動する際に表示される濃いグレーの縦のバーで示される特定の列位置だけを使用できます。ブロックを移動する際に図に表示されるグリッドを使用すると、ブロックおよびピンを揃えやすくなります。

コピーと貼り付け

図内でブロックをコピーして貼り付けるには、Ctrl + C および Ctrl + V を使用できます。

階層の作成

次の図に示すように、Ctrl を押しながらクリックして必要な IP ブロックを選択し、右クリックで [Create Hierarchy] をクリックすると、階層ブロックを作成できます。

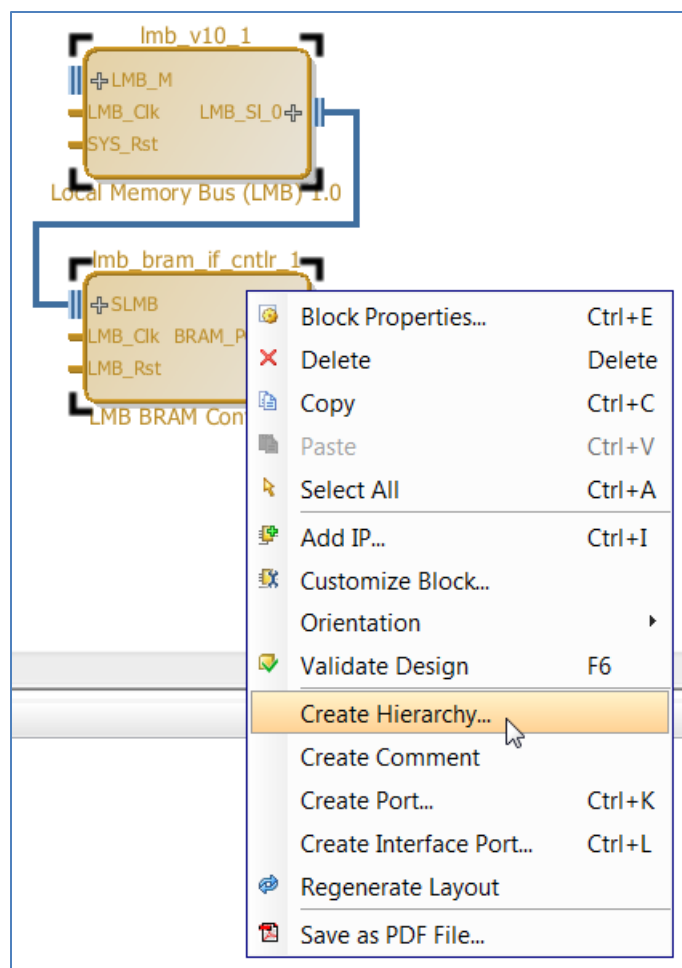


図 29 : 階層ブロック図の作成

選択したブロックを含む新しいレベルの階層が作成されます。空のレベルの階層を作成しておいて、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左上のエクスプローラータイプのパス情報を使用すると確認できます。[Ungroup Instances] を使用すると、IP ブロックを階層グループから出すことができます。

[Create Hierarchy] をクリックすると、次の図のように [Create Hierarchy] ダイアログ ボックスが表示され、階層名を指定できます。

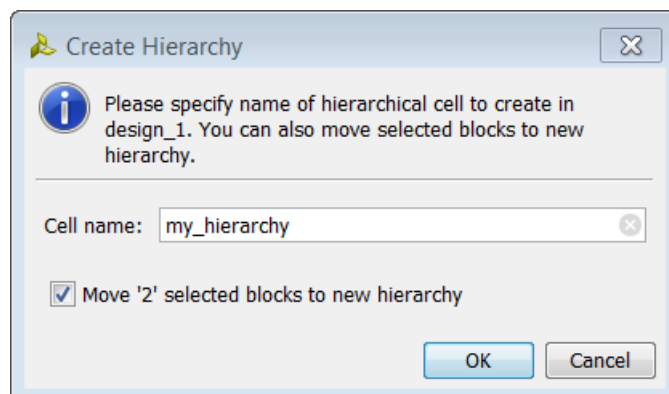


図 30 : [Create Hierarchy] ダイアログ ボックス

これにより、1 つのブロックの下に 2 つのブロックのグループができます。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。- マークをクリックすると、元のグループ表示に戻ります。

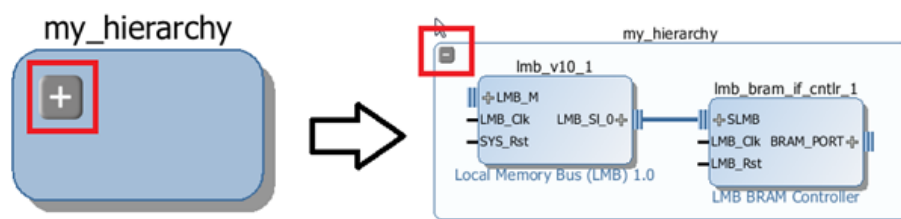


図 31 : 2 つのブロックを 1 つのブロックに統合

メモリ マップの作成

このデザインのアドレス マップを生成するには、[Address Editor] タブをクリックします。通常、アドレスはブロック図でスレーブをインスタンスエートすると自動的にマップされますが、[Auto Assign Address] ボタン (左側一番下のボタン) をクリックしてもマップされます。アドレスを生成せずに IP インテグレーターから RTL を生成した場合、アドレスの自動割り当てを選択可能にするプロンプトが表示されます。アドレスは、[Offset Address] および [Range] 列に値を入力すると、手動で設定することもできます。

注記 : [Address Editor] タブはバス マスターとして機能する IP ブロック (次の図の場合、MicroBlaze プロセッサ) が図に含まれている場合にのみ表示されます。

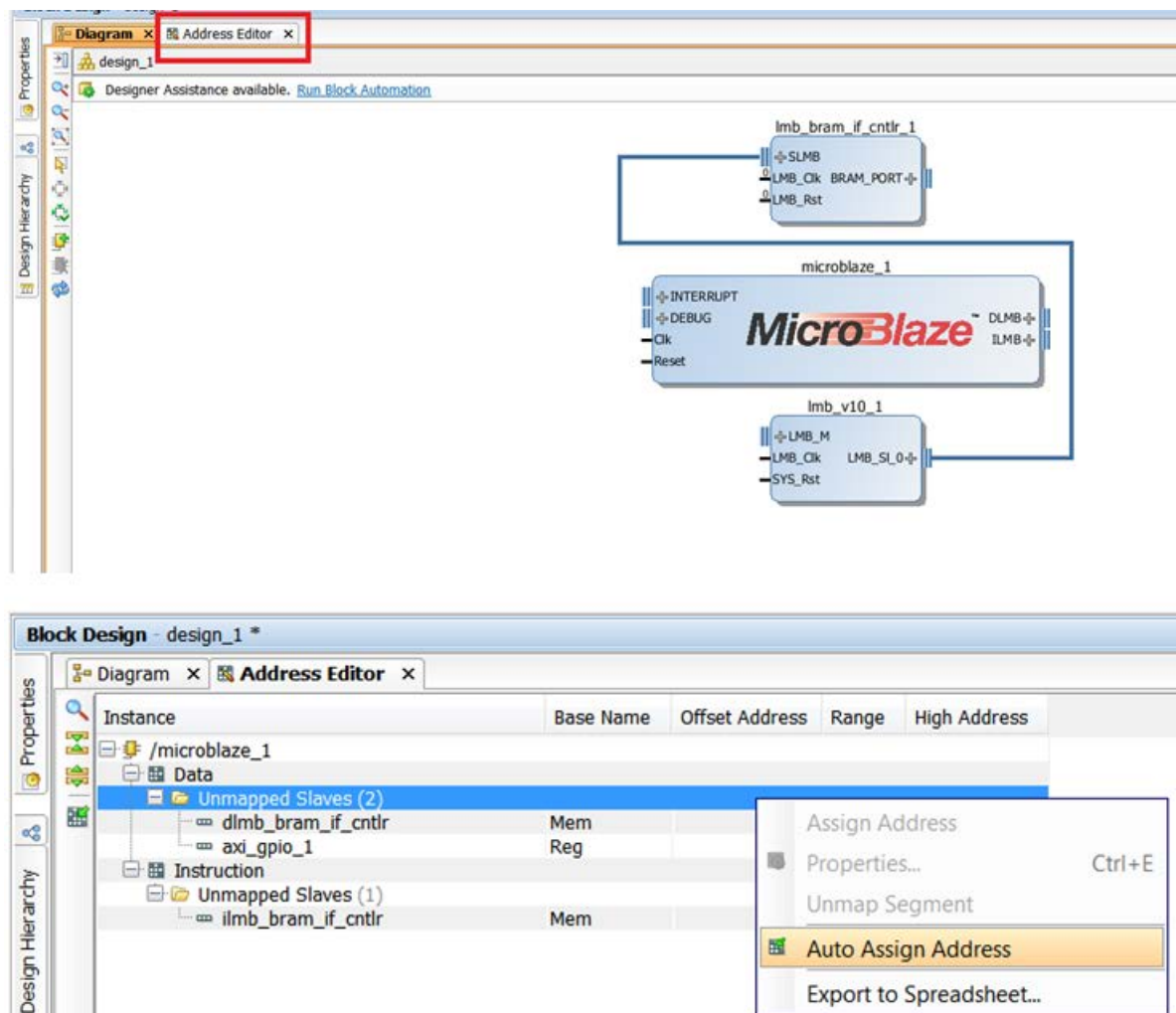


図 32 : [Address Editor] タブ

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らかの潜在的な問題が発生することもあります。たとえば、クロックピンの周波数が正しく設定されないことがあります。次の図の [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。

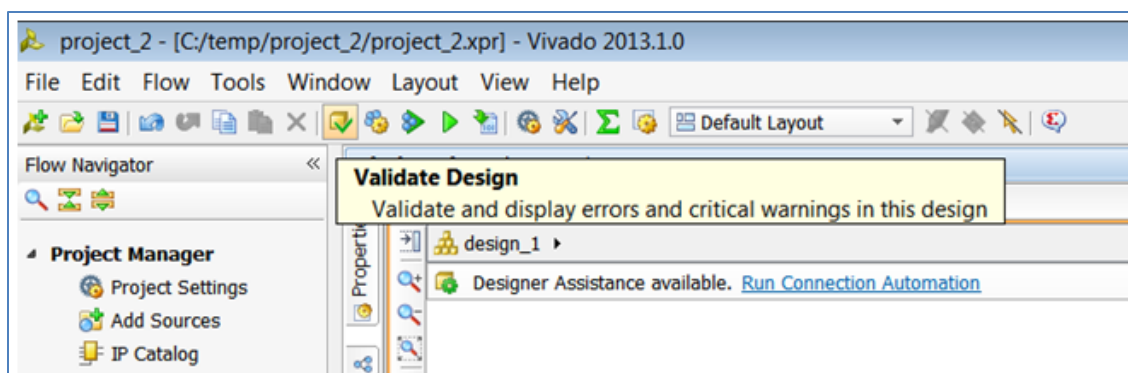


図 33 : デザインの検証

デザインに警告またはエラーがない場合、[Validate Design] を実行後に次の図のようなメッセージが表示されます。

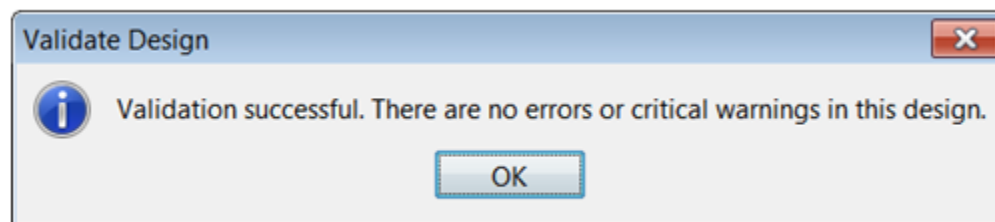


図 34 : 検証に問題がなかったことを示すメッセージ

ブロック図の最上位デザインへの統合

ブロック図を完了してデザインを検証したら、残す手順は 2 つです。まず、出力ファイルを生成する必要があります。これは、すべての IP のソース ファイルと適切な制約が生成され、Vivado の [Sources] ビューから使用できるようになってから実行します。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、適切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生成できない場合、コンソールにそれを示すメッセージが表示されます。出力ファイルを生成するには、次の図のように 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。

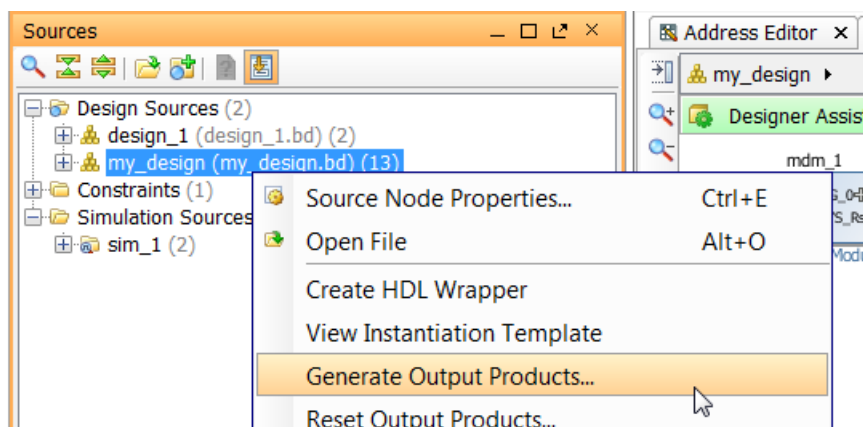


図 35 : [Generate Output Products] コマンド

IP インテグレーターのブロック図は、それより上位のデザイン レベルに統合したり、デザイン階層の最上位レベルにしたりできます。IP インテグレーター デザインを上位のデザインに統合するには、単にそのデザインを最上位 HDL ファイルにインスタンスエートします。

Vivado の [Sources] ビューでブロック図を右クリックして、[Create HDL Wrapper] をクリックしても、上位にブロック図をインスタンスエートできます。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。これで、デザインはエラボーレーション、合成、インプリメンテーションできるようになりました。

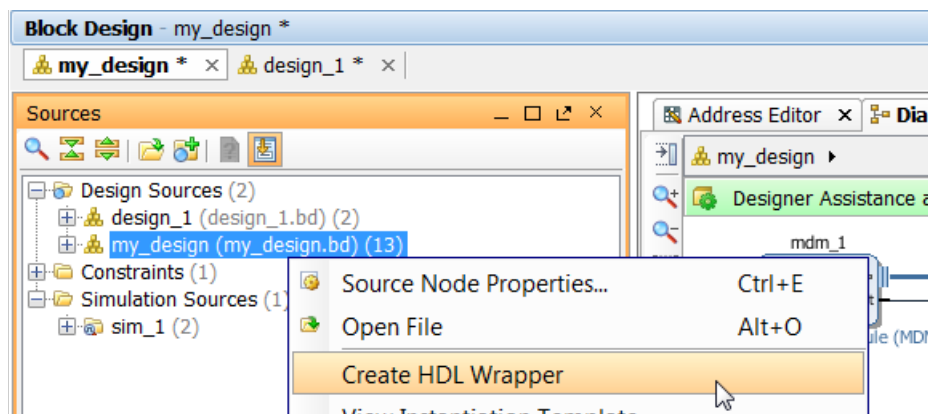


図 36 : HDL ラッパーの作成

ハードウェア定義の SDK へのエクスポート

MicroBlaze や Zynq7 のようなプロセッサを含むデザインの場合、プロジェクトのハードウェア定義をソフトウェア開発キット (SDK) にエクスポートできます。これにより、デザインに使用される IP を理解するために SDK で必要とされる XML ファイルがエクスポートされるほか、プロセッサの全体像からメモリ マップもエクスポートされます。ハードウェア定義をエクスポートするには、デザインをインプリメンテーションして、できればビットストリームを生成しておく必要があります。Zynq7 ベースのデザインが FPGA のプロセッシング ロジック ファブリックにない場合は (まれなケースですが)、ビットストリームは生成する必要がありません。

SDK にハードウェアをエクスポートするには、[File] → [Export Hardware for SDK] をクリックします。

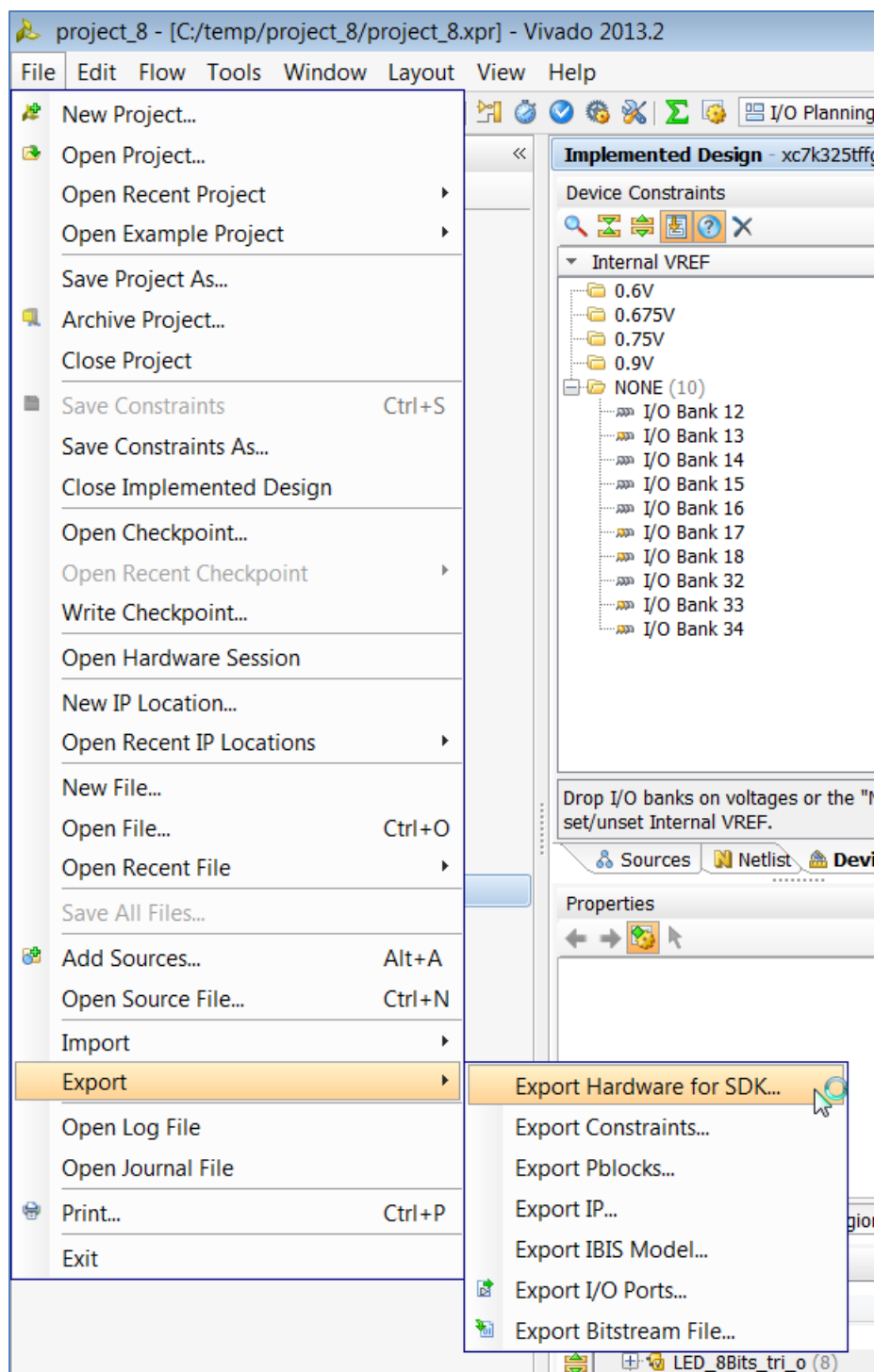


図 37 : ハードウェア定義のエクスポート

[Export Hardware for SDK] ダイアログ ボックスが開きます。

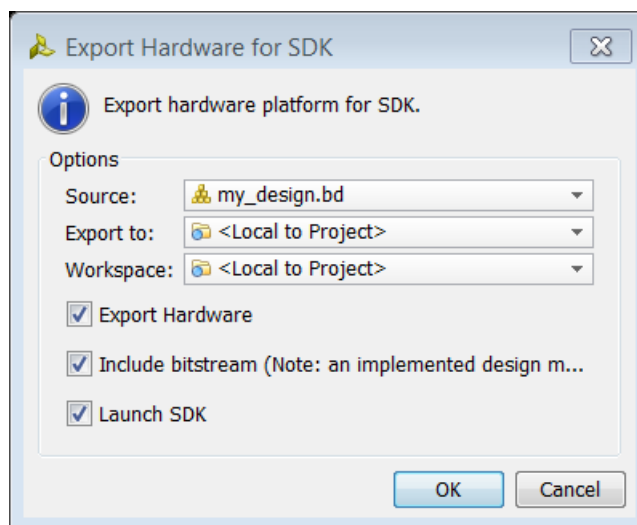


図 38 : ハードウェア定義のエクスポート

上記の図には、3 つのチェック ボックスがあります。最初のチェック ボックスのみをオンにすると、SDK に必要な XML ファイルのみが作成されます。2 つ目のチェック ボックスをオンにすると、エクスポート プロセスの一部としてビットストリームの生成が含まれます。3 つ目のチェック ボックスをオンにすると、Vivado から SDK が起動され、右のワークスペースに開きます。

注記 : [Export Hardware] が実行されるようにするには、ブロック図を開いておく必要があります。[Export Hardware] は、アクティブなブロック図にしか実行されません。また、[Include bitstream] が実行されるようにするには、インプリメント済みデザインを開いておく必要があります。

プロジェクト ベースのフローの場合、ハードウェアは次のディレクトリにエクスポートされます。

project_name/project_name.sdk/SDK/SDK_Export/hw

SDK が起動されると、そのエクスポートしたハードウェア定義を使用してカスタム アプリケーション プロジェクトを作成できます。SDK では、ターゲット ハードウェアに合わせて必要なドライバーおよびボード サポート パッケージが作成されます。

ブロック図のパッケージ

IP インテグレーター デザインを作成し、インプリメントしてターゲット ハードウェアでテストした後、そのデザインをパッケージにして、別のデザインで再利用可能な IP に変換しておくことができます。デザインをパッケージにすると、それが IP に変換されて、IP カタログから使用できるようになります。その IP は別のデザインの一部としてインスタンスエートできます。

ブロック図をパッケージにするには、Vivado IDE の [Sources] ビューでそのブロック図を右クリックして、[Package Block Design] をクリックします。

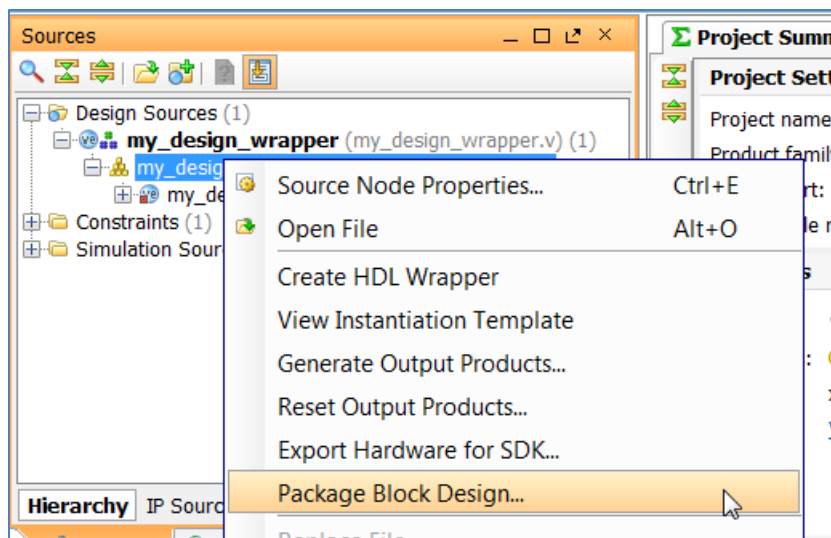


図 39 : [Package Block Design] コマンド

ブロック図を生成する必要があることを示す [Package Block Design] ダイアログ ボックスが開きます。[Yes] をクリックします。

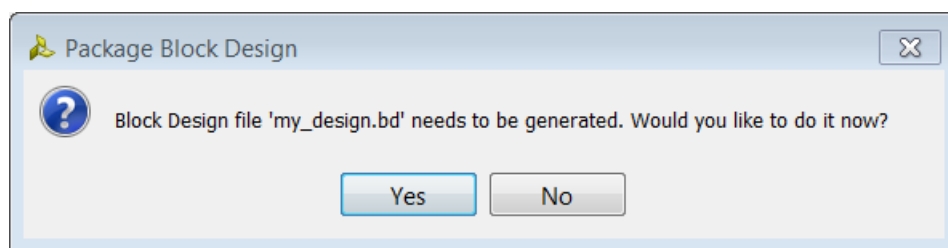


図 40 : [Package Block Design] ダイアログ ボックス

上記のダイアログ ボックスで [Yes] をクリックすると、ブロック図にインスタンス化されたすべての IP の出力ファイルが生成されて、[Manage Output Products] ダイアログ ボックスが開きます。

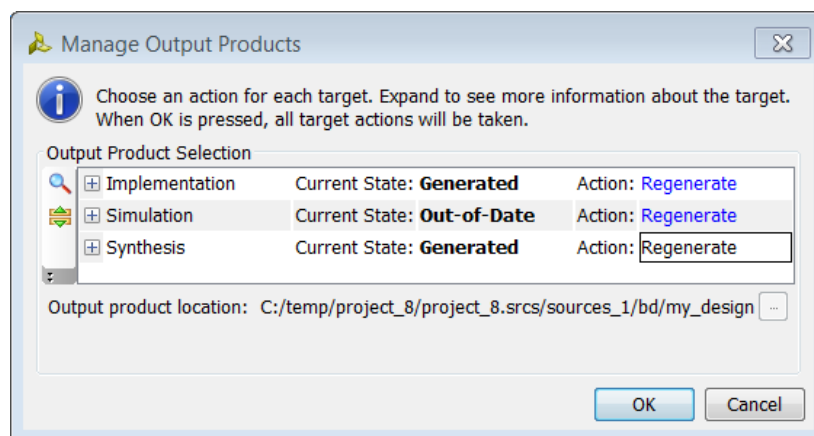


図 41 : [Manage Output Products] ダイアログ ボックス

[Manage Output Products] ダイアログ ボックスの [Action] フィールドを必要であれば [Regenerate] に変えて、[OK] をクリックします。

[Package IP] ウィンドウが開きます。

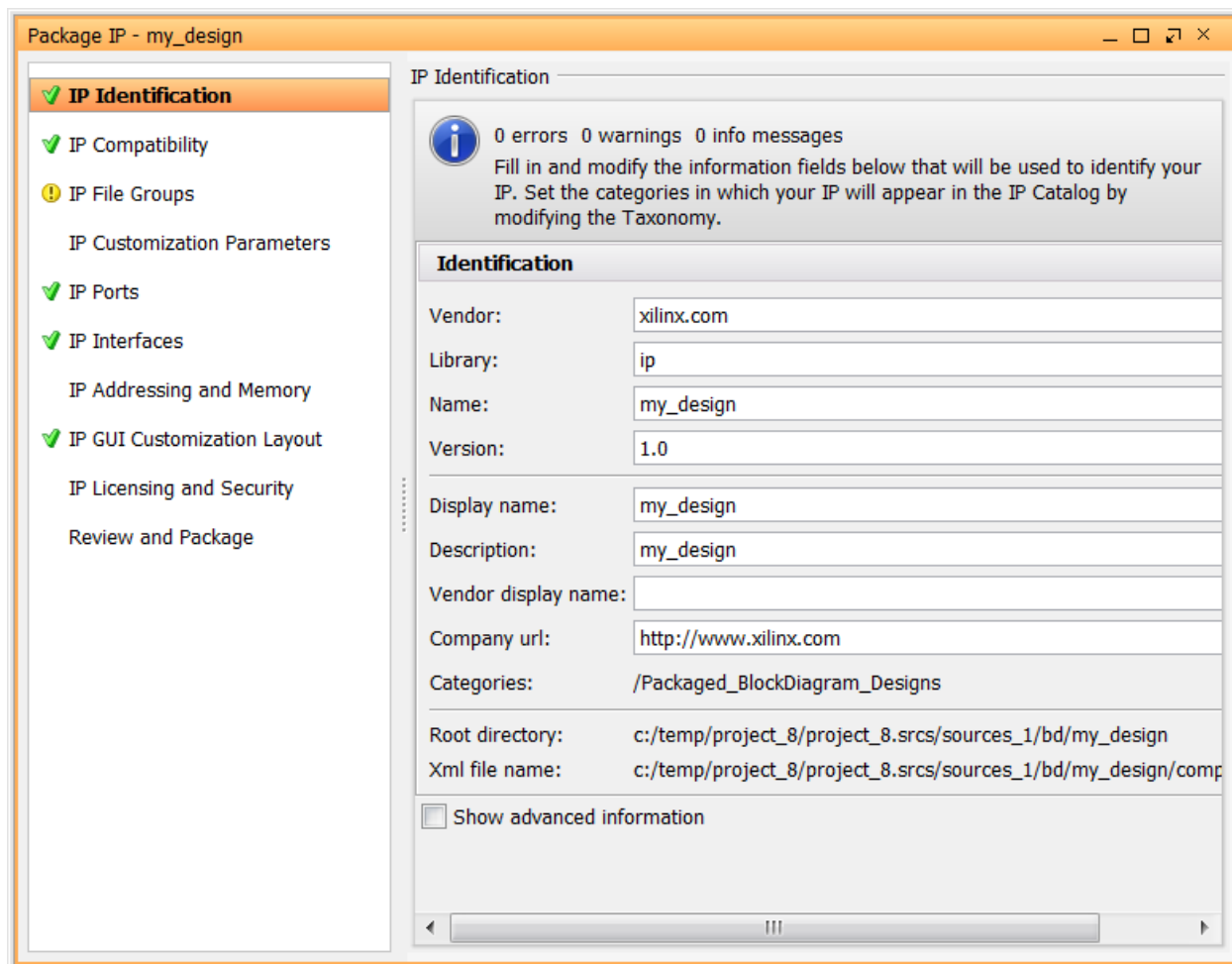


図 42 : [Package IP] ウィンドウ

[Review and Package] をクリックし、[Package IP] をクリックします。パッケージされると、次の図のように IP が IP インテグレーター カタログに表示されるようになります。

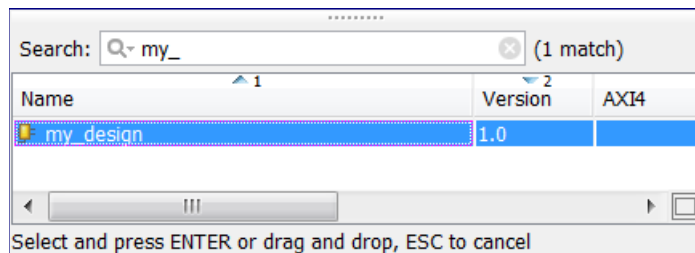


図 43 : [Package IP] ウィンドウ

新しくパッケージしたデザインは、[Packaged BlockDiagram Designs] カテゴリの下の Vivado IP カタログにも表示されます。このカテゴリ名は、ブロック図をパッケージする際に変更できます。

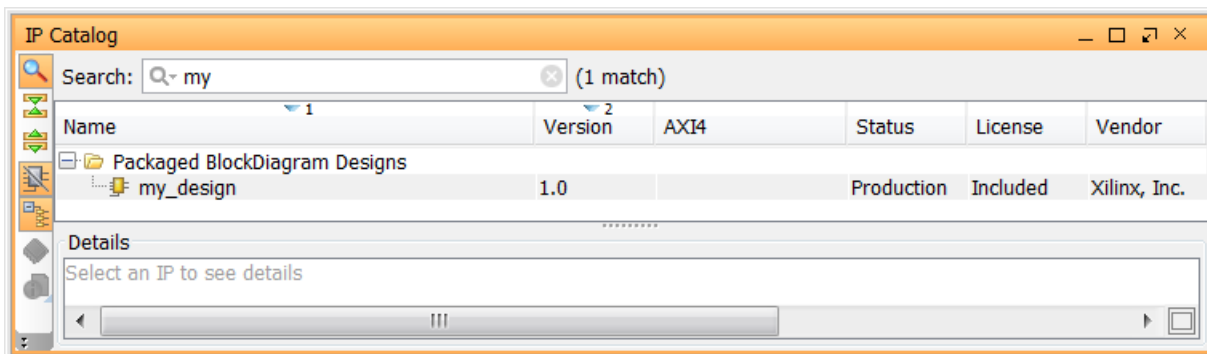


図 44 : [Packaged BlockDiagram Designs] カテゴリ

ELF ファイルのエンベデッド デザインへの追加および関連付け

MicroBlaze または Zynq7 ベース デザインのようなマイクロプロセッサ ベースのデザインの場合、SDK (またはその他のソフトウェア開発ツール) で生成される ELF ファイルをインポートして、Vivado IDE のブロック デザインと関連付けることができます。この後、ELF ファイルと一緒に Vivado IDE からビットストリームをプログラムし、ターゲット ハードウェアで実行できます。

1. このタスクを実行するには、[Sources] ビューの [Design Sources] を右クリックして [Add Sources] をクリックし、ELF ファイルを追加します。

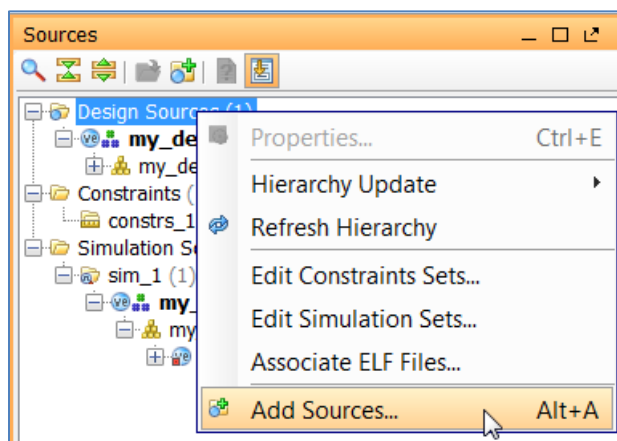


図 45 : [Add Sources] コマンド

2. [Add Sources] ダイアログ ボックスが開きます。[Add or Create Design Sources] がデフォルトで選択されます。[Next] をクリックします。
3. [Add or Create Design Sources] ページで [Add Files] をクリックします。

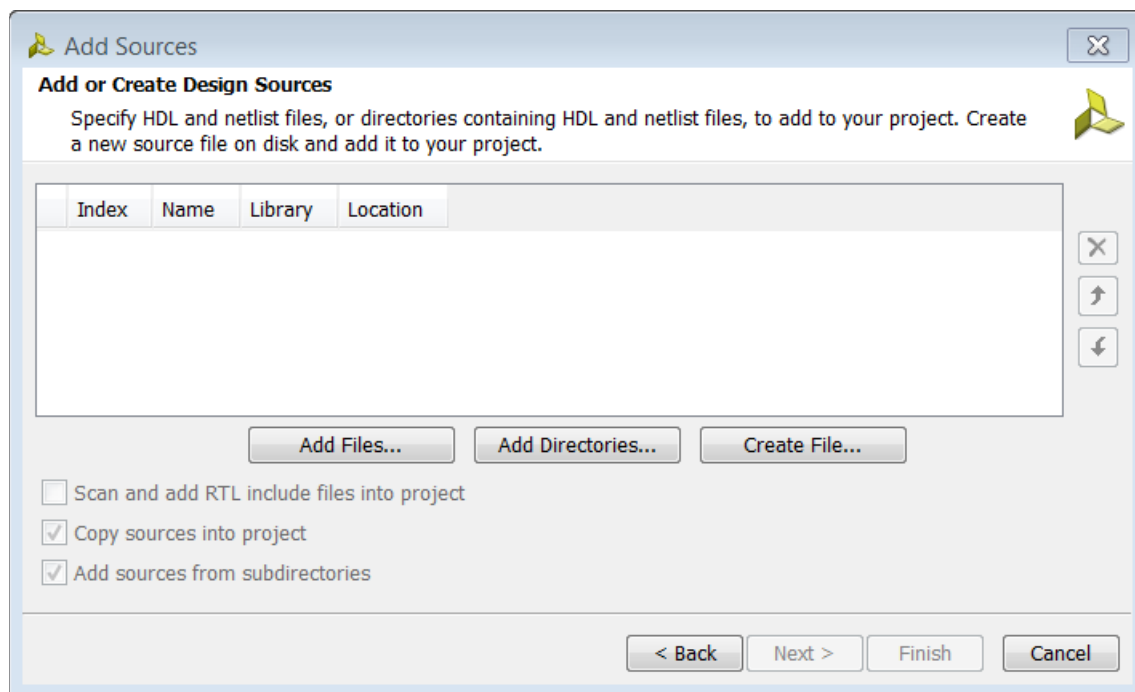


図 46 : [Add Sources] ダイアログ ボックス : [Add or Create Design Sources] ページ

4. [Add Source Files] ダイアログ ボックスが開きます。ELF ファイルを選択して、[OK] をクリックします。

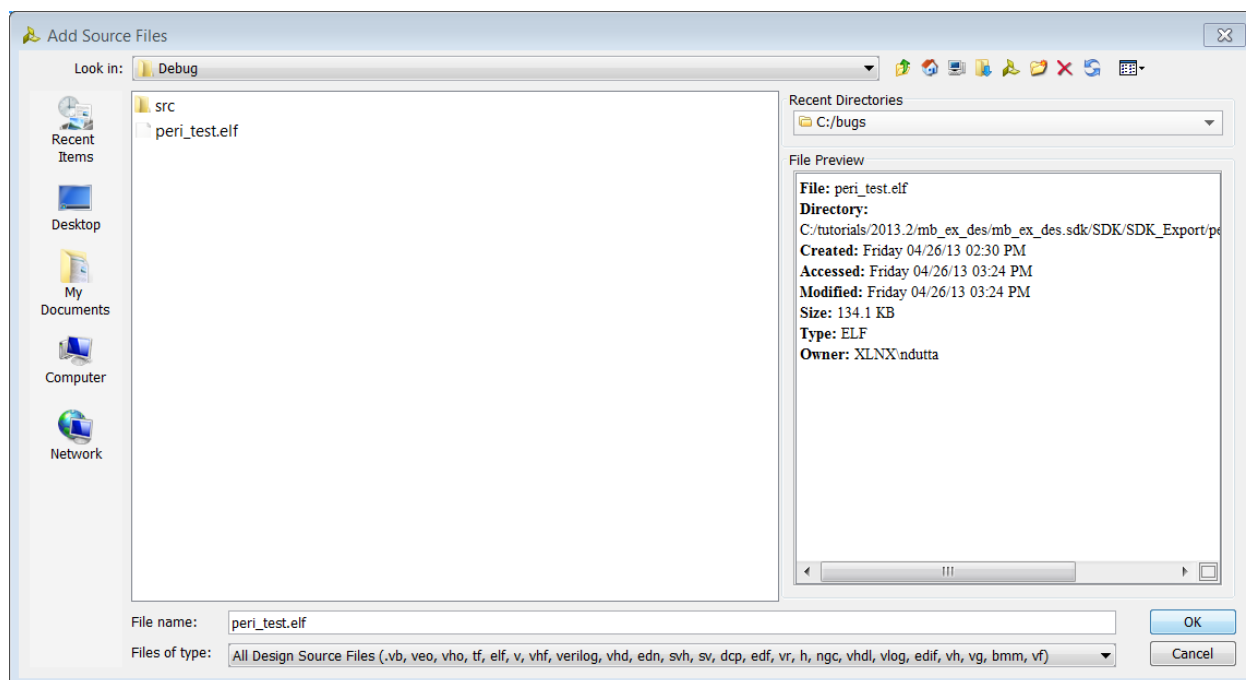


図 47 : [Add Source Files] ダイアログ ボックス

5. [Add Sources] ダイアログ ボックスの [Add or Create Design Sources] ページには、プロジェクトへ追加された ELF ファイルが表示されます。[Copy sources into project] をオンにして ELF ファイルをプロジェクトにコピーすることもできます。元の ELF ファイル ファイルで作業する場合はチェックボックスをオフにしておきます。[Finish] をクリックします。
6. [Sources] ビューの ELF フォルダに追加した ELF ファイルが表示されるようになります。

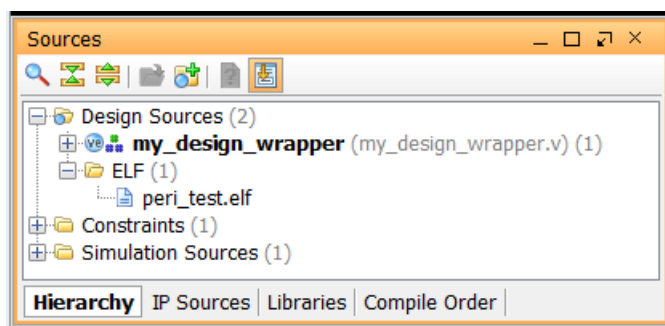


図 48 : ELF ファイルを含む [Sources] ビュー

7. 次にその ELF ファイルをマイクロプロセッサ デザインと関連付けます。これには、[Sources] ビューの [Design Sources] フォルダを右クリックし、[Associate ELF Files] をクリックします。

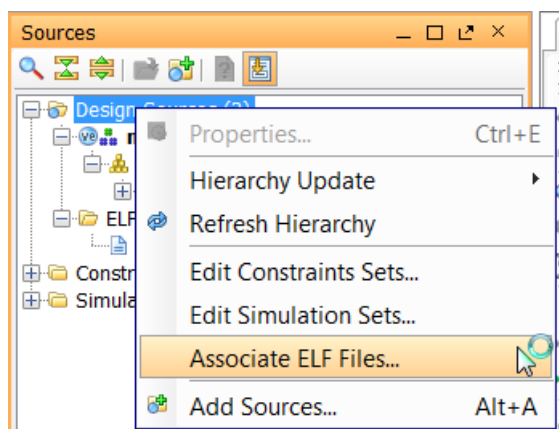


図 49 : [Associate ELF Files] コマンド

8. 合成およびシミュレーション用に ELF ファイルを追加できます。該当する参照アイコン ([Design Sources] または [Simulation Sources] の下) をクリックして、新しく追加した ELF ファイルを参照します。

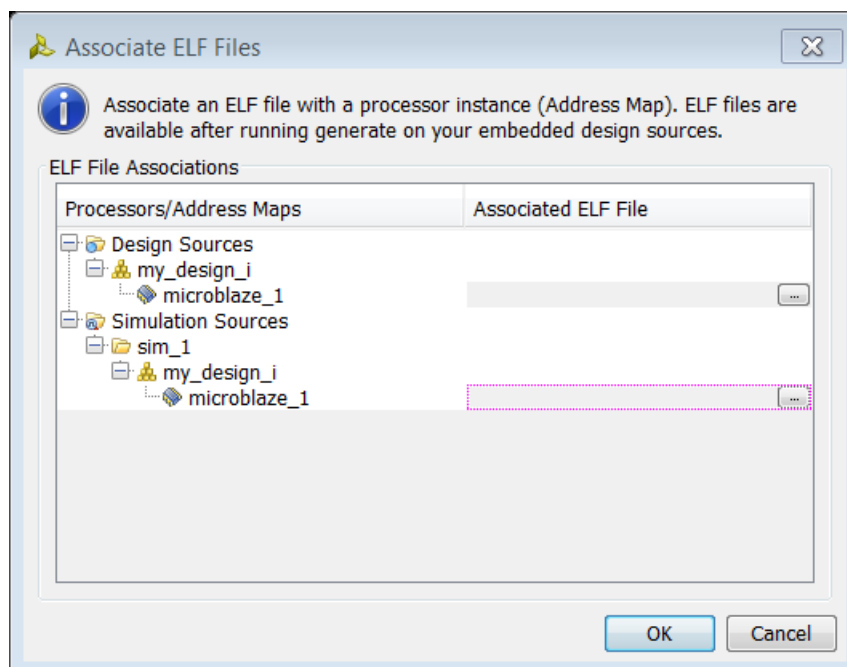


図 50 : ELF ファイルとマイクロプロセッサの関連付け

[Associate ELF Files] ダイアログ ボックスが開きます。ファイルをハイライトして、[OK] をクリックします。

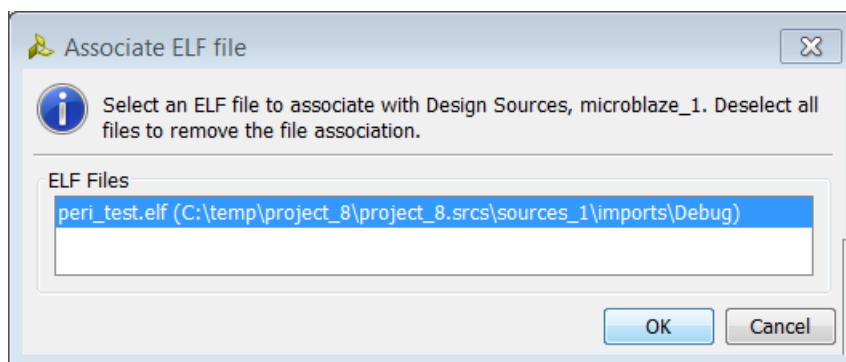


図 51 : 関連付ける ELF ファイルのハイライト

[Associated ELF File] 列に ELF ファイルが生成されたのを確認したら、[OK] をクリックします。

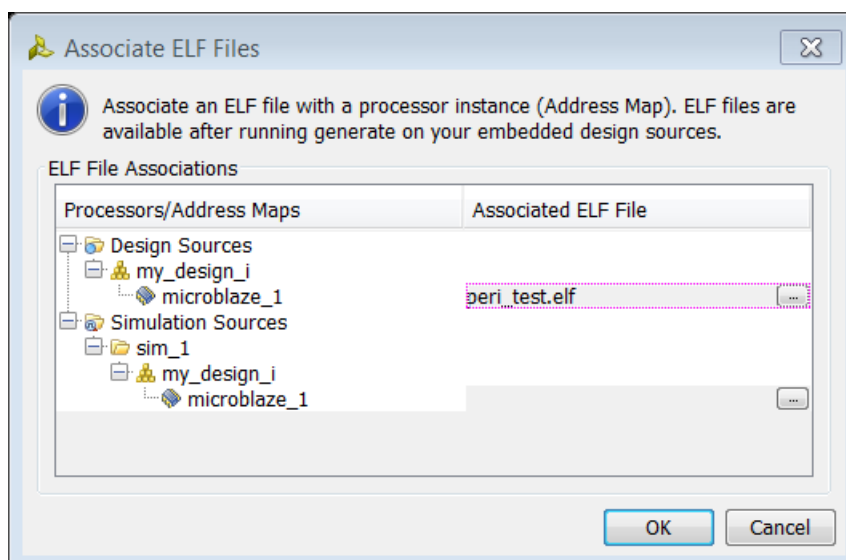


図 52 : ELF ファイルの確認