

階層を考慮した論理等価性チェック・フォーマル検証

シノプシス・フォーマル検証ツール Formality Hier-IQ 検証テクノロジー概要

今日の複雑なシステムオンチップ (SoC) 設計を実現するために設計チームでは多数の設計課題を抱えています。過去の実績を見ると、デザイン・インプリメンテーション工程においてデザインの整合性の課題を克服するため、エンジニアはボトムアップ階層論理等価性チェック手法を使用し、検証対象となる設計の規模と複雑さの低減をはかっていました。この手法は高いカバレッジを実現し、ゲートレベル・シミュレーションに比べ高速な手法です。しかし、SoC設計の厳しいタイミングと面積の制約条件を満たすため、設計者は、高度な合成および配線テクニックを使用する必要があります。このためボトムアップ検証フローは複雑さを極め、実用に耐え得る検証ができなくなります。

ボトムアップ検証フローを困難にする設計の変化

- 様々なレベルの階層間でロジックが移動され、変更を考慮しないと検証結果が不一致となるバウンダリの最適化 (例：インバージョン・プッシング)
- モジュール名のマッピングに悪影響を及ぼす設計フローにおけるブロック名の変更
- クロックピンのミスマッチの原因となる階層バウンダリをまたぐクロックポート付きのクロックツリー (コンペア・ポイントのマッピングエラーを引き起こす)
- 異なる設計段階で行う階層を取り除くセル平坦化 (各設計段階で行うセル平坦化により、ユーザはリファレンス設計のブロックの平坦化の結果きちんと検証できているかどうか確認するために、実行できなかった検証を特定する必要がある。網羅的な検証を実現するためにデザイン全体を平坦化しなければならない場合もある。そのため階層設計のメリットが失われる。)

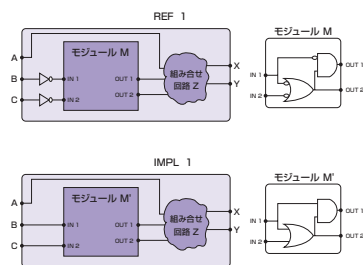


図1 バウンダリ最適化

図1は、バウンダリ最適化を行った設計事例を示しています。インプリメンテーション中の設計IMPL 1のモジュールM'に対するリファレンス設計REF 1のモジュールMの機能検証結果は、不等価となります。しかし、もし全体の設計において同じモジュールを観測した場合、REF 1のトップレベルにあるインバータは機能的に等価である点がわかります。

図2は、配線最適化が行われた同様の設計を示しています。より下位階層レベルのモジュールで検証を行うと、IMPL 2のモジュールL'に対するREF 2のモジュールLの機能は不等価となります。しかし、もし設計全体のなかで両方のモジュールを考慮した場合、ポートBはOUT 1に接続されポートCはOUT 2に接続されます。フラットな検証 (設計全体を考慮する) では、この設計は問題ありません。

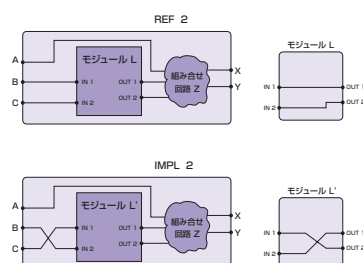


図2 配線最適化

図1や図2に示される設計変更を入念にドキュメント化しておくことにより、検証エンジニアは論理等価性チェック用制約条件の適切なセットアップを行えます。例えば (図2参照)、階層検証は、制約条件がモジュールLのポートINとモジュールL'のポートIN 2に適合するように追加されているかをチェックします。

しかし、設計者はブロックレベルの機能的な情報が全くない設計のRTLとネットリストを提供されることが頻繁にあります。図1や図2例のような階層内の変更に関する情報が検証エンジニアに提供されることはまれです。

階層検証で生じる問題を防ぐため、設計を単一のフラット設計として検証するケースもあるかもしれません。フラット検証の利点は、主要な入力間に関する全てのロジックがシングルパスで考慮される点にあります。そのため、フラット検証は最低限のセットアップだけですみます。しかし、フラット検証は実行が容易ですが、大規模なフルチップ設計を検証するには時間がかかり、非効率的です。

フラット検証を非効率的にする原因

- 膨大で複雑なロジック・コーンを検証するため、
 - 検証の収束が困難
 - 膨大な検証実行時間の増加を招く
- 膨大なメモリ容量が必要

図3は平坦化がどのように検証の非効率を引き起こしているかを示しています。この設計を階層的に検証するため、U_1をブラックボックスとして検証しトップレベルで検証を行います。ボトムアップ手法を使用し、ロジック・コーン1、2および3を個々に検証します。もしも、設計全体を平坦化し検証すると、ひとつの大規模なロジック・コーンになります。

設計を平坦化することは検証工程を困難にすることがあります。平坦化されたロジック・コーンの規模や複雑度により、検証実行時間が長くなり、メモリ容量が膨大に必要となってしまうこともあります。また、1つもしくは複数のロジック・コーンに算術演算器が含まれていると、設計の平坦化は検証の収束不能を引き起こします。

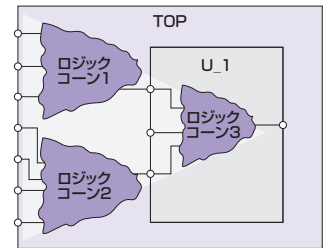


図3 フラット検証

最適なソリューション

設計全体を平坦化するかわり、インテリジェントな階層設計手法を検証ツールに組み込むとどうなるでしょうか？例えば、論理等価性を検証する必要があるので、図1のモジュールMとモジュールM'の検証を行うと、上位レベルのインパータがモジュールMの一部として考慮されるとします。組み合わせロジックZは別のロジック・コーンとしてみなされます。一方、図3の設計は階層的に検証され、そのため前後関係を考慮する必要はありません。もし検証ツールが階層検証に必要なときだけバウンダリを越えて検証を行うといったインテリジェントな手法を提供したら、設計者は両方の手法の利点を活用することができます：小規模なロジック・コーンを対象にした高速でメモリ効率の良い検証とわずかのセットアップで行える設計環境です。シノプシスのフォーマル検証ツールFormality Hier-IQ検証テクノロジーはそれを実現します。

Formalityは、設計全体の内容を考慮するあらゆるコンペア・ポイントを検証し、中間バウンダリを考慮し、自動的にロジックの規模と複雑さに制限を与え検証を行います。

中間ポイントをインテリジェントに選択することにより階層検証の高速化を実現、一方フラット検証がもたらす高精度で操作性の良い検証環境を提供します。さらに、Hier-IQテクノロジーは階層検証で使用する各階層レベルに対しメモリ消費量の制限を設ける独自のメソッドを採用しています。

Formality Hier-IQは以下の利点があります。

- バウンダリ最適化、モジュール名変更、クロック・ポート名変更、およびセル平坦化に伴うパラメータの追加変更不要な簡単セットアップ
- セットアップパラメータの不足によるエラー削除などのデバッグ作業の低減となるfalse-negativeが無い
- 階層検証と同等な検証パフォーマンスの実現
- 階層検証に匹敵する効率の良いメモリ使用量の実現

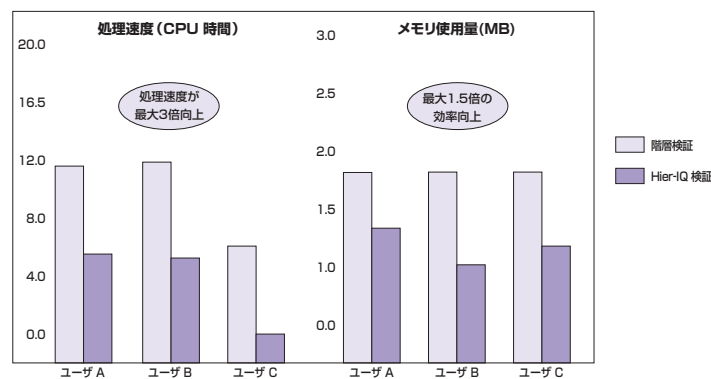


図4 Hier-IQテクノロジーはFormalityの実行速度およびメモリ使用量を飛躍的に改善しました。

■詳しい情報

Formalityの詳細につきましては、日本シノプシス(株)のホームページ [http://www.synopsys.co.jp] をご覧になるか、弊社までお問い合わせ下さい。

お問い合わせ先：

日本シノプシス株式会社

〒163-0420 東京都新宿区西新宿2-1-1 新宿三井ビルディング20階 TEL.03-3346-7030(代) FAX.03-3346-7050
〒531-0072 大阪府大阪市北区豊崎3-19-3 ピアスタワー13階 TEL.06-6359-8139(代) FAX.06-6359-8149

SYNOPTIS®