UltraFast 設計手法
クイック リファレンス ガイド (UG1231)

概要

UltraFast™ 設計手法はザイリンクスの推奨するベスト プラクティスを集めたもので、使用すると、生産性を最大限に高め、複雑な電子システム (エンベデッド プロセッサ サブシステム、アナログおよびデジタル プロセッシング、高速コネクティビティ、ネットワーク プロセッシングなど) のデザイン イテレーションを削減できます。詳細は、『UltraFast 設計手法ガイド (Vivado Design Suite 用)』(UG949) を参照してください。

『UltraFast 設計手法チェックリスト』(XTP301) には、デザインの決定事項が下流の工程に影響する典型的な分野に関してよくある質問が含まれており、気づかなかったり無視されがちな問題に気付くことができます。関連資料へのリンクも含まれます。チェックリストは、ザイリンクス ウェブサイトから入手できます。また、Xilinx Documentation Navigator ツール (DocNav) 内でも使用できます (英語版のみ)。

このクイック リファレンス ガイドでは、システム統合およびデザイン インプリメンテーションをすばやく実行し、ザイリンクス デバイスおよびツールを最大限に活用する設計手法手順について説明します。関連資料へのリンクも含みます。この資料で説明される主な設計タスクは、次のとおりです。

- ボードおよびデバイス プランニング
- デザイン入力およびインプリメンテーション
- 最上位デザインの検証
- デザイン解析
- デザイン クロージャ

すべてのデザイン ハブおよび特定の資料へのリンクは、Xilinx Documentation Navigator ツール (DocNav) の「UltraFast 設計手法 - システム レベル デザイン フロー」を参照してください。

XILINX
ALL PROGRAMMABLE™

ボードおよびデバイス プランニング

PCB 設計

主なインターフェイスの確認
- パーツの向きと主なインターフェイスを検証

PCB レイアウトの確認
- メモリ インターフェイス チェックリストおよびトランシーバー チェックリストを実行
- PCB レイアウトの推奨事項に従う
- FPGA 設計者が最終的な FPGA のピン配置をサインオフ

回路図の確認
- PCB チェックリストの見直し
- PDS、コンフィギュレーション、電源のチェック
- コンフィギュレーション前、中、後の I/O ステートを検証

製造およびテスト
- テスト I/O プロジェクトを使用してコンフィギュレーション シーケンス、電源、I/O パフォーマンスを確認

参照:
UG949: ボードおよびデバイス プランニング
PCB デザイン チェックリスト
メモリ インターフェイス IP デザイン チェックリスト
回路図デザイン チェックリスト

FPGA 設計

デバイスのピン配置の解析
- トランシーバーおよびボンディングされた I/O 位置を確認
- SSI テクノロジの I/O プランニングを確認
- パーツの向きと主なインターフェイスを検証

主なインターフェイスの I/O ピン配置の定義
- I/O プランニング プロジェクトを作成
- メモリ コントローラー、GT、PCIe® テクノロジの位置を定義して検証
- クロッキング スケルトンの構築
- 接続された IP 間のフロアプランでの距離を最小限に抑える

最終ピン配置の定義
- 最終 I/O プロジェクトにインターフェイス プロジェクトを統合
- DRC および SSN 解析を検証
- インプリメントしてクロッキングおよび I/O ルールをチェック
- 最終 I/O プロジェクトを使用してプロダクション テスト

消費電力の見積もり
- Xilinx Power Estimator (XPE) を使用して消費電力バジェットと熱マージンを決定
- 以前のデザインの知識を生かしてトグル レートを適用

参照:
UG949: ボードおよびデバイス プランニング
消費電力の見積もりおよび最適化デザイン ハブ
I/O およびクロック プランニング デザイン ハブ

デザイン入力およびインプリメンテーション

ロジック設計

適切なデザイン階層の定義
- グローバルな配置とフロアプランを実行しやすいように関連する階層を定義
- I/O およびクロック コンポーネントを最上位近くに挿入
- 主な階層境界にレジスタを追加
- IP を生成し、ターゲット デバイスの使用率を確認

RTL サブモジュールの構築と確認
- デザインが RTL コーディング ガイドラインに従っていることを確認
- DSP およびメモリの周囲に十分なレジスタを追加
- 制御信号は絶対不可欠な場合にのみ使用
- 合成属性を使用して最終的なロジック マッピングを制御
- シンプルなタイミング制約を作成してタイミングの見積もりを確認し、ロジック レベル数の多すぎるパスに対処
- 合成ログ ファイル、使用率レポート、エラボレーション済みビューを確認し、最適でないマッピングを検出
- 設計手法および RTL チェックを実行して問題を確認
- アウト オブ コンテキスト (OOC) モードのサブモジュールをインプリメントし、インプリメント後のパフォーマンスを検証
- 使用率と消費電力を元のバジェットと比較
- デザインをシミュレーションして機能を検証

最上位デザインのアセンブルと検証
- 最上位 RTL デザインを合成して、すべての接続性問題を解決
- 最上位の使用率とクロッキング ガイドラインを確認
- 最上位制約を作成して検証
- RTL と制約を繰り返し実行し、設計手法および DRC問題を修正し、タイミングを満たす
- インプリメンテーションに進む

参照:
UG949: デザイン入力およびインプリメンテーション
IP を使用した設計デザイン ハブ
IP インテグレーターの使用デザイン ハブ
論理合成デザイン ハブ
デザイン制約の適用デザイン ハブ
インプリメンテーション デザイン ハブ

UG1231 (v2017.2) 2017 年 6 月 7 日

UltraFast 設計手法
クイック リファレンス ガイド (UG1231)

最上位制約の検証

ベースライン制約の作成

- ほとんどのブロックおよび主な IP が使用可能になった後タイミング クロージャが実現可能かどうかを早期に検証
- 必須の制約のみを指定:
  - すべての IP 制約を使用
  - 現実的なプライマリおよび生成クロックを定義
  - すべてのクロック乗せ換え制約を定義
  - 必要な箇所にマルチサイクル パスを追加
  - この段階では I/O 制約は使用しない
- パス要件が妥当であるかどうかを確認
- フローの次の各段階で report_timing_summary を使用して WNS ≈ 0.0 ns であることを確認
  - 合成後
  - 配置前
  - 配線前後
- フローの早期にタイミング違反を解決
- RTL および合成で QoR (結果の品質) 問題を解決

タイミング制約の検証

- report_timing_summary または check_timing を実行して、すべてのクロックが定義され、すべてのレジスタ、入力ポート、出力ポートに制約が設定されていることを確認
- report_methodology を実行し、すべてのタイミングおよび XDC 問題を解決
- report_clock_interaction を実行し、各クロック ペアが妥当なパス要件で安全にタイミング解析されることを確認
- report_cdc を実行し、すべての非同期クロック乗せ換えパスに適切な制約が設定され、安全な同期回路が使用されていることを確認
- report_exceptions を実行し、重複したり、無視されたり、不十分であったりするタイミング例外を検出
- デザインを読み込んで制約を適用したときに表示されるクリティカル警告をすべて解決

参照:

UG949: デザイン クロージャ
- デザインの制約が適切かどうかのチェック
- ベースライン制約の作成

デザイン制約の適用デザイン ハブ
タイミング クロージャおよびデザイン解析デザイン ハブ

デザイン解析およびクロージャ

タイミング違反の根本的な原因の特定

- report_qor_suggestions で自動解析およびタイミング クロージャの推奨事項を試す
- report_timing_summary または report_design_analysis を使用してタイミング違反の根本的な原因を特定
- セットアップ パスの場合は次の原因でデータパス遅延が大きいものをチェック:
  - セル遅延が大きい (7 シリーズ > 25%、UltraScale デバイス > 50%)
  - ネット遅延が大きい (7 シリーズ > 75%、UltraScale デバイス > 50%)
- ホールド パスの場合はホールド要件 (> 0 ns) をチェック
- クロック スキューの大きいパス (> 500 ps)、クロックのばらつきが多いパス (> 200 ps)、またはその両方をチェック

ロジック遅延の削減

- 並列演算子または効率的な演算子を使用するように RTL を変更
- パイプライン レジスタを追加し合成リタイミングを使用
- DSP またはブロック RAM 出力にレジスタを追加
- SRL の SRL 入力、出力、またはその両方からレジスタを取り出す
- KEEP/DONT_TOUCH/MARK_DEBUG を削除して最適化がすべて実行されるようにする

ネット遅延の削減

- フロアプラン制約を確認して調整
- 次でレベル 4 を超える場合は密集を緩和
  - report_design_analysis の配置密集表
  - ログ ファイルの配線密集の初期見積もり

クロック スキューの削減

- カスケード バッファーの代わりにパラレル バッファーを使用
- 同じ入力または PLL からの同期クロック間に CLOCK_DELAY_GROUP を使用
- 非同期クロック間にタイミング例外を追加

クロックのばらつきの削減

- MMCM 設定を最適化
- UltraScale™ デバイスで BUFGCE_DIV を使用してクロックを分周

参照:

UG949: デザイン クロージャ
- タイミング レポートの理解
- タイミング違反の根本的な原因の特定

タイミング クロージャおよびデザイン解析デザイン ハブ

制御セットの削減

- 制御信号には MAX_FANOUT を使用しない
- 合成制御セットのしきい値を増加
- opt_design で同等の制御信号を統合

ファンアウトの大きいネットの最適化

- RTL で階層ベースのレジスタ複製を使用
- 複製に opt_design -hier_fanout_limit および place_design -fanout_opt を使用
- クリティカルでなければグローバル クロッキングに変更
- phys_opt_design を使用して強制的に複製

密集の緩和

- デバイス使用率を削減し、SLR 使用率のバランスを調整
- 配置の -directive オプションの AltSpreadLogic* または SSI_Spread* を試す
- report_design_analysis -complexity -congestion で密集したモジュールを検出
- 密集したモジュールで AlternateRoutability ブロックレベル合成ストラテジを試すか、opt_design で MUXF*/CARRY* を削減
- 密集した領域のクリティカルではないファンアウトの大きいネットにグローバル クロッキングを使用
- 以前の密集の低いインプリメンテーションの DSP とブロック RAM の配置を再利用

コンパイル フローの調整

- place_design -directive オプションの複数の設定を試す
- 最適なネットリストにするためにブロックレベル合成ストラテジを使用
- set_clock_uncertainty を使用して配置および物理最適化中のクリティカル クロックの制約を厳しくする
- デザインを少し変更した後、インクリメンタル コンパイルを使用して QoR を保ち、実行時間を短縮

消費電力の解析と最適化

- 動作、環境、プロセスに制約を設定
- power_opt を試して消費電力を削減
- ブロック RAM カスケードを最大限に使用

参照:

UG949: インプリメンテーションおよびデザイン クロージャ
- タイミング違反の解析および修正
- 一般的なタイミング クロージャ手法

インプリメンテーション デザイン ハブ
タイミング クロージャおよびデザイン解析デザイン ハブ