XAPP1099 (v1.0) 2016 年 3 月 2 日

3D IC 用のエンベデッド プロセッシングを使用したローカル パーシャル リコンフィギュレーション

著者 : George Duffy

概要

ザイリンクス All Programmable 3D IC には、パーシャル リコンフィギュレーション (PR) と呼ばれる強力な機能があります。これにより設計者は、マイクロプロセッサによるタスクの切り替えと同様の方法で、デバイス リソースを時分割できます。このテクノロジは、ソフトウェア インプリメンテーションの柔軟性とハードウェア インプリメンテーションのパフォーマンスの両利点をもたらします。このテクノロジを活用したデザイン アプリケーションは多岐にわたります。一般的なアプリケーションの例と、各種のモードおよび機能オプションについては、『Vivado Design Suite ユーザー ガイド : パーシャル リコンフィギュレーション』(UG909) [参照 1] で説明しています。

3D IC はスタックド シリコン インターコネクト (SSI) テクノロジを利用し、製品ライフサイクルの早い段階で複数の小規模なプログラマブル デバイスを単一の大きなパッケージに統合して、最高の容量およびパフォーマンスを実現します。詳細は、『高集積度 FPGA 設計手法ガイド (SSI テクノロジを含む)』(UG872) [参照 2] を参照してください。Virtex-7 2000T デバイス (図 1) は、4 つの接続された SLR (Super Logic Region) で構成されます。SLR は、3D IC に含まれる単一のダイ スライスです。

これら 2 つのテクノロジの組み合わせにより、デザインの機能にかつてない柔軟性とパフォーマンスがもたらされますが、デザイン プロセスで管理する必要がある制限が伴います。このアプリケーション ノートは、パーシャル リコンフィギュレーションおよび SSI テクノロジを使用したデザインの管理について説明し、例を示します。

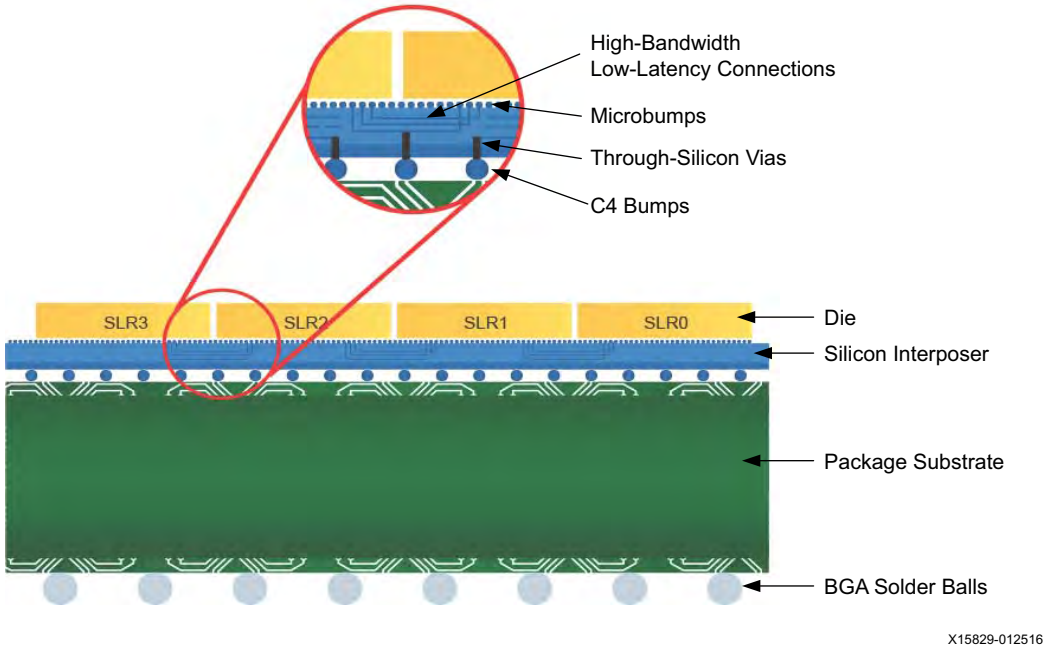


図 1 : SSI テクノロジで実現した Virtex-7 2000T FPGA

パーシャル リコンフィギュレーションを実行する 1 つの方法は、マスターの内部コンフィギュレーション アクセス ポート (ICAP) を使用することです。ICAP インターフェイスにより、プログラマブル ロジックは 7 シリーズ コンフィギュレーション システムへのアクセスが可能になります。ただし、ICAP の使用には次の制約があります。

- SPI 以外の初期コンフィギュレーション方法を使用する場合、リコンフィギュレーション可能なパーティションがターゲットのスレーブ ICAP と同じ SLR に存在しない限り、パーシャル ビットストリームの転送にはマスター SLR (SLR1) 内の ICAP を使用する必要があります。
- SPI モードでコンフィギュレーションする場合、マスター ICAP はスレーブ SLR 内のコンフィギュレーション メモリにアクセスできません。
- モード ピンが JTAG モードに設定されている場合、マスター ICAP はスレーブ SLR にアクセスできません。
- SPI x2 または SPI x4 モードでコンフィギュレーションする場合、スレーブ SLR の ICAP を使用できません。その他のすべてのコンフィギュレーション モードでは、ローカル ICAP アクセスを使用できます。
- SPIx2 または SPIx4 コンフィギュレーション モードでは、SEM IP (誤り訂正) は使用できません。

SSI デバイスのコンフィギュレーション メモリへの ICAP アクセスの詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) [参照 3] を参照してください。

重要 : 7 シリーズ デバイスでは、PR 領域は SLR の境界をまたぐことはできず、単一の SLR 内に完全に収まっている必要があります。この制約は、UltraScale™ アーキテクチャ ベースのデバイスでは解決済みです。

SPI x1 コンフィギュレーション モードのみの使用に課される制約に対処する 1 つの方法は、各 SLR の ICAP を制御する MicroBlaze デザインをインプリメントすることです。これにより、関連するパーシャル ビットストリームを正しい ICAP に適用できます。IP インテグレーター (IPI) を使用すれば、大きなオーバーヘッドなしに MicroBlaze デザインを既存のプロセッサ デザインに追加できます。このアプリケーション ノートで示すエンベデッド ソリューションの代替方法または補完方法では、デザインのリコンフィギュレーションを管理するために、Vivado™ Design Suite に含まれるザイリンクスの Partial Reconfiguration Controller IP を使用します。このコアの詳細は、『Partial Reconfiguration Controller v1.0 LogiCORE IP 製品ガイド』(PG193) [参照 4] で説明しています。

このアプリケーション ノートでは、IPI デザインの作成手法を説明しています。これは最大で 4 つの ICAP を制御して、特定の ICAP にローカルなモジュールをリコンフィギュレーションできます。リファレンス デザインは、Vivado ツール ベースのパーシャル リコンフィギュレーション デザインを開始するために必要なリソースを提供します。パーシャル リコンフィギュレーション デザイン フローの詳細は、『Vivado Design Suite ユーザー ガイド : パーシャル リコンフィギュレーション』(UG909) [参照 1] を参照してください。リファレンス デザインは、ボードでパーシャル リコンフィギュレーション ソリューションを開発するためのガイドラインです。これは各 SLR の ICAP に接続される MicroBlaze システムをインプリメントし、Virtex-7 2000T デバイス (XC7V2000T、FHG1761 パッケージ、-1 スピード仕様) をターゲットにしています。このデザイン アプローチは、UltraScale デバイスを含むすべてのザイリンクス SSI デバイスをターゲットにできますが、新機能の追加により一部の機能は異なります。たとえば、UltraScale デバイスでは、リコンフィギャラブル モジュールは SLR 境界をまたぐことができます。デバイス ファミリごとのパーシャル リコンフィギュレーションの機能および制限の詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) [参照 3] を参照してください。

はじめに

このアプリケーション ノートで説明されている IPI デザインは、MicroBlaze™ システム、クロッキング ウィザード、UART Lite インターフェイス、および 4 つの axi_hwicap IP (各 SLR に 1 つずつ) で構成されます。MicroBlaze ベースの IPI デザインの作成例は、『Vivado Design Suite チュートリアル : エンベデッド プロセッサ ハードウェア デザイン』(UG940) [参照 5] を参照してください。

リコンフィギュレーションする SLR の選択から、選択したパーシャル ビットストリームの ICAP へのロードまで、このデザインのすべての操作の管理に MicroBlaze システムが使用されます。この例では、パーシャル ビットストリームは UART インターフェイスを使用して、MicroBlaze システムにバイト単位で転送されます。それ後、MicroBlaze システムは ICAP への書き込みを実行します。このセットアップはデモンストレーションのみを目的としていますが、ビットストリームの転送を SPI、BPI、SD、またはイーサネットなどのボード リソースで使用して、パーシャル ビットストリームを取得できます。

デザイン フローの概要

パーシャル リコンフィギュレーション可能なデザインをインプリメントすることは、共通ロジックを共有する複数の非 PR デザインをインプリメントすることと同じです。複数のデザイン間の共通ロジックを同一にするために、パーティションが使用されます。

各モジュールの HDL ソースは、別々に合成してから、スタティック ロジック ネットリストに追加する必要があります。コンフィギュレーションのフル/パーシャル ビット ファイルを生成するために、各デザインには適切なネットリストがインプリメントされます。最初のインプリメンテーションのスタティック ロジックは、後続のすべてのデザイン インプリメンテーションで共有されます。フル/パーシャル リコンフィギュレーション フローの詳細は、『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470) [参照 3] を参照してください。このフローのその他の使用例については、『Vivado Design Suite チュートリアル : パーシャル リコンフィギュレーション』(UG947) [参照 6] を参照してください。

次に、パーシャル リコンフィギュレーション フローについて簡単に説明します。

1. スタティック モジュールとリコンフィギャラブル モジュールを個別に合成します。
2. 物理制約 (Pblock) を作成して、リコンフィギャラブル領域を定義します。
3. 各リコンフィギャラブル パーティションに HD.RECONFIGURABLE プロパティを設定します。
4. コンテキスト内で完全なデザイン (スタティックであり、リコンフィギャラブル パーティションごとに 1 つのリコンフィギャラブル モジュール) をインプリメントします。
5. 完全に配線されたデザインのデザイン チェックポイントを保存します。
6. このデザインからリコンフィギャラブル モジュールを削除して、スタティック専用のデザイン チェックポイントを保存します。
7. スタティックな配置配線をロックします。
8. 新規のリコンフィギャラブル モジュールをスタティック デザインに追加して、この新規コンフィギュレーションをインプリメントします。
9. すべてのリコンフィギャラブル モジュールがインプリメントされるまで、手順 8 を繰り返します。
10. すべてのコンフィギュレーションで検証ユーティリティ (pr_verify) を実行します。
11. 各コンフィギュレーションのビットストリームを作成します。

アプリケーション ソフトウェア

アプリケーション ソフトウェアは、MicroBlaze システム用にザイリンクス ソフトウェア開発キット (SDK) [参照 7] を使用してコンパイルされ、ワークスペースにバイナリとソース コードを組み込みます。図 2 に概要フロー図を示します。ビットストリーム サイズを設定し、ロード用のリコンフィギュレーション済みモジュールを含む SLR を選択します。ビットストリームが送信可能になると、プロセッサに信号が送られ、受信可能の肯定応答 (ACK) が返されます。プロセッサへのビットストリームの送信には、Tera Term のファイル転送機能 [参照 8] を使用します。プロセッサは選択した ICAP への書き込みと、デバイスのパーシャル リコンフィギュレーションを実行します。

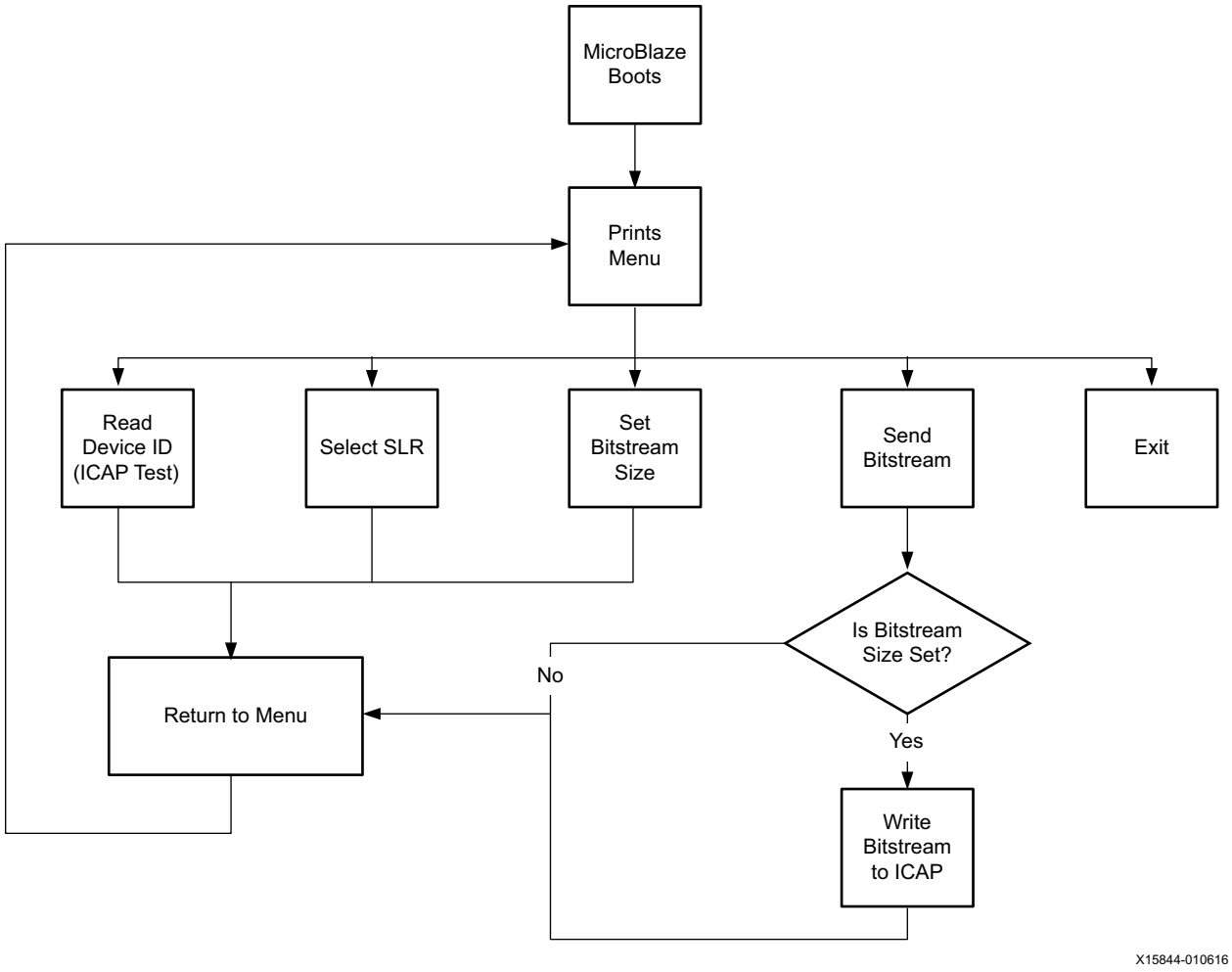


図 2 : アプリケーション フローチャート

MicroBlaze システムは、書き込まれたバイト数が入力ビットストリーム サイズと一致するまで、受信したワードの ICAP への書き込みを継続します。

プロセッサ システム プロジェクトの作成

「リファレンス デザイン」ファイルをザイリンクスのウェブサイトからダウンロードします。ZIP ファイルの内容を、書き込みアクセス可能な場所に抽出します。解凍された ssi_icap_xapp データ ディレクトリは、このアプリケーション ノートでは <Extract_Dir> と表記します。デザインのプロジェクト部分によって、デザインの IPI 部分が作成され、デザインのスタティック部分が合成されます。2 つのリコンフィギャラブル ブロックでは、付属のネットリストの作成には、ボトムアップ合成が既に使用されています。これらのネットリストの 1 つによって 6 つの LED が左にシフトし、別のネットリストによって右にシフトします。デザインには、LED を駆動するリコンフィギャラブル パーティションを選択するプッシュボタンもあります。これにより、出力を多重化して、各 SLR でリコンフィギュレーションが正常に実行されていることを視覚的に検証できます。

新規プロジェクトの作成

1. Vivado Design Suite バージョン 2015.3 以降を開きます。
2. 以降のステップでは、新規プロジェクトの作成について詳述します。<Extract_DIR>/ssi_icap_base の下には、事前に作成済みのプロジェクトがあります。このプロジェクトには、デザイン フローが完了したすべての必須ファイルが含まれています。ここでは、VC707 ボードに適したデバイスが使用され、I/O 制約はどのカスタム ボードにも適合しないと想定しています。デザインには独自の I/O 制約を適用する必要があります。事前に含まれているビットストリーム ファイルは使用しないでください。
3. Vivado Design Suite を開き、Getting Started ページで [Create a New Project] をクリックします。あるいは、source コマンドで <Extract_DIR>/scripts/project.tcl を実行してプロジェクトを生成すると、手順 22 にスキップできます。[Next] をクリックします。
4. 図 3 に示すように、[Project name] に「ssi_icap_prj」と入力し、[Project location] に <Extract_Dir> を指定します。[Create project subdirectory] は必ずオンにします。[Next] をクリックします。

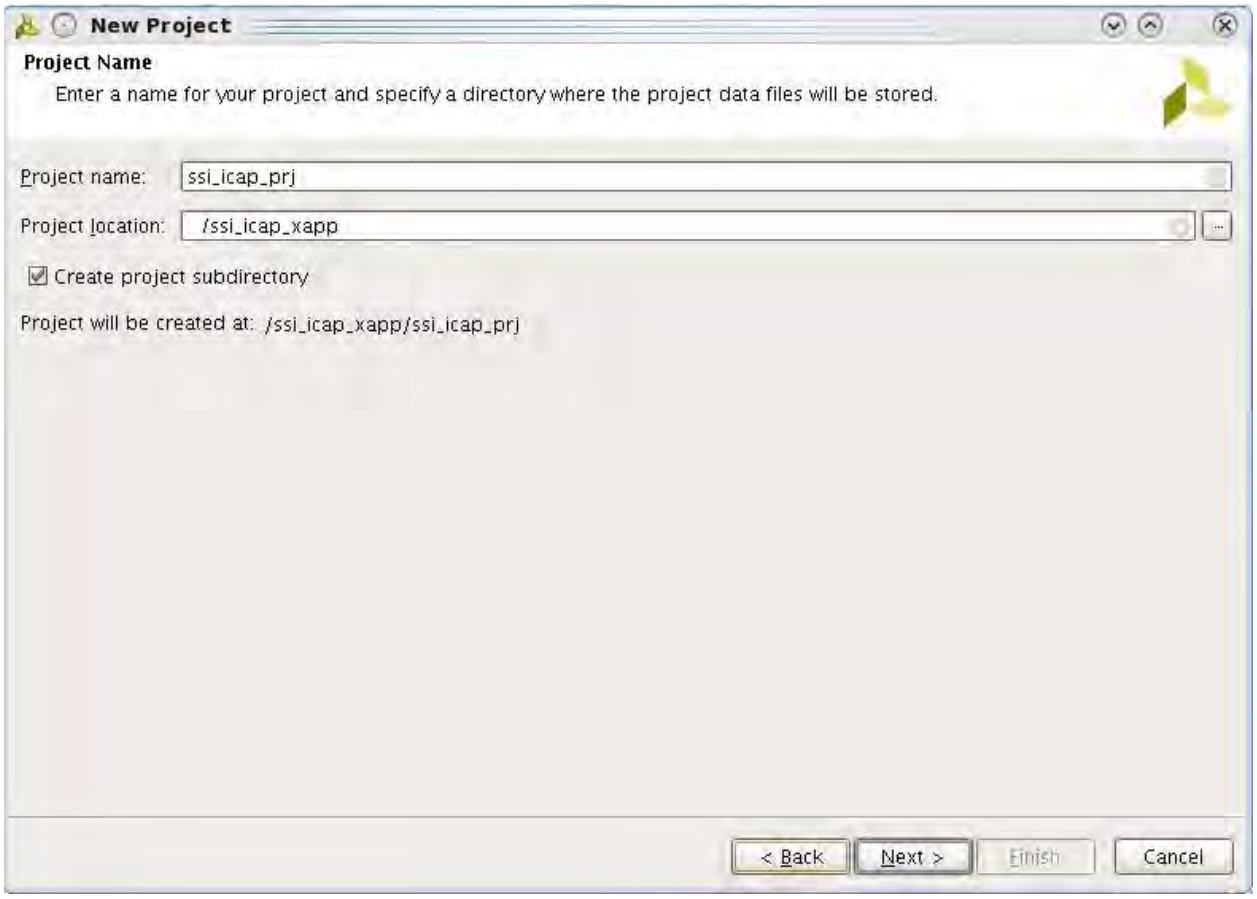


図 3 : 新規プロジェクトの作成ウィザード

5. [RTL Project] → [Do not specify sources at this time] → [Next] をクリックします。

6. デバイス選択ウィンドウ (図 4) で、xc7v2000tfhg1761-1 を選択します。使用するサンプルが別の SSI デバイスをターゲットとしている場合は、その特定のデバイスを入力して [Next] をクリックします。

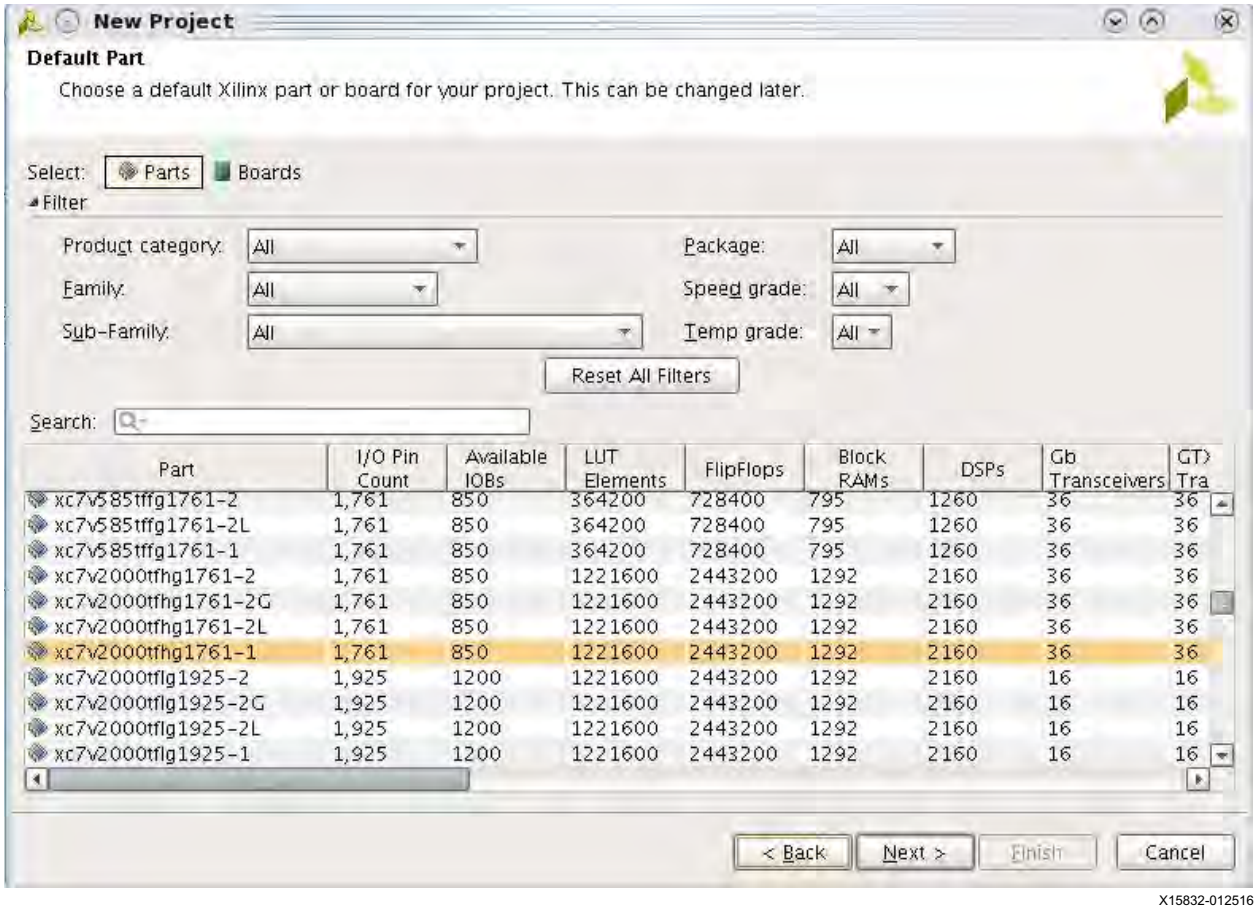


図 4 : デバイス選択の GUI

7. [New Project Summary] ページ (図 5) で、選択したオプションが正しいことを確認し、[Finish] をクリックします。

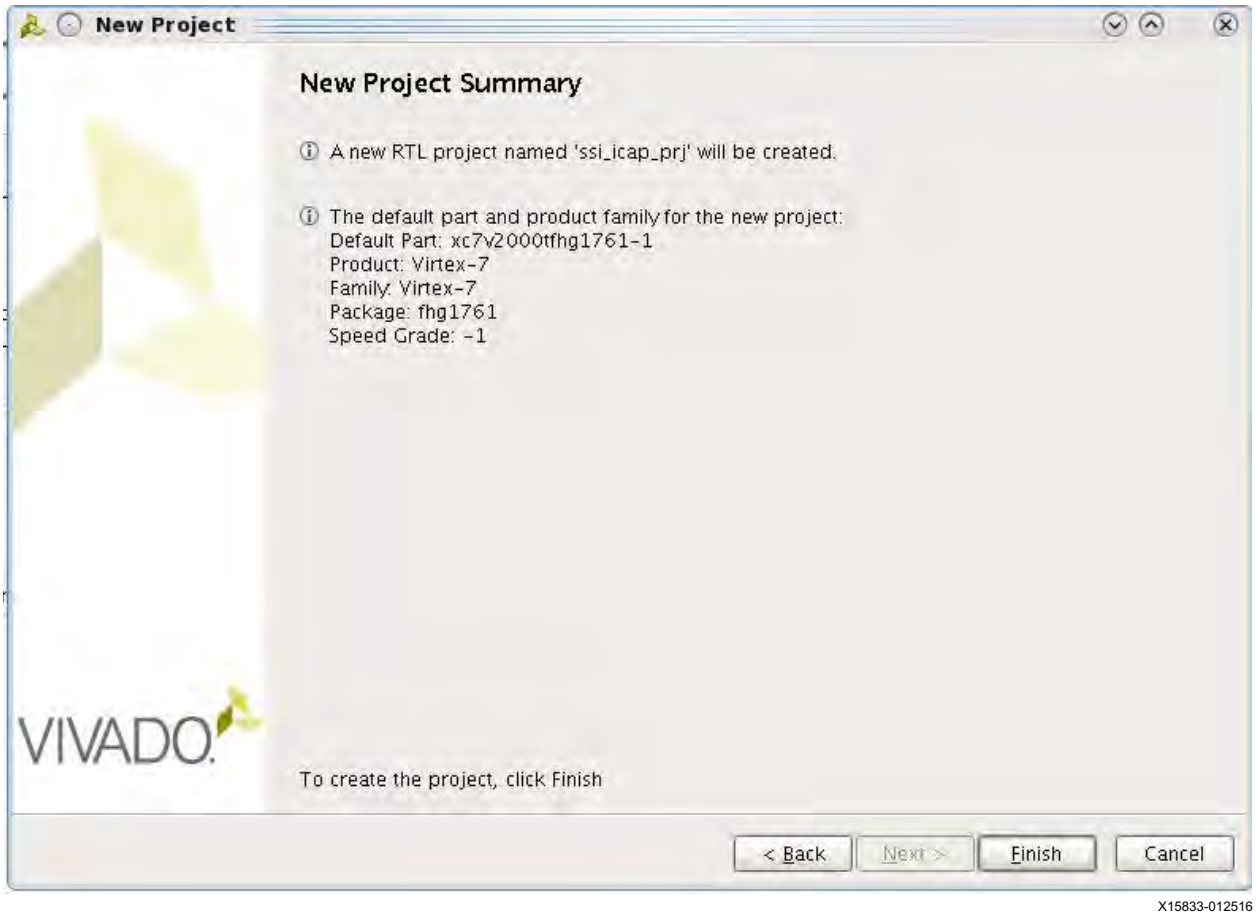
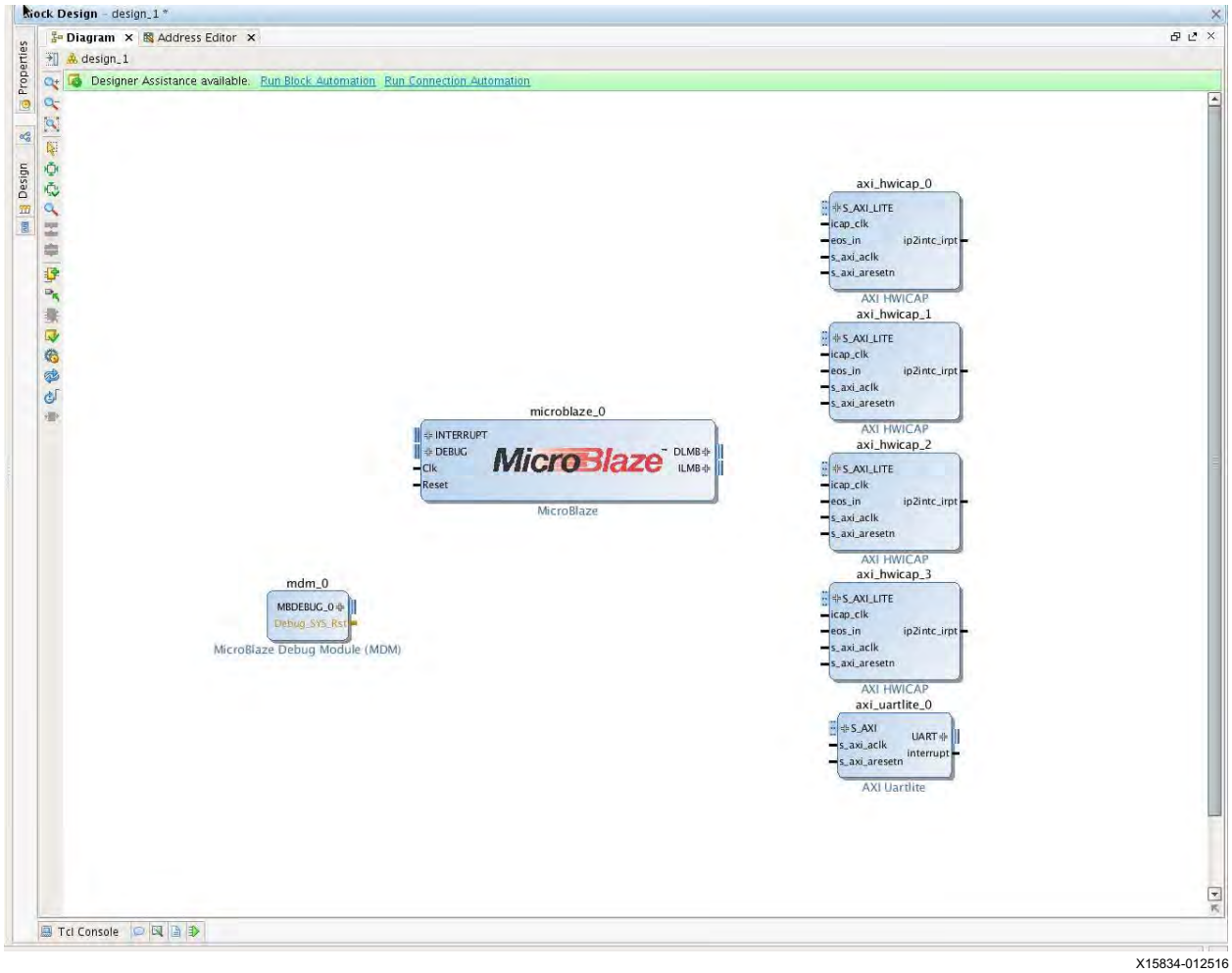


図 5 : [New Project Summary] ページ

8. プロジェクトの初期化が完了したら、Project Manager で [Create Block Design] をクリックし、hwicap_design1 と名前を付けます。ディレクトリはプロジェクトに対してローカルのままとしておき、ソースを [Design Sources] に設定します。あるいは、source コマンドで <Extract_DIR>/scripts/build_bd_design.tcl を実行してブロック デザインを構築し、手順 22 にスキップできます。[OK] をクリックします。

9. 図 6 で空白部分を右クリックして、[Add IP] をクリックします。1 つの MicroBlaze プロセッサ、1 つの MicroBlaze デバッグ モジュール、4 つの AXI ハードウェア ICAP IP、および 1 つの AXI UART (Uuniversal Aynchronous Receiver Transmitter) Lite IP を追加します。



X15834-012516

図 6 : 初期の IPI プロジェクト

10. MicroBlaze ページをダブルクリックして、カスタマイズを開始します。[Enable MicroBlaze Debug Module Interface] をオンにします。[Next] をクリックして、4 番目のページに進みます (図 7)。[Enable Local Memory Bus Instruction Interface] と [Enable Local Memory Bus Data Interface] の両方をオンにします。[OK] をクリックしてカスタマイズを終了します。

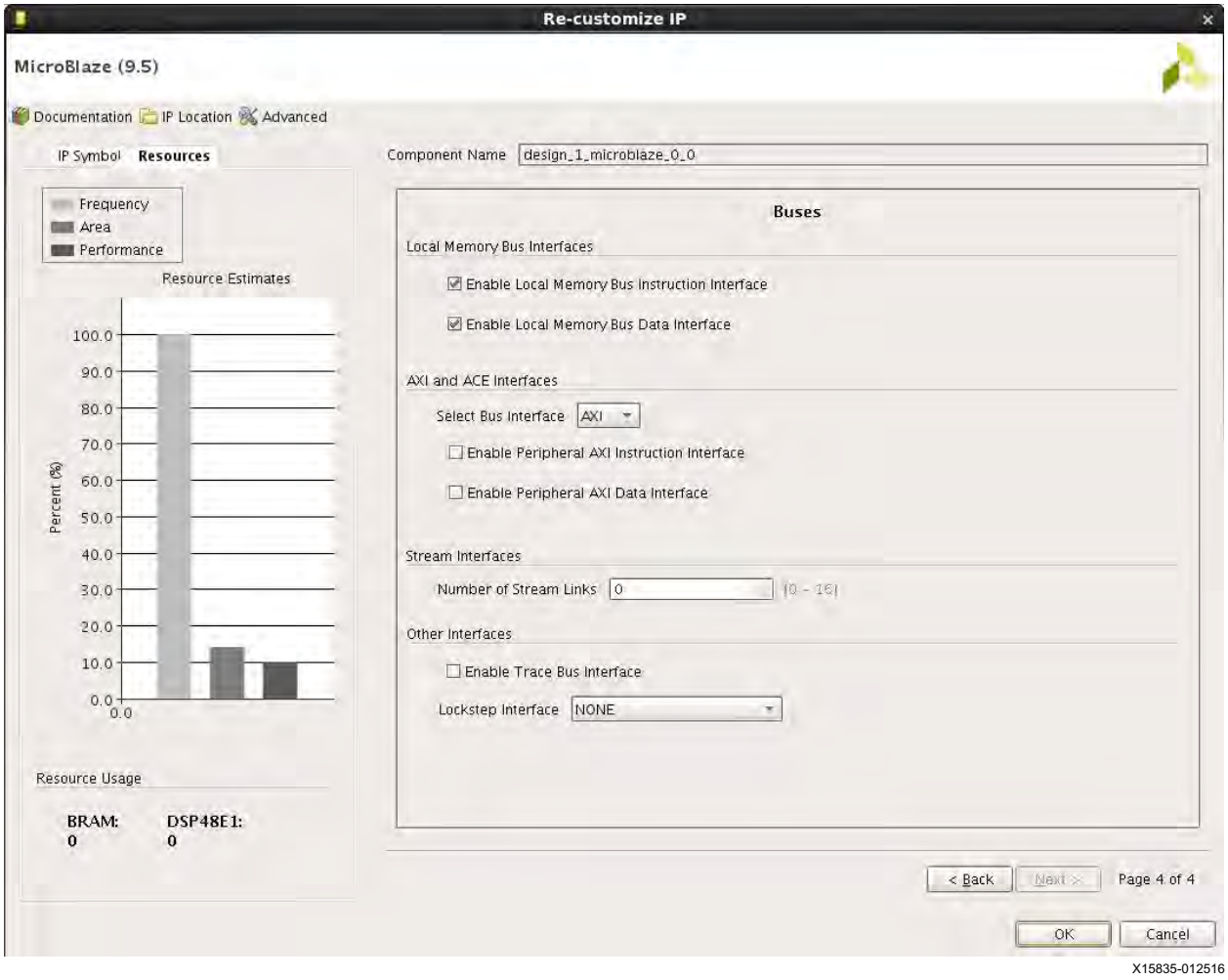


図 7 : MicroBlaze カスタマイズ ウィザード

11. [Designer Assistance available] の横にある [Run Block Automation] リンクをクリックしてダイアログ ボックスを開きます (図 8)。[Local Memory] を 128KB に、[Cache Configuration] を None にそれぞれ変更します。その他のすべての値はデフォルトのままにします。[OK] をクリックします。このステップでは、Clocking Wizard IP と、プロセッサのブロック RAM リソースが自動的に追加されます。

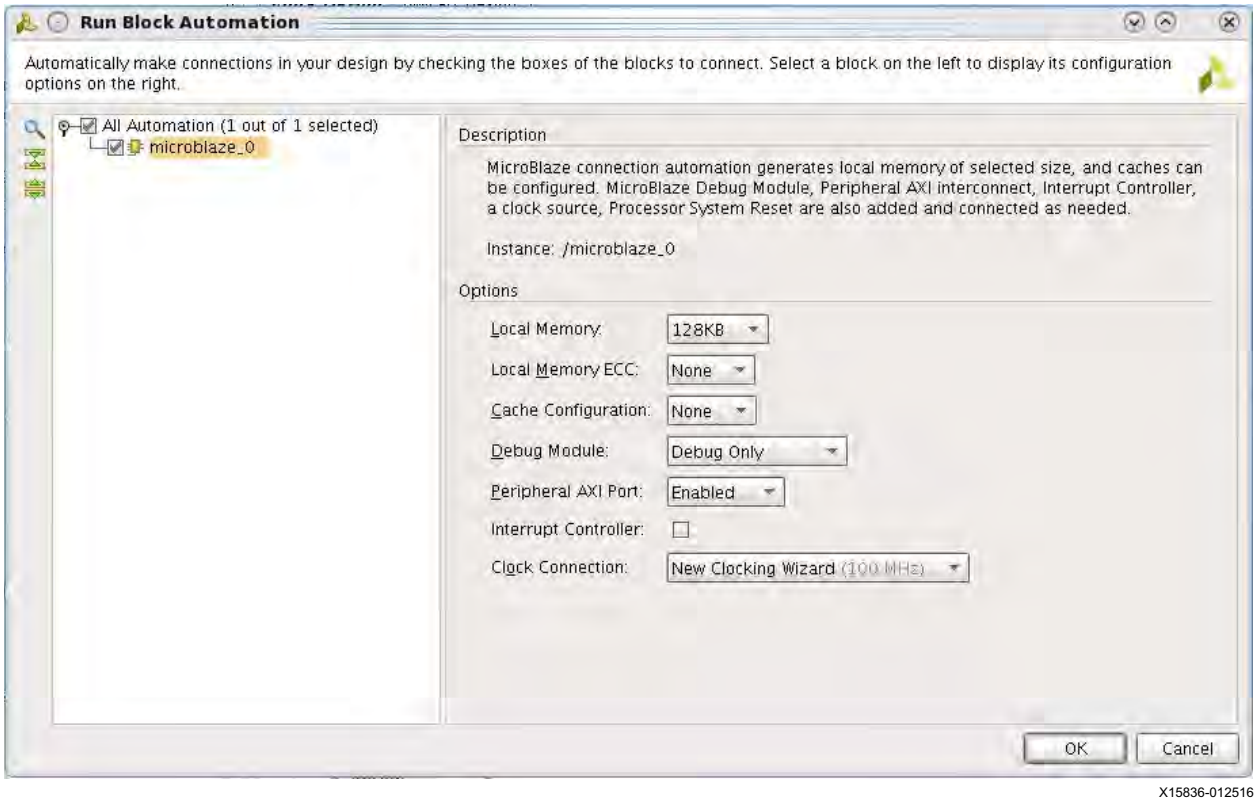


図 8 : [Run Block Automation] ダイアログ ボックス

12. [Run Connection Automation] リンクをクリックしてダイアログ ボックスを開き、[All Automation] をオンにします (図 9)。rst_clk_wiz IP に対して [ext_reset_in] ポートをオンにし、[Select Reset Polarity] を ACTIVE_HIGH に変更します。[OK] をクリックします。

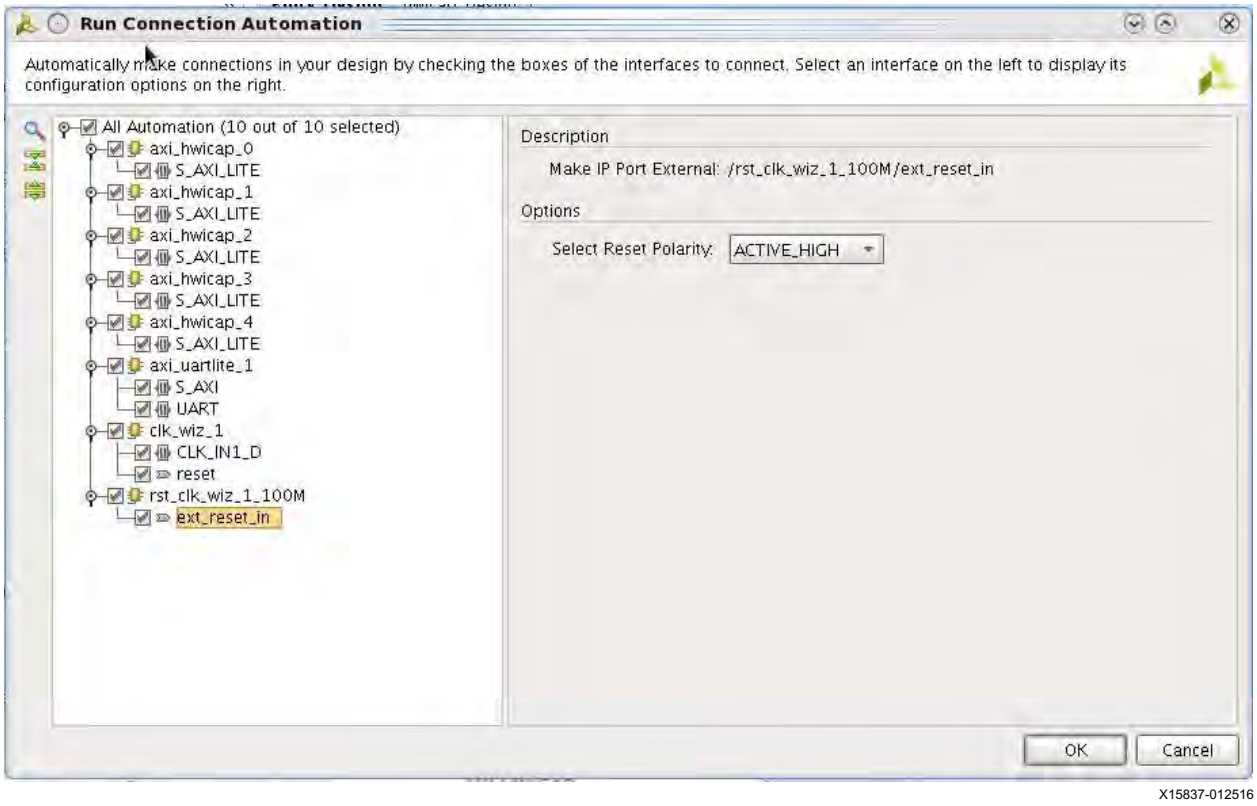


図 9 : [Run Connection Automation] ダイアログ ボックス

13. レイアウト再生成ボタン (図 10) をクリックすると、デザイン レイアウトは自動的に整頓されます。

図 10 : レイアウト再生成ボタン

14. AXI UART Lite をダブルクリックします (図 11)。[Baud Rate] を 115200 に、[Data Bits] を 8 に、[Parity] を No Parity にそれぞれ設定します。[OK] をクリックします。

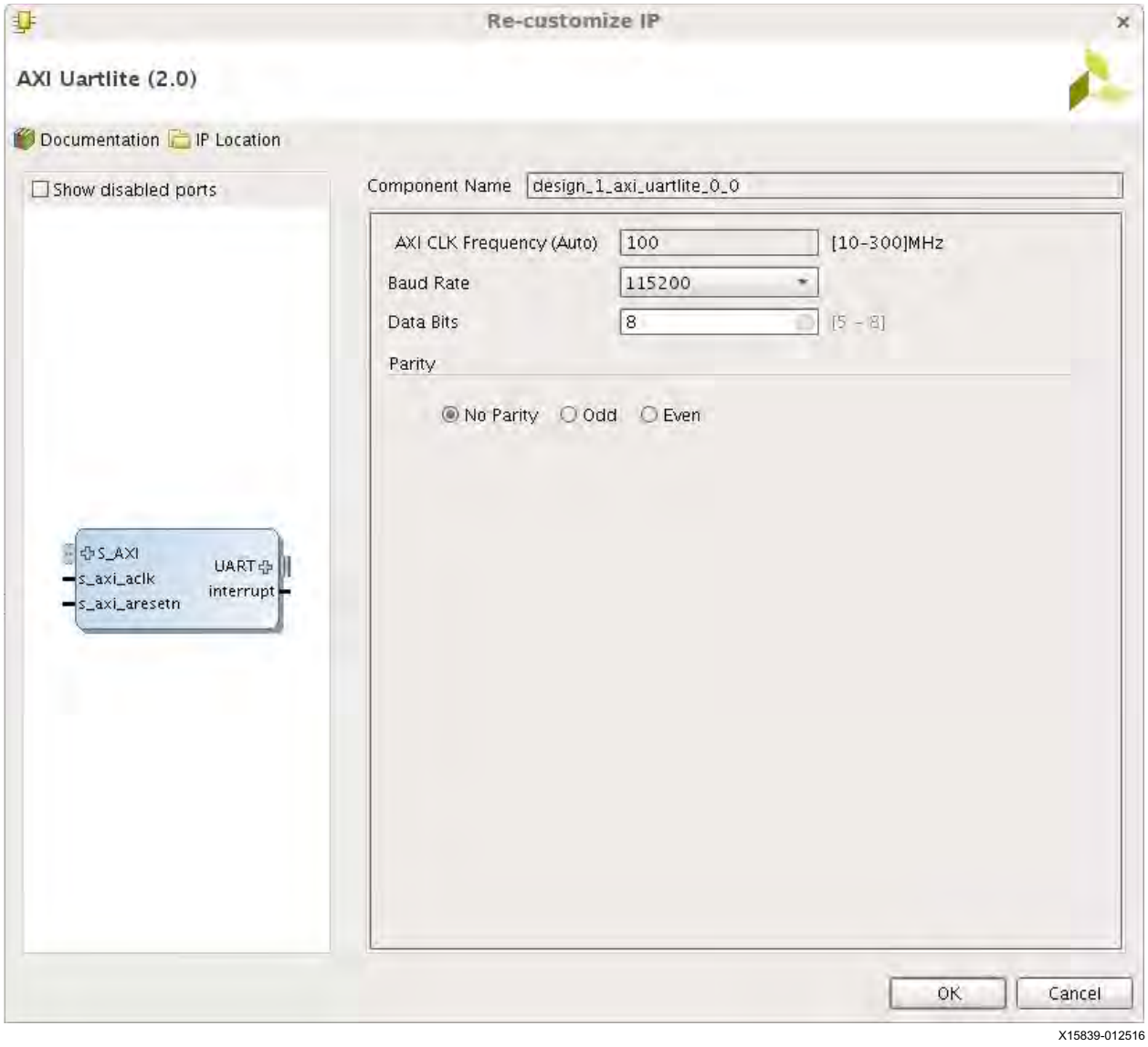


図 11 : AXI UART Lite の GUI

15. Clocking Wizard をダブルクリックします (図 12)。このデザインは、200MHz 差動入力クロックを使用します。カスタム ボードまたは異なるクロック入力には、正しい入力周波数の入力と適切なソース タイプの選択が必要です。[Output Clocks] ウィンドウを開いて、出力が 100MHz に設定されていることを確認します。[OK] をクリックします。

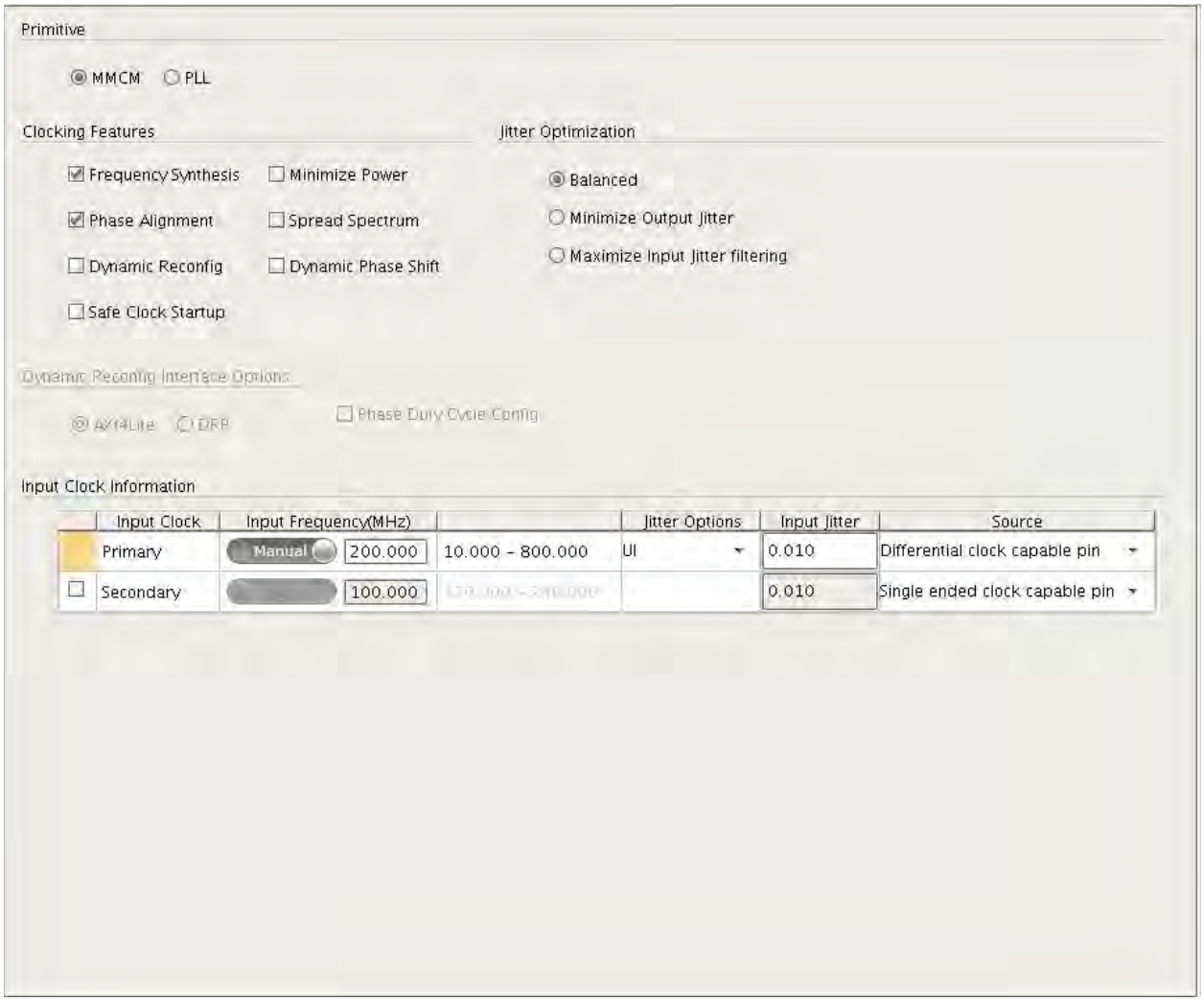


図 12 : Clocking Wizard のカスタマイズ GUI

16. Clocking Wizard の clk_out1 ポートを AXI HWICAP ブロックの icap_clk ポートに接続します。Clocking Wizard の clock_out1 ポートを右クリックし、[Create Port] をクリックします。デフォルトの選択のままにします。[OK] をクリックします。

17. diff_clock_rtl ポートをダブルクリックし、周波数を入力クロックと一致するように設定します (図 13)。[OK] をクリックします。

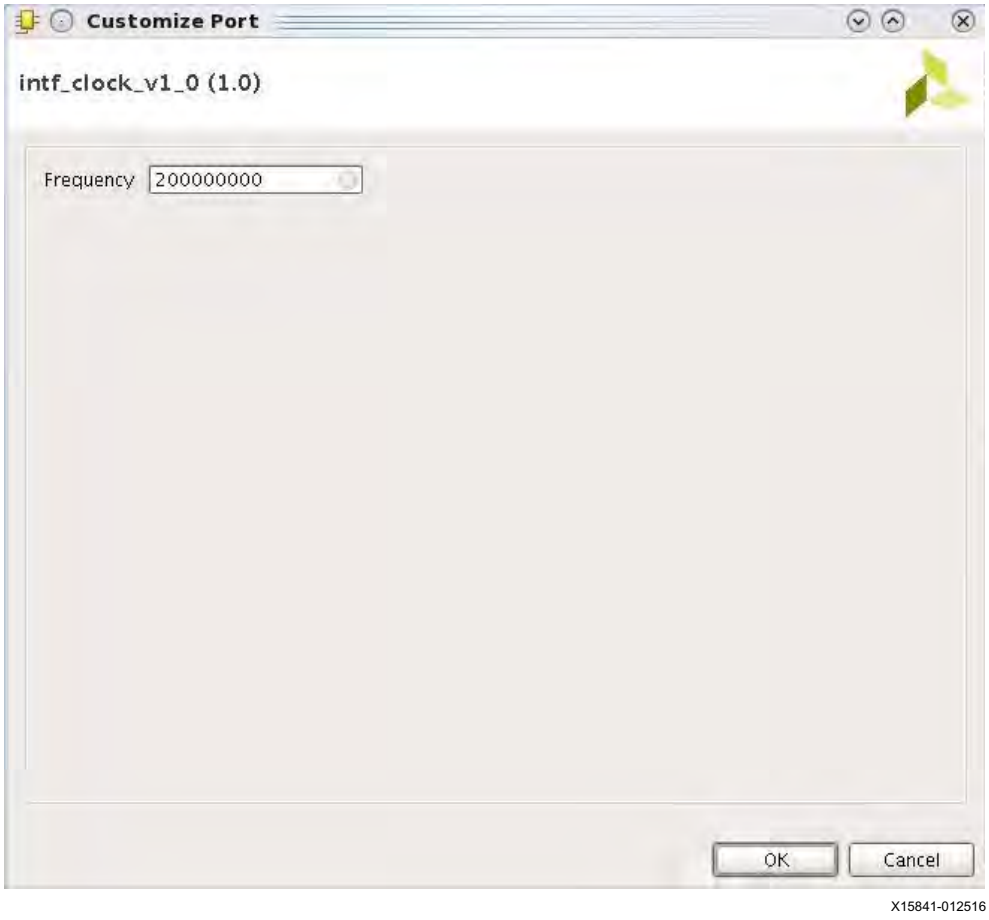


図 13 : 入力クロック周波数の設定

18. デザイン検証ボタンをクリックします。これにより IPI デザインは完全に有効であり、ポートの不一致がないことが確認されます。

19. 完成した IPI デザインは、図 14 のようになります。

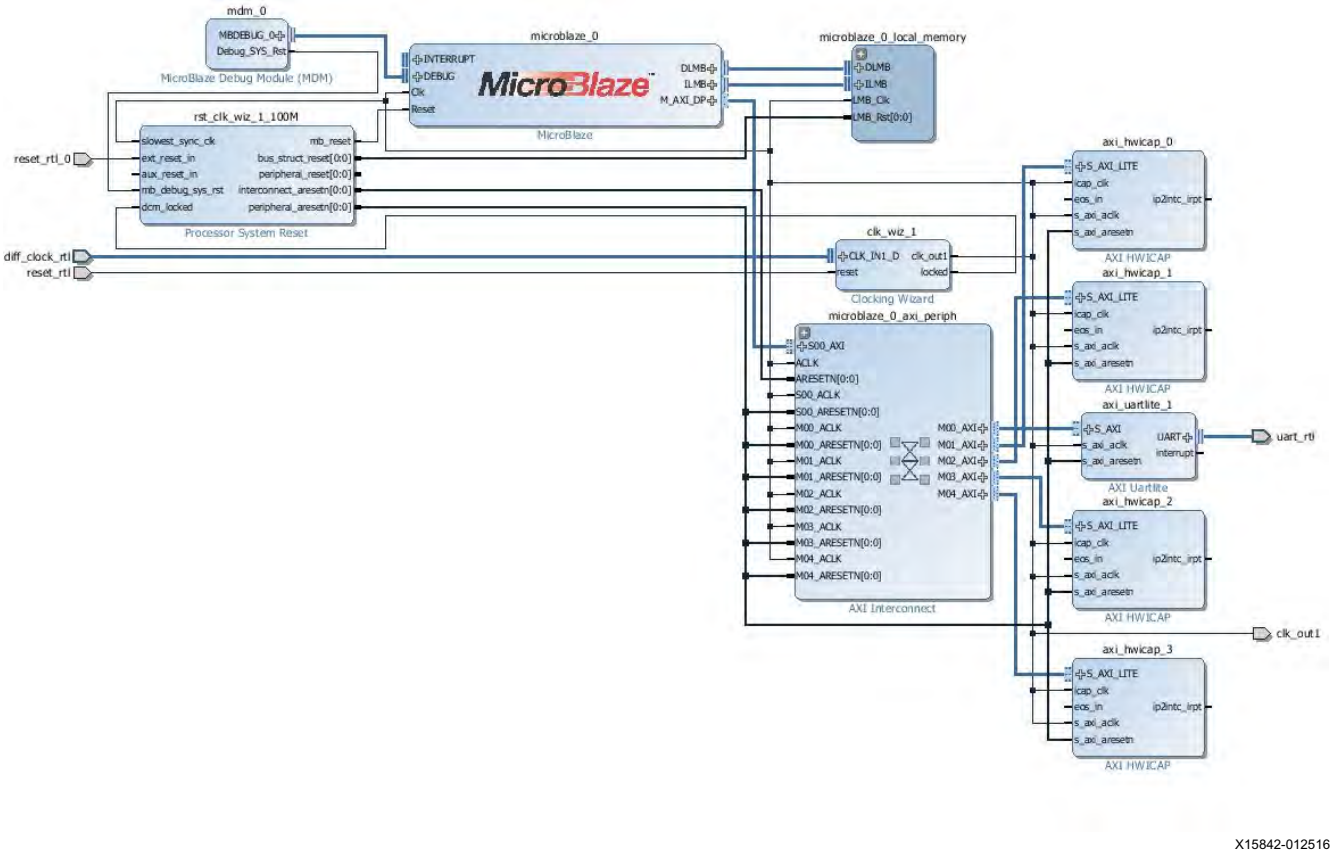


図 14 : 完成した IPI デザインの例

20. 次のステップは、最上位モジュール用の Verilog ソース ファイルのインポートです。これは、デザインで使用されるブロック デザインとリコンフィギャラブル モジュールをインスタンシエートします。[Sources] ウィンドウ内で右クリックして、[Add Sources] をクリックします。[Add or Create Design Sources] をクリックして、[Next] をクリックします。次のファイルを選択します。

```
<Extract_DIR>/Sources/hdl/hwicap_design_1_wrapper.v
<Extract_DIR>/Sources/hdl/shift_wrapper.v
<Extract_DIR>/Sources/hdl/debounce.v
```

[Copy sources into project] がオンになっていることも確認します。

21. ソースを再度追加しますが、今回は [Add or Create Constraints] をクリックします。ファイル `<Extract_DIR>/Sources/xdc/hwicap_const.xdc` をプロジェクトに追加します。これには ICAP サイトのさまざまな Pblock の制約と LOC 制約が含まれており、それらすべてを個別の SLR に適用します。Virtex-7 2000T はザイリンクス評価ボードに搭載されておらず、ユーザー デザインのセットアップと一致するクロック制約や I/O 制約がありません。ハードウェア上でデザインを実行するには、ユーザー アプリケーションに応じて、ファイルを編集して制約を追加します。XDC には、必要なすべての制約が含まれています。変更する必要があるのは、設定されているパッケージ ピンとクロック周期制約のみです。

22. [Sources] ウィンドウで hwicap_design1.bd を右クリックします。[Generate Output Products] をクリックします。[Generate] をクリックします。

23. これでデザインの該当部分が完成したので、後にザイリンクス SDK でソフトウェア アプリケーションを作成する際に使用するハードウェア仕様をエクスポートする必要があります。[File] → [Export] → [Export Hardware] をクリックします。[Export to] に Local to Project を選択して、[OK] をクリックします。これにより、プロジェクト ディレクトリ内に `ssi_icap_prj.sdk` というフォルダーが作成されます。このフォルダーにはツールでアプリケーションを作成するために必要となる HDF ファイルが含まれます。

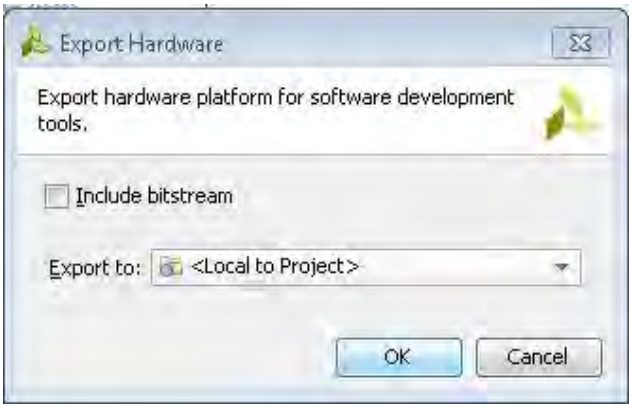


図 15 : [Export Hardware]

24. パーシャル リコンフィギュレーション インプリメンテーション フローを実行するには、TCL コンソールで source コマンドで `<Extract_Dir>/scripts/design.tcl` を実行します。このスクリプトはデザインを合成し、パーシャル リコンフィギュレーション デザイン フローを実行し、ハードウェア上でデザインを実行するために必要なフルおよびパーシャルのすべてのビットストリームを生成します。ビットストリームは、`<Extract_Dir>/ssi_icap_prj/implement/` に生成されます。これらのビットストリームは、UpdateMEM コマンドを使用して、ブロック RAM 初期値に埋め込まれている MicroBlaze システム アプリケーション ELF ファイルと結合できます。これにより、コンフィギュレーションが完了したらすぐにアプリケーションを実行できます。このプロセスを必要とするのは、初期のフル ビットストリームのみです。詳細は、『Vivado Design Suite チュートリアル : エンベデッド プロセッサ ハードウェア デザイン』(UG940) [参照 5] の第 6 章に記載されています。

SDK のステップ

コンパイル済みバイナリを含む事前に準備されたソフトウェア ワークスペースは、`<Extract_Dir>/sw/workspace` にあります。これは、複数の ICAP インスタンスのアプリケーション制御を確認するためにリファレンス用に組み込まれています。類似のワークスペースを最初から作成するには、『Vivado Design Suite チュートリアル : エンベデッド プロセッサ ハードウェア デザイン』(UG940) [参照 5] に概説されているステップを参照し、以前の Vivado プロジェクトからエクスポートしたハードウェア プラットフォーム仕様を使用してください。

ハードウェア上でのデザインの実行

ハードウェア上でリファレンス デザインを実行するには、Silicon Labs 社のドライバーが必要です。これは Silicon Labs 社のウェブサイト [参照 9] の CP210x USB - UART ブリッジ VCP ドライバー ページから入手できます。Tera Term などのターミナル エミュレーション アプリケーションは、Tera Term のホーム ページ [参照 8] で入手できます。

このデザインは、4 つのリコンフィギャラブル パーティション (RP) で構成されます。単一の High ビットが、ボード上の LED に出力される 6 つのビットに同調して左または右方向にシフトされます。各 RP は、デバイスのそれぞれの SLR 内に配置されます。RP の出力は多重化されるので、一時点で LED に表示されるのは 1 つの SLR 出力のみです。左端の 2 つの LED は、バイナリで現在の SLR 出力を示します。接続されたプッシュボタンを使用して、出力 RP を選択できます。図 16 に、デザインのブロック図を示します。

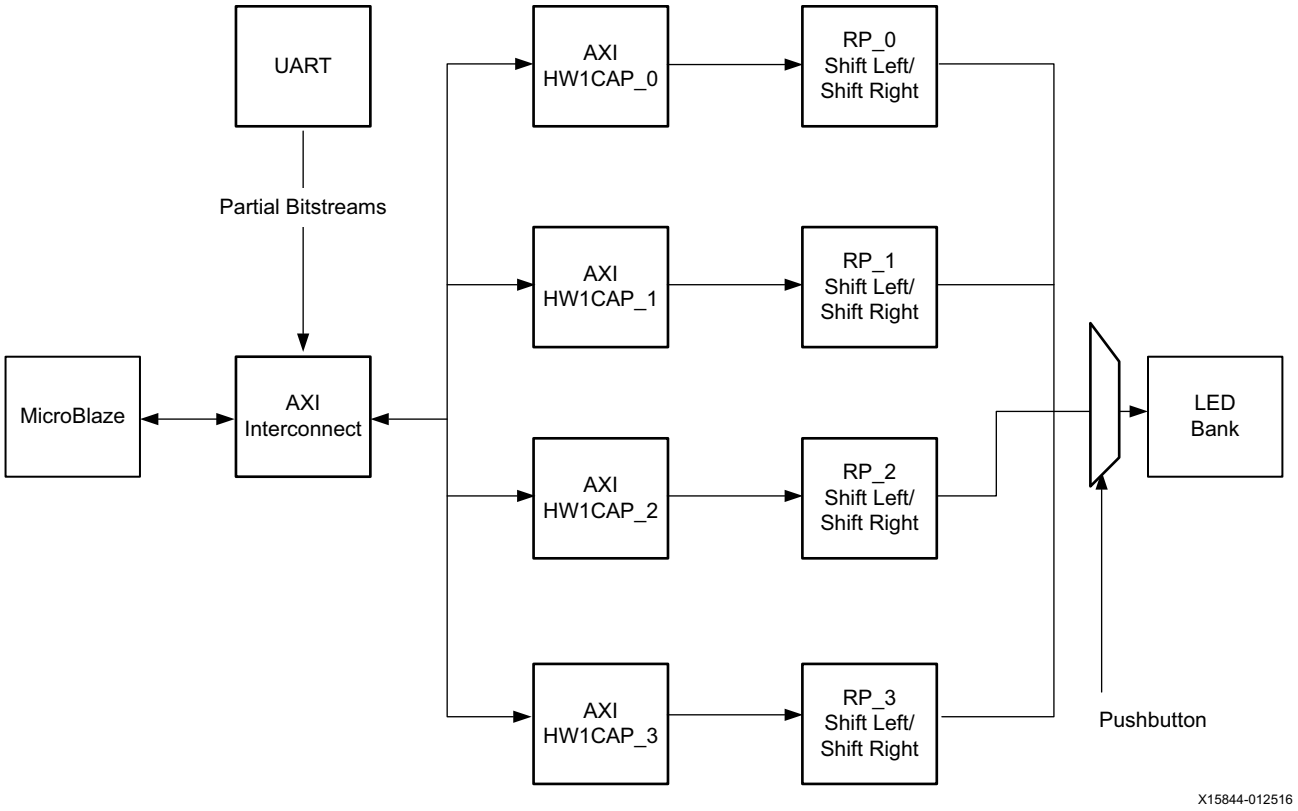


図 16 : フル デザインのブロック図

リコンフィギュレーションが正常に実行されていることを確認するには、選択したローカル ICAP を使用して、LED パターンが左シフトから右シフトのパターンに、または右シフトから左シフトのパターンに変わることをモニターします。次の手順に従って、このデザインをハードウェア上で実行します。

1. Tera Term を開き、[Serial] をオンにして [Port] に関連ポートを選択し (図 17)、ボー レートを設定します (図 18)。

図 17 : Tera Term での通信ポートの選択

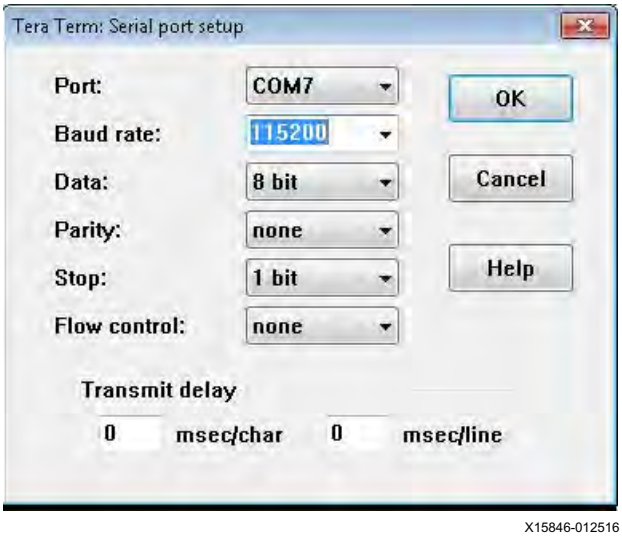


図 18 : 通信ポートのボー レートの設定

2. Vivado ハードウェア マネージャーを使用して、最初のコンフィギュレーション ビットストリームをデバイスにロードします。『Vivado Design Suite チュートリアル : パーシャル リコンフィギュレーション』(UG947) [参照 6] にある例に従ってアプリケーション ソフトウェアを実行します。ソフトウェア メニューがターミナルに表示されます (図 19)。

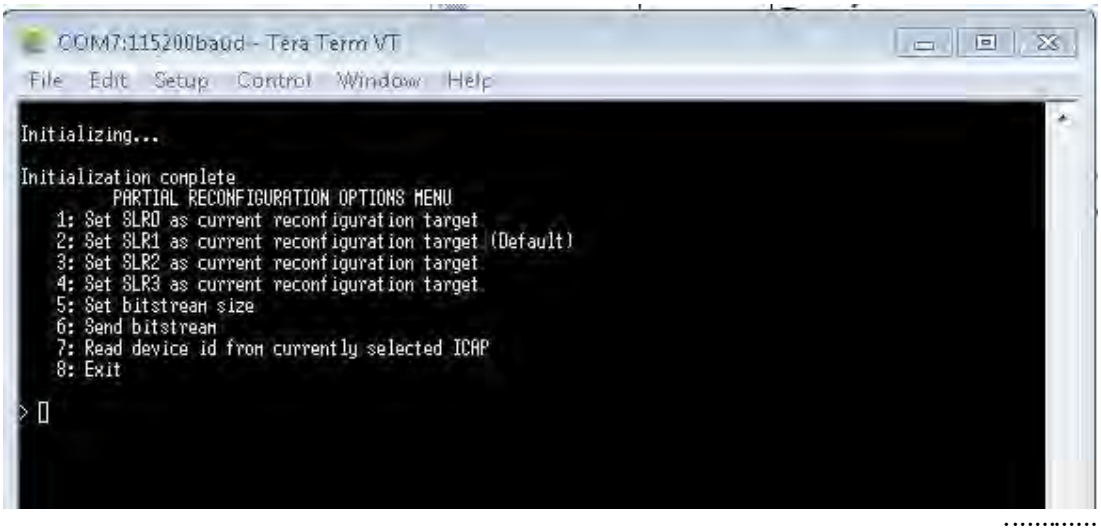


図 19 : Tera Term に表示されるユーザー アプリケーション メニュー

3. Tera Term で、メニューに従ってリコンフィギュレーション対象の SLR を設定し、パーシャル ビットストリーム サイズを入力します。
4. メニューから [Send bitsteam] を選択すると、Tera Term を使用してパーシャルを送信するように指示するメッセージが表示されます。

注意 : この段階では、これ以上キーを押さないでください。シリアル ポートで送信されたバイトが ICAP に書き込まれ、リコンフィギュレーションを無効化してしまいます。その場合は、ボードの電源を切って入れ直し、ハードウェア テストを再開します。

ヒント : これと同じソフトウェア アプローチは、連結されたクリアなパーシャル ビットストリームを使用するか、または個々のロードにビットストリーム サイズを設定するかにより、UltraScale デバイスをターゲットにしているデザインにも使用できます。

5. Tera Term で [File] → [Send file] をクリックし、パーシャル BIN ファイルを選択して [Binary] がオンになっていることを確認します (図 20)。

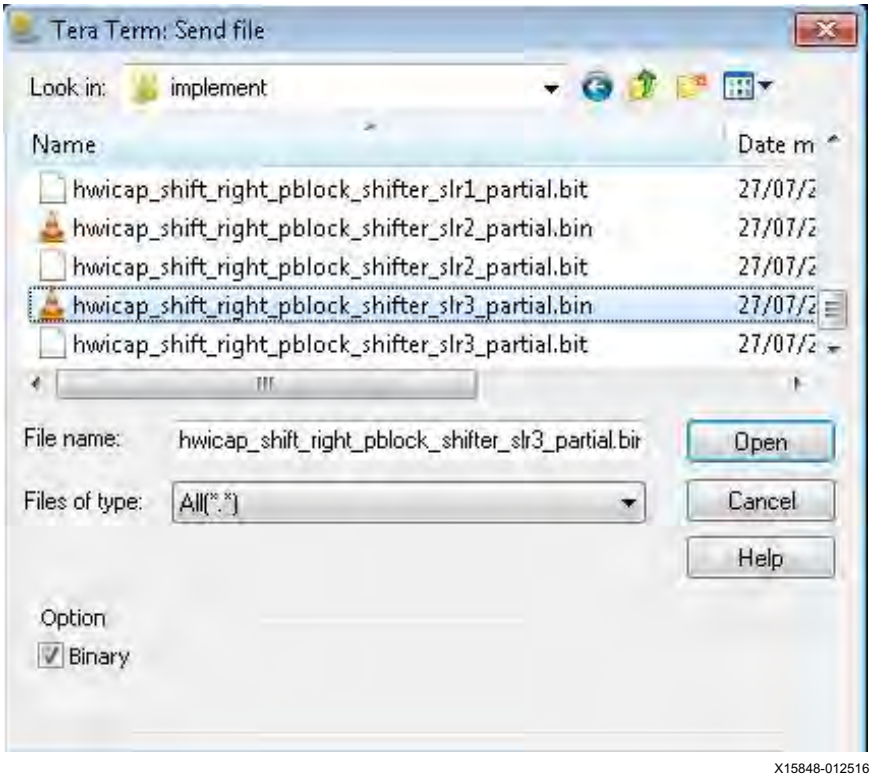


図 20 : UART 接続経由の Bin ファイルの送信

6. [Open] をクリックすると、ファイル転送ステータスが表示されます。
7. 転送が完了すると、MicroBlaze システム アプリケーションによりファイルが完全に送信されたことが通知され、ユーザー メニューが再表示されます。
8. 最初の RP で LED が左シフト パターンで点灯し、2 番目の RP で右シフト パターンで点灯するかを確認することで、正しいパーシャル ビットストリームがロードされたことを視覚的に検証できます。

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、ザイリンクスのウェブサイトからダウンロードできます。

表 1 に、リファレンス デザインの詳細を示します。これは、リファレンス デザインで使用されるツール フローおよび検証手順を示しています。

表 1 : リファレンス デザインの詳細

パラメーター	説明
一般	
開発者	ザイリンクス
ターゲット デバイス	XC7V2000T
ソース コードの提供	あり
ソース コードの形式	VHDL/Verilog
既存のザイリンクス アプリケーション ノート/リファレンス デザイン、またはサードパーティからデザインへのコード/IP の使用	Vivado Design Suite
シミュレーション	
論理シミュレーションの実施	なし
タイミング シミュレーションの実施	なし
論理シミュレーションおよびタイミング シミュレーションでのテストベンチの利用	なし
テストベンチの形式	
使用したシミュレータ/バージョン	
SPICE/IBIS シミュレーションの実施	なし
インプリメンテーション	
使用した合成ツール/バージョン	
使用したインプリメンテーション ツール/バージョン	
スタティック タイミング解析の実施	なし
ハードウェア検証	
ハードウェア検証の実施	あり
使用したハードウェア プラットフォーム	

リソース使用量

表 2 に、リファレンス デザインで使用される FPGA リソース数をまとめています。かっこ外の数値は単一ブロックのものです。かっこ内の数値は 4 つのモジュールを使用するデザインのものです。

表 2 : スタティック パーティションとリコンフィギャラブル パーティションのリソース使用量

リソース	スタティック パーティション	左シフトの単一 RP (4 つの RP の合計)	右シフトの単一 RP (4 つの RP の合計)
LUT	2876	12 (48)	11 (44)
フリップフロップ	5201	33 (132)	33 (132)
ブロック RAM	36	0	0

参考資料

注記 : 日本語版のバージョンは、英語版より古い場合があります。

1. 『Vivado Design Suite ユーザー ガイド : パーシャル リコンフィギュレーション』(UG909 : 英語版、日本語版)
2. 『高集積度 FPGA 設計手法ガイド (SSI テクノロジを含む)』(UG872 : 英語版、日本語版)
3. 『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』(UG470 : 英語版、日本語版)
4. 『Partial Reconfiguration Controller v1.0 LogiCORE IP 製品ガイド』(PG193 : 英語版、日本語版)
5. 『Vivado Design Suite チュートリアル : エンベデッド プロセッサ ハードウェア デザイン』(UG940)
6. 『Vivado Design Suite チュートリアル : パーシャル リコンフィギュレーション 』(UG947)
7. ザイリンクス ソフトウェア開発キット (SDK)
8. Tera Term ホーム ページ
9. Silicon Labs ウェブサイト : CP210x USB - UART ブリッジ VCP ドライバー

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
2016 年 3 月 2 日	1.0	初版

法的通知

本通知に基づいて貴殿または貴社 (本通知の被通知者が個人の場合には「貴殿」、法人その他の団体の場合には「貴社」。以下同じ) に開示される情報 (以下「本情報」といいます) は、ザイリンクスの製品を選択および使用することのためにのみ提供されます。適用される法律が許容する最大限の範囲で、(1) 本情報は「現状有姿」、およびすべて受領者の責任で (with all faults) という状態で提供され、ザイリンクスは、本通知をもって、明示、黙示、法定を問わず (商品性、非侵害、特定目的適合性の保証を含みますがこれらに限られません)、すべての保証および条件を負わない (否認する) ものとします。また、(2) ザイリンクスは、本情報 (貴殿または貴社による本情報の使用を含む) に関係し、起因し、関連する、いかなる種類・性質の損失または損害についても、責任を負わない (契約上、不法行為上 (過失の場合を含む)、その他のいかなる責任の法理によるかを問わない) ものとし、当該損失または損害には、直接、間接、特別、付随的、結果的な損失または損害 (第三者が起こした行為の結果被った、データ、利益、業務上の信用の損失、その他あらゆる種類の損失や損害を含みます) が含まれるものとし、それは、たとえ当該損害や損失が合理的に予見可能であったり、ザイリンクスがそれらの可能性について助言を受けていた場合であったとしても同様です。ザイリンクスは、本情報に含まれるいかなる誤りも訂正する義務を負わず、本情報または製品仕様のアップデートを貴殿または貴社に知らせる義務も負いません。事前の書面による同意のない限り、貴殿または貴社は本情報を再生産、変更、頒布、または公に展示してはなりません。一定の製品は、ザイリンクスの限定的保証の諸条件に従うこととなるので、http://japan.xilinx.com/legal.htm#tos で見られるザイリンクスの販売条件を参照してください。IP コアは、ザイリンクスが貴殿または貴社に付与したライセンスに含まれる保証と補助的条件に従うことになります。ザイリンクスの製品は、フェイルセーフとして、または、フェイルセーフの動作を要求するアプリケーションに使用するために、設計されたり意図されたりしていません。そのような重大なアプリケーションにザイリンクスの製品を使用する場合のリスクと責任は、貴殿または貴社が単独で負うものです。http://japan.xilinx.com/legal.htm#tos で見られるザイリンクスの販売条件を参照してください。

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、または各ページの右下にある [フィードバック送信] ボタンをクリックすると表示されるフォームからお知らせください。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。