XAPP1159 (v1.0) 2013 年 1 月 21 日

Zynq-7000 All Programmable SoC デバイスにおけるハードウェア アクセラレーターのパーシャル リコンフィギュレーション

著者 : Christian Kohn

概要

システムが複雑化し、設計の効率化が要求されるようになるのに伴い、FPGA が果たす役割が重要になってきています。ザイリンクス FPGA は、オンサイトで再プログラムできるという優れた柔軟性を備えていますが、低コスト、省ボード スペース、低消費電力といったニーズが高まる今日では、さらに効果的なデザイン ソリューションが求められています。

ザイリンクスのパーシャル リコンフィギュレーション (PR) 機能を利用することにより、変更不要なデバイス部分にあるアプリケーションの動作を継続しながら特定部分のみを再プログラムして新たな機能を追加できるため、FPGA 本来の柔軟性がさらに高まります。パーシャル リコンフィギュレーションは、従来のフル コンフィギュレーションに比べて、主に次のようなメリットがあります。

- ハードウェア リソースの使用率を低減 : デザインの機能を動的に時分割多重化することによって、既存デバイスにより多くのロジックを収容できる
- 生産性と拡張性を向上 : 検証済みの変更不要な部分を残し、変更した機能のみをインプリメントできる
- 保守性が高まり、システム ダウンタイムが削減 : システム動作を停止することなく、新しい機能を配置および挿入できる

このアプリケーション ノートでは、デバイス コンフィギュレーション (DevC)/プロセッサ コンフィギュレーション アクセス ポート (PCAP) インターフェイスを介して、ザイリンクス Zynq™-7000 All Programmable SoC デバイスでパーシャル リコンフィギュレーションを実行する際のツール フロー、概念、およびテクニックについて説明します。ここで提供するリファレンス デザインは、ZC702 ベース ターゲット リファレンス デザイン (TRD) [参照 1] を基盤に構築したエンベデッド ビデオ プロセッシング アプリケーションであり、演算量の多いビデオ フィルター アルゴリズムをプログラマブル ロジックに実装することによって、ソフトウェアの負担を軽減できるという利点を実証しています。このデザインでは、ソフトウェア制御のパーシャル リコンフィギュレーション機能を用いて、2 種類のビデオ フィルター IP コアを 1 つずつロジックの特定部分に動的にリコンフィギュレーションし、モニターに出力される映像を確認する方法を説明します。

はじめに

Zynq-7000 AP SoC は、デュアルコア ARM Cortex-A9 ベースのプロセッシング システム (PS) とプログラマブル ロジック (PL) を 1 つのデバイスに統合したものです。このリファレンス デザインは、PS と PL を共に活用し、制御 (PS へマップ) とデータ パス (PL へマップ) を分割することによって、最良の結果が得られることを実証します。PL には、入力、プリ プロセッシング、コア プロセッシング、ポスト プロセッシング、そして出力ステージを構成するパワフルな高精細度ビデオ パイプラインを実装します (図 1 参照)。IP コアの設定やデータ フロー制御は、PS で実行されます。付属のリファレンス デザインでは、PS 上で動作するベアメタル ソフトウェア アプリケーションか 2 つの Linux OS ベース ソフトウェア アプリケーションからユーザーが選択可能です。

ZC702 ベース TRD [参照 1] は、演算が集中する輪郭検出アルゴリズム (Sobel フィルター) を PL で実現するメリットを示します。この場合、ハードウェア アクセラレーションにより、1080p60 ビデオ ストリームをリアルタイム処理できるだけでなく、処理済みビデオ ストリーム上にグラフィカル ユーザー インターフェイス (GUI) をレンダリングするなどのユーザー固有のタスクに CPU リソースを使用できる、という二重のメリットがあります。

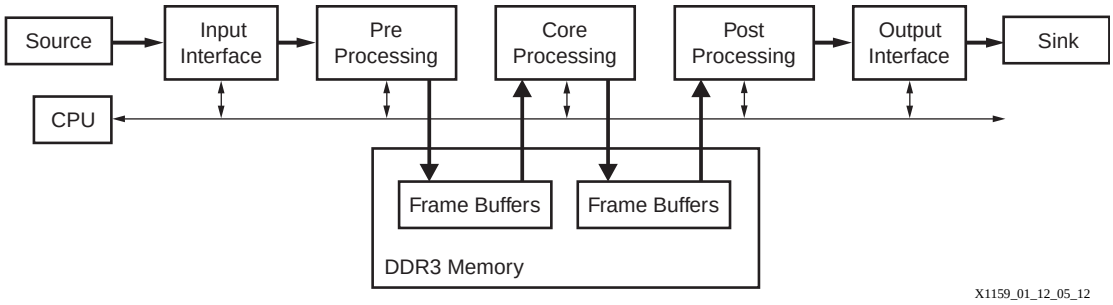

図 1 : ビデオ処理パイプライン

ZC702 ベース TRD を基盤とするこのアプリケーション ノートでは、任意の機能を必要に応じてロードできるパーシャル リコンフィギュレーション機能を使用して、2 種類のビデオ処理アクセラレーターを PL にインプリメントする方法について説明します。1 つ目のビデオ フィルター IP コアは、入力ビデオ ストリームで輪郭を検出して表示する Sobel フィルターです。2 つ目のビデオ フィルター IP コアは、入力ビデオ ストリームに茶色がかった単色を適用する Sepia フィルターです。図 2 に、オリジナル画像、Sobel 処理された画像、Sepia 処理された画像をそれぞれ示します。2 つのビデオ フィルター IP コアの RTL は、Vivado™ HLS (高位合成) ツールを使用して C 言語のアルゴリズム記述から生成されます。Sobel フィルター アルゴリズム、Vivado HLS ツール フロー、Linux デバイス ドライバー、およびシステム統合の詳細は、『Vivado HLS ツールを使用した Zynq All Programmable SoC での Sobel フィルターの実装』(XAPP890) [参照 2] を参照してください。Sepia フィルターにも、同様の概念を適用できます。

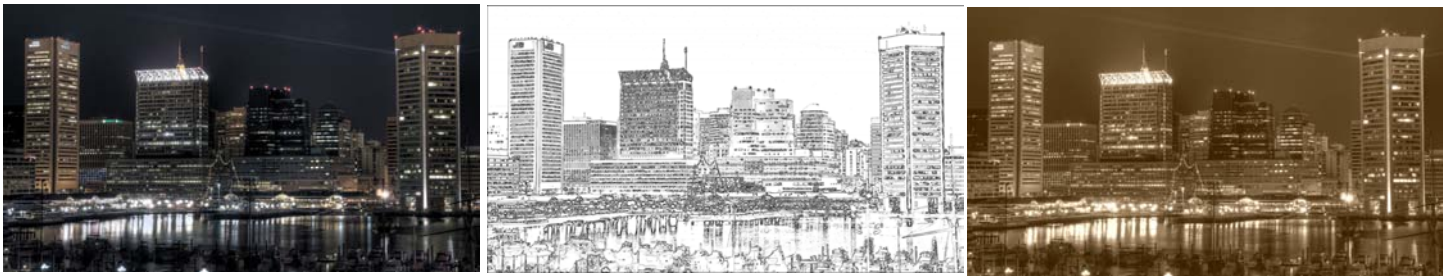

図 2 : オリジナル画像 (左)、Sobel 処理済み画像 (中央)、Sepia 処理済み画像 (右)

このアプリケーション ノートでは、ユーザーに ZC702 ベース TRD [参照 1] とそのハードウェア/ソフトウェアに関する知識があることを前提に説明します。主に、次の内容に焦点を当てています。

- パーシャル リコンフィギュレーションに最適な機能
- リコンフィギャラブル モジュールの設計上の注意事項とインターフェイス要件
- パーシャル リコンフィギュレーション デザイン フローの詳細手順
- PS DevC/PCAP インターフェイスを介してソフトウェア制御のパーシャル リコンフィギュレーションを実行

システムの概要

このセクションでは、付属するリファレンス デザインにおけるプログラマブル ロジック (PL) とプロセッシング システム (PS) について、システム レベルで説明します。図 3 に、リファレンス デザインのブロック図を示します。

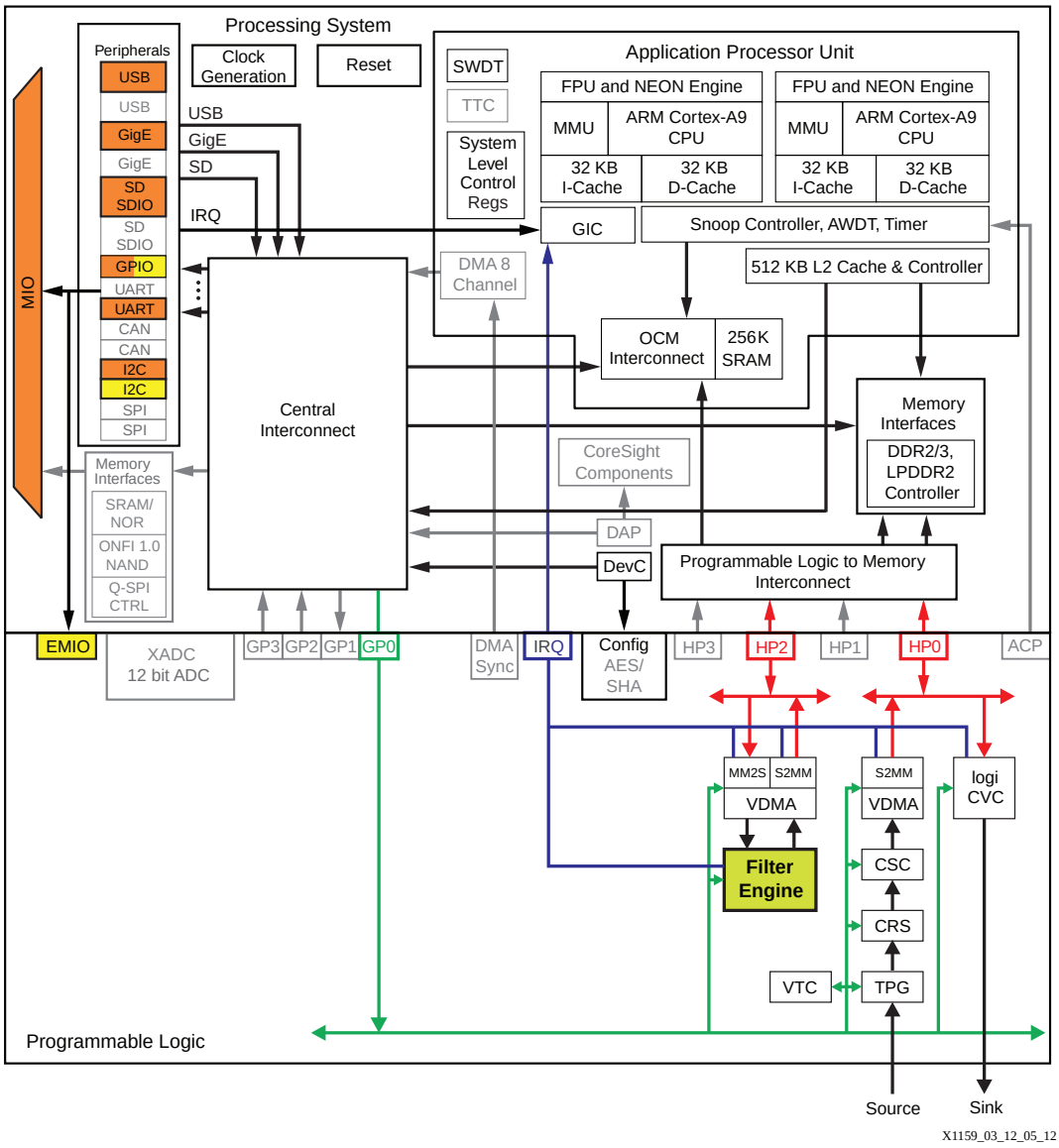

図 3 : PR リファレンス デザインのシステム レベル ブロック図

プログラマブル ロジック

PL リファレンス デザインには、次の IP コアが含まれています。

- AXI Interconnect [参照 14]
- AXI Video DMA (VDMA) [参照 15]
- Xylon Compact Video Controller (logiCVC) [参照 16]
- AXI Performance Monitor [参照 17]
- Video Timing Controller (VTC) [参照 18]
- Video Test Pattern Generator (TPG) [参照 19]
- Chroma Resampler (CRS) [参照 20]
- YCrCb to RGB Color-Space Converter (CSC) [参照 21]
- Video In to AXI4-Stream [参照 22]
- Filter Engine : Sobel または Sepia フィルター(1)

1. Vivado HLS で生成された IP コアです。

図 3 のプリ プロセッシング パイプラインは、VTC、TPG、CRS、CSC、および VDMA で構成されています。VTC は、TPG が FPGA 内でテスト パターンを生成するのに必要なビデオ タイミング信号を生成します。オプションで、FMC ドーター カードを使用して外部ビデオ入力ソースを接続できます。いずれの場合も、入力ビデオ フォーマットは YCbCr 4:2:2 です。このフォーマットは CRS および CSC ブロックで RGB 4:4:4 に変換されます。そして、VDMA が、入力されるビデオ ストリームを DDR メモリ内にある専用のビデオ フレーム バッファーへ書き込みます。

コア プロセッシング パイプラインは、Filter Engine と VDMA (2 つ目) で構成されています。コア プロセッシング パイプラインを無効にすると、オリジナルのビデオ入力ストリームを表示できます。有効にした場合は、VDMA が DDR からビデオ フレームを読み出し、Filter Engine へ転送します。ビデオ フレームは Filter Engine 内で処理された後、VDMA によって DDR メモリへ書き戻されます。この Filter Engine (薄緑色のボックス) が、このデザインでリコンフィギュレーションされるブロックとなります。特定のインプリメンテーションを言及していない場合には、Filter Engine という名称を用います。このブロックにインプリメンテーション可能な 2 つのフィルターは、Sobel フィルターと Sepia フィルターです。リファレンス デザインでは、Sobel フィルターを最初のフィルターとして使用しています。Sobel/Sepia フィルター モジュールは、パーシャル リコンフィギュレーション機能を使用して、動作中にロードできます。

ポスト プロセッシング パイプラインは、logiCVC ディスプレイ コントローラーで構成されています。この部分には、DDR メモリからフレーム バッファーを読み出すための統合 DMA エンジンと、複数ビデオ ストリームをアルファブレンドするためのマルチレイヤー アルファブレンダーが含まれます。ビルトインのカラー スペース コンバーターおよびクロマ リサンプラ ユニットが、ビデオ データを YCbCr 4:2:2 出力ビデオ フォーマットへ変換します。出力ビデオ ストリームは、HDMI を介してモニター ディスプレイへ転送されます。

プロセッシング システム

プロセッシング システム (PS) では、図 3 に示す I/O ペリフェラル (USB0、Ethernet0、SD0、UART1、I2C0、I2C1、および GPIO) が有効に設定されています。EMIO (黄色のボックス) 用に構成されている I2C1 および一部の GPIO を除き、すべての I/O ペリフェラル インターフェイスは MIO (オレンジ ボックス) 用に構成されています。I2C0 コントローラーは、ZC702 ボード上の ADV7511 HDMI トランスミッター (正確なビデオ クロックを供給) と SI570 クロック シンセサイザーを設定するために使用されます [参照 3]。I2C1 コントローラー (信号は PL を介して配線) は、Avnet 社製 FMC-IMAGEON ドーター カード上の ADV7611 HDMI レシーバーを設定するために使用されます (オプション) [参照 4]。GPIO 信号は PL へ配線され、TPG のリセット、Filter Engine のリセット、および TPG と (FMC からの) 外部ビデオ入力のマルチプレクスに使用されます。割り込み信号は、3 つの VDMA チャネル、logiCVC、および Filter Engine へ接続されます。汎用マスター ポートの GP0 は、メモリ マップされた IP コアを AXI4-lite を介して設定および制御するために使用されます。高性能ポートの HP0 および HP2 は、AXI4 を介して接続されます。つまり、コア プロセッシング パイプラインが HP2 読み出し/書き込みチャネルへ、入力/プリプロセッシング パイプラインが HP0 書き込みチャネルへ、そして出力/ポストプロセッシング パイプラインが HP0 読み出しチャネルへ接続されます。PS の内部クロック ジェネレーターが PL へ 100MHz クロック (図には非表示) を供給し、PL では 150MHz クロック (データ パス) および 75MHz クロック (制御パス) が生成されます。未使用の I/O ペリフェラルおよび PL インターフェイスは、グレーアウト表示されています。

ハードウェア デザイン

ZC702 ベース TRD のハードウェア デザインの詳細は、『Zynq-7000 All Programmable SoC ZC702 ベース ターゲット リファレンス デザイン ユーザー ガイド』(UG925) [参照 1] を参照してください。基本的には、このアプリケーション ノートで提供するリファレンス デザインと次の点で同じです。

- PS のコンフィギュレーションと主な機能
- IP コアのコンフィギュレーション、アドレス マップ、および主な機能
- PL ペリフェラル用の割り込み ID

- GPIO 信号のマッピング
- クロッキングおよびリセットの構造

リコンフィギャラブル モジュールの設計における留意事項

このセクションでは、パーシャル リコンフィギュレーションを実行する IP を選択および設計する際の留意事項について説明します。また、リファレンス デザインの Filter Engine コアがパーシャル リコンフィギュレーションに適する理由についても述べています。

用語

このアプリケーション ノートで使用する用語について説明します。リコンフィギャラブル パーティション (RP) とは、パーシャル リコンフィギュレーション用に選択された FPGA の物理的な部分のことです。その他の部分は、スタティック ロジックといいます。Filter Engine は、ハードウェア インターフェイスとポートで定義される一般的なソケットであり、RP にマップされます。このソケットへは、任意のフィルター インプリメンテーションをプラグインできるため、リコンフィギャラブル モジュール (RM) と呼ばれます。このデザインでは、2 つの RM (Sobel フィルターと Sepia フィルター) が提供されています。スタティック ロジックと RM を併せてコンフィギュレーションといいます。このデザインで提供する 2 つのコンフィギュレーションは、Sobel コンフィギュレーションと Sepia コンフィギュレーションです。コンフィギュレーションとは、完成した FPGA デザインを表し、RM とスタティック ロジック用のフル ビットストリームおよび RM 用のパーシャル ビットストリームを生成します。

フィルター動作

デザインにおける選択をより具体的に理解するため、Filter Engine IP コアの動作を確認します。このコアには AXI4-Lite インターフェイスがあり、これは PS がコアの動作を設定および制御するために使用されます。最初にこの IP コアを現ビデオ ストリームのビデオ サイズ (幅と高さ) で設定する必要があります。その後、コアが開始し、すべてのビデオ フレームを処理した後に割り込みが発行されます。割り込みハンドラーが割り込みソースを特定し、その後コアを再開します。これで次のビデオ フレームを処理できるようになります。このようにして、ビデオ コアは入力されるビデオ ストリームをフレームごとに継続的に処理します。ユーザーは、現フレームが処理された後にコアを再開させないよう割り込みハンドラーへ命令するフラグをセットすることで、コアの動作を停止できます。さらに、コアのリセット ラインへ接続された、対応する PS-GPIO 信号を駆動することによって、コアをリセットできます。つまり、ユーザーがコアの動作 (設定、開始、停止、リセット) を管理できます。

ハードウェア インターフェイス

Filter Engine IP コアは、システムのスタティック部分に対して次に示すハードウェア インターフェイスおよび接続を備えています。

- GP0 マスター インターフェイスへ接続された 32 ビットの AXI4-Lite インターフェイス
- VDMA MM2S マスター インターフェイスへ接続された 32 ビットの AXI4-Stream インターフェイス
- VDMA S2MM スレーブ インターフェイスへ接続された 32 ビットの AXI4-Stream インターフェイス
- PS-GIC へ接続された割り込み信号
- PS-GPIO コントローラーへ接続されたリセット信号
- AXI4-Lite および AXI4-Stream インターフェイスと同じクロック ドメイン (このデザインでは 150MHz) へ接続されたクロック信号

パーシャル インターフェイス用の IP コアを選択する際には、すべての RM のハードウェア インターフェイスおよびポートがシステムのスタティック部分に対して同一であることを確認する必要があります。RP フローでは、デザインの RP とスタティック部分の境界にポートが挿入されます。したがって、すべての RM (つまり、RP へマップされた特定インプリメンテーションすべて) は、スタティック ロジックに対して同じ位置にある 1 つのポートを共有する必要があります。

インターフェイスの非干渉化

リコンフィギャラブル ロジックは FPGA デバイスが動作している間に変更されるため、パーシャル リコンフィギュレーション中は、RM の出力に接続されたスタティック ロジックは RM からのデータを無視する必要があります。RM は、パーシャル リコンフィギュレーションが完了して再コンフィギュレーションされたロジックがリセットされるまで有効な値を出力しません。一般的な対応策として、すべての出力信号にレジスタを介したり、ハンドシェークを使用する、あるいは無効なトランザクションを回避するためにバス インターフェイスを無効化する方法があります。スタティック ロジックには、データおよびインターフェイスを管理するロジックを含める必要があります。

このリファレンス デザインでは、AXI インターフェイスおよび割り込みラインに注意する必要があります。パーシャル リコンフィギュレーション中は、AXI インターフェイス上に保留中のトランザクションがないように、設計者の責任で十分注意を払う必要があります。このようなトランザクションがあると、AXI バスが停止する可能性があります。また、リコンフィギュレーション中は GCI で割り込みハンドラーを無効にすることも推奨します。割り込みハンドラーは、完全にロードされていないフィルター モジュール内のハードウェア レジスタへアクセスするなど、割り込みがシステムを停止させるような不正な割り込みや処理を行う可能性があります。このリファレンス デザインでは、ソフトウェアを使用して上記の状況を回避しています。場合によっては、これらを適切に分離させるために、RM のハードウェア インターフェイスへハードウェア グルー ロジックを追加する必要があります。リコンフィギュレーション後、指定したステートへ RM を確実に遷移させるために、リコンフィギュレーション プロセスの直前、途中、および後にフィルター IP コアをリセットすることを推奨します。リコンフィギュレーション完了後、リセット信号がディアサートされて、コアのコンフィギュレーションと開始の準備が整います。詳細は、『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) [参照 5] の第 7 章「デザインの注意事項」を参照してください。

ドライバー アーキテクチャ、レジスタ インターフェイス、アドレス マップ

Filter Engine IP コアは、一般的なハードウェア レジスタ インターフェイスを 1 つ実装し、同一のメモリ アドレス マップを使用します。この場合、要件が厳しくない上に、ソフトウェア ドライバーのアーキテクチャが非常にシンプルになります。このデザインで唯一コンフィギュレーション可能なフィルター コアのパラメーターは、両方の RM に共通するビデオ サイズです。より高機能な例になると、Sobel フィルターは輪郭検出の感度を制御する係数でプログラム可能になり、Sepia フィルターは、セピア色以外の色付け効果をもたらすなど個別にカラー コンポーネントを制御するパラメーターを用いてプログラムできるようになります。そのような場合、対応するドライバーは、すべての RM が共有する基本機能を提供する一般的なコア ドライバーと、多様なインプリメンテーションに独自機能を提供するフィルター固有のドライバーに分けられることがあります。今回のリファレンス デザインでは、Sobel フィルターと Sepia フィルターのインプリメンテーションに、まったく同じドライバーが使用可能です。

全体的な設計フロー

パーシャル リコンフィギュレーション可能な FPGA デザインをインプリメントすることは、共通のロジックを共有する通常のリコンフィギュレーション デザインを複数インプリメントすることと同じです。PlanAhead™ ツールを使用する場合、PR 可能なプロジェクトはネットリスト レベルで開始します。

事前に必要な手順

必要なネットリストとそれに対応する制約ファイルを生成するには、このリファレンス デザインでは次の 3 つの手順を実行します。

1. Vivado HLS ツールを使用して、Sobel フィルターおよび Sepia フィルターの IP コアとそれらに対応するスタンドアロンまたはベアメタル ドライバーを生成します。Vivado HLS では、RTL コードを生成することによって、C、C++、または System-C 言語で記述されたアルゴリズムを Zynq PS から PL へ移行できます。生成された RTL コードは、XPS pcode としてエクスポートでき、その後 XPS プロジェクトへ簡単に適用できます。Vivado HLS を使用した Sobel フィルター インプリメンテーション フローのチュートリアルは、『Vivado HLS ツールを使用した Zynq All Programmable SoC での Sobel フィルターの実装』(XAPP890) [参照 2] を参照してください。Sepia

フィルターに対しても同じ手法を適用できます。

2. ハードウェア デザインは、ZC702 ベース TRD を基盤とする PlanAhead/XPS プロジェクトとして提供され、Sobel フィルターの pcore (最初の手順で取得) が既にインスタンシエートされています。Sobel フィルター デザインは合成されて、システム全体および各 IP コアそれぞれのネットリストと制約ファイルを生成します。
3. 最初の合成が実行された後、Sobel フィルターの pcore は Sepia フィルターの pcore に置き換えられ、新たなデザインが再合成されます。この実行では、Sepia フィルター IP コア用のネットリストと制約ファイルを取得します。その他すべてのネットリストおよび制約ファイルは、前の手順で取得済みです。

パーシャル リコンフィギュレーションの設計フロー

PlanAhead ツールのパーシャル リコンフィギュレーションは、ライセンスが必要な特別な機能です。詳細は、ザイリンクスのパーシャル リコンフィギュレーションに関するウェブ ページ [参照 13] を参照してください。

次に PlanAhead PR 設計フローの各手順について簡単に説明します。

1. ZC702 評価プラットフォームをターゲットとした PlanAhead PR プロジェクトを作成し、「事前に必要な手順」セクションの手順 2 と 3 で生成した Sobel および Sepia フィルター用のネットリスト/制約ファイルを除いた、その他のネットリストと制約ファイルをインポートします。これらは、デザインのスタティック ロジックを表します。
2. 合成したデザインをロードする場合、Filter Engine モジュールには関連するネットリストがないため、これはブラック ボックスとして扱われます。Filter Engine モジュールをリコンフィギャラブル パーティション (RP) として定義します。このパーティションによって、各複数デザインに共通するロジックと配線が確実に同じになります。
3. Sobel フィルター コアおよび Sepia フィルター コアそれぞれのネットリスト/制約ファイルを追加し、すでに作成した RP に対して 2 つのリコンフィギャラブル モジュール (RM) を作成します。特定 RM にのみ適用する制約は、モジュール レベルで適用し、対応するネットリストと共に提供する必要があります。また、スタティック ロジックへ適用する制約およびすべての RM で共有される制約は、トップレベルの制約ファイルに含める必要があります。
4. パーティションの物理的なサイズや必要なリソース タイプを設定して、リコンフィギャラブル パーティションのフロアプランを作成します。ザイリンクス FPGA は、CLB (フリップフロップ、LUT、分散 RAM、マルチプレクサなど)、BRAM、DSP ブロック、および関連するすべての配線リソースのリコンフィギュレーションをサポートしています。設計者は、RM に必要なリソースを収容できるように RP をフロアプランする必要があります (表 1 参照)。配線リソースに関しては、約 20% のオーバーヘッドを考慮しておく必要があります。スタティック ロジックに対して、PL 内に RP を配置する位置は、データ フローや RM がその他のデザイン部分とどのように関連しているかによって異なります。シンプルなストラテジは、フロアプランを行わずに最も高いリソース使用率でコンフィギュレーションをインプリメントし、大半のリソースが配置されている位置を特定して、その領域の周辺にこれらすべてのリソースを含めるのに十分な大きさのパーティションを作成することです。このとき、フレームに沿って RP を作成すると、最適な配置配線結果が達成できます。リコンフィギャラブル フレームとは、リコンフィギュレーションできる物理的な最小領域で、クロック領域の境界に対して垂直に揃っている部分です。7 シリーズ デバイスの場合、リコンフィギャラブル フレームは、高さが CLB 50 個分で幅が CLB 1 個分です。RP の物理領域は、プライマリ制約ファイルに AREA_GROUP RANGE 制約として格納されます。RP のフロアプラン方法の詳細は、『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) [参照 5] を参照してください。

表 1 : Sobel および Sepia の RM で必要なリソース

	RP	Sobel RM		Sepia RM	
サイト タイプ	利用可能な数	必要な数	使用率	必要な数	使用率
LUT	1600	987	62	547	35
FD_LD	3200	1007	32	613	20
SLICEL	250	149	60	76	31
SLICEM	150	99	66	62	42
DSP48E1	20	2	10	5	25
RAMBFIFO18E1	20	3	15	0	0
RAMBFIFO36E1	10	0	0	0	0

5. リコンフィギャラブル デザインのコンフィギュレーションを構築する場合には、最も実現が難しいと思われるコンフィギュレーションを最初に選択します。後続のコンフィギュレーションすべての RM はより小規模または低速になるため、要件が満たしやすくなります。表 1 に示す使用率に基づいて、まず最初に Sobel コンフィギュレーションをインプリメントします。PlanAhead では、RM の構築に使用されるリソースが、指定された RP 領域内に確実に含まれるように管理されるため、デザインのスタティック部分に障害が生じることはありません。インプリメンテーション完了後、スタティック部分のインプリメンテーション結果が Sepia コンフィギュレーションで再利用できるように、Sobel コンフィギュレーションを進めます。
6. Sepia コンフィギュレーションをインプリメントします。前の手順で生成された Sobel コンフィギュレーションからスタティック ロジックをインポートします。図 4 に、Sobel コンフィギュレーションと Sepia コンフィギュレーションの RP 内の物理的なリソースを比較する PlanAhead 画面を示します。

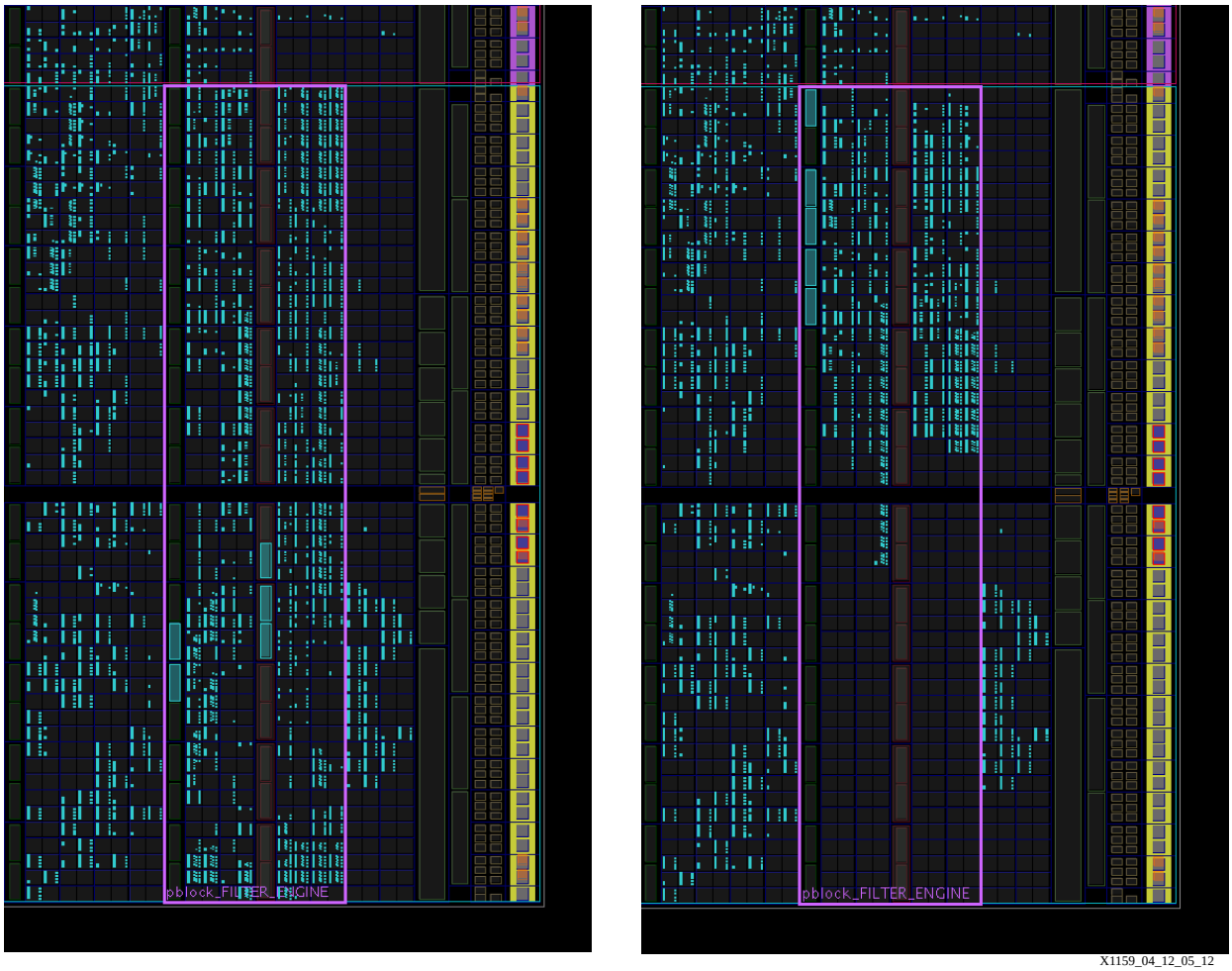

図 4 : PlanAhead 画面 - Sobel フィルター (左) および Sepia フィルター (右) で配置された RP リソース

7. PR Verify Configuration Utility を実行して、Sobel コンフィギュレーションと Sepia コンフィギュレーションのインプリメンテーションの一貫性を検証します。
8. Sobel コンフィギュレーションおよび Sepia コンフィギュレーションのフル ビットストリームとパーシャル ビットストリームを生成します。ここでは、Zynq デバイスを起動する際に、Sobel コンフィギュレーションのフル ビットストリームをデフォルトのスタートアップ コンフィギュレーションとして使用します。
9. PROMGen ツールを使用して、Sobel および Sepia のパーシャル ビットストリームをバイナリ形式に変換します。PROMGen は、DevC の DMA エンジンを使用して PL へコンフィギュレーション データを送信する際に必要となる、生成されたパーシャル バイナリ ファイルのサイズもレポートします。

図 5 に、リファレンス デザインを使用する全体的な設計フローを示します。濃いグレーのボックスは Sobel コンフィギュレーションを示し、最初の合成およびインプリメンテーション プロセスで使用されます。2 番目のプロセスでは、Sepia コンフィギュレーションが使用されます (薄いグレーのボックス)。結果として 4 つのビットストリームが生成されて、そのうちの 3 つ (実線枠) が最終的なアプリケーションで使用され、Sepia のフル ビットストリーム (破線枠) は破棄されます。各手順の詳細説明は、Zynq PR リファレンス デザインに関する Wiki ページ [参照 6] を参照してください。PR デザイン フローおよび設計上の注意事項に関する詳細は、『パーシャル リコンフィギュレーション ユーザー ガイド』(UG702) [参照 5] を参照してください。

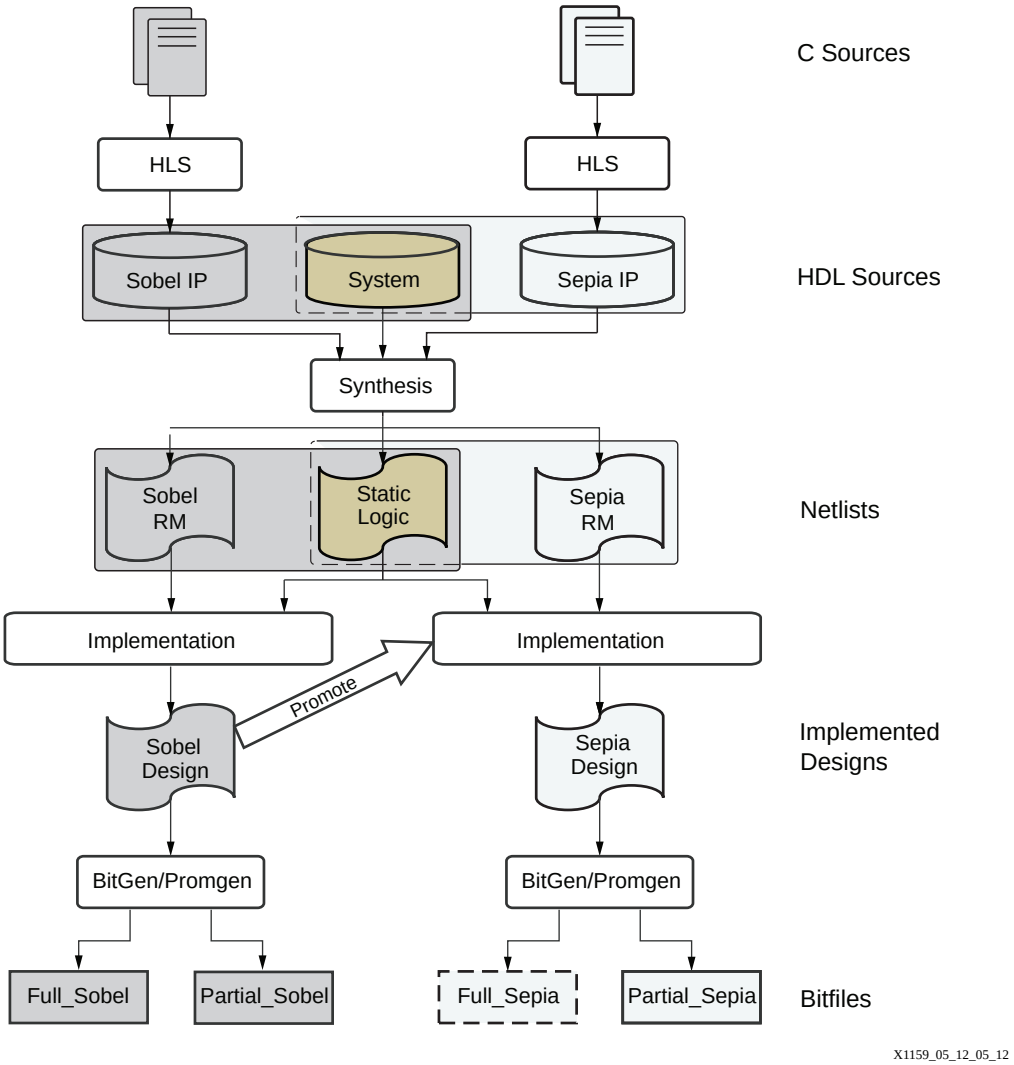

図 5 : PR リファレンス デザインの全体的な設計フロー

デバイス コンフィギュレーション

図 6 に示すように、Device Configuration インターフェイス (DevC) には、PL コンフィギュレーション ロジックとインターフェイスする AXI-PCAP ブリッジ (中央)、デバイスのセキュリティ管理を行う Security Management (左)、XADC インターフェイス (右) の3 つの主要ブロックがあります。このアプリケーション ノートに関係するブロックは、AXI-PCAP ブリッジのみです。

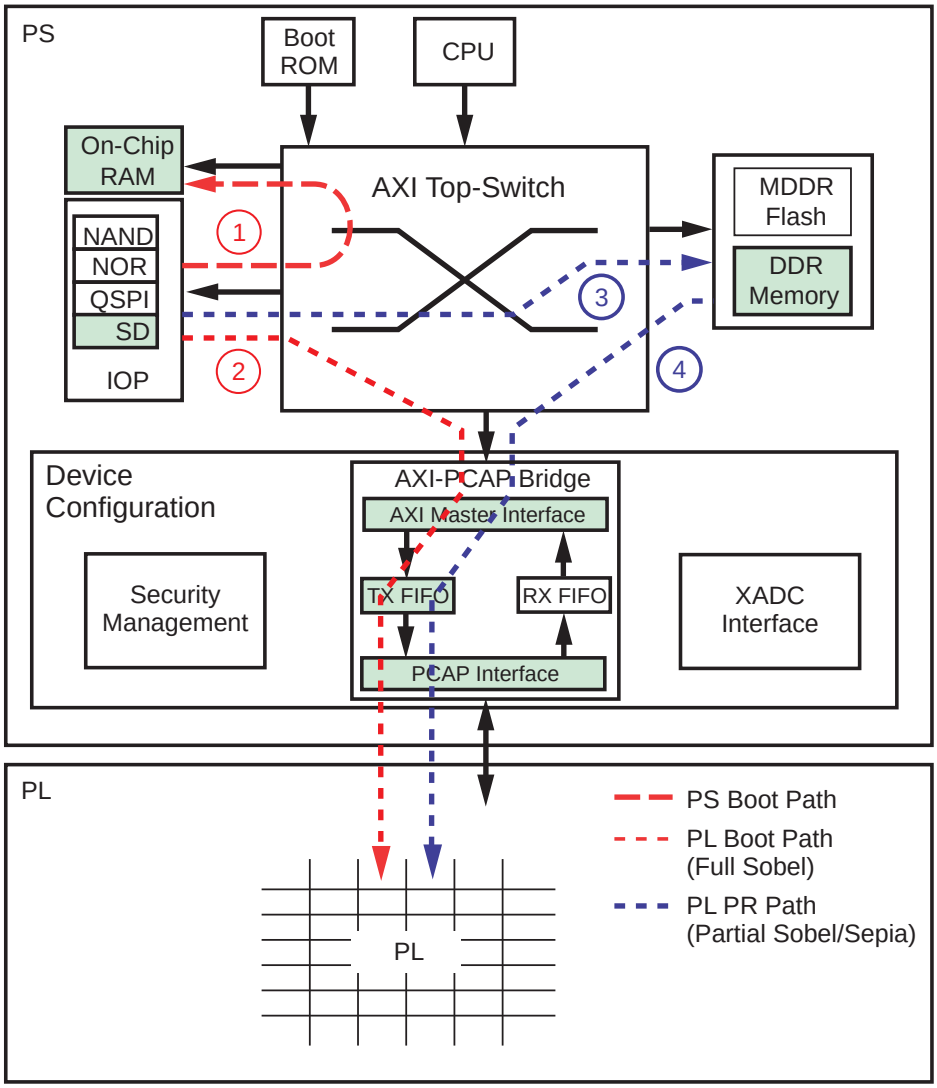

図 6 : デバイス コンフィギュレーション フロー (ブートおよびパーシャル リコンフィギュレーション)

AXI-PCAP ブリッジ

AXI-PCAP ブリッジでは、32 ビットの AXI 形式データを 32 ビットの PCAP プロトコルへ (またはその逆に) 変換します。TX および RX FIFO が AXI インターフェイスと PCAP インターフェイス間のデータをバッファリングします。FIFO とメモリ デバイス (通常は OCM、DDR メモリ、またはペリフェラル メモリのいずれか 1 つ) 間のデータ移動は、DMA エンジンで駆動されます。32 ビットの PCAP インターフェイスは 100MHz クロックで駆動されて、ノンセキュアの PL コンフィギュレーションの場合には 400MB/s のダウンロードをサポートし、4 クロック サイクルに 1 回のみデータが送信されるセキュアな PL コンフィギュレーションでは 100MB/s のスループットをサポートします。PCAP インターフェイスを介してデータ転送を行う場合は、DevC ドライバーの関数を呼び出す必要があります。このドライバーは適切な PCAP モードの選択と DMA 転送の開始を管理します。関数の呼び出しは、AXI 転送および PCAP 転送の両方が完了した後にのみ返されます。

デバイス コンフィギュレーションおよびブート フロー

デバイス コンフィギュレーションのフローとパーシャル リコンフィギュレーションのフローは、図 6 で説明しています。シーケンスは次のとおりです。

1. パワーオン リセット後、Boot ROM が外部メモリ インターフェイスまたはブート モード (SD フラッシュ メモリ) と暗号化ステータス (ノンセキュア) を決定します。Boot ROM は、DevC の DMA を使用して FSBL (第 1 段階ブート ローダー) を OCM (オンチップ RAM) へロードします。
2. Boot ROM が動作を終了すると、CPU の制御権は FSBL へ引き渡されます。そして FSBL が PCAP (プロセッサ コンフィギュレーション アクセス ポート) を介して Sobel フル ビットストリームを PL へロードします。これで、デバイスは完全にコンフィギュレーションされて動作可能な状態となります。
   a. スタンドアロン : FSBL がロード実行後、制御権はスタンドアロン ユーザー アプリケーションへ引き渡されます。
   b. Linux : FSBL がロード実行後、制御権は第 2 段階ブート ローダー (u-boot) へ引き渡されます。u-boot は Linux カーネル イメージ、Linux デバイス ツリー バイナリおよび Linux ルート ファイル システムをロードします。その後、制御権 は Linux カーネルへ引き渡されます。ブート プロセスの最後で、Linux Qt ユーザー アプリケーションが自動的に開始されます。
3. スタートアップ時に、ユーザー アプリケーションがパーシャル ビットストリームを DDR メモリへロードします。これは、PCAP インターフェイス経由でのコンフィギュレーションのスループットを最大限にするためです。これによってコンフィギュレーションが高速化し、キャッシュ機能を活用できます。
4. この時点で、アプリケーションは随時パーシャル ビットストリームを使用して、あらかじめ指定した PL 領域を変更できます。この間、FPGA のその他の部分はアクティブ状態を保ち、動作を継続できます。ここでは、PCAP を介して Sobel パーシャル ビットストリームまたは Sepia パーシャル ビットストリームのいずれかを DDR から PL へ転送することで、パーシャル リコンフィギュレーションを実行できます。

1 つのコンフィギュレーション エンジンがフル コンフィギュレーションとパーシャル コンフィギュレーションの両方を管理します。パーシャル ビットストリームにはコンフィギュレーション フレームのアドレス情報が含まれているため、これを PL へロードする際に、リコンフィギャラブル モジュールの物理的な位置を把握している必要はありません。ブートおよびコンフィギュレーションの詳細は、『Zynq-7000 AP SoC テクニカル リファレンス マニュアル (TRM)』(UG585) [参照 7] の第 6 章「ブートおよびコンフィギュレーション」、および『Zynq-7000 EPP ソフトウェア開発者向けガイド』(UG821) [参照 8] の第 3 章を参照してください。コンフィギュレーション全般に関する情報は、『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 9] を参照してください。

ソフトウェア アプリケーション

付属のリファレンス デザインでは、スタンドアロン/ベアメタル ソフトウェア アプリケーション、コマンド ライン ベースの Linu アプリケーション、Qt GUI ベースの Linu アプリケーションの 3 つのソフトウェア アプリケーションを提供しています。これら 3 つのアプリケーションはすべて同じ機能を備えており、ユーザーはオリジナルの未処理ビデオ ストリーム (コア プロセシング パイプラインをバイパス)、Sobel 処理されたビデオ ストリーム、Sepia 処理されたビデオ ストリームのいずれかの実行モードを選択できます。また、PL 内のテスト パターン ジェネレーター (TPG) と外部ビデオ ソースのいずれかも選択可能です。ただし、外部ビデオ ソースを選択する場合には、ZC702 評価ボード [参照 3] と FMC-IMAGEON ドーター カード [参照 4] が必要です。Linux ソフトウェア アプリケーションは、ZC702 ベース TRD ソフトウェアに基づいています。ソフトウェア アーキテクチャの詳細は、『Zynq-7000 All Programmable SoC ZC702 ベース ターゲット リファレンス デザイン ユーザー ガイド』(UG925) [参照 1] を参照してください。このアプリケーション ノートでは、ソフトウェア制御による Filter Engine のパーシャル リコンフィギュレーションを可能にするために強化された機能について説明します。

ソフトウェア制御フロー

付属のソフトウェア アプリケーションでは、固定のビデオ解像度を使用するため、ビデオ タイミングに対応するペリフェラル (VTC、logiCVC、および SI570 クロック シンセサイザー) はプログラム実行時に一度のみ設定します。Linux アプリケーションのクロック シンセサイザーは logiCVC フレーム

バッファー ドライバーで設定されますが、スタンドアロン アプリケーションの場合には個別に設定する必要があります。また、スタンドアロン アプリケーションでは、ADV7511 HDMI トランスミッターおよび ADV7611 HDMI レシーバー (FMC がある場合) がスタートアップ時に初期化されますが、いずれか一方の Linux アプリケーションを使用する場合は、FSBL 内で初期化が実行されます。

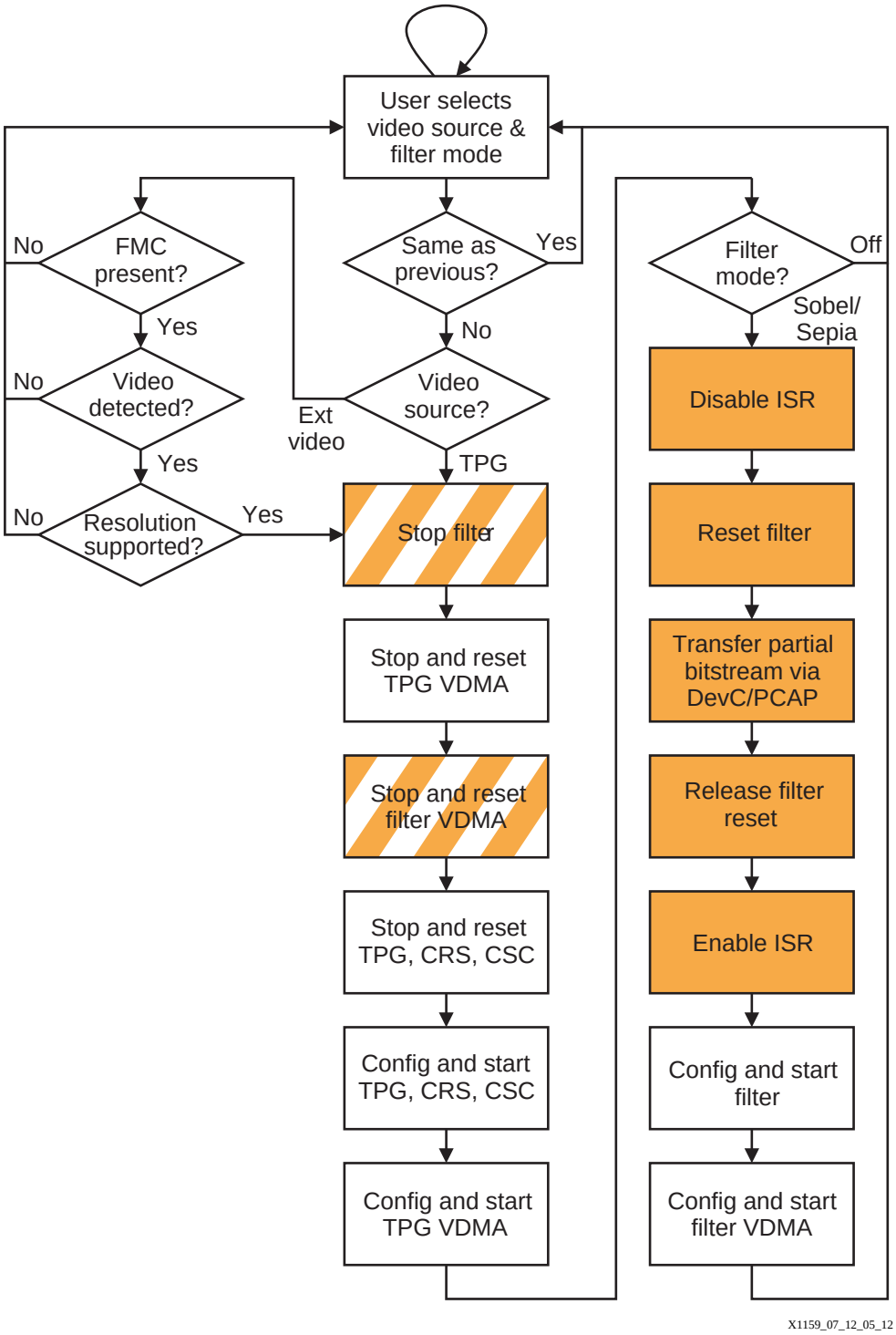

図 7 : ソフトウェア制御フロー

スタートアップ時の初期化が完了すると、ソフトウェア アプリケーションはコマンド ラインまたは GUI からのユーザー入力を待機します。その後のスタンドアロン アプリケーションのソフトウェア制御フローは、図 7 のようになります。Linux アプリケーションでも同様のフローが適用されます。オレンジ色で塗りつぶされたボックスは、パーシャル リコンフィギュレーション フロー専用となるため、通

常はこの部分を飛ばします。オレンジ色のストライプ ボックスは、PR 制御フロー専用ではありませんが、パーシャル リコンフィギュレーションを開始する前に AXI4-Streaming インターフェイスで保留中のトランザクションがないことを確認するために非常に重要な部分です。ソフトウェアでこのことが保証されない場合、ハードウェア グルー ロジックを使用し、パーティション境界で AXI バス インターフェイスを切り離す方法もあります。

PCAP インターフェイスを介してパーシャル ビットストリームを転送する前には、ユーザーが割り込みコントローラーで割り込みハンドラーが無効に設定されていることを確認する必要があります。無効に設定されていないと、リコンフィギュレーションが不正な割り込みをトリガーし、ロードが完了していない RM 内のハードウェアへ ISR がアクセスしてしまう可能性があります。フィルターを既知の適切な状態にするために、パーシャル リコンフィギュレーションの前、途中、および直後にリセットをアサートすることを推奨します。PCAP を介して DMA 転送を初期化する場合は、DDR メモリ内のパーシャル ビットストリームが格納されているアドレスまたはバッファー、およびビットストリームのサイズ (PROMgen ツールから取得) を示す値を、対応するドライバー関数へ渡す必要があります。ビットストリームの転送が完了するとリセットがリリースされ、ISR が有効になり、フィルターが設定されて動作可能な状態となります。

フィルター エンジン ドライバー

Linux の場合、フィルターの開始/停止関数が自動的に ISR を有効化/無効化します。ISR はドライバーに含まれ、カーネル モードで実行します。スタンドアロンの場合は、ISR の有効化/無効化関数が別に呼び出され、ISR はドライバーではなくユーザー アプリケーションに含まれます。フィルターのリセット関数は、PS GPIO ドライバーを使用してインプリメントされます。スタンドアロン フィルター ドライバーは Vivado HLS で生成され、すぐに利用できます。『Vivado HLS ツールを使用した Zynq All Programmable SoC での Sobel フィルターの実装』(XAPP890) [参照 2] では、生成されたスタンドアロン ドライバーを基に Linux ドライバーを記述する方法について説明しています。Linux ドライバーは Linux カーネルのパッチとして提供され、スタンドアロン ドライバーは SDK ユーザー リポジトリ内に提供されます。Sobel フィルター インプリメンテーションと Sepia フィルター インプリメンテーションでは同じレジスタ インターフェイスとアドレス マップを使用するため、両方に同一のドライバーが使用できます。

デバイス コンフィギュレーション ドライバー

Linux DevC デバイス ドライバーは、Linux で提供される仮想ファイル システム sysfs 上に構築され、カーネルからユーザー空間へデバイスやドライバーの情報をエクスポートします。PCAP インターフェイス経由でパーシャル ビットストリームを転送する前に、`is_partial_bitstream` デバイス属性が 1 に設定されている必要があります。DevC ノード上での書き込みファイル動作が、パーシャル ビットストリームを転送に使用されます。DevC の書き込みドライバー関数が、DMA トランザクションを開始し、その後、転送が完了したことを示す割り込み信号が現れるまで待機します。パーシャル リコンフィギュレーションは、ユーザー アプリケーション外でシェルを使用して開始することも可能です。たとえば、次のように入力します。

```
% echo 1 > /sys/devices/amba.0/f8007000.devcfg/is_partial_bitstream
% cat /mnt/sobel.bin > /dev/xdevcfg
```

通常、パーシャル リコンフィギュレーションを実行する前後に、システムが指定した状態になっていることを確認するためにユーザー アプリケーションを実行する必要があることに留意してください。スタンドアロンでは、シンプルな関数呼び出しによってビットストリームが転送されます。DMA トランザクションおよび PCAP 転送の完了を判断するためには、ポーリングが使用されます。

パフォーマンスの測定基準

リファレンス デザインでは、メモリ スループットとコンフィギュレーション時間という 2 つの基準を用いてパフォーマンスを測定しています。

メモリ スループット

1920 x 1080 @60fps のビデオ解像度 (1080p60) の場合、フィルターを無効に設定したリファレンス デザインの総メモリ スループットは、次のようになります。

(1920 x 1080 x 4 バイト x 60Hz) x 2 = 497MB/s x 2 ≈ 1GB/s

フィルターを有効に設定した場合は、次のようになります。

(1920 x 1080 x 4 バイト x 60Hz) x 4 = 497MB/s x 4 ≈ 2GB/s

GUI ベースの Linux アプリケーションを使用する場合は、GUI レイヤーでさらに 125MB/s が使用されます。GUI では HP0 および HP1 ポートのパフォーマンス測定値が表示されるため、リアルタイムにメモリ スループットをモニタリングできます。533MHz (1066MHz データ レート) クロックで駆動される 32 ビット幅の DDR3 メモリの理論上のメモリ スループットは、次のようになります。

4 バイト x 1066MHz = 4.264GB/s

したがって、このデザインの最大メモリ使用率は約 50% です。

(2GB/s + 0.125GB/s) /4.264 ≈ 0.5

コンフィギュレーション時間

フレームの位置や内容によってわずかに変動するリコンフィギャラブル フレーム数に伴ってビットストリーム サイズが大きくなると、コンフィギュレーション時間は、ほぼ線形的に増加します。PCAP インターフェイスは、32 ビット幅でクロック周波数は 100MHz です。

表 2 では、デザインのフル ビットストリームとパーシャル ビットストリームのサイズ、およびスタンドアロンと Linux アプリケーションで測定されたコンフィギュレーション時間を比較しています。コンフィギュレーション時間は、DevC DMA 転送ドライバーの関数呼び出しが開始されてから終了するまでの時間を計測しています。

表 2 : フル ビットストリームとパーシャル ビットストリームのコンフィギュレーション時間

XC7Z020 デバイス	フル ビットストリーム	パーシャル ビットストリーム
ビットストリーム サイズ	4,045,564 バイト	134,392 バイト
コンフィギュレーション時間 - スタンドアロン	32ms(1)	1,060µs(1)
コンフィギュレーション時間 - Linux	44ms(1)	2,000µs(1)

注記 :

1. 測定値は平均を示しています。

ビデオ アプリケーションでは、リアルタイム処理の要件が厳しく、ビデオ フレームが 1 つ欠けるだけでも、安全性を重視するシステムでは許容されないことがあります。そのようなリアルタイム システムでパーシャル リコンフィギュレーションが実行可能かどうかを判断するには、2 つの連続するビデオ フレーム間の時間を測定し、それがパーシャル リコンフィギュレーションを実行して、その後ビデオ パイプラインを再開するのに十分な長さであることを確認する必要があります。

1 つのビデオ フレーム期間には、アクティブ期間と非アクティブ期間 (ブランキング期間) があります。アクティブ期間中に、ビデオ画素データが転送されます。つまり、従来の CRT モニターでは、水平および垂直ブランキング期間を使用して、行の終わりから次行の開始 (水平ブランキング)、またはフレームの終わり (右下) から次のフレームの開始 (左上) へ電子ビームを再配置していました。図 8 に、ビデオ フレームのアクティブ期間とブランキング期間を示します。ビデオ タイミング パラメーターの詳細は、『AXI4-Stream Video IP およびシステム デザイン ガイド』(UG934) [参照 10] を参照してください。

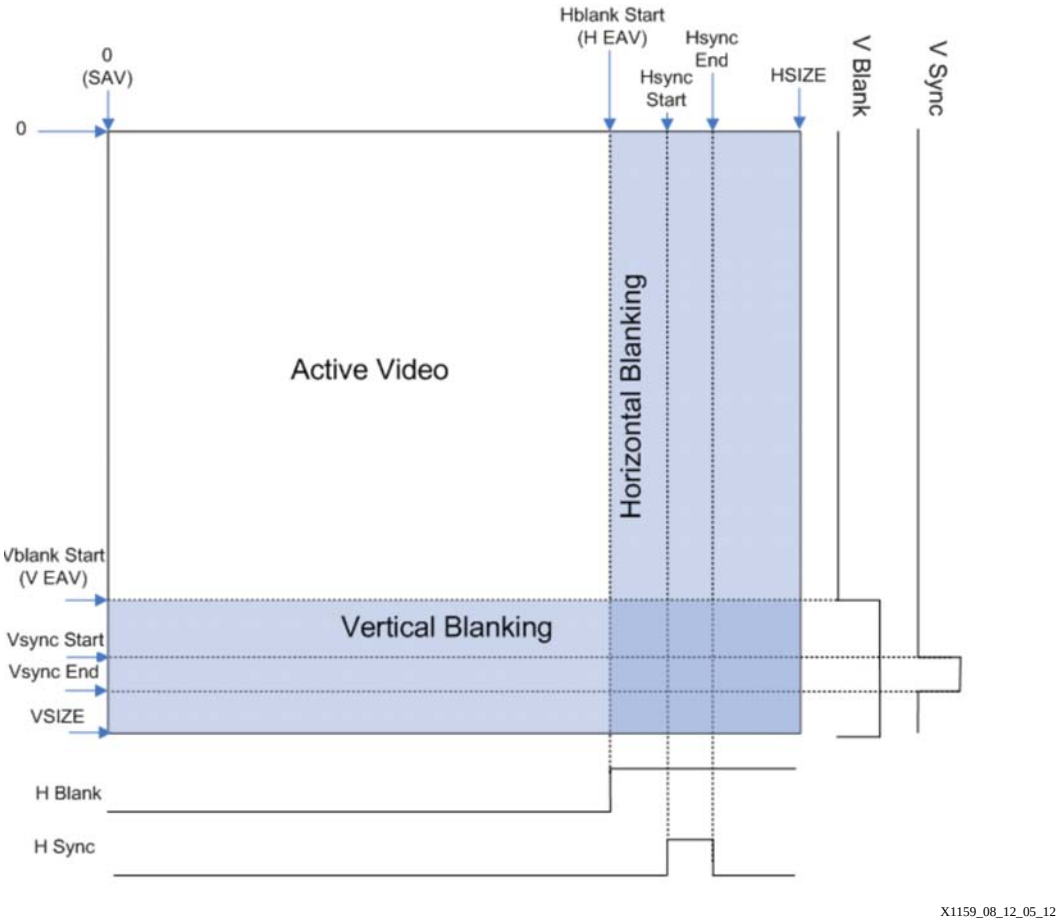

図 8 : ビデオ タイミング パラメーター

1080p60 ビデオ ストリームの場合、垂直ブランキングはラインあたり 45 ピクセルに相当し、水平ブランキングはビデオ フレームあたり 280 ラインに相当します。必要なビデオ クロック周波数は、次のように計算されます。

(1920 + 280) x (1080 + 45) x 60Hz = 148.5MHz

ビデオ アクセラレーターのパーシャル リコンフィギュレーションは、フレーム f-1 の最後のピクセルとフレーム f の最後のピクセル間 (垂直ブランキング期間と 1 水平ブランキング期間に相当) で行われることが理想的です。有効なタイム ウィンドウは、次のようになります。

(1 x 45 + 280 x (1080 + 45)) / 148.5 MHz ≈ 2.1 ms

パーシャル リコンフィギュレーション時間は、使用するソフトウェア アプリケーションと OS によって異なります (表 2 参照)。スタンドアロンの場合は、有効とされる 2.1ms よりかなり短くなります。ビデオ モードを切り換える場合は、ビデオ パイプライン IP コアをリセットおよびコンフィギュレーションするために、追加のオーバーヘッドが必要です。リファレンス デザインは、前述したリアルタイム ビデオ アプリケーションの要件を満たすことを目的として設計されたものではないことに留意してください。フレーム間でフレームの損失が生じないように、ブランキング期間中のみビデオ モードを切り替えるという特別な配慮は行っていません。しかしながら、有効な 2.1ms タイム ウィンドウとリファレンス デザインで測定されたタイミング値を考える限り、低レイテンシのベアメタルあるいはリアルタイム OS (RTOS) を使用した場合、システムは実現可能です。

リファレンス デザインをインプリメントおよび実行

リファレンス デザインのインプリメントと実行に関する詳細およびチュートリアルは、Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイトを参照してください。

http://Wiki.xilinx.com/zynq-pr-rd

構築済みリファレンス デザインを実行する

次の情報については、Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイト [参照 6] のセクション 7 を参照してください。

- ZC702 ハードウェア プラットフォームのセットアップ
- リファレンス デザインの実行および動作

ソフトウェア ツールおよびシステムの要件

次の情報については、Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイト [参照 6] のセクション 1 を参照してください。

- ハードウェア/ソフトウェアの要件およびライセンス取得
- デザインの概要およびプロジェクトのディレクトリ構造

ハードウェアを構築する

次の情報については、Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイト [参照 6] のセクション 2、3、および 4 を参照してください。

- Sobel/Sepia フィルター IP コアを生成するための Vivado HLS 設計フロー ガイド
- PlanAhead/XPS のシステム設計フロー ガイド
- PlanAhead パーシャル リコンフィギュレーションの設計フロー ガイド

ソフトウェアを構築する

次の情報については、Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイト [参照 6] のセクション 5 および 7 を参照してください。

- U-boot ブート ローダーおよび Linux カーネルのコンパイル フロー ガイド
- FSBL およびスタンドアロン/Linux ソフトウェア アプリケーションのコンパイルおよびスタンドアロン/Linux ブート イメージ作成における SDK フロー ガイド

リファレンス デザイン

リファレンス デザインは、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=199619

表 3 に、リファレンス デザインの詳細を示します。

表 3 : リファレンス デザインの詳細

パラメーター	内容
全般	
開発元	ザイリンクス
ターゲット デバイス (ステッピング レベル、ES、プロダクション、スピード グレード)	Zynq-7000 AP SoC (ES)
ソース コードの提供	あり
ソース コードの形式	VHDL、Verilog、C (一部は暗号化済み)
既存のアプリケーション ノート / リファレンス デザイン、CORE Generator ツール、サードパーティからデザインへのコード /IP の使用	あり
シミュレーション	
論理シミュレーションの実施	N/A
タイミング シミュレーションの実施	N/A
論理およびタイミング シミュレーションでのテストベンチの利用	N/A
テストベンチの形式	N/A
使用したシミュレータ / バージョン	N/A
SPICE/IBIS シミュレーションの実施	N/A
インプリメンテーション	
使用した合成ソフトウェア ツール / バージョン	Vivado HLS 2012.4、XST 14.4
使用したインプリメンテーション ツール / バージョン	ISE Design Suite 14.4 System Edition
スタティック タイミング解析の実施	はい
ハードウェア検証	
ハードウェア検証の実施	はい
検証に使用したハードウェア プラットフォーム	ZC702 評価ボード

表 4 に、Sobel コンフィギュレーションのデバイス使用率を示します。Sobel コンフィギュレーションのデバイス使用率は、Sepia コンフィギュレーションの使用率より高くなります。Sepia コンフィギュレーションの場合は、これらより少し低い値になります。

表 4 : デバイスのリソース使用率

項目	値
デバイス	XC7Z020
デバイス スピード	-2
グレード パッケージ	CLG484
スライス レジスタ	24,190 (22%)
配置済みスライス	9,049 (68%)
スライス LUT	20,185 (37%)

表 4 : デバイスのリソース使用率

項目	値
I/O	42 (21%)
RAMB36E1	21 (15%)
RAMB18E1	15 (5%)
DSP48E1	18 (8%)

参考資料

次の文書は、このアプリケーション ノートに役立つ補足資料です。

1. UG925 :『Zynq-7000 All Programmable SoC ZC702 Base ターゲット リファレンス デザイン (ISE Design Suite 14.4) ユーザー ガイド』
2. XAPP890 :『 ZC702 ボードを使用したゾーベル エッジ検出フィルターのインプリメント』
3. ザイリンクスの Zynq-7000 SoC ZC702 評価キットのウェブ ページ
   http://japan.xilinx.com/products/boards-and-kits/EK-Z7-ZC702-G.htm
4. HDMI Input/Output FMC モジュールのウェブ ページ
   http://www.em.avnet.com/en-us/design/drc/Pages/HDMI-Input-Output-FMC-module.aspx
5. UG702 :『パーシャル リコンフィギュレーション ユーザー ガイド』
6. Zynq-7000 パーシャル リコンフィギュレーション リファレンス デザイン Wiki ウェブサイト
   http://Wiki.xilinx.com/zynq-pr-rd
7. UG585 :『Zynq-7000 AP SoC テクニカル リファレンス マニュアル (TRM)』
8. UG821 :『Zynq-7000 EPP ソフトウェア開発者向けガイド』
9. UG470 :『7 シリーズ FPGA コンフィギュレーション ユーザー ガイド』
10. UG934 :『AXI4-Stream Video IP およびシステム デザイン ガイド 』
11. UG744 :『プロセッサペリフェラルのパーシャル リコンフィギュレーション』
12. Zynq-7000 Base ターゲット リファレンス デザイン 14.4 の Wiki ウェブサイト
    http://wiki.xilinx.com/zynq-base-trd-14-4
13. パーシャル リコンフィギュレーションのウェブサイト
    http://japan.xilinx.com/tools/partial-reconfiguration

リファレンス デザインで使用される IP コアの詳細は、次のウェブページを参照してください。

14. AXI Interconnect
    http://japan.xilinx.com/products/intellectual-property/axi_interconnect.htm
15. AXI Video DMA (VDMA)
    http://japan.xilinx.com/products/intellectual-property/axi_video_dma.htm
16. Xylon Compact Video Controller (logiCVC)
    http://japan.xilinx.com/products/intellectual-property/logiCVC.htm
17. AXI Performance Monitor
    http://japan.xilinx.com/products/intellectual-property/axi_perf_mon.htm
18. Video Timing Controller (VTC)
    http://japan.xilinx.com/products/intellectual-property/EF-DI-VID-TIMING.htm
19. Test Pattern Generator (TPG)
    http://japan.xilinx.com/products/intellectual-property/tpg.htm
20. Chroma Resampler (CRS)

http://japan.xilinx.com/products/intellectual-property/EF-DI-CHROM-RESAMP.htm

21. YCrCb to RGB Color-Space Converter (CSC)
    http://japan.xilinx.com/products/intellectual-property/YCrCb_to_RGB.htm
22. Video In to AXI4-Stream
    http://japan.xilinx.com/products/intellectual-property/video_in_to_axi4_stream.htm

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 1 月 21 日	1.0	初版

Notice of Disclaimer