



XAPP1168 (v1.0) 2013 年 6 月 1 日

Vivado IP インテグレーターへのカスタム AXI IP の組み込み

著者 : Dylan Buli, Kyle Corbett

概要

このアプリケーション ノートでは、Vivado™ デザイン ツールを十分に活用できるように IP インテグレーターにカスタム AXI IP を組み込む方法について説明します。ここでは、AXI ベースの 6 種類の IP コアについて解説し、実際に Vivado 対応 IP としてこれらを組み込む手順を示します。また、IP インテグレーターが備える自動化機能の利用方法についても詳しく述べています。

このアプリケーション ノートに基づいて、AXI システム用のカスタム IP の作成を開始できます。次の 6 種類の IP について説明します。

- AXI4 マスター
- AXI4 スレーブ
- AXI4-Lite マスター
- AXI4-Lite スレーブ
- AXI4-Stream マスター
- AXI4-Stream スレーブ

はじめに

AXI は、AMBA (Advanced Microcontroller Bus Architecture) 4 仕様 [参照 2] に基づいて標準化された IP インターフェイス プロトコルです。リファレンス デザインでは、AXI4 仕様に記載されている AXI4、AXI4-Lite、および AXI4-Stream の各インターフェイスを使用しています。これらのインターフェイスは、システムを構築するための共通した IP インターフェイス プロトコルのフレームワークを提供します。

IP インテグレーター内でカスタム IP を使用するには、その IP を組み込む必要があります。Vivado デザイン ツール チェーンの一部として簡単に IP を組み込むことができるように、Vivado はそのシンプルな方法を提供しています。カスタム AXI IP の IP ポート名が AXI4 仕様に従っていれば、Vivado は自動的に AXI インターフェイスを推論できます。

Vivado は複数の方法で IP を組み込むことができますが、このアプリケーション ノートでは次の手順に従います。

1. 組み込む IP ごとに個別の Vivado プロジェクトを作成します。
2. 共通したポート命名規則を使用する、既存の RTL をインポートします。
3. Package IP Wizard を実行し、追加のメタデータおよびパラメーターを関連付けます。
4. 組み込んだ IP を .zip ファイルに圧縮し、リポジトリ ディレクトリに保存します。
5. 組み込んだ IP を使用する、システムレベルのプロジェクトを作成します。
6. リポジトリ ディレクトリをプロジェクト プリファレンスに追加し、IP をリポジトリに追加します。
7. IP インテグレーターのプロック図を作成し、組み込んだ IP のシステムへの追加と接続を行い、システムのシミュレーションを実行します。

リファレンス デザインの内容

付属の ZIP ファイルには、次が含まれます。

<https://secure.xilinx.com/webreg/clickthrough.do?cid=338300>

- **プロジェクトの Tcl**: source.tcl

この Vivado Tcl ファイルは、アプリケーション ノート通して使用されます。このファイルを使用するには、次の手順に従います。

- a. Vivado ツールを開きます。
- b. Vivado Tcl コンソールで、このファイルを展開したディレクトリに移動します。
例: `cd C:/designs/xapp1168`
- c. Vivado Tcl コンソールで、`source source.tcl` と入力します。

- **AXI4-Lite マスター**: hdl/verilog/axi_lite_master.v

このサンプル AXI4-Lite マスターは、固定されたデータ幅 (32 ビット) と固定されたアドレス幅 (32 ビット) を持つ AXI4-Lite インターフェイスを 1 つ備えています。この IP は、パラメーター (C_TRANSACTIONS_NUM) で指定された数のトランザクションを発行するように設計されています。リセット信号がデアサートされると、IP はアドレス 0x4000 から書き込みを開始し、順次アドレスへの書き込みを C_TRANSACTIONS_NUM ビート分続けます。書き込みが完了すると、WCOMPLETE 信号がアサートされます。次に、コアはアドレス 0x4000 から始まる読み出しを発行し、順次アドレスからの読み出しを C_TRANSACTIONS_NUM ビート分続けます。読み出し実行中に、同じアドレスに書き込まれた値を使用して読み出されたデータを確認します。このプロセスが完了すると、RCOMPLETE 信号がアサートされます。

AXI4-Lite マスター インターフェイスのポート名は、マスター AXI ポートを示す M_AXI_ で始まります。

- **AXI4-Lite スレーブ**: hdl/verilog/axi_lite_slave.v

このサンプル AXI4-Lite スレーブは、固定されたデータ幅 (32 ビット) と固定されたアドレス幅 (5 ビット) を持つ AXI4-Lite インターフェイスを 1 つ備えています。この IP は 32 バイトのみのメモリ アドレス空間を持ち、ほかのすべてのアクセスはエイリアスを使用して同じ 32 バイトを参照します。32 バイトのうち、下位 16 バイトだけにメモリ エレメントが含まれます。アドレス 0x10 ~ 0x1F へのアクセスは無効です。

このサンプル スレーブは、AW チャネルと W チャネルの両方が適切な VALID 信号を示すまで、いずれのチャネルも受け入れません。書き込みの実行後、コアは B チャネル ハンドシェイクを発行します。

AXI4-Lite スレーブ インターフェイスのポート名は、スレーブ AXI ポートを示す S_AXI_ で始まります。

- **AXI4 マスター**: hdl/verilog/axi_master.v

このサンプル AXI4 マスターは、固定されたアドレス幅 (40 ビット) と 64 ビットまたは 32 ビットにプログラム可能なデータ幅を持つ AXI4 インターフェイスを 1 つ備えています。この IP は、パラメーター (C_TRANSACTIONS_NUM) で指定された数の転送を発行するように設計されています (各転送の長さは C_BURST_LEN)。リセット信号がデアサートされると、IP はアドレス 0x80000 から書き込みを開始し、順次アドレスへの長さ C_BURST_LEN ビートの転送を C_TRANSACTIONS_NUM 回続けます。書き込みが完了すると、WCOMPLETE 信号がアサートされます。次に、コアはアドレス 0x80000 から始まる読み出しを発行し、順次アドレスからの長さ C_BURST_LEN ビートの転送を C_TRANSACTIONS_NUM 回続けます。読み出し実行中に、同じアドレスに書き込まれた値を使用して読み出されたデータを確認します。このプロセスが完了すると、RCOMPLETE 信号がアサートされます。

AXI4 マスター インターフェイスのポート名は、マスター AXI ポートを示す M_AXI_ で始まります。

- **AXI4 スレーブ** : hdl/verilog/axi_slave.v

このサンプル AXI4 スレーブは、固定されたアドレス幅 (14 ビット) と 64 ビットまたは 32 ビットにプログラム可能なデータ幅を持つ AXI4 インターフェイスを 1 つ備えています。この IP は 16K バイトのメモリ アドレス空間を持ち、そのうち 2K バイト (64 ビットの場合) または 1K バイト (32 ビットの場合) だけにメモリ エLEMENT が含まれます。IP のすべてのメモリ アクセスはエイリアスを使用してこのメモリ エLEMENT を参照します。

このサンプル スレーブは、AW チャネルと W チャネルの両方が適切な VALID 信号を示すまで、いずれのチャネルも受け入れません。書き込みの実行後、コアは B チャネル ハンドシェイクを発行します。

AXI4 スレーブ インターフェイスのポート名は、スレーブ AXI ポートを示す S_AXI_ で始まります。

- **AXI4-Stream マスター** : hdl/verilog/axi_stream_master.v

このサンプル AXI4-Stream マスターは、1 ~ 64 バイトの範囲でプログラム可能なデータ幅を持つ AXI4-Stream インターフェイスを 1 つ備えています。IP は、長さ C_PACKET_LENGTH のトランザクションを続けて発行するように設計されています。リセット信号がデアサートされると、IP は、各トランザクションの長さが C_PACKET_LENGTH ビートのトランザクションの発行を開始します。トランザクションが完了すると、IP は任意のサイクル数待機し、長さ C_PACKET_LENGTH の新しいトランザクションを発行します。

AXI4-Stream マスター インターフェイスのポート名は、マスター AXI4-Stream ポートを示す M_AXIS_ で始まります。

- **AXI4-Stream スレーブ** : hdl/verilog/axi_stream_slave.v

このサンプル AXI4-Stream スレーブは、1 ~ 64 バイトの範囲でプログラム可能なデータ幅を持つ AXI4-Stream インターフェイスを 1 つ備えています。この IP は、S_AXIS_TREADY ポートに適用される擬似ランダム バックオフを提供するように設計されています。

AXI4-Stream スレーブ インターフェイスのポート名は、スレーブ AXI4-Stream ポートを示す S_AXIS_ で始まります。

テストベンチ

組み込んだ IP コアのインスタンスエーションの実例を示すために、3 種類のサンプル ブロック図が作成されています。次のテストベンチ ファイルは、これらのデザインに対する簡単なシミュレーション ステミュラスとして機能します。ZIP ファイルには次のテストベンチが含まれています。

- tb/verilog/lite_system_wrapper.v
- tb/verilog/axi_system_wrapper.v
- tb/verilog/axi_stream_system_wrapper.v

AXI4 IP の組み込み

このセクションでは、6 種類の AXI4 IP (AXI4、AXI4-Lite、AXI4-Stream それぞれのマスターとスレーブ) の組み込み手順を詳しく説明します。システムで IP ブロックを再利用できるように、Vivado Design Suite を使用して IP ブロックを組み込む方法について解説します。

注記 : 掲載されている図には、手順に記載されているもの以外のコンフィギュレーションや GUI のオプションが示されていることがあります。このような場合、特に設定変更の指示がなければ、デフォルト値のままにしてください。

各タイプの AXI4 IP の組み込み手順はほぼ同じです。手順が異なる場合は、IP のタイプごとに変更点が明記されています。

Vivado ツールの新規プロジェクトの開始とプロジェクト オプションの設定

このセクションでは、RTL IP を作成し、組み込むために各コアの Vivado プロジェクト作成する手順を詳しく説明します。

1. Vivado Design Suite (Logic Edition 以上) をインストールします (IP インテグレーターのライセンスが必要)。
2. リファレンス デザイン ファイルをローカル フォルダー (以降は <design_dir> と表記) に解凍します。
3. 次の手順に従って、Vivado ツールを開きます。
 - Windows の場合 : [スタート] → [Xilinx Design Tools] → [Vivado 2013.1] → [Vivado] をクリックします。
 - Linux の場合 : ザイリンクス Vivado デザイン ツールのセットアップ後、Linux に **vivado** コマンドを入力します。
4. Getting Started セクションの [Create New Project] をクリックして、新規プロジェクトを作成します。
5. [New Project] で [Next] をクリックします。
6. 各 IP に個別のプロジェクトが必要です。[Project Name] に適切な名前を入力します。
 - AXI4-Lite スレーブ : package_axi_lite_slave
 - AXI4-Lite マスター : package_axi_lite_master
 - AXI4 マスター : package_axi_master
 - AXI4 スレーブ : package_axi_slave
 - AXI4-Stream マスター : package_axi_stream_master
 - AXI4-Stream スレーブ : package_axi_stream_slave

新規プロジェクト用に選択したディレクトリは <user_dir> と表記されます。[Next] をクリックします。図 1 では、例として package_axi_lite_slave を使用しています。

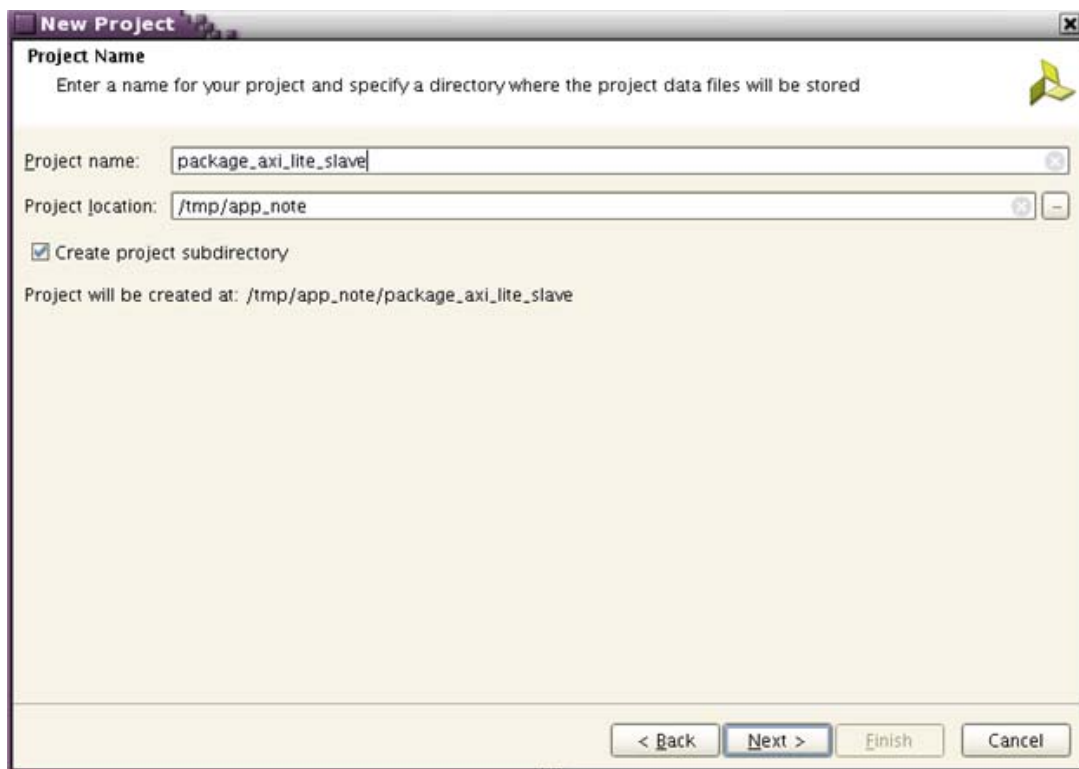


図 1 : [New Project] : [Project Name] ページ

7. [Project Type] で [RTL Project] をオンにします。[Next] をクリックします。
8. [Add Files] をクリックして、プロジェクトにソースを追加します。
9. <design_dir>/hdl/verilog ディレクトリに 6 つの Verilog ファイルがあります。
 - AXI4-Lite スレーブ : axi_lite_slave.v
 - AXI4-Lite マスター : axi_lite_master.v
 - AXI4 マスター : axi_master.v
 - AXI4 スレーブ : axi_slave.v
 - AXI4-Stream マスター : axi_stream_master.v
 - AXI4-Stream スレーブ : axi_stream_slave.v

組み込む IP のタイプに合ったソース ファイルを選択します。図 2 の例は AXI4-Lite スレーブを組み込む例を示しており、axi_lite_slave.v ファイルが選択されています。

ファイル名の選択後、[OK] をクリックします。

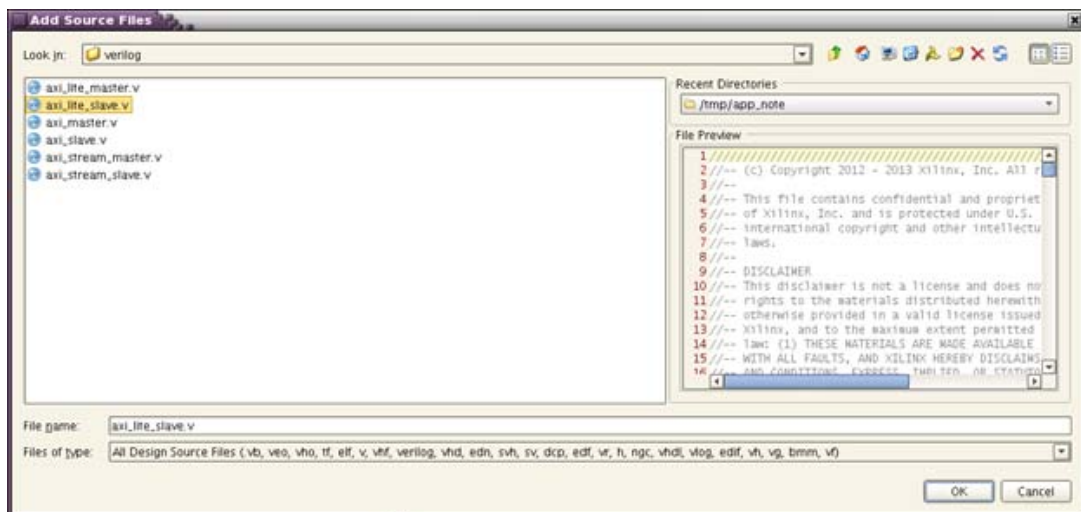


図 2 : [Add Source Files] : axi_lite_slave.v ソース ファイルの追加

ファイル axi_lite_slave.v が新規プロジェクトのソースとしてリストされます (図 3)。

10. [Next] をクリックします。

注記：ファイル名は組み込む IP のタイプに対応している必要があります。

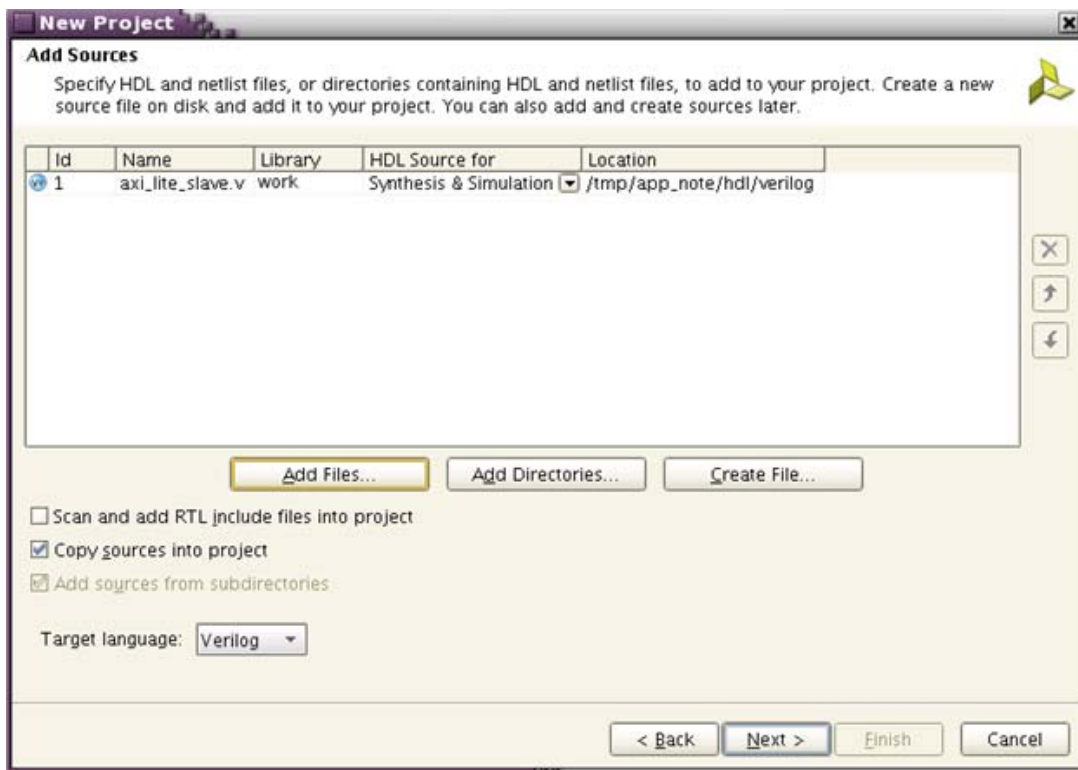


図 3 : [New Project] : [Add Sources] ページの axi_lite_slave.v

11. [Add Existing IP] で [Next] をクリックします。

12. [Add Constraints] で [Next] をクリックします。

13. Vivado Design Suite で選択されたデバイスのまま、[Default Part] ページで [Next] をクリックします (図 4)。

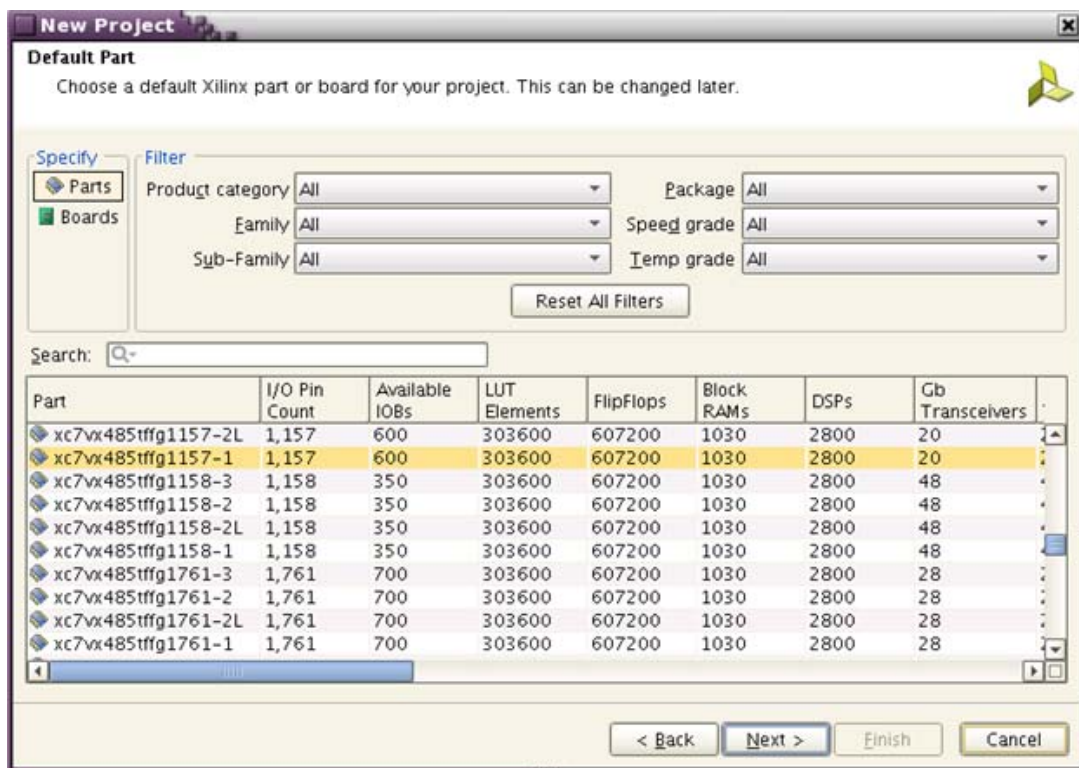


図 4 : [New Project] : [Default Part] ページ

[New Project Summary] ページが表示されます。図 5 に AXI4-Lite スレーブ IP のサマリ ページを示します。ほかの IP のプロジェクト作成手順との違いは、作成されるプロジェクト名だけです。

14. ソースファイルが 1 つ追加されることを確認して、[Finish] をクリックします。

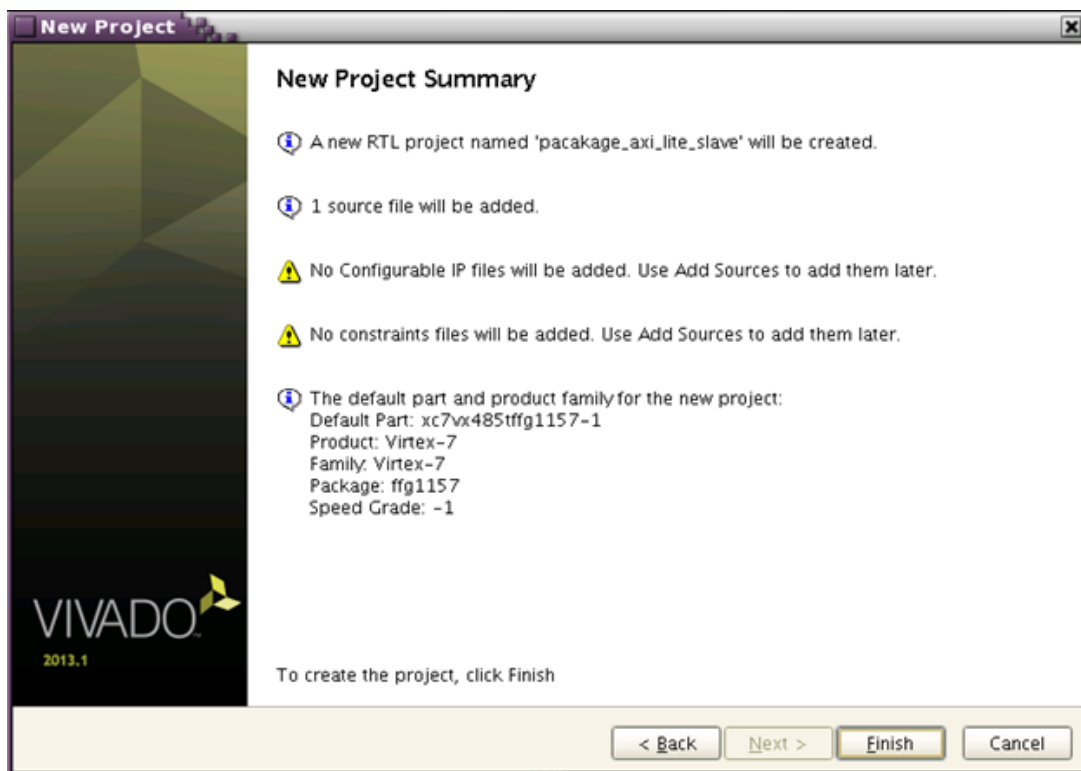


図 5 : [New Project] : AXI4-Lite スレーブの [New Project Summary] ページ

Vivado プロジェクトの組み込み

Vivado Design Suite では、開いている Vivado プロジェクトを組み込んだ IP に簡単に交換できます。次の手順でこのプロセスを詳しく説明します。

1. [Tools] → [Package IP Wizard] をクリックします (図 6)。

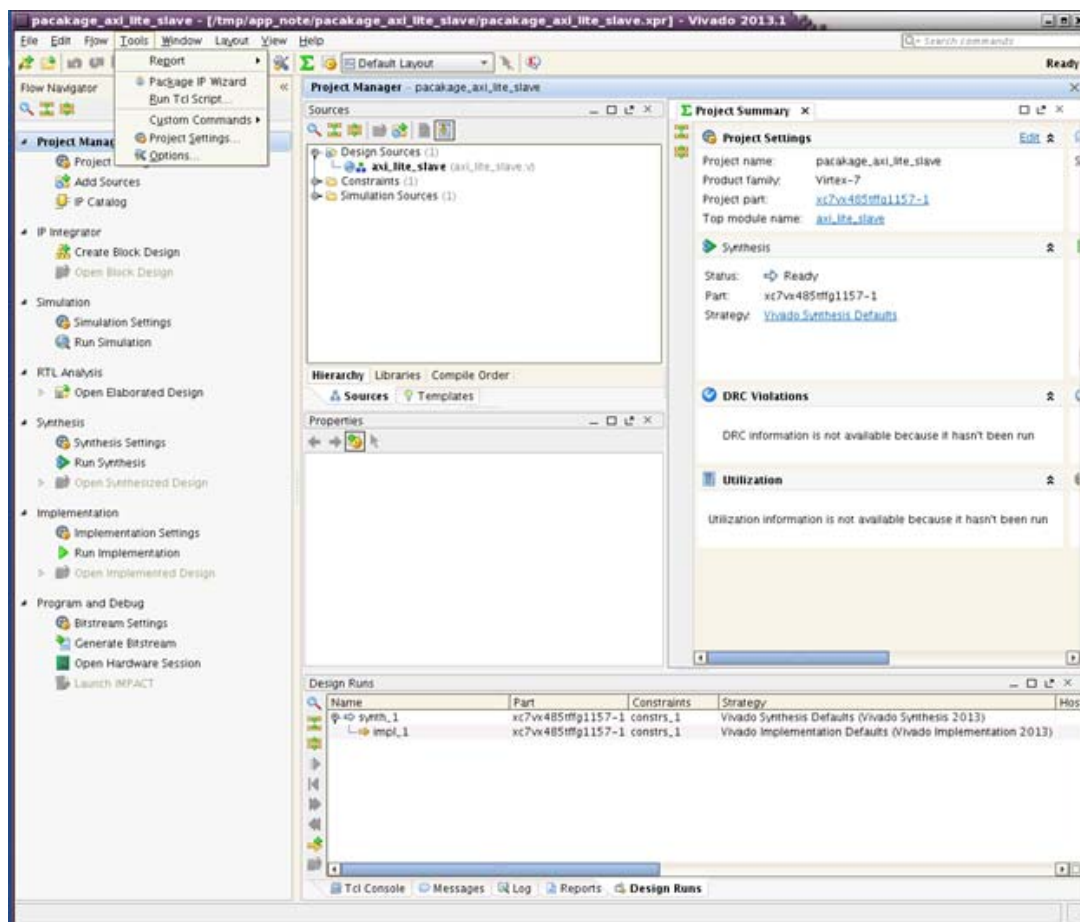


図 6 : [Tools] → [Package IP Wizard] を選択

2. [Package New IP] で [Next] をクリックします。
3. [Choose IP Source Location] ページで [Package your project] をオンにし、[Next] をクリックします (図 7)。

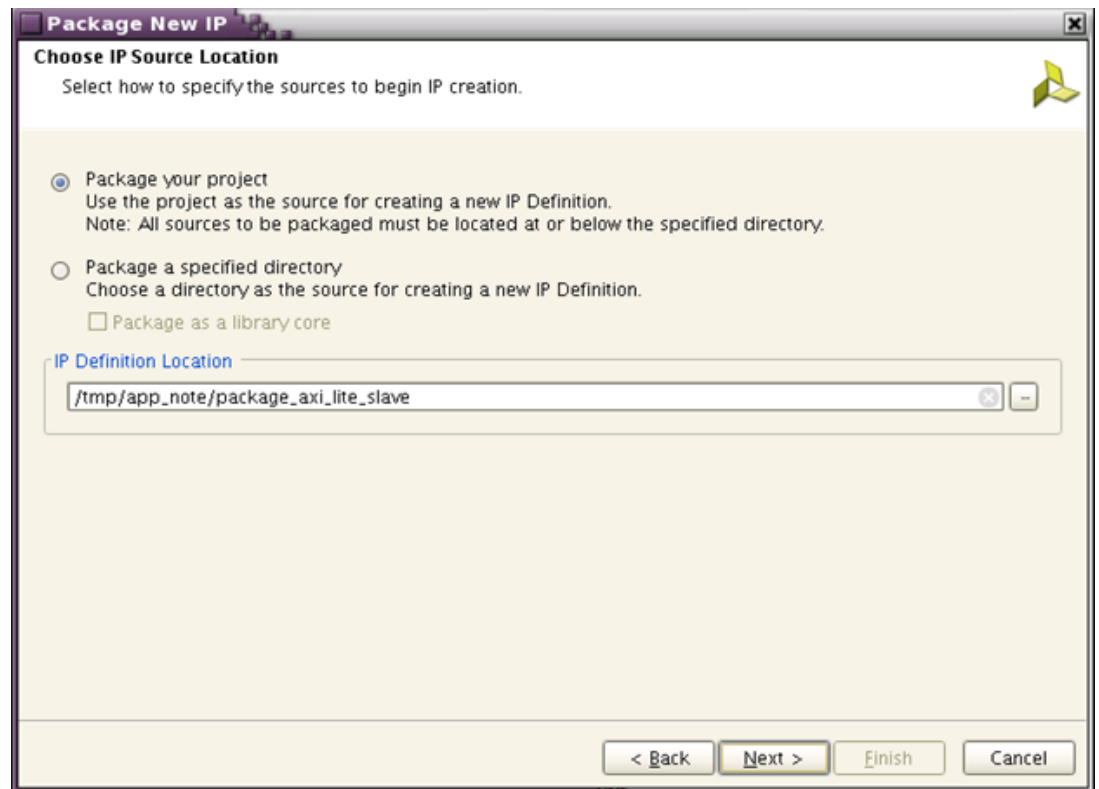


図 7 : [Package New IP] : [Choose IP Source Location] ページ

4. [Begin IP Creation] ページが表示されます。[Finish] をクリックします。
5. ウィザードでの処理が終了すると、[IP Packager Summary] が表示されます。ここには、HDL からインポートされたインターフェイス、ポート、およびパラメーターの数が詳しく表示されます。[OK] をクリックします。図 8 に AXI4-Lite スレーブ IP の結果を示します。各フィールドの値は IP ごとに異なります。

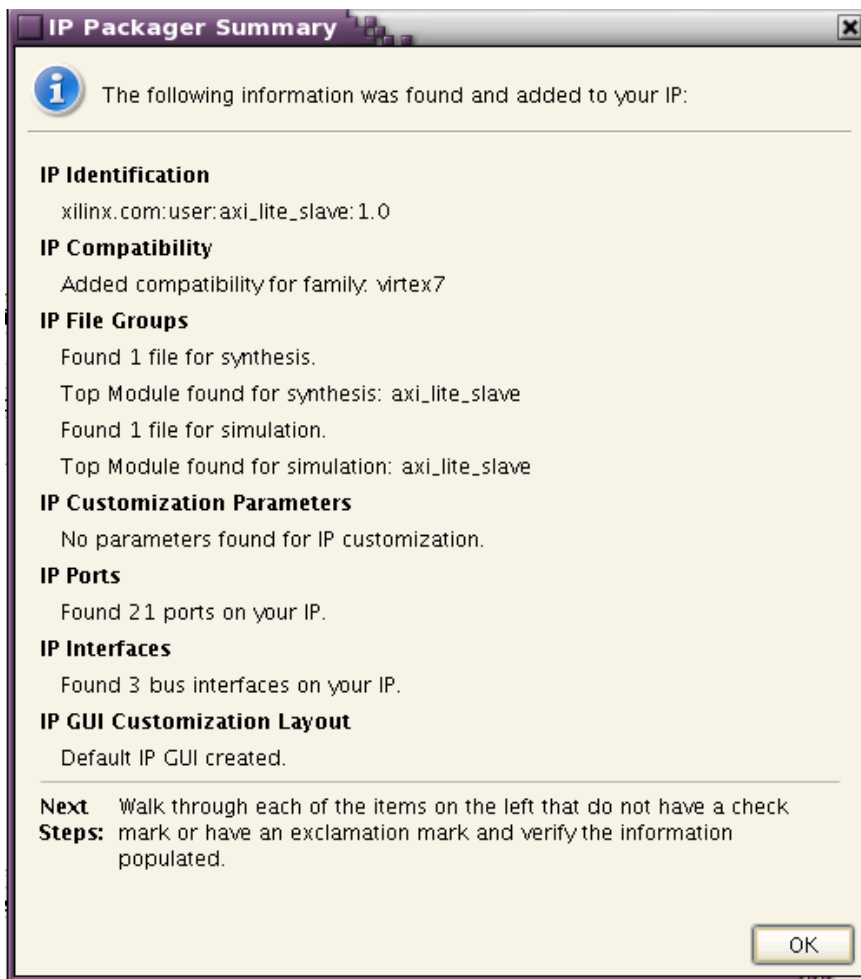


図 8 : AXI4Lite スレーブの [IP Packager Summary]

自動生成された IP 定義の更新

プロジェクトの RTL がインポートされ、ウィザードによって IP の特性が判断されます。インポートが成功した場合は、[Package IP - <IP_NAME>] というビューが [Project Summary] ビューの横に表示されます。図 9 に AXI4-Lite スレーブ IP の結果を示します。[Name]、[Display Name]、[Description] の値は、組み込んだ IP ごとに異なります。

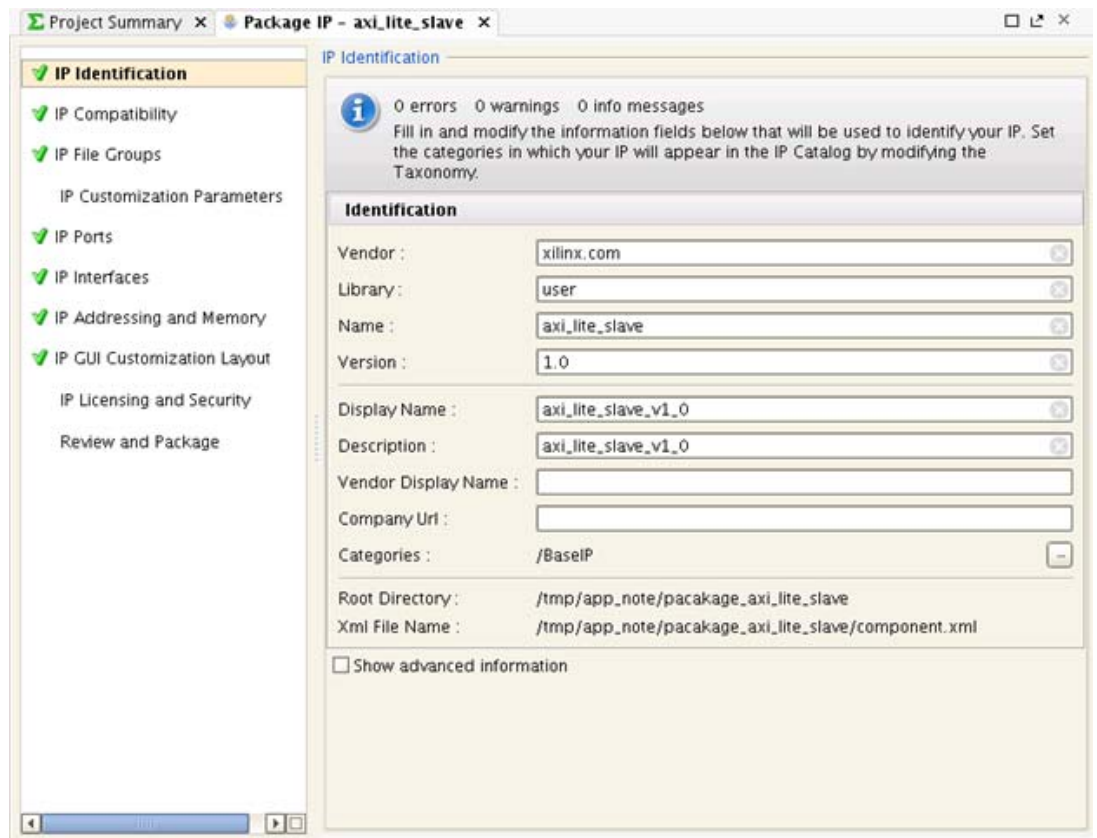


図 9 : [Package IP] ビュー

6. [IP Identification] ページで次のフィールドを設定し、必要に応じて値を置き換えます。
- Vendor : 会社名.com
 - Vendor Display Name : 会社名
 - Company Url : ウェブサイトの URL

図 10 に AXI4-Lite スレーブの結果を示します。

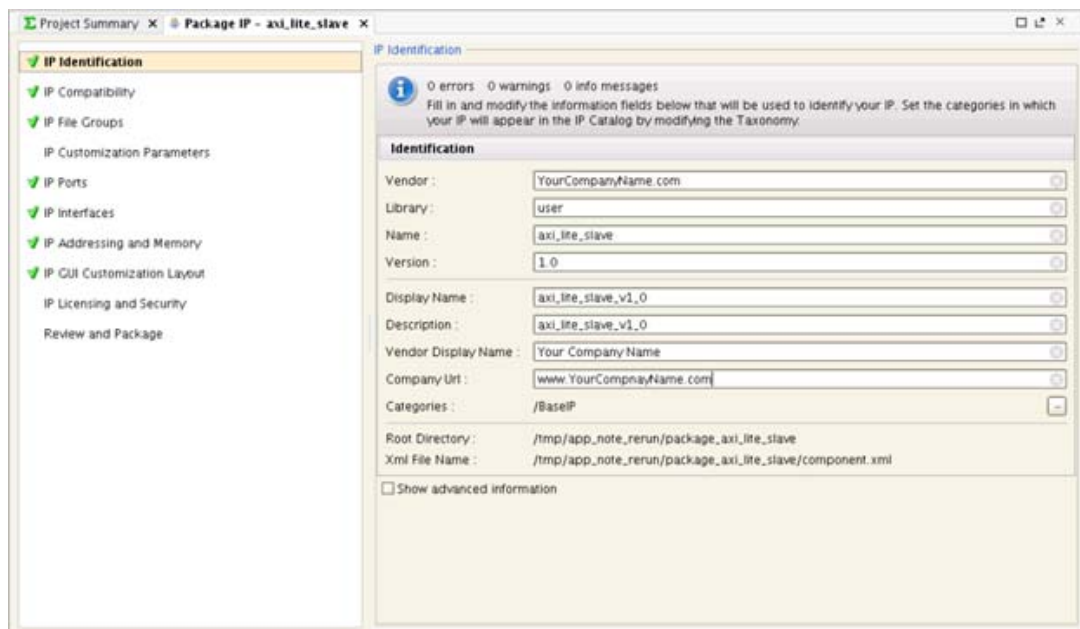


図 10 : ベンダー カスタマイズ後の [Package IP] ビュー

7. IP カタログの特定のカテゴリに IP が表示されるようにするには、それらのカテゴリの一部になるように IP を構成する必要があります。IP カタログ内で IP が表示されるカテゴリを変更するには、[Categories] の行にある [...] をクリックします。[Choose IP Categories] が表示されます (図 11)。

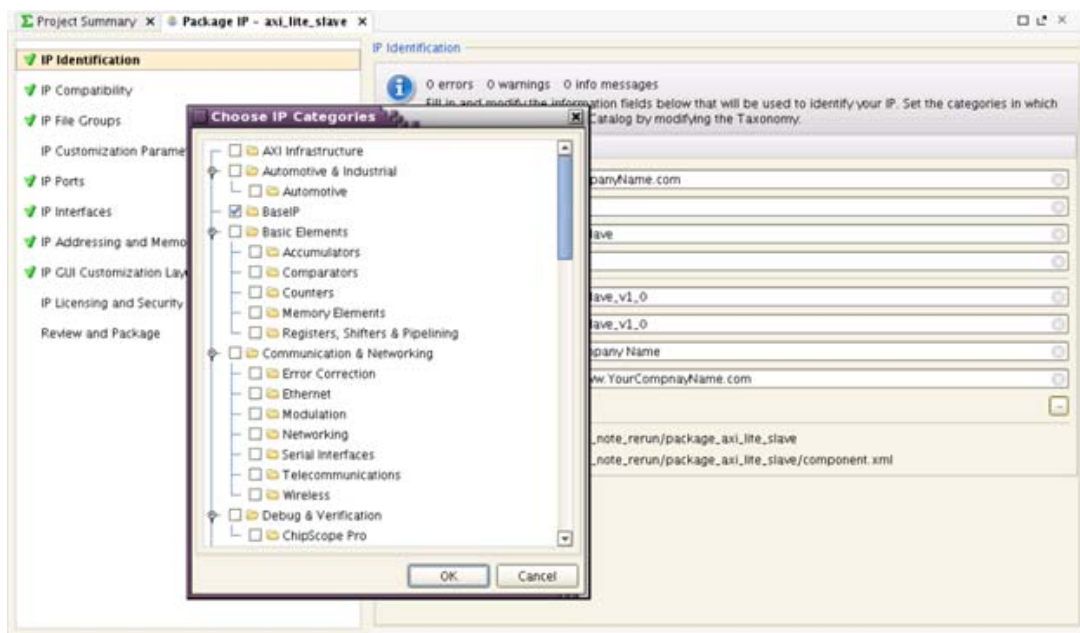


図 11 : [Choose IP Categories]

[Choose IP Categories] で次のように設定し、[OK] をクリックします。

- [BaseIP] をオフにします。
- [AXI Infrastructure] をオンにします。

8. [IP Compatibility] をクリックします。

この IP がサポートするザイリンクス FPGA ファミリが表示されます。プロジェクトで設定されているファミリが表示されます。これを変更するには、[Family Support] の表で右クリックし、メニューから [Add Family] を選択します (図 12)。IP コアの [Family Support] リストに表示されないデバイスでは、その IP は利用できません。

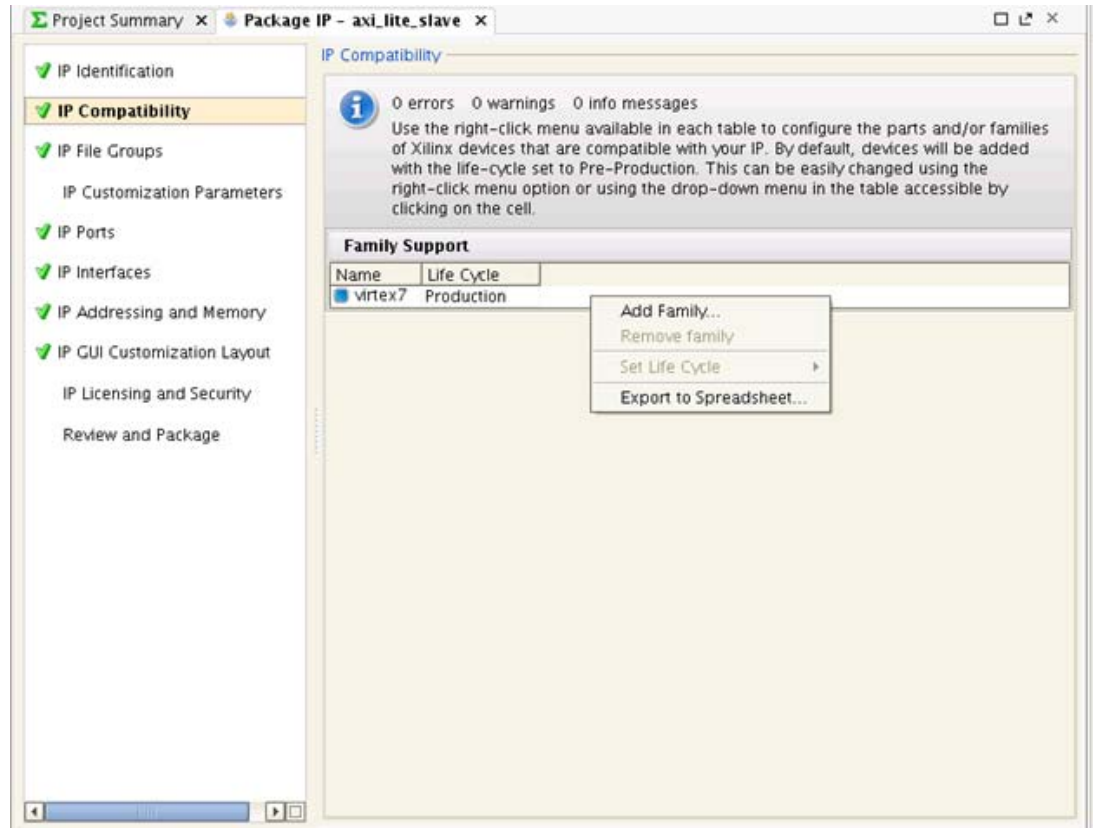


図 12 : [IP Compatibility] : [Add Family] メニュー

9. [Choose Family Support] で [kintex7] をオンにし、[OK] をクリックします (図 13)。

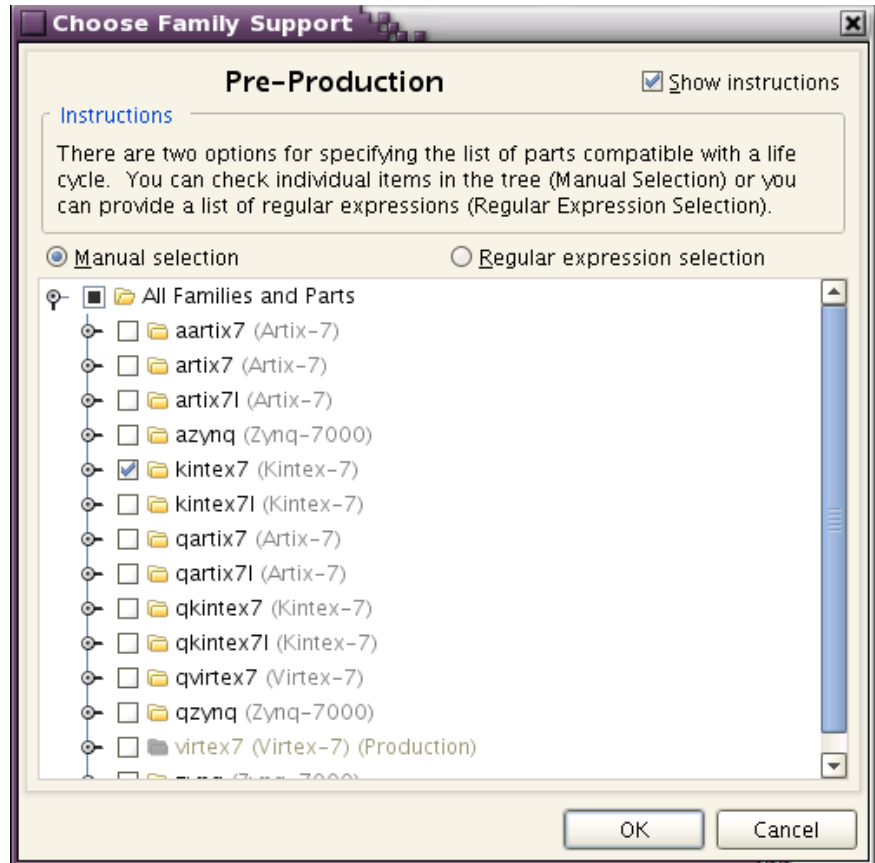


図 13 : [Choose Family Support]

10. [IP Ports] をクリックします。インポートされた Verilog のすべてのポートに使用される命名規則に注意してください。「S_AXI」などのインターフェイス ラベルの後に、標準 AXI 信号名および [Size Left] 列のベクター幅が続きます。

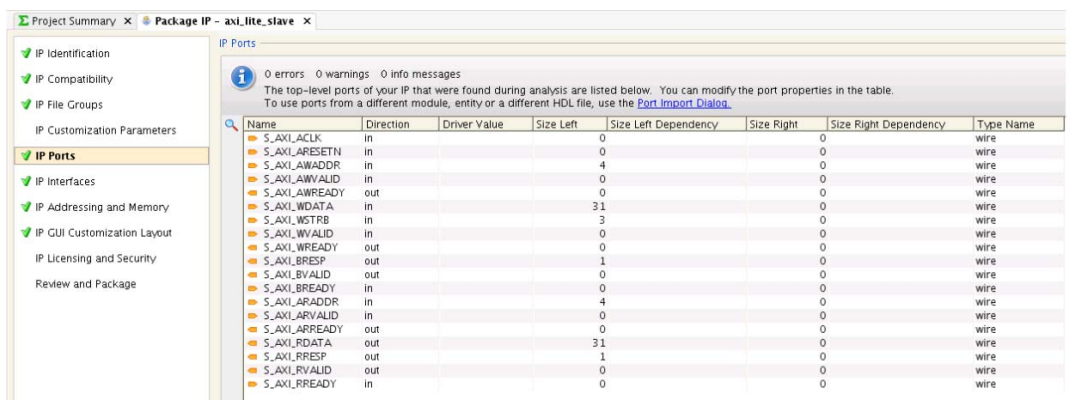


図 14 : [IP Ports] ページ

11. [IP Interfaces] をクリックします。自動的に推論されるクロック、リセット、および AXI インターフェイスに注意してください。AXI インターフェイスは、名前に一貫性のある AXI 信号を集め、抽象化されたユニットとして扱うことができる 1 つのグループにまとめます。これにより、デザイン内のほかの AXI インターフェイスへの接続が簡単になります。

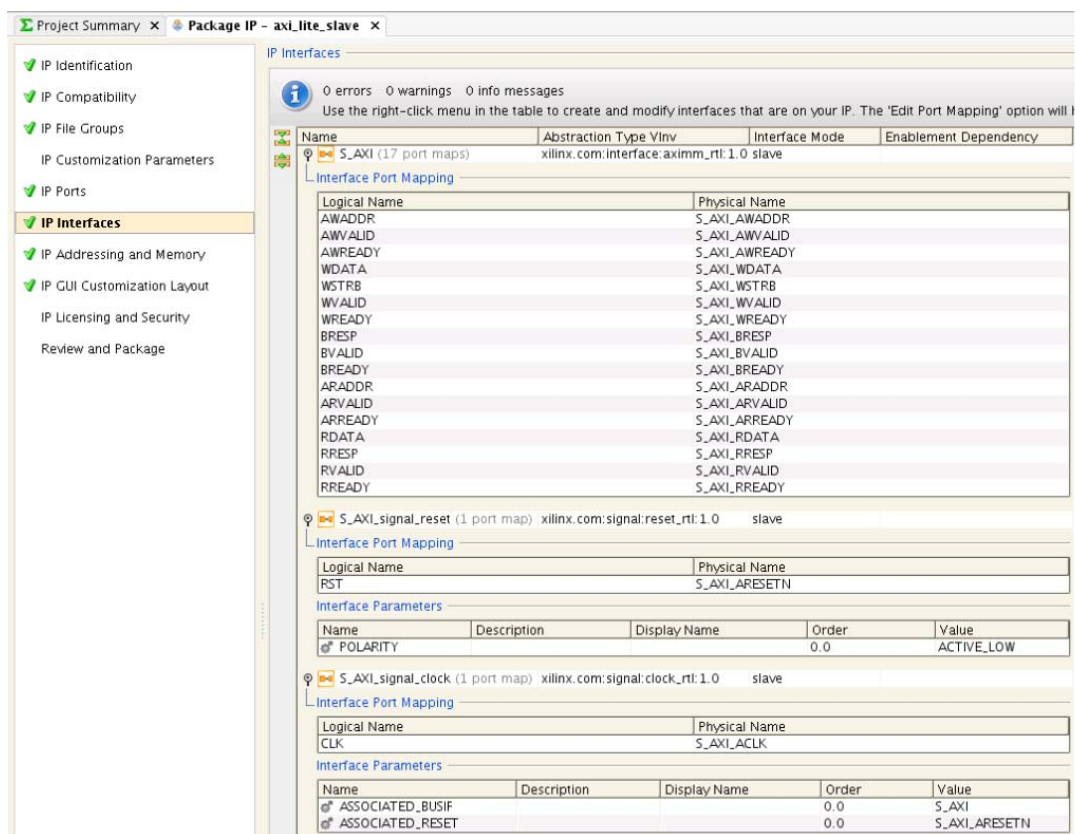


図 15 : [IP Interfaces] ページ

組み込みプロセスのこの段階では、IP ごとにカスタマイズの手順が異なります。次のいずれかを選択してカスタマイズを行い、同じ手順を繰り返して、その他のタイプの IP をカスタマイズしてください。

- [AXI4-Lite スレーブ IP のカスタマイズ](#)
- [AXI4-Lite マスター IP のカスタマイズ](#)
- [AXI4 マスター IP のカスタマイズ](#)
- [AXI4 スレーブ IP のカスタマイズ](#)
- [AXI4-Stream マスター IP のカスタマイズ](#)
- [AXI4-Stream スレーブ IP のカスタマイズ](#)

AXI4-Lite スレーブ IP のカスタマイズ

AXI4-Lite スレーブ IP は、S_AXI インターフェイス上に 5 ビットの固定されたアドレス幅を持つ AxADDR ポートを備えています。

1. [IP Addressing and Memory] をクリックします。[Range] フィールドの 32 は、IP 内のアドレス指定可能なバイト数です (図 16)。これはペリフェラルが割り当てられる最小アドレス空間です。

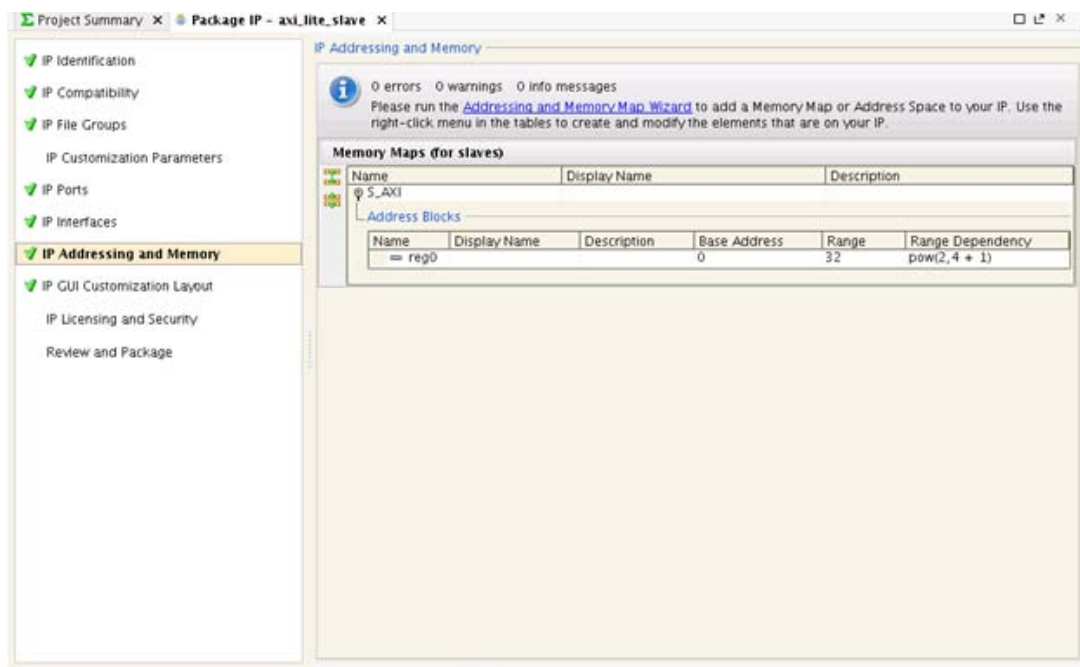


図 16 : AXI4-Lite スレーブの [IP Addressing and Memory] ページ

2. 「確認と組み込み」の[手順 1](#)に進みます。

AXI4-Lite マスター IP のカスタマイズ

AXI4-Lite マスター IP をカスタマイズすると、この HDL パラメーターを示す IP のコンフィギュレーション GUI の見た目が変わります。この IP を使用する場合は、ユーザーが入力できる値の範囲に制限が加えられます。

1. [IP Customization Parameters] をクリックします。[User Parameters] 表で、C_TRANSACTIONS_NUM の次のように各フィールドを更新します ([図 17](#))。
 - [Description] : Number of Transactions
 - [Display Name] : Number of Transactions
 - [Maximum] : 64
 - [Minimum] : 1

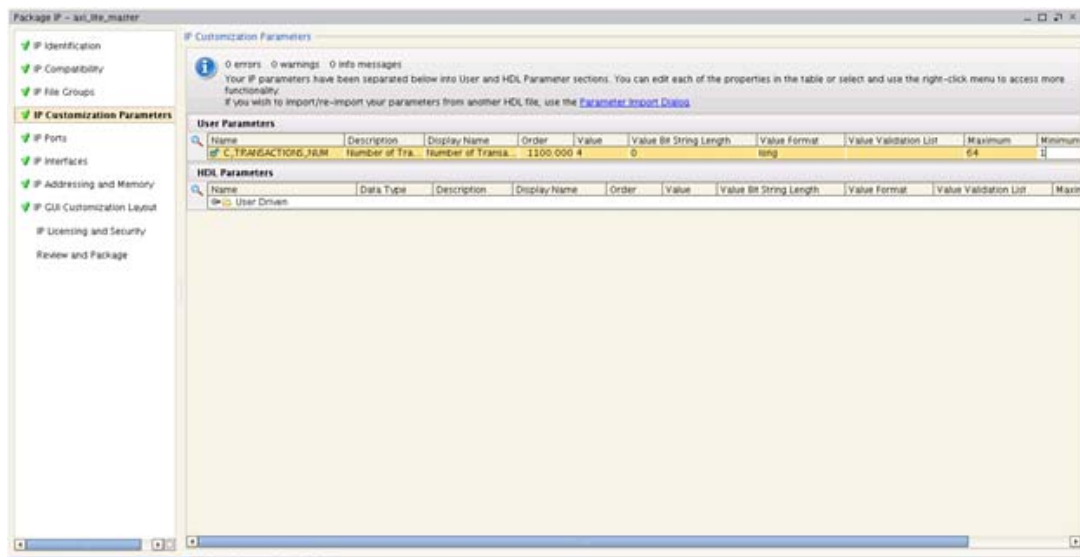


図 17 : AXI4-Lite マスターの [IP Customization Parameters] ページ

2. [IP Addressing and Memory] をクリックします。AXI4-Lite マスター IP は最大 4G バイトのアドレスにアクセスできます。この値は図 18 の [Range] フィールドに表示されます。

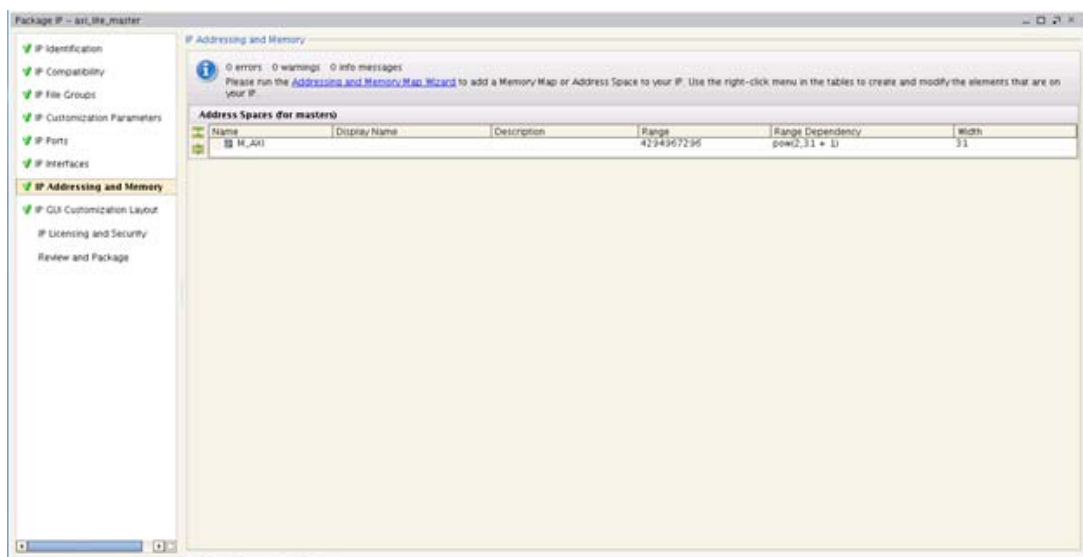


図 18 : AXI4-Lite マスターの [IP Addressing and Memory] ページ

3. 「確認と組み込み」の **手順 1** に進みます。

AXI4 マスター IP のカスタマイズ

1. [IP Customization Parameters] をクリックします。
2. 表内で C_M_AXI_DATA_WIDTH の値をクリックします。右クリックして [Edit Parameter] を選択します (図 19)。

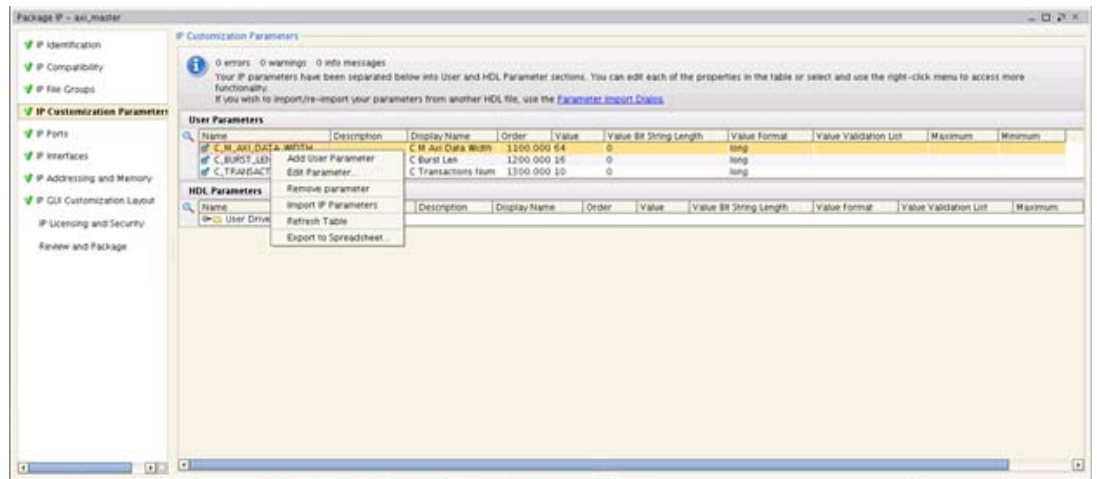


図 19 : AXI4 マスター IP の [IP Customization Parameters] ページ、C_M_AXI_DATA_WIDTH

3. [Edit Parameter] で次のように各フィールドを更新し (図 20)、[OK] をクリックします。これらのフィールドによって GUI が更新され、AXI データ幅のパラメーターの選択肢が 2 つに限られます。
- [Display name] : Data Width
 - [Should the value be restricted to a list or range?] : Yes
 - [Simple list] をオンにし、Instructions の説明に従って 32 と 64 を入力します。

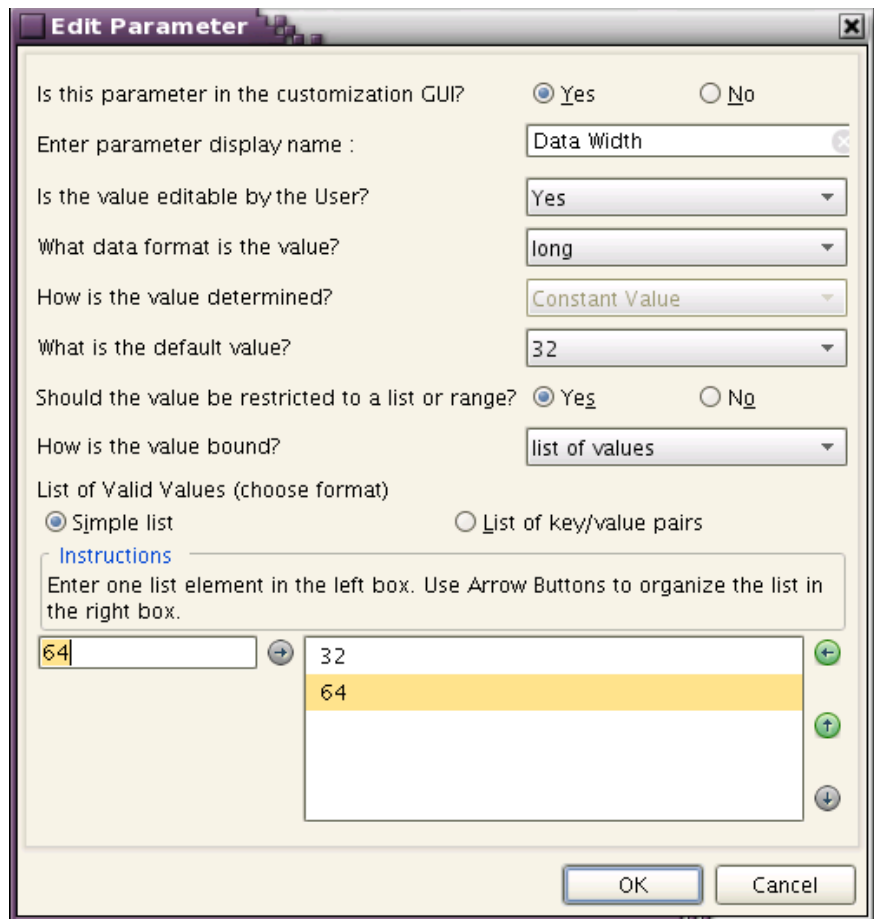


図 20 : AXI4 マスターの [Edit Parameter]

4. C_BURST_LEN ユーザー パラメーターをクリックし、次のように値を変更します。

- [Description] : Burst Length
 - [Display Name] : Burst Length
 - [Maximum] : 256
 - [Minimum] : 1
5. C_TRANSACTIONS_NUM ユーザー パラメーターをクリックし、次のように値を変更します。
 - [Description] : Number of Transactions
 - [Display Name] : Number of Transactions
 - [Maximum] : 64
 - [Minimum] : 1
 6. [IP Addressing and Memory] をクリックします。AXI4 マスターは、M_AXI インターフェイス上に 40 ビットの AxADDR ポートを備えています (図 21)。

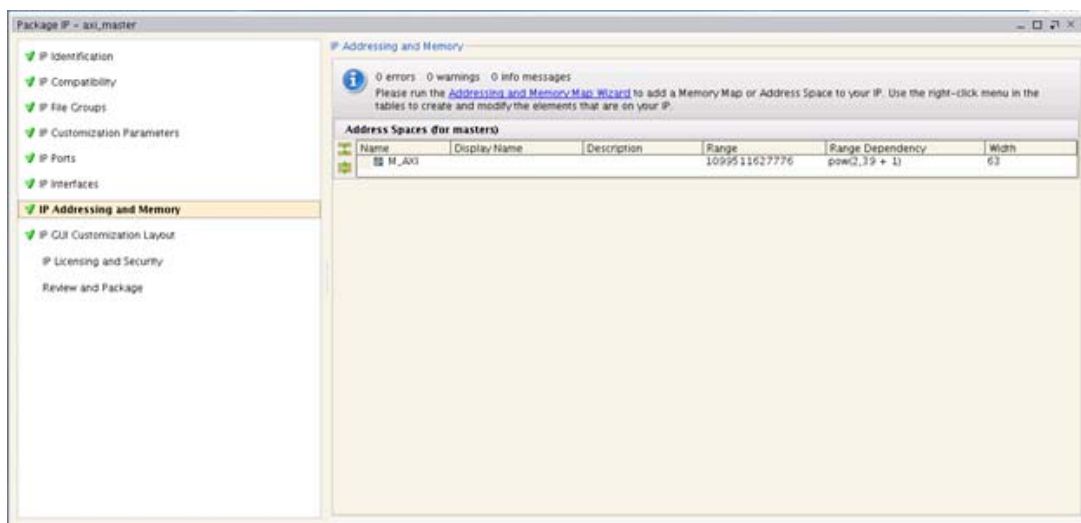


図 21 : AXI4 マスターの [IP Addressing and Memory] ページ

7. 「確認と組み込み」の [手順 1](#) に進みます。

AXI4 スレーブ IP のカスタマイズ

1. [IP Customization Parameters] をクリックします。
2. 表内で C_S_AXI_DATA_WIDTH の値をクリックします。右クリックして [Edit Parameter] を選択します (図 22)。

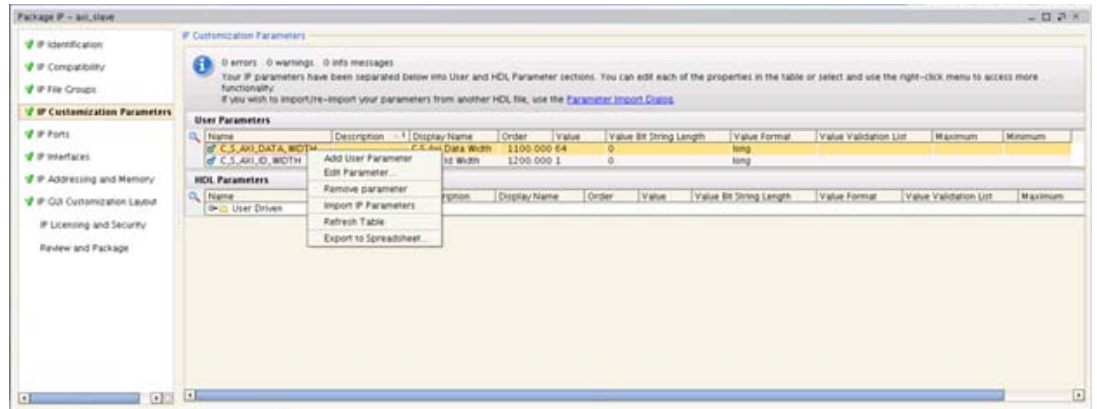


図 22 : AXI4 スレーブ IP の [IP Customization Parameters] ページ、C_S_AXI_DATA_WIDTH

- [Edit Parameter] で次のように各フィールドを更新し (図 23)、[OK] をクリックします。これらのフィールドによって GUI が更新され、AXI データ幅のパラメーターの選択肢が 2 つに限られます。
 - [Display name] : Data Width
 - [Should the value be restricted to a list or range?] : Yes
 - [Simple list] をオンにし、Instructions の説明に従って 32 と 64 を入力します。

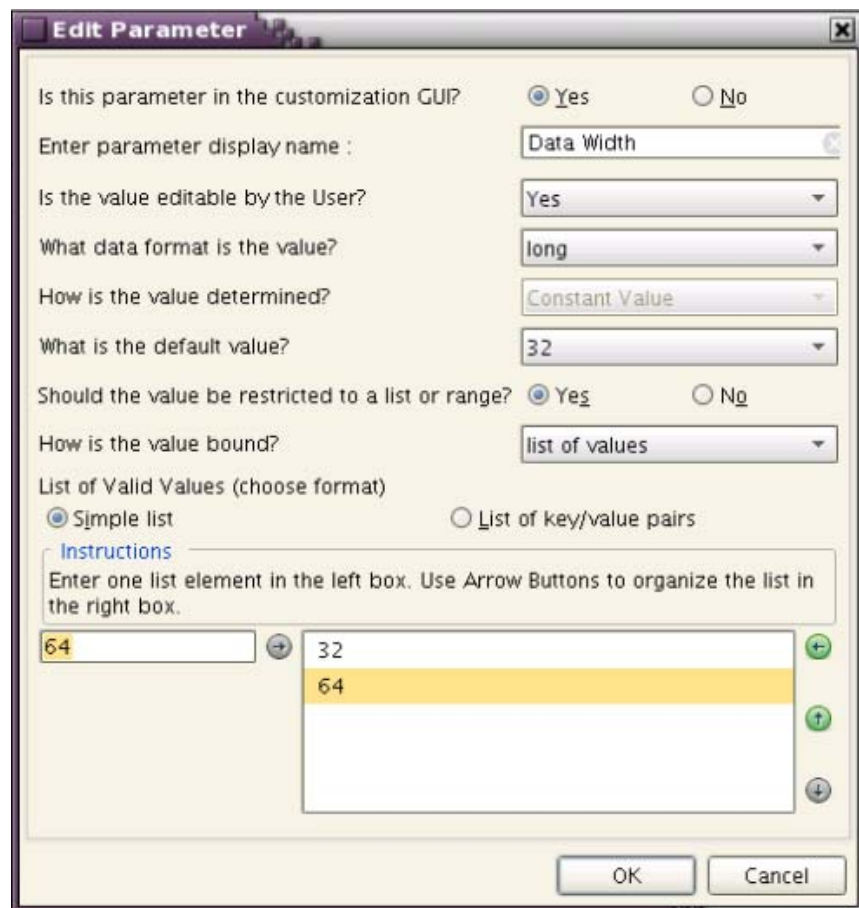


図 23 : AXI4 スレーブの [Edit Parameter]

- C_S_AXI_ID_WIDTH ユーザー パラメーターをクリックし、次のように値を変更します。
 - [Description] : ID Width
 - [Display Name] : ID Width

- [Maximum] : 32
 - [Minimum] : 1
5. [IP Addressing and Memory] をクリックします。AXI4 スレーブは、S_AXI インターフェイス上に 14 ビットの AxADDR ポートを備えています (図 24)。



図 24 : AXI4 スレーブの [IP Addressing and Memory] ページ

6. 「確認と組み込み」の [手順 1](#) に進みます。

AXI4-Stream マスター IP のカスタマイズ

1. [IP Customization Parameters] をクリックします。
2. C_PACKET_LENGTH ユーザー パラメーターをクリックし、次のように値を変更します。
 - [Description] : Packet Length
 - [Display Name] : Packet Length
 - [Maximum] : 64
 - [Minimum] : 1
3. C_M_AXIS_TDATA_NUM_BYTES ユーザー パラメーターをクリックし、次のように値を変更します。
 - [Description] : Data Width in bytes
 - [Display Name] : Data Width in bytes
 - [Maximum] : 512
 - [Minimum] : 1

[図 25](#) に最終的な結果を示します。

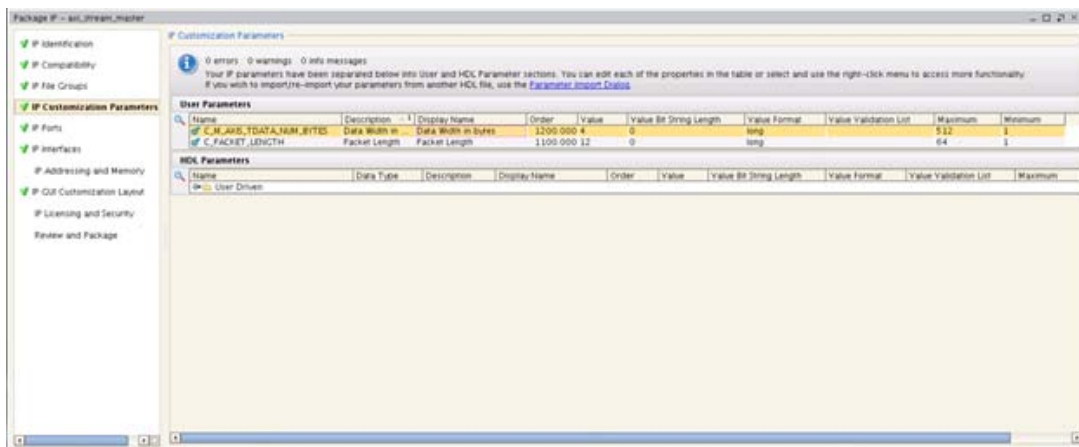


図 25 : AXI4-Stream マスターの [IP Customization Parameters] ページ

4. 「確認と組み込み」の [手順 1](#) に進みます。

AXI4-Stream スレーブ IP のカスタマイズ

- [IP Customization Parameters] をクリックします。
- C_S_AXIS_TDATA_NUM_BYTES ユーザー パラメーターをクリックし、次のように値を変更します。
 - [Description] : Data Width in bytes
 - [Display Name] : Data Width in bytes
 - [Maximum] : 512
 - [Minimum] : 1

[図 26](#) に最終的な結果を示します。



図 26 : AXI4-Stream スレーブの [IP Customization Parameters] ページ

3. 「確認と組み込み」の [手順 1](#) に進みます。

確認と組み込み

- [Review and Package] をクリックし、[Archive IP] をクリックします ([図 27](#))。

注記 : [Archive IP] ボタンをクリックし、組み込んだ IP コアの出先ディレクトリを選択して、その IP の ZIP アーカイブを作成できます。[Add to Catalog] をクリックすると、プロジェクト自体の内部に IP を追加できます。ここでは IP ごとに個別のプロジェクトを作成しているため、IP のアーカイブを作成しません。

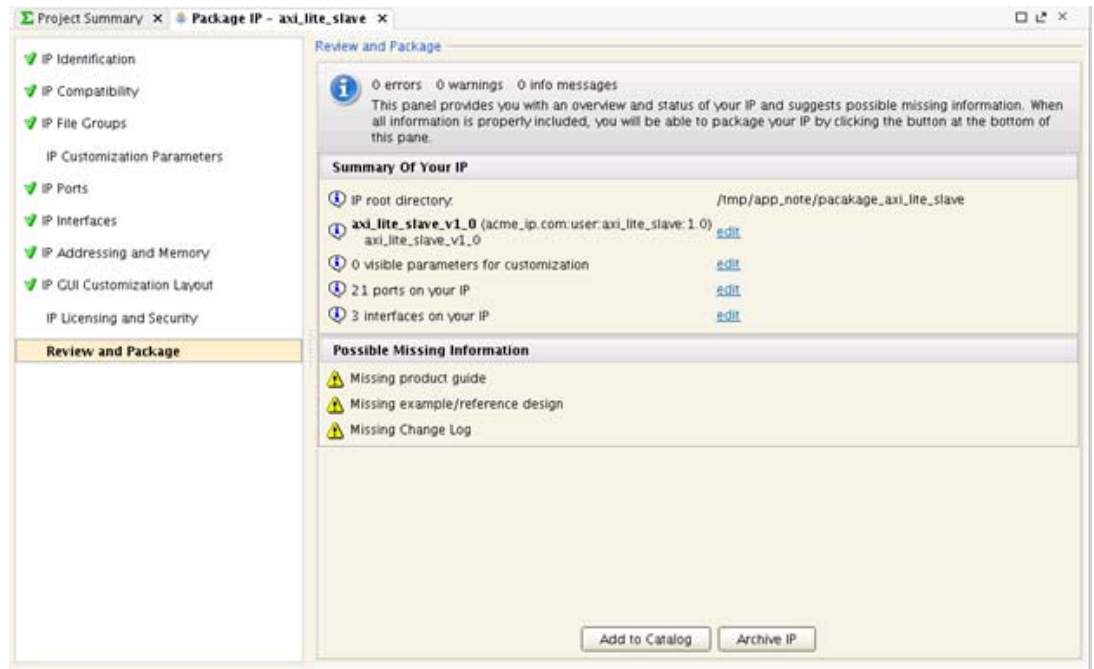


図 27 : AXI4-Lite スレーブの [Package IP] ビュー ([Review and Package] ページ)

2. [Package IP] ビューで、アーカイブの位置を次のように設定します。<design_dir>/ip_repo
図には、AXI4-Lite スレーブ IP の例を示しています。[OK] をクリックします (図 28)。

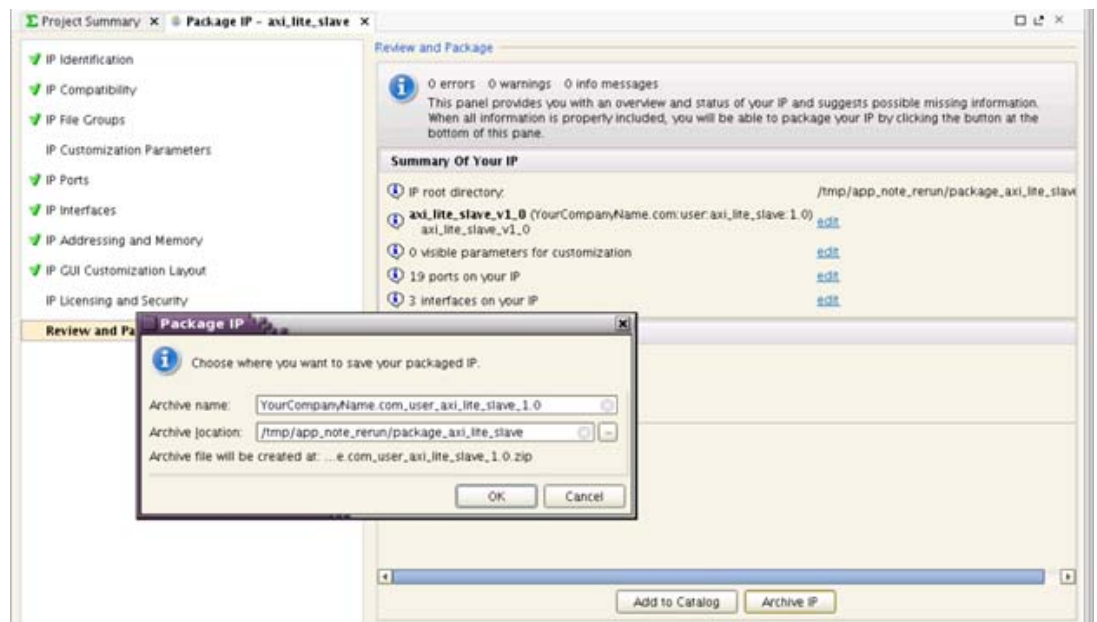


図 28 : [Package IP] ビュー (IP のアーカイブ)

3. IP は正常に組み込まれました。[File] → [Close Project] をクリックします。
組み込むその他のコアに対しても「AXI4 IP の組み込み」の手順を繰り返します。

IP インテグレーター プロジェクトへの生成済み IP の統合

AXI4-Lite システム

このセクションでは、(「[AXI4 IP の組み込み](#)」で作成した) Vivado で組み込んだ IP をプロジェクトに統合する手順について説明します。このフローは、複数のチーム、プロジェクト、または IP カスタマーの間で IP を共有する手順と類似しています。

Vivado ツールの新規プロジェクトの開始

1. 前のセクションで IP の生成が完了していない場合は、リファレンス デザイン ファイルをローカル フォルダー (<design_dir>) に解凍し、<design_dir>/ip_repo_complete ディレクトリの名前を <design_dir>/ip_repo に変更します。
2. Vivado デザイン ツールのセットアップ後、Windows で [スタート] → [Xilinx Design Tools] → [Vivado] をクリックするか、または Linux に **vivado** コマンドを入力し、Vivado ツールを開きます。
3. [Getting Started] → [Create New Project] をクリックし、新規プロジェクトを作成します。
4. [New Project] で [Next] をクリックします。
5. プロジェクト名を ipi_lite_project とし、[Next] をクリックします。
6. [Project Type] で [RTL Project] をクリックし、[Next] をクリックします。
7. [Add Sources] で [Add Files] をクリックし、<design_dir>/tb/verilog ディレクトリから lite_system_wrapper_tb.v を選択します。このファイルはテストベンチのシミュレーション ファイルです。デザインの残りの部分は、このセクションの後半で作成されます。
8. [HDL Source for:Simulation only] をオンにし、[Next] をクリックします。
9. [Add Existing IP] で [Next] をクリックします。
10. [Add Constraints] で [Next] をクリックします。
11. Vivado Design Suite によって選択されたデバイスのまま、[Default Part] ページで [Next] をクリックします。
12. [New Project Summary] ページで [Finish] をクリックします。

IP リポジトリの追加

1. [Window] → [IP Catalog] をクリックします。
2. [IP Catalog] ビューで  をクリックします。
3. [Project Settings] で [Add Repository] をクリックします (図 29)。

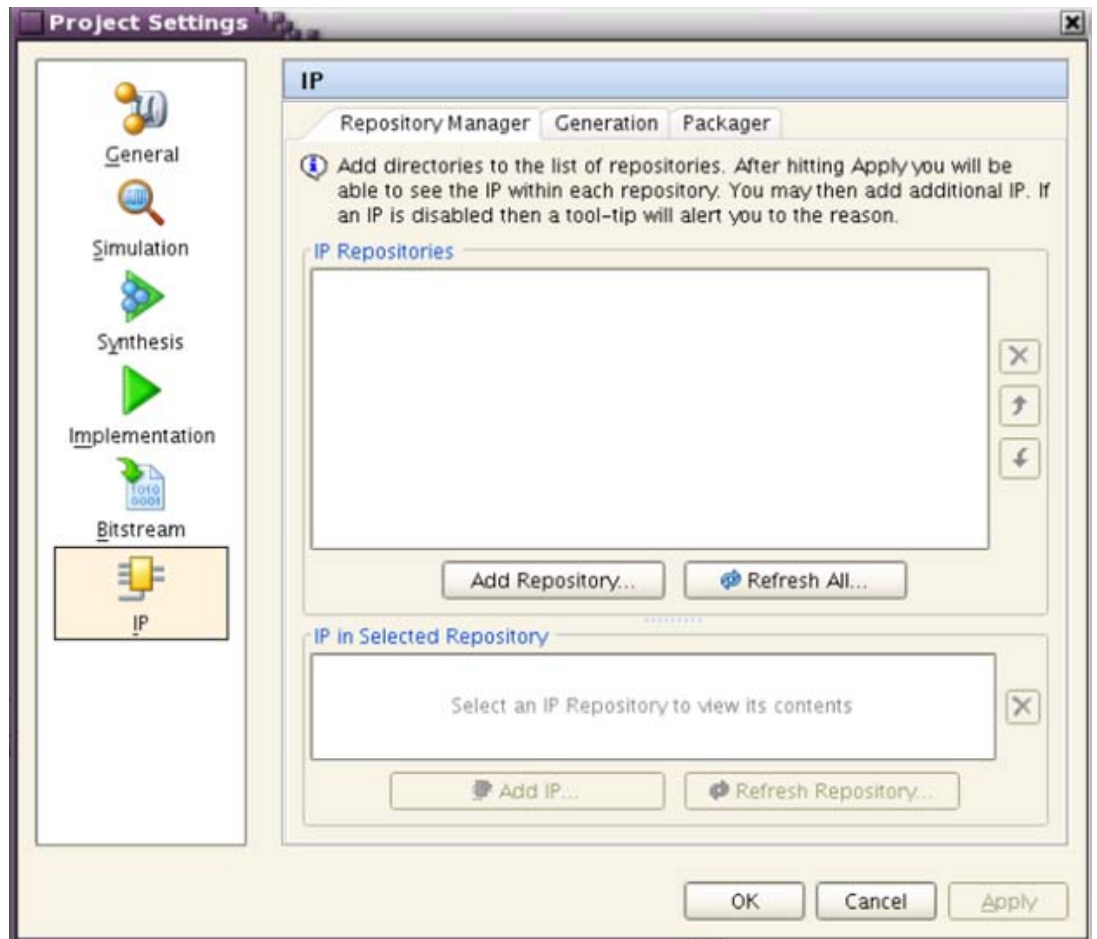


図 29 : [Project Settings] : IP Repository Manager

4. `<design_dir>/ip_repo` ディレクトリ内の IP が [IP Repositories] に既に表示されている場合は、[OK] をクリックして手順 1 に進みます。
5. `<design_dir>/ip_repo` ディレクトリを探して選択し、[Select] をクリックします。これでこのディレクトリが [IP Catalog] ビューに表示されます。
6. [Project Settings] で [Add IP] をクリックします。
7. [Select IP To Add To Repository] で `YourCompanyName.com_user_axi_lite_master_1.0.zip` を選択し、[OK] をクリックします (図 30)。これで、圧縮された ZIP ファイルは使用に適した形式で `<design_dir>/ip_repo` ディレクトリに展開されます。

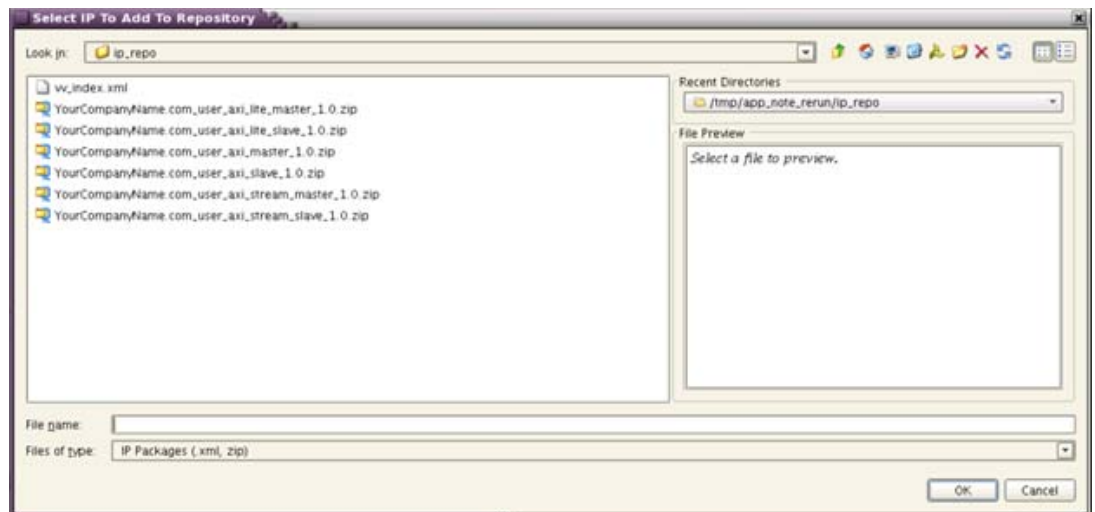


図 30 : AXI4-Lite マスターが選択されている [Select IP To Add To Repository]

この手順を繰り返して、次のファイルを追加します。

- YourCompanyName.com_user_axi_lite_slave_1.0.zip
- YourCompanyName.com_user_axi_master_1.0.zip
- YourCompanyName.com_user_axi_slave_1.0.zip
- YourCompanyName.com_user_axi_stream_master_1.0.zip
- YourCompanyName.com_user_axi_stream_slave_1.0.zip

図 31 に結果の [Project Settings] を示します。[OK] をクリックします。

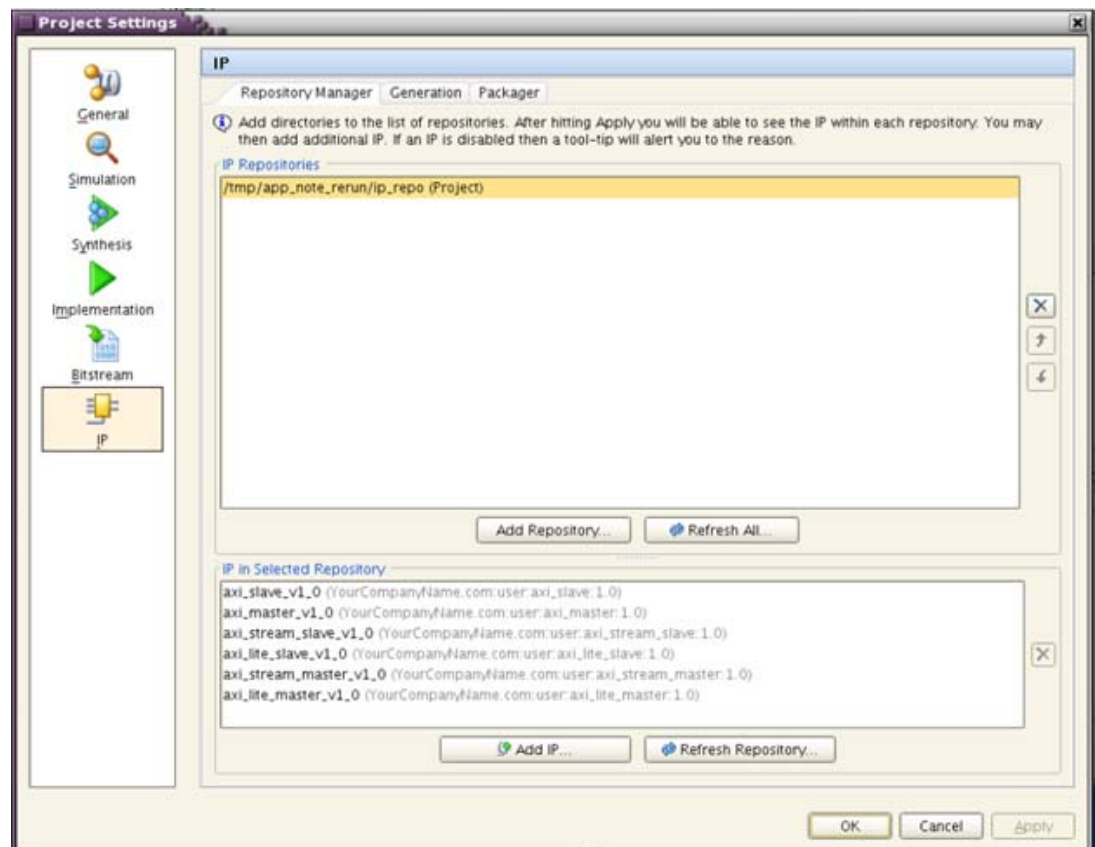


図 31 : すべての IP が選択されている [Project Settings]

システム ブロック図の作成

1. [Flow] → [Create Block Design] をクリックします。
2. [Create Block Design] で、次のように設定します。
 - [Design name] : lite_system
3. [Add IP] を右クリックして IP をキャンバスに追加します。
4. [Search] に「axi_lite_master」と入力し、Enter キーを押します (図 32)。

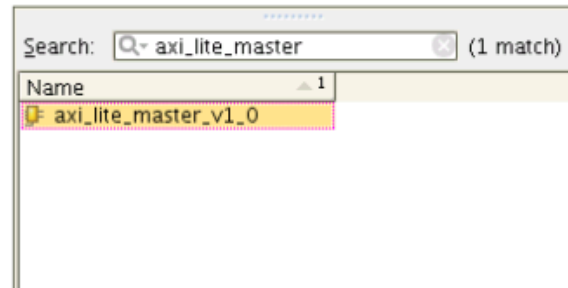


図 32 : IP インテグレーターの IP 検索ボックス

5. axi_lite_slave に対して手順 4 を繰り返します。
6. キャンバスを右クリックして [Create Port] を選択します。

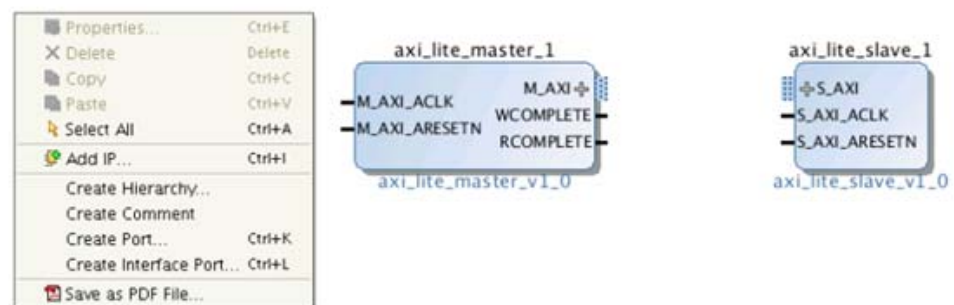


図 33 : IP インテグレーターのキャンバス : ポートの作成

7. [Create Port] ビューに次のように入力し、[OK] をクリックします (図 34)。
 - [Port name] : ACLK
 - [Type] : Clock
 - [Frequency] : 100

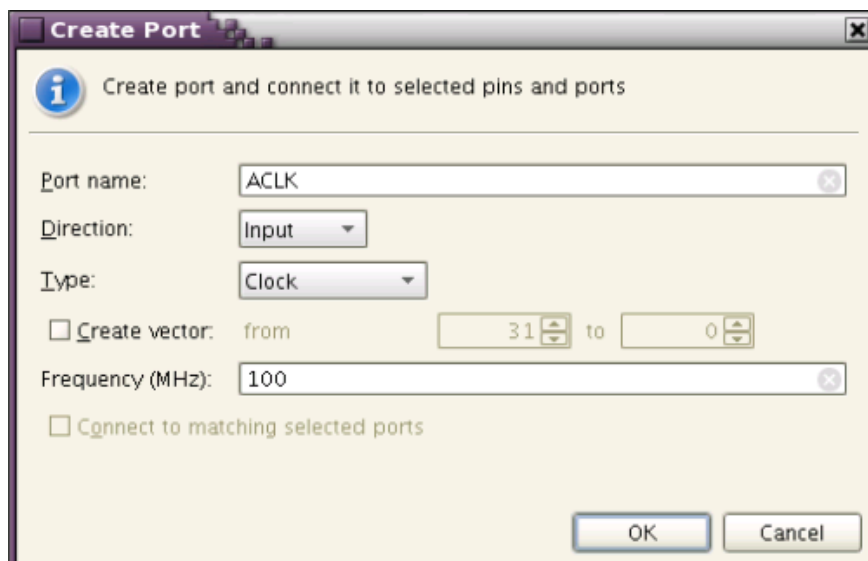


図 34 : [Create Port] : ACLK ポートの追加

8. キャンバスを右クリックして [Create Port] を選択します。
9. [Create Port] で次のように入力し、[OK] をクリックします (図 35)。
 - [Port name] : ARESETN
 - [Direction] : Input
 - [Type] : Reset

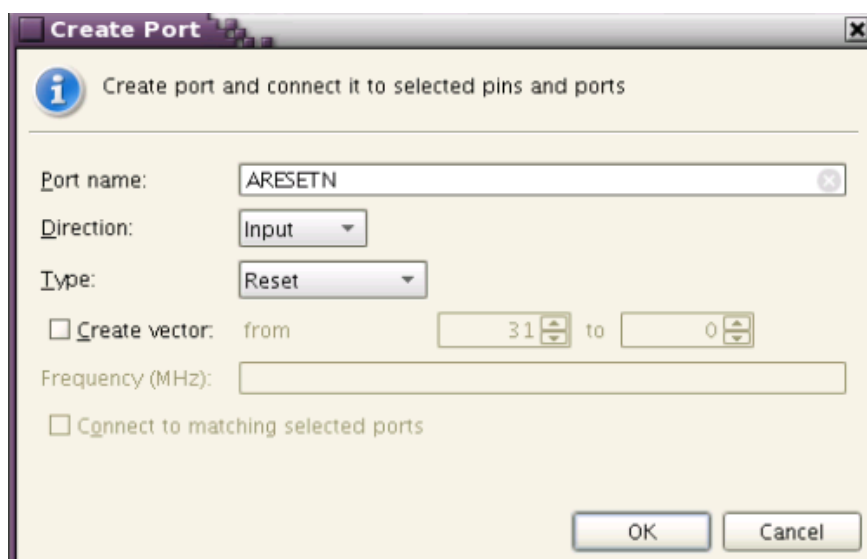


図 35 : [Create Port] : ARESETN の追加

10. ACLK ポートを次のインターフェイスに接続します。
 - axi_lite_master_1 の M_AXI_ACLK
 - axi_lite_slave_1 の S_AXI_ACLK
11. ARESETN ポートを次のインターフェイスに接続します。
 - axi_lite_master_1 の M_AXI_ARESETN
 - axi_lite_slave_1 の S_AXI_ARESETN

12. axi_lite_master_1 の M_AXI インターフェイスを axi_lite_slave_1 の S_AXI インターフェイスに接続します。

図 36 に完成した回路図を示します。

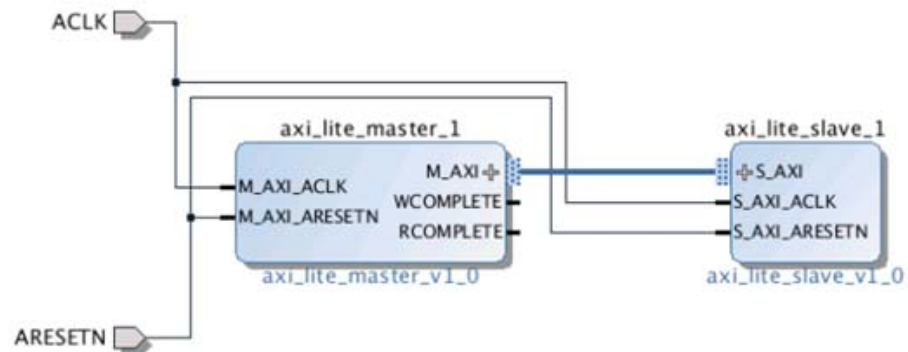


図 36 : IP インテグレーターのキャンバス上の接続された AXI4-Lite システム

13. キャンバスの [Address Editor] ビューをクリックします。
14. M_AXI をクリックし、右クリックして [Auto Assign Address] を選択します (図 37)。axi_lite_slave の AXI アドレスが選択されます。

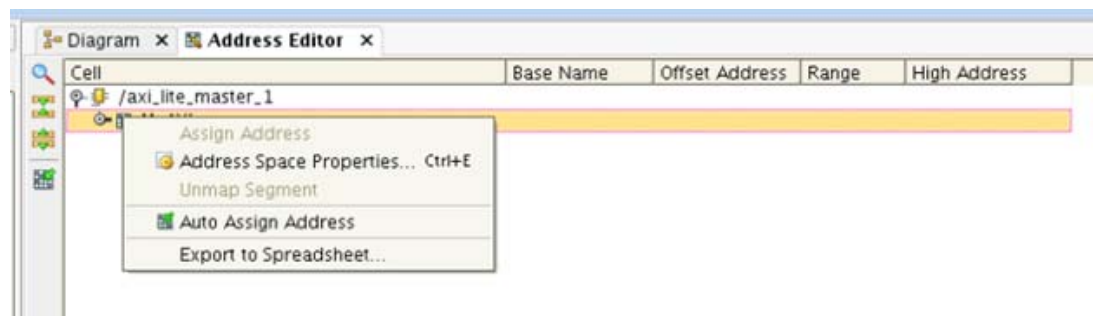


図 37 : Address Editor で axi_lite_master の [Auto Assign Address] を選択

15. [Tools] → [Validate Design] をクリックし、[OK] をクリックします (図 38)。

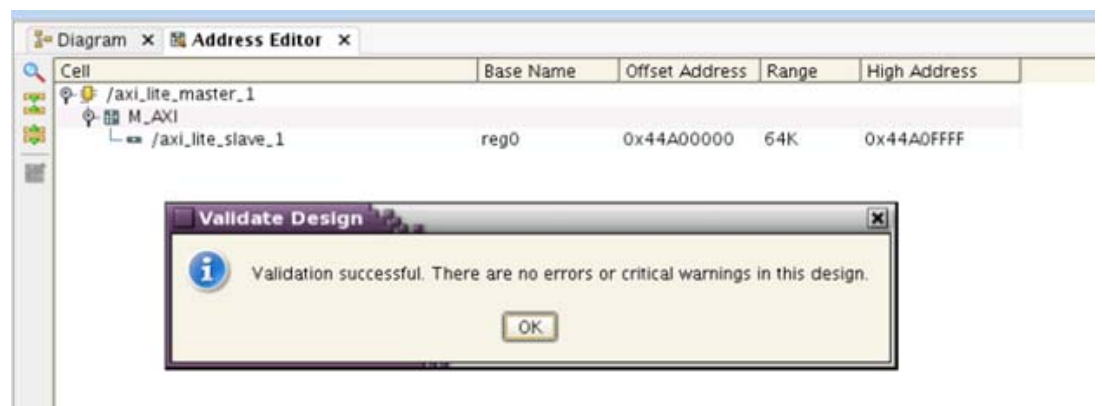


図 38 : IP インテグレーターのキャンバス : 「Validation successful」メッセージ

16. [File] → [Save Block Design] をクリックします。

システムの統合とシミュレーション

1. [Window] → [Sources] をクリックします。

- [Sources] ビューで [lite_system] を選択して右クリックし、[Create HDL Wrapper] を選択します (図 39)。ブロック図はデザインの最上位オブジェクトになることはできないため、この手順で HDL ラッパーが生成されます。生成されたファイルは、既存のテストベンチ ラッパーで置き換えられます。

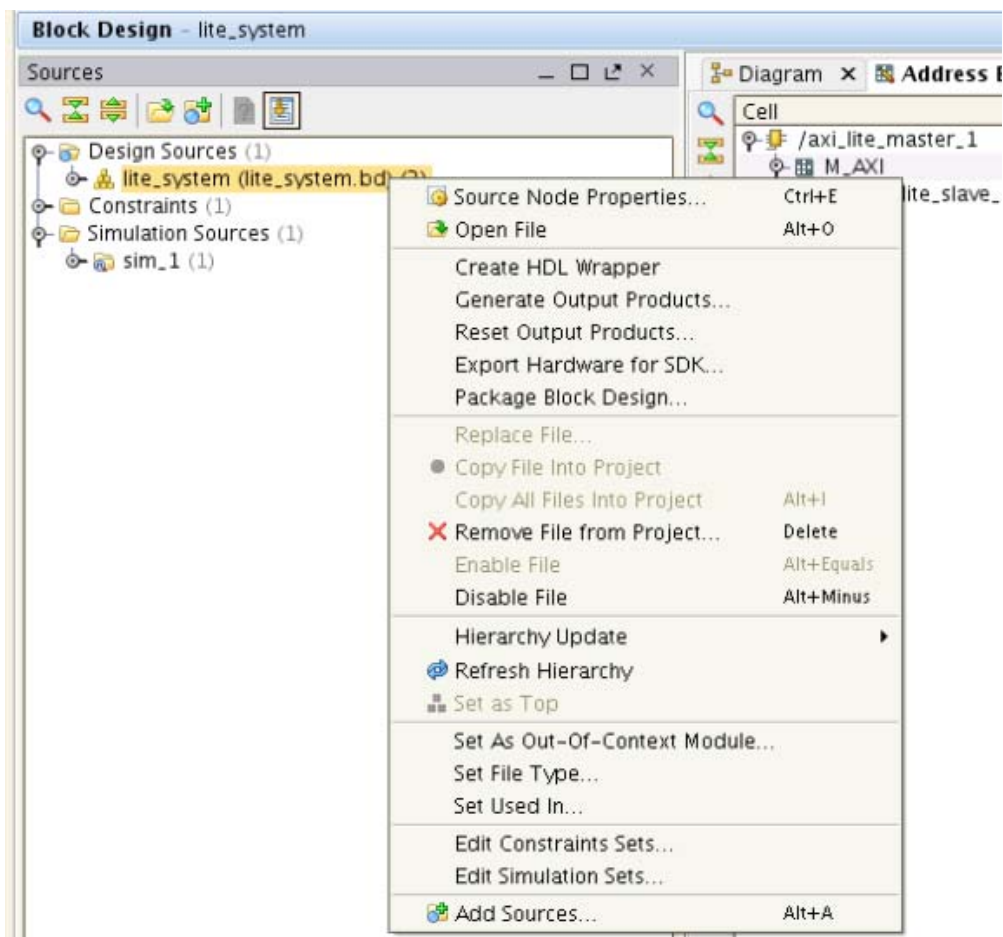


図 39 : [Sources] ビュー : HDL ラッパーの作成

- [Flow Navigator] → [Simulation] → [Simulation Settings] をクリックします。

4. [Simulation] ビューで次の値を変更し、[OK] をクリックします (図 40)。
 - [Simulation Run Time] :

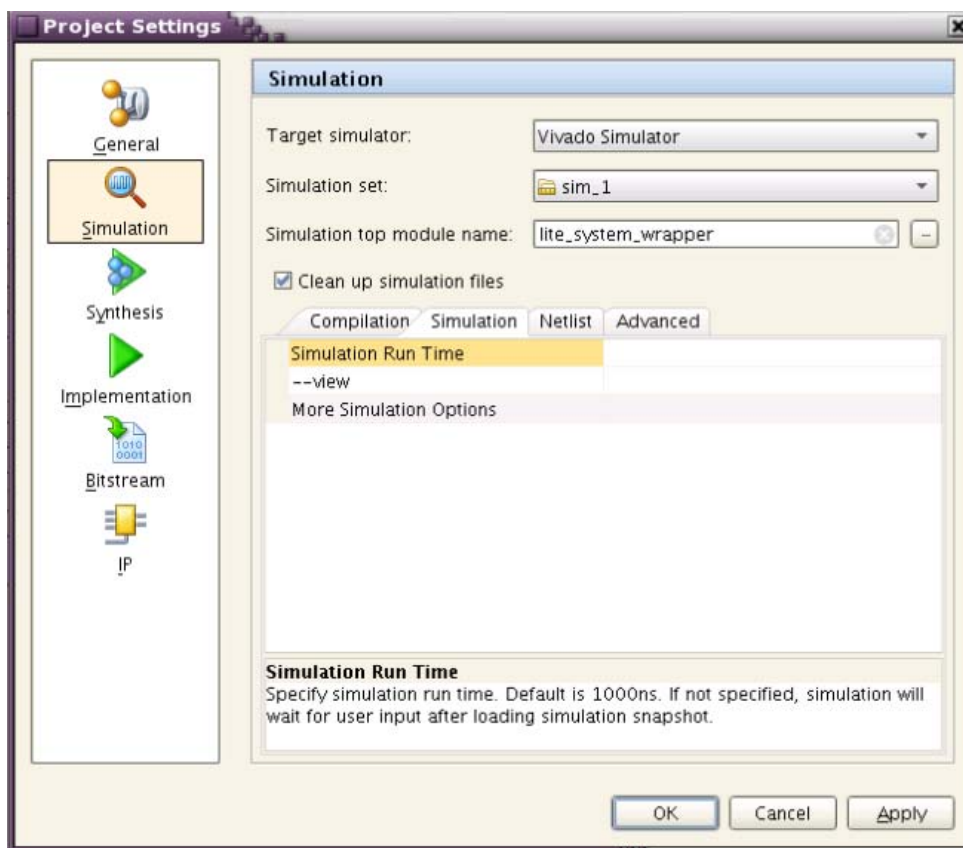


図 40 : [Project Settings] の [Simulation] ビュー : ランタイム オプション

5. [Flow Navigator] → [Simulation] → [Run Simulation] をクリックし、[Run Behavioral Simulation] をクリックします。
6. [Scopes] ビューで [lite_system_wrapper] を展開し、[lite_system_wrapper_tb] → [lite_system_wrapper_i] → [lite_system_i instance] を [Untitled 1] 波形ビューにドラッグして、シミュレーションによって AXI インターフェイスをブロック図からキャプチャできるようにします。
7. [Run] → [Run All] をクリックします。
8. [View] → [Zoom Fit] をクリックします。図 41 に得られる波形を示します。4 回の書き込みに続いて 4 回の読み出しが実行されていることがわかります。
9. 準備ができれば [File] → [Close Design] をクリックし、次のセクションに進む用意をします。

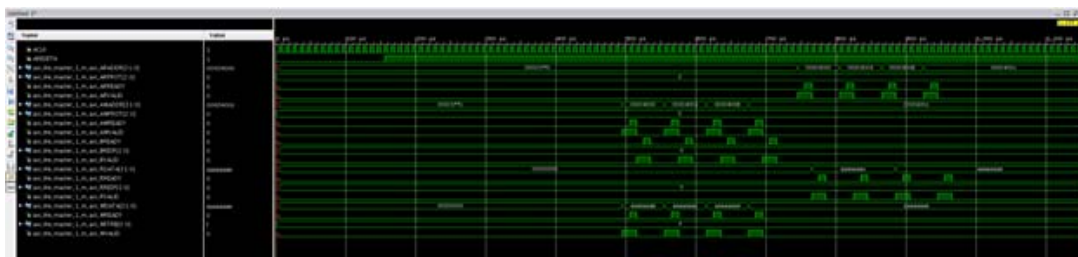


図 41 : AXI4-Lite IP インテグレーターのシステム シミュレーション

AXI4 システム

AXI4 マスターと AXI4 スレーブを含む AXI4 システムを構築する場合は、「AXI4-Lite システム」の構築と同じ手順に従いますが、次の点を変更する必要があります。

Vivado ツールの新規プロジェクトの開始

- 手順 5 では、プロジェクト名を ipi_axi_project とします。
- 手順 7 では、[Add Sources] で axi_system_wrapper_tb.v を選択します。

システム ブロック図の作成

- 手順 2 では、デザイン名を axi_system とします。
- 手順 4 では、axi_master と入力します。
- 手順 5 では、axi_slave と入力します。
- 手順 10 では、ACLK ポートを次のインターフェイスに接続します。
 - axi_master_1 の M_AXI_ACLK
 - axi_slave_1 の S_AXI_ACLK
- 手順 11 では、ARESETN ポートを次のインターフェイスに接続します。
 - axi_master_1 の M_AXI_ARESETN
 - axi_slave_1 の S_AXI_ARESETN
- 手順 12 では、axi_master_1 の M_AXI インターフェイスを axi_slave_1 の S_AXI インターフェイスに接続します。

図 42 に完成した回路図を示します。

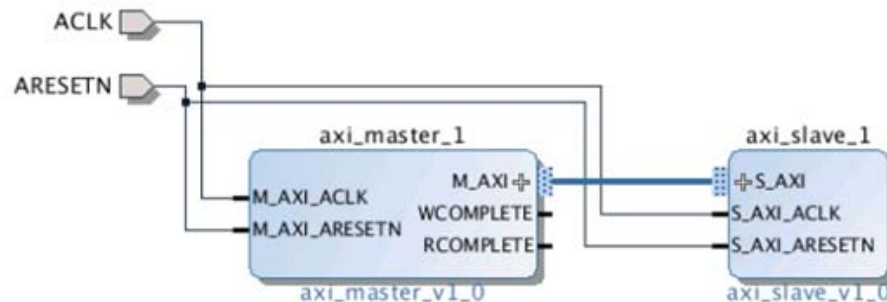


図 42 : IP インテグレーターのキャンパス上の完成した AXI4 システム

図 43 に、手順 16 まで実行した結果を示します。図には、AXI4 マスターと AXI4 スレーブの ID 幅が一致しないことを指摘する警告が示されています。サンプル AXI4 マスターは、複数のスレッドをサポートあるいは生成しないため、ID ビットがありません。サンプル AXI4 スレーブは複数のポートを備えているため、IP インテグレーターは幅の不一致が示し、AxID ポートをゼロに接続します。



図 43 : ID_WIDTH に対する警告を示す IP インテグレーター キャンパスの「Validation successful」メッセージ

システムの統合とシミュレーション

- 手順 2 では、[Sources] ビューで [axi_system] を選択して右クリックし、[Create HDL Wrapper] を選択します。
- 手順 6 では、[Scopes] ビューで [axi_system_wrapper] を展開し、[axi_system_wrapper_tb] → [axi_system_wrapper_i] → [axi_system_i] インスタンスを [Untitled] 波形ビューにドラッグします。

図 44 に波形を示します。



図 44 : AXI4 IP インテグレーターのシステム シミュレーション

AXI4-Stream システム

AXI4-Stream マスターと AXI4-Stream スレーブを含む AXI4-Stream システムを構築する場合は、「AXI4-Lite システム」の構築と同じ手順に従いますが、次の点を変更する必要があります。

Vivado ツールの新規プロジェクトの開始

- 手順 5 では、プロジェクト名を ipi_stream_project とします。
- 手順 7 では、[Add Sources] で axi_stream_system_wrapper_tb.v を選択します。

システム ブロック図の作成

- 手順 2 では、デザイン名を axi_stream_system とします。
- 手順 4 では、axi_stream_master と入力します。
- 手順 5 では、axi_stream_slave と入力します。
- 手順 10 では、ACLK ポートを次のインターフェイスに接続します。
 - axi_stream_master_1 の AXIS_ACLK
 - axi_stream_slave_1 の AXIS_ACLK
- 手順 11 では、ARESETN ポートを次のインターフェイスに接続します。
 - axi_stream_master_1 の AXIS_ARESETN
 - axi_stream_slave_1 の AXIS_ACLK
- 手順 12 では、axi_stream_master_1 の M_AXIS インターフェイスを axi_stream_slave_1 の S_AXIS インターフェイスに接続します。

図 45 に完成した回路図を示します。

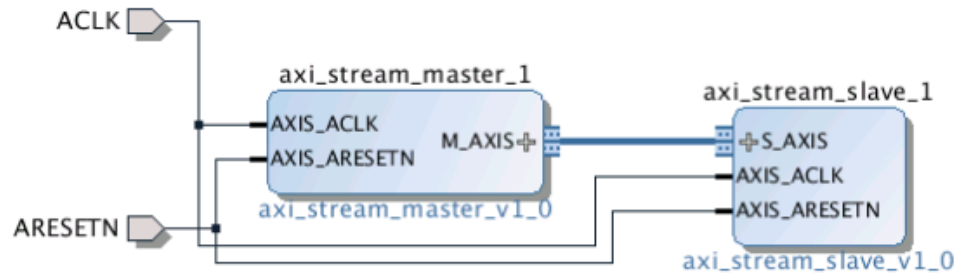


図 45 : IP インテグレーターのカンパス上の完成した AXI4-Stream システム

- AXI4-Stream はアドレスを使用しないため、手順 14 は省略します。

システムの統合とシミュレーション

- 手順 2 では、[Sources] ビューで [axi_stream_system] を選択して右クリックし、[Create HDL Wrapper] を選択します。
- 手順 6 では、[Scopes] ビューで [axi_stream_system_wrapper_tb] を展開して、[axi_stream_system_wrapper_tb] → [axi_stream_system_wrapper_i] → [axi_stream_system_i] インスタンスを [Untitled] 波形ビューにドラッグします。

図 46 に波形を示します。

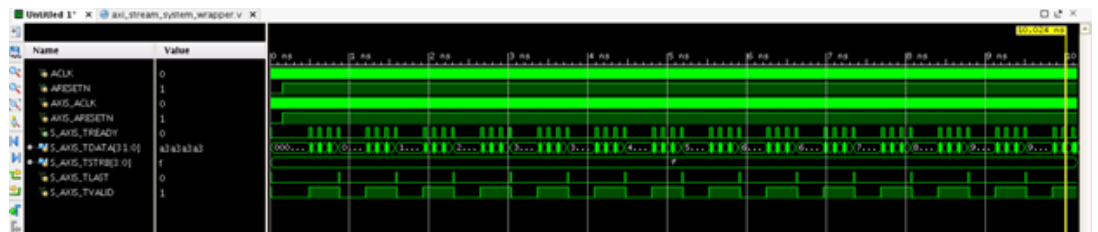


図 46 : AXI4-Stream IP インテグレーターのシステム シミュレーション

リファレンス デザイン

このアプリケーション ノートのリファレンス デザイン ファイルは、次のサイトからダウンロードできます。

<https://secure.xilinx.com/webreg/clickthrough.do?cid=338300>

参考資料

この文書では、次の参考資料を使用しています。

1. [Vivado Design Suite のマニュアル](#)
2. [AMBA \(Advanced Microcontroller Bus Architecture\) ARM AXI4 仕様](#)
3. 『AXI リファレンス ガイド』(UG761)

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	内容
2013 年 6 月 1 日	1.0	初版

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>.

本資料は英語版 (v1.0) 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。